

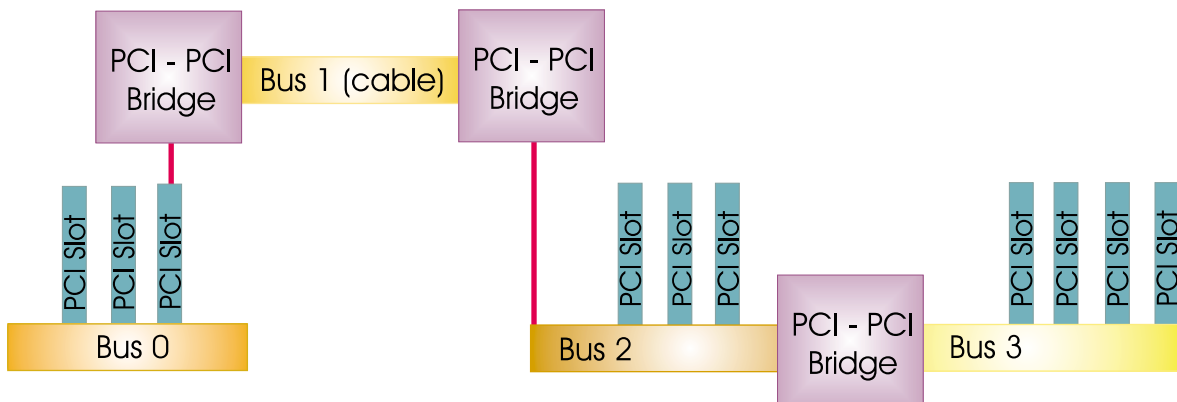
Why Worry About PCI-PCI Bridges When Designing PCI Cards & PCI Systems?

PCI to PCI bridges' most fundamental purpose is to increase the number of devices that a PCI system can support. Loading characteristics of the PCI bus limit the number of devices and/or slots that can reside on a single PCI bus segment. When designing a PCI card, it is best to assume that the card will be plugged into a slot behind a PCI-PCI bridge. In fact, the card could reside anywhere in the PCI bus hierarchy. Similarly, when designing a PCI system, assume it will have cards with PCI-PCI bridges installed.

There are several reasons why PCI add-in cards may be placed behind a PCI-PCI bridge – an embedded bridge may be on the motherboard or, possibly, the cards are installed in a PCI-PCI expansion chassis. As higher speed PCI buses such as 66 MHz PCI and PCI-X are implemented, the likelihood of a card being installed behind a PCI-PCI bridge increases because the number of electrical loads supported decreases.

Description of a System Using PCI-PCI Bridges

The PCI bus hierarchy can take on virtually any configuration. Any combination of parallel and/or daisy-chained bus segments could be implemented. The following example describes a daisy-chained PCI bus hierarchy using an expansion chassis:



- PCI Bus 0 is on the host motherboard.
 - An expansion chassis host card with a PCI-PCI bridge is installed in one motherboard slot. The other side of the PCI-PCI bridge is connected to an external cable creating PCI Bus 1.
 - The cable plugs into an expansion chassis backplane controller card that also has a PCI-PCI bridge. This creates PCI Bus 2. Bus 1 has only two PCI devices on it connecting the host card bridge with the backplane controller card bridge, while Bus 2 on the expansion chassis backplane is connected to three slots and another PCI-PCI bridge.
 - The PCI-PCI bridge on the backplane creates Bus 3 that is connected to four more slots.
- Installing a card with a PCI-PCI bridge in any of the slots can completely change how the bus numbers are assigned; therefore, no assumptions can be made about the bus number of a particular PCI bus segment.

BIOS Considerations

The PCI BIOS is required to traverse the PCI bus hierarchy, and identify and configure all devices connected to it. Instead of assuming there is only a certain number of bus segments or a particular bus structure (i.e. daisy-chained versus parallel or a combination thereof), the BIOS should be generic, not written for a special case.

During configuration, the BIOS should program the bridge device's *Latency Timer* registers and *Cacheline Size* register to a non-zero value. Otherwise, system performance could be severely hampered. For example, leaving the *Latency Timer* register programmed to 0 could cause the bridge to disconnect bursts after only one data phase. Although bridge devices do not have a *MIN_GNT* register since they do not know what device will be behind it, the *Latency Timer* register still needs to be programmed.

Interrupts

Handling interrupts generated by devices behind a PCI-PCI bridge can cause problems if not done properly. In the case of an expansion chassis, all interrupts for devices in the expansion chassis are routed to the interrupt lines on a single slot in the host computer. As a result, the device driver must poll its own hardware to determine whether it is to provide service or not. If its hardware does not have a pending interrupt, the device driver should return control as quickly as possible for the next device driver's use.

The PCI Bridge Specification defines how interrupts on the secondary side of a bridge must be bound to interrupts on the primary side. INTA of a device on the secondary side can generate interrupts INTA-INTD on the primary side (host PCI bus slot). Which interrupt line is asserted on the host bus is determined by the device number of the interrupting device as defined by the PCI Bridge Specification. For example, INTA of a card plugged into a secondary side slot corresponding to PCI Device Number 1 should be connected to INTB on the primary side. The BIOS should be aware of this mapping to properly assign interrupts to each device.

Performance

Secondary Side Arbitration

The PCI-PCI bridge in an expansion chassis provides arbitration for the secondary side bus that it creates. While any type of arbitration scheme could be used, most bridges provide programmable priority assignment; an advantage when devices on the secondary bus are known. If another bridge is on the secondary bus, it may be advantageous to give that bus the highest priority since its single GNT# represents multiple devices on its other side.

Latency Timer Value

The *Latency Timer* register of the PCI-PCI bridge should be set to a value that allows at least one system cacheline to be transferred. Again, since the bridge actually represents multiple devices on its secondary side, it may be best to program the *Latency Timer* to the maximum value.

Cacheline Size

The *Cacheline Size* register is used by the PCI-PCI bridge to determine how much read data are prefetched. To optimize reads, the *Cacheline Size* register should be set to the system cacheline size. If the register is set too small, multiple reads are required to complete a cacheline. If the register is set too high, unnecessary data are prefetched and discarded, wasting bandwidth and increasing latency.

PCI Compatibility Testing

For the highest level of compatibility, test cards and motherboards/BIOS in a system using PCI-PCI bridges. Ideally, execute tests with multiple cards in the system.