NXP Semiconductors
Mask Set Errata

Kinetis_V_1N86P
Rev. 14 JUN 2016

Mask Set Errata for Mask 1N86P

This report applies to mask 1N86P for these products:

- MKV58F1M0VMD24
- MKV58F1M0VLL24
- MKV58F1M0VLQ24
- MKV56F1M0VMD24
- MKV56F1M0VLL24
- MKV56F1M0VLQ24
- MKV58F512VMD24
- MKV58F512VLL24
- MKV58F512VLQ24
- MKV56F512VMD24
- MKV56F512VLL24
- MKV56F512VLQ24

Table 1. Errata and Information Summary

Erratum ID	Erratum Title
e9588	Core: Incorrect behavior of profiling counters
e9586	Core: Lock Status Indication incorrectly reads as one for debugger reads
e9587	Core: TPIU cannot be flushed in Debug state if Cortex-M7 TPIU is used
e10083	eFlexPWM: Half cycle automatic fault clearing does not work for PWM submodule 0 under certain conditions
e10420	Flashloader: Returns incorrect value to the host for property ID FlashBlockCount.
e8341	FlexCAN: Entering Freeze Mode or Low Power Mode from Normal Mode can cause the FlexCAN module to stop operating.
e9265	FTM: Incorrect match may be generated if intermediate load feature is used in toggle mode
e7735	MCG: IREFST status bit may set before the IREFS multiplexor switches the FLL reference clock
e4647	UART: Flow control timing issue can result in loss of characters if FIFO is not enabled

Table 2. Revision History

Revision	Changes
8 FEB 2016	Initial revision
14 JUN 2016	The following errata were added. • e10420

e9588: Core: Incorrect behavior of profiling counters

Description: ARM Errata 850724: The profiling counters are provided to enable non-intrusive counting of events with limited accuracy. Due to this erratum, some events update the wrong event counter.

The DWT_LSUCNT counter should be incremented for all extra cycles that the processor spends executing normal load-store instructions. Because of this erratum, stall cycles in the memory system will cause the DWT_CPICNT to be incremented instead of the DWT_LSUCNT. The DWT_LSUCNT will only count non-stalled extra cycles spent handling load-store instructions.

The DWT_EXCCNT counter should be incremented whilst lazy VFP state preservation is being performed.

Because of this erratum, these cycles increment the DWT_LSUCNT counter and, on stall cycles, the DWT_CPICNT instead.

Note that no cycles are missed entirely or double counted. This erratum only means that the cycles are counted in the wrong counter.

The following conditions are required to hit this erratum:

· The DEMCR.TRCENA bit is set

· Any or all of the following bits are set:

· DWT_CTRL.LSUEVTENA to enable the DWT_LSUCNT counter

· DWT_CTRL.CPICNT to enable the DWT_CPICNT counter

This erratum results in incorrect values in the profiling counters.

It has no other functional impact.

Workaround: There is no workaround for this erratum.

e9586: Core: Lock Status Indication incorrectly reads as one for debugger reads

Description: ARM Errata 851031: The FPB, DWT, and ITM components each implement a CoreSight lock mechanism. This lock is intended to prevent accidental software writes to control registers and is not required for external debugger accesses.

For an external debugger:

· LSR.LSI should RAZ to show that the lock is not implemented.

· LSR.SLK should RAZ to show that the component is not locked.

Because of this erratum, the LSR.LSI bit incorrectly reads as one for debugger accesses. An external debugger reading the LSR might be misled by the incorrect LSR.LSI value and attempt to obtain the lock. If the debugger polls until LSR.SLK reads as one, then the debugger could livelock. If the debugger does not poll on LSR.SLK, and instead assumes it has the lock and carries on, then there are no implications of this erratum.

Workaround: External debuggers should ignore the SLI field of the FPB_LSR, DWT_LSR and ITM_LSR registers. External tools should not attempt to lock or unlock the lock by writing to FPB_LAR, DWT_LAR and ITM_LAR registers.

e9587: Core: TPIU cannot be flushed in Debug state if Cortex-M7 TPIU is used

Description: ARM Errata 850725: The Cortex-M7 TPIU requires a DSYNC to flush any trace data that does not form a full frame and to allow trace synchronization.

A write to the DWT_CYCCNT register in the DWT is specified to generate this DSYNC to the Cortex-M7 TPIU.

This mechanism provides a means to flush out any buffered trace data in the TPIU.

Because of this erratum, a write to the DWT_CYCCNT register does not generate this DSYNC.

This erratum can cause the loss of the final bytes in a trace session. This applies to a maximum of 14 bytes. Additionally, it leaves an external debugger with no way to flush out any buffered trace data. This affects the standard step and trace model such that data generated during a step event will not be visible until a subsequent entry into halt mode.

Workaround: The DSYNC signal can be manually triggered, under the same conditions as the disabled method, by causing the processor to leave halted state and then halting the processor a second time.

e10083: eFlexPWM: Half cycle automatic fault clearing does not work for PWM submodule 0 under certain conditions

Description: When Submodule 0 selects the EXT_SYNC signal to cause initialization (PWMx_SM0_CTRL2[INIT_SEL] = 0b11) and a FAULTx input is associated with submodule 0 outputs using the Submodule 0 Fault Disable Mapping Registers (PWMx_SM0DISMAPn) and

1) the respective bit for that FAULTx is 1 in the FHALF bitfield of the Fault Status Register PWMx_FSTSn and

2) the respective bit for that FAULTx is 0 in the FFULL bitfield of the Fault Status Register PWMx_FSTSn and

3) the respective bit for that FAULTx is 1 in the FAUTO bitfield of the Fault Control Register PWM_FCTRLn,

then the automatic fault clearing will not work in submodule 0 and its outputs will remain disabled.

Workaround: When the EXT_SYNC signal is used to cause initialization in submodule 0 and the submodule 0 PWM outputs are disabled by a specific FAULTx input, use full cycle automatic fault clearing for the specific FAULTx input by setting the corresponding bit of the Fault Status Register PWM_FSTSn[FFULL].

e10420: Flashloader: Returns incorrect value to the host for property ID FlashBlockCount.

Description: The Kinetis Flashloader will return incorrect value to the host for property ID FlashBlockCount. It returns a value of 2 whereas the correct value is 1.

Workaround: None. Software suites utilizing the FlashBlockCount property should either ignore the FlashBlockCount property or hard-code this value to 1.

e8341: FlexCAN: Entering Freeze Mode or Low Power Mode from Normal Mode can cause the FlexCAN module to stop operating.

Description: In the Flexible Controller Area Network (FlexCAN) module, if the Freeze Enable bit (FRZ) in the Module Configuration Register (MCR) is asserted and the Freeze Mode is requested by asserting the Halt bit (HALT) in MCR, in some cases, the Freeze Mode Acknowledge bit (FRZACK) in the MCR may never be asserted.

In addition, the Low-Power Mode Acknowledge bit (LPMACK) in the MCR may never be asserted in some cases when the Low-Power Mode is requested.

Under the two scenarios described above, the loss of ACK assertion (FRZACK, LPMACK) causes a lock condition. A soft reset action is required in order to remove the lock condition.

The change from Normal Mode to Low-Power Mode cannot be done directly. Instead, first change mode from Normal to Freeze Mode, and then from Freeze to Low-Power Mode.

Workaround: To avoid the lock condition, the following procedures must be used:

A) Procedure to enter in Freeze Mode:

1. Set both the Freeze Enable bit (FRZ) and the Halt bit (HALT) in the Module Control Register (MCR).

2. Check if the Module Disable bit (MDIS) in MCR register is set. If yes, clear the MDIS bit.

3. Poll the MCR register until the Freeze Mode Acknowledge bit (FRZACK) in MCR is set or the timeout is reached (see NOTE below).

4. If the Freeze Mode Acknowledge bit (FRZACK) is set, no further action is required. Skip steps 5 to 8.

5. If the timeout is reached because the Freeze Mode Acknowledge bit (FRZACK) is still cleared, then set the Soft Reset bit (SOFTRST) in MCR.

6. Poll the MCR register until the Soft Reset bit (SOFTRST) bit is cleared.

7. Reconfigure the Module Control Register (MCR)

8. Reconfigure all the Interrupt Mask Registers (IMASKn).

After Step 8, the module will be in Freeze Mode.

NOTE: The minimum timeout duration must be equivalent to:

a) 730 CAN bits if the CAN FD Operation Enable bit (FDEN) in MCR is set (CAN bits calculated at arbitration bit rate),

b) 180 CAN bits if the FDEN bit is cleared.

B) Procedure to enter in Low-Power Mode:

1. Enter in Freeze Mode (execute the procedure A).
2. Request the Low-Power Mode.
3. Poll the MCR register until the Low-Power Mode Acknowledge (LPMACK) bit in MCR is set.

e9265: FTM: Incorrect match may be generated if intermediate load feature is used in toggle mode

Description: When a channel (n) match is used as an intermediate reload, an incorrect second match may occur immediately following the correct match. The issue is problematic only if channel (n) is configured for output compare with the output configured to toggle mode. In this scenario, channel (n) toggles on the correct match and again on the incorrect match. The issue may also occur if a certain channel has a match which is coincident with an intermediate reload point of any other channel.

Workaround: If any channel is configured for output compare mode with the output set for toggle mode, the intermediate reload feature must not be used.

e7735: MCG: IREFST status bit may set before the IREFS multiplexor switches the FLL reference clock

Description: When transitioning from MCG clock modes FBE or FEE to either FBI or FEI, the MCG_S[IREFST] bit will set to 1 before the IREFS clock multiplexor has actually selected the slow IRC as the reference clock. The delay before the multiplexor actually switches is:

2 cycles of the slow IRC + 2 cycles of OSCERCLK

In the majority of cases this has no effect on the operation of the device.

Workaround: In the majority of applications no workaround is required. If there is a requirement to know when the IREFS clock multiplexor has actually switched, and OSCERCLK is no longer being used by the FLL, then wait the equivalent time of:

2 cycles of the slow IRC + 2 cycles of OSCERCLK

after MCG_S[IREFST] has been set to 1.

e4647: UART: Flow control timing issue can result in loss of characters if FIFO is not enabled

Description: On UARTx modules with FIFO depths greater than 1, when the /RTS flow control signal is used in receiver request-to-send mode, the /RTS signal is negated if the number of characters in the Receive FIFO is equal to or greater than the receive watermark. The /RTS signal will not negate until after the last character (the one that makes the condition for /RTS negation true) is completely received and recognized. This creates a delay between the end of the STOP bit and the negation of the /RTS signal. In some cases this delay can be long enough that a transmitter will start transmission of another character before it has a chance to recognize the negation of the /RTS signal (the /CTS input to the transmitter).

Workaround: Always enable the RxFIFO if you are using flow control for UARTx modules with FIFO depths greater than 1. The receive watermark should be set to seven or less. This will ensure that there is space for at least one more character in the FIFO when /RTS negates. So in this case no data would be lost.

Note that only UARTx modules with FIFO depths greater than 1 are affected. The UARTs that do not have the RxFIFO feature are not affected. Check the Reference Manual for your device to determine the FIFO depths that are implemented on the UARTx modules for your device.

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.