[Public]

Revision Guide for AMD Family 19h Models 10h-1Fh Processors

Publication # **57095**	Revision: **1.01**
Issue Date: **May 2023**	

Advanced Micro Devices

Trademarks

List of Figures

List of Tables

Revision History

Date	Revision	Description
May 2023	1.01	Added #1446, #1448, #1452, #1454, #1455, #1458, #1460, #1462, #1463, #1464, #1465, #1467, #1469, #1475, #1478.
December 2022	1.00	Initial public release.

Overview

The purpose of the *Revision Guide for AMD Family 19h Models 10h-1Fh* is to communicate updated product information to designers of computer systems and software developers. This revision guide includes information on the following products:

- AMD EPYC™ 9004 Series Processors

Feature support varies by brands and OPNs (Ordering Part Number). To determine the features supported by your processor, contact your customer representative.

This guide consists of these major sections:

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

Revision Guide Policy

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

Conventions

Numbering

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

Arithmetic and Logical Operators

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

Table 1. Arithmetic and Logic Operators

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

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 *Processor Programming Reference (PPR) for AMD Family 19h Model 10h-1Fh Processors*, order# 55901.

Processor Identification

This section shows how to determine the processor revision.

Revision Determination

A processor revision is identified using a unique value that is returned in the EAX register after executing the CPUID instruction function 0000_0001h (CPUID Fn0000_0001_EAX). Figure 1 shows the format of the value from CPUID Fn0000_0001_EAX.

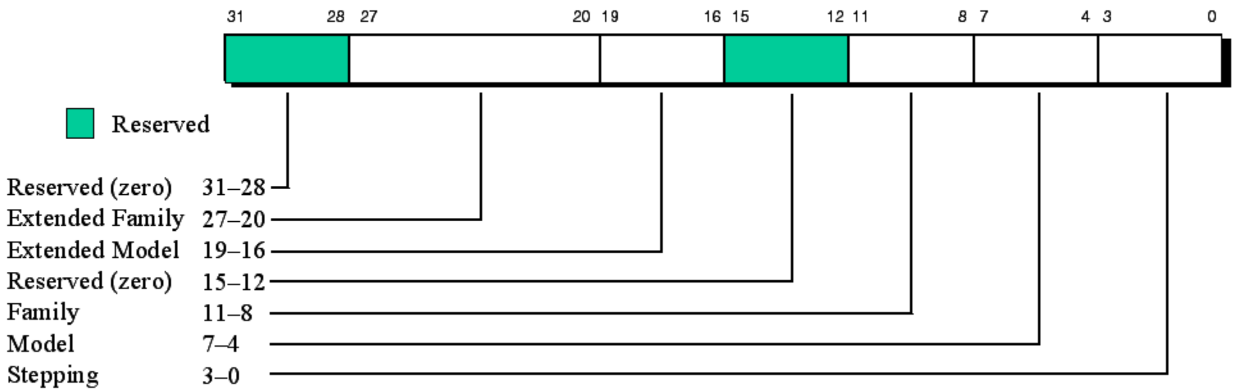

Figure 1. **Format of CPUID Fn0000_0001_EAX**

The following tables show the identification numbers from CPUID Fn0000_0001_EAX and SMUTHMx00000394 (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 19h Models 10h-1Fh SP5 Processor Revisions

CPUID Fn0000_0001_EAX	AMD EPYC™ 9004 Series Processors
00A10F11h (Genoa-B1)	X

Mixed Processor Revision Support

AMD Family 19h processors with different revisions may not be mixed in a multiprocessor system.

Programming and Displaying the Processor Name String

This section, intended for system software 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 system software 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 system software 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 PPR 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 PPR 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].

Operating System Visible Workarounds

This section describes how to identify operating system visible workarounds.

MSRC001_0140 OS Visible Work-around MSR0 (OSVW_ID_Length)

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

The reset default value of this register is 0000_0000_0000_0000h.

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

Table 3. OSVW ID Length Register

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 system software 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.

Table 4. OSVW Status Register

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

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

Table 5. Cross Reference of Product Revision to OSVW ID

CPUID Fn0000_0001_EAX (Mnemonic)	MSRC001_0141 Bits
00A10F11h (Genoa-B1)	0000_0000_0000_0000h

Product Errata

This table 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 6. Cross-Reference of Processor Revision to Errata

No.	Errata Description	CPUID
		00A10F11h (Genoa-B1)
1155	DMA or Peer-to-peer Accesses Using Guest Physical Addresses (GPAs) May Cause IOMMU Target Abort	X
1237	MCA_STATUS_LS[ExtErrorCode] May Contain Incorrect Code After a Store Queue Address Fatal Parity Error	X
1321	Processor May Generate Spurious #GP(0) Exception on WRMSR Instruction	X
1332	Processor May Fail to Generate a #UD Exception For Certain AVX-512 Instructions	X
1339	Processor May Fail to Generate a #UD Exception For EVEX Encoded Instructions	X
1344	Processor May Fail to Take #VMEXIT(NPF) When SNP-Active (Secure Nested Paging) Guest Writes an Improperly Configured Page	X
1358	TLB Way May Not Be Reported Correctly When Reporting L2DTLB Errors	X
1365	Fatal Error May Log Incorrect Location Information	X
1372	#GP May Occur On Returning to 32-Bit Mode After Five-Level Paging Was Enabled	X
1374	MCA::LS::MCA_SYND_LS[ErrorInformation] Bits [5:0] May Be Incorrect for STORE_DATA_OTHER Errors	X
1384	Certain MCA Extended Error Codes and MCA Control Bits Are Incorrect	X
1393	Guest in 32-Bit Mode With rIP Larger Than 32 Bits May Exhibit Unpredictable Behavior	X
1394	Processor May Cause Unexpected Collisions on SMBUS (System Management Bus)	X
1395	PCIe® Margining Lane Status Register Fields May Be Incorrectly Set to Zero	X
1401	PCIe® Controller May Erroneously Generate Correctable Receiver Error	X
1403	SKP Ordered Set May Be Dropped on x16 Lane CXL® Port With Sync Header Bypass Enabled	X
1416	Non-L3 Miss IBS (Instruction Based Sampling) Samples May Be Reported when IBS L3 Miss Fetch Filtering Enabled	X
1421	Speculative Fetch Activity May Be Underreported by IBS (Instruction Based Sampling)	X
1426	Poisoned TLP Egress Blocked Error May Fail to be Reported	X
1430	AHCI Controller May Incorrectly Assert or Incorrectly De-assert SATA Device Error Indicator	X
1431	CPU Core May Hang If SMT (Symmetric Multi Threading) Is Enabled and Bus Lock Occurs	X
1432	The L3CacheBwVicMon Bandwidth Event May Overcount	X
1441	Processor May Not Correctly Store All Data of DMA Write From Device to Memory	X
1442	Certain CXL® Link Capability Structure Registers Are Located at the Wrong Offset	X
1444	Advanced Platform Management Link (APML) May Cease to Function After Incomplete Read Transaction	X
1446	Incorrect Initialization of On-die 1.8V Regulator During Power Up May Cause Permanent Failure to Boot	X
1448	Processor May Delay PCIe® ACK DLLP Resulting in Correctable Errors Being Logged	X
1452	Non-Branch Entries May Erroneously Be Recorded In the LBR (Last Branch Record) Stack	X

Table 6. Cross-Reference of Processor Revision to Errata (continued)

No.	Errata Description	CPUID 00A10F11h (Genoa-B1)
1454	Processor May Log Unexpected LS MCE Error When Executing Virtualized VMLOAD or Virtualized VMSAVE	X
1455	PCID-Based INVLPGB May Fail to Flush Global Translations	X
1458	Processor May Fail to Accelerate Guest Access to Extended Interrupt Local Vector Table Registers	X
1460	IBS (Instruction Based Sampling) Sample Will Report Incorrect Misaligned Information for AVX-512 Loads	X
1462	System May Not Reboot or Shut Down Successfully After a Fatal Error	X
1463	Processor May Incorrectly Log Additional Watchdog Timeout Error After Fatal Error on GMI Link	X
1464	32-Byte Misaligned Supervisor Shadow Stack Pointer or Supervisor Shadow Stack in Non-WB Memory May Result in a Non-Restartable Guest	X
1465	SEV-SNP (Secure Encrypted Virtualization - Secure Nested Paging) Guests May Cause Core to Hang With Hypervisors That Do Not Intercept HLT Instruction	X
1467	Unexpected #PF for Hypervisor Write to 2M or 1G Page When SEV-SNP (Secure Encrypted Virtualization - Secure Nested Paging) Enabled	X
1469	Misaligned AVX-512 512-Bit Store to Top of Effective Address Space May Access Wrong Page	X
1475	Processor May Associate RMID and COS With an Incorrect Logical Processor	X
1478	Number of Unified Memory Controller Performance Counters Reported Incorrectly by CPUID_Fn80000022_EBX[21:16]	X

Cross-Reference of Errata to Package Type

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

Table 7. Cross-Reference of Errata to Package Type

Errata	Package
	SP5
1155	X
1237	X
1321	X
1332	X
1339	X
1344	X
1358	X
1365	X
1372	X
1374	X
1384	X
1393	X
1394	X
1395	X
1401	X
1403	X
1416	X
1421	X
1426	X
1430	X
1431	X
1432	X
1441	X
1442	X
1444	X
1446	X
1448	X
1452	X
1454	X
1455	X

Table 7. Cross-Reference of Errata to Package Type (continued)

Errata	Package
	SP5
1458	X
1460	X
1462	X
1463	X
1464	X
1465	X
1467	X
1469	X
1475	X
1478	X

Cross-Reference of Errata to Processor Segment

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

Table 8. Cross-Reference of Errata to Processor Segments

Errata	Processor Segment
	AMD EPYC™ 9004 Series Processors
1155	X
1237	X
1321	X
1332	X
1339	X
1344	X
1358	X
1365	X
1372	X
1374	X
1384	X
1393	X
1394	X
1395	X
1401	X
1403	X
1416	X
1421	X
1426	X
1430	X
1431	X
1432	X
1441	X
1442	X
1444	X
1446	X
1448	X
1452	X

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

Errata	Processor Segment
	AMD EPYC™ 9004 Series Processors
1454	X
1455	X
1458	X
1460	X
1462	X
1463	X
1464	X
1465	X
1467	X
1469	X
1475	X
1478	X

1155 DMA or Peer-to-peer Accesses Using Guest Physical Addresses (GPAs) May Cause IOMMU Target Abort

Description

In systems where:

- Virtualization is enabled, and
- IOMMU is in pass-through mode

DMA or peer-to-peer accesses using Guest Physical Addresses (GPAs) occurring within the regions defined below trigger a target abort.

- 0x00FD_0000_0000->0x00FD_F8FF_FFFF, or
- 0x00FD_F910_0000->0x00FD_F91F_FFFF, or
- 0x00FD_FB00_0000->0x00FD_FFFF_FFFF

Potential Effect on System

A DMA device will receive a target abort from the IOMMU.

Suggested Workaround

System software must mark the following block of memory as reserved:

- FD_0000_0000 -> FD_FFFF_FFFF

Fix Planned

No fix planned

1237 MCA_STATUS_LS[ExtErrorCode] May Contain Incorrect Code After a Store Queue Address Fatal Parity Error

Description

Under a highly specific and detailed set of internal timing conditions, an error that should be logged with MCA_STATUS_LS[ExtErrorCode]=11 (store queue address fatal parity error) may be incorrectly logged with MCA_STATUS_LS[ExtErrorCode]=23 (other store data fatal error).

Potential Effect on System

Diagnostic software may encounter incorrect information in MCA_STATUS_LS[ExtErrorCode] after a fatal error.

Suggested Workaround

None

Fix Planned

No fix planned

1321 Processor May Generate Spurious #GP(0) Exception on WRMSR Instruction

Description

The processor will generate a spurious #GP(0) exception on a WRMSR instruction if the following conditions are all met:

- The target of the WRMSR is the SYSCFG register.
- The write changes the value of Secure Nested Paging enable (SYSCFG.SNPEn) from 0 to 1.
- One of the threads that share the physical core has a non-zero value in the VM_HSAVE_PA MSR.

Potential Effect on System

Unexpected #GP(0) exception during processor boot.

Suggested Workaround

When enabling Secure Nested Paging, program VM_HSAVE_PA to 0h on both threads that share a physical core before setting SYSCFG.SNPEn.

Fix Planned

No fix planned

1332 Processor May Fail to Generate a #UD Exception For Certain AVX-512 Instructions

Description

The processor will fail to generate a #UD exception on the following AVX-512 mask instructions when VEX.vvvv = 0xxxb:

- KADDW kreg, kregv, kregm [VEX.L1.0F.W0 4A /r]
- KADDB kreg, kregv, kregm [VEX.L1.66.0F.W0 4A /r]
- KADDQ kreg, kregv, kregm [VEX.L1.0F.W1 4A /r]
- KADDD kreg, kregv, kregm [VEX.L1.66.0F.W1 4A /r]
- KANDW kreg, kregv, kregm [VEX.NDS.L1.0F.W0 41 /r]
- KANDB kreg, kregv, kregm [VEX.L1.66.0F.W0 41 /r]
- KANDQ kreg, kregv, kregm [VEX.L1.0F.W1 41 /r]
- KANDD kreg, kregv, kregm [VEX.L1.66.0F.W1 41 /r]
- KANDNW kreg, kregv, kregm [VEX.NDS.L1.0F.W0 42 /r]
- KANDNB kreg, kregv, kregm [VEX.L1.66.0F.W0 42 /r]
- KANDNQ kreg, kregv, kregm [VEX.L1.0F.W1 42 /r]
- KANDND kreg, kregv, kregm [VEX.L1.66.0F.W1 42 /r]
- KORW kreg, kregv, kregm [VEX.NDS.L1.0F.W0 45 /r]
- KORB kreg, kregv, kregm [VEX.L1.66.0F.W0 45 /r]
- KORQ kreg, kregv, kregm [VEX.L1.0F.W1 45 /r]
- KORD kreg, kregv, kregm [VEX.L1.66.0F.W1 45 /r]
- KUNPCKBW kreg, kregv, kregm [VEX.NDS.L1.66.0F.W0 4B /r]
- KUNPCKWD kreg, kregv, kregm [VEX.NDS.L1.0F.W0 4B /r]
- KUNPCKDQ kreg, kregv, kregm [VEX.NDS.L1.0F.W1 4B /r]
- KXNORW kreg, kregv, kregm [VEX.NDS.L1.0F.W0 46 /r]
- KXNORB kreg, kregv, kregm [VEX.L1.66.0F.W0 46 /r]
- KXNORQ kreg, kregv, kregm [VEX.L1.0F.W1 46 /r]
- KXNORD kreg, kregv, kregm [VEX.L1.66.0F.W1 46 /r]
- KXORW kreg, kregv, kregm [VEX.NDS.L1.0F.W0 47 /r]
- KXORB kreg, kregv, kregm [VEX.L1.66.0F.W0 47 /r]
- KXORQ kreg, kregv, kregm [VEX.L1.0F.W1 47 /r]
- KXORD kreg, kregv, kregm [VEX.L1.66.0F.W1 47 /r]

The processor will fail to generate a #UD exception on the following AVX-512 mask instructions when VEX.R = 1:

- KADDW kreg, kregv, kregm [VEX.L1.0F.W0 4A /r]
- KADDB kreg, kregv, kregm [VEX.L1.66.0F.W0 4A /r]
- KADDQ kreg, kregv, kregm [VEX.L1.0F.W1 4A /r]
- KADDD kreg, kregv, kregm [VEX.L1.66.0F.W1 4A /r]
- KANDW kreg, kregv, kregm [VEX.L1.0F.W0 41 /r]
- KANDB kreg, kregv, kregm [VEX.L1.66.0F.W0 41 /r]
- KANDQ kreg, kregv, kregm [VEX.L1.0F.W1 41 /r]
- KANDD kreg, kregv, kregm [VEX.L1.66.0F.W1 41 /r]
- KANDNW kreg, kregv, kregm [VEX.L1.0F.W0 42 /r]
- KANDNB kreg, kregv, kregm [VEX.L1.66.0F.W0 42 /r]
- KANDNQ kreg, kregv, kregm [VEX.L1.0F.W1 42 /r]
- KANDND kreg, kregv, kregm [VEX.L1.66.0F.W1 42 /r]

- KMOVW kreg, kregm [VEX.L0.0F.W0 90 /r]
- KMOVW kreg, m16 [VEX.L0.0F.W0 90 /r]
- KMOVB kreg, kregm [VEX.L0.66.0F.W0 90 /r]
- KMOVB kreg, m8 [VEX.L0.66.0F.W0 90 /r]
- KMOVQ kreg, kregm [VEX.L0.0F.W1 90 /r]
- KMOVQ kreg, m64 [VEX.L0.0F.W1 90 /r]
- KMOVD kreg, kregm [VEX.L0.66.0F.W1 90 /r]
- KMOVD kreg, m32 [VEX.L0.66.0F.W1 90 /r]
- KMOVW m16, kreg [VEX.L0.0F.W0 91 /r]
- KMOVB m8, kreg [VEX.L0.66.0F.W0 91 /r]
- KMOVQ m64, kreg [VEX.L0.0F.W1 91 /r]
- KMOVD m32, kreg [VEX.L0.66.0F.W1 91 /r]
- KMOVW kreg, r32 [VEX.L0.0F.W0 92 /r]
- KMOVB kreg, r32 [VEX.L0.66.0F.W0 92 /r]
- KMOVQ kreg, r64 [VEX.L0.F2.0F.W1 92 /r]
- KMOVD kreg, r32 [VEX.L0.F2.0F.W0 92 /r]
- KMOVW r32, kreg [VEX.L0.0F.W0 93 /r]
- KMOVB r32, kreg [VEX.L0.66.0F.W0 93 /r]
- KMOVQ r64, kreg [VEX.L0.F2.0F.W1 93 /r]
- KMOVD r32, kreg [VEX.L0.F2.0F.W0 93 /r]
- KNOTW kreg, kregm [VEX.L0.0F.W0 44 /r]
- KNOTB kreg, kregm [VEX.L0.66.0F.W0 44 /r]
- KNOTQ kreg, kregm [VEX.L0.0F.W1 44 /r]
- KNOTD kreg, kregm [VEX.L0.66.0F.W1 44 /r]
- KORW kreg, kregv, kregm [VEX.L1.0F.W0 45 /r]
- KORB kreg, kregv, kregm [VEX.L1.66.0F.W0 45 /r]
- KORQ kreg, kregv, kregm [VEX.L1.0F.W1 45 /r]
- KORD kreg, kregv, kregm [VEX.L1.66.0F.W1 45 /r]
- KORTESTW kreg, kregm [VEX.L0.0F.W0 98 /r]
- KORTESTB kreg, kregm [VEX.L0.66.0F.W0 98 /r]
- KORTESTQ kreg, kregm [VEX.L0.0F.W1 98 /r]
- KORTESTD kreg, kregm [VEX.L0.66.0F.W1 98 /r]
- KSHIFTLW kreg, kregm, imm8 [VEX.L0.66.0F3A.W1 32 /r]
- KSHIFTLB kreg, kregm, imm8 [VEX.L0.66.0F3A.W0 32 /r]
- KSHIFTLQ kreg, kregm, imm8 [VEX.L0.66.0F3A.W1 33 /r]
- KSHIFTLD kreg, kregm, imm8 [VEX.L0.66.0F3A.W0 33 /r]
- KSHIFTRW kreg, kregm, imm8 [VEX.L0.66.0F3A.W1 30 /r]
- KSHIFTRB kreg, kregm, imm8 [VEX.L0.66.0F3A.W0 30 /r]
- KSHIFTRQ kreg, kregm, imm8 [VEX.L0.66.0F3A.W1 31 /r]
- KSHIFTRD kreg, kregm, imm8 [VEX.L0.66.0F3A.W0 31 /r]
- KTESTW kreg, kregm [VEX.L0.0F.W0 99 /r]
- KTESTB kreg, kregm [VEX.L0.66.0F.W0 99 /r]
- KTESTQ kreg, kregm [VEX.L0.0F.W1 99 /r]
- KTESTD kreg, kregm [VEX.L0.66.0F.W1 99 /r]
- KUNPCKBW kreg, kregv, kregm [VEX.L1.66.0F.W0 4B /r]
- KUNPCKWD kreg, kregv, kregm [VEX.L1.0F.W0 4B /r]
- KUNPCKDQ kreg, kregv, kregm [VEX.L1.0F.W1 4B /r]
- KXNORW kreg, kregv, kregm [VEX.L1.0F.W0 46 /r]
- KXNORB kreg, kregv, kregm [VEX.L1.66.0F.W0 46 /r]
- KXNORQ kreg, kregv, kregm [VEX.L1.0F.W1 46 /r]

- KXNORD kreg, kregv, kregm [VEX.L1.66.0F.W1 46 /r]
- KXORW kreg, kregv, kregm [VEX.L1.0F.W0 47 /r]
- KXORB kreg, kregv, kregm [VEX.L1.66.0F.W0 47 /r]
- KXORQ kreg, kregv, kregm [VEX.L1.0F.W1 47 /r]
- KXORD kreg, kregv, kregm [VEX.L1.66.0F.W1 47 /r]

Potential Effect on System

None expected. Properly coded software will not encounter this issue.

Suggested Workaround

None

Fix Planned

No fix planned

1339 Processor May Fail to Generate a #UD Exception For EVEX Encoded Instructions

Description

The processor will fail to generate a #UD exception for EVEX encoded instructions under the following conditions:

EVEX.V' = 0 and the processor is not in 64-bit mode.

Or

EVEX.b = 1 and EVEX.b is not used by the instruction being executed.

Or

EVEX.aaa=000b and EVEX.z=1b when the following instructions are executed:

- VAESDEC
- VAESDECLAST
- VAESENC
- VAESENCLAST
- VCOMISD
- VCOMISS
- VCVTSD2SI
- VCVTSD2USI
- VCVTSI2SD
- VCVTSI2SS
- VCVTSS2SI
- VCVTSS2USI
- VCVTTSD2SI
- VCVTTSD2USI
- VCVTTSS2SI
- VCVTTSS2USI
- VCVTUSI2SD
- VCVTUSI2SS
- VEXTRACTPS
- VINSERTPS
- VMOVD
- VMOVHLPS
- VMOVHPD
- VMOVHPS
- VMOVLHPS
- VMOVLPD
- VMOVLPS
- VMOVNTDQA
- VMOVQ
- VPCLMULQDQ
- VPEXTRB
- VPEXTRD
- VPEXTRQ
- VPEXTRW
- VPINSRB

- VPINSRD
- VPINSRQ
- VPINSRW
- VPSADBW
- VPSLLDQ
- VPSRLDQ
- VUCOMISD
- VUCOMISS

Potential Effect on System

None expected. Properly coded software will not encounter this issue.

Suggested Workaround

None

Fix Planned

No fix planned

1344 Processor May Fail to Take #VMEXIT(NPF) When SNP-Active (Secure Nested Paging) Guest Writes an Improperly Configured Page

Description

The processor may fail to take a #VMEXIT(NPF) under the following conditions:

- An SNP-Active guest executes an instruction that performs a memory write or a masked write (VMASKMOV), and
- The page that is accessed by the memory operation is illegally configured to have VMPL (Virtual Machine Privilege Level) write permissions but not VMPL read permissions.

Potential Effect on System

None expected. Properly coded software will always enable VMPL read permissions if VMPL write permissions are enabled.

Suggested Workaround

None

Fix Planned

No fix planned

1358 TLB Way May Not Be Reported Correctly When Reporting L2DTLB Errors

Description

When MCA::LS::MCA_SYND_LS[ErrorInformation] is reporting L2DTLB errors, a 5-bit field (ErrorInformation[11:7]) is used to report which of 24 TLB ways has the error. When reporting L2DTLB errors, ErrorInformation[11] will always return the value 0b which will cause errors in ways 16 through 23 to alias to errors in ways 0 through 7.

Potential Effect on System

When reporting L2DTLB errors, the TLB Way modulo 16 is reported as the error location.

Suggested Workaround

None

Fix Planned

No fix planned

1365 Fatal Error May Log Incorrect Location Information

Description

Under a highly specific and detailed set of internal timing conditions, a fatal error logged with MCA::LS::MCA_STATUS_LS[ErrorCodeExt]=0xC:

- May log the incorrect location in MCA::LS::MCA_SYND_LS[ErrorInformation], or
- May be recorded for the other logical thread sharing the same core.

Potential Effect on System

The system may log errors erroneously.

Suggested Workaround

None

Fix Planned

No fix planned

1372 #GP May Occur On Returning to 32-Bit Mode After Five-Level Paging Was Enabled

Description

Under the following sequence of events, a #GP may occur:

- Five-Level paging is enabled.
- Software writes a 57-bit canonical (but not 48-bit canonical) virtual address into one of the following registers: LDTR_BASE, IDTR_BASE, GDTR_BASE, GS_BASE, FS_BASE.
- Software then switches to 32-bit mode with CR4.LA57 cleared.

Potential Effect on System

None expected. Properly coded software will not depend on the hardware to ignore the upper 32 bits in this situation.

Suggested Workaround

None

Fix Planned

No fix planned

1374 MCA::LS::MCA_SYND_LS[ErrorInformation] Bits [5:0] May Be Incorrect for STORE_DATA_OTHER Errors

Description

Under a highly specific and detailed set of internal timing conditions, MCA::LS::MCA_SYND_LS[ErrorInformation] bits [5:0] may be incorrect under the following conditions:

- An error is logged with MCA::LS::MCA_STATUS_LS[ErrorCodeExt]=0x17 occurs in the presence of AVX-512 masked stores.
- MCA::LS::MCA_SYND_LS[ErrorInformation] bit [6] is equal to 1b.

Potential Effect on System

The system may log the incorrect location for an error.

Suggested Workaround

None

Fix Planned

No fix planned

1384 Certain MCA Extended Error Codes and MCA Control Bits Are Incorrect

Description

MCA::CS::MCA_STATUS_CS[ErrorCodeExt] Extended Error Codes are incorrect for the following error types:

- For a CNTR_UNFL error, ErrorCodeExt = 11 instead of the correct value of 13
- For a CNTR_OVFL error, ErrorCodeExt = 10 instead of the correct value of 12
- For a SDP_UNEXP_RETRY error, ErrorCodeExt = 9 instead of the correct value of 11
- For a SPF_ECC_ERR error, ErrorCodeExt = 13 instead of the correct value of 10
- For a SPF_PRT_ERR error, ErrorCodeExt = 12 instead of the correct value of 9

MCA::CS::MCA_CTL_CS operates incorrectly for the following error types:

- A CNTR_UNFL error is enabled for reporting by bit 11 instead of the correct bit 13
- A CNTR_OVFL error is enabled for reporting by bit 10 instead of the correct bit 12
- A SDP_UNEXP_RETRY error is enabled for reporting by bit 9 instead of the correct bit 11
- A SPF_ECC_ERR error is enabled for reporting by bit 13 instead of the correct bit 10
- A SPF_PRT_ERR error is enabled for reporting by bit 12 instead of the correct bit 9

Potential Effect on System

None

Suggested Workaround

Software reading MCA::CS::MCA_STATUS_CS[ErrorCodeExt] or accessing MCA::CS::MCA_CTL_CS should refer to the description of this erratum.

Fix Planned

No fix planned

1393 Guest in 32-Bit Mode With rIP Larger Than 32 Bits May Exhibit Unpredictable Behavior

Description

When a VMRUN command launches a guest operating in 32-bit mode with an rIP (Instruction Pointer) larger than 32 bits saved in its VMCB (Virtual Machine Control Block), that guest and only that guest may exhibit unpredictable behavior.

Potential Effect on System

None expected. A 32-bit guest is not expected to have an rIP larger than 32 bits saved in its VMCB.

Suggested Workaround

None

Fix Planned

No fix planned

1394 Processor May Cause Unexpected Collisions on SMBUS (System Management Bus)

Description

The processor may violate the Data Setup Time parameter of the SMBUS specification which may cause unexpected collisions to be observed on the SMBUS. The processor may require up to 500ns of Data Setup Time on SMBUS.

Potential Effect on System

Unexpected collisions may be observed on the SMBUS if Data Setup Time is less than 500ns. In rare cases, these collisions may prevent reading the DIMM SPD ROMs correctly.

Suggested Workaround

None

Fix Planned

No fix planned

1395 PCIe® Margining Lane Status Register Fields May Be Incorrectly Set to Zero

Description

Under the following conditions, the Margining Lane Status Register fields (Margin Type, Margin Receiver Number, and Margin Payload) may be incorrectly set to 0 for one clock cycle:

- The host controller is 16 lanes, and
- The host controller is running in 2 Symbols Per Clock mode, and
- The link is operating in Gen5 or Gen4 mode, and
- The host controller is margining the retimer on the motherboard, and
- The host controller receives a Control SKP Ordered Set (CSKP).

If the retimer on the motherboard sends bad receiver number in the CSKP, then the value in the Margining Lane Status Register will remain zero until the next CSKP.

Potential Effect on System

None expected. If the lane margining software is not designed to ignore invalid (all zero) values in the Margining Lane Status Register, it may return invalid results.

Suggested Workaround

Margining software should ignore zeroes when reading the Margining Lane Status Register and reread the register until valid (non-zero) data is detected.

Fix Planned

No fix planned

1401 PCIe® Controller May Erroneously Generate Correctable Receiver Error

Description

The PCIe® controller may erroneously generate a false token error (correctable receiver error) under the following conditions:

- PCIe link is operating in x16 2SPC (Symbols Per Clock) or x8 4SPC or x8 2SPC or x4 4SPC, and
- There is a nullified TLP, and
- There is a SKP ordered set immediately following the nullified TLP.

Potential Effect on System

Unexpected entry into PCIe link recovery state and logging a spurious correctable receiver error

Suggested Workaround

None recommended

Fix Planned

No fix planned

1403 SKP Ordered Set May Be Dropped on x16 Lane CXL® Port With Sync Header Bypass Enabled

Description

A x16 CXL® port operating with Sync Header Bypass enabled (CXL_Sync_Hdr_Bypass_Enable programmed to 1b) may drop a SKP ordered set.

Potential Effect on System

Unpredictable CXL link behavior with devices that do not tolerate dropped SKP ordered set.

Suggested Workaround

Do not override the default state by programming CXL_Sync_Hdr_Bypass_Enable to 1b on a x16 lane CXL port.

Fix Planned

No fix planned

1416 Non-L3 Miss IBS (Instruction Based Sampling) Samples May Be Reported when IBS L3 Miss Fetch Filtering Is Enabled

Description

Under a highly specific and detailed set of internal timing conditions, IBS samples that are not an L3 miss (Core::X86::Msr::IBS_FETCH_CTL[IbsFetchL3Miss]=0) may erroneously be reported even though L3 miss IBS fetch filtering is enabled (Core::X86::Msr::IBS_FETCH_CTL[IbsL3MissOnly]=1).

Potential Effect on System

None expected.

Suggested Workaround

None.

Software can use Core::X86::Msr::IBS_FETCH_CTL[IbsFetchL3Miss] to determine if the sample is an L3 miss.

Fix Planned

No fix planned

1421 Speculative Fetch Activity May Be Underreported by IBS (Instruction Based Sampling)

Description

When Fetch IBS is enabled, the processor fails to send interrupts for sampled fetches that were aborted before being delivered to the decoder. Instead, the processor tags a new fetch. If the new tagged fetch gets sent to the decoder, then an interrupt is asserted.

If the new tagged fetch is aborted, then another new fetch is tagged. It is expected that eventually a fetch will be tagged and sent to the decoder, which will cause an IBS interrupt to be asserted.

Potential Effect on System

Reduced IBS visibility into speculative fetch activity

Suggested Workaround

None

Fix Planned

No fix planned

1426 Poisoned TLP Egress Blocked Error May Fail to be Reported

Description

The Poisoned TLP Egress Blocked error will fail to be reported when PCIERCCFG::PCIE_DPC_CNTL[POISONED_TLP_EGRESS_BLOCKING_ENABLE] is programmed to 1b, and a poisoned egress completion with a data payload size of 16B or less is dropped.

Potential Effect on System

Any action that was programmed to take place when a Poisoned TLP Egress Blocked error should have been reported will not happen.

Suggested Workaround

None

Fix Planned

No fix planned

1430 AHCI Controller May Incorrectly Assert or Incorrectly De-assert SATA Device Error Indicator

Description

The last data bit from an SGPIO transfer to a SATA device may be overridden by a subsequent SGPIO transfer when SGPIO is enabled for multiple SATA instances on the AHCI controller. The overridden data bit will be associated with the error indicator for the eighth device (device number 7) on a SATA controller.

Potential Effect on System

The error indicator for SATA device number 7 on a particular SATA controller may be incorrectly asserted or incorrectly de-asserted.

Suggested Workaround

The issue can be avoided by one of the following methods:

- Connecting all SATA devices using SGPIO to a single SATA controller.
- Depopulating the highest-numbered SATA device (device 7) for each SATA controller supported on the SGPIO link.

Or, for systems that have a CPLD distributor for SGPIO linked SATA devices, one of the following methods:

- Programming the motherboard CPLD (Complex Programmable Logic Device) to use the 24th bit value in the 4th bit stream of the SGPIO transfer to overwrite the 24th bit value in the fifth bit stream of the SGPIO transfers.
- Programming the motherboard CPLD (Complex Programmable Logic Device) to ignore the data from the 5th bit stream of the SGPIO transfer.

Fix Planned

No fix planned

1431 CPU Core May Hang If SMT (Symmetric Multi Threading) Is Enabled and Bus Lock Occurs

Description

Under a highly specific and detailed set of internal timing conditions, if SMT (Symmetric Multi Threading) is enabled and a bus lock occurs, the system may hang or experience a watchdog timeout on a core.

Potential Effect on System

System may hang or reset.

Suggested Workaround

System software may contain the workaround for this erratum.

Fix Planned

No fix planned

1432 The L3CacheBwVicMon Bandwidth Event May Overcount

Description

The L3CacheBwVicMon Bandwidth Event as defined in the AMD64 Technology Platform Quality of Service Extensions document (PID # 56375) may erroneously count up to twice the amount of victim data actually generated from the QOS (Quality of Service) Domain.

Potential Effect on System

Software monitoring the L3CacheBwVicMon Bandwidth Source may experience inaccuracies. QOS Bandwidth Enforcement may control the bandwidth of a class of service to a lower actual bandwidth than the configured limit.

Suggested Workaround

Program QOS_EVT_CFG_0[6] to 0b and program QOS_EVT_CFG_1[6] to 0b (refer to AMD64 Technology Platform Quality of Service Extensions document) to avoid monitoring the L3CacheBwVicMon Bandwidth Source.

Programming QOS_EVT_CFG_0[6] to 0b and QOS_EVT_CFG_1[6] to 0b will not change the QOS Bandwidth Enforcement behavior.

Fix Planned

No fix planned

1441 Processor May Not Correctly Store All Data of DMA Write From Device to Memory

Description

The processor may not correctly store all of the data of a DMA write from the device to memory.

Potential Effect on System

Data corruption

Suggested Workaround

System software may contain the workaround for this erratum.

Fix Planned

No fix planned

1442 Certain CXL® Link Capability Structure Registers Are Located at the Wrong Offset

Description

The processor does not implement the high 32 bits of the following registers in the CXL® link capability structure. These three registers are defined as 64 bits in the CXL 1.1 specification with the upper 32 bits being read-only all zero.

- 0x0C High 32 bits of CXL Link Ctrl & Status Register
- 0x2C High 32 bits of CXL Link Ack Timer Ctrl Register
- 0x34 High 32 bits of CXL Link Defeature

The other registers in the CXL link capability structure filled gaps left by the unimplemented register bits, leaving the CXL capability registers as follows:

- 0x00 CXL Link Layer Capability Register (Low)
- 0x04 CXL Link Layer Capability Register (High)
- 0x08 CXL Link Control and Status Register (Low)
- 0x0C CXL Link Rx Credit Control Register (Low)
- 0x10 CXL Link Rx Credit Control Register (High)
- 0x14 CXL Link Rx Credit Return Status Register (Low)
- 0x18 CXL Link Rx Credit Return Status Register (High)
- 0x1C CXL Link Tx Credit Control Register (Low)
- 0x20 CXL Link Tx Credit Control Register (High)
- 0x24 CXL Link Ack Timer Control Register (Low)
- 0x28 CXL Link Defeature (Low)

Potential Effect on System

None expected. The CXL link capability registers are expected to be accessed only by AMD firmware and are not expected to be accessed by software. Software or firmware using the offsets specified in the CXL specification to access the registers listed in the description of this erratum may observe incorrect results.

Suggested Workaround

Software accessing the registers listed in the description of this erratum should refer to the description of this erratum and use the implemented offsets rather than the offsets specified in the CXL spec. AMD firmware that accesses these registers has implemented this workaround.

Software that accesses CXL Link Control and Status Register should only perform a 32-bit access at offset 0x08.

Software that accesses CXL Link Ack Timer Control Register should only perform a 32-bit access at offset 0x24.

Software that accesses CXL Link Defeature Register should only perform a 32-bit access at offset 0x28.

Software that accesses the following registers should subtract 4 from the CXL defined offset to determine the correct offset.

- CXL Link Rx Credit Control Register (Low) use offset 0x10 - 0x4 = 0xC
- CXL Link Rx Credit Control Register (High) use offset 0x14 - 0x4 = 0x10
- CXL Link Rx Credit Return Status Register (Low) use offset 0x18 - 0x4 = 0x14
- CXL Link Rx Credit Return Status Register (High) use offset 0x1C - 0x4 = 0x18
- CXL Link Tx Credit Control Register (Low) use offset 0x20 - 0x4 = 0x1C
- CXL Link Tx Credit Control Register (High) use offset 0x24 - 0x4 = 0x20

Fix Planned

No fix planned

1444 Advanced Platform Management Link (APML) May Cease to Function After Incomplete Read Transaction

Description

The APML responder will incorrectly enter a state where it will issue a NAK to all subsequent transactions if it receives the first phase of an APML read but not the second phase of the APML read. The BMC (Baseboard Management Controller) will not issue the second phase of a read transaction if it is reset by an asynchronous event such as a watchdog timeout or a firmware update.

Potential Effect on System

APML may cease to function.

Suggested Workaround

System software may contain the workaround for this erratum.

When system software has been upgraded to a version that implements the workaround the BMC can recover from the error by using the following procedure. The BMC should execute the error detection and recovery flow as specified in the following section of the Processor Programming Preference as part of the BMC firmware boot flow or after three consecutive retries due to a NAK:

- Chapter: Advanced Platform Management Link (APML)
- Section: SBI Protocols
- Sub-Section: SBI Error Detection and Recovery

Fix Planned

No fix planned

1446 Incorrect Initialization of On-die 1.8V Regulator During Power Up May Cause Permanent Failure to Boot

Description

Under a highly specific and detailed set of internal timing conditions, an on-die 1.8V voltage regulator may initialize incorrectly during a power up sequence resulting in an over-voltage condition. If this over-voltage condition occurs, the processor may be rendered permanently unable to boot.

Potential Effect on System

Processor may become permanently inoperable.

Suggested Workaround

If failure occurs, contact your AMD representative for remediation.

For systems where this failure has not yet occurred, system software may contain a mitigation for this erratum.

Fix Planned

No fix planned

1448 Processor May Delay PCIe® ACK DLLP Resulting in Correctable Errors Being Logged

Description

The processor may delay scheduling a PCIe® ACK DLLP beyond the Maximum ACK Latency Limits in the PCIe specification. A GEN 3 or earlier link partner that is not tolerant of this delay may replay a transaction in response to this delay.

Potential Effect on System

Spurious replay transactions resulting in replay correctable errors being logged.

Suggested Workaround

None

Fix Planned

No fix planned

1452 Non-Branch Entries May Erroneously Be Recorded In the LBR (Last Branch Record) Stack

Description

Non-Branch entries may erroneously be recorded in the LBR (Last Branch Record) Stack. These spurious entries will have the following properties:

- LastBranchStackToIp[Valid] (bit 63) = 1
- LastBranchStackToIp[Speculative] (bit 62) = 1
- LastBranchStackToIp[Reserved] (bit 61) = 1

Potential Effect on System

Valid LBR Stack entries may be overwritten by the spurious entries. The spurious entries appear to be valid but do not contain valid branch data.

Suggested Workaround

Software should ignore LBR Stack entries that have LastBranchStackToIp[Reserved] (bit 61) = 1.

Fix Planned

No fix planned

1454 Processor May Log Unexpected LS MCE Error When Executing Virtualized VMLOAD or Virtualized VMSAVE

Description

Under a highly specific and detailed set of internal timing conditions, when a virtualized VMLOAD or virtualized VSAVE instruction is executed the processor may log an uncorrectable LS MCE error of the type "HWA" with:

- MCA::LS::MCA_STATUS_LS[ErrorCodeExt] = 0x16 and
- MCA::LS::MCA_SYND_LS[ErrorInformation] = 0x00094.

Potential Effect on System

System may log a fatal error and reset.

Suggested Workaround

System software may contain the workaround for this erratum.

Fix Planned

No fix planned

1455 PCID-Based INVLPGB May Fail to Flush Global Translations

Description

An INVLPGB instruction will fail to flush global translations on a target processor under the following conditions:

- The rAX operand has both bit 1 (valid PCID) and bit 3 (include Global) set, and
- The target processor is not currently running with the PCID set to the value encoded in EDX[27:16].

Potential Effect on System

None expected. Software is not expected to use INVLPGB with PCID-based flushing.

Suggested Workaround

None

Fix Planned

No fix planned

1458 Processor May Fail to Accelerate Guest Access to Extended Interrupt Local Vector Table Registers

Description

With AVIC (Advanced Virtual Interrupt Controller) enabled, processors that set CPUID_Fn8000_00A_EDC[27]=1 will:

- Generate a #VMEXIT(fault) instead of allowing guest reads of the Extended Interrupt Local Vector Table Registers.
- Generate a #VMEXIT(fault) instead of generating a #VMEXIT(trap) on guest write access to the Extended Interrupt Local Vector Table Registers.

Potential Effect on System

Unexpected #VMEXIT(fault) on accesses to the Extended Interrupt Local Vector Table Registers.

Suggested Workaround

System software may contain the workaround for this erratum.

If the workaround mentioned above is not available, program MSRC001_10E6[32] to 0b.

Fix Planned

No fix planned

1460 IBS (Instruction Based Sampling) Sample Will Report Incorrect Misaligned Information for AVX-512 Loads

Description

When sampled by IBS (Instruction Based Sampling), AVX-512 loads always set Core::X86::Msr::IBS_OP_DATA3[IbsDcMisAcc] = 1 regardless of whether the memory access is misaligned or not.

Potential Effect on System

Inaccuracies in performance monitoring software may be experienced in the presence of AVX-512 instructions.

Suggested Workaround

Software can use Core::X86::Msr::IBS_DC_LINADDR and IBS_OP_DATA3[IbsOpMemWidth] to determine if the memory access is 64B (i.e. cacheline) aligned.

Fix Planned

No fix planned

1462 System May Not Reboot or Shut Down Successfully After a Fatal Error

Description

Under a highly specific and detailed set of internal timing conditions, if the processor encounters a fatal error condition, system may fail to respond to a reboot or a shut down request including

- a 4-second-long PWR_BTN_L shutdown,
- toggling KBRST_L and
- toggling SYS_RESET_L.

Potential Effect on System

In the highly unlikely event that this occurs, system may fail to respond to a reboot or a shut down request after a fatal error condition.

Suggested Workaround

System software may contain the workaround for this erratum. For systems with ADDC (Autonomous Debug Data Collection):

- AMD system software will detect the error condition and
  - set SBRMI.RasStatus.bit[1] = 1 and
  - send APML_ALERT to BMC.
- BMC should then log the error and provide the option to auto-recover by sending an additional SYS_RESET_L toggle.

For systems without ADDC, update the board power sequencing FPGA to detect a 4 second PWR_BTN_L shutdown hang event in order to trigger an additional SYS_RESET_L toggle and issue another 4 second PWR_BTN_L shutdown.

Fix Planned

No fix planned

1463 Processor May Incorrectly Log Additional Watchdog Timeout Error After Fatal Error on GMI Link

Description

When there is a fatal error on the GMI link the processor may incorrectly log an additional fatal Watchdog Timeout error. Logging the additional fatal Watchdog Timeout error after a GMI fatal error is more likely to occur if ADDC (Autonomous Debug Data Collection) is enabled.

Potential Effect on System

The processor may log a watchdog timeout error that is unrelated to the cause of GMI link error.

Suggested Workaround

None

Fix Planned

No fix planned

1464 32-Byte Misaligned Supervisor Shadow Stack Pointer or Supervisor Shadow Stack in Non-WB Memory May Result in a Non-Restartable Guest

Description

Under the either of following conditions, the processor may VMEXIT a guest with the Busy Bit set in the Supervisor Shadow Stack Token while not having completed the far transfer in the guest:

- The 32 bytes of data accessed at the new supervisor shadow stack pointer are not entirely contained within one page, or
- The memory for the new supervisor shadow stack is not mapped as WB Memory.

Potential Effect on System

None expected.

Software is not expected to create either of the conditions listed above. If the conditions occurred, the guest would not be able to be restarted.

Suggested Workaround

None

Fix Planned

No fix planned

1465 SEV-SNP (Secure Encrypted Virtualization - Secure Nested Paging) Guests May Cause Core to Hang With Hypervisors That Do Not Intercept HLT Instruction

Description

A core will hang under the following conditions:

- SEV-SNP Guest has SMT Protection for Guests enabled, and
- The hypervisor does not intercept the HLT instruction, and
- The guest executes a HLT instruction.

Potential Effect on System

None expected. Hypervisors are expected to intercept HLT when running SEV-SNP Guests.

Suggested Workaround

None

Fix Planned

No fix planned

1467 Unexpected #PF for Hypervisor Write to 2M or 1G Page When SEV-SNP (Secure Encrypted Virtualization - Secure Nested Paging) Enabled

Description

A hypervisor may take an unexpected #PF Exception under the following conditions:

- SEV-SNP (Secure Encrypted Virtualization - Secure Nested Paging) is enabled.
- The hypervisor writes to a 2M or larger page, and
- An SVM data structure (VMCB, VMSA, or AVIC backing page) associated with a currently running guest is allocated starting at a 2M boundary within the 2M or larger page used for the access.

Potential Effect on System

Unexpected #PF Exception

Suggested Workaround

Software should:

- Use a 4K page size for memory access within a region where an SVM page structure is allocated on a 2M-aligned boundary, or
- Avoid allocating SVM pages on 2M-aligned boundaries.

Fix Planned

No fix planned

1469 Misaligned AVX-512 512-Bit Store to Top of Effective Address Space May Access Wrong Page

Description

Under the following conditions an AVX-512 512-Bit store instruction may perform part of the operation to the wrong address:

- In 64-bit mode, the address size of the operation is 32 bits, and the targeted memory location is misaligned across the 4 GB address boundary, or
- In Protected Mode, the address size is 32 bits, and the targeted effective memory address is 0xFFFFFFE0, or
- In Protected Mode, the address size is 16 bits, and the targeted memory location is misaligned across the 64K boundary.

The incorrect portion of the memory operation will be to effective address zero and is subject to normal protection checks for that address.

Potential Effect on System

Legacy mode or compatibility mode software using 32-bit or 16-bit addressing mode may encounter an unexpected memory protection violation or may erroneously read from or write to effective address zero.

Suggested Workaround

None

Fix Planned

No fix planned

1475 Processor May Associate RMID and COS With an Incorrect Logical Processor

Description

The AMD64 Technology Platform Quality of Service Extension associates an RMID (Resource Monitoring ID) and/or a COS (Class Of Service) to a logical processor through the logical processors' write to the PQR_ASSOC (MSR C8Fh). Only products with less than 8 physical cores in a complex (cores sharing an L3) are affected and may associate the RMID and COS written into PQR_ASSOC with an incorrect logical processor in the same complex.

Potential Effect on System

AMD64 Technology Platform Quality of Service Resource monitoring or enforcement may be applied to the incorrect logical processor.

Suggested Workaround

System software may contain the workaround for this erratum.

Fix Planned

No fix planned

1478 Number of Unified Memory Controller Performance Counters Reported Incorrectly by CPUID_Fn80000022_EBX[21:16]

Description

The number of UMC (Unified Memory Controller) Performance Counters reported by CPUID_Fn80000022_EBX[21:16] is incorrect. The actual number of available UMC Performance Counters is the number reported by CPUID_Fn80000022_EBX[21:16] times 2.

Potential Effect on System

System software may not be able to access all available UMC Performance counters.

Suggested Workaround

Software should multiply the number reported in CPUID_Fn80000022_EBX[21:16] by 2.

Fix Planned

Yes

Documentation Support

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

- *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 I/O Virtualization Technology (IOMMU) Specification*, order # 48882
- *Processor Programming Reference (PPR) for AMD Family 19h Models 10h-1Fh Processors*, order # 55901
- *AMD64 Technology Platform Quality of Service Extensions*, order # 56375

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