Using Serial RapidIO for FPGA co-processing

Navneet Rao, Xilinx - November 08, 2007

For more on Serial RapidIO, see The RapidIO High-Speed Interconnect: A Technical Overview, Easy multiprocessor design with sRIO and MSGQ, and Partitioning video across multiple DSPs and FPGAs.

Today's ever increasing demand for high-speed communication and super-fast computing in support of "triple-play" applications is creating new challenges for system developers, algorithm developers and hardware engineers alike who need to draw together a multitude of standards, components and networking equipment. At the same time, developers need to keep pace with increasing demands for performance while keeping costs low. These feats can be accomplished by leveraging Serial RapidIO-enabled FPGAs as DSP co-processors.

Because triple-play applications unite voice, video and data, development and system optimization strategies must be parameterized using newer algorithms. Specific challenges that developers need to address include building scalable and extensible architectures, supporting distributed processing, using standards-based design, and optimizing for performance and cost.

A closer look at these challenges reveals two themes: Connectivity—which is essentially "fast" data movement across devices, boards and systems—and Computing power—i.e., the individual processing resources that are available in the devices, boards and systems—address the needs of the application.

**Connectivity across compute platforms**
Standards-based designs are usually much easier than "roll your own" designs, and are the norm of the day. Parallel connectivity standards (PCI, PCI-X, EMIF, etc) can meet today's demands, but fall short when scalability and extensibility are taken into consideration. With the advent of packet-based processing, the trend is clearly towards high-speed serial connectivity. Figure 1 illustrates this trend.
High speed serial standards like PCIe and GbE/XAUI have been adopted in the desktop and networking industry. Meanwhile, data processing systems in wireless infrastructure have slightly different interconnect requirements:

- Low pin count
- Backplane and chip-to-chip connectivity
- Bandwidth and speed scalability
- DMA and message passing
- Support for complex scalable topologies
- Multicast
- High reliability
- Time of day synchronization
- Quality of Service (QoS)

The Serial RapidIO (SRIO) protocol standard can easily meet and exceed most of these requirements and has become the dominant interconnect for data-plane connectivity in wireless infrastructure equipment. SRIO networks are built around two "Basic Blocks" - Endpoints and Switches. Endpoints source and sink packets, while Switches pass packets between ports without interpreting them. Figure 2 shows SRIO network building blocks.
Serial RapidIO is specified as a 3-layer architectural hierarchy, illustrated in Figure 3. It has the following elements:

Physical Layer—describes the device-level interface specifics such as packet transport mechanisms, flow control, electrical characteristics, and low-level error management.

Transport Layer—provides routing information for moving packets between endpoints. Switches operate at the transport layer by using device-based routing.

Logical Layer—defines the overall protocol and packet formats. All packets contain 256 payload bytes or less. The transactions use Load, Store, or DMA operations targeting a 34 / 50 / 66-bit address space.

For detailed information on the RapidIO specifications, please refer to: http://www.rapidio.org/specs/
latency and increased system bandwidth relative to other bus interfaces. But unlike most other bus interfaces, SRIO has low pin count interfaces and scalable bandwidth based on high speed serial links, which can scale from 1.25 - 3.125 Gbps. Figure 4 illustrates the SRIO specification.

![SRIO specification](Click to enlarge)
Figure 4. SRIO specification

**Computing resources in platforms:**
With the availability of configurable processing resources, developers are implementing applications in hardware. For example, data compression and encryption algorithms, even complete firewall and security applications, which were previously implemented in software, are now implemented in hardware. These hardware implementations demand a massive parallel ecosystem of shared bandwidth and processing power. They require shared or distributed processing through CPUs, NPUs, FPGAs, and/or ASICs. Some of the computing resource requirements for building such a system include:

- Distributed processing supporting complex topologies
- Direct peer-to-peer communication with high reliability
- Multiple heterogeneous OS’s
- Ability to support communications data plane using multiple heterogeneous OS’s
- Availability of modular and extendable platforms that have broad ecosystem support

The SRIO protocol was architected and specified to support the disparate requirements of compute devices in the embedded and wireless infrastructure space. SRIO makes it possible to achieve architectural independence, the ability to deploy scalable systems with carrier grade reliability, advanced traffic management, and provisioning for high performance and throughput. In addition, a broad ecosystem of vendors makes it easy to build SRIO systems with off-the-shelf components. SRIO is a packet-based protocol that supports:

- Data movement using packet-based operations (read, write, message)
- I/O non-coherent functions and cache coherence functions

- Efficient interworking and protocol encapsulation through support for data streaming, and SAR functions
- A traffic management framework, by enabling millions of streams, support for 256 traffic classes,
and lossy operations

- Flow control to support for multiple transaction request flows including provision for QoS
- Priorities support to alleviate problems like bandwidth allocation, transaction ordering, and deadlock avoidance
- Topology support for standard (trees and meshes) and arbitrary (daisy-chains) hardware topologies through system discovery, configuration and bring-up including support for multiple hosts
- Error management and classification (recoverable, notification and fatal)

**IP Solutions for Serial RapidIO**

To support fully compliant maximum-payload operations for both sourcing and receiving user data through target and initiator interfaces on the Logical (I/O) and Transport Layer IP, vendors like Xilinx offer endpoint IP solutions for Serial RapidIO designed to the latest RapidIO Specification v1.3.

The complete Xilinx Endpoint IP solution for SRIO is shown in Figure 5. It consists of the following components:

1. LogiCORE RapidIO Logical (I/O) and Transport Layer IP
2. Buffer Layer Reference Design
3. LogiCORE Serial RapidIO Physical Layer IP
4. Register Manager Reference Design

![Figure 5. Xilinx Endpoint IP architecture for SRIO](image)

**The Serial RapidIO IP**

**The IP**

Xilinx provides the Buffer Layer Reference Design as source-code that performs automatic packet re-prioritization and queuing. The SRIO Physical Layer IP implements link training and initialization, discovery and management, and error and retry recovery mechanisms. Additionally, high-speed transceivers are instantiated in the Physical Layer IP to support 1- and 4-lane SRIO bus links at line rates of 1.25Gbps, 2.5Gbps, 3.125Gbps.

A Register Manager reference design enables the SRIO host device to configure and maintain endpoint device configuration, link status, control, and time-out mechanisms. In addition, ports are provided on the Register Manager for the user-design to probe the status of the endpoint device.

LogiCORE provides complete endpoint IP. It has been tested by leading SRIO device vendors. LogiCORE is delivered through the Xilinx CoreGen GUI tool, which allows users to configure the baud-rates and endpoints. It supports extended features like flow-control, re-transmit suppression, doorbell and messaging. This enables the user to create a flexible, scalable and customized SRIO endpoint IP optimized to the needs of the application. Further details can be found at [http://www.xilinx.com/rapidio](http://www.xilinx.com/rapidio).
Using the varied resources available in most high-performance FPGAs from Xilinx and other vendors, a system designer can easily create and deploy intelligent solutions to harness advantage scenarios like time-to-market, scalability and extensibility, future-proofing, and so forth. Below we outline some system design examples using SRIO and DSP technologies.

**A SRIO System Application I: Embedded System**

CPU architectures such as the x86 are optimized for general-purpose applications that do not require extensive use of multiplication. In contrast, DSP architectures are optimized for signal-processing operations including filtering, FFTs, vector multiplication and searching, and image or video analysis.

Embedded systems that use CPUs and DSPs can easily be architected to take advantage of both general-purpose and signal processing. An example of such a system is outlined in Figure 6. It features FPGAs, CPUs and DSP architectures.

![Figure 6. CPU-based high-performance DSP sub-system](Click to enlarge)

Serial RapidIO has become the primary data interconnect for high-end DSPs. In x86 CPUs, the primary data interconnect is PCI Express. As shown in Figure 6, FPGAs can easily be deployed for scaling the DSP application, and/or for bridging disparate data interconnect standards like PCI Express and Serial RapidIO.

In the system depicted here, the PCI Express system is hosted by the Root Complex chipset. The SRIO system is hosted by a DSP. The 32/64-bit PCIe address space (base-address) can be intelligently mapped to the 34/66-bit SRIO Address space (base-address). The PCIe application communicates with the Root Complex through Memory or I/O Reads and Writes. These transactions can be easily mapped to SRIO space through I/O operations such as streaming writes, atomic and acknowledged read/write transactions (SWRITEs, ATOMIC, NREADs, NWRITE/NWRITE_Rs).

Designing such bridge functions is easy in Xilinx FPGAs, since the back-end interfaces for the PCI Express and Serial RapidIO endpoint functional blocks are similar. The Packet Queue block can then perform the crossover from PCIe to SRIO or vice-versa to establish the data flow between these two protocol domains.

**A SRIO System Application II: DSP processing application**
In applications where DSP processing is the primary architectural requirement, the system architecture can be designed as depicted in Figure 7.

Xilinx Virtex-5 FPGAs can act as co-processors to other DSP devices in the system. The complete DSP system solution can be scaled easily if SRIO is used as the data interconnect. These solutions can be future-proofed, made extensible, and be supported across multiple form-factors.

If the DSP-intensive application additionally requires fast number-crunching or data processing, such processing can be offloaded to x86 CPUs. Xilinx Virtex-5 FPGAs allow the PCIe sub-system and the SRIO architecture to be bridged to enable efficient offloading of functionality.

A SRIO System Application III: Baseband processing system

With 3G networks maturing rapidly, OEMs are deploying new form-factors to alleviate capacity and coverage problems FPGA-based DSP architectures using SRIO are the ideal solution to such challenges. Legacy DSP systems can also be retargeted to such fast, low-power FPGA-based architectures to harness FPGA’s scalability advantage.

In such a system, as depicted in Figure 8, FPGAs can meet the demands of line-rate processing of antenna traffic and also provide connectivity to other system resources through SRIO. Migration of existing legacy DSP applications, which have inherently slow parallel connectivity, is easy thanks to the high speed and bandwidth provided by the Serial RapidIO protocol.
Summary:
Serial RapidIO is appearing in a wide array of new applications, largely centered around DSPs in wired and wireless applications. The key advantages of implementing Serial RapidIO in Xilinx devices are:

- Availability of complete SRIO endpoint solution
- Flexibility and scalability to produce different classes of products with the same hardware and software architecture
- Low power with new GTP transceivers and 65nm technologies
- Easy configurability and flexibility through the CoreGen GUI tool
- Proven hardware interoperability with leading industry vendors supporting SRIO connectivity on their devices
- Lower overall system cost by achieving system integration through use of integrated IO blocks like PCIe and TEMAC.

In addition, FPGAs such as Xilinx's Virtex-5 can meet the power, performance and bandwidth requirements of legacy DSP applications. Additional benefits accrue in terms of system integration, due to the availability of functional blocks like Ethernet MACs, endpoint blocks for PCIe functions, processor IP blocks, memory elements and controllers. The exhaustive list of IP Cores that are supported also means that significant overall system cost savings can be realized.

About the author:

Previously he led teams in architecting and designing Transceivers, Switch Fabric ASICs at Mindspeed Technologies (aka Hotrail, Inc). Navneet has excellent experience having worked in product development teams at Philips Semiconductors developing a communication ASICs for basestation platforms and at LSI Logic designing microcontrollers.

Navneet is also an active member in trade associations like FSA, RapidIO, PCI Express, and HyperTransport. He has been an invited speaker at a number of seminars with industry experts including Embedded Systems Conference, ATCA Summit, Linley Group, RapidIO TA, National’s Analog by Design.

Navneet has 13 years industry experience after completing his MSEE from Indian Institute of Technology, Kharagpur, India.