

Xcell journal

SOLUTIONS FOR A PROGRAMMABLE WORLD

Xilinx 7 Series: FPGAs Make Play for Logic IC Dominance



INSIDE

Xilinx Rad-Hard FPGA Reaches for the Stars

Biometrics App Rides Reconfigurable Hardware

How to Maintain Repeatable Results in Xilinx FPGA Designs

Timing Constraints Tutorial

ISE Design Suite 12.2 Tips 4th-Generation Partial Reconfiguration

Design Platforms Accelerate Application Success

Development kits help ramp up new Spartan®-6 or Virtex®-6 FPGA designs

Avnet Electronics Marketing introduces three new development kits based on the Xilinx Targeted Design Platform (TDP) methodology. Designers now have access to the silicon, software tools and reference designs needed to quickly ramp up new designs. This approach accelerates time-to-market and allows you to focus on creating truly differentiated products.

Critical to the TDP methodology is the FPGA Mezzanine Card (FMC) from the VITA standards body. Avnet has collaborated with several industry-leading semiconductor manufacturers to create a host of FMC modules that add functionality and interfaces to the new baseboards, allowing for easy customization to meet design-specific requirements.

Learn more about the new Spartan-6 and Virtex-6 FPGA baseboards and FMC modules designed by Avnet at www.em.avnet.com/drc



DESIGNED BY AVNET

New baseboards for Spartan®-6 and Virtex®-6 FPGAs

- » *Spartan-6 LX16 Evaluation Kit*
- » *Spartan-6 LX150T Development Kit*
- » *Virtex-6 LX130T Development Kit*

New FMC Modules for Baseboards

- » *Dual Image Sensor FMC*
- » *DVI I/O FMC*
- » *Industrial Ethernet FMC*

More are soon to be released!



Xcell journal

PUBLISHER	Mike Santarini mike.santarini@xilinx.com 408-626-5981
EDITOR	Jacqueline Damian
ART DIRECTOR	Scott Blair
DESIGN/PRODUCTION	Teie, Gelwicks & Associates 1-800-493-5551
ADVERTISING SALES	Dan Teie 1-800-493-5551 xcelladsales@aol.com
INTERNATIONAL	Melissa Zhang, Asia Pacific melissa.zhang@xilinx.com Christelle Moraga, Europe/ Middle East/Africa christelle.moraga@xilinx.com Miyuki Takegoshi, Japan miyuki.takegoshi@xilinx.com
REPRINT ORDERS	1-800-493-5551



www.xilinx.com/xcell/

Xilinx, Inc.
2100 Logic Drive
San Jose, CA 95124-3400
Phone: 408-559-7778
FAX: 408-579-4780
www.xilinx.com/xcell/

© 2010 Xilinx, Inc. All rights reserved. XILINX, the Xilinx Logo, and other designated brands included herein are trademarks of Xilinx, Inc. All other trademarks are the property of their respective owners.

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

Teens to Technologists: Thanks for a Great Childhood

I've seen many things over my years covering the electronics industry, but at the Design Automation Conference this past June, I witnessed a first: a group of EEs and EDA folks, eyes abrim with tears of pride. What could cause such a reaction in what would otherwise seem a prosaic event? No, it wasn't a leak from one of the foundry demos, nor an announcement that all EDA tools will henceforth be government subsidized. In fact, the moment came in the closing seconds of an event called "High School Panel: You Don't Know Jack," when four very bright teen panelists looked over the crowd and then did the unexpected: thanked them. "Thank all of you for the chips and technologies you create—thank you for making my childhood so great," said one of them, without a hint of irony. The gesture prompted engineers from Broadcom, Qualcomm, Nvidia, Cadence and Synopsys alike...and yours truly...to all tear up.

"High School Panel: You Don't Know Jack" is becoming a regular highlight of the Design Automation Conference's Pavilion Panel series. As in years past, this year's panel featured Jasper Design Automation CEO Kathryn Kranen interviewing four teenagers (two girls and two boys) about their technology usage, what products are in, what products are out and what features they would like to see in future gadgets. The panel is meant to give attendees a glimpse into the technology usage of this finicky yet vitally important set of consumers and purchasing influencers. This year's foursome was exceptionally impressive and surprisingly gracious. Over years of exposure to technology and social media, these kids have become master multitaskers while still maintaining stellar GPAs (three are off to prominent colleges, while the fourth has one year of high school left).

If you are the parent of a teen, you probably won't find it too shocking that all four panelists described how, from the moment school lets out, they immediately connect to the Internet, mostly via laptops. "I have to stay connected," said one boy. A fellow panelist boots her laptop and downloads the photos she took that day from her cell phone or camera. She opens the photo files in Adobe Photoshop to airbrush any skin blemishes and then downloads those modified pictures onto her Facebook page. All four panelists said they have tens of photo albums on Facebook and have friends with hundreds. Increasingly, they are adding video to their Facebook profiles or launching their own YouTube channels. Facebook is the hub of their social lives because it allows them, as one panelist summarized, "to see what my friends are doing and what outings I was not invited to attend."

The kids gave Facebook, Twitter, YouTube and Hulu big thumbs-up, while giving thumbs-down to the once popular MySpace, which panelists said has turned into a site to merely hear band demos. Panelists liked the iPhone, even though none of them own one because of the relatively high price of the phone and the access plan. But they did not like the iPad—"it's just a big iPod Touch that I can't fit into my pocket," said one panelist. Panelists also said they prefer laptops over desktops but note that desktops are more reliable and are upgradable, which is good for power gaming. They had mixed feelings about TV, indicating they almost never watch TV on the flat-screen anymore but instead catch their favorite shows at a time of their choosing on the Web, typically via Hulu or YouTube.

The technology improvements these young people would most like to see are largely in line with the top IC and system design challenges of the day. Longer battery life topped the list. A close second was devices and applications that better facilitate multitasking. "I have six different IM/chats that I use regularly and it's hard to use all those at once," said one panelist. "I'd like to have them all in one place."

What all these data points mean, I'll leave you to interpret. But certainly one thing is clear. The technology you create is having a remarkable effect on our youth and, seemingly, the future they will march into. And if these DAC panelists are any indication, it's a future that we can all be proud of.

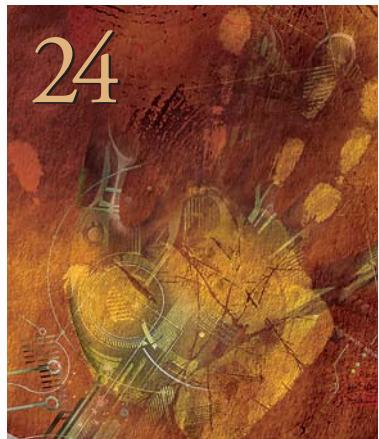
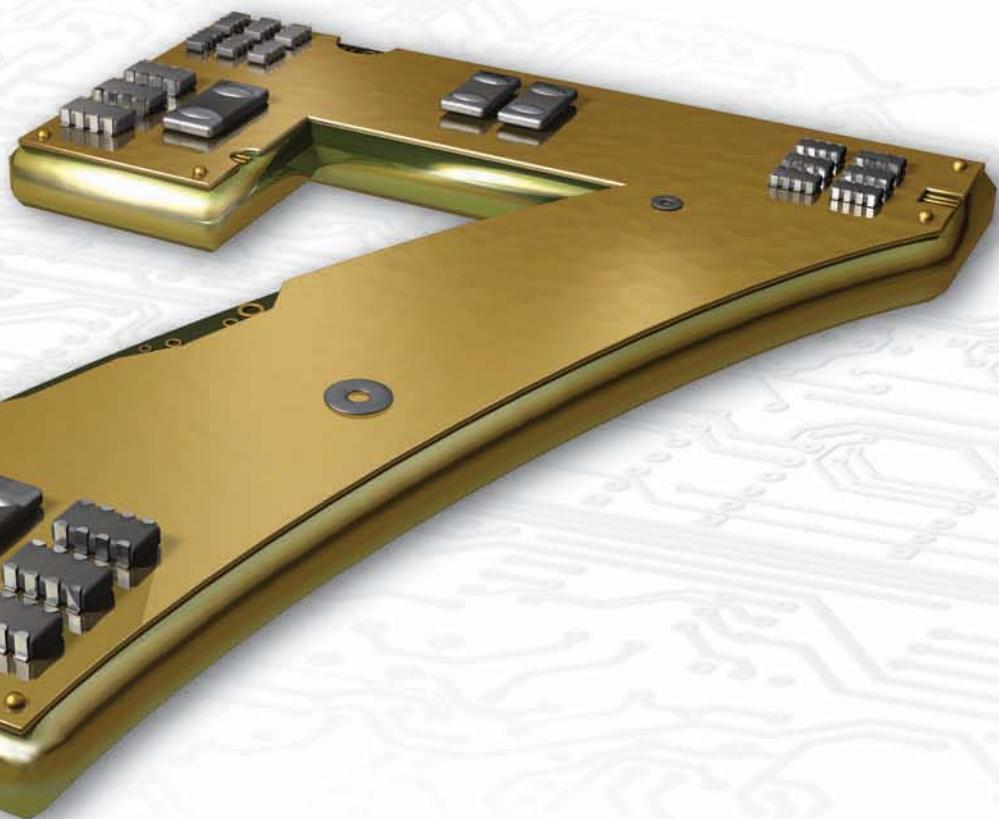


Mike Santarini
Publisher

VIEWPOINTS

Letter from the PublisherTeen Panelists at DAC Say
Thanks for the Memories...3**Cover Story**Xilinx Redefines State of the Art
with New 7 Series FPGAs

6

XCELLENCE BY DESIGN
APPLICATION FEATURES**Xcellence in A&D**New Xilinx Rad-Hard FPGA
Reaches for the Stars...12**Xcellence in Automotive**Multiple MicroBlazes Ease
Integration in Real-Time System...18**Xcellence in ISM**Biometrics May Be Killer App
for Dynamic Partial Reconfiguration...24**Xcellence in Communications**A Better Crypto Engine,
the Programmable Way...32**Xcellence in Wireless Communications**LTE Simulator Rides
Xilinx Virtex-5 FPGAs...36

THE XILINX XPERIENCE FEATURES

Xperts Corner Maintaining Repeatable Results in Xilinx FPGA Designs...**40**

Xplanation: FPGA 101
A Tutorial on Timing Constraints...**46**

Xplanation: FPGA 101
Simplifying Metastability with IDDR...**52**

XTRA READING

Xtra, Xtra The latest Xilinx tool updates and patches, as of July 2010...**55**

Are You Xperienced? Training at the cusp of the programmable imperative...**58**

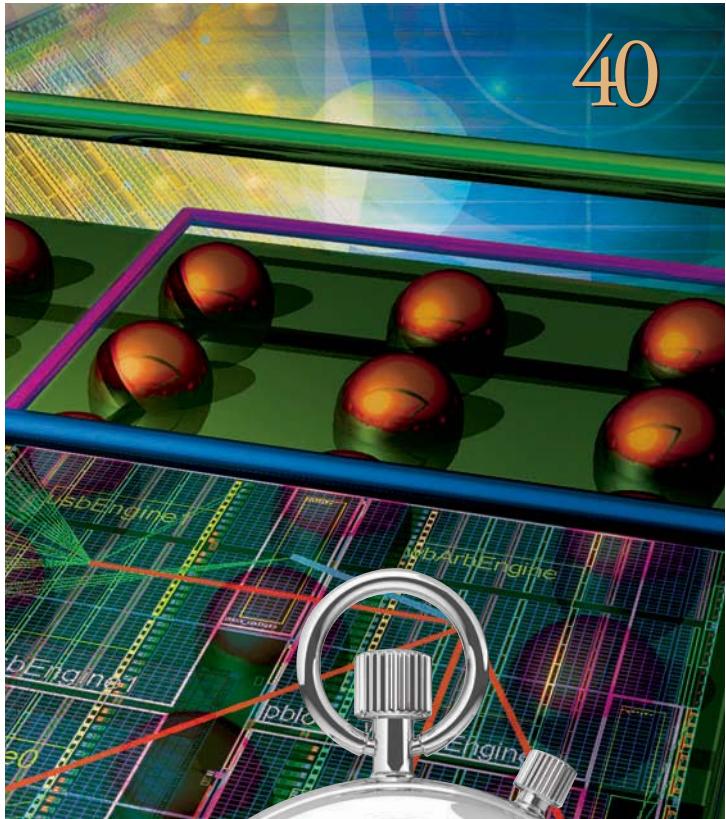
Xamples... A mix of new and popular application notes...**60**

Tools of Xcellence A bit of news about our partners and their latest offerings...**62**

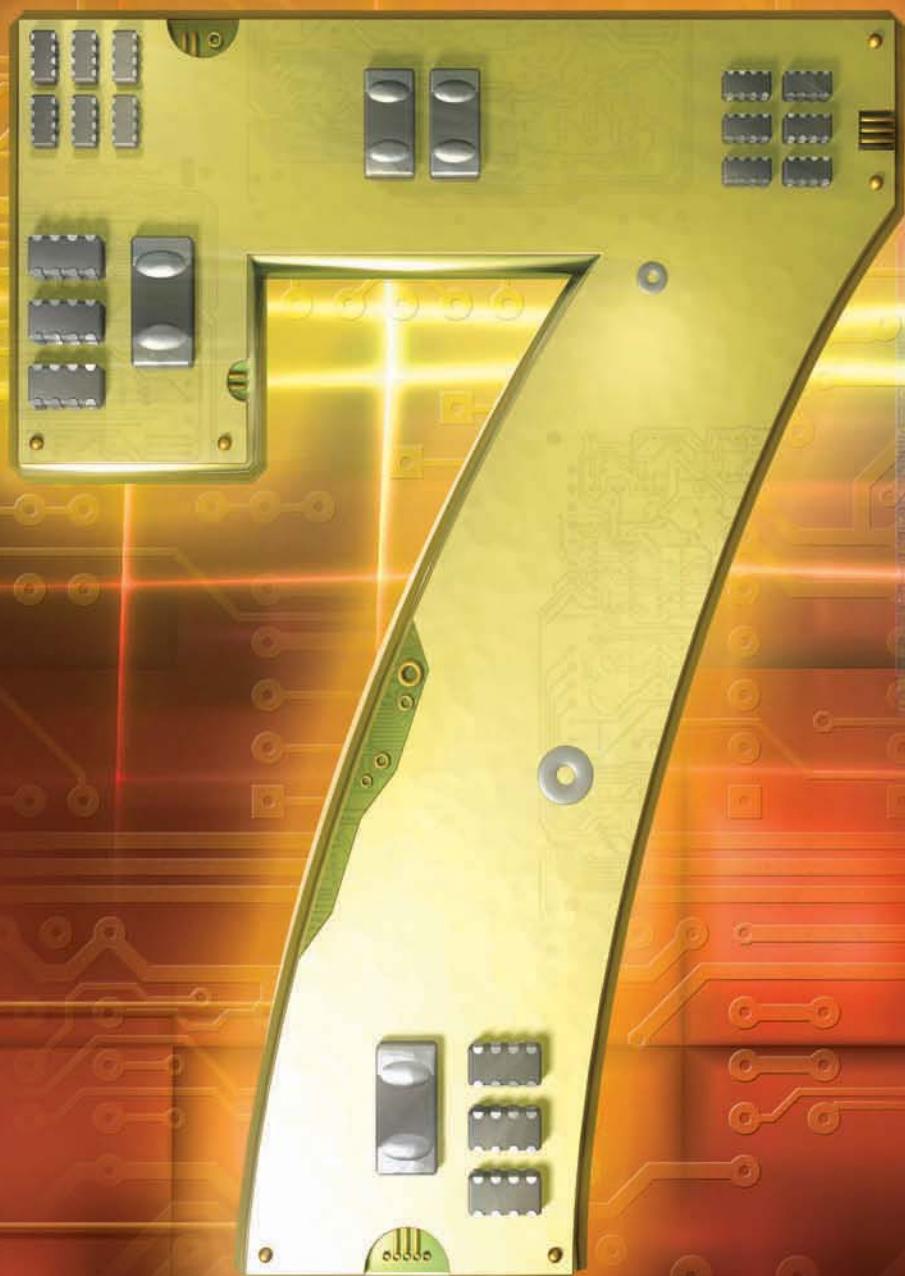
Xclamations! Share your wit and wisdom by supplying a caption for our wild and wacky artwork...**66**



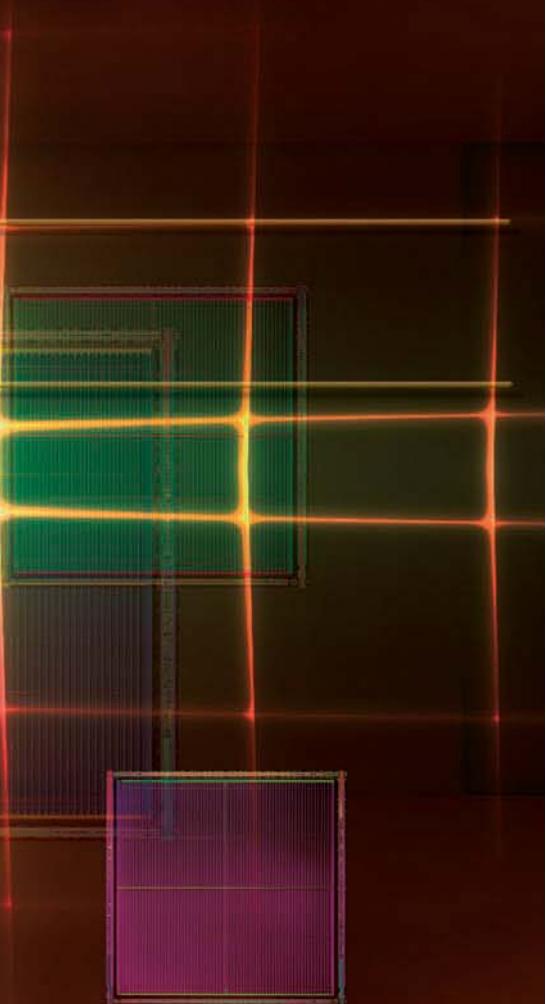
Xcell Journal recently received 2010 APEX Awards of Excellence in the categories "Magazine & Journal Writing" and "Magazine & Journal Design and Layout".



Xilinx Redefines State of the Art With New 7 Series FPGAs



Three families of 28-nm devices attack the mainstream and high-end ASIC and ASSP markets.



by Mike Santarini
Publisher, Xcell Journal
Xilinx, Inc.
mike.santarini@xilinx.com

FPGAs have advanced remarkably ever since they first hit the market in the mid-1980s as 1,500-ASIC-gate-equivalent devices. Two decades later, with the launch of Xilinx's new 7 series, the FPGA stands poised to fulfill its historic promise of one day displacing ASICs as the electronics industry's mainstream logic IC. With the introduction of the 7 series FPGAs, Xilinx® is transitioning from being just a PLD maker to a premier supplier of logic ICs by providing lower total cost of ownership for low- to medium-volume applications and equivalent total cost of ownership for higher-volume applications traditionally addressed by ASICs and ASSPs. What's more, this total-cost-of-ownership benefit combines with the traditional FPGA advantages of faster time-to-market and risk reduction. Together, all of these factors mean that FPGAs are emerging as the de facto logic IC solution for most applications.

As part of the 7 series release, Xilinx will bring to market an unprecedented 2 million-logic-cell FPGA, which is 2.5x the capacity of the largest Virtex®-6 device. Depending on whom you ask, how you design and what application you are targeting, that means the largest 7 series FPGA delivers the clout of anywhere from 15 million to 40 million equivalent ASIC gates. Thus, in the last 10 years, Xilinx has increased the capacity of its FPGAs by more than 30x at equivalent price points of the devices it produced 10 years ago.

But a huge capacity increase is only the beginning of the 7 series story. These beefy FPGAs run faster than the previous-generation Virtex-6, but at half the power consumption.

"ASICs are not dead, nor will they die entirely, but they really are only viable for a very small number of applications that have the very highest volumes," said Moshe Gavrielov, Xilinx's chief executive officer. "Where once you had to ask why would you go with an FPGA, today you have to seriously ask yourself why wouldn't we use an FPGA?"

The new 7 series is the first Xilinx FPGA family created entirely under the watch of Gavrielov, who joined the company in late 2007 after serving as a CEO of Verisity, a design tool provider. Before that, he worked for many years in management at ASIC house LSI Logic. Gavrielov has set Xilinx on an aggressive path to growth, with the main driver being an industry-leading line of FPGAs, which culminates in the 7 series, and the Targeted Design Platform strategy (see cover story, *Xcell Journal* No. 68; <http://www.xilinx.com/publications/archives/xcell/Xcell68.pdf>).

To enable this growth, the 7 series boasts several significant refinements, including a new unified and scalable architecture, a primary emphasis on power reduction and massive capacity, enabling better overall system performance (Figure 1).

Starts With a Unified Architecture

Up until the introduction of the 7 series, Xilinx's FPGA landscape has primarily centered on the high-performance Virtex family and the high-volume Spartan® family. When Xilinx originally introduced these two lines in the late 1990s, the Virtex and Spartan devices used radically different architectures. From a user perspective, the two families had notable differences, and so did the IP for each device and the design experience in working with them. If you wanted to increase the size of your end product from a Spartan design to a Virtex design or vice versa, the differences in architecture, IP and pin counts became apparent.

But with the unified architecture of the 7 series, those variances disappear. With the 7 series, Xilinx will not be introducing a new device under the Spartan name and has instead developed a complete lineup of FPGAs—primarily in three families, from lowest cost to highest performance—all based on the familiar Virtex FPGA architecture (Figure 2).

Virtex remains the moniker for the 7 series' highest-end FPGAs. This new Virtex-7 family delivers breakthrough capacity with up to 2 million logic cells and better than twofold system performance improvement over previous generations.

For a smooth transition from Spartan-6 FPGAs in the low-cost market, the new Artix™-7 family leads the industry in price, power and small form factor for cost-sensitive, low-power applications.

The final member of the trio smoothly fills the space between the Virtex-7 on the high end and the Artix-7 line at the mass-market level. The Kintex™-7 introduces a new price/performance advantage and gives Xilinx a platform for displacing mainstream ASICs and ASSPs.

Victor Peng, senior vice president for programmable platforms development at

Xilinx, predicts that having a solid midrange product in the form of Kintex-7 will allow Xilinx to offer a comprehensive FPGA lineup that is more highly application targeted.

"In previous generations, Xilinx would fill that middle ground by creating a higher-performance, higher-capacity version of Spartan and at the same time a lower-cost, lower-capacity and lower-performance version of Virtex," said Peng. "But the Spartan and Virtex architectures, IP and pin counts were very different. With the Artix, Kintex and Virtex families all built

in a unified 7 series architecture, customers will find it much easier to migrate their designs across the families, enabling better leverage of their development investments in IP."

They will also be able to migrate design blocks in the 7 series to the logic portion of the upcoming Extensible Processing Platform (see cover story, *Xcell Journal* No. 71; <http://www.xilinx.com/publications/archives/xcell/Xcell71.pdf>), because the EPP and 7 series devices all use the same Virtex logic architecture structure. Further, the common logic architecture supports

going forward. "It means our organization can focus on doing things once rather than twice," said Peng.

28-nm HPL: Right Mix of Power, Capacity and Performance

With the new 7 series family, Xilinx has also modified its manufacturing strategy to better align with the realities of modern IC design by choosing to implement its devices on a newly refined high-k metal gate (HKMG) high-performance, low-power (HPL) process with Taiwanese foundry TSMC.

Traditionally, FPGA vendors have implemented their designs on the highest-performance variation of each new silicon process as fast as foundries could make the processes available. However, starting with 90-nm process technologies, leakage started to become a big problem. It only got worse at 65 nm and 40 nm. At the 28-nm process node, if unaddressed, leakage current can account for well over 50 percent of a device's power consumption. In addition to using power when a device isn't running, during operation leakage creates extra heat, which in turn increases the leakage. Especially in continual-use, high-performance applications, this vicious cycle can lead to shortened device lifetimes and catastrophic IC failures. This greatly impacts the viability of using an FPGA in a given application as well as the reliability of a system.

The foundries have made remarkable strides to stem the leakage in their high-performance processes at 28 nm. Xilinx worked with its new foundry partner, TSMC, to refine the foundry's new HKMG HPL process for the 7 series FPGA, emphasizing low power combined with the usual gains in capacity and system performance when shrinking the geometry.

Peng said that by going with the HPL rather than the HP process, Xilinx will reduce static power by 50 percent with less than a 3 percent impact on performance. The use of the HPL process combined with the comprehensive power-savings enhancements implemented in the 7 series results in a 50 percent reduction in total power compared with devices at the same densities in the last generation of products.

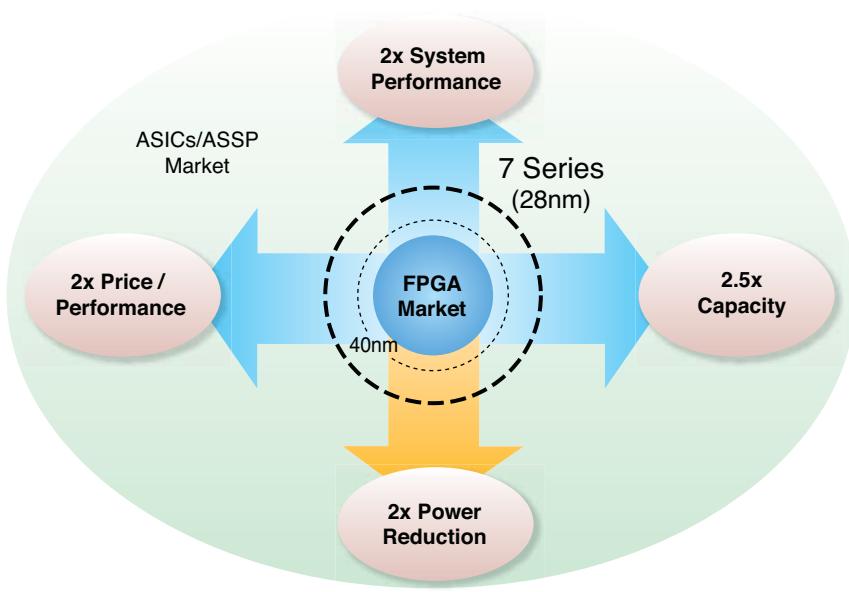


Figure 1 – The new Xilinx 7 series pushes the boundaries for FPGAs.

Xilinx, predicts that having a solid midrange product in the form of Kintex-7 will allow Xilinx to offer a comprehensive FPGA lineup that is more highly application targeted.

the ARM AXI4 (Advanced Extensible Interface) protocol. This means that Xilinx's internal IP developers and hundreds of IP partners can more easily target and implement AXI-compliant IP on Xilinx FPGAs. Chances are many customers have their own IP already built to comply with AXI, further facilitating moving designs from ASIC or ASSP to 7 series FPGAs.

Peng notes that in addition to offering great benefits for customers and IP partners, the unified architecture allows Xilinx to become more focused and aligned in all product development efforts

The 50 percent lower-power benefit gives design teams two options, said Peng. "You either run a similar-size Virtex-6 or Spartan-6 design at half the power in the 7 series, or you can double the size of the logic functions in your [new] design and remain at your previous power budget," he said. "By going with the HPL process, we have given customers much more usable performance as well as more logic gates to implement more functions in their designs."

achieve performance is through higher integration and efficiency, as opposed to simply making things move faster," said Gavrielov. "With today's processes, if you just make things faster, you drain more power and create thermal problems—which degrades power and performance. We need to pay a lot of attention to end-customer applications and ensure we strike the right balance between meeting low-power needs while simultaneously meeting

even fan or liquid cooling and related power circuitry to the end system.

HPL is just one of about a dozen technologies Xilinx employs to reduce power in the 7 series, Gavrielov said. For example, Xilinx reduced configuration logic voltage from 2.5 to 1.8 V, and optimized each of the hard blocks—DSP, Block RAM, SelectIO™ and others—using HVT, RVT and LVT transistors to reduce static power while optimizing performance and area. As a result, each

3 New Families Based on a Unified Architecture

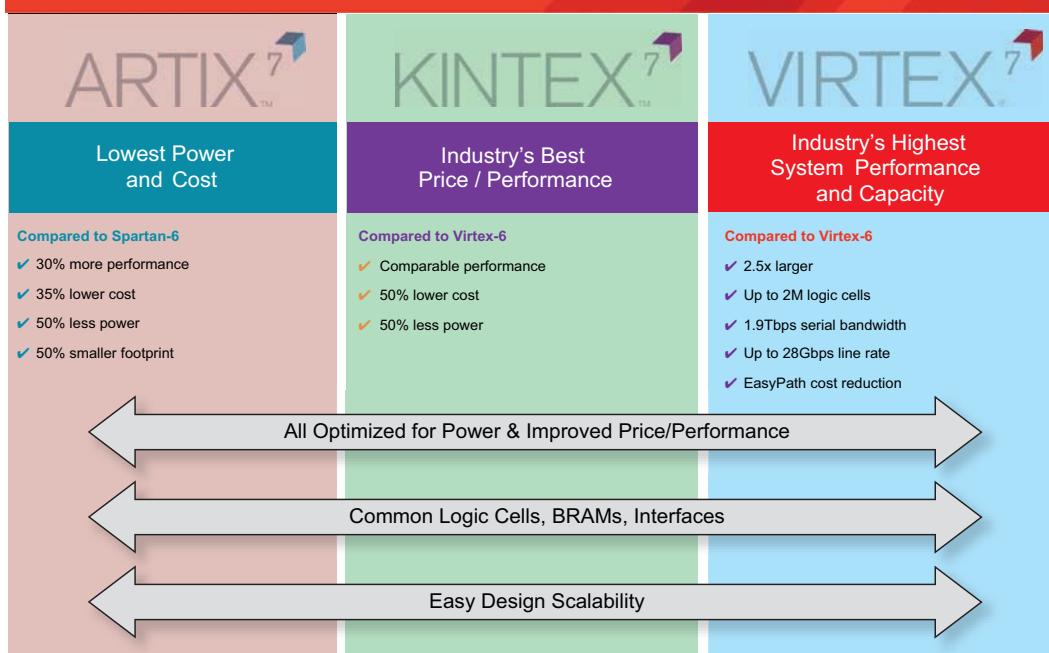


Figure 2 – The three new families in the 7 series unified architecture offer users a smooth path from lowest cost to highest volume.

Xilinx CEO Gavrielov notes that by choosing the higher-capacity but lower-power variant of the 28-nm process, Xilinx is leading the FPGA industry in aligning with the microprocessor industry. Almost a decade ago, MPU makers realized that cranking up clock rates in these newer process geometries would only create extremely leaky, thermally challenged devices that would fail.

"We learned from the processor side of the semiconductor business that given the realities of processes today, the best way to

the system performance needs of the application. We believe we will deliver a great value proposition with the 7 series FPGAs that will delight customers."

Peng notes that had Xilinx gone with an HP process for an incremental clock speedup, the significant increase in power for the fairly insignificant increase in performance would have burdened users with paying extra attention to designing around power and thermal issues. They might have incurred extra systems costs associated with adding elaborate heat spreading or

DSP slice consumes 1/12 the power of the equivalent logic implementation. By optimizing the ratio of these tightly integrated hard blocks throughout the FPGA fabric, Xilinx was able to achieve the greatest performance and lowest power while preserving flexibility.

Customers can also use the intelligent clock-gating feature introduced in the ISE® Design Suite 12 to give their 7 series designs an additional 20 percent reduction in dynamic power consumption. And finally, users can get a dramatic savings in power consumption by leveraging Xilinx's fourth-

The Virtex-7 family takes the industry's most successful FPGA architecture to new heights by doubling the capacity and boosting system performance at 50 percent lower power.

generation partial-reconfiguration methodology to effectively "turn off" portions of the design when they are not in use.

The upshot? By going with an HPL process, taking other power reduction measures and rolling out its new devices in a unified architecture, Xilinx now offers a comprehensive line of FPGAs, from the high-volume low-power lines to those boasting the highest system performance and capacity the industry has seen to date.

The Virtex-7, Kintex-7 and Artix-7 Families

Patrick Dorsey, senior director of marketing at Xilinx, said the three new families in the 7 series will allow Xilinx to capture an even greater share of the ASIC and ASSP market and penetrate even more deeply into a broader number of vertical markets, from low-powered medical devices to the highest-performing wired and wireless networking equipment.

At the entry level, "The new Artix-7 family provides the lowest absolute power and cost, with small-form-factor packaging," said Dorsey. Densities range from 20,000 to 355,000 logic cells. The devices are 30 percent faster and consume 50 percent less power than Spartan-6 FPGAs at 35 percent lower price points. When moving from Spartan-6 FPGAs to Artix-7 devices, designers can expect up to 85 percent lower static power and up to 35 percent lower dynamic power consumption.

GTP serial transceivers support line rates up to 3.75 Gbits/second. Other key features include 3.3-volt-capable I/O for interfacing to legacy components and wire-bond packaging for the lowest cost, with optional chip-scale packaging for the smallest form factor and 1.0-mm ball spacing for low-cost PCB manufacturing.

Dorsey said that because the new family is based upon the Virtex architecture, it also now includes many of the advanced features of the Virtex family that were not available in the Spartan line. For example, the Artix-7 includes an enhanced System Monitor ana-

log function, now called XADC (analog capability), to allow users to monitor the functionality, temperature, touch sensor, motion control and other real-world analog activities in the system. The integrated XADC technology will enable a whole new class of mixed-signal applications.

Further, with these refined specs, the Artix-7 FPGAs better target the low-power performance requirements for applications such as ultrasound equipment. The devices also now address the small-form-factor, low-power requirements of lens control modules for high-end commercial digital cameras as well as next-generation automotive infotainment systems driven by 12 V. Artix-7 devices also meet the strict SWAP-C (size, weight, power and cost) requirements of military avionics and communications applications.

Kintex-7 FPGA Family

Dorsey said that with the new midrange family, Kintex-7, Xilinx now provides FPGAs with the best price-for-performance on the market. "With the Kintex-7 family, we offer devices that are less than half the price and power consumption of Virtex-6 FPGAs but equal in performance and functionality," said Dorsey.

The Kintex-7 devices will be especially welcomed in applications that require cost-effective signal processing, he said. That's because they offer abundant DSP slices (from 120 to 1,540), up to 5,663 kbytes of distributed static RAM and 28,620 kbytes of internal block static RAM, and between four and sixteen 10.3-Gbps GTX serial transceivers. Dorsey said Kintex-7 devices will be equally attractive to Virtex users seeking a lower-cost alternative as well as to customers who have traditionally used Spartan FPGAs but are scaling their designs to the next level of system performance. Indeed, with logic densities ranging from 30,000 to 400,000 gates and with 40 percent higher performance than Artix-7 FPGAs, Kintex-7 devices equal the performance of Virtex-6 and are significantly faster than Spartan-6 FPGAs.

Dorsey said the Kintex-7 parts are ideal for implementing Long Term Evolution (LTE) wireless radio and baseband subsystems. And thanks to the recent release of Xilinx's fourth-generation partial-reconfiguration technology, 7 series customers can further reduce power and cost, for wide deployment in femto, pico and mainstream base stations. The serial connectivity, memory and logic performance of these devices is a good fit for high-volume wired communications as well, Dorsey added, citing equipment such as 10G passive optical network (PON) optical line terminal (OLT) line cards that bring high-speed networking to the neighborhood and home as one example.

In addition, Kintex-7 FPGAs are also suited for use in high-definition 3D flat-panel displays in consumer electronics markets; video-over-Internet Protocol bridges that enable next-generation broadcast video-on-demand systems; and high-performance image processing required for military avionics and ultrasound equipment that can support up to 128 high-resolution channels.

Virtex-7 FPGA Family

For its part, the high-end Virtex-7 family takes the industry's most successful FPGA architecture to new heights by delivering more than a doubling in capacity and 30 percent faster system performance along with 50 percent lower power than the Virtex-6 FPGA predecessors.

Dorsey said the Virtex-7 FPGAs are well suited for communications systems requiring the highest performance, capacity and bandwidth. With its Virtex-7T and Virtex-7XT variants, this FPGA line boasts ultrahigh-end devices that push the limits of FPGA technologies in terms of the number and performance of embedded serial transceivers, DSP slices, memory blocks and high-speed I/O to establish new benchmarks for the industry.

Virtex-7 devices offer up to 36 GTX 10.3-Gbps serial transceivers, ultrahigh-end logic capacity with as many as 2 million

logic cells and the highest parallel I/O bandwidth in the industry, with up to 1,200 SelectIO™ pins. This I/O configuration enables the greatest number of parallel banks of 72-bit DDR3 memory available on the market, supporting 2,133 Mbps.

Meanwhile, the new Virtex-7XT devices also provide the highest serial bandwidth in a single FPGA, with up to 72 GTH transceivers at 13.1 Gbps, or 80 GTH and GTX transceivers (24 running at 13.1 Gbps and 56 at 10.3 Gbps, respectively). In addition, the devices feature higher DSP-to-logic ratios for greater throughput, with up to 3,960 DSP slices at 600 MHz delivering 4.7 TMACs. Also, the 7XT FPGAs have higher on-chip BRAM-to-logic ratios with up to 65 Mbits, for low-latency data buffering. Dorsey said that Xilinx will eventually add devices with 28-Gbps transceivers to this family; the release details are forthcoming.

The new Virtex-7 FPGAs target the highest-performance wireless, wired and broadcast infrastructure subsystems, said Dorsey. The teraMACC signal-processing capabilities of Virtex-7 FPGAs enable advanced radar and high-performance computing systems. Product developers can replace ASICs and multichip-set ASSP solutions with single-FPGA implementations of 100GE line cards to increase bandwidth, while simultaneously lowering the power and cost. Other applications include 100-Gbit Optical Transport Network (OTN) muxponders for integrated multiplexer/transponder applications, 300G Interlaken bridges and 400G optical network cards.

In addition, these ultrahigh-end devices provide the logic density, performance and I/O bandwidth needed to build next-generation test and measurement equipment. For systems where ASIC production is justified, Virtex-7 FPGAs enable designers to use fewer devices during prototyping and emulation in order to lower cost and reduce interconnect/design complexity.

EasyPath—a Further Cost Alternative

Dorsey said the company's EasyPath™ program extends the value of Xilinx 7 series FPGAs to provide the lowest total cost of ownership for medium- to higher-volume

applications on the order of 100,000 units. This total cost of ownership requires that customers assume only the development and unit cost. In addition, they receive the full advantages of time-to-market and risk reduction that FPGAs offer. This further bolsters Xilinx's value as a strategic logic IC supplier.

EasyPath provides a cost reduction by coupling Xilinx's FPGA manufacturing process to the customer's design. This results in the same silicon with the same features, but only guaranteed to work with a given design. Dorsey said that EasyPath-7 takes six weeks to complete from design freeze and offers a guaranteed 35 percent cost reduction and no minimum-order quantity, with no customer engineering effort required—all for a \$300,000 nonrecurring engineering cost

"Now you have the peace of mind that once you design the FPGA, you can get to lower cost by targeting either Kintex-7 or Artix-7 and, if further cost reductions are necessary to support higher volumes, by going to EasyPath-7," said Dorsey. "What's more, if you've already completed your FPGA design and want to go with EasyPath, you can let your purchasing department handle the rest, since no further customer engineering resources are required."

Next-Generation Targeted Design Platforms

Along with announcing the new family, Xilinx is also launching a second generation of Targeted Design Platforms, application-specific design aids which the company first rolled out in 2009 in tandem with the release of the Virtex-6 and Spartan-6 FPGAs. Xilinx's Targeted Design Platform strategy gives system designers access to simpler, smarter design methodologies for creating FPGA-based solutions through the integration of five key elements: FPGA devices, design tools, IP, development kits and targeted reference designs.

Early-access ISE Design Suite software supporting the new FPGA families has shipped to a limited number of early-adopter customers and partners. First shipments of the devices will begin in the first quarter of 2011.

For more information on the 7 series FPGAs, visit <http://www.xilinx.com/technology/roadmap/7series-fpgas.htm>.

SPARTAN-6 MODULES

DESIGNED FOR LOWEST COST

108 I/Os
-40/+85°C
68x30 MM

- SO-DIMM Form Factor
- 16 MB Quad SPI Flash
- 128 MB DDR3 SDRAM
- Real-Time Clock

MARS MX1	▪ Dual Fast Ethernet PHYs (Designed for EtherCAT)
MARS MX2	▪ High Speed GTP Transceivers ▪ PCIe 1X Endpoint ▪ Gigabit Ethernet PHY

ALSO AVAILABLE:

- CARRIER BOARD DESIGN SERVICES
- CUSTOM CONFIGURATIONS

SPARTAN-3A DSP

PERFECT FOR DSP & SDR

USB 2.0 HIGH SPEED
142 I/Os
76x54 MM

- 128 MB DDR2 SDRAM
- 1 MB 250 MHz SSRAM
- 3 User Configurable PLLs
- Windows & Linux Drivers
- System Monitor

TFT CONTROLLER

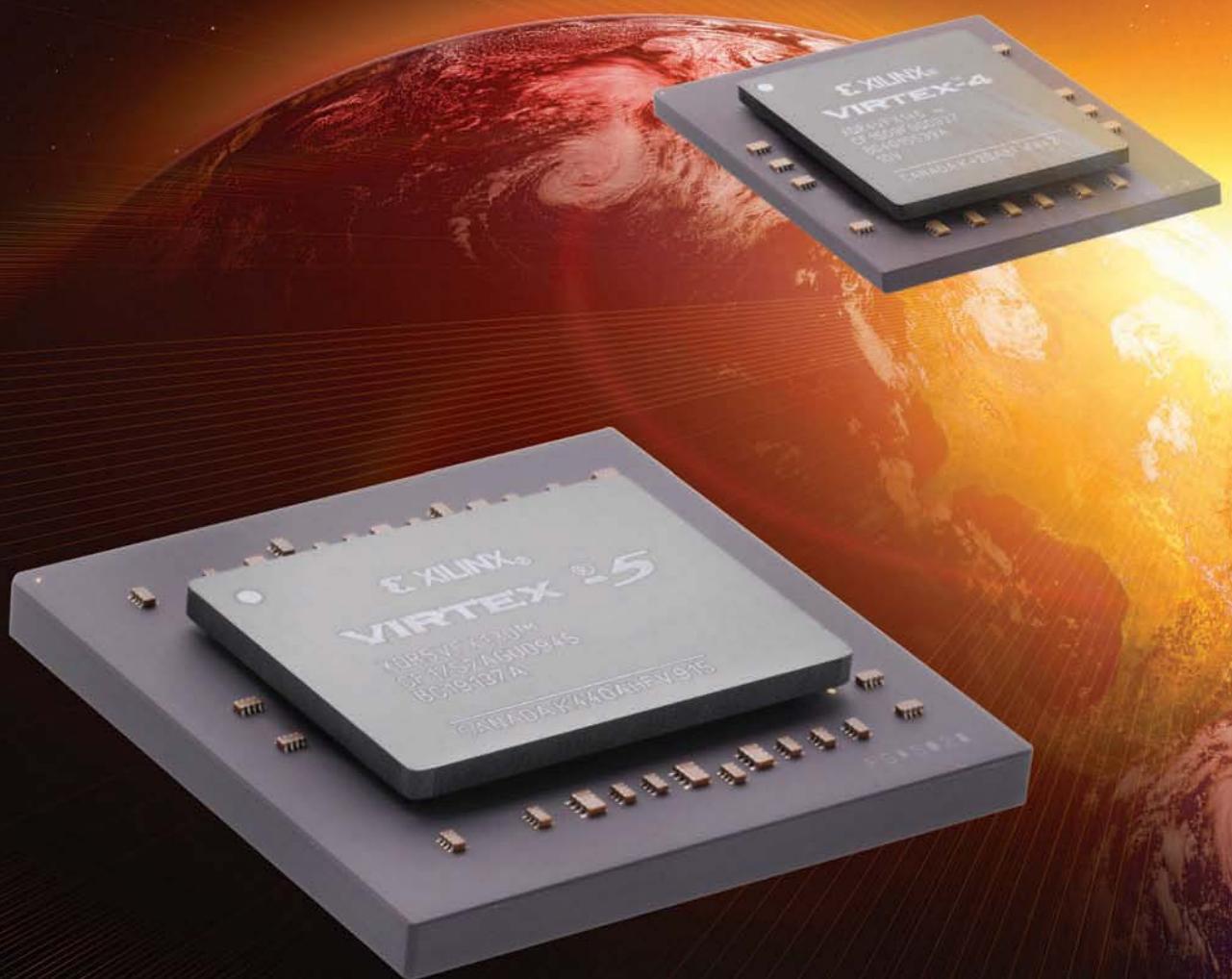
- Microblaze Interface
- Parallel, LVDS and DVI
- Up to 1920 x 1080
- Optional 2D Accelerator

DRIVE CONTROLLER

- DC, BLDC and Stepper
- Field Oriented Control
- Up to 100 kHz Control Loop
- Up to 8 Drives per Core

enCLUSTRA
FPGA SOLUTIONS
WWW.ENCLUSTRA.COM

New Xilinx Rad-Hard FPGA Reaches for the Stars



Virtex-5QV device offers a flexible, cost-effective alternative for design teams working on advanced, reconfigurable space applications.



by Maury Wright
President
WDC Marketing

Electronic systems designs headed to space naturally require high reliability, but the design task is further complicated by exposure to radiation that can cause sporadic circuit failures. From a functional perspective, FPGAs with inherent reconfigurable attributes are a perfect match for space. FPGAs enable a single system to perform multiple tasks and let mission teams remotely reconfigure a system, either fixing a bug or adding new functionality. Now Xilinx® has an FPGA—the Virtex®-5QV—that is rad-hard and can deliver the full benefits of programmability to space programs. The design teams get an off-the-shelf solution with all the advantages of a 65-nanometer commercial SRAM-based FPGA, including ready access to development and prototyping tools.

It's hard to underestimate the value an FPGA can offer in an application such as space-bound systems. Once a system, satellite, rocket or spacecraft is deployed, there is little or no ability to make hands-on changes to it, so the reprogrammability of an FPGA is a huge benefit. To be sure, microprocessors and microcontrollers can also be reprogrammed. But FPGAs excel in data-flow applications where functions such as packet inspection or signal-processing algorithms implemented in hardware logic offer far more processing throughput than do traditional microprocessors. And the FPGA hardware can be easily reconfigured to support new algorithms.

Given the advances in circuit density and the mix of hardwired IP blocks and configurable logic, the latest FPGA technologies can capture the bulk of a system's functionality. For example, the Virtex-5QV includes Ethernet MAC functions and high-speed transceivers to go along with DSP slices and configurable logic (for details on the FPGA capabilities, see sidebar, next page).

A rad-hard IC that's derived from a commercial FPGA family also offers significant benefits in the development process. Design teams can do development work

with readily available commercial devices and development tools and then seamlessly move the design to the rad-hard target system platform at any point in the development process.

Space Presents Reliability Challenges

To deploy FPGAs in space applications, however, designers have to understand the environment and learn how to mitigate issues that affect reliability. For example, a number of radiation-induced effects have been identified as a problem area for space-based designs. The list includes single-event upsets, single-event functional interrupts, single-event latchups, single-

event transients and total ionizing dose effects. (See the second sidebar for more information on these effects.)

Designers working on space applications haven't traditionally had the freedom to use ICs such as FPGAs without carefully considering ways to mitigate radiation effects. Specialty ASIC houses have radiation-hardened IC manufacturing processes. But ASIC design cycles are lengthy and expensive, and the quantity of devices the application will actually need simply doesn't justify the time and effort, given viable alternatives.

The radiation-hardened ASIC processes are also many generations behind state-of-

the-art commercial IC processes. For example, the rad-hard ASICs are still in the 150-nm or less-dense process nodes. Indeed, modern FPGAs offer performance and circuit density that match those of radiation-hardened ASICs, along with much faster development cycles.

Radiation Tolerance and TMR

In the past, designers who wanted to use FPGAs have had to combine radiation-tolerant ICs with techniques that further mitigate single-event upset (SEU) effects. Xilinx has long addressed the need for radiation resistance in space-targeted designs. Radiation-tolerant FPGAs such as the

Rad-Hard FPGA Delivers State-of-the-Art Benefits

The Virtex-5QV offers a unique value proposition. This FPGA is rad-hard out of the box and also offers state-of-the-art reprogrammable-logic density and hardwired IP blocks. Design teams working on space applications get ASIC-like circuit density without the ASIC NRE costs.

The FPGA includes more than 130,000 logic cells for large, complex designs. The architecture is based on six-input LUTs and the IC employs a diagonal interconnect structure that ultimately packs designs more efficiently in terms of silicon utilization and results in better performance and lower power consumption.

The design is based on the second generation of Xilinx's Advanced Silicon Modular Block (ASMBL™) column-based architecture. ASMBL has allowed Xilinx to produce mixes of configurable logic and hardwired IP that are optimized for specific applications.

The Virtex-5QV includes 320 Enhanced DSP slices to complement the programmable logic. Each slice includes a 25 x 18-bit multiplier, an adder and an accumulator. Designers can cascade the IC's 36-kbit Block RAM elements to produce large, general-purpose memory arrays. The device includes 298 such blocks. Each block can also be configured as two 18-kbit blocks, so there is little wasted silicon for applications requiring smaller RAM arrays.

For networking and I/O operations, the Virtex-5QV includes a number of hardwired IP blocks. Six Ethernet media-access controller (MAC) functions can operate in 10-, 100 and 1,000-Mbps modes. Eighteen RocketIO™ transceivers support data transfers at rates ranging from 150 Mbps to 3.125 Gbps. The MACs can use some of the RocketI/O transceivers for physical-layer (PHY) connections or link to external PHYs via a soft Media Independent Interface implemented in programmable logic.

The IC also includes three PCI Express® blocks compatible with the PCI Express Base Specification version 1.1. Designs can implement x1-, x4- or x8-lane channels with each of the three blocks. The RocketIO transceivers are also available for PCI Express I/O.

The device features a number of other functions important in high-performance system designs. Six clock management tiles (CMTs) can each generate clocks that operate up to 450 MHz. Each CMT includes dual digital clock managers (DCMs) and a phase-locked loop (PLL). The DCMs enable zero-delay buffering, frequency synthesis and clock-phase shifting. The PLLs add support for input jitter filtering and phase-matched clock division.

Xilinx will manufacture the IC in a 65-nm copper CMOS process with a 1-V core voltage. A ceramic flip-chip column grid array package will ensure signal integrity. And Xilinx will guarantee operation over the full military temperature range of -55°C to +125°C.

— Maury Wright

The Virtex-5QV design team aimed to provide intrinsic hardness from single-event upsets and transients to critical elements in the device.

A novel latch implementation offers almost 1,000 times the hardness to SEUs of commercial versions of the FPGA.

Virtex-4QV have immunity to single-event latchup (SEL), and can withstand a total ionizing dose (TID) up to 300 krads(Si). Xilinx combines these radiation-tolerant ICs with system-level techniques such as triple-modular redundancy (TMR) to ensure reliability. In a TMR design, three separate instantiations of a system perform the same task. A voting circuit compares the results and considers it correct if at least two systems produce the same result.

Xilinx has developed a tool that can greatly simplify the implementation of a TMR methodology. The TMRTTool accelerates the design cycle by allowing the design team to focus on design and debug rather than TMR. The tool works seamlessly with any HDL and synthesis tool to automatically build TMR into a design.

The TMRTTool also goes beyond baseline TMR functionality. It triplicates all clocks and throughput logic to protect against single-event transients (SETs). It also triplicates feedback logic and inserts majority voters on all feedback paths. And the tool triplicates all outputs and uses minority voters to detect and disable incorrect output paths.

By inserting voters on all feedback paths, the TMRTTool overcomes a problem with the technology in designs with finite state machines. In most TMR-based state-machine designs, an SEU that causes an error in one of the three state machines ultimately requires that the state machines be reset for synchronization. But the voters in feedback paths ensure that the state machines remain continuously synchronized and can operate continuously through SEUs.

Rad-Hard by Design

While earlier FPGAs such as the Virtex-4QV have been successfully deployed in space applications and have been radiation-tolerant, the new Virtex-5QV was designed

from the ground up with a rad-hard-by-design (RHBD) methodology. The resulting FPGA is truly a rad-hard space-grade IC. Where the Virtex-4QV requires that designers add mitigation, the Virtex 5QV is rad-hard out of the box.

The Virtex-5QV design team's specific goal was to provide intrinsic hardness from

sions of the FPGA. Moreover, the latch is virtually impervious to proton interaction.

Upset Hardening in Hardware

Effectively obviating the need for TMR at the application design level, the Virtex-5QV design team used a variety of techniques in implementing the underlying



Figure 1 – Xilinx built its latest space-grade FPGA, the Virtex-5QV, to be rad-hard.

SEU, SET and other effects to critical circuit elements in the device. As with all SRAM-based FPGAs, configuration memory controls all aspects of device operation and is therefore critical to reliable operation. The Virtex-5QV design utilizes dual-node latches that control write operations to memory cells. Writes occur only when both latches are enabled synchronously. The implementation offers 1,000 times the hardness to SEUs relative to memory latches in commercial ver-

FPGA memory and circuit elements in the device. Xilinx took special care in hardening the 35 million configuration cells and the 81,920 user flip-flops. Both incorporate a clever self-redundant storage circuit that has double the normal number of transistors. The result is a very low susceptibility to SEU. Xilinx optimized the layout of this important structure by fabricating many variants and subjecting them to in-beam irradiation studies.

In addition, Xilinx made sure that all the clock, data and asynchronous inputs to the flip-flops are protected. These are capable of suppressing single-event transients and prevent them from turning into upsets.

Xilinx employs proprietary methods to protect against SEUs during the critical start-

up period when the FPGA is configured. One result is that designers don't have to take extra steps to ensure the elimination of single-event functional interrupts (SEFIs), which traditionally require an intrusive FPGA restart and reconfiguration. Gary Swift, senior staff engineer for space products

development and radiation testing at Xilinx, said the implementation reduces GEO (geostationary earth orbit) environment SEFI rates by more than two orders of magnitude (relative to the already low once per century for the rad-tolerant Virtex-4QV family) to about one SEFI fault in 10,000 years. "Interestingly,

Understanding Radiation-Induced Effects

Going back to the 1950s, engineers have documented the adverse effect that radiation can have on electronic circuits. With the advent of ICs and the constant move to finer process geometries, the potential for radiation-induced errors grew. The impact ranges from soft errors that are easy to detect and correct to actual device failures.

Design engineers working on space applications must prepare for a number of problems.

- A single-event upset (SEU) is a change of state in an IC, such as a change in the value of a memory bit caused by a radiation strike. SEUs are also called soft errors because the instance of an SEU has no long-term effect on the IC and in fact, the soft error can often be found and fixed.
- A single-event functional interrupt (SEFI) is similar to an SEU in that it is typically the result of a single ion strike. But SEFIs result in a temporary instance of some element of the IC not functioning properly. In some cases, an SEFI-induced fault remains until power cycles, and in other cases the condition is truly temporary.
- A single-event latchup (SEL) is a potentially more damaging event typically caused by ions or protons generated by cosmic rays or solar flares. The radiation induces a high-current state that results in a full or partial loss of IC functionality. In some cases, power-cycling an IC eliminates the SEL condition. In other cases the device may be permanently flawed.
- The single-event transient (SET) encompasses the concept of an SEU, but includes more-complex errors induced by a radiation strike. An SET, for instance, might affect a clock and propagate multiple errors throughout memory or logic.
- Total ionizing dose (TID) effects lead to the failure of an IC based on the aggregate exposure to radiation over time. Typically, performance parameters decline in an IC as the TID, measured in rads, increases over time. Radiation creates electron-hole pairs in the oxide layer of an IC—slowly changing the threshold voltage of transistors.

Xilinx and its partners have gone to great lengths to understand the radiation effects and to test and characterize mitigation techniques in FPGAs. Back in 2002, Xilinx and the Jet Propulsion Laboratory founded the Xilinx Radiation Test Consortium (XRTC—originally referred to as the Single-Event Effects Consortium, or SEEC). The consortium currently has 14 members, including universities, research laboratories, and defense and space contractors.

The partners are generally organizations that need to use FPGAs in space applications and have a vested interest in accurately assessing radiation-induced effects and radiation-tolerant and -hardened designs. The consortium website includes a comprehensive library of research papers (http://www.xilinx.com/esp/aero_defradiation_effects.htm) that have validated the reliable use of SRAM-based FPGAs in space applications.

Memory—including SRAM—has long been considered among the most susceptible circuits to radiation-induced effects and specifically to SEUs. That fact led some to question the suitability of SRAM-based FPGAs in space applications.

SRAM-based FPGAs, however, offer tremendous flexibility relative to programmable devices based on nonvolatile memory. For starters, SRAM-based FPGAs, and specifically those from Xilinx, have consistently offered the highest level of integration available in programmable ICs. Moreover, SRAM-based FPGAs are easily reconfigured, allowing a system to serve multiple applications and allowing teams to remotely reconfigure a system to fix a flaw in the system implementations that isn't revealed until after deployment. The capacity and performance of the most advanced SRAM-based FPGAs far exceed those of the most advanced programmable nonvolatile devices.

— Maury Wright

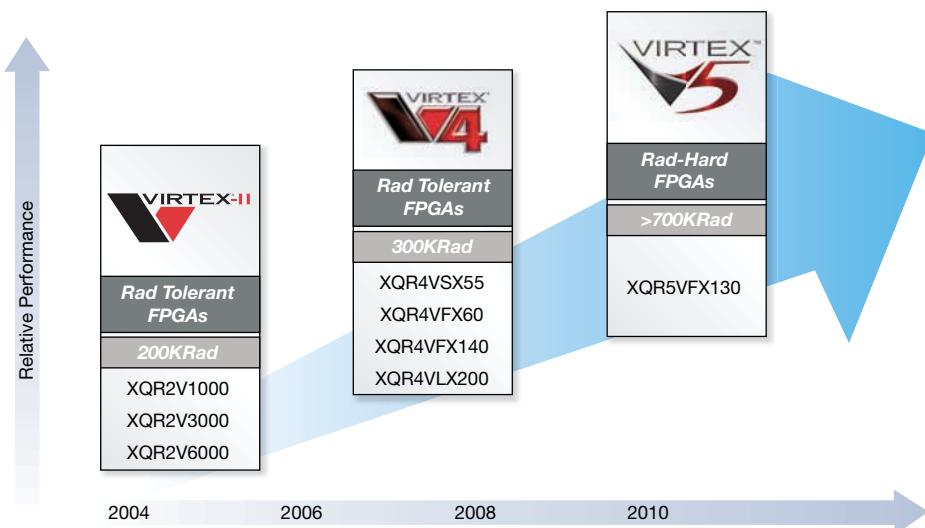


Figure 2 – The Virtex-5QV rad-hard FPGA is a crowning achievement in Xilinx's long, rich history of providing leading-edge IC technology for space applications.

the upset mitigation is so effective with virtually no susceptibility to protons that GEO is the worst-case orbit for SEFIs and SEUs, exactly the opposite of what space radiation experts have learned to expect," said Swift.

In addition, Xilinx's Digitally Controlled Impedance (DCI) allows adjustment output impedance and input termination values without external components. Finally, Xilinx ensured its Block RAMs are protected from outputting erroneous data in spite of upsets via an error-detection-and-correction circuit to eradicate upsets.

Virtex-5QV Delivers Results

The Virtex-5QV delivers results unmatched by previous FPGAs. The ICs are fully characterized for space radiation effects in heavy ion and proton environments. These FPGAs will withstand a TID of 700 krads(Si), based on method 1019 as defined in MIL-STD-833.

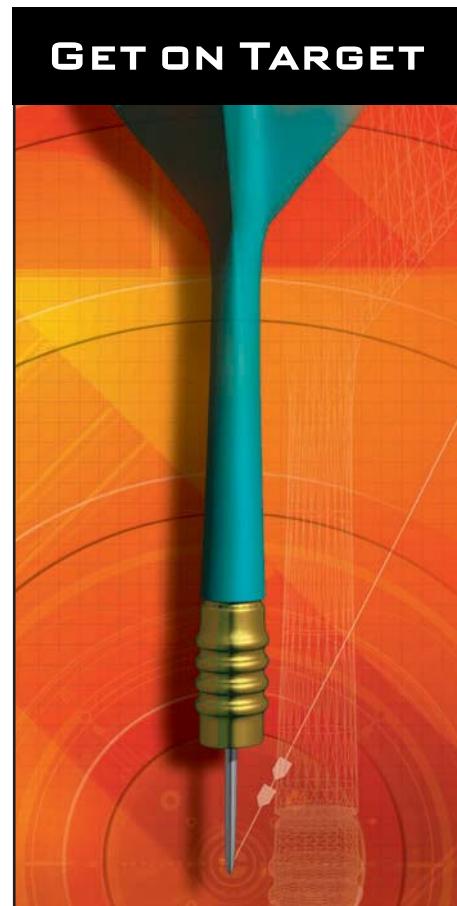
Immunity to single-event latchup is defined by a threshold to linear-energy transfer (LET)—the amount of energy transferred to material as an ionizing particle travels through that material. The Virtex-5QV meets the MIL-STD-833 requirement that LET is greater than 100 MeV/mg·cm² (mega electronvolts per milligram per centimeter squared). In short, the device is essentially immune to SEL effects.

SEU immunity in the configuration memory and control logic is defined in terms of deployment in a GEO environment relative to a space platform that travels 36,000 km per day. Based on 35 Mbits on an IC that could be subject to an SEU, the IC will suffer 3.8×10^{-10} errors per bit per day.

With the availability of the Virtex-5QV FPGA, space teams will have access to a state-of-the-art reprogrammable platform built on a 65-nm copper process technology. The teams can prototype their work with widely available commercial FPGAs and easily accessible development tools, and then deploy systems that the Xilinx Radiation Test Consortium has proven to be reliable in the harsh radiation environment in space.

The Virtex-5QV device will sample in the current quarter, with general production availability planned for first half of 2011. For more information, visit <http://www.xilinx.com/products/virtex5qv/index.htm>.

Maury Wright is an electronics engineer turned technology journalist and industry consultant, with broad experience in technology areas ranging from microprocessors to digital media, wireless and power management. Wright worked at EDN Magazine for 22 years, serving as editor-in-chief and editorial director for five years. Wright also served as editor of the EE Times Digital Home and Power Management websites.



IS YOUR MARKETING MESSAGE REACHING THE RIGHT PEOPLE?

Hit your target by advertising your product or service in the Xilinx *Xcell Journal*, you'll reach thousands of qualified engineers, designers, and engineering managers worldwide.

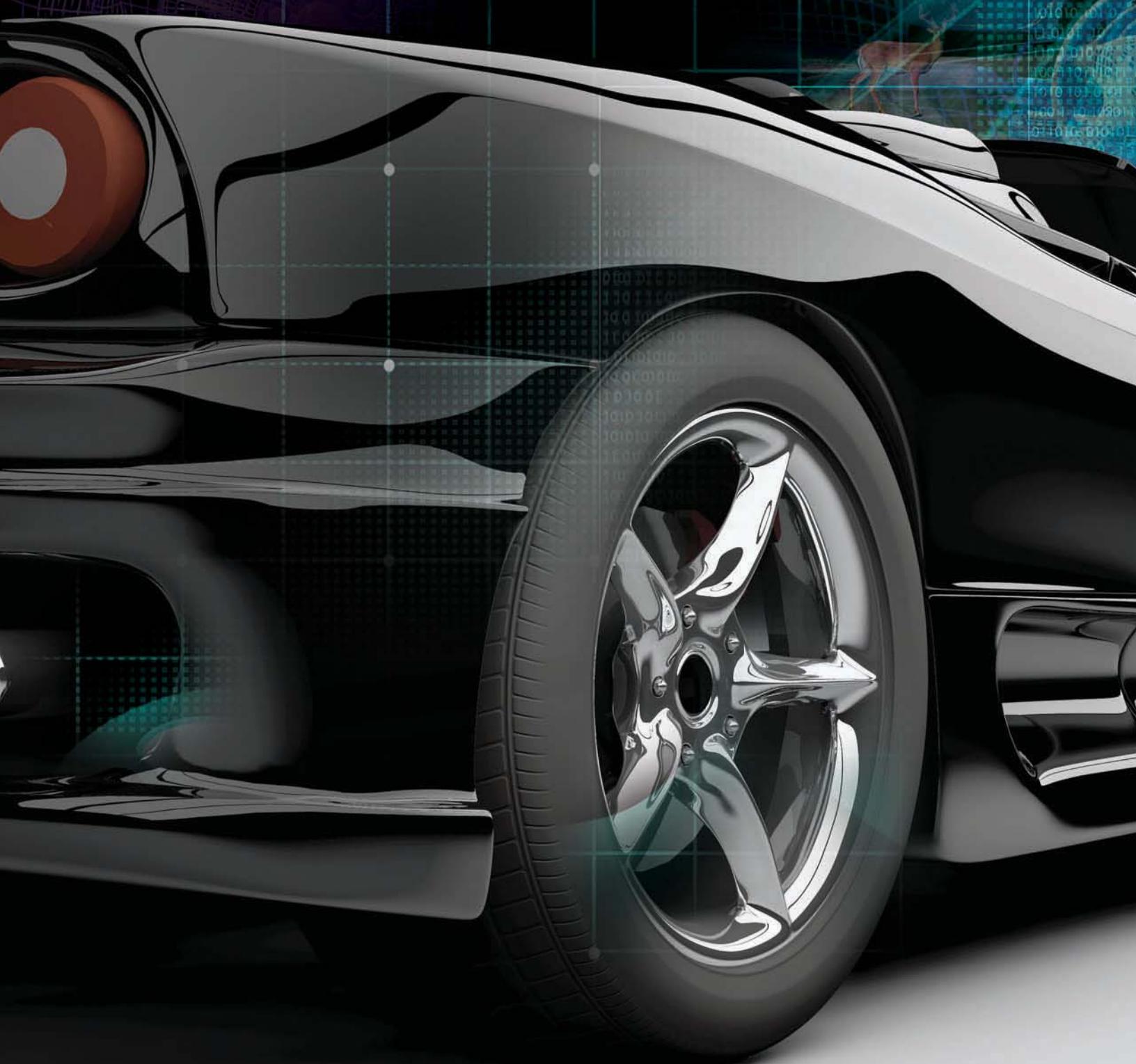
The Xilinx *Xcell Journal* is an award-winning publication, dedicated specifically to helping programmable logic users – and it works.

We offer affordable advertising rates and a variety of advertisement sizes to meet any budget!

Call today:
(800) 493-5551
or e-mail us at
xcelladsales@aol.com

Xcell
PUBLICATIONS

Multiple MicroBlazes Ease Integration in Real-Time Automotive System



When facing a project with ever-changing requirements and software contributions from multiple locations, multiprocessors can actually help.

by Martin Thompson
Principal Product Engineer
TRW Conekt
martin.j.thompson@trw.com

It is a commonly held view that it is harder to develop software for multiple-processor systems than single-processor systems. But in fact, this is not always the case. Our design team at TRW Conekt, the consultancy arm of TRW Automotive, recently undertook a project that demonstrates how partitioning the hardware to match the problem at hand allows development of very efficient systems using many processors.

Our team was tasked with providing embedded processing electronics to run in cars for a project known as “Foot-LITE” (led by MIRA Ltd. and sponsored by the U.K. government-backed Technology Strategy Board, the Department for Transport and the Engineering and Physical Sciences Research Council). This project provides feedback to drivers about their driving habits from the perspective of both safety and fuel economy.

The system gives the feedback to the driver in two ways. First, a dashboard-mounted smartphone display system (designed by Brunel University and developed by HW Communications Ltd.) provides real-time communication with the driver about events that require immediate attention. In addition, the system collects continuous journey data, including video streams of particular “events,” and then uploads them to an Internet-based server for users to view at their leisure. The decision about which events to flag to the user is made by an algorithm developed by partner Ricardo UK, based on driving advice from another partner, the Institute of Advanced Motorists.

The project will fit this system to a small fleet of 30 vehicles (see Figure 1). Test drivers will be members of the public employed by project partner Hampshire County Council.

The project has progressively incorporated the research results obtained by the collaboration of 12 industrial, governmental and academic partners. This means that we've needed a very flexible solution to our processing challenges.

At the beginning, we envisioned providing a single-processor system. However, it was soon apparent that a dedicated processor would ease the task of integration for each iteration of algorithmic development.

Base System

We had available to us a processing system, already under development in another project, to perform image-processing tasks (Figure 2).

This system is based around a single Xilinx® Spartan®-3A XC3SD3400A device connected to four independent blocks of DDR memory, an architecture that allows users to implement many different

additional processors by making use of embedded BRAM blocks.

In addition, I/O to the outside world is configurable using small daughterboards, which allows for quick turnaround of customized I/O sets for different projects.

The project partners decided very early on that a USB interface would be desirable, as it allows you to add a wide variety of peripherals to the system. This necessitated

with a USB Memory Stick that contains a script to upload new code to the internal flash. In a traditional embedded system, all this would require the software team to write low-level application code. However, with Linux available to us, we can easily write simple Bash scripts to control these processes.

Foot-LITE Algorithms

Ricardo developed the core algorithms that assess the driver's actions and implemented them on its rCube rapid prototyping system (<http://www.ricardo.com/en-gb/Engineering-Consulting/Automotive-Expertise/Controls--Electronics/Embedded-Software/rCube>). We used this approach for initial simulator trials and three test vehicles. In the test vehicles, an embedded vision system (based around an existing TRW product—coincidentally also containing an FPGA) measured distance to the vehicle in front and assessed the vehicle's position within the lane. A radar system provided an alternative source of range information in the test vehicles. As a step toward a production implementation, we eliminated the radar system for the larger-scale trials, as the vision system provided sufficient information for the application

We fitted the vehicle with a forward-facing video camera and processing subsystem, which are combined into a small unit that fits near the rear-view mirror. Embedded algorithms in this subsystem process the video images to measure the distance between the car and the edge of the lane. In addition, a parallel algorithm detects vehicles in front of the Foot-LITE vehicle and provides a measure of the headway distance. This subsystem transmits its data to the Foot-LITE system unit using the automotive-standard controller-area network (CAN) bus.

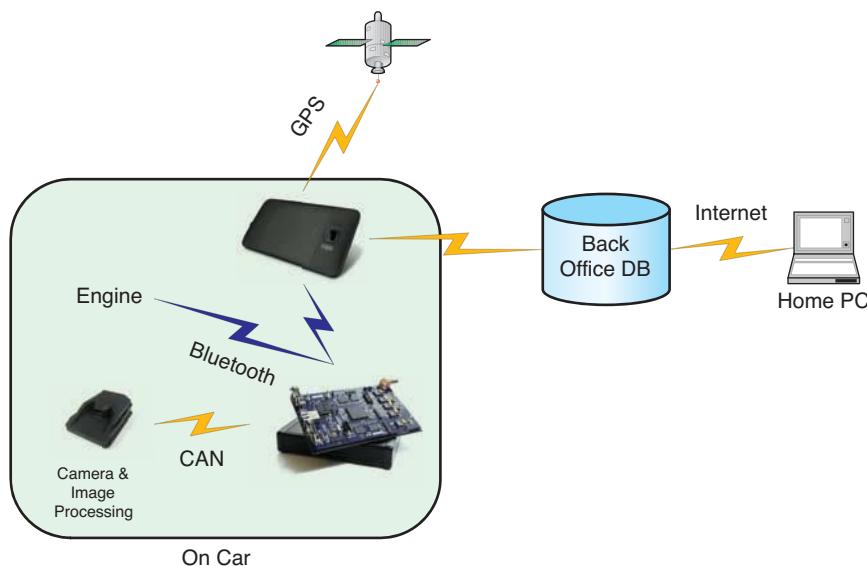


Figure 1 – Foot-LITE system

processor/logic configurations. For example, you could use the whole FPGA fabric as pure logic resources crafted entirely in HDL. Alternatively, you could use higher-level tools such as the Xilinx EDK to implement four (or more) soft-core microcontrollers. Each of the four would have access to its own private DDR memory device, protecting data from interference by the other microcontrollers. For other simple tasks, you could include

some form of USB stack—we obtained one using the Petalinux implementation of ucLinux—and a daughterboard with a USB host device.

The use of Linux also gives us a simple way to manage the SPI flash devices that the system provides for FPGA bitstream and application code storage. We installed a simple JFFS2 file system to allow in-field application updates, either over Ethernet (using FTP) or by booting

We integrated a three-axis accelerometer and yaw-rate sensing package into the Foot-LITE unit, potentially providing the Foot-LITE algorithms with access to high-rate, low-latency vehicle dynamics information when required.

The Foot-LITE algorithms fuse all this data to provide a set of simple outputs to the driver relating to his or her driving style.

Algorithm Implementation

At the beginning, we envisioned providing a single-processor system. However, it was soon apparent that a dedicated processor would ease the task of integration for each iteration of algorithmic development. We isolated the main host processor and the Foot-LITE algorithm processor by imple-

to the host application will not affect their application's performance.

We developed a collection of wrapper functions that allow us to "drop in" the code-generated C from the Simulink® compiler without having to make major changes to the interface. We provide a small amount of nonvolatile memory onboard via an I2C bus, which is required for storing various tune parameters within the Foot-LITE algorithms. This necessitated a simple wrapper to provide the algorithms with easy access from the Simulink environment to read this memory at startup and write it back at shutdown.

The system needed to measure accelerations and yaw rate, as well as communicating over the CAN bus with the lane- and

approach allowed us to split the task of writing the software to allow parallel development. Once again, we used FSL as the interface between the dynamics processor and the Foot-LITE algorithm processor.

Video Capture and Compression

The initial conception of the system provided simple measures of lane width and offset, distance to the vehicle in front and so on from the vision system, transmitted over CAN to the Foot-LITE algorithm unit. The project partners decided to enhance this setup by capturing video frames for transmission to the server, to provide off-line contextual assistance for interpreting the advice the system gave. Given that the requirement was only for "Internet-quality" video (300 x 200 pixels at 5 Hz), we felt we could easily assign a fourth MicroBlaze to the task of compressing the video stream to a simple set of JPEG images in real time. The image coming from the camera was a wide-VGA (720 x 480 at 30 Hz) video stream. Clearly, downsampling the image was a task to be performed in hardware.

We designed a simple peripheral to handle the downsampling operation by simply dropping alternate pixels and lines to produce a 360 x 240 image. This peripheral also drops four in five frames to produce the required frame rate. Nothing more complex is needed to produce visibly acceptable results, since the JPEG process renders aliasing artifacts invisible. We used System Generator to develop this peripheral, as it makes export to EDK very straightforward, and we already have experience of using System Generator for more-complex image-processing tasks.

The downsampling peripheral bus masters the data into the SDRAM connected to the JPEG processor, which then compresses each frame, as it arrives, into a circular buffer until the Foot-LITE algorithm sends a flag. The JPEG processor sends the compressed video frames (again over FSL) to the host MicroBlaze. We used a code library from the Independent JPEG Group and found that it needed very little optimization to operate at 5 Hz. Again, having an isolated processor



Figure 2 – Processing module is Spartan-based.

menting communications using the MicroBlaze™ Fast Simplex Link (FSL) bus system. This allows a complete isolation of the processors' memories (unlike the popular shared-memory methodologies), which greatly eases the integration task, since bugs cannot "migrate" from one processor to another via memory corruptions.

In addition, there is no competition for processor cycles, which means our partners can be confident that any changes we make

vehicle-detection system. As we already had low-level CAN drivers and were concerned as to the timeliness of a Linux application measuring the vehicle dynamics information at 40-millisecond rates, we decided to insert a third MicroBlaze into the system. This saved porting CAN drivers to Linux, and allowed deterministic performance via another isolated processing node—critical to the algorithms—which made use of the dynamics measures. In addition, this

enabled another software engineer (based at a different site) to work on this aspect of the system in parallel.

Bluetooth to Smartphone and OBD

Ease of installation was a critical factor for the project. Minimizing the number of

wires that the system requires was also an important consideration. We chose Bluetooth as the interface to the smartphone. Drivers for standard USB Bluetooth dongles are standard in the uClinux kernel, although we had to build the user-space tools ourselves. These have a

number of dependencies on other items of code, which are also cross-compiled with the Petalinux tool set and added to the uClinux file system.

Once we had decided on Bluetooth for the smartphone interface, it was a natural choice to use Bluetooth for the interface to

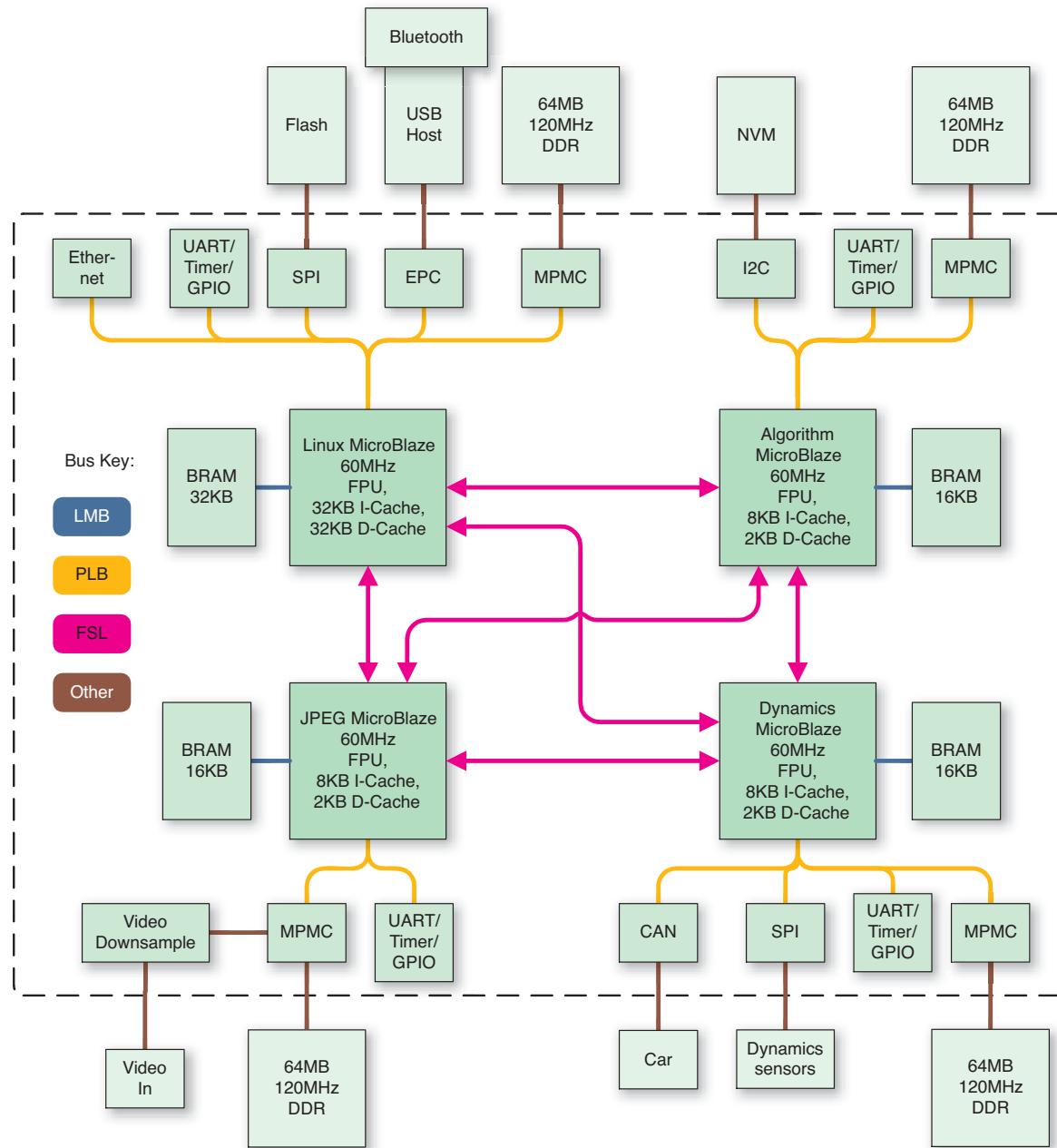


Figure 3 – FPGA block diagram showing major external elements

the On-Board Diagnostic System. We made use of a standard off-the-shelf Bluetooth-OBD interface module, removing another wired link from the system.

Easier Debugging

Debugging a system with multiple, parallel threads of execution is always challenging. But splitting the system across multiple processors actually makes things easier. There is no requirement for a multithread-aware debugger (as might be needed when trying to debug multiple processors within the Linux environment). The Xilinx debugger (XMD) can connect to multiple processors, and by using TCL (the Tool Command Language, which XMD understands), we can automate the setup and download of the code under test to multiple processors. Of course, the common embedded-system debug approach using printf statements was also available, since each processor has its own serial port.

Another tool of great value when debugging the interprocessor communications was ChipScope™ Pro. This embedded logic analyzer built into the FPGA fabric allowed us to capture the data passing over the FSL links and narrow down subtle bugs to either the sender or the receiver, and from there to the offending line of code.

The isolation provided by using four processors means that once a particular element is debugged it will (to a large extent) not need to be looked at again. There are none of the weird interactions that always cause problems when integrating disparate code into a large, monolithic application, or when running multiple processes on a single processor.

FPGA Implementation

In this project, there is almost no HDL—simply a top-level wrapper integrating the EDK-based design with a tiny piece of watchdog code to guarantee the system shuts down after the driver has turned the ignition off. EDK generates the vast majority of the FPGA (the MHS file is more than 1,300 lines long!), with System Generator producing the video downsample. We configured all four microcontrollers with caches and floating-point

units. With four processors, four DDR memory interfaces and a collection of peripherals (including Ethernet, SPI, IIC, CAN, UARTs, timers, GPIOs), around 70 percent of the device's lookup tables are occupied (around 28,000 LUTs). As is usual with microcontroller-based FPGAs, the block memories are very highly utilized at better than 90 percent, or 119 BRAMs, but the DSP blocks are relatively lightly used: Only the floating-point units in each processor (eight in each, for a total of 32) need them.

Bringing it All Together

The host microprocessor boots the Linux kernel from internal flash and then mounts its local file systems. The slave processors each have an FSL-based boot loader, which accepts a standard S-record, parses it, copies it into local memory and executes it. The Linux processor simply streams the S-record from the file system direct to the FSL pseudo-file (using the built-in dd utility). As described already, all interprocessor communication takes place over a fully connected mesh of FSL links. These are all 32 bits wide and operate at 60 MHz, providing plenty of low-latency communication bandwidth. Although avoiding shared memory may seem limiting, the upside is that this system provides the isolation benefits already discussed. The hardware architecture matches the application requirement subdivision well, which creates an intuitive software partition.

The Foot-LITE algorithm microprocessor sends triggers to the JPEG compressor when required and communicates with the smartphone display. The Linux processor intermediates between the Bluetooth communications and the rest of the system (Figure 3). In addition to the immediate signals to the driver, it sends a continuous stream of information about the state of the vehicle and occasional streams of video for onward transmission to a central server via the smartphone.

At the end of a journey, when the driver switches off the ignition, the main processor informs the slave processors, which can then perform their own shutdown procedures (such as writing updated

parameters to the nonvolatile tune storage) before informing the main processor that they are in a safe state for shutdown. At this point, the host processor signals to the power supply and the system enters a very low-power sleep mode, awaiting the next turn of the ignition. In the unlikely case that the software does not send the shutdown signal within a couple of minutes of the ignition turnoff, a hardware timer in the FPGA fabric pulls the power, to avoid flattening the vehicle battery.

In the final stages of the project, two academic members of the consortium (Newcastle University and the University of Southampton) will analyze the data streamed out of the vehicle in actual highway use to evaluate the effectiveness of the system in altering driver behavior.

The FPGA Advantage

FPGAs provide huge flexibility, which meets the needs of evolving projects much more easily than a fixed hardware platform. The ability to mix in custom hardware for intensive applications (for example, video) is also beneficial. If you use Linux to gain the huge benefits of ready-made high-level access to peripherals such as Ethernet, you don't need to compromise on real-time performance, as you can push those critical tasks into their own microprocessor. Finally, if a large, geographically distributed team is developing the software, having a hardware architecture that matches the functional split provides benefits in development and integration.

For more information, contact the author at martin.j.thompson@trw.com. You can read more about the Foot-LITE project at <http://www.foot-lite.net>. MIRA Ltd. is the project lead. The other industrial project partners are Auto-txt Ltd., Hampshire County Council, HW Communications Ltd., The Institute of Advanced Motorists Ltd., Ricardo UK Ltd., Transport for London, TRW Conekt and Zettlex Printed Technologies Ltd. The academic partners are Brunel University, Newcastle University Transport Operations Research Group and University of Southampton Transportation Research Group.

Making Biometrics the Killer App of FPGA Dynamic Partial Reconfiguration

Run-time reconfigurable hardware technology
brings key advantages in the design of
automatic personal recognition systems.

by Francisco Fons

PhD Candidate

University Rovira i Virgili, Tarragona, Spain
francisco.fons@estudiants.urv.cat

Mariano Fons

PhD Candidate

University Rovira i Virgili, Tarragona, Spain
mariano.fons@estudiants.urv.cat

In the current era of communications and information technologies, automatic biometric personal recognition systems represent the state of the art in high-performance signal- and image-processing applications. In fact, it is not difficult to find in our daily lives systems requesting our personal authentication/identification before allowing us to use them; electronic tellers, computers, mobile phones and even cars require such authorization. Many end-user applications that demand better levels of security than PINs, passwords or ID cards use personal recognition algorithms based on biometric (physiological or behavioral) characteristics, usually delivering them as a kernel.

As a proof of concept, we developed an automatic fingerprint authentication system (AFAS) on the second smallest Xilinx® FPGA device in the Virtex®-4 LX family, making use of the Xilinx Early Access Partial Reconfiguration design flow and tools. The experimental results demonstrate it is possible to embed a full, highly demanding biometric recognition algorithm in such a small FPGA at an extremely low cost, processing it in real-time while preserving data accuracy and precision in its physical implementation by multiplexing functionality on the fly over a reduced set of resources placed in a partially reconfigurable region (PRR) of the device. These promising results, together with the proven maturity of the technology we used, encourage us to move this solution from research to industry, in an attempt to make partial reconfiguration (PR) available to the consumer world in the way of secure commercial products.

Basics of Biometrics

Computationally complex applications processed in real time, driven at low rates of power consumption and synthesized at low cost are unavoidable requirements today in the design and development of embedded systems, particularly when addressed to mass-production niches. In this context, dynamic partial self-reconfiguration of single-context FPGAs arises as a firm technological alternative, able to deliver a high functional density of resources to efficiently balance all those demands for time-, power- and cost-sensitive applications.

Software-defined radio, aerospace missions and cryptography are some of the known applications that exploit the benefits of dynamic partial reconfiguration of programmable logic devices today. In this context, our group is applying PR to an application space that hasn't traditionally leveraged it: biometrics. As security has become a major issue in today's digital information environment, especially for application fields like e-commerce, e-health, e-passports, e-banking or e-voting, among others, we believe the use of PR in biometrics holds great promise.

However, biometrics is complex. It requires stringent and computationally intensive image/signal processing in real time, along with a great deal of flexibility.

Finally, cost-effectiveness is probably the most important reason for biometrics to make use of partial reconfigurability. In aggressive markets like consumer electronics or automotive, vendors must market their systems at a competitive cost. Customers demand products with the highest level of security at the lowest possible price point.

The way to improve security and reliability is by increasing the computational power of the biometric recognition algorithm. This increment of computation usually involves a like increment in execution time and also in cost (resources). However, the cost is hardly affected in those scenarios where the design is based on dynamic-partial-reconfiguration technology. Using PR, designers can partition that new computation and schedule it as

Given that progress in biometrics technology is expected to continue in the future, biometric products already in the market will have to admit upgrades in the field just to avoid getting obsolete, and for this they require open system architectures.

In addition, personal recognition algorithms are in continuous evolution. As the research community expends major effort in this field, error rates like false acceptance and false rejection are improving. As a consequence, consumers are growing more confident about biometric systems, and acceptance is increasing. Given that progress in biometrics technology is expected to continue in the future, biometric products already in the market will have to admit upgrades in the field just to avoid getting obsolete, and for this they require open system architectures. In this regard, the flexible hardware found in run-time reconfigurable FPGA devices enables the versatility and scalability needed.

new processing stages added to the current sequential execution flow of the application. Thus, cost often can be held invariant to functional changes of the algorithm.

Designers can partition the biometric recognition algorithm into a series of mutually exclusive stages that are processed sequentially, where the outputs or results of one stage become the input data for the next. This sequential order means designers can multiplex hardware resources in time and customize them to execute a different task or role at each moment, increasing their functional density and thus keeping constant the total number of resources needed to process the entire algorithm. Moreover, the reconfiguration overhead is short enough

so as not to eclipse the benefits gained by hardware acceleration.

Furthermore, reconfiguring one set of resources on the fly will not interrupt the rest of the resources available in the FPGA. In this way, the resources that are not reconfigured continue to operate and guarantee the link with the exterior world for the entire life cycle of the application.

Our challenge in this work consisted in demonstrating that PR fits well in the development of complex personal recognition algorithms based on biometric characteristics, making use of a two-dimensional design abstraction level through which the functionality is managed not only in space but also in time. We describe this target step by step in the next sections.

Automatic Fingerprint Authentication System

Fingerprint verification is one of the most popular and reliable biometric techniques used in automatic personal recognition. Essentially, the technique splits the AFAS application into two processes or stages carried out at different times and in different conditions: enrollment and recognition.

Enrollment is the system configuration process through which the user gets registered. Generally, the user exposes his or her fingerprint to the system, which submits it to a set of computationally intensive image-processing phases aimed at extracting all relevant, permanent and distinctive information that will permit the system to unequivocally recognize the fingerprint's genuine owner. This set of characteristics becomes the user ID, which the system stores in its database. This process is normally conducted off-line, in a secure environment and under the guidance of expert staff.

Once the user is registered, the next time his or her fingerprint is exposed to the system in the recognition stage, the system will check to see if it corresponds with any

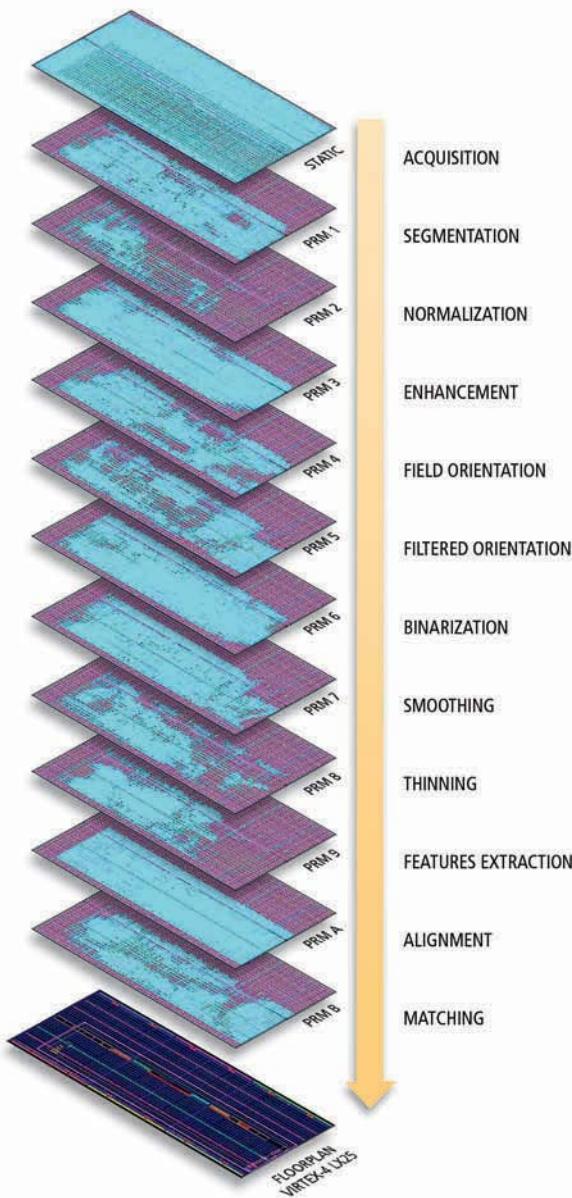


Figure 1 – Spatial partitioning and floorplanning of the AFAS take place in one static region and one reconfigurable region of the Virtex-4. Temporal partitioning of the application in sequential stages occurs in the reconfigurable region.

authorized member within the database. All the processing tasks performed in the enrollment are repeated now to again extract those distinctive characteristics from the live fingerprint sample. The system then compares these characteristics with the information stored as user templates in the database to conclude whether the live scan matches any of the registered templates. Recognition comes in two modalities depending on the size of the database: authentication, when a one-to-one (or one-to-few) matching is

processed; and identification, when the matching is one-to-many due to the fact that many users are registered in the system. Recognition is normally performed online in a less-secure environment and under real-time constraints.

Each of these stages is, in its turn, partitioned into a series of mutually exclusive tasks designed to extract from the fingerprint image such information as will distinguish one user from the others. With that object in view, the system carries out specific computations, such as image processing (2D convolution, morphologic operations), trigonometrics (\sin , \cos , \atan , $\sqrt{}$) [1] or statistics (average value, variance).

Thus, the biometric application is organized in a set of tasks that are processed following a sequential flow. A task cannot start unless the previous task has finished, since the output data of a given task is the input data for the next one in the chain. Moreover, most of these tasks are repeated in both enrollment and recognition stages.

Figure 1 enumerates the tasks that take place in the presented algorithm. The first task is the image acquisition. Depending on the size of the sensor, a system may acquire the whole image at one touch (complete image sensor) or in slices (sweeping sensor). In the second scenario—which was the case we used—an additional image reconstruction phase is necessary. The full fingerprint image gets composed by the set of consecutive and partially overlapped slices acquired [2].

Once we have the whole reconstructed image, the next task consists of segmenting it in the foreground (that is, the region of interest, based on the ridges and valleys of the fingertip skin) from the background. We perform this process by convolving the image, pixel by pixel, with directional filters made up of Sobel masks of kernel 5×5 . Afterwards, we normalize the image at a specific mean and variance.

Next, we enhance this normalized image through an isotropic filtering, which retrieves relevant image information from some potential regions of the captured

image initially lost or disturbed by the noise in the acquisition phase, making use of a kernel 13x13 [3]. Once this step has improved the quality of the image, the next task is to compute the field orientation map, which determines the dominant direction of ridges and valleys in each local region of the image foreground. The resultant field orientation is then submitted to a new filtering stage (kernel 5x5) to obtain a refined field orientation map.

Until this point, the image has been worked at 8-bit gray scale. Now, in the binarization process, Gabor directional filters of kernel 7x7 convolve the gray-scale image to improve the definition of the ridges and valleys and convert each of the gray-scale pixels to a 1-bit binary (black or white) dot. The image is then submitted to a new loop to smooth and redraw the shapes of the resultant ridges and valleys. Later, the thinning or skeletonization task converts the black-and-white image to one with black ridges one pixel wide. From that image it is not difficult to extract the fingerprint characteristics.

tic points or minutiae, that is, the ridge endings and bifurcations.

Finally, with the minutiae and the field orientation data already obtained, the fingerprint template and sample can be aligned. The first way of accomplishing this is through a brute-force algorithm that moves one image over the other—taking into consideration both translation and rotation movements as well as some admissible tolerances due to the image distortion coming from the skin elasticity in the acquisition phase—to find the best alignment between them [4]. The next step is to match the sample and template to obtain a level of similarity between them, which the automatic system will use to decide if both images correspond to the same person [5].

All this processing, illustrated in Figure 4, is performed on fingerprint images of 500-dpi resolution, 8-bit gray scale and up to 280 x 512 pixels, acquired through sweeping technology via the thermal fingerprint sensor FingerChip from Atmel Corp. and computed in the Xilinx Virtex-4 XC4VLX25 FPGA device.

System Architecture

The Virtex-4 FPGA device becomes the computational unit of the AFAS platform. Flash memory plays the role of system database, storing nonvolatile information like bitstreams as well as specific application data such as user fingerprint templates or configuration settings of the biometric algorithm. The system also uses DDR-SDRAM memory to temporarily store intermediate data or images obtained in each processing stage. We implemented a serial communication link, in our case an RS-232 transceiver connected to a UART controller—the latter synthesized in the resources of the FPGA—to use for debugging purposes, just to transfer the resulting image of each stage to a PC in order to visualize the fingerprint images or results of each step. Finally, a sweeping fingerprint sensor, addressed to capture the biometric characteristic of the user, acts as input of the recognition algorithm, as depicted in Figure 2.

Regarding the computation unit, the FPGA is detached in two regions, as shown in Figure 3: a static region occupied by a

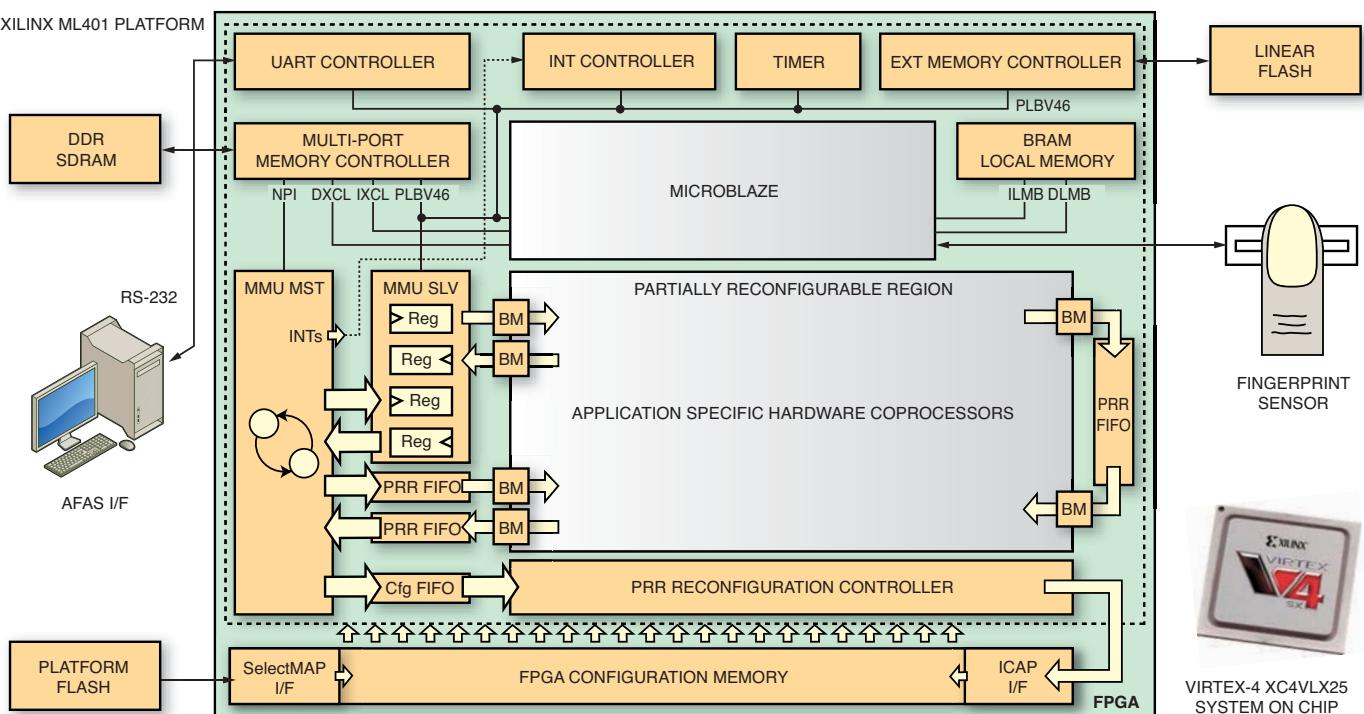


Figure 2 – System architecture and functional components breakdown of the suggested AFAS.

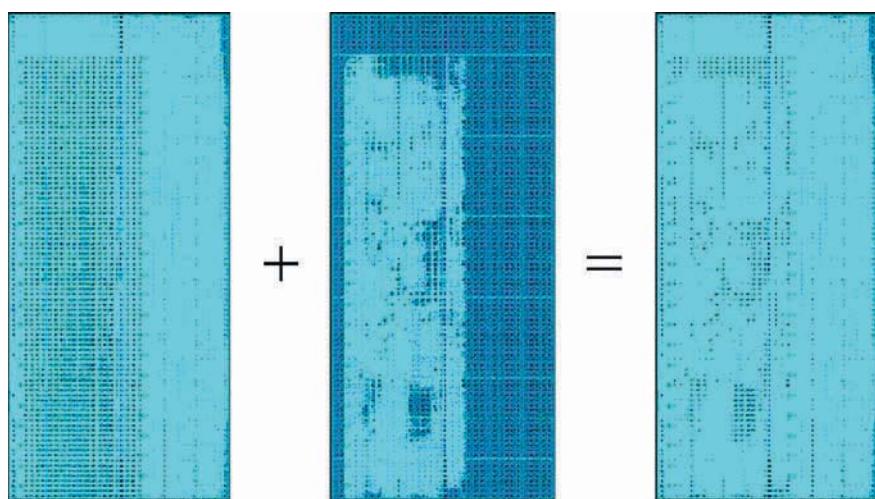


Figure 3 – Composition of the full bitstream placed in the FPGA at a given time along the application execution flow. The static region (far left) and one of the dynamically reconfigurable biometric coprocessors instantiated and shared in time in the PRR (center) make up the full bitstream (far right).

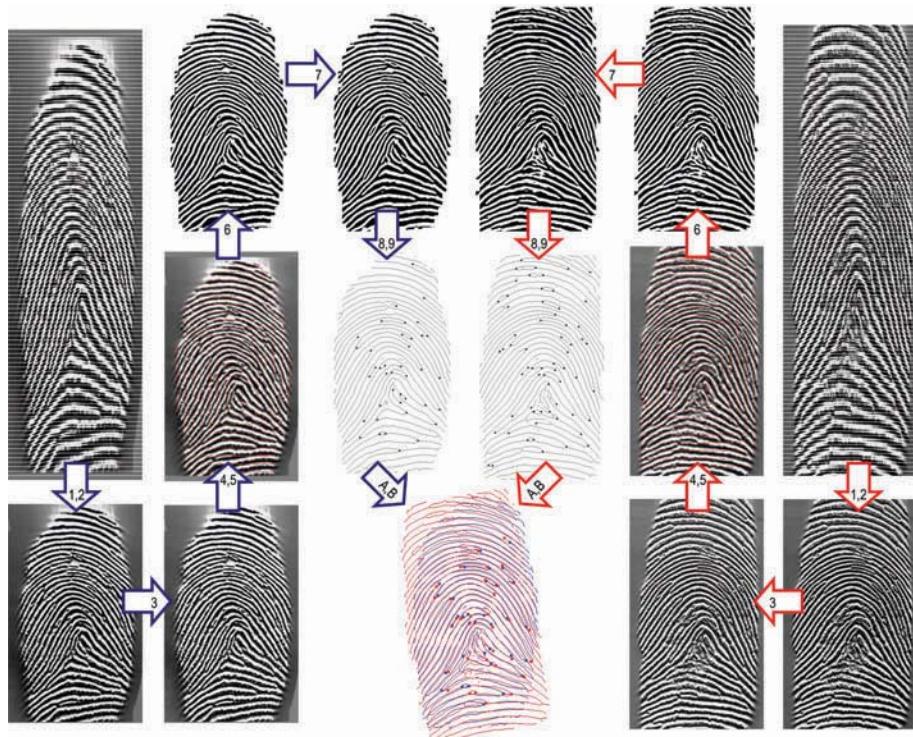


Figure 4 – Resultant images obtained in each of the sequential stages of the biometric recognition algorithm (fingerprint template processing on the left and fingerprint sample processing on the right).

whole multiprocessor CoreConnect bus system; and a partially reconfigurable region that is used to place—on demand and multiplexed in time as long as the processing advances—the custom biometric coprocessors or IP responsible for the different

sequential tasks of the recognition algorithm. The multiprocessor CoreConnect bus system mainly comprises a MicroBlaze™ processor and other standard peripherals along with a custom reconfiguration controller, this one linked to the ICAP port.

All the processing tasks are enumerated from 0 (static) to B in Figure 1, according to sequential execution order. Custom hardware coprocessors implement all the tasks in the PRR, with the exception of the fingerprint acquisition process, which the MicroBlaze performs in software.

The reason behind this specific hardware/software partitioning is that the sweeping sensor needs an integration time of 5 milliseconds to acquire consecutive slices. That's enough time for it to perform the image reconstruction on the fly directly in software under MicroBlaze control. Therefore, it is not necessary to implement this image reconstruction with a custom hardware coprocessor.

The image acquisition consists of capturing 100 slices at a rate of 5 ms per slice, with each slice consisting of 280 x 8 pixels. Software handles the reconstruction in real time by detecting the overlapping of rows of pixels between each two consecutive image slices.

We implemented the rest of the tasks, however, as custom hardware coprocessors in the PRR of the FPGA simply because of real-time constraints. Once the processing of each particular task is finished, the reconfiguration controller, located on the static region of the device and instructed by the MicroBlaze processor, replaces the coprocessor currently instantiated in the PRR by the one corresponding to the next stage of the biometric algorithm. The reconfiguration controller does this job by simply downloading the new partial bitstream into the PRR and transferring this data directly from DDR-SDRAM to the internal FPGA configuration memory via the ICAP interface.

It is important to note that we used a standard interface based on FIFO memories and flip-flop registers between the static and the reconfigurable regions. This allows us to develop standard biometric coprocessors or IP placed in the PRR that are totally independent of the multiprocessor bus the system uses, be it AMBA®, CoreConnect, Wishbone or some other, as depicted in Figure 2. This point is fundamental in order to guarantee standardization and portability of the biometric algorithm to different platforms.

Reconfiguration Controller

The design of an efficient reconfiguration controller is key to success in the deployment of PR systems oriented to single-context FPGAs. Although the nonreconfigured area of the FPGA remains in operation while the PRR is reconfigured, the PRR resources are not operative at that time, so it is desirable to speed up the reconfiguration process as much as possible so as to minimize this overhead. The reconfiguration time depends on three factors: data bus width, reconfiguration frequency and bitstream size—the first two relate to interface aspects, while the last is closely tied to the PRR size and the design complexity of the partially reconfigurable module (PRM) located there.

Our work implements a reconfiguration controller that is able to transfer partial bitstreams from external memory to the FPGA's on-chip configuration memory at run-time with a high bandwidth. It is possible to reach the maximum reconfiguration bandwidth of Virtex-4 technology with no constraints in the partial bitstream size and with the external memory as a shared resource that different processors can access concurrently from the system buses.

In the system initialization, the partial bitstreams to be downloaded at run-time into the FPGA configuration memory move from the external nonvolatile memory (flash) to the external DDR-SDRAM. This memory is connected to a multiport memory controller (MPMC), so it becomes a shared resource accessible by any master or slave processor in the system. Different buses can be connected to the MPMC, for instance the CoreConnect PLBv46 bus, used as general-purpose system bus, or even the Xilinx CacheLink (XCL) bus, oriented to fast instruction and data caches of the CPU. The system CPU (MicroBlaze) is, in fact, connected to these two buses.

Our reconfiguration solution, however, is based on a new bus, the Native Port Interface (NPI), which is specifically designed to establish a fast link between the external DDR-SDRAM repository and the ICAP primitive. As part of our reconfiguration controller, we have designed a master

memory-management unit (MMU) that handles the NPI protocol. The link between external DDR-SDRAM (partial bitstreams) and the ICAP primitive goes through an internal FIFO memory. In this way, we can implement two different made-to-measure interfaces, with independent data bus size and speed—one coupled to the NPI protocol and the other to the ICAP protocol.

The write port of the FIFO is connected to the NPI and uses a 64-bit data bus. The read port of the FIFO, joined to the ICAP, uses a data width of 32 bits—the maximum data width of ICAP in Virtex-4 devices. Regarding frequency, both read and write ports of the FIFO (on the NPI and ICAP sides) run at 100 MHz, although the NPI side could work at a higher rate if necessary. To keep the transfer latency to a minimum, the master MMU performs the bitstream reconfiguration in 64-word (32-bit) burst transfers to the internal FIFO. This is the maximum length of burst accepted, so all the partial

bitstream transactions are done at the lowest burst latency. On the other side, the reconfiguration controller reads the stored FIFO data and transfers it in 32-bit format to the ICAP primitive, as long as the FIFO is not empty. The reconfiguration controller (just the master MMU) is handling the direct memory access (DMA) to huge DDR-SDRAM memory. We set up this part with several configuration registers implemented in another custom slave MMU controller connected to the PLBv46 bus and directly managed by the CPU.

In this way, the CPU only needs to do two things: configure the initial address and size of the partial bitstream to be downloaded in the PRR, and then give the go-ahead command to the MMU master to start the reconfiguration process. At that point, the MMU master starts the bitstream DMA transfer to the internal FIFO and from this to the ICAP primitive. Once the transfer is finished, the reconfiguration controller notifies the CPU.

AUTOMATIC FINGERPRINT AUTHENTICATION SYSTEM	PROCESSING TIME (ms)		
	PC PLATFORM	EMBEDDED SYSTEM	
		SW Core2Duo 1.83 GHz	SW MicroBlaze Virtex-4 100 MHz
Acquisition	500.000	500.000	500.000
Segmentation	2.810	232.046	0.672
Normalization	0.470	33.087	1.691
Enhancement	7.030	512.171	3.608
Field Orientation	2.500	337.419	1.694
Filtered Orientation	0.620	22.178	1.465
Binarization	15.940	774.750	3.572
Smoothing	14.220	287.507	1.492
Thinning	1.410	417.350	1.794
Features Extraction	0.630	32.497	8.549
Alignment	3224.530	139935.838	158.716
Matching	4.220	108.608	21.772
TOTAL	3774.380	143193.451	705.025

Table 1 – Processing time breakdown (in milliseconds) of the different tasks executed in different AFAS platforms: a software-only approach on a personal computer platform, embedded-software approach on a Xilinx Virtex-4 XC4VLX25 FPGA and HW/SW co-design based on partial reconfiguration.

AUTOMATIC FINGERPRINT AUTHENTICATION SYSTEM	PERFORMANCE PR-HW & SW APPROACH					
	Time (ms)		Hardware Resources			
	RECONF. (100 MHz)	PROCESS. (50/100 MHz)	SLICE FLIP-FLOPS	4-INPUT LUTs	18-KBIT RAMB16	DSP48 BLOCKS
Application Flow (static)	—	—	7005	8888	41	4
Acquisition	—	500.000	—	—	—	—
Segmentation	—	0.672	4978	4612	8	20
Normalization	0.841	0.850	371	334	0	8
Enhancement	1.045	2.563	5275	5831	5	28
Field Orientation	1.025	0.669	3339	3166	5	8
Filtered Orientation	1.046	0.419	2857	2983	7	0
Binarization	1.107	2.465	5462	4166	17	29
Smoothing	1.045	0.447	4892	3265	8	0
Thinning	0.974	0.820	1013	2821	13	0
Features Extraction	0.943	7.606	487	3379	3	0
Alignment	1.045	157.671	2632	8943	21	0
Matching	1.035	20.737	642	4379	14	5
TOTAL	10.106	694.919	38953	52767	142	102

Table 2 – Time and resources breakdown of the different tasks executed by the AFAS driven by partial reconfiguration technology on a Virtex-4 XC4VLX25 FPGA composed of 21,504 flip-flops, 21,504 four-input LUTs, 72 RAMB16 blocks and 48 DSP48 blocks.

As a result, we achieve the transfer of the partial bitstream at maximum throughput even if the DDR-SDRAM is accessed by the CPU via XCL or PLBv46 buses at the same time. That's because, in the end, the CPU runs the program flow in internal BRAM cache, freeing the access to the external DDR-SDRAM to the reconfiguration controller. It is important to note here that this DDR-SDRAM memory where both partial bitstreams and software application are allocated is not a dedicated resource but a shared resource. Even so, this scheme significantly improves upon other existing reconfiguration controller approaches, since it reaches the maximum reconfiguration throughput of Virtex-4 technology (transfer of the partial bitstream to the ICAP through a 32-bit data bus at a rate of 100 MHz, or 3.2 Gbps).

Experimental Results

The embedded automatic fingerprint authentication system described here is essentially a high-performance image-

processing application, since it exhibits a great deal of parallelism and demands a real-time authentication response. From an ergonomic standpoint, that could mean, for instance, not to exceed 2 or 3 seconds in the authentication process of any user.

The design flow entails several development loops. Initially, we fully developed the algorithm in software in MATLAB® on a PC platform. Afterward, we ported this software code to embedded software in the C programming language and executed it first in the same PC, just to confirm that we would obtain the same results, and then on an embedded microprocessor like the MicroBlaze synthesized in our FPGA device.

In this approach, the Virtex-4 device implements a software-only solution based on MicroBlaze, without any custom hardware coprocessor in use and without reaching real-time performance. To improve the time, and based on the resultant tasks profiling we obtained, our next step consisted

of switching to a HW/SW co-design solution by introducing the PRR, where we located the different custom biometric coprocessors. At this point, we have fully developed the system in both the C programming language and VHDL hardware description language.

We have conducted some recognition tests with 8-bit gray-scale fingerprint images of 268 x 460 pixels. We deployed the same tests in two platforms: in our PR system based on Virtex-4 and also in a personal computer based on an Intel Core 2 Duo T5600 processor running at 1.83 GHz. We then ran the same algorithm, either implemented purely in software or by combining software with flexible hardware, just to compare the performance in both enrollment and recognition stages.

We obtained identical recognition results in both platforms, as expected. However, the processing time spent in each case differed dramatically. Table 1 shows the time needed when the algorithm is deployed on different platforms and architectures: a software approach on the Intel Core 2 Duo PC platform; embedded-software approach on an ML401 platform powered by a Virtex-4 XC4VLX25 FPGA based on a MicroBlaze processor at 100 MHz; and HW/SW co-design approach on an identical ML401 platform equipped with dedicated biometric coprocessors running at either 50 or 100 MHz, instantiated in the PRR and reconfigured on demand.

Without considering the acquisition task, which is fixed at 500 ms due to the sweeping-sensor restrictions (100 slices captured with an integration time of 5 ms and image reconstructed from them on the fly), the PR approach reduces latency due to the rest of the processing tasks to 205 ms. That compares with latency of 3,274 ms in the pure-software approach on the PC, which means a speedup of 16x in favor of the PR solution.

Thus, Table 1 makes it evident that real-time authentication is feasible with HW/SW co-design that exploits parallelism and pipeline techniques, along with PR technology, thanks to its low reconfiguration latency. Furthermore, in the PR approach, each

task can run at a specific frequency; this frequency is established each time we reconfigure the PRR to download a new module with new specific characteristics. Our approach ran all the tasks performed in hardware at either 50 or 100 MHz.

Furthermore, the reconfiguration process was always performed at 100 MHz, transferring 32-bit words per clock, a fact that guarantees the lowest reconfiguration latency on the Virtex-4. Each reconfiguration process took between 0.8 ms (for example, normalization) and 1.1 ms (e.g., binarization), depending on the bitstream complexity of each PRR hardware context. This reconfiguration time is negligible in comparison to the total processing time of the biometric recognition application, as depicted in Table 2.

But we not only addressed time in this PR design. We also carefully considered cost-effectiveness by means of the time-sharing of the resources involved. The XC4VLX25 FPGA device contains 21,504 slice flip-flops, 21,504 four-input LUTs, 72 18-kbit RAMB16 blocks and 48 DSP48 blocks. Regarding the partitioning of resources in both static and reconfigurable regions, the reconfigurable region takes 11,264 slice flip-flops, 11,264 four-input LUTs, 22 18-kbit RAMB16 blocks and 44 DSP48 blocks, while the rest of the resources of the device keep static for the entire life cycle of the application.

The PRR is in charge of the execution of up to 11 different sequential tasks of the recognition algorithm. As shown in Table 2, the same application synthesized on a fully static design would not fit fully on the XC4VLX25 FPGA; therefore, that would typically force designers to choose a bigger and more expensive device with the proper amount of resources. However, using PR eliminates this issue. Table 2 definitely demonstrates that automatic personal authentication can be performed at extremely low cost today with the reuse of logic resources thanks to PR technology.

The set of tools we used, available in the Xilinx Early Access Partial Reconfiguration Tools Lounge, are ISE® 9.02.04i together with the PR_12 patch, EDK 9.02.02i, and PlanAhead™ 9.2.7. Finally, we validated

the system on real fingerprint images acquired by the system as well as other fingerprint images that exist in public databases based on the same fingerprint sweeping sensor (Fingerprint Verification Competition databases).

Now that we have successfully completed the proof of concept, we plan to port this prototype to the coming next generation of low-end Xilinx 28-nanometer FPGA devices provided with PR capability in the Artix™-7 family, and the new PR design flow based on partitions that Xilinx recently released. Our goal is to design a system able to embed high-performance and real biometric security in any consumer electronics product at the lowest possible cost.

The time for run-time reconfigurable computing in biometric applications is definitely now. For further information about this project, you can contact the authors at {francisco.fons, mariano.fons}@estudants.urv.cat.

References

- [1] F. Fons et al., "Trigonometric Computing Embedded in a Dynamically Reconfigurable CORDIC System-on-Chip," *Reconfigurable Computing: Architectures and Applications, Lecture Notes in Computer Science, Vol. 3985, pp. 122-127, ISSN 0302-9743, Springer, 2006.*
- [2] M. Fons et al., "Hardware-Software Co-design of an Automatic Fingerprint Acquisition System," *IEEE International Symposium on Industrial Electronics, ISIE 2005 Conference Proceedings, pp. 1123-1128, Dubrovnik, Croatia, June 2005.*
- [3] F. Fons et al., "Approaching Fingerprint Image Enhancement through Reconfigurable Hardware Accelerators," *IEEE International Symposium on Intelligent Signal Processing, WISP 2007 Conference Proceedings, pp. 457-462, Alcalá de Henares, Spain, October 2007.*
- [4] M. Fons et al., "Design of a Hardware Accelerator for Fingerprint Alignment," *IEEE International Conference on Field Programmable Logic and Applications, FPL 2007 Conference Proceedings, pp. 485-488, Amsterdam, The Netherlands, August 2007.*
- [5] M. Fons et al., "Hardware-Software Co-design of a Fingerprint Matcher on Card," *IEEE International Conference on Electro/Information Technology, EIT 2006 Conference Proceedings, East Lansing, Michigan, USA, May 2006.*

Imagine.Combine.Implement.Today

IP Cores for Your Low-Volume FPGA Products

you have an amazing **idea?**
looking for the perfect graphics controller?
choose & configure IPs
implement your FPGA w/o coding
logicBRICKS™ make your ideas come true

Low Cost IP Cores:

- * Graphics and video solutions
- * Fully compatible with Xilinx® tools
- * Simple GUI parameterization
- * Evaluation prior to purchase
- * Support from skilled FPGA Engineers
- * Available reference hardware platforms
- * Support all the latest Xilinx FPGA families

Get Online, Plug and Play!

Prices start at €500

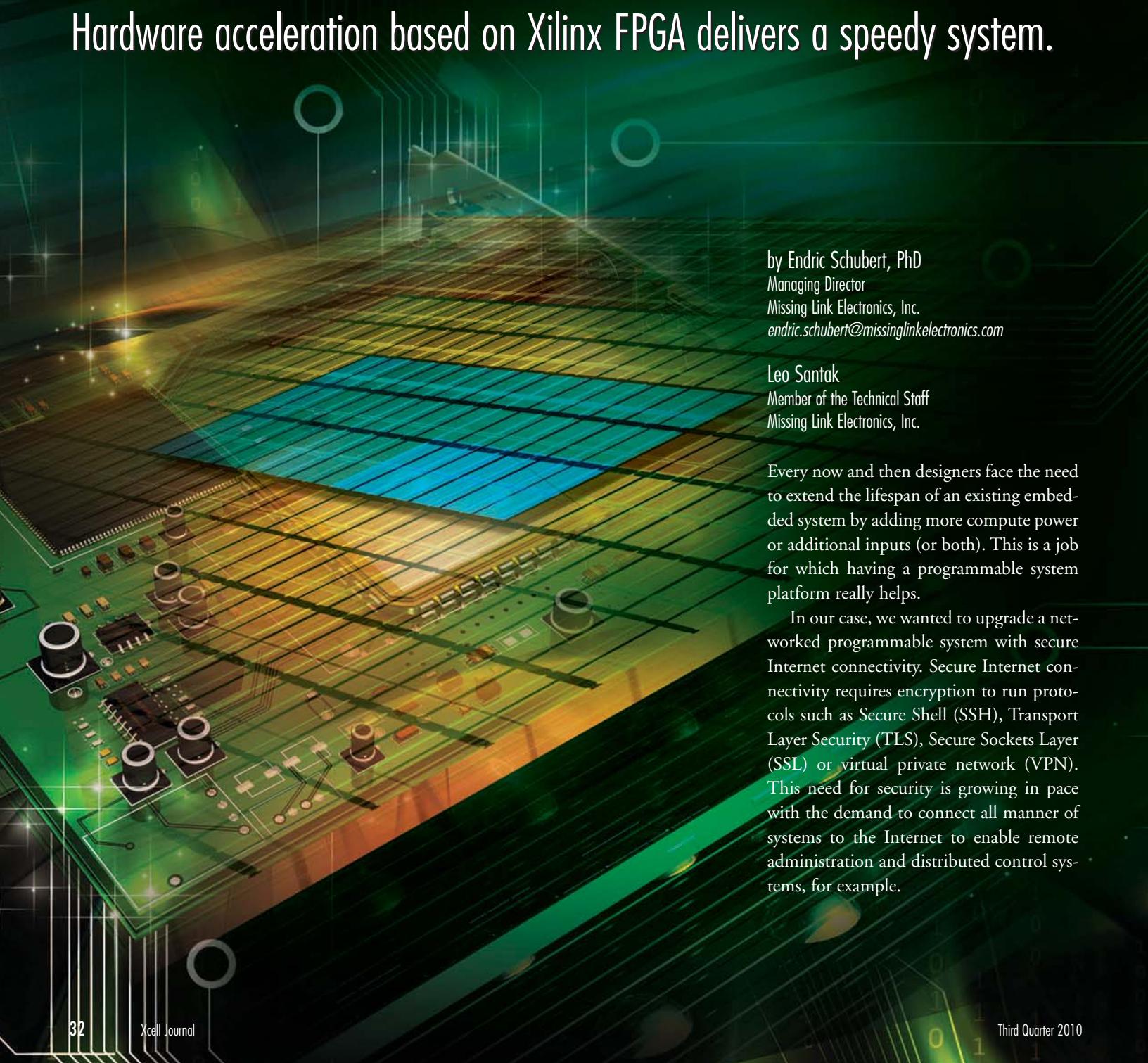
Visit our web shop today:
www.logicbricks.com

logicBRICKS™
Designed by XYLON

logicBRICKS is a registered trademark of Xylon d.o.o.
All other trademarks are the property of their respective owners

Building a Better Crypto Engine the Programmable Way

Hardware acceleration based on Xilinx FPGA delivers a speedy system.



by Endric Schubert, PhD
Managing Director
Missing Link Electronics, Inc.
endric.schubert@missinglinkelectronics.com

Leo Santak
Member of the Technical Staff
Missing Link Electronics, Inc.

Every now and then designers face the need to extend the lifespan of an existing embedded system by adding more compute power or additional inputs (or both). This is a job for which having a programmable system platform really helps.

In our case, we wanted to upgrade a networked programmable system with secure Internet connectivity. Secure Internet connectivity requires encryption to run protocols such as Secure Shell (SSH), Transport Layer Security (TLS), Secure Sockets Layer (SSL) or virtual private network (VPN). This need for security is growing in pace with the demand to connect all manner of systems to the Internet to enable remote administration and distributed control systems, for example.

Because this field is still evolving and standards are not yet set, costs are dominated by nonrecurring-engineering fees. Therefore, FPGA technology offers the best value for implementations.

Our system was built on top of the Missing Link Electronics (MLE) "Soft" Hardware Platform, where the flexible I/Os of the FPGA enable connection to a wide range of sensors and actuators. This platform uses the programmable logic to implement a system-on-chip with either the MicroBlaze™ CPU or the PowerPC® CPU at its heart. The CPU runs the MLE Linux software stack for the operating system and the user-space application software. With the MicroBlaze or the PowerPC as the main CPU, the system was obviously not suitable for delivering the required compute performance when running embedded Linux plus strong encryption on top. And changing the physical hardware was not an option.

Instead, we utilized the power of programmable systems to migrate computations from the software domain to the hardware side for system acceleration.

Coprocessing Hardware

A programmable system is basically a combination of one or more CPUs—running an operating system and application software—plus an FPGA. The FPGA is there as a flexible interface “adapter” and as coprocessing hardware. You can make programmable systems from separate companion chips or integrate everything into one single device. Depending on how the FPGA device and the CPU are communicating with each other, you have different options in adjusting the system for performance and functionality.

One possibility is to add a peer processor, which synchronizes with the CPU via memory-mapped status and control registers. Because running all communication over the same system bus may quickly suffocate performance, you really want to separate the data stream of the CPU from the peer processor. This is easy to do by using system-on-chip components such as the Xilinx Central DMA or the Multiport Memory Controller (MPMC).

Alternatively, you can add a coprocessor, in which case you effectively extend the instruction set of the CPU by adding custom instructions (also called compiler-known functions). This is, for example, the case for floating-point units, and the Xilinx technology of Fabric Coprocessor Modules

(FCM) readily supports that. The advantage here is to free up the memory-to-system bus by using a dedicated communication channel between the CPU and the coprocessor. For the PowerPC this is the Auxiliary Processing Unit (APU) and for MicroBlaze, the Fast Simplex Link (FSL).

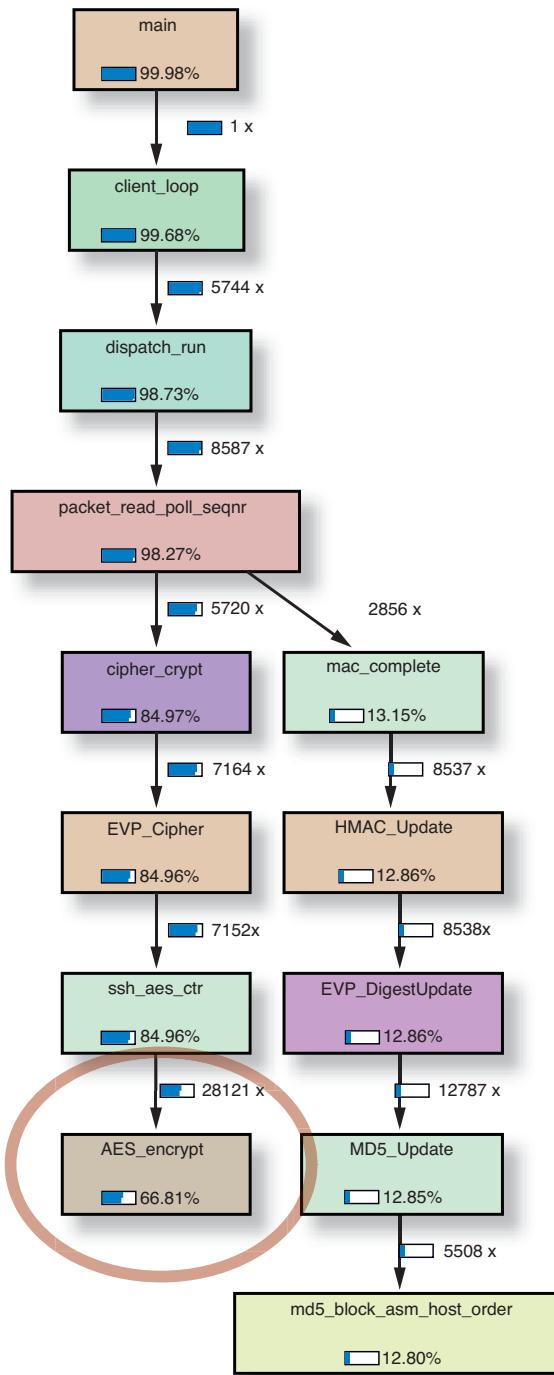


Figure 1 – In an SCP transfer using the Valgrind tool, the AES encryption occupies two-thirds of the computations.

In encryption and decryption, most of the operations are performed on either the rows or the columns, leaving four operations that can be calculated in parallel—a job well suited for hardware.

AES: the Gold Standard

But how do you really accelerate encryption without a major system redesign?

For encryption, the Advanced Encryption Standard (AES) is really the de facto standard. With AES encryption, the computations are irreducible by definition, bringing an embedded system

array of bytes. We used a 128-bit block size, which withstands all known attacks and is even supposed to be more secure than the 192-bit and 256-bit versions.

With 128-bit AES, it takes 12 rounds, each with several steps, to perform the encryption and decryption. The first task is to compute the round keys from the secret

erously fits into the common Xilinx BRAM primitives. Several S-box instances speed up the IP core and supply the core in place with the data needed, without waiting on long-lasting bus accesses to main memory. The decryption occurs in a similar fashion, using the same secret key, but in the opposite direction and with a different S-box.

12 Times Faster

In encryption and decryption, most of the operations are performed on either the rows or the columns, leaving four operations that can be calculated in parallel—a job well suited for hardware. Thus, various hardware implementations of AES are available from different sources. To accelerate our system, we took an AES core from the great and fast-growing OpenCores.org repository (http://opencores.org/project,avs_aes). We removed the original bus interface, which was targeted for another FPGA architecture, and added an interface for the APU to connect the AES core as an FCM coprocessor to a PowerPC. We used a total of eight so-called UDI commands to transfer data between the PowerPC and the AES FCM.

The result of that work was very satisfying (see Figure 2). The hardware-accelerated system (green bar, center) ran faster than a standalone PowerPC or an Atom processor. It took 17.8 microseconds to encrypt one single block using a standalone PowerPC running at 300 MHz, but only 1.5 µs to do this with an AES FCM running at 150 MHz. For those who are tempted to just switch to a faster CPU for a speedup, our hardware-accelerated speed of 1.5 µs outperformed a pure-software implementation on an Intel Atom 1.6-GHz CPU, which took 2.7 µs.

These results demonstrate the outstanding potential of hardware acceleration using FPGA technology. For details of the analysis and exemplary code, contact our applications team at <http://www.missinglinkelectronics.com>.

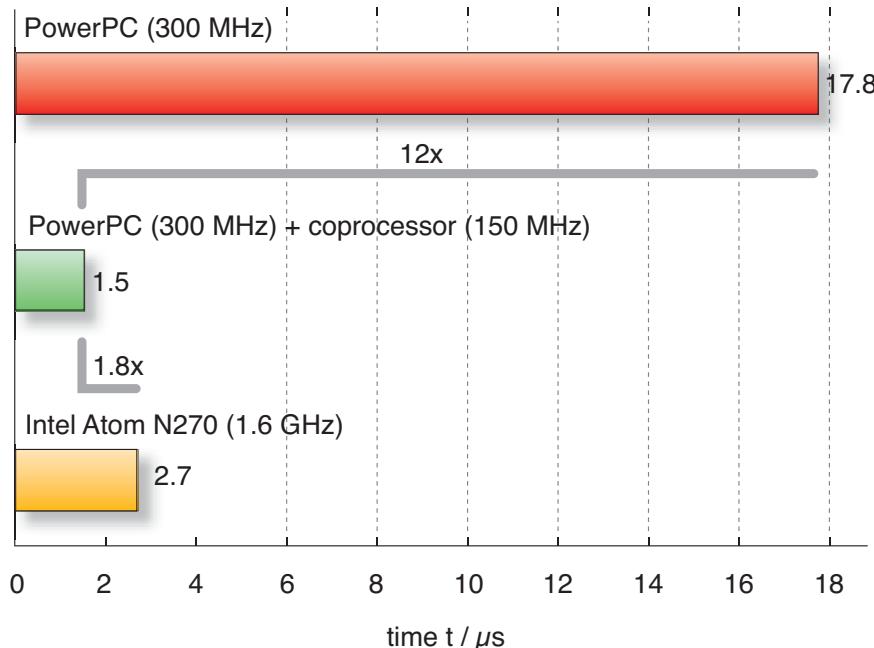


Figure 2 – The hardware-accelerated system (green bar, center) ran faster than a standalone PowerPC or an Atom processor.

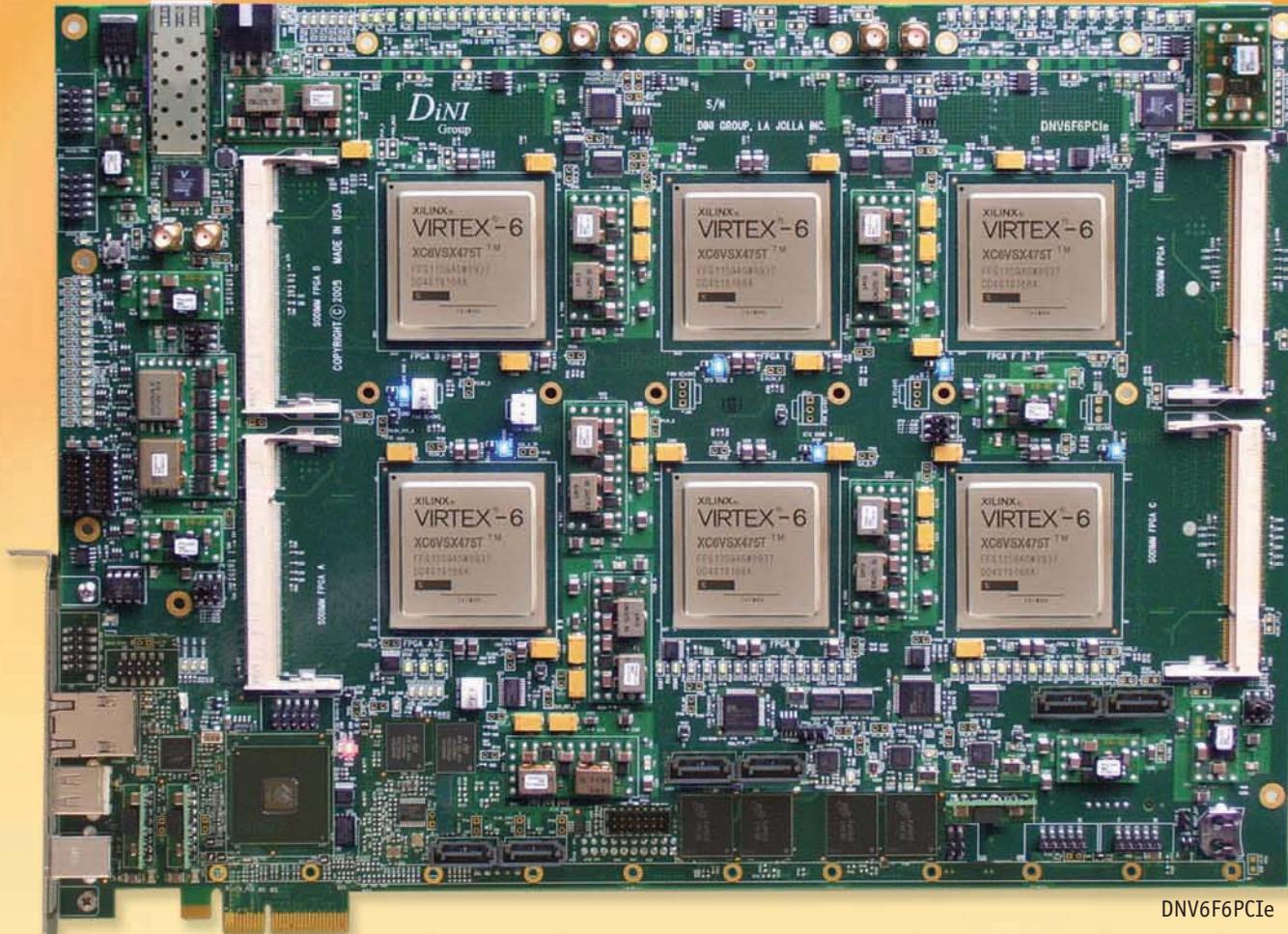
quickly to its performance limits. This is clearly illustrated in Figure 1, which shows the profiling results of a file transfer with SCP (SSH session) using the Valgrind analysis tool. In this case, the AES encryption takes up two-thirds of the computations.

AES-128, with a key and block length of 128 bits, utilizes many concurrent 8-byte operations. AES is a block cipher and operates on fixed block sizes organized as a 4 x 4

key by means of the so-called key expansion process. In every round, the plain text is bit-wise XOR-ed with its own round key. Then sub-byte, row-shifting and column-mixing operations follow, and the round key gets XOR-ed once again.

The final round slightly differs, omitting some steps. The encryption process performs substitution using a so-called S-box, which provides nonlinearity. We can arrange it in a 16 x 16 x 8-bit matrix so that it gen-

NEW Virtex-6x6 — More Power and Speed for ASIC Prototyping and High Performance Computing



Six powerful Virtex®-6 FPGAs, up to 24 Million ASIC gates, clock speeds to 710 Mhz: this new board races ahead of last generation solutions. The Dini Group has implemented new Xilinx V6 technology in an easy to use PCIe hosted or stand alone board that features:

- 4 DDR3 SODIMMs, up to 4GB per socket
- Hosted in a 4-lane PCIe, Gen 1 slot
- 4 Serial-ATA ports for high speed data transfer
- Easy configuration via PCIe, USB, or GbE
- Three independent low-skew global clock networks

The higher gate count FPGAs, with 700 MHz LVDS chip to chip interconnects, provide easier logic partitioning. The on-board Marvell Dual Sheeva processor provides multiple high speed interfaces optimized for data throughput. Both CPUs are capable of 2 GFLOPS and can be dedicated to customer applications.

Order this board stuffed with 6 SX475Ts—that's 12,096 multipliers and more than 21 million ASIC logic gates—an ideal platform for your DSP based algorithmic acceleration and HPC applications.

Don't spin your wheels with last year's FPGAs, call Dini Group today and run your bigger designs even faster!

D
DNI
Group

LTE Simulator Relies on Xilinx Virtex-5 FPGAs

Powerful programmable logic platform enables Prisma Engineering to provide reconfigurable radio test equipment for all cellular networks.

by Catello Antonio De Rosa
Senior Design Engineer and FPGA Specialist
Prisma Engineering
cateloldi@prisma-eng.com

LTE (Long Term Evolution), the new 3GPP standard for broadband mobility, disrupts the existing paradigms of cellular networks. In addition to high-spectral-efficiency radio techniques, LTE boasts a very simplified architecture in comparison to the prior-generation UMTS and GSM standards. Evolved Node-B's, the radio-access part of the LTE system, are the edge between the radio and all-Internet Protocol core networks. This architecture makes it impossible to monitor and test the equivalent of intermediate links in UMTS. An effective testing of LTE network elements must involve the radio interface.

This is exactly the challenge addressed by our design team in Prisma Engineering's Line Server Unit (LSU) UeSIM LTE. The simulator is a complete solution for all LTE testing needs, allowing network equipment designers to stress and monitor both the air interface and the core network. This single hardware platform can simulate up to 1,024 pieces of user equipment per sector. Load-and-stress and functional testing over the radio interface encompass complete LTE protocol stacks and their applications. A radio front end handles bandwidths of 5, 10, 15 and 20 MHz in a native multiple-input, multiple-output (MIMO) design.

Three Xilinx® Virtex®-5 FPGAs (XC5VSX50T) reside at the heart of this advanced simulator, enabling a high level of software-defined radio reconfiguration. Our team at Prisma Engineering, which is headquartered in Milan, Italy, quickly realized we needed a powerful and reprogrammable architecture in order to gain the flexibility to address a multitude of radio access standards using the same board. Our main goal was, as our CEO, Enrico Bendinelli, put it, “to create the industry’s most flexible and easy-to-use management software.”

Two user test tools—the LTE Test Manager (primarily for LTE equipment vendors) and the Quick GUI (primarily for LTE network operators)—are available. The Quick GUI provides pass/fail-type testing scenarios while the Test Manager allows for more complex analysis.

LSU UeSIM LTE Architecture

The LSU UeSIM LTE Simulator is based on a CompactPCI standard architecture comprising a protocol-processing unit (PPU) board, a software-defined radio (SDR) board and two radio modules for MIMO operations.

Based on Intel technology, the PPU board, which is the main processor card, is able to manage multiple SDR boards in order to improve the load-and-stress capacity. The software-defined radio board is designed to extend the operation of our previous LSU systems on the radio interfaces. The CompactPCI radio mezzanine cards provide radio-frequency (RF) transmission/reception capability at different radio standard bandwidths: GSM (850 and 900 MHz; 1.8 and 1.9 GHz), LTE (700 MHz, 2.1, 2.3, 2.5 and 2.6 GHz) and WiMAX (2.4, 3.5 and 5 GHz).

SDR Card Architecture

The SDR card is a high-performance platform integrated within the LSU hardware/software environment to extend the connectivity of the system with the baseband (CPRI/OBSAI), the radio interface or both. The card supports different wireless standards such as GSM/EDGE, UMTS, HSPA, WiMAX and LTE using different external radio modules operating in the specific frequency bands.

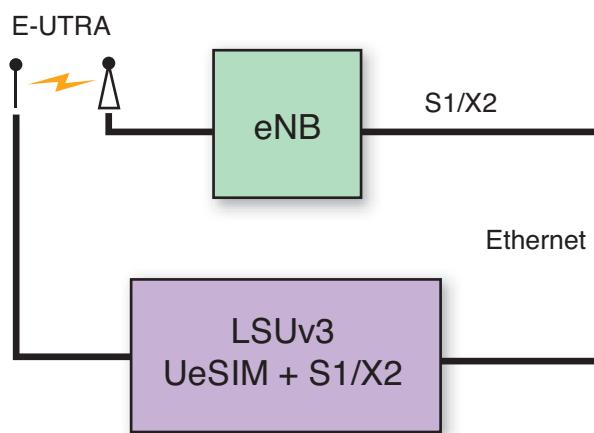


Figure 1 – In an LTE test scenario, the simulator either replaces a radio sector or provides a test interface for the core network.

We completed the design of the Xilinx-based SDR card with three 1-GHz Texas Instruments DSPs (we chose the TMS320C6455 device) and two pairs of Analog Devices analog-to-digital (AD9640) and digital-to-analog (AD9779) converters. The clocking network, based on an Analog Devices AD9549, provides a very high, flexible timing base for the conversion and digital signal-processing devices (FPGAs, DSPs).

LTE Elaboration Datapath

Prisma divided the LTE elaboration datapath into two sections: the radio front end, which we implemented in an FPGA, and the physical-resource allocation and data- and control-channel termination, which we implemented in a DSP.

In the uplink direction, one DSP handles MAC-layer to physical-layer exchange and some functions of the physical layer. It

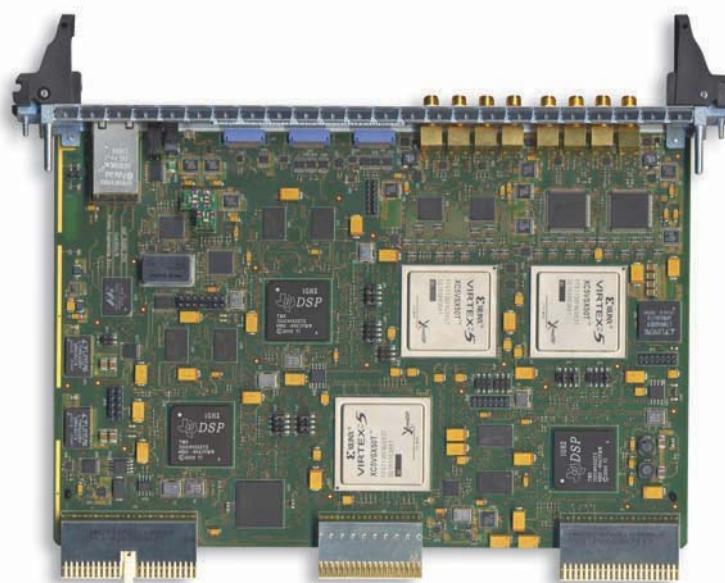


Figure 2 – Xilinx Virtex-5 FPGAs reside on the LSU’s software-defined radio card, along with TI DSPs.

provides coding, interleaving, scrambling, symbol mapping and subcarrier allocation with reference signal (pilots), source data and control channels. Discrete Fourier transform (DFT) functions transform data from different terminals according to the SC-FDMA standard. The system transfers every OFDM symbol to the uplink FPGA using an EMIF interface.

This FPGA changes the data rate from 125 MHz (DSP EMIF interface clock) to 245.76 MHz (the FPGA elaboration rate). Then the FPGA performs a number of other operations: a 2,048-point inverse fast Fourier transform, a cyclic prefix insertion, a PRACH data channel insertion, a half-shift function that translates the OFDM symbol spectrum at 7.5 kHz, a shaping and interpolation filtering and an intermediate-frequency (IF) conversion at 24 MHz. The device sends IF data to the DAC at a clock rate of 122.88 MHz. The radio card, meanwhile, converts the analog signal to RF and sends it to the transmitter amplifier.

In the downlink direction, after the LNA amplification, programmable-gain and conversion stages, the radio card will send IF received data to the SDR card (140 MHz). The ADCs subsample the analog data at 122.88 MHz and the FPGA handles the final 17.12-MHz frequency conversion to baseband. This data can be related either to two single-input, single-output channels or to one MIMO channel.

The IF data enters into the downlink FPGA, which converts it to baseband and then filters it. Polyphase decimation filters implement Nyquist filtering, spectrum image rejection and data-rate reduction at a symbol rate of 30.72 MHz, even though the chip rate remains at 245.76 MHz.

The FPGA incoming data flow looks like a stream of data instead of a series of OFDM symbols. The synchronization function slices the data stream properly to delineate the OFDM symbols. (To achieve this result, the synchronization circuit must detect Zadoff-Chu primary synchronization signals using multiple correlators on

deeply decimated input data. Afterwards, it will be possible to obtain OFDM symbols.) Finally, FFT transformation follows the removal of the cyclic prefix and the resulting data passes to another DSP using the EMIF interface.

The downlink flow involves two DSPs mutually connected by means of a serial RapidIO interface. These DSPs perform frequency correction, channel estimation, equalization and MIMO decoding. Then they do data- and control-channel extraction, Viterbi and turbo decoding, deinterleaving and descrambling prior to MAC-layer interworking.

On the uplink side, the third FPGA handles the loopback test between uplink and downlink FPGAs and ensures the SDR board's conformance to the CPRI/OBSAI standards.

Our design team extensively used Xilinx CORE Generator™ IP to produce filters, DDS, FFTs, Block RAMs, FIFOs and MACC functions, using DSP48E and DCMs for the clocking deskew section of

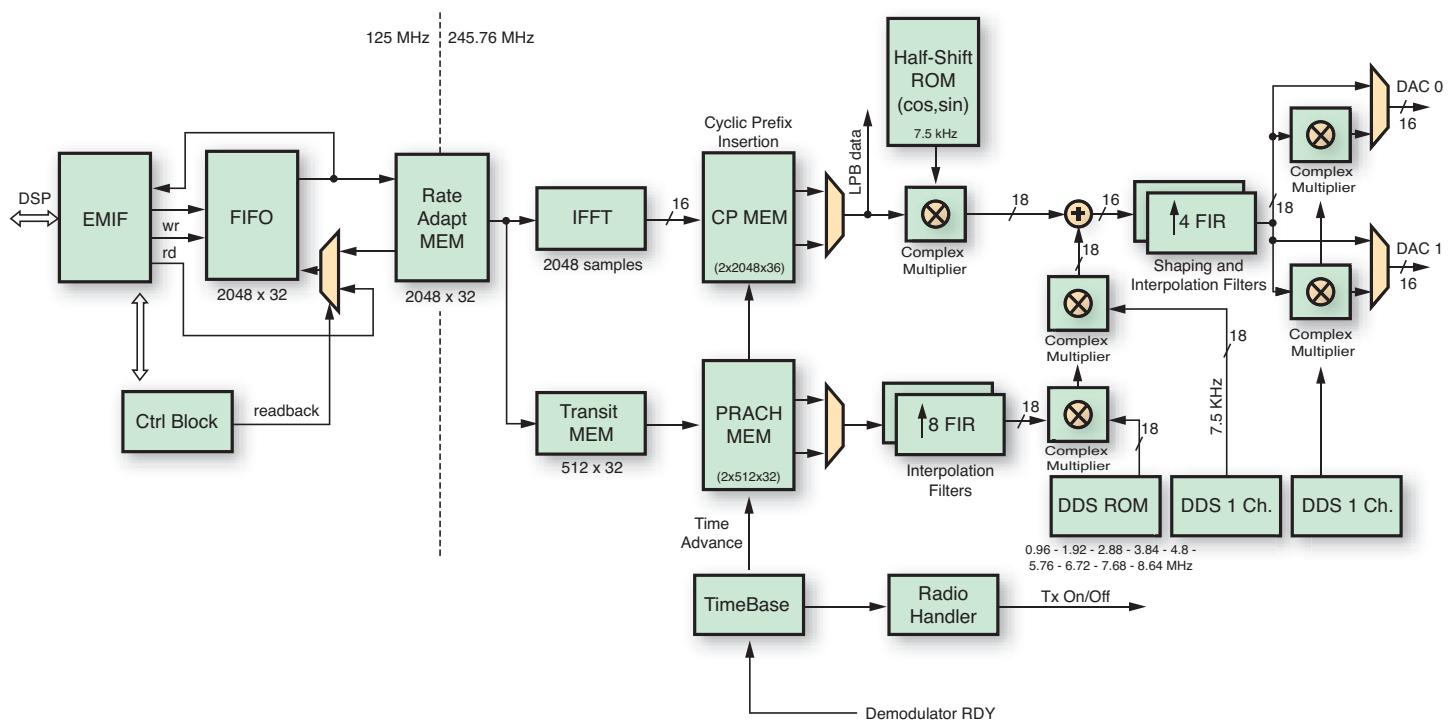


Figure 3 – The front-end “uplink FPGA” implements inverse FFT, cyclic prefix insertion, filtering, IF upconversion and other operations for time-division duplex and PRACH handling. The system sends the same signal into two DACs for redundancy.

Because this project had a very aggressive time-to-market deadline, we made a careful analysis of functions partitioning. The FPGAs would have accommodated even more LTE functions, but one of our design goals was to find a balance between the system's FPGA and DSP sections.

the design. This massive instantiation methodology produced a compact design in a reduced development time.

FPGA Design Strategy

Because this project had a very aggressive time-to-market deadline, our team made a careful analysis of functions partitioning among FPGAs and DSPs. It's worth noting that the FPGAs would have accommodated even more LTE functions, but one of our design goals was to find a balance between the FPGA and DSP sections of the system.

The FPGA clock rate was one of the tougher challenges in this design. Using a clock rate of 245.76 MHz for a large design

like a modulation system is not a trivial matter. Our design team had many issues to consider, such as power consumption, design constraints, placement and routing efforts. Nevertheless, thanks to the ISE® Design Suite, which produced stable and quality results over the various design iterations, an oversampled factor of eight (FPGA clock rate/OFDM symbol rate) kept design items like filters and FFT transforms as small as possible with respect to the required LTE functionality. The ISE software also helped us achieve a reasonable synchronization circuit area.

Key to our design was devising a radio card architecture that in uplink, instead of

using a direct-conversion methodology with I/Q unbalance drawback, received the FPGA data from an intermediate frequency. Using Xilinx Direct Digital Synthesizers, an 18-bit sine/cosine wave performed a perfect signal carrier to the complex modulation, as confirmed by the error vector magnitude measured on the transmitted radio signal.

Thanks to the use of Xilinx Virtex-5 FPGA and TI DSP technologies, the LSU UeSIM LTE Simulator has become the leading-edge test equipment for load-and-stress solutions in the cellular world. It provides a powerful, flexible and scalable solution for SDR systems. ●

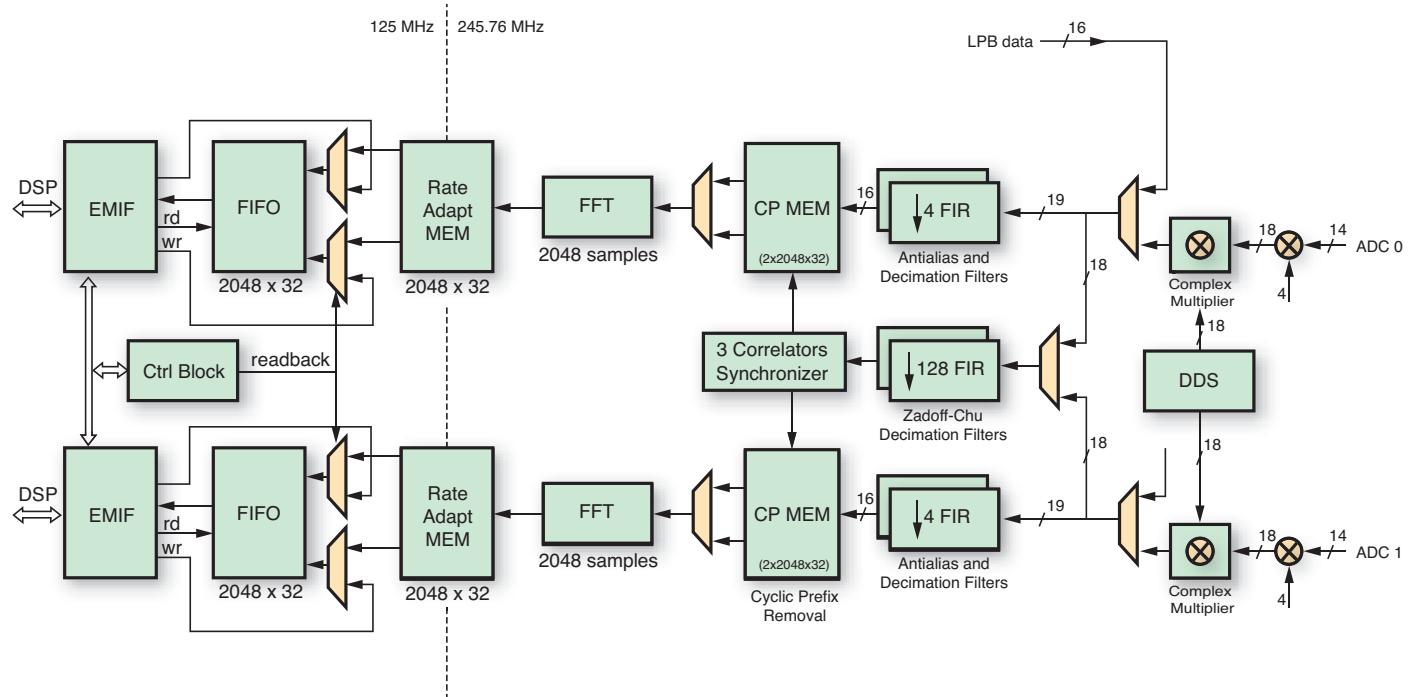
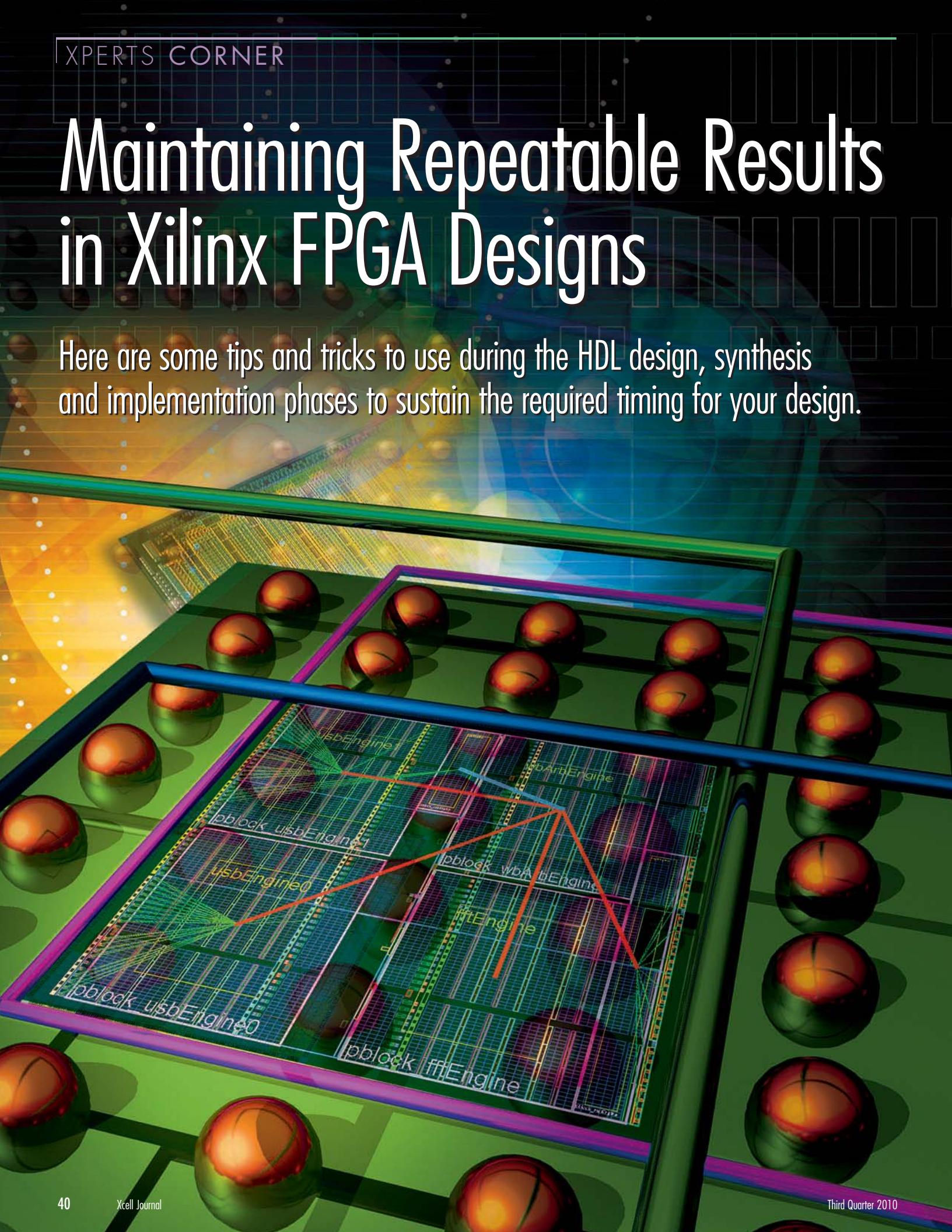


Figure 4 – The front-end “downlink FPGA” implements IF downconversion, polyphase decimation filtering, synchronization, cyclic prefix removal and direct FFT. The system uses two chains to support MIMO operations for TDD and FDD modes.

Maintaining Repeatable Results in Xilinx FPGA Designs

Here are some tips and tricks to use during the HDL design, synthesis and implementation phases to sustain the required timing for your design.



by Kate Kelley
Staff Product Marketing Engineer
Xilinx, Inc.
kate.kelley@xilinx.com

Meeting the timing requirements in a design can be difficult in itself, but producing a design whose timing is 100 percent repeatable can sometimes seem nearly impossible. Fortunately, designers have access to design flow concepts that can help to maintain repeatable timing results. The four areas that have the most impact are HDL design practice, synthesis optimizations, floorplanning and implementation options.

Designs with very high resource utilization and frequency (QoR) requirements are the most challenging in terms of obtaining repeatable results. They are also the designs that need a repeatable-results flow the most. The first step in getting repeatable results is to use good design practices during the HDL design phase. Following good hierarchical-boundary practices helps to keep logic together, which helps to maintain repeatable results when making design changes. One good rule is to put logic that needs to be optimized, implemented and verified together in the same hierarchy. Also, register the inputs and outputs of modules. This keeps the timing paths contained within a module, so that changes to one module are less likely to affect another module. Finally, keep all logic that needs to be packed into larger FPGA resources (Block RAM or DSP, for example) in the same level of hierarchy.

Logic Levels

Repeatable results are very difficult to obtain from designs that have too many lookup-table (LUT) logic levels for the required QoR results. Often, it is not the LUT delay that is the issue, but rather the routing delay between the LUTs. This is extremely important in the high-performance areas of the design.

Some common sources of too many logic levels include large if/else constructs and large case statements. When appropriate, use “full_case” and “parallel_case” Verilog directives to optimize the case statement with less

logic, a technique that often results in fewer logic levels. Large multiplexers or decoders can create routing congestion, resulting in unrepeatable results. A multistage registered multiplexer/decoder path can help with this issue. For adders, using a registered adder chain instead of a registered adder tree can improve performance. The chain will introduce more latency than the tree if all the adders are registered.

For more information on best practices for coding, see the Xilinx® white paper “HDL Coding Practices to Accelerate Design Performance” (WP231) at http://www.xilinx.com/support/documentation/white_papers/wp231.pdf.

Resets and Other Control Signals

The choice of resets affects the performance, area and power of a design. A global reset is not necessary for circuit initialization on power-up, but it can have a major effect on the type of resources you can use in a design. Shift registers (SRLs) cannot be inferred if there is a global reset in the HDL. One shift register produces more-repeatable results than 10 registers.

Also, the DSP and Block RAM registers contain only synchronous resets. If you put an asynchronous reset in the code, these registers cannot be used, forcing the design to use configurable logic block (CLB) registers instead. The same results are easier to maintain if the registers are packed into the DSP, Block RAM or both.

Using synchronous resets on general logic might reduce the levels of logic. The slice registers can have asynchronous or synchronous resets. If the design uses the synchronous reset, then the synchronous set is available for use by the combinatorial logic. This could reduce the logic levels by one LUT.

A control set consists of a unique grouping of clock, clock enable, set, reset and, in the case of distributed RAM, write-enable signals. Control set information is important because registers must share the same control set to be packed in the same slice. This can affect packing and utilization, creating repeatable-result issues.

For more information on resets, see Xilinx WP272, “Get Smart About Reset:

Think Local, Not Global” (http://www.xilinx.com/support/documentation/white_papers/wp272.pdf). For more information on control sets, see WP309, “Targeting and Retargeting Guide for Spartan®-6 FPGAs” (http://www.xilinx.com/support/documentation/white_papers/wp309.pdf). While this white paper is specific to Spartan-6 devices, it contains good general information applicable to all FPGAs.

Understanding FPGA Resources

It is important to understand what FPGA resources are available and when it is best to use them. Often, there are synthesis directives to define which resources to use. For instance, Block RAM is best for deep-memory requirements, while distributed RAM works well for wide buses, especially where regional clocks are clocking high-speed data. Both Block RAM and distributed RAM can have issues with large fanout on control signals. Duplicating control signals and using floorplanning techniques to keep blocks with the same signals together can help maintain repeatable results.

Shift registers can reduce the utilization of a design, which helps repeatability. There are several performance issues to keep in mind. The clock-to-out of an SRL is slower than clock-to-out of a flip-flop; therefore, it is best to use a flip-flop as the last stage of a shift register. Most synthesis tools do this automatically, but if there is an issue with a path involving shift registers, it is good to confirm that the last stage is a register.

Similar issues are associated with the initial register. Having a flip-flop in front of an SRL gives the placer more options to meet timing, therefore maintaining results. Again, most synthesis tools do this automatically, but if there is an issue with a path involving shift registers, it is good to confirm that the first stage is a register.

FPGAs have many registers, making pipelining a useful technique to improve performance. It is important to disable SRL inference on multiple pipelined flip-flops.

The white paper cited above on HDL coding practices (WP231) offers more information on Block RAM. For more

It is important to use timing constraints when running synthesis. Often, users overconstrain during synthesis, then relax the timing constraints in the Xilinx implementation tools. This makes the synthesis tool work harder, relieving the burden on the implementation tools.

information on shift registers, see WP271, “Saving Costs with the SRL16E” (http://www.xilinx.com/support/documentation/white_papers/wp271.pdf).

Clock Domain Issues

Designers must take care to correctly constrain paths that cross unrelated clock domains. The tools automatically relate clocks from the same source clock (for example, a DCM). The PERIOD constraint can also relate external clocks. Unrelated clocks that are not created internal to the device take special consideration. By default, the tools will not constrain these clocks. If there are special timing considerations, designers must use the FROM:TO constraint to correctly constrain the path. The DATAPATHONLY keyword tells the tools not to include clock skew in the equations.

For more information, see the “Asynchronous Clock Domains” section in UG625, *Xilinx Constraints Guide* (http://www.xilinx.com/support/documentation/sw_manuals/xilinx11/cgd.pdf), or WP257, “What Are PERIOD Constraints?” (http://www.xilinx.com/support/documentation/white_papers/wp257.pdf).

It is also important to ensure that race conditions do not occur. FIFOs can help when crossing from one domain to another. Otherwise, designers should double-synchronize one—and only one—control signal, and use it in the receiving clock domain to receive other signals.

High-Fanout Signals

Often, high-fanout signals can be the gating factor in a design. Even though most synthesis tools have fanout control, it’s a good idea to duplicate these signals in the HDL to get more-repeatable results.

Designers should combine this tactic with directives to ensure that the synthesis tools do not remove the duplicates. If a high-fanout signal is in the top-level logic, one method is to duplicate the signal and then drive each top-level module with a separate signal.

If the synthesis tool fanout control is not giving the desired results and modifying the HDL is not a palatable option, then using the Register Duplication constraint within the MAP logic of the BRAM, along with the max-fanout constraint, often makes better register duplication choices than synthesis. For more information, see MAX_FANOUT in the Constraints Guide (UG625).

As a general debug issue, meanwhile, it is easier to trace a problem path if a signal name is kept constant when crossing hierarchies. If the name constantly changes, it is difficult to follow in the timing reports and other debug output. It is also helpful to put the signal direction on the port definitions for all modules or entities.

Synthesis Optimizations

Synthesis has a big effect on repeatable results. If the output netlist from synthesis is not optimal, then it is impossible to have ideal conditions in the implementation tools. Designers can use several synthesis techniques to help improve implementation results.

It is important to use timing constraints when running synthesis. Often, users overconstrain during synthesis, then relax the timing constraints in the Xilinx implementation tools. This makes the synthesis tool work harder, relieving the burden on the implementation tools.

Use the timing report from the synthesis tools. If a path is failing timing in syn-

thesis and implementation, modify the HDL or synthesis options to meet timing after synthesis. This saves time during implementation runs.

The best way to have repeatable results in the implementation tools is to have repeatable results during synthesis. Most synthesis tools support a bottom-up flow, with separate synthesis projects for the top level of the design and each of the lower-level modules. The user is in control of which netlist is updated, based upon HDL changes. Most commercially available synthesis tools have an incremental flow.

Importance of Floorplanning

Floorplanning locks placement of components to a specific location in the design or to a range. This reduces the variability of placement, increasing the repeatability of a design. You can almost always obtain better performance by floorplanning or by using location constraints, or both.

That said, a bad floorplan or poor location-constraint choices can make it impossible to meet timing. Floorplanning is somewhat of an art and requires advanced knowledge of the tools and the design. You can use implementation results that meet timing as a guide to creating a good floorplan.

If board requirements are the main means for selecting pinouts, FPGA implementation tools might have a difficult time maintaining repeatable results. But designers have access to several techniques that can help achieve repeatability.

First, be aware of the data flow. For example, data can go from the center I/Os to the side I/Os. Keep all of the pins associated with the bus in the same area of the FPGA to limit the routing distance on control signals. Place I/O bus control signals

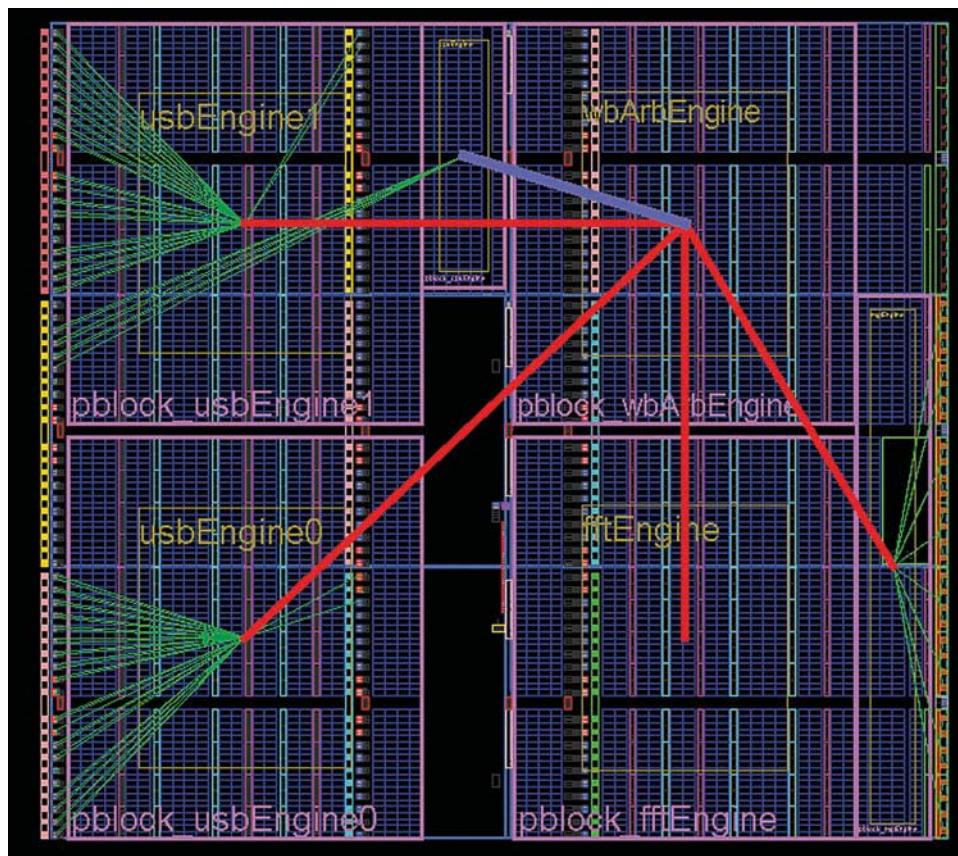


Figure 1 – PlanAhead shows connections between modules, providing guidance when creating an area group floorplan.

near the address and data buses. Signals that are to be optimized together need to be placed together. If board routing is a bigger concern, pipelining registers on the I/Os can help FPGA routing with less-than-ideal pinouts.

Area Group Floorplanning

Area group floorplanning is a high-level floorplanning technique that defines where modules are located within the FPGA. It is very easy to do, but it is often misused, delivering poor floorplans that create more issues than they resolve. There are general guidelines for good floorplans that will help you avoid these pitfalls.

Keep utilization similar across all area groups. For example, do not have one at 60 percent and another at 99 percent. Do not overlap area groups. The one exception is if two different area groups have some logic elements that need to be

placed together, it is acceptable to overlap by one or two rows or columns of CLBs. The user is then responsible for making sure there are enough resources for both area group constraints.

If two different logical portions of the design need to be in the same physical location, put both of them into the same area group. One level of nesting, a child area group within a parent area group, is typically acceptable. This can be necessary if a small portion of a larger area group needs to be located in a tight region.

It is important to floorplan only the critical portion of the design and let the tools determine the placement of the non-critical logic. Logic connected to fixed resources (for example, I/O, transceiver or processor blocks) might benefit from floorplanning. Use the results of a good implementation run as a guideline to identify placement or timing issues. Tools

like Xilinx's PlanAhead™ software (Figure 1) and Timing Analyzer can help visualize the issues.

It is often helpful to minimize the number of regions used for each global clock and the number of clocks (regional and global) in each region. Do not over-constrain and plan accordingly if you are going to add more logic to a clock region. When all clocks in a clock region are used, it can be difficult to find a valid placement. The PlanAhead software's ability to snap to a clock region will make this floorplanning easier. For Virtex® FPGA designs with more than 10 global clocks, the clock regions used in the current implementation are in the .map report file, along with UCF constraints.

For more information on area group floorplanning, see UG632, *PlanAhead User Guide* (http://www.xilinx.com/support/documentation/sw_manuals/xilinx11/)

PlanAhead_UserGuide.pdf, and UG633, *Floorplanning Methodology Guide* (http://www.xilinx.com/support/documentation/sw_manuals/xilinx12_1/Floorplanning_Methodology_Guide.pdf).

Locating Blocks, Module, Path

Often, locating the core components, such as the Block RAMs, FIFOs, DSPs, DCMs and global clocking resources, can help

information is not saved. More information on location constraints in the PlanAhead software can be found in the “Floorplanning the Design” chapter in UG632, *PlanAhead User Guide*; in UG633, *Floorplanning Methodology Guide*, and in the PlanAhead tutorial.

If locking down an entire module is overkill, it is possible within PlanAhead software to lock down a critical path. But

are not pushing the absolute maximum QoR or utilization. If neither design preservation nor SmartGuide technology is appropriate for a design, then use SmartXplorer or PlanAhead software strategies to maintain timing.

For designs with high QoR requirements, there are advanced implementation options to help maintain timing. Often, managing utilization is key to maintaining repeatable

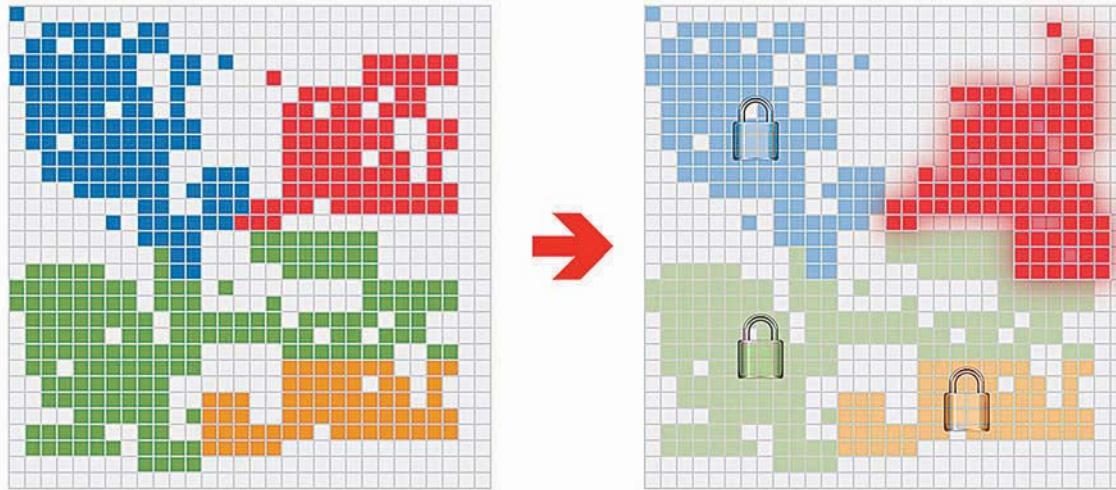


Figure 2 –The design preservation flow locks down unchanged portions of the design and implements the rest.

achieve repeatability. This is best done by looking at a good placement and using the design knowledge to verify these are good locations. You need to take control-signal and data flow (bus alignment) into account when locating these BRAM, FIFO and DSP components. Constraints to locate the clock regions of an existing design are found in the .map report file. Keeping the same clock regions prevents the placer from making changes to the clock region partitioning, which could change the floorplanning of the design. Use reportgen – clock_regions design.ncd to create the report.

PlanAhead software has the ability to lock down all the placement information on critical modules. On the next run, the placement is the same, but the routing

you should use this technique only in a very limited manner. If there is a specific path that is causing the majority of problems, it is better to fix the timing issue by changing the HDL. If this is not possible, limited use of locating specific timing paths can be helpful.

Implementation Options

Several options in the implementation tools improve repeatability. Design preservation using partitions is the best methodology to preserve implementation, but it is not a good fit for all designs and it does have HDL design requirements. Xilinx SmartGuide™ technology is another option for maintaining repeatable results. This is best for designs that

results. As designs increase in size it is more difficult to maintain results. Staying with the same software release for the entire design phase helps achieve repeatable results.

Design Preservation

The design preservation flow in PlanAhead makes use of partitions; this is the only method that guarantees repeatable results. The main goal of design preservation is to enable consistent module performance to reduce the amount of time spent in the timing-closure phase. It also requires the greatest commitment from the user to following good design practices.

Partitions preserve unchanged portions of the design that have been previously implemented. If a partition's netlist is

The best way to influence design repeatability is to follow good design methodology in the HDL and fix any timing issues by changing the HDL. If that is not possible, synthesis, floorplanning and implementation techniques can help.

unchanged, the implementation tools use a copy-and-paste process to guarantee that the implementation data for that partition is preserved. By preserving implementation results, partitions let you implement the modified portions of the design without affecting the preserved portion. In Figure 2, the red module has been changed and is therefore implemented, while the rest of the modules are locked in place.

In version 12.1 and future releases, the PlanAhead software and command-line tools support design preservation. For more information, see WP362, “Repeatable Results with Design Preservation” (http://www.xilinx.com/support/documentation/white_papers/wp362.pdf), and UG748, *Hierarchical Design Methodology Guide* (http://www.xilinx.com/support/documentation/sw_manuals/xilinx12_1/Hierarchical_Design_Methodology_Guide.pdf).

SmartGuide Technology

SmartGuide technology uses the previous implementation results as a starting point when running an implementation. The main goal is to reduce run-time. Guided placement, routing or both can be moved in order to route the design or meet timing. SmartGuide technology works best for designs that are not trying to push the limit on QoR or utilization.

Previous versions of the tools featured an exact guide and a leveraged guide. Often, the exact-guide methodology resulted in unroutable designs. If exact preservation is required, then the suggested flow is design preservation. SmartGuide technology is the replacement for the leveraged guide.

Designers often ask whether to use SmartGuide technology or partitions. The answer depends upon where you are in the

design flow. SmartGuide technology works best at the end of the design cycle when you are making small design changes. Using this flow, it is easy to see if the proposed changes work with the design. Partitions require a greater commitment up front to following good design hierarchy rules. You should decide whether to adopt the design preservation flow with partitions when starting to organize the HDL. An exception to this rule is when the design already follows the hierarchical rules for partitions.

For more information, refer to UG748, the *Hierarchical Design Methodology Guide* (http://www.xilinx.com/support/documentation/sw_manuals/xilinx12_1/Hierarchical_Design_Methodology_Guide.pdf).

SmartXplorer

SmartXplorer and PlanAhead software strategies are similar tools that help achieve timing closure. They run different sets of implementation options to find the best fit for the design. You can then use these results to see what placements tend to have better timing results and to create good area group floorplanning. The different results can also point to a design issue. If the same path is failing across all runs, it is beneficial to change the HDL to remove the timing issue.

In the beginning of the design, it is best to use the default effort levels for MAP and PAR. Using too many advanced options in the beginning can hide a timing issue that might be best solved by modifying the HDL. When the device utilization increases, it becomes harder and harder for the tools to converge on a solution that meets timing. If you use the default options, then the higher-effort options are available to get the last few picoseconds of timing later

in the design flow, allowing timing results to be maintained.

Designs with low utilization of LUTS/FFS (<25 percent) or high utilization of LUTS/FFS (>75 percent) can be difficult to place and route with consistency. For designs with high utilization, look at slice control sets, resets (FPGAs often do not require synchronous resets/sets), modules with higher-than-expected logic usage (easily done in PlanAhead) or SRL/DSP48 inference.

The flip side of high utilization is low utilization. With designs that have 25 percent utilization or less of all component types, the low-utilization algorithm takes effect and keeps components tightly placed. However, if I/O utilization exceeds 25 percent, then the implementation tools could spread out the design in order to keep logic near the I/Os. Careful placement of I/Os and use of area groups can minimize this issue.

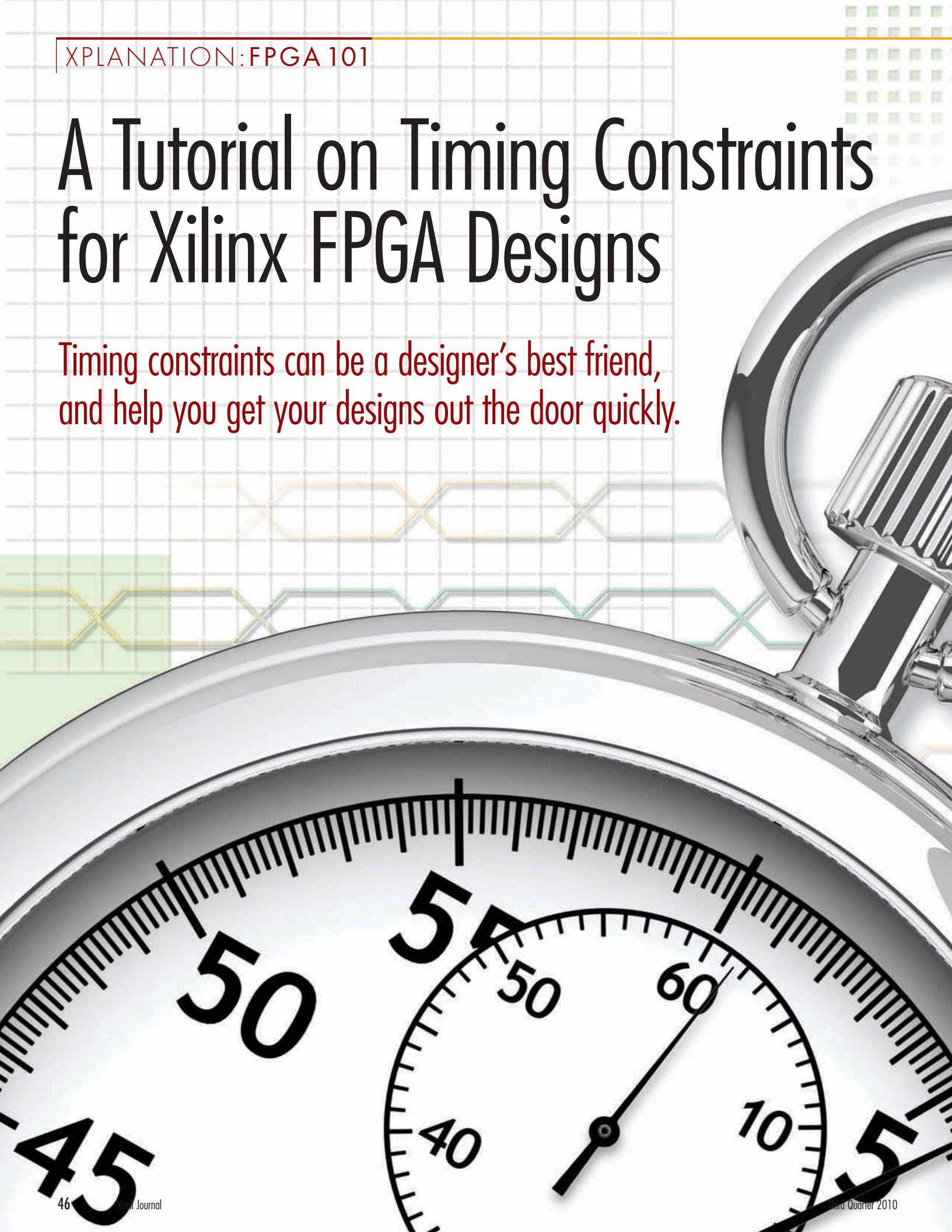
Software Releases

Try to use the same major software release during the timing-closure phase. Because algorithms change from one release to another, a technique that worked in one might not work the next time. Also, methods that rely on the previous results (partitions and SmartGuide technology) might not work across major releases.

The best way to influence design repeatability is to follow good design methodology in the HDL and fix any timing issues by changing the HDL. If that is not possible, synthesis, floorplanning and implementation techniques can help. Design preservation using partitions is the flow that guarantees instance performance. SmartGuide technology is another solution that uses previous implementation results. 

A Tutorial on Timing Constraints for Xilinx FPGA Designs

Timing constraints can be a designer's best friend, and help you get your designs out the door quickly.



by Austin Lesea
Principal Engineer
Xilinx, Inc.
austin.lesea@xilinx.com



As someone who regularly participates in Xilinx's user forums (see <http://forums.xilinx.com>), I've noticed that new users often find timing closure, and the use of timing constraints to achieve it, a mystery. To help those who are new to FPGA design achieve timing closure, let's take an in-depth look at timing constraints and how you can leverage them to get optimal results in your FPGA design projects.

What Are Timing Constraints?

To guarantee your design will be successful, you have to ensure that it will perform the tasks it was designed to do in a specific time frame. To make sure this happens, we apply timing constraints to the nets—the path or paths taken from one FPGA element to the inputs of subsequent elements in the FPGA or on the PCB in which the FPGA resides.

In FPGAs, there are mainly four types of timing constraints: PERIOD, OFFSET IN, OFFSET OUT and FROM:TO (multicycle) constraints.

PERIOD Constraint and Grouping

Every synchronous design will have at least one PERIOD constraint (Clock Period Specification), the most basic type of constraint, which specifies the clock and its duty cycle. If there is more than one clock in your design, each clock will have its own PERIOD constraint. The PERIOD constraint will dictate how we must route nets to meet the timing requirements a design needs to operate properly.

To simplify the process of applying timing constraints, you'll often group nets that have similar attributes as, for example, a bus or a control group. Doing so will also help you perform the critical step of properly prioritizing design constraints.

Prioritize Design Constraints

When you have a design with multiple constraints, you need to prioritize those constraints. Typically, the more general the

constraint, the lower its priority. Conversely, the more specific a constraint, the higher its priority. For example, a PERIOD constraint on a clock net or network is very general and will be overruled by a higher-priority FROM:TO constraint on a specific net or network.

The specific constraint for the FROM:TO (or FROM:THRU:TO) is deemed more important than the more general constraint for any net within a clock domain.

To help you prioritize constraints, you can run the Xilinx® Timing Analyzer (a static timing-analysis tool in the ISE® Design Suite) and have it generate a timing-specification interaction report, or as it is commonly called, a .tsi report. This report will let you see how the constraints are interacting and what priorities the tool has set them to by default.

You can override the assumed priorities and manually set the priority of any timing constraint by using the PRIORITY constraint keyword. This is especially useful in situations where there is a conflict between two or more timing constraints that cover the same path. Priority here means which of any number of timing constraints will be applied if two or more constraints cover the path. The other, lower constraints are ignored. You can set priority from -10 to +10. The lower the PRIORITY value, the higher the priority. Note that this value does not affect which paths are placed and routed first. It only affects which constraint controls the path when two constraints of equal priority cover the same path.

Let's take a closer look at the following example in which PERIOD only covers nets from synchronous elements to synchronous elements, like FFS to FFS (constraints are in blue, below):

```
NET "clk20" TNM_NET = "tnm_clk20";
TIMESPEC "TS_clk20" = PERIOD
"tnm_clk20" 20 ns HIGH 50 %;
```

A TIMEGRP (timing group) is created called tnm_clk20 and contains all of the downstream synchronous components that net clk20 drives. All of the paths between

Only use FPGA Editor to ‘see under the hood’ and learn what the tools are doing with your design in order to fit it into the FPGA device.

Try first to rearchitect the circuit to meet your design’s timing requirement.

these synchronous elements are then constrained with the timing specification “TS_clk20: 20 ns”—a 20-nanosecond requirement from synchronous element to synchronous element. “HIGH 50%” means that clk20 has a 50/50 duty cycle.

In a second example, we use **FROM:TO** constraints to define a requirement for paths that go between two groups, as shown:

```
TIMESPEC TS_my_fromto = FROM
my_from_grp TO my_to_grp 40 ns;
```

That command tells the tools to ensure that data makes it from the components in the timing group “my_from_grp” to “my_to_grp” in 40 ns. Timing Analyzer will still calculate the clock skew from source group to destination group but at a lower priority if the clocks are related. You can also use predefined groups such as the following:

```
TIMESPEC TS_F2F = FROM FFS
TO FFS 40 ns;
```

If you need to leave out the time unit (nanoseconds, picoseconds, etc.), then the tools automatically assume everything is in nanoseconds. For example, you could write a constraint as

```
TIMESPEC TS_P2P = FROM PADS
TO PADS 30;
```

You could also leave FROM or TO off the constraint and make it more generic:

```
TIMESPEC TS_2P = TO PADS 40;
```

As previously stated, the tools will automatically assume all these FROM:TO constraints in the examples above are higher priority than the PERIOD constraint unless you specify otherwise.

Closer Look at a .tsi Report

In addition to helping you observe timing constraint interactions, the .tsi report will also make suggestions on ways to improve constraints in the universal constraints file (UCF). The report will also notify you if any paths are constrained by multiple clock domains. Here is an example of a constraint interaction report:

```
Constraint interactions for
TS_clk0_1 = PERIOD TIMEGRP "clk0_1"
TS_clk HIGH 50%;
1 paths removed by TS_my_fromto =
MAXDELAY FROM TIMEGRP "my_to_grp" TO
TIMEGRP "FFS" 40 ns;
```

In this example, the higher-priority FROM:TO constraint (just one) was applied ahead of the PERIOD constraint.

Setup and Hold

In a practical synchronous digital system, the data must arrive before the clock edge that samples it. The minimum amount of time it takes for this to happen is called the “setup time.”

As well as arriving before the clock edge, the data must persist for some finite amount of time at the clock edge, a period called “hold time.” A hold time may be negative, in which case the data goes away before the clock edge; zero, which means the data persists until the clock edge samples it; or positive, which means the data persists for some time after the clock edge has completed sampling it.

By design, in the FPGA fabric, for all speed grades, hold times are not positive (they are either zero or negative). This simplifies the placement and routing, since the data only needs to arrive before the clock edge and is allowed to change immediately after a clock edge sampling takes place.

The value by which the data exceeds the minimum setup time is known as slack.

Slack should always be positive. If a report shows a negative slack, then the setup timing has not been met adequately—the data arrived too late.

The clock path itself has delay, or skew. Thus, to analyze the timing, the tools will calculate the arrival time of the data and the clock at the flip-flop of interest.

Easy Remedies for Constraint Violations

To recap for a moment: The PERIOD constraint defines the clock period for synchronous elements of interest, such as flip-flops. You can use the timing analyzer to verify that all paths between synchronous elements meet the setup-and-hold timing for your design. A violation of this PERIOD constraint will appear in the timing report and have a negative slack value, identified as violating either the setup requirement or the hold requirement.

So what happens if the report shows that the setup has indeed been violated? You know that you will either have to find a faster path between the two synchronous elements in question, or at least a way to ensure the data arrives at a proper time and sticks around long enough so that the clock edge registers it. If the place-and-route software cannot find a faster path, you have the option of placing the path manually in the FPGA Editor tool.

But this is a tool of last resort. Do not use it to solve problems before you have learned how to solve the problems without it. Only use FPGA Editor to “see under the hood” and learn what the tools are doing with your design in order to fit it into the FPGA device. Try first to rearchitect the circuit to meet your design’s timing requirement. One of the simpler ways to do this is to place a flip-flop earlier in the path. This technique, known as pipelining, will add latency to the signal, but it will also allow the value to be captured properly.

If the hold has been violated (the data went away before the clock edge arrived), then this is often an indication that you have a design problem (a bad architecture). Values should only change on the clock edge, and not before. If an external value is changing before the clock edge, you need to delay the clock edge (using a DCM or PLL) so that the data is now registered properly by the new, delayed clock.

An alternative is to use the IDELAY element in the input/output block to move the data to where the clock is valid.

Data-Valid Window and Metastability

The time from before the clock edge (setup) plus the time after the edge (hold) is known as the “data-valid window,” or the time the data must be stable to be properly registered. If the data is not valid for at least this amount of time, then the results are indeterminate, or unknown.

But just because the data was not valid for as long as required does not mean that the output of the flip-flop is metastable. Metastable is different from indeterminate. An output could be 0 or 1, seemingly at random, if the timing is not met. Metastability means the edge was “almost” capable of capturing the state, and the flip-flop output is in some intermediate state (not 1, not 0) for some time after the clock edge. Metastability cannot be prevented, as it is a fact of the physics of the circuits if the clock edge and the data are almost perfectly “missed.”

In a properly designed synchronous system, there are no problems with metastability. Metastability is a problem when something is asynchronous (like pressing a key on a keyboard) or when two synchronous clocks are asynchronous to each other. In general, when something is asynchronous, it needs to be synchronized.

If you would like to learn how to deal with metastability, here is a link to a fabulous presentation on the subject: http://www.stanford.edu/class/ee183/handouts_spr2003/synchronization_pres.pdf. (For more on metastability, see the second FPGA101 article in this issue.)

Propagation Time and OFFSET Constraints

The time it takes to get a signal from point A to point B is called the propagation time. It is based on the speed of light, in the medium it is in. For example, a trace on a printed-circuit board carries signals at around 6 to 7 ps per millimeter. You can find this number in a variety of ways, including running simulations or solving equations when you know the dielectric constant for the material and the geometry of the wiring traces. Inside the silicon device, the signals behave in much the same way, but also may be delayed by

Putting Constraints to Use with DDR Memory

Now that we've gone through the basics of timing constraints, let's look at how we can put them to use, in this case with double-data-rate (DDR) memory.

DDR interfacing uses both the rising and falling edges of the clock in a source-synchronous interface to capture or transfer twice as much data per clock cycle.

To properly constrain data arriving at the device, you must first constrain the clock you are using to capture the data. At the same time, you must also constrain the arrival of the data for both the rising and falling edges of the clock.

For this example, the complete OFFSET_IN specification with associated PERIOD constraint would look like this:

```
NET "SysClk" TNM_NET = "SysClk";
TIMESPEC "TS_SysClk" = PERIOD "SysClk" 5 ns HIGH
50%;
OFFSET = IN 1.25 ns VALID 2.5 ns BEFORE "SysClk"
RISING;
OFFSET = IN 1.25 ns VALID 2.5 ns BEFORE "SysClk"
FALLING;
```

where “VALID” and “BEFORE” are reserved words defining the timing relationship in the constraint.

This global constraint covers both of the data bits of the bus, since in each clock period two bits are captured, namely ?data1 and ?data2.

In much the same way as specifying when the data arrives, you also need to specify the output of the DDR data.

For this example, the complete OFFSET_OUT specification for both the rising and falling clock edges is based on the clock supplied to the DDR register:

```
NET "ClkIn" TNM_NET = "ClkIn";
OFFSET = OUT AFTER "ClkIn"
REFERENCE_PIN "ClkOut" RISING;
OFFSET = OUT AFTER "ClkIn"
REFERENCE_PIN "ClkOut" FALLING;
```

Note here that in the complete constraint format, OFFSET=OUT <value>, determines the maximum time from the rising clock edge at the input clock port until the data first becomes valid at the data output port of the FPGA device.

When you omit <value> from the OFFSET_OUT constraint (as in the example above), the constraint becomes a report-only specification that reports the skew of the output bus. The REFERENCE_PIN keyword defines the regenerated output clock as the reference point against which the skew of the output data pins is reported.

Of course, do not forget that the output clock also needs a PERIOD constraint. It was not needed for the specification of the output timing, but is required for getting the data to the DDR output register.

— Austin Lesea

**In general, the fewer constraints the better.
Complex constraints can often cause more problems
than they solve. You may wish to declare that no constraints
should be applied to certain noncritical paths or nets.**

going through active circuits such as buffers, inverters, logic and interconnect.

You can also measure propagation times, often with the help of an oscilloscope. Propagation times generally do not vary much at all when the path has no active elements. If the path is in silicon, the strength of the transistors will cause the path delay to vary with both a maximum value and a minimum value. A design needs to meet timing for both.

In order to tell the tools when data arrives at a particular location, you need to use another type of constraint called OFFSET_IN. An OFFSET_IN constraint defines the relationship of a clock and data as they enter the device. Take, for example, the following constraint:

```
OFFSET = IN 2 ns VALID 16 ns
BEFORE "clk20";
```

This constraint tells the tools that data will be set up at PADs 2 ns before the clk20 rising edge. It tells the tools that data will remain valid for 16 ns after it arrives. This constraint applies only to PADs that go to registers that are clocked by clk20 or a derivative (that is, a derived constraint).

OFFSET requires a PERIOD constraint on clk20, so that it understands the clocking structure. This is also acceptable:

```
OFFSET = IN 2 ns BEFORE "clk20";
```

However, this constraint will not check the hold time, because we don't know when the data goes away at the pin of the FPGA. If the data won't be set up until 2 ns after the clock edge, then we use the following:

```
OFFSET = IN -2 ns VALID 16 ns
BEFORE "clk20"; #
```

While the OFFSET_IN pertains to the clock and data entering the device,

another common constraint called OFFSET_OUT defines the amount of time it takes for data to make it out of the device after a clock transition at the input to the FPGA. Here is a common use of OFFSET_OUT:

```
OFFSET = OUT 3 ns AFTER "clk20";
```

This constraint tells the tools that you need to ensure data is at the output pin of the FPGA 3 ns after a clock transition at the input of the specified clock to the FPGA. This constraint applies only to PADs that are driven by registers that are clocked by clk20 or a derivative (a derived constraint). OFFSET requires a PERIOD constraint on clk20, so that it understands the clocking structure. Hold times are not constrained for OFFSET_OUT.

If we need the data 2 ns before the clock edge, then we use this:

```
OFFSET = OUT -2 ns AFTER "clk20";
```

Groups and Group Names

A time group is a way to identify a constraint for a collection of paths or nets between synchronous elements. To add components to a time group, you would use TNM, TNM_NET or TIMEGRP.

Paths are constrained by defining groups and then giving requirements between those groups. A few constraints do not require time groups, such as NET MAXDELAY. The maximum delay (MAXDELAY) attribute defines the maximum allowable delay on a net.

Timing Names

To add a component to a user-defined group, you can do the following:

```
[NET|INST|PIN] object_name TNM =
predefined_group identifier;
```

where “TNM” is a reserved word defining the name for a timing group.

In this case, object_name is the name of the element or signal to be grouped, predefined_group is an optional keyword and identifier can be any combination of letters, numbers or underscores.

Do not use reserved words such as FFS, LATCHES or RAMS. This variable is case-sensitive (TNM=abc is not the same as TNM=ABC). You can apply TNM to any net, element pin, primitive or macro.

Components can be part of more than one group. For example, my_ffs_group TNM can have the my_ff component in it. Likewise, my_ffs_group2 TNM can also have the my_ff component in it.

To create a group

```
NET CLOCK TNM=clk_group;
```

you can make any keyword element into a group for timing purposes. In this example, the NET CLOCK is traced forward to the flip-flops (FFS). These flip-flops are timing-named (TNM) with the name clk_group. As a result, clk_group can now be referenced by this TNM in TIMESPECs.

You can also create a group using an instance, such as:

```
INST macrol TNM = LATCHES
latchgroup;
```

All LATCHES in the macro called macrol will be in a group called latchgroup. Likewise, in the constraint INST mymac TNM = RAMS memories; all RAMS in the macro called mymac will be in a group called memories. And in the constraint

```
INST tester TNM = coverall;
```

all PADS, LATCHES, RAMS and FFS in the macro called tester will be in a group called coverall. The applicable

Constraints Guide will contain a complete listing of the predefined groups.

Less Is More

In general, the fewer constraints, the better. Complex constraints can often cause more problems than they solve. In addition, some paths or nets may be noncritical, and you may wish to declare that no constraints should be applied to these nets.

TIG (timing-ignore) constraints are used to remove things we don't care about or to remove constraints from a false path. Here is a common TIG:

```
NET "rst" TIG;
```

This tells the tools that you do not need to constrain this path. It is important to

spell this out so that the tools do not work to meet timing on paths you do not care about. Setting timing to ignore such paths will also reduce tool run-times and may improve the quality of the timing on the paths you do care about.

You can also use TIG with FROM:TO constraints, as in the following :

```
TIMESPEC TS_my_fromto = FROM  
"my_to_grp" TO "FFS" TIG;
```

Xilinx has a number of great resources on timing constraints, the most notable of which I've cited in the references below. Please feel free to contact me if you have any further questions. I invite you all to participate in Xilinx's community forums, which offer a plethora of

insights and answers to some of FPGA design's most vexing questions. ☺

References

Constraints Guide: Constraint syntax for UCF, PCF, HDL, http://www.xilinx.com/support/documentation/sw_manuals/xilinx11/cgd.pdf

Timing Constraints User Guide: Conceptual information on how to constrain a design, http://www.xilinx.com/support/documentation/sw_manuals/xilinx11/ug612.pdf

Timing Analyzer Help: General information on how to use Xilinx Timing Analyzer, http://www.xilinx.com/support/documentation/sw_manuals/help/iseguide/mergedProjects/timingan/timingan.htm

Make your FPGA verification and debugging process



FLY!

Whether you're building complex FPGA-based systems or ASIC/ASSP prototypes, you're probably experiencing one or more of these issues:



Weeks of painful debugging
in the system lab



Hundreds of long synthesis + place
& route iterations



Slow simulation/multi-day
regression tests



IP or RTL block mismatches between
simulation and system

The GateRocket debug and verification solution cuts debug time in half by making simulation fast, accurate and eliminating many of the synthesis-place-route cycles in the lab. Problems that would otherwise slip through to the system lab are exposed in simulation, on the desktop, where they are easily and rapidly fixed. Learn why industry leading companies like Qualcomm have adopted this solution by visiting GateRocket.



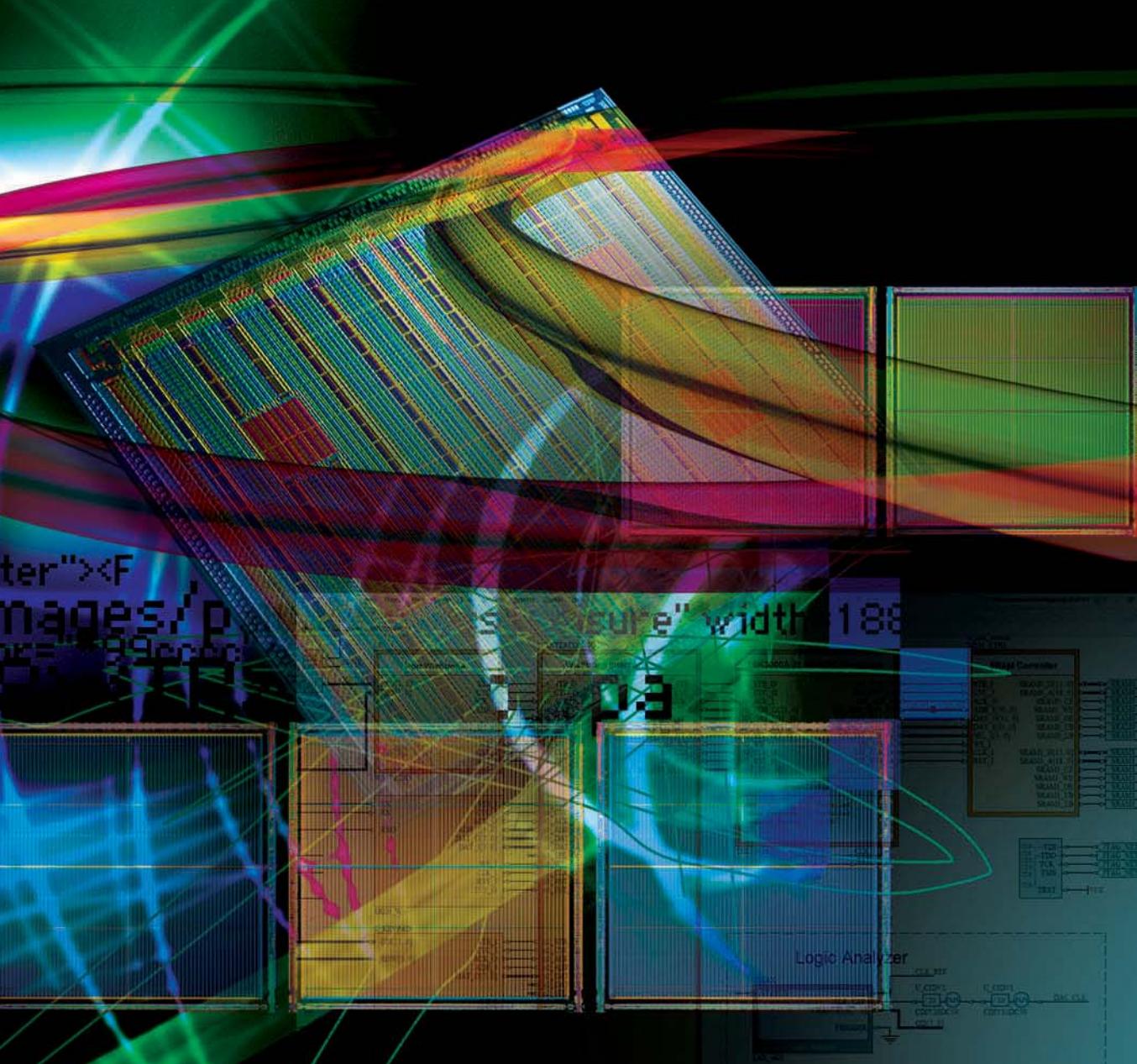
Phone: +1 (781) 908-0082, Option 3

Email: RocketDrive@GateRocket.com

Visit: <http://www.GateRocket.com>

Simplifying Metastability with IDDR

Use the flip-flop chain that's part of the ILOGIC block in Xilinx FPGAs to limit metastability events in your designs.



by Primitivo Matas Sanz
Technology Expert
Telefonica I+D, Madrid, Spain
primi@tid.es

If you've ever used an FPGA in an asynchronous system with multiple clocks, or in one that uses a clock with a frequency or phase that differs from the one your FPGA uses, your design can encounter metastability problems. Unfortunately, if your design falls into one of these system scenarios, there's no way to completely eliminate metastability, but there are several methods you can employ to reduce the likelihood your system will encounter it.

Let's take a closer look at what causes metastability and then examine some methods we can employ to attack it.

What is Metastability?

In synchronous-logic digital devices such as FPGAs, each device's register cell has pre-defined signal-timing requirements that allow the device to correctly capture data and in turn generate a reliable output signal. When another device sends data to the FPGA, the FPGA's input register must be stable for a minimum setup time before the clock edge and also for a minimum hold time after the clock edge to receive the signal properly and in its entirety.

After the specified delay, the register output can then send the signal to the rest of the FPGA. But if a signal transition violates the times specified, the output register may go into the so-called metastable state, in which the register output will hover at a value between the high and low states for an indeterminate period. The result is to delay the stable output state beyond the time specified for the register, a condition that can cause a slight delay in performance or a logical-behavior side effect.

Addressing the Issue

Usually, to connect an FPGA to another digital device that has a different clock domain, we need to add a synchronization stage to the input section of the FPGA and make the first register in the FPGA clock domain act as a synchronization register. To do this, we can use either a

sequence of registers or a synchronization register chain in the FPGA device's input stage. This chain allows additional time for a potentially metastable signal to resolve before the input registers pass the signal to other regions of the FPGA. The metastable settling time is typically much less than a clock cycle, so a delay of even half the clock period may reduce the probability of a metastable value by many orders of magnitude.

To reduce the chances of encountering a metastability problem, the sequence of registers (wired as shift registers) that we implement in a design must meet the following criteria:

- All registers must be clocked by the same clock or by the same phase-related clocks.
- Each register in the chain must fan out only to the next register.

Because we cannot completely eliminate metastability problems, we must still account for them. To do this, the design community uses the term mean time between failures (MTBF) to estimate the average time between instances when the problem could cause a failure. A higher MTBF indicates a more robust design. A "failure," in this case, is a failure to resolve metastability and not an actual system failure per se.

To see how metastability is measured, read the Xilinx Application Note

XAPP094, at http://www.xilinx.com/support/documentation/application_notes/xapp094.pdf.

We can calculate the MTBF for one register with the following formula:

$$MTBF = \frac{e^{t_{MET}/C_2}}{C_1 \cdot f_{CLK} \cdot f_{DATA}}$$

In this instance, C_1 and C_2 are constants related to the register technology and t_{MET} is the metastability settling time.

We can determine the overall MTBF by looking at the MTBF of each register. The failure rate for a synchronizer is $1/MTBF$, and we can calculate the failure rate for the entire design by adding the failure rates for each synchronizer, as follows:

$$\text{failure_rate}_{\text{design}} = \frac{1}{MTBF_{\text{design}}} = \sum_{i=1}^{\text{number of chains}} \frac{1}{MTBF_i}$$

Given this formula, it's clear that there are ways to get a better MTBF. We can, for example, improve the architecture of our register cells; optimize the design to increase the t_{MET} in the synchronization registers; or even increase the number of registers in the chain.

High-Level Code and Placement Result

When we find an input signal with a potential metastability problem, we can address the issue by simply creating a reg-

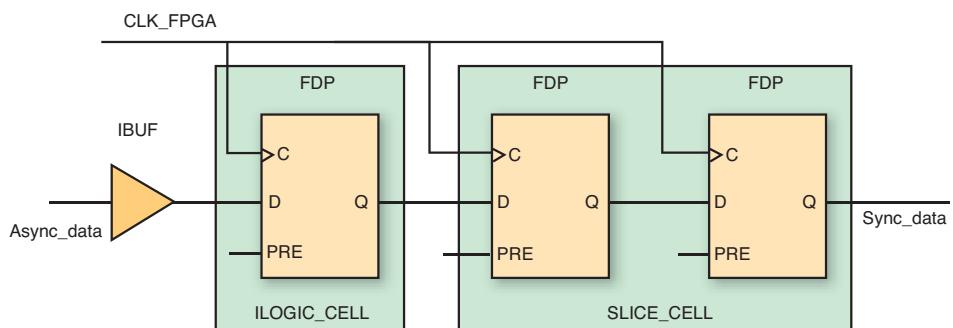


Figure 1 – Synchronizer chain showing placement by default

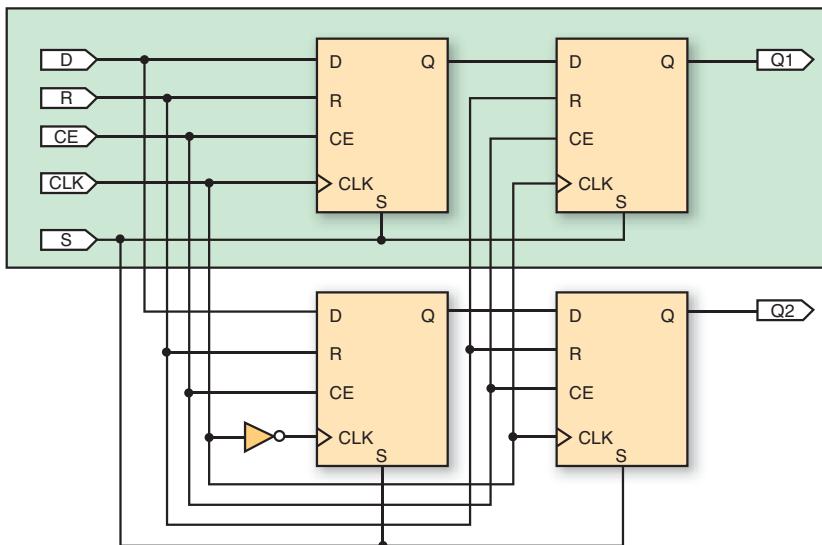


Figure 2 – Input DDR in SAME_EDGE_PIPELINED mode

One of the modes of operation for this primitive is called SAME_EDGE_PIPELINED. Figure 2 shows the input DDR registers and the signals involved in using this mode. The rectangle in green shows a perfect sequence of registers we can use to resolve the metastability problem. What's more, using this IDDR method has an additional advantage—namely, we can use two or three times as many main clocks without introducing any latency problems into the design.

A Bit of Code Is All it Takes

In the *Virtex-4 User Guide*, pages 328-329, you can find examples that illustrate the instantiation of the IDDR primitive in VHDL and Verilog. Here is a typical example in Verilog:

```
defparam IDDR_INT2.DDR_CLK_EDGE =
"SAME_EDGE_PIPELINED";

defparam IDDR_INT2.INIT_Q1 = 1'b1;

defparam IDDR_INT2.INIT_Q2 = 1'b1;

defparam IDDR_INT2.SRTYPE = "SYNC";

IDDR_IDDR_INT2( .Q1(sync_data),
.Q2(signal_noload), .C(CLK_2X),
.CE(1'b1), .D(async_data), .R(), .S());
```

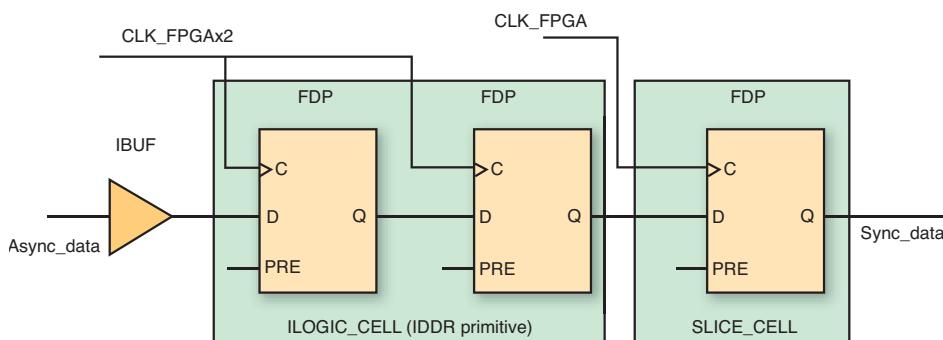


Figure 3 – Synchronizer chain showing placement with IDDR

ister chain with the same phase-related clock. When we do this, we will come up with a circuit that resembles the one we show in Figure 1.

In this figure, we placed the register chain into two cells: the first is an ILOGIC cell while the other two registers are within a SLICE cell (we select a chain with three registers and the same clock). This is one quick and fairly simple way to mitigate metastability issues, but there are others that also optimize performance.

IDDR Method Using Xilinx ILOGIC Blocks

In the Virtex®-4 and Virtex-5 FPGAs, Xilinx® places its ILOGIC blocks directly behind the I/O drivers and receivers. The blocks include four storage-element registers and a programmable absolute-delay element.

The Virtex-4 and Virtex-5 devices use those four registers to implement input double-data-rate (IDDR) registers, a feature designers can access only by instantiating the IDDR primitive. We can use this to our advantage.

In Figure 3, we can see the new placement results. Using this methodology, we've placed the register chain into two cells: the first two registers are contained within an ILOGIC cell and the other is in a SLICE cell (here, we select a chain with three registers and two different clocks, one that is twice as fast as the other).

Overall, metastability issues can be an inconvenience in your design, but by employing a few quick and easy fixes, including using the IDDR primitive in a new way, you can drastically reduce the chances your design will encounter metastability issues. By making use of these methods as you are creating the design, rather than afterwards, you can craft metastability-resilient architectures optimized upfront for area, performance and cost. 

Xilinx Tool & IP Updates

Xilinx continues to improve products and IP in the ISE® Design Suite. Here is the latest quarterly release for Xilinx® design products and IP as of July 2010. Quarterly releases, which offer significant enhancements and new features to the ISE Design Suite, are now full installations, designed to coexist with previous quarterly or major Xilinx releases. By installing the latest Xilinx release, you are taking advantage of an easy way to ensure the best results for your design.

The latest release of ISE Design Suite is always available from the Xilinx Download Center at www.xilinx.com/download. For more information or to download a free 30-day evaluation of the ISE Design Suite, visit www.xilinx.com/ise. Also, look for new Xilinx tools and IP as well as new IP, tools and development boards from Xilinx partners in the Tools of Xcellence section of this issue.

ISE Design Suite: Logic Edition

Ultimate productivity for FPGA logic design

Latest version number: 12.2

Date of latest release: July 2010

Previous release: 12.1

URL to download the latest release:
www.xilinx.com/download

Revision highlights:

This latest release of the ISE Design Suite expands support for both the Spartan®-6 (XA and XQ devices) and Virtex®-6 (XQ devices) FPGA families. In addition, the ISE Design Suite: Logic Edition 12 boasts 2X faster run-times for Xilinx Synthesis Technology (XST) and a 1.3X speedup for implantation of large designs, along with 15 to 20 percent faster implementation run-times using multithreading.

Project Navigator: Xilinx has made improvements to design hierarchy parsing, such as upfront HDL syntax error checking, user control for enabling or disabling hierarchy reparsing and full support for, and automatic detection of, “include” files, along with process dependency and source management. The tool also supports netlist sources (EDIF, NGC/NGO) as submodules in the design hierarchy, including process dependency and source

management. Project Navigator also now includes “find” support in the design hierarchy view, which enables you to search for sources in the design hierarchy based on file name, module name or instance name, and to search for missing modules. Process status improvements include a new process monitor, improved status indicator behavior and the ability to run downstream processes based on the presence of necessary files rather than the error status of previous steps. Finally, the Design Summary has a new System Settings report that displays environment settings and process properties used during design implementation.

Power optimization: Intelligent clock gating, available for Virtex-6 (version 12.1) and Spartan-6 (12.2) devices, minimizes logic toggling to reduce dynamic power consumption.

Partial reconfiguration: PR enables dynamic design modification of a configured FPGA for Virtex-6. The ISE software (12.2) uses partition technology to define and implement static and reconfigurable regions of the device. Note: This software feature requires an additional license code.

FPGA Editor: This tool, which has approximately 40 percent smaller memory foot-

print for large devices compared with the 11.1 release, boasts faster loading of device graphics. The List Window has copy, paste and cut keyboard shortcuts that can be used in the Name Filter.

ChipScope™: This version adds support for continuous trigger with multiple ILA cores. Analyzer adds a repetitive-run trigger option to monitor repetitive events without having to manually rearm the trigger.

Device programming: ChipScope Pro and iMPACT now support JTAG cables sold by third-party partners ByteTools and Digilent; new flash device support has also been added to iMPACT. The following third-party devices may be programmed: Numonyx N25Q, Numonyx P30 (now up to 1Gb), Winbond W25Q, Spansion S25FLP and Spansion S29GLP.

ISim: Improved integration and interoperability deliver the ability to simulate embedded designs, with integration into XPS and Project Navigator. The tool offers complete OS support, including native support for 64-bit Windows, and ensures ease of use for batch-mode user (you can programmatically configure the waveform via Tcl). Memory viewing and debug are easier thanks to a new memory editor and viewers. The tool automatically parses the design and identifies memory elements. Waveform enhancements include the ability to adjust the time scale automatically for optimal viewing, and to override HDL stimulus with user-defined values.

PlanAhead™: A new, simpler and intuitive RTL-to-bitstream pushbutton task-based flow takes you through three main steps: synthesis, implementation, program and debugging. Design preservation and partial reconfiguration are supported for the command-line tools and the standalone version of the PlanAhead software.

SmartXplorer: SmartXplorer now supports synthesis when using the command line. The new custom file format that delivers this feature will also let you specify synthesis and implementation strategies simultaneously. You can display area information in the SmartXplorer report table by using the –area_report option, and run the power analyzer and display power information in the report table by using the –pwo option (command-line mode only). Also, you can use power as an additional best-strategy selection criterion if you are using the –pwo option, and control TRCE by using the –to option. This option lets you generate verbose TRCE reports during SmartXplorer runs.

XPower Analyzer: This version has reorganized several views and consolidated related data in an organization similar to that of XPE. The hierarchy view adds resource utilization for each hierarchy level (LUTs, FFs), while additional statistical data appears in the tool-tip window. In clock domain view, you can edit the frequency for all clocks and see added details regarding clock tree topology. It allows you to specify custom off-chip I/O termination to calculate off-chip power. An added Confidence Level view assists you in obtaining realistic power data.

XPower Estimator: Readability is significantly improved with a new color scheme and improved presentation of data. The tool provides power distribution by resource type on a summary sheet and reports total power supplied to I/O termination (off-chip power). It allows you to specify custom off-chip I/O termination to calculate off-chip power.

XST: This version adds inference support for asymmetric-port Block RAM. Power optimization dedicated to BRAM optimization is now implemented for Virtex-6 and Spartan-6 devices. For more information see the –power option and RAM style (RAM_STYLE) constraint. A new automax value for the Use DSP Block (USE_DSP48) constraint instructs XST to maximize utilization of DSP

resources within the limits of available resources on the selected device and lets you implement more logic on DSP blocks than can typically be achieved with the auto value. This can be particularly useful when a tightly packed device is your primary concern. A new Shift Register Minimum Size (SHREG_MIN_SIZE) option allows you to control the minimum size of shift registers that are inferred and implemented using SRL-type resources. While the default minimal size is 2, you may need to raise that threshold for more efficient resource placement and circuit performance.

ISE Design Suite: Embedded Edition

An integrated software solution for designing embedded processing systems

Latest version number: 12.2
Date of latest release: July 2010
Previous release: 12.1
URL to download the latest patch:
www.xilinx.com/download

Revision highlights:

All ISE Design Suite Editions include the enhancements listed above for the ISE Design Suite: Logic Edition. The following is the list of enhancements specific to the Embedded Edition.

Xilinx Platform Studio: XPS now supports the ISim Simulator, to simulate embedded designs using the included ISE HDL simulator. Native support for 64-bit Windows NT is included. XPS has the ability to “Export Hardware Design to SDK” from ISE Project Navigator—a fast, transparent setup of hardware definitions for software design. Simplified Project Options Dialog delivers easier startup of embedded projects, while a new Cygwin version allows multiple installations of Cygwin to exist on one system.

Software Design Kit: Xilinx has upgraded the SDK to the latest Eclipse version (v3.5.1), delivering all the most recent Eclipse standards. Also, the SDK is updated to the latest C/C++ Development Tools release (CDT v6.0.1), to utilize the industry’s latest C/C++

software design support. Simplified flows for creating the initial C application will get you through your software design flow faster.

MicroBlaze™ Soft Processor: A new configuration wizard, with multiple starting MicroBlaze preconfiguration options, guides you through processor setup. The branch target buffer stores recently taken branch locations, speeding up future execution, while branch prediction keeps the instruction pipeline full, maximizing processor performance. Victim cache stores recently flushed cache lines locally for faster processing.

Embedded IP: Spartan-6 and Virtex-6 production support (except the MPMC Virtex-6, which remains preproduction) means you can design to the newest Xilinx FPGA devices. PLBv34 and OPB-based cores have been removed, providing faster download and a smaller IP installation footprint.

ISE Design Suite: DSP Edition

Flows and IP tailored to the needs of algorithm, system and hardware developers

Latest version number: 12.2
Date of latest release: July 2010
Previous release: 12.1
URL to download the latest patch:
www.xilinx.com/download

Revision highlights:

All ISE Design Suite Editions include the enhancements listed above for the ISE Design Suite: Logic Edition. The following is the list of enhancements specific to the DSP Edition.

64-bit OS support: Native support for the 64-bit version of Windows XP and Windows Vista 64-bit SPI allows System Generator to access the additional memory possible with these operating systems.

Hardware co-simulation improvements: System Generator now supports Ethernet point-to-point hardware co-simulation for the Spartan-6 FPGA SP601 and SP605 platforms. System Generator provides support for more than one JTAG cable to the computer, so that customers can have more than one hardware co-simulation token in a

design. For increased flexibility, it also supports custom JTAG cables, like the cable used to connect ChipScope.

Enhanced integration with ChipScope: A JTAG hardware co-simulation design can include a ChipScope block to provide additional visibility during debug. You can import repetitive-trigger data with the xlLoadChipScopeData utility function. Some situations require the repetitive-trigger mode to capture the necessary data during debug.

Xilinx IP Updates

Name of IP: ISE IP Update 12.2

Type of IP: All

Targeted application:

Xilinx develops IP cores and partners with third-party IP providers to decrease customer time-to-market. The powerful combination of Xilinx FPGAs with IP cores provides functionality and performance similar to ASSPs, but with flexibility not possible with ASSPs.

Latest version number: 12.2

Date of latest release: July, 2010

URL to access the latest version:

www.xilinx.com/download

Informational URL:

www.xilinx.com/ipcenter

New Cores in ISE Design Suite 12

Video and image processing:

- Image Characterization v1.0 — Calculates important statistical data for video input streams. Characterization is an important processing block for many applications, including facial recognition and object detection.

Wired communications:

- SGMII over LVDS — Provides designers with a GMII-to-SGMII bridge function using LVDS (SelectIO™) instead of transceivers for chip-to-chip applications on Virtex-6 family devices. This new feature is an addition to the

Ethernet 1000BASE-X PCS/PMA or SGMII IP LogiCORE™ IP v10.5. The core is included at no additional charge with the ISE Design Suite software.

Wireless communications:

- 3GPP LTE RACH Detector v1.0 — Provides designers with an LTE RACH detecting block, which decodes P-RACH data encoded according the 3GPP TS 36.211 v8.6.0, Physical Channels and Modulation specification.
- DUC/DDC Compiler — Implements high-performance, optimized digital up- and downconverter modules for use in wireless base stations and other suitable applications. In addition to a wide range of parameter options, resource trade-off options enable you to tailor the core to specific design requirements. The core is included at no additional charge with the ISE Design Suite software.
- 3GPP LTE Channel Estimator — Implements channel estimation to support decoding of the Physical Uplink Shared Channel (PUSCH) in 3GPP LTE eNodeB applications as defined in the 3GPP TS 36.211 specification. Includes support for single-input, single-output (SISO); single-input, multiple-output (SIMO); and multiuser multiple-input, multiple-output (MIMO) communication modes.

- 3GPP LTE FFT — Implements all transform lengths required by the 3GPP LTE specification, including the 1,536-point transform for 15-MHz bandwidth support.

New CORE Generator™ features:

- Enhanced assistance with design migration, including improved messaging for cores and core versions that have been removed from 12.1. A new warning message advises users about newer IP versions that are available along with information on core versions in the current project that can be automatically

upgraded. The project IP panel now displays missing IP as “upgradable,” “removed” or “unavailable.”

- IP catalog enhancements mean the IP core now displays the life cycle status of preproduction and production devices on a per-family basis.
- Xilinx has added the capability for automated core upgrade to the latest version for the following IP cores: Block Memory Generator v4.1, CIC Compiler v2.0, Clocking Wizard v1.5, Fast Fourier Transform v7.1, FIFO Generator v6.1, SelectIO Interface Wizard v1.3 and System Monitor Wizard v2.0

IP Updates – Highlights

- Memory IP: Block Memory Generator v4.2 — Designers can use a new “write first” mode for single-dual-port (SDP) memory type (Virtex-6 only) instead of “read first” for SDP BRAM when the read and write ports are clocked by different clocks. This reduces BRAM utilization in SDP mode. Block Memory Generator also now supports soft hamming error correction for SDP BRAM configurations for data widths < 64 bits (Virtex-6 and Spartan-6 only).
- FIFO Generator v6.2: This IP now supports “write first” mode for SDP BRAM-based FIFO configurations for reduced BRAM utilization in SDP mode.
- SelectIO Wizard v1.4: Added support for Virtex-6 FPGAs
- Video IP: The Defective Pixel Correction, Gamma Correction, Color Correction Matrix and Color Filter Array Interpolation cores have increased maximum supported sensor resolution to 4K x 4K. They now support Spartan-6 and Virtex-6 devices as well as 32- and 64-bit Linux.

A comprehensive listing of cores that have been updated in this release is available at www.xilinx.com/ipcenter/coregen/12_2_datasheets.htm.

At the Cusp of the Programmable Imperative

Xilinx has altered its training strategy to keep pace with engineers' interest in FPGA design.

Over the past two years, Xilinx has striven to take full advantage of what Xilinx CEO Moshe Gavrielov calls "the programmable imperative"—the confluence of market forces driving hardware engineers away from ASICs and ASSPs and toward FPGAs.

As process nodes shrink, the cost of designing and developing ASIC and ASSP devices increases exponentially to the point where, at the 40/45- and 28-nanometer nodes, the return on investment becomes untenable. On the other hand, FPGAs have taken advantage of these smaller process nodes to incorporate more functions, making the programmable devices an attractive alternative. And with the FPGA's inherent flexibility and reduced NRE costs, engineers are finding FPGAs more compelling than ever before.

As the market for FPGAs has grown, there has been increasing pressure to help engineers make the transition from ASIC and ASSP design to FPGA design. In response, Xilinx® has re-engineered its customer training program to make it more accessible to a wider and more diverse audience.

In the past, Xilinx customer training lagged new silicon or software releases by six months. This generally aligned to the time it took for traditional FPGA users to adopt new products. Over the last two years, however, we have found increasing interest in new Xilinx products, especially among engineers new to FPGA design, coincident with product launch.

To address this new group, Xilinx has accelerated training updates so that our classes reflect the latest and greatest technology on the market today. We update classes at product launch or within 30 days of launch. All courses are now updated to the ISE® 12.1 design tools.

Finding the Right Mix

Learning how to design with an FPGA is like learning to fly. Success takes a mix of theoretical knowledge and hands-on skill. A pilot needs to understand how weather can affect the flight

dynamics of an airplane, but must also gain experience controlling the aircraft in those weather conditions. Similarly, an engineer implementing an FPGA design needs to understand the different tools, IP and design techniques to steer a successful FPGA design through the ever-changing weather of evolving standards and system change requirements.

For the expanding market of FPGA engineers, Xilinx had to decide how to strike a balance between the practical and theoretical training approaches. Because engineers tend to learn by doing, Xilinx has implemented a number of changes to its training curriculum, including:

- Focusing classroom learning on hands-on labs. On average, engineers spend more than 50 percent of all class time in labs.
- Modularizing content so instructors can shape the training to meet each class' needs. Some groups may need more theoretical training, others more hands-on experience.
- Providing labs in multiple formats to accommodate each learner's unique style. This includes labs that provide high-level instructions for engineers who want to explore the tools on their own, as well as step-by-step instructions for those who want to learn the best method for solving problems.

Expanding the Reach of Training

To reach out to this ever-expanding pool of FPGA engineers, Xilinx has stepped up its efforts to connect learners with expert trainers regardless of where they reside. Xilinx has partnered with nearly 30 Authorized Training Providers, expert FPGA engineers who enjoy sharing their knowledge and expertise with others. Many of these training providers are also consultants in digital signal processing, embedded processing and connectivity designs.

Over the past two years, Xilinx has focused on enhancing the depth and breadth of its Authorized Training Providers roster. In North America alone, Xilinx has recently added five new training providers to the team.

The Response?

The response has been overwhelming. Learners, on average, rate Xilinx customer training classes at 8.6 out of 10, with a median score of 9. In addition, learners tell us their FPGA knowledge doubled as a result of the training.

Xilinx FAEs who engage regularly with customers clearly see the results. Says FAE Don Schaeffer, "Engineers that have taken classes are more likely to make use of advanced features, IP and design techniques, which dramatically improves the quality of their products and significantly reduces their time-to-market."

Adds Mike Cole, "Customers really enjoyed the Embedded Systems Design class and stated that they feel the class will save them at least four weeks of development time. The instructor was outstanding and the course content gave them confidence in creating custom peripherals and utilizing advanced cores in EDK. Through the training the customer was able to ask higher-level questions, utilize advanced techniques and ultimately have a better product and get to market more quickly."

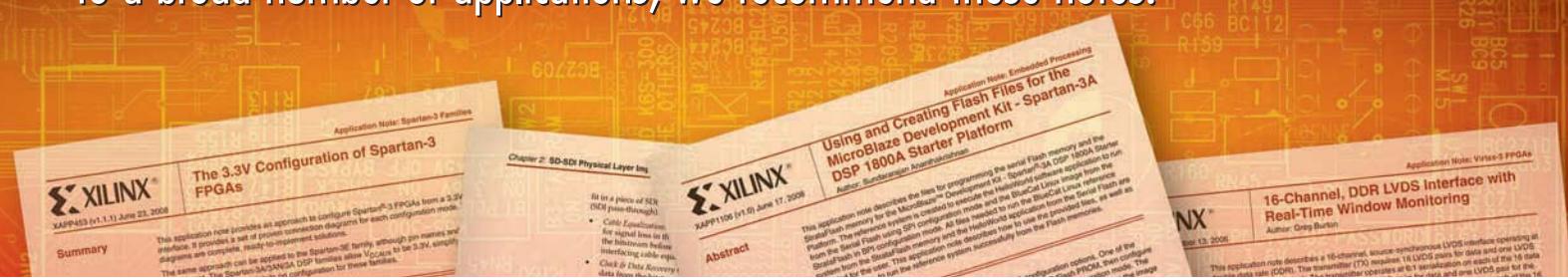
Additionally, Xilinx instructor Bill Kafig has this to say: "My favorite quote from a student taking the Essentials of FPGA Design and Designing for Performance classes—'I learned in three days what it would have taken me three years to learn!'"

Space is limited, so sign up for a course today by visiting <http://www.xilinx.com/training/worldwide-schedule.htm>. For more information on the Authorized Training Providers, see <http://www.xilinx.com/training/>.

AUTHORIZED TRAINING PARTNER	CONTACT	COUNTRY/REGION(S) SUPPORTED
Xilinx Training Worldwide	www.xilinx.com/training	Worldwide
AMERICAS	registrar@xilinx.com	
Anacom Eletrônica	www.anacom.com.br	Brazil
Bottom Line Technologies	www.bltinc.com	Delaware, District of Columbia, Maryland, New Jersey, New York, Eastern Pennsylvania, Virginia
Doulos	www.doulos.com/xilinxNC	Northern California
Faster Technology	www.fastertechnology.com	Arkansas, Colorado, Louisiana, Montana, Oklahoma, Texas, Utah, Wyoming
Hardent	www.hardent.com	Alabama, Connecticut, Eastern Canada, Florida, Georgia, Maine, Massachusetts, Mississippi, New Hampshire, North Carolina, Rhode Island, South Carolina, Tennessee, Vermont
North Pole Engineering	www.npe-inc.com	Illinois, Iowa, Kansas, Minnesota, Missouri, Nebraska, North Dakota, South Dakota, Wisconsin
Technically Speaking	www.technically-speaking.com	Arizona, British Columbia, Southern California, Idaho, New Mexico, Nevada, Oregon, Washington
Vai Logic	www.vailogic.com	Indiana, Kentucky, Michigan, Ohio, Western Pennsylvania, West Virginia
EUROPE, MIDDLE EAST & AFRICA (EMEA)	eurotraining@xilinx.com	
Arcobel Embedded Solutions	www.arcobel.nl	The Netherlands, Belgium, Luxembourg
Bitsim AB	www.bitsim.com/education	Sweden, Norway, Denmark, Finland, Lithuania, Latvia, Estonia
Doulos Ltd.	www.doulos.com/xilinx	United Kingdom, Ireland
Inline Group	www.plis.ru	Moscow Region
Logtel Computer Communications	www.logtel.com	Israel, Turkey
Magnetic Digital Systems	www.magneticgroup.ru	Urals Region
Mindway	www.mindway-design.com	Italy
Multi Video Designs (MVD)	www.mvd-fpga.com	France, Spain, Portugal, Switzerland, Mexico, Brazil, Argentina
Pulsar Ltd.	pulsar.co.ua/en/index	Ukraine
Programmable Logic Competence Center (PLC2)	www.plc2.de	Germany, Switzerland, Poland, Hungary, Czech Republic, Slovakia, Slovenia, Greece, Cyprus, Turkey, Russia
SO-Logic Consulting	www.so-logic.co.at	Austria, Brazil, Czech Republic, Hungary, Slovakia, Slovenia
ASIA PACIFIC	education_ap@xilinx.com	
Active Media Innovation	www.activemedia.com.sg	Malaysia, Singapore, Thailand
Black Box Consulting	www.blackboxconsulting.com.au	Australia, New Zealand
E-elements	www.e-elements.com	China, Hong Kong, Taiwan
Libertron	www.libertron.com	Korea
OE-Galaxy	edu-electronic@oegalaxy.com.vn	Vietnam
Sandeepani Programmable Solutions	www.sandeepani-vlsi.com	India
Symmid	www.symmid.com	Malaysia
WeDu Solution	www.wedusolution.com	Korea
JAPAN	education_kk@xilinx.com	
Avnet Japan	www.jp.avnet.com	Japan
Paltek	www.paltek.co.jp	Japan
Shinko Shoji	xilinx.shinko-sj.co.jp	Japan
Tokyo Electron Device	ppg.teldevice.co.jp	Japan

Application Notes

If you want to do a bit more reading about how our FPGAs lend themselves to a broad number of applications, we recommend these notes.



XAPP459: Eliminating I/O Coupling Effects when Interfacing Large-Swing Single-Ended Signals to User I/O Pins on Spartan-3 Families

http://www.xilinx.com/support/documentation/application_notes/xapp459.pdf

Xilinx® Spartan®-3, Spartan-3E and Extended Spartan-3A devices support an exceptionally robust and flexible I/O feature set that easily meets the signaling requirements of most applications. It is possible to program user I/O pins of these families to handle many single-ended signal standards.

The standard single-ended signaling voltage levels are 1.2 V, 1.5 V, 1.8 V, 2.5 V and 3.3 V. But in a number of applications, it is desirable to receive signals with a greater voltage swing than user I/O pins ordinarily permit. The most common use case involves receiving 5-V signals on user I/O pins that are powered for use with one of the standard single-ended signaling levels.

This application note by Eric Crabill describes ways to receive the resulting “large-swing signals” by design. In one solution (and in the general case of severe positive or negative overshoot), parasitic leakage current between user I/O in differential-pin pairs might occur, even though the user I/O pins are configured with single-ended I/O standards. The application note addresses the parasitic leakage current behavior.

XAPP1075: Implementing Triple-Rate SDI with Virtex-6 FPGA GTX Transceivers

http://www.xilinx.com/support/documentation/application_notes/xapp1075_V6GTX_TripleRateSDI.pdf

Professional broadcast video equipment makes wide use of the triple-rate serial digital interface (SDI) supporting the SMPTE SD-SDI, HD-SDI and 3G-SDI standards. In broadcast studios and video production centers, SDI interfaces carry uncompressed digital video along with embedded ancillary data, such as multiple audio channels.

Xilinx Virtex®-6 FPGA GTX transceivers are well-suited for implementing triple-rate SDI receivers and transmitters, providing a high

degree of performance and reliability while occupying a relatively small amount of FPGA logic resources. In this application note, John Snow describes how to implement these triple-rate SDI interfaces.

Doing so requires only two reference clock frequencies to support all SDI modes: 148.5 MHz for SD-SDI at 270 Mbps, HD-SDI at 1.485 Gbps and 3G-SDI at 2.97 Gbps; and 148.5/1.001 MHz for HD-SDI at 1.485/1.001 Gbps and 3G-SDI at 2.97/1.001 Gbps. The transceiver transmits preformatted dual-link HD-SDI streams via either dual-link HD-SDI or 3G-SDI Level B formats. With the addition of a 3G-SDI Level A mapping module, it supports all 3G-SDI Level A-compatible video formats. In addition, the transceiver directly supports transmission of two independent HD-SDI streams in the 3G-SDI Level B mode. Only a single global clock is required for the transmitter. No mixed-mode clock managers (MMCMs) are required.

XAPP496: Creating Wider Memory Interfaces Using Multiple Spartan-6 FPGA Memory Controller Blocks

http://www.xilinx.com/support/documentation/application_notes/xapp496.pdf

The Memory Controller Block (MCB) is a dedicated embedded multiport memory controller that greatly simplifies the task of interfacing Spartan-6 devices to DDR3, DDR2, DDR and LPDDR memories. Spartan-6 devices contain two to four MCBs, each of which can implement a single-component interface to a 4-bit, 8-bit or 16-bit memory. However, some applications with higher memory bandwidth or density requirements benefit from using memory interfaces wider than 16 bits. This application note by Derek Curd describes how to merge the operation of two or more MCBs to implement effective 32-bit or wider memory interfaces.

Both MCBs must be in a single-port configuration mode. Each MCB still operates at full performance (up to 800 Mbits/second), allowing the user application to realize the full benefit of using these dedicated embedded memory controllers for wider interfaces. The

author has verified the associated reference design in hardware and analyzed it for both performance and device utilization. However, the design does not support merging MCBs that are configured in the multiport configuration mode.

XAPP1073: NSEU Mitigation in Avionics Applications

http://www.xilinx.com/support/documentation/application_notes/xapp1073_NSEU_Mitigation_Avionics.pdf

Neutron-induced single-event upset (NSEU) is a known phenomenon in the memory structures of modern ICs used in terrestrial applications. With current and next-generation aircraft operating at altitudes of 40,000 feet and higher, the increased atmospheric neutron flux raises the likelihood of this phenomenon by several orders of magnitude, with the potential to affect flight safety. The decreasing feature size of memory structures combined with the growth in memory size means that systems are becoming ever more susceptible to NSEUs.

SRAM-based FPGAs pose a unique challenge for avionics manufacturers, because an FPGA's functionality depends on the integrity of its configuration memory. It is vital for FPGA designers to achieve NSEU hardness for critical avionics systems through a combination of soft and hard mitigation techniques.

This application note by Ching Hu and Suhail Zain provides background on NSEUs in SRAM-based FPGAs, mitigation techniques (with a focus on configuration memory) suggested by Xilinx and an overview of how to calculate projected failures-in-time (FIT) rates at altitude.

While SRAM-based FPGAs have an additional susceptibility over other programmable technologies due to the volatility of their configuration memories, Xilinx has developed a number of mitigation techniques and structures. They include SRAM cells built to logic-design rules (not to memory-design rules), built-in ECC structures in Block RAM, and SEU detection and correction structures built into hardware. All memory elements are susceptible to NSEUs, but with proper mitigation techniques, SRAM-based FPGAs provide avionics designers with a wide range of possible solutions.

XAPP879: PLL Dynamic Reconfiguration

http://www.xilinx.com/support/documentation/application_notes/xapp879.pdf

This application note by Karl Kurbjun and Carl Ribbing provides a method to dynamically change the clock output frequency, phase shift and duty cycle of the Spartan-6 FPGA phase-locked loop (PLL) through its Dynamic Reconfiguration Port (DRP). After explaining the behavior of the internal DRP control registers, the authors provide a reference design that uses a state machine to drive the DRP so as to ensure that the registers are controlled in the correct sequence.

The PLL used in conjunction with the DRP interface is recommended for advanced users when the basic PLL functionality is not sufficient. The DCM_CLKGEN primitive can be a useful alternative.

The reference design, which supports two reconfiguration state addresses, can be extended to support additional states. Each state does a full reconfiguration of the PLL so that most parameters can

be changed. Its modular nature means you can use the design as a full solution for DRP or easily extend it to support additional reconfiguration states. The design uses minimal Spartan-6 FPGA resources, consuming only 25 slices.

However, if designers need postconfiguration cyclic redundancy check (CRC) functionality in their design, they cannot use the PLL DRP port to dynamically reconfigure the PLL. Doing so breaks the functionality of postconfiguration CRC.

XAPP1146: Embedded Platform Software and Hardware

In-the-Field Upgrade Using Linux

http://www.xilinx.com/support/documentation/application_notes/xapp1146.pdf

New features and bug fixes often necessitate upgrading flash images to replace the existing FPGA bitstream, boot loader, Linux kernel or file system. This application note describes an in-the-field upgrade of the Spartan-6 FPGA bitstream, Linux kernel and loader flash images, using the presently running Linux kernel. Upgrade files are obtained from a CompactFlash storage device or over the network from an FTP server. Author Brian Hill includes one reference design built for the Xilinx SP605 Rev C board.

XAPP498: Source Control and Team-Based Design in System Generator

http://www.xilinx.com/support/documentation/application_notes/xapp498.pdf

This application note by Douang Phanthavong provides an overview of how to perform source version control and team-based design using the System Generator tool. Designers can accomplish these tasks using the version control features native to the MATLAB® Simulink® software environment, or with an external source control system. While this application note focuses on Subversion, a well-known, free, open-source control system, other version control software such as CVS, MS Source Safe and Clear Case can also be used—depending on the design environment.

Collaborative development allows developers who are physically dispersed to concurrently and collaboratively design, test, debug and document the same design. Team-based design in MATLAB/Simulink requires coordination of modeling activities between team members. If properly managed, dozens of geographically dispersed developers can effectively share their work in a safe, secure and productive design environment. However, if not managed well, dealing with many design versions and their dependencies can lead to severe loss in productivity and reduced confidence in product quality. Version control is the key to managing an organization's MATLAB/Simulink designs.

This application note provides the basic knowledge required to manage model versions using Simulink's native features. It also shows how to use source control systems such as Subversion internally and externally to the MATLAB/Simulink software environment. Users find out how to graphically compare and merge models using the SimDiff and SimMerge tools and how to use these tools with source control systems such as TortoiseSVN. ●

Tools of Xcellence

News and the latest products from Xilinx partners

by Mike Santarini
Publisher, Xcell Journal
Xilinx, Inc.
mike.santarini@xilinx.com

Synopsys Debuts Latest HAPS Prototyping System

Synopsys Inc. recently released the sixth generation of its HAPS ASIC and ASSP prototyping system, based on the Xilinx® Virtex®-6 FPGAs. The HAPS-60 series has more than double the capacity and up to 30 percent better performance than the prior-generation product, said Neil Songcuan, product marketing manager for FPGA-based prototyping solutions at Synopsys. In addition, the product offers new IP support via Synopsys' DesignWare IP library, along with advanced verification functionality to facilitate ASIC design and even end-product debug.

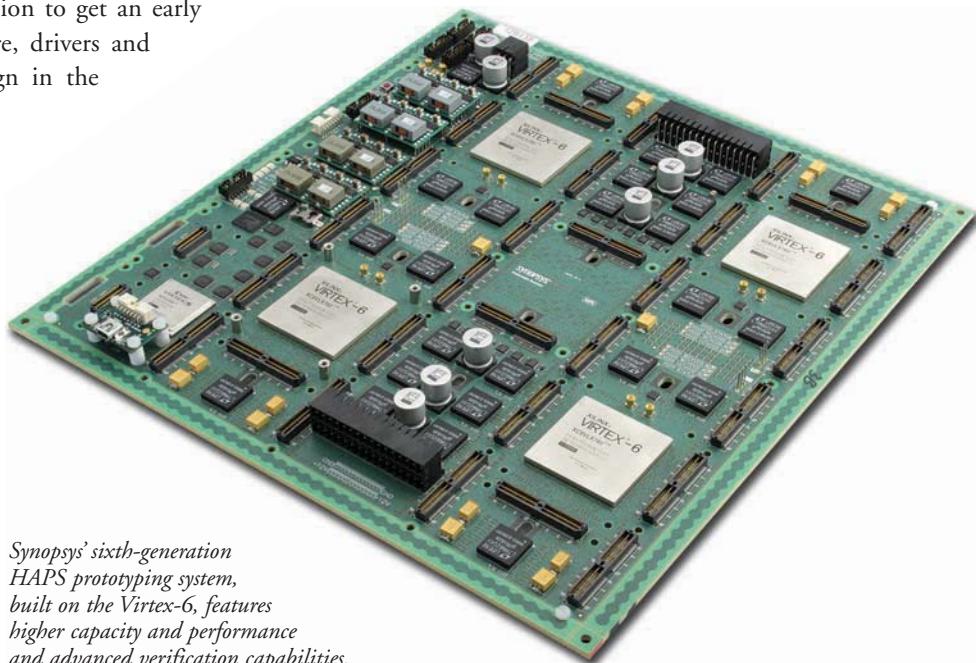
Chip companies typically use the HAPS systems to verify and validate the functionality of their ASIC designs in real hardware. They use Synopsys' Simplify synthesis in concert with the Certify partitioning tool to program their ASIC design into the FPGAs running on the HAPS system of choice. Once they have the hardware validated, they can use this HAPS version to get an early jump on developing and validating firmware, drivers and application software running on their design in the context of the entire end product.

Synopsys' new HAPS-60 line consists of three systems. The HAPS-61, which uses one Virtex-6 FPGA, supports up to 4.5 million ASIC gates. The HAPS-62, equipped with two Virtex-6 FPGAs, supports up to 9 million ASIC gates. The biggest system in the lineup, the HAPS-64, uses four Virtex-6 FPGAs and supports up to 18 million ASIC gates.

Songcuan said that all these systems achieve clock frequencies of up to 200 MHz (25 to 70 MHz typical). That performance allows the HAPS-60 series to support applications requiring real-time interfaces such as video, cellular data or live network traffic.

In addition to improved capacity and performance, Synopsys is also offering pretested intellectual property from its DesignWare IP library to facilitate implementing soft-IP blocks from an ASIC design into the FPGAs on the HAPS-60 system. "Customers are using in HAPS the exact same RTL they are using in their SoC (system-on-chip)," said Songcuan. "It allows you to make sure you are verifying exactly what is going to be in your SoC and focus on debugging your design rather than your prototyping system." The IP includes Synopsys' SuperSpeed USB 3.0, PCI Express and HDMI cores.

Contact your local sales representative for more information on availability and pricing of the HAPS-60 series of rapid prototyping systems. A list of Synopsys sales offices can be found at <http://www.synopsys.com/apps/company/locations.html>.



Synopsys' sixth-generation HAPS prototyping system, built on the Virtex-6, features higher capacity and performance and advanced verification capabilities.

Mentor Graphics Fields Precision Rad-Tolerant Synthesis Tool with Versatile TMR Capabilities

Mentor Graphics Corp. (Wilsonville, Ore.) recently released its new vendor-independent Precision Rad-Tolerant FPGA design solution for aerospace and high-reliability applications. Developed with the guidance of NASA, the tool provides synthesis-based radiation-effects mitigation. It includes all the features of the latest version of the company's traditional Precision synthesis tool, along with specialized features and flows for mil-aero and safety-critical applications.

Precision Rad-Tolerant is the first synthesis-based radiation-effects mitigation solution to reduce the risk of functionality problems including soft errors caused by single-event upset (SEU) and single-event transient (SET) disruptions," said Daniel Platzker, FPGA synthesis product-line director at Mentor Graphics' Design Creation and Synthesis Division. "It primarily targets aerospace-and-defense applications that use FPGAs, but there are an increasing amount of commercial applications that require high reliability and are also susceptible to SEU and SET disruptions."

Platzker cited communications and high-performance computing as two applications that demand ever-more-sophisticated radiation-effect mitigation technologies.

Platzker pointed to an automated multimode triple-modular redundancy (TMR) feature as one of the most advanced features in the new tool. In a TMR design, a voting circuit compares the results of three separate instantiations of a system that perform the same task. If at least two of them produce the same result, the voter deems it to be correct.

In a Xilinx FPGA environment, this new Precision synthesis feature is another approach to Xilinx's internally developed and well-matured TMRTTool. Platzker said that where the Xilinx TMRTTool implements redundancy post-synthesis (taking in a netlist and outputting a new netlist), the Precision Rad-Tolerant TMR tool performs triple-modular redundancy a step earlier in the design process—during logic synthesis. "We can decide if it is more optimal to infer one memory over another, or infer one kind of DSP over another, and decide at an earlier stage of the design cycle where to insert a voter," said Platzker. "You typically aren't making these decisions after you've designed the circuit."

Further, users can choose what types of TMR they would like the tool to implement. "It gives new levels of granularity in what you want to triplicate," he said. Designers can, for example, opt to implement redundancy of sequential or combinatorial logic. "The tool has a Local TMR mode," said Platzker. "You can choose to only triplicate flops, for example. You can also choose to triplicate a certain flop or feature."

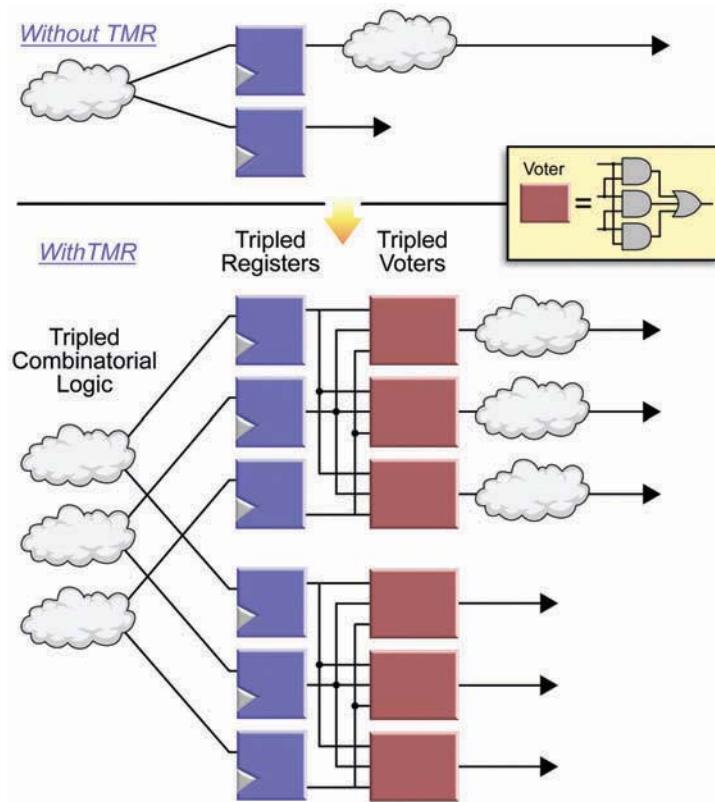
The tool also has a Distributed TMR mode. "If you are also concerned with SET, with distributed TMR, you can not only triplicate the selected flops, but also the logic on the drive of the flops and the voters," said Platzker. "It's a great complement to Xilinx's mitigation solutions."

In the third and final mode, the Global TMR mode, the tool not only triples all sequential elements, combinatorial logic and majority voters, but also global buffers.

Another feature of the Precision Rad-Tolerant product is synthesis-based insertion of fault-tolerant finite state machines (FSMs). "We have a mode called 'detection and recovery,'" said Platzker. "Essentially, we add two parity bits to the logic and regardless of the state size and encoding, if your design takes an SEU hit, it will return the state machine into its default state. At that time, the designer can decide what next action the design should take."

Another FSM mode is called "fault-tolerant FSM." Here, "We use Hamming-distance-3 error correction to allow the FSM to absorb the SEUs without disruption," said Platzker.

Platzker said the new Precision Rad-Tolerant product is available now and is priced higher than the standard version of the tool. For more information, visit www.mentor.com/precision-radtolerant.



The Distributed TMR and Global TMR modes triple sequential elements, combinatorial logic and majority voters.

GateRocket Offers Virtex-6 FPGA Version of RocketDrive Debug Environment

GateRocket Inc. (Bedford, Mass) has released a new version of its RocketDrive FPGA verification and debug environment for the Xilinx Virtex-6 family of high-performance programmable devices.

The RocketDrives are unique in that the devices, which are the size of a hard drive, actually include the Virtex-6 FPGAs you are targeting. In GateRocket's Device Native methodology, as you are simulating your design, you offload blocks of it onto the RocketDrive to speed up the performance of debug and verification. It also gives you a way to verify the design running on the same FPGA device on which you plan to implement your design.

"Today's FPGAs are simply too big to use the traditional 'blow and go' methodology, where you program your design into an FPGA, find and fix a problem and then repeat the process exhaustively until you hopefully find all bugs," said Dave Orecchio, the president and CEO of GateRocket. "That's why most designers now use simulation, but simulation alone tends to be slow and doesn't give you the true hardware functionality."

In contrast, GateRocket's Device Native methodology allows designers to run design blocks in the FPGA they are targeting, bringing more order to the debug process and speeding up verification and accuracy. "Designers typically see their FPGA verification and debug times cut in half compared to the traditional software-only verification cycle," said Orecchio.

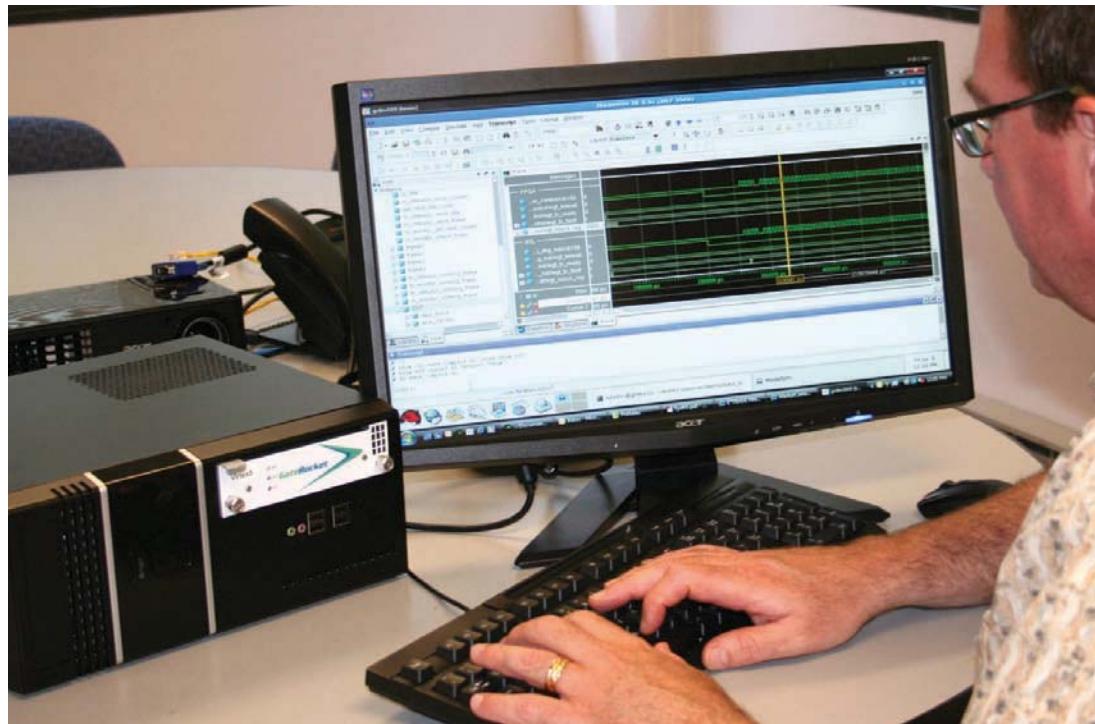
The GateRocket solution allows designers using Virtex-6 devices to move effortlessly between RTL and the specific FPGA being targeted, combining actual FPGA hardware and RTL simulation models together in a single verification run, without changes in the design flow or methodology. The company calls this technique "soft patch," and it gives engineers the ability to make a change to one or more RTL blocks and rerun them along with the hardware implementations of the other blocks. That sidesteps the need to rebuild the device for each fix and enables multiple design-change-debug iterations in a single day.

The new Virtex-6 RocketDrives use the largest LX and SX devices for advanced logic and DSP applications respectively.

GateRocket also offers a cost-effective midrange device configuration targeted at users who do not require the largest FPGA device in the family.

By using devices optimized for specific needs, GateRocket says it can deliver cost savings for an even greater customer return on investment. Each RocketDrive configuration offers the same enhanced verification performance and debug efficiency, and maintains complete compatibility with popular EDA logic simulators from Cadence, Mentor and Synopsys.

Along with the new Virtex-6 configurations, GateRocket also sells versions of its RocketDrive supporting Xilinx Virtex-4 and Virtex-5 FPGAs. Pricing for the Virtex-6 version starts at \$25,000. For more information about GateRocket, go to www.gaterocket.com.



GateRocket's RocketDrive, a unique verification tool that's the size of a hard drive, now comes in a Virtex-6 version.

Enclustra Introduces Compact FPGA Module with Fast Ethernet

FPGA design services, module and IP vendor Enclustra (Zurich, Switzerland) has just released an FPGA module equipped with two Fast Ethernet PHYs and a fast DDR2 SDRAM. As such, the Mars MX1 module is ideal for system-on-programmable-chip designs that pair a soft-core processor and real-time Ethernet functionality, said Oliver Brundler, development engineer at Enclustra. A reference MicroBlaze™ system using Xilinx Platform Studio is available and allows access to all on-module peripherals.

"The Mars MX1 is designed for applications such as industrial automation, where dual Ethernet PHYs are used for real-time Ethernet," said Brundler. "However, since Enclustra optimized the module for low cost, it can also be used for applications requiring low production volume or for rapid prototyping. Using the MX1 module, the hardware design effort can be reduced from over 100 components to just one."

Enclustra built the new Mars MX1 around Xilinx's most recent low-cost FPGA, the Spartan®-6 LX. The module comes in two standard configurations, carrying either the XC6SLX16

(14,579 LUT4 equivalents) or the XC6SLX45 (43,661 LUT4 equivalents) FPGA. Both variants are fitted with a 128-Mbyte DDR2 SDRAM, 16-Mbyte SPI Flash and real-time clock. The modules operate from a single 3.3-volt supply and provide 108 user I/Os, which you can also configure as 54 differential pairs. Since the form factor of the module is SO-DIMM (68 x 30 mm), you can use space- and cost-saving standard connectors to easily integrate the MX1 into your targeted system, the company said. Custom configuration options and suitable carrier boards are available upon request.

Enclustra is a company built around the FPGA technology. It offers not only FPGA modules but also IP and design services for various applications such as software-defined radio, drive control and single-chip systems. Along with the Mars MX1 module, the company also fields the MX2, a module equipped with PCI Express. Enclustra has a second module line named Saturn which is DSP-optimized and fitted with a Spartan-3A DSP FPGA.

For more information, visit www.enclustra.com/marsmx1.



Enclustra's Mars MX1 is very compact module designed for developing systems that combine Ethernet and system-on-chip processing.

Xpress Yourself in Our Caption Contest

GETTY IMAGES



Send your entries to xcell@xilinx.com. Include your name, job title, company affiliation and location, and indicate that you have read the contest rules at www.xilinx.com/xcellcontest. After due deliberation, we will print the submissions we like the best in the next issue of *Xcell Journal* and award the winner the new Xilinx® SP601 Evaluation Kit, our entry-level development environment for evaluating the Spartan®-6 family of FPGAs (approximate retail value, \$295; see <http://www.xilinx.com/sp601>). Runners-up will gain notoriety, fame and a cool, Xilinx-branded gift from our SWAG closet.

The deadline for submitting entries is 5:00 pm Pacific Time (PT) on Sept. 15, 2010. So, cast your nets and get writing!

There's something fishy going on here, and we're counting on you to tell us what it is. If you have a yen to Xercise your funny bone, step up to our verbal challenge and submit an engineering- or technology-related caption for this image of robotic fish developed at the Massachusetts Institute of Technology. MIT scientists say the critters could potentially be used to detect underwater pollutants and inspect submerged boats or structures. The group seen here might inspire a caption like "Herbie angled to debug the sole remaining catch that was making his design flounder."

MARCO RIVERO,

an electrical engineer at the Anderson Research Group at Harvard University, has won an SP601 Evaluation Kit with his caption for the photograph of the man arguing with a robot in Issue 71 of *Xcell Journal*.



GETTY IMAGES

"C'mon Ref! He's clearly using his 'acting' function!"

Congratulations as well to our two runners-up:

Playing soccer—check. Understanding what a bribe or taking a dive is—fail.

— Matthew Hicks, PhD Candidate,
University of Illinois

"AI Divorce Court 2020!! – New This Fall! – Check local listings in your area."

— David Santoro, PhD Candidate,
Electrical Engineering,
City College of New York

The Best Prototyping System Just Got Better

HAPS™ High-performance ASIC Prototyping Systems™

- ✓ Highest Performance
- ✓ Integrated Software Flow
- ✓ Pre-tested DesignWare® IP
- ✓ Advanced Verification Functionality
- ✓ High-speed TDM



SPARTAN[®]6

- Easy-to-use, low power, serial transceivers support up to 3.125Gbps to enable industry standards such as PCIe®
- Low voltage option reduces total power consumption by 65% over previous generations
- Integrated DSP, memory controllers, and clocking technology simplifies designs

VIRTEX[®]6

- High bandwidth serial connectivity with up to 72 low-power transceivers supporting up to 11.18Gbps
- Ultra high-performance DSP using up to 2016 low-power, performance-optimized DSP slices
- Integrated high-performance ExpressFabric™ technology running at 600 MHz clocking and performance-tuned IP blocks
- Proven cost-reduction with EasyPath™-6 FPGAs

Potential. Realized.

Unleash the full potential of your product design with Xilinx® Virtex®-6 and Spartan®-6 FPGA families — the programmable foundation for Targeted Design Platforms.

- Reduce system costs by up to 60%
- Lower power by 65%
- Shrink development time by 50%

Realize your potential. Visit www.xilinx.com/6.