

RapidIO[®] Technology and PCI Express[™] – A Comparison

The embedded system engineer, faced with development of a next generation system, has a desire to increase performance, improve efficiency, and lower cost. As part of these efforts the engineer must choose the most appropriate interconnect technology. The system interconnect serves as a cornerstone technology in most embedded infrastructure equipment: many OEMs rely on the long term stability of the interconnect technology to base a rich future of system improvements and an upgrade business. The engineer's interconnect choices may include use of proprietary, home-grown technologies, legacy interfaces, or application-appropriate emerging standard technologies. This paper compares the two leading choices: PCI Express and RapidIO technology. It concludes that while the two interconnect technologies have some similarities, they are quite different in terms of technical merit. In many cases, they can be highly complementary in the overall system architecture landscape.

RapidIO was designed specifically as a widely applicable, flexible, extensible system fabric for embedded infrastructure equipment including networking, storage, and communication systems. PCI Express was formulated as an improvement on the Peripheral Component Interconnect bus, primarily for the commercial computing market. Historically, PCI because of its ubiquitous nature and the consequent economies of scale, has been adopted within embedded systems despite not necessarily providing optimum functionality. There may be a similar desire to force-fit PCI Express into applications beyond the intent of the architectural scope of that interconnect. However, this is likely to be at the expense of inferior functionality, reduced performance, non-standard bridging, and more complex system design than the adoption of an application appropriate standard, such as RapidIO technology.

PCI Express: A Heritage in Commercial Desktop Systems

The purpose of PCI is exactly defined by its name, which is to provide a mechanism to connect peripheral components to a centralized host memory controller. PCI Express offers a vast enhancement in performance over PCI while continuing a software transparent peripheral programming model. The original 33 MHz implementation of PCI provided a peak bandwidth of 1 Gbps and sustainable throughput of approximately 800 Mbps. As desktop peripheral throughputs have increased such as the introduction of Gigabit Ethernet and high speed graphics, the demand on PCI has increased. PCI Express extends the potential bandwidth to 32 Gbps in each direction (16-bit interface) – a substantial improvement on the original implementation. Where previous iterations of improvements on the PCI bus (PCI 2.2, PCI-X) increased bandwidth with higher frequencies, PCI Express achieves a modern high performance interconnect by replacing the multi-drop, parallel bus with a point-to point interface using bonded serial lanes.

The primary focus of the PCI Express architecture was backward compatibility, an essential for the commercial market. This includes the standard address mapped host memory and peripheral connectivity programming model that exists for plug and play PCs today. PCI Express was designed for applications demanding increased bandwidth within commercial PC systems, such as providing a chipset interface for Gigabit Ethernet and replacing AGP graphics card interfaces – this is a function in which PCI Express is already penetrating. It is expected that by the end of 2004, 50% of all new PCs will contain a PCI Express Graphics card. PCI Express should also find an application in modular PC design, and home entertainment hubs. However, beyond these applications, the *need* for PCI Express implementation appears limited. Part of the reason for this is the fact that PCs have become less discrete. Much of the PC has now been integrated into the chipset. This has relieved the burden of having discrete components like

those for modem, audio, disk, and other common interfaces. Peripheral expansion is now divided between PCI Express and the likes of USB 2.0 and Firewire.

Unfortunately, the legacy of PCI places a substantial hindrance for PCI Express in addressing many features provided by modern communications systems (direct peer to peer communication; classes of service; source-based routing; multicast support; message passing protocol; and topological flexibility.) These features could not be implicitly defined in the PCI Express protocol, and thus a complementary system fabric is necessary for PCI Express in these embedded applications. PCI interoperability was explicitly defined in the RapidIO standard, making RapidIO an ideal backplane interconnect or system fabric for PCI Express subsystems.

RapidIO Technology: A Heritage in Embedded Infrastructure Equipment

The demands for processing performance in embedded equipment have increased at a greater rate than discrete processor performance. To address the performance demands, system integrators have turned to more distributed processing models whereby several processing elements are deployed to work in conjunction with each other to solve the bigger problem. Many of these systems are very modular – based around a proprietary standard chassis with backplanes. An advantage of moving to modular distributed processing systems is the ability to deploy just the right amount of processing power to meet the specific needs. By adding or deleting blades from such systems different capabilities and performance levels are possible as well as the associated price points.

Such modularity has put a burden on the system integrator to deliver an underlying system interconnect solution that allows full deployment and exploitation of such topologies in the most efficient form. The requirements for such system connectivity are the driving force behind the definition of the RapidIO architecture. The RapidIO architecture began its definition as a high-performance parallel, chip-to-chip interconnect and has evolved to be a full system interconnect including these completed or in progress specifications that are part of the standard's RapidFabric™ extensions:

- RapidFabric **Flow Control** Logical Layer Extensions Specification
- RapidFabric **Data Streaming** Logical Layer Extension Specifications
 - Phase I: Encapsulation & Traffic Management Framework
 - Phase II: Advanced Traffic Management
- RapidFabric **Multicast** Extensions Specifications
- RapidFabric **Serial Physical Layer** Specification
- RapidFabric **Next Generation Physical Layer** Specifications

The RapidIO architecture has no inherent limitations preventing it from scaling indefinitely into the future, following or anticipating industry requirements.

Embedded System Architectural Needs

It is a common misconception that embedded systems are comprised of the same components as commercial desktop systems. If one were to compare a commercial desktop motherboard, part by part, with an equivalent Compact PCI X86 host card, there is actually very little common silicon. In addition, as commercial motherboards achieve greater levels of integration, concentrating functionality in fewer and fewer parts, their similarity with embedded systems decreases further. The demands of each distinct application dictate specific design needs: embedded infrastructure equipment markets demand long product lifetimes, high reliability and must be insensitive to wide temperature ranges. Thus, while technologies used in commercial computing may be transferred to embedded systems, generally parts in those more demanding systems require application specific design, manufacturing and support. Mass market PC vendors are simply focused on the needs of the desktop market and as a rule choose not to incorporate features or processes that are demanded by embedded system providers.

The architectural demands imposed on PCI Express by the PCI legacy and by the needs of the commercial computing market limit its application in embedded infrastructure systems. Thus, a bifurcation has occurred in the connectivity solutions. There is a desire for a peripheral interface that can connect discrete peripheral components to a memory controller and, in addition, a desire for a system interconnect that can connect processing elements to form a distributed processing machine. The RapidIO interconnect was designed specifically to support the embedded system architecture.

System Topology

Embedded systems are often comprised of hierarchies of processors each with unique tasks. In most embedded systems there is no concept of a central processing element, rather processing elements are distributed based on the task that they perform. Most infrastructure equipment has stringent reliability requirements thus driving the elimination of single points of failure. The interconnect's support for rich system topologies that enable flexibility, performance, and reliability are key.

A fundamental weakness of PCI Express is its dependence on a single host, multiple peripheral communication model. This is an entirely sensible architecture for the host-centric desktop PC where traffic in the system is to and from host memory. In such systems there would be little benefit for an Ethernet controller to directly communicate with an EIDE controller. However, many embedded systems are comprised of multiple processing elements cooperatively working as peers to address a workload. PCI Express does not have such device level peer to peer communication capability. Also missing are the capabilities for fault tolerance through redundant hosts and the ability to do multiple host system exploration and dynamic system discovery. PCI Express architecturally imposes a spanning tree topology therefore eliminating the possibility for topologies commonly found in infrastructure equipment such as the star and mesh.

RapidIO technology on the other hand, is entirely flexible and may be designed with dual-star, mesh, daisy-chain or tree topologies. Data flow can be optimized per application. It allows direct peer to peer communication, and is capable of dynamic discovery with redundant hosts.

In order for PCI Express to emulate the flexibility of RapidIO technology, it would have to terminate transactions using proprietary switches. By stretching the standard, a switch can be designed to allow concurrent data transfers without tying up the entire hierarchy, provided the system does not require data coherency. In such systems a lightweight software synchronization primitive is needed for which none exists in PCI Express. One option is to use interrupts, but these only flow upstream in the PCI hierarchy. Another option is to use buffer queues managed by software. Here designers must allocate a separate queue at each destination for each processor that might send data. In contrast, RapidIO technology has atomic operations eliminating the need for multiple queues. It also offers a messaging facility that can handle the entire queue management operation in hardware, including segmentation and reassembly of large packets.

The limitations of PCI Express not only increases eventual system cost, but the use of proprietary or special purpose switches departs from the mass market economies of scale that purportedly are offered by PCI Express for embedded system design. The spanning tree topology imposed by PCI Express also limits the total number of end points. It has a single unified 64-bit address space and supports a maximum of 32 devices each for 256 buses, and so reaches a maximum of 8k devices in a system. RapidIO technology, in contrast, scales to 64k devices. This, again, reflects the demands of the application of each interconnect: third generation communication systems require thousands of connected devices, whereas for a desktop PC, this capacity accounts for substantial redundancy.

Device Addressing

Devices connected through a common PCI Express network share a common memory map. Device level addressing is accomplished by taking the overall address space and dividing it into chunks that are future subdivided with each subsequent subordinate bus. Each device in the tree is assigned an address space(s) in the overall address map. The target of a transaction is therefore discovered by doing a full address decode. This is sometimes referred to as destination decoding. This type of scheme can become crowded in systems that support many devices with large memories. This can be further exacerbated by systems with topologies that change during runtime. Memory mapped architectures, like PCI Express, also suffer from a significant vulnerability relating to errant devices writing over each other's memory space. To date, designers have worked around this problem, at the expense of system cost, power, and latency in PCI systems by adding non-transparent bridge functionality that enforces memory protection of non-shared address spaces.

Because RapidIO technology was not bound by the legacy support issues of PCI a source based device ID routing scheme was chosen. The beauty of device ID-based routing is that a single routing architecture can be used for both unicast and multicast traffic. Device-based routing simplifies the system and a change in topology requires updates to only the devices in the transaction path. Each RapidIO device is assigned a unique system ID.

Embedded System Functional Needs

Embedded systems with distributed processing elements require a superset of functionality found in host-peripheral DMA based interconnect technologies like PCI. The specific needs of embedded systems are adhered to directly within the RapidIO specification: many functions cater to those unique demands. An embedded system interconnect must have ability to deal with many processors running both homogeneous and heterogeneous OSs, must have synchronization semantics, must have ability to manage both control and data traffic with deterministic transfer characteristics and must be reliable.

Error Recovery

The RapidIO error recovery mechanism was designed for the demands dictated by embedded infrastructure systems. This includes reliable packet delivery, intolerance to packet loss and recovery from errors with minimal system interruption. RapidIO technology is designed to detect and correct errors in hardware at each link in the system. PCI Express adds similar detection and hardware error correction to PCI. However, PCI Express' ability to handle errors is less advanced. Conditions exist that can result in a PCI Express link transmitting the same data packet twice. PCI Express cannot perform retry based flow control nor can it resynchronize its ACK ID counters if things go wrong.

PCI Express has no concept of end-to-end write acknowledgement, and thus cannot verify if an individual data packet was actually transferred. If this level of reliability is required by a system, a read back all data is required. This is a highly inefficient mechanism.

RapidIO packets are explicitly acknowledged link by link. In the case of an out of sequence acknowledgement or a packet detected with a bad CRC, control symbols (used for packet acknowledgement, flow control information and maintenance functions) are sent to resynchronize the sender and receiver and a packet retry is requested. If a packet is lost, a watchdog timeout occurs and an auto-recovery state machine attempts to resynchronize and retransmit. If a transmission fails severely, RapidIO hardware can generate interrupt messages to a system management processor(s), invoking a higher level of error recovery in software.

In contrast, all error notifications in PCI Express flow to the single upstream host without enough contextual information for software to determine the cause of error. This is typical of the needs of host - peripheral connectivity where error detection is important but recovery is usually not.

PCI Express by default requires an LCRC to protect against bit errors on the link. Because the packet changes with each hop through the system the LCRC must be recalculated at each hop. Therefore in order to provide a similar level of end to end error coverage as RapidIO technology, PCI Express requires the additional transmission of an ECRC. The ECRC covers the bits in the packet that do not change end to end. This increases the bit overhead for the header associated with error detection.

These differences again reflect the two interconnects' alternate (and *complementary*) applications: in a desktop PC, errors, and their associated delays, are less crucial than in a communications system.

Message Passing Transactions

In a communication system, data often needs to be shared between multiple processing elements and management of the ownership is required. In such systems it is undesirable to give the producer of the data direct ownership of address space to the consumer as is done in DMA based programming models. Such handshakes in distributed processing machines can be very expensive with respect to processing overhead. RapidIO technology adheres to these demands with a hardware-based Message Passing facility.

The RapidIO Message Passing extensions define *mailbox* and *doorbell* transactions. A mailbox is a port through which devices may communicate, while a doorbell is a lightweight port-based transaction, which can be used for in-band interrupts. These message passing facilities are fundamental to a system's multiprocessor ability to share local memory globally. The transactions enable a reliable, highly efficient, low overhead, means of intra-device communication, providing real-time data transactions throughout a system.

PCI Express has no such interoperable, real-time support.

Data Streaming

Data streaming is clearly a fundamental application for present and future communication systems. Video and voice applications are extremely sensitive to loss and latency: late information is effectively worthless. In contrast, data may be lossy – loss may in fact be designed into a system as a means of flow control. Thus, classification of data priority is crucial in many communications applications.

While PCI Express defines capacity for 8 traffic classes, RapidIO technology distinguishes 256, intended to optimize capacity for carrying streamed data alongside bursty data, and accounting for substantial future extensibility.

The 256 byte maximum packet data payload size used by RapidIO technology is also advantageous in system level data streaming, as compared with PCI Express' 4k. RapidIO technology has much lower bit overhead than PCI Express therefore reaching higher transfer efficiencies with smaller payloads.

By all accounts, PCI Express is essentially a functional subset of RapidIO technology.

Embedded System Performance Needs

Embedded systems process real-time traffic that may consist of voice, video and data of varying packet sizes and requiring different levels of service. The pure speed of the interface is important, but also the way the interconnect handles the packets can affect performance.

Header/Packet Size

The packet sizes of RapidIO technology and PCI Express reflect their primary application focus. RapidIO technology reaches its highest efficiency with a relatively small packet size (256 bytes maximum). This tends to optimize its ability to carry various types of information, allowing for streamed data

transmission alongside bursty system control traffic. PCI Express reaches optimum header to packet ratio with larger packet sizes. This is useful in applications where a peripheral is directly connected to a host memory controller and transferring large DMA payloads.

The use of 4 KByte packets in PCI Express is problematic for larger system topologies that require deterministic throughput. In such systems, it is difficult to manage fairness when a mix of large and small payload traffic exists. For example a large payload transfer can block smaller critical control traffic creating increased transfer jitter. Large packets also demand larger buffers in switches and at end points, increasing eventual system costs. Figure 1 indicates the packet efficiency of four interconnects for various packet sizes. At packet sizes of 256 bytes, RapidIO offers the highest efficiency. However, for larger packet sizes, PCI Express can potentially provide a higher bandwidth for applications requiring vast data transfers.

Transfer Size	Ethernet TCP/IP	Infiniband	PCI Express 4x	RapidIO 4x Serial
	48-bit MAddr	16-bit LID	32-bit Addr	42-bit Addr
32B	16%	32%	36%	53%
64B	28%	48%	53%	70%
128B	44%	65%	69%	80%
256B	61%	79%	82%	88%

- Packet efficiency with link & endpoint acknowledge
- Ethernet and Infiniband have no link acknowledge

Figure 1: Packet Efficiency

Transfer Rates

The maximum serial transfer rates presently available differ between RapidIO technology and PCI Express. RapidIO technology offers a higher speed physical layer and will adopt even faster options in the future. Because PCI Express is targeted as a peripheral interface, higher throughputs are compensated by allowing wider channel bonding than currently defined in the RapidIO specification. Thus, in some chips that allow designers the option of using either PCI Express or RapidIO interconnect, the PCI Express option is wider and thereby offers more bandwidth. However, with a 32-lane port, PCI Express still suffers a 25% deficit in overall bandwidth to eight RapidIO 4x ports. The implementation is also substantially impractical, given the number of SerDes required and the associated higher silicon area and power consumption. Wider interfaces are also impractical for larger topologies as the required switching infrastructure, connectorization, and backplane traces required are not feasible. By contrast, vendors in the embedded space are already working to scale the speed of the RapidIO serial PHY.

While both interconnects were architected with higher throughput in mind, the only driving force for PCI Express to increase substantially in bandwidth seems to be high-end graphics accelerators. With integrated graphics controllers now dominating in volume there is becoming less leverage around the graphics port. RapidIO technology, in comparison, will adopt 5 Gbps and drive it into the embedded ecosystem as soon as it is available. The demand for higher throughput narrow channels comes from the need for higher performance backplane connectivity.

RapidIO technology is building above the communication industry's common roadmap at the physical layer. It uses a variant of IEEE 802.3 XAUI today and will use a variant of the Optical Internetworking Forum (OIF) CEI work in the future. The RapidIO standard will allow for double or quadruple of current rates by the end of 2005. In contrast, PCI Express defines a unique physical layer that must support today's plug-in card designs today and many, many years into the future. The PCI Express community is busy updating the physical layer to higher throughputs, but market inertia could greatly slow the adoption of this technology. Newer technology could obsolete plug-in adapters purchased by desktop consumers every 18 months, which would have the stymie the ecosystem.

Conclusion

Fundamentally, PCI Express and RapidIO technology are complementary interconnect technologies. PCI Express provides a high performance interconnect for peripheral to host DMA connectivity, while RapidIO technology is a high performance switched interconnect designed with specific features for distributed memory, multiprocessor systems.

PCI Express provides an adequate, extensible interconnect for present and future host-based applications, notably graphics cards and IO subsystems in desktop and server PCs. PCI Express will also find its way into embedded equipment in the form of directly attaching peripheral components to memory controllers. While PCI Express can be used in embedded infrastructure equipment, the need of maintaining compatibility with the PCI legacy hinders usefulness of this technology to meet the needs of a full system interconnect.

RapidIO technology complements PCI Express by delivering the required system connectivity features that were never a part of the definition of PCI Express. These include direct peer-to-peer communications, source directed routing, a message passing protocol, classes of service, multicast transactions, and topological flexibility. RapidIO technology can easily interoperate and co-exist with PCI Express as needed – usually with RapidIO serving as the system fabric; however, for the most reliable, scalable, and efficient designs, RapidIO technology is being leveraged for the three C's of connectivity: chip-to-chip, card-to-card, and chassis-to-chassis.

