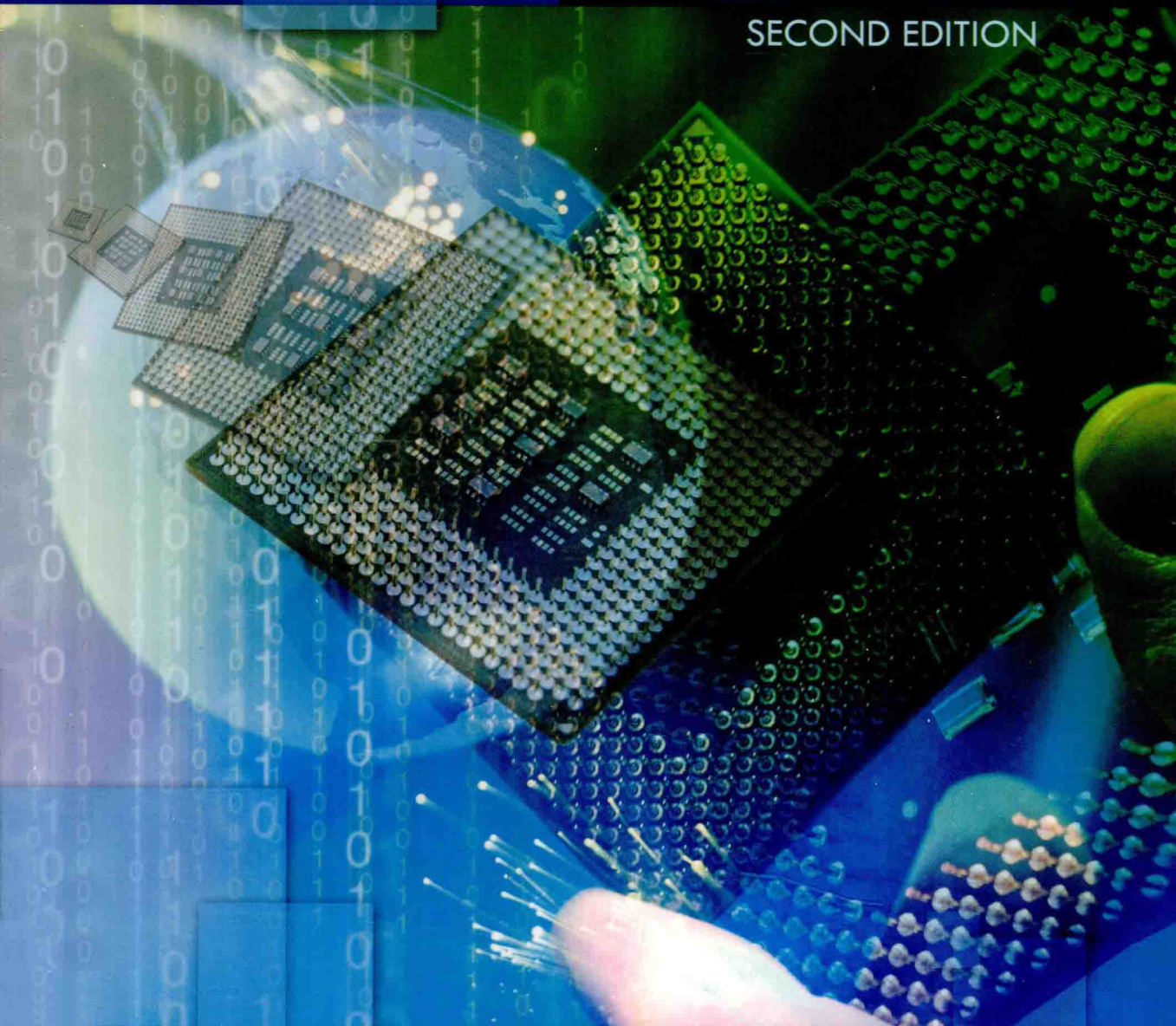


Digital Design

with RTL Design, VHDL, and Verilog

SECOND EDITION



Frank Vahid

Digital Design

with RTL Design, VHDL, and Verilog

SECOND EDITION

FRANK VAHID

University of California, Riverside



 **WILEY**

A John Wiley & Sons, Inc., Publication

*To my family, Amy, Eric, Kelsi, and Maya;
and to all engineers who apply their skills
to improve the human condition.*

VP and EXECUTIVE PUBLISHER	Don Fowley
ASSOCIATE PUBLISHER	Dan Sayre
EDITORIAL ASSISTANT	Katie Singleton
SENIOR PRODUCTION MANAGER	Micheline Frederick
SENIOR PRODUCTION EDITOR	Kerry Weinstein
EXECUTIVE MARKETING MANAGER	Christopher Ruel
COVER DESIGNER	Jim O'Shea
MEDIA EDITOR	Lauren Sapira
COVER PHOTO	Comstock Images/Getty Images, Inc.;
	iStockphoto

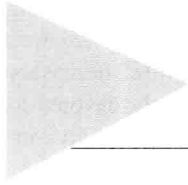
This book was set in 10/12 Times Roman by Frank Vahid. The text and cover were printed by Donnelley/Crawfordsville.

Copyright © 2011, 2007 John Wiley & Sons, Inc. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except as permitted under Sections 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, Inc. 222 Rosewood Drive, Danvers, MA 01923, website www.copyright.com. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030-5774, (201)748-6011, fax (201)748-6008, website <http://www.wiley.com/go/permissions>.

Evaluation copies are provided to qualified academics and professionals for review purposes only, for use in their courses during the next academic year. These copies are licensed and may not be sold or transferred to a third party. Upon completion of the review period, please return the evaluation copy to Wiley. Return instructions and a free of charge return shipping label are available at www.wiley.com/go/returnlabel. Outside of the United States, please contact your local representative.

ISBN 978-0-470-53108-2

Printed in the United States of America
10 9 8 7 6 5 4 3



Preface

TO STUDENTS ABOUT TO STUDY DIGITAL DESIGN

Digital circuits form the basis of general-purpose computers and also of special-purpose devices like cell phones or video game consoles. Digital circuits are dramatically changing the world. Studying digital design not only gives you the confidence that comes with fundamentally understanding how digital circuits work, but also introduces you to an exciting possible career direction. This statement applies regardless of whether your major is electrical engineering, computer engineering, or computer science; in fact, the need for digital designers with strong computer science skills continues to increase. I hope you find digital design to be as interesting, exciting, and useful as I do.

Throughout this book, I have tried not only to introduce concepts in the most intuitive manner, but I have also tried to show how those concepts can be applied to real-world systems, such as pacemakers, ultrasound machines, printers, automobiles, or cell phones.

Young and capable engineering students sometimes leave their major, claiming they want a job that is more “people-oriented.” Yet we need those people-oriented students more than ever, as engineering jobs are increasingly people-oriented, in several ways. First, engineers usually work in *tightly integrated groups* involving numerous other engineers, rather than “sitting alone in front of a computer all day” as many students believe. Second, engineers often work *directly with customers*, such as business people, doctors, lawyers, or government officials, and must therefore be able to connect with those customers. Third, and in my opinion most importantly, *engineers build things that dramatically impact people’s lives*. Needed are engineers who combine their enthusiasm, creativity, and innovation with their solid engineering skills to invent and build new products that improve people’s quality of life.

I have included “Designer Profiles” at the end of most chapters. The designers, whose experience levels vary from just a year to several decades, and whose companies range from small to huge, share with you their experiences, insights, and advice. You will notice how commonly they discuss the people aspects of their jobs. You may also notice their enthusiasm and passion for their jobs.

TO INSTRUCTORS OF DIGITAL DESIGN

This book has several key features that distinguish it from existing digital design books.

- *RTL design*. In the 1970s/1980s, chips had hundreds or thousands of gates, and hence digital design emphasized gate-level minimization. Today’s chips hold millions of gates, and modern design is thus dominated by *register-transfer level (RTL)* design. A student exposed to RTL design in a first course will have a more relevant view of the modern digital design field, leading not only to a better appreciation of modern computers and other digital devices, but to a more accurate

understanding of careers involving digital design. Such an accurate understanding is critical to attract computing majors to digital design careers, and to create a cadre of engineers with the comfort in both “software” and “hardware” necessary in modern embedded computing system design. Chapter 5 is entirely devoted to RTL design and is one of the only concise introductions to basic RTL design concepts and examples to be found anywhere.

- *Comprehensive and flexible HDL coverage.* HDLs are an important part of modern digital design, but they must be introduced carefully, such that students continue to learn fundamental digital design concepts along with the appropriate role of HDLs. Thus, this book covers HDLs in a separate chapter (Chapter 9), whose subsections each correspond to an earlier chapter, such that Section 9.2 can directly follow Chapter 2, Section 9.3 can follow Chapter 3, Section 9.4 can follow Chapter 4, and Section 9.5 can follow Chapter 5. This approach provides instructors the flexibility to cover HDLs in the latter part of a course only, or intermixed throughout, but in either case clearly showing students that HDLs are a mechanism for supporting digital design while being distinct from basic concepts. Furthermore, rather than the book choosing just one of the popular languages—VHDL, Verilog, or the relatively new SystemC—the book provides equal coverage of all three of those HDLs. We use our extensive experience in synthesis with commercial tools to create HDL descriptions well suited for synthesis, in addition to being suitable for simulation. Furthermore, for courses that cover HDLs in more depth or that have a follow-up course emphasizing more HDL design, two low-cost books have been created (one for VHDL, one for Verilog) specifically to accompany this book. Those HDL-introduction books use the same chapter structure and examples from this textbook, eliminating the common situation of students struggling to correlate their distinct and sometimes contradicting HDL book and digital design book. Our HDL-introduction books discuss language, simulation, and testing concepts in more depth than digital design books that incorporate HDL coverage, providing numerous HDL examples. The HDL books are also usable by themselves for HDL learning or reference. The HDL-introduction books improve upon the plethora of existing HDL books by emphasizing use of the language for real design, clearly distinguishing HDL use for synthesis from HDL use for testing, and by using extensive examples and figures throughout to illustrate concepts. The HDL-introduction books also come with complete PowerPoint slides that use graphics and animations to serve as an easy-to-use tutorial on the HDL.
- *Top-down design versus optimization.* Digital design and logic-size optimization were inseparably intertwined in the 1970s/1980s’ small-capacity chip era. This book cleanly distinguishes design concepts from optimization concepts by using a distinct chapter for optimization (Chapter 6), expanding optimization coverage to also include tradeoffs and to include RTL topics. Nevertheless, the book provides an instructor maximum flexibility to introduce optimization at the times and to the extent desired by the instructor. In particular, the optimization chapter’s subsections each correspond directly to one earlier chapter, such that Section 6.2 can directly follow Chapter 2, Section 6.3 can follow Chapter 3, Section 6.4 can

follow Chapter 4, and Section 6.5 can follow Chapter 5. The book also emphasizes the modern approach of top-down design, involving capturing desired behavior and then converting to a circuit. At the same time, this book, like other books, uses a concrete bottom-up approach, starting from transistors, and building incrementally up to gates, flip-flops, registers, controllers, datapath components, and RTL.

- *Extensive use of applied examples and figures.* After describing a new concept and providing basic examples, the book provides examples that apply the concept to applications recognizable to a student, like a “seat belt unfastened” warning system, a computerized checkerboard game, a color printer, or a digital video camera. Furthermore, the end of most chapters includes a product profile, intended to give students an even broader view of the applicability of the concepts, and to introduce clever application-specific concepts the students may find interesting—like the idea of beamforming in an ultrasound machine or of filtering in a cellular phone. The book extensively uses figures to illustrate concepts; it contains over 600 figures.
- *Learning through discovery.* The book emphasizes understanding the need for new concepts, which not only helps students learn and remember the concepts, but develops reasoning skills that can apply the concepts to other domains. For example, rather than just defining a carry-lookahead adder, the book shows intuitive but inefficient approaches to building a faster adder, eventually solving the inefficiencies and leading to (“discovering”) the carry-lookahead design.
- *Introduction to FPGAs.* The book includes a fully bottom-up introduction to FPGAs, showing students concretely how a circuit can be converted into a bit-stream that programs the individual lookup tables, switch matrices, and other programmable components in an FPGA. This concrete introduction eliminates the mystery of the increasingly common FPGA devices.
- *Author-created graphical animated PowerPoint slides.* A rich set of PowerPoint slides is available to instructors. The slides were created by the textbook’s author, resulting in consistency of perspective and emphasis between the slides and book. The slides are designed to be a truly effective teaching tool for the instructor. Most slides are graphical, avoiding slides consisting of just bulleted lists of text. The slides make extensive use of animation, where appropriate, to gradually unveil concepts or build up circuits, yet animated slides are carefully created so they can be printed out and understood. Nearly every figure, concept, and example from this book is included in the set of almost 500 slides.
- *Complete solutions manual.* Instructors may obtain a complete solutions manual (about 200 pages) containing solutions to every end-of-chapter exercise in this book. The manual extensively utilizes figures to illustrate solutions.

Many of the above features can be seen in the sample book materials available at <http://www.ddvahid.com>. Materials are available to instructors via the instructors site.

The second edition of this book includes a rewrite of the RTL design introduction in Chapter 5 to more intuitively introduce the subject, a further emphasis of top-down

design (capture and convert) throughout Chapters 2–5, and improvements and additions to the descriptions, examples, and exercises in all chapters of the book.

HOW TO USE THIS BOOK

This book was designed to allow flexibility for instructors to choose among the most common approaches of material coverage. We describe several approaches below.

RTL-Focused Approach

An RTL-focused approach would simply cover the first 6 chapters in order:

1. Introduction (Chapter 1)
2. Combinational logic design (Chapter 2)
3. Sequential logic design (Chapter 3)
4. Combinational and sequential component design (Chapter 4)
5. RTL design (Chapter 5)
6. Optimizations and tradeoffs (Chapter 6), to the extent desired
7. Physical implementation (Chapter 7) and/or processor design (Chapter 8), to the extent desired

We think this is a great way to order the material, resulting in students doing interesting RTL designs in about seven weeks. HDLs can be introduced at the end if time permits, or left for a second course on digital design (as done at UCR), or covered immediately after each chapter—all three approaches are common.

Traditional Approach with Some Reordering

This book can be readily used in a traditional approach that introduces optimization along with basic design, with a slight difference from the traditional approach being the swapping of coverage of combinational components and sequential logic, as follows:

1. Introduction (Chapter 1)
2. Combinational logic design (Chapter 2) followed by combinational logic optimization (Section 6.2)
3. Sequential logic design (Chapter 3) followed by sequential logic optimization (Section 6.3)
4. Combinational and sequential component design (Chapter 4) followed by component tradeoffs (Section 6.4)
5. RTL design (Chapter 5) to the extent desired, followed by RTL optimization/tradeoffs (Section 6.5)
6. Physical implementation (Chapter 7) and/or processor design (Chapter 8), to the extent desired

This is a reasonable and effective approach, completing all discussion of one topic (e.g., FSM design as well as optimization) before moving on to the next topic. The reordering from a traditional approach introduces basic sequential design (FSMs and controllers) before combinational components (e.g., adders, comparators, etc.). Such reordering may lead into RTL design more naturally than a traditional approach, following instead an

approach of increasing abstraction rather than the traditional approach that separates combinational and sequential design. HDLs can again be introduced at the end, left for another course, or integrated after each chapter. This approach could also be used as an intermediary step when migrating from a traditional approach to an RTL approach. Migrating might involve gradually postponing the Chapter 6 sections—for example, covering Chapters 2 and 3, and then Sections 6.2 and 6.3, before moving on to Chapter 4.

Traditional Approach

This book could also be used in a traditional approach, as follows:

1. Introduction (Chapter 1)
2. Combinational logic design (Chapter 2) followed by combinational logic optimization (Section 6.2)
3. Combinational component design (Sections 4.1, 4.3–4.8) followed by combinational component tradeoffs (Section 6.4—Faster Adders)
4. Sequential logic design (Chapter 3) followed by sequential logic optimization (Section 6.3)
5. Sequential component design (Sections 4.9, 4.10) followed by sequential component tradeoffs (Section 6.4—Smaller Multiplier)
6. RTL design (Chapter 5) to the extent desired, followed by RTL optimization/tradeoffs (Section 6.5)
7. Physical implementation (Chapter 7) and/or processor design (Chapter 8), to the extent desired.

Coverage of the first five topics has been the most widespread approach during the past two decades, with the above adding RTL design towards the end of the approach. Although the emphasized distinction between combinational and sequential design may no longer be relevant in the era of RTL design (where both types of design are intermixed), some people believe that such distinction makes for an easier learning path. HDLs can be included at the end, left for a later course, or integrated throughout.

ACKNOWLEDGEMENTS

Many people and organizations contributed to the making of this book.

- Staff members at John Wiley and Sons Publishers extensively supported the book’s development. Dan Sayre inspired and oversaw the development of the second edition, and Micheline Frederick oversaw production. Kelly Applegate and Foti Kutil from Publication Services assisted greatly with composition and formatting of the second edition. Bill Zobrist supported my earlier “Embedded System Design” book, and motivated me to write the first edition of the book.
- Ryan Mannion contributed many items, including the appendices, numerous examples and exercises, several subsections, the complete exercise solutions manual, fact-checking, extensive proofreading, tremendous assistance during production, help with the slides, plenty of ideas during discussions, and much more.

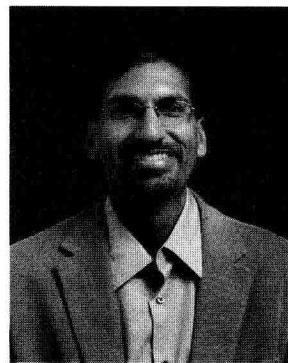
- Roman Lysecky developed numerous examples and exercises, contributed most of the content of the HDL chapter, and co-authored our accompanying HDL-introduction books. Scott Sirowy contributed some of the HDL code for the second edition. Francesca Perkins did extensive proofreading of the second edition. Scott Sirowy, David Sheldon, and Bailey Miller helped with proofreading also.
- Numerous reviewers provided outstanding feedback on various versions of the book. Special thanks go to first-edition adopters who have provided great feedback, including Greg Link, Mark Brehob, Sharon Hu, Nikil Dutt, Eli Bozorgzadeh, and Jay Brockman (who has also made his lectures available on the web).
- The importance of the support provided to my research and teaching career by the National Science Foundation cannot be overstated.

ABOUT THE COVER

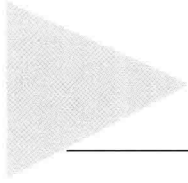
The cover's image of shrinking chips is more than just a nice visual; the image graphically depicts the amazing real-life phenomenon of digital circuits ("computer chips") shrinking in size by about one half every 18 months, for several decades now, a phenomenon referred to as Moore's Law. Such shrinking has enabled incredibly powerful computing circuits to fit inside tiny devices, like modern cell phones, medical devices, and portable video games.

ABOUT THE AUTHOR

Frank Vahid is a Professor of Computer Science and Engineering at the University of California, Riverside. He received his bachelor's degree in electrical engineering from the University of Illinois at Urbana-Champaign, and his master's and doctoral degrees in computer science from the University of California, Irvine. He has worked for Hewlett Packard and AMCC, and has consulted for Motorola, NEC, Atmel, Cardinal Health, and several other engineering firms. He is the inventor on three U.S. patents, has published over 150 research papers and two books on embedded systems, and helped establish the Embedded Systems Week conference. He established UCR's Computer Engineering program, and has received several UCR teaching awards. His research includes incorporating FPGAs into embedded systems, and networked sensor blocks that ordinary people can configure to monitor their surroundings.



See this book's website at <http://www.ddvahid.com> for additional book materials, for access to the publisher's book website and instructor materials, or to submit comments, corrections, or suggestions.



Reviewers and Evaluators

Rehab Abdel-Kader
Otmane Ait Mohamed
Hussain Al-Asaad
Rocio Alba-Flores
Bassem Alhalabi
Zekeriya Aliyazicioglu
Vishal Anand
Bevan Baas
Noni Bohonak
Don Bouldin
David Bourner
Elaheh Bozorgzadeh
Frank Candocia
Ralph Carestia
Rajan M. Chandra
Ghulam Chaudhry
Michael Chelian
Russell Clark
James Conrad
Kevan Croteau
Sanjoy Das
James Davis
Edward Doering
Travis Doom
Jim Duckworth
Nikil Dutt
Dennis Fairclough
Paul D. Franzon
Subra Ganesan
Zane Gastineau
J. David Gillanders
Clay Gloster
Ardian Greca
Eric Hansen
Bruce A. Harvey
John P. Hayes
Michael Helm
William Hoff
Erh-Wen Hu
Xiaobo Sharon Hu
Baback Izadi

Georgia Southern University
Concordia University
University of California, Davis
University of Minnesota, Duluth
Florida Atlantic University
California Polytechnic State University, Pomona
SUNY Brockport
University of California, Davis
University of South Carolina, Lancaster
University of Tennessee
University of Maryland Baltimore County
University of California, Irvine
Florida International University
Oregon Institute of Technology
California Polytechnic State University, Pomona
University of Missouri, Kansas City
California State University, Long Beach
Saginaw Valley State University
University of North Carolina, Charlotte
Francis Marion University
Kansas State University
University of South Carolina
Rose-Hulman Institute of Technology
Wright State University
Worcester Polytechnic Institute
University of California, Irvine
Utah Valley State College
North Carolina State University
Oakland University
Harding University
Arkansas State University
Howard University
Georgia Southern University
Dartmouth College
FAMU-FSU College of Engineering
University of Michigan
Texas Tech University
Colorado School of Mines
William Paterson University of New Jersey
University of Notre Dame
SUNY New Paltz

Jeff Jackson	University of Alabama
Anura Jayasumana	Colorado State University
Bruce Johnson	University of Nevada, Reno
Richard Johnston	Lawrence Technological University
Rajiv Kapadia	Minnesota State University, Mankato
Bahadir Karuv	Fairleigh Dickinson University
Robert Klenke	Virginia Commonwealth University
Clint Kohl	Cedarville University
Hermann Krompholz	Texas Tech University
Timothy Kurzweg	Drexel University
Jumoke Ladeji-Osias	Morgan State University
Jeffrey Lillie	Rochester Institute of Technology
David Livingston	Virginia Military Institute
Hong Man	Stevens Institute of Technology
Gihan Mandour	Christopher Newport University
Diana Marculescu	Carnegie Mellon University
Miguel Marin	McGill University
Maryam Moussavi	California State University, Long Beach
Olfa Nasraoui	University of Memphis
Patricia Nava	University of Texas, El Paso
John Nestor	Lafayette College
Rogelio Palomera	Garcia University of Puerto Rico, Mayaguez
James Peckol	University of Washington
Witold Pedrycz	University of Alberta
Andrew Perry	Springfield College
Denis Popel	Baker University
Tariq Qayyum	California Polytechnic State University, Pomona
Gang Qu	University of Maryland
Mihaela Radu	Rose-Hulman Institute of Technology
Suresh Rai	Louisiana State University, Baton Rouge
William Reid	Clemson University
Musoke Sendaula	Temple University
Martha Sloan	Michigan Technological University
Scott Smith	Boise State University
Gary Spivey	George Fox University
Larry Stephens	University of South Carolina
James Stine	Illinois Institute of Technology
Philip Swain	Purdue University
Shannon Tauro	University of California, Irvine
Carlos Tavora	Gonzaga University
Marc Timmerman	Oregon Institute of Technology
Hariharan Vijayaraghavan	University of Kansas
Bin Wang	Wright State University
M. Chris Wernicki	New York Institute of Technology
Shanchieh Yang	Rochester Institute of Technology
Henry Yeh	California State University, Long Beach
Kathleen Whitehorn	Colorado School of Mines
Naeem Zaman	San Joaquin Delta College



Contents

Preface ix

To Students About To Study Digital Design ix

To Instructors of Digital Design ix

How to Use This Book xii

RTL-Focused Approach xii

Traditional Approach with Some Reordering xii

Traditional Approach xiii

Acknowledgements xiii

About the Cover xiv

About the Author xiv

Reviewers and Evaluators xv

▶ CHAPTER 1

Introduction 1

1.1 Digital Systems in the World Around Us 1

1.2 The World of Digital Systems 4

Digital versus Analog 4

Digital Encodings and Binary Numbers—0s and 1s 9

1.3 Implementing Digital Systems:

Microprocessors versus Digital Circuits 22

Software on Microprocessors: The Digital

Workhorse 22

Digital Design—When Microprocessors Aren't Good Enough 26

1.4 About this Book 28

1.5 Exercises 29

▶ CHAPTER 2

Combinational Logic Design 35

2.1 Introduction 35

2.2 Switches 36

Electronics 101 36

The Amazing Shrinking Switch 37

2.3 The CMOS Transistor 40

2.4 Boolean Logic Gates—Building Blocks for Digital Circuits 43

Boolean Algebra and its Relation to Digital Circuits 43

AND, OR, & NOT Gates 46

Building Simple Circuits Using Gates 49

2.5 Boolean Algebra 52

Notation and Terminology 53

Some Properties of Boolean Algebra 55

Complementing a Function 60

2.6 Representations of Boolean Functions 61

Equations 62

Circuits 62

Truth Tables 62

Converting among Boolean Function

Representations 64

Standard Representation and Canonical Form 68

Multiple-Output Combinational Circuits 71

2.7 Combinational Logic Design Process 73

2.8 More Gates 80

NAND & NOR 80

XOR & XNOR 81

Interesting Uses of these Additional Gates 82

Completeness of NAND and of NOR 82

Number of Possible Logic Gates 83

2.9 Decoders and Muxes 84

Decoders 84

Multiplexers (Muxes) 86

2.10 Additional Considerations 91

Nonideal Gate Behavior—Delay 91

Active Low Inputs 92

Demultiplexers and Encoders 93

Schematic Capture and Simulation 93

2.11 Combinational Logic Optimizations and Tradeoffs (See Section 6.2) 95

2.12 Combinational Logic Description Using Hardware Description Languages (See Section 9.2) 95

2.13 Chapter Summary 96

2.14 Exercises 96

► **CHAPTER 3****Sequential Logic Design: Controllers 105**

- 3.1** Introduction 105
- 3.2** Storing One Bit—Flip-Flops 106
 - Feedback—The Basic Storage Method 106
 - Basic SR Latch 107
 - Level-Sensitive SR Latch 111
 - Level-Sensitive D Latch—A Basic Bit Store 112
 - Edge-Triggered D Flip-Flop—A Robust Bit Store 113
 - Clocks and Synchronous Circuits 117
 - Basic Register—Storing Multiple Bits 120
- 3.3** Finite-State Machines (FSMs) 122
 - Mathematical Formalism for Sequential Behavior—FSMs 124
 - How to Capture Desired System Behavior as an FSM 129
- 3.4** Controller Design 132
 - Standard Controller Architecture for Implementing an FSM as a Sequential Circuit 132
 - Controller (Sequential Logic) Design Process 133
 - Converting a Circuit to an FSM (Reverse Engineering) 140
 - Common Mistakes when Capturing FSMs 142
 - FSM and Controller Conventions 145
- 3.5** More on Flip-Flops and Controllers 146
 - Non-Ideal Flip-Flop Behavior 146
 - Flip-Flop Reset and Set Inputs 149
 - Initial State of a Controller 150
 - Non-Ideal Controller Behavior: Output Glitches 151
- 3.6** Sequential Logic Optimizations and Tradeoffs (See Section 6.3) 153
- 3.7** Sequential Logic Description Using Hardware Description Languages (See Section 9.3) 153
- 3.8** Product Profile—Pacemaker 153
- 3.9** Chapter Summary 156
- 3.10** Exercises 157

► **CHAPTER 4****Datapath Components 167**

- 4.1** Introduction 167
- 4.2** Registers 168
 - Parallel-Load Register 168
 - Shift Register 173
 - Multifunction Registers 175
 - Register Design Process 179
- 4.3** Adders 181
 - Adder—Carry-Ripple Style 183
- 4.4** Comparators 191
 - Equality (Identity) Comparator 191
 - Magnitude Comparator—Carry-Ripple Style 192
- 4.5** Multiplier—Array-Style 195
- 4.6** Subtractors and Signed Numbers 196
 - Subtractor for Positive Numbers Only 196
 - Representing Negative Numbers: Two’s Complement Representation 200
 - Building a Subtractor Using an Adder and Two’s Complement 203
 - Detecting Overflow 205
- 4.7** Arithmetic-Logic Units—ALUs 207
- 4.8** Shifters 210
 - Simple Shifters 211
 - Barrel Shifter 214
- 4.9** Counters and Timers 215
 - Up-Counter 216
 - Up/Down-Counter 217
 - Counter with Load 218
 - Timers 222
- 4.10** Register Files 225
- 4.11** Datapath Component Tradeoffs (See Section 6.4) 230
- 4.12** Datapath Component Description Using Hardware Description Languages (See Section 9.4) 230
- 4.13** Product Profile: An Ultrasound Machine 230
 - Functional Overview 231
 - Digital Circuits in an Ultrasound Machine’s Beamformer 234
 - Future Challenges in Ultrasound 237
- 4.14** Chapter Summary 237
- 4.15** Exercises 238

▶ **CHAPTER 5****Register-Transfer Level (RTL) Design 247**

- 5.1** Introduction 247
- 5.2** High-Level State Machines 248
- 5.3** RTL Design Process 255
 - Step 2A—Creating a Datapath using Components from a Library 259
 - Step 2B—Connecting the Datapath to a Controller 262
 - Step 2C—Deriving the Controller’s FSM 263
- 5.4** More RTL Design 264
 - Additional Datapath Components for the Library 264
 - RTL Design Involving Register Files or Memories 265
 - RTL Design Pitfall Involving Storage Updates 271
 - RTL Design Involving a Timer 272
 - A Data-Dominated RTL Design Example 275
- 5.5** Determining Clock Frequency 278
- 5.6** Behavioral-Level Design: C to Gates (Optional) 281
- 5.7** Memory Components 285
 - Random Access Memory (RAM) 286
 - Bit Storage in a RAM 288
 - Using a RAM 290
 - Read-Only Memory (ROM) 292
 - ROM Types 294
 - Using a ROM 297
 - The Blurring of the Distinction between RAM and ROM 299
- 5.8** Queues (FIFOs) 299
- 5.9** Multiple Processors 303
- 5.10** Hierarchy—A Key Design Concept 305
 - Managing Complexity 305
 - Abstraction 306
 - Composing a Larger Component from Smaller Versions of the Same Component 307
- 5.11** RTL Design Optimizations and Tradeoffs (See Section 6.5) 309
- 5.12** RTL Design Using Hardware Description Languages (See Section 9.5) 310

5.13 Product Profile: Cell Phone 310

- Cells and Basestations 310
- How Cellular Phone Calls Work 311
- Inside a Cell Phone 312

5.14 Chapter Summary 316**5.15** Exercises 317▶ **CHAPTER 6****Optimizations and Tradeoffs 325**

- 6.1** Introduction 325
- 6.2** Combinational Logic Optimizations and Tradeoffs 327
 - Two-Level Size Optimization Using Algebraic Methods 327
 - A Visual Method for Two-Level Size Optimization—K-Maps 329
 - Don’t Care Input Combinations 336
 - Automating Two-Level Logic Size Optimization 339
 - Multilevel Logic Optimization—Performance and Size Tradeoffs 348
- 6.3** Sequential Logic Optimizations and Tradeoffs 351
 - State Reduction 351
 - State Encoding 354
 - Moore versus Mealy FSMs 360
- 6.4** Datapath Component Tradeoffs 365
 - Faster Adders 365
 - Smaller Multiplier—Sequential (Shift-and-Add) Style 375
- 6.5** RTL Design Optimizations and Tradeoffs 377
 - Pipelining 377
 - Concurrency 380
 - Component Allocation 381
 - Operator Binding 382
 - Operator Scheduling 383
 - Moore versus Mealy High-Level State Machines 386
- 6.6** More on Optimizations and Tradeoffs 386
 - Serial versus Concurrent Computation 386
 - Optimizations and Tradeoffs at Higher versus Lower Levels of Design 387
 - Algorithm Selection 388
 - Power Optimization 389

- 6.7 Product Profile: Digital Video Player/
Recorder 393
 - Digital Video Overview 393
 - DVD—One Form of Digital Video Storage 393
 - MPEG-2 Video Encoding—Sending Frame
Differences Using I-, P-, and B-Frames 395
 - Transforming to the Frequency Domain for
Further Compression 396
- 6.8 Chapter Summary 402
- 6.9 Exercises 403

▶ CHAPTER 7

Physical Implementation on ICs 413

- 7.1 Introduction 413
- 7.2 Manufactured IC Types 414
 - Full-Custom Integrated Circuits 414
 - Semicustom (Application-Specific) Integrated
Circuits—ASICs 415
- 7.3 Off-the-Shelf Programmable IC
Type—FPGA 423
 - Lookup Tables 424
 - Mapping a Circuit among Multiple Lookup
Tables 426
 - Programmable Interconnects (Switch Matrices)
432
 - Configurable Logic Block 434
 - Overall FPGA Architecture 436
- 7.4 Other Off-the-Shelf IC Types 438
 - Off-the-Shelf Logic (SSI) IC 438
 - Simple Programmable Logic Device (SPLD)
441
 - Complex Programmable Logic Device (CPLD)
445
 - FPGA-to-Structured-ASIC Flows 445
- 7.5 IC Tradeoffs, Trends, and Comparisons 446
 - Tradeoffs Among IC Types 447
 - IC Technology Trend—Moore’s Law 448
 - Relative Popularity of IC Types 450
 - ASSPs 450
 - IC Types versus Processor Varieties 451
 - FPGAs alongside Microprocessors 452
- 7.6 Product Profile: Giant LED-Based Video
Display with FPGAs 453
- 7.7 Chapter Summary 457
- 7.8 Exercises 457

▶ CHAPTER 8

Programmable Processors 461

- 8.1 Introduction 461
- 8.2 Basic Architecture 462
 - Basic Datapath 462
 - Basic Control Unit 465
- 8.3 A Three-Instruction Programmable
Processor 469
 - A First Instruction Set with Three Instructions
469
 - Control Unit and Datapath for the Three-
Instruction Processor 471
- 8.4 A Six-Instruction Programmable Processor
475
 - Extending the Instruction Set 475
 - Extending the Control Unit and Datapath 476
- 8.5 Example Assembly and Machine Programs
478
- 8.6 Further Extensions to the Programmable
Processor 480
 - Instruction Set Extensions 480
 - Input/Output Extensions 481
 - Performance Extensions 481
- 8.7 Chapter Summary 482
- 8.8 Exercises 483

▶ CHAPTER 9

Hardware Description Languages 487

- 9.1 Introduction 487
- 9.2 Combinational Logic Description Using
Hardware Description Languages 489
 - Structure 489
 - Combinational Behavior 494
 - Testbenches 498
- 9.3 Sequential Logic Description Using
Hardware Description Languages 501
 - Register 501
 - Oscillator 503
 - Controllers 505
- 9.4 Datapath Component Description
Using Hardware Description Languages
509
 - Full-Adders 509

- Carry-Ripple Adders 511
- Up-Counter 514
- 9.5** RTL Design Using Hardware Description Languages 517
 - High-Level State Machine of the Laser-Based Distance Measurer 517
 - Controller and Datapath of the Laser-Based Distance Measurer 523
- 9.6** Chapter Summary 532
- 9.7** Exercises 532

▶ **APPENDIX A**

Boolean Algebras 537

- A.1** Boolean Algebra 537
- A.2** Switching Algebra 538
- A.3** Important Theorems in Boolean Algebra 540
- A.4** Other Examples of Boolean Algebras 545
- A.5** Further Readings 545

▶ **APPENDIX B**

Additional Topics in Binary Number Systems 547

- B.1** Introduction 547
- B.2** Real Number Representation 547
- B.3** Fixed Point Arithmetic 550
- B.4** Floating Point Representation 551
 - The IEEE 754-1985 Standard 552
- B.5** Exercises 556

▶ **APPENDIX C**

Extended RTL Design Example 557

- C.1** Introduction 557
- C.2** Designing the Soda Dispenser Controller 558
- C.3** Understanding the Behavior of the Soda Dispenser Controller and Datapath 562

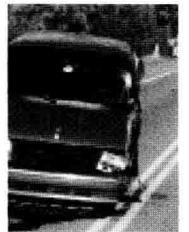
Introduction

▶ 1.1 DIGITAL SYSTEMS IN THE WORLD AROUND US

Meet Arianna. Arianna is a five-year-old girl who lives in California. She's a cheerful, outgoing kid who loves to read, play soccer, dance, and tell jokes that she makes up herself.



One day, Arianna's family was driving home from a soccer game. She was in the middle of excitedly talking about the game when suddenly the van in which she was riding was clipped by a car that had crossed over to the wrong side of the highway. Although the accident wasn't particularly bad, the impact caused a loose item from the rear of the van to project forward inside the van, striking Arianna in the back of the head. She became unconscious.



Arianna was rushed to a hospital. Doctors immediately noticed that her breathing was very weak—a common situation after a severe blow to the head—so they put her onto a ventilator, which is a medical device that assists with breathing. She had sustained brain trauma during the blow to the head, and she remained unconscious for several weeks. All her vital signs were stable, except she continued to require breathing assistance from the ventilator. Patients in such a situation sometimes recover, and sometimes they don't. When they do recover, sometimes that recovery takes many months.