

## Mask Set Errata for Mask 0N95W

This report applies to mask 0N95W for these products:

- Contact your NXP representative for orderable part number information.

**Table 1. Errata and Information Summary**

Erratum ID	Erratum Title
e50061	ANALOG: DPLL loses lock under corner conditions
e50055	ANALOG: DRC temperature sensor causes coupling errors
e50068	AUDIO: Incorrect 24 MHz clock source at Audio Clock Mux input
e50059	DC: Display write back function is not functional
e50060	DC: PRG on the fly bypass switch issue
e50125	DRAM: Controller automatic derating logic may not work as intended when the LPDDR4 memory temperature is above 85C at initialization
e10947	DRAM: DQS/DQSN glitch suppression resistors must be enabled during read-levelling
e50054	DRAM: Extra boot time is required with DDR3L ECC
e10944	DRAM: In LPDDR4 mode, tMPCWR timing violation in incremental DQS2DQ Training
e10946	DRAM: In LPDDR4 mode: REFRESH must be disabled during DQS2DQ training
e10945	DRAM: PUB does not program LPDDR4 DRAM MR22 prior to running DRAM ZQ calibration
e10948	DRAM: Timing Violation from Read/Write to MRW in LPDDR4 mode
e11543	FlexCAN: Nominal Phase SJW incorrectly applied at CRC Delimiter
e50057	GPU: OpenCV and Vulkan conformance issue
e50067	ISI: Adjacent processing pipelines within the ISI sub-system can experience loss of data
e50066	ISI: Data overflows occur when input streams exceed AXI transaction frequency
e50058	ISI: Incomplete frames when using virtual channels 1, 2, 3
e50135	JPEG DECODER: multi-frame jpeg bitstream may not be correctly decoded when there is a small size frame inside
e11439	MIPI DSI: Checksum is incorrect for DCS command long packet writes with zero-length data payload
e10930	PCIE: EOM single point sample error/valid result is not correct
e11193	PCIE: EP, PM_PME: L1 Exit Does Not Occur when PME Service Timeout Mechanism Expires
e11194	PCIE: Plesiochronous loopback is not functional in PCIe Gen3

*Table continues on the next page...*



**Table 1. Errata and Information Summary (continued)**

Erratum ID	Erratum Title
e50108	ROM: eMMC/SD boot failure due to ROM code timeout under certain conditions
e50052	ROM: NAND boot fails when image header points to an unprogrammed block
e50053	ROM: USB HID device cannot be re-enumerated successfully after an unplug/plug USB cable operation
e50056	SNVS: LDO startup is too slow
e50141	USB2: Endpoint conflict issue in device mode
e50147	USB3: Multiple DMA write transfer complete interrupts are generated before final write access handshake to the AXI bus
e50115	USB3: Port Configuration Response is not compliant with the USB compliance TD 7.17 test case
e50148	USB3: Race condition possible during software update to TRB in the system memory and DMA reads of same TRB
e50149	USB3: TRB OUT endpoints transfer blockage and performance delays

**Table 2. Revision History**

Revision	Changes
0, 05/2019	Initial revision

**e50061: ANALOG: DPLL loses lock under corner conditions**

**Description:** At cold temperatures, the DPLL does not lock for some parts at the extreme edge of the supply voltage tolerance.

**Workaround:** The DPLL is sensitive to noise on its VDD supply, in general this is VDD\_MAIN, particularly at low temperature ( < -20 degrees Celsius).

Numerous SCU related workarounds have been applied to reduce this sensitivity, however the issue remains.

The recommended workaround is close monitoring of the decoupling capacitors used on the power supply design, to ensure close proximity and a low impedance path to the regulator, as defined in the Hardware Development Guide (HDG).

Validation must include board testing at low temperature to check for any power supply sensitivity concerns.

**e50055: ANALOG: DRC temperature sensor causes coupling errors**

**Description:** DRC temperature sensor turning on and off in its normal operation couples and corrupts other analog blocks through top level metal coupling of the SoC. This causes symptoms such as variation in the DDR clock.

**Workaround:** Disable DRC temperature sensor and only use the SCU temperature sensor. Little variation is seen between the readings of these 2 temperature sensors. Margin for Overheat shutdown is still sufficient to avoid damage.

### **e50068: AUDIO: Incorrect 24 MHz clock source at Audio Clock Mux input**

**Description:** The DSC of the Audio subsystem provides two 24 MHz clock sources.

One comes directly from the 24 MHz oscillator (“24\_MHz\_functional”) and the other goes through SW gating used during the Reset sequence (“24\_MHz\_reset”).

This second 24 MHz source is currently connected to the Audio Clock Mux (ACM) input and can be selected as the functional clock for the Audio GP-Timers. As this “24\_MHz\_reset” clock is should be enabled and disabled according to requested reset sequences, it must not be used as functional clock for Audio modules. The “24\_MHz\_functional” must be used.

Only the GP-Timers are impacted by this incorrect connection.

**Workaround:** Do NOT select the 24 MHz clock source from the ACM’s (audio clock mux) GPT0..5 external clock generator. Instead, select the 24 MHz clock source available from the GPT’s internal clock multiplexer, which comes from the correct source. That is to say, within the GPT Control Register (CR), CLKSRC[8:6] must use “101b - Crystal oscillator as the Reference Clock (ipg\_clk\_24M)”, and avoid use of the “011b - External Clock”.

### **e50059: DC: Display write back function is not functional**

**Description:** The mechanism to enable an output stream to a display and also to be copied into memory using one of the internal camera streams is not functional due to the pixel link receiver address being incorrectly tied-off.

**Workaround:** The display controller signature unit (CRC) can be used to check for display changes.

### **e50060: DC: PRG on the fly bypass switch issue**

**Description:** When the display controller switches the DPR/PRG from bypass to non-bypass on the fly, it causes a sync error. A screen artifact (3-4 lines of the overlay) can be seen at the top of the overlay. The bypass to non-bypass on the fly switch can occur when the overlay pixel format changes from DPR/PRG unsupported to supported.

**Workaround:** Careful timing of the overlay change can hide this problem. Following the sequence “overlay OFF – 1 frame – overlay ON” can also hide the problem. Because this workaround requires a deterministic handling of interrupts, a non-realtime OS, such as Linux, cannot guarantee the timing of the overlay change.

## **e50125: DRAM: Controller automatic derating logic may not work as intended when the LPDDR4 memory temperature is above 85C at initialization**

**Description:** LPDDR4 memories require periodic refreshes to maintain memory contents. Per the JEDEC specification JESD209-4 the memory refresh rate needs to increase and timings de-rated as the memory operational temperature exceeds vendor defined temperature thresholds. The LPDDR4 Mode Register 4 (MR4) contains temperature/refresh rate information and a Temperature Update Flag (TUF).

An issue exists with the automatic derating logic of the DDR controller that only samples the LPDDR4 MR4 register when the Temperature Update Flag (TUF) field (MR4[7] ) is 1'b1. If the LPDDR4 memory is initialized and starts operation above 85C (MR4[2:0] > 3'b011) , the MR4 Temperature Update Flag (TUF) will not set. The DDR Controller will therefore not automatically adjust the memory refresh rate or de-rate memory timings based on the LPDDR4 memory temperature. This may result in the controller incorrectly setting the refresh period and de-rated memory timing parameters (tRCD, tRAS, tRP, tRRD), potentially causing the LPDDR4 memory losing data contents and leading to possible data integrity issues above 85C. The actual case surface temperature on the center-top side of the LPDDR4 memory that is used to trigger the MR4 register may vary depending on memory vendors.

If the LPDDR4 memory temperature remains below 85C at initialization, then the derating logic works as intended, automatically adjusting the memory refresh period and memory timing during the entire system operation. This erratum does not impact other SoC supported DDR memory interfaces.

**Workaround:** The software workaround is to check the memory temperature at initialization, via the MR4 register and determine if it is above 85°C. If the temperature is below 85°C, then the automatic temperature derating logic should be left enabled (default setting), otherwise the derating logic should be disabled and software should manually adjust the memory refresh rate and memory timing parameters (tRCD, tRAS, tRP, tRRD). Once the memory temperature is below 85C (MR4[2:0] == 3'b011), software should re-adjust the memory refresh rate and memory timings to nominal settings and then reenble the automatic derating logic.

The software workaround will be integrated into the next BSP GA release (imx\_4.14.98\_2.0.0\_p1 (SCFW v1.2.1)).

## **e10947: DRAM: DQS/DQSN glitch suppression resistors must be enabled during read-levelling**

**Description:** By default DQS/DQSN glitch suppression registers are disabled. When external DQS/DQSn are not driven to valid differential states, the DQS cell's core-side outputs become unknown. This causes errors in the read-levelling gate training.

**Workaround:** Enable the strongest 355 ohm pull resistors during gate training. Scripts provided by NXP's DRAM RPA (Register Programming Aid) implement the required workaround.

## **e50054: DRAM: Extra boot time is required with DDR3L ECC**

**Description:** For DDR3L with ECC, due to design limitations and ECC scrub requirements, the SCU must initialize the entire memory for ECC before beginning to load the M4 image. This requires up to 300 msec extra.

**Workaround:** The future planned fix will initialize only a small region of RAM, reduce the impact on boot time.

**e10944: DRAM: In LPDDR4 mode, tMPCWR timing violation in incremental DQS2DQ Training**

**Description:** In LPDDR4 mode with incremental DQS2DQ Training enabled and speed grade > 2133 Mbps, the algorithm performs Power Down (PD) Entry-Exit Cycle to reset the MPC WR-RD FIFO pointers in the DRAM. The PUB sends the MPC WRFIFO command after waiting for tXP from PD Exit. However, JEDEC specification requires waiting an additional tMPCWR after tXP timing. This extra tMPCWR timing is not handled by the PUB Training algorithm, resulting in a violation of tMPCWR JEDEC parameter.

**Workaround:** Do not run incremental DQS2DQ Training of the PHY in LPDDR4 mode.

**e10946: DRAM: In LPDDR4 mode: REFRESH must be disabled during DQS2DQ training**

**Description:** If REFRESH is enabled during DQS2DQ training, a JEDEC Specification violation may occur. REFRESH must be disabled during DQS2DQ training, which is performed during initial power-up (cold boot).

**Workaround:** For initial power-up (cold boot), disable REFRESH during DQS2DQ training. For self-refresh (warm boot), do not run DQS2DQ training. Restore the saved register values prior to self-refresh entry. Scripts provided by NXP's DRAM RPA (Register Programming Aid) implement the required workaround.

**e10945: DRAM: PUB does not program LPDDR4 DRAM MR22 prior to running DRAM ZQ calibration**

**Description:** When the PHY Utility Block (PUB) initializes the DRAM, the DRAM MR22 is programmed after ZQ Calibration. This may result in incorrect ZQ calibration results on the LPDDR4 DRAM side, because MR22[2:0] works as the controller On Die Termination (ODT) replica during the Pull-Up calibration. Therefore the expected controller ODT must be programmed into MR22 prior to the DRAM ZQ calibration.

**Workaround:** Run DRAM Initialization twice. Scripts provided by NXP's DRAM RPA (Register Programming Aid) implement the required workaround.

**e10948: DRAM: Timing Violation from Read/Write to MRW in LPDDR4 mode**

**Description:** When software sends a MRW command in parallel with a Read/Write transaction, the Read/Write command can be followed by the MRW command, which can result in the following timing violations:

1. RD to MRW
2. RDA to MRW
3. WRA/MWRA to MRW

This can occur only in LPDDR4 mode. When the memory clock frequency is lower than 450 MHz, then one of above 3 violations may occur, when the memory clock frequency is NOT lower than 450 MHz, then above item 1 or item 2 violation may occur.

The above timing constraints were introduced in the LPDDR4 specification JESD209-4A.

**Workaround:** MRW commands sends in parallel with a Read/Write transaction must follow a specific sequence.

### **e11543: FlexCAN: Nominal Phase SJW incorrectly applied at CRC Delimiter**

**Description:** During the reception of a CAN-FD frame when the Bit Rate Switch (BRS) is enabled, the Synchronization Jump Width (SJW) for the CRC Delimiter bit is incorrectly defined by the Nominal Phase SJW. The CAN specification stipulates that the CRC Delimiter bit should have a SJW set by the Data Phase SJW.

When a resynchronization event is triggered for the CRC delimiter bit (recessive in correct operation), the sample point will be adjusted by an amount as defined by the Nominal Phase SJW rather than the specified Data Phase SJW. This may result in the incorrect detection of a dominant bit leading to a CAN error frame. However, as the CRC delimiter bit position will only apply the SJW upon the detection of an unexpected dominant bit on the CAN bus, an error frame is already likely. For the case the SJW is applied at the CRC delimiter and a recessive bit is not detected, the receiving node will issue an error frame.

The CAN protocol is designed to handle resynchronization errors and hence the CAN bus will recover from the insertion of the incorrect SJW at the CRC delimiter. Upon detecting the error frame the transmitting node will re-transmit the frame.

The following FlexCAN configurations are not affected:

- Classical CAN frames (CAN 2.0B)
- CAN FD frames with bit rate switch disabled (BRS = 0)
- CAN FD frames with Nominal Phase SJW equal to Data Phase SJW
- CAN FD transmissions

Configuration for the FlexCAN:

- Nominal Phase SJW is configured by the Resync Jump Width bit in the CAN Control Register 1 (CAN\_CTRL1[RJW]) or by the Extended Resync Jump Width bit in the CAN Bit Timing Register (CAN\_CBT[ERJW])
- Data Phase SJW is configured by the Fast Resync Jump Width bit in the CAN FD Bit Timing Register (CAN\_FDCBT[FRJW])

**Workaround:** The robustness of the CAN protocol ensures that the receiver automatically recovers from the application of the incorrect SJW. The CAN protocol is designed to recover from resynchronization errors and hence any frame that is not correctly received will be re-sent by the transmitting node.

### **e50057: GPU: OpenCV and Vulkan conformance issue**

**Description:** GPU may hang when running OpenCV or Vulkan conformance tests under corner conditions.

**Workaround:** Software workaround has been integrated into L4.14 BSP release and later release. This workaround has a small performance impact <1% during OpenCV or Vulkan tests.

### **e50067: ISI: Adjacent processing pipelines within the ISI sub-system can experience loss of data**

**Description:** Using adjacent channels where one channel's line ends when the next channel's line begins (common in virtual channel functionality) can cause the second channel to skip a line every 8 or 16 lines.

Using adjacent channels can also effect the width and format of the line by creating a final write that does not fill a 128 byte buffer.

**Workaround:** For virtual channel applications the pipeline order can be adjusted to avoid adjacent channel assignments, for example, VC 0, 1, 2, and 3 assigned to pipelines 0, 2, 1, 3.

### **e50066: ISI: Data overflows occur when input streams exceed AXI transaction frequency**

**Description:** The Image Sensing Interface (ISI) has a short elasticity buffer relative to the length of a line. The buffer can be as few as 85 pixels or as many as 512 pixels depending on the output format. Most RGB formats have 128 pixels. Because of the short buffer, if there is any delay in latency, then an overflow can occur. The possibility of overflow increases when the number of active channels increases.

In addition, memory reads and the last line of a scaling process consume data as fast as possible (instead of at the rate of the incoming pixel stream), therefore, the output buffer fills faster and requires even lower latency to process the data.

**Workaround:** The design target was intended to support up to a single 8 Mpixel (4K) stream at 30 fps, or multiple streams up to the equivalent data rate. However, combinations of sensors which add up to less than 2Mpixel are supported with current design. That's to say, if 1 sensor is used, 2Mpixels stream can be supported; if 2 sensors are used, 1Mpixels of each stream can be supported; and so on.

In the case of scaling, the last line of each frame must be cropped and discarded.

### **e50058: ISI: Incomplete frames when using virtual channels 1, 2, 3**

**Description:** Except for virtual channel 0, virtual channels 1, 2, and 3 do not have proper VSYNC timing from MIPI CSI2 when different cameras are multiplexed together. As a result, frames stored in memory can be corrupted due to missing last lines.

**Workaround:** Virtual channels 1, 2, and 3 do not work normally for multiplexed cameras. Only single camera operation is supported by the MIPI CSI2 interface.

### **e50135: JPEG DECODER: multi-frame jpeg bitstream may not be correctly decoded when there is a small size frame inside**

**Description:** When the JPEG decoded frame with a resolution that is no larger than 64x 64 and it is followed by a next decoded frame with a larger resolution, then this next decoded frame may be corrupted.

**Workaround:** The decoded image resolution should be larger than 64x 64.

### **e11439: MIPI DSI: Checksum is incorrect for DCS command long packet writes with zero-length data payload**

**Description:** According to the MIPI DSI specification, long packets are comprised of a Packet Header and a payload of 0 to  $2^{16}-1$  bytes. For the special case of a zero-length payload, the specification requires the checksum must be set to 0xFFFF.

The MIPI DSI controller produces an incorrect checksum for DCS commands issued via long packets with zero-length payloads in DSI Low-Power mode (LP). There is no issue for similar commands issued in DSI High-Power mode (HP).

This issue should not affect normal application operation because packets with zero data length will normally be sent using the short packet format. However, because the MIPI DSI spec specifically states this behavior, MIPI DSI certification will fail with long packets of zero-length.

**Workaround:** Use short packet format to send DCS commands with zero length data payloads.

### **e10930: PCIE: EOM single point sample error/valid result is not correct**

**Description:** There is an eye monitor in the SerDes analysis which can monitor the following:

- a. Error and valid bits of a certain duration
- b. Eye width
- c. Eye height
- d. Eye area

However, there is a design issue with item (a) causing incorrect error/valid bit results.

**Workaround:** Customers must not use the error/valid count results to check the eye quality. Instead, use the eye width, eye height, or eye area.

### **e11193: PCIE: EP, PM\_PME: L1 Exit Does Not Occur when PME Service Timeout Mechanism Expires**

**Description:** Impacted Configuration(s): Upstream Port configurations:

CC\_DEVICE\_TYPE =DM and device\_type =4'b0000

Defect Summary:

When a function issues a PM\_PME Message, it sets the PME\_Status bit. If the Downstream port has not cleared the PME\_Status bit within 100ms, a PME Service timeout occurs. At this point, the Upstream port must resend the PM\_PME message.

In the current implementation of the controller, the PME Service timeout does not trigger an exit from L1 to resend the PM\_PME message.

System Usage Scenario:



Upstream ports using a wake-up mechanism followed by a power management event (PME) message.

Consequence(s):

The defect has the following effect:

The PME service routine cannot make forward progress until the PM\_PME message is resent.

**Workaround:** Poll the PME\_Status bit after sending the PME message to exit L1 state. If this bit remains 1 for 100ms or more, SW must re-toggle bit 10 “PCIE\_CTRL\_APPS\_PME” of register “SRC\_PCIEPHY\_RCR” .

## **e11194: PCIE: Plesiochronous loopback is not functional in PCIe Gen3**

**Description:** Customers should be using mesochronous loopback when sending arbitrary bit streams.

Plesiochronous loopback: Is loopback from Rx back to Tx after the PCIe elastic buffer function in the PCS.

The intent of this reverse loopback scheme is to send arbitrary bit-streams through the elastic FIFO on the Rx side of the PCS and back through the Tx side of the PCS into the PMA. However, this does not work at Gen3 speed. This mode is not practical because the entire PCS PCIe pipeline is designed for protocol-dependent data, and requires many bypass paths to enable arbitrary bit streams through it. Moreover, there is no way to support elasticity when the bit-stream is protocol-agnostic, rendering the elastic FIFO useless.

Mesochronous (meso) loopback: Is loopback from Rx back to Tx before any elastic buffer, hence requiring 0ppm frequency difference between TxClk and RxClk, and requires TxClk and RxClk to be phase-adjusted using an automatic CDR skip-bit routine (as described in the PUG). Meso loopback assumes that the intersection set of the setup+hold margin for all 20 bits in the Rx to Tx STA path has a large open window. The SDC constraints were originally intended to contain max\_delay and min\_delay constraints to ensure this, but customers may not have optimized the window. Historically, mesochronous mode rarely worked at the highest protocol speeds due to this dependency on customer’s timing optimization.

**Workaround:** Customers must use meso loopback when sending arbitrary bit streams.

## **e50108: ROM: eMMC/SD boot failure due to ROM code timeout under certain conditions**

**Description:** This issue is related to boot from eMMC or SD devices.

On power-up, a boot monitor timer is initialized. On a successful boot, SECO firmware is loaded and run from the boot device—that is, eMMC or SD, which, after loading and verifying SCU firmware (SCFW), disables the timer. When a successful boot requires more than 300 msecs, a timeout occurs that is considered a boot failure, and therefore generates a warm reset, which results in a looping boot failure.

Typically, the initialization time for most eMMC/SD devices is about 100 to 200 msecs, however, the eMMC/SD specification allows up to 1 second for this initialization time.

Any eMMC/SD device that exceeds the timeout to initialization will fail to boot. Sudden power loss or power cycle stress testing to eMMC/SD devices can cause data corruption, which can force the eMMC/SD device to run an internal data check on the next power up, which results in a longer initialization time, forcing a timeout and a looping boot failure.

**Workaround:** Generally reducing eMMC/SD initialization time under 300 msec is the most effective way to avoid this looping boot failure.

For the data corruption case, change the boot mode to serial download mode then load and run an image via USB. This image can initialize the eMMC/SD device and exit out of the internal data check state.

In future silicon releases, ROM will consider this case and wait 1 second to avoid the boot failure.

#### **e50052: ROM: NAND boot fails when image header points to an unprogrammed block**

**Description:** ROM first reads the Image Container Set 0 header, and then the Image Container Set 1 header, unless the secondary boot has been disabled by a fuse, in which case the Set 1 header will not be read. ROM will then select the header with the newest software version for primary boot and the oldest one for secondary boot.

For NAND boot, the block where the Image Container Set 0 is programmed is specified by fuses. If the specified block has been erased or has random data on it, the NAND read API returns a failure. In this case, ROM will attempt to read from the block with the Image Container Set 1 header programming, which is also specified by fuses.

However, there is a bug in the ROM code. It will not read data from the block with the Image Container Set 1 header, and instead continues to read data from the block with the Image Container Set 0 header. However, because this block has not been programmed boot failure occurs, even if there is a valid Image Container Set 1 header available.

**Workaround:** Avoid upgrading Image Container Set 0. Only upgrade Image Container Set 1 if the boot image needs upgraded. Per detailed description, ROM always tries to read both image container sets and selects the one with the newest software version to boot.

#### **e50053: ROM: USB HID device cannot be re-enumerated successfully after an unplug/plug USB cable operation**

**Description:** The USB HID device enumerates successfully on the Host side when booting from serial download mode. However, after disconnecting the USB cable and re-connecting the cable again, the USB HID device will not re-enumerate on Host side because ROM incorrectly resets the USB.

**Workaround:** Reset the device, or power down and re-power on the device.

#### **e50056: SNVS: LDO startup is too slow**

**Description:** SNVS startup on LDO is too slow at low temperature with VDD\_SNVS less than 2.8V. This issue may cause some parts to take longer to startup due to boot retry in SCU. The boot time to M4 operation may increase from <50ms to around 80ms because of the re-boot attempts.

**Workaround:** The workaround is to ensure that voltage is maintained at, or above, 2.8 volts at low temperature.

## **e50141: USB2: Endpoint conflict issue in device mode**

**Description:** An endpoint conflict occurs when the USB is working in device mode and an isochronous IN endpoint exists.

When the endpointA IN direction is an isochronous IN endpoint, and the host sends an IN token to endpointA on another device, then the OUT transaction may be missed regardless the OUT endpoint number. Generally, this occurs when the device is connected to the host through a hub and other devices are connected to the same hub.

The affected OUT endpoint can be either control, bulk, isochronous, or an interrupt endpoint.

After the OUT endpoint is primed, if an IN token to the same endpoint number on another device is received, then the OUT endpoint may be unprimed (Cannot be detected by SW), which causes this endpoint to no longer respond to the host OUT token, and thus, no corresponding interrupt occurs.

**Workaround:** Do not connect to a hub in the case when ISO IN endpoint(s) is used. When the hub(s) must be connected in this scenario, the endpoint number(s) of the ISO IN endpoint(s) should be different from the endpoint number(s) of any type of IN endpoint(s) used in any other device(s) connected to the same host.

## **e50147: USB3: Multiple DMA write transfer complete interrupts are generated before final write access handshake to the AXI bus**

**Description:** In USB device mode and Multiple DMA transfers mode, the DMA write-transfer-complete-interrupt is generated multiple clock cycles after the final DMA write access on the AXI bus. The transfer does not wait for completion of the system memory write access handshake.

Delay between the last DMA write access and the DMA interrupt request is determined by an internal operation of the DMA and lasts longer than 50ns. The current DMA interrupt request delay is shorter after DMA write access. Within the interrupt handler, software checks the interrupt source to determine which source introduces the additional delay. During these checks, software has the opportunity to access the system memory data before the DMA write is complete.

This issue may be critical for AXI interconnects that use buffering for write accesses. For these systems, READ access to the system memory may be executed before the WRITE access is complete to the same location even if the WRITE access was requested much earlier than read access.

**Workaround:** Using Singular DMA transfer mode can avoid this issue, by setting DSING to 1 and set DMULT to 0 in register USB\_CONF.

## **e50115: USB3: Port Configuration Response is not compliant with the USB compliance TD 7.17 test case**

**Description:** USB 3.0 Compliance TD 7.17 test case is used to verify that a downstream PUT will go to SS.Inactive if tPortConfiguration expires, and an upstream PUT will go to SS.Disabled if tPortConfiguration expires. However, this test case fails because the port configuration response is not compliant with the TD 7.17 test case.

This requirement is not present in the USB 3.0 specification, however, the USB 3.0 compliance TD 7.17 test case requires it. This test does not affect user applications and it does not affect USB 3.0 function.

**Workaround:** To pass the USB compliance test waive the TD 7.17 tPortConfiguration test.

### **e50148: USB3: Race condition possible during software update to TRB in the system memory and DMA reads of same TRB**

**Description:** Transfer Ring Block (TRB) data structure is larger than 64-bit and therefore requires two separate read accesses on a 64-bit data bus. Because of race conditions between software updates to TRB and DMA reads of the TRB, it is possible that DMA read access may be interleaved with the software write access to the same TRB. The race condition might cause TRB content read by DMA to be inconsistent leading to data corruption during the USB transfer.

This situation can occur in USB device mode.

Critical race condition scenario:

Initial assumption: TRB ownership (cycle bit) is set to software and software is expected to update TRB sequence of events.

1. DMA reads first part of TRB that stores pointer to the USB data buffer.
2. Software writes first part of the TRB and sets new value of the pointer.
3. Software writes second part of the TRB that stores TRB ownership bit (cycle bit) and sets ownership to DMA.
4. DMA reads second part of the TRB and determines that ownership is set to DMA and begins processing data buffer using incorrect pointer that has been fetched during step 1.

**Workaround:** Recommend software driver workaround:

Software checks DMA enqueue and dequeue pointers to determine status of the DMA ring. If the DMA is near the end of the TRB ring the software postpones the update of the ownership bit in the system memory. Software waits until DMA stops and reports the end of the transfer ring by indicating a “descriptor missing” interrupt. The ownership (cycle) bit is updated by software when the DMA is stopped.

Limitations of the Software workaround:

There is a potential performance impact although none observed in real applications.

### **e50149: USB3: TRB OUT endpoints transfer blockage and performance delays**

**Description:** During USB device mode, the on-chip buffer for OUT endpoints is implemented as a FIFO queue for all USB OUT packets.

All configured and enabled Device OUT endpoints are ready to receive OUT data packets when the Device FIFO queue is available whether or not the TRB ring is prepared by software and whether the DMA is ready to read OUT packets.

When an OUT packet is received but the DMA is not prepared for transfer (TRB is missing) the DMA generates a “descriptor missing” interrupt to notify software that the transfer ring for DMA should be prepared.

---

Linux Class driver cannot guarantee creation of the TRB in response to “descriptor missing” interrupt.

**Workaround:** Recommend software driver workaround:

In response to the “descriptor missing” interrupt the software driver prepares the local buffer and enables DMA to receive data from the OUT FIFO to the local buffer in the system memory.

Limitations of the software workaround:

- The local buffer created by the software driver may overflow when a USB Class Application in Linux does not receive data for an extended time.
- The “Descriptor missing” interrupt service impacts application performance (particularly ISO transfers) especially when the “descriptor missing” interrupt is serviced with extended delays.

**How to Reach Us:****Home Page:**[nxp.com](http://nxp.com)**Web Support:**[nxp.com/support](http://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'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](http://nxp.com/SalesTermsandConditions).

While NXP has implemented advanced security features, all products may be subject to unidentified vulnerabilities. Customers are responsible for the design and operation of their applications and products to reduce the effect of these vulnerabilities on customer's applications and products, and NXP accepts no liability for any vulnerability that is discovered. Customers should implement appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP, the NXP logo, NXP SECURE CONNECTIONS FOR A SMARTER WORLD, COOLFLUX, EMBRACE, GREENCHIP, HITAG, I2C BUS, ICODE, JCOP, LIFE VIBES, MIFARE, MIFARE CLASSIC, MIFARE DESFire, MIFARE PLUS, MIFARE FLEX, MANTIS, MIFARE ULTRALIGHT, MIFARE4MOBILE, MIGLO, NTAG, ROADLINK, SMARTLX, SMARTMX, STARPLUG, TOPFET, TRENCHMOS, UCODE, Freescale, the Freescale logo, AltiVec, C-5, CodeTEST, CodeWarrior, ColdFire, ColdFire+, C-Ware, the Energy Efficient Solutions logo, Kinetis, Layerscape, MagniV, mobileGT, PEG, PowerQUICC, Processor Expert, QorIQ, QorIQ Qonverge, Ready Play, SafeAssure, the SafeAssure logo, StarCore, Symphony, VortiQa, Vybrid, Airfast, BeeKit, BeeStack, CoreNet, Flexis, MXC, Platform in a Package, QUICC Engine, SMARTMOS, Tower, TurboLink, and UMEMS are trademarks of NXP B.V. All other product or service names are the property of their respective owners. AMBA, Arm, Arm7, Arm7TDMI, Arm9, Arm11, Artisan, big.LITTLE, Cordio, CoreLink, CoreSight, Cortex, DesignStart, DynamIQ, Jazelle, Keil, Mali, Mbed, Mbed Enabled, NEON, POP, RealView, SecurCore, Socrates, Thumb, TrustZone, ULINK, ULINK2, ULINK-ME, ULINK-PLUS, ULINKpro,  $\mu$ Vision, Versatile are trademarks or registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere. The related technology may be protected by any or all of patents, copyrights, designs and trade secrets. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. The Power Architecture and Power.org word marks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org.

© 2019 NXP B.V.

