Notice: The Intel® EP80579 Integrated Processor Product Line may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available upon request.
Contents

Revision History ................................................................. 4
Introduction ........................................................................ 5
Summary Tables of Current Product Issue Activity .................. 7
Identification Information .................................................. 11
IA-32 Core Errata ............................................................. 14
Intel® EP80579 Integrated Processor Product Line Errata ........... 25
Specification Changes ...................................................... 30
Specification Clarifications ................................................ 31
Document-Only Changes .................................................... 32
## Revision History

<table>
<thead>
<tr>
<th>Date</th>
<th>Revision</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>May 2012</td>
<td>011</td>
<td>Updated parts information to include FLI Consolidation to 65nm.</td>
</tr>
</tbody>
</table>
| July 2010 | 010      | Added new EP80579 errata:  
• 13. Potential Power Sequencing Issue |
| June 2010 | 009      | Added new EP80579 errata:  
• 12. Gigabit Ethernet MAC0 PMCS_P.S read only is nondeterministic  
Added the following Specification Clarification;  
• 1. “DC Characteristics for Gigabit Ethernet Inputs: SYS_PWR_OK and GBE_AUX_PWR_GOOD”  
Added the following Document Corrections;  
• 1, “Figure 26-1 USB Port Connections Correction”  
• 2, “Table 49-63 RTC DC Clock Input Characteristics Correction” |
| September 2009 | 008 | Removed Document Only changes #1  
The IEEE 1588-2008 logic does not lock snapshots for certain Layer-2 PTP Ethernet packets. |
| July 2009 | 007 | Added pin name changes and corrections to Document-Only Changes  
Changed the following signal names:  
EX_REQ_GNT# to Reserved 19  
EX_SLAVE_CS# to Reserved 20  
EX_GNT_REQ# to NC57  
EX_WAIT# to NC58  
EX_WDCTXFER to NC59  
Corrected signal name:  
SIU_CTS1 to SIU_CTS1#  
SIU_CTS2 to SIU_CTS2# |
Deleted all entries in the Documentation Changes section. All previous entries have been incorporated into the documentation set. |
Updated IA-32 Core Erratum X48. description.  
| February 2009 | 004 | Modified identification tables. No errata or document changes. |
| February 2009 | 003 | Added Document-Only Changes and Correction for How to Strap LEB_SIZE. |
| December 2008 | 002 | Removed Document Only changes #1 and #2. |
| August 2008 | 001 | Initial release. |
Introduction

Purpose/Scope/Audience

This document is an update to the specifications listed in the Affected Documents/Related Documents table that follows. This document is a compilation of Intel® EP80579 Integrated Processor Product Line Errata, Specification Changes, Specification Clarifications, and Document-Only Changes. It is intended for hardware and software system designers and manufacturers as well as developers of applications, operating systems, or tools.

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

Table 1. Affected Documents/Related Documents

<table>
<thead>
<tr>
<th>Title</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mechanical Design Guide</td>
<td></td>
</tr>
<tr>
<td>Design Guide</td>
<td></td>
</tr>
</tbody>
</table>

Conventions and Terminology

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

Table 2. Conventions and Terminology (Sheet 1 of 2)

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Document-Only Changes</td>
<td>Document-Only Changes are changes to an Intel Parent Specification that result in changes only to an Intel customer document but no changes to a specification or to a parameter for an Intel product. An example of a document-only change is the correction of a typographical error.</td>
</tr>
<tr>
<td>Intel® EP80579 Integrated Processor Product Line Errata</td>
<td>Intel® EP80579 Integrated Processor Product Line Errata are design defects or errors. These may cause the EP80579’s behavior to deviate from published specifications. Hardware and software designed to be used with any given stepping must assume that all errata documented for that stepping are present on all devices.</td>
</tr>
<tr>
<td>(plural) Erratum (singular)</td>
<td></td>
</tr>
</tbody>
</table>
### Table 2. Conventions and Terminology (Sheet 2 of 2)

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parent Specification</td>
<td>A parent specification is a top-level specification from which other documents can be derived, depending on the product or platform. Typically, a parent specification includes a product’s pinout, architectural overview, device operation, hardware interface, or electrical specifications. Examples of parent specifications include the following: Datasheet, Developer’s Manual, Technical Product Specification (also known as “TPS”). The derived documents may be used for purposes other than that for which the parent specification is used.</td>
</tr>
<tr>
<td>Specification Changes</td>
<td>Specification Changes are the result of adding, removing, or changing a feature, after which an Intel product subsequently operates differently than specified in an Intel Parent Specification, but typically the customer does not have to do anything to achieve proper device functionality as a result of Intel adding, removing, or changing a feature.</td>
</tr>
<tr>
<td>Specification Clarifications</td>
<td>Specification Clarifications are changes to a document that arise when an Intel Parent Specification must be reworded so that the specification is either more clear or not in conflict with another specification.</td>
</tr>
</tbody>
</table>
Summary Tables of Current Product Issue Activity

Table 4 through Table 8 indicate the Intel® EP80579 Integrated Processor Product Line Errata, Specification Changes, Specification Clarifications, or Document-Only Changes that apply to the Intel® EP80579 Integrated Processor Product Line. Intel may fix some of the Errata in a future stepping of the component as noted in Table 3 or account for the other outstanding issues through Specification Changes, Specification Clarifications, or Document-Only Changes. Table 4 through Table 8 use the codes listed in Table 3.

Table 3. Codes Used in Summary Tables

<table>
<thead>
<tr>
<th>Code</th>
<th>Column</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>Stepping</td>
<td>Indicates either that, for the stepping/revision listed,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• an erratum exists and is not yet fixed</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• a specification change or specification clarification applies</td>
</tr>
<tr>
<td>No mark or blank</td>
<td>Stepping</td>
<td>Indicates either that, for the stepping/revision listed,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• an erratum is fixed</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• a specification change or specification clarification does not apply</td>
</tr>
<tr>
<td>Plan Fix</td>
<td>Status</td>
<td>This erratum may be fixed in a future stepping/revision.</td>
</tr>
<tr>
<td>Fixed</td>
<td>Status</td>
<td>This erratum has been previously fixed.</td>
</tr>
<tr>
<td>No Fix</td>
<td>Status</td>
<td>There are no plans to fix this erratum.</td>
</tr>
</tbody>
</table>

A change bar to the left of a table row indicates an item that is either new or modified from the previous version of the Specification Update document.

Table 4. IA-32 Core Errata (Sheet 1 of 3)

<table>
<thead>
<tr>
<th>No.</th>
<th>Stepping/Revision</th>
<th>Status</th>
<th>Errata</th>
</tr>
</thead>
<tbody>
<tr>
<td>B0</td>
<td></td>
<td></td>
<td>Code Segment (CS) Is Wrong on SMM Handler When SMBASE Is Not Aligned</td>
</tr>
<tr>
<td>X1. x</td>
<td>No Fix</td>
<td></td>
<td>IFU/BSU Deadlock May Cause System Hang</td>
</tr>
<tr>
<td>X2. x</td>
<td>No Fix</td>
<td></td>
<td>Memory Aliasing with Inconsistent A and D Bits May Cause Processor Deadlock</td>
</tr>
<tr>
<td>X3. x</td>
<td>No Fix</td>
<td></td>
<td>Move to Control Register Instruction May Generate a Breakpoint Report</td>
</tr>
<tr>
<td>X4. x</td>
<td>No Fix</td>
<td></td>
<td>RDMSR or WRMSR to Invalid MSR Address May Not Cause GP Fault</td>
</tr>
<tr>
<td>X5. x</td>
<td>No Fix</td>
<td></td>
<td>Unable to Disable Reads/Writes to Performance Monitoring Related MSRs</td>
</tr>
<tr>
<td>X6. x</td>
<td>No Fix</td>
<td></td>
<td>Upper Four PAT Entries Not Usable with Mode B or Mode C Paging</td>
</tr>
<tr>
<td>X8. x</td>
<td>No Fix</td>
<td></td>
<td>SSE/SSE2 Streaming Store Resulting in a Self-Modifying Code (SMC) Event May Cause Unexpected Behavior</td>
</tr>
<tr>
<td>X11. x</td>
<td>No Fix</td>
<td></td>
<td>Code Segment Limit Violation May Occur on 4 Gigabyte Limit Check</td>
</tr>
<tr>
<td>No.</td>
<td>Stepping/Revision</td>
<td>Status</td>
<td>Errata</td>
</tr>
<tr>
<td>-----</td>
<td>------------------</td>
<td>---------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>B0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X12</td>
<td>x</td>
<td>No Fix</td>
<td>FST Instruction with Numeric and Null Segment Exceptions May Cause General Protection Faults to Be Missed and FP Linear Address (FLA) Mismatch</td>
</tr>
<tr>
<td>X13</td>
<td>x</td>
<td>No Fix</td>
<td>Page with PAT (Page Attribute Table) Set to USWC (Uncacheable Speculative Write Combine) While Associated MTRR (Memory Type Range Register) Is UC (Uncacheable) May Consolidate to UC</td>
</tr>
<tr>
<td>X14</td>
<td>x</td>
<td>No Fix</td>
<td>Under Certain Conditions LTR (Load Task Register) Instruction May Result in System Hang</td>
</tr>
<tr>
<td>X15</td>
<td>x</td>
<td>No Fix</td>
<td>Loading from Memory Type USWC (Uncacheable Speculative Write Combine) May Get Its Data Internally Forwarded from a Previous Pending Store</td>
</tr>
<tr>
<td>X16</td>
<td>x</td>
<td>No Fix</td>
<td>FXSAVE after FNINIT without an Intervening FP (Floating Point) Instruction May Save Uninitialized Values for FDP (x87 FPU Instruction Operand (Data) Pointer Offset) and FDS (x87 FPU Instruction Operand (Data) Pointer Selector)</td>
</tr>
<tr>
<td>X17</td>
<td>x</td>
<td>No Fix</td>
<td>FSTP (Floating Point Store) Instruction Under Certain Conditions May Result in Erroneously Setting a Valid Bit on an FP (Floating Point) Stack Register</td>
</tr>
<tr>
<td>X18</td>
<td>x</td>
<td>No Fix</td>
<td>An Execute Disable Bit Violation May Occur on a Data Page-Fault</td>
</tr>
<tr>
<td>X19</td>
<td>x</td>
<td>N/A</td>
<td>REMOVED</td>
</tr>
<tr>
<td>X20</td>
<td>x</td>
<td>No Fix</td>
<td>INIT Does Not Clear Global Entries in the TLB</td>
</tr>
<tr>
<td>X21</td>
<td>x</td>
<td>No Fix</td>
<td>Use of Memory Aliasing with Inconsistent Memory Type May Cause System Hang</td>
</tr>
<tr>
<td>X22</td>
<td>x</td>
<td>No Fix</td>
<td>Machine Check Exception May Occur When Interleaving Code between Different Memory Types</td>
</tr>
<tr>
<td>X23</td>
<td>x</td>
<td>N/A</td>
<td>REMOVED</td>
</tr>
<tr>
<td>X24</td>
<td>x</td>
<td>No Fix</td>
<td>General Protection (#GP) Fault May Not Be Signaled on Data Segment Limit Violation above 4-G Limit</td>
</tr>
<tr>
<td>X25</td>
<td>x</td>
<td>No Fix</td>
<td>DR3 Address Match on MOVQ/MOVQ/MOVNTQ Memory Store Instruction May Incorrectly Increment Performance Monitoring Count for Saturing SIMD Instructions Executed (Event B1h)</td>
</tr>
<tr>
<td>X26</td>
<td>x</td>
<td>No Fix</td>
<td>Pending x87 FPU Exceptions (#MF) Following STI May Be Serviced before Higher Priority Interrupts</td>
</tr>
<tr>
<td>X27</td>
<td>x</td>
<td>No Fix</td>
<td>Processor INIT# Will Cause a System Hang if Triggered during an NMI Interrupt Routine Performed during Shutdown</td>
</tr>
<tr>
<td>X28</td>
<td>x</td>
<td>No Fix</td>
<td>Certain Performance Monitoring Counters Related to Bus, L2 Cache and Power Management Are Inaccurate</td>
</tr>
<tr>
<td>X29</td>
<td>x</td>
<td>No Fix</td>
<td>CS Limit Violation on RSM May Be Serviced before Higher Priority Interrupts/Exceptions</td>
</tr>
<tr>
<td>X30</td>
<td>x</td>
<td>No Fix</td>
<td>A Write to an APIC Register Sometimes May Appear to Have Not Occurred</td>
</tr>
<tr>
<td>X31</td>
<td>x</td>
<td>No Fix</td>
<td>The Processor May Report a #TS Instead of a #GP Fault</td>
</tr>
<tr>
<td>X32</td>
<td>x</td>
<td>No Fix</td>
<td>BTS Message May Be Lost When the STPCLK# Signal Is Active</td>
</tr>
<tr>
<td>X33</td>
<td>x</td>
<td>No Fix</td>
<td>Last Exception Record (LER) MSR May Be Incorrectly Updated</td>
</tr>
<tr>
<td>X34</td>
<td>x</td>
<td>No Fix</td>
<td>Writing the Local Vector Table (LVT) When an Interrupt Is Pending May Cause an Unexpected Interrupt</td>
</tr>
<tr>
<td>X35</td>
<td>x</td>
<td>No Fix</td>
<td>Global Pages in the Data Translation Look-Aside Buffer (DTLB) May Not Be Flushed by RSM instruction before Restoring the Architectural State from SMRAM</td>
</tr>
<tr>
<td>X36</td>
<td>x</td>
<td>No Fix</td>
<td>Using 2M/4M Pages When A20M# Is Asserted May Result in Incorrect Address Translations</td>
</tr>
<tr>
<td>X37</td>
<td>x</td>
<td>No Fix</td>
<td>Premature Execution of a Load Operation Prior to Exception Handler Invocation</td>
</tr>
</tbody>
</table>

Table 4. IA-32 Core Errata (Sheet 2 of 3)
### Table 4. IA-32 Core Errata (Sheet 3 of 3)

<table>
<thead>
<tr>
<th>No.</th>
<th>Stepping/Revision</th>
<th>Status</th>
<th>Errata</th>
</tr>
</thead>
<tbody>
<tr>
<td>X38.</td>
<td>x</td>
<td>No Fix</td>
<td>#GP Fault is NOT Generated on Writing IA32_MISC_ENABLE [34] When Execute Disable (XD) is Not Supported</td>
</tr>
<tr>
<td>X39.</td>
<td>x</td>
<td>No Fix</td>
<td>SSE/SSE2 Streaming Store Resulting in a Self-Modifying Code (SMC) Event May Cause Unexpected Behavior</td>
</tr>
<tr>
<td>X40.</td>
<td>x</td>
<td>No Fix</td>
<td>Incorrect Address Computed for Last Byte of FXSAVE/FXRSTOR Image Leads to Partial Memory Update</td>
</tr>
<tr>
<td>X41.</td>
<td>x</td>
<td>No Fix</td>
<td>Values for LBR/BTS/BTM Will Be Incorrect after an Exit from SMM</td>
</tr>
<tr>
<td>X42.</td>
<td>x</td>
<td>No Fix</td>
<td>The BS Flag in DR6 May Be Set for Non-Single-Step #DB Exception</td>
</tr>
<tr>
<td>X43.</td>
<td>x</td>
<td>No Fix</td>
<td>BTM/BTS Branch-From Instruction Address May Be Incorrect for Software Interrupts</td>
</tr>
<tr>
<td>X44.</td>
<td>x</td>
<td>No Fix</td>
<td>Fault on ENTER Instruction May Result in Unexpected Values on Stack Frame</td>
</tr>
<tr>
<td>X45.</td>
<td>x</td>
<td>No Fix</td>
<td>Unaligned Accesses to Paging Structures May Cause the Processor to Hang</td>
</tr>
<tr>
<td>X46.</td>
<td>x</td>
<td>No Fix</td>
<td>INVLPG Operation for Large (2M/4M) Pages May be Incomplete under Certain Conditions</td>
</tr>
<tr>
<td>X47.</td>
<td>x</td>
<td>No Fix</td>
<td>Page Access Bit May be Set Prior to Signaling a Code Segment Limit Fault</td>
</tr>
<tr>
<td>X48.</td>
<td>x</td>
<td>No Fix</td>
<td>EFLAGS, CR0, CR4 and the EXF4 Signal May be Incorrect after Shutdown</td>
</tr>
<tr>
<td>X49.</td>
<td>x</td>
<td>No Fix</td>
<td>Store Ordering May be Incorrect between WC and WP Memory Types</td>
</tr>
</tbody>
</table>

### Table 5. Intel® EP80579 Integrated Processor Product Line Errata

<table>
<thead>
<tr>
<th>No.</th>
<th>Stepping/Revision</th>
<th>Status</th>
<th>Errata</th>
</tr>
</thead>
<tbody>
<tr>
<td>B0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>x</td>
<td>No Fix</td>
<td>Local Expansion Bus Controller may perform extra reads</td>
</tr>
<tr>
<td>2</td>
<td>x</td>
<td>No Fix</td>
<td>Thermal Diode does not accurately report temperature</td>
</tr>
<tr>
<td>3</td>
<td>x</td>
<td>No Fix</td>
<td>Gigabit Ethernet MAC Receive Timer interrupt problems</td>
</tr>
<tr>
<td>4</td>
<td>x</td>
<td>No Fix</td>
<td>Gigabit Ethernet MAC Large Segment Offload (LSO) premature descriptor write back</td>
</tr>
<tr>
<td>5</td>
<td>x</td>
<td>No Fix</td>
<td>Gigabit Ethernet MAC XOFF from link partner can pause flow-control (XON/XOFF) transmission</td>
</tr>
<tr>
<td>6</td>
<td>x</td>
<td>No Fix</td>
<td>Gigabit Ethernet MAC transmit descriptor use of Report Status (RS) bit for non-data (Context &amp; Null) descriptors</td>
</tr>
<tr>
<td>7</td>
<td>x</td>
<td>No Fix</td>
<td>Gigabit Ethernet MAC transmit descriptors may be written back to host, even without the Report Status (RS) bit set</td>
</tr>
<tr>
<td>8</td>
<td>x</td>
<td>No Fix</td>
<td>Gigabit Ethernet MAC legacy transmit descriptor write-back may occur before the packet data associated with the descriptor is fetched</td>
</tr>
<tr>
<td>9</td>
<td>x</td>
<td>No Fix</td>
<td>Gigabit Ethernet MAC may have EEPROM deadlock when using manual software EEPROM access</td>
</tr>
<tr>
<td>10</td>
<td>x</td>
<td>No Fix</td>
<td>The IEEE 1588-2008 logic does not lock snapshots for certain Layer-2 PTP Ethernet packets</td>
</tr>
<tr>
<td>11</td>
<td>x</td>
<td>No Fix</td>
<td>External Mastering of the Local Expansion Bus Controller has been Defeatured</td>
</tr>
<tr>
<td>12</td>
<td>x</td>
<td>No Fix</td>
<td>Gigabit Ethernet MAC0 PMCS.PS read only is nondeterministic</td>
</tr>
<tr>
<td>13</td>
<td>x</td>
<td>No Fix</td>
<td>Potential Power Sequencing Issue</td>
</tr>
</tbody>
</table>
### Table 6. Specification Changes

<table>
<thead>
<tr>
<th>No.</th>
<th>Stepping/Revision</th>
<th>Specification Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>#</td>
<td>#</td>
<td>None for this revision of this Specification Update.</td>
</tr>
</tbody>
</table>

### Table 7. Specification Clarifications

<table>
<thead>
<tr>
<th>No.</th>
<th>Stepping/Revision</th>
<th>Specification Clarifications</th>
</tr>
</thead>
<tbody>
<tr>
<td>B0</td>
<td>#</td>
<td>DC Characteristics for Gigabit Ethernet Inputs: SYS_PWR_OK and GBE_AUX_PWR_GOOD</td>
</tr>
</tbody>
</table>

### Table 8. Document-Only Changes

<table>
<thead>
<tr>
<th>No.</th>
<th>Document Title</th>
<th>Rev.</th>
<th>Document-Only Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Intel® EP80579 Integrated Processor Product Line Datasheet</td>
<td>320066-003</td>
<td>Figure 26-1 USB Port Connections Correction</td>
</tr>
<tr>
<td>2</td>
<td>Intel® EP80579 Integrated Processor Product Line Datasheet</td>
<td>320066-003</td>
<td>Table 49-63 RTC DC Clock Input Characteristics Correction</td>
</tr>
</tbody>
</table>
Identification Information

The Intel® EP80579 Integrated Processor Product Line can be identified by the following values:

<table>
<thead>
<tr>
<th>Extended Model</th>
<th>Family</th>
<th>Model</th>
<th>Brand ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>0001b</td>
<td>0110b</td>
<td>0101b</td>
<td>00000000b</td>
</tr>
</tbody>
</table>

Notes:
3. The Brand ID corresponds to bits [7:0] of the EBX register after the CPUID instruction is executed with a 1 in the EAX register.

The silicon stepping for Intel® EP80579 Integrated Processor Product Line can be identified by:
- Bus/Device/Function/Base: 0/31/0/MANID
- Address Offset: F8h
- Access: RO
- Size: 32 bit

<table>
<thead>
<tr>
<th>Bits</th>
<th>Symbol</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>23:16</td>
<td>Stepping Identifier</td>
<td>RO</td>
<td>Stepping Identifier: This field increments for each stepping of the part. <strong>Note:</strong> This field can be used by software to differentiate steppings when the Revision ID may not change. <strong>Note:</strong> 01h for B0 stepping</td>
</tr>
</tbody>
</table>
Component Marking Information

The Intel® EP80579 Integrated Processor Product Line may be identified by the following component markings:

**Figure 1. Markings**

![Component Markings](image)

- **GRP1LINE1**: Processor Number
- **GRP1LINE2**: This line will be blank
- **GRP1LINE3**: {Finished Process Order}
- **GRP1LINE4**: INTEL {M} {C} ‘YY{el}

**Table 9. Identification Table for the Intel® EP80579 (Sheet 1 of 2)**

<table>
<thead>
<tr>
<th>Processor Number</th>
<th>Material Master (MM)</th>
<th>Product Stepping</th>
<th>CPU Signature</th>
<th>Core Speed</th>
<th>Package</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>NU80579EZ600C</td>
<td>896645</td>
<td>B0</td>
<td>10650h</td>
<td>600MHz</td>
<td>1088 FCBGA</td>
<td>1</td>
</tr>
<tr>
<td>NU80579EZ600C</td>
<td>915599</td>
<td>B0</td>
<td>10650h</td>
<td>600MHz</td>
<td>1088 FCBGA</td>
<td>1, 2</td>
</tr>
<tr>
<td>NU80579EZ600CT</td>
<td>896644</td>
<td>B0</td>
<td>10650h</td>
<td>600MHz</td>
<td>1088 FCBGA</td>
<td>1</td>
</tr>
<tr>
<td>NU80579EZ600CT</td>
<td>915601</td>
<td>B0</td>
<td>10650h</td>
<td>600MHz</td>
<td>1088 FCBGA</td>
<td>1, 2</td>
</tr>
<tr>
<td>NU80579EZ004C</td>
<td>896643</td>
<td>B0</td>
<td>10650h</td>
<td>1066MHz</td>
<td>1088 FCBGA</td>
<td>1</td>
</tr>
<tr>
<td>NU80579EZ004C</td>
<td>915598</td>
<td>B0</td>
<td>10650h</td>
<td>1066MHz</td>
<td>1088 FCBGA</td>
<td>1, 2</td>
</tr>
<tr>
<td>NU80579EZ009C</td>
<td>896642</td>
<td>B0</td>
<td>10650h</td>
<td>1200MHz</td>
<td>1088 FCBGA</td>
<td>1</td>
</tr>
<tr>
<td>NU80579EZ009C</td>
<td>915597</td>
<td>B0</td>
<td>10650h</td>
<td>1200MHz</td>
<td>1088 FCBGA</td>
<td>1, 2</td>
</tr>
<tr>
<td>NU80579EB600C</td>
<td>89664</td>
<td>B0</td>
<td>10650h</td>
<td>600MHz</td>
<td>1088 FCBGA</td>
<td>1</td>
</tr>
<tr>
<td>NU80579EB600C</td>
<td>915596</td>
<td>B0</td>
<td>10650h</td>
<td>600MHz</td>
<td>1088 FCBGA</td>
<td>1, 2</td>
</tr>
</tbody>
</table>

**Notes:***
1. CPU signature is the value in the EAX registers after executing a CPUID instruction with EAX=01h.
2. Manufactured on 65 nm process.
### Table 9. Identification Table for the Intel® EP80579 (Sheet 2 of 2)

<table>
<thead>
<tr>
<th>Processor Number</th>
<th>Material Master (MM)</th>
<th>Product Stepping</th>
<th>CPU Signature</th>
<th>Core Speed</th>
<th>Package</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>NU80579ED004C</td>
<td>896639</td>
<td>B0</td>
<td>10650h</td>
<td>1066MHz</td>
<td>1088 FCBGA</td>
<td>1</td>
</tr>
<tr>
<td>NU80579ED004C</td>
<td>915595</td>
<td>B0</td>
<td>10650h</td>
<td>1066MHz</td>
<td>1088 FCBGA</td>
<td>1, 2</td>
</tr>
<tr>
<td>NU80579ED004CT</td>
<td>896640</td>
<td>B0</td>
<td>10650h</td>
<td>1066MHz</td>
<td>1088 FCBGA</td>
<td>1</td>
</tr>
<tr>
<td>NU80579ED004CT</td>
<td>915600</td>
<td>B0</td>
<td>10650h</td>
<td>1066MHz</td>
<td>1088 FCBGA</td>
<td>1, 2</td>
</tr>
<tr>
<td>NU80579ED009C</td>
<td>896638</td>
<td>B0</td>
<td>10650h</td>
<td>1200MHz</td>
<td>1088 FCBGA</td>
<td>1</td>
</tr>
<tr>
<td>NU80579ED009C</td>
<td>915594</td>
<td>B0</td>
<td>10650h</td>
<td>1200MHz</td>
<td>1088 FCBGA</td>
<td>1, 2</td>
</tr>
</tbody>
</table>

**Notes:**
1. CPU signature is the value in the EAX registers after executing a CPUID instruction with EAX=01h.
2. Manufactured on 65 nm process.
IA-32 Core Errata

X1. Code Segment (CS) Is Wrong on SMM Handler When SMBASE Is Not Aligned

Problem: With SMBASE being relocated to a non-aligned address, during SMM entry the CS can be improperly updated, which can lead to an incorrect SMM handler.

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

Workaround: Align SMBASE to 32 KB.

Status: No Fix

X2. IFU/BSU Deadlock May Cause System Hang

Problem: A lockable instruction with memory operand that spans across two pages may, given some rare internal conditions, hang the system.

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

Workaround: Lockable data should always be contained in a single page.

Status: No Fix

X3. Memory Aliasing with Inconsistent A and D Bits May Cause Processor Deadlock

Problem: In the event that software implements memory aliasing by having two page directory entries (PDEs) point to a common page table entry (PTE) and the Accessed and Dirty bits for the two PDEs are allowed to become inconsistent the processor may become deadlocked.

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

Workaround: Software that needs to implement memory aliasing in this way should manage the consistency of the Accessed and Dirty bits.

Status: No Fix

X4. RDMSR or WRMSR to Invalid MSR Address May Not Cause GP Fault

Problem: The RDMSR and WRMSR instructions allow reading or writing of MSR’s (Model Specific Registers) based on the index number placed in ECX. The processor should reject access to any reserved or unimplemented MSRs by generating #GP(0). However, there are some invalid MSR addressers for which the processor will not generate #GP(0). This erratum has not been observed with commercially available software.

Implication: For RDMSR, undefined values will be read into EDX:EAX. For WRMSR, undefined processor behavior may result.

Workaround: Do not use invalid MSR addresses with RDMSR or WRMSR.

Status: No Fix
X5. **Unable to Disable Reads/Writes to Performance Monitoring Related MSRs**

**Problem:** The Performance Monitoring Available bit in the miscellaneous processor features MSR (IA32_MISC_ENABLES.7) was defined so that when it is cleared to a 0, RDMSR/WRMSR/RDPMC instructions would return all zeros for reads of and prevent any write to Performance Monitoring related MSRs. Currently it is possible to read from or write to Performance Monitoring related MSRs when the Performance Monitoring Available bit is cleared to a 0.

**Implication:** It is not possible to disallow reads and writes to the Performance Monitoring MSRs. Intel has not observed this erratum with commercially available software or system.

**Workaround:** None.

**Status:** No Fix

X6. **Move to Control Register Instruction May Generate a Breakpoint Report**

**Problem:** A move (MOV) to Control register (CR) instruction where Control register is CR0, CR3 or CR4 may generate a breakpoint report.

**Implication:** MOV to Control Register Instruction is not expected to generate a breakpoint report.

**Workaround:** Ignore breakpoint data from MOV to CR instruction.

**Status:** No Fix

X7. **REMOVED**

X8. **Code Fetch Matching Disabled Debug Register May Cause Debug Exception**

**Problem:** The bits L0-3 and G0-3 enable breakpoints local to a task and global to all tasks, respectively. If one of these bits is set, a breakpoint is enabled, corresponding to the addresses in the debug registers DR0-DR3. If at least one of these breakpoints is enabled, any of these registers are disabled (i.e., Ln and Gn are 0), and RWn for the disabled register is 00 (indicating a breakpoint on instruction execution), normally an instruction fetch will not cause an instruction-breakpoint fault based on a match with the address in the disabled register(s). However, if the address in a disabled register matches the address of a code fetch which also results in a page fault, an instruction-breakpoint fault will occur.

**Implication:** While debugging software, extraneous instruction-breakpoint faults may be encountered if breakpoint registers are not cleared when they are disabled. Debug software which does not implement a code breakpoint handler will fail, if this occurs. If a handler is present, the fault will be serviced. Mixing data and code may exacerbate this problem by allowing disabled data breakpoint registers to break on an instruction fetch.

**Workaround:** The debug handler should clear breakpoint registers before they become disabled.

**Status:** No Fix

X9. **Upper Four PAT Entries Not Usable with Mode B or Mode C Paging**

**Problem:** The Page Attribute Table (PAT) contains eight entries, which must all be initialized and considered when setting up memory types for the Pentium M processor and Intel Processors A100 and A110. However, in Mode B or Mode C paging, the upper four entries do not function correctly for 4-Kbyte pages. Specifically, bit 7 of page table entries that translate addresses to 4-Kbyte pages should be used as the upper bit of a 3-bit index to determine the PAT entry that specifies the memory type for the page. When Mode B (CR4.PSE = 1) and/or Mode C (CR4.PAE) are enabled, the processor forces this bit to zero when determining the memory type regardless of the value in the page table entry. The upper four entries of the PAT function correctly for 2-Mbyte and
4-Mbyte large pages (specified by bit 12 of the page directory entry for those translations).

Implication: Only the lower four PAT entries are useful for 4-KB translations when Mode B or C paging is used. In Mode A paging (4-Kbyte pages only), all eight entries may be used. All eight entries may be used for large pages in Mode B or C paging.

Workaround: None.

Status: No Fix

**X10. SSE/SSE2 Streaming Store Resulting in a Self-Modifying Code (SMC) Event May Cause Unexpected Behavior**

Problem: An SSE or SSE2 streaming store that results in a Self-Modifying Code (SMC) event may cause unexpected behavior. The SMC event occurs on a full address match of code contained in L1 cache.

Implication: Due to this erratum, any of the following events may occur: A data access break point may be incorrectly reported on the instruction pointer (IP) just before the store instruction. A non-cacheable store can appear twice on the external bus (the first time it will write only 8 bytes, the second time it will write the entire 16 bytes). Intel has not observed this erratum with any commercially available software. This erratum has been seen in a synthetic test environment.

Workaround: None.

Status: No Fix

**X11. Code Segment Limit Violation May Occur on 4 Gigabyte Limit Check**

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

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

Workaround: Avoid code that wraps around segment limit.

Status: No Fix

**X12. FST Instruction with Numeric and Null Segment Exceptions May Cause General Protection Faults to Be Missed and FP Linear Address (FLA) Mismatch**

Problem: FST instruction combined with numeric and null segment exceptions may cause General Protection Faults to be missed and FP Linear Address (FLA) mismatch.

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

Workaround: None identified.

Status: No Fix

**X13. Page with PAT (Page Attribute Table) Set to USWC (Uncacheable Speculative Write Combine) While Associated MTRR (Memory Type Range Register) Is UC (Uncacheable) May Consolidate to UC**

Problem: A page whose PAT memory type is USWC while the relevant MTRR memory type is UC, the consolidated memory type may be treated as UC (rather than WC as specified in IA-32 Intel® Architecture Software Developer’s Manual).

Implication: When this erratum occurs, the memory page may be as UC (rather than WC). This may have a negative performance impact.

Workaround: None identified.
IA-32 Core Errata—Intel® EP80579 Integrated Processor Product Line

Status: No Fix

X14. Under Certain Conditions LTR (Load Task Register) Instruction May Result in System Hang

Problem: An LTR instruction may result in a system hang if all the following conditions are met:
Invalid data selector of the TR (Task Register) resulting with either #GP (General Protection Fault) or #NP (Segment Not Present Fault). GDT (Global Descriptor Table) is not 8-bytes aligned. Data BP (breakpoint) is set on cache line containing the descriptor data.

Implication: This erratum may result in system hang if all conditions have been met. This erratum has not been observed in commercial operating systems or software. For performance reasons, GDT is typically aligned to 8 bytes.

Workaround: Software should align GDT to 8-bytes.

Status: No Fix

X15. Loading from Memory Type USWC (Uncacheable Speculative Write Combine) May Get Its Data Internally Forwarded from a Previous Pending Store

Problem: A load from memory type USWC may get its data internally forwarded from a pending store. As a result, the expected load may never be issued to the external bus.

Implication: When this erratum occurs, a USWC load request may be satisfied without being observed on the external bus. There are no known usage models where this behavior results in any negative side-effects.

Workaround: Do not use memory type USWC for memory that has read sideeffects.

Status: No Fix

X16. FXSAVE after FNINIT without an Intervening FP (Floating Point) Instruction May Save Uninitialized Values for FDP (x87 FPU Instruction Operand (Data) Pointer Offset) and FDS (x87 FPU Instruction Operand (Data) Pointer Selector)

Problem: An FXSAVE after FNINIT without an intervening FP instruction may save uninitialized values for FDP and FDS.

Implication: When this erratum occurs, the values for FDP/FDS in the FXSAVE structure may appear to be random values. These values will be initialized by the first FP instruction executed after the FXRSTOR that restore the saved floating point state. Any FP instruction with memory operand will initialize FDP/FDS. Intel has not observed this erratum with any commercially available software.

Workaround: After an FINIT, do not expect the FXSAVE memory image to be correct, until at least one FP instruction with a memory operand has been executed.

Status: No Fix

X17. FSTP (Floating Point Store) Instruction Under Certain Conditions May Result In Erroneously Setting a Valid Bit on an FP (Floating Point) Stack Register

Problem: An FSTP instruction with a PDE/PTE (Page Directory Entry/Page Table Entry) A/D bit update followed by user mode access fault due to a code fetch to a page that has supervisor only access permission may result in erroneously setting a valid bit of an FP stack register. The FP top of stack pointer is unchanged.

Implication: This erratum may cause an unexpected stack overflow.

Workaround: User mode code should not count on being able to recover from illegal accesses to memory regions protected with supervisor only access when using FP instructions.
Status: No Fix

X18. **An Execute Disable Bit Violation May Occur on a Data Page-Fault**

Problem: Under a combination of internal events, unexpected Execute Disable violations may occur on data accesses that are Execute Disable protected.

Implication: This erratum may cause unexpected Execute Disable violations.

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

Status: No Fix

X19. **REMOVED**

X20. **INIT Does Not Clear Global Entries in the TLB**

Problem: INIT may not flush a TLB entry when: The processor is in protected mode with paging enabled and the page global enable flag is set (PGE bit of CR4 register) G bit for the page table entry is set TLB entry is present in TLB when INIT occurs

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

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

Status: No Fix

X21. **Use of Memory Aliasing with Inconsistent Memory Type May Cause System Hang**

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

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

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

Status: No Fix

X22. **Machine Check Exception May Occur When Interleaving Code between Different Memory Types**

Problem: A small window of opportunity exists where code fetches interleaved between different memory types may cause a machine check exception. A complex set of micro-architectural boundary conditions is required to expose this window.

Implication: Interleaved instruction fetches between different memory types may result in a machine check exception. The system may hang if machine check exceptions are disabled. Intel has not observed the occurrence of this erratum while running commercially available applications or operating systems.

Workaround: Software can avoid this erratum by placing a serializing instruction between code fetches between different memory types.

Status: No Fix
X23. **REMOVED**

X24. **General Protection (#GP) Fault May Not Be Signaled on Data Segment Limit Violation above 4-G Limit**

Problem: Memory accesses to flat data segments (base = 00000000h) that occur above the 4-G limit (0xffffffffh) may not signal a #GP fault.

Implication: When such memory accesses occur, the system may not issue a #GP fault.

Workaround: Software should ensure that memory accesses do not occur above the 4-G limit (0xffffffff).

Status: No Fix

X25. **DR3 Address Match on MOVQ/MOVQ/MOVQNTQ Memory Store Instruction May Incorrectly Increment Performance Monitoring Count for Saturating SIMD Instructions Executed (Event B1h)**

Problem: Performance monitoring for Event B1h normally increments on saturating SIMD instruction executed. Regardless of DR7 programming, if the linear address of a memory store MOVQ/MOVQ/MOVQNTQ instruction executed matches the address in DR3, the B1h counter may be incorrectly incremented.

Problem: The value observed for performance monitoring count for saturating SIMD instructions executed may be too high.

Workaround: None identified.

Status: No Fix

X26. **Pending x87 FPU Exceptions (#MF) Following STI May Be Serviced before Higher Priority Interrupts**

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

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

Workaround: None identified.

Status: No Fix

X27. **Processor INIT# Will Cause a System Hang if Triggered during an NMI Interrupt Routine Performed during Shutdown**

Problem: During the execution of an NMI interrupt handler, if shutdown occurs followed by the INIT# signal being triggered, the processor will attempt initialization but fail soft reset.

Implication: Due to this erratum the system may hang.

Workaround: None identified.

Status: No Fix

X28. **Certain Performance Monitoring Counters Related to Bus, L2 Cache and Power Management Are Inaccurate**

Problem: All Performance Monitoring Counters in the ranges 21H-3DH and 60H- 7FH may have inaccurate results up to ±7.

Implication: There may be a small error in the affected counts.

Workaround: None identified.

Status: No Fix
X29. CS Limit Violation on RSM May Be Serviced before Higher Priority Interrupts/Exceptions

Problem: When the processor encounters a CS (Code Segment) limit violation, a #GP (General Protection Exception) fault is generated after all higher priority Interrupts and exceptions are serviced. Because of this erratum, if RSM (Resume from System Management Mode) returns to execution flow where a CS limit violation occurs, the #GP fault may be serviced before a higher priority Interrupt or Exception (e.g., Non-Maskable Interrupt, Debug break(#DB), Machine Check (#MC), etc.).

Implication: Operating systems may observe a #GP fault being serviced before higher priority Interrupts and Exceptions.

Workaround: None identified.

Status: No Fix

X30. A Write to an APIC Register Sometimes May Appear to Have Not Occurred

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

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

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

Status: No Fix

X31. The Processor May Report a #TS Instead of a #GP Fault

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

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

Workaround: None identified.

Status: No Fix

X32. BTS Message May Be Lost When the STPCLK# Signal Is Active

Implication: STPCLK# is asserted to enable the processor to enter a low-power state. Under some circumstances, when STPCLK# becomes active, a pending BTS (Branch Trace Store) message may be either lost and not written or written with corrupted branch address to the Debug Store area.

Implication: BTS messages may be lost in the presence of STPCLK# assertions.

Workaround: None identified.

Status: No Fix
X33. **Last Exception Record (LER) MSRs May Be Incorrectly Updated**

**Problem:** The LASTINTTOIP and LASTINTFROMIP MSRs (1DDH-1DEH) may contain incorrect values after the following events: masked SSE2 floating-point exception, StopClk, NMI and INT.

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

**Workaround:** None identified.

**Status:** No Fix

X34. **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:** No Fix

X35. **Global Pages in the Data Translation Look-Aside Buffer (DTLB) May Not Be Flushed by RSM instruction before Restoring the Architectural State from SMRAM**

**Problem:** The Resume from System Management Mode (RSM) instruction does not flush global pages from the Data Translation Look-Aside Buffer (DTLB) prior to reloading the saved architectural state.

**Implication:** If SMM turns on paging with global paging enabled and then maps any of linear addresses of SMRAM using global pages, RSM load may load data from the wrong location.

**Workaround:** Do not use global pages in system management mode.

**Status:** No Fix

X36. **Using 2M/4M Pages When A20M# Is Asserted May Result in Incorrect Address Translations**

**Problem:** An external A20M# pin if enabled forces address bit 20 to be masked (forced to zero) to emulate real-address mode address wraparound at 1 megabyte. However, if all of the following conditions are met, address bit 20 may not be masked: paging is enabled a linear address has bit 20 set the address references a large page A20M# is enabled

**Implication:** When A20M# is enabled and an address references a large page the resulting translated physical address may be incorrect. This erratum has not been observed with any commercially available operating system.

**Workaround:** Operating systems should not allow A20M# to be enabled if the masking of address bit 20 could be applied to an address that references a large page. A20M# is normally only used with the first megabyte of memory.

**Status:** No Fix
X37. Premature Execution of a Load Operation Prior to Exception Handler Invocation

Problem: If any of the below circumstances occur it is possible that the load portion of the instruction will have executed before the exception handler is entered. If an instruction that performs a memory load causes a code segment limit violation If a waiting floating-point instruction or MMX instruction that performs a memory load has a floating-point exception pending If an MMX or SSE instruction that performs a memory load and has either CR0.EM=1 (Emulation bit set), or a floating-point Top-of-Stack (FP TOS) not equal to 0, or a DNA exception pending

Implication: In normal code execution where the target of the load operation is to write back memory there is no impact from the load being prematurely executed, nor from the restart and subsequent re-execution of that instruction by the exception handler. If the target of the load is to uncached memory that has a system side-effect, restarting the instruction may cause unexpected system behavior due to the repetition of the side-effect.

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

Status: No Fix

X38. #GP Fault is NOT Generated on Writing IA32_MISC_ENABLE [34] When Execute Disable (XD) is Not Supported

Problem: #GP fault is not generated on writing to IA32_MISC_ENABLE [34] bit in a processor which does not support Execute Disable (XD) functionality.

Implication: Writing to IA32_MISC_ENABLE [34] bit is silently ignored without generating a fault.

Workaround: None identified.

Status: No Fix

X39. SSE/SSE2 Streaming Store Resulting in a Self-Modifying Code (SMC) Event May Cause Unexpected Behavior

Problem: An SSE or SSE2 streaming store that results in a Self-Modifying Code (SMC) event may cause unexpected behavior. The SMC event occurs on a full address match of code contained in L1 cache.

Implication: Due to this erratum, any of the following events may occur: 6. A data access break point may be incorrectly reported on the instruction pointer (IP) just before the store instruction. 7. A non-cacheable store can appear twice on the external bus (the first time it will write only 8 bytes, the second time it will write the entire 16 bytes).

Workaround: None identified.

Status: No Fix

X40. Incorrect Address Computed for Last Byte of FXSAVE/FXRSTOR Image Leads to Partial Memory Update

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

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

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

Status: No Fix
X41. **Values for LBR/BTS/BTM Will Be Incorrect after an Exit from SMM**

**Problem:** After a return from SMM (System Management Mode), the CPU will incorrectly update the LBR (Last Branch Record) and the BTS (Branch Trace Store), hence rendering their data invalid. The corresponding data if sent out as a BTM on the system bus will also be incorrect. Note: This issue would only occur when one of the 3 above mentioned debug support facilities are used.

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

**Workaround:** None identified.

**Status:** No Fix

X42. **The BS Flag in DR6 May Be Set for Non-Single-Step #DB Exception**

**Problem:** DR6 BS (Single Step, bit 14) flag may be incorrectly set when the TF (Trap Flag, bit 8) of the EFLAGS Register is set, and a #DB (Debug Exception) occurs due to one of the following: DR7 GD (General Detect, bit 13) being bit set; INT1 instruction; Code breakpoint

**Implication:** The BS flag may be incorrectly set for non-single-step #DB exception.

**Workaround:** None identified.

**Status:** No Fix

X43. **BTM/BTS Branch-From Instruction Address May Be Incorrect for Software Interrupts**

**Problem:** When BTM (Branch Trace Message) or BTS (Branch Trace Store) is enabled, a software interrupt may result in the overwriting of BTM/BTS branch-from instruction address by the LBR (Last Branch Record) branch-from instruction address.

**Implication:** A BTM/BTS branch-from instruction address may get corrupted for software interrupts.

**Workaround:** None identified.

**Status:** No Fix

X44. **Fault on ENTER Instruction May Result in Unexpected Values on Stack Frame**

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

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

**Workaround:** None identified.

**Status:** No Fix

X45. **Unaligned Accesses to Paging Structures May Cause the Processor to Hang**

**Problem:** When an unaligned access is performed on paging structure entries, accessing a portion of two different entries simultaneously, the processor may live lock.

**Implication:** When this erratum occurs, the processor may live lock causing a system hang.
Workaround: Do not perform unaligned accesses on paging structure entries.
Status: No Fix

**X46. INVLP Gibson for Large (2M/4M) Pages May be Incomplete under Certain Conditions**

**Problem:** The INVLP Gibson instruction may not completely invalidate Translation Look-aside Buffer (TLB) entries for large pages (2M/4M) when both of the following conditions exist: Address range of the page being invalidated spans several Memory Type Range Registers (MTRRs) with different memory types specified INVLP Gibson operation is preceded by a Page Assist Event (Page Fault (#PF) or an access that results in either A or D bits being set in a Page Table Entry (PTE))

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

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

Status: No Fix

**X47. Page Access Bit May be Set Prior to Signaling a Code Segment Limit Fault**

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

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

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

Status: No Fix

**X48. EFLAGS, CR0, CR4 and the EXF4 Signal May be Incorrect after Shutdown**

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

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

**Workaround:** None identified.

Status: No Fix

**X49. Store Ordering May be Incorrect between WC and WP Memory Types**

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

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

**Workaround:** None identified.

Status: No Fix
1. **Local Expansion Bus Controller may perform extra reads**

Problem: In Micron* ZBT mode on reads, there are two extra reads to the last address.

Implication: When the Local Expansion Bus is in 16-bit Micron* ZBT mode and the transaction is a single read or burst read, the last address will be read two additional times as shown in Figure 2.

**Figure 2. Micron ZBT Read Errata**

![Diagram of Micron ZBT Read Errata]

Workaround: None.

Status: No Fix

2. **Thermal Diode does not accurately report temperature**

Problem: The on-die thermal diode (THERMDC and THERMDA) does not accurately report temperature.

Implication: Dynamic temperature monitoring will return inaccurate values. 

*Note:* THRMTRIP and PROCHOT are functional.

Workaround: Do not use the on-die thermal diode to monitor temperature.

Status: No Fix
3. **Gigabit Ethernet MAC Receive Timer interrupt problems**

**Problem:** When receive timers are used in the Gigabit Ethernet MAC, under certain timing and network traffic conditions, expiration of the timers may fail to generate an interrupt, set the Receive Timer interrupt bits in the Interrupt Cause Read registers (ICR0.RXTO, ICR1.RXTO) and/or may erroneously disable a subsequent re-start of the timers. There are two scenarios:

1. A receive timer expires while a descriptor writeback is pending and the writeback includes the descriptor of the packet most recently written to memory. In this case, the writeback completes normally and timers are halted, but the RXTO interrupt bits are not set. After the next packet is written to memory, an immediate descriptor writeback occurs, and the RXTO interrupt bits are set.

2. A receive timer expires while a packet is being written to memory. In this case, the descriptor writeback triggered by the timer expiration (which does not include the current packet's descriptor) is scheduled and timers are halted. When the current packet write completes, the writeback operation completes properly, and the RXTO interrupt bits are set, but the receive timers are disabled and will not re-start until the next packet is received.

**Implication:** If software uses the receive timer interrupt to detect the presence of incoming packets, lost or delayed timer interrupts may cause excessive latency in processing any packets that have been received since the last timer interrupt. This latency depends on network traffic conditions and could be quite large if receive traffic slows considerably.

**Workaround1:** Receive timers can be programmed for immediate interrupt generation whenever a packet is received by setting the RDTR.RPDT field to all 0s. Receive timer interrupt events can be accumulated by using the interrupt throttling registers ITR0 and ITR1 to avoid excessive interrupts.

**Workaround2:** Software can periodically flush accumulated descriptors by performing a write to the RDTR register with bit 31 set to a 1.

*Note:* Check with the software documentation to determine whether the software workarounds described here are implemented by software.

**Status:** No Fix

4. **Gigabit Ethernet MAC Large Segment Offload (LSO) premature descriptor write back**

**Problem:** For large send fetches ONLY (not normal or jumbo frames) the Gigabit Ethernet MAC internal DMA engine will decompose the large-send data fetch into a series of individual requests that are completed sequentially. When all read data associated with the first internal DMA request has been fetched, the descriptor is flagged as ready for writeback. Though all data associated with the entire Large Segment Offload descriptor will eventually be fetched, the descriptor writeback may occur prematurely. The device should wait until all bytes associated with the data descriptor have been completely fetched before writing back the transmit descriptor.

**Implication:** Due to premature write back, an operating system may release and reallocate the buffer, potentially causing buffer re-use and transmission of incorrect data.

**Workaround:** Utilize a second descriptor to point to the last four bytes of the large-send transmit data, and ensure that the buffer is not freed to the operating system/application until the second descriptor has been marked as complete via a status writeback operation.

*Note:* Check with the software documentation to determine whether the software workarounds described here are implemented by software.

**Status:** No Fix
5. **Gigabit Ethernet MAC XOFF from link partner can pause flow-control (XON/XOFF) transmission**

**Problem:** When the Gigabit Ethernet MAC transmitter is paused (by having received an XOFF from link partner), not only is the transmit of normal packets paused, but also of outbound XON/XOFF frames resulting from Receive Packet Buffer levels and Flow-Control Thresholds. Normally, partner’s XOFF packets only pause the LAN controller for a finite time interval, after which outbound XON/OFFs due to Receive Packet-Buffer fullness are again permitted to be sent.

**Implication:** If the transmitter is paused when a Receive FIFO XOFF threshold is reached, the transmission of XOFF frames does not occur, and Receive FIFO overrun may potentially occur, resulting in lost packets. This is only expected to be seen with an abnormally high pause time from link partner’s XOFF packet(s).

**Workaround:** Receive Flow-Control Thresholds may be tuned/lowered based on the maximum pause interval expected from link partner’s XOFF packet in order to minimize the likelihood of Receive FIFO overruns.

*Note:* Check with the software documentation to determine whether the software workarounds described here are implemented by software.

**Status:** No Fix

6. **Gigabit Ethernet MAC transmit descriptor use of Report Status (RS) bit for non-data (Context & Null) descriptors**

**Problem:** Due to a Gigabit Ethernet MAC internal logic error in the descriptor internal queue, if the internal descriptor queue becomes completely full of pending descriptor status writebacks, the descriptor logic may issue a writeback request with an incorrect writeback amount. The internal descriptor queue may accumulate pending writebacks if transmit descriptors that do not directly refer to transmit data buffers (e.g. context or Null descriptors) are submitted with a status-writeback request (RS asserted) and legacy writeback (status byte writeback only) is utilized.

**Implication:** The Gigabit Ethernet MAC may hang.

**Workaround:** Ensure that status-writeback reporting (RS) is not set on context or Null descriptors. Alternatively, utilize full-descriptor writebacks (TXDCTL.WTHRESH >= 1). The former workaround is the recommended alternative.

*Note:* Check with the software documentation to determine whether the software workarounds described here are implemented by software.

**Status:** No Fix

7. **Gigabit Ethernet MAC transmit descriptors may be written back to host, even without the Report Status (RS) bit set**

**Problem:** In the Gigabit Ethernet MAC, if the RS bit in the command (CMD) field of the transmit descriptor is set on at least some transmit descriptors submitted to the device, then it is possible that some other transmit descriptors without the RS bit set will be incorrectly written back to host memory.

**Implication:** The unnecessary descriptor write-backs will not cause a functional issue, but they may result in a small amount of unnecessary host bus bandwidth being consumed.

**Workaround:** None.

**Status:** No Fix

8. **Gigabit Ethernet MAC legacy transmit descriptor write-back may occur before the packet data associated with the descriptor is fetched**

**Problem:** If a legacy transmit operation directly follows a TCP Segmentation Offload transmit operation, the logic may incorrectly associate the successful completion of the TSO
transmit with the next descriptor. If the next descriptor is a legacy descriptor, under certain timing scenarios it is possible for the legacy descriptor to be incorrectly written back to host memory with the Descriptor Done (DD) bit set in the status (STATUS) field of the transmit descriptor. This might occur even though the packet data for the legacy descriptor has not yet been fetched.

Implication: Due to the premature write back, an operating system may release and reallocate the transmit buffer, potentially causing buffer re-use or transmission of incorrect data.

Workaround: Utilize at least two descriptors for any legacy transmit operation. Do not reallocate any buffers associated with the transmit operation until the last descriptor has been written back.

Note: Check with the software documentation to determine whether the software workarounds described here are implemented by software.

Status: No Fix

9. **Gigabit Ethernet MAC may have EEPROM deadlock when using manual software EEPROM access**

Problem: The EEPROM is a shared resource between four clients: Hardware Auto read, software accesses by Gigabit Ethernet port 0, Gigabit Ethernet port 1 and Gigabit Ethernet port 2 drivers. However, the software clients can control the EEPROM by successive writes to the EEPROM_CTRL register. In this case, there is a request/grant fixed priority arbitration method such that a software process can obtain exclusive access to the EEPROM. If this process were to hang or take an excessive amount of time to execute, other GbEs would be prevented from accessing the EEPROM, resulting in either a hung hardware state, or a timeout elsewhere in the system. In addition, no instructions will be processed if the GbE is in the D3 state or if the EEPROM is busy due to MAC access (such as caused by a reset of the GbE). If a process has claimed access to the EEPROM via the EEPROM_CTRL register for one GbE and does not release it between transactions, a new process controlling a different GbE could be started up and initiate a soft reset to that GbE. This GbE will not gain access to the EEPROM for its start-up initialization and so would become hung.

Implication: EEPROM deadlock can arise when software uses the EEPROM_CTRL (M:0:0, M:1:0 or M:2:0) register to access the EEPROM.

Workaround: Software must limit the length of time a process can hold the lock on the EEPROM when using the EEPROM_CTRL register for serial access. It is recommended that software release ownership of the EEPROM between each read/write transaction. Software must ensure that ownership is relinquished by the driver after accesses, or the operation of the other GbEs will be compromised.

Note: Check with the software documentation to determine whether the software workarounds described here are implemented by software.

Status: No Fix

10. **The IEEE 1588-2008 logic does not lock snapshots for certain Layer-2 PTP Ethernet packets**

Problem: When the IEEE 1588 logic is configured in Mode 8 (Timestamp user defined messages BM:D7:F0 register 40h, 60h, 80h, A0h, C0h, E0h, 100h or 120h) the 1588 logic does not recognize Layer-2 PTP packets with EtherType values in the range 0x0800 - 0x08FF.

Implication: If the IEEE 1588 logic is configured in Mode 8 (Timestamp user defined messages), timestamp capture will not happen for Layer-2 PTP packets with custom EtherType in the range of 0x0800 - 0x08FF (all other EtherType are supported). This errata does not affect Mode 0, 1, 2, and 3 (BM:D7:F0 register 40h, 60h, 80h, A0h, C0h, E0h, 100h or 120h)

Workaround: None
11. **External Mastering of the Local Expansion Bus Controller has been Defeatured**

- **Problem:** The external mastering feature of the Local Expansion Bus Controller (LEB) has been defeatured.
- **Implication:** Devices connected to the LEB can not master EP80579.
- **Workaround:** None
- **Status:** No Fix

12. **Gigabit Ethernet MAC0 PMCS.PS read only is nondeterministic**

- **Problem:** When the GbE EEPROM Initialization Control Word 1 bit 3 (ICW1.3) is 0 the Power Management Registers are not always read only (refer to EP80579 Datasheet 32066-003, Table 37-7 Initialization Control Word 1). This configuration is nondeterministic and the PMCS.PS register is not always read only.
- **Implication:** If ICW1.3 = 0, MAC0 transitions to D3, the system returns from S5 -> S0 and the Power Management Registers are set to read only then the CPU will hang if it reads from MAC0.
- **Workaround:** Set ICW1.3 = 1 to enable full support for power management and will ensure the BIOS or OS places MAC0 into D0 before trying to read from the device.
- **Status:** No Fix

13. **Potential Power Sequencing Issue**

- **Problem:** There is a power path between VCCPSUS/VCCGBEPSUS (3.3V sustain) and VCC1P2_USBSUS/VCCSUS1 (1.2V sustain) power wells.
- **Implication:** When power is applied to VCCPSUS and VCCGBEPSUS, the USBSUS and VCCSUS1 power rails are driven to approximately 500 mV. Depending on the power delivery design, this 500 mV might cause a violation of the EP80579 power up sequence.
- **Workaround:** Ensure that the power delivery design does not violate the EP80579 power up sequence defined in Section 6.1.3 *EP80579 Power Sequencing and Reset Sequence* of the Intel® *EP80579 Integrated Processor Product Line Datasheet* (320066-003).
- **Status:** No Fix.
Specification Changes

None for this revision of this Specification Update.
Specification Clarifications

1. **DC Characteristics for Gigabit Ethernet Inputs: SYS_PWR_OK and GBE_AUX_PWR_GOOD**

   All of the Gigabit Ethernet Inputs for the EP80579 are 3.3V tolerant, which also includes SYS_PWR_OK and GBE_AUX_PWR_GOOD. Therefore, the DC input characteristics defined in the 49.5.13.2 Gigabit Ethernet DC Characteristics section in the Intel® EP80579 Integrated Processor Product Line Datasheet (320066-003) applies to SYS_PWR_OK and GBE_AUX_PWR_GOOD as well.
Document-Only Changes

1. Figure 26-1 USB Port Connections Correction

Issue: In the Intel® EP80579 Integrated Processor Product Line Datasheet (320066-003) Figure 26-1 currently illustrates 4 USB ports. However, the EP80579 supports only 2 USB ports.

The content of Figure 26-1 will be updated in the next revision of the EP80579 Datasheet.

Old Text:

Figure 26-1. USB Port Connections
Figure 26-1. USB Port Connections

2. **Table 49-63 RTC DC Clock Input Characteristics Correction**

**Issue:**
Note 1 in Table 49-63 in the *Intel® EP80579 Integrated Processor Product Line Datasheet* (320066-003) incorrectly states “3.3V Clock input”, but input $V_{IH}$ max is 1.2V. The following clarification and corrections will be added to the next revision of EP80579 Datasheet.

Note 1 in Table 49-63 will be removed in the next release of the EP80579 datasheet.

**Old Text:**

Table 49-63.RTC DC Clock Input Characteristics (RTCX[2:1])

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter</th>
<th>Min</th>
<th>Typical</th>
<th>Max</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_{IL}$</td>
<td>Input voltage low</td>
<td>-</td>
<td>-</td>
<td>0.10</td>
<td>V</td>
</tr>
<tr>
<td>$V_{IH}$</td>
<td>Input voltage high</td>
<td>0.40</td>
<td>-</td>
<td>1.2</td>
<td>V</td>
</tr>
<tr>
<td>$C_L$</td>
<td>RTCX1 typical value</td>
<td>6</td>
<td></td>
<td></td>
<td>pf</td>
</tr>
<tr>
<td>$C_L$</td>
<td>RTCX2 typical value</td>
<td>6</td>
<td></td>
<td></td>
<td>pf</td>
</tr>
</tbody>
</table>

Notes:
1. 3.3 V clock Input
2. Guaranteed by design.

**New Text:**

Table 49-63.RTC DC Clock Input Characteristics (RTCX[2:1])

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter</th>
<th>Min</th>
<th>Typical</th>
<th>Max</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_{IL}$</td>
<td>Input voltage low</td>
<td>-</td>
<td>-</td>
<td>0.10</td>
<td>V</td>
</tr>
<tr>
<td>$V_{IH}$</td>
<td>Input voltage high</td>
<td>0.40</td>
<td>-</td>
<td>1.2</td>
<td>V</td>
</tr>
<tr>
<td>$C_L$</td>
<td>RTCX1 typical value</td>
<td>6</td>
<td></td>
<td></td>
<td>pf</td>
</tr>
<tr>
<td>$C_L$</td>
<td>RTCX2 typical value</td>
<td>6</td>
<td></td>
<td></td>
<td>pf</td>
</tr>
</tbody>
</table>

Notes:
1. Guaranteed by design.

**Affected Docs:** Intel® EP80579 Integrated Processor Product Line Datasheet, Revision 003 (320066-003).