

Mask Set Errata for Mask 0N73V

This report applies to mask 0N73V for these products:

- S32K146

Table 1. Errata and Information Summary

Erratum ID	Erratum Title
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
e11097	LPSPi: Command word not loaded correctly when TXMSK=1
e11089	LPSPi: In Continuous transfer mode with CPHA =1, WCF bit is not set for every word.
e10777	SCG: Corrupted status when the system clock is switching.
e11063	SMC: An asynchronous wakeup event during VLPS mode entry may result in possible system hang scenario.
e11114	SMC: invalid data might be fetched while accessing Flash in VLP modes

Table 2. Revision History

Revision	Changes
24 July 2017	Initial revision
30/Nov/2017	The following errata were added. <ul style="list-style-type: none">• e11114• e11097• e11063• e11089 The following errata were revised. <ul style="list-style-type: none">• e6939



Table 2. Revision History

Revision	Changes
	<ul style="list-style-type: none">• e9005• e6940

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.

e11097: 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.

e11089: 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.

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.

e11063: 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] = 0b0001 (divider of 1), (PREDIV_SYS_CLK = 8 MHz) / (BUS_CLK = 4 MHz) , a ratio of 1:2.

e11114: 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.

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, COOLFLUX, EMBRACE, GREENCHIP, HITAG, I2C BUS, ICODE, JCOP, LIFE VIBES, MIFARE, MIFARE CLASSIC, MIFARE DESFire, MIFARE PLUS, MIFARE FLEX, MANTIS, MIFARE ULTRALIGHT, MIFARE4MOBILE, MIGLO, NTAG, ROADLINK, SMARTLX, SMARTMX, STARPLUG, TOPFET, TRENCHMOS, UCODE, Freescale, the Freescale logo, Altivec, C-5, CodeTest, CodeWarrior, ColdFire, ColdFire+, C-Ware, the Energy Efficient Solutions logo, Kinetis, Layerscape, MagniV, mobileGT, PEG, PowerQUICC, Processor Expert, QorIQ, QorIQ Qonverge, Ready Play, SafeAssure, the SafeAssure logo, StarCore, Symphony, VortiQa, Vybrid, Airfast, BeeKit, BeeStack, CoreNet, Flexis, MXC, Platform in a Package, QUICC Engine, SMARTMOS, Tower, TurboLink, and UMEMS are trademarks of NXP B.V. All other product or service names are the property of their respective owners. ARM, AMBA, ARM Powered, Artisan, Cortex, Jazelle, Keil, SecurCore, Thumb, TrustZone, and μ Vision are registered trademarks of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. ARM7, ARM9, ARM11, big.LITTLE, CoreLink, CoreSight, DesignStart, Mali, mbed, NEON, POP, Sensinode, Socrates, ULINK and Versatile are trademarks of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. The Power Architecture and Power.org word marks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org.

© 2018 NXP B.V.

