Intel® Core™ i7-600, i5-500, i5-400 and i3-300 Mobile Processor Series

Specification Update

September 2015

Revision 024

Document Number: 322814-024US

Contents

§ §

Revision History

Revision	Description	Date
001	Initial Release	January 2010
002	• Update processor identification table: Intel® Core™ i3-350, i3-330, and i5-430 Mobile Processor Series do not support Intel® VT-d. • Added Errata AAT85, AAT86, and AAT87	February 2010
003	• Added Errata AAT88, AAT89, and AAT90. • Added Documentation Changes	March 2010
004	• Added Erratum AAT91.	April 2010
005	• Added Errata AAT92, AAT93, AAT94, AAT95, AAT96 • Corrected part information on Intel® Core™ i3-350, 370, i5-450 and others	May 2010
006	• Added Errata AAT97-AAT100 • Fixed AAT70	July 2010
007	• Added Errata AAT97-AAT100 • Fixed AAT70	August 2010
008	• Added Erratum AAT101 • Updated processor identification table with new sku info	September 2010
009	• Added Errata AAT102-104 • Changed wording on AAT32 and AAT77	October 2010
010	• Added Errata AAT106-107	December 2010
011	• Added Errata AAT108-109 • Updated processor table	January 2011
012	• Updated processor identification table with new sku info • Added Errata AAT110 through AAT115	February 2011
013	• Added Errata AAT116	March 2011
014	• Added Errata AAT117-118	September 2011
015	• Added Errata AAT119-123	December 2011
016	• Added Erratum AAT125	May 2013
017	• Added Errata AAT126-AAT128	June 2013
018	• Added Errata AAT129	August 2013
019	• No errata updates added • Document standardization	October 2013
020	• No errata updates added • Document standardization	December 2013
021	• Updated link to access Intel® 64and IA-32 Architecture Software Developer's Manual • Added Errata AAT130	July 2014

Revision	Description	Date
022	• Removed Erratum AAT65 • Updated Erratum AAT38	November 2014
023	• Updated Erratum AAT125	February 2015
024	• Added Erratum AAT131	September 2015

§ §

Preface

This document is an update to the specifications contained in the Affected Documents table below. This document is a compilation of device and documentation errata, specification clarifications and changes. It is intended for hardware system manufacturers and software developers of applications, operating systems, or tools.

Information types defined in Nomenclature are consolidated into the specification update and are no longer published in other documents.

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

Affected Documents

Document Title	Document Number / Location
Intel® Core™ i7-600, i5-500, i5-400, and i3-300 Mobile Processor Series Datasheet - Volume 1	322812
Intel® Core™ i7-600, i5-500, i5-400, and i3-300 Mobile Processor Series Datasheet - Volume 2	322813

Related Documents

Document Title	Document Number / Location
AP-485, Intel® Processor Identification and the CPUID Instruction (see Note 2)	http://www.intel.com/design/processor/applnots/241618.htm
Intel® 64 and IA-32 Architectures Software Developer's Manual, Volume 1: Basic Architecture *Intel® 64 and IA-32 Architectures Software Developer's Manual, Volume 2A: Instruction Set Reference Manual A-M* *Intel® 64 and IA-32 Architectures Software Developer's Manual, Volume 2B: Instruction Set Reference Manual N-Z* *Intel® 64 and IA-32 Architectures Software Developer's Manual, Volume 3A: System Programming Guide* *Intel® 64 and IA-32 Architectures Software Developer's Manual, Volume 3B: System Programming Guide* *Intel® 64 and IA-32 Intel Architecture Optimization Reference Manual* *Intel® 64 and IA-32 Architectures Software Developer's Manual Documentation Changes* (see note 1)	http://www.intel.com/content/www/us/en/processors/architectures-software-developer-manuals.html
ACPI Specifications	www.acpi.info

Notes:
1. Documentation changes for Intel® 64 and IA-32 Architecture Software Developer's Manual volumes 1, 2A, 2B, 3A, and 3B, and bug fixes are posted in the *Intel® 64 and IA-32 Architecture Software Developer's Manual Documentation Changes*.
2. Document Number subject to change. See IBL/IBP for the most up-to-date collateral list. Contact your Intel representative to receive the latest revisions for these documents.

Nomenclature

Errata are design defects or errors. These may cause the Intel® Core™ i7-600, i5-500, i5-400, and i3-300 Mobile Processor Series 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 is a five-digit code used to identify products. Products are differentiated by their unique characteristics,e.g., core speed, L3 cache size, package type, etc. as described in the processor identification information table. Read all notes associated with each S-Spec number.

Specification Changes are modifications to the current published specifications. These changes will be incorporated in any new release of the specification.

Specification Clarifications describe a specification in greater detail or further highlight a specification's impact to a complex design situation. These clarifications will be incorporated in any new release of the specification.

Documentation Changes include typos, errors, or omissions from the current published specifications. These will be incorporated in any new release of the specification.

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

§ §

Summary Tables of Changes

The following tables indicate the errata, specification changes, specification clarifications, or documentation changes which apply to the processor. Intel may fix some of the errata in a future stepping of the component, and account for the other outstanding issues through documentation or specification changes as noted. These tables uses the following notations:

Codes Used in Summary Tables

Stepping

X:	Errata exists in the stepping indicated. Specification Change or Clarification that applies to this stepping.
(No mark)	
or (Blank box):	This erratum is fixed in listed stepping or specification change does not apply to listed stepping.

Page

(Page):	Page location of item in this document.

Status

Doc:	Document change or update will be implemented.
Plan Fix:	This erratum may be fixed in a future stepping of the product.
Fixed:	This erratum has been previously fixed.
No Fix:	There are no plans to fix this erratum.

Row

Change bar to left of table row indicates this erratum is either new or modified from the previous version of the document.

Errata (Sheet 1 of 6)

Number	Steppings		Status	ERRATA
	C-2	K-0		
AAT1	X	X	No Fix	The Processor May Report a #TS Instead of a #GP Fault
AAT2	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
AAT3	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
AAT4	X	X	No Fix	Performance Monitor SSE Retired Instructions May Return Incorrect Values
AAT5	X	X	No Fix	Premature Execution of a Load Operation Prior to Exception Handler Invocation
AAT6	X	X	No Fix	MOV To/From Debug Registers Causes Debug Exception
AAT7	X	X	No Fix	Incorrect Address Computed for Last Byte of FXSAVE/FXRSTOR Image Leads to Partial Memory Update
AAT8	X	X	No Fix	Values for LBR/BTS/BTM Will Be Incorrect after an Exit from SMM
AAT9	X	X	No Fix	Single Step Interrupts with Floating Point Exception Pending May Be Mishandled
AAT10	X	X	No Fix	Fault on ENTER Instruction May Result in Unexpected Values on Stack Frame
AAT11	X	X	No Fix	IRET under Certain Conditions May Cause an Unexpected Alignment Check Exception
AAT12	X	X	No Fix	General Protection Fault (#GP) for Instructions Greater Than 15 Bytes May Be Preempted
AAT13	X	X	No Fix	General Protection (#GP) Fault May Not Be Signaled on Data Segment Limit Violation above 4-G Limit
AAT14	X	X	No Fix	LBR, BTS, BTM May Report a Wrong Address When an Exception/Interrupt Occurs in 64-bit Mode
AAT15	X	X	No Fix	MONITOR or CLFLUSH on the Local XAPIC's Address Space Results in Hang
AAT16	X	X	No Fix	Corruption of CS Segment Register during RSM While Transitioning from Real Mode to Protected Mode
AAT17	X	X	No Fix	Performance Monitoring Events for Read Miss to Level 3 Cache Fill Occupancy Counter May Be Incorrect
AAT18	X	X	No Fix	A VM Exit on MWAIT May Incorrectly Report the Monitoring Hardware As Armed
AAT19	X	X	No Fix	Performance Monitor Event SEGMENT_REG_LOADS Counts Inaccurately
AAT20	X	X	No Fix	#GP on Segment Selector Descriptor That Straddles Canonical Boundary May Not Provide Correct Exception Error Code
AAT21	X	X	No Fix	Improper Parity Error Signaled in the IQ Following Reset When a Code Breakpoint Is Set on a #GP Instruction

Errata (Sheet 2 of 6)

Number	Steppings		Status	ERRATA
	C-2	K-0		
AAT22	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
AAT23	X	X	No Fix	IA32_MPERF Counter Stops Counting during On-Demand TM1
AAP24	X	X	No Fix	The Memory Controller tTHROT_OPREF Timings May Be Violated during Self-Refresh Entry
AAT25	X	X	No Fix	Synchronous Reset of IA32_APERF/IA32_MPERF Counters on Overflow Does Not Work
AAT26	X	X	No Fix	Disabling Thermal Monitor While Processor Is Hot, Then Re-enabling, May Result in Stuck Core Operating Ratio
AAT27	X	X	No Fix	Writing the Local Vector Table (LVT) When an Interrupt Is Pending May Cause an Unexpected Interrupt
AAT28	X	X	No Fix	xAPIC Timer May Decrement Too Quickly Following an Automatic Reload While in Periodic Mode
AAT29	X	X	No Fix	Changing the Memory Type for an In-Use Page Translation May Lead to Memory-Ordering Violations
AAT30	X	X	No Fix	Infinite Stream of Interrupts May Occur If an ExtINT Delivery Mode Interrupt Is Received While All Cores Are in C6
AAT31	X	X	No Fix	Two xAPIC Timer Event Interrupts May Unexpectedly Occur
AAT32	X	X	No Fix	EOI Transaction May Not Be Sent If Software Enters Core C6 during an Interrupt Service Routine
AAT33	X	X	No Fix	FREEZE_WHILE_SMM Does Not Prevent Event from Pending PEBS during SMM
AAT34	X	X	No Fix	APIC Error "Received Illegal Vector" May Be Lost
AAT35	X	X	No Fix	DR6 May Contain Incorrect Information When the First Instruction after a MOV SS,r/m or POP SS Is a Store
AAT36	X	X	No Fix	An Uncorrectable Error Logged in IA32_CR_MC2_STATUS May Also Result in a System Hang
AAT37	X	X	No Fix	IA32_PERF_GLOBAL_CTRL MSR May Be Incorrectly Initialized
AAT38	X	X	No Fix	Performance Monitor Counter MEM_INST_RETIRED.STORES May Count Higher Than Expected
AAT39	X	X	No Fix	Sleeping Cores May Not Be Woken up on Logical Cluster Mode Broadcast IPI Using Destination Field Instead of Shorthand
AAT40	X	X	No Fix	Faulting Executions of FXRSTOR May Update State Inconsistently
AAT41	X	X	No Fix	Performance Monitor Event EPT.EPDPE_MISS May Be Counted While EPT Is Disabled
AAT42	X	X	No Fix	Memory Aliasing of Code Pages May Cause Unpredictable System Behavior
AAT43	X	X	No Fix	Performance Monitor Counters May Count Incorrectly
AAT44	X	X	No Fix	Performance Monitor Event Offcore_response_0 (B7H) Does Not Count NT Stores to Local DRAM Correctly

Errata (Sheet 3 of 6)

Number	Steppings		Status	ERRATA
	C-2	K-0		
AAT45	X	X	No Fix	EFLAGS Discrepancy on Page Faults and on EPT-Induced VM Exits after a Translation Change
AAT46	X	X	No Fix	Back to Back Uncorrected Machine Check Errors May Overwrite IA32_MC3_STATUS.MSCOD
AAT47	X	X	No Fix	Corrected Errors with a Yellow Error Indication May Be Overwritten by Other Corrected Errors
AAT48	X	X	No Fix	Performance Monitor Events DCACHE_CACHE_LD and DCACHE_CACHE_ST May Overcount
AAT49	X	X	No Fix	Rapid Core C3/C6 Transitions May Cause Unpredictable System Behavior
AAT50	X	X	No Fix	Performance Monitor Events INSTR_RETIRED and MEM_INST_RETIRED May Count Inaccurately
AAT51	X	X	No Fix	A Page Fault May Not Be Generated When the PS bit Is Set to "1" in a PML4E or PDPTE
AAT52	X	X	No Fix	BIST Results May Be Additionally Reported after a GETSEC[WAKEUP] or INIT-SIPI Sequence
AAT53	X	X	No Fix	Pending x87 FPU Exceptions (#MF) May Be Signaled Earlier Than Expected
AAT54	X	X	No Fix	VM Exits Due to "NMI-Window Exiting" May Be Delayed by One Instruction
AAT55	X	X	No Fix	VM Exits Due to EPT Violations Do Not Record Information about Pre-IRET NMI Blocking
AAT56	X	X	No Fix	Multiple Performance Monitor Interrupts Are Possible on Overflow of IA32_FIXED_CTR2
AAT57	X	X	No Fix	LBRs May Not Be Initialized during Power-On Reset of the Processor
AAT58	X	X	No Fix	LBR, BTM or BTS Records May Have Incorrect Branch from Information after an Enhanced Intel SpeedStep® Technology Transition, T-states, C1E, or Adaptive Thermal Throttling
AAT59	X	X	No Fix	VMX-Preemption Timer Does Not Count Down at the Rate Specified
AAT60	X	X	No Fix	Multiple Performance Monitor Interrupts Are Possible on Overflow of Fixed Counter 0
AAT61	X	X	No Fix	VM Exits Due to LIDT/LGDT/SIDT/SGDT Do Not Report Correct Operand Size
AAT62	X	X	No Fix	DPRSLPVR Signal May Be Incorrectly Asserted on Transition between Low Power C-states
AAT63	X	X	No Fix	Performance Monitoring Events STORE_BLOCKS.NOT_STA and STORE_BLOCKS.STA May Not Count Events Correctly
AAT64	X	X	No Fix	Storage of PEBS Record Delayed Following Execution of MOV SS or STI
AAT65	X	X	No Fix	<Erratum Removed>.
AAT66	X	X	No Fix	INVLPG Following INVEPT or INVVPID May Fail to Flush All Translations for a Large Page
AAT67	X	X	No Fix	LER MSRs May Be Unreliable

Errata (Sheet 4 of 6)

Number	Steppings		Status	ERRATA
	C-2	K-0		
AAT68	X	X	No Fix	MCi_Status Overflow Bit May Be Incorrectly Set on a Single Instance of a DTLB Error
AAT69	X	X	No Fix	Debug Exception Flags DR6.B0-B3 Flags May Be Incorrect for Disabled Breakpoints
AAT70	X	X	No Fix	Erratum removed as it does not apply to the Intel® Core™ i7-600, i5-500, i5-400 and i3-300 Mobile Processor Series
AAT71	X	X	Plan Fix	Delivery of Certain Events Immediately Following a VM Exit May Push a Corrupted RIP onto the Stack
AAT72	X	X	No Fix	A String Instruction that Re-maps a Page May Encounter an Unexpected Page Fault
AAT73	X	X	No Fix	Logical Processor May Use Incorrect VPID after VM Entry That Returns From SMM
AAT74	X	X	No Fix	MSR_TURBO_RATIO_LIMIT MSR May Return Intel® Turbo Boost Technology Core Ratio Multipliers for Non-Existent Core Configurations
AAT75	X	X	No Fix	PCI Express x16 Port Logs Bad TLP Correctable Error When Receiving a Duplicate TLP
AAT76	X	X	No Fix	PCI Express x16 Root Port Incorrectly NAK's a Nullified TLP
AAT77	X	X	No Fix	PCI Express Graphics x16 Receiver Error Reported When Receiver With L0s Enabled and Link Retrain Performed
AAT78	X	X	No Fix	Internal Parity Error May Be Incorrectly Signaled during C6 Exit
AAT79	X	X	No Fix	PMIs during Core C6 Transitions May Cause the System to Hang
AAT80	X	X	No Fix	2-MB Page Split Lock Accesses Combined with Complex Internal Events May Cause Unpredictable System Behavior
AAT81	X	X	No Fix	Extra APIC Timer Interrupt May Occur during a Write to the Divide Configuration Register
AAT82	X	X	Plan Fix	TXT.PUBLIC.KEY Is Not Reliable
AAT83	X	X	Plan Fix	8259 Virtual Wire B Mode Interrupt May Be Dropped When It Collides with Interrupt Acknowledge Cycle from the Preceding Interrupt
AAT84	X	X	No Fix	CPUID Incorrectly Reports a C-State as Available When This State Is Unsupported
AAT85	X	X	No Fix	The Combination of a Page-Split Lock Access and Data Accesses That Are Split across Cacheline Boundaries May Lead to Processor Livelock
AAT86	X	X	No Fix	Processor Hangs on Package C6 State Exit
AAT87	X	X	No Fix	A Synchronous SMI May Be Delayed
AAT88	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
AAT89	X	X	No Fix	PCI Express Cards May Not Train to x16 Link Width
AAT90	X	X	No Fix	The APIC Timer Current Count Register May Prematurely Read 0x0 While the Timer Is Still Running
AAT91	X	X	No Fix	IO_SMI Indication in SMRAM State Save Area May Be Lost

Errata (Sheet 5 of 6)

Number	Steppings		Status	ERRATA
	C-2	K-0		
AAT92	X	X	No Fix	FSW May Be Corrupted If an x87 Store Instruction Causes a Page Fault in VMX Non-Root Operation After a PD Exit
AAT93	X	X	No Fix	CKE May go Low Within tRFC(min) After a PD Exit
AAT94	X	X	No Fix	Under Certain Low Temperature Conditions, Some Uncore Performance Monitoring Events May Report Incorrect Results
AAT95	X	X	No Fix	VM Entry to 64-Bit Mode May Fail if Bits 48 And 47 of Guest RIP Are Different
AAT96	X	X	No Fix	VM Entry Loading an Unusable SS Might Not Set SS.B to 1
AAT97	X	X	No Fix	Accesses to a VMCS May Not Operate Correctly If CR0.CD is Set on Any Logical Processor of a Core
AAT98	X	X	No Fix	Performance Monitor Events for Hardware Prefetches Which Miss The L1 Data Cache May be Over Counted
AAT99	X		No Fix	Correctable and Uncorrectable Cache Errors May be Reported Until the First Core C6 Transition
AAT100	X	X	No Fix	VM Exit May Incorrectly Clear IA32_PERF_GLOBAL_CTRL [34:32]
AAT101	X	X	No Fix	DTS Temperature Data May Be Incorrect On a Return From the Package C6 Low Power State
AAT102	X	X	No Fix	USB Devices May Not Function Properly With Integrated Graphics While Running Targeted Stress Graphics Workloads With Non-Matching Memory Configurations
AAT103	X	X	No Fix	VM Entry May Omit Consistency Checks Related to Bit 14 (BS) of the Pending Debug Exception Field in Guest-State Area of the VMCS
AAT104		X	No Fix	Intel Turbo Boost Technology Ratio Changes May Cause Unpredictable System Behavior
AAT105	—	—	—	—
AAT106	X	—	No Fix	Execution of VMPTRLD May Corrupt Memory If Current-VMCS Pointer is Invalid
AAT107	X	X	No Fix	PerfMon Overflow Status Can Not be Cleared After Certain Conditions Have Occurred
AAT108	X	X	No Fix	An Unexpected Page Fault or EPT Violation May Occur After Another Logical Processor Creates a Valid Translation for a Page
AAT109	X	X	No Fix	L1 Data Cache Errors May be Logged With Level Set to 1 Instead of 00
AAT110	X	X	No Fix	Executing The GETSEC Instruction While Throttling May Result in a Processor Hang
AAT111	X	X	No Fix	PerfMon Event LOAD_HIT_PRE.SW_PREFETCH May Overcount
AAT112	X	X	No Fix	Successive Fixed Counter Overflows May be Discarded
AAT113	X	X	No Fix	#GP May be Signaled When Invalid VEX Prefix Precedes Conditional Branch Instructions
AAT114	X	X	No Fix	A Logical Processor May Wake From Shutdown State When Branch-Trace Messages or Branch-Trace Stores Are Enabled

Errata (Sheet 6 of 6)

Number	Steppings		Status	ERRATA
	C-2	K-0		
AAT115	X	X	No Fix	Task Switch to a TSS With an Inaccessible LDTR Descriptor May Cause Unexpected Faults
AAT116	X	X	No Fix	MCIP Bit Not Checked on SENTER or ENTERACCS
AAT117	X	X	No Fix	EOI-Broadcast Suppression May Not Function Properly if Enabled or Disabled While an Interrupt is in Service
AAT118	X	X	No Fix	Unexpected Load May Occur on Execution of Certain Opcodes
AAT119	X	X	No Fix	VM Entry Loading an Unusable SS Might Not Clear Bits 3:0 of the SS Base Address
AAT120	X	X	No Fix	A First Level Data Cache Parity Error May Result in Unexpected Behavior
AAT121	X	X	No Fix	A Page Fault May Not be Generated When the PS bit is set to "1" in a PML4E or PDPTE
AAT122	X	X	No Fix	An Unexpected Page Fault or EPT Violation May Occur After Another Logical Processor Creates a Valid Translation for a Page
AAT123	X	X	No Fix	Intel® Trusted Execution Technology ACM Revocation
AAT124	X	X	No Fix	Intel® Processor Graphics VT-d IOTLB Invalidation Operation May Fail to Complete
AAT125	X	X	No Fix	The Corrected Error Count Overflow Bit in IA32_ MC0_STATUS is Not Updated When the UC Bit is Set
AAT126	X	X	No Fix	The Upper 32 Bits of CR3 May be Incorrectly Used With 32-Bit Paging
AAT127	X	X	No Fix	EPT Violations May Report Bits 11:0 of Guest Linear Address Incorrectly
AAT128	X	X	No Fix	IA32_VMX_VMCS_ENUM MSR (48AH) Does Not Properly Report The Highest Index Value Used For VMCS Encoding
AAT129	X	X	No Fix	Virtual-APIC Page Accesses With 32-Bit PAE Paging May Cause a System Crash
AAT130	X	X	No Fix	VM Exit May Set IA32_EFER.NXE When IA32_MISC_ENABLE Bit 34 is Set to 1
AAT131	X	X	No Fix	An IRET Instruction That Results in a Task Switch Does Not Serialize The Processor

Specification Changes

Number	SPECIFICATION CHANGES
	None for this revision of this specification update.

Specification Clarifications

Number	SPECIFICATION CLARIFICATIONS
	None for this revision of this specification update.

Documentation Changes

Number	DOCUMENTATION CHANGES
AAT1	On-Demand Clock Modulation Feature Clarification

§ §

Identification Information

Component Identification via Programming Interface

The Intel® Core™ i7-600, i5-500, i5-400, and i3-300 Mobile Processor Series stepping can be identified by the following processor signatures:

Reserved	Extended Family[1]	Extended Model[2]	Reserved	Processor Type	Family Code[3]	Model Number[4]	Stepping ID[5]
31:28	**27:20**	**19:16**	**15:14**	**13:12**	**11:8**	**7:4**	**3:0**
	00000000b	0010b		00b	0110	0101b	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 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.
4. 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.
5. The Stepping ID in bits [3:0] indicates the revision number of that model. See Table 1 for the processor stepping ID number in the CPUID information.

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

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

The Intel® Core™ i7-600, i5-500, i5-400, and i3-300 Mobile Processor Series can be identified by the following register contents:

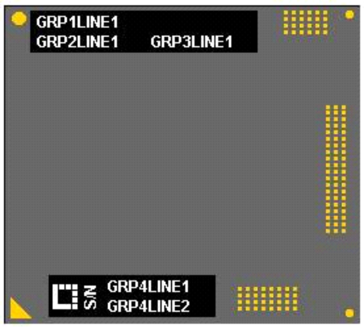
Processor Stepping	Vendor ID[1]	Device ID[2]	Revision ID[3]
C-2	8086h	0044h	12h

Notes:
1. The Vendor ID corresponds to Bits 15:0 of the Vendor ID Register located at offset 00–01h in the PCI function 0 configuration space.
2. The Device ID corresponds to Bits 15:0 of the Device ID Register located at Device 0 offset 02–03h in the PCI function 0 configuration space.
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.
4. Correct Host Device ID requires firmware support.

Component Marking Information

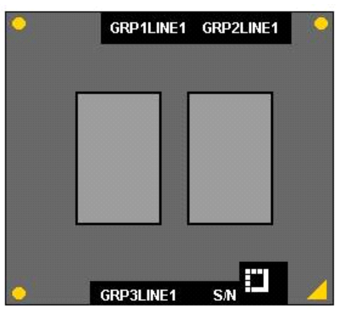
The processor stepping can be identified by the following component markings:

Figure 1. Intel® Core™ i7-600, i5-500, i5-400, and i3-300 Mobile Processor Series PGA Component Markings



- For PGA
  - GRP1LINE1 (limited to 18 char, 13 pt font): INTEL{M}{C}'YY
  - GRP2LINE1 (limited to 15 char, 13 pt font): PROC#
  - GRP3LINE1: (limited to 1 (ex) char, 15 pt font): {e4}
  - GRP4LINE1 (limited to 9 char, 13 pt font): SSPEC
  - GRP4LINE2 (limited to 9 char, 13 pt font): {FPO}

Figure 2. Intel® Core™ i7-600, i5-500, i5-400, and i3-300 Mobile Processor Series BGA Component Markings



- For BGA
  - GRP1LINE1 (limited to 23 char, 13 pt font): i{M}{C}'YY SSPEC PROC#
  - GRP2LINE1: (limited to 1 (ex) char, 15 pt font): {e1}
  - GRP3LINE1 (limited to 12 char, 13 pt font): {FPO}

Table 1. Processor Identification (Sheet 1 of 4)

S-Spec Number	Processor Number	Stepping / Processor Signature / Host Device ID / Host Revision ID	L3 Cache (MB)	Frequency	Max Intel® Turbo Boost Technology Frequency	LFM Frequency	Package	Notes
				Core Base (GHz) Graphics Base (MHz) DDR3 (MT/s)	Single Core Turbo Dual Core Turbo Graphics Turbo			
SLBPD	i7-620M	C-2/ 20652h/ 0044h/12h	4 MB	Core: 2.66 GHz Gfx: 500 MHz DDR3: 1066/800 MT/s	2 core: 3.06 GHz 1 core: 3.33 GHz Gfx: 766 MHz	1.20 GHz	rPGA	1, 2, 3, 4, 5, 6, 7, 8, 10, 13, 14
SLBPE	i7-620M	C-2/ 20652h/ 0044h/12h	4 MB	Core: 2.66 GHz Gfx: 500 MHz DDR3: 1066/800 MT/s	2 core: 3.06 GHz 1 core: 3.33 GHz Gfx: 766 MHz	1.20 GHz	BGA	1, 2, 3, 4, 5, 6, 7, 8, 10, 13, 14
SLBPG	i5-540M	C-2/ 20652h/ 0044h/12h	3 MB	Core: 2.53 GHz Gfx: 500 MHz DDR3: 1066/800 MT/s	2 core: 2.80 GHz 1 core: 3.06 GHz Gfx: 766 MHz	1.20 GHz	rPGA	1, 2, 3, 4, 5, 6, 7, 8, 10, 13, 14
SLBPF	i5-540M	C-2/ 20652h/ 0044h/12h	3 MB	Core: 2.53 GHz Gfx: 500 MHz DDR3: 1066/800 MT/s	2 core: 2.80 GHz 1 core: 3.06 GHz Gfx: 766 MHz	1.20 GHz	BGA	1, 2, 3, 4, 5, 6, 7, 8, 10, 13, 14
SLBNB	i5-520M	C-2/ 20652h/ 0044h/12h	3 MB	Core: 2.40 GHz Gfx: 500 MHz DDR3: 1066/800 MT/s	2 core: 2.66 GHz 1 core: 2.93 GHz Gfx: 766 MHz	1.20 GHz	rPGA	1, 2, 3, 4, 5, 6, 7, 8, 10, 13, 14
SLBNA	i5-520M	C-2/ 20652h/ 0044h/12h	3 MB	Core: 2.40 GHz Gfx: 500 MHz DDR3: 1066/800 MT/s	2 core: 2.66 GHz 1 core: 2.93 GHz Gfx: 766 MHz	1.20 GHz	BGA	1, 2, 3, 4, 5, 6, 7, 8, 10, 13, 14
SLBPN	i5-430M	C-2/ 20652h/ 0044h/12h	3 MB	Core: 2.26 GHz Gfx: 500 MHz DDR3: 1066/800 MT/s	2 core: 2.53 GHz 1 core: 2.53 GHz Gfx: 766 MHz	1.20 GHz	rPGA	1,3, 6,8,10, 13, 14
SLBPM	i5-430M	C-2/ 20652h/ 0044h/12h	3 MB	Core: 2.26 GHz Gfx: 500 MHz DDR3: 1066/800 MT/s	2 core: 2.53 GHz 1 core: 2.53 GHz Gfx: 766 MHz	1.20 GHz	BGA	1,3,6,8,10, 13, 14
SLBPK	i3-350M	C-2/ 20652h/ 0044h/12h	3 MB	Core: 2.26 GHz Gfx: 500 MHz DDR3: 1066/800 MT/s	2 core: 2.26 GHz 1 core: 2.26 GHz Gfx: 667 MHz	933 MHz	rPGA	1,3,6,9,10, 13, 14
SLBPL	i3-350M	C-2/ 20652h/ 0044h/12h	3 MB	Core: 2.26 GHz Gfx: 500 MHz DDR3: 1066/800 MT/s	2 core: 2.26 GHz 1 core: 2.26 GHz Gfx: 667 MHz	933 MHz	BGA	1,3,6,9,10, 13, 14
SLBMD	i3-330M	C-2/ 20652h/ 0044h/12h	3 MB	Core: 2.13 GHz Gfx: 500 MHz DDR3: 1066/800 MT/s	2 core: 2.13 GHz 1 core: 2.13 GHz Gfx: 667 MHz	933 MHz	rPGA	1,3,6,8,10, 13, 14
SLBNF	i3-330M	C-2/ 20652h/ 0044h/12h	3 MB	Core: 2.13 GHz Gfx: 500 MHz DDR3: 1066/800 MT/s	2 core: 2.13 GHz 1 core: 2.13 GHz Gfx: 667 MHz	933 MHz	BGA	1,3,6,8,10, 13, 14

Table 1. Processor Identification (Sheet 2 of 4)

S-Spec Number	Processor Number	Stepping/ Processor Signature/ Host Device ID/ Host Revision ID	L3 Cache (MB)	Frequency Core Base (GHz) Graphics Base (MHz) DDR3 (MT/s)	Max Intel® Turbo Boost Technology Frequency Single Core Turbo Dual Core Turbo Graphics Turbo	LFM Frequency	Package	Notes
SLBMK	i7-640LM	C-2/ 20652h/ 0044h/12h	4 MB	Core: 2.13 GHz Gfx: 266 MHz DDR3: 1066/800 MT/s	2 core: 2.66 GHz 1 core: 2.93 GHz Gfx: 566 MHz	1.20 GHz	BGA	1, 2, 3, 4, 5, 6, 7, 8, 11, 13, 14
SLBML	i7-620LM	C-2/ 20652h/ 0044h/12h	4 MB	Core: 2.00 GHz Gfx: 266 MHz DDR3: 1066/800 MT/s	2 core: 2.53 GHz 1 core: 2.80 GHz Gfx: 566 MHz	1.20 GHz	BGA	1, 2, 3, 4, 5, 6, 7, 8, 11, 13, 14
SLBMM	i7-640UM	C-2/ 20652h/ 0044h/12h	4 MB	Core: 1.20 GHz Gfx: 166 MHz DDR3: 800 MT/s	2 core: 1.86 GHz 1 core: 2.26 GHz Gfx: 500 MHz	667 MHz	BGA	1, 2, 3, 4, 5, 6, 7, 8, 12, 13, 14
SLBMN	i7-620UM	C-2/ 20652h/ 0044h/12h	4 MB	Core: 1.06 GHz Gfx: 166 MHz DDR3: 800 MT/s	2 core: 1.73 GHz 1 core: 2.13 GHz Gfx: 500 MHz	667 MHz	BGA	1, 2, 3, 4, 5, 6, 7, 8, 12, 13, 14
SLBQP	i5-520UM	C-2/ 20652h/ 0044h/12h	3 MB	Core: 1.06 GHz Gfx: 166 MHz DDR3: 800 MT/s	2 core: 1.60 GHz 1 core: 1.86 GHz Gfx: 500 MHz	667 MHz	BGA	1, 2, 3, 4, 5, 6, 7, 8, 12, 13, 14
SLBSV	i7-640LM	K-0/ 20655h 0044h/18h	4M	Core: 2.13 GHz Gfx: 266 MHz DDR3: 1066/800 MT/s	2 core: 2.66 GHz 1 core: 2.93 GHz Gfx: 566 MHz	1.20 GHz	BGA	1, 2, 3, 4, 5, 6, 7, 8, 11, 13, 14
SLBSU	i7-620LM	K-0/ 20655h 0044h/18h	4M	Core: 2.00 GHz Gfx: 266 MHz DDR3: 1066/800 MT/s	2 core: 2.53 GHz 1 core: 2.80 GHz Gfx: 566 MHz	1.20 GHz	BGA	1, 2, 3, 4, 5, 6, 7, 8, 11, 13, 14
SLBSR	i7-640UM	K-0/ 20655h 0044h/18h	4M	Core: 1.20 GHz Gfx: 166 MHz DDR3: 800 MT/s	2 core: 1.86 GHz 1 core: 2.26 GHz Gfx: 500 MHz	667 MHz	BGA	1, 3, 6, 7, 8, 12, 13, 14
SLBSX	i7-620UM	K-0/ 20655h 0044h/18h	4M	Core: 1.06 GHz Gfx: 166 MHz DDR3: 800 MT/s	2 core: 1.73 GHz 1 core: 2.13 GHz Gfx: 500 MHz	667 MHz	BGA	1, 3, 6, 7, 8, 12, 13, 14
SLBSQ	i5-520UM	K-0/ 20655h 0044h/18h	3M	Core: 1.06 GHz Gfx: 166 MHz DDR3: 800 MT/s	2 core: 1.60 GHz 1 core: 1.86 GHz Gfx: 500 MHz	667 MHz	BGA	1, 3, 6, 7, 8, 12, 13, 14
SLBSS	i7-660UM	K-0/ 20655h 0044h/18h	4M	Core: 1.33 GHz Gfx: 166 MHz DDR3: 800 MT/s	2 core: 2.0 GHz 1 core: 2.4 GHz Gfx: 500 MHz	667 MHz	BGA	1, 3, 6, 7, 8, 12, 13, 14
SLBUJ	i5-540UM	K-0/ 20655h 0044h/18h	3M	Core: 1.2 GHz Gfx: 166 MHz DDR3: 800 MT/s	2 core: 1.73 GHz 1 core: 2.0 GHz Gfx: 500 MHz	667 MHz	BGA	1, 3, 6, 7, 8, 12, 13, 14
SLBVS	i5-430UM	K-0/ 20655h 0044h/18h	3M	Core: 1.2 GHz Gfx: 166 MHz DDR3: 800 MT/s	2 core: 1.46 GHz 1 core: 1.73 GHz Gfx: 500 MHz	667 MHz	BGA	1, 3, 6, 7, 8, 12, 13, 14

Table 1. Processor Identification (Sheet 3 of 4)

S-Spec Number	Processor Number	Stepping / Processor Signature / Host Device ID / Host Revision ID	L3 Cache (MB)	Frequency: Core Base (GHz) Graphics Base (MHz) DDR3 (MT/s)	Max Intel® Turbo Boost Technology Frequency: Single Core Turbo Dual Core Turbo Graphics Turbo	LFM Frequency	Package	Notes
SLBUG	i3-330UM	K-0/ 20655h 0044h/18h	3M	Core: 1.33 GHz Gfx: 166 MHz DDR3: 800 MT/s	2 core: 1.2 GHz 1 core: 1.2 GHz Gfx: 500 MHz	667 MHz	BGA	1, 3, 6, 7, 8, 12, 13, 14
SLBSW	i7-660LM	K-0/ 20655h 0044h/18h	4MB	Core 2.26 GHz Gfx: 266 MHz DDR3: 1066/800 MT/s	2 core: 2.80 GHz 1 core: 3.06 GHz Gfx: 566 MHz	1.20 GHz	BGA	1, 2, 3,4, 5, 6, 7, 8, 11, 13, 14, 15
SLBST	i7-680UM	K-0/ 20655h 0044h/18h	4M	Core: 1.46 GHz Gfx: 166 MHz DDR3: 800 MT/s	2 core: 2.13 GHz 1 core: 2.53 GHz Gfx: 500 MHz	667 MHz	BGA	1, 2, 3,4, 5, 6, 7, 8, 12, 13, 14, 15
SLBSN	i5-560UM	K-0/ 20655h 0044h/18h	3M	Core 1.33 GHz Gfx: 166 MHz DDR3: 800 MT/s	2 core: 1.86 GHz 1 core:2.13 GHz Gfx: 500 MHz	667 MHz	BGA	1, 2, 4, 5, 6, 7, 8, 12, 13, 14, 15
SLBTQ	i7-620M	K-0/ 20655h 0044h/18h	4M	Core: 2.66 GHz Gfx: 500 MHz DDR3: 1066/800 MT/s	2 core: 3.06 GHz 1 core: 3.33 GHz Gfx: 766 MHz	1.20 GHz	rPGA	1, 2, 3, 4, 5, 6, 7, 8, 10, 13, 14
SLBXP	i5-470UM	K-0/ 20655h 0044h/18h	3M	Core: 1.33 GHz Gfx: 166 MHz DDR3: 800 MT/s	2 core: 1.6 GHz 1 core: 1.86 GHz Gfx: 500 MHz	667 MHz	BGA	1,3, 6, 8, 12, 13, 14, 15
SLBSL	i3-380UM	K-0/ 20655h 0044h/18h	3M	Core: 1.33 GHz Gfx: 166 MHz DDR3: 800 MT/s	2 core: N/A GHz 1 core: N/A GHz Gfx: 500 MHz	667 MHz	BGA	1, 3, 6, 8, 12, 13, 14, 15
SLBTZ	i5-450UM	K-0/ 20655h 0044h/18h	3M	Core: 2.40 GHz Gfx: 500 MHz DDR3: 1066/800 MT/s	2 core: 2.66 GHz 1 core: 2.66 GHz Gfx: 766 MHz	1.20 GHz	PGA	1,3, 6, 8, 10, 13, 14
SLBUK	i3-370M	K-0/ 20655h 0044h/18h	3M	Core: 2.40 GHz Gfx: 500 MHz DDR3: 1066/800 MT/s	2 core: NA 1 core: NA Gfx: 667 MHz	1.20 GHz	BGA	1,3,6,8, 10, 13, 14
SLBZU	i7-640M	K-0/ 20655h 0044h/18h	4M	Core: 2.8 GHz Gfx: 500 MHz DDR3: 1066/800 MT/s	2 core: 3.20 GHz 1 core: 3.46 GHz Gfx: 766 MHz	1.20 GHz	BGA	1, 2, 3, 4, 5, 6, 7, 8, 10, 13, 14, 16, 17
SLBTN	i7-640M	K-0/ 20655h 0044h/18h	4M	Core: 2.8 GHz Gfx: 500 MHz DDR3: 1066/800 MT/s	2 core: 3.20 GHz 1 core: 3.46 GHz Gfx: 766 MHz	1.20 GHz	PGA	1, 2, 3, 4, 5, 6, 7, 8, 10, 13, 14, 15
SLBZX	i3-380M	K-0/ 20655h 0044h/18h	3M	Core: 2.53 GHz Gfx: 500 MHz DDR3: 1066/800 MT/s	2 core: NA 1 core: NA Gfx: 667 MHz	933 MHz	PGA	1,3,6,8, 10, 13

Table 1. Processor Identification (Sheet 4 of 4)

S-Spec Number	Processor Number	Stepping / Processor Signature / Host Device ID / Host Revision ID	L3 Cache (MB)	Frequency	Max Intel® Turbo Boost Technology Frequency	LFM Frequency	Package	Notes
				Core Base (GHz) Graphics Base (MHz) DDR3 (MT/s)	Single Core Turbo Dual Core Turbo Graphics Turbo			
SLBZZ	i3-380M	K-0/ 20655h 0044h/18h	3M	Core: 2.53 GHz Gfx: 500 MHz DDR3: 1066/800 MT/s	2 core: NA 1 core: NA Gfx: 667 MHz	933 MHz	BGA	1,3,6,8, 10, 13
SLC27	i5-480M	K-0/ 20655h 0044h/18h	3M	Core: 2.66 GHz Gfx: 500 MHz DDR3: 1066/800 MT/s	2 core: 2.93 GHz 1 core: 2.93 GHz Gfx: 766 MHz	1.20 GHz	PGA	1,3,6,8, 10, 13, 14
SLC26	i5-480M	K-0/ 20655h 0044h/18h	3M	Core: 2.66 GHz Gfx: 500 MHz DDR3: 1066/800 MT/s	2 core: 2.93 GHz 1 core: 2.93 GHz Gfx: 766 MHz	1.20 GHz	BGA	1,3,6,8, 10, 13
SLC28	i5-580M	K-0/ 20655h 0044h/18h	3M	Core: 2.66 GHz Gfx: 500 MHz DDR3: 1066/800 MT/s	2 core: 3.06 GHz 1 core: 3.33 GHz Gfx: 766 MHz	1.20 GHz	PGA	1,3,6,8, 10, 13
SLC29	i5-580M	K-0/ 20655h 0044h/18h	3M	Core: 2.66 GHz Gfx: 500 MHz DDR3: 1066/800 MT/s	2 core: 3.06 GHz 1 core: 3.33 GHz Gfx: 766 MHz	1.20 GHz	BGA	1,3,6,8, 10, 13
SLC25	i3-390M	K-0/ 20655h 0044h/18h	3M	Core: 2.66 GHz Gfx: 500 MHz DDR3: 1066/800 MT/s	2 core: NA 1 core: NA Gfx: 667 MHz	933 MHz	PGA	1,3,6,8, 10, 13, 14
SLC24	i3-390M	K-0/ 20655h 0044h/18h	3M	Core: 2.66 GHz Gfx: 500 MHz DDR3: 1066/800 MT/s	2 core: NA 1 core: NA Gfx: 667 MHz	933 MHz	BGA	1,3,6,8, 10, 13, 14

Notes:
1. Intel® Hyper-Threading Technology enabled.
2. Intel® Trusted Execution Technology (Intel® TXT) 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® AESNI enabled.
6. Intel SSE4.1, 4.2 enabled.
7. Intel® vPro™ technology capable
8. Core Tjmax = 105°C, Graphics Tjmax = 100°C
9. Core Tjmax = 90°C, Graphics Tjmax = 85°C
10. Standard voltage with 35-W TDP
11. Low voltage with 25-W TDP
12. Ultra low voltage with 18-W TDP
13. The core frequency reported in the processor brand string is rounded to 2 decimal digits. (For example, core frequency of 2.6666, repeating 6, is reported as @2.67 in brand string. Core frequency of 2.5333, is reported as @2.53 in brand string.)
14. This part supports C1, C1E, C3 and C6
15.
16. In some applications, the Core i7-640M BGA may perform lower than Core i7-640M PGA and core i7-620M BGA because it has a higher power consumption therefore leaves less room for turbo performance
17. Standard voltage with 37-W TDP

Errata

AAT1. 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.

AAT2. 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 Pentium 4, Intel Xeon, 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.

AAT3. 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.

AAT4. 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.

AAT5. Premature Execution of a Load Operation Prior to Exception Handler Invocation

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

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

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

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

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

AAT6. 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.

AAT7. Incorrect Address Computed for Last Byte of FXSAVE/FXRSTOR Image Leads to Partial Memory Update

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

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

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

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

AAT8. 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.

AAT9. 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.

AAT10. 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. Refer to "Procedure Calls for Block-Structured Languages" in *IA-32 Intel® Architecture Software Developer's Manual, Vol. 1, Basic Architecture*, for information on the usage of the ENTER instructions. This erratum is not expected to occur in Ring 3. Faults are usually processed in Ring 0 and stack switch occurs when transferring to Ring 0. Intel has not observed this erratum on any commercially-available software.

Workaround: None identified.

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

AAT11. 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.

AAT12. 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.

AAT13. General Protection (#GP) Fault May Not Be Signaled on Data Segment Limit Violation above 4-G Limit

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

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

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

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

AAT14. 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.

AAT15. 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.

AAT16. 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.

AAT17. Performance Monitoring Events for Read Miss to Level 3 Cache Fill Occupancy Counter May Be Incorrect

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

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

Workaround: None identified.

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

AAT18. A VM Exit on MWAIT May Incorrectly Report the Monitoring Hardware As Armed

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

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

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

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

AAT19. Performance Monitor Event SEGMENT_REG_LOADS Counts Inaccurately

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

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

Workaround: None identified.

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

AAT20. #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.

AAT21. Improper Parity Error Signaled in the IQ Following Reset When a Code Breakpoint Is Set on a #GP Instruction

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

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

Workaround: None identified.

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

AAT22. 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.

AAT23. IA32_MPERF Counter Stops Counting during On-Demand TM1

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

Implication: The incorrect ratio of IA32_APERF/IA32_MPERF can mislead software P-state (performance state) management algorithms under the conditions described above. It is possible for the Operating System to observe higher processor utilization than actual, which could lead the OS into raising the P-state. During Intel TM1 activation, the OS P-state request is irrelevant and while on-demand throttling is enabled, it is expected that the OS will not be changing the P-state. This erratum should result in no practical implication to software.

Workaround: None identified.

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

AAT24. The Memory Controller tTHROT_OPREF Timings May Be Violated during Self-Refresh Entry

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

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

Workaround: None identified.

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

AAT25. Synchronous Reset of IA32_APERF/IA32_MPERF Counters on Overflow Does Not Work

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

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

Workaround: None identified.

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

AAT26. Disabling Thermal Monitor While Processor Is Hot, Then Re-enabling, May Result in Stuck Core Operating Ratio

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

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

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

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

AAT27. Writing the Local Vector Table (LVT) When an Interrupt Is Pending May Cause an Unexpected Interrupt

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

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

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

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

AAT28. xAPIC Timer May Decrement Too Quickly Following an Automatic Reload While in Periodic Mode

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

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

Workaround: None identified.

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

AAT29. 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.

AAT30. Infinite Stream of Interrupts May Occur If an ExtINT Delivery Mode Interrupt Is Received While All Cores Are in C6

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

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

Workaround: None identified.

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

AAT31. Two xAPIC Timer Event Interrupts May Unexpectedly Occur

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

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

Workaround: None identified.

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

AAT32. EOI Transaction May Not Be Sent If Software Enters Core C6 during an Interrupt Service Routine

Problem: If core C6 is entered after the start of an interrupt service routine but before a write to the APIC EOI register, the core may not send an EOI transaction (if needed) and further interrupts from the same priority level or lower may be blocked.

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

Workaround: None identified.

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

AAT33. FREEZE_WHILE_SMM Does Not Prevent Event from Pending PEBS during SMM

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

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

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

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

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

Workaround: None identified.

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

AAT34. 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.

AAT35. DR6 May Contain Incorrect Information When the First Instruction after a MOV SS,r/m or POP SS Is a Store

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

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

Workaround: None identified.

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

AAT36. 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.

AAT37. IA32_PERF_GLOBAL_CTRL MSR May Be Incorrectly Initialized

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

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

Workaround: None identified.

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

AAT38. Performance Monitor Counter MEM_INST_RETIRED.STORES May Count Higher Than Expected

Problem: Performance Monitoring counter MEM_INST_RETIRED.STORES (Event: 0BH, Umask: 02H) is used to track retired instructions which contain a store operation. Due to this erratum, the processor may also count other types of instructions including WRMSR and MFENCE.

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

Workaround: None identified.

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

AAT39. Sleeping Cores May Not Be Woken up on Logical Cluster Mode Broadcast IPI Using Destination Field Instead of Shorthand

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

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

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

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

AAT40. Faulting Executions of FXRSTOR May Update State Inconsistently

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

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

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

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

AAT41. Performance Monitor Event EPT.EPDPE_MISS May Be Counted While EPT Is Disabled

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

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

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

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

AAT42. 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 uncachable and certain instruction fetch timing conditions occur, the system may experience unpredictable behavior.

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

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

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

AAT43. Performance Monitor Counters May Count Incorrectly

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

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

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

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

AAT44. Performance Monitor Event Offcore_response_0 (B7H) Does Not Count NT Stores to Local DRAM Correctly

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

- OFFCORE_RSP_0 MSR bit [14] is set to 1 (LOCAL_DRAM) and bit [7] is set to 1 (OTHER): NT Stores to Local DRAM are not counted when they should have been.
- OFFCORE_RSP_0 MSR bit [9] is set to (OTHER_CORE_HIT_SNOOP) and bit [7] is set to 1 (OTHER): NT Stores to Local DRAM are counted when they should not have been.

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

Workaround: None identified.

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

AAT45. 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.

AAT46. Back to Back Uncorrected Machine Check Errors May Overwrite IA32_MC3_STATUS.MSCOD

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

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

Workaround: None identified.

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

AAT47. Corrected Errors with a Yellow Error Indication May Be Overwritten by Other Corrected Errors

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

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

Workaround: None identified.

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

AAT48. Performance Monitor Events DCACHE_CACHE_LD and DCACHE_CACHE_ST May Overcount

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

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

Workaround: None identified.

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

AAT49. Rapid Core C3/C6 Transitions May Cause Unpredictable System Behavior

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

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

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

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

AAT50. Performance Monitor Events INSTR_RETIRED and MEM_INST_RETIRED May Count Inaccurately

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

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

Workaround: None identified.

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

AAT51. A Page Fault May Not Be Generated When the PS bit Is Set to "1" in a PML4E or PDPTE

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

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

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

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

AAT52. BIST Results May Be Additionally Reported after a GETSEC[WAKEUP] or INIT-SIPI Sequence

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

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

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

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

AAT53. 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.

AAT54. 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.

AAT55. VM Exits Due to EPT Violations Do Not Record Information about Pre-IRET NMI Blocking

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

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

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

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

AAT56. Multiple Performance Monitor Interrupts Are Possible on Overflow of IA32_FIXED_CTR2

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

Implication: Multiple counter overflow interrupts may be unexpectedly generated.

Workaround: None identified.

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

AAT57. LBRs May Not Be Initialized during Power-On Reset of the Processor

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

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

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

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

AAT58. LBR, BTM or BTS Records May Have Incorrect Branch from Information after an Enhanced Intel SpeedStep® Technology Transition, T-states, C1E, or Adaptive Thermal Throttling

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

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

Workaround: None identified.

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

AAT59. VMX-Preemption Timer Does Not Count Down at the Rate Specified

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

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

Workaround: None identified.

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

AAT60. Multiple Performance Monitor Interrupts Are Possible on Overflow of Fixed Counter 0

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

This erratum only occurs when IA32_FIXED_CTR0 overflows and the processor and counter are configured as follows:

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

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

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

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

AAT61. VM Exits Due to LIDT/LGDT/SIDT/SGDT Do Not Report Correct Operand Size

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

Implication: Virtual-machine monitors cannot rely on bit 11 of the VM-exit instruction information field to determine the operand size of the instruction causing the VM exit.

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

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

AAT62. DPRSLPVR Signal May Be Incorrectly Asserted on Transition between Low Power C-states

Problem: On entry to or exit from package C6 states, DPRSLPVR (Deeper Sleep Voltage Regulator) signal may be incorrectly asserted.

Implication: Due to this erratum, platform voltage regulator may shutdown

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.

AAT63. Performance Monitoring Events STORE_BLOCKS.NOT_STA and STORE_BLOCKS.STA May Not Count Events Correctly

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

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

Workaround: None identified.

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

AAT64. 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.

AAT65. <Erratum Removed>.

AAT66. INVLPG Following INVEPT or INVVPID May Fail to Flush All Translations for a Large Page

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

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

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

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

AAT67. 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.

AAT68. 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.

AAT69. 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.

AAT70. Erratum removed as it does not apply to the Intel® Core™ i7-600, i5-500, i5-400 and i3-300 Mobile Processor Series

Problem: NA

Implication: NA

Workaround: NA

Status: NA

AAT71. Delivery of Certain Events Immediately Following a VM Exit May Push a Corrupted RIP onto the Stack

Problem: If any of the following events is delivered immediately following a VM exit to 64-bit mode from outside 64-bit mode, bits 63:32 of the RIP value pushed on the stack may be cleared to 0:

- A non-maskable interrupt (NMI);
- A machine-check exception (#MC);
- A page fault (#PF) during instruction fetch; or
- A general-protection exception (#GP) due to an attempt to decode an instruction whose length is greater than 15 bytes.

Implication: Unexpected behavior may occur due to the incorrect value of the RIP on the stack. Specifically, return from the event handler via IRET may encounter an unexpected page fault or may begin fetching from an unexpected code address.

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.

AAT72. A String Instruction that Re-maps a Page May Encounter an Unexpected Page Fault

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

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

- The paging structures initially specify a valid translation for the page.
- Software modifies the paging structures so that there is no valid translation for the page (e.g., by clearing to 0 the present bit in one of the paging-structure entries used to translate the page).
- An iteration of a string instruction modifies the paging structures so that the translation is again a valid translation for the page (e.g., by setting to 1 the bit that was cleared earlier).
- A later iteration of the same string instruction loads from a linear address on the page.
- Software did not invalidate TLB entries for the page between the first modification of the paging structures and the string instruction. In this case, the load in the later iteration may cause a page fault that indicates that there is no translation for the page (e.g., with bit 0 clear in the page-fault error code, indicating that the fault was caused by a not-present page).

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

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

AAT73. Logical Processor May Use Incorrect VPID after VM Entry That Returns From SMM

Problem: A logical processor in VMX root operation should use VPID 0000H. Due to this erratum, a logical processor may instead use VPID 1FB3H if VMX root operation was entered using a VM entry that returns from SMM.

Implication: After a VM entry that sets the "enable VPID" VM-execution control and that establishes VPID 1FB3H, the logical processor may erroneously use TLB entries that were cached in VMX root operation.

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.

AAT74. MSR_TURBO_RATIO_LIMIT MSR May Return Intel® Turbo Boost Technology Core Ratio Multipliers for Non-Existent Core Configurations

Problem: MSR_TURBO_RATIO_LIMIT MSR (1ADH) is designed to describe the maximum Intel Turbo Boost Technology potential of the processor. On some processors, a non-zero Intel Turbo Boost Technology value will be returned for non-existent core configurations.

Implication: Due to this erratum, software using the MSR_TURBO_RATIO_LIMIT MSR to report Intel Turbo Boost Technology processor capabilities may report erroneous results.

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.

AAT75. PCI Express x16 Port Logs Bad TLP Correctable Error When Receiving a Duplicate TLP

Problem: In the PCI Express 2.0 Specification a receiver should schedule an ACK and discard a duplicate TLP (Transaction Layer Packet) before ending the transaction within the data link layer. In the processor, the PCI Express x16 root port will set the Bad TLP status bit in the Correctable Error Status Register (Bus 0; Device 1 and 6; Function 0; Offset 1D0h; bit 6) in addition to scheduling an ACK and discarding the duplicate TLP. Note: The duplicate packet can be received only as a result of a correctable error in the other end point (Transmitter).

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.

AAT76. PCI Express x16 Root Port Incorrectly NAK's a Nullified TLP

Problem: In the processor, the PCI Express root port may NAK a nullified TLP (Transaction Layer Packet). This behavior is a result of an incorrect DW (Double Word) enable generation on the processors when packets end with EDB (End Bad Symbol). This also occurs only if total TLP length <= 8 DW in which CRC (Cyclic Redundancy Check) check/framing upstream checks will fail. This failure causes a NAK to be unexpectedly generated for TLP's which have packets with inverted CRC and EDB's. The PCI-e specification revision 2.0 states that such cycles should be dropped and no NAK should be generated. The processor should NAK a nullified TLP only when there is a CRC error or a sequence check fail.

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.

AAT77. PCI Express Graphics x16 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 and 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 and 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.

AAT78. Internal Parity Error May Be Incorrectly Signaled during C6 Exit

Problem: In a complex set of internal conditions an internal parity error may occur during a Core C6 exit.

Implication: Due to this erratum, an uncorrected error may be reported and a machine check exception may be triggered.

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.

AAT79. PMIs during Core C6 Transitions May Cause the System to Hang

Problem: If a performance monitoring counter overflows and causes a PMI (Performance Monitoring Interrupt) at the same time that the core enters C6, then this may cause the system to hang.

Implication: Due to this erratum, the processor may hang when a PMI coincides with core C6 entry.

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.

AAT80. 2-MB Page Split Lock Accesses Combined with Complex Internal Events May Cause Unpredictable System Behavior

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

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

Workaround: None identified.

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

AAT81. Extra APIC Timer Interrupt May Occur during a Write to the Divide Configuration Register

Problem: If the APIC timer Divide Configuration Register (Offset 03E0H) is written at the same time that the APIC timer Current Count Register (Offset 0390H) reads 1H, it is possible that the APIC timer will deliver two interrupts.

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

Workaround: Software should reprogram the Divide Configuration Register only when the APIC timer interrupt is disarmed.

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

AAT82. TXT.PUBLIC.KEY Is Not Reliable

Problem: Intel TXT (Intel Trusted Execution Technology) capable processors, the TXT.PUBLIC.KEY value (Intel TXT registers FED3_0400H to FED3_041FH) is not reliable.

Implication: Due to this erratum, the TXT.PUBLIC.KEY value should not be relied on or used for retrieving the hash of the Intel TXT public key for the platform.

Workaround: None identified.

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

AAT83. 8259 Virtual Wire B Mode Interrupt May Be Dropped When It Collides with Interrupt Acknowledge Cycle from the Preceding Interrupt

Problem: If an un-serviced 8259 Virtual Wire B Mode (8259 connected to IOAPIC) External Interrupt is pending in the APIC and a second 8259 Virtual Wire B Mode External Interrupt arrives, the processor may incorrectly drop the second 8259 Virtual Wire B Mode External Interrupt request. This occurs when both the new External Interrupt and Interrupt Acknowledge for the previous External Interrupt arrive at the APIC at the same time.

Implication: Due to this erratum, any further 8259 Virtual Wire B Mode External Interrupts will subsequently be ignored.

Workaround: Do not use 8259 Virtual Wire B mode when using the 8259 to deliver interrupts.

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

AAT84. CPUID Incorrectly Reports a C-State as Available When This State Is Unsupported

Problem: CPUID incorrectly reports a non-zero value in CPUID MONITOR/MWAIT leaf (5H) EDX [19:16] when the processor does not support an MWAIT with a target C-state EAX [7:4] > 3.

Implication: If an MWAIT instruction is executed with a target C-state EAX [7:4] > 3 then 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.

AAT85. The Combination of a Page-Split Lock Access and Data Accesses That Are Split across Cacheline Boundaries May Lead to Processor Livelock

Problem: Under certain complex micro-architectural conditions, the simultaneous occurrence of a page-split lock and several data accesses that are split across cacheline boundaries may lead to processor livelock.

Implication: Due to this erratum, a livelock may occur that can only be terminated by a processor reset. 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.

AAT86. Processor Hangs on Package C6 State Exit

Problem: An internal timing condition in the processor power management logic will result in processor hangs upon a Package C6 state exit.

Implication: Due to this erratum, the processor will hang during Package C6 state exitNone identified.

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

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

AAT87. A Synchronous SMI May Be Delayed

Problem: A synchronous SMI (System Management Interrupt) occurs as a result of an SMI generating I/O Write instruction and should be handled prior to the next instruction executing. Due to this erratum, the processor may not observe the synchronous SMI prior to execution of the next instruction.

Implication: Due to this erratum, instructions after the I/O Write instruction, which triggered the SMI, may be allowed to execute before the SMI handler. Delayed delivery of the SMI may make it difficult for an SMI Handler to determine the source of the SMI. Software that relies on the IO_SMI bit in SMM save state or synchronous SMI behavior may not function as expected.

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

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

AAT88. 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.

AAT89. PCI Express Cards May Not Train to x16 Link Width

Problem: The Maximum Link Width field in the Link Capabilities register (LCAP; Bus 0; Device 1; Function 0; offset 0xAC; bits [9:4]) may limit the width of the PCI Express link to x8, even though the processor may actually be capable of supporting the full x16 width.

Implication: Implication: PCI Express x16 Graphics Cards used in normal operation and PCI Express CLB (Compliance Load Board) Cards used during PCI Express Compliance mode testing may only train to x8 link width.

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

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

AAT90. The APIC Timer Current Count Register May Prematurely Read 0x0 While the Timer Is Still Running

Problem: The APIC Timer Current Counter Register may prematurely read 0x00000000 while the timer is still running. This problem occurs when a core frequency or C-state transition occurs while the APIC timer countdown is in progress.

Implication: Due to this erratum, certain software may incorrectly assess that the APIC timer countdown is complete when it is actually still running. This erratum does not affect the delivery of the timer interrupt.

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.

AAT91. IO_SMI Indication in SMRAM State Save Area May Be Lost

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

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

Workaround: None identified.

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

AAT92. FSW May Be Corrupted If an x87 Store Instruction Causes a Page Fault in VMX Non-Root Operation After a PD Exit

Problem: The X87 FSW (FPU Status Word) may be corrupted if execution of a floating-point store instruction (FST, FSTP, FIST, FISTP, FISTTP) causes a page fault in VMX non-root operation.

Implication: This erratum may result in unexpected behavior of software that uses x87 FPU instructions.

Workaround: None identified.

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

AAT93. CKE May go Low Within tRFC(min) After a PD Exit

Problem: After a refresh command is issued, followed by an early PD(Power Down) Entry and Exit, the CKE (Clock Enable) signal may be asserted low prior to tRFC(min), the Minimum Refresh Cycle timing. This additional instance of CKE being low causes the processor not to meet the JEDEC DDR3 DRAM specification requirement (Section 4.17.4 Power-Down clarifications - Case 3).

Implication: Due to this erratum, the processor may not meet the JEDEC DDR3 DRAM specification requirement that states: "CKE cannot be registered low twice within a tRFC(min) window". Intel has not observed any functional failure due to this erratum.

Workaround: None identified.

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

AAT94. Under Certain Low Temperature Conditions, Some Uncore Performance Monitoring Events May Report Incorrect Results

Problem: Due to this erratum, under certain low operating temperatures, a small number of Last Level Cache and external bus performance monitoring events in the uncore report incorrect counts. This erratum may affect event codes in the ranges 00H to 0CH and 40H to 43H.

Implication: Due to this erratum, the count value for some uncore Performance Monitoring Events may be inaccurate. The degree of under or over counting 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.

AAT95. VM Entry to 64-Bit Mode May Fail if Bits 48 And 47 of Guest RIP Are Different

Problem: VM entry to 64-bit mode should allow any value for bits [47:0] of the RIP field in the guest-state area as long as bits 63:48 are identical. Due to this erratum, such a VM entry may fail if bit 47 of the field has a value different from that of bit 48.

Implication: It is not possible to perform VM entry to a 64-bit guest that has made a transition to a non-canonical instruction pointer.

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.

AAT96. VM Entry Loading an Unusable SS Might Not Set SS.B to 1

Problem: If the unusable bit (bit 16) is 1 in the guest SS (Stack Segment) access-rights field, VM entry should set the B bit (default stack-pointer size) in the SS (stack segment) register to 1. Due to this erratum, VM entry may instead load SS.B from bit 14 of the guest SS access-rights field, potentially clearing SS.B to 0.

Implication: This erratum can affect software only if a far RET instruction is executed after a VM entry that erroneously clears the B bit and only if the following other three conditions are also true: (1) the SS register is not loaded between VM entry and far RET; (2) the far RET instruction is executed in 64-bit mode with an immediate operand; (3) the far RET instruction makes a transition to compatibility mode without changing CPL (Current Privilege Level). Due to the far RET being executed with an immediate operand, an adjustment is made to the stack pointer. Normally, when SS is unusable the SS.B bit is 1 and the adjustment will be to the 32-bit ESP register. Due to this erratum, the adjustment will incorrectly be made to the 16-bit SP register. 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.

AAT97. Accesses to a VMCS May Not Operate Correctly If CR0.CD is Set on Any Logical Processor of a Core

Problem: The VMX (virtual-machine extensions) are controlled by the VMCS (virtual-machine control structure). If CR0.CD is set on any logical processor of a core, operations using the VMCS may not function correctly. Such operations include the VMREAD and VMWRITE instructions as well as VM entries and VM exits.

Implication: If CR0.CD is set on either logical processor in a core, the VMWRITE instruction may not correctly update the VMCS and the VMREAD instruction may not return correct data. VM entries may not load state properly and may not establish VMX controls properly. VM exits may not save or load state properly.

Workaround: VMMs (Virtual-machine monitors) should ensure that CR0.CD is clear on all logical processors of a core before entering VMX operation on any logical processor. Software should not set CR0.CD on a logical processor if any logical processor of the same core is in VMX operation. VMM software should prevent guest software from setting CR0.CD by setting bit 30 in the CR0 guest/host mask field in every VMCS.

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

AAT98. Performance Monitor Events for Hardware Prefetches Which Miss The L1 Data Cache May be Over Counted

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

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

Workaround: None identified.

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

AAT99. Correctable and Uncorrectable Cache Errors May be Reported Until the First Core C6 Transition

Problem: On a subset of processors it is possible that correctable/uncorrectable cache errors may be logged and/or a machine check exception may occur prior to the first core C6 transition. The errors will be logged in IA32_MC5_STATUS MSR (415H) with the MCACOD (Machine Check Architecture Error Code) bits [15:0] indicating a Cache Hierarchy Error of the form 000F 0001 RRRR TTLL.

Implication: Due to this erratum, correctable/uncorrectable cache error may be logged or signaled.

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.

AAT100. VM Exit May Incorrectly Clear IA32_PERF_GLOBAL_CTRL [34:32]

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

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

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

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

AAT101. DTS Temperature Data May Be Incorrect On a Return From the Package C6 Low Power State

Problem: The DTS (Digital Thermal Sensor) temperature value may be incorrect for a small period of time (less than 2ms) after a return from the package C6 low power state.

Implication: The DTS temperature data (including temperatures read by Platform Environment Control Interface) may be reported lower than the actual temperature. Fan speed control or other system functions which are reliant on correct DTS temperature data may behave unpredictably.

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.

AAT102. USB Devices May Not Function Properly With Integrated Graphics While Running Targeted Stress Graphics Workloads With Non-Matching Memory Configurations

Problem: When the integrated graphics engine continuously generates a large stream of writes to system memory, and Intel Flex Memory Technology is enabled, with a different amount of memory in each channel, the memory arbiter may temporarily stop servicing other device-initiated traffic. In some cases this can cause certain USB devices, such as keyboard and mouse, to become unresponsive. Intel has only observed this erratum with targeted stress content. This erratum is not seen when the platform is configured with single channel or dual channel symmetric memory and is not dependent on the memory frequency.

Implication: Due to this erratum, certain USB devices may become unresponsive.

Workaround: None identified.

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

AAT103. VM Entry May Omit Consistency Checks Related to Bit 14 (BS) of the Pending Debug Exception Field in Guest-State Area of the VMCS

Problem: Section "Checks on Guest Non-Register State" of Volume 3B specifies consistency checks that VM entry should perform for bit 14 (BS, indicating a pending single-step exception) of the pending debug exception field in guest-state area of the VMCS. These checks enforce the consistency of that bit with other fields in the guest-state area. Due to this erratum, VM entry may fail to perform these checks.

Implication: A logical processor may enter VMX non-root operation with a pending single-step debug exception that not consistent other register state; this may result in unexpected behavior. Intel has not observed this erratum with any commercially available software.

Workaround: When using VMWRITE to write to a field in the guest-state area, software should ensure that the value written is consistent with the state of other guest-state fields.

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

AAT104. Intel Turbo Boost Technology Ratio Changes May Cause Unpredictable System Behavior

Problem: When Intel Turbo Boost Technology is enabled as determined by the TURBO_MODE_DISABLE bit being "0" in the IA32_MISC_ENABLES MSR (1A0H), the process of locking to new ratio may cause the processor to run with incorrect ratio settings. The result of this erratum may be unpredictable system behavior.

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

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.

AAT105. -

AAT106. Execution of VMPTRLD May Corrupt Memory If Current-VMCS Pointer is Invalid

Problem: If the VMCLEAR instruction is executed with a pointer to the current-VMCS (virtual-machine control structure), the current-VMCS pointer becomes invalid as expected. A subsequent execution of the VMPTRLD (Load Pointer to Virtual-Machine Control Structure) instruction may erroneously overwrite the four bytes at physical address 0000008FH.

Implication: Due to this erratum, the four bytes in system memory at physical address 0000008FH may be corrupted.

Workaround: None identified.

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

AAT107. PerfMon Overflow Status Can Not be Cleared After Certain Conditions Have Occurred

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

Implication: Due to this erratum, software may be unable to clear the PerfMon counter overflow status indication.

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

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

AAT108. 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.

AAT109. L1 Data Cache Errors May be Logged With Level Set to 1 Instead of 0

Problem: 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 01b instead of 00b.

Implication: 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 01b is the L2 Cache.

Workaround: None identified.

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

AAT110. 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.

AAT111. 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.

AAT112. Successive Fixed Counter Overflows May be Discarded

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

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

Workaround: Software can avoid this by:

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

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

AAT113. #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.

AAT114. A Logical Processor May Wake From Shutdown State When Branch-Trace Messages or Branch-Trace Stores Are Enabled

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

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

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

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

AAT115. Task Switch to a TSS With an Inaccessible LDTR Descriptor May Cause Unexpected Faults

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

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

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

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

AAT116. MCIP Bit Not Checked on SENTER or ENTERACCS

Problem: Problem: When an ILP (Initiating Logical Processor) executes GETSEC with either the SENTER or ENTERACCS leaf function, the processor should check the MCIP (Machine Check In Progress) bit in the IA32_MCG_STATUS MSR (17AH) to determine if any machine check exception is being processed. If a machine check is in progress the ILP should generate a general protection exception. Due to this erratum, the general protection exception is not generated.

Implication: If GETSEC is executed with either the SENTER or ENTERACCS leaf function, and a machine check exception is being processed, ILP will enter an authenticated execution mode instead of generating a general protection exception.

Workaround: None identified.

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

AAT117. EOI-Broadcast Suppression May Not Function Properly if Enabled or Disabled While an Interrupt is in Service

Problem: If a processor supports EOI-broadcast suppression, a write to the local APIC's EOI register does not generate a broadcast EOI (even if the interrupt is level-triggered) if bit 12of the local APIC's SVR (Spurious-Interrupt Vector Register) is set at the time of the write. Due to this erratum, the local APIC decides whether to generate a broadcast EOI based on the value that bit 12 of the SVR had at the time at which the most recent interrupt was delivered or the time of the most recent write to the EOI register (whichever is later).

Implication: If software modifies bit 12 of SVR while servicing an interrupt, the next write to the EOI register may not use the new bit value.

Workaround: Software should not modify bit 12 of SVR while servicing a level-triggered interrupt.

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

AAT118. Unexpected Load May Occur on Execution of Certain Opcodes

Problem: If software executes an instruction with an opcode of the form 66 0F 38 8x (where x is in therange 0 to 6), the processor may unexpectedly perform a load operation (the data loaded is not used). The load occurs even if the instruction causes a VM exit or a fault (including an invalid-opcode exception). If the VMXON instruction has been executed successfully, the load is from the physical address in the VMXON pointer plus 408H;otherwise, it is from physical address 407H. The affected opcodes include the INVEPT and INVVPID instructions as well as five invalid opcodes.

Implication: This erratum may cause incorrect side effects if the load accesses a memory-mapped I/O device. Intel has not observed this erratum with any commercially available system.

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.

AAT119. VM Entry Loading an Unusable SS Might Not Clear Bits 3:0 of the SS Base Address

Problem: If the unusable bit (bit 16) is 1 in the guest SS access-rights field, VM entry should clear bits 3:0 of the base address of the SS register. Due to this erratum, VM entry may instead load these bits from the guest SS base-address field, leaving them with a non-zero value.

Implication: Following a VM entry affected by this erratum, the exception caused by an SSE access through SS outside 64-bit mode may use the wrong exception vector (#GP instead of #SS or vice versa, dependent on the linear address being aligned or misaligned). 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.

AAT120. A First Level Data Cache Parity Error May Result in Unexpected Behavior

Problem: A First Level Data Cache PWhen 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. rity Error May Result in 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.

AAT121. A Page Fault May Not be Generated When the PS bit is set to “1” in a PML4E or PDPTE

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

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

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

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

AAT122. An Unexpected Page Fault or EPT Violation May Occur After Another Logical Processor Creates a Valid Translation for a Page

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

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

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

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

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

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

AAT123. Intel® Trusted Execution Technology ACM Revocation

Problem: SINIT ACM i7_QUAD_SINIT_20.BIN or earlier are revoked and will not launch with new processor configuration information.

Implication: Due to this erratum, SINIT ACM i7_QUAD_SINIT_20.BIN and earlier will be revoked.

Workaround: It is possible for the BIOS to contain a workaround for this erratum. All Intel® TXT enabled software must use SINIT ACM i7_QUAD_SINIT_51.BIN or later.

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

AAT124. Intel® Processor Graphics VT-d IOTLB Invalidation Operation May Fail to Complete

Problem: Intel® Processor Graphics IOTLB (Input Output Translation Look-aside Buffer) invalidation operation submitted through IVT (Invalidate IOTLB) bit 63 of the IOTLB_REG (GFXVTBAR offset 0x108) register on the Intel VT-d (Intel Virtualization Technology for Directed I/O) engine may fail to complete if there are ongoing translations active in the graphics device command streamer. As a result of the failed operation the VT-d software could remain indefinitely in a polling loop waiting for the IVT bit to become clear.

Implication: Due to this erratum, the VT-d software may continuously execute code in a polling loop causing the system software to hang.

Workaround: This erratum can be avoided by software supporting VT-d ensuring that the graphics command streamer is idle as a precondition for IOTLB invalidations affecting Intel Processor Graphics.

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

AAT125. The Corrected Error Count Overflow Bit in IA32_ MC0_STATUS is Not Updated When the UC Bit is Set

Problem: After a UC (uncorrected) error is logged in the IA32_MC0_STATUS MSR (401H), corrected errors will continue to be counted in 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 when the UC bit (bit 61) is set to 1.

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.

AAT126. 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.

AAT127. 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.

AAT128. 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.

AAT129. 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.

AAT130. VM Exit May Set IA32_EFER.NXE When IA32_MISC_ENABLE Bit 34 is Set to 1

Problem: When “XD Bit Disable” in the IA32_MISC_ENABLE MSR (1A0H) bit 34 is set to 1, it should not be possible to enable the “execute disable” feature by setting IA32_EFER.NXE. Due to this erratum, a VM exit that occurs with the 1-setting of the “load IA32_EFER” VM-exit control may set IA32_EFER.NXE even if IA32_MISC_ENABLE bit 34 is set to 1. This erratum can occur only if IA32_MISC_ENABLE bit 34 was set by guest software in VMX non-root operation.

Implication: Software in VMX root operation may execute with the “execute disable” feature enabled despite the fact that the feature should be disabled by the IA32_MISC_ENABLE MSR. Intel has not observed this erratum with any commercially available software.

Workaround: A virtual-machine monitor should not allow guest software to write to the IA32_MISC_ENABLE MSR.

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

AAT131. An IRET Instruction That Results in a Task Switch Does Not Serialize The Processor

Problem: An IRET instruction that results in a task switch by returning from a nested task does not serialize the processor (contrary to the Software Developer’s Manual Vol. 3 section

Implication: Software which depends on the serialization property of IRET during task switching may not behave as expected. Intel has not observed this erratum to impact the operation of any commercially available software.

Workaround: None identified. Software can execute an MFENCE instruction immediately prior to the IRET instruction if serialization is needed.

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

§ §

Specification Changes

There are no, new Specification Changes in this Specification Update revision.

§ §

Specification Clarifications

There are no, new Specification Clarifications in this Specification Update revision.

§ §

Documentation Changes

AAT1. 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	-	-	-

§ §