AM1705
ARM Microprocessor
Silicon Revisions 3.0, 2.1, and 2.0

Silicon Errata

Texas Instruments

Literature Number: SPRZ311E
March 2010–Revised June 2014

Contents

Copyright © 2010–2014, Texas Instruments Incorporated

Silicon Errata

SPRZ311E–March 2010–Revised June 2014

AM1705
Silicon Revisions 3.0, 2.1, and 2.0

1 Introduction

This document describes the known exceptions to the functional specifications for the AM1705 ARM Microprocessor devices (for example, AM1705). For more detailed information on these devices, see the device-specific data manual: *AM1705 ARM Microprocessor* data manual (literature number SPRS657).

For additional peripheral information, see the latest version of the *AM1705 ARM Microprocessor* Technical Reference Manual (Literature Number: SPRUH93).

The advisory numbers in this document may not be sequential. Some advisory numbers may be moved to the next revision and others may have been removed and documented in the device-specific data manual or peripheral user's guide. When items are moved or deleted, the remaining numbers remain the same and are not resequenced.

This document also contains Usage Notes. Usage Notes highlight and describe particular situations where the device's behavior may not match presumed or documented behavior. This may include behaviors that affect device performance or functional correctness. These notes will be incorporated into future documentation updates for the device (such as the device-specific data manual), and the behaviors they describe will not be altered in future device revisions.

1.1 Device and Development Support Tool Nomenclature

Device development evolutionary flow:

X Experimental device that is not necessarily representative of the final device's electrical specifications

P Final silicon die that conforms to the device's electrical specifications but has not completed quality and reliability verification

NULL Fully-qualified production device

Support tool development evolutionary flow:

TMDX Development-support product that has not yet completed Texas Instruments internal qualification testing

TMDS Fully-qualified development-support product

X and P devices and TMDX development-support tools are shipped against the following disclaimer:

> "Developmental product is intended for internal evaluation purposes."

TMS devices and TMDS development-support tools have been characterized fully, and the quality and reliability of the device have been demonstrated fully. TI's standard warranty applies.

Predictions show that prototype devices (X or P) have a greater failure rate than the standard production devices. Texas Instruments recommends that these devices not be used in any production system because their expected end-use failure rate still is undefined. Only qualified production devices are to be used.

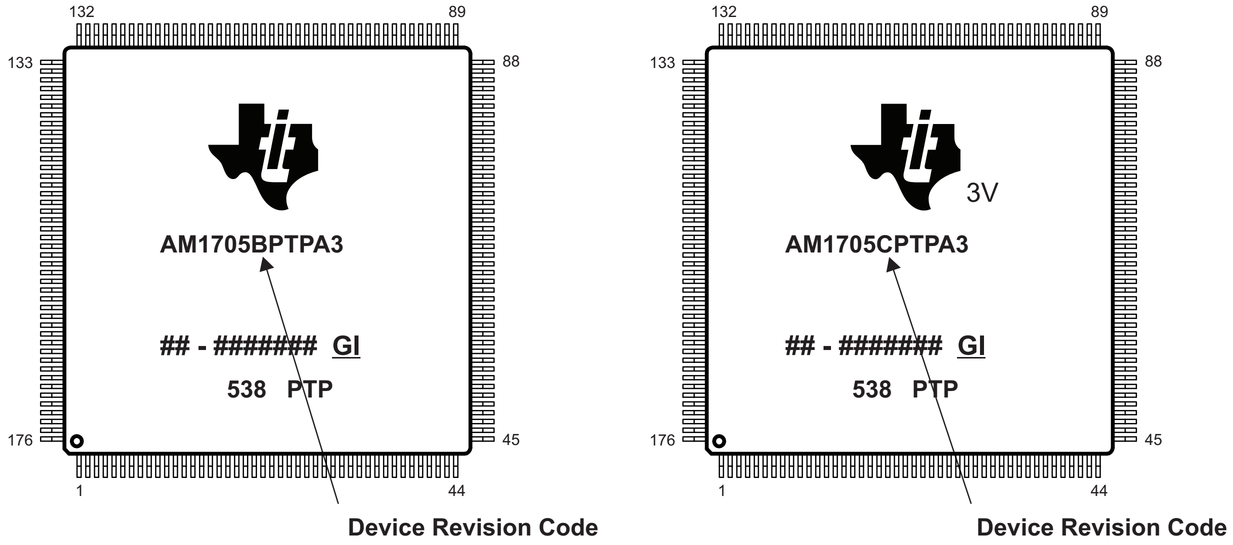
Copyright © 2010–2014, Texas Instruments Incorporated

1.2 Package Symbolization and Revision Identification

Example, Device Revision Codes for AM1705 (PTP Package) shows an example of the AM1705 processor package symbolization. The device revision code can be identified by the markings on the top of the package; for example, the "C" between the device number and the package identifier indicates the device revision code (that is, Silicon Revision 2.1) A "D" between the device number and the package identifier would indicate the device revision code for Silicon Revision 3.0.

Table 1 lists the device revision codes for the AM1705 processor, as well as shows the relationship between the device revision codes and silicon revisions.

For more detailed information on device nomenclature breakdowns, see the "*Device and Documentation Support*" section of the device-specific data manual.



Example, Device Revision Codes for AM1705 (PTP Package)

Table 1. AM1705 Device Revision Codes

DEVICE REVISION CODE (xx)	SILICON REVISION	COMMENTS
D	3.0	All Silicon Revision 3.0 parts support a minimum operating DVDD (I/O) voltage of 3.0V; therefore, no additional marking is necessary.
C	2.1	Silicon Revision 2.1 parts that have a minimum operating DVDD (I/O) voltage of 3.0V can be identified by the 3V marking next to the TI logo on the package. See Example, Device Revision Codes for AM1705 (PTP Package) for details.
B	2.0	Silicon Revision 2.0 parts that have a minimum operating DVDD (I/O) voltage of 3.0V can be identified by the 3V marking next to the TI logo on the package. See Example, Device Revision Codes for AM1705 (PTP Package) for details.

Copyright © 2010–2014, Texas Instruments Incorporated

2 Silicon Revision 3.0 Usage Notes and Known Design Exceptions to Functional Specifications

This section describes the usage notes and advisories that apply to silicon revision 3.0 of the device.

2.1 Usage Notes for Silicon Revision 3.0

Usage notes highlight and describe particular situations where the device's behavior may not match presumed or documented behavior. This may include behaviors that affect device performance or functional correctness. These usage notes will be incorporated into future documentation updates for the device (such as the device-specific data sheet), and the behaviors they describe will not be altered in future silicon revisions.

2.1.1 McASP: Inactive Slot Usage Note

On all the silicon revisions, McASP underflow (underruns) can occurs if any of the McASP serialzer is configured as a transmit serializer with any of the time slot-n in XTDM register field is set to inactive and EDMA is used to do the transfers. This is independent of whether the EDMA/CPU transfers are through peripheral configuration bus or the DMA port.

If the EDMA is used to service the McASP with any of the time slot-n in XTDM register field is set to inactive, EDMA may not be triggered upon enabling serializer and hence data transfer will be stalled or if data transfer has begun with some random value and goes to underrun errors, that breaks the data transfer operation.

To ensure proper McASP (transmit) data transfer either through config or EDMA port, ensure the time slot-n in XTDM register field is set to active. For example, if a serializer is configured for transmit operation with a 5-slot TDM frame in which it is only required to transmit data in slots 0 to 2, but make sure all five slots (0 to 4) should be configured as active. The EDMA configuration and user application should account for the transfer of irrelevant data to the McASP for slots 3 and 4.

2.1.2 USB0: Generic RNDIS Usage Note

On all silicon revisions, when using Generic RNDIS mode, the user should ensure that the DMA configuration has completed prior to the host starting a transfer. This condition is sometimes violated when performing back-to-back data transfers (not transactions). If a new transfer is scheduled by a host while the device is working on the previous transfer and the data transfer size for the new transfer is different than the previous transfer data size, then there exists a contention between the two transfer sizes creating undesired behavior resulting with a DMA lock up. A case in point where this violation could happen is demonstrated by the example below.

A user configures the DMA in Generic RNDIS mode expecting a data size of 512 bytes or less from a host. The host sends 512 bytes or less of data to the device. While the device is in the process of working on the received data to figure out the size of the next data transfer, the host starts a new data transfer addressing the same endpoint. Since the endpoint FIFO is empty, the device accepts the data and the DMA starts to transfer the received data from the receive FIFO to memory. At the same time, the application on the device side finishes and figures out the next transfer data size (using the data received from previous transfer) and reconfigures the Generic DMA Size register for the second transfer. If the second transfer size is different from the first transfer size, the contention happens at this point. The host has already started the second transfer prior to the device re-configuring the DMA parameters. The application on the device side, updates the DMA size register content for the second transfer while the DMA is in the middle of the second transfer using the DMA size register content of the first transfer. This effectively results with altering the DMA size register content while the DMA is in the middle of a transfer. Changing DMA parameters while in the middle of a transfer is not allowed and when done it will create undesirable outcomes.

Workaround: This is not a bug and for this reason, there exists no workaround. This is a caution for the user to be aware of this issue and hence to ensure that this scenario is avoided. If there exists an idle time in between the two back-to-back transfers, this issue will not exist. When expecting a back-to-back transfer where RNDIS mode can not be used, the user needs to use TRANSPARENT mode. When using TRANSPARENT mode, the application will be receiving more interrupts, that is, interrupts will be generated on each USB packets as opposed to receiving a single interrupt on the completion of a transfer.

Copyright © 2010–2014, Texas Instruments Incorporated

2.1.3 USB0: Isochronous Interrupt Loading Usage Note

On all silicon revisions, when the USB Controller Endpoint is enabled to handle Isochronous type of transfer, the controller supports a single configuration for interrupt generation, which is for interrupts to be generated for every ISO packet received or sent, that is, transfer size is equal to packet size. The option of generating interrupt on multiple ISO packets received or sent is not available. Since ISO transfer can be scheduled to happen on every micro-frame or frame, the number of interrupts generated could overwhelm the system. This is not a problem as long as there is enough CPU power available to handle all interrupts. However, some applications may be running low on available CPU time and may desire to service/process multiple ISO packets at a time. The option for handling ISO interrupts in a batch is not available. The user should ensure that enough CPU power is available to handle all ISO interrupts in order to avoid missing interrupts resulting with missing ISO packets.

2.1.4 Emulation Suspend Usage Note

Device peripherals often include an emulation suspend function that gracefully halts peripheral activity. This function is activated when the target CPU is halted through emulator debug. While halted, the control and status registers for the module can be viewed and manipulated for debug purposes.

The peripheral emulation suspend functions on this device are disabled by default. In order to suspend a peripheral, the application must first enable the suspend function on an individual peripheral basis via the SYSCFG_SUSPSRC register.

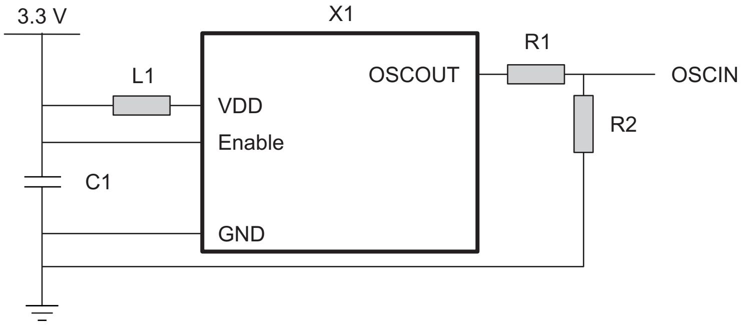
2.1.5 Clock Input Buffers Updated for Noise Immunity

For silicon revision 2.1 and later, the I/O buffers for all McASP clock inputs have been improved to provide more robust noise immunity than in previous silicon revisions. Timing specifications are not affected as a result. In particular, the specification for maximum input transition time remains the same; the improvements simply provide more margin to the existing specification.

2.1.6 System-Level ESD Immunity Usage Note

On all silicon revisions, certain design elements make this device susceptible to radiated noise during an ESD strike, as described in the standard IEC 61000-4-2. Exposure to the electrical noise caused by the ESD can cause soft device failures due to noise coupling on the system clock (OSCIN). ESD events within the IEC spec range do not cause permanent device damage and full functionality is recoverable with a device reset. The sensitivity to this noise issue is primarily due to the 1.2V oscillator/clock input implemented on this device. The low voltage range, coupled with slow rise and fall times, provides a lower noise margin than other TI devices with higher voltage internal oscillators (for example, 1.8V or 3.3V oscillators).

If ESD robustness is a concern, it is strongly recommended to avoid using the internal oscillator as a clock source. An external 3.3V clock source with a resistor voltage divider as in Figure 1 can be used to externally generate the required 1.2V input clock. By using an external clock input with fast rise/fall times (less than 5 ns), the noise margin improves significantly, increasing ESD noise resistance.



Legend: L1 = ferrite bead; C1 = 0.1 uF; R1 = 165 ohm / 5%; R2 = 100 ohm / 5%

Figure 1. External 3.3V Clock Source

Copyright © 2010–2014, Texas Instruments Incorporated

In addition to using an external clock source, several other board and software recommendations specific to this device can improve system-level ESD immunity:

- The OSCIN and OSCVSS (and OSCOUT, if used) should be routed as short as possible to reduce their ability to pick up EMI noise.
- Route the OSCIN signal on inner board layers where it is shield by power and ground planes.
- The processor should be provided as much power supply decoupling as is practical and placed as close to the processor as possible.
- Implement the PLL filtering circuits shown in the device datasheet.

These recommendations are in addition to standard methods for increasing system ESD immunity, such as using shielding enclosures, proper grounding and PCB stackup, and ESD protection circuitry.

2.2 Silicon Revision 3.0 Known Design Exceptions to Functional Specifications

The advisories are not enumerated in sequential order and hence some numbers may not appear in the document

Table 2. Silicon Revision 3.0 Advisory List

Title **Page**

Copyright © 2010–2014, Texas Instruments Incorporated

Advisory 3.0.6 *USB0: Extraneous RESET Interrupt*

Revision(s) Affected 3.0 and earlier

Details When the USB controller is operating as a device and an attached host resets the device after the completion of the “Device Attached” state by driving both differential data lines low, the USB controller operating as a device could receive multiple RESET interrupts for the single RESET signaling invoked by the host. The multiple interrupt generation only happens for the duration of the RESET signaling on the bus. RESET Interrupt is not generated before or after the completion of RESET.

Workaround(s) Software must service every USB RESET interrupt received. Software should not proceed on performing any other task, like initialization, until RESET duration has come to completion. The POWER[RESET] bit field will be cleared by the USB Controller when RESET Signaling on the bus is removed by the Host. The USB Controller clearing the POWER[RESET] bit field should be used by software as an indication for the completion of RESET signaling.

Copyright © 2010–2014, Texas Instruments Incorporated

Advisory 3.0.13 *EMIFA: Asynchronous Memory Timeout Error Persistence*

Revision(s) Affected 3.0 and earlier

Details

In Extended Wait mode, during a read access to an asynchronous memory, if the WAIT input does not go inactive within maximum extended wait cycles programmed in the Async Wait Cycle Config register, the EMIFA will report a time-out error. The data returned for this access will be all zeros. If this access is followed by a read to the EMIFA's memory-mapped register (MMR) space, the EMIFA will still report a time-out error but with the correct data for the MMR read. The EMIFA will hold the time-out error until another asynchronous access without a time-out error or an SDRAM access is performed.

This issue is only applicable if all of the following are true:

- The EMIFA is used for asynchronous memory accesses in Extended Wait mode.
- There is a potential for a time-out error to occur, that is, the asynchronous memory will not de-assert the WAIT input.
- If asynchronous memory read with time-out error is followed by an MMR read.

Workaround(s)

If a time-out occurs, perform any of the following:

- A dummy read to another asynchronous memory chip select that is not configured to be in Extended Wait mode.
- A dummy read to the same asynchronous memory chip select after disabling the Extended Wait mode on that chip select.
- A dummy read to SDRAM

Copyright © 2010–2014, Texas Instruments Incorporated

Advisory 3.0.15 *Potential USB2.0 Soft Reset Timing Violation*

Revision(s) Affected 3.0 and earlier

Details

When a soft reset is invoked by setting the RESET bit of the USB CTRLR register (CTRLR[RESET] = 1), the internal reset timing requirements may be violated. Although this timing violation has not been observed in practice, the potential for a timing violation exists.

USB resets initiated by system-reset and power-on-reset are immune from the timing violation.

There is no plan to fix this issue in future silicon revisions because:

1. No functional problems have been observed to date
2. A software workaround has been developed to avoid the problem

Workaround(s)

The reset timing violation can be avoided by providing the modified soft reset activation sequence outlined below:

1. Enable the USB controller module clock
2. Perform a soft USB reset
3. Wait for the USB soft reset bit to clear
4. Disable the USB controller module clock
5. Configure the USB PHY parameters
6. Enable the PHY
7. Enable the USB controller module clock

Copyright © 2010–2014, Texas Instruments Incorporated

Advisory 3.0.17 *ARM Interrupt Controller Vector Size Register (VSR) Initialization*

Revision(s) Affected 3.0 and earlier

Details The VSR register in the ARM Interrupt Controller (AINTC) is not correctly initialized after reset. If this register is not explicitly configured, the AINTC will only allocate 1 byte per interrupt (instead of 4).

Workaround(s) The desired value (even if it is the default value) should be written to the VSR prior to using the interrupt controller.

Copyright © 2010–2014, Texas Instruments Incorporated

Advisory 3.0.18 *A Single CHIPINTn Interrupt Event Can Register Multiple Times in the AINTC*

Revision(s) Affected

3.0 and earlier

Details

Interrupts destined for the ARM CPU are managed by the ARM Interrupt Controller (AINTC). The AINTC detects, combines, and routes system interrupts to the two native ARM interrupt signals FIQ and IRQ. See the device for additional information about the AINTC.

The AINTC module expects all incoming interrupts to be pulse interrupts, however the [SYSCFG_CHIPSIG_]CHIPINTn interrupts are level interrupts. This mismatch in interrupt types will cause a single CHIPINTn interrupt event to register as multiple interrupt pulses in the AINTC. However, the AINTC does not have the capacity to count the number of interrupt pulses received per system interrupt – it only maintains interrupt flags. A system interrupt is flagged as active until its status is cleared by the user through the AINTC, regardless of the number of interrupts detected.

If the status flag for AINTC CHIPINTn is cleared while the CHIPINTn interrupt is still active, the AINTC will continue to detect CHIPINTn interrupts and its status flag will be set again. This additional setting of the AINTC CHIPINTn status flag is false.

Workaround(s)

Method 1

Do not execute the intended interrupt service routine code if the associated CHIPSIGn status flag is not set in the SYSCFG_CHIPSIG register. A cleared CHIPSIGn status flag indicates that the device is responding to a false interrupt. This method is easy to implement, but does not eliminate false interrupts.

```
/** Pseudo code only **/

    void CHIPINT0_ISR(void) {
        /* Exit immediately if CHIPSIG0 is not set */
        if( (SYSCFG->CHIPSIG & 0x1) == 0 ) {
            return;
        }

        /* Intended service routine code */
        SYSCFG->CHIPSIG_CLR = 0x1;
        printf(“Hello World!\n”);
    }
```

Method 2

Do not clear the AINTC CHIPINTn status flag until the CHIPSIGn status has been cleared. This method will eliminate false interrupts, but requires changes to the AINTC interrupt dispatch code. Changing the dispatch code may introduce undesired behavior in the application.

```
/** Pseudo code only **/

    /* Sequence that is susceptible to false CHIPINTn interrupts */
    void AINTC_ISR_DISPATCH_1(void) {
        Get_Interrupt_Information();

        /* CHIPINTn interrupts continue to be generated after */
        /* AINTC CHIPINTn flag is cleared.                    */
        Clear_AINTC_Interrupt_Flag();

        /* CHIPINTn interrupts are only stopped after ISR clears */
        /* the status flag.                                      */
        Branch_To_ISR();
    }
```

Copyright © 2010–2014, Texas Instruments Incorporated

```
/* Sequence that is not susceptible to false CHIPINTn interrupts */
   void AINTC_ISR_DISPATCH_2(void) {
       Get_Interrupt_Information();

       /* ISR will clear CHIPSIGn flag and discontinue CHIPINTn */
       /* interrupts to AINTC.                                   */
       Branch_To_ISR();

       /* Ok to clear AINTC CHIPINTn flag now.                   */
       Clear_AINTC_Interrupt_Flag();
   }
```

Copyright © 2010–2014, Texas Instruments Incorporated

Advisory 3.0.22 *USB 2.0 On-The-Go (OTG) Session Request Protocol (SRP) Is Not Supported*

Revision(s) Affected 3.0 and earlier

Details The USB 2.0 On-The-Go (OTG) Session Request Protocol (SRP) allows a USB-peripheral to request the USB-host to enable Vbus and start a session. On this device, the SRP protocol is not supported.

The OTG Host Negotiation Protocol (HNP), which allows USB-devices to swap roles between host and peripheral, is supported.

Workaround(s) None

Copyright © 2010–2014, Texas Instruments Incorporated

Advisory 3.0.24 *USB0 PLL Mean Frequency Can Drift Across Large Temperature Swings*

Revision(s) Affected 3.0 and earlier

Details Under conditions in which the device is subjected to large variations in operating temperatures, the USB0 PLL temperature compensation circuitry does not have enough margin to guarantee compensation for PLL drift across all temperature ranges.

As a result, the mean frequency generated by the USB0 (USB 2.0 OTG) PHY PLL will begin to drift (relative to the expected 480 Mbps) when the temperature of the device is subjected to large swing from the original temperature in which the USB0 PHY was most recently calibrated (initialized).

Once the onset of the PLL drift occurs, the mean frequency will continue to drift outside the expected frequency eventually resulting in failure of USB packet reception and/or transmission. This break in transmission will continue until the USB0 PHY is recalibrated during a USB0 PHY Reset.

If the device is not exposed to large variations in temperature relative to the temperature at which the USB0 PHY was most recently initialized, the temperature compensation circuitry is expected to provide the proper compensation to prevent the mean PLL frequency from losing lock and beginning to drift.

More specifically, this advisory is most applicable in applications where the device is expected to operate outside the commercial temperature space (0˚C-90˚C). TI has identified a point-to-point device temperature range of 0˚C-65˚C in which there is very high confidence in which the compensation circuitry will properly compensate for all variations in temperature provided that the USB0 PHY was most recently initialized (calibrated) within this same temperature range.

Operating outside the 0˚C-65˚C temperature range increases the susceptibility of the device to experience PLL drift, but does not mean that the application will always experience a failure in USB transmission.

Root Cause The Voltage Controlled Oscillator (VCO) Compensation circuitry local to the USB0 PHY was not designed with a large enough range to compensate for all variations in temperature across the specified operating range of the device.

How to Most Easily Reproduce the Issue: Reproduction of this issue can most easily be accomplished by the following steps:

1. Allowing the unit to soak in an ambient temperature of -35˚C until the device temperature reaches approximately the same temperature.
2. Power up the device and provide the necessarily software programming in order to invoke the USB Signal Quality Test Pattern.
3. Using a USB 2.0 Certified Test Platform, execute the USB signal quality test procedure across the following temperature set points. -35˚C, 0˚C, +35˚C, +70˚C. Record the measured mean frequency by the compliance software.

NOTE: The set points can be varied to obtain finer temperature resolution of when the PLL begins to drift a per platform basis. The above temperature profile is provided for reference.

Copyright © 2010–2014, Texas Instruments Incorporated

Workaround(s)

When a break in transmission is detected, USB0 traffic can be recovered by a software reset of the USB0 PHY. A PHY reset implies recalibration of the PHY PLL at the reset temperature. The system has not been observed to reliably recover on its own. A PHY reset also implies re-enumeration of all devices. There is no way to recalibrate the USB0 PHY without a re-enumeration.

In order to invoke the recovery mechanism (that is a USB0 PHY reset) one needs to determine when the issue is present. One such approach is to look for an absence of USB0 Core interrupts over a specified time window. This window should be optimized for the expected USB traffic based upon the application.

As an additional safeguard, an application can also intentionally schedule pre-determined USB PHY resets at specific temperature points if operation over a broad range is expected.

Here is an example of one way to power cycle the USB0 PHY via the Chip Configuration 2 Register in the System Configuration (SYSCFG) Module:

```
#define CFGCHIP2  *((volatile unsigned int *) 0x01C14184)
    #define USBPHY_PHYPDWN 0x00000200

Void phy_reset(void) {
        CFGCHIP2 |= USBPHY_PHYPDWN;        /* Power down the USB PHY */

        mdelay(1);                         /* Wait 500ms */

        CFGCHIP2 &= ~USBPHY_PHYPDWN;       /* Power up the USB PHY */
}
```

Copyright © 2010–2014, Texas Instruments Incorporated

Advisory 3.0.25 *USB0: CPU gets Stale Receive Data from the Data Buffer located in External Memory*

Revision(s) Affected 3.0 and earlier

Details When CPPI DMA completes a receive data transaction it posts a write to the Rx data buffer located in external memory, posts a write to update the descriptor located in external memory, and raises an interrupt to CPU. When the system load is high, the posted writes to DDR may not be complete before the CPU receives the interrupt. In this case, the CPU would fetch stale receive data from the Rx data buffer located in external memory.

Workaround(s) Initialize the datalength descriptor field to zero. CPPI DMA updates this field after the completion of an RX DMA operation with the actual number of bytes received. In the ISR (actually in a deferred call context), poll this field until it becomes a non-zero value to ensure data buffer has been updated with actual data. The descriptor buffer write is posted after the data buffer write, so waiting for the descriptor field to be updated ensures the data buffer has been updated. Since this workaround involves deferred procedure calls (whose schedule can be delayed depending on OS load), the latency sensitive application (like ISO Audio) might be affected by delay in notification to the application.

Copyright © 2010–2014, Texas Instruments Incorporated

Advisory 3.0.26 — *USB0: Early DMA Completion in DMA Receive Mode and More Than One Endpoint is Transferring Data*

Revision(s) Affected 3.0 and earlier

Details The erroneous short packet status can be detected on current endpoint and XDMA closes the Rx transfer in current endpoint. When more than one endpoint have been processed, if one of the endpoints has a short packet, then the short packet status is broadcasting to all endpoints.

This results in premature completion of a Rx descriptor in generic RNDIS CPPI DMA mode.

Workaround(s) The workaround involves monitoring transfer data size before and after transferring and reconfiguring data transfer size by software if the before and after size is different. Software must keep tracking every endpoint data transferring size. When DMA completion interrupt is received, software checks size difference. If the size is not equal, software requests the remaining data.

Copyright © 2010–2014, Texas Instruments Incorporated

Advisory 3.0.27 *USB0: DMA Hung up in Frequent Teardowns*

Revision(s) Affected 3.0 and earlier

Details Teardown receive DMA is not working perfectly. This happens when a teardown is initiated by software during the endpoint is still active. Frequent teardown results in XDMA hung up situation.

Workaround(s) Software should make sure that DMA does not get to an unknown state during teardown by disabling the DMAEN bit in the RXCSR register. After this the teardown procedure can be initiated. Software should also add 250 ms delay during teardown.

Copyright © 2010–2014, Texas Instruments Incorporated

3 Silicon Revision 2.1 Usage Notes and Known Design Exceptions to Functional Specifications

This section describes the usage notes and advisories that apply to silicon revision 2.1 of the *AM1705*.

3.1 Usage Notes for Silicon Revision 2.1

Usage notes highlight and describe particular situations where the device's behavior may not match presumed or documented behavior. This may include behaviors that affect device performance or functional correctness. These usage notes will be incorporated into future documentation updates for the device (such as the device-specific data sheet), and the behaviors they describe will not be altered in future silicon revisions.

Silicon revision 2.1 applicable usage notes have been found on a later silicon revision. For more details, see Section 2.1 *Usage Notes for Silicon Revision 3.0.*

3.2 Silicon Revision 2.1 Known Design Exceptions to Functional Specifications

Silicon revision 2.1 applicable advisories have been found on a later silicon revision. For more details, see Section 2.2 *Silicon Revision 3.0 Known Design Exceptions to Functional Specifications*

Table 3. Silicon Revision 2.1 Advisory List

Title	Page
Advisory 2.1.23 — Digital I/O Buffers Age Prematurely	21

Copyright © 2010–2014, Texas Instruments Incorporated

Advisory 2.1.23 *Digital I/O Buffers Age Prematurely*

Revision(s) Affected 2.1 and earlier

Details The 3.3 V digital I/O buffers on the device exhibit accelerated aging under use conditions with heavy switching activity. As a result, the recommended Power-On Hours (POH) listed in the device-specific data manual may not apply in all cases. For examples of representative lifetimes, see Examples of Representative Lifetimes (Power-On Hours) .

In a typical use case, the EMIFB clock pin (EMB_CLK) may be impacted due to the high switching rate and may eventually lead to SDRAM-related failures.

NOTE: The information in Examples of Representative Lifetimes (Power-On Hours) and Examples of Representative Lifetimes (Power-On Hours) is provided solely for user convenience and ***does not*** extend or modify the warranty provided under any terms and conditions, including TI's Standard Terms and Conditions of Sale for Semiconductor Products.

Examples of Representative Lifetimes (Power-On Hours)

NOMINAL DVDD VOLTAGE (V)	I/O SWITCHING FREQUENCY (MHz)	DPPM[1] ESTIMATE	FAIL RATE (%)	LIFETIME ESTIMATES (YEARS)					PERIPHERAL
				HOURS (HRS) PER DAY OF USE					
				2 HRS	3 HRS	7 HRS	12 HRS	24 HRS	
3.30	133	1000	0.1	8.9	5.9	2.5	1.5	0.7	EMIFB
		2500	0.25	>15	11.7	5.0	2.9	1.5	
		5000	0.5	> 15	> 15	7.3	4.3	2.1	
3.30	16.94	1000	0.1	> 15	> 15	> 15	11.6	5.8	McASP
		2500	0.25	> 15	> 15	> 15	> 15	11.5	
		5000	0.5	> 15	> 15	> 15	> 15	> 15	

[1] Defective parts per million.

Accelerated aging of buffers depends on I/O switching frequency and I/O voltage. Prolonged high frequency switching and operating at higher voltages causes buffer performance to degrade more rapidly.

Workaround(s) The following methods should be used to prolong the lifetime of the device.

METHOD 1

The 3.3 V output signals should only toggle when required for system functionality. The options are to tristate output buffers when not in use, or to run at the minimum frequency required for the specific application.

METHOD 2

Reduce the I/O voltage, DVDD.

Note: Do not reduce I/O voltage on devices already deployed or showing premature aging effects. To see the effects of premature aging on output signals, See the IO Buffer Premature Aging Assessment Wiki page.

Examples of Representative Lifetimes (Power-On Hours) depicts the subsequent performance improvements by reducing the I/O voltage.

Copyright © 2010–2014, Texas Instruments Incorporated

Examples of Representative Lifetimes (Power-On Hours)

NOMINAL DVDD VOLTAGE (V)	I/O SWITCHING FREQUENCY (MHz)	DPPM[1] ESTIMATE	FAIL RATE (%)	LIFETIME ESTIMATES (YEARS)					PERIPHERAL
				HOURS (HRS) PER DAY OF USE					
				2 HRS	3 HRS	7 HRS	12 HRS	24 HRS	
3.15	133	1000	0.1	> 15	> 15	7.7	4.5	2.2	EMIFB
		2500	0.25	> 15	> 15	> 15	8.8	4.4	
		5000	0.5	> 15	> 15	> 15	12.9	6.4	
3.15	16.94	1000	0.1	> 15	> 15	> 15	> 15	> 15	McASP
		2500	0.25	> 15	> 15	> 15	> 15	> 15	
		5000	0.5	> 15	> 15	> 15	> 15	> 15	

[1] Defective parts per million.

The Recommended Operating Conditions range for DVDD Supply Voltage, I/O, 3.3V, has been expanded to allow for a minimum voltage of 3.0V **for Silicon Revision 2.1 and 2.0 parts with the "3V" marking only** (for more details on the die symbolization and device revision codes, see Section 1.2, *Package Symbolization and Revision Identification*).

Recommended Operating Conditions

		MIN	NOM	MAX	UNIT
DVDD	Supply voltage, I/O, 3.3V	**3.0**	3.3	3.45	V

The Absolute Maximum Ratings requirement for the Input voltage ranges, V_I I/O, 3.3V (Steady State) has been modified.

Absolute Maximum Ratings Over Operating Case Temperature Range

Input voltage ranges	V_I I/O, 3.3V (Steady State)	–0.3V to DVDD + **0.350**V

See the IO Buffer Premature Aging Assessment Wiki page for more details on the implementation of the above workarounds.

Copyright © 2010–2014, Texas Instruments Incorporated

4 Silicon Revision 2.0 Usage Notes and Known Design Exceptions to Functional Specifications

This section describes the usage notes and advisories that apply to silicon revision 2.0 of the *AM1705*.

4.1 Usage Notes for Silicon Revision 2.0

Usage notes highlight and describe particular situations where the device's behavior may not match presumed or documented behavior. This may include behaviors that affect device performance or functional correctness. These usage notes will be incorporated into future documentation updates for the device (such as the device-specific data sheet), and the behaviors they describe will not be altered in future silicon revisions.

Silicon revision 2.0 applicable usage notes have been found on a later silicon revision. For more details, see Section 2.1 *Usage Notes for Silicon Revision 3.0.*.

4.2 Silicon Revision 2.0 Known Design Exceptions to Functional Specifications

Some silicon revision 2.0 applicable advisories have been found on a later silicon revision. For more details, see

- Section 2.2, *Silicon Revision 3.0 Known Design Exceptions to Functional Specifications*
- Section 3.2, *Silicon Revision 2.1 Known Design Exceptions to Functional Specifications*

Copyright © 2010–2014, Texas Instruments Incorporated

Advisory 2.0.20 *Intermittent Boot Failures*

Revision(s) Affected 2.0 and earlier

Details

For affected silicon revisions, the DSP initiates the system boot sequence when the device is released from reset. To prepare the ARM for the user, the DSP will first initialize the ARM reset vector table with an infinite “idle” loop.

The ARM reset vector table is located in the ARM’s local RAM, however the ARM local RAM can only be accessed by two bus masters: ARM and PRU0. Therefore, the DSP must program PRU0 to copy the desired reset vector table into the ARM’s local RAM.

The PRU instructions are located inside of an instruction RAM (IRAM) which is initialized by the DSP during ROM boot (see). After the instructions are stored to IRAM, the PRU is reset and enabled to execute its instructions. In this case, the PRU is instructed to initialize the ARM reset vector table.

When the device is first powered-on, the read bus from the PRU IRAM is not initialized and will contain random values (see). Under unpredictable circumstances, the random value on the read bus may resemble a reserved instruction which can be interpreted by the PRU when the core is reset and not enabled.

If the PRU core executes this reserved instruction, it will not be able to properly execute the first functional op-code in the PRU IRAM when the core is later enabled. In this fail state, the PRU will never acknowledge to the DSP that the reset vector table was successfully initialized and the DSP will be stuck in a polling loop waiting for the PRU to complete its task.

Although the PRU core execution is stuck, the PRU IRAM read bus is now initialized with a non-reserved instruction that was fetched from the IRAM by the PRU core (see). If a secondary reset is provided to the device (either POR or WARM), the PRU will be able to execute its functional instructions as expected.

Note that in order to recover from this fail state with a secondary reset, the DSP must be allowed to execute its boot ROM up to the point where the PRU has fetched a known instruction from the PRU IRAM. The approximate count of 15k cycles into the boot ROM is sufficient.

The 15k clock cycle count does not include the 6192 clock cycles required to complete a device POR reset (see). With a 24MHz crystal, the first RESET signal must be asserted high for at least 883us (or approximately 1ms).

The long-term solution for this problem is to update the DSP boot ROM with a new PRU initialization sequence that is immune to the described fail mode (see):

1. Before resetting the PRU, the DSP will perform a read-back-verify of the PRU IRAM so that the IRAM read bus will be initialized with a known and safe state.
2. The PRU will be reset and enabled in the same clock cycle by using a single register write so that the core does not have the opportunity to interpret reserved instructions.
3. The DSP will write additional, non-critical op-codes at the beginning of the PRU IRAM so that the PRU can self-recover even if it interprets a reserved instruction.

The following symptoms are all observable for this fail mode at power-on:

1. The RESETOUT signal toggles as expected 6192 cycles after the RESET signal is asserted high.
2. The device produces no boot-mode related activity and the user’s boot program will not be loaded or executed.
3. Connecting to the DSP through JTAG emulation will show that the DSP is stuck in a loop inside of the DSP boot ROM (0x00700000 – 0x007FFFFF memory space).
4. A subsequent reset (either POR or WARM) which is initiated at least 1ms after the first POR reset is asserted high will always produce a successful boot.
5. Following a secondary reset, the device will function as expected without fail until the

Copyright © 2010–2014, Texas Instruments Incorporated

device is powered-off again.

Modify the target board so that the affected device is given a secondary reset on power-up as shown in . Two example methods are described in the sections that follow.

Although secondary resets are compatible with future silicon revisions, they are not required for devices where the root cause has been fixed via an updated DSP boot ROM. In order to reduce BOM costs, board designers may want to route a reset signal bypass path so that the workaround circuit can be depopulated on future PCB builds.

❑ Use a reset supervisor device that includes a watchdog timeout function so that the reset supervisor will issue a secondary reset if the device fails to boot. The watchdog should be serviced with a device signal that is controlled by software. Options for servicing the watchdog timeout include GPIO, unused clock sources such as OBSCLK or a periodic output peripheral like TIMER and ePWM.

Some TPS382x reset supervisors include a watchdog function (shown in).

The watchdog supervisor workaround is easy to implement, however the watchdog timeout period may exceed application boot-up time requirements. For example, the TPS3820 has a typical watchdog timeout period of 200ms. The second workaround can speed up the reset process.

❑ Implement a logic-based secondary reset circuit which is timed using RC components. For the circuit shown in , a single board reset control signal can trigger three logic transitions in a dual XOR gate device.

This is possible because each RC load connected to the board reset control signal can output a different rising-edge waveform. With increasing RC load, the resulting control signal will reach the Schmitt buffers' Vih level at a later point in time. shows the relationship between the board reset signal and the RESET signal produced by the circuit. The blue and green lines represent the voltage as seen by the Schmitt buffers. The output voltage of a charging RC circuit is defined as: Vo = Vi * (1 - e^[-t / RC])

Given ideal conditions, a 3.3V board reset signal, and an input buffer Vih of 1.4V, the following set of component values would generate an initial RESET high period (R1C1 region) of approximately 2ms and a RESET low period (R2C2 region) of approximately 0.5ms:

- R1 = 36k, C1 = 100nF
- R2 = 45k, C2 = 100nF
- R3 = 450k

When implementing this workaround, some important aspects should be kept in mind:

(a) The dual Schmitt buffer is included because the dual XOR gate has an input rise-time requirement that is violated by the RC circuits.

(b) The Board Reset signal must meet the XOR gate input rise-time requirement and must provide enough output current to charge the RC circuits to the target Vih level.

(c) It is critical for the Vih level of the two input buffers to be very close together so only single-device buffers should be considered for this circuit (such as the 2-in-1 dual Schmitt buffer device used in this example).

(d) Variations in the electrical characteristics of the circuit components may produce waveforms that deviate from ideal calculations.

(e) The sole purpose of the R3 pull-down resistor is to discharge the RC components before the board reset signal is driven high. Therefore, the value selected for R3 should be sufficiently large enough to not interfere with the RC circuits as they are charging.

Copyright © 2010–2014, Texas Instruments Incorporated

Revision History

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

This silicon errata revision history highlights the technical changes made to SPRZ311D revision to make it SPRZ311E revision.

Scope: Applicable updates relating to the device have been incorporated. Added Silicon Revision 3.0 device-specific information.

Revision History

SEE	ADDITIONS/MODIFICATIONS/DELETIONS
Section 2.1	Added the following new usage note to Usage Notes for Silicon Revision 3.0: • Section 2.1.1, McASP: Inactive Slot Usage Note
Section 2.2	Added the following new advisories to Silicon Revision 3.0 Known Design Exceptions to Functional Specifications: • Advisory 3.0.25, USB0: CPU gets Stale Receive Data from the Data Buffer located in External Memory • Advisory 3.0.26, USB0: Early DMA Completion in DMA Receive Mode and More Than One Endpoint is Transferring Data • Advisory 3.0.27, USB0: DMA Hung up in Frequent Teardowns

Copyright © 2010–2014, Texas Instruments Incorporated

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, enhancements, improvements and other changes to its semiconductor products and services per JESD46, latest issue, and to discontinue any product or service per JESD48, latest issue. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All semiconductor products (also referred to herein as “components”) are sold subject to TI’s terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its components to the specifications applicable at the time of sale, in accordance with the warranty in TI’s terms and conditions of sale of semiconductor products. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by applicable law, testing of all parameters of each component is not necessarily performed.

TI assumes no liability for applications assistance or the design of Buyers’ products. Buyers are responsible for their products and applications using TI components. To minimize the risks associated with Buyers’ products and applications, Buyers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI components or services are used. Information published by TI regarding third-party products or services does not constitute a license to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of significant portions of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI components or services with statements different from or beyond the parameters stated by TI for that component or service voids all express and any implied warranties for the associated TI component or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Buyer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products, and any use of TI components in its applications, notwithstanding any applications-related information or support that may be provided by TI. Buyer represents and agrees that it has all the necessary expertise to create and implement safeguards which anticipate dangerous consequences of failures, monitor failures and their consequences, lessen the likelihood of failures that might cause harm and take appropriate remedial actions. Buyer will fully indemnify TI and its representatives against any damages arising out of the use of any TI components in safety-critical applications.

In some cases, TI components may be promoted specifically to facilitate safety-related applications. With such components, TI’s goal is to help enable customers to design and create their own end-product solutions that meet applicable functional safety standards and requirements. Nonetheless, such components are subject to these terms.

No TI components are authorized for use in FDA Class III (or similar life-critical medical equipment) unless authorized officers of the parties have executed a special agreement specifically governing such use.

Only those TI components which TI has specifically designated as military grade or “enhanced plastic” are designed and intended for use in military/aerospace applications or environments. Buyer acknowledges and agrees that any military or aerospace use of TI components which have ***not*** been so designated is solely at the Buyer's risk, and that Buyer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI has specifically designated certain components as meeting ISO/TS16949 requirements, mainly for automotive use. In any case of use of non-designated products, TI will not be responsible for any failure to meet ISO/TS16949.

Products		**Applications**	
Audio	www.ti.com/audio	Automotive and Transportation	www.ti.com/automotive
Amplifiers	amplifier.ti.com	Communications and Telecom	www.ti.com/communications
Data Converters	dataconverter.ti.com	Computers and Peripherals	www.ti.com/computers
DLP® Products	www.dlp.com	Consumer Electronics	www.ti.com/consumer-apps
DSP	dsp.ti.com	Energy and Lighting	www.ti.com/energy
Clocks and Timers	www.ti.com/clocks	Industrial	www.ti.com/industrial
Interface	interface.ti.com	Medical	www.ti.com/medical
Logic	logic.ti.com	Security	www.ti.com/security
Power Mgmt	power.ti.com	Space, Avionics and Defense	www.ti.com/space-avionics-defense
Microcontrollers	microcontroller.ti.com	Video and Imaging	www.ti.com/video
RFID	www.ti-rfid.com		
OMAP Applications Processors	www.ti.com/omap	**TI E2E Community**	e2e.ti.com
Wireless Connectivity	www.ti.com/wirelessconnectivity		

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2014, Texas Instruments Incorporated