Pearson International Edition

NINTH EDITION

Digital Fundamentals

FLOYD

SOFTWARE ENCLOSED

DIGITAL FUNDAMENTALS

Ninth Edition

Thomas L. Floyd

PEARSON
Prentice Hall

Pearson Education International

Acquisitions Editor: Kate Linsner
Production Editor: Rex Davidson
Design Coordinator: Diane Ernsberger
Editorial Assistant: Lara Dimmick
Cover Designer: Jason Moore
Cover art: Getty One
Production Manager: Matt Ottenweller
Marketing Manager: Ben Leonard
Illustrations: Jane Lopez

This book was set in Times Roman by Carlisle Communications, Ltd. and was printed and bound by Courier Kendallville, Inc. The cover was printed by Coral Graphic Services, Inc.

Chapter opening photos and special feature photos by Getty Images.

Pearson Education Ltd.
Pearson Education Singapore Pte. Ltd.
Pearson Education Canada, Ltd.
Pearson Education—Japan
Pearson Education, Upper Saddle River, New Jersey
Pearson Education Australia Pty. Limited
Pearson Education North Asia Ltd.
Pearson Educación de Mexico, S.A. de C.V.
Pearson Education Malaysia Pte. Ltd.

PEARSON
Prentice Hall

10 9 8 7 6 5 4 3 2
ISBN: 0-13-197255-3

Preface

Welcome to *Digital Fundamentals, Ninth Edition*. A strong foundation in the core fundamentals of digital technology is vital to anyone pursuing a career in this exciting, fast-paced industry. This text is carefully organized to include up-to-date coverage of topics that can be covered in their entirety, used in a condensed format, or omitted altogether, depending upon the course emphasis.

The topics in this text are covered in the same clear, straightforward, and well-illustrated format that has been so successful in the previous editions of *Digital Fundamentals*. Many topics have been strengthened or enhanced and numerous improvements can be found throughout the book.

You will probably find more topics than you can cover in a single course. This range of topics provides the flexibility to accommodate a variety of program requirements. For example, some of the design-oriented or system application topics may not be appropriate in some courses. Other programs may not cover programmable logic, while some may not have time to include topics such as computers, microprocessors, or digital signal processing. Also, in some courses there may be no need to go into the details of "inside-the-chip" circuitry. These and other topics can be omitted or lightly covered without affecting the coverage of the fundamental topics. A background in transistor circuits is not a prerequisite for this textbook although coverage of integrated circuit technology (inside-the-chip circuits) is included in a "floating chapter," which is optional.

Following this Preface is a color-coded table of contents to indicate a variety of approaches for meeting most unique course requirements. The text has a modular organization that allows inclusion or omission of various topics without impacting the other topics that are covered in your course. Because programmable logic continues to grow in importance, an entire chapter (Chapter 11) is devoted to the topic, including PALs, GALs, CPLDs, and FPGAs; specific Altera and Xilinx devices are introduced. Also a generic introduction to programmable logic software is provided and boundary scan logic is covered.

New in This Edition

- The Hamming error detecting and correcting code
- Carry look-ahead adders
- A brief introduction to VHDL
- Expanded and improved coverage of test instruments
- An expanded and reorganized coverage of programmable logic
- Improved troubleshooting coverage
- New approach to Digital System Applications

Features

- Full-color format
- Margin notes provide information in a very condensed form.
- Key terms are listed in each chapter opener. Within the chapter, the key terms are in boldface color. Each key term is defined at the end of the chapter, as well as at the end of the book in the comprehensive glossary along with other glossary terms that are indicated by black boldface in the text.

- Chapter 14 is designed as a "floating chapter" to provide optional coverage of IC technology (inside-the-chip circuitry) at any point in your course.
- Overview and objectives in each chapter opener
- Introduction and objectives at the beginning of each section within a chapter
- Review questions and exercises at the end of each section in a chapter
- A Related Problem in each worked example
- Computer Notes interspersed throughout to provide interesting information about computer technology as it relates to the text coverage
- Hands-On Tips interspersed throughout to provide useful and practical information
- The Digital System Application is a feature at the end of many chapters that provides interesting and practical applications of logic fundamentals.
- Chapter summaries at the end of each chapter
- Multiple-choice self-test at the end of each chapter
- Extensive sectionalized problem sets at the end of each chapter include basic, troubleshooting, system application, and special design problems.
- The use and application of test instruments, including the oscilloscope, logic analyzer, function generator, and DMM, are covered.
- Chapter 12 provides an introduction to computers.
- Chapter 13 introduces digital signal processing, including analog-to-digital and digital-to-analog conversion.
- Concepts of programmable logic are introduced beginning in Chapter 1.
- Specific fixed-function IC devices are introduced throughout.
- Chapter 11 provides a coverage of PALs, GALs, CPLDs and FPGAs as well as a generic coverage of PLD programming.

- Selected circuit diagrams in the text, identified by the special icon shown here, are rendered in Multisim® 2001 and Multisim® 7, and these circuit files are provided on the enclosed CD-ROM. These files (also available on the Companion Website at **www.prenhall.com/floyd**) are provided at no extra cost to the consumer and are for use by anyone who chooses to use Multisim software. Multisim is widely regarded as an excellent simulation tool for classroom and laboratory learning. However, successful use of this textbook is not dependent upon use of the circuit files.
- Boundary scan logic associated with programmable devices is introduced in Chapter 11.
- In addition to boundary scan, troubleshooting coverage includes methods for testing programmable logic, such as traditional, bed-of-nails, and flying probe. Boundary scan and these other methods are important in manufacturing and industry.
- For those who wish to include ABEL programming, an introduction is provided on the Companion Website at **www.prenhall.com/floyd.**

Accompanying Student Resources

- *Experiments in Digital Fundamentals*, a laboratory manual by David M. Buchla. Solutions for this manual are available in the Instructor's Resource Manual.
- Two CD-ROMs included with each copy of the text:
  Circuit files in Multisim for use with Multisim software
  Texas Instruments digital devices data sheets

Instructor Resources

- *PowerPoint® slides.* These presentations feature Lecture Notes and figures from the text. (On CD-ROM and online.)
- *Companion Website.* (**www.prenhall.com/floyd**). For the instructor, this website offers the ability to post your syllabus online with our Syllabus Manager™. This is a great solution for classes taught online, that are self-paced, or in any computer-assisted manner.
- *Instructor's Resource Manual.* Includes worked-out solutions to chapter problems, solutions to Digital System Applications, a summary of Multisim simulation results, and worked-out lab results for the lab manual by David M. Buchla. (Print and online.)
- *Test Item File.* This edition of the Test Item File features over 900 questions.
- *TestGen.®* This is an electronic version of the Test Item File, enabling instructors to customize tests for the classroom.

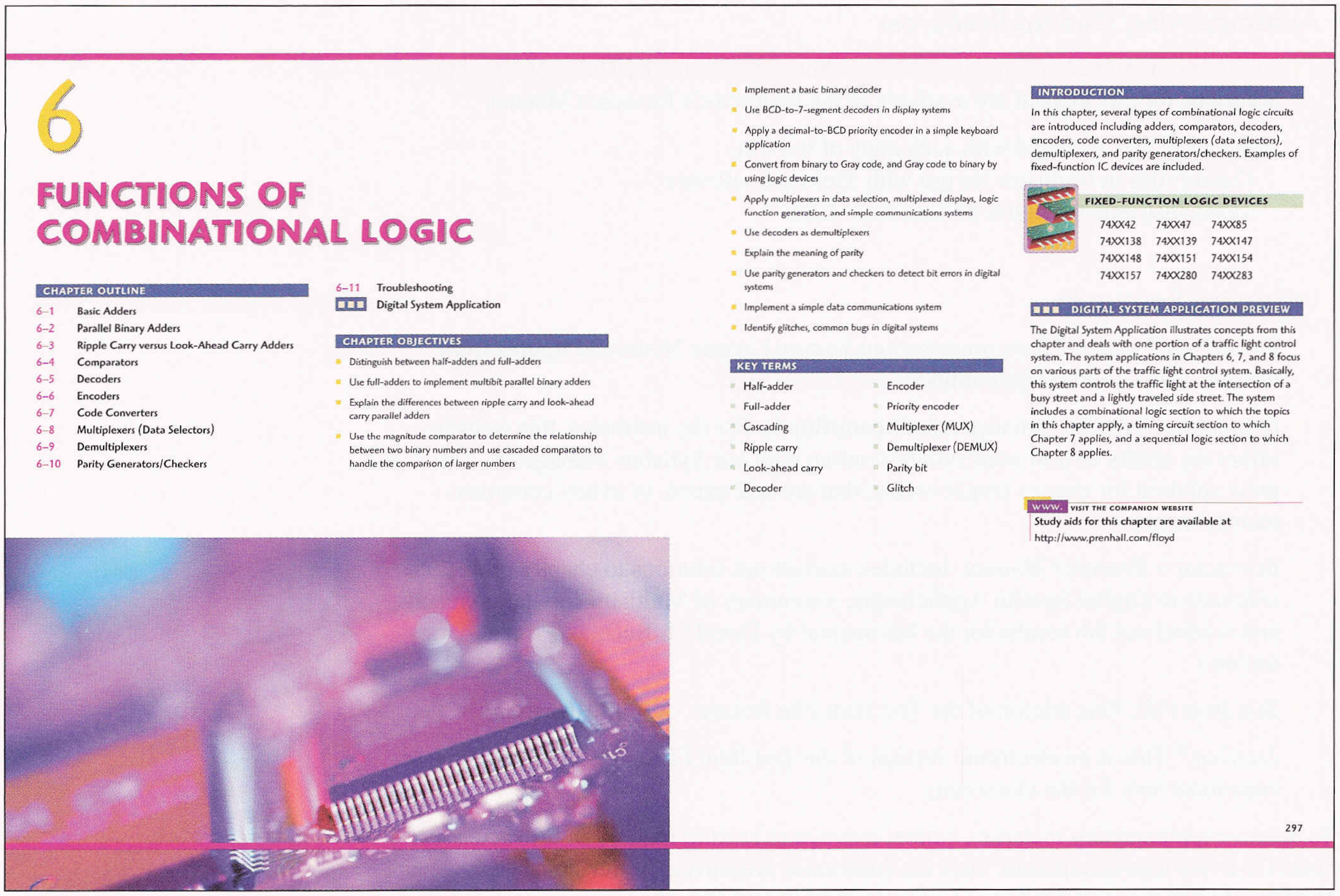
To access supplementary materials online, instructors need to request an instructor access code. Go to **www.prenhall.com**, click the **Instructor Resource Center** link, and then click **Register Today** for an instructor access code. Within 48 hours after registering you will receive a confirming e-mail including an instructor access code. Once you have received your code, go to the website and log on for full instructions on downloading the materials you wish to use.

Illustration of Chapter Features

Chapter Opener Each chapter begins with a two-page spread, as shown in Figure P–1. The left page includes a list of the sections in the chapter and a list of chapter objectives. A typical right page includes an overview of the chapter, a list of specific devices introduced in the chapter (each new device is indicated by an IC logo at the point where it is introduced), a brief Digital System Application preview, a list of key terms, and a website reference for chapter study aids.

Section Opener Each of the sections in a chapter begins with a brief introduction that includes a general overview and section objectives. An illustration is shown in Figure P–2.

Section Review Each section ends with a review consisting of questions or exercises that emphasize the main concepts presented in the section. This feature is shown in Figure P–2. Answers to the Section Reviews are at the end of the chapter.



6

FUNCTIONS OF COMBINATIONAL LOGIC

CHAPTER OUTLINE

6–1 Basic Adders
6–2 Parallel Binary Adders
6–3 Ripple Carry versus Look-Ahead Carry Adders
6–4 Comparators
6–5 Decoders
6–6 Encoders
6–7 Code Converters
6–8 Multiplexers (Data Selectors)
6–9 Demultiplexers
6–10 Parity Generators/Checkers
6–11 Troubleshooting
■■■ Digital System Application

CHAPTER OBJECTIVES

- Distinguish between half-adders and full-adders
- Use full-adders to implement multibit parallel binary adders
- Explain the differences between ripple carry and look-ahead carry parallel adders
- Use the magnitude comparator to determine the relationship between two binary numbers and use cascaded comparators to handle the comparison of larger numbers
- Implement a basic binary decoder
- Use BCD-to-7-segment decoders in display systems
- Apply a decimal-to-BCD priority encoder in a simple keyboard application
- Convert from binary to Gray code, and Gray code to binary by using logic devices
- Apply multiplexers in data selection, multiplexed displays, logic function generation, and simple communications systems
- Use decoders as demultiplexers
- Explain the meaning of parity
- Use parity generators and checkers to detect bit errors in digital systems
- Implement a simple data communications system
- Identify glitches, common bugs in digital systems

KEY TERMS

- Half-adder
- Full-adder
- Cascading
- Ripple carry
- Look-ahead carry
- Decoder
- Encoder
- Priority encoder
- Multiplexer (MUX)
- Demultiplexer (DEMUX)
- Parity bit
- Glitch

INTRODUCTION

In this chapter, several types of combinational logic circuits are introduced including adders, comparators, decoders, encoders, code converters, multiplexers (data selectors), demultiplexers, and parity generators/checkers. Examples of fixed-function IC devices are included.

FIXED-FUNCTION LOGIC DEVICES

74XX42	74XX47	74XX85
74XX138	74XX139	74XX147
74XX148	74XX151	74XX154
74XX157	74XX280	74XX283

■■■ DIGITAL SYSTEM APPLICATION PREVIEW

The Digital System Application illustrates concepts from this chapter and deals with one portion of a traffic light control system. The system applications in Chapters 6, 7, and 8 focus on various parts of the traffic light control system. Basically, this system controls the traffic light at the intersection of a busy street and a lightly traveled side street. The system includes a combinational logic section to which the topics in this chapter apply, a timing circuit section to which Chapter 7 applies, and a sequential logic section to which Chapter 8 applies.

WWW. VISIT THE COMPANION WEBSITE
Study aids for this chapter are available at
http://www.prenhall.com/floyd

297

▲ FIGURE P–1

Chapter opener.

Worked Examples and Related Problems An abundance of worked examples help to illustrate and clarify basic concepts or specific procedures. Each example ends with a Related Problem that reinforces or expands on the example by requiring the student to work through a problem similar to the example. A typical worked example with a Related Problem is shown in Figure P–3.

Troubleshooting Section Many chapters include a troubleshooting section that relates to the topics covered in the chapter and that emphasizes troubleshooting techniques and the use of test instruments. A portion of a typical troubleshooting section is illustrated in Figure P–4.

Digital System Application Appearing at the end of many chapters, this feature presents a practical application of the concepts covered in the chapter. This feature presents a "real-world" system in which analysis, troubleshooting, and design elements are implemented using procedures covered in the chapter. Some Digital System Applications are limited to a single chapter and others extend over two or more chapters. Specific Digital System Applications are as follows:

- Tablet counting and control system: Chapter 1
- Digital display: Chapters 4 and 11.
- Storage tank control system: Chapter 5

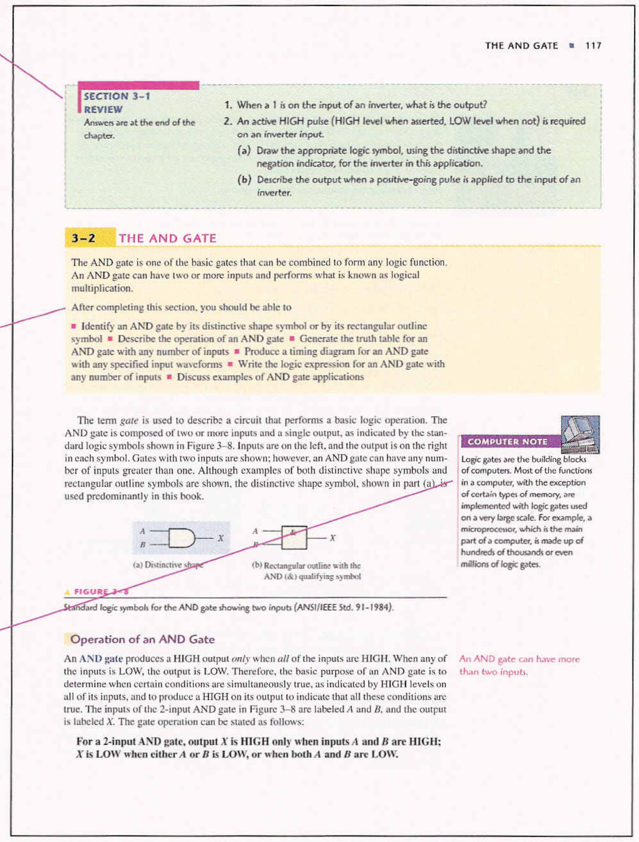
▶ **FIGURE P–2**

Section opener and section review.

Review exercises end each section.

Introductory paragraph and a list of performance-based section objectives begin each section.

Computer Notes are found throughout the text.


THE AND GATE ■ 117

SECTION 3–1 REVIEW
Answers are at the end of the chapter.

1. When a 1 is on the input of an inverter, what is the output?
2. An active HIGH pulse (HIGH level when asserted, LOW level when not) is required on an inverter input.
   (a) Draw the appropriate logic symbol, using the distinctive shape and the negation indicator, for the inverter in this application.
   (b) Describe the output when a positive-going pulse is applied to the input of an inverter.

3–2 THE AND GATE

The AND gate is one of the basic gates that can be combined to form any logic function. An AND gate can have two or more inputs and performs what is known as logical multiplication.

After completing this section, you should be able to

■ Identify an AND gate by its distinctive shape symbol or by its rectangular outline symbol ■ Describe the operation of an AND gate ■ Generate the truth table for an AND gate with any number of inputs ■ Produce a timing diagram for an AND gate with any specified input waveforms ■ Write the logic expression for an AND gate with any number of inputs ■ Discuss examples of AND gate applications

The term *gate* is used to describe a circuit that performs a basic logic operation. The AND gate is composed of two or more inputs and a single output, as indicated by the standard logic symbols shown in Figure 3–8. Inputs are on the left, and the output is on the right in each symbol. Gates with two inputs are shown; however, an AND gate can have any number of inputs greater than one. Although examples of both distinctive shape symbols and rectangular outline symbols are shown, the distinctive shape symbol, shown in part (a), is used predominantly in this book.

COMPUTER NOTE
Logic gates are the building blocks of computers. Most of the functions in a computer, with the exception of certain types of memory, are implemented with logic gates used on a very large scale. For example, a microprocessor, which is the main part of a computer, is made up of hundreds of thousands or even millions of logic gates.

(a) Distinctive shape (b) Rectangular outline with the AND (&) qualifying symbol

▲ FIGURE 3–8
Standard logic symbols for the AND gate showing two inputs (ANSI/IEEE Std. 91-1984).

Operation of an AND Gate

An **AND gate** produces a HIGH output *only* when *all* of the inputs are HIGH. When any of the inputs is LOW, the output is LOW. Therefore, the basic purpose of an AND gate is to determine when certain conditions are simultaneously true, as indicated by HIGH levels on all of its inputs, and to produce a HIGH on its output to indicate that all these conditions are true. The inputs of the 2-input AND gate in Figure 3–8 are labeled A and B, and the output is labeled X. The gate operation can be stated as follows:

An AND gate can have more than two inputs.

For a 2-input AND gate, output X is HIGH only when inputs A and B are HIGH; X is LOW when either A or B is LOW, or when both A and B are LOW.

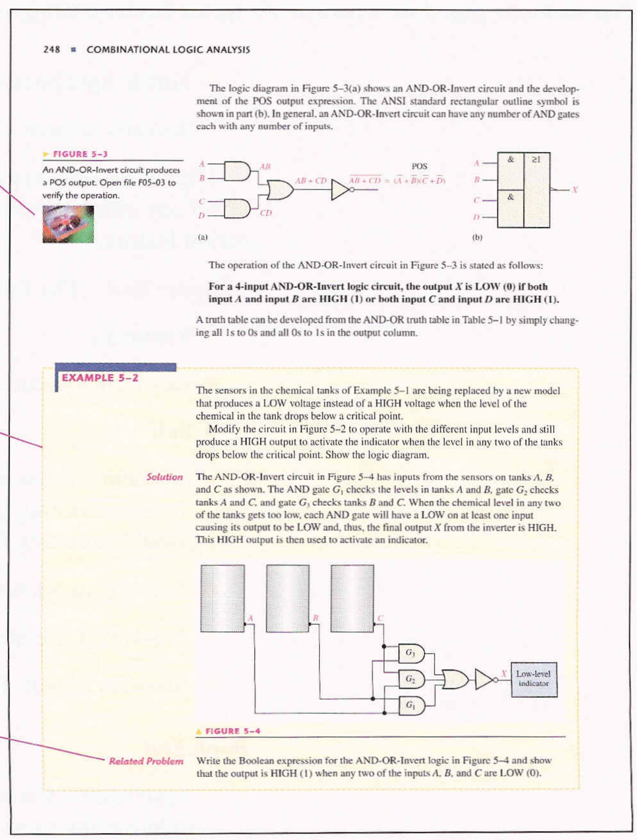
▶ **FIGURE P–3**

An example and related problem.

A special icon indicates selected circuits that are on the CD-ROM packaged with the text.

Examples are set off from text.

Each example contains a problem related to the example.


248 ■ COMBINATIONAL LOGIC ANALYSIS

The logic diagram in Figure 5–3(a) shows an AND-OR-Invert circuit and the development of the POS output expression. The ANSI standard rectangular outline symbol is shown in part (b). In general, an AND-OR-Invert circuit can have any number of AND gates each with any number of inputs.

▶ FIGURE 5–3
An AND-OR-Invert circuit produces a POS output. Open file F05-03 to verify the operation.

The operation of the AND-OR-Invert circuit in Figure 5–3 is stated as follows:

For a 4-input AND-OR-Invert logic circuit, the output X is LOW (0) if both input A and input B are HIGH (1) or both input C and input D are HIGH (1).

A truth table can be developed from the AND-OR truth table in Table 5–1 by simply changing all 1s to 0s and all 0s to 1s in the output column.

EXAMPLE 5–2

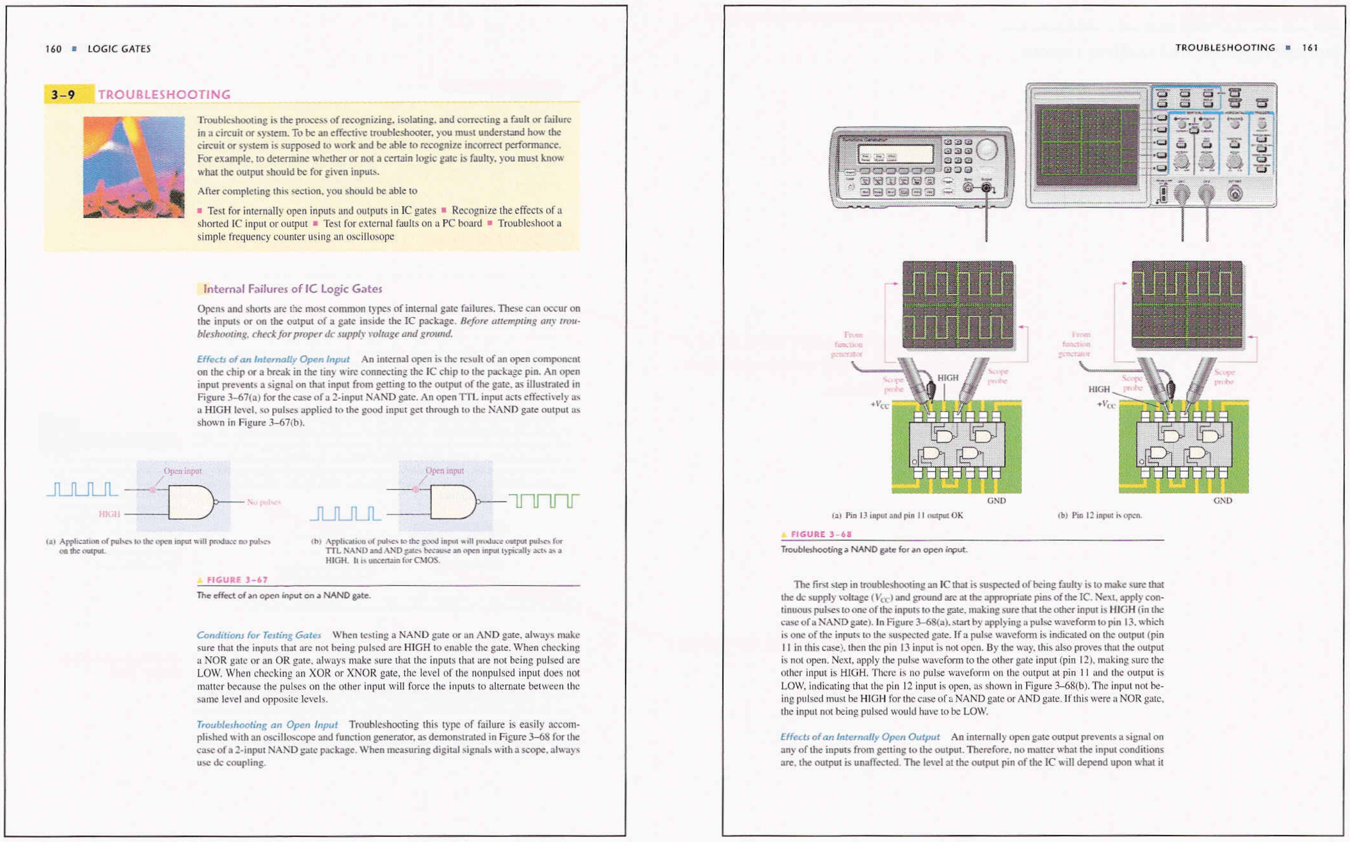
The sensors in the chemical tanks of Example 5–1 are being replaced by a new model that produces a LOW voltage instead of a HIGH voltage when the level of the chemical in the tank drops below a critical point.

Modify the circuit in Figure 5–2 to operate with the different input levels and still produce a HIGH output to activate the indicator when the level in any two of the tanks drops below the critical point. Show the logic diagram.

Solution The AND-OR-Invert circuit in Figure 5–4 has inputs from the sensors on tanks A, B, and C as shown. The AND gate G_1 checks the levels in tanks A and B, gate G_2 checks tanks A and C, and gate G_3 checks tanks B and C. When the chemical level in any two of the tanks gets too low, each AND gate will have a LOW on at least one input causing its output to be LOW and, thus, the final output X from the inverter is HIGH. This HIGH output is then used to activate an indicator.

▲ FIGURE 5–4

Related Problem Write the Boolean expression for the AND-OR-Invert logic in Figure 5–4 and show that the output is HIGH (1) when any two of the inputs A, B, and C are LOW (0).



160 ■ LOGIC GATES

3–9 TROUBLESHOOTING

Troubleshooting is the process of recognizing, isolating, and correcting a fault or failure in a circuit or system. To be an effective troubleshooter, you must understand how the circuit or system is supposed to work and be able to recognize incorrect performance. For example, to determine whether or not a certain logic gate is faulty, you must know what the output should be for given inputs.

After completing this section, you should be able to

■ Test for internally open inputs and outputs in IC gates ■ Recognize the effects of a shorted IC input or output ■ Test for external faults on a PC board ■ Troubleshoot a simple frequency counter using an oscillosope

Internal Failures of IC Logic Gates

Opens and shorts are the most common types of internal gate failures. These can occur on the inputs or on the output of a gate inside the IC package. *Before attempting any troubleshooting, check for proper dc supply voltage and ground.*

Effects of an Internally Open Input An internal open is the result of an open component on the chip or a break in the tiny wire connecting the IC chip to the package pin. An open input prevents a signal on that input from getting to the output of the gate, as illustrated in Figure 3–67(a) for the case of a 2-input NAND gate. An open TTL input acts effectively as a HIGH level, so pulses applied to the good input get through to the NAND gate output as shown in Figure 3–67(b).

(a) Application of pulses to the open input will produce no pulses on the output.

(b) Application of pulses to the good input will produce output pulses for TTL NAND and AND gates because an open input typically acts as a HIGH. It is uncertain for CMOS.

▲ FIGURE 3–67

The effect of an open input on a NAND gate.

Conditions for Testing Gates When testing a NAND gate or an AND gate, always make sure that the inputs that are not being pulsed are HIGH to enable the gate. When checking a NOR gate or an OR gate, always make sure that the inputs that are not being pulsed are LOW. When checking an XOR or XNOR gate, the level of the nonpulsed input does not matter because the pulses on the other input will force the inputs to alternate between the same level and opposite levels.

Troubleshooting an Open Input Troubleshooting this type of failure is easily accomplished with an oscilloscope and function generator, as demonstrated in Figure 3–68 for the case of a 2-input NAND gate package. When measuring digital signals with a scope, always use dc coupling.

TROUBLESHOOTING ■ 161

(a) Pin 13 input and pin 11 output OK

(b) Pin 12 input is open.

▲ FIGURE 3–68

Troubleshooting a NAND gate for an open input.

The first step in troubleshooting an IC that is suspected of being faulty is to make sure that the dc supply voltage (V_{CC}) and ground are at the appropriate pins of the IC. Next, apply continuous pulses to one of the inputs to the gate, making sure that the other input is HIGH (in the case of a NAND gate). In Figure 3–68(a), start by applying a pulse waveform to pin 13, which is one of the inputs to the suspected gate. If a pulse waveform is indicated on the output (pin 11 in this case), then the pin 13 input is not open. By the way, this also proves that the output is not open. Next, apply the pulse waveform to the other gate input (pin 12), making sure the other input is HIGH. There is no pulse waveform on the output at pin 11 and the output is LOW, indicating that the pin 12 input is open, as shown in Figure 3–68(b). The input not being pulsed must be HIGH for the case of a NAND gate or AND gate. If this were a NOR gate, the input not being pulsed would have to be LOW.

Effects of an Internally Open Output An internally open gate output prevents a signal on any of the inputs from getting to the output. Therefore, no matter what the input conditions are, the output is unaffected. The level at the output pin of the IC will depend upon what it

▲ FIGURE P–4

Representative pages from a portion of a typical Troubleshooting section.

- Traffic light control system: Chapters 6, 7, and 8
- Security system: Chapters 9 and 10

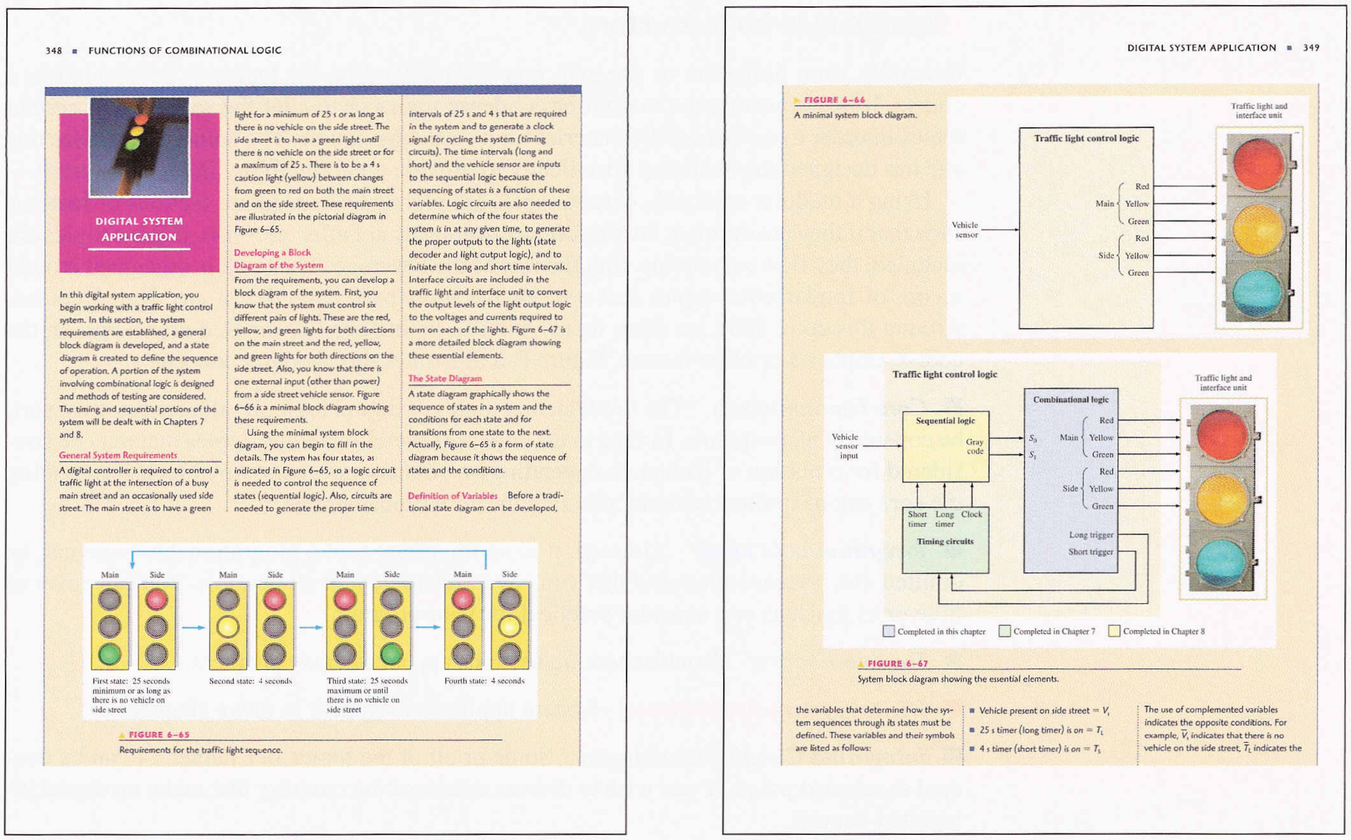
Digital System Applications may be treated as optional because omitting them will not affect any other material in the text. Figure P–5 shows a portion of a Digital System Application feature.

Chapter End The following study aids end each chapter:

- Summary
- Key term glossary
- Self-test
- Problem set that includes some or all of the following categories: Basic, Troubleshooting, Digital System Application, Design, and Multisim Troubleshooting Practice
- Answers to Section Reviews
- Answers to Related Problems for Examples
- Answers to Self-Test

Book End

- Appendices: Code conversion and table of powers of two (Appendix A) and traffic light interface circuits (Appendix B)


348 ■ FUNCTIONS OF COMBINATIONAL LOGIC

DIGITAL SYSTEM APPLICATION

In this digital system application, you begin working with a traffic light control system. In this section, the system requirements are established, a general block diagram is developed, and a state diagram is created to define the sequence of operation. A portion of the system involving combinational logic is designed and methods of testing are considered. The timing and sequential portions of the system will be dealt with in Chapters 7 and 8.

General System Requirements

A digital controller is required to control a traffic light at the intersection of a busy main street and an occasionally used side street. The main street is to have a green light for a minimum of 25 s or as long as there is no vehicle on the side street. The side street is to have a green light until there is no vehicle on the side street or for a maximum of 25 s. There is to be a 4 s caution light (yellow) between changes from green to red on both the main street and on the side street. These requirements are illustrated in the pictorial diagram in Figure 6–65.

Developing a Block Diagram of the System

From the requirements, you can develop a block diagram of the system. First, you know that the system must control six different pairs of lights. These are the red, yellow, and green lights for both directions on the main street and the red, yellow, and green lights for both directions on the side street. Also, you know that there is one external input (other than power) from a side street vehicle sensor. Figure 6–66 is a minimal block diagram showing these requirements.

Using the minimal system block diagram, you can begin to fill in the details. The system has four states, as indicated in Figure 6–65, so a logic circuit is needed to control the sequence of states (sequential logic). Also, circuits are needed to generate the proper time intervals of 25 s and 4 s that are required in the system and to generate a clock signal for cycling the system (timing circuits). The time intervals (long and short) and the vehicle sensor are inputs to the sequential logic because the sequencing of states is a function of these variables. Logic circuits are also needed to determine which of the four states the system is in at any given time, to generate the proper outputs to the lights (state decoder and light output logic), and to initiate the long and short time intervals. Interface circuits are included in the traffic light and interface unit to convert the output levels of the light output logic to the voltages and currents required to turn on each of the lights. Figure 6–67 is a more detailed block diagram showing these essential elements.

The State Diagram

A state diagram graphically shows the sequence of states in a system and the conditions for each state and for transitions from one state to the next. Actually, Figure 6–65 is a form of state diagram because it shows the sequence of states and the conditions.

Definition of Variables Before a traditional state diagram can be developed,

FIGURE 6–65

Requirements for the traffic light sequence.

DIGITAL SYSTEM APPLICATION ■ 349

FIGURE 6–66

A minimal system block diagram.

FIGURE 6–67

System block diagram showing the essential elements.

the variables that determine how the system sequences through its states must be defined. These variables and their symbols are listed as follows:

- Vehicle present on side street = V_s
- 25 s timer (long timer) is *on* = T_L
- 4 s timer (short timer) is *on* = T_S

The use of complemented variables indicates the opposite conditions. For example, $\overline{V}_s$ indicates that there is no vehicle on the side street, $\overline{T}_L$ indicates the

▲ FIGURE P–5

Representative pages from a typical Digital System Application.

- Answers to odd-numbered problems
- Comprehensive glossary
- Index

To the Student

Digital technology is hot! Most everything has already gone digital or will in the near future. For example, cell phones and other types of wireless communication, television, radio, process controls, automotive electronics, consumer electronics, global navigation, military systems, to name only a few applications, depend heavily on digital electronics.

A strong foundation in the fundamentals of digital technology will prepare you for the highly skilled and high-paying jobs of the future. The single most important thing you can do is to understand the core fundamentals. From there you can go anywhere.

In addition, programmable logic is becoming extremely important in today's technology and that topic is introduced in this book. Of course, efficient troubleshooting is a skill that is also widely sought. Troubleshooting and testing methods from traditional testing to manufacturing techniques, such as bed-of-nails, flying probe, and boundary scan, are covered in this book. These are examples of the skills you can acquire with a serious effort to learn the concepts presented.

The CD-ROMs Two CDs are included with this book. One contains Texas Instruments data sheets for digital integrated circuits. The other contains circuit files in Multisim for use with Multisim software Versions 2001 or 7. (These Version 2001 and Version 7 circuit files—as well as those for use with Multisim 8—also appear on the Companion Website at **www.prenhall.com/floyd**.)

User's Guide for Instructors

Generally, time limitation or program emphasis determine the topics to be covered in a course. It is not uncommon to omit or condense topics or to alter the sequence of certain topics in order to customize the material for a particular course. The author recognizes this and has designed this textbook specifically to provide great flexibility in topic coverage.

Using a modular approach, certain topics are organized in separate sections or features such that if they are omitted, the rest of the coverage is not affected. Also, if these topics are included, they flow seamlessly with the rest of the coverage. The book is organized around a core of fundamental topics that are, for the most part, essential in any digital course. Around this core, there are other topics that can be included or omitted depending on the course emphasis or other factors. Figure P-6 illustrates this modular concept.

■ ***Core Fundamentals*** The fundamental topics of digital logic should, for the most part, be covered in all programs. Linked to the core are several "satellite" topics that may be considered for omission or inclusion, depending on your course goals. Any block surrounding the core can be omitted without affecting the core fundamentals.

■ ***Programmable Logic*** Although it is an important topic, programmable logic can be omitted, but it is recommended that you cover this topic if at all possible. You can cover as little or as much as you consider practical for your program.

■ ***Troubleshooting*** Troubleshooting sections appear in many chapters.

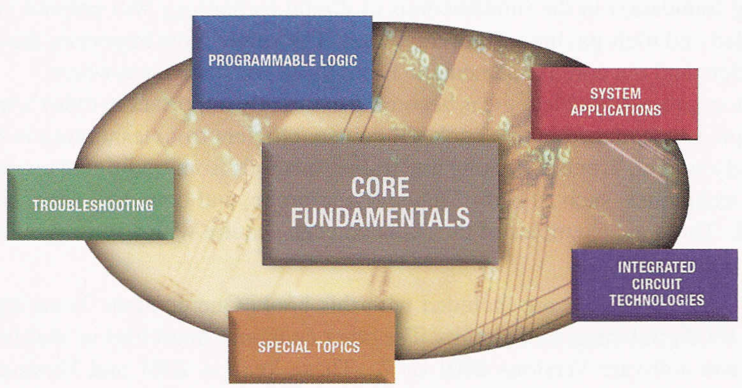
■ ***Digital System Applications*** System applications appear in many chapters.

■ ***Integrated Circuit Technologies*** Some or all of the topics in Chapter 14 can be covered at selected points if you wish to discuss details of the circuitry that make up digital integrated circuits.

■ ***Special Topics*** These topics are *Introduction to Computers* and *Digital Signal Processing* in Chapters 12 and 13, respectively. These are special topics and may not be essential to your digital course.

Also, within each block in Figure P-6 you can choose to omit or de-emphasize some topics because of time constraints or other priorities. For example, in the core fundamentals, error correction codes, carry look-ahead adders, sequential logic design, and other selected topics could be omitted.

Customizing the Table of Contents You can take any one of several paths through *Digital Fundamentals, Ninth Edition,* depending on the goals of your particular program. Whether you choose a minimal coverage of only core fundamentals, a full-blown coverage of all the topics, or anything in between, this book can be adapted to your needs. The



▲ **FIGURE P–6**

Table of Contents following this preface is color coded to match the blocks in Figure P-6. This allows you to identify topics for omission or inclusion for customizing your course.

Several options for use of *Digital Fundamentals, Ninth Edition* are shown below in terms of topics color coded to Figure P-6. Other options are possible, too, including partial coverage of some topics.

Option 1 ■

Option 2 ■■

Option 3 ■■■

Option 4 ■■■■

Option 5 ■■■■■■

Acknowledgments

This innovative text has been realized by the efforts and the skills of many people. I think that we have accomplished what we set out to do, which was to produce a textbook second to none. At Prentice Hall, Kate Linsner and Rex Davidson have contributed a great amount of time, talent, and effort to move this project through its many phases in order to produce the book as you see it. Lois Porter has done a fantastic job of editing the manuscript. She has unraveled the mysteries of this author's markups and often nearly illegible notes and, from that tangled mess, extracted an unbelievably organized and superbly edited manuscript. Also, Jane Lopez has done another beautiful job with the graphics. Another individual who contributed significantly to this book is Gary Snyder, who has provided all of the Multisim circuit files (in Multisim Versions 2001, 7, and 8, all of which appear on the Companion Website at **www.prenhall.com/floyd**). I extend my thanks and appreciation to all of these people and others who were indirectly involved in the project.

In the revision of this and all textbooks, I depend on expert input from many users as well as non-users. I want to offer my sincere thanks to the following reviewers, who submitted many valuable suggestions and provided lots of constructive criticism: Bo Barry, University of North Carolina–Charlotte; Chuck McGlumphy, Belmont Technical College; and Amy Ray, Mitchell Community College.

My appreciation goes to David Buchla for his efforts to make sure that the lab manual is closely coordinated with the text and for his valuable input. I would also like to mention Muhammed Arif Shabir for his suggestion concerning shift registers.

I thank all of the members of the Prentice Hall sales force whose efforts have helped make my books available to a large number of users throughout the world. In addition, I am grateful to all of you who have adopted this text for your classes or for your own use. Without you we would not be in business. I hope that you find this book to be a valuable learning tool and reference for students.

Tom Floyd