1 Introduction
This document describes the known exceptions to the functional and performance specifications to TI CMOS Radar Devices (AWR1843).

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: X1x or AWR1x (for example: X1642BIGABL). These prefixes represent evolutionary stages of product development from engineering prototypes (X1x) through fully qualified production devices (AWR1x).

Device development evolutionary flow:

X1x — Experimental device that is not necessarily representative of the final device’s electrical specifications and may not use production assembly flow.

AWR1x — Production version of the silicon die that is fully qualified.

X1x 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.
Device Markings

Figure 1 shows an example of the AWR1843 Radar Device's package symbolization.

![Device Markings](image)

**Figure 1. Example of Device Part Markings**

This identifying number contains the following information:

- **Line 1**: Device Number
- **Line 2**: Temperature and Security Grade
- **Line 3**: Lot Trace Code
  - YM = Year/Month Code
  - PLLL = Assembly Lot
  - S = Assembly Site Code
- **Line 4**:
  - 502 = AWR1843 Identifier
  - ABL = Package Identifier
  - G1 = "Green" Package Build (must be underlined)
4 Usage Notes

Usage notes highlight and describe particular situations where the device's behavior may not match presumed or documented behavior. This may include behaviors that affect device performance or functional correctness. These usage notes will be incorporated into future documentation updates for the device (such as the device-specific data sheet), and the behaviors they describe will not be altered in future silicon revisions.

4.1 MSS: SPI Speed in 3-Wire Mode Usage Note

The maximum SPI speed under 3-wire operation was only tested up to 33 MHz. This affects AWR1843 ES1.0.
### Table 1. Advisory to Silicon Variant / Revision Map

<table>
<thead>
<tr>
<th>Advisory Number</th>
<th>Advisory Title</th>
<th>AWR18xx</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSS#03</td>
<td>Incorrect Handling of “Saturation” in FFT Hardware Accelerator’s Input / Output Formatter and Statistics Block</td>
<td>X</td>
</tr>
<tr>
<td>MSS#13</td>
<td>Incorrect Read from FFT Hardware Accelerator After Complex Multiplication Operation</td>
<td>X</td>
</tr>
<tr>
<td>MSS#17</td>
<td>Invalid Pre-fetch from MSS CR4 Processor (due to Speculative Read Operation from Tightly Coupled Memory Instance) Leads to Generation of MSS_ESM Group 3 Channel 7: MSS_TCMA_FATAL_ERR</td>
<td>X</td>
</tr>
<tr>
<td>MSS#18</td>
<td>Core Compare Module (CCM-R4F) may Cause nERROR Toggle After First Reset De-assertion Subsequent to Power Application</td>
<td>X</td>
</tr>
<tr>
<td>MSS#19</td>
<td>DMA Read from Unimplemented Address Space may Result in DMA Hang Scenario</td>
<td>X</td>
</tr>
<tr>
<td>MSS#21</td>
<td>Issue with HWA Input Formatter 16 bit Real Signed Format</td>
<td>X</td>
</tr>
<tr>
<td>MSS#22</td>
<td>CAN-FD: Message Transmitted With Wrong Arbitration and Control Fields</td>
<td>X</td>
</tr>
<tr>
<td>MSS#23</td>
<td>HWA Read Registers Cannot be Read Reliably When the HWA is Executing a ParamSet Instruction</td>
<td>X</td>
</tr>
<tr>
<td>MSS#24</td>
<td>Limitation With Peak Grouping Feature in Hardware Accelerator</td>
<td>X</td>
</tr>
<tr>
<td>MSS#25</td>
<td>Debugger May Display Unpredictable Data in the Memory Browser Window if a System Reset Occurs</td>
<td>X</td>
</tr>
<tr>
<td>MSS#26</td>
<td>DMA Requests Lost During Suspend Mode</td>
<td>X</td>
</tr>
<tr>
<td>MSS#27</td>
<td>MibSPI in Slave Mode in 3- or 4-Pin Communication Transmits Data Incorrectly for Slow SPICLK Frequencies and for Clock Phase = 1</td>
<td>X</td>
</tr>
<tr>
<td>MSS#28</td>
<td>A Data Length Error is Generated Repeatedly in Slave Mode When IO Loopback is Enabled</td>
<td>X</td>
</tr>
<tr>
<td>MSS#29</td>
<td>Spurious RX DMA REQ From a Slave Mode MibSPI</td>
<td>X</td>
</tr>
<tr>
<td>MSS#30</td>
<td>MibSPI RX RAM RXEMPTY Bit Does Not Get Cleared After Reading</td>
<td>X</td>
</tr>
<tr>
<td>MSS#31</td>
<td>CPU Abort Generated on a Write to Implemented CRC Space After a Write to Unimplemented CRC Space</td>
<td>X</td>
</tr>
<tr>
<td>MSS#32</td>
<td>DMMGLBCTRL BUSY Flag Not Set When DMM Starts Receiving A Packet</td>
<td>X</td>
</tr>
<tr>
<td>MSS#33</td>
<td>MibSPI RAM ECC is Not Read Correctly in DIAG Mode</td>
<td>X</td>
</tr>
<tr>
<td>MSS#34</td>
<td>HS Device Does Not Reboot Successfully on Warm Reset Getting Triggered by Watchdog Expiry</td>
<td>X</td>
</tr>
<tr>
<td>MSS#35</td>
<td>EDMA TPTC Generates an Incorrect Address on the Read Interface, Causing one or More Data Integrity Failures, Hangs, or Extra Reads</td>
<td>X</td>
</tr>
<tr>
<td>ANA#08</td>
<td>Doppler Spur Observed for Narrow Chirps Spanning 79.2 GHz</td>
<td>X</td>
</tr>
<tr>
<td>ANA#09</td>
<td>Synthesizer Frequency Nonlinearity at 76.8 GHz when Synthesizer (Chirp) Frequency Monitor Enabled</td>
<td>X</td>
</tr>
<tr>
<td>ANA#10</td>
<td>Unreliable Readings from Synthesizer Supply Voltage Monitor</td>
<td>X</td>
</tr>
<tr>
<td>ANA#11</td>
<td>TX, RX Gain Calibrations Sensitive to Large External Interference</td>
<td>X</td>
</tr>
<tr>
<td>ANA#12</td>
<td>Second Harmonic (HD2) is Present When Receiver is Tested Standalone Using CW Input</td>
<td>X</td>
</tr>
<tr>
<td>ANA#13</td>
<td>TX1 to TX3 Phase Mismatch Variation over Temperature is Double that of TX2/TX1 and TX3/TX2 Combinations</td>
<td>X</td>
</tr>
</tbody>
</table>
## Known Design Exceptions to Functional Specifications

### Table 2. Advisory List

<table>
<thead>
<tr>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSS#03 — Incorrect Handling of “Saturation” in FFT Hardware Accelerator’s Input / Output Formatter and Statistics Block</td>
<td>6</td>
</tr>
<tr>
<td>MSS#13 — Incorrect Read from FFT Hardware Accelerator After Complex Multiplication Operation</td>
<td>6</td>
</tr>
<tr>
<td>MSS#17 — Invalid Pre-fetch from MSS CR4 Processor (due to Speculative Read Operation from Tightly Coupled Memory Instance) Leads to Generation of MSS_ESM Group 3 Channel 7: MSS_TCMA_FATAL_ERR</td>
<td>7</td>
</tr>
<tr>
<td>MSS#18 — Core Compare Module (CCM-R4F) may Cause nERROR Toggle After First Reset De-assertion Subsequent to Power Application</td>
<td>7</td>
</tr>
<tr>
<td>MSS#19 — DMA Read from Unimplemented Address Space may Result in DMA Hang Scenario</td>
<td>7</td>
</tr>
<tr>
<td>MSS#21 — Issue with HWA Input Formatter 16 bit Real Signed Format</td>
<td>8</td>
</tr>
<tr>
<td>MSS#22 — CAN-FD: Message Transmitted With Wrong Arbitration and Control Fields</td>
<td>9</td>
</tr>
<tr>
<td>MSS#23 — HWA Read Registers Cannot be Read Reliably When the HWA is Executing a ParamSet Instruction</td>
<td>10</td>
</tr>
<tr>
<td>MSS#24 — Limitation With Peak Grouping Feature in Hardware Accelerator</td>
<td>11</td>
</tr>
<tr>
<td>MSS#25 — Debugger May Display Unpredictable Data in the Memory Browser Window if a System Reset Occurs</td>
<td>12</td>
</tr>
<tr>
<td>MSS#26 — DMA Requests Lost During Suspend Mode</td>
<td>13</td>
</tr>
<tr>
<td>MSS#27 — MibSPI in Slave Mode in 3- or 4-Pin Communication Transmits Data Incorrectly for Slow SPICLK Frequencies and for Clock Phase = 1</td>
<td>14</td>
</tr>
<tr>
<td>MSS#28 — A Data Length Error is Generated Repeatedly in Slave Mode When IO Loopback is Enabled</td>
<td>15</td>
</tr>
<tr>
<td>MSS#29 — Spurious RX DMA REQ From a Slave Mode MibSPI</td>
<td>16</td>
</tr>
<tr>
<td>MSS#30 — MibSPI RX RAM RXEMPTY bit Does Not Get Cleared After Reading</td>
<td>17</td>
</tr>
<tr>
<td>MSS#31 — CPU Abort Generated on a Write to Implemented CRC Space After a Write to Unimplemented CRC Space</td>
<td>18</td>
</tr>
<tr>
<td>MSS#32 — DMMGLBCtrl BUSY Flag Not Set When DMM Starts Receiving A Packet</td>
<td>19</td>
</tr>
<tr>
<td>MSS#33 — MibSPI RAM ECC is Not Read Correctly in DIAG Mode</td>
<td>20</td>
</tr>
<tr>
<td>MSS#34 — HS Device Does Not Reboot Successfully on Warm Reset Getting Triggered or With Internal Software Reset</td>
<td>21</td>
</tr>
<tr>
<td>MSS#35 — EDMA TPTC Generates an Incorrect Address on the Read Interface, Causing one or More Data Integrity Failures, Hangs, or Extra Reads</td>
<td>22</td>
</tr>
<tr>
<td>ANA#08 — Doppler Spur Observed for Narrow Chirps Spanning 79.2 GHz</td>
<td>27</td>
</tr>
<tr>
<td>ANA#09 — Synthesizer Frequency Nonlinearity at 76.8 GHz when Synthesizer (Chirp) Frequency Monitor Enabled</td>
<td>27</td>
</tr>
<tr>
<td>ANA#10 — Unreliable Readings from Synthesizer Supply Voltage Monitor</td>
<td>27</td>
</tr>
<tr>
<td>ANA#11 — TX, RX Gain Calibrations Sensitive to Large External Interference</td>
<td>28</td>
</tr>
<tr>
<td>ANA#12 — Second Harmonic (HD2) is Present When Receiver is Tested Standalone Using CW Input</td>
<td>28</td>
</tr>
<tr>
<td>ANA#13 — TX1 to TX3 Phase Mismatch Variation over Temperature is Double that of TX2/TX1 and TX3/TX2 Combinations</td>
<td>28</td>
</tr>
</tbody>
</table>
### MSS#03 Incorrect Handling of “Saturation” in FFT Hardware Accelerator’s Input / Output Formatter and Statistics Block

<table>
<thead>
<tr>
<th>Revision(s) Affected:</th>
<th>AWR1843 ES1.0</th>
</tr>
</thead>
</table>

**Description:**
- Input formatter block performs saturation based on signed or unsigned samples. However, the compute engine module always saturates the input as a 24 bit signed number.
- Output formatter block saturates the 16 bits unsigned number as signed number. This causes magnitude outputs to have a smaller max range.
- Statistics block always assumes that input is signed when checking for saturation, but the input can be unsigned in some cases.

**Workaround(s):** None.

### MSS#13 Incorrect Read from FFT Hardware Accelerator After Complex Multiplication Operation

<table>
<thead>
<tr>
<th>Revision(s) Affected:</th>
<th>AWR1843 ES1.0</th>
</tr>
</thead>
</table>

**Description:**
Read-back from FFT hardware accelerator slave, static configuration registers, Window RAM, Param RAM, and First stage RAM gives incorrect data if the last operation performed by the accelerator was a complex multiplication.

**Workaround(s):** None.
MSS#17  Invalid Pre-fetch from MSS CR4 Processor (due to Speculative Read Operation from Tightly Coupled Memory Instance) Leads to Generation of MSS_ESM Group 3 Channel 7: MSS_TCMA_FATAL_ERR

Revision(s) Affected: AWR1843 ES1.0

Description: The CR4 processor may perform an invalid pre-fetch access due to speculative TCM read leading to an invalid address access. This can result in a TCERROR and also a 2-bit ECC fatal error. The TCERROR is ignored by the processor since these correspond to instructions that are pre-fetched but never executed. However, the invalid MSS_TCMA_FATAL_ERR is generated on the ESM group3 channel-7.

Implication: In case of a genuine TCMA ECC fatal error, nERROR will not be generated directly through ESM.

Workaround(s): Mask Group 3 channel 7: MSS_TCMA_FATAL_ERR to ESM can be masked by writing into MSS_RCM:ESMGATE0 register. CR4F abort handler should handle the nERROR generation
OR
Disable branch prediction for MSS-CR4F

MSS#18  Core Compare Module (CCM-R4F) may Cause nERROR Toggle After First Reset De-assertion Subsequent to Power Application

Revision(s) Affected: AWR1843 ES1.0

Description: The CCM-R4F module compares the outputs of the two Cortex-R4F CPU cores and generates an error on any mis-compare. This ensures the lock-step operation of the two Cortex-R4F CPUs. The nERROR signal should only be set by the CCM-R4F module by a valid core mismatch. At power-on, some uninitialized circuits may cause the CCMR4-F to falsely detect a mis-compare.

Workaround(s): The anomalous nERROR toggle would need to be ignored by the external monitoring circuit (if deployed).

MSS#19  DMA Read from Unimplemented Address Space may Result in DMA Hang Scenario

Revision(s) Affected: AWR1843 ES1.0

Description: The MSS DMA generates a BER (Bus Error) interrupt when the DMA detects a bus error due to a read from unimplemented address space. This interrupt is available on VIM Interrupt Channel-70 for DMA1 and VIM Interrupt Channel-51 for DMA2. This read from unimplemented address space results in a hang condition in the DMA infrastructure bridge that connects it to the main interconnect.

Implication: A DMA read from an unimplemented address can result in a DMA hang condition. In the resulting state the DMA will not respond to any further DMA requests.

Workaround(s): The MSS CR4F processor will have to invoke a warm reset or generate an nERROR if it receives a DMA BER error.
<table>
<thead>
<tr>
<th>MSS#21</th>
<th><strong>Issue with HWA Input Formatter 16 bit Real Signed Format</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Revision(s) Affected:</td>
<td>AWR1843 ES1.0</td>
</tr>
<tr>
<td>Description:</td>
<td>Wrong sign extension is implemented for 16 bit signed format in real only mode operation. Hence, signed 16-bit real format cannot be supported for input formatter.</td>
</tr>
<tr>
<td>Workaround(s):</td>
<td>None, there is no planned update. The 16 bit signed format in real only mode can't be used with HWA.</td>
</tr>
</tbody>
</table>
MSS#22  CAN-FD: Message Transmitted With Wrong Arbitration and Control Fields

Revision(s) Affected:  AWR1843 ES1.0

Description:
Under the following conditions a message with wrong ID, format, and DLC is transmitted:
- M_CAN is in state “Receiver” (PSR.ACT = “10”), no pending transmission
- A new transmission is requested before the 3rd bit of Intermission is reached
- The CAN bus is sampled dominant at the third bit of Intermission which is treated as SoF (see ISO11898-1:2015 Section 10.4.2.2)

Under the conditions listed above it may happen, that:
- The shift register is not loaded with ID, format, and DLC of the requested message
- The M_CAN will start arbitration with wrong ID, format, and DLC on the next bit
- In case the ID wins arbitration, a CAN message with valid CRC is transmitted
- In case this message is acknowledged, the ID stored in the Tx Event FIFO is the ID of the requested Tx message and not the ID of the message transmitted on the CAN bus, no error is detected by the transmitting M_CAN

The erratum is limited to the case when M_CAN is in state “Receiver” (PSR.ACT = “10”) with no pending transmission and a new transmission is requested before the 3rd bit of Intermission is reached and this 3rd bit of intermission is seen dominant.

When a transmission is requested by the CPU, the Tx Message Handler performs an internal arbitration and loads the pending transmit message with the highest priority into its output buffer and then sets the transmission request for the CAN Protocol Controller. The problem occurs only when the transmission request for the CAN Protocol Controller is activated between the sample points of the 2nd and 3rd bit of Intermission and if that 3rd bit of intermission is seen dominant.

This dominant level at the 3rd bit of Intermission may result from an external disturbance or may be transmitted by another node with a significantly faster clock.

In the described case it may happen that the shift register is not loaded with arbitration and control field of the message to be transmitted. The frame is transmitted with wrong ID, format, and DLC but with the data field of the requested message. The message is transmitted in correct CAN (FD) frame format with a valid CRC.

If the message loses arbitration or is disturbed by an error, it is retransmitted with correct arbitration and control fields.

Workaround(s):
Request a new transmission only if another transmission is already pending or when the M_CAN is not in state “Receiver” (when PSR.ACT ≠ “10”).

Another option would be to add a checksum to the data field covering arbitration and control fields of the message to be transmitted.
MSS#23  **HWA Read Registers Cannot be Read Reliably When the HWA is Executing a ParamSet Instruction**

Revision(s) Affected:  AWR1843 ES1.0

Description:  Any read from the HWA configuration or status registers can be corrupted, if the read access is performed when the HWA is active. Reads from the HWA registers can be performed correctly, only after the execution of the entire ParamSet (i.e., after the ACC_DONE_INTR interrupt) or when the HWA is in IDLE mode waiting for the trigger to start the execution of the next ParamSet instruction.

Workaround(s):  Perform the following:

- Read-back of signature registers: Software needs to maintain a soft copy of the one-hot encoded signature registers and use that copy location for the EDMA programming.
- Read-back of static registers on the HWA ParamSet interrupt. There is no reliable way to read the HWA static registers, if the HWA is active.
- Read-back of Debug/status registers: The User can only read these registers when the HWA is not active.
### MSS#24  Limitation With Peak Grouping Feature in Hardware Accelerator

<table>
<thead>
<tr>
<th>Revision(s) Affected:</th>
<th>AWR1843 ES1.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description:</td>
<td>Peak is declared only if the cell under test is greater than its most immediate neighboring cells to its left and right. In the case where CFAR qualified peaks in the two adjacent cells happen to be equal in magnitude, enabling peak grouping can lead to the peak being lost.</td>
</tr>
<tr>
<td>Workaround(s):</td>
<td>Do not enable the peak grouping feature in the hardware accelerator.</td>
</tr>
</tbody>
</table>
MSS#25  Debugger May Display Unpredictable Data in the Memory Browser Window if a System Reset Occurs

Revision(s) Affected: AWR1843 ES1.0

Description: If a system reset (nRST goes low) occurs while the debugger is performing an access on the system resource using system view, a slave error should be replied to the debugger. If the access was a read, instead the response might indicate that the access completed successfully and return unpredictable data.

This issue occurs under this condition: when a system reset is asserted (nRST low) on a specific cycle, while the debugger is completing an access on the system, using the system view. An example would be, when a debugger, like the CCS-IDE memory browser window, is refreshing content using the system view. This is not an issue for a CPU only reset and, this is not an issue during a power-on-reset (nPORRST) either.

Workaround(s): Avoid performing debug reads and writes while the device might be in reset.
MSS#26  \textit{DMA Requests Lost During Suspend Mode}

\begin{tabular}{|l|}
\hline
Revision(s) Affected: & AWR1843 ES1.0 \\
\hline
Description: & While the device is halted in suspend mode by the debugger, the DMA is expected to complete the remaining transfers of a block, if the DEBUG MODE bit field of the GCTRL register is configured to ‘01’. Instead, the DMA does not complete the remaining transfers of a block but, rather stops after two more frames of data are transferred. Subsequent DMA requests from a peripheral to trigger the remaining frames of a block can be lost. \\
 & This issue occurs only in the following conditions: \\
 & • The device is suspended by a debugger \\
 & • A peripheral continues to generate requests while the device is suspended \\
 & • The DMA is setup to continue the current block transfer during suspend mode with the DEBUG MODE bit field of the GCTRL register set to ‘01’ \\
 & • The request transfer type TTYPE bit in the CHCTRL registers is set to frame trigger (‘0’) \\
\hline
Workaround(s): & \textbf{Workaround #1:} \\
& Use TTYPE = block transfer (‘1’), when the DEBUG MODE bit field is ‘01’ (\textit{finish current block transfer}) \\
& \textit{or} \\
& \textbf{Workaround #2:} \\
& Use the DMA DEBUG MODE = ‘00’ (ignore suspend), when using TTYPE = frame transfer (‘0’) to complete the block transfer, even after suspend/halt is asserted. \\
\hline
\end{tabular}
MSS#27  MibSPI in Slave Mode in 3- or 4-Pin Communication Transmits Data Incorrectly for Slow SPICLK Frequencies and for Clock Phase = 1

Revision(s) Affected: AWR1843 ES1.0

Description:
The MibSPI module, when configured in multibuffered slave mode with 3-functional pins (CLK, SIMO, SOMI) or 4-functional pins (CLK, SIMO, SOMI, nENA), could transmit incorrect data when all the following conditions are met:
- MibSPI module is configured in multibuffered mode,
- Module is configured to be a slave in the SPI communication,
- SPI communication is configured to be in 3-pin mode or 4-pin mode with nENA,
- Clock phase for SPICLK is 1, and
- SPICLK frequency is MSS_VCLK frequency / 12 or slower

Workaround(s):
The issue can be avoided by setting the CSHOLD bit in the control field of the TX RAM (Multi-Buffer RAM Transmit Data Register). The nCS is not used as a functional signal in this communication; hence, setting the CSHOLD bit does not cause any other effect on the SPI communication.
MSS#28  

**A Data Length Error is Generated Repeatedly in Slave Mode When IO Loopback is Enabled**

**Revision(s) Affected:** AWR1843 ES1.0

**Description:** When a DLEN error is created in Slave mode of the SPI using nSCS pins in IO Loopback Test mode, the SPI module re-transmits the data with the DLEN error instead of aborting the ongoing transfer and stopping. This is only an issue for an IOLPBK mode Slave in Analog Loopback configuration, when the intentional error generation feature is triggered using CTRLDLENERR (IOLPBKTSTCR.16).

**Workaround(s):** After the DLEN_ERR interrupt is detected in IOLPBK mode, disable the transfers by clearing the SPIEN (bit 24) in the SPIGCR1 register and then, re-enable the transfers by resetting the SPIEN bit.
MSS#29  Spurious RX DMA REQ From a Slave Mode MibSPI

Revision(s) Affected: AWR1843 ES1.0

Description: A spurious DMA request could be generated even when the SPI slave is not transferring data in the following condition sequence:

- The MIBSPI is configured in standard (non-multibuffered) SPI mode, as a slave
- The DMAREQEN bit (SPIINT0.16) is set to enable DMA requests
- The Chip Select (nSCS) pin is in an active state, but no transfers are active
- The SPI is disabled by clearing the SPIEN (SPIGCR1.24) bit from '1' to '0'

The above sequence triggers a false request pulse on the Receive DMA Request as soon as the SPIEN bit is cleared from '1' to '0'.

Workaround(s): Whenever disabling the SPI, by clearing the SPIEN bit (SPIGCR1.24), first clear the DMAREQEN bit (SPIINT0.16) to '0', and then, clear the SPIEN bit.
MSS#30 | **MibSPI RX RAM RXEMPTY bit Does Not Get Cleared After Reading**

**Revision(s) Affected:** AWR1843 ES1.0

**Description:** The RXEMPTY flag may not be auto-cleared after a CPU or DMA read when the following conditions are met:

- The TXFULL flag of the latest buffer that the sequencer read out of transmit RAM for the currently active transfer group is 0,
- A higher-priority transfer group interrupts the current transfer group and the sequencer starts to read the first buffer of the new transfer group from the transmit RAM, and
- Simultaneously, the Host (CPU/DMA) is reading out a receive RAM location that contains valid received data from the previous transfers.

**Workaround(s):** If at all possible, avoid transfer groups interrupting one another.

If dummy buffers are used in lower-priority transfer groups, select the appropriate "BUFMODE" for them (like, SKIP/DISABLED) unless, there is a specific need to use the "SUSPEND" mode.
MSS#31  

**CPU Abort Generated on a Write to Implemented CRC Space After a Write to Unimplemented CRC Space**

**Revision(s) Affected:** AWR1843 ES1.0

**Description:** An abort could be generated on a write to a legal address in the address offset region (0x0000–0x01FF) of the CRC register space when a normal mode write to an unimplemented address region (0x0200–0xFFFF) of the CRC register space is followed by a write to a legal address region (0x0000–0x01FF) of the CRC register space.

**Workaround(s):** None.
MSS#32  **DMMGLBCTRL BUSY Flag Not Set When DMM Starts Receiving A Packet**

**Revision(s) Affected:** AWR1843 ES1.0

**Description:** The BUSY flag in the DMMGLBCTRL register should be set when the DMM starts receiving a packet or has data in its internal buffers. However, the BUSY flag (DMMGLBCTRL.24) may not get set when the DMM starts receiving a packet under the following condition:

- The BUSY bit is set only after the packet has been received, de-serialized, and written to the internal buffers. It stays active while data is still in the DMM internal buffers. If the internal buffers are empty (meaning that no data needs to be written to the destination memory) then, the BUSY bit will be cleared.

**Workaround(s):** Wait for a number of DMMCLK cycles (for example, 95 DMMCLK cycles) beyond the longest reception and deserialization time needed for a given packet size and DMM port configuration, before checking the status of the BUSY flag, and after the DMM ON/OFF bit field (DMMGLBCTRL.[3:0]) has been programmed to OFF.
MSS#33  MiBSPI RAM ECC is Not Read Correctly in DIAG Mode

Revision(s) Affected:  AWR1843 ES1.0

Description:  A Read operation to the ECC address space of the MiBSPI RAM in DIAG mode, does not return the correct ECC value for the first 128 buffers, if the Extended Buffer support is implemented but, the Extended Mode is disabled for the particular MiBSPI instance.

Workaround(s):  None.
MSS#34  

**HS Device Does Not Reboot Successfully on Warm Reset Getting Triggered or With Internal Software Reset**

**Revision(s) Affected:** AWR1843 ES1.0

**Description:** A warm reset triggered by a watchdog expiry (MSS Wdog), a software register write (SOFTSYSRST), or an external warm reset pin does not ensure a successful reboot of the device in a secure device (HS device).

**Workaround(s):**

- A warm reset should not be triggered externally or internally by a watchdog expiry, a software write, or other trigger mechanisms.

To initiate a reset cycle, external circuitry should be used on the sensor design. The external circuitry uses the watchdog, nERROR OUT monitoring, or other kinds of GPIO signaling to trigger a reset using the nRST pin of the device.
EDMA TPTC Generates an Incorrect Address on the Read Interface, Causing one or More Data Integrity Failures, Hangs, or Extra Reads

Revision(s) Affected: AWR1843 ES1.0

Description:
Certain scenarios could lead to an incorrect read, hang, or data integrity issues in the EDMA TPTC block. Table 3 shows the various scenarios and the resulting effects of each scenario.

A scenario happens, if ALL conditions listed for that scenario are satisfied (true); that is, "AND" of all conditions.

A “hang” outcome means that one or more attempts of the hang causing scenarios can progressively lead to not receiving a "transfer completion" indication from the TPTC. The last transfer attempt which does not receive the completion indication can be any transfer – any scenario transfer within this advisory or even outside of this advisory.

Table 3. EDMA TPTC Scenario IDs and Condition Results

<table>
<thead>
<tr>
<th>SCENARIO ID</th>
<th>CONDITIONS</th>
<th>DATA INTEGRITY FAILURES</th>
<th>HANGS</th>
<th>EXTRA READS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>AB-sync; BCNT &gt; 1; ACNT not in [2,4,8,16,32]; (ACNT &lt; 64) OR ((ACNT = 64) AND (SRCBIDX != ACNT)); Source Addressing crossing 4-KB boundary for any of the BCNT number of ACNT transfers. condition [1]</td>
<td>Yes (see Figure 5)</td>
<td>Possible</td>
<td>Possible</td>
</tr>
<tr>
<td>2</td>
<td>AB-sync with BCNT=1 or A-sync; ACNT not in [2,24, 32]; ACNT &lt;= 64; Source Addressing crossing 4-KB boundary for any of the BCNT number of ACNT transfers. condition [1]</td>
<td>No</td>
<td>Possible</td>
<td>Yes</td>
</tr>
<tr>
<td>3</td>
<td>AB-sync with BCNT=1 or A-sync; ACNT in [2,24, 32]; Source Addressing crossing 4-KB boundary for any of the BCNT number of ACNT transfers. condition [1]</td>
<td>No</td>
<td>No</td>
<td>Yes (see Figure 2)</td>
</tr>
<tr>
<td>4</td>
<td>AB-sync; BCNT &gt; 1; ACNT in [2,4,8,16,32]; SRCBIDX = ACNT; ACNT * BCNT &lt;=64; Source Addressing crossing 4-KB boundary for any of the BCNT number of ACNT transfers. condition [1]</td>
<td>No</td>
<td>No</td>
<td>Yes (see Figure 3)</td>
</tr>
<tr>
<td>5</td>
<td>AB-sync; BCNT &gt; 1; ACNT in [2,4,8,16,32]; SRCBIDX = ACNT; ACNT * BCNT &lt;=64; Source Addressing does NOT cross 4-KB boundary for any of the BCNT number of ACNT transfers; that is, NOT of condition [1]; Source Addressing crossing 4-KB boundary for merged source array of size ACNT * BCNT. condition [2]</td>
<td>No</td>
<td>No</td>
<td>Yes (see Figure 4)</td>
</tr>
</tbody>
</table>

Source Addressing crossing 4-KB boundary cross condition [1] in Table 3 is defined as follows:

\[ X(i) = \text{LSB}_{12}\text{bits}(\text{SRC_ADDR} + i \times \text{SRC_BIDX}) + \text{ACNT} > 0x1000 \]

where 0 <= i < BCNT

Source Addressing crossing 4-KB boundary cross condition [2] in Table 3 is defined as follows:

\[ \text{LSB}_{12}\text{bits}(\text{SRC_ADDR}) + (\text{ACNT} \times \text{BCNT}) > 0x1000 \]
For the condition [1] and condition [2] expressions above, note that the SRC_ADDR is candidate source address considering the C-dimension. If CCNT > 1, then, the SRC_ADDR would be every candidate source address for all CCNTs depending on the type of transfer and the SRCCIDX.

For each extra read in SCENARIO ID #3 for which condition [1] is applicable, each index i above that satisfies the condition results in extra read from starting address SRC_ADDR + (i + 1) * SRCBIDX and of length equal to the distance from the start address to the boundary [that is, 0x1000 – X(i)]. Note: length < ACNT.

![Figure 2. Scenario ID #3 – Extra Reads](image-url)
For each extra read in SCENARIO ID #4 for which condition [1] is applicable, there is an extra read from starting address SRC + BCNT * ACNT (=SRCBIDX) of length equal to the distance of the start address to the boundary [that is, length is 0x1000 – X(i)], where i is the index of one and only 4-KB boundary crossing condition. Note: length < ACNT.
For the extra read in SCENARIO ID #5 for which condition [2] is applicable, there is an extra read from starting address SRC + BCNT * ACNT (=SRCBIDX) of length equal to the distance of the start address and is of length ACNT.

---

**Figure 4. Scenario ID #5 – Extra Reads**
Figure 5. Scenario ID #1 – Data Integrity Failure

Workaround(s):

- **Workaround #1 - ALL SCENARIO IDs (see Table 3)**
  Prevent one or more of the conditions necessary for the problematic scenarios to happen.

- **Workaround #2 - SCENARIO ID #3, SCENARIO ID #4, and SCENARIO ID #5 – EXTRA READS**
  For SCENARIO ID #3, SCENARIO ID #4, and SCENARIO ID #5 – EXTRA READS, another workaround besides avoiding the 4-KB boundary cross conditions, is to ensure that buffers involved in this kind of transfer are positioned so that extra reads stay within the physical memory boundaries. If the extra reads go to Reserved space or space blocked by the Memory Protection Unit (MPU), the TPTC generates a bus error interrupt to the processor.
**ANA#08**  
*Doppler Spur Observed for Narrow Chirps Spanning 79.2 GHz*

**Revision(s) Affected:** AWR1843 ES1.0  
**Description:** There is a nonlinearity of the synthesizer when crossing 79.2 GHz due to coupling from its reference to the VCO.  
*Implication:* There is a spur in non-zero Doppler bin if the synthesizer crosses 79.2 GHz during a chirp. The exact Doppler bin depends on the slope of the ramp. This is not observed for wide bandwidth or higher ramp slopes.  
**Workaround(s):** Avoid narrow, slow ramps near 79.2 GHz.

**ANA#09**  
*Synthesizer Frequency Nonlinearity at 76.8 GHz when Synthesizer (Chirp) Frequency Monitor Enabled*

**Revision(s) Affected:** AWR1843 ES1.0  
**Description:** When the synthesizer (chirp) frequency monitor is enabled and the synthesizer chirp reaches 76.8 GHz, the frequency error can be as high as 500 kHz due to coupling between the monitor and the synthesizer.  
*Implication:* Increased nonlinearity in the chirp can lead to up to 20 dB degradation in the noise floor surrounding large objects, including the bumper reflection. This leads to potential loss of dynamic range when large and small objects are present simultaneously.  
**Workaround(s):**  
1. Disable the synthesizer frequency monitor during profiles where the LO crosses 76.8GHz.  
2. Use non-functional chirps to detect nonlinearities in the synthesizer.

**ANA#10**  
*Unreliable Readings from Synthesizer Supply Voltage Monitor*

**Revision(s) Affected:** AWR1843 ES1.0  
**Description:** During monitoring, the thresholds used to determine if the synthesizer supply voltage is within limits are much stricter than necessary for proper circuit operation. This can lead to occasional, erroneous reporting of supply failures even when there is no adverse impact on circuit or system behavior.  
*Implication:* The user cannot rely on supply failure indication from the supply monitors of PM, Clock and LO subsystems. The affected field is STATUS SUPPLY_PMCLKLO in the monitoring report message: AWR_MONITOR_PMCLKLO_INTERNAL_ANALOG_SIGNALS_REPORT_AE_SB.  
**Workaround(s):** Ignore the field STATUS_SUPPLY_PMCLKLO in the monitoring report message: AWR_MONITOR_PMCLKLO_INTERNAL_ANALOG_SIGNALS_REPORT_AE_SB.
ANA#11  TX, RX Gain Calibrations Sensitive to Large External Interference

Revision(s) Affected: AWR1843 ES1.0

Description: External interference present on the RX or TX pins with level >-10dBm can lead to degraded accuracy or errors in the peak detector, TX, and RX gain calibrations. If the interference changes its level while these calibrations are actively running, the calibration algorithm may interpret this as a change in signal power, leading to incorrect convergence. This applies to boot-time PD, TX, and RX calibrations, as well as run-time TX output power calibration.

Workaround(s): The incident power detector in the TX output power detector, along with the absolute level of the PA loopback used during the PA loopback monitors, are insensitive to this, and they can be used to check that the calibrations converged correctly. Calibration can be re-run if large interference was observed.

ANA#12  Second Harmonic (HD2) is Present When Receiver is Tested Standalone Using CW Input

Revision(s) Affected: AWR1843 ES1.0

Description: When the receiver is tested standalone using a CW input, a second harmonic (HD2) can be observed in the final ADC output at a level of –55 dBc.

Workaround(s): No workaround available at this time. However, in many typical radar use-cases the HD2 does not affect the system performance due to two reasons
1. Since the HD2 comes from a coupling to the LO signal, there is an inherent suppression of the HD2 level due to the self-mixing effect (i.e., phase noise and phase spur suppression effect at the mixer).
2. In real-life scenarios there is often a double-bounce effect of the radar signal reflected from the target, which leads to a ghost object at twice the distance of the actual object. This effect is often indistinguishable from the effect of HD2 itself.

ANA#13  TX1 to TX3 Phase Mismatch Variation over Temperature is Double that of TX2/TX1 and TX3/TX2 Combinations

Revision(s) Affected: AWR1843 ES1.0

Description: TX3/TX1 combination exhibits a phase mismatch variation of ±6° from –40°C to 140°C whereas, TX2/TX1 and TX3/TX2 combinations exhibit a lower variation of ±3°C over the same temperature range.

Workaround(s): In applications requiring high phase accuracy across TX channels, a background angle calibration or a 2-point calibration can be used to control phase variation over temperature.
Trademarks

All trademarks are the property of their respective owners.
# Revision History

## Changes from December 28, 2018 to August 31, 2019 (from * Revision (December 2018) to A Revision)  

<table>
<thead>
<tr>
<th></th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Table 1 (Advisory to Silicon Variant / Revision Map): Added MSS#22 through MSS#34 advisories, all silicon revisions</td>
<td>4</td>
</tr>
<tr>
<td>• Table 1: Added ANA#11 and ANA#13 advisories, all silicon revisions</td>
<td>4</td>
</tr>
</tbody>
</table>

## Changes from August 31, 2019 to September 30, 2019 (from A Revision (August 2019) to B Revision)  

<table>
<thead>
<tr>
<th></th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Table 1 (Advisory to Silicon Variant / Revision Map): Added MSS#03, MSS#13, MSS#21, and MSS#35 advisories, all silicon revisions</td>
<td>4</td>
</tr>
</tbody>
</table>
IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale (www.ti.com/legal/termsofsale.html) or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2019, Texas Instruments Incorporated