Errata

Errata AM263Px Sitara™ Microcontroller Silicon Revision 1.0

TEXAS INSTRUMENTS

ABSTRACT

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

Table of Contents

1 Usage Notes and Advisories Matrices........2
2 Silicon Revision *1.0* Usage Notes and Advisories........3
2.1 Silicon Revision *1.0* Usage Notes........3
2.2 Silicon Revision *1.0* Advisories........3
3 Trademarks........10
4 Revision History........10

1 Usage Notes and Advisories Matrices

Table 1-1 lists all usage notes and the applicable silicon revision(s). Table 1-2 lists all advisories, modules affected, and the applicable silicon revision(s).

Table 1-1. Usage Notes Matrix

Module	DESCRIPTION	SILICON REVISIONS AFFECTED
		AM263Px
		1.0
CLOCKS	i2324 — No synchronizer present between GCM and GCD status signals	YES

Table 1-2. Advisories Matrix

MODULE	DESCRIPTION	SILICON REVISIONS AFFECTED
		AM263Px
		1.0
CONTROLSS	i2352 — CONTROLSS-SDFM: Dynamically Changing Threshold Settings (LLT, HLT), Filter Type, or COSR Settings Will Trigger Spurious Comparator Events	YES
CONTROLSS	i2353 — CONTROLSS-SDFM: Dynamically Changing Data Filter Settings (Such as Filter Type or DOSR) Will Trigger Spurious Data Acknowledge Events	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
CONTROLSS	i2356 — CONTROLSS-ADC: Interrupts may Stop if INTxCONT (Continue-to-Interrupt Mode) is not Set	YES
CONTROLSS	i2357 — CONTROLSS-ePWM: An ePWM Glitch can Occur if a Trip Remains Active at the End of the Blanking Window	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
CONTROLSS	i2359 — CONTROLSS-CMPSS:Prescaler counter behavior different from spec when DACSOURCE is made 0 or reconfigured as 1	YES
CPSW	i2345 — CPSW: Ethernet Packet corruption occurs if CPDMA fetches a packet which spans across memory banks	YES
UART	i2310 — USART: Erroneous triggering of timeout interrupt	YES
UART	i2311 — USART: Spurious DMA Interrupts	YES

2 Silicon Revision *1.0* Usage Notes and Advisories

This section lists the usage notes and advisories for this silicon revision.

2.1 Silicon Revision *1.0* Usage Notes

This section lists all the usage notes that are applicable to silicon revision *1.0* [and earlier silicon revisions].

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.

2.2 Silicon Revision *1.0* Advisories

The following advisories are known design exceptions to functional specifications. Advisories are numbered in the order in which they were added to this document. Some advisory numbers may be removed in future revisions of this document because the design exception was fixed or documented in the device-specific data manual or technical reference manual. When items are deleted, the remaining advisory numbers are not re-sequenced.

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

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.

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).

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.

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.

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

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.

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:

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

```
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.

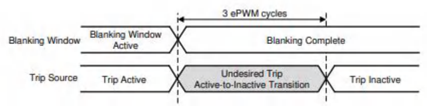
```
AdcaRegs.ADCINTFLGCLR.bit.ADCINT1 = 1; //clear INT1 flag
if(1 == AdcaRegs.ADCINTOVF.bit.ADCINT1) //ADCINT overflow
{
AdcaRegs.ADCINTFLGCLR.bit.ADCINT1 = 1; //clear INT1 again
// If the ADCINTOVF condition will be ignored by the application
// then clear the flag here by writing 1 to ADCINTOVFCLR.
// 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***

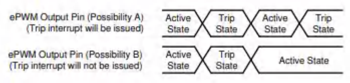
Details:

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.


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 behavior 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, 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.

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:

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:

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

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

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

Details:

The CPSW offers two mechanisms for communicating packet ingress timestamp information to the host.

The first mechanism is via the CPTS Event FIFO which records timestamps when triggered by certain events. One such event is the reception of an Ethernet packet with a specified EtherType field. Most commonly this is used to capture ingress timestamps for PTP packets. With this mechanism the host must read the timestamp (from the CPTS FIFO) separately from the packet payload which is delivered via DMA. This mode is supported and is not affected by this errata.

The second mechanism is to enable receive timestamps for all packets, not just PTP packets. With this mechanism the timestamp is delivered alongside the packet payload via DMA. This second mechanism is the subject of this errata.

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 is corrupted due to EMI or any other corruption mechanism a timestamp request may 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):

Ethernet to host timestamps must be disabled.

CPTS Event FIFO timestamping can be used instead of CPTS host timestamps.

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.

i2404 (continued) ***MBOX: Race condition in mailbox registers resulting in events miss***

(OR)

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

3 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.

DATE	REVISION	NOTES
November 2023	*	Initial Release