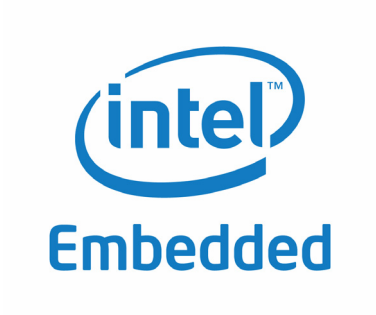


Intel® Xeon® Processor C5500/ C3500 Series

Specification Update

April 2012

Notice: The Intel® Xeon® Processor C5500/C3500 Series may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Order Number: 323105, Revision: -020

Contents

Revision History

Issue	Revision	Description
February 2010	001	Initial release.
March 2010	002	Added Errata BF114-BF129
April 2010	003	Corrected document title Added BF130 and BF131 Modified errata titles (not noted by change bars)
May 2010	004	Updated text in BF113. Added errata BF132 to BF137 Added document changes 1 - 7.
June 2010	005	Added errata BF138 and BF139 Modified document change 6 . Added document changes 8-12.
July 2010	006	Added errata BF140 through BF154. Added document changes 13-14. Deleted Table 5: Specification Update Key
August 2010	007	Added document changes 15-21 .
September 2010	008	Added errata BF155. Added document changes 22-24. Added specification changes 1-2. Added specification clarification 1.
October 2010	009	Added errata BF156-BF157 Updated erratum BF41 Added document change 25 Updated document change 5 and 6.
November 2010	010	Added erratum BF158.
December 2010	011	No updates
January 2011	012	Added erratum BF159-BF163
February 2011	013	Added erratum BF164-BF167
March 2011	014	No updates
May 2011	015	Added erratum BF168
July 2011	016	Added erratum BF169
September 2011	017	Added erratum BF170 Updated Microcode Updates table
October 2011	018	Updated erratum BF160 to change publication volume number in Workaround.
January 2012	019	
April 2012	020	Added erratum BF171

Introduction

Purpose/Scope/Audience

This document is an update to the specifications listed in the Related Documents table that follows. This document is a compilation of Intel® Xeon® Processor C5500/C3550 Series Errata, Specification Changes, Specification Clarifications, and Document Changes. It is intended for hardware and software system designers and manufacturers as well as developers of applications, operating systems, or tools.

This document may also contain information that was not previously published.

Note: Storage customers: Contact your field representative for additional specification update information that applies to the storage use of this product.

Table 1. Affected Documents

Title	Document Number
Intel® Xeon® Processor C5500/C3500 Series Datasheet Volume 1	323103
Intel® Xeon® Processor C5500/C3500 Series Datasheet, Volume 2	323317

Table 2. Related Documents

Title	Document Number / Location
Intel® Xeon® Processor C5500/C3500 Series Thermal/ Mechanical Design Guide	323107
AP-485, Intel® Processor Identification and CPUID Instruction	http://www.intel.com/design/processor/applnots/241618.htm
Intel® 64 and IA-32 Architecture Software Developer's Manual • Volume 1: Basic Architecture • Volume 2A: Instruction Set Reference Manual A-M • Volume 2B: Instruction Set Reference Manual N-Z • Volume 3A: System Programming Guide • Volume 3B: System Programming Guide	http://www.intel.com/products/processor/manuals/index.htm
Intel® 64 and IA-32 Architecture Optimization Reference Manual	http://www.intel.com/products/processor/manuals/index.htm
Intel® 64 and IA-32 Architecture Software Developer's Manual Documentation Changes	http://www.intel.com/design/processor/specupdt/252046.htm

Conventions and Terminology

Table 3. Conventions and Terminology

Term	Definition
Document Changes	Document Changes are changes to an Intel Parent Specification that result in changes only to an Intel customer document but no changes to a specification or to a parameter for an Intel product. An example of a document-only change is the correction of a typographical error.
Intel® Xeon® Processor C5500/ C3550 Series Errata	Intel® Xeon® Processor C5500/C3550 Series Errata are design defects or errors. These may cause the C5500/C3500 Series's behavior to deviate from published specifications. Hardware and software designed to be used with any given stepping must assume that all errata documented for that stepping are present on all devices.
Parent Specification	A parent specification is a top-level specification from which other documents can be derived, depending on the product or platform. Typically, a parent specification includes a product's pinout, architectural overview, device operation, hardware interface, or electrical specifications. Examples of parent specifications include the following: Datasheet, Developer's Manual, Technical Product Specification (also known as "TPS"). The derived documents may be used for purposes other than that for which the parent specification is used.
QDF Number	The QSF Number is a four-digit code used to distinguish between engineering samples. These samples are used for qualification and early design validation. The functionality of these parts can range from mechanical-only to fully functional. This document has a processor identification information table that lists the QDF numbers and the corresponding product details.
S-Spec Number	The S-Spec Number is a five-digit code used to identify products. Products are differentiated by their unique characteristics. E.g. core speed, L2 cache size, package type, etc., as described in the processor identification table. Read the notes associated with each S-Spec number.
Specification Changes	Specification Changes are the result of adding, removing, or changing a feature, after which an Intel product subsequently operates differently than specified in an Intel Parent Specification, but typically the customer does not have to do anything to achieve proper device functionality as a result of Intel adding, removing, or changing a feature.
Specification Clarifications	Specification Clarifications are changes to a document that arise when an Intel Parent Specification must be reworded so that the specification is either more clear or not in conflict with another specification.

Note: Errata remain in the Specification Update throughout the product's lifecycle, or until a particular stepping is no longer commercially available. Under these circumstances, errata removed from the Specification Update are archived and available upon request. Specification Changes, Specification Clarifications and Document Changes are removed from the Specification Update when the appropriate changes are made to the appropriate product specification or user documentation (datasheets, manuals, etc.).

Summary Tables of Current Product Issue Activity

Table 5 through Table 8 indicate the Intel® Xeon® Processor C5500/C3550 Series Errata, Specification Changes, Specification Clarifications, or Document Changes that apply to the Intel® Xeon® Processor C5500/C3500 Series. Intel may fix some of the Errata in a future stepping of the component as noted in Table 5 or account for the other outstanding issues through Specification Changes, Specification Clarifications, or Document Changes. Table 5 through Table 8 use the codes listed in Table 4.

Table 4. Codes Used in Summary Tables

Code	Column	Definition
X	Stepping	Indicates either that, for the stepping/revision listed: • an erratum exists and is not yet fixed • a specification change or specification clarification applies
No mark or blank	Stepping	Indicates either that, for the stepping/revision listed: • an erratum is fixed • a specification change or specification clarification does not apply
Plan Fix	Status	This erratum may be fixed in a future stepping/revision.
Fixed	Status	This erratum has been previously fixed.
No Fix	Status	There are no plans to fix this erratum.
A change bar to the left of a table row indicates an item that is either new or modified from the previous version of the Specification Update document.		

Table 5. Summary Table of Changes (Sheet 1 of 7)

Number	Steppings	Status	Errata
	B-0		
BF1.	X	No Fix	The Processor Reports #TS Instead Of #GP Fault
BF2.	X	No Fix	REP MOVS/STOS Executing with Fast Strings Enabled And Crossing Page Boundaries With Inconsistent Memory Types Use Incorrect Data Size Or Lead To Memory-Ordering Violations
BF3.	X	No Fix	Code Segment Limit/Canonical Faults On RSM Serviced Before Higher Priority Interrupts/Exceptions And Pushes Wrong Address Onto Stack
BF4.	X	No Fix	Performance Monitor SSE Retired Instructions Return Incorrect Values
BF5.	X	No Fix	Premature Execution Of Load Operation Prior To Exception Handler Invocation
BF6.	X	No Fix	MOV To/From Debug Registers Causes Debug Exception
BF7.	X	No Fix	Incorrect Address Computed For Last Byte Of FXSAVE/FXRSTOR Image Leads To Partial Memory Update
BF8.	X	No Fix	Values for LBR/BTS/BTM Incorrect After Exit from SMM

Table 5. Summary Table of Changes (Sheet 2 of 7)

Number	Steppings	Status	Errata
	B-0		
BF9.	X	No Fix	Single Step Interrupts With Floating Point Exception Pending Mishandled
BF10.	X	No Fix	Fault On ENTER Instruction Results In Unexpected Values On Stack Frame
BF11.	X	No Fix	IRET Causes Unexpected Alignment Check Exception
BF12.	X	No Fix	General Protection Fault (#GP) for Instructions Greater Than 15 Bytes Preempted
BF13.	X	No Fix	General Protection (#GP) Fault Not Signaled On Data Segment Limit Violation Above 4-G Limit
BF14.	X	No Fix	LBR, BTS, BTM May Report Wrong Address When Exception/ Interrupt Occurs In 64-bit Mode
BF15.	X	No Fix	MONITOR Or CLFLUSH On Local XAPIC Address Space Results In Hang
BF16.	X	No Fix	Corruption Of CS Segment Register During RSM While Transitioning From Real Mode To Protected Mode
BF17.	X	No Fix	Performance Monitoring Events For Read Miss to Level 3 Cache Fill Occupancy Counter Incorrect
BF18.	X	No Fix	VM Exit On MWAIT Incorrectly Reports Monitoring Hardware As Armed
BF19.	X	No Fix	Delivery Status Of LINT0 Register Of Local Vector Table Lost
BF20.	X	No Fix	Performance Monitor Event SEGMENT_REG_LOADS Counts Inaccurately
BF21.	X	No Fix	#GP On Segment Selector Descriptor that Straddles Canonical Boundary May Not Provide Correct Exception Error Code
BF22.	X	No Fix	Improper Parity Error Signaled In IQ Following Reset When Code Breakpoint Set On #GP Instruction
BF23.	X	No Fix	An Enabled Debug Breakpoint or Single Step Trap May Be Taken after MOV SS/POP SS Instruction if it is Followed by an Instruction That Signals a Floating Point Exception
BF24.	X	No Fix	IA32_MPERF Counter Stops Counting During On-Demand TM1
BF25.	X	No Fix	Intel® QuickPath Memory Controller tTHROT_OPREF Timings May be Violated During Self Refresh Entry
BF26.	X	No Fix	Processor Over Counts Correctable Cache MESI State Errors
BF27.	X	No Fix	Synchronous Reset of IA32_APERF/IA32_MPERF Counters On Overflow Does Not Work
BF28.	X	No Fix	Disabling Thermal Monitor While Processor Hot, Then Re-enabling Results In Stuck Core Operating Ratio
BF29.	X	No Fix	PECI Does Not Support PCI Configuration Reads/Writes To Misaligned Addresses
BF30.	X	No Fix	OVER Bit For IA32_MCi_STATUS Register Set On Specific Internal Error
BF31.	X	No Fix	Writing Local Vector Table (LVT) When Interrupt is Pending Causes Unexpected Interrupt
BF32.	X	No Fix	Faulting MMX Instruction Incorrectly Updates x87 FPU Tag Word
BF33.	X	No Fix	xAPIC Timer Decrements Too Quickly Following Automatic Reload While In Periodic Mode
BF34.	X	No Fix	Reported Memory Type May Not Be Used To Access the VMCS And Referenced Data Structures

Table 5. Summary Table of Changes (Sheet 3 of 7)

Number	Steppings	Status	Errata
	B-0		
BF35.	X	No Fix	B0-B3 Bits In DR6 For Non-Enabled Breakpoints May be Incorrectly Set
BF36.	X	No Fix	Core C6 Clears Previously Logged TLB Errors
BF37.	X	No Fix	Performance Monitor Event MISALIGN_MEM_REF May Over Count
BF38.	X	No Fix	Changing Memory Type for In-Use Page Translation Leads To Memory-Ordering Violations
BF39.	X	No Fix	Infinite Stream of Interrupts Occur If ExtINT Delivery Mode Interrupt Received While All Cores In C6
BF40.	X	No Fix	Two xAPIC Timer Event Interrupts Unexpectedly Occur
BF41.	X	No Fix	EOI Transaction May Not be Sent if Software Enters Core C6 During an Interrupt Service Routine
BF42.	X	No Fix	FREEZE_WHILE_SMM Does Not Prevent Event From Pending PEBS During SMM
BF43.	X	No Fix	APIC Error "Received Illegal Vector" Lost
BF44.	X	No Fix	DR6 May Contain Incorrect Information When the First Instruction After a MOV SS, r/m or POP SS is a Store
BF45.	X	No Fix	Uncorrectable Error Logged In IA32_CR_MC2_STATUS Results In System Hang
BF46.	X	No Fix	IA32_PERF_GLOBAL_CTRL MSR May be Incorrectly Initialized
BF47.	X	No Fix	ECC Errors Cannot Be Injected On Back-to-Back Writes
BF48.	X	No Fix	Performance Monitor Interrupts Generated From Uncore Fixed Counters (394H) May be Ignored
BF49.	X	No Fix	Performance Monitor Counter INST_RETIRED.STORES May Count Higher than Expected
BF50.	X	No Fix	Sleeping Cores May Not be Woken Up on Logical Cluster Mode Broadcast IPI Using Destination Field Instead of Shorthand
BF51.	X	No Fix	Faulting Executions of FXRSTOR Update State Inconsistently
BF52.	X	No Fix	Performance Monitor Event EPT.EPDPE_MISS Counted While EPT is Disabled
BF53.	X	No Fix	Memory Aliasing Of Code Pages Causes Unpredictable Behavior
BF54.	X	No Fix	Performance Monitor Counters May Count Incorrectly
BF55.	X	No Fix	Performance Monitor Event Offcore_Response_0 (B7H) Does Not Count NT Stores To Local DRAM Correctly
BF56.	X	No Fix	EFLAGS Discrepancy On Page Faults and EPT-Induced VM Exits After Translation Change
BF57.	X	No Fix	System Hang if MC_CHANNEL_{0,1}_MC_DIMM_INIT_CMD.DO_ZQCL Commands Not Issued In Increasing Populated DDR3 Rank Order
BF58.	X	No Fix	Back-To-Back Uncorrected Machine Check Errors Overwrite IA32_MC3_STATUS.MSCOD
BF59.	X	No Fix	Memory Intensive Workloads With Core C6 Transitions Cause System Hang
BF60.	X	No Fix	Corrected Errors With Yellow Error Indication Overwritten By Other Corrected Errors
BF61.	X	No Fix	Performance Monitor Events DCACHE_CACHE_LD and DCACHE_CACHE_ST Over Count
BF62.	X	No Fix	Rapid Core C3/C6 Transitions May Cause Unpredictable System Behavior

Table 5. Summary Table of Changes (Sheet 4 of 7)

Number	Steppings	Status	Errata
	B-0		
BF63.	X	No Fix	Performance Monitor Events INSTR_RETIRED and MEM_INST_RETIRED May Count Inaccurately
BF64.	X	No Fix	Page Fault Not Generated When PS Bit set to "1" in PML4E or PDPTE
BF65.	X	No Fix	CPURESET Bit Does Not Get Cleared
BF66.	X	No Fix	PHOLD Disable In MISCCTRLSTS Register Does Not Work
BF67.	X	No Fix	PCIe PMCSR Power State Field Incorrectly Allows Requesting D1 and D2 Power States
BF68.	X	No Fix	Concurrent Updates to a Segment Descriptor May be Lost
BF69.	X	No Fix	PMIs May be Lost During Core C6 Transitions
BF70.	X	No Fix	Uncacheable Access To Monitored Address Range Prevent Future Triggering Of Monitor Hardware
BF71.	X	No Fix	BIST Results Additionally Reported After GETSEC[WAKEUP] or INIT-SIPI Sequence
BF72.	X	No Fix	Pending x87 FPU Exceptions (#MF) Signaled Earlier Than Expected
BF73.	X	No Fix	VM Exits Due to "NMI-Window Exiting" May Be Delayed by One Instruction
BF74.	X	No Fix	Malformed PCIe Packet Generated Under Heavy Outbound Load
BF75.	X	No Fix	VM Exits Due to EPT Violations Do Not Record Information About Pre-IRET NMI Blocking
BF76.	X	No Fix	S1 Entry May Cause Cores to Exit C3 or C6 C-State
BF77.	X	No Fix	Multiple Performance Monitor Interrupts Possible On Overflow Of IA32_FIXED_CTR2
BF78.	X	No Fix	LBRs Not Initialized During Processor Power-On Reset
BF79.	X	No Fix	Unexpected Interrupts May Occur on C6 Exit If Using APIC Timer to Generate Interrupts
BF80.	X	No Fix	Package C6 Exit With Memory In Self-Refresh When Using DDR3 RDIMM Memory Leads to Hang
BF81.	X	No Fix	LBR, BTM or BTS Records Have Incorrect Branch From Information After EIST Transition, T-states, C1E, Or Adaptive Thermal Throttling
BF82.	X	No Fix	PECI GetTemp() Reads May Return Invalid Temperature Data in Package C6 State
BF83.	X	No Fix	VMX-Preemption Timer Does Not Count Down At Rate Specified
BF84.	X	No Fix	Multiple Performance Monitor Interrupts Possible On Overflow Of Fixed Counter 0
BF85.	X	No Fix	SVID and SID Of Devices 8 And 16 Only Implement Bits [7:0]
BF86.	X	No Fix	No_Soft_Reset Bit In PMCSR Does Not Operate As Expected
BF87.	X	No Fix	VM Exits Due To LIDT/LGDT/SIDT/SGDT Do Not Report Operand Size
BF88.	X	No Fix	Performance Monitoring Events STORE_BLOCKS.NOT_STA and STORE_BLOCKS.STA May Not Count Events Correctly
BF89.	X	No Fix	Storage Of PEBS Record Delayed Following Execution Of MOV SS Or STI
BF90.	X	No Fix	Performance Monitoring Event FP_MMX_TRANS_TO_MMX Not Counting Some Transitions
BF91.	X	No Fix	INVLPG Following INVEPT Or INVVPID Fail to Flush All Translations For Large Page

Table 5. Summary Table of Changes (Sheet 5 of 7)

Number	Steppings	Status	Errata
	B-0		
BF92.	X	No Fix	PECI Bus Tri-Stated After System Reset
BF93.	X	No Fix	LER MSRs May Be Unreliable
BF94.	X	No Fix	Multiple ECC Errors Result in Logging Incorrect Syndrome
BF95.	X	No Fix	MCi_Status Overflow Bit Incorrectly Set On Single Instance Of DTLB Error
BF96.	X	No Fix	Debug Exception Flags DR6.B0-B3 Flags Incorrect For Disabled Breakpoints
BF97.	X	No Fix	Intel® QuickPath Memory Controller Hangs If Uncorrectable ECC Errors Occur On Both Channels In Mirror Channel Mode
BF98.	X	No Fix	Simultaneous Correctable ECC Errors on Different Memory Channels With Patrol Scrubbing Enabled May Result in Incorrect Information Being Logged
BF99.	X	No Fix	In Memory Lockstep Mode Per DIMM Correctable ECC Errors Logged Incorrectly
BF100.	X	No Fix	Memory Controller Address Parity Error Injection Does Not Work Correctly
BF101.	X	No Fix	Failing DIMM ID Incorrect In 2DPC Configuration When Mirroring Enabled
BF102.	X	No Fix	ISSUEONCE Bit In MC_SCRUB_CONTROL Register Not Working Correctly
BF103.	X	No Fix	Memory Thermal Throttling May Not Work as Expected in Lockstep Channel Mode
BF104.	X	No Fix	A String Instruction that Re-maps a Page May Encounter an Unexpected Page Fault
BF105.	X	No Fix	Transactions To Address Above TOCM Not Setting Master Abort Error Bit In IIOERRST
BF106.	X	No Fix	HPA_LIMIT Check Violations Not Logged As Error
BF107.	X	No Fix	Memory Writes To Certain Address Range Considered Advisory Non-Fatal
BF108.	X	No Fix	PCIe Packet Detect Power Saving Mode Causes Packet Loss Or Corruption
BF109.	X	No Fix	Intel® VT-d Translated Write Transactions Targeting Interrupt Address Range Blocked But Not Recorded as Errors
BF110.	X	No Fix	ERRSID Not Logging ReqID For Inbound PCIe Error Messages
BF111.	X	No Fix	MSI With Greater Than One DWord Payload Not Logged As Error In Proper XPUNCERRSTS Register
BF112.	X	No Fix	Error Pin Status Register For Pins SYS_ERR_STAT[2:0] Not Updated In Mode 01b
BF113.	X	No Fix	Writes to SDOORBELL or B2BDOORBELL in Conjunction With Inbound Access to NTB MMIO Space May Hang System
BF114.	X	No Fix	Enabling Demand/Patrol Scrubs Along With WMM May Cause Unpredictable System Behavior
BF115.	X	No Fix	2MB Page Split Lock Accesses Combined With Complex Internal Events May Cause Unpredictable System Behavior
BF116.	X	No Fix	L1 Cache Uncorrected Errors May be Recorded as Correctable in 16K Mode
BF117.	X	No Fix	Remote MEMRD Peer-to-Peer PCIe Transactions with Non-Contiguous Byte Enables May Return UnExpected Data
BF118.	X	No Fix	NTB Endpoint Does Not Implement LNKCON2 or LNKSTS Registers

Table 5. Summary Table of Changes (Sheet 6 of 7)

Number	Steppings	Status	Errata
	B-0		
BF119.	X	No Fix	Source ID for Errors Internally Detected by PCIe Devices 3, 4, 5, 6 is Logged Incorrectly
BF120.	X	No Fix	Intel® VT-d: Address Remapping Error When DMA/Interrupt Remapping is Active
BF121.	X	No Fix	PCIe Link Bit Errors Present During L0s Entry May Cause the System to Hang During L0s Exit
BF122.	X	No Fix	Electrical Idle Exit Sequence Incorrect Upon Exit From PCIe L0s State
BF123.	X	No Fix	PCIe AER HDRLOG Register Does Not Record Header of Packets with Errors
BF124.	X	No Fix	Aborted Inbound PCIe Memory Reads Return Incorrect Lower Address Field in the Completion
BF125.	X	No Fix	Clearing the Memory_Space_Enable Bit For NTB Does not Disable Accesses to All MMIO Regions
BF126.	X	No Fix	PCIe Slave Loopback Mode May Exit Prematurely
BF127.	X	No Fix	FP Data Operand Pointer May Be Incorrectly Calculated After an FP Access Which Wraps a 4-Gbyte Boundary in Code That Uses 32-Bit Address Size in 64-Bit Mode
BF128.	X	No Fix	PCIe Ports Violate TTX-Rise-Fall Specification When Operating at 2.5GT/s
BF129.	X	No Fix	NTB/RP Link Will Send Extra TS2 Ordered Set During Link Training
BF130.	X	No Fix	PECI PCIConfigRd() Followed by a GetTemp() May Cause System Hang in Package C6 State
BF131.	X	No Fix	IO_SMI Indication in SMRAM State Save Area May Be Lost
BF132.	X	No Fix	TSC Values When Observed Cross-Socket May Be Out of Sync After a Warm Reset
BF133.	X	No Fix	Electromechanical Interlock Control Not Functioning Correctly on Hot Plug SMBus
BF134.	X	No Fix	PCIe Port's LTSSM May Not Transition Properly in the Presence of TS1 or TS2 Ordered Sets That Have Unexpected Symbols Within those Sets
BF135.	X	No Fix	NTB Secondary LNKCAP Register Does not Reflect Programmed Link Capabilities
BF136.	X	No Fix	Using Intel® VT-d with NTB in Certain Platform Configurations Can Alias Requester ID's and Cause Unexpected Behavior
BF137.	X	No Fix	PCIe Squelch Detect May be Slow to Respond During L0s Entry and May Cause a Surprise Link Down Condition
BF138.	X	No Fix	Ports May Not Enter Slave Loopback Mode From the Configuration LTSSM State
BF139.	X	No Fix	Multiple UC Errors Reported Via MSI May Result in Lost Interrupts
BF140.	X	No Fix	Performance Monitor Events for Hardware Prefetches Which Miss The L1 Data Cache May be Over Counted
BF141.	X	No Fix	DP System Using Package C3 or C6 Power States May Record Spurious Poisoned Packet Errors
BF142.	X	No Fix	Revision ID For Non-Legacy Processor is Incorrect
BF143.	X	No Fix	NTB Operating In NTB/RP Mode With MSI/MSI-X Interrupts May Cause System Hang
BF144.	X	No Fix	NTB May Incorrectly Set MSI or MSI-X Interrupt Pending Bits
BF145.	X	No Fix	NTB/RP Completer and Requester ID May be Unreliable

Table 5. Summary Table of Changes (Sheet 7 of 7)

Number	Steppings	Status	Errata
	B-0		
BF146.	X	No Fix	NTB Does Not Set PME_TO_ACK After a PME_TURN_OFF Request
BF147.	X	No Fix	Poisoned Write Caused by an Internal Parity Error Targeting IIO PCI Configuration Registers or MMIO Space will Not be Suppressed
BF148.	X	No Fix	In-flight DMA Requests Received During the Implicit DMA Draining Window When Enabling Intel® VT-d Hardware Will Result in a Spurious DMA Fault
BF149.	X	No Fix	Writes to B2BSPAD[15:0] Registers May Transfer Corrupt Data Between NTB Connected Systems
BF150.	X	No Fix	Unable to Clear Received PME_TO_ACK in NTB
BF151.	X	No Fix	Using Intel® VT-d With IIO Legacy PCI Interrupts Targeting the PCH I/OxAPIC May Result in a System Hang
BF152.	X	No Fix	Using I/O Peer-to-Peer Write Traffic Across an NTB May Lead to a Hang
BF153.	X	No Fix	VM Exit May Incorrectly Clear IA32_PERF_GLOBAL_CTRL [34:32]
BF154.	X	No Fix	MCTP PCIe Messages Not Silently Discarded by Non-Legacy Processor
BF155.	X	No Fix	NTB Secondary Link Disable Control Not Functional
BF156.	X	No Fix	Unexpected DMI/ESI and PCIe Link Retraining and Correctable Errors Reported
BF157.	X	No Fix	QPI Lane May Be Dropped During Full Frequency Deskew Phase of Training
BF158.	X	No Fix	VM Entries that Return from SMM May Incorrectly Write to the SMRR Protected Region
BF159.	X	No Fix	PerfMon Overflow Status May Remain Always Set After Certain Conditions Have Occurred
BF160.	X	No Fix	An Unexpected Page Fault or EPT Violation May Occur After Another Logical Processor Creates a Valid Translation for a Page
BF161.	X	No Fix	L1 Data Cache Errors May be Logged With Level Set to 1 Instead of 0
BF162.	X	No Fix	Stack Pushes May Not Occur Properly for Events Delivered Immediately After VM Entry to 16-Bit Software
BF163.	X	No Fix	A Logical Processor May Wake From Shutdown Mode When Branch-Trace Messages Are Enabled
BF164.	X	No Fix	PerfMon Event LOAD_HIT_PRE.SW_PREFETCH May Overcount
BF165.	X	No Fix	Successive Fixed Counter Overflows May be Discarded
BF166.	X	No Fix	#GP May be Signaled When Invalid VEX Prefix Precedes Conditional Branch Instructions
BF167.	X	No Fix	Task Switch to a TSS With an Inaccessible LDTR Descriptor May Cause Unexpected Faults
BF168.	X	No Fix	VM Entries That Return From SMM Using VMLAUNCH May Not Update The Launch State of the VMCS
BF169.	X	No Fix	VM Entry May Clear Bytes 81H-83H on Virtual-APIC Page When "Use TPR Shadow" Is 0
BF170.	X	No Fix	A First Level Data Cache Parity Error May Result in Unexpected Behavior
BF171.	X	No Fix	Dual Processor Systems Running with DCA Enabled May Cause the System to Hang

Table 6. Specification Changes

No.	Document Title	Specification Changes
1.	• Intel® Xeon® Processor C5500/C3500 Series Datasheet, Volume 1	Changes for SMB_CLK, SMB_DATA, PE_HP_DATA, PE_HP_CLK
2.	• Intel® Xeon® Processor C5500/C3500 Series Datasheet, Volume 1	Specification Changing on DP_SYNCRST#, PM_SYNC, DDR_ADR, PE_GEN2_DISABLE#, PE_CFG[2:0], PE_NTBXL, DMI_PE_CFG#, and EXTSYSTRG

Table 7. Specification Clarifications

No.	Document Title	Specification Clarifications
1.	• Intel® Xeon® Processor C5500/C3500 Series Datasheet, Volume 1	Table 161, Signals with On-Die Termination (ODT) is Being Clarified

Table 8. Document Changes (Sheet 1 of 2)

No.	Document Title	Document Changes
1.	• Intel® Xeon® Processor C5500/C3500 Series Datasheet, Volume 1	The Text for Section 4.2.1, "Intel® QuickData Technology" has Changed.
2.	• Intel® Xeon® Processor C5500/C3500 Series Datasheet, Volume 1	ERR[2:0] has Changed to SYS_ERR_STAT[2:0] in Several Places in the Text.
3.	• Intel® Xeon® Processor C5500/C3500 Series Datasheet, Volume 1	The Description for the PE_CFG[2:0] Signal in Table 140, "PCI Express* Signals" has Changed.
4.	• Intel® Xeon® Processor C5500/C3500 Series Datasheet, Volume 1	The Vtta and Vttd Minimum and Maximum Values in Table 161, "Processor Absolute Minimum and Maximum Ratings" has Changed.
5.	• Intel® Xeon® Processor C5500/C3500 Series Datasheet, Volume 1	The VID Maximum Value in Table 162, "Voltage and Current Specifications" has Changed.
6.	• Intel® Xeon® Processor C5500/C3500 Series Datasheet, Volume 1	The Row that Includes the VTT Symbol in Table 162, "Voltage and Current Specifications" has Been Changed
7.	• Intel® Xeon® Processor C5500/C3500 Series Datasheet, Volume 1	The TDP for SKU P1053 in Table 162, "Voltage and Current Specifications" has Changed.
8.	• Intel® Xeon® Processor C5500/C3500 Series Datasheet, Volume 1	Register Description for MC_CLOSED_LOOP[2:0] has Changed
9.	• Intel® Xeon® Processor C5500/C3500 Series Datasheet, Volume 1	Table 70. Inbound PCI Express Messages Supported has Changed
10.	• Intel® Xeon® Processor C5500/C3500 Series Datasheet, Volume 1 •	The POC Settings for MSID[2:0] POC Bits Allocation has Changed
11.	• Intel® Xeon® Processor C5500/C3500 Series Datasheet, Volume 1	Text Deleted from Section 11.3.3.7.1, Feature Requirements

Table 8. Document Changes (Sheet 2 of 2)

No.	Document Title	Document Changes
12.	• Intel® Xeon® Processor C5500/C3500 Series Datasheet, Volume 1	Text Deleted from 179, Feature Requirements
13.	• Intel® Xeon® Processor C5500/C3500 Series Datasheet, Volume 1	Deleted
14.	• Intel® Xeon® Processor C5500/C3500 Series Datasheet, Volume 1	Text Changed and Deleted from Table 155, Processor Power Supply Voltages
15.	• Intel® Xeon® Processor C5500/C3500 Series Datasheet, Volume 1	Text Changed and Deleted in Section 3.1, Introduction
16.	• Intel® Xeon® Processor C5500/C3500 Series Datasheet, Volume 1	Text Added in Section 3.2.1, Features Not Supported on the Intel® Xeon® Processor C5500/C3500 Series NTB
17.	• Intel® Xeon® Processor C5500/C3500 Series Datasheet, Volume 1	Text Deleted fom Section 3.6.6.1, Direct Address Translation
18.	• Intel® Xeon® Processor C5500/C3500 Series Datasheet, Volume 2	Text Added to Section 3.6.7.27, IIONFERRHD: IIO Core Non-Fatal FERR Header
19.	• Intel® Xeon® Processor C5500/C3500 Series Datasheet, Volume 2	Text Added to Section 3.6.7.24, IIOFFERRHD: IIO Core Fatal FERR Header
20.	• Intel® Xeon® Processor C5500/C3500 Series Datasheet, Volume 2	Register Added: LA_DPCNTRL: Lock Arbiter Dependent Port Control
21.	• Intel® Xeon® Processor C5500/C3500 Series Datasheet, Volume 2	Text Added to Section 3.6.7.37, MINFERRHD: Miscellaneous Local Non-Fatal FERR Header
22.	• Intel® Xeon® Processor C5500/C3500 Series Datasheet, Volume 2	Deleted.
24.	• Intel® Xeon® Processor C5500/C3500 Series Datasheet, Volume 2	New Section Added Before Section 3.8, Outbound Transactions
25.	• Intel® Xeon® Processor C5500/C3500 Series Datasheet, Volume 1	Deleted Section 13.4
27.	•	Updated Table 24-7 in Platform Design Guide
28.	• Intel® Xeon® Processor C5500/C3500 Series Datasheet, Volume 2	Text Change in Table 158, VTT Voltage Identification Definition Table
29.	• Intel® Xeon® Processor C5500/C3500 Series Datasheet, Volume 1	Text Change in Section 13.1.10.4, Processor VTT Voltage Identification (VTT_VID) Signals
30.	• Intel® Xeon® Processor C5500/C3500 Series Datasheet, Volume 1	Updated Table 178 System Reference Clock AC Specifications and Added More Detailed BCLK Period Specification

Identification Information

Component Identification via Programming Interface

The Intel® Xeon® Processor C5500/C3550 Series stepping can be identified by the following register contents:

Reserved	Extended Family[1]	Extended Model[2]	Reserved	Processor Type[3]	Family Code[4]	Model Number[5]	Stepping ID[6]
31:28	**27:20**	**19:16**	**15:14**	**13:12**	**11:8**	**7:4**	**3:0**
	00000000b	0001b		00b	0110	1110b	xxxxb

Note:
1. The Extended Family, bits [27:20] are used in conjunction with the Family Code, specified in bits [11:8], to indicate whether the processor belongs to the Intel386, Intel486, Pentium, Pentium Pro, Pentium 4, or Intel® Core™ processor family.
2. The Extended Model, bits [19:16] in conjunction with the Model Number, specified in bits [7:4], are used to identify the model of the processor within the processor's family.
3. The Processor Type, specified in bits [13:12] indicates whether the processor is an original OEM processor, an OverDrive processor, or a dual processor (capable of being used in a dual processor system).
4. The Family Code corresponds to bits [11:8] of the EDX register after RESET, bits [11:8] of the EAX register after the CPUID instruction is executed with a 1 in the EAX register, and the generation field of the Device ID register accessible through Boundary Scan.
5. The Model Number corresponds to bits [7:4] of the EDX register after RESET, bits [7:4] of the EAX register after the CPUID instruction is executed with a 1 in the EAX register, and the model field of the Device ID register accessible through Boundary Scan.
6. The Stepping ID in bits [3:0] indicates the revision number of that model. See Table 9 for the processor stepping ID number in the CPUID information.

When EAX is initialized to a value of '1', the CPUID instruction returns the *Extended Family, Extended Model, Processor Type, Family Code, Model Number and Stepping ID* value in the EAX register. Note that the EDX processor signature value after reset is equivalent to the processor signature output value in the EAX register.

Cache and TLB descriptor parameters are provided in the EAX, EBX, ECX and EDX registers after the CPUID instruction is executed with a 2 in the EAX register.

The Intel® Xeon® Processor C5500/C3550 Series can be identified by the following register contents:

Stepping	Vendor ID[1]	Device ID[2]	Revision ID[3]
B-0	8086h	370xh	10h

Notes:
1. The Vendor ID corresponds to bits 15:0 of the Vendor ID Register located at offset 00–01h in the PCI function 0 configuration space.
2. The Device ID corresponds to bits 15:0 of the Device ID Register located at Device 0 offset 02–03h in the PCI function 0 configuration space. Lower 4 bits will vary by specific SKU.
3. The Revision Number corresponds to bits 7:0 of the Revision ID Register located at offset 08h in the PCI function 0 configuration space.

Component Marking Information

The processor stepping can be identified by the following component markings.

Figure 1. Processor Production Top-Side Markings (Example)

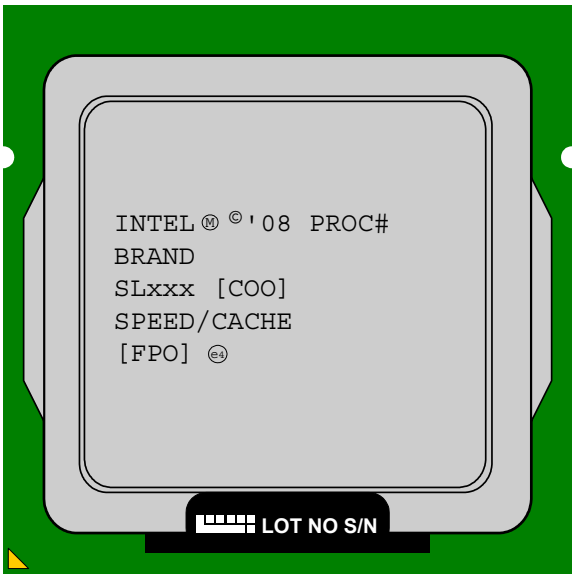


Table 9. Processor Identification

S-Spec Number	Processor Number	Stepping	Processor Signature	Core Frequency (GHz) / Intel® Quickpath Interrerconnect (Gt/s) / DDR3 (MHz)	Max Intel® Turbo Boost Technology Frequency (GHz)[2]	Shared L3 Cache Size (MB)	Notes
SLBWP	EC5549	B-0	106E4h	2.53 / 5.86 / 1333	4 core: 3.20 3 core: 3.20 2 core: 3.46 1 core: 3.07	8	1, 3, 4, 6
SLBWM	EC5509	B-0	106E4h	2.00 / 4.8 / 1066	N/A	8	1, 3, 4, 6
SLBWJ	EC3539	B-0	106E4h	2.13 / N.A. / 1066	N/A	8	1, 3, 4, 7
SLBWK	LC5528	B-0	106E4h	2.13 / 4.8 / 1066	1 core: 2.53	8	1, 3, 4, 8
SLBWL	EC5539	B-0	106E4h	2.27 / 5.86 / 1333	N/A	4	1, 3, 4, 9
SLBWF	LC5518	B-0	106E4h	1.73 / 4.8 / 1066	1 core: 2.13	8	1, 3, 4, 10
SLBWN	P1053	B-0	106E4h	1.33 / N.A / 800	N/A	2	1, 3, 4, 5, 11
SLBWG	LC3528	B-0	106E4h	1.73 / N.A. / 1066	1 core: 1.87	4	1, 3, 4, 12
SLBWH	LC3518	B-0	106E4h	1.73 / N.A. / 800	N/A	2	1, 3, 4, 13

Notes:

1. Intel® Hyper-Threading Technology enabled.
2. This column indicates maximum Intel® Turbo Boost Technology frequency (GHz) for 4, 3, 2, or 1 cores active respectively.
3. Intel® Virtualization Technology for IA-32, Intel® 64 and Intel® Architecture (Intel® VT-x) enabled.
4. Intel® Virtualization Technology for Directed I/O (Intel® VT-d) enabled.
5. Processor number P1053 is the Intel® Celeron® processor P1053. As such, it has string information identifying it as an Intel® Celeron® processor.
6. This is an Intel® Xeon® processor C5500 series with 85W TDP (Thermal Design Power).
7. This is an Intel® Xeon® processor C3500 series with 65W TDP (Thermal Design Power).
8. This is an Intel® Xeon® processor C5500 series with 60W TDP (Thermal Design Power).
9. This is an Intel® Xeon® processor C5500 series with 65W TDP (Thermal Design Power).
10. This is an Intel® Xeon® processor C5500 series with 48W TDP (Thermal Design Power).
11. This is the Intel® Celeron® processor P1053 with 30W TDP (Thermal Design Power).
12. This is an Intel® Xeon® processor C3500 series with 35W TDP (Thermal Design Power).
13. This is an Intel® Xeon® processor C3500 series with 23W TDP (Thermal Design Power).

Mixing Processors Within DP Platforms

Intel supports dual processor (DP) configurations consisting if processors:

- From the same power optimization segment.
- That support the same maximum Intel® QuickPath® Interconnect and DDR3 memory speeds.
- That share symmetry across physical packages with respect to the number of logical processors per package, number of cores per package, number of Intel® QuickPath® Interconnect interfaces, and cache topology.
- That have identical Extended Family, Extended Model, Processor Type, Family Code, and Model Number as indicated by the function 1 of the CPUID instruction.

Note: Processors must operate with the same Intel® QuickPath® Interconnect, DDR3 memory and core frequency.

While Intel does not prevent processors from operating together, some combinations may not be supported due to limited validation, which may result in uncharacterized errata. Coupling this fact with the large number of Intel® Xeon® processor C5500/C3500 series attributes, the following population rules and stepping matrix have been developed to clearly define supported configurations.

- Processors must be of the same power-optimization segment. This insures processors include the same maximum Intel® QuickPath® Interconnect and DDR3 operating speeds and cache sizes.
- Processors must operate at the same frequency. Processors within the same power-optimization segment supporting different maximum core frequencies can be operated within a system. Mixing components operating at different internal clock frequencies is not supported and will not be validated by Intel.
- Processors must share symmetry across physical packages with respect to the number of logical processors per package, number of Intel® QuickPath® interfaces, and cache topology.
- Mixing dissimilar steppings is only supported with processors that have identical Extended Family, Extended Model, Processor Type, Family Code, and Model Number, as indicated by function 1 of the CPUID instruction. Mixing processors of different steppings, but the same model, as per CPUID instruction, is supported. Details regarding the CPUID instruction are provided in AP-487, *Intel® Processor Identification and the CPUID Instruction* application note and in the *Intel® 64 and IA-32 Architectures Software Developer's Manual*, Volume 2A.
- After AND'ing the feature flag and extended feature flag from the installed processors, any processor whose set of feature flags exactly matches the AND'ed feature flags can be selected by the IOS as the BSP. If no processor exactly matches the AND'ed feature flag values, then the processors with the numerically lower CPUID should be selected as the BSP.
- Intel requires that the processor microcode update be loaded on each processor operating within the system. Any processor that does not have the proper microcode update loaded is considered by Intel to be operating out of specification.
- The workarounds identified in this, and subsequent specification updates, must be properly applied to each processor in the system. Certain errata are specific to the multi-processor environment. Errata for all processor steppings will affect system performance if not properly worked around.
- Customers are fully responsible for the validation of their system configurations.

Intel® Xeon® Processor C5500/C3550 Series Errata

BF1. The Processor Reports #TS Instead Of #GP Fault

Problem: A jump to a busy TSS (Task-State Segment) may cause a #TS (invalid TSS exception) instead of a #GP fault (general protection exception).

Implication: Operation systems that access a busy TSS may get invalid TSS fault instead of a #GP fault. Intel has not observed this erratum with any commercially available software.

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes.

BF2. REP MOVS/STOS Executing with Fast Strings Enabled And Crossing Page Boundaries With Inconsistent Memory Types Use Incorrect Data Size Or Lead To Memory-Ordering Violations

Problem: Under certain conditions as described in the Software Developers Manual section "Out-of-Order Stores For String Operations in Pentium 4, Intel Xeon, and P6 Family Processors" the processor performs REP MOVS or REP STOS as fast strings. Fast string REP MOVS/REP STOS instructions that cross page boundaries from WB/WC memory types to UC/WP/WT memory types may start using an incorrect data size or may observe memory ordering violations.

Implication: Upon crossing the page boundary the following may occur, dependent on the new page memory type:

- UC the data size of each write will now always be 8 bytes, as opposed to the original data size.
- WP the data size of each write will now always be 8 bytes, as opposed to the original data size and there may be a memory ordering violation.
- WT there may be a memory ordering violation.

Workaround: Software should avoid crossing page boundaries from WB or WC memory type to UC, WP or WT memory type within a single REP MOVS or REP STOS instruction that will execute with fast strings enabled.

Status: For the steppings affected, see the Summary Table of Changes.

BF3. Code Segment Limit/Canonical Faults On RSM Serviced Before Higher Priority Interrupts/Exceptions And Pushes Wrong Address Onto Stack

Problem: Normally, when the processor encounters a Segment Limit or Canonical Fault due to code execution, a #GP (General Protection Exception) fault is generated after all higher priority Interrupts and exceptions are serviced. If RSM (Resume from System Management Mode) returns to execution flow that results in a Code Segment Limit or Canonical Fault, the #GP fault may be serviced before a higher priority Interrupt or Exception (e.g. NMI (Non-Maskable Interrupt), Debug break(#DB), Machine Check (#MC), etc.). If the RSM attempts to return to a non-canonical address, the address pushed onto the stack for this #GP fault may not match the non-canonical address that caused the fault.

Implication: Operating systems may observe a #GP fault being serviced before higher priority Interrupts and Exceptions. Intel has not observed this erratum on any commercially available software.

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes.

BF4. Performance Monitor SSE Retired Instructions Return Incorrect Values

Problem: Performance Monitoring counter SIMD_INST_RETIRED (Event: C7H) is used to track retired SSE instructions. The processor may also count other types of instructions resulting in higher than expected values.

Implication: Performance Monitoring counter SIMD_INST_RETIRED may report count higher than expected.

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes.

BF5. Premature Execution Of Load Operation Prior To Exception Handler Invocation

Problem: If any of the below circumstances occur, it is possible that the load portion of the instruction will have executed before the exception handler is entered.

- If an instruction that performs a memory load causes a code segment limit violation.
- If a waiting X87 floating-point (FP) instruction or MMX™ technology (MMX) instruction that performs a memory load has a floating-point exception pending.
- If an MMX or SSE/SSE2/SSE3/SSSE3 extensions (SSE) instruction that performs a memory load and has either CR0.EM=1 (Emulation bit set), or a floating-point Top-of-Stack (FP TOS) not equal to 0, or a DNA exception pending.

Implication: In normal code execution where the target of the load operation is to write back memory there is no impact from the load being prematurely executed, or from the restart and subsequent re-execution of that instruction by the exception handler. If the target of the load is to uncached memory that has a system side-effect, restarting the instruction may cause unexpected system behavior due to the repetition of the side-effect. Particularly, while CR0.TS [bit 3] is set, a MOVD/MOVQ with MMX/XMM register operands may issue a memory load before getting the DNA exception.

Workaround: Code which performs loads from memory that has side-effects can effectively workaround this behavior by using simple integer-based load instructions when accessing side-effect memory and by ensuring that all code is written such that a code segment limit violation cannot occur as a part of reading from side-effect memory.

Status: For the steppings affected, see the Summary Table of Changes.

BF6. MOV To/From Debug Registers Causes Debug Exception

Problem: When in V86 mode, if a MOV instruction is executed to/from a debug registers, a general-protection exception (#GP) should be generated. However, in the case when the general detect enable flag (GD) bit is set, the observed behavior is that a debug exception (#DB) is generated instead.

Implication: With debug-register protection enabled (i.e., the GD bit set), when attempting to execute a MOV on debug registers in V86 mode, a debug exception will be generated instead of the expected general-protection fault.

Workaround: In general, operating systems do not set the GD bit when they are in V86 mode. The GD bit is generally set and used by debuggers. The debug exception handler should check that the exception did not occur in V86 mode before continuing. If the exception did occur in V86 mode, the exception may be directed to the general-protection exception handler.

Status: For the steppings affected, see the Summary Table of Changes.

BF7. Incorrect Address Computed For Last Byte Of FXSAVE/FXRSTOR Image Leads To Partial Memory Update

Problem: A partial memory state save of the 512-byte FXSAVE image or a partial memory state restore of the FXRSTOR image may occur if a memory address exceeds the 64KB limit while the processor is operating in 16-bit mode or if a memory address exceeds the 4GB limit while the processor is operating in 32-bit mode.

Implication: FXSAVE/FXRSTOR will incur a #GP fault due to the memory limit violation as expected but the memory state may be only partially saved or restored.

Workaround: Software should avoid memory accesses that wrap around the respective 16-bit and 32-bit mode memory limits.

Status: For the steppings affected, see the Summary Table of Changes.

BF8. Values for LBR/BTS/BTM Incorrect After Exit from SMM

Problem: After a return from SMM (System Management Mode), the CPU will incorrectly update the LBR (Last Branch Record) and the BTS (Branch Trace Store), hence rendering their data invalid. The corresponding data if sent out as a BTM on the system bus will also be incorrect.

Note: This issue would only occur when one of the 3 above mentioned debug support facilities are used.

Implication: The value of the LBR, BTS, and BTM immediately after an RSM operation should not be used.

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes.

BF9. Single Step Interrupts With Floating Point Exception Pending Mishandled

Problem: In certain circumstances, when a floating point exception (#MF) is pending during single-step execution, processing of the single-step debug exception (#DB) may be mishandled.

Implication: When this erratum occurs, #DB will be incorrectly handled as follows:

- #DB is signaled before the pending higher priority #MF (Interrupt 16)
- #DB is generated twice on the same instruction

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes.

BF10. Fault On ENTER Instruction Results In Unexpected Values On Stack Frame

Problem: The ENTER instruction is used to create a procedure stack frame. Due to this erratum, if execution of the ENTER instruction results in a fault, the dynamic storage area of the resultant stack frame may contain unexpected values (i.e. residual stack data as a result of processing the fault).

Implication: Data in the created stack frame may be altered following a fault on the ENTER instruction. Please refer to "Procedure Calls For Block-Structured Languages" in *IA-32 Intel® Architecture Software Developer's Manual, Vol. 1, Basic Architecture*, for information on the usage of the ENTER instructions. This erratum is not expected to occur in ring 3. Faults are usually processed in ring 0 and stack switch occurs when transferring to ring 0. Intel has not observed this erratum on any commercially available software.

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes.

BF11. IRET Causes Unexpected Alignment Check Exception

Problem: In IA-32e mode, it is possible to get an Alignment Check Exception (#AC) on the IRET instruction even though alignment checks were disabled at the start of the IRET. This can only occur if the IRET instruction is returning from CPL3 code to CPL3 code. IRETs from CPL0/1/2 are not affected. This erratum can occur if the EFLAGS value on the stack has the AC flag set, and the interrupt handler's stack is misaligned. In IA-32e mode, RSP is aligned to a 16-byte boundary before pushing the stack frame.

Implication: In IA-32e mode, under the conditions given above, an IRET can get a #AC even if alignment checks are disabled at the start of the IRET. This erratum can only be observed with a software generated stack frame.

Workaround: Software should not generate misaligned stack frames for use with IRET.

Status: For the steppings affected, see the Summary Table of Changes.

BF12. General Protection Fault (#GP) for Instructions Greater Than 15 Bytes Preempted

Problem: When the processor encounters an instruction that is greater than 15 bytes in length, a #GP is signaled when the instruction is decoded. Under some circumstances, the #GP fault may be preempted by another lower priority fault (e.g. Page Fault (#PF)). However, if the preempting lower priority faults are resolved by the operating system and the instruction retried, a #GP fault will occur.

Implication: Software may observe a lower-priority fault occurring before or in lieu of a #GP fault. Instructions of greater than 15 bytes in length can only occur if redundant prefixes are placed before the instruction.

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes.

BF13. General Protection (#GP) Fault Not Signaled On Data Segment Limit Violation Above 4-G Limit

Problem: In 32-bit mode, memory accesses to flat data segments (base = 00000000h) that occur above the 4G limit (0ffffffffh) may not signal a #GP fault.

Implication: When such memory accesses occur in 32-bit mode, the system may not issue a #GP fault.

Workaround: Software should ensure that memory accesses in 32-bit mode do not occur above the 4G limit (0ffffffffh).

Status: For the steppings affected, see the Summary Table of Changes.

BF14. LBR, BTS, BTM May Report Wrong Address When Exception/Interrupt Occurs In 64-bit Mode

Problem: An exception/interrupt event should be transparent to the LBR (Last Branch Record), BTS (Branch Trace Store) and BTM (Branch Trace Message) mechanisms. However, during a specific boundary condition where the exception/interrupt occurs right after the execution of an instruction at the lower canonical boundary (0x00007FFFFFFFFFFF) in 64-bit mode, the LBR return registers will save a wrong return address with bits 63 to 48 incorrectly sign extended to all 1's. Subsequent BTS and BTM operations which report the LBR will also be incorrect.

Implication: LBR, BTS and BTM may report incorrect information in the event of an exception/interrupt.

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes.

BF15. MONITOR Or CLFLUSH On Local XAPIC Address Space Results In Hang

Problem: If the target linear address range for a MONITOR or CLFLUSH is mapped to the local xAPIC's address space, the processor will hang.

Implication: When this erratum occurs, the processor will hang. The local xAPIC's address space must be uncached. The MONITOR instruction only functions correctly if the specified linear address range is of the type write-back. CLFLUSH flushes data from the cache. Intel has not observed this erratum with any commercially available software.

Workaround: Do not execute MONITOR or CLFLUSH instructions on the local xAPIC address space.

Status: For the steppings affected, see the Summary Table of Changes.

BF16. Corruption Of CS Segment Register During RSM While Transitioning From Real Mode To Protected Mode

Problem: During the transition from real mode to protected mode, if an SMI (System Management Interrupt) occurs between the MOV to CR0 that sets PE (Protection Enable, bit 0) and the first FAR JMP, the subsequent RSM (Resume from System Management Mode) may cause the lower two bits of CS segment register to be corrupted.

Implication: The corruption of the bottom two bits of the CS segment register will have no impact unless software explicitly examines the CS segment register between enabling protected mode and the first FAR JMP. *Intel® 64 and IA-32 Architectures Software Developer's Manual Volume 3A: System Programming Guide, Part 1*, in the section titled "Switching to Protected Mode" recommends the FAR JMP immediately follows the write to CR0 to enable protected mode. Intel has not observed this erratum with any commercially available software.

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes.

BF17. Performance Monitoring Events For Read Miss to Level 3 Cache Fill Occupancy Counter Incorrect

Problem: Whenever an Level 3 cache fill conflicts with another request's address, the miss to fill occupancy counter, UNC_GQ_ALLOC.RT_LLC_MISS (Event 02H), will provide erroneous results.

Implication: The Performance Monitoring UNC_GQ_ALLOC.RT_LLC_MISS event may count a value higher than expected. The extent to which the value is higher than expected is determined by the frequency of the L3 address conflict.

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes.

BF18. VM Exit On MWAIT Incorrectly Reports Monitoring Hardware As Armed

Problem: A processor write to the address range armed by the MONITOR instruction may not immediately trigger the monitoring hardware. Consequently, a VM exit on a later MWAIT may incorrectly report the monitoring hardware as armed, when it should be reported as unarmed due to the write occurring prior to the MWAIT.

Implication: If a write to the range armed by the MONITOR instruction occurs between the MONITOR and the MWAIT, the MWAIT instruction may start executing before the monitoring hardware is triggered. If the MWAIT instruction causes a VM exit, this could cause its exit qualification to incorrectly report 0x1. In the recommended usage model for MONITOR/MWAIT, there is no write to the range armed by the MONITOR instruction between the MONITOR and the MWAIT.

Workaround: Software should never write to the address range armed by the MONITOR instruction between the MONITOR and the subsequent MWAIT.

Status: For the steppings affected, see the Summary Table of Changes.

BF19. Delivery Status Of LINT0 Register Of Local Vector Table Lost

Problem: The Delivery Status bit of the LINT0 Register of the Local Vector Table will not be restored after a transition out of C6 under the following conditions

- LINT0 is programmed as level-triggered
- The delivery mode is set to either Fixed or ExtINT
- There is a pending interrupt which is masked with the interrupt enable flag (IF)

Implication: The Delivery Status bit of the LINT0 Register will unexpectedly not be set. Intel has not observed this erratum with any commercially available software or system.

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes.

BF20. Performance Monitor Event SEGMENT_REG_LOADS Counts Inaccurately

Problem: The performance monitor event SEGMENT_REG_LOADS (Event 06H) counts instructions that load new values into segment registers. The value of the count may be inaccurate.

Implication: The performance monitor event SEGMENT_REG_LOADS may reflect a count higher or lower than the actual number of events.

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes.

BF21. #GP On Segment Selector Descriptor that Straddles Canonical Boundary May Not Provide Correct Exception Error Code

Problem: During a #GP (General Protection Exception), the processor pushes an error code on to the exception handler's stack. If the segment selector descriptor straddles the canonical boundary, the error code pushed onto the stack may be incorrect.

Implication: An incorrect error code may be pushed onto the stack. Intel has not observed this erratum with any commercially available software.

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes.

BF22. Improper Parity Error Signaled In IQ Following Reset When Code Breakpoint Set On #GP Instruction

Problem: While coming out of cold reset or exiting from C6, if the processor encounters an instruction longer than 15 bytes (which causes a #GP) and a code breakpoint is enabled on that instruction, an IQ (Instruction Queue) parity error may be incorrectly logged resulting in an MCE (Machine Check Exception).

Implication: When this erratum occurs, an MCE may be incorrectly signaled.

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes.

BF23. An Enabled Debug Breakpoint or Single Step Trap May Be Taken after MOV SS/POP SS Instruction if it is Followed by an Instruction That Signals a Floating Point Exception

Problem: A MOV SS/POP SS instruction should inhibit all interrupts including debug breakpoints until after execution of the following instruction. This is intended to allow the sequential execution of MOV SS/POP SS and MOV [r/e]SP, [r/e]BP instructions without having an invalid stack during interrupt handling. However, an enabled debug breakpoint or single step trap may be taken after MOV SS/POP SS if this instruction is followed by an instruction that signals a floating point exception rather than a MOV [r/e]SP, [r/e]BP instruction. This results in a debug exception being signaled on an unexpected instruction boundary since the MOV SS/POP SS and the following instruction should be executed atomically.

Implication: This can result in incorrect signaling of a debug exception and possibly a mismatched Stack Segment and Stack Pointer. If MOV SS/POP SS is not followed by a MOV [r/e]SP, [r/e]BP, there may be a mismatched Stack Segment and Stack Pointer on any exception. Intel has not observed this erratum with any commercially available software or system.

Workaround: As recommended in the *IA32 Intel® Architecture Software Developer's Manual*, the use of MOV SS/POP SS in conjunction with MOV [r/e]SP, [r/e]BP will avoid the failure since the MOV [r/e]SP, [r/e]BP will not generate a floating point exception. Developers of debug tools should be aware of the potential incorrect debug event signaling created by this erratum.

Status: For the steppings affected, see the Summary Table of Changes.

BF24. IA32_MPERF Counter Stops Counting During On-Demand TM1

Problem: According to the *Intel® 64 and IA-32 Architectures Software Developer's Manual* Volume 3A: System Programming Guide, the ratio of IA32_MPERF (MSR E7H) to IA32_APERF (MSR E8H) should reflect actual performance while TM1 or on-demand throttling is activated. Due to this erratum, IA32_MPERF MSR stops counting while TM1 or on-demand throttling is activated, and the ratio of the two will indicate higher processor performance than actual.

Implication: The incorrect ratio of IA32_APERF/IA32_MPERF can mislead software P-state (performance state) management algorithms under the conditions described above. It is possible for the Operating System to observe higher processor utilization than actual, which could lead the OS into raising the P-state. During TM1 activation, the OS P-state request is irrelevant and while on-demand throttling is enabled, it is expected that the

OS will not be changing the P-state. This erratum should result in no practical implication to software.

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes.

BF25. Intel® QuickPath Memory Controller tTHROT_OPREF Timings May be Violated During Self Refresh Entry

Problem: During self refresh entry, the memory controller may issue more refreshes than permitted by tTHROT_OPREF (bits 29:19 in MC_CHANNEL_{0,1}_REFRESH_TIMING CSR).

Implication: The intention of tTHROT_OPREF is to limit current. Since current supply conditions near self refresh entry are not critical, there is no measurable impact due to this erratum.

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes.

BF26. Processor Over Counts Correctable Cache MESI State Errors

Problem: Under a specific set of conditions, correctable Level 2 cache hierarchy MESI state errors may be counted more than once per occurrence of a correctable error.

Implication: Correctable Level 2 cache hierarchy MESI state errors may be reported in the MCi_STATUS register at a rate higher than their actual occurrence.

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes.

BF27. Synchronous Reset of IA32_APERF/IA32_MPERF Counters On Overflow Does Not Work

Problem: When either the IA32_MPERF or IA32_APERF MSR (E7H, E8H) increments to its maximum value of 0xFFFF_FFFF_FFFF_FFFF, both MSRs are supposed to synchronously reset to 0x0 on the next clock. This synchronous reset does not work. Instead, both MSRs increment and overflow independently.

Implication: Software can not rely on synchronous reset of the IA32_APERF/IA32_MPERF registers.

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes.

BF28. Disabling Thermal Monitor While Processor Hot, Then Re-enabling Results In Stuck Core Operating Ratio

Problem: If a processor is at its TCC (Thermal Control Circuit) activation temperature and then Thermal Monitor is disabled by a write to IA32_MISC_ENABLES MSR (1A0H) bit [3], a subsequent re-enable of Thermal Monitor will result in an artificial ceiling on the maximum core P-state. The ceiling is based on the core frequency at the time of Thermal Monitor disable. This condition will only correct itself once the processor reaches its TCC activation temperature again.

Implication: Since Intel requires that Thermal Monitor be enabled in order to be operating within specification, this erratum should never be seen during normal operation.

Workaround: Software should not disable Thermal Monitor during processor operation.

Status: For the steppings affected, see the Summary Table of Changes.

BF29. PECI Does Not Support PCI Configuration Reads/Writes To Misaligned Addresses

Problem: The PECI (Platform Environment Control Interface) specification allows for partial reads from or writes to misaligned addresses within the PCI configuration space. However, the PECI client does not properly interpret addresses that are Dword (4 byte) misaligned and may read or write incorrect data.

Implication: Writes to or reads from Dword misaligned addresses could result in unintended side effects and unpredictable behavior.

Workaround: PECI host controllers may issue byte, word and Dword reads and writes as long as they are aligned to Dword addresses.

Status: For the steppings affected, see the Summary Table of Changes.

BF30. OVER Bit For IA32_MCi_STATUS Register Set On Specific Internal Error

Problem: If a specific type of internal unclassified error is detected, as identified by IA32_MCi_STATUS.MCACOD=0x0405, the IA32_MCi_ STATUS.OVER (overflow) bit [62] may be erroneously set.

Implication: The OVER bit of the MCi_STATUS register may be incorrectly set for a specific internal unclassified error.

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes.

BF31. Writing Local Vector Table (LVT) When Interrupt is Pending Causes Unexpected Interrupt

Problem: If a local interrupt is pending when the LVT entry is written, an interrupt may be taken on the new interrupt vector even if the mask bit is set.

Implication: An interrupt may immediately be generated with the new vector when a LVT entry is written, even if the new LVT entry has the mask bit set. If there is no Interrupt Service Routine (ISR) set up for that vector the system will GP fault. If the ISR does not do an End of Interrupt (EOI) the bit for the vector will be left set in the in-service register and mask all interrupts at the same or lower priority.

Workaround: Any vector programmed into an LVT entry must have an ISR associated with it, even if that vector was programmed as masked. This ISR routine must do an EOI to clear any unexpected interrupts that may occur. The ISR associated with the spurious vector does not generate an EOI, therefore the spurious vector should not be used when writing the LVT.

Status: For the steppings affected, see the Summary Table of Changes.

BF32. Faulting MMX Instruction Incorrectly Updates x87 FPU Tag Word

Problem: Under a specific set of conditions, MMX stores (MOVD, MOVQ, MOVNTQ, MASKMOVQ) which cause memory access faults (#GP, #SS, #PF, or #AC), may incorrectly update the x87 FPU tag word register.

This erratum will occur when the following additional conditions are also met.

- The MMX store instruction must be the first MMX instruction to operate on x87 FPU state (i.e. the x87 FP tag word is not already set to 0x0000).
- For MOVD, MOVQ, MOVNTQ stores, the instruction must use an addressing mode that uses an index register (this condition does not apply to MASKMOVQ).

Implication: If the erratum conditions are met, the x87 FPU tag word register may be incorrectly set to a 0x0000 value when it should not have been modified.

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes.

BF33. xAPIC Timer Decrements Too Quickly Following Automatic Reload While In Periodic Mode

Problem: When the xAPIC Timer is automatically reloaded by counting down to zero in periodic mode, the xAPIC Timer may slip in its synchronization with the external clock. The xAPIC timer may be shortened by up to one xAPIC timer tick.

Implication: When the xAPIC Timer is automatically reloaded by counting down to zero in periodic mode, the xAPIC Timer may slip in its synchronization with the external clock. The xAPIC timer may be shortened by up to one xAPIC timer tick.

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes.

BF34. Reported Memory Type May Not Be Used To Access the VMCS And Referenced Data Structures

Problem: Bits 53:50 of the IA32_VMX_BASIC MSR report the memory type that the processor uses to access the VMCS and data structures referenced by pointers in the VMCS. Due to this erratum, a VMX access to the VMCS or referenced data structures will instead use the memory type that the MTRRs (memory-type range registers) specify for the physical address of the access.

Implication: Bits 53:50 of the IA32_VMX_BASIC MSR report that the WB (write-back) memory type will be used but the processor may use a different memory type.

Workaround: Software should ensure that the VMCS and referenced data structures are located at physical addresses that are mapped to WB memory type by the MTRRs.

Status: For the steppings affected, see the Summary Table of Changes.

BF35. B0-B3 Bits In DR6 For Non-Enabled Breakpoints May be Incorrectly Set

Problem: Some of the B0-B3 bits (breakpoint conditions detect flags, bits [3:0]) in DR6 may be incorrectly set for non-enabled breakpoints when the following sequence happens:

1. MOV or POP instruction to SS (Stack Segment) selector;
2. Next instruction is FP (Floating Point) that gets FP assist
3. Another instruction after the FP instruction completes successfully
4. A breakpoint occurs due to either a data breakpoint on the preceding instruction or a code breakpoint on the next instruction.

A non-enabled breakpoint triggered on step 1 or step 2 may be reported in B0-B3 after the breakpoint occurs in step 4.

Implication: Due to this erratum, B0-B3 bits in DR6 may be incorrectly set for non-enabled breakpoints.

Workaround: Software should not execute a floating point instruction directly after a MOV SS or POP SS instruction.

Status: For the steppings affected, see the Summary Table of Changes.

BF36. Core C6 Clears Previously Logged TLB Errors

Problem: Following an exit from core C6, previously logged TLB (Translation Lookaside Buffer) errors in IA32_MCi_STATUS may be cleared.

Implication: TLB errors logged in the associated machine check bank prior to core C6 entry may be cleared. Provided machine check exceptions are enabled, the machine check exception handler can log any uncorrectable TLB errors prior to core C6 entry. The TLB marks all detected errors as uncorrectable.

Workaround: As long as machine check exceptions are enabled, the machine check exception handler can log the TLB error prior to core C6 entry. This will ensure the error is logged before it is cleared.

Status: For the steppings affected, see the Summary Table of Changes.

BF37. Performance Monitor Event MISALIGN_MEM_REF May Over Count

Problem: The MISALIGN_MEM_REF Performance Monitoring (Event 05H) may over count memory misalignment events, possibly by orders of magnitude.

Implication: Software relying on MISALIGN_MEM_REF to count cache line splits for optimization purposes may read excessive number of memory misalignment events.

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes.

BF38. Changing Memory Type for In-Use Page Translation Leads To Memory-Ordering Violations

Problem: Under complex microarchitectural conditions, if software changes the memory type for data being actively used and shared by multiple threads without the use of semaphores or barriers, software may see load operations execute out of order.

Implication: Memory ordering may be violated. Intel has not observed this erratum with any commercially available software.

Workaround: Software should ensure pages are not being actively used before requesting their memory type be changed.

Status: For the steppings affected, see the Summary Table of Changes.

BF39. Infinite Stream of Interrupts Occur If ExtINT Delivery Mode Interrupt Received While All Cores In C6

Problem: If all logical processors in a core are in C6, an ExtINT delivery mode interrupt is pending in the xAPIC and interrupts are blocked with EFLAGS.IF=0, the interrupt will be processed after C6 wakeup and after interrupts are re-enabled (EFLAGS.IF=1). However, the pending interrupt event will not be cleared.

Implication: An infinite stream of interrupts will occur on the core servicing the external interrupt. Intel has not observed this erratum with any commercially available software/system.

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes.

BF40. Two xAPIC Timer Event Interrupts Unexpectedly Occur

Problem: If an xAPIC timer event is enabled and while counting down the current count reaches 1 at the same time that the processor thread begins a transition to a low power C-state, the xAPIC may generate two interrupts instead of the expected one when the processor returns to C0.

Implication: Due to this erratum, two interrupts may unexpectedly be generated by an xAPIC timer event.

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes.

BF41. EOI Transaction May Not be Sent if Software Enters Core C6 During an Interrupt Service Routine

Problem: If core C6 is entered after the start of an interrupt service routine but before a write to the APIC EOI (End of Interrupt) register, and the core is woken up by an event other than a fixed interrupt source the core may drop the EOI transaction the next time APIC EOI register is written and further interrupts from the same or lower priority level will be blocked.

Implication: EOI transactions and interrupts may be blocked when core C6 is used during interrupt service routines. Intel has not observed this erratum with any commercially available software.

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes.

BF42. FREEZE_WHILE_SMM Does Not Prevent Event From Pending PEBS During SMM

Problem: In general, a PEBS record should be generated on the first count of the event after the counter has overflowed. However, IA32_DEBUGCTL_MSR.FREEZE_WHILE_SMM (MSR 1D9H, bit [14]) prevents performance counters from counting during SMM (System Management Mode). Due to this erratum, if

1. A performance counter overflowed before an SMI
2. A PEBS record has not yet been generated because another count of the event has not occurred
3. The monitored event occurs during SMM

then a PEBS record will be saved after the next RSM instruction.

When FREEZE_WHILE_SMM is set, a PEBS should not be generated until the event occurs outside of SMM.

Implication: A PEBS record may be saved after an RSM instruction due to the associated performance counter detecting the monitored event during SMM; even when FREEZE_WHILE_SMM is set.

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes.

BF43. APIC Error "Received Illegal Vector" Lost

Problem: APIC (Advanced Programmable Interrupt Controller) may not update the ESR (Error Status Register) flag Received Illegal Vector bit [6] properly when an illegal vector error is received on the same internal clock that the ESR is being written (as part of the write-read ESR access flow). The corresponding error interrupt will also not be generated for this case.

Implication: Due to this erratum, an incoming illegal vector error may not be logged into ESR properly and may not generate an error interrupt.

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes.

BF44. DR6 May Contain Incorrect Information When the First Instruction After a MOV SS, r/m or POP SS is a Store

Problem: Normally, each instruction clears the changes in DR6 (Debug Status Register) caused by the previous instruction. However, the instruction following a MOV SS,r/m (MOV to

the stack segment selector) or POP SS (POP stack segment selector) instruction will not clear the changes in DR6 because data breakpoints are not taken immediately after a MOV SS,r/m or POP SS instruction. Due to this erratum, any DR6 changes caused by a MOV SS,r/m or POP SS instruction may be cleared if the following instruction is a store.

Implication: When this erratum occurs, incorrect information may exist in DR6. This erratum will not be observed under normal usage of the MOV SS,r/m or POP SS instructions (i.e., following them with an instruction that writes [e/r]SP). When debugging or when developing debuggers, this behavior should be noted.

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes.

BF45. Uncorrectable Error Logged In IA32_CR_MC2_STATUS Results In System Hang

Problem: Uncorrectable errors logged in IA32_CR_MC2_STATUS MSR (409H) may also result in a system hang causing an Internal Timer Error (MCACOD = 0x0400h) to be logged in another machine check bank (IA32_MCi_STATUS).

Implication: Uncorrectable errors logged in IA32_CR_MC2_STATUS can further cause a system hang and an Internal Timer Error to be logged.

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes.

BF46. IA32_PERF_GLOBAL_CTRL MSR May be Incorrectly Initialized

Problem: The IA32_PERF_GLOBAL_CTRL MSR (38FH) bits [34:32] may be incorrectly set to 7H after reset; the correct value should be 0H.

Implication: The IA32_PERF_GLOBAL_CTRL MSR bits [34:32] may be incorrect after reset (EN_FIXED_CTR{0, 1, 2} may be enabled).

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes.

BF47. ECC Errors Cannot Be Injected On Back-to-Back Writes

Problem: ECC errors should be injected on every write that matches the address set in the MC_CHANNEL_{0,1}_ADDR_MATCH CSRs. Due to this erratum if there are two back-to-back writes that match MC_CHANNEL_{0,1}_ADDR_MATCH, the 2nd write will not have the error injected.

Implication: The 2nd back-to-back write that matches MC_CHANNEL_{0,1}_ADDR_MATCH will not have the ECC error properly injected. Setting MC_CHANNEL_{0,1}_ADDR_MATCH to a specific address will reduce the chance of being impacted by this erratum.

Workaround: Only injecting errors to specific address should reduce the chance on being impacted by this erratum.

Status: For the steppings affected, see the Summary Table of Changes.

BF48. Performance Monitor Interrupts Generated From Uncore Fixed Counters (394H) May be Ignored

Problem: Performance monitor interrupts (PMI's) from Uncore fixed counters are ignored when Uncore general performance monitor counters 3B0H-3BFH are not programmed.

Implication: This erratum blocks a usage model in which each of the cores can sample its own performance monitor events synchronously based on single interrupt from the Uncore.

Workaround: Program any one of the Uncore general performance monitor counters with a valid performance monitor event and enable the event by setting the local enable bit in the corresponding performance monitor event select MSR. For the usage model where no counting is desired, program that Uncore general performance counter's global enable bit to be zero.

Status: For the steppings affected, see the Summary Table of Changes.

BF49. Performance Monitor Counter INST_RETIRED.STORES May Count Higher than Expected

Problem: Performance Monitoring counter INST_RETIRED.STORES (Event: C0H) is used to track retired instructions which contain a store operation. Due to this erratum, the processor may also count other types of instructions including WRMSR and MFENCE.

Implication: Performance Monitoring counter INST_RETIRED.STORES may report counts higher than expected.

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes.

BF50. Sleeping Cores May Not be Woken Up on Logical Cluster Mode Broadcast IPI Using Destination Field Instead of Shorthand

Problem: If software sends a logical cluster broadcast IPI using a destination shorthand of 00B (No Shorthand) and writes the cluster portion of the Destination Field of the Interrupt Command Register to all ones while not using all 1s in the mask portion of the Destination Field, target cores in a sleep state that are identified by the mask portion of the Destination Field may not be woken up. This erratum does not occur if the destination shorthand is set to 10B (All Including Self) or 11B (All Excluding Self).

Implication: When this erratum occurs, cores which are in a sleep state may not wake up to handle the broadcast IPI. Intel has not observed this erratum with any commercially available software.

Workaround: Use destination shorthand of 10B or 11B to send broadcast IPIs.

Status: For the steppings affected, see the Summary Table of Changes.

BF51. Faulting Executions of FXRSTOR Update State Inconsistently

Problem: The state updated by a faulting FXRSTOR instruction may vary from one execution to another.

Implication: Software that relies on x87 state or SSE state following a faulting execution of FXRSTOR may behave inconsistently.

Workaround: Software handling a fault on an execution of FXRSTOR can compensate for execution variability by correcting the cause of the fault and executing FXRSTOR again.

Status: For the steppings affected, see the Summary Table of Changes.

BF52. Performance Monitor Event EPT.EPDPE_MISS Counted While EPT is Disabled

Problem: Performance monitor event EPT.EPDPE_MISS (Event: 4FH, Umask: 08H) is used to count Page Directory Pointer table misses while EPT (extended page tables) is enabled. Due to this erratum, the processor will count Page Directory Pointer table misses regardless of whether EPT is enabled or not.

Implication: Due to this erratum, performance monitor event EPT.EPDPE_MISS may report counts higher than expected.

Workaround: Software should ensure this event is only enabled while in EPT mode.

Status: For the steppings affected, see the Summary Table of Changes.

BF53. Memory Aliasing Of Code Pages Causes Unpredictable Behavior

Problem: The type of memory aliasing contributing to this erratum is the case where two different logical processors have the same code page mapped with two different memory types. Specifically, if one code page is mapped by one logical processor as write-back and by another as uncachable and certain instruction fetch timing conditions occur, the system may experience unpredictable behavior.

Implication: If this erratum occurs the system may have unpredictable behavior including a system hang. The aliasing of memory regions, a condition necessary for this erratum to occur, is documented as being unsupported in the *Intel 64 and IA-32 Intel® Architecture Software Developer's Manual, Volume 3A*, in the section titled *Programming the PAT*. Intel has not observed this erratum with any commercially available software or system.

Workaround: Code pages should not be mapped with uncacheable and cacheable memory types at the same time.

Status: For the steppings affected, see the Summary Table of Changes.

BF54. Performance Monitor Counters May Count Incorrectly

Problem: Under certain circumstances, a general purpose performance counter, IA32_PMC0-4 (C1H - C4H), may count at core frequency or not count at all instead of counting the programmed event.

Implication: The Performance Monitor Counter IA32_PMCx may not properly count the programmed event. Due to the requirements of the workaround there may be an interruption in the counting of a previously programmed event during the programming of a new event.

Workaround: Before programming the performance event select registers, IA32_PERFEVTSELx MSR (186H - 189H), the internal monitoring hardware must be cleared. This is accomplished by first disabling, saving valid events and clearing from the select registers, then programming three event values 0x4300D2, 0x4300B1 and 0x4300B5 into the IA32_PERFEVTSELx MSRs, and finally continuing with new event programming and restoring previous programming if necessary. Each performance counter, IA32_PMCx, must have its corresponding IA32_PREFEVTSELx MSR programmed with at least one of the event values and must be enabled in IA32_PERF_GLOBAL_CTRL MSR (38FH) bits [3:0]. All three values must be written to either the same or different IA32_PERFEVTSELx MSRs before programming the performance counters. Note that the performance counter will not increment when its IA32_PERFEVTSELx MSR has a value of 0x4300D2, 0x4300B1 or 0x4300B5 because those values have a zero UMASK field (bits [15:8]).

Status: For the steppings affected, see the Summary Table of Changes.

BF55. Performance Monitor Event Offcore_Response_0 (B7H) Does Not Count NT Stores To Local DRAM Correctly

Problem: When a IA32_PERFEVTSELx MSR is programmed to count the Offcore_response_0 event (Event:B7H), selections in the OFFCORE_RSP_0 MSR (1A6H) determine what is counted. The following two selections do not provide accurate counts when counting NT (Non-Temporal) Stores:

- OFFCORE_RSP_0 MSR bit [14] is set to 1 (LOCAL_DRAM) and bit [7] is set to 1 (OTHER): NT Stores to Local DRAM are not counted when they should have been.

- OFFCORE_RSP_0 MSR bit [9] is set to (OTHER_CORE_HIT_SNOOP) and bit [7] is set to 1 (OTHER): NT Stores to Local DRAM are counted when they should not have been.

Implication: The counter for the Offcore_response_0 event may be incorrect for NT stores.

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes.

BF56. EFLAGS Discrepancy On Page Faults and EPT-Induced VM Exits After Translation Change

Problem: This erratum is regarding the case where paging structures are modified to change a linear address from writable to non-writable without software performing an appropriate TLB invalidation. When a subsequent access to that address by a specific instruction (ADD, AND, BTC, BTR, BTS, CMPXCHG, DEC, INC, NEG, NOT, OR, ROL/ROR, SAL/SAR/SHL/SHR, SHLD, SHRD, SUB, XOR, and XADD) causes a page fault or an EPT-induced VM exit, the value saved for EFLAGS may incorrectly contain the arithmetic flag values that the EFLAGS register would have held had the instruction completed without fault or VM exit. For page faults, this can occur even if the fault causes a VM exit or if its delivery causes a nested fault.

Implication: None identified. Although the EFLAGS value saved by an affected event (a page fault or an EPT-induced VM exit) may contain incorrect arithmetic flag values, Intel has not identified software that is affected by this erratum. This erratum will have no further effects once the original instruction is restarted because the instruction will produce the same results as if it had initially completed without fault or VM exit.

Workaround: If the handler of the affected events inspects the arithmetic portion of the saved EFLAGS value, then system software should perform a synchronized paging structure modification and TLB invalidation.

Status: For the steppings affected, see the Summary Table of Changes.

BF57. System Hang if MC_CHANNEL_{0,1}_MC_DIMM_INIT_CMD.DO_ZQCL Commands Not Issued In Increasing Populated DDR3 Rank Order

Problem: ZQCL commands are used during initialization to calibrate DDR3 termination. A ZQCL command can be issued by writing 1 to the MC_CHANNEL_{0,1}_MC_DIMM_INIT_CMD.DO_ZQCL (Device 4,5,6, Function 0, Offset 15, bit[15]) field and it targets the DDR3 rank specified in the RANK field (bits[7:5]) of the same register. If the ZQCL commands are not issued in increasing populated rank order then ZQ calibration may not complete, causing the system to hang.

Implication: Due to this erratum the system may hang if writes to the MC_CHANNEL_{0,1}_MC_DIMM_INIT_CMD.DO_ZQCL field are not in increasing populated DDR3 rank order.

Workaround: A BIOS code change has been identified and may be implemented as a workaround for this erratum.

Status: For the steppings affected, see the Summary Table of Changes.

BF58. Back-To-Back Uncorrected Machine Check Errors Overwrite IA32_MC3_STATUS.MSCOD

Problem: When back-to-back uncorrected machine check errors occur that would both be logged in the IA32_MC3_STATUS MSR (40CH), the IA32_MC3_STATUS.MSCOD (bits [31:16]) field may reflect the status of the most recent error and not the first error. The rest of the IA32_MC3_STATUS MSR contains the information from the first error.

Implication: Software should not rely on the value of IA32_MC3_STATUS.MSCOD if IA32_MC3_STATUS.OVER (bit [62]) is set.

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes.

BF59. Memory Intensive Workloads With Core C6 Transitions Cause System Hang

Problem: Under a complex set of internal conditions, a system running a high cache stress and I/O workload combined with the presence of frequent core C6 transitions may result in a system hang.

Implication: Due to this erratum, the system may hang.

Workaround: It is possible for the BIOS to contain a workaround for this erratum.

Status: For the steppings affected, see the Summary Table of Changes.

BF60. Corrected Errors With Yellow Error Indication Overwritten By Other Corrected Errors

Problem: A corrected cache hierarchy data or tag error that is reported with IA32_MCi_STATUS.MCACOD (bits [15:0]) with value of 000x_0001_xxxx_xx01 (where x stands for zero or one) and a yellow threshold-based error status indication (bits [54:53] equal to 10B) may be overwritten by a corrected error with a no tracking indication (00B) or green indication (01B).

Implication: Corrected errors with a yellow threshold-based error status indication may be overwritten by a corrected error without a yellow indication.

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes.

BF61. Performance Monitor Events DCACHE_CACHE_LD and DCACHE_CACHE_ST Over Count

Problem: The performance monitor events DCACHE_CACHE_LD (Event 40H) and DCACHE_CACHE_ST (Event 41H) count cacheable loads and stores that hit the L1 cache. Due to this erratum, in addition to counting the completed loads and stores, the counter will incorrectly count speculative loads and stores that were aborted prior to completion.

Implication: The performance monitor events DCACHE_CACHE_LD and DCACHE_CACHE_ST may reflect a count higher than the actual number of events.

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes.

BF62. Rapid Core C3/C6 Transitions May Cause Unpredictable System Behavior

Problem: Under a complex set of internal conditions, cores rapidly performing C3/C6 transitions in a system with Intel® Hyper-Threading Technology enabled may cause a machine check error (IA32_MCi_STATUS.MCACOD = 0x0106), system hang or unpredictable system behavior.

Implication: This erratum may cause a machine check error, system hang or unpredictable system behavior.

Workaround: It is possible for the BIOS to contain a workaround for this erratum.

Status: For the steppings affected, see the Summary Table of Changes.

BF63. Performance Monitor Events INSTR_RETIRED and MEM_INST_RETIRED May Count Inaccurately

Problem: The performance monitor event INSTR_RETIRED (Event C0H) should count the number of instructions retired, and MEM_INST_ RETIRED (Event 0BH) should count the number of load or store instructions retired. However, due to this erratum, they may undercount.

Implication: The performance monitor event INSTR_RETIRED and MEM_INST_RETIRED may reflect a count lower than the actual number of events.

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes.

BF64. Page Fault Not Generated When PS Bit set to "1" in PML4E or PDPTE

Problem: On processors supporting Intel® 64 architecture, the PS bit (Page Size, bit 7) is reserved in PML4Es and PDPTEs. If the translation of the linear address of a memory access encounters a PML4E or a PDPTE with PS set to 1, a page fault should occur. Due to this erratum, PS of such an entry is ignored and no page fault will occur due to its being set.

Implication: Software may not operate properly if it relies on the processor to deliver page faults when reserved bits are set in paging-structure entries.

Workaround: Software should not set bit 7 in any PML4E or PDPTE that has Present Bit (Bit 0) set to "1".

Status: For the steppings affected, see the Summary Table of Changes.

BF65. CPURESET Bit Does Not Get Cleared

Problem: CPURESET (bit 10 of SYRE Device 8; Function 2; Offset 0CCH) allows the processor to be independently reset without assertion of the PLTRST# signal upon a 0 to 1 transition. The CPURESET bit does not get cleared and must be cleared by software.

Implication: The processor will not be reset if a 1 is written to this bit while it is already a one.

Workaround: The CPURESET bit must be cleared by software prior to setting it.

Status: For the steppings affected, see the Summary Table of Changes.

BF66. PHOLD Disable In MISCCTRLSTS Register Does Not Work

Problem: PHOLD Disable (PCI Hold Disable, bit [23] in MISCCTRLSTS Device 0; Function 0; Offset 188H) does not function as described. Setting this bit will not cause the processor to respond with Unsupported Request and log a fatal error upon receiving an Assert_PHOLD message from the PCH (Platform Controller Hub).

Implication: Due to this erratum, it is not possible to disable PHOLD requests from the PCH.

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes.

BF67. PCIe PMCSR Power State Field Incorrectly Allows Requesting D1 and D2 Power States

Problem: The PCIe PMCSR (Power Management Control and Status Register, Device 3,4,5,6; Function 0; Offset E4H) incorrectly allows the writing/requesting of the D1 and D2 Power States in the Power State field (bits[1:0] of PMCSR) when these states are not supported.

Implication: Given that the device does not support the D1 and D2 states, attempts to write those states should have been ignored. The PCIe port does not change power state from D0

or D3hot when the Power State bits are written to D1 or D2, so there is no functional impact to the PCIe port. However, the Power State field is incorrectly modified.

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes.

BF68. Concurrent Updates to a Segment Descriptor May be Lost

Problem: If a logical processor attempts to set the accessed bit in a code or data segment descriptor while another logical processor is modifying the same descriptor, both modifications of the descriptor may be lost.

Implication: Due to this erratum, updates to segment descriptors may not be preserved. Intel has not observed this erratum with any commercially available software or system.

Workaround: It is possible for the BIOS to contain a workaround for this erratum.

Status: For the steppings affected, see the Summary Table of Changes.

BF69. PMIs May be Lost During Core C6 Transitions

Problem: If a performance monitoring counter overflows and causes a PMI (Performance Monitoring Interrupt) at the same time that the core is entering C6, then the PMI may be lost.

Implication: PMIs may be lost during a C6 transition.

Workaround: It is possible for the BIOS to contain a workaround for this erratum.

Status: For the steppings affected, see the Summary Table of Changes.

BF70. Uncacheable Access To Monitored Address Range Prevent Future Triggering Of Monitor Hardware

Problem: It is possible that an address range which is being monitored via the MONITOR instruction could be written without triggering the monitor hardware. A read from the monitored address range which is issued as uncacheable (for example having the CR0.CD bit set) may prevent subsequent writes from triggering the monitor hardware. A write to the monitored address range which is issued as uncacheable, may not trigger the monitor hardware and may prevent subsequent writes from triggering the monitor hardware.

Implication: The MWAIT instruction will not exit the optimized power state and resume program flow if the monitor hardware is not triggered.

Workaround: It is possible for the BIOS to contain a workaround for this erratum.

Status: For the steppings affected, see the Summary Table of Changes.

BF71. BIST Results Additionally Reported After GETSEC[WAKEUP] or INIT-SIPI Sequence

Problem: BIST results should only be reported in EAX the first time a logical processor wakes up from the Wait-For-SIPI state. Due to this erratum, BIST results may be additionally reported after INIT-SIPI sequences and when waking up RLP's from the SENTER sleep state using the GETSEC[WAKEUP] command.

Implication: An INIT-SIPI sequence may show a non-zero value in EAX upon wakeup when a zero value is expected. RLP's waking up for the SENTER sleep state using the GETSEC[WAKEUP] command may show a different value in EAX upon wakeup than before going into the SENTER sleep state.

Workaround: If necessary software may save the value in EAX prior to launching into the secure environment and restore upon wakeup and/or clear EAX after the INIT-SIPI sequence.

Status: For the steppings affected, see the Summary Table of Changes.

BF72. Pending x87 FPU Exceptions (#MF) Signaled Earlier Than Expected

Problem: x87 instructions that trigger #MF normally service interrupts before the #MF. Due to this erratum, if an instruction that triggers #MF is executed while Enhanced Intel SpeedStep® Technology transitions, Intel® Turbo Boost Technology transitions, or Thermal Monitor events occur, the pending #MF may be signaled before pending interrupts are serviced.

Implication: Software may observe #MF being signaled before pending interrupts are serviced.

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes.

BF73. VM Exits Due to "NMI-Window Exiting" May Be Delayed by One Instruction

Problem: If VM entry is executed with the "NMI-window exiting" VM-execution control set to 1, a VM exit with exit reason "NMI window" should occur before execution of any instruction if there is no virtual-NMI blocking, no blocking of events by MOV SS, and no blocking of events by STI. If VM entry is made with no virtual-NMI blocking but with blocking of events by either MOV SS or STI, such a VM exit should occur after execution of one instruction in VMX non-root operation. Due to this erratum, the VM exit may be delayed by one additional instruction.

Implication: VMM software using "NMI-window exiting" for NMI virtualization should generally be unaffected, as the erratum causes at most a one-instruction delay in the injection of a virtual NMI, which is virtually asynchronous. The erratum may affect VMMs relying on deterministic delivery of the affected VM exits.

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes.

BF74. Malformed PCIe Packet Generated Under Heavy Outbound Load

Problem: When running the PCIe ports in a 2x8 configuration at 5.0GT/S speed with heavy outbound write traffic, malformed packets could be generated. The length in the header field will not match the actual payload size.

Implication: Due to this erratum, malformed PCIe packets could be transmitted.

Workaround: A BIOS code change has been identified and may be implemented as a workaround for this erratum.

Status: For the steppings affected, see the Summary Table of Changes.

BF75. VM Exits Due to EPT Violations Do Not Record Information About Pre-IRET NMI Blocking

Problem: With certain settings of the VM-execution controls VM exits due to EPT violations set bit 12 of the exit qualification if the EPT violation was a result of an execution of the IRET instruction that commenced with non-maskable interrupts (NMIs) blocked. Due to this erratum, such VM exits will instead clear this bit.

Implication: Due to this erratum, a virtual-machine monitor that relies on the proper setting of bit 12 of the exit qualification may deliver NMIs to guest software prematurely.

Workaround: It is possible for the BIOS to contain a workaround for this erratum.

Status: For the steppings affected, see the Summary Table of Changes.

BF76. S1 Entry May Cause Cores to Exit C3 or C6 C-State

Problem: Under specific circumstances, S1 entry may cause a logical processor to spuriously wake up from C3 or C6 and transition to a C0/S1 state. Upon S1 exit, these logical processors will be operating in C0.

Implication: In systems where S1 is used for power savings, customers may observe higher S1 power than expected and software may observe a different C-state on S1 exit than on S1 entry.

Workaround: It possible for the BIOS to contain a workaround for this erratum.

Status: For the steppings affected, see the Summary Table of Changes.

BF77. Multiple Performance Monitor Interrupts Possible On Overflow Of IA32_FIXED_CTR2

Problem: When multiple performance counters are set to generate interrupts on an overflow and more than one counter overflows at the same time, only one interrupt should be generated. However, if one of the counters set to generate an interrupt on overflow is the IA32_FIXED_CTR2 (MSR 30BH) counter, multiple interrupts may be generated when the IA32_FIXED_CTR2 overflows at the same time as any of the other performance counters.

Implication: Multiple counter overflow interrupts may be unexpectedly generated.

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes.

BF78. LBRs Not Initialized During Processor Power-On Reset

Problem: If a second reset is initiated during the power-on processor reset cycle, the LBRs (Last Branch Records) may not be properly initialized.

Implication: Due to this erratum, debug software may not be able to rely on the LBRs out of power-on reset.

Workaround: Ensure that the processor has completed its power-on reset cycle prior to initiating a second reset.

Status: For the steppings affected, see the Summary Table of Changes.

BF79. Unexpected Interrupts May Occur on C6 Exit If Using APIC Timer to Generate Interrupts

Problem: During a complex set of conditions, if the APIC timer is being used to generate interrupts, unexpected interrupts not related to the APIC timer may be signaled when a core exits the C6 power state. The APIC timer stops counting in C6 and as such isn't typically used to generate interrupts when the C6 core power state is enabled.

Implication: Unexpected interrupt vectors could be sent from the APIC to a logical processor. Intel has not observed this erratum with any commercially available software.

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes.

BF80. Package C6 Exit With Memory In Self-Refresh When Using DDR3 RDIMM Memory Leads to Hang

Problem: When using DDR3 RDIMM memory and exiting from the C6 low power state with memory in self-refresh the CS (Chip Select) signals may remain in tri-state during tSTAB (CLK Stabilization time) thus violating the JEDEC Standard: *Definition of the SSTE32882 Registering Clock Driver with Parity and Quad Chip Selects for DDR3*

RDIMM Applications. As detailed in the JEDEC specification the CS signals should transition from tri-state to high to exit the Clock Stopped Power Down Mode.

Implication: When this erratum occurs the processor may hang.

Workaround: It is possible for the BIOS to contain a workaround for this erratum.

Status: For the steppings affected, see the Summary Table of Changes.

BF81. LBR, BTM or BTS Records Have Incorrect Branch From Information After EIST Transition, T-states, C1E, Or Adaptive Thermal Throttling

Problem: The "From" address associated with the LBR (Last Branch Record), BTM (Branch Trace Message) or BTS (Branch Trace Store) may be incorrect for the first branch after an EIST (Enhanced Intel® SpeedStep Technology) transition, T-states, C1E (C1 Enhanced), or Adaptive Thermal Throttling.

Implication: When the LBRs, BTM or BTS are enabled, some records may have incorrect branch "From" addresses for the first branch after an EIST transition, T-states, C1E, or Adaptive Thermal Throttling.

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes.

BF82. PECI GetTemp() Reads May Return Invalid Temperature Data in Package C6 State

Problem: The PECI (Platform Environment Control Interface) GetTemp() command may occasionally return incorrect temperature data.

Implication: The temperature data reported over PECI should always be a negative value and represents a delta below the onset of TCC (thermal control circuit) activation, as indicated by PROCHOT#. The PECI GetTemp() command may occasionally return incorrect temperature data when the processor is in the package C6 state. The error occurrence rate and returned processor temperature values are random including both hot and cold readings. Note that this error may cause the processor to return positive PECI temperature values that may not necessarily be indicative of a thermal event requiring an immediate shutdown.

Workaround: Intel recommends discarding processor temperature values less than -100 or greater than 0, and the use of appropriate temperature smoothing filters in the range -100 to 0 to minimize fan speed fluctuations, if any, due to these errors. Intel does not recommend initiating system shutdown solely based on PECI readings. For systems using the PECI temperature data to facilitate system shutdown, Intel recommends initiating a shutdown only if a PECI value of 0 is returned over three consecutive PECI temperature reads.

Status: For the steppings affected, see the Summary Table of Changes.

BF83. VMX-Preemption Timer Does Not Count Down At Rate Specified

Problem: The VMX-preemption timer should count down by 1 every time a specific bit in the TSC (Time Stamp Counter) changes. (This specific bit is indicated by IA32_VMX_MISC bits [4:0] (0x485h) and has a value of 5 on the affected processors.) Due to this erratum, the VMX-preemption timer may instead count down at a different rate and may do so only intermittently.

Implication: The VMX-preemption timer may cause VM exits at a rate different from that expected by software.

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes.

BF84. Multiple Performance Monitor Interrupts Possible On Overflow Of Fixed Counter 0

Problem: The processor can be configured to issue a PMI (performance monitor interrupt) upon overflow of the IA32_FIXED_CTR0 MSR (309H). A single PMI should be observed on overflow of IA32_FIXED_CTR0, however multiple PMIs are observed when this erratum occurs. This erratum only occurs when IA32_FIXED_CTR0 overflows and the processor and counter are configured as follows:

- Intel® Hyper-Threading Technology is enabled
- IA32_FIXED_CTR0 local and global controls are enabled
- IA32_FIXED_CTR0 is set to count events only on its own thread (IA32_FIXED_CTR_CTRL MSR (38DH) bit [2] = '0)
- PMIs are enabled on IA32_FIXED_CTR0 (IA32_FIXED_CTR_CTRL MSR bit [3] = '1)
- Freeze_on_PMI feature is enabled (IA32_DEBUGCTL MSR (1D9H) bit [12] = '1)

Implication: When this erratum occurs there may be multiple PMIs observed when IA32_FIXED_CTR0 overflows

Workaround: Disable the FREEZE_PERFMON_ON_PMI feature in IA32_DEBUGCTL MSR (1D9H) bit [12].

Status: For the steppings affected, see the Summary Table of Changes.

BF85. SVID and SID Of Devices 8 And 16 Only Implement Bits [7:0]

Problem: Bits [15:8] of SVID (Subsystem Vendor ID, Offset 2CH) and the SID (Subsystem Device ID, Offset 2EH) of devices 8 and 16 are not implemented. Only the lower bits [7:0] of these registers can be written to, though the PCI-e specification indicates that these are 16-bit registers.

Implication: Only bits [7:0] of SVID and SID can be written. Bits [15:8] will always be read as 0.

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes.

BF86. No_Soft_Reset Bit In PMCSR Does Not Operate As Expected

Problem: When the No_Soft_Reset bit in the Power Management Control and Status Register (PMCSR; Bus 0; Devices 0, 3, 4, 5; Function 0; Offset 0xE4; Bit 3) is cleared the device should perform an internal reset upon transitioning from $D3_{hot}$ to D0. Due to this erratum the device does not perform an internal reset upon transitioning from $D3_{hot}$ to D0.

Implication: When the No_Soft_reset bit in the PMCSR register is set or cleared no internal reset of the device will be preformed when transitioning from $D3_{hot}$ to D0.

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes.

BF87. VM Exits Due To LIDT/LGDT/SIDT/SGDT Do Not Report Operand Size

Problem: When a VM exit occurs due to a LIDT, LGDT, SIDT, or SGDT instruction with a 32-bit operand, bit 11 of the VM-Exit Instruction-Information Field should be set to 1. Due to this erratum, this bit is instead cleared to 0 (indicating a 16-bit operand).

Implication: Virtual Machine Monitors cannot rely on bit 11 of the VM-Exit Instruction-Information Field to determine the operand size of the instruction causing the VM exitVirtual Machine Monitors cannot rely on bits 13:12 of the VM-Exit Instruction-Information Field to determine the operand size of the instruction causing the VM exit.

Workaround: Virtual Machine Monitor software may decode the instruction to determine operand size.

Status: For the steppings affected, see the Summary Table of Changes.

BF88. Performance Monitoring Events STORE_BLOCKS.NOT_STA and STORE_BLOCKS.STA May Not Count Events Correctly

Problem: Performance Monitor Events STORE_BLOCKS.NOT_STA and STORE_BLOCKS.STA should only increment the count when a load is blocked by a store. Due to this erratum, the count will be incremented whenever a load hits a store, whether it is blocked or can forward. In addition this event does not count for specific threads correctly.

Implication: If Intel® Hyper-Threading Technology is disabled, the Performance Monitor events STORE_BLOCKS.NOT_STA and STORE_BLOCKS.STA may indicate a higher occurrence of loads blocked by stores than have actually occurred. If Intel Hyper-Threading Technology is enabled, the counts of loads blocked by stores may be unpredictable and they could be higher or lower than the correct count.

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes.

BF89. Storage Of PEBS Record Delayed Following Execution Of MOV SS Or STI

Problem: When a performance monitoring counter is configured for PEBS (Precise Event Based Sampling), overflow of the counter results in storage of a PEBS record in the PEBS buffer. The information in the PEBS record represents the state of the next instruction to be executed following the counter overflow. Due to this erratum, if the counter overflow occurs after execution of either MOV SS or STI, storage of the PEBS record is delayed by one instruction.

Implication: When this erratum occurs, software may observe storage of the PEBS record being delayed by one instruction following execution of MOV SS or STI. The state information in the PEBS record will also reflect the one instruction delay.

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes.

BF90. Performance Monitoring Event FP_MMX_TRANS_TO_MMX Not Counting Some Transitions

Problem: Performance Monitor Event FP_MMX_TRANS_TO_MMX (Event CCH, Umask 01H) counts transitions from x87 Floating Point (FP) to MMX™ instructions. Due to this erratum, if only a small number of MMX instructions (including EMMS) are executed immediately after the last FP instruction, a FP to MMX transition may not be counted.

Implication: The count value for Performance Monitoring Event FP_MMX_TRANS_TO_MMX may be lower than expected. The degree of undercounting is dependent on the occurrences of the erratum condition while the counter is active. Intel has not observed this erratum with any commercially available software.

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes.

BF91. INVLPG Following INVEPT Or INVVPID Fail to Flush All Translations For Large Page

Problem: This erratum applies if the address of the memory operand of an INVEPT or INVVPID instruction resides on a page larger than 4KBytes and either (1) that page includes the low 1 MBytes of physical memory; or (2) the physical address of the memory operand matches an MTRR that covers less than 4 MBytes. A subsequent execution of INVLPG that targets the large page and that occurs before the next VM-entry instruction may fail to flush all TLB entries for the page. Such entries may persist in the TLB until the next VM-entry instruction.

Implication: Accesses to the large page between INVLPG and the next VM-entry instruction may incorrectly use translations that are inconsistent with the in-memory page tables.

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes.

BF92. PECI Bus Tri-Stated After System Reset

Problem: During power-up, the processor may improperly assert the PECI (Platform Environment Control Interface) pin. This condition is cleared as soon as Bus Clock starts toggling. However, if the PECI host (also referred to as the master or originator) incorrectly determines this asserted state as another PECI host initiating a transaction, it may release control of the bus resulting in a permanent tri-state condition.

Implication: Due to this erratum, the PECI host may incorrectly determine that it is not the bus master and consequently PECI commands initiated by the PECI software layer may receive incorrect/invalid responses.

Workaround: To workaround this erratum the PECI host should pull the PECI bus low to initiate a PECI transaction.

Status: For the steppings affected, see the Summary Table of Changes.

BF93. LER MSRs May Be Unreliable

Problem: Due to certain internal processor events, updates to the LER (Last Exception Record) MSRs, MSR_LER_FROM_LIP (1DDH) and MSR_LER_TO_LIP (1DEH), may happen when no update was expected.

Implication: The values of the LER MSRs may be unreliable.

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes.

BF94. Multiple ECC Errors Result in Logging Incorrect Syndrome

Problem: When multiple correctable DRAM ECC errors occur the processor may log the syndrome of a previous error or may log an unknown value.

Implication: Due to this erratum, the value logged in the IA32_MCi_MISC MSR will not correspond to the most recent error.

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes.

BF95. MCi_Status Overflow Bit Incorrectly Set On Single Instance Of DTLB Error

Problem: A single Data Translation Look Aside Buffer (DTLB) error can incorrectly set the Overflow (bit [62]) in the MCi_Status register. A DTLB error is indicated by MCA error

code (bits [15:0]) appearing as binary value, 000x 0000 0001 0100, in the MCi_Status register.

Implication: Due to this erratum, the Overflow bit in the MCi_Status register may not be an accurate indication of multiple occurrences of DTLB errors. There is no other impact to normal processor functionality.

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes.

BF96. Debug Exception Flags DR6.B0-B3 Flags Incorrect For Disabled Breakpoints

Problem: When a debug exception is signaled on a load that crosses cache lines with data forwarded from a store and whose corresponding breakpoint enable flags are disabled (DR7.G0-G3 and DR7.L0-L3), the DR6.B0-B3 flags may be incorrect.

Implication: The debug exception DR6.B0-B3 flags may be incorrect for the load if the corresponding breakpoint enable flag in DR7 is disabled.

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes.

BF97. Intel® QuickPath Memory Controller Hangs If Uncorrectable ECC Errors Occur On Both Channels In Mirror Channel Mode

Problem: If an uncorrectable ECC error occurs on the mirrored channel before an uncorrectable ECC error on the other channel can be resolved, the Intel QuickPath Memory Controller will hang without an uncorrectable ECC error being logged.

Implication: The processor may hang and not report the error when uncorrectable ECC errors occur in close proximity on both channels in a mirrored channel pair. No uncorrectable ECC error will be logged in the machine check banks.

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes.

BF98. Simultaneous Correctable ECC Errors on Different Memory Channels With Patrol Scrubbing Enabled May Result in Incorrect Information Being Logged

Problem: When a correctable patrol scrub ECC error occurs simultaneously with a correctable system read ECC error on different memory channels, IA32_MCi_STATUS and IA32_MCi_MISC should log the system read error. Due to this erratum IA32_MCi_MISC may incorrectly contain the patrol scrub error information and the IA32_MCi_ADDR may not be correct.

Implication: IA32_MCi_MISC and IA32_MCi_STATUS information may be inconsistent. IA32_MCi_ADDR may be incorrect.

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes.

BF99. In Memory Lockstep Mode Per DIMM Correctable ECC Errors Logged Incorrectly

Problem: In lockstep mode, if a correctable ECC error occurs on DIMMx of one DDR3 channel, the corrected error count, MC_COR_ECC_CNT {0, 1, 2, 3} will be incremented on both channels for that DIMM, instead of just the channel and DIMM on which the error has occurred.

Implication: Neither MC_COR_ECC_CNT {0, 1, 2, 3} nor IA32_MCi_MISC Channel/DIMM fields can be used to determine which channel and DIMM the error occurred on.

Workaround: Information logged into IA32_MCi_MISC syndrome field can be used to decode and identify the actual failing DIMM.

Status: For the steppings affected, see the Summary Table of Changes.

BF100. Memory Controller Address Parity Error Injection Does Not Work Correctly

Problem: When MC_CHANNEL_{0,1,2}_ECC_ERROR_INJECT.INJECT_ADDR_PARITY bit [4] = 1 an error may be injected on any command on the channel and not just RD or WR CAS commands that match MC_CHANNEL_{0,1,2}_ADDR_MATCH.

Implication: Address parity error injection cannot be used to reliably target a DIMM or memory location within a channel. When the address parity errors occur, the IA32_MCi_MISC register reflects the DIMM ID of the DIMM that detected error and not necessarily the DIMM that was targeted by the error injection settings.

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes.

BF101. Failing DIMM ID Incorrect In 2DPC Configuration When Mirroring Enabled

Problem: When redundancy is lost in the 2DPC (two DIMMs Per Channel) configuration, MC_SMI_SPARE_DIMM_ERROR_STATUS CSR bits [13:12] (REDUNDANCY_LOSS_FAILING_DIMM) may indicate the incorrect failing DIMM ID. The 2DPC configuration is indicated when MC_CHANNEL_{0,1}_DIMM_INIT_PARAMS CSR bit [24] (THREE_DIMMS_PRESENT) is 0.

Implication: The failing DIMM ID may be reported incorrectly in the 2DPC configuration when mirroring is enabled. The 3DPC configuration is not affected.

Workaround: Only use the value in bit [13] to determine the failing DIMM ID in the non-3PDC configurations when mirroring is enabled. This workaround will show correct results for both the 1DPC and 2DPC configurations.

Status: For the steppings affected, see the Summary Table of Changes.

BF102. ISSUEONCE Bit In MC_SCRUB_CONTROL Register Not Working Correctly

Problem: When ISSUEONCE (bit [25]) in the MC_SCRUB_CONTROL register (Device 3, Function 2, Offset 4CH) is set, the memory controller should issue one patrol scrub. Due to this erratum, scrubbing requests continue to be issued.

Implication: ISSUEONCE bit in MC_SCRUB_CONTROL register does not work correctly.

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes.

BF103. Memory Thermal Throttling May Not Work as Expected in Lockstep Channel Mode

Problem: Thermal Throttling on a channel that is in lockstep mode affects all channels in order to maintain lockstep requirements. If throttling parameters are modified at different times during runtime, throttling on one channel is likely to be out of phase with throttling on other channels. Throttling that is out of phase will result in more throttling than anticipated. If the throttling duty cycle exceeds 50%, certain phase relationships can result in persistent memory traffic blockage.

Implication: Runtime modification of throttling parameters may result in a system hang.

Workaround: Since Thermal Throttling on one channel affects all channels while in lockstep mode, throttling should only be applied to one channel.

Status: For the steppings affected, see the Summary Table of Changes.

BF104. A String Instruction that Re-maps a Page May Encounter an Unexpected Page Fault

An unexpected page fault (#PF) may occur for a page under the following conditions:

- The paging structures initially specify a valid translation for the page.
- Software modifies the paging structures so that there is no valid translation for the page (e.g., by clearing to 0 the present bit in one of the paging-structure entries used to translate the page).
- An iteration of a string instruction modifies the paging structures so that the translation is again a valid translation for the page (e.g., by setting to 1 the bit that was cleared earlier).
- A later iteration of the same string instruction loads from a linear address on the page.

Problem: Software did not invalidate TLB entries for the page between the first modification of the paging structures and the string instruction. In this case, the load in the later iteration may cause a page fault that indicates that there is no translation for the page (e.g., with bit 0 clear in the page-fault error code, indicating that the fault was caused by a not-present page).

Implication: Software may see an unexpected page fault that indicates that there is no translation for the page. Intel has not observed this erratum with any commercially available software or system.

Workaround: Software should not update the paging structures with a string instruction that accesses pages mapped the modified paging structures.

Status: For the steppings affected, see the Summary Table of Changes.

BF105. Transactions To Address Above TOCM Not Setting Master Abort Error Bit In IIOERRST

Problem: Inbound transactions to addresses above TOCM (top of physical addressable memory—- 2^40) are master aborted, but the IIOERRST.C4 (Device 8; Function 2; Offset 300H; bit [4]) bit is not being set to record the error if Intel® VT-d (Intel® Virtualization Technology for Directed I/O) is disabled.

Implication: Master abort address error is not recorded in IIOERRST for inbound accesses to addresses above TOCM.

Workaround: If Intel® VT-d is disabled, then program the HPA_LIMT field (bits [7:4] of VTGENCTRL Device 8; Function 0; Offset 184H) to the maximum value.

Status: For the steppings affected, see the Summary Table of Changes.

BF106. HPA_LIMIT Check Violations Not Logged As Error

Problem: Accesses above HPA_LIMIT (bits[7:4] of VTGENCTRL Device 8; Function 0; Offset 184H) are not logged as an error if the access occurs after an Intel® Vt-d(Intel® Virtualization Technology for Directed I/O) address translation. A UR (Unsupported Request) response is returned to the requester.

Implication: HPA_LIMIT check violations after Intel® VT-d (Intel® Virtualization Technology for Directed I/O) address translation won’t cause an error to be logged.

Workaround: None

Status: For the steppings affected, see the Summary Table of Changes.

BF107. Memory Writes To Certain Address Range Considered Advisory Non-Fatal

Problem: Accesses above TOCM (top of physical addressable memory— 2^40) are required to be master aborted and the error is to be logged in UNCERRSTS (Device 0, 3, 4, 5, or 6; Function 0; Offset 104H). The accesses are aborted and logged. However, when the severity of master-abort UR (Unsupported Request) is set to non-fatal, memory write accesses are not to be considered advisory non-fatal. Instead, they should be considered normal non-fatal. This erratum affects the range of addresses from 0x8_0000_0000_0000 to 0xF_FFFF_FFFF_FFFF and incorrectly sets the advisory non-fatal status bit in CORERRSTS (Device 0, 3, 4, 5, or 6; Function 0; Offset 110H; bit [13]). This erratum does not arise if UR severity is set to Fatal.

Implication: The advisory non-fatal status bit is set when it should remain clear.

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes.

BF108. PCIe Packet Detect Power Saving Mode Causes Packet Loss Or Corruption

Problem: When Packet Detect mode is enabled by setting PKTDETEN (bit[29] of CGCTRL (Device 8; Function 3; Offset 40H) PCIe packet loss or corruption can occur.

Implication: Loss or corruption of PCIe packets.

Workaround: Set DLYTOGATE (bits[9:0] of CGCTRL Device:8; Function 3; Offset 40H) to 012CH (300 clocks) when enabling Packet Detect mode.

Status: For the steppings affected, see the Summary Table of Changes.

BF109. Intel® VT-d Translated Write Transactions Targeting Interrupt Address Range Blocked But Not Recorded as Errors

Problem: The Intel® Virtualization Technology for Direction I/O engine blocks translated write transactions to the interrupt address range (0FEExxxxxH) but does not record the error.

Implication: For translated write transactions to the interrupt address range (0FEExxxxxH), the transaction is blocked, but evidence of the blocked transaction will not be available in any error logging register.

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes.

BF110. ERRSID Not Logging ReqID For Inbound PCIe Error Messages

Problem: The ERRSID (Error Source Identification) Register (Devices: 0, 3, 4 5, 6; Function: 0; Offset: 134H) does not log the ReqID (Requester ID) for inbound ERR_FATAL, ERR_NONFATAL or ERR_COR messages.

Implication: While processor internally generated error messages will have their ReqID logged correctly, externally generated error messages will not have their ReqID logged.

Workaround: None identified. Software needs to read the downstream device's error log to identify the source of the error.

Status: For the steppings affected, see the Summary Table of Changes.

BF111. MSI With Greater Than One DWord Payload Not Logged As Error In Proper XPUNCERRSTS Register

Problem: When the PCIe ports are used in a x8 or x16 configuration an MSI (Message Signaled Interrupt) with greater than one DWORD of payload should be logged as an error condition in bit[8] of XPUNCERRSTS (Devices: 3, 5; Function 0; Offset 208H) on the port that received the MSI. Due to this erratum, the error indication may be logged in any of the XPUNCERRSTS registers (Devices: 3, 4, 5, 6; Function: 0; Offset 208H) and may not correspond to the port where the MSI was received.

Implication: The expected error indication is not available on the port that received the malformed MSI message.

Workaround: None identified. Software may read bit [8] of the XPUNCERRSTS register (Device 3, 4, 5, 6; Function 0; Offset 208H) on all partner ports to see the status.

Status: For the steppings affected, see the Summary Table of Changes.

BF112. Error Pin Status Register For Pins SYS_ERR_STAT[2:0] Not Updated In Mode 01b

Problem: ERRPINST (Error Pin Status) (Device: 8; Function: 2; Offset: 0A8H) is not updated when Error[n] Pin Assertion Control in ERRPINCTL (Error Pin Control) (Device: 8; Function: 2; Offset: 0A4H) is set to mode 01b.

Implication: Due to this erratum, ERRPINST does not reflect true state of the error pins if Error[n] Pin Assertion Control is set to mode 01b. Mode 10b works properly.

Workaround: Software can read the ERRPINDAT (Error Pin Data) (Device: 8; Function: 2; Offset: 0ACH) to determine the state of the SYS_ERR_STAT[2:0] pins when Error[n] Pin Assertion Control is 01b.

Status: For the steppings affected, see the Summary Table of Changes.

BF113. Writes to SDOORBELL or B2BDOORBELL in Conjunction With Inbound Access to NTB MMIO Space May Hang System

Problem: A posted write targeting the SDOORBELL (Offset 64H) or B2BDOORBELL (Offset 140H) MMIO registers in the region define by Base Address Register PB01BASE (Bus 0; Device 3; Function 0: Offset 10H) or SB01BASE (Bus M; Device 0; Function 0; Offset 10H) may hang the system. This system hang may occur if the NTB (Non-Transparent Bridge) is processing a transaction from the secondary side of the NTB that is targeting the NTB shared MMIO registers or targeting the secondary side configuration registers when the write arrives.

Implication: The system may hang if the processor writes to the local SDOORBELL or B2BDOORBELL register at the same time that the NTB is processing an inbound transaction.

Workaround: In NTB/NTB (back-to-back) mode, do not use the B2BDOORBELL to send interrupts from the local to remote host. Instead, configure one of the following local register pairs to point to the remote SB01BASE region:

- PB23BASE (Device: 3; Function: 0; Offset: 18H) and PBAR2XLAT (Offset 10H) from PB01BASE or SB01BASE regions;
- PB45BASE (Device: 3; Function: 0; Offset: 20H) and PBAR4XLAT(Offset 18H) from PB01BASE, or SB01BASE regions;

The local host may then write directly to the PDOORBELL (Offset 60H) from the PB23BASE/PB45BASE region defined above.

In NTB/RP (bridge to root port) mode, the SDOORBELL register cannot be used by the processor on the primary side of the NTB to interrupt the processor on the secondary side. Instead, dedicate a BAR and XLAT pair, either PB23BASE/PBAR2XLAT or PB45BASE/PBAR4XLAT, to generate an interrupt directed directly into the MSI/MSIx (Message Signaled Interrupt) interrupt range on the remote processor. The device driver or client on the remote host must point the appropriate PBARnXLAT register to its MSI/MSIx interrupt range. The processor on the primary side can then write the MSI/MSIx interrupt to the dedicated BAR which will be translated by the NTB to the MSI/MSIx region of the secondary side's processor.

Status: For the steppings affected, see the Summary Table of Changes

BF114. Enabling Demand/Patrol Scrubs Along With WMM May Cause Unpredictable System Behavior

Problem: Under a specific set of conditions, enabling Demand scrubs (Bit 6, DEMAND_SCRUB_EN of MS_SSRCONTROL CSR ADDR 48H) and/or Patrol Scrubs (Bits 1:0, SSR_MODE of MS_SSRCONTROL CSR ADDR 48H) along with WMM (Write Major Mode) may cause unpredictable system behavior.

Implication: Due to this erratum unpredictable system behavior may occur.

Workaround: WMM must be disabled if Demand and/or Patrol Scrubs are enabled.

Status: For the steppings affected, see the Summary Table of Changes

BF115. 2MB Page Split Lock Accesses Combined With Complex Internal Events May Cause Unpredictable System Behavior

Problem: A 2MB Page Split Lock (a locked access that spans two 2MB large pages) coincident with additional requests that have particular address relationships in combination with a timing sensitive sequence of complex internal conditions may cause unpredictable system behavior.

Implication: This erratum may cause unpredictable system behavior. Intel has not observed this erratum with any commercially available software.

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes

BF116. L1 Cache Uncorrected Errors May be Recorded as Correctable in 16K Mode

Problem: When the L1 Cache is operating in 16K redundant parity mode and a parity error occurs on both halves of the duplicated cache on the same cacheline, an uncorrectable error should be logged. Due to this erratum, the uncorrectable error will be recorded as correctable, however a machine check exception will be appropriately taken in this case.

Implication: Due to this erratum, the IA32_MCi_STATUS.UC bit will incorrectly contain a value of 0 indicating a correctable error.

Workaround: It is possible for the BIOS to contain a workaround for this erratum.

Status: For the steppings affected, see the Summary Table of Changes

BF117. Remote MEMRD Peer-to-Peer PCIe Transactions with Non-Contiguous Byte Enables May Return UnExpected Data

Problem: A MEMRD (memory read) transaction with a length of two DWORDs that is forwarded between sockets (remote peer-to-peer) without contiguous byte enables may not return the expected data. Peer-to-Peer traffic within the same socket operates correctly.

Implication: Remote peer-to-peer MEMRD transactions with a length of two DWORDs must have contiguous byte enables. This is in addition to the requirements specified in the PCI Express specification. Intel has not observed this erratum with any commercially available system.

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes

BF118. NTB Endpoint Does Not Implement LNKCON2 or LNKSTS Registers

Problem: When using the NTB (Non-Transparent Bridge) as a PCIe endpoint connected to a remote root port (programmed on a bus M), the LNKCON2 (Bus: M; Device: 3; Function: 0; Offset: C0H) and LNKSTS2 (Bus: M; Device: 3; Function: 0; Offset: C2H) registers are not visible to the remote host. Inbound PCIe configuration write cycles that target these registers will be silently dropped and reads will return 0.

Implication: NTB Endpoint extended PCI configuration space is not compliant with the PCI specification as the LNKCON2 and LNKSTS2 registers are not implemented. The remote host can not write these registers.

Workaround: The LNKCON2 (Bus: 0; Device: 3; Function: 0; Offset: 1C0H) and LNKSTS2 (Bus: 0; Device: 3; Function: 0; Offset: 1C2H) registers can be written by the local host connected to the primary side on the NTB.

Status: For the steppings affected, see the Summary Table of Changes

BF119. Source ID for Errors Internally Detected by PCIe Devices 3, 4, 5, 6 is Logged Incorrectly

Problem: In a dual processor system, the Source ID for errors internally detected by PCIe root port controllers (PCI Devices: 3, 4, 5, 6) in the non-legacy processor (not directly connected to PCH) is logged incorrectly in the associated ERRSID register (Bus: IIOBUSNO; Device: 3, 4, 5, 6; Function: 0; Offset: 134H). A bus number of 0 will be logged in ERRSID instead of IIOBUSNO (Device: 8; Function: 0; Offset: 10AH) – the BIOS programmed PCI bus number for the non-legacy processor's PCI devices.

Implication: The ERRSID registers do not log the correct bus number for internally detected errors in the non-legacy processor.

Workaround: Software may read the UNCERRSTS (Device: 3, 4, 5, 6; Function: 0 Offset: 104H) and CORERRSTS (Device: 3, 4, 5, 6; Function: 0; Offset: 110H) registers to determine if ports 3, 4, 5, 6 detected an error as well as the type of error.

Status: For the steppings affected, see the Summary Table of Changes

BF120. Intel® VT-d: Address Remapping Error When DMA/Interrupt Remapping is Active

Problem: With Intel® VT-d enabled, when software updates the root table pointer or interrupt remapping table pointer while DMA/interrupt remapping is active, it is possible that the address used to access the page-table structure for DMA requests or interrupt remapping could be corrupted and cause an address remapping error.

Implication: Software can not update the root table pointer or interrupt remapping table pointer while DMA/interrupt remapping is active.

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes

BF121. PCIe Link Bit Errors Present During L0s Entry May Cause the System to Hang During L0s Exit

Problem: During L0s entry PCIe link bit errors may be generated due to a slow shutdown response from the PCIe analog circuits. As a result, the PCIe analog circuits may now take longer to establish bit lock during the L0s exit sequence. In some cases bit lock may not be achieved and may result in a system hang.

Implication: While exiting from L0s the PCIe bus may go into recovery mode. At the 5 GB/s rate system hangs may occur while exiting from L0s; however the hangs have not been seen on commercially available systems.

Workaround: A BIOS code change has been identified and may be implemented as a workaround for this erratum.

Status: For the steppings affected, see the Summary Table of Changes

BF122. Electrical Idle Exit Sequence Incorrect Upon Exit From PCIe L0s State

Problem: When running at 5.0 GT/s, the PCIe ports will not transmit the correct EIEOS (electrical idle exit ordered set) when exiting from the L0s link state. The first two symbols (K28.5 and the 1st K28.7) of the ordered set will not be seen on the link. The EIEOSs sent periodically during link training are not impacted by this erratum.

Implication: The receiver may be slightly delayed in recognizing the exit from the L0s state. No performance impact has been observed in test systems.

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes

BF123. PCIe AER HDRLOG Register Does Not Record Header of Packets with Errors

Problem: The PCIe AER (Advanced Error Reporting) HDRLOG register (Device: 0, 3, 4, 5, 6; Function: 0; Offset 120H) does not record the header of the packets with errors.

Implication: The HDRLOG register will not contain information useful to debug errors reported by the PCIe link's AER mechanism.

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes

BF124. Aborted Inbound PCIe Memory Reads Return Incorrect Lower Address Field in the Completion

Problem: Inbound PCIe Memory Reads which are aborted will return invalid data in the Lower Address Field (Bits [6:0] of 2nd DWORD) of the completion response. The completion status will correctly reflect an error occurred.

Implication: There will be incorrect data in the lower address field of a PCIe completion packet which has a status of CA (Completer Abort).

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes

BF125. Clearing the Memory_Space_Enable Bit For NTB Does not Disable Accesses to All MMIO Regions

Problem: Accesses to an NTB (Non-Transparent Bridge) region defined by PB23BASE (Device: 3; Function: 0; Offset 18H) or PB45BASE (Device: 3; Function: 0; Offset 20H) Memory Mapped I/O (MMIO) regions can not be disabled by clearing the Memory_Space_Enable bit in PCICMD (Device: 3; Function: 0; Offset: 04H; bit[1]) after PBAR23SZ (Device: 3;

Function: 0; Offset: D0H) or PBAR45SZ (Device: 3, Function: 0; Offset: D1H) registers have been programmed to a value other than 0.

Implication: Due to this erratum, accesses to some MMIO regions when the NTB Memory_Space_Enable bit is clear may still be claimed.

Workaround: A BIOS code change has been identified and may be implemented as a workaround for this erratum.

Status: For the steppings affected, see the Summary Table of Changes

BF126. PCIe Slave Loopback Mode May Exit Prematurely

Problem: The PCIe ports will exit slave loopback mode after receiving 3 consecutive EIOSs (Electrical Idle Ordered Sets) rather than 4 consecutive EIOS as called for in the PCI Express* Base Specification.

Implication: The PCIe ports exit slave loopback mode prematurely after 3rd consecutive EIOS is received. There is no other side affect due to this erratum.

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes.

BF127. FP Data Operand Pointer May Be Incorrectly Calculated After an FP Access Which Wraps a 4-Gbyte Boundary in Code That Uses 32-Bit Address Size in 64-Bit Mode

Problem: The FP (Floating Point) Data Operand Pointer is the effective address of the operand associated with the last non-control FP instruction executed by the processor. If an 80-bit FP access (load or store) uses a 32-bit address size in 64-bit mode and the memory access wraps a 4-Gbyte boundary and the FP environment is subsequently saved, the value contained in the FP Data Operand Pointer may be incorrect.

Implication: Due to this erratum, the FP Data Operand Pointer may be incorrect. Wrapping an 80-bit FP load around a 4-Gbyte boundary in this way is not a normal programming practice. Intel has not observed this erratum with any commercially available software.

Workaround: If the FP Data Operand Pointer is used in a 64-bit operating system which may run code accessing 32-bit addresses, care must be taken to ensure that no 80-bit FP accesses are wrapped around a 4-Gbyte boundary.

Status: For the steppings affected, see the Summary Table of Changes.

BF128. PCIe Ports Violate $T_{TX\text{-}Rise\text{-}Fall}$ Specification When Operating at 2.5GT/s

Problem: The PCIe transmitter rise/fall time is the same regardless of operating speed. When operating at 2.5GT/S, the PCIe transmitters will not meet the TTX-Rise-Fall specification as the rise/fall time will be too fast.

Implication: PCIe ports violate TTX-Rise-Fall specification at 2.5GT/s. Operation at 5.0GT/s is compliant with the PCIe electrical specification.

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes.

BF129. NTB/RP Link Will Send Extra TS2 Ordered Set During Link Training

Problem: The NTB (Non-Transparent Bridge) when operating in NTB/RP (Root Port) mode will send a superfluous TS2 ordered set after transitioning to the CONFIGURATION.IDLE state during link training. This TS2 ordered set may contain invalid capability data.

Implication: NTB/RP Link will transmit a TS2 ordered set after transitioning to the CONFIGURATION.IDLE state. No impact expected for specification compliant PCIe partners. Specification compliant PCIe link partners will have transitioned to CONFIGURATION.IDLE before this ordered set is sent and will ignore it.

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes.

BF130. PECI PCIConfigRd() Followed by a GetTemp() May Cause System Hang in Package C6 State

Problem: The PECI (Platform Environment Control Interface) PCIConfigRd() command immediately followed by a PECI GetTemp() command may result in a system hang.

Implication: When the processor is in the package C6 state, a PECI PCIConfigRd() command immediately followed by a GetTemp() command may result in a system hang. If PCIConfigRd() is never used, then this erratum will not be observed.

Workaround: A PCIConfigWr() command should be issued in between PCIConfigRd() and GetTemp() commands. The PCIConfigWr() command may be issued to any valid PECI writable CSR address, including a benign CSR address such as 0x23058.

Status: For the steppings affected, see the Summary Table of Changes.

BF131. IO_SMI Indication in SMRAM State Save Area May Be Lost

Problem: The IO_SMI bit (bit 0) in the IO state field at SMRAM offset 7FA4H is set to "1" by the processor to indicate a System Management Interrupt (SMI) is either taken immediately after a successful I/O instruction or is taken after a successful iteration of a REP I/O instruction. Due to this erratum, the setting of the IO_SMI bit may be lost. This may happen under a complex set of internal conditions with Intel® Hyper-Threading Technology enabled and has not been observed with commercially available software.

Implication: Due to this erratum, SMI handlers may not be able to identify the occurrence of I/O SMIs.

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes.

BF132. TSC Values When Observed Cross-Socket May Be Out of Sync After a Warm Reset

Problem: In a two socket platform with package C6 enabled, the TSC (Time Stamp Counter) cross-socket values should remain synchronous if the conditions specified in the processor AC timing Waveforms section of the Intel® Xeon 5600 Series EMTS (Electrical Mechanical and Thermal Specifications) are met. Due to this erratum the TSC may become out of sync between the processor packages after a warm reset even if the Reset# de-assertion requirements are met.

Implication: Certain software applications that rely on hardware based TSC cross-socket synchronization may not function correctly.

Workaround: It is possible for the BIOS to contain a workaround for this erratum.

Status: For the steppings affected, see the Summary Table of Changes.

BF133. Electromechanical Interlock Control Not Functioning Correctly on Hot Plug SMBus

Problem: The PCIe Hot Plug SMBus does not uniquely pulse the EMIL (Electromechanical Interlock) control virtual pins in response to software writes to the Electromechanical

Interlock Control Bit (Device 3,4,5,6; Function 0; Offset A8H, Bit 11). A write to a single Electromechanical Interlock Control Bit will pulse the EMIL virtual control pins on both 8-bit ports (ports 0 and 1) of the SMBUS I/O extender.

Implication: If a SMBUS I/O extender with two ports is used, both PCIe cards will be ejected whenever software attempts to eject either card.

Workaround: None Identified.

Status: For the steppings affected, see the Summary Table of Changes.

BF134. PCIe Port's LTSSM May Not Transition Properly in the Presence of TS1 or TS2 Ordered Sets That Have Unexpected Symbols Within those Sets

Problem: When a PCIe port receives TS1 and/or TS2 ordered sets with unexpected symbols (per the PCIe Base Specification), the port's LTSSM (Link Training State Machine) might not transition according to the PCIe Base Specification requirements. The LTSSM may incorrectly stay in its current state, or transition to an incorrect state. If the unexpected symbols are sporadic in nature the link will recover and go to the proper state.

Implication: PCIe Port's LTSMM may not transition according to PCIe Base Specification as described above. This problem has not been seen in real system testing, but was discovered by synthetic tests designed to check for illegal conditions.

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes.

BF135. NTB Secondary LNKCAP Register Does not Reflect Programmed Link Capabilities

Problem: The NTB (Non Transparent Bridge) secondary LNKCAP (Link Capabilities) register (Offset 59CH) in the region defined by the Base Address Register SB01BASE (Bus M; Device 0; Function 0; Offset 10H) does not reflect the link capabilities programmed by BIOS. The register always reflects the reset default states for L0s/L1 exit latencies, power management support and maximum link width.

Implication: Software can not rely on the following information reported by the secondary LNKCAP register:

- L0s and L1 exit latencies: May report incorrect values.
- Link Speed: Link will always show support for 5GT/s operation but may only train to 2.5GT/s if so limited by BIOS or pin strap
- Link width: Link will always show x8 regardless of maximum link width imposed by BIOS or strap
- Active State Link Power Management Support: Link will show support for L0s and L1 even if those states have been disabled by BIOS

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes.

BF136. Using Intel® VT-d with NTB in Certain Platform Configurations Can Alias Requester ID's and Cause Unexpected Behavior

Problem: Due to possible aliasing of requestor-IDs, the NTB (non-transparent bridge) is incompatible with Intel VT-d (Virtualization Technology for Directed I/O) in certain platform configurations. The following platform configurations may exhibit unexpected system behavior if Intel VT-d is used to virtualize NTB traffic:

1. Any platform with the NTB configured in NTB/RP (NTB to Root Port) Mode

2. Dual socket platforms with the NTB configured in NTB/NTB (NTB to NTB) mode and the NTB is used on both sockets.

Implication: The Intel VT-d engine can not be used to virtualize the NTB port for the system configurations given above.

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes.

BF137. PCIe Squelch Detect May be Slow to Respond During L0s Entry and May Cause a Surprise Link Down Condition

Problem: PCIe Squelch Detect May be Slow to Respond During L0s Entry and May Cause a Surprise Link Down Condition

Implication: This erratum may cause a system hang while trying reach the L0s state.

Workaround: A BIOS code change has been identified and may be implemented as a workaround for this erratum.

Status: For the steppings affected, see the Summary Table of Changes.

BF138. Ports May Not Enter Slave Loopback Mode From the Configuration LTSSM State

Problem: If a PCIe port's LTSSM (Link Training State Machine) is in the CONFIG.LINK_WIDTH_START state, it may not enter slave loopback mode when requested to do so by the link partner. If the request is missed the link will continue to train and enter the Slave loopback mode after it first transitions through the L0 and RECOVERY LTSSM states.

Implication: PCIe ports may be delayed in entering the slave loopback mode.

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes.

BF139. Multiple UC Errors Reported Via MSI May Result in Lost Interrupts

Problem: Using MSI (Message Signaled Interrupt) to forward PCIe UC (Uncorrectable) error interrupts, which are programmed to generate interrupts for both fatal and non-fatal errors, may result in the loss of an unrelated interrupt when both fatal and non-fatal error interrupts occur in close proximity to an unrelated interrupt.

Implication: Interrupts may be lost if MSI is used to report PCIe UC errors as a mix of fatal and non-fatal errors.

Workaround: A BIOS code change has been identified and may be implemented as a workaround for this erratum.

Status: For the steppings affected, see the Summary Table of Changes.

BF140. Performance Monitor Events for Hardware Prefetches Which Miss The L1 Data Cache May be Over Counted

Problem: Hardware prefetches that miss the L1 data cache but cannot be processed immediately due to resource conflicts will count and then retry. This may lead to incorrectly incrementing the L1D_PREFETCH.MISS (event 4EH, umask 02H) event multiple times for a single miss.

Implication: The count reported by the L1D_PREFETCH.MISS event may be higher than expected.

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes.

BF141. DP System Using Package C3 or C6 Power States May Record Spurious Poisoned Packet Errors

Problem: In a DP (dual-processor) system using package C3 or C6 power states, spurious "Protocol Layer received Poisoned Packet" errors may be logged in QPIP0ERRST (Device 8; Function 2; Offset 230H; Bit[2]) as the system transitions into a lower power state.

Implication: Spurious errors may be logged.

Workaround: A BIOS code change has been identified and may be implemented as a workaround for this erratum. In addition to suppressing the spurious error logging, the work-around has a side affect of disabling logging of poisoned packets in QPIP0ERRST bit 2 which were caused by processor writes. Those errors will be logged in the processor's Machine Check Architecture mechanism.

Status: For the steppings affected, see the Summary Table of Changes.

BF142. Revision ID For Non-Legacy Processor is Incorrect

Problem: In a DP (dual-processor) system, the revision ID for the non-legacy processor (processor not directly connected to the Intel® 3420 Chipset) as reported in RID (Bus IIOBUSNO, Device 0; Function 0; Offset 08H) is incorrect. RID should report 10H instead of 00H.

Implication: Incorrect revision ID for non-legacy processor.

Workaround: A BIOS code change has been identified and may be implemented as a workaround for this erratum.

Status: For the steppings affected, see the Summary Table of Changes.

BF143. NTB Operating In NTB/RP Mode With MSI/MSI-X Interrupts May Cause System Hang

Problem: The NTB (Non-transparent Bridge) operating in NTB/RP (NTB to Root Port mode) using Message Signaled Interrupts (MSI or MSI-X) in the presence of locks may result in a system hang.

Implication: The system may hang under the condition described above.

Workaround: A BIOS code change has been identified and may be implemented as a workaround for this erratum.

Status: For the steppings affected, see the Summary Table of Changes.

BF144. NTB May Incorrectly Set MSI or MSI-X Interrupt Pending Bits

Problem: The NTB (Non-transparent Bridge) may incorrectly set MSI (Message Signaled Interrupt) pending bits in MSIPENDING (BAR PB01BASE,SB01BASE; Offset 74H) while operating in MSI-X mode or set MSI-X pending bits in PMSIXPBA (BAR PB01BASE, SB01BASE; Offset 03000H) while operating in MSI mode.

Implication: NTB incorrectly sets MSI or MSI-X pending bits. The correct pending bits are also set and it is safe to ignore the incorrectly set bits.

Workaround: None Identified.

Status: For the steppings affected, see the Summary Table of Changes.

BF145. NTB/RP Completer and Requester ID May be Unreliable

Problem: When configured in NTB/RP (Non-transparent bridge Root Port Mode), the Completer ID of the NTB may contain random data rather than the BDF (Bus:Device:Function) assigned during configuration. The Requester ID initiated from the NTB will always be

Bus 00H, Device 03H, Function 00H regardless of the BDF assigned during configuration.

Implication: Any device which relies on the Completer ID (in a completion coming from the NTB) or Requester ID (from a request originating from the NTB) to be the value assigned during configuration will not be compatible with the NTB in NTB/RP mode. In addition, the processor will log the erroneous Requester/Completer ID in the HDRLOG (Device 3, Function 0, Offset 11CH) register when the NTB encounters an error while transmitting a packet.

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes.

BF146. NTB Does Not Set PME_TO_ACK After a PME_TURN_OFF Request

Problem: The NTB (Non-transparent Bridge) does not set PME_TO_ACK in MISCCTRLSTS (Device 3; Function 0: Offset 188H; Bit [48]) after a PME_TURN_OFF request.

Implication: The NTB will not acknowledge a PME_TURN_OFF request.

Workaround: ACPI or other software must have a time-out to proceed with the power management event and should not wait indefinitely for the NTB to acknowledge the PME_TURN_OFF request.

Status: For the steppings affected, see the Summary Table of Changes.

BF147. Poisoned Write Caused by an Internal Parity Error Targeting IIO PCI Configuration Registers or MMIO Space will Not be Suppressed

Problem: When due to an internal parity error, a processor attempts to write poisoned data to a PCI configuration register in the IIO (Integrated I/O) module (internal PCI devices on bus IIOBUSNO) or to the MMIO space decoded by a BAR in the IIO module, the poisoned data will not be dropped. However, even though the poisoned data will not be dropped the internal Intel® QuickPath Interconnect logic will log and report an error in the IA32_MC0_STATUS MSR (401H) with MCACOD equal to 0000 1110 xxxx xx11 and bit 16 or 17 set.

Implication: Poisoned data may be written to PCI configuration registers or MMIO space causing a machine check exception. It is possible for these writes to lead to unpredictable system behavior.

Workaround: None identified

Status: For the steppings affected, see the Summary Table of Changes.

BF148. In-flight DMA Requests Received During the Implicit DMA Draining Window When Enabling Intel® VT-d Hardware Will Result in a Spurious DMA Fault

Problem: In-flight DMA requests during the 2 cycle window for DMA draining when enabling Intel® VT-d hardware will result in a spurious DMA fault.

Implication: BIOS features, such as legacy keyboard emulation, can result in in-flight DMA requests (such as from a USB controller) when the OS/VMM is booting and enabling Intel® VT-d. If such DMA requests happen to arrive during the 2 cycle implicit DMA draining window when enabling Intel® VT-d, they result in a DMA fault irrespective of the programming of VT-d translation structures.

Workaround: A BIOS code change has been identified and may be implemented as a workaround for this erratum.

Status: For the steppings affected, see the Summary Table of Changes.

BF149. Writes to B2BSPAD[15:0] Registers May Transfer Corrupt Data Between NTB Connected Systems

Problem: Writes to the NTB (Non-Transparent Bridge) B2BSPAD[15:0] registers (BAR PB01BASE, SB01BASE; Offsets 100H - 13FH) may result in corrupted data transfer between systems.

Implication: Using B2BSPAD[15:0] registers to transfer data may not work as expected.

Workaround: Do not use the B2BSPAD[15:0] to send data from the local to remote host. Instead, configure one of the following local register pairs to point to the remote SB01BASE region:

- PB23BASE (Device 3; Function 0; Offset 18H) and PBAR2XLAT (Offset 10H) from PB01BASE or SB01BASE regions
- PB45BASE (Device 3; Function 0; Offset 20H) and PBAR4XLAT(Offset 18H) from PB01BASE, or SB01BASE regions

The local host may then write directly to the SPAD[15:0] registers (Offsets 80H - 0BFH) of the remote system from the PB23BASE/PB45BASE region defined above.

Status: For the steppings affected, see the Summary Table of Changes.

BF150. Unable to Clear Received PME_TO_ACK in NTB

Problem: When the NTB (Non-transparent Bridge) is enabled, the Received PME_TO_ACK bit in MISCCTRLSTS (Device 3; Function 0; Offset 188H; Bit[48]) can not be cleared if the timeout for receiving PME_TO_ACK is enabled (Device 3; Function 0; Offset 188H, Bit[5] is 0).

Implication: Software may be unable to clear Received PME_TO_ACK.

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes.

BF151. Using Intel® VT-d With IIO Legacy PCI Interrupts Targeting the PCH I/OxAPIC May Result in a System Hang

Problem: When Intel® VT-d is enabled, using legacy PCI interrupts originating from the IIO (Integrated I/O) or PCIe ports targeting the PCH I/OxAPIC may result in a system hang.

Implication: Using legacy PCI interrupts from the IIO or PCIe ports targeting the PCH IOxAPIC with Intel® VT-d enabled may lead to a hang.

Workaround: A BIOS code change has been identified and may be implemented as a workaround for this erratum.

Status: For the steppings affected, see the Summary Table of Changes.

BF152. Using I/O Peer-to-Peer Write Traffic Across an NTB May Lead to a Hang

Problem: If two systems are connected via an NTB (Non-Transparent Bridge), either the internal NTB or an external NTB, and both systems attempt to send I/O peer-to-peer write traffic across the NTB either to memory or an I/O device on the remote system, it is possible for both systems to deadlock

Implication: Using I/O peer-to-peer write traffic across an NTB may lead to a hang.

Workaround: A BIOS code change has been identified and may be implemented as a workaround for this erratum.

Status: For the steppings affected, see the Summary Table of Changes.

BF153. VM Exit May Incorrectly Clear IA32_PERF_GLOBAL_CTRL [34:32]

Problem: If the "load IA32_PERF_GLOBAL_CTRL" VM-exit control is 1, a VM exit should load the IA32_PERF_GLOBAL_CTRL MSR (38FH) from the IA32_PERF_GLOBAL_CTRL field in the guest-state area of the VMCS. Due to this erratum, such a VM exit may instead clear bits 34:32 of the MSR, loading only bits 31:0 from the VMCS.

Implication: All fixed-function performance counters will be disabled after an affected VM exit, even if the VM exit should have enabled them based on the IA32_PERF_GLOBAL_CTRL field in the guest-state area of the VMCS.

Workaround: A VM monitor that wants the fixed-function performance counters to be enabled after a VM exit may do one of two things: (1) clear the "load IA32_PERF_GLOBAL_CTRL" VM-exit control; or (2) include an entry for the IA32_PERF_GLOBAL_CTRL MSR in the VM-exit MSR-load list.

Status: For the steppings affected, see the Summary Table of Changes.

BF154. MCTP PCIe Messages Not Silently Discarded by Non-Legacy Processor

Problem: MCTP (Management Control Transport Protocol) messages on PCIe are not supported and should be silently discarded. However, MCTP messages with a vendor ID of 1AB4H received by the non-legacy processor (processor not directly attached to the Intel® 3420 chipset) in a DP (dual-processor) system will be forwarded to PCIe port 0 (device 3, function 0) of the non-legacy processor.

Implication: Unsupported MCTP messages with vendor ID 1AB4H will be forwarded to PCIe port 0. Further implications are dependent on how the end point processes these messages.

Workaround: None Identified.

Status: For the steppings affected, see the Summary Table of Changes.

BF155. NTB Secondary Link Disable Control Not Functional

Problem: The Secondary Link Disable Control (BAR PB01BASE; Offset 58H; bit 1) does not fully disable the link when set. The link will partially train even with the bit set, possibly leaving the LTSSM (Link Training and Status State Machine) in an invalid state.

Implication: In NTB (Non-transparent Bridge) mode, the link might not train correctly. Drivers can not use the Secondary Link Disable Control bit to control NTB link state. BIOS must clear this bit before initializing link training.

Workaround: None identified. However to initialize NTB mode BIOS must clear the Secondary Link Disable Control bit.

Status: For the steppings affected, see the Summary Table of Changes.

BF156. Unexpected DMI/ESI and PCIe Link Retraining and Correctable Errors Reported

Problem: When the processor exits the package C6 power state, the PCIe and DMI/ESI ports may enter a state where they will NAK all packets for a short time. If this condition persists long enough so that the same packet is NAKed four times, the link will retrain and a correctable error may be signaled by the PCIe end point. Overall performance of the link is not impacted.

Implication: Due to this erratum, unexpected link retraining and correctable errors may be reported.

Workaround:

Workaround: A BIOS code change has been identified and may be implemented as a workaround for this erratum.

Status: For the steppings affected, see the Summary Table of Changes.

BF157. QPI Lane May Be Dropped During Full Frequency Deskew Phase of Training

Problem: A random QPI Lane may be dropped during the lane deskew phase while the QPI Bus is training at full frequency.

Implication: When there are multiple resets after the QPI Bus has reached full speed operation there is a small chance that a lane could be dropped during the deskew phase of training. In the case of a lane being dropped this will be detected and a retry will be done until the link is established and the lane is re-trained.

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes.

BF158. VM Entries that Return from SMM May Incorrectly Write to the SMRR Protected Region

Problem: If the executive-VMCS pointer field in the VMCS does not contain the VMXON pointer and the "use TPR shadow" VM-execution control is 1 in the executive VMCS, a VM entry that returns from SMM may write to the virtual-APIC page. Due to this erratum, this write may occur even if the virtual-APIC page is in the region protected by the SMRR (system-management range register).

Implication: The writes to the virtual-APIC page may corrupt data in SMRAM.

Workaround: If software sets the "use TPR shadow" VM-execution control to 1, it should not VMWRITE the virtual-APIC address to an address in the range protected by the SMRR.

Status: For the steppings affected, see the Summary Table of Changes.

BF159. PerfMon Overflow Status May Remain Always Set After Certain Conditions Have Occurred

Problem: Under very specific timing conditions, if software tries to disable a PerfMon counter through MSR IA32_PERF_GLOBAL_CTRL (0x38F) or through the per-counter event-select (e.g. MSR 0x186) and the counter reached its overflow state very close to that time, then the overflow status indication in MSR IA32_PERF_GLOBAL_STAT (0x38E) may be left set with no way for software to clear it.

Implication: Software may be unable to clear the PerfMon counter overflow status indication.

Workaround: Software may avoid this erratum by clearing the PerfMon counter value prior to disabling it and then clearing the overflow status indication bit.

Status: For the steppings affected, see the Summary Table of Changes.

BF160. An Unexpected Page Fault or EPT Violation May Occur After Another Logical Processor Creates a Valid Translation for a Page

Problem: An unexpected page fault (#PF) or EPT violation may occur for a page under the following conditions:

- The paging structures initially specify no valid translation for the page.
- Software on one logical processor modifies the paging structures so that there is a valid translation for the page (e.g., by setting to 1 the present bit in one of the paging-structure entries used to translate the page).
- Software on another logical processor observes this modification (e.g., by accessing a linear address on the page or by reading the modified paging-structure entry and seeing value 1 for the present bit).
- Shortly thereafter, software on that other logical processor performs a store to a linear address on the page.

In this case, the store may cause a page fault or EPT violation that indicates that there is no translation for the page (e.g., with bit 0 clear in the page-fault error code, indicating that the fault was caused by a not-present page). Intel has not observed this erratum with any commercially available software.

Implication: An unexpected page fault may be reported. There are no other side effects due to this erratum.

Workaround: System software can be constructed to tolerate these unexpected page faults. See Section "Propagation of Paging-Structure Changes to Multiple Processors" of Volume 3A of *IA-32 Intel® Architecture Software Developer's Manual*, for recommendations for software treatment of asynchronous paging-structure updates.

Status: For the steppings affected, see the Summary Table of Changes.

BF161. L1 Data Cache Errors May be Logged With Level Set to 1 Instead of 0

Problem: When an L1 Data Cache error is logged in IA32_MCi_STATUS[15:0], which is the MCA Error Code Field, with a cache error type of the format 0000 0001 RRRR TTLL, the LL field may be incorrectly encoded as 01 instead of 00.

Implication: An error in the L1 Data Cache may report the same LL value as the L2 Cache. Software should not assume that an LL value of 01 is the L2 Cache.

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes.

BF162. Stack Pushes May Not Occur Properly for Events Delivered Immediately After VM Entry to 16-Bit Software

Problem: The stack pushes for an event delivered after VM entry and before execution of an instruction in VMX non-root operation may not occur properly. The erratum applies only if the VM entry establishes IA32_EFER.LMA = 0 and CS.D = 0 and only if the event handler is also invoked with CS.D = 0.

Implication: This erratum affects events that are pending upon completion of VM entry and that do not cause VM exits. Examples include debug exceptions, interrupts, and general-protection faults generated in virtual-8086 mode by the mode's virtual interrupt mechanism. The erratum applies only if the VM entry is not to IA-32e mode and is to 16-bit operation, and only if the relevant handler uses 16-bit operation. The incorrect stack pushes resulting from the erratum may cause incorrect guest operation. Intel has not observed this erratum with any commercially available software.

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes.

BF163. A Logical Processor May Wake From Shutdown Mode When Branch-Trace Messages Are Enabled

Problem: Normally, a logical processor enters shutdown mode and remains in this mode until a break event (NMI, SMI, INIT) occurs. Due to this erratum, if CR4.MCE (Machine Check Enable) is 0 and a branch trace message is pending at the time of a machine check, the processor may not remain in shutdown mode. In addition, if the processor is in VMX non-root operation, a subsequent VM exit may save value 2 into the activity-state field in the VMCS (indicating shutdown) even though the VM exit did not occur while in shutdown state.

Implication: This erratum may result in unexpected system behavior. If a VM exit saved a value of 2 into the activity-state field in the VMCS, the next VM entry will take the processor to shutdown state.

Workaround: Software should ensure that CR4.MCE is set whenever IA32_DEBUGCTL MSR (60EH) TR bit [6] is set.

Status: For the steppings affected, see the Summary Table of Changes.

BF164. PerfMon Event LOAD_HIT_PRE.SW_PREFETCH May Overcount

Problem: PerfMon event LOAD_HIT_PRE.SW_PREFETCH (event 4CH, umask 01H) should count load instructions hitting an ongoing software cache fill request initiated by a preceding software prefetch instruction. Due to this erratum, this event may also count when there is a preceding ongoing cache fill request initiated by a locking instruction.

Implication: PerfMon event LOAD_HIT_PRE.SW_PREFETCH may overcount.

Workaround: None identified

Status: For the steppings affected, see the Summary Table of Changes.

BF165. Successive Fixed Counter Overflows May be Discarded

Problem: Under specific internal conditions, when using Freeze PerfMon on PMI feature (bit 12 in IA32_DEBUGCTL.Freeze_PerfMon_on_PMI, MSR 1D9H), if two or more PerfMon Fixed Counters overflow very closely to each other, the overflow may be mishandled for some of them. This means that the counter's overflow status bit (in MSR_PERF_GLOBAL_STATUS, MSR 38EH) may not be updated properly; additionally, PMI interrupt may be missed if software programs a counter in Sampling-Mode (PMI bit is set on counter configuration).

Implication: Successive Fixed Counter overflows may be discarded when Freeze PerfMon on PMI is used.

Workaround: Software can avoid this by:

1. Avoid using Freeze PerfMon on PMI bit
2. Enable only one fixed counter at a time when using Freeze PerfMon on PMI

Status: For the steppings affected, see the Summary Table of Changes.

BF166. #GP May be Signaled When Invalid VEX Prefix Precedes Conditional Branch Instructions

Problem: When a 2-byte opcode of a conditional branch (opcodes 0F8xH, for any value of x) instruction resides in 16-bit code-segment and is associated with invalid VEX prefix, it may sometimes signal a #GP fault (illegal instruction length > 15-bytes) instead of a #UD (illegal opcode) fault.

Implication: Due to this erratum, #GP fault instead of a #UD may be signaled on an illegal instruction.

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes.

BF167. Task Switch to a TSS With an Inaccessible LDTR Descriptor May Cause Unexpected Faults

Problem: A task switch may load the LDTR (Local Descriptor Table Register) with an incorrect segment descriptor if the LDT (Local Descriptor Table) segment selector in the new TSS specifies an inaccessible location in the GDT (Global Descriptor Table).

Implication: Future accesses to the LDT may result in unpredictable system behavior.

Workaround: Operating system code should ensure that segment selectors used during task switches to the GDT specify offsets within the limit of the GDT and that the GDT is fully paged into memory.

Status: For the steppings affected, see the Summary Table of Changes.

BF168. VM Entries That Return From SMM Using VMLAUNCH May Not Update The Launch State of the VMCS

Problem: Successful VM entries using the VMLAUNCH instruction should set the launch state of the VMCS to "launched". Due to this erratum, such a VM entry may not update the launch state of the current VMCS if the VM entry is returning from SMM.

Implication: Subsequent VM entries using the VMRESUME instruction with this VMCS will fail. RFLAGS.ZF is set to 1 and the value 5 (indicating VMRESUME with non-launched VMCS) is stored in the VM-instruction error field. This erratum applies only if dual monitor treatment of SMI and SMM is active.

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes.

BF169. VM Entry May Clear Bytes 81H-83H on Virtual-APIC Page When "Use TPR Shadow" Is 0

Problem: VM entry should not clear bytes 81H-83H on the virtual-APIC page if the "use TPR shadow" VM-execution control is 0. Due to this erratum, VM entry will do so if the "virtualize x2APIC mode" VM-execution control is 1.

Implication: VM entries with the 0-setting of the "use TPR shadow" VM-execution control and the 1-setting of the "virtualize x2APIC mode" VM-execution control cause any non-zero data at bytes 81H-83H on the virtual-APIC page to be lost. Note that this combination of settings is not allowed; any such VM entry will fail after clearing these bytes.

Workaround: Software should always set the "use TPR shadow" VM-execution control to 1 whenever it sets that "virtualize x2APIC mode" VM-execution control to 1.

Status: For the steppings affected, see the Summary Table of Changes.

BF170. A First Level Data Cache Parity Error May Result in Unexpected Behavior

Problem: When a load occurs to a first level data cache line resulting in a parity error in close proximity to other software accesses to the same cache line and other locked accesses the processor may exhibit unexpected behavior.

Implication: Unpredictable system behavior may occur. Intel has not observed this erratum with any commercially available system.

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes.

BF171. Dual Processor Systems Running with DCA Enabled May Cause the System to Hang

Problem: A dual processor system running with DCA (Direct Cache Access) enabled may cause the system to hang. The processor will signal an internal error via the CATERR# pin and an Internal Timer Error will be logged in IA32_MCi_STATUS.MCACOD (bits[15:0]) containing a value of 0000_0100_0000_0000.

Implication: Due to this erratum, the system may hang.

Workaround: None identified. DCA must be disabled by BIOS in dual processor systems.

Status: For the steppings affected, see the Summary Table of Changes.

Intel® Xeon® Processor C5500/C3550 Series Storage-Specific Errata

Note: References to DMA (direct memory access) "XOR Operations" include the following operation types as defined by the DMA Descriptor Control Field [31:24] Operation Type:

- 89H - XOR with Galios Field (GF) Multiply Generation
- 8AH - XOR with Galios Field (GF) Multiply Validate
- 8BH - XOR with Galios Field (GF) Multiply Update Generation

All other supported DMA operations are referred to as "legacy".

BF500S. 2858762:Using DMA XOR With DCA May Cause a Machine Check

Problem: If both DCA (direct cache access) and DMA XOR operations are active at the same time, then invalid prefetch hints may be generated. These prefetch transactions may not complete and could result in a timeout machine check, which will cause CATERR# to become asserted.

Implication: Invalid prefetch hints may not complete resulting in a machine check.

Workaround: If using DMA XOR operations, disable DCA by clearing CHANCTRL.Completion_Write_DCA_Enable (Offset 80H; Bit 9) in the region described by CB_BAR (Device: 10; Function 0-7; 0ffset 10H) on the processor's internal IO bus (as defined in the IIOBUSNO register).

Status: For the steppings affected, see the Storage-Specific Errata Summary Table of Changes.

BF501S. 2858702:Mixed DMA XOR and Legacy Operations in The Same Channel May Cause Data to be Observed Out of Order

Problem: For mixed channel DMA (XOR and legacy operations active on the same channel) completion writes from legacy operations may pass completion writes from XOR operations resulting in out of order descriptor updates/completions.

Implication: DMA descriptor progress may appear out of order with incorrect data.

Workaround: In the DMA driver each DMA XOR descriptor must be followed by an additional legacy descriptor. The legacy descriptor must have a non-zero transfer length and the "NULL Transfer" bit and "Completion Interrupt" in the Descriptor Control field set to '1'. The transfer will not actually occur, but a completion interrupt will be generated that indicates that the XOR operation has completed. This causes all completion interrupts to be of the legacy type.

Status: For the steppings affected, see the Storage-Specific Errata Summary Table of Changes.

BF502S. 2858773:Unexpected DMA XOR Halt and Errors When Using Descriptors With P or Q Operations Disabled

Problem: If a Galois Field Generate/Validate base descriptor has either the P Operations Disable or Q Operation Disable bit set and the corresponding disabled P Parity Address or Q Parity Address field of the descriptor does not contain a valid/aligned address, the DMA

channel may halt unexpectedly with destination address errors. The destination address errors will be logged in CHANERR.DMA Transfer Destination Address Error (Offset A8H; Bit 1) and the CHANERR_INT.DMA Transfer Destination Address Error (Device 10; Function 0-1; Offset 180H; Bit 1). CHANERR is in the region described by CB_BAR (Device 10; Function 0-1; 0ffset 10H) on the processor's internal IO bus (as defined in the IIOBUSNO register).

Implication: The DMA may only partially process a DMA XOR descriptor when a disabled P or Q Parity Address field of the descriptor does not contain a valid/aligned address, resulting in incomplete data, an unexpected DMA channel halt and destination address errors.

Workaround: At all times, software must place a valid/aligned address in both the P Parity Address field and the Q Parity Address field of a DMA XOR with Galois Field Generate/Validate base descriptor even if the P Operations Disable or Q Operations Disable descriptor fields are set to disable either P or Q operations for the descriptor.

Status: For the steppings affected, see the Storage-Specific Errata Summary Table of Changes.

BF503S. 2858563:DMA XOR Channel May Hang on Source Read Completion Data Parity Error For >8K Descriptors

Problem: If a parity error occurs of source read completion data while inside the DMA for >8K descriptor transfer lengths, the DMA channel will hang until the next platform reset. This behavior only applies if the data arrived at the DMA unit error free (from DRAM and QPI) but then had a parity error in the completion data FIFO inside the DMA.

Implication: The effected DMA channel will hang until the next platform reset.

Workaround: None identified.

Status: For the steppings affected, see the Storage-Specific Errata Summary Table of Changes.

BF504S. 2858561:DMA CB_BAR Decode May be Incorrect After DMA FLR

Problem: A PCIe FLR (function level reset) of the DMA function, may result in an incorrect CB_BAR (Device 10; Function 0-7; Offset 10h) decode when a memory read of the CB_BAR occurs around the same time as the FLR.

Implication: A FLR may cause a PCIe memory read to decode to channel 0 instead of the intended channel resulting in incorrect read data returned.

Workaround: Software must quiesce the DMA function before issuing FLR including:

- Ensure clients are no longer referencing the driver.
- Ensure all outstanding descriptors have completed via the normal completion writeback notifications by reading CHANCMP, CHANSTS, and DMACOUNT.
- Issue FLR and ensure no new DMA transactions are started until FLR has completed.

CHANCMP (Offset 98H) and CHANSTS (Offset 88H), and DMACOUNT (Offset 86H) are offsets relative to CB_BAR on the processor's internal IO bus (as defined in the IIOBUSNO register).

For the steppings affected, see the Storage-Specific Errata Summary Table of Changes.

BF505S. XOR DMA Restricted to <=8KB Transfers When Multiple Channels Are in Use

Problem: Incorrect data transfers can occur if more than one DMA channel is in operation and >8KB XOR DMA transfer sizes are being used. XOR DMA transfer size is set by software in the Block Size field of the XOR with Galios Field Generate/Validate base descriptor.

Implication: XOR DMA operation is restricted to <=8KB transfer sizes when multiple DMA channels are in use. Legacy DMA operations may still use up to the maximum 1MB transfer length.

Workaround: Software may either:

- Use a single DMA channel for both legacy and XOR operation types both up to the maximum 1MB transfer size.
- Use multiple DMA channels where XOR operation types are <=8KB transfer size and legacy operation types are up to 1MB transfer size.

Status: For the steppings affected, see the Storage-Specific Errata Summary Table of Changes.

BF506S. Unable to Restart DMA After Poisoned Error During an XOR Operation

Problem: If the CHANERR field Read Data Error (Offset A8H; Bit 8) is set due to a poisoned completion error during a DMA XOR operation, the DMA stays in the halted state and the Read Data Error bit does not clear.

Implication: The XOR operations on the DMA can not be restarted after a read data error due to a poisoned XOR operation.

Workaround: At least one XOR descriptor with no read data errors has to be processed for a new chain of XOR descriptors to work correctly with the corresponding CHANERRMSK (Offset ACH; Bit 8) bit set. Upon detection of a Read Data Error, software must clear the CHANERR and CHANERR_INT (Device 10; Function 0-7; Offset 180H) registers and disable the corresponding error mask bit by setting CHANERRMSK. Then new descriptors can be added to the chain and the DMA started by writing the DMACOUNT (Offset 86H). Once the DMA channel is in the running state, software can clear the CHANERRMSK. CHANERR, CHANERRMSK, and DMACOUNT are offsets relative to CB_BAR (Device 10; Function 0-7; 0ffset 10H) on the processors internal IO bus (as defined in the IIOBUSNO register).

Status: For the steppings affected, see the Storage-Specific Errata Summary Table of Changes.

BF507S. DMA Restart Hang When First Descriptor is a Legacy Type Following Channel HALT Due to an Extended Descriptor Error

Problem: When using multiple DMA channels, all DMA channels may hang if a DMA channel restart is attempted with a Legacy descriptor as the first descriptor following an error/HALT on an Extended descriptor on Channel 0 or 1.

Implication: Following an extended descriptor error on Channel 0 or 1, the channel must be not be restarted with a first descriptor of legacy type including NULL. Does not apply for single channel operation.

Workaround: Software must guarantee that the first descriptor processed on restart is an XOR GF Multiply Generation (base type) before using legacy descriptors with interrupts and completions.

Status: For the steppings affected, see the Storage-Specific Errata Summary Table of Changes.

BF508S. Operation With DMA XOR Interrupts/Completions Enabled Restricted to Channel 0 and 1

Problem: If DMA XOR interrupts and completions are enabled on channel 0 or 1 concurrent with operation on channels 2-7, incorrect data transfers can occur on DMA channels 2-7. DMA XOR interrupts and completions are enabled by setting bits 0 and 3 of descriptor control field of a DMA XOR with Galios Field Generate/Validate base descriptor.

Implication: If DMA XOR interrupts and completions are enabled, only one interrupt/completion type may be used on any single channel and only channels 0 and 1 may be used.

Workaround: Software must either:

- Only use only legacy interrupts and completions on all channels.
- Use only DMA channels 0 and 1 where:
  - — Only DMA XOR interrupts/completions are enabled on channel 0 and is only used for DMA XOR operations.
  - — Only legacy interrupts/completions are enabled on channel 1 and is only used for DMA legacy operations.

Status: For the steppings affected, see the Storage-Specific Errata Summary Table of Changes.

BF509S. Suspending/Resetting an Active DMA XOR Channel May Cause an Incorrect Data Transfer on Other Active Channels

Problem: Suspending an active DMA XOR channel by setting CHANCMD.Suspend DMA bit (Offset 84; Bit 2) while XOR type DMA channels are active may cause incorrect data transfer on the other active legacy channels. This erratum may also occur while resetting an active DMA XOR channel CHANCMD.Reset DMA bit (Offset 84; Bit 5). CHANCMD is in the region described by CB_BAR(Device; 10; function 0-7; Offset 10H) on the processor's internal IO bus (as defined in the IIOBUSNO register).

Implication: An incorrect data transfer may occur on the active legacy DMA channels.

Workaround: Software must suspend all legacy DMA channels before suspending an active DMA XOR channel (channel 0 or 1.

Status: For the steppings affected, see the Storage-Specific Errata Summary Table of Changes.

BF510S. DWORD-Aligned DMA XOR Descriptors With Fencing And Multi-Channel Operation May Cause a Channel Hang

Problem: DMA XOR descriptors with DWORD aligned sources and fencing enabled may result in a XOR channel hang until the next platform reset. XOR DMA fencing is set by software in Descriptor Control.Fence (XOR base descriptor, bit 4).

Implication: An XOR DMA descriptor with non cacheline aligned sources may hang until the next platform reset.

Workaround: Do not enable fencing on XOR descriptors. Fencing can be enabled on legacy descriptors. It is recommended that a NULL legacy descriptor must be paired with each XOR descriptor. Software can use fencing of the legacy NULL descriptor to track full completion of its associated XOR descriptor.

Status: For the steppings affected, see the Storage-Specific Errata Summary Table of Changes.

BF511S. DWORD Aligned XOR DMA Sources May Prevent Further DMA XOR Progress

Problem: XOR DMA channels may stop further progress in the presence of Locks/PHOLDs if the source pointed to by a DMA XOR descriptor is not cacheline aligned.

Implication: Non-cacheline aligned DMA XOR sources may hang both channels 0 and 1. A reset is required in order to recover from the hang. Legacy DMA descriptors on any channel have no source alignment restrictions.

Workaround: Software must either:

- Ensure XOR DMA descriptors only point to cache-line aligned sources (best performance) OR
- A legacy DMA copy must be used prior to non-cacheline aligned DMA operations to guarantee that the source mis-alignment is on DWORD15 of the cacheline. The

required source that must be misaligned to DWORD15, depends on the following desired subsequent DMA XOR operations:

- DMA XOR Validate (RAID5/ P-Only): The P-source must be mis-aligned to DWORD15 (last DWORD).
- DMA XOR Validate (RAID6/P+Q): The Q-source must be mis-aligned to DWORD15 (last DWORD).
- DMA XOR Generate or Update: The last source (which will be different based on numblk) must be misaligned to DWORD15 (last DWORD).

Status: For the steppings affected, see the Storage-Specific Errata Summary Table of Changes.

Specification Changes

The specification changes listed in this section may apply to the following documents:

- Intel® Xeon® Processor C5500/C3500 Series Datasheet, Volume 1
- Intel® Xeon® Processor C5500/C3500 Series Datasheet, Volume 2

1. Changes for SMB_CLK, SMB_DATA, PE_HP_DATA, PE_HP_CLK

Old Text:

Table 171. SMBus Clock DC Electrical Limits

Symbol	Parameter	Min	Typ	Max	Units	Notes[1]
V_{IL}	Input Low Voltage			$0.64 * V_{TTA}$	V	2,3
V_{IH}	Input High Voltage	$0.76 * V_{TTA}$				2
V_{OL}	Output Low Voltage			$V_{TTA} * R_{ON} / (R_{ON} + R_{SYS\_TERM})$	V	2,4
V_{OH}	Output High Voltage	V_{TTA}			V	2
R_{ON}	IO Buffer On Resistance	10		18	Ohms	
I_{LI}	Input Leakage Current			± 200	µA	

Notes:
1. Unless otherwise noted, all specifications in this table apply to all processor frequencies.
2. The V_{TTA} referred to in these specifications refers to instantaneous V_{TTA}.
3. Based on a test load of 50 Ohms to V_{TTA}.
4. R_{SYS_TERM} is the termination on the system and is not controlled by the C5500/C3500 Series.

New Text: Changed text is shown in red.

Table 171. SMBus Clock DC Electrical Limits

Symbol	Parameter	Min	Typ	Max	Units	Notes[1]
V_{IL}	Input Low Voltage			$0.4 * V_{TTA}$	V	2,3
V_{IH}	Input High Voltage	$0.76 * V_{TTA}$				2
V_{OL}	Output Low Voltage			$V_{TTA} * R_{ON} / (R_{ON} + R_{SYS\_TERM})$	V	2,4
V_{OH}	Output High Voltage	V_{TTA}			V	2
R_{ON}	IO Buffer On Resistance	8	16	25	Ohms	
I_{LI}	Input Leakage Current			± 200	µA	

Notes:
1. Unless otherwise noted, all specifications in this table apply to all processor frequencies.
2. The V_{TTA} referred to in these specifications refers to instantaneous V_{TTA}.
3. Based on a test load of 50 Ohms to V_{TTA}.
4. R_{SYS_TERM} is the termination on the system and is not controlled by the C5500/C3500 Series.

Affected Docs:Intel® Xeon® Processor C5500/C3500 Series Datasheet, Volume 1

2. Specification Changing on DP_SYNCRST#, PM_SYNC, DDR_ADR, PE_GEN2_DISABLE#, PE_CFG[2:0], PE_NTBXL, DMI_PE_CFG#, and EXTSYSTRG

Old Text:

Table 174. Reset and Miscellaneous Signal Group DC Specifications

Symbol	Parameter	Min	Typ	Max	Units	Notes[1]
V_{IL}	Input Low Voltage			$0.64 * V_{TTA}$	V	2,3
V_{IH}	Input High Voltage	$0.76 * V_{TTA}$			V	2
V_{OL}	Output Low Voltage			$V_{TTA} * R_{ON} / (R_{ON} + R_{SYS\_TERM})$	V	2,4
V_{OH}	Output High Voltage	V_{TTA}			V	2
ODT	On-Die Termination	45		55		5
R_{ON}	Buffer On Resistance	10		18	Ohms	

Notes:
1. Unless otherwise noted, all specifications in this table apply to all processor frequencies.
2. The V_{TTA} referred to in these specifications refers to instantaneous V_{TTA}.
3. Based on a test load of 50 Ohms to V_{TTA}.
4. R_{SYS_TERM} is the termination on the system and is not controlled by the C5500/C3500 Series.
5. Applies to all signals, unless otherwise mentioned in Table 160.

New Text: Changed text is shown in red.

Table 174. Reset and Miscellaneous Signal Group DC Specifications

Symbol	Parameter	Min	Typ	Max	Units	Notes[1]
V_{IL}	Input Low Voltage			$0.64 * V_{TTA}$	V	2
V_{IL}	Input Low Voltage			$0.40 * V_{TTA}$	V	2,4
V_{IH}	Input High Voltage	$0.76 * V_{TTA}$			V	2
V_{OL}	Output Low Voltage			$V_{TTA} * R_{ON} / (R_{ON} + R_{SYS\_TERM})$	V	2,3
V_{OH}	Output High Voltage	V_{TTA}			V	2
R_{ON}	Buffer On Resistance	10		18	Ohms	

Notes:
1. Unless otherwise noted, all specifications in this table apply to all processor frequencies.
2. The V_{TTA} referred to in these specifications refers to instantaneous V_{TTA}.
3. R_{SYS_TERM} is the termination on the system and is not controlled by the C5500/C3500 Series.
4. Applies to DP_SYNCRST#, PM_SYNC, DDR_ADR, PE_GEN2_DISABLE#, PE_CFG[2:0], PE_NTBXL, DMI_PE_CFG#, EXTSYSTRG.

Affected Docs:Intel® Xeon® Processor C5500/C3500 Series Datasheet, Volume 1

Specification Clarifications

The specification clarifications listed in this section may apply to the following documents:

- Intel® Xeon® Processor C5500/C3500 Series Datasheet, Volume 1
- Intel® Xeon® Processor C5500/C3500 Series Datasheet, Volume 2
-

1. Table 161, Signals with On-Die Termination (ODT) is Being Clarified

Old Text: Table 161 is being clarifed by moving the notes into the table.

Table 161. Signals With On-Die Termination (ODT)

Intel® QuickPath Interface Signal Group[1]
QPI_RX_DP[19:0], QPI_RX_DN[19:0], QPI_TX_DP[19:0], QPI_TX_DN[19:0], QPI_CLKRX_DN, QPI_CLKRX_DP, QPI_CLKTX_DN, QPI_CLKTX_DP
PCI Express Signals
PE_RX_DN[15:0], PE_RX_DP[15:0], PE_TX_DN[15:0], PE_TX_DP[15:0]
DDR3 Signal Group[2]
DDRA_DQ[63:0], DDRB_DQ[63:0], DDDRC_DQ[63:0], DDRA_DQS_N[17:0], DDRA_DQS_P[17:0], DDRB_DQS_N[17:0], DDRB_DQS_P[17:0], DDRC_DQS_N[17:0], DDRC_DQS_P[17:0], DDRA_ECC[7:0], DDRB_ECC[7:0], DDRC_ECC[7:0], DDRA_PAR_ERR#[2:0], DDRB_PAR_ERR#[2:0], DDRC_PAR_ERR#[2:0][3]
Reset and Miscellanous Signal Group and Thermal Signal Group[1]
BPM#[7:0][6], PECI_ID#[7], PREQ#[6], DP_SYNCRST#[9], EXTSYSTRG[9], PMSYNC[9], DDR_ADR[9]
Test Access Port (TAP) Signal Group
TCK[4], TDI[5], TMS[5], TRST#[5], TDI_M[9]
Power/Other Signal Group[8]
VCCPWRGOOD, VTTPWRGOOD, DDR_DRAMPWROK

Notes:
1. Unless otherwise specified, signals have ODT in the package with a 50 Ohm pull-down to V_{SS}.
2. Unless otherwise specified, all DDR3 signals are terminated to $V_{DDQ}/2$.
3. DDRA_PAR_ERR#[2:0], DDRB_PAR_ERR#[2:0], and DDRC_PAR_ERR#[2:0] are terminated to V_{DDQ}.
4. TCK does not include ODT, this signal is weakly pulled-down via a 1-5 kOhm resistor to V_{SS}.
5. TDI, TMS, TRST# do not include ODT, these signals are weakly pulled-up via ~ 10 kOhm resistor to V_{TT}.
6. BPM[7:0]# and PREQ# signals have ODT in package with 35 Ohm pull-ups to V_{TT}.
7. PECI_ID# has ODT in package with a 1-5 kOhm pull-up to V_{TT}.
8. VCCPWRGOOD, VTTPWRGOOD, and DDR_DRAMPWROK have ODT in package with a 5-20 kOhm pull-down to V_{SS}.
9. DP_SYNCRST, EXTSYSTRG, PMSYNC, and TDI_M have a 50 ohm ODT to Vtt.

New Text: New text is in red.

Table 161. Signals With On-Die Termination (ODT) or Pullups/ Pulldowns

Intel® QuickPath Interface Signal Group	
QPI_RX_DP/N[19:0], QPI_TX_DP/N[19:0], QPI_CLKRX_DP/N, QPI_CLKTX_DP/N	50 Ohm (Typical) to V_{SS}
PCI Express Signals	
PE_RX_DP/N[15:0], PE_TX_DP/N[15:0]	50 Ohm (Typical) to V_{SS}
DDR3 Signal Group	
DDRA/B/C_DQ[63:0], DDRA/B/C_DQS_P/N[17:0], DDRA/B/C_ECC[7:0]	See Table 167
DDRA/B/C_PAR_ERR#[2:0]	See Table 167
Control Sideband Signals	
BPM#[7:0], PREQ#	35 Ohm (Typical) to V_{TT}
PECI_ID#	1K - 5K Ohm to V_{TT}
DP_SYNCRST#, EXTSYSTRG, PMSYNC, DDR_ADR	50 Ohm (Typical) to V_{TT}
Test Access Port (TAP) Signal Group	
TCK	1K - 5K Ohm to V_{SS}
TDI, TMS, TRST#	10K (Typical) Ohm to V_{TT}
TDI_M	50 Ohm (Typical) to V_{TT}
Power/Other Signal Group	
VCCPWRGOOD, VTTPWRGOOD, DDR_DRAMPWROK	5K - 20K to V_{SS}

Affected Docs:Intel® Xeon® Processor C5500/C3500 Series Datasheet, Volume 1

Document Changes

The document changes listed in this section may apply to the following documents:

- Intel® Xeon® Processor C5500/C3500 Series Datasheet, Volume 1
- Intel® Xeon® Processor C5500/C3500 Series Datasheet, Volume 2
-

All documentation changes will be incorporated into a future version of the appropriate document.

1. The Text for Section 4.2.1, "Intel® QuickData Technology" has Changed

Old Text: Intel® QuickData Technology makes Intel® chipsets excel with Intel network controllers. The Intel® Xeon® processor C5500/C3500 series uses the third generation of the Intel® QuickData Technology.

The Intel® Xeon® processor C5500/C3500 series supports Intel® QuickData Technology. A NIC that is Intel® QuickData Technology capable can be plugged into any of the processor PCIe* ports, or be plugged into a PCIe ports below the PCH, and use the Intel® QuickData Technology capabilities.

New Text: Intel® QuickData Technology makes Intel® chipsets excel with Intel network controllers. The Intel® Xeon® processor C5500/C3500 series uses the third generation of the Intel® QuickData Technology.

The Intel® Xeon® processor C5500/C3500 series supports Intel® QuickData Technology. A NIC that is Intel® QuickData Technology capable can be plugged into any of the local processor PCIe* ports, or be plugged into a PCIe ports below the PCH if the PCH is directly attached to the processor, and use the Intel® QuickData Technology capabilities. In a dual-processor system, the Intel® QuickData Technology on one socket may not be used with PCIe devices attached to the other socket.

Affected Docs: Intel® Xeon® Processor C5500/C3500 Series Datasheet, Volume 1

2. ERR[2:0] has Changed to SYS_ERR_STAT[2:0] in Several Places in the Text

Old Text: Section 3.19.2.4, PCISTS PCI Status Register, bit 14
Section 3.20.2.4, PCISTS: PCI Status Register, bit 14
Section 11.3.2: fourth bullet under "Asynchronous inband error reporting"

Affected Docs: Intel® Xeon® Processor C5500/C3500 Series Datasheet, Volume 1

3. The Description for the PE_CFG[2:0] Signal in Table 140, "PCI Express* Signals" has Changed

Old Text: The desription has been replaced with a reference to Table 68, "Link Width Strapping Options."

Table 140. PCI Express Signals

Signal Names	I/O Type	Description
PE_CFG[2:0]	I/O	PCI Express* Port Bifurcation Configuration: 111 = One x16 PCI Express I/O. 110 = Two x8 PCI Express I/O. 101 = Four x4 PCI Express I/O. 100 = Wait for BIOS to configure PCI Express I/O. 011 = One x8 (port 1-2) and two x4 PCI Express I/O. 010 = Two x4 and one x8 (port 3-4) PCI Express I/O. 001 = Reserved. 000 = Reserved.

New Text: Changed text in red:

Table 140. PCI Express Signals

Signal Names	I/O Type	Description
PE_CFG[2:0]	I/O	See Table 68. "Link Width Strapping Options."

Affected Docs:Intel® Xeon® Processor C5500/C3500 Series Datasheet, Volume 1

4. The V_{tta} and V_{ttd} Minimum and Maximum Values in Table 161, "Processor Absolute Minimum and Maximum Ratings" has Changed

Old Text: The following two rows in the table are changed

Table 161. Processor Absolute Minimum and Maximum Ratings

Symbol	Parameter	Min	Nominal	Max	Unit	Notes[1,2]
V_{TTA}	Processor uncore analog voltage with respect to V_{SS}			1.155	V	3
V_{TTD}	Processor uncore digital voltage with respect to V_{SS}			1.155	V	3

New Text: Changed text in red:

Table 161. Processor Absolute Minimum and Maximum Ratings

Symbol	Parameter	Min	Nominal	Max	Unit	Notes[1,2]
V_{TTA}	Processor uncore analog voltage with respect to V_{SS}	-0.3		1.21	V	3
V_{TTD}	Processor uncore digital voltage with respect to V_{SS}	-0.3		1.21	V	3

Affected Docs:Intel® Xeon® Processor C5500/C3500 Series Datasheet, Volume 1

5. The VID Maximum Value in Table 162, "Voltage and Current Specifications" has Changed

Old Text: The following row in the table has changed:

Table 162. Voltage and Current Specifications

Symbol	Parameter	Voltage Plane	Min	Typ	Max	Unit	Notes[1]
VTT_VID	V_{tt} VID Range	-	1.045		1.220	V	2, 3

New Text: Changed text in red:

Table 162. Voltage and Current Specifications

Symbol	Parameter	Voltage Plane	Min	Typ	Max	Unit	Notes[1]
VTT_VID	V_{tt} VID Range	-	1.075		1.075	V	2, 3

Affected Docs:Intel® Xeon® Processor C5500/C3500 Series Datasheet, Volume 1

6. The Row that Includes the VTT Symbol in Table 162, "Voltage and Current Specifications" has Been Changed

Old Text: The following row in the table and note 8 have been changed:

Table 162. Voltage and Current Specifications

Symbol	Parameter	Voltage Plane	Min	Typ	Max	Unit	Notes[1]
V_{TT}	Uncore Voltage (Launch - FMB)	V_{TT}	See Table 173			V	3,5,8,11

8. See Table 173 and corresponding Figure 94. Do not subject processor to any static Vtt level exceeding Vtt_max associated with any particular current. Failure to adhere to this specification can shorten processor lifetime.

New Text: Changed text in red:

Table 162. Voltage and Current Specifications

Symbol	Parameter	Voltage Plane	Min	Typ	Max	Unit	Notes[1]
V_{TT}	Uncore Voltage	V_{TT}	(VTT_VID + 25mV) -5%		(VTT_VID + 25mv) +5%	V	5,8,11

8. Do not subject processor to any static Vtt level exceeding Vtt_max associated with any particular current. Failure to adhere to this specification can shorten processor lifetime. Specification includes AC and DC tolerances.

Affected Docs:Intel® Xeon® Processor C5500/C3500 Series Datasheet, Volume 1

7. The TDP for SKU P1053 in Table 162, "Voltage and Current Specifications" has Changed

Old Text: The following row in the table has changed:

Table 162. Voltage and Current Specifications

Symbol	Parameter	Voltage Plane	Min	Typ	Max	Unit	Notes[1]
I_{CC_MAX} I_{CCPLL_MAX} I_{DDQ_MAX} I_{TT_MAX}	P1053: TDP = 50W	V_{CC} V_{CCPLL} V_{DDQ} V_{TTA} V_{TTD}			13 1.5 4.5 4.2 22	A	11
	LC3528: TDP = 32W	V_{CC} V_{CCPLL} V_{DDQ} V_{TTA} V_{TTD}			24 1.5 5.5 4 15	A	11
	LC3518: TDP = 23W	V_{CC} V_{CCPLL} V_{DDQ} V_{TTA} V_{TTD}			11 1.5 4.5 4 13	A	11

New Text: Changed text in red:

Table 162. Voltage and Current Specifications

Symbol	Parameter	Voltage Plane	Min	Typ	Max	Unit	Notes[1]
I_{CC_MAX} I_{CCPLL_MAX} I_{DDQ_MAX} I_{TT_MAX}	P1053: TDP = 30W	V_{CC} V_{CCPLL} V_{DDQ} V_{TTA} V_{TTD}			13 1.5 4.5 4.2 22	A	11
	LC3528: TDP = 35W	V_{CC} V_{CCPLL} V_{DDQ} V_{TTA} V_{TTD}			24 1.5 5.5 4 15	A	11
	LC3518: TDP = 23W	V_{CC} V_{CCPLL} V_{DDQ} V_{TTA} V_{TTD}			11 1.5 4.5 4 13	A	11

Affected Docs:Intel® Xeon® Processor C5500/C3500 Series Datasheet, Volume 1

Affected Docs:

8. Register Description for MC_CLOSED_LOOP[2:0] has Changed

Old Text: The following row has changed in Section 4.15.7:

4	RW	0	**REF_2X_NOW.** Direct control of dynamic 2X refresh if MC_THERMAL_CONTROL.THROTTLE_MODE = 2.

New Text: Changed text in red::

4	RW	0	**REF_2X_NOW.** Direct control of dynamic 2X refresh if MC_THERMAL_CONTROL.THROTTLE_MODE = 2. This bit must only be set when MC_CHANNEL_X_REFRESH_THROTTLE_SUPPORT.ASR_PRESENT bit is set.

Affected Docs:Intel® Xeon® Processor C5500/C3500 Series Datasheet, Volume 2

9. Table 70. Inbound PCI Express Messages Supported has Changed

Old Text: The following row has changed:

Vendor-defined	ASSERT_GPE DEASSERT_GPE (Intel-specific)	Vendor-specific message indicating assertion/ deassertion of PCI-X hotplug event in PXH. Message forwarded to DMI port.
	MCTP	Management Control Transport Protocol messages - forwards MCTP messages received on its PCI-E ports to PCH over DMI interface.
	All Other Messages	Silently discard if message type is type 1 and drop and log error if message type is type 0

New Text: Changed text in red:

	ASSERT_GPE¶ DEASSERT_GPE¶ (Intel-specific)§	Vendor-specific message indicating assertion/ deassertion of PCI-X hotplug event in PXH. Message forwarded to DMI port.§
Vendor-defined§	MCTP §	Management Control Transport Protocol messages are silently discarded.§
	All Other Messages§	Silently discard if message type is type 1 and drop and log error if message type is type 0§

Affected Docs:Intel® Xeon® Processor C5500/C3500 Series Datasheet, Volume 1

10. The POC Settings for MSID[2:0] POC Bits Allocation has Changed

Old Text: Table 157 in the Datasheethas changed:

Table 157. Power-On Configuration (POC[7:0]) Decode

Function	Bits	POC Settings		Description
MSID[2:0]	VID[2:0]	000 001 010 011 100 101 110 111	Undefined Undefined Undefined 60W TDP / 80A ICC_MAX 80W TDP / 100A ICC_MAX 95W TDP / 120A ICC_MAX 130W TDP / 150A ICC_MAX Undefined	MSID[2:0] signals are provided to indicate the Market Segment for the processor and may be used for future processor compatibility or keying. See Figure 86 for platform timing requirements of the MSID[2:0] signals.

New Text: Changed text in red:

Table 157. Power-On Configuration (POC[7:0]) Decode

Function	Bits	POC Settings		Description
MSID[2:0]	VID[2:0]	000 001 010 011 100 101 110 111	30W TDP / 30A ICC_MAX 40W TDP / 40A ICC_MAX 55W TDP / 60A ICC_MAX 65W TDP / 80A ICC_MAX 80W TDP / 100A ICC_MAX 95W TDP / 120A ICC_MAX 130W TDP / 150A ICC_MAX Undefined	MSID[2:0] signals are provided to indicate the Market Segment for the processor and may be used for future processor compatibility or keying. See Figure 86 for platform timing requirements of the MSID[2:0] signals.

Affected Docs:

- Intel® Xeon® Processor C5500/C3500 Series Datasheet, Volume 1
-

11. Text Deleted from Section 11.3.3.7.1, Feature Requirements

Old Text: The following text has been deleted:

- Each x4 PCI Express port contains one 7-bit counter (ERRCNT[6:0]) with a correctable error status selection register.
  - — Bit[7] is an overflow bit; all bits are sticky with a write logic 1 to clear.
- The DMI port contains one 7-bit counter (ERRCNT[6:0]) with a correctable error status selection register.
  - — Bit[7] is an overflow bit; all bits are sticky with a write logic 1 to clear.

New Text: There is no replacement text.

Affected Docs:Intel® Xeon® Processor C5500/C3500 Series Datasheet, Volume 1

12. Text Deleted from 179, Feature Requirements

Old Text: The sentence, “This signal has an ODT pull-up” has been deleted for TRST#, TMS, TDI, and TDI_M in the following table:

Table 179. Processor Boundary-Scan TAP Pin Interface (Datasheet)

Pin	Direction	Description
TRST#	Input	Boundary-scan test reset pin. This signal has an ODT pull-up
TCK	Input	Test clock pin for Boundary-scan TAP Controller and test logic. This signal has an ODT pull-down.
TMS	Input	Boundary-scan Test Mode Select pin. Sampled by the TAP on the Rising edge of TCK to control the operation of the TAP state machine. It is recommended that TMS is held high when the Boundary-scan reset is driven from low to high, to ensure deterministic operation of the test logic. This signal has an ODT pull-up.
TDI	Input	Boundary-scan Test Data Input pin, sampled on the Rising edge of TCK to provide serial test instructions and data. This signal has an ODT pull-up.
TDO	Output	Boundary-scan Test Data Output pin. In inactive drive state except when instructions or data are being shifted. TDO changes on the falling edge of TCK. This signal has no pull-up or pull-down.
TDI_M	Input	Intermediate Boundary-scan connection. This signal must be connected to TDO_M for correct operation. This signal has an ODT pull-up.
TDO_M	Output	Intermediate Boundary-scan connection. This signal must be connected to TDI_M for correct operation. This signal has no pull-up or pull-down.

New Text: Affected rows/cells are in red:

Table 179. Processor Boundary-Scan TAP Pin Interface (Datasheet)

Pin	Direction	Description
TRST#	Input	Boundary-scan test reset pin.
TCK	Input	Test clock pin for Boundary-scan TAP Controller and test logic. This signal has an ODT pull-down.
TMS	Input	Boundary-scan Test Mode Select pin. Sampled by the TAP on the Rising edge of TCK to control the operation of the TAP state machine. It is recommended that TMS is held high when the Boundary-scan reset is driven from low to high, to ensure deterministic operation of the test logic.
TDI	Input	Boundary-scan Test Data Input pin, sampled on the Rising edge of TCK to provide serial test instructions and data.
TDO	Output	Boundary-scan Test Data Output pin. In inactive drive state except when instructions or data are being shifted. TDO changes on the falling edge of TCK. This signal has no pull-up or pull-down.
TDI_M	Input	Intermediate Boundary-scan connection. This signal must be connected to TDO_M for correct operation.
TDO_M	Output	Intermediate Boundary-scan connection. This signal must be connected to TDI_M for correct operation. This signal has no pull-up or pull-down.

Affected Docs:Intel® Xeon® Processor C5500/C3500 Series Datasheet, Volume 1

13. Deleted

14. Text Changed and Deleted from Table 155, Processor Power Supply Voltages

Old Text: Text has been deleted and changed in Table 155, Processor Power Supply Voltages, in Section 13Datasheet. Deleted text in red:

Table 1. Table 155. Processor Power Supply Voltages

Power Rail	Nominal Voltage	Notes
V_{CC}	See Table 163; Figure 86	Each processor includes a dedicated VR11.1 regulator.
V_{CCPLL}	1.80 V	Each processor includes dedicated V_{CCPLL} and PLL circuits.
V_{DDQ}	1.50 V	Each processor and DDR3 stack shares a dedicated voltage regulator.
V_{TTA}, V_{TTD}	See Table 173; Figure 94	Each processor includes a dedicated VR11.0 regulator. V_{TT} = V_{TTA} = V_{TTD}; P1V1_Vtt is VID[4:2] controlled, VID range is 1.0255-1.2000V. The tolerance is +/- 2% at the processor pin. (This assumes that the filter circuit described in the *Picket Post: Intel® Xeon® Processor C5500/C3500 Series with the Intel® 3420 Chipset Platform Design Guide (PDG)* is used.)

New Text: Changed text in red:

Table 155. Processor Power Supply Voltages

Power Rail	Nominal Voltage	Notes
V_{CC}	See Table 163; Figure 86	Each processor requires a dedicated VR11.1 regulator.
V_{CCPLL}	1.80 V	Each processor requires dedicated V_{CCPLL} and PLL circuits.
V_{DDQ}	1.50 V	Each processor and DDR3 stack shares a dedicated voltage regulator.
V_{TTA}, V_{TTD}	See Table 162	Each processor requires a dedicated VR11.0 regulator.

Affected Docs: Intel® Xeon® Processor C5500/C3500 Series Datasheet, Volume 2

15. Text Changed and Deleted in Section 3.1, Introduction

Old Text: When used in conjunction with Intel® VT-d2 both primary and secondary addresses are guest addresses. When Intel® VT-d2 is not used the secondary side of the bridge is a guest address and the primary side of the bridge is a physical address.

New Text: The NTB and Intel® VT-d2 may not both be enabled at the same time. The NTB is not compatible with Intel® VT-d2.

Affected Docs: Intel® Xeon® Processor C5500/C3500 Series Datasheet, Volume 1

16. Text Added in Section 3.2.1, Features Not Supported on the Intel® Xeon® Processor C5500/C3500 Series NTB

Old Text:

- NTB does not support x16 link configuration
- NTB does not support IO space BARs
- NTB does not support vendor defined PCIE message transactions. These messages are silently dropped if received.

New Text: New text is in red:

- NTB does not support x16 link configuration
- NTB does not support IO space BARs
- NTB does not support vendor defined PCIE message transactions. These messages are silently dropped if received.
- The NTB and Intel® Vt-d2 can not both be enabled at the same time.

Affected Docs: Intel® Xeon® Processor C5500/C3500 Series Datasheet, Volume 1

17. Text Deleted fom Section 3.6.6.1, Direct Address Translation

Old Text: Deleted text in red:

- Address Translation
  - — Inbound with VT-d2 turned off.
    - Translate a remote address to a local physical address.
  - — Inbound with VT-d2 turned on.
    - Translate a remote address to a local guest physical address that is then forwarded to the VT-d2 logic. The VT-d2 logic then converts the guest physical address to a host physical address.
  - — Outbound:
    - Translate a local physical address to a remote guest address.

New Text:

- Address Translation
  - — Inbound with VT-d2 turned off.
    - Translate a remote address to a local physical address.
  - — Outbound:
    - Translate a local physical address to a remote guest address.

Affected Docs: Intel® Xeon® Processor C5500/C3500 Series Datasheet, Volume 1

18. Text Added to Section 3.6.7.27, IIONFERRHD: IIO Core Non-Fatal FERR Header

Issue: Description was unclear for PCIe error register.

New Text: Additional and changed text in red:

Register: IIONFERRHD
Device:8
Function:2
Offset:324h

<table>
<tr><th>Bit</th><th>Attr</th><th>Default</th><th>Description</th></tr>
<tr><td>127:0</td><td>ROS</td><td>0</td><td>
IIO Core Error Header log
Header log stores the IIO data path header information of the associated IIO core error. The header indicates where the error is originating from and the address of the cycle.
[127:90] Reserved not used
[89] Error Type (MA = ‘0’, CA = ‘1’)
[88:81] PCIe Message Code [7:0] -- See PCI-E base spec
[80:65] MSI Data [15:0]
[64:58] IIO Internal Switch routing ID (SWRID [6:0]) -- See decode below
[57:51] PCIe {Fmt [1:0], Type[4:0]} -- See PCI-E base spec
[50:0] Address [50:0]
...
The two upper bits of the Destination ID (Dest ID[31:30]) will not be logged when Interrupt Remapping is enabled

NOTE: IIO Internal Switch routing ID is not equivalent to the PCIe requester ID. It is used to facilitate routing of completions through the switch.
<table>
<tr><th>SWRID[6:3]</th><th>SWRID[2:0]</th></tr>
<tr><td rowspan="5">0000: PCIe x16</td><td>000: Device 3</td></tr>
<tr><td>001: Device 4</td></tr>
<tr><td>010: Device 5</td></tr>
<tr><td>011: Device 6</td></tr>
<tr><td>0001: Reserved</td><td>n/a</td></tr>
<tr><td>0010: PCH</td><td>XXX</td></tr>
<tr><td rowspan="3">0011: CBDMA</td><td>000 – Descriptor fetch, M2M/M2IO Request for Ownership (RFO) for Writes</td></tr>
<tr><td>001 – M2M/M2IO Reads, Writes</td></tr>
<tr><td>010 – XOR Descriptor Reads/RFOs/Writes</td></tr>
<tr><td>0010: Reserved</td><td>n/a</td></tr>
<tr><td rowspan="8">0101: MISC</td><td>000: CFG</td></tr>
<tr><td>001, 010: VT</td></tr>
<tr><td>011: APIC</td></tr>
<tr><td>100: JSM</td></tr>
<tr><td>101: MSG</td></tr>
<tr><td>110: HPPM</td></tr>
<tr><td>111: SMBUS</td></tr>
<tr><td>0110: QPI (CPU-IIO I/F)</td><td>000: QPI</td></tr>
<tr><td>0111-> 1111: Reserved</td><td>n/a</td></tr>
</table>
</td></tr>
</table>

Affected Docs:Intel® Xeon® Processor C5500/C3500 Series Datasheet, Volume 2

19. Text Added to Section 3.6.7.24, IIOFFERRHD: IIO Core Fatal FERR Header

Issue: The description was unclear for the PCIe error register.

New Text: Additional and changed text in red:

Register: IIOFFERRHD
Device:8
Function:2
Offset:30Ch

<table>
<tr><th>Bit</th><th>Attr</th><th>Default</th><th>Description</th></tr>
<tr><td>127:0</td><td>ROS</td><td>0</td><td>
IIO Core Error Header log
Header log stores the IIO data path header information of the associated IIO core error. The header indicates where the error is originating from and the address of the cycle.
[127:90] Reserved not used
[89] Error Type (MA = ‘0’, CA = ‘1’)
[88:81] PCIe Message Code [7:0] -- See PCI-E base spec
[80:65] MSI Data [15:0]
[64:58] IIO Internal Switch routing ID (SWRID [6:0]) -- See decode below
[57:51] PCIe {Fmt [1:0], Type[4:0]} -- See PCI-E base spec
[50:0] Address [50:0]
Note: For interrupts Address(50:0) will be logged as follows when Interrupt Remapping is enabled:
Address(50:19) = DW Address = (Dest ID[29:0], Redirect Hint, Mode)
Mode denotes 0 for physical and 1 for logical
Address(18:0) - NA for interrupts and could be zeros or ones.
The two upper bits of the Destination ID (Dest ID[31:30]) will not be logged when Interrupt Remapping is enabled

NOTE: IIO Internal Switch routing ID is not equivalent to the PCIe requester ID. It is used to facilitate routing of completions through the switch.

<table>
<tr><th>SWRID[6:3]</th><th>SWRID[2:0]</th></tr>
<tr><td rowspan="4">0000: PCIe x16</td><td>000: Device 3</td></tr>
<tr><td>001: Device 4</td></tr>
<tr><td>010: Device 5</td></tr>
<tr><td>011: Device 6</td></tr>
<tr><td>0001: Reserved</td><td>n/a</td></tr>
<tr><td>0010: PCH</td><td>XXX</td></tr>
<tr><td rowspan="3">0011: CBDMA</td><td>000 – Descriptor fetch, M2M/M2IO Request for Ownership (RFO) for Writes</td></tr>
<tr><td>001 – M2M/M2IO Reads, Writes</td></tr>
<tr><td>010 – XOR Descriptor Reads/RFOs/Writes</td></tr>
<tr><td>0010: Reserved</td><td>n/a</td></tr>
<tr><td rowspan="8">0101: MISC</td><td>000: CFG</td></tr>
<tr><td>001, 010: VT</td></tr>
<tr><td>011: APIC</td></tr>
<tr><td>100: JSM</td></tr>
<tr><td>101: MSG</td></tr>
<tr><td>110: HPPM</td></tr>
<tr><td>111: SMBUS</td></tr>
<tr><td>0110: QPI (CPU-IIO I/F)</td><td>000: QPI</td></tr>
<tr><td>0111-> 1111: Reserved</td><td>n/a</td></tr>
</table>
</td></tr>
</table>

Affected Docs:Intel® Xeon® Processor C5500/C3500 Series Datasheet, Volume 2

20. Register Added: LA_DPCNTRL: Lock Arbiter Dependent Port Control

Issue: A new section 3.6.5.24 has been added to Volume 2 of the Datasheet

New Text: New text is in red:

3.6.5.24 LA_DPCNTRL: Lock Arbiter Dependent Port Control

This register is used to define which devices need special handling by the switch lock arbiter due to posted to posted dependencies created during lock flows. See section TBD for information on how to set this register. This register must be setup by BIOS prior to operation.

Register: LA_DPPCNTRL
Device: 8
Function: 2
Offset: 0F0h

Bit	Attr	Default	Description
31	RW	0b	Lock arbiter connectivity. This bit must be set when connecting C5500/C3500 Series IIO (legacy) to a Tylersburg IIO (non-legacy) via CSI. When connecting C5500/C3500 Series to C5500/C3500 Series (DP) via CSI1 or C5500/C3500 Series UP this bit is cleared. 0 = C5500/C3500 Series UP (no connection on CSI1) / C5500/C3500 Series DP 1 = C5500/C3500 Series UP connected to Tylersburg via CSI ***Note:*** The C5500/C3500 Series must always be the legacy IIO in the C5500/C3500 Series /Tylersburg configuration.
30:07	**RV**	0000000h	**Reserved**
06	RW	0b	Posted to posted dependence (PCIE Device 6) 0 = Non dependent port 1 = Dependent port
05	RW	0b	Posted to posted dependence (PCIE Device 5) 0 = Non dependent port 1 = Dependent port
04	RW	0b	Posted to posted dependence (PCIE Device 4) 0 = Non dependent port 1 = Dependent port
03	RW	0b	Posted to posted dependence (PCIE Device 3) 0 = Non dependent port 1 = Dependent port
02:01	**RV**	00b	**Reserved**
00	RW	0b	Posted to posted dependence (PCIE/DMI Device 0) 0 = Non dependent port 1 = Dependent port

Affected Docs:Intel® Xeon® Processor C5500/C3500 Series Datasheet, Volume 2

21. Text Added to Section 3.6.7.37, MINFERRHD: Miscellaneous Local Non-Fatal FERR Header

Issue: Description was unclear for PCIe error register.

New Text: Additional and changed text in red:

Register:MINFERRHD
Device:8
Function:2
Offset:3A4h

<table>
<tr><th>Bit</th><th>Attr</th><th>Default</th><th>Description</th></tr>
<tr><td>127:0</td><td>ROS</td><td>0</td><td>
Miscellaneous Error Header log
Header log stores the IIO data path header information of the associated IIO miscellaneous error. The header indicates where the error is originating from and the address of the cycle.
[127:90] Reserved not used
[89] Error Type (MA = ‘0’, CA = ‘1’)
[88:81] PCIe Message Code [7:0] -- See PCI-E base spec
[80:65] MSI Data [15:0]
[64:58] IIO Internal Switch routing ID (SWRID [6:0]) -- See decode below
[57:51] PCIe {Fmt [1:0], Type[4:0]} -- See PCI-E base spec
[50:0] Address [50:0]
NOTE: IIO Internal Switch routing ID is not equivalent to the PCIe requester ID. It is used to facilitate routing of completions through the switch.
<table>
<tr><th>SWRID[6:3]</th><th>SWRID[2:0]</th></tr>
<tr><td rowspan="4">0000: PCIe x16</td><td>000: Device 3</td></tr>
<tr><td>001: Device 4</td></tr>
<tr><td>010: Device 5</td></tr>
<tr><td>011: Device 6</td></tr>
<tr><td>0001: Reserved</td><td>n/a</td></tr>
<tr><td>0010: PCH</td><td>XXX</td></tr>
<tr><td rowspan="3">0011: CBDMA</td><td>000 – Descriptor fetch, M2M/M2IO Request for Ownership (RFO) for Writes</td></tr>
<tr><td>001 – M2M/M2IO Reads, Writes</td></tr>
<tr><td>010 – XOR Descriptor Reads/RFOs/Writes</td></tr>
<tr><td>0010: Reserved</td><td>n/a</td></tr>
<tr><td rowspan="7">0101: MISC</td><td>000: CFG</td></tr>
<tr><td>001, 010: VT</td></tr>
<tr><td>011: APIC</td></tr>
<tr><td>100: JSM</td></tr>
<tr><td>101: MSG</td></tr>
<tr><td>110: HPPM</td></tr>
<tr><td>111: SMBUS</td></tr>
<tr><td>0110: QPI (CPU-IIO I/F)</td><td>000: QPI</td></tr>
<tr><td>0111-> 1111: Reserved</td><td>n/a</td></tr>
</table>
</td></tr>
</table>

Affected Docs:Intel® Xeon® Processor C5500/C3500 Series Datasheet, Volume 2

22. Deleted.

23. Text Deleted from Table 189, Thermal Signal Group AC Specifications

Old Text: Deleted text is shown in red.

T# Parameter	Min	Max	Unit	Figure	Notes 1,2,3,4
Tq: PROCHOT# pulse width	500		µs	117	
Tr:THERMTRIP# assertion until V_{CC} / V_{TT} removed		500	ms	118	
T_{SU}: Input signals with respect to BCLK	600		ps		5
T_H: Input signals with respect to BCLK	600		ps		5
T_{SU}: Power-On Configuration Setup Time (PROCHOT#)	2		BCLK	117	5, 6
T_H: Power-On Configuration Hold Time (PROCHOT#)	106		BCLK	117	5, 6

Notes:
1. Unless otherwise noted, all specifications in this table apply to all processor frequencies.
2. All AC timings for the Asynchronous GTL signals are referenced to the BCLK_P rising edge at Crossing Voltage (V_{CROSS}). VCCPWRGOOD, VTTPWRGOOD and DDR_DRAMPWROK are referenced to BCLK_P rising edge at 0.5 * V_{TT}.
3. These signals may be driven asynchronously.
4. See Section 8.0, "Power Management" for additional timing requirements for entering and leaving low power states.

New Text:

T# Parameter	Min	Max	Unit	Figure	Notes 1,2,3,4
Tq: PROCHOT# pulse width	500		µs	117	
Tr:THERMTRIP# assertion until V_{CC} / V_{TT} removed		500	ms	118	

Notes:
1. Unless otherwise noted, all specifications in this table apply to all processor frequencies.
2. All AC timings for the Asynchronous GTL signals are referenced to the BCLK_P rising edge at Crossing Voltage (V_{CROSS}). VCCPWRGOOD, VTTPWRGOOD and DDR_DRAMPWROK are referenced to BCLK_P rising edge at 0.5 * V_{TT}.
3. These signals may be driven asynchronously.
4. See Section 8.0, "Power Management" for additional timing requirements for entering and leaving low power states.
5. Specified for synchronous signals.
6. Applies to PROCHOT# signal only. See Section 13.1.10.3.2 and Section 8.1 for information regarding Power-On Configuration options.

Affected Docs:

24. New Section Added Before Section 3.8, Outbound Transactions

New Text: Section 3.8 is now Deadlock Avoidance, pushing Outbound Transactions to Section 3.9. The new text is as follows.

3.8 Deadlock Avoidance

When connecting the NTB in a DP configuration and fail over mode. Deadlock condition may occur if circular posted to posted dependencies exist between the two systems.

During silicon validation stress testing the following synthetic deadlock was created.

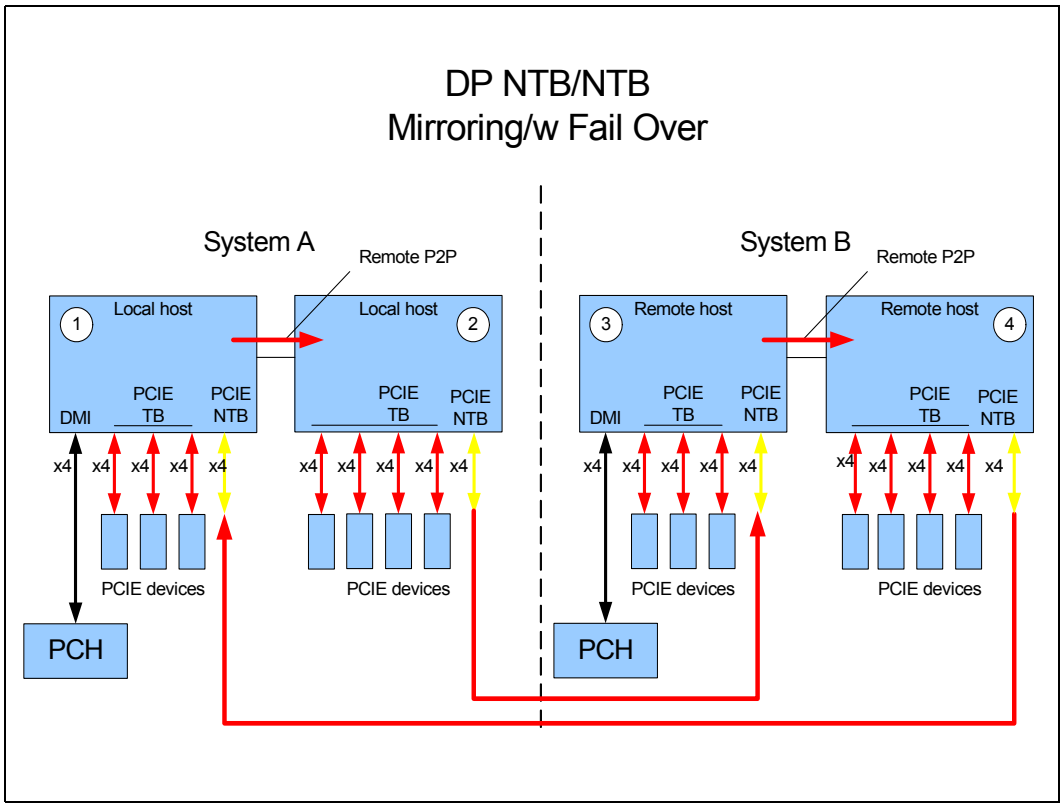
1. Dev3 (NTB) wants to request to memory, but no ORBs are available for RFO (posted prefetch). Device 4 and Device 6 are requesting to local memory and remote p2p. They have already sent RFOs and are holding onto the ORBs, but the ORBs cannot be released until the write data is sent to memory. For both devices, the write data to memory is blocked by a remote p2p request at the top of the queue. The remote p2p is blocked due to blockage in IIO-1.

2. NCBs from CPU and IIO-0 are blocked due to Device 3 (NTB) outbound write queue being full.

3. Dev3 (NTB) wants to request to memory, but no ORBs are available for RFO (posted prefetch). Device 4 and Device 6 are requesting to local memory and remote p2p. They have already sent RFOs and are holding onto the ORBs, but the ORBs cannot be released until the write data is sent to memory. For both devices, the write data to memory is blocked by a remote p2p request at the top of the queue. The remote p2p is blocked due to blockage in IIO-1.

4. NCBs from CPU and IIO-0 are blocked due to Device 3 (NTB) outbound write queue being full.

Figure 60. DP with Dual NTB Links Deadlock



If this case is created deadlock will occur. A workaround is needed to remove the full system closed loop posted to posted dependencies.

1. Remove Remote P2P operations.

2. Use only a single NTB link.

Affected Docs: Intel® Xeon® Processor C5500/C3500 Series Datasheet, Volume 1

25. Deleted Section 13.4

Section 13.4 has been deleted. Deleted text is shown below in red.

13.4 Flexible Motherboard Guidelines (FMB)

The Flexible Motherboard (FMB) guidelines are estimates of the maximum values the Intel® Xeon® processor C5500/C3500 series will have over certain time periods. The values are only estimates and actual specifications for future processors may differ. Processors may or may not have specifications equal to the FMB value in the foreseeable future. System designers should meet the FMB values to ensure their systems will be compatible with future Bloomfield processor.

Affected Docs:Intel® Xeon® Processor C5500/C3500 Series Datasheet, Volume 1

§ §