

Intel[®] Xeon[®] Processor 7400 Series

Specification Update

| December 2010



INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL® PRODUCTS. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER, AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY RELATING TO SALE AND/OR USE OF INTEL PRODUCTS, INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT, OR OTHER INTELLECTUAL PROPERTY RIGHT.

Intel products are not intended for use in medical, life saving, life sustaining, critical control or safety systems, or in nuclear facility applications.

Intel may make changes to specifications and product descriptions at any time, without notice.

Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them.

The Intel® Xeon® Processor 7400 Series may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

¹Hyper-Threading Technology requires a computer system with an Intel® processor supporting HT Technology and a Hyper-Threading Technology enabled chipset, BIOS and operating system. Performance will vary depending on the specific hardware and software you use. See http://www.intel.com/products/ht/hyperthreading_more.htm/ for more information including details on which processors support HT Technology.

^Φ Intel® 64 (Formerly Intel® EM64T) requires a computer system with a processor, chipset, BIOS, operating system, device drivers and applications enabled for Intel 64. Processor will not operate (including 32-bit operation) without an Intel 64 enabled BIOS. Performance will vary depending on your hardware and software configurations. See <http://www.intel.com/technology/64bitextensions/> for more information including details on which processors support Intel 64 or consult with your system vendor for more information.

[±]Intel® Virtualization Technology requires a computer system with an enabled Intel® processor, BIOS, virtual machine monitor (VMM) and for some uses, certain platform software enabled for it. Functionality, performance or other benefit will vary depending on hardware and software configurations. Intel Virtualization Technology-enabled BIOS and VMM applications are currently in development.

^Δ Intel processor numbers are not a measure of performance. Processor numbers differentiate features within each processor family, not across different processor families. Over time processor numbers will increment based on changes in clock, speed, cache, FSB, or other features, and increments are not intended to represent proportional or quantitative increases in any particular feature. Current roadmap processor number progression is not necessarily representative of future roadmaps. See www.intel.com/products/processor_number for details.

Enhanced HALT State (C1E) and Enhanced Intel SpeedStep® Technology (EIST) for specified units of this processor available Q4/06. See the Processor Spec Finder at <http://processorfinder.intel.com> or contact your Intel representative for more information.

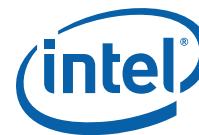
Celeron®, Celeron® D, Celeron® M, Intel® Core™, Intel® Core™ Duo, Intel® Core™ Solo, Intel NetBurst®, Intel® Xeon®, Mobile Intel®, Pentium® III Processor-M, Mobile Intel® Pentium® 4 Processor-M, Pentium® II, Pentium® II Xeon®, Pentium® III, Pentium® III Xeon®, Pentium® 4, Pentium® D, Pentium® M, Pentium® Pro and the Intel® logo are trademarks of Intel Corporation in the U.S. and other countries.

Copies of documents which have an ordering number and are referenced in this document, or other Intel literature may be obtained by calling

1-800-548-4725 or by visiting Intel's website at <http://developer.intel.com/design/litcentr>.

Copyright © 2010, Intel Corporation. All rights reserved.

*Other names and brands may be claimed as the property of others.



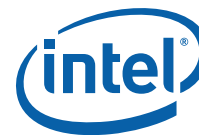
Contents

Revision History	4
Preface	5
Identification Information	7
Package Markings	9
Summary Tables of Changes	10
Errata	16
Specification Changes	36
Specification Clarifications	37
Documentation Changes	38



Revision History

Document Number	Description	Date
320336-001	Initial Release	September 2008
320336-002	Added Specification Clarification 2	September 2008 (out of cycle)
320336-003	Removed Specification Clarification 2 Added AAI61 - AAI65	October 2008
320336-004	Updated AA127	November 2008
320336-005	Updated erratum AA124	March 2009
320336-006	Added AAI66, AAI67, AAI68	July 2009
320336-007	Added AAI69	March 2010
320336-008	Added AAI70	December 2010



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® Xeon® Processor 7400 Series Datasheet	320335

Related Documents

Document Title	Document Number/ Location
<i>AP-485, Intel® Processor Identification and the CPUID Instruction</i>	241618
<i>Intel® 64 and IA-32 Intel® Architectures Software Developer's Manual</i> <ul style="list-style-type: none"> • Volume 1: Basic Architecture • Volume 2A: Instruction Set Reference, A-M • Volume 2B: Instruction Set Reference, N-Z • Volume 3A: System Programming Guide • Volume 3B: System Programming Guide 	253665 253666 253667 253668 253669
<i>Intel® 64 and IA-32 Intel® Architectures Optimization Reference Manual</i>	248966
<i>Intel® 64 and IA-32 Intel® Architectures Software Developer's Manual Documentation Changes</i>	252046
<i>64-bit Extension Technology Software Developer's Guide</i> <ul style="list-style-type: none"> • Volume 1 • Volume 2 	300834 300835

Nomenclature

Errata are design defects or errors. These may cause the Intel® Xeon® Processor 7400 Series's behavior to deviate from published specifications. Hardware and software designed to be used with any given stepping must assume that all errata documented for that stepping are present on all devices.

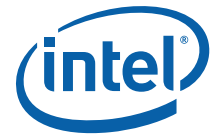
Specification Changes are modifications to the current published specifications. These changes will be incorporated in any new release of the specification.

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.).



Identification Information

The Intel® Xeon® Processor 7400 Series can be identified by the following register contents:

Extended Family ¹	Extended Model ²	Type ³	Family ⁴	Model ⁵	L3 Cache Descriptor ⁶
00000000b	0001b	00	0110b	1101b	0x4B or 0x4C or 0x4D

Notes:

1. The Extended Family corresponds to bits [27:20] of the EDX register after RESET, bits [27:20] 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.
2. The Extended Model corresponds to bits [19:16] of the EDX register after RESET, bits [19:16] 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.
3. The Type corresponds to bits [13:12] of the EDX register after RESET, bits [13:12] 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 Family corresponds to bits [11:8] of the EDX register after RESET, bits [11:8] of the EAX register after the CPUID instruction is executed with a 1 in the EAX register, and the generation field of the Device ID register accessible through Boundary Scan.
5. The Model corresponds to bits [7:4] of the EDX register after RESET, bits [7:4] of the EAX register after the CPUID instruction is executed with a 1 in the EAX register, and the model field of the Device ID register accessible through Boundary Scan.
6. 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 value returned is dependant on the L3 cache size of the processor installed, see [Table 1](#).

Please refer to the *AP-485 Intel® Processor Identification and the CPUID Instruction Application Note* for further information on the CPUID instruction and the software algorithm to distinguish between processors.

Table 1. The Intel® Xeon® Processor 7400 Series Identification Information (Sheet 1 of 2)

S-Spec 2,3,4,5,6,7	CPU Stepping	L2 Cache Size (bytes)	L3 Cache Size (bytes)	CPUID	Core Freq (GHz) ¹	Front Side Bus (MTS)	TDP (W)	Package and Revision
SLG9K	A1	3M x 3	12M	000106D1h	2.4	1066	90	604-pin micro-PGA with 53.3 x 53.3 mm FC-mPGA8 packg Rev 01
SLG9P	A1	3M x 3	16M	000106D1h	2.67	1066	130	604-pin micro-PGA with 53.3 x 53.3 mm FC-mPGA8 packg Rev 01
SLG9J	A1	3M X 2	16M	000106D1h	2.40	1066	90	604-pin micro-PGA with 53.3 x 53.3 mm FC-mPGA8 packg Rev 01
SLG9G	A1	3M X 2	8M	000106D1h	2.13	1066	90	604-pin micro-PGA with 53.3 x 53.3 mm FC-mPGA8 packg Rev 01



Table 1. The Intel® Xeon® Processor 7400 Series Identification Information (Sheet 2 of 2)

S-Spec 2,3,4,5,6,7	CPU Stepping	L2 Cache Size (bytes)	L3 Cache Size (bytes)	CPUID	Core Freq (GHz) ¹	Front Side Bus (MTS)	TDP (W)	Package and Revision
SLG9L	A1	3M X 2	12M	000106D1h	2.13	1066	50	604-pin micro-PGA with 53.3 x 53.3 mm FC-mPGA8 packg Rev 01
SLG9H	A1	3M X 2	12M	000106D1h	2.13	1066	90	604-pin micro-PGA with 53.3 x 53.3 mm FC-mPGA8 packg Rev 01
SLG9M	A1	3M x 3	12M	000106D1h	2.4	1066	65	604-pin micro-PGA with 53.3 x 53.3 mm FC-mPGA8 packg Rev 01

Notes:

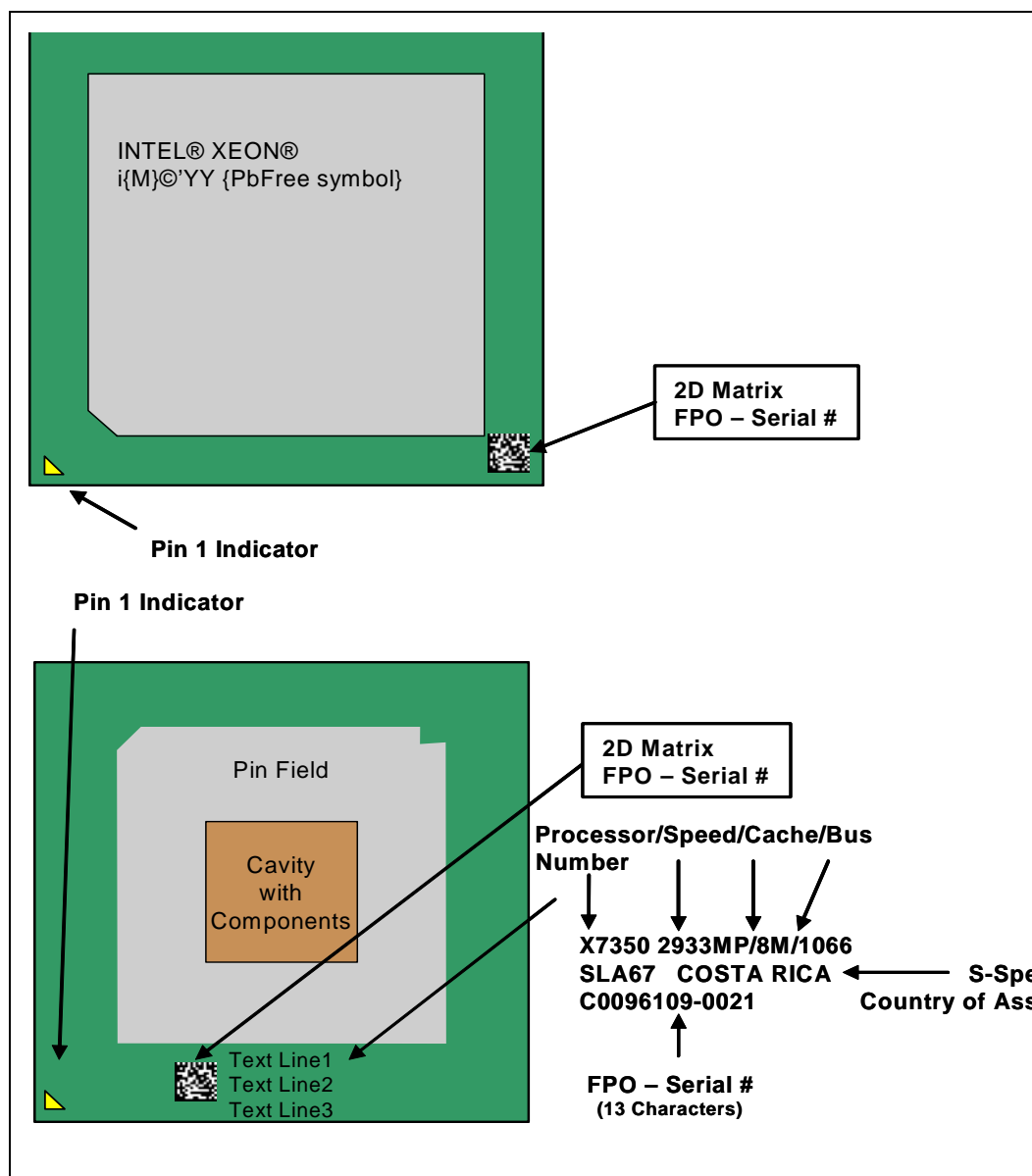
1. Processors with a frequency of 2.13 GHz do not support Intel® SpeedStep Technology, Intel® Thermal Monitor 2 or C1E.
2. The PIROM offset 0Eh to 13h will return a S-SPEC number for sample processors.
3. The PIROM offset 14h will return 00h for sample processors.
4. The string returned from production processors (S Spec) by the CPUID instructions 0x80000002, 0x80000003 and 0x80000004 is "Intel® Xeon® CPU x.xxGHz "
5. The PIROM offset 0Eh to 13h will return a S-Spec number for production processors
6. The PIROM offset 14h will return 01h for production processors.



Package Markings

Intel® Xeon® Processor 7400 Series Package Markings

Figure 1. Processor Topside and Bottomside Markings (Example)



- Notes:**
1. All characters will be in upper case.
 2. Drawing is not to scale



Summary Tables of Changes

The following tables indicate the errata, specification changes, specification clarifications, or documentation changes which apply to the Intel® Xeon® Processor 7400 Series product. 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.

Each Specification Update item will be prefixed with a capital letter(s) to distinguish the product. The key below details the letters that are used in Intel's microprocessor

- A = Dual-Core Intel® Xeon® processor 7000 sequence
- C = Intel® Celeron® processor
- D = Dual-Core Intel® Xeon® processor 2.80 GHz
- E = Intel® Pentium® III processor
- F = Intel® Pentium® processor Extreme Edition and Intel® Pentium® D processor
- I = Dual-Core Intel® Xeon® processor 5000 series
- J = 64-bit Intel® Xeon® processor MP with 1MB L2 cache
- K = Mobile Intel® Pentium® III processor
- L = Intel® Celeron® D processor
- M = Mobile Intel® Celeron® processor



- N = Intel® Pentium® 4 processor
- O = Intel® Xeon® processor MP
- P = Intel® Xeon® processor
- Q = Mobile Intel® Pentium® 4 processor supporting Hyper-Threading technology on 90-nm process technology
- R = Intel® Pentium® 4 processor on 90 nm process
- S = 64-bit Intel® Xeon® processor with 800 MHz system bus (1 MB and 2 MB L2 cache versions)
- T = Mobile Intel® Pentium® 4 processor-M
- U = 64-bit Intel® Xeon® processor MP with up to 8MB L3 cache
- V = Mobile Intel® Celeron® processor on .13 micron process in Micro-FCPGA package
- W = Intel® Celeron® M processor
- X = Intel® Pentium® M processor on 90nm process with 2-MB L2 cache and Intel® processor A100 and A110 with 512-KB L2 cache
- Y = Intel® Pentium® M processor
- Z = Mobile Intel® Pentium® 4 processor with 533 MHz system bus
- AA = Intel® Pentium® D Processor 900 sequence and Intel® Pentium® processor Extreme Edition 955, 965
- AB = Intel® Pentium® 4 Processor 6x1 sequence
- AC = Intel® Celeron® processor in 478 pin package
- AD = Intel® Celeron® D processor on 65nm process
- AE = Intel® Core™ Duo processor and Intel® Core™ Solo processor on 65nm process
- AF = Dual-Core Intel® Xeon® processor LV
- AG = Dual-Core Intel® Xeon® processor 5100 series
- AH = Intel® Core™2 Duo/Solo Processor for Intel® Centrino® Duo Processor Technology'
- AI = Intel® Core™2 Extreme processor X6800^d and Intel® Core™2 Duo desktop processor E6000^d and E4000^d sequence
- AJ = Quad-Core Intel® Xeon® processor 5300 series
- AK = Intel® Core™2 Extreme quad-core processor QX6000^d sequence and Intel® Core™2 Quad processor Q6000^d sequence
- AL = Dual-Core Intel® Xeon® processor 7100 series
- AM = Intel® Celeron® processor 400 sequence
- AN = Intel® Pentium® dual-core processor
- AO = Quad-Core Intel® Xeon® processor 3200 series
- AP = Dual-Core Intel® Xeon® processor 3000 series
- AQ = Intel® Pentium® dual-core desktop processor E2000^d sequence
- AR = Intel® Celeron processor 500 series
- AS = Intel® Xeon® Processor 7400 Series



AT = Intel® Celeron® Processor 200 Series

AV = Intel® Core™2 Extreme processor QX9000^d series and Intel® Core™2 Quad processor Q9000^d series

AW = Intel® Core™ 2 Duo processor E8000 series

AX = Quad-Core Intel® Xeon® processor 5400 series

AY = Dual-Core Intel® Xeon® processor 5200 series

AZ = Intel® Core™2 Duo Processor and Intel® Core™2 Extreme Processor on 45-nm Process

AAA = Quad-Core Intel® Xeon® processor 3300 series

AAB = Dual-Core Intel® Xeon® E3110 Processor

AAC = Intel® Celeron® dual-core processor E1000 series

AAI = Intel® Xeon® Processor 7400 Series

The Specification Updates for the Pentium® processor, Pentium® Pro processor, and other Intel products do not use this convention.



Errata (Sheet 1 of 3)

Number	Stepping	Status	ERRATA
	A-1		
AAI1	X	No Fix	An xTPR Update Transaction Cycle, if Enabled, May be Issued to the FSB after the Processor has Issued a Stop-Grant Special Cycle
AAI2	X	No Fix	LER MSRs May be Incorrectly Updated
AAI3	X	No Fix	Premature Execution of a Load Operation Prior to Exception Handler Invocation
AAI4	X	No Fix	Performance Monitoring Events for Retired Instructions (COH) May Not Be Accurate
AAI5	X	No Fix	Upper 32 bits of 'From' Address Reported through BTMs or BTSs May be Incorrect
AAI6	X	No Fix	General Protection (#GP) Fault May Not Be Signaled on Data Segment Limit Violation above 4-G Limit
AAI7	X	No Fix	Writing Shared Unaligned Data that Crosses a Cache Line without Proper Semaphores or Barriers May Expose a Memory Ordering Issue
AAI8	X	No Fix	Code Segment Limit/Canonical Faults on RSM May be Serviced before Higher Priority Interrupts/Exceptions
AAI9	X	No Fix	Address Reported by Machine-Check Architecture (MCA) on Single-bit L2 ECC Errors May be Incorrect
AAI10	X	No Fix	VERW/VERR/LSL/LAR Instructions May Unexpectedly Update the Last Exception Record (LER) MSR
AAI11	X	No Fix	Pending x87 FPU Exceptions (#MF) Following STI May Be Serviced Before Higher Priority Interrupts
AAI12	X	No Fix	The Processor May Report a #TS Instead of a #GP Fault
AAI13	X	No Fix	A Write to an APIC Register Sometimes May Appear to Have Not Occurred
AAI14	X	No Fix	Programming the Digital Thermal Sensor (DTS) Threshold May Cause Unexpected Thermal Interrupts
AAI15	X	No Fix	Code Segment limit violation may occur on 4 Gigabyte limit check
AAI16	X	No Fix	Incorrect Address Computed For Last Byte of FXSAVE/FXRSTOR Image Leads to Partial Memory Update
AAI17	X	No Fix	CMPSB, LODSB, or SCASB in 64-bit Mode with Count Greater or Equal to 2 ⁴⁸ May Terminate Early
AAI18	X	No Fix	Update of Read/Write (R/W) or User/Supervisor (U/S) or Present (P) Bits without TLB Shutdown May Cause Unexpected Processor Behavior
AAI19	X	No Fix	INIT Does Not Clear Global Entries in the TLB
AAI20	X	No Fix	Last Branch Records (LBR) Updates May be Incorrect after a Task Switch
AAI21	X	No Fix	Code Breakpoint May Be Taken after POP SS Instruction if it is followed by an Instruction that Faults
AAI22	X	No Fix	IRET under Certain Conditions May Cause an Unexpected Alignment Check Exception
AAI23	X	No Fix	REP MOVSB/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
AAI24	X	No Fix	EFLAGS Discrepancy on a Page Fault After a Multiprocessor TLB Shutdown
AAI25	X	No Fix	Returning to Real Mode from SMM with EFLAGS.VM Set May Result in Unpredictable System Behavior
AAI26	X	No Fix	An Asynchronous MCE During a Far Transfer May Corrupt ESP
AAI27	X	No Fix	B0-B3 Bits in DR6 May Not be Properly Cleared After Code Breakpoint
AAI28	X	No Fix	Store to WT Memory Data May be Seen in Wrong Order by Two Subsequent Loads
AAI29	X	No Fix	Non-Temporal Data Store May be Observed in Wrong Program Order



Errata (Sheet 2 of 3)

Number	Stepping	Status	ERRATA
	A-1		
AAI30	X	No Fix	INVLPG Operation for Large (2M/4M) Pages May be Incomplete under Certain Conditions
AAI31	X	No Fix	Page Access Bit May be Set Prior to Signaling a Code Segment Limit Fault
AAI32	X	No Fix	EFLAGS, CR0, CR4 and the EXF4 Signal May be Incorrect after Shutdown
AAI33	X	No Fix	Storage of PEBS Record Delayed Following Execution of MOV SS or STI
AAI34	X	No Fix	Store Ordering May be Incorrect between WC and WP Memory Types
AAI35	X	No Fix	Updating Code Page Directory Attributes without TLB Invalidation May Result in Improper Handling of Code #PF
AAI36	X	No Fix	Performance Monitoring Event MISALIGN_MEM_REF May Over Count
AAI37	X	No Fix	A REP STOS/MOVS to a MONITOR/MWAIT Address Range May Prevent Triggering of the Monitoring Hardware
AAI38	X	No Fix	Performance Monitoring Event FP_MMX_TRANS_TO_MMX May Not Count Some Transitions
AAI39	X	No Fix	Instruction Fetch May Cause a Livelock During Snoops of the L1 Data Cache
AAI40	X	No Fix	Use of Memory Aliasing with Inconsistent Memory Type may Cause a System Hang or a Machine Check Exception
AAI41	X	No Fix	A WB Store Following a REP STOS/MOVS or FXSAVE May Lead to Memory-Ordering Violations
AAI42	X	No Fix	Using Memory Type Aliasing with cacheable and WC Memory Types May Lead to Memory Ordering Violations
AAI43	X	No Fix	Benign Exception after a Double Fault May Not Cause a Triple Fault Shutdown
AAI44	X	No Fix	VM Exit Caused by a SIPI Results in Zero Being Saved to the Guest RIP Field in the VMCS
AAI45	X	No Fix	IO_SMI Indication in SMRAM State Save Area May be Set Incorrectly
AAI46	X	No Fix	Split Locked Stores May not Trigger the Monitoring Hardware
AAI47	X	No Fix	IA32_MC1_STATUS MSR Bit[60] Does Not Reflect Machine Check Error Reporting Enable Correctly
AAI48	X	No Fix	A VM Exit Due to a Fault While Delivering a Software Interrupt May Save Incorrect Data into the VMCS
AAI49	X	No Fix	A VM Exit Occurring in IA-32e Mode May Not Produce a VMX Abort When Expected
AAI50	X	No Fix	VM Exit with Exit Reason "TPR Below Threshold" Can Cause the Blocking by MOV/POP SS and Blocking by STI Bits to be Cleared in the Guest Interruptibility-State Field
AAI51	X	No Fix	NMIs May Not Be Blocked by a VM-Entry Failure
AAI52	X	No Fix	Self/Cross Modifying Code May Not be Detected or May Cause a Machine Check Exception
AAI53	X	No Fix	Data TLB Eviction Condition in the Middle of a Cacheline Split Load Operation May Cause the Processor to Hang
AAI54	X	No Fix	RSM Instruction Execution under Certain Conditions May Cause Processor Hang or Unexpected Instruction Execution Results
AAI55	X	No Fix	Short Nested Loops That Span Multiple 16-Byte Boundaries May Cause a Machine Check Exception or a System Hang
AAI56	X	No Fix	CPUID Returns Incorrect Information Regarding TM2 Support on the Intel® Xeon® E7420 Processor and Intel® Xeon® Processor L7400 Series
AAI57	X	No Fix	RDPNC Instruction Does Not Clear EDX for Fast Reads



Errata (Sheet 3 of 3)

Number	Stepping	Status	ERRATA
	A-1		
AAI58	X	No Fix	CPUID Reporting Non-Power of 2 for Some Processor Relationship Values
AAI59	X	No Fix	An BWL.INVLD Transaction may be Issued to the FSB after the Processor has Issued a Stop-Grant Special Cycle
AAI60	X	No Fix	VM Entry May Fail When Attempting to Set IA32_DEBUGCTL.FREEZE_WHILE_SMM_EN
AAI61			Removed - Duplicate of AAI63
AAI62	X	No Fix	LBR, BTS, BTM May Report a Wrong Address when an Exception/Interrupt Occurs in 64-bit Mode
AAI63	X	No Fix	Memory Ordering Violation With Stores/Loads Crossing a Cacheline Boundary
AAI64	X	No Fix	Corruption of CS Segment Register During RSM While Transitioning From Real Mode to Protected Mode
AAI65	X	No Fix	MONITOR/MWAIT May Have Excessive False Wakeups
AAI66	X	No Fix	A Page Fault May Not be Generated When the PS bit is set to "1" in a PML4E or PDPTE
AAI67	X	No Fix	Not-Present Page Faults May Set the RSVD Flag in the Error Code
AAI68	X	No Fix	VM Exits Due to "NMI-Window Exiting" May Be Delayed by One Instruction
AAI69	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
AAI70	X	No Fix	A 64-bit Register IP-relative Instruction May Return Unexpected Results

Specification Changes

No.	SPECIFICATION CHANGES
AAI1	Implementation of System Management Range Registers

Specification Clarifications

No.	SPECIFICATION CLARIFICATIONS
AAI1	Clarification of TRANSLATION LOOKASIDE BUFFERS (TLBS) Invalidation

Documentation Changes

No.	DOCUMENTATION CHANGES
	None for this revision of this specification update



Errata

AAI 1. An xTPR Update Transaction Cycle, if Enabled, May be Issued to the FSB after the Processor has Issued a Stop-Grant Special Cycle

Problem: According to the FSB (Front Side Bus) protocol specification, no FSB cycles should be issued by the processor once a Stop-Grant special cycle has been issued to the bus. If xTPR update transactions are enabled by clearing the IA32_MISC_ENABLES[bit 23] at the time of Stop-Clock assertion, an xTPR update transaction cycle may be issued to the FSB after the processor has issued a Stop Grant Acknowledge transaction.

Implication: When this erratum occurs in systems using C-states C2 (Stop-Grant State) and higher the result could be a system hang. N/A

Workaround: BIOS must leave the xTPR update transactions disabled (default).

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

AAI 2. LER MSRs May be Incorrectly Updated

Problem: The LER (Last Exception Record) MSRs, MSR_LER_FROM_LIP (1DDH) and MSR_LER_TO_LIP (1DEH) may contain incorrect values after any of the following: • Either STPCLK#, NMI (NonMaskable Interrupt) or external interrupts •CMP or TEST instructions with an uncacheable memory operand followed by a conditional jump • STI/POP SS/MOV SS instructions followed by CMP or TEST instructions and then by a conditional jump.

Implication: The value of the LER MSR may be incorrectly updated to point to a SIMD Floating-Point instruction even though no exception occurred on that instruction or to point to an instruction that was preceded by a StopCik interrupt or rarely not to be updated on Interrupts (NMI and INT).

Workaround: None Identified.

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

AAI 3. Premature Execution of a Load Operation Prior to Exception Handler Invocation

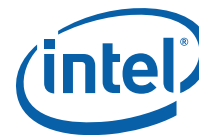
Problem: If any of the below circumstances occur, it is possible that the load portion of the instruction will have executed before the exception handler is entered. If an instruction that performs a memory load causes a code segment limit violation.

- 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 CRO.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 uncache 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 CRO.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 affected steppings, see the Summary Tables of Changes



AAI4. Performance Monitoring Events for Retired Instructions (COH) May Not Be Accurate

- Problem:** The INST_RETIRE performance monitor may miscount retired instructions as follows:
- Repeat string and repeat I/O operations are not counted when a hardware interrupt is received during or after the last iteration of the repeat flow
 - VMLAUNCH and VMRESUME instructions are not counted.
 - 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
 - HLT and MWAIT instructions are not counted. The following instructions, if executed during HLT or MWAIT events, are also not counted:
 - a) RSM from a C-state SMI during an MWAIT instruction.
 - b) RSM from an SMI during a HLT instruction.

Implication: There may be a smaller than expected value in the INST_RETIRE performance monitoring counter. The extent to which this value is smaller than expected is determined by the frequency of the above cases.

Workaround: None identified.

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

AAI5. Upper 32 bits of 'From' Address Reported through BTMs or BTSs May be Incorrect

Problem: When a far transfer switches the processor from 32-bit mode to IA-32e mode, the upper 32 bits of the 'From' (source) addresses reported through the BTMs (Branch Trace Messages) or BTSs (Branch Trace Stores) may be incorrect.

Implication: The upper 32 bits of the 'From' address debug information reported through BTMs or BTSs may be incorrect during this transition.

Workaround: None identified

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

AAI6. 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 4G limit (0fffffffh) may not signal a #GP fault.

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

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

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

AAI7. Writing Shared Unaligned Data that Crosses a Cache Line without Proper Semaphores or Barriers May Expose a Memory Ordering Issue

Problem: Software which is written so that multiple agents can modify the same shared unaligned memory location at the same time may experience a memory ordering issue if multiple loads access this shared data shortly thereafter. Exposure to this problem requires the use of a data write which spans a cache line boundary.

Implication: This erratum may cause loads to be observed out of order. Intel has not observed this erratum with any commercially available software or system.

Workaround: Software should ensure that at least one of the following is true when modifying shared data by multiple agents:

- The shared data is aligned



- Proper semaphores or barriers are used in order to prevent concurrent data accesses.

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

AAI 8. Code Segment Limit/Canonical Faults on RSM May be Serviced before Higher Priority Interrupts/Exceptions

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.)

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 affected steppings, see the Summary Tables of Changes.

AAI 9. Address Reported by Machine-Check Architecture (MCA) on Single-bit L2 ECC Errors May be Incorrect

Problem: When correctable Single-bit ECC errors occur in the L2 cache, the address is logged in the MCA address register (MCI_ADDR). Under some scenarios, the address reported may be incorrect.

Implication: Software should not rely on the value reported in MCI_ADDR, for Single-bit L2 ECC errors.

Workaround: None Identified.

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

AAI 10. VERW/VERR/LSL/LAR Instructions May Unexpectedly Update the Last Exception Record (LER) MSR

Problem: The LER MSR may be unexpectedly updated, if the resultant value of the Zero Flag (ZF) is zero after executing the following instructions

- VERR (ZF=0 indicates unsuccessful segment read verification)
- VERW (ZF=0 indicates unsuccessful segment write verification)
- LAR (ZF=0 indicates unsuccessful access rights load)
- LSL (ZF=0 indicates unsuccessful segment limit load)

Implication: The value of the LER MSR may be inaccurate if VERW/VERR/LSL/LAR instructions are executed after the occurrence of an exception.

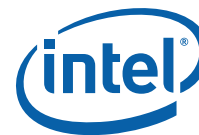
Workaround: Software exception handlers that rely on the LER MSR value should read the LER MSR before executing VERW/VERR/LSL/LAR instructions.

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

AAI 11. Pending x87 FPU Exceptions (#MF) Following STI May Be Serviced Before Higher Priority Interrupts

Problem: Interrupts that are pending prior to the execution of the STI (Set Interrupt Flag) instruction are serviced immediately after the STI instruction is executed. Because of this erratum, if following STI, an instruction that triggers a #MF is executed while STPCLK#, Enhanced Intel SpeedStep® Technology transitions or Thermal Monitor 1 events occur, the pending #MF may be serviced before higher priority interrupts.

Implication: Software may observe #MF being serviced before higher priority interrupts.



Workaround: None Identified.

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

AAI 12. 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 affected steppings, see the Summary Tables of Changes.

AAI 13. A Write to an APIC Register Sometimes May Appear to Have Not Occurred

Problem: With respect to the retirement of instructions, stores to the uncacheable memorybased APIC register space are handled in a non-synchronized way. For example if an instruction that masks the interrupt flag, e.g. CLI, is executed soon after an uncacheable write to the Task Priority Register (TPR) that lowers the APIC priority, the interrupt masking operation may take effect before the actual priority has been lowered. This may cause interrupts whose priority is lower than the initial TPR, but higher than the final TPR, to not be serviced until the interrupt enabled flag is finally set, i.e. by STI instruction. Interrupts will remain pending and are not lost.

Implication: In this example the processor may allow interrupts to be accepted but may delay their service.

Workaround: This non-synchronization can be avoided by issuing an APIC register read after the APIC register write. This will force the store to the APIC register before any subsequent instructions are executed. No commercial operating system is known to be impacted by this erratum.

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

AAI 14. Programming the Digital Thermal Sensor (DTS) Threshold May Cause Unexpected Thermal Interrupts

Problem: Software can enable DTS thermal interrupts by programming the thermal threshold and setting the respective thermal interrupt enable bit. When programming DTS value, the previous DTS threshold may be crossed. This will generate an unexpected thermal interrupt.

Implication: Software may observe an unexpected thermal interrupt occur after reprogramming the thermal threshold.

Workaround: In the ACPI/OS implement a workaround by temporarily disabling the DTS threshold interrupt before updating the DTS threshold value.

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

AAI 15. Code Segment limit violation may occur on 4 Gigabyte limit check

Problem: Code Segment limit violation may occur on 4 Gigabyte limit check when the code stream wraps around in a way that one instruction ends at the last byte of the segment and the next instruction begins at 0x0.

Implication: This is a rare condition that may result in a system hang. Intel has not observed this erratum with any commercially available software, or system.

Workaround: Avoid code that wraps around segment limit.

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

**AAI 16. Incorrect Address Computed For Last Byte of FXSAVE/FXRSTOR Image Leads to Partial Memory Update**

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

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

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

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

AAI 17. CMPSB, LODSB, or SCASB in 64-bit Mode with Count Greater or Equal to 2⁴⁸ May Terminate Early

Problem: In 64-bit Mode CMPSB, LODSB, or SCASB executed with a repeat prefix and count greater than or equal to 248 may terminate early. Early termination may result in one of the following.

- The last iteration not being executed
- Signaling of a canonical limit fault (#GP) on the last iteration

Implication: While in 64-bit mode, with count greater or equal to 248, repeat string operations CMPSB, LODSB or SCASB may terminate without completing the last iteration. Intel has not observed this erratum with any commercially available software.

Workaround: Do not use repeated string operations with RCX greater than or equal to 2⁴⁸.

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

AAI 18. Update of Read/Write (R/W) or User/Supervisor (U/S) or Present (P) Bits without TLB Shutdown May Cause Unexpected Processor Behavior

Problem: Updating a page table entry by changing R/W, U/S or P bits without TLB shutdown (as defined by the 4 step procedure in "Propagation of Page Table and Page Directory Entry Changes to Multiple Processors" In volume 3A of the IA-32 Intel® Architecture Software Developer's Manual), in conjunction with a complex sequence of internal processor micro-architectural events, may lead to unexpected processor behavior.

Implication: This erratum may lead to livelock, shutdown or other unexpected processor behavior. Intel has not observed this erratum with any commercially available system.

Workaround: None Identified.

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

AAI 19. INIT Does Not Clear Global Entries in the TLB

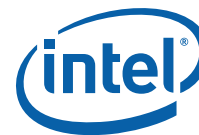
Problem: INIT may not flush a TLB entry when:

- The processor is in protected mode with paging enabled and the page global enable flag is set (PGE bit of CR4 register)
- G bit for the page table entry is set
- TLB entry is present in TLB when INIT occurs

Implication: Software may encounter unexpected page fault or incorrect address translation due to a TLB entry erroneously left in TLB after INIT.

Workaround: Write to CR3, CR4 (setting bits PSE, PGE or PAE) or CR0 (setting bits PG or PE) registers before writing to memory early in BIOS code to clear all the global entries from TLB.

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

**AAI20. Last Branch Records (LBR) Updates May be Incorrect after a Task Switch**

Problem: A Task-State Segment (TSS) task switch may incorrectly set the LBR_FROM value to the LBR_TO value.

Implication: The LBR_FROM will have the incorrect address of the Branch Instruction.

Workaround: None identified.

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

AAI21. Code Breakpoint May Be Taken after POP SS Instruction if it is followed by an Instruction that Faults

Problem: A POP SS instruction should inhibit all interrupts including Code Breakpoints until after execution of the following instruction. This allows sequential execution of POP SS and MOV ESP, EBP instructions without having an invalid stack during interrupt handling. However, a code breakpoint may be taken after POP SS if it is followed by an instruction that faults, this results in a code breakpoint being reported on an unexpected instruction boundary since both instructions should be atomic.

Implication: This can result in a mismatched Stack Segment and SP. 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 "POP SS" in conjunction with "MOV ESP, EBP" will avoid the failure since the "MOV" will not fault.

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

AAI22. 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. I

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 affected steppings, see the Summary Tables of Changes.

AAI23. REP MOVSB/STOSB 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 MOVSB or REP STOSB as fast strings. Due to this erratum fast string REP MOVSB/REP STOSB 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 MOVSB or REP STOSB instruction that will execute with fast strings enabled.

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

AAI 24. EFLAGS Discrepancy on a Page Fault After a Multiprocessor TLB Shutdown

Problem: This erratum may occur when the processor executes one of the following read-modify-write arithmetic instructions and a page fault occurs during the store of the memory operand: ADD, AND, BTC, BTR, BTS, CMPXCHG, DEC, INC, NEG, NOT, OR, ROL/ROR, SAL/SAR/SHL/SHR, SHLD, SHRD, SUB, XOR, and XADD. In this case, the EFLAGS value pushed onto the stack of the page fault handler may reflect the status of the register after the instruction would have completed execution rather than before it. The following conditions are required for the store to generate a page fault and call the operating system page fault handler:

- The store address entry must be evicted from the DTLB by speculative loads from other instructions that hit the same way of the DTLB before the store has completed. DTLB eviction requires at least three-load operations that have linear address bits 15:12 equal to each other and address bits 31:16 different from each other in close physical proximity to the arithmetic operation.
- The page table entry for the store address must have its permissions tightened during the very small window of time between the DTLB eviction and execution of Another processor, without corresponding synchronization and TLB flush, must cause the permission change. Examples of page permission tightening include from Present to Not Present or from Read/Write to Read Only, etc.
- Another processor, without corresponding synchronization and TLB flush, must cause the permission change.

Implication: This scenario may only occur on a multiprocessor platform running an operating system that performs "lazy" TLB shutdowns. The memory image of the EFLAGS register on the page fault handler's stack prematurely contains the final arithmetic flag values although the instruction has not yet completed. Intel has not identified any operating systems that inspect the arithmetic portion of the EFLAGS register during a page fault nor observed this erratum in laboratory testing of software applications.

Workaround: No workaround is needed upon normal restart of the instruction, since this erratum is transparent to the faulting code and results in correct instruction behavior. Operating systems may ensure that no processor is currently accessing a page that is scheduled to have its page permissions tightened or have a page fault handler that ignores any incorrect state.

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

AAI 25. Returning to Real Mode from SMM with EFLAGS.VM Set May Result in Unpredictable System Behavior

Problem: Returning back from SMM mode into real mode while EFLAGS.VM is set in SMRAM may result in unpredictable system behavior.

Implication: If SMM software changes the values of the EFLAGS.VM in SMRAM, it may result in unpredictable system behavior. Intel has not observed this behavior in commercially available software.

Workaround: SMM software should not change the value of EFLAGS.VM in SMRAM.

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

AAI 26. An Asynchronous MCE During a Far Transfer May Corrupt ESP

Problem: If an asynchronous machine check occurs during an interrupt, call through gate, FAR RET or IRET and in the presence of certain internal conditions, ESP may be corrupted.



Implication: If the MCE (Machine Check Exception) handler is called without a stack switch, then a triple fault will occur due to the corrupted stack pointer, resulting in a processor shutdown. If the MCE is called with a stack switch, e.g. when the CPL (Current Privilege Level) was changed or when going through an interrupt task gate, then the corrupted ESP will be saved on the new stack or in the TSS (Task State Segment), and will not be used.

Workaround: Use an interrupt task gate for the machine check handler.

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

AAI27. B0-B3 Bits in DR6 May Not be Properly Cleared After Code Breakpoint

Problem: B0-B3 bits (breakpoint conditions detect flags, bits [3:0]) in DR6 may not be properly cleared when the following sequence happens

- POP instruction to SS (Stack Segment) selector;
- Next instruction is FP (Floating Point) that gets FP assist followed by code breakpoint.

Implication: B0-B3 bits in DR6 may not be properly cleared.

Workaround: None identified.

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

AAI28. Store to WT Memory Data May be Seen in Wrong Order by Two Subsequent Loads

Problem: When data of Store to WT memory is used by two subsequent loads of one thread and another thread performs cacheable write to the same address the first load may get the data from external memory or L2 written by another core, while the second load will get the data straight from the WT Store.

Implication: Software that uses WB to WT memory aliasing may violate proper store ordering.

Workaround: Do not use WB to WT aliasing.

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

AAI29. Non-Temporal Data Store May be Observed in Wrong Program Order

Problem: When non-temporal data is accessed by multiple read operations in one thread while another thread performs a cacheable write operation to the same address, the data stored may be observed in wrong program order (i.e. later load operations may read older data).

Implication: Software that uses non-temporal data without proper serialization before accessing the non-temporal data may observe data in wrong program order.

Workaround: Software that conforms to the Intel® 64 and IA-32 Architectures Software Developer's Manual, Volume 3A, section "Buffering of Write Combining Memory Locations" will operate correctly.

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

AAI30. INVLPG Operation for Large (2M/4M) Pages May be Incomplete under Certain Conditions

Problem: The INVLPG instruction may not completely invalidate Translation Look-aside Buffer (TLB) entries for large pages (2M/4M) when both of the following conditions exist:

- Address range of the page being invalidated spans several Memory Type Range Registers (MTRRs) with different memory types specified
- INVLPG operation is preceded by a Page Assist Event (Page Fault (#PF) or an access that results in either A or D bits being set in a Page Table Entry (PTE))



Implication: Stale translations may remain valid in TLB after a PTE update resulting in unpredictable system behavior. Intel has not observed this erratum with any commercially available software.

Workaround: Software should ensure that the memory type specified in the MTRRs is the same for the entire address range of the large page.

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

AAI 31. Page Access Bit May be Set Prior to Signaling a Code Segment Limit Fault

Problem: If code segment limit is set close to the end of a code page, then due to this erratum the memory page Access bit (A bit) may be set for the subsequent page prior to general protection fault on code segment limit.

Implication: When this erratum occurs, a non-accessed page which is present in memory and follows a page that contains the code segment limit may be tagged as accessed.

Workaround: Erratum can be avoided by placing a guard page (non-present or non-executable page) as the last page of the segment or after the page that includes the code segment limit.

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

AAI 32. EFLAGS, CR0, CR4 and the EXF4 Signal May be Incorrect after Shutdown

Problem: When the processor is going into shutdown due to an RSM inconsistency failure, EFLAGS, CR0 and CR4 may be incorrect. In addition the EXF4 signal may still be asserted. This may be observed if the processor is taken out of shutdown by NMI#.

Implication: A processor that has been taken out of shutdown may have an incorrect EFLAGS, CR0 and CR4. In addition the EXF4 signal may still be asserted.

Workaround: None identified.

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

AAI 33. 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 affected steppings, see the Summary Tables of Changes.

AAI 34. Store Ordering May be Incorrect between WC and WP Memory Types

Problem: According to Intel® 64 and IA-32 Intel Architecture Software Developer's Manual, Volume 3A "Methods of Caching Available", WP (Write Protected) stores should drain the WC (Write Combining) buffers in the same way as UC (Uncacheable) memory type stores do. Due to this erratum, WP stores may not drain the WC buffers.

Implication: Memory ordering may be violated between WC and WP stores.

Workaround: None identified.

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

**AAI 35. Updating Code Page Directory Attributes without TLB Invalidation May Result in Improper Handling of Code #PF**

Problem: Code #PF (Page Fault exception) is normally handled in lower priority order relative to both code #DB (Debug Exception) and code Segment Limit Violation #GP (General Protection Fault). Due to this erratum, code #PF may be handled incorrectly, if all of the following conditions are met:

- A PDE (Page Directory Entry) is modified without invalidating the corresponding TLB (Translation Look-aside Buffer) entry
- Code execution transitions to a different code page such that both
 - The target linear address corresponds to the modified PDE
 - The PTE (Page Table Entry) for the target linear address has an A (Accessed) bit that is clear
- One of the following simultaneous exception conditions is present following the code transition
 - Code #DB and code #PF
 - Code Segment Limit Violation #GP and code #PF

Implication: Software may observe either incorrect processing of code #PF before code Segment Limit Violation #GP or processing of code #PF in lieu of code #DB.

Workaround: None identified.

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

AAI 36. Performance Monitoring Event MISALIGN_MEM_REF May Over Count

Problem: Performance monitoring event MISALIGN_MEM_REF (05H) is used to count the number of memory accesses that cross an 8-byte boundary and are blocked until retirement. Due to this erratum, the performance monitoring event MISALIGN_MEM_REF also counts other memory accesses.

Implication: The performance monitoring event MISALIGN_MEM_REF may over count. The extent of over counting depends on the number of memory accesses retiring while the counter is active.

Workaround: None identified.

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

AAI 37. A REP STOS/MOVS to a MONITOR/MWAIT Address Range May Prevent Triggering of the Monitoring Hardware

Problem: The MONITOR instruction is used to arm the address monitoring hardware for the subsequent MWAIT instruction. The hardware is triggered on subsequent memory store operations to the monitored address range. Due to this erratum, REP STOS/MOVS fast string operations to the monitored address range may prevent the actual triggering store to be propagated to the monitoring hardware.

Implication: A logical processor executing an MWAIT instruction may not immediately continue program execution if a REP STOS/MOVS targets the monitored address range.

Workaround: Software can avoid this erratum by not using REP STOS/MOVS store operations within the monitored address range.

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

AAI 38. Performance Monitoring Event FP_MMX_TRANS_TO_MMX May Not Count Some Transitions

Problem: Performance Monitor Event FP_MMX_TRANS_TO_MMX (Event CCH, Umask 01H) counts transitions from x87 Floating Point (FP) to MMX™ instructions. Due to this erratum, if



only a small number of MMX instructions (including EMMS) are executed immediately after the last FP instruction, a FP to MMX transition may not be counted.

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

Workaround: None identified.

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

AAI 39. Instruction Fetch May Cause a Livelock During Snoops of the L1 Data Cache

Problem: A livelock may be observed in rare conditions when instruction fetch causes multiple level one data cache snoops.

Implication: Due to this erratum, a livelock may occur. Intel has not observed this erratum with any commercially available software.

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

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

AAI 40. Use of Memory Aliasing with Inconsistent Memory Type may Cause a System Hang or a Machine Check Exception

Problem: Software that implements memory aliasing by having more than one linear addresses mapped to the same physical page with different cache types may cause the system to hang or to report a machine check exception (MCE). This would occur if one of the addresses is non-cacheable and used in a code segment and the other is a cacheable address. If the cacheable address finds its way into the instruction cache, and the non-cacheable address is fetched in the IFU, the processor may invalidate the non-cacheable address from the fetch unit. Any micro-architectural event that causes instruction restart will be expecting this instruction to still be in the fetch unit and lack of it will cause a system hang or an MCE.

Implication: This erratum has not been observed with commercially available software.

Workaround: Although it is possible to have a single physical page mapped by two different linear addresses with different memory types, Intel has strongly discouraged this practice as it may lead to undefined results. Software that needs to implement memory aliasing should manage the memory type consistency.

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

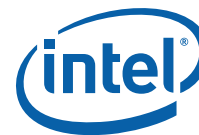
AAI 41. A WB Store Following a REP STOS/MOVS or FXSAVE May Lead to Memory-Ordering Violations

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 may perform REP MOVS or REP STOS as write combining stores (referred to as "fast strings") for optimal performance. FXSAVE may also be internally implemented using write combining stores. Due to this erratum, stores of a WB (write back) memory type to a cache line previously written by a preceding fast string/FXSAVE instruction may be observed before string/FXSAVE stores.

Implication: A write-back store may be observed before a previous string or FXSAVE related store. Intel has not observed this erratum with any commercially available software.

Workaround: Software desiring strict ordering of string/FXSAVE operations relative to subsequent write-back stores should add an MFENCE or SFENCE instruction between the string/FXSAVE operation and following store-order sensitive code such as that used for synchronization.

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

**AAI42. Using Memory Type Aliasing with cacheable and WC Memory Types May Lead to Memory Ordering Violations**

Problem: Memory type aliasing occurs when a single physical page is mapped to two or more different linear addresses, each with different memory types. Memory type aliasing with a cacheable memory type and WC (write combining) may cause the processor to perform incorrect operations leading to memory ordering violations for WC operations.

Implication: Software that uses aliasing between cacheable and WC memory types may observe memory ordering errors within WC memory operations. Intel has not observed this erratum with any commercially available software.

Workaround: None identified. Intel does not support the use of cacheable and WC memory type aliasing, and WC operations are defined as weakly ordered.

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

AAI43. Benign Exception after a Double Fault May Not Cause a Triple Fault Shutdown

Problem: According to the Intel® 64 and IA-32 Architectures Software Developer's Manual, Volume 3A, "Exception and Interrupt Reference", if another exception occurs while attempting to call the double-fault handler, the processor enters shutdown mode. Due to this erratum, any benign faults while attempting to call double-fault handler will not cause a shutdown. However Contributory Exceptions and Page Faults will continue to cause a triple fault shutdown.

Implication: If a benign exception occurs while attempting to call the double-fault handler, the processor may hang or may handle the benign exception. Intel has not observed this erratum with any commercially available software.

Workaround: None identified.

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

AAI44. VM Exit Caused by a SIPI Results in Zero Being Saved to the Guest RIP Field in the VMCS

Problem: If a logical processor is in VMX non-root operation and in the wait-for-SIPI state, an occurrence of a start-up IPI (SIPI) causes a VM exit. Due to this erratum, such VM exits always save zero into the RIP field of the guest-state area of the virtual-machine control structure (VMCS) instead of the value of RIP before the SIPI was received.

Implication: In the absence of virtualization, a SIPI received by a logical processor in the wait-for-SIPI state results in the logical processor starting execution from the vector sent in the SIPI regardless of the value of RIP before the SIPI was received. A virtual-machine monitor (VMM) responding to a SIPI-induced VM exit can emulate this behavior because the SIPI vector is saved in the lower 8 bits of the exit qualification field in the VMCS. Such a VMM should be unaffected by this erratum. A VMM that does not emulate this behavior may need to recover the old value of RIP through alternative means. Intel has not observed this erratum with any commercially available software.

Workaround: VMM software that may respond to SIPI-induced VM exits by resuming the interrupt guest context without emulating the non-virtualized SIPI response should (1) save from the VMCS (using VMREAD) the value of RIP before any VM entry to the wait-for-SIPI state; and (2) restore to the VMCS (using VMWRITE) that value before the next VM entry that resumes the guest in any state other than wait-for-SIPI.

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

AAI45. IO_SMI Indication in SMRAM State Save Area May be Set Incorrectly

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

Implication: A PDE (Page Directory Entry) is modified without invalidating the corresponding TLB (Translation Look-aside Buffer) entry



- SMI is pending while a lower priority event interrupts
- A REP I/O read
- A I/O read that redirects to MWAIT
- In systems supporting Intel® Virtualization Technology a fault in the middle of an IO operation that causes a VM Exit

Implication: SMM handlers may get false IO_SMI indication.

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

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

AAI 46. Split Locked Stores May not Trigger the Monitoring Hardware

Problem: Logical processors normally resume program execution following the MWAIT, when another logical processor performs a write access to a WB cacheable address within the address range used to perform the MONITOR operation. Due to this erratum, a logical processor may not resume execution until the next targeted interrupt event or O/S timer tick following a locked store that spans across cache lines within the monitored address range.

Implication: The logical processor that executed the MWAIT instruction may not resume execution until the next targeted interrupt event or O/S timer tick in the case where the monitored address is written by a locked store which is split across cache lines.

Workaround: Do not use locked stores that span cache lines in the monitored address range.

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

AAI 47. IA32_MC1_STATUS MSR Bit[60] Does Not Reflect Machine Check Error Reporting Enable Correctly

Problem: IA32_MC1_STATUS MSR (405H) bit[60] (EN- Error Enabled) is supposed to indicate whether the enable bit in the IA32_MC1_CTL MSR (404H) was set at the time of the last update to the IA32_MC1_STATUS MSR. Due to this erratum, IA32_MC1_STATUS MSR bit[60] instead reports the current value of the IA32_MC1_CTL MSR enable bit.

Implication: IA32_MC1_STATUS MSR bit [60] may not reflect the correct state of the enable bit in the IA32_MC1_CTL MSR at the time of the last update.

Workaround: None identified.

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

AAI 48. A VM Exit Due to a Fault While Delivering a Software Interrupt May Save Incorrect Data into the VMCS

Problem: If a fault occurs during delivery of a software interrupt (INTn) in virtual-8086 mode when virtual mode extensions are in effect and that fault causes a VM exit, incorrect data may be saved into the VMCS. Specifically, information about the software interrupt may not be reported in the IDT-vectoring information field. In addition, the interruptibility-state field may indicate blocking by STI or by MOV SS if such blocking were in effect before execution of the INTn instruction or before execution of the VM-entry instruction that injected the software interrupt.

Implication: In general, VMM software that follows the guidelines given in the section “Handling VM Exits Due to Exceptions” of Intel® 64 and IA-32 Architectures Software Developer’s Manual Volume 3B: System Programming Guide should not be affected. If the erratum improperly causes indication of blocking by STI or by MOV SS, the ability of a VMM to inject an interrupt may be delayed by one instruction.



Workaround: VMM software should follow the guidelines given in the section “Handling VM Exits Due to Exceptions” of Intel® 64 and IA-32 Architectures Software Developer’s Manual Volume 3B: System Programming Guide.

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

AAI49. A VM Exit Occuring in IA-32e Mode May Not Produce a VMX Abort When Expected

Problem: If a VM exit occurs while the processor is in IA-32e mode and the “host address-space size” VM-exit control is 0, a VMX abort should occur. Due to this erratum, the expected VMX aborts may not occur and instead the VM Exit will occur normally. The conditions required to observe this erratum are a VM entry that returns from SMM with the “IA-32e guest” VM-entry control set to 1 in the SMM VMCS and the “host address-space size” VM-exit control cleared to 0 in the executive VMCS.

Implication: A VM Exit will occur when a VMX Abort was expected.

Workaround: An SMM VMM should always set the “IA-32e guest” VM-entry control in the SMM VMCS to be the value that was in the LMA bit (IA32_EFER.LMA.LMA[bit 10]) in the IA32_EFER MSR (C0000080H) at the time of the last SMM VM exit. If this guideline is followed, that value will be 1 only if the “host address-space size” VM-exit control is 1 in the executive VMCS.

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

AAI50. VM Exit with Exit Reason “TPR Below Threshold” Can Cause the Blocking by MOV/POP SS and Blocking by STI Bits to be Cleared in the Guest Interruptibility-State Field

Problem: As specified in Section, “VM Exits Induced by the TPR Shadow”, in the Intel® 64 and IA-32 Architectures Software Developer’s Manual, Volume 3B, a VM exit occurs immediately after any VM entry performed with the “use TPR shadow”, “activate secondary controls”, and “virtualize APIC accesses” VM-execution controls all set to 1 and with the value of the TPR shadow (bits 7:4 in byte 80H of the virtual-APIC page) less than the TPR-threshold VM-execution control field. Due to this erratum, such a VM exit will clear bit 0 (blocking by STI) and bit 1 (blocking by MOV/POP SS) of the interruptibility-state field of the guest-state area of the VMCS (bit 0 - blocking by STI and bit 1 - blocking by MOV/POP SS should be left unmodified).

Implication: Since the STI, MOV SS, and POP SS instructions cannot modify the TPR shadow, bits 1:0 of the interruptibility-state field will usually be zero before any VM entry meeting the preconditions of this erratum; behavior is correct in this case. However, if VMM software raises the value of the TPR-threshold VM-execution control field above that of the TPR shadow while either of those bits is 1, incorrect behavior may result. This may lead to VMM software prematurely injecting an interrupt into a guest. Intel has not observed this erratum with any commercially available software.

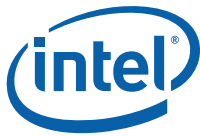
Workaround: VMM software raising the value of the TPR-threshold VM-execution control field should compare it to the TPR shadow. If the threshold value is higher, software should not perform a VM entry; instead, it could perform the actions that it would normally take in response to a VM exit with exit reason “TPR below threshold”.

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

AAI51. NMIs May Not Be Blocked by a VM-Entry Failure

Problem: The Intel® 64 and IA-32 Architectures Software Developer’s Manual Volume 3B: System Programming Guide, Part 2 specifies that, following a VM-entry failure during or after loading guest state, “the state of blocking by NMI is what it was before VM entry.” If non-maskable interrupts (NMIs) are blocked and the “virtual NMIs” VM-execution control set to 1, this erratum may result in NMIs not being blocked after a VM-entry failure during or after loading guest state.

Implication: VM-entry failures that cause NMIs to become unblocked may cause the processor to deliver an NMI to software that is not prepared for it.



Workaround: VMM software should configure the virtual-machine control structure (VMCS) so that VM-entry failures do not occur.

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

AAI 52. Self/Cross Modifying Code May Not be Detected or May Cause a Machine Check Exception

Problem: If instructions from at least three different ways in the same instruction cache set exist in the pipeline combined with some rare internal state, self-modifying code (SMC) or cross-modifying code may not be detected and/or handled.

Implication: An instruction that should be overwritten by another instruction while in the processor pipeline may not be detected/modified, and could retire without detection. Alternatively the instruction may cause a Machine Check Exception. 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 affected steppings, see the Summary Tables of Changes

AAI 53. Data TLB Eviction Condition in the Middle of a Cacheline Split Load Operation May Cause the Processor to Hang

Problem: If the TLB translation gets evicted while completing a cacheline split load operation, under rare scenarios the processor may hang.

Implication: The cacheline split load operation may not be able to complete normally, and the machine may hang and generate Machine Check Exception. 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 affected steppings, see the Summary Tables of Changes

AAI 54. RSM Instruction Execution under Certain Conditions May Cause Processor Hang or Unexpected Instruction Execution Results

Problem: RSM instruction execution, under certain conditions triggered by a complex sequence of internal processor micro-architectural events, may lead to processor hang, or unexpected instruction execution results.

Implication: In the above sequence, the processor may live lock or hang, or RSM instruction may restart the interrupted processor context through a nondeterministic EIP offset in the code segment, resulting in unexpected instruction execution, unexpected exceptions or system hang. 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 affected steppings, see the Summary Tables of Changes

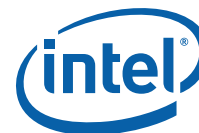
AAI 55. Short Nested Loops That Span Multiple 16-Byte Boundaries May Cause a Machine Check Exception or a System Hang

Problem: Under a rare set of timing conditions and address alignment of instructions in a short nested loop sequence, software that contains multiple conditional jump instructions and spans multiple 16-byte boundaries, may cause a machine check exception or a system hang .

Implication: Due to this erratum, a machine check exception or a system hang may occur.

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

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



AAI56. CPUID Returns Incorrect Information Regarding TM2 Support on the Intel® Xeon® E7420 Processor and Intel® Xeon® Processor L7400 Series

Problem: When CPUID instruction is executed with EAX = 1, feature information is returned in ECX. Bit 8 indicates TM2 (Thermal Monitor2) support. For the Intel® Xeon® E7420 Processor and Intel® Xeon® Processor L7400 Series, the value in these bits should be zero. It is currently returning 1, which indicates TM2 is supported on this SKU.

Implication: The CPUID instruction returns incorrect information regarding TM2 support. In this case, TM2 is disabled but reported incorrectly as enabled.

Workaround: None identified. TM1 (Thermal Monitor1) must be enabled for the Intel® Xeon® E7420 Processor and Intel® Xeon® Processor L7400 Series , to operate with specification conditions.

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

AAI57. RDPMC Instruction Does Not Clear EDX for Fast Reads

Problem: The RDPMC instruction reads performance monitoring counters specified in ECX and returns data in EDX:EAX. For fast reads, RDPMC instruction only reads the lower 32 bits of the Performance Monitoring Counter into the EAX register. EDX is incorrectly not written with zeroes.

Implication: An incorrect EDX value is returned for the Performance Monitoring Counters specified in ECX when fast read RDPMC instruction is used for addresses 80000002H - 8000000DH.

Workaround: Clear EDX after fast reads to addresses 80000002H - 8000000DH performed by the RDPMC instruction.

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

AAI58. CPUID Reporting Non-Power of 2 for Some Processor Relationship Values

Problem: Some software assumes that maximum number of addressable IDs for logical processors in the package (MLPP), maximum number of addressable IDs for processor cores in the package (MCPP), and maximum number of addressable IDs for logical processors sharing a cache (NTSC) will be reported as values that are a power of two. When the reported non-power of 2 values are used (without rounding to a power of 2) to generate masks for determining processor relationships, the resulting incorrect mask values can not correctly identify processor relationships. The incorrect processor relationship information may result in performance issues. When the incorrect mask values are used to determine licensing requirements , they may cause excess licenses to be required. When power of two values are reported these algorithms function correctly. The following table gives the location and values of the fields reported by CPUID that may cause problems for a 6 core processor.

Value	CPUID Inputs		Bit Field Location	Initial Value	Corrected Value
	EAX	ECX			
MLPP	1	x	EBX[23:16]	6	8
MCPP	4	x	EAX[31:26]*	5	7
NTSC	4	3	EAX[25:14]*	5	7

x = Don't care

* For these fields you must add one to the return value to get the result



Implication: Some software incorrectly calculates the masks for determining processor topology when given non-power of two counts. This may lead to performance problems and may cause software that licenses processors to require an excessive number of licensees.

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

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

AAI 59. An BWL.INVLD Transaction may be Issued to the FSB after the Processor has Issued a Stop-Grant Special Cycle

Problem: When the STPCLK# pin is asserted, all the processors enter the Stop-Grant state of the processor and remain in that state until STPCLK# is deasserted. According to the FSB (Front Side Bus) protocol specification, no FSB cycles should be issued by the processor once a Stop-Grant special cycle has been issued to the bus. A BWL.INVLD transaction may be issued to the FSB after the processor has issued a Stop Grant Acknowledge transaction.

Implication: This is a violation of the FSB Protocol.

Workaround: None Identified.

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

AAI 60. VM Entry May Fail When Attempting to Set IA32_DEBUGCTL.FREEZE_WHILE_SMM_EN

Problem: If bit 14 (FREEZE_WHILE_SMM_EN) is set in the IA32_DEBUGCTL field in the guest-state area of the VMCS, VM entry may fail as described in Section “VM-Entry Failures During or After Loading Guest State” of Intel® 64 and IA-32 Architectures Software Developer’s Manual Volume 3B: System Programming Guide, Part 2. (The exit reason will be 80000021H and the exit qualification will be zero.) Note that the FREEZE_WHILE_SMM_EN bit in the guest IA32_DEBUGCTL field may be set due to a VMWRITE to that field or due to a VM exit that occurs while IA32_DEBUGCTL.FREEZE_WHILE_SMM_EN=1.

Implication: A VMM will not be able to properly virtualize a guest using the FREEZE_WHILE_SMM feature.

Workaround: It is possible for the BIOS to contain a workaround for this erratum. Alternatively, the following software workaround may be used. If a VMM wants to use the FREEZE_WHILE_SMM feature, it can configure an entry in the VM-entry MSR-load area for the IA32_DEBUGCTL MSR (1D9H); the value in the entry should set the FREEZE_WHILE_SMM_EN bit. In addition, the VMM should use VMWRITE to clear the FREEZE_WHILE_SMM_EN bit in the guest IA32_DEBUGCTL field before every VM entry. (It is necessary to do this before every VM entry because each VM exit will save that bit as 1.) This workaround prevents the VM-entry failure and sets the FREEZE_WHILE_SMM_EN bit in the IA32_DEBUGCTL MSR.

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

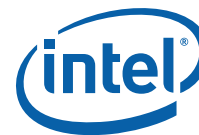
AAI 61. Removed - Suplicate of AAI 63

AAI 62. 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 (0x00007FFFFFFFFF) 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 affected steppings, see the Summary Tables of Changes.

AAI63. Memory Ordering Violation With Stores/Loads Crossing a Cacheline Boundary

Problem: When two logical processors are accessing the same data that is crossing a cacheline boundary without serialization, with a specific set of processor internal conditions, it is possible to have an ordering violation between memory store and load operations.

Implication: Due to this erratum, proper load/store ordering may not be followed when multiple logical processors are accessing the same data that crosses a cacheline boundary without serialization.

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

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

AAI64. 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 affected steppings, see the Summary Tables of Changes

AAI65. MONITOR/MWAIT May Have Excessive False Wakeups

Problem: Normally, if MWAIT is used to enter a C-state that is C1 or higher, a store to the address range armed by the MONITOR instruction will cause the processor to exit MWAIT. Due to this erratum, false wakeups may occur when the monitored address range was recently written prior to executing the MONITOR instruction.

Implication: Due to this erratum, performance and power savings may be impacted due to excessive false wakeups.

Workaround: Execute a CLFLUSH Instruction immediately before every MONITOR instruction when the monitored location may have been recently written.

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

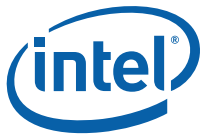
AAI66. A Page Fault May Not be Generated When the PS bit is set to "1" in a PML4E or PDPTE

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 affected steppings, see the Summary Tables of Changes

**AAI 67. AAI 67. Not-Present Page Faults May Set the RSVD Flag in the Error Code**

Problem: An attempt to access a page that is not marked present causes a page fault. Such a page fault delivers an error code in which both the P flag (bit 0) and the RSVD flag (bit 3) are 0. Due to this erratum, not-present page faults may deliver an error code in which the P flag is 0 but the RSVD flag is 1.

Implication: Software may erroneously infer that a page fault was due to a reserved-bit violation when it was actually due to an attempt to access a not-present page. Intel has not observed this erratum with any commercially available software.

Workaround: Page-fault handlers should ignore the RSVD flag in the error code if the P flag is 0.

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

AAI 68. 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 affected steppings, see the Summary Tables of Changes

AAI 69. 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 affected steppings, see the Summary Tables of Changes

AAI 70. A 64-bit Register IP-relative Instruction May Return Unexpected Results

Problem: Under an unlikely and complex sequence of conditions in 64-bit mode, a register IP-relative instruction result may be incorrect.

Implication: A register IP-relative instruction result may be incorrect and could cause software to read from or write to an incorrect memory location. This may result in an unexpected page fault or unpredictable system behavior.

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



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



Specification Changes

The Specification Changes listed in this section apply to the following documents:

Intel® Xeon® Processor 7400 Series Datasheet (Order Number 320335)

AAI 1. Implementation of System Management Range Registers

Intel® Xeon® Processor 7400 Series has implemented SMRRs (System Management Range Registers). SMRRs are defined in Section 10.11.2.4 of the Intel® 64 and IA-32 Architectures Software Developer's Manual, Volume 3A: System Programming Guide.

SMM (System Management Mode) code and data reside in SMRAM. The SMRR interface is an enhancement in Intel® 64 and IA-32 Architectures to limit cacheable reference of addresses in SMRAM to code running in SMM. The SMRR interface can be configured only by code running in SMM.

Under certain circumstances, an attacker who has gained administrative privileges, such as ring 0 privileges in a traditional operating system, may be able to reconfigure an Intel processor to gain access to SMM. The implementation of SMRR mitigates this issue. Intel has provided a recommended update to system and BIOS vendors to incorporate into their BIOS to resolve this issue.



Specification Clarifications

The Specification Clarifications listed in this section apply to the following documents:

Intel® Xeon® Processor 7400 Series Datasheet (Order Number 320335)

All Specification Clarifications will be incorporated into a future version of the appropriate Intel® Xeon® Processor 7400 Series documentation.

Intel processor numbers are not a measure of performance. Processor numbers differentiate features within each processor family, not across different processor families. Over time processor numbers will increment based on changes in clock, speed, cache, FSB, or other features, and increments are not intended to represent proportional or quantitative increases in any particular feature. Current roadmap processor number progression is not necessarily representative of future roadmaps. See http://www.intel.com/products/processor_number for details.

AAI 1. Clarification of TRANSLATION LOOKASIDE BUFFERS (TLBS) Invalidation

Section 10.9 INVALIDATING THE TRANSLATION LOOKASIDE BUFFERS (TLBS) of the Intel® 64 and IA-32 Architectures Software Developer's Manual, Volume 3A: System Programming Guide will be modified to include the presence of page table structure caches, such as the page directory cache, which Intel processors implement. This information is needed to aid operating systems in managing page table structure invalidations properly.

Intel will update the Intel® 64 and IA-32 Architectures Software Developer's Manual, Volume 3A: System Programming Guide in the coming months. Until that time, an application note, *TLBs, Paging-Structure Caches, and Their Invalidation* (<http://www.intel.com/design/processor/applnots/317080.pdf>), is available which provides more information on the paging structure caches and TLB invalidation.

In rare instances, improper TLB invalidation may result in unpredictable system behavior, such as system hangs or incorrect data. Developers of operating systems should take this documentation into account when designing TLB invalidation algorithms.



Documentation Changes

The Documentation Changes listed in this section apply to the following documents:

Intel® Xeon® Processor 7400 Series Datasheet (Order Number 320335)