



# Xcell journal

SOLUTIONS FOR A PROGRAMMABLE WORLD

## Inside Look: Connectivity Illuminated

### INSIDE

PCI Express and FPGAs

Reducing CPU Load for  
Ethernet Applications

A High-Speed Serial  
Connectivity Solution  
with Aurora IP

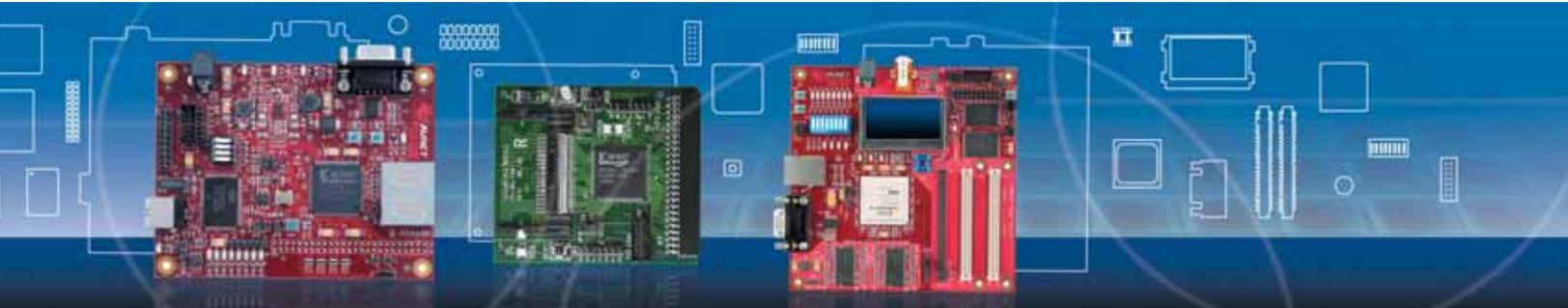
Xilinx FPGAs Adapt to  
Ever-Changing Broadcast  
Video Landscape

Making the Most of  
MOST Control Messaging



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

# Support Across The Board.™



## Design Kits Fuel Feature-Rich Applications

### Build your own system by mixing and matching:

- Processors
- FPGAs
- Memory
- Networking
- Audio
- Video
- Mass storage
- Bus interface
- High-speed serial interface

### Available add-ons:

- Software
- Firmware
- Drivers
- Third-party development tools

Avnet Electronics Marketing designs, manufactures, sells and supports a wide variety of hardware evaluation, development and reference design kits for developers looking to get a quick start on a new project.

With a focus on embedded processing, communications and networking applications, this growing set of modular hardware kits allows users to evaluate, experiment, benchmark, prototype, test and even deploy complete designs for field trial.

By providing a stable hardware platform that enhances system development, design kits from Avnet Electronics Marketing help original equipment manufacturers (OEMs) bring differentiated products to market quickly and in the most cost-efficient way possible.

For a complete listing of available boards, visit [www.em.avnet.com/drc](http://www.em.avnet.com/drc)



*Enabling success from the center of technology™*



1 800 332 8638  
[em.avnet.com](http://em.avnet.com)

# IS YOUR CURRENT FPGA DESIGN SOLUTION HOLDING YOU BACK?



**FPGA Design** | Ever feel tied down because your tools didn't support the FPGAs you needed? Ever spend your weekend learning yet another design tool? Maybe it's time you switch to a truly vendor independent FPGA design flow. One that enables you to create the best designs in any FPGA. Mentor's full-featured solution combines design creation, verification, and synthesis into a vendor-neutral, front-to-back FPGA design environment. Only Mentor can offer a comprehensive flow that improves productivity, reduces cost and allows for complete flexibility, enabling you to always choose the right technology for your design. To learn more go to [mentor.com/techpapers](http://mentor.com/techpapers) or call us at 800.547.3000.

DESIGN FOR MANUFACTURING + INTEGRATED SYSTEM DESIGN  
ELECTRONIC SYSTEM LEVEL DESIGN + FUNCTIONAL VERIFICATION

**Mentor  
Graphics**<sup>®</sup>  
THE EDA TECHNOLOGY LEADER

# INNOVATION

Leadership above all...



*Xilinx brings your biggest ideas to reality:*

**Virtex™-5 FPGAs — Highest Performance.** With multiple platforms optimized for logic, serial connectivity, DSP, and embedded processing, Virtex-5 FPGAs lead the industry in performance and density.

**Spartan™-3 Generation FPGAs — Lowest Cost.** A unique balance of features and price for high-volume applications. Multiple platforms allow you to choose the lowest cost device to fit your specific needs.

**CoolRunner™-II CPLDs — Lowest Power.** Unbeatable for low-power and handheld applications, CoolRunner-II CPLDs deliver more for the money than any other competitive device.

**ISE™ Software — Ease-of-Design.** With SmartCompile™ technology, users can achieve up to 6X faster runtimes, while preserving timing and implementation. Highest performance in the fastest time — the #1 choice of designers worldwide.

Visit our website today, and find out why Xilinx products are world renowned for leadership... *above all.*

*Get started quickly with easy-to-use kits*



Order online at [www.xilinx.com/getstarted](http://www.xilinx.com/getstarted)



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

***At the Heart of Innovation***

## Xcell journal

PUBLISHER	Forrest Couch forrest.couch@xilinx.com 408-879-5270
EDITOR	Charmaine Cooper Hussain
ART DIRECTOR	Scott Blair
DESIGN/PRODUCTION	Teie, Gelwicks & Associates 1-800-493-5551
ADVERTISING SALES	Dan Teie 1-800-493-5551
TECHNICAL COORDINATOR	Alex Goldhammer
INTERNATIONAL	Dickson Seow, Asia Pacific dickson.seow@xilinx.com  Andrea Barnard, Europe/ Middle East/Africa andrea.barnard@xilinx.com  Yumi Homura, Japan yumi.homura@xilinx.com
SUBSCRIPTIONS	All Inquiries www.xcellpublications.com
REPRINT ORDERS	1-800-493-5551



www.xilinx.com/xcell/

Xilinx, Inc.  
2100 Logic Drive  
San Jose, CA 95124-3400  
Phone: 408-559-7778  
FAX: 408-879-4780  
www.xilinx.com/xcell/

© 2007 Xilinx, Inc. All rights reserved. XILINX, the Xilinx Logo, and other designated brands included herein are trademarks of Xilinx, Inc. All other trademarks are the property of their respective owners.

The articles, information, and other materials included in this issue are provided solely for the convenience of our readers. Xilinx makes no warranties, express, implied, statutory, or otherwise, and accepts no liability with respect to any such articles, information, or other materials or their use, and any use thereof is solely at the risk of the user. Any person or entity using such information in any way releases and waives any claim it might have against Xilinx for any loss, damage, or expense caused thereby.

# Can You Hear Me Now?

We've all experienced it with our mobile phones: poor voice quality, dropped calls, not enough "bars" to connect. Although inconvenient, the problem usually goes away by moving around or waiting a few minutes before trying your call again.

But if you're a system designer, connectivity problems launch a multi-dimensional conversation that's quite a bit more complicated to solve. Let's start with your target market. Is your product a solution for the aerospace/defense, automotive, broadcast, consumer, server/storage, industrial/scientific/medical, wired, or wireless market? What are the design challenges in that market, such as signal integrity, power management, area, and performance? In addition to these decisions, you must select the proper protocol for the unique components of your system hierarchy, such as the ubiquitous PCI Express at one end and Gigabit Ethernet at the other.

Connectivity, in its myriad of forms, is an increasingly important factor in successful system design. Whether designing chip-to-chip, board-to-board, or box-to-box, Xilinx has a connectivity solution that helps you differentiate your product within your target market, bring it to market as early as possible and at the lowest cost. With our connectivity hard blocks, you get the benefits of an ASSP as well as the flexibility of an FPGA.

This issue of *Xcell Journal* focuses on connectivity and brings light to some of the key issues, as well as presenting implementation examples. Additionally, Xilinx offers a variety of resources to help with your connectivity design challenges. Here are a few connectivity resources that may help.

## Connectivity Central

([www.xilinx.com/products/design\\_resources/conn\\_central/](http://www.xilinx.com/products/design_resources/conn_central/))

Xilinx provides end-to-end connectivity solutions supporting serial and parallel protocols. Design for success with Virtex™ and Spartan™ series FPGAs, IP cores, tools, and development kits.

## Virtex-5 LXT FPGA Development Kit for PCI Express

([www.xilinx.com/xlnx/xebiz/designResources/ip\\_product\\_details.jsp?key=HW-V5-ML555-G](http://www.xilinx.com/xlnx/xebiz/designResources/ip_product_details.jsp?key=HW-V5-ML555-G))

The Virtex-5 LXT FPGA Development kit for PCI Express supports PCIe/PCI-X/PCI. This complete development kit passed PCI-SIG compliance for PCI Express v1.1 and enables you to rapidly create and evaluate designs using PCI Express, PCI-X, and PCI interfaces.

## Spartan-3 PCI Express Starter Kit

([www.xilinx.com/onlinestorespartan\\_boards.htm](http://www.xilinx.com/onlinestorespartan_boards.htm))

This complete development board solution gives you instant access to the capabilities of the Spartan-3 family and the Xilinx® PCI Express core.

## IP Evaluation

([www.xilinx.com/ipcenter/lpevaluation/index.htm](http://www.xilinx.com/ipcenter/lpevaluation/index.htm))

Take advantage of Xilinx evaluation IP to "try before you buy." Visit the Xilinx IP Evaluation Lounge and download the evaluation libraries using the links under the "Related Products" tab on the right side of the page. The IP on this site is covered by the Xilinx LogiCORE™ Evaluation License Agreement.

## On-Site Help

([www.xilinx.com/xlnx/xebiz/designResources/ip\\_product\\_details.jsp?key=xgs\\_ss\\_titanium](http://www.xilinx.com/xlnx/xebiz/designResources/ip_product_details.jsp?key=xgs_ss_titanium))

For help with your connectivity designs, Titanium Technical Service provides a dedicated application engineer, either on-site or at Xilinx, who can help with system architecture optimization, tool coaching, and back-end optimization.

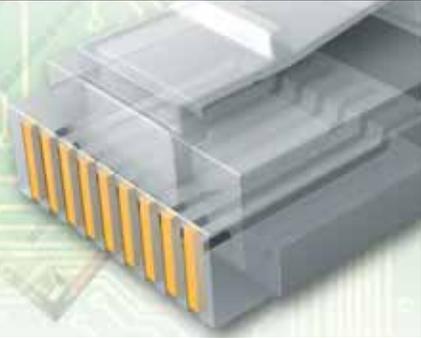


*Forrest Couch*

Forrest Couch  
Publisher

CONNECTIVITY

20



**Reducing CPU Load for Ethernet Applications**  
A TOE makes 10 Gigabit Ethernet possible.

CONNECTIVITY

31



**Xilinx FPGAs Adapt to Ever-Changing Broadcast Video Landscape**  
New digital broadcast standards and applications are addressed by advanced silicon, software, and free reference designs available with Xilinx FPGAs.

CONNECTIVITY

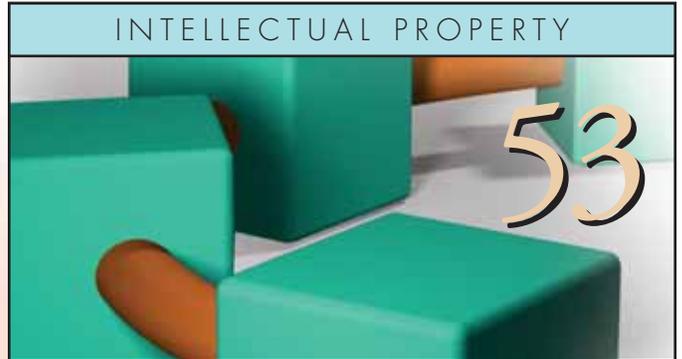
26



**A High-Speed Serial Connectivity Solution with Aurora IP**  
Aurora is a highly scalable protocol for applications requiring point-to-point connectivity.

INTELLECTUAL PROPERTY

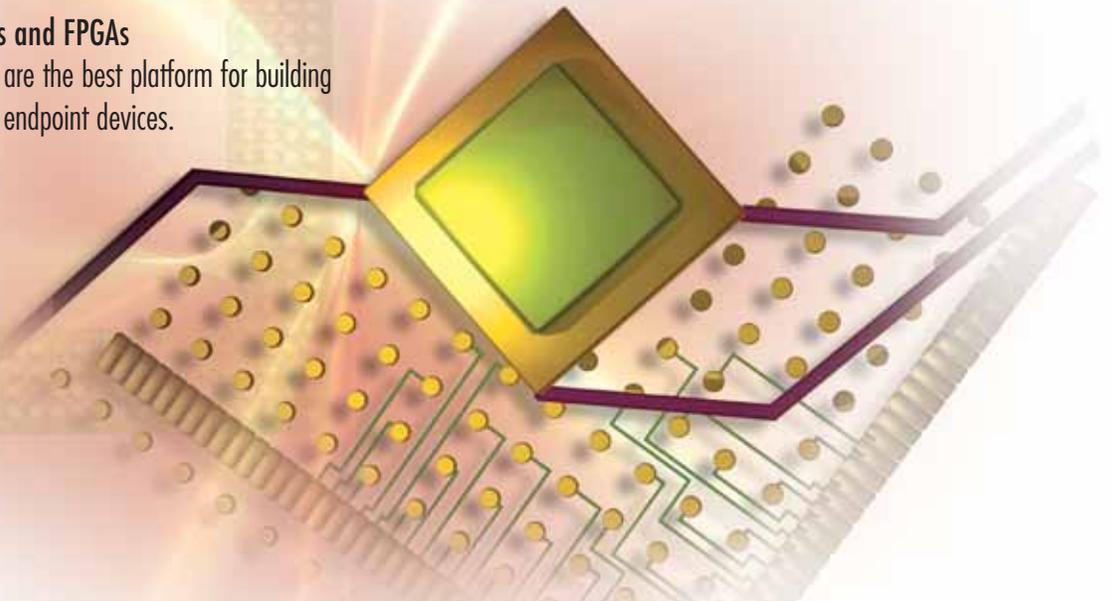
53

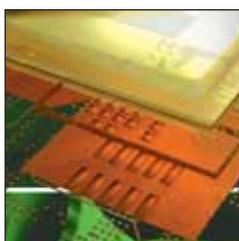
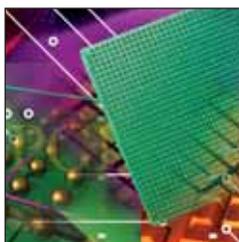
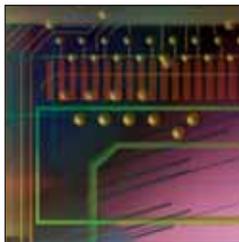
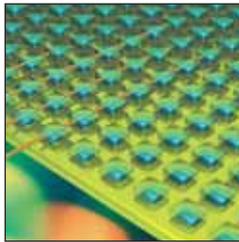


**Making the Most of MOST Control Messaging**  
This case study presents the design of Xilinx LogiCORE MOST NIC control message processing.

14

**PCI Express and FPGAs**  
Why FPGAs are the best platform for building PCI Express endpoint devices.





## VIEWPOINTS

Letter from the Publisher . . . . .	5
Selecting the Right Interconnect . . . . .	8

## FEATURES

### Connectivity

Scaling Chip-to-Chip Interconnect Made Simple . . . . .	11
PCI Express and FPGAs . . . . .	14
Virtex-5 FPGA Techniques for High-Performance Data Converters . . . . .	18
Reducing CPU Load for Ethernet Applications . . . . .	20
Automated MGT Serial Link Tuning Ensures Design Margins . . . . .	23
A High-Speed Serial Connectivity Solution with Aurora IP . . . . .	26
Xilinx FPGAs Adapt to Ever-Changing Broadcast Video Landscape . . . . .	31
Serial RapidIO Connectivity Enhances DSP Co-Processing . . . . .	36
The NXP/PLDA Programmable PCI Express Solution . . . . .	42

### Memory Interface

Create Memory Interface Designs Faster with Xilinx Solutions . . . . .	47
--	----

### Intellectual Property

Driving Home Multimedia . . . . .	50
Making the Most of MOST Control Messaging . . . . .	53
Leveraging HyperTransport on Xilinx FPGAs . . . . .	56

## GENERAL

FPGA-Based Simulation for Rapid Prototyping . . . . .	60
---	----

## RESOURCES

Featured Connectivity Application Notes . . . . .	62
Connectivity Boards and Kits . . . . .	63
The Connectivity Curriculum Path . . . . .	64

## XPECTATIONS

Developing Technical Leaders through Knowledge Communities . . . . .	66
--	----



# Selecting the Right Interconnect

Because interconnect characteristics vary, define your requirements clearly before selecting the appropriate interconnect.



by Jag Bolaria  
Sr. Analyst  
The Linley Group  
[jag@linleygroup.com](mailto:jag@linleygroup.com)

Interconnects have evolved from parallel to serial and increased in complexity

to enable communications with greater efficiency and less congestion or hot points. In addition to connecting endpoints, modern interconnects define comprehensive protocols for moving data efficiently across a network of endpoints.

Thus, the networks and endpoints that must be interconnected often drive the requirements for an interconnect. These requirements include data rate, latency, lossy or lossless links, scalability, and redundancy. These requirements then drive the selection of the appropriate interconnects for a specific network.

The existing ecosystem for an interconnect technology is another important factor in selection. A good ecosystem can help reduce development cost and time to market. In this article, I'll look at some of the leading interconnects and position those for specific applications or market segments. The Linley Group's report on High-Speed Interconnects, available at [www.linleygroup.com](http://www.linleygroup.com), provides more details on various interconnects and the leading products for each.

## PCIe and Ethernet

Our research shows that the leading interconnects are driven by large-volume platforms. The economies of scale from larger volume platforms ensure low-cost building blocks and broad availability. Additionally, large deployments lead to field-proven technologies that can be applied in other platforms with minimal risk.

Two of the largest platforms are PCs and networking equipment. The PC platform drives PCI Express (PCIe) and Ethernet, while networking equipment drives only Ethernet. But because these interconnects were developed for specific applications, they are not a natural fit in many other markets. The semiconductor industry and system vendors are evolving these interconnects to meet requirements for new applications.

For example, PCIe scalability has evolved to support greater data rates and more lanes. With IOV (I/O virtualization), PCIe is evolving to support virtualization, which enables its deployment in storage systems and blade servers.



With IEEE802.3ar and BCN (backward congestion notification), Ethernet enhancements include better flow control, congestion management, and attempts to address its inherently lossy nature. These enhancements will strengthen Ethernet applicability for storage systems, data centers, and backplanes.

Although Ethernet and PCIe are now suitable for more applications, they still fall short in meeting the technical and business requirements for all systems. Blade servers, for example, use a combination of Ethernet and Fibre Channel (FC). Although OEMs may want to consolidate these fabrics, end users have a large investment in FC and want support for that now and in the future.

PCIe and Ethernet also fall short in meeting the scalability, latency, and lossless requirements of high-performance computing (HPC) applications. HPC uses a specialized interconnect such as InfiniBand, which provides better latency and scalability. In this case, OEMs will need flexible interconnect solutions to enable common platforms and thus service different user requirements.

## Both dominant and specialized interconnects will continue to evolve to support greater data rates, reduced latency, and better scalability.

### Everyone Else

Endpoints and specific system requirements often drive the development of specialized interconnects. RapidIO is one such example. Steered by system and chip vendors, RapidIO has evolved to address the unique requirements of the wireless infrastructure. It enables distributed computing on line cards and networking/wireless infrastructure systems better than most competing interconnects. RapidIO is also integrated on DSPs from Texas Instruments and PowerPC CPUs from Freescale.

Because base stations use farms of DSPs, it is an easy decision to use RapidIO as an interconnect in these applications. Over time, we expect RapidIO to expand to other platforms that perform digital signal processing on multiple data streams.

Examples of other specialized interconnects include XFI, SFI, XAUI, SPAUI, Interlaken, SPI-S, and KR. These interconnects were developed to address the very specific low-level requirements of each application. Although addressing all interconnects is beyond the scope of this article, let's look at a few to highlight the problems each solves and its impact in systems.

XFI and SFI are used to connect optical modules at 10 Gbps. At these data rates, the major challenge is signal conditioning, including electronic dispersion compensation for the fiber and equalization for the board traces and connectors. These requirements drive specialized components designed specifically for the characteristics of the channel through which the signal travels.

Because data at these rates may be channeled – that is, include multiple streams on a single physical link – it becomes important to add traffic management. Specifications such as Interlaken, SPI-S, and SPAUI address high data rates as well as traffic management. Because no single standard exists, we believe system designers

need to design in solutions that provide flexibility in meeting current and future requirements.

The combination of 10 Gbps rates on the network and multiport line cards drives the need for greater bandwidth and therefore greater data rates over the backplane. The IEEE 802.3ap addresses this with its 10GBase-KR specification, which defines 10-Gbps serial links. In addition to equalization and pre-emphasis, it may be necessary to include forward error correction for acceptable performance over a couple of connectors and up to 40-inch traces common in backplanes. Additionally, these systems may need compatibility to older line cards, driving the need for backplane operation at 1 Gbps or 3.125 Gbps. Again, a flexible solution is critical to meet the system requirements.

### Conclusion

There are many different applications for interconnects and many interconnect choices for system designers. We expect PCIe and Ethernet to be the dominant interconnects. These will be used in servers, networking, storage systems, wireless networks, and many other systems. There is, however, no one interconnect (or two) that can meet the requirements of all systems. Therefore, the industry has developed and will continue to support specialized interconnects for different applications.

Both dominant and specialized interconnects will continue to evolve to support greater data rates, reduced latency, and better scalability. Additionally, systems will need to support legacy cards.

We recommend that system designers select the best interconnect and design in flexibility to cover different interconnects and the evolving changes in each. FPGAs play a critical role in offering system designers this flexibility and supporting the broad interconnect landscape. ●●

## XILINX EVENTS

Xilinx participates in numerous trade shows and events throughout the year. This is a perfect opportunity to meet our silicon and software experts, ask questions, see demonstrations of new products and technologies, and hear other customers' success stories with Xilinx products.

For more information and the most up-to-date schedule, visit [www.xilinx.com/events](http://www.xilinx.com/events).

### North America

**July 23-27**

Nuclear and Space Radiation Effects Conference 2007  
Honolulu, HI

**August 7-9**

NI Week 2007  
Austin, TX

**September 18-20**

Intel Developer Forum  
San Francisco, CA

**October 29-31**

Military Communications Conference 2007  
Orlando, FL

**November 5-9**

Software Defined Radio Forum 2007  
Denver, CO

### Europe, Middle East, and Africa

**August 27-29**

International Conference on Field Programmable Logic and Applications  
Amsterdam, Netherlands

**September 6-11**

IBC  
Amsterdam, Netherlands

### Asia Pacific

**August 7-10**

Agilent Digital Measurement Forum  
Taiwan



# Logic analyzers up to 50% off

## A special limited-time offer from Agilent.



### Agilent portable and modular logic analyzers

- Increased visibility with FPGA dynamic probe
- Customized protocol analysis with Agilent's exclusive packet viewer software
- Low-cost embedded PCI Express packet analysis
- Pricing starts at \$9,450



Now you can see inside your FPGA designs in a way that will save weeks of development time.

The FPGA dynamic probe, when combined with an Agilent Windows®-based logic analyzer, allows you to access different groups of signals inside your FPGA for debug—without requiring design changes. You'll increase visibility into internal FPGA activity by gaining access up to 128 internal signals with each debug pin.

Our 16800 Series logic analyzers offer you unprecedented price-performance in a portable family, with up to 204 channels, 32 M memory depth and a pattern generator available.

**And now for a limited time, you can receive up to 50% off our newest 16901A modular logic analyzer mainframe when you purchase eligible measurement modules. Offer valid February 1, 2007 through August 15, 2007. Reference promotion 5.564.**

[www.agilent.com/find/logic-offer](http://www.agilent.com/find/logic-offer)



# Scaling Chip-to-Chip Interconnect Made Simple

Sarance's high-performance Interlaken IP cores connect devices at up to 50 Gbps.

by Farhad Shafai  
 Vice President, R&D  
 Sarance Technologies Inc.  
[farhad.shafai@sarance.com](mailto:farhad.shafai@sarance.com)

Kelvin Spencer  
 Senior Design Engineer  
 Sarance Technologies Inc.  
[kelvin.spencer@sarance.com](mailto:kelvin.spencer@sarance.com)

As the world gets connected, the demand for bandwidth continues to increase. The interconnect technology for communication systems must not only connect devices today, but also provide a roadmap for the future. Traditional solutions, such as XAUI or SPI-4.2, cannot scale beyond 10 Gbps. SPI-4.2 uses a low-speed parallel bus, which requires a large number of pins to transfer 10 Gbps of data. XAUI does not have any provisions for channelizing the packet stream, making it unsuitable for applications that differentiate between packets. Several attempts have been made to build on XAUI and SPI-4.2; all derivatives, however, suffer from the inherent limitations of the solutions on which they are based, and are therefore not optimal.

Interlaken is a new chip-to-chip, channelized packet interface protocol developed by Cisco Systems and Cortina Systems. It is based on SERDES technology and provides

a framework for an efficient and robust chip-to-chip packet interface that easily scales from 10 Gbps to 50 Gbps and beyond. Several applications for Interlaken are shown in Figure 1. Simply put, you can use Interlaken as the connectivity solution for all devices in a typical network communication line card. The devices include front-end aggregation ICs or FPGAs, network processors, and traffic managers. Additionally, you can use translation FPGAs to bridge between legacy and modern devices that have Interlaken interfaces.

Sarance Technologies has developed a suite of Interlaken IP cores (IIPC) targeted at Xilinx® Virtex™-5 FPGAs. The IIPC family of cores is a highly optimized implementation of Interlaken that takes advantage of the advanced features of the Virtex-5 device. The IIPC abstracts all of the details of Interlaken and provides a very simple and straightforward interface. All members of the IIPC family use the same user-side protocol and programming interface, greatly simplifying performance scaling: using a 10-Gbps IIPC core is no different than using a 50-Gbps IIPC.

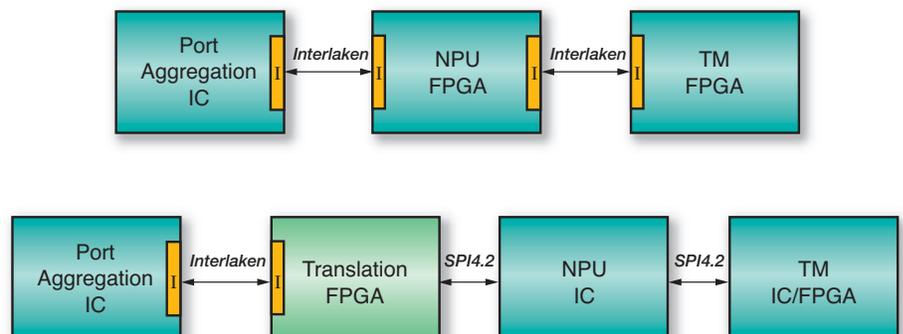


Figure 1 – Typical applications

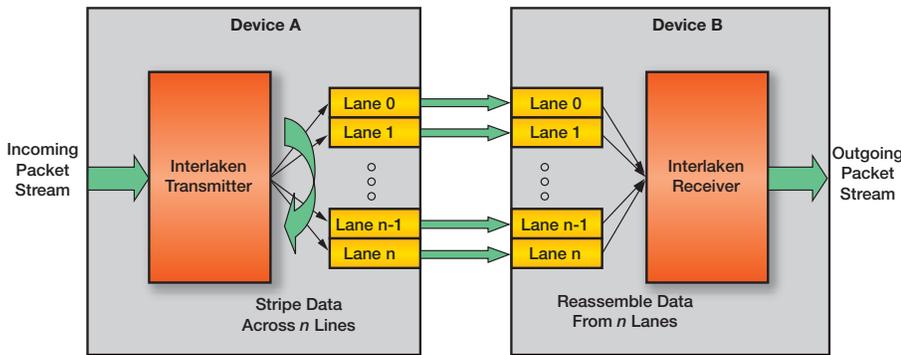


Figure 2 – A typical chip-to-chip implementation (only the unidirectional link is shown)

Bandwidth	MGT Lanes	MGT Rate	Logic LUTs	Block RAMs
12.5 Gbps	4	3.125 Gbps	9,000	5
25 Gbps	8	3.125 Gbps	18,000	5
50 Gbps	16	3.125 Gbps	32,500	5

Table 1 – Configuration and Virtex-5 LXT resource utilization

**Interlaken Basics**

Interlaken is a narrow, high-speed channelized chip-to-chip interface. For simplicity’s sake, we will not discuss the finer details of the protocol. At a high level, the basic concepts of the protocol include:

- Support for 256 logic channels
- Data scrambling and 64B/67B data encoding to ensure proper DC balance
- Segmentation of data into bursts delineated by control words
- CRC24 protection for each data burst
- CRC32 protection for each lane
- Protocol independence from the number of SERDES lanes and SERDES rate
- Support for in-band and out-of-band flow-control mechanisms
- Lane diagnostics and lane decommissioning

A typical chip-to-chip implementation is shown in Figure 2. The packet data is striped across any number of high-speed serial lanes by the transmitting device and then reassembled by the receiving device.

The protocol is independent from the number of SERDES lanes and the SERDES rate, which makes the performance proportional to the number of SERDES lanes. As an example, consider a 10-Gbps system. Using four multi-gigabit transceivers (MGTs) running at 3.125 Gbps, we can build an interface with a total raw bandwidth of 12.5 Gbps that has enough headroom for protocol overhead

and can transmit 10 Gbps of real payload. Scaling the interface up to 20G simply requires doubling the number of MGTs to eight; the bandwidth scales accordingly.

**Sarance’s IIPC Family**

The IIPC is a highly optimized implementation of Interlaken and, remaining true to the intent of Interlaken, is architected to offer the same flexibility and scalability offered by the protocol. IIPC can stripe and reassemble data across any number of SERDES lanes running at any SERDES rate and has support for as many as 256 channels. It is fully compliant with revision 1.1 of the Interlaken specification document and is hardware-proven to be interoperable with several ASIC implementations of Interlaken.

Table 1 lists the device utilization and implementation details for three different IIPC cores targeted to Virtex-5 LXT FPGAs.

Figure 3 shows the block diagram of the IIPC. At the time of this writing, IIPC is capable of supporting as much as 50 Gbps of raw bandwidth. The IIPC is divided into two major functional partitions:

- Lane logic that is replicated for each SERDES lane
- Striping and reassembly logic and the user-side interface

Lane-logic circuitry is the dominant portion of the overall utilization of the

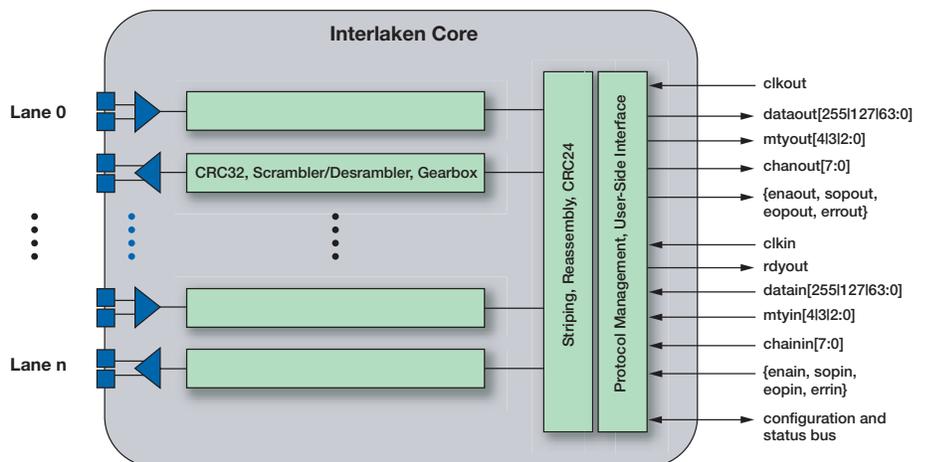


Figure 3 – Block diagram (SERDES is on the left; the user-side interface is on the right)

IIPC, since it is replicated for each lane. Each lane has a gear box, CRC32, and a descrambler/scrambler module. All of these functions have traditionally been very expensive in FPGA technology. Our implementation of these functions, however, takes full advantage of the Virtex-5 device's six-input LUTs in a way that makes the circuits very compact and efficient.

The striping and reassembly logic performs the required MUXing to transfer data between the user-side interface and the lane circuitry and handles link-level functions. Although the striping and reassembly logic is relatively smaller in terms of area, it has to process the total bandwidth of the interface and therefore is the most timing-critical part.

## Our roadmap has us increasing the bandwidth to 120 Gbps in the very near future.

Specifically, the CRC24 function has to potentially process 50 Gbps of data. Again, we have made the most of the FPGA's hardware features to come up with a very efficient and high-performance implementation of the CRC24 function.

### User-Side Interface

To help ease the integration of the IIPC with other logic in the FPGA, we have implemented a very simple and straightforward user-side interface. The bus protocol for transferring packet data to and from the IIPC is similar to the familiar SPI-like bus protocols that are commonly used in the industry. The configuration interface comprises a set of configuration input signals and another set of status output signals that can be easily connected to any processor interface. The status signals monitor the status of the link and identify possible configuration or transmission errors.

One key feature of our user-side interface is that it can be set to be identical for the entire IIPC family. This feature allows you to implement the same user-side logic in all of your designs, independent of the configuration or bandwidth of the Interlaken interface. You can build a 10G design today knowing that your configura-

tion software and FPGA architecture do not change when the design is scaled to 20G, 40G, and beyond. Even if you decide to change the SERDES rate or number of lanes, the user-side interface is still not affected. IIPC provides you with a solution for today – and for the future – in a single, highly optimized package.

### Ease of Use

Using the IIPC is as simple as powering up the FPGA, setting the configuration registers, resetting the core, and waiting for the core to signal that it is ready. The IIPC will automatically communicate with the other device and, when link integrity is established, will set a status signal. All you have

to do is monitor the status signal and start sending packets when it is asserted.

The IIPC handles all of the details of Interlaken, including automatic word and lane alignment and automatic scrambler/descrambler synchronization. In addition, the IIPC performs full protocol checking and error handling. It recovers from all error conditions (any number of bit errors are properly detected and appropriately handled) and will never violate the user-side protocol.

### Conclusion

Interlaken is the future of chip-to-chip packet interfaces. It combines the benefits of the latest SERDES technology and a simple yet robust protocol layer to define a flexible and scalable interconnect technology. Sarance Technologies's IIPC is an optimized implementation of the Interlaken revision 1.1 specification targeted for the Virtex-5 FPGA.

Our core is hardware-proven to interoperate with several ASIC implementations of Interlaken and can support up to 50 Gbps of raw bandwidth. Our roadmap has us increasing the bandwidth to 120 Gbps in the very near future. For the latest updates and information about our interactive demonstration platform, e-mail [interlaken@sarance.com](mailto:interlaken@sarance.com).

Supporting Your Future  
**HUNT ENGINEERING**

## USB connected Programmable FPGA systems

### V-II Pro PowerPC

- Virtex-II Pro XC2VP7
- 256 Mbytes DDR Memory
- Configurable digital I/Os
- PowerPC boot FLASH
- USB 2 or Standalone



### Software Defined Radio

- Virtex-II FPGA 1M gates
- 2 ch 125Msps A/D and D/A
- TI C6203 DSP
- 32Mbytes SDRAM
- Configurable Digital I/O
- USB 2 or Standalone



### Imaging with Virtex-4FX

- Virtex-4 FPGA FX12
- 128Mbytes DDR Memory
- CameraLink connection
- VHDL and PowerPC Imaging Libs
- USB 2 or Standalone



Programmable hardware with cables, device drivers, loading tools, examples and Power Supply. Systems can be used connected to a PC using USB, or can function standalone (without USB) using the initialisation PROMs.

sales@hunteng.co.uk  
+44 (0)1278 760188

[www.hunt-rtg.com](http://www.hunt-rtg.com)



# PCI Express and FPGAs

Why FPGAs are the best platform for building PCI Express endpoint devices.

by Alex Goldhammer

Technical Marketing Manager, Platform Solutions  
Xilinx, Inc.

[alex.goldhammer@xilinx.com](mailto:alex.goldhammer@xilinx.com)

PCI Express is a high-speed serial I/O interconnect scheme that employs a clock data recovery (CDR) technique. The PCI Express Gen1 specification defines a line rate of 2.5 Gbps per lane, allowing you to build applications that have a throughput of 2 Gbps (after 8B/10B encoding) for a single-lane (x1) link to 64 Gbps for 32 lanes. This allows a significant reduction in pin count while maintaining or improving throughput. It also reduces the size of the PCB, the number of traces and layers, and simplifies layout and design. Fewer pins also translate to reduced noise and electromagnetic interference (EMI). CDR eliminates the clock-to-data skew problem prevalent in wide parallel buses, making interconnect implementations easier.

The PCI Express interconnect architecture is primarily specified for PC-based (desktop/laptop) systems. But just like PCI, PCI Express is also quickly moving into other system types, such as embedded systems. It defines three types of devices: root complex, switch, and endpoint (Figure 1). The CPU, system memory, and graphics controller connect to a root complex, which is roughly equivalent to a PCI host. Because of PCI Express' point-to-point nature, switch devices are necessary to expand the number of system functions. PCI Express switch devices connect a root complex device on the upstream side to endpoints on the downstream side.

Endpoint functionality is similar to a PCI/PCI-X device. Some of the most common endpoint devices are Ethernet controllers or storage HBAs (host-bus adapters). Because FPGAs are most frequently used for data processing and bridging functions, the largest target function for FPGAs is endpoints. FPGA implementations are ideally suited for video, medical imaging, industrial, test and measurement, data acquisition, and storage applications.

The PCI Express specification maintained by the PCI-SIG mandates that every PCI Express device use three distinct proto-

col layers: physical, data-link, and transaction. You can build a PCI Express endpoint using a single- or two-chip solution. For example, you can use a low-cost FPGA such as a Xilinx® Spartan™-3 device for building data-link and transaction layers with a commercially available discrete PCI Express PHY (Figure 2). This option is best suited for x1 lane applications such as bus controllers, data-acquisition cards, and performance-boosting PCI 32/33 devices. Or you can use a single-chip solution such as Virtex™-5 LXT or SXT FPGAs, which have an integrated PCI Express PHY. This option is best for communications or high-definition audio/video endpoint devices (Figure 3) that require higher performance of x4 (8-Gbps throughput) or x8 (16-Gbps throughput) links.

Before selecting a technology for implementing a PCI Express design, you must carefully consider the choice of IP, link efficiency, compliance testing, and availability of resources for the application. In this article, I'll review the factors for building single-chip x4- and x8-lane PCI Express designs with the latest FPGA technology.

**Choice of IP**

As a designer, you can choose to build your own soft IP or buy IP from either a third party or an FPGA vendor. The challenge of building your own IP is that not only do you have to create the design from scratch, you also have to worry about verification, validation, compliance, and hardware evaluation. IP purchased from a third party or FPGA vendor will have gone through all of the rigors of compliance testing and hardware evaluation, making it plug and play. When working with a commercially available, proven, compliant PCI Express interface, you can focus on the most value-added part of the design: the user application. The challenge of using soft IP is the availability of resources for the application. As the PCI Express MAC, data-link, and transaction layers in soft IP cores are implemented using programmable fabric, you must pay special attention to the number of remaining block RAMs, look-up tables, and fabric resources.

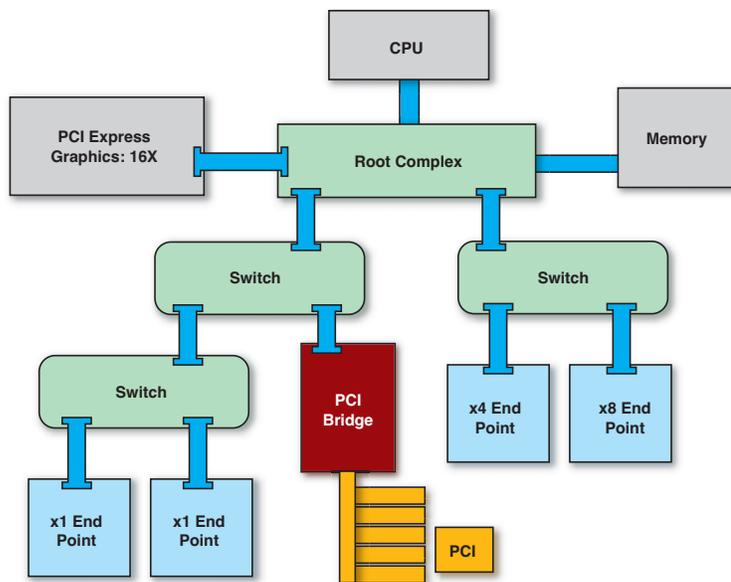


Figure 1 – PCI Express topology

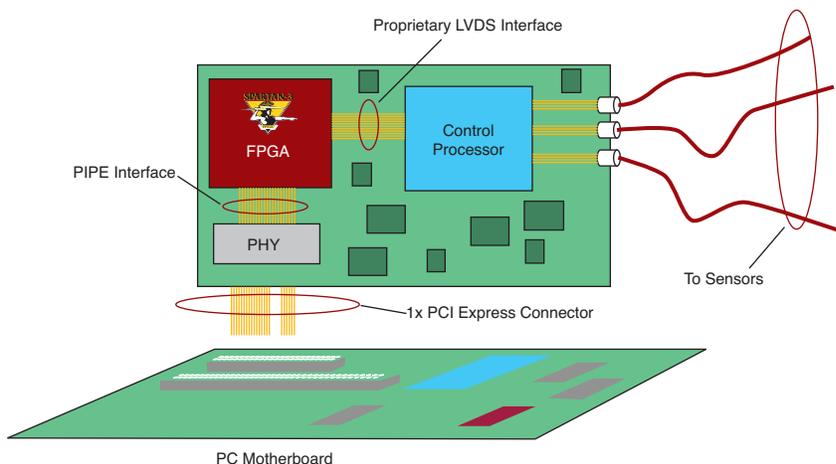


Figure 2 – Spartan-3 FPGA-based data-acquisition card

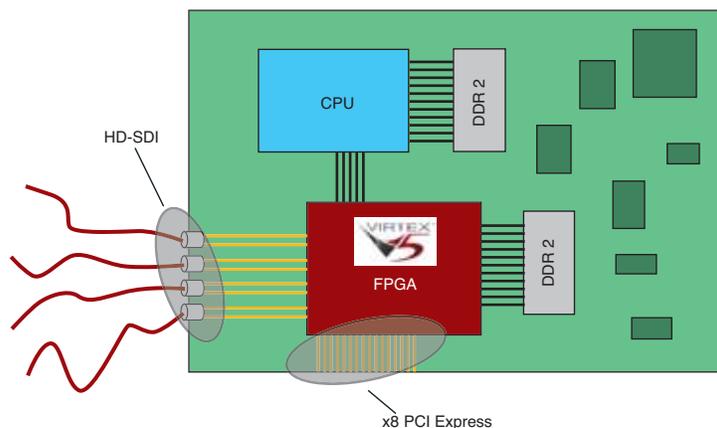


Figure 3 – Virtex-5 LXT FPGA-based video application

Another option is to use an FPGA with the latest technology. The Virtex-5 LXT and SXT have an integrated x8-lane PCI Express controller implemented in dedicated gates (Figure 4). This type of implementation is very advantageous, as it requires a minimal number of FPGA logic resources because the design is implemented in hard silicon. For example, in the Virtex-5 LXT FPGA, an x8-lane soft IP core can consume up to 10,000 logic cells, while the hard implementation will need about 500 logic cells, mostly for interfacing. This resource savings sometimes allows you to choose a smaller device, which is generally cheaper. Integrated implementations also have typically higher performance, wider data paths, and are software-configurable.

Another challenge with soft IP implementations is the number of features. Typically, such cores only implement the minimum features required by the specification to meet performance or compliance goals. Hard IP, on the other hand, can support a comprehensive feature list based on customer demand and full compliance (Table 1). There are no major performance or resource-related issues.

**Latency**

Although the latency of a PCI Express controller will not have a huge impact on overall system latency, it does affect the performance of the interface. Using a narrower data path helps latency.

For PCI Express, latency is the number of cycles it takes to transmit a packet and receive that packet across the physical, logical, and transaction layers. A typical x8-lane PCI Express endpoint will have a latency of 20-25 cycles. At 250 MHz, that translates into 80-100 ns. If the interface is implemented with a 128-bit data path to make timing easier (such as 125 MHz), the latency doubles to 160-200 ns. Both the soft and hard IP implementations in the latest Virtex-5 LXT and SXT devices implement a 64-bit data path at 250 MHz for x8 implementations.

**Link Efficiency**

Link efficiency is a function of latency, user application design, payload size, and overhead. As payload size (commonly referred to as maximum payload size) increases, the

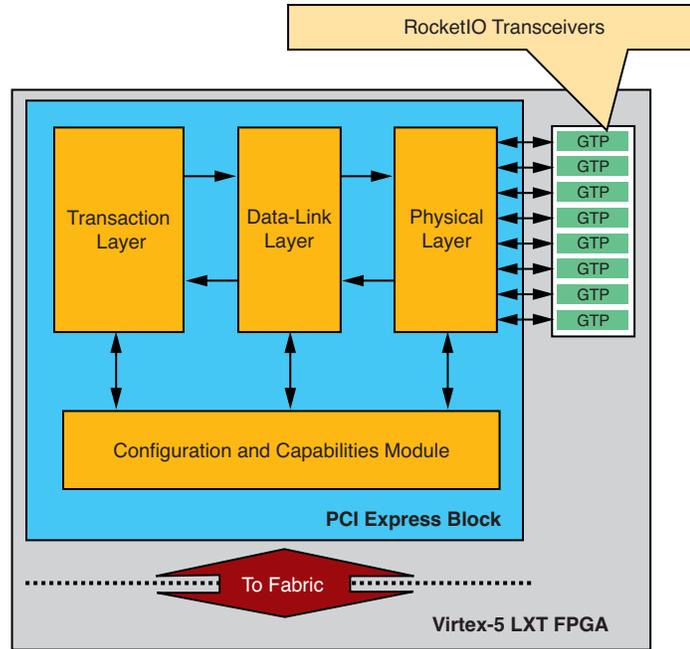


Figure 4 – Virtex-5 LXT FPGA PCI Express endpoint block diagram

Performance			
Lane Width	Interface Data Width	Interface Speed	Bandwidth (each direction)
x1	64	62.5/125/250 MHz	2 Gbps
x2	64	62.5/125/250 MHz	4 Gbps
x4	64	125/250 MHz	8 Gbps
x8	64	250 MHz	16 Gbps

PCI Express Specification v1.1 Compliance	
Requirement	Support (Y/N)
Clock Tolerance (300 ppm)	Y
Spread-Spectrum Clocking	Y
Electrical Idle Generate and Detect	Y
Hot Plug	Y
De-Emphasis	Y
Jitter Specifications	Y
CRC	Y
Automatic Retry	Y
QOS	2 VC/round robin, weighted round robin, or strict priority
MPS	128-4096 bytes
BARs	Configurable 6 x 32 bit or 3 x 64 bit for memory or I/O
Required Power Management States	Y

Table 1 – Virtex-5 LXT FPGA PCI Express capabilities

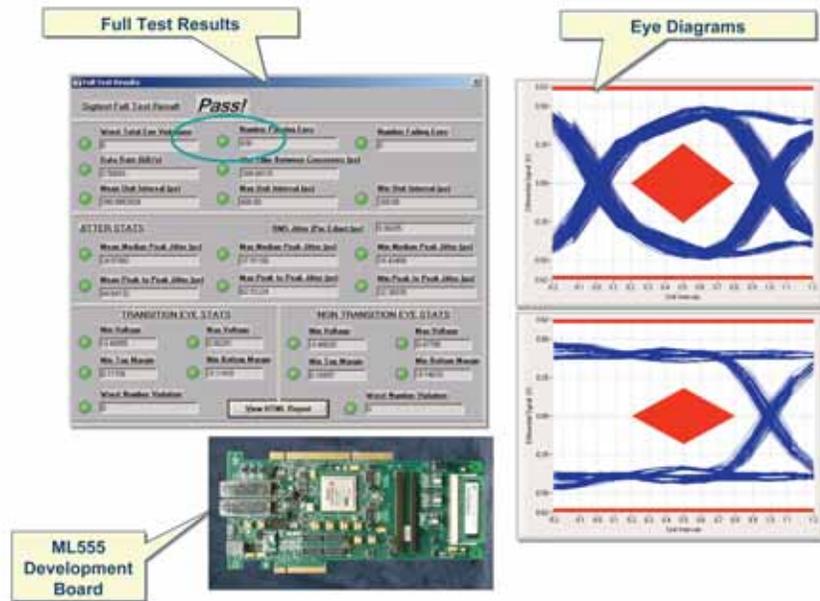


Figure 5 – Virtex-5 LXT FPGA PCI Express compliance workshop results

effective link efficiency also increases. This is caused by the fact that the packet overhead is fixed; if the payload is large, the efficiency goes up. Normally, a payload of 256 bytes can get you a theoretical efficiency of 93% (256 payload bytes + 12 header bytes + 8 framing bytes). Although PCI Express allows packet sizes up to 4 KB, most systems will not see improved performance with payload sizes larger than 256 or 512 bytes. A x4 or x8 PCI Express implementation in the Virtex-5 LXT FPGA will have a link efficiency of 88-89% because of link protocol overhead (ACK/NAK, re-transmitted packets) and flow control protocol (credit reporting).

Using FPGAs for implementation gives you better control over link efficiency because it allows you to choose the receive buffer size that corresponds to the endpoint implementation. If both link partners do not implement the data path in a similar way, the internal latencies on both will be different. For example, if link partner #1 uses the 64-bit, 250-MHz implementation with a latency of 80 ns and link partner #2 uses the 128-bit, 125-MHz implementation with a latency of 160 ns, the combined latency for the link will be 240 ns. Now, if link partner #1's receive buffer was designed for a latency of 160 ns – expecting that the link partner would also be a 64-bit, 250-MHz implementation – then the link efficiency would

go down. With an ASIC implementation, it would be impossible to change the size of the receive buffer, and the loss of efficiency would be real and permanent.

User application design will also have an impact on link efficiency. The user application must be designed so that it drains the receive buffer of the PCI Express interface regularly and keeps the transmit buffer full all the time. If the user application does not use packets received right away (or does not respond to transmit requests immediately), the overall link efficiency will be affected regardless of the performance of the interface.

When designing with some processors, you will need to implement a DMA controller if the processors cannot perform bursts longer than 1 DWORD. This translates into poor link utilization and efficiency. Most embedded CPUs can transmit bursts

longer than 1 DWORD, so the link efficiency for such designs can be effectively managed with a good FIFO design.

### PCI Express Compliance

Compliance is important detail that is frequently missed and often undervalued. If you are building PCI Express applications that must work with other devices and applications, ensuring that your design is compliant is a must.

The compliance is not just for the IP but for the entire solution, including the IP, user application, silicon device, and hardware board (Figure 5). If the entire solution has been validated at a PCI-SIG PCI Express compliance workshop (also known as a “plug fest”), it is pretty much guaranteed that the PCI Express portion of your design will always work.

### Conclusion

Replacing PCI, PCI Express has become the de facto system interconnect standard and has jumped from the PC into other system markets including embedded system design. FPGAs are ideally suited to build PCI Express endpoint devices, as they allow you to create compliant PCI Express devices with the added customization that embedded users desire.

New 65-nm FPGAs like the Virtex-5 LXT and SXT families are fully compliant to the PCI Express specification v1.1 and offer an abundance of logic and device resources to the user application. The Spartan-3 family of FPGAs with external PHY offer a low-cost solution. These factors, combined with the inherent programmable logic advantages of flexibility, reprogrammability, and risk reduction, make FPGAs the best platforms for PCI Express. ●●●

#### TAKE THE NEXT STEP (Digital Edition: [www.xcellpublications.com/subscribe/](http://www.xcellpublications.com/subscribe/))

- Buy the Development Kit for PCI Express:
  - Virtex-5 FPGA edition
  - Spartan-3 FPGA edition
- Download the Protocol Pack for PCI Express
- Watch our live demo from the Embedded Systems Conference
- Register for a PCIe class



# Virtex-5 FPGA Techniques for High-Performance Data Converters

You can harness the DSP resources of Virtex-5 devices to interface to the analog world.

by Luc Langlois  
Global Technical Marketing Manager, DSP  
Avnet EM  
[luc.langlois@avnet.com](mailto:luc.langlois@avnet.com)

The incessant demand for higher bandwidths and resolutions in communication, video, and instrumentation systems has propelled the development of high-performance mixed-signal data converters in recent years. This poses a challenge to system designers seeking to preserve the exceptional signal-to-noise specifications of these devices in the signal processing chain. Xilinx® Virtex™-5 FPGAs provide extensive resources for high-performance mixed-signal systems, supported by efficient development tools spanning all phases of design, from system-level exploration to final implementation.

## Key Specifications of Data Converters

A typical mixed-signal processing chain starts at the analog-to-digital converter (ADC). Modern high-performance ADCs provide sampling rates extending into the hundreds of megasamples per second (MSPS) for 12- and 14-bit devices. For example, the Texas Instruments ADS5463

ADC provides 12 bits at 500 MSPS, with 64.5 dB full-scale (dBFS) of signal-to-noise ratio (SNR) to 500 MHz.

Fast sampling rates offer several benefits, including the ability to digitize wide-band signals, reduced complexity of anti-alias filters, and lower noise power spectral density. The result is improved SNR in the system. Your challenge is to implement the high-speed interface between the data converter and FPGA while preserving the SNR throughout the signal processing chain in the FPGA.

Before the digital ADC data is captured in the FPGA, you must take careful precautions to minimize jitter on the data converter sampling clock. Jitter degrades SNR depending on the signal bandwidth of interest. For example, preserving 74 dB of SNR - or approximately 12 effective number of bits (ENOB) - for signal bandwidths extending to 100 MHz requires a maximum 300 fs (femtoseconds) of clock jitter. Modern ADCs provide clever interfaces that simplify distribution of clean low-jitter clocks on the board. Let's examine how key features of Virtex-5 FPGAs are used to implement these interfaces.

## High-Performance ADC Interface

High-performance ADC sampling rates often exceed the minimum rate necessary to avoid aliasing, or Nyquist rate, defined as twice the highest frequency component in the analog input signal. The highly over-sampled digital signal entering the FPGA need not maintain a fast sampling rate throughout the signal processing chain; it can be decimated with negligible distortion in the digital domain by a high-quality decimation filter. This offers the benefits of a slower system clock in subsequent processing stages for easier timing closure and lower power consumption.

Xilinx Virtex-5 and Spartan™-3A DSP FPGAs provide the ideal resources to implement high-performance decimation filters for fast ADCs using a technique known as polyphase decomposition. A polyphase decimation filter performs a sampling rate change by allocating the DSP workload among a set of D sub-filters, where D = decimation rate. Each subfilter is only required to sustain a throughput of  $f_s/D$ , a fraction of the fast incoming sampling rate  $f_s$  from the ADC.

As the decimation filter is often the first stage of digital processing, it calls for

the highest performance resources closest to the FPGA pins. A Virtex-5 FPGA's input/output block contains an IDDR (input double-data-rate register) driven directly from the FPGA input buffer. Several differential signal standards are supported, including LVDS, which can sustain in excess of 1-Gbps data rates while providing excellent board-level noise immunity.

The IDDR is used to de-multiplex the fast incoming digital signal from the ADC into two single-data-rate data streams, each at one-half the ADC sampling rate. This is the ideal format to feed a 2x polyphase decimation filter. Using the Virtex-5 DSP48E, each subfilter can sustain 550 MSPS for a maximum 1.1-GSPS ADC sampling rate. Similarly, the Spartan-3A DSP can sustain 500 MSPS ADC sampling rates.

With the benefits of faster ADC sampling rates come the challenges of smaller data-valid windows to latch the data into the FPGA. Furthermore, the wider the ADC data-word precision, the more daunting the layout task and the higher potential for skew across individual signals of the data bus, resulting in corrupted data. Virtex-5 FPGAs provide a robust solution called IODELAY, a programmable delay element contained in every I/O block. IODELAY can individually time-shift each signal of the data bus to accurately position the data-valid window at the optimal transition of the half-rate data-ready signal (DRY).

Figure 1 illustrates the unique features in Virtex-5 devices that serve to implement a high-performance ADC interface. To minimize sampling jitter, the ADC forwards the source-synchronous DRY signal along with the data, while the clean, low-jitter sampling clock routes directly to the ADC without passing through the FPGA.

### High-Performance DAC Interface

For equal data-word precision, digital-to-analog converters (DACs) typically offer higher sampling rates than ADCs, resulting in significant design challenges at the DAC extremity of the signal chain. Several features of the Virtex-5 architecture can help surmount the task. For example, consider

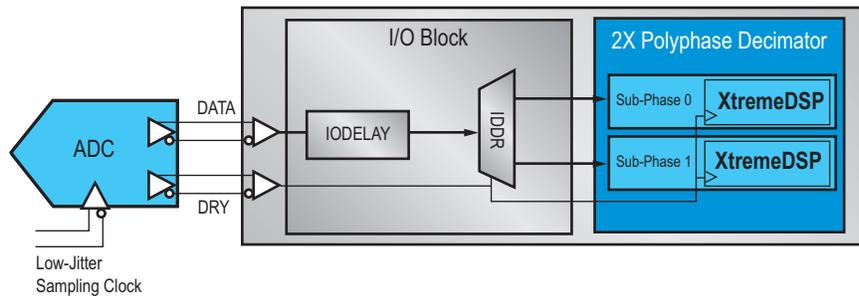


Figure 1 - High-performance ADC interface

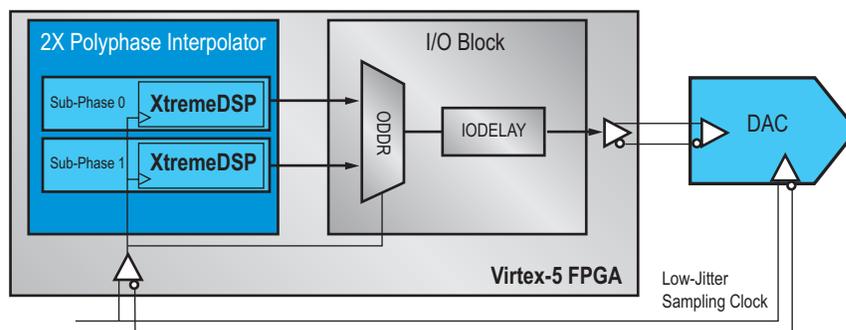


Figure 2 - DAC: polyphase interpolation + ODDR + LVDS

the Virtex-5 interface to a Texas Instruments (TI) DAC5682Z 16-bit dual-DAC with 1-GSPS sampling rate and LVDS inputs.

In a practical system, the 1-GSPS sampling rate need only be deployed at the final output stage to the DAC, while intermediate stages in the FPGA signal processing chain can work at a slower sampling rate commensurate with the signal bandwidth. This allows a slower system clock in the intermediate processing stages, with the benefits of easier timing closure and lower power consumption.

As is the case for the ADC, polyphase filters are efficient DSP structures to realize sampling rate changes at the DAC end of the signal chain. To attain a 1-GSPS output sampling rate to the TI DAC5682Z, a 2x polyphase interpolation filter uses two sub-filters, each with a throughput of 500 MSPS. These rates are within the performance specifications of the Virtex-5 DSP48E slice.

A multiplexer is required to combine the outputs of the sub-filters to attain the fast output rate from the polyphase. For a 1-GSPS output sampling rate, it is advisable

to situate the polyphase interpolator multiplexer as close as possible to the LVDS output buffers driving the DAC5682Z. Virtex-5 FPGAs provide a dedicated resource within the I/O block that is ideally suited to this purpose: the ODDR (output double-data-rate) registers. The ODDR routes directly to fast LVDS differential output buffers able to sustain output rates of 1 GSPS (and beyond) while maintaining signal integrity on the PCB.

### Conclusion

In this article, I've presented DSP and interface techniques for mixed-signal systems using Xilinx Virtex-5 FPGAs. You can optimize system performance by preserving the outstanding SNR specifications of modern high-performance data converters using key features of Virtex-5 devices.

The techniques described in this article will be featured in Speedway 2007 DSP sessions, in collaboration with major Avnet data converter suppliers Texas Instruments, Analog Devices, and National Semiconductor. For details, visit <http://em.avnet.com/xilinxspeedway>.



# Reducing CPU Load for Ethernet Applications

## A TOE makes 10 Gigabit Ethernet possible.

by Andreas Magnusson  
CTO  
IPBlaze  
[andreas.magnussen@ipblaze.com](mailto:andreas.magnussen@ipblaze.com)

Ethernet is playing an increasingly important role today, as it is used anywhere to connect everything. Not surprisingly, bandwidth requirements are increasing in the backbone as well as in end systems.

The implication of this increasing bandwidth is an increased traffic processing load in end systems. Today, most end systems use one or more CPUs with an OS and a network stack to implement network interface functions. For many applications, the increasing traffic load leads to performance issues in the network stack implementation. As these performance issues are seen already at 1 Gigabit Ethernet (GbE), implementing 10 GbE using a software stack is not a viable solution.

To solve these problems, IPBlaze has developed a unique and highly configurable TCP/IP offload engine (TOE). The TOE processes the TCP/IP stack at wire speed in

hardware instead of using a host CPU and thus reduces its processing burden.

IPBlaze has implemented the TOE for various end systems. Our measurement results show that using an IPBlaze TOE reduces latencies and CPU utilization considerably. In this article, I'll give an overview of the implementation options available today for Ethernet applications and show where the IPBlaze TOE can give you the upper edge in terms of performance. Figure 1 shows performance examples with and without the IPBlaze TOE.

### Ethernet Configurations

Table 1 shows the five different implementation options available today.

A CPU plus a simple network interface card (NIC) is a very general solution used in most PCs and servers today. A new CPU generation always provides higher performance and thus also increases network performance. However, the increase in network processing performance is significantly lower than the increase in CPU power. The same problem exists with

embedded CPUs – the performance issues arise at 10 to 100 times lower data rates.

A high-performance CPU and simple NIC is a flexible solution with a well-known API (socket), and it is easy to implement applications in common PC/server environments. One of the downsides is the difficulty in scaling efficiently to 10 GbE because the load distribution between CPU cores and process intercommunication creates a bottleneck. Power dissipation (heat) is also a limiting factor in many systems.

Some NIC acceleration ASICs on the market perform protocol offload for specific applications such as storage. A high-performance CPU and NIC acceleration ASIC is a good solution if the ASIC supports all of the features needed (iSCSI, TCP offload). The functionality and bandwidth is fixed, however, which makes it very hard to add functionality or adapt to protocol changes without a huge performance impact.

FPGAs are so powerful today that it is possible to implement an accelerated 10

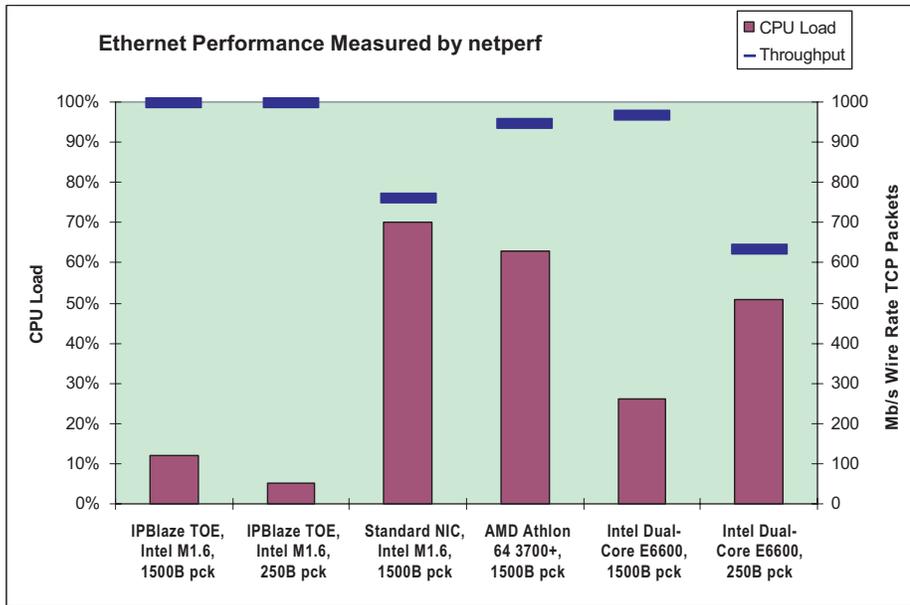


Figure 1 – The CPU power needed to process network data and throughput, tested at 1 GbE links, using the netperf performance measurements program.

## Application Examples

The IPBlaze TOE is a general protocol offload engine including TCP offload. The IPBlaze TOE product lines include a number of TOEs with different functionality targeting different applications (Figure 2).

One example is an FPGA working as an intelligent high-performance TOE NIC at n-times-1 GbE or n-times-10 GbE connected to a PC through the PCIe interface. The NIC can be customized with add-on functions such as advanced switch flow control. You can also easily add protocol processing functions such as RDMA and iSCSI for storage applications.

Another example would be embedded video applications where Ethernet is used as a backplane. The TOE can transfer the images or video streams without the need for a CPU.

## High Flexibility with FPGAs

FPGAs hold a number of advantages compared to ASICs and offer scalable solutions with high data rates: 1 GbE, n-times-1 GbE, 10 GbE, and n-times-10 GbE. The flexibility of the FPGA makes it possible to add new functionality and adapt to protocol changes. Advanced network functions like switch-specific flow control, or performance-enhancing features can be added. In load distribution, the IPBlaze TOE can distribute traffic to multiple CPU cores for higher system performance. Maintenance and upgrade of the TOE hardware functions located in the FPGA is done by firmware upgrade – much like driver software updates.

## IPBlaze 10 GbE TOE core

The IPBlaze 10 GbE TOE core is an implementation of a full-featured TCP/IP networking subsystem in the form of a “drop-in” silicon IP core. It includes a standards-compliant TCP/IP stack implemented as a compact high-performance synchronous state machine.

The TOE core is built from a collection of synthesizable Verilog modules, which are customized before delivery to provide the best possible performance and feature set in a given system.

High Performance	Properties
High-performance CPU + simple NIC	High CPU load, CPU limits performance
High-performance CPU + NIC acceleration ASIC	Good with in supported feature set, but inflexible solution
High-performance CPU + NIC acceleration FPGA	Flexible, scalable solution
Medium Performance	Properties
ASIC SoC (communication controller CPU, network if)	General solution, performance limited by software
FPGA SoC (integrated CPU, network acceleration)	Good for high data rates, flexible and scalable

Table 1 – Overview of implementation options

GbE NIC in an FPGA at a competitive cost point. One example is the IPBlaze TOE implemented in Xilinx® Virtex™-4 and Virtex-5 devices. The Virtex-5 LXT device has hardware support for PCIe (PCI Express), which makes it an attractive choice to use in PCs and servers.

Examples of ASIC SoCs are controller-type ASICs and communication controller ASICs. This is a good solution for multi-function systems with limited bandwidth (up to 10-20 Mbps). However, it is limited by low bandwidth.

A powerful example of an FPGA SoC solution is an FPGA with an IPBlaze TOE and an embedded CPU (a PowerPC hard-core or MicroBlaze™ soft-core processor). Applications processing can be done in

software by the PowerPC while the FPGA hardware, together with the TOE, handles bulk transfers like video or images at wire speed (1 or 10 GbE). This solution makes it possible to use FPGA logic for fast data processing and fast data communication and software for complex low-bandwidth operations. Software in the embedded CPU can be Xilinx microkernel, Embedded Linux, or VxWorks.

The IPBlaze 10 GbE TOE NIC FPGA solution has the same performance as a TOE NIC ASIC while providing much higher flexibility. High-performance TOE NICs are now available using cost-effective FPGAs. FPGAs help you with fast time to market and ensure compatibility and interoperability.

The TOE core can be instantiated along with your own IP and configured to operate without the need for host CPU support.

The IPBlaze TOE core can support as many as 1,000 concurrent TCP connections in on-chip memory. When a higher number of connections are required, a connection cache will allow you to access

The host interface is essential for system behavior and performance and represents a significant part of the system complexity.

The host CPU has register access to TOE registers while the TOE has access to main memory. The main memory contains the send and receive data buffers, queue structures, and TCP connection state information if connection caching is

The flexibility in an FPGA solution allows easy upgrade of the FPGA code to support new protocol features.

The IPBlaze TOE core is designed to target a variety of FPGA families depending on the performance and functionality requirements. The IPBlaze TOE core has been implemented in Virtex-II, Virtex-4, and Virtex-5 devices.

Here are some example numbers from a 10 GbE TOE implemented in Virtex-5 devices:

- Core clock: 125 MHz
- TOE latency: 560 ns
- Throughput: 10G line rate (10G bidirectional)
- Packet processing rate: 12 million packets per second

Xilinx and IPBlaze provide TCP offload solutions that can be implemented as is or customized for functionality, size, speed, or target application. IPBlaze also offers a comprehensive set of high-performance TOE solutions for different market segments. Current TOE solutions include:

- IPBlaze General TOE for NIC applications.
- IPBlaze Security TOE for high-performance (2-times-10 GbE) bump-in-the-wire network monitoring. The security TOE supports more than 1 million concurrent TCP connections.
- IPBlaze Embedded TOE for SoC applications that require high-speed network communication (1 GbE and 10 GbE).

### Conclusion

With standards-compliant TOE cores, IPBlaze has created a technology platform for the 10 GbE TOE market, working with industry leaders in high-performance computing solutions. Programmable solutions enable system architects to add functionality as needed. Integrating multiple IP cores into a single FPGA can reduce board costs and time to market.

To learn more about protocol offload solutions, visit [www.ipblaze.com](http://www.ipblaze.com).

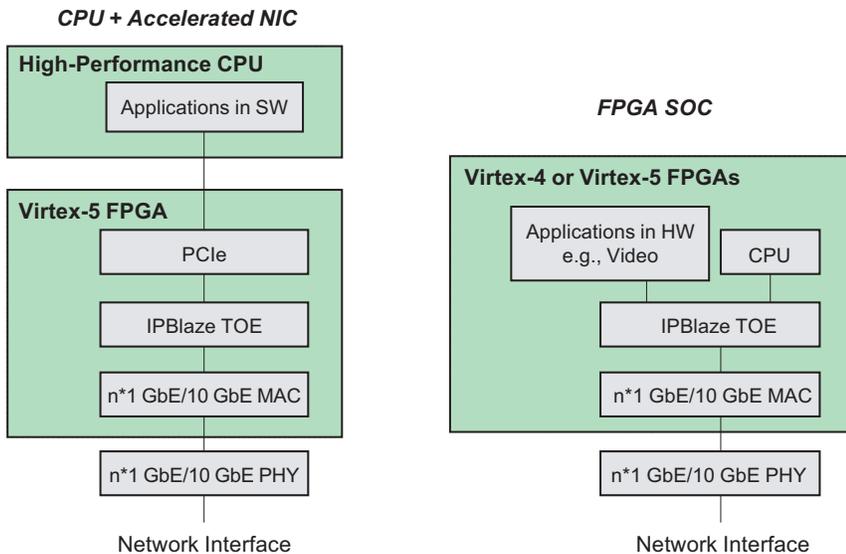


Figure 2 – Block diagram for a high-speed TOE solution for a PC server and embedded system

the most recently used connection states on-chip. The state information for non-cached TCP connections is stored in host memory or in dedicated external memory.

For applications where the TOE does not terminate connections but monitors all connections on a backbone, an external dedicated high-speed memory is used to support a very high number (1 million and above) of concurrent TCP connections.

The IPBlaze TOE core uses three well-defined interfaces:

- Xilinx 1 GbE and 10 GbE MAC network interfaces for either 1 GbE or 10 GbE support
- A high-speed RAM option for dedicated external memory for TCP connection state information
- Xilinx PCIe type or memory host processor interface

enabled. An event system is used for fast message signaling between the host and the TOE. The event system includes a number of optimizations for high throughput and low latency.

### OS-Compliant Socket API

Applications software interfaces to the TOE through a fully standards-compliant socket API, with significantly higher performance than a standard software socket implementation. The socket API is implemented at the CPU kernel level. The CPU load is reduced by 5-10 times and latency is improved by 4-5 times.

Hardware-based acceleration should be able to scale well. The evolution of performance in future Xilinx FPGA generations provides a good upgrade path. Furthermore, continuous improvements in the IPBlaze TOE technology also provide increased performance.

# Automated MGT Serial Link Tuning Ensures Design Margins

You can now streamline the serial link tuning process using Agilent's Serial Link Optimizer tool with Xilinx IBERT measurement cores.

by Brad Frieden  
Applications Development Engineer  
Agilent Technologies  
[brad\\_frieden@agilent.com](mailto:brad_frieden@agilent.com)

When implementing high-speed serial links in FPGAs, you must consider and address the effects of transmission line signal integrity. Used together, transmitter pre-emphasis and receiver equalization extend the rate at which high-speed serial data can be transferred by opening up the eye diagram, despite physical channel limitations.

An internal bit error ratio tester (IBERT) measurement core from Xilinx can view serial signals at an internal receiver point. Used in conjunction with the Agilent Serial Link Optimizer, you can have both a graphical view of the BER across the unit interval and automatically adjust pre-emphasis and equalization settings to optimize the channel.

In this article, I'll show you how to optimize a Virtex-4 MGT high-speed serial link through this process and discuss the results.

## Challenges of Signal Degradation

At 3.125- and 6-Gbps rates and rise times of 130 ps or shorter, it is no wonder that most applications end up with significant signal integrity effects from the physical channel that distort the signal at the receiver input. Distortion can come from multiple reflections caused by impedance discontinuities, but a more fundamental effect – especially in FR4 dielectric PC boards – slows down the edge speeds. Frequency-dependent skin effect causes a “slow tail” to the pulse. To compensate for this, you can apply a time-domain technique called pre-emphasis to the transmit pulse, with significant improvement at the receiver.

Additionally, because the channel is bandwidth-limited, you can apply a frequency-domain technique called equalization at the receiver to compensate for channel frequency roll off. A peaked frequency response yields a more flat response when combined with the channel roll off. The effects of equalization at the receiver input are quite drastic, but this is only visible inside the chip at the actual receiver input (post-equalization).

## Measuring Link Performance

It is important to be able to verify the performance of a link and to optimize it through the combined adjustment of transmitter pre-emphasis and receiver equalization. Unfortunately, taking a measurement on the pins of the FPGA where the receiver is located yields a very distorted signal, since you are observing the signal with pre-emphasis applied but without corresponding equalization.

Figure 1 is an example of such a measurement made with an Agilent digital communications analyzer on a 6-Gbps channel implemented with ASIC technology. Notice that the eye diagram actually appears completely closed, even though good signals are present at the receiver input inside the chip.

**IBERT Measurement Core**

Fortunately, you can observe what is going on at the FPGA's receiver input by using an IBERT measurement core that is part of the Xilinx® ChipScope™ Pro Serial I/O toolkit. The normal FPGA design is temporarily replaced with one that creates stimulus at the transmitter and measures BER at the receiver. These stimulus/response core pairs are placed in the design using a core generation process. Now it is possible to observe BER at the receiver inside the chip. Using that basic measurement capability, you can apply a variety of pre-emphasis and equalization combinations to optimize the channel response.

**Basic BERT Measurements**

To understand link performance, you must have the ability to measure bit error rate. To do this, a new tool called the Agilent Serial Link Optimizer takes control of IBERT core stimulus and response in the serial link to create such a measurement. A measurement system as shown in Figure 2 allows for this kind of test. Here, two Virtex™-4 FPGAs comprise the link: one implements the transmitter, the other the receiver. Both FPGAs are under JTAG control from the Serial Link Optimizer software, and measurements are taken at the receiver input point inside the FPGA to measure BER.

Some of the selectable measurement attributes include:

- Loopback mode (internal, external, or none)
- Test pattern type
- Dwell time at each point across the unit interval
- Manual injection of errors

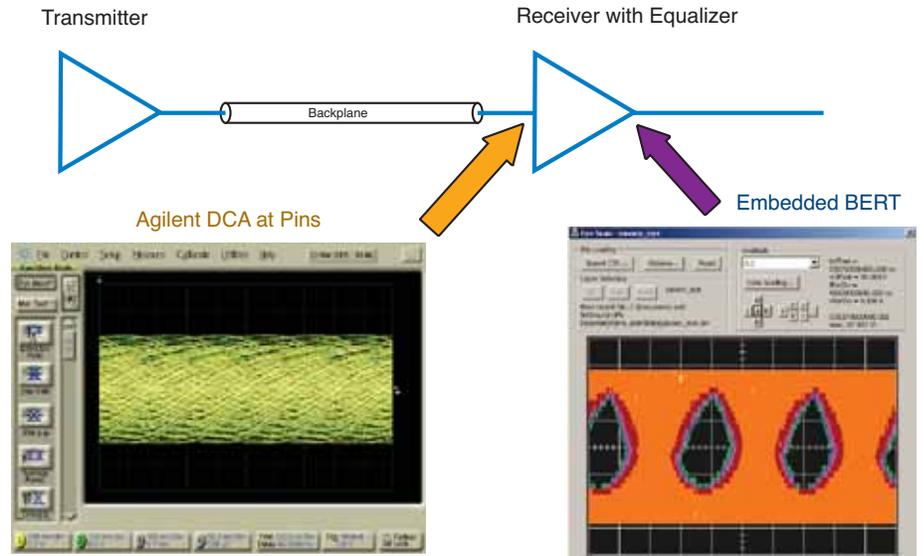


Figure 1 – DCA measurement on FPGA serial I/O pins versus on-chip measurement at the receiver input

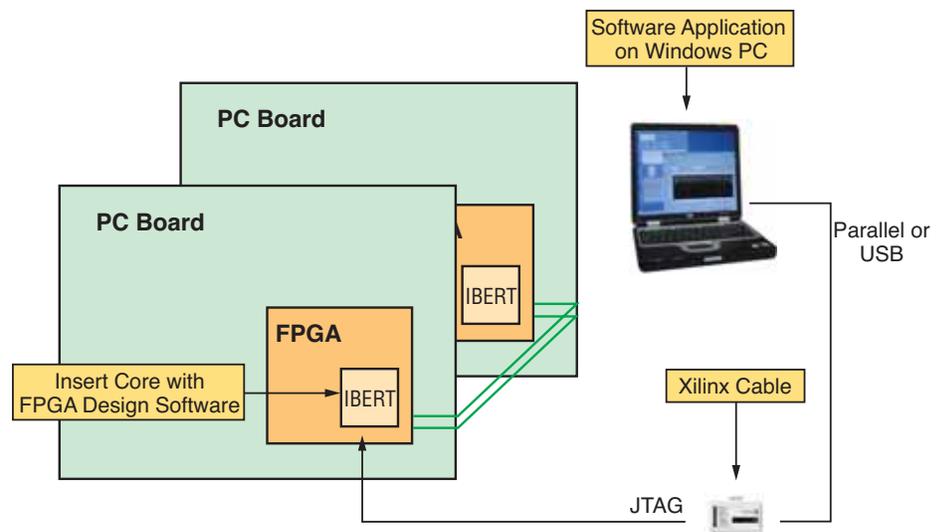


Figure 2 – Serial Link Optimizer block diagram

I obtained a measurement on a serial link that comprises a Xilinx Virtex-4 XC4VFX20 multi-gigabit transceiver (MGT) 113A transmitter on one Xilinx ML405 board, with a connection through a SATA cable over to an MGT 113A receiver in a second Virtex-4 XC4VFX20 FPGA on a second ML405 board. I made a USB JTAG connection to the first FPGA and IBERT core associated with the transmitter, and a parallel JTAG connection to the second IBERT core associated with the receiver. These cores, along

with the topology of the ML405 boards, dictate the physical channel.

Steps to set up this measurement include (assuming that the IBERT cores are already created and loaded into the FPGAs):

1. Start tool and configure USB JTAG for TX and verify connection
2. Configure parallel JTAG for RX and verify connection
3. Select MGT 113A transmitter on USB cable

4. Select MGT 113A receiver on parallel cable
5. Select loopback (external)
6. Select test pattern type (PRBS7)
7. Setup TX and RX line rates (reference clock 150 MHz, line rate 6 Gbps)
8. Select BERT tab and press “Run”

I made a measurement on the link with zero errors after  $1\text{E}+12$  bits (~ 3-min measurement). This measurement of BER on the channel occurred at the receiver input internal to the chip and required no external measurement hardware. It forms the basis for additional capability in the Serial Link Optimizer tool.

### A Graphical View of BER

The next step toward understanding link performance is to have the ability to graph BER as a function of the unit interval. To do this, the Agilent Serial Link Optimizer takes control of IBERT core stimulus and response in the serial link to create such a graph.

The same measurement system is used as with the basic BER test, but now measurements are taken at 32 discrete steps across the unit interval to show where error-free performance is achieved. The resulting graph is shown in the upper BER plot in Figure 3. The Virtex-4 MGT macro has the ability to adjust the sampling position across these 32 discrete points; the Serial Link Optimizer uses the IBERT core in conjunction with sampling position control to make the measurements. The link performance with default MGT pre-emphasis and equalization settings has zero errors for 0.06 (6%) of the unit interval. This is quite narrow, indicating very little margin. The system could benefit from proper link tuning.

### Automated Link Tuning

Let’s extend this measurement process yet again by automatically trying combinations of pre-emphasis and equalization settings until the error-free zone in the unit interval is maximized, resulting in the best link performance for speed and margin. The Serial Link Optimizer does just that. Link configuration options include an internal loopback test inside the I/O ring of a single FPGA; transmit and receive from one FPGA; and a transmit/receive pair test between two different FPGAs. It is also possible to inject signals from an external BERT instrument and monitor signals at the receiver inside the FPGA.

In our example, let’s use the same serial channel between two Virtex-4 FPGAs with the MGT 113A/B TX/RX pair. On the Serial Link Optimizer interface, select the “Tuning” tab and Run button as shown in Figure 3. The error-free zone with default

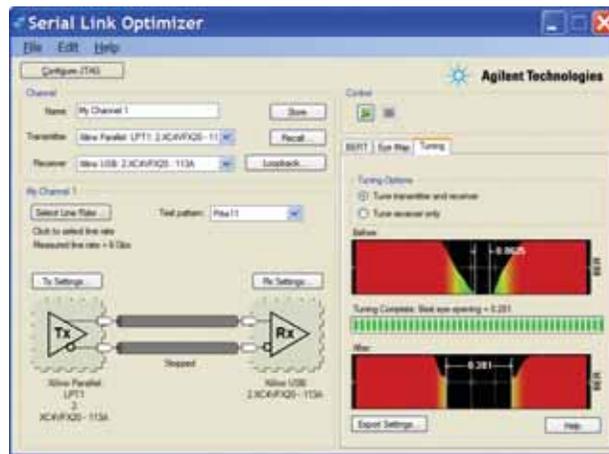


Figure 3 – BER plot before and after automatic adjustment of pre-emphasis and equalization to tune the serial link

FPGA settings was 0.06 unit interval. After attempting 141 combinations of pre-emphasis and equalization, the error-free zone increased to 0.281 (28%) unit interval, also shown in Figure 3. This means significantly better margins in the link design at the 6-Gbps speed.

### Exporting Link Design Constraints

These new pre-emphasis and equalization settings must now be incorporated into a real design that ultimately uses the serial link. Up until this point, I placed a temporary test

design into each FPGA to determine the optimal combination of pre-emphasis and equalization. Now that I have determined the optimal combination, the Serial Link Optimizer outputs the corresponding pre-emphasis and equalization settings for the MGTs. The MGT parameters are represented in VHDL, Verilog, and UCF formats and are output when clicking the “Export Setting” button. You can cut and paste these parameters back into the source design, thus incorporating the optimal link settings when the final design is programmed into the FPGAs.

For example, the Verilog settings for the first transceiver are:

```
// Verilog
//— Rocket IO MGT Preemphasis and Equalization —
defparam MGTx.RXAFEEQ = 9'b1111;
defparam MGTx.RXSELDACFIX = 5'b11111;
defparam MGTx.RXSELDACTRAN = 5'b11111;
```

### Required Components

To take advantage of these measurements, you will need:

1. A PC with Windows XP installed (SP2 or higher)
2. Xilinx programming cable(s), parallel and/or USB
3. Xilinx ChipScope Pro Serial I/O Toolkit (for IBERT core)
4. Agilent E5910A Serial Link Optimizer software

### Conclusion

Through automated adjustments of pre-emphasis and equalization, while simultaneously monitoring BER at the receiver input inside the FPGA, it is now possible to automatically optimize an MGT-based serial link.

At 3.125-Gbps rates, such an optimization is helpful to get good margins. But at 6-Gbps rates, optimization becomes crucial to achieve link performance that ensures reliable data transfer. This process can save significant time in reaching your desired design margins and can likely achieve better results than what is possible through traditional manual tuning.

For more information, visit [www.agilent.com/find/serial\\_io](http://www.agilent.com/find/serial_io) or [www.agilent.com/find/xilinxfpga](http://www.agilent.com/find/xilinxfpga).



# A High-Speed Serial Connectivity Solution with Aurora IP

Aurora is a highly scalable protocol for applications requiring point-to-point connectivity.

by Mrinal J. Sarmah  
Hardware Design Engineer  
Xilinx, Inc.  
[mrinal.sarmah@xilinx.com](mailto:mrinal.sarmah@xilinx.com)

Hemanth Puttashamaiah  
Hardware Design Engineer  
Xilinx, Inc.  
[hemanth.puttashamaiah@xilinx.com](mailto:hemanth.puttashamaiah@xilinx.com)

With advances in communication technology, you can achieve gigahertz data-transfer rates in serial links without having to make trade-offs in data integrity. The proliferation of serial connectivity can be attributed to its advantages over parallel communication, including:

- Improved system scalability
- More flexible, thinner cabling
- Increased throughput on the line with minimal additional resources
- More deterministic fault isolation
- Predictable and reliable signaling schemes
- Topologies that promise to scale to the needs of the end user
- Exceptional bandwidth per pin with very high skew immunity
- Reduced system costs because of smaller form factors, fewer PCB traces and layers, and lower pin/wire count

Although it offers real benefits, serial I/O has some negative attributes. Serial interfaces require high-bandwidth management inside the chip, special initialization and monitoring, bonding of lanes in an aggregated channel of multiple lanes, elastic buffers for data alignment, and de-skewing. Also, flow control is complex and you must maintain the correct balance between high-level features and total chip area.

## Multi-Gigabit Transceivers

Following the industry-wide migration from parallel to serial interfaces, Xilinx introduced multi-gigabit transceivers (MGTs) to meet bandwidth requirements as high as 6.5 Gbps.

The common functional blocks in these transceivers are an 8B/10B encoder/decoder, transmit buffer, SERDES, receive buffer, loss-of-sync finite state machine (FSM), comma detection, and channel bonding logic. These transceivers have built-in clock data recovery (CDR) circuits that can perform at gigahertz rates. Built-in phase-locked loops (PLL) generate the fabric and transceiver clocks.

Transceivers have several advantages:

- With their self-synchronous timing models, they reduce the number of traces on the boards and eliminate clock-to-data skew
- Multiple MGTs can achieve higher bandwidths

- MGTs with point-to-point connections make switched fabric architectures possible
- The elastic buffers available in MGTs provide high skew tolerance for channel-bonded lanes

MGTs are designed for configurable support of multiple protocols; hence their control is fairly complex. When designing or integrating designs with a high-speed connectivity solution using MGTs, you will have to consider MGT initialization, alignment, channel bonding, idle sequence generation, link management, data delineation, clock skew, clock compensation, error detection, and data striping and destriping. Configuring transceivers for a particular application is challenging, as you are expected to tune more than 200 attributes.

## The Aurora Solution

The Xilinx® Aurora protocol and its associated designs address these challenges by managing the MGT's control interface.

Aurora is free, small, scalable, and customizable. With low overhead, Aurora is a protocol-agnostic, lightweight, link-layer protocol that can be implemented in any silicon device/technology.

With Aurora, you can connect one or more MGTs to form a communication channel. The Aurora protocol defines the structure of data packets and procedures for

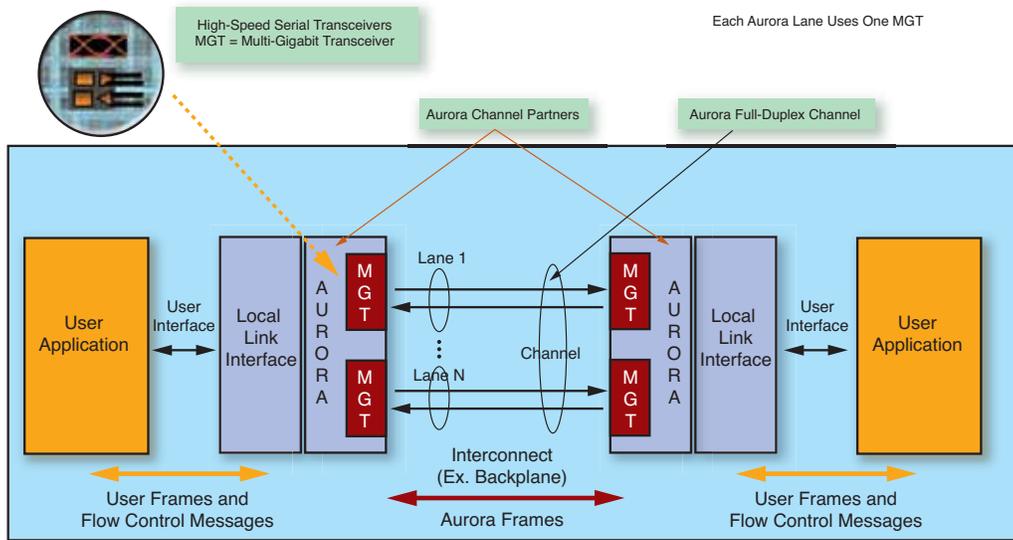


Figure 1 – Connectivity scenario using Aurora

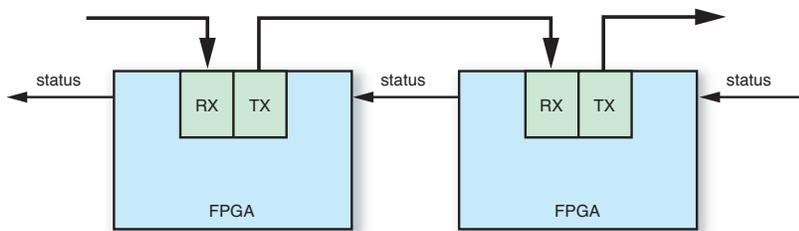


Figure 2 – Simplex token ring structure

flow control, data striping, error handling, and initialization to validate MGT links.

Aurora shrink-wraps MGTs by providing a transparent interface to them, allowing the upper layers of proprietary or industry-standard protocols such as Ethernet and TCP/IP to ride on top of it and provide easily access.

This easy-to-use pre-defined protocol needs very little time to integrate with existing user designs. Being lightweight, Aurora does not have an addressing scheme and cannot support switching. It does not define correction within data payloads.

Aurora is defined for physical and data-link layers in the Open Systems Interconnection (OSI) model and can easily integrate into existing networks.

**A Typical Connectivity Scenario**

Figure 1 is an overview of a typical Aurora application. The Aurora interface transfers data to and from user application functions through the user interface. The

user interface is not part of the Aurora protocol specification.

The Aurora protocol engine converts generic arbitrary-length data from the Xilinx LocalLink user interface to Aurora protocol-defined frames and transfers it across channel partners comprising one or more high-speed serial links. The number of links between channel partners is configurable and device-dependent.

Most Xilinx IPs are developed based on the legacy LocalLink interface. Any user interface designed for other LocalLink-based IPs can be directly plugged into Aurora. For more information on the LocalLink interface, visit [www.xilinx.com/products/design\\_resources/conn\\_central/localink\\_member/sp006.pdf](http://www.xilinx.com/products/design_resources/conn_central/localink_member/sp006.pdf).

You can think of Aurora as a bridge between the LocalLink interface and the MGT.

Aurora’s LocalLink interface is customizable for 2- or 4-byte data. Your selection of

a 2- or 4-byte interface should be based on the throughput requirements and latency involved. A 4-byte design has more latency than a 2-byte design, but offers better throughput and consumes less resources than an equivalent 2-byte design.

Aurora channels can function as uni-directional (simplex) or bi-directional (full-duplex). Full-duplex module initialization data comes from the channel partner through a reverse path, whereas in simplex the initialization happens through four side-band signals. Aurora is available in simplex TX only, RX only, or both. “Both” operation is like full duplex, except that the

transmit and receive sections communicate independently. You can use Aurora simplex in token-ring fashion. Figure 2 shows the ring structure of Aurora-S (simplex).

**Data Flow in Aurora**

Data is transferred between Aurora channel partners as frames. Data flow primarily comprises the transfer of protocol data units (PDUs) between the user application and the Aurora interface, as well as the transfer of channel PDUs between channel partners.

Soon after power-up, when the core comes out of all reset states, the transmitter sends initialization sequences. If the link is fine and the link partner recognizes these patterns, it sends acknowledgement patterns. After receiving sufficient acknowledgement patterns, the transmitter transits to a link-up state, indicating that the individual transceiver connection between channel partners is established. Once the link is up, the Aurora protocol engine moves to a channel verification stage for a single-lane channel, or a channel-bonding stage (preceeding a verification stage) for a multi-lane channel. When channel verification is complete, Aurora sends a channel-up signal, which is followed by actual data transmission. The link initialization process is shown in Figure 3.

**Frame Types**

As stated previously, data is sent over Aurora channels as frames. There are five frame types, listed in the order of their priority:

- Clock compensation (CC)
- Initialization sequences
- Native flow control (NFC)
- User flow control (UFC)
- Channel PDUs
- Idles

Aurora allows channel partners to use separate reference clocks by inserting CC sequences at regular intervals. It can accommodate differential clock rates up to 200 parts per million (ppm) between the transmitter and the receiver.

An Aurora frame sends data in a quantity of two bytes called a symbol. In case the number of bytes in the PDU is odd, it appends one extra byte called a pad. On the receive side, the pad is removed by the receive logic and data is presented to the LocalLink interface.

The transmit LocalLink frame is encapsulated in the Aurora frame, as shown in Figure 4. Aurora encapsulates frames with start-channel PDU (SCP) and end-channel PDU (ECP) symbols. On the receive side, the reverse process occurs. The transceivers are responsible for encoding/decoding the frame into 8B/10B characters, serialization, and de-serialization.

**Flow Control**

Aurora supports an optional flow control mechanism to prevent data loss caused by different source and sink rates between channel partners (Figure 5).

**Native Flow Control**

NFC is meant for data-link layer-rate control. NFC operation is governed by two NFC FSMs: RX and TX.

The RX NFC FSM monitors the state of the RX FIFO. When there is a risk of overflow, it generates NFC PDUs, requesting that the channel partner pause transmission of user PDUs for a specific time duration. The TX NFC state machine responds by waiting during the requested time, allowing the RX

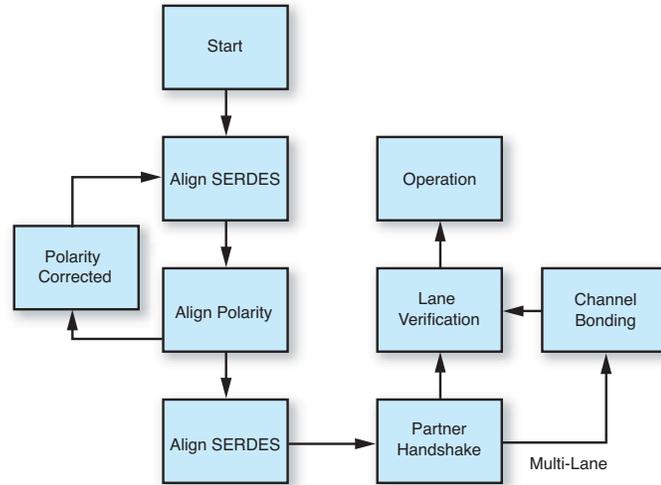


Figure 3 – Data flow in Aurora

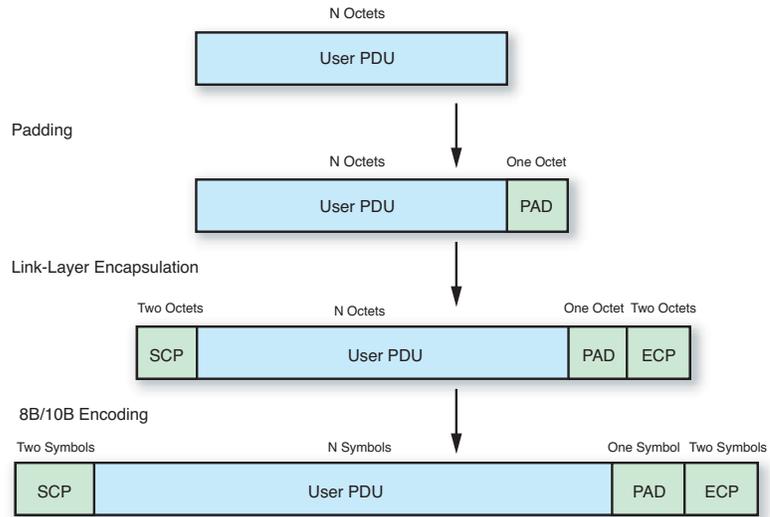


Figure 4 – Frame encapsulation in transmit Aurora

FIFO to come out of its overflow state.

While sending NFC requests, the TX NFC FSM should eliminate any round-trip delay. Ideally, NFC requests are sent before receive FIFO overflows to account for this delay. You can program the NFC pause value from 0 to 256; the maximum pause value is infinite. An NFC pause value is non-cumulative and new NFC request values override the old.

There are two NFC request types: immediate mode and completion mode. In immediate mode, an NFC request is processed while low-priority requests are halted. In completion mode, the current frame is

transmitted; only after completion of the transfer is the NFC request processed.

**User Flow Control**

UFC is used to implement user-defined flow control at any layer. The number of UFC messages can be 2 to 16 bytes in a frame. UFC messages are generated and interpreted by the user application.

**Applications**

Aurora is a simple, scalable open protocol for serial I/O. It can be implemented in FPGAs or ASICs. Aurora can be applied where an inexpensive, high-performance

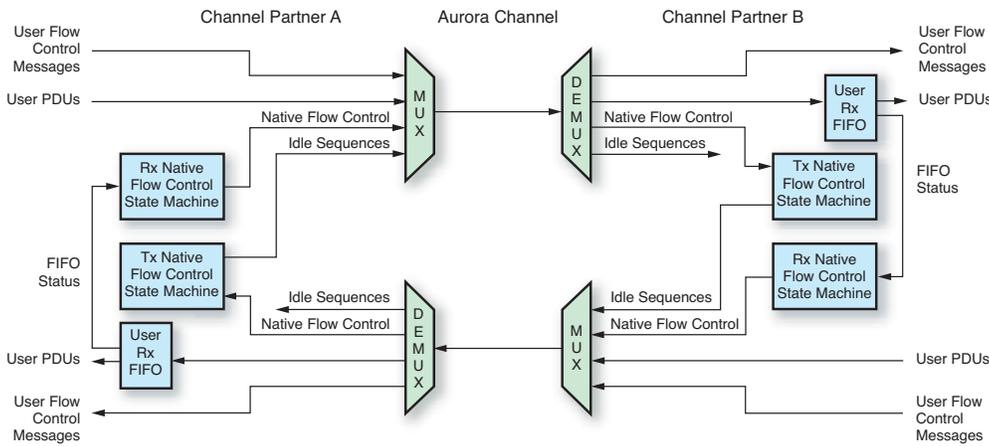


Figure 5 – NFC operation in Aurora

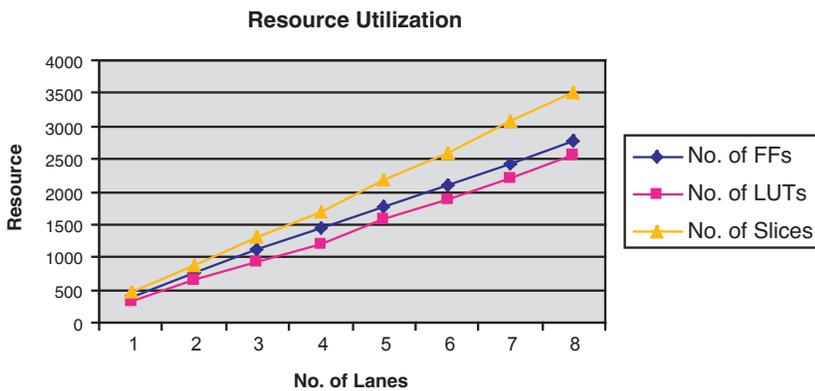


Figure 6 – Performance curves for various Aurora flavors

link layer with little resource consumption is required, saving many hours of work for users who were considering developing their own MGT protocols.

Aurora has a variety of applications, including:

- Video
- Medical
- Backplane
- Bridging
- Chip-to-chip and board-to-board communication
- Splitting functionality among several FPGAs
- Simplex communication

Aurora provides more throughput for fewer resources. Plus, it can adapt to customer needs, allowing additional func-

tions on top. Aurora simplex operation offers the most flexibility and the least resource use where users require high throughput in one direction. Simplex is ideal for non-data communication like streaming video.

### Performance Statistics

Aurora is designed to use minimal FPGA resources. A single-lane 2-byte framing design consumes approximately 383 look-up tables and 374 flip-flops. The curves in Figure 6 show the resource utilization statistics for different lane configurations on a Virtex™-5 LXT device.

The latency for a single-lane 2-byte design is 10 cycles. Transceivers insert some inherent latency. Thus, the overall latency is approximately 29 cycles for an Aurora design on a RocketIO™ GTP transceiver.

You have the freedom to choose reference clocks from a range of values. The overall throughput depends on the reference clock value and the line rate chosen.

### Aurora: CORE Generator IP

Aurora is released as a part of CORE Generator™ software, with about 10 configurable parameters. You can configure an Aurora core by selecting streaming/framing interface, simplex/full-duplex data flow, single/multiple MGTs, reference clock value, line rate, MGT location(s) based on number of MGTs selected, and reference clock source. The supported line rate can range from 0.1 Gbps to 6.5 Gbps depending on the Virtex device selected. The core comes with simulation scripts supporting MTI, NC-SIM, and VCS simulators and build scripts to ease design synthesis and bit generation.

### Conclusion

Aurora is easy to use. The IP is efficient, scalable, and versatile enough to transport legacy protocols and implement proprietary protocols. Aurora is also backward-compatible: an Aurora design in a Virtex-5 device can talk to an Aurora design in a Virtex-4 device, making the IP independent of underlying MGT technologies. Xilinx IP and software tools allow you to take advantage of Aurora's improved feature set. One future enhancement currently in the research phase is the inclusion of a packet-retry feature to provide link reliability.

For more information, visit [www.xilinx.com/aurora](http://www.xilinx.com/aurora).

### TAKE THE NEXT STEP (Digital Edition: [www.xcellpublications.com/subscribe/](http://www.xcellpublications.com/subscribe/))

- Download the Aurora protocol specification and bus functional model
- Order your free Aurora LogiCORE design
- Read about Aurora application examples

# 30% faster than last year's model...



Here are 6 of the new, faster, bigger, Virtex-5 FPGAs on a 12 Million ASIC Gate Board that offers unmatched performance to ASIC Prototypers, IP Designers, and FPGA Developers. The V5 65nm process, with 6 input LUT and advanced interconnect, enables 30% faster clock speeds in your application. The Dini DN9000k10PCI captures this performance on an easy to use board with these handy features:

- 33/66 MHz PCI bus or stand-alone operation
- 6 - DDR2 SODIMM Sockets
- 7 - Global Clock Networks
- 3 - 400pin FCI-MEG Connectors for daughter cards
- Easy configuration via CompactFlash, USB or PCI

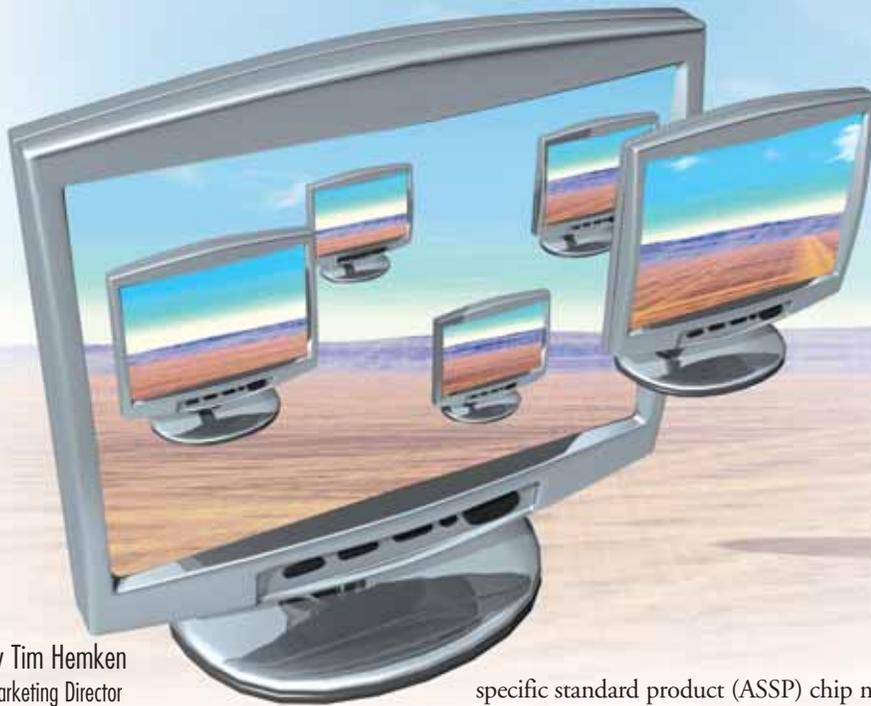
All necessary operating software, including reference designs and Synplicity Certify™ models to simplify partitioning, is supplied with the board. If your need is speed — visit The Dini Group web site [www.dinigroup.com](http://www.dinigroup.com) for complete details on the fastest FPGA board ever.

The  
**DiNI**  
Group

1010 Pearl Street, Suite 6  
La Jolla, CA 92037  
(858) 454-3419  
[sales@dinigroup.com](mailto:sales@dinigroup.com)

# Xilinx FPGAs Adapt to Ever-Changing Broadcast Video Landscape

New digital broadcast standards and applications are addressed by advanced silicon, software, and free reference designs available with Xilinx FPGAs.



by Tim Hemken  
Marketing Director  
Xilinx, Inc.  
[themken@xilinx.com](mailto:themken@xilinx.com)

The serial digital interface (SDI) protocol for transporting uncompressed standard definition video evolved when broadcast studios wanted to convert from analog audio and video to digital audio and video without replacing the enormous coaxial transmission cable infrastructure. Today, the ever-increasing screen resolutions and associated data rates have spawned new serial data communication formats with the same goal of coax reuse.

The first such standard for SDI, authored by the Society of Motion Picture and Television Engineers (SMPTE), is known as SMPTE 259M, commercialized in 1989. At its introduction, the primary chips used for the interface were provided by application-

specific standard product (ASSP) chip makers. The data rate for SDI is nominally 270 Mbps, adequate for standard definition television (SDTV) resolutions.

In 2002, Xilinx announced the immediate availability of Virtex™-II Pro FPGAs. The feature set of this device included multi-gigabit transceivers (MGTs) capable of operating at bit rates as fast as 3.125 Gbps. About the same time, the broadcast studios were starting their adoption of the new high-definition television (HDTV) standards with greater screen resolutions and higher data-rate requirements. SMPTE authored a standard known as SMPTE 292M, supporting the serial transmission of uncompressed HDTV video content at a nominal rate of 1.5 Gbps, known as HD-SDI.

Xilinx® Senior Staff Applications Engineer John Snow realized that the two

data rates could be supported by a single Virtex™-II Pro MGT and saw an opportunity to integrate multiple ASSP chips into a single Xilinx FPGA. The integration would vastly reduce the cost of these interfaces, especially in video switcher and master controller designs where there are multiple video streams.

Snow authored the first application notes for these two interface standards using FPGAs. These application notes included free reference designs written in Verilog and VHDL source code. The code and documentation allowed broadcast system engineers to easily implement fully featured SDI and HD-SDI receivers and transmitters and other associated functions, such as video test pattern generators, in Virtex-II Pro FPGAs. Over the next year, Virtex-II Pro FPGAs began to appear in broadcast equipment throughout the world, consolidating multiple ASSP chips.

HDTV's acceptance among the masses continues to grow. This amazing technology brings viewers closer to reality and, as a result, the amount of HD content is increasing. Today, HD-capable receivers are achieving mass-market price points and consumers are buying HDTVs at accelerating rates. The multi-rate SDI and HD-SDI reference designs are increasingly important for distributing digital audio and video content inside the broadcast studio.

Today, the SMPTE organization is not standing still. They consistently publish new standards to handle video formats that

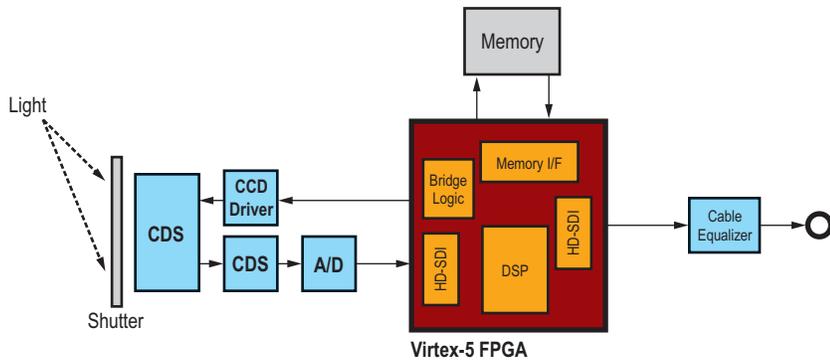


Figure 1 – CCD camera

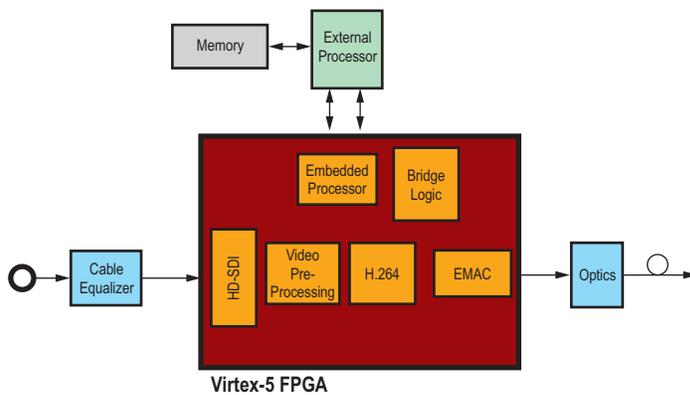


Figure 2 – Video over IP

require higher bandwidth. Two of the latest standards are known as dual-link HD-SDI (SMPTE 372M) and 3G-SDI (SMPTE 424M and SMPTE 425M), both providing 3 Gbps of total bandwidth.

The dual-link HD-SDI standard uses two HD-SDI rate links combined together to facilitate the transfer of richer color (more pixel color data) or faster update rates (1080 lines at a 60-Hz progressive frame rate as opposed to a 30-Hz frame rate). The two coax cables forming a dual-link HD-SDI interface can be replaced with a single coax cable using 3G-SDI.

An example of an application requiring the increased data rates is driven by the cinema business as it migrates to digital data. Digital cinema standards use 36 bits of data per video sample, compared to the 20 bits per sample typically used by HDTV formats. The increased number of bits per video sample coupled with higher screen resolutions results in market demand for

digital interfaces running at 3 Gbps.

Even within a single SMPTE standard there are opportunities for FPGA “future proofing.” 3G-SDI is actually defined by two standards: SMPTE 424M and SMPTE 425M. SMPTE 424M defines the physical and electrical characteristics of the serial interface itself. SMPTE 425M defines how to map various video formats to the interface. Even though it was only published in 2006, SMPTE is already at work modifying SMPTE 425M to accommodate additional video formats.

The SMPTE organization has already defined interfaces that will run at 10 Gbps and is contemplating interfaces that will run even faster. Designing these interfaces with Xilinx FPGAs eliminates the risk that a newly designed piece of video equipment will be outdated by quickly evolving standards before it ships.

Let’s look at a few emerging applications that use HD-SDI for video connectivity.

### CCD Cameras

Ultra high-speed, high-sensitivity broadcast cameras capable of capturing clear, smooth, slow-motion video – even in limited lighting – are extremely useful for recording events such as professional baseball games played at night. These cameras can capture fast-moving phenomena that cannot be perceived clearly with the naked eye, such as the footage of a ball’s impact with a bat. Showing these events in slow-motion video improves the viewer experience significantly.

The CCD (charge-coupled device) camera shown in Figure 1 uses an FPGA to perform signal synthesis processing and color processing, as well as interfacing to the CCD driver. The CCD driver in turn drives the CCD, mechanical shutter control, and trigger control. The incoming video signal is converted to digital format by the analog-to-digital converter and then stored in off-chip memory. When the data transfer for an entire frame is complete, the data from the memory is synthesized by the FPGA and sent over the network using HD-SDI. The processing time required from trigger to HD-SDI output is one second or less. The FPGA also controls the memory and the ADC.

### Video over IP

Some video production centers are starting to use Ethernet to transmit crystal-clear HD streams across the network. Images are pre- and post-processed to enhance picture quality in real time with low latency and then transported over the network using various encoding and decoding standards (codecs). The data must be compressed, as the stream size and rate are very high. For example, a transmission of 1920 x 1080 pixels at 30 fps requires a data rate of 1.5 Gbps uncompressed. Add to that multiple channels and the rate goes even higher.

Application-optimized FPGAs with embedded DSP blocks, on- and off-chip memories, abundant logic to build bridging functions, and Ethernet and HD-SDI connectivity are the ideal solutions to build such systems. Figure 2 shows a block diagram of a video over IP system. The FPGA reads the data presented over an HD-SDI

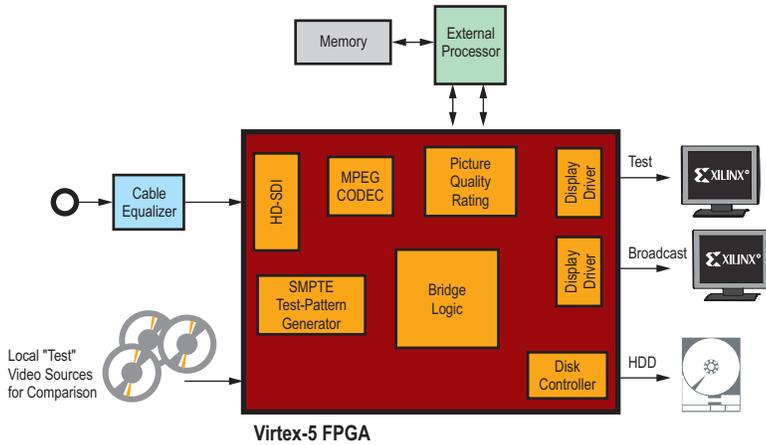


Figure 3 – HDTV picture quality monitor

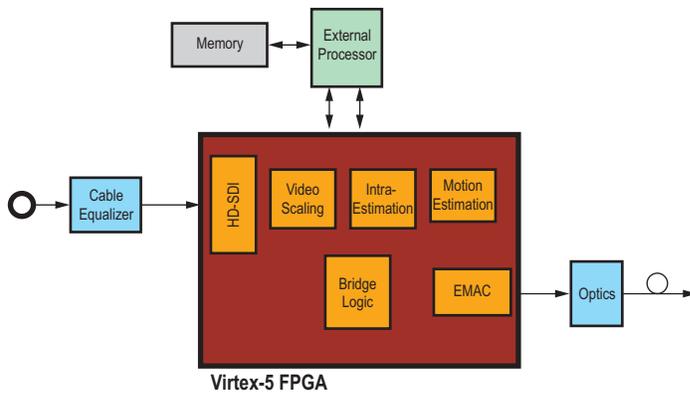


Figure 4 – Real-time HD AVC

link and then processes it. A codec, such as H.264, is used to compress the data. The data is then converted into Ethernet packets with the appropriate header information for decoding at the receiving end and finally sent over Ethernet links using a MAC.

### HDTV Picture Quality Monitor

Previously, consumers could access high-quality video and multi-channel audio through DVDs only. As HD broadcasts become commonplace, comparisons with DVDs are natural. As a result, viewers are more aware of picture quality, particularly for HD. Picture quality is likely to become a major differentiator between service providers. Picture quality monitors for objective and subjective testing of picture quality in conjunction with perceptible error measurements are now required.

Figure 3 shows a picture quality monitor implemented in a Xilinx Virtex-5 FPGA. Objective testing is performed using commonly used test pattern formats

defined by SMPTE RP 198. Subjective testing is performed by comparing the broadcast feed with local test video sources. The FPGA collects the data from different sources, pre-processes it, and then sends it to an external processor for analysis.

### Real-Time HD AVC

Advanced video coding (AVC) is a video compression technique used to deliver video at half the required bit rates. AVC was first used with standard-definition videos, but is even more compelling for HD service providers. AVC achieves its most significant gains over MPEG-2 through substantial improvements to the

motion-compensated prediction process. AVC doubles the accuracy of motion prediction, uses smaller block sizes to allow objects to be tracked more accurately, and has many more reference frames to search for a good motion-predictive match. Thus, a real-time high-definition AVC video encoder can deliver broadcast-image quality at half the bandwidth of MPEG-2.

FPGAs perform the computationally intensive motion estimation task, as shown in Figure 4. Motion estimation is performed using the repeated sum of absolute difference calculations. The data comparisons are very repetitive and many of the calculations are reused. CPU-based implementations tend to struggle to feed the arithmetic logic units from cache; FPGA designs can be customized to retain all of the values in a custom register pipeline.

### Conclusion

The newest devices, such as Xilinx Virtex-5 FPGAs, have enormous amounts of logic and offer ASIC-like levels of performance. FPGAs pack in all of the features that broadcast equipment designers want:

- Embedded low-power 3.2-Gbps transceivers capable of supporting several standards such as SDI, HD-SDI, dual-link HD-SDI, 3G-SDI, DVB-ASI, AES digital audio, Ethernet, and PCI Express
- High-speed DSP blocks
- Embedded processors
- Ethernet MACs
- PCI Express cores
- Many video IP cores

By designing video connectivity applications in Xilinx silicon products, broadcast equipment manufacturers can lower costs, differentiate their products from the competition, and lower the inherent risk caused by changing standards. ●●●

#### TAKE THE NEXT STEP (Digital Edition: [www.xcellpublications.com/subscribe/](http://www.xcellpublications.com/subscribe/))

- Learn more about Xilinx solutions for video and audio interfacing, real-time HD video processing, video and audio codecs, forward error correction (FEC) and modulation, and other broadcast system functions.
- Read audio, video, and image processing application notes.
- Purchase the Xilinx serial digital video board for HD-SDI, SDI, and DVB-ASI.

# ASIC PROTOTYPING

## HAPS™ – High-performance ASIC Prototyping System™

a high performance, high capacity FPGA platform for ASIC prototyping and emulation composed of multi-FPGA boards and standard or custom-made daughter boards

## HapsTrak™

a set of rules for pinout and mechanical characteristics, which guarantees compatibility with previous and future generation HAPS motherboards and daughter boards



HAPS-50



HAPS-30



HAPS-20



HAPS-10



Synplicity®

Simply Better Results

[www.synplicity.com](http://www.synplicity.com), [haps@synplicity.com](mailto:haps@synplicity.com)

Synplicity, Inc. 600 West California Avenue, Sunnyvale, CA 94086

Phone: (U.S.) +1 408 215-6000

Copyright © 2007 Synplicity, Inc. All rights reserved. Specifications subject to change without notice. Synplicity, the Synplicity logo, and "Simply Better Results" are registered trademarks of Synplicity, Inc. HAPS, "High-performance ASIC Prototyping System", and HapsTrak are trademarks of Synplicity, Inc. Virtex-II, Virtex-II Pro, Virtex-4, and Virtex-5 are registered trademarks of Xilinx Inc.

# FREE on-line tutorials with Demos On Demand



**A** series of compelling, highly technical product demonstrations, presented by Xilinx experts, is now available on-line. These comprehensive videos provide excellent, step-by-step tutorials and quick refreshers on a wide array of key topics. The videos are segmented into short chapters to respect your time and make for easy viewing.

## **Ready for viewing, anytime you are**

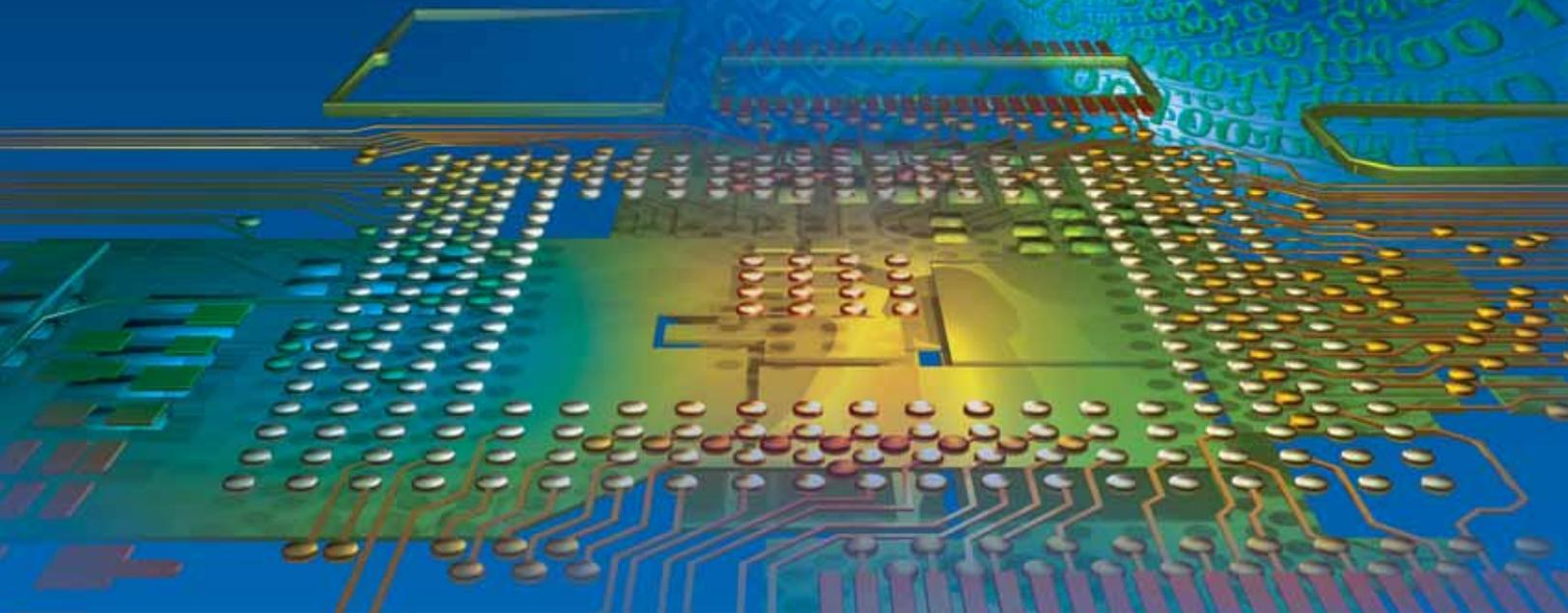
Offering live demonstrations of powerful tools, the videos enable you to achieve complex design requirements and save time. A complete on-line archive is easily accessible at your fingertips. Also, a free DVD containing all the video demos is available at [www.xilinx.com/dod](http://www.xilinx.com/dod). Order yours today!





# Serial RapidIO Connectivity Enhances DSP Co-Processing

The Virtex-5 LXT FPGA-based SRIO IP solution significantly enhances connectivity by providing a true bi-directional data flow.



by Navneet Rao  
Technical Marketing Manager, Platform Solutions  
Xilinx, Inc.  
[navneet.rao@xilinx.com](mailto:navneet.rao@xilinx.com)

Today, the demand for high-speed communication and super-fast computing is spiraling. Wired and wireless communication standards are being deployed everywhere and the data-processing infrastructure is scaling everyday. The pervasive means of wired communication is Ethernet through LAN, WAN, and MAN networks. The pervasive means of wireless communication is through cell phones, enabled by infrastructures using DSP. What was primarily a means of voice connectivity – the phone – now caters to the ever-increasing demands for voice, video, and data.

System designers must create architectures that will alleviate the exorbitant demands of triple-play scenarios while meeting such requirements as:

- High performance
- Low latency
- Lower system costs (NRE included)
- Scalable, extensible architectures
- Integrating off-the-shelf (OTS) components
- Distributed processing
- Support for multiple standards and protocols

What emerges from these challenges are two primary themes: connectivity

between compute platforms/boxes in wired or wireless infrastructures and the individual computing resources in these platforms/boxes.

## Connectivity Between Compute Platforms

Standards-based connectivity is common today. Parallel connectivity standards (PCI, PCI-X, EMIF) may be able to meet current demands, but will come up short when scalability and extensibility are involved. With the advent of packet-based processing, clearly the trend is toward high-speed serial connectivity (Figure 1).

The desktop and networking industries have adopted standards like PCI Express (PCIe) and Gigabit Ethernet/XAUI. Data-processing systems in wireless infra-

structures, however, have slightly different interconnect requirements:

- Low pin counts
- Backplane chip-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. As such, SRIO has become the dominant interconnect for data-plane connectivity in wireless infrastructure equipment.

SRIO networks are built around two basic blocks – endpoints and switches (Figure 2). Endpoints source and sink packets, while switches pass packets between ports without interpreting them.

SRIO is specified in a three-layer architectural hierarchy (Figure 3):

- The physical layer specification describes device-level interface specifics such as packet transport mechanisms, flow control, electrical characteristics, and low-level error management.
- The transport layer specification provides the necessary route information for a packet to move from endpoint to endpoint. Switches operate at the transport layer by using device-based routing.
- The logical layer specification defines the overall protocol and packet formats. All packets are 256 payload bytes or less. The transactions use load/store/DMA operations targeted to a 34-/50-/66-bit address space. The transactions include:
  - NREAD – read operation (data returned is the response)
  - NWRITE – write operation, no response

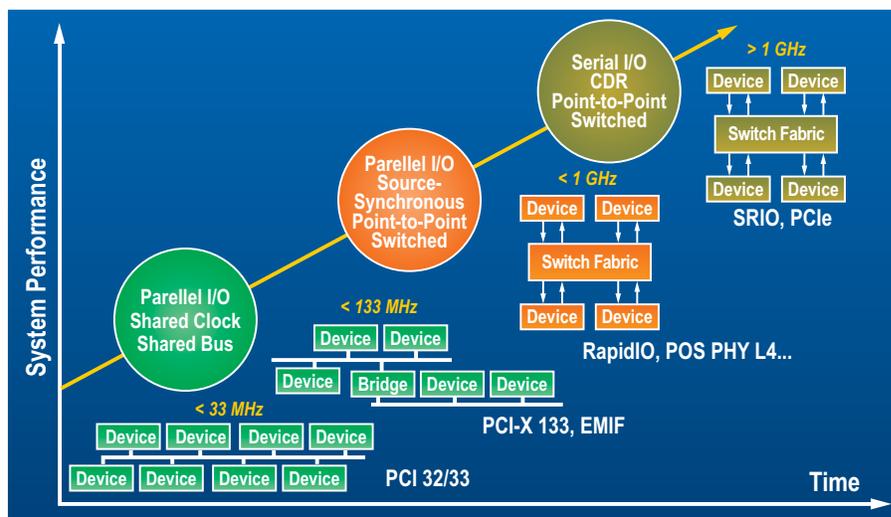
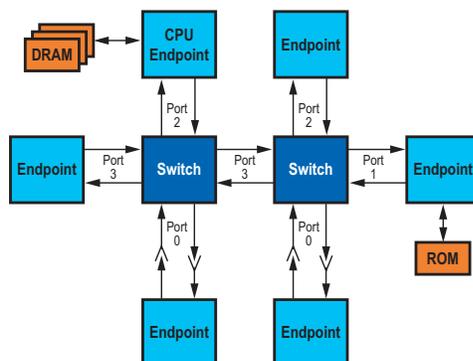


Figure 1 – Trend toward serial connectivity



- SRIO networks are built around two basic blocks:
  - Endpoints
  - Switches
- Endpoints source and sink packets
- Switches pass packets between ports without interpreting them
- All devices support maintenance transactions for access to configuration registers

Figure 2 – SRIO network building blocks

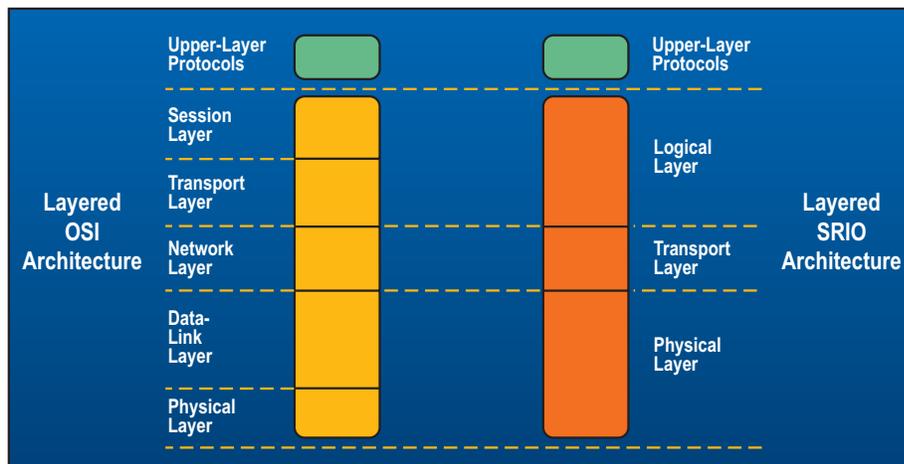


Figure 3 – Layered SRIO architecture



- NWRITE\_R – robust write, with response from the target endpoint
- SWRITE – streaming write
- ATOMIC – atomic read-modify-write
- MAINTENANCE – system discovery, exploration, initialization, configuration, and maintenance operations

### SRIO – An Advantage Scenario

A four-lane SRIO link running at 3.125 Gbps can deliver 10 Gbps throughput with full data integrity. Because SRIO is similar to microprocessor buses – memory and device addressing instead of the software management of LAN protocols – packet processing is implemented in hardware. This means significantly lower I/O processing overhead, lower latency, and increased system bandwidth. But unlike most bus interfaces, SRIO has low pin count interfaces and scalable bandwidth based on 3.125-Gbps links.

### Computing Resources in Platforms

Today's applications demand greater processing resources. Hardware-based implementations are gaining traction. Compression/decompression algorithms, firewall applications like anti-virus and intrusion detection, and security applications requiring encryption engines like AES, Triple DES, and Skipjack have been targeted for hardware implementations after being initially implemented in software. This demands a massive parallel ecosystem of shared bandwidth and processing power. Shared or distributed processing harnessing through systems using CPUs, NPUs, FPGAs, or ASICs is required.

Considering all of these application-specific requirements for building a future-proof system, the requirements for computing resources include:

- Multiple hosts – distributed processing
- Direct peer-to-peer communications
- Multiple heterogeneous OSs
- Complex topologies:
  - Discovery mechanisms
  - Redundant paths (failover)

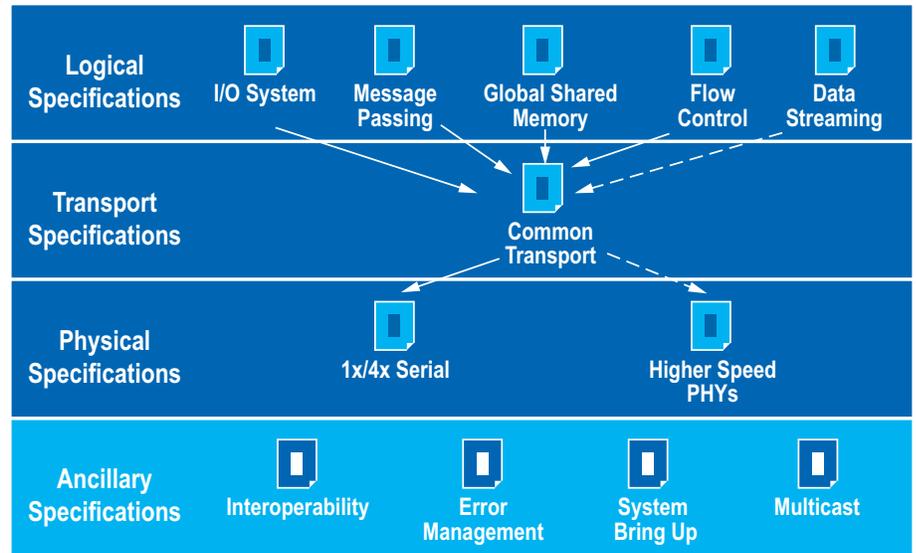


Figure 4 – SRIO specification

- Ability to support high reliability:
  - Lossless protocol
  - Automatic retraining and device synchronization
  - System-level error management
- Ability to support communications data plane:
  - Multicast
  - Traffic-managed (lossy) operation
  - Link, class, and stream-based flow control
  - Protocol interworking
  - Higher transaction concurrency
- Modular and extendable
- Broad ecosystem support

The SRIO protocol was designed to support all of the disparate requirements driven by compute devices in wireless infrastructures.

The SRIO specification (Figure 4) defines a packet-based layered architecture to support multiple domains or market segments for system architects to design next-generation computing platforms. Features like architectural independence, ability to deploy scalable systems with carrier-grade reliability, advanced traffic management, and provisioning for high performance and

throughput can easily be accomplished by adopting SRIO as the computing interconnect. Furthermore, a broad ecosystem of vendors makes the selection of OTS parts and components easy.

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 segmentation and reassembly functions
- A traffic-management framework by enabling millions of streams, support for 256 traffic classes, and lossy operations
- Flow control to support multiple transaction request flows, 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 hardware (daisy-chain) topologies through system discovery, configuration, and bring-up, including support for multiple hosts
- Error management and classification (recoverable, notification, and fatal)

### Xilinx IP Solutions for SRIO

Xilinx endpoint IP solutions for SRIO are designed to RapidIO specification v1.3. The complete Xilinx endpoint IP solution for SRIO comprises the following components (Figure 5):

- The Xilinx endpoint IP for SRIO is a soft LogiCORE™ solution. It supports fully compliant maximum-payload operations for both sourcing and receiving user data through target and initiator interfaces on the logical (I/O) and transport layers.
- The buffer layer reference design is provided as source code to perform 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, the high-speed transceivers are instantiated in the physical layer IP to support one- and four-lane SRIO bus links at line rates of 1.25 Gbps, 2.5 Gbps, and 3.125 Gbps.
- The 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.

The complete Xilinx endpoint IP LogiCORE solution for SRIO has been exhaustively tested, hardware validated, and is undergoing interoperability testing with leading SRIO device vendors. The LogiCORE IP is delivered through the Xilinx CORE Generator™ software GUI tool that enables user customization for baud rates, endpoint configuration, and support for extended features like flow control, re-transmit suppression, doorbell, and messaging. This enables you to create a flexible, scalable, and customized SRIO endpoint IP optimized for your application.

### Virtex-5 FPGA Computing Resources

The Xilinx endpoint IP for SRIO ensures that high-speed connectivity is established between the link partner using the SRIO protocol. The IP consumes <20% of the available logic resources in the smallest Virtex™-5 device, thereby ensuring that the user design has access to the most logic/memory/I/Os for targeting a system application. Let's review the resources in Virtex-5 devices.

### Logic Blocks

The Virtex-5 logic architecture, with six-input look-up tables (LUTs) based on the 65-nm process, offers the highest FPGA capacity. Along with improved carry logic, this provides a 30% performance benefit over previous devices. Power consumption is significantly lowered

because fewer LUTs are required, and the device has a highly optimized symmetric routing architecture.

### Memory

Virtex-5 memory solutions include LUT RAMs, block RAMs, and memory controllers for interfacing to large memories. The block RAM structure includes pre-engineered FIFO logic – embedded error checking and correction (ECC) logic that can be used for external memories. In addition, Xilinx provides comprehensive design resources to instantiate memory controller blocks in a system design through the Memory Interface Generator (MIG) tool. This enables you to leverage hardware-verified solutions and focus your efforts on other crucial sections of your designs.

### Parallel and Serial I/Os

SelectIO™ technology is capable of virtually any parallel source-synchronous interface the customer needs in the design. Using the SelectIO interface, you can easily create industry-standard interfaces for more than 40 different electrical standards or proprietary interfaces. The SelectIO interface offers maximum rates at 700 Mbps single-ended and 1.25 Gbps differential.

All Virtex-5 LXT FPGAs have a GTP transceiver capable of running at speeds from 100 Mbps to 3.2 Gbps. The GTP transceiver is also one of the industry's lowest power MGTs, with less than 100 mW per transceiver. The design flow process for high-speed serial design is alleviated by the introduction of proven design techniques and methodologies to simplify the design.

In addition, new design tools (RocketIO™ transceiver wizard and IBERT) and new silicon capabilities (TX and RX equalization and built-in pseudo-random bit sequence (PRBS) generator and checker) allow you to exploit the features and benefits of migrating architectures from parallel I/O standards to more than 30 serial standards and emerging serial technologies.

### DSP Blocks

Each DSP48E slice can provide 550-MHz performance, enabling you to create appli-

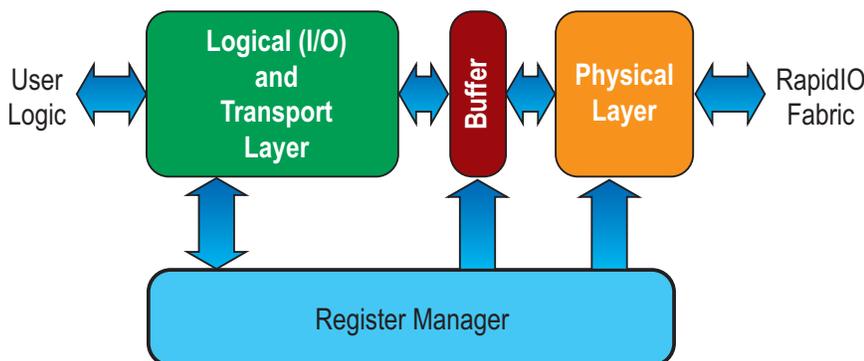


Figure 5 – Xilinx endpoint IP architecture for SRIO

cations that require single-precision floating-point capability like multimedia, video and imaging applications, and digital communications. This provides expanded functionality compared to previous devices as well as providing a power advantage by reducing dynamic power consumption by more than 40%. The number of DSP48E slices has also been increased in Virtex-5 FPGAs, which optimizes the ratio of these blocks relative to the available logic resources and memory.

### Integrated I/O Blocks

All Virtex-5 LXT FPGA devices have one endpoint block for implementing a PCIe function. This hard IP endpoint block enables easy scalability from x1 to x2 and x4 or x8 painlessly with easy reconfigurability. The block has also passed stringent PCI-SIG compliance and interoperability tests for x1, x4, and x8 links, considerably easing the user adoption for PCIe.

All Virtex-5 LXT FPGA devices also have four tri-mode Ethernet media access controllers (TEMACs) capable of 10-/100-/1,000-Mbps speeds. The block provides dedicated Ethernet functionality, which together with Virtex-5 LXT RocketIO transceivers and SelectIO technology enables you to connect to a wide variety of network devices.

Using the two integrated I/O blocks for PCIe and Ethernet, you can create a range of customized packet processing and network products that provide a significant reduction in utilization and power consumption. Using these varied resources available in Xilinx FPGAs, you can easily create and deploy intelligent solutions.

Let's look at some system design examples using SRIO and DSP technologies.

### SRIO Embedded System Application

Consider an embedded system built around x86 architecture-based CPUs. The CPU architecture has been highly optimized and can easily cater to applications requiring "number crunching." You can easily implement algorithms in hardware and software that use CPU resources to perform functions like e-mail, database management, and word processing that do not require extensive multiplication.

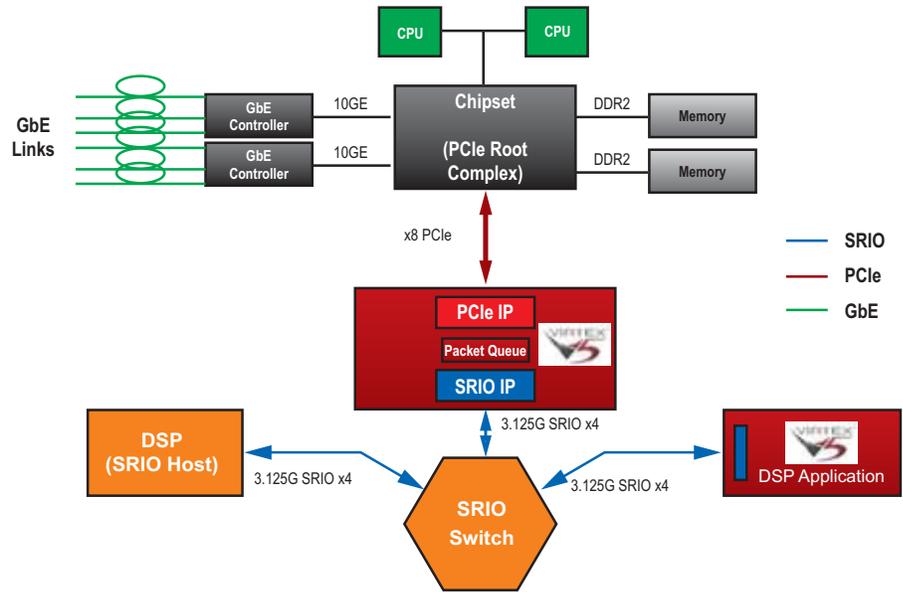


Figure 6 – CPU-based scalable, high-performance embedded system

Performance is measured in millions or billions of instructions/operations per second and efficiency is measured in terms of time/cycles required to complete a specific operation.

High-performance applications requiring a wide range of fixed- and floating-point operations take longer to process the data. Examples include signal filtering, fast Fourier transforms, vector multiplication and searching, image/video analysis and format conversion, and simple number-crunching algorithms. High-end signal processing architectures implemented in DSPs can easily perform these tasks and optimize such operations. The performance of these DSPs is measured in multiply-accumulates per second.

You can easily design embedded systems that use CPUs and DSPs to take advantage of both processing techniques. Figure 6 shows an example system using FPGAs, CPUs, and DSP architectures.

In high-end DSPs, the primary data interconnect is SRIO. In x86 CPUs, the primary data interconnect is PCIe. As shown in Figure 6, FPGAs can easily be deployed for scaling the DSP application or for bridging across disparate data interconnect standards, like PCIe and SRIO.

In the system depicted in Figure 6, the PCIe system is hosted by the root complex

chip set. 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 NReads/NWrites/SWrites.

Designing such bridge functions is easy in Xilinx FPGAs because the back-end interfaces for these Xilinx endpoint functional blocks, PCIe, and SRIO are similar. The "packet-queue" block can then perform the crossover from PCIe to SRIO or vice-versa to establish packet flow across either protocol domain.

### SRIO DSP System Application

In applications where DSP processing is the primary architectural requirement, the system architecture can be designed as depicted in Figure 7.

Virtex-5 FPGA-based DSP processing can act as an intelligent co-processing solution with other DSP devices in the system. The complete DSP system solution can be scaled easily if SRIO is used as the data interconnect. Such solutions can be future-proofed, provide extensibility, and can be supported across multiple form factors. In DSP-intensive applications, fast number

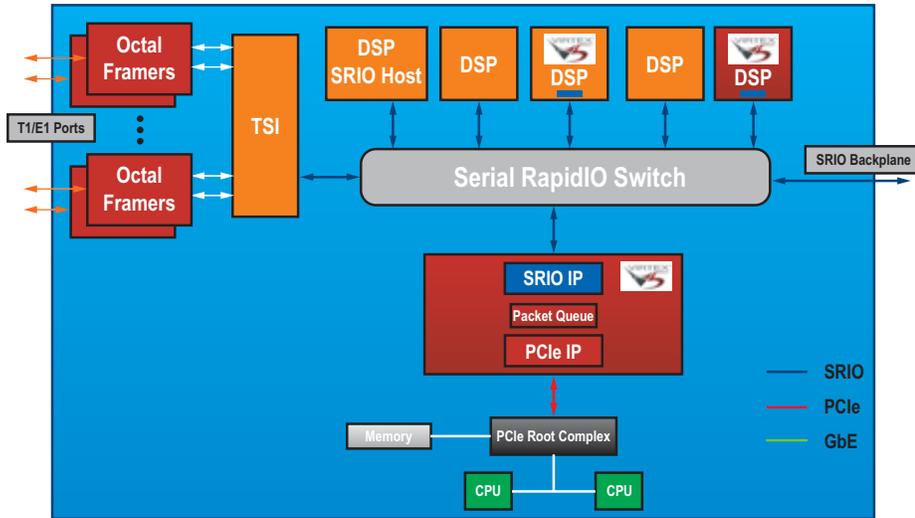


Figure 7 – DSP-intensive farms

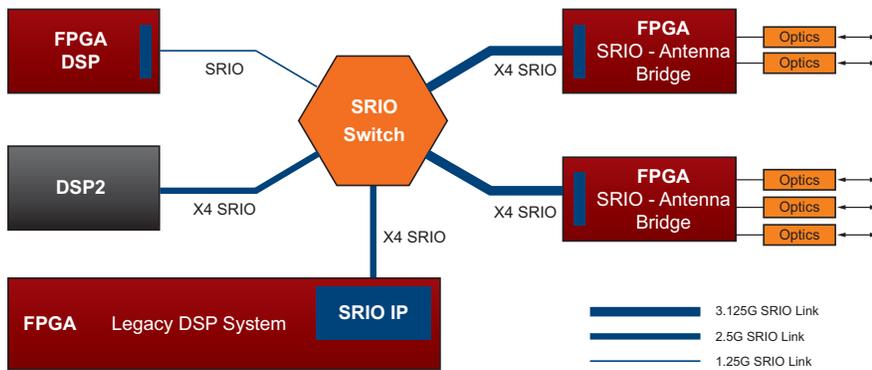


Figure 8 – Scalable baseband uplink/downlink card

crunching or data processing can be accomplished by offloading that processing to the x86 architectures. The Virtex-5 FPGA can be easily used to connect the PCIe subsystem and SRIO architecture to enable efficient function offload.

**SRIO Baseband System Application**

Existing 3G networks are beginning to mature at a rapid pace and OEMs are deploying new form-factors to alleviate specific capacity and coverage problems. To solve such unique challenges to assess market trends, FPGA-based DSP architectures using SRIO as the data-plane standard makes perfect sense. In addition, legacy DSP systems can be quickly provisioned and targeted to new, fast, low-power FPGA DSP

architectures to gain scalability advantages.

As shown in the system depicted in Figure 8, you can design Virtex-5 FPGAs to meet existing demands of line-rate processing of antenna traffic and also provide connectivity to other system resources through SRIO. Migrating existing legacy DSP applications, which have inherently

slow parallel connectivity, is easy because of the SRIO endpoint functions that can be targeted to Virtex-5 FPGAs.

**Conclusion**

SRIO is appearing in a wide array of new applications, largely centered around DSPs in wired and wireless applications. The key advantages of implementing SRIO architectures 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 65-nm technologies
- Easy configurability through the CORE Generator software 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 I/O blocks like PCIe and TEMAC

In addition, Virtex-5 FPGAs have DSP resources that can meet the requirements of existing legacy DSP systems in terms of power, performance, and bandwidth. Additional benefits accrue in terms of system integration – through availability of functional blocks like Ethernet MACs, endpoint blocks for PCIe, processor IP blocks, memory elements, and controllers. Also, you can achieve significant overall system cost savings through the exhaustive list of IP cores to support multiple source aggregation in the FPGA. ●●

**TAKE THE NEXT STEP** (Digital Edition: [www.xcellpublications.com/subscribe/](http://www.xcellpublications.com/subscribe/))

- Learn more about RapidIO
- Evaluate the RapidIO physical layer core
- Evaluate the RapidIO logical I/O and transport layer core
- Sign up for a connectivity training course
- Download the Virtex-5 RocketIO GTP Transceiver User Guide



# The NXP/PLDA Programmable PCI Express Solution

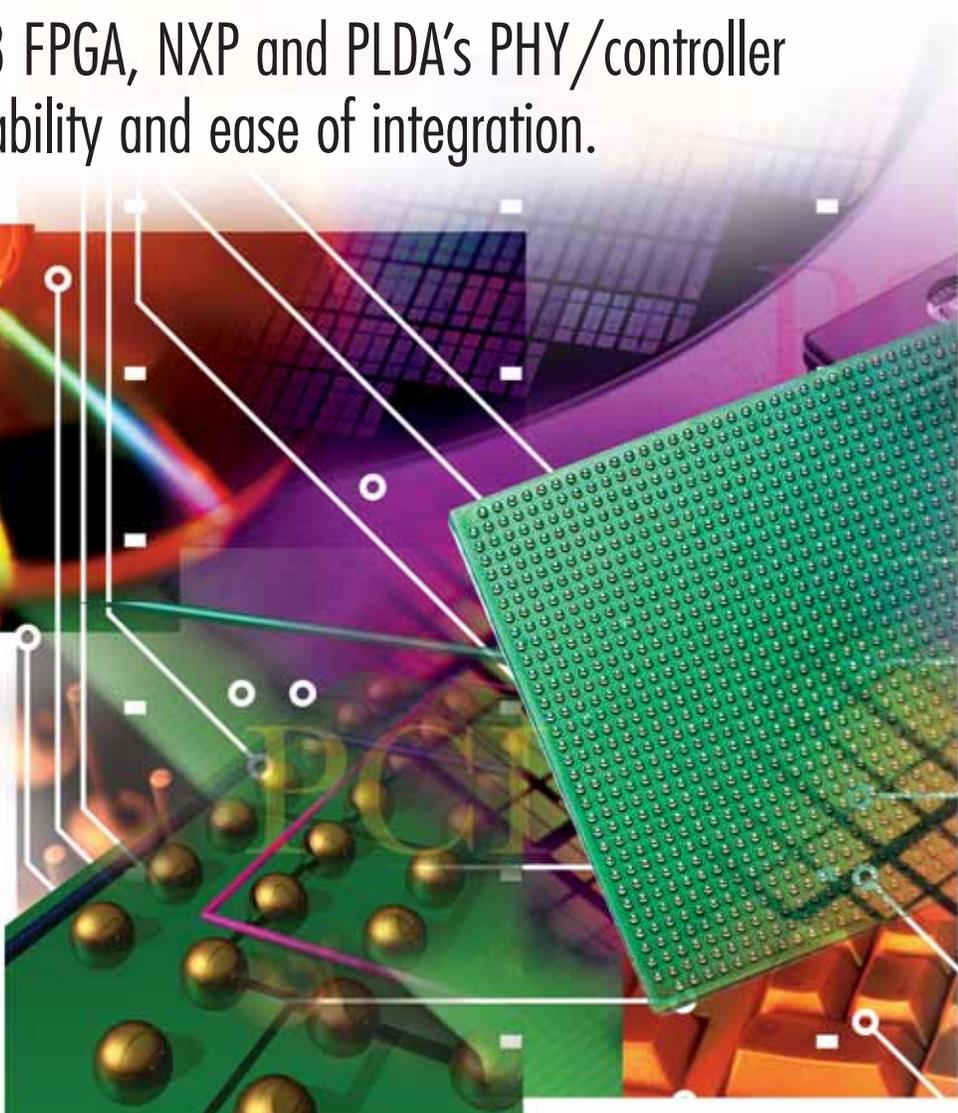
Based on the Spartan-3 FPGA, NXP and PLDA's PHY/controller solution delivers affordability and ease of integration.

by Ho Wai Wong-Lam  
Marketing Manager  
NXP Semiconductors  
[ho.wai.wong-lam@nxp.com](mailto:ho.wai.wong-lam@nxp.com)

Martin Gallezot  
Marketing and Sales Director  
PLDA  
[mgallezot@plda.com](mailto:mgallezot@plda.com)

Developers today are increasingly squeezed between providing ever-greater throughput for data-hungry applications and shrinking product lifecycles. Take frame grabber products, for example, which are image-processing computer boards that capture, process, and store image data for a wide range of scientific, medical, and industrial applications. Advances in image sensing continue to put a greater demand on the interconnect bandwidth as the number of pixels, frame rates, and number of cameras served by a single frame grabber increases. Some applications already require throughput exceeding 1 Gbps.

To meet the demands of today's development environment, companies are adopting PCI Express, the next-generation I/O interconnect technology of choice, and increasingly relying on FPGA-based interoperable, proven solutions that reduce the integration risk and decrease a product's time to market.



PCI Express is designed to replace PCI and PCI-X interconnect technologies, providing a scalable bandwidth from x1 at 250 Mbps to x32 at 8 Gbps. The v2.0 specification further increases bandwidth by doubling the one-lane data rate from 2.5 Gbps in the v1.1 specification to 5 Gbps. Besides increased throughput, PCI Express offers a smaller connector than PCI/PCI-X, which lowers costs and eases

PCB routing. And to ease the transition from PCI/PCI-X, PCI Express is backward-compatible with PCI with regards to software performance.

Interoperable solutions such as the NXP PHY/PLDA IP controller offer the advantages of PCI Express coupled with the security of a PCI-SIG-proven solution, allowing you to concentrate on application-specific functionality.

## NXP's PX1012A

NXP Semiconductors (previously Philips Semiconductors) offers the PX1012A single-lane 2.5-Gbps PCI Express PHY device, which serves as a companion chip to FPGAs or digital ASICs. The PX1012A is ideal for use with PLDA's XpressLite PCI Express IP core.

The PX1012A is optimized for use with low-cost FPGAs. It is available in a very small package, delivers superior transmit and receive performance, and is compliant to PCI Express specifications v1.0a and v1.1. The PX1012A is designed to serve PCI Express applications in all kinds of form factors, from space-constrained and low-power ExpressCard modules to desktop add-on cards to PXIe test equipment rack add-on modules. The NXP PCI Express PHY PX1012A has the following key features:

- Compliant to PCI Express base specification v1.0a and v1.1; the NXP PHY passed the first official U.S. PCI-SIG v1.1 electrical compliance tests in December 2006.
- The Xilinx® Spartan™-3 FPGA communicates with the PX1012A PHY using the source-synchronous 250-MHz PXPIPE standard, based on SSTL\_2 I/O. The use of source-synchronous clocking eases PCB layout by making the data transactions between the FPGA and PHY devices more robust than the original PIPE standard, which uses only one clock for both transmit and receive data.
- A small 9 x 9-mm BGA package with two signal rings (so that only one inner-ring signal needs to escape between the balls). In fact, the NXP PHY can be laid out with just two PCB signal layers, which have been proven in actual hardware designs. Example reference schematics are available on request.
- Low power dissipation in normal L0 mode (typically <300 mW including I/O). For further reduction in power dissipation, the removal of termination resistors on the PHY/FPGA interface and optimized PCB layout can further reduce PX1012A power dissipation to <150 mW.

- Both commercial temperature grade (0 to 70°C) and industrial grade (-40 to 85°C) are available; this is a unique industrial temperature grade device for PCI Express PHY, making NXP PHY suitable for various industrial applications.

## PLDA's XpressLite IP Core

PLDA's XpressLite PCI Express IP core is a complete PCIe digital controller optimized for Spartan-3 FPGAs. It includes all three layers of the PCIe specification (physical, data link, and transaction layer), plus an additional application layer called the EZ DMA interface.

PLDA's EZ DMA interface is suitable for designers who have little or no experience with the PCI Express protocol or for experienced designers looking for a robust yet simple PCI Express interface. The EZ DMA interface provides designers with a target path, which includes a simple address/data bus, and a master path, which comprises multiple DMA engines that handle the transfer of data to the host system memory. The EZ DMA interface can also be used together with the built-in PCI Express controller in Virtex™-5 LXT devices.

The XpressLite controller is an RTL-level IP core fully compliant with the PCI Express protocol. As required, the configuration space (located within the transaction layer) implements all configuration registers and associated functions. The configuration space also generates all messages (PME#, INT, error, power slot limit), MSI requests, and completion packets from configuration requests that flow in the direction of the root complex.

The XpressLite is fully configurable through a graphical wizard, making it simple to customize such parameters as maximum payload size, configuration space registers, buffer sizes, and number of DMA channels. The wizard generates a wrapper that instantiates the core top level and connects ports and assigns parameters according to specified options. Unused core features are not synthesized.

The XpressLite is PCIe specification 1.1-compliant. It has a small footprint and minimal memory utilization. Typical implementation requires about 8,000 LUTs and seven block RAMs in a Spartan-3 device. These figures include the EZ multi-DMA application layer configured with two DMA channels. PCI Express features supported by the XpressLite controller are summarized in Table 1.

Core Type	Legacy or Native Endpoint
Maximum Payload	Up to 2 KB
Backend Data Path	64 bit
Virtual Channels	One
BARs, Expansion ROM	User-defined, set by the XpressLite wizard
PCI ID	User-defined, set by the XpressLite wizard
Legacy Power Management	Minimal or full, set by the XpressLite wizard
Message Signaled Interrupt (MSI)	<ul style="list-style-type: none"> <li>• Message count: 1 to 32, set by the the XpressLite wizard</li> <li>• 64-bit address: Yes</li> <li>• Per-vector masking: No</li> </ul>
EZ Multi-DMA Interface	<ul style="list-style-type: none"> <li>• DMA channels: up to eight, set by the XpressLite wizard</li> <li>• Number of outstanding requests: one to eight simultaneous requests, set by the XpressLite wizard</li> <li>• Maximum DMA transfer size: up to 4 GB</li> </ul>

Table 1 – PCI Express features supported by the XpressLite controller

PLDA's core package also includes a complete test bench with an RTL-level PCI Express bus functional model (BFM), transactor, monitor, and checker.

**Fully Compliant and Interoperable Solution**

The NXP/PLDA joint solution successfully completed PCI Express compliance tests administered at the PCI-SIG Compliance Workshop #48 in December 2005 and later workshops in 2006 (see transmitter eye pattern results in Figure 1).

NXP, PLDA, and mutual customers have run extensive and successful system tests in a multitude of PC systems, including:

- ASUS A8NE (x16, two x1 slots and one x4 slot)
- ASUS P5GP motherboard (Intel 915G chipset)
- ASUS P5LD2 DELUXE (Intel i945 chipset)
- Dell Dimension 4700 x1 and x16 slots (Intel 915G chipset)
- Dell Dimension 8400 (Intel 925G chipset)
- DELL Precision 370
- Dell Precision 470 x8 and x16 slots (Intel Turnwater E7525 chipset)
- HP XW4200
- HP XW4300
- MSI (Intel i915 chipset)
- Serverworks GC-SL
- Shuttle ATI Express 200 chipset
- Supermicro X6DA8G (x16 and x4 slots)

In addition to the standard PCI-SIG compliance tests, NXP and PLDA also used a variety of system tests developed internally and from third-party test equipment vendors:

- PCI scan diagnostic utility
- Transmitter electrical compliance tests v1.0a and v1.1
- PCI-SIG PCI-ECV v1.2

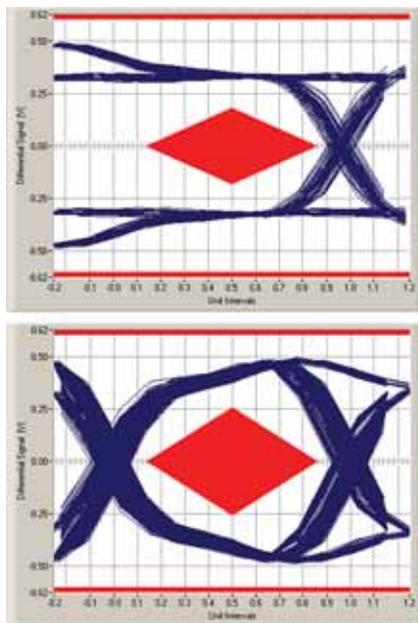


Figure 1 – PLDA XpressLite SP3 board transmitter eye pattern with v1.1 PCI Express template (top: non-transition eye; bottom: transition eye)

- Agilent protocol test card (PTC) test
- Agilent receiver bit error rate (BER) test
- NXP receiver performance test
- PLDA throughput measurement test

**High Throughput**

Theoretical maximum throughput is a function of payload size and PCI Express protocol overhead. Actual throughput results depend on factors such as software driver efficiency, PCI Express IP core efficiency, the user's application design, transmitter jitter, and receiver BER performance. Throughput might be further compromised by link-layer protocol overhead such as ACK/NAK packets (acknowledge/non-acknowledge), re-transmitted packets, and flow control protocol (credit reporting).

**Theoretical Maximum Throughput**

The line speed of one-lane PCI Express is 2.5 Gbps. However, 8B/10B encoding overhead reduces the maximum PXPIPE data throughput (2.5 Gbps divided by 10 bits per byte = 250 Mbps). Given the overhead performance cost, throughput typically increases with payload size – up to a point. For example, for a payload of 128 bytes, the theoretical efficiency is 86% (128 B(ytes) payload + 12B header + 8B framing) and the maximum theoretical throughput is 216 Mbps. Although the PCI Express specification specifies a potential maximum payload size of as much as 4 KB, most existing applications only implement a maximum payload size of 128B or 256B. Throughput measurements for the NXP/PLDA solution are listed in Table 2.

**Receiver Performance**

NXP and PLDA have performed extensive proprietary system tests in many PC systems to ensure that there are no recoverable receiver errors for extended periods of time (many hours). Test results yield a BER of 1x10<sup>-12</sup>.

PCI Express specifications v1.0a and v1.1 require 0.6 UI (unit interval) of receiver jitter tolerance, but the specifications do not precisely indicate how the jitter components are composed. NXP performed receiver BER tests using an Agilent BER tester.

- Agilent J-BERT N4903A
- TJ = total jitter; RJ = random jitter; PJ = periodic jitter; DDJ = data-dependent jitter; UI = unit interval; ISI = inter-symbol interference; BER
- 0.60UI TJ = 0.25UI RJ + 0.25UI PJ (at 15 MHz) + 0.1UI DDJ

PCIe PHY	IP Core	Computer Platform	Payload	Throughput	Throughput
				DMA Read (Card->PC)	DMA Write (PC->Card)
PX1012A	PLDA	ASUS A8NE Supermicro X6DA8G	128 bytes	200 Mbps	175 Mbps

Table 2 – Throughput measurement results



Figure 2 – The PLDA XpressLite SP3 is based on a Xilinx Spartan-3 FPGA and includes NXP's PCIe PX1012A PHY and PLDA's XpressLite IP controller.

- 0.1 UI DDJ = ISI module, which stimulates 9 inches of PCB trace, equivalent to about 0.1-UI jitter and significant amplitude degradation.

NXP obtained excellent receiver performance results with the Agilent bit error tester.

- PX1012A achieved  $< 1 \times 10^{-12}$  BER with a 800-mV<sub>diff. p-p</sub> input signal, which is the minimum transmit output level allowed by the PCI Express specification. The 800-mV<sub>diff. p-p</sub> signal goes from the pattern generator to the BERT ISI module before feeding into the PHY receiver.
- Without the ISI module, the PX1012A can achieve  $< 1 \times 10^{-12}$  BER with a 400-mV<sub>diff. p-p</sub> input signal.

#### PLDA XpressLite SP3 Design Kit

The PLDA XpressLite SP3 Design Kit (see Figure 2) is based on a Spartan-3 FPGA (XC3S2000 device) and integrates the NXP PX1012A. The design kit includes a Protocore (board-only) license of the PLDA XpressLite IP core.

The Protocore license is a full-featured RTL-level license that is valid for an unlimited duration when used with the design kit. It also includes a software development kit, with a library of C source code func-

tions and a driver. With the Protocore license, you can freely simulate and synthesize your own design connected to the XpressLite IP core and re-program the FPGA device accordingly.

The complete design has gone through extensive validation testing, is hardware-proven, and readily available for customer prototyping.

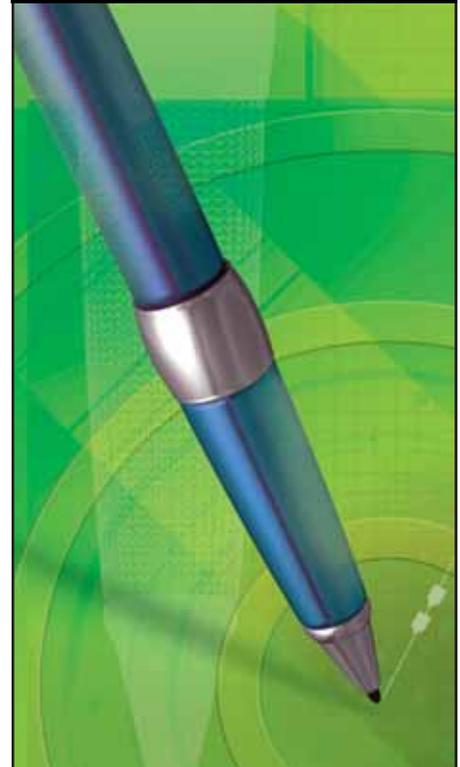
The XpressLite SP3 design kit provides a low-cost solution to prototype your design with PCI Express. It can also be used as part of your final product if you want to save the effort of designing your own PCIe board. A 400-hole matrix (2.54-mm step) for prototyping and probing purposes is also provided.

#### Conclusion

A prototyping board like PLDA's XpressLite SP3 Design Kit, which combines the power of the Xilinx Spartan-3 FPGA with a proven PCI Express NXP PHY/PLDA IP controller solution, responds to today's development challenges to produce applications that support ever greater data throughput in an ever-shrinking product lifecycle. For more information, visit [www.standardics.nxp.com/products/pci/phys/](http://www.standardics.nxp.com/products/pci/phys/) and [www.plda.com/products/board\\_pcie\\_sp3.php](http://www.plda.com/products/board_pcie_sp3.php). ●●

NXP also provides a joint solution with the Xilinx PCI Express PIPE core for Spartan-3 FPGAs. You can buy this solution from Xilinx, which includes an x1 PCI Express add-in card and an evaluation version of the PIPE core.

## GET PUBLISHED



### WOULD YOU LIKE TO BE PUBLISHED IN XCELL PUBLICATIONS?

It's easier than you think!

Submit an article draft for our Web-based or printed Xcell Publications and we will assign an editor and a graphic artist to work with you to make your work look as good as possible.

For more information on this exciting and highly rewarding program, please contact:

Forrest Couch  
 Publisher, Xcell Publications  
[xcell@xilinx.com](mailto:xcell@xilinx.com)



# Great Test, Less Filling

SystemBIST™ enables FPGA Configuration that is less filling for your PCB area, less filling for your BOM budget and less filling for your prototype schedule. All the things you want less of and more of the things you do want for your PCB – like embedded JTAG tests and CPLD reconfiguration.

Typical FPGA configuration devices blindly “throw bits” at your FPGAs at power-up. SystemBIST is different – so different it has three US patents granted and more pending. SystemBIST’s associated software tools enable you to develop a complex power-up FPGA strategy and validate it. Using an interactive GUI, you determine what SystemBIST does in the event of a failure, what to program into the FPGA when that daughterboard is missing, or which FPGA bitstreams should be locked from further updates. You can easily add PCB 1149.1/JTAG tests to lower your down-stream production costs and enable in-the-field self-test. Some capabilities:

- User defined FPGA configuration/CPLD re-configuration
- Run Anytime-Anywhere embedded JTAG tests
- Add new FPGA designs to your products in the field
- “Failsafe” configuration – in the field FPGA updates without risk
- Small memory footprint offers lowest cost per bit FPGA configuration
- Smaller PCB real-estate, lower parts cost compared to other methods
- Industry proven software tools enable you to get-it-right before you embed
- FLASH memory locking and fast re-programming
- New: At-speed DDR and RocketIO™ MGT tests for V4/V2

If your design team is using PROMS, CPLD & FLASH or CPU and in-house software to configure FPGAs please visit our website at <http://www.intellitech.com/xcell.asp> to learn more.



# Create Memory Interface Designs Faster with Xilinx Solutions

Xilinx simplifies memory interface design with hardware-verified reference designs, easy-to-use software tools, and complete development kits.

by Adrian Cosoroaba  
Marketing Manager  
Xilinx, Inc.  
adrian.cosoroaba@xilinx.com

Xilinx® FPGAs provide I/O blocks and logic resources that make interface design easier. Nonetheless, designers must still configure, verify, implement, and properly connect these I/O blocks – along with extra logic – to the rest of the system in the source RTL code, carefully simulated and verified in hardware.

In this article, I'll review the performance requirements, design challenges, and Xilinx solutions for memory interface design, from low-cost implementations with Spartan™-3 Generation FPGAs to the highest bandwidth interfaces using Virtex™-5 FPGAs.

## Performance Requirements and Xilinx Solutions

In the late 1990s, memory interfaces evolved from single-data-rate SDRAMs to double-data-rate (DDR) SDRAMs, with today's DDR2 SDRAMs running at 667 Mbps per pin or higher.

Applications can generally be classified in two categories:

- Low-cost applications, where the cost of the device is most important
- High-performance applications, where getting the highest bandwidth is paramount

DDR SDRAMs and low-end DDR2 SDRAMs running below 400 Mbps per pin are adequate to meet the memory bandwidth requirements of most low-cost systems. For these applications, Xilinx offers Spartan-3 Generation FPGAs: Spartan-3, 3E, 3A, and 3AN devices.

For applications that push the limits of memory interface bandwidths, like 667-Mbps-per-pin DDR2 SDRAMs, Xilinx offers Virtex-5 FPGAs.

Bandwidth is a factor related to both the data rate per pin and the width of the data bus. Both Spartan-3 Generation and Virtex-5 FPGAs offer distinct options, spanning a range from smaller low-cost systems with data bus widths of less than 72 bits to those as wide as 576 bits for the larger Virtex-5 packages (Figure 1).

Wider buses at these speeds make chip-to-chip interfaces all the more challenging, requiring larger packages and better power-to-signal and ground-to-signal ratios. Virtex-5 FPGAs have been built with advanced SparseChevron packaging that provides superior signal-to-power and ground-pin ratios. Every I/O pin is surrounded by sufficient power and ground pins and planes to ensure proper shielding for minimum crosstalk noise caused by simultaneously switching outputs (SSO).

## Memory Interfaces with Spartan-3 FPGAs

For low-cost applications where 400-Mbps bit rates per pin are sufficient, Spartan-3 Generation FPGAs coupled with Xilinx software tools provide an easy-to-implement, economical solution.

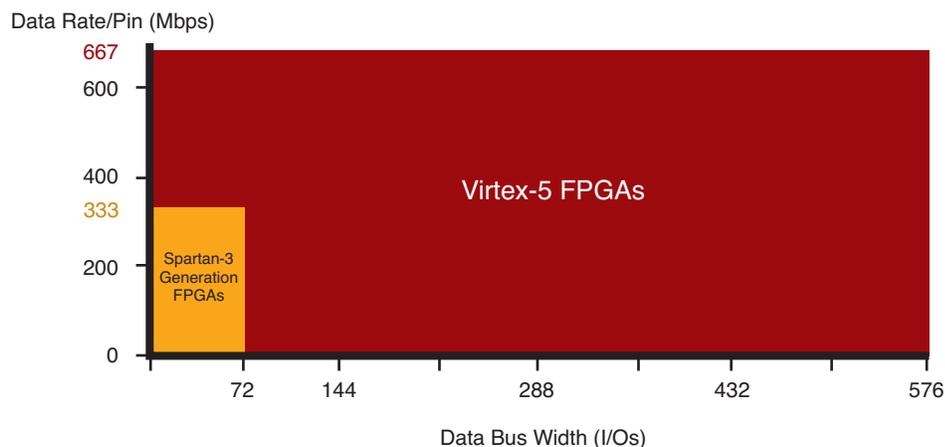


Figure 1 – Xilinx FPGAs and memory interface bandwidth

Three fundamental building blocks comprise a memory interface and controller for an FPGA-based design: the read and write data interface, the memory controller state machine, and the user interface that bridges the memory interface design to the rest of the FPGA design. Implemented in the fabric, these blocks are clocked by the output of the digital control manager (DCM) that, in the Spartan-3 Generation implementation, also drives the look-up table (LUT) delay calibration monitor (a block of logic that ensures proper timing for read data capture).

In the Spartan-3 Generation implementation, read data capture is implemented using the LUTs in the configurable logic blocks (CLBs). During a read transaction,

puts to transmit the DQS strobe properly aligned to the command and data bits.

The implementation of a DDR2 SDRAM memory interface has been fully verified in hardware. The design was implemented in the Spartan-3A starter kit board using a 16-bit-wide DDR2 SDRAM memory device and the XC3S700A-FG484 device. The reference design uses only a small portion of the Spartan-3A FPGA's available resources: 13% of IOBs, 9% of logic slices, 16% of global buffer (BUFG) multiplexers (MUXs), and one of the eight DCMs.

You can easily customize Spartan-3 Generation memory interface designs to fit your application using the Memory Interface Generator (MIG) software tool.

This trend is readily apparent when comparing the data valid windows of DDR SDRAMs running at 400 Mbps and DDR2 memory technology, which runs at 667 Mbps. The DDR device with a 2.5-ns data period has a data valid window of 0.7 ns, while the DDR2 device with a 1.5-ns period has a mere 0.14 ns.

Virtex-5 FPGAs address this challenge with dedicated delay and clocking resources in the I/O blocks – called ChipSync™ technology. The ChipSync block built into every I/O contains a string of delay elements (also known as tap delays), called IODELAY, with a resolution of 75 ps.

The architecture of the implementation is based on several building blocks. The user interface that bridges the memory controller and physical layer interface to the rest of the FPGA design uses a FIFO architecture. The FIFOs hold the command, address, write data, and read data. The main controller block controls the read, write, and refresh operations. Two other logic blocks execute the clock-to-data centering for read operations: the initialization controller and the calibration logic (Figure 2).

The physical layer interface for address, control, and data is implemented in the IOBs. The DQ capture uses the memory strobe to capture corresponding DQ and register it with a delayed version of the strobe. This data is then synchronized to the system clock domain in a second stage of flip-flops. The input serializer/deserializer feature in the I/O block is used for read capture – the first pair of flip-flops transfer the data from the delayed strobe to the system clock domain. The technique involves the use of 75-ps tap delay (IODELAY) elements that are varied during a calibration routine implemented by the calibration logic. This routine is performed during system initialization to set the optimal phase between strobe, data, and the system clock to maximize timing margins.

There are other aspects of the design, such as the overall controller state machine logic generation and user interface. To make the complete design easier for the FPGA designer, Xilinx developed the Memory Interface Generator.

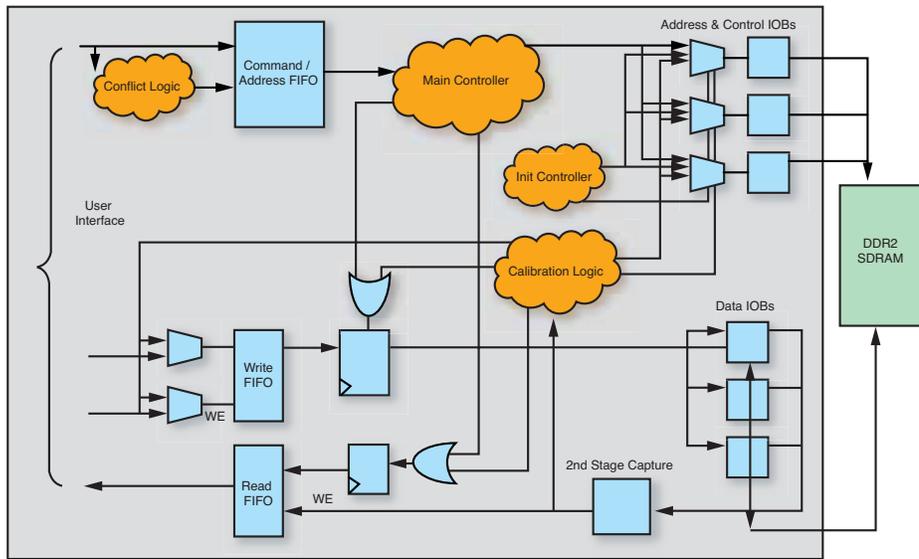


Figure 2 – Virtex-5 FPGA memory interface architecture

the DDR2 SDRAM device sends the read data strobe (DQS) and associated data to the FPGA edge-aligned with the read data (DQ). Capturing the DQ is a challenging task in source-synchronous interfaces because data changes at every edge of the non free-running DQS strobe. The read data capture implementation uses a tap-delay mechanism based on LUTs.

Write data commands and timings are generated and controlled through the write data interface. The write data interface uses input/output block (IOB) flip-flops and the DCM's 90-, 180-, and 270-degree out-

### Memory Interfaces with Virtex-5 FPGAs

With higher data rates, interface timing requirements become more challenging. The trend toward higher data rates presents a serious problem to designers in that the data valid window – that period within the data period during which DQ can be reliably obtained – is shrinking faster than the data period itself. This is because the various uncertainties associated with system and device performance parameters, which impinge on the size of the data valid window, do not scale down at the same rate as the data period.

### Design and Integration with MIG

Integrating all of the building blocks, including the memory controller state machine, is essential for the completeness of your designs. Controller state machines vary with the memory architecture and system parameters. State machine code can

memory module (DIMM). The same GUI provides a selection of bus widths and clock frequencies. Other options provide control of the clocking method, CAS latency, burst length, and pin assignments.

The MIG tool can generate in a matter of minutes the RTL and UCF files, which are the HDL code and constraints files, respectively. These files are generated using a library of hardware-verified reference designs, with modifications based on user inputs.

You have complete flexibility to further modify the RTL code. Unlike other solutions that

example. After the optional code change, you can perform additional simulations to verify the functionality of the overall design.

The MIG also generates a synthesizable test bench with memory checker capability. The test bench is a design example used in the functional simulation and the hardware verification of the Xilinx base design.

The final stage of the design is to import the MIG files in the ISE project, merge them with rest of your FPGA design files, conduct synthesis and place and route, run additional timing simulations if needed, and finally perform verification in hardware. MIG software also generates a batch file with the appropriate synthesis, map, and place and route options to help you optimally generate the final bit file.

### Development Boards and Kits

Hardware verification of reference designs is an important final step to ensure a robust and reliable solution. Xilinx has verified the memory interface designs for both Spartan-3 Generation and Virtex-5 FPGAs. Table 1 shows the memory interfaces supported for each of the development boards.

The development boards range from low-cost Spartan-3 Generation FPGA implementations to the high-performance solutions offered by the Virtex-5 FPGA family.

### Conclusion

You can complete your memory interface designs faster with a wide spectrum of Xilinx FPGA devices, dedicated software tools like the MIG, and development boards that fit your application needs, from low cost to high bandwidth.

For more information and details on these memory interface solutions, visit [www.xilinx.com/memory](http://www.xilinx.com/memory).

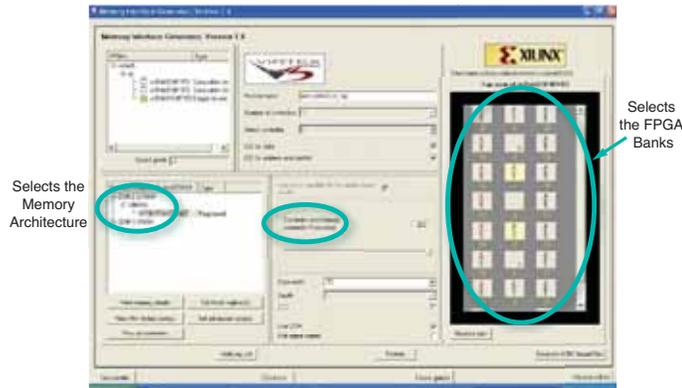


Figure 3 – MIG GUI

Xilinx FPGA	Spartan-3E	Spartan-3A	Spartan-3AN	Virtex-5
Development Board/Kit	Starter Kit	Development Kit	Starter Kit	ML-561
Memory Interfaces Supported	DDR	DDR2	DDR2	DDR DDR2 QDR-II RLDRAM II

Table 1 – Development boards and kits for memory interfaces

also be complicated and a function of many variables, such as architecture, data bus width, depth, access algorithms, and data-to-strobe ratios.

The complete design can be generated with the MIG, a software tool freely available from Xilinx as part of the ISE™ software CORE Generator™ suite of reference designs and IP. The MIG design flow is very similar to the traditional FPGA design flow. The benefit of the MIG for designers is that there is no need to generate RTL code from scratch for the physical layer interface or the memory controller.

You can use the MIG's GUI to set system and memory parameters (Figure 3). For example, after selecting the FPGA device, package, and speed grade, you can select the memory architecture and pick the actual memory device or dual in-line

offer “black-box” implementations, the code is not encrypted, providing complete flexibility to change and further customize a design. The output files are categorized in modules that apply to different building blocks of the design: user interface, physical layer, or controller state machine. You may wish to customize the state machine that controls the bank access algorithm, for

### TAKE THE NEXT STEP

- Download the Memory Interface Generator to generate your reference designs for Virtex-5, Virtex-4, and Spartan-3 devices, including HDL code and pin placements
- See a demo on Memory Interface Solutions with Xilinx FPGAs
- View an on-demand webcast on Low-Cost to High-Performance Memory Interface Design Made Easy with Xilinx FPGAs
- Contact Xilinx

# Driving Home Multimedia

Infotainment hits the road with affordable in-car networking.



by Carl Rohrer  
Senior Design Engineer  
Xilinx, Inc.  
[carl.rohrer@xilinx.com](mailto:carl.rohrer@xilinx.com)

Per square foot, you probably have more multimedia in your car than you do in your house. There are the two LCD screens for the kids in the backseat, controlled by a DVD player or a video game console. An audio system is powered by the latest MP3 player. There's the navigation system and even broadcast television in some luxury vehicles. Plus, you've probably got more speakers in your car than in a high-end surround sound system. It's no wonder there are so many distracted drivers on the road. What you need is a single, simple control interface; what manufacturers need is a sophisticated network.

MOST (Media Oriented Systems Transport) is a network standard on the rise among automakers and automotive suppliers. It provides one single interface for managing all of your multimedia devices. Its strength resides in the ability to handle multiple streams of data intended for different targets without losing congruity. On-time data: this is something even your home network cannot guarantee.

In this article, I'll explore the MOST network and demonstrate the flexibility of the Xilinx® MOST solution.

## Under the Hood

The MOST network runs on optical fiber, typically in a ring topology. The clock and serialized data are bi-phase encoded, requiring just one single piece of fiber to be routed. MOST offers up to 25 Mbps of

aggregate bandwidth, far surpassing classical automobile networks. In other words, you could play 15 distinct audio streams at the same time.

Each multimedia device is represented by a node in the ring. A typical MOST network will have between three and 10 nodes. There is one timing master that drives the system clock and generates frames, or 64-byte sequences, of data. The remaining nodes all act as slaves. One node acts as the user control interface, or MMI (man machine interface). Often, this is also the timing master. Figure 1 illustrates a basic MOST ring.

The main payload comprises 60 bytes within the 64-byte frame. This payload is made up of synchronous and asynchronous fields. The synchronous field is used for streaming contiguous data; audio and video fall under this category. The asynchronous field is used for sporadic data transfers in applications such as Internet access, navigation data transfer, and phone book synchronization. Also, this channel could be used for firmware upgrades of the control units.

A node may send or receive data during its assigned time slots. A time slot – one synchronous byte within the payload – is dynamically allocated between a requesting node and the timing master. Typically, one node will transmit data onto a time slot, while any number of other nodes will collect the data from that time slot.

The boundary between synchronous and asynchronous is dynamically controlled by the timing master. In any given frame, the synchronous field may be from 4 to 60 bytes, leaving the remainder of the 60 bytes to the asynchronous field.

The remaining four bytes of the frame are used for header, trailer, and control information. The header contains a preamble for frame alignment. The trailer, among other things, does a parity check. The control field is used for network-related messaging. These messages may be low level, like the allocation and de-allocation of time slots. Conversely, they can be high-level application messages sent from the operator, such as play next track, volume control, or repeat play.

### Getting More Out of MOST

Instead of having an external MOST controller chip connected to a microcontroller or DSP, you can integrate all of your components into one single FPGA. Fewer external components and reduced PCB space translates into cost savings for developers.

Xilinx offers a fully parameterizable MOST Network Interface Controller (NIC) IP core. You can customize the core to be a timing master or reduce logic with a slave-only configuration. The core is controlled by a full suite of registers accessible through an on-chip peripheral (OPB) interface. The OPB interface works seamlessly with the Xilinx MicroBlaze™ 32-bit RISC processor core included in Xilinx Platform Studio.

A full set of low-level driver files are already available in C source code. The drivers provide a series of functions for accessing the register space, handling interrupts, and streaming data to the core. Mocean Laboratories AB provides MOST network services for the IP core for a complete network stack, leaving you only to write your desired application.

Unique to the Xilinx MOST NIC is a streaming port interface that allows data to be preprocessed in real time. It is ideal for bolting on data filters or encryption/decryption modules. This LocalLink interface, a Xilinx standard, will significantly reduce traffic on the processor and processor bus by offloading dedicated procedures. It's versatile too. You can read or write data in either the receive or transmit directions. Best of all, if you don't use it, the Xilinx implementation tools will remove the unnecessary logic, allowing you to conserve resources and fit your design into a smaller device.

Synchronous data can be transmitted and received on either the streaming port or the OPB interface. Regardless of the method you choose and how many time slots you have assigned, the core formats data into 32-bit words for these user interfaces. Through register definitions, the core accumulates received time slot data that is deposited through one of 16 logical channels. The reverse is true in the transmit direction. The use of these logical channels allows for 16 distinct streams of data in each direction.

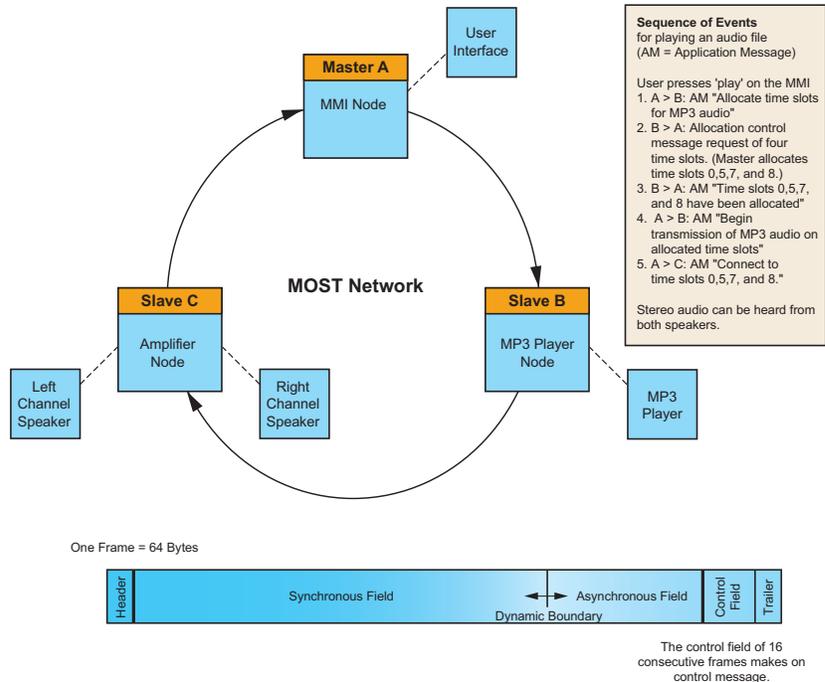


Figure 1 – An example MOST automotive ring

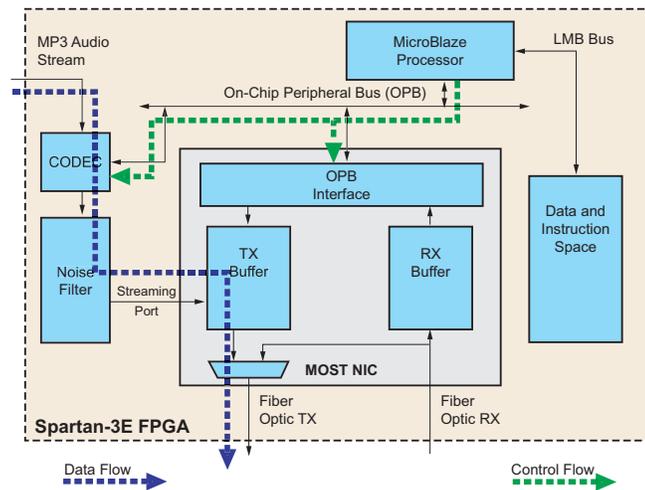


Figure 2 – Theoretical MP3 node

The Xilinx MOST NIC core is flexible. Consider the MOST ring in Figure 1 again, which illustrates how you might design each node with the Xilinx MOST NIC. You could configure the core as a timing master to operate as the MMI. As the timing master, the core will send and receive control messages that regulate ring operation. This node will also send application messages, also through the control field, on behalf of the user. You can also use the driver files and Mocean's

network services on top of MicroBlaze for event scheduling.

You can turn the MP3 player into a high-end audio feed by adding a noise filter bolt-on to eliminate the artifacts of audio compression. The payload data can run from the codec through the noise filter directly into the streaming port, completely avoiding the OPB bus. Once again, you can use the MicroBlaze embedded processor for interrupt handling and event scheduling. Figure 2 depicts the block diagram of such a design.

For the amplifier, imagine a minimal design that simply receives data and forwards it to the speakers. Instead of using an embedded processor, as in the MP3 node, you can implement a smaller user design capable of full network negotiation and data collection. Such a compact design may be placed in a smaller device for further cost savings.

**Conclusion**

If you drive a high-end European automobile, you may already have a MOST network. It has gained acceptance as one of the de facto standards for automotive networking among European-based OEMs. Those of us driving more affordable cars will not have too long to wait, however. With the advent of competition, this once private standard is

becoming more affordable for cost-conscious automakers.

As demand grows for larger amounts of data – from audio to video, telematics, and navigation-based applications – the MOST network technology has plans to grow as well. The next-generation standard, MOST 50, has already been defined and offers twice as much bandwidth as the original. At the time of this writing, the MOST Cooperative is planning a third-generation network expected to reach data rates of 150

Mbps and beyond. These updates will eventually increase available application bandwidth by more than an order of magnitude and are expected to support both a copper and optical physical medium.

The Xilinx MOST NIC is available today in CORE Generator™ software. At six block RAMs and about 2,600 slices, it will fit in a mid-sized Spartan™-3E device with room to spare for an embedded processor, peripherals, buffers, and your own custom circuitry. ●●

**TAKE THE NEXT STEP** (Digital Edition: [www.xcellpublications.com/subscribe/](http://www.xcellpublications.com/subscribe/))

- Read the MOST NIC data sheet for more details
- Contact your Xilinx FAE to test-drive the MOST NIC core

**Stay Ahead!**

**Xilinx® Virtex™-5 LX FPGA**  
A new generation of performance

Analog I/O, Camera Link, LVDS, FPDP-II, RS485/422 & L-Band Receiver options  
Fast, integrated I/O without bottlenecks

Multiple banks of fast memory  
DSP & I/O optimized memory architecture

PCI-X Interface with multiple DMA controllers  
More than 1GB/s bandwidth to host

Libraries and Example Code  
Easy to use with head-start time-to-market

with the PMC-FPGA05 Range of Virtex-5 based PMCs

Processing & FPGA    Input/Output    Data Recorders & Storage    Bus Analyzers

For more information, please visit <http://virtex5.vmetro.com> or call (281) 584-0728

**VIMETRO**  
Innovation deployed

# Making the Most of MOST Control Messaging

This case study presents the design of Xilinx LogiCORE MOST NIC control message processing.

by Stacey Secatch  
Staff Design Engineer  
Xilinx, Inc.  
[stacey.secatch@xilinx.com](mailto:stacey.secatch@xilinx.com)

At only 2,600 slices, the Xilinx® MOST (Media Oriented Systems Transport) network interface controller (NIC) occupies roughly half of a Spartan™-3E 500 device, which leaves plenty of room for the microprocessor and other peripherals. This full-featured embedded core provides easy access to data and network control as a slave or master node.

In this article, I'll present a control message processing design case study and show how the MOST NIC is able to autonomously handle messages without any host software intervention. With careful planning, the minimal-resource PicoBlaze™ configurable state machine can handle control messages correctly and efficiently.

## Control Messages on a MOST Network

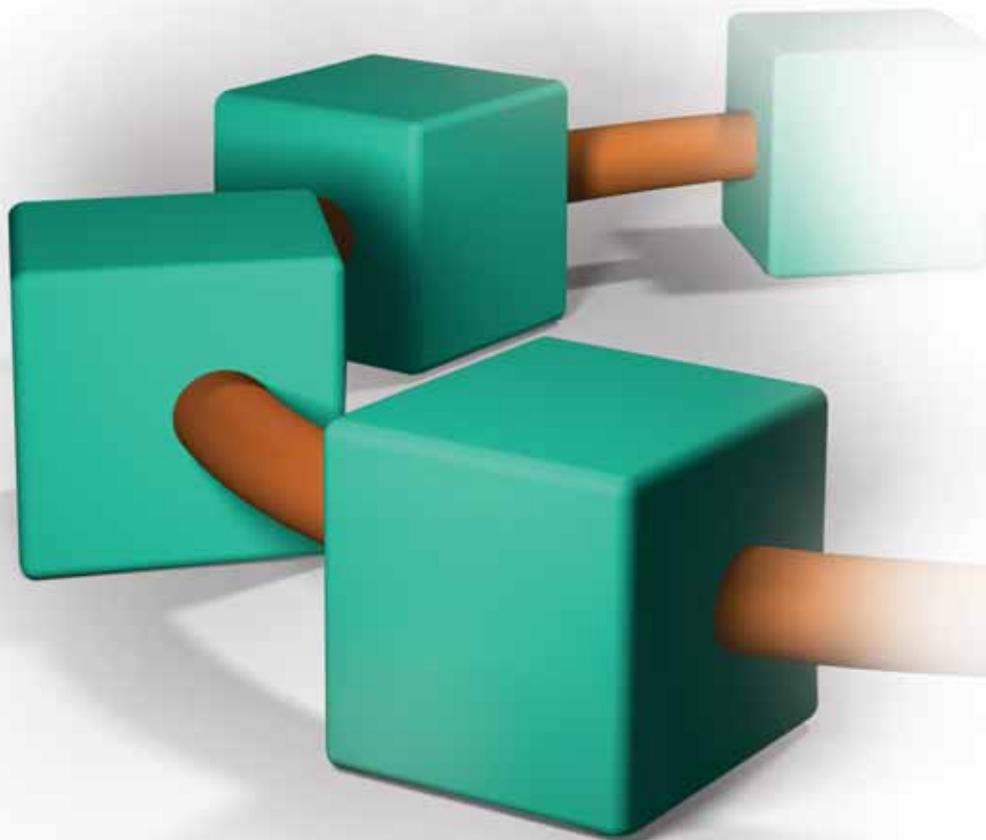
MOST is an automotive infotainment-oriented network standard with a ring topology. One of its strengths is that it supports multiple simultaneous streams of media intended for several nodes in the ring. However, like any network, more than just data is sent between nodes; control communication is also required. A simple MOST ring network is shown in Figure 1, with one network master and several slaves. Six different control message types are available. Let's focus on the three that are most commonly used during operation of a MOST network.

### Normal Messages

Normal messages are distinct from other types in that they are intended for high-level application use in the host processor. All other messages are used for ring operation and debugging. For example, as part of the boot sequence of the ring, the master could send a normal message to determine the identification of slave B. Normal messages may be combined in the application layer when there is more information than one message can hold.

### Channel Allocation and Deallocation

Allocation messages are sent to the master node to gain access to the portion of the frame set aside for media data. The requestor indicates how many byte-wide channels are desired and the responder (master) confirms both the status of the request (granted or denied) and which channels have been granted. Deallocation messages are similar to allocations, where the requestor indicates which channels to deallocate and the master responds with deallocation granted.



**Choosing the Hardware/Software Partition**

Embedded design involves choosing the best boundary for the hardware/software interface in terms of functionality and resources. Choosing to implement too much in hardware is costly because of the additional resources required to implement the design. Choosing to implement too much in software can result in an overburdened microprocessor and stalled logic, possibly even resulting in a non-functional design if the processor drops events.

For the MOST LogiCORE™ solution, we found a clear boundary for dividing up control message processing.

**Delay Time Requirements**

Control messages are broken across multiple frames. As messages travel around the ring, specific handshaking is required between the communicating nodes, with a turnaround time of roughly one of those frames (about 23 μs). Message-response generation usually requires calculations. For example, allocation responses depend on determining whether enough free channels are available.

Because this service requirement is so tight, it makes sense to place control message handling within the core as hardware, leaving the external processor available to run the network stack and media-oriented user application.

**Items to Leave to Software**

Although nearly all control message types can be handled completely by the LogiCORE system, normal control messages must be passed on to the application. Therefore, the MOST NIC LogiCORE solution provides interrupts to signal that a normal message has been received and is available for processing. And because all messages are initiated by the application, there are configuration registers for the application to indicate that a message is available for sending and interrupts to alert the application that the message sequence is complete.

**Selecting the Implementation**

The next step after deciding to process control messages in hardware was to choose the most efficient design possible. We could

have decided to implement this in traditional fashion as a custom state machine. Preliminary estimates placed the size of the state machine at more than 1,000 slices, plus a block RAM to contain all of the message buffers. Note that control messages only use 2 bytes out of a 64-byte frame, yet processing just these two bytes would have accounted for more than a third of the core.

Instead, we chose to incorporate the PicoBlaze configurable state machine as the backbone for control channel processing and minimize core resources. In our implementation, this includes two block RAMs to hold the instruction memory, approximately 100 slices for the PicoBlaze component, and 150 slices for peripherals, as well as the original block RAM for message buffering.

Choosing a PicoBlaze controller offers you similar opportunities for resource minimization. You will probably also see the same benefits that we did, namely a shortened design cycle and easier code maintenance thanks to the simplicity of coding message processing procedurally.

**Organizing Data Storage**

Determining what kind of memory to use for variables when designing an RTL state machine is rather straightforward. In general, you use RAM for large buffers and stick everything else in a register. When using a PicoBlaze controller, you need to do a bit more up-front planning to minimize resources while still providing access to all of your data.

**State Machine Variables**

The 64 “free” scratch pad registers are ideal for storing variables that you will never use outside of the PicoBlaze controller. For example, the accumulators to store intermediate checksum results for MOST control messages are only needed by the state machine.

**Larger Buffers**

Block RAM is an ideal location for buffering. Although the PicoBlaze controller only provides direct access to 256 locations, it is probably more efficient for you to use one block RAM than to instantiate distributed

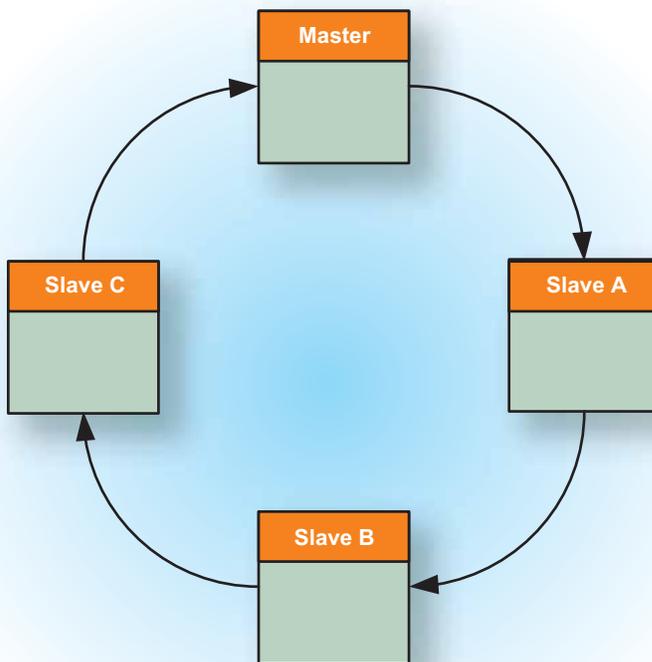


Figure 1 – A sample MOST ring

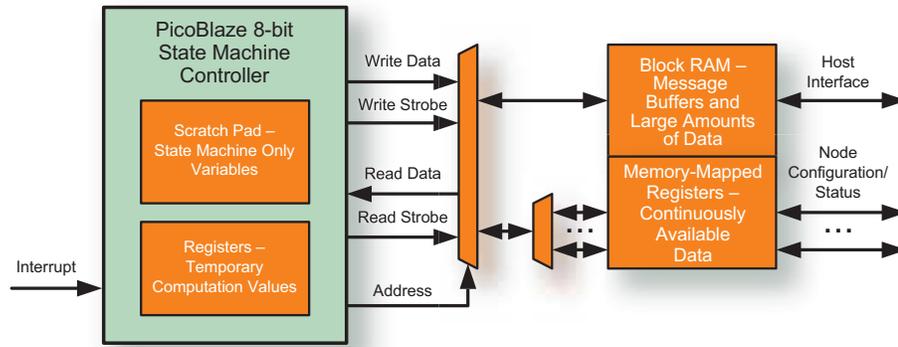


Figure 2 – Memory organization for Xilinx MOST NIC control message processing

memory for all of your buffers. And because a PicoBlaze configurable state machine executes sequentially, you will never need to access more than one buffer at a time, which makes grouping together all of your memory easy. As a bonus, Xilinx block RAM automatically converts data widths, seamlessly allowing 32-bit access from the host side and 8-bit access from the PicoBlaze controller side.

**Shared Variables**

Some configuration and status variables will need to be continually available. You can map these variables into registers in the PicoBlaze controller external memory space for your state machine to access, just like the memory in the MOST LogiCORE control processing module (Figure 2). In the LogiCORE system, the PicoBlaze controller external memory port is mapped to a shared block RAM, with the exception that the end of the memory range is memory-mapped registers.

As an example, the currently programmed addresses of the MOST node are mapped into a read-only “register” (a mux input) to determine if a received message is meant for this node. Status information about the current occupancy of the buffers is placed into a write-only register for access by the external host.

**Ordering Your Processing**

Complex state machines are generally built from smaller interlocking state machines that operate in parallel. PicoBlaze configurable state machines execute sequentially, providing advantages in terms of conflict

avoidance and race conditions. You will need to plan carefully, however, to ensure that your event scheduling is correct and that that no queued events get dropped. Here are some guidelines that might help in designing your PicoBlaze application:

- Prioritize your event processing. If you have events that are not important, choose to execute those last. For example, the value of buffer occupancy counts is changed by a MOST host buffer read. However, this update may be deferred for several frames. Because responding to a message must complete in one frame, the core services message responds first.
- Preserve event ordering. If your processing requires that one event complete before another, you will want to execute the code for the first event, even if the second event may have a tighter timing requirement. For example, control bytes must be sent out before the bytes in the current frame

are even received. Therefore, for a given frame, transmit events are always processed before receive events.

- Consider using a busy-wait loop. If you have a very long event that cannot be processed in the time frame required by other events, you might consider placing the long event in the main processing wait loop.

For example, preparing a MOST control message for transmit can take multiple frames to complete. We could have set up multiple levels of interrupt servicing to ensure that we could still process frame data, but this type of task fits very well into a busy-wait loop, which automatically gives it the lowest priority.

If you do choose to process data in the main loop, some additional care is needed to ensure that you do not corrupt your main loop when servicing interrupts. At the start of interrupt servicing, you should copy all registers that might get overwritten, and restore those registers as interrupt servicing completes. There are also PicoBlaze commands for disabling interrupts when executing critical portions of your code.

**Conclusion**

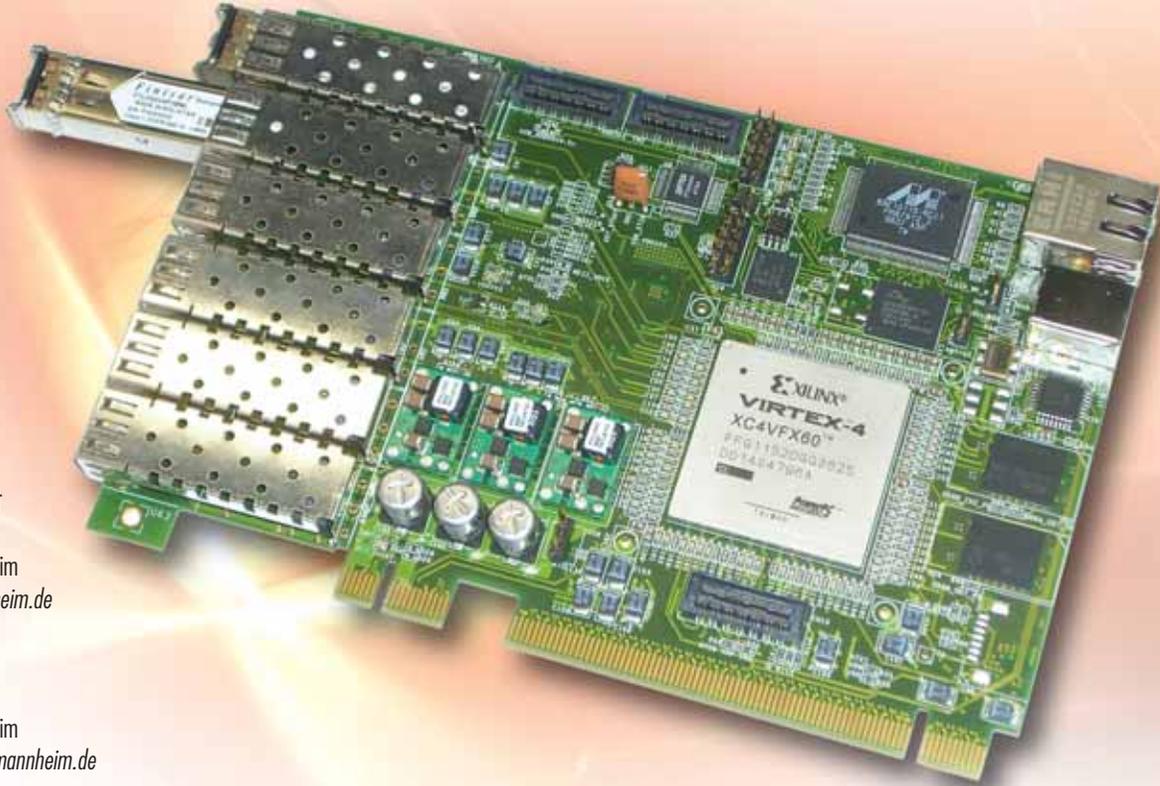
The Xilinx MOST NIC has been shown to efficiently handle control message processing. It is ideal for automotive infotainment systems, as it provides similar efficiency in streaming data. With upcoming device support for the Spartan-3A DSP family, this will allow even more efficient multimedia processing for your design.

**TAKE THE NEXT STEP** (Digital Edition: [www.xcellpublications.com/subscribe/](http://www.xcellpublications.com/subscribe/))

- Get more information about embedded design using the Xilinx MOST NIC LogiCORE solution in the MOST NIC lounge.
- The MOST NIC User Guide, available after generating a LogiCORE implementation, provides more information about all MOST control message formats for the MOST link layer.
- Download the PicoBlaze microcontroller reference design, which includes supporting files such as the assembler and extensive documentation such as the User Guide, with many coding examples and documentation for the complete instruction set.

# Leveraging HyperTransport on Xilinx FPGAs

An open-source HyperTransport IP core allows FPGAs to directly connect to AMD Opteron processors.



by David Slogsnat  
Research Associate  
University of Mannheim  
[slogsnat@uni-mannheim.de](mailto:slogsnat@uni-mannheim.de)

Alexander Giese  
Research Associate  
University of Mannheim  
[agiese@rumms.uni-mannheim.de](mailto:agiese@rumms.uni-mannheim.de)

Ulrich Bruening  
Professor  
University of Mannheim  
[bruening@uni-mannheim.de](mailto:bruening@uni-mannheim.de)

FPGA-based devices are used widely in standard computing systems, either as reprogrammable coprocessors or as prototyping devices. Usually, they are connected to the system using peripheral buses like PCI Express.

AMD's Opteron processors offer a unique opportunity to improve how devices are connected to the system. Rather than using a proprietary front-side bus, there are three HyperTransport (HT) interfaces per processor. The adoption of the HyperTransport Expansion Connector (HTX) specification and mainboards

equipped with this connector now make it possible to directly connect devices to an Opteron processor.

The Computer Architecture Group at the University of Mannheim researches networks and network interface controllers (NICs) for high-performance computing. For the construction of high-speed prototypes, we sought to implement an efficient HyperTransport IP on an FPGA. Implemented on a Virtex-4 FX device, the IP is capable of running in HT400 mode, with a link clock of 400 MHz. Of course, not only NICs benefit from a direct HyperTransport connection. Any device with high bandwidth or low latency requirements, like FPGA coprocessors, will benefit from the increased performance.

## Advantages of HyperTransport

Both HyperTransport and its closest competitor, PCI Express, are very capable of delivering bandwidth. For example, the 3.6-Gbps bi-directional bandwidth of our HT400 core is also possible using a PCI Express x8 device.

Besides bandwidth, there is a second important criterion to specify the performance of an interconnect: latency. Compared to PCI Express, HyperTransport adds significantly less latency to the transfer of data between a device and a host system comprising processors and memory.

One reason for this is the protocol itself, which does not require steps like 8B/10B coding and high-speed serialization. (A detailed latency analysis of both protocols

can be found on [www.hypertransport.org](http://www.hypertransport.org).) The main reduction in latency, however, stems from the fact that the HTX slot is directly connected to Opteron processors instead of having to go through one or more I/O bridges, as depicted in Figure 1. This avoids time-consuming protocol conversions and usually saves one chip-to-chip hop.

### Overview of HyperTransport

HyperTransport is a packet-based communication protocol for data transfer. There are three versions: HT 1.05 was developed in 2001 and updated to HT 2.0 in 2004. In April 2006, HT 3.0 was defined as the next successor. Current Opteron processors adhere to the HT 2.0b specification; no HT 3.0 devices or systems are currently available. Therefore, this article focuses on the implementation of an HT 2.0b device.

A HyperTransport link comprises two sets of unidirectional signals. Each set can be distinguished into three signal types: CAD (command, address, and data), CTL (control), and CLK (clock signals). The CAD lines are used to transport command and data packets, while the CTL line distinguishes between command and data packets on the CAD lines. The HyperTransport protocol supports CAD buses with a width of 2, 4, 8, 16, or 32 bit, as depicted in Table 1. The width of the CAD bus is usually known as the width of the HyperTransport link. If more than eight CAD lines are used per link and direction, every group of eight signals has its own CLK signal. These groups of signals are synchronously transmitted with the source-associated CLK signal.

The data transferred on the CAD bus is 32-bit aligned independent of the bus width. All transferred packets are at least the size of one doubleword (32 bit). HyperTransport allows clock frequencies from 200 MHz to 1.4 GHz in HT 2.0 and up to 2.6 GHz in HT 3.0. On the CAD bus, double-data-rate signaling is used. Current Opteron processors use 16-bit link widths and frequencies up to 1 GHz. In Opteron systems, all devices start at power up of the system with 200-MHz and 8-bit-

wide links. The BIOS checks the capabilities of all devices by accessing the device's HyperTransport register space and sets new values for frequency and width for every link according to the capabilities of the two devices that share the link. After that, it forces a re-initialization of all HyperTransport devices to establish the new parameters.

All transfers in HyperTransport are packet-based. To decouple the transmission

that is sent on the link belongs to this packet. A data packet can have a maximum size of up to 64 bytes.

Sending other control packets during a stream of data packets at every 32-bit boundary is allowed, but only if this control packet would not be followed by data. Otherwise it could not be possible to determine which control packet the data belongs to. This mechanism makes it possible to send urgent control packets with high priority. Flow control on the link is imposed by a credit-based protocol.

### Mastering Challenges in FPGA Design

The HyperTransport core has two different interfaces: one is the HyperTransport send and receive links, as specified in the HyperTransport protocol. On the other side, there is the application interface, which allows FPGA designs to access the HyperTransport core. This interface consists of three queues in each direction, one for every virtual channel. Applications can access these using a valid-stop synchronization mechanism. Control and attached data packets can be delivered simultaneously over the 160-bit-wide interface (96 bit to handle extended control packets, 64 bit for data packets).

Of course, our aim was to build a core that is as fast as possible. One limiting factor is the speed of the serial I/Os. In the Xilinx® Virtex™-4 FX

devices we used, the speed is limited to 400-MHz DDR, thus HT400. Xilinx SERDES blocks can parallelize/serialize the link by a factor of four so that the clock frequency of the core is 200 MHz. The SERDES blocks are also controlled by a bit-slip module to generate proper 32-bit boundary alignment.

Our second challenge was to process this data stream with the lowest number of pipeline stages and reasonable resource requirements. The decode unit, responsible for decoding the incoming packet stream and putting packets into the corresponding application interface queues or

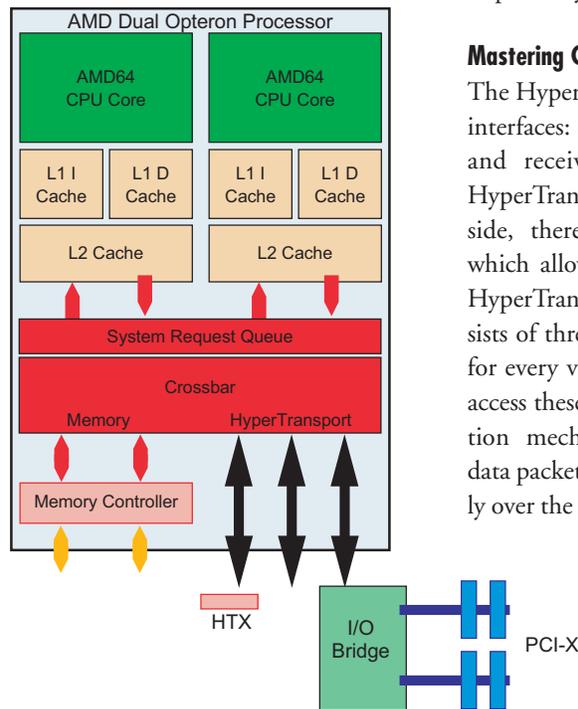


Figure 1 – Block diagram of an Opteron dual-core processor in a system with HTX and PCI-X slots

response from the request, the packets are transferred in a split phase transaction. A transfer always starts with one of three kinds of control packets: information, request, or response. Information packets are used for flow control and synchronization. Request packets are sent to write data to a receiver and are also used to initiate reads. Response packets contain the answer to a corresponding request.

Control packets have a size of 4 or 8 bytes or, if they use addresses of 64 bits instead of 40-bit addresses, the extended format has a size of 12 bytes. If a transfer contains payload data, the next data packet

Resource	8-Bit Link, HT200		16-Bit Link, HT400	
	Count	Usage	Count	Usage
Logic Slices	2,699	6.5%	6,700	15.8%
FIFO16/RAMB16s	30	7.5%	30	7.5%
DCM_ADVs	3	25%	4	33.4%
ISERDESs	10	1%	19	2%
OSERDES	9	1%	19	2%

Table 1 – Resource requirements in a Virtex-4 XC4VFX100 FPGA

Direction	Clock Cycles	Delay@ HT200	Delay@ HT400
In	11	55 ns	27.5 ns
Out	7	35 ns	17.5 ns

Table 2 – Hardware latency of the HyperTransport core

HyperTransport core units, is the most complex unit in the design. There may be two incoming 32-bit control packets at the same time, so there are two decode sub-units in parallel. On the other hand, they may be parts of one large 64-bit or 96-bit control packet.

Our third challenge was to reach an internal clock frequency of 200 MHz. We are currently using the core with only 100 MHz internally, and thus in HT200 for more than half a year. The HT400 version still requires some optimizations to run reliably at speed.

**Results**

Table 1 shows the resource requirements of the HT400 core (which we did not yet optimize for resource utilization) in com-

parison with an HT200 version that supports only 8-bit-wide links. The pure internal hardware latency of the core, shown in Table 2, is very low: 11 clock cycles for the input path from the HyperTransport link to the application interface and seven clock cycles outbound.

You can download the HT200 core from our website, as well as up-to-date information about the core and its performance. The HT400 core is still in the verification phase and will be available soon.

**Implementing a Low-Latency NIC**

As a university research group, our main focus is in high-performance computing. We used the HyperTransport core to build up a prototype NIC for the EXTOLL network. We developed this network to

research and implement new methods in high-performance networks and network interfaces. Figure 2 is a block diagram of such a NIC. The current prototype is implemented using the HT200 8-bit core with an internal clock frequency of 100 MHz. Nevertheless, the Netpipe ([www.scl.ameslab.gov/netpipe/](http://www.scl.ameslab.gov/netpipe/)) benchmark shows a latency of <1.5 μs on the software API level for sending small messages, which is an excellent result.

The hardware of our prototyping platform comprises the Iwill DK8-HTX dual Opteron mainboard and the HTX board. The Xilinx Virtex-4 FPGA-based board that we developed in 2006 can be plugged directly into the HTX slot of the Iwill mainboard. This is also the only environment in which we have verified both the core and the board in depth thus far.

**Conclusion**

The HT IP core successfully exploits the potential of the FPGA in terms of bandwidth, latency, and resource utilization. Offering an HT400 connection and thus 3.6 GB of bi-directional bandwidth, the HyperTransport core can be used for more than just prototyping – its performance is sufficiently good enough to serve as a production coprocessor as well.

The HyperTransport core can also be mapped to an ASIC implementation to reach higher link speeds. For FPGAs, faster links may be possible with faster FPGA families such as the Virtex-5 device family.

We are also working on a coherent HyperTransport core with HT400 speed that allows devices to participate in the cache-coherence protocol of Opteron processors. In addition to the functionality of a peripheral device, this will, for example, allow a device or coprocessor to act like an Opteron system memory controller or a coherent cache. In contrast to the non-coherent HyperTransport, the use of the coherent version requires a license from AMD.

For more information about the open-source HyperTransport core or the HTX board, visit our Center of Excellence for HyperTransport Technology: [www.ra.informatik.uni-mannheim.delcoehb](http://www.ra.informatik.uni-mannheim.delcoehb).

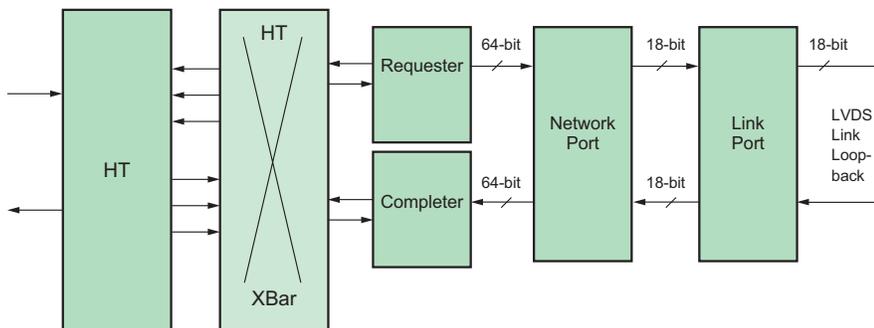


Figure 2 – Prototype NIC device

# New PCI Express Analog/FPGA Solutions



## Ultra Performance X5 Modules

Xilinx Virtex5 FPGA logic  
 1GB DDR2 DRAM  
 4MB QDR SRAM  
 8 Rocket IO Private Links  
 2.5 Gbps each  
 >1GB/s  
 8-lane PCI Express Host

### Available I/O Solutions:

X5-400M:  
 400 MSPS,  
 14-bit A/D (x2)  
 14-bit D/A (x2)  
 1GB Memory



## Flexible PMC Module Form-Factor

Range of adapters supports module use in desktop PCs, industrial cPCI chassis, cabled, dedicated 2.5 Gbps PCIe LANs, 100 Mbps ethernet WANs or completely stand-alone!

Continuous stream rates:  
 1 GB/s (X5), 266 MB/s (X3)

Stand-alone (embedded) carriers

Completely autonomous or tethered to PCs or network

Cabled PCIe: 10 meter, 200 MB/s cable to PC/cPCI

10/100 Ethernet : Autonomous

12V dc-only operation



## Low Cost X3 Modules

Xilinx Spartan 3 FPGA (1 or 2 MGate)  
 Two 2MB SRAMs, 48-bit DIO  
 PCI Express with >200 MB/s data rates

Available I/O Solutions:

X3-25M: 16-bit, 25MHz, A/D + D/A (x2)

X3-A4D4: 16-bit, 4 MHz A/D+ D/2 (x4)

X3-DIO: 64-bit, 66 MHz LVDS

X3-SD: 216 kHz, 16-bit, A/D (x16)

X3-SDF: 20 MHz, 24-bit, A/D (x4)

X3-Servo: 16-bit, 250 kHz, A/D+D/A (ultra low-latency & glitch energy)

Using Innovative Integration's **FrameWork Logic VHDL source-code** or **MATLAB board support package** you can readily customize FPGA functionality to include real-time processing such as independent FIR and IIR filters on each channel, real-time FFT processing, ultra-fast feedback and control loops and much more.

### Full-featured C++ libraries

allow rapid, seamless integration into OEM equipment, high-speed data loggers/waveform generators or custom instrumentation.

### Develop DSP in MATLAB Simulink ...then go straight to hardware!

Using MATLAB Simulink with Xilinx System Generator, develop DSP systems in the MATLAB environment then run them directly on hardware. All with bit-true, cycle-true results that bring the real world data and hardware into MATLAB. With our FrameWork Logic and MATLAB board support packages, you can quickly integrate signal processing into the hardware without lengthy, complex coding!



FrameWork Logic



www.innovative-dsp.com • sales@innovative-dsp.com

**Innovative Integration**  
 ... real time solutions!

# FPGA-Based Simulation for Rapid Prototyping

You can run an HDL simulator along with an FPGA through USB using iNCITE.

by Ando Ki  
R&D Director  
Dynalith Systems  
adki@dynalith.com

Designing a hardware block does not simply mean RTL coding and simulation; the process usually includes FPGA-based prototyping. To do this requires a huge effort – including the preparation of a PCB board that implements your design and its surrounding functions in hardware – even though you probably only want to see the functionality of your design and not its surrounding components.

One possible solution around this has actually existed for some time: conventional emulation. This hardware-assisted acceleration technique connects an HDL simulator to programmable devices, allowing a design to run on real hardware while its surrounding functions are simulated on top of software.

In this article, I'll describe Dynalith Systems's iNCITE USB-connected FPGA board, supporting Xilinx® Spartan™-3 FPGAs. The iNCITE board provides a USB-based communication channel between software and an FPGA. Using industry-standard HDL simulator software, you can run your design in the FPGA under software control.

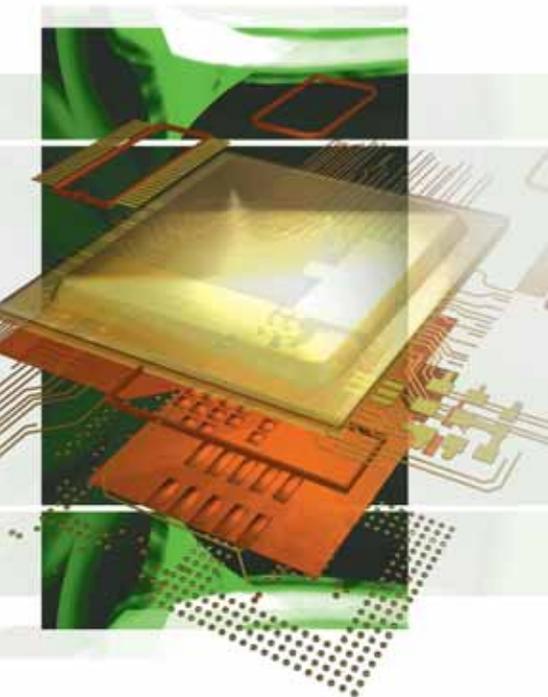
## iNCITE

iNCITE is an FPGA-mounted PCB board featuring various on-board memories, a

host computer interface using USB 2.0, and a board-to-board connector (see Figure 1). For application-specific peripherals, the iNCITE-AVREM application board incorporates audio, video, RS-232C, Ethernet, MMC, and PS/2. With both boards, you can easily implement a complete system-on-chip (SOC) or embedded system.

## Design Flow for FPGA-Based Simulation

As shown in Figure 2, there are four design steps. In the first step, the design under test (DUT) and its test bench are developed using a pure software simulator, such as the free ModelSim-XE HDL simulator. When a synthesizable RTL version of the DUT is ready, the DUT is synthesized using an industry-standard FPGA synthesizer such as XST (Xilinx Synthesis Technology), a proprietary synthesis engine in Xilinx ISE™ software.



The synthesis result EDIF is processed by iNSPIRE-Lite, which is the GUI-based integrated design environment for iNCITE (Xilinx place and route is used internally). During this step, two files are generated: an emulation information file (EIF) and a proxy module. The EIF contains the necessary data and bitstream to configure the FPGA. The proxy module handles communication between the simulator and the FPGA through USB.

The test bench runs on top of the HDL simulator while the DUT runs in the FPGA. During simulation with iNCITE, the EIF is automatically downloaded to the FPGA through the USB channel at the start of simulation.

The channel between iNCITE and software is a generic one; thus, you can use any programming language including C, C++, SystemC, and MATLAB/Simulink. Essentially, you can build a virtual system

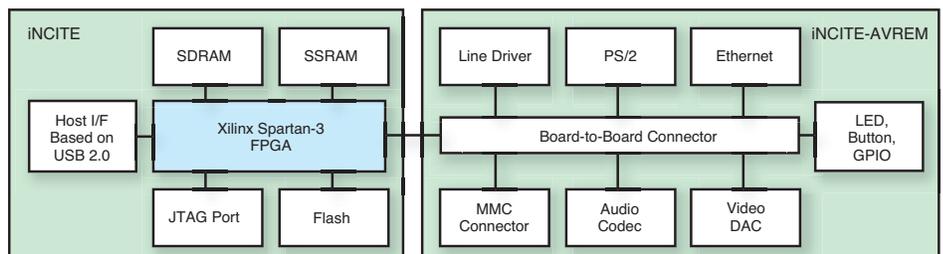


Figure 1 – Functional block diagram of iNCITE and iNCITE-AVREM

around the actual FPGA without preparing a PCB board, since the DUT is implemented in the FPGA and its surrounding blocks are modeled using your preferred language, including HDL and high-level languages.

### Rapid Prototyping

Functional verification utilizing hardware has three major categories: logic simulation acceleration, emulation, and prototyping. Acceleration is based on the same idea described previously: some parts of the design run in the hardware while other parts are simulated on top of the software.

Emulation uses special hardware in the context of a real environment, where our design runs in programmable devices connected to a target hardware board.

Prototyping is a customized emulation system encompassing all parts of a system, including user design. It is usually implemented using an FPGA and PCB board. In other words, prototyping requires that you design the system before final production. Although the first two categories normally use off-the-shelf products, prototyping is time-consuming and costly, as it requires PCB design and debugging.

To assist beginners or those designing small- to medium-scale projects, the iNCITE application board can serve as a ready-to-prototype system (Figure 3). With this system, you can easily implement a complete SOC or embedded system. For the example shown in Figure 3, we built and mapped a complete OpenRISC-based SOC on the FPGA using iNCITE, while other peripherals and memories were incorporated using iNCITE-AVREM.

### Conclusion

iNCITE, iNCITE-AVREM, and iNSPIRE-Lite provide an ideal design and verification environment, allowing you to run your design through an FPGA board. You can work in the same test bench from RTL design to FPGA-based gate-level verification without having to prototype a PCB.

These tools are also ideal for teaching, seminars, and small- and medium-sized

designs, as the environment runs the gamut from pure simulation to FPGA prototyping through logic synthesis and FPGA place and route.

For more information about iNCITE, visit [www.dynalith.com/incite.php](http://www.dynalith.com/incite.php). For more information about Dynalith Systems, visit [www.dynalith.com](http://www.dynalith.com).

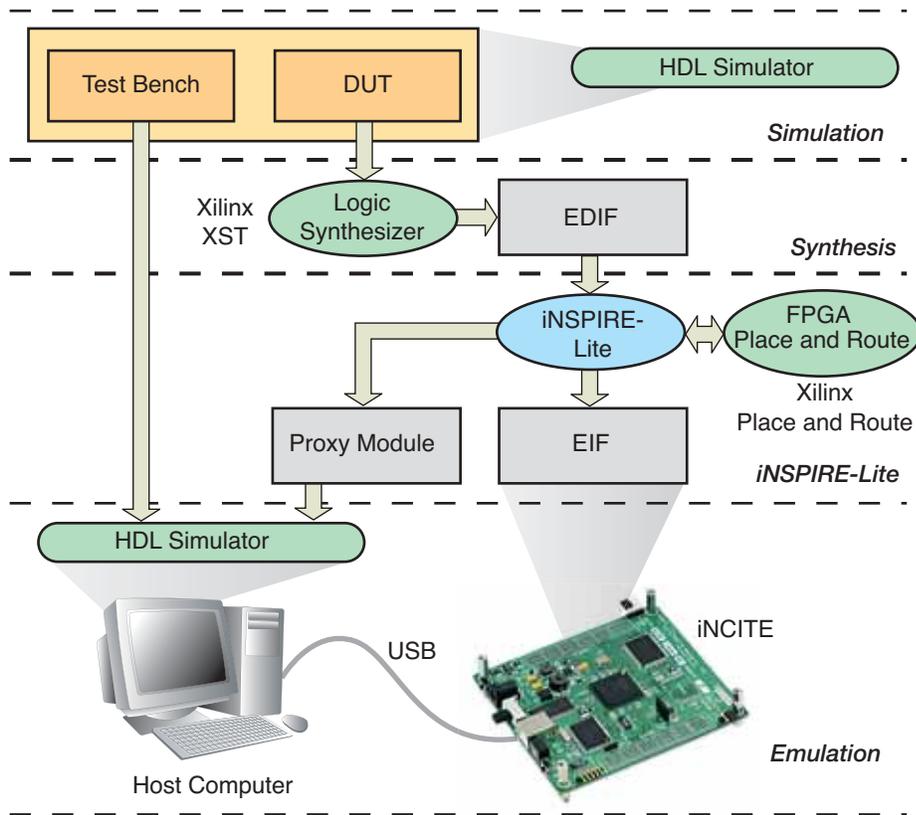


Figure 2 – iNCITE design flow

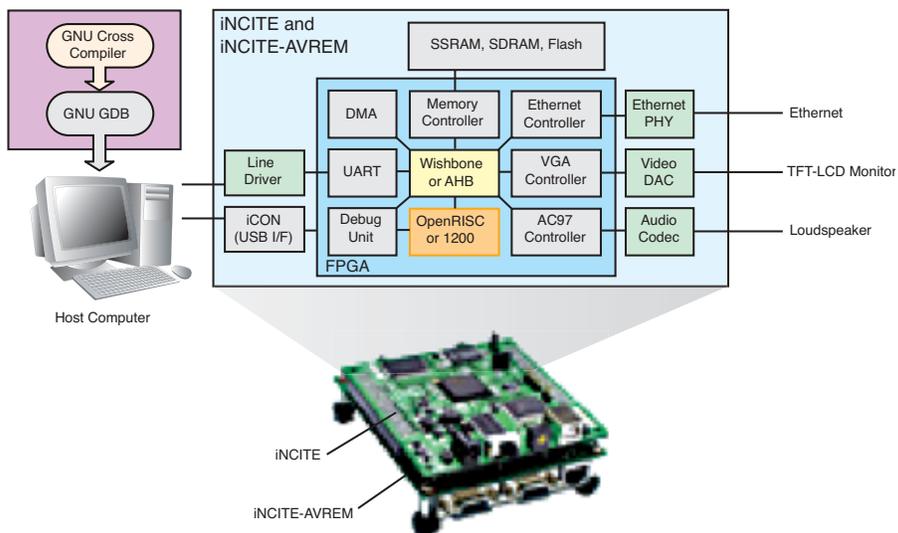


Figure 3 – Prototyping example of OpenRISC-based SOC



# Featured Connectivity Application Notes

Create a differentiated connectivity solution with product descriptions and associated design files.

## Embedded Serial ATA Storage System

[www.xilinx.com/bvdocs/appnotes/xapp716.pdf](http://www.xilinx.com/bvdocs/appnotes/xapp716.pdf)

This application note describes the design and implementation of an embedded Serial Advanced Technology Attachment (SATA) storage system on a Xilinx® Virtex™-4 platform. SATA is the successor of the prevalent Parallel Advanced Technology Attachment (PATA) interface. SATA overcomes many limitations of PATA and offers a maximum bandwidth of 150 Mbps. Combining SATA with a high-speed network interface, Gigabit Ethernet (1000 Base-X) creates an ideal solution for many high-performance storage applications such as network attached storage (NAS), storage area network (SAN), and redundant array of independent disks (RAID).

## Audio/Video Connectivity Solutions for the Broadcast Industry

[www.xilinx.com/bvdocs/appnotes/xapp514.pdf](http://www.xilinx.com/bvdocs/appnotes/xapp514.pdf)

This comprehensive reference design describes how to use Xilinx FPGAs to implement serial digital video interfaces commonly used in the professional video broadcast industry. The serial video interfaces described in this document are:

- SD-SDI (SMPTE-259M), used to transport uncompressed standard-definition digital video
- HD-SDI (SMPTE-292M), used to transport uncompressed high-definition digital video
- DVB-ASI, used to transport compressed digital video
- AES, used to transport digital audio



## Virtex-5 Embedded Tri-Mode Ethernet MAC Hardware Demonstration Platform

[www.xilinx.com/bvdocs/appnotes/xapp957.pdf](http://www.xilinx.com/bvdocs/appnotes/xapp957.pdf)

This application note describes a system using the Virtex-5 embedded tri-mode Ethernet MAC (Ethernet MAC) wrapper core on an ML505 development board. The system provides an example of how to integrate the embedded tri-mode Ethernet MAC and embedded tri-mode Ethernet MAC wrapper, using a hardware design to target the development board and a PC-based GUI to control the demonstration platform.

## 10 Gigabit Ethernet Hardware Demonstration Platform

[www.xilinx.com/bvdocs/appnotes/xapp955.pdf](http://www.xilinx.com/bvdocs/appnotes/xapp955.pdf)

This application note describes the functionality of the LogiCORE™ 10 Gigabit Ethernet and XAUI cores in Xilinx FPGA hardware. Development board requirements, setup instructions, MAC core-specific design components, and a description of the GUI used to control the demonstration platform are included. 🌈

# Connectivity Boards and Kits



Xilinx, together with distributors and partners, offers a complete line of development and expansion boards to help you test and develop designs using Xilinx® devices. The following boards are specific to connectivity applications.

## Virtex-5 LXT ML505 Evaluation Platform

[www.xilinx.com/xlnx/xebiz/designResources/ip\\_product\\_details.jsp?key=HW-V5-ML505-UNI-G](http://www.xilinx.com/xlnx/xebiz/designResources/ip_product_details.jsp?key=HW-V5-ML505-UNI-G)

Create high-speed serial designs using Virtex™-5 RocketIO™ GTP transceivers:

- Provides a feature-rich general-purpose evaluation and development platform
- Includes on-board memory and industry-standard connectivity interfaces
- Delivers a versatile development platform for embedded applications



## Virtex-5 LXT ML52x RocketIO Characterization Platforms

[www.xilinx.com/xlnx/xebiz/designResources/ip\\_product\\_details.jsp?key=HW-V5-ML52X-UNI-G](http://www.xilinx.com/xlnx/xebiz/designResources/ip_product_details.jsp?key=HW-V5-ML52X-UNI-G)



Xilinx ML52x platforms are ideal for characterization and evaluation of Virtex-5 LXT RocketIO GTP transceivers. Each RocketIO GTP transceiver is accessible through four SMA connectors. ML52x platforms are available with XC5VLX50T-FF665 (ML521), XC5VLX110T-FF1136 (ML523), and XC5VLX330T-FF1738 FPGA devices.

## Virtex-5 LXT ML555 FPGA Development Kit for PCI Express and PCI

[www.xilinx.com/xlnx/xebiz/designResources/ip\\_product\\_details.jsp?key=HW-V5-ML555-G](http://www.xilinx.com/xlnx/xebiz/designResources/ip_product_details.jsp?key=HW-V5-ML555-G)



The Virtex-5 LXT FPGA Development kit for PCI Express supports PCIe/PCI-X/PCI. This complete development kit passed PCI-SIG compliance for PCI Express specification v1.1 and enables you to rapidly create and evaluate designs using PCI Express, PCI-X, and PCI interfaces.

## Virtex-5 SXT ML506 Evaluation Platform

[www.xilinx.com/xlnx/xebiz/designResources/ip\\_product\\_details.jsp?key=HW-V5-ML506-UNI-G](http://www.xilinx.com/xlnx/xebiz/designResources/ip_product_details.jsp?key=HW-V5-ML506-UNI-G)

The ML506 is a feature-rich DSP general-purpose evaluation and development platform. Though economically priced, the ML506 offers you the ability to create DSP-based and high-speed serial designs using the Virtex-5 DSP48E slices and RocketIO GTP transceivers. A variety of on-board memories and industry-standard connectivity interfaces add to the ML506's ability to serve as a versatile development platform for embedded applications. ●●●





# The Connectivity Curriculum Path

Xilinx Education Services is ready to help you use PCI Express, Gigabit Ethernet, or Rapid IO protocols in your next design.

by Craig Willert

Global Training Solutions Communications Manager  
Xilinx, Inc.  
[craig.willert@xilinx.com](mailto:craig.willert@xilinx.com)

By learning advanced programmable logic design techniques and methodologies, you can take full advantage of today's advanced FPGA capabilities, including high-speed serial I/O. Equipped with this knowledge, you can better innovate when developing products for your market, reduce R&D costs through improved process efficiency, and lower production costs by using smaller devices in a slower speed grade.

The Xilinx® Connectivity Curriculum Path offers a variety of classes to help you quickly understand how to build the most optimal system using one of the many high-speed serial I/O protocols that Xilinx supports. The curriculum path also includes courses on the use of Xilinx RocketIO™ GTP serial transceivers, signal integrity, leading-edge programmable logic technology, and Xilinx design flows.

## Designing with Multi-Gigabit Serial I/O

The “Designing with Multi-Gigabit Serial I/O” course will teach you how to employ RocketIO transceivers in your Virtex™-5 LXT FPGA designs.

After completing this comprehensive training, you will have the skills to:

- Describe and utilize the ports and attributes of the RocketIO multi-gigabit transceiver in the Virtex-5 LXT FPGA
- Effectively use such features as:
  - Comma detection, CRC, clock correction, and channel bonding
  - 8B/10B encoding/decoding, programmable termination, and pre-emphasis

- GTP primitive instantiation in a design using the GTP wizard
- Reference material for board design issues
- Power supply, oscillators, and trace design

## Designing a LogiCORE PCI Express System

The “Designing a LogiCORE™ PCI Express System” course focuses on the key PCI Express protocol subjects and targets hard and soft PCI Express cores in the Virtex-5 FPGA.

This course is ideal for:

- Hardware designers creating applications using Xilinx IP cores for PCI Express
- Software engineers creating APIs, GUIs, and driver software development
- System architects leveraging Xilinx performance, latency, and bandwidth

After completing this training, you will have the necessary skills to:

- Use Xilinx PCI Express cores in your own design environments
- Select the appropriate PCIe solution for a specific application
- Identify how PCI Express specifications apply to Xilinx PCI Express cores

## Designing with Ethernet MAC Controllers

The “Designing with Ethernet MAC Controllers” course teaches the basics of the Ethernet standard, protocol, and OSI model and related Xilinx solutions.

This course is ideal for:

- Engineers using Xilinx Ethernet connectivity solutions

After completing this training, you will have the necessary skills to:

- Use various Ethernet cores alone or as a peripheral in processor-based designs
- Select the appropriate core for a specific design
- Develop software to drive the core and achieve desired functionality

## Conclusion

Xilinx programs provide targeted, high-quality education products and services that are designed by experts in programmable logic design and delivered by Xilinx-qualified trainers. We offer classes at all expertise levels and create an engaging learning environment by blending lectures, hands-on labs, interactive discussions, tips, and best practices.

Xilinx delivers training when and where you want it by leveraging our global network of authorized training providers (ATPs) and online learning systems. 🌟

### TAKE THE NEXT STEP (Digital Edition: [www.xcellpublications.com/subscribe/](http://www.xcellpublications.com/subscribe/))

- Register today for any of these courses.
- View the full Connectivity Curriculum Path.
- Contact your Xilinx sales representative.

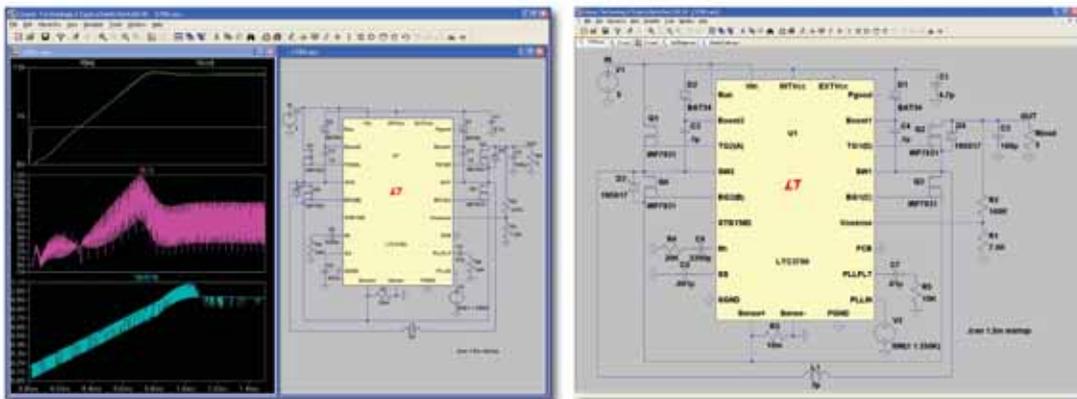
# LTspice/SwitcherCAD III Seminar Series



Featuring  
Mike Engelhardt  
(Author/Creator)

In this session, you will learn the ins and outs of LTspice/SwitcherCAD™ III including informative tutorials on how to simulate switch mode power supplies, compute efficiencies and observe power supply start-up behavior and transient response. You will also utilize LTspice/SwitcherCAD III as a general purpose SPICE simulator for AC analysis, DC sweeps, noise analysis and circuit simulations. At the end of the presentation, a complete question and answer period will be offered to answer any of your unique questions on LTspice/SwitcherCAD III.

- Over 1 million engineers have downloaded Linear Technology's LTspice/SwitcherCAD III program.
- Every 4.5 minutes another free copy of Linear Technology's LTspice/SwitcherCAD III program is downloaded.
- Come to our seminar and learn first hand from the program's author, Mike Engelhardt.



Register Online by Visiting  
[www.nuhorizons.com/xpresstrack/lcswitcher](http://www.nuhorizons.com/xpresstrack/lcswitcher)



## Upcoming XpressTrack Seminars

### INDIA

#### LTspice/SwitcherCAD III Seminar

- 20 Aug Bangalore, India
- 21 Aug Mumbai, India
- 22 Aug Delhi, India

### NORTH AMERICA

#### LTspice/SwitcherCAD III Seminar

- Sep 18 Orlando, FL
- Sep 19 Atlanta, GA
- Sep 20 Raleigh, NC
- Oct 2 Seattle, WA
- Oct 3 Portland, OR
- Oct 4 Denver, CO
- Oct 5 Salt Lake City
- Oct 9 San Jose, CA
- Oct 10 San Jose, CA
- Oct 23 Boston, MA
- Oct 24 Rochester, NY
- Oct 25 Syracuse, NY
- Oct 26 Philadelphia, PA

For a complete list of course offerings, or to register for a seminar near you, please visit:

[www.nuhorizons.com/xpresstrack](http://www.nuhorizons.com/xpresstrack)

## New Xilinx Development Board

The new Virtex-5 LXT Platform FPGA is optimized for high-performance logic with low-power serial connectivity. Learn about the features and capabilities of the Virtex-5 LXT Platform FPGA from Xilinx using this general purpose evaluation kit from Nu Horizons.



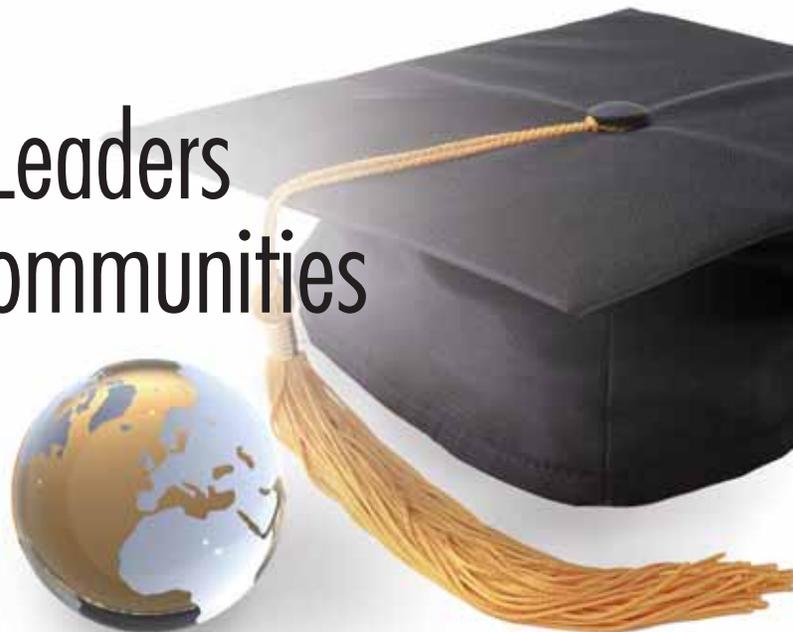
For more information on this board and our complete family of Xilinx development tools, please visit:

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



# Developing Technical Leaders through Knowledge Communities

Knowledge communities will better prepare students to become team players and problem solvers.



by Ivo Bolsens  
CTO  
Xilinx, Inc.  
[ivo.bolsens@xilinx.com](mailto:ivo.bolsens@xilinx.com)

Today's college freshmen will enter the workforce in the 2010-2011 time frame. By then, chip manufacturing will be at the 32-nm technology mode, allowing manufacturers to build single chips that contain tens of billions of transistors.

FPGAs will be the leading platforms leveraging this manufacturing technology, capable of implementing programmable systems that will comprise nearly 100 million programmable logic gates. As such, the FPGA will become the heart of many electronic products.

The main building blocks that will make up a system in an FPGA will be soft processors running complex embedded software; highly parallel or pipelined arithmetic functions executing real-time signal-processing algorithms; packet-processing engines securing embedded connectivity; and complex hierarchies of memories handling efficient data transfer and storage mechanisms. These programmable systems will be adaptable and partially reconfigurable in the field.

Triple play (the convergence of data, voice, and video over a peer-to-peer network) will be the driving force setting the requirements for these platforms. The pace of this evolution, however, leaves very little time to educate today's students so that they become tomorrow's technical leaders. The challenge

and complexity of this task by far exceeds the capacity and domain of a single professor.

## Knowledge Communities

For this reason, industry and academia have come together to create "knowledge communities" around specific technology domains. Universities seek to create a wide association of people with common technology interests who wish to continue learning, enhancing the skills of their members and preserving lessons learned. The industry wants to be an active partner, educating those in academia in global system issues and creating a realistic agenda through intensive dialogue between industry leaders and visionary academics.

As these knowledge communities take hold, students will be better prepared to become the team players and problem solvers for future industry projects. Gradually, they will ascend the "knowledge value chain," from reproducing textbook content to applying judgment calls in situational problems.

Xilinx is constantly deepening its partnership with academia and is committed to jointly developing new ideas and opportunities and solving key challenges.

Triple play will permit FPGA platforms to become the core infrastructure; however, there are challenges to solve. First is the further digitization and processing of signals with growing resolutions, requiring an exponential growth in DSP compute capability. Second is the secure and reliable transport of this data with increasing band-

width over wireless and wired infrastructures. And third is the need for high-performance computing to extract information from large amounts of raw data.

For example, the RAMP research community (<http://ramp.eecs.berkeley.edu>) is exploring a programming environment based on the BEE2 FPGA platform. The NetFPGA platform (<http://yuba.stanford.edu/nf2/>) provides an open platform for research on high-bandwidth networking applications. And the WARP project (<http://warp.rice.edu>) has built a repository to support complex wireless DSP application designs.

These activities are part of the Xilinx university outreach program, supporting continuing education and research based on our latest technology. The Xilinx University Program (XUP, [www.xilinx.com/univ/](http://www.xilinx.com/univ/)) not only allows easy access to Xilinx best-in-class tools but also provides free training, high-quality support, and curriculum development assistance. XUP donations take many forms, from design software, FPGA devices, and IP cores to reference designs and student competitions.

As a business that is continually looking globally for new ideas and technologies, Xilinx has established relationships with a large number of universities around the world to allow their engineering students to receive hands-on experience with our technology. These students will enter the workforce with the necessary skills to be quickly productive in the fast-paced world of semiconductors. ●●

# PERFORMANCE



Tired of spinning in circles to meet timing on your FPGA designs?

Synplicity's **Synplify® Premier** solution solves FPGA timing closure through a unique graph-based physical synthesis technology providing highly accurate correlation to final timing and 5 to 20% better performance in a single-pass flow.



FPGA Implementation

For more information on the Synplify Premier product and how it can help you quickly reach aggressive timing goals for your FPGAs, visit:

[www.synplicity.com/products](http://www.synplicity.com/products)



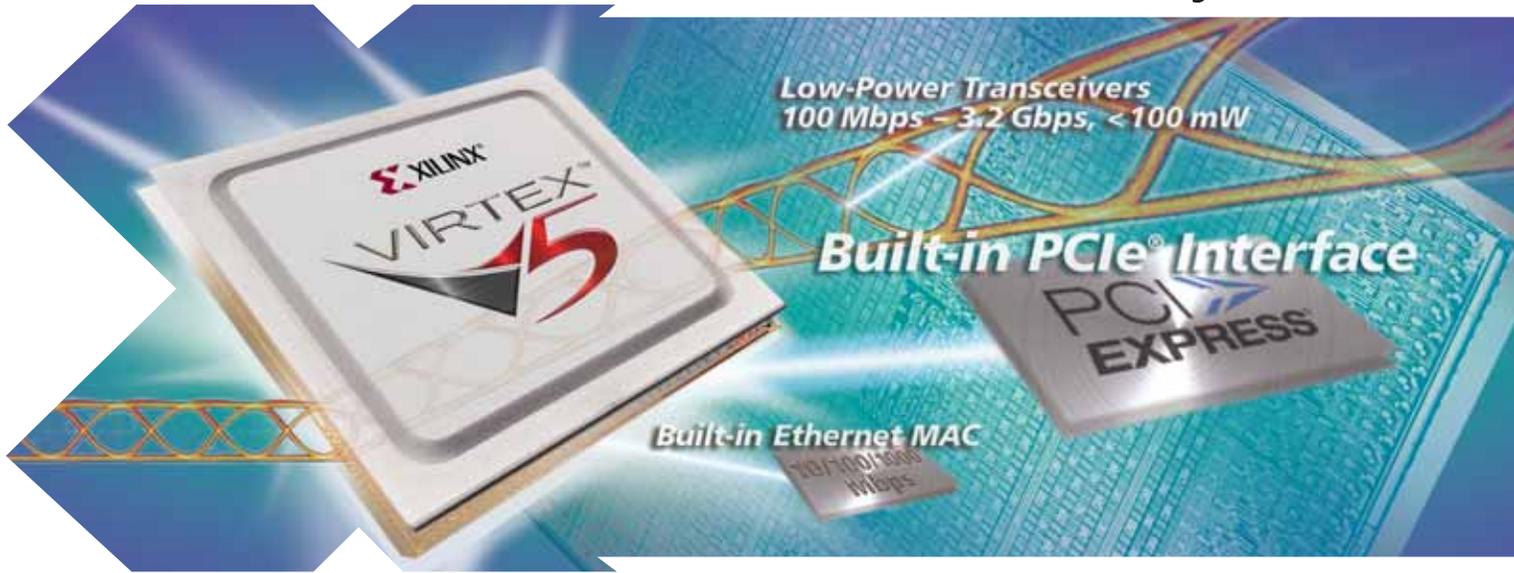
**Synplicity®**

Simply Better Results

The **Synplify Premier** tool was awarded the prestigious LSI of the Year Grand Prix Award in June 2006. This award is sponsored by Semiconductor News (Japan). The software was also a finalist for the 2005 EDN Innovation Award, and 2005 DesignVision Award.

# LOW-POWER TRANSCEIVERS

## Ultimate Connectivity . . .



**Reduce serial I/O power, cost and complexity with the world's first 65nm FPGAs.**

With a unique combination of up to 24 low-power transceivers, and built-in PCIe® and Ethernet MAC blocks, Virtex™-5 FPGAs get your system running fast. Whether you are an expert or just starting out, only Xilinx delivers everything you need to simplify high-speed serial design, including protocol packs and development kits.

### Lowest-power, most area-efficient serial I/O solution

RocketIO™ GTP transceivers deliver up to 3.2 Gbps connectivity at less than 100 mW to help you beat your power budget. The embedded PCI Express endpoint block saves 60% in power and 90% in area compared to other solutions. Virtex-5 chips and boards are on the PCI-SIG® Integrators List. Plus our UNH-tested Ethernet MAC blocks make connectivity easier than ever.

Visit our website to get the chips, boards, kits, and documentation you need to start designing today.



**Available Now!**  
Built-in x1-x8 PCIe Solution  
\$2,200 USD MSRP

### Complete PCI Express® Development Kit



**Delivering Benefits of 65nm FPGAs Since May 2006!**



Optimized for logic, serial connectivity and DSP.



[www.xilinx.com/virtex5](http://www.xilinx.com/virtex5)

**The Ultimate System Integration Platform**