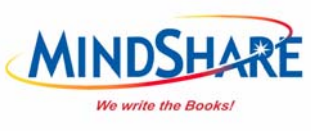
An Introduction to PCI Express

by Ravi Budruk



Abstract

The Peripheral Component Interconnect - Express (PCI Express) architecture is a third-generation, high-performance I/O bus used to interconnect peripheral devices in applications such as computing and communication platforms. The term “third generation” describes the developmental history of the bus: first-generation buses included the ISA, EISA, VESA, and Micro Channel buses, while second-generation buses include PCI, AGP, and PCI-X. Due to its comparitively high speed, low cost and low power, PCI Express is very likely to find a home in mobile, desktop, workstation, server, embedded computing and communication platforms. The intent of this paper is to introduce the reader to the terminology and basics of this exciting new technology.

The Role of the Original PCI Solution

Keep the Good Parts of PCI

The PCI Express architects have carried forward many of the best features of previous generation bus architectures and taken advantage of new developments in computer architecture to improve on them.

For example, PCI Express architecture employs the same usage model and load-store communication model that PCI and PCI-X used, and supports familiar transactions such as memory read/write, IO read/write and configuration read/write transactions. The memory, IO and configuration address space model is also the same as in PCI and PCI-X, which allows existing OSs and driver software to run in a PCI Express system without any modifications. This makes PCI Express software backwards compatible with PCI and PCI-X systems. As an example of this, PCI/ACPI power management software written for current designs should be able to run without modification on a PCI Express machine.

Like its predecessor buses, PCI Express supports chip-to-chip interconnect and board-to-board interconnect via cards and connectors. The connector and card structure are similar to those used for PCI and PCI-X, and were designed so the older PCI and newer PCI Express cards could reside side by side on the same motherboard. A PCI Express motherboard will have a form factor similar to an existing FR4 ATX motherboards so it can fit in the familiar PC package.

... And Make Improvements

To improve bus performance, reduce overall system cost and take advantage of new developments in computer design, the PCI Express architecture was significantly re-designed from its predecessor buses. For example, PCI and PCI-X buses are multi-drop parallel interconnect buses in which many devices share one bus. PCI Express on the other hand implements a serial, point-to-point type interconnect for communication between just two devices on one link. Multiple PCI Express devices can be connected using switches that fan out the buses, making it possible to connect a large number of devices together in a system.

Regarding the point-to-point connection, this implies a reduction in the allowable electrical load on the link, but also permits much higher transmission frequencies which can readily migrate to even higher rates. Currently, the PCI Express transmission data rate is 2.5 Gbits/sec, but work is already in progress on the next higher speed, which will likely be either 5.0Gbits/sec, or 6.25 Gbits/sec. A serial interconnect between the two devices on a given link results in fewer pins per device package, reducing PCI Express chip and board design cost as well as board layout complexity. The performance for a given PCI Express link is highly scalable, and this is achieved by implementing more pins and signal Lanes per interconnect link. The number of lanes needed will, of course, depend on the performance requirements for that interconnect.

To highlight some features of PCI Express, consider that:

- Switches can be used to connect a large number of PCI Express devices in a system.
- Serial communication over the interconnect uses packet-based transactions, and the PCI-X split-transaction protocol.
- Quality Of Service (QoS) features provide differentiated transmission performance for varied applications.
- Hot Plug/Hot Swap support enables “always-on” systems.
- Advanced power management features allow for low-power (mobile) applications.
- Robust error detection and handling features make PCI Express ideal for high-end server applications.
- Hot plug, power management, error handling and interrupt signaling can all be sent in-band using packet-based messaging rather than side-band signals, helping reduce pin count and system cost.
- The configuration address space available per function is extended to 4KB, allowing designers to define additional registers. However, new software is required to access this extended configuration register space.
- PCI-like card and connectors of various sizes are defined for PCI Express. In addition, a mini-PCI Express card and connector as well as a PCMCIA-like Express card and connector are defined.

Looking into the Future

In the future, PCI Express communication frequencies are expected to double and quadruple to 5 Gbits/sec and 10 Gbits/sec. Taking advantage of these frequencies will require a Physical Layer re-design of a device, but may not require any changes to the higher layers of the device design. Support for a Server IO Module, backplane, and Cable form factors are expected in the future.

PCI Express Aggregate Throughput

A PCI Express interconnect is referred to as a Link, and connects two devices. A link consists of either 1, 2, 4, 8, 12, 16 or 32 signals in each direction (note that, because the system uses full-differential signaling, each signal actually needs two wires). These signals are referred to as Lanes. A designer determines how many lanes to implement based on the targeted performance benchmark required on a given link. In the nomenclature, the width of a link is shown with an "x" in front of a number, where the "x" is pronounced as "by", so that a link with 4 signals in each direction, for example, is referred to as "by four" link.

Table 1 shows the aggregate bandwidth numbers for various Link width implementations. As is apparent from this table, the peak bandwidth achievable with PCI Express is significantly higher than most existing buses today. Let's consider how these bandwidth numbers are calculated. The transmission/reception rate is currently 2.5 Gbits/sec per Lane per direction. To support a greater degree of robustness during data transmission and reception, each byte of data to be transmitted is converted into a 10-bit code (via an 8b/10b encoder in the transmitter device). In other words, for every Byte of data to be sent, 10-bits of encoded data are actually transmitted. The result is a 25% overhead to transmit a byte of data. PCI Express implements a dual-simplex Link which implies that data is both transmitted and received simultaneously. The aggregate bandwidth assumes simultaneous traffic in both directions.

To obtain the aggregate bandwidth numbers in Table 1, multiply 2.5 Gbits/sec by 2 (to account for both directions), then multiply by the number of Lanes, and finally divide by 10-bits per Byte (to account for the 8-to-10 bit encoding) to arrive at a bytes/second

result.

Table 1: PCI Express Aggregate Throughput for Various Link Widths

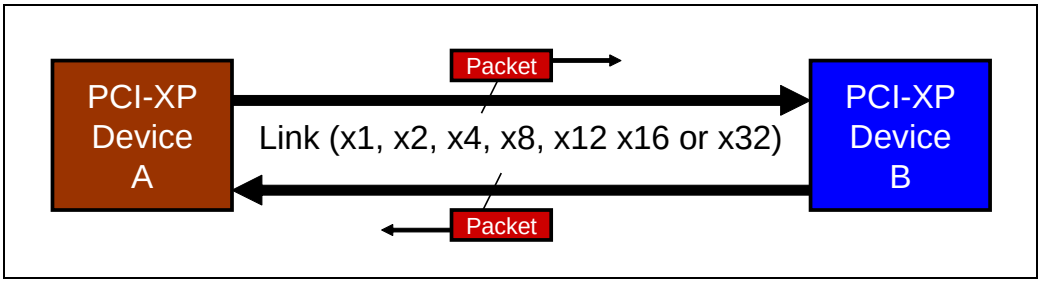
PCI Express Link Width	x1	x2	x4	x8	x12	x16	x32
Aggregate Bandwidth (GBytes/sec)	0.5	1	2	4	6	8	16

PCI Express Features

The Link: A Point-to-Point Interconnect

As shown in Figure 1, a PCI Express interconnect consists of either a x1, x2, x4, x8, x12, x16 or x32 point-to-point Link. To review: a PCI Express Link is the physical connection between just two devices. A Lane consists of signal pairs in each direction. A x1 Link consists of 1 Lane or 1 differential signal pair in each direction for a total of 4 signals. A x32 Link consists of 32 Lanes or 32 signal pairs for each direction for a total of 128 signals. Note that the Link only supports a symmetric number of Lanes in each direction, and does not support asymmetric topologies that would have more lanes from Device A than are sent by Device B in return. During hardware initialization, the Link is automatically initialized for Link width and frequency of operation by the devices on opposite ends of the Link. Neither the OS nor firmware is involved during Link level initialization.

Figure 1: PCI Express Link

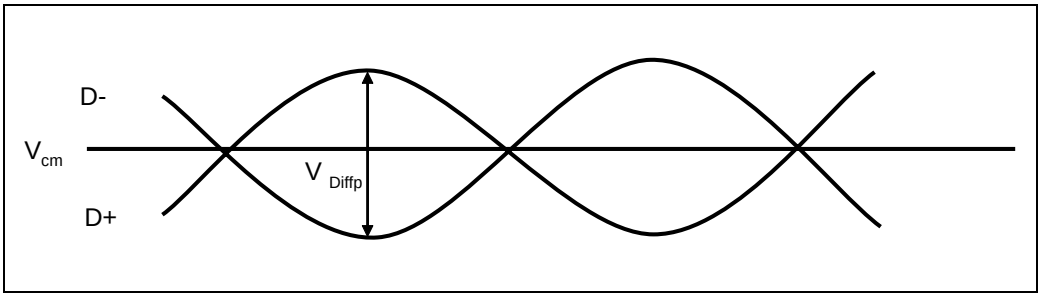


Differential Signaling

PCI Express devices employ differential drivers and receivers at each port. Figure 2 shows the electrical characteristics of a PCI Express signal. Unlike a single-ended signal, whose voltage is compared with system ground, a differential signal is compared only with it's opposite mate, so that the difference between them is what is measured. A positive voltage difference between the D+ and D- terminals implies Logical 1. A negative voltage difference implies a Logical 0. No voltage difference between D+ and D- means that the driver is in the high-impedance tri-state condition, referred to as the electrical-idle and low-power state of the Link.

The PCI Express Differential Peak-to-Peak signal voltage at the transmitter ranges from 800 mV - 1200 mV, while the common mode voltage can be any voltage between 0 V and 3.6 V. The differential driver is DC isolated from the differential receiver at the opposite end of the Link by placing a capacitor at the driver side of the Link, which means that devices at opposite ends of a Link may see different DC common mode voltages. The differential impedance at the receiver is matched with the board impedance to prevent signal reflections.

Figure 2: PCI Express Differential Signal



Switches Used to Connect Multiple Devices

Switches can be implemented in systems that require many devices to be interconnected. The spec does not specify a maximum number of ports a switch can implement, although support for port arbitration would seem to limit them to 256 ports.. A switch may be incorporated into a Root Complex device (Host bridge or North bridge equivalent), resulting in a multi-port root complex. Figure 4 on page 15 and Figure 6 on page 17 show examples of PCI Express systems with multi-ported devices such as the root complex or switches.

Packet-Based Protocol

Unlike the familiar bus cycles used by PCI and PCI-X architectures, PCI Express encodes transactions using a packet-based protocol. Packets are transmitted and received serially across all the available Lanes of the Link at the same time. The more Lanes implemented on a Link the faster a given packet is transmitted and the greater the bandwidth of the Link. The packets are used to support the split-transaction protocol for transactions that are not posted. Several types of packets are defined, such as memory read and write requests, IO read and write requests, configuration read and write requests, message requests, and completions for split transactions.

Clocking

Using a common clock on a bus running at 2.5GHz would be very difficult and require prohibitively short trace lengths to implement. The designers of the PCI Express architecture solved these problems by using an embedded clock instead. As a result, there is no actual clock signal on the Link. Instead, each packet transmitted over the Link consists of bytes of information that have been encoded into a 10-bit symbol. These symbols are guaranteed to have a sufficient number of transitions (from one to zero or vice-versa) that a receiving device, knowing what frequency to expect, can generate a receive clock that closely matches the transmit clock. The receiver uses a PLL and the transitions of the incoming bit stream to create its clock.

Address Space

PCI Express supports the same address spaces as PCI for memory, IO and configuration addresses. However, it also extends the maximum configuration address space per device function from the 256 bytes supported by conventional PCI up to 4 KBytes. To be able to take advantage of this additional configuration space, though, the OS, drivers and applications will need to updated. The reason for this is that the new space must be accessed using a memory-mapped model, which existing software will not understand or be able to access. Also, a new messaging transaction provides messaging capability between devices. Some messages are PCI Express standard messages used for error reporting, interrupt and power management messaging. Others are vendor-defined messages.

PCI Express Transaction Model

PCI Express transactions can be divided into two categories. Those transactions that are non-posted and those that are posted. Non-posted transactions, such as memory reads, implement the split-transaction communication model that PCI-X uses. For example, a requester device transmits a non-posted memory read request packet to a completer, which later returns a completion packet with the read data to the requester. Posted trans-

actions, such as memory writes, consist of a memory write packet transmitted from the requester with no completion packet returned from the completer.

Error Handling and Robust Data Transfer

To guard against data corruption while a transaction is in flight over the link, CRC fields are embedded within each packet transmitted. A required CRC field supports a Link-level error checking protocol whereby the receiver of a packet checks for Link-level CRC errors. Packets that experience corruption are recognized as a CRC error at the receiver and the transmitting device is notified of the error. The transmitter then automatically retries sending the packet in an effort to correct the error (note that this is handled completely by hardware without software involvement). The process by which a receiver indicates the good or erroneous reception of packets to the transmitter is referred to as the ACK/NAK protocol, and is described in more detail in our book.

In addition, an optional end-to-end CRC field can be embedded within a packet to allow for data integrity checking at the destination device. This provides a means to ensure the packets data integrity regardless of how many links it had to traverse to get from the requester to the completer, providing the level of data integrity required for high-availability applications.

Error handling in PCI Express can be as rudimentary as PCI level error handling or can be robust enough for server-level requirements. A rich set of error logging registers and error reporting mechanisms provide for improved fault isolation and recovery solutions required by RAS (Reliable, Available, Serviceable) applications.

Quality of Service (QoS), Traffic Classes (TCs) and Virtual Channels (VCs)

The Quality of Service feature of PCI Express refers to the capability to route packets from different applications through the fabric with different priorities and even with deterministic latencies and bandwidth. For example, it may be desirable to ensure that Isochronous applications, such as video data packets, move through the fabric with higher priority and guaranteed bandwidth, while other packets from lower-speed devices may not have specific bandwidth or latency requirements.

PCI Express packets contain a Traffic Class (TC) number between 0 and 7 that is assigned by the device's application or device driver. Packets with different TCs can move through the fabric with different priority, resulting in varying levels of performance. The packets are routed through the fabric by using virtual channel (VC) buffers implemented in switches, endpoints and root complex devices.

Each Traffic Class is mapped to a Virtual Channel (a VC can have several TCs mapped

to it, but a TC cannot be mapped to multiple VCs). The TC in each packet is used by the transmitting and receiving ports to determine into which VC buffer to drop the packet. Switches and devices are configured to arbitrate between packets from different VCs before forwarding and thus prioritize them. This arbitration is referred to as VC arbitration. In addition, packets arriving at different ingress ports are forwarded to VC buffers at the egress port. These packets are prioritized based on the ingress port number when being merged into a common VC output buffer for delivery across the egress link. This arbitration is referred to as Port arbitration.

The result is that packets with different TC numbers could experience different performance when routed through the PCI Express fabric, depending on how the arbitration has been set up.

Flow Control

A packet transmitted by a device is received into a VC buffer in the receiver at the opposite end of the Link. The receiver periodically updates the transmitter with the amount of buffer space it has available. The transmitter device will only transmit a packet to the receiver if it knows that the receiving device has sufficient buffer space to hold the next transaction. The protocol by which the transmitter ensures that the receiving buffer has sufficient space available is referred to as flow control. The flow control mechanism guarantees that a transmitted packet will be accepted by the receiver, baring error conditions. As such, a PCI Express transaction will not require packet retry unless an error condition is detected in the receiver, greatly improving bus efficiency compared to PCI and PCI-X.

MSI Style Interrupt Handling Similar to PCI-X

Interrupt handling is accomplished in-band using the MSI protocol. PCI Express devices use a memory write packet to transmit an interrupt vector to the host bridge device (root complex), which in turn interrupts the CPU. PCI Express devices are required to implement the MSI capability register block, but also support legacy interrupt handling in-band by encoding interrupt signal transitions (for INTA#, INTB#, INTC# and INTD#) using Message transactions. Only endpoint devices that must support legacy functions and PCI Express-to-PCI bridges are allowed to support legacy interrupt generation.

Power Management

The PCI Express fabric consumes comparitively less power than PCI or PCI-X because the connections consist of fewer signals with smaller signal swings. Each device's power state is individually managed using PCI/PCI Express power management config-

uration registers. Software determines the power management capability of each device and manages it individually in a manner similar to PCI. Devices can notify software of their current power state, and power management software can propagate a wake-up event through the fabric to power-up a device or group of devices. Devices can also signal a wake-up event using an in-band mechanism or a side-band signal.

Devices can place a Link into a power savings state automatically after a time-out when they recognize that there are no packets to transmit over the Link. This capability, which is not conrolled by or visible to software, is referred to as Active State power management.

PCI Express supports the PCI device power states: D0, D1, D2, D3-Hot and D3-Cold, where D0 is the full-on power state and D3-Cold is the lowest power state. It also supports Link power states L0, L0s, L1, L2 and L3, where L0 is the full-on Link state and L3 is the Link-Off power state.

Hot Plug Support

PCI Express supports hot plug and surprise hot unplug without requiring sideband signals. Hot plug interrupt messages, communicated in-band to the root complex, trigger software to detect a hot-add or removal event. Rather than implementing a centralized hot plug controller as in PCI platforms, there is a hot plug controller function for the logic associated with each hot plug capable port of a switch or root complex. Green and amber LEDs, a Manually-operated Retention Latch (MRL), MRL sensor, attention button, power control signal and PRSNT2# signal are some of the elements of a hot-plug-capable port.

PCI Compatible Software Model

PCI Express employs the same programming model as PCI and PCI-X systems and uses the same memory and IO address space. The first 256 Bytes of configuration space for each PCI Express function is the same as PCI configuration address space, ensuring that current OSs and device drivers will be able to find the configuration space they expect and be able to run on a PCI Express system. As mentioned earlier, PCI Express extends the configuration address space to 4 KB per function, but updated OSs and device drivers will be required to take advantage and access this additional configuration address space.

The PCI Express configuration model supports two mechanisms:

1. The PCI compatible configuration model, which is 100% compatible with existing OSs and bus enumeration and configuration software for PCI/PCI-X systems.
2. The PCI Express enhanced configuration mechanism which provides access to additional configuration space beyond the first 256 Bytes, up to 4 KBytes per func-

tion.

Mechanical Form Factors

PCI Express architecture supports multiple platform interconnects such as chip-to-chip, board-to-peripheral card via PCI-like connectors and Mini PCI Express form factors for the mobile market. Specifications for these are fully defined.

PCI-like Peripheral Card and Connector. Currently, x1, x4, x8 and x16 PCI-like connectors are defined along with associated peripheral cards. Desktop computers implementing PCI Express can have the same look and feel as current computers with no changes required to existing system form factors. PCI Express motherboards can have an ATX-like motherboard form factor.

Mini PCI Express Form Factor. The Mini PCI Express connector and add-in card implements a subset of signals that exist on a standard PCI Express connector and add-in card. The form factor, as the name implies, is much smaller and targets the mobile computing market. The Mini PCI Express slot supports x1 PCI Express signals including power management signals. In addition, the slot supports LED control signals, a USB interface and an SMBus interface. The Mini PCI Express module is similar but smaller than a PC Card.

Express Card Form Factor. Another new form factor that will service both mobile and desktop markets is the Express Card form factor. The definition for this form factor is controlled by the PCMCIA group and will be very similar to that standard. Still, it will be about half the size and will support x1 PCI Express signals including power management signals. The slot will also support USB and SMBus interfaces. There are two different sizes defined, a narrower version and a wider version, though the thickness and depth remain the same.

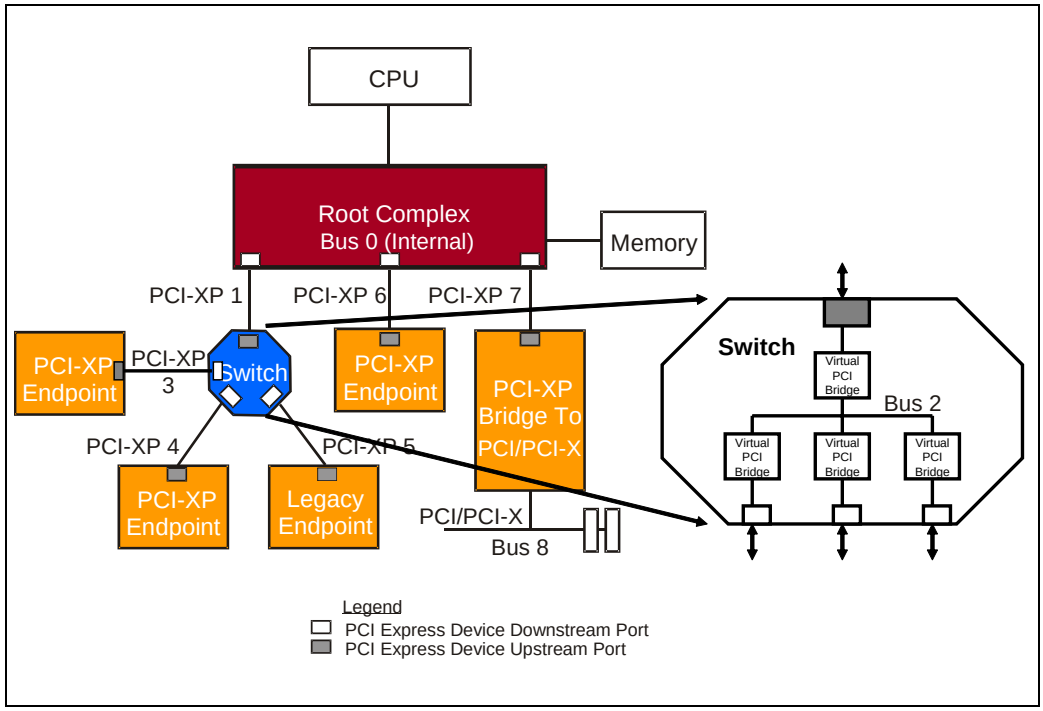
Mechanical Form Factors Pending Release

As of February 2004, Server IO Module form factor specification has not been released, but publicly available information about this form factor indicates that it is a family of designs targeting the workstation and server market. These devices will be designed to support of larger PCI Express Lane widths and higher frequency bit rates beyond 2.5 Gbits/s. Four form factors are under consideration. The base version with single- and double-width modules, and a full-height version with single- and double-width modules.

PCI Express Topology

The major components of the PCI Express system shown in Figure 3 include a root complex, switches, and endpoint devices.

Figure 3: PCI Express Topology



A **Root Complex** connects the CPU and memory subsystem to the PCI Express fabric. It may support several PCI Express ports, and this example shows it supporting 3 ports. Each port is connected to an endpoint device or else to a switch that then forms a sub-hierarchy. The root complex generates transaction requests on behalf of the CPU. In response to CPU commands, it generates configuraion, memory and IO requests as well as locked transaction requests on the PCI Express fabric. The Root complex transmits packets out of its ports and also receives packets into its ports which it then forwards to memory or the CPU. A multi-port root complex may also optionally route packets from one port to another port (supporting peer-to-peer transactions) but is NOT required by the specification to do so.

A Root complex can be used to implement central system resources such as: hot plug

controllers, power management controller, interrupt controller, and error detection and reporting logic. The root complex has a bus number, device number and function number which are used to form a requester ID or completer ID for its transactions, and these all initialize to zeroes.

A **Hierarchy** is the network or fabric of all the devices and Links associated with a root complex that are either directly connected to the root complex via its port(s) or indirectly connected via switches and bridges. In Figure 3 on page 11, the entire PCI Express fabric associated with the root is one hierarchy.

A **Hierarchy Domain** is a fabric of devices and Links that are associated with one port of the root complex. For example in Figure 3 on page 11, there are 3 hierarchy domains.

Endpoints are devices other than root complex and switches that are requesters or completers of PCI Express transactions. They are peripheral devices such as Ethernet, USB or graphics devices. Endpoints initiate transactions as a requester or respond to transactions as a completer. Two types of endpoints exist, PCI Express endpoints and legacy endpoints. **Legacy Endpoints** may support IO transactions, and may support locked transaction semantics as a completer but not as a requester. Interrupt-capable legacy devices may support legacy style interrupt generation using message requests but must also support MSI generation using memory write transactions. Legacy devices are not required to support 64-bit memory addressing capability. **PCI Express (native) Endpoints** must not support IO or locked transaction semantics and must support MSI style interrupt generation. They must also support 64-bit memory addressing capability in prefetchable memory address space, though their non-prefetchable memory address space is permitted to map the below 4GByte boundary. Both types of endpoints implement Type 0 PCI configuration headers and respond to configuration transactions as completers. Each endpoint is initialized with a device ID (**requester ID** or **completer ID)** which consists of a bus number, device number, and function number. Endpoints are always device 0 on a bus.

Multi-Function Endpoints. Like PCI devices, PCI Express devices may support up to 8 functions per endpoint with at least one function being number 0. However, a PCI Express Link supports only one endpoint, numbered as device 0.

PCI Express-to-PCI(-X) Bridge is a bridge between PCI Express fabric and a PCI or PCI-X hierarchy.

A **Requester** is a device that originates a transaction in the PCI Express fabric. The Root complex and endpoints are examples of requester devices.

A **Completer** is a device addressed or targeted by a requester. A requester reads data from a completer or writes data to a completer. The Root complex and endpoints are examples of completer devices.

A **Port** is the interface between a PCI Express component and the Link, and consists of differential transmitters and receivers. An **Upstream Port** is a port that points in the direction of the root complex. A **Downstream Port** is a port that points away from the root complex. An endpoint port is therefore, by definition, always an upstream port. A root complex port(s) is a downstream port. To complete the discussion of port nomeclature, an **Ingress Port** is a port that receives a packet, while an **Egress Port** is a port that transmits a packet.

A **Switch** can be thought of as consisting of two or more logical PCI-to-PCI bridges, each bridge associated with a switch port. Each bridge implements configuration header 1 registers. Configuration and enumeration software will detect and initialize each of the header 1 registers at boot time. A four-port switch as shown in Figure 3 on page 11 consists of four virtual bridges. These bridges are internally connected via a non-defined bus. One port of a switch pointing in the direction of the root complex is an upstream port. All other ports pointing away from the root complex are downstream ports.

A switch forwards packets in a manner similar to PCI bridges using memory, IO or configuration address-based routing. Switches must forward all transactions from any ingress port to any egress port. Switches forward these packets based on one of three routing mechanisms, two of which, address routing and ID routing, are carried forward from PCI and one of which, called implicit routing, is new with PCI Express. The logical bridges within the switch each implement a PCI configuration header 1, containing registers for base and limit addresses downstream from them, as well as for the bus numbers contained in their domain. These registers are used by the switch to determine packet routing and forwarding.

Switches implement both port arbitration and VC arbitration to determine the priority with which to forward packets from ingress ports to egress ports.

Enumerating the System

Standard PCI Plug and Play enumeration software will still be able to enumerate a PCI Express system. The Links are numbered as they are in PCI, using a depth-first search algorithm. An example of the bus numbering is shown in Figure 3 on page 11. Each PCI Express Link is equivalent to a logical PCI bus and is assigned a bus number by the bus enumeration software. A PCI Express endpoint is device zero on a PCI Express Link of

a given bus number, since only one device (device zero) exists per PCI Express Link. The internal bus within a switch that connects all the virtual bridges together is also numbered. The first Link used by the root complex, for example, is bus number one because bus zero is an internal bus. Buses downstream of a PCI Express-to-PCI(-X) bridge are enumerated as they are in a PCI(-X) system.

As in conventional PCI, endpoints may implement up to 8 functions per device and a system could theoretically include up to 256 PCI Express Links or PCI(-X) buses.

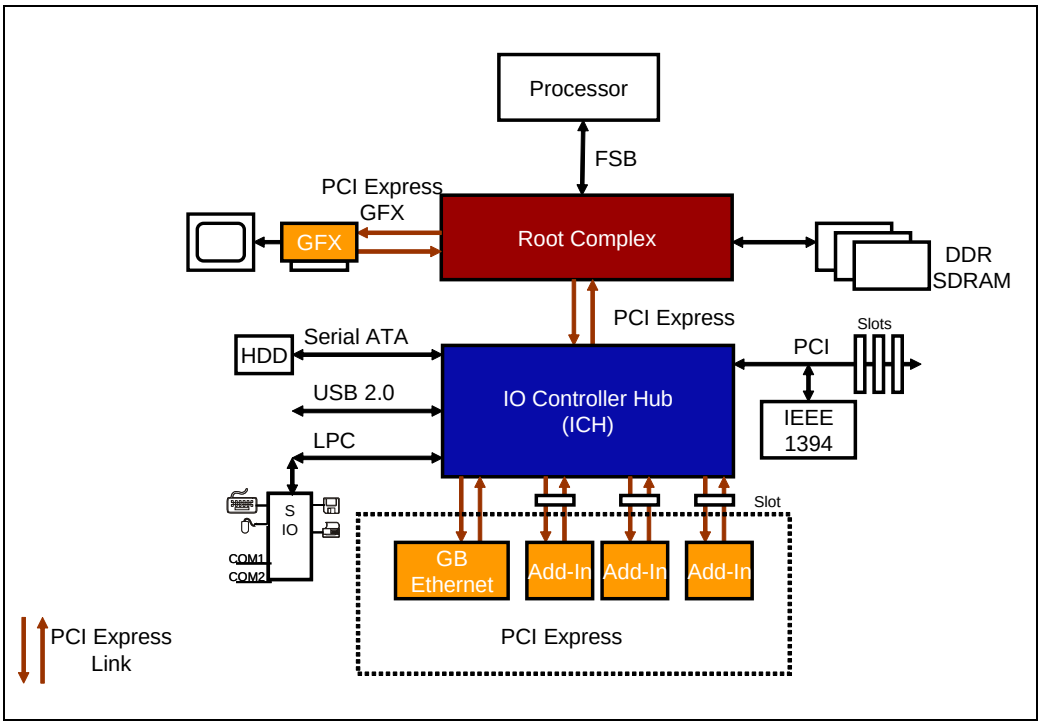
PCI Express System Block Diagram

Low Cost PCI Express Chipset

Figure 4 on page 15 is a block diagram of a hypothetical low-cost PCI Express-based system based on existing chipset architectures. In this solution, the AGP connection between the MCH and a graphics controller in earlier designs is replaced with a PCI Express Link, and so is the Hub Link that connects MCH to ICH. The ICH chip itself supports 4 PCI Express Links which can connect directly to devices on the motherboard or be routed to connectors where peripheral cards are installed.

The CPU can communicate with PCI Express devices via the ICH as well as through the PCI Express graphics controller. PCI Express devices can communicate with system memory or the graphics controller through the MCH, and PCI devices may also communicate with PCI Express devices and vice versa. In other words, the chipset supports packet routing between PCI Express endpoints, PCI devices, memory and graphics. It is yet to be determined if the first PCI Express chipsets will actually support peer-to-peer packet routing between PCI Express endpoints. The specification does not require the root complex to support peer-to-peer between the multiple Links of the root complex. The design shown in Figure 4 on page 15 does not show any switches and, if the number of PCI Express devices to be connected does not exceed the number of Links available, none would be required for this design.

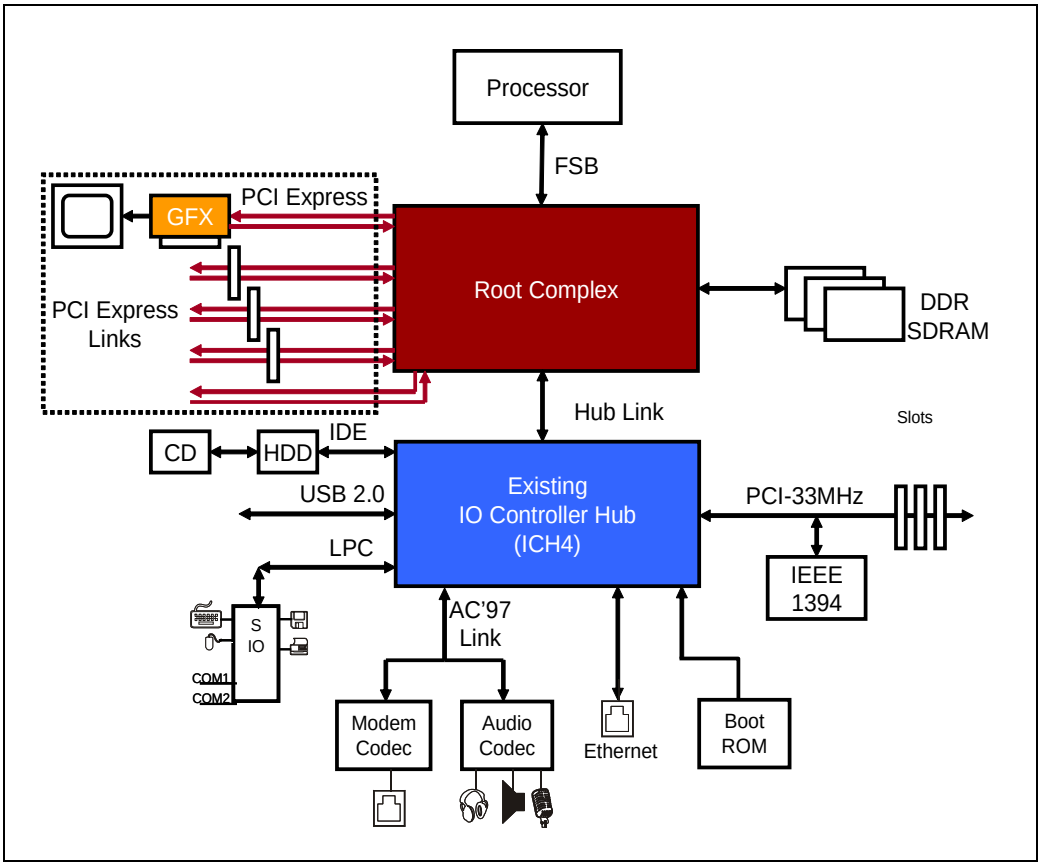
Figure 4: Low Cost PCI Express System



Another Low Cost PCI Express Chipset

Figure 5 on page 16 is a block diagram of another low cost PCI Express system. In this design, a Hub Link connects the root complex to an ICH device, so the ICH can be an existing design which has no PCI Express Links. Instead, all PCI Express Links are associated with the root complex. One of these Links connects to a graphics controller, while the other Links connect to PCI Express endpoints on the motherboard or to PCI Express endpoints on peripheral cards inserted in slots. Publicly available information suggests that Intel is planning to create a chipset with a design very similar to this one.

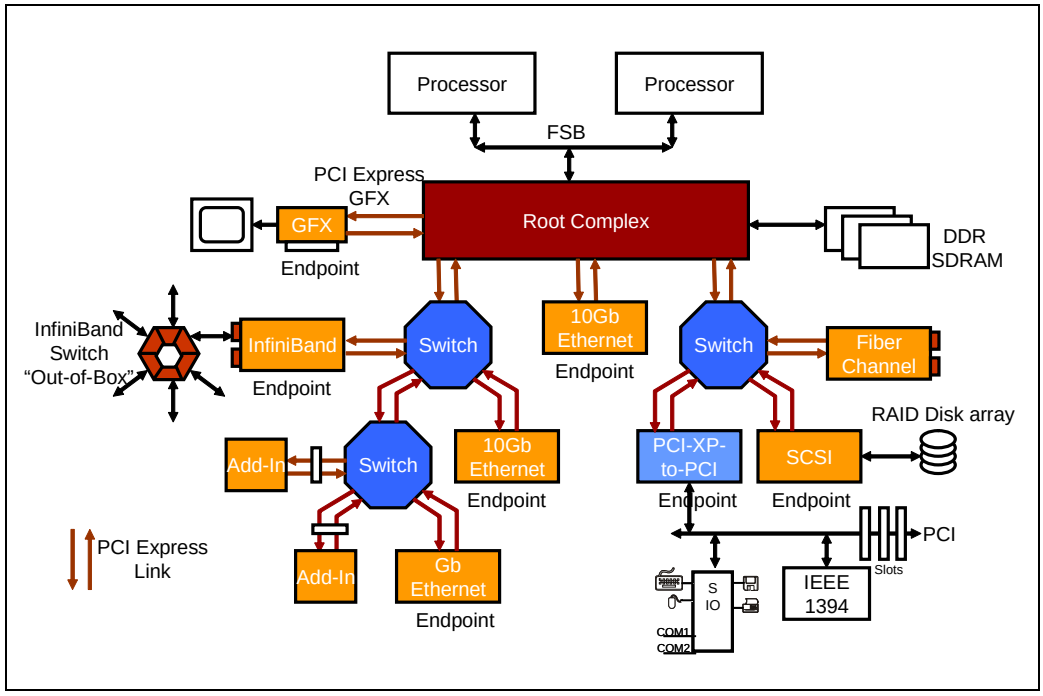
Figure 5: Another Low Cost PCI Express System



High-End Server System

Figure 6 on page 17 shows a more complex system that requires a large number of devices to be connected together. Multi-port switches are necessary to provide the necessary number of connections in this system. To support PCI or PCI-X buses, a PCI Express-to-PCI(-X) bridge is connected to one switch port. PCI Express packets can be routed from any device to any other device because, unlike the root complex, switches are required to support peer-to-peer packet routing.

Figure 6: PCI Express High-End Server System



Conclusion

There is, of course, much more to say about the operation of the the different PCI Express protocols, layers, packets, error handling and so on. The goal for this introduction has simply been to provide a starting point for future study and help the reader understand the terminology and identify areas of interest. To learn more, please visit MindShare's website at www.mindshare.com.

About the author

Ravi Budruk is a Senior Staff Engineer and Instructor with MindShare, Inc, where he shares his in-depth understanding of the PC architecture with hundreds of engineers each year in the hardware and software fields. Ravi is an industry expert on such topics as Intel Processor Architecture, PC architecture, and bus architectures including PCI Express, PCI, PCI-X, HyperTransport, IEEE 1394 and ISA. An excellent presenter, Ravi applies his solid industry experience to the classroom experience to make it more practical and relevant. Before joining MindShare, Inc., Ravi was a PC chipset architect and designer at VLSI Technology, Inc. He obtained an MS degree in Electrical Engineering from Purdue University and a BS degree in Electrical Engineering from Texas Tech University.