Mask Set Errata for Mask 2M25V

Introduction

This report applies to mask 2M25V for these products:

- PXD10

Errata ID	Errata Title
3194	ADC: Using the PIT trigger to start a conversion does not work under certain clock combination.
3442	CMU monitor: FXOSC/FIRC and FMPLL/FIRC relation
3449	DEBUG: Device may hang due to external or 'functional' reset while using debug handshaking mechanism
3230	DSPI: Correct usage for Tx FIFO when continuous clock mode is enabled
1082	DSPI: set up enough ASC time when MTFE=1 and CPHA=1
3110	Debugging functionality could be lost when unsecuring a secured device.
3324	FIRC - FIRC_CTL[TRIM] does not display correct trim value after reset
2656	FlexCAN: Abort request blocks the CODE field
3407	FlexCAN: CAN Transmitter Stall in case of no Remote Frame in response to Tx packet with RTR=1
2685	FlexCAN: Module Disable Mode functionality not described correctly
3021	LINFlex: Unexpected LIN timeout in slave mode
3219	MC_CGM: System clock may stop for case when target clock source stops during clock switching transition
3202	MC_ME: Invalid Configuration not flagged if PLL is on while OSC is off.
3269	MC_ME: Peripheral clocks may get incorrectly disabled or enabled after entering debug mode
3570	MC_ME: Possibility of Machine Check on Low-Power Mode Exit
3321	MC_RGM: A non-monotonic ramp on the VDDR line can cause the RGM module to clear all flags in the DES register, meaning that code cannot tell if a POR has occurred.
3060	MC_RGM: SAFE mode exit may be possible even though condition causing the SAFE mode request has not been cleared
3326	SIRC - SIRC_CTL[TRIM] does not display correct trim value after reset
3119	SWT: SWT interrupt does not cause STOP0 mode exit

e3194: ADC: Using the PIT trigger to start a conversion does not work under certain clock combination.

Errata type: n/a

Description: It is possible for the PIT to start an ADC conversion by enabling the TRGEN bit in the ADC MCR register. However, this does not work when the clock to the ADC is slower than the system clock.

The clock to the ADC is configured in the CGM_SC_DC2 register in the CGM. The system clock is configured through the Mode Entry module

Workaround: There are two possible workarounds:

1. Trigger conversions by writing to the ADC registers directly

2. When using the PIT trigger make sure that ADC is running at same clock frequency as the system clock.

e3442: CMU monitor: FXOSC/FIRC and FMPLL/FIRC relation

Errata type: Errata

Description: Functional CMU monitoring can only be guaranteed when the following conditions are met:

- FXOSC frequency must be greater than (FIRC / 2^RCDIV) + 0.5MHz in order to guarantee correct FXOSC monitoring

- FMPLL frequency must be greater than (FIRC / 4) + 0.5MHz in order to guarantee correct FMPLL monitoring

Workaround: Refer to description

e3449: DEBUG: Device may hang due to external or 'functional' reset while using debug handshaking mechanism

Errata type: Errata

Description: If the low-power mode debug handshake has been enabled and an external reset or a 'functional' reset occurs while the device is in a low-power mode, the device will not exit reset.

Workaround: The NPC_PCR[LP_DBG_EN] bit must be cleared to ensure the correct reset sequence.

e3230: DSPI: Correct usage for Tx FIFO when continuous clock mode is enabled

Errata type: n/a

Description: If the FIFO is enabled with continuous Serial Communication Clock (SCK) mode, a change in SCK frequency may occur if the Tx FIFO is not cleared and the Clock and Transfer Attributes Register 0 (CTAR0) is not used.

Workaround: When in continuous SCK mode, always use CTAR0 for the SPI transfer and always clear the TX-FIFO using the Clear TX FIFO (CLR_TXF) field of the Module Configuration Register (MCR) before initiating transfer.

e1082: DSPI: set up enough ASC time when MTFE=1 and CPHA=1

Errata type: Information

Description: When the DSPI is being used in the Modified Transfer Format mode (DSPI_MCR[MTFE]=1) with the clock phase set for Data changing on the leading edge of the clock and captured on the following edge in the DSPI Clock and Transfer Attributes Register (DSPI_CTARn[CPHA]=1), if the After SCK delay scaler (ASC) time is set to less than 1/2 SCK clock period the DSPI may not complete the transaction - the TCF flag will not be set, serial data will not received, and last transmitted bit can be truncated.

Workaround: If the Modified Transfer Format mode is required DSPI_MCR[MTFE]=1 with the clock phase set for serial data changing on the leading edge of the clock and captured on the following edge in the SCK clock (Transfer Attributes Register (DSPI_CTARn[CPHA]=1) make sure that the ASC time is set to be longer than half SCK clock period.

e3110: Debugging functionality could be lost when unsecuring a secured device.

Errata type: n/a

Description: Providing the backdoor password via JTAG or via serial boot would unsecure the device, but on some devices may leave the Nexus interface and potentially the CPUs in an undetermined state.

Normal operation without a debugger is unaffected, debugging unsecured devices is also unaffected.

Workaround: A second connection attempt may be successful.

Boot in serial mode (using the Flash password), then execute code which unsecures the device.(The JTAG interface needs to be inactive while the unsecure happens.)

Implement a separate backdoor in application software. Once the software detects the custom backdoor sequence it can unlock the device via Flash write.

Leave device unsecured for debugging.

e3324: FIRC - FIRC_CTL[TRIM] does not display correct trim value after reset

Errata type: n/a

Description: The FIRC is trimmed during reset using a factory programmed value stored in flash. However after reset the trim value is not copied to FIRCRC_CTL[TRIM] as one would expect. Any read of FIRCRC_CTL[TRIM] will read 0 and in all likelihood this will not be the factory programmed value. Therefore any read-modify-write on the 32-bit register will set the FIRC to a trim value of 0 and not the factory programmed value.

Workaround: As the lower 16-bits of FIRC_CTL register only contain the TRIM field it is recommended that if the user wishes to program any other field they should only access the upper 16-bits of this register.

If the user wishes to calibrate the FIRC this should be performed using the CMU.

e2656: FlexCAN: Abort request blocks the CODE field

Errata type: Errata

Description: An Abort request to a transmit Message Buffer (TxMB) can block any write operation into its CODE field. Therefore, the TxMB cannot be aborted or deactivated until it completes a valid transmission (by winning the CAN bus arbitration and transmitting the contents of the TxMB).

Workaround: Instead of aborting the transmission, use deactivation instead.

Note that there is a chance the the deactivated TxMB can be transmitted without setting IFLAG and updating the CODE field if it is deactivated.

e3407: FlexCAN: CAN Transmitter Stall in case of no Remote Frame in response to Tx packet with RTR=1

Errata type: Errata

Description: FlexCAN does not transmit an expected message when the same node detects an incoming Remote Request message asking for any remote answer.

The issue happens when two specific conditions occur:

1) The Message Buffer (MB) configured for remote answer (with code "a") is the last MB. The last MB is specified by Maximum MB field in the Module Configuration Register (MCR[MAXMB]).

2) The incoming Remote Request message does not match its ID against the last MB ID.

While an incoming Remote Request message is being received, the FlexCAN also scans the transmit (Tx) MBs to select the one with the higher priority for the next bus arbitration. It is expected that by the Intermission field it ends up with a selected candidate (winner). The coincidence of conditions (1) and (2) above creates an internal corner case that cancels the Tx winner and therefore no message will be selected for transmission in the next frame. This gives the appearance that the FlexCAN transmitter is stalled or "stops transmitting".

The problem can be detectable only if the message traffic ceases and the CAN bus enters into Idle state after the described sequence of events.

There is NO ISSUE if any of the conditions below holds:

a) The incoming message matches the remote answer MB with code "a".

b) The MB configured as remote answer with code "a" is not the last one.

c) Any MB (despite of being Tx or Rx) is reconfigured (by writing its CS field) just after the Intermission field.

d) A new incoming message sent by any external node starts just after the Intermission field.

Workaround: Do not configure the last MB as a Remote Answer (with code "a").

e2685: FlexCAN: Module Disable Mode functionality not described correctly

Errata type: Errata

Description: Module Disable Mode functionality is described as the FlexCAN block is directly responsible for shutting down the clocks for both CAN Protocol Interface (CPI) and Message Buffer Management (MBM) sub-modules. In fact, FlexCAN requests this action to an external logic.

Workaround: In FlexCAN documentation chapter:

Section "Modes of Operation", bullet "Module Disable Mode":

Where is written:

"This low power mode is entered when the MDIS bit in the MCR Register is asserted. When disabled, the module shuts down the clocks to the CAN Protocol Interface and Message Buffer Management sub-modules.."

The correct description is:

"This low power mode is entered when the MDIS bit in the MCR Register is asserted by the CPU. When disabled, the module requests to disable the clocks to the CAN Protocol Interface and Message Buffer Management sub-modules."

Section "Modes of Operation Details", Sub-section "Module Disable Mode":

Where is written:

"This low power mode is entered when the MDIS bit in the MCR Register is asserted. If the module is disabled during Freeze Mode, it shuts down the clocks to the CPI and MBM sub-modules, sets the LPM_ACK bit and negates the FRZ_ACK bit.."

The correct description is:

"This low power mode is entered when the MDIS bit in the MCR Register is asserted. If the module is disabled during Freeze Mode, it requests to disable the clocks to the CAN Protocol Interface (CPI) and Message Buffer Management (MBM) sub-modules, sets the LPM_ACK bit and negates the FRZ_ACK bit."

e3021: LINFlex: Unexpected LIN timeout in slave mode

Errata type: n/a
Description: If the LINFlex is configured in LIN slave mode, an unexpected LIN timeout event (LINESR[OCF]) may occur during LIN Break reception.

Workaround: It is recommended to disable this functionality during LINFlex initialization by clearing LINTCSR[IOT] and LINIER[OCIE] bits, and ignore timeout events.

e3219: MC_CGM: System clock may stop for case when target clock source stops during clock switching transition

Errata type: n/a
Description: The clock switching is a two step process. The availability of the target clock is first verified. Then the system clock is switched to the new target clock source within two target clock cycles.

For the case when the FXOSC stops during the required two cycles, the switching process may not complete, causing the system clock to stop and prevent further clock switching. This may happen if one of the following cases occurs while the system clock source is switching to FXOSC:

- FXOSC oscillator failure

- SAFE mode request occurs, as this mode will immediately switch OFF the FXOSC (refer to ME_SAFE_MC register configuration)

Workaround: The device is able to recover through any reset event ('functional', 'destructive', internal or external), so typically either the SWT (internal watchdog) will generate a reset or, in case it is used in the application, the external watchdog will generate an external reset. In all cases the devices will restart properly after reset.

To reduce the probability that this issue occurs in the application, disable SAFE mode transitions when the device is executing a mode transition with the FXOSC as the system clock source in the target mode.

e3202: MC_ME: Invalid Configuration not flagged if PLL is on while OSC is off.

Errata type: n/a

Description: PLL clock generation requires oscillator to be on. Mode configuration in which PLLON bit is "1" and OSCON bit is "0" is an invalid mode configuration. When ME_XXX_MC registers are attempted with such an invalid configuation, ME_IS.I_ICONF is not getting set which is wrong. Eventually the mode transition did not complete and system hangs.

Workaround: Always program Oscillator to be on when PLL is required.

e3269: MC_ME: Peripheral clocks may get incorrectly disabled or enabled after entering debug mode

Errata type: Errata

Description: If ME_RUN_PCx, ME_LP_PCx, ME_PCTLx registers are changed to enable or disable a peripheral, and the device enters debug mode before a subsequent mode transition, the peripheral clock gets enabled or disabled according to the new configuration programmed. Also ME_PSx registers will report incorrect status as the peripheral clock status is not expected to change on debug mode entry.

Workaround: After modifying any of the ME_RUN_PCx, ME_LP_PCx, ME_PCTLx registers, request a mode change and wait for the mode change to be completed before entering debug mode in order to have consistent behaviour on peripheral clock control process and clock status reporting in the ME_PSx registers.

e3570: MC_ME: Possibility of Machine Check on Low-Power Mode Exit

Errata type: Errata

Description: When executing from the flash and entering a Low-Power Mode (LPM) where the flash is in low-power or power-down mode, 2-4 clock cycles exist at the beginning of the RUNx to LPM transition during which a wakeup or interrupt will generate a checkstop due to the flash not being available on RUNx mode re-entry. This will cause either a checkstop reset or machine check interrupt.

Workaround: If the application must avoid the reset, two workarounds are suggested:

1) Configure the application to handle the machine check interrupt in RAM dealing with the problem only if it occurs

2) Configure the MCU to avoid the machine check interrupt, executing the transition into low power modes in RAM

There is no absolute requirement to work around the possibility of a checkstop reset if the application can accept the reset, and associated delays, and continue. In this event, the WKPU.WISR will not indicate the channel that triggered the wakeup though the F_CHKSTOP flag will indicate that the reset has occurred. The F_CHKSTOP flag could still be caused by other error conditions so the startup strategy from this condition should be considered alongside any pre-existing strategy for recovering from an F_CHKSTOP condition.

e3321: MC_RGM: A non-monotonic ramp on the VDDR line can cause the RGM module to clear all flags in the DES register, meaning that code cannot tell if a POR has occurred.

Errata type: Errata

Description: When system powers up, the 5V regulator supply (VDDR) shows a non-linearity in ramp up which coincides with the ramp-up of the 1.2V supply (VDD12). When a "slow" ramp is used the VDDR can dip to the LVDHV3 (2.9V - 2.6V) level causing it to re-fire.

With a "fast" ramp rate, the non linearity on 5V VDDR is shifted it does not dip to the LVDHV3 level. The F_POR flag is reported correctly in this instance.

If the LVDHV3 is re-fired just after the LVDLVCOR is released, then the POR & LVDHV3 flags will not be captured by the RGM. This is possible due to the soft startup mechanism of the voltage regulator power up profile.

This errata only affects the flag circuit and not a the device initialization. Device initializes correctly under all conditions.

Workaround: Potential solutions to avoid non-linearity from triggering LVDHV3:

1) Ensure that non linearity on VDDR is < 100mV . This is the hysteresis limit of the device. Board regulator should be chosen accordingly.

2) Make sure that VDDR ramp rate is fast enough. Criteria: when the VDD12 reaches its LVD levels, VDDR should have got enough margin from the LVDHV3 levels.

Software Workaround:

In the event of a boot where neither the F_POR or LVD27/LVD12 flags are set, the SRAM should be checked for ECC errors. If ECC errors are detected then it can be assumed that a POR has occurred.

e3060: MC_RGM: SAFE mode exit may be possible even though condition causing the SAFE mode request has not been cleared

Errata type: n/a

Description: A SAFE mode exit should not be possible as long as any condition that caused a SAFE mode entry is still active. However, if the corresponding status flag in the RGM_FES register has been cleared, the SAFE mode exit may incorrectly occur even though the actual condition is still active.

Workaround: Software must clear the SAFE mode request condition at the source before clearing the corresponding RGM_FES flag. This will ensure that the condition is no longer active when the RGM_FES flag is cleared and thus the SAFE mode exit can occur under the correct conditions.

e3326: SIRC - SIRC_CTL[TRIM] does not display correct trim value after reset

Errata type: n/a

Description: The SIRC is trimmed during reset using a factory programmed value stored in flash. After reset this trim value is not visible at SIRCRC_CTL[TRIM]. Therefore any read-modify-write on the 32-bit register will potentially set the SIRC to an unoptimised trim value.

Workaround: As the lower 16-bits of SIRC_CTL register only contain the TRIM field it is recommended that if the user wishes to program any other field they should only access the upper 16-bits of this register.

If the user wishes to calibrate the SIRC this should be performed using the CMU.

e3119: SWT: SWT interrupt does not cause STOP0 mode exit

Errata type: n/a

Description: While in STOP0 mode, if the SWT is configured to generate an interrupt and the system clock is disabled (ME_STOP0_MC[SYSCLK] = 0xF), a SWT interrupt event will not trigger an exit from STOP0 mode.

Other internal or external wakeup events (RTC, API, WKUP pins) are not affected and will trigger a STOP0 exit independent of the ME_STOP0_MC[SYSCLK] configuration.

Workaround: If a SWT interrupt is to be used to wake the device during STOP0 mode, software may not disable the system clock (ME_STOP0_MC[SYSCLK] != 0xF).

How to Reach Us:

Home Page:
www.freescale.com

Web Support:
http://www.freescale.com/support

USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, EL516
2100 East Elliot Road
Tempe, Arizona 85284
+1-800-521-6274 or +1-480-768-2130
www.freescale.com/support

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
www.freescale.com/support

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor China Ltd.
Exchange Building 23F
No. 118 Jianguo Road
Chaoyang District
Beijing 100022
China
+86 10 5879 8000
support.asia@freescale.com

Document Number: PXD10_2M25V
Rev. 11 APR 2012