Fundamentals of
Digital Logic and Microcomputer Design

Fundamentals of

Digital Logic and Microcomputer Design

Fifth Edition

M. RAFIQUZZAMAN, Ph.D.
Professor
California State Polytechnic University
Pomona, California
and
President
Rafi Systems, Inc.

WILEY-INTERSCIENCE
A JOHN WILEY & SONS, INC., PUBLICATION

Published by John Wiley & Sons, Inc., Hoboken, New Jersey.
Published simultaneously in Canada.

Library of Congress Cataloging-in-Publication Data:

Rafiquzzaman, Mohamed.
Fundamentals of digital logic and microcomputer design / M. Rafiquzzaman.—5th ed.
p. cm.
Includes bibliographical references and index.
ISBN 0-471-72784-9 (cloth)
1. Logic circuits. 2. Microcomputers—Design and construction. 3. Electronic digital computers—Circuits. I. Title.

TK7888.4.R34 2005
621.39'5—dc22 2004065974

Printed in the United States of America.

10 9 8 7 6 5 4 3 2 1

In memory of my beloved parents, who gave me tremendous support, encouragement, and guidance in achieving my career goals. I will always miss them.

To my wife, Kusum, and brother, Elan

Contents

Preface

In this book we cover all basic concepts of computer engineering and science, from digital logic circuits to the design of a complete microcomputer system in a systematic and simplified manner. We have endeavored to present a clear understanding of the principles and basic tools required to design typical digital systems such as microcomputers.

To accomplish this goal, the computer is first defined as consisting of three blocks: central processing unit (CPU), memory, and I/O. We point out that the CPU is analogous to the brain of a human being. Computer memory is similar to human memory. A question asked of a human being is analogous to entering a program into a computer using an input device such as a keyboard, and answering the question by the human is similar in concept to outputting the result required by the program to a computer output device such as a printer. The main difference is that human beings can think independently whereas computers can only answer questions for which they are programmed. Due to advances in semiconductor technology, it is possible to fabricate the CPU on a single chip. The result is the microprocessor. Intel's Pentium and Motorola's Power PC are typical examples of microprocessors. Memory and I/O chips must be connected to the microprocessor chip to implement a microcomputer so that these microprocessors will be able to perform meaningful operations.

We clearly point out that computers understand only 0's and 1's. It is therefore important that students be familiar with binary numbers. Furthermore, we focus on the fact that computers can normally only add. Hence, all other operations such as subtraction are performed via addition. This can be accomplished via two's-complement arithmetic for binary numbers. This topic is therefore also included, along with a clear explanation of signed and unsigned binary numbers.

As far as computer programming is concerned, assembly language programming is covered in this book for typical Intel and Motorola microprocessors. An overview of C, C++, and Java high-level languages is also included. These are the only high-level languages that can perform I/O operations. We point out the advantages and disadvantages of programming typical microprocessors in C and assembly languages.

Three design levels are covered in this book: device level, logic level, and system level. Device-level design, which designs logic gates such as AND, OR, and NOT using transistors, is included from a basic point of view. Logic-level design is the design technique in which logic gates are used to design a digital component such as an adder. Finally, system-level design is covered for typical Intel and Motorola microprocessors. Microcomputers have been designed by interfacing memory and I/O chips to these microprocessors.

Digital systems at the logic level are classified into two types of circuits, combinational and sequential. Combinational circuits have no memory whereas sequential circuits contain memory. Microprocessors are designed using both combinational and sequential circuits. Therefore, these topics are covered in detail. The fifth edition of this book contains an introduction to synthesizing digital logic circuits using popular hardware description languages such as Verilog and VHDL. These two languages are included in Appendices I and J, independently of each other in such a way that either Verilog or VHDL can be covered in a course without confusion.

The material included in this book is divided into three sections. The first section contains Chapters 1 through 5. In these chapters we describe digital circuits at the gate and flip-flop levels and describe the analysis and design of combinational and sequential circuits. The second section contains Chapters 6 through 8. Here we describe microcomputer organization/architecture, programming, design of computer instruction sets, CPU, memory, and I/O. The third section contains Chapters 9 through 11. These chapters contain typical 16-, 32-, and 64-bit microprocessors manufactured by Intel and Motorola. Future plans of Intel and Motorola are also included. Details of the topics covered in the 11 chapters of this book follow.

- Chapter 1 presents an explanation of basic terminologies, fundamental concepts of digital integrated circuits using transistors; a comparison of LSTTL, HC, and HCT IC characteristics, the evolution of computers, and technological forecasts.
- Chapter 2 provides various number systems and codes suitable for representing information in microprocessors.
- Chapter 3 covers Boolean algebra along with map simplification of Boolean functions. The basic characteristics of digital logic gates are also presented.
- Chapter 4 presents the analysis and design of combinational circuits. Typical combinational circuits such as adders, decoders, encoders, multiplexers, demultiplexers and, ROMs/PLDs are included.
- Chapter 5 covers various types of flip-flops. Analysis and design of sequential circuits such as counters are provided.
- Chapter 6 presents typical microcomputer architecture, internal microprocessor organization, memory, I/O, and programming concepts.
- Chapter 7 covers the fundamentals of instruction set design. The design of registers and ALU is presented. Furthermore, control unit design using both hardwired control and microprogrammed approaches is included. Nanomemory concepts are covered.
- Chapter 8 explains the basics of memory, I/O, and parallel processing. Topics such as main memory array design, memory management concepts, cache memory organization, and pipelining are included.
- Chapters 9 and 10 contain detailed descriptions of the architectures, addressing modes, instruction sets, I/O, and system design concepts associated with the Intel 8086 and Motorola MC68000.
- Chapter 11 provides a summary of the basic features of Intel and Motorola 32- and 64-bit microprocessors. Overviews of the Intel 80486/Pentium/Pentium Pro/Pentium II/Celeron/Pentium III, Pentium 4, and the Motorola 68030/68040/68060/PowerPC

(32- and 64-bit) microprocessors are included. Finally, future plans by both Intel and Motorola are discussed.

The book can be used in a number of ways. Because the materials presented are basic and do not require an advanced mathematical background, the book can easily be adopted as a text for three quarter or two semester courses. These courses can be taught at the undergraduate level in engineering and computer science. The recommended course sequence can be digital logic design in the first course, with topics that include selected portions from Chapters 1 through 5; followed by a second course on computer architecture/organization (Chapters 6 through 8). The third course may include selected topics from Chapters 9 through 11, covering Intel and/or Motorola microprocessors.

The audience for this book can also be graduate students or practicing microprocessor system designers in the industry. Portions of Chapters 9 through 11 can be used as an introductory graduate text in electrical/computer engineering or computer science. Practitioners of microprocessor system design in the industry will find more simplified explanations, together with examples and comparison considerations, than are found in manufacturers' manuals.

Because of increased costs of college textbooks, this book covers several topics including digital logic, computer architecture, assembly language programming, and microprocessor-based system design in a single book. Adequate details are provided. Coverage of certain topics listed below makes the book very unique:

i) A clear explanation of signed and unsigned numbers using computation of ($X^2/255$) as an example (Section 2.2). The same concepts are illustrated using assembly language programming with Intel 8086 microprocessor (Example 9.2), and Motorola 68000 microprocessor (Example 10.2).

ii) Clarification of packed vs. unpacked BCD (Section 2.3.2). Also, clear explanation of ASCII vs. EBCDIC using an ASCII keyboard and an EBCDIC printer interfaced to a computer as an example (Section 2.3.2); illustration of the same concepts via Intel 8086 assembly language programming using the XLAT instruction (Section 9.5.1).

iii) Simplified explanation of Digital Logic Design along with numerous examples (Chapters 2 through 5). A clear explanation of the BCD adder (Section 4.5.1). An introduction to basic features of Verilog (Appendix I) and VHDL (Appendix J) along with descriptions of several examples of Chapters 3 through 5. Verilog and VHDL descriptions and syntheses of an ALU and a typical CPU. Coverage of Verilog and VHDL independent of each other in separate appendices without any confusion.

iv) CD containing a step by step procedure for installing and using Altera Quartus II software for synthesizing Verilog and VHDL descriptions of several combinational and sequential logic design. Screen shots included in CD providing the waveforms and tabular forms illustrating the simulation results.

v) Application of C language vs. assembly language along with advantages and disadvantages of each (Section 6.6.4).

vi) Numerous examples of assembly language programming for both Intel 8086 (Chapter 9) and Motorola 68000 (Chapter 10).

vii) A CD containing a step by step procedure for installing and using MASM 6.11

(8086) and 68asmsim (68000). Screen shots are provided on CD verifying the correct operation of several assembly language programs (both 8086 and 68000) via simulations using test data. The screen shots are obtained by simulating the assembly language programs using DEBUG (8086) and SIM (68000).

viii) A concise and simplified explanation of system design concepts including programmed I/O and interrupts with the Intel 8086 (Chapter 9) and Motorola 68000 (Chapter 10). Hardware aspects including design of reset circuitry and a simple microcomputer with these microprocessors from the chip level.

ix) A simplified comparison of RISC vs. CISC relating to Pentium architecture which is comprised of both RISC and CISC (Section 7.3.5). Unique feature of the PowerPC (Section 11.7.4).

The author wishes to express his sincere appreciation to his students, Rami Yassine, Teren Abear, Vireak Ly, Henry Zhong, Roel Delos Reyes, Vu Tran, Henry Ongkoputro, Rega Setiawan, Xibin Wu, Ryan DeGuzman, Angelo Terracina, Javier Ruiz, Yi Ting Huang, Eric Fang, Cindy Yeh, King Lam, Luis Galdamez, Elias Younes, Beniamin Petreaca, and to all others for making constructive suggestions. The author is indebted to his colleagues Dr. R. Chandra, Dr. M. Davarpanah, Dr. T. Sacco, Dr. S. Monemi, and Dr. H. El Naga of California State Poly University, Pomona for their valuable comments. The author is also grateful to Dr. W. C. Miller of University of Windsor, Canada and to his good friends U.S. Congressman Duke Cunningham (TOPGUN, Vietnam) and U.S. Congressman Jerry Weller for their inspiration during the writing effort. Finally, the author would like to thank CJ Media of California for preparing the final version of the manuscript.

M. RAFIQUZZAMAN

Pomona, California