
PRACTICAL DIGITAL DESIGN USING ICs

SECOND EDITION

Joseph D. Greenfield

PRACTICAL DIGITAL DESIGN USING ICs

SECOND EDITION

Joseph D. Greenfield

ROCHESTER INSTITUTE OF TECHNOLOGY

JOHN WILEY & SONS

NEW YORK CHICHESTER BRISBANE TORONTO SINGAPORE

MANUSCRIPT EDITOR: Dex Ott
PRODUCTION SUPERVISOR: Deborah Pokos
COVER DESIGN: Jules Perlmutter

Copyright © 1977, 1983, by John Wiley & Sons, Inc.

All rights reserved. Published simultaneously in Canada.

Reproduction or translation of any part of
this work beyond that permitted by Sections
107 and 108 of the 1976 United States Copyright
Act without the permission of the copyright
owner is unlawful. Requests for permission
or further information should be addressed to
the Permissions Department, John Wiley & Sons

Library of Congress Cataloging in Publication Data:

Greenfield, Joseph D., 1930-

Practical digital design using ICs

(Electronic technology series, ISSN 0422-910X)

Includes bibliographical references and index

1. Digital integrated circuits 2. Digital electronics
3. Logic circuits I Title II Title Practical
digital design using ICs III Series
TK7874.G72 1983 621.3815'3 82-10931
ISBN 0-471-05791-6

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

PREFACE

The purpose of the second edition of *Practical Digital Design Using ICs* is to update the material in the first edition. The general thrust of the first edition—explaining the **7400** series of ICs—has been retained because it was well received, and because the **7400** is still the most popular and widely used IC series.

The major event in the digital field in the past five years has been the proliferation of microprocessors. I have accommodated this event in the second edition by providing a chapter on microprocessor interfacing and by emphasizing 3-state gates and line drivers and transceivers. Many new and more sophisticated ICs that work with microprocessors, such as UARTs and PIAs, have also been produced, and these are covered in as much detail as space and their importance allow.

Though the microprocessor seems to be ubiquitous, the use of small and medium scale ICs is even more widespread. Microprocessors will not soon replace ICs. Instead, they will work in conjunction with ICs to provide more sophisticated and powerful systems. As I point out in Chapter 1, the Apple computer uses about 50 ICs, most of which are discussed in this book, in addition to its **6502** microprocessor. Thus the study of ICs remains a necessary part of the curriculum for digital engineers and technologists.

Chapter 1 is an introduction to binary arithmetic. 2s complement and hexadecimal arithmetic are introduced in Chapter 14, where they can be presented in conjunction with the arithmetic circuits that actually use them.

An introduction to the basic IC gates has been moved into Chapter 2 to facilitate laboratory work with the Boolean algebra presented in the chapter. I have added a section on multiple outputs to Chapter 3 on reduction and Karnaugh maps. In Chapter 5 the material on 3-state gates has been expanded because of their widespread use.

Chapters 6, 7, 8, and 9 on flip-flops, one-shots, counters, and shift registers all have been modernized and some new material is introduced. Greater use of oscilloscope traces has been made where they can best illustrate a point.

In Chapter 10 on IC construction and testing, I have added a section on logic analyzers. The chapter on multiplexers has been moved to Chapter 11, so it can precede the chapters on multiplexing 7-segment displays and arithmetic circuits.

Chapter 15 has been updated to include more commonly used memories. The sections on dynamic RAMs and EPROMs have been expanded.

Chapter 16 is an entirely new chapter explaining how a digital computer works and how it can be built using the ICs introduced in the previous chapters. A very small computer, built by RIT students, is also introduced and explained. Making students build their own computer is, I feel, the best way to be sure they understand how computers work.

Chapter 17 is on minicomputer interfacing, and Chapter 19, which is new, examines microprocessor interfacing. In Chapter 17, the material on A/D converters has been expanded.

Chapter 18 discusses RS-232 and 20-mA current loop conversion. A detailed discussion of a commonly used UART has been included.

Chapter 20 is on the CRT Display Generator. It has been updated to conform to modern practice. I have added appendixes on the ASCII code and answers to selected problems. The latter has been requested by many people.

I would like to thank all the people who helped me with the second edition. I must mention my colleagues in the Department of Electronic Technology at Rochester Institute of Technology, and my students there, whose input kept my courses up to date. I must also thank the people at John Wiley, who kept pushing and prodding to get the manuscript done, and the pre-publication reviewers, who were very kind and enthusiastic, and provided many excellent suggestions for improvement.

I especially thank my wife Gladys. Without her love, encouragement, co-operation, and typing, this book would never have been written.

JOSEPH D. GREENFIELD

Rochester, New York

CONTENTS

1	BINARY NUMBERS	1
1-1	Introduction to Digital Electronics	1
1-2	Instructional Objectives	3
1-3	Self-Evaluation Questions	3
1-4	Uses of Binary Bits	3
1-5	Binary-to-Decimal Conversion	5
1-6	Decimal-to-Binary Conversion	8
1-7	Addition and Subtraction of Binary Numbers	14
	Glossary	16
	References	17
	Problems	17
2	BOOLEAN ALGEBRA	19
2-1	Instructional Objectives	19
2-2	Self-Evaluation Questions	19
2-3	Introduction to Boolean Algebra	20
2-4	Operations with Boolean Variables	23
2-5	Theorems	25
2-6	Logical Manipulation	28
2-7	IC Gates	32
2-8	Complementation of Functions	37
2-9	Inverting Gates	41
2-10	Implementation of Logic Expressions Using Gates	45
2-11	Relays	52
	Glossary	55
	References	56
	Problems	56
3	SYSTEMATIC REDUCTION OF BOOLEAN EXPRESSIONS	60
3-1	Instructional Objectives	60
3-2	Self-Evaluation Questions	60
3-3	Standard Forms—Introduction	61
3-4	Equivalence of SOP and POS Standard Forms	72

iii CONTENTS

3-5	Karnaugh Maps	74
3-6	Don't Cares	91
3-7	Five-Variable Maps	93
3-8	Practical Examples	95
	Summary	101
	Glossary	101
	References	102
	Problems	102
4	LOGIC FAMILIES AND THEIR CHARACTERISTICS	105
4-1	Instructional Objectives	105
4-2	Self-Evaluation Questions	105
4-3	Evaluation of IC Families	106
4-4	Transistor-Transistor Logic	109
4-5	Characteristics of TTL Gates	111
4-6	Emitter-Coupled Logic	116
4-7	Complementary Metal Oxide Semiconductor Gates	118
	Summary	122
	Glossary	123
	References	123
5	BASIC TTL GATE	125
5-1	Instructional Objectives	125
5-2	Self-Evaluation Questions	125
5-3	Introduction	126
5-4	Schmitt Triggers	127
5-5	Open and Unused Inputs	130
5-6	Wire-ANDing and Open Collector Gates	132
5-7	Three-State Devices	139
5-8	Strobed Gates, Expandable Gates, and Expanders	143
5-9	AND-OR-INVERT Gates	147
5-10	The EXCLUSIVE-OR Gate	151
	Summary	152
	Glossary	153
	References	154
	Problems	154
6	FLIP-FLOPS	158
6-1	Instructional Objectives	158
6-2	Self-Evaluation Questions	158
6-3	Introduction	159
6-4	The Basic Flip-Flop	159
6-4	NOR gate Flip-Flops	160
6-6	NAND Gate Flip-Flops	162

6-7	D-type Flip-Flops	165
6-8	Bistable Latches	167
6-9	J-K Master-Slave Flip-Flops	167
6-10	Edge-Triggered Flip-Flops	172
6-11	Timing Charts	173
6-12	Direct SETS and Direct CLEARS	175
6-13	Race Conditions	176
6-14	Flip-Flop Parameters	178
6-15	Uses of Flip-Flops	179
6-16	Synchronizing Flip-Flops	186
6-17	Glitches	187
	Summary	189
	References	189
	Problems	190
7	ONE-SHOTS	195
7-1	Instructional Objectives	195
7-2	Self-Evaluation Questions	195
7-3	Introduction to One-Shots	196
7-4	The 74121 One-Shot	197
7-5	Retriggerable One-Shots	201
7-6	Integrated Circuit Oscillators	207
7-7	Timing Generation Problems	215
7-8	Switch Bounce	217
7-9	Debouncing Class A Switches	221
7-10	The One-Shot Discriminator	224
	Summary	225
	Glossary	225
	References	226
	Problems	226
8	COUNTERS	230
8-1	Instructional Objectives	230
8-2	Self-Evaluation Questions	230
8-3	Introduction	231
8-4	Divide-by-N Circuits	231
8-5	Ripple Counters	232
8-6	Synchronous Counters	234
8-7	The 3s Counters	238
8-8	Irregular and Truncated Count Sequences	239
8-9	IC Counters	45
8-10	UP-DOWN Counters	2
8-11	Divide-by-N Circuits Using Counters	7
	Summary	62

x CONTENTS

Glossary	262
References	263
Problems	263
9 SHIFT REGISTERS	267
9-1 Instructional Objectives	267
9-2 Self-Evaluation Questions	267
9-3 The Basic Shift Register	268
9-4 LEFT-RIGHT Shift Registers	270
9-5 Serial Inputs and Parallel Loading of Shift Registers	273
9-6 Parallel Load and Parallel Output Shift Registers	275
9-7 Applications	283
9-8 MOS Shift Registers	288
Summary	293
Glossary	293
References	293
Problems	294
10 CONSTRUCTION AND DEBUGGING OF IC CIRCUITS	297
10-1 Instructional Objectives	297
10-2 Self-Evaluation Questions	297
10-3 Wire Wrapping	298
10-4 Printed Circuits	301
10-5 Construction of Wire-Wrap Circuits	303
10-6 Error Detection in Combinatorial Circuits	310
10-7 Error Detection in Sequential Circuits	312
10-8 Logic Analyzers	316
Summary	321
Glossary	322
References	322
11 MULTIPLEXERS AND DEMULTIPLEXERS	323
11-1 Instructional Objectives	323
11-2 Self-Evaluation Questions	323
11-3 Multiplexers	324
11-4 Demultiplexers	332
11-5 Practical Applications	336
Summary	344
Glossary	344
References	344
Problems	344
12 BINARY CODED DECIMAL	346
12-1 Instructional Objectives	346

12-2	Self-Evaluation Questions	346
12-3	Expressing Numbers in Binary Coded Decimal	347
12-4	Conversion Using Algorithms	349
12-5	Conversion Using ICs	355
12-6	Indicating Lights	362
12-7	Multiplexed Displays	369
12-8	Liquid Crystal Displays	370
	Summary	371
	Glossary	371
	References	371
	Problems	372
13	EXCLUSIVE OR CIRCUITS	374
13-1	Instructional Objectives	374
13-2	Self-Evaluation Questions	374
13-3	Comparison Circuits	375
13-4	Parity Checking and Generation	380
13-5	Parity Checking and Generation Using the 74180	387
13-6	More Sophisticated Error-Correcting Routines	390
13-7	The Gray Code	393
13-8	Liquid Crystal Displays	398
	Summary	398
	Glossary	399
	References	399
	Problems	400
14	ARITHMETIC CIRCUITS	402
14-1	Instructional Objectives	403
14-2	Self-Evaluation Questions	403
14-3	The Basic Adder	403
14-4	Subtraction	406
14-5	2s Complement Arithmetic	409
14-6	Hexadecimal Arithmetic	414
14-7	The 7483 4-Bit Adder	420
14-8	Overflow and Underflow in 2s Complement Arithmetic	425
14-9	BCD Arithmetic	427
14-10	Arithmetic/Logic Units	431
14-11	Look-Ahead Carry	435
14-12	Binary Multiplication	438
14-13	Arithmetic Processing Units	444
	Summary	445
	Glossary	445
	References	446
	Problems	446

xii CONTENTS

15	MEMORIES	452
15-1	Instructional Objectives	452
15-2	Self-Evaluation Questions	452
15-3	Memory Concepts	453
15-4	Core Memories	455
15-5	Introduction to Semiconductor Memories	459
15-6	Bipolar RAMs	461
15-7	MOS Memories	468
15-8	Dynamic RAMs	472
15-9	Read Only Memories	480
15-10	IC ROMs	483
	Summary	487
	Glossary	487
	References	488
	Problems	488
16	THE BASIC COMPUTER	491
16-1	Instructional Objectives	491
16-2	Self-Evaluation Questions	491
16-3	Introduction to the Computer	492
16-4	Flowcharts	496
16-5	Branch Instructions and Loops	498
16-6	The Control Unit	505
16-7	The Hardware Design of a Computer	509
16-8	The Complete Computer	514
16-9	Building the Computer	517
	Summary	529
	Glossary	529
	References	530
	Problems	531
17	COMPUTER INTERFACES	532
17-1	Instructional Objectives	532
17-2	Self-Evaluation Questions	532
17-3	Introduction to Computer Interfaces	533
17-4	The I-O Bus	534
17-5	Direct Memory Accesses	546
17-6	Interrupts	552
17-7	Communication Between the Analog and Digital World	557
	Summary	565
	Glossary	566
	References	567
	Problems	568

18 MODEMS AND TELETYPES	570
18-1 Instructional Objectives	570
18-2 Self-Evaluation Questions	570
18-3 Introduction to MODEMs	571
18-4 Low Speed MODEMs	572
18-5 High Speed MODEMs	575
18-6 Teletypes	581
18-7 UARTs	586
Summary	592
Glossary	592
References	593
Problems	593
19 MICROPROCESSOR INTERFACING	595
19-1 Instructional Objectives	595
19-2 Self-Evaluation Questions	595
19-3 Introduction to Microprocessor Interfacing	596
19-4 Input and Output Ports	601
19-5 Interrupts on the 8080 and 8085	605
19-6 Memory Mapped I-O	612
19-7 The Motorola PIA	613
19-8 CA2 and CB2	617
19-9 Interrupts on the 6800	620
19-10 Other Interface ICs	622
19-11 The SDK-85 Kit	623
19-12 Other Microprocessor Busses	625
Summary	626
Glossary	626
References	627
Problems	627
20 DISPLAY GENERATORS	629
20-1 Instructional Objectives	629
20-2 Self-Evaluation Questions	630
20-3 Introduction to the Display Generator	630
20-4 The Raster-Scan Display Generator	632
20-5 Components of the Display Generator	637
20-6 Timing the Display Generator	643
20-7 The Cursor	649
20-8 CRT Controllers	652
Summary	653
Glossary	654
References	655
Problems	655

xiv CONTENTS

APPENDIX A TABLE OF POWERS OF 2	659
APPENDIX B RESISTOR CALCULATIONS FOR OPEN COLLECTOR ICs	660
APPENDIX C REDUCTION OF EQUATION 12-2 TO EXCLUSIVE ORs	661
APPENDIX D PROOF OF EXCLUSIVE OR-PARITY RELATIONSHIP	662
APPENDIX E THE ASCII CODE	663
APPENDIX F ANSWERS TO SELECTED PROBLEMS	664
INDEX	709
SUPPLEMENTARY INDEX	715

CHAPTER

1

BINARY NUMBERS

1-1 INTRODUCTION TO DIGITAL ELECTRONICS

This book introduces the reader to digital electronics and enables him or her to understand, design, and construct digital circuits. The book is primarily concerned with the functions and uses of digital integrated circuits (ICs) and emphasizes the 7400 TTL (transistor transistor logic) series. These ICs have experienced a phenomenal growth in the past several years and millions of them have been manufactured. They can be purchased from manufacturers, electronics distributors, electronics stores such as Radio Shack, many computer stores, and by mail or phone from discount houses.¹ Their low price and wide availability have caused them to be designed into almost all digital circuits produced in the last 10 years.

Microprocessors (μ Ps) have also enjoyed phenomenal growth recently. Some engineers believed that μ Ps would replace ICs, but this has not happened. Rather, μ Ps work in conjunction with digital ICs to provide a high level of intelligence and sophistication to digital hardware. The Apple computer, for example, uses a 6502 μ P as its main computing element. It also uses about 50 TTL ICs (mostly types described in this book), plus ROMs and RAMs. Thus even with a μ P in the system, many small and medium scale ICs are required to support it.

¹These discount houses often advertise in magazines such as *Popular Electronics* or *Byte*.

2 BINARY NUMBERS

Both standard and special purpose ICs are also needed to enable a μ P to interface or communicate with other devices such as disks and printers. Some of these special purpose ICs and interaction between μ Ps and ICs is covered in the later chapters of this book.

Both ICs and μ Ps are the results of advances in *digital electronics*. Modern digital electronics is the product of a marriage between switching theory (a discipline in its infancy in the 1950s) and integrated circuits (a family of devices that was not developed until the 1960s). However, digital circuits have proven to be the best way to construct many useful devices, including the digital computer; consequently, its growth has been rapid, and a significant percentage of the jobs currently available in electrical engineering require digital training.

The output of most electronic circuits is an *analog quantity*, typically a voltage. An analog quantity is a quantity that may assume any numerical value within the range of possible outputs. Therefore, an electronic circuit is capable of producing many outputs in response to different inputs. For example, 5.12 volts (V) might be one output, 3.76 V another; any voltage within the precision of the voltmeter is a legitimate output. *Digital circuits* depart from analog circuits by providing only two values as an output: an output can either be a one (1) or a zero (0), and *nothing else*. Of course, digital engineers are still dealing with electronic circuits whose outputs are voltages. What they have done is to *define* a certain range of voltages as a logic 1 and another range of voltages as a logic 0. Typically, the 1 and 0 ranges are separated by a *forbidden* range of voltages. TTL integrated circuits define a 1 output as any voltage between 2 and 5 V and a 0 as any voltage between ground (zero) and 0.8 V. But what if the actual output of the circuit is in a forbidden or undefined range, 1 V, for instance? "Then," says the digital engineer succinctly, "the circuit is malfunctioning. Fix it." Methods of diagnosing and repairing malfunctions in digital circuits are discussed in Chapter 10.

Advances in digital technology have been more spectacular than advances in switching theory. The first computers were built in the early 1950s using large vacuum tubes that had low reliability and consumed a great deal of power. Tubes were replaced by transistors as soon as the latter became available and reliable. Engineers quickly found ways to build small, efficient, and inexpensive digital circuits using a single transistor, a couple of diodes, and a few resistors. Now, however, these discrete components (the resistors, transistors, and diodes) have all disappeared inside the IC. Recently ICs have become slightly larger and far more complex. The microprocessor itself is an example of a very large scale (VLSI) integrated circuit containing several thousand gates.

The first three chapters of this book develop the required switching theory. The remaining chapters describe the behavior of ICs, predominantly the 7400 TTL series, and ways in which to design useful circuits with them. These circuits can then be used to build computers, or *controllers* and *interfaces* that allow other devices to communicate with a computer. Examples of computer interfaces and controller design are presented in the later chapters of this book.

1-2 INSTRUCTIONAL OBJECTIVES

This first chapter acquaints the student with the binary number system and gives some facility in handling binary numbers. After reading this chapter, the student should be able to:

1. Convert binary numbers to decimal numbers.
2. Convert decimal numbers to binary numbers.
3. Find the sum and difference of two binary numbers.

1-3 SELF-EVALUATION QUESTIONS

As the student reads the chapter, he or she should be able to answer the following questions:

1. What is the difference between digital and analog circuits?
2. How are the outputs of a circuit defined to make the output digital?
3. What are the advantages of digital circuits?
4. What is the difference between a *bit* and a *decimal digit*? In what respects are they similar?
5. In a *flow chart*, what is the function of the rectangular and diamond-shaped boxes?
6. How are binary addition and subtraction different from decimal addition and subtraction? How are they similar?

1-4 USES OF BINARY BITS

The output of a digital circuit is a single binary digit (a 1 or a 0) commonly called a *bit*. There are two advantages gained by restricting the output of an electronic circuit to one of two possible values. First, it is rarely necessary to make fine distinctions. Whether an output is 3.67 or 3.68 V no longer matters; in both cases it is a logic 1. Since well-designed logic circuits produce voltages near the middle of the range defined for 1 or 0, there is no difficulty in distinguishing between them. In addition, a digital circuit is very tolerant of any drift in the output caused by component aging or changes. A change in a component would almost have to be catastrophic

4 BINARY NUMBERS

to cause the output voltage to drift from a 1 to a 0 or an undefined value. The second advantage is that it is far easier for electronic circuits to remember a 1 or a 0 than to remember an analog quantity like 3.67 V. Since all but the simplest digital circuits require the ability to remember the value of a voltage *after the conditions that caused that voltage have disappeared*, this is a very important consideration.

The output of a single digital circuit, a single bit, is enough to answer any question that has *only two* possible answers. For example, a typical job application might ask "What is your sex?" A 1 could arbitrarily be assigned to a male and a 0 to a female, so a single bit is enough to describe the answer to this question. A single bit is all the space a programmer needs to reserve in his computer for this answer.

However, another question on the job application might be "What is the color of your hair?" If the possible answers are black, brown, blonde, and red, a single bit cannot possibly describe them all. Now *several* bits are needed to describe all possible answers. We could assign one bit to each answer (i.e., brown = 0001, black = 0010, blonde = 0100, red = 1000), but if there are many possible answers to the given question many bits are required. The coding scheme presented above is not optimum; it requires more bits than are really necessary to answer the question.

It is most economical to use *as few bits as possible* to express the answer to a question, or a number, or a choice. The crucial question is:

"What is the minimum number of bits required to distinguish between n different things?"

Whether these n things are objects, possible answers, or n numbers is immaterial. To answer this question we realize that each bit has two possible values. Therefore, k bits would have 2^k possible values. This fact is the basis for Theorem 1.

Theorem 1

The minimum number of bits required to express n different things is k , where k is the smallest number such that $2^k \geq n$.

A few examples should make this clear.

EXAMPLE 1-1

What is the minimum number of bits required to answer the hair color question, and how could they be coded to give distinct answers?