



AMD A55/A60M/A68/A68M/A70M/ A75/A85X Errata

Publication #	48671	Revision:	3.00
Issue Date:	July 2012		

© 2012 Advanced Micro Devices, Inc. All rights reserved.

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. AMD assumes no liability whatsoever, and disclaims any express or implied warranty, relating to this document including, but not limited to, the implied warranty of merchantability, fitness for a particular purpose, or infringement of any intellectual property right. AMD shall not be liable for any damage, loss, expense, or claim of loss of any kind or character (including without limitation direct, indirect, consequential, exemplary, punitive, special, incidental or reliance damages) arising from use of or reliance on this document. No license, whether express, implied, arising by estoppel, or otherwise, to any intellectual property rights are granted by this publication. Except for AMD product purchased pursuant to AMD's Standard Terms and Conditions of Sale, and then only as expressly set forth therein, 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 and combinations thereof are trademarks of Advanced Micro Devices, Inc.

Microsoft and Windows are registered trademarks of Microsoft Corporation.

PCIe is a registered trademark of PCI-SIG.

Linux is a registered trademark of Linus Torvalds.

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
July 2012	3.00	Initial public release

Introduction

This document provides details of the silicon errata for the following products:

FCH Product	Codename	Family
A60M	Hudson-M2	Hudson-2
A55	Hudson-D2	Hudson-2
A68M	Hudson-M3L	Hudson-3
A70M	Hudson-M3	Hudson-3
A68	Hudson-D3L	Hudson-3
A75	Hudson-D3	Hudson-3
A85X	Hudson-D4	Hudson-3

Product Errata Summary

A unique errata reference number (ERN) has been assigned to each erratum within this document for user convenience in tracking the errata within specific revision levels. Table 1 cross-references the revisions of the part to each erratum. An “X” indicates that the erratum applies to the revision. The absence of an “X” indicates that the erratum does not apply to the revision. An “*” indicates advance information that the erratum has been fixed but not yet verified. “No fix planned” indicates that no fix is planned for current or future revisions of the ASIC.

Table 1: Cross-Reference of Product Revision to Errata

#	Errata Description	ASIC Revision	
		A13	A14
2	Indeterminate Boot Up State of RTC Bank Selection Bit (DV0)	No Fix Planned	
21	Error in USB Frame List Processing	No Fix Planned	
24	Efuse Data May Fail to Load During G3 to S5 Power-up Transitions	No Fix Planned	
25	Interrupt is Not Serviced in PIC Mode While the IMC is Accessing ACPI Registers	No Fix Planned	
27	SmiCmdStatus Decoding Failure	No Fix Planned	
29	Incorrect Initialization of XHCI DMA Logic (Hudson-3 only)	No Fix Planned	
30	Incorrect BANDGAP_ADJUSTMENT Programming	No Fix Planned	
32	SD Controller INT_DISABLE Disables MSI	No Fix Planned	
33	Incorrect Implementation of the IOAPIC Delivery Status Bit	No Fix Planned	
34	Incorrect ASPM Transitions on the GPP Interface	No Fix Planned	
35	Non-compliance of the S field of the USB Start-Split Transaction Token	No Fix Planned	
37	XHCI State Machine Locks Up When Swapping USB 3.0 Devices During S3 (Hudson-3 only)	No Fix Planned	
38	XHCI Controller U1 Exit Time Does Not Meet LFPS Timing Requirements (Hudson-3 only)	No Fix Planned	
39	ACPI PM Timer Read Returns PMIO Register Data	X	
41	SATA PxIS.DPS Status May Not Set on PRD Completion	No Fix Planned	
42	XHCI Controller May Be in Wake State While System is in S3 State	No Fix Planned	
43	A20M Port Access May Cause Incorrect Interrupt Acknowledge	No Fix Planned	
46	LPC SYNC Timeout Violation	No Fix Planned	

Product Errata

2 Indeterminate Boot Up State of RTC Bank Selection Bit (DV0)

Description

The RTC Bank Selection (DV0) bit (RTC_Reg:0A[4]) is not guaranteed to be initialized to the default value (DV0 = 0) by hardware on power cycles involving a VBAT power ramp. This will result in the software reading data from RTC memory bank 1 instead of bank 0. Of the RTC registers, only the DV0 bit is expected by software to be in the default state (bank 0 selected) on power up.

Potential Effect on System

Unexpected system POST behavior may occur if the DV0 bit comes up in a non-default state (i.e., DV0 = 1). The failure will occur only if the platform BIOS is using the standard bank-dependent indexed register method for accessing the RTC memory through the use of I/O port registers 70h and 71h. A platform BIOS using the AMD proprietary bank-independent indexed register method for accessing the RTC memory (using I/O port registers 72h and 73h) will not be impacted. Once the failure is observed (i.e., DV0 = 1 on power up), the failure will be persistent through warm and cold boot resets. Conversely, if the DV0 bit comes up in the proper default state, it is unaffected by cold or warm resets.

Suggested Workaround

Set DV0=0 during early Southbridge initialization and ahead of any access to RTC RAM. This change is implemented in CIMx version 0.0.6.3.

Fix Planned

No

21 Error in USB Frame List Processing

Description

An error in processing the USB frame list may result in the USB controller reading from memory address 0 and, if the contents in that location correspond to a valid Transfer Descriptor, writing to a memory address location outside of the EHCI controller's memory space. The necessary conditions to expose this issue are systems that encounter longer than normal latency (i.e., exceeding 125 μ s) on DMA reads and software that implements a USB Periodic Frame List Structure where the Frame List Link Queue Head pointer is a null pointer with T-bit = 1.

Linux®, VMware and any custom (in house) BIOS implementations that implement this particular USB Periodic Frame List Structure are exposed. There is no exposure to Microsoft® Windows® operating systems.

Potential Effect on System

The manifestation of this problem is dependent on the function of the memory address mistakenly accessed by the USB controller. Observed cases lead to system hangs due to a corrupted interrupt vector.

Suggested Workaround

USB Periodic Frame List Structures must be generated where the periodic frame list element pointer sets T-bit = 0 and uses a pointer to a valid but inactive queue head. This change is included in USB driver updates to VMware (ESX 4.1P3 and ESX 5.0) and Linux (version 2.6.37 kernel).

Fix Planned

No

24 Efuse Data May Fail to Load During G3 to S5 Power-up Transitions**Description**

Efuse configuration data (accessed via PM_Reg: D9h) may be incorrect on the first de-assertion of RSM_RST# during G3 to S5 power-up transitions. On every subsequent de-assertion of RSM_RST#, the eFuse contents are loaded correctly. During POST, the system BIOS reads this register to determine the supported FCH features.

Potential Effect on System

Unpredictable system behavior may occur due to improper feature initialization.

Suggested Workaround

Use CIMx revision 0.0.9.0 and above which will calculate the checksum of the contents of this register and verify it against the saved on-chip checksum. If the checksum verification fails, software will force a reload of the Efuse configuration bits.

Fix Planned

No

25 Interrupt is Not Serviced in PIC Mode While the IMC is Accessing ACPI Registers

Description

The FCH may fail to service a pending interrupt request when the IMC is simultaneously accessing ACPI-related FCH registers while the system is in PIC interrupt mode.

Potential Effect on System

General system instability has been observed, including a system hang before booting the operating system. For platforms supporting IMC-based fan control, the fan may intermittently get stuck at maximum speed.

Suggested Workaround

Use IMC firmware version 1.05 or later and modify the platform BIOS to ensure that the IMC does not access any FCH registers while in PIC mode. Enabling of fan control, for example, must be done only after the operating system has started to load and is guaranteed to be operating in APIC mode. This can be done in the AcpiEnable BIOS function which will be called by ACPI-capable operating systems.

With this firmware update and BIOS change, IMC-based POST code displays are not supported on the platform.

Fix Planned

No

27 SmiCmdStatus Decoding Failure**Description**

The SMI Command Port Status register (SmiCmdStatus [SmiCmdBlk: 01h]) is not decoded properly when the IMC is simultaneously accessing ACPI-related register blocks. As a result, write cycles to SmiCmdStatus (located at SMI Command Port Base + 1 byte) will complete, but the register is not updated. Similarly, reads to SmiCmdStatus may not return the actual register contents. On IMC-enabled platforms where SmiCmdStatus is used, the improper decoding of this register does not affect SMI generation or IMC reads or writes.

Potential Effect on System

Some platform BIOS code bases use SmiCmdStatus as a status register during SMI processing. For these platforms, failure symptoms may vary depending on the specific usage of this register. The observed failure was a system soft hang during POST after resuming from the S4 state.

Suggested Workaround

For IMC-enabled platforms that make use of SmiCmdStatus, read or write to the status register using 16-bit accesses to SmiCmdPort [SmiCmdBlk: 00h]. To ensure that SMI generation is not affected, additional details of this workaround should be consulted in the SMI Programming chapter of the AMD Hudson-2/Hudson-3 FCH BIOS Developer's Guide (PID # 47518, version 3.0 or later).

Fix Planned

No

29 Incorrect Initialization of XHCI DMA Logic (Hudson-3 only)**Description**

The Hudson-3 XHCI (USB 3.0) controller DMA logic may not correctly initialize during transitions from S4 to S0 or from S5 to S0. If this occurs, DMA logic may fail to complete DMA upstream cycles, thereby blocking forward progress of any subsequent DMA requests to and from memory.

Potential Effect on System

Systems may intermittently hang or fail to enumerate USB devices on USB 3.0 ports during S4 and S5 transitions if the XHCI controller is enabled.

Suggested Workaround

BIOS should perform a soft reset to the XHCI DMA logic during the affected power state transitions. This is implemented in CIMX version 1.0.0.1 or later.

Fix Planned

No

30 Incorrect BANDGAP_ADJUSTMENT Programming**Description**

The VGA BANDGAP_ADJUSTMENT and DAC_ADJUSTMENT registers are configured to a specific reset value to comply with RGB signal amplitude requirements (levels between 665 mV and 770 mV) as defined in the VGA specification. Some parts were programmed with a non-optimal value for BANDGAP_ADJUSTMENT and are not meeting these RGB signal amplitude requirements.

Potential Effect on System

RGB signal amplitudes are measured at approximately 640 mV resulting in a non-optimal VGA output where colors may appear to be washed out.

Suggested Workaround

The VBIOS should program the DAC registers to optimal values. This is implemented in VBIOS version 042A.

Fix Planned

No

32 SD Controller INT_DISABLE Disables MSI

Description

The SD controller provides two registers for enabling or disabling interrupts. Interrupt Disable (PCI_Reg:04h[10]) is used to enable or disable the device from asserting the physical interrupt pin. MSI_Enable (SD_PCI_CFG:82h[0]) is used to enable or disable the use of Message Signalled Interrupts (MSI). Setting MSI_Enable = 1 will automatically disable the use of the physical interrupt pin as intended, however, setting Interrupt Disable incorrectly disables the interrupt pin as well as the MSI functionality.

Potential Effect on System

There is no impact to Windows-based operating systems as Microsoft does not set the Interrupt Disable bit to 1b when MSI is used for the SD controller. However, Linux-based operating systems set both Interrupt Disable as well as MSI_Enable when MSI is intended to be used. Under these operating systems, the SD interface may not function.

Suggested Workaround

BIOS should hide the SD controller's MSI capability by setting PCI_Reg:ACH[1] = 0. The SD controller will use legacy PCI interrupts due to this workaround.

Fix Planned

No

33 Incorrect Implementation of the IOAPIC Delivery Status Bit**Description**

The implementation of the APIC hardware does not comply with the functionality of the Delivery Status Bit (DSB) as defined in the IOAPIC specification.

Potential Effect on System

Diagnostics that check the functionality of the DSB may report an error. No functional failures have been observed during normal system operation outside of such diagnostic environments.

Suggested Workaround

None required. Software normally uses other means of determining interrupt status and does not rely on the DSB.

Fix Planned

No

34 Incorrect ASPM Transitions on the GPP Interface

Description

The PCIe® core logic may not function properly during Active State Power Management (ASPM) L1 and L0s transitions. The necessary conditions required to expose this problem are:

- all attached GPP devices support ASPM L1
- at least one of the attached devices is a PCIe Gen2 device
- PLL powerdown in L1 must be enabled on the FCH (AXINDC_Reg:0x40[3] = 1)

Potential Effect on System

General system instability may be observed such as a soft lock where the affected PCIe Gen2 device is no longer responding, system hangs during S4 or warmboot cycles, or intermittent hangs when using register read/write tools to access the PCI configuration space of the GPP devices/ports.

Suggested Workaround

Disable PLL powerdown on the GPP/UMI links in the FCH by setting AXINDC_Reg:0x40 = 0 when the conditions that expose this issue are present. This change is implemented in CIMx 1.0.0.3.

Fix Planned

No

35 Non-compliance of the S field of the USB Start-Split Transaction Token**Description**

The EHCI controller does not fully comply with the definition of the speed field (S bit) of the Start-Split Transaction Token as defined in the USB 2.0 specification. The requirement that the S bit must be set to 0 for isochronous IN start/splits is not met.

Potential Effect on System

No failures have been observed on platforms using Hudson-2/Hudson-3.

Suggested Workaround

There is no workaround for this issue.

Fix Planned

No

37 XHCI State Machine Locks Up When Swapping USB 3.0 Devices During S3 (Hudson-3 only)**Description**

Swapping USB SuperSpeed devices on a USB 3.0 port while the system is in S3 state may result in a lock up of the XHCI controller state machine. This issue occurs if the new device connection status is registered in the XHCI controller during the sleep state.

Potential Effect on System

The affected USB 3.0 port will not be functional until after a system reset.

Suggested Workaround

BIOS should perform a soft reset of the XHCI controller if it determines that the XHCI state machine lock up has occurred by reading the status of the controller during the resume time. This is implemented in CIMx version 1.0.0.5 and newer.

Fix Planned

No

38 XHCI Controller U1 Exit Time Does Not Meet USB 3.0 Specification (Hudson-3 only)**Description**

The XHCI controller U1 exit LFPS (Low Frequency Periodic Signaling) timing is set incorrectly, resulting in the host response to the device U1 exit signal to be outside the required window of 300 ns to 900 ns.

Potential Effect on System

USB 3.0 devices supporting U1/U2 wake capability may intermittently re-connect at USB 2.0 speeds.

Suggested Workaround

BIOS should program ACPI_USB3.0_REG:90h [29:28] to 0b00. This change is implemented in CIMx 1.0.0.5.

Fix Planned

No

39 ACPI PM Timer Read Returns PMIO Register Data**Description**

An ACPI PM timer IO read of AcpiPmTmrBlk:00h by software may return invalid timer data if there is a concurrent upstream DMA write from HD audio, SD or any PCI device. In this scenario, internal logic may select the incorrect read source resulting in a read of the last accessed PMIO register instead of the PM timer.

Potential Effect on System

Windows XP platforms, which make frequent use of the ACPI PM timer, may experience a BSOD 0xEA failure. Operating systems that use either HPET or TSC timers are unaffected.

Suggested Workaround

BIOS programs the PM timer address to MSRC001_0055[15:0] and sets MSRC001_0055[30] (EnablePMTmrCheckLoop) to enable a workaround. A workaround for processor erratum #686 may also be necessary. Refer to the revision guide for the processor for information on erratum #686.

Fix Planned

No

41 SATA PxIS.DPS Status May Not Set on PRD Completion**Description**

Under specific timing conditions, the Descriptor Processed (DPS) bit of the SATA Port x Interrupt Status register (offset 10h: PxIS, bit 5) may not get set on the completion of the physical region descriptor (PRD) transfer. The data transfer is completed successfully but the status bit may not get set.

Potential Effect on System

Software that relies on the PxIS.DPS status bit for any host-related operations in AHCI mode may encounter a false status indication of the transfer of the PRD. Currently available Windows and Linux operating systems do not rely on PxIS.DPS, however, third party applications, tools or diagnostics that make use of this status bit may experience unexpected behavior including a failure to detect attached SATA devices.

Suggested Workaround

Software should not use the PxIS.DPS status bit as notification of a completion of a PRD transfer. Alternatively, depending on the nature of the FIS (Frame Information Structure) packet, bit 3 (SDBS), bit 2 (DSS), bit 1 (PSS) and bit 0 (DHRS) of the Port x Interrupt Status register can be used to monitor completion of the transfers.

Fix Planned

No

42 XHCI Controller May Be in Wake State While System is in S3 State**Description**

When a system is set to go to S3, the operating system normally disables the USB driver before transitioning the XHCI controller to the D3 (sleep) state. However, during this time, a remote wake event received by a wake-capable device (e.g., a keyboard or mouse) attached to a USB 3.0 port reverts the XHCI controller to a resume state. If this occurs, the USB driver cannot clear the pending wake event prior to the system entering the S3 state, since the driver is disabled. The system is in S3 state while the XHCI controller is in a wake state and, given that USB S3 wake logic will not generate a wake event to ACPI if the XHCI controller is in a wake state, the XHCI controller does not respond to any subsequent wake events.

Potential Effect on System

If the condition described above occurs, wake-capable devices attached to any USB 3.0 ports are not able to wake the system from S3. The power button can be used to wake the system, as this is unaffected by the erratum.

Suggested Workaround

Use updated firmware that checks that the XHCI controller is in a sleep state just before the system enters S3 and forces the controller to the sleep state if it is not. Any pending wake events should be cleared by the BIOS before the system enters the sleep state. The updated firmware and BIOS code with these changes are included in Hudson-2/3 CIMx 1.0.0.8.

Fix Planned

No

43 A20M Port Access May Cause Incorrect Interrupt Acknowledge**Description**

The FCH may respond to an interrupt acknowledge message with a fixed value of 7Fh instead of the actual interrupt vector if a write to IO port 92 to control address bit 20 masking (A20M) occurs at the same time as the interrupt is received from a downstream device. Provided that the BIOS properly passes control to the operating system with A20M disabled, there is no exposure to an operating system that does not use A20M.

Potential Effect on System

Unpredictable behavior may occur if the interrupt handler for vector 7Fh has not been initialized by BIOS during system boot up. This issue was observed as a soft hang on a platform running a protected-mode DOS diagnostic that used A20M.

Suggested Workaround

To prevent unpredictable behavior with legacy operating systems that require use of A20M, BIOS should initialize interrupt vector 7Fh with a valid pointer to a no-op ISR. Although the interrupt request will not get serviced as expected by the initiating device or device driver, this workaround ensures that a soft hang will not occur as a result of an uninitialized interrupt vector.

Fix Planned

No

46 LPC SYNC Timeout Violation**Description**

The LPC host controller does not meet the SYNC timeout requirements as defined in the LPC specification where a memory, IO or DMA cycle started by the host can be aborted if it observes three consecutive clocks without a defined SYNC. Instead, the LPC controller will abort the cycle if it observes two clocks without a defined SYNC. LPC devices that do not respond within two clocks will consistently fail to claim any cycles.

Potential Effect on System

The observed failure was limited to an LPC-to-ISA bridge where ISA devices behind the bridge did not get detected by the operating system. To ensure that other LPC devices have an opportunity to claim the cycle first, this LPC-to-ISA bridge device intentionally waits for the third clock before claiming and passing the memory, IO or DMA cycle to the ISA bus. No failures have ever been observed on commonly used LPC devices such as SIOs, embedded controllers, BMCs or LPC ROMs.

Suggested Workaround

None

Fix Planned

No