

IMX8MN_0N14Y

Mask Set Errata

Rev. 1.4 — 25 March 2025

Errata

1 Mask Set Errata for Mask 0N14Y

1.1 Revision History

This report applies to mask 0N14Y for these products:

- MIMX8MN6DVTJZDA
- MIMX8MN1CVTIZAA
- MIMX8MN5DVTJZAA
- MIMX8MN6DVTJZCA
- MIMX8MN3DVTJZAA
- MIMX8MN1DVTJZAA
- MIMX8MN6CVTIZAA
- MIMX8MN2DVTJZAA
- MIMX8MN4CVTIZAA
- MIMX8MN3CVTIZAA
- MIMX8MN5CVPIZAA
- MIMX8MN3CVPIZAA
- MIMX8MN1DVPIZAA
- MIMX8MN5DVPIZCA
- MIMX8MN5DVPIZDA
- MIMX8MN3DVPIZAA
- MIMX8MN5DVPIZAA
- MIMX8MN1CVPIZAA
- MIMX8MN6DVTJZAA
- MIMX8MN5CVTIZAA
- MIMX8MN4DVTJZAA
- MIMX8MN2CVTIZAA

Table 1. Revision History

Revision	Release Date	Significant Changes
1.4	3/2025	The following errata were removed. <ul style="list-style-type: none">• ERR052162 The following errata were added. <ul style="list-style-type: none">• ERR052039• ERR051123
1.2	5/2022	The following errata were added. <ul style="list-style-type: none">• ERR050814• ERR051272• ERR051182• ERR051198
1	2/2021	Initial Revision



1.2 Errata and Information Summary

Table 2. Errata and Information Summary

Erratum ID	Erratum Title
ERR050447	[SPDIF]: SPDIF clock limitation
ERR003774	AIPS: Unaligned access to AIPS internal registers will result in an abort response.
ERR050358	BSDL: The GPIO1_IO02 used as WDOG_B is set output low when entering boundary scan mode
ERR050310	CM7 Icache/Dcache are not operational
ERR052039	Core: Cortex-M hang if stopped from Cortex-A
ERR050814	DDR: Register corruption possible when software triggered mode register (MR) operations performed in DDR4 mode.
ERR050381	DRAM: DRAM data may be lost while exiting from DDR IO retention mode
ERR009535	ECSPI: Burst completion by SS signal in slave mode is not functional
ERR009606	ECSPI: In master mode, burst lengths of 32n+1 will transmit incorrect data
ERR009165	ECSPI: TXFIFO empty flag glitch can cause the current FIFO transfer to be sent twice
ERR050537	FlexSPI: Read timing sequence mismatches with several existing SPI NOR devices in dual, quad, and octal modes
ERR050226	GPU: Texture L2 Cache idle signal may incorrectly clock gate the texture engine in GPU
ERR007805	I2C: When the I2C clock speed is configured for 400 kHz, the SCL low period violates the I2C spec of 1.3 uS min
ERR050045	IOMUX: Setting ODE control bit of I2C IOs causes malfunction
ERR051123	ISI: Memory overwrite occurring outside of allocated buffer space corrupting system memory
ERR051182	ISI: U and V colors are reversed when horizontal flip is enabled in ISI configuration
ERR051198	PWM: PWM output may not function correctly if the FIFO is empty when a new SAR value is programmed
ERR050350	ROM: Exception raised when ROM accesses a reserved region when Field Return fuse is enabled
ERR050359	ROM: USB Serial Download mode supports maximum 3 devices per USB host
ERR050144	SAI: Setting FCONT=1 when TMR>0 may not function correctly
ERR050542	SAI: The Bit Count Timestamp Register (TBCTR, RBCTR) may return a live rather than latched Timestamp
ERR050362	TCM: AXI2AHB cannot handle partial write and causes redundant write operations to TCM
ERR051272	TMU: Bit 31 of registers TMU_TSCR/TMU_TRITSR/TMU_TRATSR invalid

2 Known Errata

ERR050447: [SPDIF]: SPDIF clock limitation

Description

The SPDIF IP includes a DPLL driven from the subsystem clock, which is used to generate a data strobe to sample the incoming bitstream

When the subsystem clock to SPDIF bitrate ratio is too high, the DPLL might not lock to the correct sampling frequency and phase.

For example: If the DPLL is using a 400MHz subsystem clock, it is able to lock in normal use cases of 44.1kHz and above, but it cannot track jitter reliably.

Workaround

Configure the audio_ahb_clk as 200MHz which was connected to gclkw_t0. It impacts the SDMA which share the audio_ahb_clk, while SDMA performance can be restored by partitioning the workload across both SDMA2 and SDMA3.

ERR003774: AIPS: Unaligned access to AIPS internal registers will result in an abort response.

Description

Unaligned access to AIPS internal registers will return an abort response.

Workaround

Only aligned AIPS internal register access is supported. Software should not issue unaligned accesses to AIPS internal registers.

ERR050358: BSDL: The GPIO1_IO02 used as WDOG_B is set output low when entering boundary scan mode

Description

Only GPIO1_IO02 can be multiplexed as WDOG_B to toggle PMIC. When entering boundary scan mode, the GPIO1_IO02 is always low. If this pin is connected with PMIC WDOG_B during boundary scan mode, WDOG_B low will reset the PMIC and prevent normal system boot-up.

Workaround

If the PMIC supports WDOG reset by default, the PMIC WDOG_B pin cannot be connected to GPIO1_IO02 and should be pulled-up to a 100K ohm resistor during boundary scan test. Otherwise, use the WDOG timer buffer circuit. The related reference workaround circuit can be found from i.MX8M Nano Hardware Developer's Guide.

ERR050310: CM7 Icache/Dcache are not operational**Description**

CM7 Icache/Dcache tag memories were incorrectly integrated, preventing the cache memories from working properly.

If the I-cache is enabled, there will be a slight performance degradation (based on cache not-enabled), with no other system implications. However, if the D-cache is enabled the system will return false cache hits and may cause unexpected system behavior.

Workaround

Do not enable the caches on the CM7 core. If the CM7 code size is below 256K, load and run the code from Tightly Coupled Memory (TCM).

The following steps outline loading and running CM7 code from the TCM memory:

Reg 0x30340054 defines the “CM7 Vector table offset register out of reset” and should be set as default 0x00000000 (ITCM) in this case.

Step 1: Reg 0x3039000c 0xaa // release reset, enable TCM

Step 2: Load program bin file to TCM

Step 3: Reg 0x30340058 0x0 //disable CPUWAIT, set CPU in run mode

Note: In CA53/system perspective: DTCM address is from 0x0080_0000 to 0x0081_FFFF, and ITCM address is from 0x007E_0000 to 0x007F_FFFF.

In CM7 perspective: DTCM address is from 0x2000_0000 to 0x2001_FFFF, and ITCM address is from 0x0000_0000 to 0x0001_FFFF.

ERR052039: Core: Cortex-M hang if stopped from Cortex-A**Description**

Stopping the Arm Cortex-M, using the software running on an Arm Cortex-A, could cause the SoC to hang, due to pending hardware level bus transactions issued by the Arm Cortex-M, which are not properly stopped.

Workaround

Refer to application note AN5317 (<https://www.nxp.com/docs/en/application-note/AN5317.pdf>) on how to load code into Cortex-M from the software running on Cortex-A cores.

ERR050814: DDR: Register corruption possible when software triggered mode register (MR) operations performed in DDR4 mode.**Description**

The DDR controller allows user software to manually trigger a mode register read or write operation by setting the MRCTRL0.mr_wr=1. Under certain specific conditions listed below it is possible that DDR controller registers can be corrupted while an internal hardware driven MR access is occurring. The impact of the corruption depends on the registers being accessed.

This issue can only occur when:

1. DDR controller is configured in DDR4 mode
2. A mode register read or write operation is performed by user software by programming the register MRCTRL0.mr_wr=1
3. Separate DDR controller register R/W (or R/W1S or R/W1C) APB write accesses occur close together and
4. An internal hardware driven MR access occurs concurrently
- 4.a. Entering and exiting Self-Refresh or MPSM or when
- 4.b. Per DRAM Addressability mode (PDA) or Per Buffer Addressability(PBA) mode is enabled

Workaround

When performing a software driven MR access, the following polling sequence must be done automatically before performing other DDR controller register accesses:

1. Set the register MRCTRL0.mr_wr=1 (When the MR operation is complete, the DDRC automatically clears this bit)
2. Check the DDR Controller register MRSTAT.mr_wr_busy = 0 If not, go to step (2)
3. Check the DDR Controller register MRSTAT.mr_wr_busy = 0 again (for the second time). If not, go to step (2)

ERR050381: DRAM: DRAM data may be lost while exiting from DDR IO retention mode

Description

There are two DDR PHY input signals PwrOkIn and atpg_mode that have combinational logic powered by VDD_DRAM. When DDR PHY exits from IO retention mode, VDD_DRAM will ramp-up during the time it is OFF, therefore the two signals cannot to be assured as 0 and will not meet the PHY requirement. The issue may cause the IO retention mode to work improperly and data in DRAM might be lost.

Workaround

Keep the VDD_DRAM on while in DDR IO retention mode.

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

Description

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

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

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

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

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

According to the SPI protocol, negation of SSB always causes completion of the burst. However, due to the above issue, the data is not sampled correctly in RxFIFO when $\{\text{BURST_LENGTH}+1\} \bmod 32$ is not equal to $\{\text{actual burst length}\} \bmod 32$.

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

Workaround

Do not use the SS_CTL[x] = 1 option in the Slave mode. The accurate burst length should always be specified using the BURST_LENGTH parameter.

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

Description

When the ECSPi is configured in master mode and the burst length is configured to a value $32n+1$ (where $n=0,1,2,\dots$), the ECSPi will transmit the portions of the first word in the FIFO twice.

For example, if the transmit FIFO is loaded with:

[0] 0x00000001

[1] 0xAAAAAAAA

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

The transmitted sequence in this example will be:

[0] 0x00000001

[1] 0x00000001

[2] 0x00000000

[3] 0xAAAAAAAA

Workaround

Do not use burst lengths of $32n+1$ (where $n=0,1,2,\dots$).

ERR009165: ECSPi: TXFIFO empty flag glitch can cause the current FIFO transfer to be sent twice

Description

When using DMA to transfer data to the TXFIFO, if the data is written to the TXFIFO during an active ECSPi data exchange, this can cause a glitch in the TXFIFO empty signal, resulting in the TXFIFO read pointer (TXCNT) not updating correctly, which in turn results in the current transfer getting resent a second time.

Workaround

This errata is only seen when the SMC (Start Mode Control) bit is set. A modified SDMA script with TX_THRESHOLD = 0 and using only the XCH (SPI Exchange) bit to initiate transfers prevents this errata from

occurring. There is an associated performance impact with this workaround. Testing transfers to a SPI-NOR flash showed approximately a 5% drop in write data rates and a 25% drop in read data rates.

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.

ERR050226: GPU: Texture L2 Cache idle signal may incorrectly clock gate the texture engine in GPU

Description

While running certain graphics cases, the Texture Engine's L2 Cache is waiting too long for more data to be returned from the AXI bus, hence the L2 cache will become idle and incorrectly clock gate the Texture Engine, leading to an eventual hang.

Workaround

GC7000UL GPU module level clock gating enables by default. As a workaround, add an exception in the GPU software to disable TX module level clock gating. The GPU has multiple levels of power management one at the module clock gating and GPU driver level power management. The GPU driver level power management will be used with this workaround.

The related driver is located in the file "drivers/mxc/gpu-viv/hal/kernel/arch/gc_hal_kernel_hardware.c". The code detail are as follows:

```
gcmkONERROR(  
gckOS_WriteRegisterEx(Hardware->os,  
Hardware->core,  
0x00414,  
data));  
}
```

```
if (!_IsHardwareMatch(Hardware, gcv4000, 0x5222)
<b>IsHardwareMatch(Hardware, gcv2000, 0x5108)
<b>IsHardwareMatch(Hardware, gcv7000, 0x6202)
<b>IsHardwareMatch(Hardware, gcv7000, 0x6203)
|| (gckHARDWARE_IsFeatureAvailable(Hardware, gcvFEATURE_TX_DESCRIPTOR)
&& !gckHARDWARE_IsFeatureAvailable(Hardware, gcvFEATURE_TX_DESC_CACHE_CLOCKGATE_FIX)
)
)
{
if (regPMC == 0)
{
gcmkONERROR(
gckOS_ReadRegisterEx(Hardware->os,
Hardware->core,
```

ERR007805: I2C: When the I2C clock speed is configured for 400 kHz, the SCL low period violates the I2C spec of 1.3 uS min

Description

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

Workaround

To meet the clock low period requirement in fast speed mode, SCL must be configured to 384KHz or less.

ERR050045: IOMUX: Setting ODE control bit of I2C IOs causes malfunction

Description

The I2C module supports open drain. The I2C module drives the open-drain signal of the output data. However, setting the ODE bit in the I2C IOMUXC registers results in malfunctions due to internal logic.

Workaround

Do not set the ODE bit in the I2C IOMUX registers because I2C module already supports open drain.

ERR051123: ISI: Memory overwrite occurring outside of allocated buffer space corrupting system memory**Description**

Under marginal timing conditions, when an incomplete frame is received, resulting in an early or late VSYNCH error, it is possible for the ISI to overwrite system memory outside its allocated buffer space, resulting in unpredictable behavior.

Workaround

Allocate a large capture buffer as a margin for frames, e.g., width × height × (bpp/8) × 4, to ensure sufficient space.

ERR051182: ISI: U and V colors are reversed when horizontal flip is enabled in ISI configuration**Description**

When ISI horizontal flip is enabled in YUYV mode, colors are wrong because U and V are reversed.

Workaround

Do not use ISI horizontal flip. If horizontal flip is required, use sensor or G2D library to perform flip.

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.

ERR050350: ROM: Exception raised when ROM accesses a reserved region when Field Return fuse is enabled**Description**

A Slave Error exception (SLVERR) is generated by the hardware when attempting to access a reserved region in field return mode. Because the exception interrupt is masked in the ROM the pending error defers to the bootloader and will cause the system to hang if not handled correctly.

Workaround

Because the Arm architecture does not provide any mechanisms to clear pending exceptions before they are taken, users must correctly handle the exception. Hence the initial bootloader is required to handle the exception generated by the core before proceeding further execution. The NXP provided SPL bootloader handles this exception.

Below are the steps a typical application may have to handle for the SError exception:

1. Add an exception handler for system error (SError) in the vector table. The offset 0x380 of the vector table should be used to set the handler for SError exception. Exceptions raised while rom is in execution happens at exception Level 3 (EL3).
2. Clear the SError exception mask. By default, ROM masks the asynchronous exceptions. The mask bit should be cleared at an early stage of application startup code so that exception handler routine gets executed to take the exception.
3. The exception handler routine should return for the very first SError exception if the field return fuse bit is blown, this condition guarantees that the exception is raised due to ROM execution. For all other conditions, application should handle the exception as the case may be.

ERR050359: ROM: USB Serial Download mode supports maximum 3 devices per USB host

Description

The ROM USB HID driver supports up to a maximum of 3 devices per host for simultaneous download, when the SOC is in Serial Download mode.

Workaround

Serial Download Mode allows up to 3 devices to be connected to a single USB host channel. If more than 3 devices are required, the user must have additional USB host channels available. Limit the number of devices simultaneously connected on PC to 3 per USB host.

ERR050144: SAI: Setting FCONT=1 when TMR>0 may not function correctly

Description

When FCONT=1 the transmitter will recover after a FIFO error when the FIFO is no longer empty and starting again from the same word in the following frame where the error occurred.

Configuring TMR > 0 will configure one or more words in the frame to be masked (nothing transmitted during that slot). If anything other than the last word(s) in the frame are masked when FCONT=1 and a FIFO Error Flag is set, then the transmitter will not recover and will set FIFO Error Flag during each frame.

Workaround

To avoid this issue, set FCONT in TCR4 to be 0.

ERR050542: SAI: The Bit Count Timestamp Register (TBCTR, RBCTR) may return a live rather than latched Timestamp**Description**

A SAI Timestamp Counter instance implements independent 32-bit counters for BCLK and a Timestamp based on the sub-system clock (AUDIO_AHB_CLK_ROOT, typically 400MHz). The current value of the timestamp count is latched on a BCLK edge and the contents of that latch is further latched into the xBCTR register whenever the BCLK count is read (xBCCR). However, reading xBCCR sometimes results in xBCTR latching the current value of the timestamp count, not the value latched on the most recent BCLK edge. This introduces uncertainty in the timestamp of up to 1 BCLK period.

Workaround

A BCLK period is sampling frequency and format dependent e.g. 142 sub-system clocks for 44.1kHz I2S or 33 sub-system clocks for 48kHz TDM8. These represent 3.5ppm or 825ppb respectively when measuring at 10Hz, compared to the 25ppb design aim. The uncertainty in the timestamp is instantaneous not accumulating and should be considered when designing any PLL or ASRC correction.

ERR050362: TCM: AXI2AHB cannot handle partial write and causes redundant write operations to TCM**Description**

The AXI2AHB bridge is used to access the TCM in CM7. On the AXI side, the bus data width is 64-bit with 8-bit write strobe signals.

For normal write operations at least one bit of the WSTB signal is asserted on the AXI side, therefore, AHB can handle the related write operations correctly.

For burst write operations there are no write strobe signals ASSERTED for some beats such as in the SDMA case. In this case, the first beats with write strobe asserted the write operations from AXI to AHB are correct; however, for the remaining beats without write strobe asserted in AXI (WSTB=0x00), AHB will write data from the invalid beats to TCM and cause error.

This issue impacts all masters that access TCM through system bus.

Workaround

Set bit-1 of register 0x32504044 to 1.

This enables the bridge to correctly handle the situation when there is write request but no write strobe is asserted in AXI side.

ERR051272: TMU: Bit 31 of registers TMU_TSCR/TMU_TRITSR/TMU_TRATSR invalid**Description**

Bit 31 of registers TMU_TSCR/TMU_TRITSR/TMU_TRATSR might be set as invalid value when the temperature varies in range.

Workaround

Do not use Bit 31 of registers TMU_TSCR/TMU_TRITSR/TMU_TRATSR. Suggest to read TMU value and use 1 point calibration to justify if the temperature is in range.

NXP Linux BSP does not use those bits since rel_imx_4.14.98_2.0.0_ga.

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.

Limiting values — Stress above one or more limiting values (as defined in the Absolute Maximum Ratings System of IEC 60134) will cause permanent damage to the device. Limiting values are stress ratings only and (proper) operation of the device at these or any other conditions above those given in the Recommended operating conditions section (if present) or the Characteristics sections of this document is not warranted. Constant or repeated exposure to limiting values will permanently and irreversibly affect the quality and reliability of the device.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at <https://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.

No offer to sell or license — Nothing in this document may be interpreted or construed as an offer to sell products that is open for acceptance or the grant, conveyance or implication of any license under any copyrights, patents or other industrial or intellectual property rights.

Quick reference data — The Quick reference data is an extract of the product data given in the Limiting values and Characteristics sections of this document, and as such is not complete, exhaustive or legally binding.

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.

Suitability for use in non-automotive qualified products — Unless this document expressly states that this specific NXP Semiconductors product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. NXP Semiconductors accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without NXP Semiconductors' warranty of the product for such automotive applications, use and specifications, and (b) whenever customer uses the product for automotive applications beyond NXP Semiconductors' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies NXP Semiconductors for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond NXP Semiconductors' standard warranty and NXP Semiconductors' product specifications.

HTML publications — An HTML version, if available, of this document is provided as a courtesy. Definitive information is contained in the applicable document in PDF format. If there is a discrepancy between the HTML document and the PDF document, the PDF document has priority.

Translations — A non-English (translated) version of a document, including the legal information in that document, is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Security — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately.

Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP.

NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

NXP B.V. — NXP B.V. is not an operating company and it does not distribute or sell products.

Trademarks

Notice: All referenced brands, product names, service names, and trademarks are the property of their respective owners.

NXP — wordmark and logo are trademarks of NXP B.V.

Contents

1 Mask Set Errata for Mask 0N14Y 1

1.1 Revision History 1

1.2 Errata and Information Summary 2

2 Known Errata 3

Legal information 13

Please be aware that important notices concerning this document and the product(s) described herein, have been included in section 'Legal information'.