Preface

Digital electronic circuits are the engines of cell phones, MPEG players, digital cameras, computers, data servers, personal digital devices, GPS displays, and many other consumer products that process and use information in a digital format. This book presents a basic treatment of digital circuits and the fundamental concepts used in their design. It is suitable for use as a textbook in an introductory course in an electrical engineering, computer engineering, or computer science curriculum.

Each significant advance in industry practice ultimately works its way into the engineering curriculum. Since the mid-1980's, the use of computer-based design tools has transformed the electronics industry worldwide. Application specific integrated circuits (ASICs) are designed today by using a hardware description language (HDL), such as Verilog or VHDL, to write a behavioral model of the circuit's functionality, and then synthesizing that description into a hardware realization in a particular technology, e.g., CMOS integrated circuits or field-programmable gate arrays (FPGAs). No longer a novelty, these design tools are now readily available to universities, and are migrating in a strategic way from graduate level curricula into undergraduate courses. It is clear that HDLs have an essential, significant role in educating our future engineers. Learning to design with an HDL is as important to today's students, we think, as oscilloscopes, breadboards, and logic analyzers were to previous generations of engineers, so this edition of the text adds more weight to the use of hardware description languages in designing digital circuits.

We note that introducing HDLs in a first course in designing digital circuits is not intended to replace fundamental understanding of the building blocks of such circuits or to eliminate a discussion of manual methods of design. It is still essential for a student to understand *how hardware works*. Thus, we retain a thorough treatment of combinational and sequential logic devices. Manual design practices are presented, and their results are compared with those obtained with a HDL-based paradigm. What we are presenting, however, is a shift in emphasis

on *how hardware is designed*, a shift that, we think, better prepares a student for a career in today's industry, where HDL-based design practices are prevalent.

FLEXIBILITY

The sequence of topics in the text can accommodate courses that adhere to traditional, manual-based, treatments of digital design, courses that treat design using an HDL, and courses that are in transition between or blend the two approaches. Because modern synthesis tools automatically perform logic minimization, Karnaugh maps and related topics in optimization can be presented at the beginning of a treatment of digital design, or they can be presented after circuits and their applications are examined, designed, and simulated with an HDL. The text includes both manual and HDL-based design examples. Our end-of-chapter problems further facilitate this flexibility by cross-referencing problems that address a traditional manual design task with a companion problem that uses an HDL to accomplish the task. Additionally, we link manual and HDL-based approaches by presenting annotated results of simulations in the text, in answers to selected problems at the end of the text, and in the solutions manual.

WHAT'S NEW?

The previous edition of this text recognized the importance of hardware description languages in the design of digital circuits, and incorporated new material and examples introducing students to the Verilog language, as defined by IEEE Standard 1364-1995. This revision updates and expands that treatment by:

- revising HDL-based examples to present the ANSI-C like syntax that was adopted in the standards IEEE 1364-2001 and IEEE 1364-2005
- ensuring that all HDL examples conform to industry-accepted practices for modelling digital circuits
- providing a systematic methodology for designing a datapath controller
- presenting selected exercises and solutions to end-of-chapter problems in Verilog 1995 and Verilog 2001/2005 syntax
- introducing an important design tool – the algorithmic state machine and datapath (ASMD) chart
- revising the end-of-chapter problems and expanding the set of problems by including over 75 additional problems
- providing students with fully developed answers to selected problems, including simulation results
- providing students with a CD-ROM containing simulator-ready HDL solutions of answers to selected problems
- expanding the treatment of programmable logic devices to include FPGAs

- revising the solutions manual and web-based materials and ensuring that solutions of HDL-based exercises conform to industry practices for modelling with an HDL
- discussing and demonstrating the importance of test plans for verifying HDL models of circuits
- providing instructors with verified, simulator-ready source code and test benches for all end-of chapter problems
- making all figures, tables, and HDL examples available to instructors for downloading in PDF format from the publisher
- including with the book a CD-ROM with tutorials and simulators for the IEEE-1995 and IEEE-2001 Standards of the Verilog language

In addition to the above enhancements, the text incorporates more graphical material to better serve learners who are oriented toward a graphical medium. Annotated graphical results and explanations of simulations are presented to help students understand digital circuits and to facilitate classroom discussions of them. Karnaugh maps are presented with additional graphics.

DESIGN METHODOLOGY

This edition of the text extends the previous edition's treatment of synchronous finite state machines by presenting a systematic methodology for designing a state machine to control the datapath of a digital system. Moreover, the framework in which this material is presented treats the realistic situation in which the controller uses signals from the datapath, i.e., the system has feedback. The methodology is applicable to manual and HDL-based approaches to design.

HDL-BASED APPROACH

It is not sufficient for an introduction to HDLs to dwell on language syntax. We present only those elements of the Verilog language that are matched to the level and scope of this text. Also, correct syntax does not guarantee that a model meets a functional specification or that it can be synthesized into physical hardware. We introduce students to a disciplined use of industry-based practices for writing models to ensure that a behavioral description can be synthesized into physical hardware, and that the behavior of the synthesized circuit will match that of the behavioral description. Failure to follow this discipline can lead to software race conditions in the HDL models of such machines, race conditions in the testbench used to verify them, and a mismatch between the results of simulating a behavioral model and its synthesized physical counterpart. Similarly, failure to abide by industry practices may lead to designs that simulate correctly, but which have hardware latches that are introduced into the design accidentally as a consequence of the modelling style used by the designer. The industry-based methodology we present leads to race-free and latch-free designs. It is important that students learn and follow industry practices in using HDL models, independent of whether a student's curriculum has access to synthesis tools.

VERIFICATION

In industry, significant effort is expended to verify that the functionality of a circuit is correct. Yet not much attention is given to verification in introductory texts on digital design, where the focus is on design itself, and testing is perhaps viewed as a secondary undertaking. Our experience is that this view can lead to premature declarations that "the circuit works beautifully." Likewise, industry gains repeated returns on its investment in an HDL model by ensuring that it is readable, portable and reusable. We demonstrate naming practices and the use of parameters. We also provide test benches for all of the solutions and exercises to (1) verify the functionality of the circuit, (2) underscore the importance of thorough testing, and (3) introduce students to important concepts, such as self-checking test benches. Advocating and illustrating the development of a *test plan* to guide the development of a test bench, we introduce them in the text and expand them in the solutions manual and in the answers to selected problems at the end of the text.

HDL CONTENT

This edition of the text updates and expands its treatment of the Verilog Hardware Description Language (HDL) and exploits key enhancements available in IEEE Standards 1364-2001 and 1364-2005. We have ensured that all examples in the text and all answers in the solution manual conform to accepted industry practices for modeling digital hardware. As in the previous edition, HDL material is inserted in separate sections so it can be covered or skipped as desired, does not diminish treatment of manual-based design, and does not dictate the sequence of presentation. The treatment is at a level suitable for beginning students that are learning digital circuits and a hardware description language at the same time. The text prepares students to work on significant independent design projects and to suceed in a later course in computer architecture.

- Digital circuits are introduced in Chapters 1 through 3 with an introduction to Verilog HDL in Section 3.10.
- Further discussion of modeling with HDLs occurs in Section 4.12 following the study of combinational circuits.
- Sequential circuits are covered in Chapters 5 and 6 with corresponding HDL examples in Sections 5.6 and 6.6.
- The HDL description of memory is presented in Section 7.2.
- The RTL symbols used in Verilog are introduced in Sections 8.3.
- Examples of RTL and structural models in Verilog are provided in Sections 8.6 and 8.9. Chapter 8 also presents a new, comprehensive treatment of HDL-based design of a datapath controller.
- Section 10.10 covers switch-level modeling corresponding to CMOS circuits.
- Section 11.20 supplements the hardware experiments of Chapter 11 with HDL experiments. Now the circuits designed in the laboratory can be checked by modeling them in Verilog and simulating their behavior. Then they can be synthesized and implemented with an FPGA on a prototyping board.

HDL SIMULATORS

The CD-ROM in the back of the book contains the Verilog HDL source code files for the examples in the book and two simulators provided by SynaptiCAD. The first simulator is *VeriLogger Pro*, a traditional Verilog simulator that can be used to simulate the HDL examples in the book and to verify the solutions of HDL problems. This simulator accepts the syntax of the IEEE-1995 Standard and will be useful to those who have legacy models. As an interactive simulator, *Verilogger Extreme*, accepts the syntax of IEEE-2001 as well as IEEE-1995, allowing the designer to simulate and analyze design ideas before a complete simulation model or schematic is available. This technology is particularly useful for students, because they can quickly enter Boolean and *D* flip-flop or latch input equations to check equivalency or to experiment with flip-flops and latch designs.

INSTRUCTOR RESOURCES

Instructors can download the following classroom-ready resources from the publisher (www.prenhall.com/mano):

- Source code and test benches for all Verilog HDL examples in the test
- All figures and tables in the text
- Source code for all HDL models in the solutions manual

A solution manual in typed hardcopy format with graphics, suitable for classroom presentation, will also be provided instructors.

CHAPTER SUMMARY

The following is a brief summary of the topics that are covered in each chapter.

Chapter 1 presents the various binary systems suitable for representing information in digital systems. The binary number system is explained and binary codes are illustrated. Examples are given for addition and subtraction of signed binary numbers and decimal numbers in BCD.

Chapter 2 introduces the basic postulates of Boolean algebra and shows the correlation between Boolean expressions and their corresponding logic diagrams. All possible logic operations for two variables are investigated and from that, the most useful logic gates used in the design of digital systems are determined. The characteristics of integrated circuit gates are mentioned in this chapter but a more detailed analysis of there the electronic circuits of the gates is done in Chapter 10.

Chapter 3 covers the map method for simplifying Boolean expressions. The map method is also used to simplify digital circuits constructed with AND-OR, NAND, or NOR gates. All other possible two-level gate circuits are considered and their method of implementation is explained. Verilog HDL is introduced together with simple gate-level modeling examples.

Chapter 4 outlines the formal procedures for the analysis and design of combinational circuits. Some basic components used in the design of digital systems, such as adders and code

converters, are introduced as design examples. Frequently used digital logic functions such as parallel adders and subtractors, decoders, encoders, and multiplexers are explained, and their use in the design of combinational circuits is illustrated. HDL examples are given in the gate-level, dataflow, and behavioral modeling to show the alternative ways available for describing combinational circuits in Verilog HDL. The procedure for writing a simple test bench to provide stimulus to an HDL design is presented.

Chapter 5 outlines the formal procedures for the analysis and design of clocked (synchronous) sequential circuits. The gate structure of several types of flip-flops is presented together with a discussion on the difference between level and edge triggering. Specific examples are used to show the derivation of the state table and state diagram when analyzing a sequential circuit. A number of design examples are presented with emphasis on sequential circuits that use D-type flip-flops. Behavioral modeling in Verilog HDL for sequential circuits is explained. HDL Examples are given to illustrate Mealy and Moore models of sequential circuits.

Chapter 6 deals with various sequential circuits components such as registers, shift registers, and counters. These digital components are the basic building blocks from which more complex digital systems are constructed. HDL descriptions of shift registers and counter are presented.

Chapter 7 deals with random access memory (RAM) and programmable logic devices. Memory decoding and error correction schemes are discussed. Combinational and sequential programmable devices are presented such as ROMs, PLAs, PALs, CPLDs, and FPGAs.

Chapter 8 deals with the register transfer level (RTL) representation of digital systems. The algorithmic state machine (ASM) chart is introduced. A number of examples demonstrate the use of the ASM chart, ASMD chart, RTL representation, and HDL description in the design of digital systems. The design of a finite state machine to control a datapath is presented in detail, including the realistic situation in which status signals from the datapath are used by the state machine that controls it. This chapter is the most important chapter in the book as it provides the student with a systematic approach to more advanced design projects.

Chapter 9 presents formal procedures for the analysis and design of asynchronous sequential circuits. Methods are outlined to show how an asynchronous sequential circuit can be implemented as a combinational circuit with feedback. An alternate implementation is also described that uses SR latches as the storage elements in asynchronous sequential circuits.

Chapter 10 presents the most common integrated circuit digital logic families. The electronic circuits of the common gate in each family are analyzed using electrical circuit theory. A basic knowledge of electronic circuits is necessary to fully understand the material in this chapter. Examples of Verilog switch-level descriptions demonstrate the ability to simulate circuits constructed with MOS and CMOS transistors.

Chapter 11 outlines experiments that can be performed in the laboratory with hardware that is readily available commercially. The operation of the integrated circuits used in the experiments is explained by referring to diagrams of similar components introduced in previous chapters. Each experiment is presented informally and the student is expected to produce the circuit diagram and formulate a procedure for checking the operation of the circuit in the laboratory. The last section supplements the experiments with corresponding HDL experiments. Instead of, or in addition to, the hardware construction, the student can use the Verilog HDL software provided on the CD-ROM to simulate and verify the design.

Chapter 12 presents the standard graphic symbols for logic functions recommended by an ANSI/IEEE Standard. These graphic symbols have been developed for SSI and MSI

components so that the user can recognize each function from the unique graphic symbol assigned. The chapter shows the standard graphic symbols of the integrated circuits used in the laboratory experiments. The various digital components that are represented throughout the book are similar to commercial integrated circuits. However, the text does not mention specific integrated circuits except in Chapters 11 and 12. Doing the suggested experiments in Chapter 11 while studying the theory presented in the text will enhance the practical application of digital design.

LAB EXPERIMENTS

The book may be used in a stand-alone course or with a companion lab based on the lab experiments included with the text. The lab experiments can be used in a stand-alone manner too, and can be accomplished by a traditional approach, with a breadboard and TTL circuits, or with an HDL/synthesis approach using FPGAs. Today, software for synthesizing an HDL model and implementing a circuit with an FPGA is available at no cost from vendors of FPGAs, allowing students to conduct a significant amount of work in their personal environment before using prototyping boards and other resources in a lab. Circuit boards for rapidly prototyping circuits with FPGAs are available at nominal cost, and typically include push buttons, switches, and seven-segment displays, LCDs, keypads and other I/O devices. With these resources, students can work prescribed lab exercises or their own projects and get results immediately.

The operation of the integrated circuits used in the experiments is explained by referring to diagrams of similar components introduced in previous chapters. Each experiment is presented informally and the student is expected to produce the circuit diagram and formulate a procedure for verifying the operation of the circuit in the laboratory. The last section supplements the experiments with corresponding HDL experiments. Instead of, or in addition to, the hardware construction, the student can use the Verilog HDL software provided on the CD-ROM to simulate and check the design. Synthesis tools can then be used to implement the circuit in an FPGA on a prototyping board.

Our thanks go to the editorial team at Prentice Hall for committing to this timely revision of the text. Finally, we are grateful to our wives, Sandra and Jerilynn, for encouraging our pursuit of this project.

M. MORRIS MANO
Emeritus Professor of Computer Engineering
California State University, Los Angeles

MICHAEL D. CILETTI
Department of Electrical and Computer Engineering
University of Colorado at Colorado Springs