Three port PCI bridge performance advantages

By Mohamad Tisani

The PCI Interface, which was originally created for the personal computer, has now been universally adopted by system designers for use in datacom, telecom, PCs, servers, and many other systems. Currently, the PCI Interface is used mainly as an expansion bus to add PCI cards that have wide ranging applications. For example, a router, which was designed by a datacom company, could have eight expansion slots that allow a user to connect eight Ethernet cards. In this application, the designer would have integrated two PCI-to-PCI bridges enabling any Ethernet PCI card to transfer data to any other Ethernet device.

The new 3-Port PCI-to-PCI Bridge from Pericom, a one-chip solution, enables designers to connect all eight Ethernet cards to one PCI bridge. This breakthrough architecture, which has two secondary PCI buses integrated onto the same chip, enables two devices sitting on two different secondary PCI buses to communicate to each other directly without involving the host CPU or main memory.

The Triple Port PCI-to-PCI Bridge has a number of benefits:

- Board real estate saving
- Cost reduction
- Secondary-to-Secondary performance improvement
- Ease of design for redundant applications
- Isolating and saving the bandwidth on the primary bus
- Less loading on the primary device

Saving real estate space on the board

Because the triple-port bridge has dual secondary buses, it can replace two single-port bridges, thus alleviating the problem of crowded boards. This feature is extremely important for telephony or datacom systems. In these applications, the designer must design for as many Ethernet cards as possible, especially for a telephony application. In a CompactPCI system design, you can insert up to 16 PCI peripheral cards on a single bridge using Pericom's PI7C7300 Triple-Port PCI Bridge. When using a dual-port competing application, the limit is eight cards. The Single-Channel PCI-to-PCI Bridge is the same size as the Triple-Port PCI-to-PCI Bridge, but the Triple-Port PCI-to-PCI Bridge provides an additional secondary PCI channel.

Cost reduction

Because the Triple-Port PCI-to-PCI architecture replaces two Single-Port PCI-to-PCI bridges, using Pericom's device can result in reducing costs by one half.

Theoretical analysis: Secondary-to-Secondary performance improvement

Since the Triple-Port PCI-to-PCI Bridge has dual secondary channel communication between the two secondary devices, it takes much less time when transferring data from one bridge to another via main memory or another device to link the two secondary buses created by two separate PCI-to-PCI bridges.

To further analyze performance, consider a 400-byte (100 double word) transfer between two devices, each on a secondary PCI bus:

- Device 1 on the secondary bus 1 in Slot 1
- Device 2 on the secondary bus 2 in Slot 1

Case #1: Using two PCI-to-PCI bridges

Here data must go through main memory, traveling as follows:

- Expansion PCI device transfers data to the PCI bridge
- PCI bridge initiates a transfer to the Primary PCI device
- North Bridge picks up the data and writes it to main memory
- CPU performs an I/O write to indicate that the data transfer is complete
- DMA in the North Bridge reads data from main memory and writes it to the primary PCI bus
- PCI bridge transfers data to the second PCI-to-PCI Bridge and to the destination PCI device

Assuming that the PCI devices can run burst PCI cycles of 4 transfers each, this is what is involved:

- PCI device to PCI bridge. Each burst is 3-1-1-1. Each burst is 6 clocks long. 100/4 = 25
- Add two turnaround cycles for each transfer. Total time = 8 x 25 = 200 cycles
- PCI Bridge-to-PCI Bridge. To transfer data to main memory 8-1-1-1 = 11 PCI clocks
- Add two turnaround clocks. Each transfer takes 13 PCI clocks. Total time = 13 x 25= 325 cycles
- 6 PCI clocks for the I/O cycle
- Main memory to Primary PCI Bridge. Same as above: 13 x 25 = 325 cycles

Reprinted from CompactPCI Systems / July-August 2001

- Primary PCI bus to destination. Same as above. Total time is 8 x 25 = 200 cycles
- Total time to transfer 400 bytes is T: T = 200 + 325 + 6 + 325 + 200 = 1056 PCI clocks

Case #2: Using Pericom's 3-Port PCI Bridge (data only goes through the bridge)

Pericom's PCI-to-PCI bridge, which has 128 bytes FIFO for each bus, keeps accepting data for each PCI clock as long as there is room in the FIFO. To receive the first few D_words from the first secondary bus and post them into the FIFO it takes a maximum of 5 clocks. After the first data has been received, the bridge will request the second secondary bus and initiate the transfer. If both PCI devices are capable of sustaining burst, then the bridge will be able to do 5-1-1-1-1…1 transfer. That means it takes 5 clocks to the first data transfer but following data will take only one PCI clock. The bridge will be able to burst as long as the 4 Kbit line boundary transfer is not crossed.

Total time in this example is T2 = 5 + 99 = 104.

Comparison

In terms of performance, we obviously have tremendous gains; T2 is about 10% of T1. Using Pericom's 3-Port PCI Bridge device, PI7C7300, the gain is: [(1056 –104) / 1056] x 100 = 90%

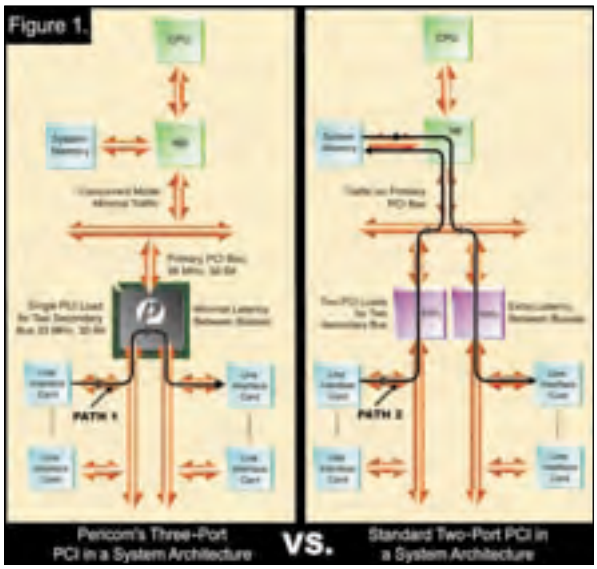
Actual analysis, Secondary-to-Secondary performance improvement

To get real data on performance, we used a PCI Exerciser/Analyzer and the Logic Analyzer to send traffic (100 D_words) through the bridge to the main memory and back. This analysis shows how long it will take to transfer data from one I/O card to another I/O card when going through the main memory.

The other example was to send the data across Pericom's 3-Port PCI-to-PCI Bridge. We captured the waveform and recorded how long it took to transfer the data.

Performance data will depend on what motherboard and chipset designers are using. In the following example, we send 100 D_words from the S1 secondary bus. We wrote 100 D_words to the main memory and then we read the data back, 100 memory read from S2 secondary PCI bus to main memory. Let us call this Path 2 (see Figure 1).

We can achieve the same purpose if we wrote 100 D_words from S1 to S2 on the PI6C7300. Let us call this Path 1 (see Figure 1).



For comparison, in Table 1 we used different chipsets – 430 BX, 440GX, and the 840 Camino from Intel. Table 1 shows actual time in clock cycles to transfer 100 D_words from one I/O card to the next on two different secondary buses, S1 and S2 (Figure 2 shows this information graphically).

Notice that in the Table 1, the performance increase is between 82% to 91%. Also notice that the performance does not change when going from S1 to S2 on the 3-Port PCI Bridge. It is constant with the chipset that is on the motherboard. Also notice that there is a difference as to what command on the PCI bus is used. The other parameter to consider is the Cache Line Size programmed into the chipset and the PCI-to-PCI Bridge. Normally, if you are doing long bursts, then using the largest Cache Line Size and the commands *Memory Read Multiple,*

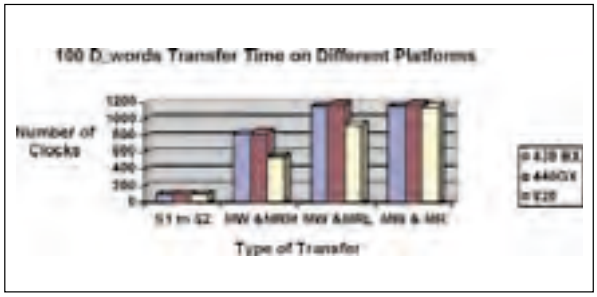


Figure 2

	Path 1 :S1 to S2 Memory write	Path 2 : Memory Write & Memory Read	Path 2 : Memory Write & Memory Read Line	Path 2: Memory Write and Memory Read Multiple	Performance Gain
430 BX	103	1161	1160	821	87% - 91%
440GX	103	1169	1169	827	88% - 91%
820 Camino	103	1152	933	562	82% - 91%

Table 1

and *Memory Write and Invalidate,* will lead to the best performance (especially true of this experiment).

Figure 3 is a logic analyzer snapshot illustrating S1 to S2, 400 Byte transfer performed in a single sustained burst, taking 103 clocks to transfer 100 double words.

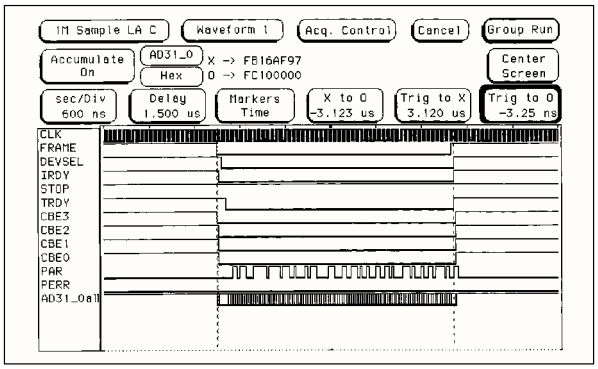


Figure 3

Figure 4 is a logic analyzer snapshot illustrating 100 D_word memory write cycles followed by 100 D_word memory Read cycles. Initiator is on S1, the secondary bus, behind the 3-Port Bridge. This is performed on the 840 Camino chipset.

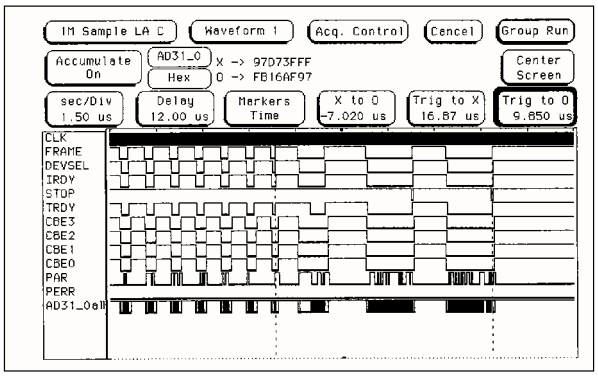


Figure 4

Ease of Design for Redundant Applications

The architecture of the 3-Port PCI Bridge makes it ideal for redundant applications (see Figure 5). In such an application, the system resources are duplicated so that if one resource goes down, the second, or slave resource, will come up and run the system. This feature is very important for banking, defense, and other critical systems

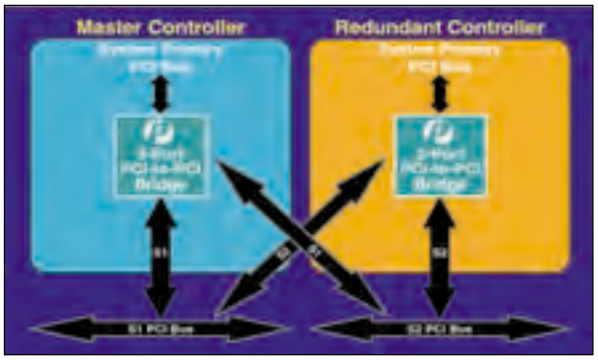


Figure 5

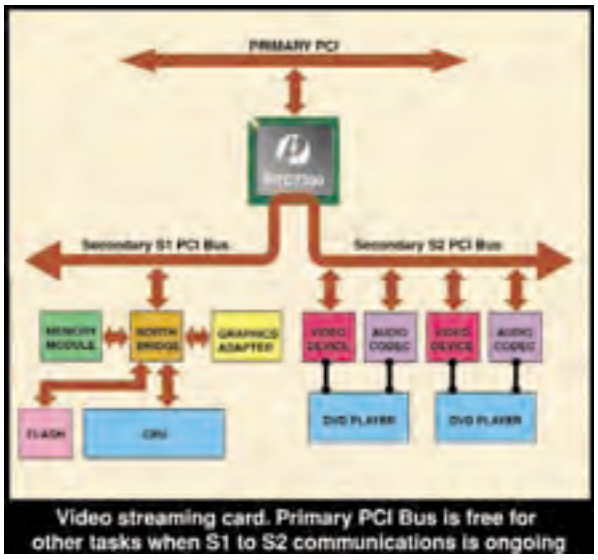


Figure 6

Isolation and saving the bandwidth on the Primary Bus

Reviewing the numbers mentioned in Table 1, notice that by moving data from one secondary to the other, you are actually isolating traffic on the secondary bus alone. Therefore, you are making the Primary PCI bus and the memory bus idle during this time resulting in better I/O performance on the primary PCI bus by up to 80 or 90%. Meanwhile, the memory bus is idle and ready to be used by the CPU or other Primary PCI devices resulting in greater system bandwidth and performance (see Figure 6).

Conclusion

Pericom's 66 MHz Triple-Port PCI-to-PCI Bridge is a hot-swap friendly CompactPCI device that enhances overall system performance. The PI7C7300 permits the attachment of up to 15 CompactPCI cards onto the same board, thus enabling a system designer to save board space and component cost.

Mohamad Tisani *is the Director of Application Engineering at Pericom Semiconductor Corporation. His experience includes System Design and Architecture. He has an MSEE from Santa Clara University, and a BSEE from U.C. San Diego. In his spare time he enjoys Racquetball, Tennis, and Swimming.*

PRODUCT FOCUS AUTHOR

For more information, contact Mohamad at:
Mohamad Tisani
Pericom Semiconductor Corporation
2380 Bering Drive • San Jose, CA 95131
Tel: 408-435-0800, ext. 347
Fax: 408-435-1100
E-mail: at:mtisani@pericom.com
Web: www.Pericom.com

Reprinted from CompactPCI Systems / July-August 2001