FPGA '95

1995 ACM Third International Symposium on Field-Programmable Gate Arrays

February 12-14, 1995
Monterey Marriott
Monterey, California USA

FPGA '95 is sponsored by the ACM Special Interest Group on Design Automation (SIGDA) with support from Actel, Altera and Xilinx.
1995 ACM Third International Symposium on Field-Programmable Gate Arrays

February 12-14, 1995
Marriott Hotel, Monterey, CA, USA

Sponsored by ACM SIGDA.
The organizers gratefully acknowledge support from Actel Corp., Altera Corp. and Xilinx, Inc.

Field-Programmable Gate Arrays are revolutionizing the business of ASIC design by providing fast turnaround and negligible non-recurring engineering costs. The challenge in FPGA research is to improve their speed and density through architectural and process innovation, as well as finding new CAD synthesis and testing algorithms that can make effective use of the new architectures. The objective of this symposium is to bring together people who are working in the many areas of research that are necessary to make a complete FPGA, and a Field-Programmable System. We hope that you find these proceedings interesting and exciting.
Organizing and Program Committee

**General Chair:** Pak K. Chan, UC Santa Cruz

**Program Chair:** Jonathan Rose, University of Toronto

**Program Committee**

Duncan Buell, SRC  
Carl Ebeling, U. Washington  
Dwight Hill, Synopsys  
Jonathan Rose, U. Toronto  
Takayasu Sakurai, Toshiba  
Steve Trimberger, Xilinx  

Pak K. Chan, UCSC  
Bob Condon, Exemplar  
Sinan Kaptanoglu, Actel  
Richard Rudell, Synopsys  
Martine Schlag, UCSC  

Jason Cong, UCLA  
Frederic Furtek, Atmel  
John McCollum, Actel  
Rob Rutenbar, CMU  
Tim Southgate, Altera

**Acknowledgements**

The organizers would like to thank Colleen Matteis, Debbie Hall and Lisette Burgos from ACM for their grand organizational talents and work in putting the workshop together. Jennifer Rodrigues did ALL of the correspondence for prospective authors and program committees as well as putting together these final notes. Thanks!
Panel: The Architecture/Software Boundary: Motherhood and Lies

The most crucial element in the creation of an FPGA is the interaction between the device architecture and the software tools that map circuits into the device. The architect should determine the ability of the placement and routing software, for example to:

i. Deal with routing architecture’s - symmetry, hierarchy, segmentation, etc.
ii. Handle special purpose connections- carry chains, local interconnect or hard-wired connections.
iii. Deal intelligently the fixed amount of interconnect - should different-sized parts be given different quantities of routing?

Similarly the architecture should be able to deal with the capability of the logic synthesis tools to handle:

i. The structure and function of the logic block.
ii. Special logic block features such as adder logic, clock qualifiers and logic sharing capability.
iii. The effect of the synthesis on the routability of the synthesized netlist.

Although FPGA vendor and academic architects will immediately agree with that this interaction is essential, and is indeed a motherhood issue, it is rare that these interactions are enforced. Similarly, synthesis vendors (and University CAD researchers) may claim to produce FPGA architecture-specific algorithms but the reality is otherwise. What makes it so difficult?

Is interaction really important, or will the effects of poor interaction be swallowed by the next generation IC process advance? Perhaps some very clever interactions can produce major density and speed gains FPGA devices. If interaction is difficult to enforce for general-purpose FPGA architectures, will it be possible to create a next generation of special-purpose architectures? The following people explored these issues:

1 Jonathan Rose, University of Toronto
   Introduced the subject, asked a few penetrating questions and expressed the hope that more sophisticated interaction would provided major wins in the future.

2 Tim Southgate, Director of Software Engineering, Altera
   Described Altera’s architecture exploration process by giving a fascinating time-line history of the development on the Flex 8000 architecture, which began in November 1990 and shipped silicon roughly two years later. He showed how simultaneously development and use of the “compiler” (mapping, placement and routing software) was used to ensure usability of the architecture, and how results from experiments often forced important changes in the first device.

3 Steve Kelem, Software System Architect, Xilinx.
   Described how the XBLOX synthesis/placement/routing system worked and how it directly leverages the architectural features of the FPGA.