

## PCI-X, NGIO and The Future I/O

The next generation of system bus architectures is being created as we speak. The path that began with the move away from minicomputers and mainframes to the first PCs is now widening back to its origins with the development of these new peripheral connection systems. In a few years, PC servers will have dedicated I/O processors, high-bandwidth channels for peripheral cards, and even distributed peripheral systems. Very shortly, desktop system designs will begin to look suspiciously similar to mainframes.

PCI-X is a speed enhancement to the existing PCI bus. The spec promises much higher bandwidth rates of up to 1GBps. Alternately, NGIO moves to an alternative peripheral connect system that

provides speed increments of about 500MBps. Future I/O will be a contemporary to NGIO, and will provide incremental links of 2.5GBps.

### What goes around, comes around:

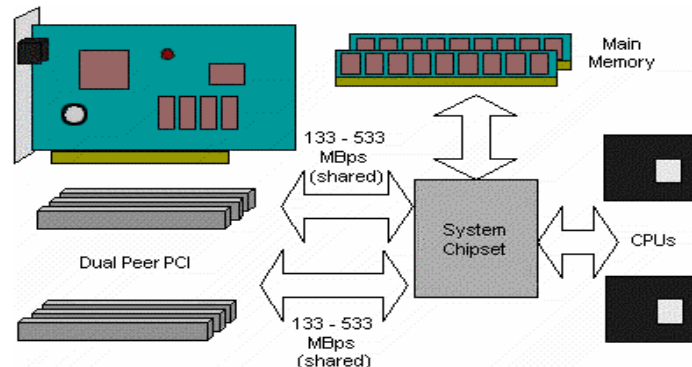
This evolution from mainframe to PC and back again began with the PC's need for more than one PCI bus. The bus is limited to three slots, and most PC systems usually combine two PCI buses into one using a PCI-to-PCI bridge. The bridge simply extends the PCI to support more slots, thus dividing the overall bandwidth across even more cards.

In server models, there was a need for even higher bandwidth for peripheral devices, so dual and triple peer PCI buses were implemented. Each bus in this case is a separate PCI system, all connected to the same system controller or chipset. The system chipset had multiple PCI controllers embedded within.

With the implementation of the I2O system, the PC gained a separate coprocessor based upon the

-1-

Figure: Peer-PCI



Intel i960 microcontroller that could take over some of the duties of the main CPU. With I2O, peripheral cards which require frequent data transfers to communicate with each other, such as storage controllers and network interfaces, are coordinated by the I2O processor. This takes the duties of handling direct memory access transfers over the peripheral bus off the hands of the main CPU(s),

thereby reducing the over workload on the CPU and operating system. I2O is supported in Windows 2000 but not in the Windows 95/98 systems.

But I2O and multiple peer PCI buses are still not enough for today's PC users and their endless peripheral devices sprawling like tentacles and demanding ever greater amounts of bandwidth. Emerging bus architectures like PCI-X and NGIO aim to take connection speeds to another level.

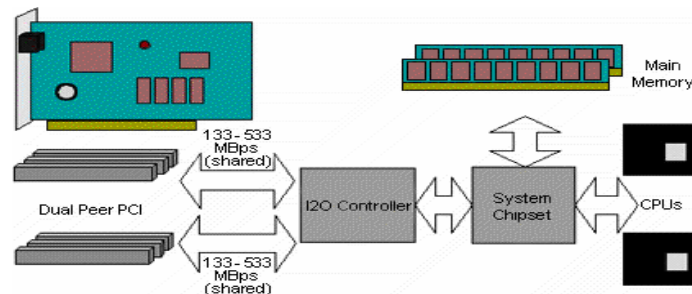


Figure: Peer PCI with Intelligent I/O

-2-

## PCI-X

PCI-X doubles the total bandwidth of PCI to 1.066GBps of shared bandwidth and possibly more. It does this by doubling the frequency of the PCI bus to 133 MHz. A slightly slower version running at 100 MHz may also appear. A PCI bus is backward compatible with cards intended for lower frequencies, such as 33MHz and 66MHz. But since this is a shared bus system, all cards on that bus will run at the same frequencies. Thus, if you have a 33MHz PCI card on the same bus as a 133MHz PCI card, the latter card will slow down to 33MHz. This problem exists even with 66MHz vs. 33MHz PCI cards. With newer PCI systems, you have to worry about what speed your cards run at now – yet another variable to remember in the ever growing complexity of computer systems.

The real limitation of PCI is the signaling it uses. The CPU must slow down to the speed of the PCI bus each time it wants to communicate with

peripheral devices. For every bus operation, the CPU could be performing 10 to 20 other operations, or accessing memory a few more times. It is possible to increase the frequency of the bus, but this introduces even more electromagnetic noise and data unreliability into the system, requiring costly countermeasures.

Thus, the motherboard itself becomes more costly to implement. Moving to 266MHz, for example is too expensive and impractical for vendors today. Moving away from PCI will leave too many vendors behind, on the other hand. Furthermore, PCI is a shared bus system. The effective bandwidth per card decreases as you add more peripheral cards to the system. While the PCI bus is being used by one card, the other cards have to wait until it has completed its business. Although the transfers work very fast, multiprocessor systems become bottlenecked at the PCI bus because of this one-device-at-a-time method.

-3-

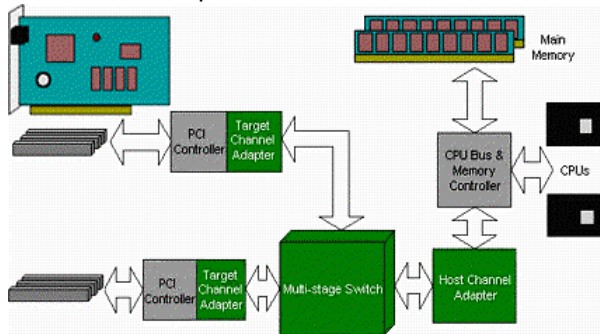
Essentially, with PCI-X we will have reached the limits of the PCI system. It would be time to think of a different method of linking peripheral devices, and this method is what NGIO and Future I/O explore.

**NGIO**

Next Generation I/O (NGIO) combines external (between separate computers) and internal (within a single computer between the CPU and peripherals) interconnects through high-speed serial links to a switch. Rather than replace the format for peripheral cards, at the end of each connection point is a PCI bus. Other connection points can directly interface with Fibre Channel devices.

NGIO introduces channel adapters connecting into a switch fabric. Host channel adapters are directly connected to the main bus of the processor unit, linking processors and memory to the NGIO switch. Target channel adapters at each peripheral unit connect directly to peripherals or to PCI (or other

buses) on one side, and to the switch fabric on the other. There may be multiple physical links between the channel adapters and the switch itself. Each link



**Figure: Next Generation I/O**

is a high-speed serial connection capable of running at speeds of 2.5Gbps. and unlike PCI, which has 100 separate pin connections, these serial lines only need four pins, thus saving in cost.

NGIO changes the method that the CPU uses to transfer data. Instead of the directly synchronized transfers between the peripheral and a CPU, NGIO uses messages to create packets of data that can be sent between each Target Channel Adapter and the Host Channel Adapter.

Even I2O, which offloads I/O processing from the CPU to the I/O controller, still uses synchronous communications, which is clearly a limiting factor. NGIO's asynchronous communication frees the CPU from having to wait for data from the slower peripherals. With multiprocessor systems, this becomes even more important, as processors tend to compete for the slower peripheral bus.

NGIO devices can be up to 17 meters away from the switch, running at speeds of up to 2.5 GBps per Target Channel Adapter. As for total number of devices, NGIO supports a 16-bit addressing scheme, allowing over 65,000 separate devices. NGIO can also connect multiple switches together to create a hierarchy of peripheral systems.

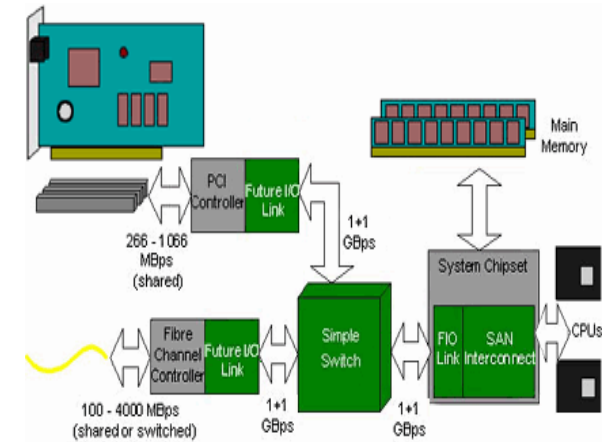
**Future I/O**

The overall concept behind Future I/O is similar to that of NGIO. It's the implementation specifics that differ, along with the capacities and communications protocols.

Future I/O links will have a bidirectional bandwidth of 2GBps and links can be aggregated to support even higher capacities. Copper interconnects can go up to almost 10 meters at this speed and use a parallel cable, compared to NGIO's serial cable. Multimode fiber allows up to 300 meters of range. There is future plan for even greater distances, but this has not yet been disclosed. The general goal is to double and quadruple bandwidth as technology improves and implementation costs go down.

One similarity between Future I/O and an Ethernet network is that Future I/O will use the Internet protocol to communicate between the devices. There is no ordinary IP, however. Future I/O may be the first practical implementation of IP version 6 (IPv6) to go into production machines.

IPv6 has numerous benefits for security, scalability and performance intended to replace IPv4, widely used on the Internet today.



**Fig: Future I/O**

Unlike NGIO, Future I/O is not dependent upon separate external serial lines. Such external lines allow interconnection between two different physical chassis units, but do not allow direct implementation on a motherboard. This means that NGIO devices will always be separate units, whereas Future I/O can either be in a separate chassis or exist as part of the server's motherboard.

This, however, also means that the Future I/O group will have to develop new card formats that plug directly onto a Future I/O channel on a motherboard.

....

