DRA78x SoC for Automotive Infotainment Silicon Revision 2.0A, 2.0

Errata

Literature Number: SPRZ444C
FEBRUARY 2018 – REVISED NOVEMBER 2026

Table of Contents

1 Introduction.....5
2 Silicon Advisories.....9
3 Silicon Limitations.....37
4 Silicon Cautions.....43
5 Revision History.....53

Copyright © 2026 Texas Instruments Incorporated

This page intentionally left blank.

Copyright © 2026 Texas Instruments Incorporated

Chapter 1
Introduction

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

Related Documentation

DRA78x SoC for Automotive Infotainment Silicon Revision 2.0A, 2.0 Technical Reference Manual (SPRUIF4)

DRA78x Infotainment Applications Processor Data Manual (SPRS975)

DRA78x Pad Configuration Tool

DRA78x Clock Tree Tool (CLOCKTREETOOL-AUTOMOTIVE)

Copyright © 2026 Texas Instruments Incorporated

Trademarks

SmartReflex is a trademark of Texas Instruments.

MMC and eMMC are trademarks of MultiMediaCard Association.

JTAG is a registered trademark of JTAG Technologies B.V.

All other trademarks are the property of their respective owners.

Copyright © 2026 Texas Instruments Incorporated

Modules Impacted

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

MODULE	DESCRIPTION	SILICON REVISIONS AFFECTED	
		DRA78x	
		2.0	2.0A
NA	i781: Power Delivery Network Verification	Yes	Yes
	i864: VDDS18V to VDDSHVn Current Path	Yes	Yes
Control Module	i813: Spurious Thermal Alert Generation When Temperature Remains in Expected Range	Yes	Yes
	i814: Bandgap Temperature Read Dtemp Can Be Corrupted	Yes	Yes
	i827: Thermal Alert Will Not Be Generated When Bandgap Is Configured in "Smart Idle" Mode	Yes	Yes
	i828: IO Input Glitches May Occur When Switching Pull Type and Mux Mode Simultaneously	Yes	Yes
DCAN	i893: DCAN Initialization Sequence	Yes	Yes
DEBUG	i879: DSP MStandby Requires CD_EMU in SW_WKUP	Yes	Yes
DSP	i872: DSP MFlag Output Not Initialized	Yes	Yes
	i879: DSP MStandby Requires CD_EMU in SW_WKUP	Yes	Yes
	i883: DSP Doesn't Wake From Subsystem Internal Interrupts	Yes	Yes
	i898: DSP Pre-fetch Should Be Disabled before Entering Power Down Mode	Yes	Yes
DSS	i829: Reusing Pipe Connected to Writeback Pipeline On-the-Fly to an Active Panel	Yes	Yes
	i839: Some RGB and YUV Formats Have Non-Standard Ordering	Yes	Yes
	i873: DSS: First Two Columns of Active Video Are Always Black at the Output of Video Encoder	Yes	Yes
	i894: DSS VENC Early Termination on Last Line of Frame	Yes	Yes
	i914: Limitations with DISPC Write-Back Region-Based Mechanism	Yes	Yes
	i936: DSS LCD/DPI Out Field Reversal in Interlaced RGB Mode	Yes	Yes
	i938: SD-DACxx: SD-DAC CVIDEO_TVOUT Can Power up to an Inoperable State	Yes	Yes
EMIF	i727: Refresh Rate Issue after Warm Reset	Yes	Yes
	i729: DDR Access Hang after Warm Reset	Yes	Yes
GMAC_SW	i877: RGMII Clocks Should Be Enabled at Boot Time	Yes	Yes
	i899: Ethernet DLR Is Not Supported	Yes	Yes
I2C	i694: System I2C Hang Due to Miss of Bus Clear Support	Yes	Yes
	i833: I2C Module in Multislave Mode Potentially Acknowledges Wrong Address	Yes	Yes
	i930: I2C1 and I2C2 May Drive Low During Reset	Yes	Yes
INTC	i883: DSP Doesn't Wake From Subsystem Internal Interrupts	Yes	Yes
MCAN	i939: MCAN: Message Transmitted with Wrong Arbitration and Control Fields (Early Start of Frame)	Yes	Yes
MPU	i940: MPU COUNTER_REALTIME saturates after several hundred days	Yes	Yes
PRCM	i826: HSDIVIDER1 CLKOUT4 Could Glitch During On-the-Fly Divider Change to/from Divide-by-2.5	Yes	Yes
	i876: DVFS Not Supported	Yes	Yes
	i886: FPDLink PLL Unlocks with Certain SoC PLL M/N Values	Yes	Yes
QSPI	i912: QSPI_SPI_CMD_REG [25:24] Masked from Read in RTL	Yes	Yes
	i916: QSPI: Relying on Pull-ups for 'Default Values' Fails at Higher Clock Rates, Causing EDMA Read Failures	Yes	Yes
SDIO	i836: Bus Testing Commands CMD19 Incorrectly Waits for CRC Status Return	Yes	Yes
TIMERS	i767: Delay Needed to Read Some Timer Registers after Wakeup	Yes	Yes
	i874: TIMER5/6/7/8 Interrupts Not Propagated	Yes	Yes

Copyright © 2026 Texas Instruments Incorporated

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

MODULE	DESCRIPTION	SILICON REVISIONS AFFECTED	
		DRA78x	
		2.0	2.0A
UART	i202: MDR1 Access Can Freeze UART Module	Yes	Yes
	i889: UART Does Not Acknowledge Idle Request After DMA Has Been Enabled	Yes	Yes
VIP	i839: Some RGB and YUV Formats Have Non-Standard Ordering	Yes	Yes
VPE	i839: Some RGB and YUV Formats Have Non-Standard Ordering	Yes	Yes

Copyright © 2026 Texas Instruments Incorporated

Chapter 2

Silicon Advisories

Copyright © 2026 Texas Instruments Incorporated

i202 *MDR1 Access Can Freeze UART Module*

CRITICALITY

Medium

DESCRIPTION

Because of a glitchy structure inside the UART module, accessing the UART_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 UART_MDR1 must be changed:

1. If needed, setup the UART by writing the required registers, except UART_MDR1
2. Set appropriately the UART_MDR1[2:0] MODE_SELECT bit field
3. Wait for 5 L4 clock cycles + 5 UART functional clock cycles
4. Clear TX and RX FIFO in UART_FCR register to reset its counter logic
5. Read UART_RESUME register to resume the halted operation

Step 5 is for IrDA mode only and can be omitted in UART mode.

REVISIONS IMPACTED

SR 2.0A, 2.0

TDA3x: 2.0A, 2.0, 1.0A, 1.0

DRA78x: 2.0A, 2.0

DM50x: 2.0

Copyright © 2026 Texas Instruments Incorporated

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 2.0A, 2.0

TDA3x: 2.0A, 2.0, 1.0A, 1.0

DRA78x: 2.0A, 2.0

DM50x: 2.0

Copyright © 2026 Texas Instruments Incorporated

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 workarounds:

1. Avoid use of warm reset. This can be done by converting warm reset to PORz by using external circuitry to monitor the rstoutn signal (which asserts on warm or cold reset) and drive PORz when rstoutn is asserted. 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 2.0A, 2.0

TDA3x: 2.0A, 2.0, 1.0A, 1.0

DRA78x: 2.0A, 2.0

DM50x: 2.0

Copyright © 2026 Texas Instruments Incorporated

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:

- 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.
- 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 workarounds:

1. Avoid use of warm reset. This can be done by converting warm reset to PORz by using external circuitry to monitor the rstoutn signal (which asserts on warm or cold reset) and drive PORz when rstoutn is asserted. 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 2.0A, 2.0

TDA3x: 2.0A, 2.0, 1.0A, 1.0

DRA78x: 2.0A, 2.0

DM50x: 2.0

Copyright © 2026 Texas Instruments Incorporated

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

CRITICALITY

Medium

DESCRIPTION

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:

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.

WORKAROUND

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'.

REVISIONS IMPACTED

SR 2.0A, 2.0

TDA3x: 2.0A, 2.0, 1.0A, 1.0

DRA78x: 2.0A, 2.0

DM50x: 2.0

Copyright © 2026 Texas Instruments Incorporated

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_CORE.

WORKAROUND

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

1. Disable the alert interrupt.
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 2.0A, 2.0

TDA3x: 2.0A, 2.0, 1.0A, 1.0

DRA78x: 2.0A, 2.0

DM50x: 2.0

Copyright © 2026 Texas Instruments Incorporated

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_CORE_n (n = 0 to 4)
- CTRL_CORE_TEMP_SENSOR_CORE

WORKAROUND

Multiple reads to CTRL_CORE_TEMP_SENSOR_CORE[9:0]:

BGAP_DTEMP_CORE is needed to discard false value and read right value:

1. Perform two successive reads to BGAP_DTEMP_CORE 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 2.0A, 2.0

TDA3x: 2.0A, 2.0, 1.0A, 1.0

DRA78x: 2.0A, 2.0

DM50x: 2.0

Copyright © 2026 Texas Instruments Incorporated

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.

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 2.0A, 2.0

TDA3x: 2.0A, 2.0, 1.0A, 1.0

DRA78x: 2.0A, 2.0

DM50x: 2.0

Copyright © 2026 Texas Instruments Incorporated

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_VID_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_VID_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_VID_ATTRIBUTES[0] ENABLE = 0x1) at the end.

REVISIONS IMPACTED

SR 2.0A, 2.0

TDA3x: 2.0A, 2.0, 1.0A, 1.0

DRA78x: 2.0A, 2.0

DM50x: 2.0

Copyright © 2026 Texas Instruments Incorporated

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 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 2.0A, 2.0

TDA3x: 2.0A, 2.0, 1.0A, 1.0

DRA78x: 2.0A, 2.0

DM50x: 2.0

Copyright © 2026 Texas Instruments Incorporated

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 2.0A, 2.0

TDA3x: 2.0A, 2.0, 1.0A, 1.0

DRA78x: 2.0A, 2.0

DM50x: 2.0

Copyright © 2026 Texas Instruments Incorporated

i879 ***DSP MStandby Requires CD_EMU in SW_WKUP***

CRITICALITY

Low

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 2.0A, 2.0

TDA3x: 2.0A, 2.0, 1.0A, 1.0

DRA78x: 2.0A, 2.0

DM50x: 2.0

Copyright © 2026 Texas Instruments Incorporated

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 2.0A, 2.0

TDA3x: 2.0A, 2.0, 1.0A, 1.0

DRA78x: 2.0A, 2.0

DM50x: 2.0

Copyright © 2026 Texas Instruments Incorporated

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 2.0A, 2.0

TDA3x: 2.0A, 2.0, 1.0A, 1.0

DRA78x: 2.0A, 2.0

DM50x: 2.0

Copyright © 2026 Texas Instruments Incorporated

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
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 uart1_rtsn: CTRL_CORE_PAD_UART1_RTSN.UART1_RTSN_MUXMODE = 0xF
   - Enable DCAN module
     - DCAN1: CM_WKUPAON_DCAN1_CLKCTRL.MODULEMODE = 0x2
   - Perform RAM_INIT sequence
     - DCAN1: CTRL_CORE_CONTROL_IO_2.DCAN_RAMINIT_START = 0x1
     - Poll for CTRL_CORE_CONTROL_IO_2.DCAN_RAMINIT_START = 0x1
   - 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 uart1_rtsn: CTRL_CORE_PAD_UART1_RTSN.UART1_RTSN_MUXMODE = 0xC

REVISIONS IMPACTED

SR 2.0A, 2.0

TDA3x: 2.0A, 2.0, 1.0A, 1.0

DRA78x: 2.0A, 2.0

DM50x: 2.0

Copyright © 2026 Texas Instruments Incorporated

i894 ***DSS VENC Early Termination on Last Line of Frame***

CRITICALITY

Low

DESCRIPTION

When the DSS is used to display analog video via the VENC interface, the DSS to VENC interface drops 22 pixels of data in the bottom/right corner of the video frame. The expected behavior is for every pixel of the video data to be properly managed from the memory/frame buffer, through the DSS via the VENC, to the cvideo_tvout pin. The DSS to VENC interface ignores/drops 22 pixels worth of data in the bottom line, right most pixels of the video frame. This can result in perceived image, for example a box drawn on the periphery of the display, will have a noticeable missing line at the bottom right hand corner.

In most systems, the issue is not perceptible since many Analog TV Monitors do not display the perimeter pixels of the transmitted video frame.

WORKAROUND

None needed if the specific analog TV monitor does not display the final 22 pixels.

If those pixels are visible, in order to minimize the effect of this issue, the bottom most line of the Frame Buffer can be kept a constant black signal (no Chroma, appropriate Luma value for NTSC/PAL).

REVISIONS IMPACTED

SR 2.0A, 2.0

TDA3x: 2.0A, 2.0, 1.0A, 1.0

DRA78x: 2.0A, 2.0

DM50x: 2.0

Copyright © 2026 Texas Instruments Incorporated

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 2.0A, 2.0

TDA3x: 2.0A, 2.0, 1.0A, 1.0

DRA78x: 2.0A, 2.0

DM50x: 2.0

Copyright © 2026 Texas Instruments Incorporated

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 2.0A, 2.0

TDA3x: 2.0A, 2.0, 1.0A, 1.0

DRA78x: 2.0A, 2.0

DM50x: 2.0

Copyright © 2026 Texas Instruments Incorporated

i914 *Limitations with DISPC Write-Back Region-Based Mechanism*

CRITICALITY

Medium

DESCRIPTION

The region-based mechanism in the Write-Back pipeline relies on the SW performing certain tasks (setting and resetting of REGION_BASED bit of DISPC_WB_ATTRIBUTES2 register) inside the WBSYNC_IRQ and WBREGIONBASEDEVENT_IRQ every time those interrupts are triggered. The feature also requires those interrupts to be handled in the same order as they are raised by the HW. In real life use-cases, it is possible, due to system latencies and due to the position of the region being written back, that the above requirements cannot be satisfied during all frames. Under such circumstances following issues can happen

- Frame Drop: Region Based Write-Back can write unwanted data into the memory for some frames which cannot be used reliably by the application, before the HW recovers in subsequent frames.
- Buffer Overflow: Region Based Write-Back can end up writing the entire frame into the memory. The application should therefore ensure that a memory size equivalent to the full frame size is always reserved for the region based write-back
- Stale Configuration: The new Write-Back configuration (for example, BaseAddress) updated in the WBREGIONBASEDEVENT_IRQ is not seen by HW during the next frame, resulting in the HW using the old configuration during the next frame too.

WORKAROUND

1. Frame Drop, Buffer Overflow and Stale configuration
   The detection and recovery from the states mentioned in Description section is to be handled in SW by adding additional checks inside the WBSYNC_IRQ and WBREGIONBASEDEVENT_IRQ interrupt service routines (ISR). The additional checks to be performed in SW are listed below (the below SW steps assume that the priority in which SW checks for the IRQs is such that the WBSYNC_IRQ is always at a higher priority than the WBREGIONBASEDEVENT_IRQ).
   Case 1: WBSYNC_IRQ is delayed
   The WBSYNC_IRQ for Frame-n (Fn) is delayed such that, when inside the WBSYNC_IRQ (Fn) routine SW finds that WBREGIONBASEDEVENT_IRQ (Fn+1) is also set. This implies that the region-based capture in Fn+1 is not proper and should be ignored. The SW then has to reset the REGION_BASED bit of DISPC_WB_ATTRIBUTES2 register in WBSYNC_IRQ (Fn) ISR, which will result in a full frame write-back for the next frame (Fn+2). There will be no WBREGIONBASEDEVENT_IRQ (Fn+2). In the WBSYNC_IRQ (Fn+1), SW will set the REGION_BASED bit of DISPC_WB_ATTRIBUTES2 register again, which will result in proper region being captured from Fn+3 onwards.
   Case 2: WBREGIONBASEDEVENT_IRQ is delayed
   The WBREGIONBASEDEVENT_IRQ for Frame-n (Fn) is delayed such that, when inside the WBSYNC_IRQ (Fn) routine SW finds that WBREGIONBASEDEVENT_IRQ (for Fn) is also set. This implies that the region-based capture in Fn+1 will not be proper and should be ignored. The SW then has to reset the REGION_BASED bit of DISPC_WB_ATTRIBUTES2 register in WBSYNC_IRQ (Fn) ISR, which will result in a full frame capture for the Fn+2 frame. There will be no WBREGIONBASEDEVENT_IRQ (Fn+2). In the next WBSYNC_IRQ (Fn+1), SW will set the REGION_BASED bit of DISPC_WB_ATTRIBUTES2 again, which will result in proper region being captured from Fn+3 onwards. The delay in WBREGIONBASEDEVENT_IRQ can also cause stale configuration (new Write-Back configuration programmed is not updated in the HW for next frame).
   To avoid this SW needs to ensure that a new configuration is programmed only if the WBREGIONBASEDEVENT_IRQ is taken before WBSYNC_IRQ.

Copyright © 2026 Texas Instruments Incorporated

i914 (continued) ***Limitations with DISPC Write-Back Region-Based Mechanism***

This can be achieved by ensuring that a new configuration update is done inside the WBREGIONBASEDEVENT_IRQ routine only after the VSYNC_IRQ or WBSYNC_IRQ are checked and found to be not set (via DISPC_VP1_IRQSTATUS[1] VPVSYNC_IRQ and DISPC_WB_IRQENABLE[4] WBSYNC_EN bits, respectively).

2. Dynamic disabling of Write-Back Region based mechanism
   If the Write-Back Region based mechanism has to be disabled dynamically (keeping the rest of the pipelines and display running as before) then the below sequence needs to be followed:
   a. Disable the REGION_BASED bit of DISPC_WB_ATTRIBUTES2 register
   b. Wait for one frame
   c. Disable the Write-Back pipeline in the next VSYNC ISR (via DISPC_WB_ATTRIBUTES[0] ENABLE bit)

REVISIONS IMPACTED

SR 2.0A, 2.0

TDA3x: 2.0A, 2.0, 1.0A, 1.0

DRA78x: 2.0A, 2.0

DM50x: 2.0

Copyright © 2026 Texas Instruments Incorporated

i916 ***QSPI: Relying on Pull-ups for 'Default Values' Fails at Higher Clock Rates, Causing EDMA Read Failures***

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 the SoC QSPI IP.

REVISIONS IMPACTED

SR 2.0A, 2.0

TDA3x: 2.0A, 2.0, 1.0A, 1.0

DRA78x: 2.0A, 2.0

DM50x: 2.0

Copyright © 2026 Texas Instruments Incorporated

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 reset. This occurs due to an internal node floating to a random state inside of the I2C output buffer during reset.

Note that other I2C instances on the SoC are not affected by this issue since they use a standard LVCMOS IO buffer (not the I2C IO 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 2.0A, 2.0

TDA3x: 2.0A, 2.0, 1.0A, 1.0

DRA78x: 2.0A, 2.0

DM50x: 2.0

Copyright © 2026 Texas Instruments Incorporated

i936 ***DSS LCD/DPI Out Field Reversal in Interlaced RGB Mode***

CRITICALITY

High

DESCRIPTION

Interlaced RGB mode if set on DSS DPI outputs (VOUT1/2/3) can result in display artifacts due to field reversal. DSS does not support an interlaced compliant VSYNC signal on the DPI output. If the external device/monitor hooked to the DSS in this mode requires an interlaced compliant VSYNC signal then it will have this issue randomly. If the external monitor is able to lock the polarity correctly then the display will be correct otherwise the display will generate field reversed view.

WORKAROUND

Use Interlaced YUV mode with embedded sync (BT.656, BT.1120) if possible.

REVISIONS IMPACTED

SR 2.0A, 2.0

TDA3x: 2.0A, 2.0, 1.0A, 1.0

DRA78x: 2.0A, 2.0

DM50x: 2.0

Copyright © 2026 Texas Instruments Incorporated

i938 ***SD-DACxx: SD-DAC CVIDEO_TVOUT Can Power up to an Inoperable State***

CRITICALITY

High

DESCRIPTION

The CVIDEO_TVOUT output from the SD-DAC can get stuck at a logic-high (logic-1) state on power-up and can remain in that state. A power cycle or hardware reset may not change this state of the CVIDEO_TVOUT output.

The SD-DAC CVIDEO_TVOUT can get stuck at logic-high due to an uninitialized flip-flop inside the VENC that intermittently powers up to an invalid (logic-1) state, which then disables the VENC Composite Video Baseband Signal (CVBS) output. This uninitialized flip-flop may not be reset by a hardware reset, preventing a reset-cycle from resolving this issue.

WORKAROUND

A software API has been created to reset the affected flip-flop power-up state and correct the issue. The API uses internal test logic to reset the DSS while the clocks are running to the DSS and the VENC.

This API can be called from the SBL (recommended) or Application (feasible for existing systems). Note that this API needs to be executed during initial boot prior to the VENC being enabled/used in the application. The API should be run every time DSS is powered down completely and brought back up. The DSS is considered powered down if the PRCM is programmed after boot to transition the PM_DSS_PWRSTST[1:0] POWERSTATEST to 0x0.

The software patch for enabling this workaround is available at http://processors.wiki.ti.com/images/d/d8/PDK-3634.zip.

REVISIONS IMPACTED

SR 2.0A, 2.0

TDA3x: 2.0A, 2.0, 1.0A, 1.0

DRA78x: 2.0A, 2.0

DM50x: 2.0

Copyright © 2026 Texas Instruments Incorporated

i939 *MCAN: Message Transmitted with Wrong Arbitration and Control Fields (Early Start of Frame)*

CRITICALITY

Low

DESCRIPTION

The erratum is limited to the case when MCAN is in state "Receiver" (MCAN_PSR[4:3] ACT = "10") with no pending transmission (no MCAN_TXBRP bit set) and a new transmission is requested before the 3rd bit of Intermission is reached and this 3rd bit of intermission is seen dominant. This issue occurs only with disturbed CAN bus.

Under the following conditions, a message with wrong ID, format, and DLC is transmitted:

- MCAN is in state "Receiver" (MCAN_PSR[4:3] ACT = "10"), no pending transmission
- A new transmission is requested before the 3rd bit of Intermission is reached
- The CAN bus is sampled dominant at the third bit of Intermission which is treated as SoF (see ISO 11898-1:2015 Section 10.4.2.2).

Under the conditions listed above, it may happen, that:

- The shift register is not loaded with ID, format, and DLC of the requested message
- The MCAN will start arbitration with wrong ID, format, and DLC on the next bit
- In case the ID wins arbitration, a CAN message with valid CRC is transmitted
- In case this message is acknowledged, the ID stored in the Tx Event FIFO is the ID of the requested Tx message and not the ID of the message transmitted on the CAN bus, no error is detected by the transmitting MCAN
- If the message loses arbitration or is disturbed by an error, it is retransmitted with correct arbitration and control fields.

WORKAROUND

Workaround 1:

Request a new transmission only if another transmission is already pending or when the MCAN is not in state "Receiver" (when MCAN_PSR[4:3] ACT ≠ "10").

To avoid activating the transmission request in the critical time window between the sample points of the 2nd and 3rd bit of Intermission, following can be done:

- Evaluate the Rx Interrupt flags MCAN_IR[19] DRX, MCAN_IR[0] RF0N, MCAN_IR[4] RF1N which are set at the last bit of EoF when a received and accepted message gets valid. Prevent the transmission of any message within 3 bit times after detecting the Rx interrupt flags.

Workaround 2:

A checksum covering arbitration and control fields can be added to the data field of the message to be transmitted, to detect frames transmitted with wrong arbitration and control fields.

Workaround 3:

Set MCAN_CCCR[0] INIT bit, add transmission request for the message and then clear MCAN_CCCR[0] INIT bit. This needs to be done for each message to be transmitted.

Workaround 4:

Keep the number of pending transmissions always at ">0" by frequently requesting a message so that the condition "no pending transmission" is never met. The frequently requested message may be given a low priority, losing arbitration to all other messages.

In case, where all the nodes present on the CAN bus have this workaround implemented, at least two nodes shall have this frequently sent low priority message with different

Copyright © 2026 Texas Instruments Incorporated

i939 (continued) ***MCAN: Message Transmitted with Wrong Arbitration and Control Fields (Early Start of Frame)***

priorities/identifiers. Rest of the nodes can have one of these priorities/identifiers (same) as the priority of this frequently sent message.

Comparison within available Workarounds is shown in Table 2-1:

Table 2-1. Comparison within available Workarounds

Workaround #	Advantages	Disadvantages
1	• Easy to implement. • Fix is only limited to a local node with the erratum.	• Only viable in interrupt context. • Effectiveness is vastly dependent on interrupt service latency. • Application becomes complicated.
2	• Simple and easy to implement.	• All nodes on the bus must implement this WA. Restricting use of device for end points and all the existing nodes needs to be updated to support this WA.
3	• Simple and easy to implement.	• Node will miss any messages received during this sequence. • System (including existing nodes) needs to be updated for detecting the dropped/ missed message.
4	• DMA can be used to send the frequently sent low priority message to reduce CPU load. • Most effective and viable amongst available workarounds. • Fix is only limited to a local node with the erratum.	• 100% CAN Bus utilization at all times. • Increased CPU load in case CPU is used to send the message.

REVISIONS IMPACTED

SR 2.0A, 2.0

TDA3x: 2.0A, 2.0

DRA78x: 2.0A, 2.0

Copyright © 2026 Texas Instruments Incorporated

i940 *MPU COUNTER_REALTIME saturates after several hundred days*

Details

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 (200 to 388 days).

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.

Copyright © 2026 Texas Instruments Incorporated

Chapter 3

Silicon Limitations

Copyright © 2026 Texas Instruments Incorporated

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 2.0A, 2.0

TDA3x: 2.0A, 2.0, 1.0A, 1.0

DRA78x: 2.0A, 2.0

DM50x: 2.0

Copyright © 2026 Texas Instruments Incorporated

i873 *DSS: First Two Columns of Active Video Are Always Black at the Output of Video Encoder*

CRITICALITY

Low

DESCRIPTION

The Video Encoder (VENC) should accurately output the input frame buffer in DDR without any artifacts. In actual operation the first two columns of pixels from the input frame buffer are ignored by the VENC. The VENC outputs black pixels in place of the original pixels for the first two columns.

The issue is due to a two cycle difference in the expected versus actual delay between the VENC and the Display Controller (DISPC).

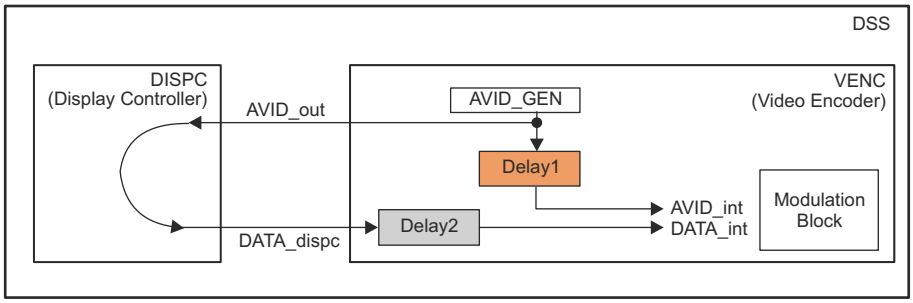


Figure 3-1. VENC and Display Controller

For the design to work, Delay1 should be equal to Delay2. In the hardware implementation it is Delay1 = Delay2 +2. This results in the first two pixels of every line being missed.

The system impact depends on the specific use-case. Most TVs have over-scan (the visible pixels on the display are less than the pixels transmitted), and in those cases the bug will not impact any functionality. In other cases, the display is missing the first two vertical columns of data which could be critical for the use-case.

WORKAROUND

A software work-around is readily available whereby the incoming frame data is padded with two extra pixels at every line inside the DISPC and then the extended window is sent to the VENC. The VENC will then display the correct frame onto the TV (the padded pixels get dropped by the VENC in place of actual pixels).

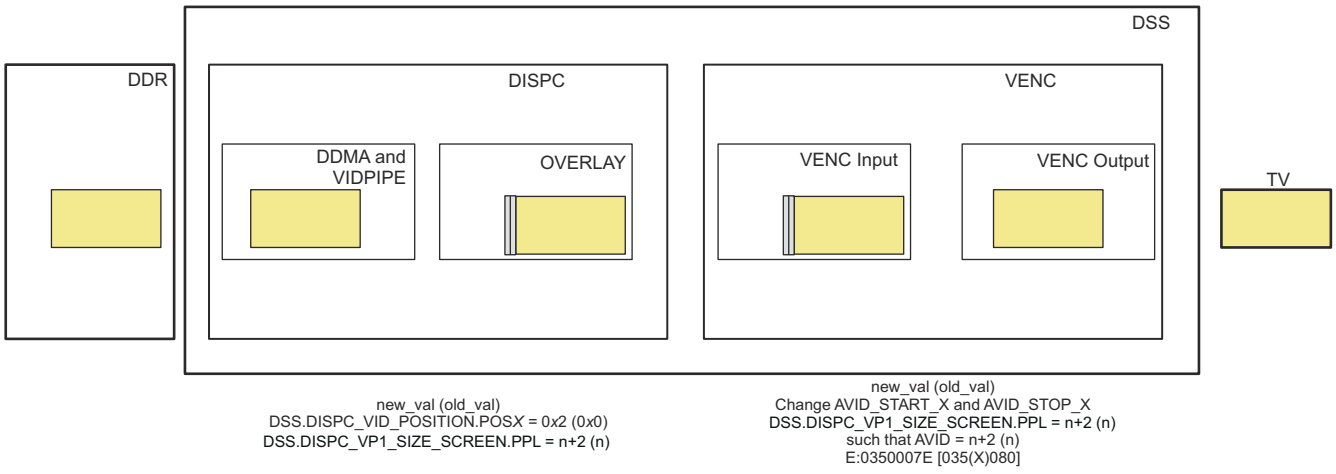


Figure 3-2. VENC

Copyright © 2026 Texas Instruments Incorporated

i873 (continued) ***DSS: First Two Columns of Active Video Are Always Black at the Output of Video Encoder***

Table 3-1. Register Changes for Software Workaround

Register	Address	Field Name	Current Value	New Value
DISPC_VID_POSITION	0x5801 7214	POSX	X Eg: 0x0	X+2 Eg:0x2
DISPC_VP1_SIZE_SCREEN	0x5801 AC50	PPL	N	N+2
VENC_AVID_START_STOP_X	0x5800 5090	AVID_START_X	S	S-2

REVISIONS IMPACTED

SR 2.0A, 2.0

TDA3x: 2.0A, 2.0, 1.0A, 1.0

DRA78x: 2.0A, 2.0

DM50x: 2.0

Copyright © 2026 Texas Instruments Incorporated

i876 ***DVFS Not Supported***

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 does not support DVFS.

WORKAROUND

The supply rails and OPP conditions (Frequency/Voltage) should be set at boot-time and remain at that OPP voltage level during device run-time.

REVISIONS IMPACTED

SR 2.0A, 2.0

TDA3x: 2.0A, 2.0, 1.0A, 1.0

DRA78x: 2.0A, 2.0

DM50x: 2.0

Copyright © 2026 Texas Instruments Incorporated

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 2.0A, 2.0

TDA3x: 2.0A, 2.0, 1.0A, 1.0

DRA78x: 2.0A, 2.0

DM50x: 2.0

Copyright © 2026 Texas Instruments Incorporated

Chapter 4

Silicon Cautions

Copyright © 2026 Texas Instruments Incorporated

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 (CORE, DSPEVE, etc.). Refer to the Data Manual for AVS requirements.
- 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 2.0A, 2.0

TDA3x: 2.0A, 2.0, 1.0A, 1.0

DRA78x: 2.0A, 2.0

DM50x: 2.0

Copyright © 2026 Texas Instruments Incorporated

i827	***Thermal Alert Will Not Be Generated When Bandgap Is Configured in "Smart Idle" Mode***
CRITICALITY	Low
DESCRIPTION	When the Core bandgap cell is in smart idle state (CTRL_CORE_BANDGAP_MASK_1[31:30] SIDLEMODE = 0x2), the Core bandgap will not generate THERMAL_ALERT and hence no Thermal Alert interrupt will be generated by the control module.
GUIDELINES	Do not idle the Control Module Bandgap (keep CTRL_CORE_BANDGAP_MASK_1 [31:30] SIDLEMODE = 0x0)
REVISIONS IMPACTED	SR 2.0A, 2.0 TDA3x: 2.0A, 2.0, 1.0A, 1.0 DRA78x: 2.0A, 2.0 DM50x: 2.0

Copyright © 2026 Texas Instruments Incorporated

i828 *IO Input Glitches May Occur When Switching Pull Type and Mux Mode Simultaneously*

CRITICALITY

Low

DESCRIPTION

If any multiplexed IO pin is reconfigured with a single register (CTRL_CORE_PAD_x) write where:

- Input in mux mode MA (where MA is any available mode) with pull type PA (where PA is pullup or pulldown) enabled:
- (CTRL_CORE_PAD_x[17] PULLTYPESELECT = 0xX (where X = 0 for pulldown or X = 1 for pullup) and CTRL_CORE_PAD_x[16] PULLUDENABLE = 0x1 and CTRL_CORE_PAD_x[3:0] MUXMODE = 0xX (where X equals to any available value between 0 and 15)

This could result in a high level glitch being seen on the corresponding peripheral input. In some cases, such glitch could yield unexpected or undesired results depending on the corresponding peripheral function.

GUIDELINES

To avoid such glitches is to use two separate register writes to transition to the desired state. The first write can change the pulltype (for example from pullup enabled to pulldown enabled) without changing the mux mode. The second write can change the mux mode.

REVISIONS IMPACTED

SR 2.0A, 2.0

TDA3x: 2.0A, 2.0, 1.0A, 1.0

DRA78x: 2.0A, 2.0

DM50x: 2.0

Copyright © 2026 Texas Instruments Incorporated

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 2.0A, 2.0

TDA3x: 2.0A, 2.0, 1.0A, 1.0

DRA78x: 2.0A, 2.0

DM50x: 2.0

Copyright © 2026 Texas Instruments Incorporated

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/or DSS 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 and/or DSS 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 2.0A, 2.0

TDA3x: 2.0A, 2.0, 1.0A, 1.0

DRA78x: 2.0A, 2.0

DM50x: 2.0

Copyright © 2026 Texas Instruments Incorporated

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.

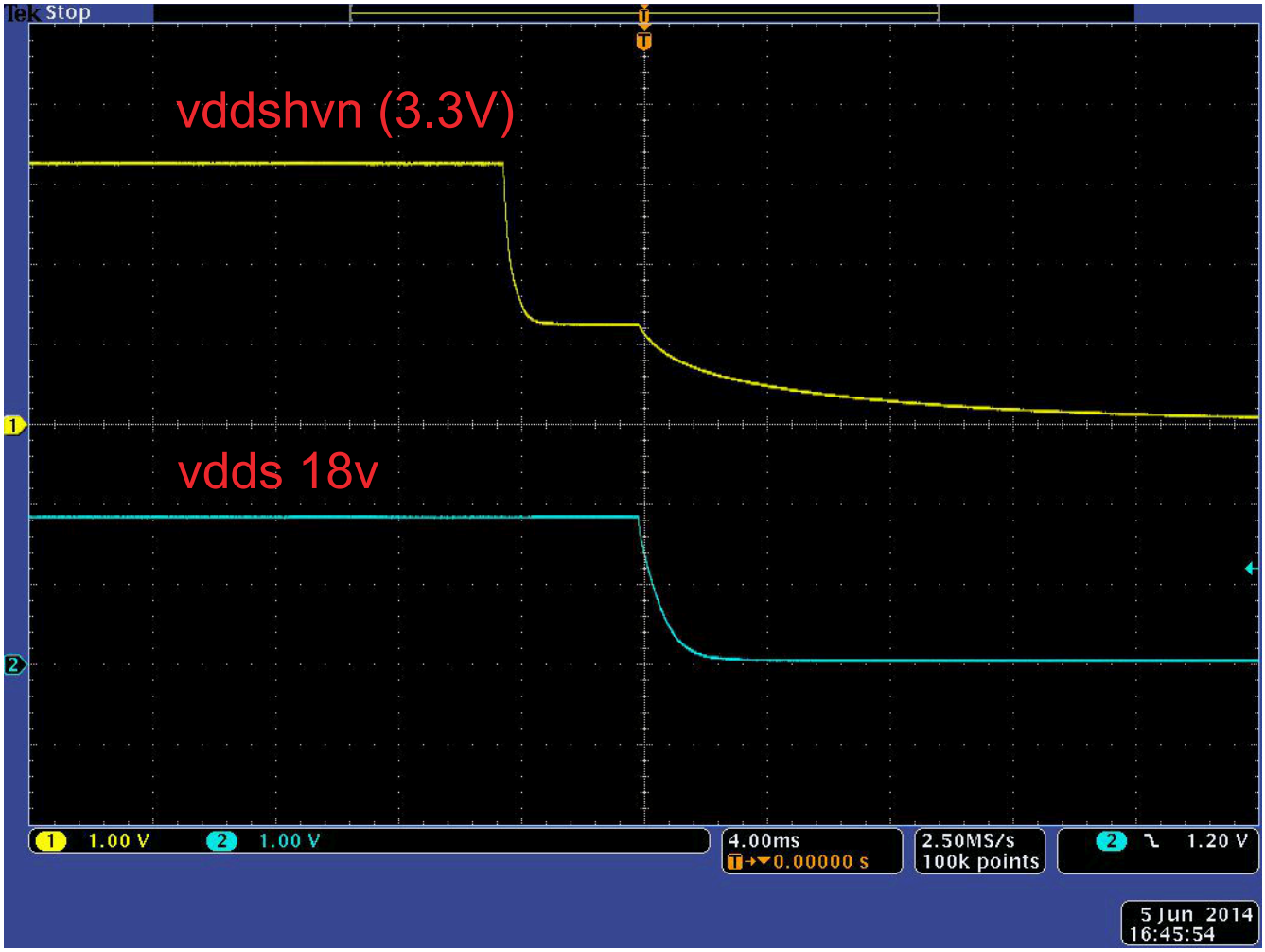


Figure 4-1. PAB_RevA

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

Copyright © 2026 Texas Instruments Incorporated

i864 (continued) ***VDDS18V to VDDSHVn Current Path***

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 uA 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 2.0A, 2.0

TDA3x: 2.0A, 2.0, 1.0A, 1.0

DRA78x: 2.0A, 2.0

DM50x: 2.0

Copyright © 2026 Texas Instruments Incorporated

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 SPRAC62 for information on how to best work around this issue in a given system.

REVISIONS IMPACTED

SR 2.0A, 2.0

TDA3x: 2.0A, 2.0, 1.0A, 1.0

DRA78x: 2.0A, 2.0

DM50x: 2.0

Copyright © 2026 Texas Instruments Incorporated

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 2.0A, 2.0

TDA3x: 2.0A, 2.0, 1.0A, 1.0

DRA78x: 2.0A, 2.0

DM50x: 2.0

Copyright © 2026 Texas Instruments Incorporated

Revision History

Changes from October 31, 2019 to November 15, 2026 (from Revision B (October 2019) to Revision C (November 2026))	Page
• Added Advisory i940; MPU COUNTER_REALTIME saturates after several hundred days	36

Copyright © 2026 Texas Instruments Incorporated

This page intentionally left blank.

Copyright © 2026 Texas Instruments Incorporated

IMPORTANT NOTICE AND DISCLAIMER

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

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, 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 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, TI’s General Quality Guidelines, 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. Unless TI explicitly designates a product as custom or customer-specified, TI products are standard, catalog, general purpose devices.

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

Copyright © 2026, Texas Instruments Incorporated

Last updated 10/2025