Zynq-7000 AP SoC Production Errata

EN247 (v1.11) May 22, 2017 **Errata Notification**

Introduction

Thank you for designing with the Xilinx Zynq®-7000 All Programmable SoC (AP SoC) family of devices. Although Xilinx has made every effort to ensure the highest possible quality, the devices listed in Table 1 are subject to the limitations described in the following errata.

Devices

These errata apply to the devices shown in Table 1.

Table 1: **Devices Affected by These Errata**

Product Family	Device	JTAG ID (Revision Code)	Packages	Speed Grades	Junction Temperature Range
Zynq-7000	XC7Z007S	1 or later	All	All	All
	XC7Z012S	0 or later			
	XC7Z014S	2 or later			
	XC7Z010	1 or later			
	XC7Z015	0 or later			
	XC7Z020	2 or later			
	XC7Z030	1 or later			
	XC7Z035	2 or later			
	XC7Z045	2 or later			
	XC7Z100	0 or later			

Processor System (PS) Errata Details

This section provides a detailed description of each processor system issue known at the release time of this document, including applicable errata from third-party IP, which has been modified to reflect implementation in the devices listed in Table 1. Additional information for each issue is available in the associated answer record. For a disposition of each ARM Cortex-A9 errata, see Answer Record 55518.

APU

Under Very Rare Timing Circumstances, Transition Into Streaming Mode Might Create A Data Corruption

Answer Record 65545

Under very rare timing circumstances, a data corruption might occur on a dirty cache line that is evicted from the L1 Data Cache due to another cache line being entirely written.

The erratum requires the following conditions:

- The CPU contains a dirty line in its data cache.
- The CPU performs at least four full cache line writes, one of which is causing the eviction of the dirty line.
- Another CPU, or the ACP, is performing a read or write operation on the dirty line.

This is a third-party errata (ARM, Inc. 845369); this issue will not be fixed.

Processor Might Miss Watchpoint On Second Part Of Unaligned Access Crossing Page Boundary

Answer Record 47546

Under rare conditions, a watchpoint on the second part of an unaligned access crossing a page boundary that misses in the microTLB for the second part of its request might not be detected. A guard watchpoint can be set on the last byte of the first page and handle the false-positive matches when they occur.

This is a third-party errata (ARM, Inc. 751476); this issue will not be fixed.

Following An ASID Switch, Faulty MMU Translations Can Occur

Answer Record 47547

A microTLB entry can be corrupted following an ASID switch, possibly corrupting subsequent MMU translations. The issue occurs when a speculative explicit memory access is executed under wrong speculation. This typically happens when the memory access occurs under a mispredicted branch or in case it is conditional and the condition fails.

The work-around is to add a DSB instruction in the ASID switch code sequence.

This is a third-party errata (ARM, Inc. 754322); this issue will not be fixed.

Ordering Of Read Accesses To The Same Memory Location Might Not Be Ensured

Answer Record 47548

The processors operating in an SMP environment can experience a read access bypassed by a following read access to the same memory location. The issue only occurs for read accesses to a memory region that is marked as a Normal Memory Write-Back Shared.

This situation is not common and there are multiple work-arounds.

This is a third-party errata (ARM, Inc. 761319); this issue will not be fixed.

System Deadlock Can Occur In SMP Mode When The Same Cache Line Is Accessed By Both CPUs And The ACP

Answer Record 47549

Under very rare circumstances, some ACP requests in hazard with multiple full cache line writes by the CPUs can cause arbitration issues in the SCU leading to processor deadlock. For the condition to occur, the CPUs must be operating in SMP mode and on a coherent memory region while one CPU writes a cache line at approximately the same time as the other CPU is reading the same cache line and the ACP has also issued a request to the cache line.

The situation affects SMP systems and is very improbable because the timing window is very small and the three requests all center around one cache line. If the condition cannot be avoided in the operating system, then a control bit can be set so the problem does not occur.

This issue only applies to the dual-core devices (XC7Z010, XC7Z015, XC7Z020, XC7Z030, XC7Z035, XC7Z045, and XC7Z100).

This is a third-party errata (ARM, Inc. 761320); this issue will not be fixed.

Cache Line Maintenance Operations By MVA Might Not Succeed On An Inner Shareable Memory Region

Answer Record 47550

Under certain timing circumstances, an MVA data or unified cache line maintenance operation that targets an Inner Shareable memory region can fail to proceed up to either the Point of Coherency or to the Point of Unification of the system. This is likely to affect self-modifying code. For this problem to occur, the two processors work in the SMP mode with the broadcasting of CP15 maintenance operations being enabled. This issue has a known work-around.

This is a third-party errata (ARM, Inc. 764369); this issue will not be fixed.

ISB Instruction Is Counted In Performance Monitor Events 0x0C and 0x0D

Answer Record 47551

The ISB instruction is implemented as a branch in the Cortex™-A9 microarchitecture. This implies that events `0x0C` (software change of PC) and `0x0D` (immediate branch) are asserted when an ISB occurs, which is not compliant with the ARM® v7 architecture. Therefore, the count of events `0x0C` and `0x0D` are not accurate when using the Performance Monitor counters.

To obtain the precise count of software change of PC (`0x0C`) and immediate branches (`0x0D`), count ISB instructions alone with event `0x90` and subtract this ISB count from the events `0x0C` and `0x0D` results.

This is a third-party errata (ARM, Inc. 725631); this issue will not be fixed.

ARM MainID Registers Are Not Aliased To Debug Interface On APB

Answer Record 47552

Debug Registers 838 and 839 defined by the ARM Debug Architecture as the alias of the MainID register are not implemented on the APB. If the debugger, or any other external agent, tries to read the MIDR register using the alias addresses, it receives an incorrect answer (`0x0`), which can cause multiple types of malfunctions in the debugger.

To avoid this, always access the MIDR at its original address, `0xD00`, and not at any of its alias addresses.

This is a third-party errata (ARM, Inc. 729817); this issue will not be fixed.

ARM Debug Execution Stalls When An Instruction Is Written To The ITR Following An Aborted Load/Store

Answer Record 47553

When the processor is in debug state and the sdabort flag is set, an instruction written to the ITR after an aborted Load/Store instruction is erroneously executed instead of the intended operation of being discarded. When the ITR instruction is executed, different failures can occur.

This is a third-party errata (ARM, Inc. 729818); this issue will not be fixed.

Debug Program Counter Sampling (DBGPCSR) Register Format Is Incorrect

Answer Record 47554

The DBGPCSR register format is not strictly correct, but the debug tools can calculate the expected PC value and instruction state.

This is a third-party errata (ARM, Inc. 751471); this issue will not be fixed.

Imprecise Abort Can Be Reported Twice On Non-Cacheable Reads

Answer Record 47555

When the CPU has two outstanding read memory requests to a device or non-cacheable normal memory region, and the first one receives an imprecise external abort, then the second access can falsely report an imprecise external abort, too. In practice, imprecise aborts are usually unrecoverable failures and include a processor reset or whole system reboot, so the second abort would have not impact.

This is a third-party errata (ARM, Inc. 752519); this issue will not be fixed.

Repeated CPU Store Instructions Within Same Cache Line Can Delay Visibility Of The Store

Answer Record 47556

The CPU has a Store Buffer with merging capabilities within a cache line for Normal Memory regions. The buffer continues to merge data as long as the write accesses are performed to the same cache line. The Store Buffer has a small counter to push the data out to memory after a period of time to provide external visibility of the stores. The issue is that the counter is reset each time new data is merged. If a software code sequence is looping, and continues writing data in the same cache line repeatedly, the external visibility of the written data is potentially delayed indefinitely.

The recommended work-around is to insert a DMB operation after the write operation(s) continually being held in the store buffer.

This is a third-party errata (ARM, Inc. 754323); this issue will not be fixed.

Sticky Pipeline Advance Bit Is Not Supported

Answer Record 47557

The Sticky Pipeline Advance bit in DBGDSCR register enables the debugger to detect whether the processor is idle. The CPU does not implement accesses to DBGDRCR[3] via the debug APB interface, so the debugger is unable to clear the Sticky Pipeline Advance bit.

This is a third-party errata (ARM, Inc. 756421); this issue will not be fixed.

Unallocated Memory Hint Instruction Can Generate An Undefined Exception Instead Of Being Treated As A NOP

Answer Record 47558

The Unallocated Memory Hint instruction should execute as a NOP, but the CPU generates an UNDEF exception when bits [15:12] of the instruction encoding are not `0x0F`. In practice, this issue is not expected to be significant because such instruction encodings are not to be generated by the compiler, nor used in handcrafted programs.

The work-arounds consist of modifying the instruction encoding so bits [15:12] = `0x0F` or make the exception handler emulate the expected behavior of the instruction, i.e., NOP, before returning to normal program execution.

This is a third-party errata (ARM, Inc. 757119); this issue will not be fixed.

MRC And MCR Instructions Are Not Counted In Event 0x68

Answer Record 47559

MRC and MCR instructions are not counted in the total number of instructions passing through the Register rename pipeline stage. The values of event `0x68` and PMUEVENT[9:8] are imprecise.

This is a third-party errata (ARM, Inc. 761321); this issue will not be fixed.

Read Accesses To DBGPRSR And DBGOSLSR Can Generate An Unexpected Undefined Exception

Answer Record 47560

CP14 read accesses to the DBGPRSR and DBGOSLSR registers generate an unexpected UNDEF exception when DbgSwEnable = `0`, even in privileged mode. In practice, these registers are not accessible when DbgSwEnable = `0`. However, this is not expected to cause any significant issue because these accesses are mainly intended to be used as part of debug across power-down sequences, which is not a supported feature.

The work-around is to set the DbgSwEnable bit to `1` temporarily so that the DBGPRSR and DBGOSLSR registers can be accessed. Note, the Control/Status Word register in DAP can only be accessed through PS JTAG.

This is a third-party errata (ARM, Inc. 764319); this issue will not be fixed.

The High Priority For SO And Dev Reads Feature Can Cause QoS Issues To Cacheable Read Transactions

Answer Record 47561

When the "High Priority for SO and Dev reads" feature is enabled, the L2 cache controller gives a higher priority to Strongly Ordered (SO) and Device read requests than normal cacheable reads. When the controller receives a continuous flow of SO/Device reads, the activity can prevent L2 cache line fill requests from being forwarded to the memory.

A work-around is only necessary in systems that are able to issue a continuous flow of SO or Device reads. In such a case, the work-around is to disable the "High Priority for SO and Dev reads" feature. This is the default setting in the L2 Controller.

This is a third-party errata (ARM, Inc. 729815); this issue will not be fixed.

A Continuous Write Flow Can Stall A Read Targeting The Same Memory Area

Answer Record 47562

When a read (cacheable or not) with Normal Memory attributes is received by the L2 cache controller, hazard checking is performed on the read with the active writes in the store buffer. If an address match is detected, the read is stalled until the write completes. However, a continuous flow of writes can stall a read targeting the same memory area.

This issue does not lead to data corruption and normal software code is not expected to contain long write sequences.

This is a third-party errata (ARM, Inc. 754670); this issue will not be fixed.

L2 Cache Controller Can Prefetch Across 4 KB Boundary With Offset Set To 23

Answer Record 47563

When prefetch is enabled and the prefetch offset is equal to 23 (`0x17`), then the L2 cache controller prefetches across a 4 KB address boundary. This can cause system issues because those cache line-fills can target a new 4 KB page of memory space, regardless of page attribute settings in the L1 MMU.

The offset values for the prefetch unit can be set from 0 to 31, but to avoid prefetching across the 4 KB boundary, it must never be set to 23. The default value is 0 and enables the next cache line to be prefetched.

This is a third-party errata (ARM, Inc. 765569); this issue will not be fixed.

PLD Instructions Might Allocate Even In A Disabled Data Cache

Answer Record 47584

PLD instructions prefetch and allocate any data marked as Write-Back (either Write-Allocate or Non-Write-Allocate, Shared or Non-Shared), regardless of the processor configuration settings, including the Data Cache Enable bit value. This can create data consistency issues. This issue does not occur if the data cache is enabled.

The work-around requires software to set a bit in an undocumented Control register. Setting this bit causes all PLD instructions to be treated as NOPs.

This is a third-party errata (ARM, Inc. 771221); this issue will not be fixed.

Visibility Of Debug Enable Access Rights To Enable/Disable Tracing Is Not Ensured By An ISB Instruction

Answer Record 47585

Although visibility is correctly achieved for all debug-related features, the ISB instruction is not sufficient to make the Authentication Status Register changes visible to the trace flow. As a consequence, the trace stops with the current waypoint up to the next exception entry or return, or to the next serial branch, even when an ISB is executed.

To work around the issue, the ISB instruction must be replaced by one of the events causing the change to be visible. In particular, replacing the ISB by a MOVS PC to the next instruction achieves the correct functionality.

This is a third-party errata (ARM, Inc. 771224); this issue will not be fixed.

Speculative Cacheable Reads To Aborting Memory Regions Clear The Internal Exclusive Monitor, Can Lead To Livelock

Answer Record 47586

When a cacheable read receives an external abort, the aborted line is allocated as invalid in the Data Cache and any allocation in the Data Cache clears the internal exclusive monitor. Therefore, if a program executes a LDREX/STREX loop, with the DSB instruction within, and it keeps on receiving an abort answer in the middle of the LDREX/STREX sequence, then the LDREX/STREX sequence never succeeds.

The issue happens in systems that might generate external aborts in answer to cacheable memory requests. There are two work-arounds: turn on the branch prediction or remove the DSB in the middle of the LDREX/STREX sequence.

This is a third-party errata (ARM, Inc. 771225); this issue will not be fixed.

Parity Errors On BTAC And GHB Are Always Reported Regardless Of The Parity Enable Bit Setting

Answer Record 47587

With dynamic branch prediction enabled, the CPU reports parity errors occurring on the BTAC and GHB RAMs. This reporting is done even when the parity error detection mechanism for the RAM is disabled.

Parity error detection is usually enabled. A work-around, if needed, is to enable parity error detection prior to enabling the dynamic branch prediction. In systems where branch prediction is enabled while parity error detection remains disabled, the work-around is to ignore any parity issues.

This is a third-party errata (ARM, Inc. 771223); this issue will not be fixed.

Strongly Ordered Write Followed By LDREX Might Deadlock Processor

Answer Record 51122

A write to a Strongly Ordered memory region, followed by a condition-failed LDREX instruction, might deadlock the processor.

This is a third-party errata (ARM, Inc. 782772); this issue will not be fixed.

A Data Cache Maintenance Operation Which Aborts, Followed By An ISB, Without Any DSB In-between, Might Lead To Deadlock

Answer Record 52031

Under certain circumstances, a data cache maintenance operation that aborts and is followed by an ISB, without a DSB occurring between these events, might lead to processor deadlock.

This is a third-party errata (ARM, Inc. 775420); this issue will not be fixed.

A Short Loop Including A DMB Instruction Might Cause A Denial Of Service On Another Processor That Is Attempting To Execute A CP15 Broadcast Operation

Answer Record 52032

A short code loop that includes a DMB instruction might cause a denial of service on another processor that is attempting to execute a CP15 broadcast operation.

This is a third-party errata (ARM, Inc. 794072); this issue will not be fixed.

Speculative Instruction Fetches With MMU Disabled Might Not Comply With Architectural Requirements

Answer Record 52033

The CPU usually operates with both the MMU and branch prediction enabled. If the processor operates in this condition for any significant amount of time, the branch target address cache (BTAC) will contain branch predictions. If the MMU is then disabled, but branch prediction remains enabled, these stale BTAC entries can cause the processor to make speculative instruction fetches to read-sensitive locations. This violates the ARMv7 architectural rules regarding speculative fetches documented in the ARM Architecture Reference Manual.

This is a third-party errata (ARM, Inc. 794073); this issue will not be fixed.

A Write Request To Uncacheable, Shareable Normal Memory Region Might Be Executed Twice, Possibly Causing A Software Synchronization Issue

Answer Record 52034

Under certain timing circumstances specific to the Cortex-A9 microarchitecture, a write request to an Uncacheable, Shareable Normal memory region might be executed twice, causing the write request to be sent twice on the AXI bus. The repetition of the write usually has no impact on the overall behavior of the system, unless the repeated write is used for synchronization purposes.

This is a third-party errata (ARM, Inc. 794074); this issue will not be fixed.

Updating A Translation Entry To Move A Page Mapping Might Erroneously Cause An Unexpected Translation Fault

Answer Record 52035

Under certain conditions specific to the Cortex-A9 microarchitecture, a write operation that updates a Cacheable translation table entry might cause both the old and new translation entries to be temporarily invisible to translation table walks, thus erroneously causing a translation fault.

This is a third-party errata (ARM, Inc. 782773); this issue will not be fixed.

CPU Performance Monitor Event 0x0A Might Count Twice The LDM PC ^ Instructions

Answer Record 52036

The LDM PC ^ instructions with base address register write-back might be counted twice in the Performance Monitor event `0x0A`, which is counting the number of exception returns.

The associated PMUEVENT[11] signal is also affected by this issue, and might be asserted twice by a single LDM PC ^ with base address register write-back.

This is a third-party errata (ARM, Inc. 775419); this issue will not be fixed.

A Spurious Event 0x63, "STREX Passed," Might Be Reported On An LDREX Instruction

Answer Record 55018

A write to Strongly Ordered memory region, followed by the execution of an LDREX instruction, might cause the "STREX passed" event to be signaled even if no STREX instruction is executed. As a result, the reported count of `0x63` events might be more than the actual number that took place. This issue also affects the associated PMUEVENT[27] signal, which again will report the same spurious events.

This is a third-party errata (ARM, Inc. 782774); this issue will not be fixed.

Possible Denial Of Service For Coherent Requests On An L1 Cache Line Which Is Continuously Written By A Processor

Answer Record 55325

A processor that performs a continuous stream of writes to the same cache line might prevent the other coherent processor or ACP from accessing the line.

This is a third-party errata (ARM, Inc. 791420); this issue will not be fixed.

A Branch-To-Self Instruction In The Last L1 Cache Line Of A Page Might Cause A Denial Of Service

Answer Record 55326

The execution of a branch-to-self loop from the last L1 cache line of a page might cause a continuous stream of requests within the processor, which might stall a CP15 broadcast operation that is executed by the other processor.

Note that because the processor executing the CP15 broadcast operation cannot complete that operation, it cannot enter any debug-mode, and cannot take any interrupts. If the processor executing the branch-to-self loop cannot be interrupted (e.g., if it has disabled its interrupts or if all interrupts are routed to other processor), then this issue might cause a system livelock.

This is a third-party errata (ARM, Inc. 799769); this issue will not be fixed.

"Write Context ID" Event In A CPU Is Updated On Read Access

Answer Record 55327

An instruction that reads the Context ID register, CONTEXTIDR, also updates the Write Context ID event counter.

This is a third-party errata (ARM, Inc. 795769); this issue will not be fixed.

DBGPRSR Sticky Reset Status Bit Is Set To 1 By The CPU Debug Reset Instead Of By The CPU Non-Debug Reset

Answer Record 55328

The ARM architecture specifies that the processor sets the Sticky Reset Status Bit, DBGPRSR[SR], to 1 when the non-debug logic of the processor is in reset state. Instead, the processor sets this bit to 1 when the debug logic of the processor is in reset state.

This is a third-party errata (ARM, Inc. 799770); this issue will not be fixed.

DDR

When ECC Is Enabled, The CHE_CORR_ECC_ADDR_REG_OFFSET Register Might Report An Incorrect Column Address

Answer Record 64258

When ECC is enabled, ddrc.CHE_CORR_ECC_ADDR_REG_OFFSET [CORR_ECC_LOG_COL] and ddrc.CHE_UNCORR_ECC_ADDR_REG_OFFSET[UNCORR_ECC_LOG_COL] fields might not update the column address properly depending on the position of the erroneous bit in the received data word.

This is a third-party errata; this issue will not be fixed.

LPDDR2 Per-Bank Refresh Is Not Supported

Answer Record 47580

The LPDDR2 per-bank refresh function is not supported. The work-around is to use the all-bank refresh instead.

This is a third-party errata; this issue will not be fixed.

Read Operations Malfunction When They Follow An MRW Within 128 DDR Clock Cycles

Answer Record 47581

The MRW operation requires time to execute. If an MRR or normal memory read operation occurs within 128 DDR clock cycles after the MRW cycle, the data from the MRR or normal memory read operation is corrupted. The corruption can be avoided by not issuing either read operation within the 128 clock cycle period after the MRW operation.

This is a third-party errata; this issue will not be fixed.

Gigabit Ethernet

Unicast And Broadcast Pause Frames Received By The Controller Are Not Filtered Out

Answer Record 52025

Unicast and broadcast pause frames should be filtered out by the controller when the controller's Disable Copy of Pause Frames bit is set, gem.NET_CFG [23] = 1. However, this feature does not work and these pause frames will be passed on to memory regardless of this bit setting. Pause frames rarely occur and when they do, the software can filter them out.

This is a third-party errata; this issue will not be fixed.

Back-off Time Is More Aggressive Than The Standard Requirement

Answer Record 52026

The controller's back-off time does not strictly adhere to the Truncated Binary Exponential Back-off algorithm. In some cases, the controller is more aggressive. Depending on the system, more or less collisions will occur. The algorithm discrepancy is not observed until after five consecutive collisions and only applies to the controller in 100 Base-T mode.

This is a third-party errata; this issue will not be fixed.

Packets Up To 1,536 Bytes Are Allowed With VLAN Tagging

Answer Record 52027

The IEEE Std 802.3 specification states that the maximum size of a non-jumbo packet should be 1,518 bytes. With VLAN tagging, this maximum size is raised to 1,522 bytes. When the Receive 1536-Byte Frames bit is set, gem.NET_CFG[8] = 1, the controller allows up to 1,536 bytes in the packet to be received when the controller should only allow up to 1,522 bytes.

This is a third-party errata; this issue will not be fixed.

Receive Path Lock-Up Might Occur When A Large Number Of Receive Resource Errors Are Generated

Answer Record 52028

Under heavy traffic of small sized Ethernet frames (~64 bytes) and when the controller encounters a large number of resource errors, there is a rare chance for the receive logic to completely lock down the receive path. The controller can again receive frames after a flush and reset of the receive logic.

This is a third-party errata; this issue will not be fixed.

I2C

I2C Slave Monitor Mode Can Lock The I2C Bus

Answer Record 63245

When the PS I2C is in slave monitor mode and the slave device does not respond with an ACK, the host software cannot deactivate the mode by setting the i2c.Control_reg0[SLVMON] register field to 0. If the PS I2C master is the only master on the I2C bus, it cannot communicate with other slaves on the bus. In a multi-master system, this issue can lock the bus and prevents another I2C master from owning the bus.

This is a third-party errata; this issue will not be fixed.

I2C Transaction Corruption In Slave Receiver Mode

Answer Record 63025

When the PS I2C controller is in slave receiver mode and more than one I2C slave is connected to the same bus, the PS I2C controller can receive and acknowledge data sent to a different I2C slave if the transferred data pattern contains `0xF0` or `0xF1` followed by the slave address of the PS I2C controller. For data bytes that follow the slave address, the PS I2C controller sets the i2c.Status_reg0[RXDV] register field to 1 and generates an ACK.

If the intended I2C slave does not acknowledge the data, then the bus becomes corrupted because the I2C master sees an erroneous ACK instead of the NACK from the intended I2C slave.

This is a third-party errata; this issue will not be fixed.

Glitch Filter In I2C Fast Mode Not Implemented

Answer Record 61861

The I2C controller specification v2.1 requires filtering out glitches/spikes up to 50 ns wide on the SDA and SCL lines in fast mode.

The I2C controller does not implement circuitry to filter glitches. Glitches on the SDA or SCL lines can cause a false trigger on the signal line. Glitches on the SDA line can result in an incorrect START condition or an incorrect STOP condition being recognized. A glitch on the SCL line can result in incorrect data transfer. In these cases, data transfer can be corrupted and the bus can hang.

This is a third-party errata; this issue will not be fixed.

I2C Master Generates Invalid Read Transactions

Answer Record 61664

When the I2C controller is operating as a master and has issued a read transaction (i.e., it is a master receiver), the HOLD bit (i2c.Control_reg0[HOLD]), when set, prevents the controller from issuing a STOP condition at the end of the transfer. Instead, when transfer_size (i2c.Transfer_size_reg0) is 0, SCL is held Low until software decides what to do next. If the controller timeout expires (see i2c.Time_out_reg0[TO]) while the controller is holding SCL Low, it incorrectly transfers 16 bytes of additional read data from the slave device. The transfer_size register gets corrupted and rolls over to `0xFF` as well. If there is room in the FIFO, these additional bytes are saved in the FIFO. The issue happens even if the timeout interrupt is disabled.

This is a third-party errata; this issue will not be fixed.

Missing I2C Master Completion Interrupt

Answer Record 61665

When the I2C controller is operating as a master, a completion interrupt should be signaled when a transfer is complete. However, this interrupt is not signaled after a transfer if the transfer was initiated with a repeated start condition. The exact sequence is as follows:

1. Master begins a read transfer.
   a. This transfer could begin with a Start or a Repeated Start condition.
   b. The HOLD bit (i2c.Control_reg0[HOLD]) must be set at the end of the transfer.
   c. The COMP interrupt (i2c.Interrupt_status_reg0[COMP]) is properly signaled when this transfer is done.
2. Master begins a second read transfer with a new address.
   a. Because the HOLD bit was set in step1, the controller begins this transfer with a repeated start condition.
   b. The HOLD bit must be cleared at the end of the transfer. This causes a STOP condition.
   c. The COMP interrupt is not signaled when this transfer is done.

This is a third-party errata; this issue will not be fixed.

I2C Violates $t_{HD;STA}$ Requirement For Standard-Mode When It Runs Faster Than 90 kHz

Answer Record 59366

The I2C controller violates the I2C-bus specification 4.0 μs minimum requirement on $t_{HD;STA}$ (hold time [repeated] START condition) when it runs above 90 kHz in Standard-mode. At the maximum Standard-mode SCL clock frequency (100 kHz), the I2C controller $t_{HD;STA}$ time is 3.6 μs, and I2C devices that require the minimum 4.0 μs start condition can fail to recognize the start condition from the I2C controller.

This is a third-party errata; this issue will not be fixed.

I2C Violates t_{LOW} Requirement For Fast-Mode When It Runs Faster Than 384 kHz

Answer Record 60693

The I2C controller violates the I2C-bus specification 1.3 µs minimum requirement on t_{LOW} (LOW period of the SCL clock) when it runs above 384 kHz in Fast-mode. At the maximum Fast-mode SCL clock frequency (400 kHz), the I2C controller t_{LOW} time is 1.25 µs, and I2C devices that require the minimum 1.3 µs t_{LOW} time can fail to operate with the I2C controller.

This is a third-party errata; this issue will not be fixed.

I2C Violates t_{BUF} Requirement For Fast-Mode When It Runs Faster Than 384 kHz

Answer Record 60694

The I2C controller violates the I2C-bus specification 1.3 µs minimum requirement on t_{BUF} (bus free time between a STOP and START condition) when it runs above 384 kHz in Fast-mode. At the maximum Fast-mode SCL clock frequency (400 kHz), the I2C controller t_{BUF} time is 1.25 µs, and I2C devices that require the minimum 1.3 µs t_{BUF} time can fail to prepare for the next START condition from the I2C controller.

This is a third-party errata; this issue will not be fixed.

I2C Missing Arbitration On Repeated Start

Answer Record 60695

When the I2C controller is used as a master on a multi-master bus, the I2C controller does not detect when another master I2C controller drives SDA Low for a repeated START. The I2C controller can corrupt a transaction on the bus when *all* of the following conditions are met:

- The other master starts a transaction at the same time.
- The other master addresses the same slave.
- The other master selects the same transaction (read or write). For a write, the other master drives the same data. For a read, the other master ACKs at the same time.
- The other master reads or writes the same amount of data.
- At the end of the transaction, the other master issues a repeated start.

Under these conditions, the I2C controller can corrupt a transaction on the bus.

This is a third-party errata; this issue will not be fixed.

QSPI

QSPI RxFIFO Not Empty Status Is Not Updated Promptly

Answer Record 47575

There is a delay in updating the QSPI Intr_status_reg.RX_FIFO_not_empty bit. This can cause polling software to erroneously assume that there is still data in the RxFIFO when there is none and cause the RxFIFO to under-run. This leads to invalid data being read. To avoid this, software can read the Intr_status_reg.RX_FIFO_not_empty bit twice to allow enough time for the controller to update the status bit.

This is a third-party errata; this issue will not be fixed.

QSPI Controller Reports Wrong "Busy Status" Of Flash Memories In Dual Parallel Configuration When Auto CS And "Divide by 2" Baud Rate Is Used

Answer Record 60978

When the QSPI controller is in dual parallel mode and issues a read status (RDSR) command to external flash memory devices using TXD2, the controller does not combine the Write-In-Progress (WIP) bit from both flash devices to generate the final status. When the QSPI controller Manual_CS is set to auto mode and BAUD_RATE_DIV is divide by 2, only the status of the lower device is reported. If the upper memory is still busy, the reported status is incorrect if the lower memory is ready.

This issue will not be fixed.

SDIO

ADMA2 Burst Transactions Alignment And Length Requirements

Answer Record 47531

The length of an ADMA2 burst must be an aligned multiple of 4 bytes to avoid overwriting data in the ADMA2 buffer.

This is a third-party errata; this issue will not be fixed.

Software Reset Sequence For SDIO Can Hang The Interconnect

Answer Record 47532

The SDIO controller requires a software resetting sequence that includes enabling the SD internal clock and performing two soft reset all commands when a card is inserted, or the interconnect can hang.

This is a third-party errata; this issue will not be fixed.

Second CMD12 Can Be Issued If Auto CMD12 Is Enabled

Answer Record 47533

During Auto CMD12 execution, if software sends another command, such as CMD13, to poll for the program state, this other CMD is supposed to be driven in the CMD line. However, CMD12 is driven again.

Avoid this by disabling Auto CMD12 or only send data transfer commands while the multiple block transfer is in progress.

This is a third-party errata; this issue will not be fixed.

ADMA2 Mode Fails To Release Properly When Abort CMD Is Issued

Answer Record 47534

If the device driver sends an abort command during an ADMA2 multiple-block transfer and then initiates a DMA transfer after this abort, the controller fails to perform the DMA operation. Do not issue a DMA transaction after the ADMA2 transaction is aborted.

This is a third-party errata; this issue will not be fixed.

Transfer Complete Asserts Before Completing Busy Due To CMD13

Answer Record 47535

The SDIO controller can send CMD13 during card programming mode to check the card status. If software sends an R1b response type command (e.g., CMD38, CMD6) followed by a CMD13 (status check), the controller sends the transfer complete before completing busy due to the CMD13 status check. This can be avoided by not sending a CMD13 immediately after sending an R1b command (e.g., CMD38 and CMD6).

This is a third-party errata; this issue will not be fixed.

CMD13 Not Handled Properly When CMD19 Is In Progress

Answer Record 47536

If the software driver issues CMD13 to read the card bus when a CMD19 bus-test transaction is in progress on the bus, then the interface controller is affected by the CMD13 waiting for CRC status on the bus-test data.

This can be avoided by not sending a CMD13 when bus testing is in progress.

This is a third-party errata; this issue will not be fixed.

SMC

SMC Parallel (SRAM/NOR) Interface Does Not Correctly Assert CS0 For 64 MB Memories

Answer Record 61637

Chip Select 0 (CS0) does not go active when accessing an address in range `0xE4000000 - 0xE5FFFFFF` in the SRAM/NOR interface when the memory controller is configured to access a 64 MB memory device.

This issue will not be fixed.

SMC Parallel (SRAM/NOR) Interface Address Bit 25 Is Inverted For 64 MB Memories

Answer Record 61638

The address bit 25 (A25) appears as '`1`' when accessing `0xE2000000 - 0xE3FFFFFF` and '`0`' when accessing `0xE4000000 - 0xE5FFFFFF` in the SRAM/NOR interface when the memory controller is configured to access a 64 MB memory device.

This issue will not be fixed.

NAND With ECC Might Not Deassert CS Between Data Transactions

Answer Record 47517

The NAND flash controller normally deasserts the chip select (CS) between data transfers and keeps the chip select asserted between a data transaction and a command operation. This protocol is compatible with most devices. In certain modes of operation, the controller might incorrectly believe that it is about to perform a command operation after a data operation, and the controller keeps the chip select asserted between two data transactions.

This is avoided when ECC is disabled. When ECC is enabled:

- Use full commands for writes
- Read the ECC codes between the reading of blocks

This is a third-party errata; this issue will not be fixed.

Potential SRAM/NOR Data Error

Answer Record 47518

A potential SRAM/NOR data error can occur if all the write data of a transaction is contained in a single AXI data bus cycle. Always perform writes that require multiple AXI data bus cycles in the transaction.

This is a third-party errata; this issue will not be fixed.

NAND ECC Status Register Can Incorrectly Report A Failure For One Clock Cycle

Answer Record 47520

Software might trigger an abort that can lead to the ecc_last_status value being incorrectly shown for one cycle after the ecc_status bit has gone Low.

Software must re-read the ecc_last_status whenever it detects a non-zero value.

This is a third-party errata; this issue will not be fixed.

SPI

RxFIFO Not Empty Status Is Not Updated Promptly

Answer Record 47575

There is a delay in updating the SPI Intr_status_reg0.RX_FIFO_not_empty bit. This can cause polling software to erroneously assume that there is still data in the RxFIFO when there is none and cause the RxFIFO to under-run. This leads to invalid data being read. To avoid this, software can read the Intr_status_reg0.RX_FIFO_not_empty bit twice to allow enough time for the controller to update the status bit.

This is a third-party errata; this issue will not be fixed.

Timers

Global Timer Can Send Two Interrupts For The Same Event

Answer Record 47545

The Global Timer in single-shot mode can generate two end-of-count interrupt requests instead of one.

This can be avoided by using the auto-increment mode. Software can work around the issue by clearing the Global Timer flag after having incremented the Comparator register value.

This is a third-party errata (ARM, Inc. 740657); this issue will not be fixed.

USB

OTG In Device Mode Does Not Generate A Port Change Interrupt When Session Is No Longer Valid

Answer Record 47538

The USB controller in OTG mode and acting as self-powered device does not generate a port change interrupt when the session is no longer valid (VBUS not present). To work around this, software can enable the session valid VBUS comparator and associated interrupt to detect when the VBUS voltage level drops.

This is a third-party errata; this issue will not be fixed.

Suspend Bit Is Asserted Before The Port Enters The Suspend State

Answer Record 47539

When software instructs the USB controller (host, device, or OTG) to enter the suspend state, the controller logic immediately sets the suspend status bit. If the controller is busy with a USB transaction, it waits before actually entering the suspend state.

Although the behavior violates the EHCI specification, it does not cause a functional issue because the delay is always less than the 10 milliseconds that the application program must wait before it initiates a forced resume.

This is a third-party errata; this issue will not be fixed.

First SOF After A Software Reset Can Be Corrupted

Answer Record 47540

During the USB reset process (speed negotiation and chirp), if the protocol engine sends SOF commands to the port control, the port control filters out those SOFs. However, at the end of reset (end of chirp back from Host), when the Protocol Engine sends an SOF, the ULPI Port Control sends the SOF to the PHY before sending the update opmode command. This results in an invalid packet being sent on the line. The invalid packet in ULPI protocol is a NOPID Tx Command immediately followed by an STP pulse.

Do not enable USBCMD.RS after the USB reset (PORTSCx.PR=0) until the ULPI post reset processing has been completed, which can be checked by reading the prtsc.pr register. This ensures that the host does not send the SOF prematurely.

This is a third-party errata; this issue will not be fixed.

In HS Host Mode, NYET Decrements NAK Counter

Answer Record 47541

When a NYET handshake is received for an OUT transaction in high speed (HS) mode, the controller erroneously decrements the NAK counter.

This is a third-party errata; this issue will not be fixed.

ULPI Viewport Does Not Work With Extended Addresses

Answer Record 47543

It is not possible to read or write the ULPI PHY extended register set (addresses `0x40` and greater) using the ULPI viewport. The write operation writes the address itself as data, and a read operation returns incorrect data.

This is a third-party errata; this issue will not be fixed.

Traceability

Figure 1 shows an example device top mark for the devices listed in Table 1. Refer to XCN16014 for details on the addition of the 2D barcode.

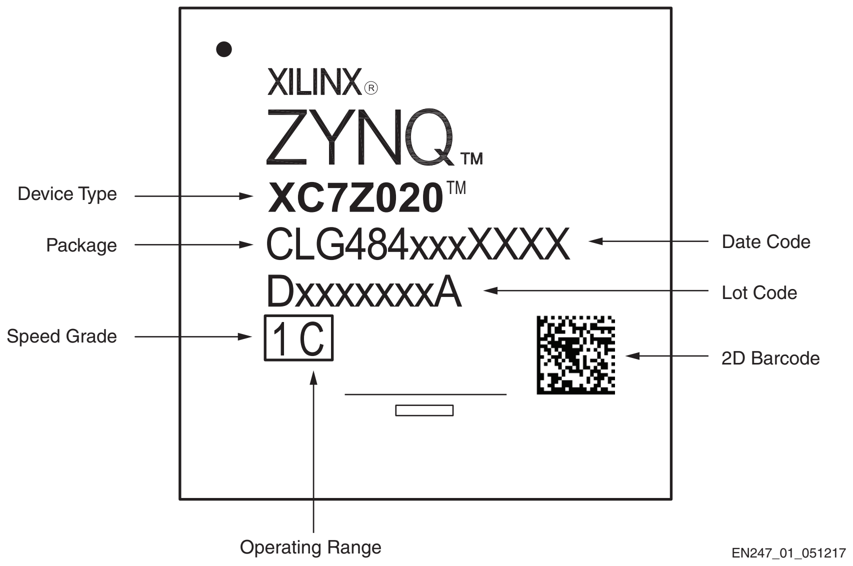

Figure 1: **Example Device Top Mark**

Additional Questions or Clarifications

For additional questions regarding these errata, contact Xilinx Technical Support: http://www.xilinx.com/support/clearexpress/websupport.htm or your Xilinx Sales Representative: http://www.xilinx.com/company/contact/index.htm.

Revision History

The following table shows the revision history for this document:

Date	Version	Description of Revisions
02/11/13	1.0	Initial Xilinx release.
03/14/13	1.1	Removed Design Tool section; customers should consult the latest data sheet. Added XC7Z045 devices (production released) to Table 1.
03/19/13	1.2	Added XC7Z010 devices (production released) to Table 1.
03/27/13	1.3	Added XC7Z030 devices (production released) to Table 1. Added the following errata items: A Data Cache Maintenance Operation Which Aborts, Followed By An ISB, Without Any DSB In-between, Might Lead To Deadlock; A Short Loop Including A DMB Instruction Might Cause A Denial Of Service On Another Processor That Is Attempting To Execute A CP15 Broadcast Operation; Speculative Instruction Fetches With MMU Disabled Might Not Comply With Architectural Requirements; A Write Request To Uncacheable, Shareable Normal Memory Region Might Be Executed Twice, Possibly Causing A Software Synchronization Issue; Updating A Translation Entry To Move A Page Mapping Might Erroneously Cause An Unexpected Translation Fault; CPU Performance Monitor Event 0x0A Might Count Twice The LDM PC ^ Instructions; and A Spurious Event 0x63, “STREX Passed,” Might Be Reported On An LDREX Instruction.
05/03/13	1.4	Added Possible Denial Of Service For Coherent Requests On An L1 Cache Line Which Is Continuously Written By A Processor; A Branch-To-Self Instruction In The Last L1 Cache Line Of A Page Might Cause A Denial Of Service; "Write Context ID" Event In A CPU Is Updated On Read Access; and DBGPRSR Sticky Reset Status Bit Is Set To 1 By The CPU Debug Reset Instead Of By The CPU Non-Debug Reset. Updated disposition to include third-party information.
06/26/13	1.5	Added the XC7Z100 device (production released) to Table 1.
02/25/14	1.6	Added the XC7Z015 device (production released) to Table 1.
10/31/14	1.7	Added the I2C section:Glitch Filter In I2C Fast Mode Not Implemented; I2C Master Generates Invalid Read Transactions; Missing I2C Master Completion Interrupt; I2C Violates $t_{HD;STA}$ Requirement For Standard-Mode When It Runs Faster Than 90 kHz; I2C Violates t_{LOW} Requirement For Fast-Mode When It Runs Faster Than 384 kHz; I2C Violates t_{BUF} Requirement For Fast-Mode When It Runs Faster Than 384 kHz; and I2C Missing Arbitration On Repeated Start. Added the QSPI section: QSPI RxFIFO Not Empty Status Is Not Updated Promptly; and QSPI Controller Reports Wrong "Busy Status" Of Flash Memories In Dual Parallel Configuration When Auto CS And "Divide by 2" Baud Rate Is Used. Added: SMC Parallel (SRAM/NOR) Interface Does Not Correctly Assert CS0 For 64 MB Memories and SMC Parallel (SRAM/NOR) Interface Address Bit 25 Is Inverted For 64 MB Memories. Added the SPI section: RxFIFO Not Empty Status Is Not Updated Promptly.
02/10/15	1.8	Added I2C Slave Monitor Mode Can Lock The I2C Bus and I2C Transaction Corruption In Slave Receiver Mode.
10/19/15	1.9	Added Under Very Rare Timing Circumstances, Transition Into Streaming Mode Might Create A Data Corruption and When ECC Is Enabled, The CHE_CORR_ECC_ADDR_REG_OFFSET Register Might Report An Incorrect Column Address.
09/27/16	1.10	Added single-core devices (XC7Z007S, XC7Z012S, and XC7Z014S) to Table 1. Updated System Deadlock Can Occur In SMP Mode When The Same Cache Line Is Accessed By Both CPUs And The ACP.
05/22/17	1.11	Updated Traceability and Figure 1 with 2D barcode information.

Disclaimer

Automotive Applications Disclaimer