TDA2x SoC for Advanced Driver Assistance Systems (ADAS) Silicon Revision 2.0, 1.1

Errata

Literature Number: SPRZ397J
NOVEMBER 2012 – REVISED FEBRUARY 2021

Table of Contents

This page intentionally left blank.

Chapter 1

Introduction

This document describes the known exceptions to the functional specifications for the device.

Related Documentation

ADAS Applications Processor TDA2x System-on-Chip Technical Brief (SPRT680)

TDA2x SoC for Advanced Driver Assistance Systems (ADAS) Technical Reference Manual (SPRUI29)

TDA2x ADAS Applications Processor 23mm Package (ABC Package) Silicon Revision 1.1 Data Manual (*SPRS859*)

TDA2x ADAS Applications Processor 23mm Package (ABC Package) Silicon Revision 2.0 Data Manual (SPRS951)

TDA2x ADAS Applications Processor 17mm Package (AAS Package) Silicon Revision 1.1 Data Manual (*SPRS884*)

TDA2x ADAS Applications Processor 17mm Package (AAS Package) Silicon Revision 2.0 Data Manual (SPRS952)

TDA2x Pad Configuration Tool (Contact your TI representative.)

TDA2x Clock Tree Tool (CLOCKTREETOOL-AUTOMOTIVE)

TDA2x Register Descriptor Tool

TDA2x Code Composer Chip Support Packages (Automotive)

TDA2x Lauterbach Package

Trademarks

OMAP and SmartReflex are trademarks of Texas Instruments.

Cortex is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere.

SD is a registered trademark of Toshiba Corporation.

MMC and eMMC are trademarks of MultiMediaCard Association.

JTAG is a registered trademark of JTAG Technologies, Inc.

Linux is a registered trademark of Linus Torvalds.

All other trademarks are the property of their respective owners.

Modules Impacted

Table 1-1. Silicon Advisories, Limitations, and Cautions by Module

MODULE	DESCRIPTION	SILICON REVISIONS AFFECTED		
		TDA2x		
		1.0	1.1	2.0
NA	i781: Power Delivery Network Verification	Yes	Yes	Yes
	i842: Multiple Resets Required Before Chip Is Functional	Yes		
	i862: Reset Should Use PORz		Yes	Yes
	i864: VDDS18V to VDDSHVn Current Path	Yes	Yes	Yes
	i931: VDD to VDDA_"PHY" Current Path	Yes	Yes	Yes
ATL	i933: Access to IODELAY at Same Time as Other Peripheral on L4_PER2 Can Hang	Yes	Yes	Yes
BOOT	i875: Power-on-Reset (PORz) Warm Boot Hang	Yes	Yes	
	i927: SoC Doesn't Read Redundant ONFI Parameter Pages in NAND Boot Mode	Yes	Yes	Yes
Control Module	i813: Spurious Thermal Alert Generation When Temperature Remains in Expected Range	Yes	Yes	Yes
	i814: Bandgap Temperature Read Dtemp Can Be Corrupted	Yes	Yes	Yes
	i827: Thermal Alert Will Not Be Generated When Bandgap Is Configured in "Smart Idle" Mode	Yes	Yes	Yes
	i857: Optional VOUT3 Clock Muxing Not Meeting IO Timing	Yes		
	i858: DELAYMODE Mechanism Not Selecting Proper Delay for Some IP Modes	Yes		
	i863: MMC2 Has PU/PD Contention Immediately after Release from Reset	Yes	Yes	Yes
	i869: IO Glitches Can Occur When Changing IO Settings	Yes	Yes	Yes
	i870: PCIe Unaligned Read Access Issue	Yes	Yes	Yes
	i885: Software Requirements for Data Manual IO Timing	Yes	Yes	Yes
	i900: SoC Will Hang If Region 5 Accessed While CTRL_CORE_MMR_LOCK_5 Is Locked	Yes	Yes	Yes
DCAN	i841: DCAN Ram Initialization Issue	Yes		
	i893: DCAN Initialization Sequence	Yes	Yes	Yes
	i933: Access to IODELAY at Same Time as Other Peripheral on L4_PER2 Can Hang	Yes	Yes	Yes
DEBUG	i840: DSP Trace Data Corruption	Yes		
	i879: DSP MStandby Requires CD_EMU in SW_WKUP	Yes	Yes	Yes
DMA	i378: sDMA Channel Is Not Disabled after a Transaction Error	Yes	Yes	Yes
	i698: DMA4 Generates Unexpected Transaction on WR Port	Yes	Yes	Yes
	i699: DMA4 Channel Fails to Continue With Descriptor Load When Pause Bit Is Cleared	Yes	Yes	Yes
	i868: McASP to EDMA Synchronization Level Event Can Be Lost	Yes	Yes	
DSP	i872: DSP MFlag Output Not Initialized	Yes	Yes	Yes
	i879: DSP MStandby Requires CD_EMU in SW_WKUP	Yes	Yes	Yes
	i883: DSP Doesn't Wake from Subsystem Internal Interrupts	Yes	Yes	Yes
	i898: DSP Pre-fetch Should Be Disabled before Entering Power Down Mode	Yes	Yes	Yes
DSS	i596: BITMAP1-2-4 Formats Not Supported by the Graphics Pipeline	Yes	Yes	Yes
	i631: Wrong Access in 1D Burst for YUV4:2:0-NV12 Format	Yes	Yes	Yes
	i641: Overlay Optimization Limitations	Yes	Yes	Yes
	i734: LCD1 Gamma Correction Is Not Working When GFX Pipe Is Disabled	Yes	Yes	Yes
	i815: Power Management Enhancement Implemented Inside DSS Leads to DSS Underflows	Yes	Yes	Yes
	i829: Reusing Pipe Connected to Writeback Pipeline On-the-Fly to an Active Panel	Yes	Yes	Yes
	i838: DSS BT.656/BT.1120 Max Horizontal Blanking Is Non Compliant	Yes	Yes	Yes

Table 1-1. Silicon Advisories, Limitations, and Cautions by Module (continued)

MODULE	DESCRIPTION	SILICON REVISIONS AFFECTED		
		TDA2x		
		1.0	1.1	2.0
	i839: Some RGB and YUV Formats Have Non-Standard Ordering	Yes	Yes	Yes
	i901: DSS VOUT3 on VDDSHV6 Domain (vin1a Pins) Should Not Be Used in 3.3V Mode	Yes	Yes	Yes
	i932: DPLL_VIDEOn May Require Multiple Lock Attempts	Yes	Yes	Yes
	i936: DSS LCD/DPI Out Field Reversal in Interlaced RGB Mode	Yes	Yes	Yes
EDMA	i837: MMU1 Not Functional With EDMA	Yes		
	i844: EDMA to VCP Stream Burst Is Not Functional	Yes	Yes	Yes
	i868: McASP to EDMA Synchronization Level Event Can Be Lost	Yes	Yes	
EMIF	i727: Refresh Rate Issue after Warm Reset	Yes	Yes	Yes
	i729: DDR Access Hang after Warm Reset	Yes	Yes	Yes
	i854: EMIF CC 2b Error Can Cause Corrupt Internal Bus Read Response	Yes		
	i878: MPU Lockup With Concurrent DMM and EMIF Accesses	Yes	Yes	Yes
	i882: EMIF: DDR ECC Corrupted Read/Write Status Response	Yes	Yes	
	i895: EMIF_FW: System Hang When EMIF Firewall Is Reconfigured While There Is Activity on EMIF Interface	Yes	Yes	Yes
eMMC/SD/ SDIO	i802: MMCHS DCRC Errors During Tuning Procedure	Yes	Yes	Yes
	i803: MMCHS Read Transfer With CMD23 Never Complete When BCE=0 and ADMA Used	Yes	Yes	Yes
	i832: DLL SW Reset Bit Does Not Reset to 0 after Execution	Yes	Yes	Yes
	i834: MMCHS HS200 and SDR104 Command Timeout Window Too Small	Yes	Yes	Yes
	i836: Bus Testing Commands CMD19 Incorrectly Waits for CRC Status Return	Yes	Yes	Yes
	i843: MMC1/2/3 Speed Issues	Yes	Yes	
	i853: MMC2,3,4 Do Not Support 3.3V/1.8V Dynamic Switch	Yes		
	i856: 32k Oscillator Fails to Start-Up at POR	Yes	Yes	Yes
	i863: MMC2 Has PU/PD Contention Immediately after Release from Reset	Yes	Yes	Yes
	i884: MMC4 Speed Limited to 38.5 MHz	Yes	Yes	
	i887: MMC3 Speed Limited to 64 MHz	Yes	Yes	Yes
	i890: MMC1 IOs and PBIAS Must Be Powered-Up before Isolation	Yes	Yes	Yes
	i929: MMC1/2 SDR104/HS200 Mode DLL Delay Value May Result In Unexpected Tuning Pattern Errors	Yes	Yes	Yes
GMAC_SW	i877: RGMII Clocks Should Be Enabled at Boot Time	Yes	Yes	Yes
	i880: Ethernet RGMII2 Limited to 10/100 Mbps	Yes	Yes	
	i899: Ethernet DLR Is Not Supported	Yes	Yes	Yes
	i903: Ethernet RMII Interface RMII_MHZ_50_CLK Not Supported as Output Reference Clock	Yes	Yes	Yes
GPIO	i856: 32k Oscillator Fails to Start-Up at POR	Yes	Yes	Yes
GPMC	i858: DELAYMODE Mechanism Not Selecting Proper Delay for Some IP Modes	Yes		
	i927: SoC Doesn’t Read Redundant ONFI Parameter Pages in NAND Boot Mode	Yes	Yes	Yes
I2C	i694: System I2C Hang Due to Miss of Bus Clear Support	Yes	Yes	Yes
	i833: I2C Module in Multislave Mode Potentially Acknowledges Wrong Address	Yes	Yes	Yes
	i930: I2C1 and I2C2 May Drive Low During Reset	Yes	Yes	Yes
INTC	i883: DSP Doesn't Wake from Subsystem Internal Interrupts	Yes	Yes	Yes
Interconnect	i871: L4_PER3 Firewall Initiator ConnID Value Left-Shift 1-Bit	Yes	Yes	Yes
McASP	i848: McASP IO Pad Loopback Not Functional	Yes	Yes	Yes

Table 1-1. Silicon Advisories, Limitations, and Cautions by Module (continued)

MODULE	DESCRIPTION	SILICON REVISIONS AFFECTED		
		TDA2x		
		1.0	1.1	2.0
	i858: DELAYMODE Mechanism Not Selecting Proper Delay for Some IP Modes	Yes		
	i868: McASP to EDMA Synchronization Level Event Can Be Lost	Yes	Yes	
	i933: Access to IODELAY at Same Time as Other Peripheral on L4_PER2 Can Hang	Yes	Yes	Yes
MLB	i933: Access to IODELAY at Same Time as Other Peripheral on L4_PER2 Can Hang	Yes	Yes	Yes
MMU	i837: MMU1 Not Functional With EDMA	Yes		
MPU	i878: MPU Lockup With Concurrent DMM and EMIF Accesses	Yes	Yes	Yes
	i940: MPU COUNTER_REALTIME saturates after several hundred days	Yes	Yes	Yes
PCIe	i847: PCIe TXP Output Drives 1/2 Amplitude for Some Data Bits	Yes		
	i870: PCIe Unaligned Read Access Issue	Yes	Yes	Yes
	i909: PCIe Unintentional Translation of Outbound Message TLPs	Yes	Yes	Yes
	i926: PCIe Preferred PCIe_PHY_RX SCP Register Settings Updated	Yes	Yes	Yes
	i935: MSI Bit in PCIECTRL_TI_CONF_IRQSTATUS_MSI Register Does Not Clear Automatically	Yes	Yes	Yes
PRCM	i810: DPLL Controller Can Get Stuck While Transitioning to a Power Saving State	Yes	Yes	Yes
	i826: HSDIVIDER1 CLKOUT4 Could Glitch During On-the-Fly Divider Change to/from Divide-by-2.5	Yes	Yes	Yes
	i850: ESD Fail on MPU PLL Power	Yes		
	i852: IODelay Recalibration Issue	Yes		
	i876: DVFS Only Supported on MPU	Yes	Yes	Yes
	i886: FPDLink PLL Unlocks With Certain SoC PLL M/N Values	Yes	Yes	Yes
	i892: L3 Clocks Should Be Enabled at All Times	Yes	Yes	Yes
PWMSS	i933: Access to IODELAY at Same Time as Other Peripheral on L4_PER2 Can Hang	Yes	Yes	Yes
QSPI	i851: QSPI Redundant SBL Feature Sector Size Mismatch With Flash	Yes		
	i855: QSPI Mode0/1/2 Not Functional and Mode3 Limited to 48MHz	Yes	Yes	Yes
	i858: DELAYMODE Mechanism Not Selecting Proper Delay for Some IP Modes	Yes		
	i861: QSPI-4 Boot Mode Is Not Functional	Yes		
	i912: QSPI_SPI_CMD_REG [25:24] Masked from Read in RTL	Yes	Yes	Yes
	i916: QSPI Reads Can Fail For Flash Devices with HOLD Function	Yes	Yes	Yes
SATA	i782: SATA AHCI Command Issue Order	Yes	Yes	Yes
	i783: SATA Lockup after SATA DPLL Unlock/Relock	Yes	Yes	Yes
	i807: SATA Host Controller Locks Up if PIO Setup FIS Is Received and Bus Busy and Data Request Bits Are Cleared	Yes	Yes	Yes
	i808: SATA Link Locks Up Under Certain Conditions	Yes	Yes	Yes
	i809: SATA Command Does Not Complete and Software Must Issue a Port Reset Under Certain Conditions	Yes	Yes	Yes
	i818: SATA PHY Reset Required Following SATA PLL Unlock	Yes	Yes	Yes
	i859: SATA 6-Gbps to 3-Gbps Negotiation Can Fail	Yes		
TIMERS	i767: Delay Needed to Read Some Timer Registers after Wakeup	Yes	Yes	Yes
	i856: 32k Oscillator Fails to Start-Up at POR	Yes	Yes	Yes
	i874: TIMER5/6/7/8 Interrupts Not Propagated	Yes	Yes	Yes
UART/IrDA/CIR	i202: MDR1 Access Can Freeze UART Module	Yes	Yes	Yes
	i849: UART2_RXD Is Not Working for MUXMODE=0	Yes	Yes	Yes
	i889: UART Does Not Acknowledge Idle Request after DMA Has Been Enabled	Yes	Yes	Yes
	i933: Access to IODELAY at Same Time as Other Peripheral on L4_PER2 Can Hang	Yes	Yes	Yes

Table 1-1. Silicon Advisories, Limitations, and Cautions by Module (continued)

MODULE	DESCRIPTION	SILICON REVISIONS AFFECTED		
		TDA2x		
		1.0	1.1	2.0
USB	i819: A Device Control Bit Meta-Stability for USB3.0 Controller in USB2.0 Mode	Yes	Yes	Yes
	i820: Unexpected USB Link State Value upon U3 Exit by USB3.0 Link	Yes	Yes	Yes
	i824: USB3.0 Link Cannot Be Established When Suspend Mode Is Enabled	Yes	Yes	Yes
	i845: USB2.0 False Detection of Disconnect Condition	Yes	Yes	Yes
	i896: USB xHCI Port Disable Feature Does Not Work	Yes	Yes	Yes
	i897: USB xHCI Stop Endpoint Command Does Not Work in Certain Circumstances	Yes	Yes	Yes
VCP	i933: Access to IODELAY at Same Time as Other Peripheral on L4_PER2 Can Hang	Yes	Yes	Yes
VIP	i839: Some RGB and YUV Formats Have Non-Standard Ordering	Yes	Yes	Yes
VPE	i839: Some RGB and YUV Formats Have Non-Standard Ordering	Yes	Yes	Yes

Chapter 2

Silicon Advisories

This Chapter describes advisories for the given architecture and provides information for working with those issues.

Revisions SR 2.0, 1.1, 1.0 - Advisories List

i202	***MDR1 Access Can Freeze UART Module***
CRITICALITY	Medium
DESCRIPTION	Because of a glitchy structure inside the UART module, accessing the MDR1 register may create a dummy underrun condition and freeze the UART in IrDa transmission. In UART mode, this may corrupt the transferred data(received or transmitted).
WORKAROUND	To ensure this problem does not occur, the following software initialization sequence must be used each time MDR1 must be changed: 1. If needed, setup the UART by writing the required registers, except MDR1 2. Set appropriately the MDR1.MODE_SELECT bit field 3. Wait for 5 L4 clock cycles + 5 UART functional clock cycles 4. Clear TX and RX FIFO in FCR register to reset its counter logic 5. Read RESUME register to resume the halted operation Step 5 is for IrDA mode only and can be omitted in UART mode.
REVISIONS IMPACTED	SR 2.0, 1.1, 1.0 TDA2x: 2.0, 1.1, 1.0 DRA75x, DRA74x: 2.0, 1.1, 1.0 AM572x: 2.0, 1.1

i378	***sDMA Channel Is Not Disabled after a Transaction Error***
CRITICALITY	Medium
DESCRIPTION	In case of destination synchronized transfer on the write port (or source sync with SDMA.DMA4_CCRi[25] BUFFERING_DISABLE = 1), if a transaction error is reported at the last element of the transaction, the channel is not automatically disabled by DMA.
WORKAROUND	Whenever a transaction error is detected on a transaction on the write side of the channel i, software must disable the channel(i) by setting the DMA4_CCRi[7] ENABLE bit to 0.
REVISIONS IMPACTED	SR 2.0, 1.1, 1.0 TDA2x: 2.0, 1.1, 1.0 DRA75x, DRA74x: 2.0, 1.1, 1.0 AM572x: 2.0, 1.1

i631	*Wrong Access in 1D Burst for YUV4:2:0-NV12 Format*
CRITICALITY	Low
DESCRIPTION	When in YUV4:2:0 format in 1D burst, the DISPC DMA skips lines when fetching Chroma sampling.
WORKAROUND	If YUV4:2:0-1D burst is required: • Set DISPC_VIDp_ATTRIBUTES[22] DOUBLESTRIDE to 0x0 and DISPC_VIDp_ATTRIBUTES[13:12] ROTATION to 0x1 or 0x3
REVISIONS IMPACTED	SR 2.0, 1.1, 1.0 TDA2x: 2.0, 1.1, 1.0 DRA75x, DRA74x: 2.0, 1.1, 1.0 AM572x: 2.0, 1.1

i694	***System I2C Hang Due to Miss of Bus Clear Support***
CRITICALITY	Low
DESCRIPTION	There is no H/W mechanism preventing violating below I2C Bus clear standard requirement. If the data line (SDA) is stuck LOW, the master should send 9 clock pulses. The device that held the bus LOW should release it sometime within those 9 clocks. If not, then use the HW reset or cycle power to clear the bus. Sys_Warmreset doesn't reset the I2C IP it does at IC level. So, once the situation is reached, IC is seeing bus busy status bit.
WORKAROUND	I2C SW handler could be programmed to detect such a locked situation. In this case, it will check the Bus Busy bit and issue the needed clock pulses.
REVISIONS IMPACTED	SR 2.0, 1.1, 1.0 TDA2x: 2.0, 1.1, 1.0 DRA75x, DRA74x: 2.0, 1.1, 1.0 AM572x: 2.0, 1.1

i698 ***DMA4 Generates Unexpected Transaction on WR Port***

CRITICALITY Medium

DESCRIPTION The DMA4 channel generates an unexpected transaction on WR port under the following 2 scenarios:

- Scenario 1
  1. Software synchronization: Bit fields SYNCHRO_CONTROL and SYNCHRO_CONTROL_UPPER are set to 0 in register DMA4_CCRi
     - Channel element number: Bit field CHANNEL_ELMNT_NBR is set to 0x9 in register DMA4_CENi
     - Channel frame number: Bit field CHANNEL_FRAME_NBR is set to 0x1 in register DMA4_CFNi
     - Element size: Bit field DATA_TYPE is set to 0x2 in register DMA4_CSDPi
     - Destination addressing mode: Bit field DST_AMODE is set to 0x1 in register DMA4_CCRi
     - Destination is packed: Bit field DST_PACKED is set to 0x1 in register DMA4_CSDPi
     - Destination endianism: Bit field DST_ENDIAN is set to 0x0 in register DMA4_CSDPi
     - Destination burst enable: Bit field DST_BURST_EN is set to 0x1 in register DMA4_CSDPi
     - Destination start address: Register DMA4_CDSAi is set to 0xabcd0000
     - Disable graphics operation: Bit fields CONST_FILL_ENABLE and TRANSPARENT_COPY_ENABLE are set to 0x0 in register DMA4_CCRi

     The channel has got an ERR response on the WR port before the end of block transfer. The channel has gone for clean abort and got disabled. The same channel has been configured with soft-sync and included in the channel chaining (This channel is not the head of the chain). When this channel gets enabled through the link, the channel is writing the data out as soon as it fetches the data from Read side. It is expected that the channel should go with burst transfer, but it is going for single transfers.
     This results in a performance issue as DMA is executing single transfers instead of burst transfers. This performance issue is also observed while using the channel with destination synchronization and prefetch enabled.
  2. Destination sync with Prefetch enabled: Bit field SEL_SRC_DST_SYNC is set to 0x0; Bit fields SYNCHRO_CONTROL_UPPER and SYNCHRO_CONTROL should not be set to 0x0; Bit field PREFETCH is set to 0x1 in register DMA4_CCRi. The other settings remain same as in use case #1 described above
- Scenario 2
  - The channel has got an ERR response on the WR port before the end of block transfer. The channel has gone for clean abort and got disabled. The same channel has been configured with destination-sync with prefetch enabled and included in the channel chaining (This channel is not the head of the chain). When this channel gets enabled through the link, the read port will start its transaction. If the HWR request to this channel comes before the channel gets its first response, the channel will start a WR transaction with byte enable 0. Also, the internal data counters get updated and the corresponding data will never come out of DMA4. The Data FIFO locations are also not recovered.
    This results in a Data Integrity issue.

WORKAROUND There is a software workaround to solve this issue

1. Work around to resolve both Data Integrity and Performance issue:
   - Dummy enable-disable for an aborted Channel. i.e. on abort, configure the channel as soft sync with No of frames = 0 and enable the channel by writing 0x1 into the ENABLE bitfield of register DMA4_CCRi. Wait for the Address Misaligment Interrupt. The channel is now ready for reuse.
   - Ensure that clean drain happens for a channel that is or is to be used as part of a channel chain. i.e. ensure that the abort conditions never occur for this channel
   - If a channel gets aborted, do not reuse the channel in a chain
   - Don't use channel chaining
2. Work around to resolve the data integrity only.
   Disable prefetch in all channels that are part of a channel chain

REVISIONS IMPACTED

SR 2.0, 1.1, 1.0

TDA2x: 2.0, 1.1, 1.0

DRA75x, DRA74x: 2.0, 1.1, 1.0

AM572x: 2.0, 1.1

i699 ***DMA4 Channel Fails to Continue With Descriptor Load When Pause Bit Is Cleared***

CRITICALITY Low

DESCRIPTION This Bug can occur only in a channel that is part of a channel chain. If channel chaining is not used, this bug is never seen.

An exact corner case sequence of events must occur. The sequence is:

- The channel is enabled and then aborted*.
- This same channel is now configured as part of a channel chain (it should not be the head of the channel chain).
- The channel is configured as "software synchronized" or "hardware synchronized at destination with prefetch enabled"
- The channel gets enabled through linking.

* Following is the subset of abort conditions for this scenario:

- The channel is disabled in the middle of transaction and channnel is not a drain candidate.
- The channel gets a transaction error on write port but not at the end-of-block transaction.
- The channel gets a read transaction error and is not a drain candidate.

WORKAROUND The software workaround is to configure DMA4 to be in no-standby or force-standby mode before clearing the PAUSE bit. The DMA4 can be reverted back to smart-standby mode after a certain period (after detecting DMA4_CSRi[15:15] of corresponding channel to be 0 or ensuring DMA4_CSRi[7:7] bit of corresponding channel to be 0. This ensures descriptor load completion or channel termination.

REVISIONS IMPACTED SR 2.0, 1.1, 1.0

TDA2x: 2.0, 1.1, 1.0

DRA75x, DRA74x: 2.0, 1.1, 1.0

AM572x: 2.0, 1.1

i727	***Refresh Rate Issue after Warm Reset***
CRITICALITY	Medium
DESCRIPTION	The refresh rate is programmed in the EMIF_SDRAM_REFRESH_CONTROL[15:0] REFRESH_RATE parameter and is calculated based off of the frequency of the DDR clock during normal operation. When a warm reset is applied to the system, the DDR clock source is set to PLL bypass frequency which is much lower than the functional frequency of operation. Due to this frequency change, upon warm reset de-assertion the refresh rate will be too low until the DDR PLL is set to the functional frequency. This could result in unexpected behavior on the memory side.
WORKAROUND	There are 2 possible work-arounds: 1. Use workaround as outlined in Errata i862 to convert warm reset to PORz. Warm reset will function the same as cold reset with this approach. 2. Use external circuitry to apply reset on DDR RESET# pin when warm reset is asserted. DDR contents will be erased upon warm reset with this approach.
REVISIONS IMPACTED	SR 2.0, 1.1, 1.0 TDA2x: 2.0, 1.1, 1.0 DRA75x, DRA74x: 2.0, 1.1, 1.0 AM572x: 2.0, 1.1

i729 ***DDR Access Hang after Warm Reset***

CRITICALITY Medium

DESCRIPTION When warm reset is asserted, EMIF will preserve the contents of the DDR by entering self-refresh. During warm reset the DDR clock source is set to a slower PLL bypass than during normal operation. This causes the following JEDEC spec violations and could result in a DDR access hang after warm reset:

- DDR clock frequency to the DDR memory is lower than the JEDEC min. clock frequency specified as tCK(avg) min parameter in the JEDEC JESD79-3F DDR3 standard.
- Upon warm reset de-assertion, DDR is taken out of self-refresh and DDR clock frequency is changed from PLL bypass to normal operating frequency. This violates the JEDEC JESD79-3F DDR3 standard that requires input clock to be stable during normal operation.

WORKAROUND There are 2 possible work-arounds:

1. Use workaround as outlined in Errata i862 to convert warm reset to PORz. Warm reset will function the same as cold reset with this approach.
2. Use external circuitry to apply reset on DDR RESET# pin when warm reset is asserted. DDR contents will be erased upon warm reset with this approach.

REVISIONS IMPACTED SR 2.0, 1.1, 1.0

TDA2x: 2.0, 1.1, 1.0

DRA75x, DRA74x: 2.0, 1.1, 1.0

AM572x: 2.0, 1.1

i734 ***LCD1 Gamma Correction Is Not Working When GFX Pipe Is Disabled***

CRITICALITY High

DESCRIPTION LCD1 output supports gamma correction. The color look-up table (CLUT) is shared between the BITMAP to RGB conversion module on GFX pipeline and Gamma correction on the LCD1 output. LUT table can be loaded by SW through DISPC slave port (interconnect) or by DISPC master port using the DISPC DMA.

However, LCD1 gamma correction LUT loading is not working properly and require to enable GFX pipeline for LUT loading. Depending on the load mode (DISPC_CONFIG1[2:1] LOADMODE) used, GFX pipeline can then be disabled after 1st frame.

WORKAROUND

Table 2-1. Workaround/Load mode settings

Load Mode (DISPC_CONFIG1[2:1] LOADMODE)	GFX Enable Condition	Workaround
0x0 (load LUT and data every frame)	Always Enabled	WA1

WA1

To use gamma correction on LCD1 output, software must:

1. Enable the GFX pipeline by setting DISPC_GFX_ATTRIBUTES[0] ENABLE to 0x1.
2. Set the GFX base address (DISPC_GFX_BA_i[31:0] BA) to an accessible frame buffer.
3. Set the GFX window to minimum size by configuring the DISPC_GFX_SIZE[27:16] SIZEY and DISPC_GFX_SIZE[10:0] SIZEX bits.
4. If the GFX pipeline is not to be displayed, set GFX LYR to bottom LYR in LCD1 overlay by setting appropriate DISPC_GFX_ATTRIBUTES[27:26] ZORDER bit field and make GFX fully transparent by setting the global alpha of the GFX to 0x00 in the DISPC_GLOBAL_ALPHA[7:0] GFXGLOBALALPHA bit field.

REVISIONS IMPACTED

SR 2.0, 1.1, 1.0

TDA2x: 2.0, 1.1, 1.0

DRA75x, DRA74x: 2.0, 1.1, 1.0

AM572x: 2.0, 1.1

i767 *Delay Needed to Read Some Timer Registers after Wakeup*

CRITICALITY

Medium

DESCRIPTION

For GP timers:

If a General Purpose Timer (GPTimer) is in posted mode (TSICR[2] POSTED=1), due to internal resynchronizations, values read in TCRR, TCAR1 and TCAR2 registers right after the timer interface clock (L4) goes from stopped to active may not return the expected values. The most common event leading to this situation occurs upon wake up from idle.

GPTimer non-posted synchronization mode is not impacted by this limitation.

For watchdog timers:

Due to internal resynchronizations, values read in Watchdog timers WCRR registers right after the timer interface clock (L4) goes from stopped to active may not return the expected values. The most common event leading to this situation occurs upon wake up from idle. All watchdog timers support only POSTED internal synchronization mode. There is no capability to change the internal synchronization scheme to NON-POSTED by software.

WORKAROUND

For GP timers:

For reliable counter read upon wakeup from IDLE state, software need to issue a non posted read to get accurate value.

To get this non posted read, TSICR[2] POSTED needs to be set at '0' and TSICR[3] READ_MODE needs to be set at '1'.

Note: For GP Timers 1/2/10 the TSICR[3] READ_MODE is a write only bit and reads to this register always return 0.

For watchdog timers:

Software has to wait at least (2 timer interface clock cycles + 1 timer functional clock cycle) after L4 clock wakeup before reading WCRR register of the Watchdog timers.

REVISIONS IMPACTED

SR 2.0, 1.1, 1.0

TDA2x: 2.0, 1.1, 1.0

DRA75x, DRA74x: 2.0, 1.1, 1.0

AM572x: 2.0, 1.1

i782	*SATA AHCI Command Issue Order*
CRITICALITY	Low
DESCRIPTION	Advanced host controller Interface (AHCI) specification defines up to 32 command slots per port. AHCI defines the highest priority slot as (pSlotLoc+1) mod MaxSlots, where pSlotLoc is the last slot from which a command was issued and MaxSlots is the number of slots per port (32 in this case). The device implementation treats command slot 0 as the highest-priority slot and slot 31 as the lowest-priority slot. Example: Assuming that the last command was issued from slot 17 (and SATA_PxCI[31:0] CI contained 0 at an earlier time), when SATA_PxCI[31:0] CI is set to 0xFFFF FFFF, the command issue order should be 18, 19, 20 ... 31, 0, 1, 2 ... 17. Instead, we observe 0, 1, 2, 3 ... 31. This is a bug in the SATA controller.
WORKAROUND	No workaround is available. Implementation does not conform to AHCI specifications, but it does not affect a major functionality of SATA.
REVISIONS IMPACTED	SR 2.0, 1.1, 1.0 TDA2x: 2.0, 1.1, 1.0 DRA75x, DRA74x: 2.0, 1.1, 1.0 AM572x: 2.0, 1.1

i783 ***SATA Lockup after SATA DPLL Unlock/Relock***

CRITICALITY

Low

DESCRIPTION

Consider the following scenario:

1. Initialize SATA.
2. Enable TX/RX PHYs, start controller DMA engine, spin up the device (SATA_PxCMD[1] SUD = 0x1).
3. Enable aggressive transitions to partial or slumber: SATA_PxCMD[26] ALPE = 0x1 and SATA_PxCMD[27] = 0x0/0x1
4. Perform DMA/PIO transfers.
5. Wait until all commands are finished. Interface (only physical lines) should go to low power mode.
6. Check that transition to partial is complete.
7. Stop all DMA machines, set SATA_PxCMD[1] SUD bit to 0, power down the PHYs.
8. Unlock SATA DPLL (DPLLCTRL_SATA.PLL_GO[0] PLL_GO = 0x0)
9. Relock SATA DPLL (DPLLCTRL_SATA.PLL_GO[0] PLL_GO = 0x1), go out to low power mode.
10. Go to Step 2.

After the first loop, when re-executing Step 2 and spinning up the device, communication is blocked between the host and the device, and the SATA is locked up.

A simpler scenario can be used to reproduce the issue. In this case, no SATA commands are issued by the host.

1. Initialize the SATA.
2. Enable PHYs, start RX DMA engine, initiate staggered spin-up, and start TX DMA engine.
3. Read SATA status register SATA_PxTFD.
4. Stop all DMA engine, set SATA_PxCMD[1] SUD bit to 0, power down the PHYs.
5. Unlock and relock SATA DPLL (DPLLCTRL_SATA.PLL_GO[0] PLL_GO = 0x0 then DPLLCTRL_SATA.PLL_GO[0] PLL_GO = 0x1).
6. Go to Step 2.

These issues are usually encountered immediately after the first loop, although this is not always the case.

WORKAROUND

To prevent the SATA Lockup the SATA DPLL Unlock sequence must be performed as follows:

1. Unlock SATA DPLL (DPLLCTRL_SATA.PLL_GO[0] PLL_GO = 0x0)
2. Toggle SATA_PLL_SOFT_RESET bit of CTRL_CORE_SMA_SW_0 register from 0->1
3. Toggle SATA_PLL_SOFT_RESET bit of CTRL_CORE_SMA_SW_0 register from 1->0

REVISIONS IMPACTED

SR 2.0, 1.1, 1.0

TDA2x: 2.0, 1.1, 1.0

DRA75x, DRA74x: 2.0, 1.1, 1.0

AM572x: 2.0, 1.1

i802 ***MMCHS DCRC Errors During Tuning Procedure***

CRITICALITY Low

DESCRIPTION In UHS=I mode, the SD bus operates in high clock frequency mode and the data windows from card on CMD and DAT lines get smaller. The position of the data windows varies depending on the card and the host system. To adjust the sampling clock when SDR104/ HS200 operation mode is used the MMC/SDIO host controller supports a tuning procedure. This tuning circuit is a dedicated DLL which delays the clock signal used, for data sampling.

DCRC error interrupts (MMCHS_STAT[21] DCRC=0x1) can occur during the tuning procedure.

As explained in SD Host Controller Spec version 3.00 Feb 18, 2010, the controller is supposed to inhibit all interrupts except BRR (block read ready) during the tuning procedure (ET=1).

Some DCRC interrupts occur from time to time during tuning upon CMD19 (send tuning block).

This DCRC interrupt, occurs when the last tuning block fails (the last ratio tested).

The root cause is that the delay from CRC check until the interrupt is asserted is bigger than the delay until assertion of the tuning end flag (which masks the interrupts); therefor, when the interrupt bit toggles, the tuning has already ended.

WORKAROUND After the DCRC interrupt occurs during the tuning procedure, software should clear the interrupt before the next command is sent: (MMCHS_STAT[21] DCRC=0x1).

Another workaround is to disable DCDR interrupt during the tuning procedure:

MMCHS_IE [21] DCRC_ENABLE is set to 0x0 (masked).

REVISIONS IMPACTED SR 2.0, 1.1, 1.0

TDA2x: 2.0, 1.1, 1.0

DRA75x, DRA74x: 2.0, 1.1, 1.0

AM572x: 2.0, 1.1

i803 ***MMCHS Read Transfer With CMD23 Never Complete When BCE=0 and ADMA Used***

CRITICALITY Low

DESCRIPTION A data time-out (DTO) error interrupt (MMCHS_STAT[20] DTO=0x1) occurs at the end of reading transfer with CMD23. In this case, transfer completed (TC) interrupt (MMCHS_STAT[1] TC = 0x1) never occurs.

This bug appears only with following restrictive conditions:

- Use CMD23 (command used to specify number of reading and writing block)
- Block count enable BCE (MMCHS_CMD[1] = 0x0
- ADMA (integrated controller DMA system engine) used
- Reading transfer (writing transfer is not affected)
- Only finite transfer affected. CMD12 (stop infinite transfer command) command are not affected.
- Issue happens with both large block count (>2^16) and small block count (<=2^16)

Note: These conditions are used typically with UHS SD cards with 32-bit-wide CMD23 capability.

If software is modified to skip TC occurrence and proceed with data comparison, it appears that all data was received correctly.

WORKAROUND If we configure the card to send one more block (through the CMD23 argument) compared to what is configured in the host, then the Read transfer completes correctly (in this case, no DTO, TC occurs and data is correct).

REVISIONS IMPACTED SR 2.0, 1.1, 1.0

TDA2x: 2.0, 1.1, 1.0

DRA75x, DRA74x: 2.0, 1.1, 1.0

AM572x: 2.0, 1.1

i807 ***SATA Host Controller Locks Up if PIO Setup FIS Is Received and Bus Busy and Data Request Bits Are Cleared***

CRITICALITY Low

DESCRIPTION A bug in the SATA core is integrated into the SATA controller.

The host fails to proceed when receiving a D2H PIO setup FIS with bus busy (BSY) and data request (DRQ) bits cleared.

When the three following events occur simultaneously, the host controller fails to proceed and locks up:

- Host controller receives a PIO setup FIS (D=0/write):
  - SATA_PxIS[1] PSS = 0x1
- SATA_PxTFD[7] STS_BSY bit is cleared
- SATA_PxTFD[3] STS_DRQ bit is cleared

The bug is due to a state-machine in the SATA core that is not well implemented for this scenario.

A reset is required to continue communication between the host and the device.

From a user point of view, the impact can be some latency that is seen while proceeding.

WORKAROUND Implement a software time-out for locks and then issue one of the followoing two resets, first the least intrusive and/or more intrusive if it does not solve the lock.

Least intrusive, software reset:

- To issue a software reset, the user must prepare two H2D register FISs into the emptied command list of the port:
  - The first FIS must have bits SRST = 0b1 and C = 0b0. The first FIS corresponding command header bits C and R are set as follows: C = 0b1 and R = 0b1.
  - The second FIS has bits SRST = 0b0 and C = 0b0. The second FIS corresponding command header bits C and R are set as follows: C = 0b0 and R = 0b0.

More intrusive, Port reset (or COMRESET):

- SATA_PxSCTL[3:0] DET = 0x1

REVISIONS IMPACTED SR 2.0, 1.1, 1.0

TDA2x: 2.0, 1.1, 1.0

DRA75x, DRA74x: 2.0, 1.1, 1.0

AM572x: 2.0, 1.1

i808 ***SATA Link Locks Up Under Certain Conditions***

CRITICALITY Medium

DESCRIPTION A bug in the SATA core is integrated into the SATA controller.

Due to this bug, under certain conditions, the link locks up between the host and the device and the command times out:

- SATA_PxIS[27] IFS = 0x1 (interface fatal error status)
- SATA_PxSERR[23] DIAG_S = 0x1 (link sequence error)

Following cases can bring to these conditions (cases below are not exhaustive):

- First case:
  - D2H FIS is received;
  - D2H FIS SYNC-Escape is generated (Link illegal state transition/sequence error);
  - Link locks ups because this "bad scenario" is not well implemented;
  - Command times out;
  - Soft reset will not work because of the bug.
- Second case:
  - D2H FIS PIO Read CONTp primitive is corrupted;
  - Link locks ups because this "bad scenario" is not well implemented;
  - Command times out;
  - Soft reset will not work because of the bug.

WORKAROUND Use a port reset (COMRESET) instead of a software reset when SATA_PxIS[27] IFS = 0x1 and SATA_PxSERR[23] DIAG_S = 0x1.

Standard software most likely uses port reset rather than a software reset to recover from such an error.

REVISIONS IMPACTED SR 2.0, 1.1, 1.0

TDA2x: 2.0, 1.1, 1.0

DRA75x, DRA74x: 2.0, 1.1, 1.0

AM572x: 2.0, 1.1

i809	***SATA Command Does Not Complete and Software Must Issue a Port Reset Under Certain Conditions***
CRITICALITY	Medium
DESCRIPTION	When a Device-to-Host register FIS is received from the device and the FIS length exceeds eight DWORDs, the command may not complete due to an internal receive FIFO overflow condition. As a consequence, the host controller is locked and a latency is seen. The length of the FIS is specified by the specification and having more is a specification violation/error case. The issue is how a host controller is implemented.
WORKAROUND	A port reset (COMRESET) must be done to reestablish the communication between the host and the device.
REVISIONS IMPACTED	SR 2.0, 1.1, 1.0 TDA2x: 2.0, 1.1, 1.0 DRA75x, DRA74x: 2.0, 1.1, 1.0 AM572x: 2.0, 1.1

i810	***DPLL Controller Can Get Stuck While Transitioning to a Power Saving State***
CRITICALITY	Low
DESCRIPTION	NOTE: The previous title for this advisory was "DPLL Controller Sticks When Left Clock Requests Are Removed" The DPLL Controller can get stuck if it is in transition to a low power state while its M/N ratio is being programmed.
WORKAROUND	Before re-programming the M/N ratio, SW has to ensure the DPLL cannot start an idle state transition. SW can disable DPLL idling by setting the DPLL AUTO_DPLL_MODE=0 or keeping a clock request active by setting a dependent clock domain in SW_WKUP.
REVISIONS IMPACTED	SR 2.0, 1.1, 1.0 TDA2x: 2.0, 1.1, 1.0 DRA75x, DRA74x: 2.0, 1.1, 1.0 AM572x: 2.0, 1.1

i813 ***Spurious Thermal Alert Generation When Temperature Remains in Expected Range***

CRITICALITY Medium

DESCRIPTION Spurious Thermal Alert: Talert can happen randomly while the device remains under the temperature limit defined for this event to trig. This spurious event is caused by a incorrect re-synchronization between clock domains. The comparison between configured threshold and current temperature value can happen while the value is transitioning (metastable), thus causing inappropriate event generation.

No spurious event occurs as long as the threshold value stays unchanged. Spurious event can be generated while a thermal alert threshold is modified in CTRL_CORE_BANDGAP_THRESHOLD_MPU/GPU/CORE/DSPEVE/IVA.

WORKAROUND Spurious event generation can be avoided by performing following sequence when the threshold is modified:

1. Disable the alert interrupt: MPU_IRQ_126 into the interrupt handler.
2. Modify Threshold.
3. Clear the interrupt (cancel potential spurious event).
4. Enable the thermal alert interrupt again into the interrupt handler.

REVISIONS IMPACTED SR 2.0, 1.1, 1.0

TDA2x: 2.0, 1.1, 1.0

DRA75x, DRA74x: 2.0, 1.1, 1.0

AM572x: 2.0, 1.1

i814 ***Bandgap Temperature Read Dtemp Can Be Corrupted***

CRITICALITY Medium

DESCRIPTION Read accesses to registers listed below can be corrupted due to incorrect resynchronization between clock domains.

Read access to registers below can be corrupted:

- CTRL_CORE_DTEMP_MPU/GPU/CORE/DSPEVE/IVA_n (n = 0 to 4)
- CTRL_CORE_TEMP_SENSOR_MPU/GPU/CORE/DSPEVE/IVA

WORKAROUND Multiple reads to CTRL_CORE_TEMP_SENSOR_MPU/GPU/CORE/DSPEVE/IVA[9:0]:

BGAP_DTEMP_MPU/GPU/CORE/DSPEVE/IVA is needed to discard false value and read right value:

1. Perform two successive reads to BGAP_DTEMP bit field.
   a. If read1 returns Val1 and read2 returns Val1, then right value is Val1.
   b. If read1 returns Val1, read 2 returns Val2, a third read is needed.
2. Perform third read
   a. If read3 returns Val2 then right value is Val2.
   b. If read3 returns Val3, then right value is Val3.

Note: A maximum of three reads is required. Those three reads must be performed within the delay between two consecutive measurements, otherwise, methodology is not conclusive. This delay is configured in the COUNTER_DELAY field of CTRL_CORE_BANDGAP_MASK_1.

REVISIONS IMPACTED SR 2.0, 1.1, 1.0

TDA2x: 2.0, 1.1, 1.0

DRA75x, DRA74x: 2.0, 1.1, 1.0

AM572x: 2.0, 1.1

i815	***Power Management Enhancement Implemented Inside DSS Leads to DSS Underflows***
CRITICALITY	Medium
DESCRIPTION	An enhanced standby behavior is implemented inside DSS to avoid the usage of SW procedure, calculating the optimal DMA thresholds. This enhanced standby behavior allows DSS to go into standby even for cases where the threshold values programmed are non-optimal. A bug is identified in this implementation, which causes DSS underflows when DSS is in smart-standby mode and when multiple pipelines are enabled. An additional bit (bit 0) has been added to DISABLE_MSTANDBY_ENHANCEMENT register (physical address: 0x58001858) to enable(0x0)/disable(0x1) the enhanced standby behavior.
WORKAROUND	Setting DISABLE_MSTANDBY_ENHANCEMENT[0] bit to '1' (disables the enhanced standby behavior) and prevents the occurrence of this bug.
REVISIONS IMPACTED	SR 2.0, 1.1, 1.0 TDA2x: 2.0, 1.1, 1.0 DRA75x, DRA74x: 2.0, 1.1, 1.0 AM572x: 2.0, 1.1

i818	*SATA PHY Reset Required Following SATA PLL Unlock*
CRITICALITY	Medium
DESCRIPTION	If SATA controller is in slumber or partial low-power mode, SATA PHY is in low-power mode, and SATA 1.5 GHz PLL is relocked for any reason, the PHY receiver looses lock. In result the receiver / de-serializer is unable to produce parallel data from a correct serial source, and will not detect the attached SATA drive.
WORKAROUND	Workaround is to disable and re-enable both analog LDO of the transceiver, using the corresponding SW programmable bits of power control MMR: The CTRL_CORE_PHY_POWER_SATA[21:14] SATA_PWRCTL_CLK_CMD must be set to 0x0 to power down the SATA PHY TX and RX modules. The rest of the workaround sequence is the same as upon initial SATA PHY power-up, and includes setting above bits back to 0x2.
REVISIONS IMPACTED	SR 2.0, 1.1, 1.0 TDA2x: 2.0, 1.1, 1.0 DRA75x, DRA74x: 2.0, 1.1, 1.0 AM572x: 2.0, 1.1

i819	***A Device Control Bit Meta-Stability for USB3.0 Controller in USB2.0 Mode***
CRITICALITY	Medium
DESCRIPTION	When USB3.0 controller core is programmed to be a USB 2.0-only device, possible hardware meta-stability on USB_DCTL[31] RUNSTOP bit causing the core to attempt high speed as well as SuperSpeed connection or completely miss the attach request.
WORKAROUND	If the requirement is to always function in USB 2.0 mode, there is no workaround. Otherwise, you can always program the USB controller core to be SuperSpeed 3.0 capable (USB_DCFG[2:0] DEVSPD = 0x4).
REVISIONS IMPACTED	SR 2.0, 1.1, 1.0 TDA2x: 2.0, 1.1, 1.0 DRA75x, DRA74x: 2.0, 1.1, 1.0 AM572x: 2.0, 1.1

i820	***Unexpected USB Link State Value upon U3 Exit by USB3.0 Link***
CRITICALITY	Low
DESCRIPTION	When USB3.0 link is exiting U3 state (as result of link recovery request USB_DCTL[8:5] ULSTCHNGREQ=0x8), the USB_DSTS[21:18] USBLNKST field is updated to 0xF (Reset/ Resume State), which state is not a valid LTSSM state in this case.
WORKAROUND	Ignore the Reset/Resume State value (USB_DSTS[21:18] USBLNKST=0xF).
REVISIONS IMPACTED	SR 2.0, 1.1, 1.0 TDA2x: 2.0, 1.1, 1.0 DRA75x, DRA74x: 2.0, 1.1, 1.0 AM572x: 2.0, 1.1

i824 ***USB3.0 Link Cannot Be Established When Suspend Mode Is Enabled***

CRITICALITY Medium

DESCRIPTION When suspend mode is enabled (USB_GUSB3PIPECTL[17]SUSPENDENABLE = 0x1), the communication between the host controller and the USB device is broken. This happens in the following two sequences:

Sequence 1:

- Enable and configure the DWC_USB3 controller as host
- Set the USB_GUSB3PIPECTL[17] SUSPENDENABLE = 0x1 and USB_GUSB2PHYCFG[6] SUSPHY = 0x1
- Connect a USB3.0 device

When USB3.0 device is connected there is not enough time to establish the USB3.0 link so the host controller falls back to USB2.0 mode.

Sequence 2:

- Enable and configure the DWC_USB3 controller as host
- Wait until the USB3.0 link is established
- Set the USB_GUSB3PIPECTL[17] SUSPENDENABLE = 0x1 and USB_GUSB2PHYCFG[6] SUSPHY = 0x1
- Host controller initiates a transition to U3.
- Wait until transition U3 is completed successfully
- Host controller initiates a transition to U0.

The USB3.0 link cannot be reestablished and the host controller falls back to USB2.0.

WORKAROUND First set CTRL_CORE_PHY_POWER_USB[21:14] USB_PWRCTL_CLK_CMD to 0x43 then to 0x03 just after U3_EXIT state (setting the PORTSC2 in U0 by Polling on USB_PORTSC2[8:5] PLS=RESUME) and before Recovery state.

REVISIONS IMPACTED SR 2.0, 1.1, 1.0

TDA2x: 2.0, 1.1, 1.0

DRA75x, DRA74x: 2.0, 1.1, 1.0

AM572x: 2.0, 1.1

i826	***HSDIVIDER1 CLKOUT4 Could Glitch During On-the-Fly Divider Change to/from Divide-by-2.5***
CRITICALITY	Low
DESCRIPTION	When HSDIVIDER1/2 CLKOUT4 (CM_DIV_H14_DPLL_PER/ CM_DIV_H24_DPLL_CORE[5:0] DIVHS) configuration is changed between an odd divide value and divide-by-2.5 setting, the clock output could glitch. This could result in unexpected behavior of the peripheral receiving the clock. This case impacts GPU clock.
WORKAROUND	To avoid glitch, the user can always change first to/from an even divider setting, such as divide-by-4, before reconfiguring to divide-by-2.5 or from divide-by-2.5 to an odd divider. Sequence to switch HSDIVIDER1/2 CLKOUT4 from any odd divider to divide-by-2.5: • Current divider setting is set to any odd divider • Change divider setting to any even divider not exceeding maximum frequency for that clock (e.g. divide-by-4) • Change divider setting to divide-by-2.5 Sequence to switch HSDIVIDER1/2 CLKOUT4 from divide-by-2.5 to any odd divider: • Current divider setting is set divide-by-2.5 • Change divider setting to any even divider not exceeding maximum frequency for that clock (e.g. divide-by-4) • Change frequency to the desired odd divider
REVISIONS IMPACTED	SR 2.0, 1.1, 1.0 TDA2x: 2.0, 1.1, 1.0 DRA75x, DRA74x: 2.0, 1.1, 1.0 AM572x: 2.0, 1.1

i829 ***Reusing Pipe Connected to Writeback Pipeline On-the-Fly to an Active Panel***

CRITICALITY Low

DESCRIPTION Any pipe connected to writeback (WB) in memory-to-memory (m2m) mode (DISPC_WB_ATTRIBUTES[19] WRITEBACKMODE=0x1) cannot be connected on the fly to an active panel when m2m operation is complete. Trying to attempt this will cause sync-lost interrupt and one corrupted frame.

When a pipe is connected to WB pipeline in m2m mode, after m2m operation, it remains enabled. The HW does not disable the pipeline by clearing the enable bit associated with this pipeline (DISPC_VID1/2/3_ATTRIBUTES[0] ENABLE=0x0), though it disables the writeback by clearing the WB pipeline enable bit (DISPC_WB_ATTRIBUTES[0] ENABLE=0x0). If this pipe is then connected to an active panel, the connection will not be synchronized to a frame start. This will result in current frame getting corrupted and sync-lost.

WORKAROUND The SW should use following exit sequence from m2m operation:

1. When m2m operation is completed and hardware is automatically disabled writeback by setting DISPC_WB_ATTRIBUTES[0]ENABLE bit to 0x0, SW should disable the pipe connected to WB in m2m mode: DISPC_VID1/2/3_ATTRIBUTES[0] ENABLE = 0x0;
2. Writeback should be re-enabled (DISPC_WB_ATTRIBUTES[0]ENABLE=0x1) and after that disabled by SW (DISPC_WB_ATTRIBUTES[0] ENABLE = 0x0);
3. The direction of the pipe to the active panel should be changed and all new programming for the pipe should be made;
4. Pipe is enabled again (DISPC_VID1/2/3_ATTRIBUTES[0] ENABLE = 0x1) at the end,

REVISIONS IMPACTED SR 2.0, 1.1, 1.0

TDA2x: 2.0, 1.1, 1.0

DRA75x, DRA74x: 2.0, 1.1, 1.0

AM572x: 2.0, 1.1

i834 ***MMCHS HS200 and SDR104 Command Timeout Window Too Small***

CRITICALITY Medium

DESCRIPTION Under high speed HS200 and SDR104 modes, the functional clock for MMC modules will reach up to 192 MHz. At this frequency, the maximum obtainable timeout through of MMC host controller using MMCHS_SYSCTL[19:16] DTO = 0xE is (1/192MHz)*2^27 = 700ms. Commands taking longer than 700ms may be affected by this small window frame.

WORKAROUND If the command requires a timeout longer than 700ms, then the MMC host controller command timeout can be disabled (MMCHS_CON[6] MIT=0x1) and a software implementation may be used in its place. Detailed steps as follows (in Linux):

1. During MMC host controller probe function (omap_hsmmc.c:omap_hsmmc_probe()), inform processor that the host controller is incapable of supporting all the necessary timeouts.
2. Modify the MMC core software layer functionality so the core times out on its own when the underlying MMC host controller is unable to support the required timeout.

REVISIONS IMPACTED SR 2.0, 1.1, 1.0

TDA2x: 2.0, 1.1, 1.0

DRA75x, DRA74x: 2.0, 1.1, 1.0

AM572x: 2.0, 1.1

i837	***MMU1 Not Functional With EDMA***
CRITICALITY	Medium
DESCRIPTION	The system level MMU1 may hang when posted writes from EDMA are routed through the MMU.
WORKAROUND	None
REVISIONS IMPACTED	SR 1.0 TDA2x: 1.0 DRA75x, DRA74x: 1.0 AM572x: 1.0

i840	*DSP Trace Data Corruption*
CRITICALITY	Low
DESCRIPTION	DSP trace can send corrupt data and is thus not usable. At startup of trace stream of any pin width, random data could be sent and is not maskable. Then during trace after startup if 20 pin trace is used, 2 bits of the trace can be continuously corrupted.
WORKAROUND	None.
REVISIONS IMPACTED	SR 1.0 TDA2x: 1.0 DRA75x, DRA74x: 1.0 AM572x: 1.0

i841	***DCAN Ram Initialization Issue***
CRITICALITY	Medium
DESCRIPTION	To initialize the parity RAM of DCAN, SW must perform a subsequent write of 0b to the dcan*_raminit_start bit directly after writing the bit to 1b. Due to the bug, the subsequent write to 0b must complete no more than 64 DCAN ICLK periods after the write of 1b. If the time exceeds 64 cycles, the DCAN module could become unusable until the next device reset. A minimum toggle duration of 1 ICLK period is also required.
WORKAROUND	Software should implement the minimum toggle (write 1b then write 0b) to keep the pulse duration less than 64 ICLK cycles while keeping while still meeting the 1 ICLK minimum. The instructions should be executed from internal RAM to minimize latency variance due to internal bus traffic, interrupts, DDR refresh cycles, etc. It is difficult to ensure the less that 64 cycle limits under every system scenario, as it is sensitive to non-deterministic system latencies. Therefore the workaround should be used only for development and not in production.
REVISIONS IMPACTED	SR 1.0 TDA2x: 1.0 DRA75x, DRA74x: 1.0 AM572x: 1.0

i842	***Multiple Resets Required Before Chip Is Functional***
CRITICALITY	High
DESCRIPTION	Multiple resets (up to three) are needed after the initial PORz reset before the device is fully functional and allow cores to connect via emulation debugger. The resets can be cold (PORz) or warm (RESETn) pin resets, or emulation system resets issued from the debugger.
WORKAROUND	After initial PORz deassertion, wait at least 2.5ms and then issue a second and a third reset at least 2.5ms apart (either PORz or RESETn pin assertions at least 2.5ms duration or emulation system resets from the debugger).
REVISIONS IMPACTED	SR 1.0 TDA2x: 1.0 DRA75x, DRA74x: 1.0 AM572x: 1.0

i843 ***MMC1/2/3 Speed Issues***

CRITICALITY Medium

DESCRIPTION MMC1, MMC2 and MMC3 data write operations (single and multiple block writes) fail at speed due to timing issue with CRC response which causes the response to be latched as CRC bad status even if the status from the memory device was good. This limits MMC1 data write operations to 96MHz and MMC2/3 data write operations to 48MHz clock frequency. (See Figure 2-1 for MMCi Maximum Supported Frequencies).

	Maximum Frequency [MHz]				
	read operation		write operation		
	DM (target)	Actual	DM (target)	Actual	
				W/A 1	
MMC1	192	192	192	96	
MMC2	192	192	192	48	
MMC3	96	96	96	48	

Figure 2-1. MMCi Maximum Supported Frequencies

WORKAROUND Reduce clock frequency when performing data writes (96MHz for MMC1, 48MHz for MMC2, MMC3). Note, for MMC2, 48MHz DDR mode can be used for bandwidth equivalent to 96MHz. Optionally Increase to full frequency only for read operations. If mixed reads and writes are performed, then the lower frequency must be used.

REVISIONS IMPACTED SR 1.1, 1.0

TDA2x: 1.1, 1.0

DRA75x, DRA74x: 1.1, 1.0

AM572x: 1.1

i847	***PCIe TXP Output Drives 1/2 Amplitude for Some Data Bits***
CRITICALITY	High
DESCRIPTION	Some data patterns result in PCIe TXP transmit data output being driven only to half amplitude. This causes significant eye diagram marginality in both Gen1 and Gen2. Gen1 meets basic electrical requirements and loopback is functional. However, functional issues have been observed with external devices (failure to get into L0 state). Gen2 does not meet basic electrical requirements and is not functional.
WORKAROUND	None. Gen1 loopback may be used for development, but is not guaranteed to meet functional compliance.
REVISIONS IMPACTED	SR 1.0 TDA2x: 1.0 DRA75x, DRA74x: 1.0 AM572x: 1.0

i849	***UART2_RXD Is Not Working for MUXMODE=0***
CRITICALITY	Low
DESCRIPTION	The UART2_RXD functionality does not work when the pin is configured at its default of MUXMODE=0x0 in the corresponding CTRL_CORE_PAD* register.
WORKAROUND	Use MUXMODE=0x4 setting to select UART2_RXD function instead of MUXMODE=0x0 setting. The functionality is exactly equivalent.
REVISIONS IMPACTED	SR 2.0, 1.1, 1.0 TDA2x: 2.0, 1.1, 1.0 DRA75x, DRA74x: 2.0, 1.1, 1.0 AM572x: 2.0, 1.1

i852	*IODelay Recalibration Issue*
CRITICALITY	Medium
DESCRIPTION	If AVS voltage change on CORE_VD domain is used, there is a recalibration sequence which is required to ensure that IO timings meet timings per the Data Manual (software sequence to be included in TRM). This recalibration sequence is not reliable, and must be avoided. Therefore, IO timings may not be met after any CORE_VD voltage change. If AVS is not used on CORE_VD, the IO timings are fine since there is an automatic hardware calibration that happens one time directly after POR.
WORKAROUND	Keep the CORE_VD at the initial boot voltage. Power consumption may be slightly higher than optimal since AVS voltage is not used.
REVISIONS IMPACTED	SR 1.0 TDA2x: 1.0 DRA75x, DRA74x: 1.0 AM572x: 1.0

i854 ***EMIF CC 2b Error Can Cause Corrupt Internal Bus Read Response***

CRITICALITY Medium

DESCRIPTION For EMIF 2-bit Error Detection, errors are reported in one of two ways. First, the error is recorded in EMIF control registers and an error interrupt is asserted. Second, the EMIF returns a bus error on the read response status bus along with the read data via the internal bus protocol. The requesting master will receive the error code along with read data.

The EMIF control register and error interrupts correctly record/signal detection of 2-b errors, but the internal bus read response status for a transaction with a 2b error can sometimes be incorrect. The incorrect response occurs due to a logic bug that results in slot21 in the EMIF's read FIFO using the error status of slot20 for its read response status (slot21 and slot20 can contain unrelated transactions from the same or different masters). (Note that there are 32-entries in the read response FIFO that are all used in round-robin order. Slot 20 and 21 are used with equal probability relative to the remaining FIFO slot entries).

Two scenarios are possible:

1. If slot 20 transaction had an error and slot 21 didn't, then the ECC error Interrupt will be triggered and read response status returned for both slot 20 and 21 will indicate an error. Thus slot 21 will indicate a false error.
2. If slot 21 transaction had an error and slot 20 didn't, then the ECC error Interrupt will be triggered and read response status returned for both slot 20 and 21 will indicate no error. Thus the true error status of slot 21 will not be directly communicated back to the requesting master via the read response bus along with the corresponding read data.

Since the read response status for detection of 2-b errors may be incorrect, this may cause some masters to consume erroneous data and/or cause a false exception on valid data. If no 2b error has occurred in the system, there is no impact.

WORKAROUND Software should rely on EMIF error interrupt (EMIF_SYSTEM_OCP_INTERRUPT_RAW_STATUS[4] TWOBIT_ECC_ERR_SYS/ EMIF_SYSTEM_OCP_INTERRUPT_STATUS[4] TWOBIT_ECC_ERR_SYS=0x1) and error address control registers(EMIF_2B_ECC_ERR_ADDR_LOG[31:0] REG_2B_ECC_ERR_ADDR) to comprehend 2-b ECC errors in the system. It is still advantageous to enable 2b ECC (EMIF_SYSTEM_OCP_INTERRUPT_ENABLE_SET[4] TWOBIT_ECC_ERR_SYS) versus leaving it off, as statistically many errors would still be properly handled.

REVISIONS IMPACTED SR 1.0

TDA2x: 1.0

DRA75x, DRA74x: 1.0

AM572x: 1.0

i855	***QSPI Mode0/1/2 Not Functional and Mode3 Limited to 48MHz***
CRITICALITY	Medium
DESCRIPTION	QSPI Mode 0 (Rev 1.0) and Mode 1,2 (rev 1.0, 1.1, and 2.0) are not usable due to hold time issues. Further, the usable frequency limit on QSPI Mode3 is 48MHz (Rev 1.0).
WORKAROUND	1. Rev 1.0, only use QSPI Mode3, and the frequency must be 48MHz or less. 2. Rev 1.1, only use QSPI Mode0 or Mode3 up to the frequency, specified in DM. 3. Rev 2.0, only use QSPI Mode0 or Mode3 up to the frequency, specified in DM.
REVISIONS IMPACTED	SR 2.0, 1.1, 1.0 TDA2x: 2.0, 1.1, 1.0 DRA75x, DRA74x: 2.0, 1.1, 1.0 AM572x: 2.0, 1.1

i856 ***32k Oscillator Fails to Start-Up at POR***

CRITICALITY Medium

DESCRIPTION The on-chip 32k oscillator fails to start up after release of power-on-reset input (porz).

SYSBOOT[9:8]=00b mode is not usable with internal 32k oscillator as the device will not boot. This means internal only oscillator frequencies of 20MHz, 27MHz or 19.2MHz may be used. Startup of 32k depends on user software writing to RTCSS GZ bit and only RTCSS can use the 32k clock (after writing GZ bit to 0), all other FUNC_32K_CLK targets (e.g. timers, mmc and gpio de-bounce) can only use SYSCLK1. Full features of RTC mode are still functional after the 32k software enables the oscillator.

WORKAROUND Avoid SYSBOOT[9:8]=00b setting. This means internal only oscillator frequencies of 20MHz, 27MHz or 19.2MHz may be used. This also means that the 32k_FUNC clock within the device (feeding MMC, GPIO, Timers) will always come from SYSCLK1/610 and not the true 32k clock, which will only feed the RTCSS.

Note

NOTE: For legacy systems, if use of 26MHz clock is required, then SYSBOOT[9:8]=00b may be used by feeding a 32kHz clock to the rtc_osc_xi_clking32 pin from an external source. In this case, the 32k clock from the pad will also be used as 32k FUNC clock to the device. Note: This operation is not defined by the Data Manual.

REVISIONS IMPACTED SR 2.0, 1.1, 1.0

TDA2x: 2.0, 1.1, 1.0

DRA75x, DRA74x: 2.0, 1.1, 1.0

AM572x: 2.0, 1.1

i859	*SATA 6-Gbps to 3-Gbps Negotiation Can Fail*
CRITICALITY	Medium
DESCRIPTION	The device is specified to support only 3-Gbps speed. Per the standard, when used with 6-Gbps capable devices, a speed negotiation sequence must take place. During speed negotiation sequence with a 6-Gbps device, the SATA controller may potentially misinterpret a valid 3-Gbps stream and attempt to establish link even though the external SATA device is sending at 6-Gbps speed. If this occurs, the link controller will be at an invalid state requiring a reset to recover.
WORKAROUND	There is no direct workaround to reliably prevent the invalid state during gen3 negotiation. For development, the user may retry multiple times to attempt to establish the link. However, a robust connection requires avoiding the 6-Gbps negotiation completely. This can be done by: 1. Use device which supports only 3-Gbps or lesser speed. 2. Use device which is configurable via jumper setting to support only 3-Gbps or lesser speed. 3. Use a device which has pre-configured non-volatile setting to support only 3-Gbps or lesser speed. Preconfigure the non-volatile speed setting offline with another device or via port multiplier which does not have such issue with 6-Gbps negotiation. 4. If the device does not support features as mentioned in 1, 2, 3, then connect to the device via a port multiplier which allows explicit configuration for 3-Gbps or lesser speed on the device side.
REVISIONS IMPACTED	SR 1.0 TDA2x: 1.0 DRA75x, DRA74x: 1.0 AM572x: 1.0

i861	***QSPI-4 Boot Mode Is Not Functional***
CRITICALITY	Medium
DESCRIPTION	QSPI-4 boot modes do not function. The ROM code writes to a non-volatile bit (QE bit) to enable quad mode and then attempts the first read access before the flash device is ready. The flash does not respond and bad data is read and executed by the ROM. The flash device models used for verification failed to model this characteristic of the flash properly.
WORKAROUND	QSPI-1 boot mode can be used with the CH (configuration header) feature to switch to 4-bit mode at 48MHz after the first 512byte is loaded. This requires changing the sysboot pulls to select a QSPI-1 mode and including the appropriate configuration header values in the boot image. QSPI-4 mode is not usable without this workaround requires Sysboot pull change on board to select QSPI-1 mode, slightly longer boot time since first 512bytes are loaded at single-bit 12MHz mode (additional time is ~0.5ms if CH configures for 48MHz quad mode).
REVISIONS IMPACTED	SR 1.0 TDA2x: 1.0 DRA75x, DRA74x: 1.0 AM572x: 1.0

i862 ***Reset Should Use PORz***

CRITICALITY High

DESCRIPTION Power-on-reset (porz SoC input signal) is the only 100% reliable reset type. If any reset source other than porz is used, there is a chance the SoC may hang during boot after the reset source is de-asserted. Examples of other reset sources include software resets (global cold, global warm), hardware exception resets (Watchdog, Thermal Shutdown, Security violations), or the Warm Reset input (resetn SoC input). Entry into reset will be successful with these reset sources, but code execution may hang if reset is initiated by any reset source other than porz.

Two examples: A watchdog reset will indicate a runaway code event has occurred by resetting the SoC and asserting rstoutn. A thermal shutdown reset (TSHUT) will reset the SoC and assert rstoutn which prevents the SoC from overheating. However, code execution my hang when the SoC attempts to reboot from any source other than porz (including a watchdog and thermal shutdown reset).

Power-On-Reset (porz SoC input) is 100% reliable and can recover from the SoC hang.

WORKAROUND PORz should be used for all reset occurrences.

Two recommended implementations are provided below. Note: All reset sources will assert reset to the system via the SoC rstoutn output. This allows external visibility to software or watchdog resets, which would otherwise be invisible to components outside of the SoC. Both recommended implementations will use the rstoutn output.

Implementation 1: PMIC asserts porz when rstoutn is connected to PMIC NRESWARM input

- When the rstoutn output from the SoC is connected to the external PMIC's NRESWARM input, the PMIC companion device approved for use with the SoC can be configured to detect the rstoutn/NRESWARM assertion and assert porz/RESET_OUT. All PMIC companion devices which have been approved for the SoC implement this feature. The feature is bootstrap selectable via one of the PMIC's BOOT pin(s). Refer to PMIC User Guide for additional details. Note: This implementation option has no added cost to the customer since the SoC must be used with one of the approved PMIC devices.
- To implement the workaround:
  - Connect the rstoutn output from the SoC to the PMIC's NRESWARM input (and to any other components that need to reset when the SoC undergoes a reset). Note: When the rstoutn output is operating in 3.3V mode, a 3.3 volt to 1.8 volt level translator will be required to level shift the rstoutn output connected to the PMIC’s NRESWARM input to 1.8 volts.
  - Pull-up the appropriate PMIC BOOTx pin, to configure the PMIC’s RESET_OUT to assert porz on warm reset.
  - The PMIC’s POWERHOLD (GPIO7) input must be pulled high.
- Example use cases for this implementation include:
  - A switch connected to the PMIC’s POWERHOLD input is used to turn the board on/off.
  - The PMIC applies power to the SoC as soon as the board is powered when the POWERHOLD input is tied high to an always-on supply LDOVRTC_OUT.
  - The PMIC applies power to the SoC once the PWRON input is pulled low by pressing a normally open push-button switch when the POWERHOLD input is pulled high by one of the supplies enabled during device start-up.
- The side effects/risks of this implementation include:

  - This implementation does not allow software to shut down the PMIC outputs that power the SoC. Only the PMIC RESET_IN can shut down the PMIC outputs while POWERHOLD is pulled high.
  - Risk of exceeding the 200 hour limit defined by Advisory i863, if the PMIC applies power with eMMC in contention longer than 200 hours.

Implementation 2: Additional circuit implemented that generates porz without PMIC support

- This implementation enables software shutdown of the PMIC since the PMIC's POWERHOLD input remains low during operation.
- To implement the workaround:
  - Pull-down the appropriate PMIC's BOOTx input.
  - Use an external circuit that generates a finite length active low pulse to porz when the circuit detects the assertion of rstoutn. This feedback path from rstoutn through the pulse generating circuit to porz insures any reset source other than porz generates a valid reset for the SoC.
- Example use cases for this implementation include:
  - A normally open push button switch (on the system board) connected to the PMIC's PWRON input is used to initiate PMIC applying power to the device.
  - Software writes to the PMIC registers to power off the device.
- The benefits/side effects of this implementation include:
  - This implementation allows software to shut down the PMIC since the PMIC's POWERHOLD input remains low during operation.
  - Reduces the risk described in Advisory i863. This implementation will automatically shut-off power to the SoC seven seconds after the PMIC's PWRON event unless software writes to appropriate registers to remove contention from the eMMC signals before writing to appropriate PMIC registers that allows the SoC to remain powered.

Other implementations are also possible. For instance, an external watchdog timer could be implemented to assert porz when the SoC becomes unresponsive.

In general, any valid workaround that generates a porz whenever any reset is initiated has the following side effects:

- Reset status information is lost in PRM_RSTSTAT register.
  - Visibility into the cause of the last reset is lost. To maintain some visibility software may be able to store information in PMIC BACKUP or other PMIC registers.
- Ethernet Reset isolation feature is not supported.
- Boot device reordering on warm reset is not supported.

The workaround has the advantage of guaranteeing the entire SoC is in a known good and consistent state for every reboot. For example, there are no software residual effects due to watchdog warm reset.

REVISIONS IMPACTED

SR 2.0, 1.1

This erratum is fixed on TDA2x SR 2.0. However, the i862 workaround may still be required for some use cases. Refer to i727 and i729 for more details.

TDA2x: 2.0, 1.1

DRA75x, DRA74x: 2.0, 1.1

AM572x: 2.0, 1.1

i863 ***MMC2 Has PU/PD Contention Immediately after Release from Reset***

CRITICALITY High

DESCRIPTION On SR1.x, the MMC2_DAT[x] terminals have internal weak pull-down resistors (PD) in the range of 8 kΩ ~ 36 kΩ (1.8 V) or 9 kΩ ~ 82 kΩ (3.3 V) which are turned on by default immediately after the device has been released from reset. The JEDEC eMMC standard requires external weak pull-up resistors (PU) on eMMC CMD and DAT signals, and internal weak pull-up resistors on DAT[7-0] terminals of eMMC devices to prevent inputs from floating. The external resistors are in the range of 4.7 kΩ ~ 100 kΩ and the internal eMMC device resistors are in the range of 10 kΩ ~ 150 kΩ. After reset, these weak pull-up resistors contradict the internal pull state of the device and presents a PU/PD contention on the eMMC DAT signals; this may lead to reliability issues if not handled properly.

On SR2.x, the MMC2 DAT[x] terminals have internal weak pull-down resistors (PD) which are permanently disabled when SYSBOOT15=1 or enabled by default when SYSBOOT15=0. Refer to the TRM section “Permanent PU/PD disabling” for details.

PU/PD Contention Reliability Issue:

The PU/PD contention applies a mid-supply voltage to the input buffer which may cause excessive current to flow through the input buffer. In this scenario, both FETs (PMOS/NMOS) in the input buffer are partially turned ON, resulting in a current path from VDD through the input buffer to VSS. Total leakage power during this state may be up to 800 µA per input buffer operating at 1.8 V, or up to 2 mA per input buffer operating at 3.3 V. Hysteresis on the input buffers prevents the noise from causing the input logic level to change state, but it does not prevent the current path.

To maintain system reliability, SW should minimize the duration eMMC DAT lines spend in this invalid state.

WORKAROUND SW should minimize the time eMMC DAT terminals spend in the PU/PD contention state to a maximum of 200 hours in a device life cycle.

On SR1.x, this is done by configuring MMC pinmux configuration to turn off the internal pull-down resistors as early as possible in secondary boot loader (SBL, i.e. the initial software image loaded by the device’s ROM boot loader; one that is responsible for loading subsequent boot images or the main OS). If external pulls are not implemented on the PCB, then the internal PU on eMMC DAT signals should be enabled simultaneously. Alternately, if external pulls are implemented (as recommended by the JEDEC JESD84-B451) the internal pull resistors can be disabled. SW should take care of writing the below values in the listed registers-bit fields:

```
CTRL_CORE_PAD_GPMC_A24[3:0] = 1   ; mmc2_dat0
CTRL_CORE_PAD_GPMC_A24[16] = 0    ; Enables weak Pull Up/Down
CTRL_CORE_PAD_GPMC_A24[17] = 1    ; Pull Up is selected
CTRL_CORE_PAD_GPMC_A24[18] = 1    ; Receive Mode is Enabled
CTRL_CORE_PAD_GPMC_A24[19] = 0    ; Fast slew is selected
CTRL_CORE_PAD_GPMC_A25[3:0] = 1   ; mmc2_dat1
CTRL_CORE_PAD_GPMC_A25[16] = 0    ; Enables weak Pull Up/Down
CTRL_CORE_PAD_GPMC_A25[17] = 1    ; Pull Up is selected
CTRL_CORE_PAD_GPMC_A25[18] = 1    ; Receive Mode is Enabled
CTRL_CORE_PAD_GPMC_A25[19] = 0    ; Fast slew is selected
CTRL_CORE_PAD_GPMC_A26[3:0] = 1   ; mmc2_dat2
CTRL_CORE_PAD_GPMC_A26[16] = 0    ; Enables weak Pull Up/Down
CTRL_CORE_PAD_GPMC_A26[17] = 1    ; Pull Up is selected
CTRL_CORE_PAD_GPMC_A26[18] = 1    ; Receive Mode is Enabled
CTRL_CORE_PAD_GPMC_A26[19] = 0    ; Fast slew is selected
CTRL_CORE_PAD_GPMC_A27[3:0] = 1   ; mmc2_dat3
CTRL_CORE_PAD_GPMC_A27[16] = 0    ; Enables weak Pull Up/Down
```

```
CTRL_CORE_PAD_GPMC_A27[17] = 1    ; Pull Up is selected
CTRL_CORE_PAD_GPMC_A27[18] = 1    ; Receive Mode is Enabled
CTRL_CORE_PAD_GPMC_A27[19] = 0    ; Fast slew is selected
CTRL_CORE_PAD_GPMC_A19[3:0] = 1   ; mmc2_dat4
CTRL_CORE_PAD_GPMC_A19 [16] = 0    ; Enables weak Pull Up/Down
CTRL_CORE_PAD_GPMC_A19 [17] = 1    ; Pull Up is selected
CTRL_CORE_PAD_GPMC_A19 [18] = 1    ; Receive Mode is Enabled
CTRL_CORE_PAD_GPMC_A19 [19] = 0    ; Fast slew is selected
CTRL_CORE_PAD_GPMC_A20[3:0] = 1   ; mmc2_dat5
CTRL_CORE_PAD_GPMC_A20[16] = 0    ; Enables weak Pull Up/Down
CTRL_CORE_PAD_GPMC_A20[17] = 1    ; Pull Up is selected
CTRL_CORE_PAD_GPMC_A20[18] = 1    ; Receive Mode is Enabled
CTRL_CORE_PAD_GPMC_A20[19] = 0    ; Fast slew is selected
CTRL_CORE_PAD_GPMC_A21[3:0] = 1   ; mmc2_dat6
CTRL_CORE_PAD_GPMC_A21[16] = 0    ; Enables weak Pull Up/Down
CTRL_CORE_PAD_GPMC_A21[17] = 1    ; Pull Up is selected
CTRL_CORE_PAD_GPMC_A21[18] = 1    ; Receive Mode is Enabled
CTRL_CORE_PAD_GPMC_A21[19] = 0    ; Fast slew is selected
CTRL_CORE_PAD_GPMC_A22[3:0] = 1   ; mmc2_dat7
CTRL_CORE_PAD_GPMC_A22[16] = 0    ; Enables weak Pull Up/Down
CTRL_CORE_PAD_GPMC_A22[17] = 1    ; Pull Up is selected
CTRL_CORE_PAD_GPMC_A22[18] = 1    ; Receive Mode is Enabled
CTRL_CORE_PAD_GPMC_A22[19] = 0    ; Fast slew is selected
```

The 200 hours can be distributed any way throughout the lifetime of a device, and can be one instance of 200 hours or any number of occurrences totaling 200 hours.

On SR2.x, if SYSBOOT15=1 then no software workaround is required since the internal pulls are permanently disabled. Note that external pull-up resistors on the MMC data bus are mandatory in this case. It is OK if the software workaround remains since accesses to configure the internal pulls has no effect.

REVISIONS IMPACTED

SR 1.1, 1.0

SR 2.0 (if SYSBOOT15=0, as described in i863 above)

TDA2x: 2.0, 1.1, 1.0

DRA75x, DRA74x: 2.0, 1.1, 1.0

AM572x: 2.0, 1.1

i868 ***McASP to EDMA Synchronization Level Event Can Be Lost***

CRITICALITY Medium

DESCRIPTION The McASP FIFO events to the EDMA or System DMA can be lost depending on the timing between the McASP side activity and the DMA side activity. The problem is most likely to occur in a heavily loaded system which can cause the DMA latency to increase and potentially hit the problematic timing window. When an event is lost, the McASP FIFO Rx path will overflow or the Tx path will underflow. Software intervention is required to recover from this condition.

The issue results due to a state machine boundary condition in the McASP FIFO logic. In normal operation, when "Threshold" (set by the RFIFOCTL[15:8] RNUMEVT and WFIFOCTL[15:8] WNUMEVT registers) words of data are read/written by the DMA then the previous event would be cleared. Similarly, when "Threshold" words of data are written/read from the pins, a new event should be set. If these two conditions occur at the same exact time (within a 2 cycle window), then there is a conflict in the set/clear logic and the event is cleared but is not re-asserted to the DMA.

WORKAROUND Since the McASP is a real time peripheral, any loss of data due to underflow/overflow should be avoided by eliminating the possibility of DMA read/write completing at the same time as a new McASP Event. Software should configure the system to:

1. Maximize time until the deadline for the McASP FIFO
2. Minimize DMA service time for McASP related transfers

In order to maximize time until deadline, the RNUMEVT and WNUMEVT should be set to the largest multiple of "number of serializers active" that is less-than-equal-to 32 words. Since the FIFO is 64-Words deep, this gives the maximum time to avoid the boundary condition.

In order to minimize DMA service time for McASP related transfers multiple options are possible. For example, McASP buffers can be placed in OCMCRAM or DSP's L2 SRAM (since on chip memories provide a more deterministic and lower latency path compared to DDR memory). In addition, a dedicated Queue/TC can be allocated to McASP transfers. At minimum, care should be taken to avoid any long transfers on the same Queue/TC to avoid head-of-line blocking latency.

REVISIONS IMPACTED SR 1.1, 1.0

TDA2x: 1.1, 1.0

DRA75x, DRA74x: 1.1, 1.0

AM572x: 1.1

i869	***IO Glitches Can Occur When Changing IO Settings***
CRITICALITY	Medium
DESCRIPTION	Glitches up to multiple nano-seconds in length can occur on a Device IO when changing the IO setting via either of the below methods: 1. Changing the value of the MUXMODE, DELAYMODE or MODESELECT fields of the corresponding CTRL_CORE_PAD_* register. 2. Changing the value of the CFG_*_IN, CFG_*_OEN, and CFG_*_OUT registers in the IODELAYCONFIG Module.
WORKAROUND	To workaround this issue, the Device LVCMOS IOs should be placed into Isolation mode when changing the IO settings as described above. Refer to the Device TRM section "Isolation Requirements" for more details.
REVISIONS IMPACTED	SR 2.0, 1.1, 1.0 TDA2x: 2.0, 1.1, 1.0 DRA75x, DRA74x: 2.0, 1.1, 1.0 AM572x: 2.0, 1.1

i870	***PCIe Unaligned Read Access Issue***
CRITICALITY	Medium
DESCRIPTION	Access to the PCIe slave port that are not 32-bit aligned will result in incorrect mapping to TLP Address and Byte enable fields. Therefore, byte and half-word accesses are not possible to byte offset 0x1, 0x2, or 0x3.
WORKAROUND	To workaround this issue, there are two options: 1. Avoid issuing read accesses to the PCIe slave port that are not 32-bit aligned 2. Set the PCIE_SS1_AXI2OCP_LEGACY_MODE_ENABLE and PCIE_SS2_AXI2OCP_LEGACY_MODE_ENABLE bits to 0x1 in the CTRL_CORE_SMA_SW_7 Control Module register. This will make all Read TLPs 32-bit aligned with all byte enables set to 1.
REVISIONS IMPACTED	SR 2.0, 1.1, 1.0 TDA2x: 2.0, 1.1, 1.0 DRA75x, DRA74x: 2.0, 1.1, 1.0 AM572x: 2.0, 1.1

i871 ***L4_PER3 Firewall Initiator ConnID Value Left-Shift 1-Bit***

CRITICALITY Low

DESCRIPTION The initiator ConnID value is used for configuring Firewall to setup the protection of target IPs. The Technical Reference Manual Table "L4 ConnID Definition" is correct for L4_CFG, L4_PER1, L4_PER2, and L4_WKUP. However, the L4_PER3 initiator ConnID value propagated from L3 interconnect is left-shifted by 1-bit comparing to the other L4 instances. As a result, the L4_PER3 firewall uses ConnIDx2 to determine the permission of each initiator accessing L4_PER3 targets. L4_PER3 firewall also logs error with ConnIDx2 value when violations occur.

Due to ConnID value left-shifted by 1-bit, the default Protection Group Member, defined by L4_AP_PROT_GROUP_MEMBERS_0_L register, will only enable the initiators with ConnID < 8 for L4_PER3 access since the upper 16-bits are all 0s after reset. The initiators with ConnID >= 8 won't be able access L4_PER3 targets out of reset. User has to program the protection group register to enable the access to L4_PER3 targets for those initiators.

WORKAROUND To configure the L4_PER3 firewall protection correctly, use ConnIDx2 to program protection group and identify the initiator that caused the error. Refer to the below table for L4_PER3 ConnID values.

Table 2-2. ConnID Values

ConnID per TRM (hex)	ConnID For L4_PER3 FW (decimal)	CONNID_BIT_VECTOR For L4_PER3 FW	Initiator(1)
0	0	BIT 0	Cortex-A15 MPU subsystem
1	2	BIT 2	Debug subsystem
2	4	BIT 4	DSP1 subsystem (CFG, EDMA, MDMA), DSP2 (EDMA)
3	6	BIT 6	IVAHD, DSP2 (CGF, MDMA)
4	8	BIT 8	EVE1, EVE2, EVE3, EVE4
5	10	BIT 10	PRUSS1, PRUSS2
6	12	BIT 12	IPU1/2, SYSTEM_DMA
7	14	BIT 14	EDMA
8	16	BIT 16	DSS, MLB, MMU1, MMU2, PCIE1 and PCIE2
9	18	BIT 18	VIP1, VIP2, VIP3, VPE
A	20	BIT 20	MMC1, MMC2, GPU, BB2D, GMAC
B	22	BIT 22	USB1, USB2, USB3, USB4
C	24	BIT 24	SATA

(1) For the list of active initiators, please refer to the specific device TRM.

REVISIONS IMPACTED SR 2.0, 1.1, 1.0

TDA2x: 2.0, 1.1, 1.0

DRA75x, DRA74x: 2.0, 1.1, 1.0

AM572x: 2.0, 1.1

i872 *DSP MFlag Output Not Initialized*

CRITICALITY

Medium

DESCRIPTION

The DSP1 and DSP2 Subsystems include MFlag output signals that are under DSP software control and are used to control arbitration at various points in the system interconnect, including in the DMM and EMIF command queues.

Each DSP subsystem's MFlag output signal is uninitialized in hardware until the DSP is powered up and clocked, and can default to a value of either 0 or 1. This can have unanticipated and non-deterministic effects on system traffic dependent on the power-on state of the MFlag signals.

WORKAROUND

In order to ensure that a known value is driven by the DSP's MFlag outputs, software should power-up the DSP(s) and enable the clocks for a brief time. After the DSP is enabled, it can immediately be disabled if desired. Once the DSP is enabled and clocked the MFlag output will be 0.

The sequence to perform a DSPn enable and then power down is as below:

```
/* Start a SW force wakeup for DSPSS */
WR_MEM_32(CM_DSPn_CLKSTCTRL, 0x2);

/* Enable DSPSS clock */
WR_MEM_32(CM_DSPn_DSPn_CLKCTRL, 0x1);

/* Reset de-assertion for DSP SS logic */
WR_MEM_32(RM_DSPn_RSTCTRL, 0x1);
/* Wait till module is functional*/
while ((RD_MEM_32(CM_DSPn_DSP_CLKCTRL) & 0x30000) != 0x0 or TIMEOUT(100ms));
/* Make the DSPn CLK CTRL to HW auto */
WR_MEM_32(CM_DSPn_CLKSTCTRL, 0x3);
/* Make the DSPn POWER domain to go to power off mode */
WR_MEM_32(PM_DSPn_PWRSTCTRL, (RD_MEM_32(PM_DSPn_PWRSTCTRL)& 0xFFFFFFF0));
/* Disable DSPSS clock */
WR_MEM_32(CM_DSPn_DSPn_CLKCTRL, 0x0);
/* Reset assertion for DSP SS logic */
WR_MEM_32(RM_DSPn_RSTCTRL, 0x3);
```

This sequence should be performed even for devices where one or both DSPs are not supported.

The timeout value shown in the while loop is recommended as a software best practice. The poll for completion should always succeed before the timeout expires.

REVISIONS IMPACTED

SR 2.0, 1.1, 1.0

TDA2x: 2.0, 1.1, 1.0

DRA75x, DRA74x: 2.0, 1.1, 1.0

AM572x: 2.0, 1.1

i874	*TIMER5/6/7/8 Interrupts Not Propagated*
CRITICALITY	Low
DESCRIPTION	When TIMER5, TIMER6, TIMER7, or TIMER8 clocks are enabled (CM_IPU_TIMER5/6/7/8_CLKCTRL[1:0] MODULEMODE=0x2:ENABLE) and the CD-IPU is in HW_AUTO mode (CM_IPU_CLKSTCTRL[1:0] CLKTRCTRL=0x3:HW_AUTO) the corresponding TIMER will continue counting, but enabled interrupts will not be propagated to the destinations (MPU, DSP, etc) in the SoC until the TIMER registers are accessed from the CPUs (MPU, DSP etc.). This can result in missed timer interrupts.
WORKAROUND	In order for TIMER5/6/7/8 interrupts to be propagated and serviced correctly the CD_IPU domain should be set to SW_WKUP mode (CM_IPU_CLKSTCTRL[1:0] CLKTRCTRL=0x2:SW_WKUP)
REVISIONS IMPACTED	SR 2.0, 1.1, 1.0 TDA2x: 2.0, 1.1, 1.0 DRA75x, DRA74x: 2.0, 1.1, 1.0 AM572x: 2.0, 1.1

i875 ***Power-on-Reset (PORz) Warm Boot Hang***

CRITICALITY

High

DESCRIPTION

Following a warm Power-on-Reset (PORz) event, in which the supplies to the SoC remain ON throughout the assertion of PORz, the boot process may hang if the SoC is within a narrow temperature range.

The sensitive temperature range can be different on each device, is typically ~5ºC wide, and typically occurs below 25ºC.

The boot issue occurs only when the PORz signal is asserted to the SoC without turning-off the supplies to the SoC. No issue is observed in normal cold-boot operation in which the SoC boots up from a full power-off condition.

This erratum does not occur in typical use case scenarios. The boot hang may occur only if the PORz event is generated as a consequence of some other issue (e.g., run-time reset from an external MCU, or an SoC watchdog timer expiration resulting in PORz assertion) while the SoC temperature is within a narrow range.

The issue results from improper power sequencing to an internal SRAM that is accessed during execution of the early stages of the boot process. During assertion of PORz, the SoC's on-chip LDO regulator supplying the SRAM tri-states its output, thus allowing internal supplies to the SRAM to drift down uncontrolled. This can result in improper sequencing as the LDO reapplies power to the SRAM upon de-assertion of PORz before internal supplies to the SRAM have fully ramped down. This situation can result in incorrect accesses to the SRAM during boot, causing the boot process to halt.

WORKAROUND

A board level workaround requires adding a 220 Ohm (+/- 5%) resistor onto the SRAM LDO supply (Cap_vddram_mpu1, ball K16 on the SoC package with ABC designator (that is, the 23 mm package)), Cap_vddram_mpu1, ball F15 on package with AAS designator (that is, the 17 mm package)). This resistor provides a controlled discharge path for the charge contained within the external 1µF LDO capacitor during the reset operation. The workaround effectiveness assumes that the active duration of the PORz signal is a typical 3.4 ms or greater.

In systems with an MCU present, a second workaround technique can be utilized. The MCU performs a handshake with the device following a warm PORz to ensure that the device is responsive after the reset. On the occasion a hang occurs, the MCU should assert PORz low for 200 ms. This eliminates the need for an external 220 Ohm resistor.

REVISIONS IMPACTED

SR 1.1, 1.0

TDA2x: 1.1, 1.0

DRA75x, DRA74x: 1.1, 1.0

AM572x: 1.1

i878 ***MPU Lockup With Concurrent DMM and EMIF Accesses***

CRITICALITY High

DESCRIPTION The MPU has two primary paths to DDR and system address space via the MPU Memory Adapter (MPU_MA).

The Low Latency path is the predominant path for DDR accesses and provides direct/low latency/interleaved access to the two EMIFs.

The L3 Interconnect path (via MPU_AXI2OCP bridge) is most typically used for access to non-DDR address space, but is also used for access to DMM and EMIF control registers and to Tiled regions of DDR address space.

Issue is seen to come when there is a heavy memory access through the MPU L3 path, if the MPU is concurrently issuing write transactions via the Low Latency path to DDR and via the L3 Interconnect to the DMM/EMIF/Tiler address space then the transactions can hang and the MPU and DMM/DDR become unresponsive. A device reset is required in order to recover from this condition.

WORKAROUND In order to completely avoid the issue, the MPU can avoid concurrent accesses to the DMM/EMIF/DDR address space via the Low Latency path and the L3 Interconnect path. In order to accomplish this, the MPU should avoid use of the L3 Interconnect path via the MPU by using DSP, IPU, or DMA to proxy accesses to the EMIF/DMM registers or Tiler DDR address space.

In order to greatly reduce the probability of the issue occurring, the MPU_MA register at 0x482AF400 bits 2 and 1 can be set, that is, 0x482AF400 |= 0x6. With this setting of MPU_MA register, the 3 different heavily loaded application scenario which earlier reproduced the issue was seen working fine for long duration testing.

Note: The MPU_MA register is a valid register address location even though it is located outside the MPU memory space as specified in the device TRM.

REVISIONS IMPACTED SR 2.0, 1.1, 1.0

TDA2x: 2.0, 1.1, 1.0

DRA75x, DRA74x: 2.0, 1.1, 1.0

AM572x: 2.0, 1.1

i879	***DSP MStandby Requires CD_EMU in SW_WKUP***
CRITICALITY	Low
DESCRIPTION	Issue is seen to come when there is need to place the DSP subsystem to a low power state. The DSP requires the internal emulation clock to be actively toggling in order to successfully enter a low power mode via execution of the IDLE instruction and PRCM MStandby/Idle handshake. This assumes that other prerequisites and software sequence are followed.
WORKAROUND	The CD_EMU domain can be set in SW_WKUP mode via the CM_EMU_CLKSTCTRL[1:0] CLKTRCTRL field. The emulation clock to the DSP is free-running anytime CCS is connected via JTAG debugger to the DSP subsystem or when the CD_EMU clock domain is set in SW_WKUP mode. **Note:** If it is sure that the DSP would never enter any low power state (in other words the DSP would never execute IDLE instruction), the workaround can be ignored.
REVISIONS IMPACTED	SR 2.0, 1.1, 1.0 TDA2x: 2.0, 1.1, 1.0 DRA75x, DRA74x: 2.0, 1.1, 1.0 AM572x: 2.0, 1.1

i880	***Ethernet RGMII2 Limited to 10/100 Mbps***
CRITICALITY	High
DESCRIPTION	RGMII2 (rgmii1_* signals) Switching Characteristics only support 10/100 Mbps operation. RGMII2 Switching Characteristics are not compatible with 1000 Mbps operation. Refer to the device Data Manual for Switching Characteristics timing details.
WORKAROUND	RGMII2 (rgmii1_* signals) should be used at 10/100 Mbps operation only. If 1000 Mbps operation is required, RGMII1 (rgmii0_* signals) can be used instead.
REVISIONS IMPACTED	SR 1.1, 1.0 TDA2x: 1.1, 1.0 DRA75x, DRA74x: 1.1, 1.0 AM572x: 1.1

i882 ***EMIF: DDR ECC Corrupted Read/Write Status Response***

CRITICALITY

High

DESCRIPTION

For ECC-enabled DDR regions, only full quanta aligned writes are allowed (16-b quanta for 16-b data bus, and 32-b quanta for 32-b data bus). The EMIF has a mechanism to detect illegal sub-quanta writes to ECC-enabled space. It has two methods of reporting such errors. In case of such an illegal write:

1. An error interrupt will be generated to the CPU.
2. The EMIF will give an error status response to the offending initiator on the internal bus.

In some corner conditions the internal bus response (method #2 above) may become corrupted and give a false read or write error response in spite of all transactions in the EMIF command fifo being valid/legal. Depending on how each initiator handles the specific internal bus error response, this can result in system instability. For instance, a false error response on an MPU read may result in an abort.

The error interrupt behavior (method #1 above) is always correct. No false error interrupts are created, and only true illegal writes to ECC space are reported via interrupt. Access type and initiator for the last offending access will be logged.

WORKAROUND

Disable ECC.

OR

Enable ECC for desired ranges in EMIF1, and ensure that all DDR write accesses to all of EMIF1 (including ECC protected and unprotected ranges) from all initiators are a multiple of quanta size and are quanta aligned.

REVISIONS IMPACTED

SR 1.1, 1.0

TDA2x: 1.1, 1.0

DRA75x, DRA74x: 1.1, 1.0

AM572x: 1.1

i883	***DSP Doesn't Wake from Subsystem Internal Interrupts***
CRITICALITY	Medium
DESCRIPTION	When the C66x DSP CorePac enters a low power state (via the IDLE instruction and setting the Power-Down Controller Command Register (PDCCMD) bit 16) and the DSP subsystem remains active (e.g., EDMA is still active), the DSP should be able to wake from any interrupt source including EDMA completion interrupts. However, the DSP Internal IRQs (mapped to evt_in[31:16]) are unable to wake the DSP from a sleep/IDLE state, whereas DSP External IRQs (from the SoC IRQ_Crossbar) (mapped to evt_in[95:32]) are able to wake the DSP.
WORKAROUND	The EDMA Completion Interrupts (DSPi_IRQ_TPCC_REGION[7:0] and DSPi_IRQ_TPCC_GLOBAL) are mapped to DSP Internal IRQs, and are also provided as outputs from the DSP subsystem and are mapped as inputs to the IRQ_CROSSBAR. In order to allow the C66x DSP CorePac to wake from a low power state when a subsystem EDMA interrupt is asserted, the desired interrupt can be mapped via the IRQ_CROSSBAR to one of the DSP External IRQs.
REVISIONS IMPACTED	SR 2.0, 1.1, 1.0 TDA2x: 2.0, 1.1, 1.0 DRA75x, DRA74x: 2.0, 1.1, 1.0 AM572x: 2.0, 1.1

i884	***MMC4 Speed Limited to 38.5 MHz***
CRITICALITY	Low
DESCRIPTION	MMC4 Switching Characteristics only support 38.5 MHz, not 48 MHz as per original target. Refer to the device Data Manual for Switching Characteristics timing details.
WORKAROUND	MMC4 interface should be used at a maximum of 38.5 MHz. The other MMC interfaces [3,2,1] support higher clock rates.
REVISIONS IMPACTED	SR 1.1, 1.0 TDA2x: 1.1, 1.0 DRA75x, DRA74x: 1.1, 1.0 AM572x: 1.1

i887	***MMC3 Speed Limited to 64 MHz***
CRITICALITY	Low
DESCRIPTION	MMC3 Switching Characteristics only support 64 MHz, not 96 MHz as per original target. Refer to the device Data Manual for Switching Characteristics timing details.
WORKAROUND	MMC3 interface should be used at a maximum of 64 MHz. The other MMC interfaces [2,1] support higher clock rates.
REVISIONS IMPACTED	SR 2.0, 1.1, 1.0 TDA2x: 2.0, 1.1, 1.0 DRA75x, DRA74x: 2.0, 1.1, 1.0 AM572x: 2.0, 1.1

i889 *UART Does Not Acknowledge Idle Request after DMA Has Been Enabled*

CRITICALITY

Medium

DESCRIPTION

All UART modules in the SoC do not acknowledge an idle request after enabling the module's DMA feature, even if the DMA is subsequently disabled. Thus, the UART module cannot be clock idled after enabling DMA with

- UART_SCR.DMA_MODE_CTL = 1 and UART_SCR.DMA_MODE_2 != 0

  OR

- UART_SCR.DMA_MODE_CTL = 0 and UART_FCR.DMA_MODE = 1

A consequence of this is that UARTx_CLKCTRL will remain in transition when trying to disable the module (UARTx_CLKCTRL = 0x10000) and the associated CLKACTIVITY bit will remain active.

WORKAROUND

Initiating a soft reset (UART_SYSC.SOFTRESET = 1) will allow the module to acknowledge the idle request.

REVISIONS IMPACTED

SR 2.0, 1.1, 1.0

TDA2x: 2.0, 1.1, 1.0

DRA75x, DRA74x: 2.0, 1.1, 1.0

AM572x: 2.0, 1.1

i890	***MMC1 IOs and PBIAS Must Be Powered-Up before Isolation***
CRITICALITY	Medium
DESCRIPTION	IO Isolation, as described in TRM section “Isolation Requirements”, may fail if the MMC1 IOs and PBIAS are not powered-up.
WORKAROUND	Power-up the MMC1 IOs and PBIAS before starting the Isolation Sequence. This can be done by setting the CTRL_CORE_CONTROL_PBIAS[27] SDCARD_BIAS_PWRDNZ and CTRL_CORE_CONTROL_PBIAS[26] SDCARD_IO_PWRDNZ bits to 1.
REVISIONS IMPACTED	SR 2.0, 1.1, 1.0 TDA2x: 2.0, 1.1, 1.0 DRA75x, DRA74x: 2.0, 1.1, 1.0 AM572x: 2.0, 1.1

i893 ***DCAN Initialization Sequence***

CRITICALITY Low

DESCRIPTION If the DCAN module is allowed to enter/exit clock-gated mode dynamically while traffic is present on the DCAN interface (even if the traffic is not to/from the SoC) then the DCAN module and PRCM handshake state machines can become out of sync resulting in the DCAN module hanging.

WORKAROUND In order to cleanly initialize the DCAN module the following sequence should be followed. Steps 1 and 2 can happen in any order, but should occur before Step 3.

1. Configure the DCAN module's clock domain in SW_WKUP mode
   - DCAN1: CM_WKUPAON_CLKSTCTRL. CLKTRCTRL = 0x2
   - DCAN2: CM_L4PER2_CLKSTCTRL. CLKTRCTRL= 0x2
2. Configure CD_L4_CFG for NO_SLEEP mode
   - CM_L4CFG_CLKSTCTRL. CLKTRCTRL = 0x0
3. Execute RAM Init Sequence:
   - Mask the RX input via pinmux configuration
     - Select default/gpio function instead of dcan rx. Specific register and MUXMODE value depends on pin-mux used on the board
     - For DCAN1 muxed with WAKEUP0: CTRL_CORE_PAD_WAKEUP0.MUXMODE = 0xF
   - Enable DCAN module
     - DCAN1: CM_WKUPAON_DCAN1_CLKCTRL.MODULEMODE = 0x2
     - DCAN2: CM_L4PER2_DCAN2_CLKCTRL.MODULEMODE = 0x2
   - Perform RAM_INIT sequence
     - DCAN1: CTRL_CORE_CONTROL_IO_2. DCAN1_RAMINIT_START = 0x1
     - DCAN2: CTRL_CORE_CONTROL_IO_2. DCAN2_RAMINIT_START = 0x1
     - Poll for CTRL_CORE_CONTROL_IO_2. DCAN1_RAMINIT_DONE and DCAN2_RAMINIT_DONNE
   - Enable RX input via pin mux configuration
     - Select dcan_rx function
     - Specific register and MUXMODE value depends on pin-mux used on the board
     - For DCAN1 muxed with WAKEUP0: CTRL_CORE_PAD_WAKEUP0.MUXMODE = 0x1

REVISIONS IMPACTED SR 2.0, 1.1, 1.0

TDA2x: 2.0, 1.1, 1.0

DRA75x, DRA74x: 2.0, 1.1, 1.0

AM572x: 2.0, 1.1

i895 ***EMIF_FW: System Hang When EMIF Firewall Is Reconfigured While There Is Activity on EMIF Interface***

CRITICALITY High

DESCRIPTION If there is ongoing activity on the EMIF interface via the DMM path, any reconfiguration of EMIF_FW firewall instance can result in a system hang. This applies only to the EMIF_FW firewall instance. All other firewall instances can be safely re-configured at run-time using the sequence mentioned in TRM.

To reconfigure the firewall configuration at run-time, following sequence is to be followed:

1. Set the REGUPDATE_CONTROL[0] BUSY_REQ bit to 0x1 to ensure that no transaction can reach the slave NIU (suspend).
2. Update the firewall registers as required.
3. Set the REGUPDATE_CONTROL[0] BUSY_REQ bit to 0x0 to resume the transactions.

In case of EMIF_FW firewall instance, the transactions to the slave (EMIF) are not blocked correctly in step 1 mentioned above. Initiator remains unaware that EMIF is not accepting commands and waits indefinitely for a response resulting in a system hang.

WORKAROUND EMIF_FW firewall instance should not be reconfigured when there is on-going activity on the EMIF interface.

Following choices can be made based on practical constraints in the system.

1. All firewall configurations are done one-time during the initial boot-up phase where EMIF transactions can be easily guaranteed to be not active. This will typically be done by a secondary boot loader or any other initialization software.
2. Using inter-processor communication between all the cores in the system, ensure that there is no activity on the EMIF interface. At this point, firewall can be reconfigured safely.
3. For some use-cases, to achieve same effect as changing firewall configuration at run-time, user can use a combination of DMM/MMU/Firewalls. In this case, firewall configuration is done only once during boot. Access permissions are controlled using aliased memory map created using DMM and controlling access to this aliased memory map using MMU.

REVISIONS IMPACTED SR 2.0, 1.1, 1.0

TDA2x: 2.0, 1.1, 1.0

DRA75x, DRA74x: 2.0, 1.1, 1.0

AM572x: 2.0, 1.1

i896	***USB xHCI Port Disable Feature Does Not Work***
CRITICALITY	Low
DESCRIPTION	USB software would typically disable the port if the attached device is behaving incorrectly or has become unresponsive. Due to the bug if the port is disabled using the PED bit in the PORTSC register, it subsequently fails to detect any detach/attach events. In other words, if the attached device were to be disconnected after it has been disabled using PED bit, the USB controller will be unable to detect and report it. This issue only applies to HighSpeed mode.
WORKAROUND	Option 1: If possible, use of Port Disable should be completely avoided. Since its use is only in error scenarios for eg. unresponsive devices, this is feasible. Option 2: Instead of disabling the port, power off the port using PP and then power it back on. Now the port can detect attach/detach events again. To ensure that same scenario does not occur over and over, the erroneous device must be removed before attempting to recognize and enumerate a device again.
REVISIONS IMPACTED	SR 2.0, 1.1, 1.0 TDA2x: 2.0, 1.1, 1.0 DRA75x, DRA74x: 2.0, 1.1, 1.0 AM572x: 2.0, 1.1

i897 ***USB xHCI Stop Endpoint Command Does Not Work in Certain Circumstances***

CRITICALITY Low

DESCRIPTION USB xHCI Stop Endpoint feature can, in addition to other scenarios, be used to recover from a situation where a particular endpoint of the attached device has become unresponsive due to various reasons. By stopping the endpoint, the xHCI host would normally stop all ongoing communication with the said endpoint. Other endpoints are not affected.

Due to the bug, in the following scenario, the Stop Endpoint will not function correctly:

- USB Controller connected to a Highspeed hub through zero or more hubs.
- FullSpeed device connected to any of the above hubs.
- Assume the USB Controller has already enumerated the Hub.
- USB Controller now starts to enumerate the FullSpeed device. In the Address Device step, BSR is set to 1.
- Next it performs GetDescriptor (device not addressed yet since BSR=1).
- Suppose that the FullSpeed device has is faulty and does not respond to the GetDescriptor Setup stage.
- The GetDescriptor is now stuck in CSPLIT-SETUP-NYET.
- At this point the application intends to recover from this error.

The Stop Endpoint when issued at this point never completes and the xHCI Controller will continuous send the CSPLIT-SETUP command since it never receives an ACK. In addition an attempt to abort the command using the CA bit in CRCR register also does not succeed.

WORKAROUND Option 1: In the Address Device command set BSR=0 so that SET_ADDRESS command is actually executed before GetDescriptor.

Option 2: Instead of Stop Endpoint, perform the Disable Slot command. The difference from Stop Endpoint is that this command shall stop all endpoints and interfaces associated with the device.

REVISIONS IMPACTED SR 2.0, 1.1, 1.0

TDA2x: 2.0, 1.1, 1.0

DRA75x, DRA74x: 2.0, 1.1, 1.0

AM572x: 2.0, 1.1

i898 ***DSP Pre-fetch Should Be Disabled before Entering Power Down Mode***

CRITICALITY Medium

DESCRIPTION The DSP may hang after multiple iterations of going into C66x Corepac Power Down and wake up from external events.

The C66x XMC (External Memory Controller) can have outstanding pre-fetch requests when C66x Corepac transitions to a Power Down state. The XMC clocks are gated internally during this transition. While XMC clocks are gated, outstanding pre-fetch request responses are not seen by the XMC which leads to an inconsistent state between the XMC and the L3 Interconnect. When the DSP wakes up, this can manifest as different symptoms within the DSP subsystem, including Cache corruption, incorrect data being returned to the CPU, and can eventually lead to a DSP hang condition.

WORKAROUND The steps to avoid this issue are as given below:

1. Ensure the code which places the DSP C66x Corepac to Power Down State (power down entry procedure shown below) is placed in the DSP C66x L2 RAM memory.
2. Set the IDLE bit in PDCCMD register during initialization.
3. Inside the power down entry procedure include the following software sequence:
   a. Execute MFENCE instruction.
   b. Write 1 to XPFCMD.INV (address 0x0800_0300).
   c. Read XPFACS (address 0x0800_0304).
   d. Execute IDLE instruction.

While executing multi-threaded DSP software with C66x Corepac Power Down caution should be observed to not allow the power down entry sequence to be preempted and switch context.

The software developer can choose to not perform the above software sequence by never enabling the DSP C66x Pre-fetch. The developer should understand the impact of not enabling DSP Pre-fetch on the DSP CPU memory access performance in their application.

REVISIONS IMPACTED SR 2.0, 1.1, 1.0

TDA2x: 2.0, 1.1, 1.0

DRA75x, DRA74x: 2.0, 1.1, 1.0

AM572x: 2.0, 1.1

i899 ***Ethernet DLR Is Not Supported***

CRITICALITY Low

DESCRIPTION The DLR function is comprised of two separate functions that act together to implement DLR. The first is DLR packet detection and priority escalation. The second is DLR unicast address detection and packet forwarding.

DLR packet detection should correctly detect that a DLR packet with no VLAN, a single VLAN, or two VLAN's has a DLR LTYPE. The packet should then be sent to the highest transmit FIFO priority of each destination egress port FIFO. In the case that the host port is the egress port, the packet should also be transferred to memory on the DLR channel.

DLR unicast address detection should match a unicast destination address and flood the packet to the VLAN minus the receive port and minus the host port. For a 3-port switch, a DLR unicast packet that is received (ingress) on an Ethernet port would be sent to the other Ethernet port. A DLR unicast packet that was received via the host port would be flooded to both Ethernet ports.

DLR cannot be enabled because the switch will enter an unknown state upon detection of a DLR packet. DLR unicast addresses can be added to the address table and will correctly flood to the VLAN minus the receive port and minus the host port. However, since DLR detection is dependent on enabling DLR (DLR_EN bit) and such enablement is precluded due to the bug, no DLR packet detection or priority escalation can occur.

WORKAROUND None.

REVISIONS IMPACTED SR 2.0, 1.1, 1.0

TDA2x: 2.0, 1.1, 1.0

DRA75x, DRA74x: 2.0, 1.1, 1.0

AM572x: 2.0, 1.1

i900 ***SoC Will Hang If Region 5 Accessed While CTRL_CORE_MMR_LOCK_5 Is Locked***

CRITICALITY Medium

DESCRIPTION CTRL_CORE_MMR_LOCK_5 register has unexpected behavior.

There are five registers used to lock different memory regions of CTRL_MODULE_CORE memory space. A memory region is locked, means that all write accesses to this region are ignored. Writing a value unique for each register will lock certain memory region and writing another unique value results in unlocking of the same region.

The functionality of CTRL_CORE_MMR_LOCK_5 register is different than the other 4 registers.

If a write access to “locked” registers, which belong to MMR_LOCK_5 region is performed, all of the Control Module registers become inaccessible. Any write access to locked registers in MMR_LOCK_5 region leads to an error in Control Module interface bus.

Therefore, the write access is not only ignored but also blocks further access to the Control Module forever.

WORKAROUND For accessing Control Module's configuration registers belonging to MMR_LOCK_5 region by CTRL_CORE_MMR_LOCK_5 register the following sequence must be used:

1. Check if CTRL_CORE_MMR_LOCK_5 is locked - 0x143F832C. If yes, unlock CTRL_CORE_MMR_LOCK_5 as write 0x6F361E05.
2. Modify the selected CTRL_CORE_PAD_x registers.
3. Lock the CTRL_CORE_MMR_LOCK_5. The register is locked as write 0x143F832C.
4. Do not write CTRL_CORE_PAD_x registers when CTRL_CORE_MMR_LOCK_5 is locked. This leads to an error in Control Module interface bus.

REVISIONS IMPACTED SR 2.0, 1.1, 1.0

TDA2x: 2.0, 1.1, 1.0

DRA75x, DRA74x: 2.0, 1.1, 1.0

AM572x: 2.0, 1.1

i901	***DSS VOUT3 on VDDSHV6 Domain (vin1a Pins) Should Not Be Used in 3.3V Mode***
CRITICALITY	High
DESCRIPTION	When the VOUT3 interface is mapped to VDDSHV6 supply domain (i.e., muxed on vin1a pins) and is used in 3.3V mode of operation there can be excessive overshoot/undershoot on the IOs when the interface data bus switches concurrently. This can impact the long term reliability of the SoC.
WORKAROUND	The VOUT3 interface should not be used in 3.3V mode of operation when mapped to the VDDSHV6 domain (i.e., muxed on the vin1a pins). A different pin mux option should be chosen. Options include using VOUT3 on the VDDSHV10 domain (muxed with gpmc pins) or using a different video port (VOUT1, VOUT2). If a different pin mux option is not possible, then VOUT3 can be used in 1.8 V mode without impacting long-term reliability of the SoC. A level shifter can be implemented on the board to convert between 1.8V levels on the SoC side and 3.3V levels if needed by the external component.
REVISIONS IMPACTED	SR 2.0, 1.1, 1.0 TDA2x: 2.0, 1.1, 1.0 DRA75x, DRA74x: 2.0, 1.1, 1.0 AM572x: 2.0, 1.1

i903	***Ethernet RMII Interface RMII_MHZ_50_CLK Not Supported as Output Reference Clock***
CRITICALITY	High
DESCRIPTION	The Ethernet EMAC module when operating in RMII mode has two clocking modes. In one case, a clock is generated externally and is an input to the SoC via the RMII_MHZ_50_CLK pin. This mode of operation functions properly and the timing specified in the Data Manual is valid. In the second case, the SoC drives a 50 MHz clock as an output on the RMII_MHZ_50_CLK pin. This output drives the clock to the external PHY. This mode of operation does not meet the timing specified in the Data Manual; and the resulting timing is not compatible with the RMII standard.
WORKAROUND	When using either of the Ethernet ports in RMII mode (pins rmii0* or rmii1*) the RMII_MHZ_50_CLK signal must be configured as an input, and the clock must be generated external to the SoC. The internal clock generation mode is not supported. The following registers should be set to configure RMII_MHZ_50_CLK as an input: CM_GMAC_GMAC_CLKCTRL[CLKSEL_REF] = 1 CTRL_CORE_SMA_SW_6[RMII_CLK_SETTING] = 1 Alternatively, the Ethernet EMAC module supports MII or RGMII protocols/pins on both ports. Those modes can be used if the selected PHY supports them. The typical clocking modes for those interfaces are able to meet the timing specified in the Data Manual for 100 Mbps operation (which is the rate supported by RMII).
REVISIONS IMPACTED	SR 2.0, 1.1, 1.0 TDA2x: 2.0, 1.1, 1.0 DRA75x, DRA74x: 2.0, 1.1, 1.0 AM572x: 2.0, 1.1

i916 ***QSPI Reads Can Fail For Flash Devices with HOLD Function***

CRITICALITY

High

DESCRIPTION

The default internal pull-up/pull-down on the SoC QSPI interface can interfere with the HOLD function implemented in some QSPI FLASH devices leading to Read Failures. This is most likely to be seen at higher clock rates, and with EDMA reads of greater than 128-Bytes.

In Quad SPI mode, the SoC QSPI IP transmits the command and address to the flash device on data line D0 and reads the data back on all four data lines D0, D1, D2 and D3. The default values of the data lines i.e. values when there is no driver on lines are LOW for D0 and D1 and HIGH for D2 and D3. These values are dictated by the internal and external pull ups.

When the last bit on the last read driven by the FLASH doesn't match the 'default value', the data lines D1, D2 and D3 transition slowly to their default values i.e. LOW for D1 and HIGH for D2 and D3. The transition time is in the order of 100 ns and depends on board loadings. At higher frequencies (typically above 64MHz QSPI clock rate) the time from the last bit of data transfer to the first bit of the next command is not long enough to allow for the pull-ups to get the data lines D1, D2 and D3 to the desired state. It is possible that the D3 line is still in a LOW state when the next command transmission begins.

The D3 line is used by some flash devices as a HOLD signal. If the D3 line has not reached HIGH state by the time CS is reasserted, flash devices can infer that a HOLD is in effect and fail to service the current command.

This issue is most easily seen with EDMA reads of length greater than 128 bytes. CPU reads typically provide sufficient time between reads for the data lines to reach their default values.

WORKAROUND

The software workaround is to disable the hold functionality on the QSPI device, preferably by setting a nonvolatile configuration register. On most flash devices, placing the QSPI device in Quad read mode automatically disables the HOLD functionality. On certain flash devices, there is a separate mode bit that can be set to disable the HOLD functionality. Typically this setting would be done in a Flash Writer utility that programs the flash with the customer’s boot image and sets appropriate non-volatile mode bits. Depending on the software architecture, this mode bit setting may also be done in the boot-loader or HLOS kernel.

Disabling the HOLD functionality prevents the slow ramp on the D3 line from interrupting the operation of the QSPI flash device and allows EDMA reads at high clock speeds (64 MHz).

There are no negative effects of the workaround as HOLD functionality is not supported by SoC QSPI IP.

REVISIONS IMPACTED

SR 2.0, 1.1, 1.0

TDA2x: 2.0, 1.1, 1.0

DRA75x, DRA74x: 2.0, 1.1, 1.0

AM572x: 2.0, 1.1

i927	***SoC Doesn't Read Redundant ONFI Parameter Pages in NAND Boot Mode***
CRITICALITY	Low
DESCRIPTION	In NAND boot mode if an ONFI NAND is detected, the SoC Boot ROM reads the NAND flash geometry from the first parameter page without checking the integrity CRCs. If there are CRC errors in the page, the ROM could use incorrect information about the NAND geometry and could fail to boot. The ONFI standard indicates that the CRC should be checked and in case of CRC errors a redundant parameter page should be used.
WORKAROUND	Instead of using ONFI NAND, a non-ONFI NAND can be used. Note that when selecting a specific non-ONFI NAND, the Read ID and Geometry of the NAND should be compared to the supported configurations documented in the device Technical Reference Manual.
REVISIONS IMPACTED	SR 2.0, 1.1, 1.0 TDA2x: 2.0, 1.1, 1.0 DRA75x, DRA74x: 2.0, 1.1, 1.0 AM572x: 2.0, 1.1

i929 *MMC1/2 SDR104/HS200 Mode DLL Delay Value May Result In Unexpected Tuning Pattern Errors*

CRITICALITY Low

DESCRIPTION Internal to the MMC module, a second stage latch is used to recapture data captured by DLL delayed CLK, MMC_DLL_CLK. The second stage latch captures with the original transmitting clock, MMC_CLK.

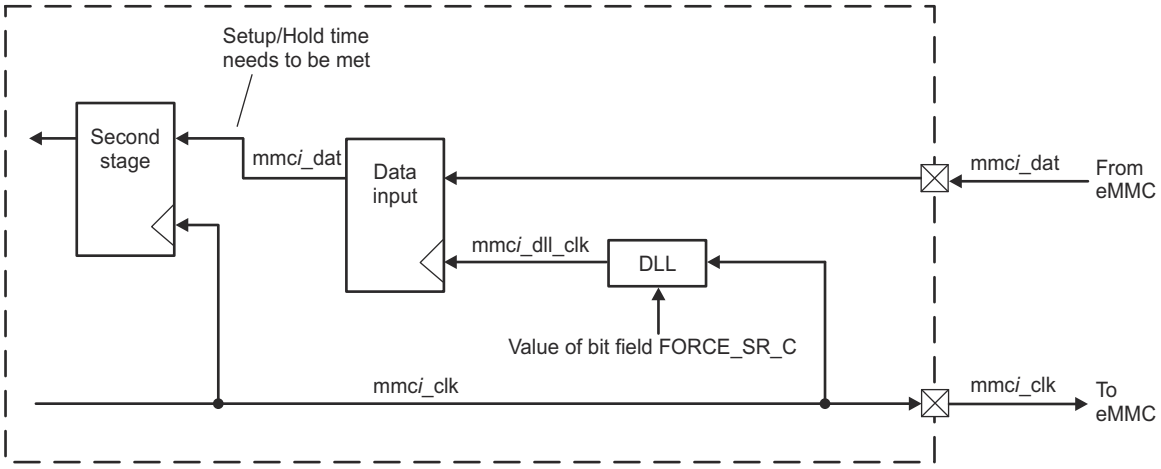

Figure 2-2. Simplified SoC 192-MHz Mode DLL Block Diagram

MMC_DLL_CLK and MMC_CLK both run at the same clock frequency. This results in a narrow range of tuning ratio elements, where the delayed MMC_DLL_CLK comes in phase with MMC_CLK. If the clocks are in phase, the data captured by the first clock violates the setup and hold time requirements needed for the second stage latch, resulting in incorrectly read data. This is known as tuning re-timing errors.

For systems in which MMC DLL tuning algorithm* choses a ratio less than 40, which is sufficiently far from the lowest re-timing error ratio element, no workaround is necessary.

WORKAROUND A DLL tuning algorithm has been implemented that can avoid the tuning re-timing errors. More details on this can be found in App Note SPRACA9. The following notes summarize the updated algorithm:

1. Implement two stage tuning. The software begins with the regular tuning algorithm, using 4-step increments, to optimize boot time. When the initial ratio is chosen within the largest passing window, the software checks 10 tuning steps in each direction, using single steps, to identify whether the chosen ratio is at risk of a tuning re-timing error. If at risk, the value of the chosen ratio is adjusted to move away from the error. If not, the chosen ratio is used unchanged.
2. Choose ratio based on temperature. Both tuning band errors and tuning re-timing errors shift with temperature. The software takes this dependency into consideration when selecting the tuning ratio element to use for functionality.

Note

NOTE: *Legacy MMC DLL tuning algorithm are algorithms that were implemented before errata i929 was published. These algorithms do not take temperature nor single step tuning into consideration and were only tuned with step size = 4.

REVISIONS IMPACTED SR 2.0, 1.1, 1.0

TDA2x: 2.0, 1.1, 1.0

DRA75x, DRA74x: 2.0, 1.1, 1.0

AM572x: 2.0, 1.1

i930	***I2C1 and I2C2 May Drive Low During Reset***
CRITICALITY	Low
DESCRIPTION	While the SoC PORz signal is asserted, one or more I2C1 and I2C2 IOs (i2c1_scl, i2c1_sda, i2c2_clk, i2c2_sda) may drive low. The Data Manual specifies that these signals should be high-z during PORz assertion. This occurs due to an internal node floating to a random state inside of the I2C output buffer during PORz assertion. Note that other I2C instances on the SoC are not affected by this issue since they use a different I/O buffer.
WORKAROUND	This issue has not resulted in any known issues in systems. Any workaround may be dependent on the characteristics of connected devices in a given system, and the external device(s) response in case a Start/Stop sequence occurs without an intermediate I2C handshake. If the I2C devices connected to I2C1 or I2C2 are sensitive to a spurious Start/Stop sequence during SoC PORz assertion, then an external switch can be implemented on a PCB between the SoC SDA/SCL signals and the external I2C component(s). The switch can be controlled by a GPIO output of the SoC. The GPIO signal will be high-z during PORz and a pull-resistor should be used to cause the external switch to be open during PORz. After PORz deassertion, software can enable the GPIO to close the switch prior to using the I2C1 or I2C2 interface.
REVISIONS IMPACTED	SR 2.0, 1.1, 1.0 TDA2x: 2.0, 1.1, 1.0 DRA75x, DRA74x: 2.0, 1.1, 1.0 AM572x: 2.0, 1.1

i932 ***DPLL_VIDEOn May Require Multiple Lock Attempts***

CRITICALITY Medium

DESCRIPTION In rare circumstances the DPLL_VIDEO1 and DPLL_VIDEO2 PLLs may not lock on the first attempt during SoC initialization. When this occurs a subsequent attempt to relock the PLL will result in the PLL successfully locking.

WORKAROUND In order to successfully lock the PLL, the following software sequence is recommended:

1. Boot SoC.
2. Configure DPLL_VIDEO1.
3. Set PLL_GO[0] PLL_GO bit to trigger Lock sequence.
4. Poll for Status bits in Locked, Non-error state OR Wait for Timeout (1000 REFCLK cycles):
   - PLL_STATUS[6] PLL_BYPASS = 0
   - PLL_STATUS[5] PLL_HIGHJITTER = 0
   - PLL_STATUS[3] PLL_LOSSREF = 0
   - PLL_STATUS[2] PLL_RECAL = 0
   - PLL_STATUS[1] PLL_LOCK = 1
   - PLL_STATUS[0] PLLCTRL_RESET_DONE = 1
5. If Fail to Lock/Error -> Set PLL_GO[0] PLL_GO bit again; Repeat up to 20 times.

REVISIONS IMPACTED SR 2.0, 1.1, 1.0

TDA2x: 2.0, 1.1, 1.0

DRA75x, DRA74x: 2.0, 1.1, 1.0

AM572x: 2.0, 1.1

i933 ***Access to IODELAY at Same Time as Other Peripheral on L4_PER2 Can Hang***

CRITICALITY Medium

DESCRIPTION If read/write accesses are performed concurrently from one initiator to the IODELAY module address space and one initiator to another peripheral address space in the L4_PER2 segment of the L4 interconnect then the access to the IODELAY module can hang, leading to an overall system hang. The concurrent accesses may be from two different initiators, or could be from one initiator capable of issuing multiple transactions through the interconnect. In this context, initiator can be a compute core (MPU, DSP, IPU, etc) or a DMA/Master peripheral (EDMA, SDMA, etc.)

The hang occurs due to a protocol violation on the interconnect OCP bus when responses from the IODELAY module and other module on the L4_PER2 segment occur on the same cycle.

The condition which hangs the system can be avoided by performing all IODELAY configurations during initial MPU boot, before other initiators are enabled. This approach may be acceptable for many peripherals, but may pose limitations for a few peripherals. For example, this approach may limit data transfer speeds of an SD Card or other device attached to the MMCn interface since IODELAY normally changes when the transfer mode is changed during run-time. In this example, the hang may occur if other initiators are accessing peripherals on L4_PER2 while IODELAY is changed to support a new SD Card or MMC transfer mode.

The following peripherals are connected to L4_PER2 and should not be accessed while IODELAY configuration is modified: UART7, UART8, UART9, MCASP4_DAT, MCASP5_DAT, MCASP6_DAT, MCASP7_DAT, MCASP8_DAT, MCASP1_CFG, MCASP2_CFG, MCASP3_CFG, MCASP4_CFG, MCASP5_CFG, MCASP6_CFG, MCASP7_CFG, MCASP8_CFG, GMAC_SW, PWMSS1, PWMSS2, PWMSS3, ATL, MLB, VCP1, VCP2, DCAN2.

WORKAROUND Avoid accessing other peripherals that are on the L4_PER2 segment of the interconnect while IODELAY configuration is occurring. This can be accomplished by performing all IODELAY configurations during boot time before other initiators are enabled. Alternatively, if run-time accesses to IODELAY are required then accesses to other peripherals on the L4_PER2 segment of the interconnect must be avoided while accessing IODELAY.

In order to support run-time SD-Card removal/detection on the MMC1 interface or other mode changes on MMCn interfaces, software should not modify IO Delay configuration when a new card is detected or speed is changed. However, limiting support of SD Card/MMC transfer modes to a common IODELAY configuration is an option. For example, the IODELAY configuration required for SDR50 is also compatible with identification, default-speed, high-speed, SDR12, and SDR25 transfer modes. Configuring IODELAY for SDR50 during boot without any further updates will avoid the hang condition and allows support all transfer modes up to SDR50. With this approach, the MMC1 interface cannot support DDR50 and SDR104 modes because IODELAY would need to be updated to support these transfer modes.

The final intended transfer mode may be known in advance when eMMC or other devices are attached to any MMCn interface. In that case, the appropriate IODELAY for the intended transfer mode may be configured at boot time (including HS200 mode if applicable).

Please Note: The standard Processor SDK software offering from TI (Linux and RTOS based) does not implement this workaround. Customers are expected to make the appropriate software modifications necessary to implement their own workaround when using this approach.

REVISIONS IMPACTED

SR 2.0, 1.1, 1.0

TDA2x: 2.0, 1.1, 1.0

DRA75x, DRA74x: 2.0, 1.1, 1.0

i936	*DSS LCD/DPI Out Field Reversal in Interlaced RGB Mode*
CRITICALITY	High
DESCRIPTION	Interlaced RGB is not supported on DSS DPI outputs (VOUT1/2/3) because the VSYNC signal's polarity is noncompliant. Some monitors that are fully compliant will exhibit display artifacts. However, there are some monitors which accept the noncompliant signal without artifacts.
WORKAROUND	Use Interlaced YUV mode with embedded sync (BT.656, BT.1120).
REVISIONS IMPACTED	SR 2.0, 1.1, 1.0 TDA2x: 2.0, 1.1, 1.0 DRA75x, DRA74x: 2.0, 1.1, 1.0 AM572x: 2.0, 1.1

i940 *MPU COUNTER_REALTIME saturates after several hundred days*

CRITICALITY Major

DESCRIPTION

After several hundred days (specific duration depends on the primary input clock SYS_CLK1 frequency, see table below for example durations) of continuous operation the MPU COUNTER_REALTIME saturates at 0xBB8000000000 and no longer generates interrupts. There is no way to reset the counter except for a system reset.

For software/systems that use this timer and operate for extended periods of time without any type of reset or power cycle, this can lead to a system hang. For systems that regularly go through a power cycle or reset there is no issue.

SYS_CLK1	Days until saturation
19.2 MHz	404
20 MHz	388
27 MHz	287

WORKAROUND

Workaround 1: Avoid the issue by requiring the user to preform a scheduled reset/restart of the system sometime prior the saturation time.

Workaround 2: Recover from the issue by enabling a system watchdog reset to expire after the issue has occurred.

Workaround 3: Use GP Timer instead of COUNTER_REALTIME in systems requiring continuous up time.

Chapter 3

Silicon Limitations

This Chapter describes limitations for the given architecture and provides information for working with those issues.

Revisions SR 2.0, 1.1, 1.0 - Limitations List

i596	***BITMAP1-2-4 Formats Not Supported by the Graphics Pipeline***
CRITICALITY	Low
DESCRIPTION	BITMAP1, BITMAP2, and BITMAP4 are not supported by the graphics pipeline.
WORKAROUND	No workaround is available.
REVISIONS IMPACTED	SR 2.0, 1.1, 1.0 TDA2x: 2.0, 1.1, 1.0 DRA75x, DRA74x: 2.0, 1.1, 1.0 AM572x: 2.0, 1.1

i641	*Overlay Optimization Limitations*
CRITICALITY	Low
DESCRIPTION	Overlay optimization does not work when resize processing is enabled on any 'Enabled' layer. When any of the 'Enabled' layers has bit field DISPC_p_ATTRIBUTES.RESIZEENABLE as nonzero it will neither be optimized nor participate in optimization of layers below.
WORKAROUND	For optimization to occur for a particular layer, make RESIZEENABLE as 0x0. With multiple layers enabled, make RESIZEENABLE for all the layers as 0x0 for every layer to participate in overlay optimization of itself or for the layers below it.
REVISIONS IMPACTED	SR 2.0, 1.1, 1.0 TDA2x: 2.0, 1.1, 1.0 DRA75x, DRA74x: 2.0, 1.1, 1.0 AM572x: 2.0, 1.1

i833	***I2C Module in Multislave Mode Potentially Acknowledges Wrong Address***
CRITICALITY	Low
DESCRIPTION	When the I2C module is in multislave mode with up to four 10-bit own addresses it may acknowledge the wrong address. According to the I2C protocol, a 10-bit address is sent via 2 bytes. The first byte is formatted as 11110XX R/W where XX are the 2 MSB of the address. The second byte contains the remaining 8 bits of the device address. When the first byte received contains 2 MSB that matches the 2 MSB of the one of the modules four own addresses, an ACK is correctly sent by the module. However, if the second byte received matches the remaining 8 bits of one of the modules other own addresses, an ACK is sent incorrectly by that module. In turn, the incorrect module then enters an internal state where starts reading data sent on the bus not intended for it. The module should only send a second ACK if the entire 10-bit address matches one of its four own addresses. However, the module incorrectly ACKs any address that matches the first 2 bits of one slave address and the last eight bits of another slave address.
WORKAROUND	The issue can be avoided by ensuring that the 2 most significant bits are identical for all multislave addresses in 10-bit addressing mode.
REVISIONS IMPACTED	SR 2.0, 1.1, 1.0 TDA2x: 2.0, 1.1, 1.0 DRA75x, DRA74x: 2.0, 1.1, 1.0 AM572x: 2.0, 1.1

i838	***DSS BT.656/BT.1120 Max Horizontal Blanking Is Non Compliant***
CRITICALITY	Medium
DESCRIPTION	When BT.656 or BT.1120 modes are used on DSS outputs (vout1,2,3), the configuration of the horizontal blanking timing is limited to a value of 256 bytes or less due to the HSW bit field being limited to 8b (256 max value). The BT.656 standard requires 280 or 268 bytes to support PAL and NTSC timings respectively. BT.1120 requires 280, 720, 830 depending on format. The DSS cannot support devices requiring such blanking duration due to this limitation.
WORKAROUND	If larger blanking period needed, use RGB or YUV mode output, not BT.656/1120.
REVISIONS IMPACTED	SR 2.0, 1.1, 1.0 TDA2x: 2.0, 1.1, 1.0 DRA75x, DRA74x: 2.0, 1.1, 1.0 AM572x: 2.0, 1.1

i844	***EDMA to VCP Stream Burst Is Not Functional***
CRITICALITY	Low
DESCRIPTION	If EDMA is configured for a constant addressing mode (also referred to as STREAM or FIFO mode) transfer to the VCP data space, the accesses fail.
WORKAROUND	Software must configure the EDMA to perform Incrementing transfers to VCP instead of constant addressing mode. A constant addressing mode transfer can be emulated by setting the EDMA PaRAM as follows: • Incrementing mode (via SAM or DAM = 0), • ACNT = 8 (to match VCP bus width), • BCNT = transfer_byte_count/8, • BIDX = 0 (via SBIDX or DBIDX), • ABSync trigger mode (via SYNCDIM = 1).
REVISIONS IMPACTED	SR 2.0, 1.1, 1.0 TDA2x: 2.0, 1.1, 1.0 DRA75x, DRA74x: 2.0, 1.1, 1.0

i845 ***USB2.0 False Detection of Disconnect Condition***

CRITICALITY Low

DESCRIPTION Per the standard, the 'Envelope detector' in the USB2.0 PHY must indicate High Speed disconnection when the amplitude of the differential signal at the downstream facing driver's connector is greater than or equal to 625 mV, and it must not be indicated when the signal amplitude is less than or equal to 525 mV at our connector. The default configuration of the USB1/USB2 phy is such that the detector circuit is monitoring the during the entire SOF frame and not just the last 8b portion of the extended EOP as indicated in the standard.

Due to this constant monitoring, it is possible that the PHY may falsely indicate disconnect condition due to reflections on the PCB. These reflections could exceed the disconnect thresholds if TI PCB layout guidelines for USB are not strictly followed.

WORKAROUND SW Workaround: During USB2.0 PHY initialization, configure: bit 31 of 0x4A08 404C (USB1) or 0x4A08 504C (USB2) to =1b. This will limit detection window to the proper time and prevent false disconnect. TI PCB layout guidelines for USB should also be strictly followed to minimize reflections to within the thresholds defined in the standard. See device Data Manual for details.

REVISIONS IMPACTED SR 2.0, 1.1, 1.0

TDA2x: 2.0, 1.1, 1.0

DRA75x, DRA74x: 2.0, 1.1, 1.0

AM572x: 2.0, 1.1

i848	***McASP IO Pad Loopback Not Functional***
CRITICALITY	Low
DESCRIPTION	Due to timing issues, the IO pad loopback feature of the McASP, configured by setting DLBEN=1 and IOLBEN=1 in the MCASP_LBCTL register, will sometimes result in corrupt serializer data. This feature is intended for development only. All McASP instances are potentially affected.
WORKAROUND	Use digital loopback, which is logically equivalent. Digital loopback is configured by setting DLBEN=1 and IOLBEN=0b in the appropriate MCASP_LBCTL register.
REVISIONS IMPACTED	SR 2.0, 1.1, 1.0 TDA2x: 2.0, 1.1, 1.0 DRA75x, DRA74x: 2.0, 1.1, 1.0 AM572x: 2.0, 1.1

i850	***ESD Fail on MPU PLL Power***
CRITICALITY	High
DESCRIPTION	Possible ESD failure on MPU PLL (VDDA_MPU) if stressed close to spec limits.
WORKAROUND	None.
REVISIONS IMPACTED	SR 1.0 TDA2x: 1.0 DRA75x, DRA74x: 1.0 AM572x: 1.0

i851	***QSPI Redundant SBL Feature Sector Size Mismatch With Flash***
CRITICALITY	Medium
DESCRIPTION	The redundant boot image in QSPI boot is too small to span multiple sectors in any standard QSPI flash. The 512byte offset between SBL images means that all 4 redundant SBL are inside a single erasable sector of the flash. ES1.1 or later devices eliminate this limitation by providing additional sysboot configuration options to select between 64, 128, 256 or 512KB offsets between SBL images (sysboot7:6 = 00, 01, 10 or 11 respectively).
WORKAROUND	None. Primary image must be used in flash.
REVISIONS IMPACTED	SR 1.0 TDA2x: 1.0 DRA75x, DRA74x: 1.0 AM572x: 1.0

i853	***MMC2,3,4 Do Not Support 3.3V/1.8V Dynamic Switch***
CRITICALITY	Medium
DESCRIPTION	The dynamic voltage switch 3.3V/1.8V, defined in SD standard, is not supported by MMC2,3 and 4. The corresponding IO voltage rails for these interfaces must remain at a fixed voltage from power-on-reset (either 1.8V or fixed 3.3V).
WORKAROUND	None. Use these interfaces at fixed voltage only.
REVISIONS IMPACTED	SR 1.0 TDA2x: 1.0 DRA75x, DRA74x: 1.0 AM572x: 1.0

i857	*Optional VOUT3 Clock Muxing Not Meeting IO Timing*
CRITICALITY	Low
DESCRIPTION	The optional vout3_clk function on the vin1a_fld0 pin (muxmode = 0x4) does not meet Device Data Manual timings for vout3 interface. Violation could be on order of approximately 2.5ns.
WORKAROUND	Use a software patch, contact your TI representative, to configure registers within Vayu which can modify the vout3_clk IO timing sufficiently to meet the Device Data Manual timings.
REVISIONS IMPACTED	SR 1.0 TDA2x: 1.0 DRA75x, DRA74x: 1.0 AM572x: 1.0

i858	***DELAYMODE Mechanism Not Selecting Proper Delay for Some IP Modes***
CRITICALITY	High
DESCRIPTION	IO timings are not met for some IPs and IP modes, and specific configuration of the DELAYMODE and SLEW configurations in the pad control registers (control module) are required for them. • McASP1,2 asynchronous transmit • GPMC synchronous mode 5 load timings • QSPI timing mode1 and timing mode2*
WORKAROUND	If these modes are used, the SoC should only be used under nominal conditions (for example, room temp/lab/software development), not for production testing.
REVISIONS IMPACTED	SR 1.0 TDA2x: 1.0 DRA75x, DRA74x: 1.0 AM572x: 1.0

i876	***DVFS Only Supported on MPU***
CRITICALITY	Low
DESCRIPTION	Dynamic Voltage Frequency Scaling (DVFS) refers to a software technique where the various SoC AVS rails are changed from one OPP level to another in order to either adapt to a changing work-load, or in order to avoid device operation outside of desired temperature bounds. The SoC only supports DVFS on the MPU rail.
WORKAROUND	Software may use DVFS only on the MPU rail. For example, SW may change the MPU operation from OPP_NOM (1 GHz, at OPP_NOM AVS voltage level) to OPP_HIGH (1.5 GHz, at OPP_HIGH AVS voltage level) in order to accommodate an increased work-load, or may change the MPU operation from OPP_HIGH to OPP_NOM if the temperature exceeds a certain threshold. The supply rails and OPP conditions (Frequency/Voltage) for non-MPU rails should be set at boot-time and remain at that OPP voltage level during device run-time.
REVISIONS IMPACTED	SR 2.0, 1.1, 1.0 TDA2x: 2.0, 1.1, 1.0 DRA75x, DRA74x: 2.0, 1.1, 1.0 AM572x: 2.0, 1.1

i877 ***RGMII Clocks Should Be Enabled at Boot Time***

CRITICALITY Medium

DESCRIPTION The RGMII 1000 Mbps Transmit timing is based on the output clock (rgmiin_txc) being driven relative to the rising edge of an internal clock and the output control/data (rgmiin_txctl/txd) being driven relative to the falling edge of an internal clock source. If the internal clock source is allowed to be static low (i.e., disabled) for an extended period of time then when the clock is actually enabled the timing delta between the rising edge and falling edge can change over the lifetime of the device. This can result in the device switching characteristics degrading over time, and eventually failing to meet the Data Manual Delay Time/Skew specs.

To maintain RGMII 1000 Mbps IO Timings, SW should minimize the duration that the Ethernet internal clock source is disabled. Note that the device reset state for the Ethernet clock is "disabled".

Other RGMII modes (10 Mbps, 100Mbps) are not affected.

WORKAROUND If the SoC Ethernet interface(s) are used in RGMII mode at 1000 Mbps, SW should minimize the time the Ethernet internal clock source is disabled to a maximum of 200 hours in a device life cycle. This is done by enabling the clock as early as possible in Secondary Boot Loader (SBL) by setting the register CM_GMAC_CLKSTCTRL[1:0] CLKTRCTRL = 0x2:SW_WKUP.

In addition to programming SW_WKUP(0x2) on CM_GMAC_CLKSTCTRL, SW should also program modulemode field as ENABLED(0x2) on CM_GMAC_GMAC_CLKCTRL register.

If the application does not require Ethernet functionality ever, the developer can choose to place the GMAC module in a power disabled state CM_GMAC_GMAC_CLKCTRL.MODULEMODE = 0x0 (disabled) and CM_GMAC_CLKSTCTRL.CLKTRCTRL = 0x1 (SW_SLEEP) during the boot operation.

REVISIONS IMPACTED SR 2.0, 1.1, 1.0

TDA2x: 2.0, 1.1, 1.0

DRA75x, DRA74x: 2.0, 1.1, 1.0

AM572x: 2.0, 1.1

i892	*L3 Clocks Should Be Enabled at All Times*
CRITICALITY	Medium
DESCRIPTION	If the L3 clock (L3_ICLK) is allowed to be gated by the PRCM, the device internal timing can degrade leading to functional issues and device failures.
WORKAROUND	The L3 clock (L3_ICLK) should be enabled at all times.
REVISIONS IMPACTED	SR 2.0, 1.1, 1.0 TDA2x: 2.0, 1.1, 1.0 DRA75x, DRA74x: 2.0, 1.1, 1.0 AM572x: 2.0, 1.1

i909	***PCIe Unintentional Translation of Outbound Message TLPs***
CRITICALITY	Medium
DESCRIPTION	There is a limitation in internal address translation unit in which unintentional translation of outbound message TLPs can occur if the third and fourth double words of the header match an iATU region. The unintentional translation is most likely to occur in the case of an address translation region at location 0x0 in address space since many message TLPs require the third and fourth double words of the header to be 0x0. Outbound completion TLPs also partially suffer from the same issue. Completion TLPs are never translated by the controller. However if the client address is held at a value which matches an outbound iATU region when a completion TLP is being transmitted, the controller will reduce the credit counter of the type specified in that region. Because the client address bus is normally held at address 0x0 when transmitting completion TLPs, this issue generally occurs when an address translation region is defined at address 0x0.
WORKAROUND	Do not configure an outbound iATU region starting at address 0x0. Instead, only configure outbound iATU regions starting at address 0x1000 or greater. At the SoC level, this workaround effectively reduces each PCIE_SSx 256MiB L3_MAIN address window to 256MiB – 4KB, starting at address 0x2000_1000 for PCIE_SS1 and address 0x3000_1000 for PCIE_SS2.
REVISIONS IMPACTED	SR 2.0, 1.1, 1.0 TDA2x: 2.0, 1.1, 1.0 DRA75x, DRA74x: 2.0, 1.1, 1.0 AM572x: 2.0, 1.1

This page intentionally left blank.

Chapter 4

Silicon Cautions

This Chapter alerts device users to silicon sensitivity concerns. Items described in the following section are compliant with specification (neither bug nor limitation), but it is mandatory to carefully respect guidelines to ensure correct device behavior.

Revisions SR 2.0, 1.1, 1.0 - Cautions List

i781	***Power Delivery Network Verification***
CRITICALITY	High
DESCRIPTION	Processor operation requires strict power requirements on the system (Processor + Power Management IC + Power Distribution Network). The Processor requires carefully controlled system margin validation and verification. In GHz systems, instability could result from marginal board design, component selection, power supply transients, susceptibility to noise, and so forth. Developers must optimize PDN board designs to ensure stable operation at all OPP's across all conditions and over the lifetime of the system. The necessary steps to follow to ensure robust operation are listed in the following Guidelines section.
GUIDELINES	• Software guidelines: 1. It is mandatory to use SmartReflex technology for the AVS power rails (MPU, CORE, DSPEVE, GPU, etc.). Refer to the Data Manual for AVS requirements. 2. For certain power rails, ABB (adaptative body bias) must be engaged in: – Refer to the device Data Manual for ABB requirements for each OPP • PCB guidelines: The Power Delivery Network should be optimized to match all OPP requirements. All PCB Design requirements for PDN optimization can be found in the Data Manual. It is mandatory for the PCB developer to align the PCB with the described guidelines and to meet TI requirements.
REVISIONS IMPACTED	SR 2.0, 1.1, 1.0 TDA2x: 2.0, 1.1, 1.0 DRA75x, DRA74x: 2.0, 1.1, 1.0

4.1

i827	***Thermal Alert Will Not Be Generated When Bandgap Is Configured in "Smart Idle" Mode***
CRITICALITY	Low
DESCRIPTION	NOTE: The previous title for this advisory was "MPU Running While Thermal Monitoring Is Stopped." The MPU will not receive an interrupt request for thermal alert (MPU_IRQ_126) if the MPU bandgap cell is in idle mode. When the MPU bandgap cell is in idle state (CTRL_CORE_BANDGAP_MASK_1[31:30] SIDLEMODE=0x2), the MPU bandgap state machine can not send THERMAL_ALERT signal and MPU interrupt request, respectively.
GUIDELINES	Do not idle the Control Module MPU Bandgap (keep CTRL_CORE_BANDGAP_MASK_1[31:30] SIDLEMODE=0x0)
REVISIONS IMPACTED	SR 2.0, 1.1, 1.0 TDA2x: 2.0, 1.1, 1.0 DRA75x, DRA74x: 2.0, 1.1, 1.0 AM572x: 2.0, 1.1

i832	*DLL SW Reset Bit Does Not Reset to 0 after Execution*
CRITICALITY	Medium
DESCRIPTION	When autoidle is enabled (MMCHS_SYSCONFIG[0] AUTOIDLE=0x1), clock gets cut off and the reset completion signal would not be recorded by the processor. Hence, though the reset executed and finished, the MMCHS_DLL[31] DLL_SOFT_RESET flag will remain asserted indefinitely and another soft reset will be ignored.
GUIDELINES	Disable autoidle (MMCHS_SYSCONFIG[0] AUTOIDLE=0x0), before DLL reset and re-enable autoidle after the reset. Set MMCHS_SYSCONFIG[0] AUTOIDLE = 0 before reset Set MMCHS_SYSCONFIG[0] AUTOIDLE = 1 after the reset
REVISIONS IMPACTED	SR 2.0, 1.1, 1.0 TDA2x: 2.0, 1.1, 1.0 DRA75x, DRA74x: 2.0, 1.1, 1.0 AM572x: 2.0, 1.1

i836	***Bus Testing Commands CMD19 Incorrectly Waits for CRC Status Return***
CRITICALITY	Low
DESCRIPTION	CMD19/CMD14 commands are required for MMC to test data bus pins functionality. After the MMC controller sends out CMD19, it incorrectly waits for the CRC status to be returned. Because no CRC status is generated by the card at this step, the MMC controller signals an interrupt for Data Timeout (DTO).
GUIDELINES	Ignore DTO generated after CMD19 is sent, then clear the interrupt and proceed with CMD14.
REVISIONS IMPACTED	SR 2.0, 1.1, 1.0 TDA2x: 2.0, 1.1, 1.0 DRA75x, DRA74x: 2.0, 1.1, 1.0 AM572x: 2.0, 1.1

i839	***Some RGB and YUV Formats Have Non-Standard Ordering***
CRITICALITY	Medium
DESCRIPTION	Data format definitions are non-standard and resulting in color components being swapped if wrong assumption is made in software. To configure data packing/unpacking logic for each active data channel, the VPDMA (in the VIP and VPE) relies on the 'data type' value in the channel's data transfer descriptor. Due to mismatches in the component order directions between what the VPDMA specifies and what commonly used image identifiers expect, color components can be swapped in the display and in the video/image data written out to the memory for some RGB and YUV data types.
GUIDELINES	Software drivers should remap custom formats to the desired industry standard formats and/or treat data as word/byte swapped as appropriate. See also Device TRM for full explanation of data formats.
REVISIONS IMPACTED	SR 2.0, 1.1, 1.0 TDA2x: 2.0, 1.1, 1.0 DRA75x, DRA74x: 2.0, 1.1, 1.0 AM572x: 2.0, 1.1

i864 ***VDDS18V to VDDSHVn Current Path***

CRITICALITY Medium

DESCRIPTION A current path exists between vdds18v and vddshvn during power down sequence.

The Data Manual allows the vddshvn rail (in either 3.3V or 1.8V mode) to power down at the same time or before the vdds18v rail. When the vddshvn rail is powered down before the vdds18v rail, the vddshvn rail stays at Plateau Level (<1.5V) until the vdds18v rail is disabled, as shown in the waveform below.

A subset of the SOC's LVCMOS IOs (refer to DM for details) have a pull-up resistor that is active by default, including during reset and power-down. The SOC's IOs that have internal or external pull-ups will stay at Plateau Level (<1.5V) during the power-down. If other ICs on the board that are interfaced to the SOC's LVCMOS signals use a separate supply then it's possible that the other IC's signals can be pulled to the Plateau Level while its power supply is off.

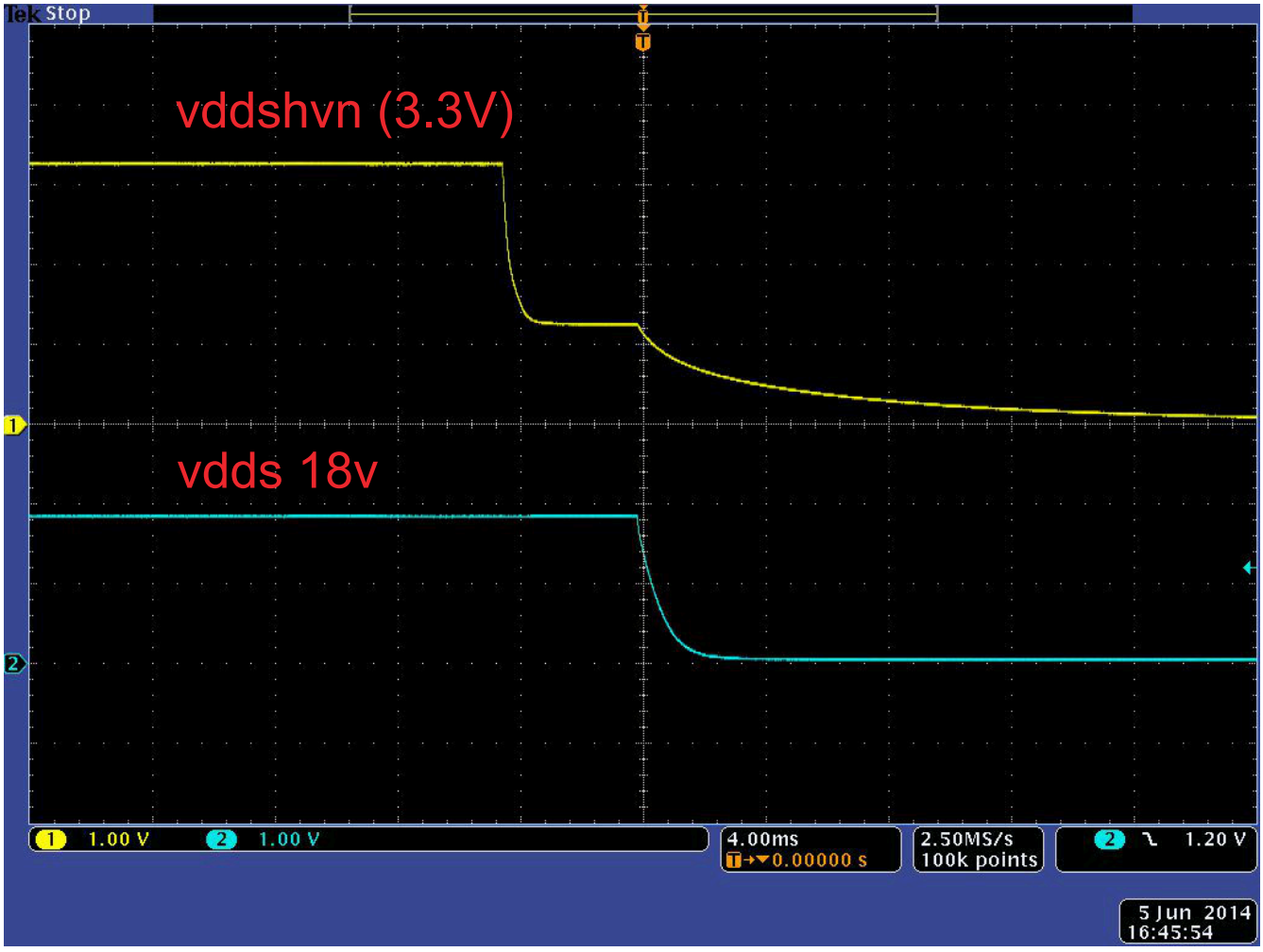

Figure 4-1. PAB_RevA

The root cause of the plateau is related to the LVCMOS IO buffer Dual Voltage detection circuitry. The LVCMOS Dual Voltage IO includes voltage comparator circuitry to determine if the IO is in 3.3V mode or 1.8V mode. During powerdown of vddshvn domains, a current path in the internal bias transistors results in the vddshvn rail being held to an

intermediate voltage level (<1.5 V). This path can consume at most 500 uA per IO - worst case estimate is ~150 mA (based on 280 IOs) from the vdds18v supply during power down. This path is not a reliability concern for the device.

The plateau is no concern for systems where the same supply/LDO is used for vddshvn rail and the other components that interface to the SOC's Dual Voltage LVCMOS IOs.

Systems that use independent supplies for the SOC rail and the other component's rail require further analysis by the system designer. There may be a state where SOC's IO's with internal or external pull-ups are pulled to plateau level (<1.5V) while the external device is powered down. In this case, the current on any given IO is limited due to the ~10 kOhm (minimum) internal pull-up resistor. The limit is 150 µA per IO (1.5 V maximum plateau / 10 kOhm minimum pull-up resistor.) Refer to the device Data Manual for details on which pins include a pull-up resistor by default.

GUIDELINES

In general, TI recommends using the same supply source for connected components. E.g., a single LDO should drive vddshvn and the related 3.3V external components.

For systems that use a different 3.3V supply for the SOC and connected ICs, customers should evaluate their system for reliability risk. If necessary, the PMIC OTP power-down sequence can be modified to delay the vddshv[11:1] powerdown to coincide with the vdds18v powerdown. [Note: The 3.3V rail must never be 2.0V above the 1.8V rail.]

VDDSHV8 is a special case. If VDDSHV8 is powered by the same LDO/switch as the other VDDSHVn rails then the VDDSHV8 rail can also be delayed. However, if the VDDSHV8 rail is supplied by a different LDO (e.g., LDO1 on EVM) than the other VDDSHVn rails, then the sequence should not be modified.

REVISIONS IMPACTED

SR 2.0, 1.1, 1.0

TDA2x: 2.0, 1.1, 1.0

DRA75x, DRA74x: 2.0, 1.1, 1.0

AM572x: 2.0, 1.1

i885 ***Software Requirements for Data Manual IO Timing***

CRITICALITY High

DESCRIPTION In order to guarantee the values presented in the Timing Requirements and Switching Characteristics tables of the device Data Manual, proper software configuration of the IOs is required.

GUIDELINES The required software IO configurations include:

- Pin Multiplexing (CTRL_CORE_PAD_*[3:0] MUXMODE) selection compliant with the IOSETS defined in the datasheet. (See IOSET tables DM Chapter Timing Requirements and Switching Characteristics.)
- Slew Control (CTRL_CORE_PAD_*[19] SLEWCONTROL) settings left at their default values. (See DM Section 1.8V and 3.3V Signal Transition Rates.)
- Virtual and Manual IO Timing Modes configured as required for the desired mode of operation. (See TRM Sections Virtual IO Timing Modes and Manual IO Timing Modes.)
- IO Delay Recalibration performed after adjusting the AVS voltage for VDD_CORE_L voltage domain. (See TRM Section IO Delay Recalibration.)

REVISIONS IMPACTED SR 2.0, 1.1, 1.0

TDA2x: 2.0, 1.1, 1.0

DRA75x, DRA74x: 2.0, 1.1, 1.0

AM572x: 2.0, 1.1

i886	***FPDLink PLL Unlocks With Certain SoC PLL M/N Values***
CRITICALITY	Medium
DESCRIPTION	FPD-Link SerDes are used to convert the Device's parallel video output interfaces into high-speed serialized interfaces. To ensure proper operation, it is important for the Devices's video output clock to meet the input jitter requirements of the SerDes component clock input. At high frequencies, some Device PLL configurations, may produce a clock signal that does not comply with the FPD-Link specifications. These PLL configurations can potentially cause the FPD-Link deserializer to loose lock, producing flicker or blanking on the system display.
GUIDELINES	See application note SPRACA9 for information on how to best work around this issue in a given system.
REVISIONS IMPACTED	SR 2.0, 1.1, 1.0 TDA2x: 2.0, 1.1, 1.0 DRA75x, DRA74x: 2.0, 1.1, 1.0 AM572x: 2.0, 1.1

i912	***QSPI_SPI_CMD_REG [25:24] Masked from Read in RTL***
CRITICALITY	Low
DESCRIPTION	There is an integration error in the device. All WLEN (QSPI_SPI_CMD_REG[25:19]) bits in the QSPI_SPI_CMD_REG register are writeable. However, on a read the QSPI_SPI_CMD_REG[25:24] bits will be masked.
GUIDELINES	None.
REVISIONS IMPACTED	SR 2.0, 1.1, 1.0 TDA2x: 2.0, 1.1, 1.0 DRA75x, DRA74x: 2.0, 1.1, 1.0 AM572x: 2.0, 1.1

i926	***PCIe Preferred PCIe_PHY_RX SCP Register Settings Updated***
CRITICALITY	Low
DESCRIPTION	The “Preferred PCIe_PHY_RX SCP Register Settings” table in version AE and later of the device-specific Technical Reference Manual (TRM) has been updated with values to select fully adaptive equalization and a second-order clock recovery algorithm. These changes have been shown to enhance PCIe receiver (RX) jitter tolerance for 5GT/s operation.
GUIDELINES	Software can be updated to the new preferred settings, especially for 5GT/s operation, if enhanced RX jitter tolerance is desired.
REVISIONS IMPACTED	SR 2.0, 1.1, 1.0 TDA2x: 2.0, 1.1, 1.0 DRA75x, DRA74x: 2.0, 1.1, 1.0 AM572x: 2.0, 1.1

i931 ***VDD to VDDA_"PHY" Current Path***

CRITICALITY Low

DESCRIPTION A current path exists between VDD and the high speed analog PHY domain (VDDA_HDMI, VDDA_PCIE, VDDA _SATA, VDDA_USB2/3) during the supply power up and power down sequences.

The device-specific Data Manual requires Core AVS rail (VDD) to power up before the 1.8 V high speed analog PHY domain(s). When this sequence is followed, the high speed analog PHY domain will have a small step-up to a voltage plateau (< 0.5 V) that aligns to the beginning of the VDD ramp-up, and is maintained until the ramp-up of high speed analog PHY rail, as shown in Figure 4-2. Note the leakage value will differ from system to system, but will be less than 500 mV. The leakage voltage in the provided capture is approximately 250 mV.

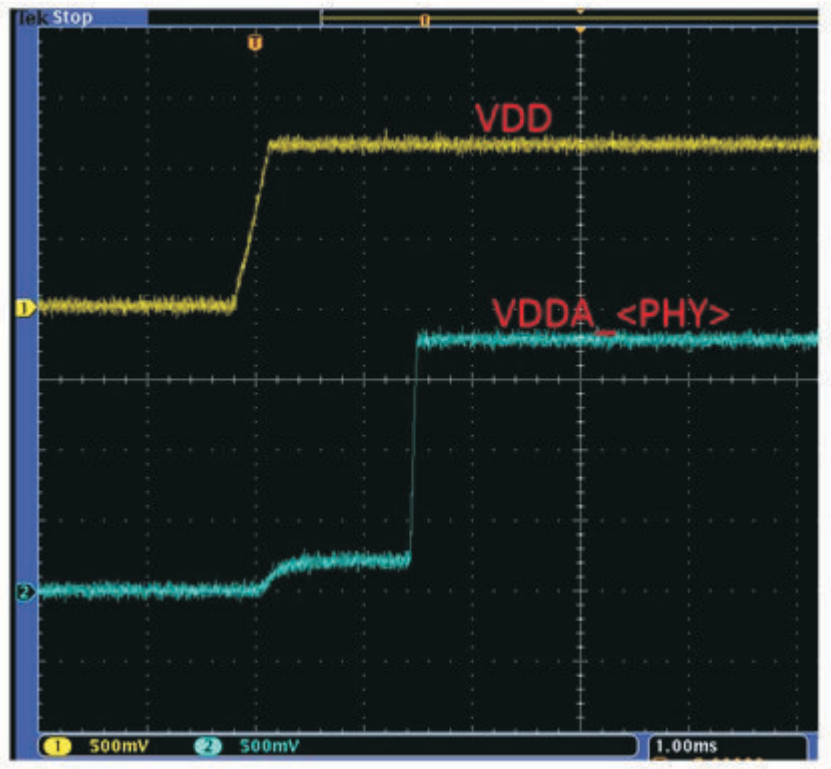

Figure 4-2. Leakage Voltage

A similar condition exists during supply power down sequence. The high speed analog PHY domain(s) are required to be powered down prior to Core AVS rail (VDD), and may cause a small plateau to exist until Core AVS rail (VDD) is ramped down.

The root cause of the voltage plateau during power-up/down sequencing is related to the parasitic diodes in logic blocks which have multiple power sources. This leakage path is not a reliability concern for this device.

GUIDELINES None. There is no reliability concern for the device.

REVISIONS IMPACTED SR 2.0, 1.1, 1.0

TDA2x: 2.0, 1.1, 1.0

DRA75x, DRA74x: 2.0, 1.1, 1.0

AM572x: 2.0, 1.1

i935	***MSI Bit in PCIECTRL_TI_CONF_IRQSTATUS_MSI Register Does Not Clear Automatically***
CRITICALITY	Low
DESCRIPTION	The MSI bit in PCIECTRL_TI_CONF_IRQSTATUS_MSI register does not clear automatically even after all the vectors in PCIECTRL_PL_MSI_CTRL_INT_STATUS_N registers are cleared.
GUIDELINES	Software should manually clear PCIECTRL_TI_CONF_IRQSTATUS_MSI[4] MSI bit after making sure there are no vectors set in PCIECTRL_PL_MSI_CTRL_INT_STATUS_N registers. If MSI bit is cleared with some of the bits of PCIECTRL_PL_MSI_CTRL_INT_STATUS_N still set then those interrupts may be lost which may lead to non-functional remote endpoints. Following is the recommended sequence for handling MSI interrupt to avoid missing any MSI interrupts: 1. Software reads PCIECTRL_TI_CONF_IRQSTATUS_MSI status register, and identifies an (unspecified) MSI event, as opposed to a PCI legacy event. 2. Software clears PCIECTRL_TI_CONF_IRQSTATUS_MSI[4] MSI. 3. Software reads PCIECTRL_PL_MSI_CTRL_INT_STATUS_N and identifies EP functions that raised MSI. 4. Software clears PCIECTRL_PL_MSI_CTRL_INT_STATUS_N status bit. 5. Software calls the EP function specific interrupt handlers for each of the vectors. 6. Repeat steps 3 to 5 until PCIECTRL_PL_MSI_CTRL_INT_STATUS_N reads 0. 7. Exit MSI IRQ handler, only if PCIECTRL_PL_MSI_CTRL_INT_STATUS_N is 0
REVISIONS IMPACTED	SR 2.0, 1.1, 1.0 TDA2x: 2.0, 1.1, 1.0 DRA75x, DRA74x: 2.0, 1.1, 1.0 AM572x: 2.0, 1.1

Revision History

Changes from February 1, 2018 to February 28, 2021 (from Revision I (February 2018) to Revision J (February 2021)) **Page**

- Added i936: DSS LCD/DPI Out Field Reversal in Interlaced RGB Mode....91
- Added i940: MPU COUNTER_REALTIME saturates after several hundred days....92
- Added i935: MSI Bit in PCIECTRL_TI_CONF_IRQSTATUS_MSI Register Does Not Clear Automatically... 125

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), 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, regulatory 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 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.

TI objects to and rejects any additional or different terms you may have proposed.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265