PREFACE

This book is intended as an introductory book for students in computer science, computer engineering and electrical engineering. Part One is suitable for a one semester course on *logic design*. It has no prerequisites, although the maturity attained through an introduction to engineering course or a first programming course would be helpful. Part Two contains an introduction to the design of computers, utilizing the material from Part One. The whole book could be used for a two-semester course, or selected parts could be used in a one-semester course.

The book stresses fundamentals. It teaches through a large number of examples. The philosophy of the author is that the only way to learn design is to do a large number of design problems. Thus, in addition to the numerous examples in the body of the text, each chapter has a set of Solved Problems, that is, problems and their solutions, a Chapter Test (with answers in Appendix C), and a large set of Exercises (with answers to Selected Exercises in Appendix B).

Although computer-aided tools are widely used for the design of large systems, the student must first understand the basics. The basics provide more than enough material for a first course. The sections on Hardware Design Languages (HDLs) in Chapters 4 and 7 provide some material for a transition to a second course based on one of the computer-aided tool sets. For the computer design material, we introduce a simplified HDL that enables us to describe the hardware without spending excessive time on the details of some of the commercial tools such as Verilog or VHDL.

Chapter 1, after brief introduction, gives an overview of number systems as it applies to the material of this book. (Those students who have studied this in an earlier course can skip this chapter.)

Chapter 2 discusses the discusses the steps in the design process for combinational systems and the development of truth tables. It then introduces switching algebra and the implementation of switching functions using common gates—AND, OR, NOT, NAND, NOR, Exclusive OR, and Exclusive NOR. We are only concerned with the logic behavior of the gates, not the electronic implementation.

For a more complete introduction to logic design, including laboratory experiments, see Marcovitz, Alan B., *Introduction to Logic Design, Second Edition*, McGraw-Hill, 2005.

Chapter 3 presents the simplification of combinational systems using the Karnaugh map. It provides methods for solving problems (up to 5 variables) with both single and multiple outputs.

Chapter 4 is concerned with the design of larger combinational systems. It introduces a number of commercially available larger devices, including adders, comparators, decoders, encoders and priority encoders, and multiplexers. That is followed by a discussion of the use of logic arrays—ROMs, PLAs and PALs for the implementation of medium scale combinational systems. Hardware Design Languages are introduced. Finally, two larger systems are designed.

Chapter 5 introduces sequential systems. It starts by examining the behavior of latches and flip flops. It then discusses techniques to analyze the behavior of sequential systems.

Chapter 6 introduces the design process for sequential systems. The special case of counters is studied next. Finally, the solution of word problems, developing the state table or state diagram from a verbal description of the problem is presented in detail.

Chapter 7 looks at larger sequential systems. It starts by examining the application of shift registers and counters. Then, PLDs (logic arrays with memory) are presented. Three techniques that are useful in the design of more complex systems, ASM diagrams, one-hot encoding, and HDLs for sequential systems, are discussed next. Finally, two examples of larger systems are presented.

Chapter 8 is concerned with computer organization. It discusses the basic structure of a computer and the various addressing modes and instruction types.

Chapter 9 looks at the logic required to move data in a large machine, and the structure of a controller. We introduce *Digital Design Language (DDL)*, a simple notation to describe the behavior of a digital system. We chose not to use one of the commercial systems because the amount of detail required to use it would detract from the fundamentals that we are teaching.

Chapter 10 deals with the details of the design of a sample computer, *MODEL*. We design the control sequence and examine the timing.

Chapter 11 describes some of the interaction of the CPU with the other components of a computer system, the primary and secondary memory and input/output controllers.

A feature of this text is the Solved Problems. Each Chapter (other than 11, which is primarily descriptive) has a large number of problems, illustrating the techniques developed in the body of the text, followed by a detailed solution of each problem. Students are urged to solve each problem (without looking at the solution) and then compare their solution with the one shown.

Each chapter contains a large set of exercises. Answers to a selection of these are contained in Appendix B. Solutions will be made available to instructors through the Web.

Each Chapter then concludes with a Chapter Test; answers are given in Appendix C.

The material of Part One can be taught easily in a four-credit semester course. Five-variable maps could be eliminated without loss of continuity. (It is only needed for a few examples in Chapter 4.) The coverage of the larger examples of Sections 4.8 and 7.7 could be reduced. Indeed, most of Chapter 7 could be omitted.

The material of Part Two would work in a second semester course, although some curricula cover computer organization in an earlier course. The material of Chapter 8 does not depend on Part One, but the remainder of the book does.

SUPPLEMENTS

In addition, the text is complemented by a wealth of supplemental resources presented through **McGraw-Hill's ARIS** (Assessment Review and Instruction System). ARIS makes homework meaningful-and manageable-for instructors and students.

Instructors can assign and grade text-specific homework within the industry's most robust and versatile homework management system.

Students can access multimedia learning tools and benefit from unlimited practice via algorithmic problems.

Go to aris.mhhe.com to learn more and register!

ACKNOWLEDGMENTS

I want to thank my wife, Allyn, for her encouragement and for enduring endless hours when I was closeted in my office working on the manuscript. Several of my colleagues at Florida Atlantic University have read parts of the manuscript and have taught from earlier drafts. Even more importantly, I want to thank my students who provided me with the impetus to write a more suitable text, who suffered through earlier drafts of the book, and who made many suggestions and corrections. The reviewers,

Michael Chelian, *California State University*, Long Beach

G.R. Dattatreya, *The University of Texas at Dallas*

Travis Doom, *Wright State University*

Clifford Fitzmorris, *The University of Oklahoma*

Donald Hung, *San Jose State University*

Cetin Kaya Koc, *Oregon State University*

Anselmo Lastra, *The University of North Carolina*
D.J. Lee, *Brigham Young University*
Hsien-Hsin Lee *Georgia Institute of Technology*
Alex Martinez, *The Ohio State University*
Jacob Savir, *New Jersey Institute of Technology*
Eric Schwartz, *University of Florida*

provided many useful comments and suggestions. The book is much better because of their efforts. Finally, the staff at McGraw-Hill, particularly Michael Hackett, Melinda Bilecki, and Kay Brimeyer have been indispensable in producing the final product, as has my copy editor, Linda Davoli.