FPGA Design Automation: A Survey

FPGA Design Automation: A Survey

Deming Chen

Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign
dchen@uiuc.edu

Jason Cong

Department of Computer Science
University of California at Los Angeles
cong@cs.ucla.edu

Peichen Pan

Magma Design Automation, Inc.
Los Angeles, CA
peichen@magma-da.com

now
the essence of knowledge
Boston – Delft

Foundations and Trends® in Electronic Design Automation

Published, sold and distributed by:
now Publishers Inc.
PO Box 1024
Hanover, MA 02339
USA
Tel. +1-781-985-4510
www.nowpublishers.com
sales@nowpublishers.com

Outside North America:
now Publishers Inc.
PO Box 179
2600 AD Delft
The Netherlands
Tel. +31-6-51115274

A Cataloging-in-Publication record is available from the Library of Congress

The preferred citation for this publication is Deming Chen, Jason Cong and Peichen Pan, FPGA Design Automation: A Survey, Foundations and Trends® in Electronic Design Automation, vol 1, no 3, pp 195–330, November 2006

Printed on acid-free paper

ISBN: 1-933019-38-7

Foundations and Trends® in Electronic Design Automation
Volume 1 Issue 3, November 2006

Editorial Board

Editor-in-Chief:
Sharad Malik
Department of Electrical Engineering
Princeton University
Princeton, NJ 08544
USA
sharad@princeton.edu

Editorial Scope

Foundations and Trends® in Electronic Design Automation will publish survey and tutorial articles in the following topics:

- System Level Design
- Behavioral Synthesis
- Logic Design
- Verification
- Test
- Physical Design
- Circuit Level Design
- Reconfigurable Systems
- Analog Design

Information for Librarians
Foundations and Trends® in Electronic Design Automation, November 2006, Volume 1, 4 issues. ISSN paper version 1551-3939. ISSN online version 1551-3947. Also available as a combined paper and online subscription.

Foundations and Trends® in
Electronic Design Automation
Vol. 1, No 3 (November 2006) 195–330

DOI: 10.1561/1000000003

FPGA Design Automation: A Survey

Deming Chen[1], Jason Cong[2] and Peichen Pan[3]

[1] *Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, dchen@uiuc.edu*
[2] *Department of Computer Science, University of California at Los Angeles, cong@cs.ucla.edu*
[3] *Magma Design Automation, Inc., Los Angeles, CA, peichen@magma-da.com*

Abstract

Design automation or computer-aided design (CAD) for field programmable gate arrays (FPGAs) has played a critical role in the rapid advancement and adoption of FPGA technology over the past two decades. The purpose of this paper is to meet the demand for an up-to-date comprehensive survey/tutorial for FPGA design automation, with an emphasis on the recent developments within the past 5–10 years. The paper focuses on the theory and techniques that have been, or most likely will be, reduced to practice. It covers all major steps in FPGA design flow which includes: routing and placement, circuit clustering, technology mapping and architecture-specific optimization, physical synthesis, RT-level and behavior-level synthesis, and power optimization. We hope that this paper can be used both as a guide for beginners who are embarking on research in this relatively young yet exciting area, and a useful reference for established researchers in this field.

Keywords: computer-aided design; FPGA design

Contents

1

Introduction

The semiconductor industry has showcased the spectacular exponential growth of device complexity and performance for four decades, predicted by Moore's Law. Programmable logic devices (PLDs), especially field programmable gate arrays (FPGAs), have also experienced an exponential growth in the past 20 years, in fact, at an even faster pace compared to the rest of the semiconductor industry. For example, when FPGAs were first debuted in the mid- to late-80s, the Xilinx XC2064 FPGA had only 64 lookup tables (LUTs) and it was used as simple glue logic. Now, both Altera's Stratix II [10] and Xilinx's Virtex-4 chips [205] offer up to over 200,000 programmable logic cells (i.e., LUTs), plus a large number of hard-wired macro blocks such as embedded memories, DSP blocks, embedded processors, high-speed IOs, and clock synchronization circuitry, representing an over 3,000 times increase in logic capacity. These FPGA devices are being used to implement highly complex system-on-a-chip (SoC) designs. To support the design of such complex programmable devices, computer-aided design (CAD) plays a critical role in delivering high-performance, high-density, and low-power design solutions using these high-end FPGAs. We witnessed the

establishment of FPGA design automation as a research area and a dramatic increase in research activities in this area in the past 17–18 years. However, there is lack of comprehensive references for the latest FPGA CAD research. Most existing books (e.g., [23, 27, 92, 150, 186]) and survey/tutorial papers (e.g., [28, 52]) in this area are 10–15 years old, and do not reflect vast amount of recent research on FPGA CAD. The purpose of this paper is to meet the demand for a comprehensive survey/tutorial on the state of FPGA CAD—with an emphasis on the recent developments that have taken place within the past 5–10 years and a focus on the theory and techniques that have been, or most likely will be, reduced to practice. We hope that this paper can be useful for both beginners and established researchers in this exciting and dynamic field.

In the remainder of this section we shall first briefly introduce some typical FPGA architectures and define the basic terminologies that will be used in the rest of this paper. Then, we shall provide an overview of the FPGA design flow.

1.1 Introduction to FPGA Architectures

An FPGA chip includes input/output (I/O) blocks and the core programmable fabric. The I/O blocks are located around the periphery of the chip, providing programmable I/O connections and support for various I/O standards. The core programmable fabric consists of programmable logic blocks and programmable routing architectures. Figure 1.1 shows a high-level view of an island-style FPGA [23], which represents a popular architecture framework that many commercial FPGAs are based on, and is also a widely accepted architecture model used in the FPGA research community. *Logic blocks* represented by gray squares consist of circuitry for implementing logic. Logic blocks are also called configurable logic blocks (CLBs). Each logic block is surrounded by routing channels connected through switch blocks and connection blocks. The wires in the channels are typically segmented and the length of each wire segment can vary. A *switch block* connects wires in adjacent channels through programmable switches such as pass-transistors or bi-directional buffers. A *connection block* connects

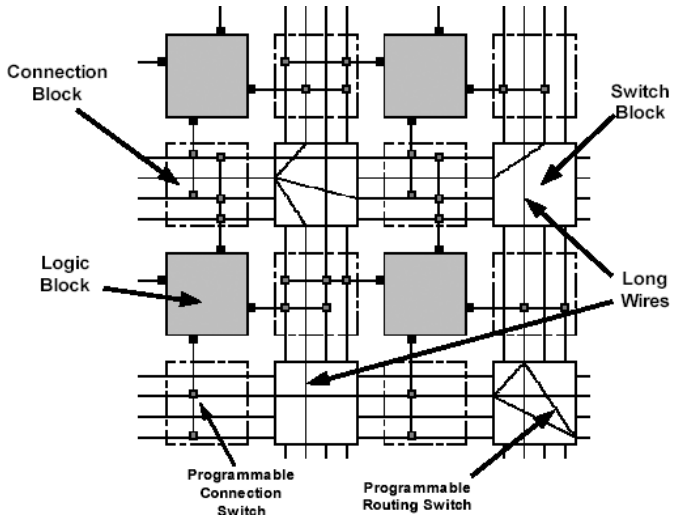

Fig. 1.1 An island-style FPGA [23].

the wire segments around a logic block to its inputs and outputs, also through programmable switches. Notice that the structures of the switch blocks are all identical. The figure illustrates the different switching and connecting situations in the switch blocks (the structures of all the connection blocks are identical as well). In [23] routing architectures are defined by the parameters of channel width (W), switch block flexibility (F_s – the number of wires to which each incoming wire can connect in a switch block), connection block flexibility (F_c – the number of wires in each channel to which a logic block input or output pin can connect), and segmented wire lengths (the number of logic blocks a wire segment spans). Modern FPGAs also provide embedded IP cores, such as memories, DSP blocks, and processors, to facilitate the implementation of SoC designs.

Commercial FPGA chips contain a large amount of dedicated interconnects with different fixed lengths. These interconnects are usually point-to-point and uni-directional connections for performance improvement. For example, Altera's Stratix II chip [10] has vertical or horizontal interconnects across 4, 16 or 24 logic blocks. There are dedicated carry chain and register chain interconnects within and between

logic blocks as well. Xilinx's Spartan-3E chip [204] has long lines, hex lines, double lines, and direct connections between the logic blocks. These lines cross different numbers of logic blocks. Specifically, the direct connect lines can route signals to neighboring tiles vertically, horizontally, and diagonally. For example, Figure 1.2 shows the direct connect lines (a) and hex lines (b) between a CLB and its neighbors in the Spartan-3E chip. The use of segmented routing makes the FPGA interconnect delays highly non-linear, discrete, and in some cases, even non-monotone (with respect to the distance). This presents unique challenges for FPGA placement and routing tools because a simple interconnect delay model using Manhattan distance between the source and the sink may not work well any more. Accurate interconnect delay modeling is a mandate for meaningful performance-driven physical design tools for FPGAs.

Further down the logic hierarchy, each logic block contains a group of basic logic elements (BLEs), where each BLE contains a

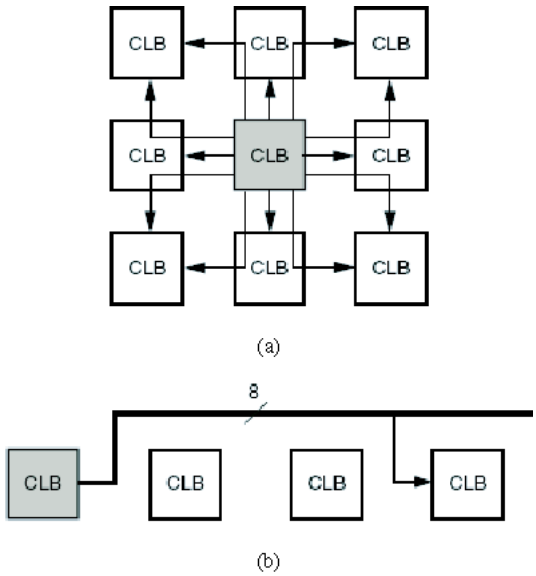

Fig. 1.2 Direct connect lines (a) and hex lines (b) in Xilinx Spartan-3E architecture [204].

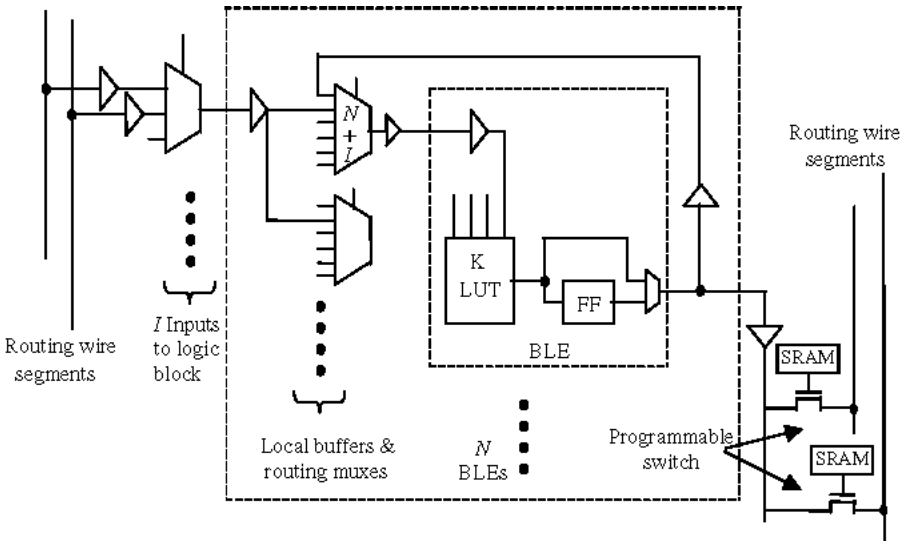

Fig. 1.3 A logic block and its peripheries.

LUT[1] and a register. Figure 1.3 shows part of a logic block with a block size N (the logic block contains N BLEs). The logic block has I inputs and N outputs. These inputs and outputs are fully connected to the inputs of each LUT through multiplexers. The figure also shows some details of the peripheral circuitry in the routing channels.

In addition to logic and routing architectures, clock distribution networks is another important aspect of FPGA chips. An H-tree based FPGA clock network is shown in Fig. 1.4 [130]. A tile is a logic block. Each clock tree buffer in the H-tree has two branches. There is a local clock buffer for each flip-flop in a tile. Both clock tree buffers in the H-tree and local clock buffers in the tiles are considered to be clock network resources. Chip area, tile size, and channel width determine the depth of the clock tree and the lengths of the tree branches.

[1] We focus on the LUT-based FPGA architecture in which the BLE consists of one k-input lookup table (k-LUT) and one flip-flop. The output of the k-LUT can be either registered or un-registered. We want to point out that commercial FPGAs may use slightly different logic architectures. For example, Altera's Stratix II FPGA [10] uses an adaptive logic module which contains a group of LUTs and a pair of flip-flops.

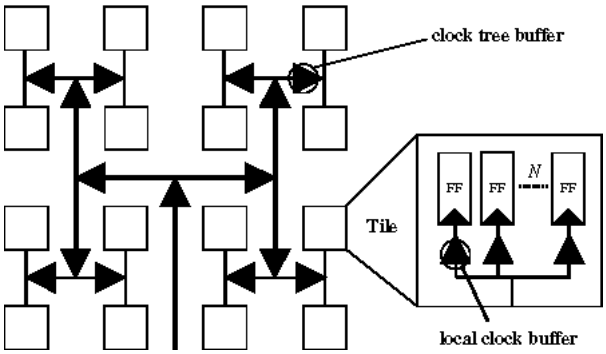

Fig. 1.4 A clock tree [130].

There is another type of programmable logic device called complex programmable logic device (CPLD). The general architecture topology of a CPLD chip is similar to that of island-based FPGAs, where routing resources surround logic blocks. One attribute of CPLD is that its interconnected structures are simpler than those of FPGAs. Therefore, the interconnect delay of CPLD is more predictable compared to that of FPGAs. The basic logic elements in the CPLD logic blocks are not LUTs. Instead, they are logic cells based on two-level AND-OR structures, where a fixed number of AND gates (also called p-terms) drive an OR gate. The output from the OR gate can be registered as well. For example, Fig. 1.5 shows such a structure (called *macrocell*) used in Altera's MAX7000B CPLD [6]. Each macrocell has five p-terms by default. It can borrow some p-terms from its neighbors. The interconnect structure PIA (programmable interconnect array) connects different logic blocks together.

1.2 Overview of FPGA Design Flow

As the FPGA architecture evolves and its complexity increases, CAD software has become more mature as well. Today, most FPGA vendors provide a fairly complete set of design tools that allows automatic synthesis and compilation from design specifications in hardware specification languages, such as Verilog or VHDL, all the way down

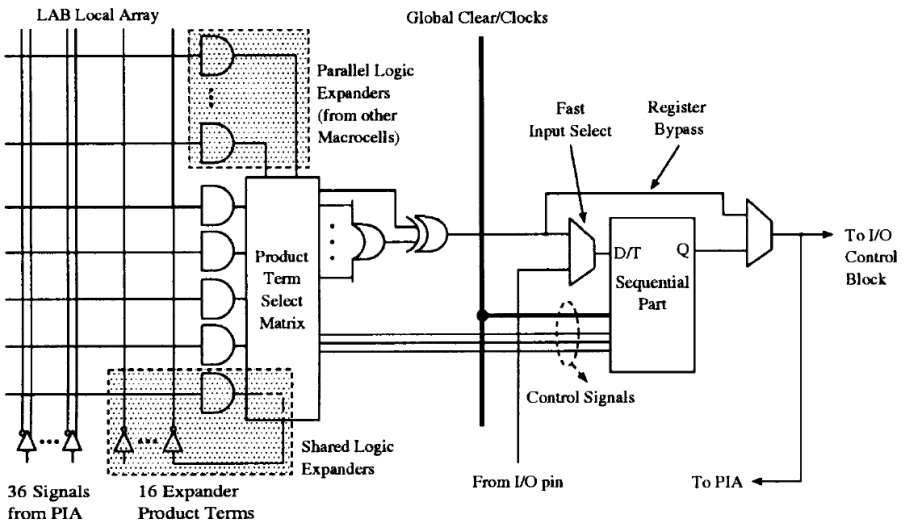

Fig. 1.5 An example of CPLD logic element, MAX 7000B macrocell [6].

to a bit-stream to program FPGA chips. A typical FPGA design flow includes the steps and components shown in Fig. 1.6.

Inputs to the design flow typically include the HDL specification of the design, design constraints, and specification of target FPGA devices. We further elaborate on these components of the design input in the following:

- The most widely used design specification languages are Verilog and VHDL at the register transfer level (RTL) which specify the operations at each clock cycle. There is a general (although rather slow) trend toward moving to specification at a higher level of abstraction, using general-purpose behavior description languages like C or SystemC [180], or domain-specific languages, such as MatLab [183] or Simulink [183]. Using these languages, one can specify the behavior of the design without going through a cycle-accurate detailed description of the design. A behavior synthesis tool is used to generate the RTL specification in Verilog or VHDL, which is then fed into the design flow as shown in Fig. 1.6. We shall discuss the behavior synthesis techniques in Section 5.

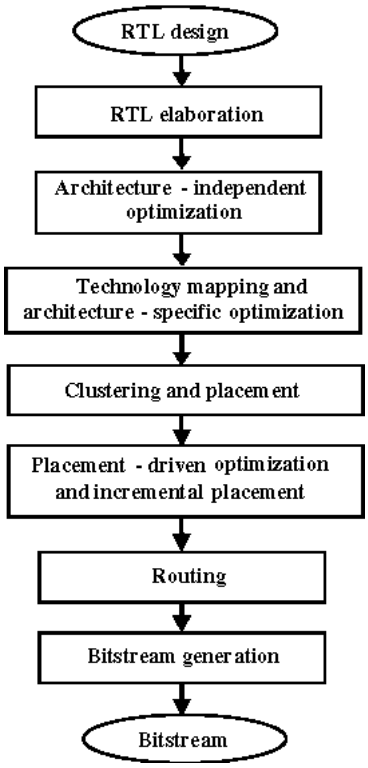

Fig. 1.6 A typical FPGA design flow starting from RTL specifications.

- Design constraints typically include the expected operating frequencies of different clocks, the delay bounds of the signal path delays from input pads to output pads (I/O delay), from the input pads to registers (setup time), and from registers to output pads (clock-to-output delay). In some cases, delays between some specific pairs of registers may be constrained. Design constraints may also include specifications of so-called *false paths* and *multi-cycle paths*. False paths will not be activated during normal circuit operation, and therefore can be ignored; multi-cycle paths refer to signal paths that carry a valid signal every few clock cycles, and therefore have a relaxed timing requirement. Typically, the designer

specifies the false paths and multi-cycle paths based on his knowledge of the design, although recently attempts have been made to automatically extract these paths from the RTL designs [86]. Finally, the design constraints may include physical location constraints, which specify that certain logic elements or blocks be placed at certain locations or a range of locations. These location constraints may be specified by the designer, or inherited from the previous design iteration (for making incremental changes), or generated automatically by the physical synthesis tools in the previous design iterations. We shall discuss the physical synthesis concept and techniques in Section 4.

- The third design input component is the choice of FPGA device. Each FPGA vendor typically provides a wide range of FPGA devices, with different performance, cost, and power tradeoffs. The selection of target device may be an iterative process. The designer may start with a small (low capacity) device with a nominal speed-grade. But, if synthesis effort fails to map the design into the target device, the designer has to upgrade to a high-capacity device. Similarly, if the synthesis result fails to meet the operating frequency, he has to upgrade to a device with higher speed-grade. In both the cases, the cost of the FPGA device will increase—in some cases by 50% or even by 100%. This clearly underscores the need to have better synthesis tools, since their quality directly impacts the performance and cost of FPGA designs.

We now briefly describe each step in the design flow shown in Fig. 1.6 and, following that, we present an outline of the remainder of this paper. Given an RTL design, a set of design constraints, and the target FPGA device, the overall FPGA synthesis process goes through the following steps:

- RTL elaboration. This identifies and/or infers *datapath* operations, such as additions, multiplications, register files, and/or memory blocks, and *control logic*, which is elaborated into a set of finite-state machines and/or generic Boolean

networks. It is important to recognize the datapath elements as most of them have special architectural support in modern FPGAs, such as adders with dedicated fast-carry chains and embedded multipliers.

- Architecture-independent optimization. This includes both *datapath optimization*, using techniques such as constant propagation, strength reduction, operation sharing, and expression optimization; and *control logic optimization*, which includes both sequential optimization, such as finite-state machine encoding/minimization and retiming, and combinational logic optimization, such as constant propagation, redundancy removal, logic network restructuring and optimization, and don't-care based optimization.
- Technology mapping and architecture-specific optimization. This maps: (i) the optimized datapath to on-chip dedicated circuit structures, such as on-chip multipliers, adders with dedicated carry-chains, and embedded memory blocks for datapath implementation, and (ii) the optimized control logic to BLEs. Note that datapath operations can be mapped to BLEs as well if the dedicated circuit structures are not available or not convenient to use.
- Clustering and placement. Placement determines the location of each element in the mapped netlist. Since most modern FPGAs are hierarchical, a separate clustering step may be performed prior to placement to group BLEs into logic blocks. Alternatively, such clustering or grouping may be carried out during the placement process.
- Placement-driven optimization and incremental placement. Once placement is available, interconnects are defined and may become a performance bottleneck (since the delay of a long interconnect can be multiples of a BLE's delay). Further optimization may be carried out in the presence of interconnect delays, including logic restructuring, duplication, rewiring, etc. After such operations, an incremental placement step is needed to legalize the placement again.

The step of placement-driven optimization is optional, but may improve design performance considerably.

- Routing. Global routing and detail routing will be performed to connect all signal paths using the available programmable interconnects on-chip.
- Bit-stream generation. This is the final step of the design flow. It takes the mapped, placed, and routed design as input and generates the necessary bit-stream to program the logic and interconnects to implement the intended logic design and layout on the target FPGA device.

Following sections present the algorithms and techniques used in these steps in reverse order of the design flow. We start with routing and placement (Section 2), then present techniques used in technology mapping and architecture-specific optimization (Section 3). The architecture-dependent optimization phase of FPGA design typically shares techniques widely used for ASIC synthesis and optimization, and we refer the reader to the available textbooks [79, 98] for details. Section 4 presents the techniques used in physical synthesis of FPGA designs, which cover the algorithms used in clustering and placement-driven optimization. Section 5 presents the techniques used in RT-level and behavior-level synthesis for FPGA designs. Section 6 discusses synthesis techniques used for FPGA power optimization, which is a design objective that has received a lot of interest in recent years. This design objective cuts cross all design steps in the flow and interacts with performance and area optimization. Finally, we conclude this paper and touch on future trends of FPGA design automation in Section 7.

References

[1] International Technology Roadmap for Semiconductors, Executive Summary, 2003. http://public.itrs.net/ Files/2003ITRS/Home2003.htm.

[2] E. Ahmed and J. Rose. The effect of LUT and cluster size on deep-submicron FPGA performance and density. In *ACM International Symposium on FPGA*, February 2000.

[3] C. Albrecht. Provably good global routing by a new approximation algorithm for multicommodity flow. In *Proc. International Symposium on Physical Design*, pages 19–25, March 2000.

[4] M. J. Alexander, J. P. Cohoon, J. L. Ganley, and G. Robins. Placement and routing for performance-oriented FPGA layout. *VLSI Design: An International Journal of Custom-Chip Design, Simulation, and Testing*, 7(1), 1998.

[5] M. J. Alexander and G. Robins. New performance-driven FPGA routing algorithms. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 15(12):1505–1517, December 1996.

[6] Altera. MAX 7000B Data Sheet. http://www.altera.com/literature/ds/m7000b.pdf.

[7] Altera. PowerPlay Early Power Estimator. http://www.altera.com/support/devices/estimator/pow-powerplay.html.

[8] Altera. PowerPlay Power Analyzer. http://www.altera.com/support/devices/estimator/pow-powerplay.html.

[9] Altera. Stratix II 90-nm Silicon Power Optimization. http://www.altera.com/products/devices/stratix2/features/st2-90nmpower.html.

[10] Altera. Stratix II Device Handbook. http://www.altera.com/literature/hb/stx2/stratix2_handbook.pdf.

[11] Altera. White paper, "Stratix II vs. Virtex-4 Power Comparison & Estimation Accuracy White Paper. http://altera.com/literature/wp/wp_s2v4_pwr_acc.pdf.
[12] Altera, August 2002. Stratix Programmable Logic Device Family Data Sheet.
[13] J. Anderson and F. N. Najm. Power-aware technology mapping for LUT-based FPGAs. In *IEEE International Conference on Field-Programmable Technology*, 2002.
[14] J. Anderson and F. N. Najm. Interconnect capacitance estimation for FPGAs. In *IEEE/ACM Asia and South Pacific Design Automation Conference*, Yokohama, Japan, 2004.
[15] J. H. Anderson and S. D. Brown. Technology mapping for large complex PLDs. In *Design Automation Conf.*, 1998.
[16] J. H. Anderson, F. N. Najm, and T. Tuan. Active leakage power optimization for FPGAs. *International Symposium on Field Programmable Gate Arrays*, February 2004.
[17] F. Assaderaghi, D. Sinitsky, S. A. Parke, J. Bokor, P. K. Ko, and C. Hu. Dynamic threshold-voltage MOSFET (DTMOS) for ultra-low voltage VLSI. *IEEE Transactions Electron Devices*, 44:414–422, March 1997.
[18] B. Awerbuch, A. Bar Noy, N. Linial, and D. Peleg. Improved routing strategies with succinct tables. *J. Algorithms*, 11(3):307–341, 1990.
[19] P. Banerjee. *Parallel Algorithms for VLSI Computer-Aided Design*. Prentice-Hall, Inc., Englewoods-Cliffs, NJ, 1994.
[20] G. Beraudo and J. Lillis. Timing optimization of FPGA placements by logic replication. In *ACM/IEEE Design Automation Conference*, pages 96–201, 2003.
[21] V. Betz and J. Rose. Cluster-based logic blocks for FPGAs: area-efficiency vs. input sharing and size. In *IEEE Custom Integrated Circuits Conference*, pages 551–554, 1997.
[22] V. Betz and J. Rose. VPR: a new packing, placement and routing tool for FPGA research. In *International Workshop on Field-Programmable Logic and Applications*, pages 213–222, 1997.
[23] V. Betz, J. Rose, and A. Marquardt. *Architecture and CAD for Deep-Submicron FPGAs*. Kluwer Academic Publishers, 1999.
[24] N. Bhat and D. D. Hill. Routable technology mapping for LUT FPGAs. In *IEEE International Conference on Computer Design*, pages 95–98, 1992.
[25] E. Bozorgzadeh, S. Ogrenci, and M. Sarrafzadeh. Routability-driven packing for cluster-based FPGAs. In *Asia South Pacific Design Automation Conf.*, 2001.
[26] R. K. Brayton. Understanding SPFDs: A new method for specifying flexibility. In *International Workshop on Logic Synthesis*, 1997.
[27] S. Brown, R. Francis, J. Rose, and Z. Vranesic. *Field-Programmable Gate Arrays*. Kluwer Academic Publishers, May 1992.
[28] S. Brown and J. Rose. FPGA and CPLD architectures: A tutorial. *IEEE Design and Test of Computers*, 12(2):42–57, 1996.

[29] S. Brown, J. Rose, and Z. G. Vranesic. A detailed router for field-programmable gate arrays. *IEEE Trans. on Computer-Aided Design*, 11(5):620–628, May 1992.

[30] T. J. Callahan, P. Chong, A. DeHon, and J. Wawrzynek. Fast module mapping and placement for datapaths in FPGAs. In *International Symposium on Field Programmable Gate Arrays*, 1998.

[31] T. Chan, J. Cong, and K. Sze. Multilevel generalized force-directed method for circuit placement. In *Proceedings of the Int'l Symposium on Physical Design*. San Francisco, CA, April 2005.

[32] H. Chang and S. Sapatnekar. Impact of process variations on power: full-chip analysis of leakage power under process variations, including spatial correlations. In *Proc. of Design Automation Conf.*, June 2005.

[33] S. C. Chang, K. T. Cheng, N.-S. Woo, and M. Marek Sadowska. Postlayout rewiring using alternative wires. *IEEE Trans. on Computer Aided Design of Integrated Circuits and Systems*, 16(6):587–96, June 1997.

[34] S. C. Chang, L. V. Ginneken, and M. Marek-Sadowska. Circuit optimization by rewiring. *IEEE Transaction on Computers*, 48(9):962–970, September 1999.

[35] Y. W. Chang and Y. T. Chang. An architecture-driven metric for simultaneous placement and global routing for FPGAs. In *Proc. Design Automation Conf.*, pages 567–572, 2000.

[36] Y. W. Chang, K. Zhu, and D. F. Wong. Timing-driven routing for symmetrical array-based FPGAs. *ACM Trans. on Design Automation of Electronic Systems*, 5(3), July 2000.

[37] C. Chen, Y. Tsay, Y. Hwang, T. Wu, and Y. Lin. Combining technology mapping and placement for delay-optimization in FPGA designs. In *Int'l Conf. Computer Aided Design*, 1993.

[38] D. Chen and J. Cong. DAOmap: a depth-optimal area optimization mapping algorithm for FPGA designs. In *Int'l Conf. Computer Aided Design*, 2004.

[39] D. Chen and J. Cong. Delay optimal low-power circuit clustering for FPGAs with dual supply voltages. *International Symposium on Low Power Electronics and Design*, August 2004.

[40] D. Chen, J. Cong, M. Ercegovac, and Z. Huang. Performance-driven mapping for CPLD architectures. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 22(10):1424–1431, October 2003.

[41] D. Chen, J. Cong, and Y. Fan. Low-power high-level synthesis for FPGA architectures. *International Symposium on Low Power Electronics and Design*, August 2003.

[42] D. Chen, J. Cong, F. Li, and L. He. Low-power technology mapping for FPGA architectures with dual supply voltages. *International Symposium on Field Programmable Gate Arrays*, February 2004.

[43] G. Chen and J. Cong. Simultaneous logic decomposition with technology mapping in FPGA designs. *International Symposium on Field-Programmable Gate-Arrays*, 2001.

[44] G. Chen and J. Cong. Simultaneous timing driven clustering and placement for FPGAs. In *International Conference on Field Programmable Logic and Its Applications*, pages 158–167, August 2004.

[45] G. Chen and J. Cong. Simultaneous timing-driven placement and duplication. *International Symposium on Field-Programmable Gate-Arrays*, 2005.

[46] K. C. Chen, et al. DAG-map: graph-based FPGA technology mapping for delay optimization. *IEEE Design and Test of Computers*, 9(3):7–20, September 1992.

[47] A. Chowdary and J. P. Hayes. Technology mapping for field-programmable gate arrays using integer programming. In *Int'l Conf. Computer Aided Design*, November 1995.

[48] J. Cong and Y. Ding. An optimal technology mapping algorithm for delay optimization in lookup-table based FPGA designs. In *Int'l Conf. Computer Aided Design*, November 1992.

[49] J. Cong and Y. Ding. Beyond the combinatorial limit in depth minimization for LUT-based FPGA designs. In *Int'l Conf. Computer Aided Design*, 1993.

[50] J. Cong and Y. Ding. FlowMap: an optimal technology mapping algorithm for delay optimization in lookup-table based FPGA designs. *IEEE Trans. on Computer Aided Design of Integrated Circuits and Systems*, 13(1):1–12, January 1994.

[51] J. Cong and Y. Ding. On area/depth trade-off in LUT-based FPGA technology mapping. *IEEE Transactions on VLSI Systems*, 2(2):137–148, 1994.

[52] J. Cong and Y. Ding. Combinational logic synthesis for LUT based field programmable gate arrays. *ACM Trans. on Design Automation of Electronic Systems*, 1(2):145–204, April 1996.

[53] J. Cong, Y. Fan, G. Han, W. Jiang, and Z. Zhang. Behavior and communication co-optimization for systems with sequential communication media. In *IEEE/ACM Design Automation Conference*, 2006.

[54] J. Cong, Y. Fan, G. Han, X. Yang, and Z. Zhang. Architecture and synthesis for on-chip multi-cycle communication. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pages 550–564, April 2004.

[55] J. Cong, Y. Fan, G. Han, and Z. Zhang. Application-specific instruction generation for configurable processor architectures. In *International Symposium on Field-Programmable Gate Arrays*, February 2004.

[56] J. Cong, J. Fang, M. Xie, and Y. Zhang. MARS—a multilevel full-chip gridless routing system. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 24(3):382–394, March 2005.

[57] J. Cong, H. Huang, and X. Yuan. Technology mapping and architecture evaluation for k/m-macrocell-based FPGAs. *ACM Trans. on Design Automation of Electronic Systems*, 10:3–23, January 2005.

[58] J. Cong and Y. Hwang. Simultaneous depth and area minimization in LUT-based FPGA mapping. *International Symposium on Field-Programmable Gate-Arrays*, February 1995.

[59] J. Cong and Y. Hwang. Structural gate decomposition for depth-optimal technology mapping in LUT-based FPGA design. In *ACM/IEEE Design Automation Conference*, 1996.

[60] J. Cong, A. Kahng, G. Robins, M. Sarrafzadeh, and C. K. Wong. Provably good performance-driven global routing. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 11(6):739–752, June 1992.

[61] J. Cong, T. Kong, J. Shinnerl, M. Xie, and X. Yuan. Large scale circuit placement. *ACM Transactions on Design Automation of Electronic Systems*, 10(2):389–430, April 2005.

[62] J. Cong, K. S. Leung, and D. Zhou. Performance-driven interconnect design based on distributed RC delay model. In *Proc. ACM/IEEE 30th Design Automation Conference*, pages 606–611, June 1993.

[63] J. Cong and S. K. Lim. Physical planning with retiming. In *IEEE International Conference on Computer Aided Design*, pages 2–7, 2000.

[64] J. Cong, Y. Lin, and W. Long. SPFD-based global reviewing. In *International Symposium on Field-Programmable Gate Arrays*, 2002.

[65] J. Cong and W. Long. Theory and algorithm for SPFD-based global rewiring. In *International Workshop on Logic Synthesis*, 2001.

[66] J. Cong and K. Minkovich. Optimality study of logic synthesis for LUT-based FPGAs. In *International Symposium on Field-Programmable Gate Arrays*, February 2006.

[67] J. Cong and B. Preas. A new algorithm for standard cell global routing. In *Proc. Int'l Conf. on Computer-Aided Design*, pages 176–179, November 1988.

[68] J. Cong and M. Romesis. Performance-driven multi-level clustering with application to hierarchical FPGA mapping. In *Design Automation Conference*, 2001.

[69] J. Cong and J. Shinnerl, editors. *Multilevel Optimization in VLSI CAD*. Kluwer Academic Publishers, 2003.

[70] J. Cong and C. Wu. FPGA synthesis with retiming and pipelining for clock period minimization of sequential circuits. In *Design Automation Conference*, 1997.

[71] J. Cong and C. Wu. Optimal FPGA mapping and retiming with efficient initial state computation. *Design Automation Conference*, 1997.

[72] J. Cong, C. Wu, and Y. Ding. Cut ranking and pruning: enabling a general and efficient FPGA mapping solution. *International Symposium on Field-Programmable Gate Arrays*, February 1999.

[73] J. Cong and S. Xu. Delay-optimal technology mapping for FPGAs with heterogeneous LUTs. In *Design Automation Conference*, 1998.

[74] J. Cong and S. Xu. Delay-oriented technology mapping for heterogeneous FPGAs with bounded resources. In *Int'l Conf. Computer Aided Design*, 1998.

[75] J. Cong and S. Xu. Technology mapping for FPGAs with embedded memory blocks. In *International Symposium on Field-Programmable Gate Arrays*, 1998.

[76] J. Cong and S. Xu. Performance-driven technology mapping for heterogeneous FPGAs. *IEEE Trans. on Computer-aided Design of Integrated Circuits and Systems*, 19(11):1268–1281, November 2000.

[77] Ph.D. Dissertation D. Chen. Design and synthesis for low-power FPGAs. *Computer Science Department, University of California*, December 2005.

[78] J. A. Davis, V. K. De, and J. Meindl. A stochastic wire-length distribution for gigascale integration (GSI)—Part I: derivation and validation. 45(3):580–589, March 1998.

[79] G. De Micheli. *Synthesis and optimization of digital circuits*. McGraw-Hill, Inc., 1994.

[80] A. Duncan, D. Hendry, and P. Gray. An overview of the COBRA-ABS high level synthesis system for multi-FPGA systems. In *IEEE Symposium on FPGAs for Custom Computing Machines*, pages 106–115, April 1998.

[81] J. M. Emmert and D. Bhatia. A methodology for fast FPGA floorplanning. In *ACM/SIGDA International Symposium on Field Programmable Gate Arrays*, pages 47–56, February 21-23 1999.

[82] L. A. Entrena and K. T. Cheng. Combinational and sequential logic optimization by redundancy addition and removal. *IEEE Transaction on Computer Aided Design of Integrated Circuits and Systems*, 14(7):909–916, 1995.

[83] W. Fang and A. Wu. Multi-way FPGA partitioning by fully exploiting design hierarchy. *ACM Transactions on Design Automation of Electronic Systems*, 5(1):34–50, January 2000.

[84] A. Farrahi and M. Sarrafzadeh. Complexity of the lookup-table minimization problem for FPGA technology mapping. *IEEE Tran. on Computer Aided Design of Integrated Circuits and Systems*, 13(11):1319–1332, November 1994.

[85] A. H. Farrahi and M. Sarrafzadeh. FPGA technology mapping for power minimization. In *International Workshop in Field Programmable Logic and Applications*, 1994.

[86] FishTail. Design Automation. http://www.fishtail-da.com/.

[87] R. J. Francis, J. Rose, and Z. Vranesic. Technology mapping for lookup table-based FPGA's for performance. In *Int'l Conf. Computer-Aided Design*, November 1991.

[88] R. J. Francis, et al. Chortle-crf: fast technology mapping for lookup table-based FPGAs. In *Design Automation Conference*, 1991.

[89] J. Frankle. Iterative and adaptive slack allocation for performance-driven layout and FPGA routing. In *Proceedings of Design Automation Conference*, pages 536–542, 1992.

[90] A. Gayasen, Y. Tsai, N. Vijaykrishnan, M. Kandemir, M. Irwin, and T. Tuan. Reducing leakage energy in FPGAs using region-constrained placement. In *ACM International Symposium on Field Programmable Gate Arrays*, February 2004.

[91] V. George and J. Rabaey. *Low-energy FPGAs—Architecture and Design*. Kluwer Academic Publishers, 2001.

[92] V. George and J. Rabaey. *Low-Energy FPGAs: Architecture and Design*. Springer, June 2001.

[93] S. Ghiasi, E. Bozorgzadeh, S. Choudhury, and M. Sarrafzadeh. A unified theory of timing budget management. In *IEEE/ACM International Conference on Computer-Aided Design*, pages 653–659, November 2004.

[94] M. Gokhale and J. Stone. Automatic allocation of arrays to memories in FPGA processors with multiple memory banks. In *IEEE Symposium on Field-Programmable Custom Computing Machines*, pages 63–69, April 1999.

[95] P. Gopalakrishnan, X. Li, and L. Pileggi. Architecture-aware FPGA placement using metric embedding. In *IEEE/ACM Design Automation Conference*, pages 460–465, 2006.

[96] M. Haldar, A. Nayak, A. Choudhary, and P. Banerjee. Parallel algorithms for FPGA placement. In *Proc. Great Lakes Symposium on VLSI (GVLSI 2000)*, March 2000.

[97] Z. Hasan, D. Harrison, and M. Ciesielski. A fast partition method for PLA-based FPGAs. *IEEE Design and Test of Computers*, December 1992.

[98] G. D. Hatchel and F. Somenzi. *Logic Synthesis and Verification Algorithms*. Kluwer Academic Publishers, 1996.

[99] P. S. Hauge, R. Nair, and E. J. Yoffa. Circuit placement for predictable performance. In *International Conference of Computer Aided Design*, pages 88–91, 1987.

[100] J. He and J. Rose. Technology mapping for heterogeneous FPGAs. In *International Symposium on Field Programmable Gate Arrays*, 1994.

[101] M. Hrkic and J. Lillis. S-Tree: a technique for buffered routing tree synthesis. In *Design Automation Conference*, 2002.

[102] M. Hrkic, J. Lillis, and G. Beraudo. An approach to placement-coupled logic replication. In *ACM/IEEE Design Automation Conference*, pages 711–716, June 2004.

[103] Y. Hu, Y. Lin, L. He, and T. Tuan. Simultaneous time slack budgeting and retiming for Dual-Vdd FPGA power reduction. In *IEEE/ACM Design Automation Conference*, 2006.

[104] S. W. Hur and J. Lillis. Mongrel: hybrid techniques for standard cell placement. In *International Conference of Computer Aided Design*, 2000.

[105] J. Hwang, F. Chiang, and T. Hwang. A re-engineering approach to low power FPGA design using SPFD. In *Design Automation Conference*, 1998.

[106] M. Inuani and J. Saul. Re-synthesis in technology mapping for heterogeneous FPGAs. In *International Conference on Computer Design*, 1998.

[107] P. Jamieson and J. Rose. A verilog RTL synthesis tool for heterogeneous FPGAs. In *International Conference on Field Programmable Logic and Applications*, August 2005.

[108] A. B. Kahng, S. Reda, and Q. Wang. Architecture and details of a high quality, large-scale analytical placer. In *ACM/IEEE Intl. Conf. on Computer-Aided Design*, pages 891–898, November 2005.

[109] G. Karypis and V. Kumar. Multilevel hypergraph partitioning. In *Design Automation Conference*, 1997.

[110] A. Kaviani and S. Brown. Technology mapping issues for an FPGA with lookup tables and PLA-like blocks. In *International Symposium on Field Programmable Gate Arrays*, 2000.

[111] K. Keutzer. DAGON: technology binding and local optimization by DAG matching. *ACM/IEEE Design Automation Conference*, pages 341–347, 1987.

[112] D. Kim, J. Jung, S. Lee, J. Jeon, and K. Choi. Behavior-to-placed RTL synthesis with performance-driven placement. In *Int. Conf. on Computer Aided Design*, pages 320–326, November 2001.

[113] A. Koch. Structured design implementation—a strategy for implementing regular datapaths on FPGAs. In *International Symposium on Field Programmable Gate Arrays*, pages 151–157, 1996.

[114] T. Kong. A novel net weighting algorithm for timing-driven placement. In *IEEE/ACM International Conference on Computer-Aided Design*, pages 172–176, 2002.

[115] M. R. Korupolu, K. K. Lee, and D. F. Wong. Exact tree-based FPGA technology mapping for logic blocks with independent LUTs. In *Design Automation Conference*, 1998.

[116] L. Kou, G. Markowsky, and L. Berman. A fast algorithm for steiner trees. *Acta Informatica*, 15:141–145, 1981.

[117] J. L. Kouloheris. *Empirical Study of the Effect of Cell Granularity on FPGA Density and Performance*. Ph.D. Thesis, Stanford University, 1993.

[118] S. Krishnamoorthy and R. Tessier. Technology mapping algorithms for hybrid FPGAs containing lookup tables and PLAs. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 22(5), May 2003.

[119] E. Kusse and J. Rabaey. Low-energy embedded FPGA structures. In *Proc. of International Symposium on Low Power Electronics and Design*, August 1998.

[120] J. Lamoureux and S. J. E. Wilton. On the interaction between power-aware FPGA CAD algorithms. In *IEEE International Conference on Computer-Aided Design*, November 2003.

[121] Lattice. Power estimation in ispMACH 5000B devices, May 2002.

[122] E. L. Lawler, K. N. Levitt, and J. Turner. Module clustering to minimize delay in digital networks. *Trans. On Computer*, C18(1), 1969.

[123] S. Lee and D. F. Wong. Timing-driven routing for FPGAs based on Lagrangian relaxation. In *Proc. of International Symposium on Physical Designs*, pages 176–181, April 2002.

[124] Y. S. Lee and C. H. Wu. A performance and routability-driven router for FPGAs considering path delay. In *Proc. of Design Automation Conference*, pages 557–561, 1995.

[125] C. Legl, B. Wurth, and K. Eckl. A Boolean approach to performance-directed technology mapping for LUT-based FPGA designs. In *Design Automation Conference*, June 1996.

[126] E. Lehman, Y. Watanabe, J. Grodstein, and H. Harkness. Logic decomposition during technology mapping. In *IEEE/ACM International Conference on Computer-Aided Design*, pages 264–271, 1995.

[127] E. Lehman, Y. Watanabe, J. Grodstein, and H. Harkness. Logic decomposition during technology mapping. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 16(8):813–834, 1997.

[128] C. E. Leiserson and J. B. Saxe. Retiming synchronous circuitry. *Algorithmica*, 6:5–35, 1991.

[129] G. Lemieux and S. D. Brown. A detailed routing algorithm for allocating wire segments in FPGAs. *ACM/SIGDA Physical Design Workshop*, 1993.

[130] F. Li, D. Chen, L. He, and J. Cong. Architecture evaluation for power-efficient FPGAs. In *ACM International Symposium on Field Programmable Gate Arrays*, pages 175–184, Monterey, California, 2003.

[131] F. Li, Y. Lin, and L. He. FPGA power reduction using configurable Dual-Vdd. In *IEEE/ACM Design Automation Conference*, pages 735–740, June 2004.

[132] F. Li, Y. Lin, and L. He. Vdd programmability to reduce FPGA interconnect power. In *IEEE/ACM International Conference on Computer-Aided Design*, pages 760–765, San Jose, November 2004.

[133] F. Li, Y. Lin, L. He, D. Chen, and J. Cong. Power modeling and characteristics of field programmable gate arrays. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 24(11), November 2005.

[134] F. Li, Y. Lin, L. He, and J. Cong. Low-power FPGA using dual-Vdd/Dual-Vt techniques. In *International Symposium on Field Programmable Gate Arrays*, pages 42–50, February 2004.

[135] H. Li, S. Katkoori, and W. K. Mak. Power minimization algorithms for LUT based FPGA technology mapping. *ACM Transactions on Design Automation of Electronic Systems*, 9(1):33–51, January 2004.

[136] R. Liberskind Hadas, N. Hasan, J. Cong, P. Mckinley, and C. L. Liu. Kluwer Academic Publishers, 1992.

[137] E. Lin and S. Wilton. Macrocell architectures for product term embedded memory arrays. *Field Programmable Logic Applications*, pages 48–58, August 2001.

[138] J. Lin, D. Chen, and J. Cong. Optimal simultaneous mapping and clustering for FPGA delay optimization. In *IEEE/ACM Design Automation Conference*, 2006.

[139] J. Lin, A. Jagannathan, and J. Cong. Placement-driven technology mapping for LUT-based FPGAs. In *International Symposium on Field Programmable Gate Arrays*, pages 121–126, February 2003.

[140] A. Ling, D. Singh, and S. Brown. FPGA technology mapping: a study of optimality. In *Design Automation Conference*, 2005.

[141] P. Maidee, C. Ababei, and K. Bazarga. Timing-driven partitioning-based placement for Island style FPGAs. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 24(3):395–406, March 2005.

[142] V. Manohararajah, S. D. Brown, and Z. G. Vranesic. Heuristics for area minimization in LUT-based FPGA technology mapping. In *International Workshop of Logic Synthesis*, 2004.

[143] A. Marquardt, V. Betz, and J. Rose. Using cluster-based logic blocks and timing-driven packing to improve FPGA speed and density. In *ACM/SIGDA International Symposium on Field Programmable Gate Arrays*, pages 37–46, 1999.

[144] A. Marquardt, V. Betz, and J. Rose. Timing-driven placement for FPGAs. In *International Symposium on Field Programmable Gate Arrays*, pages 203–213, Monterey, Ca., February 2000.

[145] A. Mathur and C. L. Liu. Performance-driven technology mapping for lookup-table based FPGAs using the general delay model. In *International Workshop on Field Programmable Gate Arrays*, February 1994.

[146] L. Mcmurchie and C. Ebeling. PathFinder: A negotiation-based performance-driven router for FPGAs. In *Proceedings of International Symposium on Field-Programmable Gate Arrays*, February 1995.

[147] A. Mishchenko, S. Chatterjee, and R. Brayton. Improvements to technology mapping for LUT-based FPGAs. In *International Symposium on Field-Programmable Gate Arrays*, 2006.

[148] R. Mukherjee and S. Ogrenci Memik. Evaluation of Dual Vdd fabrics for low power FPGAs. In *Asia-South Pacific Design Automation Conference*, January 2005.

[149] R. Murgai, R. Brayton, and A. Sangiovanni Vincentelli. On clustering for minimum delay/area. In *Int'l Conf. Computer Aided Design*, November 1991.

[150] R. Murgai, R. Brayton, and A. Sangiovanni Vincentelli. *Logic Synthesis for Field-Programmable Gate Arrays*. Springer, July 1995.

[151] R. Murgai, et al. Improved logic synthesis algorithms for table look up architectures. In *Int'l Conf. Computer Aided Design*, November 1991.

[152] S. Mutoh, et al. 1-V Power supply high-speed digital circuit technology with multi-threshold-voltage CMOS. *IEEE Journal of Solid-State Circuits*, 30(8):847–854, August 1995.

[153] S. K. Nag and R. A. Rutenbar. Performance-driven simultaneous placement and routing for FPGAs. *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, 17(6):499–518, June 1998.

[154] R. Nair. A simple yet effective technique for global wiring. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, CAD-6(6):165–172, March 1987.

[155] G. J. Nam, F. Aloul, K. A. Sakallah, and R. A. Rutenbar. A comparative study of two Boolean formulations of FPGA detailed routing constraints. *IEEE Transactions on Computers*, 53(6):688–696, June 2004.

[156] G. J. Nam, K. A. Sakallah, and R. A. Rutenbar. Satisfiability-based layout revisited: detailed routing of complex FPGAs via search-based Boolean SAT. In *ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, pages 167–175, February 1999.

[157] P. Pan and C. C. Lin. A new retiming-based technology mapping algorithm for LUT-based FPGAs. In *International Symposium on Field-Programmable Gate Arrays*, 1998.

[158] P. Pan and C. L. Liu. Optimal clock period FPGA technology mapping for sequential circuits. In *Design Automation Conf.*, June 1996.

[159] P. Pan and C. L. Liu. Technology mapping of sequential circuits for LUT-based FPGAs for performance. In *International Symposium on Field-Programmable Gate Arrays*, 1996.

[160] P. Pan and C. L. Liu. Optimal clock period FPGA technology mapping for sequential circuits. *ACM Transactions on Design Automation of Electronic Systems*, 3(3):437–462, 1998.

[161] K. Poon, S. J. E. Wilton, and A. Yan. A detailed power model for field programmable gate arrays. *ACM Transactions on Design Automation of Electronic Systems*, 10(2):279–302, April 2005.

[162] A. Rahman and V. Polavarapuv. Evaluation of low-leakage design techniques for field programmable gate arrays. In *International Symposium on Field Programmable Gate Arrays*, February 2004.

[163] R. Rajaraman and D. F. Wong. Optimal clustering for delay minimization. In *Design Automation Conference*, June 1993.

[164] J. Rose. Parallel global routing for standard cells. *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, 9(10):1085–1095, October 1990.

[165] Y. Sankar and J. Rose. Trading quality for compile time: ultra-fast placement for FPGAs. In *International Symposium on Field Programmable Gate Arrays*, pages 157–166, 1999.

[166] M. Schlag, J. Kong, and P. K. Chan. Routability-driven technology mapping for lookup table-based FPGAs. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 13(1):13–26, 1994.

[167] H. Schmit, L. Arnstein, D. Thomas, and E. Lagnese. Behavioral synthesis for FPGA-based computing. In *Workshop on FPGAs for Custom Computing Machines*, pages 125–132, 1994.

[168] E. M. Sentovich, et al. *SIS: A system for sequential circuit synthesis.* Berkeley, CA, University of California, 1992. Dept. of Electrical Engineering and Computer Science.

[169] L. Shang and N. K. Jha. High-level power modeling of CPLDs and FPGAs. In *IEEE International Conference on Computer Design*, September 2001.

[170] L. Shang, A. Kaviani, and K. Bathala. Dynamic power consumption in virtex-II FPGA family. In *ACM International Symposium on Field Programmable Gate Arrays*, February 2002.

[171] K. R. Shayee, J. Park, and P. Diniz. Performance and area modeling of complete FPGA designs in the presence of loop transformations. *IEEE Transactions on Computers*, 53(11):1420–1435, November 2004.

[172] J. P. M. Silva and K. A. Sakallah. GRASP—a new search algorithm for satisfiability. In *Proc. ACM/IEEE Int'l Conf. Computer Aided Design*, November 1997.

[173] A. Singh and M. Marek Sadowska. Efficient circuit clustering for area and power reduction in FPGAs. In *ACM International Symposium on Field Programmable Gate Arrays*, February 2002.

[174] D. Singh and S. Brown. Integrated retiming and placement for field programmable gate arrays. In *International Symposium on Field Programmable Gate Arrays*, pages 67–76, February 2002.

[175] A. Srivastava, D. Sylvester, and D. Blaauw. Power minimization using simultaneous gate sizing, Dual-Vdd and Dual-Vth assignment. In *Design Automation Conference*, 2004.

[176] H. Styles and W. Luk. Branch optimization techniques for hardware compilation. In *International Conference on Field Programmable Logic and Applications*, 2003.

[177] W. Sun, M. Wirthlin, and S. Neuendorffer. Combining module selection and resource sharing for efficient FPGA pipeline synthesis. In *Int'l Symposium on Field Programmable Gate Arrays*, 2006.

[178] J. Swartz, V. Betz, and J. Rose. A fast routability-driven router for FPGAs. In *Int'l Symposium on Field Programmable Gate Arrays*, pages 140–149, Monterey, CA, 1998.

[179] Synopsys. http://www.synopsys.com/products/products_matrix.html.
[180] SystemC. http://www.systemc.org.
[181] M. Takahashi, et al. A 60mW MPEG4 video codec using clustered voltage scaling with variable supply-voltage scheme. *Journal of Solid-State Circuits*, 1998.
[182] R. Tessier. Fast placement approaches for FPGAs. *ACM Transactions on Design Automation of Electronic Systems*, 7(2):284–305, April 2002.
[183] The MathWorks. http://www.mathworks.com/.
[184] N. Togawa, M. Sato, and T. Ohtsuki. Maple: a simultaneous technology mapping, placement, and global routing algorithm for field-programmable gate arrays. In *Int'l Conf. Computer Aided Design*, 1994.
[185] N. Togawa, M. Sato, and T. Ohtsuki. A simultaneous placement and global routing algorithm with path length constraints for transport-processing FPGAs. In *Asia South Pacific Design Automation Conf.*, pages 569–578, 1997.
[186] S. Trimberger. *Field-Programmable Gate Array Technology*. Springer, January 1994.
[187] T. Tuan, S. Kao, A. Rahman, S. Das, and S. Trimberger. A 90nm low-power FPGA for battery-powered applications. In *International Symposium on Field Programmable Gate Arrays*, 2006.
[188] T. Tuan and B. Lai. Leakage power analysis of a 90nm FPGA. In *Custom Integrated Circuits Conference*, 2003.
[189] K. Usami and M. Horowitz. Clustered voltage scaling for low-power design. In *International Symposium on Low Power Design*, April 1995.
[190] H. Vaishnav and M. Pedram. Delay optimal clustering targeting low-power VLSI circuits. *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, 18(6), June 1999.
[191] K. Wakabayashi. C-based behavioral synthesis and verification analysis on industrial design examples. In *Asian and South Pacific Design Automation Conference*, pages 344–348, January 2004.
[192] K. Wakabayashi and T.Okamoto. C-based SoC design flow and EDA tools: an ASIC and system vendor perspective. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 19(12):1507–1522, December 2000.
[193] Z. H. Wang, E. C. Liu, J. Lai, and T. C. Wang. Power minimization in LUT-based FPGA technology mapping. In *Asia South Pacific Design Automation Conference*, 2001.
[194] L. Wei, Z. Chen, M. Johnson, K. Roy, and V. De. Design and optimization of low voltage high performance dual threshold CMOS circuits. In *Design Automation Conference*, 1998.
[195] K. Weiß, C. Oetker, I. Katchan, T. Steckstor, and W. Rosenstiel. Power estimation approach for SRAM-based FPGAs. In *ACM International Symposium on Field Programmable Gate Arrays*, February 2000.
[196] S. Wilton. SMAP: heterogeneous technology mapping for area reduction in FPGAs with embedded memory arrays. In *International Symposium on Field Programmable Gate Arrays*, 1998.

[197] S. Wilton. Heterogeneous technology mapping for area reduction in FPGAs with embedded memory arrays. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 19(1), January 2000.

[198] S. Wilton. Heterogeneous technology mapping for FPGAs with dual-port embedded memory arrays. In *International Symposium on Field Programmable Gate Arrays*, 2000.

[199] F. G. Wolff, M. J. Knieser, D. J. Weyer, and C. A. Papachristou. High-level low power FPGA design methodology. In *IEEE National Aerospace Conference*, 2000.

[200] P. Wong, L. Cheng, Y. Lin, and L. He. FPGA device and architecture evaluation considering process variation. In *Proc. IEEE/ACM International Conf. on Computer-Aided Design*, November 2005.

[201] Y. L. Wu and M. Marek Sadowska. An efficient router for 2-D field programmable gate arrays. In *Proc. of European Design Automation Conference*, pages 412–416, 1994.

[202] Y. L. Wu and M. Marek Sadowska. Orthogonal greedy coupling—a new optimization approach to 2-D FPGA routing. In *Proc. of Design Automation Conference*, June 1995.

[203] Xilinx. Website, http://www.xilinx.com.

[204] Xilinx. Spartan-3E Data Sheets. http://direct.xilinx.com/bvdocs/publications/ds312.pdf.

[205] Xilinx. Virtex-4 Data Sheet. http://www.xilinx.com.

[206] Xilinx. Virtex-4 Web Power Tool Version 8.1.01. http://www.xilinx.com/products/silicon_solutions/fpgas/virtex/virtex4/index.htm.

[207] Xilinx. Virtex-5 Data Sheet. http://www.xilinx.com.

[208] Xilinx. white paper 205: hardware/software codesign for platform FPGAs. http://www.xilinx.com/products/design_resources/proc_central/resource/hardware_software_codesign.pdf.

[209] Xilinx. XPower Tool. http://www.xilinx.com/products/design_resources/design_tool/grouping/power_tools.htm.

[210] Xilinx. A simple method of estimating power in XC4OOOXL/EX/E FPGAs, June 30 1997. Application Brief X014.

[211] M. Xu and F. J. Kurdahi. ChipEst-FPGA: a tool for chip level area and timing estimation of lookup table based FPGAs for high level applications. In *Asia and South Pacific Design Automation Conference*, January 1997.

[212] M. Xu and F.J. Kurdahi. *Design Automation and Test in Europe*. Layout-driven high level synthesis for FPGA based architectures. 1998.

[213] S. Yamshita, H. Sawada, and A. Nagoya. A new method to express functional permissibilities for LUT based FPGAs and its applications. In *International Conference on Computer Aided Design*, pages 254–261, 1996.

[214] H. Yang and D. F. Wong. Edge-map: optimal performance driven technology mapping for iterative LUT based FPGA designs. In *Int'l Conf. Computer Aided Design*, November 1994.

[215] A. G. Ye, J. Rose, and D. Lewis. Synthesizing datapath circuits for FPGAs with emphasis on area minimization. In *International Conference on Field-Programmable Technology*, December 2002.

[216] A. Z. Zelikovsky. An 11/6 approximation algorithm for the network Steiner problem. *Algorithmica*, 9:463–470, 1993.

[217] L. Zhang, C. Madigan, M. Moskewicz, and S. Malik. Efficient conflict driven learning in a Boolean satisfiability solver. In *International Conference on Computer-Aided Design*, pages 279–285, 2001.

[218] Y. Zhan, et al. Statistical timing analysis: correlation-aware statistical timing analysis with non-gaussian delay distributions. In *Proc. of Design Automation Conf.*, June 2005.

[219] P. S. Zuchowski, et al. A hybrid ASIC and FPGA architecture. In *International Conference on Computer-Aided Design*, November 2002.