Revision Guide for AMD Family 11h Processors

Publication # **41788**	Revision: **3.10**
Issue Date: **December 2011**	

Advanced Micro Devices

Revision History

Date	Revision	Description
December 2011	3.10	Updated Register References and Mnemonics; Added Arithmetic and Logical Operators; Clarified Programming and Displaying the Processor Name String; Updated F4x164 Fixed Errata Register, MSRC001_0140 OS Visible Work-around MSR0 (OSVW_ID_Length), MSRC001_0141 OS Visible Work-around MSR1 (OSVW_Status) and Table 6; Added errata #415, #573, #670 and #700-#701.
July 2009	3.08	Updated Overview, Tables 2, 4 and 8 for AMD Athlon™ and AMD Sempron™ X2 Dual-Core brands; Clarified Programming and Displaying the Processor Name String; Corrected #122 and #171 applicability in Table 8; Added errata #246, #352, #353, #365, #393 and #401.
July 2008	3.00	Initial public release.

Revision Guide for AMD Family 11h Processors

Overview

The purpose of the *Revision Guide for AMD Family 11h 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 Athlon™ Processor
- AMD Athlon X2 Dual-Core Processor
- AMD Sempron™ Processor
- AMD Sempron X2 Dual-Core Processor
- AMD Turion™ X2 Dual-Core Mobile Processor
- AMD Turion X2 Ultra Dual-Core Mobile Processor

This guide consists of these major sections:

- **Processor Identification:** This section, starting on page 9, shows how to determine the processor revision and workaround requirements, and to construct, program and display the processor name string.
- **Product Errata:** This section, starting on page 16, 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:** This section, starting on page 52, 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 11h Processors*, order# 41256. 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.
- FYxXXX: PCI-defined configuration space; 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, F3x40 specifies the register at function 3, address 40h. Each processor node includes five functions, 0 through 4.
- FYxXXX_xZZZZZ: Port access through the PCI-defined configuration space; 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, F2x9C_x1C specifies the port 1Ch register accessed using the data port register at function 2, address 9Ch. Refer to the *BIOS and Kernel Developer’s Guide (BKDG) for AMD Family 11h Processors*, order# 41256 for access properties.

- 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.

Many register references use the notation "[]" to identify a range of registers. For example, F2x[1,0][4C:40] is a shorthand notation for F2x40, F2x44, F2x48, F2x4C, F2x140, F2x144, F2x148, and F2x14C.

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.

Table 1. Arithmetic and Logic Operators (Continued)

Operator	Definition
<<	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.

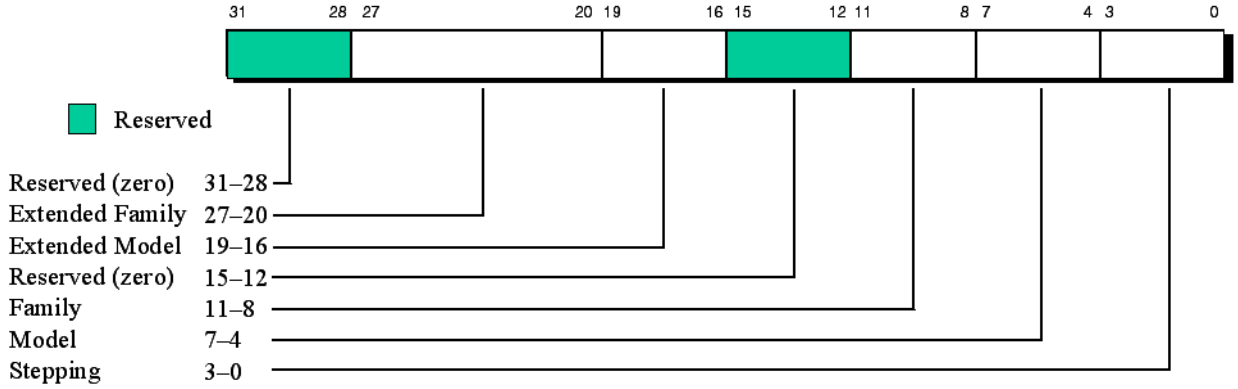

Figure 1. Format of CPUID Fn0000_0001_EAX

Table 2 shows the identification number from CPUID Fn0000_0001_EAX for each revision of the processor.

Table 2. CPUID Values for AMD Family 11h Processor Revisions

Rev.	CPUID Fn0000_0001_EAX					
	AMD Turion™ X2 Ultra Dual-Core Mobile Processor	AMD Turion™ X2 Dual-Core Mobile Processor	AMD Athlon™ Processor	AMD Athlon™ X2 Dual-Core Processor	AMD Sempron™ Processor	AMD Sempron™ X2 Dual-Core Processor
LG-B1	00200F31h	00200F31h	00200F31h	00200F31h	00200F31h	00200F31h

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 11h Processors*, order# 41256, 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 11h Processors*, order# 41256, 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 uses the following fields to create the name string:

- BrandId[15:0] is from CPUID Fn8000_0001_EBX[15:0].
  - **String1[3:0]** is defined to be BrandID[14:11]. This field is an index to a string value used to create the processor name string. The definitions of the String1 values are provided in Table 4.
  - **String2[3:0]** is defined to be BrandID[3:0]. This field is an index to a string value used to create the processor name string. The definitions of the String2 values are provided in Table 5.
  - **PartialModel[6:0]** is defined to be BrandID[10:4]. This field is normally used to create some or all of the model number in the name string. This field represents a number which should be converted to ASCII for display.
  - **Pg[0]** is defined to be BrandID[15]. This field is used to index the appropriate page for the tables.
- PkgType[3:0] is from CPUID Fn8000_0001_EBX[31:28]. This field specifies the package type as defined in the *BIOS and Kernel Developer's Guide (BKDG) for AMD Family 11h Processors*, order# 41256, and is used to index the appropriate string tables from Table 3.

- NC[7:0] is one less than the number of physical cores that are present as defined in the *BIOS and Kernel Developer's Guide (BKDG) for AMD Family 11h Processors*, order# 41256, and is used to index the appropriate strings from Tables 4 through 5. NC[7:0] is from CPUID Fn8000_0008_ECX[7:0].

The name string is formed as follows:

1. Translate PartialModel[6:0] into an ASCII value (*PartialModelAscii*). This number will range from 01-99 and should include a leading zero if less than 10, e.g., 09.
2. Select the appropriate string tables based on PkgType[3:0] from Table 3.
3. Index into the referenced tables using Pg[0], String1[3:0], String2[3:0], and NC[7:0] to obtain the *String1* and *String2* values.
4. If *String1* is an undefined value skip all remaining steps and program the name string as follows:

   Name String = AMD Processor Model Unknown
5. Else concatenate the strings with the two character ASCII translation of PartialModel[3:0] from step 1 to obtain the name string as follows:

   If *String2* is undefined, *Name string = String1, PartialModelAscii*

   Else, *Name string = String1, PartialModelAscii, String2*

Table 3. String Table Reference Per Package Type

PkgType [3:0]	String1 Table	String2 Table
0h-1h	Reserved	Reserved
2h	Table 4	Table 5
3h-Fh	Reserved	Reserved

Table 4. String1 Values for S1g2 Processors

Pg[0]	NC [7:0]	String1 [3:0]	Value	Note	Description
0b	0h	00h	AMD Sempron(tm) SI-	-	
		01h	AMD Athlon(tm) QI-	-	
	1h	00h	AMD Turion(tm) X2 Ultra Dual-Core Mobile ZM-	-	
		01h	AMD Turion(tm) X2 Dual-Core Mobile RM-	-	
		02h	AMD Athlon(tm) X2 Dual-Core QL-	-	
		03h	AMD Sempron(tm) X2 Dual-Core NI-	-	
All other values			AMD Processor Model Unknown	-	

Table 5. String2 Value for S1g2 Processors

Pg[0]	NC [7:0]	String2 [3:0]	Value	Note	Description
0b	0h	00h		1	
	1h	00h		1	
	xxh	0Fh		1	
All other values			Reserved	-	

Notes:

1. *The String2 indexes 0h and 0Fh are defined as an empty string, i.e., no suffix.*

F4x164 Fixed Errata Register

Communicating the status of an erratum within a stepping of a processor family is necessary in certain circumstances. F4x164 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.

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 0004h 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
63:4	**OsvwStatusBits:** Reserved. OS visible work-around status bits. Read-write.
3	**Osvwld3:** Reserved, must be zero.
2	**Osvwld2:** 1= Hardware contains erratum #415, an OS workaround may be applied if available. 0= Hardware has corrected erratum #415.
1	**Osvwld1:** Reserved, must be zero.
0	**Osvwld0:** Reserved, must be zero.

BIOS shall program the state of the valid status bits as shown in Table 6 prior to hand-off to the OS.

Table 6. Cross Reference of Product Revision to OSVW ID

CPUID Fn0000_0001_EAX (Mnemonic)	MSRC001_0141 Value
00200F31h (LG-B1)	0000_0000_0000_0004h

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. Table 7 cross-references the revisions of the part to each erratum. An "X" indicates that the erratum applies to the revision. The absence of an "X" indicates that the erratum does not apply to the revision. An "*" indicates advance information that the erratum has been fixed but not yet verified. "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 7. Cross-Reference of Product Revision to Errata

No.	Errata Description	Revision Number
		LG-B1
1	Inconsistent Global Page Mappings Can Lead to Machine Check Error	No fix planned
57	Some Data Cache Tag Eviction Errors Are Reported As Snoop Errors	No fix planned
60	Single Machine Check Error May Report Overflow	No fix planned
77	Long Mode CALLF or JMPF May Fail To Signal GP When Callgate Descriptor is Beyond GDT/LDT Limit	No fix planned
122	TLB Flush Filter May Cause Coherency Problem in Multicore Systems	No fix planned
144	CLFLUSH to Shadow RAM Address Will Not Invalidate	No fix planned
171	Instruction Break Point On VMRUN Instruction Leads To Unpredictable System Behavior	No fix planned
230	Misaligned I/O Reads That Span CFCh Incorrectly Generate a Downstream I/O Request	No fix planned
246	Breakpoint Due to An Instruction That Has an Interrupt Shadow May Be Delivered to the Hypervisor	No fix planned
250	I/O Reads That Span 3BBh May Be Positively Decoded When They Should Not Be Positively Decoded	No fix planned
251	Northbridge Flow Control Credits May Be Lost Due to Watchdog Time Out	No fix planned
272	Logged Sync Error Results in Error Overflow Being Set	No fix planned
288	Write To A F4x184 Register May Access Incorrect Array Register	No fix planned
297	Single Machine Check Error May Report Overflow	No fix planned
305	Northbridge Will Not Raise an MCA Exception Unless Status Bits Are Cleared	No fix planned
307	Reset Occurring Near a Dynamic Link Frequency Change May Cause System Failures	No fix planned
311	Certain Clock Divisors May Result in Unpredictable System Behavior	No fix planned
312	CVTSD2SS and CVTPD2PS Instructions May Not Round to Zero	No fix planned
316	Incorrect CpuDid May Be Applied During ACPI States	No fix planned
332	Alternative Voltage Is Not Supported	No fix planned
339	APIC Timer Rollover May Be Delayed	No fix planned
342	SMIs That Are Not Intercepted May Disable Interrupts	No fix planned
352	SYSCALL Instruction May Execute Incorrectly Due to Breakpoint	No fix planned

Table 7. Cross-Reference of Product Revision to Errata (Continued)

No.	Errata Description	Revision Number
		LG-B1
353	SYSRET Instruction May Execute Incorrectly Due to Breakpoint	No fix planned
365	C1E Mode May Cause Unpredictable System Behavior	No fix planned
393	Performance Monitor May Count Fastpath Double Operation Instructions Incorrectly	No fix planned
401	Software Interrupt Event Injection Ignores VMCB.NextRIP	No fix planned
415	HLT Instructions That Are Not Intercepted May Cause System Hang	No fix planned
573	Processor May Incorrectly Update Instruction Pointer After FSINCOS Instruction	No fix planned
670	Segment Load May Cause System Hang or Fault After State Change	No fix planned
700	LAR and LSL Instructions Do Not Check Invalid Long Mode Descriptor Types	No fix planned
701	Task Switch Intercept Does Not Store VMEXIT EXITINTINFO	No fix planned

Table 8 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 Number	AMD Turion™ X2 Ultra Dual-Core Mobile Processor	AMD Turion™ X2 Dual-Core Mobile Processor	AMD Athlon™ Processor	AMD Athlon™ X2 Dual-Core Processor	AMD Sempron™ Processor	AMD Sempron™ X2 Dual-Core Processor
1	X	X	X	X	X	X
57	X	X	X	X	X	X
60	X	X	X	X	X	X
77	X	X	X	X	X	X
122	X	X		X		X
144	X	X	X	X	X	X
171	X	X	X	X		
230	X	X	X	X	X	X
246	X	X	X	X		
250	X	X	X	X	X	X
251	X	X	X	X	X	X
272	X	X	X	X	X	X
288	X	X	X	X	X	X
297	X	X	X	X	X	X
305	X	X	X	X	X	X
307	X	X	X	X	X	X
311	X	X	X	X	X	X
312	X	X	X	X	X	X
316	X	X	X	X	X	X
332	X	X	X	X	X	X
339	X	X	X	X	X	X
342	X	X	X	X		
352	X	X	X	X	X	X
353	X	X	X	X	X	X
365	X	X	X	X	X	X
393	X	X	X	X	X	X
401	X	X	X	X		
415	X	X	X	X		
573	X	X	X	X	X	X

Table 8. Cross-Reference of Errata to Processor Segments (Continued)

Errata Number	AMD Turion™ X2 Ultra Dual-Core Mobile Processor	AMD Turion™ X2 Dual-Core Mobile Processor	AMD Athlon™ Processor	AMD Athlon™ X2 Dual-Core Processor	AMD Sempron™ Processor	AMD Sempron™ X2 Dual-Core Processor
670	X	X	X	X	X	X
700	X	X	X	X	X	X
701	X	X	X	X		

1 Inconsistent Global Page Mappings Can Lead to Machine Check Error

Description

If the same linear to physical mapping exists in multiple CR3 contexts, and that mapping is marked global in one context and not global in another context, then a machine check error may be reported by the TLB error detection logic (depending on the specific access pattern and TLB replacements encountered).

Potential Effect on System

In the somewhat unlikely event that all required conditions are present (including the effects of the TLB replacement policy), then an unexpected machine check error may be reported. If the erratum occurs in the instruction cache TLB (L1 or L2), the apparent error is logged and corrected. If the erratum occurs in the data cache TLB (L1 or L2), the apparent error is logged and reported as an uncorrectable machine check error.

Suggested Workaround

None required. This is not expected to occur in real systems.

Fix Planned

No

57 Some Data Cache Tag Eviction Errors Are Reported As Snoop Errors

Description

In some cases, the machine check error code on a data cache (DC) tag array parity error erroneously classifies an eviction error as a snoop error.

The common cases of cache line replacements and external probes are classified correctly (as eviction and snoop respectively). The erroneous cases occur when a tag error is detected during a DC eviction that was generated by a hardware prefetch, a cache line state change operation, or a number of other internal microarchitectural events. In such cases, the error code logged in the DC Machine Check Status register (MC0_STATUS, MSR0000_0401) erroneously indicates a snoop error.

Potential Effect on System

Internally detected DC tag errors may be reported to software as having been detected by snoops. Depending upon machine check software architecture, the system response to such errors may be broader than necessary.

Suggested Workaround

None required.

Fix Planned

No

60 Single Machine Check Error May Report Overflow

Description

A single parity error encountered in the data cache tag array may incorrectly report the detection of multiple errors, as indicated by the overflow bit of the DC Machine Check Status register (bit 62 of MSR0000_0401).

Potential Effect on System

System software may be informed of a machine check overflow when only a single error was actually encountered.

Suggested Workaround

Do not rely on the state of the OVER bit in the DC Machine Check Status register.

Fix Planned

No

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

122 TLB Flush Filter May Cause Coherency Problem in Multicore Systems

Description

Under a highly specific set of internal timing conditions in system configurations that include more than one processor core, coherency problems may arise between the page tables in memory and the translations stored in the on-chip TLBs. This can result in the possible use of stale translations even after software has performed a TLB flush.

Potential Effect on System

Unpredictable system failure. This scenario has only been observed in a highly randomized synthetic stress test.

Suggested Workaround

In multicore systems, disable the TLB flush filter by setting HWCR.FFDIS (bit 6 of MSRC001_0015).

Fix Planned

No

144 CLFLUSH to Shadow RAM Address Will Not Invalidate

Description

WrDram and RdDram bits in extended MTRR type registers are used to copy BIOS ROM to corresponding DRAM and then execute out of DRAM. When these are configured to direct writes to ROM (WrMem =0b) and reads to DRAM (RdMem =01b), the CLFLUSH instruction will not invalidate shadow RAM addresses in the cache.

Potential Effect on System

CLFLUSH instruction is ineffective for shadow RAM space.

Suggested Workaround

Use the WBINVD instruction instead of CLFLUSH in shadow RAM space.

Fix Planned

No

171 Instruction Break Point On VMRUN Instruction Leads To Unpredictable System Behavior

Description

VMRUN can be interrupted using a hardware instruction breakpoint using one of the debug registers, DR[0-3]. When the debug handler executes IRET, the processor is expected to execute the VMRUN instruction. However, in the failing case, the processor incorrectly re-enters the breakpoint handler with mixed guest and host state. This in turn causes erroneous execution and leads to unpredictable system behavior.

Potential Effect on System

Hypervisor developers will not be able to use hardware instruction break point on VMRUN instruction.

Suggested Workaround

Set the breakpoint on the instruction prior to VMRUN, then single step through VMRUN.

Fix Planned

No

230 Misaligned I/O Reads That Span CFCh Incorrectly Generate a Downstream I/O Request

Description

When configuration space is enabled, IOCF8[31] is 1b, an I/O read to address CFCh should result in a configuration request to the address specified in register IOCF8. However, when a misaligned downstream double word I/O read spans address CFCh the northbridge (NB) correctly sends an I/O read requests to CF8h with appropriate byte enables to the device attached to the I/O link, but incorrectly sends an I/O read request to CFCh instead of the configuration request.

Potential Effect on System

None expected.

Suggested Workaround

Software should not issue misaligned read requests to I/O addresses that span address CFCh.

Fix Planned

No

246 Breakpoint Due to An Instruction That Has an Interrupt Shadow May Be Delivered to the Hypervisor

Description

A #DB exception occurring in guest mode may be delivered in the host context 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

Unpredictable results due to an unexpected #DB exception.

Suggested Workaround

The hypervisor should have a valid interrupt gate in the IDT of the #DB handler entry and the handler must be able to determine that this event has occurred. If the event is detected, the handler should execute an IRET back to the hypervisor; one method that could be used to evaluate for this condition is to compare the RIP pushed on the stack to the RIP of the instruction following VMRUN, if they are equivalent then this event has occurred.

Fix Planned

No

250 I/O Reads That Span 3BBh May Be Positively Decoded When They Should Not Be Positively Decoded

Description

The northbridge enables positive decode within the first 64 KB of I/O space mapped by the I/O base/limit registers (F1xC0 and F1xC4) for the legacy VGA registers when F1xC0[4] (VE) is 1b and F1xF4[0] (VE) is 0b, i.e. accesses in which address bits[9:0] range from 3B0h to 3BBh or 3C0h to 3DFh and address bits[24:16] are all 0. However, if an I/O read spans address 3BBh, the northbridge will positively decode the entire access including the addresses outside the legacy VGA register space (i.e. 3B[C:E]h).

Potential Effect on System

A downstream request to I/O addresses 3B[C:E]h may not properly set the Compat bit. This may result in the packet not being forwarded to the compatibility bus.

Suggested Workaround

None required.

Fix Planned

No

251 Northbridge Flow Control Credits May Be Lost Due to Watchdog Time Out

Description

A peer to peer or CPU transaction that times out in the northbridge due to a watchdog time out may result in lost flow control credits. In addition, a master abort is not sent for peer to peer transactions that time out in the northbridge due to a watchdog time out.

Potential Effect on System

The system may hang.

Suggested Workaround

None. The watchdog time out is correctly reported as a MCA exception.

Fix Planned

No

272 Logged Sync Error Results in Error Overflow Being Set

Description

When a sync error is logged in MC4_STATUS (MSR0000_0411), an error overflow is always indicated by MC4_STATUS[Over] (MSR0000_0411[62]) being set to 1b.

Potential Effect on System

None.

Suggested Workaround

None required.

Fix Planned

No

288 Write To A F4x184 Register May Access Incorrect Array Register

Description

The contents of the Link Phy Offset Register (F4x180) may be used at the time of the write into the Link Phy Data Port register (F4x184) to determine which set of registers is being accessed. Since the software does not write F4x180 until after F4x184 when performing an array write operation, this may result in the array write not occurring or going to an unpredictable array register.

Potential Effect on System

Unpredictable results may occur.

Suggested Workaround

The registers accessed by F4x184_x[N:0] can be split into four spaces

1. Direct map registers (When DirectMapEn is 1b)
2. Link FIFO Read Pointer Optimization Registers (offsets CFh and DFh)
3. BIST registers (offsets 100h to 144h)
4. Phy registers (offset E0h)

Before performing a write to a F4x184_x[N:0] array register, when the value (i.e. from a previous access) in LinkPhyOffset or DirectMapEn maps to a different space (as defined above), software should first perform a read operation to the intended array register.

No workaround is necessary when performing an array read access or when no space switch is involved.

Fix Planned

No

297 Single Machine Check Error May Report Overflow

Description

A single tag snoop parity error encountered in the instruction cache tag array may incorrectly report the detection of multiple errors, as indicated by the overflow bit of the IC Machine Check Status register (MSR0000_0405[62]).

Potential Effect on System

System software may be informed of a machine check overflow when only a single error was actually encountered.

Suggested Workaround

None required.

Fix Planned

No

305 Northbridge Will Not Raise an MCA Exception Unless Status Bits Are Cleared

Description

A machine check (MCA) exception is not raised if the northbridge detects an overflow condition. An overflow condition exists if there is already an enabled uncorrectable machine check exception in the northbridge MCA status register at the time that a second uncorrectable error is detected.

Potential Effect on System

The system will not enter the shutdown state if another uncorrectable error occurs prior to the MCA handler clearing the northbridge MCA status register.

Suggested Workaround

None.

Fix Planned

No

307 Reset Occurring Near a Dynamic Link Frequency Change May Cause System Failures

Description

System failure can occur if reset is asserted while an LDTSTOP disconnect is in process for a Centralized Dynamic Link Frequency (CDLF) change and the HyperTransport™ link is operating at a speed of 1000 MHz or lower (Gen1).

Potential Effect on System

The system may fail to reset.

Suggested Workaround

CDLF should not be used in conjunction with Gen1 frequency speeds.

Fix Planned

No

311 Certain Clock Divisors May Result in Unpredictable System Behavior

Description

The processor may violate internal timing requirements at certain core clock divisors. This may occur if software modifies the CpuFid in the P-state registers (MSRC001_00[6B:64][5:0]) or when core clocks are ramped due to P-state transitions in response to writes to the P-state Control register (MSRC001_0062), HTC, or HLT instruction execution.

Potential Effect on System

Unpredictable system behavior.

Suggested Workaround

Contact your AMD representative for information on a BIOS update. In addition, software should not modify the P-state registers (MSRC001_00[6B:64]), or the HtcPstateLimit (F3x64[30:28]), from their reset values.

Fix Planned

No

312 CVTSD2SS and CVTPD2PS Instructions May Not Round to Zero

Description

The Convert Scalar Double-Precision Floating-Point to Scalar Single-Precision Floating-Point (CVTSD2SS) and Convert Packed Double-Precision Floating-Point to Packed Single-Precision Floating-Point (CVTPD2PS) instructions do not round to zero when the Flush to Zero and Underflow Mask bits (MXCSR bits 15 and 11) are set to 1b and the double-precision operand is less than the smallest single-precision normal number.

Potential Effect on System

The conversion result will yield the smallest single-precision normalized number rather than zero. It is not expected that this will result in any anomalous software behavior since enabling flush to zero provides less precise results.

Suggested Workaround

None.

Fix Planned

No

316 Incorrect CpuDid May Be Applied During ACPI States

Description

The processor may use an incorrect core clock divisor if it enters or exits the HTC-active state in a narrow window of time after it has applied the CpuDid for an ACPI power state transition. Under these conditions, the CpuDid from the current P-state as indicated by CurPstate (MSRC001_0063[2:0]) is used rather than the CpuDid from the ACPI power state.

Potential Effect on System

Unpredictable system behavior may result if an alternate voltage (altvid) is enabled for the ACPI power state, for example in C1E, and the CpuDid applied is less than that specified for the ACPI power state.

Suggested Workaround

Contact your AMD representative for information on a BIOS update.

Fix Planned

No

332 Alternative Voltage Is Not Supported

Description

The processor may function improperly when the alternative voltage (altvid) is applied.

Potential Effect on System

Unpredictable system behavior.

Suggested Workaround

System software should not enable the alternative voltage using the Power Management Control Low Register[PMM1[CpuAltVidEn]] (F3x80[12]) or Power Management Control Low Register[PMM3[CpuAltVidEn]] (F3x80[28]). In addition, system software should not enable the alternative voltage using the LMM Configuration Register[LmmCpuAltVidEn] (F4x174_x[0F:00][17]) when using centralized link management.

Fix Planned

No

339 APIC Timer Rollover May Be Delayed

Description

The APIC timer does not immediately rollover when it transitions to zero and Timer Local Vector Table Entry[Mode] (APIC320[17]) is configured to run in periodic mode. In addition, when Timer Local Vector Table Entry[Mask] (APIC320[16]) is configured to generate an interrupt, the interrupt is also delayed whether configured for periodic or one-shot mode.

The per rollover error that may be observed is between 85 and 210 ns.

Potential Effect on System

None expected. The standard use of the APIC timer and the level of accuracy required does not make the error significant.

Suggested Workaround

None required.

Fix Planned

No

342 SMIs That Are Not Intercepted May Disable Interrupts

Description

During a resume from SMM that is due to an unintercepted SMI from a SVM guest context, the processor core does not restore the correct effective interrupt flag (IF) if the guest VMCB V_INTR_MASKING bit (offset 060h bit 24) is 1b. Under these conditions, the effective interrupt flag may be zero.

SMIs are not intercepted if VMCB offset 00Ch bit 2 is 0b or HWCR[SmmLock] (MSRC001_0015[0]) is 1b.

Potential Effect on System

The guest context may run with interrupts disabled until the next guest intercept. The hypervisor may not be able to regain control and the system may hang.

Suggested Workaround

None. BIOS should not set HWCR[SmmLock] (MSRC001_0015[0]) and hypervisors should intercept SMIs.

Fix Planned

No

352 SYSCALL Instruction May Execute Incorrectly Due to Breakpoint

Description

A SYSCALL instruction executes incorrectly and an incorrect debug exception is taken when all of the following conditions are satisfied:

- An enabled instruction breakpoint address matches the RIP of the SYSCALL instruction.
- The processor is in 64-bit mode or compatibility mode.
- The instruction would not generate any other exception.
- SYSCALL Flag Mask Register[16] (MSRC000_0084[16]) is set to 1b.
- rFLAGS.RF is set to 1b.

Potential Effect on System

None expected during normal operation. Kernel debuggers may observe unpredictable system behavior.

Suggested Workaround

Operating system software should clear SYSCALL Flag Mask Register[16] (MSRC000_0084[16]) to 0b during initialization.

Fix Planned

No

353 SYSRET Instruction May Execute Incorrectly Due to Breakpoint

Description

A SYSRET instruction executes incorrectly and an incorrect debug exception is taken when all of the following conditions are satisfied:

- An enabled instruction breakpoint address matches the RIP of the SYSRET instruction.
- The processor is in 64-bit mode or compatibility mode.
- The instruction would not generate any other exception.
- R11[16] is cleared to 0b.
- rFLAGS.RF is set to 1b.

Potential Effect on System

None expected during normal operation. Kernel debuggers may observe unpredictable system behavior.

Suggested Workaround

Software should set R11[16] to 1b before executing the SYSRET instruction in 64-bit mode.

Fix Planned

No

365 C1E Mode May Cause Unpredictable System Behavior

Description

Unpredictable system behavior may occur during C1E entry.

Potential Effect on System

Unpredictable system behavior.

Suggested Workaround

BIOS should use Centralized Link Management Controller (CLMC) features instead of C1E and should not set Interrupt Pending and CMP-Halt Register[C1eOnCmpHalt, SmiOnCmpHalt] (MSRC001_0055[28:27]).

Fix Planned

No

393 Performance Monitor May Count Fastpath Double Operation Instructions Incorrectly

Description

The processor does not report the correct count for all fastpath double operation instructions when Performance Event Select Register (PERF_CTL[3:0]) MSRC001_000[3:0][EventSelect] is 0CCh. This erratum applies to all unit mask settings for this event.

Potential Effect on System

Performance monitoring software will not have an accurate count of fastpath double operation instructions.

Suggested Workaround

None.

Fix Planned

No

401 Software Interrupt Event Injection Ignores VMCB.NextRIP

Description

The processor does not use VMCB.NextRIP when event injection is used on an INTn, INT3, or INTO software interrupt. In the event that the injected instruction encounters a fault, the processor may incorrectly store the address following the software interrupt on the stack.

Potential Effect on System

Hypervisor software that uses event injection for software interrupts may cause an incorrect RIP to be saved to the stack. The software interrupt will not be injected after the fault is resolved.

Suggested Workaround

Hypervisor software should perform emulation of software interrupts, as if VMCB.NextRIP is not supported (CPUID Fn8000_000A_EDX[3]=0b).

Fix Planned

No

415 HLT Instructions That Are Not Intercepted May Cause System Hang

Description

In guest mode when VMCB.V_INTR_MASK flag is 1b, the processor may not process host interrupts if a guest executes a HLT instruction that is not intercepted.

Potential Effect on System

System hang.

Suggested Workaround

Hypervisors should intercept HLT instructions by setting VMCB.Intercept_HLT (offset 00Ch bit 24) to 1b.

Fix Planned

No

573 Processor May Incorrectly Update Instruction Pointer After FSINCOS Instruction

Description

After execution of an FSINCOS instruction, the processor core may incorrectly update the instruction pointer (rIP) and execute incorrect instructions or may hang.

Potential Effect on System

Unpredictable system behavior after execution of an FSINCOS instruction.

Suggested Workaround

None.

Fix Planned

No

670 Segment Load May Cause System Hang or Fault After State Change

Description

Under a highly specific and detailed set of conditions, a segment load instruction may cause a failure in one of the following instructions later in the instruction stream:

- BTC mem, imm8
- BTC mem, reg
- BTR mem, imm8
- BTR mem, reg
- BTS mem, imm8
- BTS mem, reg
- RCL mem, cl
- RCL mem, imm
- RCR mem, cl
- RCR mem, imm
- SHLD mem, reg, imm
- SHLD mem, reg, cl
- SHRD mem, reg, imm
- SHRD mem, reg, cl
- XCHG mem, reg (uses an implicit LOCK prefix)
- XCHG reg, mem (uses an implicit LOCK prefix)
- Any instruction with an explicit LOCK prefix in the instruction opcode.

Potential Effect on System

The processor may present a #PF exception after some of the instruction effects have been applied to the processor state. No system effect is observed unless the operating system's page fault handler has some dependency on this interim processor state, which is not the case in any known operating system software. The interim state does not impact program behavior if the operating system resolves the #PF and resumes the instruction. However, this interim state may be observed by a debugger or if the operating system changes the #PF to a program error (for example, a segmentation fault).

Suggested Workaround†

System software should set MSRC001_1020[8] = 1b.

†This workaround ensures that instructions with an implicit or explicit LOCK prefix do not cause a system hang due to this erratum. However, instructions may still present a #PF after altering architectural state.

Fix Planned

No

700 LAR and LSL Instructions Do Not Check Invalid Long Mode Descriptor Types

Description

The architecture specifies that the processor checks for invalid descriptor types when a Load Access Rights Byte (LAR) instruction or a Load Segment Limit (LSL) instruction is executed in long mode. An invalid descriptor type should cause the processor to clear the zero flag (ZF) and complete the instruction without modifying the destination register. However, the processor does not perform this check and loads the attribute (LAR) or segment limit (LSL) as if the descriptor type was valid.

The invalid descriptor types for LAR are 1 (available 16-bit TSS), 3 (busy 16-bit TSS), 4 (16-bit call gate) or 5 (task gate). The invalid descriptor types for a LSL instruction are types 1 (available 16-bit TSS) or 3 (busy 16-bit TSS).

Potential Effect on System

None expected, since the operating system code would typically only provide legal descriptors. However, in the case of erroneous software, the above described check would not be performed, resulting in unpredictable system failure. AMD has not observed this erratum with any commercially-available software.

Suggested Workaround

None required, it is anticipated that long mode operating system code ensures that the descriptor type is legal when executing LAR and LSL instructions.

Fix Planned

No

701 Task Switch Intercept Does Not Store VMEXIT EXITINTINFO

Description

During an SVM intercept for a task switch intercept, the processor does not store EXITINTINFO (VMCB offset 0x088) if the guest was attempting to deliver an interrupt or exception through the IDT.

Potential Effect on System

The hypervisor may incorrectly determine the cause of a guest's task switch, resulting in incorrect SVM virtual machine behavior.

Suggested Workaround

None.

Fix Planned

No

Documentation Support

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

- *BIOS and Kernel Developer's Guide (BKDG) for AMD Family 11h Processors*, order# 41256
- *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
- *AMD CPUID Specification*, order# 25481
- *HyperTransport™ I/O Link Specification* (www.hypertransport.org)

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