ISSUE 57, SECOND QUARTER 2006 XCELL JOURNAL XILINX, INC.

Issue 57
Second Quarter 2006

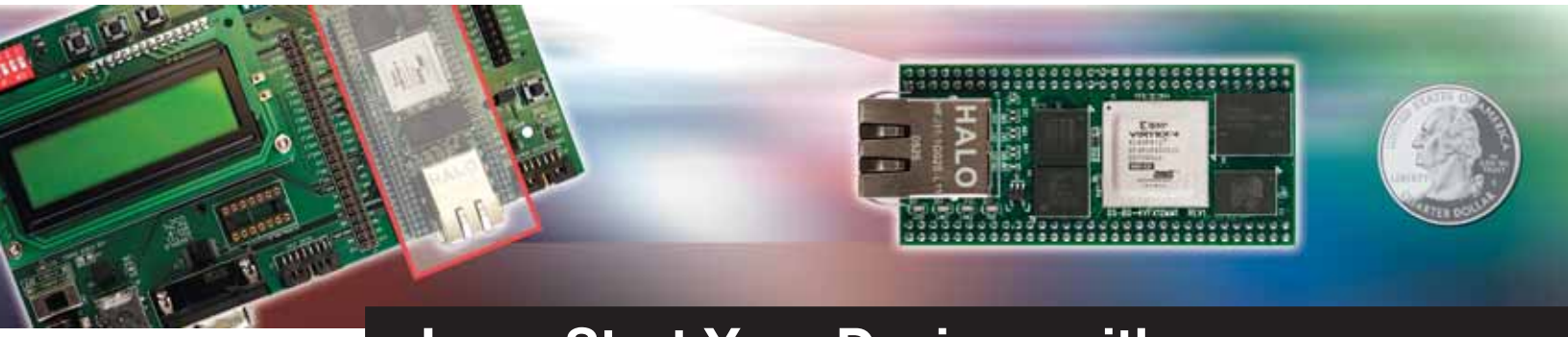
Xcell journal

THE AUTHORITATIVE JOURNAL FOR PROGRAMMABLE LOGIC USERS

Vertical Markets: Piecing the Triple Play Puzzle

XILINX®

Support Across The Board.™



Jump Start Your Designs with Mini-Module Development Platforms

Mini-Module Features

- Small footprint (30 mm x 65.5 mm)
- Complete system-on-module solution with SRAM, Flash, Ethernet port and configuration memory
- Supports MicroBlaze™ or PowerPC® processors
- Based on the Virtex™-4 FX12 and 400K Gate Spartan™-3 FPGAs
- 76 User I/O

Baseboard Features

- Supports mini-module family
- Module socket allows easy connection
- Provides 3.3 V, 2.5 V and 1.2 V
- USB and RS232 ports
- 2 x 16 character LCD

Upcoming SpeedWay Design Workshops

- Developing with MicroBlaze
- Developing with PowerPC
- Xilinx Embedded Software Development
- Embedded Networking with Xilinx FPGAs

The Xilinx® Spartan™-3 and Virtex™-4 FX 12 Mini-Modules are off-the-shelf, fully configurable, complete system-on-module (SOM) embedded solutions capable of supporting MicroBlaze™ or PowerPC® embedded processors from Xilinx.

Offered as stand-alone modules and as development kits with companion baseboards, power supplies, software and reference designs, these modules can be easily integrated into custom applications and can facilitate design migration.

Gain hands-on experience with these development kits and other Xilinx tools by participating in an Avnet SpeedWay Design Workshop™ this spring.

Learn more about these mini-modules and upcoming SpeedWay Design Workshops at:

www.em.avnet.com/spartan3mini
www.em.avnet.com/virtex4mini
www.em.avnet.com/speedway

XILINX®

AVNET® electronics marketing

Avnet Green Initiative

Supplier Authorized Distributor

Enabling success from the center of technology™

1 800 332 8638
em.avnet.com

Xcell journal

PUBLISHER Forrest Couch
forrest.couch@xilinx.com
408-879-5270

EDITOR Charmaine Cooper Hussain

ART DIRECTOR Scott Blair

ADVERTISING SALES Dan Teie
1-800-493-5551

TECHNICAL COORDINATOR Cindy Kao

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/

With Vertical Markets, There's Nowhere to Go But Up

It's a testament to the versatility of Xilinx® products and the ever-widening scope of applications that we have so many writers willing to share their knowledge on a boggling number of topics in this issue.

As Senior Director of Vertical Markets and Partnerships Krishna Rangasayee writes in his article, "Scoping the Next Killer App – Triple Play," Xilinx initially chose to focus its efforts on six vertical markets as a recovery strategy from the steep declines at the turn of the 21st century. Today, Xilinx supports a total of eight vertical markets, which are aerospace/defense, automotive, broadcast, consumer, industrial/scientific/medical (ISM), storage and servers, wired, and wireless.

The *Xcell Journal* is proud to have featured articles describing innovative applications in each of these vertical markets, and in this vertical markets-themed issue, we've got even more. "Xilinx Low-Cost Display Solutions" and "IDE Hard Disk Drive Solutions for Smart Handheld Applications" are consumer-oriented. "WiMAX – Delivering on the Promise of Wireless Broadband" fits into wired/wireless. "Data Mining Without Limits" has applications in aerospace/defense and ISM ("Accelerating Scientific Applications Using FPGAs" is a cool app befitting the latter category as well).

We also have a strong collection of articles for the storage and servers market, including "Enabling Enterprise-Wide Data Security," "Xilinx in Enclosure Management," and "Co-Processing Solutions for Delivering Storage Servers Over Distance."

Although topically diverse, the articles in this issue unanimously emphasize the strength and suitability of PLDs today and tomorrow. As vertical markets evolve and applications currently unimaginable spring to life, the key is to remain flexible, adaptable – and reprogrammable.

Forrest Couch
Publisher

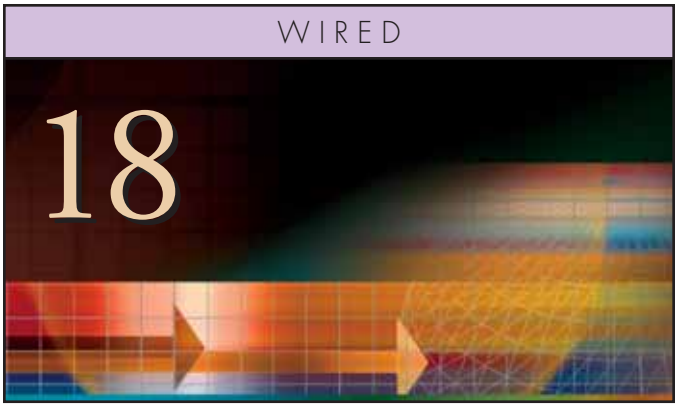
WIRELESS

FPGAs Enable Pseudo Wire Emulation Edge-to-Edge
Network convergence is possible using PWE3 for transport and delivery.

INDUSTRIAL/SCIENTIFIC/MEDICAL

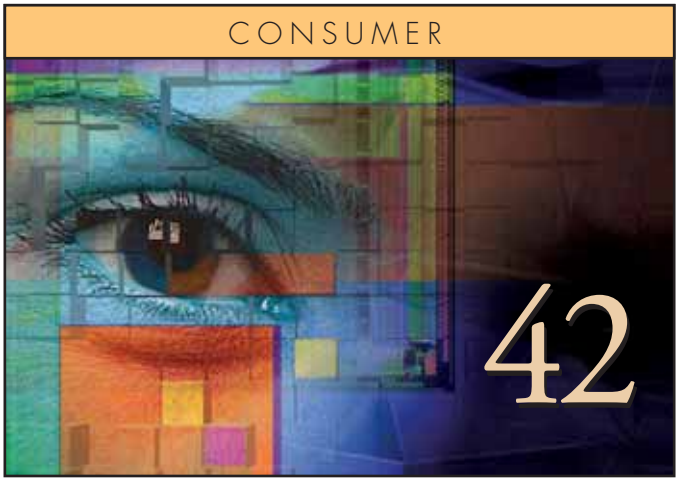
Designing an FPGA-Based RFID Reader
You can implement a practical, standards-compliant RFID reader using off-the-shelf RF components and FPGAs.

WIRED



Enabling Next-Generation MAN Equipment
You can develop flexible, IEEE 802.17 Resilient Packet Ring-compliant equipment faster and with less resources using the Xilinx RPR MAC reference design.

CONSUMER

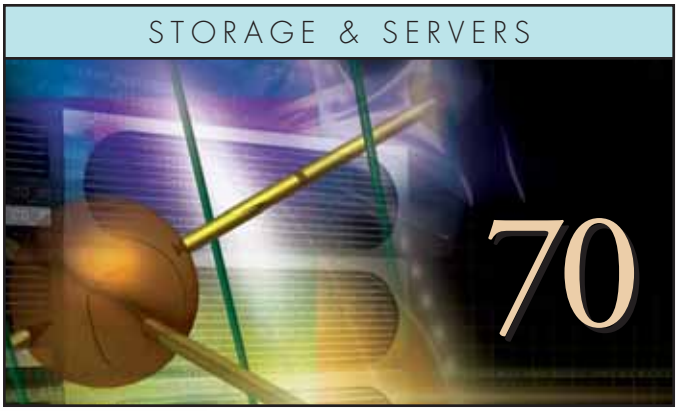


High-Performance Image Processing on FPGAs
You can develop and implement sophisticated imaging modules on low-cost devices.

Cover

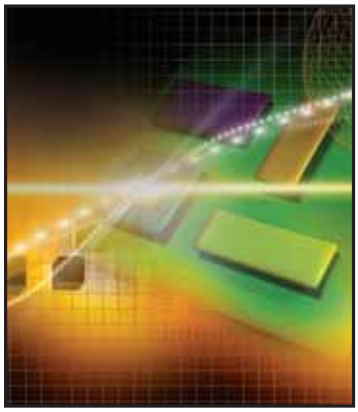
Scoping the Next Killer App – Triple Play
How consumers are influencing the Xilinx roadmap.

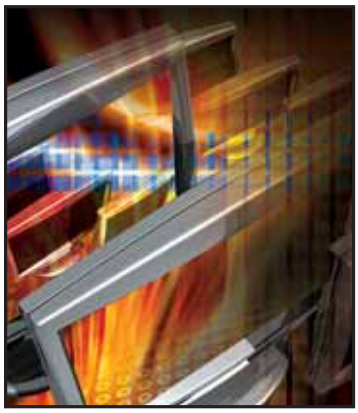
STORAGE & SERVERS

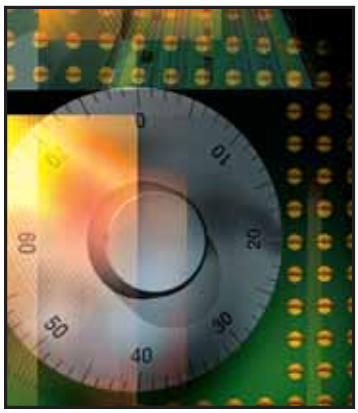


Accelerating Scientific Applications Using FPGAs
A physicist uses C-to-hardware tools with the Cray XD1 supercomputer and Xilinx FPGAs to prototype high-performance algorithms.

SECOND QUARTER 2006, ISSUE 57







Xcell journal

Programmability: The Answer to Complexity

Xilinx is poised for growth in 2006 and beyond.

by Wim Roelandts
CEO and Chairman of the Board
Xilinx, Inc.

The year 2006 will bring many opportunities for growth throughout the semiconductor industry, particularly in markets and applications where just a few years ago FPGAs were a rare commodity. This growth will be driven by several factors. On the demand side, clearly there is a set of new applications emerging that will allow the industry to grow.

Analysts predict semiconductor industry growth at around 10 percent, and I agree with this prediction. The opportunities will obviously be driven by demand. The first of these potential growth areas lies in wireless communications, driven by the strength of the cell phone industry where demand continues to rise, particularly in foreign countries where new infrastructure equipment will also be required.

Another area of tremendous growth potential lies in wired communications, specifically in fiber to the home. In Japan, fiber-to-the-home subscribers have surpassed new DSL subscribers; this year will bring similar growth to the U.S. as service providers deploy new equipment to support not only more data into the home (2/3x), but higher performance equipment to support the speed at which the data travels.

Of course the digital consumer market will continue to further drive growth in the semiconductor industry, as it did in 2005. As prices continue to drop for high-definition television and flat-panel displays, more and more consumers will replace their current televisions with this new technology. In addition, broadcasters and studios will need to replace and upgrade their transmission equipment to meet the demand.

The move from analog to digital cameras will continue at a rapid pace; consumers will need new equipment to store these higher resolution images, driving growth in the PC market as well.

Poised for Growth

Xilinx in particular is poised for growth this year and beyond. Our most recent December quarter (Q3 FY06) was our most successful in history, with net revenues of $450 million and a record 59 percent share of the FPGA market. With new product sales up by 24 percent sequentially, our success is clearly a result of the market diversification efforts we put in place at the outset of the new millennium. Xilinx pioneered the industry's transformation towards a focus on vertical markets and engaging with customers at a system architecture level. Today, this transformation helps us address our customer's complex design challenges by providing them with innovative, flexible, and compelling solutions that help them achieve their objectives of cost management, time to market, and leadership.

As a result of this focused, customer-driven approach, Xilinx continues to win designs in a myriad of emerging applications and markets, including consumer, automotive, medical imaging, AVB, and wireless communications. Adoption of our newest 90 nm platform FPGAs is a key contributor – Xilinx has generated more than 2.5 times the 90 nm sales of its nearest competitor and shipped more than 10 million 90 nm devices.

Our investment in building integrated sophisticated system-level features into Virtex™ Platform FPGAs and aggressively moving to the next node with each new generation is paying off for designers. Customer demand is on a steady ramp for applications requiring high-speed DSP, integrated IBM PowerPC™ processor cores, and multi-gigabit serial transceivers. Increasingly at the heart of the system, Virtex FPGAs have become the premier platform for a wide range of designs for high-performance applications ranging from wireless base stations, networking and telecom infrastructure, and video and audio broadcast equipment to aerospace and defense, security, and industrial control.

PLD Market Outlook

This year will bring the first 65 nm devices to the PLD market, which represents yet another cost reduction vehicle to enhance our competitive advantage over fixed ASICs. Major changes are taking place in the ASIC marketplace as designers continue the quest for a more flexible and cost-effective solution. Xilinx will continue to drive down costs through advanced manufacturing processes such as 65 nm and 300 mm wafers, in addition to adding increased functionality.

Today's consumer products are increasing in complexity with networking, computing, and imaging capabilities. Each of these capabilities has different standards – and in an environment with the convergence of many competing standards, programmability is king. In other words, as the world becomes more integrated, the world becomes more complex. And the only answer to complexity is programmability.

...as the world becomes more integrated, the world becomes more complex. And the only answer to complexity is programmability.

Scoping the Next Killer App – Triple Play

How consumers are influencing the Xilinx roadmap.

by Krishna Rangasayee
Senior Director, Vertical Markets and Partnerships
Xilinx, Inc.
krishna.rangasayee@xilinx.com

Xilinx began its market diversification efforts in earnest in early 2001, at a time when the communications industry – representing more than 80 percent of our revenues – was clearly declining. We began by establishing a team of experts in six targeted markets: aerospace/defense, automotive, broadcast, consumer, industrial/scientific/medical (ISM), and wired/wireless.

Simultaneously, we sought to elevate the strategic fit of our solutions into multi-generational applications within the communications and storage and server markets. We then focused our efforts on developing products and solutions specifically to meet the demands of these key markets, both from a cost and performance perspective.

This strategy required Xilinx to rethink its product roadmap strategy and establish a more system-level architecture and solutions-based approach to next-generation products. Thus began our transformation from an industry-leading silicon supplier to a market-driven solutions enabler.

Although the majority of semiconductor manufacturers addressed the industry downturn through significant layoffs, Xilinx reorganized its resources and strategy to focus on market diversification efforts. We combined our approach with an aggressive migration down the process technology curve, setting our sights on 90 nm. Historically, programmable logic devices (PLDs) were not a cost-effective solution for high-volume, price-sensitive applications. To compete in the larger $44 billion ASIC and ASSP market, we understood the need to reduce the price of our product offerings.

This revolutionary approach to the infamous market decline was extremely successful. Today, more than 45 percent of Xilinx® revenues can be attributed to these vertical market segments, and a one million gate device, formerly priced in the thousands, can now be purchased for as little as $8 (based on a 500,000 unit volume, second half 2006, -4 speed grade, commercial temperature, lowest cost package option). In addition, revenues for our high-volume Spartan™ FPGA family grew from less than 2 percent to more than 24 percent of Xilinx total revenue in just four years, with the majority of that revenue realized in targeted diversified markets and applications.

Going Forward: The Triple-Play Opportunity

The triple-play opportunity has a huge role in our strategy to further penetrate these key vertical markets. What exactly is triple play? Triple-play service is a marketing term for the provisioning of three services: voice, video, and data, over a single broadband (IP) connection. The Internet has emerged as a key infrastructure for service innovation, enabling IP to become the wide area network communication it is today. Over the next decade, as consumer demand increases for triple-play services, network planners must engineer broadband aggregation networks to reach new thresholds of scalability, performance, and quality of service (QoS). Experts agree that the aggregation of services over IP will transform the way we work, live, and play.

The expansion of residential services beyond high-speed Internet will require a revitalization of network infrastructure and operations. Aggregation bandwidth requirements are expected to increase 10 to 100-fold, ultimately requiring 20-50 Mbps per channel to deliver these new services.

This network infrastructure overhaul creates an amazing opportunity for the semiconductor industry across a wide range of vertical markets, including

wired/wireless communications, consumer, audio/video/broadcast, storage and servers, and ISM. Each of these markets will need to develop new products and services to satisfy consumers' hunger for more bandwidth.

The Xilinx Value Proposition

Consumers face a vast array of choices to satisfy their need for communications, entertainment, and information; loyalty to any one service provider is dwindling. In an effort to expand revenue sources and enhance customer loyalty, telecommunication providers are juggling new regulations and enabling technologies to compete with cable service providers.

There is fierce competition as each strives to deliver superior services through product differentiation. The architecture must be optimized for sustained high-bandwidth demand and must be cost-effective at both ends of the scale. The network must also allow operators to incrementally introduce new features. Time to market, design flexibility, and reprogrammability are essential. Because PLDs offer all of these benefits, Xilinx is poised to capture much of this vast market opportunity (Figure 1).

According to Bob Wheeler, a senior analyst for The Linley Group, "The accelerating trend towards convergence is making packet processing and the ability to ensure IP QoS an increasingly mainstream requirement in communications infrastructure equipment, ranging from broadband access to metro core and wireless. We anticipate the market for packet-processing silicon to grow significantly over the next few years as IP continues to displace ATM. By providing a high-performance, scalable, and flexible traffic management solution, Xilinx is well positioned to address the increased demand for triple-play services."

A number of technical challenges exist with regards to the rollout of triple-play services. Voice, video, and high-speed data all have different characteristics and place different burdens on the network. Voice services are greatly affected by jitter, whereas packet loss or packet reordering has a greater affect on video and data services.

Using a shared resource such as cable or DSL requires the network to employ QoS mechanisms. In response to this demand, Xilinx recently announced a solution to enable QoS support and efficient bandwidth utilization in 10G metro, broadband access, and wireless systems. The Virtex™-4 FPGA-based solution, jointly developed with Modelware Inc., provides product developers with a high-performance, scalable, and flexible approach to integrating advanced traffic management features into networking infrastructure equipment. By using the traffic manager solution, equipment vendors can quickly deliver cost-optimized products that ensure QoS and efficient bandwidth sharing when sending voice, video, and data simultaneously over a converged IP network.

Key benefits of the hardware-based traffic manager include high-performance throughput support from 50 Mbps to 10+ Gbps, allowing the solution to be used in high-end equipment such as metro core routers and in wired and wireless access equipment such as PON OLTs and WiMAX base stations. Scalability and flexibility are provided through support for 2 to 1.2 million queues (256K queues/level maximum), 1 to 5 levels of scheduling, and 1 to 64,000 multicast groups.

The solution also supports a diverse set of algorithms for policing, shaping, scheduling, and congestion avoidance and control. To achieve high integration and cost optimization, the traffic manager can be combined with custom logic, embedded IBM PowerPC™ processors, and multi-gigabit transceivers to provide a total custom solution. The traffic manager supports multiple memory configurations comprising embedded block RAM and a variety of external SRAM/DRAM memories. This allows for the most optimized hardware implementation in terms of cost and performance.

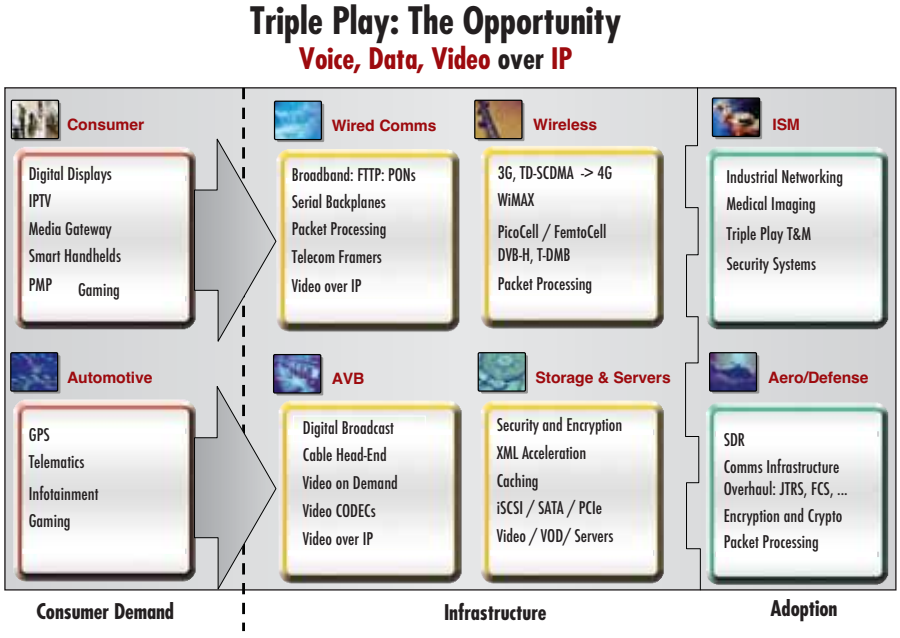


Figure 1 – Triple play in the context of key vertical markets

A Market-Driven Strategy

To take full advantage of the triple-play opportunity, Xilinx will continue to take a market-driven solutions approach to its strategy, solutions, and product portfolio as we strive to gain a thorough understanding of the specific needs of each target market.

PLDs have become a ubiquitous technology, one that has been key in catalyzing the rapid growth of a wide range of new markets and applications. Evolving from an off-the-wall technology to one that is pervasive and mainstream, PLDs today are enabling a wide spectrum of exciting end products in all vertical markets. We hope to help our customers achieve exciting technological breakthroughs and create bold new applications leveraging innovations from Xilinx, through both our silicon and our solutions.

WiMAX — Delivering on the Promise of Wireless Broadband

Using Xilinx FPGAs in the MAC subsystem provides a key differentiator in WiMAX product design.

by Amit Dhir
Director, Infrastructure Vertical Markets
Xilinx, Inc.
amit.dhir@xilinx.com

Much of the current excitement surrounding WiMAX is driven by both technological and commercial reasons. The industry is facing tremendous pressure to reduce cellular network costs, which is possible by achieving greater efficiency of the wireless spectrum than is currently feasible with today's mobile networks. This efficiency can be achieved by using a mixture of advanced technologies that allow very quick changes to user traffic modulation and coding schemes, in conjunction with advanced error correction and traffic scheduling schemes.

WiMAX builds on a network equipment design philosophy, similar to Ethernet and WLAN specifications, that has significantly reduced costs and enables increased efficiency within the enterprise environment. Its challenge is to prove that such benefits can scale and can be applied to wider public networks.

A Closer Look at the Specification

The IEEE 802.16 standard specifies a system comprising two core components: the subscriber station (SS) or customer premises equipment (CPE), and the base station (BS). A BS and one or more SS can form a cell with a point-to-multipoint (P2MP) structure. On air, the BS controls activity within the cell, including access to the medium by any SS, allocations to achieve QoS, and admission to the network based on network security mechanisms.

Multiple BSs can be configured to form a cellular wireless network. When using OFDM, the cell radius can reach as far as 30 miles; however, this requires a favorable channel environment and only the lowest data rate is achievable. Practical cell sizes usually have a small radius of around 5 miles or less. Note that the WiMAX standard also can be used in a point-to-point (P2P) or mesh topology, using pairs of directional antennas to increase the effective range of the system relative to that possible with P2MP.

The 802.16 MAC protocol is designed specifically for the P2MP wireless access environment. It supports transport protocols such as ATM, Ethernet, and Internet Protocol (IP), and can accommodate future developments using the specific convergence layer (Figure 1). The MAC also accommodates very high data throughput through the physical layer while delivering ATM-compatible quality of service (QoS), such as UGS, rtPS, nrtPS, and Best Effort (BE) (Figure 2).

The 802.16 frame structure enables terminals to be dynamically assigned uplink and downlink burst profiles according to the link conditions. This allows for a trade-off to occur – in real time – between capacity and robustness. It also provides, on average, a 2x increase in capacity when compared to non-adaptive systems.

The 802.16 MAC uses a variable-length protocol data unit (PDU) and other innovative concepts to greatly increase efficiency. Multiple MAC PDUs, for example, may be concatenated into a single burst to save PHY overhead. Multiple service data units (SDUs) may also be concatenated into a single MAC PDU, saving on MAC header overhead. Fragmentation allows very large SDUs to be sent across frame boundaries to guarantee the QoS. Payload header suppression can be used to reduce the overhead caused by the redundancy within the SDU headers.

The 802.16 MAC uses a self-correcting bandwidth request/grant scheme that eliminates any delay in acknowledgements, while allowing better QoS handling than traditional acknowledgement schemes. Depending on the QoS and traffic parameters of their services, terminals have a variety of options available to them for requesting bandwidth.

The 802.16 privacy layer follows the model adopted by DOCSIS. The data encryption standard (DES), in cipher block chaining (CBC) mode, is used to encrypt payload for transport and secondary man-

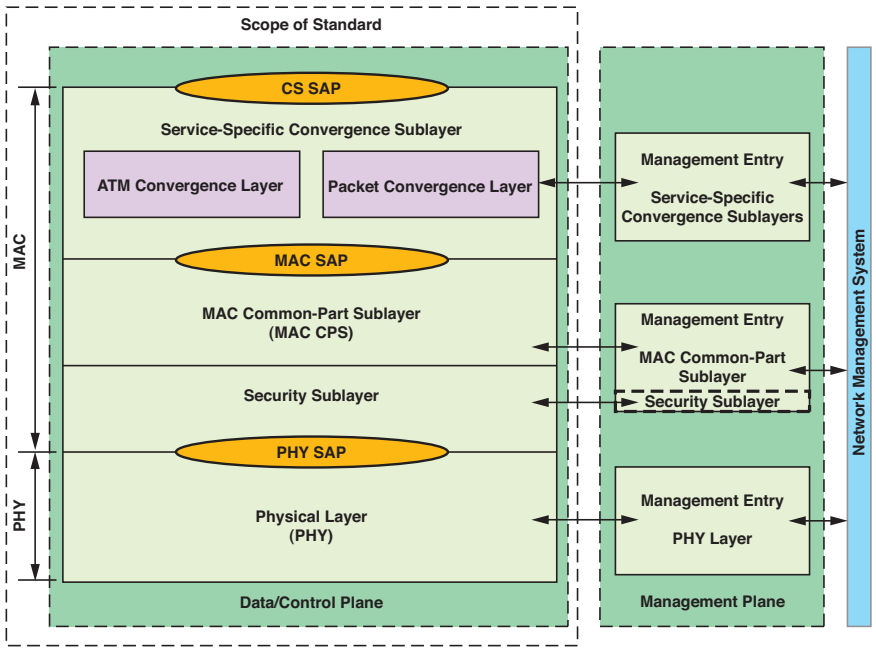


Figure 1 – The 802.16 protocol stack

Service	Definition	Applications	Mandatory QoS Parameters
UGS	Real-time data streams comprising fixed-size data packets issued at periodic intervals	T1/E1, VoIP without Silence Suppression	• Maximum Sustained Traffic Rate = Minimum Reserved Traffic Rate • Maximum Latency • Tolerant Jitter • Request/Transmission Policy
ertPS	Real-time service flows that generate variable-sized data packets on a periodic basis	VoIP with Silence Suppression	• Maximum Sustained Traffic Rate • Minimum Reserved Traffic Rate • Maximum Latency • Request/Transmission Policy
rtPS	Real-time data streams comprising variable-sized data packets that are issued at periodic intervals	MPEG Video	• Minimum Reserved Traffic Rate • Maximum Sustained Traffic Rate • Maximum Latency • Traffic Priority • Request/Transmission Policy
nrtPS	Delay-tolerant data streams comprising variable-sized data packets for which minimum data rate is required	FTP	• Minimum Reserved Traffic Rate • Maximum Sustained Traffic Rate • Traffic Priority • Request/Transmission Policy
BE	Data streams for which no minimum service level is required and therefore may be handled on a space-available basis	HTTP	• Maximum Sustained Traffic Rate • Traffic Priority • Request/Transmission Policy

Figure 2 – QoS classes of service

agement connections. The personal knowledge management (PKM) protocol provides certificate-based authorization of the SS and performs the transfer of keys between the BS and the SS using RSA public key methods and x.509 certificates.

The network entry of any device involves a set of tasks to authenticate and synchronize the BS and SS. Once the BS downlink signal is synchronized, the uplink channel descriptor (UCD) is used to get the timing parameters and the initial ranging contention slot. During the SS ranging process, the BS allocates various management messages for negotiated capabilities followed by registration. Using PKM, a secured secondary management connection is established for authorization. The system is now ready to operate through user connections using various IP protocols, by deploying various MAC services. Regular ranging and channel condition monitoring manages the channel resources.

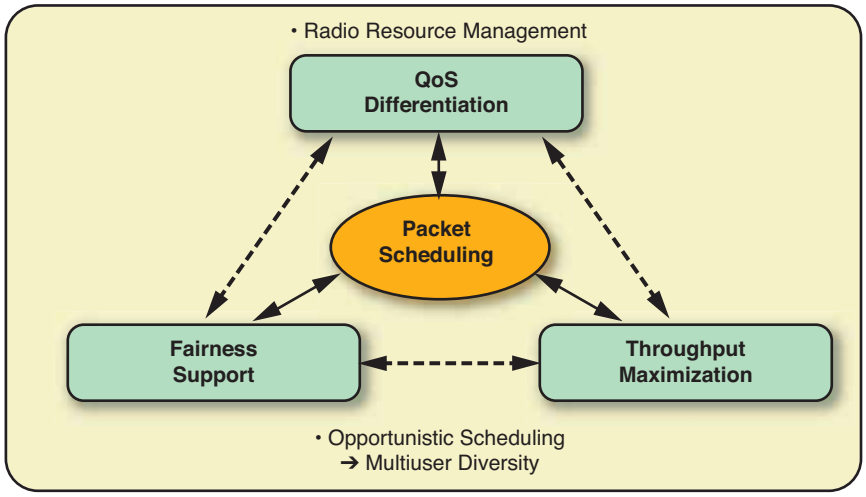


Figure 3 – Packet scheduling, as specified by 802.16, enables a maximization of resources.

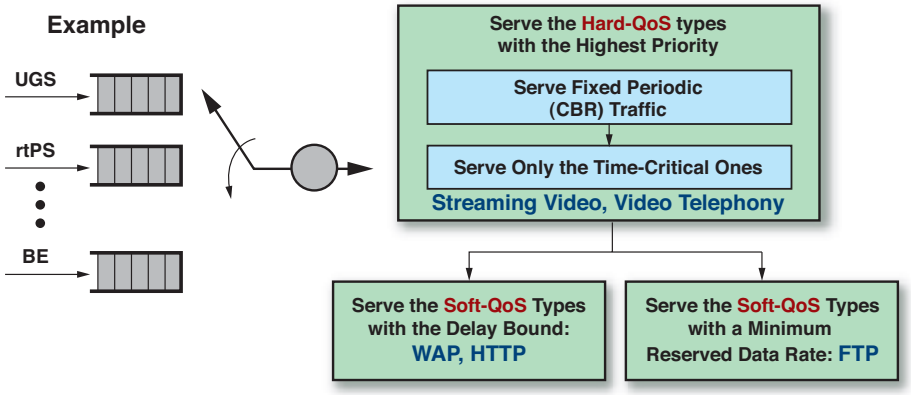


Figure 4 – The functionality of the packet scheduler entails scheduling for different classes of users.

QoS and Scheduling

A high level of QoS and scheduling support is one of the interesting features of the WiMAX standard. These service-provider features are especially valuable because of their ability to maximize air-link utilization and system throughput, as well as ensuring that service-level agreements (SLAs) are met (Figure 3).

The infrastructure to support various classes of services comes from the MAC implementation. QoS is enabled by the bandwidth request and grant mechanism between various subscriber stations and base stations. Primarily there are four buckets for the QoS (UGS, rtPS, nrtPS, and BE) to provide the service-class classification for video, audio, and data services, as they all require various levels of QoS requirements. The packet scheduler provides scheduling for different classes of services for a single user. This would mean meeting SLA requirements at the user level. Users can be classified into various priority levels, such as standard and premium (Figure 4).

Managing Mobility for IEEE 802.16-2005

Handover at high speed, and with low packet loss, is a key requirement for IEEE 802.16-2005 systems. Although much of the handover can be done at the software layer, performance requirements push a part of this functionality into the hardware level.

To identify a neighboring network and define its characteristics in relation to an associated mobile service station (MSS), base stations periodically broadcast a neighbor advertisement management message. Each MSS scans the neighbor BS and measures the signal strength. It then selects the neighbor BS and prepares for the future handover by performing ranging and association procedures.

The performance of a handover is determined by the authentication speed of a user moving from one cell to another cell. Differentiation comes from fast adaptation to changing QoS needs for mobility. The MSS can be stationary or mobile, and the QoS profile can change. The SLA may even employ a different policy for mobile and stationary use.

Sub-Channel Allocation for MAC/PHY Interaction

As the number of subscribers increase, multipath fading and path loss become more significant. OFDMA, based on OFDM, is immune to inter-symbol interference and frequency-selective fading. Achieving high transmission rates depends on the ability of the broadband wireless access (BWA) system to provide efficient and flexible resource allocation. Significant performance gains can be

Although a great deal of debate continues to swirl around the WiMAX standard and its merits in comparison to incumbent cellular standards, current trends suggest that WiMAX is an ideal platform for the delivery of voice, data, and video over wireless.

obtained if frequency hopping and adaptive modulation techniques are used in sub-carrier allocation, with channel gain in the transmitter. The optimal solution is to schedule the user with the best channel at each time. Of course, this procedure is not as simple as it sounds, because the best sub-carrier for the user may also be the best sub-carrier for another user, who has no other good sub-carrier options. The QoS requirement is defined as achieving a specified data transmission rate and bit error rate (BER) for each user in each transmission.

Scalable OFDMA

The concept of scalability was introduced as part of the OFDMA physical layer mode of the IEEE 802.16 wireless MAN standard. A scalable PHY allows standards-based solutions to deliver optimal performance in channel bandwidths ranging from 1.25 to 20 MHz, with fixed sub-carrier spacing for both fixed and portable/mobile usage models while keeping product costs low. A sub-channel structure, with variable FFT sizes per channel bandwidth, enables scalability.

The WiMAX specification also supports advanced modulation and coding (AMC) sub-channels, hybrid automatic repeat request (HARQ), high-efficiency uplink sub-channel structures, multiple-input-multiple-output (MIMO), coverage-enhancing safety channels, different sub-carrier allocations, and diversity schemes. The WiMAX MAC supports scalability using feedback from CQI and HARQ requests.

Key Implementation Challenges

Perhaps the most crucial challenge when implementing a WiMAX system is defining what should be done in the processor and what should be done in hardware, or more specifically, the FPGA-based accelerator. Key to establishing this hardware/software partition is finding the optimal trade-off between system performance and processing requirements, as compared to time-to-market needs. The way in which this trade-off is resolved will result in a variety of different CPE and base station implementations.

WiMAX CPE designs need to adopt a processor-based approach, with lower MAC functions like CRC and encryption/decryption implemented using a hardware accelerator. Base-station implementation, on the other hand, would need to evolve from a lower MAC accelerator to a slow path/fast path approach to packet processing. To achieve this in hardware, the base-station MAC implements slower management and control functions using a processor (either embedded or external to the FPGA), while implementing faster data path functions using the FPGA logic fabric for hardware acceleration.

The key to MAC implementation is to provide the support for handling the "triple play" of voice, data, and video using appropriate queuing and scheduling. Although the IEEE 802.16 standard provides an effective baseline for such functionality, differentiating between competitive solutions comes from how this feature is specifically implemented. Xilinx® FPGAs enable manufacturers to target this critical area of system design and provide a flexible platform on which new MAC functions can be targeted and developed.

The latest Virtex™-4 FX Platform FPGAs include a low-latency auxiliary processor interface (APU) that enables you to simplify the hardware/software partition by allowing customized instructions to be put into software code, which when executed are implemented in the logic fabric.

Advanced DSP functions such as high-performance FEC provide another area in which Xilinx Platform FPGAs can enable product differentiation and advanced functionality. With the release of an optimized Turbo Convolutional codec and its inclusion in a low-cost WiMAX FEC pack, Xilinx has enabled system designers to gain access to highly efficient FPGA cores that can rapidly be deployed as part of a flexible FEC solution in a WiMAX baseband design.

Of course, there are many other areas of a WiMAX base station design – beyond the scope of this article – in which Xilinx technology can add real value. Xilinx devices are now routinely used to implement advanced DSP correction algorithms in the RF card, in areas such as crest factor reduction (CFR), digital predistortion (DPD), and digital up/down conversion (DUC/DDC). By correcting the characteristics of a power amplifier (PA) digitally, you can realize large cost savings by utilizing more cost-effective analog RF circuitry, thereby significantly reducing the cost of the overall base station. For more information about this, visit *www.xilinx.com/esp/wireless*.

Conclusion

Although a great deal of debate continues to swirl around the WiMAX standard and its merits in comparison to incumbent cellular standards, current trends suggest that WiMAX is an ideal platform for the delivery of voice, data, and video over wireless. Xilinx has been involved in the early stages of the WiMAX standardization process. Our products' unique feature set enables our customers to get their WiMAX designs out into the market before the competition.

Xilinx has been working with a number of customers on accelerated traffic management solutions in wireless MAC applications. Each implementation can be customized depending on the application. If you have an opportunity, or wish to find out more, email *espteam@xilinx.com*.

FPGAs Enable Pseudo Wire Emulation Edge-to-Edge

Network convergence is possible using PWE3 for transport and delivery.

by Anthony Dalleggio
VP, Marketing
Modelware
dalleggio@modelware.com

Pseudo Wire Emulation Edge-to-Edge (PWE3) provides methods to carry networking services such as ATM, Ethernet, and TDM over a packet switched network (PSN). It has been gaining in popularity because of compelling economic and operational factors.

Let's say that you are a network operator who needs to add capacity for ATM or E1/T1 services. Should you invest in growing those networks or carry those services over your IP network, where you may have spare capacity? In most cases, the choice is clear – and the answer is PWE3.

PWE3 can consolidate the edge-to-edge delivery of multiple service offerings and traffic types such as E1/T1, ATM, and PPP/HDLC onto a single PSN. This is, of course, not without technical challenges such as accurate clock recovery.

In this article, I'll outline the networking problems that PWE3 solves, how PWE3 works, and describe a PWE3 implementation on Xilinx® Virtex™ and Spartan™ parts.

Reference Model and Services

The Internet Engineering Task Force (IETF) has done extensive work on PWE3. The IETF PWE3 reference model in Figure 1 shows two customers connected through an attachment circuit (AC) to their respective service providers. The native service exists between the customer edges (CE1, CE2) and their provider edges (PE1, PE2). The PW (pseudo wire) spans the PE-to-PE segment and carries the emulated service over the PSN.

PWE3, as any networking layer, must support data plane, control plane, and management functions.

Table 1 lists general and specific requirements for the PWE3 data plane. Some requirements are optional or not applicable depending on the native service being emulated.

Control Plane and Management Considerations

In addition to the data plane requirements, PWE3 must provide the following maintenance requirements:

- Setup and teardown of a PW
- Support and handling of CE-to-CE in-band and out-of-band maintenance messages (such as OAM)
- Support of PE-initiated maintenance messages

Management information bases (MIBs) are required to support provisioning, per-

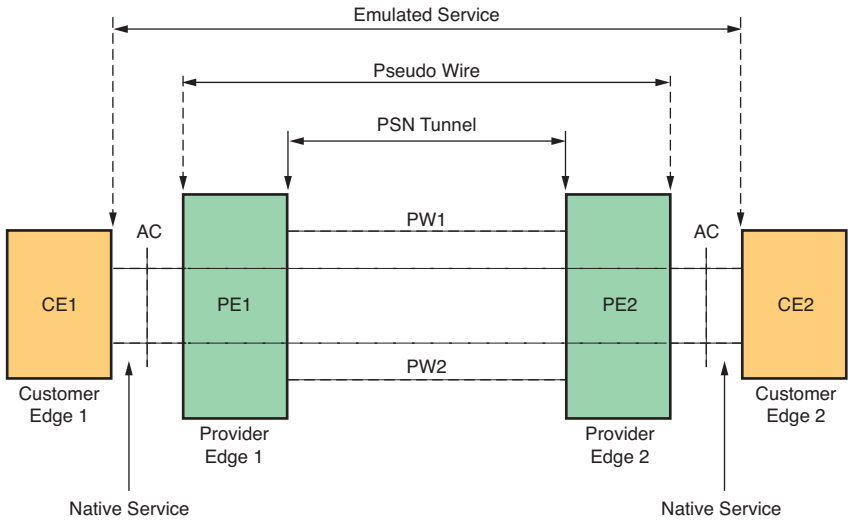


Figure 1 – PWE3 reference model (Source: RFC 3985)

General Requirement	Specific Requirement	Description
Encapsulation	L2 Header Information	To convey L2 header information to allow correct processing of the packet at the PW egress
	Variable-Length PDUs	To support variable-length PDUs if they are allowed by the native service
	Multiplexing/Demultiplexing	To support "trunking" mechanisms like multiple ATM VCCs in one VPC
	Validation of PW-PDU	To support payload validation, the L2 checksum is either passed through the PW or removed at the ingress and recalculated at the egress if the L2 header is modified at the PW entry and exit points
	Payload Type Information	To allow the differentiation between PWE3 and other traffic (such as IPv4) to reduce the probability of packet misordering by load-balancing mechanisms
Frame Ordering		To ensure in-order packet delivery for services that require it; this can be accomplished using a sequence number in the PW header
Frame Duplication		To ensure that duplicated frames are not delivered for services that do not allow it
Fragmentation		To fragment packets whose length, including PW and PSN headers, exceeds the maximum transmission unit (MTU) of the PSN; long packets may optionally be dropped
Concatenation		To concatenate short PDUs into a single PWE3 packet to increase efficiency over the PSN, considering the resulting delay and jitter
Quality of Service		To ensure low delay and jitter characteristics for services that require them
Timing		To generate a service clock at the egress of the PW with a frequency that is close or equal to the ingress service clock

Table 1 – PWE3 requirements

formance monitoring, fault management, and connection verification.

PWE3 Protocol Mapping

Depending on your target network, you will want to specify one of several forms of packet encapsulation. For example, you can carry PWE3 traffic over an IP or an MPLS network. A "light" version also exists for point-to-point solutions.

PWE3 over IP

Figure 2 shows how to map PWE3 traffic onto an IP-based PSN. The payload of the native service is normally passed as is without processing. You can rely on RTP for timing and sequencing; otherwise, a sequence number is added to the PW encapsulation. For PW demultiplexing, you can use the MPLS label, L2TP session ID, or the UDP port number.

PWE3 over MPLS

Figure 3 shows the PWE3 to MPLS mapping. The MPLS mapping is more efficient because you can compress some of the PWE3 information into one control word. For PW demultiplexing, you can use an inner MPLS label. The sequence number is carried in a control word.

PWE3 "Light"

Certain point-to-point or private network applications do not require all of the PSN overhead. For these applications, you can carry the native service directly over Ethernet with minimal PW header information to identify any subchannels and carry timing information if required.

Pseudo Wire Timing Recovery

Not all native services require timing recovery. In general, non real-time services (such as data) do not need timing recovery at the destination. However, real-time services such as TDM do require timing recovery. There are three basic approaches to timing recovery: absolute, differential, and adaptive. Regardless of the approach, you can generate the clock using analog or digital techniques.

Absolute Mode

This is the normal method used for real-time protocol (RTP). The sending end generates a time stamp that corresponds to the sampling time of the first word in the packet payload. The receiving end uses this information to sequence the messages correctly but without knowledge of the sending end's clock frequency. This method is used when absolute frequency accuracy is not required.

Differential Mode

In the differential mode, both sending and receiving ends have access to the same high-quality reference clock. The sending end generates time stamps relative to the reference clock. The receiving end uses the time stamps to generate a service clock that matches the frequency relationship of the sending end's service clock to the reference clock. This method produces the highest quality clock and is affected least by network quality of service issues.

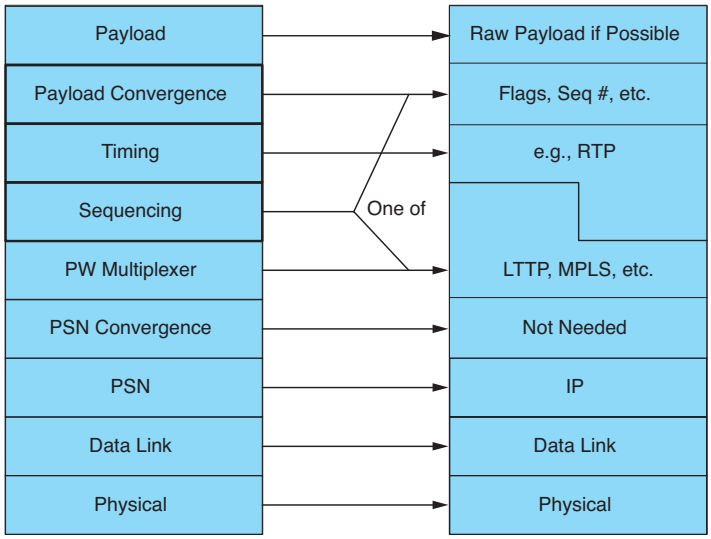


Figure 2 – PWE3 over IP (Source: RFC 3985)

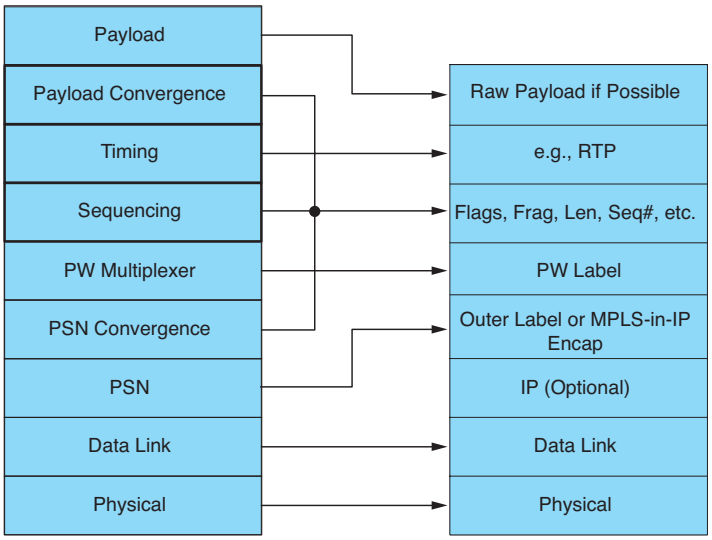


Figure 3 – PWE3 over MPLS (Source: RFC 3985)

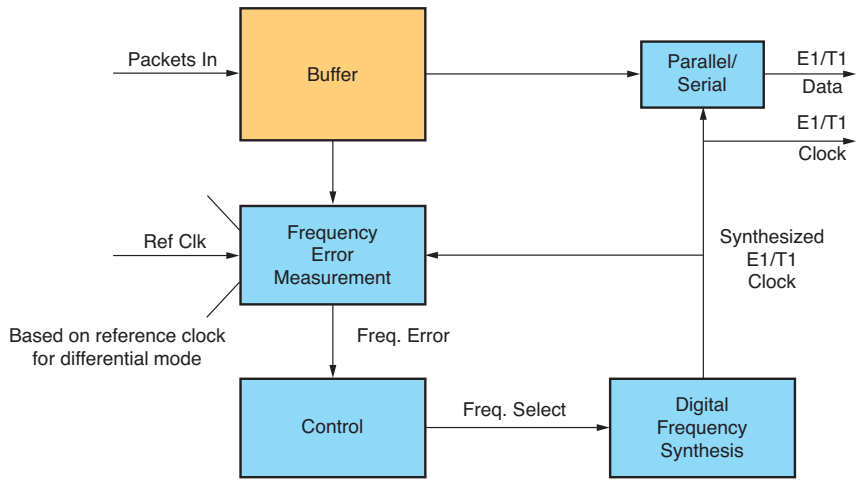


Figure 4 – E1/T1 clock generation

Adaptive Mode

The adaptive clock recovery mode relies on packet inter-arrival time to generate the service clock frequency. This method does not require time stamps or a reference clock to be present at the receiving end. However, it is affected by packet inter-arrival jitter.

Clock Generation

Modelware's PWE3 clock generator implements the adaptive and differential mode timing recovery for E1/T1, using digital clock synthesis, in Xilinx Spartan-3 and Virtex-4 FPGAs. A four-channel differential clock generator uses 679 slices, 8 block RAMs, and 1 global clock buffer.

Low frequencies such as E1 (2.048 MHz) and T1 (1.544 MHz) can be generated completely digitally using a high-speed clock while meeting jitter and wander requirements. A typical clock generation circuit is shown in Figure 4.

Note that the circuit behaves like a phase lock loop (PLL), where the frequency error is based on the buffer fill level for the adaptive mode and on the reference clock for the differential mode.

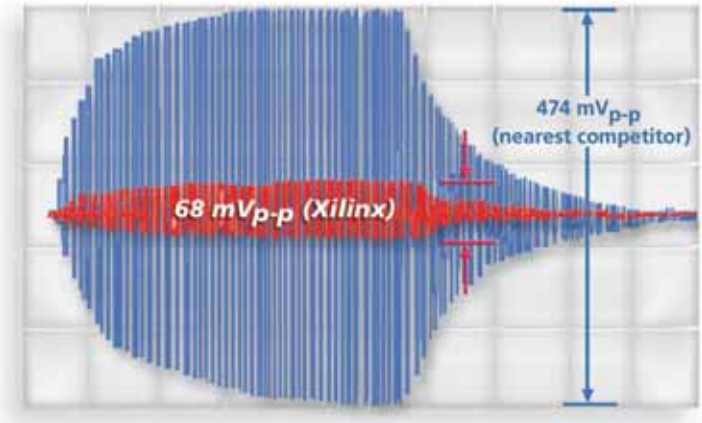
At Modelware, we have implemented adaptive- and differential-mode timing recovery for E1/T1 using digital clock synthesis. The advantages of our solution are the linearity of the digitally controlled oscillator (DCO) and the ability to configure the frequency resolution to very small values.

Conclusion

In this article, I have discussed the motivation, architecture, and services provided by PWE3. Because of its attractive cost structure and the promise of network consolidation, PWE3 is gaining momentum. This is true not only in public and private networks, but also as a low-cost approach for point-to-point intra-system connectivity.

Modelware offers PWE3-related IP cores and design services. For more information, please contact Anthony Dalleggio at (732) 936-1808 x222, or e-mail *dalleggio@modelware.com.*

"High Signal Integrity Demands a Low Noise Chip".



Design Example: 1.5 volt LVCMOS 4mA, I/O, 100 aggressors shown.

Best Signal Integrity: 7x Less SSO Noise

Virtex-4 FPGAs deliver the industry's best signal integrity, allowing you to pre-empt board issues at the chip level, for high-speed designs such as memory interfaces. Featuring a unique SparseChevron™ pin out pattern, the Virtex-4 family provides the highest ratio of VCCO/GND pin pairs to user I/O pins available in any FPGA. By strategically positioning one hard power pin and one hard ground pin adjacent to every user I/O on the device, we've reduced signal path inductance and SSO noise to levels far below what you can attain with a virtual ground or soft ground architecture.

The Industry's Highest Signal Integrity, Proven By Industry Experts

Incorporating continuous power and ground planes, plus integrated bypass capacitors, we're eliminating power-supply noise at its source. In addition, we provide on-chip termination resistors to control signal ringing. The lab tests speak for themselves. As measured by signal integrity expert Dr. Howard Johnson, no competing FPGA comes close to achieving the low-noise benchmarks of Virtex-4 devices.

Visit ***www.xilinx.com/virtex4/si*** today, and choose the right high-performance FPGA before things get noisy.

Dr. Howard Johnson, author of *High-Speed Digital Design,* frequently conducts technical workshops for digital engineers at Oxford University and other sites worldwide. Visit www.sigcon.com to register.

FPGA and Structured ASIC JOURNAL Reader's Choice Awards

Highest customer satisfaction:
- Virtex & Spartan FPGAs
- ISE design tools
- Support

XILINX®
The Programmable Logic Company℠
www.xilinx.com/virtex4/si

View The TechOnLine Seminar Today

Breakthrough Performance At The Lowest Cost

Enabling Next-Generation MAN Equipment

You can develop flexible, IEEE 802.17 Resilient Packet Ring-compliant equipment faster and with less resources using the Xilinx RPR MAC reference design.

by Delfin Rodillas
Staff System Architect
Xilinx, Inc.
delfin.rodillas@xilinx.com

Metro area network (MAN) architectures are based primarily on legacy SONET/SDH technology. SONET/SDH is reliable and ideal for carrying time-division multiplexed traffic such as voice, but inefficient for carrying "bursty" packet traffic such as IP and Ethernet. With the growth of packet-based traffic in recent years, this inefficiency has led to a bandwidth bottleneck in the MAN.

Among the options for new architectures are Ethernet over SONET/SDH (EoS), Resilient Packet Ring (RPR), and carrier-grade Ethernet. EoS overlays new technologies onto SONET/SDH to achieve efficiency and service support. Carrier-grade Ethernet defines extensions to the existing suite of Ethernet protocols to achieve similar goals. RPR, a new layer 2 transport technology defined in IEEE 802.17, is perhaps the most interesting of the emerging MAN technologies.

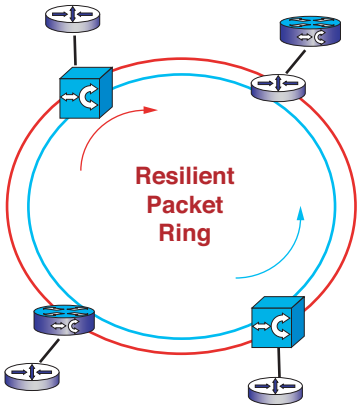


Figure 1 – RPR's dual-ring architecture with concurrent transmission of data on both rings

In this article, I'll review RPR technology and describe how Xilinx® Virtex™-4 FPGAs, along with our newly announced RPR MAC reference design, can be applied towards the development of RPR systems.

RPR Technology and Benefits

There are several key features of RPR that are highly beneficial. RPR makes efficient use of the dual-ring topology found in most MANs by sending traffic concurrently on both rings, as shown in Figure 1. This is unlike SONET/SDH, which sends data on one ring, reserving the other for protection applications only. Furthermore, this technology could not be used in metro area applications unless it provided robustness – so RPR supports a sub-50 ms restoration time, fault tolerance, and manageability.

RPR also defines multiple classes and subclasses of service, which is ideal for supporting the growing number of IP-based services such as Voice over IP (VoIP) and IPTV. For example, real-time VoIP traffic can be mapped to the highest service class in RPR, class A0, which guarantees bandwidth and provides low jitter. Internet traffic can be mapped to best-effort class C traffic.

RPR also defines a fairness algorithm that fairly allocates unused "reclaimable" bandwidth among nodes with fairness-eligible traffic. This enables a high level of oversubscription and is ideal for supporting bursty traffic like Ethernet. Table 1 shows the different classes of service supported by RPR.

The RPR MAC

All RPR nodes implement an RPR MAC, which is the entity that governs all access to the rings. Figure 2 shows the RPR protocol stack and how it maps against the OSI reference model. Functionally, the MAC has the following aspects:

- Data formatting and transport requires the MAC to service client requests for transport by framing the data into the RPR packet format. The MAC also forwards any transit traffic back to the ringlet through a single-transit queue or dual-transit queues.

Class of Service			Qualities of Service				Fairness
Class	Examples of Use	Subclass	Guaranteed BW	Jitter	Typed	Subtype	Eligible
A	Real time	A0	Yes	Low	Allocated	Reserved	No
		A1	Yes	Low	Allocated	Reclaimable	
B	Near real time	B-CIR	Yes	Bounded			
		B-EIR	No	Unbounded	Opportunistic	Reclaimable	Yes
C	Best effort						

Table 1 – RPR's different service classes

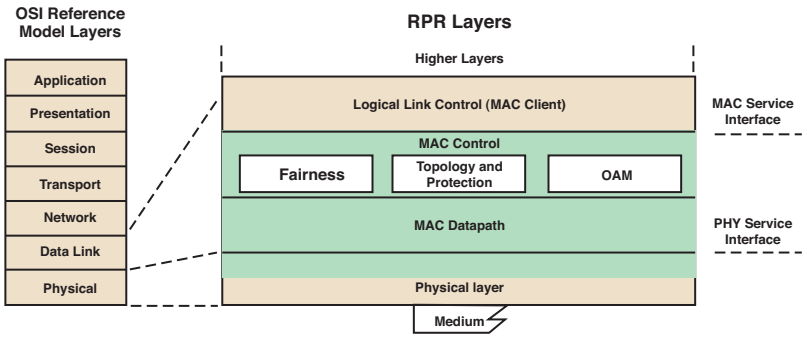


Figure 2 – RPR layers mapped against the OSI reference model

- Network protection requires the MAC to track the topology of the network and respond to any changes through protection mechanisms. As in SONET/SDH networks, RPR limits service interruptions to 50 ms. This can be achieved by either a "steering" or "wrapping" protection method.
- Quality of Service (QoS) and fairness functions require the MAC to monitor the topology of and resource utilization in the network and to perform traffic policing/shaping for the various levels of quality based on available resources. As mentioned previously, RPR defines a fairness algorithm to allocate reclaimable bandwidth among RPR nodes.
- Operations, administration, maintenance, and provisioning (OAMP) requires the MAC to gather statistics for each of the services and report the data back to the user through a standardized interface such as a management information base (MIB). In addition, the MAC is required to provide provisioning of all relevant parameters for each station.

For more details on RPR technology, please refer to the latest IEEE 802.17 specification.

RPR Network Application

The aforementioned features of RPR are key reasons why many service providers and equipment OEMs see RPR as a viable technology for next-generation MAN applications. One particular MAN application of interest is that of Ethernet bridging.

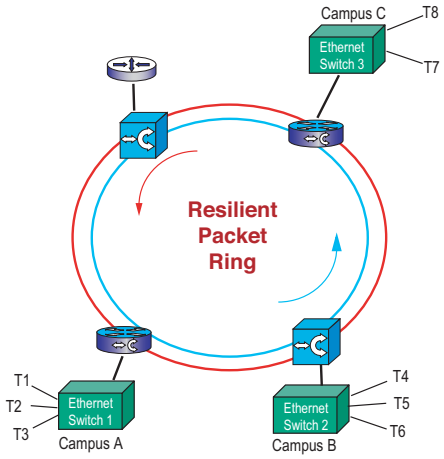


Figure 3 – Ethernet bridging application

In Ethernet bridging, an RPR network is used for interconnecting several LANs across a metropolitan area. The bridging of Ethernet to RPR and vice versa is done per the IEEE 802.1D "MAC Bridges" specification. Figure 3 shows three LANs interconnected over an RPR network.

The benefit of using RPR for this application is that priority and VLAN information is maintained end-to-end. This can be done efficiently while maintaining carrier-class protection (sub-50 ms), useful for an enterprise customer who may need simultaneous VoIP, video conferencing, and Internet services while maintaining 99.999% availability.

Xilinx Solution for RPR

To facilitate the development of RPR equipment, Xilinx developed a complete RPR MAC reference design for use with Virtex-4 FPGAs. The solution is fully compliant to the IEEE 802.17 specification, supporting 1.25 Gbps, 2.5 Gbps, and 10 Gbps rates. Other key features include support for both ringlets in a single core, support for equipment redundancy through a mate interface, and support for single- and dual-transit queues. For fairness, both aggressive and conservative types are supported. To minimize engineering effort and accelerate time to market, driver and application software are also provided. The design is available today, free of charge, under a standard Xilinx licensing agreement.

RPR MAC Functional Blocks

Figure 4 is a hardware functional block diagram of the RPR MAC reference design. It comprises several datapath, topology, and fairness controller blocks. It also includes a PHY, mate, and client interface blocks as well as a memory manager block.

There are three primary controllers in the design. The first is the datapath controller, which manages the flow of frames to and from the RPR MAC for both ringlets 1 and 0. Next is the topology controller, which determines connectivity and the ordering of stations around the ring. Lastly, the fairness controller implements the fairness algorithm per IEEE802.17.

It is worth noting that by implementing the fairness algorithm in XtremeDSP™ slices (or as they are known in software, DSP48s) and pipelining these structures, not only was the design able to work at 10 Gbps, but it was able to do so in the lowest cost -10 speed grade. Furthermore, enormous amounts of logic were saved by implementing the functions in XtremeDSP slices rather than logic.

On the right side of Figure 4 are the PHY interface blocks. There is one PHY interface block for each ringlet. These interfaces are enabled by SelectIO™ and RocketIO™ technologies embedded in the FPGA. The reference design includes PHY interface blocks, which support 2.5 Gbps SONET/SDH through SPI-3, Gigabit Ethernet through GMII interface, and 10 Gigabit Ethernet through XGMII. The design also provides a FIFO-based PHY interface to support any user-specified PHY.

Not included in the design but support-

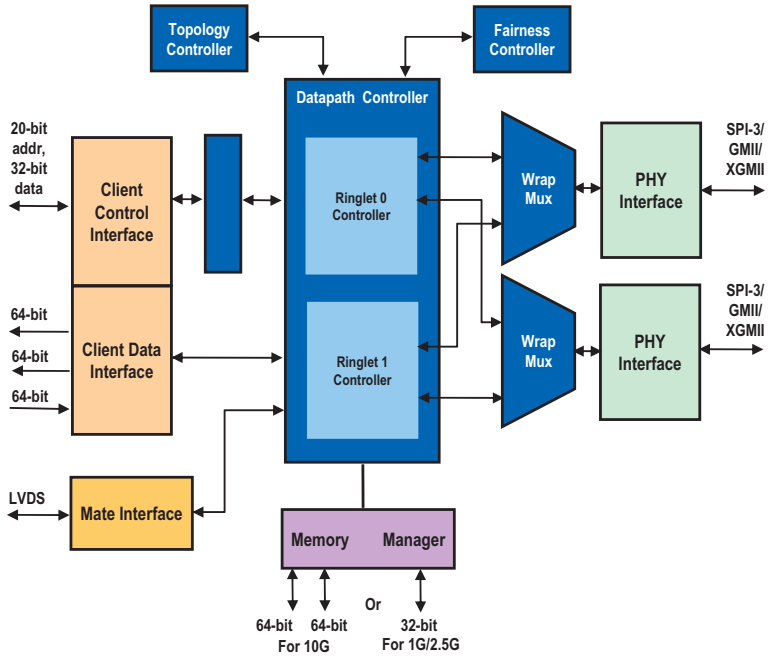


Figure 4 – RPR MAC reference design block diagram

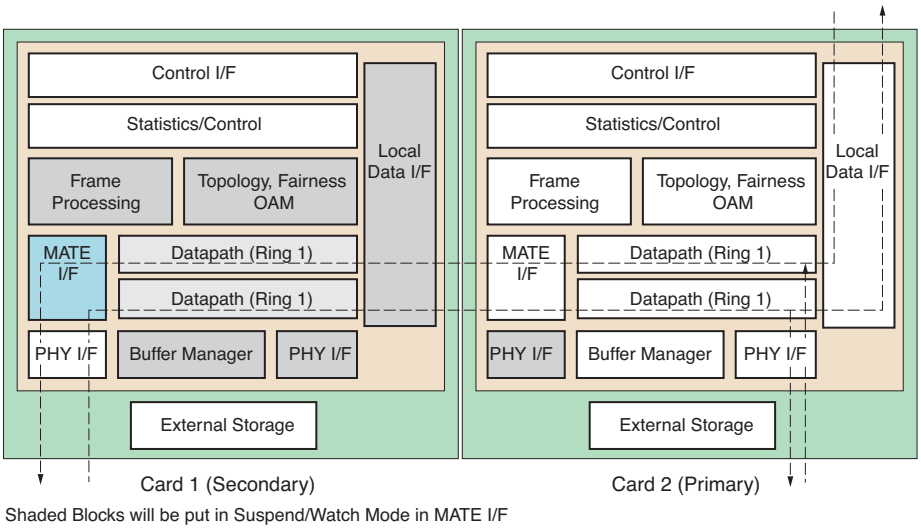


Figure 5 – Mate interface usage in redundancy application

ed by Virtex-4 FPGA features and available IP are interface blocks for TBI and RTBI. For 10 Gbps applications, Virtex-4 also supports SPI-4.2 and XSBI LVDS interfaces as well as SERDES-based XAUI interfaces through embedded RocketIO multi-gigabit transceivers. Xilinx offers IP cores and reference designs for all three of these interfaces to help save development time. Table 2 shows the different PHY interfaces supported by the reference design.

To the left of Figure 4 is the SPI-3-like client interface comprising the client data and client control interfaces. The client data interface is a 64-bit interface operating at 200 MHz in 10 Gbps designs and lower rates for 1 Gbps and 2.5 Gbps designs. There is one interface for TX and two interfaces for RX, each interface serving a ringlet. The client control interface is a 32-bit interface through which the device driver operates.

Interface	1G	2.5G	10G
SONET	N/A	SPI-3	SPI-4.2
Ethernet	GMII, TBI / RTBII	N/A	SPI 4.2, XGMII, XAUI, or XSBI

Table 2 – PHY interfaces supported by the RPR MAC reference design

The mate interface shown towards the bottom left of the MAC block diagram is what enables the creation of equipment supporting redundancy, a critical feature in carrier-class communications equipment. Through the mate interface, a primary card can have a path to a secondary card in the event that the primary card fails. Figure 5 shows how the mate interface could be used to interconnect a primary and secondary card, thus providing equipment redundancy. The reference design implements this interface in LVDS I/O. For 10 Gbps equipment, RocketIO MGTs can be a high-bandwidth alternative to the LVDS I/O.

The memory manager block comprises a buffer manager that performs arbitration and pointer management and a DDR2 SDRAM controller. As you can imagine, memory resources are required for the transit queues. A benefit of the Xilinx solution is that single-transit queue designs are implemented without the use of external memory, because they take advantage of the large block RAMs embedded in the FPGA.

Dual-transit queue designs, which require more memory resources, use external DDR2 SDRAM. The memory interface is 32 bits/200 MHz for 1 Gbps/2.5 Gbps designs. For 10 Gbps designs, two 64-bit/250 Mhz interfaces each support one ringlet.

No reference design would be complete without software, so the design includes a Linux device driver and application software as part of the deliverables. The software can be easily integrated with an SNMP agent to configure/monitor the RPR MAC parameters.

Several processor options are supported in the design. You can use the embedded PowerPC™ in the Virtex-4 FPGA, MicroBlaze™ soft cores, or external processors. By using the integrated processor in the FPGA, you can offload the external host processor and improve overall system performance. Figure 6 shows the partitioning of hardware and software functions within the RPR MAC reference design.

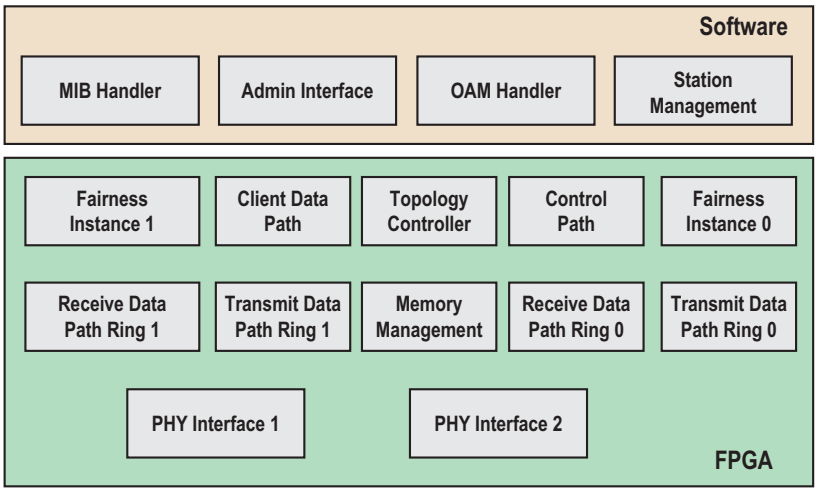


Figure 6 – Hardware/software partitioning in the RPR MAC reference design

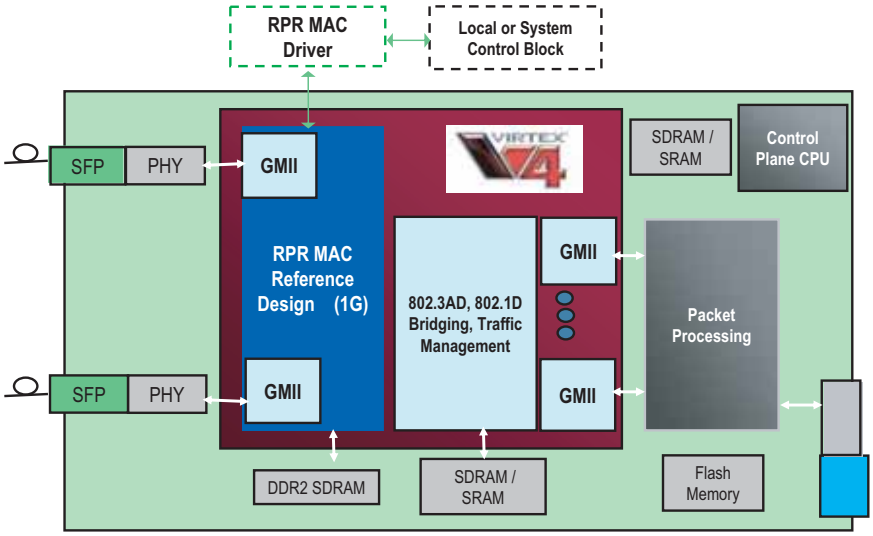


Figure 7 – Ethernet bridging application of RPR MAC reference design

Devices and Deliverables

Table 3 shows the device support and utilization for the RPR MAC reference design. The reference design supports Virtex-4 LX and FX devices and all rates in the lowest cost -10 speed grade. Depending on the options you select, the design utilizes between 15,000 and 16,500 slices and 72 to 90 block RAMs and fits in a Virtex-4 LX40 device (and up). Along with the core, the reference design includes technical documentation, a VHDL test bench, and software.

Device Family	Virtex-4 LX and FX Devices			
Speed Grades	-10 for 1 Gbps, 2.5 Gbps, and 10 Gbps			
Resources Used (exact value depends on configuration)	Slices	Block RAM	DCM	FFs
	15K–16.5K	72–90	5–7	18744

Table 3 – Device support and utilization

RPR MAC Linecard Design Example

I have already presented the RPR application of Ethernet bridging. Figure 7 shows how this linecard could be architected. Instantiated into the Virtex-4 LX FPGA is the RPR MAC reference design. It is connected to the Gigabit Ethernet PHYs through the included GMII interfaces. There are two Gigabit Ethernet PHYs connected to SFP optics, which connect to ringlets 1 and 0. Alternatively, by using a Virtex-4 FX device, you could use embedded RocketIO MGTs instead of external PHYs, allowing further integration.

An appropriate sized device can provide the additional logic needed for implementing custom functions such as 802.1D bridging and possibly 802.3AD aggregation in both the Ethernet-to-RPR and RPR-to-Ethernet directions. Traffic management functions will also most likely be necessary to manage ingress and egress flows. The flexibility and high performance of the FPGA fabric allows any traffic management algorithm to be designed without compromising functionality or throughput. You can also integrate memory controllers for the RPR MAC and any custom blocks into the design. The design includes hooks to allow communication with the driver software. Depending on the functions implemented, such a design may fit in an LX60 or larger Virtex-4 FPGA.

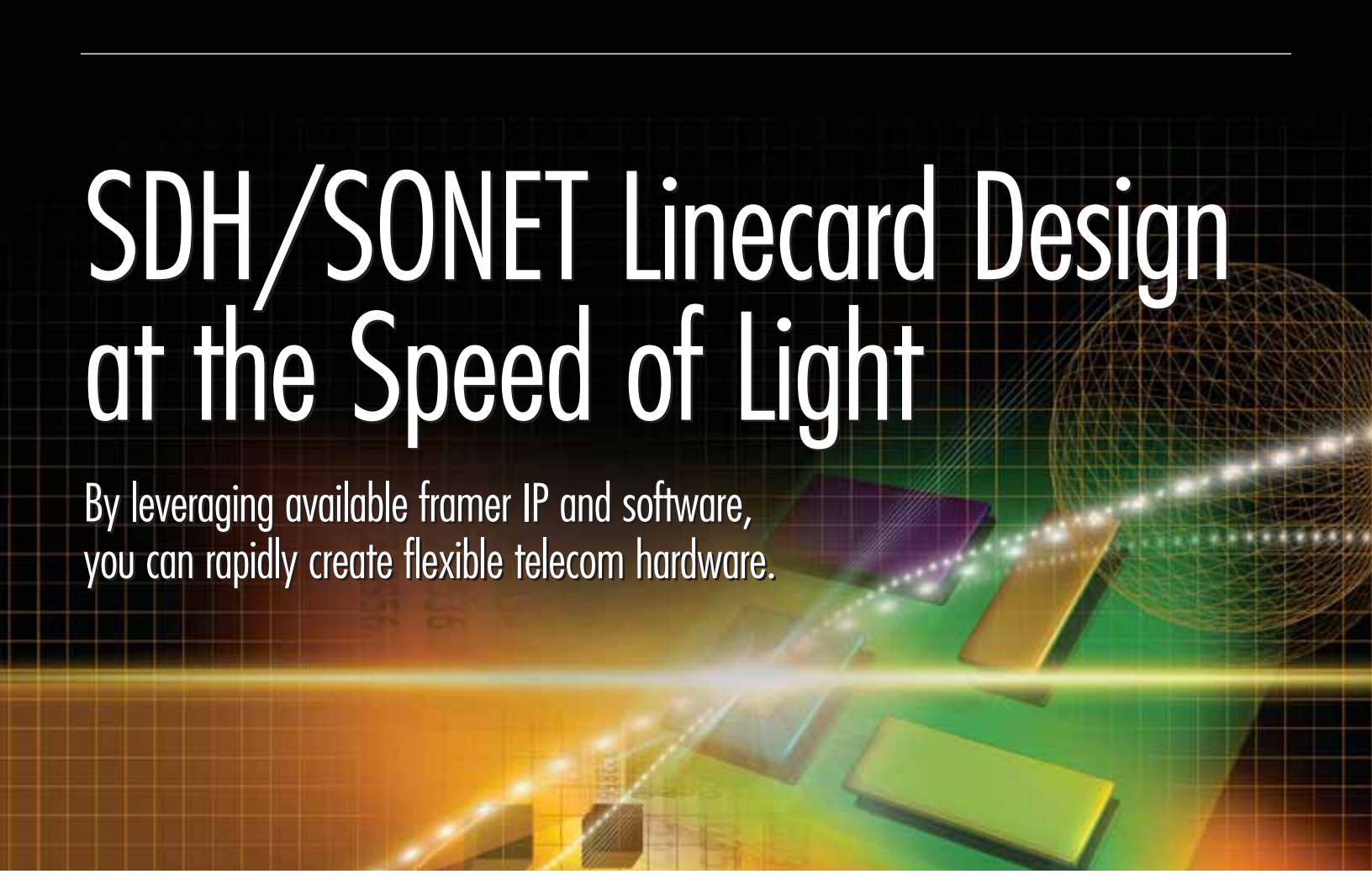
As you can see, using an FPGA allows integration and reduction of system-level costs. Furthermore, the design uses the slowest -10 speed grade even for 10 Gbps designs, as there is a significant performance margin in the FPGA. This helps reduce costs even more.

Conclusion

As triple-play services continue to be deployed, more and more service providers may look towards RPR as their transport technology for next-generation MANs, especially in newer green-field deployments. As these demands increase, equipment OEMs can leverage the Xilinx flexible RPR solution to quickly develop standards-compliant products with unique value-add capabilities.

For more information on our RPR solution, visit the Xilinx RPR webpage at *www.xilinx.com/esp/wired/rpr*, where you will find a white paper, product brief, and link to a page for requesting access to the reference design.

SDH/SONET Linecard Design at the Speed of Light



By leveraging available framer IP and software, you can rapidly create flexible telecom hardware.

by John Brandte
VP Marketing
NComm, Inc.
john@ncomm.com

Steve McDonald
Director
Aliathon Ltd.
steve@aliathon.com

Framers are an inherent part of SDH/SONET system architectures. Many telecom OEMs enhance their own framer designs by combining their proprietary ideas with off-the-shelf IP, creating both a competitive edge and shorter development times. Xilinx® Platform FPGAs coupled with new developments from Aliathon and NComm are further enabling this trend by delivering validated IP in addition to operations, administration, and maintenance (OAM) system management tools.

There are several good reasons for choosing an FPGA for your next SONET framer design. One is that you may need only a subset of features from a very large, complex ASIC or ASSP device to meet the functionality you require. For example, an STM-1/4 or OC-3/12 framer ASIC may be underutilized by the application. Perhaps you need only the transmit section, or just the receive section, or the structure being generated is fixed and the device is not being used to its fullest. With an FPGA, you can have just the functionality you need when you need it, saving board space, cost, and power.

Another reason is that in multi-service communication networks, you may be forced to put several separate framer devices on your card to get the functionality you need. Additionally, many designs require an FPGA connected to a framer. Why not put it all in one FPGA? The result could be a much smaller and lower cost design. With the low static and dynamic power consumption in the Virtex™-4 family, you will probably save power, too. Figure 1 illustrates the potential level of integration possible in today's FPGAs.

Multi-Channel Framers Increase Throughput

Aliathon Ltd., a Xilinx Alliance Program member, can support framer requirements for copper networks or the latest high-speed optical networks through its feature-rich soft IP. Aliathon targets only FPGAs and, as a result, gets the very best from the technology.

Aliathon has made good use of Xilinx logic and memory architectures to handle a very large number of channels simultaneously. For instance, Aliathon IP can process hundreds of T1s packed into a high-speed optical channel (such as a T1 over OC-3/12/48 or E1 over STM-1/4/16) using only a small amount of logic and the fast on-chip memory of Xilinx FPGAs. Table 1 is an example of logic utilization for a E1/T1/J1 multi-channel framer and other cores.

To complete the picture, Virtex-4 technology includes, along with lots of fast logic and memory, multi-gigabit transceivers to interface to your high-speed network directly, as well as the embedded PowerPC™ processor to give a bit more intelligence to your designs. All in all, Virtex-4 devices are the ideal solution for most of your framer needs.

The benefits of using FPGAs are significant. You can implement very complex systems, because you only use the logic you

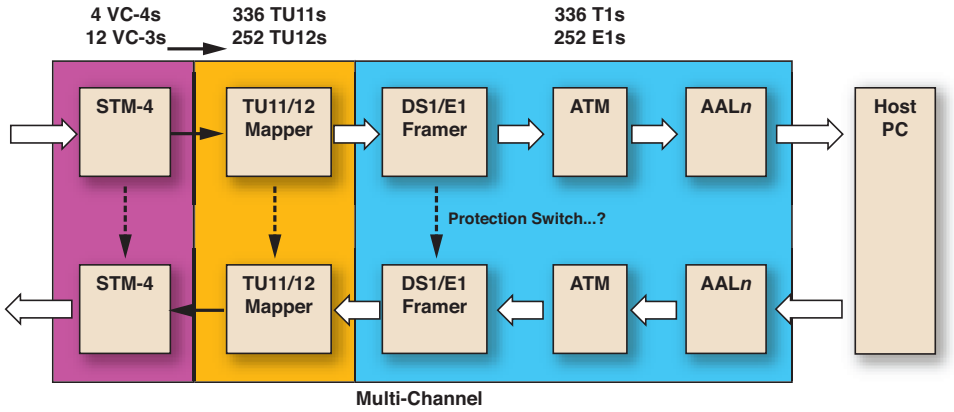


Figure 1 – Virtex-4 FPGAs enable higher design integration.

Function	Slices (Approximate)	Block RAM (Approximate)	Fmax
OC-3/12 Deframer	850	0	>80 MHz
OC-3/12 Framer	700	0	>80 MHz
OC-3/12 Demapper	1,250	5	>80 MHz
OC-3/12 Mapper	1,400	5	>80 MHz
E1/T1 Deframer	800	27	>80 MHz
E1/T1 Framer	400	2	>80 MHz
ATM Deframer	400	4	>80 MHz
ATM Framer	400	4	>80 MHz

Table 1 - Example FPGA resource utilization for cores described in Figure 1

need. You can even target very different architectures by simply loading a new FPGA image. For example, you might use the same card for concatenated or deeply channelized applications. If you don't need both at the same time, you get an even smaller design - something that is impossible with an ASSP/ASIC-based card.

Aliathon has a range of IP covering PDH, SONET/SDH, and OTN networks (see Figures 2, 3, and 4). Its IP is supported on Virtex-4 devices as well as the Spartan™-3 device family.

Drivers and OAM Application Software

When creating a SONET/SDH system with a wide area network (WAN) interface using an FPGA-based Aliathon framer, the end customer (a carrier) often requires OAM functions as part of the system design specification. Xilinx is collaborating with NComm to deliver this more comprehensive OAM solution.

NComm's Trunk Management Software, or TMS, is a completely functional suite of T1/E1, T3/E3, and SONET/SDH software providing configuration, alarm, maintenance/performance monitoring, loopback activation, robbed-bit signaling (T1) and channel-associated signaling (E1) functionality. The TMS sits between the developer's product and the framer driver below. It does not deal with the data payload, but handles the control, monitoring, and reporting required to keep the trunk operating as effectively as possible and in a standards-compliant manner.

OAM functions are typically implemented in software that requires a device driver for the framer, as shown in Figure 5. The effort required to develop the driver and application software in a WAN interface is often underestimated, resulting in missed deadlines, lost profits, unsatisfactory product performance, and cost overruns. By sourcing both software and driver from WAN experts, you can almost always accelerate time to market with the highest quality interface at the lowest cost. Aliathon, NComm, and Xilinx have collaborated to create the necessary NComm device driver module that supports Aliathon's OC-48 framer core for NComm's OAM software.

The framer establishes the network connection, inserts and removes data, and provides the information needed to manage the interface. The software required to control the framer is dependent on the design of the framer hardware.

For example, the definition of an out-of-frame (OOF) condition is defined across all interfaces of a technology - whether T1, SONET, or some other interface - but internal framer mechanics are unique to each device. As long as what enters and emerges complies with the applicable standards, you have the freedom to engineer as you see fit.

The driver enables the software on the processor to communicate with the device's registers. At the SONET/SDH driver API, the commands are the same for all framers supporting SONET/SDH. But beneath the API, the driver communicates with the specific hardware registers - allowing a clear separation between framer-specific software and hardware-independent reusable software.

Above this driver API, required functions are the same regardless of the underlying framer and implement the required OAM functions. OAM software provides non-device-specific processing and routines for the interface. While the data payload is being sent to higher layer applications, the management system performs alarm-management and performance-monitoring (PMON) functions.

The alarm management handles creating alarms from defects. Defects are an immediate indicator of the condition of the lines. When defects like AIS-L/MS-AIS

occur, they are reported up from the framer, and a timer is started. If the condition persists for 2.5 seconds, an AIS alarm is declared. AIS is propagated to the downstream entities and a RDI-L/MS-RDI is sent back to the equipment on the other end.

PMON is proactive, collecting and time-stamping performance reports and other data every second. PMON data is analyzed to detect deteriorating conditions before service interruption, but should a hard failure occur, you can use this information, along with loopbacks, to isolate the source of the problem. You can also set threshold crossing alerts (TCAs) to trigger if something exceeds an expected range.

Standards define a restoration method called automatic protection switching (APS), which enables SONET/SDH traffic to be moved from one fiber to another with minimal disruption to user data in the event of a line failure. Because SONET/SDH supports higher payload capacity, the failures have greater impact, necessitating restoration techniques like APS.

Driver/OAM Solution for Aliathon Framers

NComm's TMS provides drivers and OAM processing packages. Coupled with an Aliathon framer, developers have a full-function, standards-compliant SONET/SDH interface almost immediately.

Unlike many protocol stacks, NComm's TMS does not require additional development to make it work. Within minutes of bringing up the software, you have a complete interface, which can be demonstrated immediately.

Interfacing to the driver API, TMS performs all of the necessary framer, span configuration, alarming, performance monitoring, and line testing OAM functions that carriers and service providers expect. NComm's APS software functions whether the protection line is on the same framer/card or in a separate chassis. The SONET/SDH software is completely data-driven, allowing configuration of operating mode, alarm timers, and thresholds on a static or run-time basis.

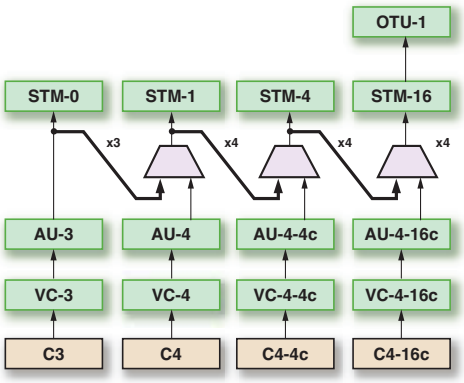


Figure 2 – SONET/SDH framing IP

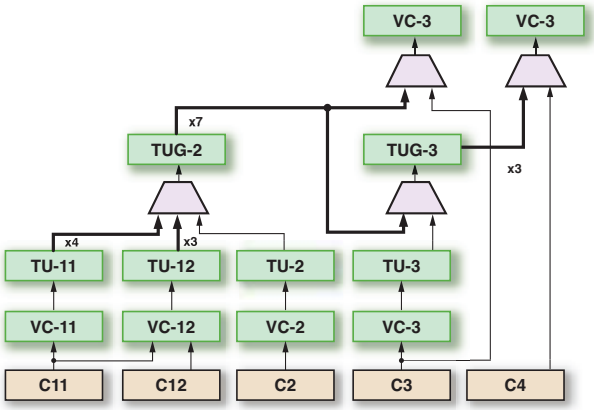


Figure 3 – SONET/SDH mapping IP

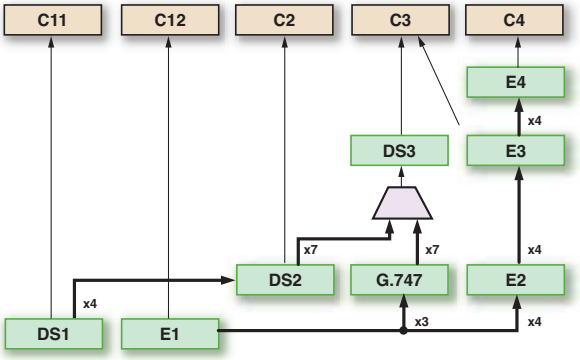


Figure 4 – PDH mapping IP

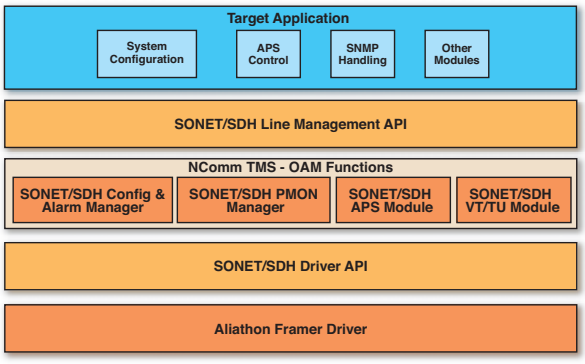


Figure 5 - Driver and OAM software

TMS is designed to work in complex, multi-technology environments where, for example, T1s are being extracted from a SONET interface using VT mapping or an M13 multiplexer. TMS handles these situations whether implemented with discrete framers and mappers or with an integrated framer/mapper.

TMS is operating system- and processor-independent. NComm offers pre-ports to the most popular operating environments: Linux (2.4/2.6), VxWorks, OSE, and Nucleus Plus. Other operating systems can typically be accommodated in a few days.

Whatever your design objectives, there is no faster, more flexible, or economical path to a robust SONET/SDH interface. NComm's TMS software coupled with Aliathon IP is a proven combination with demonstrated results.

Conclusion

SDH/SONET-based communications systems are being used ever more broadly for new applications, like triple-play services (audio, video, and data). So although many have declared that SDH/SONET will be replaced with other transport methods, this has not happened. The hardware and system software design you are creating today must be leveraged for the new developments that will inevitably occur tomorrow.

You can accomplish robust and scalable SONET/SDH framer designs from the pre-validated IP, drivers, and application software from Aliathon and NComm. Let us know if we can help you reach your next framer design goals. For more information, visit *www.aliathon.com* or *www.ncomm.com.*

Honey, will you please tell Alex
to stop programming the FPGA!

Viva! Absurdly Easy.

Viva! Software

The Easiest Development Tool for Programming FPGAs

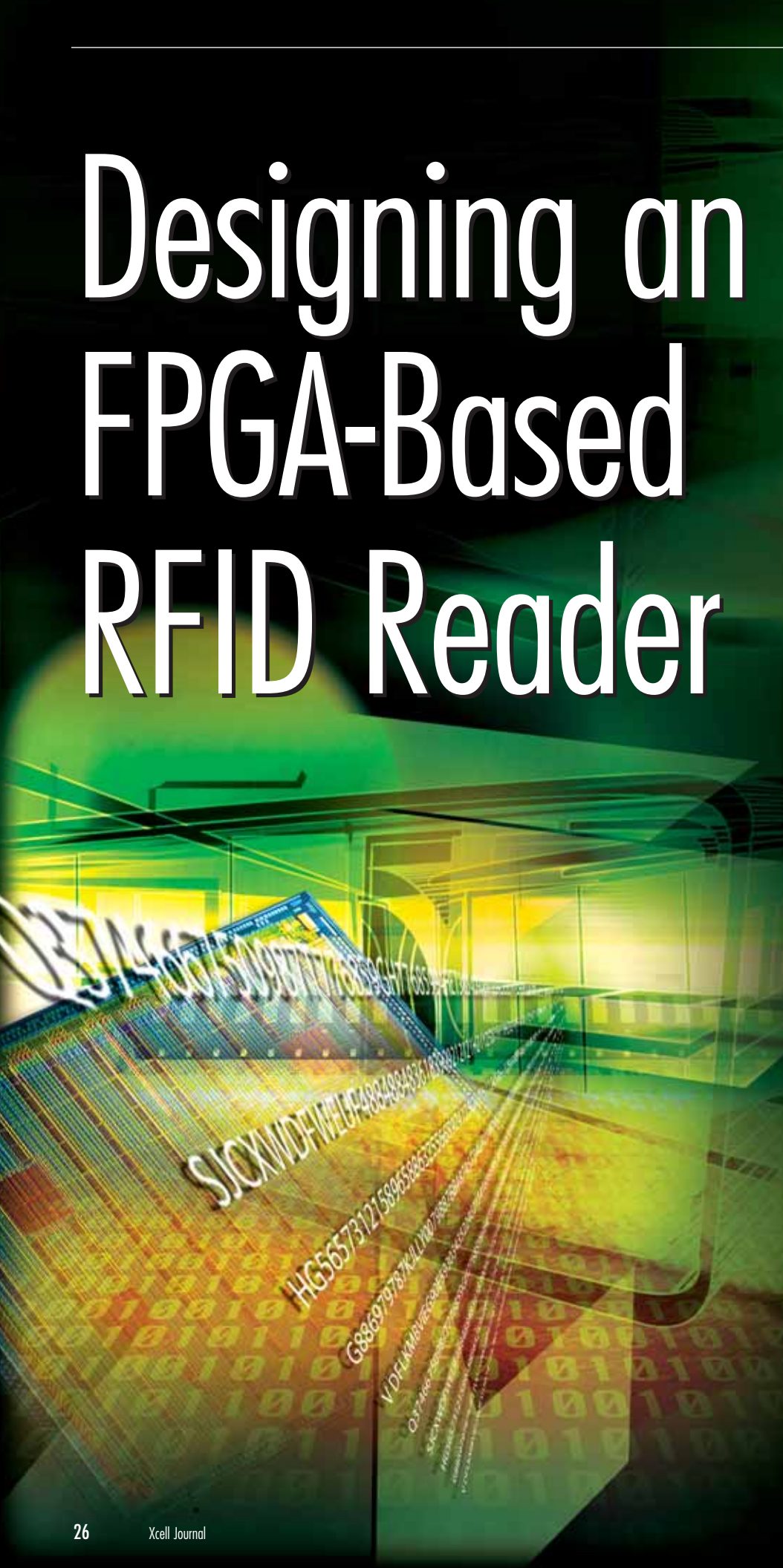
Viva may just be the easiest and most intuitive development tool ever created for FPGAs. That's because only Viva delivers a true data set recursive and polymorphic object oriented programming language with a graphical, drag-and-drop interface. In fact, Viva is so easy you can get the hang of it in a just few minutes. Whether you're working with FPGAs in high-performance or embedded environments, Viva is the right solution for you. And Viva is perfectly suited for use with hardware from SGI, Nallatech and Digilent, including Xilinx Starter Kits. Find out why organizations like NASA and the US Air Force use Viva! software to program FPGAs. Visit www.starbridgesystems.com/viva or call (801) 984-4444 today.

www.starbridgesystems.com/viva

starbridge

Viva! Absurdly Easy.

Designing an FPGA-Based RFID Reader



You can implement a practical, standards-compliant RFID reader using off-the-shelf RF components and FPGAs.

by Niladri Roy
Sr. Manager – ISM, Vertical Marketing
Xilinx, Inc.
niladri.roy@xilinx.com

Akshaya Trivedi
Sr. Systems Architect – Wireless, Vertical Marketing
Xilinx, Inc.
akshaya.trivedi@xilinx.com

James Wong
Product Marketing Manager
Linear Technology Corp.
jwong@linear.com

Radio-frequency identification (RFID) is an auto-identification technology similar to other common auto-ID technologies such as bar codes or magnetic strips. A physical object is associated with a unique identifying number (UID) that is incorporated in an RFID transponder, or "tag." The tag is usually attached to a physical object, such as a carton, a pallet, or a container filled with a product. An RFID reader (interrogator) extracts the UID from the tag.

A basic RFID system has three components: an antenna or coil, a transceiver with a RFID decoder, and an RFID tag programmed with the UID. Table 1 shows the four commonly used RFID frequencies and their potential applications. The frequency of greatest commercial interest at this time is UHF, with potentially large-volume applications in supply chain management.

	Frequency	Distance	Example Application
LF	125 KHz	Few cm	Auto-Immobilizer
HF	13.56 MHz	1m	Building Access
UHF	900 MHz	~7m	Supply Chain
µwave	2.4 GHz	10m	Traffic Toll

Table1 – RFID frequencies

EPC Tags

EPC stands for electronic product code, a standard for RFID tags, including both the data content of the tag and the open wireless communication protocols. The EPC movement combines data standards used in bar-code specifications with the wireless data communication standards that ANSI and other groups (802.11b) have developed. The EPC standard currently employed in supply-chain management is EPC Class 1 Gen II.

Class 1 tags are nominally factory programmed, but could be field downloadable. Normally, once the tag is written to, the memory is locked and further write operations disallowed. Class 1 tags use a conventional packet-oriented protocol, with the reader transmitting a packet containing commands and data, followed by a response from the tag.

Hostile Reader Environments

RFID system environments can be quite hostile. The frequency channels operate in a license-free ISM (industrial, scientific, and medical) band. RFID readers in this band are prone to external interference from cordless telephones, wireless headsets, wireless data networks, and other co-located readers. Each reader's RF receiver front end must be designed to withstand high-interference signal levels without introducing distortion products that cause interrogation errors. The receiver noise needs to be low so that it has sufficient dynamic range to allow error-free detection of low-level responding tag signals.

The reader RF transceiver architecture shown in Figure 1 is a proven design that works well in such dense, hostile environments. Both the transmitter and receiver incorporate a high dynamic range direct conversion modulator and demodulator, respectively, for maximum robustness and minimum cost.

A Practical Robust RF Receiver Design

The heart of the receiver is built around Linear Technology's LT5516, a highly integrated direct conversion quadrature demodulator with an on-chip precision quadrature phase (0° and 90°) shifter. The signal from the antenna, after passing through an RF filter, is fed directly into the demodulator inputs through a balun transformer. Because the noise figure of the LT5516 is low, an LNA (low noise amplifier) is not required, preserving its 21.5 dBm IIP3 and 9.7 dB P1dB performance.

During the receive cycle, the reader transmits a continuous wave (un-modulated) carrier to power the tag. When queried, the tag responds with a bitstream by amplitude modulating the carrier. The modulation format is ASK (amplitude shift key) or PR-ASK (phase-reversal ASK). The demodulator has two quadrature phase-detected outputs that provide an inherent diversity receive function. If one path receives no signal because of a multipath or phase cancellation, the other path (which is 90° phase shifted) would receive a strong signal, and vice versa. As a result, overall receiver robustness improves.

Once demodulated, you can AC-couple the I (in phase) and Q (quadrature phase) differential outputs to an op amp configured as a differential amplifier converted to a single-ended output. You should set the high-pass corner at 5 kHz – below the minimum signal content of the receive data stream and above the maximum Doppler frequency that may be introduced by a moving tag, while still well above the 60 Hz power-line frequency. The resulting outputs can then pass through a low-pass filter pair using the LT1568 configured as a fourth-order low pass. You should set the low-pass corner to a frequency of 5 MHz, allowing the maximum bitstream signal to pass through to the baseband.

The baseband signal can now be digitized by a dual, low-power analog-to-digital converter, the LTC2291, which has 12 bits of resolution. Because the tag bitstream has a bandwidth ranging from 5 kHz to 5 MHz, the LTC2291 provides ample oversampling at a rate of 25 MSps, accurately capturing the demodulated signal. If needed, you can implement additional digital filtering in the baseband DSP. This affords maximum flexibility for the receiver to set the logic threshold, which the baseband processor can perform digitally.

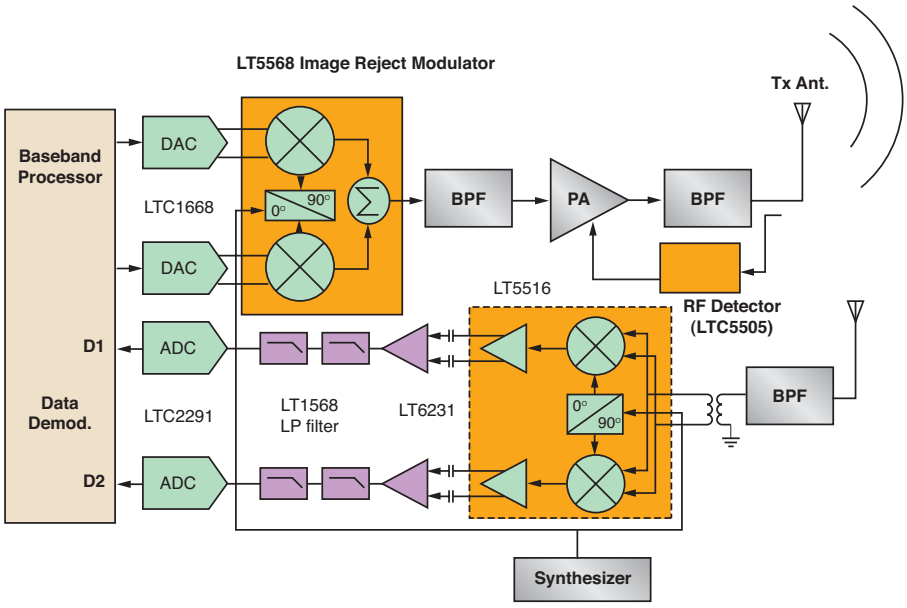


Figure 1 – A practical high-performance RFID transceiver architecture

The baseband tasks and digital RF channelization processing make an all-FPGA solution quite attractive and integrated.

A High Dynamic Range RF Transmitter Design

The transmitter employs an integrated image-reject direct conversion modulator. The LT5568 offers very high linearity and a low noise floor, providing exceptional dynamic range for the transmitted signals. The modulator accepts the baseband I and Q signals in quadrature from digital-to-analog (DAC) converters and modulates directly to the 900 MHz transmit frequency.

Internally, the LO (local oscillator) is split by a precision quadrature phase shifter. The modulated RF signals are combined to create a single-ended, single-sideband RF output with its image suppressed by 46 dBc. Moreover, the modulator has matching I and Q mixers to maximize the suppression of the LO carrier signal to -43 dBm.

The composite modulator circuit exhibits exceptional ACPR (adjacent channel power ratio) performance that helps to meet the transmit spectral mask required. For example, at a modulator RF output level of -8 dBm, the ACPR is better than -60 dBc. With its clean output, the signal can be amplified up to the maximum allowable power of 1W (+30 dBm in the U.S.) or as high as 2W for European compliance. In either case, it is important to maintain this fixed level, as it is used to power the tag and to maximize the reading range. The LTC5505 RF power detector's internal temperature compensation accurately measures the power and provides stable feedback to regulate the RF power amplifier output.

Baseband Processing and Network Interfaces

At the baseband, the FPGA performs channelization of the waveform to the DAC and from the analog-to-digital converter (ADC). This process, also called digital IF processing, will involve some filtering, gain control, frequency translation, and sample-rate change. The FPGA can even process multiple channels in parallel.

Figure 2 shows the partitioning of a potential RFID reader architecture. Other baseband processing tasks would include:

- Preamble detection
- Sequence estimation
- Modulation and demodulation (ASK, frequency and phase-shift keying)
- Signal generation
- Correlator processing
- Peak detection and thresholding
- CRC and checksum
- Encoding and decoding (NRZ, Manchester, unipolar, differential biphase, Miller)
- Frame detection
- ID descrambling
- Security crypto engines

This received RFID tag data is transmitted either over a serial port or a network interface to a corporate enterprise system server.

This traditional architecture is evolving to become part of a sophisticated distributed TCP/IP network, in which the reader will take on the management of its neighborhood tags. The reader now acts as a gateway between a tag and an intelligent distributed database system connected to enterprise software applications.

These baseband tasks could be realized on an FPGA or a DSP – or a combination of the two – based on hardware/software partitioning. Xilinx has a suite of IP cores in the form of FIR, CIC, DDS, DUC, DDC, bit correlators, and sine/cosine LUTs. The logic is well suited to perform crypto engines that use shift registers and XORs. The DSP48 engines for the Xilinx® Virtex™-4 family are well suited to perform other signal processing tasks.

A baseband processor controls the functionality and scheduling of the various baseband tasks; it is responsible for the link-layer protocols as well. These baseband tasks would include frequency-hopping, listen-before-transmit, and anti-collision algorithm processing. The baseband processor could also offer interfaces such as Ethernet, USB, or Firewire.

The baseband tasks and the digital RF channelization processing make an all-FPGA solution quite attractive and integrated. The FPGA functionality, DSP functionality, and baseband processor functionality could all be collapsed into an

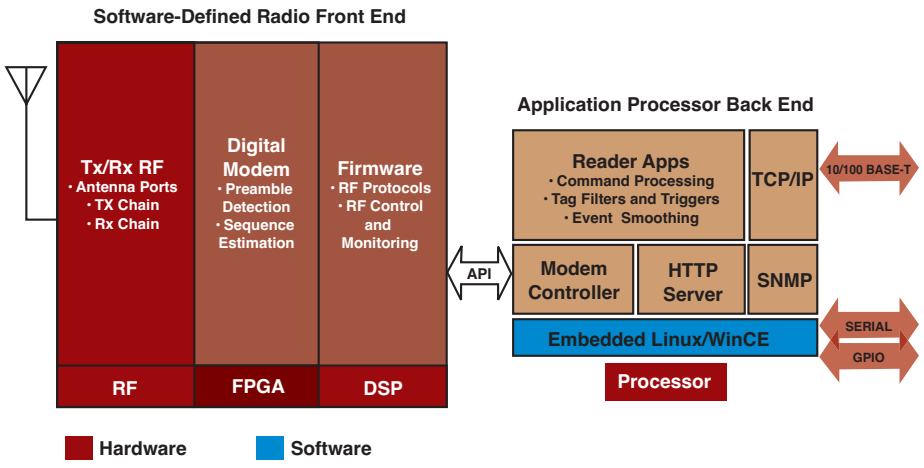


Figure 2 – Potential RFID reader architecture

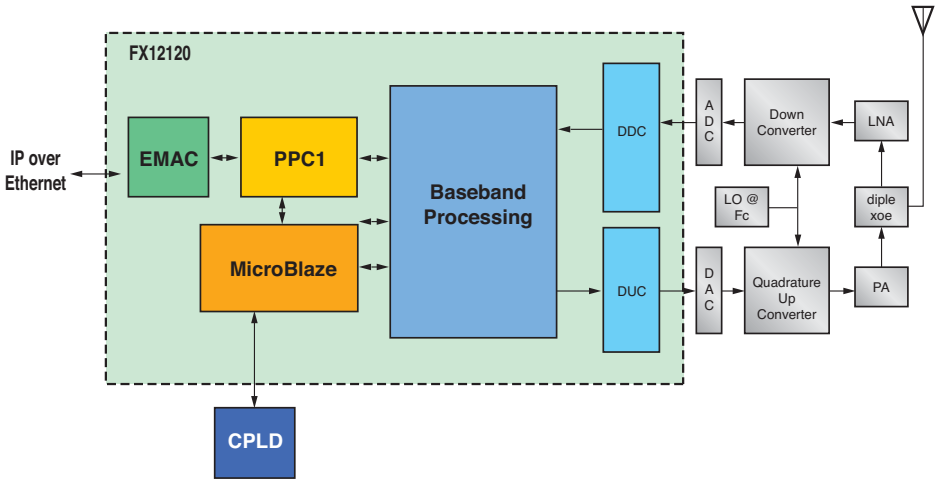


Figure 3 – Virtex-4 FX-based RFID architecture

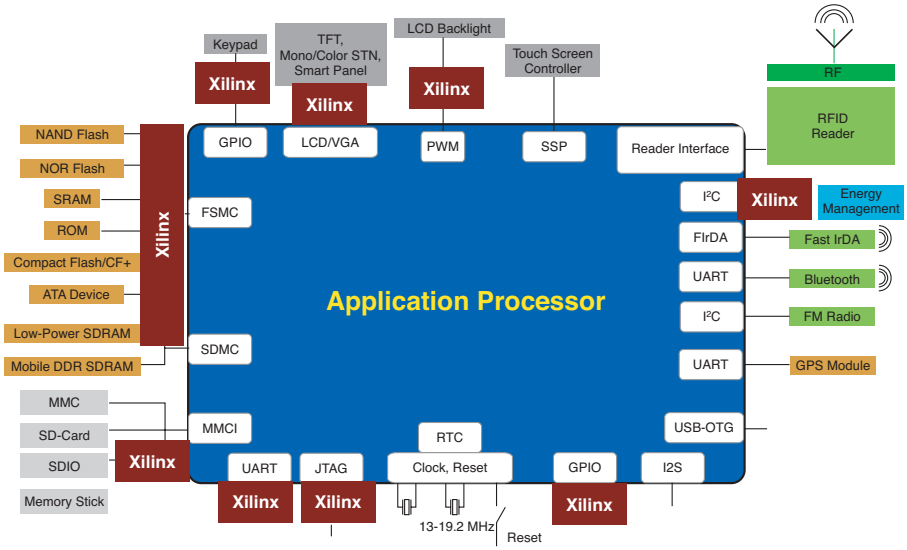


Figure 4 – CoolRunner-II CPLD in RFID readers

FPGA with an embedded processor.

Figure 3 illustrates the potential architecture of an FPGA-based RFID processor. The embedded processor could be a hard IP, like the Power PC™ in the case of the Virtex-4 FX family; it could be a soft-core MicroBlaze™ in Spartan™ devices; or even a PowerPC and MicroBlaze combination. You can connect the built-in hard Ethernet MACs (EMACs) to an external Ethernet PHY to connect to the Ethernet network. As an alternative, you can use Lite Ethernet MAC IP for 10/100-BaseT as well.

The Power PC/MicroBlaze embedded processors perform the following tasks:

- EPC data handling and forwarding
- Protocol handling
- Interrogator scheduling
- TCP/IP network interfacing
- Control and monitoring
- Modem controller
- Upgrade agent
- HTTP server
- SNMP/ MIB handling

The Xilinx Gigabit Ethernet System Reference Design (GSRD) is an EDK-based reference system geared toward high-performance bridging between TCP/IP-based protocols and user data interfaces. The components of the GSRD contain features to address the per-byte and per-packet overheads of a TCP/IP system.

TCP transmit performance benchmarks are available for Monta Vista Linux and Treck stacks. Nucleus PLUS RTOS using the Xilinx Platform Studio (XPS) microprocessor library definition (MLD) brings a new dimension to systems using MicroBlaze and PowerPC processors. Its small size means that it can use available on-chip memory to minimize power dissipation and deliver increased performance, while the extensively available middleware makes it ideal for RFID back-end networking.

Handheld readers can interface to hard disk drives, QWERTY keypads, removable memory interfaces, various display devices, and other peripherals using Xilinx CoolRunner™-II CPLDs, as shown in Figure 4. These CPLDs can also assist the application processor and address these features with little incremental power consumption, high-speed performance, and a small chip package.

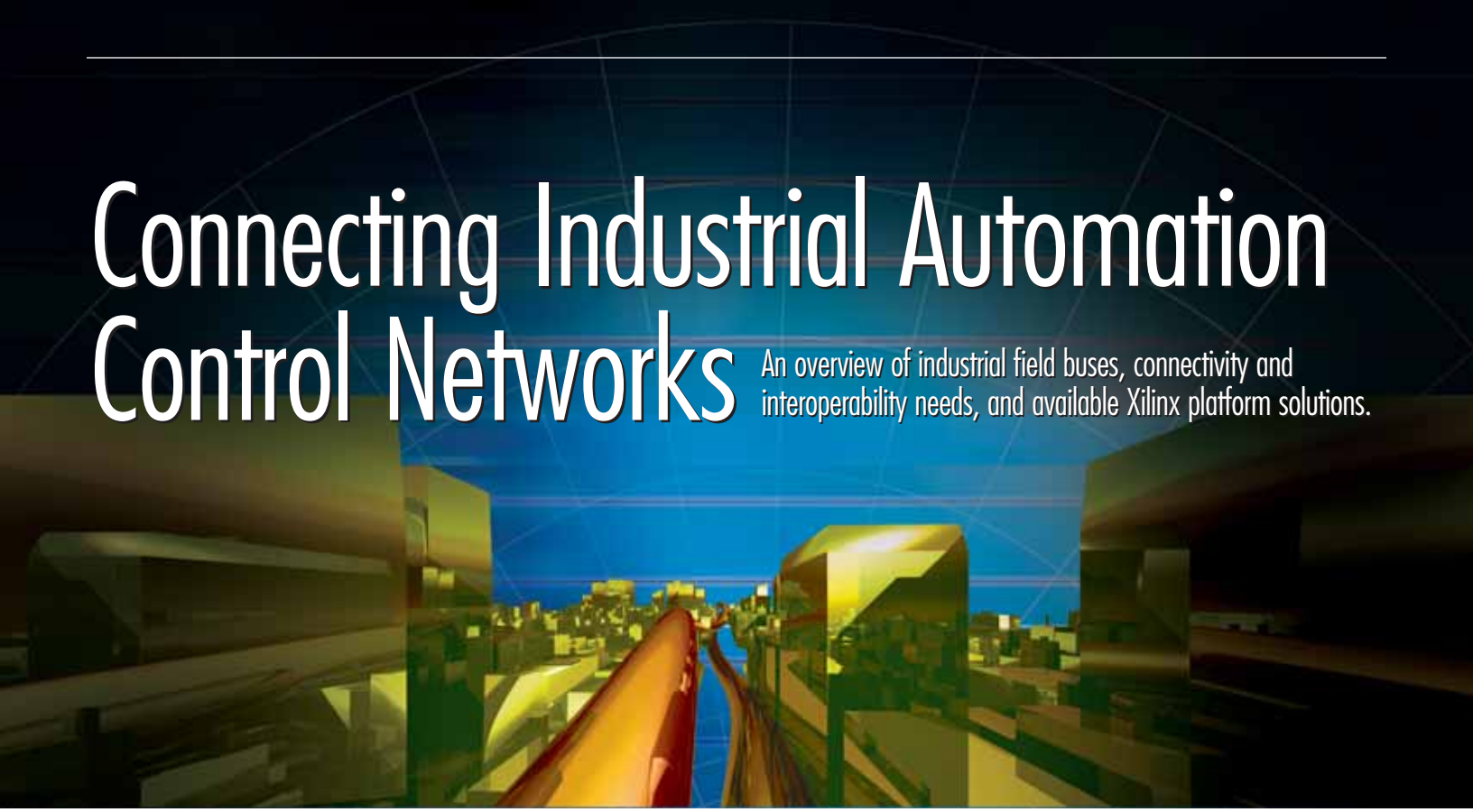
Conclusion

Going forward, RFID readers are likely to see front-end DSP functions like RF protocol processing – performed today in discrete DSP – integrated into the FPGA. Embedded soft-processor cores already provide significant DMIPS/MHz performance, and enhanced versions will soon replace back-end external processors for controlling reader applications, providing maximum flexibility and cost reductions for RFID reader equipment through programmable logic.

Xilinx is committed to providing continually enhanced DSP and embedded processing functionality in FPGAs through continuously improved XtremeDSP™ and MicroBlaze configurable soft-processor core offerings.

Connecting Industrial Automation Control Networks

An overview of industrial field buses, connectivity and interoperability needs, and available Xilinx platform solutions.



by Niladri Roy
Sr. Manager – ISM, Vertical Markets
Xilinx, Inc.
niladri.roy@xilinx.com

Industrial automation control networks are serial-bus based multidrop networks that intelligently connect industrial sensors, actuators, devices, and process control subsystems to each other and to enterprise networks. These networks demand interoperability of control functions at different hierarchical levels, ranging from field equipment to enterprise resource planning (ERP).

Open control systems still focus on data flow between devices from different vendors. Only a few efforts are underway to standardize architectures and application objects that would make systems understand each other.

A field bus is a generic term for industrial automation communication buses. More than 20 field buses have existed at various times, and current standardization attempts (the IEC IS 61158, IS 62026, and ADIS 62026) list 13 different field buses, as shown in Table 1.

IEC STANDARD	Field Bus Name	Comment
IS 61158 Type 1	Foundation Field Bus H1	US Automation Field Bus
IS 61158 Type 2	ControlNet	Control Network
IS 61158 Type 3	Profibus	Device Network
IS 61158 Type 4	P-Net	Factory, Shipbuilding
IS 61158 Type 5	Foundation Field Bus HSE	Control Network
IS 61168 Type 6	SwiftNet	Developed by Boeing
IS 61158 Type 7	WorldFIP	French Device Network
IS 61158 Type 8	Interbus	Discrete I/O
IS 61158 Type 3	Profibus DP	Control Network
IS 61158 Type 7	Acuator-Sensor Interface (Asi)	Discrete I/O
IS 62026-2	DeviceNet	Discrete I/O
IS 62026-5	Smart Distributed System	
ADIS 62026-6	Seriplex	

Note on Table 1 – Other industrial networking protocols like EtherCAT are currently undergoing integration into the IEC standards and are part of the publicly available specifications (PAS). The IEC also approved the ETHERNET Powerlink protocol in March 2005.

Table 1 – Current field buses

A unified communication between diverse field buses is driven primarily by the following requirements:

- Demand for greater control over processes
- Increased diagnostic capabilities and remote diagnostics
- Higher plant availability
- Improved wiring (reduced cabling)
- Integration of condition monitoring and plant maintenance
- Flexibility for upgrades
- Integration with enterprise (corporate) networks

There is a need for an integrated and low-cost system platform that allows intelligent features to be easily implemented. Integrating a large number and variety of legacy equipment and field buses also requires bridging disparate field buses to each other – and to the enterprise. Moreover, the IEC 61158 is not a unified standard but a collection of different field buses, ensuring that field bus "turf wars" and the need for integration will continue.

The Xilinx® Spartan™-3 generation of FPGAs, particularly the Spartan-3E family, provides a full-featured, low-cost programmable hardware platform ideally suited for the implementation of intelligent solutions in industrial automation.

Xilinx in Industrial Automation Control Networks

Xilinx offers programmable logic solutions in all networking categories and automation hierarchies. Low-cost Spartan-3E FPGAs enable extremely economical deployment in a variety of devices and controllers.

Xilinx XtremeDSP™ technology enables rapid DSP computation in hardware, bringing unprecedented resolution to multi-axis motion control through DSP digital drives. The MicroBlaze™ soft processor implemented in FPGAs enables offloading of functions that previously required external microcontrollers. This paves the way for compact, elegant, and cost-effective solutions with lower chip counts.

IP in the form of UART, Ethernet MAC, PCI, PCI Express, and CAN2.0B cores accelerate hardware design for bridging with industrial Ethernet buses and PC-based intelligent controllers.

Figure 1 shows the deployment of Xilinx FPGAs in industrial automation. In addition to being used in device and system controllers, Xilinx FPGAs address applications in the following areas:

- Media converters
- Switches
- Device servers
- Bridges
- Gateways

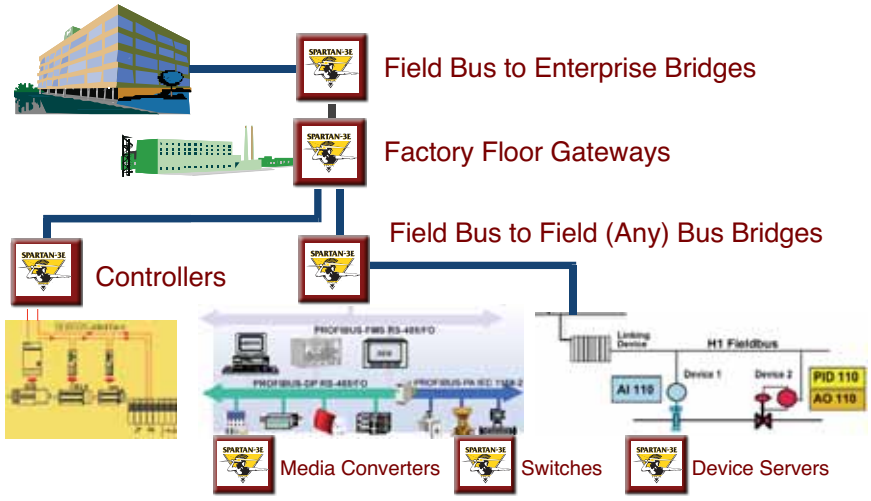
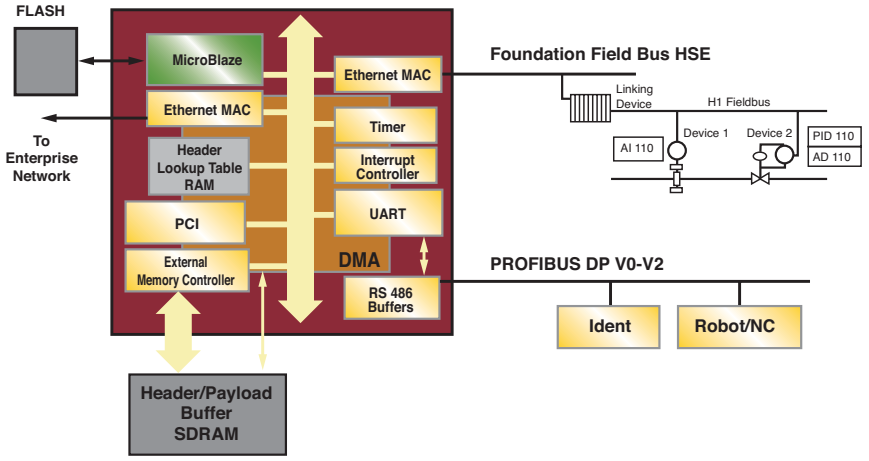


Figure 1 – Xilinx in industrial automation

The Spartan-3E family of FPGAs offers density ranges from 50,000 to 5 million system gates and 216 Kb to 1,826 Kb of block RAM to accommodate scalable processing needs. Seamless density migration over a range of devices and packages with no need for expensive re-layout protects system-vendor investments and reduces overall cost across product roadmaps (see Table 2).

In addition to the embedded MicroBlaze processor, UART and Ethernet MAC, Xilinx Spartan-3E FPGAs sport one of the world's lowest cost parallel interconnect solutions. Real PCI from Xilinx offers PCI v2.2-compliant initiator and target cores with guaranteed timing and zero wait states and a full 64-bit data path, making Spartan-3E FPGAs the family of choice for building gateways connecting field bus to field bus and field bus to enterprise, as well as field bus to intelligent PC-based controller subsystems.

A block diagram of a gateway implemented in Spartan-3E devices, along with estimated logic cell count, is shown in Figure 2. Xilinx PCI Express and CAN2.0B cores provide additional support for more industrial interfaces and modern PC-based controllers to replace programmable logic controllers (PLCs).



Functional Block	Logic Cell Count	
	Min.	Max.
MicroBlaze Processor	950	1,050
SDRAM Controller	200	250
Ethernet MAX x 2	6,594	7,520
PCI	3,950	3,950
UART	400	400
DMA Engine x3 or 4 Channels	600	800
Timer	255	255
Interrupt Controller (2 - 32 Interrupts)	46	370
Total	12,995	14,595

Figure 2 – Implementing an industrial automation gateway in Xilinx Spartan-3E FPGAs

Industrial Ethernet

Ethernet was initially ignored by the automation industry because of its perceived lack of determinism and robustness. At 10 Mbps, CSMA/CD was unable to provide the guaranteed real-time (5 to 20 ms) responses demanded by industrial control. The combination of 100-BaseFX (100 Mbps over two strands of multimode fiber per link), switched media instead of hubs, and full-duplex operation successfully overcame those limitations. Ethernet has evolved into a technology that the automation and control industry is swiftly adopting.

You can use industrial automation gateways of the type shown in Figure 2 to bridge incompatible Ethernet networks, with protocol translation implemented in software.

Xilinx sees the market for real-time Ethernet based products as strategic to industrial automation connectivity. The advent of real-time Ethernet networks offers the potential for standardization in industrial networking, starting with the physical and data link layers, as well as seamless connection to enterprise networks. Xilinx FPGAs offer low-cost platforms for the implementation of real-time Ethernet networks, and are well suited for implementation of evolving standards that require field upgradability to comply with enhancements and changes in features.

Device	XC 3S100E	XC 3S250E	XC 3S500E	XC 3S1200E	XC 3S1600E
System Gates	100K	250K	500K	1,200K	1,600K
Logic Cells	2,160	5,508	10,476	19,512	33,192
18x18 Multipliers	4	12	20	28	36
Block RAM Bits	72K	216K	360K	504K	648K
Distributed RAM Bits	15K	38K	73K	136K	231K
DCMs	2	4	4	8	8
I/O Standards	18	18	18	18	18
Max Differential I/O Pairs	40	68	92	124	156
Max Single-Ended I/O	108	172	232	304	376
Package	**User I/O**	**User I/O**	**User I/O**	**User I/O**	**User I/O**
VQ100	66	66	–	–	–
TQ144	108	108	–	–	–
PQ208	–	158	158	–	–
FT256	–	172	190	190	–
FG320	–	–	232	250	250
FG400	–	–	–	304	304
FG484	–	–	–	–	376

Table 2 – Xilinx Spartan-3E product matrix

Table 3 shows the 17 different interfaces offering real-time Ethernet performance as of November 2005. Notable among the 17 for the traction they have gained in a short time are EtherCAT, SERCOS III, and Ethernet Powerlink.

EtherCAT processes 1,000 distributed I/O in 30 µs or 100 axis in 100 µs using twisted pair or fiber optic cable. EtherCAT topology supports a simple low-cost line structure, tree structure, daisy chaining, or drop lines; no expensive infrastructure components are required.

SERCOS III merges proven mechanisms of SERCOS with the Ethernet physical infrastructure. Data rates of 100 Mbps, combined with high protocol efficiency, cut cycle times in half compared to previous SERCOS versions, setting a new standard for rugged real-time applications: a 31.25 µs cycle time for as many as 8 drives with 8-byte cyclic data.

ETHERNET Powerlink is an ISO/OSI level 2 protocol enabling deterministic, isochronous, real-time data exchange through standard Fast Ethernet (IEEE 802.3 u). It allows high-precision data communication with cycle times as low as 100 µs and network jitter well below 1 µs.

Available and ongoing Xilinx solutions in the real-time Ethernet area include Spartan-3E implementations of these interfaces.

The Software Component of Seamless Integration

The solution to seamless integration in industrial automation will probably lie in the overall adherence of all connected equipment to OPC (OLE for process control). OPC is an industry standard created through the collaboration of a number of leading worldwide automation hardware and software suppliers working in cooperation with Microsoft.

Based on Microsoft's OLE (object linking and embedding, now ActiveX), COM (component object model), and DCOM (distributed component object model) technologies, OPC comprises a standard set of interfaces, properties, and methods for use in process-control and manufacturing-automation applications.

The ActiveX/COM technologies define how individual software components can interact and share data. OPC provides a common interface for communicating with diverse process-control devices, regardless of the controlling software or devices in the process.

In the enterprise, which is predominantly a Microsoft Windows environment, OLE is already used to provide integration among applications, enabling a high degree of application compatibility. OLE technology enables the develop-

ment of reusable plug-and-play objects that are interoperable across multiple applications. It also facilitates reusable, component-based software development, where software components can be written in any language, supplied by any software vendor. Phased migration to OPC support by factory automation vendors will one day make seamless factory-to-enterprise linkage a reality. Real-time operating systems such as WindRiver VxWorks AE already offer extensions like VxOPC to support OPC.

Xilinx leverages its partnerships with embedded OS vendors to complement the Spartan-3E hardware platform and deliver complete system solutions.

	Name	Organization / Main Manufacturer
1	EtherCAT (Ethernet for Control Automation Technology)	ETG / Beckhoff
2	Ethernet/IP mit CIP Sync	ODVA / Rockwell Automation
3	ETHERNET Powerlink (EPL)	EPSG / B&R
4	Precision Time Protocol (IEEE 1588)	IEEE / -
5	PROFINET	PNO / SIEMENS
6	RTPS (Real-Time Publish-Subscribe Protocol)	Modbus-IDA / Schneider Electric
7	SERCOS-III	IGS / -
8	SynqNet * (only Layer 1 based on Ethernet)	SynqNet User Group / Motion Engineering Inc. (MEI), USA
9	JetSync	- / Jetter, Germany
10	PowerDNA (Distributed Network Automation and Control)	- / United Electronic Systems (UEI), USA
11	SynUTC	- / Oregano Systems, Austria
12	Switch mit Zeit-Server	- / Ontime Networks, Norway
13	RTnet (Open Source)	- / Real-Time Systems Group of University Hannover, Germany
14	Vnet/IP	- / Yokogawa, Japan
15	TCnet	- / Toshiba, Japan
16	EPA (Ethernet for Plant Automation)	- / ZHEJIANG SUPCON INSTRUMENT, China
17	SafetyNET p	SafetyBUS p Club / Pilz

Table 3 – Real-time Ethernet industrial interfaces

Conclusion

Increasing processor power, advances in consumer electronics, mobile communication networks, and platform-independent programming languages provide the tools to implement smart functions that were previously unrealistic.

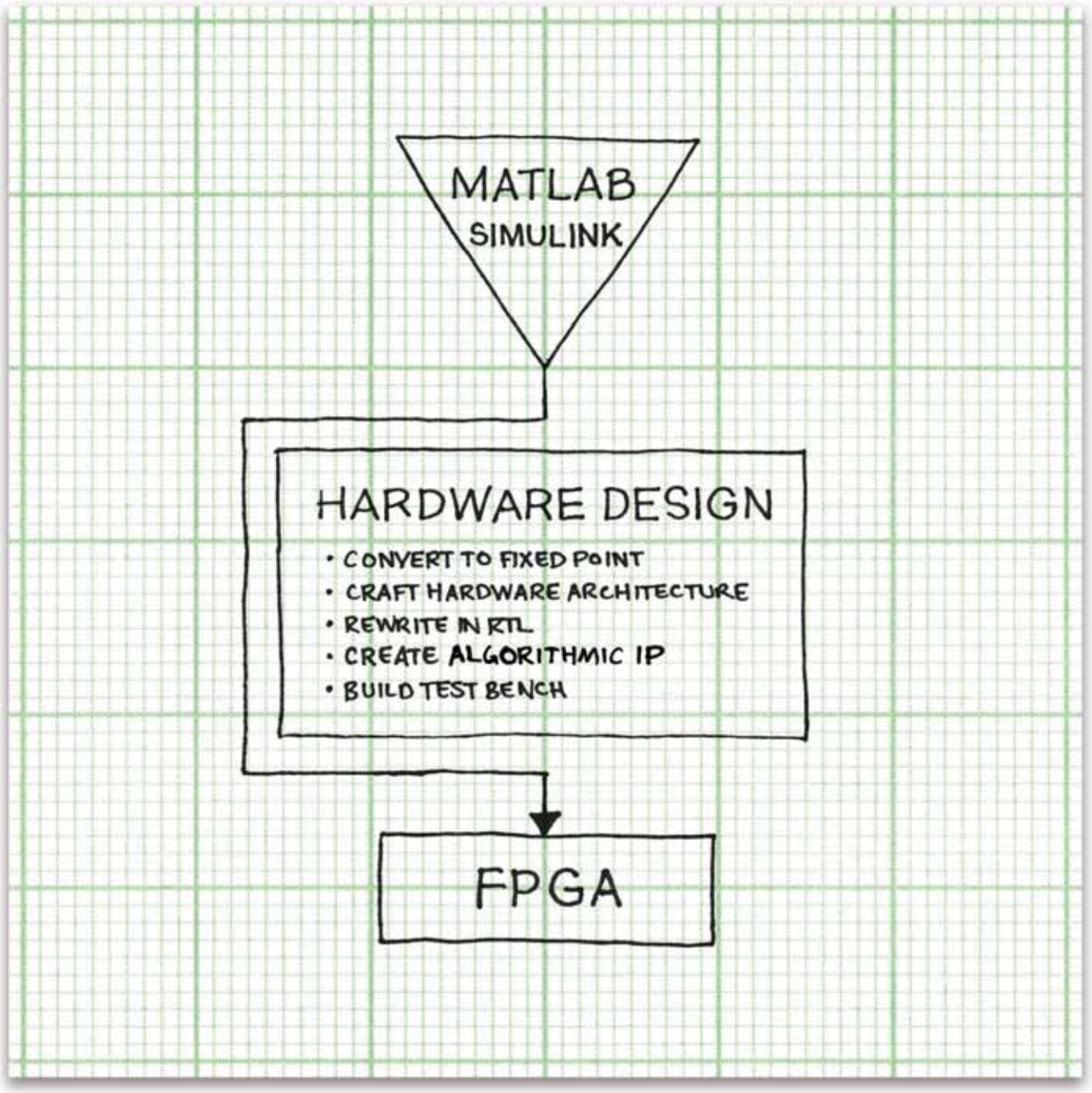
IEEE 1394, for example, can operate as a real-time capable and high-speed backbone, providing a transparent interconnection with guaranteed propagation delay for field-area network segments expected to carry multimedia traffic. Modular DSP systems employing 400 Mbps IEEE 1394 (FireWire) communications are already available for industrial automation applications, integrating high-speed digital drives for multi-axis motion control applications, digital cameras, and other FireWire peripherals.

Even the USB protocol is used in industrial automation, although its usage seems to be confined to providing a painless plug-and-play solution for migrating serial protocol (RS 232, RS 422, and RS 485) devices to Windows-based PCs, which are increasingly replacing PLCs on the factory floor.

The standardized TCP/IP protocol offers a seamless portal into the Internet; a Web server can be integrated into even small control devices, with familiar technologies such as programming languages and net browsers easily available. The eXtensible Markup Language (XML) is also a promising tool for application integration between factory and enterprise.

Wireless is likely to generate increasing interest in the industrial automation community. The benefits of wireless communication include flexibility and savings in cabling costs. Using a wireless mesh as a backbone network, for example, simplifies installation and provides an affordable, unobtrusive, and completely portable solution for both small- and large-scale deployments. In addition to being easy to install, mesh network nodes can be added virtually anywhere Ethernet is required, including time clocks, scales, surveillance cameras, and even on moving equipment such as forklifts, cranes, and conveyor systems. Wireless versions of field buses are also on the horizon.

Xilinx is committed to supporting advances in industrial automation through continued innovation, offering high-performance, feature-rich fabrics including embedded soft processors, DSP engines, and an increasing array of common discrete functions such as clock management, LVDS transceivers, and level shifters integrated on-chip. This is complemented by memory and system interface connectivity solutions to enable superior and efficient management of system cost and product lifecycles for system vendors, enabling them to exploit new market opportunities before their competitors.



Now, There's A Flow You Could Get Used To.

XtremeDSP Sidestep tedious hardware development tasks with Xilinx System Generator for DSP and AccelDSP design tools. When combined with our library of algorithmic IP and hardware platforms, Xilinx XtremeDSP Solutions get you from design to verified silicon faster than ever.

Visit ***www.xilinx.com/dsp*** today and speed your design with AccelDSP and System Generator for DSP.

XILINX®

www.xilinx.com/dsp

FREE on-line training *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**. Order yours today!

XILINX®

www.xilinx.com/dod

Pb-free devices available now

IDE Hard Disk Drive Solutions for Smart Handheld Applications

iWave Systems offers a hardware companion for easy data storage.

by Jishad C.K. Abubacker
Member Technical
iWave Systems Technologies Pvt. Ltd.
jishad@iwavesystems.com

The increasing functionality required in portable digital consumer appliances has driven the demand for increasing amounts of in-system storage. Given the significant improvements in form factor, capacity, reliability, and price, many applications now include hard disk drive (HDD) technology, including MP3 players, personal media players, GPS navigation systems, car entertainment systems, portable gaming, and smart phones. For example, the Philips HDD120 MP3 player/20 GB recorder uses a portable HDD, which is smaller in size, lightweight, and consumes less power.

The prevalent interface deployed on today's HDDs is Advanced Technology Attachment (ATA), more commonly known as Integrated Drive Electronics (IDE).

iWave Systems has introduced an IDE/ATA controller implemented in low-power Xilinx® CoolRunner™-II CPLDs, which supports the following features:

- Intel PXA270 CPU static memory interface in 16-bit variable latency input and output (VLIO) mode
- Supports ATA PIO mode 0
- Supports ATA multi-word direct memory access (DMA) mode 0, 1, and 2
- Supports external SRAM interface for data buffering
- Data throughput:
  - The maximum data throughput achieved on DMA mode 2, with a 16-bit CPU interface and SRAM access time of 55 ns, is 6.4 MBps The maximum data throughput achieved on DMA mode 2, with a 16-bit CPU interface and SRAM access time of 10 ns, is 7.4 MBps
  - The maximum data throughput possible on DMA mode 2, with a 32-bit CPU interface and SRAM access time of 10 ns, is 10 MBps
- Supports only one hard disk interface

System Overview

Figure 1 is a basic system block diagram, adding HDD storage to the Intel Bulverde PXA27x processor. The processor serves as the host in the system and manages traffic to and from the HDD. The IDE controller connects to the Intel CPU through its static memory interface (VLIO interface) and is a full-featured IDE controller companion device for the Intel PXA27x processor family.

By utilizing the DMA on the processor, the IDE controller offloads the CPU during data transfers to and from the HDD, providing improved system performance and reduced system power.

The IDE controller also manages the ATA interface, control registers, and ATA timing, in addition to providing support for PIO mode 0. Integrating an on-board SRAM (1,024 byte minimum), the IDE controller allows the CPU to perform read and write operations without interruption or waiting for the externally connected IDE device to be ready.

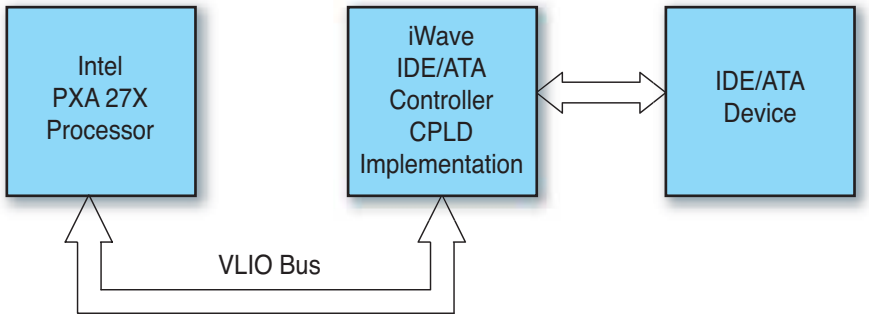


Figure 1 – Functional block diagram

PIO Modes	Maximum Throughput
0	3.3 MBps
1	5.2 MBps
2	8.3 MBps
3	11.1 MBps
4	16.7 MBps
Multi-Word DMA Modes	**Maximum Throughput**
0	4.2 MBps
1	13.3 MBps
2	16.7 MBps
Ultra DMA Modes	**Maximum Throughput**
0	16.7 MBps
1	25.0 MBps
2	33.3 MBps
3	44.4 MBps
4	66.7 MBps
5	100 MBps

Table 1 – Theoretical maximum throughput for ATA specified modes of operation

ATA Operation Modes

Table 1 outlines some of the different modes within the ATA specification and their theoretical maximum data rates.

Processor Overview

The PXA27x processor is part of the 32-bit Intel Xscale micro-architecture family of embedded processors. This family of processors integrates a set of sophisticated peripheral functions:

- CPU core: Intel XScale micro-architecture, up to 624 MHz
- Cache: 32 KB instruction, 32 KB data, 2 KB "mini" data
- Internal SRAM: 256 KB for high-speed code/data storage
- Peripherals:
  - Audio Codec '97/I2S audio port
  - USB client/host/OTG controller
  - Three UARTs
  - FIR/SIR communication port
  - LCD controller
  - SD/MMC controller
  - I^2C controller
  - Pulse width modulation (PWM)
  - JTAG
  - Camera interface
  - USIM interface
  - Keypad interface
- Memory Controller: SDRAM, Flash, SRAM, and PC cards
- Low power: <500 mW (typical)
- Package: 23 mm x 23 mm or 13 mm x 13 mm

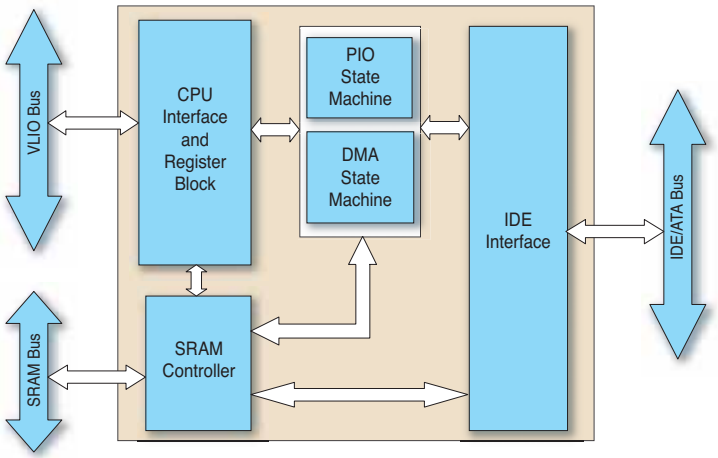


Figure 2 – ATA/IDE controller block diagram

Figure 3 – Intel PXA handset application reference platform board

Controller Overview

The iWave IDE controller design is based on the Xilinx CoolRunner-II CPLD (XC2C256) in a CP132 package. It can also be implemented in several other devices for additional logic integration and greater flexibility. The design comprises the blocks (or modules) shown in Figure 2.

CPU Interface and Register Block

The local bus interface, through use of the ready (RDY) signal, handles the PXA27x VLIO automatically. The RDY signal is used to properly synchronize the IDE controller device's responses to the CPU read and write commands during high-speed bus operation. When the SRAM is ready for a read or write operation, the IDE controller asserts an interrupt to the CPU to indicate that it is ready for transfer.

The register block has several registers for setting the controller mode, indicating the state machine status and selecting the mode of operation.

SRAM Controller

The SRAM controller clearly defines the interface between the IDE controller and SRAM. It generates the control signals and buffers the data whenever the processor or IDE controller is accessing the SRAM.

State Machines Block

This block comprises two state machines: PIO and DMA. These state machines are responsible for the reading and writing of IDE data, including accesses to external IDE device registers. All IDE signal timing is implemented in this block. The PIO state machine will control transactions while working in PIO mode. The DMA state machine will control transactions while working in DMA modes.

IDE Interface

This module controls the interface towards IDE. This will distinguish the data transaction towards IDE by the state machines block and SRAM controller block.

Mini HDD Overview

There are several drive vendors to choose from. Some of the more popular vendors are:

- Hitachi Ltd.
- Toshiba America, Inc.
- GS Magic, Inc.
- Cornice, Inc.
- Seagate Technology, Inc.

Reference Platform

The PXA27x processor handset application reference platform supporting the HDD storage capability daughtercard is shown in Figure 3. The HDD daughtercard is shown in Figure 4.

IDE Solution Pack

The iWave development package includes:

- Intel PXA27x handset application reference platform

Figure 4 – HDD daughtercard

- IDE controller reference design files
- Windows CE device driver and application note
- User guide and documentation

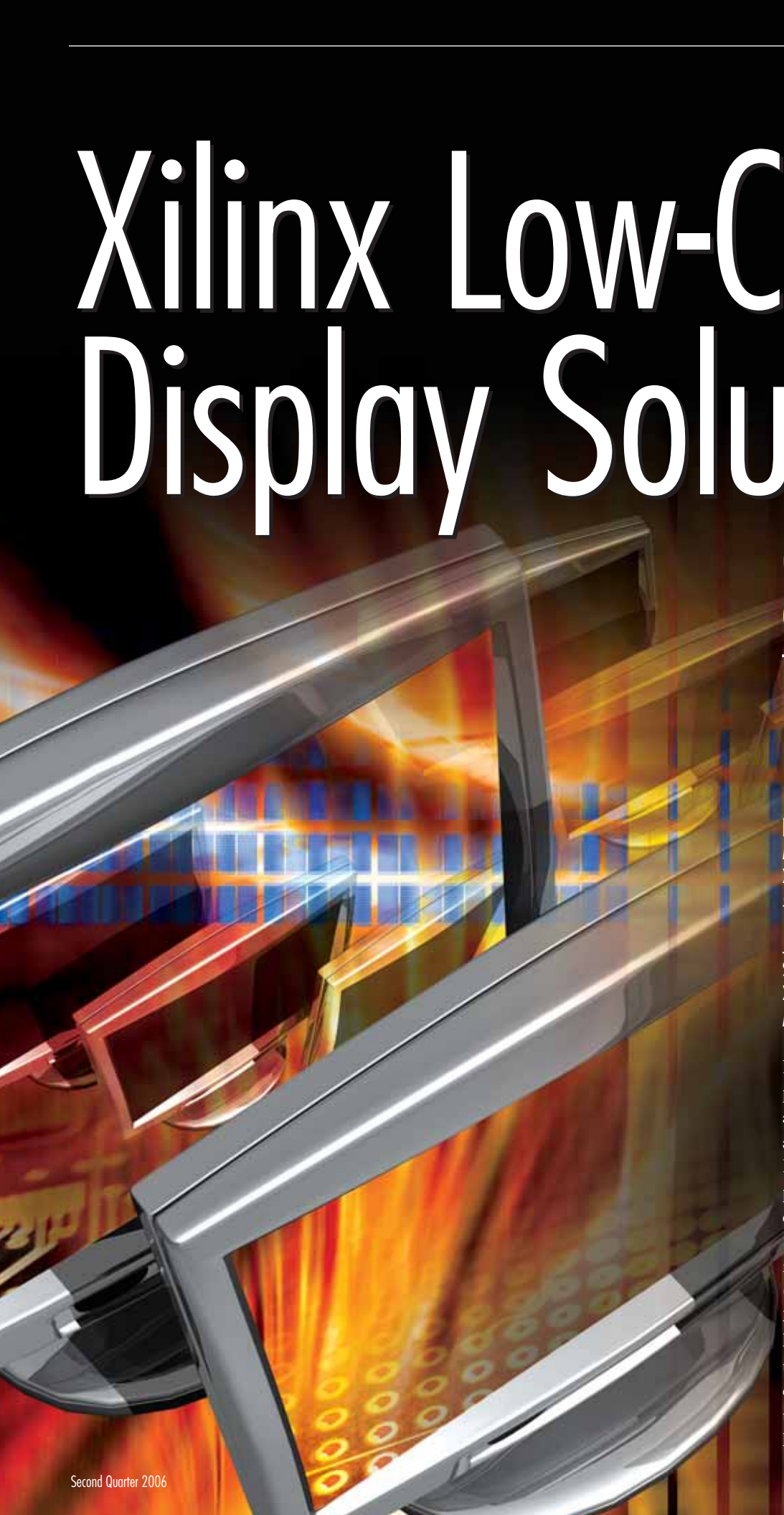
Conclusion

iWave Systems offers a very low-power IDE interface implemented in CPLD, which acts as a companion chip for the Intel Bulverde PXA 27x processor. With a 16-bit interface design, this interface provides a data rate of 7.4 MBps, which makes it possible to use in portable devices without trading off performance and power consumption.

We are currently developing a 32-bit IDE interface device, which can provide a data rate of 10 MBps in the DMA mode. Support for other processors is possible with minimal changes to the design. We also plan to provide a Linux device driver in the near future.

For more information, e-mail *mktg@iwavesystems.com*, or call Abdul Basith at +(91) 80 26683700.

Xilinx Low-Cost Display Solutions



Get your display designs to market faster with Spartan-3E FPGA hardware and IP solutions.

by Steve Prokosch
Manager, Vertical Markets, Consumer
Xilinx, Inc.
steve.prokosch@xilinx.com

Josephine Sy
Sr. Manager, Vertical Markets, Consumer
Xilinx, Inc.
josephine.sy@xilinx.com

Not long ago, high-definition flat-panel display televisions were too expensive for the average consumer. Today, they are abundant and affordable for most household incomes. Panel manufacturers are expanding their production capacities to meet demands and encourage greater adoption.

According to iSuppli, worldwide shipments of LCD and plasma display panel (PDP) TVs are expected to increase from 20 million units in 2004 to nearly 90 million units in 2009. With a wide range of technologies and sizes, they are replacing cathode ray tube (CRT) TVs at a substantial rate - even though prices are slightly higher for the same viewing area.

Consumers are looking for the features that differentiate the plain models from those with higher resolutions, better color reproduction, and the least amount of digital motion artifacts. These features determine price points beyond the lowest entry point for a specific screen size.

Some companies brand their image-enhancement technology so that they can charge a premium. So how do you cost-effectively develop, test, and manufacture premium features for your company brand without investing large capital and resources? Xilinx and its partners have developed a complete suite of solutions that let display developers speed their product to market. These solutions include the Xilinx® Spartan™-3E Display Solutions Board, along with several advanced video/image processing IP cores, to accelerate the design of your feature-rich flat-panel display.

The Spartan-3E Display Solutions Board includes the Spartan-3E device, our best-selling FPGA for consumer applications, memory, and comprehensive connectivity support to enable connections to display panels and video/tuner boards.

Why Should You Use an FPGA in Display Applications?

Digital display product designers are continually under pressure to integrate more functions, service more needs, support more standards, and still go to market faster than ever before. ASIC vendors can produce customized chips to meet your required features, but it takes more than a year of development for increasingly complex video/image processors, and you will incur a large NRE charge with finer fabrication processes.

Although many ASSP solutions are available from semiconductor vendors such as Pixelworks, Genesis, Trident, Philips, and ST, they do not allow display manufacturers to produce house-brand premium features. Hence, programmable solutions can complement ASIC solutions, with more frequent incremental improvements to the whole system, and also complement ASSP solutions, with product differentiation. This differentiation can come in the form of image-enhancement technology, peripheral interfaces, storage interfaces, connectivity, or even simple implementations such as GPIO extension.

Xilinx FPGAs are thus an excellent platform for implementing these functions because of their programmability and flexibility. In addition, they are exceptionally useful in many digital video applications because of their parallel processing operation and integrated features specifically targeted at the consumer display market. Examples of these features include multiple interface standards for both video and display panels, DSP functionality, digital clock managers, voltage-level translation, and high I/O package options.

Along with all of these features, Spartan-3E FPGAs are low priced because of the company's migration to a smaller process technology and larger wafer sizes while maintaining high wafer yields. Interconnectivity has always been the traditional strength of reprogrammable logic. Today, with changing standards and interface technology, FPGAs are becoming more of a necessity in display solutions.

Spartan-3E Display Solutions Board

Xilinx now offers a development platform that enables the efficient design and validation of leading-edge display technologies in flat-panel TV monitors, rear-projection TVs, and digital projectors to speed designs to market. With this platform, you can create and modify complex algorithms to enhance display image quality and color settings.

The Spartan-3E Display Solutions Board integrates a Spartan-3E 1.6-million gate XC3S1600E FPGA in a 484-pin BGA package, an associated configuration PROM, and a serial peripheral interface that you can use to optionally program the FPGA. Additionally, the board contains four 512 Mb x16 DDR SDRAM memories, which are ideal for supporting frame buffering.

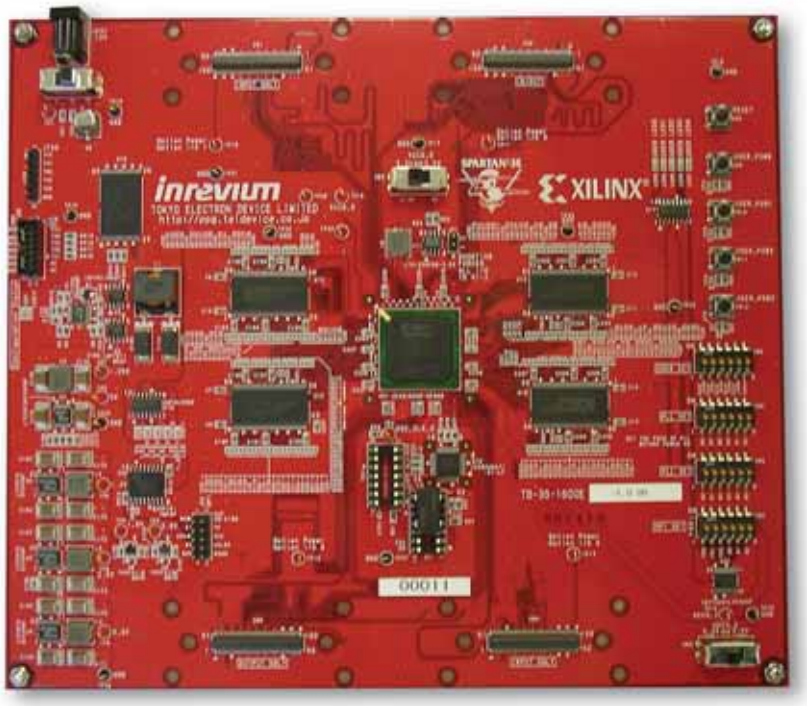


Figure 1 - Spartan-3E Display Solutions Board

With the Spartan-3E Display Solutions Board, you can expect a 266 Mbps DDR SDRAM controller and image processing algorithm reference designs. The board also comes with optional boards such as GPIO, USB, CameraLink, and a DVI option board. These optional boards allow you to easily connect to standard display panels and video

boards and to verify connectivity solutions and video/image processing algorithms.

Figure 1 shows the Spartan-3E Display Solutions Board. As a display designer, you can use this board to develop better image quality, multiple memory interfaces, and varying form factors for flexible, low-cost, high-performance solutions. You can also incorporate critical display features such as integrated microcontrollers, embedded MicroBlaze™ 32-bit RISC processors, LVDS, RSDS, and mini-LVDS interfacing, parallel digital signal processing, on-chip memory, and high-speed external memory access on a single XC3S1600E Spartan-3E device.

Reference Designs and IP

In conjunction with the Spartan-3E Display Solutions board, Xilinx also provides associated reference designs for enhancing image quality. These reference designs for display solutions include precise gamma correction, an image dithering engine, color temperature correction, and false contour reduction. With these designs, you can rapidly deploy a base platform to market and then develop other features such as sharpness enhancement, dynamic gamma correction, or noise reduction.

Xilinx also has third-party developers offering other display solutions. For instance, Digital Design Corporation (*www.digidescorp.com*) offers fully tested advanced soft cores including OpenGL, Symbology Overlay, and Scaling/Decimation. Table 1 shows the different reference designs available from Xilinx and its partners for display panel and video board applications.

Conclusion

Image quality in flat-panel displays is highly subjective and image-enhancement algorithm specifications continue to evolve. As a result of this evolution, Xilinx FPGAs and reference designs enable you to modify your image enhancement algorithms quickly and easily.

Because the Xilinx Spartan series has been broadly adopted in high-volume consumer applications such as DVD players, plasma displays, and HDTVs, you can be assured of a highly reliable and readily available supply of parts for your consumer application. Xilinx will continue to add new market-specific features to keep pace with new standards and display reference designs as it continues to lower costs.

For more information about Xilinx reference designs and how to obtain them, visit *www.xilinx.com/esp/consumer/display/*.

Display Panels	Video Boards
7:1 LVDS Rx	Adaptive Dynamic Gamma Correction
Timing Controller (TCON)	Motion Artifact Estimation and Compensation
Multiple I/O Standards – LVDS, RSDS, Mini-LVDS	Favorite Color Transformation (FCT)
Precise Gamma Correction	Brightness and Contrast Adjustment
Image Dithering	Hue and Saturation Adjustment
Color Temperature Adjustment	Sharpness Enhancement
Dynamic False Contour Reduction	Motion-Adapted Temporal Noise Reduction

Table 1 - Digital display panel and video board reference designs

High-Performance Image Processing on FPGAs

You can develop and implement sophisticated imaging modules on low-cost devices.

By Michael Tusch
CEO
Apical Limited
mtusch@apical-imaging.com

The rapid evolution of digital imaging technology, accompanied by huge market demand for cameras and displays in both consumer and industrial segments, presents a significant challenge to device developers who want to create high-quality products. To extract the best possible images from capture and display hardware, sophisticated image processing algorithms are available. But their implementation is limited by several factors: the intrinsic complexity of the algorithms; the pressure to reduce bill of materials costs; the need to support a wide variety of developing formats; and the frequent requirement to customize particular device environments.

The development and licensing of image-processing IP is central to Apical's business. Addressing the factors I have described in the preceding paragraph is therefore key to our company's success. If the design of a new IP core is hindered by difficulties in coding and verifying RTL, or if a customer has no straightforward way to evaluate and implement the IP in their products, then the initial enthusiasm of an exciting research result rapidly evaporates.

FPGAs have proven to be highly effective in enabling our company to shorten time to market for our IP, especially in video applications. By using FPGAs extensively in IP development, time to market is reduced, IP quality increases, and a ready route to commercialization is possible. In this article, I'll explore these benefits in the context of image enhancement technology for display applications.

Image Processing for Flat-Panel Displays

The image quality of a display is driven by its dynamic range and color gamut. Dynamic range is essentially the difference in brightness between the darkest and brightest image content that can be rendered under particular ambient lighting conditions. Color gamut represents the range of color space that can be accurately reproduced. In comparison to traditional cathode ray tubes (CRTs), flat panels have certain limitations in both of these key characteristics. In particular, they have a lower dynamic range and particular problems in dark regions where true blacks are difficult to produce.

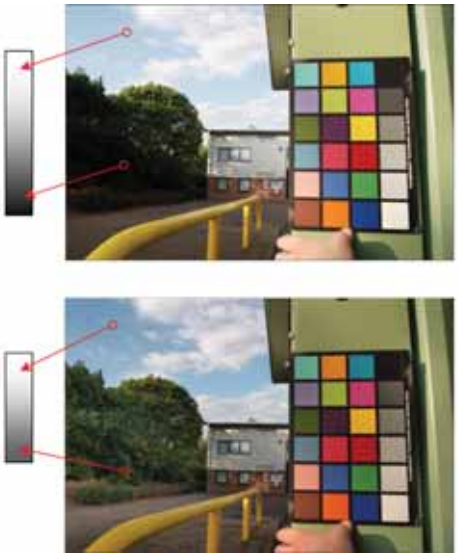


Figure 1 – Effect of space-variant dynamic range compression

The standard technology for adjusting input video imagery to the dynamic range of the display is gamma correction, or some modification where the image processing pipeline incorporates a finely tuned non-linear gain curve. However, a significant dynamic range adjustment leads to a degradation of image quality, with a loss of contrast and distortion of color.

A much better approach is to use an adaptive, space-variant approach. This applies different dynamic ranges and color correction in unique regions of the frame. Regions well within the display's capabilities are unaltered, but those outside can be drawn in. With Apical's iridix technology, a different non-linear correction is applied to each individual pixel based on a statistical analysis of the input frame. The effect of this algorithm on an image with a wide dynamic range is shown in Figure 1. Effectively, the algorithm is applying a different, optimized gamma curve to every pixel in every frame of the video signal.

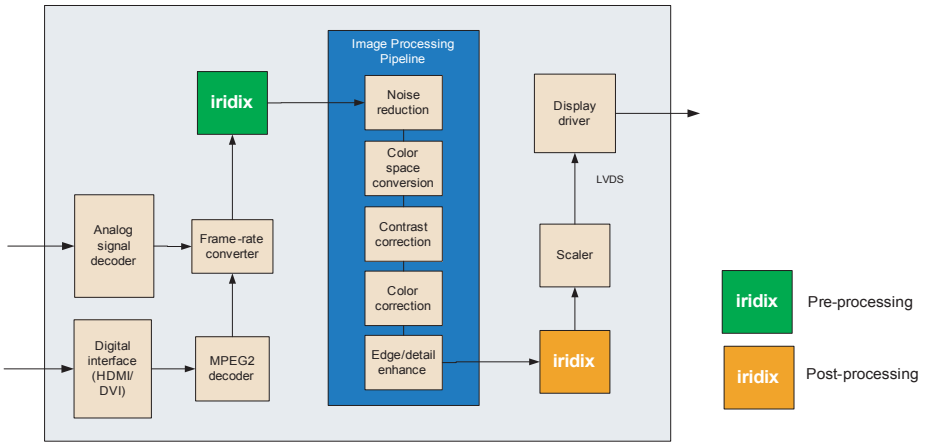


Figure 2 – Typical display pipeline showing options for pre- or post-processing through an FPGA.

Incorporating such an algorithm into a display image-processing pipeline, as shown in Figure 2, can dramatically improve the final image quality. A display equipped with such processing can render images well even under very bright ambient lighting, while retaining perfect colors and contrast under ideal viewing conditions.

In fact, you can perform this processing at any stage, on source YUV data or processed RGB. It is often most convenient to include it either before (pre-processing) or after (post-processing) the standard display pipeline. In practice, you can achieve the best results at post-processing.

Such an algorithm is much more demanding than the traditional technique. It is a two-pass process, mathematically complex, requiring high-precision operations (at least 12 bits) and high-speed calculations to transform streaming video data and handle large frames (HDTV) in real time.

In addition to algorithmic complexity, other development challenges are incorporating a continuous flow of new results from R&D, supporting many different video formats, and addressing customization requirements.

FPGAs as a Platform for Image Processing

Traditionally, the route to implementing image-processing functions in consumer devices is through an ASSP or a DSP. Both of these routes remain valid, and in some cases are still optimal. But their limitations are well-known: ASSPs are inflexible, expensive, and time-consuming to develop; powerful DSPs are costly and their corresponding software applications may not match the performance of hardware. The attractiveness of FPGAs derives from the combination of the virtues of the alternatives.

The flexibility of the FPGA is of particular importance to Apical, as both the image processing algorithms and their target display applications are evolving rapidly. Time to market is a crucial factor in all consumer applications; this must be shortened as much as possible without compromising the quality of the product.

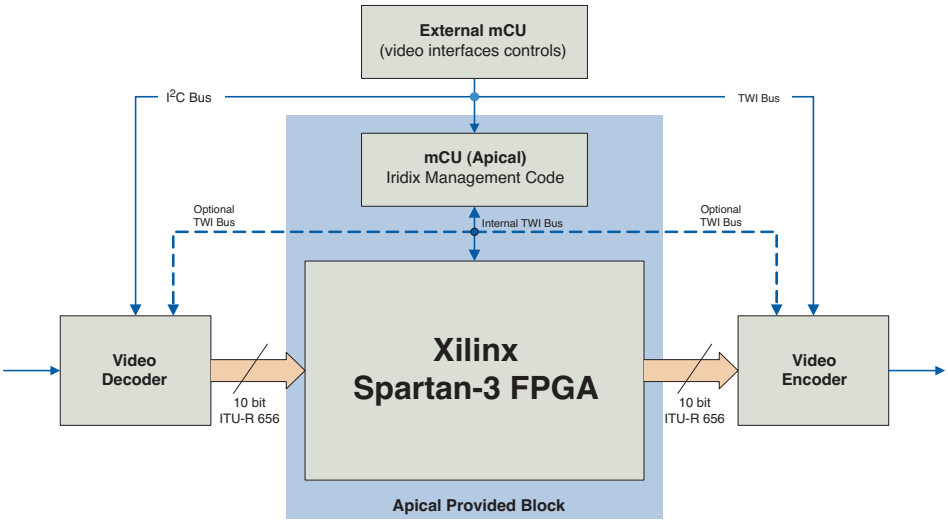


Figure 3 – Example Spartan3-based architecture for stand-alone video processing

IP Requirement	FPGA Requirement
Logical Complexity	Around 5,000 Slices or More
Real-Time Pixel-by-Pixel Transformations	Embedded Hardware Multipliers
Storage of Look-Up Tables and Statistics	Embedded Memory
Time to Market	Good Development Environment

Table 1 – FPGA characteristics required by Apical's IP

Property	Spartan X3S1500-4
Slices	4,379
Logic	28% of Logic 40% of Multipliers
Memory	17% of Memory
Clock Frequency	75 MHz

Table 2 – Statistics of Apical's "iridix" core on Spartan-3 FPGAs

Given the complexity of the algorithms, it is simply not possible to test a behavioral model exhaustively over hundreds of hours of video footage, which is necessary to analyze image quality and detect artifacts. A single frame may take several minutes to process. Therefore, the importance of implementing the design in hardware as soon as possible begins conflicting with the traditional approach of full simulation before implementation.

In addition, flexibility is critical because the IP may require customization as part of a display optimization process. Although there are standards that govern many aspects of video formatting in display applications, it is neither possible nor commercially attractive to attempt to standardize image quality. The value of the IP therefore increases with flexibility and customizability.

To make practical use of the flexibility of FPGA architectures, the device must meet the performance requirements of the application. For Apical's primary algorithms, these are listed in Table 1. Apical's principal dynamic range compression IP is significantly more complex than most image pipeline IP, but nevertheless a device such as a Xilinx® Spartan™-3 FPGA has all of the necessary resources. Even HDTV video can be processed at 60 Hz on a relatively inexpensive Spartan-3500E FPGA.

In achieving Apical's time-to-market objectives, the availability of a good development environment is crucial. Xilinx ISE™ software provides such an environment, and the recently announced Spartan-3 Display Solutions Board provides a convenient development and demonstration platform for Apical's IP.

FPGAs in Display Products

Until relatively recently programmable logic devices would not have been considered for the implementation of complex video IP in a device such as a flat-panel display, leaving ASSP the only option. However, FPGA technology has developed rapidly; low-cost devices such as Spartan-3 FPGAs are now well suited to such applications.

The performance/cost ratio of Spartan-3 FPGAs is such that they present an alternative to ASICs in numerous applications, including flat-panel displays. They provide a ready solution to a device developer's dilemma that "we need to include this new IP, but we don't want to increase our design risk or time to market." You can update the FPGA bitstream as easily as embedded software and in addition customize it to fit optimally with the panel characteristics and the rest of the display imaging pipeline.

You can conveniently control algorithm parameters through an I²C-compatible (two-wire) interface using a simple 8-bit microcontroller, as shown in Figure 3.

Given enough logic, the FPGA can deliver multiple, tailored image enhancements alongside a standard ASSP, handling additional functions such as decoding, deinterlacing, and scaling.

Conclusion

Dramatic progress has been made in image processing R&D over the last five years, with the potential for displays to deliver unprecedented visual quality even within a climate of strong price pressure. In a device build, the imaging pipeline has by far the highest contribution to the final appearance in comparison to its modular cost.

Spartan-3 devices present a new alternative to ASICs for imaging device makers to augment their products using image-enhancement IP such as that provided by Apical.

For more information about the IP and its implementation in FPGAs, visit *www.apical-imaging.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.

Intellitech®

Fibre Channel Solutions from Xilinx

Using the Virtex family and Xilinx Fibre Channel IP, you can build a complete FC solution running as fast as 4 Gbps.

by Douglas Grant
Staff Design Engineer
Xilinx, Inc.
douglas.grant@xilinx.com

The Fibre Channel (FC) standard enabled IT managers to scale compute and storage needs independently. Unlike protocols used for direct attach like SCSI, FC's switch-based architecture has enabled it to become a major player in the data center.

The advantages of FC include:

- High speed: 1 Gbps, 2 Gbps, 4 Gbps, 8 Gbps and 10 Gbps
- Low latency and high data integrity
- Upper-level protocol support: SCSI, FICON
- Distance support: >10 Km

Xilinx® technology has been used in the development of hubs, switches, and high-end enterprise storage arrays. In this article, I'll summarize FC protocol and discuss Xilinx LogiCORE™ IP solutions for Fibre Channel.

Principal Features of Fibre Channel

Figure 1 shows the relationship between Fibre Channel-defined layers FC-0 to FC-4 and the well-known ISO-OSI seven layer model. FC-0 deals with the physical transmission and reception of serial data, FC-1 refers to the encoding and synchronization, and FC-2 comprises framing and sequencing tasks and login protocols. The FC-3 layer has never been defined. FC-4 refers to the mapping of upper-layer protocols (ULPs) into Fibre Channel frames. An example of FC-4 functionality is the FCP standard, which maps SCSI commands into FC frames.

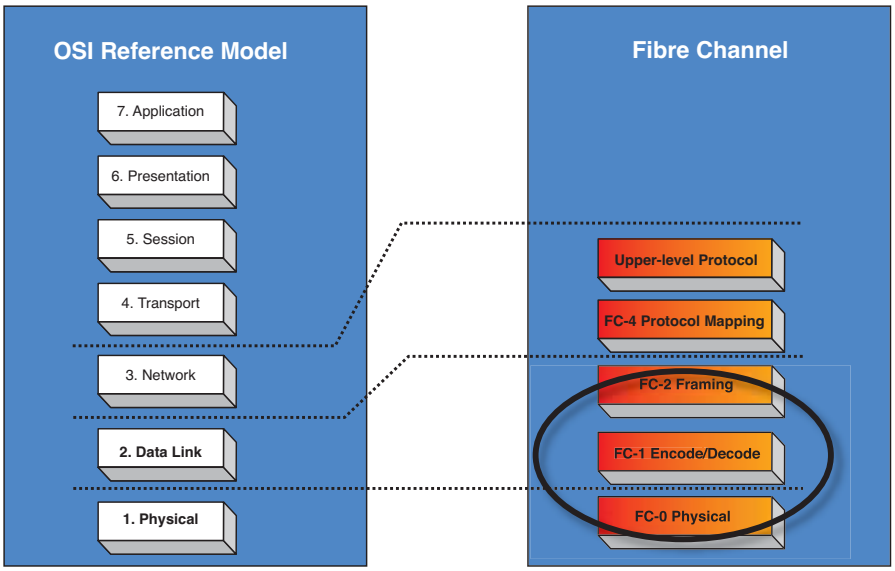


Figure 1 – OSI layers versus FC layers

Fibre Channel data is 32 bits wide at the user interface. These 32-bit words are encoded using 8b10b encoding into 40 bits in the serial domain and typically – although not exclusively – transmitted through optic fiber. Although 32-bit data words are by definition arbitrary values, Fibre Channel defines a set of so-called primitive signals, or ordered sets of specific byte values, that implement the initialization protocols. Examples of primitive signals include IDLE and RRDY.

Primitive sequences are defined as a sequence of three or more consecutive primitive signals such as LR (link reset) and LRR (link reset response). Fibre Channel also defines a number of SOF (start of frame) and EOF (end of frame) primitives that delineate the Fibre Channel frames.

Buffer-to-buffer credit is a Fibre Channel concept that allows the best use of the bandwidth available, enabling a Fibre Channel port to transmit a frame to another port only if there are receive buffers available at the other port.

Fibre Channel frames are constructed from an SOF, six header words, a payload of up to 2,112 bytes, and a cyclic redundancy check (CRC) and EOF word. Frames are separated by an inter-frame gap (IFG) of at least six so-called fill words, which are typically IDLE words but can also include RRDYs and other credit management codes.

Virtex-II Pro and Virtex-4 FPGAs

Three main features of Virtex™-II Pro and Virtex-4 FX devices allow you to create entire Fibre Channel ports on a single die:

- Multi-gigabit transceivers (MGTs) allow the reception and transmission of serial data up to 11 Gbps. For Fibre Channel, 1 Gbps, 2 Gbps, and 4 Gbps are currently defined in the standard, with 8 Gbps just around the corner.
- The embedded PowerPC™ 405 core on both Virtex-II Pro and Virtex-4 FX devices is an ideal candidate for running Fibre Channel port software, which includes fabric and port login/logout, exchange of link parameters, and other FC-2 functions that lend themselves to implementation in firmware.
- The abundance of block RAMs on the Virtex-II Pro and Virtex-4 FX devices aids the processing of multiple frames at line rates. Any Fibre Channel port will require a number of receive buffers, and these may be implemented on-chip in block RAMs.

LogiCORE IP for Fibre Channel

Xilinx currently offers 1, 2, and 4 Gbps Fibre Channel cores in two flavors: point-to-point and arbitrated loop.

Point-to-Point LogiCORE Solutions

This core supports FC-1 and the framing and signaling part of FC-2, based on the ANSI INCITS FC-FS Final Draft v1.9 standard.

Included are the port state machine (PSM) and loss of sync state machine (LOS-FSM). All classes of Fibre Channel frames are supported. Buffer-to-buffer credit management is optionally supported, along with statistics gathering. Access to the control signals is through an optional register bank with a microprocessor-agnostic management interface or, alternatively, through dedicated configuration control and configuration status vectors.

You can use this core to create any of the following Fibre Channel ports: N, F, E, and B. Arbitrated loop support is not included in this core, nor are login protocols directly supported within – these are left for you to implement in either firmware or hardware as required.

Speed negotiation (Section 28 of FC-FS) is optionally supported in hardware for multi-speed cores. You can alternatively implement this in firmware.

Figure 2 shows the architecture of the dual-speed core.

Credit Management

The optional credit management block provides simple buffer-to-buffer credit management, keeping track of RRDY and SOF primitives received and sent and supporting BB_SCx-based credit recovery as defined in FC-FS Section 18.5.11.

MAC

The MAC block, designed to FC-FS Section 7, provides the main functionality of the core. This block comprises the port state machine (PSM) as well as the framing control and checking of the data.

Link Controller

The link controller block, designed to FC-FS Sections 5 and 6 (equivalent to FC-PH

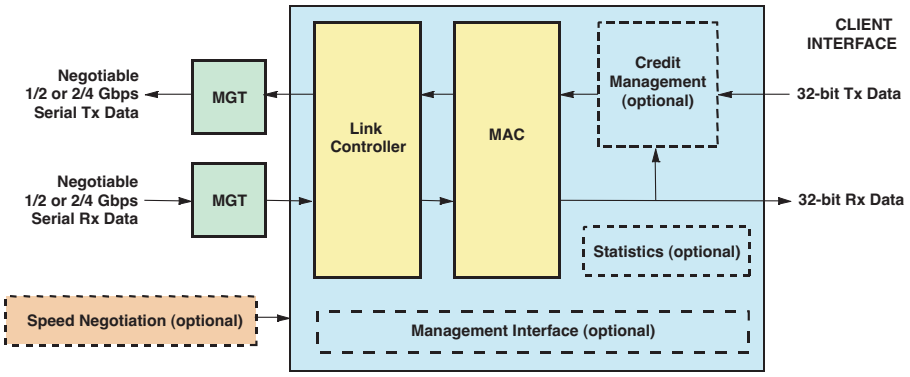


Figure 2 – Point-to-point FC LogiCORE architecture (dual speed on Virtex-II Pro FPGAs)

Parameter	Virtex-II Pro FPGA			Virtex-4 FPGA		
	Slices[1]	Total LUTs	FFs	Slices	Total LUTs	FFs
c_has_host=false c_has_bbcredit_mngmt=false c_has_speed_neeg=false c_has_state=false	1,040	1,270	970	1,040	1,270	980
c_has_host=true	+300	+390	+270	+300	+390	+280
c_has_stats=true[2]	+220	+490	+260	+210	+470	+250
c_has_bbcredit_mngmt=true	+420	+770	+190	+400	+700	+190
c_has_speed_neg=true	+120	+210	+100	+110	+200	+100

1 To discover the total resource requirements, begin with the numbers in the top row and add the numbers for the selected options.
2 Only available if c_has_host option has also been selected.

Table 1 – Resource utilization for FC point-to-point core

Sections 11 and 12), provides word alignment and synchronization of incoming data. This block also provides CRC generation and checking on outgoing data.

Statistics

The optional statistics block collects and stores statistical information in the memory of the core. Statistics may still be collected outside of the core using the statistics vector.

Speed Negotiation

The optional speed negotiation block provides the ability for a dual-speed core to implement the FC-FS Section 28 speed negotiation algorithm.

Core Generation

The core may be generated to work at the following speeds: 1 Gbps, 2 Gbps, 4 Gbps, 1-2 Gbps multispeed, and 2-4 Gbps multispeed. The 4 Gbps-capable cores are available only for Virtex-4 FX devices, while all other speed configurations are available for both Virtex-4 FX and Virtex-II Pro devices.

Table 1 shows the device utilization for the FC point-to-point LogiCORE IP.

Arbitrated Loop LogiCORE Solutions

This core supports FC-1 and the framing and signaling part of FC-2, based on the ANSI INCITS FC-FS Final Draft v1.9 and ISO/IEC FDIS 14165-122 FC-AL-2 standards.

Included within the core are the initialization state machine, based around a MicroBlaze™ soft processor; the arbitration state machine, known as the "old port" state machine to allow fallback to point-to-point operation; and the loss of sync state machine (LOS-FSM). All classes of Fibre Channel frames are supported. Both alternative and buffer-to-buffer credit management are supported, along with statistics gathering through a statistics vector. Access to control signals is through a register bank with a microprocessor-agnostic management interface or, alternatively, through a dedicated configuration vector.

You can use this core to create any of the following Fibre Channel ports: NL, FL, N, F, E, and B. It should be noted that for non-arbitrated loop applications, the point-to-point core will be a much more efficient solution. No login protocols are directly supported within the core – these are left for you to implement in either firmware or hardware as required.

A fast port interface to the core allows automated arbitration/open circuit/frame transmission/close circuit processes with very little interaction from the user interface.

Figure 3 shows the main blocks within the core.

FC-1 Layer

The FC-1 layer crosses clock domains from the RocketIO™ transceiver recovered receive clock and performs clock correction on the ordered sets allowed in the FC-AL-2 standard.

CRC

The CRC block generates and verifies CRC-32 blocks on outbound and inbound frames.

Loop Initialization Controller

The loop initialization controller controls all loop initialization functions, including exchange of loop initialization frames with other ports on the loop and address assignment. It is implemented using a MicroBlaze soft processor.

Loop Port State Machine

The loop port state machine (LPSM) controls core operation such as arbitration, circuit opening, and circuit closing when the core is placed in an arbitrated loop system.

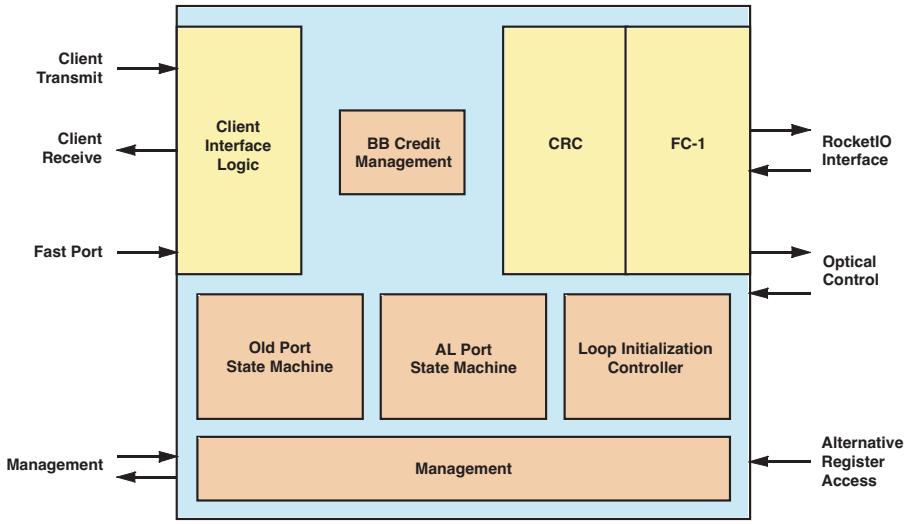


Figure 3 – FC-AL LogiCORE architecture

Old Port State Machine

Whenever the Fibre Channel arbitrated loop core fails to initialize as a loop port, it falls back to Fibre Channel point-to-point operation. The old port state machine controls core operation in point-to-point mode.

Fast Port Interface

The fast port interface manages the opening and closing of circuits in an arbitrated loop system. The core arbitrates for control of the loop by supplying the address of the destination loop port, then signaling back to the user logic when control is obtained. The availability of this fast port feature simplifies software development requirements by taking care of these functions for you.

Management Interface

Configuration of the core is performed through the management interface, a 32-bit processor-neutral control pathway independent of the data path of the core.

Credit Management

The credit management block provides both simple and alternative buffer-to-buffer credit management, keeping track of RRDY and SOF primitives received and sent. In point-to-point mode, the credit management block supports BB_SCx credit recovery as defined in FC-FS Section 18.5.11.

Core Generation

The core may be generated to work at the following speeds: 1-2 Gbps multispeed and 2-4 Gbps multispeed. The 2-4 Gbps-capable core is available only for Virtex-4 FX devices, while the former speed configuration is available for both Virtex-4 FX and Virtex-II Pro devices.

Table 2 shows the device-utilization for the FC-AL cores.

Virtex-II Pro or Virtex-4 FPGAs			
Slices	Total LUTs	FFs	Block RAMs
4,600	6,800	3,300	22

Table 2 - Resource utilization for FC-AL core

Bridge Port Reference Design

To provide an example of how the Fibre Channel cores can be integrated into a complete Fibre Channel port, Xilinx created a comprehensive Bridge (B) Port Reference Design for Virtex-II Pro devices, which is available at no cost.

The B Port Reference Design is based on an EDK project and includes:

- Fibre Channel point-to-point LogiCORE IP (v2_0)
- B Port-specific back-end hardware for automated end-end credit frame generation (ACK_1) and F_BSY/F_RJT generation
- Optional framer to encapsulate raw data in pre-programmed SOF/header and EOF
- Optional pattern generator that supports CJPAT, CRPAT, CSPAT, and a user-defined pattern
- B Port-specific back-end hardware to filter frames before login
- PPC405-based firmware, including drivers for FC core and B Port-specific hardware and ELP (exchange of link parameters) protocol implementation
- Data multiplexing to allow TX/RX without the involvement of the PPC405 subsystem
- ChipScope™ modules for enhanced debugging
- UART connecting to user-PC through RS232 to allow real-time user interaction with design
- Comprehensive documentation covering everything from register maps to adding new features

Conclusion

With Virtex-II Pro and Virtex-4 FX devices and Fibre Channel LogiCORE IP, you can quickly create an FC infrastructure including switches, bridges, and end points. Up to 4 Gbps is supported today on Virtex-4 FX FPGAs.

The cost-effective FC point-to-point and arbitrated loop cores can deliver 100% throughput. The zero-cost B Port Reference Design with an evaluation-licensed LogiCORE IP and an ML323 development platform offers a risk-free evaluation environment that you can connect to existing FC ports or networks. It also offers a strong basis for the creation of further examples of Fibre Channel ports.

For more information and technical details on Xilinx IP cores for Fibre Channel, visit *www.xilinx.com/storage.*

Utilizing Virtex-4 FPGA-Immersed IP for Hardware Acceleration for RAID 6 Parity Generation/Data Recovery

FPGAs and immersed IP blocks increase data reliability and availability in storage systems.

by Matt DiPaolo
Sr. Product Applications Engineer
Xilinx, Inc.
matthew.dipaolo@xilinx.com

Steve Trynosky
Product Applications Manager
Xilinx, Inc.
steve.trynosky@xilinx.com

A redundant array of independent disks (RAID) is a hard disk drive (HDD) array where part of the physical storage capacity stores redundant information. This information can be used to regenerate data if there is a failure in one or more of the arrays' member disks (including a single lost sector of a disk), or in the access path to the member disk.

There are many different levels of RAID. The implemented RAID level depends on several factors, including overhead of reading and writing data, overhead of storing and maintaining parity, plus the mean time to data loss (MTDL). The most recent level is RAID 6, which can have two implementations (Reed-Solomon P+Q or Double Parity). RAID 6 is the first RAID level that allows the simultaneous loss of two disks, which improves its MTDL over RAID 5.

In this article, we'll summarize the Xilinx® reference design for RAID 6, illustrating advantages of immersed hard-IP blocks of the Virtex™-4 architecture, including block select RAMs, distributed memory, FIFO memory, digital clock managers (DCM), the PowerPC™ PPC405, and DSP48 blocks. Our hardware acceleration block supports Reed-Solomon RAID 6 and can support other RAID levels when coupled with appropriate firmware.

Why RAID 6?

In RAID levels such as RAID 5, when a disk failed system firmware would use the remaining disks to regenerate the data lost from the failed disk. If another disk failed before regeneration was complete, the data was lost forever, which becomes the point of MTDL. Up to now, because of the size and probability of disk failure, the MTDL of RAID 5 was acceptable.

With the rising popularity of inexpensive disks such as Serial ATA and Serial Attached SCSI (SAS), as well as larger capacity disks, the mean time between failures (MTBF) of a disk has increased dramatically. For example, 50 disks, each with 300 GB capacity, would have an MTBF of 5 x 10^5 hours (a 10-14 read error rate), resulting in an array failure in less than eight years. According to Intel's TechOnline, "High-Performance RAID 6: Why and How," RAID 6 improves this to 80,000 years. Achieving a large MTDL justifies the increased overhead of both disk space for additional parity data, additional reads and writes to disk drives, and the system complexity required for handling multiple disk failures.

Another aspect of RAID 6 is the method in which data and redundancy information is stored on multiple disks. Figure 1 shows an example of a five-disk system. Data and parity information is stripped horizontally across the drives in blocks of data. Each block is typically a multiple of 512 bytes. To keep the parity drives from being a system bottleneck, as can occur in RAID 4, the parity information rotates around the drives, in integer increments of blocks of data. Note that the five-disk case has a 40% overhead for parity; larger disk arrays can reduce this to 16% in a 12-disk system.

Mathematics Behind the Technology

To understand Reed-Solomon RAID 6, you must have some familiarity with Galois Field (GF) mathematics. We will summarize the GF mathematics involved in RAID 6 implementation here, but for a more detailed description, please refer to:

- Xilinx XAPP731 (*www.xilinx.com/bvdocs/appnotes/xapp731.pdf*)
- Intelligent RAID 6: Theory, Overview and Implementation (*www.intel.com/design/storage/papers/308122.htm*)
- A Tutorial on Reed-Solomon Coding for Fault-Tolerance in RAID-like Systems (*www.cs.utk.edu/~plank/plank/papers/CS-96-332.html*)
- The Mathematics of RAID 6 (*http://kernel.org/pub/linux/kernel/people/hpa/raid6.pdf*)

GF mathematics define that any calculation continues to have the same number of bits as the two operands from which it was generated (that is, two 8-bit numbers using GF multiplication result in an 8-bit number). Here are the definitions of GF multiplication and division. Note that the addition/subtraction is regular integer addition/subtraction.

$2 \otimes 8 = gfilog[gflog[2] + gflog[8]] = gfilog[1 + 3 = gfilog[4] = 0x10$

$0xd \div 0x11 = gfilog[gflog[0xd] - gflog[0x11]] = gfilog[0x68 - 0x64] = gfilog[4] = 0x10$

The reference design implements the gflog and gfilog values using the polynomial $x^8 + x^4 + x^3 + x^2 + 1$. Lookup tables, stored in block RAM inside the FPGA, are placed in the data path to provide fast access, enabling calculations at DDR memory speeds.

RAID 6 Parity (P & Q) Equations

To recover from two disk failures, RAID 6 stores two unique parity values, P and Q. This second parity creates a second equation to allow you to solve for two unknowns (or disk failure points).

The equations discussed next assume a RAID 6 disk array comprising three data disks and two parity disks for each block of data to simplify the discussion. These equations extend to 255 disks, including the two parity disks, which is the mathematical limit of the equations.

The first is the P parity, which is identical to RAID 5. A simple exclusive or (XOR)

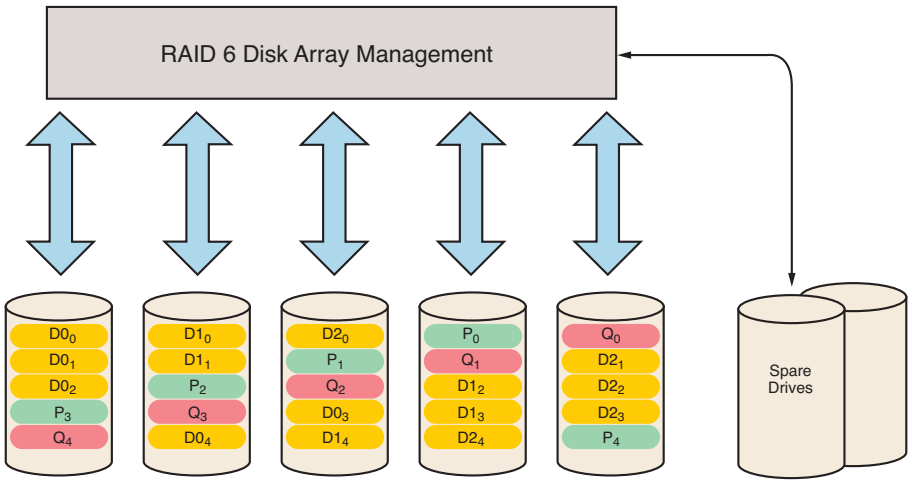


Figure 1 – RAID6 data block organization

The FPGA contains a SATA protocol controller that interfaces with the memory controller and PowerPC405.

function generates the parity block from the data values in the same sector horizontally across the data drives in an array group. In Figure 1, P_0 XORs the $D0_0$, $D1_0$, and $D2_0$.

$$P_N = D0_N \oplus D1_N \oplus D2_N \oplus \cdots \oplus D(M-1)_N$$

- N = 0 to the maximum number of blocks (sectors) on the disk drive
- M = number of data disks in the array group

The equation for the first sector of a three-data-drive system is:

$$P_0 = D0_0 \oplus D1_0 \oplus D2_0$$

In RAID 6, Q parity assigns a GF multiplier constant associated with each data disk drive. The constant applies only to the data blocks stripped horizontally across the array, not to each drive in the array. Each data block is GF multiplied by a constant, before adding to the data elements of the next data drive. The g constants are determined from the GFILOG table. If another drive was added to the array, then g3 = gfilog(3) = 0x8.

$$Q_N = (g0 \otimes D0_N) \oplus (g1 \otimes D1_N) \oplus \cdots \oplus (g(M-1) \otimes D(M-1)_N)$$

Here is the equation for the third sector of a three data drive system:

$$Q_2 = (0x1 \otimes D0_2) \oplus (0x2 \otimes D1_2)\ (0x4 \otimes D2_2)$$

FPGA-Based Hardware Acceleration Solution

The Xilinx reference design for RAID 6 is based on a sample configuration. In this case, the system contains a RAID host controller on an ML405 demonstration board. This board contains a Virtex-4 FPGA along with DDR memory and Serial ATA (SATA) connectors attached to a port multiplier, which in turn connects to five SATA HDDs.

The FPGA contains a SATA protocol controller that interfaces with the memory controller and PowerPC405. It is possible to replace the SATA protocol controller with Serial Attached SCSI (SAS), Fibre Channel (FC), or any other disk interface protocol depending on the overall system architecture.

The design concentrates on the hardware acceleration portion of a RAID 6 system (depicted in Figure 2). An embedded PowerPC405 block controls the RAID 6 hardware, setting up pointers to the data and parity blocks of memory as well as setting up the hardware. The reference design doesn't include the protocol link between the memory and the HDDs, as the possibilities here are endless.

There is a microprocessor to run the RAID firmware and a RAID hardware accelerator block. This firmware also generates typical data that would come from some type of HDD connection. The PPC405 firmware creates example data blocks in the DDR memory to emulate disk sectors retrieved from a disk array

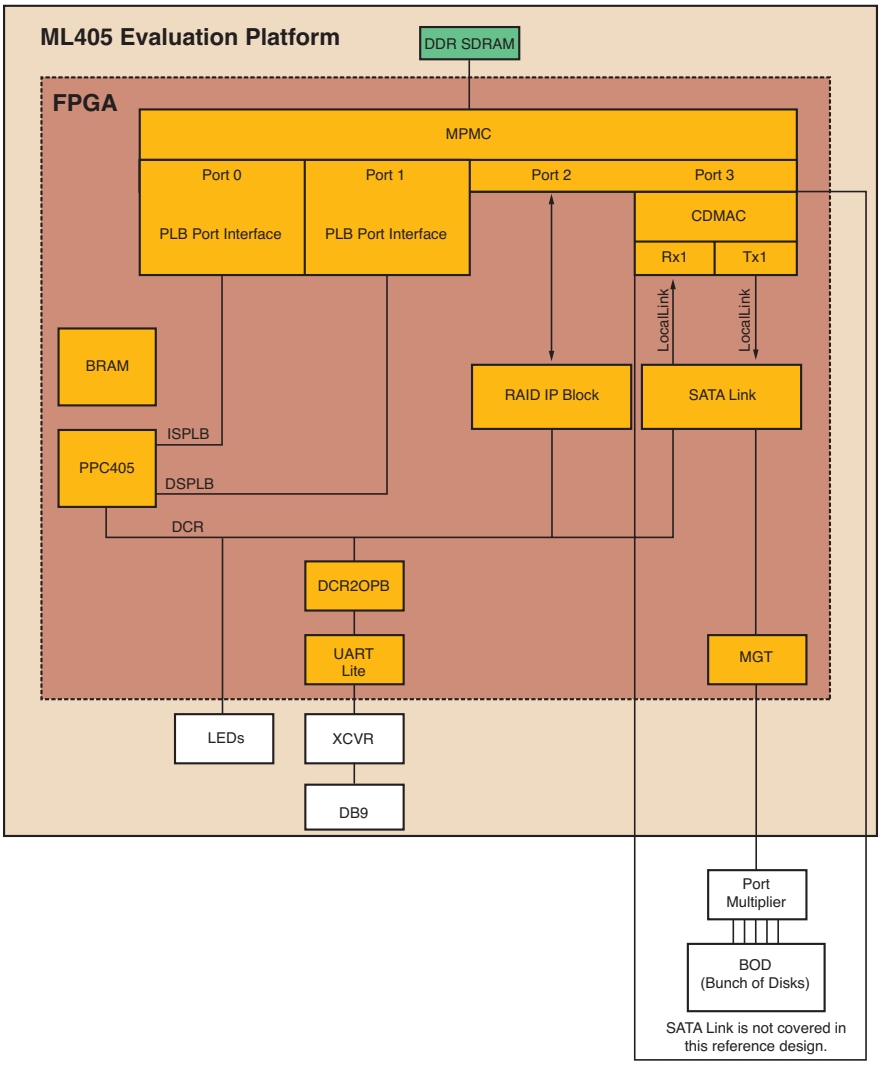


Figure 2 – Theoretical typical system

system. These data blocks are then used by the RAID accelerator to generate P and Q parity blocks.

A Xilinx multiple port memory controller (MPMC) controls the DDR memory. The MPMC provides multiple memory masters to access a shared memory. The MPMC connects to each of the PowerPC405 instruction and data-side processor local bus interfaces. Two other ports provide system-specific implementations for this reference design; one is used for the RAID hardware accelerator and the other is available for access to the hard disk. For additional information on the MPMC, refer to XAPP535, "High-Performance Multi-Port Memory Controller" (*www.xilinx.com/bvdocs/appnotes/xapp535.pdf*).

A RAID 6 hardware accelerator is mathematically intensive. The RAID 6 calculations require each block of data to be stored in a memory buffer and then read into a temporary data buffer while being XOR'ed or added to other data elements. There is a large amount of data manipulation that needs to be done. This manipulation is also time-intensive, but hardware implementations are faster than processor-only register calculations because hardware can provide parallel manipulation of data blocks, clock rate lookup table access, and multipliers that operate much faster than a processor alone. This tends to be a small portion of the overall period when including the seek time of the disks themselves.

The hardware accelerator comprises four main blocks:

- Data manipulation block (DMB) – this block performs the mathematical operations on one byte of data (depending on the system data width, more DMB logic can be added to support larger data widths). It can create parity and regenerate data of any case discussed in the RAID 6 equations.
- RAID FSM – the main logic of the reference design. It controls the MPMC_IF block along with all of the DMB signals, including block RAM address, read, write control, and MUX select lines.
- DCR FSM – communicates with the PowerPC405 device control register (DCR) bus.
- MPMC_IF – Controls the native interface of the MPMC block to transfer data. The MPMC is a four-port DDR memory controller time-shared with the processor instruction and data cache processor local bus (PLB) interfaces.

Firmware plays a major role in all RAID systems. However, the hardware changes little for the different RAID levels. The hardware accelerator can remain the same while the firmware changes to incorporate the different ways the data is organized on the HDD, and how the parity is generated.

The hardware accelerator can support other RAID levels beyond RAID 6 with minimal (if any) modifications. RAID Double Parity (DP), RAID 5, RAID 4, and RAID 3 are among the supportable RAID levels.

Conclusion

The Xilinx reference design for RAID 6 incorporates advantages of immersed hard-IP blocks of the Virtex-4 architecture, including block select RAMs, PowerPC PPC405, and DSP48 blocks in a hardware acceleration block that supports Reed-Solomon RAID 6 and can support other RAID levels when coupled with appropriate firmware.

The block RAMs are used for the GF mathematic look-up tables, the DSP48 blocks perform fast integer addition, and the PowerPC405 handles memory/address management, plus the potential to handle RAID-level data structures in HDD arrays. As in all systems, you must evaluate hardware and firmware trade-offs when architecting redundant systems.

For more information about the RAID 6 reference design, visit *www.xilinx.com/bvdocs/appnotes/xapp731.pdf* or e-mail *storage@xilinx.com*.

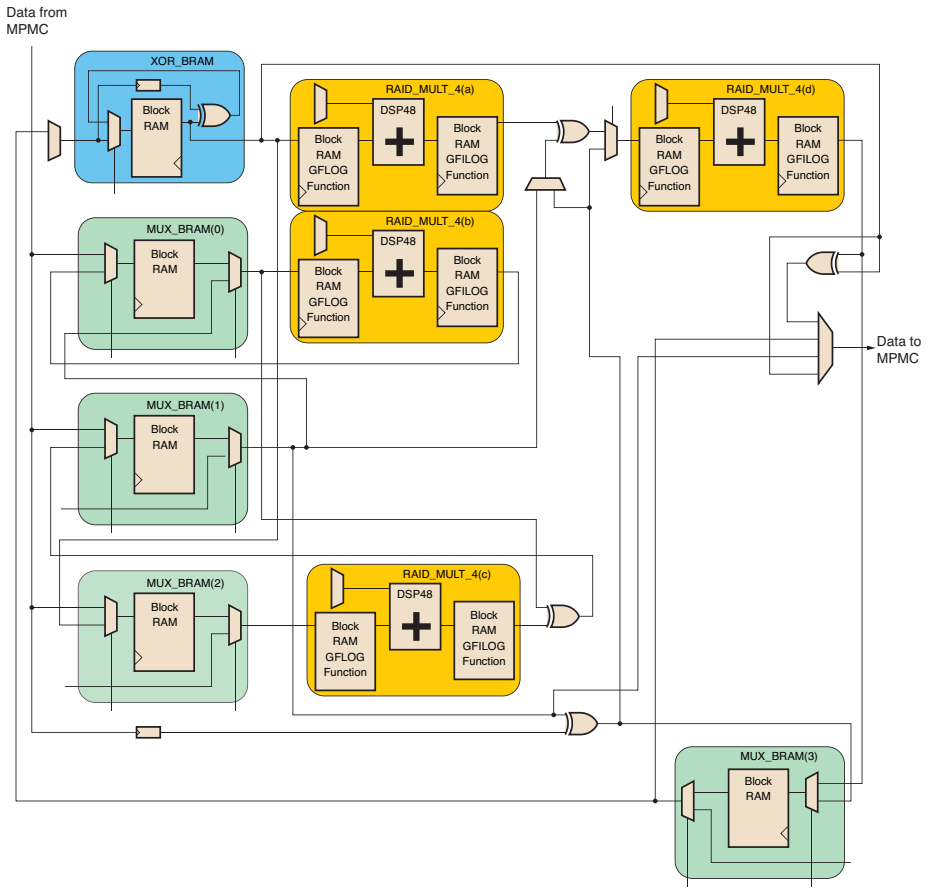
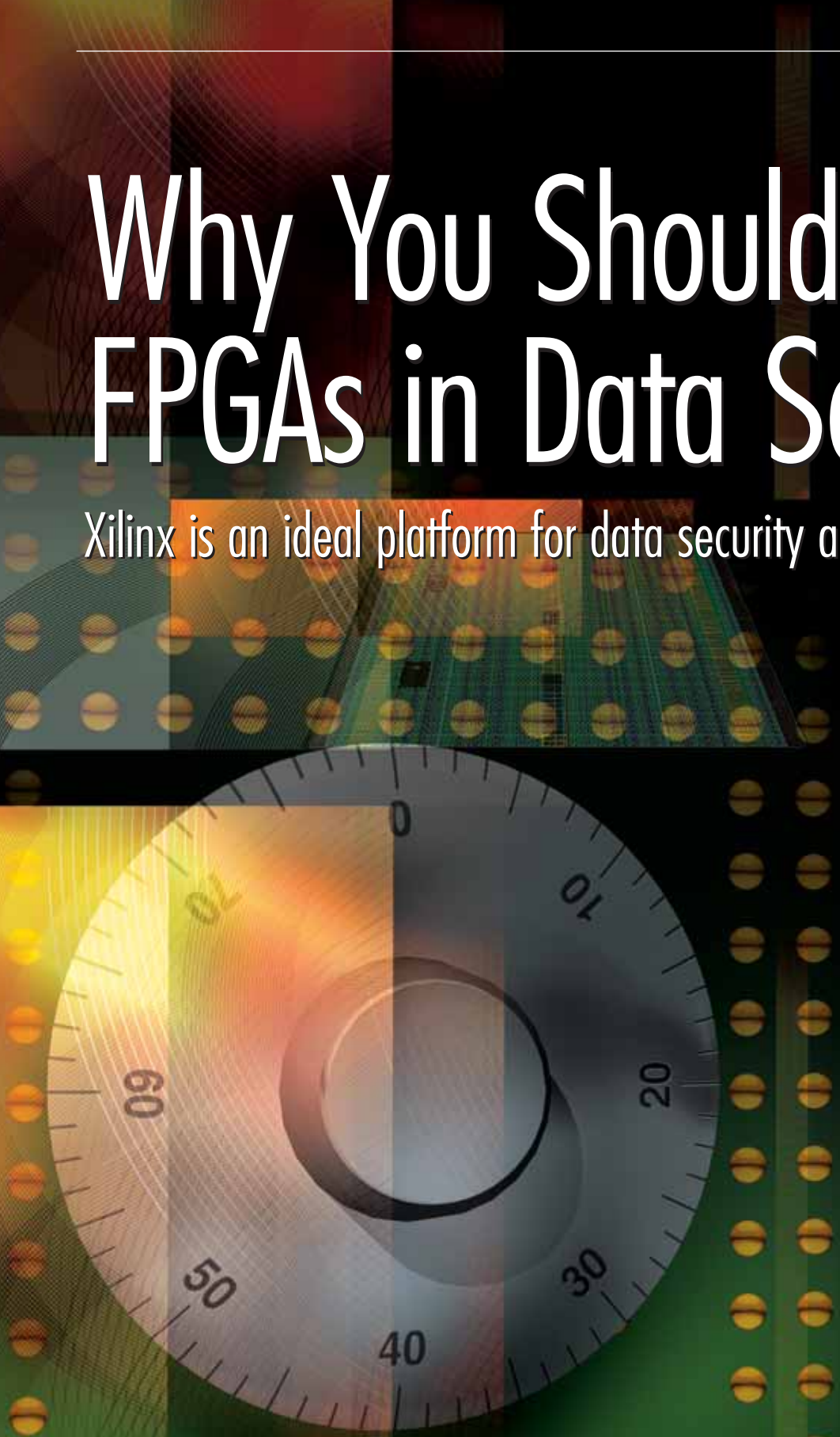


Figure 3 – Data manipulation block logic

Why You Should Use FPGAs in Data Security

Xilinx is an ideal platform for data security applications.



by Mike Nelson
Senior Manager, Storage and Servers, Vertical Markets
Xilinx, Inc.
mike.nelson@xilinx.com

Data security is fast becoming a requirement in communications and enterprise infrastructures. Secure electronic commerce is almost doubling every year. New regulations are mandating the retention and protection of ever more information (Sarbanes-Oxley, HIPAA). Legal liability is dramatically escalating for those who manage such data carelessly. And finally, the value of data as a corporate asset itself is growing in such forms as fully electronic product designs, customer databases, and supply chain management systems. All of these trends make data security a mandatory element in almost any new system architecture.

But the implementation of data security faces a number of serious challenges:

- Performance requirements vary widely
- System cost pressures remain high
- Standards vary widely and are continuously evolving
- Management becomes an integral aspect of the data security landscape as it evolves into a part of the managed IT infrastructure

In this article, I'll review the implementation options for data security and illustrate why Xilinx® FPGAs are a superior platform with which to address this application.

Implementing Data Security

The implementation of data security can range from pure software to pure hardware. Typically, pure software is an attractive approach, but because of the computational intensity of authentication and encryption algorithms, this approach is inherently limited to single user/client-side applications or very low bandwidth server applications.

The next step up in data security implementations is to accelerate software solutions with custom hardware. This is an extremely common approach in x86- and network processor unit (NPU)-based system designs. Figure 1 illustrates a classic coprocessor data security solution.

Coprocessing is an attractive option for its simplicity and ability to elegantly scale the performance of pure software solutions. Its limitation is the performance impact that servicing the coprocessor can impose on the base system's memory – and especially I/O bandwidth. This is less an issue in a dedicated implementation (such as an x86 system packaged as a data security appliance), where you can carefully allocate these resources. And, with the transition from the traditional shared bus architecture of PCI and PCI-X to the switched fabric architecture of PCI Express, coprocessing bandwidth scales considerably and has a bright future.

However, coprocessing is not a panacea, and for many applications an in-line architecture will be more appropriate. This is illustrated in Figure 2.

In-line processors integrate data security directly into the system data path. This is a common approach in communications-centric applications (VPN gateways, storage security appliances) and achieves the equivalent computational off-load as a coprocessor with little to no impact on main system resources.

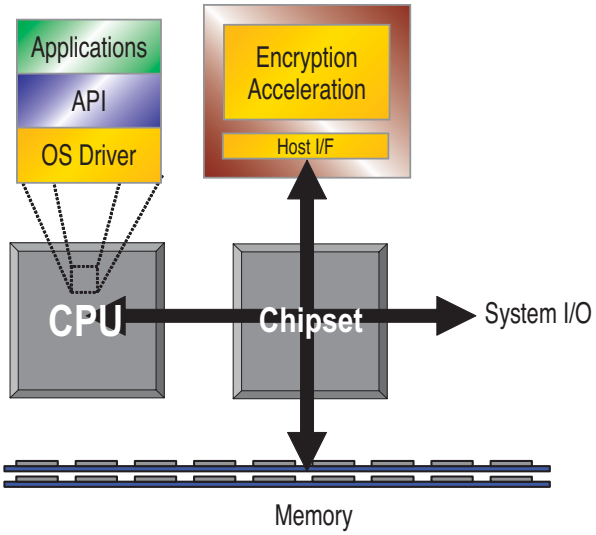


Figure 1 – Look-aside data security coprocessor

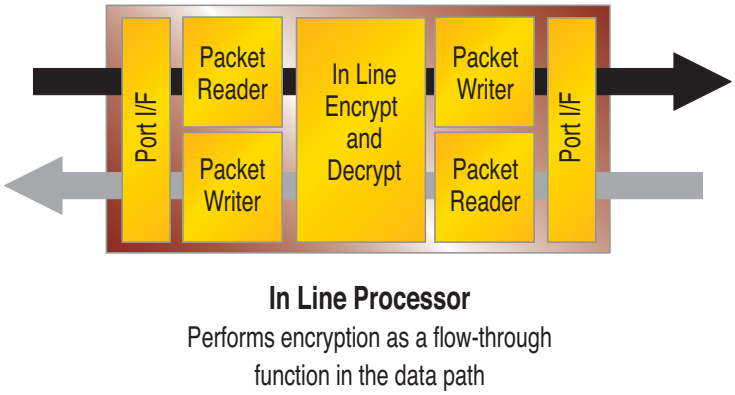


Figure 2 – In-line data security processing

Data Security Options

To scale performance beyond the abilities of pure software, data security requires additional hardware, either in the form of an add-on coprocessor or in-line data-path processor. So that brings us to the question of what kind of hardware: an ASIC, perhaps an ASSP, or an FPGA.

All of these options have advantages and disadvantages, and all will be appropriate in certain situations. However, when you consider the number of important decision criteria, the inherent advantages of FPGA solutions can be quite dramatic. To illustrate this, the relative merits of each implementation option across a range of decision criteria are listed in Table 1.

The values in Table 1 present a software-only solution as a baseline against which the three hardware options are compared on a relative basis. Let's explain each of these comparisons.

NRE

Incremental NRE (non-recurring engineering) from a pure software solution will be required for any hardware solution. In this category, ASSPs come out on top, as they are reasonably well packaged for specific target applications. However, that advantage is negated if you want to use them in a non-standard way.

FPGAs come in second, as by definition they will involve a circuit engineering exercise. However, the NRE for FPGAs is minimal, as they are the most efficient platform

	S/W Only	ASIC	ASSP	FPGA
Non-Recurring Engineering	N/A	High to Very High	Low	Medium
Performance	Low	High	High	High[1]
Unit Cost	N/A	Low	Medium	Medium
Customization	High	High	Low	High
Scalability	Low	Low	Medium	High
Device Availability	N/A	Medium Term	Medium Term	Long Term
Tamper Resistance	Low	Medium	Low to Medium	High
Re-Programmability	**High**	**Low to None**	**Low to None**	**High**

[1]With the exception of modular exponentiation, which FPGAs do well, but not efficiently for transaction server-class requirements.

Table 1 – Comparison of data security implementation options

for hardware design efforts. ASICs lag far behind, with typically much higher IP licensing costs, mask charges, tool chain costs, and long cycle times.

Performance

All of the hardware options will be dramatically superior to software only when it comes to performance. Although differences across these options do exist, it is likely not significant. All can deliver essentially wire-speed solutions for almost any application.

The one exception to this rule is modular exponentiation, a function heavily used in public key algorithms. FPGAs are quite capable of supporting 50-1,000 transactions per second (TPS) but do not scale efficiently to far higher performance. For those select applications really requiring 5K+ TPS, you will be better served by an ASIC or ASSP.

Unit Cost

In terms of physical unit cost, ASICs will obviously win, presuming that you have the necessary unit volumes against which to amortize the NRE. Because of this, ASICs are generally most compelling for high-volume (100K+) class applications. ASSPs and FPGAs, on the other hand, tend to be most suitable for low- to moderate-volume applications.

Some will argue that FPGAs cannot possibly compete against ASSPs in this category, and if you are comparing 100% equivalent configurations, this would be true. However, ASSPs are by design overloaded with features – so each configuration will have a large enough market to justify its development. This means that in most applications the system uses only a fraction of the features in any given part. In contrast, efficient FPGA design targets the specific, necessary, and sufficient features that you actually need and is only burdened with their implementation. When comparing a "kitchen sink" ASSP versus a "necessary and sufficient" FPGA design, you will find FPGA platforms to be extremely cost-effective.

Customization

Customization is a metric where ASICs and FPGAs can totally separate from ASSPs. As customer-defined purpose-built solutions, they have the inherent ability to integrate the specific features you require tailored to the specific design of your architecture, and in so doing impart an exponential benefit. This can take the form of supporting a legacy algorithm not available in an ASSP (there are many), integrating custom features for IT manageability, or tuning the solution to a specific application. In simple terms, it is about differentiation; developing features, performance, and cost efficiencies that distinguish you from your competition.

To illustrate this point, let's take a basic AES coprocessor as an example. If all you need is a basic coprocessor, there is no particular advantage, as shown in Figure 3A. However, if you are building this solution to serve a tape archive application, then the opportunity exists to integrate additional features and tune the architecture to provide a superior platform.

In our example, this could be the newly minted AES-GCM encryption cipher combined with in-line compression and block mapping to provide a fully integrated solution with significant differentiation for your platform (illustrated in Figure 3B) versus a product designed with an available ASSP.

Scalability

Scalability relates to servicing a broad price/performance range with a common

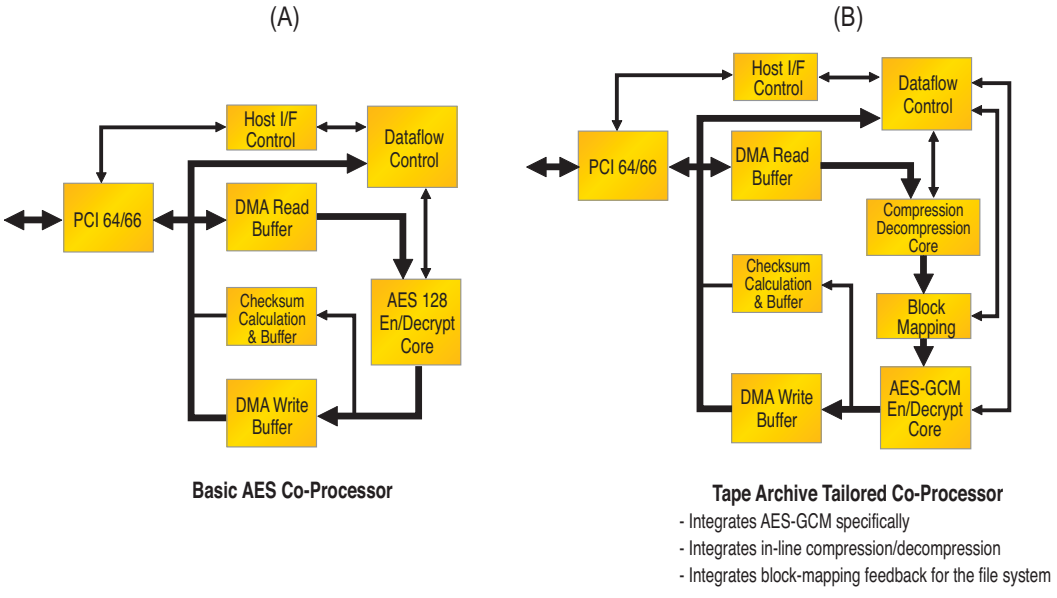


Figure 3 – A basic versus application-customized data security solution

architecture. Software is highly scalable, but as its performance tends to diminish with complexity, it is hard to exploit. One solution is to scale the performance of the host CPU platform, but complexities such as memory type and speed, chipset, and simple cost can make this less than ideal. ASICs do not scale very well either, as they are typically quite targeted by design. ASSPs offer a range of package-compatible price/performance options and thus have reasonable board-level scalability.

FPGAs provide a tremendous dynamic range in price/performance in three distinct ways:

- Custom logic configurations
- A wide range of device options
- Extremely efficient design reuse

In the simplest case, you can support a range of functional options in a single hardware design by simply loading a variety of different logic configurations into the device. In the medium case, you can achieve the same board-level scalability as ASSPs using a range of package-compatible FPGAs, enabling more performance and features. And in the maximum case, you have the ability to re-package a common core of system functionality across a variety of large-scale configurations and efficiently expand your market reach. An example here would be an in-line data-at-rest architecture reworked to support a wide variety of port requirements such as GE, 10GE, Fibre Channel, and SAS to address a range of storage equipment segments.

Device Availability

An often important consideration in system design is the long-term procurement logistics for the components used. This can be particularly important in communications infrastructures and aerospace/defense applications where data security is a common requirement. In this regard, FPGAs are second only to software, as they are historically available over very extended commercial lives. ASIC and ASSP availability will vary by vendor, but is typically far more restrictive.

Tamper Resistance

In particularly sensitive applications where the hardware is in the field, protecting the architecture against assault can be an important factor too. Data security by definition is deployed to protect valuable content, and therein lies the motivation for extreme attempts to crack a system's architecture.

ASSPs offer some protection in this regard, but as standard parts they are vulnerable to methodical analysis. Anything that is engineered can be reverse engineered; it is simply a matter of effort. ASICs present an additional barrier because of their more limited availability and undocumented nature, but still share similar vulnerabilities. FPGAs stand unique in that when appropriately packaged and safeguarded, they can fully erase themselves if they detect an apparent incursion, and an un-programmed FPGA reveals zero knowledge of your data security algorithms. Admittedly this is an extreme consideration, but one that could be decisive for your most paranoiac applications.

Reprogrammability

I have saved the best for last, as FPGA reprogrammability can bring truly unique capabilities to your solutions.

The world of data security is perpetually in flux. As standards change, methods evolve, and system and algorithmic vulnerabilities come to light. Software can change, but this is not typically so for ASICs and ASSPs, and almost certainly not at full speed. But FPGAs (field programmable gate arrays) are by definition reprogrammable too, giving them the unique ability to adapt to the changing world over time as efficiently as software. And not just in new product shipments, but in your installed base as well. This is a feature that can take the definition of a managed platform to an entirely new level.

Take for example the disaster of 802.11's WEP security technology. Launched to much acclaim, WEP proved to be amazingly vulnerable to simple attacks. Worse, once this became apparent, it took well over a year for the WPA solution to be defined, new chips designed and manufactured, and for secure 802.11i products to get to market. To add insult to injury, in the last year WPA has been superseded by WPA2. The result: products sold that did not deliver the security promised and an installed base of products that were left behind – twice.

With an FPGA enabled platform reasonable patches could have been quickly developed, the new specifications deployed immediately upon ratification, and the installed base could have been brought along in the process.

Other examples abound where reprogrammability will prove extremely valuable. Take the continual evolution of cipher modes and authentication algorithms, as seen in IPsec over the last 10 years and soon to continue as the 802.1AE (MACSec) standard emerges. Other examples include IEEE 1619 storage security standards (LRW-AES and AES-GCM) and their inevitable derivatives, as well as some 20 other data security modes currently at the proposal stage with NIST (National Institute for Standards and Technology). See *http://csrc.nist.gov/cryptotoolkit/modes/proposedmodes/* for details.

FPGA-based solutions give you the competitive advantage to support such developments before your rivals and to retrofit them into your installed base. That is data security differentiation, and that is the power that FPGAs can bring to your designs.

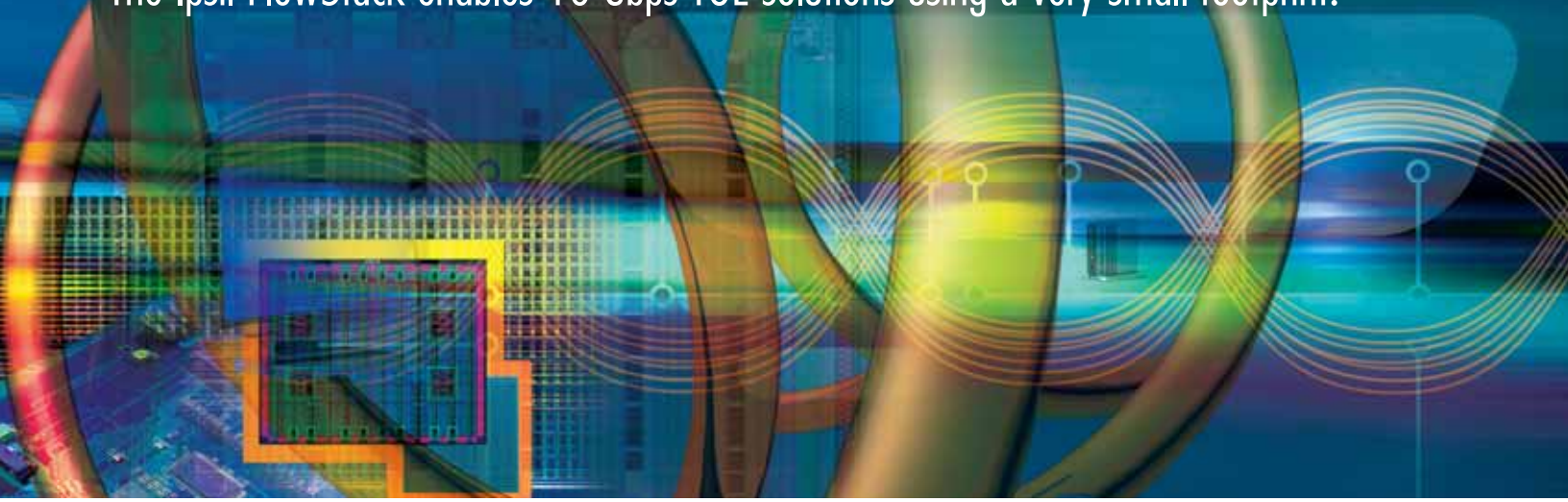
Conclusion

FPGAs give you flexibility, scalability, cost-effectiveness, and adaptability. All of these elements address the fundamental challenges facing today's data security designer: achieving performance across a range of products most cost-effectively while keeping your platform current with continuously evolving technology and integrating your product into the managed IT infrastructure.

To learn more about Xilinx FPGAs and IP solutions for data security, visit *www.xilinx.com/esp/security/data_security/index.htm.*

Virtex-II Pro FPGAs Enable 10 Gb iSCSI/TCP Offload

The Ipsil FlowStack enables 10 Gbps TOE solutions using a very small footprint.



by Sriram R. Chelluri
Senior Manager, Storage and Servers
Xilinx, Inc.
sriram.chelluri@xilinx.com

Enterprise data-center connectivity solutions have evolved over the years from direct attach storage (DAS) solutions (such as SCSI) to Ethernet-based file servers like network file system (NFS) and switch-based architectures (like Fibre Channel [FC]) that can encapsulate the SCSI/IP or FICON protocols.

Fibre Channel is a well-known high-performance protocol used in critical latency-sensitive applications like databases, video streaming, and tape backup. The most common data rate for Fibre Channel is 2 Gbps, with 4 Gbps slowly making headway into data centers, as the per-port cost factors for both are nearly the same. However, FC has not become a prevalent part of IT infrastructures because of its lack of interoperability between vendors and its high connectivity costs, requiring special host bus adaptors (HBAs).

To address the interoperability issues and costs associated with FC, IETF adopted a protocol to transport SCSI data over TCP, more commonly known as iSCSI. In this article, I'll review current iSCSI implementations and Ipsil Corporation's 10 Gbps TCP Offload Engine (TOE) solution to address this rapidly growing market.

Current iSCSI Solutions

Current iSCSI solutions rely on a complete software stack or on special network interface cards (NICs) based on ASICs for handling TCP/IP processing. Like FC, ASIC-based TOEs are expensive and have many interoperability issues. Vendors were building TOE cards long before the standards were approved. This resulted in a very slow adoption rate of iSCSI solutions. Also, ASIC-based solutions are expensive, with long lead times for development, testing, and manufacturing, and with general network performance outpacing storage device performance.

An all-software solution is acceptable for low-bandwidth applications, but high-performance applications would consume all of the CPU resources, creating a system bottleneck for critical applications. For example, every 1 Gbps in Ethernet performance requires 1 GHz of CPU resources at 100%. Also, when Ethernet was at 1 Gbps, FC had an established market place at 2 Gbps. iSCSI market penetration could not justify the price/performance.

After a few years of lackluster customer interest for iSCSI, it is slowly emerging as a viable complement to Fibre Channel. The network infrastructure is migrating to 10 Gbps, while FC is still at 4 Gbps.

To address the performance challenges of 10 Gbps iSCSI, Xilinx teamed with Ipsil Corporation to build a programmable solution for the iSCSI market based on Xilinx® Virtex™-II Pro FPGAs with RocketIO™ multi-gigabit transceivers.

FPGA-Based iSCSI/TOE Engine

With standards-compliant FC, PCIe core, and a TOE core from Ipsil, we created a technology demonstration platform for the 10 Gbps iSCSI TOE market, working with industry leaders in high-performance computing solutions. Programmable solutions enable system architects to add functionality as needed. Depending on the Xilinx product family, you can integrate multiple IP cores into a single FPGA, reducing board costs and time-to-market requirements. For example, you can:

- Combine the FC IP core with the iSCSI/TOE engine to make FC storage available to Ethernet clients
- Combine the PCIe IP core with the iSCSI/TOE engine for host-based iSCSI storage access
- Combine the TOE engine with established IPsec or the emerging 802.1AE specification for security

10 Gbps iSCSI TCP Offload Engine Core

Ipsil FlowStack 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 and high-performance synchronous state machine.

The FlowStack core also includes support for the FastPath of upper-layer application protocols such as RDDP, MPA, and RDMA. In addition, for embedded monitoring and remote management, FlowStack incorporates an embedded HTTP, FTP, and SNMP server. Ipsil FlowStack is built from a collection of synthesizable Verilog modules, which are composited to customer specifications and requirements before delivery and can therefore be customized to specific requirements.

In FPGA delivery, you can instantiate FlowStack along with customer intellectual property and configure it to operate without the need for host CPU support. The FlowStack packet processing engine operates in strictly deterministic time. FlowStack typically takes five core clock cycles to process each packet, which means that the core can be clocked at modest clock rates to achieve 1 or 10 Gbps throughput. This zero-jitter and low-overhead processing can be used to your advantage in low-latency processing and on high-speed networking applications.

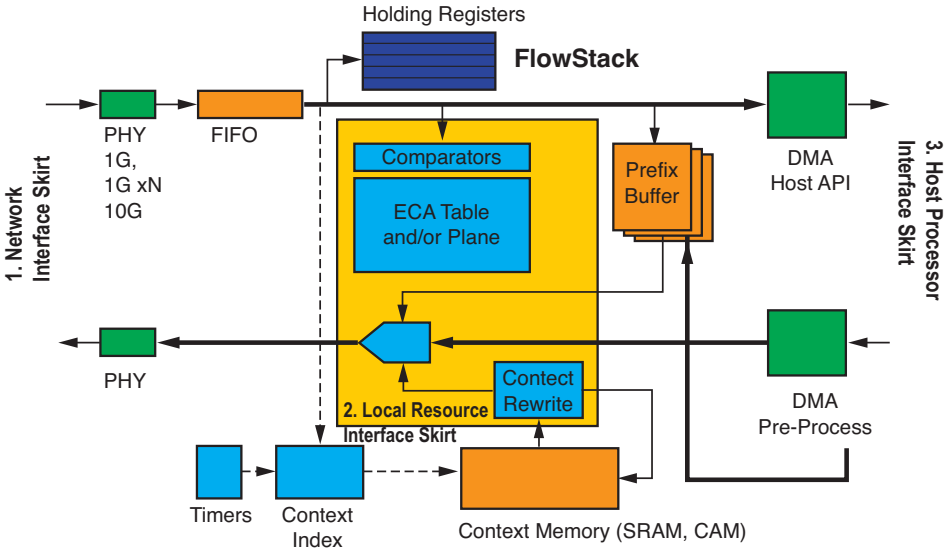


Figure 1 – Ipsil FlowStack interface

Interfacing to the Ipsil FlowStack Core

The customer interface to the Ipsil FlowStack core is through well-defined interconnect interfaces (Figure 1):

1. Network interface: Ethernet and other PHY interface
2. Local resource interface: off-core memory and local I/O interface
3. Host processor interface: SiliconSockets interface

The three interfaces are implemented independently in the form of a resource-frugal synchronous registered interface that you can integrate into customer FPGA designs in a straightforward manner, with the least overhead.

Network Interfaces

A variety of network interconnect options are available in a Xilinx FPGA delivery platform. These include:

- Industry-standard MII interface for Ethernet PHY interconnects, (10/100-BaseT Ethernet)
- Xilinx GMAC and 10GE MAC supporting 1 and 10 Gbps
- SPI-4.2 for OC-192 and OC-768 framers
- XAUI and 1-10Gbase CX-4 interfaces

Local Resource Interface

You can instantiate local memory resources in different ways. In an FPGA delivery platform, some memory can be realized with on-chip block RAM, while the FlowStack core also supports external SDR and DDR SDRAMs for additional buffer memory. In gigabit applications, FlowStack's local buffer memory requirements are very low, as it is capable of performing most operations directly to and from host memory, achieving true zero-copy operation at a high speed.

TCP state memory, required for some TCP and RDMA applications, is instantiated as combinations of SRAM, DRAM, or CAM blocks. These resources are composited by Ipsil to meet requirements according to specific customer and application needs. These resources can be on-chip or off-chip, depending on the demands, complexity, and requirements.

In a Virtex II-Pro VP40 FPGA platform, as many as 2,000 connections can be supported with internal block RAM in addition to the PCIe and 10 Gb MAC core. The FlowStack's cut-through architecture requires no data buffers and the connection count can scale, depending on the available internal and external memory.

Interface to the Host Processor: SiliconSockets

The user host processor interfaces to the Ipsil FlowStack core through the

SiliconSockets interface. The interface resources of the Ipsil FlowStack core are mapped into the processor's accessible address space, either as a set of host-accessible registers, through a coprocessor interface, or as an on-chip system peripheral (through an on-chip SoC bus) according to customer requirements. The SiliconSockets interface comprises two mechanisms: a register-based interface and a scatter/gather-based DMA.

Implementation Example – 10 Gbps TOE

The Ipsil FlowStack core was designed to be targeted to a variety of FPGA families depending on the performance and functionality requirements. It has a synchronous control path that comprises fully synthesizable Verilog modules. The fully registered data path has the flexibility of being instantiated using block RAM across the entire range of Xilinx product families.

Ipsil FlowStack SiliconSockets can be interfaced to PCI, PCI-X, or PCI-Express bus interfaces.

A sample design is illustrated with the FlowStack core targeted to a Virtex-II Pro device with a selection of Xilinx LogiCORE™ IP and AllianceCORE™ components that help to complete the application specifications.

Parameters of the Design Example

You can build a configuration with the following parameters:

- FlowStack instantiated in a Virtex-II Pro device
- 100 simultaneous TCP connections
- Connect to external gigabit PHY/GBIC using Xilinx GMAC and RocketIO transceivers

You can assume that the application will operate at the full 10 Gbps network speed and will be used in a typical Internet application. Consistent with current Internet engineering practice, a round-trip time of 200 ms is the assumed median case.

Figure 2 is a block diagram of an instance of an Ipsil IPμ FlowStack core in a Virtex-II Pro device. The FlowStack core has been composited with a selection of ancillary cores to allow it to provide an Ipsil SiliconSockets interface at the "uncooked-BSD-socket" level to TCP consumer applications resident in custom cores, either on the same Virtex-II Pro device or off-chip.

The Ipsil FlowStack core can operate at full line speed in any process technology in which the corresponding physical input and output mechanisms are available. The core uses nominally five internal machine cycles in its packet rewrite engine to process any packet.

The Ipsil FlowStack core requires approximately 2,500 LUTs for the packet rewrite engine, including 1,600 distributed bits of flip-flops. This is a very compact core that performs all TCP/IP packet processing procedures, but requires the support of several ancillary modules to manage raw memory resources and the overhead of the interface with custom cores.

TCP data buffer memory has been implemented as off-chip SDRAM. This has been sized assuming a 10 Gbps PHY connection and a 200 ms maximum TCP RTT (round-trip time). In this example, this has been realized in the form of a 32 MB DDR/SDRAM memory bank. This memory is interfaced using an SDRAM controller, which is budgeted at about 364 CLB slices.

In Figure 2, the FlowStack core has been supplied with a 10GMAC interface to an external 10 Gigabit Ethernet CX4 interface. In this example, based on resource requirements for the Xilinx LogicCORE 10GMAC, this module would need approximately 4,000 slices for a 10 Gbps PHY interconnection.

Conclusion

The Ipsil FlowStack is a very compact, fully synthesizable design. FlowStack is a processor-free design (that is, not containing within itself an embedded CPU) based on a hardware state machine; thus the clock-speed requirements are very modest. These characteristics lend themselves well to be realized in an FPGA delivery platform.

Higher level protocols such as iSCSI and RDMA are currently in the early stages of market acceptance. To ensure compatibility and interoperability with all versions of operating systems and hardware in a constantly evolving market, the high-density Virtex FPGA families with built-in multi-gigabit transceivers are an ideal choice for reducing board density while scaling to your performance requirements.

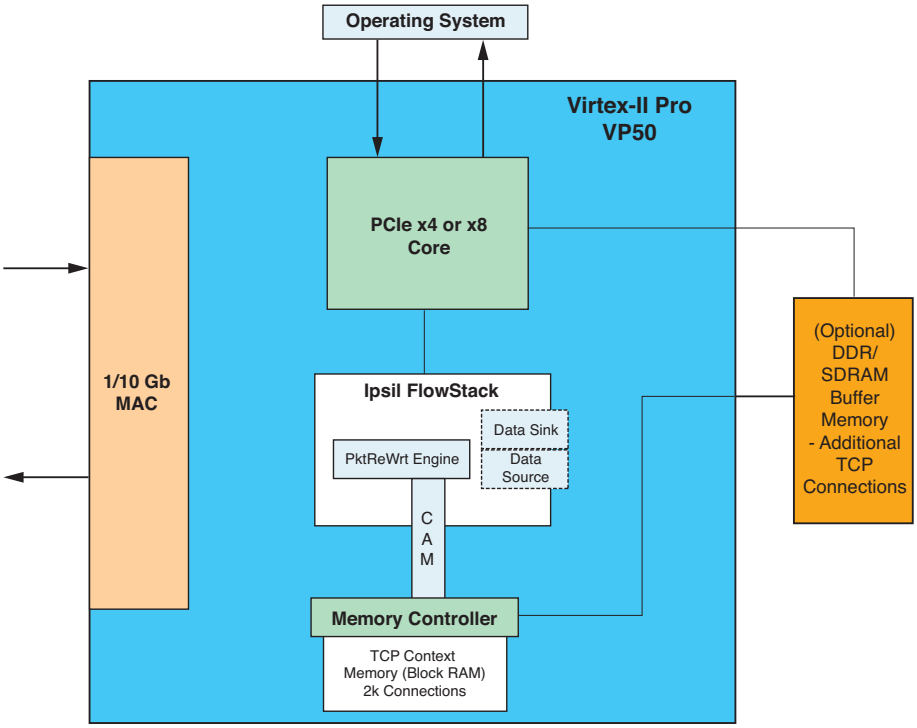
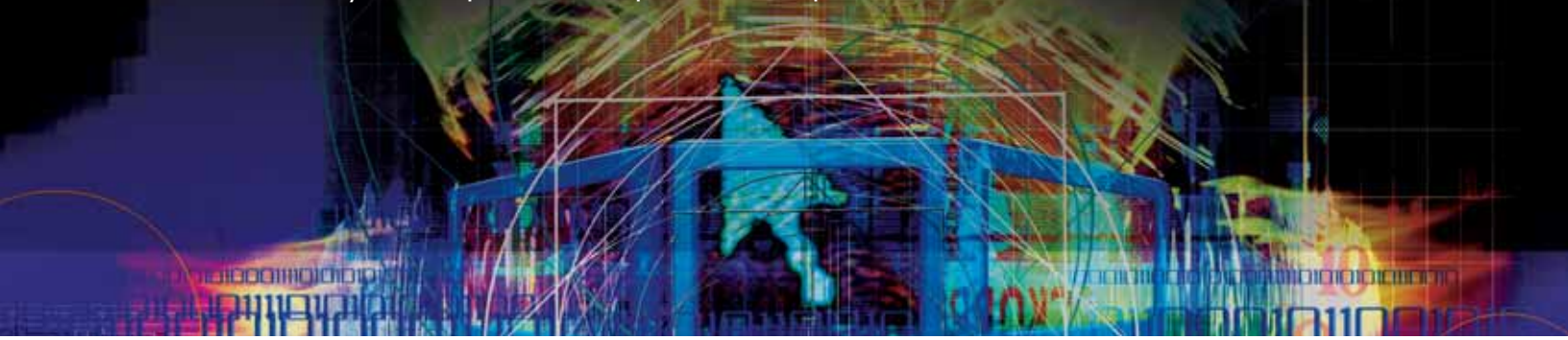


Figure 2 – FlowStack implementation on a Virtex-II Pro device

Xilinx in Enclosure Management

The FPGA-based Xilinx/Silicon Spectrum iKVM platform is compatible with the x86 software and driver environment.



by Mike Nelson
Senior Manager, Storage and Servers, Vertical Markets
Xilinx, Inc.
mike.nelson@xilinx.com

System management is one of the key distinguishing features of infrastructure-class equipment. Dedicated hardware with a private power plane provides a platform for remote health monitoring, system and software updating and configuration, statistics gathering, and many other functions. These capabilities enable today's infrastructure customers to allocate and manage their network-attached equipment far more efficiently to reduce operating and maintenance costs and improve their return on investment.

Remote operation of a platform's keyboard, video, and mouse (known as KVM) is the foundation for many system-management architectures. This approach first gained popularity through the physical connection of these signals for several to many computers to a single command console through a KVM switch. More recently, network-attached KVM (iKVM) has emerged, allowing the KVM subsystem to be directly integrated and the management architecture to easily support very large device populations.

In this article, I'll review a highly integrated iKVM platform based on the Xilinx® Spartan™-3 FPGA. This solution includes an embedded two-dimensional graphics controller, making it the one of the most integrated iKVM platforms on the market, reducing bill of materials (BOM) complexity and saving valuable PCB real estate. In addition, the unique approach this platform takes to keyboard and mouse re-direction enables state-of-the-art management capabilities in a way that is extremely compatible with today's x86 software and driver environment.

Anatomy of an iKVM Platform

The purpose of an iKVM platform is to perform the following functions:

- Support the establishment of an Ethernet-based management link from the host platform to a remote management console
- Access and compress the host platform's video stream and transmit it to the remote management console over Ethernet
- Accept incoming keyboard and mouse commands from the remote management console over Ethernet and redirect them to the host platform
- For "extra credit," it is desirable to support the creation and presentation of virtual media (vMedia) devices to the host platform – for example, one or more storage partitions physically residing on the management console appearing as local resources to the host

Conventional iKVM platforms achieve this through an architecture such as that illustrated in Figure 1.

In these designs, a high-performance microcontroller (uC) is the core of the platform. This microcontroller is responsible for the Ethernet link and for managing iKVM traffic to and from the remote management console. For the video link, these systems typically utilize a discrete PCI-based video compression engine connected to the host platform's video output. The video compressor reconstructs the host frame buffer and compresses it for transmission using a frame-differencing algorithm that only transmits the part of the screen that has changed from frame to frame. This function requires frame buffer memory borrowed from the uC (and consuming most of the bandwidth of the PCI interface). Finally, a serial or PS/2 port from the uC to the host typically provides a path for keyboard and mouse re-direction.

To meet the extra credit requirement to support remote virtual media services, such a platform would need additional capabilities. An attractive solution here would be to use USB with a hub controller as the KM redirect path to the host platform. In this case, the uC could present multiple virtual devices to the host – a keyboard, a mouse, and one or more USB storage devices. Although this is an attractive scheme, at this time few if any microcontrollers have such capability.

Xilinx/Silicon Spectrum iKVM Platform

In studying the conventional architecture, a few observations are significant:

- The subsystem requires a fairly significant device count
- This translates into a substantial PCB footprint, which can be problematic in today's space-constrained systems
- The graphics-to-RGB-to-compressor-to-Ethernet sequence, while effective, is an inefficient approach to the remote video requirement

Based on these observations, Xilinx and Silicon Spectrum designed and developed a highly integrated iKVM platform solution, illustrated in Figure 2.

This platform consolidates the graphics and iKVM subsystems into an integrated solution. This approach dramatically reduces the component count and PCB footprint (making it an exceptional cost savings opportunity on this basis alone) and enables highly efficient video compression as well. In addition, connection to the host PCI bus presents the opportunity for an extremely elegant mechanism to support keyboard and mouse redirection and vMedia requirements. Figure 3 illustrates the internal architecture of this solution.

The elements of this platform are:

- A multi-ported DDR2 memory controller
  - Shares memory across all subsystems
- An integrated PCI-based two-dimensional graphics controller
- An integrated video compression engine
- An integrated PCI-based psuedo USB host controller for keyboard, mouse, and vMedia re-direction
- A MircoBlaze™ CPU complex with Ethernet and SPI interface ports
- Integrated accelerators for modular exponentiation and IPsec to enable secure remote management sessions
- System power and reset control
- An optional DAC interface for video-out support

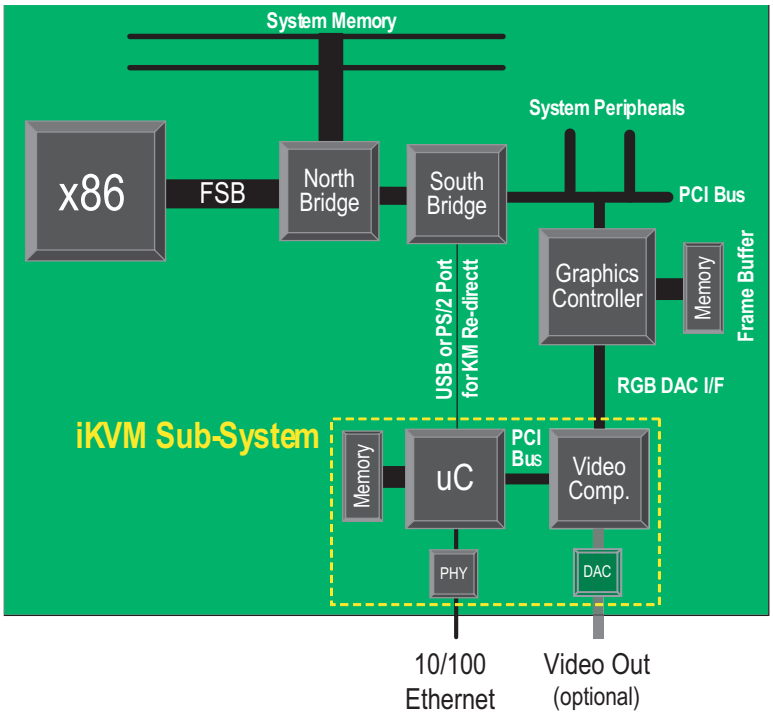


Figure 1 – Conventional iKVM subsystem platform

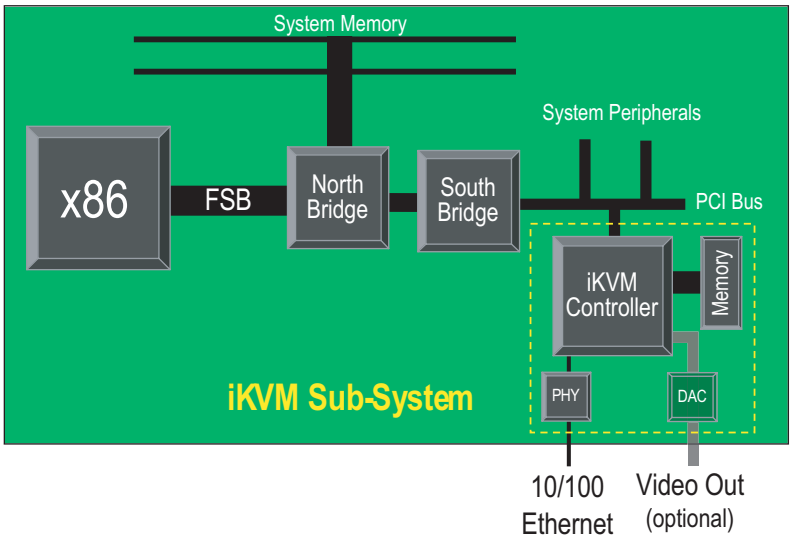


Figure 2 – Xilinx/Silicon Spectrum highly integrated iKVM platform

How It Works

Because the FPGA must be initialized, system operation starts with a device configuration from SPI Flash. However, this resource actually serves triple duty: it supports device configuration, provides the MicroBlaze CPU complex with a code store to boot into DDR2, and can act as a non-volatile storage resource for the iKVM subsystem as a whole – storing revision data, configuration parameters, and other useful information.

The fundamentals of iKVM operation are illustrated in Figure 4.

When the host system boots, as shown in Figure 4, it will see this platform as two separate peripherals on its PCI bus: a two-dimensional graphics controller and an apparent or pseudo USB host controller (apparent because there are no physical

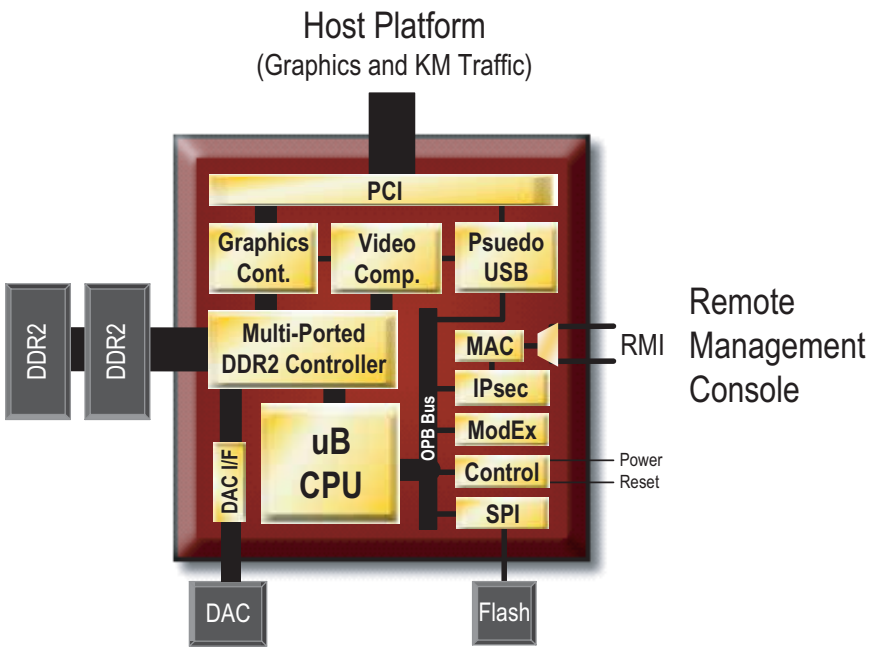


Figure 3 – Xilinx/Silicon Spectrum fully integrated iKVM architecture

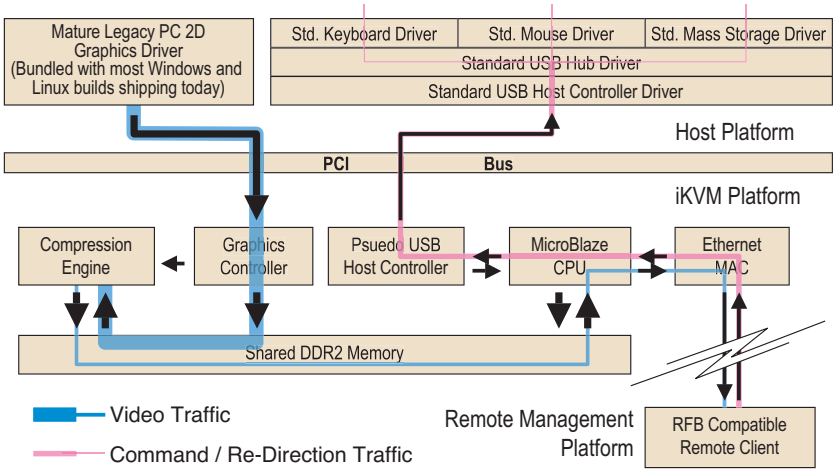


Figure 4 – Operational overview of iKVM system

USB devices anywhere within this system). It is important to note that the graphics controller is derived from a mature legacy PC architecture and is 100% compatible with drivers already bundled in almost every version of Windows and Linux shipping today. During operation, the graphics controller shares the DDR2 memory and uses a portion of it for the frame buffer.

Integrated graphics dramatically simplify video compression, as the controller explicitly knows what is being rendered frame-by-frame and can pass this information on to the compressor. In contrast, the conventional approach requires significant memory system bandwidth and computational power to perform real-time frame differencing after the fact. Compressed video is encoded to a format supported by the remote frame buffer (RFB) protocol, making it compatible with this popular iKVM management console standard.

Secure iKVM sessions are initiated by a remote management console through the MicroBlaze CPU complex. This is accomplished using public key infrastructure (PKI) techniques to authenticate each party and to exchange a shared advanced encryption standard (AES) key for encrypting session traffic. A modular exponentiation accelerator is included so that this computationally demanding process can be completed in just a few seconds. Session security is provided by a datapath accelerator that encrypts and decrypts all session traffic at full 10/100 Ethernet data rates.

Software running on the MicroBlaze processor manages all session I/O for the iKVM complex. Messages from the video compressor trigger transmission and encryption of outgoing compressed video blocks stored in DDR2. Incoming commands are parsed to drive power and reset control as well as keyboard and mouse redirection traffic through the pseudo USB host controller to standard drivers. Finally, the pseudo USB controller enables vMedia channels too, presenting them as standard USB storage devices to the host.

Conclusion

KVM technology has come a long way. Configurations that once comprised a physically shared monitor, keyboard, and mouse supporting four to eight computers became module-based network solutions using the Internet.

Now Xilinx and Silicon Spectrum have achieved maximum integration for iKVM with integrated graphics, security, intelligent video compression, and virtualized USB re-direction that is highly compatible with standard and legacy drivers. This solution allows OEMs and ODMs to reduce their BOM complexity, save precious PCB real estate, and provide state-of-the-art remote system management capabilities for today and tomorrow.

For more information about the Xilinx/Silicon Spectrum iKVM solution, visit *www.xilinx.com.*

Mining Data without Limits

FPGAs make high-throughput, low-latency data mining a reality.

by Bob Dutcher
VP of Marketing
Exegy Inc.
rdutcher@exegy.com

Data mining extracts previously unknown information from data for organizations, allowing them to make better, more timely decisions based on hard intelligence versus intuition.

Most large businesses and government agencies use some form of data mining, from unstructured textual-based mining to complex and in-depth statistical analysis. For example, Wall Street firms mine real-time market data feed to analyze large volumes of trading data, enabling them to feed automated real-time algorithmic trading systems and execute trading strategies, thus generating higher profits. In addition, government agencies mine data to analyze vast amounts of historic and real-time data for their counterterrorism efforts. This helps them identify suspicious activity and perform in-depth analysis on data to unravel terrorist plots.

In this article, I'll review how Exegy is applying cutting-edge reconfigurable technology to provide companies and government agencies with products that perform more sophisticated analysis of massive volumes of real-time and stored data at high speeds and low latencies.

The TextMiner appliance delivers a rich set of text mining functionality, providing up to 10,000-term exact matching, wild carding, approximate matching, proximity searching, and pattern matching.

Data Volumes Are Exploding

One of the greatest challenges businesses and government agencies face today is how to unlock value from their massive quantities of stored and real-time data, which is increasing at an exponential rate. Industry analysts estimate that organizations across all industries are increasing storage capacity by 80% each year. E-mail volumes are expected to increase 29% each year. In the financial industry, the Options Price Reporting Authority (OPRA) predicts that the message rate for its options data feed will approach 173,000 messages per second by July 2006, representing an increase of more than 230% in 18 months. Other organizations in every vertical market are experiencing similar explosions in market feed rates, e-mail message rates, and application-generated transactional data.

Current Solutions Are Not Keeping Pace

Historically, companies have attempted to solve their data-mining problems with software applications that run on general-purpose CPUs. Organizations continue to throw more servers at the problem, creating more instances of software applications and resulting in increased hardware, software, management, and maintenance costs – with minimal performance or scalability gains. Additional hardware and software have a marginal impact on functional scalability as well, as application execution times will continue to rise exponentially as application functionality becomes more complex. Furthermore, adding software and hardware does nothing to reduce application latency. In many cases, more hardware and software instances actually increase system latency.

As a result, data-mining solutions currently on the market focus on analyzing small to medium quantities of historic or real-time data.

To handle data-mining solutions for today's high-speed networks and multi-terabyte (TB) and petabyte data stores, a high-performance application using reconfigurable hardware that can search, analyze, filter, and transform structured and unstructured data on-demand is necessary to meet customer requirements.

A Different Approach

Exegy has combined software with reconfigurable hardware to deliver applications that perform at hardware processing speeds, while retaining the flexibility of software. This approach allows Exegy to deliver appliances with high performance, a rich feature set, and scalability to meet the requirements for mining large quantities of stored or real-time data.

Adaptable application logic requiring high data throughput and fast execution speeds is delivered with reconfigurable hardware. A software wrapper is applied around the reconfigurable hardware controlling the application modules on the Xilinx® Virtex™-II FPGA, providing a high degree of functional flexibility. Software is also used to supplement the reconfigurable hardware functionality for parts of the application that do not require high data throughput or high execution speeds. Using Virtex-II FPGAs, this approach delivers an extremely high throughput application, sustaining a rate as fast as 5 Gbps per appliance in Exegy's production version and more than 8 Gbps sustained in the next generation of appliances. This represents, in many cases, hundreds of times more throughput than traditional approaches. Additional benefits include virtually zero latency and the ability to add appliances for near linear throughput gains.

Exegy currently offers two high-performance appliances:

- TextMiner for high-performance unstructured text data mining
- DataMiner for high-performance analysis of real-time structured or stored data

Performance and Functionality Without Compromise

Exegy TextMiner mines real-time unstructured data feeds, such as news feeds, e-mail and messaging systems, and unstructured transactional or operational datastores, such as un-indexed document repositories or unstructured data staging repositories before indexing.

The TextMiner appliance delivers a rich set of text mining functionality, providing up to 10,000-term exact matching, wild carding, approximate matching, proximity searching, and pattern matching.

Approximate match identifies words that have a defined number of spelling anomalies. This is useful for applications that

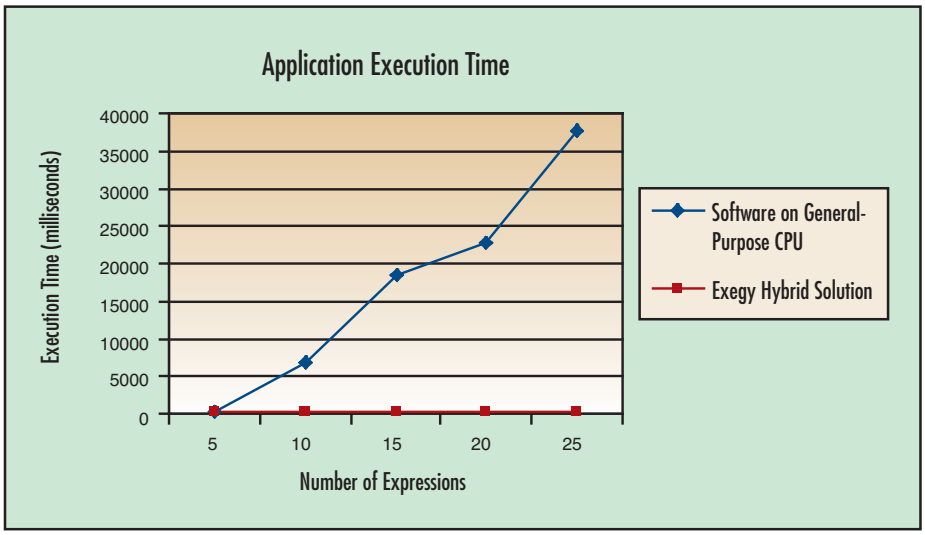


Figure 1 – Application execution time versus functional complexity

need to analyze scanned text data, which in some instances causes letters to be scanned incorrectly or to show up as a blank space.

Proximity search identifies words that are in a specified proximity to other specified words, which is useful for intelligence agencies that are looking for specific word combinations.

Pattern matching uses regular expressions to identify text that fits a pattern, such as a phone numbers or credit card numbers. Law enforcement agencies use this functionality to scan seized data to identify the scope of the criminal activity and to identify additional conspirators.

The functionality fully supports Unicode, allowing the appliances to be used in multi-lingual environments.

The Exegy hybrid application approach, which utilizes Xilinx FPGAs, delivers near-perfect functional scalability. Figure 1 demonstrates that performance for the Exegy appliance remains constant as the number of regular expressions the application runs increases. However, with a software solution running on a general-purpose CPU, application execution time significantly increases as the number of regular expressions increases. In this case, with as few as 25 expressions the Exegy hardware acceleration performs as much as 260 times faster than traditional software solutions running on a general-purpose CPU.

DataMiner

Exegy's DataMiner mines structured real-time data, such as real-time market feeds, data streams generated from a transactional system, and data derived from a database extract.

The DataMiner appliance delivers a rich set of calculation and statistical libraries, which is combined with 3-6+ TB of on-board high-speed persistent cache in each appliance. Statistical computations can be applied to the data flowing through the appliance and can be passed to end users or other applications in real time. This type of functionality is used by the financial industry to perform calculations such as volume weighted average price (VWAP), volatility, and variance for stocks and other financial instruments, feeding the results to real-time algorithmic trading systems.

In trading systems, market feeds are experiencing a massive increase in data volume. Reducing latency and the ability to deliver complex statistical analysis in this demanding environment provides a competitive advantage for Wall Street companies. Exegy data-mining appliances are able to meet the most demanding data-volume requirements while delivering functional richness and virtually zero latency. In addition, the number of calculations applied to a data stream does not negatively impact system latency or throughput. These results cannot be matched by software running on a general-purpose CPU.

With 3-6 TBs of on-board high-speed persistent cache, data streams can be cached and replayed to downstream applications that have experienced outages or for historical analysis and algorithmic modeling. Calculations and statistics can be applied to the historic data. The data stream can be transmitted with a real-time message stream, allowing an appliance to perform both real-time data processing and historic data processing simultaneously.

A Natural Choice

Reconfigurable hardware is a critical component that allows Exegy to deliver extremely high-performance solutions. Exegy chose Virtex-II FPGAs because of their high performance, high density, and available software development tools.

Time to market was reduced because Virtex-II devices have an available drop-in PCI-X interface core, which meant that the Exegy engineers did not have to spend time implementing industry-standard interfaces. Development was easier because of the uniform nature of the Xilinx logic cell and block RAM layout, which provides straightforward resource estimation and allocation for a library of application modules, working well for our modular architectural approach.

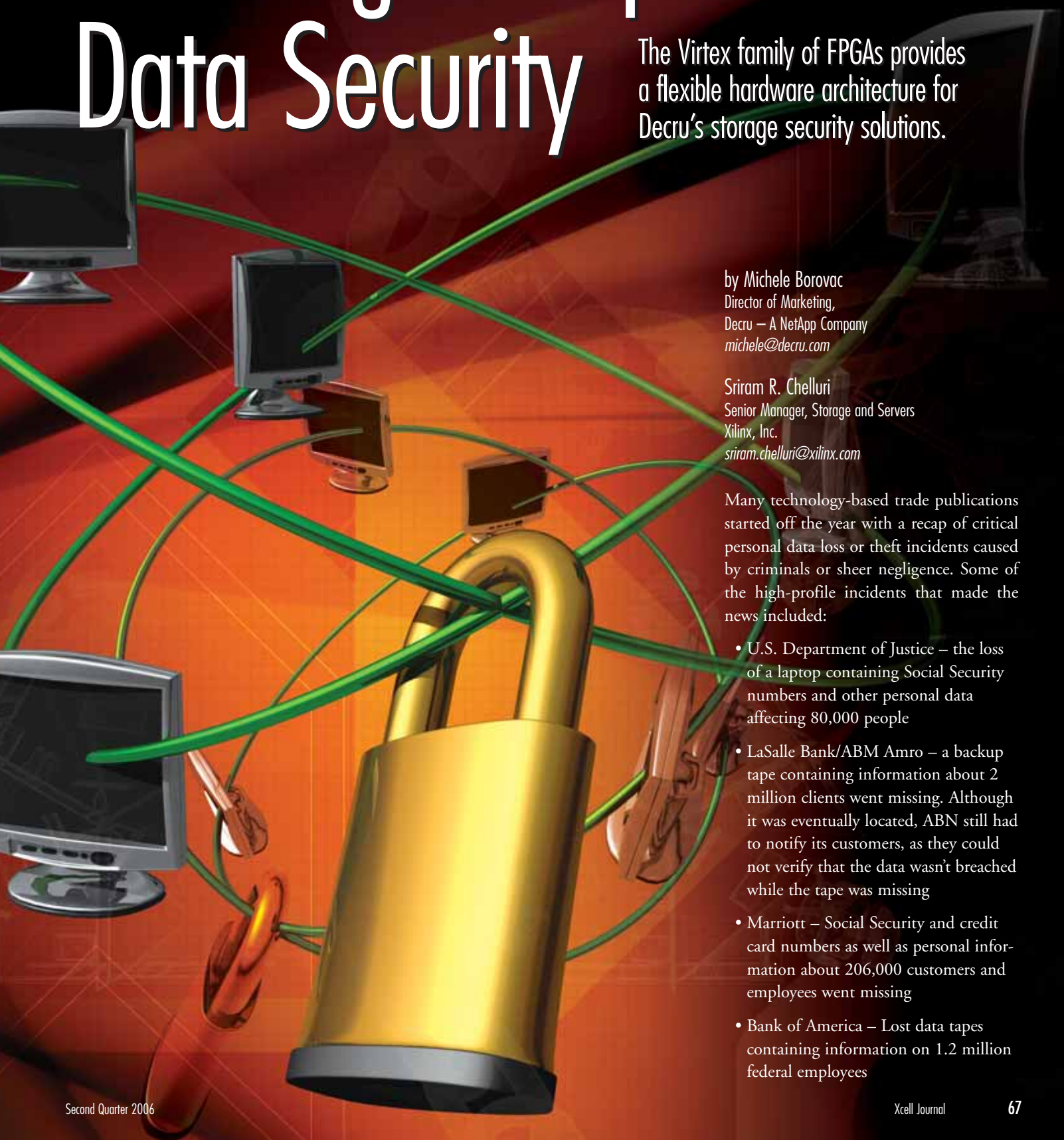
Conclusion

Exegy's hybrid approach, which incorporates Virtex-II FPGAs, delivers high-performance appliances that are fast, rich in functionality, and scalable. The Exegy solutions are hundreds and thousands of times faster than pure software applications running on general-purpose CPUs and have less latency. Exegy delivers a rich set of functionality, which can be fully utilized without any reduction in throughput or increase in latency. Customers can achieve a near linear increase in throughput as additional appliances are added to handle their most demanding data-mining needs.

For more information about Exegy reconfigurable hardware-based solutions, visit *www.exegy.com* or e-mail *info@exegy.com* to request a free white paper.

Enabling Enterprise-Wide Data Security

The Virtex family of FPGAs provides a flexible hardware architecture for Decru's storage security solutions.



by Michele Borovac
Director of Marketing,
Decru – A NetApp Company
michele@decru.com

Sriram R. Chelluri
Senior Manager, Storage and Servers
Xilinx, Inc.
sriram.chelluri@xilinx.com

Many technology-based trade publications started off the year with a recap of critical personal data loss or theft incidents caused by criminals or sheer negligence. Some of the high-profile incidents that made the news included:

- U.S. Department of Justice – the loss of a laptop containing Social Security numbers and other personal data affecting 80,000 people
- LaSalle Bank/ABM Amro – a backup tape containing information about 2 million clients went missing. Although it was eventually located, ABN still had to notify its customers, as they could not verify that the data wasn't breached while the tape was missing
- Marriott – Social Security and credit card numbers as well as personal information about 206,000 customers and employees went missing
- Bank of America – Lost data tapes containing information on 1.2 million federal employees

With Decru DataFort, enterprises and government organizations can fully leverage the benefits of networked storage, confident that their data assets are secure.

A recent survey by Network Computing magazine readers ranked "data security/privacy" as the number-one concern for IT administrators. When data goes missing, the problem is far greater than just losing data. The Federal Trade Commission estimates that at least 10 million people have had their identity stolen, resulting in an estimated $5 billion in damages for individuals and $48 billion for businesses. Both state and federal governments have stepped in with regulations designed to better protect consumer privacy and increase the penalties for companies that fail to protect this data.

2005 was the tipping point when corporate and government IT managers recognized the need to secure sensitive data at rest, before their company became front-page news. 2006 will prove to be the year that enterprises address this problem. Gartner, a major industry research group, predicts that "by year-end 2006, failure to encrypt credit card numbers stored in a database will be considered legal negligence in civil cases of unauthorized disclosures."

Decru DataFort storage security appliances represent the first and only unified wire-speed encryption platform for securing stored data across the enterprise, with support for NAS, DAS, SAN, iSCSI, and tape backup environments. DataFort appliances are operationally transparent and do not require any software to be installed on clients, servers, or storage devices (Figure 1).

Key data security applications for the enterprise include:

- Secure storage consolidation
- Insider threat mitigation
- Regulatory compliance
- Database security
- Secure tape backup and disaster recovery

With Decru DataFort, enterprises and government organizations can fully leverage the benefits of networked storage, confident that their data assets are secure.

Data Security Challenges

Traditional software-based encryption methods running on generic CPU architectures and specific OS environments are notoriously slow and cumbersome to implement. Strong encryption is computationally expensive and therefore consumes the most hardware resources, leaving little processing time for other tasks.

ASIC-based systems can be expensive, with long lead times, higher costs, and risk, and they are not upgradable. As encryption standards change, products can become obsolete, requiring customers to rip-and-replace to meet emerging security standards. This leads to a solution that is not future-proof. As an example, Microsoft banned the use of DES, MD4, MD5, and in some cases the SHA-1 encryption algorithm, making 50% of some ASIC-based encryption functionality obsolete.

ASIC-based encryption solutions cannot easily adapt to incorporate new standards, forcing customers to continue purchasing hardware each time new encryption features are needed.

Decru Storage Security Solutions

Decru DataFort storage security appliances leverage Xilinx® FPGA technology to enable high-performance encryption and compression, with the added flexibility to upgrade DataFort firmware to support new features and emerging standards. In addition to performance and easy implementation, another critical consideration for an encryption solution is the security of the system itself.

Hardened Architecture

Software-based or application-level encryption solutions typically do not have a secure method for storing encryption keys: keys are kept in clear text in an open operating system. This is a recipe for disaster if someone gains access to that machine.

DataFort appliances are designed from the ground up to protect stored data, using security-optimized hardware that is less vulnerable to attack than off-the-shelf hardware and software solutions. At the heart of the appliance is the storage encryption processor (SEP), a hardware engine

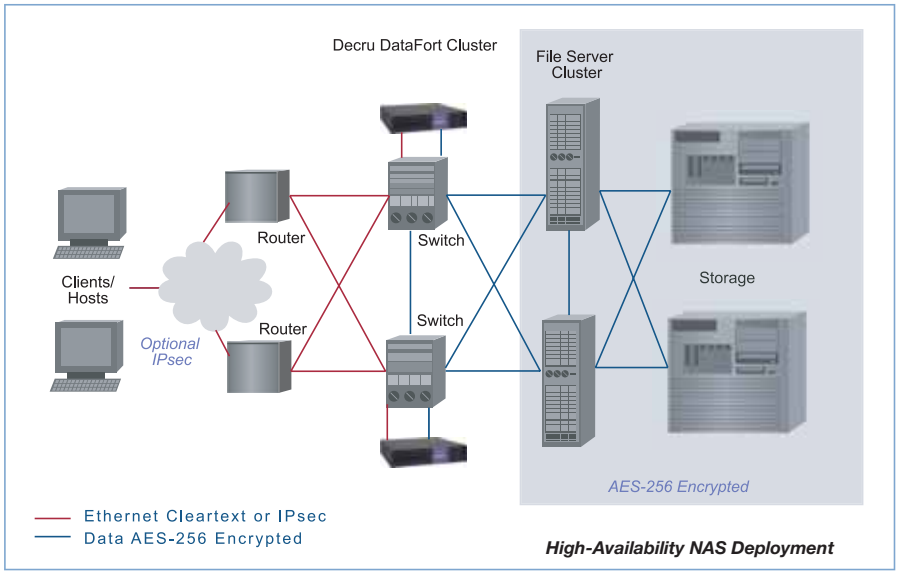


Figure 1 – High-availability deployment

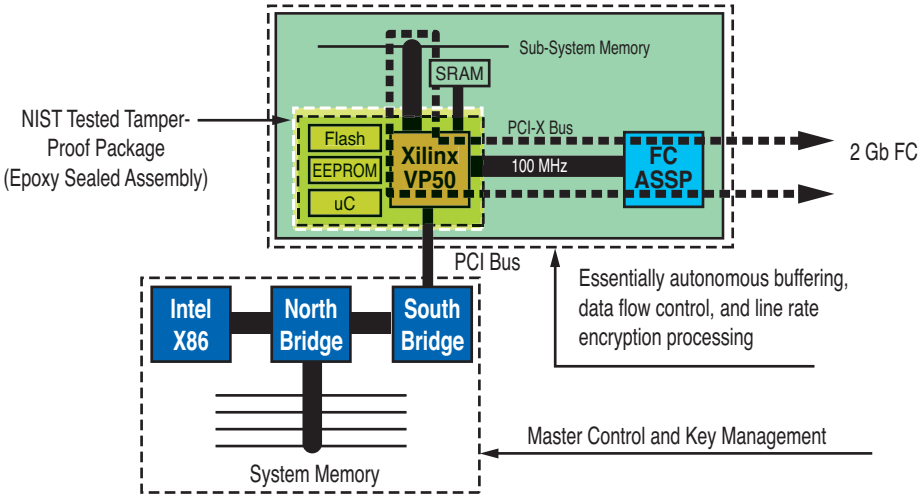


Figure 2 – Virtex-based encryption processing

that performs wire-speed, low-latency encryption and decryption while ensuring both data and key security.

DataFort appliances combine hardware-based AES-256 encryption, authentication, strong access controls, and crypto-signed logging in a hardened platform optimized for performance and reliability. DataFort appliances can be deployed in active/active clusters for availability and failover, and additional appliances can be added to address higher throughput requirements.

Application Coprocessing

To meet their design and time-to-market goals, Decru selected Xilinx Virtex™-II Pro FPGAs. An FPGA-based configurable coprocessing solution enables Decru DataFort appliances to:

- Adapt to changing encryption standards
- Add new features to systems already deployed at customer sites
- Meet performance and scalability requirements
- Support a broad range of features and performance at various price points
- Reconfigure hardware to adapt to customer requirements (Figure 2)

The Virtex product line comprises low-power and high-performance FPGAs capable of supporting 10 Gbps. Both Virtex-II Pro and Virtex-4 devices feature embedded PowerPC™ 405 hardware cores that run at speeds as fast as 450 MHz, soft IP cores such as the 32-bit MicroBlaze™ RISC processor, and up to 200,000 cells of configurable logic. The Virtex-4 series features a scalable Advanced Silicon Modular Block (ASMBL) architecture that has allowed Xilinx to produce a programmable silicon platform with unprecedented capabilities and configurations to build coprocessing solutions.

In the Decru solution, the Virtex FPGAs enable wire-speed encryption and compression for tape appliances.

Conclusion

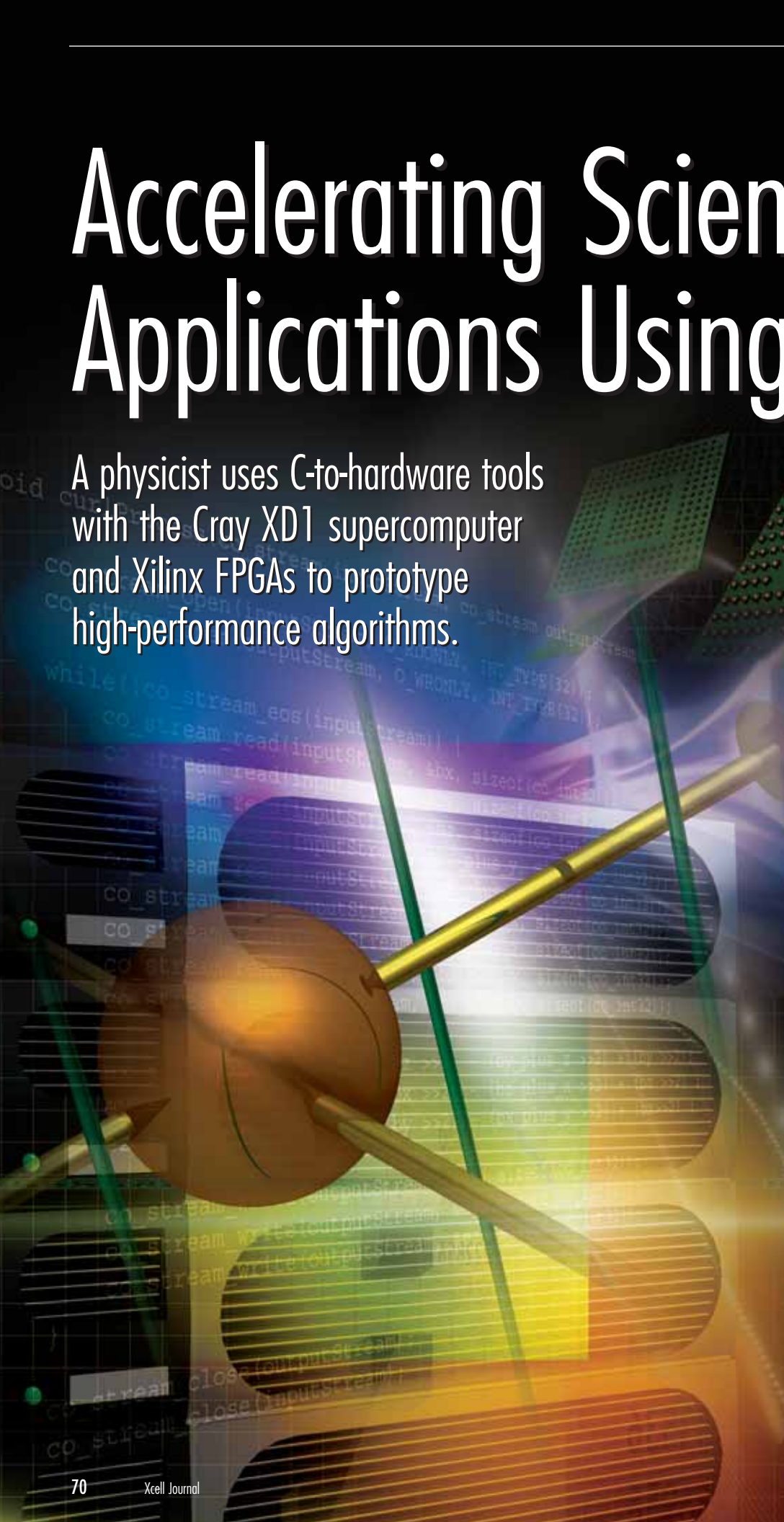
To defend against the increasing number of attacks on data at rest, IT managers are implementing data encryption technology solutions to protect sensitive or regulated data. An adaptable, special-purpose data security coprocessor is the foundation for Decru's FPGA-based encryption appliance.

Decru DataFort appliances, based on the Xilinx Virtex family, provide the only unified platform available to secure data at rest across the enterprise. DataFort appliances lock down sensitive data without degrading network performance or requiring costly changes to the storage environment. By using Xilinx FPGAs, Decru can provide a high-performance, highly scalable encryption solution.

For more information about Decru DataFort appliances, visit *www.decru.com* or e-mail *info@decru.com*. To learn more about implementing reconfigurable computing solutions, contact *sriram.chelluri@xilinx.com*.

Accelerating Scientific Applications Using FPGAs

A physicist uses C-to-hardware tools with the Cray XD1 supercomputer and Xilinx FPGAs to prototype high-performance algorithms.



by Peter Messmer, Ph.D.
Research Scientist
Tech-X Corporation
messmer@txcorp.com

Ralph Bodenner
Senior Software Engineer
Impulse Accelerated Technologies
ralph.bodenner@impulsec.com

Xilinx® FPGAs are increasingly used alongside traditional processors in a variety of hybrid parallel computing systems. In systems such as the Cray XD1 supercomputer, FPGAs play a supporting role by providing massive instruction-level parallelism and by fundamentally changing the way that key algorithms are processed.

Although many scientific applications could benefit from these hybrid systems, the complexity of the hardware programming process reduces the potentially large user community to a few hardware-savvy scientists. The promise of latest-generation software-to-hardware translation tools is to enable domain scientists (chemists, biologists, or physicists) with little or no hardware programming experience to take advantage of FPGA-enhanced systems. In this article, we'll describe the experience of one physicist using such tools to develop an FPGA-accelerated application targeting a Cray XD1 system.

Fortunately, software-to-hardware tools now exist that allow FPGA-based scientific algorithms to be described using familiar, software-oriented methods.

FPGAs for Application Acceleration

Programming software algorithms into FPGA hardware has traditionally required specific knowledge of hardware design methods, including hardware description languages such as VHDL or Verilog. Although these methods may be productive for hardware designers, they are typically not suitable for domain scientists and higher level software programmers. When considering FPGAs for high-performance computing, one challenge has been finding a way to move critical algorithms from a pure software implementation (written in C) into an FPGA, with minimal FPGA hardware design knowledge.

Fortunately, software-to-hardware tools now exist that allow FPGA-based scientific algorithms to be described using familiar, software-oriented methods. Using these tools, an application and its key algorithms can be described in standard C, with the addition of relatively simple library functions to specify inter-process communications. The critical algorithms can then be compiled automatically into HDL representations, which are subsequently synthesized into lower level hardware targeting one or more FPGA devices. Although it still may require a certain level of FPGA knowledge and in-depth hardware understanding to optimize the application for the highest possible performance, the formulation of the algorithm and initial testing can now be left to the domain scientist.

Case Study: The FDTD Algorithm

The finite-difference time-domain (FDTD) algorithm is a well-established tool in computational electrodynamics. Applications of FDTD include propagation problems (such as wave propagation in complex urban environments), photonics problems, high-frequency component modeling, scattering problems (like determination of radar signatures), electromagnetic compatibility problems (including transmitter-tissue interaction), and plasma modeling.

A particular form of the FDTD algorithm, the Yee-FDTD algorithm, discretizes the electromagnetic field in space and advances both Faraday's and Ampere's laws in time. Figure 1 illustrates this operation for one cell in the computational grid, as well as the spatial location of the field components on a staggered grid. An equivalent update operation must be performed for each field component for all cells on the computational grid.

The spatial grid has to be fine enough to accurately sample the shortest wavelength. In addition, the fastest waves in the system should not propagate more than one grid-cell per timestep, which sets an upper limit on the timestep. FDTD simulations therefore consist generally of large computational grids and a large number of timesteps.

The FDTD algorithm can be summarized as:

```
for i = 0, n steps
  apply boundary condition
  update E field: E_new = E_old + c2 * curl B * dt
  update B field: B_new = B_old - curl E * dt
end
```

The large number of cells and the high degree of parallelism in the computations make the FDTD an ideal candidate to benefit from FPGA hardware acceleration.

This project had two goals: first, we wanted to find a way to move this critical algorithm from a pure software implementation (expressed in C language) into the FPGA. Second, we wanted to investigate if modern software-to-hardware tools would allow a scientist with minimal FPGA hardware design knowledge to port other, similar algorithms to an FPGA-enhanced platform. To achieve these goals, the hardware-savvy engineer participating in this project should mainly function as a consultant.

From Software to FPGA Hardware

The first step in performing a software-to-hardware conversion was to select a target platform. We chose a Cray XD1 supercomputer system, featuring FPGA application accelerators on each compute node. Each application accelerator in the XD1 system comprises one or more Xilinx® Virtex™-II Pro or Virtex-4 FPGAs.

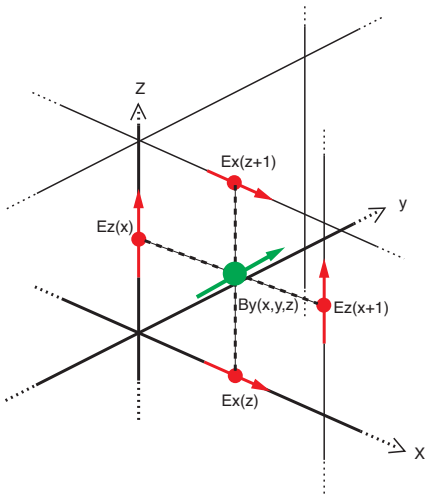


Figure 1 – The Yee-FDTD computational grid. The E field components are centered on the cell edges, the B field components at the cell faces. The dashed line indicates the computation of the y component of curl E, through finite differences, which is used to update By.

Handling the application accelerator and loading the FPGA bitstreams was simplified by Cray system tools, making the entire process easier for a software programmer to comprehend.

After choosing the platform, our next step was to select a software-to-hardware translator. We chose Impulse C from Impulse Accelerated Technologies for these experiments because of its emphasis on standard C, its ability to automatically generate and optimize parallel hardware, and its support for the Cray XD1. The Impulse

```
void curlProcess(co_stream inputStream, co_stream outputStream)
{
   co_stream_open(inputStream, O_RDONLY, INT_TYPE(32));
   co_stream_open(outputStream, O_WRONLY, INT_TYPE(32));

   while(!co_stream_eos(inputStream)) {
      co_stream_read(inputStream, &bx, sizeof(co_int32));
      co_stream_read(inputStream, &by, sizeof(co_int32));
      co_stream_read(inputStream, &bz, sizeof(co_int32));
      co_stream_read(inputStream, &bx_plus_y, sizeof(co_int32));
      co_stream_read(inputStream, &bx_plus_z, sizeof(co_int32));
      co_stream_read(inputStream, &by_plus_x, sizeof(co_int32));
      co_stream_read(inputStream, &by_plus_z, sizeof(co_int32));
      co_stream_read(inputStream, &bz_plus_x, sizeof(co_int32));
      co_stream_read(inputStream, &bz_plus_y, sizeof(co_int32));

      ex = (bz_plus_y >>2) - (bz >>2) - (by_plus_z >>2) + (by >>2);
      ey = (bx_plus_z >>2) - (bx >>2) - (bz_plus_x >>2) + (bz >>2) ;
      ez = (by_plus_x >>2) - (by >>2) - (bx_plus_y >>2) + (bx>>2) ;

      co_stream_write(outputStream, &ex, sizeof(co_int32));
      co_stream_write(outputStream, &ey, sizeof(co_int32));
      co_stream_write(outputStream, &ez, sizeof(co_int32));
   }

   co_stream_close(outputStream);
   co_stream_close(inputStream);
}
```

Figure 2 – Summarized C code for the Impulse C hardware process applying the curl operator to the B field.

C tools made it easy to prototype various hardware/software partitioning strategies and to test and debug an application using familiar C development tools.

Partitioning the Application

Identifying the critical sections of the application code is important for any hardware-accelerated algorithm. In our case, the time-consuming part of the FDTD algorithm was the computation of the curl operation, which required several additions and at least one multiplication and division per cell half-update. As it turns out, the updating of individual cells does not depend on the updating of other cells, making this code a good candidate for parallelization.

Impulse C supports the concept of multi-processing, in which multiple parallel processes can be described using the C language and interconnected to form a system of semi-autonomous processing nodes. Various methods of inter-process communication are provided in Impulse C, including streams, shared memory, and signals.

To keep things simple, we started our FPGA-based FDTD algorithm with a plain C implementation of the integer-based curl algorithm, which was isolated into a distinct C-language hardware process. Using Impulse C, this hardware process looks very much like a typical C subroutine, with the addition of Impulse C-stream read and write function calls to provide abstract data communication. The source code describing this process is summarized in Figure 2.

This splitting of an application into hardware and software processes is relatively straightforward from a programming perspective, but it is also quite communication-intensive; the input and output streams become performance bottlenecks. However, this initial partitioning allowed us to experiment with hardware generation and the hardware/software generation process, as well as to begin understanding how to optimize C code for higher performance.

After desktop software simulation using the Impulse tools, we invoked the software-to-hardware compiler. The automatically generated files included the hardware for the curl function, as well as all required interfaces implementing data streams through Cray's RapidArray interface (see Figure 3). The generated VHDL was then translated into a Virtex-II Pro FPGA bitstream using Xilinx ISE™ tools, following instructions provided by Impulse and Cray. Once again, the Impulse tools abstracted the process, including export to the Xilinx tools, such that we did not need to understand the process in detail. The resulting bitstream and the sources for the software process (which was to run on the Opteron) were transferred to the Cray XD1 platform using tools provided by Cray.

Moving More of the Application to the FPGA

After developing and deploying a single algorithm to the Cray XD1 platform, we were in a position to begin optimizing the larger application. Initial experiments led us to the conclusion that dedicating the application accelerator to computing only the curls would lead to poor performance, given the large amount of data transfer required between the CPU and the FPGA. Fortunately, the high-gate density of the Virtex-II Pro device allowed us to put far more demanding operations onto the FPGA itself.

As a next step, we implemented the entire FDTD algorithm as a hardware process on the FPGA. In this design, the host CPU is responsible only for commu-

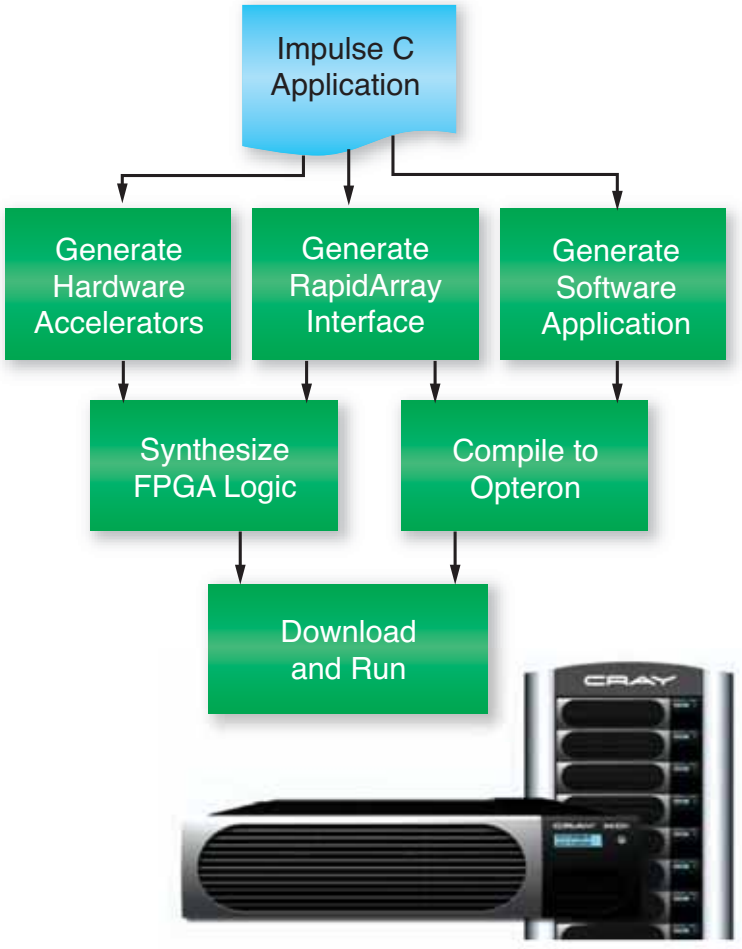


Figure 3 – Impulse C tools allow you to partition an application between hardware and software and generate the required Cray XD1 platform interfaces.

nicating the number of required timesteps to the FPGA and setting the boundary condition in a pre-defined part of the computational domain at each timestep. The FPGA then stores all of the fields and performs both the E- and B-field updates. With this more complete algorithm working in hardware – producing correct and reliable results as observed on the software side – we now have the confidence to try many different partitioning strategies and C-level optimizations to increase overall system performance.

Throughout the optimization process, we treated the generated hardware description as a black box and only performed optimizations on the high-level C source files. The iterative application development, optimization, and prototype hardware generation enabled by the Impulse C tools proved to be an enormous time-saver.

Future Optimizations

Now that we have the basic algorithm working on the Cray XD1 platform with its FPGA accelerator – a process that required little more than three weeks of total effort, including the time required to learn the Impulse tools and programming model and to become familiar with the Cray XD1 system – we are now focusing on algorithm optimization. One of the optimizations will be to utilize the high-speed QDR II memories, which will be directly supported in Impulse C using abstract memory block read/block write functions. This will reduce or eliminate the stream communication bottlenecks that we are currently observing in the system as a whole.

During this optimization phase, we are using more advanced Impulse optimization tools that allow us to analyze the generated logic and automatically generate loop pipelines for increased throughput. What we are learning as we pursue these optimizations is that the high-level language approach to hardware programming typified by Impulse C allows iterative refining of the algorithm and supports the concept of explorative optimization. Interactive design analysis tools, such as the graphical optimizer and cycle-accurate debugger/simulator provided with Impulse C, allow us to estimate the performance of an algorithm before translation and mapping. In this way, creating FPGA accelerators becomes a highly iterative, dynamic process of C coding, debugging, and high-level optimization.

Conclusion

Before the advent of software-to-hardware tools, the creation of an FPGA-accelerated application would have been impractical for all but the most hardware-savvy domain scientists. The Impulse C approach, when combined with the Cray XD1 platform, enabled us to convert prototype algorithms into hardware realizations and to quickly observe results using different methods and styles of C programming. This project has given us confidence that – even as software application programmers – we have the ability to experiment with many different algorithmic approaches and can accelerate scientific applications without relying on the knowledge of an experienced FPGA designer.

For more information on accelerating applications using FPGAs and Impulse C, visit *www.impulsec.com.*

The authors would like to thank Dave Strenski, Steve Margerm, and others at Cray, Inc. for their assistance and for providing access to the Cray XD1 system, and Roy White at Xilinx for providing access to the ISE tools used in this evaluation project.

Co-Processing Solutions for Delivering Storage Services Over Distance

Virtex FPGAs enable a services-based architecture for delivering secure file solutions over distance.

by Sriram R. Chelluri
Senior Manager, Storage and Servers
Xilinx, Inc.
sriram.chelluri@xilinx.com

Initially, storage area networks (SANs) have enabled enterprises to consolidate servers and storage, centralized backups, and implement a tiered storage model for data replication and disaster recovery within the data center. SANs are evolving into service area networks, providing geographically dispersed remote file and database access, mirroring and replication for disaster recovery, and truly centralized backups.

Consolidating resources has reduced hardware costs and simplified the purchasing and management of data centers. However, similar gains have not been made in branch and remote offices. Traditionally, these offices have a mini-IT infrastructure similar to the corporate office, comprising:

- File/print services
- Backup (local and remote)
- Wide area network (WAN) access

Having branch offices completely depend on remote access to the corporate data center does not provide the performance requirements necessary to conduct business on a daily basis. High-speed WAN links are costly, prone to reliability issues, and have a high latency that storage protocols such as SCSI cannot tolerate. Many startups and established companies are developing products to fix these issues and bring the same level of services like mirroring, replication, and backup that SANs have enabled in the data center to remote offices.

In this article, I'll cover solutions that can be enabled by reconfigurable programmable logic in service area networks, highlighting file services for remote offices often referred to as wide area data services/wide area file services (WADS/WAFS). As an example, I'll describe how you can implement data security as a co-processing solution in an x86-based appliance.

Wide Area Data Services/ Wide Area File Services

WADS/WAFS is a fairly new concept in providing IT services to branch and remote offices. The goal is to provide LAN-level performance with scalable and redundant storage services from a centralized corporate data center to remote offices, as shown in Figure 1. WADS/WAFS technology essentially enables IT administrators to provide consolidated services to remote sites. In this article, I will use the terms wide area data services and wide area file services interchangeably, because as the products mature, the distinction between the two will disappear.

Storage services appliance products enable branch/remote offices to gain Tier-1 IT services without building an entire infrastructure. WAFS are a subset of these data services and provide a range of functions, including:

- File and print services
- E-mail services
- Web services
- Network services
- RAID services

Currently, most of the services are provided in some kind of appliance running Windows or Linux file and print services supporting common Internet file systems (CIFS) or network file systems (NFS). The hardware is essentially a dual/quad Intel Xeon processor, a large RAM or hard disk to keep track of files, and high-performance gigabit network interface cards for connectivity. What separates each vendor are the software features running on these appliances: file performance, WAN optimization, quality of service, security, application support, and management features.

Remote storage services based on general-purpose architectures are not optimized for data-networking functions. Instead of throwing raw CPU power at performance and scalability problems in wide area data services, an alternative approach would be to run storage and networking services on a Xilinx® Virtex™ FPGA co-processor. Xilinx FPGAs provide high performance and scalable software-enabled services without having to over-design the hardware for future-proofing. Some of the services that can be implemented in the FPGAs are:

- File services: distributed lock manager, hashing tables, file caching, access control
- Network services: TCP offload (1G/10G), compression, web caching, link management
- Security services: encryption, anti-virus checking, intrusion detection
- Protocol support: protocol conversion (GE, FC, iSCSI, FCIP), packet multicast/broadcast, link management, routing
- System management: link statistics, file statistics, service-level agreement (SLA) monitoring

Figure 2 shows the functionality that can be implemented as a co-processing solution on one or more FPGAs, depending on the chipset. Software services can be phased into the WAFS products from a central location, fine-tuning services to branch offices based on policy-based business needs. Besides core file and network level optimization, a co-processing solution can also increase performance of protocols such as CIFS and NFS. Also, programmable logic enables services like data and network security, data classification, and mining at wire-speed for applications like

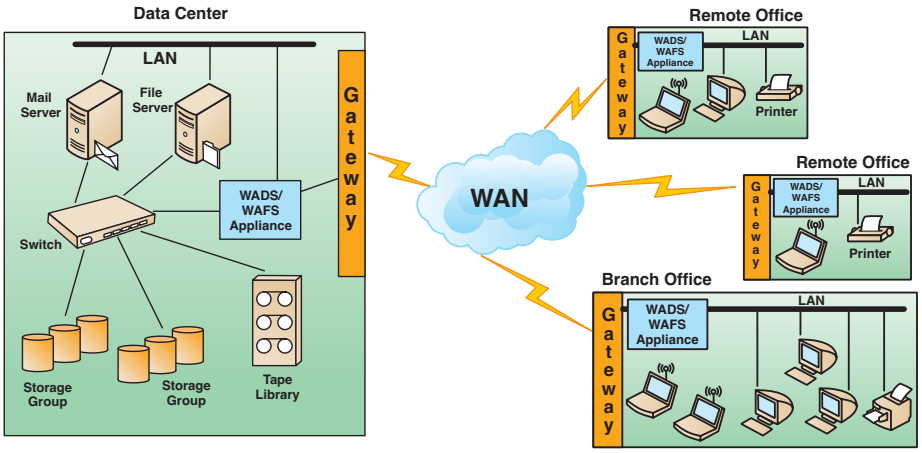


Figure 1 – Corporate IT infrastructure

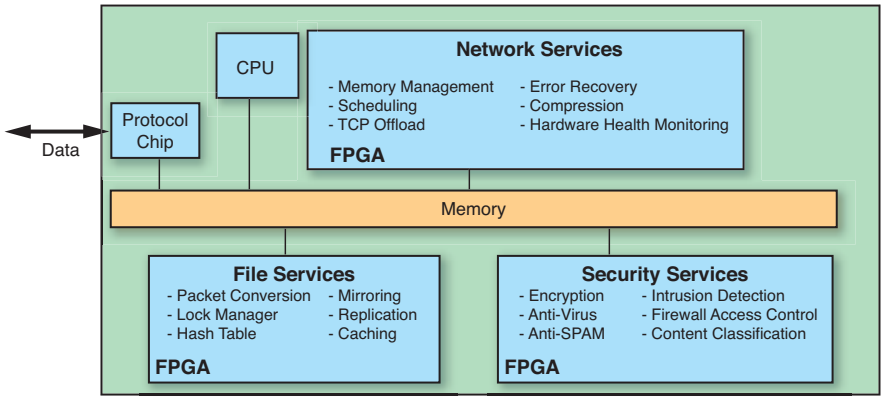


Figure 2 – Data and networking services on a Virtex-4 FPGA

Microsoft Exchange, Autodesk AutoCAD, Oracle DB, and Veritas NetBackup.

The following example shows a standard architecture that you can use to implement data security either on a motherboard (Figure 3) or as a plug-in PCI interface card (Figure 4).

Implementing Data Security

A Virtex FPGA-based co-processing solution for data security enables x86-based appliances to scale the performance bottlenecks encountered in a pure software approach. A typical x86 appliance for a mid-range to enterprise data center server comprises:

- Dual or quad CPUs
- High-speed bus (PCI Express/HyperTransport)
- More than 1 GB RAM
- Integrated GE NICs
- Internal storage

Even though these are high-performance general processors, performing encryption services will consume lots of CPU cycles for encrypting and decrypting packets, leaving little processing time for other services. By downsizing the CPU and other hardware requirements and adding a Virtex FPGA as a hybrid solution, you can optimize the hardware to perform the following functions:

Generic CPU:

- Load and run OS
- Manage peripherals
- Manage connectivity (GE/FC/iSCSI)
- Run applications (e-mail/database)
- Dataflow control
- Key management

Virtex Co-Processor:

- High-performance encryption and decryption
- High-performance compression and decompression
- High-performance protocol conversion

You can implement services such as encryption, compression, and protocol conversion in a small Virtex-4 device, with leftover slices for additional services. By implementing core services in programmable hardware logic, you can take advantage of performance gains and reprogram the hardware to add and modify hardware-assisted services as business requirements change.

Conclusion

Software services provide a competitive advantage by delivering storage services over a wide area. Xilinx Virtex-II Pro or Virtex-4 FPGA-based co-processing solutions provide a flexible and scalable high-performance architecture for delivering secure file-level solutions to branch and remote offices.

By offering proven Ethernet and Fibre Channel IP cores, as well as third-party encryption, compression, and network security solutions, Xilinx and its partners allow developers with flexible multi-protocol IP cores to reduce time-to-market delays and high NRE costs associated with ASIC-based solutions.

For more information and technical details on Xilinx and third-party IP cores for storage services co-processing solutions, visit *www.xilinx.com/storage*, or e-mail *storage@xilinx.com*.

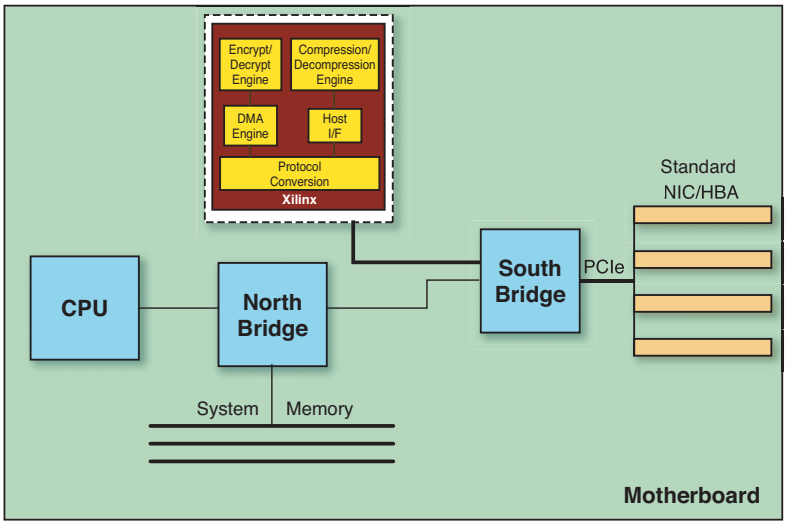


Figure 3 – Built-in co-processor on a motherboard

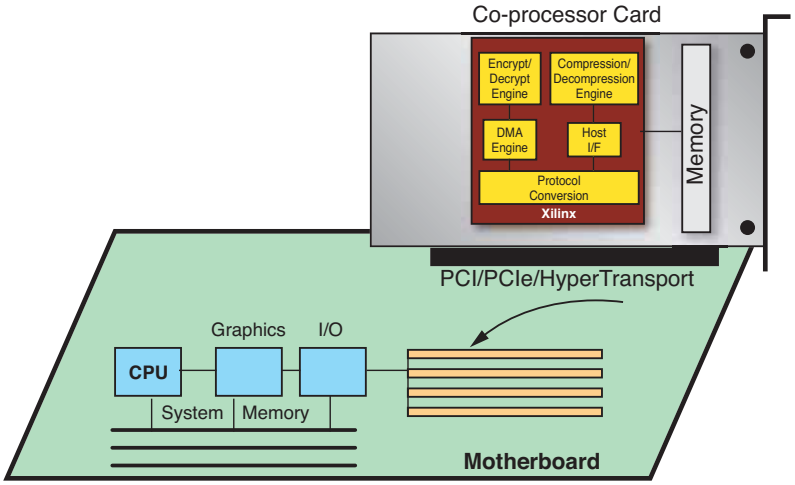
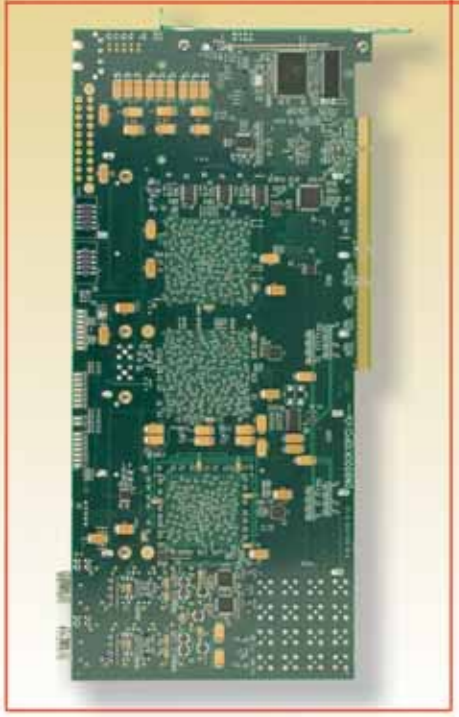


Figure 4 – Plug-in module for motherboard

VIRTEX-4

INCREDIBLY

INTERCONNECTED



Partitioning your design onto the 3.7 million ASIC gates (LSI measure) of our new board is a lot easier. With three of the biggest, fastest, new Xilinx Virtex-4 FPGAs this PCI hosted logic prototyping system takes full advantage of the integrated ISERDES/OSERDES. 400MHz LVDS differential communication with 10X multiplexing means more than 1800 signals between FPGA A & B. Synplicity Certify™ models are provided for partitioning assistance.

A dedicated PCI Bridge (533mb/s data transfer) means that all FPGA resources are available for logic emulation. Other features are designed to ease your prototyping job:

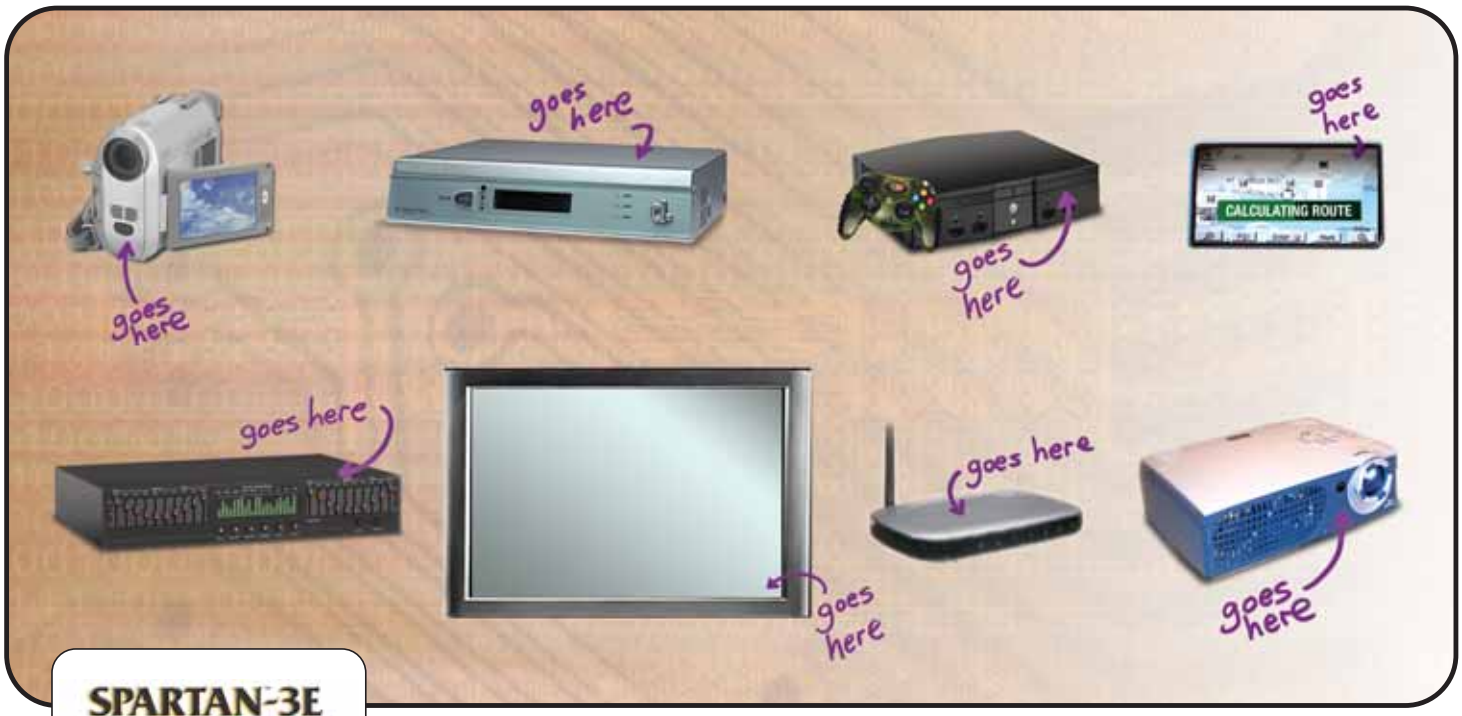
- 5 programmable clock synthesizers
- 2 DDR2 SODIMMs (custom DIMMs for SSRAM, QDR, Flash ...)
- Two, 200 pin expansion connectors for daughter cards
- 10 GB/s serial I/O interfaces with SMA connectors and XFP or SFP Modules
- Configured by PCI, USB 2.0, or SmartMedia with partial reconfiguration support on all FPGAs

Various stuffing options and a wide selection of daughter cards let you meet your exact design requirements. Prices start at less than $10,000. Call The Dini Group today for the latest ASIC prototyping solution.

The DiNI Group

1010 Pearl Street, Suite 6 • La Jolla, CA 92037 • (858) 454-3419
Email: sales@dinigroup.com
www.dinigroup.com

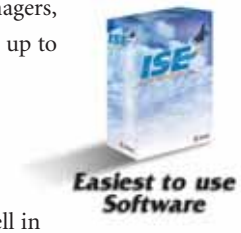
Priced to go.



SPARTAN-3E FPGA

The industry's first 100K gate FPGA for under $2.00*

Spartan-3E Platform FPGAs offer an amazing feature set for just $2.00! You get 100K gates, embedded multipliers for high-performance/low-cost DSP, plenty of RAM, digital clock managers, and all the I/O support you need. All this in production now with a density range up to 1.6 million gates.



Easiest to use Software

Perfect for digital consumer apps and much more!

With the Spartan-3E series, we've reduced the previous unit cost benchmark by over 30%. Optimized for gate-centric designs, and offering the lowest cost per logic cell in the industry, Spartan-3E FPGAs make it easy to replace your ASIC with a more flexible, faster-to-market solution. Compare the value for yourself . . . and get going on your latest design!

Over 10 million Spartan-3 devices shipped!

MAKE IT YOUR ASIC

XILINX®

The Programmable Logic Company℠

For more information visit
www.xilinx.com/spartan3e

* Pricing for 500K units, second half of 2006

Earth Friendly

Pb-free devices available now

A Programmable ExpressCard Solution

Philips Semiconductors and Xilinx offer a two-chip, low-cost, low-power, programmable PCI Express solution.

by Ho Wai Wong-Lam
Marketing Manager
Philips Semiconductors
ho.wai.wong-lam@philips.com

Abhijit Athavale
Sr. Marketing Manager
Xilinx, Inc.
abhijit.athavale@xilinx.com

ExpressCard modules are thinner, faster, and lighter slide-in cards introduced by PCMCIA to replace existing PC card and CardBus modules, which are primarily used for functionality upgrades and expansion in the notebook computer aftermarket. Likewise, the ExpressCard module is initially intended to serve the notebook computer aftermarket (Figure 1a).

In the desktop computer market (Figure 1b), both PC manufacturers and end users have a strong desire to shift towards the sealed-box model and to embrace easy and foolproof upgrades using the expansion slide-in cards that notebook computers have already deployed for a decade. This trend in desktop computers offers ease-of-use for end users and less service costs for PC manufacturers, as end users will not need to open their desktop computers and potentially damage the machine. ExpressCard technology will probably expand into PC-based appliances such as media boxes and IP set-top boxes.

ExpressCard Technology

The ExpressCard standard specifies that all ExpressCard modules can support either USB 2.0 or PCI Express (or both) and that all ExpressCard slots must accommodate modules designed to use either the 480 Mbps USB 2.0 interface or the single lane ("x1") 2.5 Gbps PCI Express interface. The ExpressCard standard also supports hot-plug and has stringent requirements on form factor, power-management modes, and thermal limits.

ExpressCard modules come in two sizes:

- ExpressCard/34:
  34 mm (W) x 75 mm (L) x 5 mm (H)
- ExpressCard/54:
  54 mm (W) x 75 mm (L) x 5 mm (H)

Host systems can provide any combination of slots for the 34- or 54-module widths. The 34-mm module will work in a slot designed for 54-mm-wide modules, but not vice versa.

Thermal limits are defined as "inside the slot" dissipation. The thermal limit is 1.3W for ExpressCard/34 modules and 2.1W for ExpressCard/54 modules.

ExpressCard technology makes active-state L0s and active-state L1 power management mandatory for PCI Express. Active state power management (ASPM) allows very aggressive hardware-initiated power management functionality that goes beyond what is achievable by using PCI-PM and ACPI (advanced configuration and power interface) software power-management techniques.

Hotplug functionality is a well-established part of CardBus and USB specifications, and is also supported by the PCI Express specification. ExpressCard module users can install and remove modules at any time without powering down the host computers. Furthermore, auto-detection and configuration of PCI Express and USB 2.0 allows a host system to support ExpressCard technology without using an external slot controller. A small ExpressCard power switch is required to control power to the slot.

Windows XP and 2000 do not provide native discovery support for PCI and PCI Express devices. But with Windows Longhorn, graceful removal and discovery of PCI Express devices will be as seamless as USB devices today.

Figure 1 – ExpressCard modules in (a) notebook and desktop (b) computers (Photo reproduction permission granted by PCMCIA)

PX1011A PCI Express PHY and Spartan-3E FPGAs

Together, Philips Semiconductor and Xilinx offer a low-cost programmable PCI Express solution for the emerging ExpressCard market. Our solution provides fully compliant x1 PCI Express endpoints using a Philips PX1011A PCI Express PHY, a Xilinx® Spartan™-3E FPGA, and a Xilinx LogiCORE™ PCI Express IP core. The solution meets the requirements of the ExpressCard specification with its low power, advanced packaging, signal integrity, and full compliance with the PCI Express specification. In addition, the programmable, reconfigurable fabric of the solution enables you to build innovative ExpressCard applications such as multi-function cards or intelligent peripherals and also reduces costs associated with part inventory and qualification.

Many emerging ExpressCard applications may not be able to utilize existing semiconductor devices because of package, power, signal integrity, or other technical requirements. Instead of waiting for chips that meet the requirements to be manufactured, our two-chip proven programmable solution offers you a low-risk, cost-effective alternative with a short time to market and a longer time in market. The final product can be designed, tested, and shipped in a manner of weeks – and you always have the chance to upgrade or repair the hardware in case of unforeseen problems.

Philips PX1011A

The Philips PX1011A is an x1 2.5 Gbps PCI Express PHY device 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. PX1011A is designed to serve both ExpressCard/34 and ExpressCard/54 applications. PX1011A has the following key features to support ExpressCard applications:

- Low power dissipation in normal L0 mode (<300 mW), including I/O
- Small and thin 81-pin package (9 x 9 x 1.05 mm), occupying only a small fraction of the real estate available on the ExpressCard/34 module
- Supports active-state L0s and L1 modes
- In the optional L2 mode, you can use a sideband WAKE# signal as a wake-up mechanism for the L2 power mode

Xilinx Spartan-3E FPGA and PCI Express IP Core

The Xilinx PCI Express IP core is designed for the Spartan-3E FPGA and has a small resource footprint. It implements complete logical and transaction layers as defined by the PCI Express specification, leaving approximately 50% of resources for designing in a Spartan-3E500, for example.

The Spartan-3E FPGA communicates with the PX1011A PHY using the source-synchronous 250 MHz PXPIPE standard – based on SSTL_2 I/O – and provides abundant block RAM resources that you can use to implement transmit-and-receive side buffers to store transaction layer packets. The Spartan-3E500 FPGA is available in very small ball-grid-array packages that meet the mechanical requirements of the ExpressCard specification. The PCI Express IP core has the following features required by ExpressCard applications:

- Fully compliant to the PCI Express base specification v1.1
- 2.5 Gbps bandwidth in each direction
- Resource usage
- 5,408-5,708 LUTs
- Six block RAMs
- 3,920-4,017 flip-flops
- PCI/PCI Express power-management functions
- Active-state power management (ASPM)
- Programmed power management (PPM)
- Compatible with current PCI software model

The combined Philips/Xilinx PCI Express solution is low in power dissipation (700-800 mW), making it suitable for both ExpressCard/34 and ExpressCard/54 applications.

Furthermore, the joint solution has successfully completed PCI Express compliance tests administered at PCI-SIG Compliance Workshop #45 in June 2005, and is listed on PCI-SIG Integrators List.

Using the Philips/Xilinx solution, applications can fully benefit from the high 2.5 Gbps throughput offered by PCI Express, while enjoying the flexibility of a programmable solution.

DVB-T ExpressCard/34 Demonstrator

At the Consumer Electronics Show (CES) in January 2006, we demonstrated the world's first programmable ExpressCard application, comprising the Philips PX1011A PCI Express PHY and PDD 2016 DVB-T module (Figure 2), as well as a Xilinx Spartan-3E FPGA with the optimized Xilinx PCI Express IP core (Figure 3).

Figure 2 – The Philips/Xilinx CES 2006 DVB-T ExpressCard/34 module demonstrates live digital TV reception on a notebook computer.

Figure 3 – Xilinx Spartan-3E FPGA and Philips PX1011A on a DVB-T ExpressCard/34 demonstrator

The Philips DVB-T ExpressCard/34 demonstrator resides in a notebook computer ExpressCard slot and receives DVB-T signals from a live TV transmitter. Streaming TV reception is displayed on a notebook computer.

Philips RF Solutions chose the joint Philips/Xilinx solution to convert the MPEG transport stream to PCI Express packets (required for communication with the host processor) based on the following benefits:

- Lower total cost than competitive solutions
- Ability to future-proof for further product enhancements and keep up with ExpressCard standard modifications
- Increased reusability across customer products

Conclusion

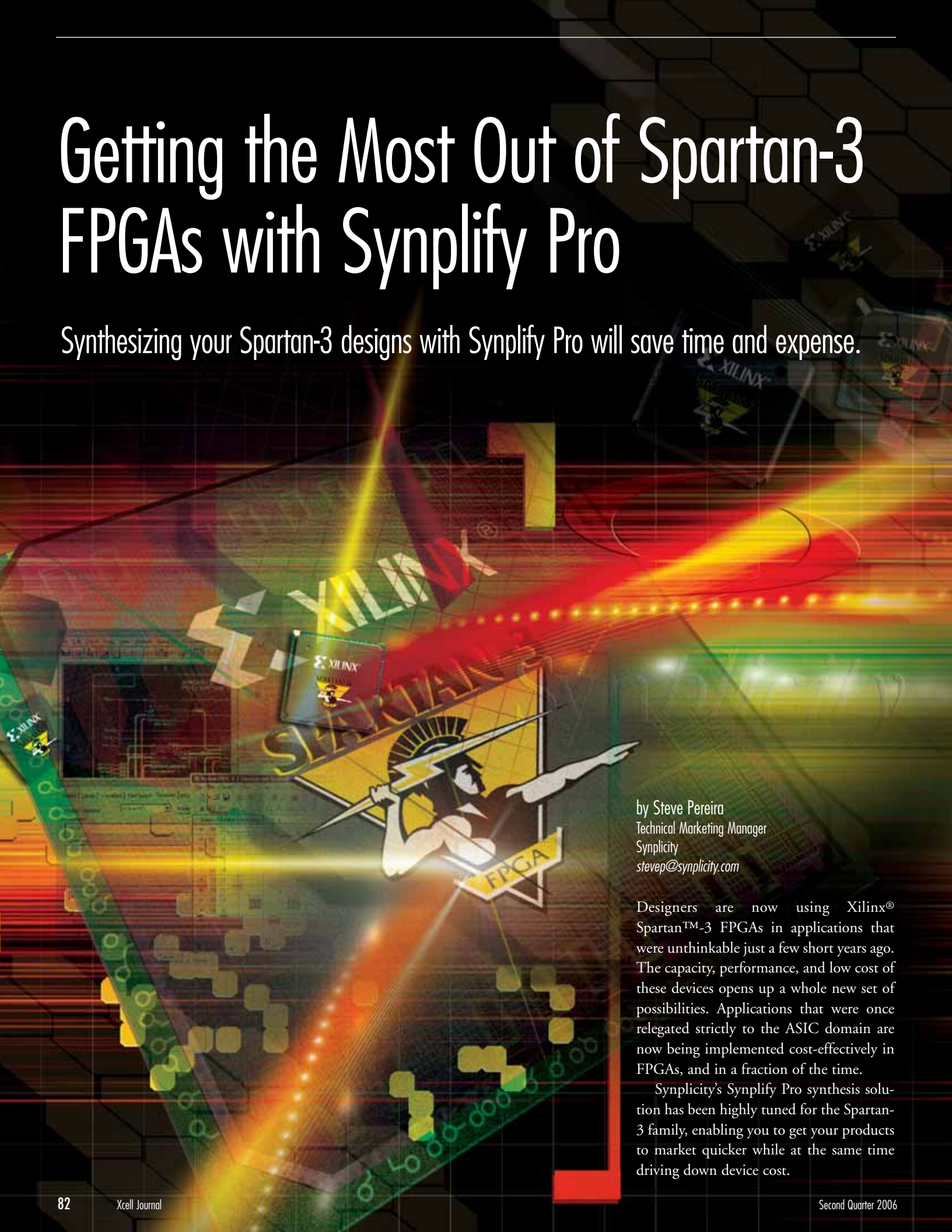
As ExpressCard technology becomes ubiquitous initially in notebook computers, it will then make headway into desktop computers and other PC-based appliances such as media boxes and IP set-top boxes. Our joint Philips/Xilinx ExpressCard solution provides a low-cost, low-power programmable solution to enable many emerging applications in the coming years.

For more information, please visit:

- *www.standardics.philips.com/products/pcie/phys/*
- *www.xilinx.com/pciexpress*
- *www.xilinx.com/spartan3e*
- *www.xilinx.com/prs_rls/silicon_spart/0603philips.htm.*

Getting the Most Out of Spartan-3 FPGAs with Synplify Pro

Synthesizing your Spartan-3 designs with Synplify Pro will save time and expense.



by Steve Pereira
Technical Marketing Manager
Synplicity
stevep@synplicity.com

Designers are now using Xilinx® Spartan™-3 FPGAs in applications that were unthinkable just a few short years ago. The capacity, performance, and low cost of these devices opens up a whole new set of possibilities. Applications that were once relegated strictly to the ASIC domain are now being implemented cost-effectively in FPGAs, and in a fraction of the time.

Synplicity's Synplify Pro synthesis solution has been highly tuned for the Spartan-3 family, enabling you to get your products to market quicker while at the same time driving down device cost.

Reducing Cost for Spartan-3 Devices

Synplify Pro software is a true timing-driven synthesis product. Basically, this means that critical paths are speed-optimized, while other non-critical paths are optimized for area. In other words, once you have met your timing constraints, the Synplify tool will then turn to optimizing for area, allowing you to meet your timing goals in the smallest device possible. The effect of timing-driven synthesis is huge area savings, while still meeting your timing goals.

Figure 1 is a graph representing the requested clock frequency versus the actual clock frequency after place and route of a design run through Synplify Pro. The Synplify Pro timing estimate follows the requested frequency line until other critical paths start appearing that the tool cannot optimize any further; hence the performance starts to degrade.

Figure 2 shows the LUT count for the same design. This underlines the importance of true timing-driven synthesis: while the requested frequency is low, the LUT count is low. The LUT count only begins to increase when the requested frequency is raised.

Other synthesis technologies tend to optimize for either area or timing. With this approach, a graph similar to Figure 1 would show a stair-step function as opposed to a smooth result matching the requested timing. The difference is that you are much more likely to be able to fit the design into a smaller part when using a timing-driven technology like that in the Synplify Pro tool. Using a smaller part results in huge cost savings when volumes are in the high hundreds or thousands of units.

Another area where Synplify Pro software can reduce area is in the automatic mapping of logic to dedicated resources within the Spartan-3 architecture. Synplify Pro software automatically extracts ROMs, RAMs, SRLs, and global control resources. With the extraction of these logic modules comes an increase in performance and in most cases a decrease of logic in slices, thus reducing LUT count and device cost.

There are a couple of attributes that can also decrease area:

- Resource sharing. Starting with Synplify 8.0, resource sharing is timing-driven; you should always leave this switch on.
- Area attributes for RAMs. For example, if you have a 1,400 x 33 RAM, you can map it in one of two ways:
  - Four tall RAM B16s. This will provide the fastest implementation without extra logic.
  - Three wide RAM B16s. Although this generates one less RAM, extra decode logic is generated.

How can Synplify Pro software reduce cost for Spartan devices? The answer is threefold:

1. For designs that have challenging performance goals, the Synplify Pro tool has very advanced logic optimizations that can meet the requested frequency and even allows you to choose a slower speed grade device. Switching to a lower speed grade can drastically reduce cost on high production runs.
2. For designs that have performance goals, size is more an issue. As the graphs in Figures 1 and 2 indicate,

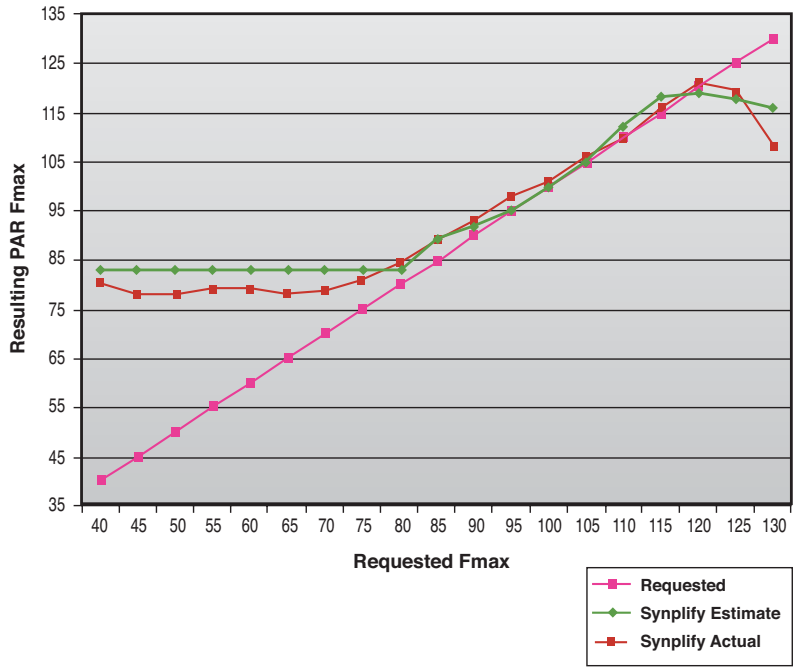


Figure 1 – Requested frequency verses place and route actual frequency using Synplify Pro

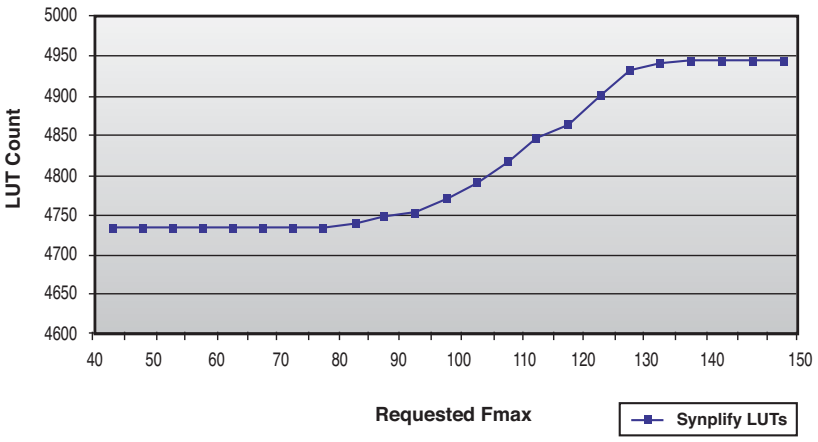


Figure 2 – Requested frequency verses LUT count using Synplify Pro

Synplify Pro software will provide performance on-demand. Although the requested performance is non-taxing, the LUT count remains very low. This can save you money by choosing a smaller device.

3. Design for modification. If modifications are required to an existing FPGA in a system, the cost of the change could be exceptionally high. If the required modifications fail to place or route, the only solution may be a complete redesign of the system, at an enormous cost. Designing with the Synplify Pro tool can reduce the probability of modification failure because of its performance benefits and performance-on-demand area savings. Other synthesis tools may require 90% utilization to implement a design, while Synplify may require only 70%. It is much easier to route a 70% full device than a 90% full device. Therefore, there is a higher probability of the modified design being successfully implemented with the lower utilization.

Conclusion

In today's challenging business environment, cost reduction is more important than ever. Together, Xilinx and Synplicity deliver everything you need to optimize your FPGA and your budget. By combining the performance of the industry's most popular FPGA synthesis product, Synplify Pro, with industry-leading FPGAs, you get the power to quickly meet aggressive performance goals on time and on budget.

To test out the Synplify Pro software for yourself and discover the cost savings that you can obtain, download a free, fully functional evaluation copy at *www.synplicity.com/downloads/download1.html*. When you install and run the tool, it will say that you do not have a license. Please follow the instructions and send the requested information to *license@synplicity.com*. You will be sent a temporary license immediately.

For more information about benchmarking Synplify Pro, visit *www.synplicity.com/literature/pdf/benchmarking_synplify.pdf*.

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 Solution

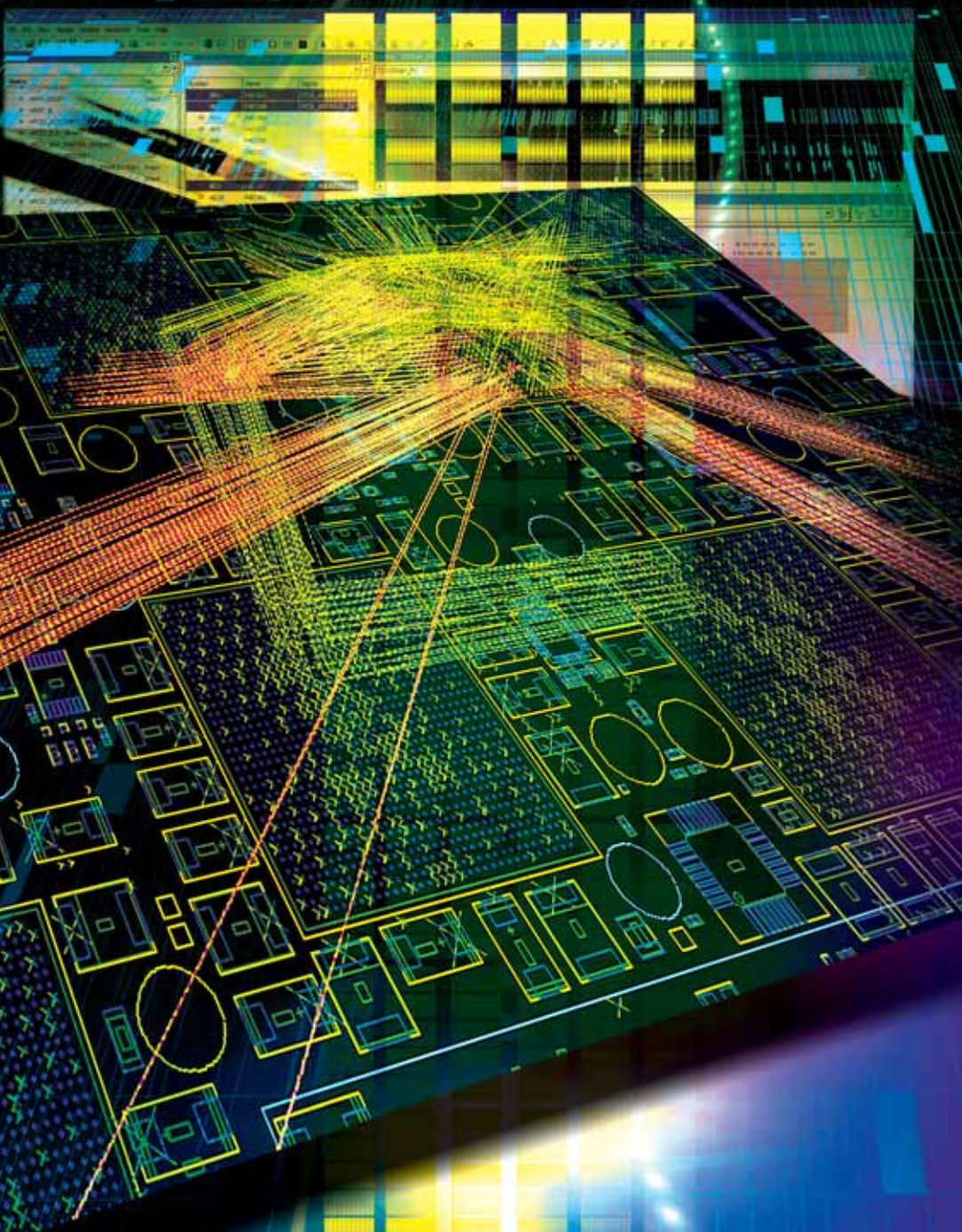
For more information on the Synplify Premier product and how it can help you quickly reach aggressive timing goals for your FPGAs, visit:
www.synplicity.com/products

Synplicity®

Simply Better Results

A Fundamentally Different Approach to FPGA I/O Design

Agilent used I/O Designer to achieve radical reductions in design cycle time.



by Bruce Riggins
Product Marketing Manager
Mentor Graphics
bruce_riggins@mentor.com

New problems require new solutions; this has never been truer in the realm of FPGA design than it is today. Devices with higher and higher pin counts have placed a significant strain on classic, over-the-wall methodologies. These techniques, which worked for many years, typically addressed FPGA and PCB design efforts as two distinct disciplines, where the FPGA designer defined the I/O assignments of the FPGA and passed those assignments to the PCB designer.

Ignoring for a moment the ramifications on the PCB, this process generally worked pretty well, especially with smaller devices and simple system performance requirements. But as FPGAs have grown larger and larger, nearing (and soon exceeding) 2,000 pins, these design philosophies simply fall apart. These devices demand a fundamentally different approach to FPGA I/O design - an approach that considers, concurrently, the effects that I/O assignments have on the FPGA and PCB.

In this article, I'll chronicle a successful deployment of this new process, based on a real-world design completed by Agilent Technologies using Mentor Graphics's I/O Designer.

Design Overview

Before discussing some of the reasons for Agilent's success, let's review the characteristics of their design:

- Eleven 1,148-pin Xilinx® Virtex™-4 LX series FPGAs
- Large DRAM buses on several devices (up to 250 pins in some cases)
- Multiple high-data-rate (10 Gbps) inter-FPGA buses approximately 48 pins wide
- Wide physical interfaces to PHY devices and backplane
- Very few spare pins
- Complex FPGA restrictions
- Fourteen differently-sized banks in the Virtex-4 device
- Clocks associated with I/O must be on specific signals
- I/O restrictions – for example, clock-capable inputs do not support outputting LVDS
- Mixed voltage environment (1.8V/2.5V/3V/3.3V) for different banks
- A 26-layer PCB
- Eight inner tracking (signal) layers
- 9,800 components
- 9,500 nets

By any measure, this was a very complex project.

From previous experience, Agilent estimated that they would need four to eight weeks per FPGA just for the I/O pin assignment process. In addition, I/O restrictions hinted that problems would be likely (and indeed, subsequent designs confirmed that). Agilent also knew that it was difficult to maintain links between the FPGA and board designs as the pinouts changed. Given these challenges, Agilent felt compelled to consider new design strategies.

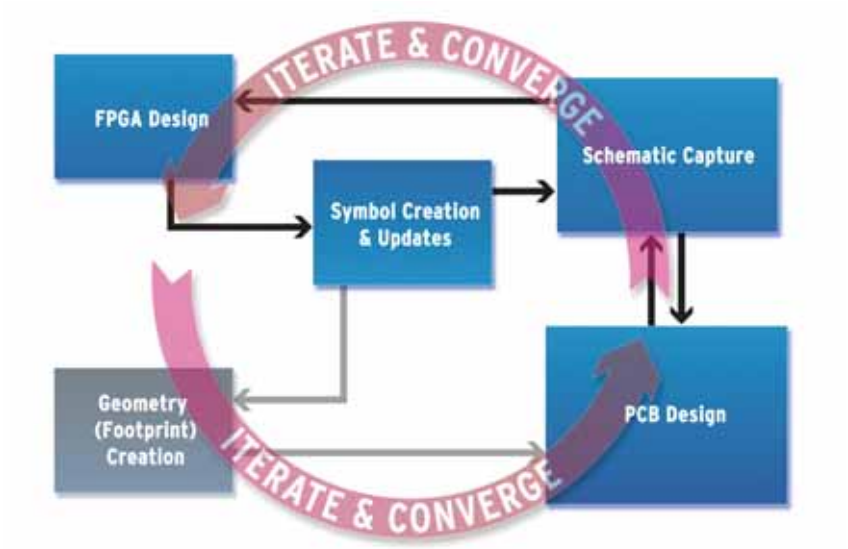


Figure 1 – A typical FPGA-based board-level design flow

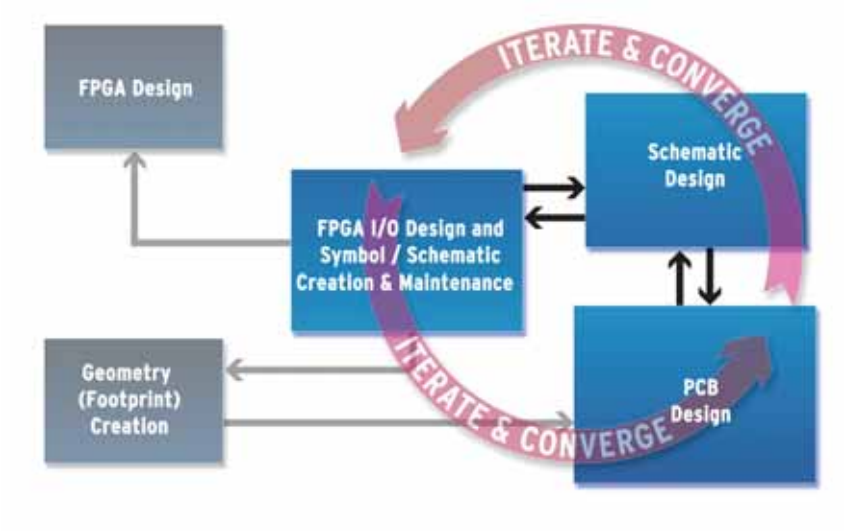


Figure 2 – A typical FPGA-based board-level design flow using I/O Designer

Simplifying the FPGA/PCB Design Process

Figures 1 and 2 illustrate typical, FPGA-based board-level design flows. On the surface these two figures look the same. But closer inspection reveals that the use of an FPGA/PCB co-design tool such as I/O Designer can simplify the flow, reduce error-prone tasks, and ultimately decrease design cycle times.

In a classic process (Figure 1), the FPGA designer works to develop an I/O assignment that is eventually deemed worthy of passing to the rest of the team. Before any downstream processes can proceed, the designer must create a schematic symbol (or symbols) and PCB footprint for the FPGA. The symbol and footprint are then used by the schematic and PCB designers in their portion of the design effort.

However, if either of these individuals require changes to the FPGA's pin assignment, they typically go back to the FPGA designer, who makes the changes in the FPGA tools and then feeds those changes to the librarian, who updates the schematic symbols. (Component footprints usually

change very little, if at all, since the geometries are mechanical in nature and are not affected by new pin assignments.) These are then given to the schematic designer and the project proceeds.

Simply speaking, converging on a completed design requires repeated iteration of these steps. This process is highly prone to mistakes because of the large amount of manual effort needed to keep the FPGA, symbols, schematics, and PCB layout synchronized.

Figure 2 also demonstrates a typical board-level design flow, but with I/O Designer introduced into the middle of the process. This is comparable to the flow adopted by Agilent, where I/O Designer was used for FPGA I/O pin assignments and for all FPGA-related symbol and schematic creation and maintenance. This allowed Agilent's engineers to focus their skills where they had the most value: the creation of intellectual property. They were thus able to iterate and converge on a smaller, more manageable subset of the overall system. (The engineers continued to use Xilinx FPGA tools at various points in the process for traditional place and route and synthesis and to check for compliance using Xilinx design rule checks.)

I/O Designer seeks to bring FPGA and PCB designers together in a common environment so that each team member can see the ramifications of FPGA pin assignments on the entire system. The mechanism that I/O Designer uses to accomplish this is conceptually simple: a view of the PCB with a dynamically assignable FPGA library element instantiated in place of the FPGA's typical PCB footprint.

The FPGA component, having come from a library of "intelligent" FPGA devices, assists in making proper pin assignments (checking for differential pairs, I/O banking rules, and SSO, for example). Because changes to these assignments are shown in real time, the potential effects on the rest of the board are immediately obvious.

Figure 3 is a screenshot of a section of Agilent's board in their PCB tool. Agilent used I/O Designer to optimize the ratsnest (some of which are shown in green and orange) across the entire board.

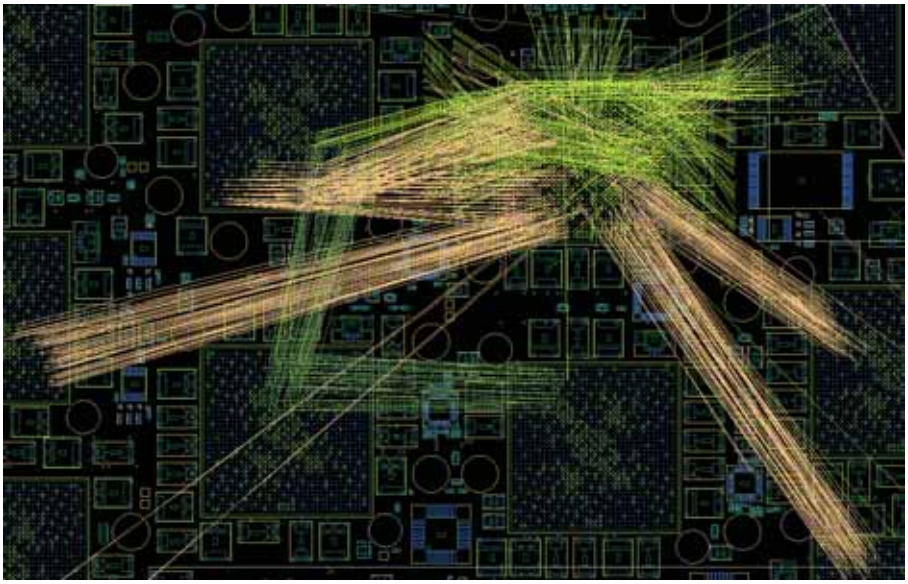


Figure 3 – A section of Agilent's 11-FPGA PCB

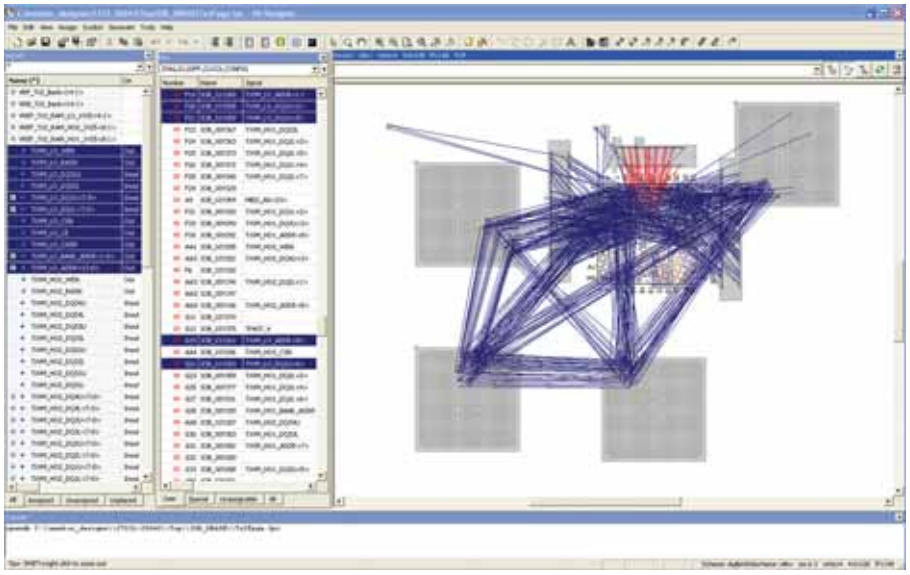


Figure 4 – A section of Agilent's design within I/O Designer

To demonstrate how I/O Designer assisted in this effort, Figure 4 shows the same Agilent design within I/O Designer after pin optimization. The area on the right is the "multi-component" window, which is a view of the PCB, and the FPGA currently undergoing active assignment. Other windows convey signal, pin, symbol (not shown), and console (transcripting) information. By displaying the inter-component connections from a board-level perspective, I/O Designer allowed Agilent to optimize the pin assignments for each FPGA.

Connecting to the FPGA Tools

I/O Designer is not an FPGA design tool: it can neither route nor synthesize. As such,

it can and must read and write the files necessary to close the loop with the FPGA designer (making pin-assignment changes in the schematic or PCB tools, or within I/O Designer, is of very little use if those changes cannot be communicated to the FPGA tools). To assist with this closure, I/O Designer can generate or update the associated place and route and synthesis constraints files through a simple pull-down menu.

The Overwhelming Benefits of Hierarchy

Flattened design techniques implement the project as one large collection of sheet after schematic sheet. In this approach, engineers begin by placing components on a schematic sheet. When that sheet is filled, they move to sheet two, then sheet three, and so on until the design is complete. This results in a database that conveys very little useful information about the logical architecture of the system. Without regular involvement in the project, it is nearly impossible to comprehend the design by looking at the schematics.

Hierarchical approaches, on the other hand, start with a top-level view of the system, then devolve the system into smaller and smaller more manageable subsystems. At some point, this devolution ends at the board or chip-level and the design begins.

Using hierarchy, every level of the system can be understood by anyone with even a cursory understanding of the project. As a result, the design database not only defines the physical realization of a subsystem, it documents the purpose of the subsystem in a way that a linear collection of schematic sheets simply cannot. Another benefit of hierarchy is that it forces engineers to consider the architecture of the design, not just its implementation.

"We realized early on that hierarchy would enable us to more fully exploit the features in I/O Designer," said Ross MacIsaac, a senior design engineer at Agilent's South Queensferry site in Scotland. "As such, although hierarchy was already used extensively within Agilent, we were careful to architect this system to ensure that I/O Designer, as well as our established EDA tools, would deliver us the most productivity. There is no doubt that the use of hierarchy reduced our design cycle time."

Why the focus on hierarchy? Because I/O Designer, when used in tight combination with a hierarchical design process, can provide significant productivity improvements. A structured, hierarchical approach enables I/O Designer to easily generate and – as the design matures, maintain – the symbols and schematics that make up the FPGA-related portion of the design. Thus, the conceptually simple burden of keeping the schematic view of the FPGA synchronized with the rest of the database, while eliminating manual, tedious, error-prone tasks, is considerably reduced or eliminated.

Caveats and Lessons Learned

Much of what Agilent undertook with this effort had never been attempted before, at least on this scale. Along the way, they learned a few lessons, as did Mentor. Here are some of their recommendations:

- Use as much of I/O Designer's automatic symbol and schematic creation and maintenance features as the process will allow.
- Finely partition the FPGA. Use an individual symbol for each signal bank, power block, configuration block, and JTAG block. This greatly reduces the potential for errors.
- Let I/O Designer handle the creation and maintenance of the schematic sheets that contain the FPGA logic banks. Do not mix power, configuration, and JTAG blocks with logic blocks, and partition the design in such a way as to separate the I/O Designer-maintained schematics from user-maintained schematics.
- Any team-based design approach, especially one with multiple FPGAs, requires that the project be forked and merged at several strategic points in the design cycle. Although I/O Designer can certainly help in these situations, getting the most from it in these cases requires careful planning and forethought. In other words, do not expect the tool to magically solve the project management problems. Take the time to learn how it can help, then deploy it accordingly.
- Cleverly partitioning the design into sections that can be attacked as homogenous blocks, and an application of hierarchy that enables the design to be quickly reconstructed from those blocks, can significantly improve design team efficiency.

Conclusion

New tools and approaches can significantly reduce project cycle times while simultaneously producing superior results. By cleverly architecting their design and molding their processes to take advantage of their EDA tools, Agilent was able to realize substantial productivity gains. Specifically, they were able to reduce their pin assignment effort from four to eight weeks per FPGA to one to two weeks per FPGA (including the time needed to compile the FPGA to check design rules), going from schematic start to PCB layout complete in less than 10 months while reducing overall system design effort by roughly 50%.

"We were aware of the emergence of I/O Designer," said MacIsaac. "We opted to introduce it into our process at a point in the design cycle that was later than we were comfortable with, as there was a strong preference to use proven tools.

"However," he continued, "the size of the task merited taking that risk. The effort involved in other schemes was just so much that it was expected to exceed any tool teething problems. Management was prepared to take the risk of introducing this new technology. This board was the right problem at the right time for the tool [I/O Designer] and the FPGAs. Both gave us the ability to do the design and alter it, quickly and reliably."

For more information about Mentor Graphics's I/O Designer, visit *www.mentor.com/products/pcb/expedition/system_design/io_designer.*

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** or call us at **800.547.3000.**

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

Mentor Graphics®

THE EDA TECHNOLOGY LEADER

X-ray vision for your designs

Agilent 16900 Series logic analysis system with FPGA dynamic probe

- *Increased visibility with FPGA dynamic probe*
- *Intuitive Windows® XP Pro user interface*
- *Accurate and reliable probing with soft touch connectorless probes*
- *16900 Series logic analysis system prices starting at $21,000*

Agilent Direct

Get a quick quote and/or FREE CD-ROM with video demos showing how you can reduce your development time.

U.S. 1-800-829-4444, Ad# 7909
Canada 1-877-894-4414, Ad# 7910
www.agilent.com/find/new16900
www.agilent.com/find/new16903quickquote

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

The FPGA dynamic probe, when combined with an Agilent 16900 Series logic analysis system, allows you to access different groups of signals to debug inside your FPGA—without requiring design changes. You'll increase visibility into internal FPGA activity by gaining access up to 64 internal signals with each debug pin.

You'll also be able to speed up system analysis with the 16900's hosted power mode—which enables you and your team to remotely access and operate the 16900 over the network from your fastest PCs.

The intuitive user interface makes the 16900 easy to get up and running. The touch-screen or mouse makes it simple to use, with prices to fit your budget. Optional soft touch connectorless probing solutions provide unprecedented reliability, convenience and the smallest probing footprint available. Contact Agilent Direct today to learn more.

Agilent Technologies
dreams made real

Simulating SSO Noise

Electromagnetic and transient circuit simulations can accurately predict SSO noise performance.

by Michael Brenneman
Technical Director
Ansoft Corporation
meb@ansoft.com

Mark Alexander
Senior Product Application Engineer
Xilinx, Inc.
mark.alexander@xilinx.com

Lawrence Williams, Ph.D.
Director of Business Development
Ansoft Corporation
williams@ansoft.com

System designs are evolving toward physical interfaces that employ fast edges, coupled with clock rates reaching into the hundreds of megahertz. Proper power and ground distribution for such systems is critical for reliable voltage levels and interface signal switching. This is particularly important in system designs that use devices with thousands of pins, of which hundreds can be I/Os carrying active signals. Accurately predicting simultaneous switching output (SSO) noise in the presence of this level of switching activity is one of the major factors for ensuring adequate signal integrity in such systems.

In this article, we will describe a repeatable, accurate, and practical methodology for SSO analysis based on three-dimensional electromagnetic (EM) field simulation and transient circuit simulation. The target device is the latest Xilinx® Virtex™-4 XC4VLX60 FPGA in an FF1148 package and its companion 24-layer FR4 test board.

SSO Noise

When multiple output drivers change state at the same time, the changing current in the power system induces a voltage that results in power supply disturbances. These disturbances, referred to as SSO noise, can cause undesired transient behavior among output drivers, input receivers, or internal logic. Simultaneous switching noise can be attributed to a poor signal-to-ground ratio or to return path discontinuities.

System design requirements specify the number and type of I/O output drivers that can be switched simultaneously while still maintaining a safe level of SSO noise. You can apply electromagnetic simulation to accurately predict SSO noise and evaluate interconnect behavior. You can then use this information to avoid return path discontinuities, minimize total return path inductance, and optimize the size and location of decoupling capacitors.

Figure 1 provides a simplified schematic view illustrating the mechanism that causes SSO noise. A BGA package is shown sandwiched between PCB power and ground planes. User I/O "A" is set to a logic low. In this condition, the voltage measured at "D" should remain at the ground potential. Inductor L_{gnd} represents the inductance of the BGA and PCB return paths. A voltage V_{glitch} will be observed at "D" as the user I/O "C" closes because of the transient current flowing though the inductor. The glitch, known as SSO noise, is caused by the L di/dt voltage that appears across the ground inductance.

The switching currents generate a magnetic field between conductors, resulting in a mutual inductance that is the main contributor to crosstalk (mutual capacitance being the lesser contributor). This mutual inductance will induce current on the victim line in the opposite direction of the driving current (Lenz's law). The coupling of energy from one line to another varies with the separation between signal pathways and the length, or in this case, height = via + ball + package thickness. It also scales with the time rate of change of the switching current.

The return path inductance depends on the BGA package, PCB via, and trace routing. Various current loops exist and couple to one another. Indeed, both self and mutual inductances exist between all vias. These inductances depend on the spatial arrangement and proximity of the power and ground pins and are therefore a focus of both package design and pin assignment as well as package and PCB trace routing. These considerations were included in the design of the Virtex-4 XC4VLX60 FPGA FF1148 pack-

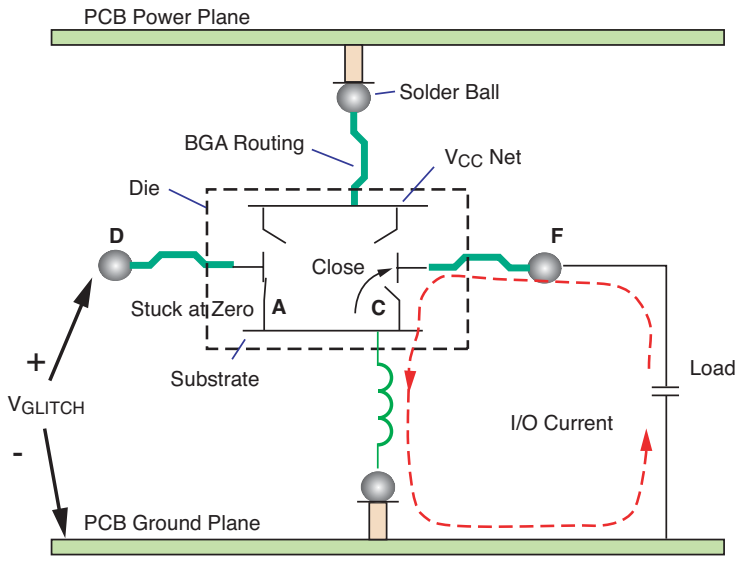


Figure 1 – Simplified view of SSO noise. A voltage V_{glitch} is observed as switch "C" closes. (Figure courtesy of Dr. Howard Johnson)

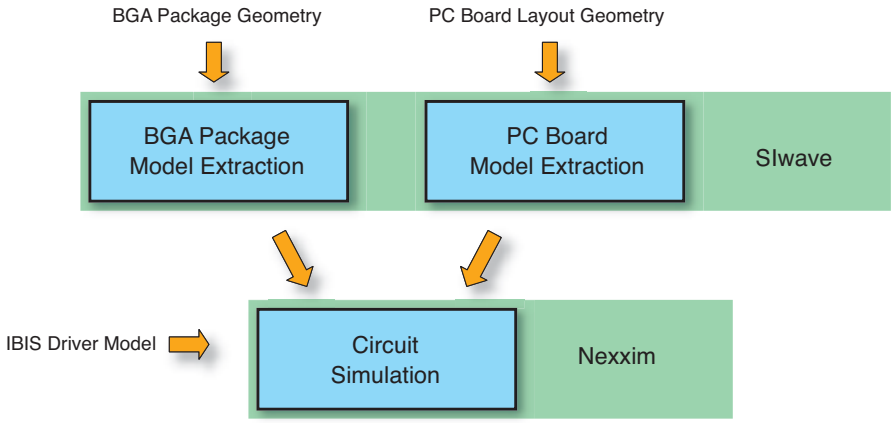


Figure 2 – We used a design flow that combines package/board electromagnetic extraction with advanced transient circuit simulation.

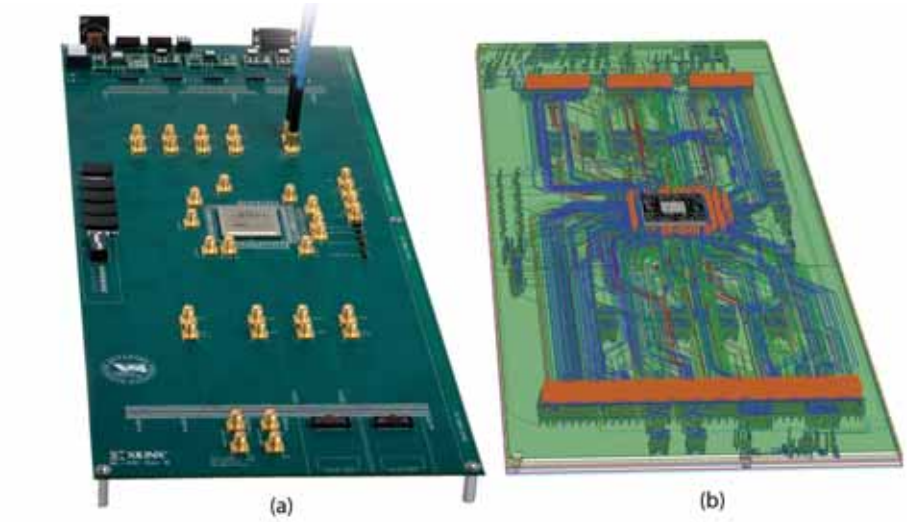


Figure 3 – We performed measurements and simulations on a BGA plus printed circuit board: (a) measurement set-up; (b) electromagnetic field simulation model.

age and pin assignments, and their efficacy was determined by EM simulation.

Package and PCB Configuration

The FF1148 package for the Virtex-4 FPGA uses a SparseChevron pinout style. The SparseChevron pinout style is tessellated with a regular array of power and ground pins to minimize power delivery system (PDS) inductance and to keep I/O signal return current paths very closely coupled to their associated I/O signal.

The via structures of the PCB and the package are the main sources of inductive crosstalk. The PCB is the primary source, as the board thickness is 3,635 µm and thus the current return loops are larger. The package is the next most significant source, with a thickness of 1,068 µm. The 50 Ohm transmission lines are not a significant contributor to inductive crosstalk in this design.

EM Simulations

Figure 2 depicts the simulation flow that we applied to complete this SSO analysis. Our goal was to perform transient simulation on a circuit that includes electromagnetic accuracy coupled with standard models for transistor drivers. We performed electromagnetic field simulation on the BGA package and PCB to extract equivalent circuit models for the three-dimensional effects.

Figure 3 provides views of the test board used for the measurements and the CAD model we used in the electromagnetic simulations. The models include an 8-layer BGA package and a 24-layer test board. Test points were included on the board and are brought out to the test equipment using SMA connectors. The BGA is a 34 x 34 mm package, with 1,148 solder balls.

Design automation is required to import the complex layout files into the EM simulator. Ansoftlinks provides the translation and editing functionality to use the data from third-party layout tools. For our simulations, the package layout was imported from Cadence APD and the board was imported from Mentor Graphics PADS (formerly PowerPCB).

We performed electromagnetic field simulations on 20 I/O interconnects (including test board and package), along with the nearest power and ground pins and their associated decoupling capacitors. We performed two simulations using the Ansoft SIwave full-wave field solver, characterizing the coupled behavior of package nets, board nets, and PDS in the frequency domain.

The package simulation results in a 49-port S-parameter file comprising 40 I/O ports, 5 "spy-hole" test-point ports, 2 VCCO_9 power supply ports, and 2 decoupling capacitor ports. The board simulation results in a 28-port S-parameter file comprising 23 I/O ports, 3 "spy-hole" test-point ports, 1 VCCO_9 power supply port, and 1 port for the voltage regulator module (VRM), the 2.5V power supply connection to the printed circuit board. The board simulation in SIwave includes all passive capacitor, resistor, and inductor components used for decoupling, terminations, and other functions.

To properly define the structure, SIwave allows easy editing of the imported layout database. Correct values from fabrication are entered for layer thickness and material properties (including frequency-dependent dielectrics). Automation allows for quick creation of solder balls and solder bumps, grouped electrical nodes from component pins, and ports. The focus of our investigation was on correlating measured versus simulated SSO noise levels on VCCO_9 and one victim net (driven low) when 20 I/Os switch simultaneously. The full-wave EM extraction of the package and board by SIwave captures the complex coupling behavior and ensures the accuracy of results during transient simulation.

Transient Simulation

We obtained the SSO effects by coupling the EM simulation data with transient circuit analysis. The extracted package and board models are excited by the IC drivers, which in this example were implemented using Xilinx Virtex-4 IBIS v3.2 models (LVCMOS 2.5V, 24 mA fast). We imported the S-parameters and IBIS models into Ansoft Designer and ran transient simulation on the full circuit using Ansoft's Nexxim circuit simulator. Figure 4(a) shows the final circuit with IBIS drivers at the package die side, on-package decoupling capacitors, package/board connection, board DC power, and board spy-hole terminations.

We observed the aggressor/victim crosstalk by clamping the die side of the victim trace to a fixed value (high or low) while examining the simulated transient voltage waveform appearing across the victim test node caused by changing signals on one or more neighboring terminals. The Nexxim circuit simulator performed transient simulations; it is specifically designed to address frequency-domain models such as S-parameters accurately in the time domain without introducing causality or passivity errors.

We conducted a range of transient tests to demonstrate the proper transient behavior of the EM models from the SIwave simulation. Figure 4(b) shows one test result on the package model that validates expected victim net crosstalk behavior. The crosstalk's inductive nature is confirmed by the victim noise spike, which is the inverse of the aggressor edge. The multiple crosstalk waveforms demonstrate the cumulative effect of adding more neighboring aggressors based on current return loop proximity.

Figures 4(c) and 4(d) show the correlated data for the victim net and VCCO_9, respectively. Replicating the test board in the EM simulator allows for a direct comparison of the measured results with the simulated SSO noise magnitudes at two different spy-hole locations. We achieved excellent agreement between hardware measurements and the simulation results for SSO noise magnitude.

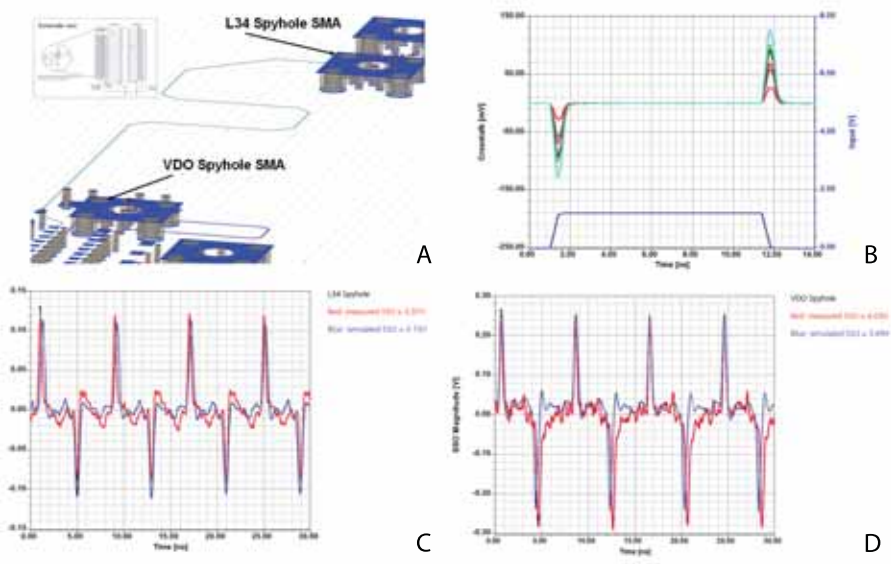


Figure 4 – Transient simulation results as computed by Nexxim: (a) simulation set-up and extracted S-parameters and IBIS drivers shown in schematic; (b) crosstalk caused by neighboring aggressors observed on victim pin; (c) measured results at L34 spy-hole compared to EM and transient simulation; (d) results at VDO spy hole.

Conclusion

Working together, Xilinx and Ansoft demonstrated a methodology to accurately predict SSO noise using advanced simulation tools and validated this with measurements. You can also use this powerful combination of simulation tools, Ansoft's SIwave and Nexxim, to develop future systems and meet specific SSO guidelines without having to build, test, and iterate.

The demand for high-performance electronics operating at faster data rates will continue to drive package and board complexity. To meet this demand, Ansoft is continually advancing the state of simulation software and working closely with leading vendors like Xilinx to handle larger problems without compromising accuracy. For more information, visit *www.ansoft.com.*

ESL Tools for FPGAs

Empowering software developers to design with programmable hardware.

by Milan Saini
Technical Marketing Manager
Xilinx, Inc.
milan.saini@xilinx.com

A fundamental change is taking place in the world of logic design. A new generation of design tools is empowering software developers to take their algorithmic expressions straight into hardware without having to learn traditional hardware design techniques.

These tools and associated design methodologies are classified collectively as electronic system level (ESL) design, broadly referring to system design and verification methodologies that begin at a higher level of abstraction than the current mainstream register transfer level (RTL). ESL design languages are closer in syntax and semantics to the popular ANSI C than to hardware languages like Verilog and VHDL.

How is ESL Relevant to FPGAs?

ESL tools have been around for a while, and many perceive that these tools are predominantly focused on ASIC design flows. The reality, however, is that an increasing number of ESL tool providers are focusing on programmable logic; currently, several tools in the market support a system design flow specifically optimized for Xilinx® FPGAs. ESL flows are a natural evolution for FPGA design tools, allowing the flexibility of programmable hardware to be more easily accessed by a wider and more software-centric user base.

Consider a couple of scenarios in which ESL and FPGAs make a great combination:

1. Together, ESL tools and programmable hardware enable a desktop-based hardware development environment that fits into a software developer's workflow model. Tools can provide optimized support for specific FPGA-based reference boards, which software developers can use to start a project evaluation or a prototype. The availability of these boards and the corresponding reference applications written in higher level languages makes creating customized, hardware-accelerated systems much faster and easier. In fact, software programmers are now able to use FPGA-based reference boards and tools in much the same way as microprocessor reference boards and tools.

2. With high-performance embedded processors now very common in FPGAs, software and hardware design components can fit into a single device. Starting from a software description of a system, you can implement individual design blocks in hardware or software depending on the applications' performance requirements. ESL tools add value by enabling intelligent partitioning and automated export of software functions into equivalent hardware functions.

ESL promotes the concept of "explorative design and optimization." Using ESL methodologies in combination with programmable hardware, it becomes possible to try a much larger number of possible application implementations, as well as rapidly experiment with dramatically different software/hardware partitioning strategies. This ability to experiment – to try new approaches and quickly analyze performance and size trade-offs – makes it possible for ESL/FPGA users to achieve higher overall performance in less time than it would take using traditional RTL methods.

Additionally, by working at a more abstract level, you can express your intent using fewer keystrokes and writing fewer lines of code. This typically means a much faster time to design completion, and less chance of making errors that require tedious, low-level debugging.

ESL's Target Audience

The main benefits of ESL flows for prospective FPGA users are their productivity and ease-of-use. By abstracting the implementation details involved in generating a hardware circuit, the tools are marketing their appeal to a software-centric user base (Figure 1). Working at a higher level of abstraction allows designers with skills in traditional software programming languages like C to more quickly explore their ideas in hardware. In most instances, you can implement an entire design in hardware without the assistance of an

experienced hardware designer. Software-centric application and algorithm developers who have successfully applied the benefits of this methodology to FPGAs include systems engineers, scientists, mathematicians, and embedded and firmware developers.

The profile of applications suitable for ESL methodologies includes computationally intensive algorithms with extensive inner-loop constructs. These applications can realize tremendous acceleration through the concurrent parallel execution possible in hardware. ESL tools have helped with successful project deployments in application domains such as audio/video/image processing, encryption, signal and packet processing, gene sequencing, bioinformatics, geophysics, and astrophysics.

ESL Design Flows

ESL tools that are relevant to FPGAs cover two main design flows:

1. High-level language (HLL) synthesis. HLL synthesis covers algorithmic or behavioral synthesis, which can produce hardware circuits from C or C-like software languages. Various partner solutions take different paths to converting a high-level design description into an FPGA implementation. How this is done goes to the root of the differences between the various ESL offerings.

   You can use HLL synthesis for a variety of use cases, including:

   - Module generation. In this mode of use, the HLL compiler can convert a functional block expressed in C (for example, as a C subroutine) into a corresponding hardware block. The generated hardware block is then assimilated in the overall hardware/software design. In this way, the HLL compiler generates a sub-module of the overall design.

     Module generation allows software engineers to participate in the overall system design by quickly generating, then integrating, algorithmic hardware components. Hardware engineers seeking a fast way to prototype new, computation-oriented hardware blocks can also use module generation.

   - Processor acceleration. In this mode of use, the HLL compiler allows time-critical or bottleneck functions running on a processor to be accelerated by enabling the creation of a custom accelerator block in the programmable fabric of the FPGA. In addition to creating the accelerator, the tools can also automatically infer memories and generate the required hardware-software interface circuitry, as well as the software device drivers that enable communication between the processor and the hardware accelerator block (Figure 2). When compared to code running on a CPU, FPGA-accelerated code can run orders of magnitude faster while consuming significantly less power.

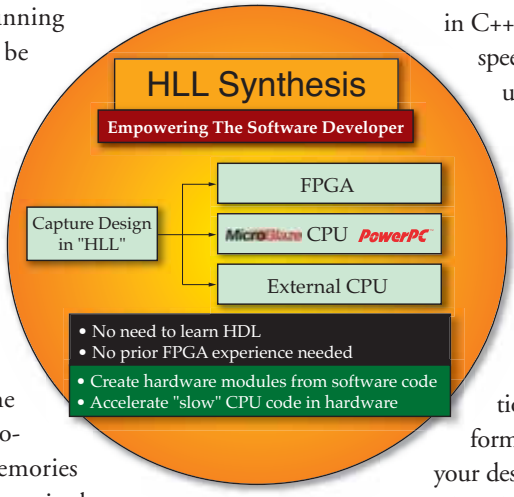


Figure 1 – Most of the ESL tools for FPGAs are targeted at a software-centric user base.

2. System modeling. System simulations using traditional RTL models can be very slow for large designs, or when processors are part of the complete design. A popular emerging ESL approach uses high-speed transaction-level models, typically written in C++, to significantly speed up system simulations. ESL tools provide you with a virtual platform-based verification environment where you can analyze and tune the functional and performance attributes of your design. This means much earlier access to a virtual representation of the system, enabling greater design exploration and what-if analysis. You can evaluate and refine performance issues such as latency, throughput, and bandwidth, as well as alternative software/hardware partitioning strategies. Once the design meets its performance objectives, it can be committed to implementation in silicon.

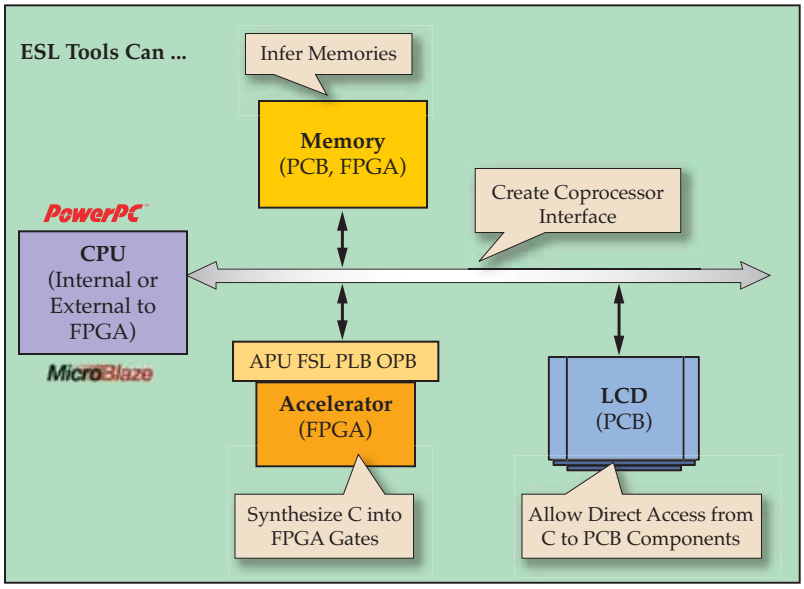


Figure 2 – ESL tools abstract the details associated with accelerating processor applications in the FPGA.

> ...today's ESL technologies are ready to deliver substantial practical value to a potentially large target audience.

The Challenges Faced by ESL Tool Providers

In relative terms, ESL tools for FPGAs are new to the market; customer adoption remains a key challenge. One of the biggest challenges faced by ESL tool providers is overcoming a general lack of awareness as to what is possible with ESL and FPGAs, what solutions and capabilities already exist, and the practical uses and benefits of the technology. Other challenges include user apprehension and concerns over the quality of results and learning curve associated with ESL adoption.

Although paradigm shifts such as those introduced by ESL will take time to become fully accepted within the existing FPGA user community, there is a need to tackle some of the key issues that currently prohibit adoption. This is particularly important because today's ESL technologies are ready to deliver substantial practical value to a potentially large target audience.

Xilinx ESL Initiative

Xilinx believes that ESL tools have the promise and potential to radically change the way hardware and software designers create, optimize, and verify complex electronic systems. To bring the full range of benefits of this emerging technology to its customers and to establish a common platform for ESL technologies that target FPGAs in particular, Xilinx has proactively formed a collaborative joint ESL Initiative with its ecosystem partners (Table 1).

The overall theme of the initiative is to accelerate the pace of ESL innovation for FPGAs and to bring the technology closer to the needs of the software-centric user base. As part of the initiative, there are two main areas of emphasis:

1. Engineering collaboration. Xilinx will work closely with its partners to continue to further increase the value of ESL product offerings. This will include working to improve the compiler quality of results and enhance tool interoperability and overall ease-of-use.
2. ESL awareness and evangelization. Xilinx will evangelize the value and benefits of ESL flows for FPGAs to current and prospective new customers. The program will seek to inform and educate users on the types of ESL solutions that currently exist and how the various offerings can provide better approaches to solving existing problems. The aim is to empower users to make informed decisions on the suitability and fit of various partner ESL offerings to meet their specific application needs. Greater awareness will lead to increased customer adoption, which in turn will contribute to a sustainable partner ESL for FPGAs ecosystem.

Getting Started With ESL

As a first step to building greater awareness on the various ESL for FPGA efforts, Xilinx has put together a comprehensive ESL website. The content covers the specific and unique aspects of each of the currently available partner ESL solutions and is designed to help you decide which, if any, of the available solutions are a good fit for your applications. To get started with your ESL orientation, visit *www.xilinx.com/esl*.

Additionally, Xilinx has also started a new ESL for FPGAs discussion forum at *http://toolbox.xilinx.com/cgi-bin/forum*. Here, you can participate in a variety of discussions on topics related to ESL design for FPGAs.

Conclusion

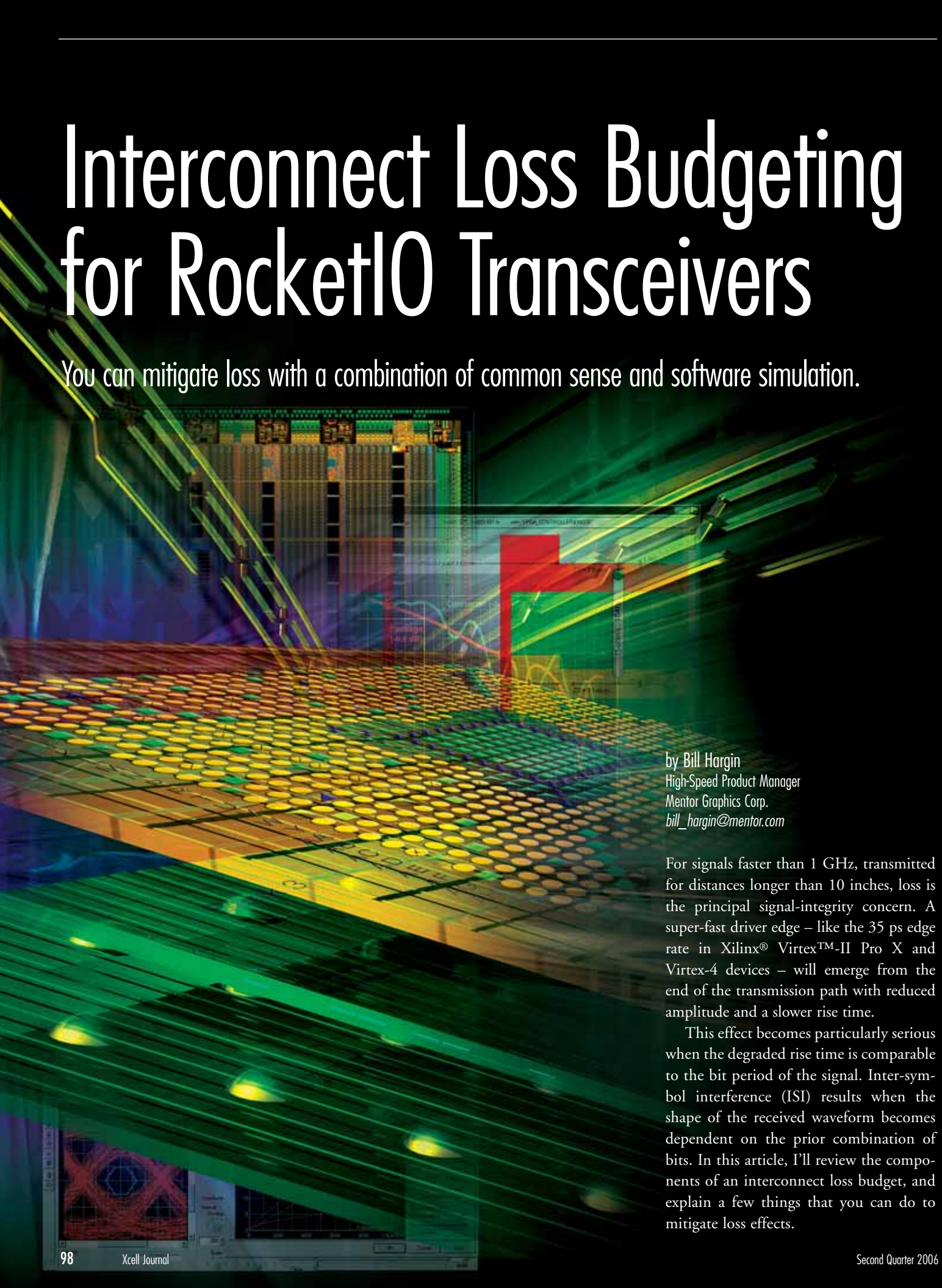
ESL tools for FPGAs give you the power to explore your ideas with programmable hardware without needing to learn low-level details associated with hardware design. Today, you have the opportunity to select from a wide spectrum of innovative and productivity-enhancing solutions that have been specifically optimized for Xilinx FPGAs. With the formal launching of the ESL Initiative, Xilinx is thoroughly committed to working with its third-party ecosystem in bringing the best-in-class ESL tools to its current and potential future customers. Stay tuned for continuing updates and new developments.

Partner	FPGA Synthesis	Xilinx CPU Support	FPGA Computing Solution
Celoxica	●	●	Handel-C, SystemC to gates
Impulse	●	●	Impulse C to gates
Poseidon	●	●	HW/SW partitioning, acceleration
Critical Blue	●		Co-processor synthesis
Teja		●	C to multi-core processing
Mitrion	●		Adaptable parallel processor in FPGA
System Crafter	●		SystemC to gates
Bluespec	●		SystemVerilog-based synthesis to RTL
Nallatech	●	●	High-performance computing

Table 1 – Xilinx ESL partners take different approaches from high-level languages to FPGA implementation.

Interconnect Loss Budgeting for RocketIO Transceivers

You can mitigate loss with a combination of common sense and software simulation.



by Bill Hargin
High-Speed Product Manager
Mentor Graphics Corp.
bill_hargin@mentor.com

For signals faster than 1 GHz, transmitted for distances longer than 10 inches, loss is the principal signal-integrity concern. A super-fast driver edge – like the 35 ps edge rate in Xilinx® Virtex™-II Pro X and Virtex-4 devices – will emerge from the end of the transmission path with reduced amplitude and a slower rise time.

This effect becomes particularly serious when the degraded rise time is comparable to the bit period of the signal. Inter-symbol interference (ISI) results when the shape of the received waveform becomes dependent on the prior combination of bits. In this article, I'll review the components of an interconnect loss budget, and explain a few things that you can do to mitigate loss effects.

Balancing Your Loss Budget

The most general way to control loss on a backplane is to avoid impedance discontinuities through transitions – keeping the same ratio between L and C throughout the signal path. But before going into specific strategies, let's discuss the loss budget as a whole.

Every link needs a link budget, with typical end-to-end attenuation budgets for today's multi-gigabit-per-second SERDES links ranging from 10-15 dB. (Although SATA can be as low as 1.8 dB and USB 2.0 is 6 dB.)

As with any budgeting exercise, exact (dB) figures for each item are not required, but it is important that you have a budget and that you consider each major element. If you are hunting for the last decibel, just knowing that one part of the system is twice as important as another provides a huge design benefit as far as knowing where to focus your time.

The component manufacturer will typically explicitly specify your loss budget, but it is also possible to derive the budget. The relationship for estimating your interconnect loss budget is:

$$\text{Attenuation Budget (dB)} = 20 \times \log \left(V_{RX,\,min} / V_{TX,\,min} \right)$$

where V_{TX} is the minimum (or target) voltage output at the driver and V_{RX} is the receiver's or eye mask's minimum input threshold. In the case of RocketIO™ technology in Virtex-4 FPGAs, an example attenuation budget is calculated as:

$$\text{Virtex-4 Budget} = 20 \log (200/400) = -6 \text{ dB}$$

This loss figure reveals your attenuation requirements before employing pre-emphasis or equalization. Let's go through the design issues in detail, first outlining the path from die at the transmitter all the way to the die at the receiver across a backplane interconnect.

Figure 1 shows a typical backplane. At a high level, there are two areas on a backplane interconnect where loss can occur:

- Transition regions that include the BGA package, vias, and connectors
- The line card and the backplane itself

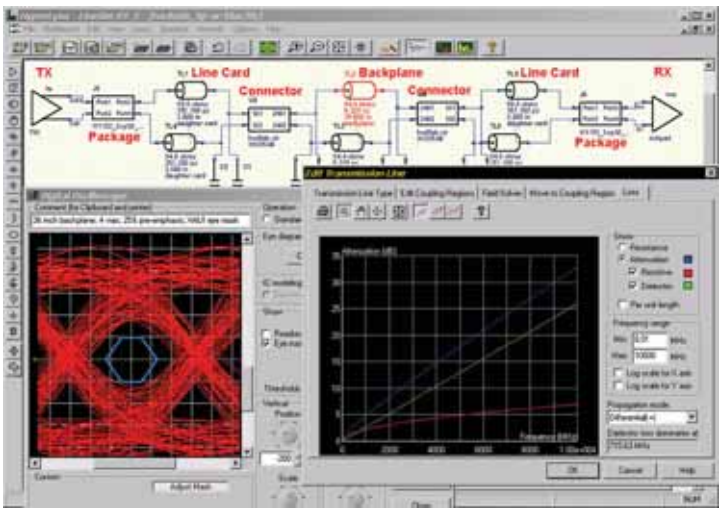


Figure 1 – A typical transmission-line schematic for a RocketIO backplane (top). The HyperLynx LineSim GHz Loss Viewer shows resistive, dielectric, and total loss (dB) as a function of frequency (MHz) for the backplane (bottom right). Simulation results for the Xilinx SIS Kit backplane example with pre-emphasis bumped up to 25 percent (bottom left).

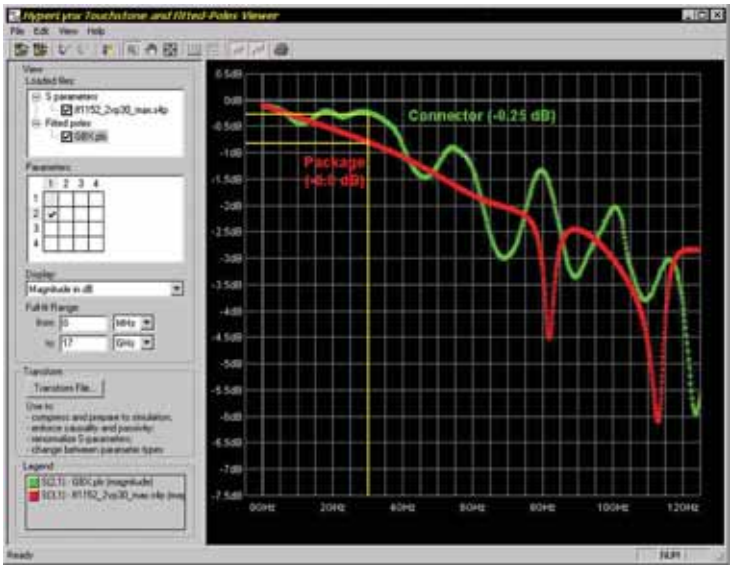


Figure 2 -The HyperLynx Touchstone Viewer shows BGA package loss (red) and connector loss (green) as a function of frequency. The values at 3 GHz will be used in the example loss budget.

Loss at Backplane Interconnect Transitions

The BGA package represents a very small piece of your overall loss budget, typically 0.5 dB or less on each end at frequencies below 2 GHz. As part of the RocketIO Design Kit provided by Mentor Graphics and Xilinx, S-parameter models are provided for detailed simulation of BGA packages. The HyperLynx Touchstone Viewer in Figure 2 shows BGA package loss as a function of frequency. At 3.0 GHz, each package would contribute 0.8 dB to total channel loss.

S-parameter models are a behavioral representation of a passive interconnect, including package, connector, or board-level interconnect components. These frequency-domain models are based on physical measurements or extracted from electromagnetic simulators, and they simulate faster than SPICE models.

Connectors in a high-speed system take another chunk out of the loss budget. As a general rule, you should look for connectors with a differential impedance of 100

Figure 3 – The HyperLynx Via Viewer shows a low-loss blind via. Also shown is the via capacitance, characteristic impedance, and signal propagation delay through the via. Via "what-if" exploration is now possible during pre-layout analysis, enabling you to perform loss and eye-quality trade-offs with different via combinations.

Ohms (+/- 10 percent) with well under 3 dB of loss at your target frequency. I would advise selecting a vendor that supplies S-parameter connector models for simulation.

At gigabit rates, poorly designed vias can contribute 0.5-1.0 dB of loss (each) against the total loss budget. The degradation effects posed by vias are influenced by the unique characteristics of each individual via, including the proximity of nearby return paths, the number and location of pads in the pad stack, pad size and shape, the layers on which the traces connect and the associated via length (through, blind, or buried), spacing between differential vias, and "stubs" (that is, any portion of the via that is not necessary for signal transmission).

Approaches for mitigating the negative effects of vias include blind and buried vias (shorter than through-hole vias), back-drilling, using smaller pads and only the pads that are needed, using large anti-pads, and carefully locating bypass capacitors and stitching vias close to vias on high-speed nets. These techniques can reduce individual via loss to less than 0.25 dB.

Software simulators vary tremendously in their ability to model the unique characteristics of each via. Successful SERDES simulation should properly account for any associated impedance discontinuities. Figure 3 shows a blind via cross section in Mentor Graphics' HyperLynx simulator.

PCB Interconnect Loss

There are three primary ways to control PCB interconnect loss:

1. Reduce resistive loss by widening traces. "Skin effect" is the result of reduced cross-sectional area in a trace, so this solution is somewhat self-explanatory.

2. Reduce dielectric loss using low-loss dielectrics. To reduce dielectric loss, you could consider more expensive materials with lower loss tangents.

3. Reduce dielectric loss and resistive loss by shortening lines. Dielectric loss is a function of the material used and the length over which a signal is transmitted. Resistive loss is tied to the line width and the line length.

Figure 1 shows the HyperLynx Loss Viewer, which is useful for tasks such as stackup planning, loss budgeting, and material selection. Using the Viewer in conjunction with the tool's Stackup Editor, you can quite easily gain an understanding of the PCB trace components of your loss budget. Depending on the trace properties, it is reasonable to see trace losses that range from 0.5 to 5.0 dB.

0.5 dB Example:
5-in. microstrip, W = 8.0 mils,
Material: Polymide/glass
Er = 4.3, Loss Tangent = 0.014
Loss Results (at 1 GHz): $Loss_{TL}$ = 0.5, $Loss_{Res}$ = 0.24, $Loss_{Dielectric}$ = 0.26 dB
Crossover Frequency = 795 MHz

5.0 dB Example:
40-in. microstrip trace, W = 5.0 mils,
Material: FR-4
Er = 4.3, Loss Tangent = 0.02
Loss Results (at 1 GHz): $Loss_{TL}$ = 5.0, $Loss_{Res}$ = 2.25, $Loss_{Dielectric}$ = 2.75 dB
Crossover Frequency = 654 MHz

Material	E(r)	Loss Tangent	Relative Cost
FR-4	4.0-4.7	0.02	1
DriClad (IBM)	4.1	0.011	1.2
GETek	3.6-4.2	0.013	1.4
BT	4.1	0.013	1.5
Polymide/Glass	4.3	0.014	2.5
Cyanate Ester	3.8	0.009	3.5
Nelco N6000 SI	3.36	0.003	3.5
Rogers RF35	3.5	0.0015	5

Table 1 – Dielectric constants and loss tangents (dissipation factors), as well as important data for material and cost trade-offs while balancing attenuation budgets.

Although it is possible to estimate rise-time degradation and loss, the only way to get a realistic prediction of actual results … is to use a software simulator with these capabilities.

Below the crossover frequency (the point at which dielectric loss begins to dominate), increasing trace width provides more loss-reducing leverage than shortening the trace or reducing the loss tangent. This is important when you are searching for ways to trim a few decibels from your total loss budget.

Table 1, from "Signal Integrity Simplified" by Eric Bogatin, provides some important trade-off data when considering material changes to balance your attenuation budget. Obviously, since you are intentionally driving up manufacturing costs when employing this loss-reduction mechanism, it is not the first place to go for loss reduction.

Compensating for Loss at the Chip Level

At high frequencies and across long backplanes, it is likely that you will still need some help at the end of the design process. That is where you can use the FPGA itself to compensate for loss outside your budget.

There are two primary ways to control loss at the chip level:

1. Transmitter pre-emphasis. Boost the initial voltage of each edge to compensate for high-frequency loss. Pre-emphasis in a Virtex-4 RocketIO implementation will compensate for as much as 17 dB of loss or as much as 10 dB in the Virtex-II Pro X family.
2. Receiver equalization. The incoming signal's high-frequency components are amplified to compensate for loss. Equalization in the Virtex-4 RocketIO implementation will compensate for as much as 12 dB of loss or as much as 16 dB in the Virtex-II Pro X family.

Conclusion

Here is an example link budget for RocketIO transceivers in Virtex-4 devices. Using the same 6 dB budget and assuming a 3 GHz operating frequency (6 Gbps), compare this budget to the anticipated channel loss, including estimates for packages and connectors, vias, and the backplane length that correlates to Figure 1. You end up with the following relationship:

Virtex-4 Loss Budget ~ 6 dB

< = > 2 x (Package + Line Card + Vias + Connector) + Backplane + Vias

~ 2 x (0.8 + 0.8 + 0.0 + 0.25) + 11.6 + 0.5 = 16 (rounded up)

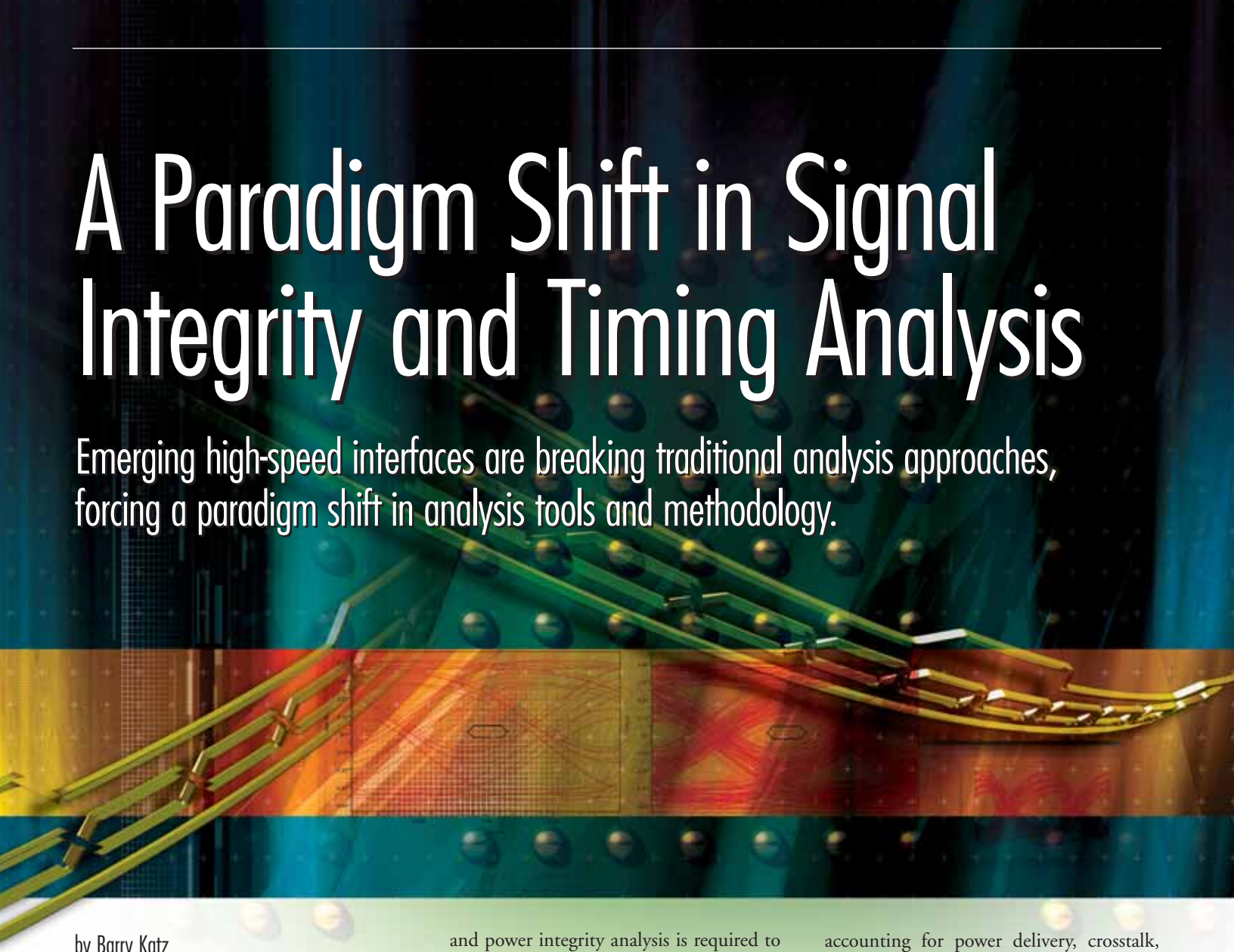
This indicates a need to use some combination of pre-emphasis and receiver equalization to compensate for the additional 10 dB. Fortunately, this is well within the capabilities of Virtex-4 RocketIO technology.

In practice, you can build inexpensive channels ranging from 12 to 18 dB while leaving additional margin for crosstalk and reflections, correcting these losses with TX and RX equalization for low bit-error-rate performance. Although it is possible to estimate rise-time degradation and loss, the only way to get a realistic prediction of actual results – including crosstalk and reflection effects – is to use a software simulator with these capabilities. Using HyperLynx, Figure 1 shows 25% driver pre-emphasis opening up the eye on such a channel.

For more information on signal integrity with RocketIO technology, visit *www.xilinx.com/signalintegrity* or *www.mentor.com/hyperlynx* for the HyperLynx RocketIO Design Kit.

A Paradigm Shift in Signal Integrity and Timing Analysis

Emerging high-speed interfaces are breaking traditional analysis approaches, forcing a paradigm shift in analysis tools and methodology.



by Barry Katz
President and CTO
SiSoft
barry.katz@sisoft.com

Simplistic rule-of-thumb approaches to interface analysis are proving to be woefully inadequate for analyzing modern high-speed interfaces like DDR2, PCI Express, and SATA-II. This situation will only worsen when emerging standards like DDR3 and 5-10 Gbps serial interfaces become commonplace.

Signal integrity analysis performed on only the shortest and longest nets in a design may not identify the worst-case inter-symbol interference, crosstalk, or pin timing scenarios caused by variations in stub length, number of vias, routing layers, AC specifications, package parasitics, and power delivery. An integrated, interface-centric approach that incorporates comprehensive signal integrity, timing, crosstalk, and power integrity analysis is required to more accurately predict system-level noise and timing margins.

Figure 1 offers the results of a simplistic versus comprehensive analysis approach to illustrate the shortcomings associated with some analysis tools, which are built on outdated rule-of-thumb methodologies and assumptions. The first waveform in Figure 1 represents a high-speed differential network using Xilinx® Virtex™-II ProX RocketIO™ IBIS models, lossless transmission lines, and ideal grounds with no crosstalk or power noise.

It is quite apparent from viewing the results that the simplistic analysis approach fails to provide the accuracy of the more comprehensive approach. The second waveform represents the progressive effect on the eye as a longer stimulus pattern is used, along with more accurate modeling of interconnect structures. The analysis also used detailed SPICE I/O models, accounting for power delivery, crosstalk, non-ideal grounds, and variations in process, voltage, and temperature.

When designers are fighting for tens of picoseconds and tens of millivolts, an approach that considers all of the factors affecting margin (see Figure 2) is essential to ensure that a design will meet its cost and performance goals.

Model Interconnect Topologies and Termination Schemes

Accurate modeling of interconnect structures and termination – including the component packaging, PCBs, connectors, and cabling – is critical for accurate simulations of high-speed networks. As edge rates have increased and interconnect structures have remained relatively long, the importance of modeling frequency-dependent loss has become much more crucial, which requires the use of two- and three-dimensional field solvers. Given the potential for wide varia-

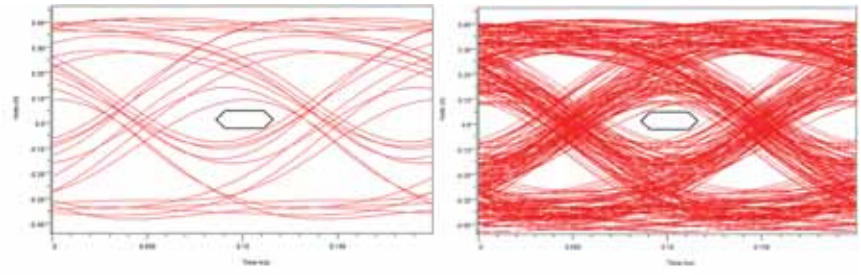


Figure 1 – Xilinx Virtex-II RocketIO transceiver simplistic versus comprehensive analysis

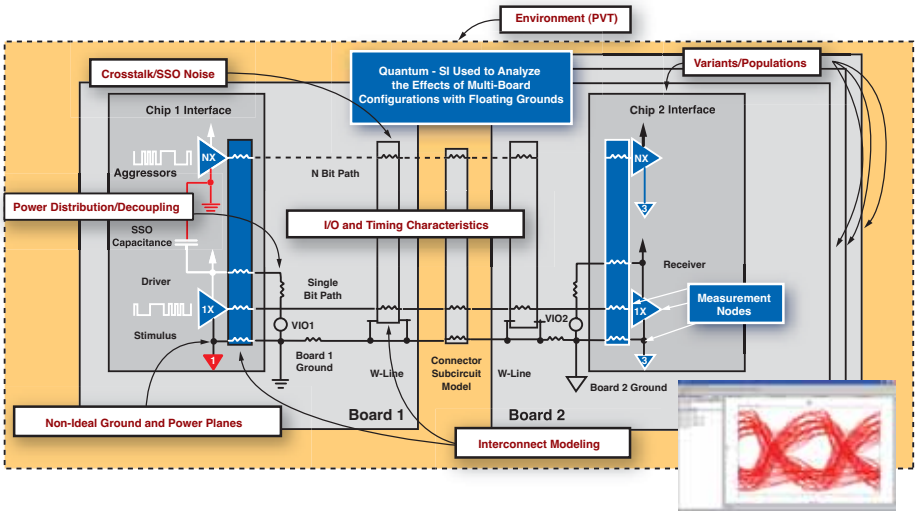


Figure 2 – Factors affecting system-level noise and timing margins

tion in the physical routing through packaging, PCBs, connectors, and cabling of many bus implementations, it is virtually impossible to identify the worst-case net without performing a comprehensive analysis on the entire interface.

Common analysis considerations that affect the analysis results include:

- Lossy versus lossless transmission lines
- Modeling vias as single- or multi-port structures
- Sensitivity to the number of vias in a net
- The use of two-dimensional distributed or three-dimensional lumped models for packages and connectors
- Modeling with S-parameters

Account for Inter-Symbol Interference

Traditional simulation approaches assume that signals are quiescent before another transition occurs. As the operating frequencies increase, the likelihood that a line has not settled to its quiescent state increases. The effect on one transition from the residual ringing on the line from one or more previous transitions results in delay variations. These delay variations, called intersymbol interference, or ISI, require complex stimulus patterns that excite the different resonances of the network to create the worst-case scenarios. For some networks, these patterns may have a handful of transitions, but for multi-gigabit serial links, it is common to use long pseudo-random bit sequence (PRBS) patterns. Because the resonant frequency of a network is a function of the electrical length, the worst-case ISI effects may or may not occur on the shortest or longest net. In addition, interconnect process variations must be accurately accounted for, as this variation will cause changes in the resonant frequency (reflections) of the network.

Multi-gigabit serial link interfaces contain embedded clocks in the serial stream and use clock recovery techniques to extract the serial data, which must meet stringent eye mask requirements. I/O buffer model accuracy that reflects pre-emphasis/de-emphasis and equalization is crucial for analyzing the effects of ISI.

Don't Forget the Effects of Crosstalk

Crosstalk is noise generated on a net from transitions on nearby interconnects in the circuit board, packages, connectors, and cables. Crosstalk can change the level of the signal on a net and therefore cause variations in the interconnect delays and reduce noise margins. Synchronous and asynchronous crosstalk are noise sources that must be fully analyzed to determine their effects on signal integrity and timing margins.

Model I/O Buffer Characteristics and Component Timing

I/O buffer electrical and timing characteristics play a key role in defining the maximum frequency of operation. A flexible methodology and automated analysis approach is required to support the wide variations in I/O technology models, including mixed IBIS and SPICE simulation. SPICE models are more accurate and very useful when simulating silicon-to-silicon. SiSoft implements this through its Core-to-Core Methodology, as shown in Figure 3. However, you should recognize that the improvement in accuracy comes at a price – a 5x to 100x simulation speed decrease.

Output buffers and input receivers are commonly characterized by numerous electrical/timing characteristics and reliability thresholds. These cells may include on-die termination, controlled impedances/slew rates, pre-emphasis, and equalization.

For high-speed parallel buses, data input timing is defined as a setup/hold time requirement with respect to a clock or strobe. Data output timing is defined by the minimum and maximum delay when driving a reference load with respect to a clock or strobe. With the advent of SSTL signaling, AC and DC levels were introduced for Vil/Vih to more accurately characterize receiver timing with respect to an input signal. Further refinements have been made through slew rate derating (required for DDR2 and DDR3), which

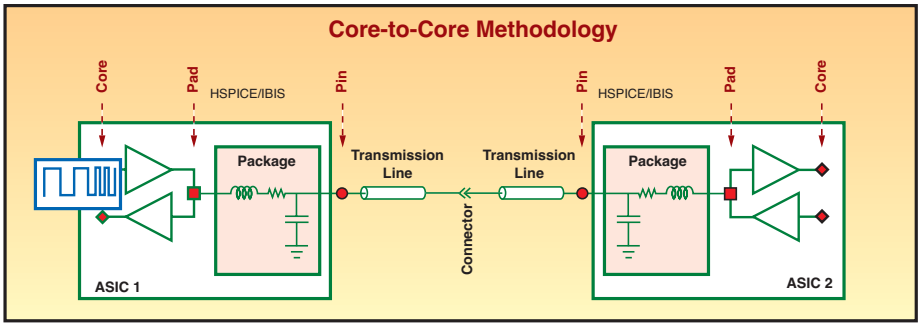


Figure 3 – SiSoft's Core-to-Core Methodology

uses tables to model the internal delay of a receiver at the core based on the slew rate at the pad. These refinements are not taken into account by simplified analysis approaches. This is why they cannot be used to accurately model the more complex behavior of many high-speed interfaces, where tens of picoseconds and tens of millivolts matter.

Don't Neglect PVT Variations

Many analysis tools and simplified methodologies neglect the effects of process, voltage, and temperature (PVT) variations, which can have disastrous results at high Mbps/Gbps signaling rates. It is especially important to consider IC process variations when modeling interconnect structures. Manufacturers typically supply data describing the AC specs and I/O buffer characteristics for fast, typical, and slow process parts, which bound the expected operating region. You should always analyze high-speed designs at the minimum/maximum operating extremes to avoid finding unpleasant surprises after the hardware is built.

Maintain Power Integrity

Maintaining the integrity of the power subsystems for both I/O and core power is critical. This requires analyzing stackups; PCB, package, and IC decoupling; routing layers and associated signal return paths. At a high level, the goal is to maintain a low impedance connection between associated voltage references across the operational frequency of interest. Simultaneous switching output (SSO) noise is commonly analyzed as part of power delivery to the I/O structures and also includes the effects of package crosstalk. SSO is often quantified in terms of a timing uncertainty penalty applied to the AC timing specs of the chip.

Accurately Determine Setup and Hold Margins

Faster interfaces require maintaining very tight timing margins. Interfaces are typically classified as either synchronous (common-clock), source-synchronous, clock recovery, or a hybrid of these types. It is important that the clock distribution is accurately simulated and used in carefully correlated ways with data nets to accurately predict timing margins and optimal clock distribution. The integration of accurate signal integrity, timing, crosstalk, and rules-driven design is the basis of a new paradigm, which we call "High-Speed Design Closure."

Required Tools and Methodology Paradigms

To overcome the shortcomings of traditional analysis methodologies and inaccuracies associated with oversimplified rules-of-thumb, today's high-speed interface designers need to adopt a more comprehensive interface-centric system-level analysis approach that addresses many (if not all) of the issues discussed in this article.

High-quality I/O buffer models, interconnect models, and accurate component AC timing/electrical specifications are fundamental to any analysis approach. The process of capturing and managing multiple interface designs; performing comprehensive simulations over process, voltage, and temperature for a large solution space of variables; and analyzing the simulation results for waveform quality, timing, crosstalk, SSO, and ISI effects is a daunting task without proper tools, which automate and integrate many manual steps and processes.

A highly automated analysis approach is also required to understand the loading effects associated with multi-board designs that include different board populations and part variants, and manage the complex set of variables within a multi-dimensional solution space. In pre-layout analysis, it is crucial to be able to mine the simulation results from different solution/space scenarios to pick an optimal solution for component placement and board routing.

Once the boards have been routed, it is equally important to verify the routed designs in the final system configuration, including different board populations and part variants to "close the loop" on signal integrity and timing. Accurate signal integrity analysis and crosstalk prediction in post-layout is essential to predicting system-level noise and timing margins.

With "High-Speed Design Closure," SiSoft is committed to providing tools for signal integrity, timing, crosstalk, and rules-driven design that meet rapidly changing signal integrity and timing requirements.

Conclusion

High-speed interface design and analysis complexity is only going to increase as edge rates and data rates get faster and voltage rails decrease. Engineering managers should recognize that setting up a high-speed interface analysis process requires an investment in simulation libraries, analysis products, and people.

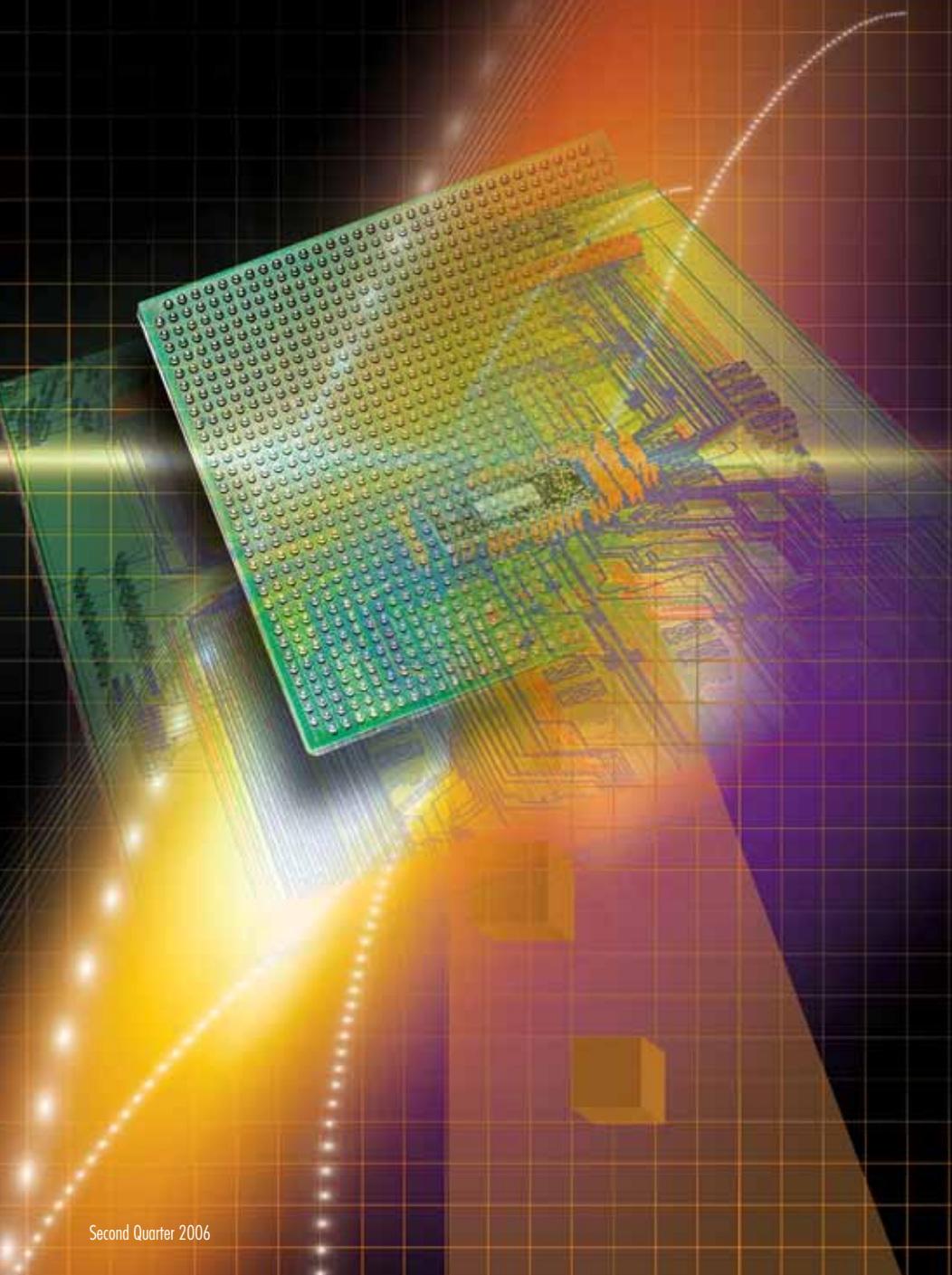
When you invest in tools, do your homework first. Check to see if prospective tools can really address some of the tough issues presented in this article and that they provide you the growth path you need for the future. Perform thorough (and possibly lengthy) comparative evaluations of potential products to see if they address your current signal integrity, timing, power delivery, and crosstalk analysis needs, but also keep an eye to the future – it will arrive sooner than you think.

To learn more about SiSoft's products and services, visit *www.sisoft.com* or e-mail *info@sisoft.com*.

Selecting a High-Speed POL Converter for FPGAs

Bellnix offers a high-quality power management solution.

by Shotaro Suzuki
President
Bellnix
s-suzuki@bellnix.co.jp



Core voltages in FPGAs have quickly dropped to lower voltages in current technologies. There is a lack of good information about power supplies that can provide not just the required level of performance, but the optimal level of performance. One design choice is point-of-load (POL) power supplies.

There is some confusion about the benefits to be gained by using POL supplies. The speed of response of these POL converters directly affects power distribution system (PDS) design choices. In this article, I'll describe a basic method for designing a PDS and review the birth of the POL converter and Bellnix's high-speed response POL converter.

We recommend a PDS (Figure 1) for resolving the low-voltage problems of FPGAs, which includes providing the required low voltages (1.2V or 1.0V) at high currents (as much as 5A to 15A, depending on the part used, the frequency of operation, and the task at hand).

As the core voltage has moved from 2.5V in .25 µm technology to 1.0V in 65 nm technology, the effects from the voltage drop by resistance (or by line inductance of the line between the DC/DC converter and load [FPGA]) must be minimized. A POL implies that regulation is provided as close as possible to the load. If a DC/DC converter can be located next to the load, many of the problems I just mentioned would be solved.

This is how the POL converter was developed. At Bellnix, we soon discovered that shortening the length between power supply and load provided additional benefits.

POL Converters: The Only Solution?

A POL converter means that power supplies are located close to the load. Technology and design improvements in the latest high-end FPGAs have improved performance to the point where precise regulation and quick response to transients in the load current are required to get the best performance from the overall system.

POL and DC/DC converters must have ultra-high-speed load transient response, high efficiency, low noise, and small size. This is necessary to track the large transient changes in current in order to optimize performance. Changes in voltage may lead to jitter, timing push-out, duty-cycle distortion, and other unwanted effects. The choice of bypass capacitors is only able to provide for so much transient current, and ultra-high-speed POL converters are able to provide a superior solution over conventional supplies and capacitors alone.

An isolated DC/DC converter is allocated near the input pin and protects from interference between other boards. This is a structure where the high-speed-response type of non-isolated POL converter is allocated near the load end of the FPGA. Figure 2 illustrates the load transient speed that is estimated on our product, the Ultra High-Speed POL Converter, BSV-m Series. The response time (recovery time) for rapid load current change from 0A to 7A is one of the highest speed responses in our industry.

A high-speed-response type POL converter also provides these benefits:

- It follows the rapid current changes of the FPGA and provides the required currents to prevent jitter and timing push-out.
- Ultra-high-speed recovery time leads to an extremely small output voltage change of the DC/DC converter. The total capacitance of the external de-coupling capacitor array can be greatly reduced.

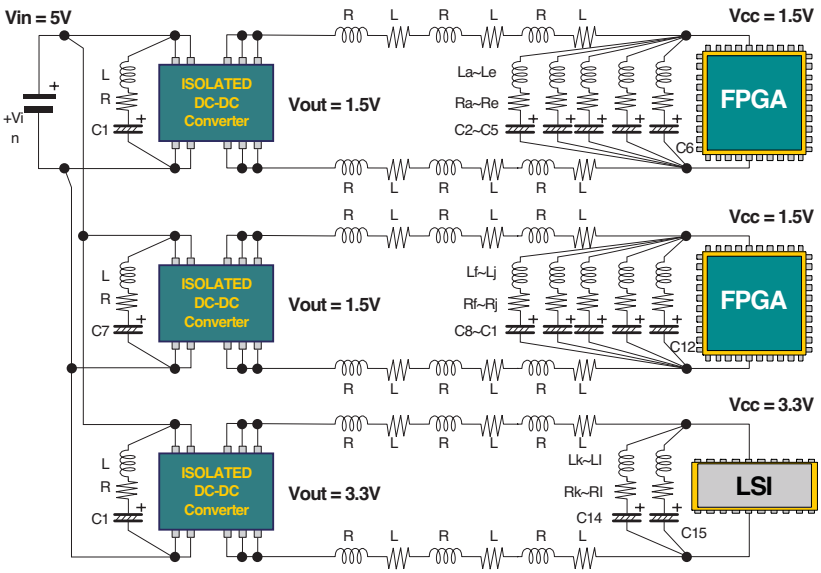


Figure 1 – Centralized power architecture

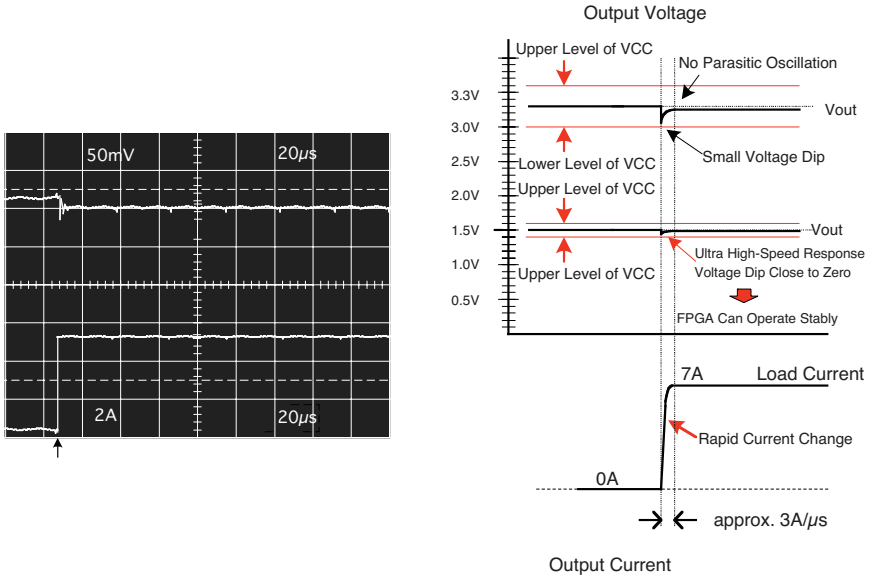


Figure 2 – Bellnix solution with distributed power architecture

- In-rush currents are largely decreased and kept under control because the total capacitance and number of external decoupling capacitors are kept small.
- The load end can be supplied optimally where required by locating minimized ultra-high-efficiency POL converters where needed.
- Extremely small packing densities.
- It comes as close as possible to an ideal power supply – basically a "battery" because of their high-speed recovery function.

High-Speed-Response Type

Figure 3 illustrates the actual measurement of the conventional DC/DC converter, which has high efficiency but a slow general-purpose response speed of about 38 µs. When load currents rapidly change from 0A to 7A, output voltage changes will occur. The recovery time is 38 µs and the voltage change is 400 mV.

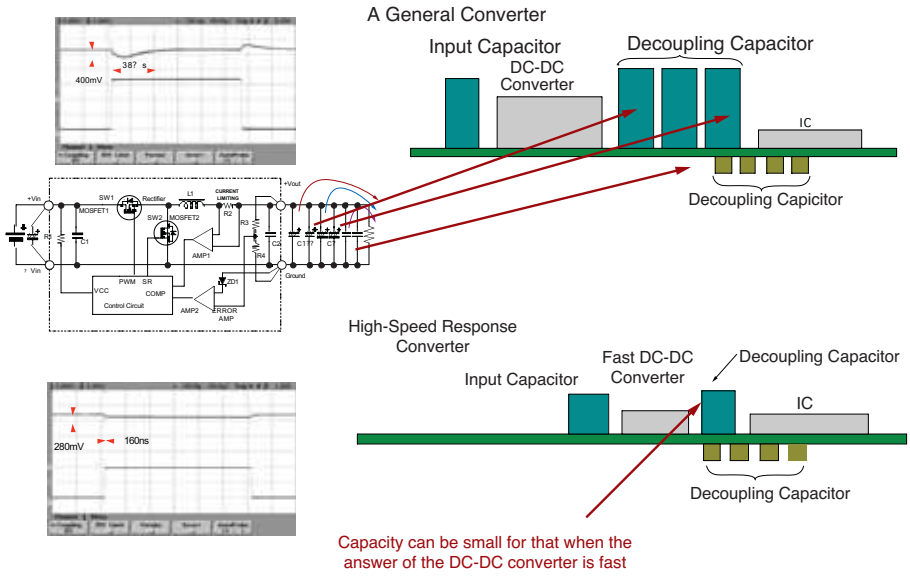


Figure 3 – Mounting image of on-board power for FPGA

This type of DC/DC converter is in use as a commercial product. However, if this DC/DC converter is used for the power supply in a high-performance FPGA, lots of external capacitors may be required to maintain voltage until the converter recovers.

If the FPGA voltage core goes lower (to around 1V) and if advanced high-speed operations are required, the capacitor array reaches its limits. Capacitors have self-inductance and self-resistance, which are not stable with temperature. Figure 3 illustrates the printed board with lots of capacitors. If many capacitors are installed, converters are forced to provide more power than is necessary while turning on.

Even if other POL converters for high-end FPGAs are unfit for high-speed response, you could stabilize output voltage by installing an array of large-value output capacitors. This is not an optimal solution, and has problems of its own.

Figure 3 also illustrates our product performance and the load transient characteristics of high-speed POL converters for high-end FPGAs. When load current changes rapidly from 0A to 6A, the time for recovering the voltage to a stable level is rapid. And the resulting voltage change is amazingly small – only 28 mV during this recovery time. Thus, the need for and number of capacitors is significantly reduced on the printed board, as in Figure 3.

The Ultra High-Speed POL Converter BSV Series offers the following benefits:

- Circuit installation is simplified.
- In-rush current to capacitors is greatly decreased; converters can be allocated near the load, and the POL converter can do what it is designed to do
- Noise performance is improved because of the small series resistance and inductance
- Released from power supply issues, cost and size reduction becomes possible
- Power becomes stable on the entire circuit

Selecting a POL Converter for High-Performance FPGAs

When selecting a POL converter, here are some guidelines you should follow:

- High efficiency is required (90% or more preferred). The POL converter can operate at a much lower temperature and does not contribute to the heat dissipation of the load (FPGA).
- High-speed response performance is required. A 160 ns recovery time is adequate for almost all high-performance FPGAs.
- Low impedance installation is recommended with surface mount device (SMD) architecture.
- It is better not to use excessive external capacitors. Decoupling capacitors are necessary to decrease circuit impedances, but you should avoid large-capacity capacitors because other problems (like in-rush current) may arise.
- A low-noise POL converter is a must.
- The FPGA may require multiple voltages. Select a series of POL converters, which are able to control rise and fall time, and power on sequences. In the case of two or three power supplies, the time sequence in each power supply should be able to turn on in the order indicated by the power sequence desired. A converter equipped with the on/off and power-good function is also recommended.

Conclusion

High-speed load-transient characteristics that provide electric currents to the FPGA core have many practical advantages.

Bellnix developed a conventional DC/DC converter for distributed power architecture (DPA) for high efficiency, small size, and adjustable voltage. But there were very few DC/DC converters that could operate with low voltage and respond to high-speed load transients.

Bellnix has also developed a high-speed POL converter that can provide 0.8V to 3.3V at up to 12A, and has achieved high-speed response in a high-density, high-performance FPGA.

One of the most important features of our POL is its small size. We have developed one of the smallest DC/DC converters for POL in the world that can be installed beside the load. It reduces the need for extra capacitors and handily succeeds as a POL supply in a small space.

For more information, visit *www.bellnix.com*, e-mail *k-suzuki@bellnix.co.jp*, or contact Bellnix USA at (408) 249-5325.

XILINX®
The Programmable Logic Company℠

Xilinx Virtex™-4 FPGAs

www.xilinx.com/devices/

Product Selection Matrix

		Virtex-4 LX (Logic)								Virtex-4 SX (Signal Processing)			Virtex-4 FX (Embedded Processing & Serial Connectivity)					
		XC4VLX15	XC4VLX25	XC4VLX40	XC4VLX60	XC4VLX80	XC4VLX100	XC4VLX160	XC4VLX200	XC4VSX25	XC4VSX35	XC4VSX55	XC4VFX12	XC4VFX20	XC4VFX40	XC4VFX60	XC4VFX100	XC4VFX140
CLB Resources	CLB Array (Row x Column)	64 x 24	96 x 28	128 x 36	128 x 52	160 x 56	192 x 64	192 x 88	192 x 116	64 x 40	96 x 40	128 x 48	64 x 24	64 x 36	96 x 52	128 x 52	160 x 68	192 x 84
	Slices	6,144	10,752	18,432	26,624	35,840	49,152	67,584	89,088	10,240	15,360	24,576	5,472	8,544	18,624	25,280	42,176	63,168
	Logic Cells	13,824	24,192	41,472	59,904	80,640	110,592	152,064	200,448	23,040	34,560	55,296	12,312	19,224	41,904	56,880	94,896	142,128
	CLB Flip Flops	12,288	21,504	36,864	53,248	71,680	98,304	135,168	178,176	20,480	30,720	49,152	10,944	17,088	37,248	50,560	84,352	126,336
Memory Resources	Max. Distributed RAM Bits	98,304	172,032	294,912	425,984	573,440	786,432	1,081,344	1,425,408	163,840	245,760	393,216	87,552	136,704	297,984	404,480	674,816	1,010,688
	Block RAM/FIFO w/ECC (18 kbits each)	48	72	96	160	200	240	288	336	128	192	320	36	68	144	232	376	552
	Total Block RAM (kbits)	864	1,296	1,728	2,880	3,600	4,320	5,184	6,048	2,304	3,456	5,760	648	1,224	2,592	4,176	6,768	9,936
Clock Resources	Digital Clock Managers (DCM)	4	8	8	8	12	12	12	12	4	8	8	4	4	8	12	12	20
	Phase-matched Clock Dividers (PMCD)	0	4	4	4	8	8	8	8	0	4	4	0	0	4	8	8	8
I/O Resources	Max Select I/O™	320	448	640	640	768	960	960	960	320	448	640	320	320	448	576	768	896
	Total I/O Banks	9	11	13	13	15	17	17	17	9	11	13	9	9	11	13	15	17
	Digitally Controlled Impedence	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	Max Differential I/O Pairs	160	224	320	320	384	480	480	480	160	224	320	160	160	224	288	384	448
	I/O Standards	LDT-25, LVDS-25, LVDSEXT-25, BLVDS-25, ULVDS-25, LVPECL-25, LVCMOS25, LVCMOS18, LVCMOS15, PCI33, LVTTL, LVCMOS33, PCI-X, PCI66, GTL, GTL+, HSTL I (1.5V,1.8V), HSTL II (1.5V,1.8V), HSTL III (1.5V,1.8V), HSTL IV (1.5V,1.8V), SSTL2I, SSTL2II, SSTL18 I, SSTL18 II																
DSP Resources	XtremeDSP™ Slices	32	48	64	64	80	96	96	96	128	192	512	32	32	48	128	160	192
Embedded Hard IP Resources	PowerPC™ Processor Blocks	—	—	—	—	—	—	—	—	—	—	—	1	1	2	2	2	2
	10/100/1000 Ethernet MAC Blocks	—	—	—	—	—	—	—	—	—	—	—	2	2	4	4	4	4
	RocketIO™ Serial Transceivers	—	—	—	—	—	—	—	—	—	—	—	0	8	12	16	20	24
Speed Grades	Commercial (slowest to fastest)	-10, -11, -12	-10, -11, -12	-10, -11, -12	-10, -11, -12	-10, -11, -12	-10, -11, -12	-10, -11, -12	-10, -11	-10, -11, -12	-10, -11, -12	-10, -11, -12	-10, -11, -12	-10, -11, -12	-10, -11, -12	-10, -11, -12	-10, -11, -12	-10, -11
	Industrial (slowest to fastest)	-10, -11	-10, -11	-10, -11	-10, -11	-10, -11	-10, -11	-10, -11	-10	-10, -11	-10, -11	-10, -11	-10, -11	-10, -11	-10, -11	-10, -11	-10, -11	-10
	Configuration Memory Bits	4,765,568	7,819,904	12,259,712	17,717,632	23,291,008	30,711,680	40,347,008	51,367,808	9,147,648	13,700,288	22,745,216	4,765,568	7,242,624	13,550,720	21,002,880	33,065,408	47,856,896
	EasyPath™ Cost Reduction Solutions[1]	—	XCE4VLX25	XCE4VLX40	XCE4VLX60	XCE4VLX80	XCE4VLX100	XCE4VLX160	XCE4VLX200	XCE4VSX25	XCE4VSX35	XCE4VSX55	—	XCE4VFX20	XCE4VFX40	XCE4VFX60	XCE4VFX100	XCE4VFX140

Package[1]	Area	MGT[2]	Pins	4VLX15	4VLX25	4VLX40	4VLX60	4VLX80	4VLX100	4VLX160	4VLX200	4VSX25	4VSX35	4VSX55	4VFX12	4VFX20	4VFX40	4VFX60	4VFX100	4VFX140
SF363	17 x 17 mm	—	240	240	240										240					
FF668	27 x 27 mm	—	448	320	448	448	448					320	448		320					
FF1148	35 x 35 mm	—	768			640	640	768	768	768				640						
FF1513	40 x 40 mm	—	960						960	960	960									
FF672	27 x 27 mm	12	352													320 (8)[3]	352 (12)[3]	352 (12)[3]		
FF1152	35 x 35 mm	20	576														448 (12)[3]	576 (16)[3]	576 (20)[3]	
FF1517	40 x 40 mm	24	768																768 (20)[3]	768 (24)[3]
FF1760	42.5 x 42.5 mm	24	896																	896 (24)[3]

Notes: 1. SFA Packages (SF): flip-chip fine-pitch BGA (0.80 mm ball spacing).
FFA Packages (FF): flip-chip fine-pitch BGA (1.00 mm ball spacing).
All Virtex-4 LX and Virtex-4 SX devices available in the same package are footprint-compatible.
2. MGT: RocketIO Multi-Gigabit Transceivers.
3. Number of available RocketIO Muti-Gigabit Transceivers.
4. EasyPath solutions provide conversion-free path for volume production.

Pb-free solutions are available. For more information about Pb-free solutions, visit www.xilinx.com/pbfree.

Important: Verify all data in this document with the device data sheets found at http://www.xilinx.com/partinfo/databook.htm

Product Selection Matrix

	CLB Resources					Memory Resources			DSP	CLK Resources		I/O Features				Speed		PROM	
	System Gates (see note 1)	CLB Array (Row x Col)	Number of Slices	Equivalent Logic Cells	CLB Flip-Flops	Max. Distributed RAM Bits	# Block RAM	Block RAM (bits)	Dedicated Multipliers	DCM Frequency (min/max)	# DCMs	Digitally Controlled Impedance	Number of Differential I/O Pairs	Maximum I/O	I/O Standards	Commercial Speed Grades (slowest to fastest)	Industrial Speed Grades (slowest to fastest)	Configuration Memory (Bits)	EasyPath
Spartan-3E Family – 1.2 Volt																			
XC3S100E	100K	16 x 22	960	2,160	1920	15K	4	72K	4	5/326	2	NO	40	108	Single-ended LVTTL, LVCMOS3.3/2.5/1.8/1.5/1.2, PCI 3.3V – 32/64-bit 33/66MHz, PCI-X 100MHz, SSTL I 1.8/2.5, HSTL I 1.8, HSTL III 1.8	-4 -5	-4	0.6M	
XC3S250E	250K	26 x 34	2448	5,508	4896	38K	12	216K	12	5/326	4	NO	68	172		-4 -5	-4	1.4M	
XC3S500E	500K	34 x 46	4656	10,476	9312	73K	20	360K	20	5/326	4	NO	92	232		-4 -5	-4	2.3M	
XC3S1200E	1200K	46 x 60	8672	19,512	17344	136K	28	504K	28	5/326	8	NO	124	304		-4 -5	-4	3.8M	
XC3S1600E	1600K	58 x 76	14752	33,192	29504	231K	36	648K	36	5/326	8	NO	156	376	Differential LVDS2.5, Bus LVDS2.5, mini-LVDS, RSDS, LVPECL	-4 -5	-4	5.9M	
Spartan-3 and Spartan-3L Families – 1.2 Volt (see note 2)																			
XC3S50		50K	16 x 12	768	1,728	1,536	12K	4	72K	4	24/280	2	YES	56	124	Single-ended LVTTL, LVCMOS3.3/2.5/1.8/1.5/1.2, PCI 3.3V – 32/64-bit 33MHz, SSTL2 Class I & II, SSTL18 Class I, HSTL Class I, III, HSTL1.8 Class I, II & III, GTL, GTL+	-4 -5	-4	.4M	
XC3S200		200K	24 x 20	1,920	4,320	3,840	30K	12	216K	12	24/280	4	YES	76	173		-4 -5	-4	1.0M	
XC3S400		400K	32 x 28	3,584	8,064	7,168	56K	16	288K	16	24/280	4	YES	116	264		-4 -5	-4	1.7M	
XC3S1000	XC3S1000L	1000K	48 x 40	7,680	17,280	15,360	120K	24	432K	24	24/280	4	YES	175	391		-4 -5	-4	3.2M	
XC3S1500	XC3S1500L	1500K	64 x 52	13,312	29,952	26,624	208K	32	576K	32	24/280	4	YES	221	487		-4 -5	-4	5.2M	✔
XC3S2000		2000K	80 x 64	20,480	46,080	40,960	320K	40	720K	40	24/280	4	YES	270	565	Differential LVDS2.5, Bus LVDS2.5, Ultra LVDS2.5, LVDS_ext2.5, RSDS, LDT2.5, LVPECL	-4 -5	-4	7.7M	✔
XC3S4000	XC3S4000L	4000K	96 x 72	27,648	62,208	55,296	432K	96	1,728K	96	24/280	4	YES	312	712		-4 -5	-4	11.3M	✔
XC3S5000		5000K	104 x 80	33,280	74,880	66,560	520K	104	1,872K	104	24/280	4	YES	344	784		-4 -5	-4	13.3M	✔

Note: 1. System Gates include 20-30% of CLBs used as RAMs.
2. Spartan-3L devices offer reduced quiescent power consumption. Package offerings may vary slightly from those offered in the Spartan-3 family. See Package Selection Matrix for details.

Package Options and UserI/O[1]

		Spartan-3E (1.2V)							Spartan-3 (1.2V)							
			XC3S100E	XC3S250E	XC3S500E	XC3S1200E	XC3S1600E		XC3S50	XC3S200	XC3S400	XC3S1000	XC3S1500	XC3S2000	XC3S4000	XC3S5000
Pins	Area[2]	I/O's	108	172	232	304	376		I/O's	124	173	264	391	487	565	712	784
PQFP Packages (PQ) – wire-bond plastic QFP (0.5 mm lead spacing)																	
208	30.6 x 30.6 mm			158	158					124	141	141					
240	34.6 x 34.6 mm																
VQFP Packages (VQ) – very thin QFP (0.5 mm lead spacing)																	
100	16.0 x 16.0 mm		66	66						63	63						
TQFP Packages (TQ) – thin QFP (0.5 mm lead spacing)																	
144	22.0 x 22.0 mm		108	108						97	97	97					
Chip Scale Packages (CP) – wire-bond chip-scale BGA (0.5 mm ball spacing)																	
132	8 x 8 mm			92	92					89							
Chip Scale Packages (CS) – wire-bond chip-scale BGA (0.8 mm ball spacing)																	
144	12 x 12 mm																
FGA Packages (FT) – wire-bond fine-pitch thin BGA (1.0 mm ball spacing)																	
256	17 x 17 mm			172	190	190					173	173	173				
FGA Packages (FG) – wire-bond fine-pitch BGA (1.0 mm ball spacing)																	
256	17 x 17 mm																
320	19 x 19 mm				232	250	250					221	221	221			
400	21 x 21 mm					304	304										
456	23 x 23 mm											264	333	333	333		
484	23 x 23 mm						376										
676	27 x 27 mm												391	487	489	489	
900	31 x 31 mm														565	633	633
1156	35 x 35 mm															712	784
BGA Packages (BG) – wire-bond standard BGA (1.27 mm ball spacing)																	
256	27 x 27 mm																

Notes: 1. Numbers in table indicate maximum number of user I/Os.
2. Area dimensions for lead-frame products are inclusive of the leads.

Pb-free solutions are available. For more information about Pb-free solutions visit *www.xilinx.com/pbfree*.

Platform Flash Features

	XCF01S	XCF02S	XCF04S	XCF08P	XCF16P	XCF32P
Density	1 Mb	2 Mb	4 Mb	8 Mb	16 Mb	32 Mb
JTAG Prog	✔	✔	✔	✔	✔	✔
Serial Config	✔	✔	✔	✔	✔	✔
SelectMap Config				✔	✔	✔
Compression				✔	✔	✔
Design Revisions				✔	✔	✔
VCC (V)	3.3	3.3	3.3	1.8	1.8	1.8
VCCO (V)	1.8 – 3.3	1.8 – 3.3	1.8 – 3.3	1.5 – 3.3	1.5 – 3.3	1.5 – 3.3
VCCJ (V)	2.5 – 3.3	2.5 – 3.3	2.5 – 3.3	2.5 – 3.3	2.5 – 3.3	2.5 – 3.3
Clock(MHz)	33	33	33	40	40	40
Packages	VO20	VO20	VO20	FS48 VO48	FS48 VO48	FS48 VO48
Pb-Free Pkg	VOG20	VOG20	VOG20	FSG48 VOG48	FSG48 VOG48	FSG48 VOG48
Availability	Now	Now	Now	Now	Now	Now

Important: Verify all data in this document with the device data sheets found at http://www.xilinx.com/partinfo/databook.htm

For the latest information and product specifications on all Xilinx products, please visit the following links:

FPGA and CPLD Devices
www.xilinx.com/devices/

Packaging
www.xilinx.com/packaging/

Development Reference Boards
www.xilinx.com/board_search/

Platform Flash
www.xilinx.com/products/silicon_solutions/proms/pfp/

Configuration and Storage Systems
www.xilinx.com/configsolns/

Software
www.xilinx.com/ise/

IP Reference
www.xilinx.com/ipcenter/

Global Services
www.xilinx.com/support/gsd/

Product Selection Matrix – CoolRunner™ Series

		System Gates	Macrocells	Product Terms per Macrocell	Input Voltage Compatible	Output Voltage Compatible	I/O Features: Maximum I/O	I/O Features: I/O Banking	Speed: Min. Pin-to-pin Logic Delay (ns)	Speed: Commercial Speed Grades (fastest to slowest)	Speed: Industrial Speed Grades (fastest to slowest)	Speed: IQ Speed Grade	Clocking: Global Clocks	Clocking: Product Term Clocks per Function Block
	CoolRunner-II Family – 1.8 Volt													
CoolRunner-II	XC2C32A	750	32	40	1.5/1.8/2.5/3.3	1.5/1.8/2.5/3.3	33	2	3.8	-4 -6	-6	-6	3	17
	XC2C64A	1,500	64	40	1.5/1.8/2.5/3.3	1.5/1.8/2.5/3.3	64	2	4.6	-5 -7	-7	-7	3	17
	XC2C128	3,000	128	40	1.5/1.8/2.5/3.3	1.5/1.8/2.5/3.3	100	2	5.7	-6 -7	-7	-7	3	17
	XC2C256	6,000	256	40	1.5/1.8/2.5/3.3	1.5/1.8/2.5/3.3	184	2	5.7	-6 -7	-7	-7	3	17
	XC2C384	9,000	384	40	1.5/1.8/2.5/3.3	1.5/1.8/2.5/3.3	240	4	7.1	-7 -10	-10	-10	3	17
	XC2C512	12,000	512	40	1.5/1.8/2.5/3.3	1.5/1.8/2.5/3.3	270	4	7.1	-7 -10	-10	-10	3	17
	CoolRunner XPLA3 Family – 3.3 Volt													
CoolRunner XPLA3	XCR3032XL	750	32	48	3.3/5	3.3	36		5	-5 -7 -10	-7 -10	-10	4	16
	XCR3064XL	1,500	64	48	3.3/5	3.3	68		6	-6 -7 -10	-7 -10	-10	4	16
	XCR3128XL	3,000	128	48	3.3/5	3.3	108		6	-6 -7 -10	-7 -10	-10	4	16
	XCR3256XL	6,000	256	48	3.3/5	3.3	164		7.5	-7 -10 -12	-10 -12	-12	4	16
	XCR3384XL	9,000	384	48	3.3/5	3.3	220		7.5	-7 -10 -12	-10 -12	-12	4	16
	XCR3512XL	12,000	512	48	3.3/5	3.3	260		7.5	-7 -10 -12	-10 -12	-12	4	16

Package Options and User I/O

Pins	Area[1]	CoolRunner-II: XC2C32A	XC2C64A	XC2C128	XC2C256	XC2C384	XC2C512	CoolRunner XPLA3: XCR3032XL	XCR3064XL	XCR3128XL	XCR3256XL	XCR3384XL	XCR3512XL
QFN Packages (QFG) – quad flat no-lead (0.5 mm lead spacing)													
32	5 x 5 mm	21											
48	7 x 7 mm		37										
PLCC Packages (PC) – wire-bond plastic chip carrier (1.27 mm lead spacing)													
44	17.5 x 17.5 mm	33	33					36	36				
PQFP Packages (PQ) – wire-bond plastic QFP (0.5 mm lead spacing)													
208	30.6 x 30.6 mm				173	173	173				164	172	180
VQFP Packages (VQ) – very thin QFP (0.5 mm lead spacing)													
44	12.0 x 12.0 mm	33	33					36	36				
100	16.0 x 16.0 mm		64	80	80				68	84			
TQFP Packages (TQ) – thin QFP (0.5 mm lead spacing)													
144	22.0 x 22.0 mm			100	118	118				108	120	118*	
Chip Scale Packages (CP) – wire-bond chip-scale BGA (0.5 mm ball spacing)													
56	6 x 6 mm	33	45						48				
132	8 x 8 mm			100	106								
Chip Scale Packages (CS) – wire-bond chip-scale BGA (0.8 mm ball spacing)													
48	7 x 7 mm							36	40				
144	12 x 12 mm									108			
280	16 x 16 mm										164		
FGA Packages (FT) – wire-bond fine-pitch thin BGA (1.0 mm ball spacing)													
256	17 x 17 mm				184	212	212				164	212	212
FBGA Packages (FG) – wire-bond fine-line BGA (1.0 mm ball spacing)													
324	23 x 23 mm					240	270					220	260

* JTAG pins and port enable are not pin compatible in this package for this member of the family.

Note 1: Area dimensions for lead-frame products are inclusive of the leads.

Product Selection Matrix – 9500 Series

	System Gates	Macrocells	Product Terms per Macrocell	Input Voltage Compatible	Output Voltage Compatible	I/O Features: Maximum I/O	I/O Features: I/O Banking	Speed: Min. Pin-to-pin Logic Delay (ns)	Speed: Commercial Speed Grades (fastest to slowest)	Speed: Industrial Speed Grades (fastest to slowest)	Speed: IQ Speed Grade	Clocking: Global Clocks	Clocking: Product Term Clocks per Function Block
XC9500XV Family – 2.5 Volt													
XC9536XV	800	36	90	2.5/3.3	1.8/2.5/3.3	36	1	5	-5 -7	-7	NA	3	18
XC9572XV	1,600	72	90	2.5/3.3	1.8/2.5/3.3	72	1	5	-5 -7	-7	NA	3	18
XC95144XV	3,200	144	90	2.5/3.3	1.8/2.5/3.3	117	2	5	-5 -7	-7	NA	3	18
XC95288XV	6,400	288	90	2.5/3.3	1.8/2.5/3.3	192	4	6	-6 -7 -10	-7 -10	NA	3	18
XC9500XL Family – 3.3 Volt													
XC9536XL	800	36	90	2.5/3.3/5	2.5/3.3	36		5	-5 -7 -10	-7 -10	-10	3	18
XC9572XL	1,600	72	90	2.5/3.3/5	2.5/3.3	72		5	-5 -7 -10	-7 -10	-10	3	18
XC95144XL	3,200	144	90	2.5/3.3/5	2.5/3.3	117		5	-5 -7 -10	-7 -10	NA	3	18
XC95288XL	6,400	288	90	2.5/3.3/5	2.5/3.3	192		6	-6 -7 -10	-7 -10	NA	3	18
XC9500 Family – 5 Volt													
XC9536	800	36	90	5	5	36		10	-5 -6 -10 -15	-7 -10 -15	-15	3	18
XC9572	1,600	72	90	5	5	72		10	-7 -10 -15	-10 -15	-15	3	18
XC95108	2,400	108	90	5	5	108		10	-7 -10 -15 -20	-7 -10 -15 -20	NA	3	18
XC95144	3,200	144	90	5	5	133		10	-7 -10 -15	-10 -15	NA	3	18
XC95216	4,800	216	90	5	5	166		10	-10 -15 -20	-10 -15 -20	NA	3	18
XC95288	6,400	288	90	5	5	192		10	-10 -15 -20	-15 -20	NA	3	18

Pb-free solutions are available. For more information about Pb-free solutions visit *www.xilinx.com/pbfree*

Package Options and User I/O

		XC9500XV				XC9500XL				XC9500					
Pins	Area[1]	XC9536XV	XC9572XV	XC95144XV	XC95288XV	XC9536XL	XC9572XL	XC95144XL	XC95288XL	XC9536	XC9572	XC95108	XC95144	XC95216	XC95288
PLCC Packages (PC) – wire-bond plastic chip carrier (1.27 mm lead spacing)															
44	17.5 x 17.5 mm	34	34			34	34			34	34				
84	30.2 x 30.2 mm										69	69			
PQFP Packages (PQ) – wire-bond plastic QFP (0.5 mm lead spacing)															
100	23.3 x 17.2 mm										72	81	81		
160	31.2 x 31.2 mm											108	133	133	
208	30.6 x 30.6 mm				168				168					166	168
VQFP Packages (VQ) – very thin TQFP (0.5 mm lead spacing)															
44	12.0 x 12.0 mm	34	34			34	34			34					
64	12.0 x 12.0 mm					36	52								
TQFP Packages (TQ) – thin QFP (0.5 mm lead spacing)															
100	16.0 x 16.0 mm		72	81			72	81			72	81	81		
144	22.0 x 22.0 mm			117	117			117	117						
Chip Scale Packages (CS) – wire-bond chip-scale BGA (0.8 mm ball spacing)															
48	7 x 7 mm	36	38			36	38			34					
144	12 x 12 mm			117				117							
280	16 x 16 mm				192				192						
BGA Packages (BG) – wire-bond standard BGA (1.27 mm ball spacing)															
256	27 x 27 mm								192						
352	35.0 x 35.0 mm													166	192
FBGA Packages (FG) – wire-bond Fine-line BGA (1.0 mm ball spacing)															
256	17 x 17 mm				192				192						

Note 1: Area dimensions for lead-frame products are inclusive of the leads.

XILINX®
The Programmable Logic Company℠

Xilinx Software – ISE 8.1i

www.xilinx.com/ise

ISE™ 8.1i – Powered by ISE Fmax Technology

	Feature	ISE WebPACK™	ISE Foundation™
Platforms		Microsoft Windows 2000 / XP Red Hat Enterprise Linux 3 (32 bit)	Microsoft Windows 2000 / XP Sun Solaris 2.8 or 2.9 Red Hat Enterprise Linux 3 (32 & 64 bit)
Devices	Virtex™ Series	**Virtex:** XCV50 - XCV600 **Virtex-E:** XCV50E- XCV600E **Virtex-II:** XC2V40 - XC2V500 **Virtex-II Pro:** XC2VP2 - XC2VP7 **Virtex-4:** **LX:** XC4VLX15, XC4VLX25 **SX:** XC4VSX25 **FX:** XC4VFX12 **Virtex Q:** XQV100- XQV600 **Virtex QR:** XQVR300, XQVR600 **Virtex-E Q:** XQV600E	ALL
	Spartan™ Series	**Spartan-II/IIE:** All **Spartan-3:** XC3S50 - XC3S1500 **Spartan-3E:** All **Spartan-3L:** XC3S1000L, XC3S1500L **XA (Xilinx Automotive) Spartan-3:** All	**Spartan-II/IIE:** All **Spartan-3:** All **Spartan-3E:** All **Spartan-3L:** All **XA (Xilinx Automotive) Spartan-3:** All
	CoolRunner™ XPLA3 CoolRunner-II™ CoolRunner-IIA	All	
	XC9500™ Series	All	
Design Entry	Schematic Editor	Yes	
	HDL Editor	Yes	
	State Diagram Editor	Microsoft Windows only	
	Xilinx CORE Generator System™	Yes	
	RTL & Technology Viewers	Yes	
	PACE (Pinout & Area Constraint Editor)	Yes	
	Architecture Wizards	Yes	
	3rd Party RTL Checker Support	Yes	
	Xilinx System Generator for DSP	Sold as an Option	
Embedded System Design	Embedded Design Kit (EDK)	Sold as an Option	
Synthesis	XST - Xilinx Synthesis Technology	Yes	
	Mentor Graphics LeonardoSpectrum	Integrated Interface (EDIF Interface on Linux)	
	Mentor Graphics Precision RTL	Integrated Interface	
	Mentor Graphics Precision Physical	EDIF Interface	
	Synopsys DC-FPGA Compiler	EDIF Interface	
	Synplify/Pro/Premier	Integrated Interface	
	Synplicity Amplify Physical Synthesis	EDIF Interface	
	ABEL	CPLD (Microsoft Windows only)	

	Feature	ISE WebPACK™	ISE Foundation™
Implementation	FloorPlanner		Yes
	PlanAhead™		Sold as an Option
	Timing Driven Place & Route		Yes
	Incremental Design		Yes
	Timing Improvement Wizard		Yes
	Xplorer		Yes
Programming	IMPACT /System ACE™/ CableServer		Yes
Board Level Integration	IBIS / STAMP / HSPICE* Models		Yes
	ELDO Models* (MGT only)		Yes
Verification	ChipScope PRO™		Sold as an Option
	Graphical Testbench Editor		Microsoft Windows only
	ISE Simulator Lite		Yes
	ISE Simulator		ISE Simulator is available as an Optional Design Tool for ISE Foundation only
	ModelSim® XE III Starter		Yes
	ModelSim XE III		Sold as an option
	Static Timing Analyzer		Yes
	FPGA Editor with Probe		Yes
	ChipViewer		Yes
	XPower (Power Analysis)		Yes
	3rd Party Equivalency Checking Support		Yes
	SMARTModels for PowerPC™ and RocketIO™		Yes
	3rd Party Simulator Support		Yes

*HSPICE and ELDO Models are available at the Xilinx Design Tools Center at ***www.xilinx.com/ise***

For more information, visit ***www.xilinx.com/ise***

What's New

To complement our flagship publication *Xcell Journal*, we've recently launched three new technology magazines:

- ***Embedded Magazine***, focusing on the use of embedded processors in Xilinx® programmable logic devices.
- ***DSP Magazine***, focusing on the high-performance capabilities of our FPGA-based reconfigurable DSPs.
- ***I/O Magazine***, focusing on the wide range of serial and parallel connectivity options available in Xilinx devices.

In addition to these new magazines, we've created a family of Solution Guides, designed to provide useful information on a wide range of hot topics such as *Broadcast Engineering*, *Power Management*, and *Signal Integrity*. Others are planned throughout the year.

Xcell PUBLICATIONS

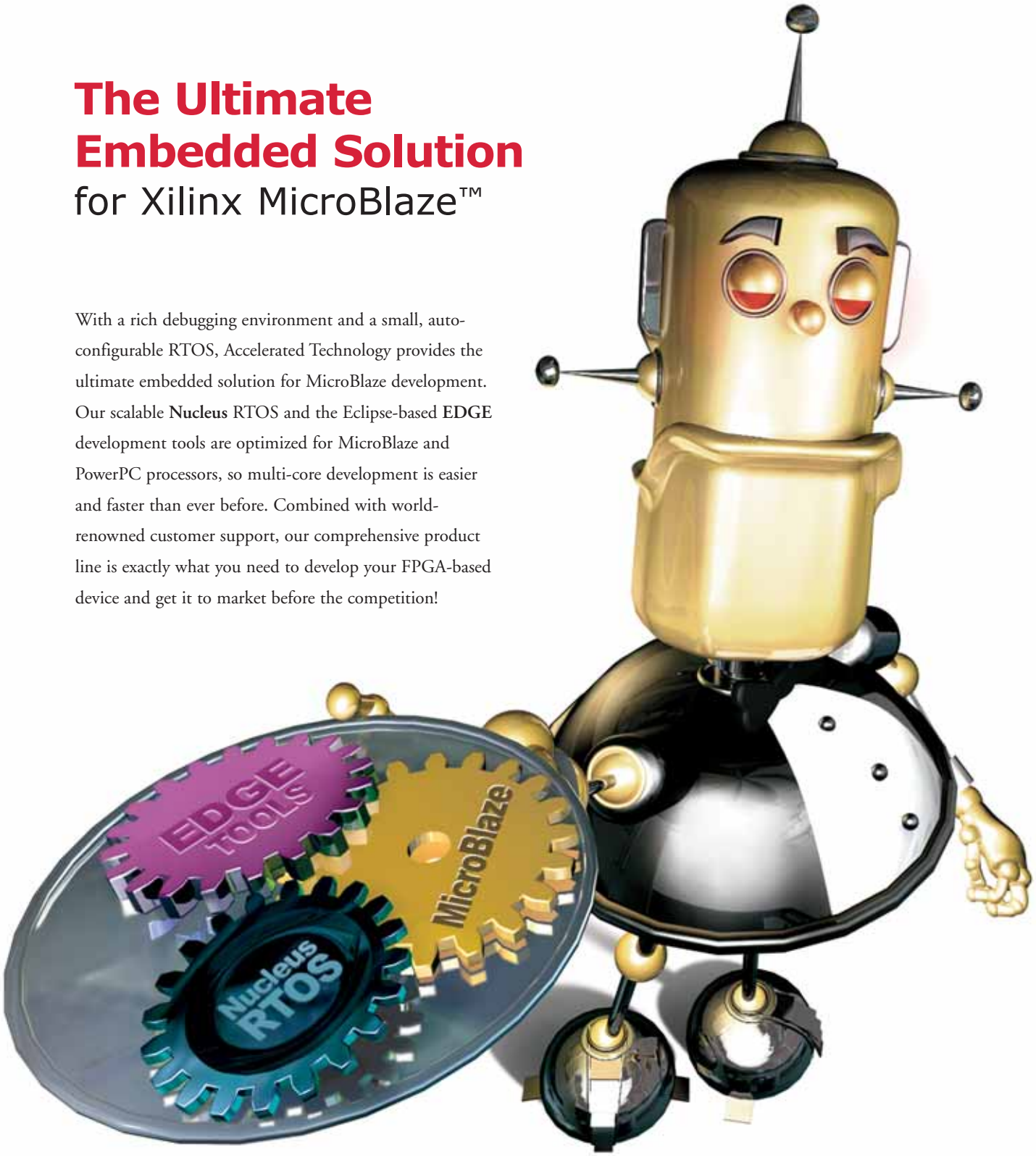
www.xilinx.com/xcell/

XILINX®

The Ultimate Embedded Solution

for Xilinx MicroBlaze™

With a rich debugging environment and a small, auto-configurable RTOS, Accelerated Technology provides the ultimate embedded solution for MicroBlaze development. Our scalable **Nucleus** RTOS and the Eclipse-based **EDGE** development tools are optimized for MicroBlaze and PowerPC processors, so multi-core development is easier and faster than ever before. Combined with world-renowned customer support, our comprehensive product line is exactly what you need to develop your FPGA-based device and get it to market before the competition!

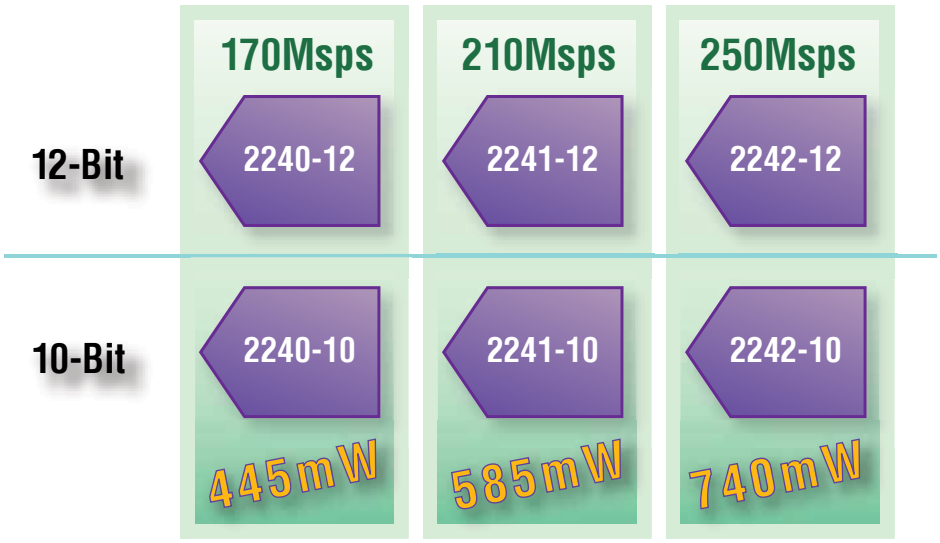


For more information on the Nucleus complete solution, go to
www.acceleratedtechnology.com/xilinx

Accelerated Technology, A Mentor Graphics Division
info@AcceleratedTechnology.com • www.AcceleratedTechnology.com

Accelerated Technology®
A Mentor Graphics Division

12-Bit, 250Msps ADC



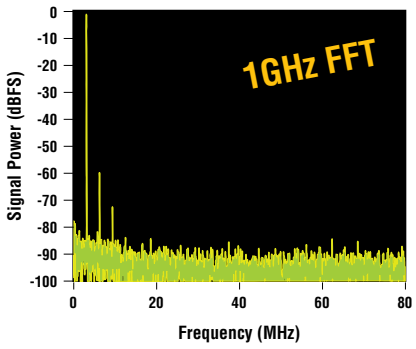
Low Power ADC Family Features 1.2GHz Full Power Bandwidth

The exceptional IF undersampling capability of the LTC®2242 family enables lower cost software-defined radios and cellular infrastructure equipment by moving the ADC closer to the antenna. Achieving SNR of 64.2dB at 500MHz input frequency, the LTC2242-12 requires just 740mW. Cable headend receivers also benefit from the low power of this pin-compatible family.

Features

- Low Power: 740mW/585mW/445mW
- Wide Input Bandwidth of 1.2GHz
- Single 2.5V Supply
- Flexible $1V_{P-P}$ or $2V_{P-P}$ Input
- LVDS/CMOS and Demultiplexed CMOS Outputs
- Optional Clock Duty Cycle Stabilizer

Undersampling a 1GHz Signal at 250Msps



Info & Free Samples

Nu Horizons Electronics Corp.

Tel: 1-888-747-NUHO

www.nuhorizons.com/linear

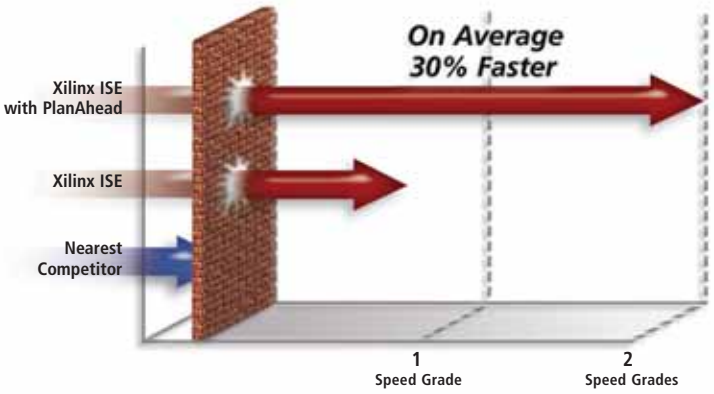
LEAD-FREE RoHS COMPLIANT

NU HORIZONS ELECTRONICS CORP.

LINEAR TECHNOLOGY

READY! SET! GO WITH PLANAHEAD PERFORMANCE!

LX SX FX



Based on benchmark data from a suite of 15 real-world customer designs targeting Xilinx and competing FPGA Solutions.

Two speed grades faster with PlanAhead software and Virtex-4

With our unique PlanAhead software tool, and our industry-leading Virtex-4 FPGAs, designers can now achieve a new level of performance. For complex, high-utilization, multi-clock designs, no other competing FPGA comes close to the Virtex-4 PlanAhead advantage:

- **30% better logic performance on average = 2 speed grade advantage**
- **Over 50% better logic performance for complex multi-clock designs**

MEET YOUR TIMING BUDGETS . . . BEAT YOUR COMPETITION TO MARKET

Meeting timing budgets is the most critical issue facing FPGA designers*. Inferior tools can hit a performance barrier, impacting your timing goals, while costing you project delays and expensive higher speed grades. To maximize the Virtex-4 performance advantage, the new PlanAhead software tool allows you to quickly analyze, floorplan, and improve placement and timing of even the most complex designs. Now, with ISE and PlanAhead you can meet your timing budgets *and* reduce design iterations, all within an easy-to-use design environment.

Download a free eval today at ***www.xilinx.com/planahead***, view the TechOnline web seminar, and prevent your next FPGA design from stalling.

* CMP: June 2005 FPGA EDA Survey

FPGA and Structured ASIC JOURNAL READER'S CHOICE AWARDS

Highest customer satisfaction:
- Virtex & Spartan FPGAs
- ISE design tools
- Support

XILINX®
The Programmable Logic Company℠

www.xilinx.com/planahead

www.techonline.com

View The TechOnLine Seminar Today

BREAKTHROUGH PERFORMANCE AT THE LOWEST COST

PN 0010923