McGraw-Hill Series on
Computer Engineering

LOGIC SYNTHESIS

Srinivas Devadas
Abhijit Ghosh
Kurt Keutzer

Logic Synthesis

Srinivas Devadas

Abhijit Ghosh

Kurt Keutzer

McGraw-Hill, Inc.

New York San Francisco Washington, D.C. Auckland Bogotá
Caracas Lisbon London Madrid Mexico City Milan
Montreal New Delhi San Juan Singapore
Sydney Tokyo Toronto

Library of Congress Cataloging-in-Publication Data

Devadas, Srinivas.
Logic synthesis / Srinivas Devadas, Abhijit Ghosh, Kurt Keutzer.
p. cm. — (McGraw-Hill series on computer engineering)
Includes bibliographical references and index.
ISBN 0-07-016500-9
1. Integrated circuits—Very large scale integration—Design—Data processing.
2. Logic design—Data processing. 3. Computer-aided design.
I. Ghosh, Abhijit. II. Keutzer, Kurt William. III. Title.
IV. Series: Series on computer engineering.
TK7874.D478 1994
621.39'5—dc20 93-41524
CIP

2 3 4 5 6 7 8 9 0 DOC/DOC 9 0 9 8 7 6

ISBN 0-07-016500-9

The sponsoring editor for this book was Stephen S. Chapman.
Printed and bound by R. R. Donnelley & Sons Company.

Logic Synthesis

Preface

The dramatic increase in designer productivity over the past decade in the area of *very large scale integrated* (VLSI) circuit design is the direct result of the development of sophisticated *computer-aided design* (CAD) tools. Today, designers routinely describe the functionality of a circuit in a *hardware description language* (HDL) (which is a high-level description akin to a programming language) and use synthesis tools to produce optimized circuit layouts that can be sent off to a chip manufacturer to be fabricated on a silicon integrated circuit.

The two major areas in VLSI synthesis that have enabled vastly improved design turnaround times are logic synthesis and layout synthesis. The logic synthesis process consists of the translation of the input HDL description into a gate-level circuit and the optimization of the gate-level circuit. An optimized layout is produced for the final gate-level circuit by the layout synthesis process. Both the logic and layout synthesis process require the solution of difficult combinatorial optimization problems. In this book we focus solely on the logic synthesis process.

Switching and automata theory form the cornerstones of logic synthesis. Combinatorial problems associated with optimizing switching circuits abound in logic synthesis. In order to meet the demands of the designers, logic optimization systems have to be versatile and efficient. Versatility implies that the system should be able to target a variety of design parameters such as circuit area, delay, power dissipation, and testability. Efficiency implies that the system should be able to produce near-optimum or at least acceptable results for large VLSI circuits with reasonable CPU time expenditure.

The development of logic optimization systems that are versatile and efficient posed one of the major challenges for CAD in the 1980s. Today, thanks to a large amount of research and developmental effort such systems are in wide use among integrated circuit designers. This book focuses on describing the basic principles of logic design as well as the practical aspects of engineering a logic synthesis system.

Very few individuals will themselves undertake to implement a logic synthesis system, but we feel that understanding the core principles of logic synthesis will be of use to a number of communities.

One community consists of educators in computer science and electrical engineering. In general an educator is looking for material that both disciplines the intellect of the student as well as prepares the student for practical problems the student is likely to encounter. In particular, the modern educator searching to find material that is relevant to the logical design of VLSI circuits has been faced with choosing either the classical works on switching and automata theory or anthologies of collected articles on logic synthesis and optimization methods. A large body of theory as well as many practical algorithms and methodologies to design logic circuits have been developed by researchers in logic synthesis, but so far theoretical and practical results have only been documented in numerous articles. This book attempts to fill the gap between the classical books written in the 1970s and modern logic optimization articles.

Another community consists of integrated circuit designers who are presently using logic synthesis to design integrated circuits. For these designers the former skills of handcrafting transistor-level layout or manually entering the schematics of a carefully designed gate-level implementation of a circuit are being replaced by the skill of writing efficient HDL models of integrated circuits. In order to develop the skill of writing HDL models of circuits that will result in efficient implementations, it is necessary for a designer to understand the basic principles underlying logic synthesis and optimization. One of the recurrent obstacles for a hardware designer using synthesis is the assumption that two functionally equivalent HDL models will produce similar circuits after logic synthesis and optimization. While we do not present a primer on HDL model development, it is our hope that by making the designer understand the capabilities and limitations of logic optimization software the designers will be able to build more efficient circuits within a synthesis framework. It is our hope that the discipline of logic synthesis will play just as central a role to the training and education of circuit designers as the discipline of compilers plays in the education and training of software developers.

A final community we wish to address are those fellow researchers in CAD who are working in logic synthesis or a related area. Research in logic synthesis is a very satisfying enterprise because improvements in algorithms can immediately translate into smaller or faster circuits, and smaller and faster circuits have substantial commercial impact. Researchers wishing to get up to speed in logic synthesis have also been forced to rely on the classical works on switching and automata theory or on the anthologies of collected articles on logic synthesis and optimization. Here we hope to provide these researchers with a self-contained reference book that covers most of the principal synthesis and optimization techniques.

This book is organized into ten chapters. We provide an introduction to synthesis, verification, and testing in Chapter 1. The translation of an HDL model into a netlist of gates is introduced in Chapter 2. While integrated circuits implement sequential circuits, the most successful logic synthesis and optimization techniques have focused on the combinational portions. Therfore, in the remainder of the book we focus on the combinational portions of the circuits. The core algorithms for logic optimization were initially developed on two-level circuits and most easily understood in that context. For these reasons we will introduce these circuits first in Chapter 3. In the following chapters, we deal with the problems of minimizing two-level circuits so as to improve area and speed. We focus on testing a two-level circuit under various models of faulty behavior. We show how logic transformations applied to minimize the two-level circuit's area affect the testability of the circuit. While logic optimization techniques were first developed on two-level circuits, multilevel circuits are of much greater practical importance. In Chapters 6 through 9 we deal with the problems of synthesizing multilevel combinational logic circuits for minimal area, maximal speed, and high testability. Strong relationships between the area, speed, and testability of a circuit are highlighted throughout the book. We summarize the state-of-the-art in logic synthesis in Chapter 10.

Circuit representations and data structures cut across all facets of design such as synthesis, testing, and verification. At the combinational or sequential circuit level, Boolean functions are manipulated in various ways. The search for more efficient representations of Boolean functions is ceaseless, mainly because discovering such representations can have a significant impact on synthesis, testing, and verification problems. In this book we describe commonly used representations for combinational circuits and their advantages and disadvantages when applied to particular problems in synthesis and test.

Since we intend this book to be useful to CAD researchers, educators, and VLSI designers, we have included considerable detail in the description of the various algorithms. Because we cannot comprehensively present the full panorama of logic optimization techniques, we present those techniques that have proven to be most useful in practice.

Acknowledgements

Over the years, several people have helped to deepen our understanding of VLSI synthesis, test generation, and synthesis for testability. We thank Jonathan Allen, Pranav Ashar, Robert Brayton, Michael Bryan, Gaetano Borriello, Raul Camposano, Steve Carlson, Tim Cheng, Aart De Geus, Giovanni De Micheli, Alfred Dunlop, Gary Hachtel, Niraj Jha, Charles Leiserson, Michael Lightner, Bill Lin, Tony Ma, Sharad Malik, Rick McGeer, Richard Newton, Paul Penfield, Sudhakar Reddy, Richard Rudell, Alexander Saldanha, Alberto Sangiovanni-Vincentelli, Fabio Somenzi, Kanwar Jit Singh, Albert Wang, Ruey-sing Wei, Jacob White, Tom Williams, and Wayne Wolf.

Some of the problems in this book have been taken or modified from Robert Brayton's class notes. Material modified from a variety of other sources is acknowledged by citations in the text.

Lastly, we thank Sulochana Devadas, Eliane Setton, Cate Hunter, and the rest of our families for their continual patience and encouragement.

Logic Synthesis

Contents