SB950 Product Errata

Silicon Errata for SB950

Publication # **49645**	Revision: **3.0**
Issue Date: **June 2012**	

Trademarks

Revision History

Date	Revision	Description
June 2012	3.0	Initial public release based on OEM version 1.4

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
		A11
1	USB Detection on Power Up from Mechanical Off and S3/S4 Resume	No Fix Planned
3	Excessive Latencies May Cause Overwritten USB OHCI Controller Request	No Fix Planned
4	Misinterpreted MSI Requests May Result in Corrupted LPC DMA Data	No Fix Planned
6	Nmi_Enable is Altered When Writing to IO_Reg:72h	No Fix Planned
7	Indeterminate Boot Up State of RTC Bank Selection Bit (DV0)	No Fix Planned
11	Sleep Resume Hang with STPCLK# Throttling Enabled	No Fix Planned
12	EHCI Controller State Machine Micro Frame Counter Synchronization	No Fix Planned
13	IDE Command Timeout Errors	No Fix Planned
14	SATA PHY DLL May Not Lock	No Fix Planned
15	USB Resets Asynchronously With Port CF9h Hard Reset	No Fix Planned
16	Improper Propagation of SATA Message Signaled Interrupts	No Fix Planned
17	Error in USB Frame List Processing	No Fix Planned
18	USB Asynchronous Data Cache Error on Back-To-Back DMA	No Fix Planned
19	USB Babble Detection Logic Disables Unaffected Ports	No Fix Planned
20	SmiCmdStatus Decoding Failure when IMC is Enabled	No Fix Planned
23	Incorrect Implementation of the IOAPIC Delivery Status Bit	No Fix Planned
24	Incorrect ASPM Transitions on the GPP Interface	No Fix Planned
25	Non-compliance of the S field of the USB Start-Split Transaction Token	No Fix Planned
27	PCIe Host Does Not Properly Handle Unsupported Request Transactions	No Fix Planned
28	LPC SYNC Timeout Violation	No Fix Planned

Product Errata

1 USB Detection on Power Up from Mechanical Off and S3/S4 Resume

Description

On power up from mechanical off (G3 → S5 →S0) or when resuming from the S3 or S4 states, some USB 2.0 devices may not get properly enumerated by the USB controller. This detection failure is attributed to the improper initialization of the USB DLL on power-up as a result of coming out of reset before its input reference clock (48 MHz) is stable.

Potential Effect on System

During loading of the operating system, some USB devices may not be detected and will not be present in the Windows® Device Manager.

Suggested Workaround

After power up, the SBIOS must reset the USB DLL by setting ForcePHYDLLreset (PM_Reg: 0xF3[5]). This reset must be performed for S5 exit and during an S3 or S4 resume. When performing this reset, the following conditions must also be satisfied:

- If an S3 or S4 resume is triggered by a USB 1.1 device then the SBIOS must block the USB HOLD resume signal and reset the DLL.
- If an S3 or S4 resume is triggered by USB 2.0 device then the SBIOS must bring the port (from which the wake occurred) out of suspend to the enable state and then issue the DLL reset.

This workaround is available in CIM-X version 0.0.7.0.

Fix Planned

No

3 Excessive Latencies May Cause Overwritten USB OHCI Controller Request

Description

Requests from USB OHCI controllers may be overwritten if the latency for any pending request by the USB controller is very long (in the range of milliseconds).

Potential Effect on System

An operating system crash may occur as a result of USB 1.1 devices becoming unresponsive. Although the conditions necessary for exposure to this failure are independent of the operating system and may occur during normal system operation, this failure has only been observed on Microsoft® operating systems as a BSOD failure (DRIVER_POWER_STATE_FAILURE STOP 0x0000009F, Sub code 003) during long run ACPI S3/S4 cycling.

Suggested Workaround

A system BIOS update available in CIMx version 1.0.0.0 or newer and AMD USB Filter driver version 1.0.14.95 or newer are required in order to avoid this failure. The CIMx change involves disabling OHCI controller pre-fetch on POST, however, given that some USB ISO OUT devices such as USB speakers may experience distorted audio if OHCI controller pre-fetch is disabled, the USB Filter update will enable OHCI controller pre-fetch only when the audio stream to the USB 1.1 ISO out device is active. The USB driver in Linux kernel version 2.6.32 includes the same change that has been applied to the AMD USB filter driver.

Fix Planned

No

4 Misinterpreted MSI Requests May Result in Corrupted LPC DMA Data

Description

An LPC device that supports DMA may encounter data corruption if used with an operating system that supports HPET MSI (e.g., Windows® 7). This is due to a logic bug in the LPC controller that may cause pending MSI requests to be interpreted as a DMA cycle.

Potential Effect on System

LPC-based components that use DMA such as legacy floppy drives or LPC-based FIR (Far Infrared) devices may fail to operate properly. For example, copying files to a legacy floppy device may result in data corruption in the FAT table of the floppy.

Suggested Workaround

Set PM IO 0x50 [4:2] = 000b to disable HPET MSI for platforms that will support LPC devices that use DMA. A platform BIOS callback option to CIMx called HpetMsiDis (available in the AMDSBCFG module of CIMx 1.0.0 or newer) is used to enable or disable HPET MSI depending on the platform configuration. Setting HpetMsiDis =1 will disable HPET MSI. This change has no affect on Windows Vista® that does not support HPET MSI and will force Windows 7 to use the same legacy type interrupts for HPET as is the case for Windows Vista.

Fix Planned

No

6 Nmi_Enable is Altered When Writing to IO_Reg:72h

Description

A write to IO_Reg:72h (Alternate RTC address) may alter bit 7 (NMI_ENABLE) of IO_Reg:70h. The altered value for IO_Reg:70h[7] is not necessarily related to the value being written into IO_Reg:72h[7].

Potential Effect on System

NMIs may be inadvertently enabled or disabled contrary to the intended error handling intentions and capabilities of the platform.

Suggested Workaround

To ensure that the value of IO_Reg:70h is unaffected after writing to IO_Reg:72h, and to ensure that NMIs cannot occur if inadvertently enabled, the following workaround should be applied on each instance of writing to IO_Reg:72h:

1. Save the current values of IO_Reg:70h and IO_Reg:61h (NMI_STATUS).
2. Write 0Ch to IO_Reg:61h to ensure that NMIs cannot occur if unintentionally enabled.
3. Write the intended value to IO_Reg:72h.
4. Restore IO_Reg:70h to original value.
5. Restore IO_Reg:61h to original value.

Fix Planned

No

7 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 (i.e., the first power up after the RTC battery is first installed or after the CMOS is cleared via a motherboard jumper). 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 1.0.0.3.

Fix Planned

No

11 Sleep Resume Hang with STPCLK# Throttling Enabled

Description

Control bits that enable and disable clock throttling (STPCLK# assertion and de-assertion) reside in registers in the S5 power domain that are not cleared by a reset. If clock throttling is enabled prior to entering the sleep state, a STPCLK# assertion message will be immediately sent upstream to the CPU when the system resumes from the sleep state. The assertion message will cause any read from the BIOS ROM to be blocked from execution as the system has not yet been initialized to allow the STPCLK# message to be sent upstream.

Potential Effect on System

Platforms supporting clock throttling may experience a system hang when resuming from a sleep state if either software or hardware throttling was not disabled prior to entering the sleep state. For software-based clock throttling (enabled and controlled by the operating system if the SBIOS exposes the ACPI capability), the conditions necessary to expose this issue are unlikely under normal operation as the throttling would be disabled by the operating system prior to entering the sleep state. For platforms that support hardware-controlled clock throttling, there is added exposure to this issue as there is no operating system intervention to control the enabling or disabling of the throttling logic.

Suggested Workaround

For platforms that support software-based clock throttling, if conditions exist that preclude the ability of the operating system to disable throttling prior to entering a sleep state, an SMI sleep trap can be implemented to ensure that this occurs. For software-controlled clock throttling, bit 4 of CLKVALUE [CpuControl: 00h] is used to enable or disable throttling.

For platforms that support hardware-based throttling, BIOS code must ensure that throttling is disabled prior to entry into any sleep state as well as for any handling of a shutdown due to a thermal condition. Clearing bit 4 of ThrottlingControl [PM_Reg:BCh) will disable clock throttling. Additionally, bit 15 (Therm2SecDelay) of ThrottlingControl can be used to enable a two second delay before clock throttling commences after a shutdown. This will allow sufficient time for the BIOS to initialize the PCIe core, thereby preventing the conditions necessary to expose this issue.

Fix Planned

No

12 EHCI Controller State Machine Micro Frame Counter Synchronization

Description

If, during a narrow window of time within micro frame 7, software enables Periodic Scheduling (USBCMD[4]) with Asynchronous Schedule (USBCMD[5]) disabled and the controller in the RUN state (USBCMD[0] = 1), the EHCI controller's schedule handling logic will lose synchronization with the main state machine and enter a state from which it cannot exit. This scenario has only been observed with Windows 7, however, the conditions necessary to expose this issue may be possible with other operating systems.

Potential Effect on System

For Microsoft-based operating systems, an intermittent BSOD 9F failure may be observed during normal system operation, however, this problem can be exacerbated by stressing the system through sleep cycles while USB flash drives are attached.

Suggested Workaround

If the device driver has both Periodic Scheduling and Asynchronous scheduling disabled, subsequently enabling Periodic Scheduling should use the following sequence:

1. Put the controller in the STOP state (USBCMD[0] = 0)
2. Enable the Periodic schedule
3. Put the controller in the RUN state (USBCMD[0] = 1)

Alternatively, the operating system can remove the conditions necessary to expose this issue by preventing the disabling of Periodic Scheduling. This alternate workaround can be enabled on affected Microsoft operating systems using the "EnHcPeriodicIdle" registry key - refer to Microsoft KB article ID 982091 for details.

Fix Planned

No

13 IDE Command Timeout Errors

Description

When configured for IDE compatibility mode, the SATA controller may fail to respond to an ATA read command if the request is for less than 1 KB of data (one sector). In order for the failure to occur, the data transfer (acknowledged by the write to the device to start the transfer) must complete before the subsequent write is issued that initiates the DMA to transfer the data to memory. Under normal circumstances, this sequence of writes occur in a timeframe where one sector transfers are properly handled, however, system interactions such as SMI activity that interrupt this sequence may expose this problem.

Potential Effect on System

The resulting command timeout errors will degrade performance on the SATA interface. This issue is constrained to environments that transfer one sector or less of data in DMA mode (data transfers of one sector or less in PIO mode are not affected). Such small transfers are not common under normal operating conditions.

Suggested Workaround

A platform BIOS change is required (outside of CIMx) that modifies the programming sequence for DMA transfers such that the Bus Master function is started (writing 1 to the Bus Master IDE command register) ahead of software issuing the DMA transfer command.

Fix Planned

No

14 SATA PHY DLL May Not Lock

Description

On power up, the SATA controller defaults to Third Generation mode with the SATA DLL (Delay Locked Loop) operating at 6 GHz. At this operating frequency and dependent on the voltage of the analog PHY, the DLL may fail to lock. There is increased exposure to this failure during system boot up when SATA ports are simultaneously initialized causing the SATA PHY analog power rail (VDDAN_11_SATA) to experience a voltage droop.

Potential Effect on System

As observed on approximately 1% of pre-production devices, the SATA PHY DLL output may be unstable or operate at the incorrect frequency possibly causing failures in SATA device detection during power up or on resume from sleep states.

Suggested Workaround

For engineering samples, pre-production and production parts, the BIOS should soft reset the SATA DLL before the initial detection process. This change is implemented in CIMx version 1.0.0.6 and newer.

Fix Planned

No

15 USB Resets Asynchronously With Port CF9h Hard Reset

Description

A CF9 reset (IO write 06h to port CF9h) does not reset the USB controller in synchronization with other system devices. Any USB transactions that occur during or after the reset may result in unexpected system behavior.

Potential Effect on System

The system may hang on initialization after the reset. For the case of Windows operating systems, it may result in a BSOD 0xFE error during operating system initialization. Failures have been observed while performing repeated warm boot cycling of the operating system or when using a non-Windows (e.g., DOS) utility that issues a CF9 reset.

Suggested Workaround

Before clearing the reset status, the platform BIOS (i.e., outside of CIMx) should check if the system was restarted due to a CF9 reset (software writing 06h to I/O address CF9h). This can be done by checking the reset status bit 19 of PM_Reg:C0h. If bit 19 is set, reset the USB controller before USB initialization and re-issue the reset by writing 06h to I/O address CF9h. The BIOS must avoid performing this workaround in a loop by skipping these steps during the re-issued CF9 reset.

The following code is used to reset the USB controller:

1. Set PM_Reg: F3h = 87h
2. Delay 1 µS
3. Set PM_Reg: F3h = 00h

Fix Planned

No

16 Improper Propagation of SATA Message Signaled Interrupts

Description

The SATA Message Signaled Interrupt (MSI) from higher port numbers may not properly propagate to port 0 before being sent to the driver.

Potential Effect on System

On a platform with SATA MSI enabled and activity occurring on SATA devices connected to three or more ports, if the failing condition occurs, the driver will not receive the interrupt and the system will hang.

Suggested Workaround

Do not expose the MSI capability in the SATA controller. This change is implemented in CIMx version 1.1.0.2.

Fix Planned

No

17 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 µs) 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

18 USB Asynchronous Data Cache Error on Back-To-Back DMA

Description

An incorrect implementation of the USB data cache address logic may result in data being cached with an incorrect address during a DMA read access that crosses a 4KB boundary. The data that should be cached using addresses in the second page is incorrectly cached with an address corresponding to the same offset in the first page. No issue exists if this address is not used within 800 microseconds when the cache is purged, however, in the event that another USB operation (read or write) corresponds to this incorrect cached address, the EHCI controller may hang (if this subsequent operation is a write), or it may use incorrect data (if this subsequent operation is a read). This erratum is only exposed with USB 2.0 devices connected to ports of the EHCI controller.

Potential Effect on System

This issue has only been observed as a hang of the EHCI controller (affecting all devices attached to that controller) when using a particular USB LAN adapter.

Suggested Workaround

An optional workaround of disabling the USB asynchronous data cache is available in order to remove exposure to this issue. This is achieved via a platform BIOS callback option to CIMx called EhciDataCacheDis introduced in CIMx version 1.1.0.7. Setting EhciDataCacheDis = 1 will disable the cache, however, the default in CIMx is cache enabled (EhciDataCacheDis = 0).

Fix Planned

No

19 USB Babble Detection Logic Disables Unaffected Ports

Description

USB logic that is used to detect and handle babble (any unexpected USB bus activity that persists beyond a specified point in a micro frame) may disable all ports of the EHCI controller.

Potential Effect on System

This issue has only been observed during USB bus enumeration of a particular USB flash drive under Windows 7, Windows Vista and Windows Server 2008. Failures have not been observed in the Linux environment. Any event that would involve enumeration of this device (e.g., hot plugging or resuming from S3 or S4) would expose the platform to the condition where devices on ports of the same EHCI controller are disabled. The affect on other USB devices attached to the same EHCI controller is dependent on whether the devices were active when the babbling condition started. Active devices may get disabled and will not be successfully re-enumerated by the operating system's USB hub driver unless they are disconnected and re-attached. Inactive devices may get disabled, however, these will be promptly re-enabled by the USB hub driver.

Suggested Workaround

Disable babble detection by setting EOR Misc Control (EOR_Reg : EHCI_EOR + 9Ch[11]) = 1b for each EHCI controller. With this workaround in place, the babbling device will be stopped and re-enumerated by the operating system due to the unexpected activity on the USB bus, however, other non-offending devices on the same EHCI controller will be unaffected.

The USB 2.0 specification requires the controller to detect frame babble. The southbridge does not comply with this specification after implementation of this workaround.

Fix Planned

No

20 SmiCmdStatus Decoding Failure when IMC is Enabled

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]. Additional details of this workaround (to ensure that SMI generation is not affected) are available in SMI Programming chapter of the AMD SB950 BIOS Developer's Guide (PID # 48701, version 1.01 and later).

Fix Planned

No

23 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

24 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 (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 by setting AXINDC_Reg:0x40 = 0 when the conditions that expose this issue are present. This change is implemented in CIMx 1.1.1.2.

Fix Planned

No

25 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

Isochronous devices attached to USB hubs that check the S bit in both start-split and end-split transactions will not function. This issue has only been observed using a USB 2.0 Full Speed camera attached to one specific vendor hub device.

Suggested Workaround

There is no workaround for this issue. Due to this erratum, the EHCI controller does not support hubs that check the speed bit for both start split and end split transactions.

Fix Planned

No

27 PCIe Host Does Not Properly Handle Unsupported Request Transactions

Description

Under specific conditions where an unsupported upstream DMA request is received from a PCIe device attached to a GPP port, the PCIe host controller may not return a completion for this request. The correct behavior would be to return a completion with Unsupported Request (UR) specified in the completion status field. Unsupported requests would not be normally generated by a PCIe device except in cases where either the device itself or the device driver is not functioning properly. After the first unsupported request is not completed, the device that sent the request and all other attached GPP PCIe devices may not receive any completions for any upstream requests.

Potential Effect on System

A GPP PCIe device performing a malformed DMA operation (for example, using an invalid memory location) may cause the PCIe device to stop functioning. In some cases, all PCIe devices attached to GPP ports may stop functioning.

Suggested Workaround

None

Fix Planned

No

28 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