PCI-SIG ENGINEERING CHANGE NOTICE

TITLE:	Interrupt Line Register Usage
DATE:	Junw 16, 2005
AFFECTED DOCUMENT:	PCI-to-PCI-Bridge Archetecture Specification, Rev 1.2
SPONSOR:	Dong Wei, Hewlett-Packard Company

Part I

1. Summary of the Functional Changes

This ECN is a request for modifications to the paragraphs describing the Interrupt Line Register usage for the PCI-to-PCI-Bridge. The purpose is to clarify the differences between the usages on PC-compatible systems and DIG64-compliant systems.

2. Benefits as a Result of the Changes

This ECN eliminates the confusions among the hardware designers and system firmware engineers. Because both PC-compatible systems and DIG64-compliant systems are regarded as industry-standard systems, designers have been using the term BIOS to refer to the system firmware on the DIG64-compliant systems as well. However, the original text only described the BIOS behavior regarding the Interrupt Line Register usage on the PC-compatible systems. Therefore, we have seen confusion among designers when developing DIG64-compliant systems.

3. Assessment of the Impact

This ECN has no impact to the hardware and no impact on PC-compatible software designs. Its purpose is to eliminate confusion by documenting the standard practice today for DIG64-compliant system firmware.

4. Analysis of the Hardware Implications

There is no impact on the design of PCI bridge hardware or systems that use them.

5. Analysis of the Software Implications

There is no content change for PC-compatible systems. There is no impact to the software or firmware on the DIG64-compliant systems because it is believed that DIG64-compliant systems and OSes for these systems (including Windows, Linux, HP-UX and VMS) already operate in the way this ECN describes.

Part II

Detailed Description of the change

Change Section 9.1, starting on page 117 as follows:

9.1. Interrupt Routing

A bridge is not required to route interrupts that originate on the PCI bus connected to the secondary interface of the bridge through the bridge. The *PCI Local Bus Specification* requires the interrupt handler (service routine), or the device which originates the interrupt, to guarantee that all buffers are flushed between the device and the final destination. This can be accomplished by the interrupt service routine of the device driver by performing a read of the device or by the device itself performing a read of the location last written by the device. In either case, the read will force buffers between the device and the final destination to be flushed. No special buffer flushing requirements exist for devices that use Message Signaled Interrupts, as defined by the *PCI Local Bus Specification*. Interrupt messages naturally flush buffers.

~~However, since bridges will be used on add-in cards, the BIOS will assume an association between device location and which INTx# line it uses when requesting an interrupt. Since only the BIOS knows how PCI INTx# lines are routed to the system interrupt controller, a mechanism is required to inform the device driver which IRQ its device will request an interrupt on. The Interrupt Line register (see Section 3.2.5.16) is used to store this information. The BIOS code will assume the following binding behind the bridge and will write the IRQ number in each device as described in Table 9-1.~~ When bridges are used on add-in cards, the binding between the interrupts from the devices behind the bridges and INTx# pins from the card is specified in Table 9-1. When bridges are used on the system board, the binding between the INTx# lines on the system board and the system interrupt controller is not specified. Since only the system firmware knows how INTx# lines are routed to the system interrupt controller, a mechanism is required to inform the device driver which interrupt vector its device will request an interrupt on. This mechanism is system architecture dependent.

For PC-compatible systems, the Interrupt Line register is used to store the IRQ number that is assigned to the INTx# pin of a function within a device (see Section 3.2.5.16). For devices on the system board, the BIOS code uses its knowledge about the routing of the INTx# lines to the system interrupt controller to write into this register. For devices on add-in cards, the BIOS uses the mandatory interrupt binding information defined in Table 9-1 and its knowledge about the routing of the INTx# lines to the system interrupt controller to write into this register.

For DIG64 compliant systems, the system firmware reports the routing of the INTx# lines on the system board to the system interrupt controller via the ACPI (Advanced Configuration and Power Interface) PRT method, as described in the Advanced Configuration and Power Interface Specification Version 1.0 or later. The operating system must assume the mandatory interrupt binding information, as defined in Table 9-1, for devices behind the bridges on add-in cards. The Interrupt Line register is not used by the system firmware or the operating system.

~~The interrupt binding defined in this table is mandatory for add-in cards utilizing a bridge.~~

Table 9-1: Interrupt Binding for Devices Behind a Bridge

Device Number on Add-in Bus	Interrupt Pin on Device	Interrupt Pin on Connector
0, 4, 8, 12, 16, 20, 24, 28	INTA#	INTA#
	INTB#	INTB#
	INTC#	INTC#
	INTD#	INTD#
1, 5, 9, 13, 17, 21, 25, 29	INTA#	INTB#
	INTB#	INTC#
	INTC#	INTD#
	INTD#	INTA#
2, 6, 10, 14, 18, 22, 26,	INTA#	INTC#
	INTB#	INTD#
	INTC#	INTA#
	INTD#	INTB#
3, 7, 11, 15, 19, 23, 27, 31	INTA#	INTD#
	INTB#	INTA#
	INTC#	INTB#
	INTD#	INTC#

Device 0 on a secondary bus will have its INTA# line connected to the INTA# line of the connector. Device 1 will have its INTA# line connected to INTB# of the connector. This sequence continues and then wraps around once INTD# has been assigned.

~~When POST code is initializing the system, it assumes the previous routing information for devices on an add-in card that utilizes a bridge. POST code writes the appropriate IRQ information in each device's Interrupt Line register.~~

Table 9-1 does not specify the routing of a bridge's interrupt pin (if implemented) to the interrupt pins of the add-in connector. Assuming that the bridge is a single function device, its interrupt pin is required to be connected to INTA# by the *PCI Local Bus Specification*.