Mask Set Errata for Mask 1N86K

This report applies to mask 1N86K for these products:

- MKL03Z16xxx4
- MKL03Z16xxx4R
- MKL03Z32xxx4
- MKL03Z32xxx4R
- MKL03Z8xxx4
- MKL03Z8xxx4R

Table 1. Errata and Information Summary

Erratum ID	Erratum Title
ERR008992	AWIC: Early NMI wakeup not detected upon entry to stop mode from VLPR mode
ERR008777	I2C: Address match wake-up from low-power mode cannot receive data
ERR009308	I2C: I2C does not hold bus between byte transfers in receive and may result in lost data
ERR009457	Kinetis Flashloader/ ROM Bootloader: The peripheral auto-detect code in bootloader can falsely detect presence of SPI host causing non-responsive bootloader
ERR008010	LLWU: CMP flag in LLWU_Fx register cleared by multiple CMP out toggles when exiting LLSx or VLLSx modes.
ERR010527	LPUART: Setting and immediately clearing SBK bit can result in transmission of two break characters
ERR008060	ROM: Bytes sent from host over I2C and SPI interfaces may be lost
ERR008058	ROM: COP can’t be re-enabled in application code due to being disabled by ROM boot code
ERR008086	ROM: Fail to setup connection by UART interface if RX pin is low after POR
ERR008059	ROM: Using UART at 57600 bits/s or greater with a core clock of 8 MHz may cause lost bytes
ERR008068	RTC: Fail to enter low power mode if RTC time invalid flag(TIF) is not cleared after POR
ERR008085	TPM: Writing the TPMx_MOD or TPMx_CnV registers more than once may fail when the timer is disabled

Table 2. Revision History

Revision	Changes
10	Initial revision

ERR008992: AWIC: Early NMI wakeup not detected upon entry to stop mode from VLPR mode

Description: Upon entry into VLPS from VLPR, if NMI is asserted before the VLPS entry completes, then the NMI does not generate a wakeup to the MCU. However, the NMI interrupt will occur after the MCU wakes up by another wake-up event.

Workaround: There are two workarounds:

1) First transition from VLPR mode to RUN mode, and then enter into VLPS mode from RUN mode.

2) Assert NMI signal for longer than 16 bus clock cycles.

ERR008777: I2C: Address match wake-up from low-power mode cannot receive data

Description: When the device is in a low-power mode that supports address match wake-up, receiving a matching address will wake up the MCU, however, the I2C will not respond correctly. I2C traffic after the wake-up address will not generate TCF interrupt events.

Workaround: Either one of the following sequences will enable the address match wake-up to operate correctly:

1) Send only the matching slave address followed by a repeated start and then resend the matching slave address including any subsequent data.

2) Send only the matching slave address followed by a stop condition and then resend the matching slave address including any subsequent data.

ERR009308: I2C: I2C does not hold bus between byte transfers in receive and may result in lost data

Description: When the I2C module is in receive mode, the bus is typically held by simply not reading the I2C_D register. However, in devices with this errata, the bus is not held between byte transfers by this action. If the I2C_D register and the data buffer are full, incoming data from an I2C device will overwrite data in the data buffer.

Workaround: When configured to receive data, the delay in processing incoming bytes should be minimized. Delay can be minimized by the use of DMA or increased interrupt priority.

ERR009457: Kinetis Flashloader/ ROM Bootloader: The peripheral auto-detect code in bootloader can falsely detect presence of SPI host causing non-responsive bootloader

Description: During the active peripheral detection process, the bootloader can interpret spurious data on the SPI peripheral as valid data. The spurious data causes the bootloader to shutdown all peripherals except the “falsely detected" SPI and enter the command phase loop using the SPI. After the bootloader enters the command phase loop using the SPI, the other peripherals are ignored, so the desired peripheral is no longer active.

The bootloader will not falsely detect activity on the I2C, UART, or USB interfaces, so only the SPI interface is affected.

Workaround: Ensure that there is an external pull-up on the SPI chip-select pin or that the pin is driven high. This will prevent the bootloader from seeing spurious data due to activity on the SPI clock pin.

ERR008010: LLWU: CMP flag in LLWU_Fx register cleared by multiple CMP out toggles when exiting LLSx or VLLSx modes.

Description: The comparator’s corresponding wakeup flag in the LLWU_Fx register is cleared prematurely if:

1.The CMP output is toggled more than one time during the LLSx wakeup sequence and the comparator’s corresponding flag in the LLWU_Fx register is cleared.

Or

2.The CMP output is toggled more than one time during the VLLSx wakeup sequence, PMC_REGSC[ACKISO] is cleared, and the comparator’s corresponding flag in the LLWU_Fx register is cleared.

Workaround: When MCU is waking up from LLS, code can implement a software flag to retain the wakeup source, if required by software.

When MCU is waking up from VLLSx, code can implement a software flag prior to clearing PMC_REGSC[ACKISO] to retain the wakeup source, if required by software.

ERR010527: 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 LUART_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.

ERR008060: ROM: Bytes sent from host over I2C and SPI interfaces may be lost

Description: Ping packet byte to the I2C and SPI loss occurs occasionally when the target is out of reset.

Workaround: 1.Run the I2C/SPI interface at 48 MHz core clock.

2.Insert a delay of at least 1 ms between the two bytes of the initial ping packet sent from the host to the device

3.If no response from the device is received, the host should re-send the ping packet

ERR008058: ROM: COP can't be re-enabled in application code due to being disabled by ROM boot code

Description: COP is disabled by ROM boot code,can't be re-enabled in application code. If the MCU boots up from ROM and then jumps into flash code, the flash code will fail to enable COP again.

Workaround: To use the COP watchdog in an application, write 00b to FOPT[BOOTSRC_SEL] bit to boot from flash out of reset.

ERR008086: ROM: Fail to setup connection by UART interface if RX pin is low after POR

Description: If LPUART0_RX pin is low after POR, the ROM bootloader auto-baud althgorithm can't get the correct baud rate, and hence it fails to comunicate via LPUART interface.

Workaround: If user want to use ROM boot loader via LPUART interface, keep LPUART0_RX pin high before communication. and one pull-up resistor is recommended.

ERR008059: ROM: Using UART at 57600 bits/s or greater with a core clock of 8 MHz may cause lost bytes

Description: When using default 8 MHz as the core clock, it may cause lost bytes if LPUART bit rate is configured to 57600 bits/s or higher.

Workaround: Run the LPUART at a maximum baud rate of 38,400 or lower with default 8 MHz core clock. Or configure core clock higher than 8 MHz.

ERR008068: RTC: Fail to enter low power mode if RTC time invalid flag(TIF) is not cleared after POR

Description: After POR, time invalid flag of RTC is set, that is, RTC_SR[TIF]=1. If this flag is not cleared, the MCU fails to enter low power mode.

Workaround: Clear time invalid flag of RTC before entering low power mode, this bit is cleared by writing the RTC_TSR register when the time counter is disabled.

ERR008085: TPM: Writing the TPMx_MOD or TPMx_CnV registers more than once may fail when the timer is disabled

Description: If writing the modulo register (TPMx_MOD) or channel value register (TPMx_CnV) more than once when the timer counter is disabled, and writes occur at a frequency faster than the TPM asynchronous clock, the registers may not update correctly.

This issue occurs when the register is written an even number of times but does not appear when the register is written an odd number of times.

If the TPM asynchronous clock is running at a higher frequency than the frequency of the writes, there are no issues writing these registers.

Workaround: When the TPMx_MOD or TPMx_CnV registers are written at a higher frequency than the TPM asynchronous clock there are two possible workarounds:

1) Do not write the TPMx_MOD or TPMx_CnV registers more than once before the timer counter is enabled.

2) If you need to update the TPMx_MOD or TPMx_CnV registers after they have been written once, before the timer counter is enabled, the new value must be written twice each time it is updated. This ensures that the registers are written an odd number of times.

How to Reach Us:

Home Page:
nxp.com

Web Support:
nxp.com/support