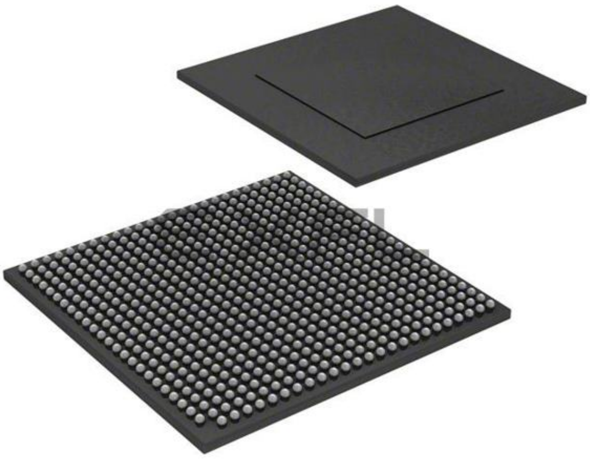
NXP USA Inc. - MCIMX6Q7CVT08AE Datasheet



Welcome to E-XFL.COM

Understanding Embedded - Microprocessors

Embedded microprocessors are specialized computing chips designed to perform specific tasks within an embedded system. Unlike general-purpose microprocessors found in personal computers, embedded microprocessors are tailored for dedicated functions within larger systems, offering optimized performance, efficiency, and reliability. These microprocessors are integral to the operation of countless electronic devices, providing the computational power necessary for controlling processes, handling data, and managing communications.

Applications of Embedded - Microprocessors

Embedded microprocessors are utilized across a broad spectrum of applications, making them indispensable in

Details	
Product Status	Active
Core Processor	ARM® Cortex®-A9
Number of Cores/Bus Width	4 Core, 32-Bit
Speed	800MHz
Co-Processors/DSP	Multimedia; NEON™ SIMD
RAM Controllers	LPDDR2, LVDDR3, DDR3
Graphics Acceleration	Yes
Display & Interface Controllers	Keypad, LCD
Ethernet	10/100/1000Mbps (1)
SATA	SATA 3Gbps (1)
USB	USB 2.0 + PHY (4)
Voltage - I/O	1.8V, 2.5V, 2.8V, 3.3V
Operating Temperature	-40°C ~ 105°C (TA)
Security Features	ARM TZ, Boot Security, Cryptography, RTIC, Secure Fusebox, Secure JTAG, Secure Memory, Secure RTC, Tamper Detection
Package / Case	624-FBGA, FCBGA
Supplier Device Package	624-FCBGA (21x21)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mcimx6q7cvt08ae

Table 3. Summary of Silicon Errata (continued)

Errata	Name	Solution	Page
ERR009218	EIM: Signals fail to drive as outputs during boundary scan test	No fix scheduled	97
ENET			
ERR004512	ENET: 1 Gb Ethernet MAC (ENET) system limitation	No fix scheduled	98
ERR005783	ENET: ENET Status FIFO may overflow due to consecutive short frames	No fix scheduled	99
ERR005895	ENET: ENET 1588 channel 2 event capture mode not functional	No fix scheduled	100
ERR006358	ENET: Write to Transmit Descriptor Active Register (ENET_TDAR) is ignored	No fix scheduled	101
ERR006687	ENET: Only the ENET wake-up interrupt request can wake the system from Wait mode [i.MX 6Dual/6Quad Only]	No fix scheduled	102
ESAI			
ERR008000	ESAI: ESAI may encounter channel swap when overrun/underrun occurs	No fix scheduled	103
EXSC			
ERR004365	EXSC: Exclusive accesses to certain memories are not supported to full AXI specification	No fix scheduled	104
ERR005828	EXSC: Protecting the EIM memory map region causes unpredictable behavior	No fix scheduled	105
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	106
GPMI			
ERR008001	GPMI: GPMI does not support the Set Feature command in Toggle mode	No fix scheduled	108
GPU			
ERR004341	GPU2D: Accessing GPU2D when it is power-gated will cause a deadlock in the system	No fix scheduled	109
ERR005908	GPU2D: Image quality degradation observed for stretch blits when the stretch factor is exactly an integer [i.MX 6Dual/6Quad Only]	No fix scheduled	110
ERR004300	GPU3D: L1 cache performance drop [i.MX 6Dual/6Quad Only]	No fix scheduled	111
ERR004484	GPU3D: L1 cache “Write Address Data” pairing error [i.MX 6Dual/6Quad Only]	No fix scheduled	112
ERR005216	GPU3D: Black texels in Android App Singularity 3D [i.MX 6Dual/6Quad Only]	No fix scheduled	113
HDMI			
ERR003744	HDMI: 9000446457—Audio DMA does not generate an interrupt after software stops DMA transaction	No fix scheduled	114
ERR003745	HDMI: 9000440660—Audio DMA fails to stop after ERROR detection	No fix scheduled	115
ERR004308	HDMI: 8000504668—The arithmetic unit may get wrong video timing values although the FC_* registers hold correct values	No fix scheduled	116

ERR003728 ARM: 740661—Event 0x74 / PMUEVENT[38:37] may be inaccurate

Description:

Event 0x74 counts the total number of Neon instructions passing through the register rename pipeline stage. Due to the erratum, the “stall” information is not taken into account. So, one Neon instruction that remains for n cycles in the register rename stage is counted as n Neon instructions. As a consequence, the count of event 0x74 might be corrupted, and cannot be relied upon. The event is also reported externally on PMUEVENT[38:37], which suffers from the same inaccuracy.

Projected Impact:

The implication of this erratum is that Neon instructions cannot be counted reliably in the versions of the product that are affected by this erratum.

Workarounds:

No workaround is possible to achieve the required functionality of counting how many Neon instructions are executed (or renamed) in the processor.

Proposed Solution:

No fix scheduled

Linux BSP Status:

Software workaround is not needed because this erratum will not be encountered in normal device operation. The Freescale Linux BSP does not support this optional profiling feature. Users may add support for this profiling feature as required, but should ensure the multiple errata impacting the ARM PMU (Performance Monitoring Unit) are considered especially for multi-core usage.

ERR003736 ARM: 756421—Sticky Pipeline Advance bit cannot be cleared from debug APB accesses

Description:

The Sticky Pipeline Advance bit is bit[25] of the DBGDSCR register. This bit enables the debugger to detect whether the processor is idle. This bit is set to 1 every time the processor pipeline retires one instruction.

A write to DBGDRCR[3] clears this bit.

The erratum is that the Cortex-A9 does not implement any debug APB access to DBGDRCR[3].

Projected Impact:

Due to the erratum, the Sticky Pipeline Advance bit in the DBGDSCR cannot be cleared by the external debugger. In practice, this makes the Sticky Pipeline Advance bit concept unusable on Cortex-A9 processors.

Workarounds:

There is no practical workaround for this erratum. The only possible way to reset the Sticky Pipeline Advance bit is to assert the nDBGRESET input pin on the processor. This obviously has the side effect to reset all debug resources in the concerned processor, and any other additional Coresight components nDBGRESET is connected to.

Proposed Solution:

No fix scheduled

Linux BSP Status:

Software workaround is not implemented because this erratum will never be encountered in normal device operation.

ERR003739 ARM: 751470—Imprecise abort on the last data of a cache linefill may not be detected

Description:

Data linefills are returned as 4-beat bursts of 64-bit data on the AXI bus. When the first three beat of data are valid, and the fourth one aborts, then the abort is not detected by the processor logic and no abort exception is taken. The processor then behaves as if no abort is reported on the line. It can allocate the line in its Data Cache, and use the aborted data during its program flow.

Conditions:

The processor needs to work with Data Cache enabled, and access some cacheable memory regions (Write Back, either Shared or Non-Shared).

The memory system underneath the processor needs to be able to generate aborts in this memory region, and must be able to generate aborts with a granularity smaller than the cache line.

Projected Impact:

When the erratum triggers, the processor does not detect the abort, so it might use some invalid (aborted) data without entering the Data Abort exception handler as it should normally do.

Workarounds:

None

Proposed Solution:

No fix scheduled

Linux BSP Status:

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

within the timing window, required to trigger this erratum, might not represent a real case. So, the erratum trigger in the case of self modifying code is probably restricted to read operations, being the consequence of either a speculative load, or a “blind” PLD instruction.

In addition, production of data to an agent external from the coherency domain might fail; particularly, the data target of the cache maintenance operation might not have been made visible to an external DMA engine when it completes. Again, false sharing on a memory region also accessed by an external agent, like a DMA engine, is extremely unlikely to exist. As such, the erratum trigger, when producing data for an external DMA agent, is probably restricted to read operations, being the consequence of either a speculative load, or a “blind” PLD instruction.

Workarounds:

To work around this erratum, ARM recommends to:

- Ensure there is no false sharing (on a cache line size alignment) for both self modifying code and data to be cleaned to an external agent, like a DMA engine.
- Set bit[0] in the undocumented SCU diagnostic control register located at offset 0x30 from the PERIPHBASE address. Setting this bit disables the “migratory bit” feature. This forces a dirty cache line to be evicted to the lower memory subsystem—which is both the point of coherency and the point of unification—when it is being read by another processor. Note that this bit can be written, but is always Read as Zero.
- Insert a DSB instruction in front of the cache maintenance operation. Note that if the cache maintenance operation is executed within a loop with no other memory operations, ARM only recommends adding a DSB prior to entering the loop.

Note that the atomicity between the DSB and the cache maintenance operation might not be ensured because an interrupt may still be taken between the two instructions. However, setting the “disable migratory line” bit and inserting the DSB in front of the cache maintenance operation will very significantly decrease the probability to trigger the erratum when false sharing for writes to either self-modifying code memory regions or DMA regions, on a cache line granularity, which is likely to be the case.

With these workarounds, the likely occurrence of this erratum is sufficiently low that the erratum does not limit or severely impair the intended use of specified features.

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.

Projected Impact:

If the program reaches a stable state where the internal exclusive monitor keeps on being cleared in the middle of the LDREX/STREX sequence, then the processor might encounter a livelock situation.

In practice, this scenario seems very unlikely to happen because several conditions might prevent the erratum from happening:

- Usual LDREX/STREX code sequences do not contain any DSB, so that it is very unlikely that the system would return the abort answer precisely in the middle of the LDREX/STREX sequence on each iteration.
- Some external irritators (for example, interrupts) might happen and cause timing changes which might exit the processor from its livelock situation.
- Branch prediction is very usually enabled, so the final branch in the loop will usually be correctly predicted after a few iterations of the loop, preventing the speculative LDR to be issued, so that the next iteration of the LDREX/STREX sequence will succeed.

Workarounds:

The following two workarounds are available for this erratum:

- Turn on the branch prediction.
- Remove the DSB in the middle of the LDREX/STREX sequence. If a DSB is truly required, it is strongly recommended to place it before the LDREX/STREX sequence, and implement the LDREX/STREX sequence as recommended by the ARM architecture.

Proposed Solution:

No fix scheduled

Linux BSP Status:

Software workaround implemented in Linux BSP codebase in all releases. Software workaround is to enable branch prediction which is enabled by default in the BSP GA release.

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 erratum only affects configurations which implement the parity support option. i.MX6 parity is not supported. In the Freescale Linux implementation, the parity error detection is disabled and GIC parity interrupt 125, is masked in the BSP. The parity feature is disabled by default and should not be enabled.

ERR009605 ARM (CA9): 761320—Full cache line writes to the same memory region from at least two processors might deadlock the processor

Description:

Under very rare circumstances, full cache line writes from (at least) 2 processors on cache lines in hazard with other requests may cause arbitration issues in the SCU, leading to processor deadlock. To trigger the erratum, at least three agents need to be working in SMP mode, and accessing coherent memory regions.

Two or more processors need to perform full cache line writes, to cache lines which are in hazard with other access requests in the SCU. The hazard in the SCU happens when another processor, or the ACP, is performing a read or a write of the same cache line.

The following example describes one scenario that might cause this deadlock:

- CPU0 performs a full cache line write to address A, then a full cache line write to address B
- CPU1 performs a full cache line write to address B, then a full cache line write to address A
- CPU2 performs read accesses to addresses A and B

Under certain rare timing circumstances, the requests might create a loop of dependencies, causing a processor deadlock.

Projected Impact:

When the erratum happens, it leads to system deadlock.

It is important to note that any scenario leading to this deadlock situation is uncommon. It requires two (or more) processors writing full cache lines to a coherent memory region, without taking any semaphore, with another processor or the ACP accessing the same lines at the same time, meaning that these latter accesses are not deterministic. This, combined with the extremely rare microarchitectural timing conditions under which the defect can happen, explains why the erratum is not expected to cause any significant malfunction in real systems.

Workarounds:

This erratum can be worked around by setting bit[21] of the undocumented Diagnostic Control Register to 1. This register is encoded as CP15 c15 0 c0 1.

The bit can be written in Secure state only, with the following Read/Modify/Write code sequence:

```
MRC p15,0,rt,c15,c0,1
ORR rt,rt,#0x200000
```

When this bit is set, the “direct eviction” optimization in the Bus Interface Unit is disabled, which means this erratum cannot occur.

Setting this bit might prevent the Cortex-A9 from utilizing the full bandwidth when performing intensive full cache line writes, and therefore a slight performance drop might be visible.

In addition, this erratum cannot occur if at least one of the following bits in the Diagnostic Control Register is set to 1:

- — Write to ADDR_AUD_N2
- — Write to ADDR_AUD_N1

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. Software workaround has been implemented with HDMI driver enabled.

ERR005192 MIPI: Reverse direction long packets with no payload incorrectly issue a CRC error for MIPI DSI

Description:

The issue appears when a DSI device, which is in reverse mode, sends a long packet with no payload. When receiving this packet, the DSI host controller checks the 16bit CRC field of the packet and incorrectly issues an error.Although there is no apparent reason for a device to send a long packet with no payload, there is no restriction in the DSI specification that forbids this. Also, there is no data loss resulting from this bug, since a long packet with no payload carries no data. The only inconvenience is that a CRC error is asserted in the bit crc_err of the register ERROR_ST1.

Projected Impact:

The CRC error is asserted in the bit crc_err of the register ERROR_ST1, when DSI received a long packet with no payload. This errata does not affect the correct functionality.

Workarounds:

To avoid the assertion of the CRC error, disable verification of CRC reception errors in bit en_CRC_rx of the register PCKHDL_CFG. When disabling the CRC verification on the receive path, users should be aware that the CRC verification will be disabled for all reverse packets and not limited just to the long packets with no payload.

Proposed Solution:

No fix scheduled

Linux BSP Status:

A software workaround is possible but it hasn’t been implemented in the Linux BSP yet. BSP functionality may be affected in some configurations and use cases as described above.

ERR004312 MLB: Multi frame per sub-buffer mode is not supported

Description:

MLB Multi frame per sub-buffer mode is not supported.

Projected Impact:

Low.

Workarounds:

Do not use Multi frame per sub-buffer mode. The user should set the MFE bit to “0” in the Channel Allocation Table (CAT) in order to avoid this issue.

Proposed Solution:

No fix scheduled

Linux BSP Status:

Software workaround not needed in the BSP. Functionality or mode of operation in which the erratum may manifest itself is not used. The BSP driver does not set the MFE bit to 1 but uses the default value 0.

ERR003754 PCIe: 9000403702—AHB/AXI Bridge Master responds with UR status instead of CA status for inbound MRd requesting greater than CX_REMOTE_RD_REQ_SIZE

Description:

The AHB/AXI Bridge RAM is sized at configuration time to support inbound read requests with a maximum size of CX_REMOTE_RD_REQ_SIZE. When this limit is violated the core responds with UR status, when it should respond with CA status.

Projected Impact:

Software gets the wrong status.

Workarounds:

None.

Proposed Solution:

No fix scheduled

Linux BSP Status:

Software workaround cannot be implemented to mask or workaround this SoC issue. This erratum will result in impacted or reduced functionality as described above.

ERR004321 PCIe: 9000470913—Power Management Control: Core might enter L0s/L1 before Retry buffer is empty

Description:

The PCIe base specification states that before the L1 state can be entered, the Retry buffer must be empty.

For PM Directed L1 Entry

5.3.2.1. Entry into the L1 State

The Downstream component then waits until it receives a Link Layer acknowledgement for the PMCSR Write Completion, and any other TLPs it had previously sent. The component must retransmit a TLP out of its Data Link Layer Retry buffer if required to do so by Data Link Layer 15 rules.

For ASPM L1 Entry

5.4.1.2.1. Entry into the L1 State

The Downstream component must wait until it receives a Link Layer acknowledgement for the last TLP it had previously sent (the retry buffer is empty). The component must retransmit 30 a TLP out of its Data Link Layer Retry buffer if required by the Data Link Layer rules.

In Addition For Entry into The L0s State

5.4.1.1.1. Entry into the L0s State

No TLP is pending to transmit over the Link, or no FC credits are available to transmit any TLPs.

This can be interpreted as meaning the retry buffer should be empty. This is because it might be necessary to retransmit a TLP over the link, until a TLP has been acknowledged. The core does not wait for the retry buffer to be empty before commencing L0s or L1 entry.

Conditions:

Scenario Setup:

1. Transmit a TLP from the DWC_pcie core
2. Suppress Ack/Nak transmission from the Link Partner.
3. Initiate PM directed L1, ASPM L0s or ASPM L1 entry.
4. The core will enter the appropriate low power state, even though it still has TLPs in the retry buffer.

Projected Impact:

Low power states are entered inappropriately.

ERR005186 PCIe: The PCIe Controller Core Does Not Send Enough TS2 Ordered Sets During Link Retrain And Speed Change

Description:

The PCIe core might send less than 32 TS2 ordered sets during link retraining and speed changing if the remote partner sends more TS1 ordered sets than expected. This occurs when the “Extended Synch” bit cleared in PCIe and set at the remote partner.

Scenario Setup:

- Link partners agree to do speed change negotiation and move to Recovery.
- The remote partner stays in Recovery.RcvrLock longer and sends more TS1s (for example, Extended Synch bit is set).
- In Recovery.RcvrCfg state core will send less than 32 TS2s before transition to Recovery.Speed

Projected Impact:

The speed change negotiation might fail.

Workarounds:

In current PCIe core, there is no signal indicating that remote partner has “Extended Synch” bit set per PCIe base spec. The workaround is to know in advance if the link partner has the “Extended Synch” bit set and in that case set that bit in the PCIe also. The “Extended Synch” is intended for a logic analyzer. It may be used if there are repeaters on the link.

Proposed Solution:

No fix scheduled

Linux BSP Status:

Software workaround is not implemented because this erratum will never be encountered in normal device operation.

ERR007573 PCIe: Link and lane number-match not checked in recovery (9000569433)

Description:

When the core's LTSSM is in Recovery.RcvLock or Recovery.RcvCfg, and it receives TS Ordered sets, it does not check whether the link and lane numbers of the received TS Ordered Sets match what is being transmitted on those same lanes.

Scenario Setup:
Core is in link state "Recovery.RcvLock" or "Recovery.RcvCfg" and receives TS Ordered Sets with link and lane number not matching what is being transmitted on those same Lanes.

The absence of link and lane number match checks in Recovery.RcvrLock and Recovery.RcvrCfg states only affects single lane configurations (CX_NL = 1). All configurations are not affect, as stated in the Impacted Configurations section above.

Projected Impact:

The core moves from Recovery.RcvLock to Recovery.RcvCfg or from Recovery.RcvCfg to Recovery.Idle and LTSSM misses the condition of link and lane number match and moves to the next state without robustness.

Workarounds:

None

Proposed Solution:

No fix scheduled.

Linux BSP Status:

Software workaround cannot be implemented to mask or workaround this SoC issue. This erratum will result in impacted or reduced functionality as described above.

ERR008587 PCIe: Random link down after warm reset [i.MX 6Dual/6Quad Only]

Description:

In rare cases, the PCIe link may go down after a warm reset occurs.

Projected Impact:

Loss of PCIe link

Workarounds:

Prior to executing a warm reset, clear the IOMUXC_GPR1[REF_SSP_EN] to disable the PCIe reference clock. Once the warm reset is complete and the clocks are stable again, re-enable the PCIe reference clock by setting IOMUXC_GPR1[REF_SSP_EN] to 1.

Proposed Solution:

No fix scheduled.

Linux BSP Status:

Software workaround integrated in Linux BSP codebase starting in release L3.10.53_1.1.0_ga.

ERR007926 ROM: 32 kHz internal oscillator timing inaccuracy may affect SD/MMC, NAND, and OneNAND boot [i.MX 6Dual/6Quad Only]

Description:

The internal boot ROM uses the general-purpose timer (GPT) as a timing reference for event and timeout measurement during the boot process. The ROM uses the 32 kHz clock as the clock source for the GPT. There will be a short period during device power-up when the SoC will be using the internal ring oscillator until the crystal oscillator is running. Once the crystal oscillator is running, the SoC will automatically switch from the internal oscillator to the crystal oscillator. Consequently, there will be a period of time when the SoC will be booting and using the internal ring oscillator as its reference clock and the ROM code will be dependent on that clock.

The internal ring oscillator is less accurate than a crystal oscillator and may be up to two times faster than a 32 kHz external crystal oscillator. The ROM code assumes the reference clock is 32 kHz, so in the presence of a faster reference clock some delays or timeout configurations in the ROM code will be shorter than expected and may affect SD/MMC boot, NAND boot, and One NAND boot.

NOR and SPI-NOR boot modes are not affected by this issue because these modes do not use timeouts.

The potential effects are:

1. The SD/MMC card specification may be violated if the SD/MMC card Nac parameter is larger than 50 ms, or if its initialization time is greater than 500 ms.
2. According to the SD 3.0 specification, the controller should wait a minimum of 5 ms after disabling SDCLK before re-enabling SDCLK when voltage switching. In the worst case, the ROM code may only wait 2.5 ms.
3. According to the SD 3.0 specification, the timeout for a CMD6 data transaction response is 100 ms. In the worst case, the ROM code may timeout after 50 ms and therefore not conform to the specification.
4. One NAND boot may fail if the One NAND memory tRD1 is greater than 1.5 ms.
5. NAND boot may fail if the NAND memory tRST parameter is greater than 11 ms or if its tR parameter is greater than 1 ms.

Projected Impact:

To date, this failure has not been observed on any system. The description and workarounds presented here are based on analysis of the timings in the ROM boot sequence and indicates the possibility that SD/MMC boot, NAND boot, or OneNAND boot may be affected. SD/MMC card specifications may not be met during boot.

Workarounds:

SDMMC boot:

1. SD/MMC: Choose an SD/MMC card for which the Nac parameter is to be specified less than 50 ms and its initialization time is less than 500 ms.

ERR003761 SATA: 9000433864—COMRESET and COMWAKE do not always contain six bursts

Description:

When a COMRESET or a COMWAKE is sent by the Host, it does not always send six bursts, but sometimes only five. It has been observed when OOB detection has failed, such as when COMWAKE is missed, or an error state or retry occurs.

NOTE

While this behavior is sufficient to complete OOB, it technically does not meet the SATA specification.

Projected Impact:

Because an OOB receiver should look for three gaps (four consecutive bursts), this behavior should not affect normal operation. It could prevent compliance from being achieved, if the number of bursts sent is tested. The probability of this problem occurring is high.

Workarounds:

None.

Proposed Solution:

No fix scheduled

Linux BSP Status:

Software workaround cannot be implemented to mask or workaround this SoC issue. This erratum will result in impacted or reduced functionality as described above.

ERR003765 SATA: 9000447627—Global reset does not clear IS.IPS register bits when P#IS is non-zero

Description:

If any of the bits of the P#IS register is set when software issues global reset by setting GHC.HR=1, then the corresponding IS.IPS bit remains set, causing an erroneous interrupt (when GHC.IE=1) due to the p#_vint signal being registered/delayed from the Port module.

Projected Impact:

This erratum causes unnecessary interrupt processing. The probability of this problem occurring is medium.

Workarounds:

Generate global reset when all P#IS and IS bits are cleared.

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. Linux libata driver avoids this issue. The GHC.IE is set after the GHC.HR is completed.

ERR007881 USB: Timeout error in Device mode

Description:

If a receive FIFO overrun occurs (due to a busy condition on the system bus) when the USB controller is in Device mode, the controller may stop responding to host tokens, causing current transactions to time out. This situation will be recovered after FIFO is not overrun.

Projected Impact:

Implementing the workaround shown will prevent receive FIFO overruns, but cause a 10%-30% impact to USB performance.

Workarounds:

Set Stream Disable mode (USB_nUSBMODE[SDIS]=1) to prevent receive FIFO overruns.

Proposed Solution:

No fix scheduled

Linux BSP Status:

Software workaround implemented in Linux BSP codebase starting in imx_3.10.53_1.1.0_ga.