# Chip Errata for the i.MX 6UltraLite

This document details the silicon errata known at the time of publication for the i.MX 6UltraLite multimedia applications processors.

Table 1 provides a revision history for this document.

**Table 1. Document Revision History**

| Rev. Number | Date | Substantive Changes |
|---|---|---|
| 2.3 | 09/2017 | • Added the following errata:<br>– ERR011115<br>– ERR011112 |
| 2.2 | 05/2017 | • Updated Figure 1, "Revision Level to Part Marking Cross-Reference"<br>• Updated the following Errata BSP status:<br>– ERR005778<br>– ERR007265<br>– ERR010690 |
| 2.1 | 03/2017 | • Updated the following Errata BSP status:<br>– ERR004446<br>– ERR005778<br>– ERR007265<br>– ERR007805<br>• Added the following errata:<br>– ERR004535 |
| 2 | 02/2017 | • Added the following errata:<br>– ERR010661<br>– ERR010690 |

**Table 1. Document Revision History (continued)**

| Rev. Number | Date | Substantive Changes |
|---|---|---|
| 1 | 04/2016 | • Updated the following errata:<br>– ERR008958<br>– ERR009606<br>– ERR009454<br>– ERR009455<br>• Added the following errata:<br>– ERR004446<br>– ERR009535<br>– ERR005829<br>– ERR005778<br>– ERR009596<br>– ERR006281 |
| 0.1 | 02/2016 | • Updated Figure 1, "Revision Level to Part Marking Cross-Reference" |
| 0 | 08/2015 | • Initial release |

Figure 1 provides a cross-reference to match the revision code to the revision level marked on the device.

MC IMX6 X @ + VV $$ % A

| Qualification Level | MC |
|---|---|
| Prototype Samples | PC |
| Mass Production | MC |
| Special | SC |

| Silicon Rev | A |
|---|---|
| Rev. 1.0 (Maskset ID: 0N52P) | A |
| Rev. 1.1 (Maskset ID: 1N52P) | |
| Rev. 1.2 (Maskset ID 2N52P) | B |

| Fuse Option | % |
|---|---|
| Reserved | A |

| i.MX 6 Family | X |
|---|---|
| i.MX 6UltraLite | G |

| Arm Cortex-A7 Frequency | $$ |
|---|---|
| 528 MHz | 05 |
| 696 MHz | 07 |

| Package Type | ROHS |
|---|---|
| MAPBGA 14x14 0.8 mm | VM |
| MAPBGA 9x9 0.5 mm | VK |

**Part Differentiator**

| | Package | Enhanced Security | Standard Security | eFuse bit | L2 Cache | USB with PHY | Ethernet (10/100M) | CAN | UART | I2C | SPI | I2S | Timer/PWM | ADC | CSI | LCD | @ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Commercial | | Y | Y | 2048 | 128 KB | 2 | 2 | 2 | 8 | 4 | 4 | 3 | 4/8 | 2 | Y | Y | |
| Industrial | VM | Y | Y | 2048 | 128 KB | 2 | 2 | 2 | 8 | 4 | 4 | 3 | 4/8 | 2 | Y | Y | |
| Commercial | | Y | Y | 2048 | 128 KB | 2 | 2 | 2 | 8 | 4 | 4 | 3 | 4/8 | 2 | Y | Y | |
| Industrial | VK | Y | Y | 2048 | 128 KB | 2 | 2 | 2 | 8 | 4 | 4 | 3 | 4/8 | 2 | Y | Y | 3 |
| Automotive | | - | Y | 1536 | 128 KB | 2 | 2 | 2 | 8 | 4 | 4 | 3 | 4/8 | 2 | Y | Y | |
| Commercial | | | Y | 1536 | 128 KB | 2 | 2 | 2 | 8 | 4 | 4 | 3 | 4/8 | 2 | Y | Y | |
| Industrial | VM | - | Y | 1536 | 128 KB | 2 | 2 | 2 | 8 | 4 | 4 | 3 | 4/8 | 2 | Y | Y | |
| Commercial | | | Y | 1536 | 128 KB | 2 | 2 | 2 | 8 | 4 | 4 | 3 | 4/8 | 2 | Y | Y | |
| Industrial | VK | - | Y | 1536 | 128 KB | 2 | 2 | 2 | 8 | 4 | 4 | 3 | 4/8 | 2 | Y | Y | 2 |
| Automotive | | - | Y | 1024 | 128 KB | 2 | 1 | 1 | 8 | 4 | 4 | 3 | 4/8 | 1 | - | - | |
| Industrial | VM | - | Y | 1024 | 128 KB | 2 | 1 | 1 | 8 | 4 | 4 | 3 | 4/8 | 1 | - | - | 1 |
| Commercial | VM | - | - | 512 | 0 KB | 1 | 1 | 0 | 4 | 2 | 2 | 1 | 2/4 | 1 | - | - | 0 |

| Junction Temperature (Tj) | + |
|---|---|
| Commercial: 0 to + 95 °C | D |
| Industrial: -40 to +105 °C | C |
| Auto: -40 to + 125 °C | A |

**Figure 1. Revision Level to Part Marking Cross-Reference**

For details on the Arm® configuration used on this chip (including Arm module revisions), please see the “Platform configuration” section of the “Arm Cortex®-A7 MPCore Platform” chapter of the *i.MX 6UltraLite Applications Processor Reference Manual*.

Table 2 summarizes errata on the i.MX 6UltraLite.

**Table 2. Summary of Silicon Errata**

| Errata | Name | Solution | Page |
|---|---|---|---|
| **Arm® – Cortex A7** | | | |
| ERR008958 | Arm/MP: 814220—B-Cache maintenance by set/way operations can execute out of order | No fix scheduled | 6 |
| ERR008959 | Arm/MP: 809719—C PMU events 0x07, 0x0C, and 0x0E do not increment correctly | No fix scheduled | 8 |
| ERR008960 | Arm/MP: 805420—C PMU event counter 0x14 does not increment correctly | No fix scheduled | 10 |
| ERR008961 | Arm/MP: 804069—C Exception mask bits are cleared when an exception is taken in Hyp Mode | No fix scheduled | 11 |
| **CCM** | | | |
| ERR007265 | CCM: When improper low-power sequence is used, the SoC enters low power mode before the Arm core executes WFI | No fix scheduled | 12 |
| **Clock** | | | |
| ERR009455 | Clock: 24 MHz Oscillator does not start up | Fixed in silicon revision 1.1 | 13 |
| **eCSPI** | | | |
| ERR009606 | eCSPI: In master mode, burst lengths of 32n + 1 will transmit incorrect data | No fix scheduled | 14 |
| ERR009535 | eCSPI: Burst completion by SS signal in slave mode is not functional | No fix scheduled | 15 |
| **EIM** | | | |
| ERR004446 | EIM: AUS mode is nonfunctional for devices larger than 32 MB | No fix scheduled | 16 |
| **FlexCAN** | | | |
| ERR005829 | FlexCAN: FlexCAN does not transmit a message that is enabled to be transmitted in a specific moment during the arbitration process | No fix scheduled | 17 |
| **I2C** | | | |
| ERR007805 | I2C: When the I2C clock speed is configured for 400 kHz, the SCL low period violates the I2C specification | No fix scheduled | 19 |
| **MMDC** | | | |
| ERR005778 | MMDC: DDR Controller's measure unit may return an incorrect value when operating below 100 MHz | No fix scheduled | 20 |
| ERR009596 | MMDC: ARCR_GUARD bits of MMDC Core AXI Re-ordering Control register (MMDC_MAARCR) does not behave as expected | No fix scheduled | 21 |
| **Power Management** | | | |
| ERR009454 | Power Management: PMIC_STBY_REQ output voltage lower than specification | Fixed in silicon revision 1.1 | 26 |

Table 2. Summary of Silicon Errata (continued)

| Errata | Name | Solution | Page |
|---|---|---|---|
| **PXP** | | | |
| ERR009541 | PXP: PXP CSC2 cannot do RGB2YCbCr or RGB2YUV conversion correctly | No fix scheduled | 22 |
| **System Boot** | | | |
| ERR011115 | System Boot: Boot failures on certain devices when the boot image targets OCRAM | No fix scheduled | 23 |
| ERR011112 | System Boot: QSPI boot failure when the boot image targets OCRAM | No fix scheduled | 24 |
| **SNVS** | | | |
| ERR010690 | SNVS: SNVS_LP Registers Reset Issue | Fixed scheduled in silicon revision 1.2 | 25 |
| **USB** | | | |
| ERR004535 | USB: USB suspend and resume flow clarifications | No fix scheduled | 27 |
| ERR006281 | USB: Incorrect DP/DN state when only VBUS is applied | No fix scheduled | 28 |
| ERR010661 | USB: VBUS leakage occurs if USBOTG1 VBUS is on and USBOTG2 VBUS transitions from on to off | No fix scheduled | 29 |

## ERR008958 Arm/MP: 814220—B-Cache maintenance by set/way operations can execute out of order

### Description:

The v7 Arm states that all cache and branch predictor maintenance operations that do not specify an address execute relative to each other, must occur in program order. However, because of this erratum, a L2 set/way cache maintenance operation can overtake a L1 set/way cache maintenance operation.

### Conditions:

For this erratum to have an observable effect, the following conditions must be met.

1. A CPU performs an L1 DCCSW or DCCISW operation.
2. The targeted L1 set/way contains dirty data.
3. Before the next DSB, the same CPU executes an L2 DCCSW or DCCISW operation while the L1 set/way operation is in progress.
4. The targeted L2 set/way is within the group of L2 set/way that the dirty data from L1 can be allocated to.

If the above conditions are met then the L2 set/way operation can take effect before the dirty data from L1 has been written to L2.

**NOTE**

Conditions (3) and (4) are not likely to be met concurrently when performing set/way operation on the entire L1 and L2 caches. This is because cache maintenance code is likely to iterate through sets and ways in a consistent ascending or descending manner across cache levels, and to perform all operations on one cache level before moving on to the next cache level. This means that, for example, cache maintenance operations on L1 set 0 and L2 set 0 will be separated by cache maintenance operations for all other sets in the L1 cache. This creates a large window for the cache maintenance operations on L1 set 0 to complete.

### Projected Impact:

Code that intends to clean dirty data from L1 to L2 and then from L2 to L3 using set/way operations might not behave as expected. The L2 to L3 operation might happen first and result in dirty data remaining in L2 after the L1 to L2 operation has completed.

If dirty data remains in L2 then an external agent, such as a DMA agent, might observe stale data.

If the processor is reset or powered-down while dirty data remains in L2 then the dirty data will be lost.

### Workarounds:

Correct ordering between set/way cache maintenance operations can be forced by executing a DSB before changing cache levels.

## Proposed Solution:

No fix scheduled

## Linux BSP Status:

BSP workaround implemented in the Linux BSP GA release

## ERR008959 Arm/MP: 809719—C PMU events 0x07, 0x0C, and 0x0E do not increment correctly

### Description:

The Cortex-A7 MPCore processor implements version 2 of the Performance Monitor Unit architecture (PMUv2). The PMU can gather statistics on the operation of the processor and memory system during runtime. This event information can be used when debugging or profiling code.

The PMU can be programmed to count architecturally executed stores (event 0x07), software changes of the PC (event 0x0C), and procedure returns (event 0x0E). However, because of this erratum, these events do not fully adhere to the descriptions in the PMUv2 architecture.

### Conditions:

Either

1. A PMU counter is enabled and programmed to count event 0x07. That is: instruction architecturally executed, condition code check pass, and store.
2. A PLDW instruction is executed. If the above conditions are met, the PMUv2 architecture specifies that the counter for event 0x07 does not increment. However, the counter does increment.

Or

1. A PMU counter is enabled and programmed to count event 0x0C. That is: instruction architecturally executed, condition code check pass, and software change of the PC.
2. An SVC, HVC, or SMC instruction is executed. If the above conditions are met, the PMUv2 architecture specifies that the counter for event 0x0C increments. However, the counter does not increment.

Or

1. One of the following instructions is executed:
   — MOV PC, LR
   — ThumbEE LDMIA R9!, {?, PC}
   — ThumbEE LDR PC, [R9], #offset
   — BX Rm, where Rm != R14
   — LDM SP, {?, PC}

If the above conditions are met, the PMUv2 architecture specifies that the counter for event 0x0E increments for (a), (b), (c) and does not increment for (d) and (e). However, the counter does not increment for (a), (b), (c) and increments for (d) and (e).

### Projected Impact:

The information returned by PMU counters that are programmed to count events 0x07, 0x0C, or 0x0E might be misleading when debugging or profiling code is executed on the processor.

## Workarounds:

Not available

## Proposed Solution:

No fix scheduled

## Linux BSP Status:

No software workaround available

## ERR008960 Arm/MP: 805420—C PMU event counter 0x14 does not increment correctly

### Description:

The Cortex-A7 MPCore processor implements version 2 of the Performance Monitor Unit architecture (PMUv2). The PMU can gather statistics on the operation of the processor and memory system during runtime. This event information can be used when debugging or profiling code. When a PMU counter is programmed to count L1 instruction cache accesses (event 0x14), the counter should increment on all L1 instruction cache accesses. Because of this erratum, the counter increments on cache hits but not on cache misses.

### Conditions:

1. A PMU counter is enabled and programmed to count L1 instruction cache accesses (event 0x14).
2. Cacheable instruction fetches miss in the L1 instruction cache.

When the above conditions are met, the event counter will not increment.

### Projected Impact:

A PMU counter that is programmed to count L1 instruction cache accesses will count instruction cache hits but not instruction cache misses.

The information returned can be misleading when debugging or profiling code executed on the processor.

Cache-bound code execution is not affected by this erratum because of the absence of cache misses.

### Workarounds:

To obtain a better approximation for the number of L1 instruction cache accesses, enable a second PMU counter and program it to count instruction fetches that cause linefills (event 0x01). Add the value returned by this counter to the value returned by the L1 instruction access counter (event 0x14). The result of the addition is a better indication of the number of L1 instruction cache accesses.

### Proposed Solution:

No fix scheduled

### Linux BSP Status:

No software workaround available

## ERR008961 Arm/MP: 804069—C Exception mask bits are cleared when an exception is taken in Hyp Mode

### Description:

The Cortex-A7 MPCore processor implements the Arm Virtualization Extensions and the Arm Security Extensions.

Exceptions can be routed to Monitor mode by setting SCR.{EA, FIQ, IRQ} to 1. Exceptions can be masked by setting corresponding bit CPSR.{A, I, F} to 1.

The Armv7-A architecture states that an exception taken in Hyp mode does not change the value of the mask bits for exceptions routed to Monitor mode. However, because of this erratum, the corresponding mask bits will be cleared to 0.

### Conditions:

1. One or more exception types are routed to Monitor mode by setting one or more of SCR.{EA, FIQ, IRQ} to 1.
2. The corresponding exception types are masked by setting the corresponding CPSR.{A, F, I} bits to 1.
3. Any exception is taken in Hyp mode.

If the above conditions are met then the exception mask bit CPSR.{A, F, I} is cleared to 0 for each exception type that meets conditions (1) and (2). The affected mask bits are cleared together regardless of the exception type in condition (3).

### Projected Impact:

If SCR.{AW, FW} is set to 0 then the clearing of corresponding bit CPSR.{A, F} to 0 has no effect. The value of CPSR.{A, F} is ignored.

Otherwise, when CPSR.{A, F, I} is set to 1, secure code cannot rely on CPSR.{A, F, I} remaining set to 1. An exception that should be masked might be routed to Monitor mode.

This is category C as it is expected that users will:

1. Set SCR.{AW, FW} to 0 when SCR.{EA, FIQ} is set to 1.
2. Set SCR.IRQ to 0.

### Workarounds:

Not available

### Proposed Solution:

No fix scheduled

### Linux BSP Status:

No software workaround available

## ERR007265 CCM: When improper low-power sequence is used, the SoC enters low power mode before the Arm core executes WFI

### Description:

When software tries to enter Low-Power mode with the following sequence, the SoC enters Low-Power mode before the Arm core executes the WFI instruction:

1. Set CCM_CLPCR[1:0] to 2'b00.
2. Arm core enters WFI.
3. Arm core wakes up from an interrupt event, which is masked by GPC or not visible to GPC, such as an interrupt from a local timer.
4. Set CCM_CLPCR[1:0] to 2'b01 or 2'b10.
5. Arm core executes WFI.

Before the last step, the SoC enters WAIT mode if CCM_CLPCR[1:0] is set to 2'b01, or STOP mode if CCM_CLPCR[1:0] is set to 2'b10.

### Projected Impact:

This issue can lead to errors ranging from module underrun errors to system hangs, depending on the specific use case.

### Workarounds:

Software workaround:

1. Software should trigger IRQ #32 (IOMUX) to be always pending by setting IOMUX_GPR1_GINT.
2. Software should then unmask IRQ #32 in GPC before setting CCM Low-Power mode.
3. Software should mask IRQ #32 right after CCM Low-Power mode is set (set bits 0-1 of CCM_CLPCR).

### Proposed Solution:

No fix scheduled

### Linux BSP Status:

Software workaround has been implemented in Linux BSP starting in release imx_3.14.38_6ul_ga.

## ERR009455 Clock: 24 MHz Oscillator does not start up

### Description:

The integrated oscillator with external 24 MHz crystal does not start up after chip power up.

### Projected Impact:

If there is no 24 MHz clock source available, the CPU will stay in reset after power up.

### Workarounds:

1. Connect 18 pF between XTALI pin and Ground.
2. Connect XTALO pin to an external 24 MHz crystal oscillator with 1k Ohm resistor, connect 499 Ohm resistor between XTALO pin and Ground.
3. The power supply of external crystal oscillator is VDD_HIGH_IN.
4. For low power applications, the output enable signal of the external crystal oscillator is controlled by PMIC_STBY_REQ signal through proper circuit (The NXP MCIMX6UL EVK board design is used as reference).

### Proposed Solution:

Fixed in silicon revision 1.1.

### Linux BSP Status:

No software workaround available

## ERR009606 eCSPI: In master mode, burst lengths of 32n + 1 will transmit incorrect data

### Description:

When the eCSPI is configured in master mode and the burst length is configured to a value 32n + 1 (where n = 0, 1, 2, …), the eCSPI will transmit the portions of the first word in the FIFO twice. For example, if the transmit FIFO is loaded with:

[0] 0x00000001

[1] 0xAAAAAAAA

And the burst length is configured for 33 bits (ECSPIx_CONREG[BURST_LENGTH] = 0x020), the eCSPI will transmit the first bit of word [0] followed by the entire word [0], then transmit the data as expected.

The transmitted sequence in this example will be:

[0] 0x00000001

[1] 0x00000001

[2] 0x00000000

[3] 0xAAAAAAAA

### Projected Impact:

Incorrect data transmission.

### Workarounds:

Do not use burst length of 32n + 1 (where n = 0, 1, 2, ...).

### Proposed Solution:

No fix scheduled.

### Linux BSP Status:

Software workaround not implemented in Linux BSP. Functionality or mode of operation in which the erratum may manifest itself is not used. The driver limits the burst length up to 32 bits.

## ERR009535 eCSPI: Burst completion by SS signal in slave mode is not functional

### Description:

According to the eCSPI specifications, when eCSPI is set to operate in the Slave mode (CHANNEL_MODE[x] = 0), the SS_CTL[x] bit controls the behavior of burst completion.

In the Slave mode, the SS_CTL bit should control the behavior of SPI burst completion as follows:

- 0—SPI burst completed when (BURST_LENGTH + 1) bits are received.
- 1—SPI burst completed when the SS input is negated.

Also, in BURST_LENGTH definition, it is stated "In the Slave mode, this field takes effect in SPI transfer only when SS_CTL is cleared."

However, the mode SS_CTL[x] = 1 is not functional in Slave mode. Currently, BURST_LENGTH always defines the burst length.

According to the SPI protocol, negation of SSB always causes completion of the burst. However, due to the above issue, the data is not sampled correctly in RxFIFO when {BURST_LENGTH+1}mod32 is not equal to {actual burst length}mod32.

Therefore, setting the BURST_LENGTH parameter to a value greater than the actual burst does not resolve the issue.

### Projected Impact:

Slave mode with unspecified burst length cannot be supported due to this issue. The burst length should always be specified with the BURST_LENGTH parameter and the SS_CTL[x] should be set to zero.

### Workarounds:

There is no workaround except for not using the SS_CTL[x] = 1 option in the Slave mode. The accurate burst length should always be specified using the BURST_LENGTH parameter.

### Proposed Solution:

No fix scheduled

### Linux BSP Status:

Software workaround not implemented in Linux BSP. Functionality or mode of operation in which the erratum may manifest itself is not used.

## ERR004446 EIM: AUS mode is nonfunctional for devices larger than 32 MB

### Description:

When the AUS bit is set, the address lines of the EIM are unshifted. By default, the AUS bit is cleared and address lines are shifted according to port size (8, 16 or 32 bits). Due to an error, the address bits 27:24 are shifted when AUS = 1. For example, CPU address 0xBD00_0000 ([A27:20]=1101 0000 becomes 0xB600_0000 ([A27:20]=0110 0000) on the EIM bus, because A[27:25] is shifted to [A26:24] and A[23:0] is not shifted. As a result A[24] is missed.

### Projected Impact:

If the memory used does not exceed 32 MB, there is no impact.

This mode is related to a unique memory configuration that is not often used. Most systems can work in the default mode (AUS = 0). Board designers should connect the EIM address bus without a shift (For example, A0A0 and A1A1), while working in AUS = 0 mode.

### Workarounds:

- Use the AUS = 0 mode (default) while connecting the address signals without a shift (for example, A0A0 and A1A1).
- For AUS = 1, for devices larger than 32 MB, it is necessary to build a memory map that take this shifting into consideration and does not include A[24] line.

### Proposed Solution:

No fix scheduled

### Linux BSP Status:

Software workaround not implemented in Linux BSP. Functionality or mode of operation in which the erratum may manifest itself is not used.

## ERR005829 FlexCAN: FlexCAN does not transmit a message that is enabled to be transmitted in a specific moment during the arbitration process

### Description:

FlexCAN does not transmit a message that is enabled to be transmitted in a specific moment during the arbitration process. The following conditions are necessary for the issue to occur:

- Only one message buffer is configured to be transmitted.
- The write which enables the message buffer to be transmitted (write on Control/Status word) happens during a specific clock during the arbitration process.
- After this arbitration process occurs, the bus goes to the Idle state and no new message is received on the bus.

For example:

1. Message buffer 13 is deactivated on RxIntermission (write 0x0 to the CODE field from the Control/Status word) [First write to CODE]
2. Reconfigure the ID and data fields
3. Enable the message buffer 13 to be transmitted on BusIdle (write 0xC on CODE field) [Second write to CODE]
4. CAN bus keeps in Idle state
5. No write on the Control/Status from any message buffer happens.

During the second write to CODE (step 3), the write must happen one clock before the current message buffer 13 to be scanned by arbitration process. In this case, it does not detect the new code (0xC) and no new arbitration is scheduled.

The problem can be detected 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:

- Any message buffer (either Tx or Rx) is reconfigured (by writing to its CS field) just after the Intermission field.
- There are other configured message buffers to be transmitted.
- A new incoming message sent by any external node starts just after the Intermission field.

### Projected Impact:

FlexCAN does not transmit a message that is enabled to be transmitted in a specific moment.

### Workarounds:

To transmit a CAN frame, the CPU must prepare a message buffer for transmission by executing the following standard 5-step procedure:

1. Check if the respective interrupt bit is set and clear it.

2. If the message buffer is active (transmission pending), write the ABORT code (0b1001) to the CODE field of the Control/Status word to request an abortion of the transmission. Wait for the corresponding IFLAG to be asserted by polling the IFLAG register or by the interrupt request if enabled by the respective IMASK. Then read back the CODE field to check if the transmission was aborted or transmitted. If backwards compatibility is desired (MCR[AEN] bit negated), just write the INACTIVE code (0b1000) to the CODE field to inactivate the message buffer, but then the pending frame might be transmitted without notification.
3. Write the ID word.
4. Write the data bytes.
5. Write the DLC, Control, and CODE fields of the Control/Status word to activate the message buffer.
6. The workaround consists of executing two extra steps:
7. Reserve the first valid mailbox as an inactive mailbox (CODE = 0b1000). If RX FIFO is disabled, this mailbox must be message buffer 0. Otherwise, the first valid mailbox can be found using the “RX FIFO filters” table in the FlexCAN chapter of the chip reference manual.
8. Write twice INACTIVE code (0b1000) into the first valid mailbox.

**NOTE**

The first mailbox cannot be used for reception or transmission process.

### Proposed Solution:

No fix scheduled

### Linux BSP Status:

Workaround possible but not implemented in the BSP, impacting functionality as described above.

## ERR007805 I2C: When the I2C clock speed is configured for 400 kHz, the SCL low period violates the I2C specification

### Description:

When the I2C module is programmed to operate at the maximum clock speed of 400 kHz (as defined by the I2C spec), the SCL clock low period violates the I2C spec of 1.3 μs min. The user needs to reduce the clock speed to get the SCL low time to meet the 1.3 μs I2C minimum required. This behavior means the SoC is not compliant with the I2C spec at 400 kHz.

### Projected Impact:

No failures have been observed when operating at 400 kHz. This erratum only represents a violation of the I2C specification for the SCL low period.

### Workarounds:

In order to exactly meet the clock low period requirement at fast speed mode, SCL must be configured to 384 kHz or less.

The following clock configuration meets the I2C specification requirements for SCL low for i.MX6 products:

I2C parent clock PERCLK_ROOT = 24M OSC

perclk_podf = 1

PERCLK_ROOT = 24M OSC/perclk_podf = 24 MHz

I2C_IFDR = 0x2A

I2C clock frequency = 24 MHz/64 = 375 kHz

### Proposed Solution:

No fix scheduled

### Linux BSP Status:

Software workaround not implemented in Linux BSP. Functionality or mode of operation in which the erratum may manifest itself is not used. The BSP configures the I2C frequency to 375 kHz by default.

## ERR005778 MMDC: DDR Controller’s measure unit may return an incorrect value when operating below 100 MHz

### Description:

The measure unit counts cycles of an internal ring oscillator. The measure unit readout is used to fine tune the delay lines for temperature/voltage changes for both DDR3 and LPDDR2 interfaces. When operating at low frequencies (below 100 MHz), the measure unit counter might overflow due to an issue in the overflow protection logic. As a result, an incorrect measure value will be read.

### Projected Impact:

This might cause a rare issue if the measure unit counter stops within a small range of values that translate to a delay that tunes the system incorrectly. This issue might not manifest in the application because it is dependent on a combination of DDR frequencies coupled with specific Process, Voltage, and Temperature conditions.

### Workarounds:

To workaround this issue, following steps should be performed by software:

1. Prior to reducing the DDR frequency (400 MHz), read the measure unit count bits (MU_UNIT_DEL_NUM).
2. Bypass the automatic measure unit when below 100 MHz, by setting the measure unit bypass enable bit (MU_BYP_EN).
3. Double the measure unit count value read in step 1 and program it in the measure unit bypass bit (MU_BYP_VAL) of the MMDC PHY Measure Unit Register, for the reduced frequency operation below 100 MHz.

Software should re-enable the measure unit when operating at the higher frequencies, by clearing the measure unit bypass enable bit (MU_BYP_EN). This code should be executed out of Internal RAM or a non-DDR based external memory.

### Proposed Solution:

No fix scheduled

### Linux BSP Status:

Software workaround has been implemented in Linux BSP starting in release imx_3.14.38_6ul_ga.

## ERR009596 MMDC: ARCR_GUARD bits of MMDC Core AXI Re-ordering Control register (MMDC_MAARCR) does not behave as expected

### Description:

The ARCR_GUARD bits of MMDC Core AXI Re-ordering Control register (MMDC_MAARCR) are used to ensure better DDR utilization while preventing starvation of lower priority transactions. After reordering is performed on previous read/write DDR transactions, the specific outstanding transaction will first obtain the maximum score in “dynamic score mode” and then wait for additional ARCR_GUARD count before achieving the highest priority. Due to a design issue, the ARCR_GUARD counter does not count up to the pre-defined value in the ARCR_GUARD bit field as expected. Therefore, the aging scheme optimizes the transaction reordering only up to the default aging level (15) and assigns a highest priority tag to the outstanding transaction.

### Projected Impact:

The aging scheme optimizes the transaction reordering only up to the default aging level (15). No functional issues have been observed with an incorrect setting.

### Workarounds:

Software should always program the ARCR_GUARD bits as 4'b0000. That means the accesses which have gained the maximum dynamic score will always become the highest priority after achieving the default highest aging level (15).

### Proposed Solution:

No fix scheduled

### Linux BSP Status:

Software workaround not implemented in Linux BSP. Functionality or mode of operation in which the erratum may manifest itself is not used.The NXP Linux BSP releases leave the ARCR_GUARD bits at the default value of 4'b0000.

## ERR009541 PXP: PXP CSC2 cannot do RGB2YCbCr or RGB2YUV conversion correctly

### Description:

When doing the RGB to YUV conversion, PXP CSC2 design can only output results from 0 to 255, which cannot meet the UV's range requirements, the Y output is correct. For RGB2YCbCr, the coef_d1/d2/d3 has not enough bit width, thus YCbCr value is not correct.

### Conditions:

This problem occurs when choosing PXP CSC2 to do RGB2YUV or RGB2YCbCr conversion.

### Projected Impact:

PXP output can only be Y8 format when CSC2 is used for RGB2YUV conversion. CSC2 cannot be used for RGB2YCbCr conversion.

### Workarounds:

Use SW to do the RGB2YUV or RGB2YCbCr conversion.

### Proposed Solution:

No fix scheduled

### Linux BSP Status:

BSP uses software to do such conversion.

## ERR011115 System Boot: Boot failures on certain devices when the boot image targets OCRAM

### Description:

A boot image corruption can result in a boot failure for certain boot devices under specific conditions as described below. The boot failure only occurs if all conditions below are satisfied.

### Conditions:

There are four specific conditions resulting in this boot failure.

- i.MX 6UltraLite silicon revision 1.2 device shipped prior to date code 1740
- The boot device is SD/eMMC/SPINOR
- Boot image runs from internal memory—On Chip RAM (OCRAM) space
- Start address of the boot image in OCRAM is between 0x907000 and 0x908000

### Projected Impact:

Boot image corruption leading to boot failure and possible entry into Serial Downloader mode.

### Workarounds:

User should set the boot image start address greater or equal to 0x908000 (if the boot image running in OCRAM) to prevent the boot failure when using the SD/eMMC/SPINOR boot devices. The impact of this workaround is a reduction in the usable memory region by 4 KB. Alternatively, users can use a boot image load address in the external DDR memory instead of the internal OCRAM.

### Proposed Solution:

Fixed in i.MX 6UltraLite silicon revision 1.2 devices shipped after date code 1740.

### Linux BSP Status:

Workaround not implemented in BSP; No impact to NXP BSP users, since the default BSP boot image uses external DDR as the load address and not internal OCRAM.

## ERR011112 System Boot: QSPI boot failure when the boot image targets OCRAM

### Description:

A boot image corruption can result in a boot failure for a QSPI boot device under specific conditions as described below. The boot failure only occurs if all conditions below are satisfied.

### Conditions:

There are four specific conditions that result in this boot failure:

1. i.MX 6UltraLite silicon revision 1.2
1. A QSPI boot device is used
2. The device is a security enabled configuration (SEC_CONFIG[1] eFUSE is programmed)
3. The boot image start address (specified in boot data) targets internal memory on Chip RAM (OCRAM) space

### Projected Impact:

QSPI boot failure and possible entry into Serial Downloader mode.

### Workarounds:

Since the boot failure only occurs when the boot image start address targets OCRAM space, the recommended workarounds are for users to ensure the QSPI boot image runs from DDR or executes in place (XIP).

### Proposed Solution:

No fix scheduled

### Linux BSP Status:

Workaround not implemented in BSP. Functionality where the erratum may manifest itself is not used.

## ERR010690 SNVS: SNVS_LP Registers Reset Issue

### Description:

SNVS_LP registers may be reset during system power down or power up.

### Conditions:

Reset of the SNVS_LP registers may happen when either one of below conditions are met:

- VDD_SOC_IN is ramping down earlier than VDD_HIGH_IN.
- VDD_SOC_IN is ramping up when VDD_HIGH_IN is higher than 3.1 V.

### Projected Impact:

All registers inside SNVS_LP may be reset.

### Workarounds:

The workarounds below should be both used to avoid SNVS_LP registers reset issue:

- VDD_HIGH_IN power down is earlier than VDD_SOC_IN.
- VDD_HIGH_IN voltage power is less than or equal to 3.0 V.

### Proposed Solution:

Fixed in silicon revision 1.2.

### Linux BSP Status:

N/A

## ERR009454 Power Management: PMIC_STBY_REQ output voltage lower than specification

### Description:

When PMIC_STBY_REQ pad is driving high state in SUSPEND mode, the output voltage level is about 2.0 V. This voltage is lower than the 3.0 V VDD_SNVS_IN supply voltage.

### Projected Impact:

This signal is used to notify external PMIC chip or discrete DC-DC/LDO power circuit moving into low power mode. Since the required voltage is 3.0 V (VDD_SNVS_IN), the PMIC or discrete power circuit may not be functional as expected with 2.0 V input.

### Workarounds:

1. Make sure the PMIC_STBY_REQ pad is set to open-drain mode with 100 kOhm pull-up.
2. For low power applications, use external circuit to cover PMIC_STBY_REQ output from 2.0 V to 3.0 V for external PMIC, or change discrete power circuit to allow 2.0 V input (The NXP MCIMX6UL-EVK board design can be used as reference).

### Proposed Solution:

Fixed in silicon revision 1.1.

### Linux BSP Status:

BSP sets PMIC_STBY_REQ as open-drain mode with 100 kOhm pull-up.

## ERR004535 USB: USB suspend and resume flow clarifications

### Description:

In device mode, The PHY can be put into Low Power Suspend when the device is not running or the host has signaled suspend. The PHY Low power suspend bit (PORTSC1.PHCD) will be cleared automatically when the host initials resume. Before forcing a resume from the device, the device controller driver must clear this bit. In host mode, the PHY can be put into Low Power Suspend when the downstream device has been placed into suspend mode (PORTSC1.SUSP) or when no downstream device is connected. Low power suspend is completely under the control of software.

To place the PHY into Low power mode, software needs to set PORTSC1.PHCD bit, set all bits in USBPHY_PWD register and set the USBPHY_CTRL.CLKGATE bit.

When a remote wakeup occurs after the Suspend (SUSP) bit is set while the PHY Low power suspend bit (PHCD) is cleared, a USB interrupt (USBSTS.PCI) will be generated. In this case, the PHCD bit will NOT be set because of the interrupt. However, if a remote wakeup occurs after the PHCD bit is set while the USB PHY Power-Down Register (USBPHY_PWD) and the UTMI clock gate (USBPHY_CTRL.CLKGATE) bit is cleared, a remote wakeup interrupt will be generated. In this case, all the bits in the HW_USBPHY_PWD register and the USBPHY_CTRL.CLKGATE bit will be set, even after the remote wakeup interrupt is generated, which is incorrect.

### Projected Impact:

Resume error, if the correct flow is not adhered to.

### Workarounds:

To place the USB PHY into low power suspend mode, the following sequence should be performed in an atomic operation (interrupts should be disabled during these three steps):

1. Set the PORTSC1.PHCD bit
2. Set all bits in the USBPHY_PWD register
3. Set the USBPHY_CTRL.CLKGATE bit

### Proposed Solution:

No fix scheduled

### Linux BSP Status:

Software workaround integrated in Linux BSP codebase starting in release imx_3.0.35_4.1.0.

## ERR006281 USB: Incorrect DP/DN state when only VBUS is applied

### Description:

When VBUS is applied without any other supplies, incorrect communication states are possible on the data (DP/DN) signals. If VDDHIGH_IN is supplied, the problem is removed.

### Projected Impact:

This issue primarily impacts applications using charger detection to signal power modes to a PMIC in an undercharged battery scenario where the standard USB current allotment is not sufficient to boot the system.

### Workarounds:

Apply VDDHIGH_IN if battery charge detection is needed. Otherwise, disable charger detection by setting the EN_B bit in USB_ANALOG_USBx_CHRG_DETECTn to 1.

### Proposed Solution:

No fix scheduled

### Linux BSP Status:

Software workaround not implemented in Linux BSP. Functionality or mode of operation in which the erratum may manifest itself is not used.

## ERR010661 USB: VBUS leakage occurs if USBOTG1 VBUS is on and USBOTG2 VBUS transitions from on to off

### Description:

When two USB ports work as OTG or device simultaneously. One VBUS (selected by PMU_REG_3P0.vbus_sel bit) voltage will not drop after cable unplug, causing the port to fail to detect the cable detach. If these two ports do not need to support detach detection, simultaneously using two OTGs or devices can be supported.

### Conditions:

When two USB ports work as OTGs or devices simultaneously.

### Projected Impact:

Do not use two OTGs or devices simultaneously. Only four scenarios are supported:

- One for OTG/Device, another for Host.
- One for OTG/Device, another is un-used.
- One for Host, another for Host.
- One for Host, another is un-used.

### Workarounds:

Only one port can be used as OTG or device. The other port must be used as host. Set the PMU_REG_3P0.vbus_sel bit to select the host port.

### Proposed Solution:

No fix scheduled

### Linux BSP Status:

No software workaround available

*How to Reach Us:*

**Home Page:**
nxp.com

**Web Support:**
nxp.com/support

Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including “typicals” must be validated for each customer application by customer, customer’s technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: nxp.com/SalesTermsandConditions.



Document Number: IMX6ULCE
Rev. 2.3
09/2017