AMD-8131™ HyperTransport™ PCI-X Tunnel Revision Guide

Publication # **26310** Rev: **3.02**
Issue Date: **July 2003**

Revision History

Date	Revision	Description
July 2003	3.02	Added erratum #52–53. Changed errata #48 Fix Planned status to “No”.
April 2003	3.00	Initial public release.

AMD-8131™ HyperTransport™ PCI-X Tunnel Revision Guide

The purpose of the *AMD-8131™ HyperTransport™ PCI-X Tunnel Revision Guide* is to communicate updated product information on the AMD-8131™ HyperTransport™ PCI-X tunnel to designers of computer systems and software developers. This guide consists of three major sections:

- **Revision Determination:** This section, which starts on page 6, describes the mechanism by which the current revision of the part is identified.
- **Product Errata:** This section, which starts on page 7, provides a detailed description of product errata, including potential effects on system operation and suggested workarounds. An erratum is defined as a deviation from product specifications, and as such may cause the behavior of the AMD-8131 HyperTransport PCI-X tunnel to deviate from the published specifications.
- **Documentation Support:** This section, which starts on page 21, provides a listing of available technical support resources.

Revision Guide Policy

Occasionally, AMD identifies product errata that cause the AMD-8131 HyperTransport PCI-X tunnel to deviate from published specifications. Descriptions of identified product errata are designed to assist system and software designers in using the AMD-8131 HyperTransport PCI-X tunnel. Furthermore, this revision guide may be updated periodically.

Revision Determination

The BIOS checks the PCI revision ID register at DevA:0x08 to determine the version of silicon as shown in Table 1.

Table 1. AMD-8131 HyperTransport PCI-X Tunnel Revision IDs

Sequence	Revision	Dev[B, A]:0x08
3	B1	12h

Product Errata

This section documents AMD-8131 HyperTransport PCI-X tunnel product errata. A unique tracking number for each erratum has been assigned within this document for user convenience in tracking the errata within specific revision levels. Table 2 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.

Note: *There may be missing errata numbers. Errata that have been resolved from early revisions of the device have been deleted, and errata that have been reconsidered may have been deleted or renumbered.*

Table 2. Cross-Reference of Product Revision to Errata

Errata Numbers and Description	Revision Number
	B1
29 Error In Fairness Algorithm	X
37 Potential PCI-X Mode Starvation Scenario	X
43 PCI-X Secondary Clock Frequency Register Not Updated Properly	X
44 Potential Hang Associated With PCI Disconnect Without Data	X
45 Secondary Bus Reset To A Non-Hot Plug Bridge	X
46 Parity Checking Gap On Peer-To-Peer I/O Writes	X
47 Issues With The PCI-X Relaxed Ordering Bit in I/O Transactions	X
48 Link Electrical Issue When Operating At 800 MHz	X
49 Failure To Complete 4 Kbyte Transfer	X
50 SERR# Enable Does Not Inhibit CRC Sync Floods	X
51 Stale Data Scenario With PCI Prefetching	X
52 Posted Write Starvation Scenario	X
53 Non-Compliant Error Indication From Special Cycles	X

29 Error In Fairness Algorithm

Description

The PCI-X tunnel does not correctly implement the fairness algorithm specified by the HyperTransport I/O link specification. In situations in which a virtual channel (VC) to the host is clogged with more transaction bandwidth than can be supported by the system, the PCI-X tunnel allocates bandwidth as follows—roughly 50% is allocated to inserted traffic from the PCI bridges and roughly 50% is allocated to forwarded traffic from the other side of the tunnel.

Potential Effect on System

Normally, bandwidth to a clogged VC should be allocated fairly between all devices in a HyperTransport chain. For example, the inserted bandwidth in a four-device chain would be allocated as follows:

Host ----	Device 0 ----	Device 1 ----	Device 2 ----	Device 3
	25%	25%	25%	25%

As a result of this erratum, inserted bandwidth to a clogged VC is allocated as follows, based on the number of PCI-X tunnel devices in the HyperTransport chain:

Host ----	Device 0 ----	Device 1
	50%	50%

Host ----	Device 0 ----	Device 1 ----	Device 2
	50%	25%	25%

Host ----	Device 0 ----	Device 1 ----	Device 2 ----	Device 3
	50%	25%	12.5%	12.5%

Note that this erratum only affects chains of three or more PCI-X tunnel devices.

Suggested Workaround

For systems that require three or more PCI-X tunnel devices in a chain, it is recommended that the devices supporting the lowest-bandwidth PCI bridges be placed furthest from the host.

Fix Planned

No

37 Potential PCI-X Mode Starvation Scenario

Description

The PCI-X bridge arbiter uses a round robin protocol for selecting between external masters and internal requests. If all eight non-posted request buffers are consumed by external masters, then the following starvation scenario is possible:

1. The PCI-X tunnel is granted the bus by the arbiter for a split completion to the PCI-X bus for an outstanding non-posted request. As a result of this split completion, there are seven outstanding non-posted requests.
2. The arbiter grants the bus to a first master. This master generates a non-posted request. As a result, all eight non-posted request buffers of the bridge are occupied.
3. The arbiter grants the bus to a second master. This master generates a non-posted request. However, since all non-posted request buffers of the bridge are occupied, this request is retried by the PCI-X tunnel.

This sequence can repeat indefinitely.

Potential Effect on System

PCI-X bus bandwidth may be granted to external masters unevenly.

Suggested Workaround

System BIOS should program the *Maximum Outstanding Split Transactions* field of the PCI-X command register (per the PCI-X 1.0 specification) in external masters as follows, such that a master cannot be starved:

Table 3. Maximum Outstanding Split Transactions

Number of Masters	1st Master	2nd Master	3rd Master	4th Master
1	8			
2	4	4		
3	2	3	3	
4	2	2	2	2

Fix Planned

No

43 PCI-X Secondary Clock Frequency Register Not Updated Properly

Description

After a Hot Plug Speed/Mode Change command (SHPC[B, A]:14[CMD]) to change the speed or mode of the PCI bus, the PCI-X capability registers Dev[B, A]:0xA0[SCF] and Dev[B, A]:0x40[CPCI66] should be updated to reflect the new speed or mode of the PCI bus. However, they remain in their default state for Hot Plug mode, indicating a 33–MHz conventional PCI bus.

Potential Effect on System

None.

Suggested Workaround

None.

Fix Planned

No

44 Potential Hang Associated With PCI Disconnect Without Data

Description

When there is a host access to a secondary-bus PCI device under the following conditions:

- The bridge is in conventional PCI mode.
- The target PCI device responds as a 32-bit device (ACK64# is not asserted).
- The access starts at or passes through the low doubleword of a naturally-aligned quadword (address bit 2 low) and continues to at least the next naturally-aligned quadword. Thus, it covers at least three doublewords.
- The target PCI device always disconnects after no more than one (32-bit) data phase (in which IRDY# and TRDY# are asserted).
- The target PCI device always disconnects without data (STOP# asserted, TRDY# deasserted).

Then the following hang scenario is possible:

1. The first address phase is followed by the first data phase of the transaction, during which the low doubleword of a naturally-aligned quadword is properly transferred.
2. The PCI target disconnects without data.
3. The PCI-X tunnel reinitiates the transaction with the same address as the first address phase (address bit 2 low) with no valid byte enables in the low doubleword (BE#[3:0]=1111b), even though it should have started at the next doubleword address (address bit 2 high).
4. The PCI target enables the transfer of a single doubleword of data corresponding to the low doubleword of the address with no valid byte enables.
5. The PCI target disconnects without data.
6. Steps 3, 4, and 5, are then repeated indefinitely, resulting in the hang condition.

Potential Effect on System

The system hangs.

Suggested Workaround

Device drivers for PCI devices that exhibit the described behavior should be written such that accesses to device registers are limited to no more than 8 bytes per request.

Fix Planned

No

45 Secondary Bus Reset To A Non-Hot Plug Bridge

Description

When (1) the PCI-X tunnel is configured such that one bridge is in Hot Plug mode and the other bridge is not in Hot Plug mode and (2) the secondary bus reset configuration bit, Dev[B, A]:0x3C[SBRST], is set in the bridge that is not in Hot Plug mode, then clearing this bit causes the SHPC hot plug controller for the other bridge to hang.

Potential Effect on System

Subsequent commands to the hot plug controller are not executed. Hot plug events such as changing PCI cards are not possible.

Suggested Workaround

Do not support configurations in which one bridge is in Hot Plug mode and the other bridge is not in Hot Plug mode. If this configuration is required, then do not support software that sets (and then clears) Dev[B, A]:0x3C[SBRST].

Fix Planned

No

46 Parity Checking Gap On Peer-To-Peer I/O Writes

Description

When a PCI device operating in conventional PCI mode generates an I/O write transaction as a master, then the following sequence of events occurs:

1. The I/O write transaction is claimed by the PCI-X tunnel.
2. When IRDY# from the PCI device is asserted, the data associated with the I/O write is captured by the PCI-X tunnel and passed into the corresponding HyperTransport packet.
3. The PCI transaction is retried (disconnected without a TRDY# assertion from the PCI-X tunnel).
4. The PCI device retries the I/O write until the PCI-X tunnel has received the TgtDone HyperTransport packet associated with the I/O write, at which point the transaction is completed with the simultaneous assertion IRDY# and TRDY#.

However, the PCI-X tunnel only checks data parity when IRDY# and TRDY# are both asserted. If there is a transient parity error on the initial transaction in which the data is captured by the PCI-X tunnel, then the error will not be detected.

Potential Effect on System

Erroneous data is passed from a PCI device to another device in the system.

Suggested Workaround

Do not populate the system with devices that generate peer-to-peer I/O writes.

Fix Planned

No

47 Issues With The PCI-X Relaxed Ordering Bit in I/O Transactions

Description

According to the PCI-X specification, the relaxed ordering attribute bit should not be set in I/O requests or completions to I/O requests. However, if the PCI-X tunnel receives an I/O-space HyperTransport read request with the RspPassPW bit set (bit 3 of the RdSized command), then the relaxed ordering attribute bit of the corresponding PCI-X transaction is set.

Note: *There are no known hosts that generate I/O-space requests in which the RspPassPW bit is set.*

Also, if (1) a PCI-X master generates a peer-to-peer I/O request that passes through a HyperTransport link and (2) the PassPW bit of the corresponding HyperTransport response is set, then the PCI-X tunnel sets the relaxed ordering attribute bit in the corresponding PCI-X completion. Even though it is a violation of PCI-X protocol to set the relaxed ordering attribute bit in the completion to an I/O request, it is not a violation of HyperTransport protocol for a target to set PassPW in the response to an I/O request. Therefore, the PCI-X tunnel should clear the relaxed ordering attribute bit in all completions to I/O requests, regardless of the state of PassPW in the response.

Potential Effect on System

The PCI-X protocol violation may result in undefined behavior of PCI-X devices.

Suggested Workaround

Do not support hosts that generate I/O-space HyperTransport read requests with the RspPassPW bit set and do not populate the system with PCI-X devices that generate peer-to-peer I/O cycles.

Fix Planned

No

48 Link Electrical Issue When Operating At 800 MHz

Description

The PCI-X tunnel links do not operate properly at 800 MHz.

Potential Effect on System

Transfer of erroneous data and system deadlocks are possible.

Suggested Workaround

The links should be configured to operate at 600 MHz instead of 800 MHz.

Fix Planned

No

49 Failure To Complete 4 Kbyte Transfer

Description

If a transaction with the following characteristics occurs, then the PCI-X tunnel may fail to transfer the last several bytes of the request:

- A PCI-X master generates a memory read request for 4096 bytes.
- The transaction starting address is not quadword aligned (A[2:0] not 000b).
- Nearly all of the completion data is continuously bursted onto the PCI-X bus without any disconnects.

Potential Effect on System

The system may hang.

Suggested Workaround

System BIOS should set the Maximum Memory Read Byte Count field of the PCI-X Command Register (in the PCI-X type 0 capabilities register set) in all secondary-bus PCI-X devices to no greater than 2 Kbytes.

Fix Planned

No

50 SERR# Enable Does Not Inhibit CRC Sync Floods

Description

If Dev[B, A]:0x04[SERREN] is clear, then no detected errors should result in sync floods on HyperTransport links. However, if DevA:0x[C8 or C4][CRCFEN] is set and a CRC error is detected on the corresponding link, then the PCI-X tunnel will generate a sync flood.

Potential Effect on System

Unexpected sync floods occur.

Suggested Workaround

If it is desired that CRC errors do not cause sync floods, then system BIOS should leave DevA:0x[C8 and C4][CRCFEN] clear.

Fix Planned

No

51 Stale Data Scenario With PCI Prefetching

Description

If the following sequence occurs, then stale data may be delivered to a PCI master:

1. A conventional PCI master initiates transaction A to cacheline n.
2. The PCI-X tunnel prefetches cachelines n, n+1, etc.
3. The response data for transaction A is burst onto the PCI bus, but it is disconnected at the end of cacheline n.
4. There is an intervening host transaction to the bridge (possibly to indicate that new data is valid to the master in question).
5. The master initiates transaction B to cacheline n again.

Then, when the PCI-X tunnel reaches cacheline n+1 on the PCI bus for transaction B, it should discard the data prefetched at cacheline n+1 for transaction A and request the data again. However, it does not do so. It allows the prefetched data associated with transaction A to be supplied for transaction B.

Potential Effect on System

Incorrect data may be supplied to the PCI master.

Suggested Workaround

System BIOS should set Dev[B, A]:0x4C[DPDH].

Fix Planned

No

52 Posted Write Starvation Scenario

Description

Consider a system which includes the AMD-8131 PCI-X tunnel connected to an upstream device (on the A-side link) and a downstream device (on the B-side link). If (1) a continuous stream of response packets are passing downstream through the PCI-X tunnel, (2) these response packets are allowed to pass posted writes (i.e., the PassPW bit is set in all of them), (3) there are always response buffers reported to be available in the downstream device, and (4) a posted write is also sent downstream through the tunnel, then the posted write stalls in the PCI-X tunnel. This posted write will not make forward progress until either there are no more response packets to forward downstream or the downstream device runs out of response buffers.

Note that subsequent downstream posted writes will stall behind this posted write. This scenario may ultimately stall the host from executing instructions if it generates enough posted writes.

Potential Effect on System

The processor may stall for long periods of time, or indefinitely, while it is waiting for the posted channel to clear. This may result in reduced processor performance or in the expiration of watchdog timers in the system.

Suggested Workaround

No workaround is expected to be required. The conditions necessary to create this scenario are not expected to occur with real world applications.

Fix Planned

No

53 Non-Compliant Error Indication From Special Cycles

Description

The upstream response to special cycles that target the PCI bus (initiated by downstream configuration-space writes to offset 0, function 7, device 31 of the secondary bus number) should never indicate a target-abort error. However, if Dev[B, A]:0x3C[MARSP] is set, then the response packet to the special cycle request indicates a target-abort error (the Error bit is set and the NXA bit is clear).

Potential Effect on System

System software may erroneously record that a target-abort error occured.

Suggested Workaround

None expected to be required. There is no known operational software that generates PCI-bus special cycles. If PCI-bus special cycles are required to be supported, then Dev[B, A]:0x3C[MARSP] may be cleared to avoid the problem.

Fix Planned

No

Documentation Support

The following documents provide additional information regarding the operation of the AMD-8131 HyperTransport PCI-X tunnel:

- *AMD-8131™ HyperTransport™ PCI-X Tunnel Data Sheet*, order# 24637.
- *HyperTransport™ I/O Link Specification* (www.hypertransport.org).

See the AMD Web site at *www.amd.com* for the latest updates to documents. For documents subject to a non-disclosure agreement, please contact your local sales representative.