Requirements for high-performance embedded interconnects

By Dan Bouvier, Dave Wickliff, and Sam Fuller

High-performance embedded systems have been around for a long time. Our current telecommunications infrastructure, wireless infrastructure, enterprise storage, defense-related communications, and signal processing equipment are all examples of high-performance embedded systems. While each of these systems has its own unique requirements, they also share much commonality in their internal interconnect requirements. These requirements are often quite different from the requirements of the volume desktop computing space.

One very important attribute of high-performance embedded systems is the very high degree of connectivity required in the systems. Systems are typically based around backplane architectures with resource boards providing specialized functionality. Several system OEMs have used proprietary home grown interconnects for this application. During the last several years, the lower-end segments of the market have made use of the PCI interconnect architecture as a solution for backplane and peripheral connectivity. However, PCI initially developed as a desktop computer technology, has many deficiencies that make it unsuitable for use in many systems.

CompactPCI has become the popular industry-standard form-factor for the embedded use of PCI. Despite this popularity, CompactPCI suffers from limits on backplane performance, scalability, and reliability. As a result, CompactPCI's PCI bus interconnect has been supplemented by a variety of switch-based interconnects such as Ethernet and StarFabric. Similarly, the next-generation telecommunications oriented AdvancedTCA specifications defines a backplane that supports both dual star and full mesh connectivity between each board instead of a bused interconnect. The base AdvancedTCA specification defines mechanicals, board dimensions, power distribution, power and data connectors, and system management while remaining agnostic on the specific interconnect protocol between the boards. It is not clear that a single interconnect standard can provide all of the necessary functionality for all system applications. Several AdvancedTCA interconnect choices have been standardized (Ethernet and InfiniBand) and more are being developed (StarFabric and PCI Express).

We believe that another choice exists for CompactPCI and AdvancedTCA – RapidIO. RapidIO has broad support from both the leading embedded semiconductor vendors as well as embedded system equipment OEMs (see page 93). RapidIO has been developed, over a period of several years, by these industry leaders in an open standards forum. It has recently been submitted to the International Standards Organization (ISO) for approval as a formal International Standard for Embedded Systems Interconnect Technology. RapidIO is a proven technology, shipping in embedded systems today. RapidIO was architected with a forward-looking vision providing the capabilities that the embedded industry seeks and a platform to grow on for many years to come. This should be no surprise, as it was developed specifically to meet the unique and demanding requirements of the embedded market. The remainder of this article will examine some of those requirements and how RapidIO addresses them.

Direct peer-to-peer communications

High-performance embedded systems are typically made up of a variety of processing elements operating in a distributed computing fashion. The processing elements move data between each other using direct peer-to-peer communications. This is different from the typical personal computer where there is a grouping of peripherals that communicate with a common host-processor through a shared memory.

As the industry moves away from shared broadcast buses (like PCI) to point-to-point links like RapidIO, processing elements should be able to communicate directly with each other without the need for intervening bridges or host controllers. The most effective way to accomplish this is to use source directed addressing.

Source addressing means that a transaction contains the explicit address or device ID of the destination device. A transaction is routed to the destination through the use of routing tables in switches. This method allows traffic to be easily rerouted to an alternate path by either updating an entry in a routing table or modifying the target address at the sender to follow a different path. Table based routing was chosen for RapidIO because it carries with it the most flexibility with the least transaction overhead. In a memory-mapped system as is used in PCI Express, the device ID is just a portion of the overall system address. Explicitly defining the device ID limits the number of entries that a switch must compare when doing routing. PCI full address decode, on the other hand, forces a bridge device to do a match over an entire address width. RapidIO is the only next-generation intra-system interconnect that uses source routing. Memory mapping also forces a hierarchical tree topology that cannot be mapped efficiently into dual star or full or partial mesh fabrics.

Software compatibility

Embedded systems typically are built around a large software legacy. Further, the hardware and software development teams are usually separate. These factors make it very desirable to limit the impact to run-time code of new hardware architectures. A memory mapped approach where the interconnect is transparent to the driver level code limits the impact to the software. Technologies such as InfiniBand have a major impact on run-time software and are consequently disadvantaged in the embedded market. Technologies such as RapidIO, PCI Express, and HyperTransport – that transparently support memory-mapped operations including support for PCI bridging – do not have a major impact on existing software bases.

Messaging protocols

Embedded systems typically carry transient data, which moves from one processing element to another for further processing. Moving transient data using DMA is not practical since this requires full memory space visibility between the sending and receiving devices. A messaging protocol is better suited for this task. Ethernet or InfiniBand are examples of messaging protocols.

Reprinted from CompactPCI Systems / May-June 2003
©Copyright 2003

The disadvantage of Ethernet and InfiniBand is the large transaction overhead required to accommodate the large networks and data security that are required for inter-system communications. This overhead becomes less burdensome for large data payload messages moving across large networks, but can severely impact performance for short payload inter-processor communications. An example of the sorts of small payloads seen in embedded systems are UDP packets (< 64 bytes) used to carry voice traffic between DSPs for an IP telephony application or the use of AAL2 based ATM cells (< 53 bytes) for carrying packet voice in a 3G base station. RapidIO is the only next-generation intra-system interconnect with a low overhead messaging protocol.

Robust protocol with precise error detection and recovery

Most embedded systems must be robust. They must be able to recover from transient errors without outside or software intervention. The higher data rates of next-generation interconnects and the proliferation of interconnects increases the probability and frequency of errors. Next-generation protocols are source synchronous and therefore carry a weak handshake between adjacent devices. Therefore, an interconnect protocol for embedded systems must be designed for reliable transactions. This means that a transaction must not be dropped at any place in the network and transmission errors should be detected on a link-by-link basis. Because these systems are typically working on real time data flows, error recovery should be managed in hardware to reduce recovery latency and simplify software design. RapidIO is the only interconnect with link-by-link error detection and recovery in hardware. The RapidIO interconnect provides the means for a system to precisely detect, correct, recover from and report a dropped packet for any transaction in the system.

Scalable, flexible topology support, and sophisticated system discovery

The embedded OEM may want to connect many processors in a system. For example, a DSP farm module may contain 24 DSPs. Several of these farms may be included in a system. It is possible that in a large system thousands of device IDs may be necessary. Additionally a system may require redundant transaction paths and multiple hosts in order to manage failed links.

This means that the topology of a system may be more complex than the simple tree or daisy chain found in the typical PC system. For example, there may be more than one path between two endpoints. System discovery software must have the ability to walk through a system and enumerate devices in a complex topology. Further, for reasons of redundancy, it is required that more than one host be able to discover the system concurrently.

RapidIO allows a small or large transport address supporting 64 or 256K devices respectively. Each RapidIO device contains maintenance registers that are used by a system host to discover, enumerate and configure a device in any arbitrary topology.

Module and backplane connectivity

Embedded systems are becoming more and more modular. Modularity allows the system OEM to build common processing modules to be deployed across multiple applications, thereby increasing volume and lowering cost. The I/O technology used must have the ability to span distance and connectors. This means parallel interfaces must be capable of communication between mezzanine boards. Serial links must be capable of communicating over backplanes and possibly between chassis over cable.

The RapidIO parallel physical layer uses LVDS, which can cross mezzanine board connectors. Serial RapidIO can sustain longer transmission distances and greater inter-signal skew for backplane and cabled applications.

Hot swap

Most embedded systems do not allow for down time to replace or upgrade field replaceable units. A protocol must be able to withstand both graceful and ungraceful hot swap. A protocol should be able to recognize loss of communications with its partner even while idle. A RapidIO link carries a constant stream of state information allowing a device to immediately detect when its link partner is no longer present or operating correctly as might occur when a board is suddenly removed. For a more graceful hot swap, software can quiesce a device through RapidIO using defined maintenance transactions.

Conclusions

High-performance embedded systems have very unique interconnect requirements. The volume desktop computer market does not share these requirements, for the most part. This article has examined some of the more important requirements for the embedded market and how RapidIO, a widely adopted and supported standard for system-level interconnect meets those requirements. Ask your semiconductor supplier about its plans for RapidIO.

Dan Bouvier *is PowerPC Architecture Manager at Motorola SPS.*

Dave Wickliff *is Technology Strategist at Lucent Technologies.*

Sam Fuller *is President of the RapidIO Trade Association.*

For more information, contact Sam at:

Sam Fuller
RapidIO Trade Association
3925 W. Braker Lane
Suite 325
Austin, TX 78759
Tel: 512-305-0070
Fax: 512-305-0009
E-mail: sam.fuller@rapidio.org
Web site: www.rapidio.org

©Copyright 2003