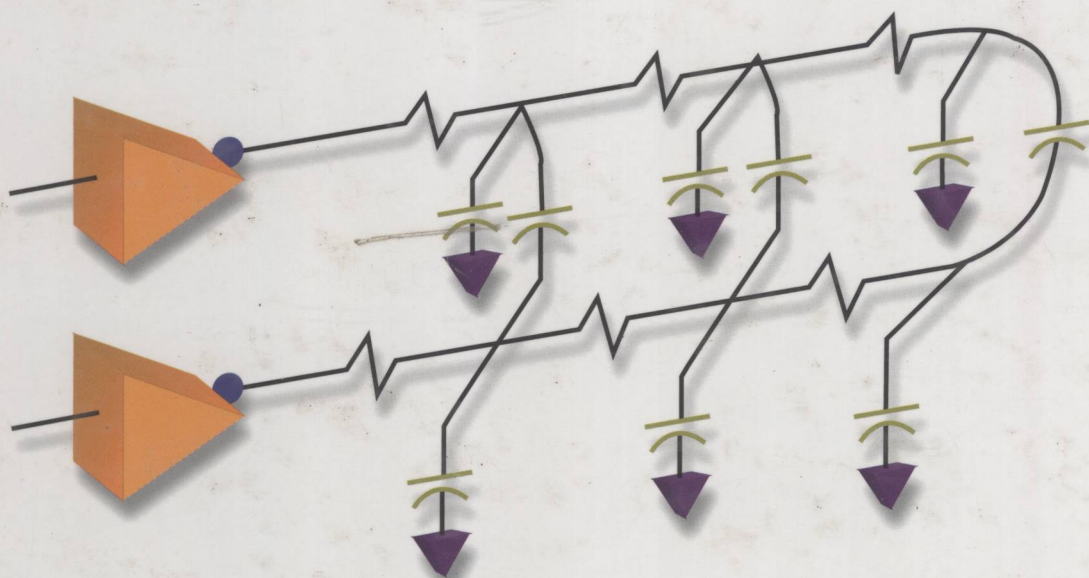


Third Edition

Modern VLSI DESIGN

System-on-Chip Design



Wayne Wolf

Prentice Hall Modern Semiconductor Design Series

TN47
W855
E.3

MODERN VLSI DESIGN

System-on-Chip Design

Third Edition

Wayne Wolf



E200201420



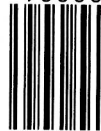
Prentice Hall PTR
Upper Saddle River, NJ 07458
www.phptr.com

ISBN 0-13-061970-1



9 790130 619708

90000



Library of Congress Cataloging-in-Publication Data

A CIP catalog record for this book can be obtained from the Library of Congress.

Editorial/production supervision: *Patti Guerrieri*
Acquisitions editor: *Bernard Goodwin*
Marketing manager: *Dan DePasquale*
Manufacturing manager: *Alexis R. Heydt-Long*
Editorial assistant: *Michelle Vincenti*
Cover design director: *Jerry Votta*
Cover designer: *Anthony Gemmellaro*



© 2002, 1998, 1994 by Prentice Hall PTR
Prentice-Hall, Inc.
Upper Saddle River, NJ 07458

Prentice Hall books are widely used by corporations and government agencies for training, marketing, and resale.

The publisher offers discounts on this book when ordered in bulk quantities. For more information, contact: Corporate Sales Department, Phone: 800-382-3419; Fax: 201-236-7141; E-mail: corpsales@prenhall.com; or write: Prentice Hall PTR, Corp. Sales Dept., One Lake Street, Upper Saddle River, NJ 07458.

Illustrated and typeset by the author. This book was typeset using FrameMaker. Illustrations were drawn using Adobe Illustrator, with layout plots generated by cif2ps.

Photo credits: Example 1-1: Intel Corporation; Figure 2-2: IBM; Figure 2-5: Agere; Figure 2-15: UMC; Figure 2-16: IBM; Figure 6-1: Hewlett-Packard Company; Example 7-1: Tom Way, IBM Microelectronics, Essex Junction, VT; Figure 7-19: Wayne Wolf; Figure 7-20: Wayne Wolf and IBM; Example 8-4: Philips.

All products or services mentioned in this book are the trademarks or service marks of their respective companies or organizations.

All rights reserved. No part of this book may be reproduced, in any form or by any means, without permission in writing from the publisher.

Second edition published under the title *Modern VLSI Design: Systems on Silicon*.

Printed in the United States of America

10 9 8 7 6 5 4 3 2

ISBN 0-13-061970-1

Pearson Education LTD.
Pearson Education Australia PTY, Limited
Pearson Education Singapore, Pte. Ltd.
Pearson Education North Asia Ltd.
Pearson Education Canada, Ltd.
Pearson Educación de Mexico, S.A. de C.V.
Pearson Education — Japan
Pearson Education Malaysia, Pte. Ltd.

for Nancy (as always)

Preface to the Third Edition

This third edition of *Modern VLSI Design* includes both incremental refinements and new topics. All these changes are designed to help keep up with the fast pace of advancements in VLSI technology and design.

The incremental refinements in the book include improvements in the discussion of low power design, the chip project, and the lexicon. Low power design was discussed in the second edition, but has become even more complex due to the higher leakages found at smaller transistor sizes. The PDP-8 used in previous editions has been replaced with a more modern data path design. Designing a complete computer is beyond the scope of most VLSI courses, but a data path makes a good class project. I have also tried to make the lexicon a more comprehensive guide to the terms in the book.

This edition shows more major improvements to the discussions of interconnect and hardware description languages. Interconnect has become increasingly important over the past few years, with interconnect delays often dominating total delay. I decided it was time to fully embrace the importance of interconnect, especially with the advent of copper interconnect. This third edition now talks more thoroughly about interconnect models, crosstalk, and interconnect-centric logic design.

The third edition also incorporates a much more thorough discussion of hardware description languages. Chapter 8, which describes architectural design, now introduces VHDL and Verilog as the major hardware description languages. Though these sections are not meant to be thorough manuals for these languages, they should provide enough information for the reader to understand the major concepts of the languages and to be able to read design examples in those languages.

As with the second edition, you can find additional helpful material on the World Wide Web at <http://www.ee.princeton.edu/~wolf/modern-vlsi>. This site includes overheads useful either for teaching or for self-paced learning. The site also includes supplementary materials, such as layouts and HDL descriptions. Instructors may request a book of answers to the problems in the book by calling Prentice Hall directly.

I'd like to thank Al Casavant and Ken Shepard for their advice on interconnect analysis and Joerg Henkel for his advice on low power design. I'd also like to thank Fred Rosenberger for his many helpful comments on the book. As always, any mistakes are mine.

Wayne Wolf

Princeton, New Jersey

Preface to the Second Edition

Every chapter in this second edition of *Modern VLSI Design* has been updated to reflect the challenges looming in VLSI system design. Today's VLSI design projects are, in many cases, mega-chips which not only contain tens (and soon hundreds) of millions of transistors, but must also run at very high frequencies. As a result, I have emphasized circuit design in a number of ways: the fabrication chapter spends much more time on transistor characteristics; the chapter on gate design covers a wider variety of gate designs; the combinational logic chapter enhances the description of interconnect delay and adds an important new section on crosstalk; the sequential logic chapter covers clock period determination more thoroughly; the subsystems chapter gives much more detailed descriptions of both multiplication and RAM design; the floorplanning chapter spends much more time on clock distribution.

Beyond being large and fast, modern VLSI systems must frequently be designed for low power consumption. Low-power design is of course critical for battery-operated devices, but the sheer size of these VLSI systems means that excessive power consumption can lead to heat problems. Like testing, low-power design cuts across all levels of abstraction, and you will find new sections on low power throughout the book.

The reader familiar with the first edition of this book will notice that the combinational logic material formerly covered in one chapter (Chapter 3) has been split into two chapters, one of logic gates and another on combinational networks. This split was the result of the great amount of material added on circuit design added to the early chapters of the book. Other, smaller rearrangements have also been made in the book, hopefully aiding clarity.

You can find additional helpful material on the World Wide Web at <http://www.ee.princeton.edu/~wolf/modern-vlsi>. This site includes overheads useful either for teaching or for self-paced learning. The site also includes supplementary materials, such as layouts and VHDL descriptions. Instructors may request a book of answers to the problems in the book by calling Prentice Hall directly.

I would especially like to thank Derek Beatty, Luc Claesen, John Darringer, Srinivas Devadas, Santanu Dutta, Michaela Guiney, Alex Ishii, Steve Lin, Rob Mathews, Cherrice Traver, and Steve Trimberger for their comments and suggestions on this second edition.

Wayne Wolf

Princeton, New Jersey

Preface

This book was written in the belief that VLSI design is *system* design. Designing fast inverters is fun, but designing a high-performance, cost-effective integrated circuit demands knowledge of all aspects of digital design, from application algorithms to fabrication and packaging. Carver Mead and Lynn Conway dubbed this approach the tall-thin designer approach. Today's hot designer is a little fatter than his or her 1979 ancestor, since we now know a lot more about VLSI design than we did when Mead and Conway first spoke. But the same principle applies: you must be well-versed in both high-level and low-level design skills to make the most of your design opportunities.

Since VLSI has moved from an exotic, expensive curiosity to an everyday necessity, universities have refocused their VLSI design classes away from circuit design and toward advanced logic and system design. Studying VLSI design as a system design discipline requires such a class to consider a somewhat different set of areas than does the study of circuit design. Topics such as ALU and multiplexer design or advanced clocking strategies used to be discussed using TTL and board-level components, with only occasional nods toward VLSI implementations of very large components. However, the push toward higher levels of integration means that most advanced logic design projects will be designed for integrated circuit implementation.

I have tried to include in this book the range of topics required to grow and train today's tall, moderately-chubby IC designer. Traditional logic design topics, such as adders and state machines, are balanced on the one hand by discussions of circuits and layout techniques and on the other hand by the architectural choices implied by scheduling and allocation. Very large ICs are sufficiently complex that we can't tackle them using circuit design techniques alone; the top-notch designer must understand enough about architecture and logic design to know which parts of the circuit and layout require close attention. The integration of system-level design techniques, such as scheduling, with the more traditional logic design topics is essential for a full understanding of VLSI-size systems.

In an effort to systematically cover all the problems encountered while designing digital systems in VLSI, I have organized the material in this book relatively bottom-up, from fabrication to architecture. Though I am a strong fan of top-down design, the technological limitations which drive architecture are best learned starting with fabrication and layout. You can't expect to fully appreciate all the nuances of why a particular design step is formulated in a certain way until you have completed a chip design yourself, but referring to the steps as you proceed on your own chip design should help guide you. As a result of the bottom-up organization, some topics may be

broken up in unexpected ways. For example, placement and routing are not treated as a single subject, but separately at each level of abstraction: transistor, cell, and floor plan. In many instances I purposely tried to juxtapose topics in unexpected ways to encourage new ways of thinking about their interrelationships.

This book is designed to emphasize several topics that are essential to the practice of VLSI design as a system design discipline:

- **A systematic design methodology reaching from circuits to architecture.** Modern logic design includes more than the traditional topics of adder design and two-level minimization—register-transfer design, scheduling, and allocation are all essential tools for the design of complex digital systems. Circuit and layout design tell us which logic and architectural designs make the most sense for CMOS VLSI.
- **Emphasis on top-down design starting from high-level models.** While no high-performance chip can be designed completely top-down, it is excellent discipline to start from a complete (hopefully executable) description of what the chip is to do; a number of experts estimate that half the application-specific ICs designed execute their delivery tests but don't work in their target system because the designer didn't work from a complete specification.
- **Testing and design-for-testability.** Today's customers demand both high quality and short design turnaround. Every designer must understand how chips are tested and what makes them hard to test. Relatively small changes to the architecture can make a chip drastically easier to test, while a poorly designed architecture cannot be adequately tested by even the best testing engineer.
- **Design algorithms.** We must use analysis and synthesis tools to design almost any type of chip: large chips, to be able to complete them at all; relatively small ASICs, to meet performance and time-to-market goals. Making the best use of those tools requires understanding how the tools work and exactly what design problem they are intended to solve.

The design methodologies described in this book make heavy use of computer-aided design (CAD) tools of all varieties: synthesis and analysis; layout, circuit, logic, and architecture design. CAD is more than a collection of programs. CAD is a way of thinking, a way of life, like Zen. CAD's greatest contribution to design is breaking the process up into manageable steps. That is a conceptual advance you can apply with no computer in sight. A designer can—and should—formulate a narrow problem and

apply well-understood methods to solve that problem. Whether the designer uses CAD tools or solves the problem by hand is much less important than the fact that the chip design isn't a jumble of vaguely competing concerns but a well-understood set of tasks.

I have explicitly avoided talking about the operation of particular CAD tools. Different people have different tools available to them and a textbook should not be a user's guide. More importantly, the details of how a particular program works are a diversion—what counts is the underlying problem formulations used to define the problem and the algorithms used to solve them. Many CAD algorithms are relatively intuitive and I have tried to walk through examples to show how you can think like a CAD algorithm. Some of the less intuitive CAD algorithms have been relegated to a separate chapter; understanding these algorithms helps explain what the tool does, but isn't directly important to manual design.

Both the practicing professional and the advanced undergraduate or graduate student should benefit from this book. Students will probably undertake their most complex logic design project to date in a VLSI class. For a student, the most rewarding aspect of a VLSI design class is to put together previously-learned basics on circuit, logic, and architecture design to understand the tradeoffs between the different levels of abstraction. Professionals who either practice VLSI design or develop VLSI CAD tools can use this book to brush up on parts of the design process with which they have less-frequent involvement. Doing a truly good job of each step of design requires a solid understanding of the big picture.

A number of people have improved this book through their criticism. The students of COS/ELE 420 at Princeton University have been both patient and enthusiastic. Profs. C.-K. Cheng, Andrea La Paugh, Miriam Leeser, and John “Wild Man” Nestor all used drafts in their classes and gave me valuable feedback. Profs. Giovanni De Micheli, Steven Johnson, Sharad Malik, Robert Rutenbar, and James Sturm also gave me detailed and important advice after struggling through early drafts. Profs. Malik and Niraj Jha also patiently answered my questions about the literature. Any errors in this book are, of course, my own.

Thanks to Dr. Mark Pinto and David Boulin of AT&T for the transistor cross section photo and to Chong Hao and Dr. Michael Tong of AT&T for the ASIC photo. Dr. Robert Mathews, formerly of Stanford University and now of Performance Processors, indoctrinated me in pedagogical methods for VLSI design from an impressionable age. John Redford of DEC supplied many of the colorful terms in the lexicon.

Wayne Wolf

Princeton, New Jersey

Table of Contents

Preface to the Third Edition xv

Preface to the Second Edition xvii

Preface xix

1 Digital Systems and VLSI 1

- 1.1 Why Design Integrated Circuits? 1
- 1.2 Integrated Circuit Manufacturing 4
 - 1.2.1 Technology 4
 - 1.2.2 Economics 6
- 1.3 CMOS Technology 15
 - 1.3.1 CMOS Circuit Techniques 15
 - 1.3.2 Power Consumption 16
 - 1.3.3 Design and Testability 17
- 1.4 Integrated Circuit Design Techniques 18
 - 1.4.1 Hierarchical Design 19
 - 1.4.2 Design Abstraction 22
 - 1.4.3 Computer-Aided Design 28
- 1.5 A Look into the Future 30
- 1.6 Summary 31
- 1.7 References 31
- 1.8 Problems 32

2 Transistors and Layout 33

- 2.1 Introduction 33
- 2.2 Fabrication Processes 34
 - 2.2.1 Overview 34
 - 2.2.2 Fabrication Steps 37
- 2.3 Transistors 40
 - 2.3.1 Structure of the Transistor 40

2.3.2	A Simple Transistor Model	45
2.3.3	Transistor Parasitics	48
2.3.4	Tub Ties and Latchup	50
2.3.5	Advanced Transistor Characteristics	53
2.3.6	Leakage and Subthreshold Currents	60
2.3.7	Advanced Transistor Structures	61
2.3.8	Spice Models	61
2.4	Wires and Vias	62
2.4.1	Wire Parasitics	65
2.4.2	Skin Effect in Copper Interconnect	72
2.5	Design Rules	74
2.5.1	Fabrication Errors	75
2.5.2	Scalable Design Rules	77
2.5.3	SCMOS Design Rules	79
2.5.4	Typical Process Parameters	83
2.6	Layout Design and Tools	83
2.6.1	Layouts for Circuits	83
2.6.2	Stick Diagrams	88
2.6.3	Hierarchical Stick Diagrams	90
2.6.4	Layout Design and Analysis Tools	95
2.6.5	Automatic Layout	99
2.7	References	102
2.8	Problems	102

3 Logic Gates 111

3.1	Introduction	111
3.2	Combinational Logic Functions	112
3.3	Static Complementary Gates	114
3.3.1	Gate Structures	115
3.3.2	Basic Gate Layouts	118
3.3.3	Logic Levels	121
3.3.4	Delay and Transition Time	126
3.3.5	Power Consumption	135
3.3.6	The Speed-Power Product	138
3.3.7	Layout and Parasitics	139
3.3.8	Driving Large Loads	142
3.4	Switch Logic	143
3.5	Alternative Gate Circuits	144
3.5.1	Pseudo-nMOS Logic	145

- 3.5.2 DCVS Logic 147
 - 3.5.3 Domino Logic 149
- 3.6 Low-Power Gates 154
- 3.7 Delay Through Resistive Interconnect 160
 - 3.7.1 Delay Through an RC Transmission Line 160
 - 3.7.2 Delay Through RC Trees 163
 - 3.7.3 Buffer Insertion in RC Transmission Lines 167
 - 3.7.4 Crosstalk Between RC Wires 169
- 3.8 Delay Through Inductive Interconnect 172
 - 3.8.1 RLC Basics 173
 - 3.8.2 RLC Transmission Line Delay 174
 - 3.8.3 Buffer Insertion in RLC Transmission Lines 175
- 3.9 References 177
- 3.10 Problems 179

4 Combinational Logic Networks 185

- 4.1 Introduction 185
- 4.2 Standard Cell-Based Layout 186
 - 4.2.1 Single-Row Layout Design 187
 - 4.2.2 Standard Cell Layout Design 196
- 4.3 Simulation 198
- 4.4 Combinational Network Delay 202
 - 4.4.1 Fanout 203
 - 4.4.2 Path Delay 204
 - 4.4.3 Transistor Sizing 209
 - 4.4.4 Automated Logic Optimization 218
- 4.5 Logic and Interconnect Design 219
 - 4.5.1 Delay Modeling 220
 - 4.5.2 Wire Sizing 221
 - 4.5.3 Buffer Insertion 222
 - 4.5.4 Crosstalk Minimization 224
- 4.6 Power Optimization 229
 - 4.6.1 Power Analysis 229
- 4.7 Switch Logic Networks 233
- 4.8 Combinational Logic Testing 237
 - 4.8.1 Gate Testing 239
 - 4.8.2 Combinational Network Testing 242
- 4.9 References 244

4.10 Problems 244

5 Sequential Machines 249

- 5.1 Introduction 249
- 5.2 Latches and Flip-Flops 250
 - 5.2.1 Categories of Memory Elements 250
 - 5.2.2 Latches 252
 - 5.2.3 Flip-Flops 259
- 5.3 Sequential Systems and Clocking Disciplines 260
 - 5.3.1 One-Phase Systems for Flip-Flops 263
 - 5.3.2 Two-Phase Systems for Latches 265
 - 5.3.3 Advanced Clocking Analysis 273
 - 5.3.4 Clock Generation 280
- 5.4 Sequential System Design 281
 - 5.4.1 Structural Specification of Sequential Machines 281
 - 5.4.2 State Transition Graphs and Tables 283
 - 5.4.3 State Assignment 292
- 5.5 Power Optimization 298
- 5.6 Design Validation 299
- 5.7 Sequential Testing 301
- 5.8 References 308
- 5.9 Problems 308

6 Subsystem Design 311

- 6.1 Introduction 311
- 6.2 Subsystem Design Principles 314
 - 6.2.1 Pipelining 314
 - 6.2.2 Data Paths 316
- 6.3 Combinational Shifters 319
- 6.4 Adders 322
- 6.5 ALUs 329
- 6.6 Multipliers 330
- 6.7 High-Density Memory 339
 - 6.7.1 ROM 341
 - 6.7.2 Static RAM 343
 - 6.7.3 The Three-Transistor Dynamic RAM 347

6.7.4 The One-Transistor Dynamic RAM	348
6.8 Field-Programmable Gate Arrays	351
6.9 Programmable Logic Arrays	352
6.10 References	356
6.11 Problems	356
7 Floorplanning	359
7.1 Introduction	359
7.2 Floorplanning Methods	360
7.2.1 Block Placement and Channel Definition	364
7.2.2 Global Routing	370
7.2.3 Switchbox Routing	372
7.2.4 Power Distribution	373
7.2.5 Clock Distribution	376
7.2.6 Floorplanning Tips	381
7.2.7 Design Validation	382
7.3 Off-Chip Connections	383
7.3.1 Packages	383
7.3.2 The I/O Architecture	387
7.3.3 Pad Design	388
7.4 References	391
7.5 Problems	393
8 Architecture Design	399
8.1 Introduction	399
8.2 Hardware Description Languages	400
8.2.1 Modeling with Hardware Description Languages	400
8.2.2 VHDL	405
8.2.3 Verilog	414
8.2.4 C as a Hardware Description Language	421
8.3 Register-Transfer Design	422
8.3.1 Data Path-Controller Architectures	424
8.3.2 ASM Chart Design	425
8.4 High-Level Synthesis	434
8.4.1 Functional Modeling Programs	436
8.4.2 Data	437
8.4.3 Control	447