

Mask Set Errata for Mask 0N57U

This report applies to mask 0N57U for these products:

- WCT1014SF

Table 1. Errata and Information Summary

Erratum ID	Erratum Title
e10658	ADC: Trigger loss is not reported
e6939	Core: Interrupted loads to SP can cause erroneous behavior
e9004	Core: ITM can deadlock when global timestamping is enabled
e9005	Core: Store immediate overlapping exception return operation might vector to incorrect interrupt
e6940	Core: VDIV or VSQRT instructions might not complete correctly when very short ISRs are used
e10856	FTM: Safe state is not removed from channel outputs after fault condition ends if SWOCTRL is being used to control the pin
e10752	LPI2C: Slave Busy Flag may incorrectly read as zero when 10-bit address mode enabled in slave mode.
e10792	LPI2C: Slave Transmit Data Flag may incorrectly read as one when TXCFG is zero.
e10655	LPSPi: Module Busy Flag may incorrectly read as zero in Master mode
e10527	LPUART: Setting and immediately clearing SBK bit can result in transmission of two break characters
e10716	RTC: Timer Alarm Flag can assert erroneously
e10777	SCG: Corrupted status when the system clock is switching.

Table 2. Revision History

Revision	Changes
17/feb/2017	Initial revision
31/May/2017	The following errata were added. <ul style="list-style-type: none">• e10856



e10658: 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.

e6939: 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.

e9004: 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.

e9005: 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");  
}
```

e6940: Core: VDIV or VSQRT instructions might not complete correctly when very short ISRs are used

Description: ARM Errata 709718: 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.

e10856: 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].

e10752: 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.

e10792: 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.

e10655: 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.

e10527: 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 LPUART_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.

e10716: 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.

e10777: 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.



How to Reach Us:

Home Page:

nxp.com

Web Support:

nxp.com/support

Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including "typicals," must be validated for each customer application by customer's technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: nxp.com/SalesTermsandConditions.

NXP, the NXP logo, NXP SECURE CONNECTIONS FOR A SMARTER WORLD, Freescale, the Freescale logo, SafeAssure and the SafeAssure logo are trademarks of NXP B.V. All other product or service names are the property of their respective owners. ARM, the ARM powered logo, Thumb, and Cortex are registered trademarks of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. All rights reserved.

© 2017 NXP B.V.

