Errata

IWRL6432AOP Device Silicon Errata

TEXAS INSTRUMENTS

ABSTRACT

This document describes the known exceptions to the functional specifications (advisories). This document may also contain usage notes. Usage notes describe situations where the device's behavior may not match the presumed or documented behavior. This may include behaviors that affect device performance or functional correctness.

Table of Contents

1 Introduction

This document describes the known exceptions to the functional and performance specifications to TI CMOS Radar Devices (IWRL6432AOP)

2 Device Nomenclature

To designate the stages in the product development cycle, TI assigns prefixes to the part numbers of Radar / mmWave sensor devices. Each of the Radar devices has one of the two prefixes: XIx or IWRLx (for example: **XI6432** QGAMY). These prefixes represent evolutionary stages of product development from engineering prototypes (XI) through fully qualified production devices (IWRL).

Device development evolutionary flow:

XI — Experimental device that is not necessarily representative of the final device's electrical specifications and may not use production assembly flow.

IWRL — Production version of the silicon die that is fully qualified.

XI devices are shipped with the following disclaimer:

"Developmental product is intended for internal evaluation purposes."

Texas Instruments recommends that these devices not to be used in any production system as their expected end –use failure rate is still undefined.

3 Device Markings

Figure 3-1 shows an example of the IWRL6432AOP Radar Device's package symbolization.

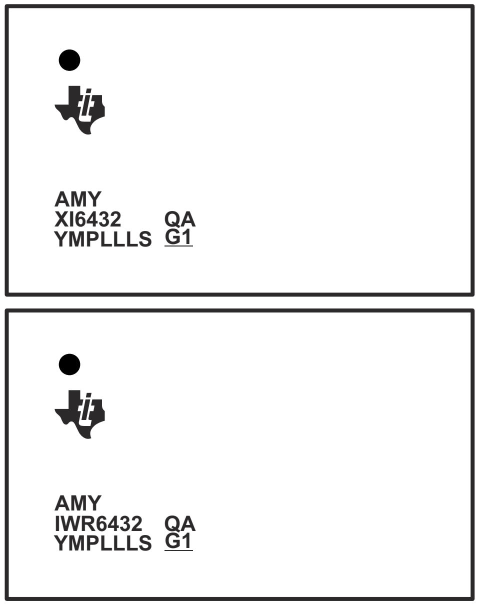

Figure 3-1. Example of Device Part Markings

This identifying number contains the following information:

- **Line 1:** TI Logo
- **Line 1:** AMY = Package Identifier
- **Line 2:**Device Number
- **Line 2:** Safety Level and Security Grade
  - Q = Non-Functional Safety Compliant
  - A = Authenticated Boot
- **Line 3:** Lot Trace Code
  - YM = Year/Month Code
  - P = Secondary Site Code
  - LLL = Assembly Lot
  - S = Primary Site Code
  - G1 = "Green" Package Build (must be underlined)

4 Usage Notes

4.1 Power up sequence in power optimized topology

When the device is in power optimized topology and 1.8V powers up before 1.2V, a momentary large current can be seen at 1.2V rail before NRESET release only during the device power up. To avoid that, power up 1.2V (VDDIN, VDD_SRAM and VNWA) before 1.8V .

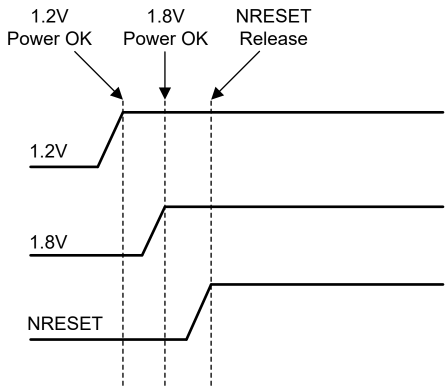

Figure 4-1. Recommended power up sequence

Note

1. This usage note is ONLY applicable for 1.2V rail. For signals other than 1.2V power supply, refer the "Device Wake-up Sequence" diagram in the data sheet.
2. The power distribution network in TI EVM/Reference designs follows "Device Wake-up Sequence" in the data sheet and supports the additional current without powering up 1.2V first.

4.2 Meeting data sheet spec for 1.2V Digital LDO output path in BOM optimized topology

As indicated in the device data sheet,1.2V Digital LDO requires one decoupling capacitor with a typical value of 4.7uF. The parasitic offered by different portion of the output path is illustrated in Figure 4-2. “RT1” and “RT2” are parasitic resistances offered by the ball to capacitor lead trace and the ground trace, respectively. Similarly, “LT1” and “LT2” are parasitic inductances offered by the ball to capacitor lead trace and the ground trace, respectively. “ESL1” and “ESR1” are the effective series inductance and resistance of the decoupling capacitor. Table 4-1 gives the minimum, maximum and typical values of the capacitance and the parasitic.

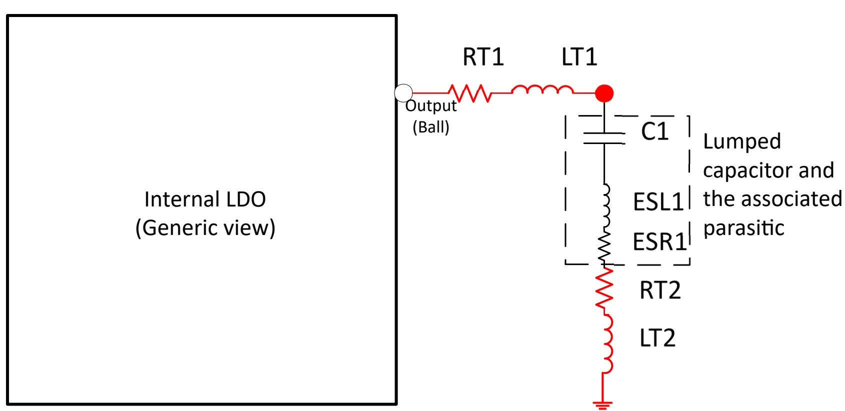

Figure 4-2. Single decoupling capacitor associated PCB parasitic

Table 4-1. 1.2V Dig LDO Output

	Min	Typ	Max	Unit
Recommended value of C	3.6	4.7	5.2	uF

Table 4-1. 1.2V Dig LDO Output (continued)

	Min	Typ	Max	Unit
Allowed output parasitic inductance Lp	1	1.5	2	nH
Allowed output parasitic resistance Rp	**15**	**20**	**35**	**mOhm**

If the R_P is not achievable in PCB design, below workarounds can be considered:

Add a series precision resistor in the VDD output path along with the decoupling capacitor to get equivalent effective series resistance of the output path within the data sheet limits.

Example part no.: PA0402FRL570R02L, 20mOhm, 1/3W; MFL0603R0200FA, 20mOhm, 1/4W

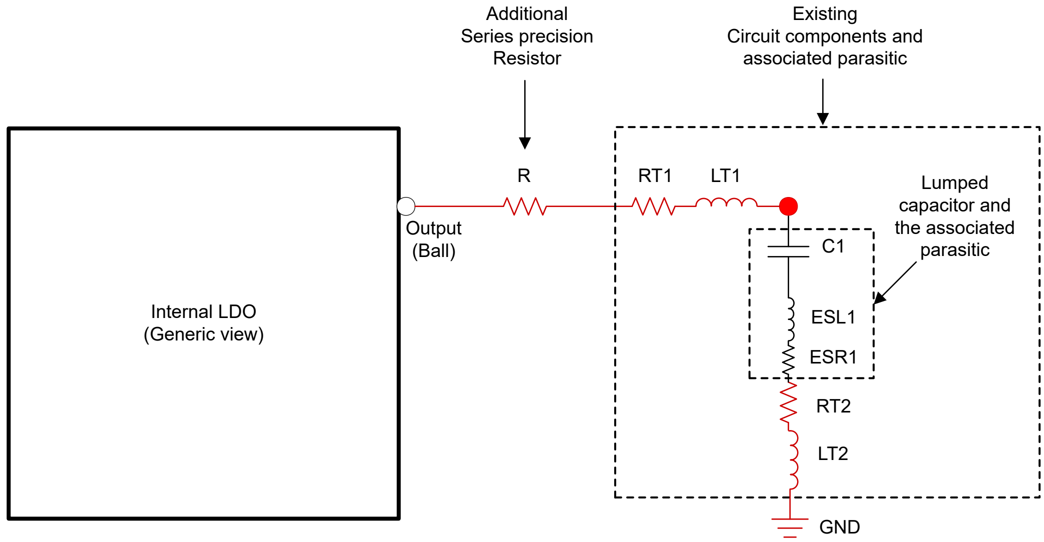

Figure 4-3. Example workaround

5 Advisory to Silicon Variant / Revision Map

Table 5-1. Advisory to Silicon Variant / Revision Map

Advisory Number	Advisory Title	IWRL6432AOP
		ES2.0
Analog / Millimeter Wave		
ANA #51	Continuous Wave Streaming CZ mode: Sudden jump in RX output codes every 20.97152msec	x
ANA#54	RX ADC saturation risk because of TX to RX coupling	x
ANA#57	SNR degradation at 60GHz in the presence of strong near range reflector	x
Digital Subsystem		
DIG #1	ePWM: Glitch during Chopper mode of operation	x
DIG #3	Limited UART baud rates	x
DIG #4	RS232 AutoBaud Rate feature doesn't support trimmed ROCSC variation.	x
DIG #5	Internal Bus access to SPI for data transfer not supported when SPI smart-idle mode is enabled.	x
DIG #6	CRC: CRC 8-bit data width and CRC8-SAE-J1850 and CRC8-H2F possible use in CAN module is not supported	x
DIG #8	Shared RAM clock gating default values	x
DIG #9	TOP_IO_MUX register space not accessible from RS232 for debug purposes.	x
DIG #10	Incorrect behavior of frame stop API	x
DIG #14	Corrupted Data Store for Partial Write in Shared Memory	x
DIG #15	Boot failure, if metaimage is multiple of 2K	x
DIG #16	Boot failure for images less than size 8k over SPI	x

6 Known Design Exceptions to Functional Specifications

ANA #51 ***Continuous Wave Streaming CZ mode: Sudden jump in RX output codes every 20.97152 msec***

Revisions Affected

IWRL6432AOP ES2.0

Details

On Continuous Wave Streaming CZ mode, the Rx data shows a sudden jump in output codes every 20.97152 milliseconds.

This is not an issue in the Radar Functional mode when chirps are used. However, this issue is seen when testing Rx chain in lab using continuous stream mode.

Workaround

To use Continuous stream (CW) mode for testing, TI recommends to start data capturing from the first sample to make sure the glitch occurs at deterministic samples. Please follow the below sequence to achieve this:

- Configure the RDIF (Radar Data Interface)
- Arm the DCA1000 (Data capture card)
- Enable the continuous stream mode.

The glitch will not be seen with this sequence. For example, if the user analyzes first 20ms of data or between 21 and 41ms.

ANA#54 — *RX ADC saturation risk because of TX to RX coupling*

Revision(s) Affected: IWRL6432AOP ES2.0

Description: There is limited isolation between Tx and Rx on this device which can cause ADC saturation depending on Tx power backoff, Rx gain setting, chirp slope and HPF cutoff frequency configurations.

Workaround(s): Please refer to Tx back off and Rx gain recommendation for xWRL6432AOP to avoid ADC saturation.

ANA #57 — *SNR degradation at 60GHz in the presence of strong near range reflector*

Revision(s) Affected IWRL6432AOP ES2.0

Details There is a non-linearity of the synthesizer when crossing 60GHz which causes increased noise floor at RX output in the presence of a strong near range reflector.

Workaround Chirps with large RF bandwidth (> 1.5GHz) have negligible noise floor impact. For lower bandwidth chirps, avoid 60GHz.

DIG #1 — *ePWM: Glitch during Chopper mode of operation*

Revisions Affected

IWRL6432AOP ES2.0

Details

During chopper mode operation, a glitch can be observed on the ePWMA and ePWMB output signals from the ePWM module.

Workaround

If the use case is impacted by a glitch, TI recommends to disable the PWM chopper control function by setting the LPRADAR:APP_PWM:PCCTL:CHPEN register bit to 0.

The below table shows the Register Address for above workaround.

Bits	Name	Address
0	LPRADAR:APP_PWM:PCCTL:CHPEN	0X57F7 FC3C

DIG #3 *UART: Limited UART baud rates*

Revisions Affected

IWRL6432AOP ES2.0

Details

Due to a design limitation (related to the clocking scheme), UART doesn't support standard baud rates above 115200 bits per second. Higher baud rates up to 1.25Mbps can be supported but the rates are non-standard.

Applications requiring UART cannot use standard baud rates above 115200 bits per second

Standard Baud Rates supported :

XTAL (MHz)	40	
Ideal Baud rate (bps)	Actual Baud	Error %
115200	113636.36	1.36
76800	75757.58	1.36

Non- Standard baud rates supported:

XTAL (MHz)	**40**
Maximum baud (bps)	1250k
	833.33k
	625k
	500k
	416.66k
	357.14k
	312.5k

Workaround

TI recommends to use the following workarounds based on application needs:

- Use of non-standard baud rates can provide up to 1.25Mbps throughput, if external MCU can support the same non-standard baud rates.
- Use SPI instead, if use-case needs higher throughput.

DIG #4 *RS232: Auto Baud Rate feature doesn't support trimmed RCOSC variation*

Revisions Affected

IWRL6432AOP ES2.0

Details

Once RCOSC is trimmed, the expected clock frequency and the variation observed in frequency (tolerance on RC clock) do not support the required Auto Baud rate setting for RS232.

Currently Auto Baud is disabled by default for ES2.0

DIG #5 ***Internal Bus access to SPI for data transfer not supported when SPI smart-idle mode is enabled.***

Revisions Affected

IWRL6432AOP ES2.0

Details

Smart-idle mode needs to be disabled for SPI before the first trigger for data transfer access. If the SPI smart-idle mode is required to be enabled, the mode has to be enabled again once the access is complete.

Workaround

TI recommends to follow the below sequence:

Auto Wake-up = 1 & Controller mode

1. Configure McSPI as required
2. Enable SmartIdle (by setting LPRADAR:APP_CTRL:SPI1_SMART_IDLE_ENABLE for SPI1 and LPRADAR:APP_CTRL:SPI2_SMART_IDLE_ENABLE for SPI 2)after making sure that there is **no** pending transaction from/to SPI or any more access to be done to McSPI by CPU or DMA
3. If any register or memory access to McSPI has to be done, disable SmartIDLE mode (by setting LPRADAR:APP_CTRL:SPI1_SMART_IDLE_ENABLE=0 for SPI 1 and LPRADAR:APP_CTRL:SPI2_SMART_IDLE_ENABLE =0 for SPI 2)
4. In Controller mode, the external host is not going to toggle the SPI_CS, hence there is not any wakeup => there is no difference between (LPRADAR:APP_CTRL:SPI1_SMART_IDLE_AUTO_EN is 1 or 0 for SPI 1 and LPRADAR:APP_CTRL:SPI2_SMART_IDLE_AUTO_EN is 1 or 0)

Auto Wake-up = 1 & Peripheral mode

1. Configure McSPI as required
2. Enable SmartIdle (by setting LPRADAR:APP_CTRL:SPI1_SMART_IDLE_ENABLE for SPI1 and LPRADAR:APP_CTRL:SPI2_SMART_IDLE_ENABLE for SPI 2) after making sure that there is **no** pending transaction from/to SPI or any more access to be done to McSPI by CPU or DMA
3. If any register or memory access to McSPI has to be done by any controller (DMA / CPU), disable SmartIDLE mode (by setting LPRADAR:APP_CTRL:SPI1_SMART_IDLE_ENABLE=0 for SPI 1 and LPRADAR:APP_CTRL:SPI2_SMART_IDLE_ENABLE =0 for SPI 2)
4. If there is wakeup from McSPI (because of some SPI_CS toggle), then the clock is automatically enabled.
5. Disable SmartIdle configuration (by setting LPRADAR:APP_CTRL:SPI1_SMART_IDLE_ENABLE=0 for SPI 1 and LPRADAR:APP_CTRL:SPI2_SMART_IDLE_ENABLE =0 for SPI 2) to do the register access.

The below table shows the Register Addresses for above workaround.

Bits	Name	Address
0	LPRADAR:APP_CTRL:SPI1_SMART_IDLE_ENABLE	0x560603A8
2	LPRADAR:APP_CTRL:SPI1_SMART_IDLE_AUTO_EN	0x560603A8
0	LPRADAR:APP_CTRL:SPI2_SMART_IDLE_ENABLE	0x560603AC

DIG #5 (continued) ***Internal Bus access to SPI for data transfer not supported when SPI smart-idle mode is enabled.***

2	LPRADAR:APP_CTRL:SPI2_SMART_IDLE_AUTO_EN	0x560603AC

DIG #6

CRC: CRC 8-bit data width and CRC8-SAE-J1850 and CRC8-H2F possible use in CAN module is not supported

Revisions Affected

IWRL6432AOP ES2.0

Details

1. 8-bit data width is not supported. Minimum data width supported is 16-bit.
2. CRC types CRC8-SAE-J1850 and CRC8-H2F are not supported.

Workaround

1. 16/32/64-bit data widths are supported.
2. TI recommends to not use the above mentioned unsupported polynomials.

DIG #8 *Shared RAM clock gating default values*

Revision(s) Affected IWRL6432AOPES2.0

Details

Possibility of Shared RAM data corruption while exiting from deep sleep mode when clock gating registers are not reprogrammed.

The reset value for Front End Controller Sub System (FECSS), Application Sub System (APPSS) and Hardware Accelerator Sub System (HWASS) shared memory clock gate control is 1 . The clock ICG controls are coming from the following registers.

Bits	Name	Address
0	LPRADAR:FEC_CTRL:FECSS_SHARED_MEM_CLK_GATE : FECSS_SHARED_MEM_CLK_GATE_HWA_ENABLE	0x5200002C
0	LPRADAR:APP_CTRL:APPSS_SHARED_MEM_CLK_GATE:APPSS_SHARED_MEM_CLK_GATE_MEM0_HWA_ENABLE	0x56060398
2	LPRADAR:APP_CTRL:APPSS_SHARED_MEM_CLK_GATE:APPSS_SHARED_MEM_CLK_GATE_MEM1_HWA_ENABLE	0x56060398

When APPSS tries to access shared memory bank 0 via VBUSM SCR while FECSS is accessing shared memory via AHB, wrong read values of zero from the shared RAM on the APPSS is observed .

If only one of the clock gates (either HWA or FEC/APP) is enabled based on the allocation, the data is read correctly. Since the clock gating controls are coming from control registers space, these values get reset again and hence needs to be re-programmed after every deep sleep exit.

Workaround

Program ICG controls of clock reaching to shared memory based on different shared memory configuration. The ICG control needs to be re-programmed after every deep sleep exit too.

Configuration	Software care-about
Memory is shared with M3	Disable the following ICG control :- LPRADAR:FEC_CTRL:FECSS_SHARED_MEM_CLK_GATE : FECSS_SHARED_MEM_CLK_GATE_HWA_ENABLE
First 128kb is shared with M4	Disable the following ICG control :- LPRADAR:APP_CTRL:APPSS_SHARED_MEM_CLK_GATE:APPSS_SHARED_MEM_CLK_GATE_MEM0_HWA_ENABLE

DIG #8 (continued) ***Shared RAM clock gating default values***

256kb is shared with M4	Disable the following ICG controls :- • LPRADAR:APP_CTRL:APPSS_SHARED_MEM_CLK_GATE:APPSS_SHARED_MEM_CLK_GATE_MEM0_HWA_ENABLE • LPRADAR:APP_CTRL:APPSS_SHARED_MEM_CLK_GATE:APPSS_SHARED_MEM_CLK_GATE_MEM1_HWA_ENABLE

DIG #9 — *TOP_IO_MUX register space not accessible from RS232 for debug purposes*

Revisions Affected IWRL6432AOPES2.0

Details

RS232 is not able to write TOP_IO_MUX registers unless the space is programmed for user mode access.

Workaround

TI recommends to use the following sequence:

1. From Processor or DAP : Unlock TOP_IO_MUX registers (by programming LPRADAR:TOP_IO_MUX:IOCFGKICK0 = 83E7 0B13h and LPRADAR:TOP_IO_MUX:IOCFGKICK1 = 95A4 F1E0h)
2. From Processor or DAP : Write to TOP_IO_MUX registers, LPRADAR:TOP_IO_MUX:USERMODEEN should be set to 0xADADADAD
3. Now TOP_IO_MUX registers can be accessed from RS232.

The below table shows the Register Addresses for above workaround.

Bits	Name	Address
0:31	LPRADAR:TOP_IO_MUX:IOCFGKICK0	0x5A000068
0:31	LPRADAR:TOP_IO_MUX:IOCFGKICK	0x5A00006C
0:31	LPRADAR:TOP_IO_MUX:USERMODEEN	0x5A000060

DIG #10 *Incorrect behavior of frame stop API*

Revisions Affected IWRL6432AOPES2.0

Details

The Frame Timer latches Frame Stop command in hardware registers which takes affect at the end of current frame. Frame Stop API issued when Frame Timer has already stopped results in unintended stop in the next frame trigger because of the latched stop bit.

Workaround

1. Avoid unnecessary Sensor Stop API should be avoided
2. The application can have to wait for one complete frame period before getting frame stop.
3. Application waits for FECSS to complete Burst End and Frame End activities after receiving the Frame Stop conformation.

DIG #14 — *Corrupted Data Store for Partial Write in Shared Memory*

Revision(s) Affected

IWRL6432AOP ES2.0

Details

Internal shared memory has ODD and EVEN banking structure. For a particular address range, partial write (less than 32 bit) to EVEN bank corrupts same address of ODD bank with next data on the bus. When shared memory is allocated to M4/M3, back to back full word write access to location A followed by sub-word write access to location B corrupts data in location A.

When memory is shared with M4/M3, issue is seen in the following address range:

Memory	Address Range
APP_CPU_SHARED_RAM	0x0048 0000 - 0x004B FFFC
FEC_CPU_SHARED_RAM	0x2120 8000 - 0x2121 FFFC

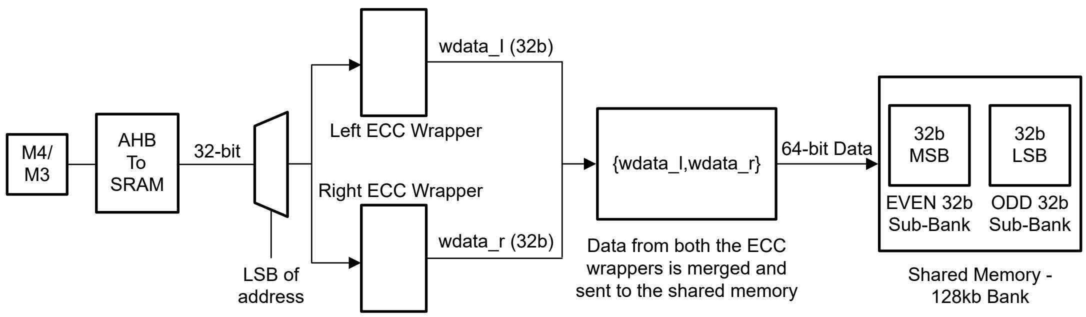

Figure 6-1. Shared Memory Logic Diagram When Shared with M4/M3

When shared with M3/M4, the incoming data bit width is 32 bit as shown in the diagram. So, depending on LSB of address, signals are sent to either left or right ECC wrapper.

Workaround

1. Use shared memories as code memory when shared with processor.
2. Disable ECC for non functional safety devices – ECC is disabled for shared memories in RBL for non functional safety devices.

DIG #15 *Boot failure, if metaimage is multiple of 2K*

Revisions Affected

IWRL6432AOP ES2.0

Details

Metaimages that are a multiple of 2048 bytes will fail to boot.

Workaround

1. Add a constant non-volatile variable to increase the metaimage size, so that the size is not aligned to 2048 bytes.
2. Update MMWAVE-L-SDK to version 5.4 or above; mmWave LSDK 5.4 and above includes changes to the metaimage generator tool to add a minimal config file (approximately 64 bytes) in case the image is a multiple of 2048 bytes.

DIG #16 — *Boot failure for images less than size 8k over SPI*

Revisions Affected

IWRL6432AOP ES2.0

Details

The EDMA address linking is not done in few cases (during SPI continuous download), due to which boot fails over SPI continuous download image for the particular metaimage size ranges mentioned in the below table:

Image Size (Bytes)	Issue Present
<2048	No
>2048 & <4096	No
>=4096 & <6144	Yes
>=6144 & <8192	Yes
>=8192	No

Workaround

Use image >8KB for boot over SPI. In case of lower image size, constant data is appended during compile time to create an image >8 KB.

7 Trademarks

All trademarks are the property of their respective owners.

Revision History

Changes from April 30, 2025 to December 15, 2025 (from Revision A (April 2025) to Revision B (December 2025)) **Page**

- (Usage note): Section added.......... 4
- *Corrupted Data Store for Partial Write in Shared Memory*: Added a new Advisory “Shared memory: Corrupted Data Store for Partial Write in Shared Memory"........ 19