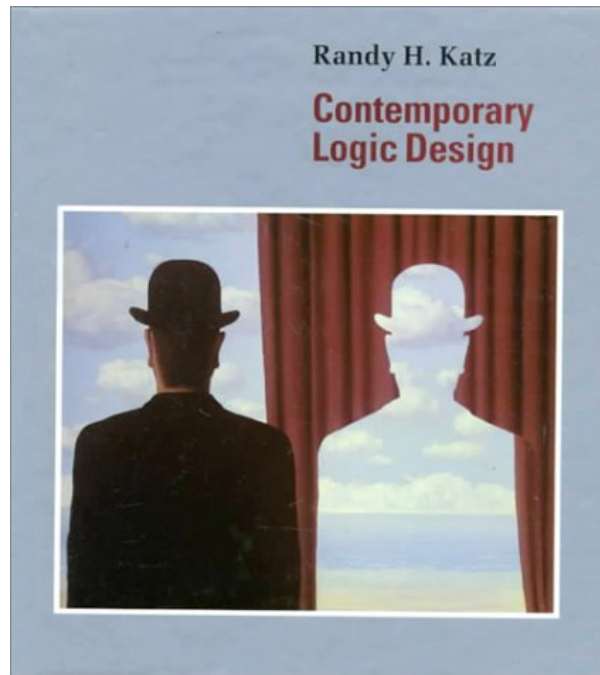


CONTEMPORARY LOGIC DESIGN BY RANDY H. KATZ



**DOWNLOAD EBOOK : CONTEMPORARY LOGIC DESIGN BY RANDY H. KATZ
PDF**



Randy H. Katz

Contemporary Logic Design



Click link bellow and free register to download ebook:
CONTEMPORARY LOGIC DESIGN BY RANDY H. KATZ

[DOWNLOAD FROM OUR ONLINE LIBRARY](#)

CONTEMPORARY LOGIC DESIGN BY RANDY H. KATZ PDF

Find out the technique of doing something from several sources. Among them is this publication entitled **Contemporary Logic Design By Randy H. Katz**. It is a very well understood book Contemporary Logic Design By Randy H. Katz that can be suggested to read now. This advised publication is among the all fantastic Contemporary Logic Design By Randy H. Katz collections that remain in this site. You will certainly also discover various other titles and styles from numerous writers to look right here.

From the Back Cover

In the past ten years there has been a revolution in the practice of hardware design. Professionals now rely on CAD software, rapid prototyping, and programmable logic devices to streamline the design process. Contemporary Logic Design is the first text to address these changes and offer a truly modern introduction to logic design. From the first chapter, the author complements his presentation of logic design theory with discussions of current design technologies.

The book provides comprehensive coverage of programmable logic, including ROMs, PALs, and PLAs. A Practical Matters section concludes most chapters, which ties theory to practice and explains design technologies in detail. To synthesize the text coverage of combinational and sequential design methods, the author uses a detailed case study of a simple processor design in the final two chapters.

The text introduces readers to a wide range of software tools, including schematic capture, logic simulation and Boolean minimization, and demonstrates how they fit into the hardware design process. The author also encourages hands-on experimentation with software tools such as LogicWorks to bolster the reader's understanding of practical design methods.

About the Author

Randy Katz is a Professor of Electrical Engineering and Computer Science at the University of California, Berkeley. He recently received a Distinguished Teaching Award for his contributions to engineering education, and he is coinvestigator on a multicampus NSF-sponsored project to restructure education. His research interests include I/O controller design and high performance striped disk and tape subsystems. He has served on the White House Technology Task Force and Vice President Gore's "Reinventing Government" Task Force.

Excerpt. © Reprinted by permission. All rights reserved.

A Second Edition

In the decade since the first edition of this book was published, the technologies of digital design have continued to evolve. The evolution has run along two closely related tracks: the underlying physical

technology and the software tools that facilitate the application of the new devices. The trends identified in the first edition have continued stronger than ever and promise to continue for some time to come. Specifically, programmable logic has become virtually the norm for digital designers and the art of digital design now absolutely requires the software skills to deal with hardware description languages.

No longer do we see the familiar yellow cover of the TTL Data Book on every designer's bookshelf. In fact, for many application areas, even small programmable logic devices (PLDs), the mainstays of the 1970s and early 1980s, are rapidly disappearing. The burgeoning market for smaller, lower power, and more portable devices has driven high levels of integration into almost every product. This also has changed the nature of optimization; the focus is now on what goes into each chip rather than on the collection of individual gates needed to realize the design. The optimizations of today are more and more often made at the architecture level rather than in the switches.

Hardware designers now spend the majority of their time dealing with software. Specifically, the tools needed to efficiently map digital designs onto the emerging programmable devices that are growing ever more sophisticated. They capture their design specifications in software with description languages appropriate for describing the parallelism of hardware; they use software tools to simulate their designs and then to synthesize it into the implementation technology of choice. Design time is reduced radically as market pressures require products to be introduced quickly, at the right price and performance.

Although the overgrowing complexity of designs necessitates more powerful abstractions, the fundamentals haven't changed. In fact, the contemporary digital designer must have a broader understanding of the discipline of computation than ever before, including both hardware and software. In this second edition, we provide this broader perspective.

Changes from the First Edition

There are many changes from the first edition that can be grouped into four rough categories. First, we updated the hardware technologies discussed in the book. Second, we added a more complete, if nevertheless introductory, treatment of the software tools that are now so commonplace in the designer's tool kit. Third, we responded to the comments and suggestions received over the years by the many faculty and practitioners who have used the book. Finally, we rationalized the organization of the text so that concepts, technologies, tools, and practical matters were more clearly defined.

New Introduction

The introduction has been changed from one that focused on the process of design to one that introduces the concepts of computation, encoding, and sequencing. This sets out a better road map to the rest of the book and provides a rationale for its organization. Rather than discussing the design process in the abstract, we now include many more case studies to help the student gain that understanding by seeing the process in action.

Repartitioning of Material

Each of the two major sections on combinational and sequential logic was divided into a set of chapters. These first cover the fundamental concepts, then describe the principles of manipulating the logic into different forms, followed by a discussion of the optimizations and tools that are available, and concludes with an overview of the technologies available to build logic circuits. Each is capped by a set of comprehensive design case studies that make each of the issues concrete.

More Emphasis on Programmable Logic

We have added new material on the latest programmable logic technologies that have quickly become the dominant style for realizing digital designs. We do not attempt to provide all the information needed to work with any one technology. Those used will vary dramatically from institution to institution. Therefore, the book needs to be supplemented with a laboratory guide that covers the specifics of a particular installation. In this text, we focused on the underlying concepts. We expect laboratory guides to be available in the form of web-based materials that can be easily customized to the variety already out there and updated as new technologies emerge.

Inclusion of Hardware Description Languages

HDLs are now given a more central role to reflect their total acceptance by the design community over the past 10 years. We describe only the basics of one of the dominant languages, namely Verilog, focusing on describing behavior, as well as covering the basics of HDL simulation models. We highlight the power of the languages in making designs more parameterizable and customizable and designers more efficient.

Design Case Studies

Nothing helps students learn design as much as designing for themselves. The next best thing is to provide a large collection of examples where the intuitions and rules of thumb are discussed explicitly. The hope is that this will help bootstrap new digital designers into the world of practical applications rather than the drill problems that were the norm in simpler times. There are many new and extensive design examples sprinkled throughout the text and in two large case study chapters focusing on combinational and sequential logic.

Elimination of Chapters on Datapath, Control, and Register-Transfer

We decided to remove the last two chapters of the first edition, that focused on datapath and register-transfer design, and a simple processor as an in-depth design case study of the interaction of control and datapath. While these topics are without a doubt important, on reflection we felt they are better left for a more extended study of digital design than could be included within the page limit of this edition. Instead, we chose more intensive coverage of programmable logic and HDLs, with extensive but smaller design examples spread throughout the text. We plan to make supplementary materials on the eliminated topics available on the web.

Navigating the Book

The book is organized into 10 chapters and three appendices. Chapter 1 is an overall introduction to the field. Chapters 2 through 5 cover combinational logic. Chapters 6 through 10 cover sequential logic. The three appendices provide some potentially useful background material that may have been part of other courses in a computer or electrical engineering curriculum.

Chapter 1 is an ambitious attempt to introduce many of the concepts of digital design through a short history of the evolution of digital hardware and two simple examples. Many may find that it introduces too many concepts too quickly for students to grasp their importance. However, this was not the intent. We fully expect students to be somewhat overwhelmed by the number of new concepts that come up in the discussion of the example. The purpose of the chapter is to provide an aerial view of the field so that students find it easier to see how the pieces they will see, in much greater detail and depth in later chapters, fit together coherently. It is certainly possible to replace this chapter with a more traditional introduction.

The next four chapters lay out the concepts of combinational logic design, closing with a set of comprehensive examples.

Chapter 2 covers the basics of combinational logic from simple gates to their time behavior. It lays out the concepts of two-level and multilevel logic and motivates why we would want to simplify logic. Some of the basic machinery for manipulating logic is presented with an emphasis on pencil-and-paper methods.

Chapter 3 delves into methods for working with combinational logic. It begins by describing the algorithms inside of today's CAD tools and ends with an overview of hardware description languages and uses Verilog to demonstrate key elements. Included is a discussion of the discrete simulation concepts that help to clarify the language constructs. There is probably not enough detail to make this book the sole resource for laboratory work with CAD tools. We wanted to keep the book focused on key concepts rather than on details of particular tools. It will need to be supplemented with appropriate manuals for the particular tools students will find in their own laboratories. This chapter also covers timing issues in more detail, including hazards and hazard-elimination strategies.

Chapter 4 presents the full range of implementation technologies available to the logic designer for combinational logic. It is paired with Chapter 9 that does the same for sequential logic. Chapter 4 starts with basic logic gates (as in the traditional TTL-based courses), but quickly progresses to programmable logic (PLDs and two-level forms) and then to field-programmable gate arrays. We also discuss other types of logic constructs such as tri-state and open-collector logic. Basic electronics to support this discussion are in Appendix B.

Chapter 5 culminates the combinational logic section of the book with seven examples of increasing complexity. We emphasize problem solving from the initial specification and have provided considerable discussion of how to transform an initial informal description of the problem into precise logical statements while keeping track of the assumptions that are being made. Our goal in this chapter is to show the range of logic design and how to judge design tradeoffs and take advantage of optimization opportunities.

The remaining five chapters do the same for sequential logic what the Chapters 2 through 5 did for combinational logic.

Chapter 6 begins this section by introducing the idea of circuits with feedback and how they can be analyzed. We develop the basic elements of sequential logic, latches, and flip-flops by recapitulating their evolution. This is coupled with a discussion of the timing methodologies that make it practical to build large sequential logic systems. These methodologies are illustrated with simple sequential systems of shift registers. The chapter concludes with a continuation of the exposition of hardware description languages started in Chapter 3 and extends it to basic sequential logic elements.

Chapter 7 covers the central concept of finite state machines. It begins by using counters as a simple form of FSM and then moves on to the basic Moore and Mealy models for organizing sequential behavior. Like Chapter 2, it concludes by motivating the various optimization opportunities.

Chapter 8 extends the basic ideas of Chapter 7 and expands on the details of FSM optimization by treating state minimization, state encoding, and FSM partitioning, in turn. Each of these is illustrated with examples that highlight the tradeoffs at each stage of optimization. An additional section at the end of the chapter provides some guidelines for structuring FSM descriptions in HDLs.

Chapter 9 concludes the discussion of implementation technologies. It recapitulates all the technologies used for combinational logic introduced in Chapter 4 but focusing on their sequential logic counterparts.

Chapter 10 is a large chapter with six comprehensive design examples that bring to practice all the concepts in the text. It begins with the sequential logic example from Chapter 1, now discussed in full detail, to tie back to the start of the text and ends with the serial transmission of characters from a keypad to display. The

latter examples focus on the partitioning of design problems into communicating pieces along two dimensions: parallel state machines and partitioning into data-path and control.

The three appendices cover number systems, basic electronics, and flip-flop, types. The first two cover concepts that students are likely to have already seen in mathematics, physics, electrical engineering, or computer science introductory courses. They are not intended to be extensive treatment of these topics but only provide the background most directly connected to the main topics of this text. The appendix on flip-flop types is provided for historical completeness.

The Complete Teaching Package

The material in this book easily fills a quarter-long course and can be comfortably covered in a semester-long course. In fact, it is likely that supplemental topics, governed by the place in the curriculum the semester-long course occupies, can and should be included. These could be: more in-depth discussion of CAD algorithms including their data structures, efficiency, and implementation; further discussion of design tradeoffs in a particular implementation technology such as FPGAs; a larger design problem that can serve as a term project to highlight issues of scale and debugging; and topics from computer organization emphasizing partitioning into data-path and control and optimizations of both. Of course, individual instructors may also find that re-ordering some of the material makes more sense in their environments. For example, it is certainly possible to proceed by following the two sections in parallel rather than serially. Chapter 2 plus 6 and 7 can be paired, followed by 3 and 8, then 4 and 9, with the larger design examples of 5 and 10 together at the end. Many topics can also be skipped altogether. For example, CAD tools and their algorithms may be relegated to another course. Similarly, HDLs do not need to be included if the design environment focuses on schematic-level design. In the technology dimension, FPGAs can be skipped as they may be included in a later course on more advanced design methods. Our goal in organizing the book was to make it easier to make these customizations.

Finally, we are making a wealth of supplementary material available to course instructors and students. Our publisher's web site includes:

- A set of CAD tools that supports all the concepts presented in this text;
- A comprehensive set of lecture slides;
- Samples of possible laboratory assignments and projects;
- Solutions to all the problems in the text; and
- Supplementary material on computer organization for those that include that material in their introductory logic design classes.

We hope you will agree with us that this second edition is a worthy successor to the first.

RANDY H. KATZ
GAETANO BORRIELLO

CONTEMPORARY LOGIC DESIGN BY RANDY H. KATZ PDF

[Download: CONTEMPORARY LOGIC DESIGN BY RANDY H. KATZ PDF](#)

Book **Contemporary Logic Design By Randy H. Katz** is one of the valuable worth that will certainly make you always abundant. It will certainly not mean as rich as the cash give you. When some people have absence to face the life, people with several e-books sometimes will be wiser in doing the life. Why need to be e-book Contemporary Logic Design By Randy H. Katz It is really not meant that publication Contemporary Logic Design By Randy H. Katz will certainly offer you power to reach every little thing. The e-book is to review and also just what we implied is guide that is read. You could likewise see exactly how guide qualifies Contemporary Logic Design By Randy H. Katz and numbers of e-book collections are giving right here.

Poses now this *Contemporary Logic Design By Randy H. Katz* as one of your book collection! But, it is not in your bookcase collections. Why? This is guide Contemporary Logic Design By Randy H. Katz that is given in soft documents. You could download and install the soft file of this incredible book Contemporary Logic Design By Randy H. Katz currently as well as in the web link supplied. Yeah, various with the other individuals who search for book Contemporary Logic Design By Randy H. Katz outside, you could obtain much easier to pose this book. When some people still walk right into the shop and also browse the book Contemporary Logic Design By Randy H. Katz, you are below only remain on your seat and also get guide Contemporary Logic Design By Randy H. Katz.

While the other people in the establishment, they are not sure to locate this Contemporary Logic Design By Randy H. Katz directly. It could require even more times to go establishment by establishment. This is why we expect you this website. We will supply the best way and also referral to get the book Contemporary Logic Design By Randy H. Katz Also this is soft file book, it will be ease to carry Contemporary Logic Design By Randy H. Katz wherever or save in your home. The difference is that you might not need relocate guide Contemporary Logic Design By Randy H. Katz place to area. You might need only duplicate to the other devices.

CONTEMPORARY LOGIC DESIGN BY RANDY H. KATZ PDF

In the past ten years there has been a revolution in the practice of hardware design. Professionals now rely on CAD software, rapid prototyping, and programmable logic devices to streamline the design process. Contemporary Logic Design is the first text to address these changes and offer a truly modern introduction to logic design. From the first chapter, the author complements his presentation of logic design theory with discussions of current design technologies. The book provides comprehensive coverage of programmable logic, including ROMs, PALs, and PLAs. A Practical Matters section concludes most chapters, which ties theory to practice and explains design technologies in detail. To synthesize the text coverage of combinational and sequential design methods, the author uses a detailed case study of a simple processor design in the final two chapters. The text introduces readers to a wide range of software tools, including schematic capture, logic simulation and Boolean minimization, and demonstrates how they fit into the hardware design process. The author also encourages hands-on experimentation with software tools such as LogicWorks to bolster the reader's understanding of practical design methods.

- Sales Rank: #1492102 in Books
- Published on: 1993-08-31
- Original language: English
- Number of items: 1
- Dimensions: 9.30" h x 1.30" w x 8.10" l, 1.10 pounds
- Binding: Hardcover
- 710 pages

From the Back Cover

In the past ten years there has been a revolution in the practice of hardware design. Professionals now rely on CAD software, rapid prototyping, and programmable logic devices to streamline the design process. Contemporary Logic Design is the first text to address these changes and offer a truly modern introduction to logic design. From the first chapter, the author complements his presentation of logic design theory with discussions of current design technologies.

The book provides comprehensive coverage of programmable logic, including ROMs, PALs, and PLAs. A Practical Matters section concludes most chapters, which ties theory to practice and explains design technologies in detail. To synthesize the text coverage of combinational and sequential design methods, the author uses a detailed case study of a simple processor design in the final two chapters.

The text introduces readers to a wide range of software tools, including schematic capture, logic simulation and Boolean minimization, and demonstrates how they fit into the hardware design process. The author also encourages hands-on experimentation with software tools such as LogicWorks to bolster the reader's understanding of practical design methods.

About the Author

Randy Katz is a Professor of Electrical Engineering and Computer Science at the University of California,

Berkeley. He recently received a Distinguished Teaching Award for his contributions to engineering education, and he is coinvestigator on a multicampus NSF-sponsored project to restructure education. His research interests include I/O controller design and high performance striped disk and tape subsystems. He has served on the White House Technology Task Force and Vice President Gore's "Reinventing Government" Task Force.

Excerpt. © Reprinted by permission. All rights reserved.

A Second Edition

In the decade since the first edition of this book was published, the technologies of digital design have continued to evolve. The evolution has run along two closely related tracks: the underlying physical technology and the software tools that facilitate the application of the new devices. The trends identified in the first edition have continued stronger than ever and promise to continue for some time to come. Specifically, programmable logic has become virtually the norm for digital designers and the art of digital design now absolutely requires the software skills to deal with hardware description languages.

No longer do we see the familiar yellow cover of the TTL Data Book on every designer's bookshelf. In fact, for many application areas, even small programmable logic devices (PLDs), the mainstays of the 1970s and early 1980s, are rapidly disappearing. The burgeoning market for smaller, lower power, and more portable devices has driven high levels of integration into almost every product. This also has changed the nature of optimization; the focus is now on what goes into each chip rather than on the collection of individual gates needed to realize the design. The optimizations of today are more and more often made at the architecture level rather than in the switches.

Hardware designers now spend the majority of their time dealing with software. Specifically, the tools needed to efficiently map digital designs onto the emerging programmable devices that are growing ever more sophisticated. They capture their design specifications in software with description languages appropriate for describing the parallelism of hardware; they use software tools to simulate their designs and then to synthesize it into the implementation technology of choice. Design time is reduced radically as market pressures require products to be introduced quickly, at the right price and performance.

Although the overgrowing complexity of designs necessitates more powerful abstractions, the fundamentals haven't changed. In fact, the contemporary digital designer must have a broader understanding of the discipline of computation than ever before, including both hardware and software. In this second edition, we provide this broader perspective.

Changes from the First Edition

There are many changes from the first edition that can be grouped into four rough categories. First, we updated the hardware technologies discussed in the book. Second, we added a more complete, if nevertheless introductory, treatment of the software tools that are now so commonplace in the designer's tool kit. Third, we responded to the comments and suggestions received over the years by the many faculty and practitioners who have used the book. Finally, we rationalized the organization of the text so that concepts, technologies, tools, and practical matters were more clearly defined.

New Introduction

The introduction has been changed from one that focused on the process of design to one that introduces the concepts of computation, encoding, and sequencing. This sets out a better road map to the rest of the book and provides a rationale for its organization. Rather than discussing the design process in the abstract, we

now include many more case studies to help the student gain that understanding by seeing the process in action.

Repartitioning of Material

Each of the two major sections on combinational and sequential logic was divided into a set of chapters. These first cover the fundamental concepts, then describe the principles of manipulating the logic into different forms, followed by a discussion of the optimizations and tools that are available, and concludes with an overview of the technologies available to build logic circuits. Each is capped by a set of comprehensive design case studies that make each of the issues concrete.

More Emphasis on Programmable Logic

We have added new material on the latest programmable logic technologies that have quickly become the dominant style for realizing digital designs. We do not attempt to provide all the information needed to work with any one technology. Those used will vary dramatically from institution to institution. Therefore, the book needs to be supplemented with a laboratory guide that covers the specifics of a particular installation. In this text, we focused on the underlying concepts. We expect laboratory guides to be available in the form of web-based materials that can be easily customized to the variety already out there and updated as new technologies emerge.

Inclusion of Hardware Description Languages

HDLs are now given a more central role to reflect their total acceptance by the design community over the past 10 years. We describe only the basics of one of the dominant languages, namely Verilog, focusing on describing behavior, as well as covering the basics of HDL simulation models. We highlight the power of the languages in making designs more parameterizable and customizable and designers more efficient.

Design Case Studies

Nothing helps students learn design as much as designing for themselves. The next best thing is to provide a large collection of examples where the intuitions and rules of thumb are discussed explicitly. The hope is that this will help bootstrap new digital designers into the world of practical applications rather than the drill problems that were the norm in simpler times. There are many new and extensive design examples sprinkled throughout the text and in two large case study chapters focusing on combinational and sequential logic.

Elimination of Chapters on Datapath, Control, and Register-Transfer

We decided to remove the last two chapters of the first edition, that focused on datapath and register-transfer design, and a simple processor as an in-depth design case study of the interaction of control and datapath. While these topics are without a doubt important, on reflection we felt they are better left for a more extended study of digital design than could be included within the page limit of this edition. Instead, we chose more intensive coverage of programmable logic and HDLs, with extensive but smaller design examples spread throughout the text. We plan to make supplementary materials on the eliminated topics available on the web.

Navigating the Book

The book is organized into 10 chapters and three appendices. Chapter 1 is an overall introduction to the field. Chapters 2 through 5 cover combinational logic. Chapters 6 through 10 cover sequential logic. The three appendices provide some potentially useful background material that may have been part of other courses in

a computer or electrical engineering curriculum.

Chapter 1 is an ambitious attempt to introduce many of the concepts of digital design through a short history of the evolution of digital hardware and two simple examples. Many may find that it introduces too many concepts too quickly for students to grasp their importance. However, this was not the intent. We fully expect students to be somewhat overwhelmed by the number of new concepts that come up in the discussion of the example. The purpose of the chapter is to provide an aerial view of the field so that students find it easier to see how the pieces they will see, in much greater detail and depth in later chapters, fit together coherently. It is certainly possible to replace this chapter with a more traditional introduction.

The next four chapters lay out the concepts of combinational logic design, closing with a set of comprehensive examples.

Chapter 2 covers the basics of combinational logic from simple gates to their time behavior. It lays out the concepts of two-level and multilevel logic and motivates why we would want to simplify logic. Some of the basic machinery for manipulating logic is presented with an emphasis on pencil-and-paper methods.

Chapter 3 delves into methods for working with combinational logic. It begins by describing the algorithms inside of today's CAD tools and ends with an overview of hardware description languages and uses Verilog to demonstrate key elements. Included is a discussion of the discrete simulation concepts that help to clarify the language constructs. There is probably not enough detail to make this book the sole resource for laboratory work with CAD tools. We wanted to keep the book focused on key concepts rather than on details of particular tools. It will need to be supplemented with appropriate manuals for the particular tools students will find in their own laboratories. This chapter also covers timing issues in more detail, including hazards and hazard-elimination strategies.

Chapter 4 presents the full range of implementation technologies available to the logic designer for combinational logic. It is paired with Chapter 9 that does the same for sequential logic. Chapter 4 starts with basic logic gates (as in the traditional TTL-based courses), but quickly progresses to programmable logic (PLDs and two-level forms) and then to field-programmable gate arrays. We also discuss other types of logic constructs such as tri-state and open-collector logic. Basic electronics to support this discussion are in Appendix B.

Chapter 5 culminates the combinational logic section of the book with seven examples of increasing complexity. We emphasize problem solving from the initial specification and have provided considerable discussion of how to transform an initial informal description of the problem into precise logical statements while keeping track of the assumptions that are being made. Our goal in this chapter is to show the range of logic design and how to judge design tradeoffs and take advantage of optimization opportunities.

The remaining five chapters do the same for sequential logic what the Chapters 2 through 5 did for combinational logic.

Chapter 6 begins this section by introducing the idea of circuits with feedback and how they can be analyzed. We develop the basic elements of sequential logic, latches, and flip-flops by recapitulating their evolution. This is coupled with a discussion of the timing methodologies that make it practical to build large sequential logic systems. These methodologies are illustrated with simple sequential systems of shift registers. The chapter concludes with a continuation of the exposition of hardware description languages started in Chapter 3 and extends it to basic sequential logic elements.

Chapter 7 covers the central concept of finite state machines. It begins by using counters as a simple form of FSM and then moves on to the basic Moore and Mealy models for organizing sequential behavior. Like

Chapter 2, it concludes by motivating the various optimization opportunities.

Chapter 8 extends the basic ideas of Chapter 7 and expands on the details of FSM optimization by treating state minimization, state encoding, and FSM partitioning, in turn. Each of these is illustrated with examples that highlight the tradeoffs at each stage of optimization. An additional section at the end of the chapter provides some guidelines for structuring FSM descriptions in HDLs.

Chapter 9 concludes the discussion of implementation technologies. It recapitulates all the technologies used for combinational logic introduced in Chapter 4 but focusing on their sequential logic parts.

Chapter 10 is a large chapter with six comprehensive design examples that bring to practice all the concepts in the text. It begins with the sequential logic example from Chapter 1, now discussed in full detail, to tie back to the start of the text and ends with the serial transmission of characters from a keypad to display. The latter examples focus on the partitioning of design problems into communicating pieces along two dimensions: parallel state machines and partitioning into data-path and control.

The three appendices cover number systems, basic electronics, and flip-flop types. The first two cover concepts that students are likely to have already seen in mathematics, physics, electrical engineering, or computer science introductory courses. They are not intended to be extensive treatment of these topics but only provide the background most directly connected to the main topics of this text. The appendix on flip-flop types is provided for historical completeness.

The Complete Teaching Package

The material in this book easily fills a quarter-long course and can be comfortably covered in a semester-long course. In fact, it is likely that supplemental topics, governed by the place in the curriculum the semester-long course occupies, can and should be included. These could be: more in-depth discussion of CAD algorithms including their data structures, efficiency, and implementation; further discussion of design tradeoffs in a particular implementation technology such as FPGAs; a larger design problem that can serve as a term project to highlight issues of scale and debugging; and topics from computer organization emphasizing partitioning into data-path and control and optimizations of both. Of course, individual instructors may also find that re-ordering some of the material makes more sense in their environments. For example, it is certainly possible to proceed by following the two sections in parallel rather than serially. Chapter 2 plus 6 and 7 can be paired, followed by 3 and 8, then 4 and 9, with the larger design examples of 5 and 10 together at the end. Many topics can also be skipped altogether. For example, CAD tools and their algorithms may be relegated to another course. Similarly, HDLs do not need to be included if the design environment focuses on schematic-level design. In the technology dimension, FPGAs can be skipped as they may be included in a later course on more advanced design methods. Our goal in organizing the book was to make it easier to make these customizations.

Finally, we are making a wealth of supplementary material available to course instructors and students. Our publisher's web site includes:

- A set of CAD tools that supports all the concepts presented in this text;
- A comprehensive set of lecture slides;
- Samples of possible laboratory assignments and projects;
- Solutions to all the problems in the text; and
- Supplementary material on computer organization for those that include that material in their introductory logic design classes.

We hope you will agree with us that this second edition is a worthy successor to the first.

RANDY H. KATZ
GAETANO BORRIELLO

Most helpful customer reviews

0 of 0 people found the following review helpful.

Thorough with the simple stuff, sloppy with the rest

By MarbleTech

This book is thorough with the simple things, but it brings a surprising combination of sloppy examples, vague descriptions, and poor organization.

That being said, anyone with a decent background in simple logic design should find this book helpful. You can learn the stuff, even if it isn't organized in the most useful way. Overall it is a decent book, and I would recommend it to people interested in low to mid range complexity logic design.

1 of 1 people found the following review helpful.

Digital Logic

By Stacie

I had to purchase this book for a Digital Logic class i am taking. I am not very happy with the book since it does not cover concepts in detail (with examples in the text) instead it has questions at the end of the chapter but they do not offer the solutions in the book or online so it makes it difficult to know if you are understanding and getting the questions correct. It also jumps around in the chapters, no chapter is stand alone for a specific area.

2 of 3 people found the following review helpful.

Katz ROCKS!

By Gadget Geek

Loved the book- all the way through. This was a required text at UC Berkeley College of Engineering for the 5 unit, upper division class I took there in my final semester of my senior year.

Not only was the book helpful to support the lecture notes and problem sets, it was also very helpful while I was designing one of the biggest projects I've ever done.

Clear, concise, well done Professor Katz!

See all 21 customer reviews...

CONTEMPORARY LOGIC DESIGN BY RANDY H. KATZ PDF

Now, reading this magnificent **Contemporary Logic Design By Randy H. Katz** will certainly be easier unless you obtain download the soft documents here. Merely right here! By clicking the connect to download and install Contemporary Logic Design By Randy H. Katz, you can start to get guide for your own. Be the very first owner of this soft data book Contemporary Logic Design By Randy H. Katz Make difference for the others as well as get the first to step forward for Contemporary Logic Design By Randy H. Katz Present moment!

From the Back Cover

In the past ten years there has been a revolution in the practice of hardware design. Professionals now rely on CAD software, rapid prototyping, and programmable logic devices to streamline the design process. Contemporary Logic Design is the first text to address these changes and offer a truly modern introduction to logic design. From the first chapter, the author complements his presentation of logic design theory with discussions of current design technologies.

The book provides comprehensive coverage of programmable logic, including ROMs, PALs, and PLAs. A Practical Matters section concludes most chapters, which ties theory to practice and explains design technologies in detail. To synthesize the text coverage of combinational and sequential design methods, the author uses a detailed case study of a simple processor design in the final two chapters.

The text introduces readers to a wide range of software tools, including schematic capture, logic simulation and Boolean minimization, and demonstrates how they fit into the hardware design process. The author also encourages hands-on experimentation with software tools such as LogicWorks to bolster the reader's understanding of practical design methods.

About the Author

Randy Katz is a Professor of Electrical Engineering and Computer Science at the University of California, Berkeley. He recently received a Distinguished Teaching Award for his contributions to engineering education, and he is coinvestigator on a multicampus NSF-sponsored project to restructure education. His research interests include I/O controller design and high performance striped disk and tape subsystems. He has served on the White House Technology Task Force and Vice President Gore's "Reinventing Government" Task Force.

Excerpt. © Reprinted by permission. All rights reserved.

A Second Edition

In the decade since the first edition of this book was published, the technologies of digital design have continued to evolve. The evolution has run along two closely related tracks: the underlying physical technology and the software tools that facilitate the application of the new devices. The trends identified in the first edition have continued stronger than ever and promise to continue for some time to come. Specifically, programmable logic has become virtually the norm for digital designers and the art of digital design now absolutely requires the software skills to deal with hardware description languages.

No longer do we see the familiar yellow cover of the TTL Data Book on every designer's bookshelf. In fact, for many application areas, even small programmable logic devices (PLDs), the mainstays of the 1970s and early 1980s, are rapidly disappearing. The burgeoning market for smaller, lower power, and more portable devices has driven high levels of integration into almost every product. This also has changed the nature of optimization; the focus is now on what goes into each chip rather than on the collection of individual gates needed to realize the design. The optimizations of today are more and more often made at the architecture level rather than in the switches.

Hardware designers now spend the majority of their time dealing with software. Specifically, the tools needed to efficiently map digital designs onto the emerging programmable devices that are growing ever more sophisticated. They capture their design specifications in software with description languages appropriate for describing the parallelism of hardware; they use software tools to simulate their designs and then to synthesize it into the implementation technology of choice. Design time is reduced radically as market pressures require products to be introduced quickly, at the right price and performance.

Although the overgrowing complexity of designs necessitates more powerful abstractions, the fundamentals haven't changed. In fact, the contemporary digital designer must have a broader understanding of the discipline of computation than ever before, including both hardware and software. In this second edition, we provide this broader perspective.

Changes from the First Edition

There are many changes from the first edition that can be grouped into four rough categories. First, we updated the hardware technologies discussed in the book. Second, we added a more complete, if nevertheless introductory, treatment of the software tools that are now so commonplace in the designer's tool kit. Third, we responded to the comments and suggestions received over the years by the many faculty and practitioners who have used the book. Finally, we rationalized the organization of the text so that concepts, technologies, tools, and practical matters were more clearly defined.

New Introduction

The introduction has been changed from one that focused on the process of design to one that introduces the concepts of computation, encoding, and sequencing. This sets out a better road map to the rest of the book and provides a rationale for its organization. Rather than discussing the design process in the abstract, we now include many more case studies to help the student gain that understanding by seeing the process in action.

Repartitioning of Material

Each of the two major sections on combinational and sequential logic was divided into a set of chapters. These first cover the fundamental concepts, then describe the principles of manipulating the logic into different forms, followed by a discussion of the optimizations and tools that are available, and concludes with an overview of the technologies available to build logic circuits. Each is capped by a set of comprehensive design case studies that make each of the issues concrete.

More Emphasis on Programmable Logic

We have added new material on the latest programmable logic technologies that have quickly become the dominant style for realizing digital designs. We do not attempt to provide all the information needed to work with any one technology. Those used will vary dramatically from institution to institution. Therefore, the book needs to be supplemented with a laboratory guide that covers the specifics of a particular installation.

In this text, we focused on the underlying concepts. We expect laboratory guides to be available in the form of web-based materials that can be easily customized to the variety already out there and updated as new technologies emerge.

Inclusion of Hardware Description Languages

HDLs are now given a more central role to reflect their total acceptance by the design community over the past 10 years. We describe only the basics of one of the dominant languages, namely Verilog, focusing on describing behavior, as well as covering the basics of HDL simulation models. We highlight the power of the languages in making designs more parameterizable and customizable and designers more efficient.

Design Case Studies

Nothing helps students learn design as much as designing for themselves. The next best thing is to provide a large collection of examples where the intuitions and rules of thumb are discussed explicitly. The hope is that this will help bootstrap new digital designers into the world of practical applications rather than the drill problems that were the norm in simpler times. There are many new and extensive design examples sprinkled throughout the text and in two large case study chapters focusing on combinational and sequential logic.

Elimination of Chapters on Datapath, Control, and Register-Transfer

We decided to remove the last two chapters of the first edition, that focused on datapath and register-transfer design, and a simple processor as an in-depth design case study of the interaction of control and datapath. While these topics are without a doubt important, on reflection we felt they are better left for a more extended study of digital design than could be included within the page limit of this edition. Instead, we chose more intensive coverage of programmable logic and HDLs, with extensive but smaller design examples spread throughout the text. We plan to make supplementary materials on the eliminated topics available on the web.

Navigating the Book

The book is organized into 10 chapters and three appendices. Chapter 1 is an overall introduction to the field. Chapters 2 through 5 cover combinational logic. Chapters 6 through 10 cover sequential logic. The three appendices provide some potentially useful background material that may have been part of other courses in a computer or electrical engineering curriculum.

Chapter 1 is an ambitious attempt to introduce many of the concepts of digital design through a short history of the evolution of digital hardware and two simple examples. Many may find that it introduces too many concepts too quickly for students to grasp their importance. However, this was not the intent. We fully expect students to be somewhat overwhelmed by the number of new concepts that come up in the discussion of the example. The purpose of the chapter is to provide an aerial view of the field so that students find it easier to see how the pieces they will see, in much greater detail and depth in later chapters, fit together coherently. It is certainly possible to replace this chapter with a more traditional introduction.

The next four chapters lay out the concepts of combinational logic design, closing with a set of comprehensive examples.

Chapter 2 covers the basics of combinational logic from simple gates to their time behavior. It lays out the concepts of two-level and multilevel logic and motivates why we would want to simplify logic. Some of the basic machinery for manipulating logic is presented with an emphasis on pencil-and-paper methods.

Chapter 3 delves into methods for working with combinational logic. It begins by describing the algorithms inside of today's CAD tools and ends with an overview of hardware description languages and uses Verilog to demonstrate key elements. Included is a discussion of the discrete simulation concepts that help to clarify the language constructs. There is probably not enough detail to make this book the sole resource for laboratory work with CAD tools. We wanted to keep the book focused on key concepts rather than on details of particular tools. It will need to be supplemented with appropriate manuals for the particular tools students will find in their own laboratories. This chapter also covers timing issues in more detail, including hazards and hazard-elimination strategies.

Chapter 4 presents the full range of implementation technologies available to the logic designer for combinational logic. It is paired with Chapter 9 that does the same for sequential logic. Chapter 4 starts with basic logic gates (as in the traditional TTL-based courses), but quickly progresses to programmable logic (PLDs and two-level forms) and then to field-programmable gate arrays. We also discuss other types of logic constructs such as tri-state and open-collector logic. Basic electronics to support this discussion are in Appendix B.

Chapter 5 culminates the combinational logic section of the book with seven examples of increasing complexity. We emphasize problem solving from the initial specification and have provided considerable discussion of how to transform an initial informal description of the problem into precise logical statements while keeping track of the assumptions that are being made. Our goal in this chapter is to show the range of logic design and how to judge design tradeoffs and take advantage of optimization opportunities.

The remaining five chapters do the same for sequential logic what the Chapters 2 through 5 did for combinational logic.

Chapter 6 begins this section by introducing the idea of circuits with feedback and how they can be analyzed. We develop the basic elements of sequential logic, latches, and flip-flops by recapitulating their evolution. This is coupled with a discussion of the timing methodologies that make it practical to build large sequential logic systems. These methodologies are illustrated with simple sequential systems of shift registers. The chapter concludes with a continuation of the exposition of hardware description languages started in Chapter 3 and extends it to basic sequential logic elements.

Chapter 7 covers the central concept of finite state machines. It begins by using counters as a simple form of FSM and then moves on to the basic Moore and Mealy models for organizing sequential behavior. Like Chapter 2, it concludes by motivating the various optimization opportunities.

Chapter 8 extends the basic ideas of Chapter 7 and expands on the details of FSM optimization by treating state minimization, state encoding, and FSM partitioning, in turn. Each of these is illustrated with examples that highlight the tradeoffs at each stage of optimization. An additional section at the end of the chapter provides some guidelines for structuring FSM descriptions in HDLs.

Chapter 9 concludes the discussion of implementation technologies. It recapitulates all the technologies used for combinational logic introduced in Chapter 4 but focusing on their sequential logic ants.

Chapter 10 is a large chapter with six comprehensive design examples that bring to practice all the concepts in the text. It begins with the sequential logic example from Chapter 1, now discussed in full detail, to tie back to the start of the text and ends with the serial transmission of characters from a keypad to display. The latter examples focus on the partitioning of design problems into communicating pieces along two dimensions: parallel state machines and partitioning into data-path and control.

The three appendices cover number systems, basic electronics, and flip-flop, types. The first two cover

concepts that students are likely to have already seen in mathematics, physics, electrical engineering, or computer science introductory courses. They are not intended to be extensive treatment of these topics but only provide the background most directly connected to the main topics of this text. The appendix on flip-flop types is provided for historical completeness.

The Complete Teaching Package

The material in this book easily fills a quarter-long course and can be comfortably covered in a semester-long course. In fact, it is likely that supplemental topics, governed by the place in the curriculum the semester-long course occupies, can and should be included. These could be: more in-depth discussion of CAD algorithms including their data structures, efficiency, and implementation; further discussion of design tradeoffs in a particular implementation technology such as FPGAs; a larger design problem that can serve as a term project to highlight issues of scale and debugging; and topics from computer organization emphasizing partitioning into data-path and control and optimizations of both. Of course, individual instructors may also find that re-ordering some of the material makes more sense in their environments. For example, it is certainly possible to proceed by following the two sections in parallel rather than serially. Chapter 2 plus 6 and 7 can be paired, followed by 3 and 8, then 4 and 9, with the larger design examples of 5 and 10 together at the end. Many topics can also be skipped altogether. For example, CAD tools and their algorithms may be relegated to another course. Similarly, HDLs do not need to be included if the design environment focuses on schematic-level design. In the technology dimension, FPGAs can be skipped as they may be included in a later course on more advanced design methods. Our goal in organizing the book was to make it easier to make these customizations.

Finally, we are making a wealth of supplementary material available to course instructors and students. Our publisher's web site includes:

- A set of CAD tools that supports all the concepts presented in this text;
- A comprehensive set of lecture slides;
- Samples of possible laboratory assignments and projects;
- Solutions to all the problems in the text; and
- Supplementary material on computer organization for those that include that material in their introductory logic design classes.

We hope you will agree with us that this second edition is a worthy successor to the first.

RANDY H. KATZ
GAETANO BORRIELLO

Find out the technique of doing something from several sources. Among them is this publication entitled **Contemporary Logic Design By Randy H. Katz**. It is a very well understood book Contemporary Logic Design By Randy H. Katz that can be a suggestion to read now. This advised publication is among the all fantastic Contemporary Logic Design By Randy H. Katz collections that remain in this site. You will certainly also discover various other titles and styles from numerous writers to look right here.