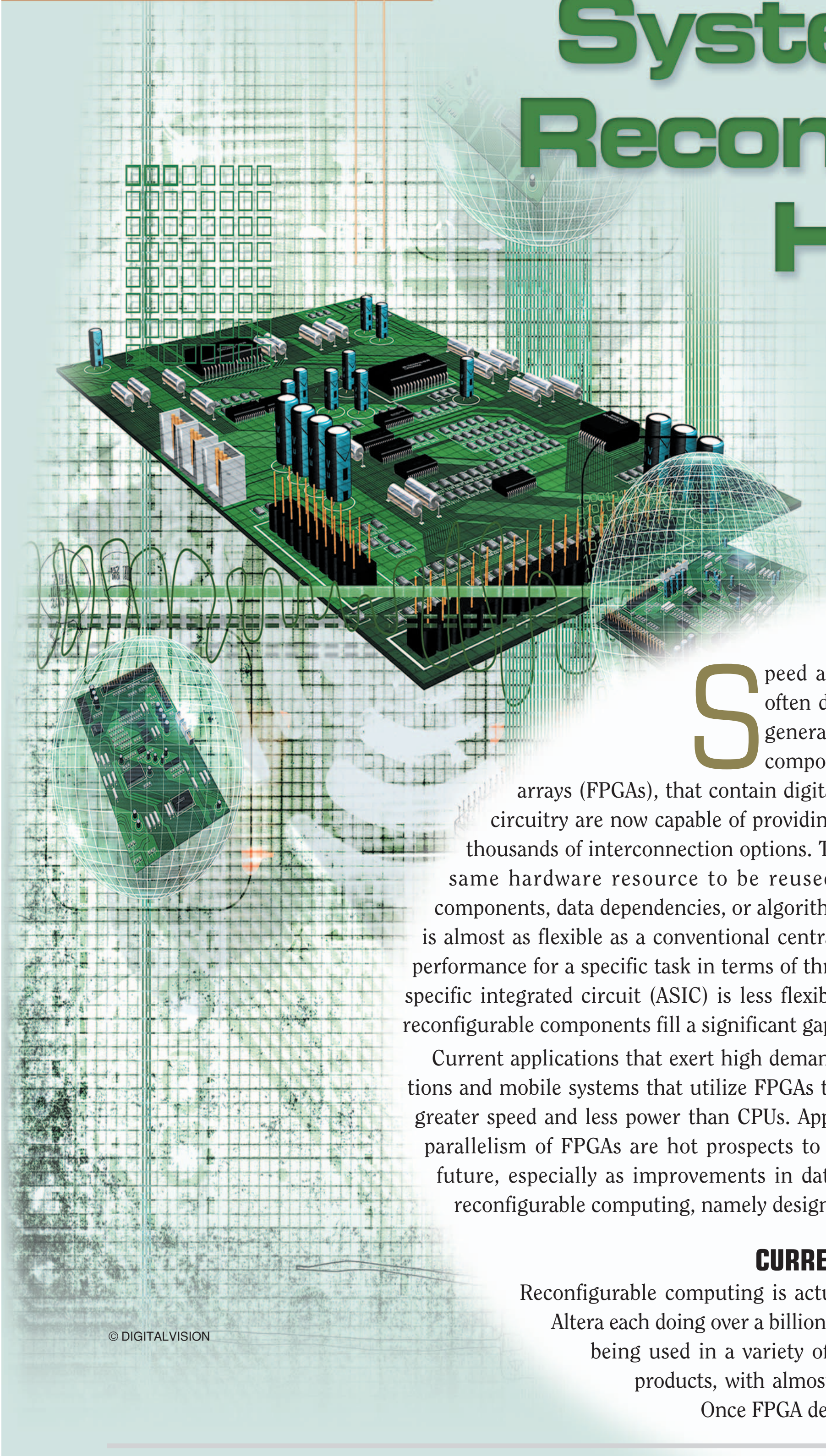
Don Bouldin

# Enhancing Electronic Systems with Reconfigurable Hardware

## Employing the Flexibility of Field-Programmable Gate Arrays



© DIGITALVISION

Speed and power requirements of electronic systems often dictate the use of hardware components instead of general-purpose processors. Reprogrammable hardware components, commonly called field-programmable gate arrays (FPGAs), that contain digital logic and interconnects and occasionally analog circuitry are now capable of providing several million logic gates on a single chip with thousands of interconnection options. They can be reconfigured at run time, enabling the same hardware resource to be reused depending on its interaction with external components, data dependencies, or algorithm requirements. As depicted in Figure 1, an FPGA is almost as flexible as a conventional central processing unit (CPU) but generally has higher performance for a specific task in terms of throughput/Watt. On the other hand, an application-specific integrated circuit (ASIC) is less flexible but may have even higher performance. Thus, reconfigurable components fill a significant gap between CPUs and ASICs.

Current applications that exert high demand for reconfigurable systems include communications and mobile systems that utilize FPGAs to provide more flexible operations than ASICs yet greater speed and less power than CPUs. Applications that can benefit from the variable-grain parallelism of FPGAs are hot prospects to emerge as high-volume applications in the near future, especially as improvements in data movement are made. The principal enabler of reconfigurable computing, namely designer productivity, is discussed in this article as well.

### CURRENT APPLICATIONS

Reconfigurable computing is actually doing quite well these days, with Xilinx and Altera each doing over a billion dollars in annual revenue [1], [2]. FPGA devices are being used in a variety of applications from communications to consumer products, with almost half of their revenue related to communications. Once FPGA device capacities exceeded 10,000 gates, they began to

8755-3996/06/$20.00 ©2006 IEEE

be used for more than just logic replacement and ASIC prototyping. The fact that they are cheaper (in low-volume situations) and more flexible than ASICs and faster and lower in power consumption than CPUs has led to their widespread use in communication systems. A stock-price comparison in [3] of Xilinx (the principal supplier) and CISCO (the principal customer) makes this very evident. In communications applications such as Internet routers and wireless base stations, the FPGA is an auxiliary device that offloads a changeable but high-speed task from other computing components.

Portable, battery-powered applications, which can benefit from the variable precision arithmetic that can be performed by FPGAs, are a natural for these devices. Rather than conform to the fixed bit-width of the arithmetic unit of a CPU, these systems can use whatever bits they need and only when they are needed. Several popular conferences (FCCM, ERSA, FPGA, and FPL) [4] annually address the underlying principles that lead to the choice of reconfigurable systems for these applications.

Even though the FPGA business is expected to reach US$5 billion annually in 2007, the ASIC market is about five times larger due to its higher volume. As ASIC mask sets escalate past US$1 million per design in the next few years, FPGA devices will capture a greater share of this large market [2].

## EMERGING APPLICATIONS

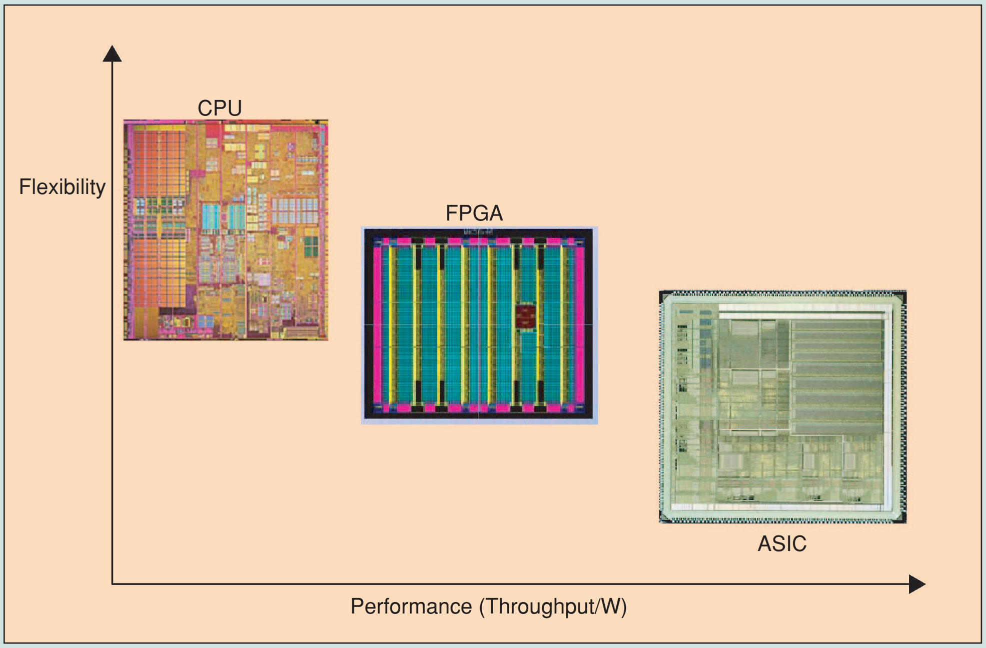
The characteristics of the applications we have seen to date primarily exploit the fact that FPGAs have flexible resources like CPUs but are faster and use less power. Furthermore, current applications of FPGAs have relied on specialists to use a hardware description language (HDL) to program them. However, this situation cannot continue if we want to transition to new applications that will surely involve multiple types of computing elements and will strive to exploit the parallelism that FPGAs can provide. The key to success will lie in improving designer productivity and the associated tool environments.

Xilinx and Altera have already introduced their programmable system-on-chip (SoC) platforms, which merge the CPU and FPGA into a single package [1], [2]. This provides higher bandwidth, lower latency, and lower power than having them in separate packages on a board. This is a powerful combination of computing elements in which we generally use the CPU to perform control and housekeeping functions while the FPGA is used to perform direct execution of a variety of arithmetic operations.

To make use of a programmable SoC platform, we must support hardware/software codesign. This is especially common in real-time embedded systems that must operate at low power. These platforms support the use of both C and HDL and require designers to partition the design and then integrate the two domains.

One fairly recent development is the advent of SystemC [5]. This augmentation of the C language allows both the software and hardware portions to be captured at the behavioral level so they can be simulated together. The HDL portion can even be generated automatically. Optimization of the HDL code can be performed later if necessary. One of the principal advantages of this approach is that the same test bench that is used at the SystemC level gets reused after synthesis.

Manufacturing can now produce SoC or system-in-package (SiP) designs that contain millions of transistors. NEC and other semiconductor foundries offer structured ASIC platforms that include programmable user logic [6]. At the University of Tennessee, we have developed an open SoC that includes only cores that can be obtained at no charge [7]. We are planning to include an embedded FPGA in our SoC like we did a decade ago when we developed an SiP[8]. Back then we called it a multichip module and got companies to donate bare dies to us. This project included a Motorola digital signal processor, a Xilinx FPGA, two SRAM chips, and an EEPROM as well as a tiny analog-to-digital converter. Initially, we planned to use the FPGA for glue logic and external communications. Of course, it also provided a means to add functionality that was not known at the time of manufacture. But then we realized we could also use it for diagnostics to test the other components inside the package. After all, those dies had been tested at various levels and were not necessarily 100% known



*1. Flexibility versus performance for CPU, FPGA, and ASIC.*

good die. So, upon power-up, we loaded in the diagnostics and checked out the other components. In some cases, we could even provide a work-around if we detected some failures.

DAFCA (design automation for flexible chip architectures) is a start-up company on the outskirts of Boston that provides presilicon and postsilicon tools for including embedded FPGA structures in SoCs [9]. A two-dimensional fabric can be used just like the FPGA in our SiP to provide flexibility for new functions. DAFCA's methodology supports wrapping each user block with a one-dimensional fabric to assist designers in debugging their chip before and after fabrication. In some cases, the wrapper can even provide work-arounds that enhance the reliability of the SoC.

Speed and power requirements of electronic systems often dictate the use of hardware components instead of general-purpose processors.

The parallelism afforded by FPGAs has been underutilized in most of the applications to date, so exploiting this should open some new doors for reconfigurable computing. Designers need to have better tools to help them determine how best to use coarse-grain and fine-grain parallelism with variable-bit precision. We also need to be able to manage run-time reconfiguration.

2. AFRL heterogeneous high-performance computer [10].

An emerging application for reconfigurable computing is acceleration for scientific computing. The Air Force Research Lab in Rome, New York, has acquired a Teraop machine, shown in Figure 2, constructed entirely from commercial off-the-shelf components [10]. The machine consists of 48 nodes, each with Xeon dual CPUs and two 6 M-gate Xilinx FPGAs. They are using the reconfigurable processors to perform parallel operations at the pixel level to find features in the data that can then be processed at the symbolic level by conventional CPUs.

NASA Langley is using a collection of FPGAs and memory chips built by Star Bridge Systems to perform finite element modeling of the Space Shuttle [11]. A model of the aircraft wing is partitioned into a three-dimensional mesh as shown in Figure 3, in which each element requires the solving of multiple sets of equations or matrices. Many researchers have indicated that finite element modeling has an insatiable appetite for computing since researchers are always wanting finer meshes and smaller time steps than those currently available.

Traditionally, scientific computing has relied on vector processors that have floating-point units built into the silicon. Now that advanced FPGA devices contain dedicated multipliers, they can support floating-point operations interleaved with the fixed point of any desired precision. The source code and test benches from a research group at the University of Southern California are expected to be made available soon on the Web at no charge [12].

Reconfigurable computing can be used to process tons of data in parallel. For example, large data collection sensors that acquire frames of images at various frequency bands for extended periods of time already exist. We have the potential of assigning FPGA logic resources to process this low-level data in parallel. The Air Force Research Lab at Kirtland Air Force Base has developed a collection of modules with thousands of inputs and outputs (I/O) arranged in a matrix or area-array that permits data to be moved in parallel rather than converted from parallel to serial and back again [13]. High-density interconnect packaging and three-dimensional stacking of chips offer significant potential for this application area.

Another application area that can benefit from clever data movement is encryption. CPUs are often too busy to handle this task, so it is off-loaded to a reconfigurable computing element that performs the operation as the data passes through the network [14]. It makes us feel like we are getting the encryption for free since we just introduce a slight latency into the network but don't really have to count the data transfer time in our calculations. As personal digital assistants and other wireless devices become more pervasive, we are likely to see a significant increase in this means of protecting information.

Cray and SRC Computers now offer scientific computing machines that have FPGAs embedded in the I/O fabric so they can perform computing operations during the data movement [15], [16]. Often, the FPGA filters through the massive raw data to detect features of interest to forward on to the next level of processors.

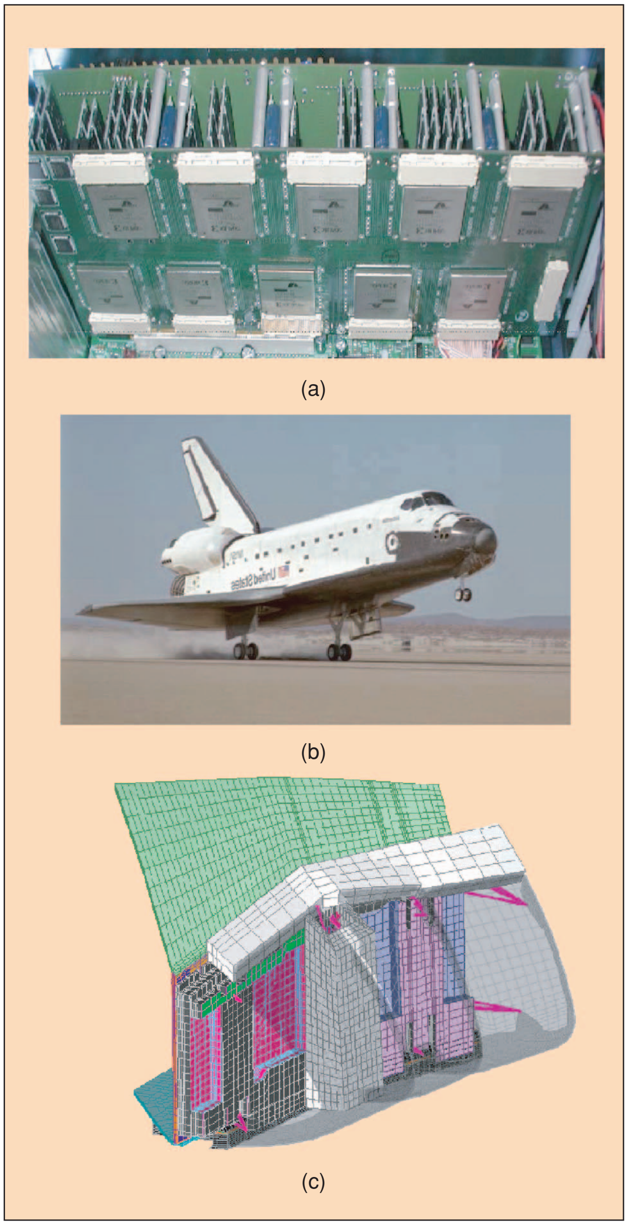
Reconfigurable computing is actually doing quite well these days.

A particularly impressive use of the reconfigurability of FPGAs was done by Cameron Patterson while he was working for Xilinx [17]. He rerouted the wiring resources on-the-fly while computing various rounds of an encryption algorithm and was able to beat the best ASIC available at that time. Being able to recognize when it is advantageous to do this may remain a human task for a long time, but we could certainly provide better tools that would enable a designer to implement and try out his ideas quickly.

Dynamic power consumption occurs on each clock edge when the FPGA is actively computing, while static power consumption occurs due to leakage when the internal flip-flops are just holding data. Over the past few years, dynamic power consumption per gate of FPGAs has continued to drop, but it has been largely offset by increased density. Meanwhile, leakage current has increased with the decrease in transistor feature size and is now a significant percentage of the total power consumption of FPGAs. The latest low-cost FPGAs, such as Xilinx's Spartan-3, Altera's Cyclone-II, Lattice's EC/ECP/XP, Actel's ProASIC-3, and QuickLogic's Eclipse II families, are all aimed at high-volume applications that often run on batteries or have limited cooling capability or restricted power supplies, all of which make lower-power operation an absolute necessity. Inside an FPGA, very thin oxide layers leak current even when transistors are not switching. To avoid the possibility of thermal runaway in 90-nm processes, Xilinx uses a thicker oxide in its Virtex-4 family for transistors involved in the routing and configuration circuitry that do not require rapid toggle rates because they generally remain in a constant state once configured. Altera dropped the traditional four-input look-up table (LUT) in favor of a seven-input variable adaptive logic module when developing its 90-nm devices. Thus, the new devices have decreased logic granularity and reduced routing-related power overhead. The devices also employ low-K dielectrics and longer transistors with increased Vt (threshold voltage) for nonperformance-critical paths.

On the tool side, every FPGA vendor offers software that will help estimate power consumption based on a spreadsheet. The designer supplies estimates of parameters like logic, memory, and I/O utilization, clock frequencies, toggle rates, and operating temperatures. The tool then produces an estimate of power consumption for those conditions. While these early estimates are the least accurate, they can serve as the basis for deciding which FPGA family to utilize. Xilinx provides Web Power Tools for preimplementation power estimates and XPower Tools for postimplementation power analysis. These tools deliver results that correlate well to actual silicon measurements when used correctly. The best power estimates are those done on a completed and routed design that has been loaded into the XPower tool and stimulated with a functionally accurate set of stimulus vectors. System architects can estimate power using high-level design details and make intelligent design choices on clock frequencies, the implementation of a function using hard IP (intellectual property blocks) or logic, the type of I/O standard to use, and other factors.

While power optimization tools have been available for ASIC design for a while, automated power reduction is still a relatively immature science for FPGAs. In ASICs, clock and power gating, buffer sizing, and voltage scaling can be employed by automated power-optimization tools. In FPGAs,


(a)

(b)

(c)

3. (a) NASA FPGA-based hypercomputer, (b) Space Shuttle, and (c) mesh for finite element modeling.

however, the options are somewhat limited by the underlying architecture, and the opportunity for improvement is reduced by the dominating presence of static power related to configuration circuitry. Nevertheless, since power is becoming a hot topic in FPGA design, serious research is underway into techniques and tools [18].

## DESIGNER PRODUCTIVITY AND DESIGN TOOLS

It is likely that graphical tools will be increasingly used for these emerging applications. The HDL expert can write modules that can be added to a library. Then, an application designer can invoke these modules in a schematic, simulate them, and eventually get the design implemented in the FPGA. The graphical tool is a means of reusing code. It is also a natural way of expressing parallelism since we expect all of the modules on the screen to be operating in parallel. Altera, Xilinx, Synplicity, and Star Bridge Systems now offer these [1], [2], [19], [20]. Generally, to be successful, libraries of modules are developed for narrowly defined domains.

A very ambitious project is currently underway at the University of Florida [21]. Researchers there are seeking to unify existing tools and fill in the gaps. Their list of desirable features required to fully exploit the capabilities of reconfigurable computing (RC) elements includes:

- dynamic RC fabric discovery and management
- coherent multitasking, multiuser environment
- robust job scheduling and management
- design for fault tolerance and scalability
- heterogeneous system support
- device-independent programming model
- debug and system health monitoring
- system performance monitoring into the RC fabric
- increased RC device and system usability.

On the other hand, standard C/C++ programming tools are being used by Stretch, Inc. to enable the automatic configuration of their off-the-shelf processors to achieve extraordinary performance, easy and rapid development, and significant cost savings. The product is flexible enough to address diverse market applications including consumer, telecom, networking, video, and medical applications. It also can support evolving standards such as H.264 video encoding and 802.16-2004 wireless communication. Stretch has added its own technology to a core chip design that it licensed from Tensilica. The Stretch S5000 processor and the accompanying development tools move software hotspots (the sequences of operations that are executed many times) into exceptionally fast custom instructions. This transformation is easy to perform, yet tens to hundreds of instructions on other processors may become just a single instruction on the Stretch processor [22].

## CONCLUSION

This article describes how current applications—communications and mobile systems—have employed FPGAs because they are more flexible than ASICs yet with higher speed and lower power consumption than CPUs. This has happened in spite of the fact that we require HDL experts to program them. New applications that can benefit from variable-grain parallelism are hot prospects to emerge as killer applications in the near future, especially as improvements in data movement are made. Enabling these new killer applications can only be accomplished by increasing designer productivity. Graphical tools that provide reusable components and means of expressing parallelism hold great promise in achieving these goals.

## REFERENCES

[1] Altera Web site [Online]. Available: http://www.altera.com

[2] Xilinx Web site [Online]. Available: http://www.xilinx.com

[3] Merrill Lynch Web site [Online]. Available: http //www.fs.ml.com

[4] Microelectronic Systems Web site [Online]. Available:http://vlsi1.engr.utk.edu/ece/msn/

[5] System C Web site [Online]. Available: http://www.systemc.com

[6] "NEC structured ASICs" [Online]. Available: http://www.am.necel.com/asics/structuredasics.html

[7] D. Bouldin and R. Srivastava, "An open system-on-chip platform for education," in *Proc. 2004 European Workshop Microelectronics Education*, Apr. 2004, pp. 58–61.

[8] P. Dehkordi, T. Powell and D. Bouldin, "Development of a DSP/MCM subsystem: Assessing low-volume, low-cost MCM prototyping for universities," in *Proc. 1996 IEEE Multi-Chip Module Conf.*, Feb. 1996, pp. 89–94.

[9] DAFCA, Inc. Web site [Online]. Available: http://www.dafca.com

[10] "AFRL high-performance computer facilities" [Online]. http://www.if.afrl.af.mil/tech/facilities/HPC/hpcf.html

[11] O. Storaasli, "Engineering applications on NASA's FPGA-based hypercomputer," in *Proc. 2004 Military and Aerospace Applications of Programmable Devices and Technologies Int. Conf.*, Sep. 2004, pp. 161–168.

[12] "Floating-point for FPGAs" [Online]. Available: http://indus.usc.edu/fpus

[13] J. Lyke, "Recent progress in cutting edge packaging technology at AFRL," in *Proc. 1999 American Institute of Physics Conf.*, Jan. 1999, pp. 709–714.

[14] Information Sciences Institute Web site [Online]. Available: http://www.isi.edu

[15] Cray, Inc. Web site [Online]. Available: http://www.cray.com

[16] SRC Computers, Inc. Web site [Online]. Available: http://www.srccomputers.com

[17] C. Patterson, "High-performance DES encryption in Virtex FPGAs using Jbits," in *Proc. 2000 IEEE Symp. Field-Programmable Custom Computing Machines*, Apr. 2000, pp. 113–121.

[18] F. Li, Y. Lin, L. He, D. Chen, and J. Cong, "Power modeling and characteristics of field programmable gate arrays," *IEEE Trans. Computer-Aided Design*, vol. 24, no. 11, pp. 1712–1724, Nov. 2005.

[19] Simplicity Web site [Online]. Available: http://www.synplicity.com

[20] Starbridge Web site [Online]. Available: http://www.starbridgesystems.com

[21] HCS Research Laboratory Web site [Online]. http://www.hcs.ufl.edu

[22] Stretch Web site [Online]. Available: http://www.stretchinc.com

*Don Bouldin* is with the Electrical and Computer Engineering Department at the University of Tennessee. E-mail: dbouldin@tennessee.edu.

CD ■