Document identifier: IMX8X_0N99Z
Rev. 3.1, 3/2023

IMX8X_0N99Z

Mask Set Errata

Mask Set Errata for Mask 0N99Z

Revision History

This report applies to mask 0N99Z for these products:

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

Table 1. Revision History

Revision	Date	Significant Changes
3.1	3/2023	The following errata were revised. • ERR051393
3	11/2022	The following errata were added. • ERR051393 • ERR051407 • ERR051198
2	9/2021	The following errata were added. • ERR050537 • ERR051041 • ERR050102 • ERR050246 The following errata were revised. • ERR050395 • ERR050341 • ERR050340
1	5/2020	The following errata were added. • ERR010527
0	2/2020	Initial Revision

Errata and Information Summary

Table 2. Errata and Information Summary

Erratum ID	Erratum Title
ERR051393	Arm/Cortex-A core memory corruption
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

Table continues on the next page...

Table 2. Errata and Information Summary (continued)

Erratum ID	Erratum Title
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
ERR050102	DRAM: Periodic hardware based DQS2DQ calibration is not supported
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
ERR050246	FlexCAN: Receive Message Buffers may have its Code Field corrupted if the Receive FIFO function is used
ERR050537	FlexSPI: Read timing sequence mismatches with several existing SPI NOR devices in dual, quad, and octal modes
ERR050135	JPEG DECODER: multi-frame jpeg bitstream may not be correctly decoded when there is a small size frame inside
ERR051041	LPIT: CVAL cannot be read correctly during timer running
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
ERR051198	PWM: PWM output may not function correctly if the FIFO is empty when a new SAR value is programmed
ERR050148	USB3: Race condition possible during software update to TRB in the system memory and DMA reads of same TRB
ERR051407	USB3: USB full speed mode may fail to work

Known Errata

ERR051393: Arm/Cortex-A core memory corruption

Description

A race condition in the Cortex-A CPU subsystem during initialization can cause memory setup values to be incorrectly applied to some Cortex-A cluster internal memories.

This may lead to memory corruption on some devices.

Workaround

An SCU firmware revision implementing SCF-838 is required to modify the Cortex-A CPU subsystem initialization sequence to avoid the race condition. Updates have been integrated to Linux BSP release starting from L5.10.35.

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

A software workaround has been included in the SCU firmware (SCFW) since version 1.4.0. The software workaround has been included in the MX8QXP DDR Register Programming Aid (RPA) since RPA version 14.

ERR050102: DRAM: Periodic hardware based DQS2DQ calibration is not supported

Description

If periodic hardware based DQS2DQ calibration is enabled, the resultant latency introduced can cause underrun conditions, or worst case a lock-up, in some of the key sub-systems, such as the display and imaging interfaces, impacting their performance capabilities.

Workaround

Currently DQS2DQ calibration only takes place on power up and when resuming from low power modes. To date no failures or stability issues have been observed across the full process, voltage and temperature ranges.

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

A software workaround has been included in the SCU firmware (SCFW) since version 1.3.0. The software workaround has been included in the MX8QXP DDR Register Programming Aid (RPA) since RPA version 13.

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

Two or more applications 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.

Or if reset is acceptable:

To recover the RX path, the ENET hardware block must be reset and re-configured

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.

ERR050246: FlexCAN: Receive Message Buffers may have its Code Field corrupted if the Receive FIFO function is used

Description

If the Code Field of a Receive Message Buffer is corrupted it may deactivate the Message Buffer, so it is unable to receive new messages. It may also turn a Receive Message Buffer into any type of Message Buffer as defined in the Message buffer structure section in the device documentation.

The Code Field of the FlexCAN Receive Message Buffers (MB) may get corrupted if the following sequence occurs.

1- A message is received and transferred to an MB (i.e. MBx)

2- MBx is locked by software for more than 20 CAN bit times (time determines the probability of erratum to manifest).

3- SMB0 (Serial Message Buffer 0) receives a message (i.e. message1) intended for MBx, but destination is locked by the software (as depicted in point 2 above) and therefore NOT transferred to MBx.

4- A subsequent incoming message (i.e. message2) is being loaded into SMB1 (as SMB0 is full) and is evaluated by the FlexCAN hardware as being for the FIFO.

5- During the message2, the MBx is unlocked. Then, the content of SMB0 is transferred to MBx and the CODE field is updated with an incorrect value.

The problem does not occur in cases when only Rx FIFO or only a dedicated MB is used (i.e. either RX MB or Rx FIFO is used). The problem also does not occur when the Enhanced Rx FIFO and dedicated MB are used in the same application. The problem only occurs if the FlexCAN is programmed to receive in the Legacy FIFO and dedicated MB at the same application.

Workaround

This defect only applies if the Receive FIFO (Legacy Rx FIFO) is used. This feature is enabled by RFEN bit in the Module Control Register (MCR). If the Rx FIFO is not used, the Receive Message Buffer Code Field is not corrupted.

If available on the device, use the enhanced Rx FIFO feature instead of the Legacy Rx FIFO. The Enhanced Rx FIFO is enabled by the ERFEN bit in the Enhanced Rx FIFO Control Register (ERFCR).

The defect does not occur if the Receive Message Buffer lock time is less than or equal to the time equivalent to 20 x CAN bit time.

The recommended way for the CPU to service (read) the frame received in a mailbox is by the following procedure:

1. Read the Control and Status word of that mailbox.
2. Check if the BUSY bit is deasserted, indicating that the mailbox is not locked. Repeat step 1) while it is asserted.
3. Read the contents of the mailbox.
4. Clear the proper flag in the IFLAG register.
5. Read the Free Running Timer register (TIMER) to unlock the mailbox

In order to guarantee that this procedure occurs in less than 20 CAN bit times, the MB receive handling process in software (step 1 to step 5 above) should be performed as a 'critical code section' (interrupts disabled before execution) and should ensure that the MB receive handling occurs in a deterministic number of cycles.

ERR050537: FlexSPI: Read timing sequence mismatches with several existing SPI NOR devices in dual, quad, and octal modes

Description

The FlexSPI controller expects every read command has at least one latency cycle between address phase and data phase to account for turnaround time on the IO bus. In multiple IO modes such as dual, quad, and octal modes, the FlexSPI controller inserts one additional clock cycle following the address (or command modifier) phase in order to prevent contention on bidirectional IO pins.

It will cause drive conflict if the SPI NOR device's timing sequence does not contain dummy cycles after the command/address cycles. Such drive conflict might result in reading wrong data value. The problem usually happens when reading a SPI slave's register space.

Workaround

For FlexSPI memory device that supports multi IO Read command with zero latency cycle between address phase and data phase, use single line mode for read command, or use different data line to issue commands and read data.

The official NXP BSP release uses a signal line (1S-1S-1S) mode, but not multiple IO modes when access FlexSPI device registers.

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.

ERR051041: LPIT: CVAL cannot be read correctly during timer running

Description

The LPIT implements a functional clock domain for the counter and a bus clock domain for the register interface. The CVAL register increments on each clock cycle of the functional clock domain. Reading the register value happens on the bus clock domain. As these clock domains are not synchronous, there is a possibility that the register is read whilst the counter value is updating. This can lead to reading incorrect values as some bits of the counter may have settled whilst others are still transitioning to the new state.

Workaround

There should be no need to read the timer value since the timer is normally used to generate a periodic interrupt. However, if the timer value needs to be read, software can read the register more than once until the value matches the previous value. This ensures that the read operation did not coincide with the timer update and the value read is the actual timer value.

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.

ERR051198: PWM: PWM output may not function correctly if the FIFO is empty when a new SAR value is programmed

Description

When the PWM FIFO is empty, a new value programmed to the PWM Sample register (PWM_PWMSAR) will be directly applied even if the current timer period has not expired.

If the new SAMPLE value programmed in the PWM_PWMSAR register is less than the previous value, and the PWM counter register (PWM_PWMCNR) that contains the current COUNT value is greater than the new programmed SAMPLE value, the current period will not flip the level. This may result in an output pulse with a duty cycle of 100%.

Workaround

Program the current SAMPLE value in the PWM_PWMSAR register before updating the new duty cycle to the SAMPLE value in the PWM_PWMSAR register. This will ensure that the new SAMPLE value is modified during a non-empty FIFO, and can be successfully updated after the period expires.

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.

ERR051407: USB3: USB full speed mode may fail to work

Description

USB3 module supports USB3 .0 PHY and USB2.0 PHY. Very limited parts may fail to work on full speed mode (both host and device modes) for USB3 port due to higher threshold in full speed receiver of USB2.0 PHY. One example failure symptom is, the enumeration is failed when connecting full speed USB mouse to USB3 port, especially under high temperature.

Workaround

The recommended workaround is to configure threshold voltage value of single ended receiver by setting USB2.0 PHY register AFE_RX_REG5[2:0] to 3'b101 (Register Address is 0x5B198048). The workaround has been integrated to Linux BSP release starting from L5.10.9_1.0.0.

Legal information

Definitions

Draft — A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at http://www.nxp.com/profile/terms, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

Suitability for use in automotive applications — This NXP product has been qualified for use in automotive applications. If this product is used by customer in the development of, or for incorporation into, products or services (a) used in safety critical applications or (b) in which failure could lead to death, personal injury, or severe physical or environmental damage (such products and services hereinafter referred to as "Critical Applications"), then customer makes the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, safety, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP. As such, customer assumes all risk related to use of any products in Critical Applications and NXP and its suppliers shall not be liable for any such use by customer. Accordingly, customer will indemnify and hold NXP harmless from any claims, liabilities, damages and associated costs and expenses (including attorneys' fees) that NXP may incur related to customer's incorporation of any product in a Critical Application.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Trademarks

Date of release: 3/2023

Document identifier: IMX8X_0N99Z