Revision Guide for AMD Family 15h Models 30h-3Fh Processors

Publication # **51603**	Revision: **1.06**
Issue Date: **April 2014**	

Advanced Micro Devices

List of Figures

List of Tables

Revision History

April 2014	1.06	Added product information to Overview, Table 2 and Table 3, Table 4, and Table 8; Added erratum #792.
January 2014	1.02	Initial public release.

Overview

The purpose of the *Revision Guide for AMD Family 15h Models 30h-3Fh Processors* is to communicate updated product information to designers of computer systems and software developers. This revision guide includes information on the following products:

- AMD Elite Performance A-Series APU with Radeon™ Graphics
- AMD R-Series Mobile Accelerated Processing Unit (APU) with AMD Radeon™ HD Graphics
- AMD Opteron™ X1200 Series Processor
- AMD Opteron™ X2200 Series Processor

This guide consists of these major sections:

- Processor Identification shows how to determine the processor revision and workaround requirements, and to construct, program, and display the processor name string.
- Product Errata provides a detailed description of product errata, including potential effects on system operation and suggested workarounds. An erratum is defined as a deviation from the product's specification, and as such may cause the behavior of the processor to deviate from the published specifications.
- Documentation Support provides a listing of available technical support resources.

Revision Guide Policy

Occasionally, AMD identifies product errata that cause the processor to deviate from published specifications. Descriptions of identified product errata are designed to assist system and software designers in using the processors described in this revision guide. This revision guide may be updated periodically.

Conventions

Numbering

- **Binary numbers**. Binary numbers are indicated by appending a "b" at the end, e.g., 0110b.
- **Decimal numbers**. Unless specified otherwise, all numbers are decimal. This rule does not apply to the register mnemonics.
- **Hexadecimal numbers**. Hexadecimal numbers are indicated by appending an "h" to the end, e.g., 45F8h.
- **Underscores in numbers**. Underscores are used to break up numbers to make them more readable. They do not imply any operation. e.g., 0110_1100b.
- **Undefined digit**. An undefined digit, in any radix, is notated as a lower case "x".

Register References and Mnemonics

In order to define errata workarounds it is sometimes necessary to reference processor registers. References to registers in this document use a mnemonic notation consistent with that defined in the *BIOS and Kernel Developer's Guide (BKDG) for AMD Family 15h Models 30h-3Fh Processors*, order# 49125 . Each mnemonic is a concatenation of the register-space indicator and the offset of the register. The mnemonics for the various register spaces are as follows:

- IOXXX: x86-defined input and output address space registers; XXX specifies the byte address of the I/O register in hex (this may be 2 or 3 digits). This space includes the I/O-Space Configuration Address Register (IOCF8) and the I/O-Space Configuration Data Port (IOCFC) to access configuration registers.
- DZFYxXXX: PCI-defined configuration space at bus 0; Z specifies the PCI device address in hex; XXX specifies the byte address of the configuration register (this may be 2 or 3 digits) in hex; Y specifies the function number. For example, D18F3x40 specifies the register at bus 0, device 18h, function 3, address 40h. Some registers in D18F2xXXX have a _dct[1:0] mnemonic suffix, which indicates there is one instance per DRAM controller (DCT). The DCT instance is selected by DCT Configuration Select[DctCfgSel] (D18F1x10C[0]).
- DZFYxXXX_xZZZZZ: Port access through the PCI-defined configuration space at bus 0; Z specifies the PCI device address in hex; XXX specifies the byte address of the data port configuration register (this may be 2 or 3 digits) in hex; Y specifies the function number; ZZZZZ specifies the port address (this may be 2 to 7 digits) in hex. For example, D18F2x9C_x1C specifies the port 1Ch register accessed using the data port register at bus 0, device 18h, function 2, address 9Ch. Refer to the *BKDG* for access properties. Some registers in D18F2xXXX_xZZZZZ have a _dct[1:0] mnemonic suffix, which indicates there is one instance per DRAM controller (DCT). The DCT instance is selected by DCT Configuration Select[DctCfgSel] (D18F1x10C[0]).
- APICXXX: APIC memory-mapped registers; XXX is the byte address offset from the base address in hex (this may be 2 or 3 digits). The base address for this space is specified by the APIC Base Address Register (APIC_BAR) at MSR0000_001B.
- CPUID FnXXXX_XXXX_RRR_xYYY: processor capability information returned by the CPUID instruction where the CPUID function is XXXX_XXXX (in hex) and the ECX input is YYY (if specified). When a register is specified by RRR, the reference is to the data returned in that register. For example, CPUID Fn8000_0001_EAX refers to the data in the EAX register after executing CPUID instruction function 8000_0001h.
- MSRXXXX_XXXX: model specific registers; XXXX_XXXX is the MSR number in hex. This space is accessed through x86-defined RDMSR and WRMSR instructions.
- PMCxXXX[Y]: performance monitor events; XXX is the hexadecimal event counter number programmed into MSRC001_020[A,8,6,4,2,0][EventSelect] (PERF_CTL[5:0] bits 7:0). Y, when specified, signifies the unit mask programmed into MSRC001_020[A,8,6,4,2,0][UnitMask] (PERF_CTL[5:0] bits 15:8).

- NBPMCxXXX[Y]: northbridge performance monitor events; XXX is the hexadecimal event counter number programmed into MSRC001_024[6,4,2,0][EventSelect] (NB_PERF_CTL[3:0] bits 7:0). Y, when specified, signifies the unit mask programmed into MSRC001_024[6,4,2,0][UnitMask] (NB_PERF_CTL[3:0] bits 15:8).

Many register references use the notation "[]" to identify a range of registers. For example, D18F2x[1,0][4C:40] is a shorthand notation for D18F2x40, D18F2x44, D18F2x48, D18F2x4C, D18F2x140, D18F2x144, D18F2x148, and D18F2x14C.

Arithmetic and Logical Operators

In this document, formulas follow some Verilog conventions as shown in Table 1.

Table 1. Arithmetic and Logic Operators

Operator	Definition
{}	Curly brackets are used to indicate a group of bits that are concatenated together. Each set of bits is separated by a comma. E.g., {Addr[3:2], Xlate[3:0]} represents a 6-bit value; the two MSBs are Addr[3:2] and the four LSBs are Xlate[3:0].
|	Bitwise OR operator. E.g. (01b | 10b == 11b).
||	Logical OR operator. E.g. (01b || 10b == 1b); logical treats multibit operand as 1 if >=1 and produces a 1-bit result.
&	Bitwise AND operator. E.g. (01b & 10b == 00b).
&&	Logical AND operator. E.g. (01b && 10b == 1b); logical treats multibit operand as 1 if >=1 and produces a 1-bit result.
^	Bitwise exclusive-OR operator; sometimes used as "raised to the power of" as well, as indicated by the context in which it is used. E.g. (01b ^ 10b == 11b). E.g. (2^2 == 4).
~	Bitwise NOT operator (also known as one's complement). E.g. (~10b == 01b).
!	Logical NOT operator. E.g. (!10b == 0b); logical treats multibit operand as 1 if >=1 and produces a 1-bit result.
==	Logical "is equal to" operator.
!=	Logical "is not equal to" operator.
<=	Less than or equal operator.
>=	Greater than or equal operator.
*	Arithmetic multiplication operator.
/	Arithmetic division operator.
<<	Shift left first operand by the number of bits specified by the 2nd operand. E.g. (01b << 01b == 10b).
>>	Shift right first operand by the number of bits specified by the 2nd operand. E.g. (10b >> 01b == 01b).

Processor Identification

This section shows how to determine the processor revision, program and display the processor name string, and construct the processor name string.

Revision Determination

A processor revision is identified using a unique value that is returned in the EAX register after executing the CPUID instruction function 0000_0001h (CPUID Fn0000_0001_EAX). Figure 1 shows the format of the value from CPUID Fn0000_0001_EAX. In some cases, two or more processor revisions may exist within a stepping of a processor family and are identified by a unique value in D18F4x164 Fixed Errata Register (see D18F4x164 Fixed Errata Register).

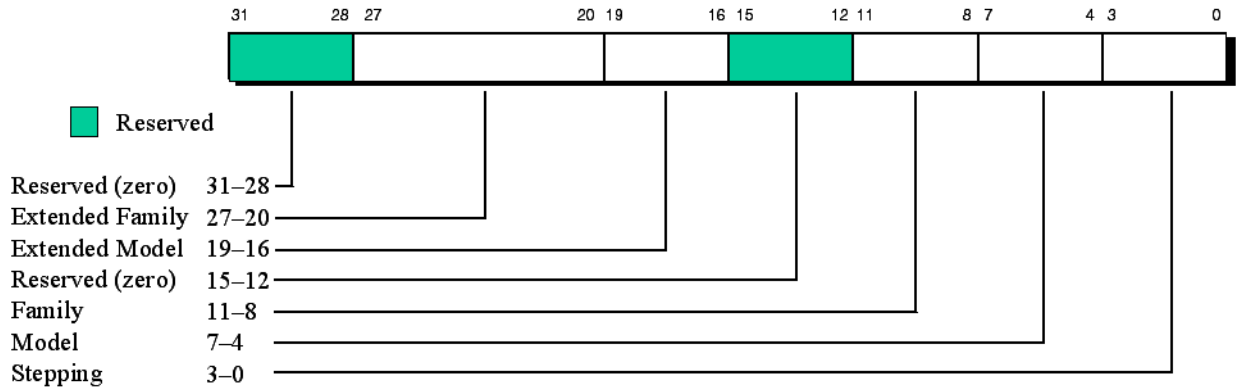

Figure 1. Format of CPUID Fn0000_0001_EAX

The following tables show the identification numbers from CPUID Fn0000_0001_EAX and D18F4x164 (if necessary) for each revision of the processor to each processor segment. "X" signifies that the revision has been used in the processor segment. "N/A" signifies that the revision has not been used in the processor segment.

Table 2. CPUID Values for AMD Family 15h Models 30h-3Fh FP3 Processor Revisions

CPUID Fn0000_0001_EAX (Mnemonic)	AMD Elite Performance A-Series APU	AMD R-Series Mobile APU	AMD Opteron™ X1200 Series Processor	AMD Opteron™ X2200 Series APU
00630F01h (KV-A1)	X	X	X	X

Table 3. CPUID Values for AMD Family 15h Models 30h-3Fh FM2r2 Processor Revisions

CPUID Fn0000_0001_EAX (Mnemonic)	AMD Elite Performance A-Series APU
00630F01h (KV-A1)	X

D18F4x164 Fixed Errata Register

Communicating the status of an erratum within a stepping of a processor family is necessary in certain circumstances. D18F4x164 is used to communicate the status of such an erratum fix so that BIOS or system software can determine the necessity of applying the workaround. Under these circumstances, the erratum workaround references the specified bit to enable software to test for the presence of the erratum. The erratum may be specific to some steppings of the processor, and the specified bit may or may not be set on other unaffected revisions within the same family. Therefore, software should use the CPUID Fn00000_0001_EAX extended model, model, and stepping as the first criteria to identify the applicability of an erratum. Once defined, the definition of the status bit will persist within the family of processors.

Bits	Description
31:0	0000_0000h. Reserved.

Graphic Device IDs

Processors with an integrated AMD Radeon HD Graphics Processing Engine use a graphics device ID at D1F0x00[31:16] to further identify the processor.

Table 4. AMD Family 15h Models 30h-3Fh Graphic Device IDs

D1F0x00[31:16	FM2r2	FP3	Notes
1309h		X	Notebook (6 SIMD & texture units, 2 render backends)
130Ah		X	Notebook (6 SIMD & texture units, 2 render backends)
130Bh		X	Notebook (3 SIMD & texture units, 1 render backend)
130Ch		X	Notebook (8 SIMD & texture units, 2 render backends)
130Dh		X	Notebook (6 SIMD & texture units, 2 render backends)
130Eh		X	Notebook (4 SIMD & texture units, 1 render backend)
1319h		X	Notebook (4 SIMD & texture units, 1 render backend)
130Fh	X		Desktop (8 SIMD & texture units, 2 render backends)
1310h	X		Workstation (8 SIMD & texture units, 2 render backends)
1311h	X		Workstation (8 SIMD & texture units, 2 render backends)

Table 4. AMD Family 15h Models 30h-3Fh Graphic Device IDs (continued)

D1F0x00[31:16	FM2r2	FP3	Notes
1313h	X		Desktop (6 SIMD & texture units, 2 render backends)
1315h	X		Desktop (4 SIMD & texture units, 1 render backend)
1316h	X		Desktop (3 SIMD & texture units, 1 render backend)
131Bh		X	Embedded (3 SIMD & texture units, 1 render backend)
131Ch		X	Embedded (8 SIMD & texture units, 2 render backends)
131Dh		X	Embedded (6 SIMD & texture units, 2 render backends)

Programming and Displaying the Processor Name String

This section, intended for BIOS programmers, describes how to program and display the 48-character processor name string that is returned by CPUID Fn8000_000[4:2]. The hardware or cold reset value of the processor name string is 48 ASCII NUL characters, so the BIOS must program the processor name string before any general purpose application or operating system software uses the extended functions that read the name string. It is common practice for the BIOS to display the processor name string and model number whenever it displays processor information during boot up.

Note: *Motherboards that do not program the proper processor name string and model number will not pass AMD validation and will not be posted on the AMD Recommended Motherboard Web site.*

The name string must be ASCII NUL terminated and the 48-character maximum includes that NUL character.

The processor name string is programmed by MSR writes to the six MSR addresses covered by the range MSRC001_00[35:30]h. Refer to the *BKDG* for the format of how the 48-character processor name string maps to the 48 bytes contained in the six 64-bit registers of MSRC001_00[35:30].

The processor name string is read by CPUID reads to a range of CPUID functions covered by CPUID Fn8000_000[4:2]. Refer to CPUID Fn8000_000[4:2] in the *BKDG* for the 48-character processor name string mapping to the 48 bytes contained in the twelve 32-bit registers of CPUID Fn8000_000[4:2].

Constructing the Processor Name String

This section describes how to construct the processor name string. BIOS forms the name string as follows:

1. If D18F5x198_x0 is 00000000h, then use a name string of "AMD Unprogrammed Engineering Sample" and skip the remaining steps.
2. Read {D18F5x198_x1, D18F5x198_x0} and write this value to MSRC001_0030.
3. Read {D18F5x198_x3, D18F5x198_x2} and write this value to MSRC001_0031.
4. Read {D18F5x198_x5, D18F5x198_x4} and write this value to MSRC001_0032.
5. Read {D18F5x198_x7, D18F5x198_x6} and write this value to MSRC001_0033.
6. Read {D18F5x198_x9, D18F5x198_x8} and write this value to MSRC001_0034.
7. Read {D18F5x198_xB, D18F5x198_xA} and write this value to MSRC001_0035.

Operating System Visible Workarounds

This section describes how to identify operating system visible workarounds.

MSRC001_0140 OS Visible Work-around MSR0 (OSVW_ID_Length)

This register, as defined in *AMD64 Architecture Programmer's Manual Volume 2: System Programming*, order# 24593, is used to specify the number of valid status bits within the OS Visible Work-around status registers.

The reset default value of this register is 0000_0000_0000_0000h.

BIOS shall program the OSVW_ID_Length to 0005h prior to hand-off to the OS.

Bits	Description
63:16	Reserved.
15:0	**OSVW_ID_Length: OS visible work-around ID length**. Read-write.

MSRC001_0141 OS Visible Work-around MSR1 (OSVW_Status)

This register, as defined in *AMD64 Architecture Programmer's Manual Volume 2: System Programming*, order# 24593, provides the status of the known OS visible errata. Known errata are assigned an OSVW_ID corresponding to the bit position within the valid status field.

Operating system software should use MSRC001_0140 to determine the valid length of the bit status field. For all valid status bits: 1=Hardware contains the erratum, and an OS software work-around is required or may be applied instead of a BIOS workaround. 0=Hardware has corrected the erratum, so an OS software work-around is not necessary.

The reset default value of this register is 0000_0000_0000_0000h.

Bits	Description
63:5	**OsvwStatusBits:** Reserved. OS visible work-around status bits. Read-write.
4	**OsvwId4:** Reserved, must be zero.
3	**OsvwId3:** Reserved, must be zero.
2	**OsvwId2:** Reserved, must be zero.
1	**OsvwId1:** Reserved, must be zero.
0	**OsvwId0:** Reserved, must be zero.

BIOS shall program the state of the valid status bits as shown in Table 5 prior to hand-off to the OS.

Table 5. Cross Reference of Product Revision to OSVW ID

CPUID Fn0000_0001_EAX (Mnemonic)	MSRC001_0141 Bits
00630F01h (KV-A1)	0000_0000_0000_0000h

Product Errata

This section documents product errata for the processors. A unique tracking number for each erratum has been assigned within this document for user convenience in tracking the errata within specific revision levels. This table cross-references the revisions of the part to each erratum. "No fix planned" indicates that no fix is planned for current or future revisions of the processor.

Note: There may be missing errata numbers. Errata that do not affect this product family do not appear. In addition, errata that have been resolved from early revisions of the processor have been deleted, and errata that have been reconsidered may have been deleted or renumbered.

Table 6. Cross-Reference of Processor Revision to Errata

No.	Errata Description	CPUID Fn0000_0001_EAX
		00630F01 (KV-A1)
361	Breakpoint Due to an Instruction That Has an Interrupt Shadow May Be Lost	X
638	Processor May Violate Trp During Dynamic Mode Switch	X
732	IOMMU Event Log Ordering Violation	X
733	IOMMU PPR Log Ordering Violation	X
737	Processor Does Not Check 128-bit Canonical Address Boundary Case on Logical Address	X
746	IOMMU Logging May Stall Translations	X
775	Processor May Present More Than One #DB Exception on REP-INS or REP-OUTS Instructions	X
786	APIC Timer Periodic Mode is Imprecise	X
787	IOMMU Interrupt May Be Lost	X
792	DRAM Scrubbing May Overwrite CC6 Core Save State Data Resulting in Unpredictable System Behavior	X

Cross-Reference of Errata to Package Type

This table cross-references the errata to each package type. "X" signifies that the erratum applies to the package type. An empty cell signifies that the erratum does not apply. An erratum may not apply to a package type due to a specific characteristic of the erratum, or it may be due to the affected silicon revision(s) not being used in this package.

Table 7. Cross-Reference of Errata to Package Type

Errata	Package	
	FM2r2	FP3
361	X	X
638	X	X
732	X	X
733	X	X
737	X	X
746	X	X
775	X	X
786	X	X
787	X	X
792	X	X

Cross-Reference of Errata to Processor Segments

This table cross-references the errata to each processor segment. "X" signifies that the erratum applies to the processor segment. An empty cell signifies that the erratum does not apply. An erratum may not apply to a processor segment due to a specific characteristic of the erratum, or it may be due to the affected silicon revision(s) not being used in this processor segment.

Table 8. Cross-Reference of Errata to Processor Segments

Errata	Processor Segment			
	AMD Elite Performance A-Series APU	AMD Opteron™ X1200 Series Processor	AMD Opteron™ X2200 Series APU	AMD R-Series Mobile APU
361	X	X	X	X
638	X	X	X	X
732	X	X	X	X
733	X	X	X	X
737	X	X	X	X
746	X	X	X	X
775	X	X	X	X
786	X	X	X	X
787	X	X	X	X
792		X	X	X

361 Breakpoint Due to an Instruction That Has an Interrupt Shadow May Be Lost

Description

A #DB exception occurring in guest mode may be discarded under the following conditions:

- A trap-type #DB exception is generated in guest mode during execution of an instruction with an interrupt shadow, and
- The instruction that generated the exception is immediately followed by an instruction resulting in #VMEXIT.

Potential Effect on System

None expected under normal conditions. Debug exceptions may not be received for programs running under a hypervisor.

Suggested Workaround

None.

Fix Planned

No fix planned

638 Processor May Violate Trp During Dynamic Mode Switch

Description

The processor may violate the precharge time (Trp) for a DIMM when sending a mode register set (MRS) command to dynamically adjust MR0[PPD] during a precharge power down.

This erratum may occur only when fast exit/slow exit (dynamic) mode is selected as follows:

- D18F2x94_dct[1:0][15] = 1b (DRAM Configuration High[PowerDownEn])
- D18F2x84_dct[1:0][23] = 1b (DRAM MRS[PchgPDModeSel])

Potential Effect on System

Unpredictable system operation.

Suggested Workaround

If D18F2x84_dct[1:0] bit 23 (PchgPDModeSel) = 1b and D18F2x94_dct[1:0] bit 15 (PowerDownEn) = 1b, then precharge time (D18F2x200_dct[1:0]_mp[1:0] bits 20:16, Trp) should be set one higher than the DIMM specified value.

Fix Planned

No fix planned

732 IOMMU Event Log Ordering Violation

Description

The processor IOMMU does not maintain producer-consumer ordering between the IOMMU event log DMA writes and IOMMU MMIO register read completions. The processor core may read stale or uninitialized event logs from memory when a read response from the event log tail pointer register passes the corresponding event log DMA write. A series or burst of event log DMA writes would normally be necessary for this ordering violation to be observed.

Potential Effect on System

Software may process an event log before it has been completely written, possibly resulting in the operating system or hypervisor taking improper corrective actions.

Suggested Workaround

The IOMMU driver of the hypervisor or operating system should initialize the event log buffer to all zeros and write event log entries to zero after they are processed. If software subsequently observes an all zero event log entry, it should re-read the buffer until a non-zero event log is returned. It is recommended that software detects that the log buffer has not been written by checking for an EventCode (bits 63:60) that is equal to 0000b.

Fix Planned

No fix planned

733 IOMMU PPR Log Ordering Violation

Description

The processor IOMMU does not maintain producer-consumer ordering between the IOMMU peripheral page service request (PPR) log DMA writes and IOMMU MMIO register read completions. The processor core may read stale or uninitialized PPR logs from memory when a read response from the PPR log tail pointer register passes the corresponding PPR log DMA write. A series or burst of PPR log DMA writes would normally be necessary for this ordering violation to be observed.

This erratum only applies in systems where a device is performing Address Translation Service (ATS) requests.

Potential Effect on System

Software may process a PPR log before it has been completely written, possibly resulting in the IOMMU software not properly processing a page service request. This may result in unpredictable IOMMU behavior.

Suggested Workaround

The IOMMU driver of the hypervisor or operating system should initialize the PPR log buffer to all zeros and write PPR log entries to zero after they are processed. If software subsequently observes an all zero PPR log entry, it should re-read the buffer until a non-zero PPR log is returned. It is recommended that software detects that the log buffer has not been written by checking for a PPRCode (bits 63:60) that is equal to 0000b.

Fix Planned

No fix planned

737 Processor Does Not Check 128-bit Canonical Address Boundary Case on Logical Address

Description

The processor core may not detect a #GP exception if the processor is in 64-bit mode and the logical address of a 128-bit operation (for example, a octal-word SSE instruction) is canonical on the first byte, but whose final byte crosses over the canonical address boundary. The processor does check the linear address and signals a #GP exception if the linear address is not canonical (for all eight bytes of the operation). Therefore, this erratum can only occur if the segment register is non-zero and causes a wrap in the logical address space only.

In the unlikely event that software causes this wrap, the processor core will execute the 128-bit operation as if the second part of the misaligned access starts at linear address equal to zero.

Potential Effect on System

None expected, as the normal usage of segment registers and segment limits does not expose this erratum.

Suggested Workaround

None required.

Fix Planned

No fix planned

746 IOMMU Logging May Stall Translations

Description

The processor IOMMU L2B miscellaneous clock gating feature may cause the IOMMU to stop processing IOMMU translations due to a perceived lack of credits for writing upstream command processor writebacks, MSI interrupts, peripheral page service request (PPR) logs, or event logs.

Potential Effect on System

The IOMMU may hang. This may be observed as a system hang.

Suggested Workaround

BIOS should disable L2B miscellaneous clock gating by setting L2_L2B_CK_GATE_CONTROL[CKGateL2BMiscDisable] (D0F2xF4_x90[2]) = 1b.

Fix Planned

No fix planned

775 Processor May Present More Than One #DB Exception on REP-INS or REP-OUTS Instructions

Description

When a REP-INSx or REP-OUTSx instruction is interrupted by a system management interrupt (SMI), the processor does not set RFLAGS.RF to 1b in the SMM save state. After the SMM code executes RSM to return from SMM, any debug exception present on the instruction may get presented again.

Potential Effect on System

Debug software may observe two or more #DB exceptions for a single execution of REP-INS or REP-OUTS instruction.

Suggested Workaround

None.

Fix Planned

No fix planned

786 APIC Timer Periodic Mode is Imprecise

Description

The APIC timer may not properly initialize back to the APIC timer initial count value (APIC380) when it transitions to zero and Timer Local Vector Table Entry[Mode] (APIC320[17]) is configured to run in periodic mode. In this mode, when the APIC timer reaches zero, the next value in the APIC current count register (APIC390) is set to the APIC initial count (APIC380), but the processor may incorrectly add or subtract an offset that is between 0 and 31.

Potential Effect on System

The standard use of the APIC timer and the level of accuracy required does not make this error significant.

Suggested Workaround

None.

Fix Planned

No fix planned

787 IOMMU Interrupt May Be Lost

Description

The IOMMU may not send a message-signaled interrupt (MSI) if the interrupt becomes pending around a very small timing window while the software is clearing a previous IOMMU interrupt. Once this erratum occurs, message-signaled interrupts are not sent until the software clears the interrupt pending status.

Potential Effect on System

The IOMMU driver may not receive IOMMU interrupts. This effect may persist and cause the IOMMU to stop processing transactions since the software normally does not clear the interrupt pending status in the absence of an actual interrupt.

AMD has only observed this erratum with IOMMU version 2 software.

Suggested Workaround

No workaround is recommended for IOMMU version 1 driver software.

After clearing interrupts, the IOMMU version 2 driver software should read the interrupt status register and process any pending interrupts.

Fix Planned

No fix planned

792 DRAM Scrubbing May Overwrite CC6 Core Save State Data Resulting in Unpredictable System Behavior

Description

The processor does not properly ensure that a DRAM scrub read and write sequence is atomic with respect to simultaneous processor core accesses to the CC6 save area. If a DRAM scrub access is to the same address as a concurrent save of the processor state, the CC6 entry may appear as if it was not written.

Potential Effect on System

Unpredictable system behavior.

Suggested Workaround

BIOS should set Scrub Rate Control D18F3x58[4:0] = 00h to disable sequential DRAM scrubbing. BIOS should set DRAM Scrub Address Low D18F3x5C[0] = 0b to disable re-direct DRAM scrubbing.

Fix Planned

No fix planned

Documentation Support

The following documents provide additional information regarding the operation of the processor:

- *BIOS and Kernel Developer's Guide (BKDG) for AMD Family 15h Models 30h-3Fh Processors*, order# 49125
- *AMD64 Architecture Programmer's Manual Volume 1: Application Programming*, order# 24592
- *AMD64 Architecture Programmer's Manual Volume 2: System Programming*, order# 24593
- *AMD64 Architecture Programmer's Manual Volume 3: General-Purpose and System Instructions*, order# 24594
- *AMD64 Architecture Programmer's Manual Volume 4: 128-Bit and 256-Bit Media Instructions*, order# 26568
- *AMD64 Architecture Programmer's Manual Volume 5: 64-Bit Media and x87 Floating-Point Instructions*, order# 26569

See the AMD Web site at www.amd.com for the latest updates to documents.