PREFACE

This book is for everyone who wants to design and build real digital circuits. It is based on the idea that, in order to do this, you have to grasp the fundamentals, but at the same time you need to understand how things work in the real world. Hence, the "principles and practices" theme.

The practice of digital design has undergone a major transformation during the past 30 years, a direct result of the stunning increases in integrated-circuit speed and density over the same time period. In the past, when digital designers were building systems with thousands or at most tens of thousands of gates and flip-flops, academic courses emphasized minimization and efficient use of chip and board-level resources.

Today, a single chip can contain tens of millions of transistors and can be programmed to create a system-on-a-chip that, using the technology of the past, would have required hundreds of discrete chips containing millions of individual gates and flip-flops. Successful product development nowadays is limited more by the design team's ability to correctly and completely specify the product's detailed functions, than by the team's ability to cram all the needed circuits into a single board or chip. Thus, a modern academic program must necessarily emphasize design methodologies and software tools, including hardware description languages (HDLs), that allow very large, hierarchical designs to be accomplished by teams of designers.

On one hand, with HDLs, we see the level of abstraction for typical designs moving higher, above the level of individual gates and flip-flops. But at the same time, the increased speed and density of digital circuits at both the chip and board level is forcing many digital designers to be more competent at a lower, electrical circuit level.

The most employable and ultimately successful digital designers are skilled, or at least conversant, at both levels of abstraction. This book gives you

the opportunity to learn the basics at the high level (HDLs), at the low level (electrical circuits), and throughout the "vast middle" (gates, flip-flops, and somewhat higher-level digital-design building blocks).

Target Audience

introductory courses

The material in this book is appropriate for introductory courses on digital logic design in electrical or computer engineering or computer science curricula.

electronics concepts

Computer science students who are unfamiliar with basic electronics concepts or who just aren't interested in the electrical behavior of digital devices may wish to skip Chapter 3; the rest of the book is written to be independent of this material as much as possible. On the other hand, *anyone* with a basic electronics background who wants to get up to speed on digital electronics can do so by reading Chapter 3. In addition, students with *no* electronics background can get the basics by reading a 20-page electronics tutorial, Section Elec at DDPPonline (DDPPonline is my shorthand for this book's web-based supplemental material; more about this later).

DDPPonline

Although this book's level is introductory, it contains much more material than can be taught in a typical introductory course. Once I started writing, I found that I had many important things to say that wouldn't fit into a one-quarter course at Stanford or in a 400-page book. Therefore, I have followed my usual practice of including *everything* that I think is at least moderately important, and leaving it up to the instructor or reader to decide what is most important in a particular environment. To help these decisions along, though, I've marked the headings of *optional sections* with an asterisk. In general, these sections can be skipped without any loss of continuity in the non-optional sections that follow. Even more optional material can be found at DDPPonline.

optional sections

advanced courses
laboratory courses
fun stuff

Undoubtedly, some people will use this book in advanced courses and in laboratory courses. Advanced students will want to skip the basics and get right into the fun stuff. Once you know the basics, the most important and fun stuff in this book is in the sections on hardware description languages ABEL, VHDL, and Verilog, where you'll discover that your programming courses actually helped prepare you to design hardware.

working digital designers

Another use of this book is as a self-study reference for a working digital designer, who may be one of either of two kinds:

Novice If you're just getting started as a working digital designer, and you took a very "theoretical" logic design course in school, you should concentrate on Chapter 3, one of the HDLs in Chapter 5 and Sections 7.11–7.13, and Chapters 6, 8, and 9 to get prepared for the real world.

Old pro If you're experienced, you may not need all of the "practices" material in this book, but the principles in Chapters 2, 4, and 7 can help you organize your thinking, and the discussions there of what's important

and what's not might relieve the guilt you feel for not having used a Karnaugh map in 10 years. The examples in Chapters 6 and 8 should give you additional insights into and appreciation for a variety of design methods. Finally, the ABEL, VHDL, and Verilog language descriptions and examples sprinkled throughout Chapters 5 through 8 may serve as your first organized introduction to HDL-based design.

All readers should make good use of the comprehensive index and of the *marginal notes* throughout the text that call attention to definitions and important topics. Maybe the highlighted topics in *this* section were more marginal than important, but I just wanted to show off my text formatting system.

marginal notes
marginal pun

A FEW WORDS TO REVIEWERS AND OTHERS

Over the years, several computer-science students and others have written reviews complaining about various aspects of what was covered and how it was presented in previous editions of this book. As for the presentation, I can only apologize for my own inability to do a better job. But as for the choice of topics, and what I emphasize as important, this is *exactly* the material that I would expect a potential digital-design employee to know. Since I've hired a lot of them, you really shouldn't complain, unless you're reading this book for some reason other than getting or keeping a job.

I'm not deprecating CS students or Computer Science. I love CS and I have a lot of CS education and experience myself—I can analyze algorithms, understand the inner workings of compilers and operating systems, and write code in LISP and a couple dozen other languages along with the best of them. But the most common complaint I hear from digital-design managers today is about new grads who know how to sling code in Verilog or VHDL but have no idea of what produces a good hardware result—or even what metrics define "good"!

Digital design is not just about writing HDL code—at least, not yet. One chip-design manager tells me that the first question he asks to a job applicant who claims to know how to do HDL-based hardware design is, "How many flip-flops are in a 16-bit counter?" This isn't even a trick question, but you'd be surprised at how many applicants can't answer it!

So, if you are using this book merely to satisfy a requirement for a CS or other degree, then I apologize in advance if you should find the book to be complex, confusing, too formal, too dry, or full of jargon, or if you think that the included software is a Frisbee. All are real comments about the previous edition made by CS and other non-EE/CE reviewers!

On the other hand, if you think you may someday be working as a digital designer at any level from circuits to systems, then please do your best to learn all of the non-optional topics in this book, including either Verilog or VHDL. It shouldn't be that difficult, since the many reviewers who liked the previous edition called it precise, logical, crystal clear, easy to understand, deep but very straightforward, practical, complete, very good, very nice, enjoyable, rockin', humorous, and, I agree, a little silly or corny at times (like now, maybe). And oh, by the way, even one CS reviewer said, "I love this book."

NOT AS LONG AS IT SEEMS

A few reviewers complained about the length of the previous edition of this book. The present, printed edition is a little shorter, but when you include the material at DDPPonline, it's actually a bit longer. (That's right, the page count of the previous edition could be written in 10 bits, but the page count of this edition plus the material at DDPPonline requires 11 bits.) But please keep in mind:

- You don't have to read everything. The headings of sections and subsections that are optional for most readers are marked with an asterisk.
- You don't have to study all of the HDLs. The HDL sections are written to be independent of each other, and non-HDL sections and subsections are written to be independent of the HDL sections. So you can ignore any or all of the HDLs. But I still recommend that you learn either Verilog or VHDL.
- I asked the publisher to print this book in a larger font (11 point) than is typical for technical texts (10 point). This is easier on your eyes and mine, and it also allowed me to put in more figures and tables while still keeping most of them on the same facing pages as the referring text. (I do the page layout myself and pay a lot of attention to this.)
- Stuff written in these "boxed comments" (a.k.a. sidebars) is usually optional too.

Chapter Descriptions

What follows is a list of short descriptions of this book's nine chapters. This may remind you of the section in typical software guides, "For People Who Hate Reading Manuals." If you read this list, then maybe you don't have to read the rest of the book.

- Chapter 1 gives a few basic definitions and lays down the ground rules for what we think is and is not important in this book.
- Chapter 2 is an introduction to binary number systems and codes. Readers who are already familiar with binary number systems from a software course should still read Sections 2.10–2.13 to get an idea of how binary codes are used by hardware. Advanced students can get a nice introduction to error-detecting codes by reading Sections 2.14 and 2.15. The material in Section 2.16.1 should be read by everyone; it is used in a lot of modern systems.
- Chapter 3 describes digital circuit operation, placing primary emphasis on the external electrical characteristics of logic devices. The starting point is a basic electronics background including voltage, current, and Ohm's law; readers unfamiliar with these concepts may wish to consult Section Elec at DDPPonline. This chapter may be omitted by readers who aren't interested in how to make real circuits work, or who have the luxury of having someone else to do the dirty work.

- Chapter 4 teaches combinational logic design principles, including switching algebra and combinational-circuit analysis, synthesis, and minimization.
- Chapter 5 gives a general introduction to HDL-based design, and then has tutorials on three HDLs—ABEL, VHDL, and Verilog. You need learn only one of these languages, typically the one used in a digital-design lab that you may work in.
- Chapter 6 begins with a discussion of digital-system documentation standards, probably the most important practice for aspiring designers to start practicing. Next, this chapter introduces programmable logic devices (PLDs), focusing on their capability to realize combinational logic functions. The rest of the chapter describes commonly used combinational logic functions and applications. For each function, it describes standard MSI building blocks, ABEL programs for PLD realizations, and VHDL- and Verilog-based designs.
- Chapter 7 teaches sequential logic design principles, starting with latches and flip-flops. The main emphasis in this chapter is on the analysis and design of clocked synchronous state machines. However, for the brave and daring, the chapter includes an introduction to fundamental-mode circuits and the analysis and design of feedback sequential circuits. The chapter ends with sections on ABEL, VHDL, and Verilog features and coding styles for sequential-circuit design.
- Chapter 8 is all about the practical design of synchronous sequential circuits. This chapter focuses on commonly used functions and gives examples using MSI building blocks, ABEL and PLDs, and VHDL and Verilog. Sections 8.8 and 8.9 discuss the inevitable impediments to the ideal of fully synchronous design and address the important problem of how to live synchronously in an asynchronous world.
- Chapter 9 is an introduction to memory devices, CPLDs, and FPGAs. Memory coverage includes read-only memory and static and dynamic read/write memories from the points of view of both internal circuitry and functional behavior. The last two sections introduce CPLD and FPGA architecture.

Most of the chapters contain references, drill problems, and exercises. Drill problems are typically short-answer or turn-the-crank questions that can be answered directly based on the text material, while exercises typically require a little more thinking. The drill problems in Chapter 3 are particularly extensive and are designed to allow non-EE types to ease into this material.

Quite a lot of additional material, including just about everything that has been deleted since the third edition of this book, is available at DDPPonline. We'll have more to say about that shortly.

Digital-Design Software Tools

Two leading suppliers have kindly allowed us to package their digital-design software tools on CD-ROMs in the domestic and some international printings of this book.

Xilinx, Inc. (`www.xilinx.com`) has provided the student edition of their ISE (Integrated Software Environment), version 6.3 or later. This tool suite includes an ABEL compiler, VHDL and Verilog language processors, a schematic drawing package, and a fitter to target HDL-based designs to Xilinx CPLDs and FPGAs. This package does not include a simulator, however.

Additional tools have been provided by Aldec, Inc. (`www.aldec.com`) in the student edition of their popular Active-HDL™ tool suite, version 6.3 or later. This package includes VHDL and Verilog language processors and a simulator and waveform viewer that allows you to test your HDL-based designs.

These tools were very useful to me as an author. Using them, I was able to write and test all of the example programs in the text. I trust that the tools will be even more useful to the students who use the text. They will allow you to write and test your own hardware designs and download them into Xilinx CPLDs and FPGAs in a digital lab environment.

Though not included with this book, Lattice Semiconductor's ispLEVER tool suite is another nice set of tools that is used at some universities. It includes basic tools for targeting ABEL-based designs to simple PLDs, as well as VHDL, Verilog, and simulator support as in the combination of the Xilinx and Aldec packages.

Even if you're not ready to do your own original designs, you can use the included Xilinx and Aldec tools to try out and modify any of the examples in the text, since the source code for all of them is available at the book's web site, discussed next.

www.ddpp.com, OneKey, and DDPPonline

Abundant support materials for this book are available on the web. These include free materials, such as selected exercise solutions and the latest errata, and protected materials, such as supplemental sections and additional exercises and solutions. The free materials are accessible by everyone, at *`www.ddpp.com`*, this book's traditional web site.

`www.ddpp.com`

OneKey

The protected materials are accessible to registered users using *OneKey*, Prentice Hall's exclusive new online resource for instructors and students, at `www.prenhall.com/OneKey`. A unique OneKey access code is included with each new copy of *Digital Design Principles and Practices*. For students who buy used books, access codes are also available for purchase separately. To obtain and use your access code, please follow the instructions that are shrink-wrapped with your new text or included in your separate OneKey purchase.

DDPPonline

Throughout this book, we use *DDPPonline* as an alias for the publisher's OneKey site, which you access at `www.prenhall.com/OneKey`. You must use your access code to register at this site, and then log in each time you use the site. If the URL `www.prenhall.com/OneKey` is too difficult for you to remember, you can also access OneKey via a prominent link at `www.ddpp.com` or its alias *`www.DDPPonline.com`*.

www.DDPPonline.com

Among other things, DDPPonline contains the following resources for students:

- Over 300 pages of additional supplemental material and design examples, organized as over two dozen sections of a few to 30 pages each. Some of these sections contain additional exercises.
- Additional exercise solutions, if your instructor also subscribes to OneKey. He or she can choose which additional solutions to make available to you. Please be aware that I haven't written a solution for every problem, so don't blame your instructor if he or she is unable to publish some solutions.
- Source files for all of the example C, ABEL, VHDL, and Verilog programs in the printed book and in the supplemental sections.

At the time of this book's publication in autumn 2005, the DDPPonline site included the following supplemental sections (names and topics):

ABEL	Miscellaneous ABEL topics.
BiPLD	Bipolar PLDs.
BJT	Bipolar junction transistors.
CAD	Computer-aided design tools.
Cntr	Counter design topics.
Dec	Decoder design topics.
DFT	Design for testability.
Diode	Diodes and diode logic.
ECL	Emitter-coupled logic.
Elec	Electrical circuits review, by Bruce M. Fleischer.
Enc	Encoder design topics.
IEEE	IEEE standard symbols.
JKSM	Analysis and synthesis of state machines using J-K flip-flops.
Min	Additional topics in combinational minimization.
Mux	Multiplexer design topics.
Pin	SSI, MSI, PLD, and ROM/RAM pinouts.
Pmin	Programmed combinational-minimization topics.
Rel	Reliability estimation.

Sreg	Shift-register design topics.
TTL	Additional topics on TTL logic.
XCabl	Examples of combinational logic design using ABEL.
XCbb	Examples of combinational logic design using MSI blocks.
XCver	Examples of combinational logic design using Verilog.
XCvhd	Examples of combinational logic design using VHDL.
XPLD	X-series PLDs.
XSabl	Examples of sequential logic design using ABEL.
XSbb	Examples of sequential logic design using MSI blocks.
XSver	Examples of sequential logic design using Verilog.
XSvhd	Examples of sequential logic design using VHDL.
Zo	Transmission lines and reflections.

Throughout this book, we refer to the above material using the underlined, color section name, sometimes adding a subsection number (e.g., Dec.1).

For Instructors

The DDPPonline web site has additional materials for instructors only; these materials are accessible by registered instructors through the OneKey system. To register or to learn more about OneKey, visit `www.prenhall.com/OneKey` or contact your Prentice Hall representative.

The instructors' site includes files with all of the figures and tables in the book. You can use these files to make presentation slides directly, or you can insert selected materials into your own customized presentations.

The site also contains answers to selected exercises—more than half of the exercises in the book, equivalent to over 200 printed pages. Using the OneKey course management system, you can choose a subset of these solutions and make them visible to your OneKey-registered students. Note that some solutions (basically, all of the third-edition *student* solutions) are available to students at `www.ddpp.com`, whether or not you make them visible on OneKey. The site tells you which ones these are, and also contains a cross-reference from third-edition to fourth-edition exercises.

Other resources for instructors include the Xilinx University Program (`www.xilinx.com/univ`) and Aldec's Educational Program (`www.aldec.com/education/university`). The Xilinx site offers a variety of product materials, course materials, and discounts on chips and boards that you can use in digital-design lab courses. Aldec's site offers both Aldec's own software packages and third-party compatible tools and prototyping systems.

www.prenhall.com/wakerlyinfo

The publisher's marketing information about this book and many accompanying resources, as well as the latest ordering information, can be found at *www.prenhall.com/wakerlyinfo*.

Errors

Warning: This book may contain errors. The author and the publisher assume no liability for any damage—incidental, brain, or otherwise—caused by errors.

There, that should make the lawyers happy. Now, to make *you* happy, let me assure you that a great deal of care has gone into the preparation of this book to make it as error free as possible. I am anxious to learn of the remaining errors so that they may be fixed in future printings, editions, and spin-offs. Therefore I will pay $5 via PayPal to the first finder of each undiscovered error—technical, typographical, or otherwise—in the printed book. Reports of errors in the web-based materials are also appreciated, but I don't pay for those. Please email your comments to me by using the appropriate link at `www.ddpp.com`.

An up-to-date list of discovered errors can always be obtained using the appropriate link at `www.ddpp.com`. It will be a very short file transfer, I hope.

Acknowledgements

Many people helped make this book possible. Most of them helped with the first three editions and are acknowledged there. Preparation of the fourth edition has been a very lonely task, but it was made easier by my friends Prem Jain and Mike Volpi at Cisco Systems. They and the company made it possible for me to cut back my commitment at Cisco to less than half time for the ten months that it took to prepare this revised edition.

For the ideas on the "principles" side of this book, I still owe great thanks to my teacher, research advisor, and friend Ed McCluskey. On the "practices" side, Dave Raaum, one of the leading members of my "Digital Designers Hall of Fame," reviewed the new Verilog material and provided many suggestions.

Since the third edition was published, I have received many helpful comments from readers. In addition to suggesting or otherwise motivating many improvements, readers have spotted dozens of typographical and technical errors whose fixes are incorporated in this fourth edition.

My sponsoring editor at Prentice Hall, Tom Robbins, deserves thanks for shepherding this project over the past years. He's the third or fourth editor who has changed jobs after (almost) completing one of my book projects, leading me to wonder whether working with me inevitably leads to burnout or success or both (and if so, then in which order?). Special thanks go to Tom's boss, Marcia Horton, who took over after his departure. If you're reading this, then she did a terrific job of pulling this one out of the fire!

Copy editor and proofreader Jennie Kaufman did a marvelous job of ensuring consistency and catching typos, including several that had been overlooked by me and everyone else since the second or third edition. Production editor Scott Disanno also deserves credit for providing a very smooth interface with the production side of the house and for inspiring me with his very quick response times during the final "crunch" stage of the project.

Thanks go to artist Ken Bakeman, whose work I discovered a couple of years ago while doing a Google search for "wakerly." His original all-electronic "painting" appears on the back cover, and the front cover is adapted from it. The circular pattern that appears in both, as well as in the chapter openings, is based on a crop circle that sources say was discovered in June 2001 near Wakerly Woods in Barrowden, Northamptonshire, England. Now, "Wakerly Woods" appears to be a misspelling of "Wakerley Woods," but I'm not complaining. I used to live on Waverley Street and people always got the spellings confused. Anyway, Ken agreed to provide his art for this edition's cover, and I think it's very striking and appropriate.

At this point in a preface, I usually thank my wife Kate for putting up with me during the arduous months of book writing or revision. I am very sad to say that after living through seven book projects and almost 34 years of marriage, Kate passed away in early 2004 after a long battle with breast cancer. So you see, when I said that preparation of this edition has been a very lonely task, I wasn't exaggerating. Still, for me, for our children, and for all of our family, friends, and community, Kate will always be in our hearts.

Sigh. As they say, life goes on. Closing thanks go to my old and new friend, fellow author and, by the time this is published, fiancée Joanne Jacobs, for her loving support and encouragement during the final preparation of this edition.

John F. Wakerly
Oakbrook Terrace, Illinois