Guest Editorial: TRETs Special Edition on the 15th International Symposium on FPGAs

2

Today's Field-Programmable Gate Arrays (FPGAs) employ modern 65 nm IC technologies and reach upwards of 250,000 logic elements, 10s of Mbps of on-chip memory and include high speed I/O such as serial transceivers running at speeds up to 6.5 Gbps. Modern FPGAs are used in a wide variety of applications including wireless and wireline infrastructure, video and image processing, prototyping and emulation, military, automotive and high-performance computing. With increasing ASIC NREs and growing complexities associated with performance and yield for deep-submicron technologies, more designs are finding their way into volume production on FPGA platforms. At the same time, FPGA architectures and design automation tools must tackle these DSM complexities in the underlying FPGA fabric and mitigate their negative effects. FPGAs have always found themselves in mixed environments interfacing with processors and ASICs, but as capacities grow the design options continue to increase, including opportunities to embed FPGA-like substrates in ASICs and opportunities to host processors on top of FPGAs.

The 15th annual ACM/SIGDA International Symposium on FPGAs (FPGA07) was held in Monterey, California in February 2007. Here, researchers shared the latest research results on FPGA architecture, circuit design, CAD and algorithms, methodologies, hardware algorithms and applications. "The FPGA Conference" remains the dominant forum bringing together all of these different aspects of FPGA design. In this inaugural edition of the ACM Transactions on Reconfigurable Technology and Systems, we are pleased to present expanded versions of five papers from the FPGA08 conference, touching on many of these different areas. Several other papers from the conference will follow in future issues of TRETs.

DSM technologies experience high variation in parameters. When variation places slower devices on critical paths, it reduces circuit performance. Since every chip sees a different set of resource delays, it is difficult to produce a single design configuration which is good for all chips. Matsumoto et al. explore the benefits of producing a set of diverse design configurations for each logical design in "Suppression of Intrinsic Delay Variation in FPGAs using Multiple Configurations." These multiple configurations route nets differently, placing critical route paths on different resource sets in each configuration. For a particular FPGA, they select the configuration which provides the best performance. This allows them to exploit the flexibility available in FPGA designs to

assign functions to different resources without demanding complete knowledge of the specific delays on a particular FPGA or demanding that backend CAD optimizations be run independently for each FPGA. For uncorrelated, random within-die variation, they estimate that a set of 10 configurations provides a 10% delay improvement at a 99% yield target for 30% V_{th} variation against nominal 90-nm process parameters.

In “Statistical Analysis and Process Variation-Aware Routing and Skew Assignment for FPGAs”, Sivaswamy and Bazargan apply modern statistical timing analysis to FPGA routing. They note that today’s typical FPGA designs do not see as strong a demand for statistical timing calculations as performance-driven ASICs. Nonetheless, they show that using statistical timing calculations in an FPGA router can increase the yield at fixed timing targets. They further show how the clock network can skew the arrival of clock signals to increase yield. For 65 nm technology, they show a combined delay improvement of 10% at a 99% yield target compared to optimizations based on a conventional, deterministic timing model.

The large capacity of modern FPGAs increases their realm of application. In “A Desktop Computer with a Reconfigurable Pentium,” Lu et al. demonstrate that it is now possible to implement a Pentium processor described in VHDL at the RTL level in half the resources of a single, large, 65 nm FPGA. The FPGA-hosted Pentium drops into a standard motherboard and can boot and run modern operating systems (e.g., Red Hat Linux, Microsoft Windows XP). While this is an older IA32 implementation, the design nonetheless highlights how far we have come from the days when it took hardware-emulation systems composed of 100s of FPGAs to emulate a commercial processor. It further illustrates how it is now viable to perform architectural what-if experiments in FPGA platforms and to run commercial-grade applications on top of the experimental architectures.

A perennial question for the FPGA conference is: “How do we improve FPGA architecture?” Related to that is the question of how we can get early guidance for promising directions to explore. In “Designing Efficient Input Interconnect Blocks for LUT Clusters Using Counting and Entropy,” Feng and Kaptanoglu explore the use of entropy as a high-level predictor of routability. Specifically, they explore the design of connections between the routing network and the logic block inputs in a logical cluster. While it is easy to count the switches in a proposed interconnect structure, without doing detailed placement and routing experiments, it is difficult to quantify how useful a particular arrangement of switches will be. They develop a new cluster interconnect switch topology based on guidance from their entropy metric and show that the entropy-based routability prediction is well correlated with routing requirements and route time.

The attractiveness of programmable fabrics is not limited to standalone commercial FPGAs. In “A Synthesizable Datapath-Oriented Embedded FPGA Fabric”, Wilton et al. consider the problem of efficiently embedding a programmable FPGA-like fabric into an ASIC design flow, enabling post-fabrication customization of the ASIC SoC design. They first identify and solve the issue of combinational cycles in an un-programmed FPGA fabric, then develop

a word-based datapath architecture whose pipeline is configured by communication between a control block and status bits. The control block itself is programmable to implement small amounts of fine-grained general purpose logic. For datapath-targeted applications such as an example debug controller illustrated in the paper they are able to achieve density comparable to general-purpose FPGA fabrics built on full-custom logic.

We hope that these exciting new research results and those in other papers from the 15th International Symposium on FPGAs that will appear in subsequent issues of TRETS illustrate the blossoming area of research in FPGAs and their applications.

—ANDRÉ DEHON AND MIKE HUTTON
Guest Editors