

Mask Set Errata for Mask 0N51R

This report applies to mask 0N51R for these products:

- MKL82Z128VMC7
- MKL82Z128VLK7
- MKL82Z128VLL7
- MKL82Z128VLH7
- MKL82Z128VMP7

Table 1. Errata and Information Summary

Erratum ID	Erratum Title
e9407	LTC: Writing individual bytes of PKHA RAM causes adjacent bytes within the same 32-bit word to be corrupted.
e7735	MCG: IREFST status bit may be set before the IREFS multiplexor switches the FLL reference clock.
e9865	PMC: Current increases in VLPS mode when the PMC_REGSC[5] bit is not cleared.
e9462	QuadSPI: DQS Learning/Calibration does not support concurrent read transactions.
e9461	QuadSPI: Read data errors may occur with data learning in 4x sampling method.
e9627	ROM Bootloader: Cannot boot into QuadSPI DDR mode.
e9879	ROM Bootloader: User code may fail to transition to FEE clock mode after booting from ROM.
e9658	SPI: Inconsistent loading of shift register data into the receive FIFO following an overflow event.
e9857	TPM: TPM1 and TPM2 cannot function when TPM0 clock is gated off and SIM_SOPT2[TPMSRC] is set to 1.
e9646	WDOG: Unexpected watchdog behavior on LLS exit.

Table 2. Revision History

Revision	Date	Substantive changes
0	01/2016	Initial release

e9407: LTC: Writing individual bytes of PKHA RAM causes adjacent bytes within the same 32-bit word to be corrupted

Description: In LTC containing PKHA, the PKHA RAM is written from a 32-bit interface. Normally, each write consists of 4 bytes of data to be written. However, for writes of only 1-3 bytes, the non-written bytes within the same word are overwritten with incorrect data.

Workaround: Always write all 32-bits of any word within PKHA RAM. If modifying an individual byte within a word of PKHA RAM is required, first read the full word, merge in the byte(s) to be written, and then write back the entire new word.

e7735: MCG: IREFST status bit may be 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 is 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.

e9865: PMC: Current increases in VLPS mode when the PMC_REGSC[5] bit is not cleared

Description: The default value of Bit 5 of the PMC_REGSC register is set to 1, which results in the current increasing in VLPS mode. To obtain normal current in VLPS mode, this bit must be cleared.

Workaround: Clear the PMC_REGSC[5] bit to obtain normal current in VLPS mode.

e9462: QuadSPI: DQS Learning/Calibration does not support concurrent read transactions

Description: Learning/calibration in DQS sampling method is semi-automated. Coarse and fine delay values (configured using QuadSPI_MCR[SCLKCFG] and QuadSPI_SOCCR respectively) are changed to test whether the learning patterns are passing or failing.

During this time if concurrent read transactions from DMA or others master occurs, it might result in incorrect read data from flash.

Workaround: It must be ensured that while this calibration is ongoing, no other accesses to QuadSPI must be done.

e9461: QuadSPI: Read data errors may occur with data learning in 4x sampling method

Description: Data learning using the 4x Sampling method may select a sampling point that is marginal. A marginal sampling point occurs when the sampling point is located on the edge of the valid sampling window. A marginal sampling point may return a positive comparison of the data learning pattern but small variations in voltage and temperature during the same read transaction may result in data errors, since the sampling point is not properly located inside the valid sampling window.

Workaround: There are two options:

- Internal DQS method allows to perform data learning as described on the Reference Manual.
- If the 4x Sampling method is used, data learning should not be used and a fixed sampling point must be selected.

e9627: ROM Bootloader: Cannot boot into QuadSPI DDR mode

Description: Certain fields required to configure QuadSPI for DDR mode are not able to be set by the ROM Bootloader. Thus a workaround is required for the application image for the ROM to boot into DDR mode.

Workaround: When writing an application image to QuadSPI, a piece of code must be loaded first and executed from RAM to configure QuadSPI DDR mode before using the ROM Bootloader to write the image to QuadSPI. When booting from QuadSPI, the QuadSPI configuration block must be located in internal Flash memory and the application must start executing from internal Flash in order to configure QuadSPI DDR mode before jumping to a QuadSPI address. The KBLQSPIUG has more information on this setup.

e9879: ROM Bootloader: User code may fail to transition to FEE clock mode after booting from ROM

Description: In some use cases, user code may fail to transition to FEE clock mode after booting with the ROM bootloader. The issue occurs when all of the following conditions are met:

- FOPT[BOOTSRC_SEL] is configured to 0'b11.
- The 'enabledPeripherals' BCA field (offset address 0x10) enables the USB peripheral.

Workaround: There are three workarounds:

- Configure FOPT[BOOTSRC_SEL] to 0'b00 or 0'b10.
- Configure the 'enabledPeripherals' BCA field (offset address 0x10) to disable USB if using the FOPT[BOOTSRC_SEL] = 0'b11 option. For example, when booting from ROM and the QuadSPI is not being configured, then the ROM code must disable the USB.
- Ensure that the user code does not transition the clock to FEE mode if FOPT[BOOTSRC_SEL] = 0'b11 and the 'enabledPeripherals' BCA field enables USB. For example, when booting from ROM and the ROM code enables the USB, then afterward the user code must not transition the clock to FEE mode.

e9658: SPI: Inconsistent loading of shift register data into the receive FIFO following an overflow event

Description: In the Serial Peripheral Interface (SPI) module, when both the receive FIFO and shift register are full (Receive FIFO Overflow Flag bit in Status Register is set (SR [RFOF] = 0b1)) and then the Clear Rx FIFO bit in Module Configuration Register (MCR [CLR_RXF]) is asserted to clear the receive FIFO, shift register data is sometimes loaded into the receive FIFO after the clear operation completes.

Workaround:

- Avoid a receive FIFO overflow condition (SR[RFOF] should never be 0b1). To do this, monitor the RX FIFO Counter field of the Status Register (SR[RXCTR]), which indicates the number of entries in receive FIFO and clear before the counter equals the FIFO depth.
- Alternatively, after every receive FIFO clears operation (MCR[CLR_RXF] = 0b1) following a receive FIFO overflow (SR[RFOF] = 0b1) scenario, perform a single read from receive FIFO and discard the read data.

e9857: TPM: TPM1 and TPM2 cannot function when TPM0 clock is gated off and SIM_SOPT2[TPMSRC] is set to 1

Description: When SIM_SOPT2[TPMSRC] is set to 1 and TPM0 clock is gated off by SIM_SCGC6[TPM0], TPM1 and TPM2 cannot function.

Workaround: To ensure TPM1 and TPM2 count function properly, enable TPM0 by setting SIM_SCGC6[TPM0] to 1.

e9646: WDOG: Unexpected watchdog behavior on LLS exit

Description: When exiting LLS mode, the watchdog counter can increment in some cases. This can cause the watchdog to time out earlier than expected in applications where the watchdog is enabled and LLS mode is used.

Workaround: When entering or exiting LLS mode, refresh the watchdog to avoid triggering the timeout event.

How to Reach Us:

Home Page:
freescale.com

Web Support:
freescale.com/support

Information in this document is provided solely to enable system and software implementers to use Freescale 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.

Freescale reserves the right to make changes without further notice to any products herein. Freescale makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale 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 Freescale 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. Freescale does not convey any license under its patent rights nor the rights of others. Freescale sells products pursuant to standard terms and conditions of sale, which can be found at the following address: freescale.com/SalesTermsandConditions.

Freescale, the Freescale logo, and Kinetis are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. 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. All rights reserved.

© 2016 Freescale Semiconductor, Inc.

Document Number: KINETIS_L_0N51R
Rev. 0
16JAN2016

