Contents

Preface to the Seventh Edition (SIE)

The seventh edition of *Digital Principles and Applications* continues with the upgradation of the work started in its previous edition. The job was to build upon the strengths of one of the best introductory and authentic texts in the field of Digital Electronics—its lucid language, down-to-earth approach, detailed analysis and ready-to-use information for laboratory practices. The sixth edition sought improvement primarily by (i) strengthening the design or synthesis aspect that included advanced material, such as a simple computer design, and (ii) incorporating many new topics like Hardware Description Language, Asynchronous Sequential Circuit, Algorithm State Machine chart, Quine-McClusky algorithm, Look Ahead Carry Adder, etc.

The tremendous response to the improvements made in the sixth edition from the academic community prompted us to work on their suggestions and come out with this seventh edition.

NEW TO THIS EDITION

The seventh edition has been revised extensively and restructured to emphasize new and important concepts in Digital Principles and Applications. This edition increases the depth and breadth of the title by incorporating latest information on existing topics like *Boolean Algebra, Schmitt Trigger, 555 Timer, Edge Triggering, Memory Cell, Computer Architecture,* and also introduces new topics like *Noise Margin, Error Detection and Correction, Universal Shift Register and Content Addressable Memory.*

The most notable change in this edition is the inclusion of two completely new features—*problem solving by multiple methods* and *laboratory experiments*—that will enable the student community develop deeper understanding of the application side of digital principles. *Problem solving by multiple methods* help students in understanding and appreciating different alternatives to reach a solution, without feeling stuck at any point of time. *Laboratory experiments* facilitate experimentation with different analysis and synthesis problems using digital integrated circuits (IC). Each experiment describes its aim, a short reference to theory, apparatus required and different work elements.

THE BASIC FEATURES

The new edition retains its appeal as a complete self-study guide for a first-level course on Digital Logic and Digital Circuits. It will serve the purpose of a textbook for undergraduate students of CSE, ECE, EEE, Electronics and Instrumentation and IT. It will also be a valuable reference for students of MCA, BCA, DOEACC 'A' Level, as well as BSc/MSc (Computer Science/IT).

The key features are:

- ➢ Presence of various applications and lab experiments considering the common digital circuit design employed in industries (e.g., LCD display and ADC0804 operation).
- ➢ In-depth coverage of important topics like clock and timing circuits, D/A-A/D conversion, register, counters and memory.
- ➢ Tutorial-based approach with section-end self test questions and problem solving through various methods.
- ➢ Useful discussion on TTL and CMOS devices and pin diagrams
- ➢ Rich Pedagogy
  - *180 Solved Examples*
  - *290 Section-end Problems*
  - *500 Chapter-end Problems*

COMPREHENSIVE WEBSITE

An important addition to this title is the accompanying website—*http://www.mhhe.com/leach/dpa7*, designed to be an exhaustive Online Learning Centre (OLC). This website contains the following:

For Students

- Downloadable codes for HDL examples in the book
- Supplementary Reading material

Besides Quine-McClusky code and HDL examples, additional information and discussion on various supplementary materials like five-variable Karnaugh Map and Petrick's Algorithm will be available here. Regular updates on different topics of Digital Electronics will be posted to keep the reader informed about recent changes in this field.

For Instructors

Instructors who have adopted this textbook can access a password-protected section that offers the following resources.

- Solution manual
- Chapterwise PowerPoint slides with diagrams and notes

ACKNOWLEDGEMENTS

I would like to acknowledge the inspiration and support I received from Prof. B N Chatterji (Retd.), Prof. R N Pal (Retd.), Prof. R V Rajakumar, Prof. A Chakraborty, Prof. P P Chakraborty, Prof. D Datta, Prof. S Bannerjee, Prof. P K Biswas, Prof. S S Pathak, Prof. S Mukhopadhyaya, Prof. A S Dhar, Prof. I Chakraborty, Prof. P Mandal, Prof. S Mahapatra, Prof. T K Bhattacharya, and all my faculty colleagues at Indian Institute of Technology (IIT) Kharagapur, especially, from the Department of Electronics and Electrical Communication Engineering and G S Sanyal School of Telecommunications. But for the paucity of space, I would have named many others.

Thanks are due to my research students—Mr S Ari, Mr Md Sahidullah, Mr Israj Ali, and Mr A Mandal for their contribution at different stages of development of the edition. I acknowledge the benefit derived from my interaction with different batches of students while teaching the Digital Electronics subject—three years at Institute of Engineering and Management, Kolkata and over six years at IIT Kharagpur.

I am grateful to the esteemed reviewers for their encouraging comments and valuable suggestions for this edition.

Sunil Mathur
Maharaja Agrasen Institute of Technology, New Delhi
V Kumar
Maharaja Surajmal Institute of Technology, New Delhi
Bijoy Bandopadhyay
University College of Science & Technology, Kolkata
Anita Kanavalli
M S Ramaiah Institute of Technology, Bangalore

I also thank the entire team of Tata McGraw Hill Education, more specifically Vibha Mahajan, Shalini Jha, Ashes Saha, Surbhi Suman, Anjali Razdan and Baldev Raj for their support.

At this point, I humbly remember all my teachers and my father (late) G N Saha who provided me a great learning environment. I also fondly recollect the contributions in my upbringing of Kharagpur Vivekananda Yuva Mahamandal, Vivekananda Study Circle, IIT Kharagpur Campus and Ramakrishna Mission. I must mention the support I always received from my family—my mother, my parents-in-law, my sisters (specially Chhordi), Chhoto Jamaibabu, and last but not the least, my wife, Sanghita, and daughter, Upasana. The effort behind this work was mine but the time was all theirs.

GOUTAM SAHA

Feedback

Due care has been taken to avoid any mistake in the print edition as well as in the OLC. However, any note on oversight as well as suggestions for further improvement sent at tmh.csefeedback@gmail.com will be gratefully acknowledged (*kindly mention the title and author name in the subject line*). Also, please report to us any piracy of the book spotted by you.

Preface

PURPOSE

The fifth edition of *Digital Principles and Applications* is completely recorgnized. It is written for the individual who wishes to learn the *principles* of digital circuits and then *apply* them to useful, meaningful design. Thus the title. The material in this book is appropriate for an introductory course in digital logic in either a computer or an electronics program. It is also appropriate for "self-study" and as a "reference" for individuals working in the field. Emphasis is given to the two most popular digital circuit (IC) families—transistor-transistor logic (TTL) and complementary metal oxide silicon (CMOS) logic. Many of these individual ICs are discussed in detail, and pinouts for more than 60 digital IC chips are summarized in Appendix 8. Standard logic symbols are used along with the new IEEE standard logic. A review of the new IEEE symbols is given in the appendix.

BACKGROUND

It is not necessary to have a background in electronics to study this text. A familiarity with Ohm's law and voltage and current in simple dc resistive circuits is helpful but not required. If you have no desire to learn about electronics, you can skip Chap. 13. To the extent possible, the remaining chapters are written to be independent of this material. If you have not studied electronics, Chap. 13 will provided the necessary background for you to converse successfully with those who have. Study it any time after Chap. 1. For "old-times" who have studied electronics, Chap. 13 will provide a good review and perhaps a new and valuable point of view. In any case, the material in Chap. 13 will certainly enhance both the knowledge and ability of anyone!

ORGANIZATION

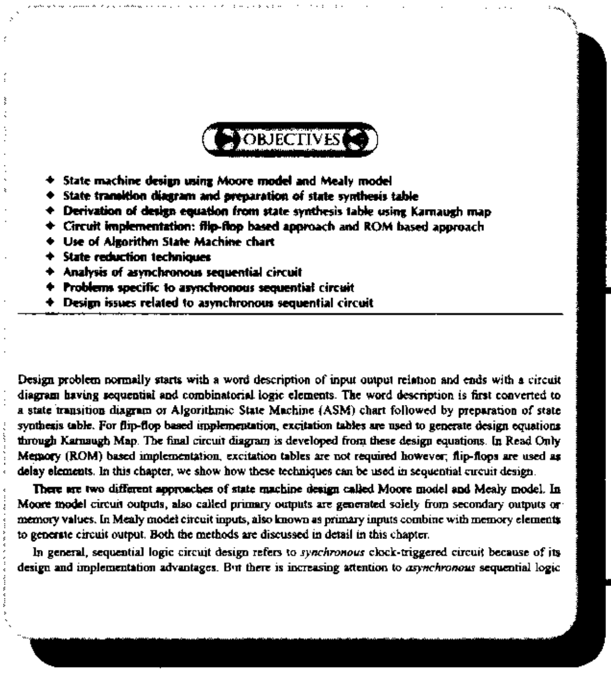
Each chapter begins with a contents that lists the subjects in each section. The contents listing is followed by a list of chapter objectives. At the end of each chapter section are review questions, called self-tests, which are intended to be a self-check of key ideas and concepts. At the end of each chapter, answers are supplied for the self-tests. A summary and a glossary are provided at the end of each chapter. In any subject area, there are many terms and concepts to be learned. The summary and glossary will provide you with the opportunity to be sure that you understand the *exact* meaning of these terms, phrases, and abbreviations, The end-of-chapter problems are arranged according to chapter sections. The problems reinforce ideas and concepts presented and allow you to apply them on your own. Solutions to selected odd-numbered problems are given at the end of the book. In addition, the appendix contains reference material that will be useful from time to time.

LABORATORY EXPERIMENTS

A complete set of experiments keyed to this text is available in a laboratory manual, *Experiments for Digital Principles*.

Donald P. Leach
Albert Paul Malvino

Visual Walkthrough



OBJECTIVES

- State machine design using Moore model and Mealy model
- State transition diagram and preparation of state synthesis table
- Derivation of design equation from state synthesis table using Karnaugh map
- Circuit implementation: flip-flop based approach and ROM based approach
- Use of Algorithm State Machine chart
- State reduction techniques
- Analysis of asynchronous sequential circuit
- Problems specific to asynchronous sequential circuit
- Design issues related to asynchronous sequential circuit

Design problem normally starts with a word description of input output relation and ends with a circuit diagram having sequential and combinatorial logic elements. The word description is first converted to a state transition diagram or Algorithmic State Machine (ASM) chart followed by preparation of state synthesis table. For flip-flop based implementation, excitation tables are used to generate design equations through Karnaugh Map. The final circuit diagram is developed from these design equations. In Read Only Memory (ROM) based implementation, excitation tables are not required however; flip-flops are used as delay elements. In this chapter, we show how these techniques can be used in sequential circuit design.

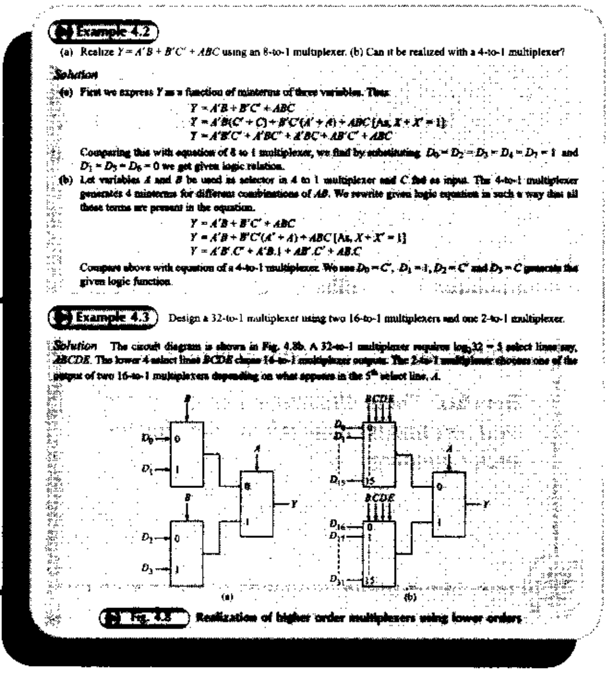
There are two different approaches of state machine design called Moore model and Mealy model. In Moore model circuit outputs, also called primary outputs are generated solely from secondary outputs or memory values. In Mealy model circuit inputs, also known as primary inputs combine with memory elements to generate circuit output. Both the methods are discussed in detail in this chapter.

In general, sequential logic circuit design refers to *synchronous* clock-triggered circuit because of its design and implementation advantages. B'it there is increasing attention to *asynchronous* sequential logic

Chapter Objectives

Every chapter opens with a set of chapter objectives.

Benefits: *These provide a quick look into the concepts that will be discussed in the chapter.*



Example 4.2

(a) Realize $Y = A'B + B'C' + ABC$ using an 8-to-1 multiplexer. (b) Can it be realized with a 4-to-1 multiplexer?

Solution

(a) First we express Y as a function of minterms of three variables. Thus

$$Y = A'B + B'C' + ABC$$
$$Y = A'B(C' + C) + B'C'(A' + A) + ABC\ [\text{As, } X + X' = 1]$$
$$Y = A'B'C' + A'BC' + A'BC + AB'C' + ABC$$

Comparing this with equation of 8 to 1 multiplexer, we find by substituting $D_0 = D_2 = D_3 = D_4 = D_7 = 1$ and $D_1 = D_5 = D_6 = 0$ we get given logic relation.

(b) Let variables A and B be used as selector in 4 to 1 multiplexer and C fed as input. The 4-to-1 multiplexer generates 4 minterms for different combinations of AB. We rewrite given logic equation in such a way that all these terms are present in the equation.

$$Y = A'B + B'C' + ABC$$
$$Y = A'B + B'C'(A' + A) + ABC\ [\text{As, } X + X' = 1]$$
$$Y = A'B'.C' + A'B.1 + AB'.C' + AB.C$$

Compare above with equation of a 4-to-1 multiplexer. We see $D_0 = C'$, $D_1 = 1$, $D_2 = C'$ and $D_3 = C$ generate the given logic function.

Example 4.3 Design a 32-to-1 multiplexer using two 16-to-1 multiplexers and one 2-to-1 multiplexer.

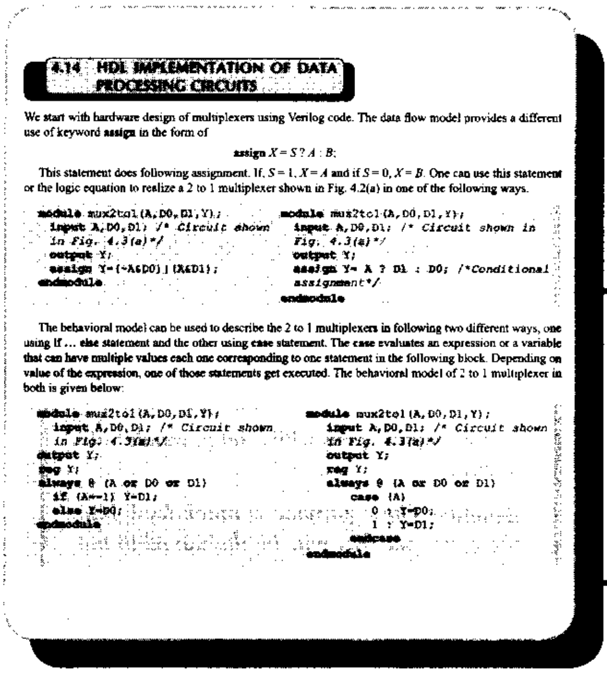
Solution The circuit diagram is shown in Fig. 4.8b. A 32-to-1 multiplexer requires $\log_2 32 = 5$ select lines say, $ABCDE$. The lower 4 select lines $BCDE$ chose 16-to-1 multiplexer outputs. The 2-to-1 multiplexer chooses one of the output of two 16-to-1 multiplexers depending on what appears in the 5th select line, A.

Fig. 4.8 Realization of higher order multiplexers using lower orders

Examples

Every chapter contains several worked out examples totalling to 180 in the book.

Benefits: *These will guide the students while understanding the concepts and working out the exercise problems.*



4.14 HDL IMPLEMENTATION OF DATA PROCESSING CIRCUITS

We start with hardware design of multiplexers using Verilog code. The data flow model provides a different use of keyword **assign** in the form of

assign $X = S\,?\,A : B$;

This statement does following assignment. If, $S = 1$, $X = A$ and if $S = 0$, $X = B$. One can use this statement or the logic equation to realize a 2 to 1 multiplexer shown in Fig. 4.2(a) in one of the following ways.

```
module mux2to1(A,D0,D1,Y);              module mux2to1(A,D0,D1,Y);
  input A,D0,D1; /* Circuit shown         input A,D0,D1; /* Circuit shown in
  in Fig. 4.3(a)*/                        Fig. 4.3(a)*/
  output Y;                               output Y;
  assign Y=(~A&D0)|(A&D1);                assign Y= A ? D1 : D0; /*Conditional
endmodule                                 assignment*/
                                        endmodule
```

The behavioral model can be used to describe the 2 to 1 multiplexers in following two different ways, one using If … else statement and the other using case statement. The case evaluates an expression or a variable that can have multiple values each one corresponding to one statement in the following block. Depending on value of the expression, one of those statements get executed. The behavioral model of 2 to 1 multiplexer in both is given below:

```
module mux2to1(A,D0,D1,Y);              module mux2to1(A,D0,D1,Y);
  input A,D0,D1; /* Circuit shown         input A,D0,D1; /* Circuit shown
  in Fig. 4.3(a)*/                        in Fig. 4.3(a)*/
output Y;                                 output Y;
reg Y;                                    reg Y;
always @ (A or D0 or D1)                  always @ (A or D0 or D1)
  if (A==1) Y=D1;                             case (A)
  else Y=D0;                                    0 : Y=D0;
endmodule                                       1 : Y=D1;
                                              endcase
                                        endmodule
```

HDL Codes

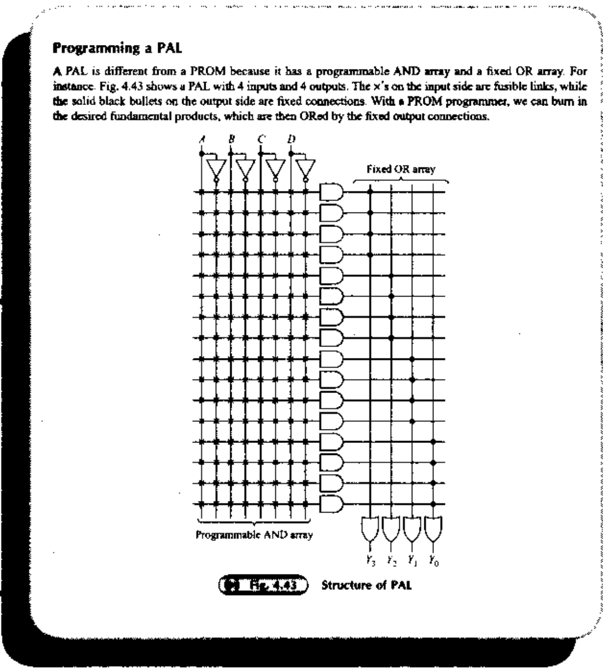
New to this edition, HDL, an interesting development in the field of hardware design, has been introduced.

Benefits: *The relevant HDL description and codes are weaved into chapters to help students implement and design digital circuits.*

Figures

Figures are used exhaustively in the text.

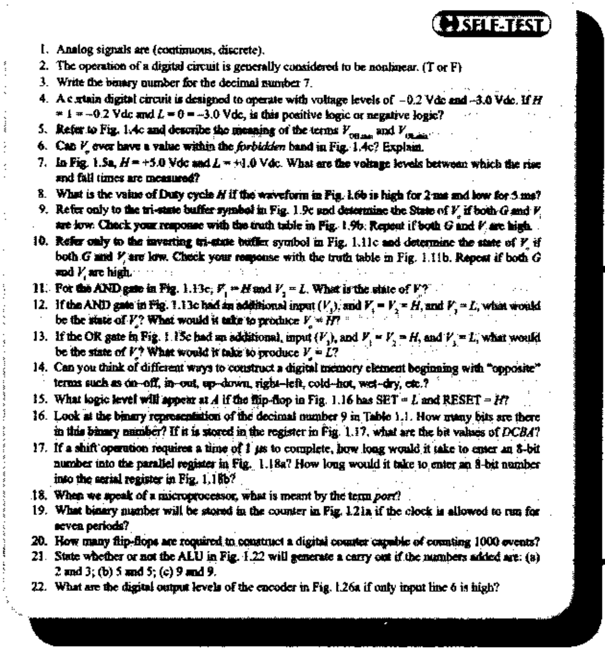
Benefits: *These illustrate the concepts and methods described for better understanding.*



Programming a PAL

A PAL is different from a PROM because it has a programmable AND array and a fixed OR array. For instance. Fig. 4.43 shows a PAL with 4 inputs and 4 outputs. The ×'s on the input side are fusible links, while the solid black bullets on the output side are fixed connections. With a PROM programmer, we can burn in the desired fundamental products, which are then ORed by the fixed output connections.

Fig. 4.43 Structure of PAL



SELF-TEST

1. Analog signals are (continuous, discrete).
2. The operation of a digital circuit is generally considered to be nonlinear. (T or F)
3. Write the binary number for the decimal number 7.
4. A certain digital circuit is designed to operate with voltage levels of −0.2 Vdc and −3.0 Vdc. If $H = 1 = -0.2$ Vdc and $L = 0 = -3.0$ Vdc, is this positive logic or negative logic?
5. Refer to Fig. 1.4c and describe the meaning of the terms $V_{OH,min}$ and $V_{OL,min}$.
6. Can V_o ever have a value within the *forbidden* band in Fig. 1.4c? Explain.
7. In Fig. 1.5a, $H = +5.0$ Vdc and $L = +1.0$ Vdc. What are the voltage levels between which the rise and fall times are measured?
8. What is the value of Duty cycle *D* if the waveform in Fig. 1.6b is high for 2 ms and low for 5 ms?
9. Refer only to the tri-state buffer symbol in Fig. 1.9c and determine the State of V_o if both G and V_i are low. Check your response with the truth table in Fig. 1.9b. Repeat if both G and V_i are high.
10. Refer only to the inverting tri-state buffer symbol in Fig. 1.11c and determine the state of V_o if both G and V_i are low. Check your response with the truth table in Fig. 1.11b. Repeat if both G and V_i are high.
11. For the AND gate in Fig. 1.13c, $V_1 = H$ and $V_2 = L$. What is the state of V_o?
12. If the AND gate in Fig. 1.13c had an additional input (V_3), and $V_1 = V_2 = H$, and $V_3 = L$, what would be the state of V_o? What would it take to produce $V_o = H$?
13. If the OR gate in Fig. 1.15c had an additional, input (V_3), and $V_1 = V_2 = H$, and $V_3 = L$, what would be the state of V_o? What would it take to produce $V_o = L$?
14. Can you think of different ways to construct a digital memory element beginning with "opposite" terms such as on–off, in–out, up–down, right–left, cold–hot, wet–dry, etc.?
15. What logic level will appear at *A* if the flip-flop in Fig. 1.16 has SET = *L* and RESET = *H*?
16. Look at the binary representation of the decimal number 9 in Table 1.1. How many bits are there in this binary number? If it is stored in the register in Fig. 1.17, what are the bit values of *DCBA*?
17. If a shift operation requires a time of 1 μs to complete, how long would it take to enter an 8-bit number into the parallel register in Fig. 1.18a? How long would it take to enter an 8-bit number into the serial register in Fig. 1.18b?
18. When we speak of a microprocessor, what is meant by the term *port*?
19. What binary number will be stored in the counter in Fig. 1.21a if the clock is allowed to run for seven periods?
20. How many flip-flops are required to construct a digital counter capable of counting 1000 events?
21. State whether or not the ALU in Fig. 1.22 will generate a carry out if the numbers added are: (a) 2 and 3; (b) 5 and 5; (c) 9 and 9.
22. What are the digital output levels of the encoder in Fig. 1.26a if only input line 6 is high?

Self-Test

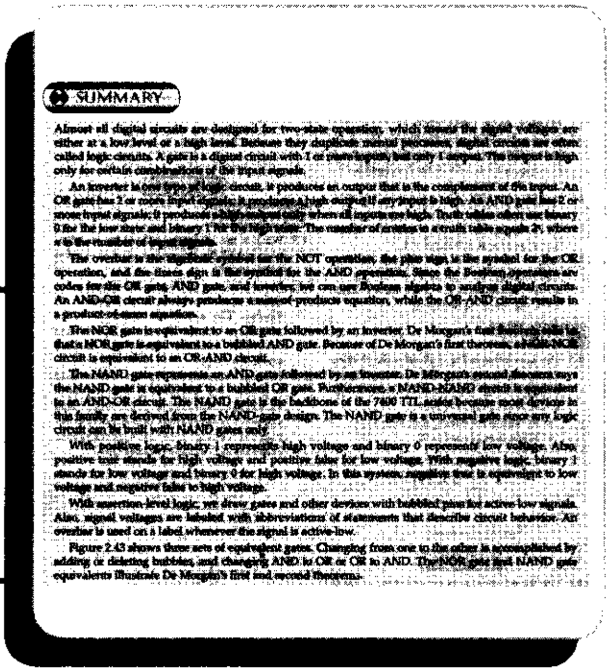
A section called Self-Test appears after every section in every chapter.

Benefits: *This will help students check their understanding of the concepts discussed in a section before moving on to the next section. Answers to Self-Tests are given at the end of that chapter.*

Summary

A Brief summary is provided at the end of the chapters.

Benefits: *Summary gives the essence of each chapter in brief and will be helpful for a quick review during the examinations.*



SUMMARY

Almost all digital circuits are designed for two-state operation, which means the signal voltages are either at a low level or a high level. Because they duplicate mental processes, digital circuits are often called logic circuits. A gate is a digital circuit with 1 or more inputs, but only 1 output. The output is high only for certain combinations of the input signals.

An inverter is one type of logic circuit; it produces an output that is the complement of the input. An OR gate has 2 or more input signals; it produces a high output if any input is high. An AND gate has 2 or more input signals; it produces a high output only when all inputs are high. Truth tables often use binary 0 for the low state and binary 1 for the high state. The number of entries in a truth table equals 2^n, where n is the number of input signals.

The overbar is the algebraic symbol for the NOT operation, the plus sign is the symbol for the OR operation, and the times sign is the symbol for the AND operation. Since the Boolean operations are codes for the OR gate, AND gate, and inverter, we can use Boolean algebra to analyze digital circuits. An AND-OR circuit always produces a sum-of-products equation, while the OR-AND circuit results in a product-of-sums equation.

The NOR gate is equivalent to an OR gate followed by an inverter. De Morgan's first theorem tells us that a NOR gate is equivalent to a bubbled AND gate. Because of De Morgan's first theorem, a NOR-NOR circuit is equivalent to an OR-AND circuit.

The NAND gate represents an AND gate followed by an inverter. De Morgan's second theorem says the NAND gate is equivalent to a bubbled OR gate. Furthermore, a NAND-NAND circuit is equivalent to an AND-OR circuit. The NAND gate is the backbone of the 7400 TTL series because most devices in this family are derived from the NAND-gate design. The NAND gate is a universal gate since any logic circuit can be built with NAND gates only.

With positive logic, binary 1 represents high voltage and binary 0 represents low voltage. Also, positive true stands for high voltage and positive false for low voltage. With negative logic, binary 1 stands for low voltage and binary 0 for high voltage. In this system, negative true is equivalent to low voltage and negative false to high voltage.

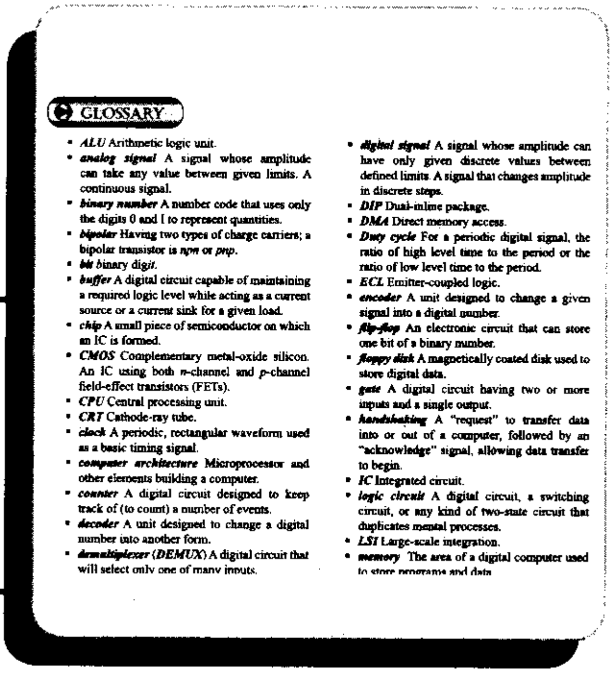
With assertion-level logic, we draw gates and other devices with bubbled pins for active-low signals. Also, signal voltages are labeled with abbreviations of statements that describe circuit behavior. An overbar is used on a label whenever the signal is active-low.

Figure 2.43 shows three sets of equivalent gates. Changing from one to the other is accomplished by adding or deleting bubbles, and changing AND to OR or OR to AND. The NOR gate and NAND gate equivalents illustrate De Morgan's first and second theorems.

Glossary

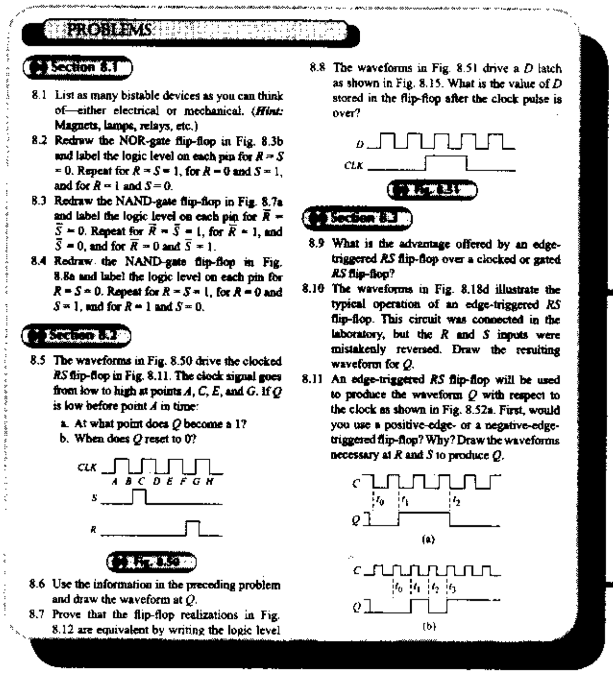
A glossary containing the important definitions and abbreviations is listed at the end of each chapter.

Benefits: *It helps in memorising the important terms discussed in the chapter.*



GLOSSARY

- ***ALU*** Arithmetic logic unit.
- ***analog signal*** A signal whose amplitude can take any value between given limits. A continuous signal.
- ***binary number*** A number code that uses only the digits 0 and I to represent quantities.
- ***bipolar*** Having two types of charge carriers; a bipolar transistor is *npn* or *pnp*.
- ***bit*** binary dig*it*.
- ***buffer*** A digital circuit capable of maintaining a required logic level while acting as a current source or a current sink for a given load.
- ***chip*** A small piece of semiconductor on which an IC is formed.
- ***CMOS*** Complementary metal-oxide silicon. An IC using both *n*-channel and *p*-channel field-effect transistors (FETs).
- ***CPU*** Central processing unit.
- ***CRT*** Cathode-ray tube.
- ***clock*** A periodic, rectangular waveform used as a basic timing signal.
- ***computer architecture*** Microprocessor and other elements building a computer.
- ***counter*** A digital circuit designed to keep track of (to count) a number of events.
- ***decoder*** A unit designed to change a digital number into another form.
- ***demultiplexer*** (***DEMUX***) A digital circuit that will select only one of many inputs.
- ***digital signal*** A signal whose amplitude can have only given discrete values between defined limits. A signal that changes amplitude in discrete steps.
- ***DIP*** Dual-inline package.
- ***DMA*** Direct memory access.
- ***Duty cycle*** For a periodic digital signal, the ratio of high level time to the period or the ratio of low level time to the period.
- ***ECL*** Emitter-coupled logic.
- ***encoder*** A unit designed to change a given signal into a digital number.
- ***flip-flop*** An electronic circuit that can store one bit of a binary number.
- ***floppy disk*** A magnetically coated disk used to store digital data.
- ***gate*** A digital circuit having two or more inputs and a single output.
- ***handshaking*** A "request" to transfer data into or out of a computer, followed by an "acknowledge" signal, allowing data transfer to begin.
- ***IC*** Integrated circuit.
- ***logic circuit*** A digital circuit, a switching circuit, or any kind of two-state circuit that duplicates mental processes.
- ***LSI*** Large-scale integration.
- ***memory*** The area of a digital computer used to store programs and data.



PROBLEMS

Section 8.1

8.1 List as many bistable devices as you can think of—either electrical or mechanical. (***Hint:*** Magnets, lamps, relays, etc.)

8.2 Redraw the NOR-gate flip-flop in Fig. 8.3b and label the logic level on each pin for $R = S = 0$. Repeat for $R = S = 1$, for $R = 0$ and $S = 1$, and for $R = 1$ and $S = 0$.

8.3 Redraw the NAND-gate flip-flop in Fig. 8.7a and label the logic level on each pin for $\overline{R} = \overline{S} = 0$. Repeat for $\overline{R} = \overline{S} = 1$, for $\overline{R} = 1$, and $\overline{S} = 0$, and for $\overline{R} = 0$ and $\overline{S} = 1$.

8.4 Redraw the NAND-gate flip-flop in Fig. 8.8a and label the logic level on each pin for $R = S = 0$. Repeat for $R = S = 1$, for $R = 0$ and $S = 1$, and for $R = 1$ and $S = 0$.

Section 8.2

8.5 The waveforms in Fig. 8.50 drive the clocked *RS* flip-flop in Fig. 8.11. The clock signal goes from low to high at points *A*, *C*, *E*, and *G*. If *Q* is low before point *A* in time:

a. At what point does *Q* become a 1?
b. When does *Q* reset to 0?

Fig. 8.50

8.6 Use the information in the preceding problem and draw the waveform at *Q*.

8.7 Prove that the flip-flop realizations in Fig. 8.12 are equivalent by writing the logic level

8.8 The waveforms in Fig. 8.51 drive a *D* latch as shown in Fig. 8.15. What is the value of *D* stored in the flip-flop after the clock pulse is over?

Fig. 8.51

Section 8.3

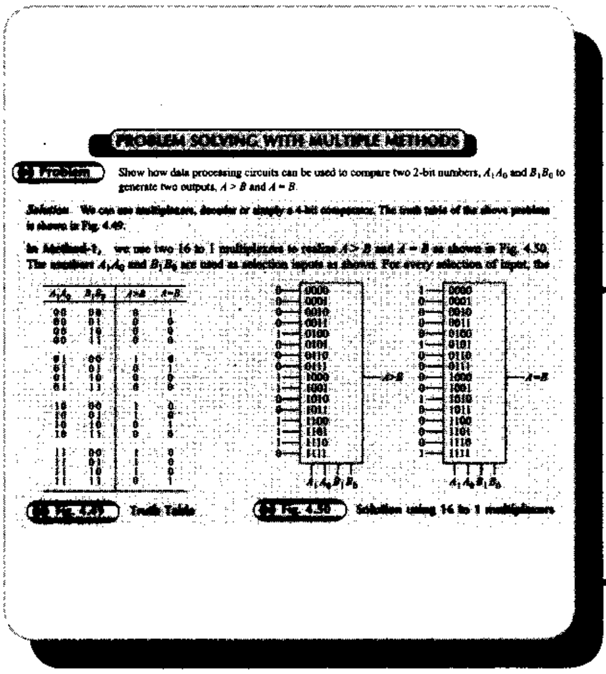
8.9 What is the advantage offered by an edge-triggered *RS* flip-flop over a clocked or gated *RS* flip-flop?

8.10 The waveforms in Fig. 8.18d illustrate the typical operation of an edge-triggered *RS* flip-flop. This circuit was connected in the laboratory, but the *R* and *S* inputs were mistakenly reversed. Draw the resulting waveform for *Q*.

8.11 An edge-triggered *RS* flip-flop will be used to produce the waveform *Q* with respect to the clock as shown in Fig. 8.52a. First, would you use a positive-edge- or a negative-edge-triggered flip-flop? Why? Draw the waveforms necessary at *R* and *S* to produce *Q*.

The text contains more than 250 section-end practice problems.

Benefits: *These will help the students in improving their problem-solving skills.*



PROBLEM SOLVING WITH MULTIPLE METHODS

Problem Show how data processing circuits can be used to compare two 2-bit numbers, A_1A_0 and B_1B_0 to generate two outputs, $A > B$ and $A = B$.

Solution We can use multiplexers, decoder or simply a 4-bit comparator. The truth table of the above problem is shown in Fig. 4.49.

In Method-1, we use two 16 to 1 multiplexers to realize $A > B$ and $A = B$ as shown in Fig. 4.50. The numbers A_1A_0 and B_1B_0 are used as selection inputs as shown. For every selection of input, the

A_1A_0	B_1B_0	$A>B$	$A=B$
00	00	0	1
00	01	0	0
00	10	0	0
00	11	0	0
01	00	1	0
01	01	0	1
01	10	0	0
01	11	0	0
10	00	1	0
10	01	1	0
10	10	0	1
10	11	0	0
11	00	1	0
11	01	1	0
11	10	1	0
11	11	0	1

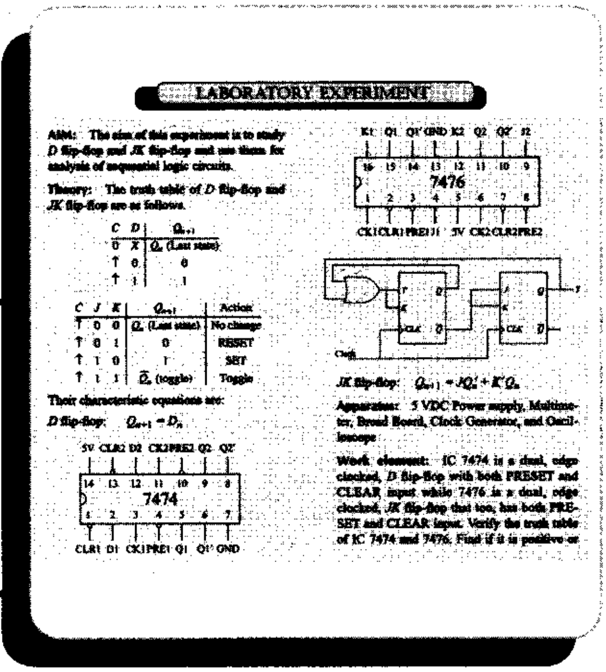
Fig. 4.49 Truth Table

Fig. 4.50 Solution using 16 to 1 multiplexers

Multiple Methods

Each chapter contains numerous problems solved using multiple methods.

Benefits: *Problem solving by multiple methods helps students in understanding and appreciating different alternatives to reach a solution, without feeling stuck at any point of time.*



LABORATORY EXPERIMENT

AIM: The aim of this experiment is to study *D* flip-flop and *JK* flip-flop and use them for analysis of sequential logic circuits.

Theory: The truth table of *D* flip-flop and *JK* flip-flop are as follows.

C	D	Q_{n+1}
0	X	Q_n (Last state)
↑	0	0
↑	1	1

C	J	K	Q_{n+1}	Action
↑	0	0	Q_n (Last state)	No change
↑	0	1	0	RESET
↑	1	0	1	SET
↑	1	1	$\bar{Q}_n$ (toggle)	Toggle

Their characteristic equations are:

D flip-flop: $Q_{n+1} = D_n$

5V CLR2 D2 CK2 PRE2 Q2 Q2'
14 13 12 11 10 9 8
7474
1 2 3 4 5 6 7
CLR1 D1 CK1 PRE1 Q1 Q1' GND
K1 Q1 Q1' GND K2 Q2 Q2' J2
16 15 14 13 12 11 10 9
7476
1 2 3 4 5 6 7 8
CK1 CLR1 PRE1 J1 5V CK2 CLR2 PRE2
J
K
CLK
Q
Clock
Y

JK flip-flop: $Q_{n+1} = JQ_n' + K'Q_n$

Apparatus: 5 VDC Power supply, Multimeter, Bread Board, Clock Generator, and Oscilloscope

Work element: IC 7474 is a dual, edge clocked, *D* flip-flop with both PRESET and CLEAR input while 7476 is a dual, edge clocked, *JK* flip-flop that too, has both PRESET and CLEAR input. Verify the truth table of IC 7474 and 7476. Find if it is positive or

Laboratory Experiments

Each chapter contains a lab experiment.

Benefits: *Laboratory experiments facilitate experimentation with different analysis and synthesis problems using digital integrated circuits (IC). These give a hands-on experience to the reader.*