MCXE31B_1P55A

Mask Set Errata

Rev. 1.2 — 24 April 2025 Errata

1 Mask Set Errata for Mask 1P55A

1.1 Revision History

This report applies to mask 1P55A for these products:

- MCXE31BMPB

Table 1. Revision History

Revision	Release Date	Significant Changes
1.2	4/2025	The following errata were revised. • ERR052645 • ERR052438 • ERR052403
1.1	3/2025	The following errata were added. • ERR052645 • ERR052558 • ERR052403
1.0	1/2025	Initial Revision

1.2 Errata and Information Summary

Table 2. Errata and Information Summary

Erratum ID	Erratum Title
ERR011573	Cortex-M7: Speculative accesses might be performed to memory unmapped in MPU.
ERR050454	eMIOS: The eMIOS channel in OPWMB and OPWMCB mode doesn't operate correctly at reload on multiple events (MCB mode channel).
ERR050456	LPSPI: Reset to fifo does not work as expected
ERR050519	FCCU: EOUT timer reloading with disabled faults
ERR050575	eMIOS: Any Unified Channel running in OPWMCB mode may function improperly if the lead or trail dead time insertion features is used and its timebase is generated by Unified channel in MCB mode
ERR050583	MC_CGM: Functional reset during Clock dividers update can result in a Power on Reset sequence
ERR050593	XRDC: DERRLOCn register may not capture the PAC and the MRC instance correctly in case of error violation
ERR050595	GMAC/EMAC: Incorrect pps output generation on target time error
ERR050597	GMAC/EMAC: Transmit MAC management counters (MMC) updated incorrectly during frame preemption
ERR050608	[MSCM] Enabling address integrity check by programming the ENEDC register bits can results in invalid safety integrity error and fault to FCCU

Table 2. Errata and Information Summary*...continued*

Erratum ID	Erratum Title
ERR050609	PFLASH: PFCR4[DERR_SUP] may not work as expected
ERR050705	GMAC/EMAC: Head-Of-Line blocking error due to incorrect packet size when gates of gate control list (GCL) are closed
ERR050706	GMAC/EMAC: MAC receiver incorrectly discards the received packets when preamble byte does not precede SFD or SMD
ERR050707	GMAC/EMAC: Incorrect Handling of Application Bus Error in Certain Boundary Conditions
ERR050727	Core: Data corruption for load following Store-Exclusive.
ERR050729	Core: ECC error causes data corruption when the data cache error bank registers are locked.
ERR050763	PIT: RTI_LDVAL_STAT not reliable in Dynamic Loading mode
ERR050875	CoreSight: AHB-AP can issue transactions where HADDR[1:0] is not aligned to HSIZE on the AHB
ERR050887	Coresight: CSTPIU fails to output sync after the pattern generator is disabled in Normal mode
ERR051040	ITCM/DTCM: On the TCM backdoor accesses, burst termination (via MRC) due to entering protected region within the burst leads to an erroneous update of the protected region accessed by the burst.
ERR051046	Core: CTI might generate interrupts even when DBGENCTRL[CDBGEN] is low.
ERR051061	PFLASH: Read-While-Write to the same block may return incorrect read data
ERR051127	PFLASH: Flash read during array integrity may return incorrect read data
ERR051134	When cores are in lockstep configuration and low power debug mode the Cortex-M7_1 is not halted while Cortex-M7_0 is in standby mode.
ERR051222	GMAC/EMAC: Un-Correctable FSM Timeout Safety Interrupt incorrectly getting generated due to long waiting FSM states.
ERR051421	SAI: Synchronous mode with bypass is not supported
ERR051588	LPSPI:Reset transmit FIFO after FIFO underrun by LPSPI Slave.
ERR051629	LPUART:Transmit Complete bit (STAT[TC]) is not set.
ERR052121	LPI2C: NACK Detect Flag can be set when IGNACK=1
ERR052226	SWT: Toggling watchdog enable may cause unexpected timeout in some boundary conditions
ERR052277	Cortex-M7: Can halt in an incorrect address when breakpoint and exception occurs simultaneously.
ERR052403	FlexCAN: CAN frame drops in Enhanced RX FIFO when message buffer (MB) is locked for more than 1 CAN frame time (33 us)
ERR052438	FlexCAN: CAN frame may drop when using Enhanced RX FIFO
ERR052460	Cortex-M7: A hang scenario can occur when a reserved read locked memory region is accessed by application cores
ERR052558	FlexCAN: Message buffer (MB) overrun status is cleared when reading Enhanced RX FIFO (ERF)
ERR052645	Flash: Incorrect read data may be returned from flash

2 Known Errata

ERR011573: Cortex-M7: Speculative accesses might be performed to memory unmapped in MPU.

Description

Arm errata 1013783-B

Fault Type: Programmer Cat B

Cortex-M7 can perform speculative memory accesses to Normal memory for various reasons. All other types of memory should never be subject to speculative accesses.

The memory attributes for a given address are defined by the settings of the MPU when it is enabled. Regions that are not mapped in the MPU do not have any explicit attributes and should not be subject to any speculative accesses.

Because of this erratum, Cortex-M7 can incorrectly perform speculative accesses to such unmapped regions.

Conditions:

To trigger this erratum, the data cache must be enabled and the MPU must be enabled with the default memory map disabled. That is:

• CCR.DC = 1; data cache is enabled.

• MPU_CTRL.ENABLE = 1; MPU is enabled.

• If MPU_CTRL.PRIVDEFNA = 1, then this erratum cannot occur from privileged mode.

• If MPU_CTRL.HFNMIENA = 1, then this erratum cannot occur from the NMI or HF handlers or exception handlers when FAULTMASK = 1.

In these situations, a PLD instruction targeting an unmapped region might result in an incorrect speculative access. The PLD instruction itself could be speculative because of branch prediction. Even a literal data value that corresponds to a PLD encoding could theoretically cause this issue. This makes it difficult to scan code to check if these conditions apply.

Therefore, Arm recommends that any software with the MPU and data cache configured as mentioned in the conditions above uses the workaround below.

Implications:

Processor execution is not directly affected by this erratum. The data returned from the speculative access is never used and if the access is inferred by the program, then an abort will be taken as required.

The only implications of this erratum are the access itself which should not have been performed. This might have an impact on memory regions with side-effects on reads or on memory which never returns a response on the bus.

Workaround

Instead of leaving memory unmapped, software should use MPU region 0 to cover all unmapped memory and make this region execute-never and inaccessible. That is, MPU_RASR0 should be programmed with:

• MPU_RASR0.ENABLE = 1; MPU region 0 enable.

• MPU_RASR0.SIZE = b11111; MPU region 0 size = 2^32 bytes to cover entire memory.

• MPU_RASR0.SRD = b00000000; All sub-regions enabled.

- MPU_RASR0.XN = 1; Execute-never to prevent instruction fetch.
- MPU_RASR0.AP = b000; No read or write access for any privilege level.
- MPU_RASR0.TEX = b000; Attributes = Strongly-ordered.
- MPU_RASR0.C = b0; Attributes = Strongly-ordered.
- MPU_RASR0.B = b0; Attributes = Strongly-ordered.

Note that the MPU supports addressing hitting in multiple regions with the highest numbered region taking priority.

Therefore, use of MPU region 0 in this way does not affect the existing organization and use of MPU regions.

ERR050454: eMIOS: The eMIOS channel in OPWMB and OPWMCB mode doesn’t operate correctly at reload on multiple events (MCB mode channel).

Description

If any local or global bus channel is programmed in MCB mode and register C2_n.UCRELDEL_INT field of this bus channel is programmed any value other than 0, then

1. Any OPWMB mode channel using above channel as bus source (MCB up) will miss PWM pulses at output.

2. Any OPWMCB mode channel using above channel as bus source (MCB up-down) will miss PWM pulses if used in

2a. LEAD dead time mode (B>0) or

2b. 100% duty cycle mode.

Workaround

If a local or global bus generated by EMIOS channel is working in MCB mode and Reload events for such a counter bus are generated at selected interval by programming C2_n.UCRELDEL_INT != 0

Then :

1. Channel working in OPWMB mode should not use such MCB mode bus as source.

2. Channel working in OPWMCB mode using such an MCB mode bus as source, should not be configured to work with dead time insertion on leading edge of PWM pulse.

3. Channel working in OPWMCB mode using such an MCB mode bus as source, should not be configured to generate 100% duty cycle.

ERR050456: LPSPI: Reset to fifo does not work as expected

Description

The user can reset the transmit FIFO using LPSPIn_CR[RTF] bit and can reset the receive FIFO using LPSPIn_CR[RRF].

However, resetting the FIFO using CR[RTF] and CR[RRF] does not clear the FIFO pointers completely.

Workaround

Either reset the entire module using LPSPIn_CR[RST] bit, or

For Receive FIFO reset , after resetting the FIFO using LPSPIn_CR[RRF],

reading the receive FIFO immediately after resetting the receive FIFO will reset the pointers correctly

In case of Transmit FIFO reset (LPSPIn_CR[RTF] software needs to then wait for the transfer to complete with SR[TCF] set to 1.

ERR050519: FCCU: EOUT timer reloading with disabled faults

Description

The FCCU EOUT timer (controlled by the register DELTA_T) gets reloaded every time a fault is detected when the FCCU is waiting for an earlier fault state to be cleared. While this is correct for an enabled fault, it also occurs in case the detected fault is not enabled.

This unexpected behavior occurs only when there is an earlier fault state to be cleared. It does not affect the internal processing of the fault state, only an external indication of this state via the FCCU error indication pins.

This unexpected behavior does not impact the internal processing of the device, but may result in an extended error signaling by the FCCU error indicator pins. This may cause external logic to perform an error recovery operation when a corresponding time limit is exceeded, if such a recovery mechanism is implemented.

In a worst case scenario this may reduce the availability of the device when the recovery operation is a reset or shutdown operation

Workaround

No immediate action is required, since the actual internal processing is correct and not affected. However the following guidelines are recommended:

• Usually all faults connected to the FCCU require some processing for functional safety reasons. Avoid disabling the processing of faults by the FCCU, unless there is a strong requirement. The above scenario cannot occur when there are no disabled faults, it is the intended behavior in case of an enabled fault.

• Clear the source of any enabled fault as one of the first steps when processing a fault, since this minimizes the fault state duration within the FCCU.

• Utilize a minimum value for the FCCU EOUT timer (DELTA_T), as this also minimizes the potential extension of the error indication time window by the FCCU. The maximum non-intentional extension of this time window after clearing the earlier enabled fault state is the duration of the configured FCCU EOUT timer delay.

• In case an external system is resetting the device or shuts it down in response to an error condition: Define a sufficiently large time for the grace period before this emergency operation is triggered. This can avoid a recovery operation caused by an unintended extension of the error indication time window.

ERR050575: eMIOS: Any Unified Channel running in OPWMCB mode may function improperly if the lead or trail dead time insertion features is used and its timebase is generated by Unified channel in MCB mode

Description

The Unified channel (UC) configured in Center Aligned Output Pulse Width Modulation Buffered (OPWMCB) mode is not working properly when:

1. It’s timebase is sourced from the UC configured in Modulus Counter Buffered (MCB) mode.

2. The lead or trail dead time insertion features is used.

3. Its channel prescaler is different than timebase channel prescaler.

Workaround

Channel configured in OPWMCB mode with lead or trail dead time insertion features enabled must have channel prescaler equal to the timebase channel prescaler configured in MCB mode.

ERR050583: MC_CGM: Functional reset during Clock dividers update can result in a Power on Reset sequence

Description

A clock dividers update or other scenarios may require to initiate a Halt Handshake.

Following is the procedure for updating the dividers using the common trigger update:

1. Configure the MUX_n_DIV_TRIG_CTRL register.

2. Wait for the update to finish (until MUX_n_DIV_UPD_STAT is 0).

3. Update the clock dividers (only 50%) per the divider update procedure.

4. After the divider update is finished, perform a write operation on the MUX_n_DIV_TRIG register.

5. Wait for the update to finish, that is, until MUX_n_DIV_UPD_STAT is 0. During this period, the following process takes place:

Halt handshake is initiated if configured in step 1 (MUX_n_DIV_TRIG_CTRL[HHEN] set to 1b1).

Clock dividers is updated only when AXBS is halted (that is, halt acknowledgement is received by MC_CGM). It is

initiated, else the dividers are updated at alignment.

After the clock dividers are updated, MUX_n_DIV_UPD_STAT is asserted to 0.

When the bit fields MUX_x_DIV_TRIG_CTRL[TCTL] and MUX_x_DIV_TRIG_CTRL[HHEN] are set to 1, then any

write operation on trigger register will assert (MUX_n_DIV_UPD_STAT). Once the dividers are updated and aligned

(MUX_n_DIV_UPD_STAT) will be deasserted.

The halt handshake happens starts when MUX_n_DIV_TRIG is written to trigger a divider update and goes on till MUX_n_DIV_UPD_STAT[DIV_STAT] again sets to zero signaling completion of halt handshake for the divider and the update to the divider value is completed

If between the duration of halt handshake start and end a functional reset occurs then the device may undergo a power on reset sequence rather than starting a functional reset sequence (when DCMROF20[POR_WDG_EN] = 1b1 (Reset value)).

Workaround

Below three conditions must be fulfilled

1 Prior to initiating a Halt Handshake the functional reset sources need to be demoted to interrupt (IRQ). This involves writing 1b1 to each individual bit (as applicable to device) MC_RGM.FERD[D_FCCU_RST] ,

MC_RGM.FERD[D_SWT0_RST] , MC_RGM.FERD[D_SWT1_RST] , MC_RGM.FERD[D_JTAG_RST] , MC_RGM.FERD[D_DEGUG_FUNC] bits.

2 A software functional reset sequence should not be applied during halt handshake (using MC_ME MODE_CONF[FUNC_RST])

Post completion of Halt handshake, the functional reset sources can again be made source of functional reset . This involves writing 0b0 to MC_RGM.FERD[D_FCCU_RST] , MC_RGM.FERD[D_SWT0_RST] , MC_RGM.FERD[D_SWT1_RST] , MC_RGM.FERD[D_JTAG_RST] , MC_RGM.FERD[D_DEGUG_FUNC] bits.

ERR050593: XRDC: DERRLOCn register may not capture the PAC and the MRC instance correctly in case of error violation

Description

XRDC provides access protection mechanism to protect all on-chip resources (registers,volatile- and non volatile memory areas and all other on-chip peripherals) against manipulation and/or unauthorized access from external or internal.

A write attempt by a non core master outside the defined ranges shall lead to an exception in case XRDC region is defined to prevent non core master access.The chip will generate bus error when XRDC policies are violated.

When either a memory region controller or a peripheral access controller detects a domain access violation, the address and attribute information of the offending access is captured. The DERRLOCn read-only registers provide additional information by signaling the instance number of the submodule for which the access violation(s) occurred.

It is recommended that the exception handler for XRDC policy violation begins by reading the HWCFG1 register to determine its domainID. Next, it should use the just-retrieved domainID to index into the DERRLOCn array. The resulting DERRLOCn value is then examined to determine

the instance number of the reporting MRC and/or PAC submodule.

XRDC implements access control and signals violations correctly but if there are multiple violation after the first violation successively then the DERRLOCn register is not able to provide the correct instance number of reporting MRC and/or PAC submodule.

Workaround

To ascertain the correct instance reporting MRC and/or PAC submodule that was source of access violation, follow the below software sequence

Read DID from HWCFG1 register to know masters DID

Read all DERR_W1_n registers to determine which all submodules failed (check EST bit)

Read DERR_W0_n registers for which DERR_W1_n[EST] bit is set

Write to DERR_W3_n corresponding to failing submodules to clear error and rearm them for subsequent captures

ERR050595: GMAC/EMAC: Incorrect pps output generation on target time error

Description

There are two scenarios.

1. When programmed target time is lesser than system time, then pulse per second (pps) output signal gets generated incorrectly despite target error.

2. loss of sub-nanosecond accuracy if time correction of nanosecond field enabled in binary rollover mode.

Workaround

Workaround for scenario#1:. Software should not program target time register with already past system time. Software must read current system time seconds register (MAC_System_Time_Seconds) and Nanoseconds register (MAC_System_Time_Nanoseconds) before programming target time seconds register (MAC_PPS(i)_Target_Time_Seconds) and Nanosecond registers (MAC_PPS(i)_Target_Time_Nanoseconds) where i can range from 0 to 3.

Workaround for scenario#2: Do not use Binary roll over mode for Nanosecond time update.

ERR050597: GMAC/EMAC: Transmit MAC management counters (MMC) updated incorrectly during frame preemption

Description

Accumulated byte counters would overflow when sum of bytes of all fragments of Frame Preempted packet exceeds the Jabber limit. This would happen only when frame preemption feature is enabled. This would result in incorrect value of transmit MAC management counters (MMC).

Workaround

The transmit MAC management counters (MMC) cannot be use when the frame pre-emption feature is enabled

ERR050608: [MSCM] Enabling address integrity check by programming the ENEDC register bits can results in invalid safety integrity error and fault to FCCU

Description

The ENEDC register has bits to enable ECC integrity check on address and data bus during an ongoing transactions

Application core, EMAC, HSE-B or eDMA3 can initiate transactions to QSPI, PFlash , AIPS0 , AIPS1, AIPS2 , SRAM , and BDRAM. When a bit is enabled in ENEDC register it will check the corresponding ECC information during the ongoing transaction for the respective Address or data ECC logic. As an example ENEDC[10 (Enable Address check P_FLASH_PORT1)] when set to 1b1 then an Address ECC at P1 port for PLFASH gets checked.

If this bit is set and an address error occurs on bus during the transactions, it is reported as a safety integrity error to the FCCU (NCF[1]) and DCMROD3[ADDR_EDC_ERR] register bit.

However in below scenarios an false integrity error may be reported on FCCU (NCF[1]) and DCMROD3 register bits (in other scenarios it will work fine)

If any of application core accesses the QSPI when QSPI Gasket is enabled (DCMROF20[5: QSPI_IAHB_BYPASS] = 1b0) and ENEDC[19 (Enable Address check QSPI)] is set to 1b1 , then an Address ECC error shall be incorrectly reported on FCCU (NCF[1]) and DCMROD3[ADDR_EDC_ERR] register.

If EMAC accesses the QSPI with settings that enable burst mode access and when QSPI Gasket is enabled (DCMROF20[5: QSPI_IAHB_BYPASS] = 1b0) and ENEDC[19 (Enable Address check QSPI)] is set to 1b1 , then an Address ECC error shall be incorrectly reported on FCCU (NCF[1]) and DCMROD3[ADDR_EDC_ERR] register.

If eDMA does an access with both TCDn_ATTR[SSIZE] = 3b101 or 3b110 and TCDn_ATTR[DSIZE] =3b101 or 3b110 from or to

a) Flash P1 port, eDMA3 (S0) gasket configuration is enabled(DCMROF20[3: DMA_AXBS_IAHB_BYPASS] = 1b0) and and ENEDC[10 (Enable Address check P_FLASH_PORT1)] = 1b1 , then an then an Address ECC error shall be incorrectly reported on FCCU (NCF[1]) and DCMROD3[ADDR_EDC_ERR]

b) SRAM0 or SRAM1 , eDMA3 (S0) gasket configuration is enabled(DCMROF20[3: DMA_AXBS_IAHB_BYPASS] = 1b0) and ENEDC[15 (Enable Address check PRAM1)] = 1b1 and/or ENEDC[13 (Enable Address check PRAM0)] = 1b1, then address ECC error shall be incorrectly reported on FCCU (NCF[1]) and DCMROD3[ADDR_EDC_ERR]

c) BDRAM (Backdoor TCM), eDMA3 (S0) gasket configuration is enabled (DCMROF20[3: DMA_AXBS_IAHB_BYPASS] = 1b0) and ENEDC[17 (Enable Address check TCM)] = 1b1 then address ECC error shall be incorrectly reported on FCCU (NCF[1] and DCMROD3[ADDR_EDC_ERR]

d) QSPI , either eDMA3 or QSPI gasket is enabled (eDMA3 (S0) gasket configuration is enabled (DCMROF20[3: DMA_AXBS_IAHB_BYPASS] = 1b0) or QSPI Gasket is enabled(Not in Bypass mode) (DCMROF20[5: QSPI_IAHB_BYPASS] = 1b0)), and ENEDC[19 (Enable Address check QSPI)] is set to 1b1 , then an Address ECC error shall be incorrectly reported on FCCU (NCF[1]) and DCMROD3[ADDR_EDC_ERR] register.

e) Any module connected to AIPS1, AIPS gasket is enabled (DCMROF20[6: AIPS_IAHB_BYPASS] = 1b0) and ENEDC[23 (Enable Address check AIPS1)] = 1b1 then address ECC error shall be incorrectly reported on FCCU (NCF[1] and DCMROD3[ADDR_EDC_ERR]

f) Any module connected to AIPS2, AIPS gasket is enabled (DCMROF20[6: AIPS_IAHB_BYPASS] = 1b0) and ENEDC[25 (Enable Address check AIPS2)] = 1b1 then address ECC error shall be incorrectly reported on FCCU (NCF[1] and DCMROD3[ADDR_EDC_ERR]

If CAAM and PKC within HSE-B access to

a) Flash P1 , HSE Gasket is enabled (DCMROF2[16: HSE_GSKT_BYPASS] = 1b0 and DCMROF21[20:19 HSE_CLK_MODE_OPTION] = 2b00 or 2b10 or 2b11) and ENEDC[10 (Enable Address check P_FLASH_PORT1)] =1b1 , then an then an Address ECC error shall be incorrectly reported on FCCU (NCF[1]) and DCMROD3[ADDR_EDC_ERR]

b) SRAM0 or SRAM1, HSE Gasket is enabled (DCMROF2[16: HSE_GSKT_BYPASS] = 1b0 and DCMROF21[20:19 HSE_CLK_MODE_OPTION] = 2b00 or 2b10 or 2b11) and ENEDC[15 (Enable Address check PRAM1)] = 1b1 and/or ENEDC[13 (Enable Address check PRAM0)] = 1b1, then address ECC error shall be incorrectly reported on FCCU (NCF[1]) and DCMROD3[ADDR_EDC_ERR]

c) BDRAM (Backdoor TCM), HSE Gasket is enabled (DCMROF2[16: HSE_GSKT_BYPASS] = 1b0 and DCMROF21[20:19 HSE_CLK_MODE_OPTION] = 2b00 or 2b10 or 2b11) and ENEDC[17 (Enable Address check TCM)] = 1b1 then address ECC error shall be incorrectly reported on FCCU (NCF[1] and DCMROD3[ADDR_EDC_ERR]

d) QSPI , either QSPI or HSE gasket is enabled (HSE Gasket is enabled (DCMROF2[16: HSE_GSKT_BYPASS] = 1b0 and DCMROF21[20:19 HSE_CLK_MODE_OPTION] = 2b00 or 2b10 or 2b11) or DCMROF20[5: QSPI_IAHB_BYPASS] = 1b0) and ENEDC[19 (Enable Address check QSPI)]= 1b1 , then an Address ECC error shall be incorrectly reported on FCCU (NCF[1]) and DCMROD3[ADDR_EDC_ERR] register.

e) Any module connected to AIPS0, HSE gasket is enabled (DCMROF2[16: HSE_GSKT_BYPASS] = 1b0 and DCMROF21[20:19 HSE_CLK_MODE_OPTION] = 2b00 or 2b10 or 2b11) and ENEDC[21 (Enable Address check AIPS0)] = 1b1 then address ECC error shall be incorrectly reported on FCCU (NCF[1] and DCMROD3[ADDR_EDC_ERR]

f) Any module connected to AIPS1, HSE or AIPS Gasket is enabled (DCMROF2[16: HSE_GSKT_BYPASS] = 1b0 and DCMROF21[20:19 HSE_CLK_MODE_OPTION] = 2b00 or 2b10 or 2b11) or DCMROF20[6:

AIPS_IAHB_BYPASS] = 1b0) and ENEDC[23 (Enable Address check AIPS1)] = 1b1 then address ECC error shall be incorrectly reported on FCCU (NCF[1] and DCMROD3[ADDR_EDC_ERR]

g) Any module connected to AIPS2, HSE or AIPS Gasket is enabled (DCMROF2[16: HSE_GSKT_BYPASS] = 1b0 and DCMROF21[20:19 HSE_CLK_MODE_OPTION] = 2b00 or 2b10 or 2b11) or DCMROF20[6: AIPS_IAHB_BYPASS] = 1b0) and ENEDC[25 (Enable Address check AIPS2)] = 1b1 then address ECC error shall be incorrectly reported on FCCU (NCF[1] and DCMROD3[ADDR_EDC_ERR]

Workaround

Do not enable address integrity on following bits:

ENEDC[10 (Enable Address check P_FLASH_PORT1)], ENEDC[13 (Enable Address check PRAM0)] , ENEDC[15 (Enable Address check PRAM1)] ,ENEDC[17 (Enable Address check TCM)] , ENEDC[19 (Enable Address check QSPI)] , ENEDC[21 (Enable Address check AIPS0)] , ENEDC[23 (Enable Address check AIPS1)] , and ENEDC[25 (Enable Address check AIPS2)]

ERR050609: PFLASH: PFCR4[DERR_SUP] may not work as expected

Description

The Data Error Suppression bit in the Platform Flash Memory Configuration 4 register (PFCR4[DERR_SUP]) is mainly intended for EEPROM emulation applications but not limited to this. The PFCR4[DERR_SUP]=1 flash controller setting doesn’t prevent FCCU (Fault Collection and Control Unit) error reaction in case of multi-bit ECC events when accessing the data flash. Therefore, if FCCU’s NCF[1] channel is configured according to safety recommendations then any ECC error while reading the data flash will generate a Functional Reset.

This may inhibit any possibility to repair data flash sectors in software, which may also result from an EEPROM emulation brown-out condition.

Workaround

There are two options, one having a larger impact on safety (impacting other safety features monitored through this NFC channel) and the other one doesn’t impact other safety features, but impacts this particular feature to a larger extent for a limited time.

Option A: Disable ECC error notification from gaskets while performing the flash check (clear bits DCMRWD5[CM7_1_AHBM_RDATA_EDC_ERR_EN] and DCMRWD5[CM7_0_AHBM_RDATA_EDC_ERR_EN]) to ensure FCCU is not receiving a single bit or multi-bit ECC error indication. This action will "emulate" the functionality of the PFCR4[DERR_SUP] =1. In this case, keep the FCCU reaction as Functional reset. This option "ignores" ECC errors which may be dangerous in the application SW, therefore this modification shall be limited to the time span when the data flash is read and multi-bit errors can be corrected (e.g. when processing EEPROM emulation data). If the flash check is ongoing and PFCR4[DERR_SUP]=1 then safety applications must not be running on CM7_0 and CM7_1. ECC error notifications can be reenabled on the fly for safety applications.

Option B: Change FCCU’s NCF[1] channel reaction to an interrupt while keeping the ECC single bit or multi-bit errors enabled. FHTI (Fault handling time Interval) for this channel needs to include additional time spent on handling the interrupt.

ERR050705: GMAC/EMAC: Head-Of-Line blocking error due to incorrect packet size when gates of gate control list (GCL) are closed

Description

Incorrect head of line blocking (HLBF) error is getting detected because a packet from a Transmit Queue is available for scheduling but the GCL gates for that Transmit Queue are closed for two complete iterations of GCL. Due to this incorrect detection of HLBF error, Packet is getting incorrectly dropped if the DDBF field of the MTL_EST_Control register is set to 0 and hence there is data loss.

Workaround

The software must set DDBF field of the MTL_EST_Control register to 1 and provide a new gate control list (GCL).

ERR050706: GMAC/EMAC: MAC receiver incorrectly discards the received packets when preamble byte does not precede SFD or SMD

Description

Received packet would be discarded if preamble byte doesn't precede the SFD,SMD-S or SMD-C byte when frame preemption is enabled. This happens because start of packet detection logic of MAC receiver incorrectly checks the preamble byte. The packets without Preamble or with corrupted Preamble byte before SFD is mostly an error case which has very low probability of occurrence.

Workaround

If remote MAC transmitter doesn't transmit preamble byte, then it should be configured to send at least one preamble byte preceding the SFD,SMD-S or SMD-C byte. In case of corruption of preamble byte during transmission, there is no software workaround. This means the applications that support packet re-transmission can re-transmit the packet.

ERR050707: GMAC/EMAC: Incorrect Handling of Application Bus Error in Certain Boundary Conditions

Description

Transmit or Receive DMA channel is not able to handle bus error gracefully under following scenario

1. When OSP (Operating on second packet) mode is enabled and the bus error occurs on the non-last descriptor of current packet because the transmit status of previous packet is still pending to be written in the descriptor memory. Transmit DMA doesn't read and discard the pending transmit status of these two packets.

2. When the bus error occurs on the first data word of transmit packet of the pre-emption queue then MAC transmitter would not transmit that single byte pre-emption fragment packet which encountered the bus error, and subsequent pre-emption and express packet.

3. When the Bus Error occurs on the first data word of the transmit packet, the start of packet is not indicated in the single dummy word, as expected. In this scenario, the start and end of packet indication is pushed to the Transmit Queue. This will lead to a situation where MTL Transmit Read Controller would not transmit subsequent packets from any transmit Queue.

Workaround

System must apply software reset to GMAC/EMAC using the soft reset configuration in the block and reconfigure the GMAC/EMAC

ERR050727: Core: Data corruption for load following Store-Exclusive.

Description

ARM errata 1315869

Affects: Cortex-M7, Cortex-M7 with FPU Fault Type: Programmer Category C

Fault Status: Present in r0p1, r0p2, r1p0, r1p1 and r1p2. Open.

A load that follows a Store-Exclusive to the same address might forward data from an earlier store, situated between the Load-Exclusive and the Store-Exclusive, and not the data from the Store-Exclusive.

Conditions:

The following sequence is required for this erratum to occur:

1. A load exclusive sets the local monitor.

2. A store to the wanted address

3. Any of the following instructions to the wanted address. This instruction must not fail either the local or global monitor check.

- STREXB.

- STREXH.

- STREX.

4. A load to the wanted address.

There must be at most one instruction between the Store-Exclusive and the load. All accesses must be to Shareable memory.

Implications:

Data corruption occurs when the load returns data from the older store instead of the newer Store-Exclusive.

Stores between a Load-Exclusive and Store-Exclusive are not expected in real code because such stores can always clear the local monitor in some implementations.

This impacts Cortex-M7 and Cortex-M7 with FPU.

Configurations Affected

All configurations are affected.

Workaround

No workaround is necessary.

ERR050729: Core: ECC error causes data corruption when the data cache error bank registers are locked.

Description

ARM errata 1267980

Affects: Cortex-M7, Cortex-M7 with FPU

Fault Type: Programmer Category C

Fault Status: Present in r0p1, r0p2, r1p0, r1p1 and r1p2. Open.

The data cache contains two error bank registers, DEBR0 and DEBR1. These registers store the locations in the cache that Error Correcting Code (ECC) errors affect and prevent future allocations to those locations.

Software can lock each DEBR and this prevents the DEBR from being automatically updated when a data cache ECC error is detected.

Because of this erratum, if both DEBR0 and DEBR1 are locked and an ECC error is detected on a cacheable store, then the store data is written onto the bus but not written into the data cache. This might result in the data cache containing stale data.

Conditions:

• DEBR0 and DEBR1 are locked.

• The wanted address has been allocated to the cache.

• A cacheable store to the wanted address looks up in the cache, and an ECC error is found in the cache set that the store addresses.

Implications:

This erratum can cause data corruption in the data cache.

Configurations Affected

All configurations with a data cache and ECC are affected.

Workaround

Software must avoid locking both error bank registers.

ERR050763: PIT: RTI_LDVAL_STAT not reliable in Dynamic Loading mode

Description

RTI_LDVAL_STAT register give the RTI timer load synchronization status. In the case of RTI timer load, it will take several cycles until this value is synchronized into the RTI clock domain. This register gives the status of the new loaded in the RTI timer load register. However in case of Dynamic loading of RTI timer load, this value might not be reliable.

Workaround

There are two options:

1. Do not use Dynamic loading feature of RTI.

2. If user needs to use dynamic loading of RTI, so to ensure the correct loading of RTI timer load register, read the current timer load value (RTI_CVAL) register after the current timer expires.

ERR050875: CoreSight: AHB-AP can issue transactions where HADDR[1:0] is not aligned to HSIZE on the AHB

Description

ARM errata 1624041

This erratum affects the following components:

• AHB Access Port.

The ARM Debug Interface v5 Architecture Specification specifies a TAR (Transfer Address Register) in the MEM-AP that holds the memory address to be accessed.

TAR[1:0] is used to drive HADDR[1:0] when accesses are made using the Data Read/Write register DRW.

When the AHB-AP is programmed to perform a word or half-word sized transaction the AHB-AP does not force HADDR[1:0] to be aligned to the access size. This can result in illegal AHB transactions that are not correctly aligned according to HSIZE if HADDR[1:0] is programmed with an unaligned value.

Conditions:

1) TAR[1:0] programmed with a value that is not aligned with the size programmed in the CSW register of the AHB-AP.

2) An access is initiated by an access to the Data Read/Write Register (DRW) in the AHB-AP.

Implications:

As a result of the programming conditions listed above, AHB-AP erroneously initiates an access on the AHB with HADDR[1:0] not aligned to the size on HSIZE. This might initiate an illegal AHB access.

Workaround

TAR[1:0] must be b00 for word accesses, TAR[0] must be b0 for half-word accesses.

Software program should program TAR with an address value that is aligned to transaction size being made.

ERR050887: Coresight: CSTPIU fails to output sync after the pattern generator is disabled in Normal mode

Description

Arm errata 341182

Affects: CoreSight SoC-400 - Perpetual

Fault Type: Programmer Category C

Fault Status: Present in: r0p0, r1p0, r2p0, r2p1, r3p0, r3p1, r3p2, Fixed in Open

This erratum affects the following components:

• Trace Port Interface Unit.

• CSTPIU and cxtpiu

• Component Revisions: r0p4, r0p5, r1p0

The TPIU includes a pattern generator that can be used to determine the operating behavior of the trace port and timing characteristics. This pattern generator includes a mode that transmits the test pattern for a specified number of cycles, and then reverts to transmitting normal trace data.

As a result of this erratum, when the TPIU is configured to operate in Normal Mode (FFCR.EnFCont==0), the synchronization sequence that is required between the test pattern and the trace data is not generated. Synchronization will be generated at later times as determined by the synchronization counter.

Conditions

The following conditions must all occur:

• The TPIU is configured in normal mode, FFCR.EnFCont==0

• The TPIU is configured with the formatter enabled, FFCR.EnFTC==1

• The pattern generator is enabled in timed mode, Current_test_pattern_mode.PTIMEEN==1

Implications

The timed mode of the TPIU is intended to permit the TPIU to transition between an initial synchronization sequence using the pattern generator and functional mode without any further programming intervention. If the synchronization sequence is not generated at the end of the test pattern, the trace port analyzer is unlikely to be able to capture the start of the trace stream correctly. Synchronization will be correctly inserted based on the value configured in the FSCR, once the specified number of frames of trace data have been output.

Workaround

This workaround requires software interaction to detect the completion of the test pattern sequence. In addition, any trace data present at the input to the TPIU is lost whilst the pattern generator is active. Any trace data present in the input to the TPIU before the formatter is re-enabled (and synchronization generated) will not be decompressible.

1) After enabling the pattern generator, set FFCR.StopOnFl==1 and FFCR.FOnMan==1.

2) Poll FFSR.FtStopped until 1 is read

3) Set FFCR.EnFTC==1

ERR051040: ITCM/DTCM: On the TCM backdoor accesses, burst termination (via MRC) due to entering protected region within the burst leads to an erroneous update of the protected region accessed by the burst.

Description

XRDC’s MRC is used to setup R/W protection to system memory including cores’ D-TCM and I-TCM through backdoor access. If core that doesn’t have access into the protected D-TCM/I-TCM region performs an access to that region, using a burst sequence with start address outside of the protected region and end address within the protected region, then D-TCM/I-TCM content within the protected region and overlaid by the burst transfer will be modified. The written values will be random.

The burst termination via the MRC is notified to the master via the bus error. However the un-intended region post the termination gets modified instead of being aborted.

Workaround

There are two possible workarounds:

1. The application should align the start and end addresses of the burst transfer within a region (protected or unprotected).

2. Disable the burst optimization (with impact to performance) by configuring IAHBCFGREG[TCM_DIS_WR_OPT] as 1 to avoid this issue.

ERR051046: Core: CTI might generate interrupts even when DBGENCTRL[CDBGEN] is low.

Description

ARM errata 585224

The Cortex-M7 integration level interrupt request outputs (CTIIRQ) are connected to the CTI outputs TRIGOUT[2:1]. These triggers should not be generated when the DBGENCTRL[CDBGEN] is low. This behavior should be guaranteed by tying off respective bits of the TODBGENSEL mask to 0. This mask has been tied off in the processor integration level to the incorrect value, which can result in the output being triggered regardless of the DBGENCTRL[CDBGEN] value.

Conditions:

• The CTIIRQ outputs are used by the system and enabled by respective bit in IRSPRCn (n refers to the irq number) (Refer to IRQ number from the CTI IRQ number provided in Interrupt map).

• Programming CTI such that it can generate triggers on these outputs (for more information on configuring the trigger outputs, see the ARM CoreSight™ SoC-400 Integration Manual).

• The processor interrupts are enabled and DBGENCTRL[CDBGEN] has to be driven low.

Implications:

Because of this erratum, debug events configured to trigger interrupts on the CTIIRQ output might generate them even if the DBGENCTRL[CDBGEN] is low.

Workaround

If the CTI interrupts are not used and debug is disabled, it is recommended to disable the respected CTI IRQ by software by disabling it in the appropriate bit in IRSPRCn for the respective core.

ERR051061: PFLASH: Read-While-Write to the same block may return incorrect read data

Description

While flash memory write (program or erase) is ongoing to a given block, other masters system can simultaneously perform a read from any other block. If Read-While-Write is performed to the same block, then system bus gets error response. However, when read occurs to the same block as being written, and this read occurs while there is previous system bus read outstanding, bus error response is not returned to the system. The scenario when this problem would occur is:

1. Master 1 initiates program/erase, considering this master won’t initiate read to same block

2. Master 2 initiates read to same block while program/erase is in progress

3. Master 3 initiates read to same block while AHB read request from master 2 is accepted but pending at flash controller

Workaround

There are 2 workaround options:

1) The software can read the Read-While-Write Event Error bit in the Module Configuration Status register (MCRS[RWE]) after system bus reads to make sure there's no error. In case of error, discard system read data and perform reads again.

2) Avoid Read-While-Write to the same block at the first place. The software shall use synchronization when sharing flash block by multiple masters similar to sharing other hardware resources.

ERR051127: PFLASH: Flash read during array integrity may return incorrect read data

Description

The Array integrity self-check can be performed to check the integrity of the embedded flash memory when it is in UTest mode. The Flash controller normally terminates AHB reads with an error response when array integrity self-check is in progress. However, the Flash controller doesn't send AHB error response for the following scenario:

1. Multiple AHB reads to flash are initiated on different AHB ports concurrently. This results in flash read from one of the AHB ports. The request is issued to flash but remains to wait for a response, while other AHB requests remain pending.

2. Meanwhile, a flash array integrity self-check is initiated by a core. This results in an incorrect AHB OK response for the first pending of the AHB requests and getting an incorrect read data from the embedded flash.

Workaround

An Array integrity self-check to the embedded flash memory must be initiated by code executed from SRAM. In a multicore environment, other cores' code should be moved to SRAM as well before initiating array integrity self-check.

ERR051134: When cores are in lockstep configuration and low power debug mode the Cortex-M7_1 is not halted while Cortex-M7_0 is in standby mode.

Description

The lockstep implementation of Cortex-M7_0 and Cortex-M7_1 is based on 2-cycle delayed redundancy. When using low power debug handshake the debugger configures the Cortex-M7_0 Debug Halting Control and Status Register (DHCSR) to enter into halt mode. The configuration of DHCSR register does not propagate to Cortex-M7_1 as delay elements are inactive in this phase. Thus Cortex-M7_1 continues to operate without DHCSR configured resulting in mismatch of Cortex-M7_0 and Cortex-M7_1 operation. It results in a core lockstep error.

Workaround

Ignore the lockstep error when in low power debug mode. This can be done by configuring the Device Configuration Module General-Purpose register's bits DCMRWD3[CM7_RCCU1_ALARM_EN]=1 and DCMRWD3[CM7_RCCU2_ALARM_EN]=1. After low power debug follow one of the following options to ensure clean operation mode of Cortex-M7 cores:

Option A: Configure DCMRWD3[CM7_RCCU1_ALARM_EN]=1 and DCMRWD3[CM7_RCCU2_ALARM_EN]=1 and configure the FCCU (Fault Collection and Control Unit) to trigger a destructive reset to the device in case of core lockstep error.

Option B: Trigger any destructive reset event (RESET_b pin, MDMAPCTL destructive reset, etc.).

ERR051222: GMAC/EMAC: Un-Correctable FSM Timeout Safety Interrupt incorrectly getting generated due to long waiting FSM states.

Description

If the DMA transmission is enabled by programming DMA_CH0_Tx_Control[ST] bit to '1', but the MAC transmission is disabled (MAC_Configuration[TE] bit set to 0), the FSM timeout is incorrectly generated after a duration configured via the MAC_FSM_ACT_Timer register field, leading to an FSM timeout interrupt.

Workaround

There are two workarounds, the software must implement either of the below:

1. Software must enable transmit DMA after enabling the MAC.
2. Software must enable both transmit DMA and MAC together.

ERR051421: SAI: Synchronous mode with bypass is not supported

Description

The SAI does not receive or transmit when:

Scenario 1. The transmitter is configured for synchronous mode (TCR2[SYNC] = 0b1), in the Transmit Configuration 2 register, and the receiver is in bypass (RCR2[BYP]=0b1), in the Receiver Configuration 2 register, then there will not be a bit clock as it is the source of the BCLK.

Scenario 2. The receiver is configured for synchronous mode (RCR2[SYNC] = 0b1) in the Receiver Configuration 2 register and the transmitter is in bypass (TCR2[BYP]=0b1), in the Transmit Configuration 2 register, then there will not be a bit clock as it is the source of the BCLK.

Workaround

If scenario 1, then set the TCR2[BCI] = 0b1, in the Transmit Configuration 2 register.

If scenario 2, then set the RCR2[BCI] = 0b1, in the Receiver Configuration 2 register.

ERR051588: LPSPI:Reset transmit FIFO after FIFO underrun by LPSPI Slave.

Description

Transmit FIFO pointers are corrupted when a transmit FIFO underrun occurs (SR[TEF]) in slave mode.

Workaround

When clearing the transmit error flag (SR[TEF] = 0b1) following a transmit FIFO underrun, reset the transmit FIFO (CR[RTF] = 0b1) before writing any new data to the transmit FIFO.

ERR051629: LPUART:Transmit Complete bit (STAT[TC]) is not set.

Description

When the CTS pin is negated and the CTS feature is enabled (MODIR[TXCTSE] = 0b1) and the TX FIFO is flushed by software then, the Transmit Complete (STAT[TC]) flag is not set.

Workaround

Clear (MODIR[TXCTSE]) bit and reset the transmit FIFO (FIFO[TXFLUSH] = 0b1) when flushing the FIFO with CTS enabled(MODIR[TXCTSE] = 0b1).

ERR052121: LPI2C: NACK Detect Flag can be set when IGNACK=1

Description

The NACK detect flag (MSR[NDF]) can be set even when the Controller Configuration 1 (MCFGR1[IGNACK]=0b1).

The LPI2C will not automatically generate a STOP or repeated START if the NACK detect flag (MSR[NDF]=0b1) and the ignore NACK are set (MCFGR1[IGNACK]=0b1). Thus, the transfer will continue as if the (MSR[NDF]) had not been set.

The LPI2C will continue to block a new START condition if (MSR[NDF]=0b1).

Workaround

When (MCFGR1[IGNACK]=0b1), the (MSR[NDF]) must be cleared by software, writing (MSR[NDF]=0b1) to allow new I2C transfers to start.

ERR052226: SWT: Toggling watchdog enable may cause unexpected timeout in some boundary conditions

Description

The Software Watchdog Timer (SWT) may timeout unexpectedly when loading a new timeout value. This can occur when the SWT is paused (CR[WEN]=0b0) to update the TO[WTO], while the counter is less than 0x14 (CO[CNT] = 0x14). When SWT is re-enabled (CR[WEN]=0b1), the SWT resumes the cycle count, but the counter is not updated (CO[CNT]) with the new timeout value before the cycle counter reaches zero.

Workaround

Before setting a new timeout value (TO[WTO]) the SWT must be updated with the watchdog keys ((SR[WSC]= 0xA602) and then (SR[WSC] = 0xB480)) to restart the counter value and have the timeout change being made within the appropriate time window, preventing the counter from reaching zero.

ERR052277: Cortex-M7: Can halt in an incorrect address when breakpoint and exception occurs simultaneously.

Description

Arm Errata 3092511

Affects: Cortex-M7, Cortex-M7 with FPU

Fault Type: Programmer Category C

When an asynchronous exception occurs at the same time as a breakpoint event (either hardware breakpoint or software breakpoint), it is possible for the processor to halt at the beginning of the exception handler instead of the instruction address pointed by the breakpoint.

Configurations Affected

This erratum affects all configurations of Cortex-M7.

When this happens:

- The BKPT bit in Debug Fault Status Register (DFSR) is set, indicating that a breakpoint event has occurred.
- The return address of the exception is the breakpoint address. As a result, if the debugger clears the halting control bit in the processor at this point, the processor will reach the breakpoint again after servicing the exception.

The correct behavior should be one of the followings:

1. Execute BKPT instruction and halt at BKPT before taking the asynchronous exception.

2. Take the asynchronous exception before BKPT and return to BKPT instruction and then halt on BKPT instruction.

In both cases, the debugger should see the processor halt on the BKPT instruction.

Workaround

This issue only affects the debugger's operation. The debugger could report the halting reason as an unknown breakpoint, and optionally resume operation. If the processor's operation is resumed, it is likely to be halted again immediately after the interrupt is serviced and returns to the breakpoint address.

ERR052403: FlexCAN: CAN frame drops in Enhanced RX FIFO when message buffer (MB) is locked for more than 1 CAN frame time (33 us)

Description

If Message Buffer (MB) and Enhanced RX FIFO both are configured for reception, and FlexCAN Message Buffer is locked for a long time (more than 1 CAN frame, 33us), FlexCAN receives some frames in FIFO and then start dropping the frames.

Workaround

There are two possible workarounds:

1) Core must read the message buffer within the 33 us after locking the MB.

2) Avoid using a few specific Message Buffers (MBs) as listed below:

ERR052438: FlexCAN: CAN frame may drop when using Enhanced RX FIFO

Description

An incoming CAN frame will be lost (i.e not latched into its expected Enhanced Rx FIFO data element), if both following two conditions are met simultaneously. There will be no indication that the frame was lost.

Conditions:

1. A write access is made to the message buffer Control and Status word (MB_CS) of a specific message buffer corresponding to the expected Enhanced Rx FIFO data element. Each Enhanced Rx FIFO data element corresponds to different message buffers impacted by this erratum and cannot be determined by software.

2. Depending on the timestamp configuration, the write access is made when receiving a frame at one certain Controller Host Interface (CHI) clock cycle either:

a. Around the time between the seventh bit of EOF and the second bit of IFS if timestamp is disabled (CTRL2[TSTAMPCAP] = 00b) or

b. Around the time between the fifth bit of EOF and seventh bit of EOF if timestamp is enabled (CTRL2[TSTAMPCAP] = 01b or 10b or 11b)

Workaround

To avoid the potential for dropped CAN frames, one of the following options may be implemented:

Workaround #1 : Disable Enhanced RX FIFO feature.

Workaround #2 : If the Enhanced RX FIFO feature is enabled, restrictions apply to certain Message Buffer (MB) numbers for both RX and TX. Either do not use these MB’s at all or at minimum, avoid updating the Control and Status word of these MBs when any reception to the Enhanced Rx FIFO could occur. This means, it would be safe to update the Control and Status word of these MBs when the FlexCAN is for example in Freeze mode or when it is ensured against frame reception from the CAN bus. Below are the MB numbers that have these restrictions:

ERR052460: Cortex-M7: A hang scenario can occur when a reserved read locked memory region is accessed by application cores

Description

Out of reset, by default, the Cortex-M7 core can perform speculative memory accesses to any region defined as Normal Device Type (Code, SRAM, RAM) and the memory attributes for a given address are defined by the settings of the device when the MPU is enabled.

A memory region in a reserved area is located between 0x1B10_0000 and 0x1B10_1FFF which lies within the Normal Code region (0x0000_0000 to 0x1FFF_FFFF) as defined by the default ARM memory map. This region is not intended to be accessed by users, but if accessed through a speculative or an explicit read, the region will not generate an error response and the core will enter into a hung state.

To reproduce the hang scenario, the user can perform an explicit read to this region. Additionally, the hang scenario can occur if the Cortex-M7 performs a speculative fetch to this region without user knowledge.

Workaround

To prevent a core hang scenario, read access to this region must be granted to the user by performing two back to back 32-bit writes to location 0x402A_C0F0. The data to be written firstly is 0x1CB0_499D followed by 0xB992_0D38. These writes should be performed as soon as possible out of reset so as to avoid a hang scenario due to a speculative read to this region. The data in this region is considered reserved and is subject to change.

ERR052558: FlexCAN: Message buffer (MB) overrun status is cleared when reading Enhanced RX FIFO (ERF)

Description

Message buffer status becomes "full" when a frame arrives, and status becomes "overrun" when a second message arrives in the same message buffer, if first message has still not been read. If frame reception is happening in ERF and the frame is being read from ERF, these reads could incorrectly clear the MB overrun status. As a result, the overrun event can be missed by the application.

Workaround

Use one of the following workarounds:

Workaround #1: Don’t use Enhanced RX FIFO (ERF).

Workaround #2: Don’t use any of the message buffers from MB0 to MB7 for reception if ERF is enabled. MB0 to MB7 can be used for transmission.

ERR052645: Flash: Incorrect read data may be returned from flash

Description

If flash prefetch is enabled when the Cortex-M7 caches are disabled, incorrect read data may be returned from the flash to the requesting Cortex-M7. The incorrect read may be returned in response to data or instruction fetches. This issue may be rarely encountered and requires a very specific data or instruction fetch sequence associated with a flash buffer miss followed by a scheduled prefetch and a new flash request to the page associated with the prefetch. When the issue does occur, incorrect data will be returned to the Cortex-M7 which will be from a valid flash location, but not from the address requested by the core and there is no direct indication that incorrect data has been returned.

Workaround

Avoid Cortex-M7 flash accesses when flash prefetching is enabled (PFCRn[Pn_mPFEN] = 1) and the Cortex-M7 instruction and data caches are disabled. Additionally, the code and data buffers can be controlled directly and independently through the PFCRn[Pn_mBFEN] bits. Therefore, if the data buffers or the code buffers have been enabled, the data or instruction caches must be enabled in all requesting cores accessing the flash.

Legal information

Definitions

Draft — A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Limiting values — Stress above one or more limiting values (as defined in the Absolute Maximum Ratings System of IEC 60134) will cause permanent damage to the device. Limiting values are stress ratings only and (proper) operation of the device at these or any other conditions above those given in the Recommended operating conditions section (if present) or the Characteristics sections of this document is not warranted. Constant or repeated exposure to limiting values will permanently and irreversibly affect the quality and reliability of the device.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at https://www.nxp.com/profile/terms, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

No offer to sell or license — Nothing in this document may be interpreted or construed as an offer to sell products that is open for acceptance or the grant, conveyance or implication of any license under any copyrights, patents or other industrial or intellectual property rights.

Quick reference data — The Quick reference data is an extract of the product data given in the Limiting values and Characteristics sections of this document, and as such is not complete, exhaustive or legally binding.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Suitability for use in non-automotive qualified products — Unless this document expressly states that this specific NXP Semiconductors product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. NXP Semiconductors accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without NXP Semiconductors' warranty of the product for such automotive applications, use and specifications, and (b) whenever customer uses the product for automotive applications beyond NXP Semiconductors' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies NXP Semiconductors for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond NXP Semiconductors' standard warranty and NXP Semiconductors' product specifications.

HTML publications — An HTML version, if available, of this document is provided as a courtesy. Definitive information is contained in the applicable document in PDF format. If there is a discrepancy between the HTML document and the PDF document, the PDF document has priority.

Translations — A non-English (translated) version of a document, including the legal information in that document, is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Security — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately.

Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP.

NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

NXP B.V. — NXP B.V. is not an operating company and it does not distribute or sell products.

Trademarks

Notice: All referenced brands, product names, service names, and trademarks are the property of their respective owners.

NXP — wordmark and logo are trademarks of NXP B.V.

MCXE31B_1P55A

Mask Set Errata

Contents

Please be aware that important notices concerning this document and the product(s) described herein, have been included in section 'Legal information'.

For more information, please visit: https://www.nxp.com

Document feedback

Date of release: 24 April 2025
Document identifier: MCXE31B_1P55A