ISSUE 77, FOURTH QUARTER 2011

Xcell journal

SOLUTIONS FOR A PROGRAMMABLE WORLD

Xilinx Ships World's Highest-Capacity FPGA With SSI Technology

Robotic-Assisted Surgical System Uses Xilinx FPGAs

Zynq-7000 EPP Makes Flexible Software-Defined Radio Platform

ISE Design Suite 13.3 Now Available for Download

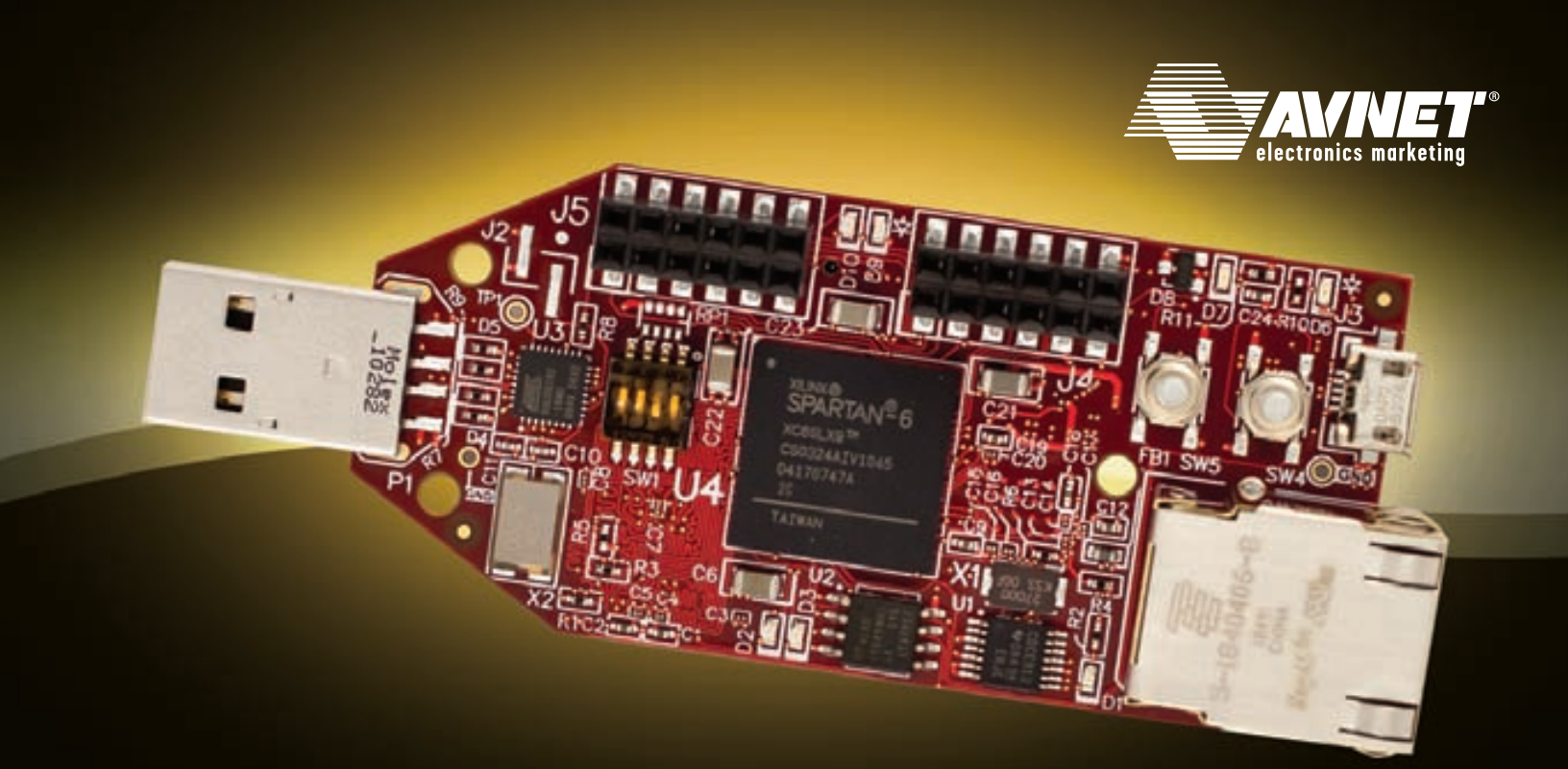
How to Mount a Full-Frontal Assault on Power

page 52

XILINX®

www.xilinx.com/xcell/

AVNET electronics marketing



DESIGNED BY AVNET

Compact, Easy-to-Use Kit Demonstrates the Versatility of Spartan-6 FPGAs

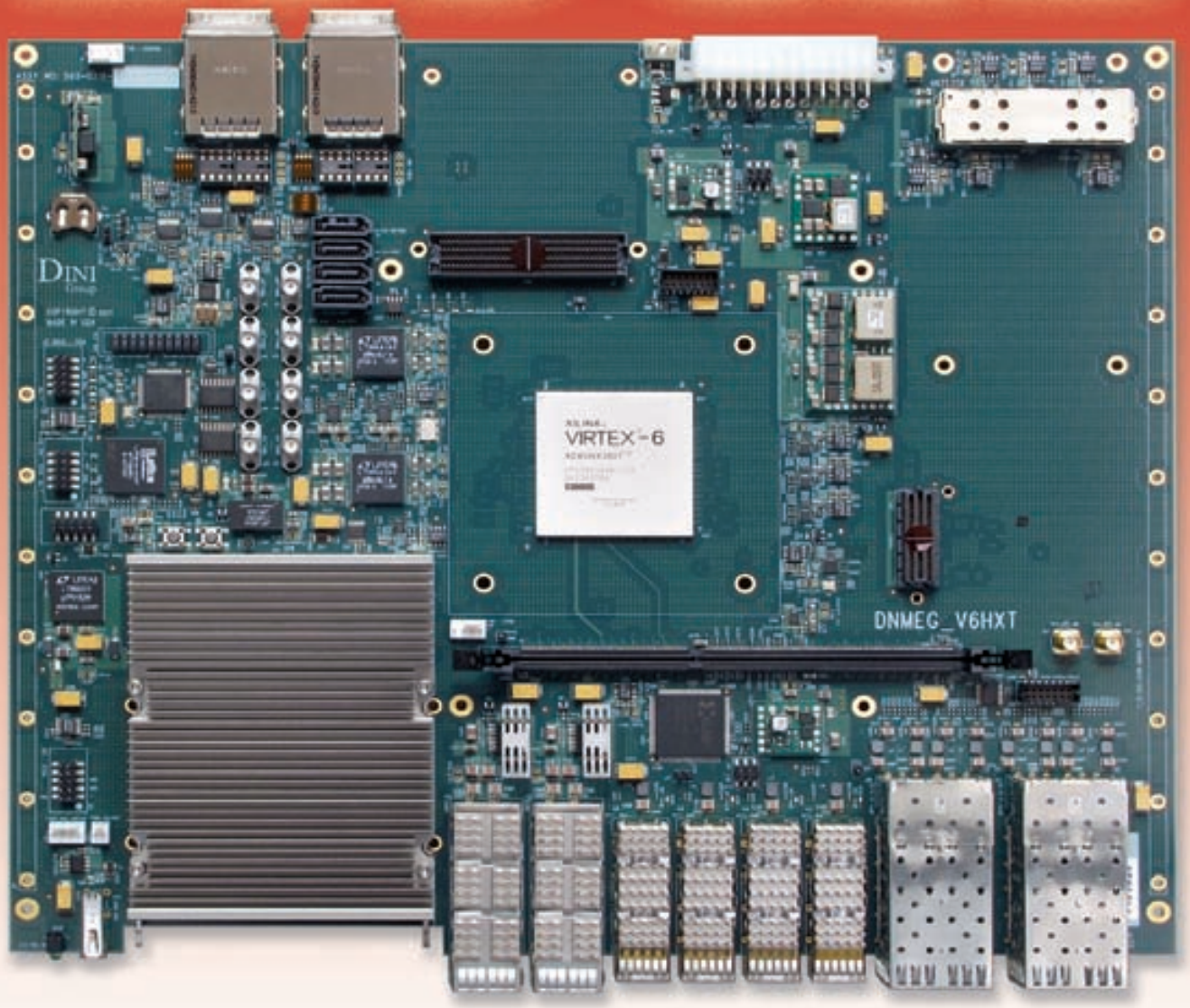
The low-cost Spartan®-6 FPGA LX9 MicroBoard is the perfect solution for designers interested in exploring the MicroBlaze™ soft processor or Spartan-6 FPGAs in general. The kit comes with several pre-built MicroBlaze "systems" allowing users to start software development just like any standard off-the-shelf microprocessor. The included Software Development Kit (SDK) provides a familiar Eclipse-based environment for writing and debugging code. Experienced FPGA users will find the MicroBoard a valuable tool for general-purpose prototyping and testing. The included peripherals and expansion interfaces make the kit ideal for a wide variety of applications.

Xilinx® Spartan®-6 FPGA LX9 MicroBoard Features

- Avnet Spartan-6 FPGA LX9 MicroBoard
- ISE® WebPACK® software with device locked SDK and ChipScope licenses
- Micro-USB and USB extension cables
- Price: $89

To purchase this kit, visit www.em.avnet.com/s6microboard or call 800.332.8638.

250 GbE



Add it up: 10 GbE, 40 GbE, and 100GbE ports offer more capacity and versatility than any High Speed Serial I/O board – *ever*. The Xilinx, Virtex 6, HXT FPGA provides user program-ability and data rates to 11 GB/s. ASIC designers, Network engineers, and IP developers, can use this tool to prototype, emulate, model and test your most challenging projects. The board maybe used stand-alone, PCIe hosted, or plugged into any Dini Group ASIC prototyping board. Users will appreciate the scope of interfaces:

- CFP Module—100 GbE or Single/Dual 40 GbE
- 2-QSFP+ Modules—40 GbE
- 4-SFP+ Modules—40 GbE
- 8-SFP+ Modules—8 GbE

All FPGA resources are user available and supported by a 240-pin DDR3 UDIMM bulk memory. Daughter cards are accommodated for expansion, customization, and FMC (Vita-57). Put your high speed serial designs to work. Here is a Dini board just for you.

DINI Group

www.dinigroup.com • 7469 Draper Avenue • La Jolla, CA 92037 • (858) 454-3419 • e-mail: sales@dinigroup.com

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

Xcell PUBLICATIONS
www.xilinx.com/xcell/

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

2 Million Logic Cell FPGA Available Today!

When I first started covering IC design methodology and the semiconductor business back in 1995, the biggest FPGA available, Xilinx's XC5215, boasted a then-mammoth 1,936 logic cells, roughly the equivalent of 10k ASIC gates. Though the XC5215 was the monster FPGA for its day and a very cool device, its use in an end product was circumscribed. Back then, an FPGA would likely serve as either glue logic to help two disparate devices communicate or would hold functionality that the system required in the 11th hour.

If a company wanted to build custom functionality into its end products, they built or commissioned an ASIC. Up to that point, ASICs were relatively easy to design and cost-effective to build. But that changed very quickly. Within a year, the ASIC industry had to confront how it was going to fill the insanely generous 100,000 gates now afforded to them with the introduction of cutting-edge 0.5-micron processes.

Alas, after much debate, the logic-chip industry collectively opted to fill all those gates by means of IP blocks and design reuse. Over the next 15 years, as silicon process geometries shrank, the challenges mounted and the industry had to deal with a succession of other issues such as timing closure, design-for-manufacturability and gate leakage. The ingenuity of the folks in this industry never ceases to amaze me, and, over time, good old engineering came up with ways to adequately address all these issues.

But while engineering has made it possible for ASIC designs to still be built, the economics of IC manufacturing haven't been so kind. Indeed, multimillion-dollar mask costs added to the tool costs and complexity of ASICs have made ASIC design feasible for fewer and fewer products. Today a company has to sell millions of ASICs to simply recoup the costs of building them in the latest process technologies.

Thankfully, good old engineering isn't concentrated only on the ASIC business. The folks at Xilinx have been busy making FPGAs not just a viable alternative to ASICs, but a better one. This issue's cover story details a huge engineering accomplishment: Xilinx is now shipping to customers its 2 million logic cell or 20 million ASIC-gate-equivalent Virtex®-7 2000T FPGA. This new device is not only the highest-capacity FPGA in the industry, it is a much better alternative—in terms of both engineering and economics—to ASICs for a growing number of applications. This chip is also the first commercially released device that uses Xilinx's stacked-silicon interconnect technology, which I described in detail in the cover story in *Xcell Journal* Issue 74.

As a refresher, SSI technology connects several silicon slices (essentially, dice) side-by-side on top of a silicon interposer. The dice are then interconnected by traces running through the interposer, much in the way that disparate ICs communicate by means of traces in a printed-circuit board. But with guidance from Xilinx tools, it doesn't look to the designer like multiple chips—just one humongous one. In pioneering this device, Xilinx's engineering staff has taken the semiconductor industry a monumental first step into the era of 3D IC technology.

I'm certain that those folks at Xilinx who have been working on SSI for the last five years will tell you that while it has been a monumental challenge, it has also been extremely rewarding as this device makes its way into customer end products. Their innovation in turn allows you, the designer, to innovate new products, penetrate new markets and achieve higher levels of prosperity.

Mike Santarini
Publisher

Pass GO, Collect 3.6 Billion Samples per Second

X6 GSPS

Extremely Versatile

Complete chassis & embedded PC solutions available! Adapt to VPX, Cabled PCI-Express, PCI-Express, 64-bit PCI, and CompactPCI with our high-quality carriers.

Features

- Two 1.8 GSPS, 12-bit A/D Channels
- Single channel interleaved @ 3.6GHz
- +/-1V, AC-Coupled, 50 ohm, SMA Inputs
- Xilinx Virtex-6 SX315T/SX475T or LX240T
- 4 Banks of 1GB DRAM (4 GB total)
- Rugged levels available

VIRTEX 6

wireless ip cores

FrameWork Logic

RoHS

805.578.4260 phone • www.innovative-dsp.com

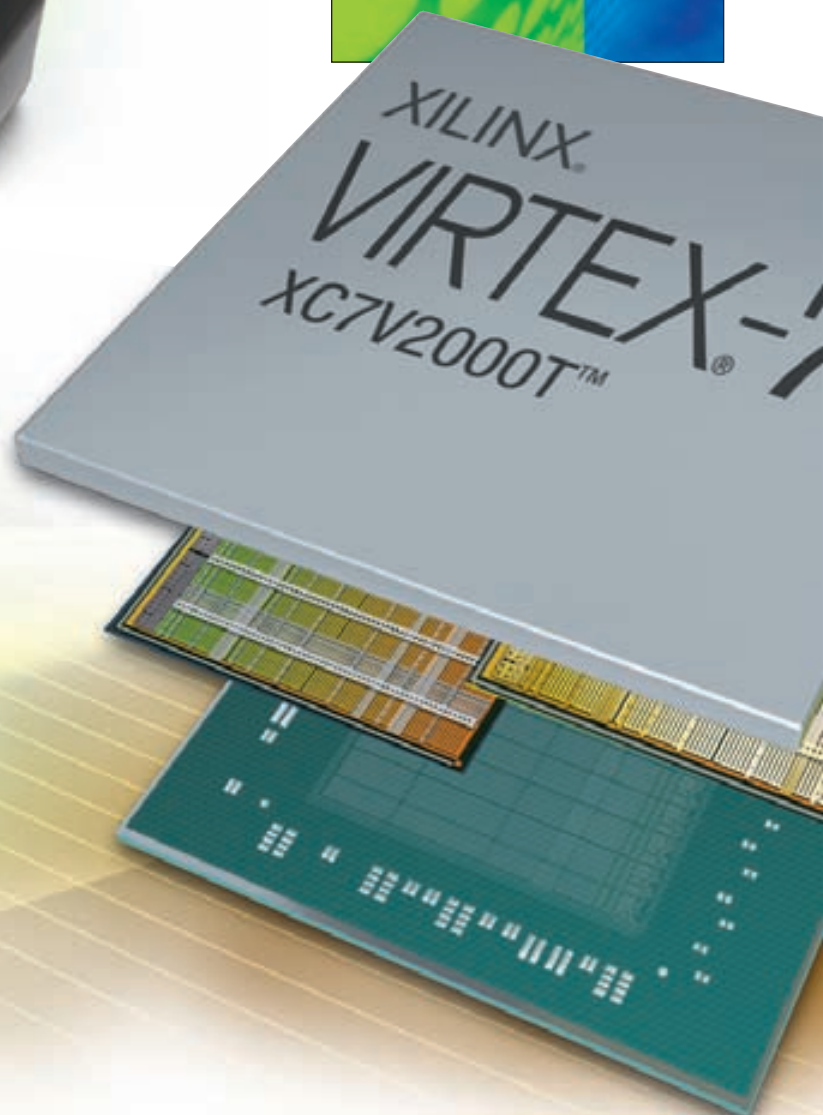
Innovative Integration
a subsidiary of ISI
... real time solutions!

VIEWPOINTS

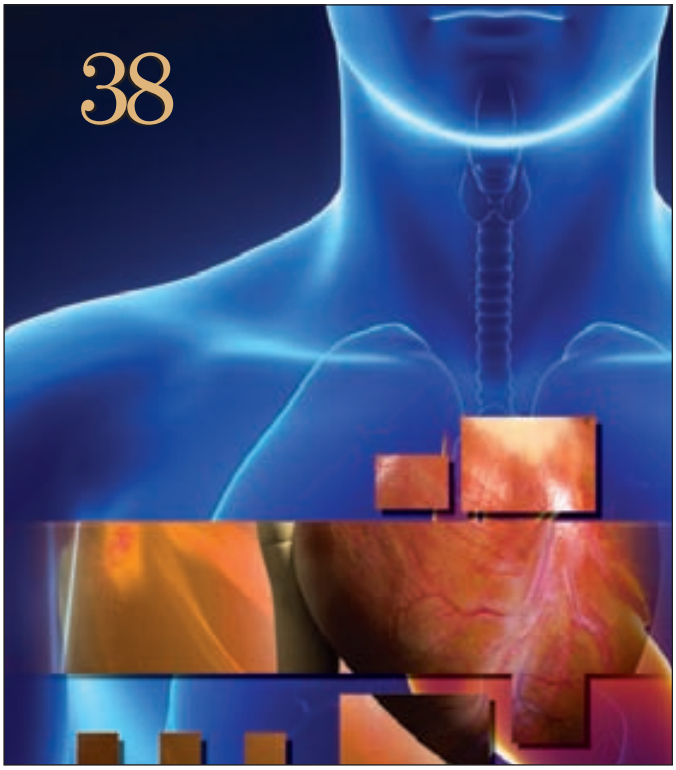
XCELLENCE BY DESIGN APPLICATION FEATURES

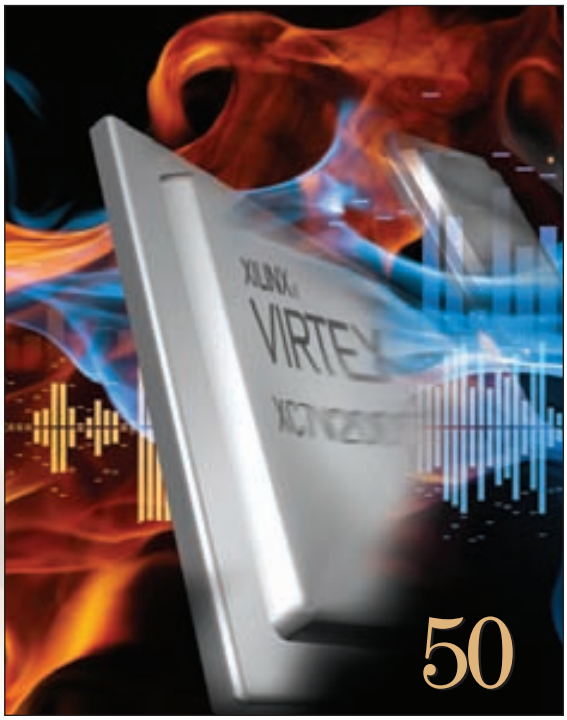
Cover Story

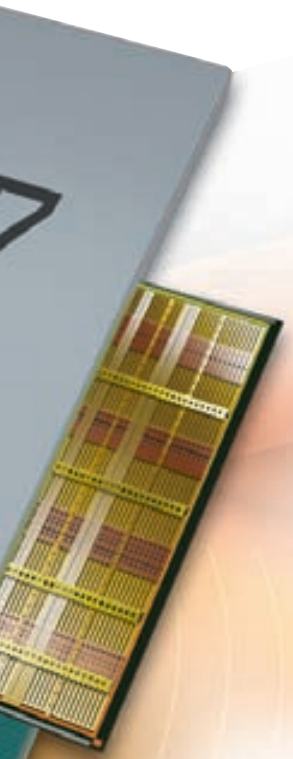
Xilinx Ships World's Highest-Capacity FPGA Using Stacked-Silicon Interconnect Technology



THE XILINX XPERIENCE FEATURES







XTRA READING

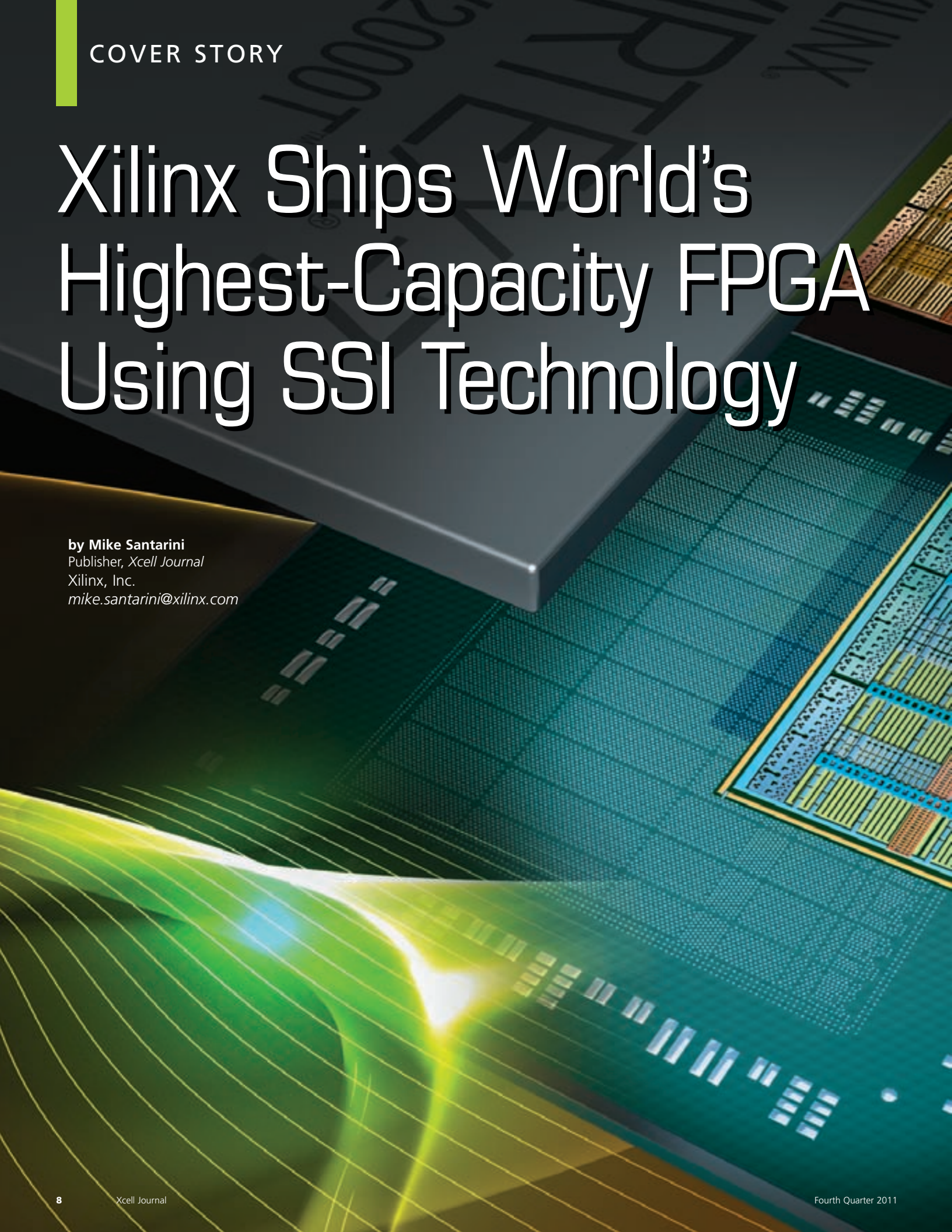
2010 APEX® AWARDS FOR PUBLICATION EXCELLENCE

Excellence in Magazine & Journal Writing
2010, 2011

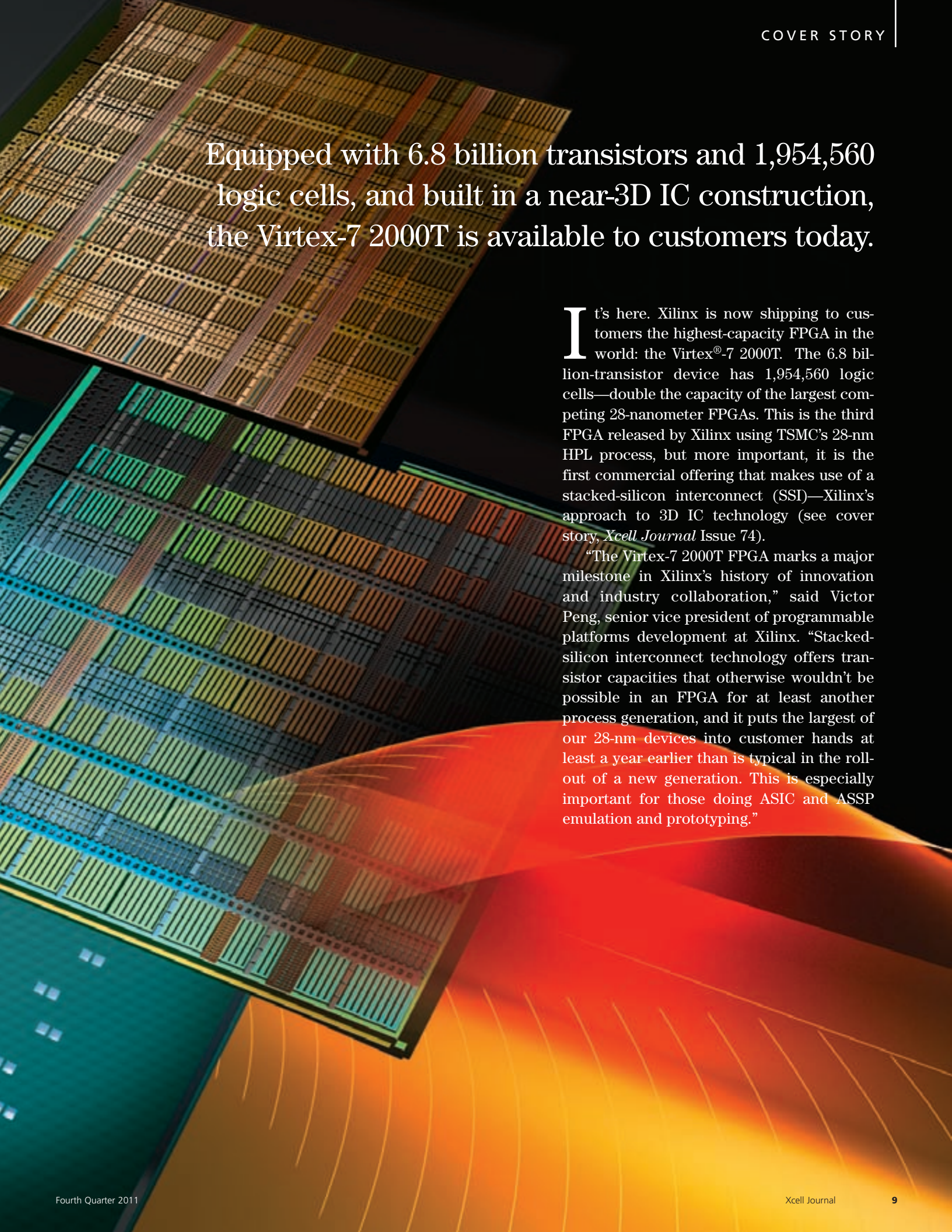
2011 APEX® AWARDS FOR PUBLICATION EXCELLENCE

Excellence in Magazine & Journal Design and Layout
2010, 2011

Xilinx Ships World's Highest-Capacity FPGA Using SSI Technology



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



Equipped with 6.8 billion transistors and 1,954,560 logic cells, and built in a near-3D IC construction, the Virtex-7 2000T is available to customers today.

It's here. Xilinx is now shipping to customers the highest-capacity FPGA in the world: the Virtex®-7 2000T. The 6.8 billion-transistor device has 1,954,560 logic cells—double the capacity of the largest competing 28-nanometer FPGAs. This is the third FPGA released by Xilinx using TSMC's 28-nm HPL process, but more important, it is the first commercial offering that makes use of a stacked-silicon interconnect (SSI)—Xilinx's approach to 3D IC technology (see cover story, *Xcell Journal* Issue 74).

"The Virtex-7 2000T FPGA marks a major milestone in Xilinx's history of innovation and industry collaboration," said Victor Peng, senior vice president of programmable platforms development at Xilinx. "Stacked-silicon interconnect technology offers transistor capacities that otherwise wouldn't be possible in an FPGA for at least another process generation, and it puts the largest of our 28-nm devices into customer hands at least a year earlier than is typical in the roll-out of a new generation. This is especially important for those doing ASIC and ASSP emulation and prototyping."

Traditionally, FPGA vendors have implemented their new architectures on the latest manufacturing process technologies to take advantage of Moore's Law, which holds that transistor counts double every 22 months with the introduction of each new silicon process. Following Moore's Law over the last two decades has allowed FPGA vendors to consistently offer new FPGAs that double the capacity of their previous-generation devices.

However, for the Virtex-7 2000T and a few other members of the Virtex-7 family, Xilinx moved beyond Moore's Law to create SSI. This technology connects several silicon slices (active dice) side-by-side on top of a passive silicon interposer. The dice are then interconnected by metal interconnect running through the interposer, much in the same way that disparate ICs communicate by means of metal interconnect in a printed-circuit board (see Figure 1). In this way, Xilinx was able to make devices that exceed the pace of Moore's Law: Virtex-7 2000T FPGAs are twice the size of the nearest competitor's largest 28-nm device and 2.5x larger than Xilinx's largest Virtex-6 FPGA. But the beauty of the architecture, said Panch Chandrasekaran, product line manager for Virtex-7 FPGAs at Xilinx, is that despite being composed of four dice, the 2000T preserves the traditional FPGA use model.

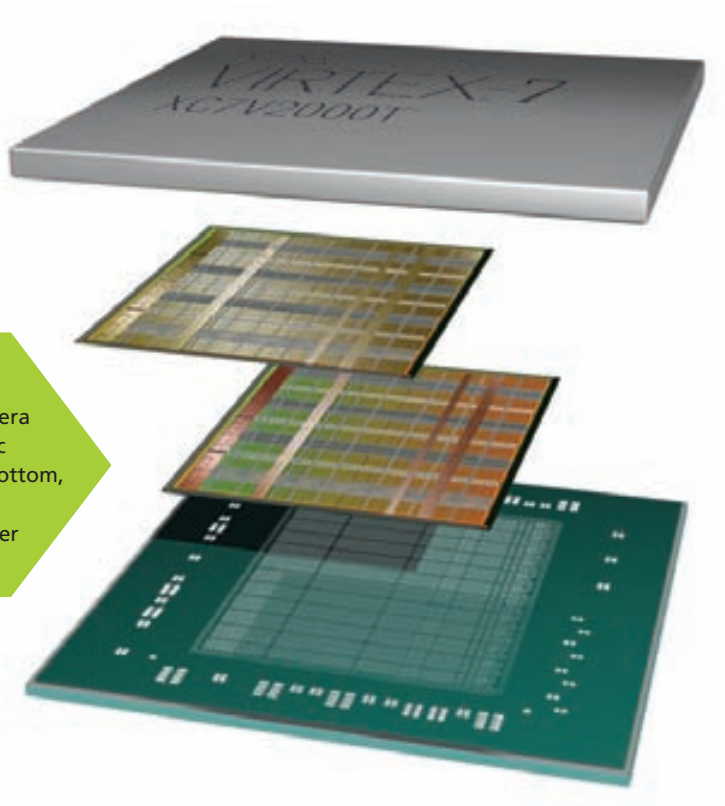
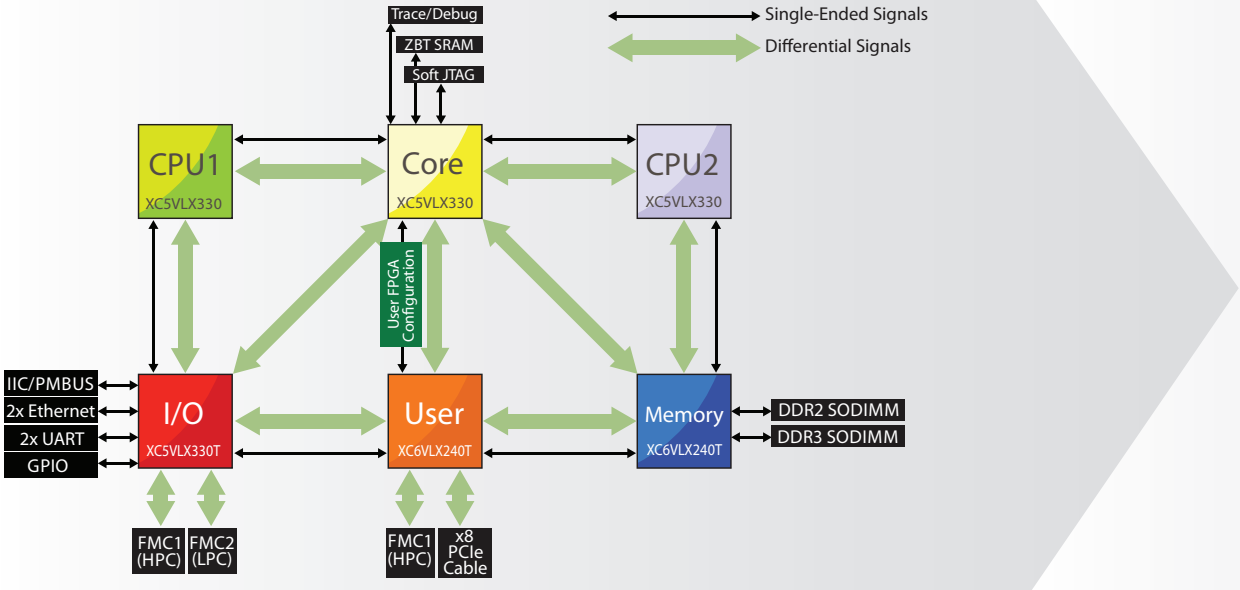
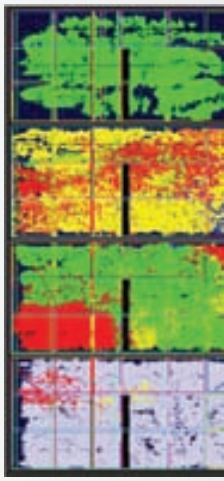


Figure 1 – Xilinx's Virtex-7 2000T takes a giant step into the era of 3D ICs. The graphic shows, from top to bottom, the package cover, slices, silicon interposer and substrate.

Figure 2 – Capacity is so ample that the Virtex-7 2000T can accommodate complex designs of up to 20 million ASIC gates while saving drastically on NRE costs. Here is an example of how one such design lays out in a Virtex-7 2000T.





Designers will program it as one extremely large FPGA using the Xilinx tool flow and methodology.

In addition to having 1,954,560 logic cells, the Virtex-7 2000T includes configurable logic blocks totaling 305,400 CLB slices and max distributed RAM of 21,550 kbits. It has 2,160 DSP slices, 46,512 Kb of block RAM, 24 clock management tiles, four PCIe® blocks and 36 GTX transceivers (each capable of 12.5 Gbits/second). It also has 24 I/O banks and a total of 1,200 user I/Os.

While the release of the Virtex-7 2000T marks a big achievement for Xilinx and a bold step into the era of the 3D IC for the semiconductor industry, Chandrasekaran said the real value is the doors it opens for user innovation and the new design capabilities it brings to customers looking for the highest-capacity devices. "Those who want to speed product development and supply silicon emulations to their software developers, those who want to condense multiple chips into a single device and those who can't justify doing ASICs for their designs—they will all greatly benefit from this fantastic technology," he said. "By implementing the device in SSI technology, Xilinx is putting next-generation capacity in designers' hands today."

ASIC AND IP EMULATION AND PROTOTYPING

Today, an average high-end ASIC or ASSP design has 420 million gates, said Gary Smith, design-tool analyst with Gary Smith EDA and an ASIC methodology expert. "The largest I've heard of was 1.1 billion," he said. Because of these high gate counts, more than 90 percent of ASIC design teams use some form of hardware-assisted verification systems, whether they be commercial emulation systems or do-it-yourself ASIC-prototyping boards.

Traditionally, companies creating commercial emulation systems or design groups prototyping their designs have been first-in-line customers for the biggest FPGAs a vendor can produce. Suppliers of commercial emulation systems are looking for FPGAs with the highest possible capacity. "This market will especially benefit from having the Virtex-7 2000T's more-than-Moore's Law capacity at their disposal," said Chandrasekaran. "It allows them to offer next-generation-capacity emulation systems today, and ultimately allows their customers to cut design cycles and bring new, more innovative products to market faster."

Most of these commercial emulation systems include two or more boards and up to several racks of FPGAs, depending on the size of the ASIC, IP or even system a customer wants to emulate. Meanwhile, customers of emulation systems use them to speed up verification and ensure their designs function properly, and to get the hardware versions of their designs to software groups so they can get a leg up on software development and have it mostly completed by the time the foundry delivers the real, silicon ASIC. The idea, of course, is to speed time-to-market.

In a typical use model of a commercial emulation system, users first design and functionally verify their ASIC or IP using traditional EDA verification software. After they've done that sufficiently, they then implement the register transfer-level (RTL) version of their design in a commercial emulator for further verification of the design. Each emulator vendor typically offers its own software that works in conjunction with Xilinx's design software to synthesize the RTL and partition the ASIC design into blocks that can be optimally distributed among the FPGAs in the emulator. The emulation vendor's software

As silicon processes have advanced, the cost of designing and manufacturing them has rapidly risen. The Virtex-7 2000T will appeal to the growing number of design groups that simply can't justify the cost and risk involved in developing an ASIC or ASSP at the 28-nm process node.

then provides an interface to a workstation or PC running various EDA verification tools to test the design as it runs on the emulator.

Emulation vendors also offer lower-cost variants—sometimes called "replicates" or, generically, "prototyping systems"—of their emulators. These variants simply emulate the ASIC's functionality. Companies give these systems to their software groups to get an early jump on developing drivers, firmware and applications that will run on the design.

Chandrasekaran said that larger FPGAs allow emulation vendors to either field higher-capacity emulation systems, or to build mid- and lower-capacity systems with fewer FPGAs, cutting power and the bill of materials while raising overall clock speeds of the designs running on them. "The Virtex-7 2000T is so large that these vendors will even be able to offer emulators built on a single FPGA chip," said Chandrasekaran. "Because the designs are running on fewer chips or even just one chip, the overall performance of the system will be faster."

The Virtex-7 2000T will also be ideal for design groups that perhaps can't afford off-the-shelf commercial emulation systems, which can cost upwards of a million dollars. "Many design groups build their own custom boards to prototype and/or emulate an ASIC, or an entire system's functionality, and get a jump on software development," said Chandrasekaran. "And even those who use emulation systems to develop their IC might create their own FPGA variants for their software groups."

Chandrasekaran said the device will also be attractive to IP vendors. Not only can they use the FPGAs to develop new blocks of IP, they can also use them to demonstrate the functionality of their cores to potential customers.

SYSTEM ARCHITECTURE CONSOLIDATION AND POWER SAVINGS

In addition to being attractive for ASIC and IP emulation and prototyping, the new Virtex-7 2000T will also be attractive to system architects looking for ways to lower the power consumption of their systems, while increasing performance and capabilities.

"There are many end products on the market that use multiple FPGAs," said Chandrasekaran. "With the Virtex-7 2000T, they can integrate the functionality of several FPGAs into a single FPGA. This system integration improves performance, because all these functions are on a single chip. System integration lowers power by eliminating the I/O interfaces between different ICs on the board. I/O interfaces increase power consumption proportionately to the number of I/Os. Thus, the higher the performance of the design and the greater the number of ICs in a system, the greater the power consumption."

Moreover, partitioning system functionality among multiple ICs is a complex job and can lead to longer development cycles and higher test costs. Consolidating the number of devices in the system reduces these partitioning challenges as well as the costs associated with verification and test. "At more than twice the capacity of competing FPGAs, the Virtex-7 2000T lets customers integrate more to reduce power fourfold compared with multiple-chip solutions. They can also increase system performance by eliminating I/O bottlenecks, and reduce system complexity by eliminating unnecessary design partitioning," said Chandrasekaran. "Architects can save a lot of board space to add other functions, or can simply shrink the size of the product."

As with other devices in the 7 series, Xilinx implemented the Virtex-7 2000T in TSMC's FPGA-specific 28-nm High Performance Low Power (HPL) process technology (detailed in the cover story of *Xcell Journal* Issue 76.) Thanks to the HPL process, the Virtex-7 2000T's transistors experience less leakage than those of competing devices implemented in 28-nm high-performance processes, Chandrasekaran said. This means the device has power consumption comparable to that of competing devices that are half its capacity.

ASIC DISPLACEMENT

Last but not least, the Virtex-7 2000T will also be attractive to the growing number of design groups that simply can't justify the cost and risk involved in developing an ASIC or ASSP at the 28-nm process node. As silicon processes have advanced, the costs of designing and manufacturing

them have rapidly risen. At 28 nm, ASIC or ASSP nonrecurring-engineering (NRE) expenses are better than $50 million, and the probability of an ASIC respin increases to nearly 50 percent. A single overlooked mistake during the design cycle can severely damage profitability; multiple mistakes can lead to project cancellations, missed market windows and even destroy companies.

The Virtex-7 2000T can replace 10 million- to 20 million-gate ASICs without the NRE costs associated with ASICs. "Instead of worrying about every little mistake leading to a catastrophic mask spin, designers can instead focus on designing," said Chandrasekaran. "What's more, the Virtex-7 2000T is reprogrammable, so if they do make a mistake they can simply reprogram the device."

METHODOLOGY STAYS THE SAME

While the Virtex-7 2000T is extremely large, programming the device won't require an extreme methodology change.

"Over the last few years Xilinx has been optimizing its design tools with ultralarge-capacity designs in mind," said Chandrasekaran. "Today, customers can efficiently partition, floorplan and optimize for power and performance." Chandrasekaran explains that most if not all large FPGAs typically require designers to perform some amount of partitioning, and to place timing-critical functions in close proximity to each other. For design groups placing large designs in the Virtex-7 2000T, Xilinx tools will assist them in floorplanning and partitioning their designs to help them achieve optimal timing and performance.

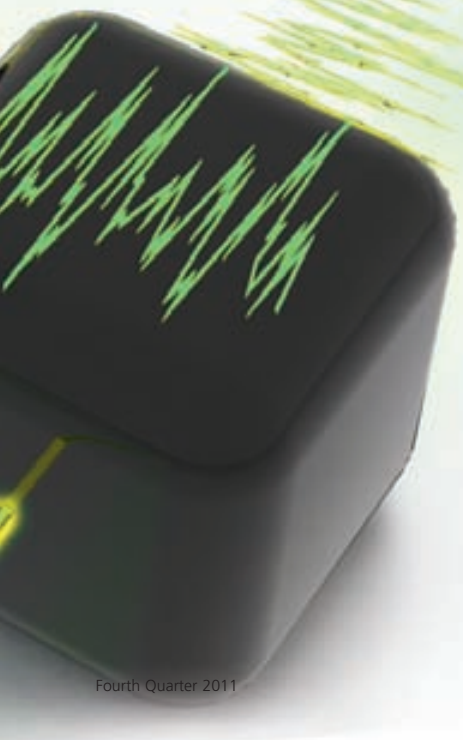
The latest release of Xilinx design tools supports the Virtex-7 2000T. "Users can get started today designing with Virtex-7 2000T," said Chandrasekaran. In the coming year, Xilinx is scheduled to release other Virtex-7 FPGAs in monolithic as well as SSI configurations.

For more information and to see the Virtex-7 2000T in action, visit *www.xilinx.com/7*.

Zynq-7000 EPP Makes Flexible Software-Defined Radio Platform

This Xilinx FPGA equipped with ARM processors integrates all the logic needed to handle multiple waveforms.

by Anita Schreiber
Senior Staff Applications Engineer
Defense Applications
and Systems Architecture
Xilinx, Inc.
Albuquerque, N.M.
Anita.Schreiber@xilinx.com



Today's tactical and commercial software-defined radios must have the flexibility and processing power to support a growing number of wideband and broadband waveforms including an extensive library of legacy waveforms. The secure Xilinx® Zynq™-7000 Extensible Processing Platform (EPP) provides an ideal solution for these applications, not only because it features a high-performance processing system that leverages ARM® technology, but also because it provides a large programmable logic unit that supports partial reconfiguration and the Xilinx Isolation Design Flow—all within a single device.

Let's take a closer look at a software-defined radio (SDR) project built on the Zynq-7000 EPP, focusing special attention on how to utilize the partial reconfiguration and Isolation Design Flow capabilities of the programmable logic unit to support various waveforms, reduce part count and save power. The Zynq-7000 EPP can cut the parts count for our example design from five devices to one, while providing plenty of flexibility to support future waveforms using the same hardware platform. Add in its power-saving features and it's clear that putting this device at the center of an SDR can significantly improve the size, weight, power and cost (SWAP-C) of the system and provide a flexible, programmable platform.

Figure 1 shows an example block diagram of a tactical SDR. As you can see, the plain-text portion of the radio (sometimes referred to as the "red" side, since the information could be classified) contains a general-purpose processor (GPP) and "red" FPGA. The plain-text information is then encrypted and transformed into cipher text. The "black" side of the radio—which contains a "black" FPGA, another GPP and a modem FPGA for waveform processing—then processes the cipher-text information. To ensure the security of the information, the design isolates and separates the plain-text and cipher-text portions of the radio to prevent an information leak of classified or sensitive data out of the system in plain text.

Therefore, a typical SDR implementation may use three different FPGAs as well as two separate GPPs, making the device count for these functions as high as five. Designers must size the modem FPGA appropriately to process all of the various waveforms the radio supports. The modem is often required to have all the functions available simultaneously even if only one is needed at a time. For example, processing the Soldier Radio Waveform needs roughly 120K logic cells, 8 Mbits of RAM and 800 DSP slices, while the Mobile User Objective System waveform requires more than 200K logic cells, 10 Mbits of RAM and 900 DSP slices, requiring the modem FPGA to be quite large. Also, the red FPGA in the crypto block must also be large enough to contain all of the various

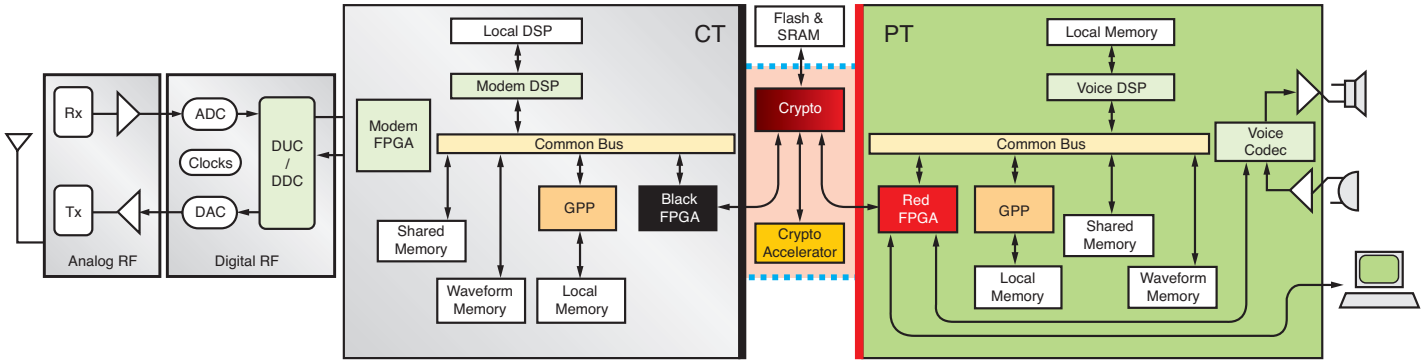


Figure 1 – Block diagram of an example software-defined radio showing the "red" (for classified information) and black FPGAs

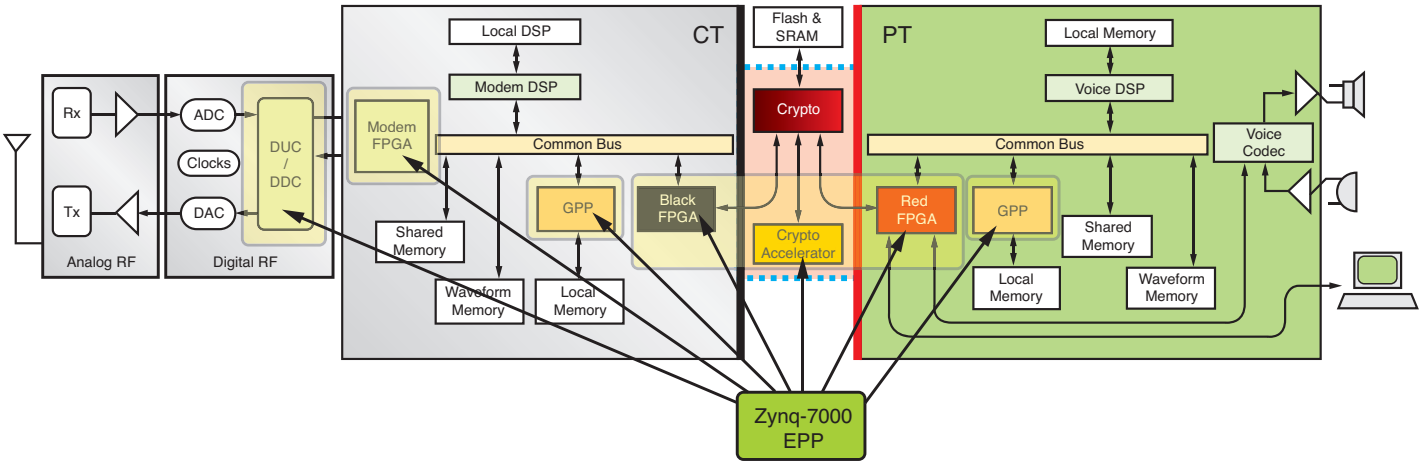


Figure 2 – The Zynq-7000 EPP integrates multiple FPGAs into one device for a simple tactical SDR.

cryptographic algorithms for the associated waveforms. The required number of devices, the amount of I/O signaling between them (which increases power dissipation) and the high logic density of the devices (which increases static current) make this a nonoptimal solution in terms of SWAP-C.

With the Xilinx Zynq-7000 Extensible Processing Platform, the modem FPGA, black FPGA, red FPGA, red GPP and black GPP can all be combined into one device, as Figure 2 shows . The Zynq-7000 EPP enables the combination and integration of these devices and provides a flexible, secure SDR solution that reduces SWAP-C.

The Xilinx Zynq-7000 EPP combines an industry-standard ARM processor-based system with Xilinx 28-nanometer low-power, high-performance programmable logic in a single device. The processing system (PS) provides dual ARM Cortex™-A9 processors, Level 2 cache and on-chip memory, as well as a rich peripheral set sufficient for general-purpose and waveform processing. The programmable logic (PL) provides ample logic cells, configurable memory (dual-port RAM, FIFOs, shift registers, BRAM) and hardware multipliers for DSP which can be utilized for high-speed parallel processing of needed functions.

PROCESSING SYSTEM

The dual ARM A9 cores each have 32 kbytes of instruction and data L1 cache and 512 kbytes of shared L2 cache. Additionally, 256 kbytes of on-chip memory are available to provide low-latency memory to both processors. A snoop control unit maintains cache coherency. Logic within the PL portion of the device and the ARM A9 processors can share memory, allowing fine-grained interaction between the processors and user logic. An Accelerator Coherency Port (ACP) allows coherent sharing of information between the processor and the PL by enabling logic within the programmable logic access to both the L2

cache and the on-chip memory. Direct access to the on-chip memory from the PL is also available. In addition, many industry-standard peripherals and memory interfaces are available to support general and waveform processing, as Figure 3 shows.

SECURE PROCESSING

With the secure processing features of the Zynq-7000 EPP and the current push on the part of the government to build secure products by "composing" solutions from commercial parts, the Zynq-7000 EPP is a good bet to replace both the red-side and the black-side GPPs in some applications. The Zynq-7000 EPP provides both a secure boot process and a secure run-time environment.

Unlike many ASSPs, the Zynq-7000 provides a Master Secure Boot Mode for configuration of secure, encrypted designs using the internal hardware AES decryption engine and SHA-256-based authentication engine (HMAC) components within the programmable logic. The secure boot sequence is shown in Figure 3.

The master ARM A9 processor will boot first from the on-chip ROM. It then reads the processing system boot image from the external boot device specified by the bootstrap pins. At the same time, the A9 configures the device configuration block to push the processing system master image through the AES decrypt and HMAC authentication engines. The configuration logic loops back the decrypted, authenticated image immediately, without internal buffering, for storage in on-chip memory within the PS. The master A9 polls the final authentication status from the configuration logic to ensure the image was properly authenticated. If it fails, the master A9 will trigger a system secure reset. Once the PS image has successfully loaded, control is turned over to the first-stage boot loader. Based on the user application, the boot loader could then either start processing, configure the programmable logic, load additional software or wait for further instruction from an external source.

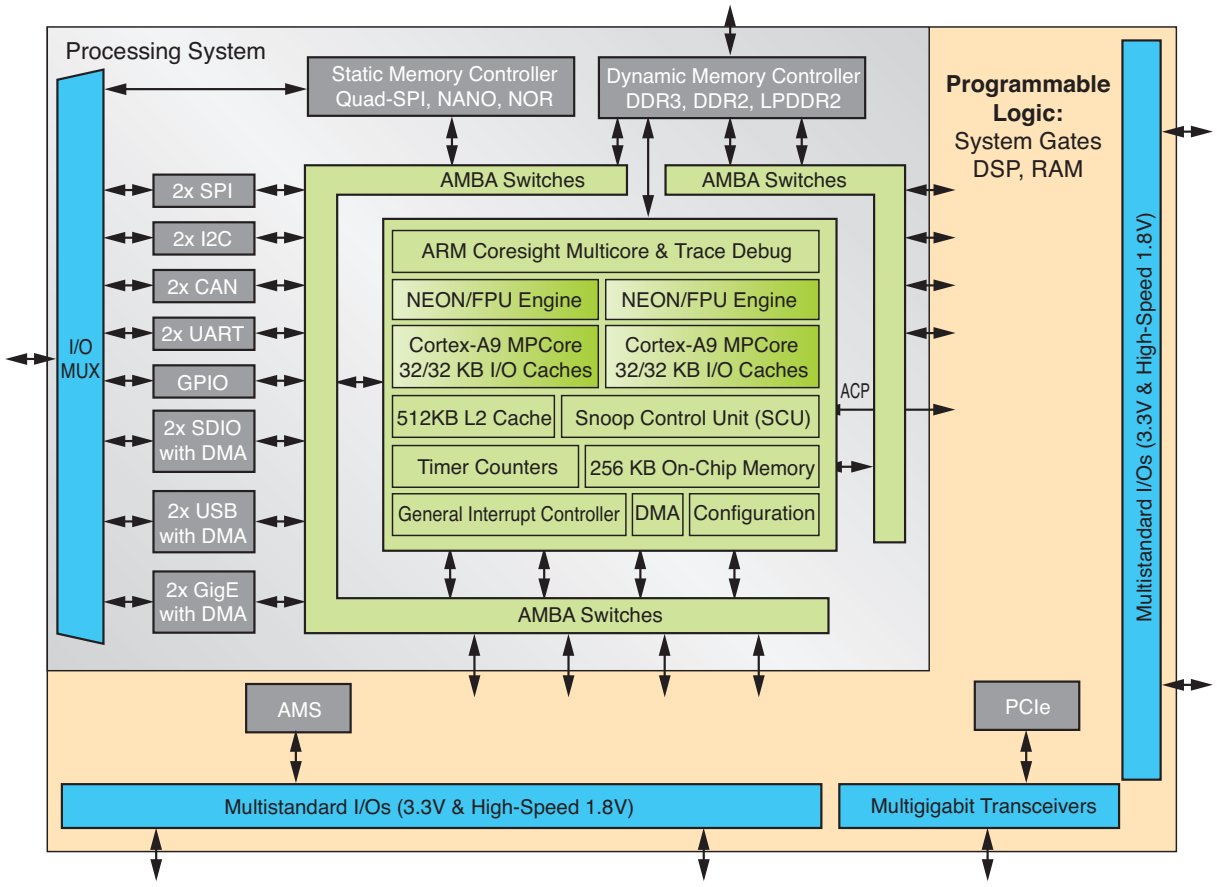


Figure 3 – Zynq-7000 EPP block diagram details the many industry-standard peripherals and memory interfaces that are available.

The programmable logic of the Zynq-7000 EPP can combine the digital RF and the logic of the modem, black and red FPGAs. This is possible not because of a large programmable logic fabric, but through the use of partial reconfiguration and the Xilinx Isolation Design Flow.

SECURE RUNTIME ENVIRONMENT

The Zynq-7000 EPP provides a secure runtime environment by incorporating ARM TrustZone technology throughout the device. TrustZone provides content protection by enabling software tasks to run in memory areas that are segregated, keeping the content safe from unauthorized reading or writing. Eliminating access by other tasks creates and maintains a trusted processing environment. TrustZone is built into the ARM A9 processors and each element within the PS.

In addition, the AMBA® TrustZone signals are extended into the programmable logic to allow the development of trusted master and slave devices within the PL. Since complete, secure programmable logic configuration support is provided, these user-developed master/slave devices can be completely trusted, as is any other hardened block within the PS. When used with TrustZone software, the Zynq-7000 EPP can enable a secure system capable of handling keys, private data and encrypted information.

PROGRAMMABLE LOGIC

The programmable logic within the Zynq-7000 EPP is built using Xilinx's 28-nm high-performance, low-power process technology. Four devices are available within the Zynq-7000 family, with logic cells ranging from 28,000 to 235,000. Like the other devices in the Xilinx 7 series family, the Zynq-7000 EPP's programmable logic provides configurable block RAM, programmable DSP functions, hardened Gen2 PCIe® (larger devices) and Agile Mixed Signal (AMS) technology. AMS technology provides dual 12-bit 1-Msample/second ADCs, dual independent track-and-hold amplifiers, on-chip voltage reference and thermal supply sensors, and external analog input channels.

As seen in Figure 2, the programmable logic of the Zynq-7000 EPP can combine the digital RF and the logic of the modem, black and red FPGAs. This is possible not because of a large programmable logic fabric, but through the use of partial reconfiguration of the PLand the Xilinx Isolation Design Flow.

PARTIAL RECONFIGURATION

Partial reconfiguration of the Zynq-7000 EPP PL takes the flexibility provided by normal FPGA technology a step further by allowing the modification of an operating design by reconfiguring portions of the programmable logic to perform a different function. After you have fully configured the PL with a complete configuration file, you can download partial configuration files to modify reconfigurable regions.

The programmable logic consists of static logic and reconfigurable logic. The static logic remains functioning and is completely unaffected by the loading of a partial configuration file. The reconfigurable logic is replaced by the contents of the partial configuration file. Partial reconfiguration files download without compromising the integrity of the applications running on those parts of the device not being reconfigured. Partial reconfiguration allows time-multiplexing of different hardware functions dynamically on a single Zynq-7000 EPP device. You can identify, isolate and implement time-independent functions as reconfigurable modules and swap them in and out of a single device as needed. This reduces the need to have a Zynq-7000 EPP with programmable logic large enough for all of the required functions that your design may need at different times. The programmable logic need only be large enough for the largest function needed at any one time. Software-defined radio, by dint of its mutually exclusive functionality, can directly benefit from this technology and see a dramatic improvement in flexibility and resource usage, and a reduction in static power.

The Zynq-7000 EPP architecture further extends the flexibility of partial reconfiguration by allowing software running in the PS to reprogram portions of the programmable logic via partial reconfiguration. Since, for secure designs, the PS and PL images are initially authenticated and trusted, the partial reconfiguration file loaded under PS or PL control can be either encrypted or unencrypted. So, not only can you swap the software processing algorithms in and out for different waveforms, but you can do the same for the corresponding programmable logic to implement those algorithms as well.

ISOLATION DESIGN FLOW

With the development of the Isolation Design Flow (IDF), red and black pro-

cessing logic can now reside on the same FPGA device, allowing designers of the cryptographic portions of SDRs to realize the full capability of programmable logic. Xilinx developed the Isolation Design Flow to allow independent red and black functions to operate on a single device and to eliminate the requirement for separate red and black FPGAs. Examples of such single-chip applications include redundant Type-I encryptors, resident red and black data, and functionality operating on multiple independent levels of security.

The Isolation Design Flow makes it possible to implement multiple physically isolated or independent

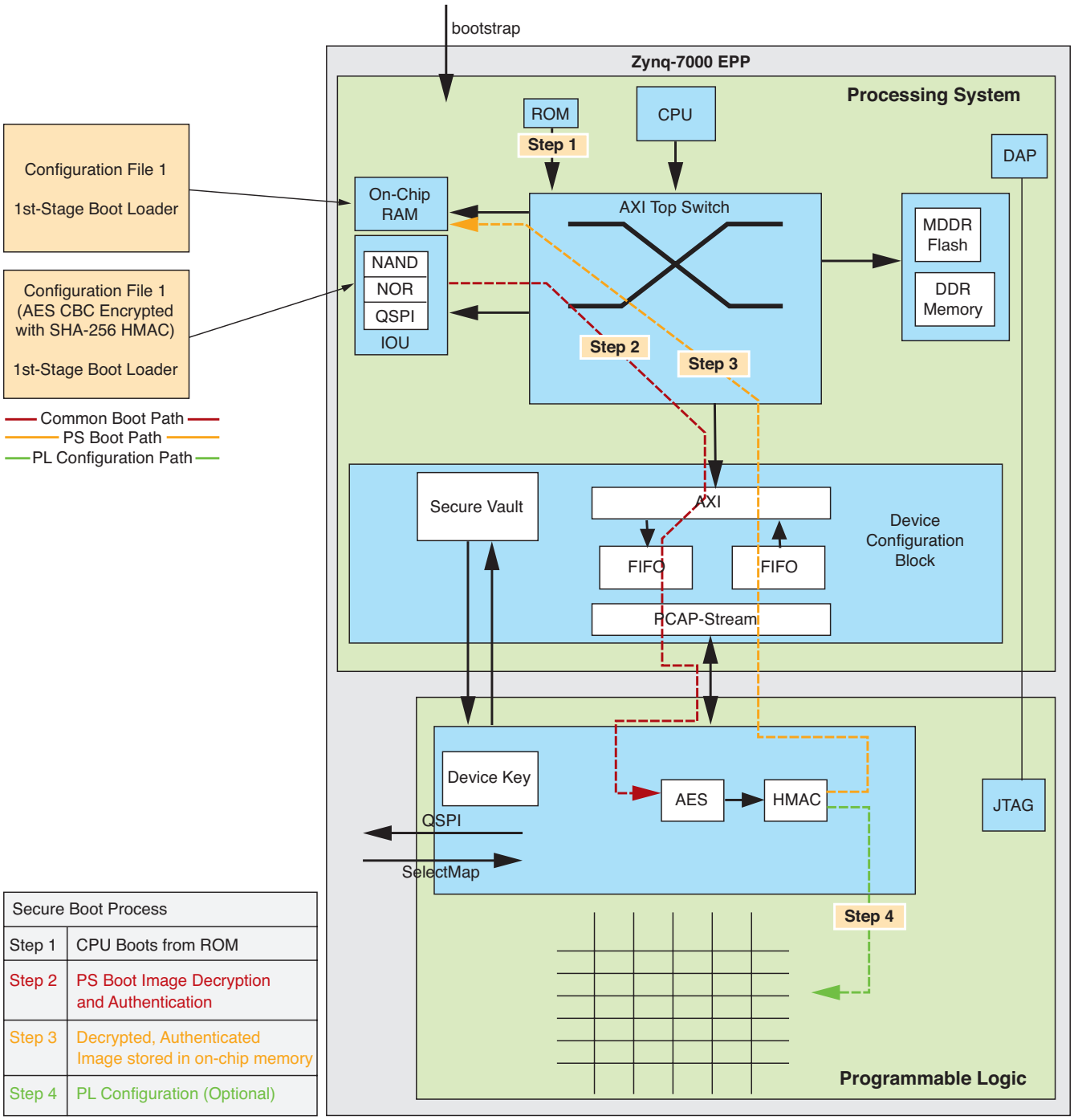


Figure 4 – Zynq-7000 EPP's secure boot process controls the device's security.

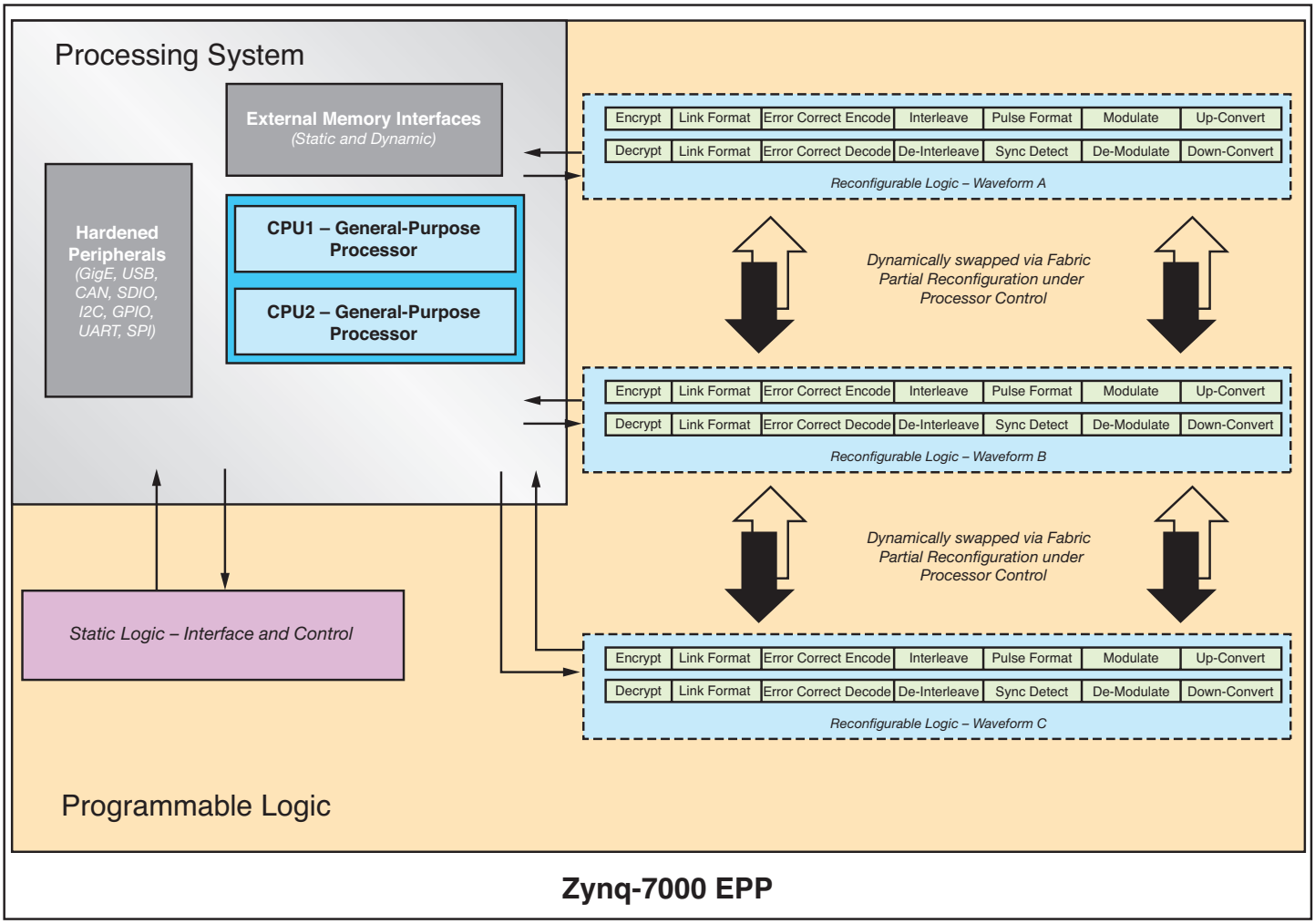


Figure 5 -- Use of Zynq-7000 EPP partial reconfiguration and the Isolation Design Flow to support multiple waveforms

functions within a single FPGA device, utilizing a fence of unused device components between each function. Each isolated function is fenced in, generating isolated regions within the device. The flow uses early floorplanning, modular design, modular synthesis and adherence to a set of guidelines and considerations to guarantee isolation between desired functions. Once a design is implemented, you can use the Xilinx Isolation Verification Tool (IVT) to visualize the modules and fence, along with verifying that you have successfully implemented the design rules for isolation.

By defining each of the waveform processing blocks as reconfigurable logic, the programmable logic within the Zynq-7000 EPP only has to be large enough to contain one waveform processing block at a time. The use of the Isolation Design Flow allows the encryption and decryption (red and black) processing logic to coexist within each waveform processing block. Partial reconfiguration allows different waveform processing blocks to be dynamically swapped in and out of the Zynq-7000 EPP PL under software control as shown in Figure 5.

Depending on the application, this same SDR can support future waveforms by developing the necessary software and waveform processing blocks that can be swapped into the Zynq-7000 EPP's PS and PL as reconfigurable logic, extending the overall product life cycle of the SDR.

POWER-SAVING FEATURES

The Zynq-7000 EPP supports many features to lower the overall system power consumption, such as the processing system's power-only mode, sleep mode and peripheral independent clock domains. Designers can use these features to significantly reduce the dynamic power consumption of the device during idle periods.

The Zynq-7000 EPP's processing system and programmable logic are on two independent power rails, each with its own dedicated power supply pins. It supports a PS power-on only mode, but does not support a PL-only mode. User software in the processing system can turn the programmable logic fabric power on or off at any point to reduce the static power required. When the PS

is powered off, it holds the PL in a permanent reset condition until the PS comes out of reset.

In sleep mode (wake from interrupt, wake from exception), a single ARM A9 processor is running at approximately 10 MHz, interconnects are shut down, DDR is in self-refresh mode, all DDR termination is off and processing system peripherals are in clock-gating mode except for the selected wakeup device (CAN, Ethernet or GPIO). The dynamic power consumption in sleep mode will come from only a small part of the CPU circuit used to monitor the wakeup interrupt, the snoop control unit and the wakeup peripheral device. You can also shut down the PL power during sleep mode, resulting in further reductions in static power.

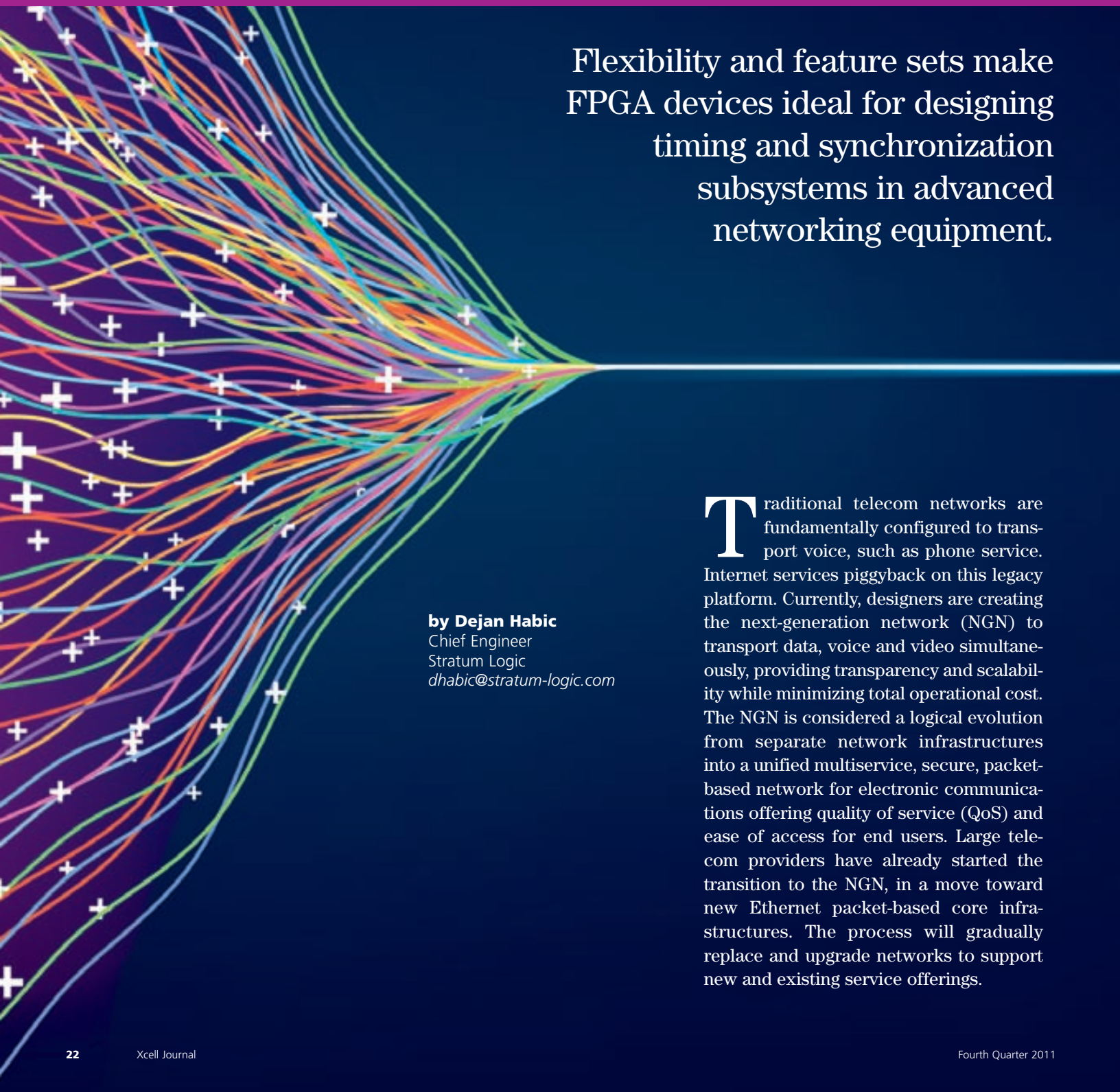
The Zynq-7000 EPP supports many clock domains, each with independent clock-gating control. You can shut down the unused clock domains in order to reduce dynamic power. Each of the peripherals within the processing system (except the GPIO, due to its small size and low speed) is on an independent clock domain, each with clock-gating control. You can turn them off via software through the system-level control registers.

SECURE INTEGRATION

Continuing the tradition of secure integration, a single Xilinx Zynq-7000 EPP now provides the processing power and logic formerly supplied by five or more separate devices for the processing and logic functions in tactical radios and commercial SDRs. The device delivers the capability and flexibility to reuse the same logic resources to process various waveforms by means of dynamically redefining some or all of the logic functions within the programmable logic, therefore enabling support of multiple waveforms and future waveforms.

Merging several devices into one Zynq-7000 EPP can result in a significant reduction in overall size, weight, power and cost of the SDR. Furthermore, the Zynq-7000 EPP provides several features that enable additional reductions in both static and dynamic power consumption, making it an ideal solution for a flexible, secure commercial or tactical SDR.

FPGAs Synchronize Next-Generation Networks



Flexibility and feature sets make FPGA devices ideal for designing timing and synchronization subsystems in advanced networking equipment.

by Dejan Habic
Chief Engineer
Stratum Logic
dhabic@stratum-logic.com

Traditional telecom networks are fundamentally configured to transport voice, such as phone service. Internet services piggyback on this legacy platform. Currently, designers are creating the next-generation network (NGN) to transport data, voice and video simultaneously, providing transparency and scalability while minimizing total operational cost. The NGN is considered a logical evolution from separate network infrastructures into a unified multiservice, secure, packet-based network for electronic communications offering quality of service (QoS) and ease of access for end users. Large telecom providers have already started the transition to the NGN, in a move toward new Ethernet packet-based core infrastructures. The process will gradually replace and upgrade networks to support new and existing service offerings.

This migration creates many technical challenges, uppermost among them the synchronization of network requirements. Traditionally, circuit-switched networks such as Sonet and SDH distribute a high-quality clock and timing source throughout the network, while Ethernet networks do not require such a strict clock hierarchy. However, the need for a method to synchronize these networks is becoming a specific requirement for carriers. The key challenge in realizing the NGN will be to standardize, implement and deploy a quality solution that enables interworking of all existing and new networks. Time and frequency alignment, also called synchronization, is critical for ensuring QoS for applications such as wireless, voice, real-time video and data over a converged network. Embedding the sync-and-timing function along with the other hardware in FPGA creates a flexible, programmable and low-cost solution that meets the highest telecommunication equipment standards.

THREE PROMISING CHOICES

At present, the three most promising technologies that distribute accurate timing and synchronization signals throughout the new networks are Synchronous Ethernet, the Precision Time Protocol (IEEE-1588) and a GPS clock. These technologies are well covered in a range of international recommendations and standards promulgated by the ITU-T and IEEE. The list is long, and includes ITU-T G.8261 (Timing and Synchronization Aspects in Packet Networks), ITU-T G.8262 (Timing Characteristics of Synchronous Ethernet Equipment Slave Clock) and ITU-T G.811 (Timing Characteristics of Primary Reference Clocks), as well as IEEE 1588 and many others.

The broadband wireless backhaul network represents the most challenging application for synchronization over the packet-switched networks, due to stringent mobile synchronization requirements driven by an optimal use of rare radio resources under high-speed mobility requirements. Two types of technology are the most prevalent duplexing schemes in broadband wireless networks: Frequency-division duplex (FDD) technologies require only network frequency synchronization, while time-division duplex (TDD) technologies require phase (and frequency) synchronization. Table 1 summarizes the mobile network requirements at the radio interface for different technologies. Video, circuit-emulation services over Ethernet and many other applications also require tight synchronization.

Radio system	Frequency accuracy	Time accuracy
GSM	50 ppb	NA
UMTS FDD	50 ppb	NA
UMTS TDD	50 ppb	2.5 µs
CDMA2000	50 ppb	3 µs
TD-SCDMA	50 ppb	3 µs
LTE FDD	50 ppb	NA
LTE TDD	50 ppb	3 µs
WiMAX 802.16e	8 ppm	0.5-25 µs

Table 1 – Wireless technologies and their synchronization requirements

Most existing designs use specific silicon devices to perform synchronization and timing functions. If GPS clock synchronization is required, designers can find on the market only OEM modules with specific functionality. All these components always carry a high unit cost, high integration cost and single-vendor dependency, and sometimes lack the features and performance to meet the application requirements.

Great flexibility, IP core availability and rich clocking resources make Xilinx® FPGA devices ideal for the implementation of synchronization and timing subsystems. In addition, most network equipment already has an FPGA on board with enough spare resources and logic to implement a synchronization subsystem that fully meets the requirements of the particular application. Designers can now embed complex synchronization circuitry in a single FPGA, along with existing hardware, by combining general and proprietary logic IP cores. The circuitry can potentially combine Synchronous Ethernet, IEEE-1588 and a GPS clock, providing the highest possible timing performance for the lowest cost.

SYNCHRONOUS ETHERNET

Over the past decade, with the emerging prominence of Ethernet in telecommunications networks, carriers have been evolving their legacy circuit-switched systems to Ethernet packet-based systems. However, Ethernet was not designed for the transport of synchronization signals, which are key requirements for some of the existing and future applications such as TDM emulation, mobile backhaul and next-generation mobile networks. Synchronous Ethernet (SyncE), described in ITU-T G8262, represents one of the key developments in the evolution of Ethernet into a carrier-grade technology suitable for telecommunication wide-area networks.

In traditional Ethernet, data is transmitted continuously. The physical-layer transmitter clock is derived from an inexpensive ±100-ppm crystal,

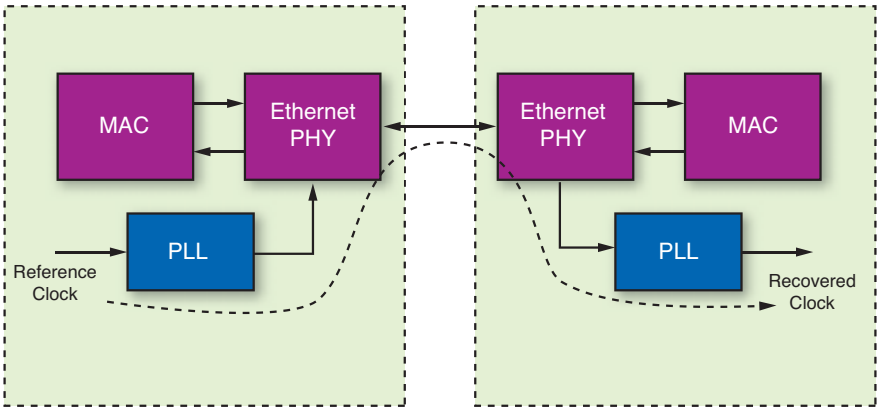


Figure 1 – Synchronous Ethernet network element and clock distribution

and the receiver locks onto it. There is no need for long-term frequency stability and consistency between the frequencies of different links, as the data is packetized and buffered.

Designed to distribute a frequency reference within an IEEE 802.3 Ethernet network, SyncE is based on a hierarchical synchronization method using a synchronous physical layer, similar to the synchronous optical networks (Sonet/SDH). In SyncE, the physical-layer transmitter clock derives a signal from a high-quality frequency reference by replacing the crystal with a frequency source traceable to a primary reference clock (Figure 1). The receiver at the other end of the link automatically locks onto the physical-layer clock of the received signal, thus itself gaining access to a highly accurate and stable frequency reference. This process does not affect the operation of any of the Ethernet layers.

In essence, supplying one Ethernet network element with a primary reference clock (PRC) and employing an Ethernet PHY with well-engineered timing-recovery circuitry similar to those used in Sonet/SDH networks, we could set up a fully time-synchronized network. This methodology provides access to a highly accurate and stable frequency reference to applications that need it. Although it requires hardware changes in equipment, SyncE is not influenced by impairments introduced by the higher levels of networking technology such as packet loss or packet delay variation. That's a big advantage over other methods that rely on sending timing information in packets over an unlocked physical layer. Hence, the SyncE frequency accuracy and stability may be expected to exceed those of networks with unsynchronized physical layers.

The SyncE timing-and-synchronization device is a digital phase-locked loop (PLL) that supports free-run, locked and holdover modes and generates multiple synchronous

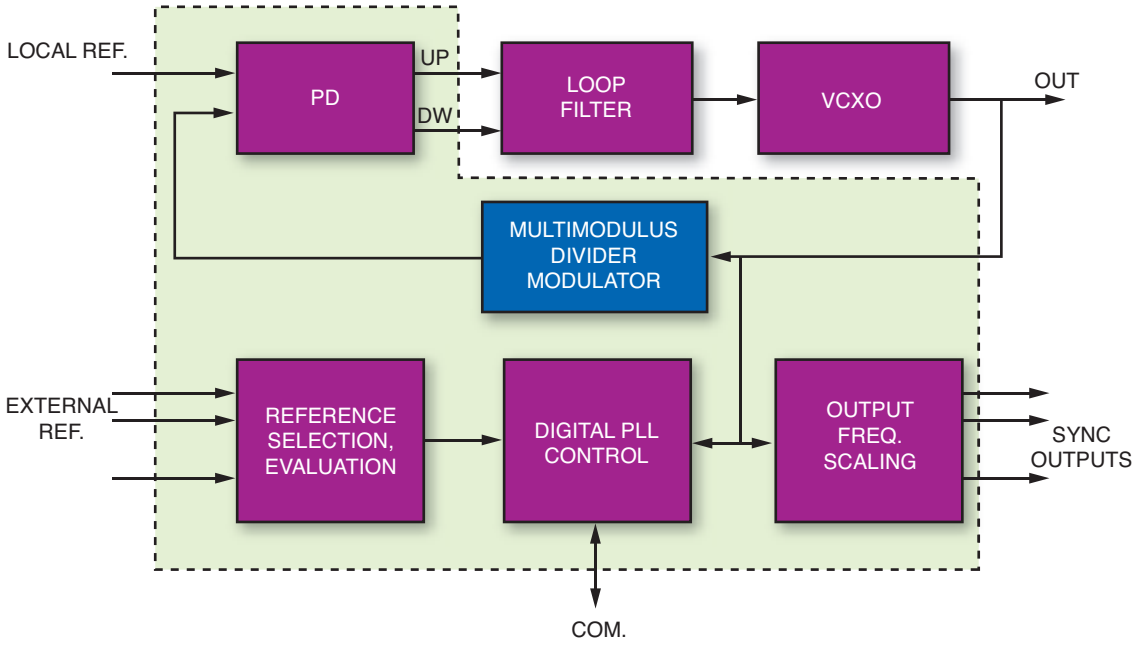


Figure 2 – Synchronous Ethernet clock

You can build a fully integrated Synchronous Ethernet solution using Xilinx devices with minimum external components. A typical FPGA implementation comprises a digital PLL, MicroBlaze system and frequency output generator.

clocks. An internal state engine controls mode selection automatically; alternatively, you could set it externally. The free-run mode occurs when the device is unlocked to either of the inputs, and output accuracy depends on the local oscillator specified as ±4.6 ppm. The holdover mode occurs when the SyncE device has lost its reference inputs and utilizes stored timing data to control the output frequency—typically, while the network synchronization is temporarily disrupted.

The locked mode is when the output of the SyncE is phase-locked to any of the selected input references. Locked mode is typically used when a slave clock source is synchronized to the network. To provide sufficient filtering of jitter and wander at reference inputs, the digital PLL is designed as a programmable narrow-loop-band device. During mode switching, its output phase must be precisely controlled to avoid disruption and signal loss.

A SyncE device continuously monitors all input references for their presence and quality. If a system requires redundant operation, the SyncE device should support master-slave configuration of two devices. Depending on the application, the SyncE devices should support different output frequencies.

You can build a fully integrated universal synchronization solution for Synchronous Ethernet that complies with ITU-T G.8262 using Xilinx FPGA devices with minimum external components. The newer Xilinx families with an improved clocking feature set, such as Spartan®-6 or Virtex®-6, are ideal for the application. A typical FPGA implementation of SyncE comprises a digital PLL, MicroBlaze™ system and frequency output generator, as shown in Figure 2. Using an external VCXO will help in achieving a very low-jitter output signal. The VCXO is running in a fractional PLL that designers lock to a selected input reference. They achieve this lock by changing the fractional ratio of the high-resolution sigma-delta modulator.

For its part, the MicroBlaze system contains a CPU, block RAM, timer, SPI or UART, I/O ports, high-resolution digital phase detector and a frequency output generator. The MicroBlaze is running application software with multiple tasks including SyncE state engine, reference monitoring, communications and so on. The frequency output generator takes the output of the digital PLL and generates multiple additional frequencies using FPGA clocking resources (CMT) and programmable dividers. Multiple CMT blocks can sysnthesize a wide range of frequencies including standard T1, E1, Sonet/SDH and Ethernet. An external TCXO or OCXO serves as the local reference.

THE PRECISION TIME PROTOCOL (IEEE-1588)

The IEEE-1588 Precision Time Protocol (PTP) is a standard for synchronizing independent clocks running on separate nodes of a distributed packet networked system to a high degree of accuracy and precision. The protocol is independent of the networking technology, and the system topology is self-configuring. Originally, the PTP was designed for applications in industrial automation and instrumentation or relatively small and local networks that require precise synchronization. With the emergence of a new generation of telecom networks in the past couple of years, designers are taking a new look at the PTP. The new version 2 of the standard provides better performance over wide-area packet-based networks. Among the many improvements are concepts of a transparent clock, boundary clock, larger maximum message rate and unicast support. The IEEE-1588 clocks, located in network nodes, are organized into a master-slave hierarchy.

A master clock exchanges two-way timing packets over Ethernet, while slave clocks embedded in the equipment require synchronization. Each slave synchronizes to its master based on four messages exchanged among them: sync, follow up, delay request and delay response. The master is sending multicast messages to the slaves, while slaves are responding to the master by unicast. In each node the messages are time-stamped on both the receiving and transmitting paths, and time stamps are embedded in the subsequent messages. To calculate offset and delay (that is, time relative to the master), the slave uses four time stamps and assumes that delays for messages traveling between master and slave are equal.

Generally, there are two types of IEEE-1588 implementations: software only and hardware-assisted software. In software-only implementations, the whole protocol executes at the application level, including the processing of the time stamps and control of the

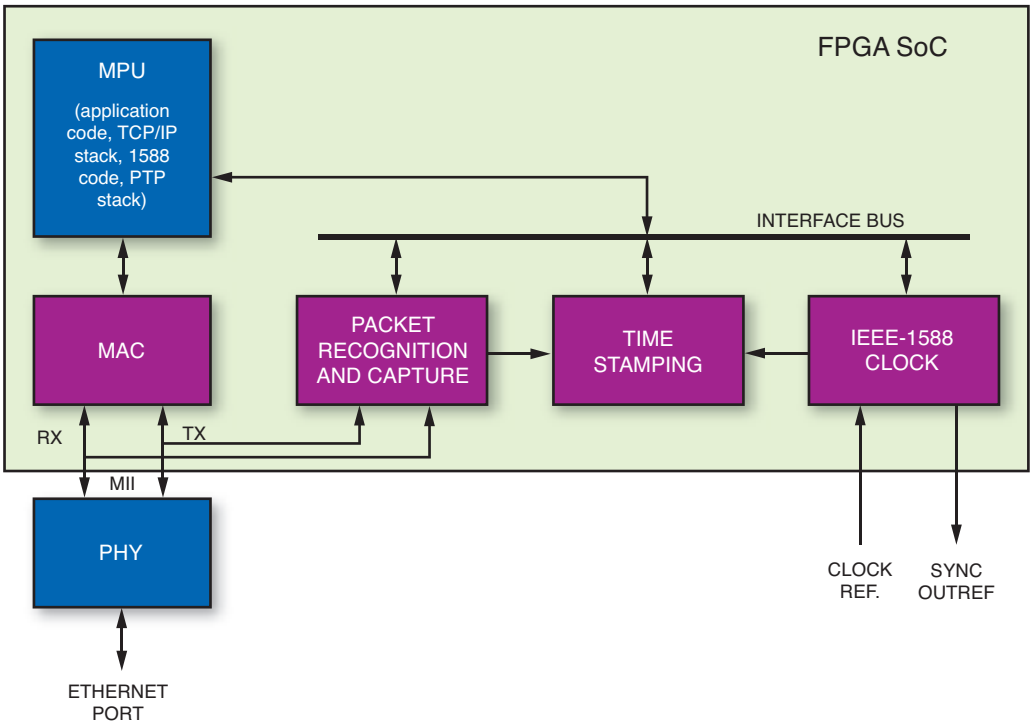


Figure 3 – Diagram of the IEEE-1588/PTP clock

clock. Since the operating system and applications introduce additional time errors into the time stamps, this implementation provides accuracy on the order of hundreds of microseconds to milliseconds, which is often not enough for telecom applications.

For higher accuracy, a hardware-assisted software implementation is the way to go. Since the hardware here performs the time-stamping close to the physical interface, this method can provide accuracies in the order of hundreds of nanoseconds. A typical hardware-assisted implementation connects a time-stamp unit (TSU) at the Media Independent Interface (MII) between PHY and MAC (Figure 3). The TSU is responsible for indentifying PTP messages, time-stamping each of them at the start-of-frame delimiter and, finally, recording time stamps along with some relevant attributes into the memory buffer or registers.

In addition to the TSU, this approach also implements the IEEE-1588 clock block in hardware. The clock maintains and distributes the current time in seconds (48-bits value) and subseconds (32-bits value). The IEEE-1588 clock is updated each cycle of the local oscillator, with the PTP software that's running in the processor controlling the update rate. The PTP software runs several tasks. Among other things, it processes time stamps, controls the clock synchronization, receives and sends PTP messages, and services the protocol stack and state engine.

A critical part of the IEEE-1588 implementation is selection of the local oscillator. You must specify the stability of the oscillator so that its drift caused by temperature and aging effect is tolerable between two corrections of the PTP clock. Often the oscillator specifications boil down to a trade-off between cost and performance.

Today's telecom networks contain many switches, routers or other network equipment, which increase the size and complexity of the network.

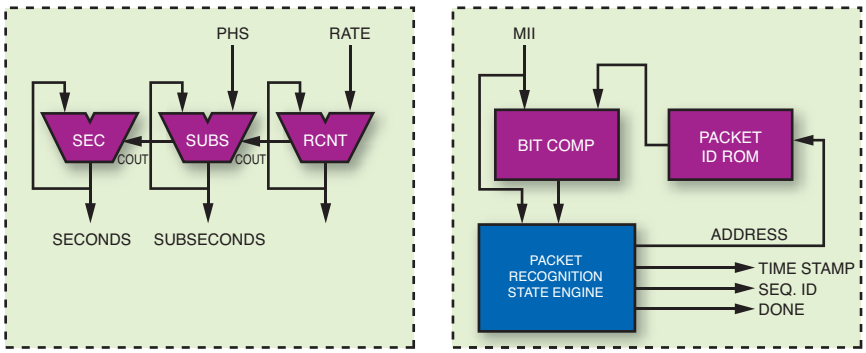


Figure 4 – IEEE-1588/PTP clock (left) and packet-recognition (right) hardware blocks

The GPS clock can synchronize both system timing and transceiver frequency, and is virtually fail-safe. It generates timing signals whenever there is power, and never needs to be recalibrated.

Each of these network elements introduces delays and fluctuations of the PTP message delivery. These fluctuations, called packet delay variation (PDV), directly affect the accuracy of the clock synchronization, since the synchronization algorithm includes the propagation delay calculation. Some routers can introduce large PDV in the order of milliseconds and more, considerably decreasing synchronization accuracy. The system must employ sufficient filtering of the PDV if the PTP is to fully meet requirements for telecommunication applications. Generally, packet delay variation can be divided into three components: transmission delay, processing delay and buffering delay.

Transmission delay is the result of the signal speed between two ports, while processing delay is the result of the processing of a timing packet within a network element. The buffering or queuing delay is the total amount of waiting time of the timing packet within different buffers or queues in a given network element before being processed. These three components have different PDV ranges. The transmission delay generates a PDV in the range of submicroseconds, while the processing delay is in the order of 1 to 10 µs. The last component, buffering delay, is often the main generator of PDV, with a 10- to 10,000-µs range.

Different filtering techniques are available at the slave clock to minimize PDV. The two most common ones use the moving average or the exponentially weighted moving average. Both techniques are based on estimating average packet time arrival, with the assumption that the PDV distribution is fixed, independent and equally distributed. The efficiency of these filtering algorithms greatly depends on network traffic load and topology. To make the PDV filtering more effective, designers use other network management techniques as well, such as increasing timing-packet rate and randomizing message transmission around a given mean value.

From a hardware perspective, FPGA implementation of the IEEE-1588 protocol is relatively straightforward. The system consists of a MicroBlaze CPU, RAM, timer, MAC, packet-recognition block, time-stamping unit and clock driven by an external oscillator. The MicroBlaze runs the PTP stack software including the filtering and clock-disciplining algorithm.

Three hardware blocks central to this architecture are not part of the standard Xilinx IP LogiCORE™ library: the PTP clock, packet-recognition block and time-stamping unit (TSU). The PTP clock is implemented as three cascaded accumulators, as shown in Figure 4. The RCNT register accumulates fractional subseconds based on the value of the RATE register, which defines the PTP clock's average speed. With each clock cycle, the SUBS register accumulates the fixed value of the PHS register plus the carry-out of the RCNT accumulator. The carry-out of the SUBS (30-bit) register accumulates into the SEC register (32 bits).

The packet-recognition block monitors the MII bus between the MAC and the PHY to identify IEEE-1588 packets and send signals to the TSU. Since packet monitoring is performed on both the receiving and transmitting paths, the design implements two identical packet-recognition blocks. To detect start-of-frame and generate the time-stamp signal, the block's state engine counts and compares subsequent bytes and determines if the message is valid. The block also generates an ID related to the specific IEEE-1588 message. Finally, the packet-recognition block generates the DONE signal to the MCU. The TSU block's task is to capture the time of the PTP clock when the time-stamp signal is asserted. For the best performance, the local oscillator should be either a TCXO or OCXO.

THE GPS CLOCK

For some time, telecommunication systems around the world have been using the Global Positioning System for accurate timing and synchronization. GPS has 24 satellites positioned in six earth-centered orbital planes, each containing four atomic clocks. Each satellite transmits information of its location. This data is modulated onto the carrier frequency and repeated at very precisely controlled intervals regulated by the atomic clocks. The GPS receiver on the ground receives and decodes these signals, effectively synchronizing itself to the atomic clocks on the satellites. A GPS clock that is used in telecommunication synchronization combines a GPS receiver with an antenna, digital PLL with disciplining software and high-quality stable oscillator. The disciplining software controls and calibrates the oscillator to remove small biases in the frequency.

The GPS clock can synchronize both system timing and transceiver frequency, and is virtually fail-safe. It generates timing signals whenever there is power, and never needs to be recalibrated.

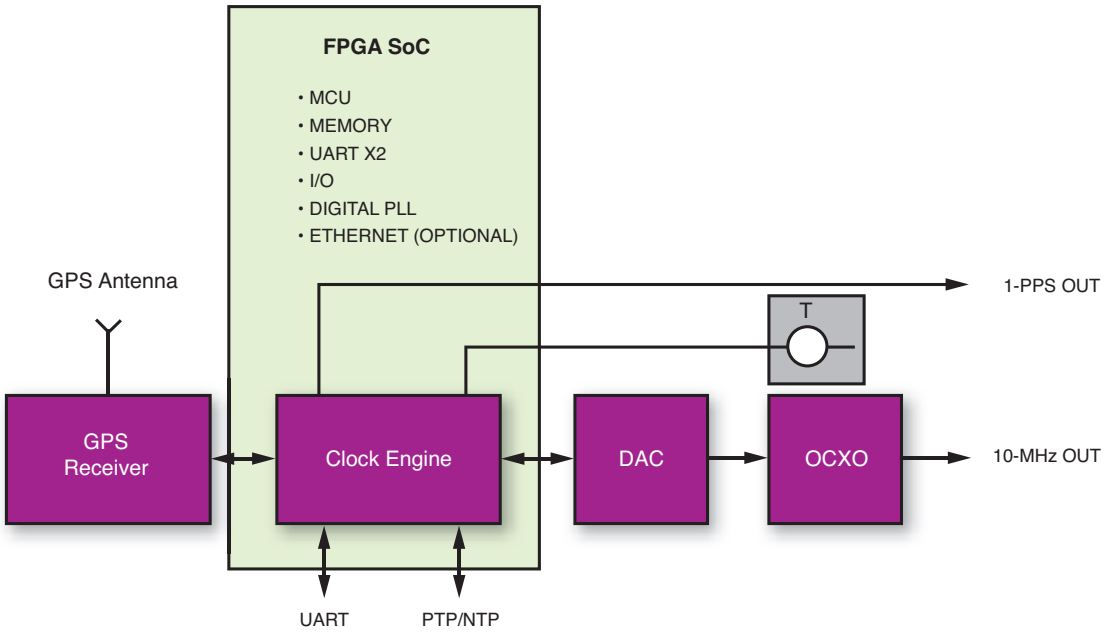


Figure 5 – Typical implementation of a GPS clock

This makes it possible to integrate GPS into many installations requiring high-quality timing subsystems or to simply use it as a primary reference source. In the past the cost of using GPS was still relatively high, especially considering antenna installation. However, as prices came down, the GPS clock became a more viable solution for timing and synchronization in next-generation network systems.

In addition, there are other global navigation satellite systems (GNSS) that can be effectively used in time and synchronization applications, such as the Russian GLONASS system, the European Union's Galileo and China's Compass. There are already a few GPS receivers available on the market supporting multiple GNS systems. These multisystem receivers track a large number of satellites effectively, thus increasing timing accuracy.

Figure 5 shows a typical implementation of the GPS clock. Essentially a digital PLL, it utilizes application-specific software using a 1-pulse/second (pps) signal from the GPS receiver as a timing reference to discipline a local oscillator and provide a high-quality frequency and timing reference. The GPS clock operates in three modes: locked, free run and holdover. In free-run mode, the unit is unlocked to the reference input. Free-run mode is typically used when the GPS signal and history of timing data are not available, or immediately following system power-up, before achieving network synchronization. In this mode, output signals are not synchronized to the reference signal and their accuracy is based on the local oscillators. In the locked mode, the output is phase-locked to the GPS 1-pps reference signal and the output frequency tracks the input reference.

The digital PLL has very low bandwidth, typically in the millihertz range. It filters a major part of the noise of the GPS' 1-pps signal. A second- or even third-order loop filter with optimized gain, bandwidth and sometimes time-varying loop parameters is usually a good choice for stable tracking and sufficient for filtering of the GPS 1-pps reference signal. In holdover mode, the device either has lost or disqualified the GPS 1-pps signal and uses stored timing data, called history, to control the output frequency. Holdover mode occurs when the GPS signal is temporarily disrupted and no valid reference is available.

To minimize frequency and phase drift caused mostly by aging and temperature performance of the oscillator, this mode relies on an adaptive algorithm. Using an oscillator model created from data collected during locked mode, this algorithm calculates prediction of the frequency drift. The data set consists of ambient temperature, frequency and time values. Designers can use some variation of Kalman filtering or recursive implementation of linear regression for the adaptive algorithm.

Although the GPS clock uses the adaptive algorithm to compensate for oscillator drift, the stability of the output signal in holdover greatly depends on the stability of the onboard oscillator. Therefore, you should take special care in selecting the oscillator. For most NGN requirements, you will need an oven-controlled crystal oscil-

lator (OCXO), although sometimes you may be able to use a lower-cost oscillator. Also, take special care in selecting the GPS receiver, since not all receivers provide a sufficiently accurate signal of 1 pps.

An FPGA implementation of the GPS clock is straightforward. A Xilinx FPGA implements all digital hardware as a system-on-chip (SoC) complemented by a GPS receiver, high-stability local oscillator (OCXO) and additional analog circuitry, including DAC, temperature sensor and others. The MicroBlaze system comprises the CPU, BRAM, timer, SPI, two UARTs, I/O ports and high-resolution digital phase/frequency detector. To achieve high resolution of the phase detector, use a DCM generating a 300-MHz clock. Such high frequency ensures resolution of ±1.66 ns when using both edges. You can achieve higher resolutions, if needed, by implementing a TDC with a tapped delay line method in the FPGA.

The SPI provides an interface to the external DAC and digital temperature sensor. The first UART controls the GPS receiver while the second is used for communication with a host. The control software with the adaptive algorithm for the MicroBlaze system was written in C/C++.

As the transmission of telecommunications data is increasingly dependent upon a new generation of packet network transport, designers are coming up with new methods of time and frequency transfer. Three synchronization techniques—Synchronous Ethernet, the Precision Time Protocol and a GPS clock—are widely accepted and standardized by the industry.

Designing synchronization and timing subsystems for next-generation network equipment requires understanding and defining system requirements, selecting appropriate sync technology and implementing the solution. Using a Xilinx FPGA with a new clocking infrastructure, combined with a set of general and proprietary IP logic cores, makes the job much easier.

Accelerating Texture Mapping with Spartan-6 FPGAs

Once an application for custom ASIC cores, this demanding computer graphics process is now the province of low-cost FPGAs.

by Sébastien Bourdeauducq
Founder
Milkymist
sebastien@milkymist.org

A handheld box that generates special video effects for parties and concerts without the help of a computer is built around an FPGA rather than a specialized multimedia system-on-chip. In our Milkymist One, a Spartan®-6 FPGA implements almost the entire digital portion of the system. What's more, the FPGA is robust enough to handle texture mapping, a high-end graphics function that represents the most intensive data-processing task our system must perform. Texture mapping is the traditional realm of ASIC graphics-processing units and, before they existed, high-end workstations.

Disc jockeys, VJs and other event organizers can use the Milkymist One (Figure 1) at concerts, during festivals and in clubs to create an entertaining video installation. Connect a camera and a video projector, press the power button and seconds later, everything you film becomes live psychedelic effects of color and light. Point the camera at a dancer onstage, at people attending your party, at toys or other objects, and dazzle your audience with the effects. If no camera setup is available, the Milkymist One can produce purely generative effects that react to the ambient sound, making it an ideal option for bands, clubs and party organizers who want a turnkey solution for simple visual effects.

The device supports inputs from many sources: MIDI keyboards, USB computer keyboards, DMX desks and OpenSoundControl (OSC) clients. You can even use a smartphone to interact with the visual performance wirelessly, by connecting a WiFi router to the Ethernet port. Another option is to use the popular Arduino board, with its many sensor interfaces, to control the Milkymist One over MIDI.

We had to overcome significant challenges to design such a device. Our processing algorithm requires a considerable amount of computing power and memory bandwidth to process the video with a high frame rate and a low latency. Further, our device has to interface with multiple I/O protocols. For this application, many engineers would choose a multimedia system-on-chip that included a CPU and graphics acceleration. They would then need a number of external chips to handle all the interfaces. But by leveraging the power and flexibility of the Xilinx® devices, we were able to implement almost all the digital portion of our system in one Spartan-6 FPGA. This reduced the cost and chip count while greatly improving flexibility.

THE MILKYMIST ONE HARDWARE

Our Milkymist One system board (see Figure 2) is centered around a Xilinx XC6SLX45. This FPGA contains all the digital logic of our system, including a soft-core CPU, a memory con-

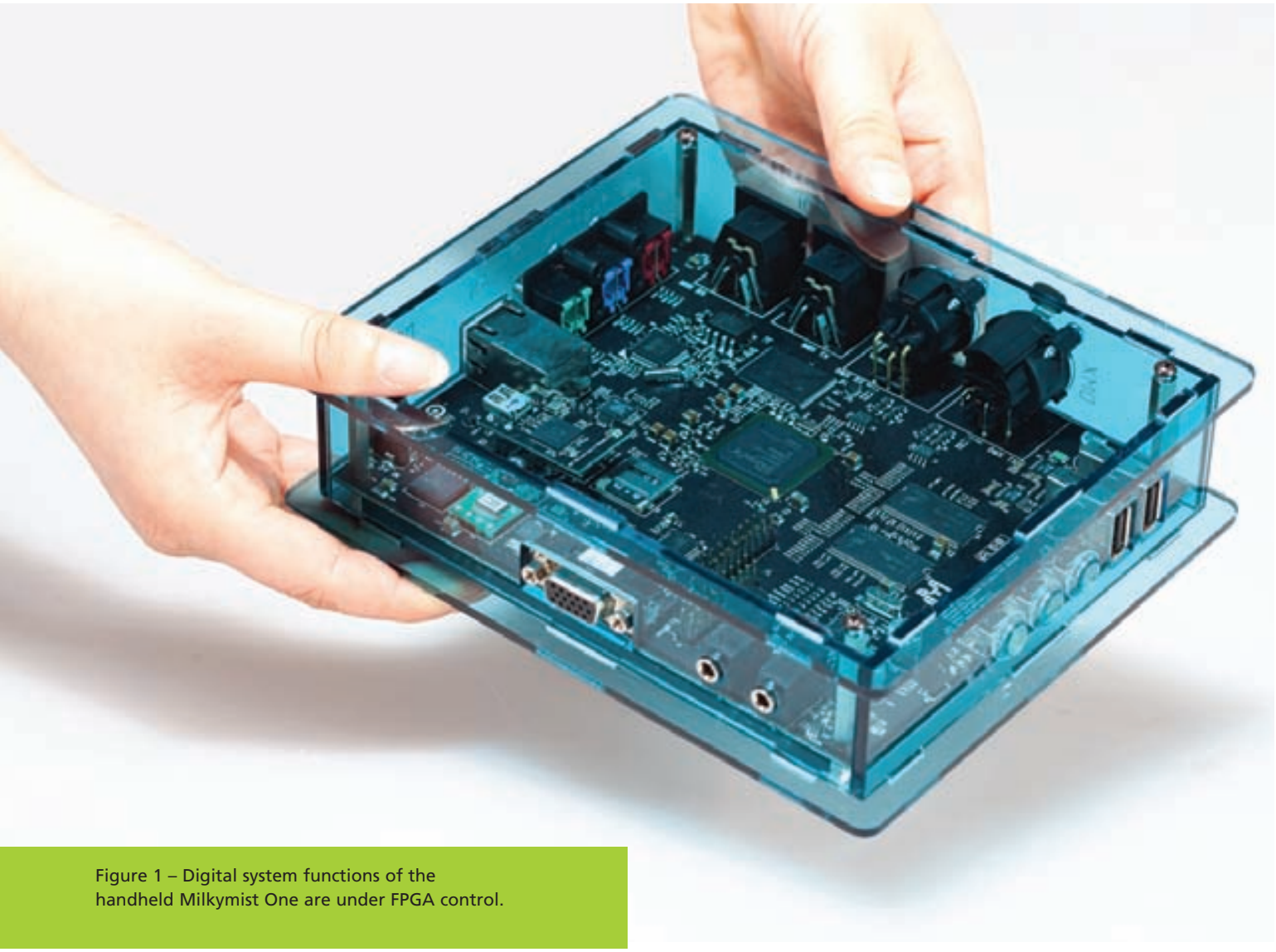


Figure 1 – Digital system functions of the handheld Milkymist One are under FPGA control.

The flexible DCM_CLKGEN primitive contained in the Spartan-6 allows the synthesis of many different frequencies for the pixel clock, enabling our device to support a large number of video modes.

troller, hardware accelerators and I/O peripherals.

The FPGA reads its configuration data from a NOR flash chip, using the "Master BPI" mode of the Spartan-6. The same flash chip then runs the boot loader using an "execute-in-place" schema, in which processor instructions are fetched from the NOR flash while they are being executed. The boot loader brings up the SDRAM and loads the application software. The same flash chip stores this application software and keeps user data using YAFFS2, a flash-optimized file system that provides wear leveling and journaling.

Our application software can download FPGA bitstream updates from the Internet and write them to the flash. Thanks to the MultiBoot feature of Spartan-6 FPGAs, if a failed Internet update should result in a corrupt bitstream, the system can fall back to a rescue "golden" bitstream programmed at our factory.

A pair of DDR SDRAM chips, directly connected to the FPGA, provide 128 Mbytes of system memory. To assist in meeting the timing requirements of this demanding interface, the Spartan-6 FPGA supplies double-data-rate I/O registers; runtime programmable delay-locked loops (with the DCMs); and I/O delay elements.

Our device supports two full-speed USB host ports. Again, the FPGA absorbs most of the hardware here. The Spartan-6 directly drives analog transceiver chips that simply convert LVCMOS 3.3-volt levels into perfectly USB-compliant signals. The serial interface engine and the host controller logic are implemented in the FPGA fabric. During prototyping, we were even able to successfully connect USB devices directly to the FPGA, using just resistors and USB connectors that we wired to the I/O expansion connector of a Xilinx ML401 development board.

For video output, the FPGA drives a triple digital-to-analog converter that generates the RGB components of the VGA port. The flexibility of the DCM_CLKGEN primitive contained in the Spartan-6 allows the synthesis of many different frequencies for the pixel clock, enabling our device to support a large number of video modes.

Also, we are currently looking into the synthesis of a composite video (CVBS) signal out of the VGA port. There are already some computer graphics cards on the market that use a low-cost passive adapter to connect CVBS devices to their VGA output. This is perfectly doable as well on a system that uses an FPGA to generate the raw color components. We would only need to implement a CVBS signal generator using digital signal-processing techniques, and feed the produced data into the VGA DAC. This would enable our device to easily connect to legacy video projectors and video mixing consoles that are still popular in the music and live-performance scenes.

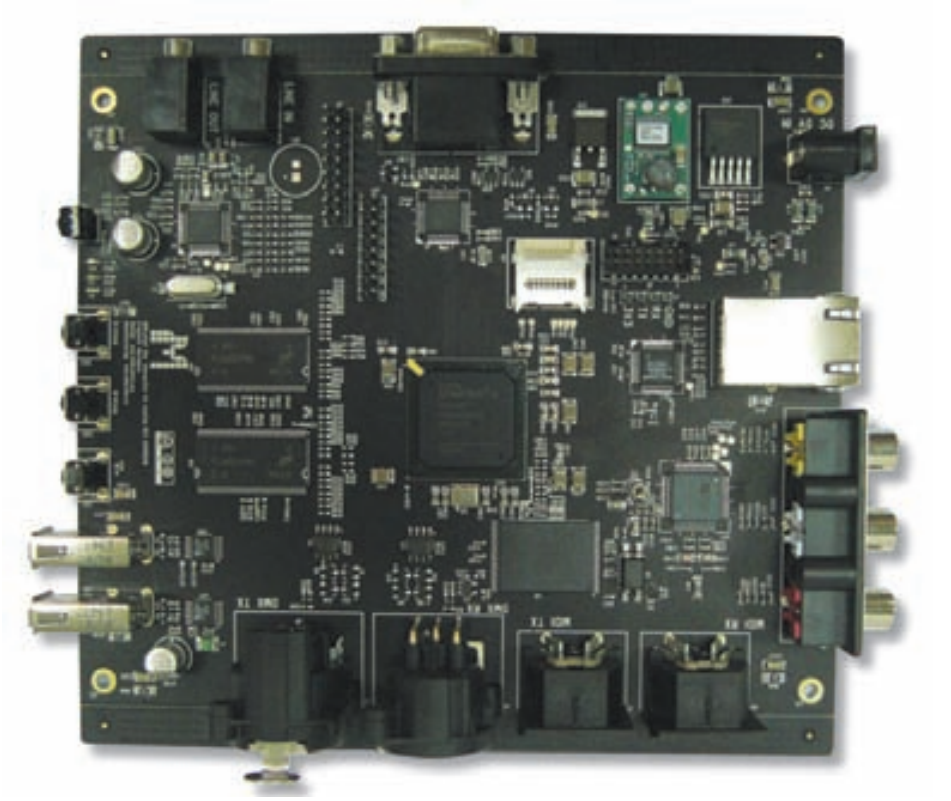


Figure 2 – The Spartan-6 FPGA resides at the center of the Milkymist One's printed-circuit board.

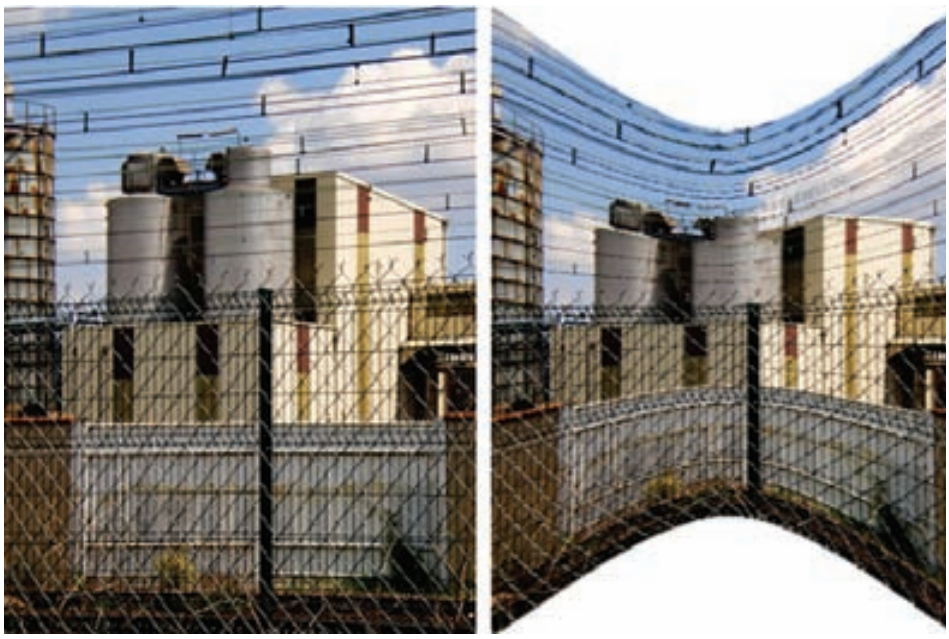
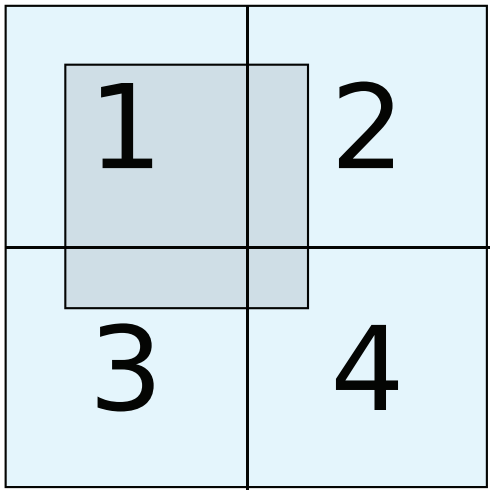


Figure 3 – Texture mapping, a common computer graphics operation found in accelerated 3D APIs, is typically used to draw textured 3D polygons. It can also distort an image, as seen here.

Our design connects the Spartan-6 to a pair of RS485 transceivers to provide DMX512 support. This protocol, which is used onstage to control lights, allows the device to synchronize the light ambiance with the visual effects. Here again, the complete DMX512 signaling system is within the FPGA, and the external components are basically analog.

To interact with popular controllers and sensors, our system also supports MIDI. Our implementation is similar to that of DMX512, with only analog external components. We also support Ethernet (using a PHY chip only), audio (through a common AC97 codec) and PAL, SECAM and NTSC video input.



1, 2, 3, 4: real texture pixels
Grayed box: wanted texture pixel with noninteger coordinates

The color of the resulting pixel is proportional to the surface of each real texture pixel it covers.

Figure 4 – In texture mapping, the results of the linear interpolation are usually not integer. For this reason, the texture should be sampled among four adjacent pixels and their colors averaged, a process called bilinear filtering.

Most of these peripherals are clocked from the FPGA, which synthesizes the necessary frequencies from a single 50-MHz source using its digital clock managers (DCMs). We have just two additional crystals on our board, and to reduce costs further, we are thinking about replacing them with more FPGA-generated clocks in a future PCB revision.

WHAT IS TEXTURE MAPPING?

Among all the data-processing tasks that the FPGA of the Milkymist device must perform, texture mapping is the most intensive. Texture mapping is a common computer graphics operation found in accelerated 3D APIs like OpenGL and DirectX. It is typically used to draw textured 3D polygons on the screen. It can also distort an image (see Figure 3), and we use it for this purpose.

Common graphics processing units perform texture mapping on triangles, and break down more-complex polygons into a series of triangles. The inputs to the algorithm are the 2D (possibly projected from the original 3D coordinates) positions of the three vertices of the triangle to be filled, and the 2D texture coordinates for these three vertices. The algorithm then draws a textured triangle pixel by pixel, by interpolating linearly the texture coordinates of the vertices for each pixel and then copying the texture pixel (also called texel) at these coordinates.

Texture mapping can implement image-processing operations like zooming, rotating or scaling by simply changing the positions of the vertices or the texture coordinates at each vertex. More often than not, the results of the linear interpolation are not integer, which means that the texture should be sampled among four adjacent pixels (see Figure 4). In this case, for a better rendering, the four pixels should be read and their colors should be averaged (with different weights depending on the fractional parts) in a process called bilinear filtering. Our applica-

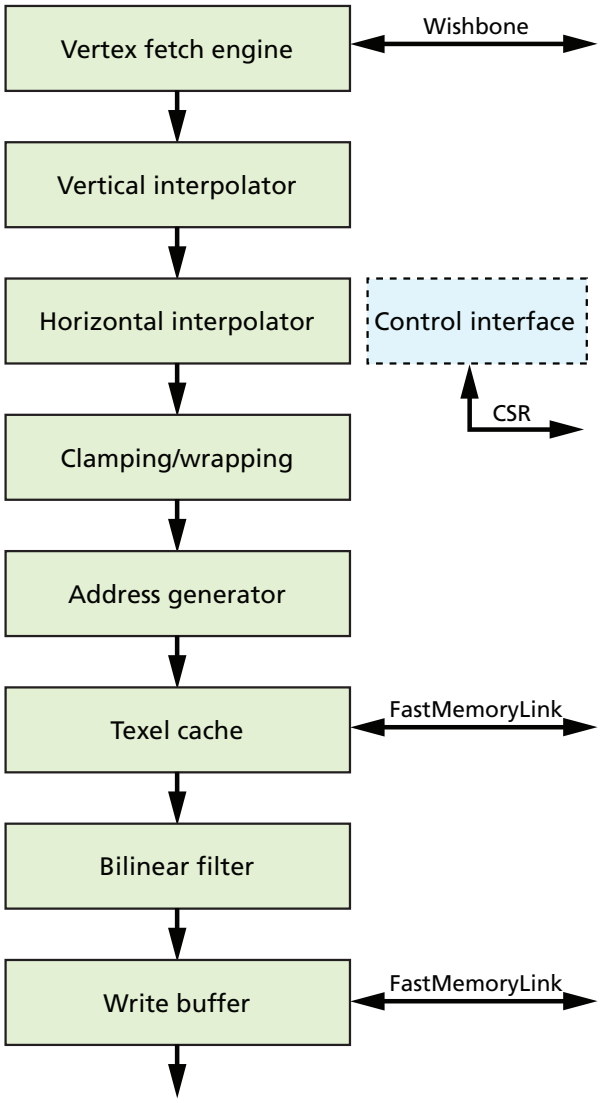


Figure 5 – Block diagram shows our deeply pipelined implementation of the texture-mapping algorithm.

tion requires bilinear filtering to obtain a good visual result.

Texture mapping, especially when bilinear filtering is desired, is a very compute- and memory-intensive process that precludes software implementations when performance is needed.

FPGA IMPLEMENTATION

It is expected that the memory latency for reading the frame buffer would be a performance-limiting factor. Instead of trying to alleviate its effects by using complex and potentially resource-intensive techniques such as advanced prefetching, we simply use a direct-mapped texel cache, for simplicity and fast hit times, and design the rest of the texture-mapping unit so that the memory read latency becomes the only limiting factor.

With a direct-mapped texel cache having a hit rate of 90 percent, a hit time of one cycle and a miss penalty of nine cycles, the average memory access time is 1.8 cycles. With an 80-MHz system clock, such a cache has a throughput of 44 megapixels per second, sufficient for our application.

To make sure that the memory access time is the only limiting factor, we designed the rest of the system to support a throughput of approximately one output pixel per clock cycle. This corresponds to a spatial implementation of the algorithm (that is, with little or no time-based resource sharing of the hardware components) but does not require resource-intensive duplication of large hardware units. A spatial implementation requires more area than a time-shared one, but it is simpler to understand, needs fewer multiplexers and is less prone to routing congestion, making it easier to achieve timing closure in FPGAs.

For this reason, we chose a deeply pipelined implementation of the texture-mapping algorithm. Figure 5 depicts a block diagram of this scheme.

The first stages of the pipeline fetch low-bandwidth vertex information from the memory, and then compute the interpolated texture and destination coordinates using a variant of the Bresenham algorithm. We implemented those stages using behavioral Verilog HDL, which the free XST synthesizer (part of the ISE® WebPACK™ design suite) processes to produce an optimized netlist. The address generator can take advantage of the hardware multipliers present in the DSP48A1 slices of the Spartan-6 FPGA to efficiently compute the memory addresses within the texture frame buffer that correspond to the interpolated coordinates. The XST synthesizer automatically infers hardware multipliers from the use of the "*" operator in the HDL source code, which makes them very easy and convenient to use.

Things get more complicated when it comes to fetching the texel data from the memory. At each clock cycle, we need to fetch four different pixels from the cache. It would not make sense to have four separate caches, since different channels of the bilinear filter often use data from the same cache line. We therefore need a

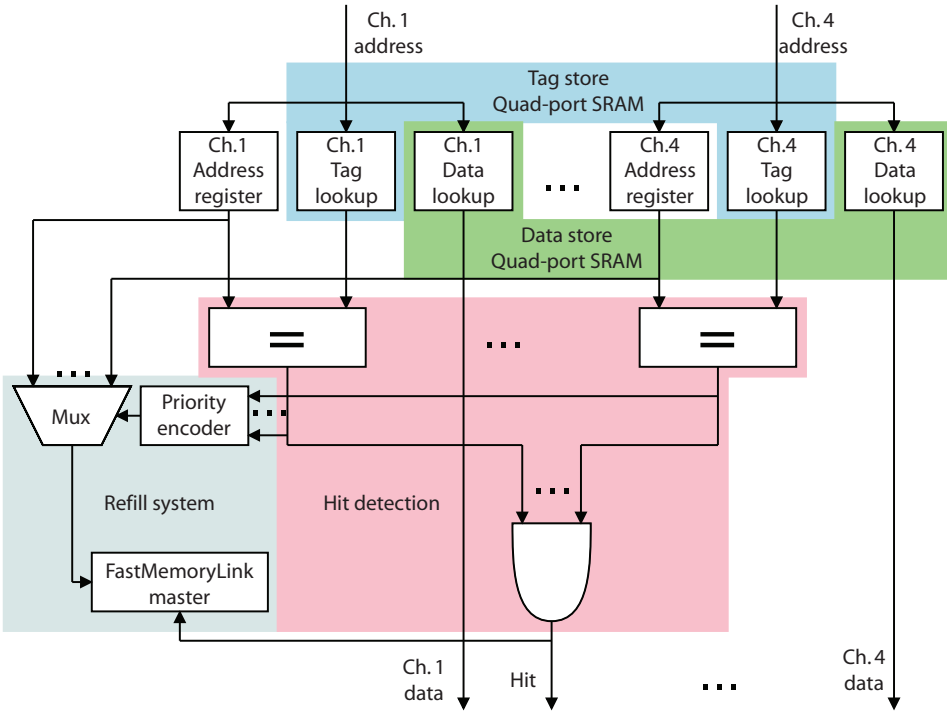


Figure 6 – The four-channel texel cache operates in a pipelined manner.

quadruple-port SRAM, which may seem difficult in an FPGA.

Fortunately, the true dual-port SRAMs of the Spartan-6 FPGAs offer an elegant solution. We can implement quadruple-port SRAM at a moderate cost by using two primitive dual-port SRAMs in which we replicate the data. During normal operation (hits), each port serves one channel. When refilling the cache on a miss, reading is disabled and two of the ports (one per primitive dual-port SRAM) are used to feed the data into the memories.

Figure 6 shows a simplified block diagram of the texel cache. At each clock cycle, the texel cache processes, in a pipelined manner, our memory addresses from each channel if they hit the caches. The "hit" signal is kept high and the pipeline is always running.

In case of a miss, the "hit" signal goes low (stalling the pipeline), and the priority encoder and the multiplexer (mux) select one of the missed addresses (there can be one or many). The memory bus master issues a memory transaction to retrieve the data from the system memory, replaces the contents of the cache line and rewrites the tag. The address now becomes a cache hit. If no other address misses the cache, the texel cache has successfully handled the four-channel transaction and the "hit" signal goes high again to proceed to the next cycle. Otherwise, the process repeats until all addresses hit the cache.

As you can see, it is possible to implement a decent four-port caching system in a modern FPGA by only doubling the number of block RAMs used for storage and with a very reasonable amount of control logic.

Following the texel cache, the bilinear filter blends together the contributions of the four fetched texels. Here again, our design takes full advantage of the DSP48A1 slices of the Spartan-6 to quickly compute the weighted sums. Finally, the result is stored into the SDRAM-based system memory using a write buffer.

Once integrated in our soft-core system-on-chip, our texture-mapping unit uses only a fraction of the resources of the low-cost Spartan-6 FPGA but provides a peak fill rate of 70 million pixels per second and an average fill rate of 37 million pixels/s. This is much faster than what software alone would provide, even when running on a high-performance (and power-consuming) ASIC CPU, and meets the demands of our application.

HIGHLY FLEXIBLE SINGLE CHIP

High-performance reconfigurable FPGAs make it possible to combine heavy graphics processing—often thought to necessitate ASICs—with very specific I/O interfaces in a highly flexible single chip.

The Milkymist system takes advantage of many features of Spartan-6 FPGAs: I/O delay elements, DDR registers, large true dual-port block RAMs, DSP slices, flexible DCM_CLKGEN elements, the ability to configure from NOR flash and MultiBoot. Our complete design uses only about half of the FPGA resources, which leaves plenty of space for future improvements and features. This is remarkable in a chip as low in cost as the XC6SLX45.

Regarding those future improvements, our whole FPGA design is open source and is available under the same licensing and development model as the Linux kernel. You are welcome to modify it and reprogram the FPGA as you wish, and use our system as a universal development platform, demonstrating the power and flexibility of the Spartan-6 family. Designers can rebuild the complete bitstream using the free ISE WebPACK design software, available for both Linux and Windows.

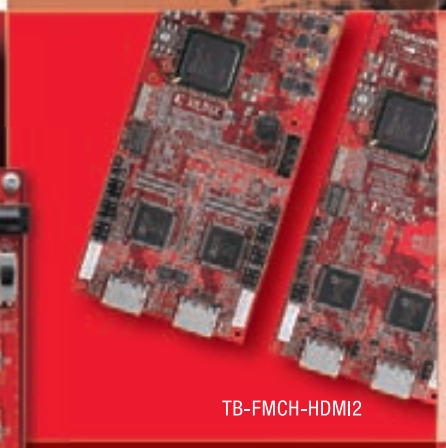
Finally, it is worth mentioning that our device uses less than 5 watts altogether, demonstrating another advantage of the single-chip FPGA-centric solution and debunking the myth that all FPGA systems must consume lots of power.

For more information, visit *http://www.milkymist.org*.

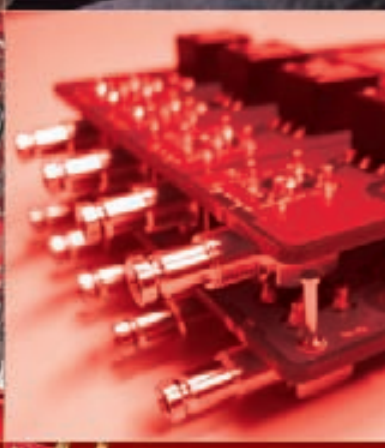
Our Platform. Your Revolution.
Broadcast Connectivity Kit.

inrevium

A Revolutionary Innovation Platform



TB-FMCH-HDMI2



TB-FMCL-AUDIO / TB-FMCH-3GSDI

TOKYO ELECTRON DEVICE LIMITED

Headquarter : Yokohama , Japan
E-mail:psd-sales@teldevice.co.jp
http://www.inrevium.jp/eng/x-fpga-board/index.html

US Offices : Fremont , CA
E-mail:psd-nasales@teldevice.co.jp

China Offices : Shanghai
E-mail:sales@teldevice.cn

XILINX® ALLIANCE PROGRAM PREMIER MEMBER

DIGILENT® BEYOND THEORY

Buy Now

http://www.digilentinc.com/INREVIUM

Spartan-6 FPGA Broadcast Connectivity Kit
Part Number : TB-6S-BCK-PRO/TB-6S-BCK-FND

www.aldec.com

Your Global Xilinx® Verification Solutions Provider

- VHDL, Verilog, SystemVerilog
- OVM/UVM, VMM
- Code and Functional Coverage
- DSP Co-Simulation (MATLAB® & Simulink®)
- Emulation (Hardware-Assisted Verification)

Headquarters-US
2260 Corporate Circle
Henderson, NV 89074
USA

Phone: +702.990.4400
E-mail: sales@aldec.com

Europe
Mercia House
51 The Green, South Bar
Banbury, OX16 9AB
United Kingdom

Phone: +44.1295. 20.1240
Email: sales-eu@aldec.com

Israel
Even Yehuda 40500
6 Macabi St.
POB 2521
Israel

Phone: +972.5.2257.3422
E-mail: sales-il@aldec.com

Japan
Shinjyuku Estate Bldg. 9F
1-34-15, Shinjyuku
Shinjyuku-ku, Tokyo 160-0022
Japan

Phone: +81.3.5312.1791
Email: sales-jp@aldec.com

China
Suite 2004, BaoAn Building
#800 DongFang Road
PuDong District
Shanghai City, 200122, P.R. China

Phone: +86.21.6875.2030
Email: info@aldec.com.cn

India
#4123, Ist Floor
6th Cross, 19th Main
HAL II Stage Indira nagar
Bangalore, 560008, India

Phone: +91.80.3255.1030
Email: sales-in@aldec.com

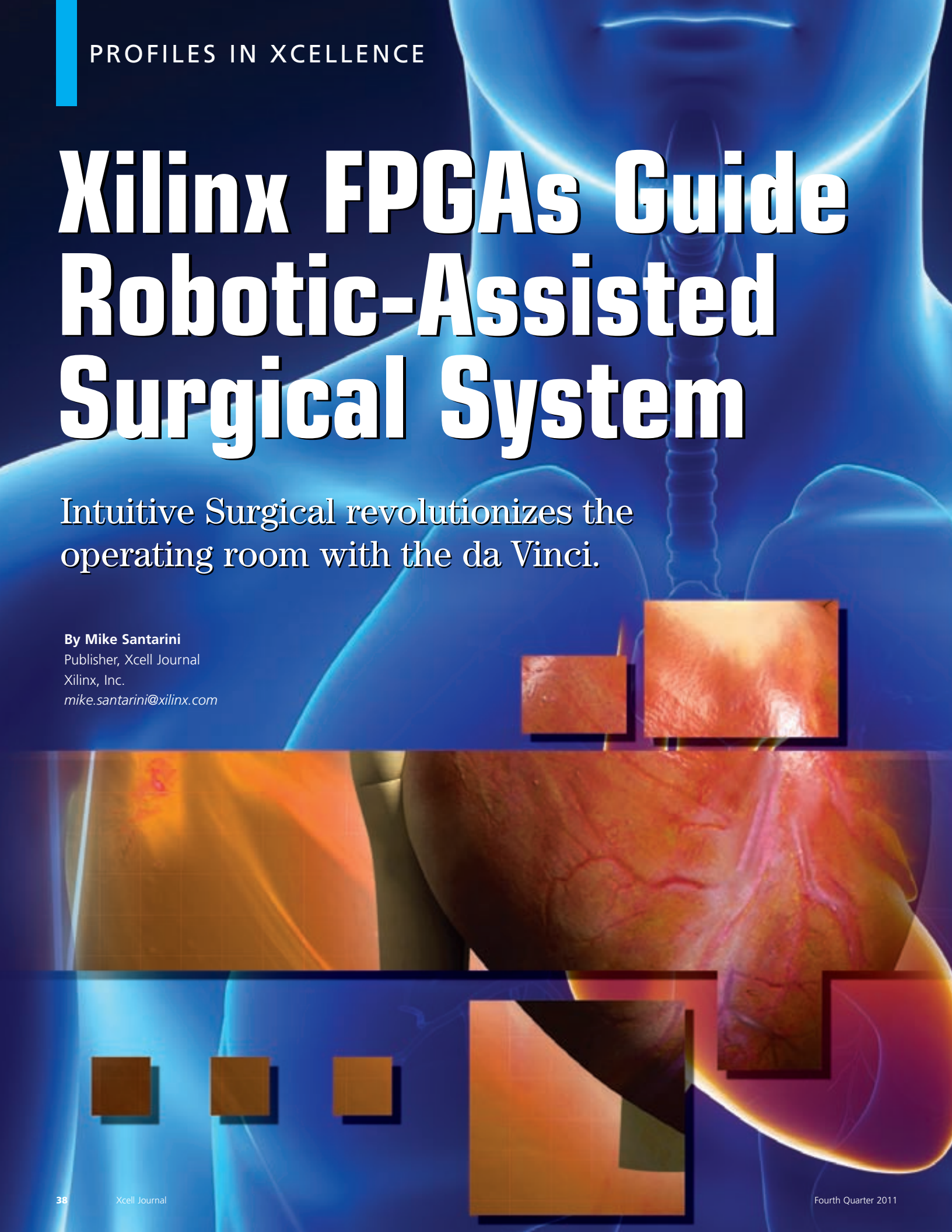
Taiwan
Room 920, 8f, no.8
Lane 360, sec.1
Neihu Rd., Taipei
Taiwan

Phone: +886.2.2659.9119
E-mail: sales-tw@aldec.com

Xilinx FPGAs Guide Robotic-Assisted Surgical System

Intuitive Surgical revolutionizes the operating room with the da Vinci.

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



Saving lives, reducing recovery times—it's all about the patient. This is the proud mantra of one of the most interesting companies in the medical-device industry: Intuitive Surgical. Already in use at more than 1,500 sites worldwide and proven in hundreds of thousands of patient surgeries, the company's robotic-assisted da Vinci Surgical Systems are the state of the art in minimally invasive surgery (MIS), allowing patients to recover much sooner than they would from traditional open surgical procedures.

Intuitive Surgical has used Xilinx® FPGAs since 2003, leveraging new device families over successive years to create ever more sophisticated versions of the da Vinci and to expand its use. Surgeons now employ the tool in a range of urologic, gynecologic, cardiac, thoracic, head and neck, as well as general surgical procedures.

The da Vinci Systems consist of three main interconnected components: a surgeon console, patient-side cart and vision system (Figure 1). The surgeon console is essentially a cockpit for surgeons doing a given procedure. Rather than standing over a patient and bending for hours to perform an operation—as is the case with traditional surgeries—physicians using the da Vinci System sit comfortably at the console with their fingers on the master controllers (Figure 2) and their eyes positioned in a 3D viewer. Here, they can see the target anatomy magnified to their liking and view the surgical instruments they'll use for the procedure.

The da Vinci System translates the movements of the surgeon's fingers and wrists in real time to guide robotic arms on the patient-side cart accordingly. Surgeons manipulate the controllers to guide various surgical instruments—scalpels, clamps, scopes, cauterizing and suturing needles, and so on—which are positioned at the end of individual robotic arms on the patient-side cart. From the comfortable cockpit, surgeons perform every critical step of the surgery, from initial cut to the final suturing, while OR assistants monitor the patient.

The da Vinci System translates the movements of the surgeon's fingers and wrists in real time to guide robotic arms on the patient-side cart accordingly. Surgeons manipulate the controllers to guide various surgical instruments.

Figure 1 – The da Vinci System improves surgical outcomes and speeds patient recovery. In this picture, physicians sit at dual surgeon consoles, at the left and left middle. The vision system is pictured in the right middle, while the patient-side cart is at the right.

For its part, the third instrument connected to the da Vinci System, the vision system, is equipped with a high-definition, 3D endoscope (a tube with a camera and light at the tip) and image-processing equipment that provides true-to-life images of the patient's anatomy. The vision system also gives the entire OR team and surgical assistants at the patient's side a wide-screen view of the operating field.

For medical facilities, da Vinci provides great advantages over tradition-

al surgical procedures in that it allows surgeons to perform complex operations using a minimally invasive approach, more comfortably and with less fatigue. What's more, the system minimizes hand tremors, enhancing the precision of surgeons' movements and thus, potentially, extending their careers.

Chris Simmonds, senior director for marketing services at Intuitive Surgical, said doctors benefit from the enhanced visualization and system ergonomics, reporting less eye strain and improved dexterity and control, especially for procedures that require very high magnification. "One doctor shared that he can now do seven or eight reverse-vasectomy procedures per day, instead of the two-per-day rate prior to da Vinci," said Simmonds.

But the biggest advantages of the system, said Sal Brogna, senior vice president of engineering at Intuitive Surgical, are for patients. "Patients are first and foremost in our minds at Intuitive Surgical, and helping patients have better surgical outcomes and faster recovery is what drives our company and the engineering decisions behind our technology," he said. "The da Vinci allows surgeons to be more precise. More-precise procedures mean shorter recovery times—and less time in the hospital makes all patients and their families happier."

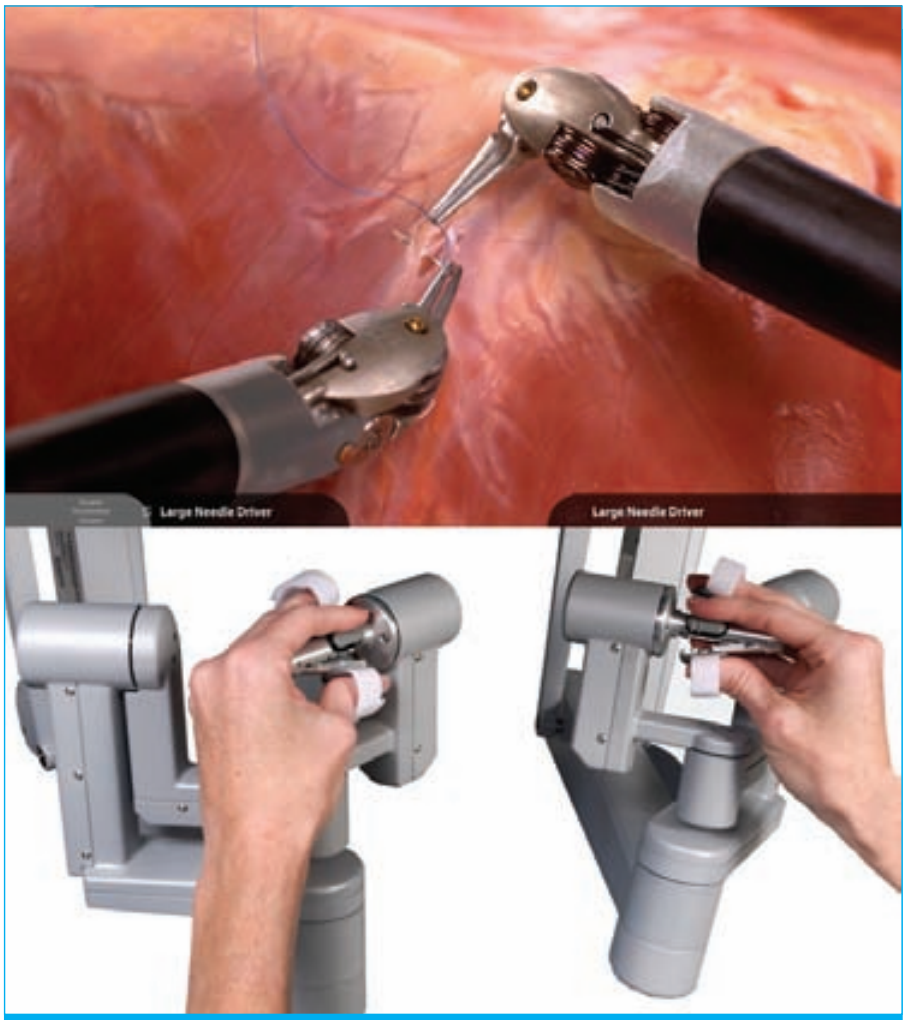


Figure 2 – The surgeons position their hands in the specialized instruments (bottom picture). The da Vinci System computes their subsequent movements in real time and guides robotic arms (seen in closeup at top) on the patient-side cart, which is positioned over the area of the patient where the surgery will be performed.

Originally inspired by technology advances stemming from research at the Defense Advanced Research Projects Agency (DARPA), Intuitive Surgical's robotic MIS system presented numerous engineering challenges for its developers. One area, video processing, has become increasingly important as the da Vinci models have evolved from 3D standard-definition stereo vision to today's dual-console, multiwindow 3D high-definition (HD) system.

"When we were updating the original video-processing subsystem, we wanted to introduce multiwindowed video sources for the surgeon, so they could monitor vital patient data during surgeries," said David Powell, principal design engineer for Intuitive Surgical's video-processing solutions. "An increase in video-processing bandwidth would allow us to display data from auxilliary video sources, along with the view of the operating field. For example, surgeons would have instant feedback from an ultrasound or heart-lung machine without taking their eyes off of the procedure in progress."

Brogna said that in addition to the technical challenges involved in giving surgeons an expanded, immersive view that could shorten procedures and improve outcomes, the video solution also had to comply with stringent safety and reliability requirements. This meant that the system needed to be flexible, upgradable and, of course, reliable.

All these requirements led Intuitive Surgical to ultimately employ a Xilinx Virtex® FPGA in the video-processing design for its second-generation da Vinci Surgery Systems, which it designed in 2003. "We initially selected a Xilinx Virtex-2 Pro FPGA solely on the performance of the DSP elements for streaming video," said Powell. "The embedded processor Xilinx had available for that device was a 'nice-to-

have' feature. We realized we could take advantage of it to reduce the real estate for video processing, but the embedded processor wasn't a fundamental reason for our selection of the Xilinx device."

As it turned out, however, "Xilinx's embedded-processor architecture has led to a pretty major revolution for us in terms of our subsequent platform designs throughout the entire system," Powell said.

By reusing cores in successive systems and adding functionality to new generations of FPGAs, Intuitive has moved to a more distributed architecture. Customers can add modules to suit their particular needs.

FPGAS HELP USHER IN MODULAR DESIGNS

Intuitive Surgical's first experience with Xilinx devices led the company to follow the FPGA technology curve and add more-advanced system functionality to the latest Xilinx FPGAs as they became available. "As we started using the Xilinx device, we discovered it to be quite a nice design platform—so nice, in fact, that follow-on platforms have evolved to employ dozens of Xilinx FPGAs in all of the main system components," said Powell. "Today, we can fit so much in each FPGA [that] we can turn a board into a chip, pretty much."

Although Intuitive's engineers did not leverage embedded processors in the FPGA designs they employed in the early generations of the da Vinci System, they did in the last two generations. For example, Intuitive used the Virtex-5 FX FPGA's PowerPC® hard processor and MicroBlaze™ soft processor in many modules in the last two generations of da Vinci Systems. Powell said that FPGA block/design reuse has also been a key in helping Intuitive Surgical bring new generations of its system to market sooner.

"Using a cookie-cutter approach, we have been able to standardize many functions and build these blocks into new designs very quickly," said Powell. "Our first board to employ a Xilinx FPGA was up and running in two hours. After that, we found we could get a board up and running in just minutes—these kinds of results are almost unheard of."

By reusing cores in successive generations of the system and adding system functionality to new generations of FPGAs, the company has been able to move to a more distributed architecture for da Vinci and usher in the module era, in which customers can add several modules to a single system to suit their particular needs. For example, Brogna said that the distributed architecture helped the company introduce systems with dual consoles. "Now two surgeons can collaborate in a robotic MIS procedure, or set up a trainer-student configuration," said Brogna. "The modular designs, which leverage Xilinx FPGAs, contribute to this capability and mark a major milestone in our product line."

Brogna said that prior to using Xilinx FPGAs, da Vinci component interconnections were varied and complex. The modules were linked via four large "garden hose" size cables, which wore out too quickly due to constant manipulation during surgical setup in the operating rooms. More significantly, system components had to be manufactured—and repaired—as an integrated unit. Thus, if one component needed repair, the entire system was out of action. Today, a single-fiber cable design provides standardized connections among all system components. The Xilinx hard-processor blocks and high-speed DSP slices provide system-on-a-chip capabilities that support eight channels of full 1080i HD video (20 Gbps) across the simplified interconnect. This new interconnect technology has cut failure rates dramatically.

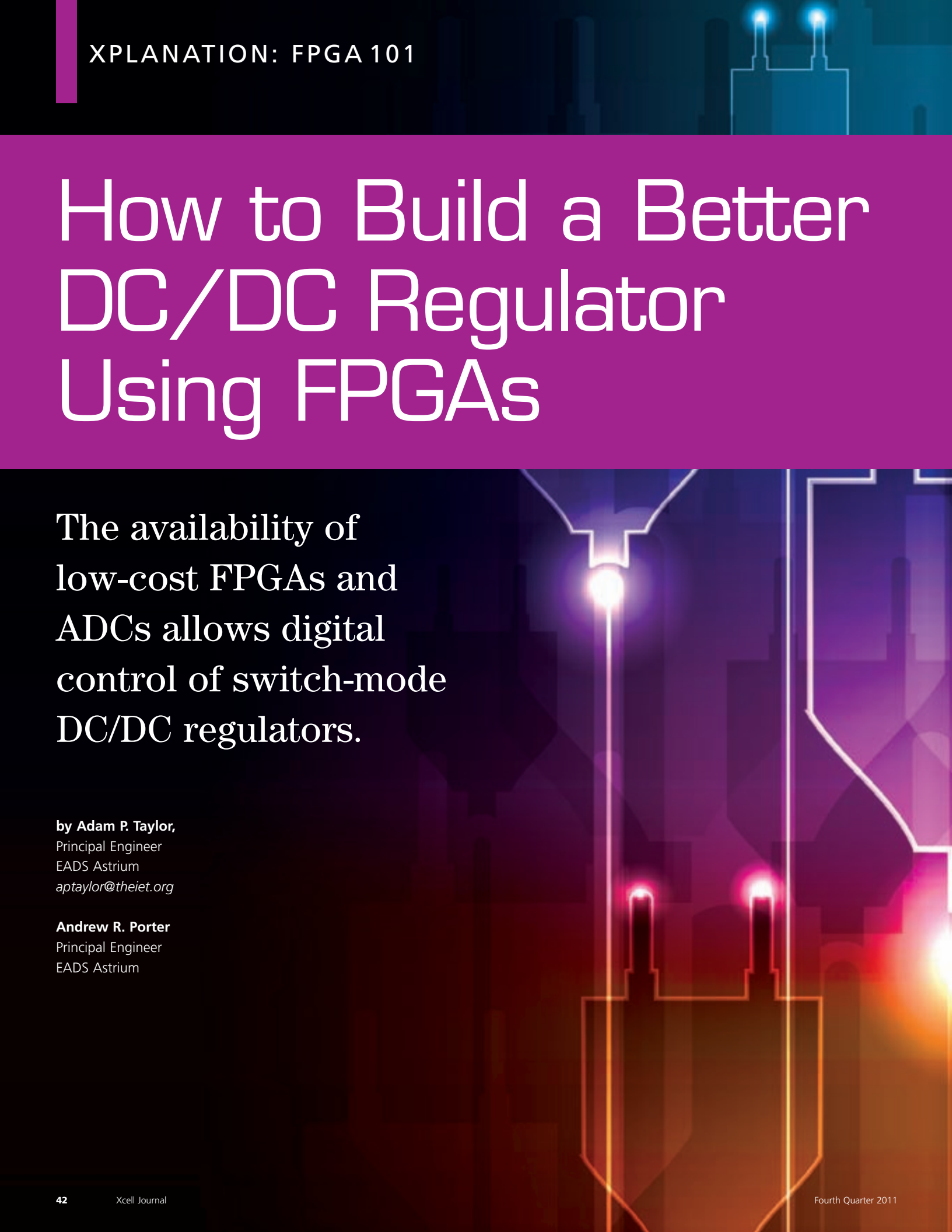
Brogna said that Intuitive Surgical's modular design has also revolutionized manufacturability, testability, reliability and serviceability. "A flexible, customizable design block made us think about everything in new ways—we now focus on modules and cards," said Brogna. "Even manufacturing doesn't talk about shipping systems—they talk about cards. This has made us incredibly agile and effective in terms of producing and testing products and servicing systems in the field."

The power of programmability also means simplified updates. Brogna said that instead of replacing modules or subsystems, Intuitive Surgical can introduce new functionality or enhance existing functionality with an in-field firmware upgrade. Service teams can also quickly query for consistency across all the processors in the system, for improved process control and to ensure that systems are optimally configured for surgery.

Powell pointed to a close partnership with Xilinx's technical staff, sales force and executives as another key to success. "We know Xilinx devices backwards and forwards now, and this really helps us make a difference in many lives," he said. "It always comes back to the patients. We hear from people every day who tell us how a new procedure changed or saved their life. That's what motivates us to deliver the best technology."

For more information on the da Vinci System, visit *http://www.intuitivesurgical.com/*.

How to Build a Better DC/DC Regulator Using FPGAs



The availability of low-cost FPGAs and ADCs allows digital control of switch-mode DC/DC regulators.

by Adam P. Taylor,
Principal Engineer
EADS Astrium
aptaylor@theiet.org

Andrew R. Porter
Principal Engineer
EADS Astrium

Designers traditionally build switch-mode DC/DC converters using analog components (bespoke ICs, operational amplifiers, resistors, capacitors and the like) to control the feedback loop and to generate the pulse-width modulation required for switching. When using analog components like these, you must consider a number of factors, taking tolerances, electrical stresses, aging drift and temperature drift into account to ensure the stability of the design. Now, the availability of affordable low-powered FPGAs coupled with analog-to-digital converters allows the FPGA to replace the traditional analog approach.

DC/DC converters are designed in one of four main topologies: buck (step-down), boost (step-up), inverting (converting a positive input to a negative output) and SEPIC (single-ended primary inductor converter). SEPIC devices maintain a constant output voltage, stepping the input voltage up or down depending upon the circumstances. For this reason, they are a popular choice for battery-based applications.

CONTROL BASICS

The basics of controlling a DC/DC converter are the same regardless of whether you implement the control loop using an FPGA or analog techniques. The switch-mode regulator stores energy within an inductor and transfers this energy to the output under the control of the regulation feedback loop, with the chosen topology determining the effect upon the output voltage. This transfer of charge from the inductor to the output is achieved via a switching FET that's driven by a pulse-width-modulated signal from the controller. The controller uses the difference between the current output voltage and the desired reference voltage to adjust the duty cycle of the PWM to ensure it maintains the desired output voltage during large current transients or during startup of the system.

Converter Equation	Output Voltage Style
Buck	Vout = Vin * D
Boost	Vout = Vin/(1-D)
Inverter	Vout = D(Vout-Vin)
SEPIC	Vout = Vin * (D/1-D)

Table 1 – Output voltage relationship to input voltage and duty cycle (D)

An FPGA-based system requires the same modules as an analog one—namely, PWM generation, error calculation and a control algorithm to adjust the PWM. At the same time, the FPGA delivers a number of distinct advantages, most of them easily achievable within the FPGA itself without any special requirements except for a bit of HDL coding. However, this technique needs an analog-to-digital conversion scheme (either ADC based or some other approach) to feed back the current output voltage into the FPGA, facilitating the control algorithm's ability to determine the adjustment required to the PWM output.

BENEFITS OF FPGA CONTROL

The benefits of digital control over the traditional analog approach are many, and far outweigh the accompanying issues with regard to component parametric shifts (Figure 1 depicts an FPGA-controlled buck converter). The

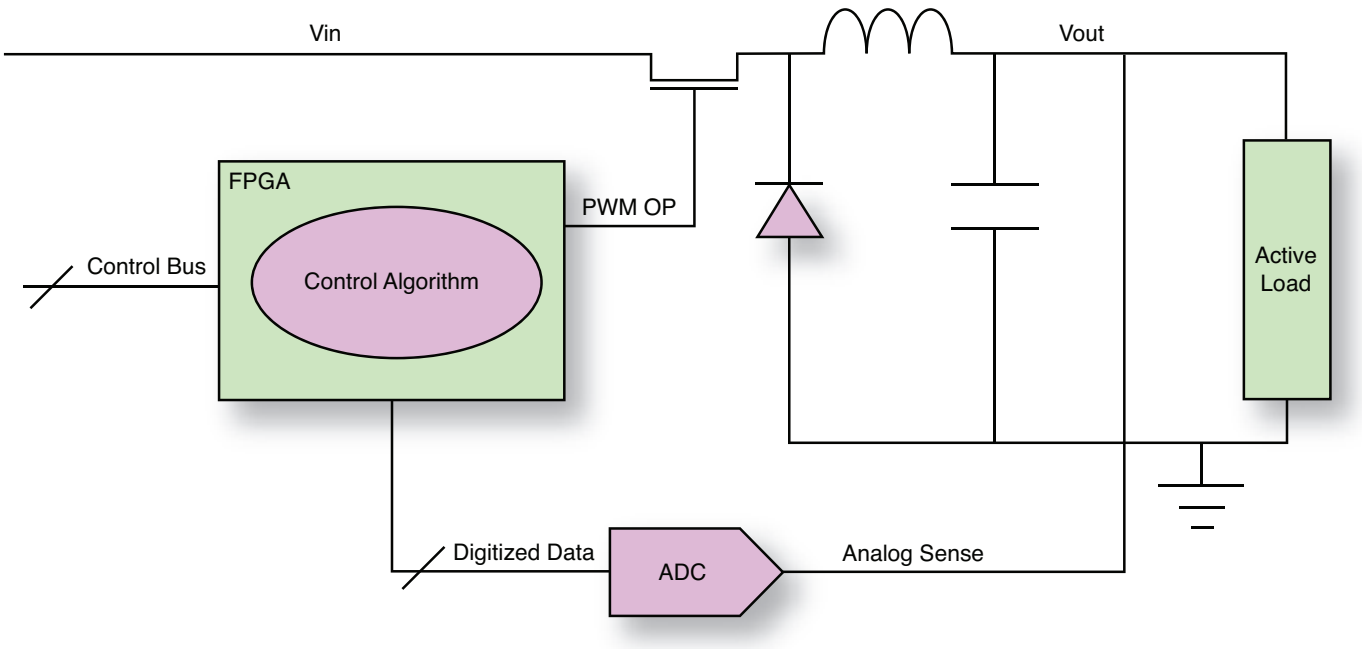


Figure 1 – The diagram shows how to use an FPGA as the basis for a buck converter.

FPGA is capable of running at much faster frequencies than the traditional approach, allowing for faster responses to transient current loads (potentially reducing the need for multiphase converters in some applications). The control algorithm can be adapted using DSP and control-theory techniques to ensure high efficiency in the implemented design—for example, pulse skipping under lighter loads or changing between continuous and discontinuous modes. DSP techniques such as filtering can replace the need for external components like filter capacitors on the feedback voltage or sense currents. Digital filters are capable of providing much tighter filtering than the simple RC filters that many analog applications use.

The relatively small size of the logic footprint required to implement a DC/DC converter when using an FPGA (less than 100 slices for a very simple converter) means that a single device can generate several output rails. That reduces the cost of the implementation even further.

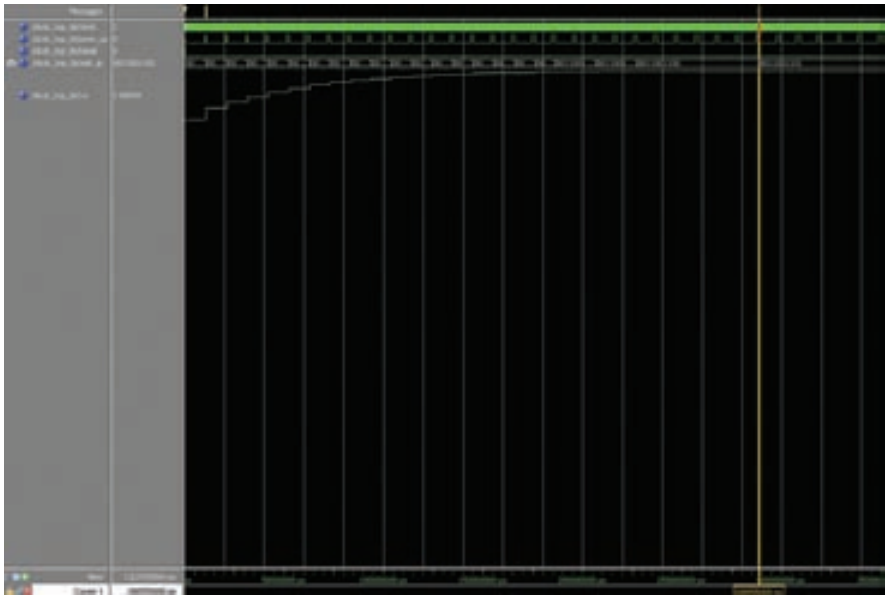
Where several modules are used together, it is possible to implement a simple serial bus (SPI or 12C), or even a more-complex interface such as Ethernet or CAN, to configure the desired output voltages, ramp rates, switching frequencies or telemetry regarding the current output voltage or load current. It's also easy to add temperature telemetry at little cost in terms of FPGA resources or board space.

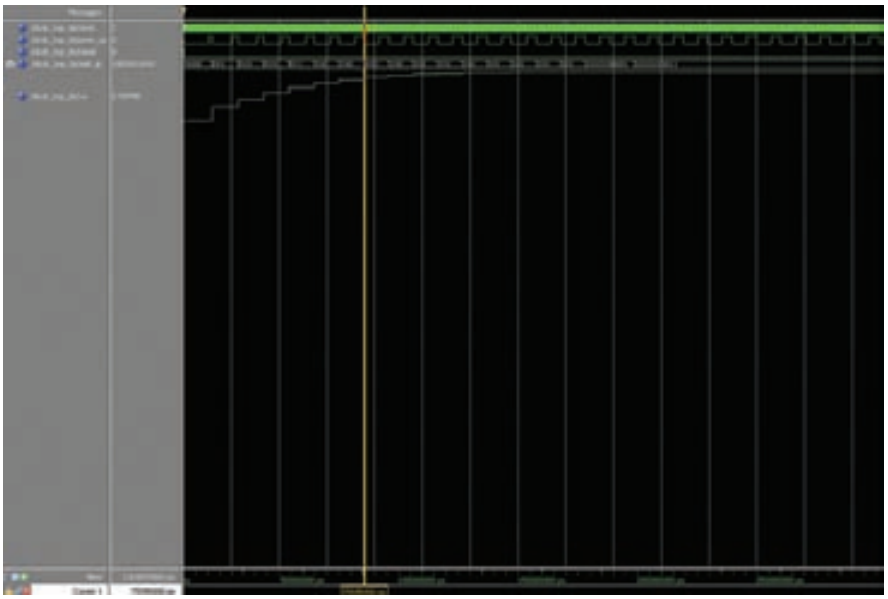
The benefits of an FPGA-based approach are not just limited to technical or performance aspects. With fewer components and not as many types of them, manufacturing is easier. It's especially helpful to cut back on resistors and capacitors with different values, as required often for loop stability. JTAG testing will allow you to determine the basic connectivity of the design with only the true analog components (inductors, FET capacitor), whereas flying-probe testing will determine the correct values have been fitted. The increased use of JTAG testing will also speed up the time required to test while reducing the time and cost of developing the flying-probe test program. One additional benefit, depending upon what component the FPGA is replacing, is the increase in reliability gained from using an integrated IC.

WHAT ABOUT CONFIGURATION, POWERING, CLOCKING?

The FPGA will of course require supporting circuitry to function correctly, as it is very unlikely that the FPGA will happily operate from the input voltage. Modern Xilinx® Spartan®-3AN devices are available in very small footprints and require little power for both the internal core and external I/O drivers. You can therefore use off-the-shelf low-dropout linear regulators to power the FPGA. Similarly, configuration used to mean adding a separate configuration device, normally flash memory, to the design. However, the Xilinx Spartan-3AN family is capable of configuring from internal flash memory, removing the need for a separate device.

Unfortunately FPGAs do not contain oscillators, and hence your converter will need one. However, FPGAs do contain digital clock managers. DCMs let you multiply up low clock frequencies, reducing the possibility of oscillator noise making it to the output voltage. Several companies produce low-footprint oscillators that would be suitable for use.





Figures 2-3 – Pictures from the simulation of our algorithm showing the voltage being regulated when driving for outputs of 1 and 3.3 V, respectively

REAL-WORLD IMPLEMENTATION

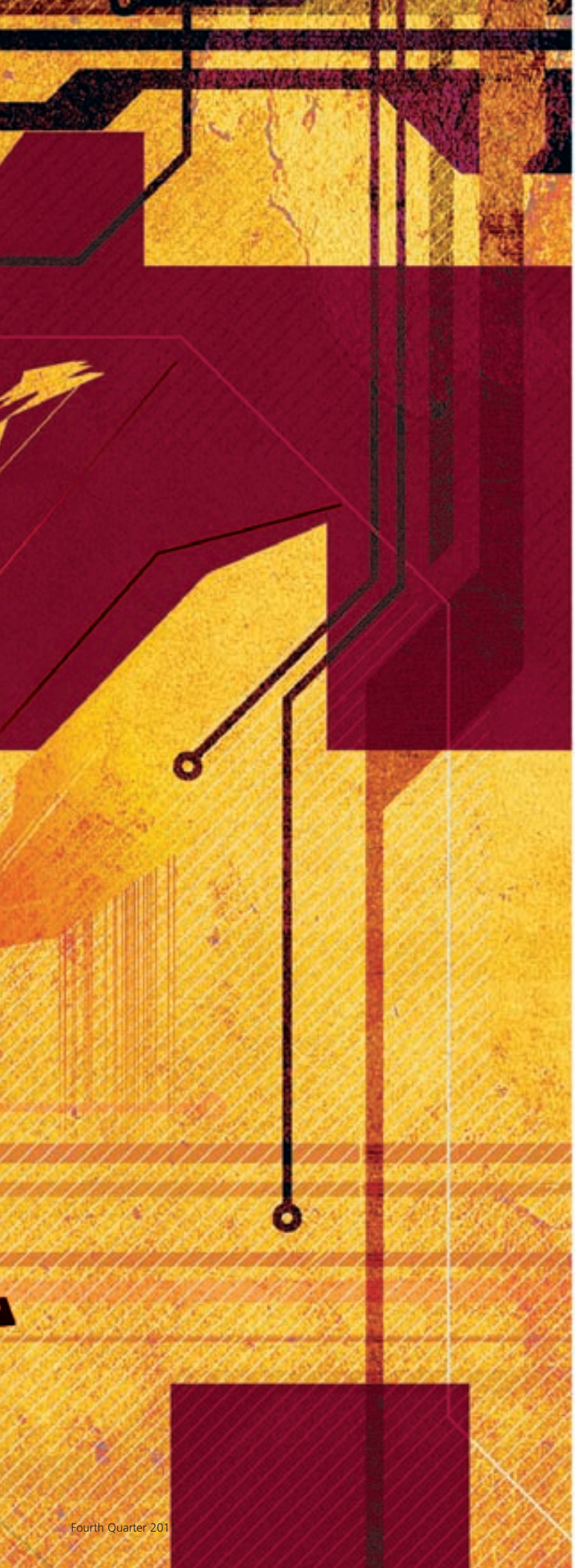
When our team at EADS Astrium implemented a digital control algorithm targeting a Xilinx XC3S50AN, the resources required were approximately 5 percent of available flip-flops and 12 percent of the available lookup tables. Figures 2 and 3 show mixed-signal simulation results of the performance of the algorithm when driving for outputs of 1 and 3.3 volts. Our results clearly demonstrate that you can accurately obtain the desired output voltages using an FPGA-based control system.

Designers are using FPGAs in an increasing number of applications that analog components or microprocessors have traditionally dominated. FPGAs have a lot to offer when used in point-of-load switch-mode power supplies, to both the manufacturer and the end user of the product

Using the Clock Period Constraint to Your Advantage

Understanding the TS_clk constraint in terms of Xilinx ISE and tool behavior will help you attain effective timing closure in FPGA designs.

by Sharad Sinha
PhD Candidate
Center for High-Performance Embedded Systems
Nanyang Technological University
sharad_sinha@pmail.ntu.edu.sg



Designers set timing constraints to meet the particular timing requirements of their chip designs. Then the physical-synthesis tools place and route the design so as to meet these timing constraints. One very common and important timing constraint is related to the maximum clock frequency and is commonly referred to as the period constraint. In the Xilinx® ISE® tool suite, this constraint goes by the name TS_clk in the user constraint file (UCF). The Xilinx Timing Constraints User Guide states that the period constraint is used to:

1) Define each clock in a design
2) Cover all synchronous paths within each clock domain
3) Cross-check paths between related clock domains
4) Define the duration of clocks
5) Define the duty cycle of clocks

The user guide offers a wealth of details and good, relevant explanations of the functions of the period constraint. But it's worth taking a closer look "under the hood" of the FPGA synthesis tools to explore a number of questions about the behavior of the period constraint, and to gain insight into the way placement-and-routing algorithms work. Specifically, let's consider ways to interpret a failing TS_clk constraint; examine whether you'll get progressively better results by constraining the tool progressively; and discuss why the tool shows a discrepancy in results. Finally, let's also ask whether such a timing constraint always helps vs. proceeding with an unconstrained design.

HOW TO INTERPRET A FAILING TS_CLK CONSTRAINT

If a design fails to meet the clock period constraint, it means that it cannot run at that clock frequency. You can attempt thereafter to pipeline the design so as to relax the timing budget in the slow paths. Pipelining may be enough to make the design meet the constraint. Another way to improve timing is to reduce the number of logic levels between two registers—essentially, you need to simplify the logic design in the critical path. These two techniques are applicable at the design level, where a designer can do the necessary modifications to the RTL code. If your design still has not met timing after the RTL modifications, the next step is to enable the Xilinx ISE switches -register_balancing (which is meant for register retiming) and -register_duplication (which duplicates registers to ease high fanout at a particular register).

Another way to improve timing is to assign pins to I/O signals properly. A good design practice is to assign adja-

cent pins to adjacent signals. For instance, you should assign all the signals on an I/O bus to adjacent pins in one bank. Use adjacent banks while assigning large numbers of pins.

These points are significant because they act as constraints for the place-and-route tool. The tool would generally try to keep related logic together. This effort improves when relevant I/Os are assigned adjacent pins, because the technique will likely decrease routing delays. The tool would then not scatter the logic on the device. Scattering logic increases routing delays.

Generally, when an FPGA has to sit on a printed-circuit board, you need to take additional board-related considerations into account while assigning pins. Since the FPGA would interface with other chips on a board, adjacent pin assignments may not always be possible. Therefore, it is always best for an FPGA designer to consult with the board designer early in the design cycle to reduce pin assignment conflicts.

Still another way to improve timing is to use a higher-speed-grade device. However, this affects the price of the product and hence is not an easy option. Not only is there the higher cost of the device itself to take into account, but also a higher-speed part has an impact on board design and could easily increase the board design cost as well.

DOES THE TOOL ALWAYS GIVE YOU A BETTER RESULT AS YOU CONSTRAIN IT PROGRESSIVELY?

Sometimes we would like to know the maximum frequency at which a particular design can run. To investigate this, we constrain the design progressively. For instance, we start with a clock period constraint of 8 nanoseconds (corresponding to a clock frequency of 125 MHz). If the tool succeeds in placing and routing the design under the constraint, it might report a minimum clock period of something like 7.68 ns. We can then constrain the clock period to 7.68 ns and rerun ISE. This time the tool might report a minimum clock period of 7.56 ns. However, when we constrain the design again to 7.56 ns, the tool might report timing failure with the minimum possible clock period being, say, 7.74 ns. This means that we need to constrain the design to 7.68 ns in order to achieve the figure of 7.56 ns. So, there is a limit to progressive constraining of a design in order to improve timing. After a certain level of constraint, the results might deteriorate.

If the design is small and the period constraint is very tight (a very small period), the tool may report a clock value less than the period constraint in the post-place-and-route (PPR) static timing report. But it would still show a timing-error score (which is zero when there is no timing error) like the one on this page, from an actual static timing report of a design (targeted at XC4VFX140-11FF1517) in which the period constraint was set to 1.5 ns with 50 percent duty cycle:

```
1 constraint not met.
Data Sheet report:
-----------------
All values displayed in nanoseconds (ns)
Clock to Setup on destination clock clk
-------------+-----------+----------+-----------+--------------+
             | Src: Rise | Src: Fall | Src: Rise |  Src: Fall   |
Source Clock | Dest: Rise | Dest: Rise | Dest: Fall | Dest: Fall|
-------------+-----------+----------+-----------+---------------+
clk          | 1.489                |           |              |
-------------+-----------+----------+-----------+---------------+
Timing summary:
---------------
Timing errors: 1  Score: 722  (Setup/Max: 0, Hold: 0, Component
Switching Limit: 722)
Constraints cover 40 paths, 0 nets and 62 connections

Design statistics:
---------------
Minimum period:   2.222 ns{1}   (Maximum frequency: 450.045 MHz)
```

The report clearly shows a clock period of 1.489 ns, which is less than 1.5 ns. However, this design was targeted at a speed-grade -11 Virtex®-4 device with a maximum frequency of 450.05 MHz. Hence, there is a timing error. The point is that it is important to read about the device's switching characteristics also while setting the constraint.

WHY DOES THE TOOL SHOW A DISCREPANCY IN RESULTS?

The tool begins to show a discrepancy in results because it works based on heuristic algorithms. Designers use these algorithms to solve problems for which exact algorithms are either unsuitable, for reasons of time and space complexities, or extremely difficult to develop. To choose a solution, heuristic algorithms generally use a so-called cost function, which takes into account some information about the device or some other empirically derived constants. These algorithms do not, however, guarantee that the solution will be the best or the optimal one.

Often heuristic algorithms start with an initial random seed value for placement of logic; then the placement process grows around the seed location based on the cost function analysis, and routing follows. Since the seed value may change with each invocation of the tool following each change in constraint, the results can get worse beyond a certain point. The tool has no reference of what it did and what results it reported in the last run so as to improve its working further. It is extremely difficult to design a place-and-route algorithm that takes a prior

placement-and-routing strategy into account, and compares the current and the previous results.

The SmartGuide™ technology in Xilinx ISE can guide a new implementation based on results from a previous one. But SmartGuide works only if there is some change in logic between two iterations of an implementation. It is not applicable when there is no change in logic and the same design is simply constrained progressively. Quite often, designers get confused with this distinction.

For its part, the SmartXplorer option in Xilinx ISE is simply one way of speeding up the process of investigating timing with different timing constraints. This strategy allows a designer to investigate the same design with different constraints as the tool executes them in parallel on different machines: on a Linux network or a Linux/Windows machine with multiple processors.

Thus, even with these options available in Xilinx ISE (and similar options in other FPGA design suites), the tool does not remember what it did in the previous run in order to compare and improve its results in the next run when timing constraints are progressively tighter. If that were the case, then beyond a certain point, the tool should simply report one minimum value instead of different values on constraining progressively. Since it's so difficult to design an algorithm that uses prior placement-and-routing information as a feedback to improve the timing of the same design, it pays to know and understand the limitations of the tools.

DOES A TIMING CONSTRAINT ALWAYS HELP OVER AN UNCONSTRAINED DESIGN?

The traditional thinking is that a constrained design will always have better timing than an unconstrained design. This is true in general. However, it is not always the case. Sometimes the tool places and routes the design in the best possible way when the design is unconstrained. The maximum achievable clock frequency is highest in the unconstrained implementation, as highlighted in Table 1. The reason for this discrepancy is again the way place-and-route algorithms work.

Our team at the Center for High-Performance Embedded Systems at Nanyang Technological University implemented the sum-of-absolute-differences (SAD) algorithm on a Virtex-4 XC4VFX140-11FF1517 FPGA using Xilinx ISE version 12.2 M.63C (see *Xcell Journal* Issue 75, page 38). We employed an 8 x 8 SAD that used eight image pixels (16 bits each) and eight reference pixels (16 bits each), using external select signals to choose which two pixels to subtract so that the design would have only one subtractor. We did not use any conditioning registers, and all internal registers were initialized to zero. We did no pin assignment for this experiment.

As you can see from Table 1, we achieved the best minimum clock period of 2.607 ns when no constraint was set. When we set 2.607 ns as the period constraint, the tool reported 2.863 ns as the best achievable clock period. Setting 2.863 ns as the period constraint resulted in a best achievable clock period of 2.795 ns. This is because the tool stops trying to meet the constraint once it achieves a value close to it. Setting 2.795 ns as the new constraint did not bring down the best achievable clock period to 2.607 ns but raised it to 2.966 ns. In this case, the tool failed to meet the constraint.

Period constraint (ns)	PPR reported value (ns)
No constraint	2.607
2.607	2.863
2.863	2.795
2.795	2.966
2.966	2.762

Table 1 – Effect of period constraint on actual timing

This randomness in results stems from the heuristic nature of place-and-route algorithms. This is also the reason why designers need to spend significant time in setting and resetting period constraints to meet timing closure.

PSEUDO-RANDOM SOLUTION

The period constraint is one of the most important constraints in FPGA design and is critical to timing closure. It is therefore important to understand how it behaves and how to interpret its results. Progressively constraining the clock period does not always improve the result. Vendors generally enhance the place-and-route algorithm implemented in their FPGA design tools upon every major release of the software. Hence, the timing results may vary from one version to another.

Unlike ASICs, where routing and placement are highly deterministic, FPGA placement-and-routing algorithms are heuristic in nature. This is easy to understand because of the very nature of FPGAs, where random logic has to be mapped onto a fixed hardware architecture with fixed components and routing resources.

FPGA placement-and-routing is an NP-complete problem —one for which there is no efficient way to locate a solution. For such problems, no known polynomial-time algorithms exist that can give an accurate or optimal solution. Hence, solving them involves using heuristics or some approximation or similar methods based on pseudo-random processes.

Also, the runtimes of these algorithms can vary rapidly with any increase in the size of the input, as many of us have experienced with large FPGA designs. This is a fundamental aspect of NP-complete problems. For this reason, the quality of results depends a lot on the type of heuristics used or the approximation method employed.

Optimizing FPGAs for Power: A Full-Frontal Attack

Here is a rundown of low-power design techniques for the latest families of 7 series FPGAs at each stage of the development cycle.

by Chandra Sekar Balakrishnan
Solutions Development Engineer
Xilinx, Inc.
cbalakr@xilinx.com

Power has become a primary factor in the ever-important search for the "perfect" FPGA for a given design. Power management is critical in most applications. Some standards specify maximum power per card or per system. As such, designers must consider power much earlier in the design flow than ever before—often starting with the selection of an FPGA.

Reducing the power consumption of the FPGA simplifies the board design by lowering the supply rails, simplifying power supply design and thermal management, and easing the requirements on the power distribution planes. Low power also contributes to longer battery life and higher reliability (cooler-running systems last longer) of the system.

POWER CHALLENGES

With each generation of process technology, transistors are becoming smaller and smaller in accordance with Moore's Law. This phenomenon has the unfortunate side effect of incurring more leakage within each transistor, which leads to higher static power consumption—that is, the amount of current an FPGA draws when not operating. Increased FPGA performance drives the clock rate higher, which leads to higher levels of dynamic power. Where static power is driven by transistor leakage current, dynamic power is based on the switching frequency in the programmable logic and I/Os. Exacerbating both types of power consumption, FPGAs are growing in capacity with each product generation. More logic means more leakage and more transistors operating at higher speeds per device.

Because of these issues, designers must be more aware of their power supply and thermal-management issues earlier in their design cycles. Slapping a heat sink over a device may not adequately resolve these issues. Instead, designers must look for opportunities to reduce the logic in the design.

Let's take a look at some guidelines that will help you understand what type of action to take at various points

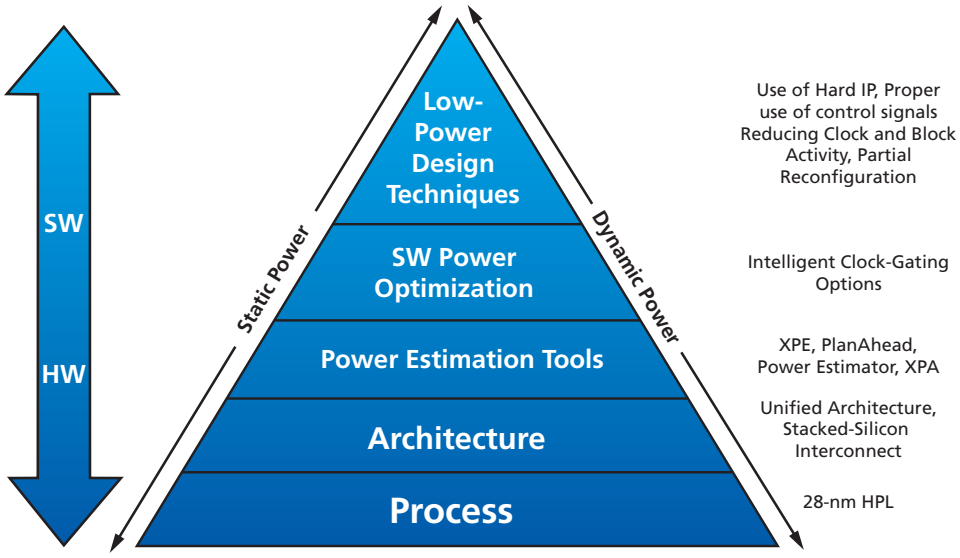


Figure 1 – Power reduction begins with device selection and embraces tools and software optimization.

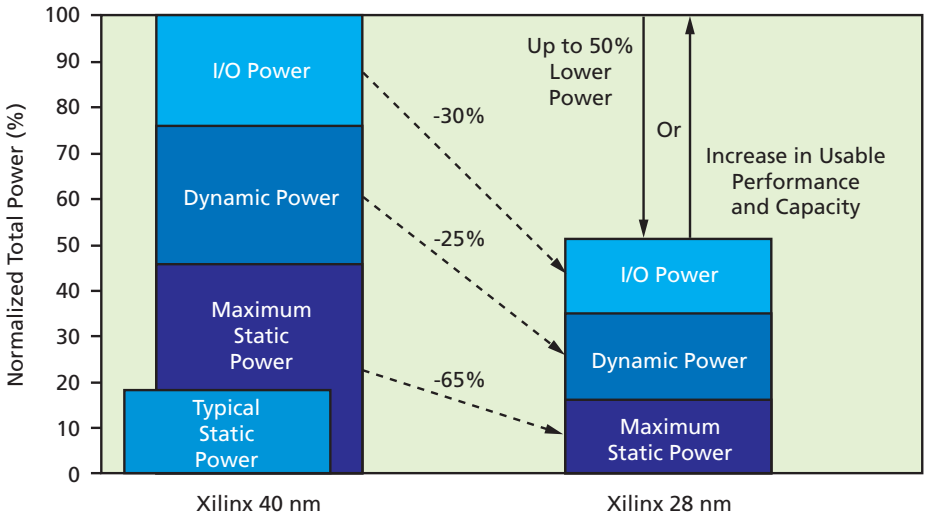


Figure 2 – Xilinx 7 series FPGAs consume half the power of devices built in the earlier 40-nm process.

in the design cycle to reduce the power consumption of an FPGA design. Clearly, having a thorough understanding of these issues early in the design process will yield the greatest reward.

Figure 1 illustrates different points in the design cycle, from FPGA selection through low-power design techniques.

7 SERIES PROCESS TECHNOLOGY

During FPGA selection, carefully consider the process technology, which helps you identify the leakage and performance of the device. The Xilinx® 7 series FPGAs are based on the 28 HPL (28-nanometer High-Performance, Low-Power) process, covering the high-performance space while also enabling significant power reduction (see cover story, *Xcell Journal* Issue 76). Choosing devices built on the lower-leakage HPL process eliminates the need for complex and expensive static-power-management schemes in an FPGA design.

FPGAs built with the 28 HP process have no performance advantage over 7 series FPGAs, while some other, competing FPGAs come with the severe penalty of more than twice the static power and present challenges in reducing leakage. Figure 2 shows a holistic power reduction approach for the 7 series family, which has half the overall power consumption of prior-generation, 40-nm FPGA devices.

7 Series FPGAs	C-Grade Devices	-2LE (1.0V)	-2LE (0.9V)
VCCINT	1.0V	1.0V	0.9V
Static Power	Nominal	-45%	-55%
Dynamic Power	Nominal	Nominal	-20%
Performance	-1, -2	-2	~ -1

Table 1 – Static, dynamic and performance power comparisons

Designers can choose a larger FPGA for purposes of development and later migrate to a smaller one in their production line. Choosing a smaller FPGA will not only bring down the cost but will also reduce the power consumption of the system.

All 7 series FPGAs are based on a unified architecture. This unified architecture enables easy upward and downward migration across different FPGA devices and families in the Xilinx 7 series portfolio. Refer to the "7 Series Migration Guide" (UG429) when considering design migration from Virtex®-6 or Spartan®-6 devices to or between 7 series device families.

XILINX STACKED-SILICON INTERCONNECT TECHNOLOGY

For larger systems, designers often choose multiple FPGAs. This type of architecture frequently requires the delicate and difficult task of moving data at rather high speeds among the various FPGAs. Choosing the larger 7 series FPGAs, such as the XC7V1500T and XC7V2000T devices, which are created using Xilinx stacked-silicon interconnect technology, can circumvent this issue. Simply stated, this SSI technology uses multiple dice residing on a silicon interposer that provides tens of thousands of connections among them, to create a single large device. One benefit of stacked-silicon interconnect technology is the reduction in maximum static power compared with a similar-size device on a standard monolithic die.

Stacked-silicon interconnect technology also provides a significant reduction in I/O interconnect power. Compared with having multiple FPGAs on a board, SSI technology boasts a reduction of I/O interconnect power by 100x (bandwidth/W) over an equivalent interface built with I/Os and transceivers. This dramatic reduction is due to all connections being built on-chip rather than having the power required to drive the signals off chip, enabling incredibly high speed and low power.

ENHANCED OPTIONS FOR VOLTAGE SCALING

Xilinx 7 series FPGAs offers significant voltage-scaling options.

The 7 series FPGAs offer an extended (E) temperature range (0–100°C) option for both -3 and -2L devices. Due to the headroom in the 28 HPL process, the -2LE devices can operate at 1 or 0.9 volt. These devices are referred to as -2L (1.0V) and -2L (0.9V). The -2L devices operating at 1.0V have the same speed-grade performance as the -2I and -2C devices, but with much lower static power. The -2L devices operating at 0.9V have performance similar to the -1I and -1C devices, but with even lower static and lower dynamic power.

At 0.9V, the voltage drop alone in these devices offers a static power reduction of around 30 percent. The voltage drop would also reduce performance, but Xilinx screens these -2L (0.9V) devices for speed and a tighter leakage specification. This screening method yields a 55 percent reduction in power at worst-case process compared with the standard-speed-grade devices.

By choosing a -2L family device, you can obtain additional power savings on dynamic power. Because dynamic power is proportional to $VCCINT^2$, a 10 percent reduction in VCCINT will provide a 20 percent reduction of power.

POWER ESTIMATION TOOLS

There is an extensive choice of tools available in the market today to help designers evaluate the thermal and supply requirements of an FPGA design throughout the development cycle. Figure 3 shows the Xilinx tools available at each stage of the FPGA development cycle.

At the outset of the design cycle, the XPower Estimator (XPE) spreadsheet provides an early estimation of power consumption even before the predesign and preimplementation phases of a project. XPE assists with architecture evaluation and device selection and helps in selecting the appropriate power supply and thermal-management components that may be required for the application.

The PlanAhead™ software estimates the design power distribution at the RTL level. Designers can specify the device operating environment, the I/O properties and the default activity rates for the design using constraints or by using the GUI. The PlanAhead software then reads the HDL code to estimate the design resources needed and reports the estimated power from a statistical analysis of the activity of each resource. With its access to more-detailed information about the design intent, the RTL power estimator should be more accurate than the XPower Estimator spreadsheet and less accurate than the post-place-and-route analysis done with the XPower Analyzer.

XPower Analyzer (XPA) is a tool dedicated to the power analysis of placed-and-routed designs. It provides a comprehensive GUI that allows a detailed analysis of the power consumed as well as thermal information for the specified operating conditions.

You can toggle between two different views to identify the power consumed either by type of blocks (clock trees, logic, signals, I/Os or hard IP such as block RAMs or DSP blocks) or over the design hierarchy. Both of these views enable you to perform a detailed power analysis. They provide a very efficient method for locating the blocks or parts of the design that are the hungriest in terms of power, thereby identifying places to begin power optimization efforts.

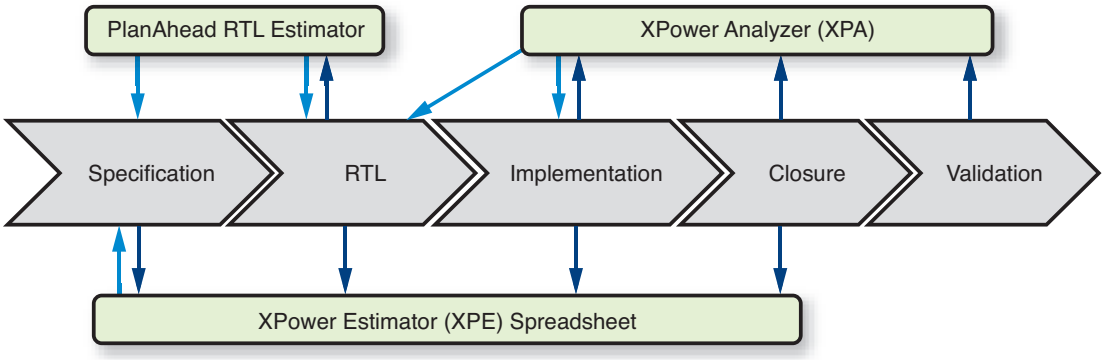


Figure 3 – Xilinx offers power estimation and analysis tools for every stage of the design cycle.

SOFTWARE POWER OPTIMIZATION

You can optimize designs using block RAMs for power by minimizing the number of simultaneously active block RAM ports. This optimization, enabled with the **–power yes** option in XST, modifies the decomposition of RAM or ROM descriptions that span multiple block RAMs. The optimization adjusts

address lines as well as port-enable and write-enable control signals to minimize the number of active block RAM ports at each clock cycle, while ensuring that your design meets timing constraints.

Next, force the most power-efficient mapping of block RAMs regardless of the impact on performance. Use the **block_power2** option to the **ram_style** constraint when you know that the timing paths related to this memory are not critical. Savings range from 15 percent to 75 percent.

Also, use the Area Optimization mode in XST. This option minimizes the number of resources your design will use. Note that when optimizing for area, performance may suffer.

An additional tactic is to enable activity-aware optimizations, another way of saying intelligent gating. These algorithms analyze the logic equations to detect for each clock cycle sourcing registers that do not contribute to the result. The software then utilizes the abundant clock-enable (CE) resources available in the FPGA logic to create fine-grained gating signals that neutralize useless switching activity. You control this intelligent clock and data gating with the **map -power high** option. Total core dynamic power reduction in excess of 15 percent is possible and in most cases, the additional gating logic inserted does not affect performance.

Another way to design for power is to use capacitance-aware optimizations. There are two main techniques:

- **Group clock loads:** This process reorganizes the placement of synchronous elements (such as flip-flops or DSP blocks) to minimize the reach of each clock net. When you place clock loads along a minimum number of horizontal or vertical clock spines, the software can disable unused branches in the clock region. This reduces both the clock resources and buffering requirements, which saves core dynamic power. This process is controlled by the **map -power on** option.

- **Group data loads:** This algorithm minimizes the total wire length in your design while ensuring that you meet performance requirements. Grouping data loads saves power because dynamic power increases with the fanout and the type and length of routing structures you use. The grouping algorithm, likewise enabled with the **map -power on** option, achieves power reduction by placing related logic closer together.

The ISE® Design Suite features predefined goals and strategies that are already tuned to enable power optimization at synthesis, map and place-and-route levels. This approach may be a good alternative to using nondefault constraint settings of all synthesis constraints. However, running this option can add some delay time on various paths.

Finally, Xilinx implementation tools automatically shut off unused transceivers, phase-locked loops, digital clock managers and I/Os. In 7 series devices, Xilinx has also added power gating of unused block RAM. Leakage in block RAM occurs only in blocks that you are using for a particular design, and not for all block RAMs on the device. Power is routed in the

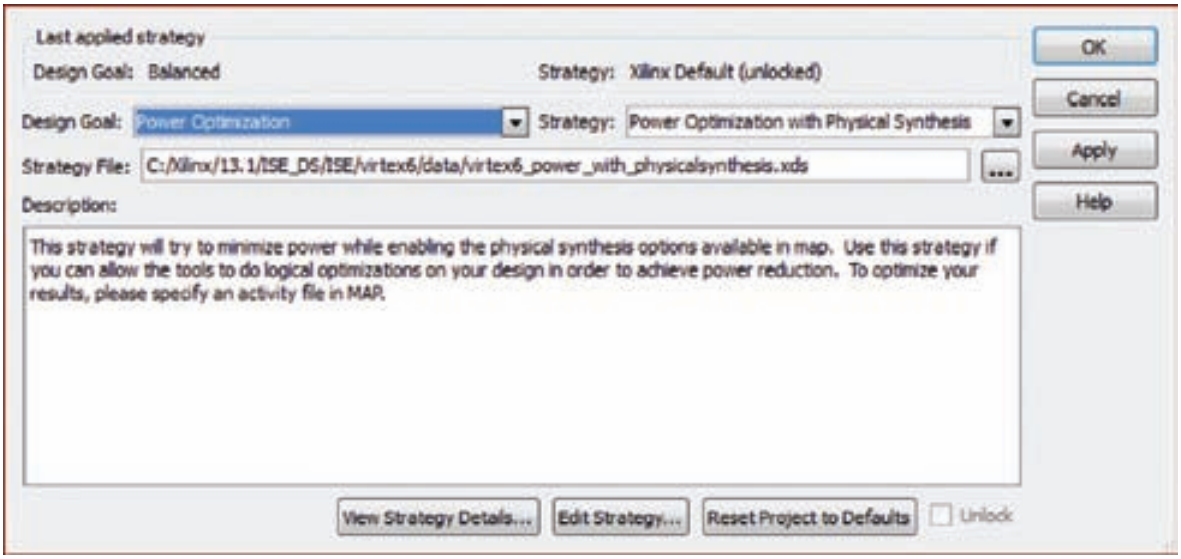


Figure 4 – Xilinx has built design goals and strategies for minimizing power into the ISE Design Suite.

To reduce power, you must look for opportunities to reduce the logic in the design. One of the first options is to use dedicated hardware blocks rather than implementing the same logic in CLBs.

device to the instantiated block RAM only, and disabled for the unused block RAMs.

LOW-POWER DESIGN TECHNIQUES

There are many tips and techniques that designers can explore to lower the power of an FPGA design. One of the first options is to use dedicated hardware blocks rather than implementing the same logic in CLBs. To reduce power, you must look for opportunities to reduce the logic in the design. This will allow you to use as small a device as possible and reduce static power consumption.

Using dedicated hard-IP blocks is one of the most important ways to lower both static and dynamic power, as well as to easily meet timing. Hard IP lowers static power because the total transistor count is less than an equivalent component with CLB logic.

As a general rule, you should attempt to infer resources as much as possible. You can steer the inferred resources individually, or as a group, toward the FPGA fabric or silicon resource via attributes in the code or within a constraint file. You can also leverage the Xilinx CORE Generator™ tool to customize the dedicated hardware for instantiating a specific resource.

Moreover, you can employ unused hard IP cleverly for other tasks that may not be obvious. DSP48 slices serve many logic functions such as multipliers, adders/accumulators, wide logic comparators, shifters, pattern matchers and counters. You can use block RAMs as state machines, math functions, ROMs and wide logic lookup tables (LUTs).

BEST USE OF CONTROL SIGNALS

The use of control signals (signals that control synchronous elements such as clock, set, reset and clock enable) can affect device density, utilization and performance. Following a few guidelines will help you keep the power impact to a minimum.

First, avoid using both a set and a reset on a register or latch. The flip-flops in Xilinx FPGAs can support both asynchronous and synchronous reset and set controls. However, the underlying flip-flop can natively implement only one set, reset, preset or clear at a time. Coding for more than one of these functions in the RTL code will result in the implementation of one condition using the SR port of the flip-flop and the other conditions in fabric logic, thus using more FPGA resources.

If one of the conditions is synchronous and the other is asynchronous, the asynchronous condition will be the one that gets implemented using the SR port and the synchronous condition in fabric logic. In general, it is best to avoid more than one set/reset/preset/clear condition. Furthermore, only one attribute for each group of four flip-flops in a slice determines if the SR ports of flip-flops are synchronous or asynchronous.

In addition, use active-high control signals. The control ports on registers are active high. Using active-low resets in an FPGA design is not recommended. Active-low signals use more lookup tables because they require inversion before they can directly drive the control port of a register. This inversion must be done with an LUT and thus takes up an LUT input.

Hence, active-low control signals may lead to longer runtimes and result in poor device utilization, which will affect timing and power.

Use active-high control signals wherever possible in the HDL code or instantiated components. When it's impossible to control a control signal's polarity within the design, you should invert the signal in the top-level hierarchy of the code. The I/O logic can absorb the inverter that's inferred without using any additional FPGA logic or routing, thereby resulting in better utilization, performance and power.

UNNECESSARY USE OF SETS OR RESETS

Unnecessary sets and resets in the code can prevent the inference of shift register LUTs (SRLs), LUT RAMs, block RAMs and other logic structures that might otherwise be inferred. Although designers may find it awkward, many circuits can be made to self-reset, or simply do not need a reset. For example, no reset is required when a circuit is only used for initialization of the register, because register initialization occurs automatically upon completion of configuration.

By reducing the use of unnecessary sets or resets, and with greater device utilization, designers can achieve better placement, improved performance and reduced power.

For more information on reset, please refer to *http://issuu.com/xcelljournal/docs/xcell_journal_issue_76/44?viewMode=magazine&mode=embed*.

With partial reconfiguration, designers can essentially time-slice an FPGA and run parts of the design independently. The design then requires a much smaller device, because not every part is needed 100 percent of the time.

Another area that demands attention if you are serious about lowering power is clock and block activity. You should take full advantage of BUFGMUX, BUFGCE and BUFHCE to gate an entire clock domain for power reduction. These constraints can pause the clock in an entire clock region. Likewise, for applications that only pause the clock on small areas of the design, use the clock-enable pin of the FPGA register.

Designs that spread across multiple clock regions utilize more clocking resources and hence consume more power. Whenever possible, place any intermittently used logic in a single clock region (Figure 5). This helps to reduce power. While the tools will attempt this automatically, some designs may require manual effort to achieve this.

Another important technique is to limit data motion (Figure 6). Instead of moving operands around the FPGA, move only the results. Using fewer and shorter buses leads to less capacitance, faster operation and less power consumption. Designers should also be careful while placing the pinout and the corresponding logic for their design during floorplanning.

PARTIAL RECONFIGURATION FOR LOWER STATIC POWER

One way to reduce static power is to simply use a smaller device. With partial reconfiguration, designers can essentially time-slice an FPGA and run parts of their design independently. The design then requires a much smaller device because not every part of the design is needed 100 percent of the time.

Partial reconfiguration has the potential to reduce dynamic power as well as static power. For example, many designs must run very fast, but that maximum performance might only be needed a small percentage of the time. To save power, designers can use partial reconfiguration to swap out a high-performance design with a low-power version of the same design—instead of designing for maximum performance 100 percent of the time. You can switch back to the high-performance design when the system needs it.

This principle can also apply to I/O standards, specifically when a design does not need a high-power interface all the time. LVDS is a high-power interface, regardless of activity, due to the high DC currents required to power it. Designers can use partial configuration to change the I/O from LVDS to a low-power interface, such as LVCMOS, at times when the design does not need the highest performance, and then switch back to LVDS when the system requires high-speed transmissions.

Making the best use of timing constraints is also important in low-power design. If you are operating in a temperature-controlled environment, remember that you can derate the part in order to meet timing. Be certain to only constrain the part to the maximum specified clock rate. Indicating that a faster clock rate is to be used does not generate a better design! Typically, it will use more fabric resources due to reduced

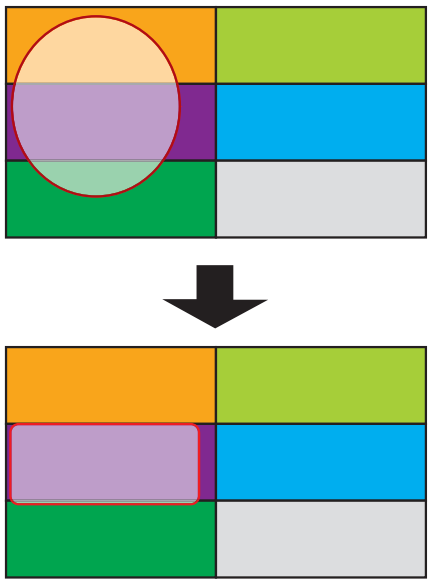


Figure 5 – Where possible, place any intermittently used logic in a single clock region.

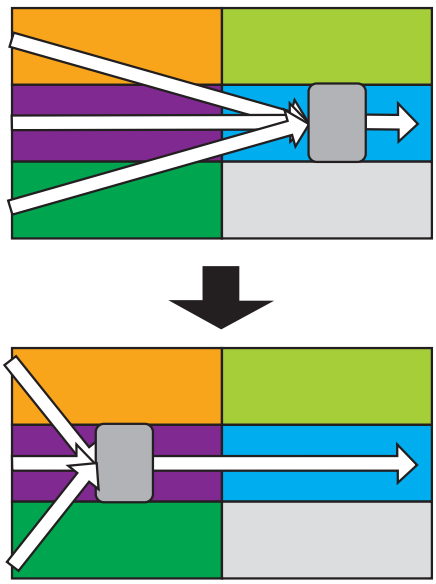


Figure 6 – Limit data motion; instead of moving operands around the FPGA, move only the results.

resource sharing, more logic/registers duplication, more routing and fewer inferences of FPGA dedicated features. All of these can significantly impact dynamic power.

I/O power has become a major contributor to total power. Some designs draw as much as 50 percent of the total power from I/Os, especially in memory-intensive systems.

The programmable slew rates and drive strength lower dynamic power in I/O drive. While many prefer fast differential I/O capabilities, not every interface requires it. There are standards such as HSLVDCI that can save considerable power in FPGA-to-FPGA communications and in lower-speed memory interfaces.

All Xilinx 7 series devices offer programmable slew rate and drive strength. Xilinx FPGAs sport digitally controlled impedance (DCI) technology, which can also be tristated. DCI eliminates termination power during memory write from the FPGA, so the device consumes termination power only during the read.

The 7 series devices incorporate a user-programmable referenced receiver power mode for HSTL and SSTL. You control these two programmable power modes on an I/O-by-I/O basis, which helps you reduce DC power by making trade-offs between power and performance.

TRANSCEIVER POWER PICTURE

Xilinx has optimized the 7 series FPGA transceivers for high performance and low jitter. These transceivers offer several low-power operating features, enabling designers to customize the flexibility of operation and granularity for balancing power and performance trade-offs.

In the 7 series FPGAs, the shared LC phase-locked loop can save a lot of power. For four-lane designs with an identical line rate (XAUI, for example), you can use a quad PLL (instead of an individual-channel PLL) to save power. Similarly, in some cases, because a PLL can run at both higher and lower rates within the range, it is better to select a lower operating range so as to save power.

You can also enable individual TX/RXPOWERDOWN options. PLL power down can be enabled in the lowest-power mode (say, in a system D3 state, which is mostly used in PCIe® systems).

EACH STAGE OF THE CYCLE

Understanding and implementing power-sensitive design techniques before you perform coding is the single largest mechanism for reducing system power. Using the various Xilinx tools at the appropriate stage of the design cycle will also help you meet power specifications, and provides the board designer with information on selecting the number, type and size of the necessary power supplies. Xilinx 7 series FPGAs provide unprecedented power efficiency through use of process technology and architectural design.

Many of the tips explained in this article are described in the FPGA Power Optimization training course. More information on Xilinx courses is available at *www.xilinx.com/training*.

Chandra Sekar Balakrishnan *is a solutions development engineer in the Xilinx Global Training Solutions team. Chandra Sekar develops content for Xilinx training courses. His areas of expertise are FPGA design and the embedded domain.*

Prior to joining Xilinx in May 2010, he was a project leader at Acme Technologies (Chennai, India), where he worked for more than eight years on projects that implemented FPGA-based embedded systems for Japanese semiconductor organizations.

Reach him at cbalakr@xilinx.com.

Dual-MicroBlaze Xilkernel System Eyes Automotive Apps

Memory segmentation and data-exchange strategies take lion's share of the implementation of deeply coupled master/slave multiprocessor systems.

by Heiko Wilken
Graduate Student
Hamburg University
of Applied Sciences, Germany
wilken.heiko@gmail.com

Marco Kirschke
Graduate Student
Hamburg University
of Applied Sciences, Germany
kirschke.marco@gmail.com

Bernd Schwarz
Professor Dr.-Ing.
Hamburg University
of Applied Sciences, Germany
schwarz@informatik.haw-hamburg.de

Multiprocessor system-on-chip (MPSoC) designs based on soft-core processors are gaining more and more importance in the context of embedded-system design. The architecture and management of memory regions represents a considerable part of the design effort so as to provide consistent data to multiple processors. For a prototype automotive application, we set out to build a dual-MicroBlaze™ system with support for Xilkernel, a kernel for the Xilinx® embedded processors, on both MicroBlazes.

We studied the common design workflows and processes according to the Xilinx specifications, reference designs and dual-processor design suite [1] before developing our prototype steering platform for an autonomous vehicle. [2] The design combines several functions such as collision avoidance, [3] lane detection [4] and automatic parking. The Xilkernel implementation provides library functions such as Posix threads in order to reutilize and partition a QNX-based collision-avoidance application.

TIGHTLY COUPLED

We built a tightly coupled processor system that contains two identically configured MicroBlaze processors, each connected to a 16-kbyte local BRAM in a Harvard architecture (Figure 1). Furthermore, they are also linked to a 128-Mbyte DDR2 memory module for code storage and data exchange. The Xilinx Platform Studio (XPS) Mutex core synchronizes the MicroBlazes' shared memory access. Both processors also communicate directly via a pair of Fast Simplex Links (FSLs), which support the development of master/slave and pipelined-processor design concepts. Applying the FSL interconnection instead of an XPS Mailbox for just two MicroBlazes increases performance while using fewer resources.

Two timer peripherals provide the interrupt sources for the Xilkernel's scheduling implementation. With the first development step, we connected a laser scanner to the system via UART interface to prepare the object detection of the collision-avoidance system. To employ the Xilkernel with an interrupt controller instead of a directly connected timer interrupt, we adapted the Xilkernel-related parameter `sysintc_spec` in the board support package (BSP) settings. [5]

We linked both MicroBlazes to the MicroBlaze Debug Module for debugging purposes and to output the STDIO over the JTAG UART of `microblaze_1`. We analyzed two implementations of the system, one for the XtremeDSP™ Starter Platform with a Spartan®-3A DSP 1800A FPGA and the second for the ML605 Evaluation Board with a Virtex®-6 XC6VLX240T FPGA.

MEMORY AND CACHE ARCHITECTURE

The Multi-Port Memory Controller (MPMC) couples the MicroBlazes through the Xilinx Cache Link (XCL) with the external DDR2 memory, and allocates the direct memory access. [6] Due to the uniform memory architecture, each processor has the same memory latency and access method. We equipped each of the MicroBlazes with an 8-kbyte instruction and 8-kbyte data cache, connected to a single MPMC PIM. This routing scheme allows us to associate a maximum number of eight MicroBlazes to one MPMC.

A data flow and instruction fetch bottleneck between the MPMC and the external memory originates from the single MPMC address- and data-bus connections. But the internal round-robin arbitration ensures that all requests get access to the memory sequentially. Furthermore, the MicroBlazes hold connections to the MPMC via their Processor Local Bus (PLB) to provide access to a non-cached shared-memory region.

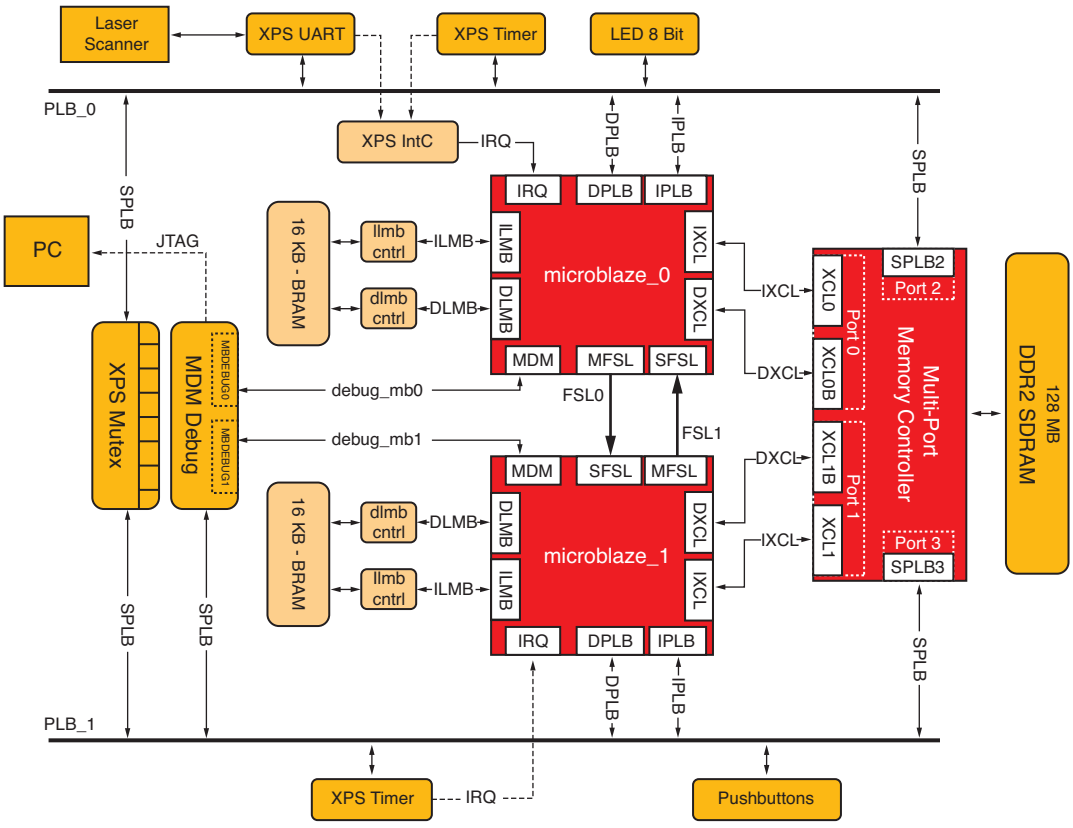


Figure 1 – The Dual-MicroBlaze-Xilkernel configuration for the XtremeDSP Starter Platform

In a multiprocessor environment it is important to ensure that the data caches are coherent. Therefore, we split the address range in the external memory into three partitions (Figure 2). Each MicroBlaze occupies its own address space and the XCL caches the data only if it is stored in its specific area. A third address space provides a separate section, which is accessible via the PLBs in order to exchange noncached shared data. With predefined variables in the linker script, the software engineer gets a pointer to the base address of the section. We configured the caches following the write-through strategy, because with a write-back cache organization, handling multiple write accesses causes more latency.

CONFIGURATION OF THE LINKER SCRIPTS

The linker script defines the memory segmentation in processor-based systems according to the hardware design information, the appropriate board support package and the software application itself. The XPS tool assigns fixed sections for the reset, interrupt and exception vectors of each MicroBlaze, which will be stored in the processor's block RAM. The common approach is also to store the .heap and .stack sections to the local BRAM to provide rapid access to local variables for thread executions.

Regarding the real-time operating system (RTOS) library functions, the size of the .text and .data sections becomes too large for the BRAM, and so we placed these functions in the external memory. In multiprocessor systems this leads to complications if the processors address the same memory regions. Therefore, the developer has to adapt the memory segmentation manually. As described above, both processors are only caching their dedicated memory regions and share access to a defined area beyond the code and data sections. Synchronized access to this shared-memory region is handled by requesting the XPS Mutex via the PLBs in order to exchange data between the MicroBlazes. For direct communication and synchronization within a master/slave or a pipelined-processor system, we employed the FSLs. This avoids the higher access latencies that multiple memory bus allocations cause when accessing shared memory.

We initially created the linker script with SDK Linker Script Generator. The script is a text file, structured in three partitions defining heap and stack sizes, the memory regions and the addresses for every single section. The section of code on the next page depicts the linker script for `microblaze_0` of the dual-MicroBlaze Xilkernel system. We modified the size of

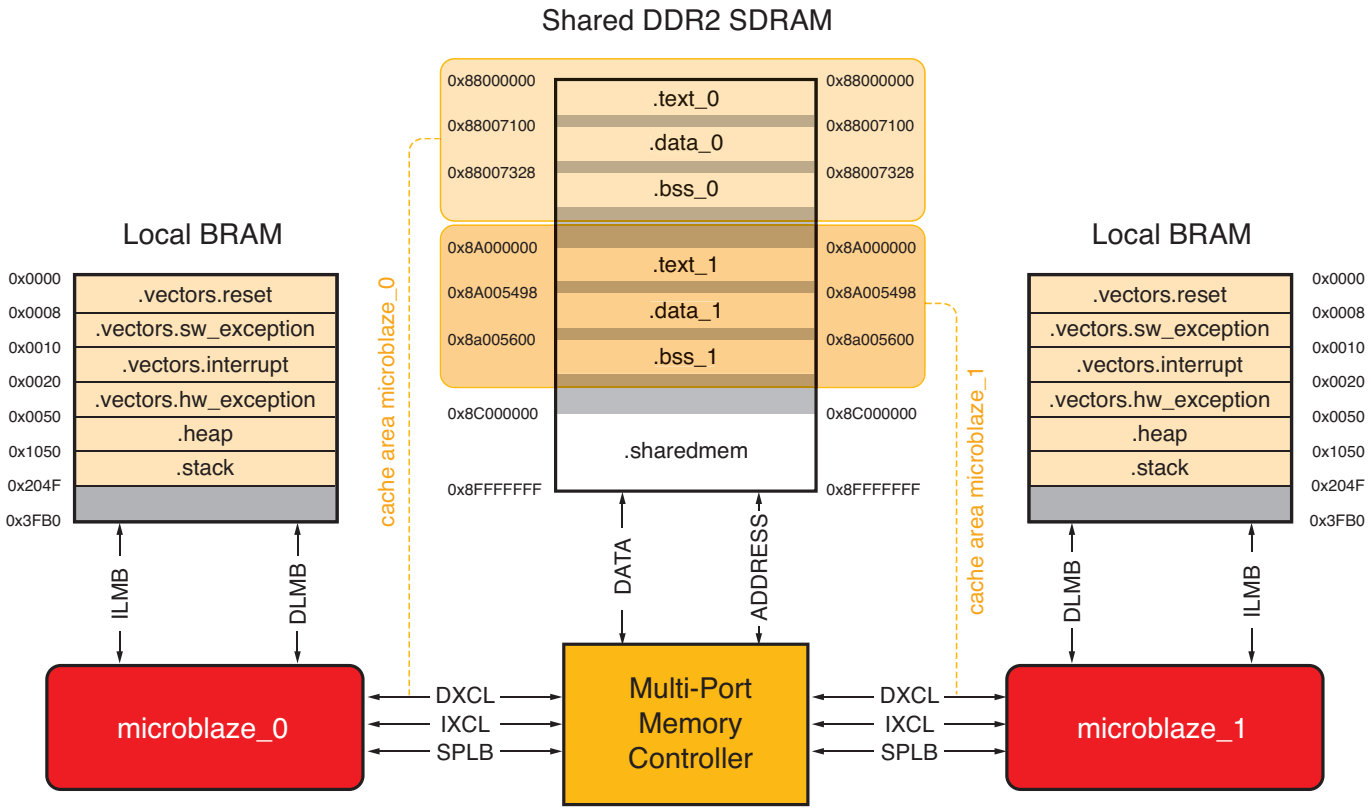


Figure 2 – Memory map of the DMX system

For direct communication and synchronization within a master/slave or a pipelined-processor system, we employed the Fast Simplex Links. This technique avoids the higher access latencies that multiple memory bus allocations cause when accessing shared memory.

the memory region for the code and data sections and added a new section for the shared-memory region. Accordingly, we adapted the linker script for `microblaze_1` as well, with a starting memory region for the code and data sections at address 0x8A000000 and an equivalent section for the shared memory.

```
/* Define heap and stack sizes for microblaze_0 */

_STACK_SIZE = DEFINED(_STACK_SIZE) ? _STACK_SIZE :
0x1000;
_HEAP_SIZE = DEFINED(_HEAP_SIZE) ? _HEAP_SIZE :
0x1000;

/* Define memories in the system for microblaze_0 */
MEMORY
{
  ilmb_cntlr_dlmb_cntlr :
ORIGIN = 0x00000050, LENGTH = 0x00003FB0
  /* resize the memory region for code and data sec-
tions*/
  DDR2_SDRAM_MPMC_BASEADDR :
ORIGIN =  0x88000000, LENGTH = 0x02000000
  /* new partition for shared memory added*/
  DDR2_SDRAM_SHARED_BASEADDR :
ORIGIN =  0x8C000000, LENGTH = 0x04000000
}
[...]
/* Define the sections, and where they are mapped in
memory */
SECTIONS
{
[...]
sharedmem : {
   __sharedmem_start = .;
   *(sharedmem)
   __sharedmem_end = .;
} > DDR2_SDRAM_SHARED_BASEADDR
[...]
}
```

DOWNLOAD AND DEBUG OPTIONS FOR MPSOC

The SDK environment offers two approaches for downloading, running and debugging the software of multiprocessor systems. On the one hand, the Eclipse-based tool provides the configuration of the download and STDIO behavior for the software projects. At the same time, the Xilinx Microprocessor Debug (XMD) Console is available to handle these tasks.

Additionally, the SDK reconfigures the FPGA with the bitstream and the .bmm file. According to the sections defined in the linker script, the Program FPGA Dialog optionally transfers the complete software sections directly to the BRAM associated with a specific processor. As discussed above, in multiprocessor systems we commonly outsource these sections to an external memory because the .elf files enlarge with applications that utilize Xilkernel library functions. In this case you choose bootloop in the Software Configuration settings of the Program FPGA Dialog to provide subsequent software downloads.

SDK ENVIRONMENT

The SDK provides Run & Debug configurations that cover the software project selection in the Project Explorer view and the opening of the "Run configurations..." in the Run menu. Every processor in the system has an associated Run & Debug configuration. In the first instance, select the project's .elf file in the main tab of the configurations dialog. Decide on "Reset Processor Only" as the reset type in the Device Initialization Tab to avoid resetting the entire system on resets.

Additionally, the STDIO directly prints to the SDK Console by configuring the STDIO Connection Tab. With a bundled group of the single processor's Run & Debug configuration, they all start running its software together. Finally, the SDK's Eclipse-based Debug View lets the developer insert breakpoints and provides information about variable assignment and memory allocation.

XMD CONSOLE

The most important XMD Console commands for our purpose are those with regard to the usage in multiprocessor systems. A more detailed listing is available directly out of the XMD Console with the help command. To apply the

Even a medium-size Spartan device offers considerable slice resources for additional accelerator IP connected to the PLB or FSL interfaces of the dual-MicroBlaze system.

XMD Console for downloading and debugging of multiprocessor systems, the command:

```
connect mb mdm –debugdevice cpunr <mb id>
```

links a processor via the MicroBlaze Debug Module (MDM) to the XMD Console. Then, the parameter `<mb id>` defines the identifier of each processor. After connecting all processors, the command:

```
target <trg no>
```

selects a MicroBlaze for the handling of a current target connection. The `target` command displays a table of connected targets. To download the .elf file to a specific processor target, the developer enters the full path and file name with the "dow" command:

```
dow <elf file>
```

You can either start the software of the selected target with the `run` command or step through the lines of execution with `stp`. If the MDM has a PLB interfacing the XMD Console reports, the JTAG UART outputs with the command:

```
read_uart start
```

A JTAG UART server receives these outputs by just defining a port, which makes the output available on a local host for terminal programs (for example, *HTerm* or *Tera Term*):

```
terminal jtag_uart_server <port>
```

PIPELINED SYSTEM

The system we created, based on our step-by-step design implementation of a dual-processor platform, consists of two Xilkernel-supporting MicroBlaze processors that share two communication interfaces via FSL and have access to shared memory. Each of the processors is configured with instruction and data caches that operate on separate memory regions in an external DDR2 SDRAM module. Furthermore, the XPS Mutex module synchronizes the shared-memory access to provide data consistency. Additionally, we adapted the linker scripts for the two MicroBlazes to allocate dedicated memory regions for the instruction and data sections of both processors and one shared-data section.

Table 1 shows the resource allocation and utilization for both the Spartan-3A DSP 1800A and the Virtex-6 XC6VLX240T FPGA with default synthesis constraints.

	Spartan-3A DSP 1800A		Virtex-6 LX240T	
	Used	Utilization	Used	Utilization
Slice DFFs	7,674	23%	10,891	3%
LUTs	9,790	29%	8,890	5%
BRAM	53	63%	49	11%
DSP48	6	7%	6	1%

Table 1 – Resource utilization of the dual-MicroBlaze Xilkernel system

Even a medium-size Spartan device offers considerable slice resources for additional accelerator IP connected to the PLB or FSL interfaces of the dual-MicroBlaze system.

These unexploited FPGA resources let us integrate an accelerator preprocessing unit for the laser scanner raw data. In the next step of this work, our focus will be on distributing the QNX-based collision-avoidance application and thread assignment to the processors.

Because the obstacle identification, environment modeling and distance control build a chain of conditional and sequential calculations, we are in favor of a pipelined-processor system. In future work, we plan the successive integration of camera-based lane detection, automatic parking and other proposed components, like vehicle odometry and cruise control.

References

1. Xilinx, Dual Processor Reference Design Suite, XAPP996, 2008, http://www.xilinx.com/ support/documentation/application_notes/xapp996.pdf.

2. "SVC-Collision Avoidance Assistant," HAW Hamburg, 2011, http://www.informatik.haw-hamburg.de/ schwarz.html.

3. D. Schetler, "Automatic collision avoidance with a laser scanner-based distance control for a driverless transport system," Master's Thesis, HAW Hamburg, 2007.

4. D. Mellert, "Modeling of video-based lane detection with the Xilinx System Generator for an SoC-platform," Bachelor's Thesis, HAW Hamburg, 2010.

5. Xilinx, Xilkernel (v5.00.a) - UG708, 2011, http://forums.xilinx.com/xlnx/attachments/xlnx/EMBEDDED/4789/1/xilkernel_v5_00_a.pdf.

6. Xilinx, Multi-Port Memory Controller (MPMC) (v6.03.a) - DS643, 2011, http://www.xilinx.com/ support/documentation/ip_documentation/mpmc.pdf.

FPGAs Made Accessible

JTAG-HS1™

Lowest-cost, highest-speed programming solution for Xilinx® devices

- Compatible w/ all Xilinx tools & devices
- Up to 30 Mbits/sec on JTAG bus
- Supports 6 & 14-pin programming headers
- Volume pricing available

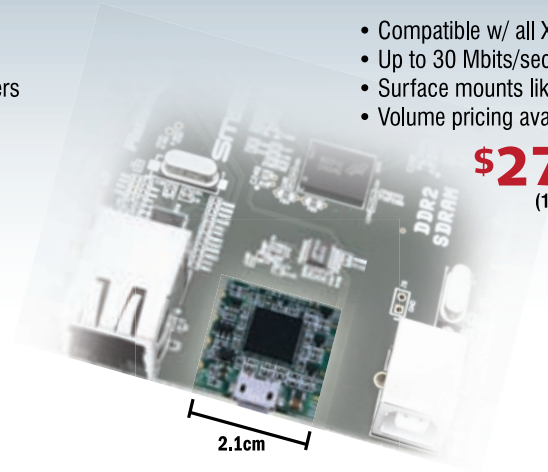
$49.99

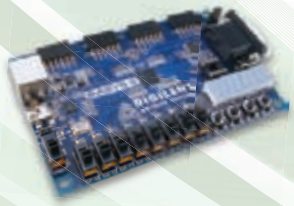
JTAG-SMT1™

Embedded programming for Xilinx® devices

- Compatible w/ all Xilinx tools & devices
- Up to 30 Mbits/sec on JTAG bus
- Surface mounts like any other component
- Volume pricing available

$27.99
(100+)





BASYS 2™

The Ultimate FPGA Starter

- Spartan® 3E
- USB-powered
- Ideal first time FPGA users

$49.00
(Student Price)

NEXYS 3™

Spartan®-6 power at an entry-level price

- Spartan®-6 LX16
- 32MB PCM & 16MB RAM
- 10 / 100 Ethernet, USB UART
- USB host for mouse & keyboard

$119.00
(Academic Price)

ATLYS™

Advanced video processing w/ Spartan-6 power

- Spartan®-6 LX45
- 128MB DDR2 & 16MB PCM
- Gigabit Ethernet, USB UART
- 4 HDMI ports, AC97 audio

$199.00
(Academic Price)

(509) 334-6306

DIGILENT®
BEYOND THEORY

www.digilentinc.com

Xilinx Tool & IP Updates

Xilinx is continually improving its products, IP and design tools as it strives to help designers work more effectively. Here, we report on the most current updates to the flagship FPGA development environment, the ISE® Design Suite, as well as to Xilinx® IP. Product updates offer significant enhancements and new features to three versions of the ISE Design Suite: the Logic, Embedded and DSP editions. Keeping your installation of ISE up to date is an easy way to ensure the best results for your design. On Oct. 25, 2011, Xilinx will release the ISE Design Suite 13.3 and make it available from the Xilinx Download Center at www.xilinx.com/download.

For more information or to download a free 30-day evaluation of ISE, visit www.xilinx.com/ise.

DOCUMENTATION NAVIGATOR

This new application allows users to view and manage Xilinx design documentation (software, hardware, IP and more) from one place, with easy-to-use download, search and notification features. To try out the new Xilinx Documentation Navigator, now in open beta, go to *www.xilinx.com/support*.

ISE DESIGN SUITE: LOGIC EDITION

Front-to-Back FPGA Logic Design

Latest version number: 13.3
Date of latest release: October 2011
Previous release: 13.2

Revision highlights:
A newly redesigned user interface for PlanAhead™ and IP suite improves productivity across system-on-chip design teams and represents progress toward true plug-and-play IP that targets Spartan®-6, Virtex®-6 and 7 series FPGAs.

PlanAhead design analysis tool: Xilinx has further enhanced the graphical user interface (GUI) to display RTL sources hierarchically according to how they are instantiated in the HDL. This feature lets the user visualize which source files contain the top-level logical hierarchy. It is useful for large numbers of sources and for projects that integrate lots of logic from other developers or from IP.

Team Design: New to ISE Design Suite 13 is a team design methodology using PlanAhead which addresses the challenge of multiple engineers working on a single project by providing a methodology for working in parallel. Building on the design preservation capability made available in ISE Design Suite 12, the team design flow provides additional functionality and allows you to lock down early implementation results on completed portions of the design without having to wait for the rest of the design team. This new capability facilitates faster timing closure and timing preservation for the remainder of the design, increasing overall productivity and reducing design iterations.

Xilinx Power Estimator (XPE) and Power Analyzer (XPA): These tools now offer improved quick design estimates, along with memory interface and transceiver enhancements with dedicated configuration windows, providing more accurate power estimation.

ISE DESIGN SUITE: EMBEDDED EDITION

Integrated Embedded Design Solution

Latest version number: 13.3
Date of latest release: October 2011
Previous release: 13.2

Revision highlights:
All ISE Design Suite editions include the enhancements listed above for the Logic Edition. The following enhancements are specific to the Embedded Edition.

Xilinx Platform Studio (XPS): The 13.3 release offers many enhancements, including a new two-page setup for Base System Builder for easier configuration. XPS now supports the Kintex™ KC705 platform and handles single or dual AXI4-based MicroBlaze™ designs. Also, the Create/Import IP wizard now supports AXI4, AXI4-Lite and AXI4-Stream IP.

SDK enhancements: The latest version supports MicroBlaze v8.20a, now with a 512-bit data width for AXI cache interconnects. Xilinx has updated the Software Development Kit to Eclipse 3.6.2 and CDT 7.0.2 releases to provide stability and enhancements in this open-source platform.

IP enhancements: The 13.3 release offers new AXI PCIe® and AXI QuadSPI cores. In addition, Xilinx has improved AXI v6 DDRx read/write arbitration.

EDK overall enhancements: The Embedded Development Kit now provides consistent SDK workspace selection behavior across Project Navigator, Xilinx Platform Studio (XPS) and the SDK.

ISE DESIGN SUITE: DSP EDITION

For High-Performance DSP Systems

Latest version number: 13.3
Date of latest release: October 2011
Previous release: 13.2

Revision highlights:
All ISE Design Suite editions include the enhancements listed above for the Logic Edition. Specific to the DSP Edition, version 13.3 introduces single, double and custom floating-point precision support, along with added support for Artix™-7 and Virtex™-7. Additionally, Xilinx has added the capability to generate PlanAhead projects supporting System Generator design, and production support for VDMA Interface 4.0 and MATLAB® 2001a.

XILINX IP UPDATES

Name of IP: ISE IP Update 13.3
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 application-specific standard parts.

Latest version number: 13.3
Date of latest release: October 2011

Installation instructions:
www.xilinx.com/ipcenter/coregen/ip_update_install_instructions.htm

Listing of all IP in this release:
www.xilinx.com/ipcenter/coregen/13_3_datasheets.htm

Revision highlights:
Starting with version 13.1, all ISE CORE Generator™ IP supports Kintex-7 and Virtex-7 devices.

New CORE Generator IP cores are available with Artix-7 and Virtex-7 XT support. However, Artix-7 and Virtex-7 XT device family support in the 13.3 release continues to be Limited Access. The set of cores supporting Artix-7 and Virtex-7 XT provide beta support for these two device families.

New CORE Generator software and IP cores:

- AXI DataMover v3.00.a – This key piece of AXI infrastructure IP enables high-throughput transfer of data between the AXI4 memory-mapped domain and the AXI4-Stream domain.
- Chroma Resampler V1.0 (AXI4-Lite) – This IP converts between chroma-subsampled YCbCr formats 4:4:4, 4:2:2 and 4:2:0. It also supports progressive and interlaced video.
- JESD204 v1.1 – Designed to the Jedec JESD204B standard, which describes the serial data interface and link protocol between data converters and logic devices, this IP supports line rates of up to 6.25 Gbps on one, two or four lanes.
- SRIO Gen 2, v1.2 – The Serial RapidIO Gen 2.1 IP soft core supports x1, x2 and x4 lane widths for line rates up to 6.25 Gbps.

Additional IP supporting AXI4 interfaces: Xilinx has updated the latest versions of CORE Generator IP with production AXI4 interface support. In general, the AXI4 interface is supported by the latest version of an IP core for Virtex-7, Kintex-7, Virtex-6 and Spartan-6 device families. Older "production" versions of IP continue to support the legacy interface for the respective core on Virtex-6, Spartan-6, Virtex-5, Virtex-4 and Spartan-3 device families only. For more detailed AXI IP support information, see *www.xilinx.com/ipcenter/axi4_ip.htm*.

For a comprehensive listing of cores that have been updated in the 13.3 release, see *www.xilinx.com/ipcenter/coregen/updates_13_3.htm*.

Xpress Yourself in Our Caption Contest

DANIEL GUIDERA

If you have a yen to Xercise your funny bone, here's your opportunity. We invite readers to step up to our verbal challenge and submit an engineering- or technology-related caption for this cartoon by Daniel Guidera, showing just how central coffee can be in keeping the engineer's creative juices flowing. The image might inspire a caption like "Now that Facilities has widened the door to get this in, let's start on our marketing plan."

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*. The winner and two runners-up will each receive an Avnet Spartan®-6 LX9 MicroBoard, an entry-level development environment for evaluating the Xilinx® Spartan®-6 family of FPGAs (*http://www.xilinx.com/products/boards-and-kits/AES-S6MB-LX9.htm*).

The deadline for submitting entries is 5:00 pm Pacific Time (PT) on Jan. 2, 2012. So, put down your cup of joe and get writing!

GINA PETER, marketing manager at Pentek (Upper Saddle River, NJ), won first place with this caption for the laboratory "beach" scene in Issue 76 of *Xcell Journal*:

DANIEL GUIDERA

"Looks like we have to explore some other cooling options."

Congratulations as well to two runners-up. Like our winner, they will each receive an Avnet Spartan®-6 LX9 MicroBoard.

"FPGA engineers—taking concurrency to a whole new level."

– Robin Debreuil, software engineer/owner, Debreuil Digital Works (Manitoba, British Columbia)

"So this is what you were referring to when you told the boss that you had found the optimal thermal management solution for the new equipment configuration?"

– Robert Inkol, senior scientist, DRDC (Ottawa)

XILINX

SYNOPSYS®
Predictable Success

Now Available!

The industry's first methodology manual for FPGA-based prototyping of SoC Designs

FPGA-Based Prototyping Methodology Manual:
Best Practices in Design-for-Prototyping

Go to:
http://www.synopsys.com/fpmm

- Purchase the book or download a free ebook version
- Become involved with the online community for prototypers

XILINX®

Twice the performance. Half the power.

Innovate without compromise with the Xilinx 7 series FPGAs. The new 7 series devices are built on the only unified architecture to give you a full range of options for making your concepts a reality. Meet your needs for higher performance and lower power. Speed up your development time with next-generation ISE® Design Suite. And innovate with the performance and flexibility you need to move the world forward.

www.xilinx.com/7

ARTIX 7™
LOWEST SYSTEM COST

KINTEX 7™
BEST PRICE / PERFORMANCE

VIRTEX 7®
HIGHEST SYSTEM PERFORMANCE

2011 WINNER
EE Times
ACE AWARDS

PN 2491