Errata

AM263x Sitara™ Microcontroller Silicon Revision 1.0A, 1.1

ABSTRACT

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

Table of Contents

1 Usage Notes and Advisories Matrices

Table 1-1 lists all usage notes and the applicable silicon revision(s). Advisories Matrix lists all advisories, modules affected, and the applicable silicon revision(s).

Table 1-1. Usage Notes Matrix

Module	DESCRIPTION	SILICON REVISIONS AFFECTED	
		AM263x	
		1.0A	1.1
CLOCKS	i2324 — No synchronizer present between GCM and GCD status signals	YES	YES
QSPI	i2364 — QSPI: Access to address beyond 8MB is not supported in mem map mode	YES	YES
VDDA	i2348 — VDDA1V8 Static Power leakage	YES	NO

Advisories Matrix

MODULE	DESCRIPTION	SILICON REVISIONS AFFECTED	
		AM263x	
		1.0A	1.1
ADC	i2346 — ADC result has Error when switching between odd and even channels	YES	NO
ADC	i2347 — VREF current consumption of ADC is random at powerup	YES	NO
ADC	i2349 — ADC VrefHi Loading increase in powerdown	YES	NO
CONTROLSS	i2352 — CONTROLSS-SDFM: Dynamically Changing Threshold Settings (LLT, HLT), Filter Type, or COSR Settings Will Trigger Spurious Comparator Events	YES	YES
CONTROLSS	i2353 — CONTROLSS-SDFM: Dynamically Changing Data Filter Settings (Such as Filter Type or DOSR) Will Trigger Spurious Data Acknowledge Events	YES	YES
CONTROLSS	i2354 — CONTROLSS-SDFM: Two Back-to-Back Writes to SDCPARMx Register Bit Fields CEVT1SEL, CEVT2SEL, and HZEN Within Three SD-Modulator Clock Cycles can Corrupt SDFM State Machine, Resulting in Spurious Comparator Events	YES	YES
CONTROLSS	i2355 — CONTROLSS-ADC: DMA Read of Stale Result	YES	YES
CONTROLSS	i2356 — CONTROLSS-ADC: Interrupts may Stop if INTxCONT (Continue-to-Interrupt Mode) is not Set	YES	YES
CONTROLSS	i2357 — CONTROLSS-ePWM: An ePWM Glitch can Occur if a Trip Remains Active at the End of the Blanking Window	YES	YES
CONTROLSS	i2358 — CONTROLSS-ePWM: Trip Events Will Not be Filtered by the Blanking Window for the First 3 Cycles After the Start of a Blanking	YES	YES
CONTROLSS	i2359 — CONTROLSS-CMPSS:Prescaler counter behaviour different from spec when DACSOURCE is made 0 or reconfigured as 1	YES	YES
CONTROLSS	i2405 — CONTROLSS: Race condition OUTPUT_XBAR and PWM_XBAR resulting in event miss	YES	YES
CPSW	i2345 — CPSW: Ethernet Packet corruption occurs if CPDMA fetches a packet which spans across memory banks	YES	YES
CPSW	i2401 — CPSW: Host Timestamps Cause CPSW Port to Lock up	YES	YES
CPSW	i2402 — CPSW: Ethernet to Host Checksum Offload does not work	YES	YES
CRC	i2386 — CRC: CRC 8-bit data width and CRC8-SAE-J1850 and CRC8-H2F possible use in CAN module is not supported	YES	YES
DCC	i2395 — DCC Module Frequency Comparison can Report Erroneous Results	YES	YES
GPMC	i2313 — GPMC: Sub-32-bit read issue with NAND and FPGA/FIFO	YES	YES
M4 ROM	i2403 — M4 ROM: SBL redundant boot image feature not supported on HSSE devices	NO	YES
MBOX	i2404 — MBOX: Race condition in mailbox registers resulting in events miss	YES	YES
McSPI	i2350 — McSPI data transfer using EDMA in ‘ABSYNC’ mode stops after 32 bits transfer	YES	YES
MDIO	i2329 — MDIO interface corruption (CPSW and PRU-ICSS)	YES	NO
PBIST	i2374 — PBIST fails if clock frequency of R5SS_CORE_CLK is not same as R5FSS_CLK_SELECTED frequency	YES	YES
REC	i2392 — Race condition in mem-init capture registers resulting in events miss	YES	YES

Advisories Matrix (continued)

MODULE	DESCRIPTION	SILICON REVISIONS AFFECTED	
		AM263x	
		1.0A	1.1
REC	i2394 — Race condition in interrupt and error aggregator capture registers resulting in events miss	YES	YES
SFDM	i2375 — SDFM module event flags (SDIFLG.FLTx_FLG_CEVTx) do not get set again if the comparator event is still active and digital filter path (using SDCOMPxCTL.CEVTxDIGFILTSEL) is being selected	YES	YES
UART	i2310 — USART: Erroneous triggering of timeout interrupt	YES	YES
UART	i2311 — USART: Spurious DMA Interrupts	YES	YES

2 Silicon Usage Notes

i2324 *No synchronizer present between GCM and GCD status signals*

Details: There is no synchronizer in between GCM and GCD, so the clock configuration register reads may be incorrect momentarily.

Severity: Minor

Workaround(s): Poll for the status registers change until it reflects the programmed SRC_SEL and DIV values.

i2348 *VDDA1V8 Static Power leakage*

Details: VDDA1V8 has static leakage when the device is booted if DACVREF is at ground.

Workaround(s): DAC reference voltage and VDDA1p8V must be shorted together.

i2364 *QSPI: Access to address beyond 8MB is not supported in mem map mode*

Details: Address lines going out from SoC interconnect to QSPI controller are 23. Hence this limits the usage of QSPI flash memory to 8MB per chip select in memmap mode.

Workaround(s):

None

3 Silicon Advisories

i2310 *USART: Erroneous clear/trigger of timeout interrupt*

Details: The USART may erroneously clear or trigger the timeout interrupt when RHR/MSR/LSR registers are read.

Workaround(s):

For CPU use-case.

If the timeout interrupt is erroneously cleared:

-This is OK since the pending data inside the FIFO will retrigger the timeout interrupt

If timeout interrupt is erroneously set, and the FIFO is empty, use the following SW workaround to clear the interrupt:

- Set a high value of timeout counter in TIMEOUTH and TIMEOUTL registers

- Set EFR2 bit 6 to 1 to change timeout mode to periodic

- Read the IIR register to clear the interrupt

- Set EFR2 bit 6 back to 0 to change timeout mode back to the original mode

For DMA use-case.

If timeout interrupt is erroneously cleared:

-This is OK since the next periodic event will retrigger the timeout interrupt

-User must ensure that RX timeout behavior is in periodic mode by setting EFR2 bit6 to 1

If timeout interrupt is erroneously set:

-This will cause DMA to be torn down by the SW driver

i2310 (continued) ***USART: Erroneous clear/trigger of timeout interrupt***

-OK since next incoming data will cause SW to setup DMA again

i2311 ***USART Spurious DMA Interrupts***

Details:

Spurious DMA interrupts may occur when DMA is used to access TX/RX FIFO with a non-power-of-2 trigger level in the TLR register.

Workaround(s):

Use power of 2 values for TX/RX FIFO trigger levels (1, 2, 4, 8, 16, and 32).

i2313 ***GPMC: Sub-32-bit read issue with NAND and FPGA/FIFO***

Details:

Sub-32-bit reads on the GPMC interface will miss portions of the data, which will result in incorrect read data. This includes 8-bit or 16-bit reads from a NAND device or from an FPGA or FIFO interface. Note that 3-byte accesses are not allowed on the GPMC interface.

Workaround(s):

Read accesses on the GPMC interface must be performed as 32-bit reads. Writes are not affected by this erratum.

i2329 ***MDIO: MDIO interface corruption (CPSW and PRU-ICSS)***

Details:

It is possible that the MDIO interface of all instances of CPSW and PRU-ICSS peripherals (if present) returns corrupt read data on MDIO reads (e.g. returning stale or previous data), or sends incorrect data on MDIO writes. It is also possible that the MDIO interface becomes unavailable until the next peripheral reset (either by LPSC reset or global device reset with reset isolation disabled in case of CPSW).

Possible system level manifestations of this issue could be (1) erroneous ethernet PHY link down status (2) inability to properly configure an ethernet PHY over MDIO (3) incorrect PHY detection (e.g. wrong address) (4) read or write timeouts when attempting to configure PHY over MDIO.

For boot mode (only CPSW if supported), there is no workaround to guarantee the primary ethernet boot is successful. If this exception occurs during primary boot, the boot may possibly initiate retries which may or may not be successful. If the retries are unsuccessful, this would result in an eventual timeout and transition to the backup boot mode (if one is selected). If no backup boot mode is selected, then such failure will result in a timeout and force device reset via chip watchdog after which the complete boot process will restart again.

To select a backup boot option (if supported), populate the appropriate pull resistors on the boot mode pins. See boot documentation for each specific device options, but the typical timeout for primary boot attempts over ethernet is 60 seconds.

Workaround(s):

On affected devices, following workaround should be used:

MDIO manual mode: applicable for PRU-ICSS and for CPSW.

MDIO protocol can be emulated by reading and writing to the appropriate bits within the MDIO_MANUAL_IF_REG register of the MDIO peripheral to directly manipulate the MDIO clock and data pins. Refer to TRM for full details of manual mode register bits and their function.

i2329 (continued) *MDIO: MDIO interface corruption (CPSW and PRU-ICSS)*

In this case the device pin multiplexing should be configured to allow the IO to be controlled by the CPSW or PRU-ICSS peripherals (same as in normal intended operation), but the MDIO state machine must be disabled by ensuring MDIO_CONTROL_REG.ENABLE bit is 0 in the MDIO_CONTROL_REG and enable manual mode by setting MDIO_POLL_REG.MANUALMODE bit to 1.

Contact TI regarding implementation of software workaround.

Note

If using Ethernet DLR (Device Level Ring) (on CPSW or PRU-ICSS) or EtherCat protocol (on PRU-ICSS) there may be significant CPU or PRU loading impact to implement the run-time workaround 1 due to required polling interval for link status checks. Resulting system impact should be considered.

In case of PRU-ICSS, the loading of the software workaround may be reduced by using the MLINK feature of MDIO to do automatic polling of link status via the MIIx_RXLINK input pin to PRU-ICSS which must be connected to a status output from the external PHY which does not toggle while the link is active. Depending on the specified behavior of the external PHY device, this PHY status output may be LED_LINK or LED_SPEED or the logic OR of LED_LINK and LED SPEED. Refer to the MDIO section of TRM for details on using the MLINK feature of MDIO. This feature is not available on the CPSW peripheral.

For EtherCAT implementation on PRU-ICSS, the software workaround will be done in RTUx/ TX_PRUx Core. The core will have to be dedicated for workaround, which means this can’t be used for other purpose. The implementation will support two user access channels for MDIO access. This provides option for R5f core and PRU core to have independent access channel. The APIs will be similar to the ones we will have in RTOS Workaround implementation.

EtherCAT will continue to use PHY fast link detection via MDIO MLINK bypassing state m/c for link status (as this path is not affected by errata). This makes sure that cable redundancy related latency requirements are still met.

i2329 (continued) ***MDIO: MDIO interface corruption (CPSW and PRU-ICSS)***

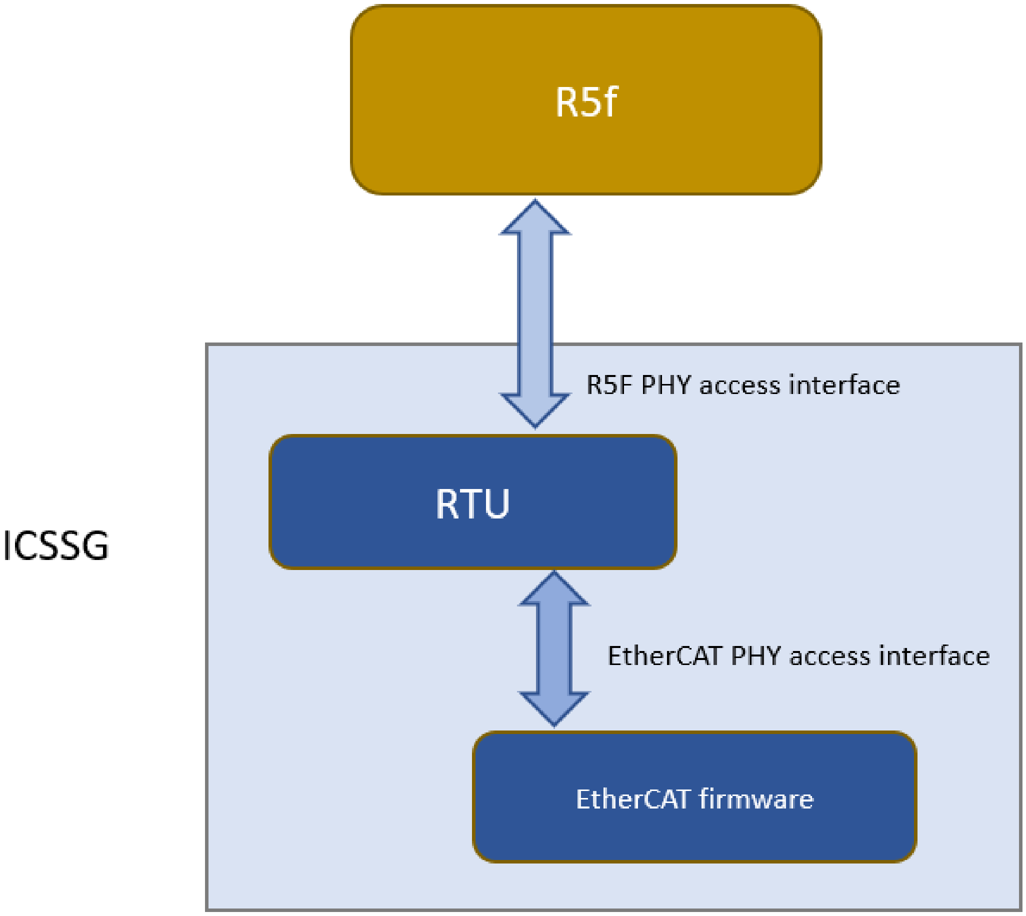


Figure 3-1. MDIO Emulation via Manual Mode using PRU Core

i2345 ***CPSW: Ethernet Packet corruption occurs if CPDMA fetches a packet which spans across memory banks***

Details: Each memory bank in SoC has a separate memory controller. Even though memory addresses are contiguous, each bank is a separate entity with a separate controller.

If a memory bank received a memory request say 32 bytes and address of memory request is 16 bytes before end of memory bank, the behavior of the memory controller will be:

When the memory controller encounters end of memory bank after 16 bytes it will wrap around and give 16 bytes from the start of the memory bank.

This results in the packet corruption.

Workaround(s): Ensure from application side single ethernet packet does not span across memory banks.

i2346 ***ADC result has Error when switching between odd and even channels***

Details: When a ADC conversion sequence involves sampling Odd and Even channels, there is a error in the conversion result whenever there is a switch from odd channel to even channel and vice versa.

i2346 (continued) — *ADC result has Error when switching between odd and even channels*

No error seen when the conversion involves switching between even channels only or odd channels only.

Workaround(s): The first sample after odd to even or even to odd channel switch must have smallest acquisition window and that result must be ignored.

i2347 — *VREF current consumption of ADC is random at powerup*

Details: VREF current consumption is high (1.6 mA)after PORZ and goes low after enabling the ADC via MMR.

The initial current consumption is random at each PORZ cycle.

Workaround(s): Never disable the ADC without resetting the DTC.

i2349 — *ADC VrefHi Loading increase in powerdown*

Details: If ADC is disabled after conversion, the loading on reference increases by 2 mA.

Workaround(s): Never disable the ADC without resetting the DTC.

i2350 — *McSPI: McSPI data transfer using EDMA in ‘ABSYNC’ mode stops after 32 bits transfer*

Details: When EDMA is programmed to transfer more than 32 bits of data in to McSPI Tx FIFO (32 Bytes), it stops working after transferring only first 32 bits data in to the FIFO.

This issue is observed only in “ABSYNC” mode of EDMA where the EDMA is configured such that transfer size is more than 32 bits.

When the issue happens the EDMA neither transferring the data and completing it nor raising any error as vbusp_sdone signal is not getting generated by McSPI for transaction from EDMA.

SPI RX mode is not affected this issue.

Workaround(s): Option1: Use ASYNC mode of EDMA for McSPI TX operation

Option2: Use acnt=4, bcnt=1, ccnt=1 if ABSYNC mode is used for McSPI TX operation

i2352 — *CONTROLSS-SDFM: Dynamically Changing Threshold Settings (LLT, HLT), Filter Type, or COSR Settings Will Trigger Spurious Comparator Events*

Details: When SDFM comparator settings—such as filter type, lower/upper threshold, or comparator OSR (COSR) settings—are dynamically changed during run time, spurious comparator events will be triggered. The spurious comparator event will trigger a corresponding CPU interrupt, CLA task, ePWM X-BAR events, and GPIO output X-BAR events if configured appropriately.

Workaround(s): When comparator settings need to be changed dynamically, follow the procedure below to ensure spurious comparator events do not generate a CPU interrupt, CLA event, or X-BAR events (ePWM X-BAR/GPIO output X-BAR events):

1. Disable the comparator filter.
2. Delay for at least a latency of the comparator filter + 3 SD-Cx clock cycles.
3. Change comparator filter settings such as filter type, COSR, or lower/upper threshold.
4. Delay for at least a latency of the comparator filter + 5 SD-Cx clock cycles.
5. Enable the comparator filter.

i2353 *CONTROLSS-SDFM: Dynamically Changing Data Filter Settings (Such as Filter Type or DOSR) Will Trigger Spurious Data Acknowledge Events*

Details: When SDFM data settings—such as filter type or DOSR settings—are dynamically changed during run time, spurious data-filter-ready events will be triggered. The spurious data-ready event will trigger a corresponding CPU interrupt, CLA task, and DMA trigger if configured appropriately.

Workaround(s): When SDFM data filter settings need to be changed dynamically, follow the procedure below to ensure spurious data-filter-ready events are not generated:

1. Disable the data filter.
2. Delay for at least a latency of the data filter + 3 SD-Cx clock cycles.
3. Change data filter settings such as filter type and DOSR.
4. Delay for at least a latency of the data filter + 5 SD-Cx clock cycles.
5. Enable the data filter.

i2354 *CONTROLSS-SDFM: Two Back-to-Back Writes to SDCPARMx Register Bit Fields CEVT1SEL, CEVT2SEL, and HZEN Within Three SD-Modulator Clock Cycles can Corrupt SDFM State Machine, Resulting in Spurious Comparator Events*

Details: Back-to-back writes to SDCPARMx register bit fields CEVT1SEL, CEVT2SEL, and HZEN within three SD-modulator clock cycles can potentially corrupt the SDFM state machine, resulting in spurious comparator events, which can potentially trigger CPU interrupts, CLA tasks, ePWM XBAR events, and GPIO output X-BAR events if configured appropriately.

Workaround(s): Avoid back-to-back writes within three SD-modulator clock cycles or have the SDCPARMx register bit fields configured in one register write.

i2355 *CONTROLSS-ADC: DMA Read of Stale Result*

Details: The ADCINT flag can be set before the ADCRESULT value is latched (see the tLAT and tINT(LATE) columns in the ADC Timings tables of the AM263x Technical Reference Manual).

The DMA can read the ADCRESULT value as soon as 3 cycles after the ADCINT trigger is set. As a result, the DMA could read a prior ADCRESULT value when the user expects the latest result if all of the following are true:

- The ADC is in late interrupt mode.
- The ADC operates in a mode where tINT (LATE) occurs 3 or more cycles before tLAT (ADCCTL2 [PRESCALE] > 2 for 12-bit mode).
- The DMA is triggered from the ADCINT signal.
- The DMA immediately reads the ADCRESULT value associated with that ADCINT signal without reading any other values first.
- The DMA was idle when it received the ADCINT trigger.

Only the DMA reads listed above could result in reads of stale data; the following non-DMA methods will always read the expected data:

- The ADCINT flag triggers a CLA task.
- The ADCINT flag triggers a CPU ISR.
- The CPU polls the ADCINT flag.

Workaround(s): Trigger two DMA channels from the ADCINT flag. The first channel acts as a dummy transaction. This will result in enough delay that the second channel will always read the fresh ADC result.

i2356 CONTROLSS-ADC: Interrupts may Stop if INTxCONT (Continue-to-Interrupt Mode) is not Set

Details:

If ADCINTSELxNx[INTxCONT] = 0, then interrupts will stop when the ADCINTFLG is set and no additional ADC interrupts will occur. When an ADC interrupt occurs simultaneously with a software write of the ADCINTFLGCLR register, the ADCINTFLG will unexpectedly remain set, blocking future ADC interrupts.

Workaround(s):

1. Use Continue-to-Interrupt Mode to prevent the ADCINTFLG from blocking additional ADC interrupts:

   ```
   ADCINTSEL1N2[INT1CONT] = 1;

   ADCINTSEL1N2[INT2CONT] = 1;

   ADCINTSEL3N4[INT3CONT] = 1;

   ADCINTSEL3N4[INT4CONT] = 1;
   ```
2. Ensure there is always sufficient time to service the ADC ISR and clear the ADCINTFLG before the next ADC interrupt occurs to avoid this condition.
3. Check for an overflow condition in the ISR when clearing the ADCINTFLG. Check ADCINTOVF immediately after writing to ADCINTFLGCLR; if it is set, then write ADCINTFLGCLR a second time to ensure the ADCINTFLG is cleared. The ADCINTOVF register will be set, indicating an ADC conversion interrupt was lost.

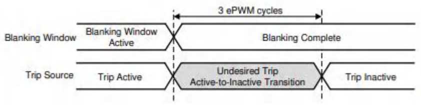
   ```
   AdcaRegs.ADCINTFLGCLR.bit.ADCINT1 = 1; //clear INT1 flag

   if(1 == AdcaRegs.ADCINTOVF.bit.ADCINT1) //ADCINT overflow

   {

   AdcaRegs.ADCINTFLGCLR.bit.ADCINT1 = 1; //clear INT1 again

   // If the ADCINTOVF condition will be ignored by the application

   // then clear the flag here by writing 1 to ADCINTOVFCLR.

   // If there is a ADCINTOVF handling routine, then either insert

   // that code and clear the ADCINTOVF flag here or do not clear

   // the ADCINTOVF here so the external routine will detect the

   // condition.

   // AdcaRegs.ADCINTOVFCLR.bit.ADCINT1 = 1; // clear OVF
   ```

i2357 CONTROLSS-ePWM: An ePWM Glitch can Occur if a Trip Remains Active at the End of the Blanking Window

Details:

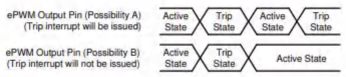
The blanking window is typically used to mask any PWM trip events during transitions which would be false trips to the system. If an ePWM trip event remains active for less than three ePWM clocks after the end of the blanking window cycles, there can be an undesired glitch at the ePWM output.

The following picture illustrates the time period which could result in an undesired ePWM output.



The following picture illustrates the two potential ePWM outputs possible if the trip event ends within 1 cycle before or 3 cycles after the blanking window closes.

i2357 (continued) — *CONTROLSS-ePWM: An ePWM Glitch can Occur if a Trip Remains Active at the End of the Blanking Window*


ePWM Output Pin (Possibility A)
(Trip interrupt will be issued)
Active State
Trip State
Active State
Trip State
ePWM Output Pin (Possibility B)
(Trip interrupt will not be issued)
Active State
Trip State
Active State

Workaround(s): Avoid configuration of blanking window such that the trip input would fall in this range (1 cycle before and 3 cycles after the blanking window closure).

i2358 — *CONTROLSS-ePWM: Trip Events Will Not be Filtered by the Blanking Window for the First 3 Cycles After the Start of a Blanking*

Details: The Blanking Window will not blank trip events for the first 3 cycles after the start of a Blanking Window. DCEVTFILT may continue to reflect changes in the DCxEVTy signals. If DCEVTFILT is enabled, this may impact subsequent subsystems that are configured (for example, the Trip Zone submodule, TZ interrupts, ADC SOC, or the PWM output).

Workaround(s): Start the Blanking Window 3 cycles before blanking is required. If a Blanking Window is needed at a period boundary, start the Blanking Window 3 cycles before the beginning of the next period. This works because Blanking Windows persist across period boundaries.

i2359 — *CONTROLSS-CMPSS:Prescaler counter behaviour different from spec when DACSOURCE is made 0 or reconfigured as 1*

Details: While the prescaler is running if we make DACSOURCE = 0 the prescale counter will not reset and and if the enable condition is LOW the value stays and when the DACSOURCE is again configured as 1 the counter starts from the previous value which was retained. This bug is present only when DACSOURCE is configured during the prescale counter running.

Workaround(s): Issue a soft reset between DACSOURCE configuration which is not a dynamic configuration.

i2374 *PBIST fails if clock frequency of R5SS_CORE_CLK is not same as R5FSS_CLK_SELECTED frequency*

Details

The R5SS memories receive the R5SS CPU clock “R5SS_CORE_CLK” which is derived from R5SS_CLOCK_SELECTED root clock using programmable divider. When R5SS memories are tested using PBIST controller, the PBIST controller receives R5SS_CLOCK_SELECTED root clock. PBIST operation fails if different frequencies are chosen for the two clocks.

Workaround

For PBIST to work with R5SS memories the frequency of both clocks need to be same. If application usage requires R5SS_CORE_CLK to be a divided frequency of R5SS_CLOCK_SELECTED, then during PBIST operation of R5SS memories, the application shall ensure the R5SS_CORE_CLK is configured to same frequency as R5SS_CLOCK_SELECTED.

i2375 *SDFM module event flags (SDIFLG.FLTx_FLG_CEVTx) do not get set again if the comparator event is still active and digital filter path (using SDCOMPxCTL.CEVTxDIGFILTSEL) is being selected*

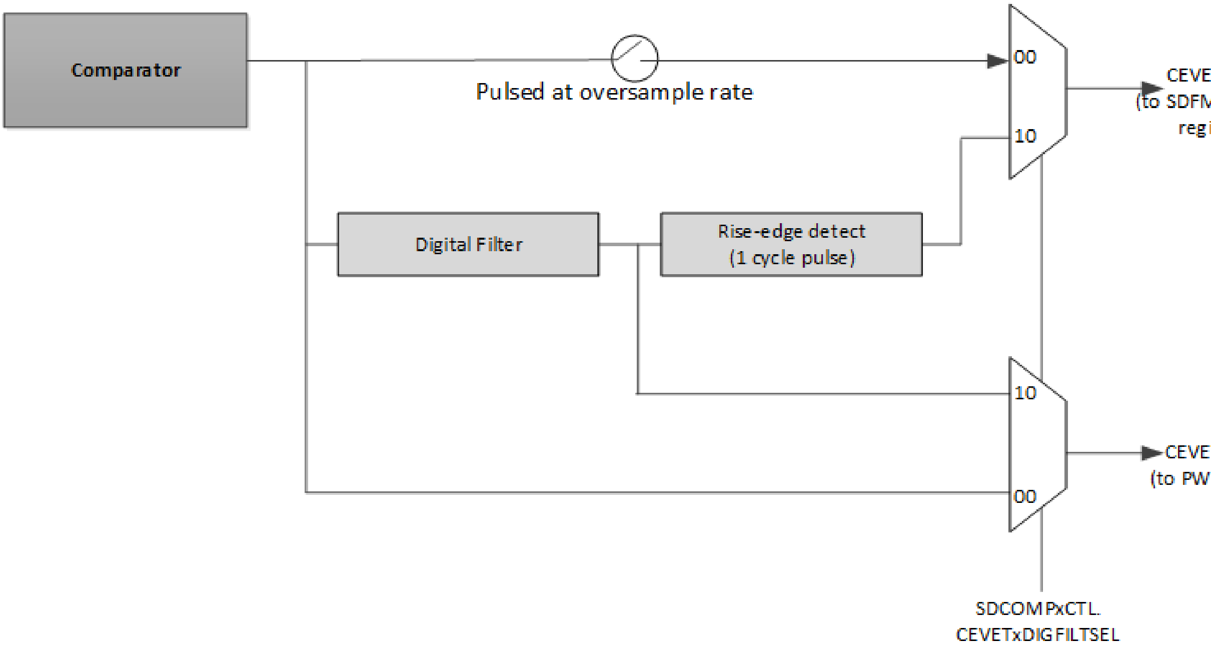
Details

The SDFM module supports a configurable Digital filter on the SDFM COMP output, which can be chosen by application for filtering glitches. The application can choose filtered or raw output of comparator to reach the Event flag register (SDIFLG.FLTx_FLG_CEVTx) and the CEVETxOUT event output of SDFM module as shown in the figure.The Path from Digital filter to event flag register has an Rise edge detection logic as shown in the figure.

When the Digital filter path is chosen, the Event flag register is set only once on the rise edge of Digital filter output. If the event flag register is cleared, it is not set again even if the comparator output is maintained high.

This issue is not present on the CEVETxOUT event going to XBAR.

Also this issue is not present if the raw output path is chosen (i.e CEVTxDIGFILTSEL = 0).



Workaround

If SDFM digital filter is used in application the following workaround options can be considered:

- Option1

XBAR status can be observed instead of the event flag register.

OR

- Option2

1. After selecting the digital filter, wait for the interrupt/Trip.

2. When the interrupt occurs, read the event flag and take appropriate application action to rectify the cause of comparator trip.

3. Before clearing the event flag register, program the unfiltered path.

4. Clear the event flag.

5. Read the event flag and if it stays cleared for at-least one oversampling duration, reprogram the digital filter path.

i2375 (continued) *SDFM module event flags (SDIFLG.FLTx_FLG_CEVTx) do not get set again if the comparator event is still active and digital filter path (using SDCOMPxCTL.CEVTxDIGFILTSEL) is being selected*

Note: In between step 2-4, the PWM trip logic will also be working on unfiltered SDFM comparator out.

i2386 *CRC: CRC 8-bit data width and CRC8-SAE-J1850 and CRC8-H2F possible use in CAN module is not supported*

Details: CRC types CRC8-SAE-J1850 and CRC8-H2F are not supported for 8-bit data width. Minimum data width supported is 16-bit.

Workaround(s): No workaround. It is recommended to not to use the above mentioned unsupported polynomials.

i2392 *Race condition in mem-init capture registers resulting in events miss*

Details: Potential race condition in capture registers resulting in events getting lost while other events in the same register are being cleared by writing to the register. Following registers are impacted by this issue:

MSS_CTRL:*MEMINIT_DONE registers

Workaround(s): Any of the following Workarounds can be used:

Sequentially trigger the mem-init and clear the status before triggering the new mem-init. This is needed if both the status are in the same register.

(OR)

If parallel triggers are must then poll for the all status-bits that got triggered to be 1'b1 and then go and clear the DONE status register

(OR)

Check the MEM_INIT_STATUS register after starting the mem-init and wait the status to go -low by checking it in regular interval and finally clear the DONE status register when the status goes low

i2394 *Race condition in interrupt and error aggregator capture registers resulting in events miss*

Details: Potential race condition in capture registers resulting in events getting lost while other events in the same register are being cleared by writing to the register. Following registers are impacted by this issue:

MSS_CTRL: *INTAGG_STATUS_REG, *TPCC_ERR/INTAGG_STATUS_RAW

Workaround(s): Follow below steps in ISR:

1) Before exiting the ISR read the *_ERRAGG_RAW and check the bit-validity by "anding" with *_ERRAGG_MASK.

2) If any bit is set that-implies there is a interrupt/Error which got missed while clearing the *_ERRAGG_STATUS.

3) Service the corresponding bit in ISR and then exit the ISR. So ISR should be exited after both STATUS and "RAW&MASK" are zero

i2395 *DCC Module Frequency Comparison can Report Erroneous Results*

Details: The Dual-clock Comparator module, which is used to monitor a clock frequency while comparing with a known clock reference, could stop earlier than expected, and, thus, indicating the measured clock frequency to be lower. This is due to a clock domain crossing issue causing a preset to the error detection logic to get triggered.

Workaround(s):

Application code, where possible, could compare the clocks using an alternate EDCC module (Present in MSS)

OR

Multiple measurements can be taken for the same clock pairs and abnormal frequencies reported can be ignored.

Work-around (1): Application code, where possible, could compare the clocks using an alternate EDCC module (Present in MSS)

Work-around (2): Multiple measurements can be taken for the same clock pairs and abnormal frequencies reported can be ignored.

i2401 *CPSW: Host Timestamps Cause CPSW Port to Lock up*

Details: When the CPTS host timestamp is enabled every packet to the internal CPSW port FIFO requires a timestamp from the CPTS. When the packet preamble gets corrupted due to EMI or any other corruption mechanism then a timestamp request might not be sent to the CPTS. In this case the CPTS will not produce the timestamp which causes a lockup condition in the CPSW port FIFO. When the CPTS host timestamp is disabled by clearing the tstamp_en bit in the CPTS_CONTROL register the lockup condition is prevented from occurring.

Workaround(s): None. Ethernet to host timestamps must be disabled.

i2402 *CPSW: Ethernet to Host Checksum Offload does not work*

Details: Ethernet to Host checksum enable has an issue that will send the CPSW into an unrecoverable error state. Host to Ethernet checksum is not effected by the issue.

Workaround(s): None. P0_TX_CHKSUM_EN must not be enabled.

i2403 *M4 ROM: SBL redundant boot image feature not supported on HSSE devices*

Details: SBL redundant boot image feature not supported on HSSE devices

Any corruption on the primary image at the following locations , SBL boot fails to boot from redundant flash region

- Image corruption at middle of the certificate
- Image corruption at end of the certificate
- Image corruption at start of the sbl binary
- Image corruption at middle of the sbl binary
- Image corruption at End of the sbl binary

Workaround(s): None.

i2404 — *MBOX: Race condition in mailbox registers resulting in events miss*

Details:

Potential race condition in capture registers resulting in events getting lost while other events in the same register are being cleared by writing to the register. Following registers are impacted by this issue:

MSS_CTRL: *_MBOX_READ_REQ

MSS_CTRL: *_MBOX_READ_DONE

Workaround(s):

Read the status(READ DONE / READ_DONE_REQ) of the other processor to check any interrupt is in flight before setting up the trigger (WRITE DONE /READ ACK) event.

(OR)

Re-trigger the (WRITE DONE /READ ACK) event if the status (READ DONE / READ_DONE_REQ) is not received within the given time.

i2405 — *CONTROLSS: Race condition OUTPUT_XBAR and PWM_XBAR resulting in event miss*

Details:

Potential race condition in capture registers resulting in events getting lost while other events in the same register are being cleared by writing to the register. Following registers are impacted by this issue:

C2K_PWMXBAR:PWMXBAR_STATUS

C2K_OUTPUTXBAR:OUTPUTXBAR_STATUS

Workaround(s):

WA -1 (For event widths > 50ns):

By default, level events (width >50ns) will be captured in “STATUS” Register, while performing “Clr Flag”, if any new event from hardware is asserted at the same time, it will be missed in FLAG Register, However, STATUS register does capture such events missed in FLAG register. After completing “Clr FLAG”, reading the “STATUS” register allows to capture/process any missed event based on “STATUS” read.

WA-1: ISR Sequence:

Read FLAG Event[x]

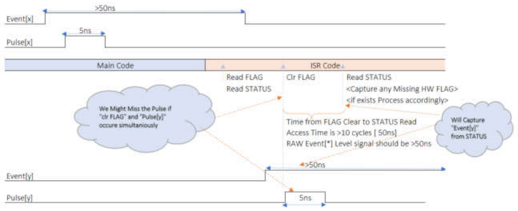
Read STATUS, All events

Clr FLAG, Event[x]

Read STATUS, All events

Capture any missing HW event FLAG

If exists, process accordingly


>50ns
Event[x]
5ns
Pulse[x]
Main Code
ISR Code
Read FLAG
Read STATUS
Clr FLAG
Read STATUS
<Capture any Missing HW FLAG>
<if exists Process accordingly>
We Might Miss the Pulse if "clr FLAG" and "Pulse[y]" occure simultaniously
Time from FLAG Clear to STATUS Read
Access Time is >10 cycles [50ns]
RAW Event[*] Level signal should be >50ns
Will Capture "Event[y]" from STATUS
>50ns
Event[y]
Pulse[y]
5ns

WA -2 (For any event widths):

Enable OUTPUTXBAR with the same events in the ISR and then “Clr PWMXBAR FLAG”.

Any missed Hardware event during the same window will be captured in OUTPUTXBAR FLAG”. Read the OUTPUTXBAR FLAG and process accordingly

i2405 (continued) *CONTROLSS: Race condition OUTPUT_XBAR and PWM_XBAR resulting in event miss*

“Clr OUTPUTXBAR FLAG” followed by disable of OUTPUTXBAR in the ISR.

WA-2: ISR Sequence:

Read FLAG Event[x]

Read STATUS, All events

Enable OutPutxBAR

- Map Same Events

Clr FLAG, Event[x] PWMXBAR

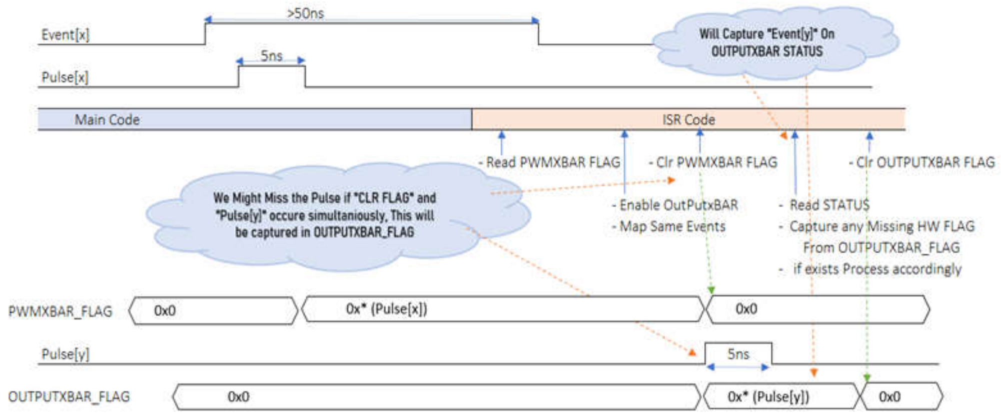
Read STATUS

- Capture any Missing HW event FLAG

from OUTPUTXBAR_FLAG

if exists Process accordingly

- Clr FLAG, Event[y] OutputXBAR



Trademarks

All trademarks are the property of their respective owners.

4 Revision History

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

Changes from March 1, 2023 to September 30, 2023 (from Revision C (March 2023) to Revision D (September 2023)) **Page**

- Updated Title.......... 1
- Added Advisory i2374; PBIST fails if clock frequency of R5SS_CORE_CLK is not same as R5FSS_CLK_SELECTED frequency.......... 12
- Added Advisory i2375; SDFM module event flags (SDIFLG.FLTx_FLG_CEVTx) do not get set again if the comparator event is still active and digital filter path (using SDCOMPxCTL.CEVTxDIGFILTSEL) is being selected.......... 13
- Added Advisory i2386: CRC: CRC 8-bit data width and CRC8-SAE-J1850 and CRC8-H2F possible use in CAN module is not supported.......... 14
- Added Advisory i2392: Race condition in mem-init capture registers resulting in events miss.......... 14
- Added Advisory i2394: Race condition in interrupt and error aggregator capture registers resulting in events miss.......... 14
- Added Advisory i2395: DCC bug description added.......... 15
- Added Advisory i2401: CPSW: Host Timestamps Cause CPSW Port to Lock up.......... 15
- Added Advisory i2402: CPSW: Ethernet to Host Checksum Offload does not work.......... 15
- Added Advisory i2403: M4 ROM: SBL redundant boot image feature not supported on HSSE devices 15
- Added Advisory i2404: MBOX: Race condition in mailbox registers resulting in events miss.......... 16
- Added Advisory i2405: CONTROLSS: Race condition OUTPUT_XBAR and PWM_XBAR resulting in event miss.......... 16

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