Mask Set Errata for Mask 0N99Z

This report applies to mask 0N99Z for these products:

- MIMX8QX#####ZAC
- MIMX8DX#####ZAC
- MIMX8UX#####ZAC

Table 1. Errata and Information Summary

Erratum ID	Erratum Title
ERR050068	AUDIO: Incorrect 24 MHz clock source at Audio Clock Mux input
ERR010947	DRAM: DQS/DQSN glitch suppression resistors must be enabled during read-leveling
ERR010944	DRAM: In LPDDR4 mode, tMPCWR timing violation in incremental DQS2DQ Training
ERR010946	DRAM: In LPDDR4 mode: Auto refresh must be disabled during DQS2DQ training
ERR050341	DRAM: LPDDR4 VREF training may result in a non-optimal value
ERR010945	DRAM: PUB does not program LPDDR4 DRAM DDRPHY_MR22 prior to running DRAM ZQ calibration
ERR050340	DRAM: The LPDDR4 DRAM initialization may experience large training time variations or stall when Read Data Bus Inversion (DBI) bit deskew training is enabled
ERR010948	DRAM: Timing Violation from Read/Write to MRW in LPDDR4 mode
ERR050395	ENET: Ethernet RX hang when receiving traffic through multiple queues
ERR011543	FlexCAN: Nominal Phase SJW incorrectly applied at CRC Delimiter
ERR050135	JPEG DECODER: multi-frame jpeg bitstream may not be correctly decoded when there is a small size frame inside
ERR010527	LPUART: Setting and immediately clearing SBK bit can result in transmission of two break characters
ERR010930	PCIE: EOM single point sample error/valid result is not correct
ERR011370	PCIE: EP, PM_PME: L1 Exit Does Not Occur when PME Service Timeout Mechanism Expires
ERR011194	PCIE: Plesiochronous loopback is not functional in PCIe Gen3
ERR050148	USB3: Race condition possible during software update to TRB in the system memory and DMA reads of same TRB

Table 2. Revision History

Revision	Changes
0, 02/2020	Initial revision
1, 05/2020	The following erratum was added. • ERR010527

ERR050068: AUDIO: Incorrect 24 MHz clock source at Audio Clock Mux input

Description: The DSC of the Audio subsystem / Audio DMA 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_rst_clk”).

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 GPT. The DSC reset clock is enabled and disabled during reset sequences, which would impact the modules using that clock through the ACM. Note that a reset sequence can take place (e.g. for the HIFI) while audio blocks are operational. The “24_MHz_functional” must be used.

Only the GPT 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”.

ERR010947: DRAM: DQS/DQSN glitch suppression resistors must be enabled during read-leveling

Description: By default DQS/DQSN glitch suppression resistors 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-leveling gate training.

Workaround: Enable the strongest 355 ohm glitch suppression resistors during gate training. Scripts provided by NXP’s DRAM RPA (Register Programming Aid) implement the required workaround through DDRPHY_DX8SLbDQSCTL register.

ERR010944: 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 hardware incremental dqs2dq training routine 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.

ERR010946: DRAM: In LPDDR4 mode: Auto refresh must be disabled during DQS2DQ training

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

Workaround: For initial power-up (cold boot), disable auto 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.

ERR050341: DRAM: LPDDR4 VREF training may result in a non-optimal value

Description: During LPDDR4 initialization, when performing LPDDR4 VREF training (through DDRPHY_PIR[VREF]), a discrepancy may be observed between the training-generated DDRPHY_MR14 value and an actual signal-measured VREF_DQ value such that the "trained" DDRPHY_MR14 value may not result in the most optimal VREF_DQ setting. However, the discrepancy is minimal such that no DRAM data failures have occurred due to this.

Workaround: The user should continue to use the hardware based VREF training as no DRAM failures occur due to the minimal discrepancy.

ERR010945: DRAM: PUB does not program LPDDR4 DRAM DDRPHY_MR22 prior to running DRAM ZQ calibration

Description: When the PHY Utility Block (PUB) initializes the DRAM, the DDRPHY_MR22 is programmed after ZQ Calibration. This may result in incorrect ZQ calibration results on the LPDDR4 DRAM side, because DDRPHY_MR22[CODT] works as the controller On Die Termination (ODT) replica during the Pull-Up calibration. Therefore the expected controller ODT must be programmed into DDRPHY_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.

ERR050340: DRAM: The LPDDR4 DRAM initialization may experience large training time variations or stall when Read Data Bus Inversion (DBI) bit deskew training is enabled

Description: Read DBI bit deskew training is an extension of the Read bit deskew training. When performing LPDDR4 Read bit deskew training (through the DDRPHY_PIR register), the following issues may be encountered:

• When Read DBI deskew training is enabled (DDRPHY_DTCR0.DTRDBITR = 2'bx1), there is a possibility to observe large training time variations or even a stall

• When Read DBI deskew training is disabled (DDRPHY_DTCR0.DTRDBITR = 2'bx0), incorrect values are programmed in DDRPHY_DXnBDLR5 (i.e. the DM Read BDL) after Read bit deskew training is completed

Workaround: For mask sets 1N94W and 0N95W

A software workaround has been include in the SCU firmware (SCFW) since version 1.1.8, released to support the Linux 4.14.78_1.1.0 BSP and version 1.2.8, released to support the Linux 4.14.98_2.0.1 BSP.

For mask set 0N99Z

A software workaround has been include in the SCU firmware (SCFW) since version 1.3.0, released to support the Linux 4.14.98_2.3.0 BSP.

ERR010948: 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.

ERR050395: ENET: Ethernet RX hang when receiving traffic through multiple queues

Description: M4-FreeRTOS and A35-Linux are enabled to share the same Ethernet module by using different queues. At least 2 queues are configured to receive packets, with flushing enabled (RX_FLUSHx). When queues become full, packets are normally flushed, but under certain conditions of traffic, a lock-up of the Rx path can happen instead. When this occurs, the buffer descriptor for the last received packet contains an incorrect packet size (equal to the maximum buffer size). Packets cannot be received anymore, but the TX path remains unaffected. To recover the RX path, the ENET hardware block must be reset and re-configured.

Workaround: Unless the use case demands it, disable flushing to ensure the problem does not happen.

ERR011543: 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.

ERR050135: 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.

ERR010527: LPUART: Setting and immediately clearing SBK bit can result in transmission of two break characters

Description: When the LPUART transmitter is idle (LPUART_STAT[TC]=1), two break characters may be sent when using LPUART_CTRL[SBK] to send one break character. Even when LUART_CTRL[SBK] is set to 1 and cleared (set to 0) immediately.

Workaround: To queue a single break character via the transmit FIFO, set LPUART_DATA[FRETSC]=1 with data bits LPUART_DATA[T9:T0]=0.

ERR010930: 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.

ERR011370: PCIE: EP, PM_PME: L1 Exit Does Not Occur when PME Service Timeout Mechanism Expires

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

device_type =4'b0000, located at PCIEX1_CTRL0, address 0x5f140000, bit[27:24]

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 8 "APPS_PM_XMT_PME" of HSIO GPR register "PCIEX1_CTRL2", address 0x5f140008.

ERR011194: 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.

ERR050148: 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.

How to Reach Us:

Home Page:
nxp.com

Web Support:
nxp.com/support