University of Pennsylvania
ScholarlyCommons

Departmental Papers (ESE)   Department of Electrical & Systems Engineering

February 2007

Special Issue on the 2006 International Symposium on Field-Programmable Gate Arrays

Kia Bazargan
University of Minnesota

André DeHon
University of Pennsylvania, andre@seas.upenn.edu

Follow this and additional works at: http://repository.upenn.edu/ese_papers

Special Issue on the 2006 International Symposium on Field-Programmable Gate Arrays

Guest Editorial

FIELD-PROGRAMMABLE gate arrays (FPGAs) continue to play growing roles in an increasing fraction of today's computing systems. As the cost of mask sets grow into the millions of dollars, and deep-submicron effects at 90 nm and below make ASIC yield optimization more challenging, more design starts turn to FPGAs. At the same time, the capacity and performance of modern FPGAs is sufficient for many applications. Even designs with moderately high volumes (e.g., 10K parts/ month) are turning to FPGAs for these advantages and their fast time-to-market as well as the opportunities FPGAs offer for rapid changes to fix bugs, add functionality, adapt to new standards, or provide new features.

Industrial and academic FPGA researchers and practitioners have gathered for the International Symposium on Field-Programmable Gate Arrays in Monterey, CA, in February since 1994 to discuss the latest advances and challenges in FPGA architecture, technology, CAD, and applications. Throughout the FPGA conference's growth from a workshop to a symposium, computed-aided design technology has played a key role in enabling FPGA technology and in the design and optimization of FPGA architectures. While papers on FPGA CAD and applications now appear in many conferences, the FPGA meeting remains the one place where the interplay of technology, CAD, architecture, and applications come together in one setting. In this Special Issue, we present expanded versions of nine CAD-oriented papers first presented at the fourteenth International Symposium on Field-Programmable Gate Arrays in 2006.

While aggressive technology scaling and improvements in CAD and architecture to exploit the technology have given us high-performance and large-capacity FPGAs which now suffice for many applications, the question remains: what are we paying for this late-bound configurability? Or how much better (e.g., smaller, faster) could our FPGAs possibly be? In "Measuring the Gap Between FPGAs and ASICs," Kuon and Rose undertake the challenging and controversial task of rigorously quantifying the area, delay, and power costs for FPGAs compared to ASIC designs. Using a large collection of designs and modern CAD flows, they show evidence for a factor of 18–35 area penalty, a factor of 3–4 delay penalty, and a factor of 7–14 dynamic power penalty for FPGAs implementing identical designs as ASICs.

There are various ways one might attempt to improve FPGAs, narrowing the gap with ASICs. In "Performance Benefits of Monolithically Stacked 3-D FPGA," Lin et al. explored the benefits a 3-D device stacking fabrication technology could provide for RAM-based FPGAs. They consider a technology in which configuration bits and pass-transistor switch devices can be moved to active device layers above the logic and routing resources layer. Even though devices within an FPGA tile are

stacked into a 3-D structure, the underlying tile architecture is still 2-D in their proposed design. Stacking devices results in smaller footprint area and, hence, shorter wires, which in turn brings about smaller delays and power consumptions.

In comparisons, such as the one by Kuon and Rose, and in FPGA architecture design in general, a common question is whether or not the CAD mappings are fully exploiting the architectures under study. In "Optimality Study of Logic Synthesis for LUT-Based FPGAs," Cong and Minkovich use synthetic designs with known optimal logic mappings and coverings to illustrate the size of the potential gap between today's state-of-the-art CAD techniques and optimum solutions. In particular, they show that although leading FPGA technology mapping algorithms can produce close to optimal solutions, the results from the entire logic synthesis flow (logic optimization and mapping) are far from optimal. Using such synthetic benchmarks can help CAD developers pinpoint weaknesses in their logic synthesis algorithms.

As Cong and Minkovich illustrate, there is a continuing need to improve FPGA technology mapping. In "Improvements to Technology Mapping for LUT-Based FPGAs," Mishchenko et al. introduce the notion of factor cuts that help reduce the number of cuts that need to be explicitly stored while technology mapping a circuit. Furthermore, they use pruning techniques to reduce enumeration time and memory requirements when exhaustively searching cuts of small sub-graphs. Area minimization heuristics are presented that reduce area by 6% compared to previous work while maintaining the same critical path delay. Finally, a solution to the structural bias is proposed that maintains intermediate circuits derived during the logic optimization phase, because some intermediate circuit might yield better technology mapping solutions. Extending the mapper to use structural choices reduces delay on average by 6% and area by 12%, compared to the previous work, while increasing run-time 1.6 times.

FPGAs have long since crossed the point where they were only used for random glue and control logic. Consequently, FPGA designs demand higher-level building blocks and CAD techniques to generate and optimize the large-scale programmable systems they can now support. In "FPGA Pipeline Synthesis Design Exploration Using Module Selection and Resource Sharing," Sun et al. describe an FPGA-oriented high-level synthesis flow that exploits both time-multiplexed resource sharing and module selection. They provide algorithms for the joint optimization and identify portions of the design space where the new algorithms yield results which are 2–3 times smaller than performing either optimization alone.

Modern FPGA capacity can support tens of soft-core processors on a single die. At the same time, application-customized processors are emerging as one stylistic approach for organizing computations for a specific application or domain of applica-

tions. In "Exploration and Customization of FPGA-Based Soft Processors," Yiannacouras *et al.* describe the SPREE system for synthesizing parameterized soft-processors for FPGAs. They explore processor pipelining and ISA subsetting and show a 25% area-time reduction for specialized designs.

As part of the growth to large-capacity programmable system devices, today's FPGAs incorporate large embedded memory blocks. CAD techniques can play a role in optimizing the power consumption of these blocks as shown in "Power-Efficient RAM Mapping Algorithms for FPGA Embedded Memory Blocks," by Tessier *et al.* The authors present a set of power-efficient logical-to-physical RAM mapping algorithms that convert user-defined memory specifications to on-chip FPGA memory block resources. The objective is to minimize RAM dynamic power. Using their method, embedded memory dynamic power can be reduced by 26% on average and overall core dynamic power can be reduced by 6% with a minimal loss (1%) in design performance.

Specialized reconfigurable devices fall in the continuum between pure ASICs and FPGAs. As such, they offer another approach to narrowing the gap, or at least, understanding what lies in the gap. In "Automatic Creation of Domain-Specific Reconfigurable CPLDs for SoC," Holland and Hauck propose a flow that automatically generates CPLD architectures with crossbar connectivity. The architecture is tailor-made to application needs. They further improve the generated architecture by sparsifying the crossbar switches, which results in significant improvements in the area and scalability of the domain-specific CPLDs.

The Kuon and Rose paper pointed out a 7–14 times gap in power between ASICs and FPGAs; as a result, modern demand for ultra-low power is one of the reasons designers may be forced to continue to use ASICs instead of FPGAs. In "A 90-nm Low-Power FPGA for Battery-Powered Applications," Tuan *et al.* show how a baseline Xilinx Spartan-3 FPGA is improved using several power optimization techniques. The resulting architecture consumes 46% less active power and 99% less standby power. Furthermore, the chip is able to wake up from standby mode in approximately 100 ns.

KIA BAZARGAN, *Guest Editor*
Department of Electrical and Computer
  Engineering
University of Minnesota
Minneapolis, MN 55455 USA

ANDRÉ DEHON, *Guest Editor*
Department of Electrical and System
  Engineering
University of Pennsylvania
Philadelphia, PA 19104 USA

Kia Bazargan (S'97–M'00) received the Bachelors degree in computer science from Sharif University, Tehran, Iran, and the M.S. and Ph.D. degrees in electrical and computer engineering from Northwestern University, Evanston, IL, in 1998 and 2000, respectively.

He is currently an Associate Professor in the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis. His research interests are computer-aided design, FPGAs, and reconfigurable computing.

Dr. Bazargan has served on the technical program committees of a number of IEEE-sponsored conferences (e.g., FPGA, ISPD, ICCAD, ASPDAC). He was a Guest Co-Editor of the *ACM Transactions on Embedded Computing Systems* (ACM TECS) Special Issue on Dynamically Adaptable Embedded Systems in 2003. He is an Associate Editor of the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS. He was the recipient of an NSF CAREER Award in 2004.

André DeHon (S'92–M'96) received the S.B., S.M., and Ph.D. degrees in electrical engineering and computer science from Massachusetts Institute of Technology, Cambridge, in 1990, 1993, and 1996 respectively.

From 1996 to 1999, he co-ran the BRASS group in the Computer Science Department at the University of California, Berkeley. From 1999 to 2006, he was an Assistant Professor of Computer Science at California Institute of Technology, Pasadena. Since 2006, he has been an Associate Professor of Electrical and Systems Engineering at the University of Pennsylvania, Philadelphia. He is broadly interested in how we physically implement computations from substrates, including VLSI and molecular electronics, up through architecture, CAD, and programming models. He places special emphasis on spatial programmable architectures (e.g., FPGAs) and interconnect design and optimization.