MCXE245_0N57U

Mask Set Errata

Rev. 1.1 — 7 April 2025 Errata

1 Mask Set Errata for Mask 0N57U

1.1 Revision History

This report applies to mask 0N57U for these products:

- MCXE245VLF
- MCXE245VLH
- MCXE245VLL

Table 1. Revision History

Revision	Release Date	Significant Changes
1.1	4/2025	No changes to errata with this revision
1.0	1/2025	Initial Revision

1.2 Errata and Information Summary

Table 2. Errata and Information Summary

Erratum ID	Erratum Title
ERR006939	Core: Interrupted loads to SP can cause erroneous behavior
ERR006940	Core: VDIV or VSQRT instructions might not complete correctly when very short ISRs are used
ERR009004	Core: ITM can deadlock when global timestamping is enabled
ERR009005	Core: Store immediate overlapping exception return operation might vector to incorrect interrupt
ERR010527	LPUART: Setting and immediately clearing SBK bit can result in transmission of two break characters
ERR010655	LPSPI: Module Busy Flag may incorrectly read as zero in Master mode
ERR010658	ADC: Trigger loss is not reported
ERR010716	RTC: Timer Alarm Flag can assert erroneously
ERR010752	LPI2C: Slave Busy Flag may incorrectly read as zero when 10-bit address mode enabled in slave mode.
ERR010777	SCG: Corrupted status when the system clock is switching.
ERR010792	LPI2C: Slave Transmit Data Flag may incorrectly read as one when TXCFG is zero.
ERR010856	FTM: Safe state is not removed from channel outputs after fault condition ends if SWOCTRL is being used to control the pin
ERR011063	SMC: An asynchronous wakeup event during VLPS mode entry may result in possible system hang scenario.
ERR011089	LPSPI: In Continuous transfer mode with CPHA =1, WCF bit is not set for every word.

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

Erratum ID	Erratum Title
ERR011097	LPSPI: Command word not loaded correctly when TXMSK=1
ERR011114	SMC: invalid data might be fetched while accessing Flash in VLP modes
ERR011543	FlexCAN: Nominal Phase SJW incorrectly applied at CRC Delimiter
ERR050443	FlexCAN: Receive Message Buffers may have its CODE Field corrupted if the Receive FIFO function is used in Classical CAN mode (CAN 2.0 version B)
ERR050606	LPSPI: TCR value does not get resampled when polling the register
ERR050607	LPSPI: TCR[FRAMSZ] can be ignored when TCR[TXMSK]=1b1
ERR050877	Core: Fused MAC instructions give incorrect results for rare data combinations
ERR050878	Core: Processor reset asserted asynchronously could corrupt FPB comparator registers and remap to wrong address
ERR051472	LPSPI: Disabling and enabling LPSPI would cause SR[REF] to assert
ERR051629	LPUART:Transmit Complete bit (STAT[TC]) is not set.
ERR052094	PMC: During the power up sequence, the microcontroller may stay in reset condition if the current demand is not satisfied.
ERR052121	LPI2C: NACK Detect Flag can be set when IGNACK=1

2 Known Errata

ERR006939: Core: Interrupted loads to SP can cause erroneous behavior

Description

Arm Errata 752770: Interrupted loads to SP can cause erroneous behavior

This issue is more prevalent for user code written to manipulate the stack. Most compilers will not be affected by this, but please confirm this with your compiler vendor. MQX™ and FreeRTOS™ are not affected by this issue.

Affects: Cortex-M4, Cortex-M4F

Fault Type: Programmer Category B

Fault Status: Present in: r0p0, r0p1 Open.

If an interrupt occurs during the data-phase of a single word load to the stack-pointer (SP/R13), erroneous behavior can occur. In all cases, returning from the interrupt will result in the load instruction being executed an additional time. For all instructions performing an update to the base register, the base register will be erroneously updated on each execution, resulting in the stack-pointer being loaded from an incorrect memory location.

The affected instructions that can result in the load transaction being repeated are:

1) LDR SP,[Rn],#imm

2) LDR SP,[Rn,#imm]!

3) LDR SP,[Rn,#imm]

4) LDR SP,[Rn]

5) LDR SP,[Rn,Rm]

The affected instructions that can result in the stack-pointer being loaded from an incorrect memory address are:

1) LDR SP,[Rn],#imm

2) LDR SP,[Rn,#imm]!

Conditions:

1) An LDR is executed, with SP/R13 as the destination.

2) The address for the LDR is successfully issued to the memory system.

3) An interrupt is taken before the data has been returned and written to the stack-pointer.

Implications:

Unless the load is being performed to Device or Strongly-Ordered memory, there should be no implications from the repetition of the load. In the unlikely event that the load is being performed to Device or Strongly-Ordered memory, the repeated read can result in the final stack-pointer value being different than had only a single load been performed.

Interruption of the two write-back forms of the instruction can result in both the base register value and final stack-pointer value being incorrect. This can result in apparent stack corruption and subsequent unintended modification of memory.

Workaround

Most compilers are not affected by this, so a workaround is not required.

However, for hand-written assembly code to manipulate the stack, both issues may be worked around by replacing the direct load to the stack-pointer, with an intermediate load to a general-purpose register followed by a move to the stack-pointer.

If repeated reads are acceptable, then the base-update issue may be worked around by performing the stack pointer load without the base increment followed by a subsequent ADD or SUB instruction to perform the appropriate update to the base register.

ERR006940: Core: VDIV or VSQRT instructions might not complete correctly when very short ISRs are used

Description

Arm Errata 776924: VDIV or VSQRT instructions might not complete correctly when very short ISRs are used

Affects: Cortex-M4F

Fault Type: Programmer Category B

Fault Status: Present in: r0p0, r0p1 Open.

On Cortex-M4 with FPU, the VDIV and VSQRT instructions take 14 cycles to execute. When an interrupt is taken a VDIV or VSQRT instruction is not terminated, and completes its execution while the interrupt stacking occurs. If lazy context save of floating point state is enabled then the automatic stacking of the floating point context does not occur until a floating point instruction is executed inside the interrupt service routine.

Lazy context save is enabled by default. When it is enabled, the minimum time for the first instruction in the interrupt service routine to start executing is 12 cycles. In certain timing conditions, and if there is only one or two instructions inside the interrupt service routine, then the VDIV or VSQRT instruction might not write its result to the register bank or to the FPSCR.

Workaround

A workaround is only required if the floating point unit is present and enabled. A workaround is not required if the memory system inserts one or more wait states to every stack transaction.

There are two workarounds:

1) Disable lazy context save of floating point state by clearing LSPEN to 0 (bit 30 of the FPCCR at address 0xE000EF34).

2) Ensure that every interrupt service routine contains more than 2 instructions in addition to the exception return instruction.

ERR009004: Core: ITM can deadlock when global timestamping is enabled

Description

ARM ERRATA 806422

The Cortex-M4 processor contains an optional Instrumentation Trace Macrocell (ITM). This can be used to generate trace data under software control, and is also used with the Data Watchpoint and Trace (DWT) module which generates event driven trace. The processor supports global timestamping. This allows count values from a system-wide counter to be included in the trace stream.

When connected directly to a CoreSight funnel (or other component which holds ATREADY low in the idle state), the ITM will stop presenting trace data to the ATB bus after generating a timestamp packet. In this condition, the ITM_TCR.BUSY register will indicate BUSY.

Once this condition occurs, a reset of the Cortex-M4 is necessary before new trace data can be generated by the ITM.

Timestamp packets which require a 5 byte GTS1 packet, or a GTS2 packet do not trigger this erratum. This generally only applies to the first timestamp which is generated.

Devices which use the Cortex-M optimized TPIU (CoreSight ID register values 0x923 and 0x9A1) are not affected by this erratum.

Workaround

There is no software workaround for this erratum. If the device being used is susceptible to this erratum, you must not enable global timestamping.

ERR009005: Core: Store immediate overlapping exception return operation might vector to incorrect interrupt

Description

Arm Errata 838869: Store immediate overlapping exception return operation might vector to incorrect interrupt

Affects: Cortex-M4, Cortex-M4F

Fault Type: Programmer Category B Rare

Fault Status: Present in: r0p0, r0p1 Open.

The Cortex-M4 includes a write buffer that permits execution to continue while a store is waiting on the bus. Under specific timing conditions, during an exception return while this buffer is still in use by a store instruction, a late change in selection of the next interrupt to be taken might result in there being a mismatch between the interrupt acknowledged by the interrupt controller and the vector fetched by the processor.

Configurations Affected

This erratum only affects systems where writeable memory locations can exhibit more than one wait state.

Workaround

For software not using the memory protection unit, this erratum can be worked around by setting DISDEFWBUF in the Auxiliary Control Register.

In all other cases, the erratum can be avoided by ensuring a DSB occurs between the store and the BX instruction. For exception handlers written in C, this can be achieved by inserting the appropriate set of intrinsics or inline assembly just before the end of the interrupt function, for example:

ARMCC:

...

schedule_barrier();

asm{DSB};

schedule_barrier();

}

GCC:

...

asm volatile ("dsb 0xf" ::: "memory"); volatile ("dsb 0xf" ::: "memory");

}

ERR010527: LPUART: Setting and immediately clearing SBK bit can result in transmission of two break characters

Description

When the LPUART transmitter is idle (LPUART_STAT[TC]=1), two break characters may be sent when using LPUART_CTRL[SBK] to send one break character. Even when LUART_CTRL[SBK] is set to 1 and cleared (set to 0) immediately.

Workaround

To queue a single break character via the transmit FIFO, set LPUART_DATA[FRETSC]=1 with data bits LPUART_DATA[T9:T0]=0.

ERR010655: LPSPI: Module Busy Flag may incorrectly read as zero in Master mode

Description

When operating in master mode, the Module Busy Flag, SR[MBF], may incorrectly read as zero. This applies only to master mode and not to slave mode.

Workaround

When using the LPSPI in master mode and the SR[MBF] bit is read as a one then the flag is set. If it is read as a zero, it must be read a second time and this second read will be the correct state of the bit.

ERR010658: ADC: Trigger loss is not reported

Description

A trigger loss will not be reported in the following cases:

1. The time between two consecutive triggers on a trigger input is less than 4 cycles of ADC's Module interface clock.

2. One or multiple triggers come on any input while a trigger either stored or under process for that input in the trigger handler.

3. The trigger on a particular input asserts during the period of end-of-conversion cycle of a conversion initiated from this input.

- In these cases the Error in Multiplexed Trigger Request (ADC_SC2[TRGSTERR]) bit in the ADC Status and Control Register 2 will not be set.

- For the third case the Trigger status (ADC_SC2[TRGSTLAT]) bit in the ADC Status and Control Register 2 will also not be set.

Workaround

The time between two consecutive triggers on a particular input should be kept more than one ADC conversion time.

ERR010716: RTC: Timer Alarm Flag can assert erroneously

Description

Writing to the Time Alarm Register (RTC_TAR) at the same time the RTC Seconds Register is incrementing can assert the Time Alarm Flag (RTC_SR[TAF]) bit in the RTC Status Register.

Workaround

Write the Time Alarm Register (RTC_TAR) when the RTC Seconds Register is not incrementing. This can be when Time Counter Enable (RTC_SR[TCE]) bit in the RTC Status Register is clear or within the RTC_SR[TAF] interrupt routine.

Alternatively, if the RTC_SR[TAF] is asserted following a write to the RTC_TAR, then write the RTC_TAR again.

ERR010752: LPI2C: Slave Busy Flag may incorrectly read as zero when 10-bit address mode enabled in slave mode.

Description

When operating in slave mode with 10-bit addressing enabled and an address match is detected on the first address byte, the Slave Busy Flag, SSR[SBF], may incorrectly read as zero.

Workaround

When using the LPI2C in slave mode when 10-bit addressing is enabled, and the SSR[SBF] bit is read as a one, then the flag is set. If it is read as a zero, it must be read a second time and this second read will be the correct state of the bit.

ERR010777: SCG: Corrupted status when the system clock is switching.

Description

The SCG_RCCR[SCS] and SCG_HCCR[SCS] may have a corrupted status during the interval when the system clock is switching

Workaround

The SCS field should be read twice by the software to ensure the system clock switch has completed.

ERR010792: LPI2C: Slave Transmit Data Flag may incorrectly read as one when TXCFG is zero.

Description

When SCFGR1[TXCFG] = 0, the slave transmit data ready flag can incorrectly assert for one cycle.

Workaround

Set SCFGR1[TXCFG] = 1.

ERR010856: FTM: Safe state is not removed from channel outputs after fault condition ends if SWOCTRL is being used to control the pin

Description

If an FTM channel output is being controlled using the software output control register (FTM_SWOCTRL) and fault detection is also enabled for the channel, then when a fault is detected the output is forced to its safe value. However, when the fault condition has been cleared, the channel output will stay in the safe state instead of reverting to the value programmed by the FTM_SWOCTRL register.

Workaround

If fault control is enabled while the software output control register is also being used (FTM_SWOCTRL), then the FTM should be configured as follows:

-- FTM_MODE[FAULTM] configured for manual fault clearing (0b10)

-- For devices that include the FTM_CONF[NUMTOF] field, it must be cleared to 0b00000 (TOF set for each counter overflow). For FTM versions that don't include the FTM_CONF[NUMTOF] field this doesn't apply.

The procedure below must be used in the TOF interrupt handler when a fault is detected to ensure that the outputs return to the value configured by FTM_SWOCTRL.

1. Check the value of FTM_FMS[FAULTF].

-- If FTM_FMS[FAULTF] = 1 (fault occurred or is occurring), then set a variable to indicate that a fault was detected and continue to step 2.

-- If FTM_FMS[FAULTF] = 0 but the fault variable is set (fault is not active, but was previously detected), continue to step 6.

2. Write the FTM_OUTMASK register to set the bit or bits corresponding to any channels that are controlled by FTM_SWOCTRL to temporarily inactivate the channel output.

3. Clear fault conditions by reading the FTM_FMS register and then writing FTM_FMS with all zeroes.

4. Clear the FTM_SC[TOF] bit by reading the FTM_SC register, then writing a 0 to FTM_SC[TOF].

5. Exit the interrupt handler to skip following steps (they will execute the next time the TOF handler is called).

6. Clear the FTM_SWOCTRL by writing all zeroes to it.

7. Write FTM_SWOCTRL with the desired value again.

8. Clear the FTM_OUTMASK bits that were set in step 2.

9. Clear the fault variable that was set in step 1 when the fault condition was originally detected.

10. Clear the FTM_SC[TOF] bit by reading the FTM_SC register, then writing a 0 to FTM_SC[TOF].

ERR011063: SMC: An asynchronous wakeup event during VLPS mode entry may result in possible system hang scenario.

Description

When the bus clock is same system clock and an asynchronous wakeup occurs during a mode transition from RUN to VLPS or VLPR to VLPS, the MCU may hang in an undetermined state, which can only be recovered by a power-on reset event or a watchdog reset.

Workaround

Before executing the transition to VLPS ensure that the PREDIV_SYS_CLK frequency / BUS_CLK frequency configuration for RUN/VLPR mode is greater than or equal to 2.

For example: Assuming a PREDIV_SYS_CLK of 8 MHz and SCG_RCCR[DIVCORE] = 0b0001 (divider of 2) and SCG_RCCR[DIVBUS] = 0b0000 (divider of 1), (PREDIV_SYS_CLK = 8 MHz) / (BUS_CLK = 4 MHz) , a ratio of 1:2.

ERR011089: LPSPI: In Continuous transfer mode with CPHA =1, WCF bit is not set for every word.

Description

When Transmit Command Register is written with TCR[CONT]=1 and TCR[CPHA]=1, SR[WCF] bit flag is not set after data is transferred. Therefore polling for SR[WCF] flag to identify if data has been sent can cause MCU to be stuck.

Workaround

When using continuous transfer mode TCR[CONT]=1 and TCR[CPHA]=1, do not use SR[WCF] flag to determine if data has been sent, fill up instead transmit FIFO with the following data without waiting for SR[WCF] flag to be set.

ERR011097: LPSPI: Command word not loaded correctly when TXMSK=1

Description

When the Transmit Command Register is written with TCR[TXMSK]=1 and the next write to the TX FIFO is another command, then the first command may not load correctly.

Workaround

When writing the Transmit Command Register with TCR[TXMSK]=1, wait for the TX FIFO to go empty (FSR[TXCOUNT] = 0) before writing another command to the Transmit Command Register.

ERR011114: SMC: invalid data might be fetched while accessing Flash in VLP modes

Description

VLPR and VLPS Low power modes are documented to work at System Clock and Core Clock at 4 Mhz and the Bus Clock at 4 MHz and DMA enabled from or to Flash memory. However any simultaneous access from any

master (Core or DMA) to Dflash and Pflash may get invalid data while being in VLP modes and System clock, Core Clock and Bus Clock are above 1 Mhz

Workaround

There are two workarounds:

1. Restrict software to use either only Pflash or only Dflash only at a time in VLP modes for all masters (CPU,DMA) . When switching from Pflash only access to Dflash only access let current DMA transactions accessing flash to complete and jump to SRAM location , wait for 40 cycles for the ongoing accesses to complete on the current flash before accessing dflash.

When switching from dflash only accesses to pflash only accesses let the current DMA transactions accessing dflash to complete

and wait for 40 cycles for accesses to complete on the dflash before accessing the pflash.

2. If both Pflash and Dflash needs to be accessed simultaneously, the VLP modes must be run with System Clock, Core Clock and Bus Clock of 1 MHz.

ERR011543: FlexCAN: Nominal Phase SJW incorrectly applied at CRC Delimiter

Description

During the reception of a CAN-FD frame when the Bit Rate Switch (BRS) is enabled, the Synchronization Jump Width (SJW) for the CRC Delimiter bit is incorrectly defined by the Nominal Phase SJW. The CAN specification stipulates that the CRC Delimiter bit should have a SJW set by the Data Phase SJW.

When a resynchronization event is triggered for the CRC delimiter bit (recessive in correct operation), the sample point will be adjusted by an amount as defined by the Nominal Phase SJW rather than the specified Data Phase SJW. This may result in the incorrect detection of a dominant bit leading to a CAN error frame. However, as the CRC delimiter bit position will only apply the SJW upon the detection of an unexpected dominant bit on the CAN bus, an error frame is already likely. For the case the SJW is applied at the CRC delimiter and a recessive bit is not detected, the receiving node will issue an error frame.

The CAN protocol is designed to handle resynchronization errors and hence the CAN bus will recover from the insertion of the incorrect SJW at the CRC delimiter. Upon detecting the error frame the transmitting node will re-transmit the frame.

The following FlexCAN configurations are not affected:

- Classical CAN frames (CAN 2.0B)
- CAN FD frames with bit rate switch disabled (BRS = 0)
- CAN FD frames with Nominal Phase SJW equal to Data Phase SJW
- CAN FD transmissions

Configuration for the FlexCAN:

- Nominal Phase SJW is configured by the Resync Jump Width bit in the CAN Control Register 1 (CAN_CTRL1[RJW]) or by the Extended Resync Jump Width bit in the CAN Bit Timing Register (CAN_CBT[ERJW])
- Data Phase SJW is configured by the Fast Resync Jump Width bit in the CAN FD Bit Timing Register (CAN_FDCBT[FRJW])

Workaround

The robustness of the CAN protocol ensures that the receiver automatically recovers from the application of the incorrect SJW. The CAN protocol is designed to recover from resynchronization errors and hence any frame that is not correctly received will be re-sent by the transmitting node.

ERR050443: FlexCAN: Receive Message Buffers may have its CODE Field corrupted if the Receive FIFO function is used in Classical CAN mode (CAN 2.0 version B)

Description

If the CODE Field of a Receive Message Buffer is corrupted it may deactivate the Message Buffer, so it is unable to receive new messages. It may also turn a Receive Message Buffer into any type of Message Buffer as defined in the Message buffer structure section in the device documentation.

The CODE Field of the FlexCAN Receive Message Buffers (MB) may get corrupted if the following sequence occurs.

1- A message is received and transferred to an MB (i.e. MBx)

2- A new message start being received (i.e. message1), SMB0 (Serial Message Buffer 0) receives the message1 intended for MBx

3- Before SMB0 being moved to MBx, MBx is locked by software for more than 20 CAN bit times (time determines the probability of erratum to manifest), therefore SMB0 is NOT transferred to MBx, it remains with message1.

4- A subsequent incoming message (i.e. message2) is being loaded into SMB1 (as SMB0 is full) and is evaluated by the FlexCAN hardware as being for the FIFO.

5- During the message2, the MBx is unlocked. Then, the content of SMB0 is transferred to MBx and the CODE field is updated with an incorrect value.

In case a customer does use Rx FIFO only or dedicated MB only, (i.e. either Rx MB or Rx FIFO is used) the problem doesn't occur. In case a customer does use Flexible Data Rate CAN (CAN FD), the problem does not occur also. So bottom line the problem does only occur if the FlexCAN is programmed to receive in the FIFO and dedicated MB at the same application.

Workaround

This defect only applies if the Receive FIFO is used. This feature is enabled by RFEN bit in the Module Control Register (MCR[RFEN]). If the Rx FIFO is not used, the Receive Message Buffer CODE Field is not corrupted.

The defect does not occur if the Receive Message Buffer lock time is less than or equal to the time equivalent to 20 x CAN bit time.

After receiving the Interrupt Flag for the corresponding MB (BUFx bit on the IFLAGx register) set by the hardware, the recommended way for the CPU to service (read) the frame received in a mailbox is by the following procedure:

1. Read the Control and Status word of that mailbox.

2. Check if the BUSY bit (CODE[0]) is deasserted, indicating that the mailbox is not locked. Repeat step 1) while it is asserted.

3. Read the contents of the mailbox.

4. Clear the proper flag in the IFLAG register.

5. Read the Free Running Timer register (TIMER) to unlock the mailbox

In order to guarantee that this procedure occurs in less than 20 CAN bit times the MB receive handling process in software (step 1 to step 5 above) should be performed as a 'critical code section' (interrupts disabled before execution) and should ensure that the MB receive handling occurs in a deterministic number of cycles.

If the MB receive handling process can't be guaranteed in a time, less than or equal to 20 CAN bit times, Rx FIFO should not be used together with the receive Message Buffers in a Classical CAN application.

ERR050606: LPSPI: TCR value does not get resampled when polling the register

Description

Reading the Transmit Command Register will return the current state of the command register.

Following a write to the TCR (Transmit Command register), if the user continuously reads the TCR (polls the register), then the read content no longer represents the contents of the Transmit Command register if it updates due to internal logic following the first read. The same value shall continue to be read.

Workaround

After reading the Transmit Command Register must always access a different register in between subsequent reads from TCR.

ERR050607: LPSPI: TCR[FRAMSZ] can be ignored when TCR[TXMSK]=1b1

Description

TCR (Transmit Command Register) is used to write new command word to the LPSPI transmit FIFO.

TCR[FRAMESZ] configures the frame size of the data to be transmitted in number of bits equal to (FRAMESZ + 1). When TCR[TXMSK] is set, transmit data is masked (no data is loaded from transmit FIFO and output pin is tristated). In master mode, the Transmit Data Mask bit will initiate a new transfer which cannot be aborted by another command word; the Transmit Data Mask bit will be cleared by hardware at the end of the transfer. TCR[CONTC] controls the continuous transfer mode. TCR[CONTC]=1b1 enables continuous transfer. In master mode, continuous transfer will keep the PCS asserted at the end of the frame size, until a command word is received that starts a new frame.

If command word is written with TCR[TXMSK]=1 and TCR[FRAMESZ]>32 and the next command word with TCR[CONTC]=0 is in the FIFO then at the end of any 32-bit word of the first command, the frame will terminate early and negate PCS.

Workaround

There are two workarounds:

1. Do not write a 2nd command word after writing command word with TXMSK=1 and FRAMESZ>32 until the first one has completed.

OR

2. Divide the command word into multiple command words with TXMSK=1 and FRAMESZ=32 (or remainder) using a continuous transfer.

ERR050877: Core: Fused MAC instructions give incorrect results for rare data combinations

Description

Arm errata 839676

Affects: Cortex-M4F

Fault Type: Programmer Category B Rare

Fault Status: Present in: r0p0, r0p1. Open.

The Cortex-M4 processor includes optional floating-point logic which supports the fused MAC instructions (VFNMA, VFNMS, VFMA, VFMS). This erratum causes fused MAC operations on certain combinations of operands to result in either or both of the following:

• A result being generated which is one Unit of Least Precision (ULP) greater than the correct result.

• Inexact or underflow flags written to the Floating-point Status Control Register, FPSCR, incorrectly.

Configurations Affected:

Cortex-M4F configured with floating-point.

Conditions:

The conditions for this erratum are all of the following:

1. Cortex-M4 is configured with floating-point and it is enabled in the Coprocessor Access Control Register, CPACR.

2. Flush-to-Zero (FZ) mode is not enabled, that is, FPSCR.FZ is clear.

3. A fused MAC instruction (VFNMA, VFNMS, VFMA, or VFMS) is executed with all of the following properties:

• The addition part of the operation is Unlike Signed Addition (USA). This implies that the combination of the instruction used and the signs of the operands means that a subtraction is being performed.

• The result of the instruction, before rounding, is subnormal. This implies that the result is smaller in magnitude than 2-126.

• The significance of the product and the addend operand are the same or differ by one.

Implications:

If this erratum occurs, then the processor either produces an incorrect result to a computation or fails to run a floating-point exception routine when it should.

Note:

It is expected that most algorithms only use normalised numbers because:

• Subnormal results have low precision.

• It is easier to avoid underflow.

This erratum does not affect algorithms which only use normalized numbers.

Workaround

Enable FZ mode in the FPSCR.

ERR050878: Core: Processor reset asserted asynchronously could corrupt FPB comparator registers and remap to wrong address

Description

Arm errata 1299509

Affects: Cortex-M4, Cortex-M4F

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r0p1. Open.

Normally, the debugger uses the Flash Patch and Breakpoint (FPB) unit for breakpoints or for patching ROM code during debugging. However, you can also use the FPB functionality as a method of in-the-field ROM code patching. On the Cortex-M4 processor, the processor reset can be asynchronously asserted and this could potentially cause problems if the processor is in the middle of writing to debug components (which are not reset by the processor reset) such as the FPB unit.

Configurations Affected:

A Cortex-M4 processor that is configured with debug. Conditions

1) The processor is programming the FPB to remap an address in ROM to fetch a replacement opcode or vector from an area in RAM.

2) The processor is asynchronously reset during the programming of the FPB.

3) The data is incorrectly written to the FPB register due to metastability.

Implications:

In the unlikely event of this occurring it may cause incorrect functional operation of the processor to occur. If the FPB is programmed with incorrect data it may cause the processor to unintentionally patch instructions.

Workaround

The problem does not arise if the FPB is not used to patch instructions. If the FPB is used to patch instructions, a software workaround is to ensure that the FPB is globally disabled before programming any comparators. If code exists which programs the FPB other than at reset, the software workaround should include: 1. Disable the FPB. 2. Program the individual comparators required. 3. Explicitly disable the individual comparators not required. 4. Re-enable the FPB. This sequence prevents any corruptly programmed FPB comparators from being activated.

ERR051472: LPSPI: Disabling and enabling LPSPI would cause SR[REF] to assert

Description

The SR[REF] asserts if software disables the LPSPI module after receiving some data. It then enables the LPSPI again without performing a software reset.

Workaround

When software disables the LPSPI and enables it again, either:

a) Clear SR[REF] flag before starting any data transfer or

b) Software reset the LPSPI before enabling

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).

ERR052094: PMC: During the power up sequence, the microcontroller may stay in reset condition if the current demand is not satisfied.

Description

A momentary high current may be seen during power up and if the external voltage regulator is not capable of satisfying the demand, the internal supply voltage ramp may stall resulting in a sustained high current state holding the device in reset.

The high current will not cause permanent damage and will not affect lifetime performance of the device.

Workaround

Use an external regulator capable of supplying the specified voltage with a minimum current of 250mA. If not, it is recommended to integrate an external watchdog circuit capable of detecting a power ramp stall and automatically recycle the power to the microcontroller or a similar mechanism that complies with the above values and recommendations.

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.

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.

Contents