Revision Guide for AMD Family 12h Processors

Publication # **44739**	Revision: **3.10**
Issue Date: **March 2012**	

Advanced Micro Devices

Revision History

Date	Revision	Description
March 2012	3.10	Added errata #721 and #725.
January 2012	3.08	Updated erratum #573 and #662; Added errata #700 and #710-#711.
September 2011	3.04	Added erratum #686.
August 2011	3.02	Added AMD A-Series Accelerated Processing Unit, AMD E2-Series Accelerated Processing Unit, AMD Sempron™ Processor and AMD Athlon™ II Processor to Overview, Tables 3 and 11; Added package FM1 to Tables 3-5, 8-9 and 12; Updated Table 6; Clarified erratum #418; Added errata #662, #665 and #670; Updated Documentation Support.
June 2011	3.00	Initial release.

Revision Guide for AMD Family 12h Processors

Overview

The purpose of the *Revision Guide for AMD Family 12h 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 E2-Series Mobile Accelerated Processing Unit (APU) with AMD Radeon HD Graphics
- AMD A-Series Accelerated Processing Unit (APU) with AMD Radeon HD Graphics
- AMD E2-Series Accelerated Processing Unit (APU) with AMD Radeon HD Graphics
- AMD Sempron™ X2 Dual-Core Processor
- AMD Athlon™ II X2 Dual-Core Processor
- AMD Athlon II X4 Quad-Core 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 18, 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 53, 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 12h Processors*, order# 41131. 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.
- 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 12h Processors*, order# 41131 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.
- FCRxXXXX_XXXX: Fixed configuration registers used for various aspects of processor initialization, XXXX_XXXX is the hexadecimal address. Refer to the *BIOS and Kernel Developer's Guide (BKDG) for AMD Family 12h Processors*, order# 41131 for access properties.
- 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_00[03:00][EventSelect] (PERF_CTL[3:0] bits 7:0). Y, when specified, signifies the unit mask programmed into MSRC001_00[03:00][UnitMask] (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).

Table 1. Arithmetic and Logic Operators (Continued)

Operator	Definition
`!`	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 page 15).

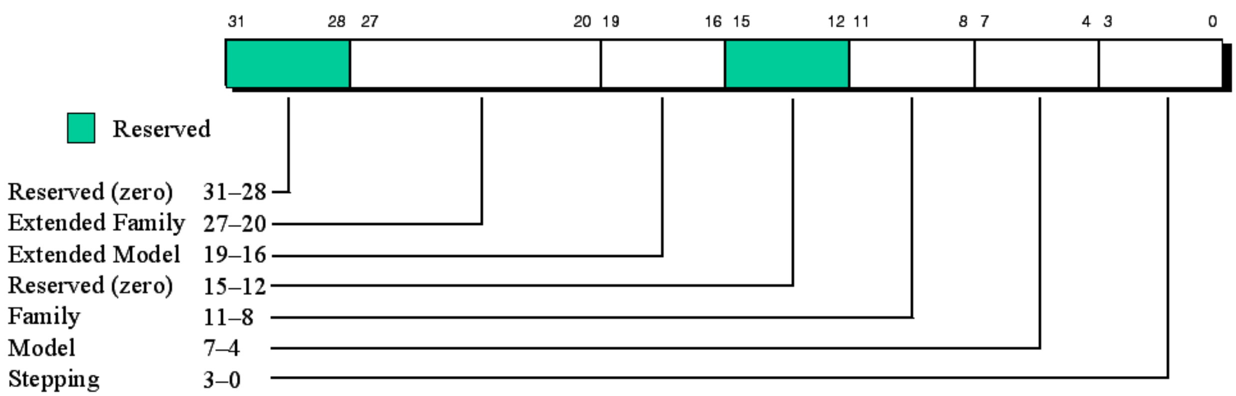

Figure 1. Format of CPUID Fn0000_0001_EAX

Table 2 cross-references the identification number 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 12h FS1 Processor Revisions

CPUID Fn0000_0001_EAX (Mnemonic)	AMD A-Series Mobile APU	AMD E2-Series Mobile APU
00300F10h (LN-B0)	X	X

Table 3. CPUID Values for AMD Family 12h FM1 Processor Revisions

CPUID Fn0000_0001_EAX	AMD A-Series APU	AMD E2-Series APU	AMD Sempron™ Dual-Core Processor	AMD Athlon™ II Dual-Core and Quad-Core Processors
00300F10h (LN-B0)	X	X	X	X

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. Refer to Table 4 for a list of graphics device ID values in use for AMD Family 12h Accelerated Processing Units.

Table 4. AMD Family 12h Graphic Device IDs

D1F0x00[31:16]	FM1	FS1	Notes
9640h	X	N/A	Desktop
9641h	N/A	X	Mobile
9642h	X	N/A	Desktop
9643h	N/A	X	Mobile
9645h	X	N/A	Desktop
9647h	N/A	X	Mobile
9648h	N/A	X	Mobile
964Ah	X	N/A	Desktop

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 12h Processors*, order# 41131, 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 12h Processors*, order# 41131, 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 Tables 6 and 8.
  - **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 Tables 7 and 9.
  - **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. This field may be decremented by one before use.
  - **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 12h Processors*, order# 41131, and is used to index the appropriate string tables from Table 5.
- 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 12h Processors*, order# 41131, and is used to index the appropriate strings from Tables 6 through 7. NC[7:0] is from CPUID Fn8000_0008_ECX[7:0].

The name string is formed as follows:

1. Decrement PartialModel[6:0] by one.
2. Translate PartialModel[6:0] into an ASCII value (*PartialModelAscii*). This number will range from 00-99 and should include a leading zero if less than 10, e.g., 09.
3. Select the appropriate string tables based on PkgType[3:0] from Table 5.
4. 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.
5. If *String1* is an undefined value skip all remaining steps and program the name string as follows:

   Name String = AMD Processor Model Unknown
6. Else concatenate the strings with the two character ASCII translation of PartialModel[3:0] from step 2 to obtain the name string as follows:

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

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

Table 5. String Table Reference Per Package Type

PkgType [3:0]	String1 Table	String2 Table
0h	Reserved	Reserved
1h	Table 6	Table 7
2h	Table 8	Table 9
3h-Fh	Reserved	Reserved

Table 6. String1 Values for FS1 Processors

Pg[0]	NC [7:0]	String1 [3:0]	Value	Note	Description
0b	1h	03h	AMD A4-33		
		05h	AMD E2-30		
	3h	01h	AMD A8-35		
		03h	AMD A6-34		
All other values			AMD Processor Model Unknown	-	

Table 7. String2 Values for FS1 Processors

Pg[0]	NC [7:0]	String2 [3:0]	Value	Note	Description
0b	01h	01h	M APU with Radeon(tm) HD Graphics		
		02h	MX APU with Radeon(tm) HD Graphics		
	03h	01h	M APU with Radeon(tm) HD Graphics		
		02h	MX APU with Radeon(tm) HD Graphics		
	xxh	0Fh		1	
All other values			Reserved	-	

1. *String2 index 0Fh is defined as an empty string, i.e., no suffix.*

Table 8. String1 Values for FM1 Processors

Pg[0]	NC [7:0]	String1 [3:0]	Value	Note	Description
0b	01h	01h	AMD A4-33		
		02h	AMD E2-32		
		04h	AMD Athlon(tm) II X2 2		
		05h	AMD A4-34		
		0Ch	AMD Sempron(tm) X2 1		
	02h	05h	AMD A6-35		
	03h	05h	AMD A8-38		
		06h	AMD A6-36		
		0Dh	AMD Athlon(tm) II X4 6		
All other values			AMD Processor Model Unknown	-	

Table 9. String2 Values for FM1 Processors

Pg[0]	NC [7:0]	String1 [3:0]	Value	Note	Description
0b	01h	01h	APU with Radeon(tm) HD Graphics	1	
		02h	Dual-Core Processor	1	
	02h	01h	APU with Radeon(tm) HD Graphics	1	
	03h	01h	APU with Radeon(tm) HD Graphics	1	
		03h	Quad-Core Processor	1	
	xxh	0Fh		2	
All other values			Reserved	-	

Notes:

1. *The string includes a space as the leading character.*
2. *The String2 index 0Fh is defined as an empty string, i.e., no suffix.*

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.

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	**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 10 prior to hand-off to the OS.

Table 10. Cross Reference of Product Revision to OSVW ID

CPUID Fn0000_0001_EAX (Mnemonic)	MSRC001_0141 Bits
00300F10h (LN-B0)	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. Table 11 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 11. Cross-Reference of Processor Revision to Errata

No.	Errata Description	CPUID Fn0000_0001_EAX (Mnemonic)
		00300F10h (LN-B0)
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
230	Misaligned I/O Reads That Span CFCh Incorrectly Generate a Downstream I/O Request	No fix planned
250	I/O Reads That Span 3BBh May Be Positively Decoded When They Should Not Be Positively Decoded	No fix planned
297	Single Machine Check Error May Report Overflow	No fix planned
343	Eviction May Occur When Using L2 Cache as General Storage During Boot	No fix planned
361	Breakpoint Due to an Instruction That Has an Interrupt Shadow May Be Lost	No fix planned
366	Improper DIMM On-Die Termination Signaling May Occur	No fix planned
418	Host Mapping of Physical Page Zero May Cause Incorrect Translation	No fix planned
430	Processor May Not Recognize A20M# Change During CC6	No fix planned
432	DEV Error May Be Erroneously Logged After a Warm Reset	No fix planned
441	Move from Stack Pointer to Debug or Control Register May Result in Incorrect Value	No fix planned
465	First MRS Command After DRAM Initialization May Time-out	No fix planned
470	Warm Reset May Cause System Hang	No fix planned
474	Memory Clear Feature May Use Non-Zero Pattern	No fix planned
541	IBS Registers May Be Unpredictable After CC6 State	No fix planned
564	Processor May Fail to Set Auto-Halt Restart in SMM Save State	No fix planned
565	Processor Cores Observe Separate IBS Control Registers	No fix planned
573	Processor May Incorrectly Update Instruction Pointer After FSINCOS Instruction	No fix planned
596	Northbridge Clock Gating May Lead to Invalid Prefetched Data	No fix planned
662	Processor May Hang While Performing Voltage Transitions Overlapping with C-state Requests	No fix planned

Table 11. Cross-Reference of Processor Revision to Errata (Continued)

No.	Errata Description	CPUID Fn0000_0001_EAX (Mnemonic)
		00300F10h (LN-B0)
665	Integer Divide Instruction May Cause Unpredictable Behavior	No fix planned
670	Segment Load May Cause System Hang or Fault After State Change	No fix planned
686	Processor Does Not Implement MSRC001_0055	No fix planned
700	LAR and LSL Instructions Do Not Check Invalid Long Mode Descriptor Types	No fix planned
710	AD631XOJZ43GX Incorrectly Reports a Maximum Rate of DDR3-1600	No fix planned
711	Processor May Hang During PCIe® Initialization	No fix planned
721	Processor May Incorrectly Update Stack Pointer	No fix planned
725	Incorrect APIC Remote Read Behavior	No fix planned

Table 12 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 12. Cross-Reference of Errata to Package Type

Errata Number	FM1	FS1
57	X	X
60	X	X
77	X	X
230	X	X
250	X	X
297	X	X
343	X	X
361	X	X
366	X	X
418	X	X
430	X	X
432	X	X
441	X	X
465	X	X
470	X	X
474	X	X
541	X	X
564	X	X
565	X	X
573	X	X
596	X	X
662	X	X
665	X	X
670	X	X
686	X	X
700	X	X
710	X	
711	X	X
721	X	X
725	X	X

Table 13 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 13. Cross-Reference of Errata to Processor Segments

Errata Number	AMD A-Series Mobile APU	AMD E2-Series Mobile APU	AMD A-Series APU	AMD E2-Series APU	AMD Sempron Dual-Core Processor	AMD Athlon II Dual-Core and Quad-Core Processors
57	X	X	X	X	X	X
60	X	X	X	X	X	X
77	X	X	X	X	X	X
230	X	X	X	X	X	X
250	X	X	X	X	X	X
297	X	X	X	X	X	X
343	X	X	X	X	X	X
361	X	X	X	X	X	X
366	X	X	X	X	X	X
418	X	X	X	X	X	X
430	X	X	X	X	X	X
432	X	X	X	X	X	X
441	X	X	X	X	X	X
465	X	X	X	X	X	X
470	X	X	X	X	X	X
474	X	X	X	X	X	X
541	X	X	X	X	X	X
564	X	X	X	X	X	X
565	X	X	X	X	X	X
573	X	X	X	X	X	X
596	X	X	X	X	X	X
662	X	X	X	X	X	X
665	X	X	X	X	X	X
670	X	X	X	X	X	X
686	X	X	X	X	X	X
700	X	X	X	X	X	X
710						X

Table 13. Cross-Reference of Errata to Processor Segments (Continued)

Errata Number	AMD A-Series Mobile APU	AMD E2-Series Mobile APU	AMD A-Series APU	AMD E2-Series APU	AMD Sempron Dual-Core Processor	AMD Athlon II Dual-Core and Quad-Core Processors
711	X	X	X	X	X	X
721	X	X	X	X	X	X
725	X	X	X	X	X	X

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

None required.

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

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

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 (D18F1xC0 and D18F1xC4) for the legacy VGA registers when D18F1xC0[4] (VE) is 1b and D18F1xF4[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

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

343 Eviction May Occur When Using L2 Cache as General Storage During Boot

Description

When system software is using the L2 cache as general storage before memory initialization, the processor may determine during speculative execution that data destined for the instruction cache is dirty. The processor will then evict these cache lines, resulting in lost data.

Potential Effect on System

System software using L2 cache as general storage before memory initialization may experience unpredictable system behavior.

Suggested Workaround

System software should set MSRC001_102A[35] to 1b prior to using L2 cache as general storage during boot. System software should clear MSRC001_102A[35] to 0b after the L2 cache is no longer used as general storage.

Fix Planned

No

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

366 Improper DIMM On-Die Termination Signaling May Occur

Description

Certain memory configurations may cause improper DIMM on-die termination (ODT) signaling when the AMD recommended settings for DCT ODT Control (D18F2x[1,0]F4_x180 and D18F2x[1,0]F4_x182) are not used. The AMD recommended values for D18F2x[1,0]F4_x180 and D18F2x[1,0]F4_x182 are provided in the *BIOS and Kernel Developer's Guide (BKDG) for AMD Family 12h Processors*, order# 41131.

Potential Effect on System

Unreliable DRAM signaling.

Suggested Workaround

D18F2x[1,0]F4_x180 and D18F2x[1,0]F4_x182 should remain at the AMD recommended values.

Fix Planned

No

418 Host Mapping of Physical Page Zero May Cause Incorrect Translation

Description

The processor may use an incorrect cached copy of translation tables during an SVM nested page translation when the host is in legacy Physical Address Extension (PAE) mode and the guest address translation tables reside in physical page zero.

Potential Effect on System

Unpredictable system behavior. This condition has not been observed in any commercially available software.

Suggested Workaround

Hypervisor software should not use physical addresses 0 through 4095 for guest pages.

Fix Planned

No

430 Processor May Not Recognize A20M# Change During CC6

Description

A processor core that is in Core C6 (CC6) state at the time that A20 Mask (A20M#) changes state may continue to operate using the previous A20M# value after CC6 exit.

Potential Effect on System

None expected. Since no multi-core operating system uses A20M#, state changes during CC6 state are not expected.

Suggested Workaround

None required.

Fix Planned

No

432 DEV Error May Be Erroneously Logged After a Warm Reset

Description

An uncorrectable DMA Exclusion Vector (DEV) table walk error may erroneously be logged if a warm reset is initiated while a DEV table walk is in progress.

Potential Effect on System

A false uncorrectable error may be reported to system software.

Suggested Workaround

None required.

Fix Planned

No

441 Move from Stack Pointer to Debug or Control Register May Result in Incorrect Value

Description

A move from the stack pointer to a debug register or a control register may store a value that does not include one or more updates based on completed pushes, pops, near calls or returns. This erratum does not occur if the instruction encoding uses the standard encoding of ModRM[7:6]=11b to indicate a register-to-register move.

Potential Effect on System

None expected based on the ModRM[7:6] normally being 11b.

Suggested Workaround

Always encode ModRM[7:6]=11b when performing a move into a debug or control register.

Fix Planned

No

465 First MRS Command After DRAM Initialization May Time-out

Description

The first DIMM Mode Register Set (MRS) command after DRAM Initialization Register[EnDramInit] (D18F2x[1,0]7C[31]) is set may take up to 2.5 ms to complete.

Potential Effect on System

BIOS time-out may occur resulting in a boot failure.

Suggested Workaround

BIOS must use a time-out value greater than 2.5 ms for the MRS command sequence.

Fix Planned

No

470 Warm Reset May Cause System Hang

Description

The processor may hang if a warm reset occurs while a register access to any of the PCI Express® controllers' internal registers is in progress. The processor may perform register accesses for internal management purposes that are transparent to software.

Potential Effect on System

System hang.

Suggested Workaround

System BIOS should set D0F0xE4_x013[2:0]_8063 bits 4, 5, 12, 13 and 14 to 1b. This must be done early in the BIOS boot sequence to minimize the possibility of a hang due to a warm reset during the boot sequence.

During a link reconfigure operation, system BIOS must perform the following steps:

1. Clear D0F0xE4_x013[2:0]_8063 bits 4, 5, 12, 13 and 14 to 0b.
2. Perform the link reconfigure operation.
3. Set D0F0xE4_x013[2:0]_8063 bits 4, 5, 12, 13 and 14 to 1b.

Fix Planned

No

474 Memory Clear Feature May Use Non-Zero Pattern

Description

During a memory clear function (DRAM Controller Select Low Register[MemClrInit], D18F2x110[3]), the processor may not use zeros as the write pattern.

Potential Effect on System

Memory is not cleared to zeros. This may lead to boot failure if BIOS assumes that unused memory is cleared.

Suggested Workaround

Before performing the memory clear function, BIOS should use the continuous pattern generator to perform a cache line write of zeros, followed by a read of the cache line. Consult the read and write pattern generation algorithms in the *BIOS and Kernel Developer's Guide (BKDG) for AMD Family 12h Processors*, order# 41131.

Fix Planned

No

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

When IBS is not enabled at the time that the processor core enters CC6 state, the erratum conditions do not apply.

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

564 Processor May Fail to Set Auto-Halt Restart in SMM Save State

Description

The processor core may not set the auto-halt restart flag (offset FEC9h of the SMM save state area) when a HLT instruction causes the processor core to enter the core C6 (CC6) state and is then interrupted by an SMI. After the SMM code executes the RSM instruction, the processor core does not re-enter the HLT or CC6 state due to this incorrect auto-halt restart flag.

Potential Effect on System

The processor may continue execution after the HLT instruction, resulting in unpredictable system behavior. The operating system is not required to have a valid instruction after a HLT instruction when rFLAGS.IF = 1.

Suggested Workaround

Contact your AMD representative for information on a BIOS update.

Fix Planned

No

565 Processor Cores Observe Separate IBS Control Registers

Description

The processor implements the IBS Control Register as multiple registers, one per processor core. A read of this register using either MSRC001_103A or D18F3x1CC from one processor core may not observe a write to D18F3x1CC that has been performed by another processor core.

Potential Effect on System

Performance monitoring software using Instruction-Based Sampling (IBS) may incorrectly detect that no Local Vector Table (LVT) has been assigned to the IBS interrupt.

Suggested Workaround

BIOS should write D18F3x1CC to the same value using all enabled processor cores. System software is not expected to modify the BIOS value, although in the event that system software writes D18F3x1CC, it must also write D18F3x1CC using all processor cores.

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

Contact your AMD representative for information on a BIOS update.

Fix Planned

No

596 Northbridge Clock Gating May Lead to Invalid Prefetched Data

Description

Under a highly specific and detailed set of internal timing conditions, the northbridge may gate the clock (NCLK) to the In-Flight Queue (IFQ) while a DRAM prefetch caused by a CPU fetch is still outstanding. This may result in the prefetch buffer being marked valid but with unpredictable data. IFQ clock gating is performed only when no cores are in C0 state.

Potential Effect on System

Unpredictable system behavior. This has not been observed with any commercially available software.

Suggested Workaround

BIOS should not set Clock Power/Timing Control 2 Register[NbClockGateEn] (D18F3xDC[30]) and leave this bit at its reset value of 0b.

Fix Planned

No

662 Processor May Hang While Performing Voltage Transitions Overlapping with C-state Requests

Description

A processor may hang while performing overlapping voltage transitions and C-state requests. The hang may result when one of the following conditions occurs:

- The processor performs a core performance boost (CPB) transition that requires a voltage transition just prior to a core entering C1 state (HLT instruction or I/O C-state based). Prior to this voltage transition completing, the processor core transitions from C1 to C0 state and another C-state is requested by software. This new C-state flushes the processor caches and performs core C6 (CC6) clock gating.
- The processor performs a hardware thermal control (HTC) transition that requires a voltage transition just prior to a core entering C1 state (HLT instruction or I/O C-state based). Prior to this voltage transition completing, the processor core recognizes a second HTC transition and the processor core transitions from C1 to C0 state and another C-state is requested by software. This new C-state flushes the processor caches and performs core C6 (CC6) clock gating.

The hang is possible only if the initial voltage transition has not completed by the time the CC6 clock gating occurs.

Potential Effect on System

Processor core hang

Suggested Workaround

Contact your AMD representative for information on a BIOS update.

Fix Planned

No

665 Integer Divide Instruction May Cause Unpredictable Behavior

Description

Under a highly specific and detailed set of internal timing conditions, the processor core may abort a speculative DIV or IDIV integer divide instruction (due to the speculative execution being redirected, for example due to a mispredicted branch) but may hang or prematurely complete the first instruction of the non-speculative path.

Potential Effect on System

Unpredictable system behavior, usually resulting in a system hang.

Suggested Workaround

BIOS should set MSRC001_1029[31].

This workaround alters the DIV/IDIV instruction latency specified in the *Software Optimization Guide for AMD Family 10h and 12h Processors*, order# 40546. With this workaround applied, the DIV/IDIV latency for AMD Family 12h Processors are similar to the DIV/IDIV latency for AMD Family 10h Processors.

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

For affected instructions that have an implicit or explicit LOCK prefix, a system hang occurs.

For affected instructions that do not have an implicit or explicit LOCK prefix, 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

686 Processor Does Not Implement MSRC001_0055

Description

The processor does not properly allow writes to MSRC001_0055 (Interrupt Pending Register). A write to MSRC001_0055 is ignored and a read to the register returns zero.

Potential Effect on System

BIOS is unable to program this register.

Suggested Workaround

Contact your AMD representative for a BIOS workaround.

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

710 AD631XOJZ43GX Incorrectly Reports a Maximum Rate of DDR3-1600

Description

OPN AD631XOJZ43GX (65W FM1 AMD Athlon™ II X4 631X Quad-Core Processor) may incorrectly report a maximum DDR transfer rate of 1600 MT/s. The actual maximum transfer rate of this OPN should be 1866 MT/s. Northbridge Capabilities Register[DdrMaxRate], D18F3xE8[7:5] may be equal to 001b, indicating DDR3-1600. The correct value should be 000b.

Potential Effect on System

The BIOS may incorrectly limit the DDR rate with DDR3-1866 DIMMs to 1600 MT/s.

Suggested Workaround

No workaround is necessary with DDR3-1600 (or lower) DIMMs, on a FS1 motherboard, or on a FM1 motherboard that has two DIMM slots per channel.

System developers with an FM1 motherboard that has only one DIMM slot per channel and that use this OPN with DDR3-1866 DIMMs may require a workaround.

To implement a workaround, platform BIOS replaces the actual reported value in the Northbridge Capabilities Register[DdrMaxRate] field (D18F3xE8[7:5]) with 000b if the above affected processor type (OPN) is detected. The affected OPN can be uniquely identified if all of the following characteristics are true:

- CPUID Fn0000_0001_EAX = 00300F10h (LN-B0)
- CPUID Fn8000_0001_EBX BrandId Identifier[PkgType, bits 31:28] = 0010b (FM1)
- D18F3xA0[27:16] = 02Ah

Fix Planned

Yes

711 Processor May Hang During PCIe® Initialization

Description

The processor may hang during BIOS boot on a downstream transaction to a PCIe® interface.

Potential Effect on System

System hang during BIOS boot.

Suggested Workaround

Contact your AMD representative for a BIOS workaround.

Fix Planned

No

721 Processor May Incorrectly Update Stack Pointer

Description

Under a highly specific and detailed set of internal timing conditions, the processor may incorrectly update the stack pointer after a long series of push and/or near-call instructions, or a long series of pop and/or near-return instructions. The processor must be in 64-bit mode for this erratum to occur.

Potential Effect on System

The stack pointer value jumps by a value of approximately 1024, either in the positive or negative direction. This incorrect stack pointer causes unpredictable program or system behavior, usually observed as a program exception or crash (for example, a #GP or #UD).

Suggested Workaround

System software may set MSRC001_1029[0] = 1b.

Fix Planned

No

725 Incorrect APIC Remote Read Behavior

Description

The processor may provide incorrect APIC register data on an APIC remote register read. A remote read is performed using Interrupt Command Register Low[MsgType] of 011b (APIC300[10:8]). The processor may, but does not always, provide an error indication in the remote read status field (APIC300[17:16]).

This erratum does not impact the use of remote APIC reads by BIOS during early power-on-self-test (POST) when the remote read is performed for addresses APIC300-APIC3F0.

Potential Effect on System

None expected, as it is anticipated that no software other than BIOS uses remote APIC reads.

Suggested Workaround

Software should not use remote APIC reads.

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 12h Processors*, order# 41131
- *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
- *Software Optimization Guide for AMD Family 10h and 12h Processors*, order# 40546

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