
“Peter Ashenden is leading the way towards a new curriculum for educating the next generation of digital logic designers. Recognizing that digital design has moved from being gate-centric assembly of custom logic to processor-centric design of embedded systems, Dr. Ashenden has shifted the focus from the gate to the modern design and integration of complex integrated devices that may be physically realized in a variety of forms. Dr. Ashenden does not ignore the fundamentals, but treats them with suitable depth and breadth to provide a basis for the higher-level material. As is the norm in all of Dr. Ashenden’s writing, the text is lucid and a pleasure to read. The book is illustrated with copious examples and the companion Web site offers all that one would expect in a text of such high quality.”

— Grant Martin, Chief Scientist, Tensilica Inc.

“Dr. Ashenden has written a textbook that enables students to obtain a much broader and more valuable understanding of modern digital system design. Readers can be sure that the practices described in this book will provide a strong foundation for modern digital system design using hardware description languages.”

— Gary Spivey, George Fox University

“The convergence of miniaturized, sophisticated electronics into handheld, low-power embedded systems such as cell phones, PDAs, and MP3 players depends on efficient, digital design flows. Starting with an intuitive exploration of the basic building blocks, Digital Design: An Embedded Systems Approach introduces digital systems design in the context of embedded systems to provide students with broader perspectives. Throughout the text, Peter Ashenden’s practical approach engages students in understanding the challenges and complexities involved in implementing embedded systems.”

— Gregory D. Peterson, University of Tennessee

“Digital Design: An Embedded Systems Approach places emphasis on larger systems containing processors, memory, and involving the design
and interface of I/O functions and application-specific accelerators. The book's presentation is based on a contemporary view that reflects the real-world digital system design practice. At a time when the university curriculum is generally lagging significantly behind industry development, this book provides much needed information to students in the areas of computer engineering, electrical engineering and computer science.”

— Donald Hung, San Jose State University

“Digital Design: An Embedded Systems Approach presents the design flow of circuits and systems in a way that is both accessible and up-to-date. Because the use of hardware description languages is state-of-the-art, it is necessary that students learn how to use these languages along with an appropriate methodology. This book presents a modern approach for designing embedded systems starting with the fundamentals and progressing up to a complete system—it is application driven and full of many examples. I will recommend this book to my students.”

— Goeran Herrmann, TU Chemnitz

“Digital Design: An Embedded Systems Approach is surprisingly easy to read despite the complexity of the material. It takes the reader in a journey from the basics to a real understanding of digital design by answering the 'whys' and 'hows'—it is persuasive and instructive as it moves deeper and deeper into the material.”

— Andrey Koptyug, Mid Sweden University

“This up-to-date text on digital design is written in a very accessible style using a modern design methodology and the real world of embedded systems as its contexts. Digital Design: An Embedded Systems Approach provides excellent coverage of all aspects of the design of embedded systems, with chapters not just on logic design itself, but also on processors, memories, input/output interfacing and implementation technologies. It's particularly good at emphasizing the need to consider more than just logic design when designing a digital system: the design has to be implemented in the real world of engineering, where a whole variety of constraints, such as circuit area, circuit interconnections, interfacing requirements, power and performance, must be considered. For those who think logic design is mundane, this book brings the subject to life.”

— Roland Ibbett, University of Edinburgh
Digital Design
An Embedded Systems Approach
Using Verilog
About the Author

Peter J. Ashenden is an Adjunct Associate Professor at Adelaide University and the founder of Ashenden Designs, a consulting business specializing in electronics design automation (EDA).

From 1990 to 2000, Dr. Ashenden was a member of the faculty in the Department of Computer Science at Adelaide. He developed curriculum and taught in a number of areas for both the Computer Science and the Electrical and Electronic Engineering departments. Topics included computer organization, computer architecture, digital logic design, programming and algorithms, at all levels from undergraduate to graduate courses. He was also actively involved in academic administration at many levels within the university.

In 2000, Dr. Ashenden established Ashenden Designs. His services include training development and delivery, advising on design methodology, research in EDA tool technology, development of design languages, and standards writing. His clients include industry and government organization in the United States, Europe and SE Asia.

Since 1992, Dr. Ashenden has been involved in the IEEE VHDL standards committees, and continues to play a major role in ongoing development of the language. From 2003 to 2005 he was Chair of the IEEE Design Automation Standards Committee, which oversees development of all IEEE standards in EDA. He is currently Technical Editor for the VHDL, VHDL-AMS, and Rosetta specification language standards.

In addition to his research publications, Dr. Ashenden is author of The Designer’s Guide to VHDL and The Student’s Guide to VHDL, and coauthor of The System Designer’s Guide to VHDL-AMS and VHDL-2007: Just the New Stuff. His VHDL books are highly regarded and are the best-selling references on the subject. From 2000 to 2004, he was Series Coeditor of the Morgan Kaufmann Series on Systems on Silicon, and from 2001 to 2004 he was a member of the Editorial Board of IEEE Design and Test of Computers magazine.

Dr. Ashenden is a Senior Member of the IEEE and the IEEE Computer Society. He is also a volunteer Senior Firefighter of 12 years standing with the South Australian Country Fire Service.
To my daughter, Eleanor
—PA
CONTENTS

Preface ................................................................. xv

CHAPTER 1 Introduction and Methodology ............. 1

1.1 Digital Systems and Embedded Systems .............. 1
1.2 Binary Representation and Circuit Elements .......... 4
1.3 Real-World Circuits ........................................... 9
  1.3.1 Integrated Circuits ......................................... 10
  1.3.2 Logic Levels ................................................ 11
  1.3.3 Static Load Levels ......................................... 13
  1.3.4 Capacitive Load and Propagation Delay .............. 15
  1.3.5 Wire Delay .................................................. 17
  1.3.6 Sequential Timing .......................................... 17
  1.3.7 Power ......................................................... 18
  1.3.8 Area and Packaging ....................................... 19
1.4 Models ........................................................... 21
1.5 Design Methodology ........................................... 26
  1.5.1 Embedded Systems Design ............................... 31
1.6 Chapter Summary ............................................... 33
1.7 Further Reading ............................................... 34
Exercises ............................................................. 35

CHAPTER 2 Combinational Basics ......................... 39

2.1 Boolean Functions and Boolean Algebra .............. 39
  2.1.1 Boolean Functions ......................................... 39
  2.1.2 Boolean Algebra ............................................ 48
  2.1.3 Verilog Models of Boolean Equations ................. 51
2.2 Binary Coding ................................................... 54
  2.2.1 Using Vectors for Binary Codes ....................... 56
  2.2.2 Bit Errors ................................................... 58
2.3 Combinational Components and Circuits .............. 62
  2.3.1 Decoders and Encoders ................................... 62
  2.3.2 Multiplexers ............................................... 68
  2.3.3 Active-Low Logic ......................................... 71
2.4 Verification of Combinational Circuits ................ 74
2.5 Chapter Summary ............................................... 81
CHAPTER 3 Numeric Basics

3.1 Unsigned Integers
3.1.1 Coding Unsigned Integers
3.1.2 Operations on Unsigned Integers
3.1.3 Gray Codes

3.2 Signed Integers
3.2.1 Coding Signed Integers
3.2.2 Operations on Signed Integers

3.3 Fixed-Point Numbers
3.3.1 Coding Fixed-Point Numbers
3.3.2 Operations on Fixed-Point Numbers

3.4 Floating-Point Numbers
3.4.1 Coding Floating-Point Numbers

3.5 Chapter Summary
3.6 Further Reading
Exercises

CHAPTER 4 Sequential Basics

4.1 Storage Elements
4.1.1 Flip-flops and Registers
4.1.2 Shift Registers
4.1.3 Latches

4.2 Counters

4.3 Sequential Datapaths and Control
4.3.1 Finite-State Machines

4.4 Clocked Synchronous Timing Methodology
4.4.1 Asynchronous Inputs
4.4.2 Verification of Sequential Circuits
4.4.3 Asynchronous Timing Methodologies

4.5 Chapter Summary
4.6 Further Reading
Exercises

CHAPTER 5 Memories

5.1 General Concepts
5.2 Memory Types
5.2.1 Asynchronous Static RAM
5.2.2 Synchronous Static RAM
5.2.3 Multiport Memories ........................................ 229
5.2.4 Dynamic RAM ......................................... 233
5.2.5 Read-Only Memories .................................. 235
5.3 Error Detection and Correction ............................. 240
5.4 Chapter Summary ........................................ 244
5.5 Further Reading .......................................... 245
Exercises ......................................................... 246

CHAPTER 6 Implementation Fabrics .......................... 249

6.1 Integrated Circuits ....................................... 249
  6.1.1 Integrated Circuit Manufacture .................... 250
  6.1.2 SSI and MSI Logic Families ....................... 252
  6.1.3 Application-Specific Integrated Circuits (ASICs) .... 255
6.2 Programmable Logic Devices ............................ 258
  6.2.1 Programmable Array Logic ......................... 258
  6.2.2 Complex PLDs ....................................... 262
  6.2.3 Field-Programmable Gate Arrays .................... 263
6.3 Packaging and Circuit Boards ........................... 269
6.4 Interconnection and Signal Integrity ..................... 272
  6.4.1 Differential Signaling ................................ 276
6.5 Chapter Summary ......................................... 278
6.6 Further Reading .......................................... 279
Exercises ......................................................... 280

CHAPTER 7 Processor Basics .................................. 281

7.1 Embedded Computer Organization ...................... 281
  7.1.1 Microcontrollers and Processor Cores ............... 283
7.2 Instructions and Data .................................... 285
  7.2.1 The Gumnut Instruction Set ......................... 287
  7.2.2 The Gumnut Assembler ................................ 296
  7.2.3 Instruction Encoding ................................ 298
  7.2.4 Other CPU Instruction Sets ......................... 300
7.3 Interfacing with Memory ................................ 302
  7.3.1 Cache Memory ....................................... 307
7.4 Chapter Summary ......................................... 311
7.5 Further Reading .......................................... 311
Exercises ......................................................... 312

CHAPTER 8 I/O Interfacing .................................... 315

8.1 I/O Devices ............................................... 315
  8.1.1 Input Devices ....................................... 316
  8.1.2 Output Devices ...................................... 321
8.2 I/O Controllers ........................................ 330
  8.2.1 Simple I/O Controllers ......................... 331
  8.2.2 Autonomous I/O Controllers ................. 335
8.3 Parallel Buses ........................................ 338
  8.3.1 Multiplexed Buses ............................... 338
  8.3.2 Tristate Buses ................................. 342
  8.3.3 Open-Drain Buses ............................. 348
  8.3.4 Bus Protocols .................................. 349
8.4 Serial Transmission ................................. 353
  8.4.1 Serial Transmission Techniques .............. 353
  8.4.2 Serial Interface Standards .................. 357
8.5 I/O Software ......................................... 360
  8.5.1 Polling ........................................... 360
  8.5.2 Interrupts ....................................... 362
  8.5.3 Timers .......................................... 366
8.6 Chapter Summary ..................................... 373
8.7 Further Reading ...................................... 374
  Exercises ............................................. 375

CHAPTER 9  Accelerators ................................. 379
9.1 General Concepts .................................... 379
9.2 Case Study: Video Edge-Detection ............... 386
9.3 Verifying an Accelerator ........................... 407
9.4 Chapter Summary ..................................... 419
9.5 Further Reading ...................................... 419
  Exercises ............................................. 420

CHAPTER 10  Design Methodology ...................... 423
10.1 Design Flow ......................................... 423
  10.1.1 Architecture Exploration ..................... 425
  10.1.2 Functional Design ............................. 427
  10.1.3 Functional Verification ....................... 429
  10.1.4 Synthesis ....................................... 435
  10.1.5 Physical Design ............................... 438
10.2 Design Optimization ................................ 441
  10.2.1 Area Optimization ............................. 442
  10.2.2 Timing Optimization ......................... 443
  10.2.3 Power Optimization ........................... 448
10.3 Design for Test ..................................... 451
  10.3.1 Fault Models and Fault Simulation ............ 452
  10.3.2 Scan Design and Boundary Scan ............... 454
  10.3.3 Built-In Self Test (BIST) ...................... 458
10.4 Nontechnical Issues ............................................. 462
10.5 In Conclusion ................................................ 463
10.6 Chapter Summary ............................................. 465
10.7 Further Reading ................................................ 466

Appendix A Knowledge Test Quiz Answers ......................... 469

Appendix B Introduction to Electronic Circuits ..................... 501

B.1 Components .................................................. 501
  B.1.1 Voltage Sources ........................................... 502
  B.1.2 Resistors .................................................. 502
  B.1.3 Capacitors ............................................... 503
  B.1.4 Inductors ................................................ 503
  B.1.5 MOSFETs ............................................... 504
  B.1.6 Diodes ................................................... 506
  B.1.7 Bipolar Transistors ...................................... 507

B.2 Circuits ........................................................ 508
  B.2.1 Kirchhoff’s Laws .......................................... 508
  B.2.2 Series and Parallel R, C, and L ......................... 509
  B.2.3 RC Circuits .............................................. 511
  B.2.4 RLC Circuits ............................................ 512

B.3 Further Reading ............................................... 515

Appendix C Verilog for Synthesis .................................. 517

C.1 Data Types and Operations .................................... 517
C.2 Combinational Functions ...................................... 518
C.3 Sequential Circuits ........................................... 522
  C.3.1 Finite-State Machines ................................... 525
C.4 Memories ...................................................... 527

Appendix D The Gumnut Microcontroller Core ....................... 531

D.1 The Gumnut Instruction Set .................................... 531
  D.1.1 Arithmetic and Logical Instructions ..................... 531
  D.1.2 Shift Instructions ....................................... 535
  D.1.3 Memory and I/O Instructions ............................. 536
  D.1.4 Branch Instructions ..................................... 537
  D.1.5 Jump Instructions ....................................... 537
  D.1.6 Miscellaneous Instructions ............................... 538

D.2 The Gumnut Bus Interface ..................................... 538

Index ............................................................... 541
PREFACE

APPROACH

This book provides a foundation in digital design for students in computer engineering, electrical engineering and computer science courses. It deals with digital design as an activity in a larger systems design context. Instead of focusing on gate-level design and aspects of digital design that have diminishing relevance in a real-world design context, the book concentrates on modern and evolving knowledge and design skills.

Most modern digital design practice involves design of embedded systems, using small microcontrollers, larger CPUs/DSPs, or hard or soft processor cores. Designs involve interfacing the processor or processors to memory, I/O devices and communications interfaces, and developing accelerators for operations that are too computationally intensive for processors. Target technologies include ASICs, FPGAs, PLDs and PCBs. This is a significant change from earlier design styles, which involved use of small-scale integrated (SSI) and medium-scale integrated (MSI) circuits. In such systems, the primary design goal was to minimize gate count or IC package count. Since processors had lower performance and memories had limited capacity, a greater proportion of system functionality was implemented in hardware.

While design practices and the design context have evolved, many textbooks have not kept track. They continue to promote practices that are largely obsolete or that have been subsumed into computer-aided design (CAD) tools. They neglect many of the important considerations for modern designers. This book addresses the shortfall by taking an approach that embodies modern design practice. The book presents the view that digital logic is a basic abstraction over analog electronic circuits. Like any abstraction, the digital abstraction relies on assumptions being met and constraints being satisfied. Thus, the book includes discussion of the electrical and timing properties of circuits, leading to an understanding of how they influence design at higher levels of abstraction. Also, the book teaches a methodology based on using abstraction to manage complexity, along with principles and methods for making design trade-offs. These intellectual tools allow students to track evolving design practice after they graduate.

Perhaps the most noticeable difference between this book and its predecessors is the omission of material on Karnaugh maps and related
logic optimization techniques. Some reviewers of the manuscript argued that such techniques are still of value and are a necessary foundation for students learning digital design. Certainly, it is important for students to understand that a given function can be implemented by a variety of equivalent circuits, and that different implementations may be more or less optimal under different constraints. This book takes the approach of presenting Boolean algebra as the foundation for gate-level circuit transformation, but leaves the details of algorithms for optimization to CAD tools. The complexity of modern systems makes it more important to raise the level of abstraction at which we work and to introduce embedded systems early in the curriculum. CAD tools perform a much better job of gate-level optimization than we can do manually, using advanced algorithms to satisfy relevant constraints. Techniques such as Karnaugh maps do have a place, for example, in design of specialized hazard-free logic circuits. Thus, students can defer learning about Karnaugh maps until an advanced course in VLSI, or indeed, until they encounter the need in industrial practice. A web search will reveal many sources describing the techniques in detail, including an excellent article in Wikipedia.

The approach taken in this book makes it relevant to Computer Science courses, as well as to Computer Engineering and Electrical Engineering courses. By treating digital design as part of embedded systems design, the book will provide the understanding of hardware needed for computer science students to analyze and design systems comprising both hardware and software components. The principles of abstraction and complexity management using abstraction presented in the book are the same as those underlying much of computer science and software engineering.

Modern digital design practice relies heavily on models expressed in hardware description languages (HDLs), such as Verilog and VHDL. HDL models are used for design entry at the abstract behavioral level and for refinements at the register transfer level. Synthesis tools produce gate-level HDL models for low-level verification. Designers also express verification environments in HDLs. This book emphasizes HDL-based design and verification at all levels of abstraction. The present version uses Verilog for this purpose. A second version, *Digital Design: An Embedded Systems Approach Using VHDL*, substitutes VHDL for the same purpose.

OVERVIEW

For those who are musically inclined, the organization of this book can be likened to a two-act opera, complete with overture, intermezzo, and finale.

Chapter 1 forms the overture, introducing the themes that are to follow in the rest of the work. It starts with a discussion of the basic ideas of the digital abstraction, and introduces the basic digital circuit elements.
It then shows how various non-ideal behaviors of the elements impose constraints on what we can design. The chapter finishes with a discussion of a systematic process of design, based on models expressed in a hardware description language.

Act I of the opera comprises Chapters 2 through 5. In this act, we develop the themes of basic digital design in more detail.

Chapter 2 focuses on combinational circuits, starting with Boolean algebra as the theoretical underpinning and moving on to binary coding of information. The chapter then surveys a range of components that can be used as building blocks in larger combinational circuits, before returning to the design methodology to discuss verification of combinational circuits.

Chapter 3 expands in some detail on combinational circuits used to process numeric information. It examines various binary codes for unsigned integers, signed integers, fixed-point fractions and floating-point real numbers. For each kind of code, the chapter describes how some arithmetic operations can be performed and looks at combinational circuits that implement arithmetic operations.

Chapter 4 introduces a central theme of digital design, sequential circuits. The chapter examines several sequential circuit elements for storing information and for counting events. It then describes the concepts of a datapath and a control section, followed by a description of the clocked synchronous timing methodology.

Chapter 5 completes Act I, describing the use of memories for storing information. It starts by introducing the general concepts that are common to all kinds of semiconductor memory, and then focuses on the particular features of each type, including SRAM, DRAM, ROM and flash memories. The chapter finishes with a discussion of techniques for dealing with errors in the stored data.

The intermezzo, Chapter 6, is a digression away from functional design into physical design and the implementation fabrics used for digital systems. The chapter describes the range of integrated circuits that are used for digital systems, including ASICs, FPGAs and other PLDs. The chapter also discusses some of the physical and electrical characteristics of implementation fabrics that give rise to constraints on designs.

Act II of the opera, comprising Chapters 7 through 9, develops the embedded systems theme.

Chapter 7 introduces the kinds of processors that are used in embedded systems and gives examples of the instructions that make up embedded software programs. The chapter also describes the way instructions and data are encoded in binary and stored in memory and examines ways of connecting the processor with memory components.

Chapter 8 expands on the notion of input/output (I/O) controllers that connect an embedded computer system with devices that sense and
affect real-world physical properties. It describes a range of devices that are used in embedded computers and shows how they are accessed by an embedded processor and by embedded software.

Chapter 9 describes accelerators, that is, components that can be added to embedded systems to perform operations faster than is possible with embedded software running on a processor core. This chapter uses an extended example to illustrate design considerations for accelerators, and shows how an accelerator interacts with an embedded processor.

The finale, Chapter 10, is a coda that returns to the theme of design methodology introduced in Chapter 1. The chapter describes details of the design flow and discusses how aspects of the design can be optimized to better meet constraints. It also introduces the concept of design for test, and outlines some design for test tools and techniques. The opera finishes with a discussion of the larger context in which digital systems are designed.

After a performance of an opera, there is always a lively discussion in the foyer. This book contains a number of appendices that correspond to that aspect of the opera. Appendix A provides sample answers for the Knowledge Test Quiz sections in the main chapters. Appendix B provides a quick refresher on electronic circuits. Appendix C is a summary of the subset of Verilog used for synthesis of digital circuits. Finally, Appendix D is an instruction-set reference for the Gumnut embedded processor core used in examples in Chapters 7 through 9.

For those not inclined toward classical music, I apologize if the preceding is not a helpful analogy. An analogy with the courses of a feast came to mind, but potential confusion among readers in different parts of the world over the terms appetizer, entrée and main course make the analogy problematic. The gastronomically inclined reader should feel free to find the correspondence in accordance with local custom.

COURSE ORGANIZATION

This book covers the topics included in the Digital Logic knowledge area of the Computer Engineering Body of Knowledge described in the IEEE/ACM Curriculum Guidelines for Undergraduate Degree Programs in Computer Engineering. The book is appropriate for a course at the sophomore level, assuming only previous introductory courses in electronic circuits and computer programming. It articulates into junior and senior courses in embedded systems, computer organization, VLSI and other advanced topics.

For a full sequence in digital design, the chapters of the book can be covered in order. Alternatively, a shorter sequence could draw on Chapter 1 through Chapter 6 plus Chapter 10. Such a sequence would defer material in Chapters 7 through 9 to a subsequent course on embedded systems design.
For either sequence, the material in this book should be supplemented by a reference book on the Verilog language. The course work should also include laboratory projects, since hands-on design is the best way to reinforce the principles presented in the book.

WEB SUPPLEMENTS

No textbook can be complete nowadays without supplementary material on a website. For this book, resources for students and instructors are available at the website:

textbooks.elsevier.com/9780123695277

For students, the website contains:

- Source code for all of the example HDL models in the book
- Tutorials on the VHDL and Verilog hardware description languages
- An assembler for the Gumnut processor described in Chapter 7 and Appendix D
- A link to the ISE WebPack FPGA EDA tool suite from Xilinx
- A link to the ModelSim Xilinx Edition III VHDL and Verilog simulator from Mentor Graphics Corporation
- A link to an evaluation edition of the Synplify Pro PFGA synthesis tool from Synplicity, Inc. (see inside back cover for more details).
- Tutorials on use of the EDA tools for design projects

For instructors, the website contains a protected area with additional resources:

- An instructor’s manual
- Suggested lab projects
- Lecture notes
- Figures from the text in JPG and PPT formats

Instructors are invited to contribute additional material for the benefit of their colleagues.

Despite the best efforts of all involved, some errors have no doubt crept through the review and editorial process. A list of detected errors will be available accumulated on the website mentioned above. Should you detect such an error, please check whether it has been previously recorded. If not, I would be grateful for notice by email to

peter@ashenden.com.au
I would also be delighted to hear feedback about the book and supplementary material, including suggestions for improvement.

ACKNOWLEDGMENTS

This book arose from my long-standing desire to bring a more modern approach to the teaching of digital design. I am deeply grateful to the folks at Morgan Kaufmann Publishers for supporting me in realizing this goal, and for their guidance and advice in shaping the book. Particular thanks go to Denise Penrose, Publisher; Nate McFadden, Developmental Editor and Kim Honjo, Editorial Assistant. Thanks also to Dawnmarie Simpson at Elsevier for meticulous attention to detail and for making the production process go like clockwork.

The manuscript benefited from comprehensive reviews by Dr. A. Bouridane, Queen’s University Belfast; Prof. Goeran Herrmann, Chemnitz University of Technology; Prof. Donald Hung, San Jose State University; Prof. Roland Ibbett, University of Edinburgh; Dr. Andrey Koptyug, Mid Sweden University; Dr. Grant Martin, Tensilica, Inc.; Dr. Gregory D. Peterson, University of Tennessee; Brian R. Prasky, IBM; Dr. Gary Spivey, George Fox University; Dr. Peixin Zhong, Michigan State University; and an anonymous reviewer from Rensselaer Polytechnic Institute. Also, my esteemed colleague Cliff Cummings of Sunburst Design, Inc., provided technical reviews of the Verilog code and related text. To all of these, my sincere thanks for their contributions. The immense improvement from my first draft to the final draft is due to their efforts.

The book and the associated teaching materials also benefited from field testing: in alpha form by myself at the University of Adelaide and by Dr. Monte Tull at The University of Oklahoma; and in beta form by James Sterbenz at The University of Kansas. To them and to their students, thanks for your forbearance with the errors and for your valuable feedback.