Zynq-7000 AP SoC XC7Z020 CES Errata

EN208 (v1.4) June 4, 2013 **Errata Notification**

Introduction

Thank you for participating in the Zynq™-7000 All Programmable SoC (AP SoC) Engineering Sample Program. As part of this program, we are pleased to provide to you engineering samples of the devices listed in Table 1. Although Xilinx has made every effort to ensure the highest possible quality, these devices 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
Zynq-7000	XC7Z020 CES	0	All	-1, -2

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

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

Boot IOP

NAND Boot Width Is Limited To 8 Bits

Answer Record 44330

The BootROM reads the lower 8 bits of the NAND interface.

FSBL Golden Image Search Might Fail

Answer Record 47571

When the BootROM detects a problem with the boot image header (incorrect data), it does not start a golden image search. The golden image search works as expected when a CRC error is detected.

BootROM Might Hang If A NAND Boot Image Is Malformed Or Corrupted

Answer Record 47572

In boot from NAND, the BootROM might hang if it encounters a malformed or corrupted image stored in the NAND flash. In such cases, the BootROM does not complete the boot process and does not go into ErrorLockDown state.

BootROM Floats Boot Device I/O Interface Signals

Answer Record 47573

The data/IO buses of NAND, NOR, and Quad-SPI devices on the MIO pins are left floating by the BootROM, leading to higher power consumption. In non-secure boot modes, the work-around is to enable the weak internal pull-up resistors on the MIO pins during the boot process.

During NOR Boot, MIO2 And MIO14 Pins Are Inadvertently Configured By BootROM

Answer Record 47593

MIO pins 2 and 14 are not used by the SRAM/NOR interface. MIO2 is configured as a toggling output; MIO14 is configured as an input. In a successful boot, these MIO configurations are maintained until changed by user software. If using NOR boot mode, the board design must use MIO2 for purposes that can tolerate this pin being driven during BootROM execution.

During NOR Boot, MIO1 Pin Is Set To Address Bit 25

Answer Record 47594

MIO1 pin is a multifunction pin. During NOR boot, the pin is configured as address bit 25 and is driven Low when the system boots from the NOR device. MIO1 pin must not be connected as a chip select to a second SRAM/NOR device or be used in a way to adversely affect other board logic during boot.

During Quad-SPI Boot, Image Search for Dual SS 8-Bit Parallel I/O Is Performed In 64 KB Steps

Answer Record 47595

During Quad-SPI Boot using the dual SS 8-bit parallel I/O wiring connection, the image search occurs in 64 KB steps instead of 32 KB steps. Boot images must be aligned to 64 KB boundaries.

Micron 8 Gb (On-Die ECC) NAND Devices Do Not Work

Answer Record 47597

The following Micron Device ID codes (byte 1 of Address `00h` of flash ID parameter tables) do not work in boot mode: `0xA3`, `0xB3`, `0xC3`, `0xD3`.

BootROM Stops Searching For Boot Image After First 16 MB For Quad-SPI x8 and SRAM/NOR

Answer Record 47598

After a system boot, the BootROM searches for a boot image. It starts at the beginning address of the boot device (offset `0x0`). The design intention is for the BootROM to search up to 32 MB. This search limit occurs in failover and multi-boot situations.

SD Card Boot Mode Runs At Low Frequency And 1-Bit Data Width

Answer Record 51907

The SD I/O interface data width and operating frequency in boot mode are programmed by the BootROM and operates at 400 KHz or slower with a data width of 1 bit.

Quad-SPI MIO Pin 8 Is Inadvertently Enabled During Boot

Answer Record 52014

The Quad-SPI I/O interface can optionally use MIO pin 8 as a feedback output clock to enable the interface to be clocked at high frequency. The BootROM inadvertently and unnecessarily enables MIO pin 8 as a toggling output during a Quad-SPI boot sequence.

SD Card Boot Mode Tests For Card Detect On MIO Pin 0

Answer Record 52016

During boot from an SD card, the BootROM reads the logic level on the MIO pin 0 as an SD Card Detect. MIO pin 0 must be Low during the boot process. If the pin is High, the boot process fails because the BootROM assumes there is no SD card available. The design intent is for the BootROM to access the SD card without testing the card detect status.

SD Card Boot Mode Does Not Comply With The 74 CLK Cycle Requirement Between The Start Of The SD CLK And First Command

Answer Record 52023

The SD specification requires a minimum of 74 CLK cycles between the start of the SD CLK and the first command. During boot from an SD card, this requirement is violated and the first command is issued to the SD card earlier.

Boot System

PC Of CPU1 Points To An Invalid Address When Booting From JTAG

Answer Record 47567

During boot, the second CPU (CPU1) executes a WFE instruction. When parking the second CPU, the BootROM does not take into consideration that memory is remapped and hidden upon completion of the BootROM process. As a result, the second CPU resumes operation from a location that is either mapped to the OCM or to DDR memory when it receives a wake-up event. The second CPU then runs random instructions, resulting in an undefined system behavior.

Use a JTAG debugger to set the start address of the second CPU to a known address.

BootROM Error Codes Are Not Unique

Answer Record 47568

When the BootROM detects an error during the boot process, it stores an error code in the REBOOT_STATUS register. Similar errors have the same error codes, potentially making the debugging process more difficult.

PS-PL AXI Interfaces Are Enabled Upon Completion Of The BootROM

Answer Record 47569

The AXI interfaces between PS and PL are enabled upon completion of the BootROM. Software must assert the AXI interface resets early in the execution of the FSBL. Software must always use proper reset and configuration procedures starting-up the PL AXI interfaces.

SLCR Registers Are Not Locked Upon Completion Of The BootROM

Answer Record 47570

SLCR registers are not locked upon completion of the BootROM (when the BootROM gives control to the FSBL). Leaving the SLCR registers unlocked exposes them to being accidentally overwritten.

MIO Pins Are Not Three-Stated When ErrorLockDown Occurs

Answer Record 47574

The MIO I/O Buffers are not three-stated when the BootROM detects an error that forces the system into the ErrorLockDown state.

MultiBoot Feature Is Not Supported

Answer Record 47588

The MultiBoot feature is not supported. There is no work-around.

Independent JTAG Is Not Supported In JTAG Boot Mode

Answer Record 47599

Independent JTAG does not work from JTAG boot mode. If independent JTAG is selected, the PL TAP controller is accessible but the PS cannot be accessed through the DAP and remains perpetually in an idle state.

INIT_B Pin Does Not Indicate Boot Error Status

Answer Record 52012

The INIT_B pin does not indicate a PS boot failure.

PS_SRST_B Asserted During A POR Boot Does Not Result In A Secure Lockdown

Answer Record 52013

The boot process caused by the PS_POR_B reset signal must not be interrupted by the assertion of the PS_SRST_B reset signal.

Reset Reason Mechanism Does Not Use slcr.REBOOT_STATUS Register

Answer Record 52030

The BootROM does not store the reason for reset in the slcr.REBOOT_STATUS[22:16] bits.

BootROM 128KB CRC Self-Check Is Not Supported

Answer Record 55329

The BootROM will not perform a full 128KB CRC self-check of the ROM contents even if the corresponding PS eFUSE bit setting is programmed.

Clocks PS

Soft Reset Of System Clocks Does Not Consistently Reset Clock Dividers To Default Values

Answer Record 47525

Soft reset of system clocks does not consistently reset clock dividers to default values. Software must configure all clock dividers after a soft reset and not depend on default settings.

DDR PS

LPDDR2 Dynamic Clock Stop Restarts Too Soon

Answer Record 47512

The user can program the LPDDR2 controller to stop the DRAM clock when there are no memory transactions to perform and restart the clock when a memory request is received. The controller correctly stops the clock whenever the transaction queue is empty, but when the clock is restarted, the controller quickly issues a DRAM transaction. It does not recognize the tXP timing parameter value.

Either do not use the LPDDR2 clock stop feature or have software ensure that the enable/disable of the clock stop is only done when there is no DRAM activity.

DDR3 Starts DRAM Clock Too Early After Exiting Self-Refresh

Answer Record 47514

The user can program the DDR3 controller to enable Self-Refresh Clock-Stop mode. The controller correctly stops the clock to DRAM, but it does not satisfy the tCKSRE time (around 5 clock cycles) before restarting the clock.

Either do not use the DDR3 Self-Refresh mode or have software ensure that the enable/disable of the Self-Refresh is only done when there is no DRAM activity.

Controller Mishandles STREX Instruction

Answer Record 47516

Under certain circumstances with the cache disabled, the DDR Controller does not respond properly to an Exclusive Store (STREX) instruction executed by a CPU. This can be avoided by executing the STREX instruction only in cacheable memory space with the cache enabled.

DCI Quiet Mode Operation Is Not Supported

Answer Record 47521

The DDR I/O buffer (DDRIOB) DCI impedance adjustment mode can be set to 'As-Required.' The DCI 'Quiet' mode normally performs only an initial impedance adjustment but it does not work.

DCI Does Not Work In LPDDR2 Mode

Answer Record 47522

The DCI impedance adjustment mechanism is not supported when the DDR controller is in LPDDR2 mode. Software can program a fixed impedance value based on information from the Xilinx Platform Studio configurator for the DDR.

DDR I/O Buffers (DDRIOB) Do Not Support External V_{REF}

Answer Record 47564

DDR I/O buffers (DDRIOB) use differential input receivers for which one input to the receiver is connected to the data input and the other is connected to a voltage reference pin called V_{REF} (set to half of the DDR VCC I/O voltage.) V_{REF} needs to be supplied from the internal voltage source because the external V_{REF} feature for DDRIOB is not supported.

Automatic ZQ Calibration Is Not Supported In LPDDR2 Mode

Answer Record 47576

The DDR controller in LPDDR2 mode does not issue automatic long/short calibration commands (ZQCL/ZQCS) during normal operation. Calibration is done during the DDR initialization phase, but temperature and voltage usually change over time and calibration must be performed periodically. Software must issue explicit ZQ calibration commands to maintain peak performance of the DRAM I/O buffers.

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

In LPDDR2 Mode, ZQCL Command Is Not Issued After Self-Refresh Exit

Answer Record 47582

The DDR controller in LPDDR2 mode does not issue the ZQCL calibration command after exiting the self-refresh operation. Although not required by the DRAM JEDEC specs, some vendors expect that the ZQCL command will be issued after self-refresh exit and before any other memory requests can be processed.

The work-around for this issue is for software to periodically issue ZQCS commands while the LPDDR2 device is in self-refresh mode.

Read Gate Training Value Is Unreliable In Slice 3

Answer Record 52021

The four most significant bits (MSBs) of the Read Gate Training register (used by the PHY trainer) are not valid. This only applies to Slice 3 of the data and leads to incomplete data being read for the 4 MSBs of this Read Gate training result. All four data slices are expected to support 11 bits to present the ratio. The four lost bits can be estimated by reading the remaining seven bits and the four MSBs from the other slices.

Debug

ITM And FTM Frames Might Be Indistinguishable By Software

Answer Record 47527

All PS Trace sources insert a 3-bit non-configurable ID in the packet to allow software to distinguish between packets from different sources. The ID that is inserted in the packet by the FTM is `010`, which conflicts with the ID that is used by the ITM for 16-bit packets. Pack the ITM with `1`s to avoid the possibility of packet compression that can change the size of the packet to 16.

System Debug Reset Does Not Work

Answer Record 52022

If the ARM PS DAP controller is used to issue a System Debug Reset, the PS will hang.

Gigabit Ethernet

Ethernet TxDMA Might Hang

Answer Record 52019

Due to timing between the internal AHB interface and the TxDMA, the TxDMA might hang.

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 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 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 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 issue will not be fixed.

ID

Incorrect PS Family IDCODE Value

Answer Record 47317

The 7-bit family IDCODE value that software can read in the PS SLCR.IDCODE[27:21] register bits is incorrect. The register value is 7'h1D instead of the correct value of 7'h1B.

PL Configuration

PL Readback Operation Of The Bitstream Through The DevC Interface Does Not Work

Answer Record 47578

It is not possible for the PS to readback the PL through the DevC interface. This includes reading the SEU detection bit, reading the PL IDCODE, and using the interface for partial reconfiguration. Instead, use the ICAP interface that is accessible via the PL JTAG.

PS AXI Interconnect

OCM Interconnect Switch Can Experience Starvation With Heavy CPU/ACP Traffic

Answer Record 47544

With very heavy CPU and ACP interconnect traffic, it might be possible to starve memory accesses by the OCM interconnect because the CPU read and write requests have higher arbitration priorities than requests coming through the OCM interconnect. The memory traffic generated by the CPUs can be reduced by enabling the MMU/Cache mechanisms and setting the address range to Cacheable or Strongly Ordered.

Deadlock Can Occur When OCM And DDR Are Accessed By AXI_HP

Answer Record 47484

There is a possibility to have deadlock on the PS AXI interconnect when an AXI_HP source and another source, such as the DMAC, each generates accesses to both the OCM and DDR in an inter-dependent way. Deadlock can be avoided by having one of the sources make requests to only one destination (OCM or DDR) or restrict the number of outstanding writes by an AXI_HP interface to one at a time. The problematic sources include the pairs of AXI_HP interfaces and the Central Interconnect (DMAC, IOP masters, etc.). The APU memory requests (CPUs and ACP) are not problematic.

Quad-SPI

Controller In Linear Addressing Mode Might Hang In A Highly Loaded System

Answer Record 47577

In linear mode, the Quad-SPI controller might hang when there is a high volume of memory requests. This issue does not appear when the Quad-SPI controller is being used as the boot device. For other situations, the data rate through the controller can be lowered.

Quad-SPI Controller Does Not Drive HOLD_B Inactive During SPI Data Phase

Answer Record 47596

Quad-SPI devices have a dual-purpose pin: HOLD_B/DQ3. An active HOLD_B gates CLK and DIN and three-states DOUT when CS_B is active. HOLD_B/DQ3 is mapped to MIO5, which is pulled High when Quad-SPI or SD card boot mode is selected. In this case, the Quad-SPI works as expected. However, certain boot modes such as JTAG, pull MIO5 Low. If QSPI is used in these other boot modes, it leads to the assertion of HOLD_B in certain Quad-SPI protocol phases, which will hang the Quad-SPI transaction.

Quad-SPI Register LPBK_DLY_ADJ Must Be Manually Set To 0

Answer Record 52015

When the Quad-SPI I/O interface is used with its internal loopback clock, the value of the qspi.LPBK_DLY_ADJ[4:0] register must be set to 0. This field must be set explicitly by the FSBL or the user application.

SDIO

SD Capability Register Shows Wrong Max_Block_Length Value

Answer Record 47529

The SDIO controller supports a maximum block length of 512 bytes. The SDIO capability register indicates a max length of 4,096 bytes. The software driver must use the 512 byte length and ignore what is read from the capabilities register.

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 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 Software Reset for All commands when a card is inserted, or the interconnect can hang.

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

ADMA2 Read Corrupts Data That Was Written Using PIO Mode

Answer Record 47537

If the controller writes data using PIO mode and later reads the data using ADMA2 mode, the controller fails to send an Auto CMD12 (stop request) during the DMA process and the read FIFO fills up with incorrect data.

This can be avoided by selecting SDMA mode during PIO mode writes or by not using PIO mode to write data.

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

Controller's CMD17 Might Not Complete

Answer Record 52020

Under extremely rare conditions, the SDIO DMA can receive the block read data from the SD card but might not complete the DMA-to-memory transfer.

Security

PL Can Retain Configuration Through A Reset Cycle

Answer Record 47530

When the BootROM begins to execute the FSBL, the contents of the PL are not always initialized. The PL contents can retain a full or partial configuration.

This can be avoided by tying the PROGRAM_B signal High and having the FSBL set the PCFG_PROG_B bit High.

Secure Boot Features Are Not Supported

Answer Record 47565

Secure boot mode is not supported. Security can be compromised and the system might hang while trying to boot in secure mode.

JTAG Chain Is Accessible Before The BootROM Completes

Answer Record 47566

The JTAG serial path can provide access to the PS DAP and PL TAP controllers before the BootROM completes the handover of the CPU and system to the user, regardless of the security mode.

Register Initialization During BootROM Handover Does Not Error on Illegal Addresses

Answer Record 52017

The boot image can contain address-data pairs that initialize control registers during the BootROM handover to user code. Unintended access to illegal addresses is enabled.

Signals

MIO Interface Signal Logic Levels Can Be Observed by GPIOs

Answer Record 47873

All interface signals connected to the MIO pins can be snooped via the GPIO block. The snooping can be done irrespective of the MIO configuration for GPIOs.

SDIO Three-State Enable Signals On EMIO Have The Wrong Polarity

Answer Record 47874

The Unisim names for the three-state enables of the SD data and command signals on EMIO include an 'N' suffix: EMIOSDIOxDATATN and EMIOSDIOxCMDTN. This normally indicates an active-Low signal. However, these signals are active-High.

SMC

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

NAND With ECC Misses Single Bit And Some Double Bit Errors

Answer Record 47519

There is a defect in the coding of the ECC algorithm and it misses single bit errors to bit 0 of byte 0 and fails to detect some double bit error cases. Only the odd half of the parity calculation is being tested to check for pass/fail. When reading the data from NAND with the ECC enabled, in some specific cases, the results indicated in the registers are incorrect.

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

SPI PS

In Master Mode On MIO, The SPI Controller Resets Itself When The SS0 Signal Asserts

Answer Record 47511

When the SPI controller is configured as a master, the SS0 signal is an output. The unused input signal from the MIO/EMIO multiplexer must remain deasserted. When using an MIO interface, route the SS0 controller signals to the EMIO interface and assign the EMIO SS0 input signal to net_vcc.

RxFIFO Not Empty Status Is Not Updated Promptly

Answer Record 47575

There is a delay in updating the SPI 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 RX_FIFO_not_empty bit twice to allow enough time for the controller to update the status bit.

SPI Master Mode Setup Timing Is Dependent On The SPI Reference Clock Period

Answer Record 47579

When operating the SPI interface in master mode, the setup timing with respect to SCLK for MI is one spi_ref_clk period.

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

Adding A dTD To A Primed Endpoint Might Not Get Recognized

Answer Record 51121

The add dTD tripwire semaphore, usb.USBCMD [ATDTW] bit, can cause the controller to ignore a dTD that is added to a primed endpoint.

Programmable Logic (PL) Errata Details

This section provides a detailed description of each PL issue known at the release time of this document.

Configuration

Single Event Upset (SEU) Detection and Correction

Answer Record 51123

SEU Readback enabled by POST_CRC=ENABLE is not supported.

Power

Static Power Is Higher Than Reported

Answer Record 47592

The V_{CCPINT} and V_{CCAUX} supplies can exhibit up to 50% higher static current and all other power supplies can exhibit up to 25% higher static current compared to the static current reported in the Xilinx Power Estimator (XPE).

Design Tool Requirements

The devices listed in Table 1, unless otherwise specified, require the following Xilinx Design Tools:

- Speed specification v1.01 (or later) of Xilinx® ISE® Design Suite 14.1 (or later) available at: http://www.xilinx.com/support/download/
- See Zynq-7000 AP SoC Answer Record 47915 for the most current known issues and work-arounds for Xilinx Design Tools.

Traceability

Figure 1 shows an example device top mark for the devices listed in Table 1.

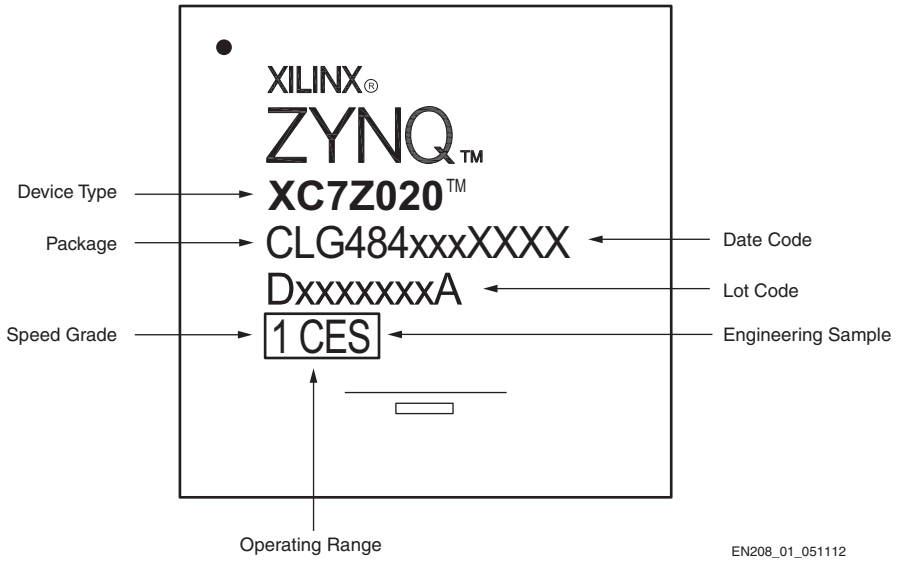

Figure 1: **Example Device Top Mark**

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
05/24/12	1.0	Initial Xilinx release.
09/20/12	1.1	Updated errata item dispositions (i.e., not fixed). Added MultiBoot Feature Is Not Supported, During NOR Boot, MIO2 And MIO14 Pins Are Inadvertently Configured By BootROM, Clocks PS, During Quad-SPI Boot, Image Search for Dual SS 8-Bit Parallel I/O Is Performed In 64 KB Steps, Quad-SPI Controller Does Not Drive HOLD_B Inactive During SPI Data Phase, and SPI Master Mode Setup Timing Is Dependent On The SPI Reference Clock Period. Updated ADMA2 Burst Transactions Alignment And Length Requirements (heading and content) and SPI Master Mode Setup Timing Is Dependent On The SPI Reference Clock Period. Removed *XADC Specification Deviates From Data Sheet*; specs have been updated in DS187, Zynq-7000 (XC7Z010 and XC7Z020) Data Sheet: DC and Switching Characteristics (v1.1) on June 27, 2012.
11/01/12	1.2	Split the Boot section into two new sections: Boot IOP and Boot System. Added Read Gate Training Value Is Unreliable In Slice 3, System Debug Reset Does Not Work, Quad-SPI Register LPBK_DLY_ADJ Must Be Manually Set To 0, Controller's CMD17 Might Not Complete, Register Initialization During BootROM Handover Does Not Error on Illegal Addresses, ADC Accuracy, Micron 8 Gb (On-Die ECC) NAND Devices Do Not Work, BootROM Stops Searching For Boot Image After First 16 MB For Quad-SPI x8 and SRAM/NOR, SD Card Boot Mode Runs At Low Frequency And 1-Bit Data Width, Quad-SPI MIO Pin 8 Is Inadvertently Enabled During Boot, SD Card Boot Mode Tests For Card Detect On MIO Pin 0, Independent JTAG Is Not Supported In JTAG Boot Mode, INIT_B Pin Does Not Indicate Boot Error Status, PS_SRST_B Asserted During A POR Boot Does Not Result In A Secure Lockdown, and Ethernet TxDMA Might Hang.
12/18/12	1.3	Added Strongly Ordered Write Followed By LDREX Might Deadlock Processor; SD Card Boot Mode Does Not Comply With The 74 CLK Cycle Requirement Between The Start Of The SD CLK And First Command; and Adding A dTD To A Primed Endpoint Might Not Get Recognized.
06/04/13	1.4	Updated disposition statements throughout document. Updated Processor System (PS) Errata Details and Software Reset Sequence For SDIO Can Hang The Interconnect. Added 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; A Spurious Event 0x63, “STREX Passed,” Might Be Reported On An LDREX Instruction; 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; DBGPRSR Sticky Reset Status Bit Is Set To 1 By The CPU Debug Reset Instead Of By The CPU Non-Debug Reset; Reset Reason Mechanism Does Not Use slcr.REBOOT_STATUS Register; Unicast And Broadcast Pause Frames Received By The Controller Are Not Filtered Out, Back-off Time Is More Aggressive Than The Standard Requirement, Packets Up To 1,536 Bytes Are Allowed With VLAN Tagging, Receive Path Lock-Up Might Occur When A Large Number Of Receive Resource Errors Are Generated; and BootROM 128KB CRC Self-Check Is Not Supported. Removed ADC Accuracy because the specifications are now documented in DS187, *Zynq-7000 All Programmable SoC (XC7Z010 and XC7Z020): DC and AC Switching Characteristics*, v1.3, February 11, 2013.

Notice of Disclaimer

The information disclosed to you hereunder (the “Materials”) is provided solely for the selection and use of Xilinx products. To the maximum extent permitted by applicable law: (1) Materials are made available “AS IS” and with all faults, Xilinx hereby DISCLAIMS ALL WARRANTIES AND CONDITIONS, EXPRESS, IMPLIED, OR STATUTORY, INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, OR FITNESS FOR ANY PARTICULAR PURPOSE; and (2) Xilinx shall not be liable (whether in contract or tort, including negligence, or under any other theory of liability) for any loss or damage of any kind or nature related to, arising under, or in connection with, the Materials (including your use of the Materials), including for any direct, indirect, special, incidental, or consequential loss or damage (including loss of data, profits, goodwill, or any type of loss or damage suffered as a result of any action brought by a third party) even if such damage or loss was reasonably foreseeable or Xilinx had been advised of the possibility of the same. Xilinx assumes no obligation to correct any errors contained in the Materials or to notify you of updates to the Materials or to product specifications. You may not reproduce, modify, distribute, or publicly display the Materials without prior written consent. Certain products are subject to the terms and conditions of the Limited Warranties which can be viewed at http://www.xilinx.com/warranty.htm; IP cores may be subject to warranty and support terms contained in a license issued to you by Xilinx. Xilinx products are not designed or intended to be fail-safe or for use in any application requiring fail-safe performance; you assume sole risk and liability for use of Xilinx products in Critical Applications: http://www.xilinx.com/warranty.htm#critapps.

Engineering Sample Disclaimer

ENGINEERING SAMPLE (ES) DEVICES ARE MADE AVAILABLE SOLELY FOR PURPOSES OF RESEARCH, DEVELOPMENT AND PROTOTYPING. ALL ES DEVICES ARE SOLD “AS-IS” WITH NO WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED. XILINX DOES NOT WARRANT THAT ES DEVICES ARE FULLY VERIFIED, TESTED, OR WILL OPERATE IN ACCORDANCE WITH DATA SHEET SPECIFICATIONS. XILINX DISCLAIMS ANY OBLIGATIONS FOR TECHNICAL SUPPORT AND BUG FIXES. XILINX SHALL NOT BE LIABLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION DIRECT, INDIRECT, INCIDENTAL, SPECIAL, RELIANCE, OR CONSEQUENTIAL DAMAGES ARISING FROM OR IN CONNECTION WITH THE USE OF ES DEVICES IN ANY MANNER WHATSOEVER, EVEN IF XILINX HAS BEEN ADVISED OF THE POSSIBILITY THEREOF. XILINX MAKES NO REPRESENTATION THAT ES DEVICES PROVIDE ANY PARTICULAR FUNCTIONALITY, OR THAT ES DEVICES WILL MEET THE REQUIREMENTS OF A PARTICULAR USER APPLICATION. XILINX DOES NOT WARRANT THAT ES DEVICES ARE ERROR-FREE, NOR DOES XILINX MAKE ANY OTHER REPRESENTATIONS OR WARRANTIES, WHETHER EXPRESS OR IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. THE FOREGOING STATES THE ENTIRE LIABILITY OF XILINX WITH RESPECT TO ES DEVICES.