64-bit Intel® Xeon® Processor with 800 MHz System Bus (1 MB and 2 MB L2 Cache Versions)

Specification Update

August 2009

Notice: The 64-bit Intel® Xeon® processor with 800 MHz system bus (1 MB and 2 MB L2 cache versions) may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Order Number: 302402-024

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.

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

Intel, the Intel logo, Pentium, Pentium III Xeon, Celeron, Intel NetBurst and Intel Xeon are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Copyright © 2004-2009, Intel Corporation. All rights reserved.

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

Contents

Revision History

Version	Description	Date
-001	• Initial release of the document.	July 2004
-002	• Removed erratum P18 and renumbered existing errata.	July 2004
-003	• Added errata S32-S35. Renamed errata numbering from P to S.	August 2004
-004	• Removed erratum S29 and renumbered existing errata. • Added errata S35-S65. • Added E0 step processor information to Table 2-1, “Identification Information”. • Added new notes to Table 2-2, “64-bit Intel® Xeon® Processor with 800 MHz System Bus (1 MB and 2 MB L2 Cache Versions) Identification Information” and deleted unnecessary notes. • Added Table 2-3, “DP Platform Population Matrix for the 64-bit Intel® Xeon® Processor with 800 MHz System Bus (1 MB and 2 MB L2 Cache Versions) FC-PGA4 Package”.	September 2004
-005	• Added errata S66-S73.	October 2004
-006	• Updated S-Spec table, code key, and mixed steppings statement to include Low Voltage Intel® Xeon® processor • Added errata S74-S75.	November 2004
-007	• Updated erratum S26. Added errata S76and S77.	December 2004
-008	• Added errata S78-S79; added additional text to “Mixed Steppings In DP Systems” chapter.	January 2005
-009	• Added 2 MB L2 cache version of the 64-bit Intel® Xeon® processor with 800 MHz system bus; added errata S80-S81; added S-spec numbers SL7ZC, SL7ZD, SL7ZE, and SL7ZF to Table 2-2.	February 2005
-010	• Updated errata S28 and S53; updated summary table entries for S19 and S43.	March 2005
-011	• Updated Table 2-2; added erratum S82; updated plans column for erratum S27; added Specification Clarification 1.	April 2005
-012	• Updated steppings affected for erratum S27.	May 2005
-013	• Updated Table 2-2.	June 2005
-014	• Added erratum S83. Updated relevant document lists.	July 2005
-015	• Updated summary table entry for S19. Added erratum S84.	August 2005
-016	• Added errata S85 and S86. Added S-spec numbers to Table 2-2. Added R-0 stepping to summary table of changes. Updated summary table entries. Removed duplicate erratum S33.	September 2005
-017	• Added erratum S87. Updated letters in *Codes used in Summary Table.* Corrected S-spec notes for 3.40 GHz entries.	October 2005
-018	• Added errata S88 and S89. Updated steppings affected for S22. Updated *Mixed Steppings in DP Systems* section.	November 2005
-019	• Added erratum S90.	December 2005
-020	• Added erratum S91. Updated erratum S17.	January 2006
-021	• Added erratum S1S and S92.	May 2006

-022	• Added erratum S93.	June 2006
-023	• Added erratum S94. Updated Summary of Changes Table.	August 2008
-024	• Updated Summary of Changes table. • Added Documentation Change S1. • Updated links to related documentation throughout this document.	August 2009

Preface 1

This document is an update to the specifications contained in the following documents:

1. *64-bit Intel® Xeon® Processor with 2 MB L2 Cache Datasheet* (Document Number 306249)
   Link: http://www.intel.com/Assets/PDF/datasheet/306249.pdf
2. *Low Voltage Intel® Xeon® Processor with 800 MHz System Bus Datasheet* (Document Number 302355)
   Link: http://www.intel.com/design/intarch/datashts/304097.htm
3. IA-32 Intel® Architecture Software Developer's Manual, Volume 1: Basic Architecture (Document Number 253665)
   Link: http://www.intel.com/products/processor/manuals/
4. IA-32 Intel® Architecture Software Developer's Manual, Volume 2A: Instruction Set Reference, A-M (Document Number 253666)
   Link: http://www.intel.com/products/processor/manuals/
5. IA-32 Intel® Architecture Software Developer's Manual, Volume 2B: Instruction Set Reference, N-Z (Document Number 253667)
   Link: http://www.intel.com/products/processor/manuals/
6. IA-32 Intel® Architecture Software Developer's Manual, Volume 3: System Programming Guide (Document Number 253668)
   Link: http://www.intel.com/products/processor/manuals/

This document is intended for hardware system manufacturers and software developers of applications, operating systems, or tools.

1.1 Nomenclature

S-Spec Number is a five-digit code used to identify products. Products are differentiated by their unique characteristics, e.g., core speed, L2 cache size, package type, etc. as described in the processor identification information table. Care should be taken to read all notes associated with each S-Spec number.

Errata are design defects or errors. Errata may cause the processor's behavior to deviate from published specifications. Hardware and software designed to be used with any given processor must assume that all errata documented for that processor are present on all devices unless otherwise noted.

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

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 the next release of the specifications.

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

Identification Information 2

2.1 64-bit Intel® Xeon® Processor with 800 MHz System Bus (1 MB and 2 MB L2 Cache Versions) Package Markings (604-pin FC-mPGA4 Package)

Figure 2-1. Top-Side Processor Marking Example

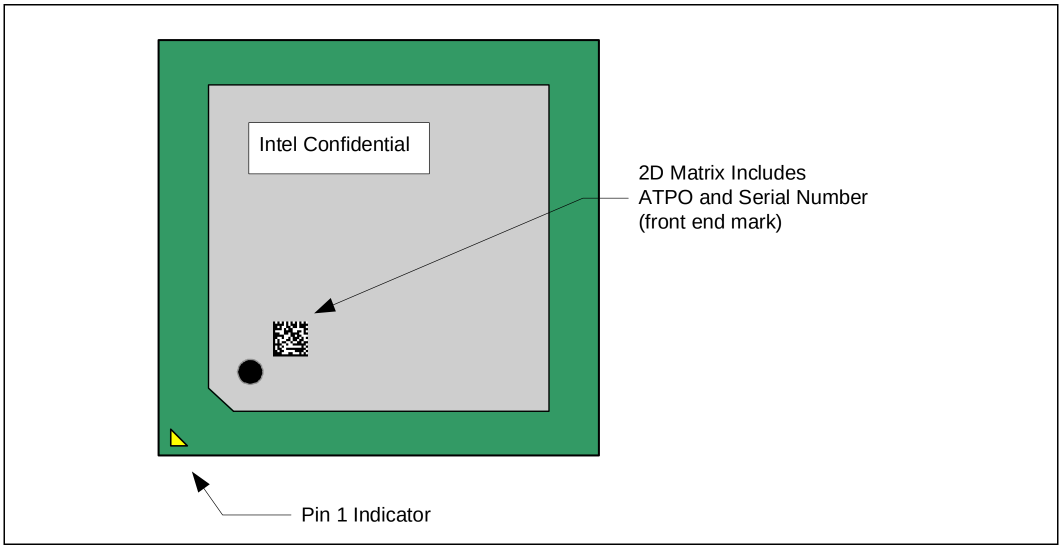
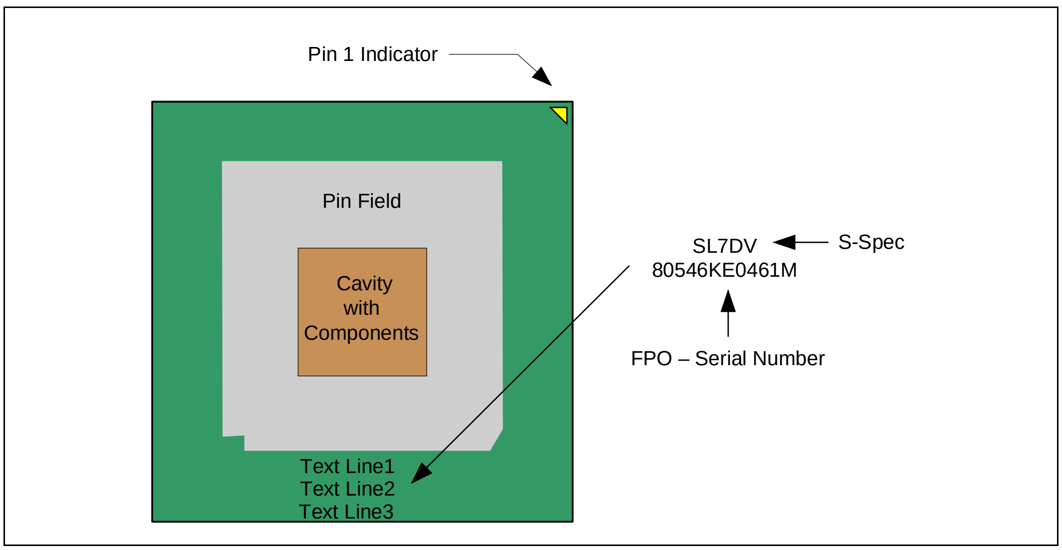


Figure 2-2. Bottom-Side Marking Example



The 64-bit Intel® Xeon® Processor with 800 MHz System Bus (1 MB and 2 MB L2 cache versions) can be identified by the following values:

Table 2-1. Identification Information

Family 1	Model 2	Brand ID 3
1111b	0011b	0000b
1111b	0100b	0000b

1. 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 registers accessible through Boundary Scan
2. 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 registers accessible through Boundary Scan.
3. Brand ID returns 0000b, which means that Brand ID is unsupported in this processor.

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. Please refer to the *Intel Processor Identification and the CPUID Instruction Application Note* (AP-485) for further information on the CPUID instruction.

Table 2-2. 64-bit Intel® Xeon® Processor with 800 MHz System Bus (1 MB and 2 MB L2 Cache Versions) Identification Information (Sheet 1 of 3)

S-Spec	Core Stepping	CPUID	Core Freq (GHz)	Data Bus Freq (MHz)	L2 Cache Size	Processor Package Revision	Package and Revision	Notes
SL7DV	D-0	0F34h	2.80	800	1 MB	01	604-pin micro-PGA with 42.5 x 42.5 mm FC-PGA4 package	
SL7HF								1
SL7DW	D-0	0F34h	3	800	1 MB	01	604-pin micro-PGA with 42.5 x 42.5 mm FC-PGA4 package	
SL7HG								1
SL7DX	D-0	0F34h	3.20	800	1 MB	01	604-pin micro-PGA with 42.5 x 42.5 mm FC-PGA4 package	
SL7HH								1
SL7DY	D-0	0F34h	3.40	800	1 MB	01	604-pin micro-PGA with 42.5 x 42.5 mm FC-PGA4 package	2, 3
SL7HJ								1, 2, 3
SL7DZ	D-0	0F34h	3.60	800	1 MB	01	604-pin micro-PGA with 42.5 x 42.5 mm FC-PGA4 package	2, 3
SL7HK								1, 2, 3

NOTES:
1. These are Intel boxed processors.
2. These parts have Thermal Monitor 2 (TM2) feature enabled. For D-0 stepping, TM2 is enabled on 3.40 GHz and above, but it is NOT supported.
3. These parts are enabled for Enhanced Intel SpeedStep® Technology (EIST).
4. These parts are enabled for Enhanced Halt State (C1E).
5. These parts are LV (low-power) processors.
6. These parts have Execute Disable bit functionality.
7. These parts are MV (mid-power) processors.

Table 2-2. 64-bit Intel® Xeon® Processor with 800 MHz System Bus (1 MB and 2 MB L2 Cache Versions) Identification Information (Sheet 2 of 3)

S-Spec	Core Stepping	CPUID	Core Freq (GHz)	Data Bus Freq (MHz)	L2 Cache Size	Processor Package Revision	Package and Revision	Notes
SL7PD	E-0	0F41h	2.80	800	1 MB	01	604-pin micro-PGA with 42.5 x 42.5 mm FC-PGA4 package	4, 6
SL7TB	E-0	0F41h	2.80	800	1 MB	01	604-pin micro-PGA with 42.5 x 42.5 mm FC-PGA4 package	1, 4, 6
SL7PE	E-0	0F41h	3	800	1 MB	01	604-pin micro-PGA with 42.5 x 42.5 mm FC-PGA4 package	4, 6
SL7TC	E-0	0F41h	3	800	1 MB	01	604-pin micro-PGA with 42.5 x 42.5 mm FC-PGA4 package	1, 4, 6
SL7PF	E-0	0F41h	3.20	800	1 MB	01	604-pin micro-PGA with 42.5 x 42.5 mm FC-PGA4 package	4, 6
SL7TD	E-0	0F41h	3.20	800	1 MB	01	604-pin micro-PGA with 42.5 x 42.5 mm FC-PGA4 package	1, 4, 6
SL7PG	E-0	0F41h	3.40	800	1 MB	01	604-pin micro-PGA with 42.5 x 42.5 mm FC-PGA4 package	3, 4, 6
SL7TE	E-0	0F41h	3.40	800	1 MB	01	604-pin micro-PGA with 42.5 x 42.5 mm FC-PGA4 package	1, 3, 4, 6
SL7PH	E-0	0F41h	3.60	800	1 MB	01	604-pin micro-PGA with 42.5 x 42.5 mm FC-PGA4 package	2, 3, 4, 6
SL7VF	E-0	0F41h	3.60	800	1 MB	01	604-pin micro-PGA with 42.5 x 42.5 mm FC-PGA4 package	1, 2, 3, 4, 6
SL84B	E-0	0F41h	2.80	800	1 MB	01	604-pin micro-PGA with 42.5 x 42.5 mm FC-PGA4 package	4, 5, 6
SL8KN	G-1	0F49h	2.80	800	1 MB	01	604-pin micro-PGA with 42.5 x 42.5 mm FC-PGA4 package	4, 6
SL8KP	G-1	0F49h	3	800	1 MB	01	604-pin micro-PGA with 42.5 x 42.5 mm FC-PGA4 package	4, 6
SL8KQ	G-1	0F49h	3.20	800	1 MB	01	604-pin micro-PGA with 42.5 x 42.5 mm FC-PGA4 package	4, 6
SL8KR	G-1	0F49h	3.40	800	1 MB	01	604-pin micro-PGA with 42.5 x 42.5 mm FC-PGA4 package	3, 4, 6
SL8KS	G-1	0F49h	3.60	800	1 MB	01	604-pin micro-PGA with 42.5 x 42.5 mm FC-PGA4 package	2, 3, 4, 6
SL8RW	G-1	0F49h	2.80	800	1 MB	01	604-pin micro-PGA with 42.5 x 42.5 mm FC-PGA4 package	4, 5, 6
SL7ZC	N-0	0F43h	3.60	800	2 MB	01	604-pin micro-PGA with 42.5 x 42.5 mm FC-PGA4 package	2, 3, 4, 6

NOTES:

1. These are Intel boxed processors.
2. These parts have Thermal Monitor 2 (TM2) feature enabled. For D-0 stepping, TM2 is enabled on 3.40 GHz and above, but it is NOT supported.
3. These parts are enabled for Enhanced Intel SpeedStep® Technology (EIST).
4. These parts are enabled for Enhanced Halt State (C1E).
5. These parts are LV (low-power) processors.
6. These parts have Execute Disable bit functionality.
7. These parts are MV (mid-power) processors.

Table 2-2. 64-bit Intel® Xeon® Processor with 800 MHz System Bus (1 MB and 2 MB L2 Cache Versions) Identification Information (Sheet 3 of 3)

S-Spec	Core Stepping	CPUID	Core Freq (GHz)	Data Bus Freq (MHz)	L2 Cache Size	Processor Package Revision	Package and Revision	Notes
SL7ZD	N-0	0F43h	3.40	800	2 MB	01	604-pin micro-PGA with 42.5 x 42.5 mm FC-PGA4 package	2, 3, 4, 6
SL7ZE	N-0	0F43h	3.20	800	2 MB	01	604-pin micro-PGA with 42.5 x 42.5 mm FC-PGA4 package	2, 4, 6
SL7ZF	N-0	0F43h	3	800	2 MB	01	604-pin micro-PGA with 42.5 x 42.5 mm FC-PGA4 package	2, 4, 6
SL8P7	R-0	0F4Ah	2.80	800	2 MB	01	604-pin micro-PGA with 42.5 x 42.5 mm FC-PGA4 package	2, 3, 4, 6
SL8P6	R-0	0F4Ah	3	800	2 MB	01	604-pin micro-PGA with 42.5 x 42.5 mm FC-PGA4 package	2, 4, 6
SL8P5	R-0	0F4Ah	3.20	800	2 MB	01	604-pin micro-PGA with 42.5 x 42.5 mm FC-PGA4 package	2, 4, 6
SL8P4	R-0	0F4Ah	3.40	800	2 MB	01	604-pin micro-PGA with 42.5 x 42.5 mm FC-PGA4 package	2, 3, 4, 6
SL8P3	R-0	0F4Ah	3.60	800	2 MB	01	604-pin micro-PGA with 42.5 x 42.5 mm FC-PGA4 package	2, 3, 4, 6
SL8P2	R-0	0F4Ah	3.80	800	2 MB	01	604-pin micro-PGA with 42.5 x 42.5 mm FC-PGA4 package	2, 3, 4, 6
SL8SV	R-0	0F4Ah	3	800	2 MB	01	604-pin micro-PGA with 42.5 x 42.5 mm FC-PGA4 package	2, 3, 4, 5, 6
SL8T3	R-0	0F4Ah	3.20	800	2 MB	01	604-pin micro-PGA with 42.5 x 42.5 mm FC-PGA4 package	2, 4, 6, 7

NOTES:
1. These are Intel boxed processors.
2. These parts have Thermal Monitor 2 (TM2) feature enabled. For D-0 stepping, TM2 is enabled on 3.40 GHz and above, but it is NOT supported.
3. These parts are enabled for Enhanced Intel SpeedStep® Technology (EIST).
4. These parts are enabled for Enhanced Halt State (C1E).
5. These parts are LV (low-power) processors.
6. These parts have Execute Disable bit functionality.
7. These parts are MV (mid-power) processors.

2.2 Mixed Steppings in DP Systems

Intel Corporation fully supports mixed steppings of the 64-bit Intel Xeon processor with 800 MHz system bus as well as mixed steppings of the 64-bit Intel Xeon processor with 2 MB L2 cache. The following list and processor matrix describes the requirements to support mixed steppings:

- Mixed steppings are only supported with processors that have identical family numbers as indicated by the CPUID instruction.
- While Intel has done nothing to specifically prevent processors operating at differing frequencies from functioning within a multiprocessor system, there may be uncharacterized errata that exist in such configurations. Intel does not support such configurations. In mixed stepping systems, all processors must operate at identical frequencies (i.e., the highest frequency rating commonly supported by all processors).
- While there are no known issues associated with the mixing of processors with differing cache sizes in a multiprocessor system, and Intel has done nothing to specifically prevent such system configurations from operating, Intel does not support such configurations since there may be uncharacterized errata that exist. In mixed stepping systems, all processors must be of the same cache size.
- While Intel believes that certain customers may wish to perform validation of system configurations with mixed frequencies, cache sizes or voltages and that those efforts are an acceptable option to our customers, customers would be fully responsible for the validation of such configurations.
- Intel requires that the proper microcode update be loaded on each processor operating in a multiprocessor system. Any processor that does not have the proper microcode update loaded is considered by Intel to be operating out of specification.
- The workarounds identified in this and following specification updates must be properly applied to each processor in the system. Certain errata are specific to the multiprocessor environment. Errata for all processor steppings will affect system performance if not properly worked around. Also see Also see Table 2 and Table 3 for additional details on which processors are affected by specific errata.
- In mixed stepping systems, the processor with the lowest feature-set, as determined by the CPUID Feature Bytes, must be the Bootstrap Processor (BSP). In the event of a tie in feature-set, the tie should be resolved by selecting the BSP as the processor with the lowest stepping as determined by the CPUID instruction.
- While there are no known issues associated with the mixing of processors of different power-optimization segments (i.e. LV or MV) in a multiprocessor system, and Intel has done nothing to specifically prevent such system configurations from operating, Intel does not support such configurations since there may be uncharacterized errata that exist. In mixed stepping systems, all processors must be of the same power-optimization segment.

Table 2-3. DP Platform Population Matrix for the 64-bit Intel® Xeon® Processor with 800 MHz System Bus (1 MB and 2 MB L2 Cache Versions) FC-PGA4 Package

Processor Signature / Core Stepping	0F34h / D-0	0F41h / E-0	0F49h / G-1	0F43h / N-0	0F4Ah / R-0
0F34h / D-0	NI	NI	NI	X	X
0F41h / E-0	NI	NI	NI	X	X
0F49h / G-1	NI	NI	NI	X	X

Table 2-3. DP Platform Population Matrix for the 64-bit Intel® Xeon® Processor with 800 MHz System Bus (1 MB and 2 MB L2 Cache Versions) FC-PGA4 Package

Processor Signature / Core Stepping	0F34h / D-0	0F41h / E-0	0F49h / G-1	0F43h / N-0	0F4Ah / R-0
0F43h / N-0	X	X	X	NI	NI
0F4Ah / R-0	X	X	X	NI	NI

NOTES:
1. X = Mixing processors of different steppings is not supported. This stepping/frequency is not supported in DP.
2. NI = Currently no known issues associated with mixing these steppings.
3. TBD = No issues are expected, however further investigation is required to fully validate this DP solution.

Summary Table of Changes 3

The following table indicates the Errata, Documentation Changes, Specification Clarifications, or Specification Changes that apply to Intel processors. Intel intends to fix some of the errata in a future stepping of the component, and to account for the other outstanding issues through documentation or specification changes as noted. This table uses the following notation:

3.1 Codes Used In Summary Table

X: Erratum, Specification Change or Clarification that applies to the given processor stepping.

(No mark) or (Blank Box): This erratum is fixed in listed stepping or specification change does not apply to listed stepping.

Doc: Document change or update that will be implemented.

Plan Fix: This erratum may be fixed in a future of the product.

Fixed: This erratum has been previously fixed.

No Fix: There are no plans to fix this erratum.

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

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

A = Dual-Core Intel® Xeon® processor 7000 sequence

B = Pentium® processor for Embedded Applications

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 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(R) Celeron(R) processor in 478 pin package
AD	=	Intel(R) Celeron(R) 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 and Intel® Core™2 Duo desktop processor E6000 and E4000 sequence
AJ	=	Quad-Core Intel® Xeon® processor 5300 series
AK	=	Intel® Core™2 Extreme quad-core processor QX6000 sequence and Intel® Core™2 Quad processor Q6000 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 sequence
AR	=	Intel® Celeron® processor 500 series
AS	=	Intel® Xeon® processor 7200, 7300 series
AU	=	Intel® Celeron® Dual Core processor T1400
AV	=	Intel® Core™2 Extreme processor QX9650 and Intel® Core™2 Quad processor Q9000 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

AAD = Intel® Core™2 Extreme Processor QX9775?
AAE = Intel® Atom™ processor Z5xx series
AAF = Intel® Atom™ processor 200 series
AAG = Intel® Atom™ processor N series
AAI = Intel® Xeon® Processor 7400 Series
AAJ = Intel® Core™ i7 and Intel® Core™ i7 Extreme Edition
AAK = Intel® Xeon® Processor 5500 Series
AAL = Intel® Pentium® Dual-Core Processor E5000 Series
AAM = Intel® Xeon® Processor 3500 Series

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

3.2 Errata (Sheet 1 of 6)

No.	D-0 / 0F34 h	E-0 / 0F41 h	G-1 / 0F49 h	N-0 / 0F43 h	R-0 / 0F4A h	Plans	Errata
1	X	X	X	X	X	No Fix	Transaction is not retired after BINIT#
2	X	X	X	X	X	No Fix	Invalid opcode 0FFFh requires a ModRM byte
3	X	X	X	X	X	No Fix	Processor may hang due to speculative page walks to non-existent system memory
4	X	X	X	X	X	No Fix	Memory type of the load lock different from its corresponding store unlock
5	X	X	X	X	X	No Fix	Machine Check Architecture error reporting and recovery may not work as expected
6	X	X	X	X	X	No Fix	Debug mechanisms may not function as expected
7	X	X	X	X	X	No Fix	Cascading of performance counters does not work correctly when forced overflow is enabled
8	X	X	X	X	X	No Fix	EMON event counting of x87 loads may not work as expected
9	X	X	X	X	X	No Fix	System bus interrupt messages without data and which receive a hard-failure response may hang the processor
10	X	X	X	X	X	No Fix	The processor signals page fault exception (#PF) instead of alignment check exception (#AC) on an unlocked CMPXCHG8B instruction
11	X	X	X	X	X	No Fix	FSW may not be completely restored after page fault on FRSTOR or FLDENV instructions
12	X	X	X	X	X	No Fix	Processor issues inconsistent transaction size attributes for locked operation
13	X	X	X	X	X	No Fix	When the processor is in the system management mode (SMM), Debug registers may be fully writeable
14	X	X	X	X	X	No Fix	Shutdown and IERR# may result due to a machine check exception on a Hyper-Threading Technology enabled processor

3.2 Errata (Sheet 2 of 6)

No.	D-0 / 0F34 h	E-0 / 0F41 h	G-1 / 0F49 h	N-0 / 0F43 h	R-0 / 0F4A h	Plans	Errata
15	X	X	X	X	X	No Fix	Processor may hang under certain frequencies and 12.5% STPCLK# duty cycle
16	X	X	X	X	X	No Fix	System may hang if a fatal cache error causes bus write line (BWL) transaction to occur to the same cache line address as an outstanding bus read line (BRL) or bus read-invalidate line (BRIL)
17	X	X	X	X	X	No Fix	A write to APIC task priority register (TPR) that lowers priority may seem to have not occurred
18	X	X	X	X	X	No Fix	Parity error in the L1 cache may cause the processor to hang
19	X	X				Fixed	Sequence of locked operations can cause two threads to receive stale data and cause application hang
20	X					Fixed	A 16-bit address wrap resulting from a near branch (jump or call) may cause an incorrect address to be reported to the #GP exception handler
21	X	X	X	X	X	No Fix	Bus locks and SMC detection may cause the processor to temporarily hang
22						Fixed	Incorrect physical address size returned by CPUID instruction
23	X	X	X	X	X	No Fix	Incorrect debug exception (#DB) may occur when a data breakpoint is set on an FP instruction
24	X	X	X	X	X	No Fix	xAPIC may not report some illegal vector errors
25	X	X	X	X	X	Plan Fix	Enabling no-eviction mode (NEM) may prevent the operation of the second logical processor in a Hyper-Threading Technology enabled boot strap processor (BSP)
26	X	X	X	X	X	Plan Fix	TPR (Task Priority Register) updates during voltage transitions of power management events may cause a system hang
27		X	X	X	X	No Fix	Interactions between the instruction translation lookaside buffer (ITLB) and the instruction streaming buffer may cause unpredictable software behavior
28	X	X	X			Fixed	STPCLK# signal assertion under certain conditions may cause a system hang
29	X	X	X	X	X	No Fix	Incorrect duty cycle is chosen when on-demand clock modulation is enabled in a processor supporting Hyper-Threading Technology
30	X	X	X	X	X	No Fix	Memory aliasing of pages as uncacheable memory type and write back (WB) may hang the system
31	X	X	X	X	X	No Fix	Using STPCLK# and executing code from very slow memory could lead to a system hang
32	X	X	X	X	X	No Fix	Processor provides a 4-byte store unlock after an 8-byte load lock
33							Duplicate Erratum: see S5
34	X	X	X	X	X	Plan Fix	Execution of IRET and INTn instructions may cause unexpected system behavior

3.2 Errata (Sheet 3 of 6)

No.	D-0 / 0F34 h	E-0 / 0F41 h	G-1 / 0F49 h	N-0 / 0F43 h	R-0 / 0F4A h	Plans	Errata
35	X	X	X	X	X	No Fix	Data breakpoints on the high half of a floating-point line split may not be captured
36	X	X	X	X	X	No Fix	Machine Check Exceptions may not update Last-Exception Record MSRs (LERs)
37	X	X	X	X	X	No Fix	MOV CR3 performs incorrect reserved bit checking when in PAE paging
38	X	X	X	X	X	No Fix	Stores to page tables may not be visible to pagewalks for subsequent loads without serializing or invalidating the page table entry
39	X					Fixed	A split store memory access may miss a data breakpoint
40	X					Fixed	EFLAGS.RF may be incorrectly set after an IRET instruction
41	X					Fixed	Writing the Echo TPR disable bit in IA32_MISC_ENABLE may cause a #GP fault
42	X					Fixed	Incorrect access controls to MSR_LASTBRANCH_0_FROM_LIP MSR registers
43	X	X				Fixed	Recursive page walks may cause a system hang
44	X					Fixed	WRMSR to bit[0] of IA32_MISC_ENABLE register changes only one logical processor on a Hyper-Threading Technology enabled processor
45	X	X	X	X		Fixed	VERR/VERW instructions may cause #GP fault when descriptor is in non-canonical space
46	X					Fixed	INS or REP INS flows save an incorrect memory address for SMI on processors supporting Intel® Extended Memory 64 Technology (Intel® EM64T)
47	X					Fixed	FXSAVE instruction may result in incorrect data on processors supporting Intel® Extended Memory 64 Technology (Intel® EM64T)
48	X	X				Fixed	The base of a null segment may be non-zero on a processor supporting Intel® Extended Memory 64 Technology (Intel® EM64T)
49	X	X				Fixed	Upper 32 bits of FS/GS with null base may not get cleared in Virtual-8086 Mode on processors with Intel® Extended Memory 64 Technology (Intel® EM64T) Enabled
50	X	X	X	X	X	No Fix	Processor may fault when the upper 8 bytes of segment selector is loaded from a far jump through a call gate via the Local Descriptor Table
51	X					Fixed	Compatibility mode STOS instructions may alter RSI register results on a processor supporting Intel® Extended Memory 64 Technology (Intel® EM64T)
52	X					Fixed	LDT descriptor which crosses 16 bit boundary access does not cause a #GP fault on a processor supporting Intel® Extended Memory 64 Technology (Intel® EM64T)

3.2 Errata (Sheet 4 of 6)

No.	D-0 / 0F34 h	E-0 / 0F41 h	G-1 / 0F49 h	N-0 / 0F43 h	R-0 / 0F4A h	Plans	Errata
53	X					Fixed	Upper reserved bits are incorrectly checked while loading PDPTR's on a processor supporting Intel® Extended Memory 64 Technology (Intel® EM64T)
54	X	X	X	X	X	No Fix	Loading a stack segment with a selector that references a non-canonical address can lead to a #SS fault on a processor supporting Intel® Extended Memory 64 Technology (Intel® EM64T)
55	X					Fixed	CPUID instruction incorrectly reports CMPXCH16B as supported
56	X	X	X	X	X	No Fix	FXRSTOR may not restore non-canonical effective addresses on processors with Intel® Extended Memory 64 Technology (Intel® EM64T) enabled
57	X	X	X	X	X	No Fix	A push of ESP that faults may zero the upper 32-bits of RSP
58		X				Fixed	Enhanced halt state (C1E) voltage transition may affect a system's power management in a Hyper-Threading Technology enabled processor
59		X	X	X	X	No Fix	Enhanced halt state (C1E) may not be entered in a Hyper-Threading Technology enabled processor
60		X				Fixed	When the Execute Disable Bit function is enabled a page fault in a mispredicted branch may result in a page fault exception
61		X				Fixed	Execute Disable Bit set with AD assist may cause livelock
62		X	X			Fixed	The Execute Disable Bit fault may be reported before other types of page fault when both occur
63		X				Fixed	Writes to IA32_MISC_ENABLE may not update flags for both logical processors
64		X				Fixed	Execute Disable Bit set with CR4.PAE may cause livelock
65	X					Fixed	SYSENTER or SYSEXIT instructions may experience incorrect canonical address checking on processors supporting Intel® Extended Memory 64 Technology (Intel® EM64T)
66	X	X	X	X	X	No Fix	Checking of Page Table Base Address may not match Address Bit Width supported by the platform
67	X	X	X	X	X	No Fix	IA32_MCi_STATUS MSR may improperly indicate that additional MCA information may have been captured
68	X	X	X	X	X	No Fix	With Trap Flag (TF) asserted, FP instruction that triggers unmasked FP Exception may take single step trap before retirement of instruction
69	X	X	X	X	X	No Fix	PDE/PTE loads and continuous locked updates to the same cache line may cause system livelock
70	X					Fixed	MCA-corrected memory hierarchy error counter may not increment correctly
71	X	X	X	X	X	No Fix	Branch Trace Store (BTS) and Precise Event-Based Sampling (PEBS) may update memory outside the BTS/PEBS buffer

3.2 Errata (Sheet 5 of 6)

No.	D-0 / 0F34 h	E-0 / 0F41 h	G-1 / 0F49 h	N-0 / 0F43 h	R-0 / 0F4A h	Plans	Errata
72		X	X	X		Fixed	L-bit of CS and LMA bit of IA32_EFER register may have erroneous value for one instruction following mode transition in Hyper-Threading Technology-Enabled processor supporting Intel® Extended Memory 64 Technology (Intel® EM64T)
73	X	X	X	X		Fixed	The base of an LDT (Local Descriptor Table) register may be non-zero on a processor supporting Intel® Extended Memory 64 Technology (Intel® EM64T)
74				X		Fixed	Unaligned Page-Directory-Pointer (PDPTR) Base with 32-bit mode PAE (Page Address Extension) paging may cause processor to hang
75	X	X	X	X	X	No Fix	Memory ordering failure may occur with snoop filtering third-party agents after issuing and completing a BWIL (Bus Write Invalidate Line) or BLW (Bus Locked Write) transaction
76	X	X	X	X	X	No Fix	Control Register 2 (CR2) can be updated during a REP MOVS/STOS instruction with fast strings enabled
77	X	X	X	X	X	No Fix	REP STOS/MOVS instructions with RCX >= 2^32 may cause system hang
78	X	X	X	X		Fixed	REP MOVS or REP STOS instruction with RCX >= 2^32 may fail to execute to completion or may write to incorrect memory locations on processors supporting Intel® Extended Memory 64 Technology (Intel® EM64T)
79	X	X	X	X	X	Plan Fix	An REP LODSB or an REP LODSD or an REP LODSQ instruction with RCX >= 2^32 may cause a system hang on processors supporting Intel® Extended Memory 64 Technology (Intel® EM64T)
80	X	X	X	X		Fixed	Data access which spans both canonical and non-canonical address space may hang system
81	X	X	X	X	X	Plan Fix	Running in System Management Mode (SMM) and L1 data cache adaptive mode may cause unexpected system behavior when SMRAM is mapped to cacheable memory
82	X	X	X	X	X	No Fix	A 64-bit value of Linear Instruction Pointer (LIP) may be reported incorrectly in the Branch Trace Store (BTS) memory record or in the Precise Event Based Sampling (PEBS) memory record
83	X	X	X	X	X	Plan Fix	It is possible that two specific invalid opcodes may cause unexpected memory accesses
84	X	X	X	X	X	No Fix	At core-to-bus ratios of 16:1 and above Defer Reply transactions with non-zero REQb values may cause a Front Side Bus stall
85	X	X	X	X	X	No Fix	Processor may issue Frost Side Bus transactions up to 6 clocks after RESET# is asserted
86	X	X	X	X	X	No Fix	Front Side Bus machine checks may be reported as a result of on-going transactions during warm reset
S87	X	X	X	X	X	No Fix	Writing the local vector table (LVT) when an interrupt is pending may cause an unexpected interrupt

3.2 Errata (Sheet 6 of 6)

No.	D-0 / 0F34 h	E-0 / 0F41 h	G-1 / 0F49 h	N-0 / 0F43 h	R-0 / 0F4A h	Plans	Errata
S88	X	X	X	X	X	No Fix	The processor may issue multiple code fetches to the same cache line for systems with slow memory
S89			X			Plan Fix	CPUID feature flag reports LAHF/SAHF as unavailable, however the execution of LAHF/SAHF may not result in an Invalid Opcode exception
S90	X	X	X	X	X	No Fix	IRET under certain conditions may cause an unexpected Alignment Check Exception
S91	X					Fixed	Upper 32 bits of ‘From’ address reported through LBR or LER MSRs, BTMs or BTSs may be incorrect
S92	X	X	X	X	X	NoFix	EXTEST/CLAMP may cause incorrect values to be driven on processor pins
S93	X	X	X	X	X	No Fix	The IA32_MC0_STATUS/ IA32_MC1_STATUS Overflow Bit is not set when multiple un-correctable machine check errors occur at the same time.
S94	X	X	X	X	X	No Fix	Debug Status Register (DR6) Breakpoint Condition Detected Flags May be Set Incorrectly
S95	X	X	X	X	X	No Fix	L2 Cache ECC Machine Check Errors May be erroneously Reported after an Asynchronous RESET# Assertion

3.3 Specification Changes

No.	SPECIFICATION CHANGES
	None for this revision of the Specification Update.

3.4 Specification Clarifications

No.	SPECIFICATION CLARIFICATIONS
	None for this revision of the Specification Update.

3.5 Documentation Changes

No.	DOCUMENTATION CHANGES
S1	VID Range Specification for Low Voltage Intel® Xeon® Processor with 800 MHz System Bus

Errata 4

S1 Transaction is not retired after BINIT#

Problem: If the first transaction of a locked sequence receives a HITM# and DEFER# during the snoop phase it should be retried and the locked sequence restarted. However, if BINIT# is also asserted during this transaction, the transaction will not be retried.

Implication: When this erratum occurs, locked transactions will not be retried.

Workaround: None at this time.

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

S2 Invalid opcode 0FFFh requires a ModRM byte

Problem: Some invalid opcodes require a ModRM byte and other following bytes, while others do not. The invalid opcode 0FFFh did not require a ModRM in previous generation microprocessors such as Pentium II or Pentium III processors, but it is required in the Intel Xeon processor.

Implication: The use of an invalid opcode 0FFFh without the ModRM byte may result in a page or limit fault on the Intel Xeon processor. When this erratum occurs, locked transactions will not be retried.

Workaround: To avoid this erratum use ModRM byte with invalid 0FFFh opcode.

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

S3 Processor may hang due to speculative page walks to non-existent system memory

Problem: A load operation issued speculatively by the processor that misses the data translation lookaside buffer (DTLB) results in a page walk. A branch instruction older than the load retires so that this load operation is now in the mispredicted branch path. Due to an internal boundary condition, in some instances the load is not canceled before the page walk is issued.

The page miss handler (PMH) starts a speculative page-walk for the Load and issues a cacheable load of the page directory entry (PDE). This PDE load returns data that points to a page table entry in uncacheable (UC) memory. The PMH issues the PTE Load to UC space, which is issued on the front side bus. No response comes back for this load PTE operation since the address is pointing to system memory, which does not exist.

This load to non-existent system memory causes the processor to hang because other bus requests are queued up behind this UC PTE load, which never gets a response. If the load was accessing valid system memory, the speculative page-walk would successfully complete and the processor would continue to make forward progress.

Implication: Processor may hang due to speculative page walks to non-existent system memory.

Workaround: Page directories and page tables in UC memory space must point to system memory that exists.

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

S4 Memory type of the load lock different from its corresponding store unlock

Problem: The Intel Xeon Processor employs a use-once protocol to ensure that a processor in a multiprocessor system may access data that is loaded into its cache on a read-for-ownership (RFO) operation at least once before it is snooped out by another processor. This protocol is necessary to avoid a dual processor livelock scenario where no processor in the system can gain ownership of a line and modify it before that data is snooped out by another processor. In the case of this erratum, the use-once protocol incorrectly activates for split load lock instructions. A load lock operation accesses data that splits across a page boundary with both pages of WB memory type. The use-once protocol activates and the memory type for the split halves get forced to UC. Since use-once does not apply to stores, the store unlock instructions go out as WB memory type. The full sequence on the Bus is: locked partial read (UC), partial read (UC), partial write (WB), locked partial write (WB). The Use-once protocol should not be applied to Load locks.

Implication: When this erratum occurs, the memory type of the load lock will be different than the memory type of the store unlock operation. This behavior (Load Locks and Store Unlocks having different memory types) does not however introduce any functional failures such as system hangs or memory corruption.

Workaround: None at this time.

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

S5 Machine Check Architecture error reporting and recovery may not work as expected

Problem: When the processor detects errors it should attempt to report and/or recover from the error. In the situations described below, the processor does not report and/or recover from the error(s) as intended.

- When a transaction is deferred during the snoop phase and subsequently receives a Hard Failure response, the transaction should be removed from the bus queue so that the processor may proceed. Instead, the transaction is not properly removed from the bus queue, the bus queue is blocked, and the processor will hang.
- When a hardware prefetch results in an uncorrectable tag error in the L2 cache, MC0_STATUS.UNCOR and MC0_STATUS.PCC are set but no machine check exception (MCE) is signaled. No data loss or corruption occurs because the data being prefetched has not been used. If the data location with the uncorrectable tag error is subsequently accessed, an MCE will occur. However, upon this MCE, or any other subsequent MCE, the information for that error will not be logged because MC0_STATUS.UNCOR has already been set and the MCA status registers will not contain information about the error which caused the MCE assertion but instead will contain information about the prefetch error event.
- When the reporting of errors is disabled for machine check architecture (MCA) Bank 2 by setting all MC2_CTL register bits to 0, uncorrectable errors should be logged in the IA32_MC2_STATUS register but no machine-check exception should be generated. Uncorrectable loads on bank 2, which would normally be logged in the IA32_MC2_STATUS register, are not logged.
- When one half of a 64 byte instruction fetch from the L2 cache has an uncorrectable error and the other 32 byte half of the same fetch from the L2 cache has a correctable

error, the processor will attempt to correct the correctable error but cannot proceed due to the uncorrectable error. When this occurs the processor will hang.

- When an L1 cache parity error occurs, the cache controller logic should write the physical address of the data memory location that produced that error into the IA32_MC1_ADDR REGISTER (MC1_ADDR). In some instances of a parity error on a load operation that hits the L1 cache, however, the cache controller logic may write the physical address from a subsequent load or store operation into the IA32_MC1_ADDR register.
- When an error exists in the tag field of a cache line such that a request for ownership (RFO) issued by the processor hits multiple tag fields in the L2 cache (the correct tag and the tag with the error) and the accessed data also has a correctable error, the processor will correctly log the multiple tag match error but will hang when attempting to execute the MCE handler.
- If a memory access receives a machine check error on both 64 byte halves of a 128-byte L2 cache sector, the IA32_MC0_STATUS register records this event as multiple errors, i.e., the valid error bit and the overflow error bit are both set indicating that a machine check error occurred while the results of a previous error were in the error-reporting bank. The IA32_MC1_STATUS register should also record this event as multiple errors but instead records this event as only one correctable error.
- The overflow bit should be set to indicate when more than one error has occurred. The overflow bit being set indicates that more than one error has occurred. Because of this erratum, if any further errors occur, the MCA overflow bit will not be updated; thereby incorrectly indicating only one error has been received.
- If an I/O instruction (IN, INS, REP INS, OUT, OUTS, or REP OUTS) is being executed, and if the data for this instruction becomes corrupted, the processor will signal a MCE. If the instruction is directed at a device that is powered down, the processor may also receive an assertion of SMI#. Since MCEs have higher priority, the processor will call the MCE handler, and the SMI# assertion will remain pending. However, while attempting to execute the first instruction of the MCE handler, the SMI# will be recognized and the processor will attempt to execute the SMM handler. If the SMM handler is successfully completed, it will attempt to restart the I/O instruction, but will not have the correct machine state due to the call to the MCE handler. This can lead to failure of the restart and shutdown of the processor.
- If PWRGOOD is deasserted during a RESET# assertion causing internal glitches, the MCA registers may latch invalid information.
- If RESET# is asserted, then deasserted, and reasserted, before the processor has cleared the MCA registers, then the information in the MCA registers may not be reliable, regardless of the state or state transitions of PWRGOOD.
- If MCERR# is asserted by one processor and observed by another processor, the observing processor does not log the assertion of MCERR#. The MCE handler called upon assertion of MCERR# will not have any way to determine the cause of the MCE.
- The Overflow Error bit (bit 62) in the IA32_MC0_STATUS register indicates, when set, that a machine check error occurred while the results of a previous error were still in the error reporting bank (i.e. The Valid bit was set when the new error occurred). If an uncorrectable error is logged in the error-reporting bank and another error occurs, the overflow bit will not be set.
- A different mechanism than the rest of the register writes the MCA Error Code field of the IA32_MC0_STATUS register. For uncorrectable errors, the other fields in the IA32_MC0_STATUS register are only updated by the first error. Any further errors that are detected will update the MCA Error Code field without updating the rest of the register, thereby leaving the IA32_MC0_STATUS register with stale information.
- When a speculative load operation hits the L2 cache and receives a correctable error, the IA32_MC1_Status Register may be updated with incorrect information. The IA32_MC1_Status Register should not be updated for speculative loads.

- The processor should only log the address for L1 parity errors in the IA32_MC1_Status register if a valid address is available. If a valid address is not available, the Address Valid bit in the IA32_MC1_Status register should not be set. In instances where an L1 parity error occurs and the address is not available because the linear to physical address translation is not complete or an internal resource conflict has occurred, the Address Valid bit is incorrectly set.
- The processor may hang when an instruction code fetch receives a hard failure response from the front side bus. This occurs because the bus control logic does not return data to the core, leaving the processor empty. IA32_MC0_STATUS MSR does indicate that a hard fail response occurred.
- The processor may hang when the following events occur and the machine check exception is enabled, CR4.MCE=1. A processor that has it's STPCLK# pin asserted will internally enter the Stop Grant State and finally issue a Stop Grant Acknowledge special cycle to the bus. If an uncorrectable error is generated during the Stop Grant process it is possible for the Stop Grant special cycle to be issued to the bus before the processor vectors to the machine check handler. Once the chipset receives its last Stop Grant special cycle it is allowed to ignore any bus activity from the processors. As a result, processor accesses to the machine check handler may not be acknowledged, resulting in a processor hang.

Implication: The processor is unable to correctly report and/or recover from certain errors

Workaround: None at this time.

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

S6 Debug mechanisms may not function as expected

Problem: If the first transaction of a locked sequence receives a HITM# and DEFER# during the snoop phase it should be retried and the locked sequence restarted. However, if BINIT# is also asserted during this transaction, the transaction will not be Certain debug mechanisms may not function as expected on the processor. The cases are as follows:

- When the following conditions occur: 1) An `FLD` instruction signals a stack overflow or underflow, 2) the `FLD` instruction splits a page-boundary or a 64 byte cache line boundary, 3) the instruction matches a Debug Register on the high page or cache line respectively, and 4) the FLD has a stack fault and a memory fault on a split access, the processor will only signal the stack fault and the debug exception will not be taken.
- When a data breakpoint is set on the ninth and/or tenth byte(s) of a floating-point store using the Extended Real data type, and an unmasked floating-point exception occurs on the store, the break point will not be captured.
- When any instruction has multiple debug register matches, and any one of those debug registers is enabled in DR7, all of the matches should be reported in DR6 when the processor goes to the debug handler. This is not true during a `REP` instruction. As an example, during execution of a `REP MOVSW` instruction the first iteration a load matches DR0 and DR2 and sets DR6 as FFFF0FF5h. On a subsequent iteration of the instruction, a load matches only DR0. The DR6 register is expected to still contain FFFF0FF5h, but the processor will update DR6 to FFFF0FF1h.

A Data breakpoint that is set on a load to uncacheable memory may be ignored due to an internal segment register access conflict. In this case the system will continue to execute instructions, bypassing the intended breakpoint. Avoiding having instructions that access segment descriptor registers e.g. LGDT, LIDT close to the UC load, and avoiding serialized instructions before the UC load will reduce the occurrence of this erratum.

Implication: Certain debug mechanisms do not function as expected on the processor.

Workaround: None at this time.

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

S7 Cascading of performance counters does not work correctly when forced overflow is enabled

Problem: The performance counters are organized into pairs. When the CASCADE bit of the Counter Configuration Control Register (CCCR) is set, a counter that overflows will continue to count in the other counter of the pair. The FORCE_OVF bit forces the counters to overflow on every non-zero increment. When the FORCE_OVF bit is set, the counter overflow bit will be set but the counter no longer cascades.

Implication: The performance counters do not cascade when the FORCE_OVF bit is set.

Workaround: None at this time.

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

S8 EMON event counting of x87 loads may not work as expected

Problem: If a performance counter is set to count x87 loads and floating-point exceptions are unmasked, the FPU Operand (Data) Pointer (FDP) may become corrupted.

Implication: When this erratum occurs, FPU Operand (Data) Pointer (FDP) may become corrupted.

Workaround: This erratum will not occur with floating point exceptions masked. If floating-point exceptions are unmasked, then performance counting of x87 loads should be disabled.

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

S9 System bus interrupt messages without data and which receive a hard-failure response may hang the processor

Problem: When a system bus agent (processor or chipset) issues an interrupt transaction without data onto the system bus, and the transaction receives a hard-failure response, a potential processor hang can occur. The processor, which generates an inter-processor interrupt (IPI) that receives hard-failure response, will still log the MCA error event cause as hard-failure, even if the APIC causes a hang. Other processors, which are true targets of the IPI, will also hang on hard failure-without-data, but will not record an MCA hard-failure event as a cause. If a hard-failure response occurs on a system bus interrupt message with data, the APIC will complete the operation so as not to hang the processor.

Implication: The processor may hang.

Workaround: None at this time.

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

S10 The processor signals page fault exception (#PF) instead of alignment check exception (#AC) on an unlocked CMPXCHG8B instruction

Problem: If a page fault exception (#PF) and alignment check exception (#AC) both occur for an unlocked `CMPXCHG8B` instruction, then #PF will be flagged.

Implication: Software that depends on the #AC before the #PF will be affected since #PF is signaled in this case.

Workaround: Remove the software's dependency on #AC having precedence over #PF. Alternately, correct the page fault in the page fault handler and then restart the faulting instruction.

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

S11 FSW may not be completely restored after page fault on FRSTOR or FLDENV instructions

Problem: If the FPU operating environment or FPU state (operating environment and register stack) being loaded by an `FLDENV` or `FRSTOR` instruction wraps around a 64-Kbyte or 4-Gbyte boundary and a page fault (#PF) or segment limit fault (#GP or #SS) occurs on the instruction near the wrap boundary, the upper byte of the FPU status word (FSW) might not be restored. If the fault handler does not restart program execution at the faulting instruction, stale data may exist in the FSW.

Implication: When this erratum occurs, stale data will exist in the FSW.

Workaround: Ensure that the FPU operating environment and FPU state do not cross 64-Kbyte or 4-Gbyte boundaries. Alternately, ensure that the page fault handler restarts program execution at the faulting instruction after correcting the paging problem.

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

S12 Processor issues inconsistent transaction size attributes for locked operation

Problem: When the processor is in the Page Address Extension (PAE) mode and detects the need to set the Access and/or Dirty bits in the page directory or page table entries, the processor sends an 8-byte load lock onto the system bus. A subsequent 8 byte store unlock is expected, but instead a 4 byte store unlock occurs. Correct data is provided since only the lower bytes change, however external logic monitoring the data transfer may be expecting an 8-byte store unlock.

Implication: This erratum affects no known commercially available chipsets.

Workaround: None at this time.

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

S13 When the processor is in the system management mode (SMM), Debug registers may be fully writeable

Problem: When in system management mode (SMM), the processor executes code and stores data in the SMRAM space. When the processor is in this mode and writes are made to DR6 and DR7, the processor should block writes to the reserved bit locations. Due to this erratum, the processor may not block these writes. This may result in invalid data in the reserved bit locations.

Implication: Reserved bit locations within DR6 and DR7 may become invalid.

Workaround: Software may perform a read/modify/write when writing to DR6 and DR7 to ensure that the values in the reserved bits are maintained.

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

S14 Shutdown and IERR# may result due to a machine check exception on a Hyper-Threading Technology enabled processor

Problem: When a MCE occurs due to an internal error, both logical processors on a Hyper-Threading (HT) Technology enabled processor normally vector to the MCE handler. However, if one of the logical processors is in the “Wait for SIPI” state, that logical processor will not have a MCE handler and will shut down and assert IERR#.

Implication: A processor with a logical processor in the “Wait for SIPI” state will shut down when an MCE occurs on the other thread.

Workaround: None at this time.

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

S15 Processor may hang under certain frequencies and 12.5% STPCLK# duty cycle

Problem: If a system deasserts STPCLK# at a 12.5% duty cycle, and the processor is running below 2 GHz, and the processor thermal control circuit (TCC) on-demand clock modulation is active, the processor may hang. This erratum does not occur under the automatic mode of the TCC.

Implication: When this erratum occurs, the processor will hang.

Workaround: If use of the on-demand mode of the processor's TCC is desired in conjunction with STPCLK# modulation, then assure that STPCLK# is not asserted at a 12.5% duty cycle.

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

S16 System may hang if a fatal cache error causes bus write line (BWL) transaction to occur to the same cache line address as an outstanding bus read line (BRL) or bus read-invalidate line (BRIL)

Problem: A processor internal cache fatal data ECC error may cause the processor to issue a bus write line (BWL) transaction to the same cache line address as an outstanding bus read line (BRL) or bus read-invalidate line (BRIL). As it is not typical behavior for a single processor to have a BWL and a BRL/BRIL concurrently outstanding to the same address, this may represent an unexpected scenario to system logic within the chipset.

Implication: The processor may not be able to fully execute the machine check handler in response to the fatal cache error if system logic does not ensure forward progress on the system bus under this scenario.

Workaround: System logic should ensure completion of the outstanding transactions. Note that during recovery from a fatal data ECC error, memory image coherency of the BWL with respect to BRL/BRIL transactions is not important. Forward progress is the primary requirement.

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

S17 A write to APIC task priority register (TPR) that lowers priority may seem to have not occurred

Problem: Uncacheable stores to the APIC space are handled in a non-synchronous way with respect to the speed at which instructions are retired. If an instruction that masks the interrupt flag (for example 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 flag is finally cleared (for example STI). Interrupts will remain pended and are not lost

Implication: This condition may allow interrupts to be accepted by the processor but may delay their service

Workaround: This can be avoided by issuing a TPR Read after a TPR Write that lowers the TPR value. This will force the store to the APIC priority resolution logic before any subsequent instructions are executed. No commercial operating system is known to be impacted by this erratum.

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

S18 Parity error in the L1 cache may cause the processor to hang

Problem: If a locked operation accesses a line in the L1 cache that has a parity error, it is possible that the processor may hang while trying to evict the line.

Implication: If this erratum occurs, it may result in a system hang. Intel has not observed this erratum with any commercially available software.

Workaround: None at this time.

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

S19 Sequence of locked operations can cause two threads to receive stale data and cause application hang

Problem: While going through a sequence of locked operations, it is possible for the two threads to receive stale data. This is a violation of expected memory ordering rules and causes the application to hang.

Implication: When this erratum occurs in an Hyper-Thread Technology enabled system, an application may hang.

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

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

S20 A 16-bit address wrap resulting from a near branch (jump or call) may cause an incorrect address to be reported to the #GP exception handler

Problem: If a 16-bit application executes a branch instruction that causes an address wrap to a target address outside of the code segment, the address of the branch instruction should be provided to the general protection exception handler. It is possible that, as a result of this erratum, that the general protection handler may be called with the address of the branch target.

Implication: A 16-bit software environment which is affected by this erratum, will see that the address reported by the exception handler points to the target of the branch, rather than the address of the branch instruction.

Workaround: None at this time.

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

S21 Bus locks and SMC detection may cause the processor to temporarily hang

Problem: The processor may temporarily hang in an HT Technology enabled system, if one logical processor executes a synchronization loop that includes one or more bus locks and is waiting for release by the other logical processor. If the releasing logical processor is executing instructions that are within the detection range of the self-modifying code (SMC) logic, then the processor may be locked in the synchronization loop until the arrival of an interrupt or other event.

Implication: If this erratum occurs in an HT Technology enabled system, the application may temporarily stop making forward progress. Intel has not observed this erratum with any commercially available software.

Workaround: None at this time.

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

S22 Incorrect physical address size returned by CPUID instruction

Problem: The `CPUID` instruction Function 80000008H (Extended Address Sizes Function) returns the address sizes supported by the processor in the EAX register. This Function returns an incorrect physical address size value of 40 bits. The correct physical address size is 36 bits.

Implication: Function 80000008H returns an incorrect physical address size value of 40 bits.

Workaround: None at this time.

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

S23 Incorrect debug exception (#DB) may occur when a data breakpoint is set on an FP instruction

Problem: The default microcode floating-point event handler routine executes a series of loads to obtain data about the `FP` instruction that is causing the FP event. If a data breakpoint is set on the instruction causing the `FP` event, the load in the microcode routine will trigger the data breakpoint resulting in a debug exception.

Implication: An incorrect debug exception (#DB) may occur if data breakpoint is placed on an `FP` instruction. Intel has not observed this erratum with any commercially available software or system.

Workaround: None at this time.

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

S24 xAPIC may not report some illegal vector errors

Problem: The local xAPIC has an error status register, which records all errors it detects. Bit 6 of this register, the receive Illegal Vector bit, is set when the local xAPIC detects an illegal vector in a message that it receives. When an illegal vector error is received on the same internal clock that the error status register is being written due to a previous error, bit 6 does not get set and illegal vector errors are not flagged.

Implication: The xAPIC may not report some Illegal Vector errors when they occur at approximately the same time as other xAPIC errors. The other xAPIC errors will continue to be reported.

Workaround: None at this time.

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

S25 Enabling no-eviction mode (NEM) may prevent the operation of the second logical processor in a Hyper-Threading Technology enabled boot strap processor (BSP)

Problem: In an HT Technology enabled system, when NEM is enabled by setting Bit 0 of MSR 080h (IA32_BIOS_CACHE_AS_RAM), the second logical processor associated with the BSP may fail to wake up from "Wait-for-SIPI" state.

Implication: In an HT Technology enabled system, the second logical processor associated with the BSP may not respond to SIPI. The OS will continue to operate but with one less logical processor than expected.

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

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

S26 TPR (Task Priority Register) updates during voltage transitions of power management events may cause a system hang

Problem: Systems with Echo TPR Disable (R/W) bit (bit [23] of the IA32_MISC_ENABLE register) set to '0' (default), where xTPR messages are being transmitted on the system bus to the processor, may experience a system hang during voltage transitions caused by power management events.

Implication: This may cause a system hang during voltage transitions of power management events.

Workaround: It is possible for the BIOS to contain a workaround for this erratum. The BIOS workaround disables the Echo TPR updates on the affected steppings.

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

S27 Interactions between the instruction translation lookaside buffer (ITLB) and the instruction streaming buffer may cause unpredictable software behavior

Problem: Complex interactions within the instruction fetch/decode unit may make it possible for the processor to execute instructions from an internal streaming buffer containing stale or incorrect information.

Implication: When this erratum occurs, an incorrect instruction stream may be executed resulting in unpredictable software behavior.

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

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

S28 STPCLK# signal assertion under certain conditions may cause a system hang

Problem: The assertion of STPCLK# signal before a logical processor awakens from the "Wait-for-SIPI" state for the first time, may cause a system hang. A processor supporting HT Technology may fail to initialize appropriately, and may not issue a Stop Grant Acknowledge special bus cycle in response to the second STPCLK# assertion.

Implication: When this erratum occurs in an HT Technology enabled system, it may cause a system hang.

Workaround: BIOS should initialize the second thread of the processor supporting HT Technology prior to STPCLK# assertion.

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

S29 Incorrect duty cycle is chosen when on-demand clock modulation is enabled in a processor supporting Hyper-Threading Technology

Problem: When a processor supporting HT Technology enables on-demand clock modulation on both logical processors, the processor is expected to select the lowest duty cycle of the two potentially different values. When one logical processor enters the AUTOHALT state, the duty cycle implemented should be unaffected by the halted logical processor. Due to this erratum, the duty cycle is incorrectly chosen to be the higher duty cycle of both logical processors.

Implication: Due to this erratum, higher duty cycle may be chosen when the on-demand clock modulation is enabled on both logical processors.

Workaround: None at this time.

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

S30 Memory aliasing of pages as uncacheable memory type and write back (WB) may hang the system

Problem: When a page is being accessed as either UC or write combining (WC) and write back (WB), under certain bus and memory timing conditions, the system may loop in a continual sequence of UC fetch, implicit write back, and RFO retries

Implication: This erratum has not been observed in any commercially available operating system or application. The aliasing of memory regions, a condition necessary for this erratum to occur, is documented as being unsupported in the *IA-32 Intel® Architecture Software Developer's Manual*, Volume 3, Section 10.12.4, Programming the PAT. However, if this erratum occurs the system may hang

Workaround: The pages should not be mapped as either UC or WC and WB at the same time.

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

S31 Using STPCLK# and executing code from very slow memory could lead to a system hang

Problem: The system may hang when the following conditions are met:

1. Periodic STPCLK# mechanism is enabled via the chipset.
2. HT Technology is enabled.
3. One logical processor is waiting for an event (i.e. hardware interrupt).
4. The other logical processor executes code from very slow memory such that every code fetch is deferred long enough for the STPCLK# to be reasserted.

Implication: If this erratum occurs, the processor will go into and out of the sleep state without making forward progress, since the logical processor will not be able to service any pending event. This erratum has not been observed in any commercial platform running commercial software.

Workaround: None at this time.

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

S32 Processor provides a 4-byte store unlock after an 8-byte load lock

Problem: When the processor is in the Page Address Extension (PAE) mode and detects the need to set the Access and/or Dirty bits in the page directory or page table entries, the processor sends an 8 byte load lock onto the system bus. A subsequent 8 byte store unlock is expected, but instead a 4 byte store unlock occurs. Correct data is provided since only the lower bytes change, however external logic monitoring the data transfer may be expecting an 8 byte load lock.

Implication: No known commercially available chipsets are affected by this erratum.

Workaround: None at this time.

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

S33 Duplicate erratum: see Erratum S5

S34 Execution of IRET and INTn instructions may cause unexpected system behavior

Problem: There is a small window of time, requiring alignment of many internal micro architectural events, during which the speculative execution of the `IRET` or `INTn` instructions in protected or IA-32e mode may result in unexpected software or system behavior.

Implication: This erratum may result in unexpected instruction execution, events, interrupts or a system hang when the `IRET` instruction is executed. The execution of the `INTn` instruction may cause debug breakpoints to be missed.

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

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

S35 Data breakpoints on the high half of a floating-point line split may not be captured

Problem: When a floating point load which splits a 64-byte cache line gets a floating point stack fault, and a data breakpoint register maps to the high line of the floating point load, internal boundary conditions exist that may prevent the data breakpoint from being captured.

Implication: When this erratum occurs, a data breakpoint will not be captured.

Workaround: None at this time.

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

S36 Machine Check Exceptions may not update Last-Exception Record MSRs (LERs)

Problem: The Last-Exception Record MSRs (LERs) may not get updated when Machine Check Exceptions occur.

Implication: When this erratum occurs, the LER may not contain information relating to the machine check exception. They will contain information relating to the exception prior to the machine check exception.

Workaround: None at this time.

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

S37 MOV CR3 performs incorrect reserved bit checking when in PAE paging

Problem: The `MOV CR3` instruction should perform reserved bit checking on the upper unimplemented address bits. This checking range should match the address width reported by `CPUID` instruction 0x8000008. This erratum applies whenever PAE is enabled.

Implication: Software that sets the upper address bits on a `MOV CR3` instruction and expects a fault may fail. This erratum has not been observed with commercially available software.

Workaround: None at this time.

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

S38 Stores to page tables may not be visible to pagewalks for subsequent loads without serializing or invalidating the page table entry

Problem: Under rare timing circumstances, a page table load on behalf of a programmatically younger memory access may not get data from a programmatically older store to the page table entry if there is not a fencing operation or page translation invalidate operation between the store and the younger memory access. Refer to the *IA-32 Intel® Architecture Software Developer's Manual* for the correct way to update page tables. Software that conforms to the Software Developer's Manual will operate correctly.

Implication: If the guidelines in the Software Developer's Manual are not followed, stale data may be loaded into the processor's Translation Lookaside Buffer (TLB) and used for memory operations. This erratum has not been observed with any commercially available software.

Workaround: The guidelines in the *IA-32 Intel® Architecture Software Developer's Manual* should be followed.

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

S39 A split store memory access may miss a data breakpoint

Problem: It is possible for a data breakpoint specified by a linear address to be missed during a split store memory access. The problem can happen with or without paging enabled.

Implication: This erratum may limit the debug capability of a debugger software.

Workaround: None at this time.

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

S40 EFLAGS.RF may be incorrectly set after an IRET instruction

Problem: EFLAGS.RF is used to disable code breakpoints. After an `IRET` instruction, EFLAGS.RF may be incorrectly set or not set depending on its value right before the `IRET` instruction.

Implication: A code breakpoint may be missed or an additional code breakpoint may be taken on next instruction.

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

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

S41 Writing the Echo TPR disable bit in IA32_MISC_ENABLE may cause a #GP fault

Problem: Writing a '1' to the Echo TPR disable bit (bit 23) in IA32_MISC_ENABLE may incorrectly cause a #GP fault.

Implication: A #GP fault may occur if the bit is set to a '1'.

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

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

S42 Incorrect access controls to MSR_LASTBRANCH_0_FROM_LIP MSR registers

Problem: When an access is made to the MSR_LASTBRANCH_0_FROM_LIP MSR register, an expected #GP fault may not happen.

Implication: A read of the MSR_LASTBRANCH_0_FROM_LIP MSR register may not cause a #GP fault.

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

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

S43 Recursive page walks may cause a system hang

Problem: A page walk, accessing the same page table entry multiple times but at different levels of the page table, which causes the page table entry to have its Access bit set may result in a system hang.

Implication: When this erratum occurs, the system may experience a hang.

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

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

S44 WRMSR to bit[0] of IA32_MISC_ENABLE register changes only one logical processor on a Hyper-Threading Technology enabled processor

Problem: On an HT Technology enabled processor, a write to the fast-strings feature bit[0] of IA32_MISC_ENABLE register changes the setting for the current logical processor only.

Implication: Due to this erratum, the non-current logical processor may not update fast-strings feature bit[0] of IA32_MISC_ENABLE register.

Workaround: BIOS may set the fast-strings enable bit on both logical processors to workaround this erratum. It is possible for the BIOS to contain a workaround for this erratum.

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

S45 VERR/VERW instructions may cause #GP fault when descriptor is in non-canonical space

Problem: If a descriptor referenced by the selector specified for the `VERR` or `VERW` instructions is in non-canonical space, it may incorrectly cause a #GP fault on a processor supporting Intel® Extended Memory 64 Technology (Intel® EM64T).

Implication: Operating systems or drivers that reference a selector in non-canonical space may experience an unexpected #GP fault. Intel has not observed this erratum with any commercially available software.

Workaround: None at this time.

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

S46 INS or REP INS flows save an incorrect memory address for SMI on processors supporting Intel® Extended Memory 64 Technology (Intel® EM64T)

Problem: In IA-32e mode of the Intel EM64T processor, an `INS` or an `REP INS` instruction, followed by an SMI, may save an incorrect memory address to the System Management Mode (SMM) save state location.

Implication: Due to this erratum, the SMM macro-handler may use the incorrect memory address while reproducing the I/O access of SMI.

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

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

S47 FXSAVE instruction may result in incorrect data on processors supporting Intel® Extended Memory 64 Technology (Intel® EM64T)

Problem: In IA-32e mode of the Intel EM64T processor, the upper 32 bits of the FDP value written out to memory by the `FXSAVE` instruction may be incorrect.

Implication: This erratum may cause incorrect data to be saved into the memory.

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

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

S48 The base of a null segment may be non-zero on a processor supporting Intel® Extended Memory 64 Technology (Intel® EM64T)

Problem: In IA-32e mode of the Intel EM64T processor, the base of a null segment may be non-zero.

Implication: Due to this erratum, Intel EM64T enabled systems may encounter unexpected behavior when accessing memory using the null selector.

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

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

S49 Upper 32 bits of FS/GS with null base may not get cleared in Virtual-8086 Mode on processors with Intel® Extended Memory 64 Technology (Intel® EM64T) Enabled

Problem: For processors with Intel EM64T enabled, the upper 32 bits of the FS and GS data segment registers corresponding to a null base may not get cleared when segments are loaded in Virtual-8086 mode.

Implication: This erratum may cause incorrect data to be loaded or stored to memory if FS/GS is not initialized before use in 64-bit mode. Intel has not observed this erratum with any commercially available software.

Workaround: None at this time.

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

S50 Processor may fault when the upper 8 bytes of segment selector is loaded from a far jump through a call gate via the Local Descriptor Table

Problem: In IA-32e mode of the Intel EM64T processor, control transfers through a call gate via the Local Descriptor Table (LDT) that uses a 16-byte descriptor, the upper 8-byte access may wrap and access an incorrect descriptor in the LDT. This only occurs on an LDT with a LIMIT>0x10008 with a 16-byte descriptor that has a selector of 0xFFFC.

Implication: In the event this erratum occurs, the upper 8-byte access may wrap and access an incorrect descriptor within the LDT, potentially resulting in a fault or system hang. Intel has not observed this erratum with any commercially available software.

Workaround: None at this time.

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

S51 Compatibility mode STOS instructions may alter RSI register results on a processor supporting Intel® Extended Memory 64 Technology (Intel® EM64T)

Problem: When a processor supporting Intel EM64T is in IA-32e mode and executes a `STOS` instruction in compatibility mode, it may modify the RSI register contents.

Implication: When this erratum occurs, systems may encounter unexpected behavior.

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

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

S52 LDT descriptor which crosses 16 bit boundary access does not cause a #GP fault on a processor supporting Intel® Extended Memory 64 Technology (Intel® EM64T)

Problem: When a processor supporting Intel EM64T in IA-32e mode accesses an LDT entry (16-byte) that crosses the 0xffff limit, a #GP fault is not signaled and instead the upper 8-bytes of the entry is fetched from the wrapped around address (usually 0x0). This will cause the erroneous data to be loaded into the upper 8-bytes of the descriptor.

Implication: When this erratum occurs, systems may encounter unexpected behavior. Intel has not observed this erratum with any commercially available software.

Workaround: Software should prevent LDT selector accesses from crossing the 0xffff limit.

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

S53 Upper reserved bits are incorrectly checked while loading PDPTR's on a processor supporting Intel® Extended Memory 64 Technology (Intel® EM64T)

Problem: In IA-32 and IA-32e mode of the Intel processor, upper reserved bits are incorrectly checked while loading PDPTR's, allowing software to set the reserved bits.

Implication: Operating system or driver software is able to set the reserved bits which may result in an unexpected system behavior.

Workaround: None identified.

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

S54 Loading a stack segment with a selector that references a non-canonical address can lead to a #SS fault on a processor supporting Intel® Extended Memory 64 Technology (Intel® EM64T)

Problem: When a processor supporting Intel EM64T is in IA-32e mode, loading a stack segment with a selector which references a non-canonical address will result in a #SS fault instead of a #GP fault.

Implication: When this erratum occurs, Intel EM64T enabled systems may encounter unexpected behavior.

Workaround: None at this time.

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

S55 CPUID instruction incorrectly reports CMPXCH16B as supported

Problem: A read of the `CMPXCHG16B` feature flag improperly indicates that the `CMPXCHG16B` instruction is supported.

Implication: When a processor supporting Intel EM64T attempts to execute a `CMPXCH16B` instruction, the system may hang rather than #UD fault.

Workaround: It is possible for the BIOS to contain a workaround for this erratum, such that the `CMPXCH16B` feature flag indicates that the instruction is not supported, and the execution of the `CMPXCHG16B` instruction results in a #UD fault.

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

S56 FXRSTOR may not restore non-canonical effective addresses on processors with Intel® Extended Memory 64 Technology (Intel® EM64T) enabled

Problem: If an x87 data instruction has been executed with a non-canonical effective address, FXSAVE may store that non-canonical FP Data Pointer (FDP) value into the save image. An `FXRSTOR` instruction executed with 64-bit operand size may signal a General Protection Fault (#GP) if the FDP or FP Instruction Pointer (FIP) is in non-canonical form.

Implication: When this erratum occurs, Intel EM64T enabled systems may encounter an unintended #GP fault.

Workaround: Software should avoid using non-canonical effective addressing in EM64T enabled processors. BIOS can contain a workaround for this erratum removing the unintended #GP fault on FXRSTOR.

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

S57 A push of ESP that faults may zero the upper 32-bits of RSP

Problem: In the event that a push `ESP` instruction, that faults, is executed in compatibility mode, the processor will incorrectly zero upper 32-bits of RSP.

Implication: A Push of ESP in compatibility mode will zero the upper 32-bits of RSP. Due to this erratum, this instruction fault may change the contents of RSP. This erratum has not been observed in commercially available software.

Workaround: None at this time.

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

S58 Enhanced halt state (C1E) voltage transition may affect a system's power management in a Hyper-Threading Technology enabled processor

Problem: In an HT Technology enabled system, the second logical Processor may fail to wake up from "Wait-for-SIPI" state during a C1E voltage transition.

Implication: This erratum may affect a system's entry into the power management mode offered by the C1E event for HT Technology enabled platforms.

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

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

S59 Enhanced halt state (C1E) may not be entered in a Hyper-Threading Technology enabled processor

Problem: If the IA32_MISC_ENABLE MSR (0x1A0) C1E enable bit is not set prior to an INIT event on an HT Technology enabled system, the processor will not enter C1E until the next SIPI wakeup event for the second logical processor.

Implication: Due to this erratum, the processor will not enter C1E state.

Workaround: If C1E is supported in the system, the IA32_MISC_ENABLE MSR should be enabled prior to issuing the first SIPI to the second logical processor.

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

S60 When the Execute Disable Bit function is enabled a page fault in a mispredicted branch may result in a page fault exception

Problem: If a page fault in a mispredicted branch occurs in the ITLB, it should not be reported by the processor. However, if the execute disable bit function is enabled (IA32_EFER.NXE = 1) and there is a page fault in a mispredicted branch in the ITLB, a page fault exception may occur.

Implication: When this erratum occurs, a page fault exception may occur.

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

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

S61 Execute Disable Bit set with AD assist may cause livelock

Problem: If Execute Disable Bit is set and the resulting page requires the processor to set the A and/or D bit (Access and/or Dirty bit) in the PTE, then the processor may livelock.

Implication: When this erratum occurs, the processor may livelock resulting in a system hang or operating system failure.

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

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

S62 The Execute Disable Bit fault may be reported before other types of page fault when both occur

Problem: If the Execute Disable Bit is enabled and both the Execute Disable Bit fault and page faults occur, the Execute Disable Bit fault will be reported prior to other types of page fault being reported.

Implication: No impact to properly written code since both types of faults will be generated but in the opposite order.

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

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

S63 Writes to IA32_MISC_ENABLE may not update flags for both logical processors

Problem: On processors supporting HT Technology with Execute Disable Bit feature, writes to IA32_MISC_ENABLE may only update IA32_EFER.NXE for the current logical processor.

Implication: Due to this erratum, the non-current logical processor may not update its IA32_EFER.NXE bit.

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

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

S64 Execute Disable Bit set with CR4.PAE may cause livelock

Problem: If the Execute Disable bit of IA32_MISC_Enable is set along with the PAE bit of CR4 (IA32_EFER.NXE & CR4.PAE), the processor may livelock.

Implication: When this erratum occurs, the processor may livelock resulting in a system hang or operating system failure.

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

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

S65 SYSENTER or SYSEXIT instructions may experience incorrect canonical address checking on processors supporting Intel® Extended Memory 64 Technology (Intel® EM64T)

Problem: Processors which support Intel EM64T always perform canonical address checks before accessing memory. SYSENTER or SYSEXIT instructions may check an incorrect address.

Implication: Due to this erratum, an unexpected #GP fault may occur, or a reference to a non-canonical address without a #GP fault may occur.

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

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

S66 Checking of Page Table Base Address may not match Address Bit Width supported by the platform

Problem: If the page table base address included in the page map level-4 table, page-directory pointer table, page-directory table, or page table exceeds the physical address range supported by the platform (e.g. 36 bits) and it is less than the implemented address range (e.g. 40 bits), the processor does not check to see if the address is invalid.

Implication: If software sets such an invalid physical address in the listed tables, the processor does not generate a page fault (#PF) upon accessing that virtual address, and the access results in an incorrect read or write. If BIOS provides only valid physical address ranges to the operating system, this erratum will not occur.

Workaround: Ensure that BIOS provides only valid physical address ranges to the operating system.

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

S67 IA32_MCi_STATUS MSR may improperly indicate that additional MCA information may have been captured

Problem: When a data parity error is detected and the bus queue is busy, the ADDRV and MISCV bits of the IA32_MCi_STATUS register may be asserted even though the contents of the IA32_MCi_ADDR and IA32_MCi_MISC MSRs were not properly captured.

Implication: If this erratum occurs, the MCA information captured in the IA32_MCi_ADDR and IA32_MCi_MISC registers may not correspond to the reported machine-check error, even though the ADDRV and MISCV are asserted.

Workaround: None at this time.

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

S68 With Trap Flag (TF) asserted, FP instruction that triggers unmasked FP Exception may take single step trap before retirement of instruction

Problem: If an FP instruction generates an unmasked exception with the EFLAGS.TF = 1, it is possible for external events to occur, including a transition to a lower power state. When resuming from a lower power state, it may be possible to take the single step trap before the execution of the original FP instruction completes.

Implication: When this erratum occurs, a single step trap will be taken unexpectedly.

Workaround: None at this time.

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

S69 PDE/PTE loads and continuous locked updates to the same cache line may cause system livelock

Problem: In a multi-processor configuration, if one processor is continuously doing locked updates to a cache line that is being accessed by another processor doing a page table walk, the page table walk may not complete.

Implication: Due to this erratum, the system may livelock until some external event interrupts the locked update. Intel has not observed this erratum with any commercially available software.

Workaround: None at this time.

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

S70 MCA-corrected memory hierarchy error counter may not increment correctly

Problem: An MCA-corrected memory hierarchy error counter can report a maximum of 255 errors. Due to the incorrect increment of the counter, the number of errors reported may be incorrect.

Implication: Due to this erratum, the MCA counter may report an incorrect number of soft errors.

Workaround: None at this time.

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

S71 Branch Trace Store (BTS) and Precise Event-Based Sampling (PEBS) may update memory outside the BTS/PEBS buffer

Problem: If the BTS/PEBS buffer is defined such that:

1.The difference between the BTS/PEBS buffer base and the BTS/PEBS absolute maximum is not an integer multiple of the corresponding record sizes,

2.The BTS/PEBS absolute maximum is less than a record size from the end of the virtual address space, and

3.The record that would cross the BTS/PEBS absolute maximum will also continue past the end of the virtual address space,

a.BTS/PEBS record can be written that will wrap at the 4-Gbyte boundary (IA-32) or 2^64 boundary (Intel EM64T mode), and write memory outside of the BTS/PEBS buffer.

Implication: Software that uses BTS/PEBS near the 4-Gbyte boundary (IA-32) or 2^64 boundary (Intel EM64T mode), and defines the buffer such that it does not hold an integer multiple of records, can update memory outside the BTS/PEBS buffer.

Workaround: Define the BTS/PEBS buffer such that the BTS/PEBS absolute maximum minus the BTS/PEBS buffer base is an integer multiple of the corresponding record sizes as recommended in the IA-32 *IA-32 Intel® Architecture Software Developer's Manual*, Volume 3.

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

S72 L-bit of CS and LMA bit of IA32_EFER register may have erroneous value for one instruction following mode transition in Hyper-Threading Technology-Enabled processor supporting Intel® Extended Memory 64 Technology (Intel® EM64T)

Problem: In an Intel EM64T-enabled processor, the L-bit of the Code Segment (CS) descriptor may not update with the correct value in a processor with HT Technology. This may occur in a small window when one logical processor is making a transition from a compatibility-mode to a 64-bit mode (or vice versa) while the other logical processor is being stalled. A similar problem may occur for the observation of the EFER.LMA bit by the decode logic.

Implication: The first instruction following a mode transition may be decoded as if it was still in the previous mode. For example, this may result in an incorrect stack size used for a stack operation, i.e. a write of only 4 bytes and an adjustment to ESP of only 4 in 64-bit mode. The problem can manifest itself on any instruction which may behave differently in 64-bit mode than in compatibility mode.

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

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

S73 The base of an LDT (Local Descriptor Table) register may be non-zero on a processor supporting Intel® Extended Memory 64 Technology (Intel® EM64T)

Problem: In IA-32e mode of an Intel EM64T-enabled processor, the base of an LDT register may be non-zero.

Implication: Due to this erratum, Intel EM64T-enabled systems may encounter unexpected behavior when accessing an LDT register using the null selector. There may be no #GP fault in response to this access.

Workaround: None identified.

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

S74 Unaligned Page-Directory-Pointer (PDPTR) Base with 32-bit mode PAE (Page Address Extension) paging may cause processor to hang

Problem: When the MOV to CR0, CR3 or CR4 instructions are executed in legacy PAE paging mode and software is using an unaligned PDPTR base the processor may hang or an incorrect page translation may be used.

Implication: Software that is written according to Intel's alignment specification (32-byte aligned PDPTR Base) will not encounter this erratum. Intel has not observed this erratum with commercially available software. Systems may hang or experience unpredictable behavior when this erratum occurs.

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

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

S75 Memory ordering failure may occur with snoop filtering third-party agents after issuing and completing a BWIL (Bus Write Invalidate Line) or BLW (Bus Locked Write) transaction

Problem: Under limited circumstances, the processors may, after issuing and completing a BWIL or BLW transaction, retain data from the addressed cache line in shared state even though the specification requires complete invalidation. This data retention may also occur when a BWIL transaction's self-snooping yields HITM snoop results.

Implication: A system may suffer memory ordering failures if its central agent incorporates coherence sequencing which depends on a full self-invalidation of the cache line associated with (1) BWIL and BLW transactions, or (2) all HITM snoop results without regard to the transaction type and snoop results' source.

Workaround: 1. The central agent can issue a bus cycle that causes a cache line to be invalidated (Bus Read Invalidate Line (BRIL) or BWIL transaction) in response to a processor-generated BWIL (or BLW) transaction to insure complete validation of the associated cache line. If there are no intervening processor-originated transactions to that cache line, the central agent's invalidating snoop will get a clean snoop result. Or,

1.Snoop filtering central agents can:

a.Not use processor-originated BWIL or BLW transactions to update their snoop filter information, or b.Update the associated cache line state information to shared state on the originating bus (rather than invalid state) in reaction to a BWIL or BLW.

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

S76 Control Register 2 (CR2) can be updated during a REP MOVS/STOS instruction with fast strings enabled

Problem: Under limited circumstances while executing a REP MOVS/STOS string instruction, with fast strings enabled, it is possible for the value in CR2 to be changed as a result of an interim paging event, normally invisible to the user. Any higher priority architectural event that arrives and is handled while the interim paging event is occuring may see the modified value of CR2.

Implication: The value in CR2 is correct at the time that an architectural page fault is signaled. Intel has not observed this erratum with any commercially available software.

Workaround: None identified.

Status: For the steppings affected, see the *Summary Table of Changes* at the beginning of this section.

S77 REP STOS/MOVS instructions with RCX >= 2^32 may cause system hang

Problem: In IA-32e mode using Intel EM64T-enabled processors, executing a repeating string instruction with the iteration count greater than or equal to 2^32 and a pending event may cause the REP STOS/MOVS instruction to live lock and hang.

Implication: When this erratum occurs, the processor may live lock and result in a system hang. Intel has not observed this erratum with any commercially available software.

Workaround: Do not use strings larger than 4 GB.

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

S78 REP MOVS or REP STOS instruction with RCX >= 2^32 may fail to execute to completion or may write to incorrect memory locations on processors supporting Intel® Extended Memory 64 Technology (Intel® EM64T)

Problem: In IA-32e mode using Intel EM64T-enabled processors, an REP MOVS or an REP STOS instruction executed with the register RCX >= 2^32, may fail to execute to completion or may write data to incorrect memory locations.

Implication: This erratum may cause incomplete instruction execution or incorrect data in the memory. Intel has not observed this erratum with any commercially available software.

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

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

S79 An REP LODSB or an REP LODSD or an REP LODSQ instruction with RCX >= 2^32 may cause a system hang on processors supporting Intel® Extended Memory 64 Technology (Intel® EM64T)

Problem: In IA-32e mode using Intel EM64T-enabled processors, a REP LOSDB or an REP LODSD or an REP LODSQ instruction executed with the register RCX >= 2^32 may fail to complete execution causing a system hang. Additionally, there may be no #GP fault due to the non-canonical address in the RSI register.

Implication: This erratum may cause a system hang on Intel EM64T-enabled platforms. Intel has not observed this erratum with any commercially available software.

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

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

S80 Data access which spans both canonical and non-canonical address space may hang system

Problem: If a data access causes a page split across the canonical to non-canonical address space, the processor may livelock which in turn would cause a system hang.

Implication: When this erratum occurs, the processor may livelock, resulting in a system hang. Intel has not observed this erratum with any commercially available software.

Workaround: None identified.

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

S81 Running in System Management Mode (SMM) and L1 data cache adaptive mode may cause unexpected system behavior when SMRAM is mapped to cacheable memory

Problem: In a HT Technology-enabled system, unexpected system behavior may occur if a change is made to the value of the CR3 result from an Resume from System Management (RSM) instruction while in L1 data cache adaptive mode (IA32_MISC_ENABLES MSR 0x1a0, bit 24). This behavior will only be visible when SMRAM is mapped into WB/WT cacheable memory on SMM entry and exit.

Implication: This erratum can have multiple failure symptoms, including incorrect data in memory. Intel has not observed this erratum with any commercially available software.

Workaround: Disable L1 data cache adaptive mode by setting the L1 data cache context mode control bit (bit 24) of the IA32_MISC_ENABLES MSR (0x1a0) to 1.

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

S82 A 64-bit value of Linear Instruction Pointer (LIP) may be reported incorrectly in the Branch Trace Store (BTS) memory record or in the Precise Event Based Sampling (PEBS) memory record

Problem: On a processor supporting Intel EM64T,

- If an instruction fetch wraps around the 4G boundary in Compatibility mode, the 64-bit value of LIP in the BTS memory record will be incorrect (upper 32 bits will be set to FFFFFFFFh when they should be 0).
- If a PEBS event occurs on an instruction whose last byte is at memory location FFFFFFFFh, the 64-bit value of LIP in the PEBS record will be incorrect (upper 32 bits will be set to FFFFFFFFh when they should be 0).

Implication: Intel has not observed this erratum on any commercially available software.

Workaround: None identified.

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

S83 It is possible that two specific invalid opcodes may cause unexpected memory accesses

Problem: A processor is expected to respond with an undefined opcode (#UD) fault when executing either opcode 0F 78 or a Grp 6 Opcode with bits 5:3 of the Mod/RM field set to 6, however the processor may respond instead, with a load to an incorrect address.

Implication: This erratum may cause unpredictable system behavior or system hang.

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

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

S84 At core-to-bus ratios of 16:1 and above Defer Reply transactions with non-zero REQb values may cause a Front Side Bus stall

Problem: Certain processors are likely to hang the Front Side Bus (FSB) if the following conditions are met:

1. A Defer Reply transaction has a REQb[2:0] value of either 010b, 011b, 100b, 110b, or 111b, and
2. The operating bus ratio is 16:1 or higher.

When these conditions are met, the processor may incorrectly and indefinitely assert a snoop stall for the Defer Reply transaction. Such an event will block further progress on the FSB.

Implication: If this erratum occurs, the system may hang. Intel has not observed this erratum with any commercially available system.

Workaround: None identified.

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

S85 Processor may issue Frost Side Bus transactions up to 6 clocks after RESET# is asserted

Problem: The processor may issue transactions beyond the documented 3 Front Side Bus (FSB) clocks and up to 6 FSB clocks after RESET# is asserted in the case of a warm reset. A warm reset is where the chipset asserts RESET# when the system is running.

Implication: The processor may issue transactions up to 6 FSB clocks after RESET# is asserted.

Workaround: None identified.

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

S86 Front Side Bus machine checks may be reported as a result of on-going transactions during warm reset

Problem: Processor Front Side Bus (FSB) protocol/signal integrity machine checks may be reported if the transactions are initiated or in-progress during a warm reset. A warm reset is where the chipset asserts RESET# when the system is running.

Implication: The processor may log FSB protocol/signal integrity machine checks if transactions are allowed to occur during RESET# assertions.

Workaround: BIOS may clear FSB protocol/signal integrity machine checks for systems/chipsets which do not block new transactions during RESET# assertions.

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

S87 Writing the local vector table (LVT) when an interrupt is pending may cause an unexpected interrupt

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

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

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

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

S88 The processor may issue multiple code fetches to the same cache line for systems with slow memory

Problem: Systems with long latencies on returning code fetch data from memory, e.g. BIOS ROM, may cause the processor to issue multiple fetches to the same cache line, once per each instruction executed.

Implication: This erratum may slow down system boot time. Intel has not observed a failure, as a result of this erratum, in a commercially available system.

Workaround: None identified.

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

S89 CPUID feature flag reports LAHF/SAHF as unavailable, however the execution of LAHF/SAHF may not result in an Invalid Opcode exception

Problem: As described in the IA-32 Intel Architecture Software Developer's Manual, support for LAHF/SAHF instructions in 64-bit mode has been added to processors with EM64T enabled. The CPUID feature flag may indicate that the LAHF/SAHF instructions are unavailable in 64-bit mode, even though the instructions are supported and able to be executed without an Invalid Opcode exception.

Implication: The CPUID feature flag incorrectly reports LAHF/SAHF instructions as unavailable in 64-bit mode, though they can be executed normally.

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

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

S90 IRET under certain conditions may cause an unexpected Alignment Check Exception

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

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

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

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

S91 Upper 32 bits of 'From' address reported through LBR or LER MSRs, BTMs or BTSs may be incorrect

Problem: When a far transfer switches the processor from IA-32e mode to 32-bit mode, the upper 32 bits of the 'From' (source) addresses reported through the LBR (Last Branch Record) or LER (Last Exception Record) MSRs (Model-Specific Registers), 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 LBR or LER MSRs, BTMs or BTSs may be incorrect.

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

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

S92 EXTEST/CLAMP may cause incorrect values to be driven on processor pins

Problem: When using TAP boundary scan instructions EXTEST/CLAMP while SLP# is asserted at certain bus ratios, incorrect values may be driven on some processor pins. Address Strobe pins do not respond at bus ratios 15:1 and 16:1, while Address pins and Request Command pins do not respond at bus ratio 22:1 and 24:1.

Implication: Due to this erratum, incorrect values may be driven on some processor pins during boundary scan testing.

Workaround: Avoid asserting SLP# when using TAP Boundary Scan instructions EXTEST/CLAMP.

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

S93 The IA32_MC0_STATUS/ IA32_MC1_STATUS Overflow Bit is not set when multiple un-correctable machine check errors occur at the same time.

Problem: When two MC0/MC1 enabled un-correctable machine check errors are detected in the same internal clock cycle, the highest priority error will be logged in IA32_MC0_STATUS / IA32_MC1_STATUS register, but the overflow bit may not be set.

Implication: The highest priority error will be logged and signaled if enabled, but the overflow bit in the IA32_MC0_STATUS/ IA32_MC1_STATUS register may not be set.

Workaround: None identified.

Status: No Fix.

S94 Debug Status Register (DR6) Breakpoint Condition Detected Flags May be Set Incorrectly

Problem: The Debug Status Register (DR6) may report detection of a spurious breakpoint condition under certain boundary conditions when either:

- A "MOV SS" or "POP SS" instruction is immediately followed by a hardware debugger breakpoint instruction, or
- Any debug register access ("MOV DRx, r32" or "MOV r32, DRx") results in a general-detect exception condition.

Implication: Due to this erratum the breakpoint condition detected flags may be set incorrectly.

Workaround: None identified.

Status: No Fix.

S95 L2 Cache ECC Machine Check Errors May be erroneously Reported after an Asynchronous RESET# Assertion

Problem: Machine check status MSRs may incorrectly report the following L2 Cache ECC machine-check errors when cache transactions are in-flight and RESET# is asserted:-

- Instruction Fetch Errors (IA32_MC2_STATUS with MCA error code 153)
- L2 Data Write Errors (IA32_MC1_STATUS with MCA error code 145)

Workaround: When a real run-time L2 Cache ECC Machine Check occurs, a corresponding valid error will normally be logged in the IA32_MC0_STATUS register. BIOS may clear IA32_MC2_STATUS and/or IA32_MC1_STATUS for these specific errors when IA32_MC0_STATUS does not have its VAL flag set.

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

Specification Changes 5

There are no new Specification Changes for this revision of the *Intel® Xeon™ Processor with 800 MHz System Bus.*

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

1. *64-bit Intel® Xeon® Processor with 2 MB L2 Cache Datasheet* (Document Number 306249)

   Link: http://www.intel.com/Assets/PDF/datasheet/306249.pdf

2. *Low Voltage Intel® Xeon® Processor with 800 MHz System Bus Datasheet* (Document Number 304097)

   Link: http://www.intel.com/design/intarch/datashts/304097.htm

All Specification Changes will be incorporated into a future version of the appropriate Intel Xeon processor documentation.

Specification Clarifications 6

There are no new Specification Clarifications for this revision of the *Intel® Xeon™ Processor with 800 MHz System Bus.*

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

1. *64-bit Intel® Xeon® Processor with 2 MB L2 Cache Datasheet* (Document Number 306249)

   Link: http://developer.intel.com/design/xeon/datashts/306249.htm

2. *Low Voltage Intel® Xeon® Processor with 800 MHz System Bus Datasheet* (Document Number 304097)

   Link: http://www.intel.com/design/intarch/datashts/304097.htm

All Specification Clarifications will be incorporated into a future version of the appropriate Intel Xeon processor documentation.

Documentation Changes 7

Note: Documentation changes for *IA-32 Intel® Architecture Software Developer's Manual,* Volumes 1, 2A, 2B, and 3 will be posted in the separate document *IA-32 Intel® Architecture Software Developer's Manual Documentation Changes*. Follow the link below to become familiar with this file.
http://www.intel.com/design/processor/specupdt/252046.htm

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

S1 VID Range Specification for Low Voltage Intel® Xeon® Processor with 800 MHz System Bus

Issue: The VID range specification for the Low Voltage Intel® Xeon® Processor with 800 MHz System Bus is incorrectly documented in the current datasheet as from 1.1125V (min) to 1.2000V (max).

Symbol	Parameter	Min	Typ	Max	Unit	Notes [1]
VID range	VID range for the Low Voltage Intel® Xeon® Processor with 800 MHz System Bus	1.1125		1.2000	V	2,3

The correct VID range specification for the Low Voltage Intel® Xeon® Processor with 800 MHz System Bus is from 1.1000V (min) to 1.2000V (max). This is a documentation issue and the datasheet VID range for the Low Voltage Intel® Xeon® Processor with 800 MHz System Bus shall be replaced with the following changes:

Symbol	Parameter	Min	Typ	Max	Unit	Notes [1]
VID range	VID range for the Low Voltage Intel® Xeon® Processor with 800 MHz System Bus	1.1000		1.2000	V	2,3

Affected Docs: *64-bit Intel® Xeon® Processor with 2 MB L2 Cache Datasheet* (Document Number 306249) Link: http://developer/design/xeon/datashts/306249.htm
Low Voltage Intel® Xeon® Processor with 800 MHz System Bus Datasheet (Document Number 304097) Link: http://www.intel.com/design/intarch/datashts/304097.htm