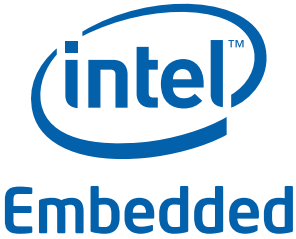


Intel® Xeon® and Intel® Core™ Processors For Communications Infrastructure

Specification Update

March 2014

Supporting Intel® Xeon® Processor E3-1125C

Supporting Intel® Xeon® Processor E3-1105C

Supporting Intel® Core™ i3 Processor 2115C

Supporting Intel® Pentium® Processor B915C

Supporting Intel® Celeron® Processor 725C

Notice: The *2nd Generation Intel®Core™ Processor Family For Communications Infrastructure* 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.

Document Number: 327335-003

Contents

Revision History

Date	Revision	Description
March 2014	003	Updated Table 5, “Errata” Added Document-Only Changes for On-Demand Clock Modulation Feature Clarification Added Errata BM115, BM116, BM117, BM118, BM119, BM120, BM121, BM122, BM123, BM124, BM125, BM126, BM128, BM129 Modified BM58 Replaced BM61 and corrected numbering from BM61 forward.
August 2013	002	Corrected Specification Changes numbering Updated Table 11, “Processor Identification” Updated Table 5, “Errata” Added Specification Clarifications for System Agent Vcc VID and DC Specifications Added Document-Only Changes for On-Demand Clock Modulation Feature Clarification Added Errata BM104, BM105, BM106, BM107, BM108, BM109, BM110, BM111, BM112, BM113, BM114, BM115, BM116, BM117, BM118, BM119, BM120, BM121, BM122, BM123, BM124, BM125, BM126
May 2012	001	Initial public release.

Introduction

Purpose / Scope / Audience

This document is an update to the specifications contained in the Affected Documents below, and is a compilation of Errata, Specification Changes, Specification Clarifications, and Document-Only 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.

Affected Documents

Table 1. Affected Documents

Document	Document Number / Location
Intel® Xeon® Processor and Intel® Core™ Processor For Communications Infrastructure Datasheet - Volume 1 of 2	327405; http://download.intel.com/embedded/processor/datasheet/327405.pdf
2nd Generation Intel® Core™ Mobile Processor Datasheet - Volume 2 of 2	324803; http://www.intel.com/content/dam/doc/datasheet/2ndgen-core-family-mobile-vol-2-datasheet.pdf

Related Documents

Table 2. Related Documents

Document	Document Number / Location
Intel® Processor Identification and the CPUID Instruction - Application Note 485	http://www.intel.com/design/processor/applnots/241618.htm
Intel® 64 and IA-32 Architectures Software Developer's Manuals: • *Volume 1: Basic Architecture* • *Volume 2A: Instruction Set Reference, A-M* • *Volume 2B: Instruction Set Reference, N-Z* • *Volume 3A: System Programming Guide* • *Volume 3B: System Programming Guide* • *Optimization Reference Manual*	http://www.intel.com/products/processor/manuals/index.htm
Intel® 64 and IA-32 Architectures Software Developer's Manual Documentation Changes	http://www.intel.com/content/www/us/en/architecture-and-technology/64-ia-32-architectures-software-developers-manual.html
Intel® Virtualization Technology Specification for Directed I/O Architecture Specification	http://download.intel.com/technology/computing/vptech/Intel(r)_VT_for_Direct_IO.pdf

Conventions and Terminology

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-Only Changes are removed from the Specification Update when the appropriate changes are made to the appropriate product specification or user documentation (datasheets, manuals, etc.).

Table 3. Conventions and Terminology

Term	Definition
Errata (plural) Erratum (singular)	Errata are design defects or errors. These may cause the *2nd Generation Intel®Core™ Processor Family For Communications Infrastructure*'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.
S-Spec Number	S-Spec Number is a five-digit code used to identify products. Products are differentiated by their unique characteristics such as, core speed, L2 cache size, package type, etc. as described in the processor identification information table. Read all notes associated with each S-Spec number.
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.
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.
Document-Only Changes	Document-Only 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.

Summary Tables

Table 5 through Table 8 summarize the Errata, Specification Changes, Specification Clarifications, or Document-Only Changes that apply to the *2nd Generation Intel® Core™ Processor Family For Communications Infrastructure* product.

Intel may fix some of the Errata in a future stepping of the component as noted in Table 4 or account for the other outstanding issues through Specification Changes, Specification Clarifications, or Document-Only Changes. Table 5 uses 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 e**X**ists 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. Errata (Sheet 1 of 5)

No.	Steppings		Status	Errata
	D-2	Q-0		
BM1	X	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
BM2	X	X	No Fix	APIC Error “Received Illegal Vector” May be Lost
BM3	X	X	No Fix	An Uncorrectable Error Logged in IA32_CR_MC2_STATUS May also Result in a System Hang
BM4	X	X	No Fix	B0-B3 Bits in DR6 For Non-Enabled Breakpoints May be Incorrectly Set
BM5	X	X	No Fix	Changing the Memory Type for an In-Use Page Translation May Lead to Memory-Ordering Violations
BM6	X	X	No Fix	Code Segment Limit/Canonical Faults on RSM May be Serviced before Higher Priority Interrupts/Exceptions and May Push the Wrong Address Onto the Stack
BM7	X	X	No Fix	Corruption of CS Segment Register During RSM While Transitioning From Real Mode to Protected Mode
BM8	X	X	No Fix	Debug Exception Flags DR6.B0-B3 Flags May be Incorrect for Disabled Breakpoints
BM9	X	X	No Fix	DR6.B0-B3 May Not Report All Breakpoints Matched When a MOV/POP SS is Followed by a Store or an MMX Instruction

Table 5. Errata (Sheet 2 of 5)

No.	Steppings		Status	Errata
	D-2	Q-0		
BM10	X	X	No Fix	EFLAGS Discrepancy on Page Faults and on EPT-Induced VM Exits after a Translation Change
BM11	X	X	No Fix	Fault on ENTER Instruction May Result in Unexpected Values on Stack Frame
BM12	X	X	No Fix	Faulting MMX Instruction May Incorrectly Update x87 FPU Tag Word
BM13	X	X	No Fix	FREEZE_WHILE_SMM Does Not Prevent Event From Pending PEBS During SMM
BM14	X	X	No Fix	General Protection Fault (#GP) for Instructions Greater than 15 Bytes May be Preempted
BM15	X	X	No Fix	#GP on Segment Selector Descriptor that Straddles Canonical Boundary May Not Provide Correct Exception Error Code
BM16	X	X	No Fix	IO_SMI Indication in SMRAM State Save Area May be Set Incorrectly
BM17	X	X	No Fix	IRET under Certain Conditions May Cause an Unexpected Alignment Check Exception
BM18	X	X	No Fix	LER MSRs May Be Unreliable
BM19	X	X	No Fix	LBR, BTS, BTM May Report a Wrong Address when an Exception/Interrupt Occurs in 64-bit Mode
BM20	X	X	No Fix	MCi_Status Overflow Bit May Be Incorrectly Set on a Single Instance of a DTLB Error
BM21	X	X	No Fix	MONITOR or CLFLUSH on the Local XAPIC's Address Space Results in Hang
BM22	X	X	No Fix	MOV To/From Debug Registers Causes Debug Exception
BM23	X	X	No Fix	PEBS Record not Updated when in Probe Mode
BM24	X	X	No Fix	Performance Monitoring Event FP_MMX_TRANS_TO_MMX May Not Count Some Transitions
BM25	X	X	No Fix	REP MOVS/STOS Executing with Fast Strings Enabled and Crossing Page Boundaries with Inconsistent Memory Types may use an Incorrect Data Size or Lead to Memory-Ordering Violations
BM26	X	X	No Fix	Reported Memory Type May Not Be Used to Access the VMCS and Referenced Data Structures
BM27	X	X	No Fix	Single Step Interrupts with Floating Point Exception Pending May Be Mishandled
BM28	X	X	No Fix	Storage of PEBS Record Delayed Following Execution of MOV SS or STI
BM29	X	X	No Fix	The Processor May Report a #TS Instead of a #GP Fault
BM30	X	X	No Fix	VM Exits Due to "NMI-Window Exiting" May Be Delayed by One Instruction
BM31	X	X	No Fix	Pending x87 FPU Exceptions (#MF) May be Signaled Earlier Than Expected
BM32	X	X	No Fix	Values for LBR/BTS/BTM Will be Incorrect after an Exit from SMM
BM33	X	X	No Fix	Unsupported PCIe* Upstream Access May Complete with an Incorrect Byte Count
BM34	X	X	No Fix	Malformed PCIe* Transactions May be Treated as Unsupported Requests Instead of as Critical Errors
BM35	X	X	No Fix	PCIe* Root Port May Not Initiate Link Speed Change
BM36	X	X	No Fix	Incorrect Address Computed For Last Byte of FXSAVE/FXRSTOR or XSAVE/XRSTOR Image Leads to Partial Memory Update
BM37	X	X	No Fix	Performance Monitor SSE Retired Instructions May Return Incorrect Values
BM38	X	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
BM39	X	X	No Fix	FP Data Operand Pointer May Be Incorrectly Calculated After an FP Access Which Wraps a 64-KByte Boundary in 16-Bit Code
BM40	X	X	No Fix	Spurious Interrupts May be Generated From the Intel® VT-d Remap Engine
BM41	X	X	No Fix	Fault Not Reported When Setting Reserved Bits of Intel® VT-d Queued Invalidation Descriptors

Table 5. Errata (Sheet 3 of 5)

No.	Steppings		Status	Errata
	D-2	Q-0		
BM42	X	X	No Fix	VPHMINPOSUW Instruction in Vex Format Does Not Signal #UD When vex.vvvv !=1111b
BM43	X	X	No Fix	LBR, BTM or BTS Records May have Incorrect Branch From Information After an EIST/T-state/S-state/C1E Transition or Adaptive Thermal Throttling
BM44	X	X	No Fix	VMREAD/VMWRITE Instruction May Not Fail When Accessing an Unsupported Field in VMCS
BM45	X	X	No Fix	Clock Modulation Duty Cycle Cannot be Programmed to 6.25%
BM46	X	X	No Fix	Execution of VAESIMC or VAESKEYGENASSIST With An Illegal Value for VEX.vvvv May Produce a #NM Exception
BM47	X	X	No Fix	Memory Aliasing of Code Pages May Cause Unpredictable System Behavior
BM48	X	X	No Fix	PCI Express Receiver Error Reported When Receiver With L0s Enabled and Link Retrain Performed
BM49	X	X	No Fix	Unexpected #UD on VZEROALL/VZEROUPPER
BM50	X	X	No Fix	Perfmon Event LD_BLOCKS.STORE_FORWARD May Overcount
BM51	X	X	No Fix	Not Applicable
BM52	X	X	No Fix	Execution of Opcode 9BH with the VEX Opcode Extension May Produce a #NM Exception
BM53	X	X	No Fix	Executing The GETSEC Instruction While Throttling May Result in a Processor Hang
BM54	X	X	No Fix	A Write to the IA32_FIXED_CTR1 MSR May Result in Incorrect Value in Certain Conditions
BM55	X	X	No Fix	Instruction Fetch May Cause Machine Check if Page Size and Memory Type Was Changed Without Invalidation
BM56	X	X	No Fix	Reception of Certain Malformed Transactions May Cause PCIe* Port to Hang Rather Than Reporting an Error
BM57	X	X	No Fix	PCIe* LTR Incorrectly Reported as Being Supported
BM58	X	X	No Fix	Performance-Counter Overflow Indication May Cause Undesired Behavior
BM59	X	X	No Fix	XSAVE Executed During Paging-Structure Modification May Cause Unexpected Processor Behavior
BM60	X	X	No Fix	C-state Exit Latencies May be Higher Than Expected
BM61	X	X	No Fix	Intel® VT-d Interrupt Remapping Will Not Report a Fault if Interrupt Index Exceeds FFFFH
BM62	X	X	No Fix	PCIe* Link Speed May Not Change From 5.0 GT/s to 2.5 GT/s
BM63	X	X	No Fix	L1 Data Cache Errors May be Logged With Level Set to 1 Instead of 0
BM64	X	X	No Fix	An Unexpected Page Fault or EPT Violation May Occur After Another Logical Processor Creates a Valid Translation for a Page
BM65	X	X	No Fix	TSC Deadline Not Armed While in APIC Legacy Mode
BM66	X	X	No Fix	PCIe* Upstream TCfgWr May Cause Unpredictable System Behavior
BM67	X	X	No Fix	Processor May Fail to Acknowledge a TLP Request
BM68	X	X	No Fix	Executing The GETSEC Instruction While Throttling May Result in a Processor Hang
BM69	X	X	No Fix	PerfMon Event LOAD_HIT_PRE.SW_PREFETCH May Overcount
BM70	X	X	No Fix	Execution of FXSAVE or FXRSTOR With the VEX Prefix May Produce a #NM Exception
BM71	X	X	No Fix	Unexpected #UD on VPEXTRD/VPINSRD
BM72	-	-	-	Erratum Removed
BM73	X	X	No Fix	Successive Fixed Counter Overflows May be Discarded
BM74	X	X	No Fix	#GP May be Signaled When Invalid VEX Prefix Precedes Conditional Branch Instructions

Table 5. Errata (Sheet 4 of 5)

No.	Steppings		Status	Errata
	D-2	Q-0		
BM75	X	X	No Fix	A Read from the APIC-Timer CCR May Disarm the TSC_Deadline Counter
BM76	X	X	No Fix	An Unexpected PMI May Occur After Writing a Large Value to IA32_FIXED_CTR2
BM77	X	X	No Fix	RDMSR From The APIC-Timer CCR May Disarm The APIC Timer in TSC Deadline Mode
BM78	X	X	No Fix	Not Applicable
BM79	X	X	No Fix	Repeated PCIe and/or DMI L1 Transitions During Package Power States May Cause a System Hang
BM80	X	X	No Fix	Not Applicable
BM81	X	X	No Fix	PCI Express Differential Peak-Peak Tx Voltage Swing May Violate the Specification
BM82	X	X	No Fix	PCIe Presence Detect State May Not be Accurate After a Warm Reset
BM83	X	X	No Fix	Not Applicable
BM84	X	X	No Fix	PCMPESTRI, PCMPESTRM, VPCMPESTRI and VPCMPESTRM Always Operate with 32-bit Length Registers
BM85	X	X	No Fix	VM Entries That Return From SMM Using VMLAUNCH May Not Update The Launch State of the VMCS
BM86	X	X	No Fix	Interrupt From Local APIC Timer May Not Be Detectable While Being Delivered
BM87	X	X	No Fix	An Unexpected Page Fault May Occur Following the Unmapping and Re-mapping of a Page
BM88	X	X	No Fix	A PCIe Device That Initially Transmits Minimal Posted Data Credits May Cause a System Hang
BM89	X	X	No Fix	Some Model Specific Branch Events May Overcount
BM90	X	X	No Fix	Some Performance Monitoring Events in AnyThread Mode May Get Incorrect Count
BM91	X	X	No Fix	PDIR May Not Function Properly With FREEZE_PERFMON_ON_PMI
BM92	X	X	No Fix	For A Single Logical Processor Package, HTT May be Set to Zero Even Though The Package Reserves More Than One APIC ID
BM93	X	X	No Fix	LBR May Contain Incorrect Information When Using FREEZE_LBRS_ON_PMI
BM94	X	X	No Fix	A First Level Data Cache Parity Error May Result in Unexpected Behavior
BM95	X	X	No Fix	Not Applicable
BM96	X	X	No Fix	Programming PDIR And an Additional Precise PerfMon Event May Cause Unexpected PMI or PEBS Events No Erratum
BM97	X	X	No Fix	Performance Monitoring May Overcount Some Events During Debugging
BM98	X	X	No Fix	LTR Message is Not Treated as an Unsupported Request Not Applicable
BM99	X	X	No Fix	Use of VMASKMOV to Access Memory Mapped I/O or Uncached Memory May Cause The Logical Processor to Hang
BM100	X	X	No Fix	PEBS May Unexpectedly Signal a PMI After The PEBS Buffer is Full
BM101	X	X	No Fix	XSAVEOPT May Fail to Save Some State after Transitions Into or Out of STM
BM102	X	X	No Fix	Performance Monitor Precise Instruction Retired Event May Present Wrong Indications
BM103	X	X	No Fix	The Value in IA32_MC3_ADDR MSR May Not be Accurate When MCACOD 0119H is Reported in IA32_MC3_Status
BM104	X	X	No Fix	MSR_PKG_Cx_RESIDENCY MSRs May Not be Accurate
BM105	X	X	No Fix	Enabling/Disabling PEBS May Result in Unpredictable System Behavior
BM106	X	X	No Fix	Execution of VAESIMC or VAESKEYGENASSIST With An Illegal Value for VEX.vvvv May Produce a #NM Exception
BM107	X	X	No Fix	Unexpected #UD on VZEROALL/VZEROUPPER

Table 5. Errata (Sheet 5 of 5)

No.	Steppings		Status	Errata
	D-2	Q-0		
BM108	X	X	No Fix	Successive Fixed Counter Overflows May be Discarded
BM109	X	X	No Fix	Execution of FXSAVE or FXRSTOR With the VEX Prefix May Produce a #NM Exception
BM110	X	X	No Fix	VM Exits Due to "NMI-Window Exiting" May Not Occur Following a VM Entry to the Shutdown State
BM111	X	X	No Fix	Execution of INVVPID Outside 64-Bit Mode Cannot Invalidate Translations For 64-Bit Linear Addresses
BM112	X	X	No Fix	VEX.L is Not Ignored with VCVT*2SI Instructions
BM113	X	X	No Fix	MCI_ADDR May be Incorrect For Cache Parity Errors
BM114	X	X	No Fix	Instruction Fetches Page-Table Walks May be Made Speculatively to Uncacheable Memory
BM115	X	X	No Fix	The Processor May Not Properly Execute Code Modified Using A Floating-Point Store
BM116	X	X	No Fix	Execution of GETSEC[SEXIT] May Cause a Debug Exception to be Lost
BM117	X	X	No Fix	VM Exits Due to GETSEC May Save an Incorrect Value for "Blocking by STI" in the Context of Probe-Mode Redirection
BM118	X	X	No Fix	Not Applicable
BM119	X	X	No Fix	IA32_MC5_CTL2 is Not Cleared by a Warm Reset
BM120	X	X	Plan Fix	Performance Monitor Counters May Produce Incorrect Results
BM121	X	X	No Fix	The Corrected Error Count Overflow Bit in IA32_MCO_STATUS is Not Updated After a UC Error is Logged
BM122	X	X	No Fix	Spurious Intel®VT-d Interrupts May Occur When the PFO Bit is Set
BM123	X	X	No Fix	Processor May Livelock During On Demand Clock Modulation
BM124	X	X	No Fix	The Upper 32 Bits of CR3 May be Incorrectly Used With 32-Bit Paging
BM125	X	X	No Fix	EPT Violations May Report Bits 11:0 of Guest Linear Address Incorrectly
BM126	X	X	No Fix	IA32_VMX_VMCS_ENUM_MSR_(48AH) Does Not Properly Report The Highest Index Value Used For VMCS Encoding
BM127	X	X	No Fix	Not Applicable
BM128	X	X	No Fix	Virtual-APIC Page Accesses With 32-Bit PAE Paging May Cause a System Crash
BM129	X	X	No Fix	SMRAM State-Save Area Above the 4GB Boundary May Cause Unpredictable System Behavior

Table 6. Specification Changes

No.	Specification Changes
1.	DDR3 Signal Group DC Specifications Table Update
2.	Control Sideband and TAP Signal Group DC Specifications

Table 7. Specification Clarifications

No.	Specification Clarifications
1.	DDR3 Signal Group DC Specifications Table Update
2.	System Agent Vcc VID and DC Specifications

Table 8. Document-Only Changes

No.	Document Title	Rev.	Document-Only Changes
1.	*Intel Xeon® and Intel® Core™ Processors for Communications Infrastructure Datasheet* - Volume 1 of 2	1.5	Dual-Channel Asymmetric Mode
2.			CKE Power-Down
3.			On-Demand Clock Modulation Feature Clarification

Identification Information

Component Identification using Programming Interface

The processor stepping can be identified by the following register contents:

Table 9. 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	0010b		00b	0110	1010b	xxxxb

Notes:
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 use 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. Refer to Table 10, "CPU ID Information" for the processor stepping ID number in the CPU ID 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 processor can be identified by the register contents in Table 10, "CPU ID Information".

Table 10. CPU ID Information

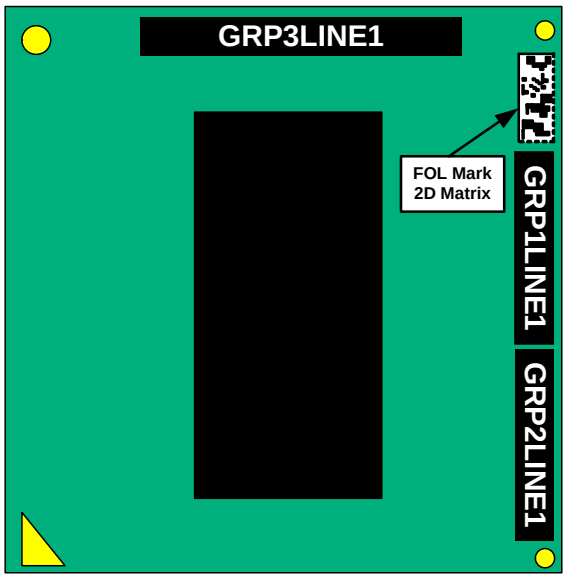
Stepping	Vendor ID[1]	Host Device ID[2]	Revision ID[3]
D-2	8086h	0104h	09h
Q-0	8086h	0104h	09h

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 Host Device ID corresponds to bits 15:0 of the Device ID Register located at Device 0 offset 02–03h
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 and Processor Identification

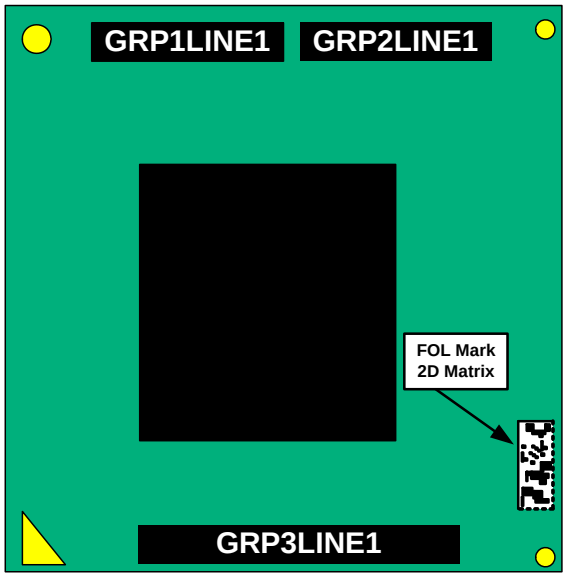
The processor stepping can be identified by the component markings found in Figure 1 and Figure 2.

Figure 1. Intel® Xeon® Processor E3-1125C and Intel® Xeon® Processor E3-1105C Component Markings



Format:
GRP1LINE1: i{M}{C}YY {FPO}
GRP2LINE1: SSPEC
GRP3LINE1: {e1}

Figure 2. Intel® Core™ Processor i3-2115C, Intel® Pentium® Processor B915C, and Intel® Celeron® Processor 725C Component Markings



Format:
GRP1LINE1: i{M}{C}YY {FPO}
GRP2LINE1: SSPEC
GRP3LINE1: {e1}

The Processor Identification table below lists the "S-Spec" component marking numbers of production parts, along with some additional SKU details.

Table 11. Processor Identification

S-SpecNumber	Processor Number	Stepping	Sample Type[6]	CPU ID	Core Freq (GHz)	DDR3 Freq. (MHz)	Active Cores	L3 Cache Size (MB)	Notes
SR0NU	Xeon E3-1125C	D-2	Prod	000206A7h	2.0	1600	4	8	2, 3, 4, 5
SR0NS	Xeon E3-1105C	D-2	Prod	000206A7h	1.0	1600	4	6	2, 3, 4, 5
SR0NV	Core i3-2115C	Q-0	Prod	000206A7h	2.0	1333	2	3	2, 3, 4, 5
SR0NZ	Pentium B915C	Q-0	Prod	000206A7h	1.5	1333	2	3	2, 3, 4, 5
SR0NX	Celeron 725C	Q-0	Prod	000206A7h	1.3	1333	1	1.5	2, 3, 4, 5

1. These are Engineering Samples only. Engineering Samples are not recommended for signal integrity, timing analysis, ESD testing, reliability testing including temperature or power cycling, testing at temperatures exceeding the specifications, or performance testing.
2. Intel® Hyper-Threading Technology enabled.
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. Intel® AES-NI enabled.
6. The *Sample Type* column uses the following abbriviations: ES1 = Engineering Sample 1; ES2 = Engineering Sample 2; QS = Qualification Sample; Prod = Production Parts

Errata

BM1 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 Tables of Changes.

BM2 APIC Error “Received Illegal Vector” May be 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 Tables of Changes.

BM3 An Uncorrectable Error Logged in IA32_CR_MC2_STATUS May also Result in a 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 Tables of Changes.

BM4 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.

Due to this erratum 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 Tables of Changes.

BM5 Changing the Memory Type for an In-Use Page Translation May Lead 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 Tables of Changes.

BM6 Code Segment Limit/Canonical Faults on RSM May be Serviced before Higher Priority Interrupts/Exceptions and May Push the Wrong Address Onto the 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. Due to this erratum, 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 Tables of Changes.

BM7 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 Tables of Changes.

BM8 Debug Exception Flags DR6.B0-B3 Flags May be 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 Tables of Changes.

BM9 DR6.B0-B3 May Not Report All Breakpoints Matched When a MOV/POP SS is Followed by a Store or an MMX Instruction

Problem: Normally, data breakpoints matches that occur on a MOV SS, r/m or POP SS will not cause a debug exception immediately after MOV/POP SS but will be delayed until the instruction boundary following the next instruction is reached. After the debug exception occurs, DR6.B0-B3 bits will contain information about data breakpoints matched during the MOV/POP SS as well as breakpoints detected by the following instruction. Due to this erratum, DR6.B0-B3 bits may not contain information about data breakpoints matched during the MOV/POP SS when the following instruction is either an MMX instruction that uses a memory addressing mode with an index or a store instruction.

Implication: When this erratum occurs, DR6 may not contain information about all breakpoints matched. This erratum will not be observed under the recommended usage of the MOV SS,r/m or POP SS instructions (i.e., following them only with an instruction that writes (E/R)SP).

Workaround: None identified.

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

BM10 EFLAGS Discrepancy on Page Faults and on EPT-Induced VM Exits after a 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 Tables of Changes.

BM11 Fault on ENTER Instruction May Result 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 the *IA-32 Intel® Architecture Software Developer's Manual*, Volume 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 Tables of Changes.

BM12 Faulting MMX Instruction May Incorrectly Update 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 Tables of Changes.

BM13 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 and 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 Tables of Changes.

BM14 General Protection Fault (#GP) for Instructions Greater than 15 Bytes May be 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 Tables of Changes.

BM15 #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 Tables of Changes.

BM16 IO_SMI Indication in SMRAM State Save Area May be Set Incorrectly

Problem: The IO_SMI bit in SMRAM's location 7FA4H is set to “1” by the CPU to indicate a System Management Interrupt (SMI) occurred as the result of executing an instruction that reads from an I/O port. Due to this erratum, the IO_SMI bit may be incorrectly set by:

- A non-I/O instruction
- SMI is pending while a lower priority event interrupts
- A REP I/O read
- A I/O read that redirects to MWAIT

Implication: SMM handlers may get false IO_SMI indication.

Workaround: The SMM handler has to evaluate the saved context to determine if the SMI was triggered by an instruction that read from an I/O port. The SMM handler must not restart an I/O instruction if the platform has not been configured to generate a synchronous SMI for the recorded I/O port address.

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

BM17 IRET under Certain Conditions May Cause an 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 Tables of Changes.

BM18 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 Tables of Changes.

BM19 LBR, BTS, BTM May Report a Wrong Address when an 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 Tables of Changes.

BM20 MCi_Status Overflow Bit May Be Incorrectly Set on a Single Instance of a 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 Tables of Changes.

BM21 MONITOR or CLFLUSH on the Local XAPIC's 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 Tables of Changes.

BM22 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 Tables of Changes.

BM23 PEBS Record not Updated when in Probe Mode

Problem: When a performance monitoring counter is configured for PEBS (Precise Event Based Sampling), overflows of the counter can result in storage of a PEBS record in the PEBS buffer. Due to this erratum, if the overflow occurs during probe mode, it may be ignored and a new PEBS record may not be added to the PEBS buffer.

Implication: Due to this erratum, the PEBS buffer may not be updated by overflows that occur during probe mode.

Workaround: None identified.

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

BM24 Performance Monitoring Event FP_MMX_TRANS_TO_MMX May Not Count 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 Tables of Changes.

BM25 REP MOVS/STOS Executing with Fast Strings Enabled and Crossing Page Boundaries with Inconsistent Memory Types may use an 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 Intel® Pentium® 4 processor, Intel® Xeon® processor, and P6 Family Processors" the processor performs REP MOVS or REP STOS as fast strings. Due to this erratum 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 Tables of Changes.

BM26 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 Tables of Changes.

BM27 Single Step Interrupts with Floating Point Exception Pending May Be 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 Tables of Changes.

BM28 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 Tables of Changes.

BM29 The Processor May Report a #TS Instead of a #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 Tables of Changes.

BM30 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 Tables of Changes.

BM31 Pending x87 FPU Exceptions (#MF) May be 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 Tables of Changes.

BM32 Values for LBR/BTS/BTM Will be Incorrect after an 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 Tables of Changes.

BM33 Unsupported PCIe* Upstream Access May Complete with an Incorrect Byte Count

Problem: PCIe* Upstream IO and Configuration accesses are not supported. If an IO or Configuration request is received upstream, the integrated PCIe* controller will treat it as an unsupported request, the request will be dropped, and a completion will be sent with the UR (Unsupported Request) completion status. This completion, according to the PCIe* specification, should indicate a byte count of 4. Due to this erratum, the byte count is set to the same byte count as the offending request.

Implication: The processor response to an unsupported PCIe* access may not fully comply to the PCIe* specification.

Workaround: PCIe* agents should not issue unsupported accesses.

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

BM34 Malformed PCIe* Transactions May be Treated as Unsupported Requests Instead of as Critical Errors

Problem: PCIe* MSG/MSG_D TLPs (Transaction Layer Packets) with incorrect Routing Code as well as the deprecated TCfgRD and TCfgWr types should be treated as malformed transactions leading to a critical error. Due to this erratum, the integrated PCIe* controller's root ports may treat such messages as UR (Unsupported Requests).

Implication: Legacy malformed PCIe* transactions may be treated as UR instead of as critical errors.

Workaround: None identified.

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

BM35 PCIe* Root Port May Not Initiate Link Speed Change

Problem: PCIe* specification rev 2.0 requires the upstream component to maintain the PCIe* link at the target link speed or the highest speed supported by both components on the link, whichever is lower. PCIe* root port will not initiate the link speed change without being triggered by the software. System BIOS will trigger the link speed change under normal boot scenarios. However, BIOS is not involved in some scenarios such as link disable/re-enable or secondary bus reset and therefore the speed change may not occur unless initiated by the downstream component. This erratum does not affect the ability of the downstream component to initiate a link speed change. All known 5.0Gb/s-capable PCIe* downstream components have been observed to initiate the link speed change without relying on the root port to do so.

Implication: Due to this erratum, the PCIe* root port may not initiate a link speed change during some hardware scenarios causing the PCIe* link to operate at a lower than expected speed. Intel has not observed this erratum with any commercially available platform.

Workaround: None identified.

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

BM36 Incorrect Address Computed For Last Byte of FXSAVE/FXRSTOR or XSAVE/XRSTOR Image Leads to Partial Memory Update

Problem: A partial memory state save of the FXSAVE or XSAVE image or a partial memory state restore of the FXRSTOR or XRSTOR 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 or XSAVE/XRSTOR 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 Tables of Changes.

BM37 Performance Monitor SSE Retired Instructions May Return Incorrect Values

Problem: Performance Monitoring counter SIMD_INST_RETIRED (Event: C7H) is used to track retired SSE instructions. Due to this erratum, 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 Tables of Changes.

BM38 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 Tables of Changes.

BM39 FP Data Operand Pointer May Be Incorrectly Calculated After an FP Access Which Wraps a 64-KByte Boundary in 16-Bit Code

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) occurs in a 16-bit mode other than protected mode (in which case the access will produce a segment limit violation), the memory access wraps a 64-KByte 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 segment 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 an operating system which may run 16-bit FP code, care must be taken to ensure that no 80-bit FP accesses are wrapped around a 64-KByte boundary.

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

BM40 Spurious Interrupts May be Generated From the Intel® VT-d Remap Engine

Problem: If software clears the F (Fault) bit 127 of the Fault Recording Register (FRCD_REG at offset 0x208 in Remap Engine BAR) by writing 1b through RW1C command (Read Write 1 to Clear) when the F bit is already clear then a spurious interrupt from Intel VT-d (Virtualization Technology for Directed I/O) Remap Engine may be observed.

Implication: Due to this erratum, spurious interrupts will occur from the Intel VT-d Remap Engine following RW1C clearing F bit.

Workaround: None identified.

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

BM41 Fault Not Reported When Setting Reserved Bits of Intel® VT-d Queued Invalidation Descriptors

Problem: Reserved bits in the Queued Invalidation descriptors of Intel VT-d (Virtualization Technology for Directed I/O) are expected to be zero, meaning that software must program them as zero while the processor checks if they are not zero. Upon detection of a non-zero bit in a reserved field an Intel VT-d fault should be recorded. Due to this erratum the processor does not check reserved bit values for Queued Invalidation descriptors.

Implication: Due to this erratum, faults will not be reported when writing to reserved bits of Intel VT-d Queued Invalidation Descriptors.

Workaround: None identified.

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

BM42 VPHMINPOSUW Instruction in Vex Format Does Not Signal #UD When vex.vvvv !=1111b

Problem: Processor does not signal #UD fault when executing the reserved instruction VPHMINPOSUW with vex.vvvv !=1111b.

Implication: Executing VPHMINPOSUW with vex.vvvv !=1111b results in the same behavior as executing with vex.vvvv=1111b.

Workaround: Software should not use VPHMINPOSUW with vex.vvvv !=1111b in order to ensure future compatibility.

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

BM43 LBR, BTM or BTS Records May have Incorrect Branch From Information After an EIST/T-state/S-state/C1E Transition 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 a transition of:

- EIST (Enhanced Intel® SpeedStep Technology)
- T-state (Thermal Monitor states)
- S1-state (ACPI package sleep state)
- C1E (Enhanced C1 Low Power state)
- 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 a transition of EIST, T-states, S-states, C1E, or Adaptive Thermal Throttling.

Workaround: None identified.

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

BM44 VMREAD/VMWRITE Instruction May Not Fail When Accessing an Unsupported Field in VMCS

Problem: The Intel® 64 and IA-32 Architectures Software Developer's Manual, Volume 2B states that execution of VMREAD or VMWRITE should fail if the value of the instruction's

register source operand corresponds to an unsupported field in the VMCS (Virtual Machine Control Structure). The correct operation is that the logical processor will set the ZF (Zero Flag), write 0CH into the VM-instruction error field and for VMREAD leave the instruction's destination operand unmodified. Due to this erratum, the instruction may instead clear the ZF, leave the VM-instruction error field unmodified and for VMREAD modify the contents of its destination operand.

Implication: Accessing an unsupported field in VMCS will fail to properly report an error. In addition, VMREAD from an unsupported VMCS field may unexpectedly change its destination operand. Intel has not observed this erratum with any commercially available software.

Workaround: Software should avoid accessing unsupported fields in a VMCS

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

BM45 Clock Modulation Duty Cycle Cannot be Programmed to 6.25%

Problem: When programming field T_STATE_REQ of the IA32_CLOCK_MODULATION MSR (19AH) bits [3:0] to '0001, the actual clock modulation duty cycle will be 12.5% instead of the expected 6.25% ratio.

Implication: Due to this erratum, it is not possible to program the clock modulation to a 6.25% duty cycle.

Workaround: None identified.

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

BM46 Execution of VAESIMC or VAESKEYGENASSIST With An Illegal Value for VEX.vvvv May Produce a #NM Exception

Problem: The VAESIMC and VAESKEYGENASSIST instructions should produce a #UD (Invalid-Opcode) exception if the value of the vvvv field in the VEX prefix is not 1111b. Due to this erratum, if CR0.TS is “1”, the processor may instead produce a #NM (Device-Not-Available) exception.

Implication: Due to this erratum, some undefined instruction encodings may produce a #NM instead of a #UD exception.

Workaround: Software should always set the vvvv field of the VEX prefix to 1111b for instances of the VAESIMC and VAESKEYGENASSIST instructions

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

BM47 Memory Aliasing of Code Pages May Cause Unpredictable System 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 uncacheable 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 Tables of Changes.

BM48 PCI Express Receiver Error Reported When Receiver With L0s Enabled and Link Retrain Performed

Problem: If the Processor PCI Express root port is the receiver with L0s enabled and the root port itself initiates a transition to the recovery state via the retrain link configuration bit in the 'Link Control' register (Bus 0; Device 1; Functions 0, 1, 2 and Device 6; Function 0; Offset B0H; bit 5), then the root port may not mask the receiver or bad DLLP (Data Link Layer Packet) errors as expected. These correctable errors should only be considered valid during PCIe* configuration and L0 but not L0s. This causes the processor to falsely report correctable errors in the 'Device Status' register (Bus 0; Device 1; Functions 0, 1, 2 and Device 6; Function 0; Offset AAH; bit 0) upon receiving the first FTS (Fast Training Sequence) when exiting Receiver L0s. Under normal conditions there is no reason for the Root Port to initiate a transition to Recovery. Note: This issue is only exposed when a recovery event is initiated by the processor.

Implication: The processor does not comply with the PCI Express 2.0 Specification. This does not impact functional compatibility or interoperability with other PCIe* devices.

Workaround: None identified.

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

BM49 Unexpected #UD on VZEROALL/VZEROUPPER

Problem: Execution of the VZEROALL or VZEROUPPER instructions in 64-bit mode with VEX.W set to 1 may erroneously cause a #UD (invalid-opcode exception).

Implication: The affected instructions may produce unexpected invalid-opcode exceptions in 64-bit mode.

Workaround: Compilers should encode VEX.W = 0 for executions of the VZEROALL and VZEROUPPER instructions in 64-bit mode to ensure future compatibility.

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

BM50 Perfmon Event LD_BLOCKS.STORE_FORWARD May Overcount

Problem: Perfmon LD_BLOCKS.STORE_FORWARD (event 3H, umask 01H) may overcount in the cases of 4KB address aliasing and in some cases of blocked 32-byte AVX load operations. 4KB address aliasing happens when unrelated load and store that have different physical addresses appear to overlap due to partial address check done on the lower 12 bits of the address. In some cases such memory aliasing can cause load execution to be significantly delayed. Blocked AVX load operations refer to 32-byte AVX loads that are blocked due to address conflict with an older store.

Implication: The perfmon event LD_BLOCKS.STORE_FORWARD may overcount for these cases.

Workaround: None identified.

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

BM51 Not Applicable

BM52 Execution of Opcode 9BH with the VEX Opcode Extension May Produce a #NM Exception

Problem: Attempt to use opcode 9BH with a VEX opcode extension should produce a #UD (Invalid-Opcode) exception. Due to this erratum, if CR0.MP and CR0.TS are both 1, the processor may produce a #NM (Device-Not-Available) exception if one of the following conditions exists:

- 66H, F2H, F3H or REX as a preceding prefix;
- An illegal map specified in the VEX.mmmmm field;

Implication: Due to this erratum, some undefined instruction encodings may produce a #NM instead of a #UD exception.

Workaround: Software should not use opcode 9BH with the VEX opcode extension.

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

BM53 Executing The GETSEC Instruction While Throttling May Result in a Processor Hang

Problem: If the processor throttles due to either high temperature thermal conditions or due to an explicit operating system throttling request (TT1) while executing GETSEC[SENTER] or GETSEC[SEXIT] instructions, then under certain circumstances, the processor may hang. Intel has not been observed this erratum with any commercially available software.

Implication: Possible hang during execution of GETSEC instruction.

Workaround: None identified.

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

BM54 A Write to the IA32_FIXED_CTR1 MSR May Result in Incorrect Value in Certain Conditions

Problem: Under specific internal conditions, if software tries to write the IA32_FIXED_CTR1 MSR (30AH) a value that has all bits [31:1] set while the counter was just about to overflow when the write is attempted (i.e. its value was 0xFFFF FFFF FFFF), then due to this erratum the new value in the MSR may be corrupted.

Implication: Due to this erratum, IA32_FIXED_CTR1 MSR may be written with a corrupted value.

Workaround: Software may avoid this erratum by writing zeros to the IA32_FIXED_CTR1 MSR, before the desired write operation.

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

BM55 Instruction Fetch May Cause Machine Check if Page Size and Memory Type Was Changed Without Invalidation

Problem: This erratum may cause a machine-check error (IA32_MCi_STATUS.MCACOD=0150H) on the fetch of an instruction that crosses a 4-KByte address boundary. It applies only if (1) the 4-KByte linear region on which the instruction begins is originally translated using a 4-KByte page with the WB memory type; (2) the paging structures are later modified so that linear region is translated using a large page (2-MByte, 4-MByte, or 1-GByte) with the UC memory type; and (3) the instruction fetch occurs after the paging-structure modification but before software invalidates any TLB entries for the linear region.

Implication: Due to this erratum an unexpected machine check with error code 0150H may occur, possibly resulting in a shutdown. Intel has not observed this erratum with any commercially available software.

Workaround: Software should not write to a paging-structure entry in a way that would change, for any linear address, both the page size and the memory type. It can instead use the following algorithm: first clear the P flag in the relevant paging-structure entry (e.g., PDE); then invalidate any translations for the affected linear addresses; and then modify the relevant paging-structure entry to set the P flag and establish the new page size and memory type.

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

BM56 Reception of Certain Malformed Transactions May Cause PCIe* Port to Hang Rather Than Reporting an Error

Problem: If the processor receives an upstream malformed non posted packet for which the type field is IO, Configuration or the deprecated TCfgRd and the format is 4 DW header, then

due to this erratum the integrated PCIe* controller may hang instead of reporting the malformed packet error or issuing an unsupported request completion transaction.

Implication: Due to this erratum, the processor may hang without reporting errors when receiving a malformed PCIe* transaction. Intel has not observed this erratum with any commercially available device.

Workaround: None identified. Upstream transaction initiators should avoid issuing unsupported requests with 4 DW header formats.

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

BM57 PCIe* LTR Incorrectly Reported as Being Supported

Problem: LTR (Latency Tolerance Reporting) is a new optional feature specified in PCIe* rev. 2.1. The processor reports LTR as supported in LTRS bit in DCAP2 register (bus 0; Device 1; Function 0; offset 0xc4), but this feature is not supported.

Implication: Due to this erratum, LTR is always reported as supported by the LTRS bit in the DCAP2 register.

Workaround: None identified.

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

BM58 Performance-Counter Overflow Indication May Cause Undesired Behavior

Problem: Under certain conditions (listed below) when a performance counter overflows, its overflow indication may remain set indefinitely. This erratum affects the general-purpose performance counters IA32_PMC{0-7} and the fixed-function performance counters IA32_FIXED_CTR{0-2}. The erratum may occur if any of the following conditions are applied concurrent to when an actual counter overflow condition is reached:

1. Software disables the counter either globally through the IA32_PERF_GLOBAL_CTRL MSR (38FH), or locally through the IA32_PERFEVTSEL{0-7} MSRs (186H-18DH), or the IA32_FIXED_CTR_CTRL MSR (38DH).
2. Software sets the IA32_DEBUGCTL MSR (1D9H) FREEZE_PERFMON_ON_PMI bit [12].
3. The processor attempts to disable the counters by updating the state of the IA32_PERF_GLOBAL_CTRL MSR (38FH) as part of transitions such as VM exit, VM entry, SMI, RSM, or processor C-state.

Implication: Due to this erratum, the corresponding overflow status bit in IA32_PERF_GLOBAL_STATUS MSR (38DH) for an affected counter may not get cleared when expected. If a corresponding counter is configured to issue a PMI (performance monitor interrupt), multiple PMIs may be signaled from the same overflow condition. Likewise, if a corresponding counter is configured in PEBS mode (applies to only the general purpose counters), multiple PEBS events may be signaled.

Workaround: None identified.

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

BM59 XSAVE Executed During Paging-Structure Modification May Cause Unexpected Processor Behavior

Problem: Execution of XSAVE may result in unexpected behavior if the XSAVE instruction writes to a page while another logical processor clears the dirty flag or the accessed flag in any paging-structure entry that maps that page.

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

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

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

BM60 C-state Exit Latencies May be Higher Than Expected

Problem: Core C-state exit can be delayed if a P-state transition is requested before the pending C-state exit request is completed. Under certain internal conditions the core C-state exit latencies may be over twice the value specified in the Intel® 64 and IA-32 Architectures Optimization Reference Manual.

Implication: While typical exit latencies are not impacted, the worst case core C-state exit latency may be over twice the value specified in the Intel® 64 and IA-32 Architectures Optimization Reference Manual and may lead to a delay in servicing interrupts. Intel has not observed any system failures due to this erratum.

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

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

BM61 Intel® VT-d Interrupt Remapping Will Not Report a Fault if Interrupt Index Exceeds FFFFH

Problem: With Intel® VT-d (Virtualization Technology for Directed I/O) interrupt remapping, if subhandle valid (bit 3) is set in the address of an interrupt request, the interrupt index is computed as the sum of the interrupt request's handle and subhandle. If the sum is greater than FFFFH (the maximum possible interrupt-remapping table size) a remapping fault with fault reason 21H should be reported. Due to this erratum, this condition is not reported as a fault; instead, the low 16 bits of the sum are erroneously used as an interrupt index to access the interrupt-remapping table.

Implication: If the interrupt index of an interrupt request exceeds FFFFH, a remapping fault with fault reason 21H is not reported and instead the request uses the IRTE (interrupt-remapping table entry) indexed by the low 16 bits of the interrupt index.

Workaround: Software can use requestor-id verification to block the interrupts that would be delivered due to this erratum. Interrupts blocked in this way produce a remapping fault with fault reason 26H.

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

BM62 PCIe* Link Speed May Not Change From 5.0 GT/s to 2.5 GT/s

Problem: If a PCI Express device changes its supported PCIe* link speed from 5.0 GT/s to 2.5 GT/s without initiating a speed change request and subsequently the L1 power management mode is entered, further retrains initiated by software will not change speed to 2.5 GT/s.

Implication: Intel has not observed any PCI Express device that changes supported link speed without actually initiating a speed change.

Workaround: None identified.

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

BM63 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 Tables of Changes.

BM64 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 3B 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 Tables of Changes.

BM65 TSC Deadline Not Armed While in APIC Legacy Mode

Problem: Under specific timing conditions, when in Legacy APIC Mode, writing to IA32_TSC_DEADLINE MSR (6E0H) may fail to arm the TSC Deadline (Time Stamp Counter Deadline) event as expected. Exposure to this erratum is dependent on the proximity of TSC_Deadline MSR Write to a Timer CCR register read or to a write to the Timer LVT that enabled the TSC Deadline mode (writing 10 to bits [18:17] of Timer LVT).

Implication: Due to this erratum the expected timer event will either not be generated or will be generated at a wrong time. The TSC Deadline may fail until an LVT write to transition from “TSC Deadline mode” back to “Timer mode” occurs or until the next reset.

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

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

BM66 PCIe* Upstream TCfgWr May Cause Unpredictable System Behavior

Problem: TCfgWr (Trusted Configuration Writes) is a PCIe* Base spec deprecated transaction type which should be treated as a malformed packet. If a PCIe* upstream TCfgWr request is received, then due to this erratum the request may not be managed as a Malformed Packet.

Implication: Upstream memory writes subsequent to a TCfgWr transaction may cause unpredictable system behavior. Intel has not observed any PCIe* Device that sends such a TCfgWr request.

Workaround: PCIe* end points should not initiate upstream TCfgWr requests.

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

BM67 Processor May Fail to Acknowledge a TLP Request

Problem: When a PCIe* root port's receiver is in Receiver L0s power state and the port initiates a Recovery event, it will issue Training Sets to the link partner. The link partner will respond by initiating an L0s exit sequence. Prior to transmitting its own Training Sets, the link partner may transmit a TLP (Transaction Layer Packet). Due to this erratum, the root port may not acknowledge the TLP request.

Implication: After completing the Recovery event, the PCIe* link partner will replay the TLP request. The link partner may set a Correctable Error status bit, which has no functional effect.

Workaround: None identified.

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

BM68 Executing The GETSEC Instruction While Throttling May Result in a Processor Hang

Problem: If the processor throttles due to either high temperature thermal conditions or due to an explicit operating system throttling request (TT1) while executing GETSEC[SENTER] or GETSEC[SEXIT] instructions, then under certain circumstances, the processor may hang.

Implication: Possible hang during execution of GETSEC instruction. Intel has not been observed this erratum with any commercially available software.

Workaround: None Identified.

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

BM69 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 Tables of Changes.

BM70 Execution of FXSAVE or FXRSTOR With the VEX Prefix May Produce a #NM Exception

Problem: Attempt to use FXSAVE or FXRSTOR with a VEX prefix should produce a #UD (Invalid-Opcode) exception. If either the TS or EM flag bits in CR0 are set, a #NM (device-not-available) exception will be raised instead of #UD exception.

Implication: Due to this erratum a #NM exception may be signaled instead of a #UD exception on an FXSAVE or an FXRSTOR with a VEX prefix.

Workaround: Software should not use FXSAVE or FXRSTOR with the VEX prefix.

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

BM71 Unexpected #UD on VPEXTRD/VPINSRD

Problem: Execution of the VPEXTRD or VPINSRD instructions outside of 64-bit mode with VEX.W set to 1 may erroneously cause a #UD (invalid-opcode exception).

Implication: The affected instructions may produce unexpected invalid-opcode exceptions outside 64-bit mode.

Workaround: Software should encode VEX.W = 0 for executions of the VPEXTRD and VPINSRD instructions outside 64-bit mode.

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

BM72 Erratum Removed

BM73 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:

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

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

BM74 #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 Tables of Changes.

BM75 A Read from the APIC-Timer CCR May Disarm the TSC_Deadline Counter

Problem: When in TSC Deadline mode with TSC_Deadline timer armed (IA32_TSC_DEADLINE<>0, MSR 6E0H), a read from the local APIC's CCR (current count register) using RDMSR 0839H may disarm the TSC Deadline timer without generating an interrupt as specified in the APIC Timer LVT (Local Vector Table) entry.

Implication: Due to this erratum, unexpected disarming of the TSC_Deadline counter and possible loss of an interrupt may occur.

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

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

BM76 An Unexpected PMI May Occur After Writing a Large Value to IA32_FIXED_CTR2

Problem: If the fixed-function performance counter IA32_FIXED_CTR2 MSR (30BH) is configured to generate a performance-monitor interrupt (PMI) on overflow and the counter's value is greater than FFFFFFFFFFC0H, then this erratum may incorrectly cause a PMI if software performs a write to this counter.

Implication: A PMI may be generated unexpectedly when programming IA32_FIXED_CTR2. Other than the PMI, the counter programming is not affected by this erratum as the attempted write operation does succeed.

Workaround: None identified.

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

BM77 RDMSR From The APIC-Timer CCR May Disarm The APIC Timer in TSC Deadline Mode

Problem: When in TSC Deadline mode with TSC_Deadline timer armed (IA32_TSC_DEADLINE<>0, MSR 6E0H), a read from the local APIC's CCR (current count register) in APIC MMIO space may disarm the TSC Deadline timer without generating an interrupt as specified in the APIC Timer LVT (Local Vector Table) entry.

Implication: Due to this erratum, unexpected disarming of the APIC timer and possible loss of an interrupt may occur.

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

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

BM78 Not Applicable

BM79 Repeated PCIe and/or DMI L1 Transitions During Package Power States May Cause a System Hang

Problem: Under a complex set of internal conditions when the processor is in a deep power state (package C3, C6 or C7) and the PCIe and/or DMI links are toggling in and out of L1 state, internal states of the processor may become inaccessible resulting in a system hang.

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

Workaround: A BIOS workaround has been identified. Please refer to BIOS Technical Advisory (CDI/IBL#: 468267) for details.

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

BM80 Not Applicable

BM81 PCI Express Differential Peak-Peak Tx Voltage Swing May Violate the Specification

Problem: Under certain conditions, including extreme voltage and temperature, the peak-peak voltage may be higher than the specification.

Implication: Violation of PCI Express Base Specification of the VTX-DIFF-PP voltage. No failures have been observed due to this erratum.

Workaround: None identified. Customers following the Sugar Bay and Bromolow-WS Platform Design Guide (PDG) Rev 2.0 will not observe this erratum at the link partner receiver.

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

BM82 PCIe Presence Detect State May Not be Accurate After a Warm Reset

Problem: Under certain conditions, when there is no PCIe device present, the status of Presence Detect State bit (SLOTSTS Device 1; Function 0,1,2; Offset BAH; bit [6] and/or Device 6; Function 0; Offset BAH; bit [6]) may not be accurate after a warm reset.

Implication: The Presence Detect State bit may incorrectly report a PCIe device is present even though no device is actually present, which may result in a system hang.

Workaround: A BIOS workaround has been identified. Please refer to the Sandy Bridge System Agent BIOS Specification and the System Agent Framework Reference Code release 1.2.0.

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

BM83 Not Applicable

BM84 PCMPESTRI, PCMPESTRM, VPCMPESTRI and VPCMPESTRM Always Operate with 32-bit Length Registers

Problem: In 64-bit mode, using REX.W=1 with PCMPESTRI and PCMPESTRM or VEX.W=1 with VPCMPESTRI and VPCMPESTRM should support a 64-bit length operation with RAX/RDX. Due to this erratum, the length registers are incorrectly interpreted as 32-bit values.

Implication: Due to this erratum, using REX.W=1 with PCMPESTRI and PCMPESTRM as well as VEX.W=1 with VPCMPESTRI and VPCMPESTRM do not result in promotion to 64-bit length registers.

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

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

BM85 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: It is possible for the BIOS to contain a workaround for this erratum.

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

BM86 Interrupt From Local APIC Timer May Not Be Detectable While Being Delivered

Problem: If the local-APIC timer’s CCR (current-count register) is 0, software should be able to determine whether a previously generated timer interrupt is being delivered by first reading the delivery-status bit in the LVT timer register and then reading the bit in the IRR (interrupt-request register) corresponding to the vector in the LVT timer register. If both values are read as 0, no timer interrupt should be in the process of being delivered. Due to this erratum, a timer interrupt may be delivered even if the CCR is 0 and the LVT and IRR bits are read as 0. This can occur only if the DCR (Divide Configuration Register) is greater than or equal to 4. The erratum does not occur if software writes zero to the Initial Count Register before reading the LVT and IRR bits.

Implication: Software that relies on reads of the LVT and IRR bits to determine whether a timer interrupt is being delivered may not operate properly.

Workaround: Software that uses the local-APIC timer must be prepared to handle the timer interrupts, even those that would not be expected based on reading CCR and the LVT and IRR bits; alternatively, software can avoid the problem by writing zero to the Initial Count Register before reading the LVT and IRR bits.

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

BM87 An Unexpected Page Fault May Occur Following the Unmapping and Re-mapping of a Page

Problem: 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).
- Software later 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 subsequent instruction loads from a linear address on the page.
- Software did not invalidate TLB entries for the page between the first modification of the paging structures and the load from the linear address.

In this case, the load by the later instruction 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.

Workaround: None identified.

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

BM88 A PCIe Device That Initially Transmits Minimal Posted Data Credits May Cause a System Hang

Problem: Under certain conditions, if a PCIe device that initially transmits posted data credits less than Max_Payload_Size/16 + 4 (16B/4DW is unit of data flow control) and is the target of a Peer-to-Peer write of Max_Payload_Size, the system may hang due to Posted Data credit starvation.

Implication: Under certain conditions, the processor may encounter a Posted Data credit starvation scenario and hang.

Workaround: A BIOS code change has been identified and may be implemented as a workaround for this erratum. Please refer to the Sandy Bridge System Agent BIOS Specification and the System Agent Framework Reference Code release 1.2.1.

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

BM89 Some Model Specific Branch Events May Overcount

Problem: Under certain internal conditions the following model specific performance monitoring branch events may overcount:

- BR_INST_RETIRED.NOT_TAKEN
- BR_INST_RETIRED.NEAR_TAKEN
- BR_MISP_RETIRED.NOT_TAKEN
- BR_MISP_RETIRED.TAKEN

Implication: Due to this erratum the events may overcount.

Workaround: None identified.

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

BM90 Some Performance Monitoring Events in AnyThread Mode May Get Incorrect Count

Problem: Performance monitoring AnyThread mode allows a given thread to monitor events as a result of any thread running on the same core. Due to this erratum, on systems with SMT enabled, counting any of the following performance monitoring events in AnyThread mode may get incorrect values:

- INST_RETIRED;

- OTHER_ASSISTS;
- UOPS_RETIRED;
- MACHINE_CLEARS;
- BR_INST_RETIRED;
- BR_MISP_RETIRED;
- SIMD_INST_RETIRED;
- FP_ASSIST;
- HW_INTERRUPTS;
- ROB_MISC_EVENTS;
- MEM_LOAD_RETIRED;
- MEM_LOAD_LLC_HIT_RETIRED;
- MEM_LOAD_LLC_MISS_RETIRED;
- MEM_LOAD_MISC_RETIRED;

Implication: Incorrect results when counting the above performance monitoring events in AnyThread mode with SMT on.

Workaround: In order to get a correct count for the above events, software may count the same event on both threads of the same physical core, and at post-processing stage sum-up the two values to get the core's net value.

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

BM91 PDIR May Not Function Properly With FREEZE_PERFMON_ON_PMI

Problem: When the PDIR (Precise Distribution for Instructions Retired) mechanism is activated (INST_RETIRED.ALL (event C0H, umask value 00H) on Counter 1 programmed in PEBS mode) along with FREEZE_PERFMON_ON_PMI, bit 11, in the IA32_DEBUGCTL MSR (1D9h), the processor may behave in an undefined manner.

Implication: Due to this erratum when FREEZE_PERFMON_ON_PMI is programmed along with PDIR the processor behavior is undefined. This can result in any of but not limited to the following: incorrect PMI interrupts, incorrect PEBS events or invalid processor state.

Workaround: A software driver should not program FreezeOnPMI in conjunction with the PDIR mechanism.

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

BM92 For A Single Logical Processor Package, HTT May be Set to Zero Even Though The Package Reserves More Than One APIC ID

Problem: When maximum number of addressable IDs for logical processors in this physical package (CPUID.01H.EBX[23:16]) and maximum number of addressable IDs for processor cores in the physical package, (CPUID.04H.EAX[31:26]) indicate more than one reserved APIC ID, HTT (Multi-Threading, CPUID.01H.EDX[28]) should be set to One. However due to this erratum it may be set to Zero.

Implication: Software written expecting HTT to be Zero only when a single APIC ID is reserved for the package 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 Tables of Changes.

BM93 LBR May Contain Incorrect Information When Using FREEZE_LBRS_ON_PMI

Problem: When FREEZE_LBRS_ON_PMI is enabled (bit 11 of IA32_DEBUGCTL MSR (1D9H) is set), and a taken branch retires at the same time that a PMI (Performance Monitor

Interrupt) occurs, then under certain internal conditions the record at the top of the LBR stack may contain an incorrect “From” address.

Implication: When the LBRs are enabled with FREEZE_LRBS_ON_PMI, the “From” address at the top of the LBR stack may be incorrect.

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

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

BM94 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: Due to this erratum, 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 Tables of Changes.

BM95 Not Applicable

BM96 Programming PDIR And an Additional Precise PerfMon Event May Cause Unexpected PMI or PEBS Events No Erratum

Problem: PDIR (Precise Distribution for Instructions Retired) mechanism is activated by programming INST_RETIRED.ALL (event C0H, umask value 00H) on Counter 1. When PDIR is activated in PEBS (Precise Event Based Sampling) mode with an additional precise PerfMon event, an incorrect PMI or PEBS event may occur.

Implication: Due to this erratum, when another PEBS event is programmed along with PDIR, an incorrect PMI or PEBS event may occur.

Workaround: Software should not program another PEBS event in conjunction with the PDIR mechanism.

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

BM97 Performance Monitoring May Overcount Some Events During Debugging

Problem: If the debug-control register (DR7) is configured so that some but not all of the breakpoints in the debug-address registers (DR0-DR3) are enabled and one or more of the following performance-monitoring counters are locally enabled (via IA32_CR_PERMON_EVNTSEL_CNTR{3:0}):

BR_INST_RETIRED
BR_MISP_RETIRED
FP_ASSIST
INST_RETIRED
MACHINE_CLEARS
MEM_LOAD_UOPS_LLC_HIT_RETIRED
MEM_LOAD_UOPS_MISC_RETIRED.LLC_MISS
MEM_LOAD_UOPS_RETIRED
MEM_TRANS_RETIRED
MEM_UOPS_RETIRED
OTHER_ASSISTS
ROB_MISC_EVENTS.LBR_INSERTS
UOPS_RETIRED

Any of the globally enabled (via IA32_CR_EMON_PERF_GLOBAL_CTRL) counters may overcount certain events when a disabled breakpoint condition is met

Implication: Performance-monitor counters may indicate a number greater than the number of events that occurred.

Workaround: Software can disable all breakpoints by clearing DR7. Alternatively, software can ensure that, for a breakpoint disabled in DR7, the corresponding debug-address register contains an address that prevents the breakpoint condition from being met (e.g., a non-canonical address).

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

BM98 LTR Message is Not Treated as an Unsupported Request Not Applicable

Problem: The PCIe* root port does not support LTR (Latency Tolerance Reporting) capability. However, a received LTR message is not treated as a UR (Unsupported Request).

Implication: Due to this erratum, an LTR message does not generate a UR error.

Workaround: None identified.

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

BM99 Use of VMASKMOV to Access Memory Mapped I/O or Uncached Memory May Cause The Logical Processor to Hang

Problem: Under a complex set of conditions, using VMASKMOV to reference memory mapped I/O or uncached memory may cause the logical processor to hang.

Implication: Due to this erratum, the logical processor may hang. Intel's Software Developers Manual states "VMASKMOV should not be used to access memory mapped I/O and uncached memory as the access and the ordering of the individual loads or stores it does is implementation specific." Intel has not observed this erratum with any commercially available software.

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

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

BM100 PEBS May Unexpectedly Signal a PMI After The PEBS Buffer is Full

Problem: The Software Developer's Manual states that no PMI should be generated when PEBS index reaches PEBS Absolute Maximum. Due to this erratum a PMI may be generated even though the PEBS buffer is full.

Implication: PEBS may trigger a PMI even though the PEBS index has reached the PEBS Absolute Maximum.

Workaround: None identified.

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

BM101 XSAVEOPT May Fail to Save Some State after Transitions Into or Out of STM

Problem: The XSAVEOPT instruction may optimize performance by not saving state that has not been modified since the last execution of XRSTOR. This optimization should occur only if the executions of XSAVEOPT and XRSTOR are either both or neither in SMM (system-management mode). Due to this erratum, this optimization may be performed by the first execution of XSAVEOPT after a transition into or out of the STM (SMM-transfer monitor) if the most recent execution of XRSTOR occurred before that transition. For transitions into the STM, the erratum applies only to transitions using the VMCALL instruction. This erratum can occur only if the two executions are at the same privilege level, use the same linear address, and are either both or neither in VMX non-root operation. The erratum does not apply if software in SMM never uses XRSTOR or XSAVEOPT.

Implication: This erratum may lead to unpredictable system behavior.

Workaround: STM software should execute the XRSTOR instruction with the value 0 in EDX:EAX after each transition into the STM (after setting CR4.OSXSAVE) and before each transition out of the STM. Bytes 512 to 575 of the save area used by XRSTOR should be allocated in memory, but bytes 0 to 511 need not be. Bytes 512 to 535 should all be 0.

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

BM102 Performance Monitor Precise Instruction Retired Event May Present Wrong Indications

Problem: When the PDIR (Precise Distribution for Instructions Retired) mechanism is activated (INST_RETIRED.ALL (event C0H, umask value 00H) on Counter 1 programmed in PEBS mode), the processor may return wrong PEBS/PMI interrupts and/or incorrect counter values if the counter is reset with a SAV below 100 (Sample-After-Value is the counter reset value software programs in MSR IA32_PMC1[47:0] in order to control interrupt frequency).

Implication: Due to this erratum, when using low SAV values, the program may get incorrect PEBS or PMI interrupts and/or an invalid counter state.

Workaround: The sampling driver should avoid using SAV<100.

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

BM103 The Value in IA32_MC3_ADDR MSR May Not be Accurate When MCACOD 0119H is Reported in IA32_MC3_Status

Problem: Under certain conditions, when the The Machine Check Error Code (MCACOD) in the IA32_MC3_STATUS (MSR 040DH) register is 0119H, the value in IA32_MC3_ADDR MSR (40EH) may refer to the incoming MLC (Mid-Level Cache) cache line instead of the evicted cache line.

Implication: The address in IA32_MC3_ADDR MSR (40EH) may not be accurate for MLC cache read errors with MSCOD of 119H.

Workaround: None identified.

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

BM104 MSR_PKG_Cx_RESIDENCY MSRs May Not be Accurate

Problem: If the processor is in a package C-state for an extended period of time (greater than 40 seconds) with no wake events, the value in the MSR_PKG_C{2,3,6,7}_RESIDENCY MSRs (60DH and 3F8H–3FAH) will not be accurate.

Implication: Utilities that report C-state residency times will report incorrect data in cases of long duration package C-states.

Workaround: None identified.

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

BM105 Enabling / Disabling PEBS May Result in Unpredictable System Behavior

Problem: Under certain conditions, enabling or disabling PEBS (Precise Event Based Sampling) via WRMSR to IA32_PEBS_ENABLE MSR may result in unpredictable system behavior near or coincident to this instruction.

Implication: Due to this erratum, unpredictable system behavior may result.

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

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

BM106 Execution of VAESIMC or VAESKEYGENASSIST With An Illegal Value for VEX.vvvv May Produce a #NM Exception

Problem: The VAESIMC and VAESKEYGENASSIST instructions should produce a #UD (Invalid-Opcode) exception if the value of the vvvv field in the VEX prefix is not 1111b. Due to this erratum, if CR0.TS is “1”, the processor may instead produce a #NM (Device-Not-Available) exception.

Implication: Due to this erratum, some undefined instruction encodings may produce a #NM instead of a #UD exception.

Workaround: Software should always set the vvvv field of the VEX prefix to 1111b for instances of the VAESIMC and VAESKEYGENASSIST instructions.

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

BM107 Unexpected #UD on VZEROALL/VZEROUPPER

Problem: Execution of the VZEROALL or VZEROUPPER instructions in 64-bit mode with VEX.W set to 1 may erroneously cause a #UD (invalid-opcode exception).

Implication: The affected instructions may produce unexpected invalid-opcode exceptions in 64-bit mode.

Workaround: Compilers should encode VEX.W = 0 for the VZEROALL and VZEROUPPER instructions.

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

BM108 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:

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

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

BM109 Execution of FXSAVE or FXRSTOR With the VEX Prefix May Produce a #NM Exception

Problem: Attempt to use FXSAVE or FXRSTOR with a VEX prefix should produce a #UD (Invalid-Opcode) exception. If either the TS or EM flag bits in CR0 are set, a #NM (device-not-available) exception will be raised instead of #UD exception.

Implication: Due to this erratum a #NM exception may be signaled instead of a #UD exception on an FXSAVE or an FXRSTOR with a VEX prefix.

Workaround: Software should not use FXSAVE or FXRSTOR with the VEX prefix.

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

BM110 VM Exits Due to “NMI-Window Exiting” May Not Occur Following a VM Entry to the Shutdown State

Problem: If VM entry is made with the “virtual NMIs” and “NMI-window exiting”, VM-execution controls set to 1, and if there is no virtual-NMI blocking after VM entry, a VM exit with exit reason “NMI window” should occur immediately after VM entry unless the VM entry

put the logical processor in the wait-for SIPI state. Due to this erratum, such VM exits do not occur if the VM entry put the processor in the shutdown state.

Implication: A VMM may fail to deliver a virtual NMI to a virtual machine in the shutdown state.

Workaround: Before performing a VM entry to the shutdown state, software should check whether the “virtual NMIs” and “NMI-window exiting” VM-execution controls are both 1. If they are, software should clear “NMI-window exiting” and inject an NMI as part of VM entry.

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

BM111 Execution of INVVPID Outside 64-Bit Mode Cannot Invalidate Translations For 64-Bit Linear Addresses

Problem: Executions of the INVVPID instruction outside 64-bit mode with the INVVPID type “individual-address invalidation” ignore bits 63:32 of the linear address in the INVVPID descriptor and invalidate translations for bits 31:0 of the linear address.

Implication: The INVVPID instruction may fail to invalidate translations for linear addresses that set bits in the range 63:32. Because this erratum applies only to executions outside 64-bit mode, it applies only to attempts by a 32-bit virtual-machine monitor (VMM) to invalidate translations for a 64-bit guest. Intel has not observed this erratum with any commercially available software.

Workaround: None identified.

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

BM112 VEX.L is Not Ignored with VCVT*2SI Instructions

Problem: The VEX.L bit should be ignored for the VCVTSS2SI, VCVTSD2SI, VCVTTSS2SI, and VCVTTSD2SI instructions, however due to this erratum the VEX.L bit is not ignored and will cause a #UD.

Implication: Unexpected #UDs will be seen when the VEX.L bit is set to 1 with VCVTSS2SI, VCVTSD2SI, VCVTTSS2SI, and VCVTTSD2SI instructions.

Workaround: Software should ensure that the VEX.L bit is set to 0 for all scalar instructions.

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

BM113 MCI_ADDR May be Incorrect For Cache Parity Errors

Problem: In cases when a WBINVD instruction evicts a line containing an address or data parity error (MCACOD of 0x124, and MSCOD of 0x10), the address of this error should be logged in the MCi_ADDR register. Due to this erratum, the logged address may be incorrect, even though MCi_Status.ADDRV (bit 63) is set.

Implication: The address reported in MCi_ADDR may not be correct for cases of a parity error found during WBINVD execution.

Workaround: None identified.

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

BM114 Instruction Fetches Page-Table Walks May be Made Speculatively to Uncacheable Memory

Problem: Page-table walks on behalf of instruction fetches may be made speculatively to uncacheable (UC) memory.

Implication: If any paging structures are located at addresses in uncacheable memory that are used for memory-mapped I/O, such I/O operations may be invoked as a result of speculative execution that would never actually occur in the executed code path. Intel has not observed this erratum with any commercially available software.

Workaround: Software should avoid locating paging structures at addresses in uncacheable memory that are used for memory-mapped I/O

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

BM115 The Processor May Not Properly Execute Code Modified Using A Floating-Point Store

Problem: Under complex internal conditions, a floating-point store used to modify the next sequential instruction may result in the old instruction being executed instead of the new instruction.

Implication: Self- or cross-modifying code may not execute as expected. Intel has not observed this erratum with any commercially available software.

Workaround: None identified. Do not use floating-point stores to modify code.

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

BM116 Execution of GETSEC[SEXIT] May Cause a Debug Exception to be Lost

Problem: A debug exception occurring at the same time that GETSEC[SEXIT] is executed or when an SEXIT doorbell event is serviced may be lost.

Implication: Due to this erratum, there may be a loss of a debug exception when it happens concurrently with the execution of GETSEC[SEXIT]. Intel has not observed this erratum with any commercially available software.

Workaround: None identified.

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

BM117 VM Exits Due to GETSEC May Save an Incorrect Value for "Blocking by STI" in the Context of Probe-Mode Redirection

Problem: The GETSEC instruction causes a VM exit when executed in VMX non-root operation. Such a VM exit should set bit 0 in the Interruptability-state field in the virtual-machine control structure (VMCS) if the STI instruction was blocking interrupts at the time GETSEC commenced execution. Due to this erratum, a VM exit executed in VMX non-root operation may erroneously clear bit 0 if redirection to probe mode occurs on the GETSEC instruction.

Implication: After returning from probe mode, a virtual interrupt may be incorrectly delivered prior to GETSEC instruction. Intel has not observed this erratum with any commercially software.

Workaround: None identified.

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

BM118 Not Applicable

BM119 IA32_MC5_CTL2 is Not Cleared by a Warm Reset

Problem: IA32_MC5_CTL2 MSR (285H) is documented to be cleared on any reset. Due to this erratum this MSR is only cleared upon a cold reset.

Implication: The algorithm documented in Software Developer's Manual, Volume 3, section titled "CMCI Initialization" or any other algorithm that counts the IA32_MC5_CTL2 MSR being cleared on reset will not function as expected after a warm reset

Workaround: None identified.

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

BM120 Performance Monitor Counters May Produce Incorrect Results

Problem: When operating in hyper-threaded mode, a memory at-retirement performance monitoring event (from the list below) may be dropped or may increment an enabled counter on the physical core's other thread rather than the thread experiencing the event. The list of affected memory at-retirement events is as follows:

- MEM_UOP_RETIRED.LOADS
- MEM_UOP_RETIRED.STORES
- MEM_UOP_RETIRED.LOCK
- MEM_UOP_RETIRED.SPLIT
- MEM_UOP_RETIRED.STLB_MISS
- MEM_LOAD_UOPS_RETIRED.HIT_LFB
- MEM_LOAD_UOPS_RETIRED.L1_HIT
- MEM_LOAD_UOPS_RETIRED.L2_HIT
- MEM_LOAD_UOPS_RETIRED.LLC_HIT
- MEM_LOAD_UOPS_MISC_RETIRED.LLC_MISS
- MEM_LOAD_UOPS_LLC_HIT_RETIRED.XSNP_HIT
- MEM_LOAD_UOPS_LLC_HIT_RETIRED.XSNP_HITM
- MEM_LOAD_UOPS_LLC_HIT_RETIRED.XSNP_MISS
- MEM_LOAD_UOPS_LLC_HIT_RETIRED.XSNP_NONE
- MEM_LOAD_UOPS_RETIRED.LLC_MISS
- MEM_LOAD_UOPS_LLC_MISS_RETIRED.LOCAL_DRAM
- MEM_LOAD_UOPS_LLC_MISS_RETIRED.REMOTE_DRAM
- MEM_LOAD_UOPS_RETIRED.L2_MISS

Implication: Due to this erratum, certain performance monitoring event will produce unreliable results during hyper-threaded operation.

Workaround: None identified.

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

BM121 The Corrected Error Count Overflow Bit in IA32_MCO_STATUS is Not Updated After a UC Error is Logged

Problem: When a UC (uncorrected) error is logged in the IA32_MC0_STATUS MSR (401H), corrected errors will continue to update the lower 14 bits (bits 51:38) of the Corrected Error Count. Due to this erratum, the sticky count overflow bit (bit 52) of the Corrected Error Count will not get updated after a UC error is logged.

Implication: The Corrected Error Count Overflow indication will be lost if the overflow occurs after an uncorrectable error has been logged.

Workaround: None identified.

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

BM122 Spurious Intel®VT-d Interrupts May Occur When the PFO Bit is Set

Problem: When the PFO (Primary Fault Overflow) field (bit [0] in the VT-d FSTS [Fault Status] register) is set to 1, further faults should not generate an interrupt. Due to this erratum, further interrupts may still occur.

Implication: Unexpected Invalidation Queue Error interrupts may occur. Intel has not observed this erratum with any commercially available software.

Workaround: Software should be written to handle spurious Intel® VT-d fault interrupts.

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

BM123 Processor May Livelock During On Demand Clock Modulation

Problem: The processor may livelock when (1) a processor thread has enabled on demand clock modulation via bit 4 of the IA32_CLOCK_MODULATION MSR (19AH) and the clock modulation duty cycle is set to 12.5% (02H in bits 3:0 of the same MSR), and (2) the other processor thread does not have on demand clock modulation enabled and that thread is executing a stream of instructions with the lock prefix that either split a cacheline or access UC memory.

Implication: Program execution may stall on both threads of the core subject to this erratum.

Workaround: This erratum will not occur if clock modulation is enabled on all threads when using on demand clock modulation or if the duty cycle programmed in the IA32_CLOCK_MODULATION MSR is 18.75% or higher.

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

BM124 The Upper 32 Bits of CR3 May be Incorrectly Used With 32-Bit Paging

Problem: When 32-bit paging is in use, the processor should use a page directory located at the 32-bit physical address specified in bits 31:12 of CR3; the upper 32 bits of CR3 should be ignored. Due to this erratum, the processor will use a page directory located at the 64-bit physical address specified in bits 63:12 of CR3.

Implication: The processor may use an unexpected page directory or, if EPT (Extended Page Tables) is in use, cause an unexpected EPT violation. This erratum applies only if software enters 64-bit mode, loads CR3 with a 64-bit value, and then returns to 32-bit paging without changing CR3. Intel has not observed this erratum with any commercially available software.

Workaround: Software that has executed in 64-bit mode should reload CR3 with a 32-bit value before returning to 32-bit paging.

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

BM125 EPT Violations May Report Bits 11:0 of Guest Linear Address Incorrectly

Problem: If a memory access to a linear address requires the processor to update an accessed or dirty flag in a paging-structure entry and if that update causes an EPT violation, the processor should store the linear address into the “guest linear address” field in the VMCS. Due to this erratum, the processor may store an incorrect value into bits 11:0 of this field. (The processor correctly stores the guest-physical address of the paging-structure entry into the “guest-physical address” field in the VMCS.)

Implication: Software may not be easily able to determine the page offset of the original memory access that caused the EPT violation. Intel has not observed this erratum to impact the operation of any commercially available software.

Workaround: Software requiring the page offset of the original memory access address can derive it by simulating the effective address computation of the instruction that caused the EPT violation.

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

BM126 IA32_VMX_VMCS_ENUM_MSR_(48AH) Does Not Properly Report The Highest Index Value Used For VMCS Encoding

Problem: IA32_VMX_VMCS_ENUM MSR (48AH) bits 9:1 report the highest index value used for any VMCS encoding. Due to this erratum, the value 21 is returned in bits 9:1 although there is a VMCS field whose encoding uses the index value 23.

Implication: Software that uses the value reported in IA32_VMX_VMCS_ENUM[9:1] to read and write all VMCS fields may omit one field.

Workaround: None identified.

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

BM127 Not Applicable

BM128 Virtual-APIC Page Accesses With 32-Bit PAE Paging May Cause a System Crash

Problem: If a logical processor has EPT (Extended Page Tables) enabled, is using 32-bit PAE paging, and accesses the virtual-APIC page then a complex sequence of internal processor micro-architectural events may cause an incorrect address translation or machine check on either logical processor.

Implication: This erratum may result in unexpected faults, an uncorrectable TLB error logged in IA32_MCi_STATUS.MCACOD (bits [15:0]) with a value of 0000_0000_0001_xxxxb (where x stands for 0 or 1), a guest or hypervisor crash, or other 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 Tables of Changes.

BM129 SMRAM State-Save Area Above the 4GB Boundary May Cause Unpredictable System Behavior

Problem: If BIOS uses the RSM instruction to load the SMBASE register with a value that would cause any part of the SMRAM state-save area to have an address above 4-GBytes, subsequent transitions into and out of SMM (system-management mode) might save and restore processor state from incorrect addresses.

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

Workaround: Ensure that the SMRAM state-save area is located entirely below the 4GB address boundary.

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

Specification Changes

1. DDR3 Signal Group DC Specifications Table Update

Affected Docs: *Intel Xeon® and Intel® Core™ Processors for Communications Infrastructure Datasheet - Volume 1 of 2*

New Text: **Table 9-10 "DDR3 Signal Group DC Specifications" --** Update the pull-up and pull-down buffer resistance specs to reflect silicon measurements. Added SM_RCOMP[0:2] values.

Implication: No Platform change needed.

Items in bold red are changed in this revision.

Symbol	Parameter	Min	Typ	Max	Units	Notes[1]
V_{IL}	Input Low Voltage	—	—	SM_VREF -0.1	V	2, 4, 11
V_{IH}	Input High Voltage	SM_VREF + 0.1	—	—	V	3, 11
V_{IL}	Input Low Voltage (SM_DRAMPWROK)	—	—	V_{DDQ}*0.55 -0.1	V	10
V_{IH}	Input High Voltage (SM_DRAMPWROK)	V_{DDQ}*0.55 +0.1	—	—	V	10
V_{OL}	Output Low Voltage	—	$(V_{DDQ} / 2)* (R_{ON} / (R_{ON}+R_{TERM}))$	—		6
V_{OH}	Output High Voltage	—	$V_{DDQ} - ((V_{DDQ} / 2)* (R_{ON}/(R_{ON}+R_{TERM}))$	—	V	4, 6
$R_{ON\_UP(DQ)}$	DDR3 Data Buffer pull-up Resistance	**23.9**	**28.2**	32.9	Ω	5
$R_{ON\_DN(DQ)}$	DDR3 Data Buffer pull-down Resistance	**21.4**	**26.8**	**34.3**	Ω	5
$R_{ODT(DQ)}$	DDR3 On-die termination equivalent resistance for data signals	83 41.5	100 **52.7**	117 65.0	Ω	
$V_{ODT(DC)}$	DDR3 On-die termination DC working point (driver set to receive mode)	0.43*V_{CC}	0.5*V_{CC}	0.56*V_{CC}	V	
$R_{ON\_UP(CK)}$	DDR3 Clock Buffer pull-up Resistance	**20.8**	**25.8**	**29.2**	Ω	5
$R_{ON\_DN(CK)}$	DDR3 Clock Buffer pull-down Resistance	20.8	**24.8**	31.2	Ω	5
$R_{ON\_UP(CMD)}$	DDR3 Command Buffer pull-up Resistance	16.0	**20.5**	**23.5**	Ω	5
$R_{ON\_DN(CMD)}$	DDR3 Command Buffer pull-down Resistance	**15.7**	**19.8**	24.0	Ω	5
$R_{ON\_UP(CTL)}$	DDR3 Control Buffer pull-up Resistance	**14.9**	**20.1**	**23.5**	Ω	5
$R_{ON\_DN(CTL)}$	DDR3 Control Buffer pull-down Resistance	**14.5**	**19.2**	**24.3**	Ω	5
I_{LI}	Input Leakage Current (DQ, CK) 0V 0.2*V_{DDQ} 0.8*V_{DDQ} V_{DDQ}	—	—	 ± 0.75 ± 0.55 ± 0.9 ± 1.4	mA	

Symbol	Parameter	Min	Typ	Max	Units	Notes[1]
I_{LI}	Input Leakage Current (CMD, CTL) 0V $0.2*V_{DDQ}$ $0.8*V_{DDQ}$ V_{DDQ}	—	—	 ± 0.85 ± 0.65 ± 1.10 ± 1.65	mA	
SM_RCOMP0	Command COMP Resistance	**138.6**	**140**	**141.4**	Ω	8
SM_RCOMP1	Data COMP Resistance	**25.74**	**26**	**26.26**	Ω	8
SM_RCOMP2	ODT COMP Resistance	**198**	**200**	**202**	Ω	8

2. Control Sideband and TAP Signal Group DC Specifications

Affected Docs: *Intel Xeon® and Intel® Core™ Processors for Communications Infrastructure Datasheet - Volume 1 of 2*

New Text: **Table 9-11 "Control Sideband and TAP Signal Group DC Specifications" --** Update the Input Leakage Current values in to reflect silicon measurements.

Implication: No Platform change needed.

Items in bold red are changed in this revision.

Symbol	Parameter	Min	Max	Units	Notes[1]
V_{IL}	Input Low Voltage		$V_{CCIO}*0.3$	V	2,3
V_{IH}	Input High Voltage	$V_{CCIO}*0.7$		V	2,3,5
V_{OL}	Output Low Voltage		$V_{CCIO}*0.1$	V	2
V_{OH}	Output High Voltage	$V_{CCIO}*0.9$		V	2,5
R_{ON}	Buffer on Resistance	23	73	Ω	
I_{LI}	Input Leakage Current **- PROCHOT#** **- TDO** **- All other signals in this group**		 **-0.20 to +2.00** **-0.20 to +2.00** **-0.20 to +0.50**	**mA**	4

Notes:
1. Unless otherwise noted, all specifications in this table apply to all processor frequencies.
2. The V_{CCIO} referred to in these specifications refers to instantaneous V_{CCIO}.
3. See the processor *I/O Buffer Models* for I/V characteristics.
4. For V_{IN} between "0" V and V_{CCIO}. Measured when the driver is tristated.
5. V_{IH} and V_{OH} may experience excursions above V_{CCIO}. However, input signal drivers must comply with the signal quality specifications.

Specification Clarifications

1. System Agent Vcc VID and DC Specifications

Affected Docs: *In2tel Xeon® and Intel® Core™ Processors for Communications Infrastructure Datasheet - Volume 1 of 2*

New Text: **Table 9-2 "VCCSA_VID Configuration" and Table 9-8 "System Agent (VCCSA) Supply DC Voltage and Current Specifications" --** Output pin VCCSA_VID may be either "0" or "1".

Implication: No Platform change needed. Because silicon that drive VCCSA_VID = 0 will function at the specified VCCSA = 0.9 voltage and those that drive VCCSA_VID = 1 will run at VCCSA = 0.8V or 0.9V. The following table specifies the different VCCSA_VID configurations. For VCCSA supply specification see Table 9-8.

Table 9-2. VCCSA_VID Configuration

VCCSA_VID	PROC_SELECT#	Selected VCCSA
0	1	0.9 V
1	1	0.9 V or 0.8 V
1	0	0.8 V[1]
0	0	0.8 V[1]

Note:
1. Some of VCCSA configurations are reserved for future Intel® processor families

Table 9-8. System Agent (V_{CCSA}) Supply DC Voltage and Current Specifications

Symbol	Parameter	Min	Typ	Max	Unit	Note
V_{CCSA}	Voltage for the System Agent and VCCSA_VCCSENCE **VCCSA_VID=0**	-	0.90	-	V	1
	Voltage for the System Agent and VCCSA_VCCSENCE **VCCSA_VID=1**		0.80		V	4
TOL_{CCSA}	V_{CCSA} Tolerance	AC+DC= ±5%			%	1
I_{CCMAX_VCCSA}	Max Current for V_{CCSA} Rail		-	6	A	1
I_{CCTDC_VCCSA}	Thermal Design Current (TDC) for V_{CCSA} Rail		-	6	A	1
Slew Rate	Voltage Ramp rate (dV/dT)	0.5		10	mV/uS	1
di/dt	Step current	2			A	2, 3

Notes:
1. Long term reliability cannot be assured in conditions above or below Max/Min functional limits.
2. Step current is done in 100nS
3. di/dt values are for platform testing only. This parameter is not tested on Intel silicon. Testing should go up to and include IccMax.
4. Silicon presenting VCCSA_VID=1 operates with VCCSA of either 0.9V or 0.8V +/-5%

Document-Only Changes

All Documentation Changes will be incorporated into a future version of the appropriate processor documentation.

Note: Documentation changes for the *Intel® 64 and IA-32 Architecture Software Developer's Manual* volumes 1, 2A, 2B, 3A, and 3B will be posted in a separate document: the *Intel® 64 and IA-32 Architecture Software Developer's Manual: Document Changes*.

Link to the document: http://www.intel.com/content/www/us/en/architecture-and-technology/64-ia-32-architectures-software-developers-manual.html

1. Dual-Channel Asymmetric Mode

Affected Docs: *Intel Xeon® and Intel® Core™ Processors for Communications Infrastructure Datasheet - Volume 1 of 2*

New Text: Update the document with the following changes:

- Remove references to Dual-Channel Asymmetric Mode in **Section 3.1.3.2 *Dual-Channel Mode - Intel® Flex Memory Technology Mode*** as it is redundant with the description of Intel Flex Memory Technology Mode
- Remove Section **3.1.3.2.2 *Dual-Channel Asymmetric Mode*** as it is redundant with the description of Intel Flex Memory Technology Mode

2. CKE Power-Down

Affected Docs: *Intel Xeon® and Intel® Core™ Processors for Communications Infrastructure Datasheet - Volume 1 of 2*

New Text: Update **Section 6.3.2 *DRAM Power Management and Initialization*** of the document to add the following:

The CKE is one of the power-save means. When CKE is off the internal DDR clock is disabled and the DDR power is reduced. The power-saving differs according the selected mode and the DDR type used. For more information please refer to the IDD table in DDR spec.

The DDR defines 3 levels of power-down which differ in power-saving and in wakeup time:

1. Active power-down (APD): This mode is entered if there are open pages when de-asserting CKE. In this mode the open pages are retained. Power-saving in this mode is the lowest. Power consumption of DDR is defined by IDD3P. Exiting this mode is fined by tXP – small number of cycles.
2. Precharged power-down (PPD): This mode is entered if all banks in DDR are precharged when de-asserting CKE. Power-saving in this mode is intermediate – better than APD, but less than DLL-off. Power consumption is defined by IDD2P1. Exiting this mode is defined by tXP. Difference from APD mode is that when waking-up all page-buffers are empty
3. DLL-off: In this mode the data-in DLL's on DDR are off. Power-saving in this mode is the best among all power-modes. Power consumption is defined by IDD2P1.

Exiting this mode is defined by tXP, but also tXPDLL (10 – 20 according to DDR type) cycles until first data transfer is allowed.

Processor supports 5 different types of powerdown. The different modes are the powerdown modes supported by DDR3 and combinations of these. Type of CKE power-down is defined by configuration. There are following options:

1. No powerdown
2. APD – The rank enters power-down as soon as idle-timer expires, no matter what is the bank status
3. PPD – When idle timer expires the MC sends PRE-all to rank and then enters power-down
4. DLL-off – same as option (2) but DDR is configured to DLL-off
5. APD, change to PPD (APD-PPD) – Begins as option (1), and when all page-close timers of the rank are expired, it wakes the rank, issues PRE-all and returns to PPD
6. APD, change to DLL-off (APD_DLLoff) –
7. Begins as option (1), and when all page-close timers of the rank are expired, it wakes the rank, issues PRE-all and returns to DLL-off power-down

The CKE is determined per rank, whenever it is inactive. Each rank has an idle-counter. The idle-counter starts counting as soon as the rank has no accesses, and if it expires, the rank may enter power-down while no new transactions to the rank arrive to queues. Note that the idle-counter begins counting at the last incoming transaction arrival.

It is important to understand that since the powerdown decision is per rank, the MC can find a lot of opportunities to powerdown ranks even while running memory intensive applications, and savings are significant (may be few Watts, according to DDR spec). This is significant when each channel is populated with more ranks.

Selection of power modes should be according to power-performance or thermal tradeoffs of a given system:

- When trying to achieve maximum performance and power or thermal consideration is non issue: use no powerdown
- In a system which tries to minimize power-consumption, try to use the deepest powerdown mode possible – DLL-off or APD_DLLoff
- In high-performance systems with dense packaging (i.e. tricky thermal design) the powerdown mode should be considered in order to reduce the heating and avoid DDR throttling caused by the heating.

Control of the power-mode through CRB-BIOS: The BIOS selects by default no-powerdown. There are knobs to change the powerdown selected mode.

Another control is the idle timer expiration count. This is set through PM_PDWN_config bits 7:0 (MCHBAR +4CB0). As this timer is set to a shorter time the MC will have more opportunities to put DDR in powerdown. Minimum recommended value for this register is 15. There is no BIOS hook to set this register. Customers who choose to change the value of this register can do it by changing his BIOS. For experiments, this register can be modified in real time if BIOS didn't lock the MC registers.

Note that in APD, APD-PPD and APD-DLLoff there is no point in setting the idle counter in the same range of page-close idle timer.

Another option associated with CKE power-down is the S_DLL-off. When this option is enabled, the SBR I/O slave DLL's go off when all channel ranks are in power-down. (Do **not** confuse it with the DLL-off mode, in which the DDR DLL's are off) This mode requires to define IO slave DLL wakeup time.

3. On-Demand Clock Modulation Feature Clarification

Software Controlled Clock Modulation section of the Intel® 64 and IA-32 Architectures Software Developer's Manual, Volume 3B: System Programming Guide will be modified to differentiate On-demand clock modulation feature on different processors. The clarification will state:

For Hyper-Threading Technology enabled processors, the IA32_CLOCK_MODULATION register is duplicated for each logical processor. In order for the On-demand clock modulation feature to work properly, the feature must be enabled on all the logical processors within a physical processor. If the programmed duty cycle is not identical for all the logical processors, the processor clock will modulate to the highest duty cycle programmed for processors if the CPUID DisplayFamily_DisplayModel signatures is listed in Table 14-2. For all other processors, if the programmed duty cycle is not identical for all logical processors in the same core, the processor will modulate at the lowest programmed duty cycle.

For multiple processor cores in a physical package, each core can modulate to a programmed duty cycle independently.

For the P6 family processors, on-demand clock modulation was implemented through the chipset, which controlled clock modulation through the processor's STPCLK# pin.

Table 14-2. CPUID Signatures for Legacy Processors That Resolve to Higher Performance Setting of Conflicting Duty Cycle Requests.

DisplayFamily_DisplayModel	DisplayFamily_DisplayModel	DisplayFamily_DisplayModel	DisplayFamily_DisplayModel
0F_xx	06_1C	06_1A	06_1E
06_1F	06_25	06_26	06_27
06_2C	06_2E	06_2F	06_35
06_36			

§ §