This chapter discusses some of the motivations leading to the development of the RapidIO interconnect technology. It examines the factors that have led to the establishment of new standards for interconnects in embedded systems. It discusses the different technical approach to in-system communications compared with existing LAN and WAN technologies and legacy bus technologies.

The RapidIO interconnect technology was developed to ease two transitions that are occurring in the embedded electronics equipment industry. The first transition is technology based. The second transition is market based. The technology transition is a move towards high-speed serial buses, operating at signaling speeds above 1 GHz, as a replacement for the traditional shared bus technologies that have been used for nearly 40 years to connect devices together within systems. This transition is driven by the increasing performance capabilities and commensurate increasing bandwidth requirements of semiconductor-based processing devices and by device level electrical issues that are raised by semiconductor process technologies with feature sizes at 130 nm and below. The second transition is a move towards the use of standards based technologies.

1.1 PROCESSOR PERFORMANCE AND BANDWIDTH GROWTH

Figure 1.1 shows the exponential growth of processor performance over the last 30 years. It also depicts the slower growth of processor bus frequency over that same period of time. The MHz scale of the chart is logarithmic and the difference between the core CPU performance, represented by the clock frequency, and the available bandwidth to the CPU, represented by the bus frequency, continues to grow. The use of cache memory and more advanced processor microarchitectures has helped to reduce this growing gap between CPU performance and available bus bandwidth. Increasingly processors are being developed with large integrated cache memories and directly integrated memory controllers. However multiple levels of
on- and off-chip cache memory and directly integrated memory controllers, while useful for reducing the gap between a processor’s data demands and the ability of its buses to provide the data, does little to support the connection of the processor to external peripheral devices or the connection of multiple processors together in multiprocessing (MP) systems.

In addition to the increasing performance of processors, the need for higher levels of bus performance is also driven by two other key factors. First, the need for higher raw data bandwidth to support higher peripheral device performance requirements, second the need for more system concurrency. The overall system bandwidth requirements have also increased because of the increasing use of DMA, smart processor-based peripherals and multiprocessing in systems.

1.2 MULTIPROCESSING

Multiprocessing is increasingly seen as a viable approach to adding more processing capability to a system. Historically multiprocessing was used only in the very highest end computing systems and typically at great cost. However, the continuing advance of semiconductor process technology has made multi-processing a more mainstream technology and its use can offer advantages beyond higher processing performance.

Figure 1.2 is a photograph of a multiprocessing computer system. This computer system uses 76 microprocessors connected together with RapidIO to solve very complex signal processing problems.

Multiprocessing can also be used to reduce cost while achieving higher performance levels. Pricing of processors is often significantly lower for lower-speed parts. The use of multiprocessing may also reduce overall system power dissipation at a given performance point. This occurs because it is often possible to operate a processor at a reduced frequency and achieve a greatly reduced power dissipation. For example, the Motorola 7447 processor has a rated maximum power dissipation of 11.9 W at an operating frequency of 600 MHz.
The same processor has a maximum power dissipation of 50 W at an operating frequency of 1000 MHz\cite{1}. If the processing work to be done can be shared by multiple processors, overall power dissipation can be reduced. In this case reducing the frequency by 40% reduces maximum power dissipation by 76%. When performance per watt is an important metric, multiprocessing of lower performance processors should always be considered as part of a possible solution.

Operating system technology has also progressed to the point where multiprocessing is easily supported as well. Leading embedded operating systems such as QNX, OSE, and Linux are all designed to easily support multiprocessing in embedded environments.

While these previous obstacles to multiprocessing have been reduced or removed, the processor interconnect has increasingly become the main limiting factor in the development of multiprocessing systems. Existing multiprocessor bus technologies restrict the shared bandwidth for a group of processors. For multiprocessing to be effective, the processors in a system must be able communicate with each other at high bandwidth and low latency.

1.3 SYSTEM OF SYSTEMS

Traditional system design has busied itself with the task of connecting processors together with peripheral devices. In this approach there is typically a single processor and a group of
peripherals attached to it. The peripherals act as slave devices with the main processor being the central point of control. This master/slave architecture has been used for many years and has served the industry well.

With the increasing use of system on a chip (SoC) integration, the task of connecting processors to peripherals has become the task of the SoC developer. Peripheral functionality is increasingly being offered as part of an integrated device rather than as a standalone component. Most microprocessors or DSPs designed for the embedded market now contain a significant amount of integration. L2 cache, Memory controllers, Ethernet, PCI, all formally discrete functions, have found their way onto the integrated processor die. This integration then moves the system developer’s task up a level to one of integrating several SoC devices together in the system. Here a peer-to-peer connection model is often more appropriate than a master/slave model.

Figure 1.3 shows a block diagram of the Motorola PowerQUICC III communications processor. This processor offers a wealth of integrated peripheral functionality, including multiple Ethernet ports, communications oriented ATM and TDM interfaces, a PCI bus interface, DDR SDRAM memory controller and RapidIO controller for system interconnect functionality. It is a good example of a leading edge system on a chip device.

1.4 PROBLEMS WITH TRADITIONAL BUSES

The connections between processors and peripherals have traditionally been shared buses and often a hierarchy of buses (Figure 1.4). Devices are placed at the appropriate level in
PROBLEMS WITH TRADITIONAL BUSES

Figure 1.4  Higher system performance levels require adoption of point-to-point switched interconnects

the hierarchy according to the performance level they require. Low-performance devices are placed on lower-performance buses, which are bridged to the higher-performance buses so they do not burden the higher-performance devices. Bridging may also be used to address legacy interfaces.

Traditional external buses used on more complex semiconductor processing devices such as microprocessors or DSPs are made up of three sets of pins, which are soldered to wire traces on printed circuit boards. These three categories of pins or traces are address, data and control. The address pins provide unique context information that identifies the data. The data is the information that is being transferred and the control pins are used to manage the transfer of data across the bus.

For a very typical bus on a mainstream processor there will be 64 pins dedicated to data, with an additional eight pins for parity protection on the data pins. There will be 32–40 pins dedicated to address with 4 or 5 pins of parity protection on the address pins and there will be approximately another 30 pins for control signaling between the various devices sharing the bus. This will bring the pin count for a typical bus interface to approximately 150. Because of the way that semiconductor devices are built there will also be a large complement of additional power and ground pins associated with the bus. These additional pins might add another 50 pins to the bus interface pin requirement, raising the total pin count attributed to the bus alone to 200. This 200 pin interface might add several dollars to the packaging and testing cost of a semiconductor device. The 200 wire traces that would be required on the circuit board would add cost and complexity there as well. If the bus needed to cross a backplane to another board, connectors would need to be found that would bridge the signals between two boards without introducing unwanted noise, signal degradation and cost. Then, if you assume that your system will require the connection of 20 devices to achieve the desired functionality, you
begin to understand the role that the bus can play in limiting the functionality and feasibility of complex embedded systems.

For the sake of simplicity, we will discuss peak data bandwidth as opposed to sustained data bandwidth which is often quite a bit lower than peak. The peak data bandwidth of this bus would be the product of the bus frequency and the data bus width. As an example the PCI-X bus is the highest performance general purpose peripheral bus available. If we assume that we are operating the PCI-X bus at 133 MHz and using the 64-bit data path, then the peak data bandwidth is $133 \times 64 = 8.5 \text{ Gbit/s}$ or approximately $1 \text{ Gbyte/s}$.

To increase the performance of the interface beyond $1 \text{ Gbyte/s}$ we can either increase the frequency or we can widen the data paths. There are versions of PCI-X defined to operate at 266 and 533 MHz. Running at these speeds the PCI-X bus can support only one attached device.

When compared with the original bus interface on the Intel 8088 processor used by IBM in the first IBM PC, we find that available bus performance has increased significantly. The original 8088 processor had an 8 bit wide data bus operating at $4.77 \text{ MHz}$. The peak data bandwidth was therefore $419 \text{ Mbit/s}$ or $4.77 \text{ Mbyte/s}$. When compared with this bus the current PCI-X peripheral bus has widened by a factor of 8 and its signaling speed has increased by a factor of 28 for an overall improvement in peak bandwidth of approximately 2000%. Owing to improvements in bus utilization the improvement of actual bandwidth over the last 20 years has been even more dramatic than this, as has the improvement in actual processor performance.

While the growth in bus performance over the last several years has been impressive there are many indications that a new approach must be taken for it to continue. Here are four important reasons why this is the case.

1.4.1 Bus Loading

Beyond $133 \text{ MHz}$ it becomes extremely difficult to support more than two devices on a bus. Additional devices place capacitive loading on the bus. This capacitive loading represents electrical potential that must be filled or drained to reach the desired signal level, the additional capacitance slows the rise and fall time of the signals.

1.4.2 Signal Skew

Because a traditional bus is a collection of parallel wires with the signal valid times referenced to a clock signal, there are limits to how much skew can exist between the transition of the clock and the transition of a signal. At higher speeds the length of the trace as well as the signal transition times out of and into the devices themselves can affect the speed at which the bus is clocked. For a $133 \text{ MHz}$ bus the cycle time is 7.5 ns, the propagation delay in FR4 printed circuit board material is approximately 180 ps/ inch. For a quarter cycle of the bus (1875 ps) this would be approximately 10 inches of trace.

1.4.3 Expense of Wider Buses

The traditional solution of widening the buses has reached the point of diminishing returns. 128 bit wide data buses have not been well accepted in the industry, despite their use on
several processors. The wider buses further reduce the frequency at which the buses may run. Wider buses also increase product cost by increasing the package size and pin count requirements of devices. They may also increase system costs by forcing more layers in the printed circuit boards to carry all of the signal trace lines.

### 1.4.4 Problems with PCI

PCI is a very common peripheral bus used in computing systems. PCI plays a limited role in embedded systems for attachment of peripheral devices. PCI introduces several additional performance constraints to a system.

1. PCI doesn’t support split transactions. This means that the bus is occupied and blocked for other uses for the entire time a transaction is being performed. When communicating with slower peripheral devices this could be for a relatively long time.
2. The length of a PCI transaction isn’t known \textit{a priori}. This makes it difficult to size buffers and often leads to bus disconnects. Wait states can also be added at any time.
3. Transactions targeting main memory typically require a snoop cycle to assure data coherency with processor caches.
4. Bus performance is reduced to the least common denominator of the peripherals that are attached. Typically this is 33 MHz, providing peak transfer rates of only 266 Mbyte/s and actual sustained transfer rates less than 100 Mbyte/s.

### 1.5 THE MARKET PROBLEM

Among the products of the various companies that together supply electronic components to the embedded marketplace, there is no single ubiquitous bus solution that may be used to connect all devices together. Many vendors offer proprietary bus or interconnect technology on their devices. This creates a market for glue chips that are used to bridge the various devices together to build systems. Common glue chip technologies are ASIC and FPGA devices, with the choice of solution typically guided by economic considerations.

The number of unique buses or interconnects in the system increases the complexity of the system design as well as the verification effort. In addition to the device buses, system developers often also develop their own buses and interconnect technologies because the device buses do not offer the features or capabilities required in their systems.

The embedded market is different from the personal computer market. In the personal computer market there is a single platform architecture that has been stretched to meet the needs of notebook, desktop and server applications. The embedded equipment market is quite different in character from the personal computer market. It is also generally older with legacy telecommunications systems architectures stretching back forty years. The technical approaches taken for embedded equipment reflect the availability of components, historical system architectures and competitive pressures. The resulting system designs are quite different from that of a personal computer.

The embedded market also does not have as much dependency on ISA compatibility, favoring architectures such as ARM, PowerPC and MIPS as opposed to the X86 architecture that is predominant on the desktop.
Despite the disparate technical problems being solved by embedded equipment manufacturers and the variety of components and architectures available to produce solutions, there is still a desire for the use of standard interconnects to simplify the development task, reduce cost and speed time to market. This desire for standard embedded system interconnects is the primary impetus behind the development of the RapidIO interconnect technology.

1.6 RAPIDIO: A NEW APPROACH

The RapidIO interconnect architecture is an open standard which addresses the needs of a wide variety of embedded infrastructure applications. Applications include interconnecting microprocessors, memory, and memory mapped I/O devices in networking equipment, storage subsystems, and general purpose computing platforms.

This interconnect is intended primarily as an intra-system interface, allowing chip-to-chip and board-to-board communications with performance levels ranging from 1 to 60 Gbit/s performance levels.

Two families of RapidIO interconnects are defined: a parallel interface for high-performance microprocessor and system connectivity and a serial interface for serial backplane, DSP and associated serial control plane applications. The serial and parallel forms of RapidIO share the same programming models, transactions, and addressing mechanisms.

Supported programming models include basic memory mapped IO transactions; port-based message passing and globally shared distributed memory with hardware-based coherency. RapidIO also offers very robust error detection and provides a well-defined hardware and software-based architecture for recovering from and reporting transmission errors.

The RapidIO interconnect is defined as a layered architecture which allows scalability and future enhancements while maintaining backward compatibility.

1.6.1 Why RapidIO?

RapidIO is categorized as an intra-system interconnect as shown in Figure 1.5. Specifically, RapidIO is targeted at intra-system interconnect applications in the high-performance embedded equipment market. This market has distinct requirements when compared with the desktop and server computer spaces. The embedded market has historically been served by a number of different vendors. The openness of the RapidIO Trade Association is well suited to this environment. The RapidIO Trade Association counts among its current members nearly two dozen leading vendors of microprocessors, DSPs, FPGAs, ASICs and embedded memories.

InfiniBand is targeted as a System Area Network (SAN) interconnect. A SAN is used to cluster systems together to form larger highly available systems. SANs usually connect whole computers together within distances of up to 30 m. Operations through a SAN are typically handled through software drivers using message channels or remote direct memory access (RDMA). InfiniBand competes more directly with Fibre Channel, a storage area network technology and Ethernet-based system area networking technologies such as iSCSI.

HyperTransport and PCI Express share some common characteristics with RapidIO, but are more appropriately described as point-to-point versions of PCI. While they maintain compatibility with the PCI interconnect architecture from a software viewpoint, which is very
WHERE WILL IT BE USED?

important for desktop computer markets, they do not offer the scalability, robustness and efficiency required by embedded systems developers.

Ethernet has found opportunities as an intra-system interconnect in some applications. Ethernet at 100 Mbps and more recently at 1 Gbps is often integrated directly onto embedded microprocessors. The abundance of low-cost switching devices and relatively low pin count requirements make it an interesting choice for systems requiring a low-cost interconnect with modest (< 50 Mbyte/s) bandwidth requirements. At higher speeds, Ethernet becomes relatively less attractive in intra-system applications due to its higher software overhead, especially when used in conjunction with a TCP/IP protocol stack.

1.7 WHERE WILL IT BE USED?

The RapidIO interconnect is targeted for use in environments where multiple devices must work in a tightly coupled environment. Figure 1.6 illustrates a generic system containing memory controllers, processors, and I/O bridges connected using RapidIO switches.

In computing applications the PCI bus[2] is frequently used. Enterprise storage applications, for example, use PCI to connect multiple disk channels to a system. As disk throughput has increased so has the need for higher system throughput. To meet the electrical requirements of higher bus frequencies the number of supportable devices per bus segment must be decreased. In order to connect the same number of devices, more bus segments are required. These applications require higher bus performance, more device fan-out, and greater device separation. PCI-to-PCI bridge devices could be used to solve this problem, but only within a tree-shaped hierarchy that increases system latency and cost as more PCI devices are added to the system.
RapidIO can be used for transparent PCI-to-PCI bridging, allowing for a flattened architecture utilizing fewer pins with greater transmission distances. Figure 1.7 shows one such bridged system. In this example, several PCI-X bridges are connected together using RapidIO switches.

Many systems require the partitioning of functions into field replaceable units. These printed circuit boards have traditionally been interconnected using multi-drop interfaces such as VME or PCI. Higher system level performance can be achieved by using RapidIO. RapidIO is well suited for hot swap applications, because RapidIO’s point-to-point topology...
AN ANALOGY enables the removal of devices with little or no electrical impact to neighboring devices or subsystems.

RapidIO was developed by a group of leading embedded system and semiconductor developers. It was conceived as an open standard approach to device connectivity that solved the technical challenges posed by buses. The RapidIO Trade Association rather than any single company oversees the development of RapidIO technology. This is done to ensure that the technology will support the needs of the embedded market as a whole rather than the needs of a single company.

1.8 AN ANALOGY

The basic problem that RapidIO is designed to solve is the movement of data and control information between semiconductor devices within a relatively complex system. The RapidIO architecture is targeted at applications where there will be tens to hundreds, perhaps thousands of devices, all communicating with one another within a single system. It is optimized for this size of connectivity problem. It also assumes that efficiency is important. By this we mean that the overhead introduced by using RapidIO should be minimal. There are two dimensions of overhead. There are the extra bits of control information that accompany the data that is being sent across the interconnect. This includes the destination address of the data, the type of transaction that is intended, the error checking code that is used to validate that the data was received correctly and other important pieces of information. Overhead also includes the amount of work that needs to be done to send the data across the interconnect. The RapidIO controller might do this work, but the software that interacts with the RapidIO controller might also do it. In a system that includes hundreds or thousands of communication links, efficiency becomes an important consideration.

We have discussed RapidIO as a technology that replaces buses. We have also mentioned that Ethernet, a local area network (WAN) technology, may also be sometimes used as a bus replacement technology. The following analogy helps to understand the differences between RapidIO and other LAN and wide area network (WAN) communications technologies.

This analogy compares interconnect technologies to freight shipping technologies. SONET and ATM are used to move data (mostly encoded voice data for telephone calls), but these technologies may also be used to send Internet data as well, around the world. These technologies are similar to freight trains and container ships. They can efficiently move data long distances at relatively low cost, but require a fairly significant and rigid infrastructure to operate in. Ports and rail stations are expensive propositions, but they have the supporting information and technology infrastructure needed to ensure that the rail cars get to their proper destinations and that the container ships are properly loaded with their cargo.

Ethernet represents a LAN technology that relies on a more intelligent driver to get the payload to its proper destination. With Ethernet the intelligence resides more in the packet (at least the packet contains more information) and the infrastructure can be simpler to build and maintain. In this analogy, Ethernet is more akin to our modern highway system. While the connectivity of Ethernet, especially in conjunction with IP, might provide the ability to move anywhere in the country (at least) on common roads, economics and local regulations might require that things get broken up and bridged across other technologies such as ATM or SONET to move data longer distances.
You might ask, at this point where does RapidIO play? Which technology does it replace? The answer is: none of them. RapidIO in this analogy is much more like a forklift that would be used in a factory or distribution warehouse. In an effort to make warehouses more efficient, a group of warehouse and factory equipment vendors have gotten together to establish standards for moving material within warehouses. Hopefully you would agree that this problem is quite different from that of moving material between warehouses.

What has changed is that, in times past the factory and warehouse builders could deliver the best return on invested capital by building their own warehouses and material movement equipment. Now they believe that warehouse material movement technology is not a differentiating technology and they are better off buying it from suppliers whose business it is to build the best in-warehouse material movement technology possible. By establishing standards for this technology they get three additional benefits. First, they expand the market for the technology. This should reduce prices as economies of scale come in to play. Second, they create market competition, which should also reduce prices as well as increase quality. Third, they can ensure that products from different vendors will interoperate well, thus reducing their own engineering and development expenses.

In times past, monopolistic telecommunications vendors operating as vertically integrated companies provided all of the services and equipment needed for communications. They could easily afford to develop their own internal interconnect technologies, because they were the largest and perhaps only market for the technology. There were no economies of scale to be had by opening up the internal standards.

In this analogy we see that RapidIO is a complementary technology to technologies such as Ethernet and ATM. We also see that RapidIO is designed to solve a different class of interconnect problem in an area where no standards have previously existed.

On occasion, technologies designed to solve one problem are repurposed to solve a different problem. In the absence of RapidIO some vendors have resorted to using Ethernet to solve some of the in-system connectivity requirements of embedded equipment. For some applications this is a perfectly acceptable solution, for others it is horrendously inefficient. Depending on the problem being solved Ethernet or RapidIO may or may not be an acceptable solution.

REFERENCES

1. Motorola MPC7xxx microprocessor documentation.
2. PCI Local Bus Specification, Rev.2.2, PCI Special Interest Group 1999.