Errata

AM64x/AM243x Processor Silicon Revision 1.0

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

Copyright © 2021 Texas Instruments Incorporated

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

ID	DESCRIPTION	SILICON REVISIONS AFFECTED
		AM64x/AM243x 1.0

Table 1-2. Advisories Matrix

MODULE	DESCRIPTION	SILICON REVISIONS AFFECTED
		AM64x/AM243x 1.0
CBASS	i2207 — CBASS: Command Arbitration Blocking	YES
CPSW	i2184 — CPSW: IET express traffic policing issue	YES
	i2185 — CPSW: Policer color marking issue	YES
	i2208 — CPSW: ALE IET Express Packet Drops	YES
DDR	i2232 — DDR: Controller postpones more than allowed refreshes after frequency change	YES
	i2244 — DDR: Valid stop value must be defined for write DQ VREF training	YES
DMSC	i2245 — DMSC: Firewall Region requires specific configuration	YES
ECC_AGGR	i2049 — ECC_AGGR: Potential IP Clockstop/Reset Sequence Hang due to Pending ECC Aggregator Interrupts	YES
Internal Diagnostic Modules	i2103 — Incorrect Reporting of ECC_GRP, ECC_BIT and ECC_TYPE Information for Functional Safety Errors	YES
Interrupt Aggregator	i2196 — IA: Potential deadlock scenarios in IA	YES
JTAG	i2228 — JTAG: TAP used by Debuggers may be inaccessible if TRSTn device pin is never asserted	YES
OSPI	i2189 — OSPI: Controller PHY Tuning Algorithm	YES
PCIe	i2236 — PCIe: SERDES output reference clock cannot be used	YES
RAT	i2062 — RAT: Error Interrupt Triggered Even When Error Logging Disable Is Set	YES

1.1 Devices Supported

This document supports the following devices:

- AM64x
- AM243x

Reference documents for the supported devices are:

- AM64x/AM243x Processors Technical Reference Manual (SPRUIM2)
- AM64x Processors Datasheet (SPRSP56)
- AM243x Processors Datasheet (SPRSP65)

Copyright © 2021 Texas Instruments Incorporated

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

No known usage notes for this silicon revision.

2.2 Silicon Revision 1.0 Advisories

i2049 ***ECC_AGGR: Potential IP Clockstop/Reset Sequence Hang due to Pending ECC Aggregator Interrupts***

Details:

The ECC Aggregator module is used to aggregate safety error occurrences (which are rare) and generate interrupts to notify software. The ECC Aggregator provides software control over the enabling/disabling and clearing of safety errors interrupts.

When software is performing a clockstop/reset sequence on an IP, the sequence can potentially not complete because the IP's associated ECC Aggregator instance is not idle. The ECC Aggregator idle status is dependent upon any pending safety error interrupts either enabled or disabled, which have not been cleared by software. As a result, the IP's clockstop/reset sequence may never complete (hang) if there are any pending safety errors interrupts that remain uncleared.

Workaround(s):

General Note:

Clockstopping the ECC Aggregator is not supported in functional safety use-cases.

Software should use the following workaround for non-functional safety use-cases:

1. Enable all ECC Aggregator interrupts for the IP
2. Service and clear all Pending interrupts
3. Step 3:
   a. Disable all interrupt sources to the ECC Aggregator, followed by performing Clockstop/reset sequence.
   b. Perform Clockstop/reset sequence, while continuing to service/clear pending interrupts.

Due to interrupts being external stimuli, software has two options for step 3:

1. Disable all interrupt sources (EDC CTRL checkers) that can generate pending ECC_AGGR interrupts prior to performing the clockstop/reset sequence
2. Continue to service/clear pending interrupts that occur while performing the clkstop/reset sequence. The sequence would proceed when all interrupts are cleared.

Software in general may need to detect pending interrupts that continuously fire during this entire sequence (ex. in the case of a stuck-at fault scenario), and disable their associated EDC CTRL safety checkers to allow the clockstop/reset sequence to progress towards completion.

i2062 ***RAT: Error Interrupt Triggered Even When Error Logging Disable Is Set***

Details:

If the RAT error logging is programmed to disable logging and enable interrupts, then an error will incorrectly trigger an interrupt but the error log registers will correctly not be updated. The error interrupt should not have been generated.

Workaround(s):

If the RAT error logging is disabled, then the error interrupt should also be disabled by software.

i2103 ***Internal Diagnostics Modules: Incorrect Reporting of ECC_GRP, ECC_BIT and ECC_TYPE Information for Functional Safety Errors***

Copyright © 2021 Texas Instruments Incorporated

Details: For functional safety errors, the logged information - ECC_GRP, ECC_BIT, and ECC_TYPE in the Error Status Registers may be incorrect for certain safety checkers. This only applies to safety checkers that map to ECC_GRP = 0,15,31,47,63...(N*16-1). In the case for the DDR Bridge/Controller, the issue only applies to the safety checkers where ECC_GRP = 0,31,63...(N*32-1).

This issue affects all Internal Diagnostics Module instances and their sub-banks.

Note: The detection and interrupt signaling of these safety errors is unaffected. Only the logging of the aforementioned fields of the Error Status Registers are affected.

Workaround(s): None. For these specific safety checkers, software is limited to knowing whether a correctable or uncorrectable error occurred and which Internal Diagnostics Module instance had the error (thus knowing the IP module), but not which exact safety checker encountered the error.

i2184 *CPSW: IET express traffic policing issue*

Details: This applies to CPSW9G, CPSW5G, and CPSW2G IET traffic.

In IET (Interspersed Express traffic), if a preempted packet was interrupted by an express packet, two things can occur:

1. If the express traffic is policed, the frame size for the preempted packets is applied to the express traffic policer. Assuming a policer was set up to rate schedule an express traffic stream, it would take a hit by the preempted packet size it interrupted. The preempted packet also takes on the express traffic policer status. As a result, preempted packets could get dropped along with other express traffic due to the express traffic policer.
2. If the express traffic was not policed, the interrupted preempt packet would not get its packet size applied to the preempted policer.

Workaround(s): Do not police IET express traffic.

i2185 *CPSW: Policer color marking issue*

Details: Only applies to CPSW9G and CPSW5G.

When packets from two different ports hit the same policer such that one port has a large packet and the other has a short packet and the short packet arrives just after the large packet starts, the short packet will stop the backlog counting, resulting in potentially flagging the next frame for this policer as yellow when it should have been green. Because the policer is normally set up to not drop yellow, it should not cause an issue. This is only true of packets that arrive on different ports that share the same policer index.

Workaround(s): Ensure policers are unique to ports.

i2208 *CPSW: ALE IET Express Packet Drops*

Details: The issue with ALE is due to CPSW frequency and IET operation with short express traffic and pre-empted packets that get pre-empted between 60-69 bytes on non-10G capable ports.

If an IET pre-emptible packet get interrupted at 60-69 bytes, the lookup will occur when the next chunk arrives. The CPSW only gives the ALE 64 bytes from the pre-emptible MAC.

As a result, a short express traffic lookup will start at the end of a 64 byte express traffic, but when the pre-empted queue continues, the pre-empted traffic will complete the

Copyright © 2021 Texas Instruments Incorporated

64 bytes and attempt a lookup for the pre-empt packet. But this lookup is less that 64 clocks from the express lookup start, so the express lookup will be aborted(express traffic dropped) and start the new lookup for the pre-empted traffic.

Rules to induce the issue:

1. You are in IET (Interspersed Express Traffic) mode on ports not capable of 5/10G operation
2. Remote express packets can be preempt packets as low as 60 bytes
3. Pre-empt packet traffic that is 128 bytes or more.
4. Express traffic that interrupts the pre-empt traffic between 60-69 bytes.
5. A short express traffic immediately followed by the continuation of the pre-empt traffic.
   a. Gap between express frame and pre-empt frame be its minimum.
6. The CPSW frequency is at its lowest capability for the speeds required.

Workaround(s): During IET negotiation, tell the remote to fragment at 128 bytes.

i2189 ***OSPI: Controller PHY Tuning Algorithm***

Details:

The OSPI controller uses a DQS signal to sample data when the PHY Module is enabled. However, there is an issue in the module which requires that this sample must occur within a window defined by the internal clock. Read operations are subject to external delays, which change with temperature. In order to guarantee valid reads at any temperature, a special tuning algorithm must be implemented which selects the most robust TX, RX, and Read Delay values.

Workaround(s): The workaround for this bug is described in detail in the application note spract2 (link: https://www.ti.com/lit/spract2). To sample data under some PVT conditions, it is necessary to increment the Read Delay field to shift the internal clock sampling window. This allows sampling of the data anywhere within the data eye. However, this has these side effects:

1. PHY Pipeline mode must be enabled for all read operations. Because PHY Pipeline mode must be disabled for writes, reads and writes must be handled separately.
2. Hardware polling of the busy bit is broken when the workaround is in place, so SW polling must be used instead. Writes must occur through DMA accesses, within page boundaries, to prevent interruption from either the host or the flash device. Software must poll the busy bit between page writes. Alternatively, writes can be performed in non-PHY mode with hardware polling enabled.
3. STIG reads must be padded with extra bytes, and the received data must be right-shifted.

i2196 ***IA: Potential deadlock scenarios in IA***

Details:

The interrupt Aggregator (IA) has one main function, which is to convert events arriving on the Event Transport Lane (ETL) bus, can convert them to interrupt status bits which are used to generate level interrupts. The block that performed this function in IA version 1.0 was called the status event block.

In addition to the status event block, there are two other main processing blocks; the multicast event block, and the counted event block. The multicast block really functions as an event splitter. For every event it takes in, it can generate two output events. The counted event block is used to convert high frequency events into a readable count. It counts input events and generates output events on count transitions to/from 0 to/from non-zero count values. Unlike the status event block, the multicast and counted event blocks generate output ETL events that are then mapped to other processing blocks.

An issue was found after design that could cause the IA to deadlock. The issue occurs when event “loops” occur between these three processing blocks. It is possible to create a

Copyright © 2021 Texas Instruments Incorporated

situation where a processing block can not output an event because the path is blocked, and since it can not output an event, it can not take any new input events. This inability to take input events prevents the output path from being able to unwind, and thus both paths remain blocked.

Workaround(s): Consider the conceptual block diagram of IA 1.0 below. Potential loops are avoided by adopting the policy of not allowing the counted event block to send events to the multicast block. This method was chosen because it is more common to split an event first, and then count one while sending the other elsewhere. With this path blocked by convention, it is not possible for a single event to visit any block more than once and thus not possible for paths to become blocked so long as the outputs remain unblocked.

IA version 1.1 introduced two additional changes to the architecture. First, there is a new processing block for “unmapped” events. These are events that are sent to a fixed destination (the IA) instead of being programmable at the source IP. Being fixed, the are called “unmapped”. Once in the IA, they are mapped to a traditional ETL event destination. The block that performs this mapping is called unmapped event block. The second change to the IA for 1.1 is that now, the multicast and counted event blocks (as well as the new unmapped event block) can directly set VINT status bits, and they do not need to forward output ETL events to the status event block.

Now consider the diagram of IA 1.1 below. Note that the same policy applies to avoid deadlock. In addition, the paths shown as dotted lines (sending ETL events from the multicast or counted blocks to the status event block) is now discouraged. These paths still function as before for backward compatibility, but are rendered obsolete as all blocks are now able to directly set status event bits.

By following the conventions outlined here, the system is safe from looping hazards that can create a deadlock scenario.

i2207 ***CBASS: Command Arbitration Blocking***

Details: When the interconnect arbitrates commands from multiple sources, the higher priority request always takes precedence. Requests that are at the same priority level are arbitrated in round-robin fashion. The issue is after the higher priority request goes idle and there are two or more pending requests that are at the same priority level, the hardware selects one of them arbitrarily. A potential issue may arise when software polls from multiple sources to the same endpoint: after servicing the higher priority source, the hardware may repeatedly select the same lower priority source for access. That means other same-lower priority requests may be blocked for a long time, and in the worst case if there are dependencies between the polling sequences, the software may run into a livelock state.

This issue only affects certain interconnect where in one switch module there are at least three sources that can access the same target simultaneously. Also note that when all requests are at the same priority level, the issue does not apply.

Workaround(s): When multiple sources are simultaneously polling from the same endpoint, and there is expected dependency based on the read data, ensure that all sources are sending the read commands at the same priority level. The source that breaks the dependency should be at equal or higher priority than other dependent sources.

i2228 ***JTAG: TAP used by Debuggers may be inaccessible if TRSTn device pin is never asserted***

Details If TRSTn is never observed LOW, access to the embedded Debugger scan chains might be blocked by uninitialized logic. JTAG bypass and Boundary Scan functionality is not affected.

Copyright © 2021 Texas Instruments Incorporated

Workaround

Prior to connecting a Debugger, ensure that the TRSTn pin is asserted LOW for 100ns and subsequently de-asserted HIGH at-least one time after device power on.

Copyright © 2021 Texas Instruments Incorporated

i2232 ***DDR: Controller postpones more than allowed refreshes after frequency change***

Details

When dynamically switching from a higher to lower clock frequency, the rolling window counters that control the postponing of refresh commands are not loaded correctly to scale to the lower clock frequency. This will result in controller postponing more refresh commands than allowed by the DRAM specification, thus violating refresh requirement for the DRAM.

Workaround

Workaround 1:Disable dynamic frequency change by programing DFS_ENABLE = 0

Workaround 2:If switching frequency, program the register field values based on the pseudo code listed below.Note that the controller requires AREF_*_THRESHOLD values to be programmed before triggering initialization. Their values cannot be changed during mission mode after initialization . Therefore, the value of these parameters must be the lowest of all values needed for every frequency change transition planned to be used.

```
if (old_freq/new_freq >= 7){
    if (PBR_EN==1) { // Per-bank refresh is enabled
        AREF_HIGH_THRESHOLD = 19
        AREF_NORM_THRESHOLD = 18
        AREF_PBR_CONT_EN_THRESHOLD = 17
        AREF_CMD_MAX_PER_TREF = 8
    }
    else { // Per-bank refresh is disabled
        AREF_HIGH_THRESHOLD = 18
        AREF_NORM_THRESHOLD = 17
        // AREF_PBR_CONT_EN_THRESHOLD <=== don't care, PBR not enabled
        AREF_CMD_MAX_PER_TREF = 8
    }
}
else {
    AREF_HIGH_THRESHOLD = 21
    AREF_NORM_THRESHOLD //<=== keep AREF_NORM_THRESHOLD < AREF_HIGH_THRESHOLD
    AREF_CMD_MAX_PER_TREF = 8
    if (PBR_EN==1) { // Per-bank refresh is enabled
    //keep AREF_PBR_CONT_EN_THRESHOLD<AREF_NORM_THRESHOLD<AREF_HIGH_THRESHOLD
        AREF_PBR_CONT_EN_THRESHOLD
    }
}
```


Copyright © 2021 Texas Instruments Incorporated

i2244 ***DDR: Valid stop value must be defined for write DQ VREF training***

Details The DDR PHY uses start, stop, and step-size values for write DQ VREF training. If the stop value is not equal to the start value + a multiple of the step-size, then the final VREF setting can go beyond the maximum VREF range, causing the training to hang.

Workaround Program the stop value as follows:

PI_WDQLVL_VREF_INITIAL_STOP = (multiple of PI_WDQLVL_VREF_INITIAL_STEPSIZE) + PI_WDQLVL_VREF_INITIAL_START

This workaround is implemented in the DDR Subsystem Register Configuration Tool v0.03.00 or later. See https://dev.ti.com/sysconfig for more details.

i2236 ***PCIe: SERDES output reference clock cannot be used***

Details:

The PCIe Reference Clock Output of the SERDES does not comply with the certain PCI-SIG specifications, which may cause issues for external PCIe components receiving and using the reference clock. Therefore, SERDES reference clock (signals SERDES0_REFCLK0P and SERDES0_REFCLK0N) cannot be used to output a PCIe reference clock.

Workaround(s): None. This issue is still under investigation.

Copyright © 2021 Texas Instruments Incorporated

i2245	***DMSC: Firewall Region requires specific configuration***
Details	The ECC Aggregator inside DMSC (DMSC0_ECC_AGGR) has an endpoint firewall which is used to protect this region. By default, this firewall blocks all the transactions except from the M3 core inside DMSC.
Workaround	If another processor or endpoint needs to access DMSC0_ECC_AGGR region, software shall configure the firewall region with starting address 0x0 and end address 0xFFFF_FFFF, using the CBASS_FW_REGION_i_START_ADDRESS and END_ADDRESS registers associated with the DMSC0_ECC_AGGR region. This is the only allowable address configuration for this region.

Copyright © 2021 Texas Instruments Incorporated

Trademarks

All trademarks are the property of their respective owners.

Copyright © 2021 Texas Instruments Incorporated

Revision History

Changes from May 30, 2021 to June 30, 2021 (from Revision A (May 2021) to Revision B (June 2021)) Page

Copyright © 2021 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, 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 (https:www.ti.com/legal/termsofsale.html) or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2021, Texas Instruments Incorporated