Freescale Semiconductor
Mask Set Errata

COLDFIREPLUS_3N89C
Rev 14 AUG 2013

Mask Set Errata for Mask 3N89C

Introduction

This report applies to mask 3N89C for these products:

- COLDFIREPLUS

Errata ID	Errata Title
3870	FTFL: Flash read errors in VLPR mode near upper range of system clock frequency specification.
2793	I2C: MCU does not wake as expected from STOP or VLPS mode on subsequent address matches if previous address is mismatched
4590	MCG: Transitioning from VLPS to VLPR low power modes while in BLPI clock mode is not supported.
6665	Operating requirements: Limitation of the device operating range
5130	SAI: Under certain conditions, the CPU cannot reenter STOP mode via an asynchronous interrupt wakeup event
5952	SMC: Wakeup via the LLWU from LLS/VLLS to RUN to VLPR incorrectly triggers an immediate wakeup from the next low power mode entry
7027	UART: During ISO-7816 T=0 initial character detection invalid initial characters are stored in the RxFIFO
7028	UART: During ISO-7816 initial character detection the parity, framing, and noise error flags can set
6472	UART: ETU compensation needed for ISO-7816 wait time (WT) and block wait time (BWT)
2582	UART: Flow control timing issue can result in loss of characters
4945	UART: ISO-7816 T=1 mode receive data format with a single stop bit is not supported
3892	UART: ISO-7816 automatic initial character detect feature not working correctly
7029	UART: In ISO-7816 T=1 mode, CWT interrupts assert at both character and block boundaries
7090	UART: In ISO-7816 mode, timer interrupts flags do not clear
5704	UART: TC bit in UARTx_S1 register is set before the last character is sent out in ISO7816 T=0 mode
7091	UART: UART_S1[NF] and UART_S1[PE] can set erroneously while UART_S1[FE] is set
7092	UART: UART_S1[TC] is not cleared by queuing a preamble or break character
5928	USBOTG: USBx_USBTRC0[USBRESET] bit does not operate as expected in all cases
3516	[PLL] If stop mode is entered with the PLL enabled and BDM active (debugger connected), the LOCK bit is incorrectly cleared when stop mode is exited.

Table continues on the next page...

Errata ID	Errata Title
3517	[PLL] Setting PLLS immediately after clearing PLLCLKEN will cause the LOCK bit to clear and stay cleared even after the PLL has re-started.

e3870: FTFL: Flash read errors in VLPR mode near upper range of system clock frequency specification.

Errata type: Errata

Description: The data sheet specifies that VLPR mode can operate with a system frequency of up to 2 MHz maximum. However, the actual maximum frequency is 1.2 MHz.

Workaround: Workaround 1: Before entering VLPR mode, select a system frequency of less than 1.2 MHz.

Workaround 2: If greater than 1.2 MHz (and less than 2 MHz) operation is desired in VLPR mode, execute out of RAM and do not access the Flash.

e2793: I2C: MCU does not wake as expected from STOP or VLPS mode on subsequent address matches if previous address is mismatched

Errata type: Errata

Description: The I2C module, acting as a slave on the I2C bus, does not wake as expected from normal STOP mode or VLPS mode on a valid address match if the previous address was not a match.

When the external I2C master sends a non-matching address, the I2C slave state machine does not look for a start bit past the first start bit on the bus. Consequently, subsequent transmissions by the I2C master with a matching address do not, on the first matching address, wake the MCU from stop mode or VLPS via the I2C interrupt.

Workaround: There are multiple workarounds:

(1) The master must continually re-transmit the MCU's slave address upon not receiving a NACK from the slave device during the slave addressing phase of the transmission. For clarification, the master must perform the following:

a) Send slave device address

b) Check for ACK bit

c) If ACK was received, continue with data transmission. Else, send repeated start signal and repeat steps a-c.

NOTE: Due to the nature of the errata, the maximum number of retransmissions needed to wake the part is nine times.

(2) When the MCU, operating as an I2C slave, is in STOP or VLPS mode: Ensure that the external I2C master sends a matching address to wake the slave MCU before it sends any transaction to other I2C slaves. The user must also ensure that MCU does not return to STOP or VLPS until after all packets to non-matching addresses have been sent.

(3) Use a pin interrupt (any pin, whether that pin is or is not being used by the active I2C module) to wake up the part before receiving I2C packets. NOTE: If using the SDA or SCL pin that the active I2C module is using, the part will wake-up on every I2C transaction on the bus.

(4) Use Wait mode instead of STOP or VLPS mode.

e4590: MCG: Transitioning from VLPS to VLPR low power modes while in BLPI clock mode is not supported.

Errata type: Errata

Description: Transitioning from VLPS mode back to VLPR (LPWUI control bit = 0) while using BLPI clock mode only, is not supported. During Fast IRC startup, the output clock frequency may exceed the maximum VLPR operating frequency. This does not apply to the BLPE clock mode.

Workaround: There are two options for workarounds

a) Exit to Run instead of VLPR. Before entering VLPR set the LPWUI bit so that when exiting VLPS mode the MCU exits to RUN mode instead of VLPR mode. With LPWUI set any interrupt will exit VLPR or VLPS back into RUN mode. To minimize the impact of the higher RUN current re-enter VLPR quickly.

or

b) Utilize MCG clock mode BLPE when transitioning from VLPS to VLPR modes.

e6665: Operating requirements: Limitation of the device operating range

Errata type: Errata

Description: Some devices, when power is applied, may not consistently begin to execute code under certain voltage and temperature conditions. Applications that power up with either VDD >= 2.0 V or temperature >= -20C are not impacted. Entry and exit of low-power modes is not impacted.

Workaround: To avoid this unwanted behavior, one or both of these conditions must be met:

a) Perform power on reset of the device with a supply voltage (VDD) equal-to or greater-than 2.0 V , or

b) Perform power on reset of the device at a temperature at or above -20 C.

e5130: SAI: Under certain conditions, the CPU cannot reenter STOP mode via an asynchronous interrupt wakeup event

Errata type: Errata

Description: If the SAI generates an asynchronous interrupt to wake the core and it attempts to reenter STOP mode, then under certain conditions the STOP mode entry is blocked and the asynchronous interrupt will remain set.

This issue applies to interrupt wakeups due to the FIFO request flags or FIFO warning flags and then only if the time between the STOP mode exit and subsequent STOP mode reentry is less than 3 asynchronous bit clock cycles.

Workaround: Ensure that at least 3 bit clock cycles elapse following an asynchronous interrupt wakeup event, before STOP mode is reentered.

e5952: SMC: Wakeup via the LLWU from LLS/VLLS to RUN to VLPR incorrectly triggers an immediate wakeup from the next low power mode entry

Errata type: Errata

Description: Entering VLPR immediately after an LLWU wakeup event from LLS/VLLS, will cause any subsequent entry into LLS/VLLS to fail if entry into VLPR mode occurs before clearing the pending LLWU interrupt.

Workaround: After an LLWU wakeup event from LLS/VLLS, the user must clear the LLWU interrupt prior to entering VLPR mode.

e7027: UART: During ISO-7816 T=0 initial character detection invalid initial characters are stored in the RxFIFO

Errata type: Errata

Description: When performing initial character detection (UART_C7816[INIT] = 1) in ISO-7816 T=0 mode with UART_C7816[ANACK] cleared, the UART samples incoming traffic looking for a valid initial character. Instead of discarding any invalid initial characters that are received, the UART will store them in the receive FIFO.

Workaround: After a valid initial charcter is detected (UART_IS7816[INITD] sets), flush the RxFIFO to discard any invalid initial characters that might have been received before the valid initial character.

e7028: UART: During ISO-7816 initial character detection the parity, framing, and noise error flags can set

Errata type: Errata

Description: When performing initial character detection (UART_C7816[INIT] = 1) in ISO-7816 mode the UART should not set error flags for any receive traffic before a valid initial character is detected, but the UART will still set these error flags if any of the conditions are true.

Workaround: After a valid initial charcter is detected (UART_IS7816[INITD] sets), check the UART_S1[NF, FE, and PF] flags. If any of them are set, then clear them.

e6472: UART: ETU compensation needed for ISO-7816 wait time (WT) and block wait time (BWT)

Errata type: Errata

Description: When using the default ISO-7816 values for wait time integer (UARTx_WP7816T0[WI]), guard time FD multiplier (UARTx_WF7816[GTFD]), and block wait time integer (UARTx_WP7816T1[BWI]), the calculated values for Wait Time (WT) and Block Wait Time (BWT) as defined in the Reference Manual will be 1 ETU less than the ISO-7816-3 requirement.

Workaround: To comply with ISO-7816 requirements, compensation for the extra 1 ETU is needed. This compensation can be achieved by using a timer, such as the low-power timer (LPTMR), to introduce a 1 ETU delay after the WT or BWT expires.

e2582: UART: Flow control timing issue can result in loss of characters

Errata type: Errata

Description: When /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: For UARTs that implement an eight entry FIFO: When the FIFO is enabled, 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.

For UARTs without a FIFO (or if the FIFO is disabled): Delay might need to be added between characters on the transmit side in order to allow time for the negation of /RTS to be recognized before the next character is sent.

e4945: UART: ISO-7816 T=1 mode receive data format with a single stop bit is not supported

Errata type: Errata

Description: Transmission of ISO-7816 data frames with single stop bit is supported in T=1 mode. Currently in order to receive a frame, two or more stop bits are required. This means that 11 ETU reception based on T=1 protocol is not supported. T=0 protocol is unaffected.

Workaround: Do not send T=1, 11 ETU frames to the UART in ISO-7816 mode. Use 12 ETU transmissions for T=1 protocol instead.

e3892: UART: ISO-7816 automatic initial character detect feature not working correctly

Errata type: Errata

Description: The ISO-7816 automatic initial character detection feature does not work. The direct convention initial character can be detected correctly, but the inverse convention initial character will only be detected if the S2[MSBF] and S2[RXINV] bits are set. This defeats the purpose of the initial character detection and automatic configuration of the S2[MSBF], S2[RXINV], and C3[TXINV] bits.

Workaround: Use software to manually detect initial characters. Configure the UART with S2[MSBF] and S2[RXINV] cleared. Then check UART receive characters looking for 0x3B or 0x03. If 0x3B is received, then the connected card is direct convention. If 0x03 is received, then the connected card is inverse convention. If an inverse convention card is detected, then software should set S2[MSBF], S2[RXINV], and C3[TXINV].

e7029: UART: In ISO-7816 T=1 mode, CWT interrupts assert at both character and block boundaries

Errata type: Errata

Description: When operating in ISO-7816 T=1 mode and switching from transmission to reception block, the character wait time interrupt flag (UART_IS7816[CWT]) should not be set, only block type interrupts should be valid. However, the UART can set the CWT flag while switching from transmit to receive block and at the start of transmit blocks.

Workaround: If a CWT interrupt is detected at a block boundary instead of a character boundary, then the interrupt flag should be cleared and otherwise ignored.

e7090: UART: In ISO-7816 mode, timer interrupts flags do not clear

Errata type: Errata

Description: In ISO-7816, when any of the timer counter expires, the corresponding interrupt status register bits gets set. The timer register bits cannot be cleared by software without additional steps, because the counter expired signal remains asserted internally. Therefore, these bits can be cleared only after forcing the counters to reload.

Workaround: Follow these steps to clear the UART_IS7816 WT, CWT, or BWT bits:

1. Clear the UART_C7816[ISO_7816E] bit, to temporarily disable ISO-7816 mode.

2. Write 1 to the WT, CWT, or BWT bits that need to be cleared.

3. Set UART_C7816[ISO_7816E] to re-enable ISO-7816 mode.

Note that the timers will start counting again as soon as the ISO_7816E bit is set. To avoid unwanted timeouts, software might need to wait until new transmit or receive traffic is expected or desired before re-enabling ISO-7816 mode.

e5704: UART: TC bit in UARTx_S1 register is set before the last character is sent out in ISO7816 T=0 mode

Errata type: Errata

Description: When using the UART in ISO-7816 mode, the UARTx_S1[TC] flag sets after a NACK is received, but before guard time expires.

Workaround: If using the UART in ISO-7816 mode with T=0 and a guard time of 12 ETU, check the UARTn_S1[TC] bit after each byte is transmitted. If a NACK is detected, then the transmitter should be reset.

The recommended code sequence is:

```
UART0_C2 &= ~UART_C2_TE_MASK; //make sure the transmitter is disabled at first

UART0_C3 |= UART_C3_TXDIR_MASK; //set the TX pin as output

UART0_C2 |= UART_C2_TE_MASK; //enable TX

UART0_C2 |= UART_C2_RE_MASK; //enable RX to detect NACK

for(i=0;i<length;i++)

{
```

```
while(!(UART0_S1&UART_S1_TDRE_MASK)){}
UART0_D = data[i];
while(!(UART0_S1&UART_S1_TC_MASK)){}//check for NACK
if(UART0_IS7816 & UART_IS7816_TXT_MASK)//check if TXT flag set
{
/* Disable transmit to clear the internal NACK detection counter */
UART0_C2 &= ~UART_C2_TE_MASK;
UART0_IS7816 = UART_IS7816_TXT_MASK;// write one to clear TXT
UART0_C2 |= UART_C2_TE_MASK; // re-enable transmit
}
}
UART0_C2 &= ~UART_C2_TE_MASK; //disable after transmit
```

e7091: UART: UART_S1[NF] and UART_S1[PE] can set erroneously while UART_S1[FE] is set

Errata type: Errata

Description: While the UART_S1[FE] framing error flag is set the UART will discard any received data. Even though the data is discarded, if characters are received that include noise or parity errors, then the UART_S1[NF] or UART_S1[PE] bits can still set. This can lead to triggering of unwanted interrupts if the parity or noise error interrupts are enabled and framing error interrupts are disabled.

Workaround: If a framing error is detected (UART_S1[FE] = 1), then the noise and parity error flags can be ignored until the FE flag is cleared. Note: the process to clear the FE bit will also clear the NF and PE bits.

e7092: UART: UART_S1[TC] is not cleared by queuing a preamble or break character

Errata type: Errata

Description: The UART_S1[TC] flag can be cleared by first reading UART_S1 with TC set and then performing one of the following: writing to UART_D, queuing a preamble, or queuing a break character. If the TC flag is cleared by queuing a preamble or break character, then the flag will clear as expected the first time. When TC sets again, the flag can be cleared by any of the three clearing mechanisms without reading the UART_S1 register first. This can cause a TC flag occurrence to be missed.

Workaround: If preamble and break characters are never used to clear the TC flag, then no workaround is required.

If a preamble or break character is used to clear TC, then write UART_D immediately after queuing the preamble or break character.

e5928: USBOTG: USBx_USBTRC0[USBRESET] bit does not operate as expected in all cases

Errata type: Errata

Description: The USBx_USBTCR0[USBRESET] bit is not properly synchronized. In some cases using the bit can cause the USB module to enter an undefined state.

Workaround: Do not use the USBx_USBTCR0[USBRESET] bit. If USB registers need to be written to their reset states, then write those registers manually instead of using the module reset bit.

e3516: [PLL] If stop mode is entered with the PLL enabled and BDM active (debugger connected), the LOCK bit is incorrectly cleared when stop mode is exited.

Errata type: Errata

Description: If stop mode is entered with the PLL enabled and BDM active (debugger connected), the LOCK bit is incorrectly cleared when stop mode is exited. This gives a false indication that the PLL has lost lock. The LOCK bit cannot be relied upon to identify if the PLL is truly locked. Once this occurs, if the PLL is then disabled and re-enabled, the LOCK bit will set correctly. In normal operation, BDM not active, the LOCK bit functions correctly.

Workaround: Perform one of the following workaround options.

Workaround Option 1

If entering stop mode with the pll enabled and BDM active (debugger connected), when exiting stop mode do not rely on the LOCK bit being set to verify the PLL is locked. The LOLS flag can be checked to ensure that it is not set to verify that PLL lock has not been lost.

-or-

Workaround Option 2

Disable and re-enable the PLL (by switching to a non-PLL mode or simply clearing PLLCLKEN if in a non-PLL mode). When the PLL is re-enabled the LOCK bit will be set correctly.

-or-

Workaround Option 3

During development, when the debugger is being used, the PLLSTEN bit can be set before entering STOP. This will keep the PLL running in STOP mode, which will also happen if BDM is active when STOP mode is entered. If the PLLSTEN bit is set then the LOCK bit will correctly remain set when STOP mode is exited. For final product production, the PLLSTEN bit should be cleared in order to reduce the power consumption in STOP mode.

e3517: [PLL] Setting PLLS immediately after clearing PLLCLKEN will cause the LOCK bit to clear and stay cleared even after the PLL has re-started.

Errata type: Errata

Description: Setting PLLS immediately after clearing PLLCLKEN will cause the LOCK bit to clear and stay cleared even after the PLL has re-started.

Workaround: Avoid disabling the PLL before immediately re-enabling it. If this cannot be avoided, add a single NOP instruction between clearing the PLLCLKEN bit and setting the PLLS bit.

How to Reach Us:

Home Page:
freescale.com

Web Support:
freescale.com/support