

coverstory *By Nicholas Cravotta, Contributing Technical Editor*

# HURRY UP



# AND WAIT

## INTERCHIP INTERCONNECTIONS NOW CHOKER TRAFFIC FLOW

### AND LIMIT PEAK PERFORMANCE. NEW STANDARDS

#### MAY BE THE WAY TO OVERCOME THIS IMBALANCE.

**F**OLLOWING THE INTERCONNECT wars is almost interesting. The Hyper-Transport Consortium says it's not directly competing with PCI Express, yet the consortium's own bandwidth comparisons line it up against PCI Express. The PCI Express folks claim that PCI Express will dominate the world because it allows you not only to use one interconnect to connect chips to chips, chips to I/O, and boards to boards, but also to leverage PC economies of scale. Embedded-system engineers shake their heads at statements such as these because in those places in

*At a glance* ..... **42** which embedded-system applications have adopted PCI-based technologies,

the specific implementations are optimized for those applications. You simply aren't going to find off-the-shelf pc boards in a central office. RapidIO, for its part, keeps repositioning itself: The new serial RapidIO is going after the proprietary-backplane market, not chip-to-chip interconnect. Yet, one essential truth remains: No one buys a processor because of the interconnect.

A processor's interconnect may affect your decision to buy a processor because you can't find a suitable coprocessor to connect across that interconnect. So, who cares about these technologies other than the chip-layout engineers?

Wide, parallel, multidrop buses have become too inefficient for today's high-performance applications. The amount of data racing around a pc board, line card, or wireless base station resembles a small network in and of itself. You can no longer lump all traffic onto a single bus, because increasing latencies cannot meet real-time constraints. In the PC world, for example, the front-side bus aggregates system buses, such as those connecting to I/O slots, memory, networking, and graphics coprocessors. To reduce contention latency, you can increase the bandwidth of the front-side bus. However, at some point, you begin to spend more on the infrastructure to connect devices than you do on the devices you want to connect.

HyperTransport, RapidIO, and PCI Express are the main contenders for the disruptive technology that will shift high-performance board design from a shared-bus to switched-bus architecture. Whereas average engineers may still have little choice in which interconnects they work with, they need a general understanding of how these technologies will impact not only chip-to-chip communication, but also board design and system topology.

For example, connecting two processors, whether you connect a processor with a coprocessor or two processors acting as their own masters, requires more engineering than just placing traces across FR-4. The logical issues involved, such as latency and data coherency, greatly complicate multiprocessor interconnect. Another key issue is the reality of homogeneity, a design strategy that assumes using one interconnect technology is more cost-effective than bridging several interconnects.

#### **EFFICIENT MULTIPROCESSING**

With higher bandwidth comes the necessity for greater processing capacity. For applications that require extensive processing of data, such as network processing and digital-signal processing, a single processor often provides inadequate performance. Serial and parallel architectures are the main options for multiprocessor design.

The serial, or datapath, approach aligns processors in sequence; each handles one aspect of the overall processing burden before handing data on to the next processor over a point-to-point in-

#### **AT A GLANCE**

- ▶ No one buys a processor because of its interconnect.
- ▶ The inevitable move from shared-bus architectures to switched-bus architectures for today's high-performance applications will significantly impact board- and system-level design.
- ▶ PCI Express, HyperTransport, and RapidIO have each stretched from their humble beginnings to attempt to become systemwide interconnects between chips, I/O, boards, and even boxes.
- ▶ Efficiently connecting multiple processors is an issue these interconnect standards have only begun to address.
- ▶ Homogeneity of interconnection across a system sounds better than it is.

terconnect. This approach requires no switched or multipoint bus because the data has only one path to follow. For data that requires exceptional or more extensive processing, such as security and encryption processing, you can hang a coprocessor off the side of the datapath, sometimes referred to as a look-aside bus. This approach removes such data from the data path and then reinserts it once the extra processing is complete.

Because the serial approach strictly partitions processing across processors, it is better suited for more deterministic processing in which each data set requires no more than a known limit of resources. A look-aside coprocessor can manage the rare exceptions to this case. The primary interconnect requirements for such an architecture are sufficient bandwidth and acceptable latency.

SPI 4.2 and HyperTransport both better target high-performance, serial, point-to-point architectures because they have less overhead than PCI Express. Although HyperTransport has made some gains in the network-processing market, SPI 4.2 is an established, well-understood interconnect. Manufacturers adopted SPI as a framer interface that supported blocks of data and channelization with flow control; it later spread to other links on the line card. Its primary advantage over HyperTransport is extensive interoperability because it is a single

interface with multiple applications, whereas HyperTransport supports a variety of electrical and protocol modes. For example, two HyperTransport interfaces may not be directly interoperable based on lane-count or features supported. For these reasons, some vendors see SPI as continuing to dominate the high-performance datapath because of its low latency and quality-of-service sensitivity for such features as flow control, congestion management, and signaling. Some vendors see HyperTransport as the coprocessor interface because its developers designed it using a processor-usage model.

The parallel-processing approach places multiple processors in equal relationship to each other. With a serial approach, the datapath is well-defined along a single path. In a parallel approach, such as a DSP farm, data can travel along a variety of paths, depending upon the current state of processor usage. Using a message-passing model, processors pass ownership of data on to other processors. A more complex approach, symmetric multiprocessing, allows several processors to simultaneously use data. Such an interconnection must support a coherency model.

Coherency manages the ownership of data. Consider when one processor modifies data that another processor is using. When you reconcile the results of the two processors, the results will be incoherent because the processors used different values. Managing coherency can be complex. The mechanism must keep track of which processor owns the data and where the most recent image of the data resides, whether that is in a shared memory or an individual processor memory or cache. For instance, the processor may have modified the data in the cache but not yet written it back into memory.

In a message-passing model, such as the one that RapidIO employs, processors have no visibility into each into other. They exchange data using any of a variety of "push" models. For example, Processor B sends messages to Processor A with the data to transfer either through mailboxes or DMA (direct-memory access), in which processors allocate a block of internal memory to each of the other processors in their cluster. Processor B then sends a final message alerting Processor A that the transfer is complete, that the data is now available, and that

Processor A owns it.

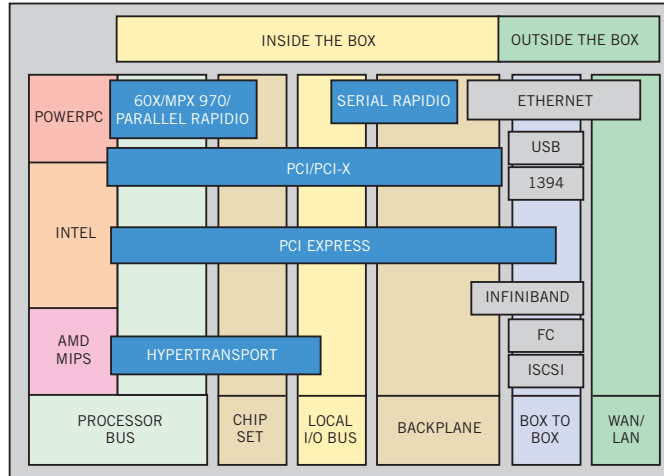
Message passing places the burden of coherency on the owner of the data to broadcast any changes to shared structures. For example, if you want to work in a shared address space, you need either a software or a hardware mechanism outside the RapidIO protocol to manage modification of certain address spaces—that is, locking memory. RapidIO simplifies the implementation of such a mechanism by supporting transactions that can test and set in a single operation to enable one agent to win when multiple agents are contending for

the same semaphore, as well by enabling multiple hosts rather than allowing only one processor to act as host and requiring that all other devices act as slaves.

AMD's new Opteron processor employs a links-based approach using three HyperTransport interconnections to enable symmetric multiprocessing, in which several processors use a GSM (global-shared-memory) architecture. In a GSM, the logical memory address space may spread across several processors and external memories. Symmetric multiprocessing is appropriate for applications that require a small cluster of processors—ideally, four when using Opteron—working tightly together on the same problem. If an application needs more processors, several coherent clusters can work noncoherently—that is, on different parts of the problem.

The links-based architecture attempts to move away from the overburdened traditional front-side-bus architecture that x86 processors employ. With a front-side bus, if Processor A needs access to data stored in external memory attached to Processor B, the situation could delay a request transaction in a variety of ways: Unrelated transactions pending on either Processor A or Processor B can cause contention, as well as the request for and retrieval of data from external memory. Additionally, Processor B may need to use execution cycles to pass this data.

Alternatively, you can use a links-based approach to connect Processor A and Processor B with a dedicated point-to-



**Figure 1** The three main contenders for disruptive interconnection technology—RapidIO, PCI Express, and HyperTransport—have all expanded the scope of applicability from their base of strength to serve as systemwide interconnections, connecting chips, I/O, boards, and boxes (courtesy PLX Technology).

point interconnection. Each Opteron link operates bidirectionally at 800 MHz. Thus, no contention occurs during any of the exchanges between Processor A and Processor B. Latency is important because Processor A doesn't know where the desired data resides. The longer it takes to obtain the data, the longer the processor stalls. According to AMD, bus contention is one of the key bottlenecks, limiting most four-processor systems to at best a 2.5-times-better efficiency than using one processor. Also, managing I/O and data reduces the capability of the processor to compute. Using a point-to-point links-based approach, AMD claims scaling efficiencies greater than three times those of a one-processor approach.

With a two-processor architecture, you can link Processor A to Processor B using one HyperTransport interconnect on each processor, and use the remaining four HyperTransport buses to connect to I/O. This configuration allows Processor A to access I/O on one of Processor B's HyperTransport interconnections without consuming any cycles on Processor B. An internal traffic manager enables this action. The same manager also handles contention when Processor A and Processor B want to access the same I/O.

The HyperTransport specification currently does not include memory coherency. AMD has created a proprietary scheme, which it licenses, that also includes cache coherency. When Processor A wants to use data from a cache, it sends a probe to all the processors. If the data

is in Processor A's cache, it just uses it. If the data is in Processor C's cache, Processor C transfers the data and ownership of the data to Processor A. The transfer from Processor C to Processor A takes more time than a cache hit in Processor A but less time than a cache miss. In this respect, a system with four processors now has four times the available cache.

AMD claims that it chose HyperTransport over PCI Express for the company's links-based architecture because of market timing and the current availability of coherency over HyperTransport. Admittedly, the

interfaces are fairly similar, and the decision may have been different if AMD were starting work on the Opteron today. As a consequence, HyperTransport has found a foothold in the PC market even as PCI Express proponents claim that PCI Express will dominate this market.

**NO BOUNDARIES**

A key aspect of the interconnect wars is homogeneity. For some applications, multiprocessor systems can cross the backplane boundary; try as you might, you just can't fit 100 processors onto the same line card. The idea behind homogeneity is that you employ a single interconnection technology between chips, between boards (backplane), and, potentially, between boxes. Using a single technology eliminates bridging and any chips, cost, and latency associated with bridging.

The push for homogeneity has tenuous beginnings. Some say that PCI Express is more an I/O than a processing bus and that Serial RapidIO targets backplanes; HyperTransport's developers claim that it is the most efficient chip-to-chip interconnect. It appears that the standards drafters, however, decided it would be a good idea to expand the applicability of each standard and stretch them into standards that do it all (Figure 1).

Keeping all your data in the same protocol domain has its appeal. Bridging requires shifting in the clock domain, encapsulation of diverse or nonexistent quality-of-service functions in the logi-



cal protocol, and an extra FPGA or bridge chip. Also, you can damage the robust mechanisms of HyperTransport or RapidIO by tunneling over PCI Express.

However, the homogeneity argument is misleading. You still need a bridge to connect RapidIO buses that are different widths or clock rates. HyperTransport requires switches, so you have to place a device on the board anyway; couldn't it be a bridge or hybrid device? Also, is the PCI Express you use between two processors the same as that you would use across backplanes? In any case, today's interconnections won't disappear overnight. The question isn't whether you'll need any bridges in a homogenous system but whether you'll be able to eliminate any of those you already have.

HyperTransport has de-emphasized the backplane, and RapidIO has scaled back its parallel version and introduced Serial RapidIO to address the backplane. Even PCI Express has recognized that an effective I/O interconnect may not cut it on the backplane; hence, the birth of Advanced Switching, which is arguably no more PCI-based or compatible than either RapidIO or HyperTransport.

As the SPI interconnect illustrates, general interconnects can't serve every application as efficiently as proprietary or specialized interconnects can. Although homogeneity promises less complexity and lower cost from higher volumes across multiple applications, it sometimes comes at the price of effi-

ciency, which can often make the difference between a leading-edge product and a me-too implementation. Not every device needs the bandwidth or high frequencies of a serial interconnect, and wide, parallel buses may offer better latency in these cases.

These interconnect technologies still need to address many significant issues, such as coherency. For many applications, these standards have positioned themselves as replacements for proprietary interconnects. Although backplane vendors are calling for a standard technology, are they willing to compromise the critical efficiency of the backplane because the standard also accommodates unrelated chip-to-chip issues?

It will be interesting to see how companies such as Intel, Motorola, and AMD drive each of these standards according to their own agendas. Perhaps one approach is to accommodate all of the standards. For example, you can buy an IP (intellectual-property) block that electrically and logically supports both HyperTransport and SPI 4.2. Although challenging to create and potentially requiring limited truncation of functions, a multiprotocol interconnection on a processor would indeed provide engineers with a choice of interconnection scheme. □



You can reach Contributing Technical Editor Nicholas Cravotta at editor@nicholascravotta.com.

TALK TO US  
Post comments via TalkBack at the online version of this article at [www.edn.com](http://www.edn.com).

### FOR MORE INFORMATION...

For more information on products such as those discussed in this article, contact any of the following manufacturers directly, and please let them know you read about their products in *EDN*.

**AMD**

[www.amd.com](http://www.amd.com)

**Fulcrum Microsystems**

[www.fulcrummicro.com](http://www.fulcrummicro.com)

**IDT**

[www.idt.com](http://www.idt.com)

**Intel**

[www.intel.com](http://www.intel.com)

**Mercury Computer Systems Inc**

[www.mercury.com](http://www.mercury.com)

**Motorola**

[www.motorola.com](http://www.motorola.com)

**National Semiconductor**

[www.national.com](http://www.national.com)

**PLX Technology**

[www.plxtech.com](http://www.plxtech.com)

**Xilinx**

[www.xilinx.com](http://www.xilinx.com)

**For a complete listing of vendors for each technology, visit the standards' Web sites:**

**Advanced Switching Interconnect Special Interest Group**

[www.asi-sig.org](http://www.asi-sig.org)

**HyperTransport Consortium**

[www.hypertransport.org](http://www.hypertransport.org)

**PCI Express**

[www.pcisig.com/specifications/pciexpress](http://www.pcisig.com/specifications/pciexpress)

**RapidIO Trade Association**

[www.rapidio.org](http://www.rapidio.org)

**System Packet Interface**

[www.oiforum.com/public/impagreements.html](http://www.oiforum.com/public/impagreements.html)