Mask Set Errata for Mask 0M33Z

Introduction

This report applies to mask 0M33Z for these products:

- KINETIS

Errata ID	Errata Title
2550	ADC: ADC abort conversion logic error
3863	ADC: In 16-bit differential mode, ADC may result in a conversion error when positive input is near upper rail reference voltage
2776	CRC: May have incorrect CRC result when performing CRC 8-bit or 16-bit writes with transpose enabled.
2547	DAC: 12-bit DAC buffer registers cannot be read.
2670	ENET: IEEE 1588 TS_AVAIL interrupt is incorrectly mapped to NVIC vector 94
2579	ENET: No support for IEEE 1588, TS_TIMER, timestamp timer overflow interrupt
2596	EzPort and FTFL: The 64-bit flash programming command PGMSEC is not fully implemented.
2647	FMC: Cache aliasing is not supported on 512 KB and 384 KB program flash only devices.
2687	FMC: Flash clock divider setting for divide-by-1 not allowed
2448	FMC: Flash prefetch could result in incorrect read data
2671	FMC: Incorrect data returned during speculative access
2590	FMC: Master Access Protection encoding for write only access does not work as specified.
2683	FMC: Program flash only configuration cache/page buffer bank control does not follow specification.
2644	FMC: Speculation logic is not supported on program flash only or program flash only with swap feature devices.
2668	FTFL: Brown-out recovery during a 32-bit EEE write operation is not supported
3854	FTFL: Cannot connect to MCU via JTAG or EzPort interfaces after a Reset, POR, LVD event during a program sequence
2667	FTFL: EEE Brownout recovery after SETRAM command is not supported
2592	FTFL: EEESPLIT feature not fully implemented
3374	FTFL: Erase operation is not reliable if VDD is approximately 1.9 V or less
3372	FTFL: Reset during an EEE program operation may result in an invalid EEE read access

Table continues on the next page...

Errata ID	Errata Title
3855	FTFL: The SWAP feature is not supported.
2597	FTFL: Version ID field is not supported.
2784	FTFL: Warm reset during EEE copy-down prevents EEE copy down from successfully completing
2781	FlexBus: False bus error on back-to-back writes when flash memory is secure
2616	FlexCAN: Module receives data frames sent by itself although the self reception feature is disabled
2583	FlexCAN: OSCERCLK clock source must be enabled for CAN to function properly
2522	FlexCAN: Rx FIFO Overflow bit is not set when one frame was read after a match when the FIFO is full
2545	GPIO: PTB[0:11], PTB[16:23], PTC[0:19], and PTD[0:7] are not 5 V tolerant
3402	GPIO: XTAL pin cannot be used as GPIO if the ERCLKEN bit is set.
2793	I2C: MCU does not wake from STOP mode on subsequent address matches if previous address is mismatched
3795	I2S: Fractional divider in SIM_CLKDIV2 is not reset when recovering from VLLSx low power modes
3714	I2S: MCLK output is disabled in asynchronous mode
2613	LLWU: MCU may not exit properly from VLLS3, VLLS2 or VLLS1 modes via a pin reset or other wakeup sources.
2674	LLWU: The LLWU glitch filter for pin and reset is not supported
2680	MC: After recovery from LLS low power mode triggered by assertion of the RESET pin, the MC_SRSL[WAKEUP] bit is not set as expected.
2678	MC: The MC_SRS[PIN] is not always set after exiting a VLLS mode due to a RESET pin assertion
2676	MC: When waking the system from VLLS modes via a RESET pin, the I/O are not immediately released to their reset state.
3801	MCG: In FEE and FEI Modes, the FLL output may stop when the DCO range is changed
2553	MCG: MCGPLLCLK stalls for 2 PLL cycles if MCG_C5 is written to after the PLL is enabled
2556	MCG: PLLCLKEN may not always enable the PLL
2796	MCG: Slow Internal Reference Clock Operating Range does not meet specification.
2555	MCG: The DCO within the FLL cannot reach the minimum specified frequency
2660	MCG: The PLL can be enabled in bypassed low power mode (BLPE) when the system is in normal stop mode.
3580	MCG: Total deviation of trimmed average DCO output frequency over voltage and temperature does not meet specification.
2554	MCG: When trimming the slow Internal Reference Clock (IRC) using the auto trim machine (ATM), the IRC is not automatically enabled
3794	NVIC: NMI interrupt does not wakeup MCU from STOP and VLPS
2682	PIT: Does not generate a subsequent interrupt after clearing the interrupt flag
2548	PMC: Power-on reset (POR) sensitive to noise on Vdd
4482	PMC: STOP mode recovery unstable
2756	PMC: Static low power modes including STOP, VLPS, LLS, VLLSx are higher than specified.
2706	PMC: VLLS3 and VLLS2 recovery time is longer than specified
2542	PMC: Very Low Power Run (VLPR) and Very Low Power Wait (VLPW) power modes are not supported
2541	PORT: JTAG debug connectivity can not be achieved when using the RMII_RXER/MII_RXER alternate pin function on port pin PTA5 to connect to an Ethernet PHY
2578	RTC: Do not set RTC Update Mode via the RTC_CR[UM] bit

Table continues on the next page...

Errata ID	Errata Title
2573	RTC: The RTC interrupt is gated by the RTC_SR[TCE] bit. The interrupt will not assert on VBAT POR, VBAT power down, or when TCE = 0.
2575	RTC: Time Alarm Flag is cleared whenever the Seconds Counter increments and the Seconds Counter does not equal the Alarm Register
2655	RTC: When RTC clock gating is disabled, accessing the VBAT register file causes the bus to hang.
2577	RTC: When RTC_CR[CLKO] = 1, the 32 kHz RTC clock to the rest of the device is not disabled.
2576	RTC: When the RTC is configured to allow supervisor access only, the write and read access registers can be modified in user mode
2574	RTC: Writing RTC_TAR[TAR] = 0 does not disable RTC alarm
3997	Reset and Boot: MCU may fail to exit reset correctly.
3981	SDHC: ADMA fails when data length in the last descriptor is less or equal to 4 bytes
3982	SDHC: ADMA transfer error when the block size is not a multiple of four
4624	SDHC: AutoCMD12 and R1b polling problem
3977	SDHC: Does not support Infinite Block Transfer Mode
4627	SDHC: Erroneous CMD CRC error and CMD Index error may occur on sending new CMD during data transfer
3980	SDHC: Glitch is generated on card clock with software reset or clock divider change
3983	SDHC: Problem when ADMA2 last descriptor is LINK or NOP
3978	SDHC: Software can not clear DMA interrupt status bit after read operation
3984	SDHC: eSDHC misses SDIO interrupt when CINT is disabled
2572	TPIU: Trace Port Interface Unit (TPIU) data setup and hold times do not conform to the ARM timing specification, ARM IHI0014O, section 8.4, "Timing specifications"
2591	TSI: TSI_SCANC[SMOD] behaves as an inactive time instead of a scan period value
2638	TSI: The counter registers are not immediately updated after the EOSF bit is set.
2582	UART: Flow control timing issue can result in loss of characters
3892	UART: ISO-7816 automatic initial character detect feature not working correctly
2584	UART: Possible conflicts between UART interrupt service routines and DMA requests
2544	USB Voltage Regulator: Universal serial bus (USB) regulator standby mode is not supported
2666	VBAT: Applying a fast ramp (>50V/ms) on the VBAT pin can cause both VBAT and VDD to latch up
2686	WDOG: A watchdog reset while the system is in STOP or VLPS modes causes an incorrect wakeup sequence

e2550: ADC: ADC abort conversion logic error

Errata type: Errata

Description: The ADC abort conversion logic does not function as specified. Writes to the ADC CV1, CV2, OFS, PG, MG, CLPx, and CLMx registers will not abort a conversion.

Workaround: The abort conversion logic protects against changes to the ADC configuration during a conversion. To avoid this issue, do not change ADC settings during a conversion.

e3863: ADC: In 16-bit differential mode, ADC may result in a conversion error when positive input is near upper rail reference voltage

Errata type: Errata
Description: In 16-bit differential mode, the ADC may result in a conversion error when the input voltage on the plus-side of the differential pair (DPx) exceeds approximately (VREFH*31/32). Other modes are unaffected.

Workaround: To avoid a conversion error near positive full-scale in this mode, do not allow the input voltage on the plus-side of the differential pair (DPx) to exceed (VREFH*31/32).

e2776: CRC: May have incorrect CRC result when performing CRC 8-bit or 16-bit writes with transpose enabled.

Errata type: Errata
Description: If performing CRC 8-bit or 16-bit writes with transpose enabled, the final checksum may have an incorrect CRC result.

Workaround: Write accesses to the CRC when transpose is enabled should always be 32-bit.

e2547: DAC: 12-bit DAC buffer registers cannot be read.

Errata type: Errata
Description: The 12-bit DAC buffer registers, DACx_DAT[1:15]L and DACx_DAT[1:15]H, cannot be read. The data that is written to these registers cannot be read. Only DACx_DAT0L and DACx_DAT0H can be read correctly.

Workaround: Treat the DACx_DAT[1:15]L and DACx_DAT[1:15]H registers as write-only registers because reads may return invalid data. The DAC buffer can still be used since the values written to these registers are valid.

e2670: ENET: IEEE 1588 TS_AVAIL interrupt is incorrectly mapped to NVIC vector 94

Errata type: Errata
Description: The IEEE 1588 Transmit Timestamp Available interrrupt (TS_AVAIL) is incorrectly mapped to NVIC vector 94, the "Error and miscellaneous interrupt", instead of NVIC vector 91, which is the "IEEE 1588 Timer Interrupt ".

Workaround: In order to use the TS_AVAIL interrupt, it is recommeded that the user utilize NVIC vector 94 for this interrupt.

e2579: ENET: No support for IEEE 1588, TS_TIMER, timestamp timer overflow interrupt

Errata type: Errata
Description: The TS_TIMER interrupt signal is not connected to the NVIC and will not generate an interrupt event. This interrupt is set when the 1588 counter matches the period register.

Workaround: One of the 1588 counter channels can be configured in output compare software-only mode to generate the periodic interrupt events.

This can be used to generate a counter periodic interrupt:

Initialize the timer:

1) Set the ENET_ATPER to the desired value

2) Set the ENET_ATINC register to match the selected 1588 clock.

3) Set the ENET_TCCRn register with ENET_ATPER – ENET_ATINC[INC] value. The ENET_ATINC[INC] offset is needed to match the internal 1588 clock synchronization.

4) Set the ENET_TCSRn[TMODE] register with the 0100 encoding for output compare software-only mode and the ENET_TCSRn[TIE] to enable the timer interrupt.

5) Set the ENET_TCCRn register again with ENET_ATPER – ENET_ATINC[INC] value because output compare value is double buffered.

6) Set the ENET_ATCR[PEREN] to enable periodical event and set the ENET_ATCR[EN] to start the timer

Configure inside the ISR:

1) 1588 interrupts are generated via the NVIC vector 91 using the periodic timer. For each interrupt event, load the output compare buffer (ENET_TCCRn register) with ENET_ATPER – ENET_ATINC[INC] value.

2) Clear ENET_TCSRn[TF] flag

3) Clear ENET_TGSRn respective channel flag.

e2596: EzPort and FTFL: The 64-bit flash programming command PGMSEC is not fully implemented.

Errata type: Errata

Description: The 64-bit flash programming command PGMSEC is not fully implemented. The command can be used by flash programming algorithms and is used by the EzPort serial flash programming interface.

Workaround: Use the 32-bit flash programming command PGM4 in flash programming algorithms until the PGMSEC command is fully supported. Do not use the EzPort module until PGMSEC is fully supported.

e2647: FMC: Cache aliasing is not supported on 512 KB and 384 KB program flash only devices.

Errata type: Errata

Description: Due to logic error, cache aliasing is not supported on 512 KB and 384 KB program flash only devices. Out of reset the FMC_PFB0CR and FMC_PFB1CR registers which control the cache aliasing mechanism are configured for full cache/page buffer for both flash banks. Devices with FlexMemory or 128 KB, 192 KB and 256 KB program flash only are not affected.

Workaround: Disable the use of the cache for 512 KB and 384 KB program flash only devices by programming FMC_PFB0CR[4:3] and [0] and FMC_PFB1C[4:3] and [0] bits all to 1'b0.

e2687: FMC: Flash clock divider setting for divide-by-1 not allowed

Errata type: Errata

Description: If the Flash clock divider is set for divide-by-1 and a system reset occurs the Flash may be clocked at too high a frequency (>25MHz) and the system may access bad data from the Flash array during reset recovery sequencing.

Workaround: The Flash Controller should enable buffering to offset improve performance when Flash clock is configured for divide-by-2 or greater.

e2448: FMC: Flash prefetch could result in incorrect read data

Errata type: Errata

Description: A flash read access could result in incorrect data being returned if the SIM_DIV1 register is written to change the divide value for the flash clock and a flash prefetch occurs during the clock change sequence.

Workaround: 1. Create a function in the SRAM that performs the following operations:

a. Save value of FMC_PFAPR[23:16].

b. Disable flash prefetching by writing 8'hff to FMC_PFAPR[23:16].

c. Change value of flash divider SIM_DIV1[OUTDIV4].

d. Wait for clock change.

e. Re-enable flash prefetching by restoring the saved FMC_PFAPR[23:16] value.

f. Return from the function.

2. Call the SRAM code function.

3. Once the function returns from its SRAM execution, the flash clock divider has been safely altered.

e2671: FMC: Incorrect data returned during speculative access

Errata type: Errata

Description: If the D-flash speculation is enabled, then reading the D-flash while it is also being written (either through execution of a flash command or a EEE write) can return incorrect data with no error indication from the FMC. The FTFL block should still report a read collision error (FTFE_FSTAT[RDCOLERR] = 1); however, since the FTFL error is decoupled from the flash read it could be difficult to determine which D-flash read might have returned incorrect data.

NOTE: If the EEE function is not being used and the application does not execute any flash commands through the FTFL interface, then the application would not be affected by this errata.

Workaround: 1. When executing flash commands or writing the EEE, software can use the FTFL_FSTAT[CCIF] flag to determine when the write/erase is complete. When the CCIF flag sets it is safe to resume read accesses to the D-flash block.

2. If it is not practical or possible to check the CCIF flag before all D-flash reads or to suspend D-flash accesses during flash command or EEE writes, then the speculation buffer can be disabled. This will guarantee that the FMC correctly detects an error if a collision occurs.

e2590: FMC: Master Access Protection encoding for write only access does not work as specified.

Errata type: Errata

Description: The Access Protection Register (FMC_PFAPR) in the Flash Memory Controller provides write and/or read access control to the FlexRAM space on a per master granularity.

The eight two bit Master Access Protection fields in the FMC_PFAPR are encoded as follows:

00 - No access may be performed by this master

01 - Read only accesses may be performed by this master

10 - Write only accesses may be performed by this master

11 - Read and Write accesses may be performed by this master

The "10" encoding (write only accesses) does not work as specified. This encoding blocks all accesses.

So, the "10" encoding functions the same as the "00" encoding. Due to this error, it is not possible to designate the FMC memory space as "write only" for any master.

Note that the only writable portion of the FMC memory space is the FlexRAM space.

Workaround: The "11" encoding, allowing read and write accesses, must be programmed to allow writes for a given master.

e2683: FMC: Program flash only configuration cache/page buffer bank control does not follow specification.

Errata type: Errata

Description: Due to a logic error in P-Flash only configurations, the design only uses PFB0CR for cache/page buffer control and ignores PFB1CR.

NOTE:

1. This will only affect cache operation modes the banks are treated differently and different cache and/or page buffer control is desired per bank.

2. The design comes out of reset with full cache and page buffer functionality enabled for both banks, so this is not an issue out of reset.

3. Speculation control does follow the specification.

Workaround: 1. Keep FMC_PFB0CR at full cache/page buffer enablement (as it is already configured coming out of reset) for the best performance.

2. Program FMC_PFB0CR's and FMCPFB1CR's cache and page buffer control identically if something different than the default settings is desired.

e2644: FMC: Speculation logic is not supported on program flash only or program flash only with swap feature devices.

Errata type: Errata

Description: Due to a logic error, speculation logic is not supported on program flash only or program flash only with swap feature devices. Out of reset the FMC_PFB0CR and FMC_PFB1CR registers which control the speculation mechanism are configured for full speculation functionality for both flash banks. Devices with FlexMemory are not affected.

Workaround: Disable the use of speculation for program flash only and program flash only with swap feature devices by programming FMC_PFB0CR[2:1] and FMC_PFB1CR[2:1] bits to 2'h0.

e2668: FTFL: Brown-out recovery during a 32-bit EEE write operation is not supported

Errata type: Errata

Description: The firmware does not support the recovery of brown-outs when executing 32-bit writes to EEPROM. A 32-bit write normally creates two new EEPROM data records, if a brown-out occurs before the first record is programmed, brown-out recovery is valid. If a brown-out occurs between the first and second record creation, brown-out recovery is considered invalid since the first record will be valid but not the second, i.e. half of the longword write was backed up.

The firmware does support brown-outs during 8-bit and 16-bit writes to EERAM and associated programming of EEE data records.

Workaround: The recommendation is to execute only 8-bit and 16-bit EEPROM writes.

e3854: FTFL: Cannot connect to MCU via JTAG or EzPort interfaces after a Reset, POR, LVD event during a program sequence

Errata type: Errata

Description: If during execution of a flash erase event started by use of the MDM-AP control register mass erase mechanism, the Erase All Blocks command, or the Erase Flash Block command a Reset, POR, or LVD event occurs, the read reference levels of programmed ('0') bits can be modified. The impacted bits can affect critical flash parameters, IFR information, and user code programmed in the flash array and inhibit communication with the part via JTAG or EzPort interfaces. The part may inaccurately report as being secured, and un-secure methods will be ineffective.

Workaround: Ensure that Reset, POR, LVD events do not occur during: a) the program or erase sequence of code during execution of JTAG based programming tools or b) in-application code. For user code flash algorithms, the erase flash sector command can be used instead of erase all blocks or erase flash block commands. The erase flash sector command is not impacted.

e2667: FTFL: EEE Brownout recovery after SETRAM command is not supported

Errata type: Errata

Description: When the SETRAM command is executed after a brown-out event and before a EEE write occurs, brown-out recovery is not supported.

Workaround: After a brown-out event, do not execute SETRAM command prior to an EEPROM write.

1. Read an EEPROM location and write the data back.
2. Execute a write of 0xFFFF to unused but valid location before executing SETRAM.

Note: Without PGMSEC (64-bit programming concern), SETRAM is only needed if the user wants to use FlexRAM as traditional RAM.

e2592: FTFL: EEESPLIT feature not fully implemented

Errata type: Errata
Description: EEESPLIT feature not fully implemented within the firmware. Only the EEESPLIT = '11' case is supported.

Workaround: Use only the EEESPLIT = '11' case for a 50%/50% split of the FlexRAM for EEPROM.

e3374: FTFL: Erase operation is not reliable if VDD is approximately 1.9 V or less

Errata type: Errata
Description: Insufficient high voltage charge pump capacity causes an unreliable flash-memory erase capability over temperature if VDD is approximately 1.9 V or less.

Workaround: Maintain VDD above 1.9 V for proper flash-memory erase operations.

e3372: FTFL: Reset during an EEE program operation may result in an invalid EEE read access

Errata type: Errata
Description: In rare occurrences, a reset during an EEE program operation may result in invalid EEE read access.

Workaround: Avoid a reset during EEE programming.

e3855: FTFL: The SWAP feature is not supported.

Errata type: Errata
Description: The SWAP feature is not supported via the Swap Control Command.

Workaround: The SWAP feature is not supported and any attempt to use the Swap Control command will result in an error (FSTAT[ACCERR] will be set).

e2597: FTFL: Version ID field is not supported.

Errata type: Errata
Description: The Version ID is not accessible via the Read Resource command using Resource Select Code of 0x01 as described.

Workaround: The Version ID is not supported and any attempt to access this field using the Read Resource command will result in an error (FSTAT[ACCERR] will be set).

e2784: FTFL: Warm reset during EEE copy-down prevents EEE copy down from successfully completing

Errata type: Errata

Description: EEE copy-down procedure to transfer data from valid FlexMemory records to the FlexRAM is executed by the flash module after every warm reset. If a second warm reset occurs during the EEE copy-down procedure, the EEE copy-down will not complete even though the EEERDY flag in the Flash Configuration Register (FCNFG) will assert.

Workaround: After the FCNFG[EEERDY] and FSTAT[CCIF] flags have asserted and before accessing the FlexRAM, the SETRAM flash command must be launched with the control code set to 0x00 which results in the execution of the EEE copy-down procedure.

If the flash is not partitioned for EEE, the SETRAM command does not need to be launched. However, if the SETRAM command is launched and flash is not partitioned for EEE, the FSTAT[ACCERR] flag will set. The ACCERR flag must then be cleared before launching any other flash command.

e2781: FlexBus: False bus error on back-to-back writes when flash memory is secure

Errata type: Errata

Description: During back-to-back writes, the FlexBus incorrectly responds with a bus error on the second write when both of these conditions apply: the flash memory is secure (per the value of the FTFL module's FSEC[SEC] field), and the SIM's SOPT2[FBSL] field is 10b. This setting of the SOPT2[FBSL] field disallows instruction accesses but allows data accesses on the FlexBus interface when the flash memory is secure.

Workaround: When the flash memory is secure and FlexBus instruction accesses are inhibited but data accesses are allowed, do not use back-to-back writes. Insert a delay or NOP instruction between the write operations.

e2616: FlexCAN: Module receives data frames sent by itself although the self reception feature is disabled

Errata type: Errata

Description: The FlexCAN receives frames transmitted by itself although the self reception feature is disabled (MCR[SRX_DIS] is asserted). As a result, the transmitted data is moved into Mailbox and the IFLAG is asserted.

The error occurs when there is at least one reception Mailbox whose ID matches a frame that is being transmitted and the FlexCAN requests Freeze mode during the frame transmission.

The occurrence of this error depends on the software strategy used to request Freeze Mode and how often Freeze Mode is requested during module operation.

Workaround: If the self reception feature is disabled (MCR[SRX_DIS] = 1), in order to avoid receiving a self transmitted frame, Freeze Mode should only be requested when all Mailboxes that are configured as TX have been transmitted or aborted.

e2583: FlexCAN: OSCERCLK clock source must be enabled for CAN to function properly

Errata type: Errata

Description: The OSCERCLK clock source must be enabled in order for CAN to function properly. If the OSCERCLK clock is not enabled prior to initializing the module, the CANx_MCR[LPMACK] bit will be read as 1, showing that the module is disabled or in a low power state. This occurs even after the CANx_MCR[MDIS] has been written as a 0 while attempting to enable the module.

Workaround: Enable the OSCERCLK clock source in OSC module prior to initializing the CAN module. In order to enable the OSCERCLK clock source, set the OSC_CR[ERCLKEN] bit.

e2522: FlexCAN: Rx FIFO Overflow bit is not set when one frame was read after a match when the FIFO is full

Errata type: Errata

Description: When the FIFO is full and a new frame is being received, the matching process rejects all matches in the FIFO because it is full. If no further match occurs in the MBs, the frame will be lost and the CANx_IFLAG1[BUF7I] should be set to indicate an Rx FIFO Overflow.

However, the Rx FIFO Overflow bit is not set in the case when the FIFO is full for the following very specific scenario:

1) A frame is being received when the FIFO is full

2) A match occurs and all matches on the FIFO are rejected due to the FIFO being full;

3) No other match in the MB happens (frame is lost due FIFO full);

4) FIFO is emptied by a FIFO read AFTER the matching process finishes and BEFORE move-in start

The result is that the frame is not saved in FIFO and that the CANx_IFLAG1[BUF7I] bit is not set, which indicates that frame was lost due to a Rx FIFO Overflow.

Workaround: In order to avoid this issue, ensure that the Rx FIFO is not full. Limit the Rx FIFO to only hold five incoming frames instead of six and this will keep the Rx FIFO Overflow condition from occurring.

e2545: GPIO: PTB[0:11], PTB[16:23], PTC[0:19], and PTD[0:7] are not 5 V tolerant

Errata type: Errata

Description: PTB[0:11], PTB[16:23], PTC[0:19], and PTD[0:7] pins must not have their external voltage level driven above VDD, as these pins are not 5 V tolerant.

Note that if these pins are multiplexed with SLCD functionality, the LCD_Px pins can still drive 3 V and 5 V glass to correct voltage levels when the LCD is enabled. They are not affected by this erratum.

Workaround: Limit the external voltage on these pins to VDD.

e3402: GPIO: XTAL pin cannot be used as GPIO if the ERCLKEN bit is set.

Errata type: Errata

Description: XTAL pin cannot be used as GPIO if the ERCLKEN bit is set. Errata applies only when an external clock is being used (the crystal oscillator is not being used) and OSC_CR[ERCLKEN] bit set to 1. In this specific case, the analog block of the crystal oscillator is enabled and the oscillator output is driving the XTAL pin even if the respective pin control register has been configured as a GPIO. This prevents the pin from being used as either an input or an ouput.

Workaround: If an external clock is not being used, then the OSCERCLK is not available and there is no need to set the ERCLKEN bit. If the ERCLKEN bit is not set then the XTAL pin can be used as a GPIO.

If an external clock is being used but the OSCERCLK is not required, then the ERCLKEN bit should not be set and the XTAL pin can be used as a GPIO.

If an external clock is being used and the OSCERCLK is required and the ERCLKEN bit is set, then there is no workaround and the XTAL pin cannot be used as a GPIO.

e2793: I2C: MCU does not wake from STOP 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 from normal stop 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 correctly idle the I2C module. Subsequent transmissions by the I2C master with a matching address do not wake the MCU from stop mode via the I2C interrupt.

Workaround: There are multiple workarounds:

(1) When the MCU, operating as an I2C slave, is in stop 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 until after all packets to non-matching addresses have been sent.

(2) To bypass the stop mode wakeup issue: Enable the port interrupt function, which shares the pin function with the SDA input, to cause an interrupt on the falling edge of the SDA input signal {start}. Once the part is in run mode, restart the I2C module by clearing the IICEN bit and then setting the IICEN bit. (NOTE: Using the port interrupt function will wake the part on every SDA pin transitions)

(3) Use a pin interrupt (any pin that is not being used by the active I2C module) to wake up the part before receiving I2C packets.

(4) Use Wait mode instead of STOP mode.

e3795: I2S: Fractional divider in SIM_CLKDIV2 is not reset when recovering from VLLSx low power modes

Errata type: Errata

Description: The Fractional divider in the SIM_CLKDIV2 register is not reset when recovering from VLLSx low power modes.

Workaround: In order to reset the Fractional divider when recovering from VLLSx low power modes, the I2S clock gate must be enabled, then disabled, and re-enabled after exiting any VLLSx mode.

e3714: I2S: MCLK output is disabled in asynchronous mode

Errata type: Errata
Description: When using I2S MCLK output in asynchronous mode, the output is disabled.

Workaround: In order to output MCLK, use synchronous mode (I2Sx_CR[SYN]) along with setting the transmit direction to use an internally generated clock output through the serial transmit clock port (I2Sx_TCR[TXDIR]).

e2613: LLWU: MCU may not exit properly from VLLS3, VLLS2 or VLLS1 modes via a pin reset or other wakeup sources.

Errata type: Errata
Description: The MCU may not exit properly from VLLS3, VLLS2 or VLLS1 modes via a pin reset or other wakeup sources. Attempts to wakeup or reset from VLLS modes may result in the MCU entering an unknown, non-responsive state. In the non-responsive state, the MCU current will not match the RUN current prior to entering the low power mode.

Workaround: Exiting from VLLS modes requires that VDD be less than 2.0 V.

e2674: LLWU: The LLWU glitch filter for pin and reset is not supported

Errata type: Errata
Description: LLWU glitch filter for pin and reset is not supported. Do not enable the filters. Writing a 1 to FLTEP or FLTR bits in the LLWU_CS register will result in abnormal LLWU behavior.

Workaround: It is recommended that the glitch filter functionality is not used. Always write a zero to the FLTEP or FLTR bits in the LLWU_CS register. If filtering is required, external components or software filters are required.

e2680: MC: After recovery from LLS low power mode triggered by assertion of the RESET pin, the MC_SRSL[WAKEUP] bit is not set as expected.

Errata type: Errata
Description: After recovery from LLS low power mode triggered by assertion of the RESET pin, the MC_SRSL[WAKEUP] bit is not set as expected.

Workaround: Reset recovery software can use the current setting of MC_PMCTRL[LPLLSM] setting to know that the system is recovering from a LLS low power mode. A setting of MC_PMCTRL[LPLLSM] = %011 indicates that the system was configured for LLS.

e2678: MC: The MC_SRS[PIN] is not always set after exiting a VLLS mode due to a RESET pin assertion

Errata type: Errata

Description: When waking the device from VLLS low power modes via a RESET pin, the MC_SRSL[PIN] bit is not reliably set on recovery and cannot be used to differentiate VLLS wakeup from the RESET pin from other low power mode wakeup sources enabled in the LLWU module. This condition is encountered when the device is in a VLLS low power mode and the RESET pin is held asserted for greater than 100us to initiate a VLLS wakeup.

Workaround: When using the RESET pin as a wakeup trigger for exit from VLLS, ensure that the wakeup pulse is short in duration (<100us).

An alternate workaround is to only use the MC_SRS[WAKEUP] bit to indicate VLLS exit recoveries and reset initialization routines to not distinguish between RESET pin triggered recoveries and other sources of VLLS wakeup.

e2676: MC: When waking the system from VLLS modes via a RESET pin, the I/O are not immediately released to their reset state.

Errata type: Errata

Description: When waking the system from VLLS modes via a RESET pin, the I/O are not immediately released to their reset state.

Workaround: The reset initialization/recovery software must set the LLWU_CS[ACKISO] bit just as it would due to a low power wakeup via any of the other low power mode wakeup sources.

e3801: MCG: In FEE and FEI Modes, the FLL output may stop when the DCO range is changed

Errata type: Errata

Description: When the DCO range is changed (MCG_C4[DRST_DRS]), it is possible for the DCO output to get latched at a low or high state. A Reset event will not allow the FLL to recover from this condition. Only a power on reset will allow the FLL to recover from this condition. If the MCG is in an FLL engaged mode (FEI or FEE) and the DCO range is changed, it is possible that the system clock may stop.

Workaround: In order to allow the MCU to continue to receive a clock and detect if the FLL output has stopped, the following procedure shioukld be followed.

When changing ranges, the MCG should be set to an FLL bypassed mode (FBI or FBE). This allows the device to continue to receive a system clock when the DCO range is changed. Once the desired final range has been selected, the MCG_C1[CLKS] bits should be programed back to 00 to select the FLL output. The MCG_S[CLKST] bits should be polled for the equivalent time of at least five FLL reference clock cycles. If the value has not changed to 00 within this time, then the FLL output has stopped.In the event of this occurrence, the system software can continue to execute and identify that the FLL clock is no longer available.

In order to reduce the chance of occurrence of this issue the following steps should be taken :

1) The DCO range should not be changed abruptly but incremented or decremented one range step at a time.

2) Additionally, if the internal reference clock is being used as the FLL reference, the MCG_C3[SCTRIM] value should first be stored temporarily. Once the MCG is in FBI mode the MCG_C3[SCTRIM] should be written with 0xFF before the MCG_C4[DRST_DRS] value is changed. Once the final range has been selected, the MCG_C3[SCTRIM] value should be written back to the original value before switching the MCG back to FEI mode.

e2553: MCG: MCGPLLCLK stalls for 2 PLL cycles if MCG_C5 is written to after the PLL is enabled

Errata type: Errata

Description: This errata applies only to the MCGPLLCLK signal from the Multipurpose Clock Generator (MCG) and not the MCGOUT signal. After the PLL is enabled and locked, if you perform any write to the MCG_C5 register this will result in the MCGPLLCLK signal stalling for two clock cycles.

Workaround: The MCG_C5 register must be configured in the required manner before the PLL is enabled. This includes the configuration of the PLLSTEN bit. Avoid writing this register after the MCGPLLCLK is being used.

e2556: MCG: PLLCLKEN may not always enable the PLL

Errata type: Errata

Description: This errata applies only when enabling the PLL by means of the MCG_C5[PLLCLKEN] bit. Enabling the PLL by means of the MCG_C6[PLLS] bit does not have an issue. If the PLL is enabled by means of setting the MCG_C5[PLLCLKEN] bit and then disabled by clearing this bit, re-enabling of the PLL by means of this bit may not be possible.

Workaround: If you need to re-enable the PLL by means of the MCG_C5[PLLCLKEN] bit, the following sequence must first be performed.

1) Write a new value to the the MCG_C5[PRDIV] field.

2) Write the desired value back to the MCG_C5[PRDIV] field.

After completing this sequence, the MCG_C5[PLLCLKEN] bit may be set to enable the PLL.

e2796: MCG: Slow Internal Reference Clock Operating Range does not meet specification.

Errata type: Errata

Description: The slow internal reference clock (slow IRC) operating range does not meet the data sheet specification. The maximum frequency of the slow IRC is less than 33kHz with some units being significantly less than this. The factory trim value will be set to 0 on units that do not meet the datasheet target of 32768 Hz. This will set the slow IRC to the maximum frequency, which can be as low as 20 kHz on some units. This will result in units that do not operate at the expected 20.97MHz immediately after exiting reset. The FLL cannot be relied upon to provide a known stable clock frequency when using FLL engaged internal mode.

Workaround: If a known clock frequency is required then a clock mode other than FEI mode must be used. After exiting reset, the multipurpose clock generator should be transitioned from FEI mode to the desired mode. Refer to the appriopriate device reference manual for information on how to perform this.

e2555: MCG: The DCO within the FLL cannot reach the minimum specified frequency

Errata type: Errata

Description: The DCO range on the device is currently limited to the high side of the specified range. It is not recommended to operate the FLL below the following frequencies:

DRS Frequency

00 24 MHz

01 48 MHz

10 72 MHz

11 96 MHz

The FLL output frequency is generated by multiplying the frequency of a reference clock by a fixed amount. Since the DCO cannot be operated below a certain frequency this has a direct impact on the minimum reference clock frequency. This limits the minimum frequency of the reference clock. For the case of the internal reference clock, the range over which it can be used with the FLL is reduced.

Workaround: For this revision of the device, the internal reference has been factory trimmed to 39.0625 kHz. This allows for the following DCO configurations when using the factory trim value:

DRS DMX32 Freq

00 0 25 MHz

00 1 INVALID

01 0 50 MHz

01 1 INVALID

10 0 75 MHz

10 1 INVALID

11 0 100 MHz

11 1 INVALID

The maximum IRC frequency is used to ensure the FLL is operating correctly out of reset.

After exiting reset, you can re-trim the IRC to a lower frequency, or use an external clock, to run the FLL at a slower frequency but the minimum frequency specified in the description above must be met. The DMX32 bit can be used as long as the maximum FLL frequency is not exceeded and the minimum FLL frequency described above is met.

e2660: MCG: The PLL can be enabled in bypassed low power mode (BLPE) when the system is in normal stop mode.

Errata type: Errata

Description: When the MCG is in BLPE mode, it should not be possible to enable the PLL. However, in BLPE mode, if the MCG_C5[PLLSTEN] bit is set to 1 and normal stop mode is enterred, the PLL will be enabled. This will result in additional current consumption in normal stop mode. The MCG_C5[PLLSTEN] bit will behave as expected in all other MCG modes allowing the PLL to remain enabled.

Workaround: If normal stop mode is to be entered with the MCG in BLPE mode, the MCG_C5[PLLSTEN] bit must be cleared before stop mode is entered.

e3580: MCG: Total deviation of trimmed average DCO output frequency over voltage and temperature does not meet specification.

Errata type: Errata

Description: The total deviation of the trimmed average DCO output frequency over voltage and temperature does not meet specifications outlined in the Datasheet when using the internal reference (slow clock) as the reference to the FLL (FEI clock mode).

Workaround: For higher accuracy, it is recommended that the user utilize an External Reference Clock. The RTC Oscillator or the System Oscillator with external crystal should be used as a reference to the FLL in order to clock the system in FEE or FBE modes. Additionally, the PLL can also be used with MHz external clock, bypassing the FLL in PEE or PBE modes.

e2554: MCG: When trimming the slow Internal Reference Clock (IRC) using the auto trim machine (ATM), the IRC is not automatically enabled

Errata type: Errata

Description: This erratum applies only to the operation of using the Auto Trim Machine (ATM) to trim the slow IRC. It does not apply when trimming the fast IRC. When using the ATM to trim an IRC, the IRC selected by the MCG_ATC[ATMS] bit should be automatically enabled by the ATM. In the case of the slow IRC, the ATM does not enable the IRC correctly when MCG_ATC[ATMS] is 0.

Workaround: When using the ATM to trim the slow IRC, you must first enable the slow IRC before enabling the ATM. This is achieved by first clearing the MCG_C2[IRCS] bit to select the slow IRC and then setting the MCG_C1[IRCLKEN] bit to enable it. The MCG_ATC[ATMS] bit must also be cleared to select the slow IRC within the ATM.

e3794: NVIC: NMI interrupt does not wakeup MCU from STOP and VLPS

Errata type: Errata

Description: NMI interrupt does not wakeup MCU from STOP and VLPS when the bits CSYSPWRUPREQ and CDBGPWRUPREQ in the Control/Status Register of the DAP Port are cleared.

Workaround: If a debugger connection is established, the CSYSPWRUPREQ and CDBGPWRUPREQ bits are set by default, so an NMI interrupt will wake up the MCU from STOP and VLPS modes. In the absence of a debug connection and after a POR event, the bits will be cleared and thus an NMI interrupt will not wake the MCU.

e2682: PIT: Does not generate a subsequent interrupt after clearing the interrupt flag

Errata type: Errata

Description: The PIT does not generate a subsequent interrupt after clearing the interrupt flag in the ISR upon receiving the first interrupt. Thus, the second interrupt will never be triggered.

Workaround: In order to enable the use of subsequent interrupts from the PIT, the user must access any PIT register after clearing the interrupt flag in the ISR.

e2548: PMC: Power-on reset (POR) sensitive to noise on Vdd

Errata type: Errata
Description: After the MCU comes out of the initial power-up POR condition, any spike on Vdd can trigger a false POR.

Workaround: 1. Use a very low impedance connection between the power supply and the MCU Vdd pins. This helps with the Vdd spike when the MCU goes from a full load to a no-load condition.

2. Avoid any supply ramp rate greater than 2.0V/100mS after the first power-on supply ramp. For example, do not power up the MCU at 2 V and then ramp Vdd to 3 V later. If you want to ramp to 3 V, do that in a single ramp.

3. Provide a clean power supply to the MCU. Ripples on Vdd can cause PORs.

e4482: PMC: STOP mode recovery unstable

Errata type: Errata
Description: Recovery from STOP mode is not guaranteed if STOP mode is used for a period of time longer than 50ms.

Workaround: There are two methods that can be used:

1. Use a different low power mode such as VLPS.

2. Periodically exit STOP mode every 50ms and wait 16us before entering STOP mode again.

e2756: PMC: Static low power modes including STOP, VLPS, LLS, VLLSx are higher than specified.

Errata type: Errata
Description: Static low power modes including STOP, VLPS, LLS, VLLSx are not meeting specified values.

Workaround: Users should account for the higher current draw in their prototype evaluation. Refer to the datasheet for production target values.

e2706: PMC: VLLS3 and VLLS2 recovery time is longer than specified

Errata type: Errata
Description: The VLLS3 and VLLS2 recovery time is expected to be a maximum of 49 us as outlined in the data sheet. However, the recovery time is approximately 75 us.

Workaround: Use another low power stop mode with faster recovery time.

e2542: PMC: Very Low Power Run (VLPR) and Very Low Power Wait (VLPW) power modes are not supported

Errata type: Errata
Description: Very Low Power Run (VLPR) and Very Low Power Wait (VLPW) power modes are not supported.

Workaround: Do not use the part in these two power modes.

e2541: PORT: JTAG debug connectivity can not be achieved when using the RMII_RXER/MII_RXER alternate pin function on port pin PTA5 to connect to an Ethernet PHY

Errata type: Errata
Description: JTAG debug connectivity can not be achieved when using the RMII_RXER/MII_RXER alternate pin function on port pin PTA5 to connect to an Ethernet PHY. This is because JTAG_TRST is the POR default pin function for port pin PTA5. This causes a conflict when a board connects RMII_RXER/MII_RXER to an external Ethernet PHY, holding the JTAG controller in the reset state.

Workaround: Either workaround can be applied:

1. Do not connect the RMII_RXER/MII_RXER alternate function on the PTA5 pin to RMII/MII Ethernet PHY. This RX error signal is optional in the Ethernet interface.

2. Use serial wire debug (SWD) mode instead of full JTAG debug mode. There is no conflict with serial wire debug (SWD) mode and using the RX error signal in the Ethernet interface.

e2578: RTC: Do not set RTC Update Mode via the RTC_CR[UM] bit

Errata type: Errata
Description: If RTC Update Mode is set via RTC_CR[UM], the RTC Time Invalid Flag (RTC_SR[TIF]) is set and a software reset is required to clear it.

Workaround: RTC Update Mode (RTC_CR[UM] = 1) is not supported.

e2573: RTC: The RTC interrupt is gated by the RTC_SR[TCE] bit. The interrupt will not assert on VBAT POR, VBAT power down, or when TCE = 0.

Errata type: Errata
Description: RTC interrupt is gated by the RTC_SR[TCE] bit. The interrupt will not assert on VBAT POR, VBAT power down, or when TCE = 0.

Workaround: VBAT POR and VBAT power down interrupts are not supported.

e2575: RTC: Time Alarm Flag is cleared whenever the Seconds Counter increments and the Seconds Counter does not equal the Alarm Register

Errata type: Errata
Description: RTC Time Alarm Flag is cleared whenever the Seconds Counter increments and the Seconds Counter does not equal the Alarm Register.

Workaround: Service the Time Alarm Flag within one second after it asserts.

e2655: RTC: When RTC clock gating is disabled, accessing the VBAT register file causes the bus to hang.

Errata type: Errata
Description: When RTC clock gating is disabled in the SIM clock gating control register, accessing the VBAT register file causes the bus to hang. This occurs regardless if VBAT is powered or not.

Workaround: In order to avoid this issue, ensure that RTC clocking gating is enabled when accessing the VBAT regfile.

e2577: RTC: When RTC_CR[CLKO] = 1, the 32 kHz RTC clock to the rest of the device is not disabled.

Errata type: Errata
Description: When RTC_CR[CLKO] = 1, the 32 kHz RTC clock to the rest of the device is not disabled.

Workaround: The RTC clock output disable feature is not supported.

e2576: RTC: When the RTC is configured to allow supervisor access only, the write and read access registers can be modified in user mode

Errata type: Errata
Description: When the RTC is configured to allow supervisor access only, the write and read access registers can be modified in user mode. A bus error is still generated.

Workaround: RTC supervisor mode access only option is not supported.

e2574: RTC: Writing RTC_TAR[TAR] = 0 does not disable RTC alarm

Errata type: Errata
Description: Setting Alarm = 0 will not disable the alarm interrupt. Setting RTC_TSR[TSR] = 0 without initializing RTC_TAR will generate an alarm interrupt.

Workaround: Software must be capable of handling alarm interrupt. Another simple workaround is to initialize RTC_TAR[TAR] to 0xFFFFFFFF, avoiding the alarm interrupt for a long time period.

e3997: Reset and Boot: MCU may fail to exit reset correctly.

Errata type: Errata
Description: A small subset of prequalified devices may sporadically fail to exit reset. Power supply slew rate, voltage, and temperature affect issue occurrence.

Workaround: Procure the latest devices.

e3981: SDHC: ADMA fails when data length in the last descriptor is less or equal to 4 bytes

Errata type: Errata
Description: A possible data corruption or incorrect bus transactions on the internal AHB bus, causing possible system corruption or a stall, can occur under the combination of the following conditions:

1. ADMA2 or ADMA1 type descriptor

2. TRANS descriptor with END flag

3. Data length is less than or equal to 4 bytes (the length field of the corresponding descriptor is set to 1, 2, 3, or 4) and the ADMA transfers one 32-bit word on the bus

4. Block Count Enable mode

Workaround: The software should avoid setting ADMA type last descriptor (TRANS descriptor with END flag) to data length less than or equal to 4 bytes. In ADMA1 mode, if needed, a last NOP descriptor can be appended to the descriptors list. In ADMA2 mode this workaround is not feasible due to ERR003983.

e3982: SDHC: ADMA transfer error when the block size is not a multiple of four

Errata type: Errata
Description: Issue in eSDHC ADMA mode operation. The eSDHC read transfer is not completed when block size is not a multiple of 4 in transfer mode ADMA1 or ADMA2. The eSDHC DMA controller is stuck waiting for the IRQSTAT[TC] bit in the interrupt status register.

The following examples trigger this issue:

1. Working with an SD card while setting ADMA1 mode in the eSDHC

2. Performing partial block read

3. Writing one block of length 0x200

4. Reading two blocks of length 0x22 each. Reading from the address where the write operation is performed. Start address is 0x512 aligned. Watermark is set as one word during read. This read is performed using only one ADMA1 descriptor in which the total size of the transfer is programmed as 0x44 (2 blocks of 0x22).

Workaround: When the ADMA1 or ADMA2 mode is used and the block size is not a multiple of 4, the block size should be rounded to the next multiple of 4 bytes via software. In case of write, the software should add the corresponding number of bytes at each block end, before the write is initialized. In case of read, the software should remove the dummy bytes after the read is completed.

For example, if the original block length is 22 bytes, and there are several blocks to transfer, the software should set the block size to 24. The following data is written/stored in the external memory:

4 Bytes valid data

4 Bytes valid data

4 Bytes valid data

4 Bytes valid data

4 Bytes valid data

2 Bytes valid data + 2 Byte dummy data

4 Bytes valid data

4 Bytes valid data

4 Bytes valid data

4 Bytes valid data

4 Bytes valid data

2 Bytes valid data + 2 Byte dummy data

In this example, 48 (24 x 2) bytes are transferred instead of 44 bytes. The software should remove the dummy data.

e4624: SDHC: AutoCMD12 and R1b polling problem

Errata type: Errata

Description: Occurs when a pending command which issues busy is completed. For a command with R1b response, the proper software sequence is to poll the DLA for R1b commands to determine busy state completion. The DLA polling is not working properly for the ESDHC module and thus the DLA bit in PRSSTAT register cannot be polled to wait for busy state ompletion. This is relevant for all eSDHC ports (eSDHC1-4 ports).

Workaround: Poll bit 24 in PRSSTAT register (DLSL[0] bit) to check that wait busy state is over.

e3977: SDHC: Does not support Infinite Block Transfer Mode

Errata type: Errata

Description: The eSDHC does not support infinite data transfers, if the Block Count register is set to one, even when block count enable is not set.

Workaround: The following software workaround can be used instead of the infinite block mode:

1. Set BCEN bit to one and enable block count

2. Set the BLKCNT to the maximum value in Block Attributes Register (BLKATTR) (0xFFFFfor 65535 blocks)

e4627: SDHC: Erroneous CMD CRC error and CMD Index error may occur on sending new CMD during data transfer

Errata type: Errata

Description: When sending new, non data CMD during data transfer between the eSDHC and EMMC card, the module may return an erroneous CMD CRC error and CMD Index error. This occurs when the CMD response has arrived at the moment the FIFO clock is stopped. The following bits after the start bit of the response are wrongly interpreted as index, generating the CRC and Index errors.

The data transfer itself is not impacted.

The rate of occurrence of the issue is very small, as there is a need for the following combination of conditions to occur at the same cycle:

- The FIFO clock is stopped due to FIFO full or FIFO empty
- The CMD response start bit is received

Workaround: The recommendation is to not set FIFO watermark level to a too small value in order to reduce frequency of clock pauses.

The problem is identified by receiving the CMD CRC error and CMD Index error. Once this issue occurs, one can send the same CMD again until operation is successful.

e3980: SDHC: Glitch is generated on card clock with software reset or clock divider change

Errata type: Errata

Description: A glitch may occur on the SDHC card clock when the software sets the RSTA bit (software reset) in the system control register. It can also be generated by setting the clock divider value. The glitch produced can cause the external card to switch to an unknown state. The occurrence is not deterministic.

Workaround: A simple workaround is to disable the SD card clock before the software reset, and enable it when the module resumes the normal operation. The Host and the SD card are in a master-slave relationship. The Host provides clock and control transfer across the interface. Therefore, any existing operation is discarded when the Host controller is reset.

The recommended flow is as follows:

1. Software disable bit[3], SDCLKEN, of the System Control Register
2. Trigger software reset and/or set clock divider
3. Check bit[3], SDSTB, of the Present State Register for stable clock
4. Enable bit[3], SDCLKEN, of the System Control Register.

Using the above method, the eSDHC cannot send command or transfer data when there is a glitch in the clock line, and the glitch does not cause any issue.

e3983: SDHC: Problem when ADMA2 last descriptor is LINK or NOP

Errata type: Errata

Description: ADMA2 mode in the eSDHC is used for transfers to/from the SD card. There are three types of ADMA2 descriptors: TRANS, LINK or NOP. The eSDHC has a problem when the last descriptor (which has the End bit '1') is a LINK descriptor or a NOP descriptor.

In this case, the eSDHC completes the transfers associated with this descriptor set, whereas it does not even start the transfers associated with the new data command. For example, if a WRITE transfer operation is performed on the card using ADMA2, and the last descriptor of the WRITE descriptor set is a LINK descriptor, then the WRITE is successfully finished. Now, if a READ transfer is programmed from the SD card using ADMA2, then this transfer does not go through.

Workaround: Software workaround is to always program TRANS descriptor as the last descriptor.

e3978: SDHC: Software can not clear DMA interrupt status bit after read operation

Errata type: Errata

Description: After DMA read operation, if the SDHC System Clock is automatically gated off, the DINT status can not be cleared by software.

Workaround: Set HCKEN bit before starting DMA read operation, to disable SDHC System Clock auto-gating feature; after the DINT and TC bit received when read operation is done, clear HCKEN bit to re-enable the SDHC System Clock auto-gating feature.

e3984: SDHC: eSDHC misses SDIO interrupt when CINT is disabled

Errata type: Errata

Description: An issue is identified when interfacing the SDIO card. There is a case where an SDIO interrupt from the card is not recognized by the hardware, resulting in a hang.

If the SDIO card lowers the DAT1 line (which indicates an interrupt) when the SDIO interrupt is disabled in the eSDHC registers (that is, CINTEN bits in IRQSTATEN and IRQSIGEN are set to zero), then, after the SDIO interrupt is enabled (by setting the CINTEN bits in IRQSTATEN and IRQSIGEN registers), the eSDHC does not sense that the DAT1 line is low. Therefore, it fails to set the CINT interrupt in IRQSTAT even if DAT1 is low.

Generally, CINTEN bit is disabled in interrupt service.

The SDIO interrupt service steps are as follows:

1. Clear CINTEN bit in IRQSTATEN and IRQSIGEN.

2. Reset the interrupt factors in the SDIO card and write 1 to clear the CINT interrupt in IRQSTAT.

3. Re-enable CINTEN bit in IRQSTATEN and IRQSIGEN.

If a new SDIO interrupt from the card occurs between step 2 and step 3, the eSDHC skips it.

Workaround: The workaround interrupt service steps are as follows:

1. Clear CINTEN bit in IRQSTATEN and IRQSIGEN.

2. Reset the interrupt factors in the SDIO card and write 1 to clear CINT interrupt in IRQSTAT.

3. Clear and then set D3CD bit in the PROCTL register. Clearing D3CD bit sets the reverse signal of DAT1 to low, even if DAT1 is low. After D3CD bit is re-enabled, the eSDHC can catch the posedge of the reversed DAT1 signal, if the DAT1 line is still low.

4. Re-enable CINTEN bit in IRQSTATEN and IRQSIGEN.

e2572: TPIU: Trace Port Interface Unit (TPIU) data setup and hold times do not conform to the ARM timing specification, ARM IHI0014O, section 8.4, "Timing specifications"

Errata type: Errata

Description: Trace Port Interface Unit (TPIU) data setup and hold times do not conform to the ARM timing specification, ARM IHI0014O, section 8.4, "Timing specifications." Trace functionality of tools that access the TPIU is dependent on the third party tool vendor.

Workaround: Please contact the respective third party tool vendor for details on support for this feature.

e2591: TSI: TSI_SCANC[SMOD] behaves as an inactive time instead of a scan period value

Errata type: Errata

Description: TSI_SCANC[SMOD] should configure the scanning interval. TSI_SCANC[SMOD] interacts with the TSI_SCANC[AMCLKS] prescaler to configure this interval: Reference clock -> AMCLKS -> SMOD. So the interval frequency should be: Reference clock / AMCLKS prescaler / SMOD. However, it is working as an inactive time. Therefore, when an SMOD value is configured, the TSI will scan all the enabled electrodes and then be inactive for as much time as the SMOD and AMCLKS registers are configured.

Workaround: TSI_SCANC[SMOD] is designed to provide a predictable and configurable scanning interval. Depending on the scenario, there are three possible workarounds:

1. When SMOD = 0, the TSI will continuously scan without a pause between scans. This scan time is dependent on the electrode capacitance value and the current configured for the electrode. If the application will use only the out-of-range interrupt and it is not necessary to continually log each measured value after each scan, then an adequate threshold configuration and SMOD = 0 will cause the TSI to only interrupt or enable the out-of-range flag whenever there is a touch detected.

2. Manually control scanning time: use a time base to trigger single scans with the TSI_GENCS[SWTS] bit with the desired period. This workaround is recommended when a stable, predictable scan period is desired.

3. Use SMOD as a complement to the scanning time. The scanning time depends on the capacitance of the external electrodes. As mentioned above, the SMOD will currently control how much time the TSI is inactive after a scan so assumming the normal scan and the inactive time SMOD is currently determining will give a variable scanning period. If the application doesn't require an exact scanning period, this mode is recommended because adding inactive time after each scan will save power between scans.

e2638: TSI: The counter registers are not immediately updated after the EOSF bit is set.

Errata type: Errata

Description: The counter registers are not immediately updated after the end of scan event (EOSF is set). The counter registers will become available 0.25 ms after the EOSF flag is set. This also applies for the end-of-scan interrupt, as it is triggered with the EOSF flag. This behavior will occur both in continuous scan and in software triggered scan modes.

Workaround: Insert a delay of 0.25 ms or greater prior to accessing the counter registers after an end of scan even or an end of scan interrupt that is triggered by the EOSF flag. This delay does not need to be a blocking delay, so it can be executing other actions before reading the counter registers. Notice that the out-of-range flag (OUTRGF) and interrupt occur after the counters have been updated, so if the OUTRGF flag is polled or the out-of-range interrupt is used, the workaround is not necessary.

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.

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].

e2584: UART: Possible conflicts between UART interrupt service routines and DMA requests

Errata type: Errata

Description: If the UARTn_S1[RDRF] and/or UARTn_S1[TDRE] flags are being used to generate DMA requests, there is a possible conflict that could occur if an interrupt service routine (ISR) or other code is used to clear any of the other flags in the UARTn_S1 register. The flags in the UARTn_S1 register use a side effect clearing mechanism where the procedure is to read the status register and then perform a read or write of the data register to clear the flag. If a DMA request for a flag bit is asserted while an ISR for another flag bit is executing, then in the process of clearing the ISR's flag bit, the ISR can also clear the flag bit for the DMA request, thereby negating the DMA request before the DMA responds to it. This could potentially cause servicing of the DMA event to be missed.

For example, assume a DMA request is being asserted for the RDRF flag. At the same time, the parity error flag (PF) sets and triggers an ISR. To clear the PF flag bit, the ISR must read the status register and read the data register. In the process, the RDRF flag would also be cleared, causing the DMA request to negate. If the DMA request asserts after the DMA has already prepared its next transfer, then it might still read from the data register, potentially causing an underflow.

Workaround: When possible, avoid enabling the UART for DMA requests and interrupts simultaneously. If error interrupts are needed while DMA requests are active, then the error ISR can be used to abort the current DMA transfer (by disabling the DMA request inside the UART and/or disabling the external request for the DMA channel) before clearing any error flags in the UARTn_S1 register.

e2544: USB Voltage Regulator: Universal serial bus (USB) regulator standby mode is not supported

Errata type: Errata
Description: USB regulator standby mode is not supported.

Workaround: Do not use the USB standby mode.

e2666: VBAT: Applying a fast ramp (>50V/ms) on the VBAT pin can cause both VBAT and VDD to latch up

Errata type: Errata
Description: Do not apply a ramp rate faster than 50V/ms on VBAT pin. Doing so may induce a latchup condition on both VBAT and VDD.

Workaround: Ensure that the ramp rate is slower than 50V/ms on VBAT.

Another option is to add a resistor (100 ohn to 1 KOhm)in line with the VBAT supply pin. This impedance increase eliminates the latchup condition.

e2686: WDOG: A watchdog reset while the system is in STOP or VLPS modes causes an incorrect wakeup sequence

Errata type: Errata
Description: A watchdog reset while the system is in STOP or VLPS modes causes an incorrect wakeup sequence

Workaround: The watchdog should not be configured to continue operation when the system has entered STOP or VLPS.

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: KINETIS_0M33Z
Rev. 28 MAR 2012