Revision Guide for AMD Family 14h Models 00h-0Fh Processors

Publication # **47534**	Revision: **3.18**
Issue Date: **February 2013**	

Advanced Micro Devices

The contents of this document are provided in connection with Advanced Micro Devices, Inc. (“AMD”) products. AMD makes no representations or warranties with respect to the accuracy or completeness of the contents of this publication and reserves the right to make changes to specifications and product descriptions at any time without notice. The information contained herein may be of a preliminary or advance nature and is subject to change without notice. No license, whether express, implied, arising by estoppel or otherwise, to any intellectual property rights is granted by this publication. Except as set forth in AMD’s Standard Terms and Conditions of Sale, AMD assumes no liability whatsoever, and disclaims any express or implied warranty, relating to its products including, but not limited to, the implied warranty of merchantability, fitness for a particular purpose, or infringement of any intellectual property right.

AMD’s products are not designed, intended, authorized or warranted for use as components in systems intended for surgical implant into the body, or in other applications intended to support or sustain life, or in any other application in which the failure of AMD’s product could create a situation where personal injury, death, or severe property or environmental damage may occur. AMD reserves the right to discontinue or make changes to its products at any time without notice.

Trademarks

AMD, the AMD Arrow logo, AMD Radeon, and combinations thereof, are trademarks of Advanced Micro Devices, Inc.

PCIe and PCI Express are registered trademarks of PCI-SIG.

Other product names used in this publication are for identification purposes only and may be trademarks of their respective companies.

Revision History

Date	Revision	Description
February 2013	3.18	Added errata #770, #780 and #784.
August 2012	3.16	Updated Table 5 for branding; Added errata #737-#738 and #747.
March 2012	3.12	Added silicon revision (ON-C0 with Fixed Errata Register bit 2 = 1b) to Table 2 and Table 8; Updated Tables 3 and 5; Updated D18F4x164 Fixed Errata Register; Updated erratum #688 in Table 8; Added errata #662, #686, #700 and #725.
October 2011	3.08	Updated Register References and Mnemonics; Added AMD Z-Series Processor to Overview, Tables 2, 5 and 9; Added silicon revision (ON-C0) to Processor Identification and Tables 7-8; Added Graphic Device IDs; Updated Table 6 for branding; Updated fix-plan for #595; Added errata #629, #632, #634, #639, #651 and #688
January 2011	3.00	Initial public release.

Revision Guide for AMD Family 14h Models 00h-0Fh Processors

Overview

The purpose of the *Revision Guide for AMD Family 14h Models 00h-0Fh Processors* is to communicate updated product information to designers of computer systems and software developers. This revision guide includes information on the following products:

- AMD C-Series Processor with AMD Radeon™ HD Graphics
- AMD C-Series Dual-Core Processor with AMD Radeon HD Graphics
- AMD E-Series Dual-Core Processor
- AMD E-Series Processor with AMD Radeon HD Graphics
- AMD E-Series Dual-Core Processor with AMD Radeon HD Graphics
- AMD G-Series Processor with AMD Radeon HD Graphics
- AMD G-Series Dual-Core Processor with AMD Radeon HD Graphics
- AMD Z-Series Dual-Core Processor with AMD Radeon HD Graphics

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 72, 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 14h Models 00h-0Fh Processors*, order# 43170. 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 14h Models 00h-0Fh Processors*, order# 43170 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 14h Models 00h-0Fh Processors*, order# 43170 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).

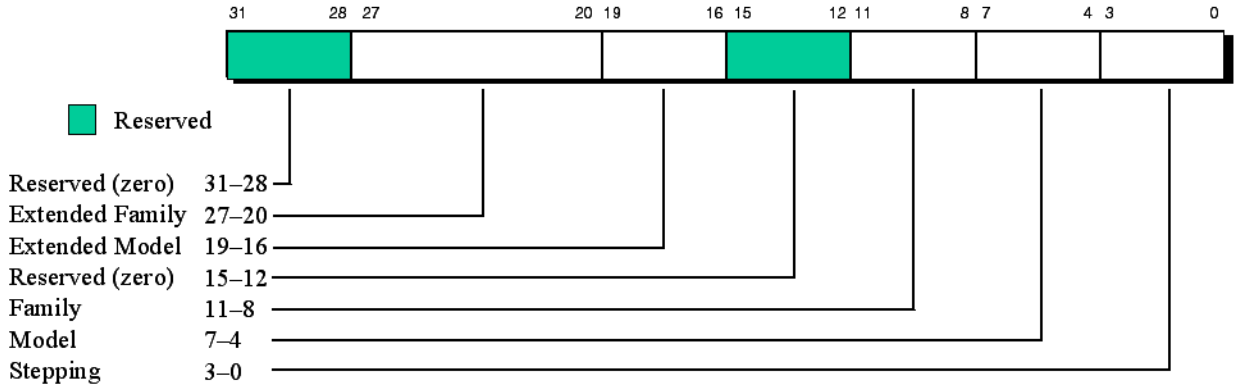

Figure 1. Format of CPUID Fn0000_0001_EAX

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 14h Models 00h-0Fh Processor Revisions

CPUID Fn0000_0001_EAX (Mnemonic)	D18F4x164	AMD C-Series Processor	AMD E-Series Processor	AMD G-Series Processor	AMD Z-Series Processor
00500F10h (ON-B0)	00000001h	X	X	X	N/A
00500F10h (ON-B0)	00000003h	X	X	X	X
00500F20h (ON-C0)	00000003h	X	X	X	X
00500F20h (ON-C0)	00000007h	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 3 for a list of graphics device ID values in use for AMD Family 14h Models 00h-0Fh Accelerated Processing Units.

Table 3. AMD Family 14h Models 00h-0Fh Graphic Device IDs

D1F0x00[31:16]	Notes
9802h	Client and Embedded
9803h	Client and Embedded
9804h	Client, Embedded and Tablet
9805h	Client and Embedded
9806h	Client and Embedded
9807h	Client and Embedded
9808h	Client
9809h	Client

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 14h Models 00h-0Fh Processors*, order# 43170, 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 14h Models 00h-0Fh Processors*, order# 43170, 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 BIOS constructs 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 5.
  - **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 6.
  - **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 14h Models 00h-0Fh Processors*, order# 43170, and is used to index the appropriate string tables from Table 4.
- 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 14h Models 00h-0Fh Processors*, order# 43170, and is used to index the appropriate strings from Tables 5 through 6. 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 00-99and 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 4.
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 4. String Table Reference Per Package Type

PkgType [3:0]	String1 Table	String2 Table
0h	Table 5	Table 6
1h-Fh	Reserved	Reserved

Table 5. String1 Values for FT1 Processors

Pg[0]	NC [7:0]	String1 [3:0]	Value	Note	Description
0b	0h	1h	AMD C-	-	Client.
		2h	AMD E-	-	Client.
		4h	AMD G-T	-	Embedded.
	1h	1h	AMD C-	-	Client.
		2h	AMD E-	-	Client.
		3h	AMD Z-	-	Tablet.
		4h	AMD G-T	-	Embedded.
		5h	AMD E1-1	-	Client.
		6h	AMD E2-1	-	Client.
		7h	AMD E2-2	-	Client.
All other values			AMD Processor Model Unknown	-	

Table 6. String2 Values for FT1 Processors

Pg[0]	NC [7:0]	String2 [3:0]	Value	Note	Description
0b	0h	1h	Processor	1	
		2h	0 Processor		
		3h	5 Processor		
		4h	0x Processor		
		5h	5x Processor		
		6h	x Processor		
		7h	L Processor		
		8h	N Processor		
		9h	R Processor		
		Ah	0 APU with Radeon(tm) HD Graphics		
		Bh	5 APU with Radeon(tm) HD Graphics		
		Ch	APU with Radeon(tm) HD Graphics	1	
		Dh	0D APU with Radeon(tm) HD Graphics		
	1h	1h	Processor	1	
		2h	0 Processor		
		3h	5 Processor		
		4h	0x Processor		
		5h	5x Processor		
		6h	x Processor		
		7h	L Processor		
		8h	N Processor		
		9h	0 APU with Radeon(tm) HD Graphics		
		Ah	5 APU with Radeon(tm) HD Graphics		
		Bh	APU with Radeon(tm) HD Graphics	1	
		Ch	E Processor		
		Dh	0D APU with Radeon(tm) HD Graphics		
	xxh	Fh		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
31:3	0000_0000h. Reserved.
2	This fixed errata register bit identifies processor revisions that share CPUID Fn0000_0001_EAX = 005000F20h (ON-C0). 0 = Erratum #688 is applicable, as shown in Table 8 and a BIOS workaround is recommended. 1 = Erratum #688 is not applicable, as shown in Table 8 and the BIOS workaround is not necessary or recommended.
1	This fixed errata register bit identifies processor revisions that share CPUID Fn0000_0001_EAX = 005000F10h (ON-B0). 0 = Erratum #578 is applicable, as shown in Table 8. 1 = Erratum #578 is not applicable, as shown in Table 8.
0	This bit is always 1b.

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
63:5	**OsvwStatusBits:** Reserved. OS visible work-around status bits. Read-write.
4	**Osvwld4:** Reserved, must be zero.
3	**Osvwld3:** Reserved, must be zero.
2	**Osvwld2:** Reserved, must be zero.
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 7 prior to hand-off to the OS.

Table 7. Cross Reference of Product Revision to OSVW ID

CPUID Fn0000_0001_EAX (Mnemonic)	MSRC001_0141 Bits
00500F10h (ON-B0)	0000_0000_0000_0000h
00500F20h (ON-C0)	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 8 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. "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 8. Cross-Reference of Processor Revision to Errata

No.	Errata Description	CPUID Fn0000_0001_EAX D18F4x164[2:0] (Mnemonic)			
		00500F10h 001b (ON-B0)	00500F10h 011b (ON-B0)	00500F20h 011b (ON-C0)	00500F20h 111b (ON-C0)
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			
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			
432	DEV Error May Be Erroneously Logged After a Warm Reset	No fix planned			
433	Performance Counters Do Not Accurately Count Memory Turnaround Time	No fix planned			
434	Processor May Violate Twr for Precharge Commands	No fix planned			
435	Precharge Failure After Self Refresh Exit	No fix planned			
441	Move from Stack Pointer to Debug or Control Register May Result in Incorrect Value	No fix planned			
455	Certain GPU and Northbridge Clock Combinations May Result in Unpredictable System Behavior	No fix planned			
461	CLFLUSH Instruction to I/O Address May Cause System Hang	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			
473	Low Power Drive Strength May Use Improper Calibration Code	No fix planned			
474	Memory Clear Feature May Use Non-Zero Pattern	No fix planned			
484	Northbridge Instruction-Based Sampling Fields Are Not Sampled for Write-Combining Operations	No fix planned			

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

No.	Errata Description	CPUID Fn0000_0001_EAX D18F4x164[2:0] (Mnemonic)			
		00500F10h 001b (ON-B0)	00500F10h 011b (ON-B0)	00500F20h 011b (ON-C0)	00500F20h 111b (ON-C0)
530	Potential Violation of Read Ordering Rules Between Semaphore Operation and Subsequent Load Operations	No fix planned			
541	IBS Registers May Be Unpredictable After CC6 State	No fix planned			
551	Processor May Not Forward Data From Store to a Page Crossing Read-Modify-Write Operation	No fix planned			
560	Processor May Incorrectly Forward Data with Non-cacheable Floating-Point 128-bit SSE Operation	No fix planned			
561	Processor May Incorrectly Walk Page Tables in I/O or Non-Cacheable Memory	No fix planned			
562	Processor Executing VMRUN to Interrupt Shadow Instruction May Lose a Breakpoint	No fix planned			
563	Processor May Store Incorrect Resume Flag in VMCB	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			
578	Branch Prediction May Cause Incorrect Processor Behavior	X			
579	Processor May Generate #GP Exception on CLFLUSH to Execute-Only Code Segment	No fix planned			
580	Instruction-Based Sampling May Bias Execution Samples	No fix planned			
581	Processor May Perform Incorrect Large Granularity TSS Limit Checking	No fix planned			
594	Move to CR3 May Cause Unexpected #GP Exception in PAE Mode	No fix planned			
595	Load Effective Address with Stack Pointer May Result in Incorrect Value	X	X		
596	Northbridge Clock Gating May Lead to Invalid Prefetched Data	No fix planned			
629	Processor May Install Duplicate Entries in L1 TLB	No fix planned			
632	LDT or GDT Segment Descriptor Wrapping at 4GB Boundary Causes Incorrect Operation	No fix planned			
634	Invalid Guest Address in CR3 May Cause System Hang	No fix planned			
639	Processor May Have Incorrect Instruction Pointer Following a Specific CALL Instruction Encoding	No fix planned			
651	CLTS or LMSW Instruction Executed in Real Mode May Cause Unpredictable Behavior	No fix planned			
662	Processor May Hang While Performing Voltage Transitions Overlapping with C-state Requests	No fix planned			
686	Processor Does Not Implement MSRC001_0055	No fix planned			
688	Processor May Cause Unpredictable Program Behavior Under Highly Specific Branch Conditions	X	X	X	
700	LAR and LSL Instructions Do Not Check Invalid Long Mode Descriptor Types	No fix planned			
725	Incorrect APIC Remote Read Behavior	No fix planned			

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

No.	Errata Description	CPUID Fn0000_0001_EAX D18F4x164[2:0] (Mnemonic)			
		00500F10h 001b (ON-B0)	00500F10h 011b (ON-B0)	00500F20h 011b (ON-C0)	00500F20h 111b (ON-C0)
737	Processor Does Not Check 128-bit Canonical Address Boundary Case on Logical Address	No fix planned			
738	Debug Exception Concurrent with Unintercepted SMI May Store Incorrect Stack Pointer in VMCB	No fix planned			
747	Spurious Debug Exception May Be Observed	No fix planned			
770	Processor Incorrectly Restores Guest Privilege Level after Unintercepted I/O C-state Request	No fix planned			
780	Processor May Cache Guest Write Combining Memory Type	No fix planned			
784	Processor May Incorrectly Provide Control Register Data as the Result of a Load Operation	No fix planned			

Table 9 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 9. Cross-Reference of Errata to Processor Segments

Errata Number	AMD C-Series Processor	AMD E-Series Processor	AMD G-Series Processor	AMD Z-Series Processor
77	X	X	X	X
230	X	X	X	X
250	X	X	X	X
361	X	X	X	X
366	X	X	X	X
432	X	X	X	X
433	X	X	X	X
434	X	X	X	X
435	X	X	X	X
441	X	X	X	X
455	X	X	X	X
461	X	X	X	X
465	X	X	X	X
470	X	X	X	X
473	X	X	X	X
474	X	X	X	X
484	X	X	X	X
530	X	X	X	X
541	X	X	X	X
551	X	X	X	X
560	X	X	X	X
561	X	X	X	X
562	X	X	X	X
563	X	X	X	X
564	X	X	X	X
565	X	X	X	X
578	X	X	X	
579	X	X	X	X
580	X	X	X	X
581	X	X	X	X
594	X	X	X	X
595	X	X	X	X

Table 9. Cross-Reference of Errata to Processor Segments (Continued)

Errata Number	AMD C-Series Processor	AMD E-Series Processor	AMD G-Series Processor	AMD Z-Series Processor
596	X	X	X	X
629	X	X	X	X
632	X	X	X	X
634	X	X	X	X
639	X	X	X	X
651	X	X	X	X
662	X	X	X	X
686	X	X	X	X
688	X	X	X	X
700	X	X	X	X
725	X	X	X	X
737	X	X	X	X
738	X	X	X	X
747	X	X	X	X
770	X	X	X	X
780	X	X	X	X
784	X	X	X	X

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

Description

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

Potential Effect on System

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

Suggested Workaround

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

Fix Planned

No

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

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 (D18F2xF4_x180 and D18F2xF4_x182) are not used. The AMD recommended values for D18F2xF4_x180 and D18F2xF4_x182 are provided in the *BIOS and Kernel Developer's Guide (BKDG) for AMD Family 14h Models 00h-0Fh Processors*, order# 43170.

Potential Effect on System

Unreliable DRAM signaling.

Suggested Workaround

D18F2xF4_x180 and D18F2xF4_x182 should remain at the AMD recommended values.

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

433 Performance Counters Do Not Accurately Count Memory Turnaround Time

Description

Write-to-read and read-to-write memory controller turnaround events are not accurately counted by PMCx0E3 (Memory Controller Turnarounds).

Potential Effect on System

Performance monitoring software will not be able to accurately itemize the DRAM bandwidth used for turnaround events.

Suggested Workaround

None.

Fix Planned

No

434 Processor May Violate Twr for Precharge Commands

Description

The processor may violate Twr for precharge commands when self-refresh commands are pending.

Potential Effect on System

Unpredictable system behavior.

Suggested Workaround

When converting the DIMM timing parameter for Twr to the encoded Twr value in the DRAM MRS Register[Twr] (D18F2x84[6:4], BIOS should add one MEMCLK to the number of clocks required.

Fix Planned

No

435 Precharge Failure After Self Refresh Exit

Description

Under a highly specific and detailed set of internal timing conditions, a processor may open a DRAM page that is already open. This may occur if a northbridge P-state transition occurs shortly after a self-refresh exit, and results in inconsistent memory controller state.

Potential Effect on System

Unpredictable system behavior, usually leading to a system hang.

Suggested Workaround

BIOS should clear DRAM Configuration High[PowerDownMode] (D18F2x94[16]) to 0b.

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

455 Certain GPU and Northbridge Clock Combinations May Result in Unpredictable System Behavior

Description

The processor may violate internal timing requirements at certain clock combinations of LCLK and NCLK. No violation occurs at boot and will not occur until both LCLK and NCLK exceed 200 MHz.

Potential Effect on System

Unpredictable system behavior.

Suggested Workaround

Contact your AMD representative for information on a BIOS update.

Fix Planned

No

461 CLFLUSH Instruction to I/O Address May Cause System Hang

Description

Under a highly specific and detailed set of conditions, a CLFLUSH instruction to an I/O address may not properly deallocate an internal buffer used for cache operations.

Potential Effect on System

System hang may occur after repetitive occurrences.

Suggested Workaround

Contact your AMD representative for information on a BIOS update.

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] (D18F2x7C[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_x0130_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_x0130_8063 bits 4, 5, 12, 13 and 14 to 0b.
2. Perform the link reconfigure operation.
3. Set D0F0xE4_x0130_8063 bits 4, 5, 12, 13 and 14 to 1b.

Fix Planned

No

473 Low Power Drive Strength May Use Improper Calibration Code

Description

During a self-refresh operation, a simultaneous DDR calibration code update may latch incorrect data used to calibrate the low power drive strength.

Potential Effect on System

The DRAM may incorrectly sense a CKE transition during self-refresh mode. This causes the DRAM to prematurely exit from self-refresh mode, leading to the loss of memory contents.

Suggested Workaround

BIOS normally programs the Phy Calibration Configuration Register[DisAutoComp] (D18F2x9C_x0D0F_E003[14]) to 1b when performing the Phy Compensation Initialization sequence as described in the *BIOS and Kernel Developer's Guide (BKDG) for AMD Family 14h Models 00h-0Fh Processors*, order# 43170. As a workaround to this erratum, BIOS should not re-enable the auto-compensation after this sequence and should keep DisAutoComp=1.

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 14h Models 00h-0Fh Processors*, order# 43170.

Fix Planned

No

484 Northbridge Instruction-Based Sampling Fields Are Not Sampled for Write-Combining Operations

Description

The IBS Op Data 2 Register (MSRC001_1036) does not provide valid status when an access to write-combining (WC) memory is tagged (IBS Op Data 3 Register[IbsDcWcMemAcc], MSRC001_1037[13], is 1b).

Potential Effect on System

Inaccuracies in performance monitoring software may be experienced.

Suggested Workaround

None.

Fix Planned

No

530 Potential Violation of Read Ordering Rules Between Semaphore Operation and Subsequent Load Operations

Description

Under a highly specific and detailed set of internal timing conditions, the memory read ordering between a semaphore operation and a subsequent load operation may be incorrect and allow the load operation to operate on the memory location ahead of the completion of the semaphore operation.

Potential Effect on System

In the unlikely event that the condition described above occurs, the load operation (in the critical section) may operate on data that existed prior to the semaphore operation.

Suggested Workaround

BIOS should set MSRC001_1020[36].

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

551 Processor May Not Forward Data From Store to a Page Crossing Read-Modify-Write Operation

Description

A read-modify-write operation that crosses a page boundary may not see the results of a previous store.

Potential Effect on System

Unpredictable system behavior.

Suggested Workaround

BIOS should set MSRC001_1020[25].

Fix Planned

No

560 Processor May Incorrectly Forward Data with Non-cacheable Floating-Point 128-bit SSE Operation

Description

Under a highly specific and detailed set of internal timing conditions, the processor may perform a non-cacheable floating-point 128-bit SSE instruction and forward the data from this instruction to a nearby unrelated non-cacheable load.

Potential Effect on System

No effect is expected under normal system operation, as software that performs floating-point 128-bit SSE instructions to non-cacheable memory types (CD, UC, or WC) is not anticipated. Under the highly unlikely circumstances that this software exists, unpredictable system behavior may occur due to the incorrect forwarding of data.

Suggested Workaround

BIOS should set MSRC001_1020[18].

Fix Planned

No

561 Processor May Incorrectly Walk Page Tables in I/O or Non-Cacheable Memory

Description

When HWCR[TlbCacheDis] (MSRC001_0015[3]) is 1b and the page tables reside in I/O or DRAM that is marked non-cacheable, the processor may incorrectly perform page table walks.

Potential Effect on System

Unpredictable system behavior.

Suggested Workaround

None required. No operating system places page tables in memory that is not marked as WB DRAM, and therefore HWCR[TlbCacheDis] can always be at its reset state of 0b.

Fix Planned

No

562 Processor Executing VMRUN to Interrupt Shadow Instruction May Lose a Breakpoint

Description

The processor may not recognize a code breakpoint #DB exception when all of the following conditions are true:

- The hypervisor is executing a VMRUN instruction with the interrupt shadow field set (VMCB offset 068h bit 0), specifying that the first guest instruction that is to be executed is in an interrupt shadow.
- The code breakpoint is on this first instruction (the instruction that is in the guest interrupt shadow).
- The global interrupt flag (GIF) for the guest virtual machine is set.
- The global interrupt flag (GIF) for the host is not set.

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

563 Processor May Store Incorrect Resume Flag in VMCB

Description

A processor core may incorrectly store rFLAGS.RF = 1 in the VMCB when the processor is performing an SVM interception on a CLI instruction, including an intercept on an exception that occurred during execution of the CLI instruction.

Potential Effect on System

None expected under normal conditions. Instruction breakpoints on CLI instructions may not be received for programs running under a hypervisor.

Suggested Workaround

None required.

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

578 Branch Prediction May Cause Incorrect Processor Behavior

Description

Under a highly specific and detailed set of internal timing conditions involving multiple events occurring within a small window of time, the processor branch prediction logic may cause the processor core to decode incorrect instruction bytes.

Potential Effect on System

Unpredictable program behavior, generally leading to a program exception.

Suggested Workaround

None.

Fix Planned

Yes

579 Processor May Generate #GP Exception on CLFLUSH to Execute-Only Code Segment

Description

The processor generates a #GP exception while executing a CLFLUSH instruction to an execute-only code segment (CS descriptor with Readable attribute bit = 0b, bit 9 of descriptor byte 4). This erratum affects only legacy and compatibility modes, as the readable attribute is deprecated in 64-bit mode.

Potential Effect on System

Unexpected program (#GP) exception.

Suggested Workaround

None expected. CLFLUSH instructions to an execute-only code segment are not normally performed in commercially available software.

Fix Planned

No

580 Instruction-Based Sampling May Bias Execution Samples

Description

The Instruction-Based Sampling (IBS) execution samples may be biased and not match expected results:

- The processor execution sample engine may select some instructions at a higher probability than other instructions that have a similar execution frequency. This error may be significant, especially when the code includes floating-point double operations.
- The processor may not perform an IBS sample after the specified number of clock cycles expires (IBS Execution Control[IbsOpCntCtl], MSRC001_1033[19] = 0b). When this occurs, the counter rolls over and the current sample is delayed for some multiple of the counter. This issue has no impact on the more common case of counting dispatched operations instead of clock cycles.

Potential Effect on System

Inaccuracies in performance monitoring software may be experienced.

Suggested Workaround

None. This erratum may prevent the software from obtaining statistically accurate data from IBS execution sampling.

Fix Planned

No

581 Processor May Perform Incorrect Large Granularity TSS Limit Checking

Description

After resuming from System Management Mode (SMM), the processor may not scale Task-State Segment (TSS) limits when the TSS descriptor granularity bit (bit 23 of descriptor byte 4) is 1b. The granularity (G) bit, when set, indicates that the segment-limit field should be scaled by 4096 bytes.

Potential Effect on System

Unexpected #GP exception, leading to system crash. This has not been observed with any commercially available operating system.

Suggested Workaround

Operating system software must use G=0 for all TSS descriptors. The largest possible size of a TSS descriptor does not require G=1.

Fix Planned

No

594 Move to CR3 May Cause Unexpected #GP Exception in PAE Mode

Description

Under highly intermittent internal timing conditions, the processor may incorrectly sense that a reserved bit is set in the Page Directory Pointer Table entries (PDPE), even when no reserved bits are set. The PDPE is checked for reserved bits during a move to CR3 (MOV CR3) instruction.

This error is intermittent and observable only when all of the following conditions are met:

- The processor is in legacy mode with physical address extension (PAE) enabled (CR4.PAE = 1).
- The MOV CR3 is the first MOV CR3 instruction executed after a reset or CC6 exit.

Potential Effect on System

Unexpected #GP exception during a MOV CR3 instruction, leading to a kernel panic or system crash.

Suggested Workaround

Contact your AMD representative for information on a BIOS update.

Fix Planned

No

595 Load Effective Address with Stack Pointer May Result in Incorrect Value

Description

Uncommon usages of the Load Effective Address (LEA) instruction, where the stack pointer is added to one of {R8 .. R15} without a displacement and without a scale applied to that register, may utilize the stack pointer with a value that does not include one or more updates based on completed pushes, pops, near calls or returns. This erratum can occur only when the processor is in 64-bit mode.

The following is a complete list of instructions that may observe this incorrect behavior:

- LEA <any destination>, [RSP + R8]
- LEA <any destination>, [RSP + R9]
- LEA <any destination>, [RSP + R10]
- LEA <any destination>, [RSP + R11]
- LEA <any destination>, [RSP + R12]
- LEA <any destination>, [RSP + R13]
- LEA <any destination>, [RSP + R14]
- LEA <any destination>, [RSP + R15]

Potential Effect on System

Unpredictable system behavior.

Suggested Workaround

Contact your AMD representative for information on a BIOS update.

Fix Planned

Yes

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

629 Processor May Install Duplicate Entries in L1 TLB

Description

Under a highly specific and detailed set of internal timing conditions, the processor may move an entry from the L2 TLB to the L1 TLB that has already been installed in the L1 TLB.

Potential Effect on System

Uncorrectable machine check exception (MC#) for an L1 TLB Multimatch error. The MC0_STATUS register (MSR0000_0401) is B6000000_00010015h. Bit 62 (error overflow) of MC0_STATUS may or may not be set.

Suggested Workaround

BIOS should set MC0_CTL_MASK[6] (MSRC001_0044[6]) = 1b.

Fix Planned

No

632 LDT or GDT Segment Descriptor Wrapping at 4GB Boundary Causes Incorrect Operation

Description

In 32-bit protected mode, a segment load using a Local Descriptor Table (LDT) or Global Descriptor Table (GDT) system-segment descriptor with a base/limit that wraps around the 4GB boundary may cause an unexpected page fault.

Potential Effect on System

None expected, since the operating system typically aligns the GDT/LDT limit such that no descriptors wrap. In the unlikely event that software creates the conditions described, the processor may generate an unexpected page fault (#PF). The address reported in CR2 is zero.

Suggested Workaround

None.

Fix Planned

No

634 Invalid Guest Address in CR3 May Cause System Hang

Description

The processor may hang when all of the following conditions are met:

- Secure Virtual Machine (SVM) mode is enabled.
- The host is not in long mode (host EFER.LMA is 1).
- The host executes a VMRUN instruction with nested paging enabled (VMCB offset 090h[0], NP_ENABLE, is 1b) and enters a guest that is allowed to go into long mode.
- The guest loads a Page-Map Level-4 Table Base Address into CR3 that is above 4GB. Since the host is not in long mode, the guest's address space is already limited to 4GB (the maximum size of the virtual address allowed in the host). Therefore, the guest is loading an illegal address into CR3.

Potential Effect on System

System hang. This has not been observed with any commercially available hypervisor.

Suggested Workaround

Hypervisors that allow long mode guests must also operate in long mode or take steps to avoid the above erratum conditions, such as intercepting all modifications of CR3 to validate the register.

Fix Planned

No

639 Processor May Have Incorrect Instruction Pointer Following a Specific CALL Instruction Encoding

Description

Under a highly specific and detailed set of internal timing conditions where the processor has performed at least 100 pushes, pops, near-calls and/or near-returns without executing any other operation that uses the stack pointer, the processor may incorrectly execute the following instruction:

- CALL RSP without any offset (instruction encoding FFD4h).

Under the above conditions, the processor does not execute the CALL instruction and instead may treat this instruction as if it is a NOP (no operation) instruction. The processor incorrectly updates the rIP to the address following the CALL.

This CALL RSP instruction should transfer instruction execution to the stack (i.e. it changes the rIP to the value that was in the rSP prior to the execution of the instruction). If the stack address is marked with a no-execute attribute (the NX bit is set in the page table), this generates a #GP exception. The stack is commonly marked with a no-execute attribute. As a result, the use of this instruction encoding is uncommon in applications.

Potential Effect on System

Unpredictable program behavior after incorrectly updating the instruction pointer (rIP). This behavior has not been observed with any commercially available software.

Suggested Workaround

Contact your AMD representative for information on a BIOS workaround.

Fix Planned

No

651 CLTS or LMSW Instruction Executed in Real Mode May Cause Unpredictable Behavior

Description

Under a highly specific and detailed set of conditions that include operation in real mode (CR0.PE = 0b) or System-Management Mode (SMM), the processor may incorrectly perform a nearby segment load while executing a CLTS instruction or an LMSW instruction that does not change CR0 bits.

Potential Effect on System

Unpredictable system behavior. This erratum is not expected to occur with any commercially available operating systems or system-management code.

Suggested Workaround

None required.

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

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 information on a BIOS update

Fix Planned

No

688 Processor May Cause Unpredictable Program Behavior Under Highly Specific Branch Conditions

Description

Under a highly specific and detailed set of internal timing conditions, the processor may incorrectly update the branch status when a taken branch occurs where the first or second instruction after the branch is an indirect call or jump. This may cause the processor to update the rIP (the instruction pointer register) after a not-taken branch that ends on the last byte of an aligned quad-word such that it appears the processor skips, and does not execute, one or more instructions. The new updated rIP due to this erratum may not be at an instruction boundary.

Potential Effect on System

Unpredictable program behavior, possibly leading to a program error or system error. It is also possible that the processor may hang or recognize an exception (for example, a #GP or #UD exception), however AMD has not observed this effect.

Suggested Workaround

BIOS should set MSRC001_1021[14] = 1b and MSRC001_1021[3] = 1b. This workaround is required only when bit 2 of Fixed Errata Status Register (D18F4x164[2]) = 0b.

Fix Planned

Yes

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

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

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

Description

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

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

Potential Effect on System

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

Suggested Workaround

None required.

Fix Planned

No

738 Debug Exception Concurrent with Unintercepted SMI May Store Incorrect Stack Pointer in VMCB

Description

During processing of a RSM instruction to return from a system management interrupt (SMI) that occurred while in secure virtual machine (SVM) mode and without the SMI being intercepted, the processor core may not restore the guest stack pointer prior to presenting a debug exception (#DB) on the resumed guest instruction. In the event that this #DB is intercepted (or some other exception during the #DB processing is intercepted), the stack pointer from system management mode (SMM) may be saved into the virtual machine control block (VMCB offset 1D8h). There is no error if an interception does not occur during the #DB.

Potential Effect on System

Unpredictable program behavior when debug exceptions are enabled for a virtual machine, likely observed as a program or guest operating system crash. AMD has not observed this erratum with any commercially available software.

Suggested Workaround

None required.

Fix Planned

No

747 Spurious Debug Exception May Be Observed

Description

Under a highly specific and detailed set of internal timing requirements, the processor may report a debug exception (#DB) on an address that has a data write breakpoint while executing instructions that do not actually write to the address. The exception is misreported due to a branch misprediction that occurs for the monitored address.

Potential Effect on System

Debug engineers or debug programs may observe a #DB exception when no write occurs.

Suggested Workaround

None.

Fix Planned

No

770 Processor Incorrectly Restores Guest Privilege Level after Unintercepted I/O C-state Request

Description

Following an unintercepted guest access to an I/O address that causes an entry to a C-state, the processor may enter core C6 (CC6) state and incorrectly clear the guest current privilege level (CPL) to zero.

Potential Effect on System

Unpredictable system behavior. AMD has not observed this erratum with any commercially available software.

Suggested Workaround

Hypervisors should intercept any guest accesses to the I/O registers associated with I/O C-states, to avoid the possibility that a guest operating system has allowed a non-privileged application to request I/O C-states.

To intercept these accesses, hypervisors should program the virtual machine control block (VMCB) I/O interception bits as follows:

- VMCB offset 00Ch bit 27 (IOIO_PROT) should be 1b.
- The I/O protection map should indicate interception on a read to the eight consecutive I/O addresses starting with the address specified at C-state Base Address Register[CstateAddr] (MSRC001_1073[15:0]). The physical address of the I/O protection map is at VMCB offset 040h.

Fix Planned

No

780 Processor May Cache Guest Write Combining Memory Type

Description

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

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

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

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

Potential Effect on System

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

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

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

Suggested Workaround

A workaround is not recommended.

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

Fix Planned

No

784 Processor May Incorrectly Provide Control Register Data as the Result of a Load Operation

Description

Under a highly specific and detailed set of internal timing conditions, the processor core may provide control register data as the result of an instruction that is performing a load from memory. In order to observe this incorrect data, the processor must be at the highest privilege level (CPL 0) and be speculatively or non-speculatively executing certain invalid opcodes.

Potential Effect on System

Data corruption causing unpredictable system behavior.

Suggested Workaround

Contact your AMD representative for information on an update.

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 14h Models 00h-0Fh Processors*, order# 43170
- *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

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