

Mask Set Errata for Mask 0N79P

This report applies to mask 0N79P for these products:

- MKE1xF512VLL16
- MKE1xF512VLH16
- MKE1xF256VLL16
- MKE1xF256VLH16

Table 1. Errata and Information Summary

Erratum ID	Erratum Title
e10575	BootROM: Two Bootloader GetProperty commands of RAM size and start address return value are not correct
e6939	Core: Interrupted loads to SP can cause erroneous behavior
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
e9004	ITM can deadlock when global timestamping is enabled
e10364	LPI2C: LPI2C sends a STOP condition if transmit FIFO is empty on the completion of a master-receive transfer
e10527	LPUART: Setting and immediately clearing SBK bit can result in transmission of two break characters
e10180	SCG: Clock switch may hang if SCG_RCCR is written to the switch system clock source with a different divide ratio while an external reset is asserted
e10536	WDOG: After getting RCS assertion by polling, 4 LPO clock-time delay is the minimum requirement before the next block

Table 2. Revision History

Revision	Changes
26 May 2016	Initial revision
15 Aug 2016	The following errata were added.

Table continues on the next page...



Table 2. Revision History (continued)

Revision	Changes
	<ul style="list-style-type: none">• e10527• e10536
29 Aug 2016	The following errata were added. <ul style="list-style-type: none">• e6940• e9005• e9004• e6939• e10180
2 Sep 2016	The following errata was added. <ul style="list-style-type: none">• e10575 The following errata was revised. <ul style="list-style-type: none">• e10364

e10575: BootROM: Two Bootloader GetProperty commands of RAM size and start address return value are not correct

Description: On MKE1xF256VLL16 and MKE1xF256VLH16 parts, two Bootloader GetProperty commands of RAM size and start address return value are not correct. The RAM start address return value mentioned is 0x1FFFE000. However, the actual RAM start address is 0x1FFFC000. The RAM size return value mentioned is 16 KB. However, the actual RAM size is 32 KB. Thus, you can only access 16 KB RAM by the BootROM.

Workaround: No workaround. Only the RAM memory access by BootROM has this limitation. Other RAM access is normal.

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.

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.

e9004: ITM can deadlock when global timestamping is enabled

Description: 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.

e10364: LPI2C: LPI2C sends a STOP condition if transmit FIFO is empty on the completion of a master-receive transfer

Description: If the transmit FIFO is empty at the end of a master receive transfer and the AUTOSTOP (MCFGR1[AUTOSTOP]) bit in the Master Configuration Register 1 is clear, the LPI2C master sends a STOP condition before the next repeated START condition.

Workaround: Use software or DMA to queue up the subsequent transfer in the transmit FIFO before the completion of the master-receive transfer.

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.

e10180: SCG: Clock switch may hang if SCG_RCCR is written to the switch system clock source with a different divide ratio while an external reset is asserted

Description: SCG: Clock switch may hang if SCG_RCCR is written to the switch system clock source with a different divide ratio while an external reset is asserted. For example, if SCG is in the FIRC mode and then configured to the SIRC mode, by writing the RCCR to switch system clock with a different divide ratio, during which the external reset is asserting, then the clock switch may hang.

Workaround: To recover the clock switch another reset must be issued.

e10536: WDOG: After getting RCS assertion by polling, 4 LPO clock-time delay is the minimum requirement before the next block

Description: WDOG cannot be unlocked if the unlock magic word are executed immediately after the RCS assert.

Workaround: After getting RCS assertion by polling, 4 LPO clock-time delay is the minimum requirement before next block.

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, and Kinetis are trademarks of NXP B.V. All other product or service names are the property of their respective owners. ARM, the ARM Powered logo, and Cortex are registered trademarks of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. mbed is a trademark of ARM Limited (or its subsidiaries) in the EU and/or elsewhere.

All rights reserved.

© 2016 NXP B.V.

Document Number: KINETIS_E_0N79P

Rev. 02SEP2016

