Revision Guide for AMD Family 16h Models 00h-0Fh Processors

Publication # **51810**	Revision: **3.00**
Issue Date: **June 2013**	

Advanced Micro Devices

Trademarks

Disclaimer

List of Figures

List of Tables

Revision History

Date	Revision	Description
June 2013	3.00	Initial public release.

Overview

The purpose of the *Revision Guide for AMD Family 16h Models 00h-0Fh 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 A-Series Mobile Accelerated Processing Unit (APU) with AMD Radeon™ HD Graphics
- AMD E-Series Mobile Accelerated Processing Unit (APU) with AMD Radeon HD Graphics
- AMD G-Series Mobile Accelerated Processing Unit (APU) with AMD Radeon HD Graphics
- AMD G-Series System on a Chip (SOC)
- AMD Opteron™ X1100 Series Processor
- AMD Opteron X2100 Series Accelerated Processing Unit (APU) with AMD Radeon HD Graphics

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 16h Models 00h-0Fh Processors*, order# 48751. 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]). Some registers in D18F2xXXX have an _mp[1:0] mnemonic suffix, which indicates there is one instance per memory P-state. The memory P-state instance is selected by DCT Configuration Select[MemPsSel] (D18F1x10C[3]).
- 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 *BIOS and Kernel Developer's Guide (BKDG) for AMD Family 16h Models 00h-0Fh Processors*, order# 48751 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]). Some registers in D18F2xXXX_xZZZZZ have an _mp[1:0] mnemonic suffix, which indicates there is one instance per memory P-state. The memory P-state instance is selected by DCT Configuration Select[MemPsSel] (D18F1x10C[3]).
- 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).

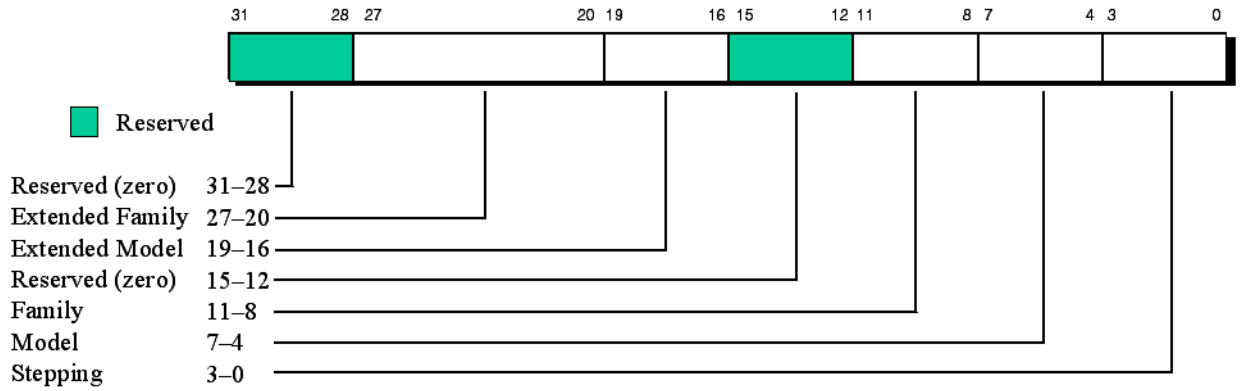

Figure 1. Format of CPUID Fn0000_0001_EAX

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 16h Models 00h-0Fh FT3 Processor Revisions

CPUID Fn0000_0001_EAX (Mnemonic)	AMD A-Series Mobile APU	AMD E-Series Mobile APU	AMD G-Series Mobile APU	AMD G-Series SOC	AMD Opteron™ X1100 Series Processor	AMD Opteron™ X2100 Series APU
00700F01h (KB-A1)	X	X	X	X	X	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 3 lists the graphic device ID values in use for AMD Family 16h Models 00h-0Fh Accelerated Processing Units.

Table 3. AMD Family 16h Graphic Device IDs

D1F0x00[31:16]	FT3	Notes
9830h	X	Notebook/Desktop (2 quad pipes, 2 SIMD & texture units, 1 render backend)
9831h	X	Embedded (2 quad pipes, 2 SIMD & texture units, 1 render backend)
9832h	X	Notebook/Desktop (2 quad pipes, 2 SIMD & texture units, 1 render backend)
9833h	X	Embedded (2 quad pipes, 2 SIMD & texture units, 1 render backend)
9834h	X	Notebook (2 quad pipes, 2 SIMD & texture units, 1 render backend)
9835h	X	Embedded (2 quad pipes, 2 SIMD & texture units, 1 render backend)
9836h	X	Notebook/Desktop (2 quad pipes, 2 SIMD & texture units, 1 render backend)
9837h	X	Embedded (2 quad pipes, 2 SIMD & texture units, 1 render backend)
9838h	X	Notebook/Desktop (2 quad pipes, 2 SIMD & texture units, 1 render backend)
9839h	X	Tablet (2 quad pipes, 2 SIMD & texture units, 1 render backend)
983Ah	X	Embedded (2 quad pipes, 2 SIMD & texture units, 1 render backend)
983Dh	X	Tablet (2 quad pipes, 2 SIMD & texture units, 1 render backend)

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 *BIOS and Kernel Developer's Guide (BKDG) for AMD Family 16h Models 00h-0Fh Processors*, order# 48751, 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 *BIOS and Kernel Developer's Guide (BKDG) for AMD Family 16h Models 00h-0Fh Processors*, order# 48751, 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 4 prior to hand-off to the OS.

Table 4. Cross Reference of Product Revision to OSVW ID

CPUID Fn0000_0001_EAX (Mnemonic)	MSRC001_0141 Bits
00700F01h (KB-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 5. Cross-Reference of Processor Revision to Errata

No.	Errata Description	CPUID Fn0000_0001_EAX
		00700F01h (KB-A1)
77	Long Mode CALLF or JMPF May Fail To Signal GP When Callgate Descriptor is Beyond GDT/LDT Limit	No fix planned
361	Breakpoint Due to an Instruction That Has an Interrupt Shadow May Be Lost	No fix planned
541	IBS Registers May be Unpredictable After CC6 State	No fix planned
638	Processor May Violate Trp During Dynamic Mode Switch	No fix planned
737	Processor Does Not Check 128-bit Canonical Address Boundary Case on Logical Address	No fix planned
756	Machine Check Information May Show Inconsistent Signature from an Older Corrected Error	No fix planned
757	L2 Tag Error Machine Check Status May Be Incorrect	No fix planned
767	Processor APM Behavior May Be Incorrect After CC6	No fix planned
776	Incorrect Processor Branch Prediction for Two Consecutive Linear Pages	No fix planned
778	Processor Core Time Stamp Counters May Experience Drift	No fix planned
779	Initial Time Stamp Counter Frequency May Be Incorrect	No fix planned
780	Processor May Cache Guest Write Combining Memory Type	No fix planned
781	Improper Handling of ECX Value During RDPMC Instruction	No fix planned
785	USB Interrupt Status May Not Be Set After a Short Packet	No fix planned
786	APIC Timer Periodic Mode is Imprecise	No fix planned

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 6. Cross-Reference of Errata to Package Type

Errata	Package
	FT3
77	X
361	X
541	X
638	X
737	X
756	X
757	X
767	X
776	X
778	X
779	X
780	X
781	X
785	X
786	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 7. Cross-Reference of Errata to Processor Segments

Errata	Processor Segment					
	AMD A-Series Mobile APU	AMD E-Series Mobile APU	AMD G-Series Mobile APU	AMD G-Series SOC	AMD Opteron™ X1100 Series Processor	AMD Opteron™ X2100 Series APU
77	X	X	X	X	X	X
361	X	X	X	X	X	X
541	X	X	X	X	X	X
638	X	X	X	X	X	X
737	X	X	X	X	X	X
756	X	X	X	X	X	X
757	X	X	X	X	X	X
767	X	X	X	X	X	X
776	X	X	X	X	X	X
778	X	X	X	X	X	X
779	X	X	X	X	X	X
780	X	X	X	X	X	X
781	X	X	X	X	X	X
785	X	X	X	X	X	X
786	X	X	X	X	X	X

77 Long Mode CALLF or JMPF May Fail To Signal GP When Callgate Descriptor is Beyond GDT/LDT Limit

Description

If the target selector of a far call or far jump (CALLF or JMPF) instruction references a 16-byte long mode system descriptor where any of the last 8 bytes are beyond the GDT or LDT limit, the processor fails to report a General Protection fault.

Potential Effect on System

None expected, since the operating system typically aligns the GDT/LDT limit such that all descriptors are legal. However, in the case of erroneous operating system software, the above described GP fault will not be signaled, resulting in unpredictable system failure.

Suggested Workaround

None required, it is anticipated that long mode operating system software will ensure the GDT and LDT limits are set high enough to cover the larger (16-byte) long mode system descriptors.

Fix Planned

No fix planned

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

541 IBS Registers May be Unpredictable After CC6 State

Description

The following Instruction-Based Sampling (IBS) registers may be unpredictable after the processor core exits the core C6 (CC6) state:

- Read-only bits MSRC001_1030 IBS Fetch Control Register
- MSRC001_1031 IBS Fetch Linear Address Register
- MSRC001_1032 IBS Fetch Physical Address Register
- MSRC001_1034 IBS Op Logical Address Register
- MSRC001_1035 IBS Op Data Register
- MSRC001_1036 IBS Op Data 2 Register
- MSRC001_1037 IBS Op Data 3 Register
- MSRC001_1038 IBS DC Linear Address Register
- MSRC001_1039 IBS DC Physical Address Register
- MSRC001_103B IBS Branch Target Address Register

The registers are predictable as long as IBS is not enabled at the time that the processor core enters CC6 state.

Potential Effect on System

In cases where the performance monitoring software fetches the IBS sampled data and the processor core has entered the CC6 state since this sample, the performance monitoring software may observe unpredictable values and may generate inaccurate results. The performance monitoring software would normally consume the sampled IBS data before a CC6 entry occurs, resulting in no observed effect under normal conditions.

Suggested Workaround

Performance monitoring software should avoid entering ACPI sleep states (C1/HALT or C2) prior to accessing the IBS registers.

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

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

756 Machine Check Information May Show Inconsistent Signature from an Older Corrected Error

Description

The processor core may not properly overwrite machine check architecture (MCA) information from an older corrected error in MC1_STATUS (MSR0000_0405) and MC1_ADDRESS (MSR0000_0406) when presenting a machine check exception (#MC) for a newer uncorrected error. Although the processor does set processor context corrupt (MC1_STATUS[PCC] bit 57) to 1b and error uncorrected status (MC1_STATUS[UC], bit 61) to 1b, it does not update the remaining bits in the registers.

Potential Effect on System

The machine check handler may be presented with status and address registers that are not consistent with the cause of the uncorrected error. It is expected that the MCA handler will respond to a #MC when MC1_STATUS[PCC] is set with an operating system panic or crash, regardless of the remaining bits in MC1_STATUS.

However, the information pertaining to the actual uncorrected error may be lost. In addition, debug engineers may note that the error signature does not align to MC1 error signatures documented in the BIOS and Kernel Developer's Guide (BKDG), as the error signature will match a possibly unrelated corrected error except for the UC and PCC bits. This effect only occurs if a processor reports both corrected and uncorrected errors.

Suggested Workaround

None.

Fix Planned

No fix planned

757 L2 Tag Error Machine Check Status May Be Incorrect

Description

Under a highly specific and detailed set of internal timing conditions, the MC2 status information (MC2_STATUS, MSR0000_0409) for a corrected or uncorrected L2 tag error may not indicate the actual operation that was occurring at the time the error was detected. The status information may report that the processor was performing an L2 fill operation when in reality the error was detected while processing a probe.

The MC2_STATUS fields that identify the possibility that this erratum applies (i.e., when MC2_STATUS may improperly indicate a fill operation) are:

- MC2_STATUS[Valid] (bit 63) = 1b
- MC2_STATUS[UC] (bit 61) = 0b or 1b (may be a corrected error or an uncorrected error)
- MC2_STATUS[ErrorCodeExt] (bits 20:16) = 0Bh or 0Fh
- MC2_STATUS[RRRR] (bits 7:4) = 0001b

Potential Effect on System

None expected.

Suggested Workaround

None required.

Fix Planned

No fix planned

767 Processor APM Behavior May Be Incorrect After CC6

Description

The processor core may not properly restore internal Application Power Management registers after it exits the core C6 (CC6) state.

Potential Effect on System

The processor may exceed the specified thermal design power (TDP) or thermal design current (TDC).

Suggested Workaround

Contact your AMD representative for information on a BIOS update.

Fix Planned

No fix planned

776 Incorrect Processor Branch Prediction for Two Consecutive Linear Pages

Description

Under a highly specific and detailed set of internal timing conditions, the processor core may incorrectly fetch instructions when the instruction pointer (rIP) changes (via a branch or other call, return) between two consecutive linear address 4K pages with the same offset in rIP[11:6].

Potential Effect on System

Unpredictable system behavior.

Suggested Workaround

BIOS should set MSRC001_1021[26] = 1b.

Fix Planned

No fix planned

778 Processor Core Time Stamp Counters May Experience Drift

Description

Following a long period in a P-state without any core P-state or C-state activity, the time stamp counter for a processor core may appear to drift slightly from other processor cores. This TSC drift does not occur unless the processor has spent over four billion consecutive clocks in a single P-state at C0.

This erratum does not apply if the processor is in the non-boosted software P0 frequency.

Potential Effect on System

System software or software with multiple threads may observe that one thread or processor core provides TSC values that are behind another thread or processor core. This can only happen if the processor core is spending very long intervals in the C0 (running) state and is either pinned to a software P-state lower than P0, or the application power management (APM) behavior of the software running on this core allows the processor to remain in a boosted state without any changes to the P-state.

A single thread operating on a single core can not observe successively stored TSC values that incorrectly decrement.

Suggested Workaround

Contact your AMD representative for information on a BIOS update.

Fix Planned

No fix planned

779 Initial Time Stamp Counter Frequency May Be Incorrect

Description

The processor core may increment the core time stamp counter (TSC) at a frequency that is equal to the startup P-state frequency, instead of incrementing the TSC at the software P0 frequency. This effect occurs until the first instance of either a P-state or a C-state change.

At the time that the first P-state or C-state change occurs, the actual value in the TSC will adjust as if it had incremented at this P-state rate for the entire duration.

Potential Effect on System

The BIOS may calculate incorrect time stamps. For example, since the TSC is incrementing at a slower rate than it should increment, calculations of the elapsed time in BIOS boot may appear to be incorrect.

In the unlikely event that the system is in a state where one or more cores are affected by the erratum (i.e., those cores have not performed any P-state or C-state changes) and other cores have performed a change, the operating system software may observe TSC synchronization failures during early boot due to the different frequencies. A normal system initialization causes at least one P-state change on all cores before the operating system boots.

Suggested Workaround

BIOS must perform a P-state change on all cores prior to the transfer of control to the operating system.

Fix Planned

No fix planned

780 Processor May Cache Guest Write Combining Memory Type

Description

The processor may incorrectly cache reads performed to memory regions that are defined as WC+ memory type. The processor may incorrectly cache this memory type based on speculative read operations. The program does not need to retire a load instruction in order for the caching to occur.

This incorrectly cached data is removed from the cache if there is any write to this address from this processor core, from another processor core, or from a device that probes all cores during the write.

The WC+ memory type is only used when all of the following conditions apply:

- An SVM guest with nested paging enabled is currently executing.
- The guest page table maps the memory to WC as the guest PAT memory type.
- The host page table maps the memory to WP, WT or WB as the host PAT memory type.
- The MTRR memory type is either WP, WT, or WB.

Potential Effect on System

Under most conditions, except as specified below, the incorrect caching has no effect as this WC+ memory type is still probed by processor and I/O accesses. As a result, these transactions still observe and maintain the most current copy of the data even in the presence of incorrect caching.

Incorrect caching may have an effect when one of these conditions occurs:

- An SVM guest program observes inconsistent "stale" data for a write-combining MMIO address due to the program observing cached data that is inconsistent with the current device state. In order for this to occur, the SVM guest must have direct mapped access to the MMIO address region for an I/O device that is capable of write-combining. However, the MTRR for the device's MMIO region would also normally be mapped as WC, and the erratum would not apply in this case.
- An SVM guest program observes inconsistent "stale" data when it has DRAM pages marked as WC in the guest PAT tables and is using this memory as a buffer that a non-coherent device may also write (a non-coherent device is one that does not probe processor caches when it reads or writes the system memory). One possible example of a device that does not probe processor caches during the upstream writes to memory is a graphics engine (GPU) writing into a DRAM mapped buffer or a PCI Express® device that is using the No Snoop attribute in its upstream transactions.

Suggested Workaround

A workaround is not recommended.

System developers may disable the ability for an I/O device to perform upstream writes without probing memory by setting D18F3x88[22] = 1b.

In order to also require probing on writes performed by the integrated graphics device, system developers may also set GMMx206C[1:0] = 01b.

Fix Planned

No fix planned

781 Improper Handling of ECX Value During RDPMC Instruction

Description

The processor may return the incorrect northbridge performance counter register, when a RDPMC instruction is executed with ECX equal to 6, 7, 8 or 9.

Potential Effect on System

Performance monitoring software receives incorrect northbridge performance monitor results.

Suggested Workaround

Contact your AMD representative for information on a BIOS update.

Fix Planned

No fix planned

785 USB Interrupt Status May Not Be Set After a Short Packet

Description

The Enhanced Host Controller Interface (EHCI) does not issue an interrupt after processing a short packet when the Interrupt on Completion (IOC) bit in the Transfer Descriptor (TD) is 0b. The specification violation is encountered only when the following conditions apply:

- A USB full-speed or USB low-speed device is attached to the EHCI controller port (USB_HSD[9:0]P/N) through a USB high-speed external hub.
- The USB device is performing bulk or interrupt transfers to the host controller.
- The USB host controller driver (HCD) is not setting the IOC bit on all transfer descriptors.

Potential Effect on System

USB 2.0 full-speed and low-speed devices connected to EHCI controller port (USB_HSD[9:0]P/N pins) via a high-speed external hub may encounter functional problems.

Suggested Workaround

The USB host controller driver should set the Interrupt On Complete bit on all Transfer Descriptors.

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

Documentation Support

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

- *BIOS and Kernel Developer's Guide (BKDG) for AMD Family 16h Models 00h-0Fh Processors*, order# 48751
- *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.