

# EL SOLUCIONARIO

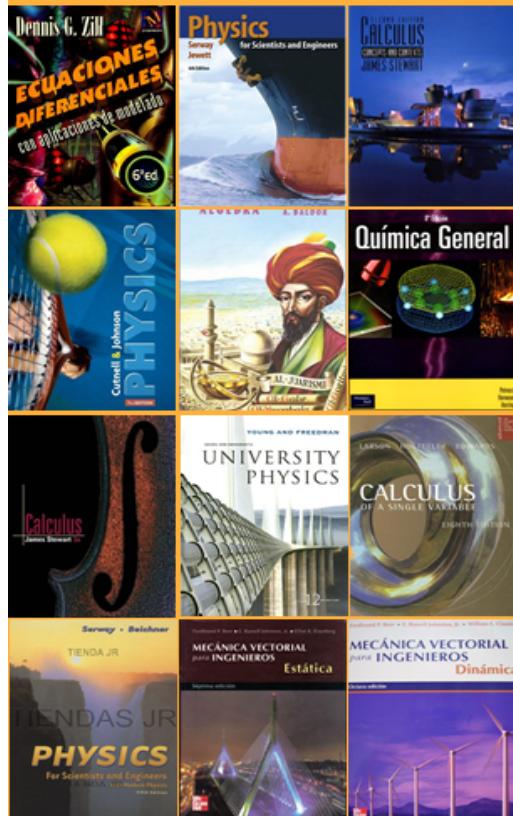
# EL SOLUCIONARIO



<http://www.elsolucionario.blogspot.com>



## EL SOLUCIONARIO



SOLUCIONARIOS  
DE LIBROS  
UNIVERSITARIOS

LIBROS UNIVERISTARIOS  
Y SOLUCIONARIOS DE  
MUCHOS DE ESTOS LIBROS

LOS SOLUCIONARIOS  
CONTIENEN TODOS LOS  
EJERCICIOS DEL LIBRO  
RESUELtos Y EXPLICADOS  
DE FORMA CLARA

VISITANOS PARA  
DESARGALOS GRATIS.

**⚠ This work is protected by  
US copyright laws and is for  
instructors' use only.**

**Instructor's Resource Manual**  
*to accompany*

# **DIGITAL FUNDAMENTALS**

**Ninth Edition**

**Thomas L. Floyd**



Upper Saddle River, New Jersey  
Columbus, Ohio



**This work is protected by United States copyright laws and is provided solely for the use of instructors in teaching their courses and assessing student learning. Dissemination or sale of any part of this work (including on the World Wide Web) will destroy the integrity of the work and is not permitted. The work and materials from it should never be made available to students except by instructors using the accompanying text in their classes. All recipients of this work are expected to abide by these restrictions and to honor the intended pedagogical purposes and the needs of other instructors who rely on these materials.**

---

**Copyright © 2006 by Pearson Education, Inc., Upper Saddle River, New Jersey 07458.**

Pearson Prentice Hall. All rights reserved. Printed in the United States of America. This publication is protected by Copyright and permission should be obtained from the publisher prior to any prohibited reproduction, storage in a retrieval system, or transmission in any form or by any means, electronic, mechanical, photocopying, recording, or likewise. For information regarding permission(s), write to: Rights and Permissions Department.

**Pearson Prentice Hall™** is a trademark of Pearson Education, Inc.

**Pearson®** is a registered trademark of Pearson plc

**Prentice Hall®** is a registered trademark of Pearson Education, Inc.

Instructors of classes using Floyd, *Digital Fundamentals, Ninth Edition*, may reproduce material from the instructor's resource manual for classroom use.

10 9 8 7 6 5 4 3 2 1



ISBN 0-13-194611-0

## CONTENTS

<b>PART 1: PROBLEM SOLUTIONS.....</b>	<b>1</b>
CHAPTER 1 Digital Concepts.....	2
CHAPTER 2 Number Systems, Operations, and Codes.....	7
CHAPTER 3 Logic Gates.....	24
CHAPTER 4 Boolean Algebra and Logic Simplification.....	35
CHAPTER 5 Combinational Logic Analysis.....	60
CHAPTER 6 Functions of Combinational Logic.....	92
CHAPTER 7 Latches, Flip-Flops, and Timers.....	111
CHAPTER 8 Counters.....	124
CHAPTER 9 Shift Registers .....	152
CHAPTER 10 Memory and Storage.....	168
CHAPTER 11 Programmable Logic and Software.....	178
CHAPTER 12 Introduction to Computers.....	188
CHAPTER 13 Introduction to Digital Signal Processing.....	194
CHAPTER 14 Integrated Circuit Technologies.....	202
<b>PART 2: DIGITAL SYSTEM APPLICATION SOLUTIONS.....</b>	<b>209</b>
CHAPTER 4 .....	210
CHAPTER 5 .....	213
CHAPTER 6 .....	215
CHAPTER 7 .....	217
CHAPTER 8 .....	218
CHAPTER 9 .....	220
CHAPTER 10 .....	222
CHAPTER 11 .....	224
<b>PART 3: OVERVIEW OF IEEE STD.91-1984.....</b>	<b>227</b>
<b>PART 4: LABORATORY SOLUTIONS FOR <i>EXPERIMENTS IN DIGITAL FUNDAMENTALS</i> by David Buchla.....</b>	<b>253</b>

To access supplementary materials online, instructors need to request an instructor access code. Go to [www.prenhall.com](http://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 the site and log on for full instructions on downloading the materials you wish to use.

NOTE: For access to hidden faults in Multisim circuits, the password is *book*.

# **PART 1**

## *Problem Solutions*

---

## **CHAPTER 1**

### **DIGITAL CONCEPTS**

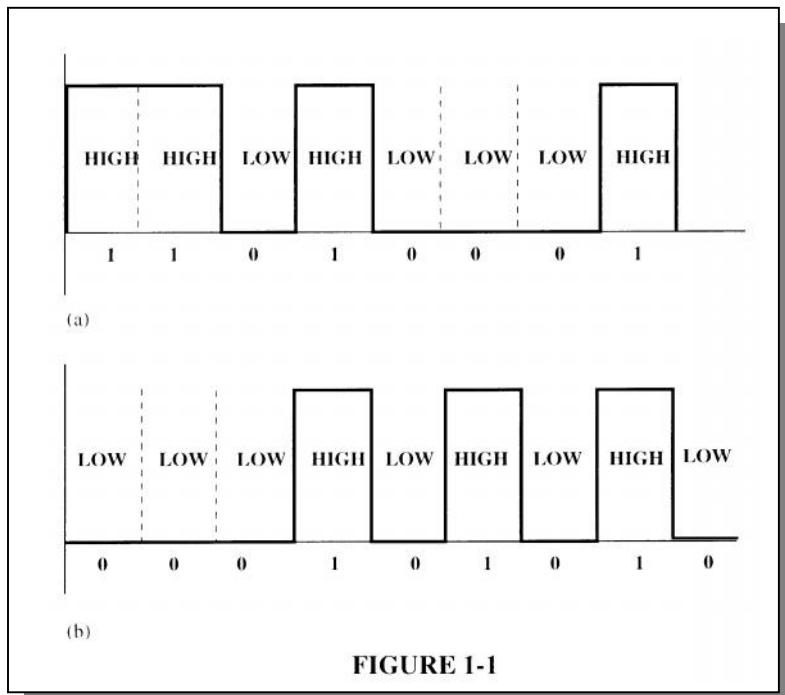
---

#### ***Section 1-1 Digital and Analog Quantities***

1. Digital data can be transmitted and stored more efficiently and reliably than analog data. Also, digital circuits are simpler to implement and there is a greater immunity to noisy environments.
2. Pressure is an analog quantity.

#### ***Section 1-2 Binary Digits, Logic Levels, and Digital Waveforms***

3. HIGH = 1; LOW = 0. See Figure 1-1.

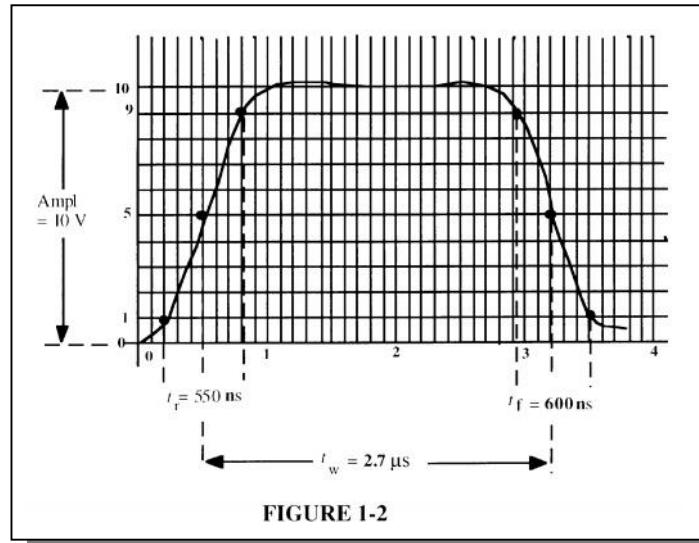


**FIGURE 1-1**

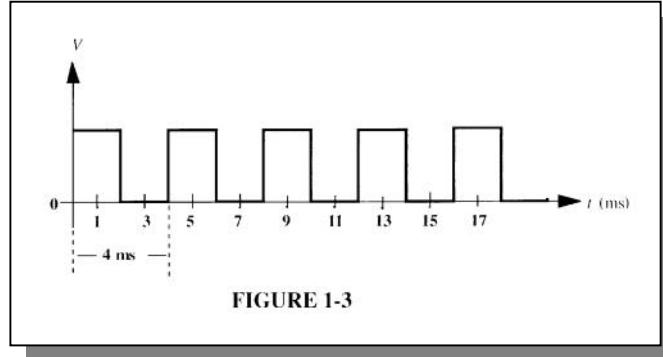
4. A 1 is a HIGH and a 0 is a LOW:
  - (a) HIGH, LOW, HIGH, HIGH, HIGH, LOW, HIGH
  - (b) HIGH, HIGH, HIGH, LOW, HIGH, LOW, LOW, HIGH

## Chapter 1

5. See Figure 1-2.



6.  $T = 4$  ms. See Figure 1-3.



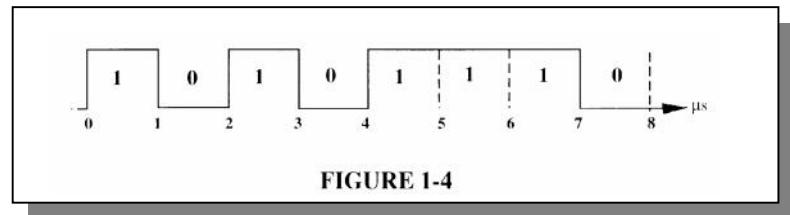
7.  $f = \frac{1}{T} = \frac{1}{4 \text{ ms}} = 0.25 \text{ kHz} = 250 \text{ Hz}$

8. The waveform in Figure 1-61 is **periodic** because it repeats at a fixed interval.

9.  $t_w = 2 \text{ ms}; T = 4 \text{ ms}$

$$\% \text{ duty cycle} = \left( \frac{t_w}{T} \right) 100 = \left( \frac{2 \text{ ms}}{4 \text{ ms}} \right) 100 = 50\%$$

10. See Figure 1-4.



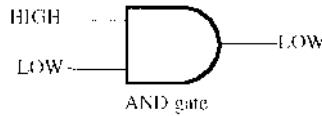
11. Each bit time = 1 μs  
 Serial transfer time = (8 bits)(1 μs/bit) = 8 μs

Parallel transfer time = 1 bit time = 1 μs

### **Section 1-3 Basic Logic Operations**

## Chapter 1

12. An AND gate produces a HIGH output only when *all* of its inputs are HIGH.
13. AND gate. See Figure 1-5.

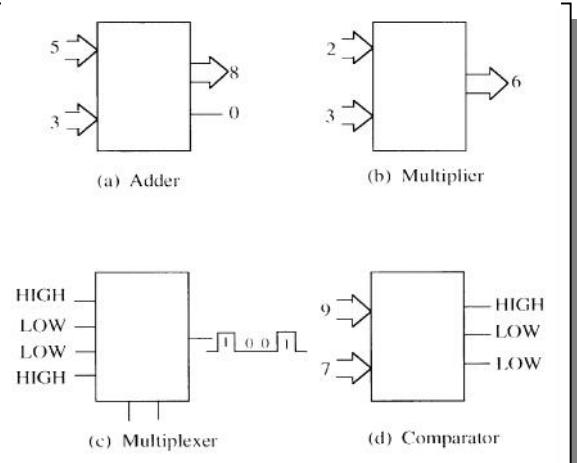


**FIGURE 1-5**

14. An OR gate produces a HIGH output when *either or both* inputs are HIGH. An exclusive-OR gate produces a HIGH if one input is HIGH and the other LOW.

### **Section 1-4 Overview of Basic Logic Functions**

15. See Figure 1-6.

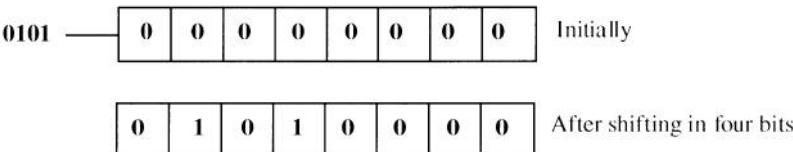


**FIGURE 1-6**

16.  $T = \frac{1}{10 \text{ kHz}} = 100 \mu\text{s}$

Pulses counted =  $\frac{100 \text{ ms}}{100 \mu\text{s}} = 1000$

17. See Figure 1-7.

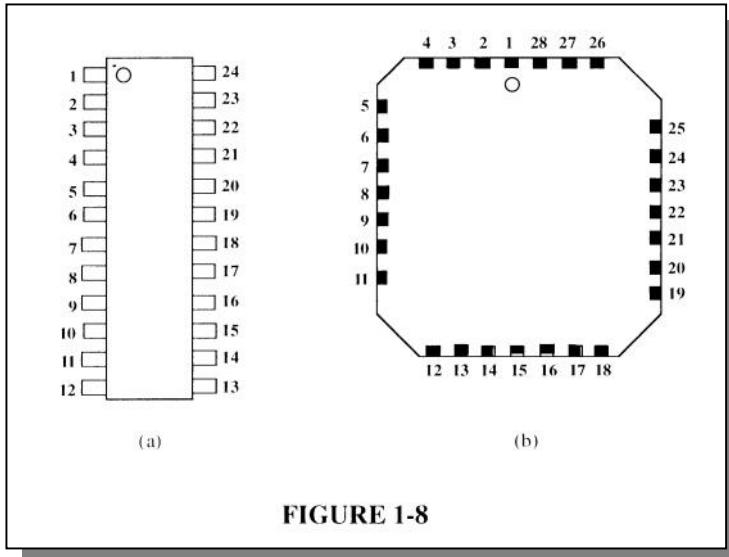


**FIGURE 1-7**

## *Chapter 1*

### **Section 1-5 Fixed-Function Integrated Circuits**

18. Circuits with complexities of from 100 to 10,000 equivalent gates are classified as large scale integration (LSI).
19. The pins of an SMT are soldered to the pads on the surface of a pc board, whereas the pins of a DIP feed through and are soldered to the opposite side. Pin spacing on SMTs is less than on DIPs and therefore SMT packages are physically smaller and require less surface area on a pc board.
20. See Figure 1-8.



**FIGURE 1-8**

## **Section 1-6 Introduction to Programmable Logic**

21. The following do not describe PLDs: ABEL, CUPL
22. SPLD: Simple Programmable Logic Device  
CPLD: Complex Programmable Logic Device  
HDL: Hardware Description Language  
FPGA: Field-Programmable Gate Array  
GAL: Generic Array Logic
23. (a) Design entry: The step in a programmable logic design flow where a description of the circuit is entered in either schematic (graphic) form or in text form using an HDL.  
(b) Simulation: The step in a design flow where the entered design is simulated based on defined input waveforms.  
(c) Compilation: A program process that controls the design flow process and translates a design source code to object code for testing and downloading.  
(d) Download: The process in which the design is transferred from software to hardware.
24. Place and route or fitting is the process where the logic structures described by the netlist are mapped into the actual structure of the specific target device. This results in an output called a bitstream.

## **Section 1-7 Test and Measurement Instruments**

25. Amplitude = top of pulse minus base line  
 $V = 8 \text{ V} - 1 \text{ V} = 7 \text{ V}$
26. A flashing probe lamp indicates a continuous sequence of pulses (pulse train).

## **Digital System Application**

27. A system is a combination of logic elements and functions arranged and interconnected to perform specified tasks.
28. The binary number representing the total number of tablets is converted from parallel to serial form by the multiplexer and sent, one bit at a time, to the remote location where the demultiplexer converts the serial number back to parallel form for decoding and display.
29. A new number of tablets per bottle can be entered with the keypad.

---

## **CHAPTER 2**

### **NUMBER SYSTEMS, OPERATIONS, AND CODES**

---

**Section 2-1 Decimal Numbers**

1. (a)  $1386 = 1 \times 10^3 + 3 \times 10^2 + 8 \times 10^1 + 6 \times 10^0$   
 $= 1 \times 1000 + 3 \times 100 + 8 \times 10 + 6 \times 1$   
 The digit 6 has a weight of  $10^0 = 1$
- (b)  $54,692 = 5 \times 10^4 + 4 \times 10^3 + 6 \times 10^2 + 9 \times 10^1 + 2 \times 10^0$   
 $= 5 \times 10,000 + 4 \times 1000 + 6 \times 100 + 9 \times 10 + 2 \times 1$   
 The digit 6 has a weight of  $10^2 = 100$
- (c)  $671,920 = 6 \times 10^5 + 7 \times 10^4 + 1 \times 10^3 + 9 \times 10^2 + 2 \times 10^1 + 0 \times 10^0$   
 $= 6 \times 100,000 + 7 \times 10,000 + 1 \times 1000 + 9 \times 100 + 2 \times 10 + 0 \times 1$   
 The digit 6 has a weight of  $10^5 = 100,000$
2. (a)  $10 = 10^1$  (b)  $100 = 10^2$   
 (c)  $10,000 = 10^4$  (d)  $1,000,000 = 10^6$
3. (a)  $471 = 4 \times 10^2 + 7 \times 10^1 + 1 \times 10^0$   
 $= 4 \times 100 + 7 \times 10 + 1 \times 1$   
 $= 400 + 70 + 1$
- (b)  $9,356 = 9 \times 10^3 + 3 \times 10^2 + 5 \times 10^1 + 6 \times 10^0$   
 $= 9 \times 1000 + 3 \times 100 + 5 \times 10 + 6 \times 1$   
 $= 9,000 + 300 + 50 + 6$
- (c)  $125,000 = 1 \times 10^5 + 2 \times 10^4 + 5 \times 10^3$   
 $= 1 \times 100,000 + 2 \times 10,000 + 5 \times 1000$   
 $= 100,000 + 20,000 + 5,000$

4. The highest four-digit decimal number is 9999.

**Section 2-2 Binary Numbers**

5. (a)  $11 = 1 \times 2^1 + 1 \times 2^0 = 2 + 1 = 3$   
 (b)  $100 = 1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0 = 4$   
 (c)  $111 = 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 4 + 2 + 1 = 7$   
 (d)  $1000 = 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 0 \times 2^0 = 8$   
 (e)  $1001 = 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 8 + 1 = 9$   
 (f)  $1100 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0 = 8 + 4 = 12$   
 (g)  $1011 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 8 + 2 + 1 = 11$   
 (h)  $1111 = 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 8 + 4 + 2 + 1 = 15$

- 6.**
- (a)  $1110 = 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 = 8 + 4 + 2 = 14$
  - (b)  $1010 = 1 \times 2^3 + 1 \times 2^1 = 8 + 2 = 10$
  - (c)  $11100 = 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 = 16 + 8 + 4 = 28$
  - (d)  $10000 = 1 \times 2^4 = 16$
  - (e)  $10101 = 1 \times 2^4 + 1 \times 2^2 + 1 \times 2^0 = 16 + 4 + 1 = 21$
  - (f)  $11101 = 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^0 = 16 + 8 + 4 + 1 = 29$
  - (g)  $10111 = 1 \times 2^4 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 16 + 4 + 2 + 1 = 23$
  - (h)  $11111 = 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 16 + 8 + 4 + 2 + 1 = 31$
- 7.**
- (a)  $110011.11 = 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2}$   
 $= 32 + 16 + 2 + 1 + 0.5 + 0.25 = 51.75$
  - (b)  $101010.01 = 1 \times 2^5 + 1 \times 2^3 + 1 \times 2^1 + 1 \times 2^{-2} = 32 + 8 + 2 + 0.25$   
 $= 42.25$
  - (c)  $1000001.111 = 1 \times 2^6 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-3}$   
 $= 64 + 1 + 0.5 + 0.25 + 0.125 = 65.875$
  - (d)  $1111000.101 = 1 \times 2^6 + 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^{-1} + 1 \times 2^{-3}$   
 $= 64 + 32 + 16 + 8 + 0.5 + 0.125 = 120.625$
  - (e)  $1011100.10101 = 1 \times 2^6 + 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^{-1} + 1 \times 2^{-3} + 1 \times 2^{-5}$   
 $= 64 + 16 + 8 + 4 + 0.5 + 0.125 + 0.03125 = 92.65625$
  - (f)  $1110001.0001 = 1 \times 2^6 + 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^0 + 1 \times 2^{-4}$   
 $= 64 + 32 + 16 + 1 + 0.0625 = 113.0625$
  - (g)  $1011010.1010 = 1 \times 2^6 + 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^1 + 1 \times 2^{-1} + 1 \times 2^{-3}$   
 $= 64 + 16 + 8 + 2 + 0.5 + 0.125 = 90.625$
  - (h)  $1111111.11111 = 1 \times 2^6 + 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-3} + 1 \times 2^{-4} + 1 \times 2^{-5}$   
 $= 64 + 32 + 16 + 8 + 4 + 2 + 1 + 0.5 + 0.25 + 0.125 + 0.0625 + 0.03125 = 127.96875$
- 8.**
- |                         |                         |
|-------------------------|-------------------------|
| (a) $2^2 - 1 = 3$       | (b) $2^3 - 1 = 7$       |
| (c) $2^4 - 1 = 15$      | (d) $2^5 - 1 = 31$      |
| (e) $2^6 - 1 = 63$      | (f) $2^7 - 1 = 127$     |
| (g) $2^8 - 1 = 255$     | (h) $2^9 - 1 = 511$     |
| (i) $2^{10} - 1 = 1023$ | (j) $2^{11} - 1 = 2047$ |
- 9.**
- (a)  $(2^4 - 1) < 17 < (2^5 - 1)$ ; 5 bits
  - (b)  $(2^5 - 1) < 35 < (2^6 - 1)$ ; 6 bits
  - (c)  $(2^5 - 1) < 49 < (2^6 - 1)$ ; 6 bits
  - (d)  $(2^6 - 1) < 68 < (2^7 - 1)$ ; 7 bits
  - (e)  $(2^6 - 1) < 81 < (2^7 - 1)$ ; 7 bits
  - (f)  $(2^6 - 1) < 114 < (2^7 - 1)$ ; 7 bits
  - (g)  $(2^7 - 1) < 132 < (2^8 - 1)$ ; 8 bits
  - (h)  $(2^7 - 1) < 205 < (2^8 - 1)$ ; 8 bits

10. (a) 0 through 7:  
 000, 001, 010, 011, 100, 101, 110, 111  
 (b) 8 through 15:  
 1000, 1001, 1010, 1011, 1100, 1101, 1110, 1111  
 (c) 16 through 31:  
 10000, 10001, 10010, 10011, 10100, 10101, 10110, 10111, 11000,  
 11001, 11010,  
 11011, 11100, 11101, 11110, 11111  
 (d) 32 through 63:  
 100000, 100001, 100010, 100011, 100100, 100101, 100110, 100111,  
 10100, 101001, 101010, 101011, 101100, 101101, 101110, 101111,  
 110000, 110001, 110010, 110011,  
 110100, 110101, 110110, 110111, 111000, 111001, 111010, 111011,  
 111100, 111101,  
 111110, 111111  
 (e) 64 through 75:  
 1000000, 1000001, 1000010, 1000011, 1000100, 1000101, 1000110,  
 1000111,  
 1001000, 1001001, 1001010, 1001011  
 100111,

### **Section 2-3 Decimal-to-Binary Conversion**

11. (a)  $10 = 8 + 2 = 2^3 + 2^1 = 1010$   
 (b)  $17 = 16 + 1 = 2^4 + 2^0 = 10001$   
 (c)  $24 = 16 + 8 = 2^4 + 2^3 = 11000$   
 (d)  $48 = 32 + 16 = 2^5 + 2^4 = 110000$   
 (e)  $61 = 32 + 16 + 8 + 4 + 1 = 2^5 + 2^4 + 2^3 + 2^2 + 2^0 = 111101$   
 (f)  $93 = 64 + 16 + 8 + 4 + 1 = 2^6 + 2^4 + 2^3 + 2^2 + 2^0 = 1011101$   
 (g)  $125 = 64 + 32 + 16 + 8 + 4 + 1 = 2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 2^0 = 1111101$   
 (h)  $186 = 128 + 32 + 16 + 8 + 2 = 2^7 + 2^5 + 2^4 + 2^3 + 2^1 = 10111010$
12. (a)  $0.32 \cong 0.00 + 0.25 + 0.0625 + 0.0 + 0.0 + 0.0078125 = 0.0101001$   
 (b)  $0.246 \cong 0.0 + 0.0 + 0.125 + 0.0625 + 0.03125 + 0.015625 = 0.001111$   
 (c)  $0.0981 \cong 0.0 + 0.0 + 0.0 + 0.0625 + 0.03125 + 0.0 + 0.0 + 0.00390625 = 0.0001101$

<b>13.</b> (a) $\frac{15}{2} = 7, R = 0$ 1 (LSB) $\frac{7}{2} = 3, R = 1$ 1 $\frac{3}{2} = 1, R = 1$ 1 $\frac{1}{2} = 0, R = 1$ 1 (MSB)	$\frac{2}{2} = 1, R = 1$ $\frac{1}{2} = 0, R = 0$ $\frac{5}{2} = 2, R = 1$ $\frac{2}{2} = 1, R = 0$ $\frac{1}{2} = 0, R = 1$ (MSB)	(b) $\frac{21}{2} = 10, R = 1$ 1 (LSB) $\frac{10}{2} = 5, R = 0$ $\frac{5}{2} = 2, R = 1$ $\frac{2}{2} = 1, R = 0$ $\frac{1}{2} = 0, R = 1$
(d) $\frac{34}{2} = 17, R = 0$ (LSB) $\frac{17}{2} = 8, R = 1$ $\frac{8}{2} = 4, R = 0$ $\frac{4}{2} = 2, R = 0$ $\frac{2}{2} = 1, R = 0$ $\frac{1}{2} = 0, R = 1$ (MSB)	(e) $\frac{40}{2} = 20, R = 0$ $\frac{20}{2} = 10, R = 0$ $\frac{10}{2} = 5, R = 0$ $\frac{5}{2} = 2, R = 1$ $\frac{2}{2} = 1, R = 0$ $\frac{1}{2} = 0, R = 1$ (MSB)	(f) $\frac{73}{2} = 36, R = 1$ 1 (LSB) $\frac{36}{2} = 18, R = 0$ $\frac{18}{2} = 9, R = 0$ $\frac{9}{2} = 4, R = 1$ $\frac{4}{2} = 2, R = 0$ $\frac{2}{2} = 1, R = 0$
(g) $\frac{65}{2} = 32, R = 1$ (LSB) $\frac{32}{2} = 16, R = 0$ $\frac{16}{2} = 8, R = 0$ $\frac{8}{2} = 4, R = 0$ $\frac{4}{2} = 2, R = 0$		

$$\begin{array}{ll}
 \frac{1}{2} = 0, \quad R = 1 \quad (\text{MSB}) & (c) \quad \frac{28}{2} = 14, \quad R = 0 \\
 & (\text{LSB}) \\
 & \frac{14}{2} = 7, \quad R = 0 \\
 & \frac{7}{2} = 3, \quad R = 1 \\
 & \frac{3}{2} = 1, \quad R = 1 \\
 & \frac{1}{2} = 0, \quad R = 1 \quad (\text{MSB}) \\
 \\ 
 (f) \quad \frac{59}{2} = 29, \quad R = 1 & \\
 & (\text{LSB}) \\
 & \frac{29}{2} = 14, \quad R = 1 \\
 & \frac{14}{2} = 7, \quad R = 0 \\
 & \frac{7}{2} = 3, \quad R = 1 \\
 & \frac{3}{2} = 1, \quad R = 1 \\
 & \frac{1}{2} = 0, \quad R = 1 \quad (\text{MSB})
 \end{array}$$

<b>14.</b>	(a)	$0.98 \times 2 = 1.96$	1 (MSB)	(b)	$0.347 \times 2 =$
	0.694 0 (MSB)	$0.96 \times 2 = 1.92$	1		$0.694 \times 2 =$
1.388 1		$0.92 \times 2 = 1.84$	1		$0.388 \times 2 =$
0.776 0		$0.84 \times 2 = 1.68$	1		$0.776 \times 2 =$
1.552 1		$0.68 \times 2 = 1.36$	1		$0.552 \times 2 =$
1.104 1		$0.36 \times 2 = 0.72$	0		$0.104 \times 2 =$
0.208 0					$0.208 \times 2 =$
		continue if more accuracy is desired			
0.416 0					continue if
		0.111110			
more accuracy is desired					0.0101100
	(c)	$0.9028 \times 2 = 1.8056$	1 (MSB)		
		$0.8056 \times 2 = 1.6112$	1		
		$0.6112 \times 2 = 1.2224$	1		
		$0.2224 \times 2 = 0.4448$	0		
		$0.4448 \times 2 = 0.8896$	0		
		$0.8896 \times 2 = 1.7792$	1		
		$0.7792 \times 2 = 1.5584$	1		
		continue if more accuracy is desired			
		0.1110011			

### **Section 2-4 Binary Arithmetic**

<b>15.</b>	(a)	$\begin{array}{r} 11 \\ + 01 \\ \hline 100 \end{array}$	(b)	$\begin{array}{r} 10 \\ + 10 \\ \hline 100 \end{array}$	(c)	$\begin{array}{r} 101 \\ + 011 \\ \hline 1000 \end{array}$
	(d)	$\begin{array}{r} 111 \\ + 110 \\ \hline 1101 \end{array}$	(e)	$\begin{array}{r} 1001 \\ + 0101 \\ \hline 1110 \end{array}$	(f)	$\begin{array}{r} 1101 \\ + 1011 \\ \hline 11000 \end{array}$
<b>16.</b>	(a)	$\begin{array}{r} 11 \\ - 01 \\ \hline 10 \end{array}$	(b)	$\begin{array}{r} 101 \\ - 100 \\ \hline 001 \end{array}$	(c)	$\begin{array}{r} 110 \\ - 101 \\ \hline 001 \end{array}$
	(d)	$\begin{array}{r} 1110 \\ - 0011 \\ \hline 1011 \end{array}$	(e)	$\begin{array}{r} 1100 \\ - 1001 \\ \hline 0011 \end{array}$	(f)	$\begin{array}{r} 11010 \\ - 10111 \\ \hline 00011 \end{array}$

<b>17.</b> (a) $\begin{array}{r} 11 \\ \times 11 \\ \hline 11 \end{array}$	(b) $\begin{array}{r} 100 \\ \times 10 \\ \hline 000 \end{array}$	(c) $\begin{array}{r} 111 \\ \times 101 \\ \hline 111 \end{array}$	(d) $\begin{array}{r} 1001 \\ \times 110 \\ \hline 0000 \end{array}$
$\begin{array}{r} 11 \\ \times 11 \\ \hline 100 \\ \hline 1001 \end{array}$	$\begin{array}{r} 100 \\ \times 100 \\ \hline 1000 \\ \hline 1000 \end{array}$	$\begin{array}{r} 000 \\ \times 111 \\ \hline 111 \\ \hline 1001 \end{array}$	$\begin{array}{r} 1001 \\ \times 1001 \\ \hline 110110 \end{array}$
(e) $\begin{array}{r} 1101 \\ \times 1101 \\ \hline 1101 \\ 0000 \\ \hline 1101 \\ 1101 \\ \hline 10101001 \end{array}$	(f) $\begin{array}{r} 1110 \\ \times 1101 \\ \hline 1110 \\ 0000 \\ \hline 1110 \\ 1110 \\ \hline 10110110 \end{array}$		

**18.** (a)  $\frac{100}{10} = 010$       (b)  $\frac{1001}{0011} = 0011$  (c)  $\frac{1100}{0100} = 0011$

### **Section 2-5 1's and 2's Complements of Binary Numbers**

- 19.** (a) The 1's complement of 101 is 010.  
 (b) The 1's complement of 110 is 001.  
 (c) The 1's complement of 1010 is 0101.  
 (d) The 1's complement of 11010111 is 00101000.  
 (e) The 1's complement of 1110101 is 0001010.  
 (f) The 1's complement of 00001 is 11110.

**20.** Take the 1's complement and add 1:

(a) $01 + 1 = 10$ (c) $0110 + 1 = 0111$ (e) $00011 + 1 = 00100$ (g) $01001111 + 1 = 01010000$	(b) $000 + 1 = 001$ (d) $0010 + 1 = 0011$ (f) $01100 + 1 = 01101$ (h) $11000010 + 1 = 11000011$
--	--

### **Section 2-6 Signed Numbers**

- 21.** (a) Magnitude of 29 = 0011101  
 $= 1010101$   
 $+ 29 = 00011101$       (b) Magnitude of 85  
 $-85 = 11010101$
- (c) Magnitude of  $100_{10} = 1100100$       (d) Magnitude of  
 $123 = 1111011$   
 $+100 = 01100100$        $-123 = 11111011$
- 22.** (a) Magnitude of 34 = 0100010  
 $= 0111001$   
 $-34 = 11011101$       (b) Magnitude of 57  
 $+57 = 00111001$
- (c) Magnitude of 99 = 1100011  
 $115 = 1110011$   
 $-99 = 10011100$       (d) Magnitude of  
 $+115 = 01110011$
- 23.** (a) Magnitude of 12 = 1100      (b) Magnitude of 68

- = 1000100  
 $+12 = 00001100$        $-68 = 10111100$
- (c) Magnitude of  $101_{10} = 1100101$   
 $125 = 1111101$   
 $+101_{10} = 01100101$       (d) Magnitude of  
 $-125 = 10000011$
- 24.** (a)  $10011001 = -25$       (b)  $01110100 = +116$       (c)  
 $10111111 = -63$
- 25.** (a)  $10011001 = -(01100110) = -102$   
(b)  $01110100 = +(1110100) = +116$   
(c)  $10111111 = -(1000000) = -64$
- 26.** (a)  $10011001 = -(1100111) = -103$   
(b)  $01110100 = +(1110100) = +116$   
(c)  $10111111 = -(1000001) = -65$
- 27.** (a)  $0111110000101011 \rightarrow \text{sign} = 0$   
 $1.11110000101011 \times 2^{14} \rightarrow \text{exponent} = 127 + 14 + 141 = 10001101$   
Mantissa =  $111100001010110000000000$   
**01000110111100001010110000000000**
- (b)  $100110000011000 \rightarrow \text{sign} = 1$   
 $1.10000011000 \times 2^{11} \rightarrow \text{exponent} = 127 + 11 = 138 = 10001010$   
Mantissa =  $110000011000000000000000$   
**11000101011000001100000000000000**
- 28.** (a)  $11000000101001001110001000000000$   
Sign = 1  
Exponent =  $10000001 = 129 - 127 = 2$   
Mantissa =  $1.01001001110001 \times 2^2 = 101.001001110001$   
 $-101.001001110001 = -5.15258789$
- (b)  $0110011001000011110100100000000$   
Sign = 0  
Exponent =  $11001100 = 204 - 127 = 77$   
Mantissa =  $1.10000111101001$   
**1.10000111101001  $\times 2^{77}$**
- Section 2-7 Arithmetic Operations with Signed Numbers**
- 29.** (a)  $33 = 00100001$   
 $00100001$   
 $00001111$   
 $00110000$
- (b)  $56 = 00111000$   
 $00111000$   
 $27 = 00011011$   
 $11100101$   
 $-27 = 11100101$   
 $00011101$
- (c)  $46 = 00101110$   
 $11010010$   
 $-46 = 11010010$   
 $00011001$   
 $11101011$
- (d)  $110_{10} = 01101110$   
 $10010010$   
 $-110_{10} = 10010010$   
 $10101100$   
 $10011110$   
 $-84 = 01010100$   
 $10101100$

30. (a) 
$$\begin{array}{r} 00010110 \\ + 00110011 \\ \hline 01001001 \end{array}$$

(b) 
$$\begin{array}{r} 01110000 \\ + 10101111 \\ \hline 100011111 \end{array}$$

31. (a) 
$$\begin{array}{r} 10001100 \\ + 00111001 \\ \hline 11000101 \end{array}$$

(b) 
$$\begin{array}{r} 11011001 \\ + 11100111 \\ \hline 11000000 \end{array}$$

32. (a) 
$$\begin{array}{r} 00110011 \\ 00110011 \\ - 00010000 \\ \hline 11110000 \end{array}$$

(b) 
$$\begin{array}{r} 01100101 \\ 01100101 \\ + 00011000 \\ \hline 01111101 \end{array}$$

33. 
$$\begin{array}{r} 01101010 \\ \times 11110001 \\ \hline \end{array}$$

$$\begin{array}{r} 01101010 \\ \times 00001111 \\ \hline 01101010 \\ 01101010 \\ 100111110 \\ 01101010 \\ 1011100110 \\ 01101010 \\ \hline 11000110110 \end{array}$$

Changing to 2's complement with sign: 100111001010

34. 
$$\begin{array}{r} 01000100 \\ 00011001 \\ \hline \end{array} = 00000010$$

$$\frac{68}{25} = 2, \text{ remainder of } 18$$

### Section 2-8 Hexadecimal Numbers

35. (a)  $38_{16} = 0011\ 1000$   
 (b)  $59_{16} = 0101\ 1001$   
 (c)  $A14_{16} = 1010\ 0001\ 0100$   
 (d)  $5C8_{16} = 0101\ 1100\ 1000$   
 (e)  $4100_{16} = 0100\ 0001\ 0000\ 0000$   
 (f)  $FB17_{16} = 1111\ 1011\ 0001\ 0111$   
 (g)  $8A9D_{16} = 1000\ 1010\ 1001\ 1101$

36. (a)  $1110 = E_{16}$   
 (b)  $10 = 2_{16}$   
 (c)  $0001\ 0111 = 17_{16}$   
 (d)  $1010\ 0110 = A6_{16}$   
 (e)  $0011\ 1111\ 0000 = 3F0_{16}$   
 (f)  $1001\ 1000\ 0010 = 982_{16}$

37. (a)  $23_{16} = 2 \times 16^1 + 3 \times 16^0 = 32 + 3 = 35$   
 (b)  $92_{16} = 9 \times 16^1 + 2 \times 16^0 = 144 + 2 = 146$   
 (c)  $1A_{16} = 1 \times 16^1 + 10 \times 16^0 = 16 + 10 = 26$   
 (d)  $8D_{16} = 8 \times 16^1 + 13 \times 16^0 = 128 + 13 = 141$   
 (e)  $F3_{16} = 15 \times 16^1 + 3 \times 16^0 = 240 + 3 = 243$   
 (f)  $EB_{16} = 14 \times 16^1 + 11 \times 16^0 = 224 + 11 = 235$   
 (g)  $5C2_{16} = 5 \times 16^2 + 12 \times 16^1 + 2 \times 16^0 = 1280 + 192 + 2 = 1474$   
 (h)  $700_{16} = 7 \times 16^2 = 1792$

38. (a)  $\frac{8}{16} = 0, \text{ remainder} = 8$

- hexadecimal number =  $8_{16}$
- (c)  $\frac{33}{16} = 2$ , remainder = 1 (LSD)  
 $\frac{2}{16} = 0$ , remainder = 2  
 hexadecimal number =  $21_{16}$
- (e)  $\frac{284}{16} = 17$ , remainder = 12 =  $C_{16}$  (LSD)  
 $\frac{17}{16} = 1$ , remainder = 1  
 $\frac{1}{16} = 0$ , remainder = 1  
 hexadecimal number =  $11C_{16}$
- (g)  $\frac{4019}{16} = 251$ , remainder = 3 (LSD)  
 $\frac{251}{16} = 15$ , remainder = 11 =  $B_{16}$   
 $\frac{15}{16} = 0$ , remainder = 15 =  $F_{16}$   
 hexadecimal number =  $FB3_{16}$
- (b)  $\frac{14}{16} = 0$ , remainder = 14 =  $E_{16}$   
 hexadecimal number =  $E_{16}$
- (d)  $\frac{52}{16} = 3$ , remainder = 4 (LSD)  
 $\frac{3}{16} = 0$ , remainder = 3  
 hexadecimal number =  $34_{16}$
- (f)  $\frac{2890}{16} = 180$ , remainder = 10 =  $A_{16}$  (LSD)  
 $\frac{180}{16} = 11$ , remainder = 4  
 $\frac{11}{16} = 0$ , remainder = 11 =  $B_{16}$   
 hexadecimal number =  $B4A_{16}$
- (h)  $\frac{6500}{16} = 406$ , remainder = 4 (LSD)  
 $\frac{406}{16} = 25$ , remainder = 6  
 $\frac{25}{16} = 1$ , remainder = 9  
 $\frac{1}{16} = 0$ , remainder = 1  
 hexadecimal number =  $1964_{16}$
- 39.** (a)  $37_{16} + 29_{16} = 60_{16}$   
 (b)  $A0_{16} + 6B_{16} = 10B_{16}$   
 (c)  $FF_{16} + BB_{16} = 1BA_{16}$
- 40.** (a)  $51_{16} - 40_{16} = 11_{16}$   
 (b)  $C8_{16} - 3A_{16} = 8E_{16}$   
 (c)  $FD_{16} - 88_{16} = 75_{16}$

## *Section 2-9 Octal Numbers*

- 41.** (a)  $12_8 = 1 \times 8^1 + 2 \times 8^0 = 8 + 2 = 10$   
 (b)  $27_8 = 2 \times 8^1 + 7 \times 8^0 = 16 + 7 = 23$   
 (c)  $56_8 = 5 \times 8^1 + 6 \times 8^0 = 40 + 6 = 46$   
 (d)  $64_8 = 6 \times 8^1 + 4 \times 8^0 = 48 + 4 = 52$   
 (e)  $103_8 = 1 \times 8^2 + 3 \times 8^0 = 64 + 3 = 67$   
 (f)  $557_8 = 5 \times 8^2 + 5 \times 8^1 + 7 \times 8^0 = 320 + 40 + 7 = 367$   
 (g)  $163_8 = 1 \times 8^2 + 6 \times 8^1 + 3 \times 8^0 = 64 + 48 + 3 = 115$   
 (h)  $1024_8 = 1 \times 8^3 + 2 \times 8^1 + 4 \times 8^0 = 512 + 16 + 4 = 532$   
 (i)  $7765_8 = 7 \times 8^3 + 7 \times 8^2 + 6 \times 8^1 + 5 \times 8^0 = 3584 + 448 + 48 + 5 =$

42.	(a)	$\frac{15}{8} = 1$ , remainder = 7 (LSD)	octal number = $33_8$
	(d)	$\frac{70}{8} = 8$ , remainder = 6 (LSD)	
		$\frac{8}{8} = 1$ , remainder = 0	
		$\frac{1}{8} = 0$ , remainder = 1	
		octal number = $106_8$	
	(f)	$\frac{142}{8} = 17$ , remainder = 6 (LSD)	
		$\frac{17}{8} = 2$ , remainder = 1	
		$\frac{2}{8} = 0$ , remainder = 2	
		octal number = $216_8$	
	(h)	$\frac{435}{8} = 54$ , remainder = 3 (LSD)	
		$\frac{54}{8} = 6$ , remainder = 6	
		$\frac{6}{8} = 0$ , remainder = 6	
		octal number = $663_8$	
	(g)	$\frac{219}{8} = 27$ , remainder = 3 (LSD)	
		$\frac{27}{8} = 3$ , remainder = 3	
		$\frac{3}{8} = 0$ , remainder = 3	
		octal number = $333_8$	
	(b)	$\frac{27}{8} = 3$ , remainder = 3 (LSD)	
		$\frac{3}{8} = 0$ , remainder = 3	



- 43.** (a)  $13_8 = 001\ 011$   
 (b)  $57_8 = 101\ 111$   
 (c)  $101_8 = 001\ 000\ 001$   
 (d)  $321_8 = 011\ 010\ 001$   
 (e)  $540_8 = 101\ 100\ 000$   
 (f)  $4653_8 = 100\ 110\ 101\ 011$   
 (g)  $13271_8 = 001\ 011\ 010\ 111\ 001$   
 (h)  $45600_8 = 100\ 101\ 110\ 000\ 000$   
 (i)  $100213_8 = 001\ 000\ 000\ 010\ 001\ 011$

- 44.** (a)  $111 = 7_8$   
 (b)  $010 = 2_8$   
 (c)  $110\ 111 = 67_8$   
 (d)  $101\ 010 = 52_8$   
 (e)  $001\ 100 = 14_8$   
 (f)  $001\ 011\ 110 = 136_8$   
 (g)  $101\ 100\ 011\ 001 = 5431_8$   
 (h)  $010\ 110\ 000\ 011 = 2603_8$   
 (i)  $111\ 111\ 101\ 111\ 000 = 77570_8$

### **Section 2-10 Binary Coded Decimal (BCD)**

- 45.** (a)  $10 = 0001\ 0000$   
 (b)  $13 = 0001\ 0011$   
 (c)  $18 = 0001\ 1000$   
 (d)  $21 = 0010\ 0001$   
 (e)  $25 = 0010\ 0101$   
 (f)  $36 = 0011\ 0110$   
 (g)  $44 = 0100\ 0100$   
 (h)  $57 = 0101\ 0111$   
 (i)  $69 = 0110\ 1001$   
 (j)  $98 = 1001\ 1000$   
 (k)  $125 = 0001\ 0010\ 0101$   
 (l)  $156 = 0001\ 0101\ 0110$

- 46.** (a)  $10 = 1010_2$       4 bits binary, 8 bits BCD  
 (b)  $13 = 1101_2$       4 bits binary, 8 bits BCD  
 (c)  $18 = 10010_2$       5 bits binary, 8 bits BCD  
 (d)  $21 = 10101_2$       5 bits binary, 8 bits BCD  
 (e)  $25 = 11001_2$       5 bits binary, 8 bits BCD  
 (f)  $36 = 100100_2$  6 bits binary, 8 bits BCD  
 (g)  $44 = 101100_2$  6 bits binary, 8 bits BCD  
 (h)  $57 = 111001_2$  6 bits binary, 8 bits BCD  
 (i)  $69 = 1000101_2$       7 bits binary, 8 bits BCD  
 (j)  $98 = 1100010_2$       7 bits binary, 8 bits BCD  
 (k)  $125 = 1111101_2$       7 bits binary, 12 bits BCD  
 (l)  $156 = 10011100_2$       8 bits binary, 12 bits BCD

**47.** (a) 104 = 0001 0000 0100

(b) 128 = 0001 0010 1000

(c) 132 = 0001 0011 0010

(d) 150 = 0001 0101 0000

(e) 186 = 0001 1000 0110

(f) 210 = 0010 0001 0000

(g) 359 = 0011 0101 1001

(h) 547 = 0101 0100 0111

(i) 1051 = 0001 0000 0101 0001

**48.** (a) 0001 = 1

(b) 0110 = 6

(c) 1001 = 9

(d) 0001 1000 = 18

(e) 0001 1001 = 19

(f) 0011 0010 = 32

(g) 0100 0101 = 45

(h) 1001 1000 = 98

(i) 1000 0111 0000 = 870

**49.** (a) 1000 0000 = 80

(b) 0010 0011 0111 = 237

(c) 0011 0100 0110 = 346

(d) 0100 0010 0001 = 421

(e) 0111 0101 0100 = 754

(f) 1000 0000 0000 = 800

(g) 1001 0111 1000 = 978

(h) 0001 0110 1000 0011 = 1683

(i) 1001 0000 0001 1000 = 9018

(j) 0110 0110 0110 0111 = 6667

$$50. \quad (a) \begin{array}{r} 0010 \\ + 0001 \\ \hline 0011 \end{array}$$

$$(b) \begin{array}{r} 0101 \\ + 0011 \\ \hline 1000 \end{array}$$

$$(c) \begin{array}{r} 0111 \\ + 0010 \\ \hline 1001 \end{array}$$

$$(d) \begin{array}{r} 1000 \\ + 0001 \\ \hline 1001 \end{array}$$

$$(e) \begin{array}{r} 00011000 \\ + 00010001 \\ \hline 00101001 \end{array}$$

$$(f) \begin{array}{r} 01100100 \\ + 00110011 \\ \hline 10010111 \end{array}$$

$$(g) \begin{array}{r} 01000000 \\ + 01000111 \\ \hline 10000111 \end{array}$$

$$(h) \begin{array}{r} 10000101 \\ + 01000111 \\ \hline 10000111 \end{array}$$

51. (a)

$$\begin{array}{r}
 1000 \\
 + 0110 \\
 \hline
 11\text{ }_{\text{---}}^{\text{---}} \textit{interval} \\
 + 0110 \\
 \hline
 00010100
 \end{array}$$

(b)

$$\begin{array}{r}
 0111 \\
 + 0101 \\
 \hline
 110\text{ }_{\text{---}}^{\text{---}} \textit{interval} \\
 + 0110 \\
 \hline
 00010010
 \end{array}$$

(c)

$$\begin{array}{r}
 1001 \\
 + 1000 \\
 \hline
 100\text{ }_{\text{---}}^{\text{---}} \textit{interval} \\
 + 0110 \\
 \hline
 00010111
 \end{array}$$

(d)

$$\begin{array}{r}
 1001 \\
 + 011\text{ }_{\text{---}}^{\text{---}} \textit{interval} \\
 \hline
 1000\text{ }_{\text{---}}^{\text{---}} \textit{interval} \\
 + 0110 \\
 \hline
 00010110
 \end{array}$$

(e)

$$\begin{array}{r}
 00100101 \\
 + 00100111 \\
 \hline
 010011\text{ }_{\text{---}}^{\text{---}} \textit{interval} \\
 + 0110 \\
 \hline
 01010010
 \end{array}$$

(f)

$$\begin{array}{r}
 01010001 \\
 + 01011000 \\
 \hline
 1010100\text{ }_{\text{---}}^{\text{---}} \textit{interval} \\
 + 0110 \\
 \hline
 000100001001
 \end{array}$$

(g)

$$\begin{array}{r}
 10011000 \\
 + 10010111 \\
 \hline
 1001011\text{ }_{\text{---}}^{\text{---}} \textit{interval} \\
 + 01100110 \\
 \hline
 000110010101
 \end{array}$$

(h)

$$\begin{array}{r}
 010101100001 \\
 + 011100001000 \\
 \hline
 110001101001\text{ }_{\text{---}}^{\text{---}} \textit{interval} \\
 + 0110 \\
 \hline
 0001001001101001
 \end{array}$$

52. (a) 
$$\begin{array}{r} 4 + 3 \\ 0100 \\ + 0011 \\ \hline 0111 \end{array}$$

(b) 
$$\begin{array}{r} 5 + 2 \\ 0101 \\ + 0010 \\ \hline 0111 \end{array}$$

(c) 
$$\begin{array}{r} 6 + 4 \\ 0110 \\ + 0100 \\ \hline 1010 \\ + 0110 \\ \hline 00010000 \end{array}$$

(d) 
$$\begin{array}{r} 17 + 12 \\ 00010111 \\ + 00100010 \\ \hline 00101001 \end{array}$$

(e) 
$$\begin{array}{r} 28 + 23 \\ 00101000 \\ + 00100011 \\ \hline 01001011 \\ + 0110 \\ \hline 01010001 \end{array}$$

(f) 
$$\begin{array}{r} 65 + 58 \\ 01100101 \\ + 01011000 \\ \hline 10111101 \\ + 01100110 \\ \hline 000100100011 \end{array}$$

(g) 
$$\begin{array}{r} 113 + 101 \\ 000100010011 \\ + 000100000001 \\ \hline 001000010100 \end{array}$$

(h) 
$$\begin{array}{r} 295 + 157 \\ 001010010101 \\ + 000101010111 \\ \hline 001111101100 \\ + 01100110 \\ \hline 010001010010 \end{array}$$

### **Section 2-11 Digital Codes**

53. The Gray code makes only one bit change at a time when going from one number in the sequence to the next number.

Gray for  $1111_2 = 1000$   
Gray for  $0000_2 = 0000$

54. (a) 
$$\begin{array}{r} 1 + 1 + 0 + 1 + 1 \\ \text{Binary} \\ 0 + 1 + 0 \\ 1 \quad 1 \quad 1 \quad 1 \quad 1 \\ \text{Gray} \end{array}$$

(b) 
$$\begin{array}{r} 1 + 0 + 0 + 1 + \\ 1 \quad 1 \quad 0 \\ 1 \quad 1 \quad 1 \quad 0 \end{array}$$

(c) 
$$\begin{array}{r} 1 + 1 + 1 + 1 + 0 + 1 + 1 + 1 + 0 + 1 + 1 + 1 + 0 \\ \text{Gray} \\ 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0 \\ \text{Binary} \end{array}$$

55. (a) 
$$\begin{array}{r} 1 \ 0 \ 1 \ 0 \\ \text{Gray} \\ 1 \ 1 \ 0 \ 0 \\ \text{Binary} \end{array}$$

(b) 
$$\begin{array}{r} 0 \ 0 \ 0 \ 1 \ 0 \\ \text{Gray} \\ 0 \ 0 \ 0 \ 1 \ 1 \\ \text{Binary} \end{array}$$

(c) 
$$\begin{array}{r} 1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 1 \\ \text{Gray} \\ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 1 \ 1 \ 0 \\ \text{Binary} \end{array}$$

56. (a)  $1 \rightarrow 00110001$   
(c)  $6 \rightarrow 00110110$   
0011000100110000

(b)  $3 \rightarrow 00110011$

(d)  $10 \rightarrow$

(e)  $18 \rightarrow 0011000100111000$   
(g)  $56 \rightarrow 0011010100110110$   
(i)  $107 \rightarrow 001100010011000000110111$

(f)  $29 \rightarrow 0011001000111001$

(h)  $75 \rightarrow 0011011100110101$

57. (a)  $0011000 \rightarrow \text{CAN}$   
(c)  $0111101 \rightarrow =$   
(d)  $0100011 \rightarrow \#$

	(e) 0111110 → >	(f) 1000010 → <b>B</b>	
58.	1001000 H . 1001000 H . 0100000 #	1100101 e #. 1101111 o . 1110111 y .	1101100 l . 0100000 w . 1101111 o .
			0101110 l . 1110001 # . 0111111 u .
			0100000 a ?
59.	1001000 48 2E	1100101 20 1001000 48 72 0100000 20	1101100 65 6C 1101111 6F 77 1101111 79 6F 0111111 75 3F
60.	30 INPUT A, B		
	3 0 SP I N P U T SP A ,	0110011 0110000 0100000 1001001 1001110 1010000 1010101 1010100 0100000 1000001 0101100 1000010	33 <sub>16</sub> 30 <sub>16</sub> 20 <sub>16</sub> 49 <sub>16</sub> 4E <sub>16</sub> 50 <sub>16</sub> 55 <sub>16</sub> 54 <sub>16</sub> 20 <sub>16</sub> 41 <sub>16</sub> 2C <sub>16</sub> 42 <sub>16</sub>

### **Section 2-12 Error Detection and Correction Codes**

61. Code (b) 011101010 has five 1s, so it is in error.
62. Codes (a) 11110110 and (c) 01010101010101010 are in error because they have an even number of 1s.
63. (a) 1 10100100      (b) 0 00001001      (c) 1 11111110

**64.**  $d = 4$

$$\begin{aligned}2^p &\geq d + p + 1 \\2^3 &= 4 + 3 + 1 = 8 \\p &= 3 \\&\text{parity = even}\end{aligned}$$

Bit Designation	$P_1$	$P_2$	$D_1$	$P_3$	$D_2$	$D_3$	$D_4$
Bit Position	1	2	3	4	5	6	7
Binary Position Number	001	010	011	100	101	110	111
Data Bits ( $D_n$ )			1		1	0	0
Parity Bits ( $P_n$ )	0	1		1			

$P_1$  checks bit positions 1, 3, 5, and 7.  
 $P_1 = 0$

$P_2$  checks bit positions 2, 3, 6, and 7.  
 $P_2 = 1$

$P_3$  checks bit positions 3, 5, 6, and 7.  
 $P_3 = 1$

The combined code is **0111100**.

**65.**  $d = 5$

$$\begin{aligned}2^p &\geq d + p + 1 \\2^4 &= 5 + 4 + 1 = 10 \\p &= 4 \\&\text{parity = odd}\end{aligned}$$

Bit Designation	$P_1$	$P_2$	$D_1$	$P_3$	$D_2$	$D_3$	$D_4$	$P_4$	$D_5$
Bit Position	1	2	3	4	5	6	7	8	9
Binary Position Number	0001	0010	0011	0100	0101	0110	0111	1000	1001
Data Bits ( $D_n$ )			1		1	0	0		1
Parity Bits ( $P_n$ )	0	0		0				0	

$P_1$  checks bit positions 1, 3, 5, 7, and 9.  
 $P_1 = 0$

$P_2$  checks bit positions 2, 3, 6, and 7.  
 $P_2 = 0$

$P_3$  checks bit positions 4, 5, 6, and 7.  
 $P_3 = 0$

$P_4$  checks bit positions 8 and 9.  
 $P_4 = 0$

The combined code is **001010001**.

## Chapter 4

**66.** (a) Even parity

	$P_1$ 001	$P_2$ 010	$D_1$ 011	$P_3$ 100	$D_2$ 101	$D_3$ 110	$D_4$ 111	Check result (0 good, 1 bad)
$P_1$ checks 1, 3, 5, 7	1 1	1 1	1 1	0 0	1 1	0 0	0 0	1 (LSB) 0 1
$P_2$ checks 2, 3, 6, 7	1	1	1	0	1	0	0	
$P_3$ checks 4, 5, 6, 7								

The error position code is 101. The corrected code is **1110000**.

(b) Even parity

	$P_1$ 001	$P_2$ 010	$D_1$ 011	$P_3$ 100	$D_2$ 101	$D_3$ 110	$D_4$ 111	Check result (0 good, 1 bad)
$P_1$ checks 1, 3, 5, 7	1 1	0 0	0 0	0 0	1 1	1 1	1 1	1 (LSB) 0 1
$P_2$ checks 2, 3, 6, 7	1	0	0	0	1	1	1	
$P_3$ checks 4, 5, 6, 7								

The error position code is 101. The corrected code is **1000011**.

**67.** (a) Odd parity

	$P_1$ 000	$P_2$ 001	$D_1$ 001	$P_3$ 010	$D_2$ 010	$D_3$ 011	$D_4$ 011	$P_4$ 100	$D_5$ 100	Check result (0 good, 1 bad)
$P_1$ checks 1, 3, 5, 7, 9	1 1	1 1	0 0	1 1	0 0	0 0	0 0	1 1	1 1	1 (LSB) 0 1
$P_2$ checks 2, 3, 6, 7	1	1	0	1	0	0	0	1	1	0
$P_3$ checks 4, 5, 6, 7	1	1								1
$P_4$ checks 8, 9										

The error position code is 1001. The corrected code is **110100010**.

(b) Odd parity

	$P_1$ 000	$P_2$ 001	$D_1$ 001	$P_3$ 010	$D_2$ 010	$D_3$ 011	$D_4$ 011	$P_4$ 100	$D_5$ 100	Check result (0 good, 1 bad)
$P_1$ checks 1, 3, 5, 7, 9	1 1	0 0	0 0	0 0	0 0	1 1	1 1	0 0	1 1	0 (LSB) 1 1
$P_2$ checks 2, 3, 6, 7	1	0	0	0	0	1	1	0	1	1
$P_3$ checks 4, 5, 6, 7	1	0	0							0
$P_4$ checks 8, 9										

The error position code is 0110. The corrected code is **100000101**.

---

## CHAPTER 3

### LOGIC GATES

---

#### Section 3-1 The Inverter

1. See Figure 3-1.

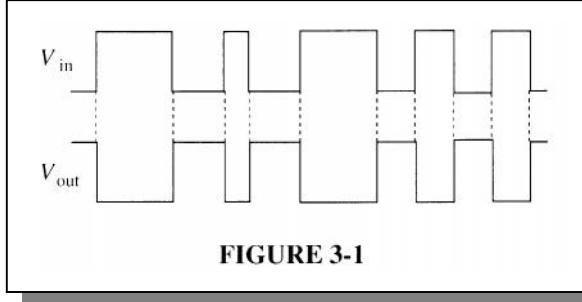


FIGURE 3-1

2. B: LOW, C: HIGH, D: LOW, E: HIGH, F: LOW

#### Section 3-2 The AND Gate

3. See Figure 3-2.

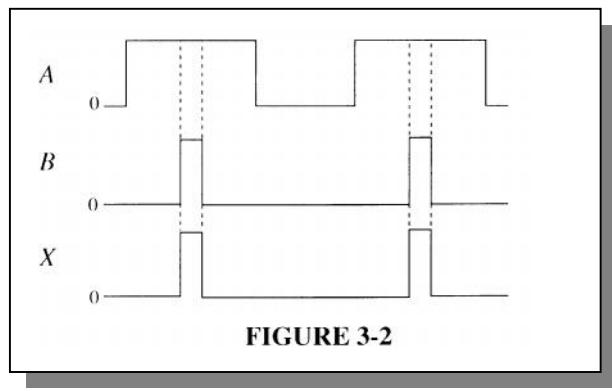


FIGURE 3-2

4. See Figure 3-3.

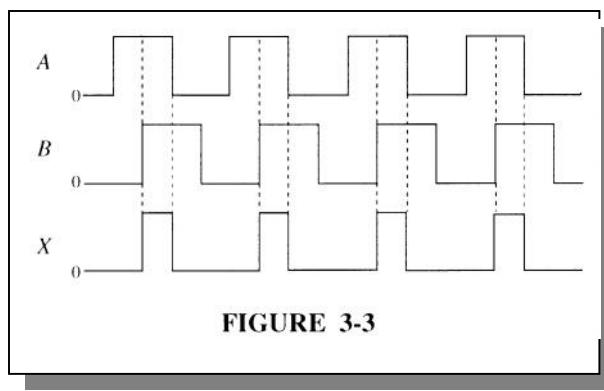
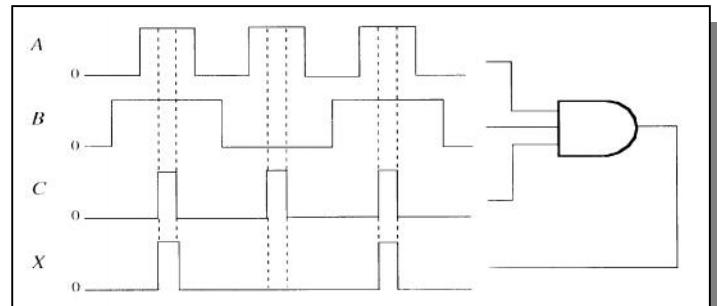


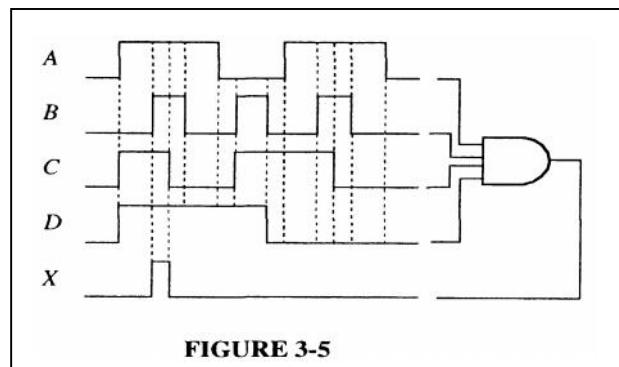
FIGURE 3-3

5. See Figure 3-4.



**FIGURE 3-4**

6. See Figure 3-5.



**FIGURE 3-5**

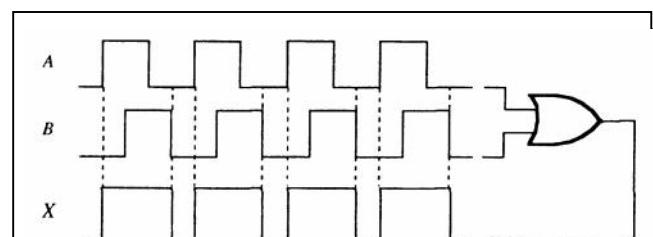
### **Section 3-3 The OR Gate**

7. See Figure 3-6.

A

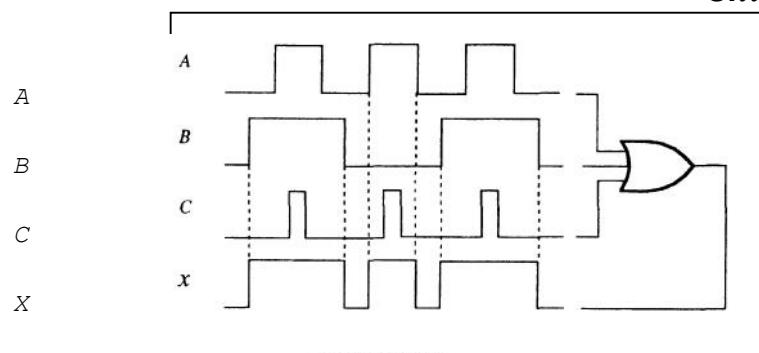
B

X



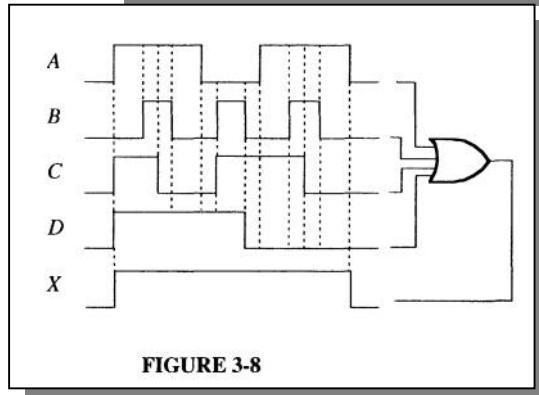
**FIGURE 3-6**

8. See Figure 3-7.



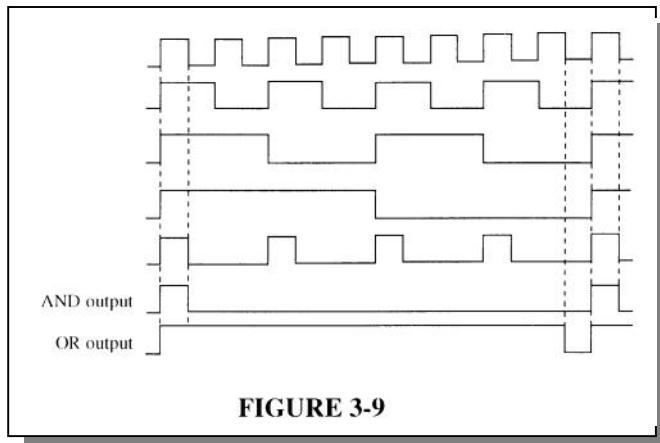
**FIGURE 3-7**

9. See Figure 3-8.



**FIGURE 3-8**

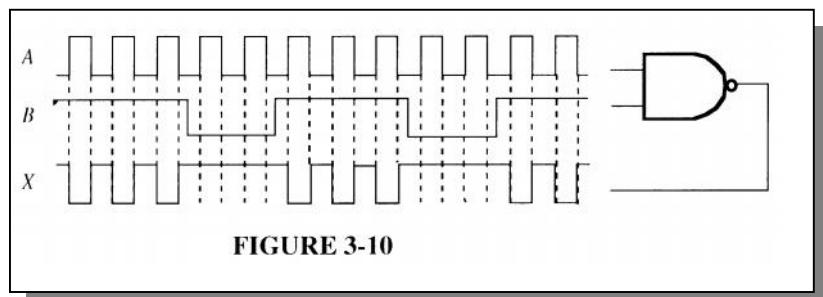
10. See Figure 3-9.



**FIGURE 3-9**

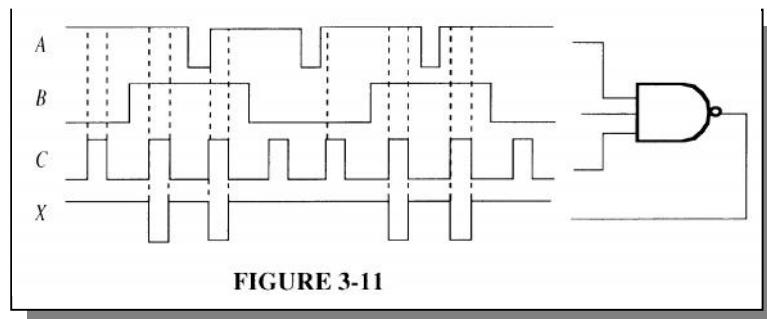
**Section 3-4 The NAND Gate**

11. See Figure 3-10.



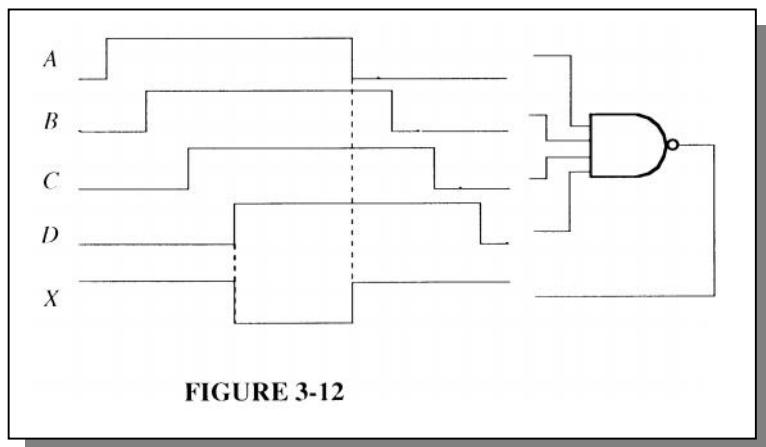
**FIGURE 3-10**

12. See Figure 3-11.



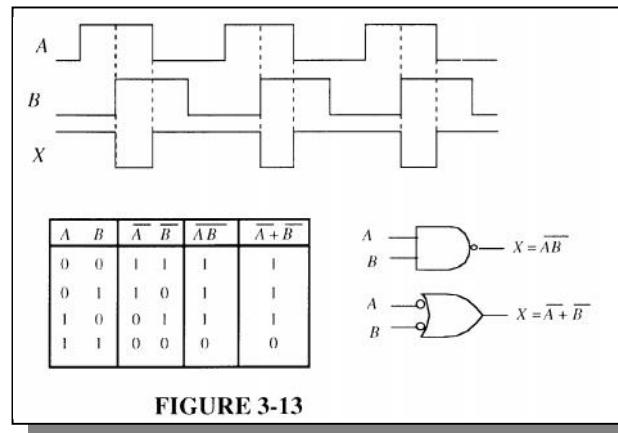
**FIGURE 3-11**

13. See Figure 3-12.



**FIGURE 3-12**

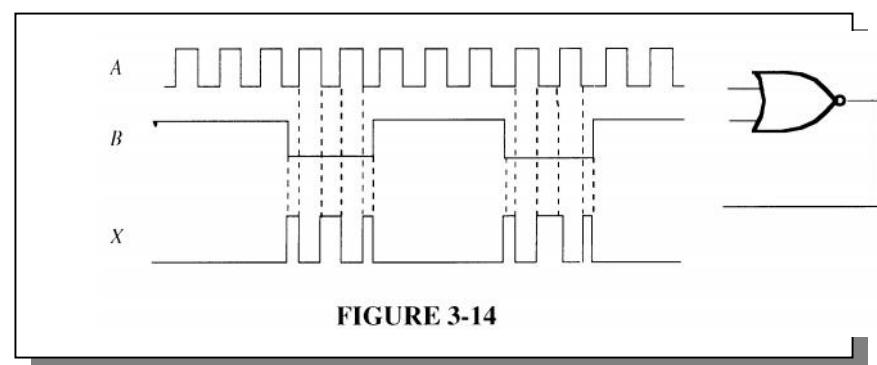
**14.** See Figure 3-13.



**FIGURE 3-13**

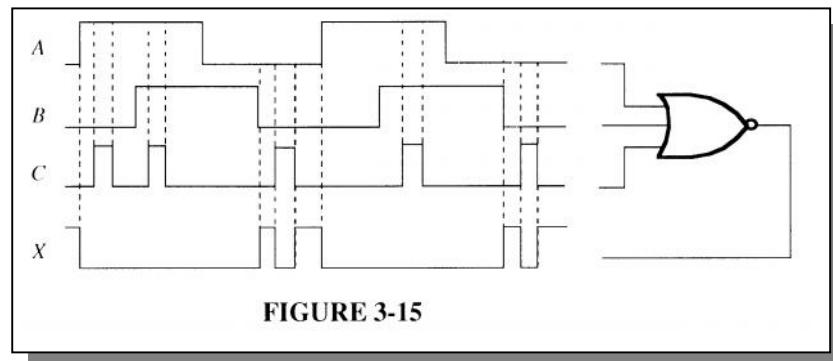
### **Section 3-5 The NOR Gate**

**15.** See Figure 3-14.



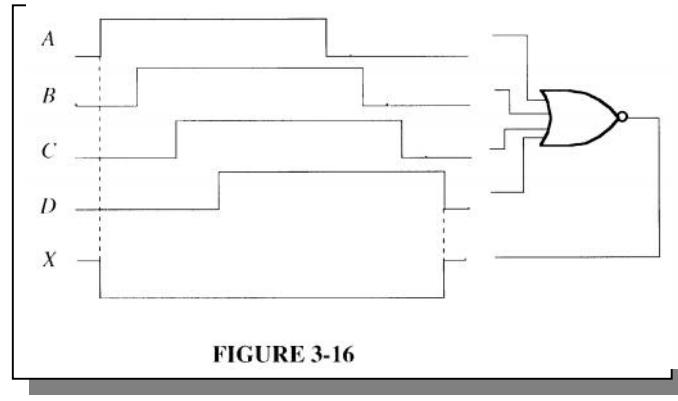
**FIGURE 3-14**

**16.** See Figure 3-15.

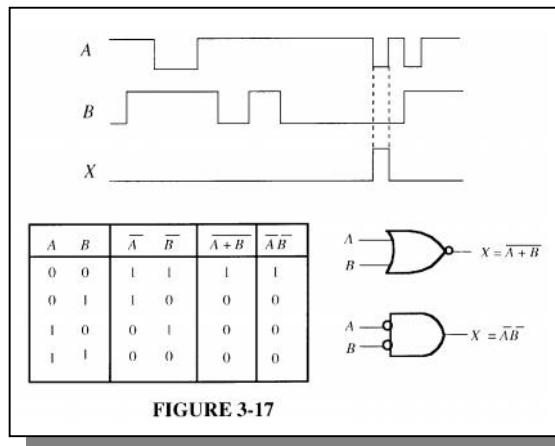


**FIGURE 3-15**

17. See Figure 3-16.



18. See Figure 3-17.



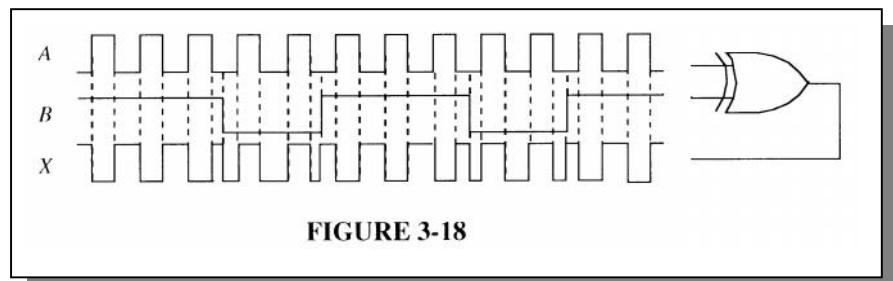
### Section 3-6 The Exclusive-OR and Exclusive-NOR Gates

19. The output of the XOR gate is HIGH only when one input is HIGH. The output of the OR gate is HIGH any time one or more inputs are HIGH.

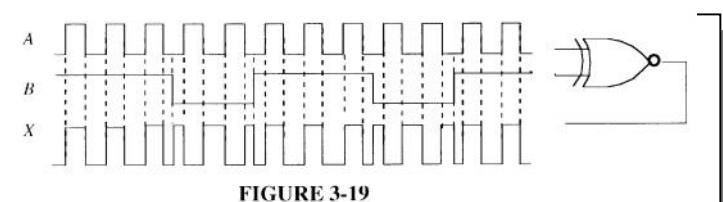
$$\text{XOR} = A\bar{B} + \bar{A}B$$

$$\text{OR} = A + B$$

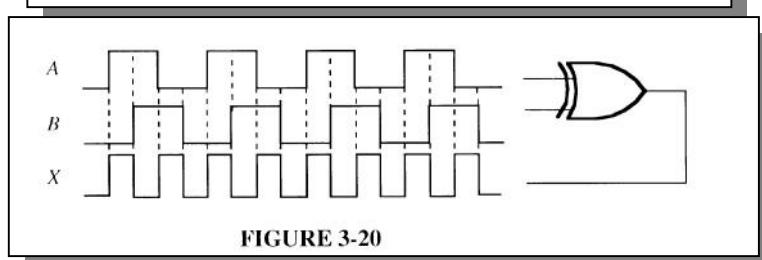
20. See Figure 3-18.



21. See Figure 3-19.



22. See Figure 3-20.



**Section 3-7 Programmable Logic**

23.  $X_1 = \overline{AB}$

$$X_2 = \overline{\overline{A}\overline{B}}$$

$$X_3 = \overline{A}\overline{B}$$

24.  $X_1 = \overline{ABC}$

Row 1: blow  $A, B, \overline{B}, C$ , and  $\overline{C}$  column fuses

Row 2: blow  $A, \overline{A}, \overline{B}, C$ , and  $\overline{C}$  column fuses

Row 3: blow  $A, \overline{A}, B, \overline{B}$ , and  $\overline{C}$  column fuses

$$X_2 = A\overline{B}\overline{C}$$

Row 4: blow  $\overline{A}, B, \overline{B}, C$ , and  $\overline{C}$  column fuses

Row 5: blow  $A, \overline{A}, \overline{B}, C$ , and  $\overline{C}$  column fuses

Row 6: blow  $A, \overline{A}, B, \overline{B}$ , and  $C$  column fuses

$$X_3 = \overline{A}\overline{B}\overline{C}$$

Row 7: blow  $A, B, \overline{B}, C$ , and  $\overline{C}$  column fuses

Row 8: blow  $A, \overline{A}, \overline{B}, C$ , and  $\overline{C}$  column fuses

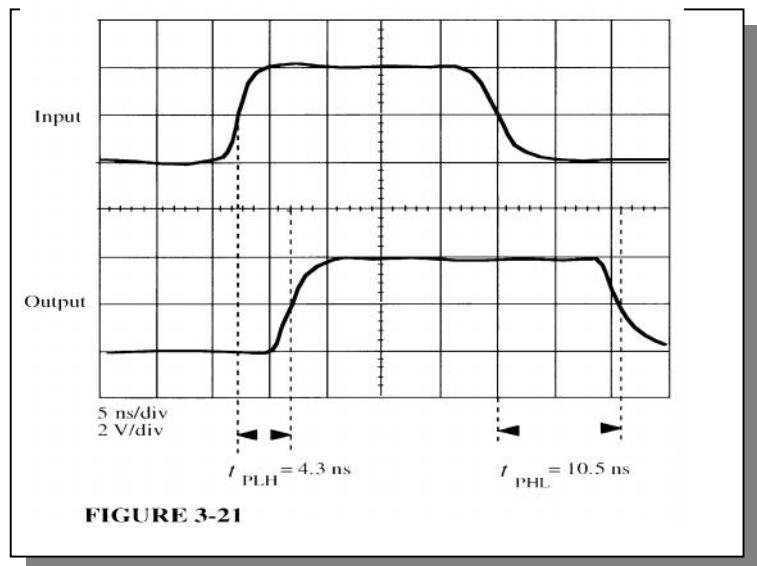
Row 9: blow  $A, \overline{A}, B, \overline{B}$ , and  $C$  column fuses

**Section 3-8 Fixed-Function Logic**

25. The power dissipation of **CMOS** increases with frequency.

26. (a)  $P = \left( \frac{I_{CCH} + I_{CCL}}{2} \right) V_{CC} = \left( \frac{1.6 \text{ mA} + 4.4 \text{ mA}}{2} \right) 5.5 \text{ V} = 16.5 \text{ mW}$   
 (b)  $V_{OH(\min)} = 2.7 \text{ V}$   
 (c)  $t_{PLH} = T_{PHL} = 15 \text{ ns}$   
 (d)  $V_{OL} = 0.4 \text{ V} (\text{max})$   
 (e) @  $V_{CC} = 2 \text{ V}$ ,  $t_{PLH} = t_{PHL} = 75 \text{ ns}$ ; @  $V_{CC} = 6 \text{ V}$ ,  $t_{PHL} = t_{PLH} = 13 \text{ ns}$

27. See Figure 3-21.



28. Gate A can be operated at the highest frequency because it has shorter propagation delay times than Gate B.

29.  $P_D = V_{CC} I_C = (5 \text{ V})(4 \text{ mA}) = 20 \text{ mW}$

30.  $I_{CCH} = 4 \text{ mA}; P_D = (5 \text{ V})(4 \text{ mA}) = 20 \text{ mW}$

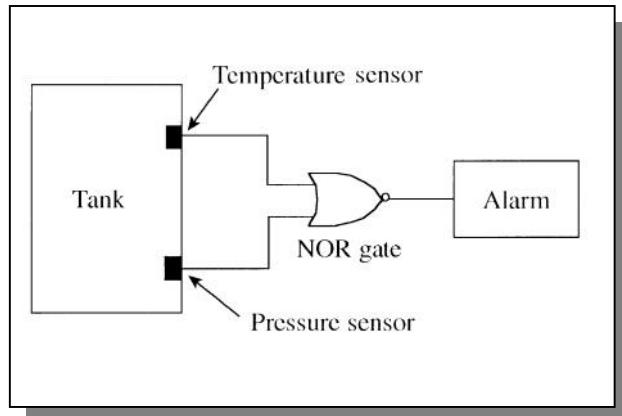
### **Section 3-9 Troubleshooting**

31. (a) NAND gate OK  
 (b) AND gate faulty  
 (c) NAND gate faulty  
 (d) NOR gate OK  
 (e) XOR gate faulty  
 (f) XOR gate OK
32. (a) NAND gate faulty. Input A open.  
 (b) NOR gate faulty. Input B shorted to ground.  
 (c) NAND gate OK  
 (d) XOR gate faulty. Input A open.
33. (a) The gate does not respond to pulses on either input when the other input is HIGH. It is unlikely that both inputs are open. The most probable fault is that the output is stuck in the LOW state (shorted to ground, perhaps) although it could be open.  
 (b) Pin 4 input or pin 6 output internally open.
34. The timer input to the AND gate is open. Check for 30-second HIGH level on this input when ignition is turned on.
35. An open seat-belt input to the AND gate will act like a constant HIGH just as if the seat belt were unbuckled.

**36.** Two possibilities: An input stuck LOW or the output stuck HIGH.

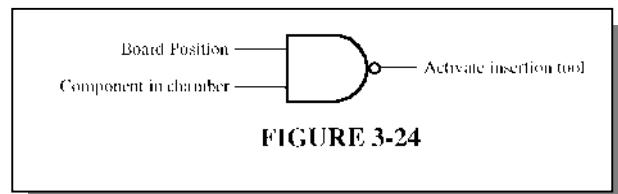
### **Special Design Problems**

**37.** See Figure 3-22.



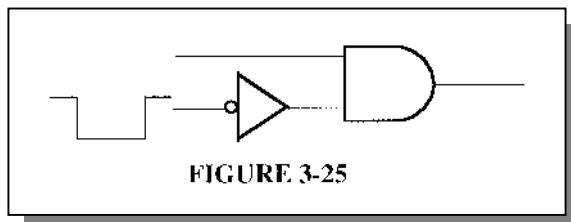
**FIGURE 3-22**

**38.** See Figure 3-23.



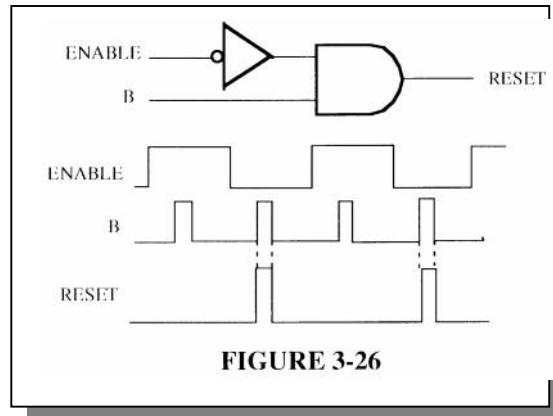
**FIGURE 3-23**

**39.** Add an inverter to the Enable input line of the AND gate as shown in Figure 3-24.

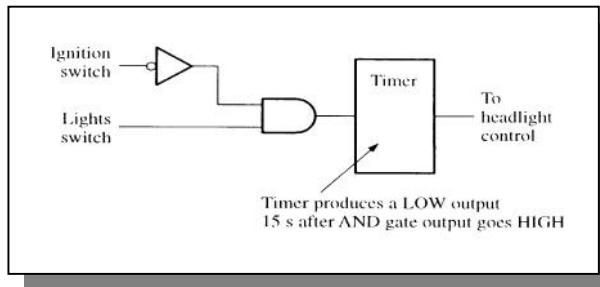


**FIGURE 3-25**

**40.** See Figure 3-25.

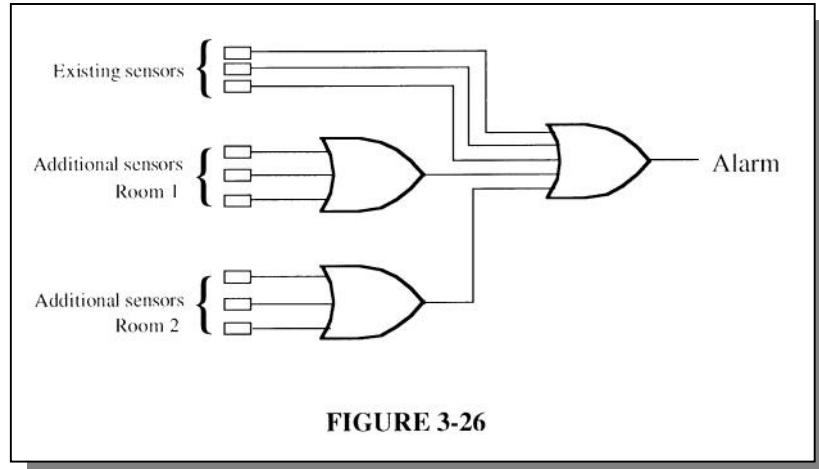


**41.** See Figure 3-26.



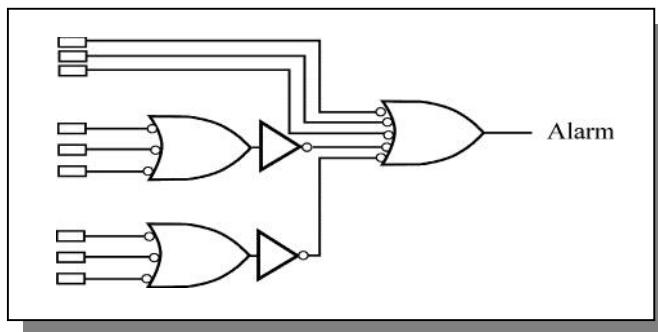
**42.** See Figure 3-27.

**FIGURE 3-**



**43.** See Figure 3-28.

**FIGURE 3-**



***Multisim Troubleshooting Practice***

- 44.** Input A shorted to output.
- 45.** Inputs shorted together.
- 46.** No fault.
- 47.** Output open.

---

## CHAPTER 4

### BOOLEAN ALGEBRA AND LOGIC SIMPLIFICATION

---

#### Section 4-1 Boolean Operations and Expressions

1.  $X = A + B + C + D$

This is an OR configuration.

2.  $Y = ABCDE$

3.  $X = \overline{A} + \overline{B} + \overline{C}$

4. (a)  $0 + 0 + 1 = 1$

(b)  $1 + 1 + 1 = 1$

(c)  $1 \cdot 0 \cdot 0 = 1$

(d)  $1 \cdot 1 \cdot 1 = 1$

(e)  $1 \cdot 0 \cdot 1 = 0$

(f)  $1 \cdot 1 + 0 \cdot 1 \cdot 1 = 1 + 0 = 1$

5. (a)  $AB = 1$  when  $A = 1, B = 1$

(b)  $\overline{ABC} = 1$  when  $A = 1, B = 0, C = 1$

(c)  $A + B = 0$  when  $A = 0, B = 0$

(d)  $\overline{A+B+C} = 0$  when  $A = 1, B = 0, C = 1$

(e)  $\overline{A+B+C} = 0$  when  $A = 1, B = 1, C = 0$

(f)  $\overline{A+B} = 0$  when  $A = 1, B = 0$

(g)  $\overline{ABC} = 1$  when  $A = 1, B = 0, C = 0$

6. (a)  $X = (A + B)C + B$

A	B	C	$A + B$	$(A + B)C$	X
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	1	0	1
0	1	1	1	1	1
1	0	0	1	0	0
1	0	1	1	1	1
1	1	0	1	0	1
1	1	1	1	1	1

(b)  $X = (\overline{A+B})C$

A	B	C	$\overline{A+B}$	X
0	0	0	1	0
0	0	1	1	1
0	1	0	0	0
0	1	1	0	0
1	0	0	0	0
1	0	1	0	0
1	1	0	0	0
1	1	1	0	0

(c)  $X = \overline{ABC} + AB$

A	B	C	$\overline{ABC}$	$AB$	X
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	0	0	0
0	1	1	0	0	0
1	0	0	0	0	0
1	0	1	0	0	0
1	1	0	0	0	0
1	1	1	0	0	0

1	0	0	0	0	0
1	0	1	1	0	1
1	1	0	0	1	1
1	1	1	0	1	1

$$(d) \quad X = (A + B)(\bar{A} + B)$$

A	B	$A + B$	$\bar{A} + B$	X
0	0	0	1	0
0	1	1	1	1
1	0	1	0	0
1	1	1	1	1

$$(e) \quad X = (A + BC)(\bar{B} + \bar{C})$$

A	B	C	$A + BC$	$\bar{B} + \bar{C}$	X
0	0	0	0	1	0
0	0	1	0	1	0
0	1	0	0	1	0
0	1	1	1	0	0
1	0	0	1	1	1
1	0	1	1	1	1
1	1	0	1	1	1
1	1	1	1	0	0

### **Section 4-2 Laws and Rules of Boolean Algebra**

7. (a) Commutative law of addition  
 (b) Commutative law of multiplication  
 (c) Distributive law
8. Refer to Table 4-1 in the textbook.

- (a) Rule 9:  $\overline{\overline{A}} = A$   
 (b) Rule 8:  $\overline{AA} = 0$  (applied to 1st and 3rd terms)  
 (c) Rule 5:  $\overline{A + A} = A$   
 (d) Rule 6:  $\overline{A + \overline{A}} = 1$   
 (e) Rule 10:  $\overline{A + AB} = A$   
 (f) Rule 11:  $\overline{A + \overline{AB}} = A + B$  (applied to 1st and 3rd terms)

### **Section 4-3 DeMorgan's Theorems**

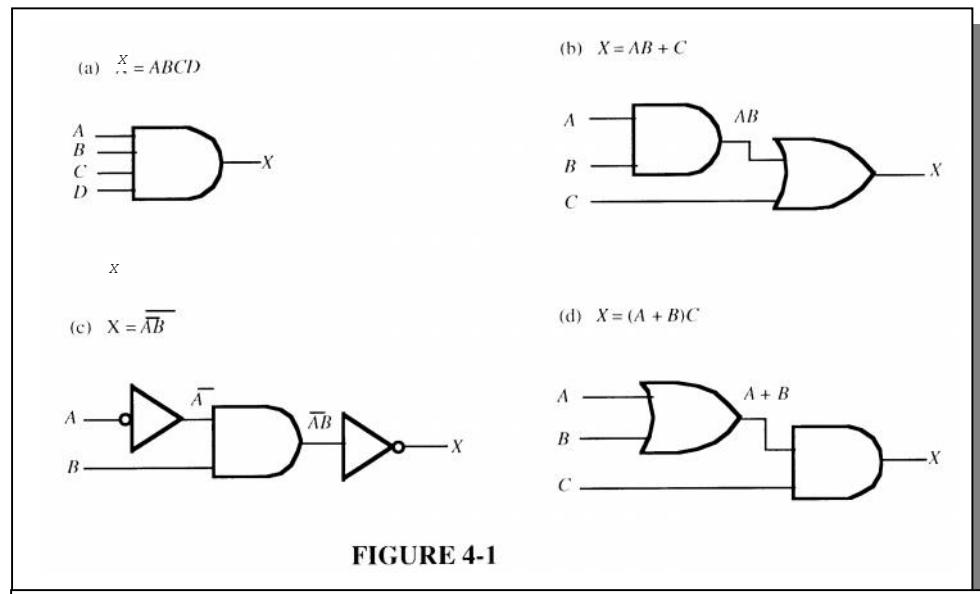
9. (a)  $\overline{A + \overline{B}} = \overline{\overline{A} \overline{B}} = \overline{A}\overline{\overline{B}}$   
 (b)  $\overline{\overline{A} + \overline{B}} = \overline{A} + \overline{\overline{B}} = \overline{A} + B$   
 (c)  $\overline{\overline{A + B + C}} = \overline{\overline{A}\overline{B}\overline{C}}$   
 (d)  $\overline{\overline{ABC}} = \overline{A} + \overline{B} + \overline{C}$   
 (e)  $\overline{A(B + C)} = \overline{A} + \overline{(B + C)} = \overline{A} + \overline{B}\overline{C}$   
 (f)  $\overline{\overline{AB} + \overline{CD}} = \overline{\overline{A} + \overline{B} + \overline{C} + \overline{D}}$   
 (g)  $\overline{AB + CD} = (\overline{AB})(\overline{CD}) = (\overline{A} + \overline{B})(\overline{C} + \overline{D})$   
 (h)  $\overline{(A + \overline{B})(\overline{C} + D)} = \overline{A + \overline{B}} + \overline{\overline{C} + D} = \overline{A}\overline{B} + \overline{C}\overline{D}$

10. (a)  $\overline{AB}(C + \overline{D}) = \overline{AB} + \overline{(C + \overline{D})} = \overline{A} + \overline{B} + \overline{CD}$
- (b)  $\begin{aligned} \overline{AB(CD + EF)} &= \overline{AB} + \overline{(CD + EF)} = \overline{A} + \overline{B} + \overline{(CD)(EF)} \\ &= \overline{A} + \overline{B} + (\overline{C} + \overline{D})(\overline{E} + \overline{F}) \end{aligned}$
- (c)  $\overline{(A + \overline{B} + C + \overline{D})} + \overline{AB\overline{CD}} = \overline{AB}\overline{CD} + \overline{A} + \overline{B} + \overline{C} + \overline{D}$
- (d)  $\begin{aligned} \overline{(\overline{A} + B + C + D)(\overline{AB}\overline{CD})} &= (\overline{AB}\overline{CD})(\overline{A} + B + C + \overline{D}) \\ &= \overline{\overline{AB}\overline{CD}} + \overline{\overline{A} + B + C + \overline{D}} = \overline{A} + \overline{B} + \overline{C} + \overline{D} + \overline{AB}\overline{CD} \end{aligned}$
- (e)  $\begin{aligned} \overline{AB(CD + \overline{EF})(\overline{AB} + \overline{CD})} &= \overline{AB} + (\overline{CD} + \overline{EF}) + (\overline{AB} + \overline{CD}) \\ &= AB + (\overline{CD})(\overline{EF}) + (\overline{AB})(\overline{CD}) \\ &= AB + (\overline{C} + \overline{D})(E + \overline{F}) + ABCD \end{aligned}$
11. (a)  $\begin{aligned} \overline{\overline{(ABC)(EFG)} + \overline{(HIJ)(KLM)}} &= \overline{\overline{ABC}} + \overline{\overline{EFG}} + \overline{\overline{HIJ}} + \overline{\overline{KLM}} \\ &= \overline{ABC} + EFG + HIJ + KLM = (\overline{ABC})(\overline{EFG})(\overline{HIJ})(\overline{KLM}) \\ &= (\overline{A} + \overline{B} + \overline{C})(\overline{E} + \overline{F} + \overline{G})(\overline{H} + \overline{I} + \overline{J})(\overline{K} + \overline{L} + \overline{M}) \end{aligned}$
- (b)  $\begin{aligned} \overline{(A + \overline{BC} + CD) + \overline{BC}} &= \overline{A}(\overline{BC})(\overline{CD}) + BC = \overline{A}(\overline{BC})(\overline{CD}) + BC \\ &= \overline{ABC}(\overline{C} + \overline{D}) + BC = \overline{ABC} + \overline{ABC}\overline{D} + BC = \overline{ABC}(1 + \overline{D}) + BC \\ &= \overline{ABC} + BC \end{aligned}$
- (c)  $\begin{aligned} \overline{\overline{(A + B)(C + D)(E + F)(G + H)}} &= \overline{(A + B)}\overline{(C + D)}\overline{(E + F)}\overline{(G + H)} = \overline{ABCDEFGH} \end{aligned}$

**Section 4-4 Boolean Analysis of Logic Circuits**

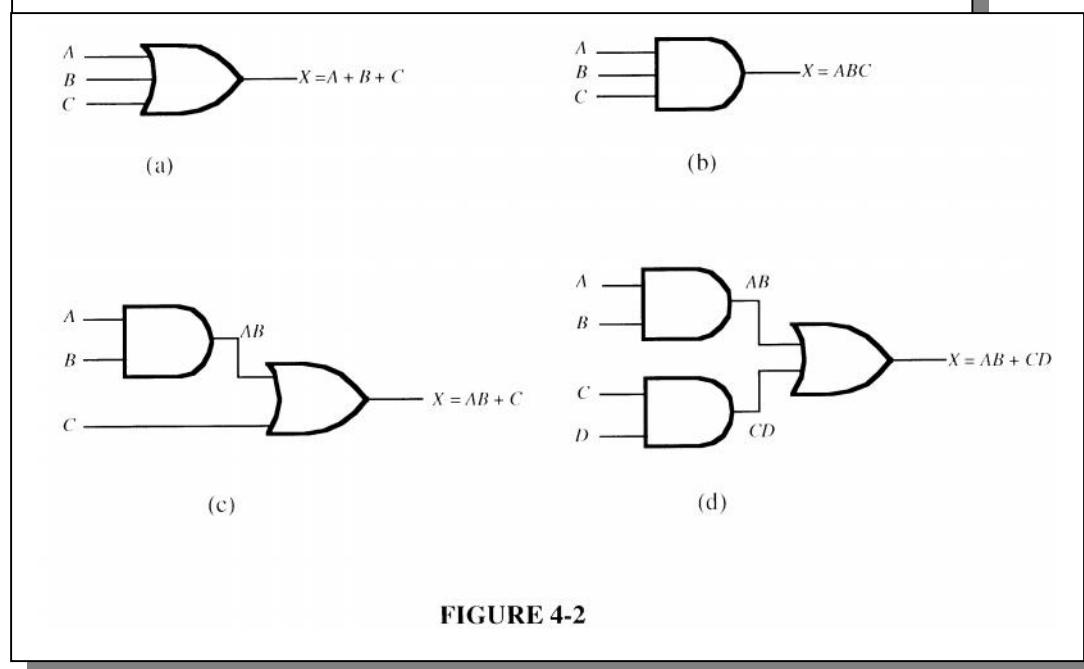
12. (a)  $AB = X$   
 (b)  $\overline{A} = X$   
 (c)  $A + B = X$   
 (d)  $A + B + C = X$

13. See Figure 4-1.



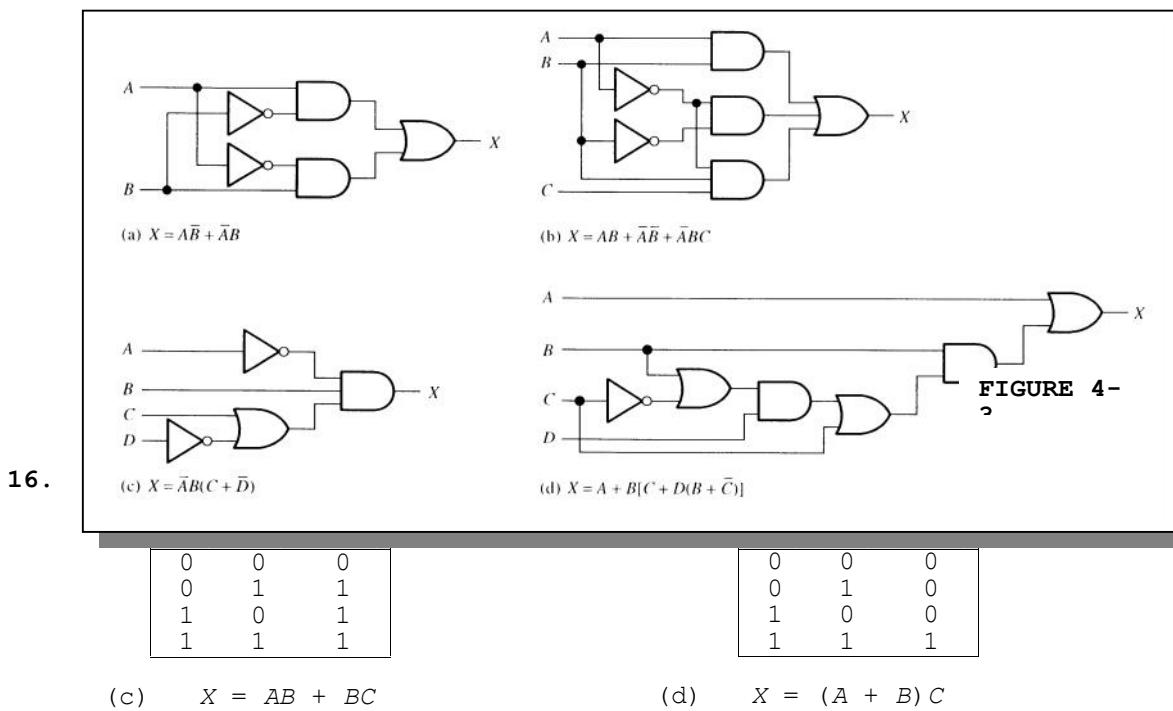
14.

**FIGURE 4-1**



**FIGURE 4-2**

15. See Figure 4-3.



16.

0	0	0
0	1	1
1	0	1
1	1	1

$$(c) \quad X = AB + BC$$

0	0	0
0	1	0
1	0	0
1	1	1

$$(d) \quad X = (A + B)C$$

A	B	C	X
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

A	B	C	X
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

$$(e) \quad X = (A+B)(\bar{B}+C)$$

A	B	C	$A + B$	$\bar{B} + C$	X
0	0	0	0	1	0
0	0	1	0	1	0
0	1	0	1	0	0
0	1	1	1	1	1
1	0	0	1	1	1
1	0	1	1	1	1
1	1	0	1	0	0
1	1	1	1	1	1

### Section 4-5 Simplification Using Boolean Algebra

$$17. \quad (a) \quad A(A+B) = AA + BB = A + AB = A(1 + B) = A$$

$$(b) \quad A(\bar{A} + AB) = A\bar{A} + AAB = 0 + AB = AB$$

$$(c) \quad BC + \bar{B}C = C(B + \bar{B}) = C(1) = C$$

$$(d) \quad A(A + \bar{A}B) = AA + A\bar{A}B = A + (0)B = A + 0 = A$$

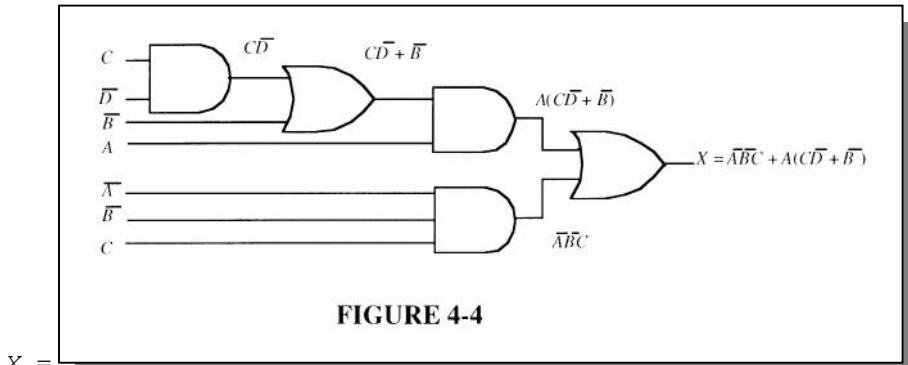
$$\begin{aligned}
 (e) \quad & A\bar{B}C + \bar{A}\bar{B}C + \bar{A}\bar{B}\bar{C} = A\bar{B}C + \bar{A}C(B + \bar{B}) = A\bar{B}C + \bar{A}C(1) \\
 & = A\bar{B}C + \bar{A}C = C(\bar{A} + A\bar{B}) = C(\bar{A} + \bar{B}) = \bar{A}\bar{C} + \bar{B}\bar{C}
 \end{aligned}$$

$$\begin{aligned}
 18. \quad (a) \quad & (A + \bar{B})(A + C) = AA + AC + A\bar{B} + \bar{B}C = A + AC + A\bar{B} + \bar{B}C \\
 & = A(1 + C + \bar{B}) + \bar{B}C = A(1) + \bar{B}C = \mathbf{A} + \bar{B}\mathbf{C} \\
 (b) \quad & \bar{A}\bar{B} + \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}CD + \bar{A}\bar{B}\bar{C}DE = \bar{A}\bar{B}(1 + \bar{C} + CD + \bar{C}DE) = \bar{A}\bar{B}(1) \\
 & = \bar{A}\bar{B} \\
 (c) \quad & AB + \bar{A}\bar{B}C + A = AB + (\bar{A} + \bar{B})C + A = AB + \bar{A}C + \bar{B}C + A \\
 & A(B + 1) + \bar{A}C + \bar{B}C = A + \bar{A}C + \bar{B}C = A + C + \bar{B}C = A + C(1 + \bar{B}) \\
 & = \mathbf{A} + \mathbf{C} \\
 (d) \quad & (A + \bar{A})(AB + A\bar{B}\bar{C}) = AAB + AAB\bar{C} + \bar{A}AB + \bar{A}AB\bar{C} \\
 & = AB + A\bar{B}\bar{C} + 0 + 0 = AB(1 + \bar{C}) = \mathbf{AB} \\
 (e) \quad & AB + (\bar{A} + \bar{B})C + AB = AB + \bar{A}C + \bar{B}C + AB = AB + (\bar{A} + \bar{B})C \\
 & = AB + \bar{A}\bar{B}C = \mathbf{AB} + \mathbf{C}
 \end{aligned}$$

$$\begin{aligned}
 19. \quad (a) \quad & BD + B(D + E) + \bar{D}(D + F) = BD + BD + BE + \bar{D}D + \bar{D}F \\
 & = BD + BE + 0 + \bar{D}F = \mathbf{BD} + \mathbf{BE} + \bar{\mathbf{DF}} \\
 (b) \quad & \bar{A}\bar{B}\bar{C} + \overline{(A + B + \bar{C})} + \bar{A}\bar{B}\bar{C}D = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}\bar{C}D = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}\bar{C}D \\
 & = \bar{A}\bar{B}(C + \bar{C}D) = \bar{A}\bar{B}(C + D) = \bar{A}\bar{B}\mathbf{C} + \bar{A}\bar{B}\mathbf{D} \\
 (c) \quad & (B + BC)(B + \bar{B}C)(B + D) = B(1 + C)(B + C)(B + D) \\
 & = B(B + C)(B + D) = (BB + BC)(B + D) = (B + BC)(B + D) \\
 & = B(1 + C)(B + D) = B(B + D) = BB + BD = B + BD = B(1 + D) = \mathbf{B} \\
 (d) \quad & ABCD + AB(\bar{C}D) + (\bar{A}\bar{B})CD = ABCD + AB(\bar{C} + \bar{D}) + (\bar{A} + \bar{B})CD \\
 & = ABCD + AB\bar{C} + AB\bar{D} + \bar{A}CD + \bar{B}CD \\
 & = CD(AB + \bar{A} + \bar{B}) + AB\bar{C} + AB\bar{D} = CD(B + \bar{A} + \bar{B}) + AB\bar{C} + AB\bar{D} \\
 & = CD(1 + \bar{A}) + AB\bar{C} + AB\bar{D} = CD + AB\bar{C} + AB\bar{D} = CD + AB(\bar{C}D) = \mathbf{CD} + \mathbf{AB} \\
 (e) \quad & ABC[AB + \bar{C}(BC + AC)] = ABABC + ABC\bar{C}(BC + AC) \\
 & = ABC + 0(BC + AC) = \mathbf{ABC}
 \end{aligned}$$

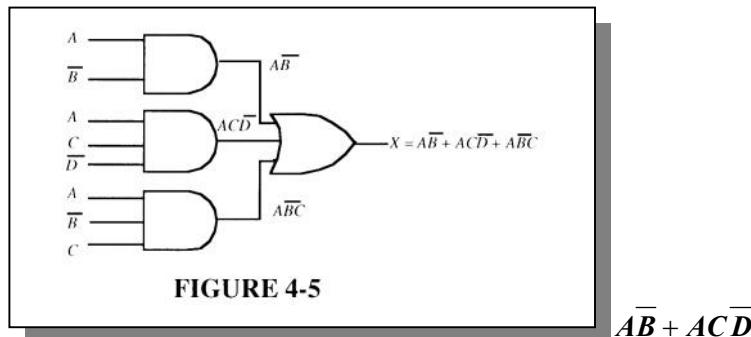
**20.** First develop the Boolean expression for the output of each gate network and simplify.

(a) See Figure 4-4.



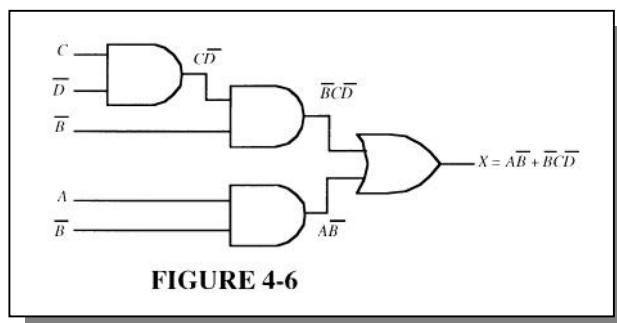
$$X = \overline{B}(A+C) + ACD = \overline{AB} + \overline{BC} + A\overline{CD}$$

(b) See Figure 4-5.



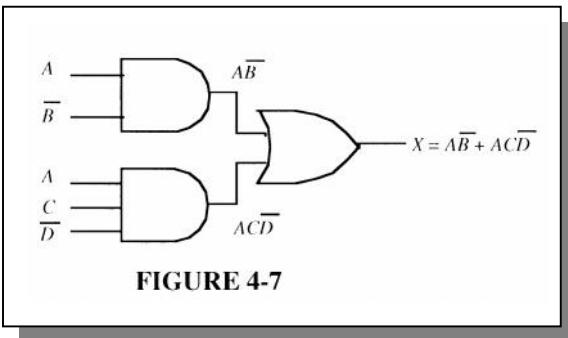
$$\overline{AB} + A\overline{CD}$$

(c) See Figure 4-6.



tion is possible.

(d) See Figure 4-7.



$$X = A\bar{B} + A\bar{C}\bar{D} \quad \text{No further simplification is possible.}$$

### **Section 4-6 Standard Forms of Boolean Expressions**

21. (a)  $(A+B)(C+\bar{B}) = AC + BC + B\bar{B} + A\bar{B} = AC + BC + A\bar{B}$   
 (b)  $(A + \bar{B}C)C = AC + \bar{B}CC = AC + \bar{B}C$   
 (c)  $(A + C)(AB + AC) = AAB + AAC + ABC + ACC = AB + AC + ABC + ACC$   
 $= (AB + AC)(1 + C) = AB + AC$
22. (a)  $AB + CD(\bar{A}\bar{B} + CD) = AB + A\bar{B}CD + CD\bar{C}D = AB + A\bar{B}CD + CD$   
 $= AB(\bar{A}\bar{B} + 1)CD = AB + CD$   
 (b)  $AB(\bar{B}\bar{C} + BD) = A\bar{B}\bar{B}\bar{C} + ABBD = 0 + ABD = ABD$   
 (c)  $A + B[AC + (B + \bar{C})D] = A + ABC + (B + \bar{C})BD$   
 $= A + ABC + BD + B\bar{C}D = A(1 + BC) + BD + B\bar{C}D = A + BD(1 + \bar{C})$   
 $= A + BD$
23. (a) The domain is  $A, B, C$   
 The standard SOP is:  $\bar{A}\bar{B}C + \bar{A}\bar{B}\bar{C} + ABC + \bar{A}BC$   
 (b) The domain is  $A, B, C$   
 The standard SOP is:  $ABC + \bar{A}\bar{B}C + \bar{A}\bar{B}\bar{C}$   
 (c) The domain is  $A, B, C$   
 The standard SOP is:  $ABC + A\bar{B}\bar{C} + \bar{A}\bar{B}C$
24. (a)  $AB + CD = ABCD + ABC\bar{D} + A\bar{B}CD + A\bar{B}\bar{C}D + \bar{A}\bar{B}CD + \bar{A}\bar{B}\bar{C}D$   
 (b)  $ABD = ABCD + A\bar{B}\bar{C}D$   
 (c)  $A + BD = \bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}\bar{C}D + \bar{A}\bar{B}C\bar{D} + \bar{A}\bar{B}CD + A\bar{B}\bar{C}\bar{D} + A\bar{B}\bar{C}D + A\bar{B}C\bar{D} + A\bar{B}CD$
25. (a)  $\bar{A}\bar{B}C + \bar{A}\bar{B}\bar{C} + ABC + \bar{A}BC$ : 101 + 100 + 111 + 011  
 (b)  $ABC + \bar{A}\bar{B}C + \bar{A}\bar{B}\bar{C}$ : 111 + 101 + 001  
 (c)  $ABC + A\bar{B}\bar{C} + \bar{A}\bar{B}C$ : 111 + 110 + 101
26. (a)  $ABCD + ABC\bar{D} + A\bar{B}CD + A\bar{B}\bar{C}D + \bar{A}\bar{B}CD + \bar{A}\bar{B}\bar{C}D + \bar{A}\bar{B}CD$ :  
 $1111 + 1110 + 1101 + 1100 + 0011 + 0111 + 1011$   
 (b)  $ABCD + A\bar{B}CD$ : 1111 + 1101  
 (c)  $\bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}\bar{C}D + \bar{A}\bar{B}C\bar{D} + A\bar{B}\bar{C}\bar{D} + A\bar{B}\bar{C}D + A\bar{B}C\bar{D} + A\bar{B}CD$ :  
 $1000 + 1001 + 1010 + 1011 + 1100 + 1101 + 1110 + 1111 + 0101 + 0111$
27. (a)  $(A + B + C)(A + B + \bar{C})(A + \bar{B} + C)(\bar{A} + \bar{B} + C)$   
 (b)  $(A + B + C)(A + \bar{B} + C)(A + \bar{B} + \bar{C})(\bar{A} + B + C)(\bar{A} + \bar{B} + C)$   
 (c)  $(A + B + C)(A + B + \bar{C})(A + \bar{B} + C)(A + \bar{B} + \bar{C})(\bar{A} + B + C)$
28. (a)  $(A + B + C + D)(A + B + C + \bar{D})(A + B + \bar{C} + D)(A + \bar{B} + C + D)(A + \bar{B} + C + \bar{D})$

$$\begin{aligned}
 & (A + \bar{B} + \bar{C} + D)(\bar{A} + B + C + D)(\bar{A} + B + C + \bar{D})(\bar{A} + B + \bar{C} + D) \\
 (\text{b}) \quad & (A + B + C + D)(A + B + C + \bar{D})(A + B + \bar{C} + D)(A + B + \bar{C} + \bar{D}) \\
 & (A + \bar{B} + C + D)(A + \bar{B} + C + \bar{D})(A + \bar{B} + \bar{C} + D)(A + \bar{B} + \bar{C} + \bar{D})(\bar{A} + B + C + D) \\
 & (\bar{A} + B + C + \bar{D})(\bar{A} + B + \bar{C} + D)(\bar{A} + B + \bar{C} + \bar{D})(\bar{A} + \bar{B} + C + D)(\bar{A} + \bar{B} + \bar{C} + D) \\
 (\text{c}) \quad & (A + B + C + D)(A + B + C + \bar{D})(A + B + \bar{C} + D)(A + B + \bar{C} + \bar{D}) \\
 & (A + \bar{B} + C + D)(A + \bar{B} + \bar{C} + D)
 \end{aligned}$$

**Section 4-7 Boolean Expressions and Truth Tables**

29. (a)

A	B	C	X
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

(b)

X	Y	Z	Q
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

30. (a)

A	B	C	D	X
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

(b)

W	X	Y	Z	Q
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

31. (a)  $\overline{AB} + ABC\bar{C} + \overline{AC} + A\bar{B}\bar{C} = \overline{ABC} + \overline{ABC} + A\bar{B}\bar{C} + \overline{ABC} + A\bar{B}\bar{C}$

(b) 
$$\begin{aligned} \overline{X} + Y\bar{Z} + WZ + X\bar{Y}Z &= \overline{WXY\bar{Z}} + \overline{WX\bar{Y}Z} + \overline{W\bar{X}YZ} + \overline{W\bar{X}\bar{Y}Z} \\ &+ \overline{WX\bar{Y}Z} + \overline{WXY\bar{Z}} + W\overline{XYZ} + W\overline{X\bar{Y}Z} \\ &+ W\overline{X\bar{Y}Z} + W\overline{XYZ} + W\overline{X\bar{Y}Z} + WXYZ \end{aligned}$$

A	B	C	X
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

W	X	Y	Z	Q
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

**32.** (a)

A	B	C	X
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

(b)

A	B	C	D	X
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

**33.** (a)

A	B	C	X
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

(b)

A	B	C	D	X
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

**34.** (a)  $X = \overline{ABC} + A\overline{BC} + A\overline{B}C + ABC$

$$X = (A + B + C)(A + \overline{B} + C)(A + \overline{B} + \overline{C})(\overline{A} + \overline{B} + C)$$

(b)  $X = A\overline{B}\overline{C} + A\overline{B}C + ABC$

$$X = (A + B + C)(A + B + \overline{C})(A + \overline{B} + C)(A + \overline{B} + \overline{C})(\overline{A} + B + C)$$

(c)  $X = \overline{ABCD} + \overline{ABC}\overline{D} + \overline{ABC}D + \overline{AB}\overline{C}D + \overline{ABC}\overline{D} + \overline{AB}\overline{C}\overline{D} + ABCD$

$$X = (A + B + \overline{C} + D)(A + \overline{B} + C + D)(A + \overline{B} + \overline{C} + \overline{D})(\overline{A} + B + C + D)(\overline{A} + B + \overline{C} + D) \\ (\overline{A} + B + \overline{C} + \overline{D})(\overline{A} + \overline{B} + C + \overline{D})(\overline{A} + \overline{B} + \overline{C} + D)(\overline{A} + \overline{B} + \overline{C} + \overline{D})$$

(d)  $X = \overline{\overline{ABC}\overline{D}} + \overline{ABC}\overline{D} + \overline{ABC}D + \overline{AB}\overline{C}\overline{D} + \overline{AB}\overline{C}D + \overline{AB}\overline{C}\overline{D} + ABCD$

$$X = (A + B + C + D)(A + B + C + \overline{D})(A + B + \overline{C} + \overline{D})(A + \overline{B} + \overline{C} + D)(\overline{A} + B + C + D)(\overline{A} + B + \overline{C} + D) \\ (\overline{A} + B + C + \overline{D})(\overline{A} + B + \overline{C} + D)(\overline{A} + \overline{B} + C + \overline{D})(\overline{A} + \overline{B} + \overline{C} + D)$$

### Section 4-8 The Karnaugh Map

**35.** See Figure 4-8.

**36.** See Figure 4-9.

37. See Figure 4-10.

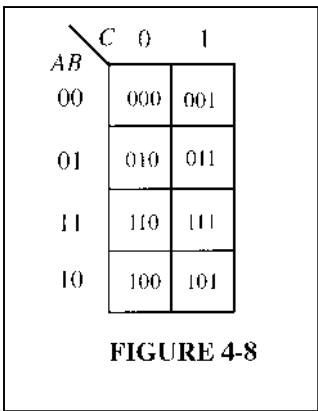


FIGURE 4-8

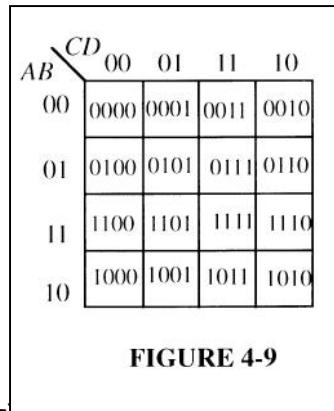


FIGURE 4-9

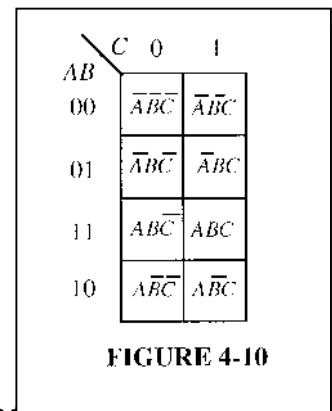
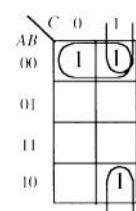


FIGURE 4-10

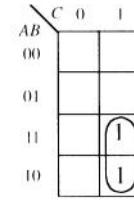
38. See Figure 4-

$$(a) X = \overline{A} \overline{B} \overline{C} + \overline{A} \overline{B} C + A \overline{B} C$$



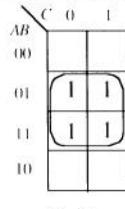
$$X = \overline{A} \overline{B} + \overline{B} C$$

$$(b) X = AC(\overline{B} + C) = AC + A\overline{B}C$$



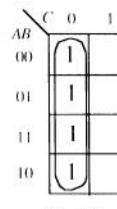
$$X = AC$$

$$(c) X = \overline{A}(\overline{B}C + B\overline{C}) + A(BC + B\overline{C}) \\ = \overline{A}BC + \overline{A}B\overline{C} + ABC + A\overline{B}\overline{C}$$



$$X = B$$

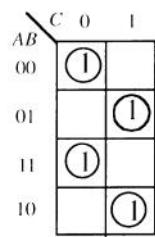
$$(d) X = \overline{A} \overline{B} \overline{C} + A \overline{B} \overline{C} + \overline{A} B \overline{C} + A B \overline{C}$$



$$X = \overline{C}$$

39. See Figure 4-

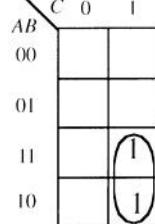
$$(a) X = \overline{A} \overline{B} \overline{C} + A \overline{B} C + \overline{A} B C + A B \overline{C}$$



No simplification

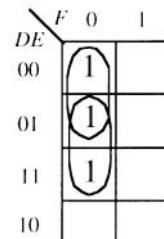
$$(b) X = AC[B + B(\overline{B} + \overline{C})]$$

$$= A\overline{B}C + ABC + AB\overline{C} = A\overline{B}C + ABC$$



$$X = AC$$

$$(c) X = D E \overline{F} + \overline{D} E \overline{F} + \overline{D} E F$$



$$X = \overline{DF} + E\overline{F}$$

40. (a)  $AB + A\overline{B}C + ABC = AB(C + \overline{C}) + A\overline{B}C + ABC$

$$\begin{aligned} &= ABC + A\overline{B}\overline{C} + A\overline{B}C + ABC \\ &= ABC + A\overline{B}C + A\overline{B}C \end{aligned}$$

(b)  $A + BC = A(B + \overline{B})(C + \overline{C}) + (\overline{A} + A)BC = (AB + A\overline{B})(C + \overline{C}) + (\overline{A} + A)BC$

$$\begin{aligned} &= ABC + A\overline{B}C + A\overline{B}C + A\overline{B}\overline{C} + \overline{A}BC + ABC \\ &= ABC + A\overline{B}C + A\overline{B}C + A\overline{B}\overline{C} + \overline{A}BC \end{aligned}$$

(c)  $A\overline{B}CD + A\overline{C}\overline{D} + B\overline{C}D + \overline{A}\overline{B}CD$

$$= A\overline{B}CD + A(B + \overline{B})CD + (A + \overline{A})B\overline{C}D + \overline{A}\overline{B}CD =$$

$$= A\overline{B}CD + ABC\overline{D} + A\overline{B}\overline{C}D = A\overline{B}CD + \overline{A}\overline{B}CD + \overline{A}\overline{B}CD$$

(d)  $A\overline{B} + A\overline{B}CD + CD + B\overline{C}D + ABCD$

$$= A\overline{B}(C + \overline{C})(D + \overline{D}) + A\overline{B}CD + (A + \overline{A})(B + \overline{B})CD + (A + \overline{A})B\overline{C}D + ABCD$$

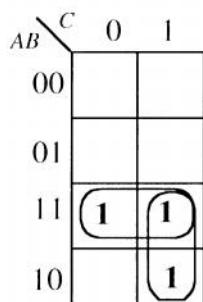
$$= A\overline{B}CD + A\overline{B}CD + ABC\overline{D} + ABCD + A\overline{B}CD + ABCD + A\overline{B}CD + \overline{A}\overline{B}CD$$

$$+ A\overline{B}CD + A\overline{B}CD + A\overline{B}CD + ABCD + A\overline{B}CD + A\overline{B}CD + A\overline{B}CD$$

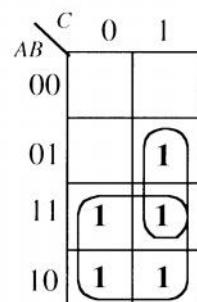
$$= A\overline{B}CD + A\overline{B}CD + ABC\overline{D} + ABCD + A\overline{B}CD + A\overline{B}CD + A\overline{B}CD + A\overline{B}CD + A\overline{B}CD$$

$$= A\overline{B}CD + A\overline{B}CD$$

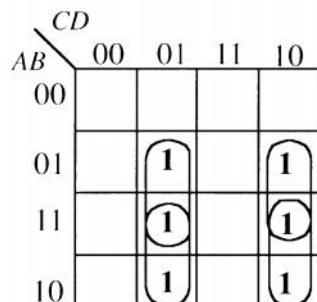
41. See Figure 4-13.



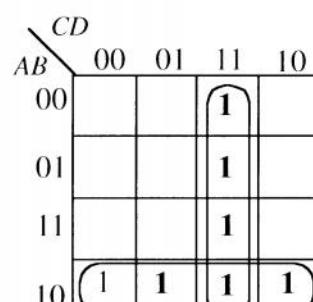
(a)  $X = AB + AC$



(b)  $X = A + BC$



(c)  $X = B\overline{C}D + B\overline{C}\overline{D} + A\overline{C}D + A\overline{C}\overline{D}$



(d)  $X = A\overline{B} + CD$

42.

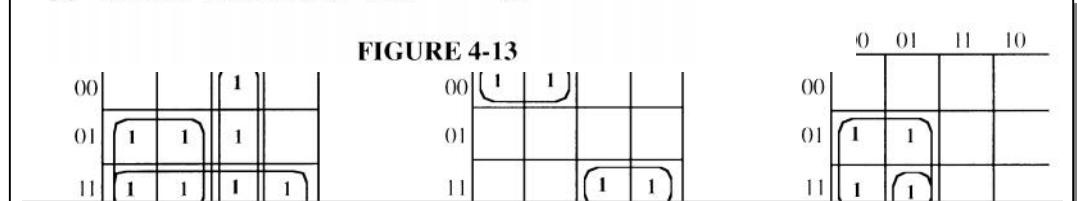
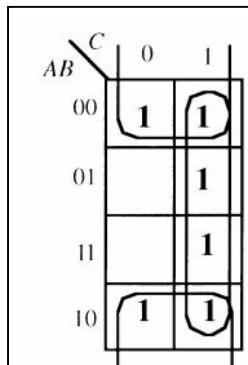


FIGURE 4-13



- 43.** Plot the 1's from Figure 4-62 in the text on the map as shown in Figure 4-15 and simplify.

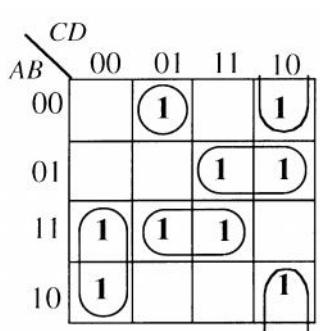


$$X = \overline{B} + C$$

- 44.** Plot the 1's from Figure 4-16 and simplify.

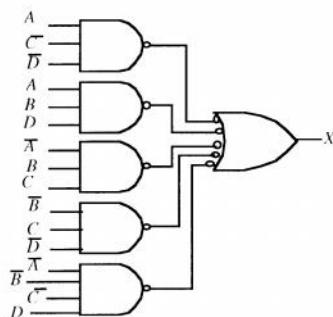
**FIGURE 4-15**

xt on the map as shown in

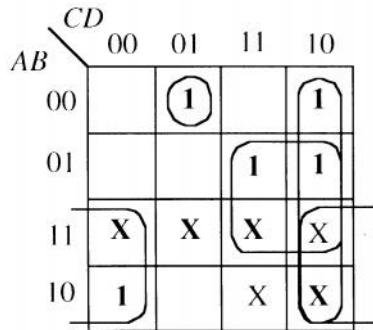


$$X = A\overline{C}\overline{D} + ABD + \overline{ABC} + \overline{BCD} + \overline{AB}\overline{CD}$$

- 45.**

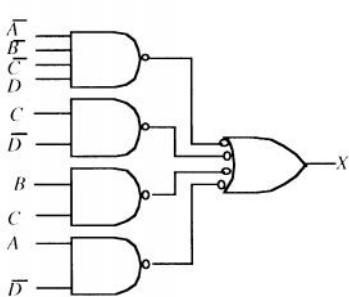


**FIGURE 4-16**



$$X = \overline{ABC}\overline{D} + C\overline{D} + BC + A\overline{D}$$

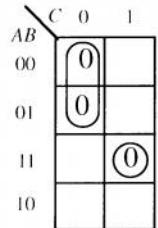
**FIGURE 4-17**



## Section 4-10 Karnaugh Map POS Minimization

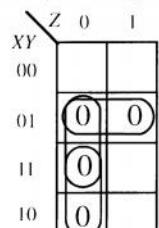
46. See Figure 4-18.

$$(a) X = (A + B + C)(\bar{A} + \bar{B} + \bar{C})(A + \bar{B} + C)$$



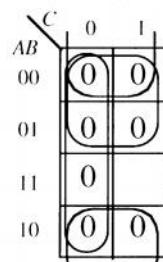
$$X = (A + C)(\bar{A} + \bar{B} + \bar{C})$$

$$(b) W = (X + \bar{Y})(\bar{X} + Z)(X + \bar{Y} + \bar{Z})(\bar{X} + Y + Z)$$

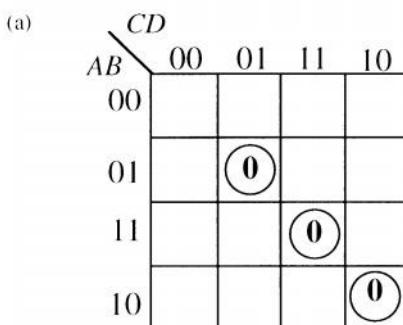


$$W = (X + \bar{Y})(\bar{X} + Z)(\bar{Y} + Z)$$

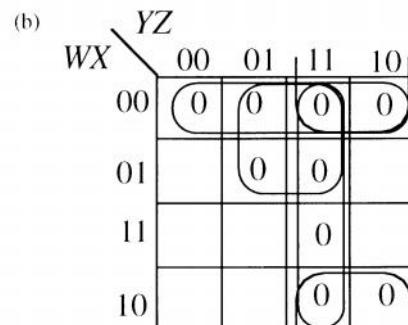
$$(c) X = A(B + \bar{C})(\bar{A} + C)(A + \bar{B} + C)(\bar{A} + B + \bar{C})$$



47.



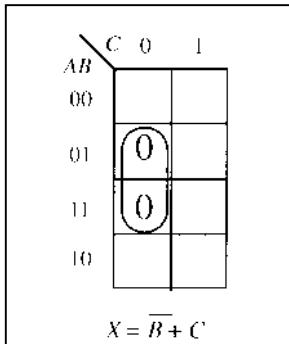
$$X = (A + \bar{B} + C + \bar{D})(\bar{A} + B + \bar{C} + D)(\bar{A} + \bar{B} + \bar{C} + \bar{D})$$



$$Q = (W + \bar{Z})(W + X)(\bar{Y} + \bar{Z})(X + \bar{Y})$$

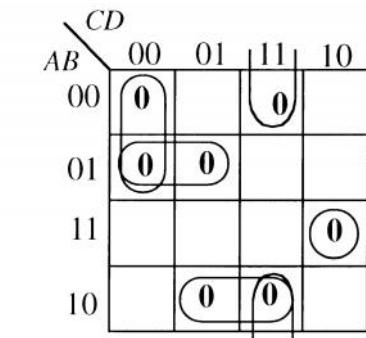
FIGURE 4-19

48. See Figure 4-20.



49. See Figure 4-

**FIGURE 4-20**

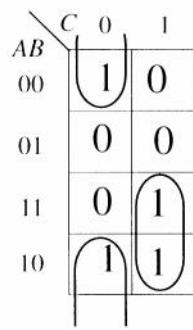


$$X = (A + C + D)(A + \overline{B} + \overline{C})(\overline{A} + B + \overline{D})$$

$$(B + \overline{C} + \overline{D})(\overline{A} + \overline{B} + \overline{C} + D)$$

50. See Figure 4-

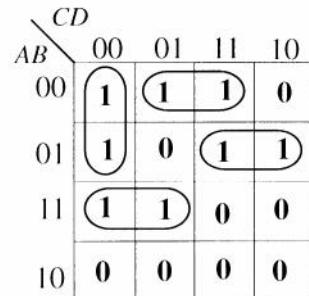
(a)  $(A + \overline{B})(A + \overline{C})(\overline{A} + \overline{B} + C)$



$$X = AC + \overline{B}\overline{C}$$

$$X = A(C + BC)$$

(b)  $(\overline{A} + B)(\overline{A} + \overline{B} + \overline{C})(B + \overline{C} + D)(A + \overline{B} + C + \overline{D})$

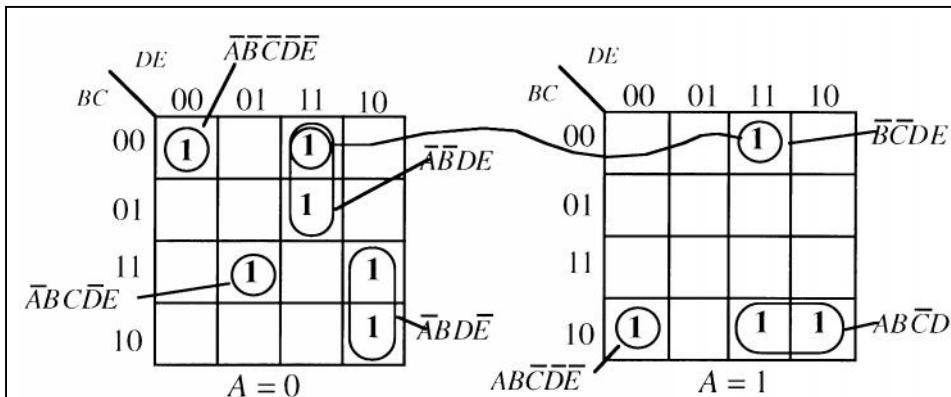


$$X = \overline{A}\overline{C}\overline{D} + ABC\overline{C} + A\overline{B}\overline{D} + A\overline{B}C$$

**FIGURE 4-22**

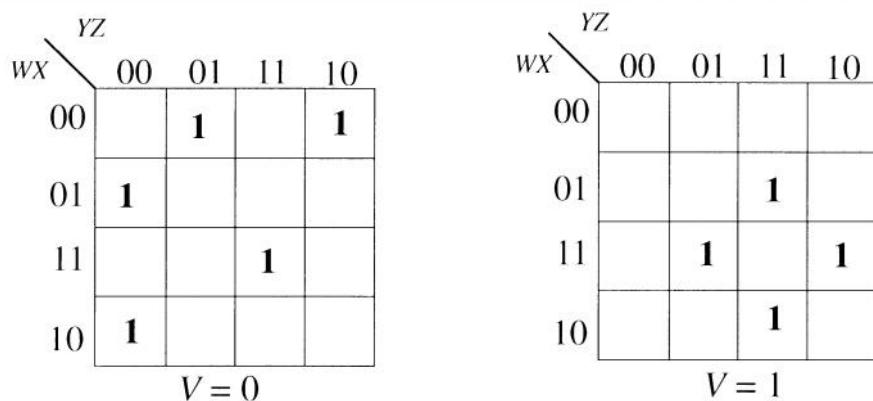
**Section 4-11 Five-Variable Karnaugh Maps**

51. See Figure 4-23.



$$X = \overline{ABC}\overline{D}\overline{E} + \overline{ABC}\overline{D}E + ABC\overline{D}\overline{E} + \overline{AB}DE + \overline{AB}D\overline{E} + \overline{BC}DE + AB\overline{C}D$$

- 52.

**FIGURE 4-23**

No simplification is possible

**FIGURE 4-24**

**Section 4-12 VHDL**

53. **entity AND\_OR is**  
**port (A, B, C, D, E, F, G, H, I: in bit; X: out bit);**  
**end entity AND\_OR;**  
**architecture Logic of AND\_OR is**  
**begin**  
**X <= (A and B and C) or (D and E and F) or (G and H and I);**  
**end architecture Logic;**
54. The VHDL program:
- ```
entity SOP is  

port (A, B, C: in bit; X: out bit);  

end entity SOP;  

architecture Logic of SOP is  

begin  

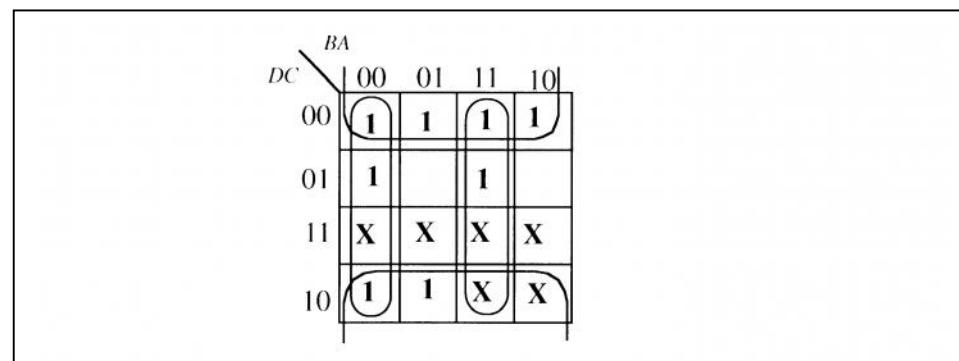
Y <= (A and not B and C) or (not A and not B and C) or  

(A and not B and not C) or (not A and B and C);  

end architecture Logic;
```

**Digital System Application**

55. An LED display is more suitable for low-light conditions because LEDs emit light and LCDs do not.
56. The codes 1010, 1011, 1100, 1101, 1110, and 1111 correspond to nondecimal digit values and are not used in the BCD code.
57. The standard SOP expression for segment *b* is:  
 $b = \overline{DCBA} + \overline{DCBA} + \overline{DCBA} + \overline{DCBA} + \overline{DCBA} + \overline{DCBA} + \overline{DCBA}$   
This expression is minimized in Figure 4-25.  
There are 6 fewer gates and one fewer inverters as a result of minimization.

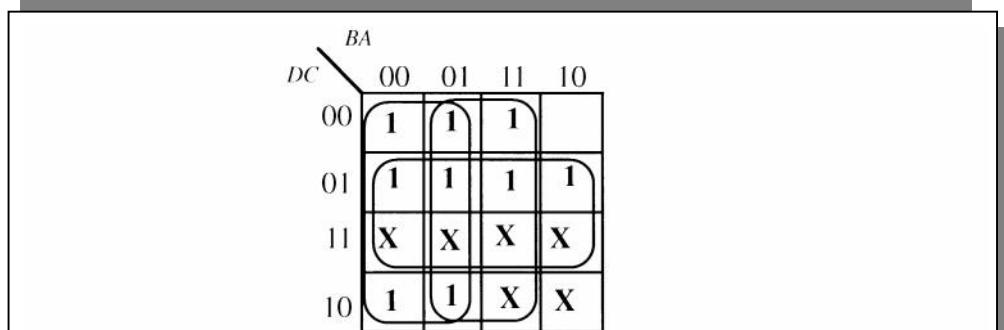


58. The standard expression requires 7 gates, one 8-input OR gate, and 4 inverters.

The minimum expression requires two 2-input AND gates, one 3-input OR gate, and 3 inverters.

**FIGURE 4-25**

minim



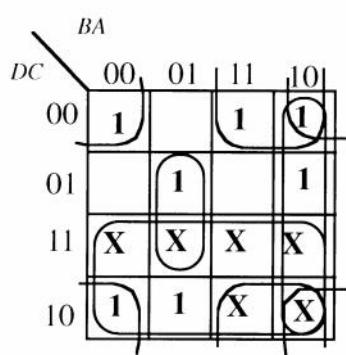
**FIGURE 4-**  
26

The standard SOP expression for segment  $d$  is:

$$d = \overline{DCBA} + \overline{DCB\bar{A}} + \overline{DCB\bar{A}} + \overline{DC\bar{B}A} + \overline{DC\bar{B}\bar{A}} + \overline{D\bar{C}BA} + D\overline{C\bar{B}A}$$

This expression is minimized in Figure 4-27.

There are 3 fewer gates and 1 fewer inverters as a result of minimization.

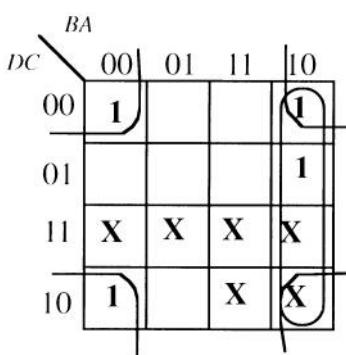


$$d = D + \overline{CA} + \overline{CB} + B\overline{A} + C\overline{BA}$$

The standard expression requires seven 2-input AND gates, one 7-input OR gate, and 4 inverters. **FIGURE 4-**  
27

The minimum expression requires three 2-input AND gates, one 3-input AND gate, one 5-input OR gate, and 3 inverters.

**FIGURE 4-27**



$$d = \overline{CA} + B\overline{A}$$

The standard expression requires four 4-input AND gates, one 4-input OR gate, and 4 inverters.

The minimum expression requires two 2-input AND gates, one 2-input OR gate, and 2 inverters.

**FIGURE 4-**  
28

The standard SOP expression for segment  $f$  is:

$$f = \overline{DCBA} + \overline{DC\bar{B}} + \overline{D\bar{C}B} + \overline{D\bar{C}\bar{B}A} + \overline{D\bar{C}B\bar{A}} + \overline{DC\bar{B}A}$$

This expression is minimized in Figure 4-29.

There are 3 fewer gates and 2 fewer inverters as a result of minimization.

|    |    | BA | 00 | 01 | 11 | 10 |
|----|----|----|----|----|----|----|
|    |    | DC | 00 | 01 | 11 | 10 |
| 00 | 00 | 1  |    |    |    |    |
| 01 | 01 | 1  | 1  |    |    | 1  |
| 11 | 11 | X  | X  | X  | X  |    |
| 10 | 10 | 1  | 1  | X  | X  |    |

$$f = C\bar{A} + \overline{B\bar{A}} + C\bar{B} + D$$

The standard expression requires six

**FIGURE 4-**

one 6-input OR gate, and 4 inverters.

The minimum expression requires three 2-input AND gates, one 4-input OR gate, and 2 inverters.

**FIGURE 4-29**

minimization.

|    |    | BA | 00 | 01 | 11 | 10 |
|----|----|----|----|----|----|----|
|    |    | DC | 00 | 01 | 11 | 10 |
| 00 | 00 |    |    |    | 1  | 1  |
| 01 | 01 | 1  | 1  |    |    | 1  |
| 11 | 11 | X  | X  | X  | X  |    |
| 10 | 10 | 1  | 1  | X  | X  |    |

$$g = \overline{C}B + B\overline{A} + C\overline{B} + D$$

The standard expression requires

**FIGURE 4-**

one 7-input OR gate, and 4 inverters.

The minimum expression requires three 2-input AND gates, one 4-input OR gate, and 3 inverters.

**Spec**

**FIGURE 4-30**

59. Connect one OR gate output for each segment to an inverter and then use the inverter output to drive the segment.

60. See Figure 4-31. The POS implementation requires one 3-input OR gate, one 4-input OR gate, one 2-input AND gate, and 2 inverters. The SOP implementation (see Figure 4-55 in text) requires two 2-input AND gates, one 4-input OR gate, and 2 inverters.

|  |  | BA | 00 | 01 | 11 | 10 |
|--|--|----|----|----|----|----|
|  |  | DC | 00 | 01 | 11 | 10 |
|  |  |    | 00 | 01 | 11 | 10 |
|  |  |    | 1  | 0  | 1  | 1  |
|  |  |    | 0  | 1  | 1  | 1  |
|  |  |    | X  | X  | X  | X  |
|  |  |    | 1  | 1  | X  | X  |

**FIGURE 4-31**  
 $a = \overline{C} + \overline{B} + A$

61. See Figure 4-32. The SOP implementation requires one 3-input OR gate, and 2 inverters.

|  |  | BA | 00 | 01 | 11 | 10 |
|--|--|----|----|----|----|----|
|  |  | DC | 00 | 01 | 11 | 10 |
|  |  |    | 00 | 01 | 11 | 10 |
|  |  |    | 1  | 1  | 1  | 1  |
|  |  |    | 1  | 0  | 1  | 0  |
|  |  |    | X  | X  | X  | X  |
|  |  |    | 1  | 1  | X  | X  |

**FIGURE 4-32**  
 $b = C + \overline{B} + A$

See Figure 4-33. The POS implementation requires one 3-input OR gate, and 2 inverters.

|  |  | BA | 00 | 01 | 11 | 10 |
|--|--|----|----|----|----|----|
|  |  | DC | 00 | 01 | 11 | 10 |
|  |  |    | 00 | 01 | 11 | 10 |
|  |  |    | 1  | 1  | 1  | 0  |
|  |  |    | 1  | 1  | 1  | 1  |
|  |  |    | X  | X  | X  | X  |
|  |  |    | 1  | 1  | X  | X  |

$$c = C + \overline{B} + A$$

See Figure 4-34. The SOP implementation requires one 3-input OR gate, and 2 inverters.

segment *b* requires two 3-inverters.

segment *c* requires one 3-input OR gate, and 2 inverters.

segment *d* requires one 4-input AND gate, and 3 inverters.

|  |  | BA | 00 | 01 | 11 | 10 |
|--|--|----|----|----|----|----|
|  |  | DC | 00 | 01 | 11 | 10 |
|  |  |    | 00 | 01 | 11 | 10 |
|  |  |    | 1  | 0  | 1  | 1  |
|  |  |    | 0  | 1  | 0  | 1  |
|  |  |    | X  | X  | X  | X  |
|  |  |    | 1  | 1  | X  | X  |

$$d = (\overline{C} + B + A)(\overline{C} + \overline{B} + \overline{A})(D + C + B + \overline{A})$$

**FIGURE 4-**

See Figure 4-35. The POS implementation of segment e requires one 2-input OR gate, one 2-input AND gate, and 2 inverters.

| DC \ BA | 00 | 01 | 11 | 10 |
|---------|----|----|----|----|
| 00      | 1  | 0  | 0  | 1  |
| 01      | 0  | 0  | 0  | 1  |
| 11      | X  | X  | X  | X  |
| 10      | 1  | 0  | X  | X  |

$e = \overline{A}(\overline{C} + B) \quad \text{FIGURE 4-35}$

See Figure 4-35.  
input OR gate  
3-input OR gate

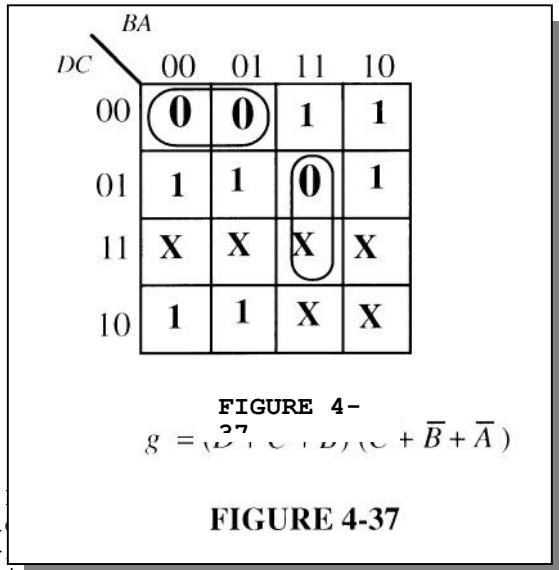
segment f requires two 2-inverters.

| DC \ BA | 00 | 01 | 11 | 10 |
|---------|----|----|----|----|
| 00      | 1  | 0  | 0  | 0  |
| 01      | 1  | 1  | 0  | 1  |
| 11      | X  | X  | X  | X  |
| 10      | 1  | 1  | X  | X  |

$f = (\overline{B} + \overline{C} + \overline{A}) + C + \overline{A} \quad \text{FIGURE 4-36}$

**FIGURE 4-36**

See Figure 4-37. The POS implementation of segment  $g$  requires two 3-input OR gates, one 2-input AND gate, and 3 inverters.



**FIGURE 4-37**  
 $g = \overline{A} + \overline{B} + \overline{C} + \overline{B} + \overline{A}$

62. For the 4 inverters:  
 4 2-  
 2 3-input AND gates.  
 9 3-input OR gates: 5 7432s  
 2 4-input OR gates: 2 7432s  
 3 2-input OR gates: 1 7432  
**Total ICs for the POS:** one 7404, one 7408s, one 7411, and eight 7432s

at decoding logic:

### **Multisim Troubleshooting Practice**

63. Input A inverter output open.  
 64. Input A of segment e OR gate open.  
 65. Segment b OR gate output open.

---

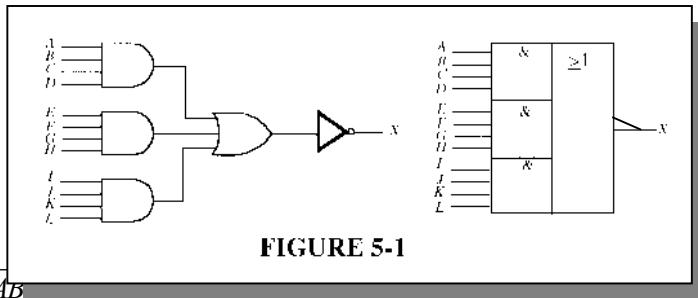
## CHAPTER 5

### COMBINATIONAL LOGIC ANALYSIS

---

#### **Section 5-1 Basic Combinational Logic Circuits**

1. See Figure 5-1.



**FIGURE 5-1**

2. (a)  $X = \overline{AB}$
- (b)  $X = \overline{\overline{AB} + \overline{ACD} + DB\overline{D}}$
3. (a)  $X = ABB$
- (b)  $X = AB + B$
- (c)  $X = \overline{A} + B$
- (d)  $X = (\overline{A} + B) + AB$
- (e)  $X = \overline{ABC}$
- (f)  $X = (A + B)(\overline{B} + C)$
4. See Figure 5-2 for the circuit corresponding to each expression.
- (a)  $X = (A + B)(C + D) = AC + AD + BC + BD$
- (b)  $X = \overline{\overline{ABC} + \overline{CD}} = (\overline{ABC})(CD) = (\overline{A} + \overline{B})CCD = \overline{ACD} + \overline{BCD}$
- (c)  $X = \overline{\overline{(AB + C)}D + E} = ABD + CD + E$
- (d)  $X = \overline{\overline{(A + B)}(\overline{BC}) + D} = \overline{\overline{(A + B)}(\overline{BC})} + D = \overline{A} + B + BC + D = \overline{A} + B + D$
- (e)  $X = \overline{\overline{(AB + \overline{C})D} + \overline{E}} = (AB + \overline{C})D + \overline{E} = ABD + \overline{CD} + \overline{E}$
- (f) 
$$\begin{aligned} X &= \overline{\overline{(AB + \overline{CD})(\overline{EF} + \overline{GH})}} = \overline{(AB + CD)(EF + GH)} = \overline{(AB + CD)} + \overline{(EF + GH)} \\ &= (\overline{AB})(\overline{CD}) + (\overline{EF})(\overline{GH}) \\ &= (\overline{A})(\overline{B})(\overline{C})(\overline{D}) + (\overline{E})(\overline{F})(\overline{G})(\overline{H}) \\ &= (\overline{A} + \overline{B})(\overline{C} + \overline{D}) + (\overline{E} + \overline{F})(\overline{G} + \overline{H}) = \overline{AC} + \overline{BC} + \overline{AD} + \overline{BD} + \overline{EG} + \overline{FG} + \overline{EH} + \overline{FH} \end{aligned}$$

5.

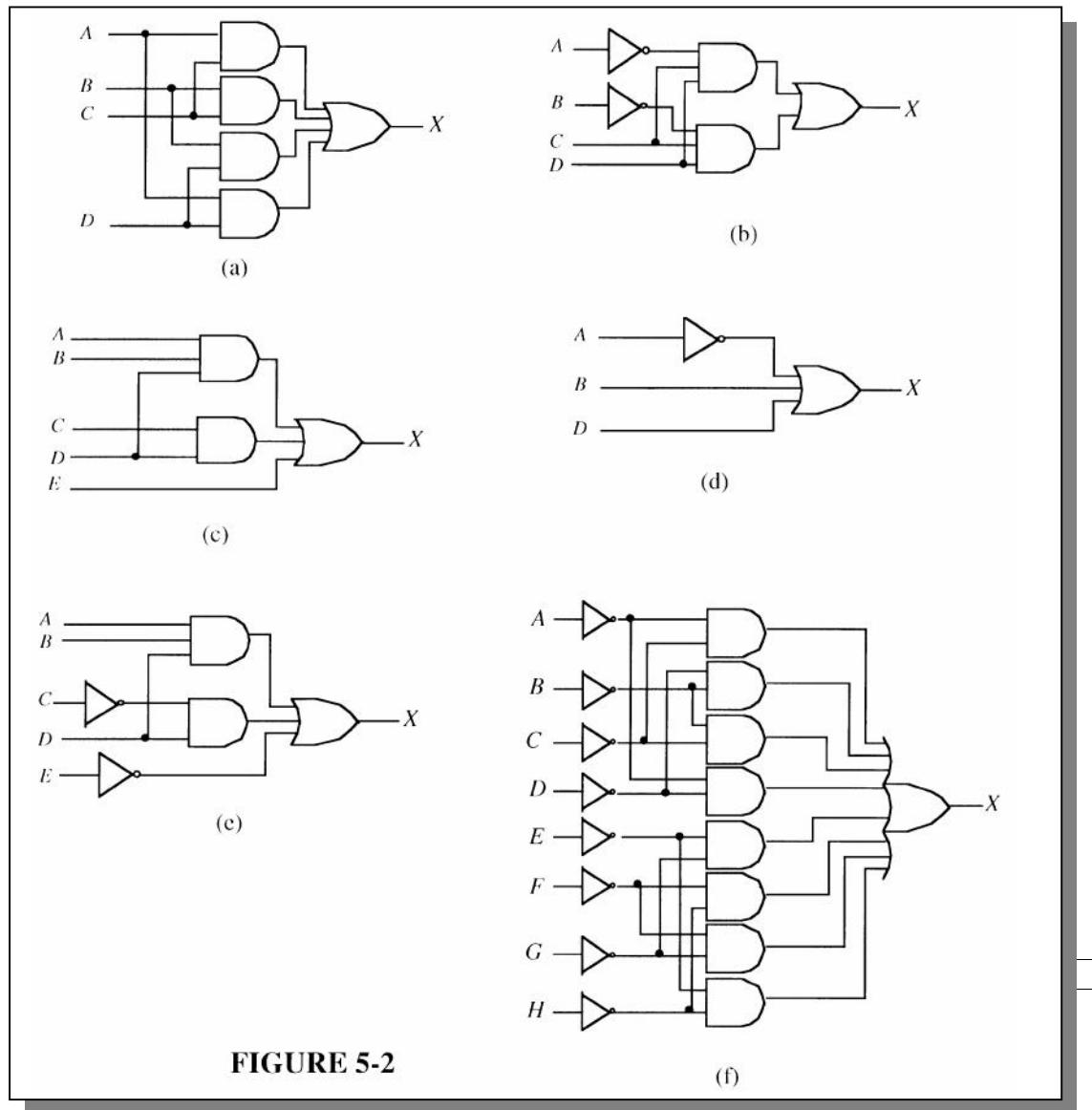


FIGURE 5-2

$$+ AB$$

$$(d) \quad X = (\bar{A} + B)$$

| A | B | X |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

$$(e)$$

$$X = \overline{\overline{ABC}}$$

| A | B | C | X |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

|   |   |   |   |
|---|---|---|---|
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

$$(c)$$

$$X = \overline{A} + B$$

| A | B | X |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

6. (a)  $X = (A + B)(C + D)$

| A | B | C | D | X |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

(b)  $X = \overline{\overline{ABC} + \overline{CD}}$

| A | B | C | D | X |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 |

(c)  $X = (AB + C)D + E$

| A | B | C | D | E | X | A | B | C | D | E | X |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

(d)  $X =$

$$\overline{\overline{(A+B)(BC)}+D}$$

| A | B | C | D | X |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 |

(e)  $X = \overline{(\overline{AB} + \overline{C})D} + \overline{E}$

| A | B | C | D | E | X |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 |

(f)  $X = \overline{(\overline{AB} + \overline{CD})(\overline{EF} + \overline{GH})}$

| A | B | C | D | E | F | G | H | I |
|---|---|---|---|---|---|---|---|---|
| 0 | X | 0 | X | X | X | X | X | 1 |
| X | 0 | 0 | X | X | X | X | X | 1 |
| 0 | X | X | 0 | X | X | X | X | 1 |
| X | 0 | X | 0 | 0 | X | X | X | 1 |
| X | X | X | X | 0 | X | 0 | X | 1 |
| X | X | X | X | 0 | X | X | 0 | 1 |
| X | X | X | X | X | 0 | X | 0 | 1 |
| X | X | X | X | X | 0 | X | 0 | 1 |

For all other entries  $X = 0$ .

$X = \text{don't care}$

An abbreviated table is shown because there are 256 combinations.

7.  $X = \overline{A}\overline{B} + \overline{A}B = (\overline{A}\overline{B})(\overline{A}B) = (\overline{A} + B)(A + \overline{B})$

## Section 5-2 Implementing Combinational Logic

8. See Figure 5-3.

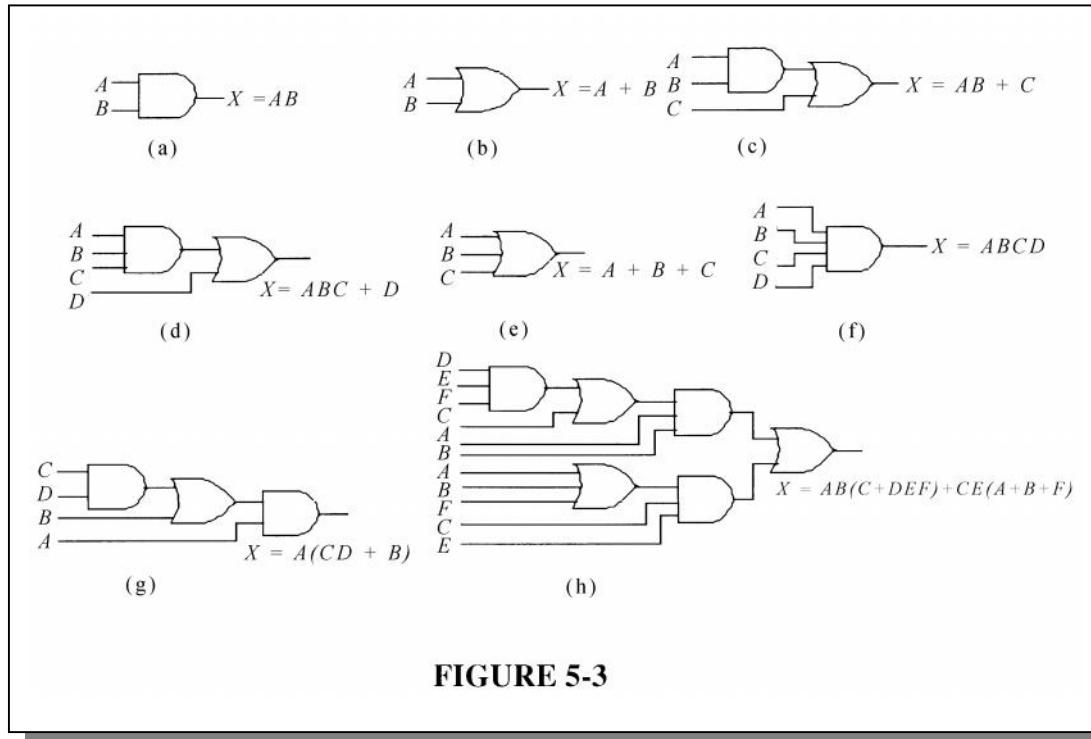
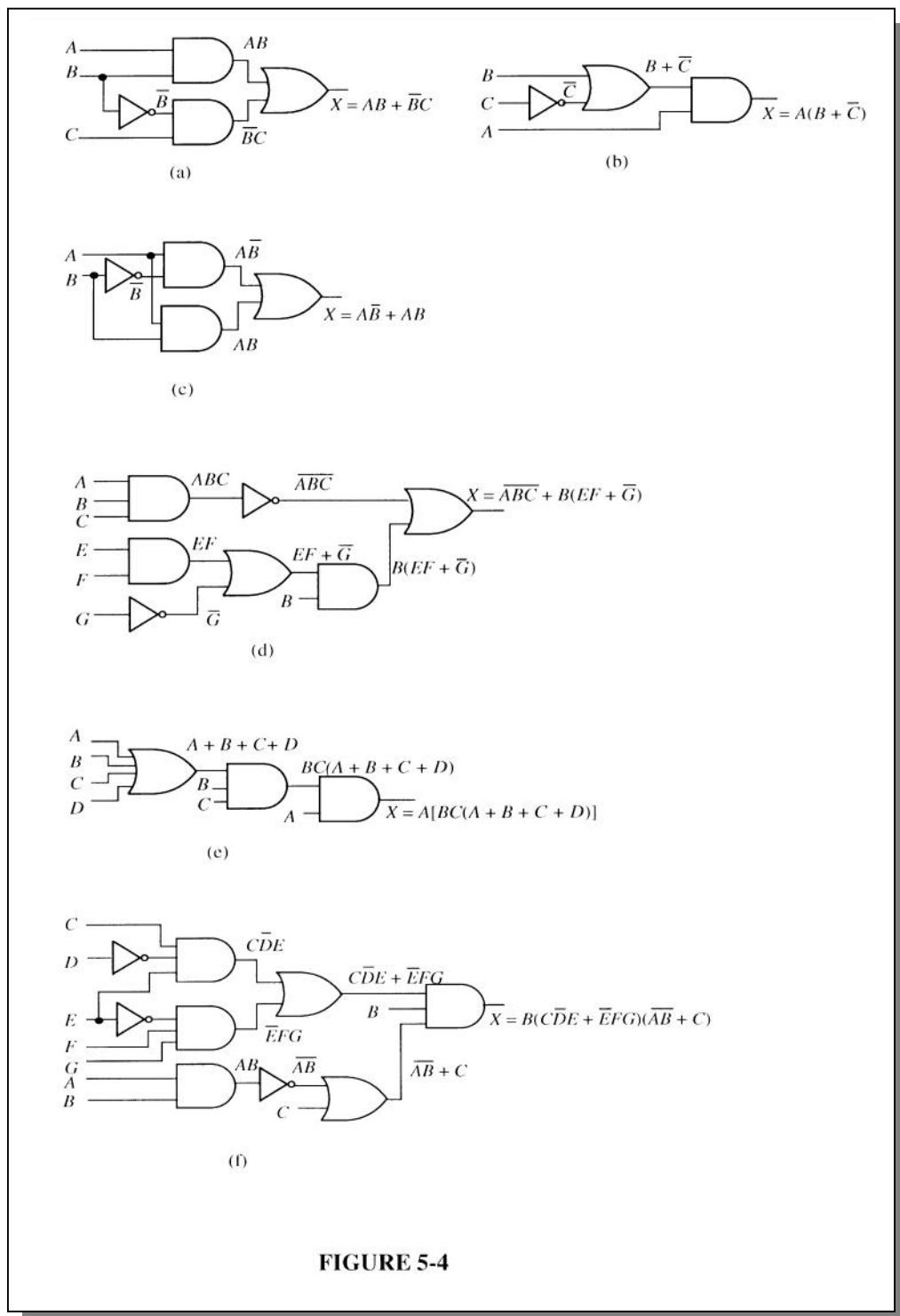


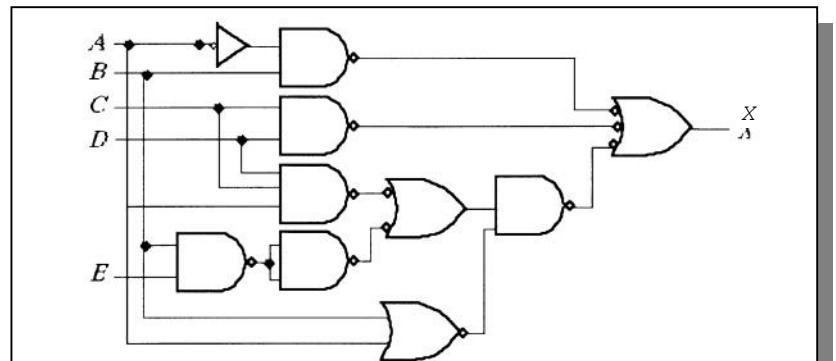
FIGURE 5-3

9. See Figure 5-4.

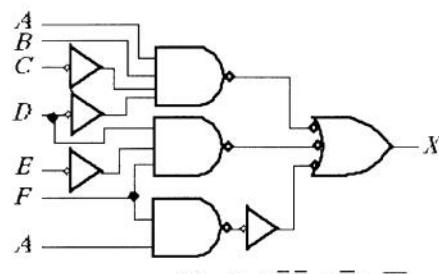


**FIGURE 5-4**

10. See Figure 5-5.



$$(a) \quad X = \bar{A}B + CD + (\bar{A} + \bar{B})(ACD + \bar{E}E)$$



$$(b) \quad X = A\bar{B}\bar{C}\bar{D} + D\bar{E}\bar{F} + \bar{A}\bar{F}$$

11.

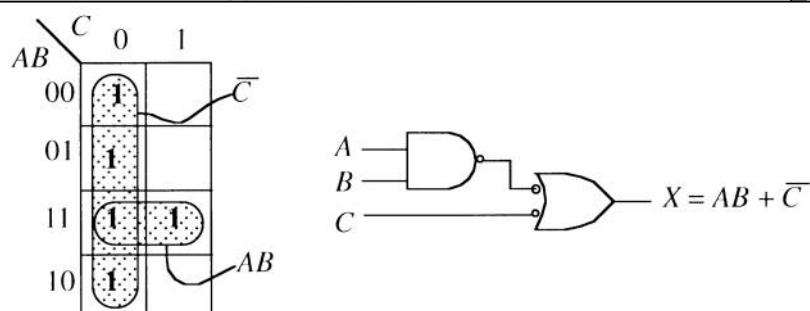
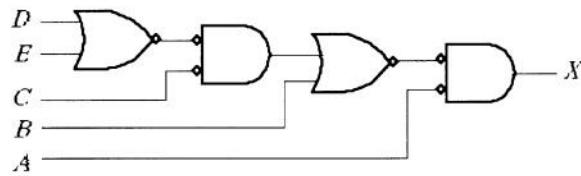


FIGURE 5-6

**12.**  $x = \overline{ABCD} + \overline{ABCD} + \overline{\overline{AB}\overline{CD}} + \overline{ABC}\overline{D} + \overline{AB}\overline{C}\overline{D} + \overline{A}\overline{B}\overline{C}\overline{D} + A\overline{B}\overline{C}\overline{D} + AB\overline{C}\overline{D} + ABC\overline{D}$   
See Figure 5-7.

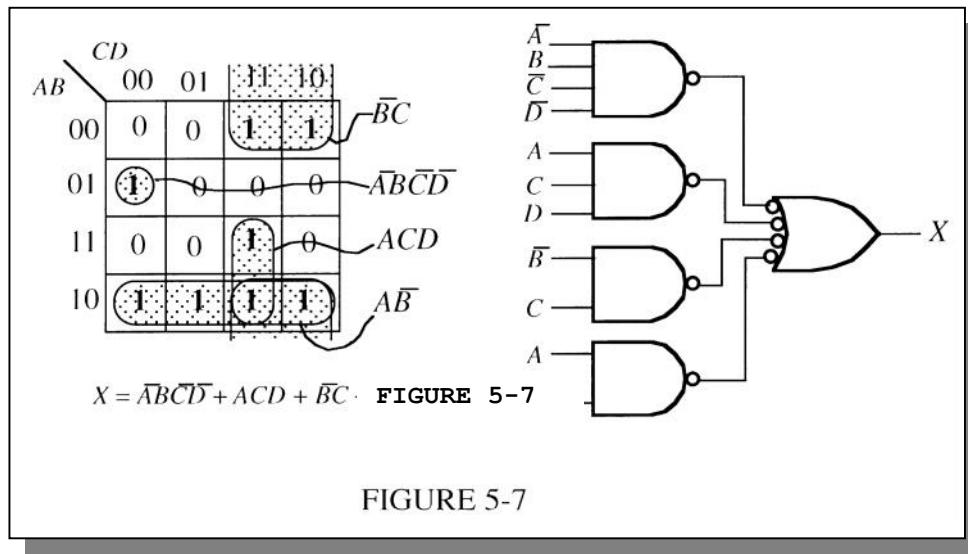


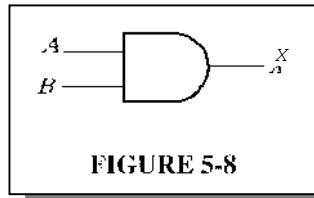
FIGURE 5-7

| A | B | C | X |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

| $A$ | $B$ | $X$ |
|-----|-----|-----|
| 0   | 0   | 0   |
| 0   | 1   | 0   |
| 1   | 0   | 0   |
| 1   | 1   | 1   |

$X = 1$  when  $AB = 1$ , no matter what  $C$  is.

Since  $C$  is a don't care variable, the output depends only on  $A$  and  $B$  as shown by the two-variable truth table above which is implemented with the AND gate in Figure 5-8.



**FIGURE 5-8**

$$\begin{aligned}
 14. \quad x &= \overline{\overline{(AB)}(B+C)} + C = (AB)(B+C)C = (AB)(B+C)\bar{C} = (\bar{A} + \bar{B})(\bar{BC})\bar{C} \\
 &= (\bar{ABC} + \bar{BC})\bar{C} = \bar{ABC} + \bar{BC} = \bar{BC}(A+1) = \bar{BC}
 \end{aligned}$$

See Figure 5

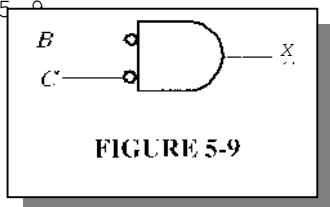


FIGURE 5-9

The output is dependent only on  $B$  and  $C$ . The value of  $A$  does not matter. The NOR gate behaves as a negative-AND.

| <i>A</i> | <i>B</i> | <i>C</i> | <i>X</i> |
|----------|----------|----------|----------|
| 0        | 0        | 0        | 1        |
| 0        | 0        | 1        | 0        |
| 0        | 1        | 0        | 0        |
| 0        | 1        | 1        | 0        |
| 1        | 0        | 0        | 1        |
| 1        | 0        | 1        | 0        |
| 1        | 1        | 0        | 0        |
| 1        | 1        | 1        | 0        |

$$15. \quad (a) \quad X = AB + \overline{BC}$$

No simplification. See Figure 5-10.

|      |                     |        |            |
|------|---------------------|--------|------------|
|      | $C$                 | 0      | 1          |
| $AB$ | 00                  | 0      | 1          |
|      | 01                  | 0      | 0          |
|      | 11                  | 1      | 0          |
|      |                     | $\sim$ | $\sim B C$ |
|      | $X = AB + \bar{B}C$ |        |            |

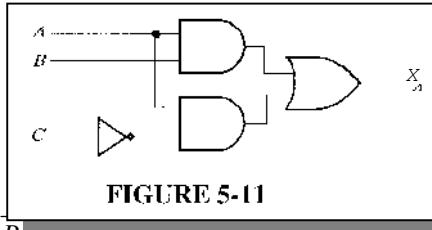
$$X = AB$$

No simplification is possible

**FIGURE 5-10**

$$(b) \quad X = A(B + C)$$

No simplification. Equation can be expressed in another form, as indicated in Figure 5-11.



**FIGURE 5-11**

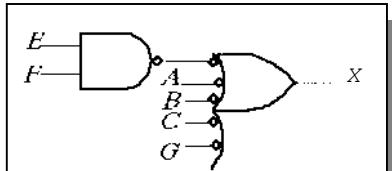
$$(c) \quad X = AB + AB - A(B + B) = A$$

A direct connection from input to output. No gates required.

$$(d) \quad X = \overline{ABC} + B(EF + \overline{G}) = \overline{A} + \overline{B} + \overline{C} + BEF + B\overline{G}$$

$$= \overline{A} + \overline{C} + BEF + \overline{B} + \overline{G} = \overline{A} + \overline{C} + \overline{B} + EF + \overline{G}$$

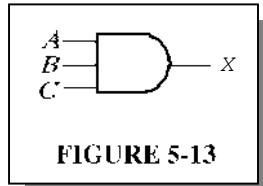
See Figure 5-12.



**FIGURE 5-12**

$$\begin{aligned}
 (e) \quad X &= A(BC(A + B + C + D)) = BCB + ABCC + ABCD \\
 &= ABC + ABC + ABCD + ABCD = ABC + ABC(I + D) \\
 &= ABC + ABC = \mathbf{ABC}
 \end{aligned}$$

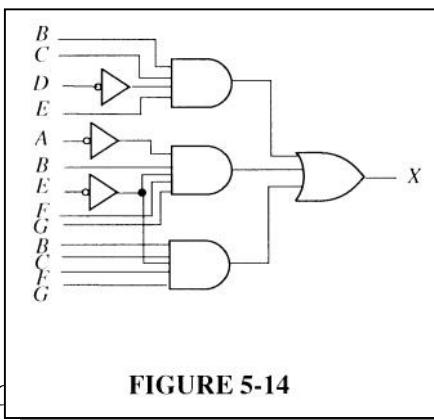
See Figure 5-13.



**FIGURE 5-13**

$$\begin{aligned}
 (f) \quad X &= B(C\bar{D}\bar{E} + \bar{E}\bar{F}\bar{G})(AB + C) = (BC\bar{D}\bar{E} + B\bar{E}\bar{F}\bar{G})(\bar{A} + \bar{B} + C) \\
 &= \bar{A}BC\bar{D}\bar{E} + \bar{A}B\bar{E}\bar{F}\bar{G} + BC\bar{D}\bar{E} + BC\bar{E}\bar{F}\bar{G} \\
 &= BC\bar{D}\bar{E}(\bar{A} + 1) + \bar{A}B\bar{E}\bar{F}\bar{G} + BC\bar{E}\bar{F}\bar{G} \\
 &= BC\bar{D}\bar{E} + \bar{A}B\bar{E}\bar{F}\bar{G} + BC\bar{E}\bar{F}\bar{G}
 \end{aligned}$$

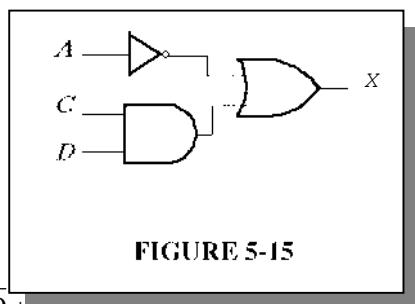
See Figure 5-14.



**FIGURE 5-14**

$$\begin{aligned}
 16. \quad (a) \quad X &= \bar{A}B + C \\
 &= \bar{A}B + CD + AB + ABE = A(B + B) + CD + ABE \\
 &= \bar{A} + \bar{A}BE + CD = \bar{A}(1 + \bar{B}E) + CD = \bar{A} + CD
 \end{aligned}$$

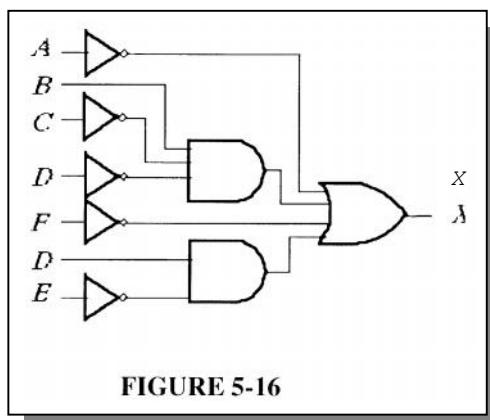
See Figure 5-15.



**FIGURE 5-15**

$$\begin{aligned}
 (b) \quad X &= ABCD + D\bar{E}F + A\bar{F} = ABCD + D\bar{E}F + A + \bar{F} \\
 &= \bar{A} + \bar{B}\bar{C}\bar{D} + \bar{F} + D\bar{E}
 \end{aligned}$$

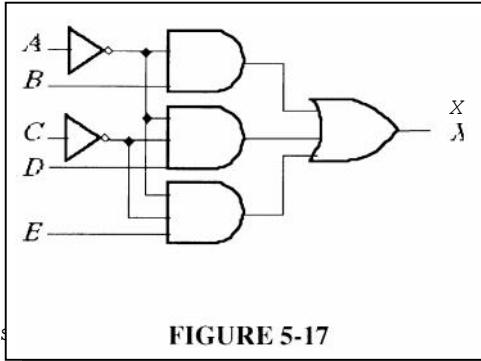
See Figure 5-16.



**FIGURE 5-16**

$$(c) \quad X = \overline{A}(B + \overline{C}(D + E)) = \overline{A}(B + \overline{C}D + \overline{C}E) = \overline{AB} + \overline{AC}\overline{D} + \overline{AC}\overline{E}$$

See Figure 5-17.



17. The SOP expressions are shown in Figure 5-18.

**FIGURE 5-17**

the resulting circuits

$$(a) \quad X = (A + B)(C + D) = \mathbf{AC} + \mathbf{AD} + \mathbf{BC} + \mathbf{BD}$$

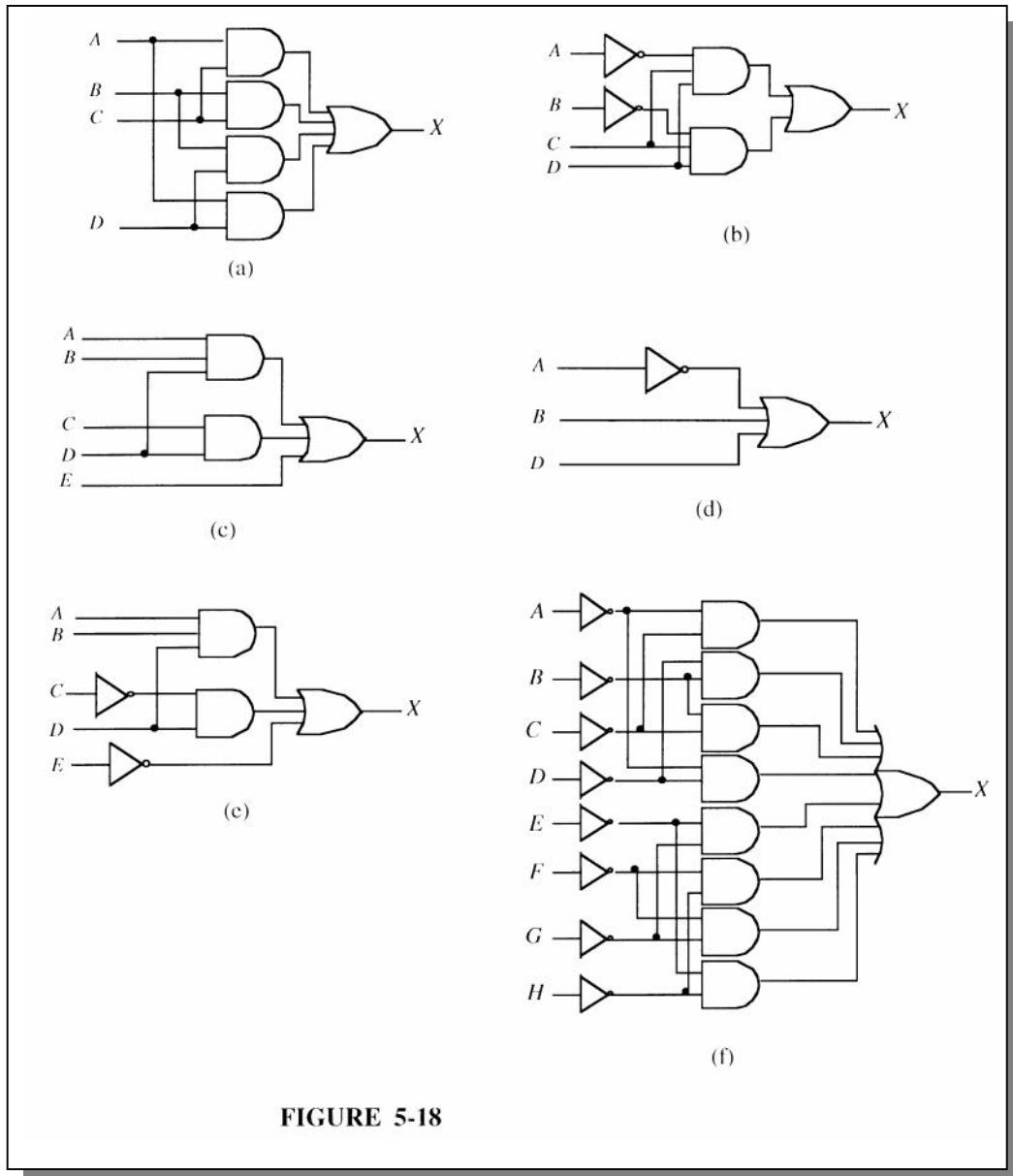
$$(b) \quad X = \overline{\overline{ABC} + \overline{CD}} = (\overline{ABC})(\overline{CD}) = (\overline{A} + \overline{B})\overline{C}\overline{D} = \overline{ACD} + \overline{BCD}$$

$$(c) \quad X = (\overline{AB} + \overline{C})D + \overline{E} = \mathbf{ABD} + \mathbf{CD} + \mathbf{E}$$

$$(d) \quad X = (\overline{\overline{A} + B})(\overline{\overline{BC}}) + D = (\overline{\overline{A} + B})(\overline{\overline{BC}}) + D = \overline{A} + B + BC + D \\ = \overline{A} + B(1 + C) + D = \overline{A} + \mathbf{B} + \mathbf{D}$$

$$(e) \quad X = (\overline{\overline{AB} + \overline{C}})D + \overline{E} = (AB + \overline{C})D + \overline{E} = \mathbf{ABD} + \overline{CD} + \overline{E}$$

$$(f) \quad X = (\overline{\overline{AB} + \overline{CD}})(\overline{\overline{EF} + \overline{GH}}) = (\overline{AB + CD})(\overline{EF + GH}) = (\overline{AB + CD}) + (\overline{EF + GH}) \\ = (\overline{AB})(\overline{CD}) + (\overline{EF})(\overline{GH}) = (\overline{A} + \overline{B})(\overline{C} + \overline{D}) + (\overline{E} + \overline{F})(\overline{G} + \overline{H}) \\ = \overline{AC} + \overline{BC} + \overline{AD} + \overline{BD} + \overline{EG} + \overline{FG} + \overline{EH} + \overline{FH}$$



**FIGURE 5-18**

**Section 5-3 The Universal Property of NAND and NOR Gates**

18. See Figure 5-19.

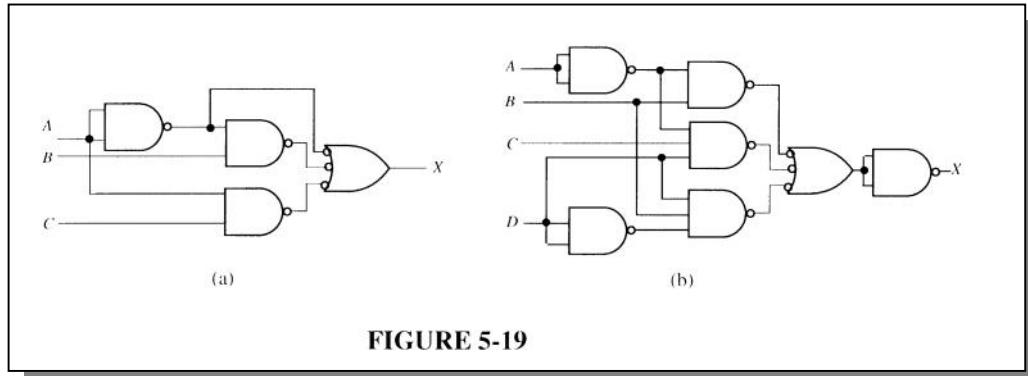


FIGURE 5-19

19.

See Figure 5-20.

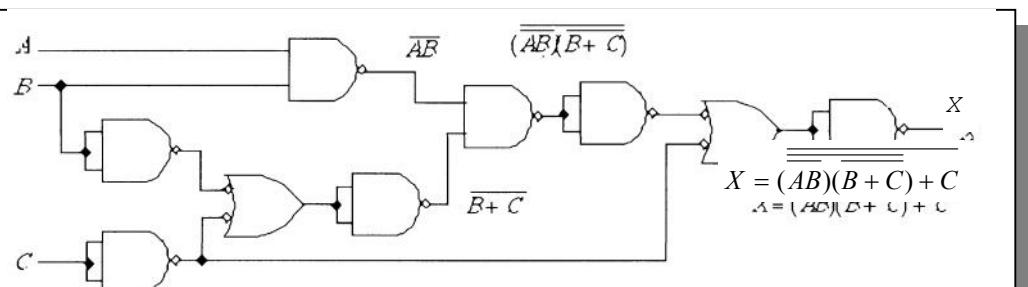


FIGURE 5-20

20.

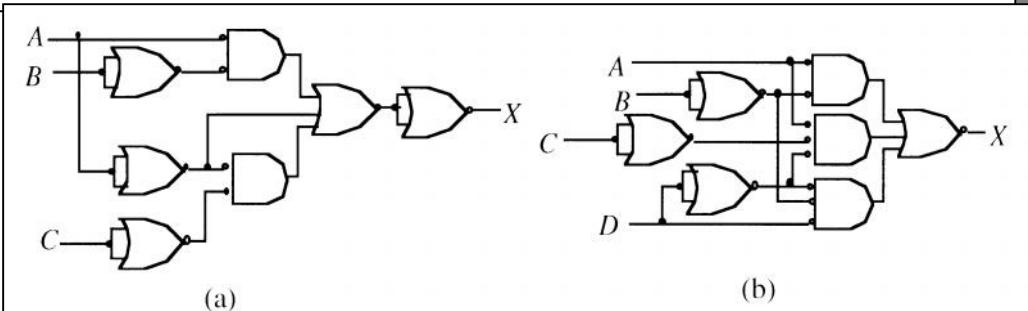
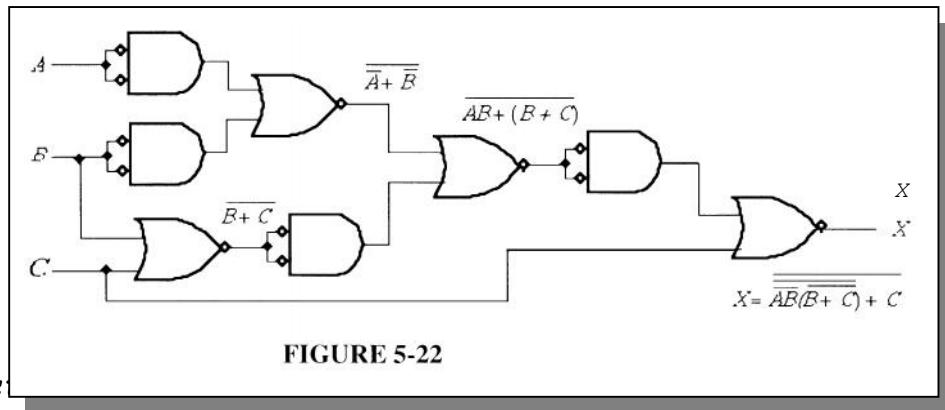


FIGURE 5-21

**21.** See Figure 5-22.



**FIGURE 5-22**

## *Sec Gates*

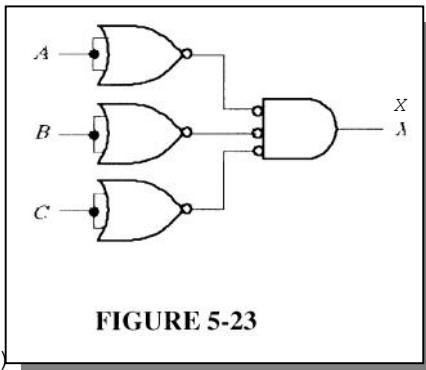
nd NOR

$$22. \quad (a) \quad X = ABC$$

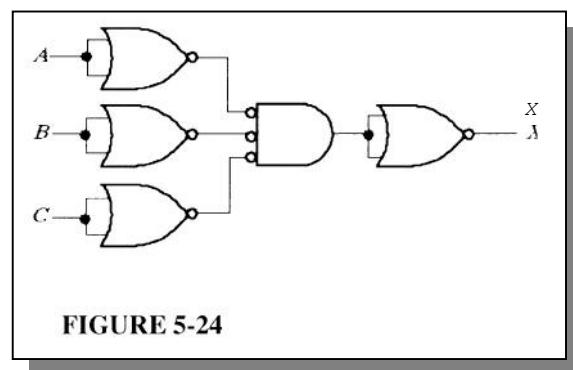
$$(b) \quad X = \overline{ABC}$$

See Figure 5-23.

See Figure 5-24.



**FIGURE 5-23**

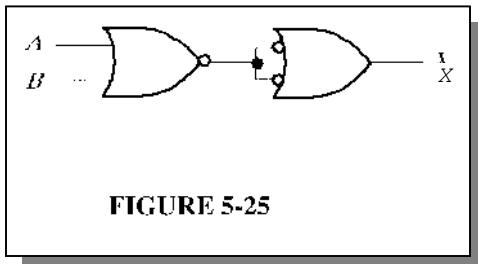


**FIGURE 5-24**

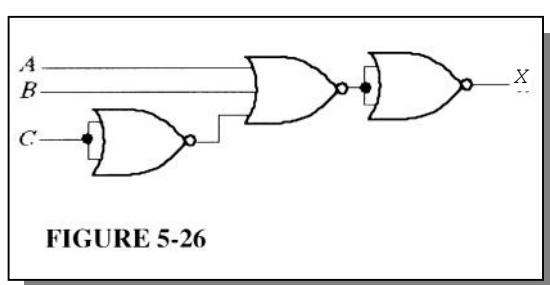
(c)

See Figure 5-25.

See Figure 5-26.



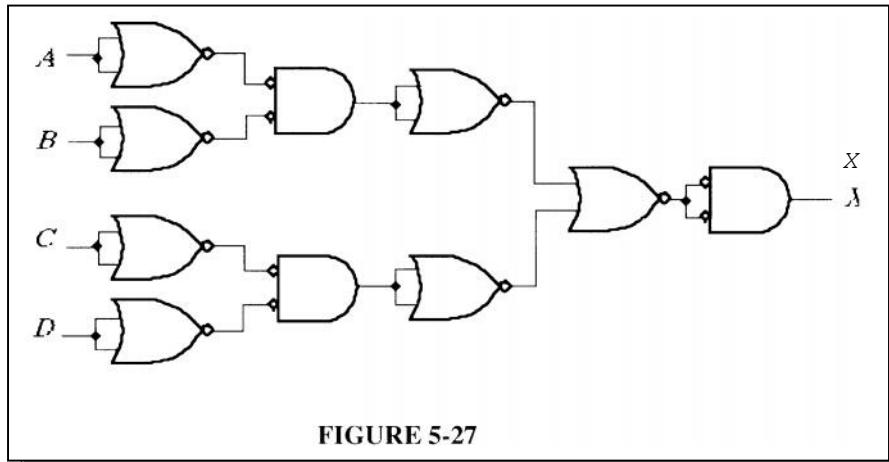
**FIGURE 5-25**



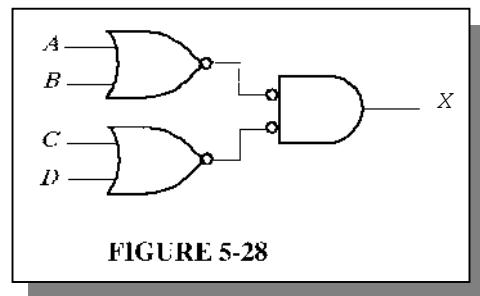
**FIGURE 5-26**

(e)  $X = \overline{AB} + \overline{CD}$

See Figure 5-27.

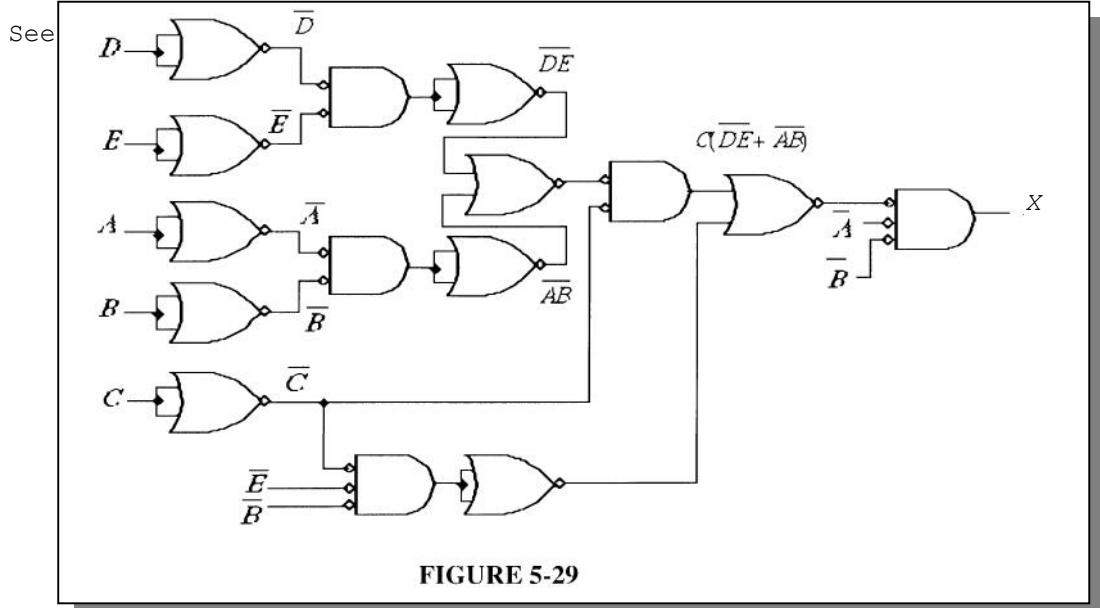


See Figure 5-28.



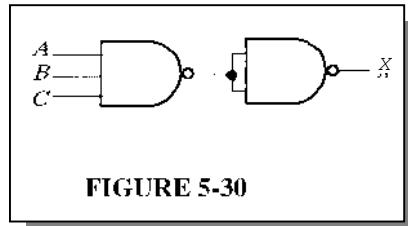
(g)

See Figure 5-29.



23. (a)  $X = ABC$

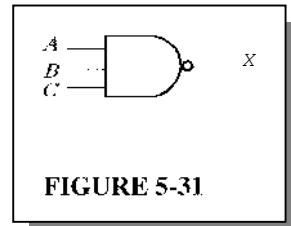
See Figure 5-30.



**FIGURE 5-30**

(b)  $X = \overline{ABC}$

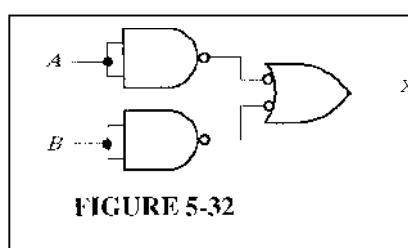
See Figure 5-31.



**FIGURE 5-31**

(c)  $X = A + B$

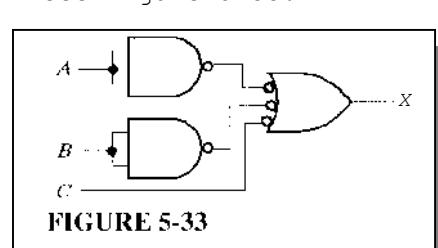
See Figure 5-32.



**FIGURE 5-32**

(d)  $X = A + B + C$

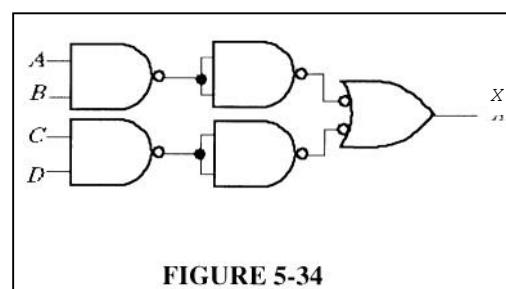
See Figure 5-33.



**FIGURE 5-33**

(e)  $X = AB + CD$

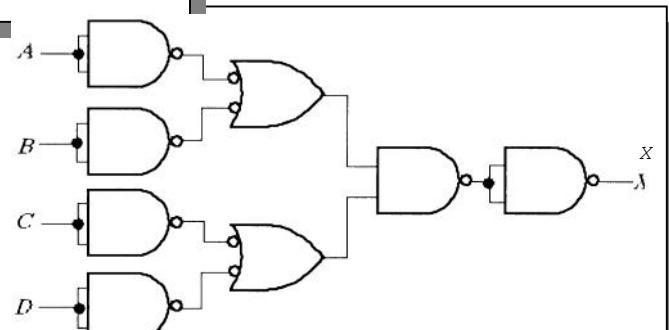
See Figure 5-34.



**FIGURE 5-34**

(f)

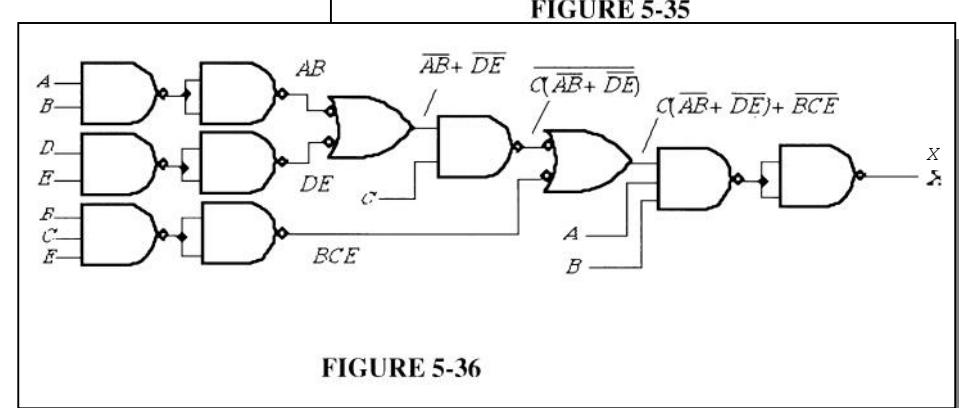
See Figure 5-35.



**FIGURE 5-35**

(g)  $X = AB[C(\overline{DE} + \overline{AB})]$

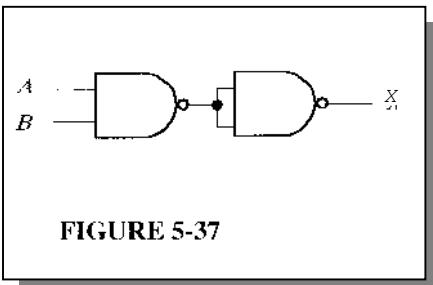
See Figure 5-36.



**FIGURE 5-36**

24. (a)  $X = AB$

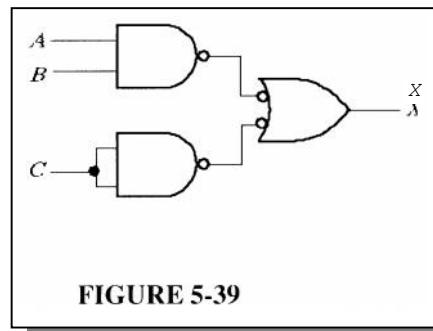
See Figure 5-37.



**FIGURE 5-37**

(c)  $X = AB + C$

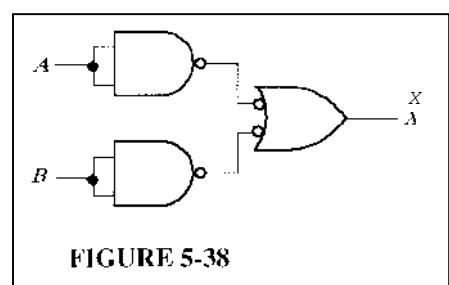
See Figure 5-39.



**FIGURE 5-39**

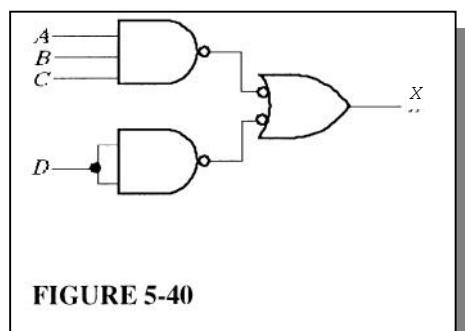
(b)  $X = A + B$

See Figure 5-38.



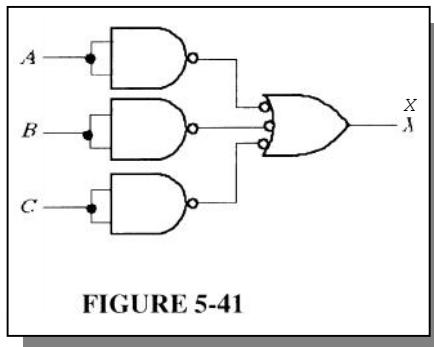
**FIGURE 5-38**

(d) See Figure 5-40.



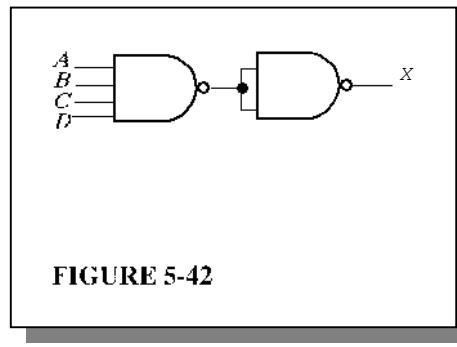
**FIGURE 5-40**

(e)  $X = A + B + C$   
See Figure 5-41.



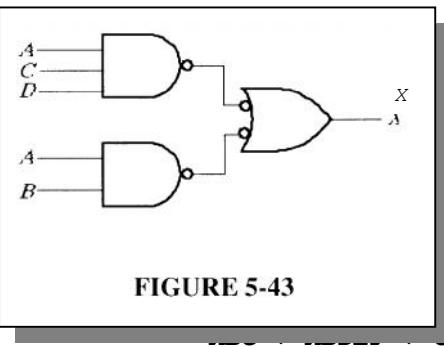
(g)  $X$  **FIGURE 5-41**

(f)  $X = ABCD$   
See Figure 5-42.



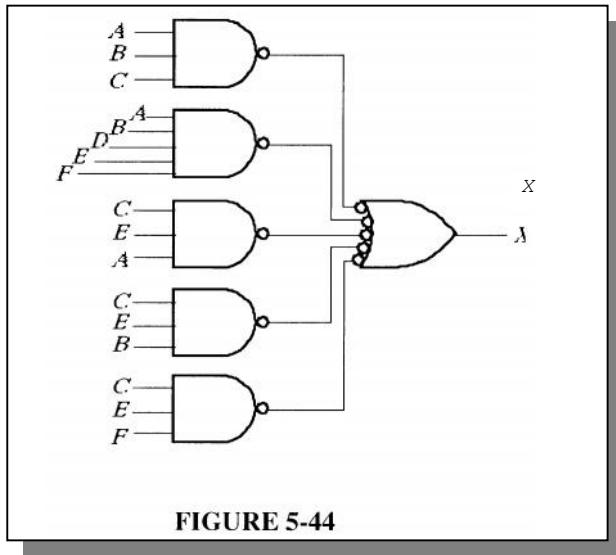
**FIGURE 5-42**

See Figure 5-43.



(h)  $+ F)$   
 $EA + CEB + CEF$  **FIGURE 5-43**

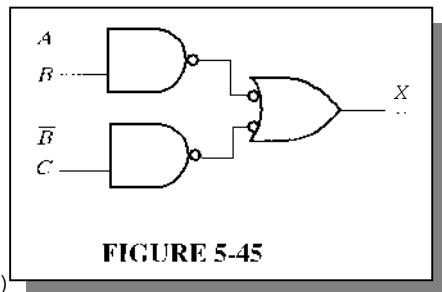
See Figure 5-44.



**FIGURE 5-44**

25. (a)  $X = AB + \overline{B}C$

See Figure 5-45.

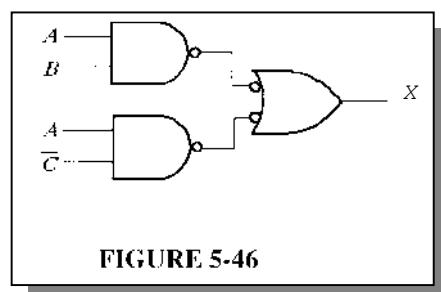


(c)

**FIGURE 5-45**

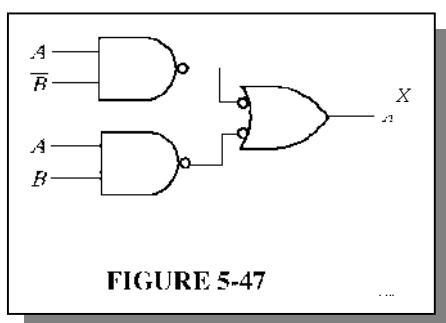
(b)  $X = A(B + \overline{C}) = AB + A\overline{C}$

See Figure 5-46.



**FIGURE 5-46**

See Figure 5-47.

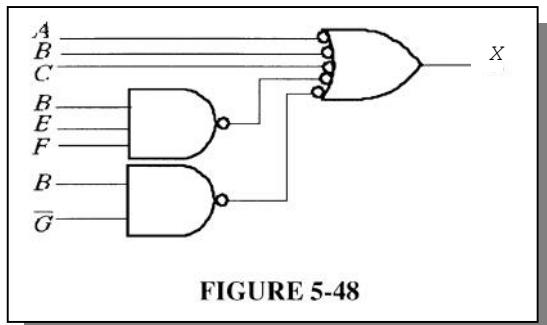


(d)

**FIGURE 5-47**

$$= \overline{C} + BEF + B\overline{G}$$

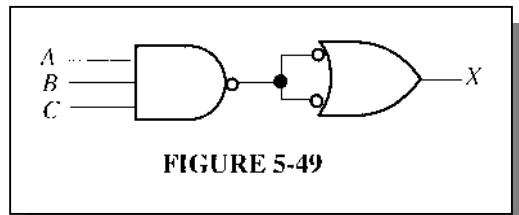
See Figure 5-48.



**FIGURE 5-48**

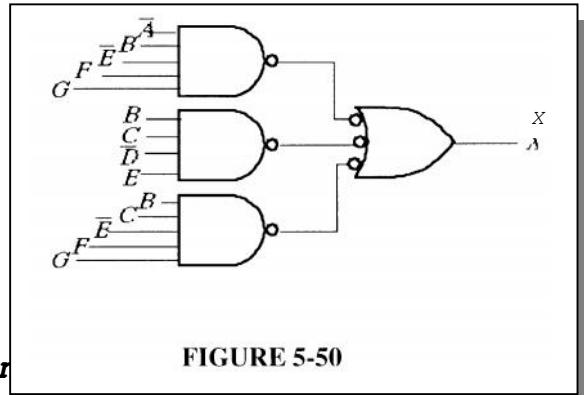
$$(e) \quad X = A[BC(A + B + C + D)] = ABCA + ABCB + ABCC + ABCD \\ = ABC + ABC + ABC + ABCD + ABC(1 + D) = \mathbf{ABC}$$

See Figure 5-49.



$$(f) \quad X = B(CDE + EFG)(AB + C) = B(CDE + \bar{E}FG)(\bar{A} + \bar{B} + C) \\ = B(\bar{A}\bar{C}\bar{D}E + \bar{A}\bar{E}FG + \bar{B}\bar{C}\bar{D}E + \bar{B}\bar{E}FG + C\bar{D}E + C\bar{E}FG) \\ = \bar{A}\bar{B}\bar{E}FG + \bar{B}\bar{B}\bar{E}FG + BC\bar{D}E + BC\bar{E}FG \\ = \bar{A}\bar{B}\bar{E}FG + BC\bar{D}E + BC\bar{E}FG$$

See Figure 5-50.



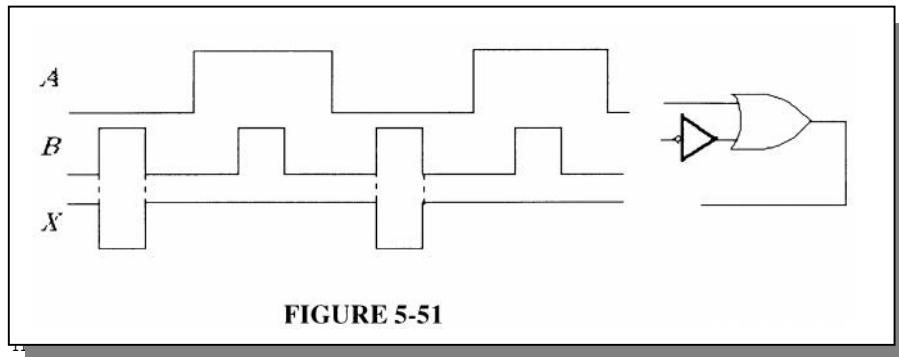
**Section  
Waveform Inputs**

**Section with Pulse**

26.  $X = \overline{\overline{A} + \overline{B} + B} = A\overline{B} = 0$   
The output X is always LOW.

27.  $X = (\overline{AB})B = A + \overline{B} + \overline{B} = A + \overline{B}$

See Figure 5-51.

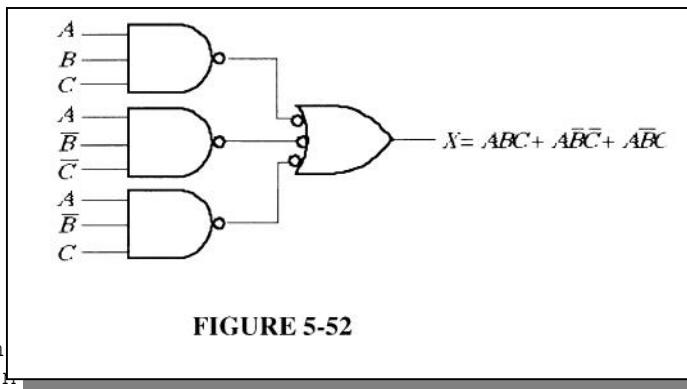


**FIGURE 5-51**

28.  $X$  is ~~LOW or when A is HIGH and B is LOW and C is HIGH.~~  $X$  is LOW or when  $A$  is HIGH and  $B$  is LOW and  $C$  is HIGH.

$$X = ABC + A\bar{B}\bar{C} + A\bar{B}C$$

See Figure 5-52.

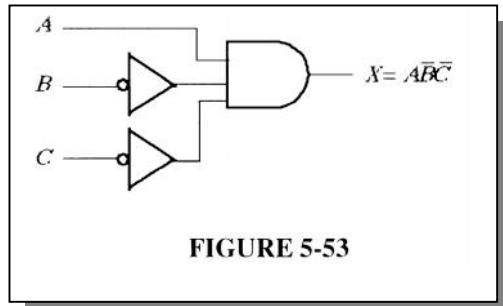


**FIGURE 5-52**

29.  $X$  is HIGH when ~~A is HIGH when X is HIGH when~~  $X$  is HIGH when ~~A is HIGH when X is HIGH when~~ not know if

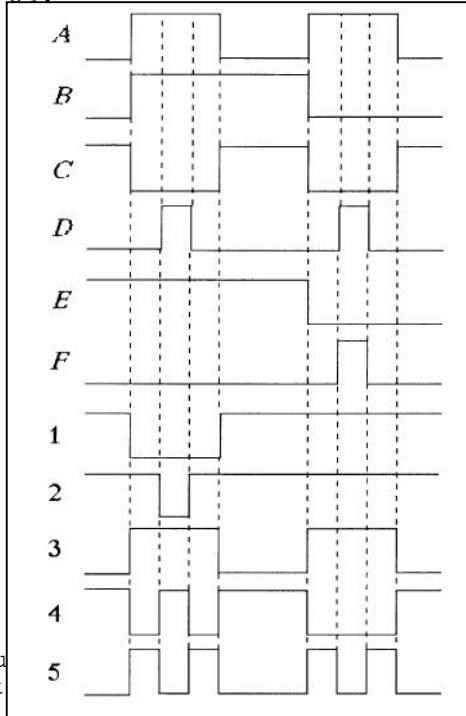
$$X = A\bar{B}\bar{C}$$

See Figure 5-53.

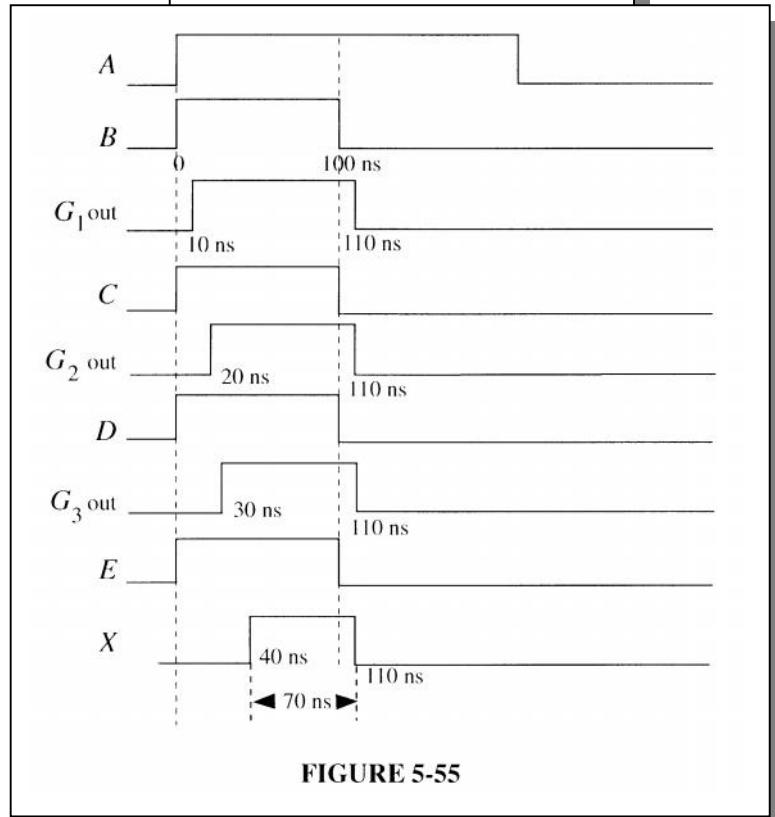


**FIGURE 5-53**

30. See Figure 5-54.



31. The output pulse width maximum is not greater than 25 ns. A



**FIGURE 5-55**

### Section 5-6 Combinational Logic with VHDL

```

32. entity Circuit5_51b is
      port (A, B, C, D: in bit; X: out bit);
    end entity Circuit5_51b;
    architecture LogicFunction of Circuit5_51b is
    begin
      X <= not(not A and B) or (not A and C and D) or (D and B and
not D);
    end architecture LogicFunction;

33. (e)   entity Circuit5_52e is
      port (A, B, C: in bit; X: out bit);
    end entity Circuit5_52e;
    architecture LogicFunction of Circuit5_52e is
    begin
      X <= (not A and B) or B or (B and not C) or (not A and not C) or (B
and not C) or not C;
    end architecture LogicFunction;

(f)   entity Circuit5_52f is
      port (A, B, C: in bit; X: out bit);
    end entity Circuit5_52f;
    architecture LogicFunction of Circuit5_52f is
    begin
      X <= (A or B) and (not B or C);
    end architecture Logic Function;

34. See Figure 5-56 for input/output, gate, and signal labeling.

```

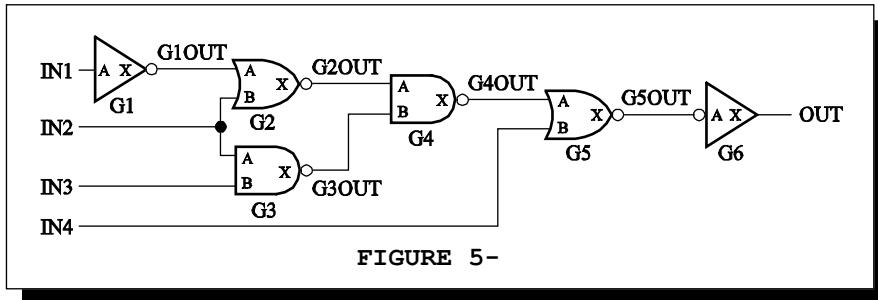


FIGURE 5-

--Program for the logic circuit in Figure 5-56 (textbook Figure 5-  
53(d))

```

entity (Circuit5_53d is
  port (IN1, IN2, IN3, IN4: in bit; OUT: out bit);
end entity Circuit5_53d;
architecture LogicOperation of Circuit5_53d is
--Component declaration for inverter
component Inverter is
  port (A: in bit; X: out bit);
end component Inverter;
--Component declaration for NOR gate
component NORgate is
  port (A, B: in bit; X: out bit);
end component NORgate;
--Component declaration for NAND gate
component NANDgate is
  port (A, B: in bit; X: out bit);
end component NANDgate;
signal G1OUT, G2OUT, G3OUT, G4OUT, G5OUT: bit;
begin
  G1: Inverter port map (A => IN1, X => G1OUT);

```

```

G2: NORgate port map (A => G1OUT, B => IN2, X => G2OUT);
G3: NAND gate port map (A => IN2, B => IN3, X => G3OUT);
G4: NANDgate port map (A => G2OUT, B => G3OUT, X => G4OUT);
G5: NORgate port map (A => G4OUT, B => IN4, X => G5OUT);
G6: Inverter port map (A => G5OUT, X => OUT);
end architecture LogicOperation;

```

35. See Figure 5-57 for input/output, gate, and signal labeling.

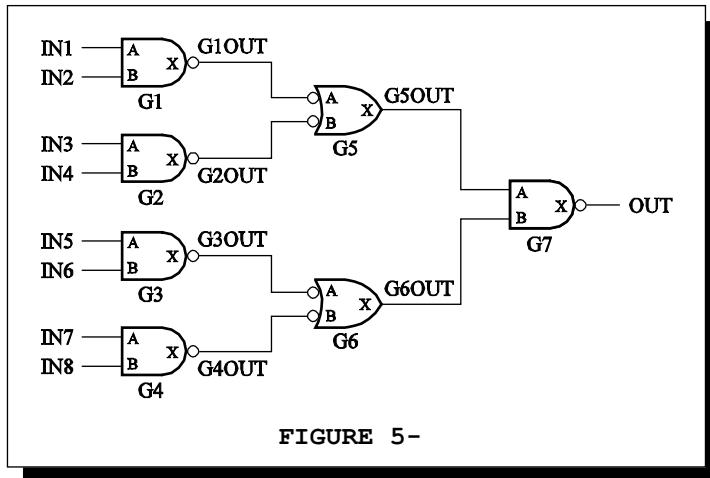


FIGURE 5-

--Program for the logic circuit in Figure 5-57 (textbook Figure 5-53(f))

```

entity Circuit5_53f is
    port (IN1, IN2, IN3, IN4, IN5, IN6, IN7, IN8: in bit; OUT:
out bit);
end entity Circuit5_53f;
architecture LogicFunction of Circuit5_53f is
    --Component declaration for NAND gate
    component NANDgate is
        port (A, B: in bit; X: out bit);
    end component NANDgate;
    signal G1OUT, G2OUT, G3OUT, G4OUT, G5OUT, G6OUT: bit;
    begin
        G1: NANDgate port map (A => IN1, B => IN2, X => G1OUT);
        G2: NANDgate port map (A => IN3, B => IN4, X => G2OUT);
        G3: NANDgate port map (A => IN5, B => IN6, X => G3OUT);
        G4: NANDgate port map (A => IN7, B => IN8, X => G4OUT);
        G5: NANDgate port map (A => G1OUT, B => G2OUT, X => G5OUT);
        G6: NANDgate port map (A => G3OUT, B => G4OUT, X => G6OUT);
        G7: NANDgate port map (A => G5OUT, B => G6OUT, X => OUT);
    end architecture LogicFunction;

```

36.  $X = \overline{ABC} + \overline{ABC} + \overline{ABC} + \overline{ABC} + ABC$

This is the SOP expression for the function in Table 5-8 of the textbook. The following program applies the data flow approach for this logic function.

```

--Program for Table5_8 SOP logic
entity Table5_8 is
    port (A, B, C: in bit; X: out bit);
end entity Table5_8;
architecture LogicOperation of Table5_8 is
    begin
        X <= (not A and not B and not C) or (not A and B and not C)
            or (A and not B and not C) or (A and B and not C) or (A
            and B and C);
    end begin

```

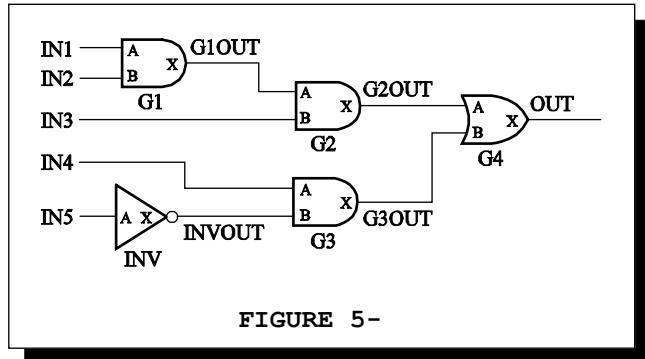
```

end architecture LogicOperation;

37. --Program for textbook Figure5_64 data flow approach
entity Fig5_64 is
    port (A, B, C, D, E:  in bit; X: out bit);
end entity Fig5_64;
architecture DataFlow of Fig5_64 is
begin
    X <= (A and B and C) or (D and not E)
end architecture DataFlow;

```

See Figure 5-58 for the circuit in textbook Figure 5-64 modified for the structural approach.



**FIGURE 5-**

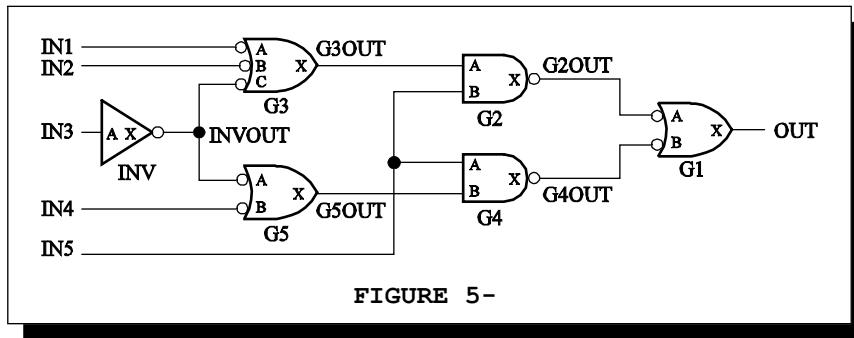
```

--Program for textbook Figure5_64 structural approach
entity Fig5_64 is
    port (IN1, IN2, IN3, IN4, IN5:  in bit; OUT: out bit);
end entity Fig5_64;
architecture Structure of Fig5_64 is
    --Component declaration for AND gate
    component AND_gate is
        port (A, B: in bit; X: out bit);
    end component AND_gate;
    --Component declaration for OR gate
    component OR_gate is
        port (A, B: in bit; X: out bit);
    end component OR_gate;
    --Component declaration for Inverter
    component Inverter is
        port (A: in bit; X: out bit);
    end component Inverter;
    signal G1OUT, G2OUT, G3OUT, INVOUT: bit;
begin
    G1: AND_gate port map (A => IN1, B => IN2, X => G1OUT);
    G2: AND_gate port map (A => G1OUT, B => IN3, X => G2OUT);
    INV: Inverter port map (A => IN5, X => INVOUT);
    G3: AND_gate port map (A => IN4, B => INVOUT, X => G3OUT);
    G4: OR_gate port map (A => G2OUT, B => G3OUT, X => OUT);
end architecture Structure;

38. --Program for textbook Figure5_68 data flow approach
entity Fig5_68 is
    port (A, B, C, D, E:  in bit; X: out bit);
end entity Fig5_68;
architecture DataFlow of Fig5_68 is
begin
    X <= (not A or not B or C) and E or (C or not D) and E;
end architecture DataFlow;

```

See Figure 5-59 for the circuit in textbook Figure 5-68 labeled for the structural approach.



**FIGURE 5-**

```
--Program for textbook Fig5_68 structural approach
entity Fig5_68 is
    port (IN1, IN2, IN3, IN4, IN5:  in bit; OUT: out bit);
end entity Fig5_68;
architecture Structure of Fig5_68 is
--Component declaration for 3-input NAND gate
component NAND_gate3 is
    port (A, B, C: in bit; X: out bit);
end component NAND_gate3;
--Component declaration for 2-input NAND gate
component NAND_gate2 is
    port (A, B: in bit; X: out bit);
end component NAND_gate2;
--Component declaration for Inverter
component Inverter is
    port (A: in bit; X: out bit);
end component Inverter;
signal G2OUT, G3OUT, G4OUT, G5OUT, INVOUT: bit;
begin
    G1:  NAND_gate2 port map (A => G2OUT, B => G4OUT, X => OUT);
    G2:  NAND_gate2 port map (A => G3OUT, B => IN5, X => G2OUT);
    INV: Inverter port map (A => IN3, X => INVOUT);
    G3:  NAND_gate3 port map (A => IN1, B => IN2, C => INVOUT, X =>
G3OUT);
    G4:  NAND_gate2 port map (A => IN5, B => G5OUT, X => G4OUT);
    G5:  NAND_gate2 port map (A => INVOUT, B => IN4, X => G5OUT);
end architecture Structure;
```

**39.** From the VHDL program, the logic expression is stated as a Boolean expression as follows:

$$\begin{aligned} X &= (\overline{AB} + \overline{AC} + \overline{AD} + \overline{BC} + \overline{BD} + \overline{DC}) \\ &= ((A+B)(A+C)(A+D)(B+C)(B+D)(D+C)) \\ &= (A+B)(A+C)(A+D)(B+C)(B+D)(D+C) \end{aligned}$$

The truth table is:

| A | B | C | D | X |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 |

|   |   |   |   |   |
|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 |

```
40. --Program for textbook Figure5_62 data flow approach
entity Fig5_62 is
    port (A1, A2, B1, B2:  in bit; X: out bit);
end entity Fig5_62;
architecture LogicCircuit of Fig5_62 is
begin
    X <= (A1 and A2) or (A2 and not B1) or (not B1 and not B2) or
(not B2 and A1);
end architecture LogicCircuit;
```

- 41.** The AND gates are numbered top to bottom G<sub>1</sub>, G<sub>2</sub>, G<sub>3</sub>, G<sub>4</sub>. The OR gate is G<sub>5</sub> and the inverters are, top to bottom. G<sub>6</sub> and G<sub>7</sub>. Change A<sub>1</sub>, A<sub>2</sub>, B<sub>1</sub>, B<sub>2</sub> to IN1, IN2, IN3, IN4 respectively. Change X to OUT.

```

entity Circuit5_62 is
    port (IN1, IN2, IN3, (IN4: in bit; OUT: out bit);
end entity Circuit5_62;
architecture Logic of Circuit5_62 is
    component AND_gate is
        port (A, B: in bit; X: out bit);
    end component AND_gate;
    component OR_gate is
        port (A, B, C, D: in bit; X: out bit);
    end component OR_gate;
    component Inverter is
        port (A: in bit; X: out bit);
    end component Inverter;
    signal G1OUT, G2OUT, G3OUT, G4OUT, G5OUT, G6OUT, G7OUT: bit;
begin
    G1: AND_gate port map (A => IN1, B => IN2, X => G1OUT);
    G2: AND_gate port map (A => IN2, B => G6OUT, X => G2OUT);
    G3: AND_gate port map (A => G6OUT, B => G7OUT, X => G3OUT);
    G4: AND_gate port map (A => G7OUT, B => IN1, X => G4OUT);
    G5: OR_gate port map (A => G1OUT, B => G2OUT, X => G3OUT,
                           D => G4OUT, X => OUT);
    G6: Inverter port map (A => IN3, X => G6OUT);
    G7: Inverter port map (A => IN4, X => G7OUT);
end architecture Logic;

```

### Section 5-7 Troubleshooting

**42.**  $X = \overline{AB} + \overline{CD} = ABCD$

X is HIGH only when ABCD are all HIGH. This does not occur in the waveforms, so X should remain LOW. **The output is incorrect.**

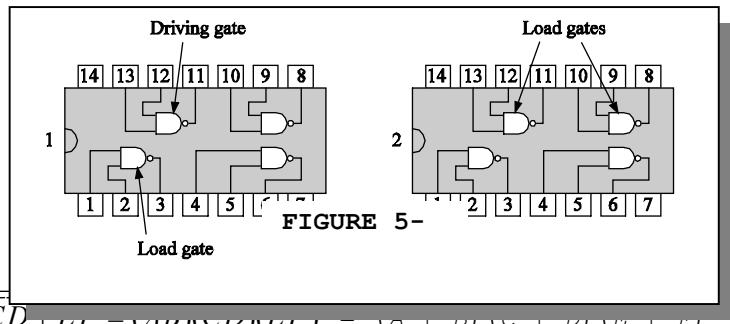
**43.**  $X = ABC + D\bar{E}$

Since X is the same as the G<sub>3</sub> output, either G<sub>1</sub> or G<sub>2</sub> has failed with its output stuck LOW.

**44.**  $X = AB + CD + EF$

X does not go HIGH when C and D are HIGH. G<sub>2</sub> has failed with the output open or stuck HIGH or the corresponding input to G<sub>4</sub> is open.

45. See Figure 5-60.

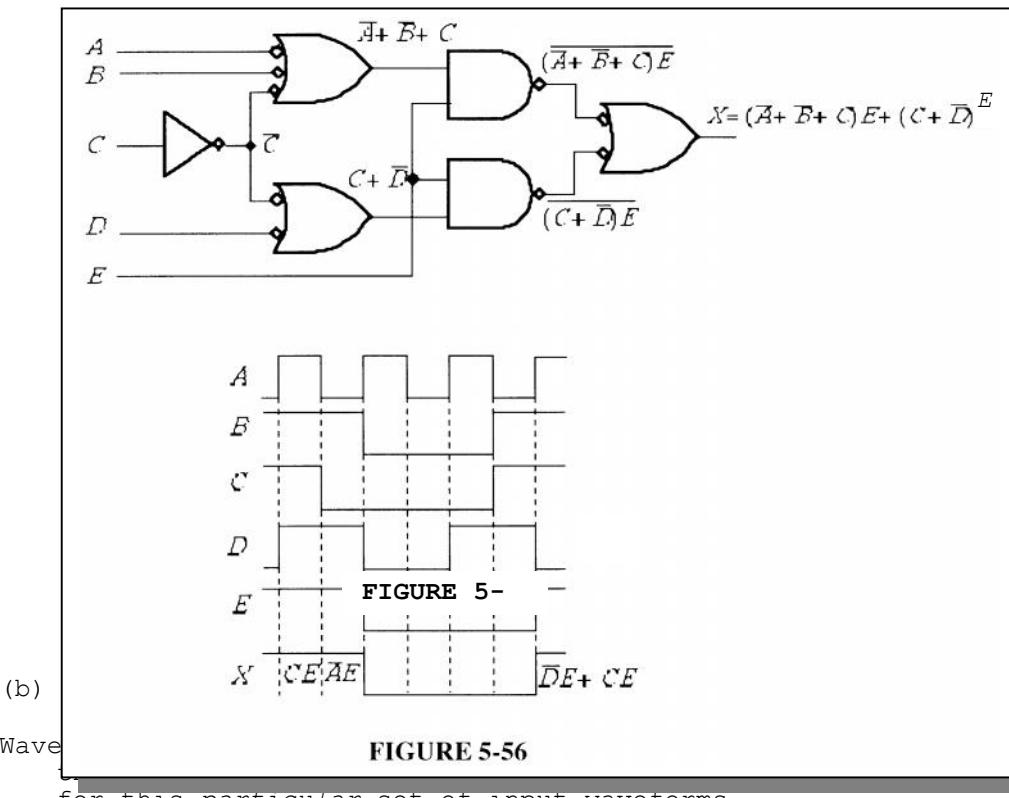


46.  $X = \overline{AB} + \overline{CD} + \overline{EF} = (\overline{AB})(\overline{CD})(\overline{EF}) = (\overline{A} + \overline{B})(\overline{C} + \overline{D})(\overline{E} + \overline{F})$

Since  $X$  does not go HIGH when  $C$  or  $D$  is HIGH, the output of gate  $G_2$  must be *stuck LOW*.

47. (a) 
$$\begin{aligned} X &= (\overline{A} + \overline{B} + C)E + (C + \overline{D})E = \overline{A}E + \overline{B}E + \cancel{CE} + CE + \overline{DE} \\ &= \overline{A}E + \overline{B}E + CE + \overline{DE} \end{aligned}$$

See Figure 5-61.



Wave  
for this particular set of input waveforms.

is up

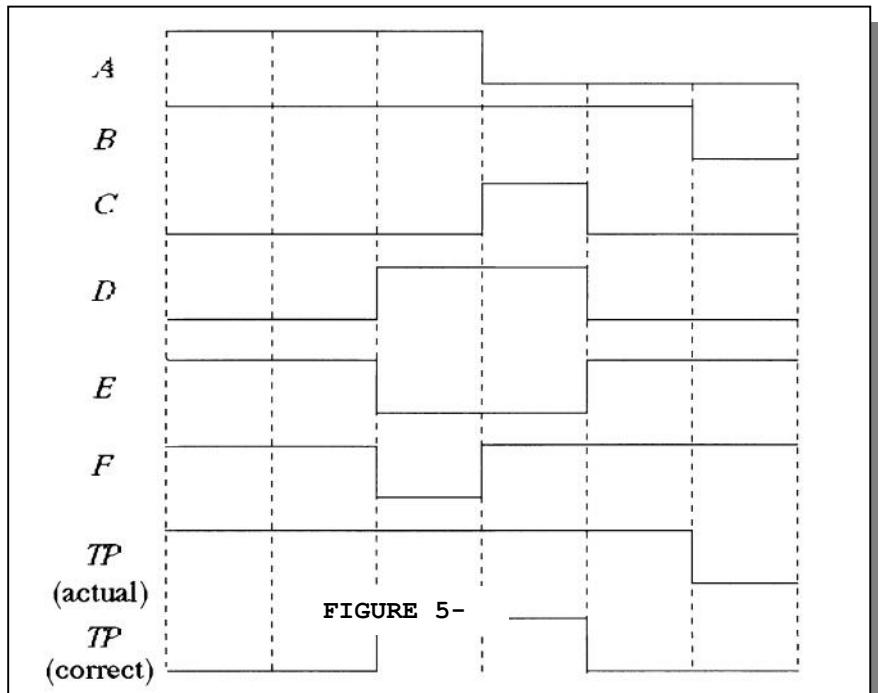
$$(c) \quad X = E + E(\bar{A} + \bar{B} + C) = E(1 + \bar{A} + \bar{B} + C) = E$$

Again waveform  $X$  is the same as waveform  $E$ . As strange as it may seem, the shorted input to  $G_5$  does not affect the output for this particular set of input waveforms.

Conclusion: the two faults are not indicated in the output waveform for these particular inputs.

48.  $TP = \overline{\overline{AB}} + \overline{\overline{CD}}$

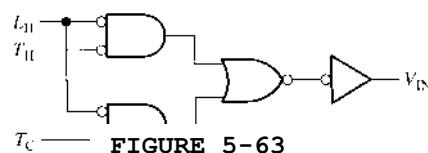
The output of the  $\overline{\overline{CD}}$  gate is stuck LOW. See Figure 5-62.



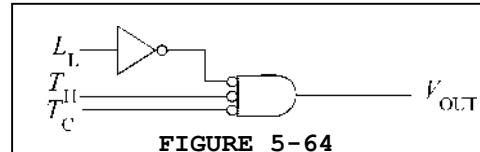
**Digital**

49. See Figure

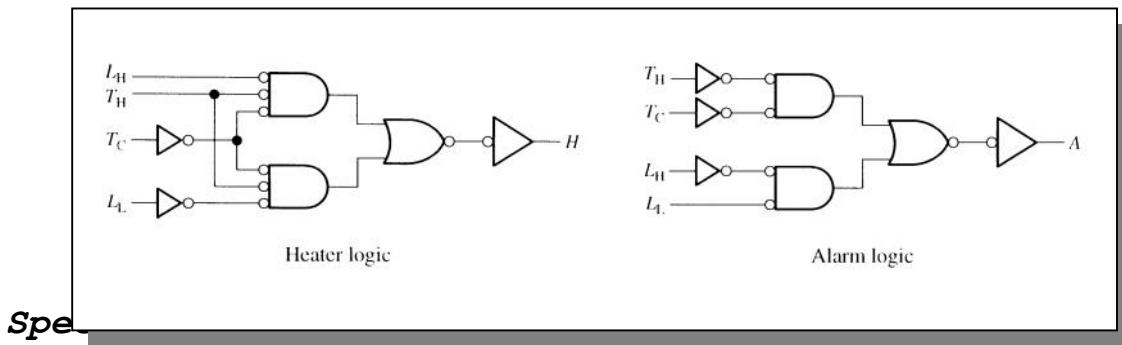
**FIGURE 5-57**



50. See Figure



51. See Figure

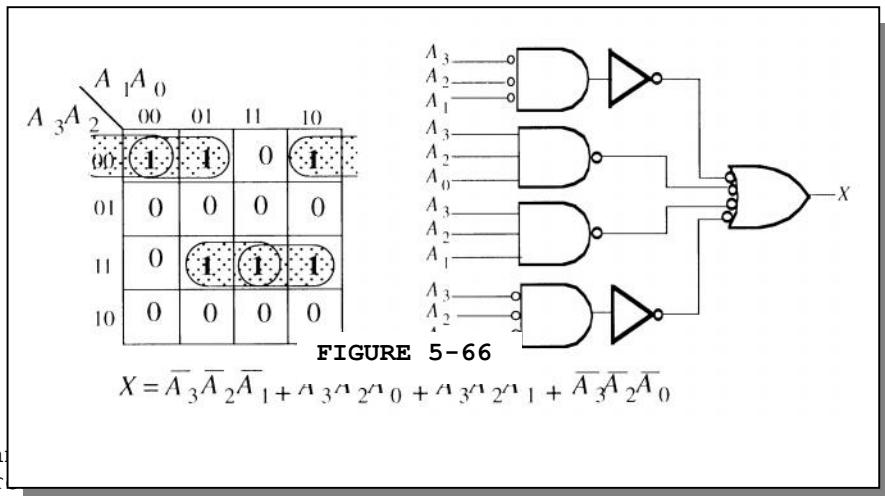


*Spec*

52.

| $A_3$ | $A_2$ | $A_1$ | $A_0$ | $X$ |
|-------|-------|-------|-------|-----|
| 0     | 0     | 0     | 0     | 1   |
| 0     | 0     | 0     | 1     | 1   |
| 0     | 0     | 1     | 0     | 1   |
| 0     | 0     | 1     | 1     | 0   |
| 0     | 1     | 0     | 0     | 0   |
| 0     | 1     | 0     | 1     | 0   |
| 0     | 1     | 1     | 0     | 0   |
| 0     | 1     | 1     | 1     | 0   |
| 1     | 0     | 0     | 0     | 0   |
| 1     | 0     | 0     | 1     | 0   |
| 1     | 0     | 1     | 0     | 0   |
| 1     | 0     | 1     | 1     | 0   |
| 1     | 1     | 0     | 0     | 0   |
| 1     | 1     | 0     | 1     | 1   |
| 1     | 1     | 1     | 0     | 1   |
| 1     | 1     | 1     | 1     | 1   |

See Figure 5-66.



53. Let  
 $X = \text{La}$   
 $A = \text{Fr}$

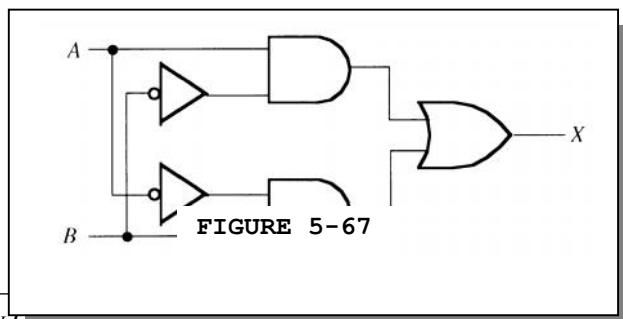
$\overline{A}$  = Front door switch off

$B$  = Back door switch on

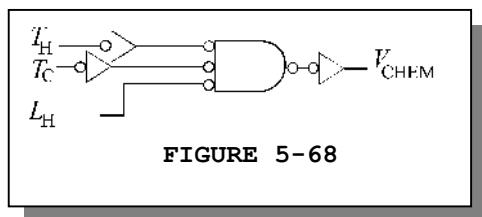
$\overline{B}$  = Back door switch off

$X = A\overline{B} + \overline{A}B$ . This is an XOR operation.

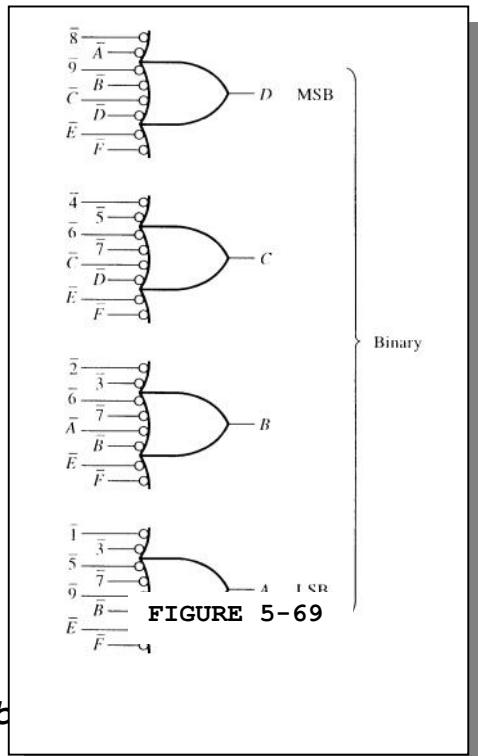
See Figure 5-67.



54.  $V_{\text{CHEM}} = T_H L_H$



**55.** See Figure 5-69.



**Multisim Troubleshooting**

- 56.** Pin B of G1 open.
- 57.** Pin C of OR gate open.
- 58.** Inverter input open.
- 59.** No fault.

---

## CHAPTER 6

### FUNCTIONS OF COMBINATIONAL LOGIC

---

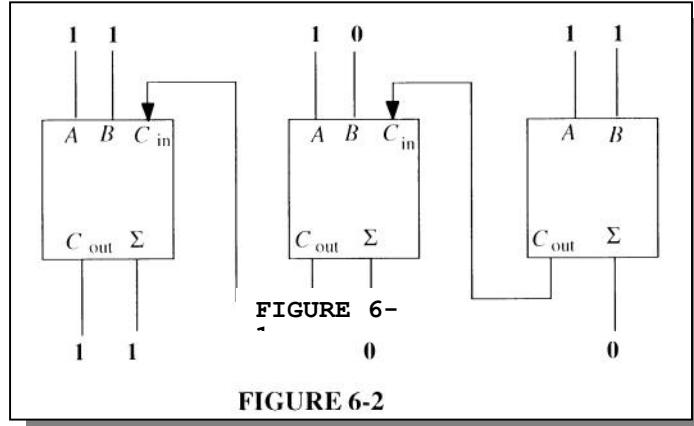
#### Section 6-1 Basic Adders

1. (a) XOR (upper) output = 0, Sum output = 1, AND (upper) output = 0,  
AND (lower) output = 1, Carry output = 1  
(b) XOR (upper) output = 1, Sum output = 0, AND (upper) output = 1,  
AND (lower) output = 0, Carry output = 1  
(c) XOR (upper) output = 1, Sum output = 1, AND (upper) output = 0,  
AND (lower) output = 0, Carry output = 0
2. (a)  $A = 0, B = 0, C_{in} = 0$   
(b)  $A = 1, B = 0, C_{in} = 0$  or  $A = 0, B = 1, C_{in} = 0$   
 $\text{or } A = 0, B = 0, C_{in} = 1$   
(c)  $A = 1, B = 1, C_{in} = 1$   
(d)  $A = 1, B = 1, C_{in} = 0$  or  $A = 0, B = 1, C_{in} = 1$   
 $\text{or } A = 1, B = 0, C_{in} = 1$
3. (a)  $\Sigma = 1, C_{out} = 0$  (b)  $\Sigma = 1, C_{out} = 0$   
(c)  $\Sigma = 0, C_{out} = 1$  (d)  $\Sigma = 1, C_{out} = 1$

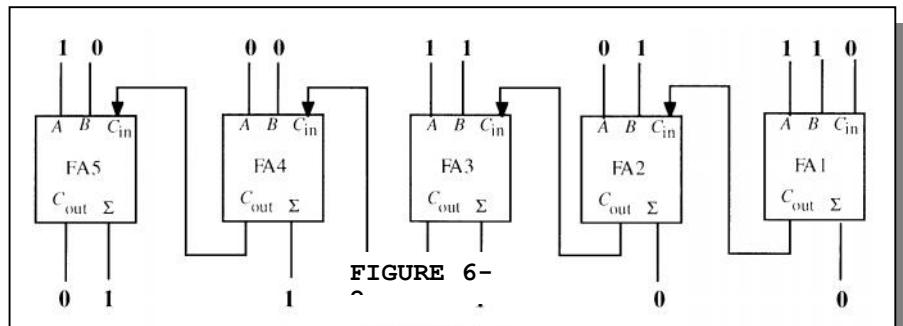
#### Section 6-2 Parallel Binary Adders

4. See Figure 6-1.

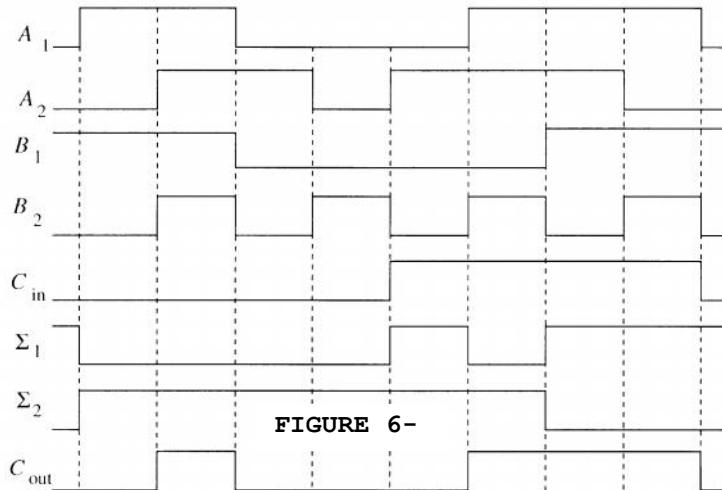
111  
101  
1100



5. 
$$\begin{array}{r} 10101 \\ 00111 \\ \hline 11100 \end{array}$$
 See Figure 6-2.



6. See Figure 6-4



7.

| $A_4$ | $A_3$ | Σ |   |   |   |   |   |   |   |   |   | Cout |   |   |   |   |
|-------|-------|---|---|---|---|---|---|---|---|---|---|------|---|---|---|---|
| 1     | 0     | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0    | 0 | 1 | 0 | 1 |
| 1     | 0     | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1    | 0 | 0 | 1 | 0 |
| 0     | 0     | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0    | 1 | 0 | 1 | 1 |
| 1     | 0     | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0    | 1 | 0 | 1 | 0 |

$$\Sigma_1 = 0110$$

$$\Sigma_2 = 1011$$

$$\Sigma_3 = 0110$$

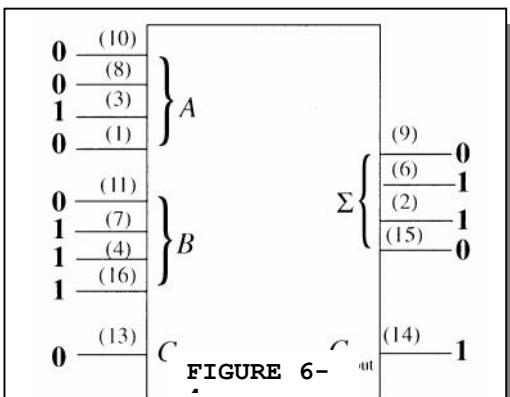
$$\Sigma_4 = 0001$$

$$\Sigma_5 = 1000$$

$$8. \quad \begin{array}{r} 0100 \\ 1110 \\ \hline 10010 \end{array}$$

$\Sigma$  outputs should be  $C_{\text{out}}\Sigma_4\Sigma_3\Sigma_2\Sigma_1 = 10010$ .  
The  $\Sigma_3$  output (pin 2) is HIGH and should be LOW.

See Figure 6-4.



### Section 6-3 Ripple Adders

FIGURE 6-4

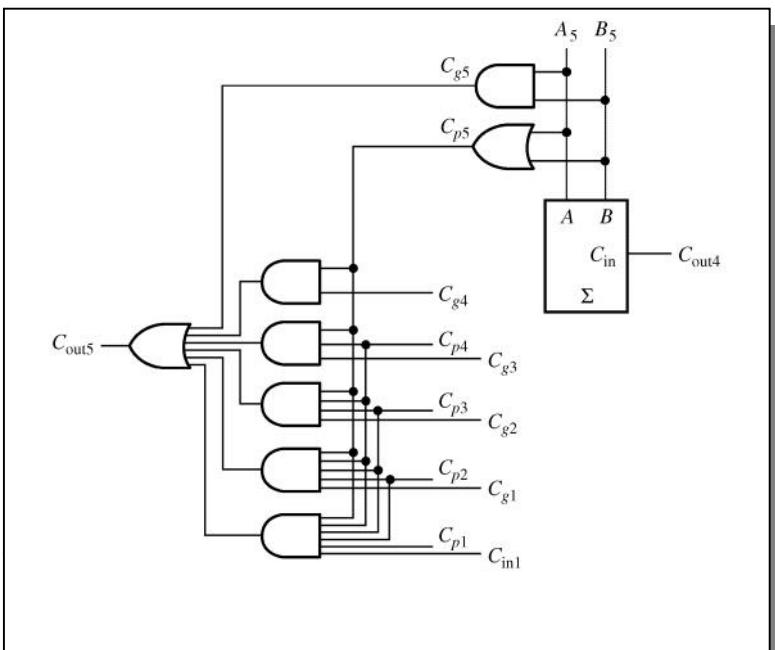
### Ahead Carry

$$9. \quad t_{p(\text{tot})} = 40 \text{ ns} + 6(25 \text{ ns}) + 35 \text{ ns} = 225 \text{ ns}$$

10. Full-adder 5:

$$\begin{aligned} C_{\text{in}5} &= C_{\text{out}4} \\ C_{\text{out}5} &= C_{g5} + C_{p5}C_{g4} + C_{p5}C_{p4}C_{g3} + C_{p5}C_{p4}C_{p3}C_{g2} + C_{p5}C_{p4}C_{p3}C_{g2}C_{g1} + \\ &\quad C_{p5}C_{p4}C_{p3}C_{p2}C_{p1}C_{\text{in}1} \end{aligned}$$

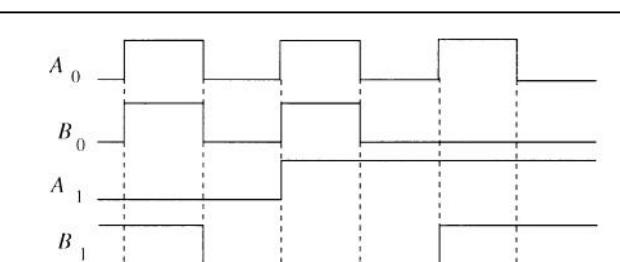
The logic to be added to text Figure 6-18 is shown in Figure 6-5.



### Section

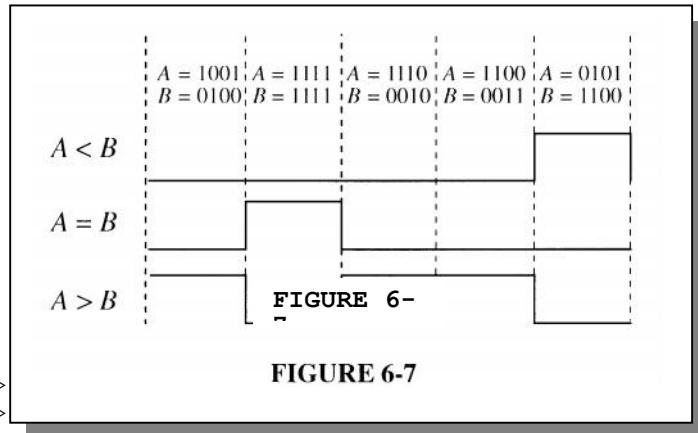
11. The A

See Figure 6-6.



**FIGURE 6-**

12. See Figure 6-7.



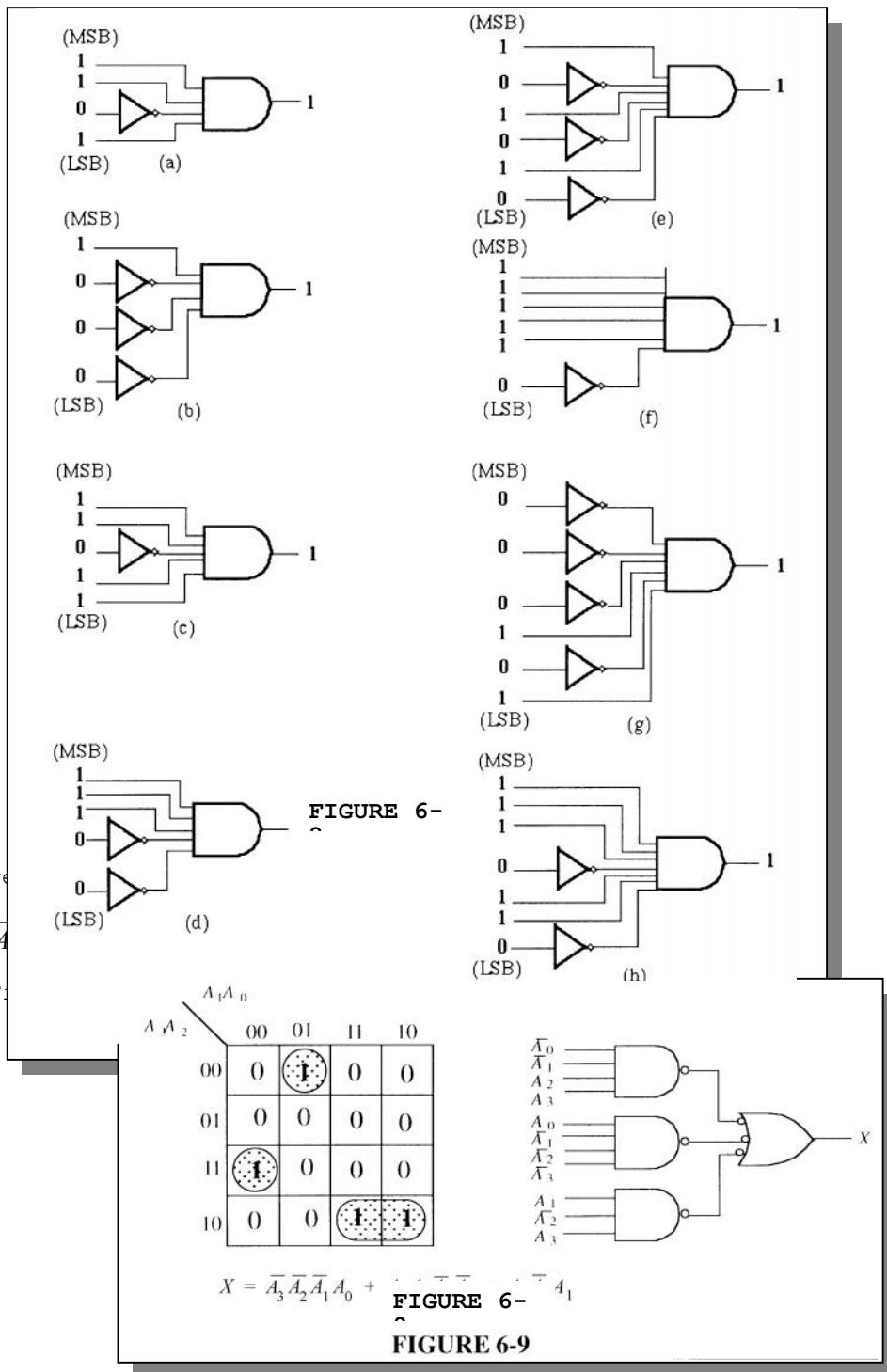
13. (a)  $A >$   
 (b)  $A >$   
 (c)  $A > B: 0, A = B: 1, A < B: 0$

**FIGURE 6-7**

### **Section 6-5 Decoders**

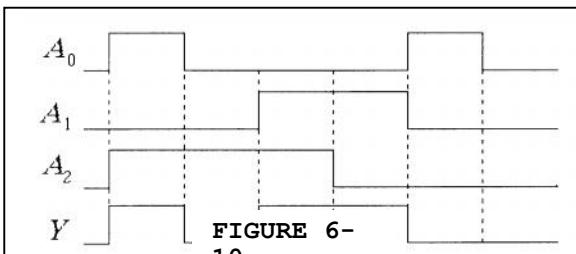
- |                               |                           |
|-------------------------------|---------------------------|
| 14. (a) $A_3A_2A_1A_0 = 1110$ | (b) $A_3A_2A_1A_0 = 1100$ |
| (c) $A_3A_2A_1A_0 = 1111$     | (d) $A_3A_2A_1A_0 = 1000$ |

15. See Figure 6-8.



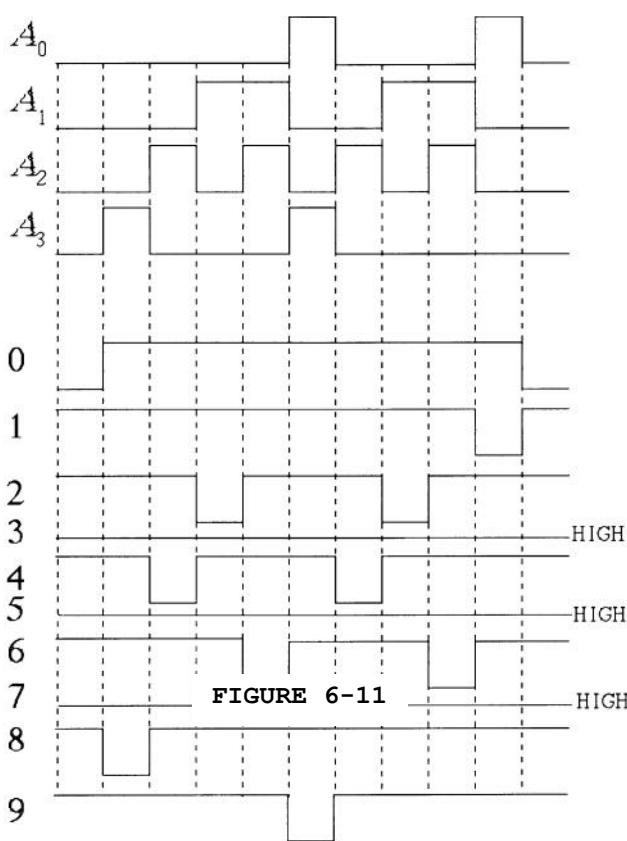
18.  $Y = A_2 A_1 \overline{A_0} + A_2 \overline{A_1} A_0 + \overline{A_2} A_1 \overline{A_0}$

See Figure 6-10.



19. See Figure 6-11.

**FIGURE 6-10**



20. 0 1 6 9

### Section 6-6

21.  $A_0$ ,  $A_1$ , and code.

valid BCD

22. Pin 2 is for decimal 5, pin 5 is for decimal 8, and pin 12 is for decimal 2.

The highest priority input is pin 5.

The completed outputs are:  $\overline{A_3} \overline{A_2} \overline{A_1} \overline{A_0} = 0111$ , which is binary 8 (1000).

### Section 6-7 Code Converters

23. (a)  $2_{10} = 0010_{BCD} = 0010_2$

(b)  $8_{10} = \text{1000}_{\text{BCD}} = \text{1000}_2$

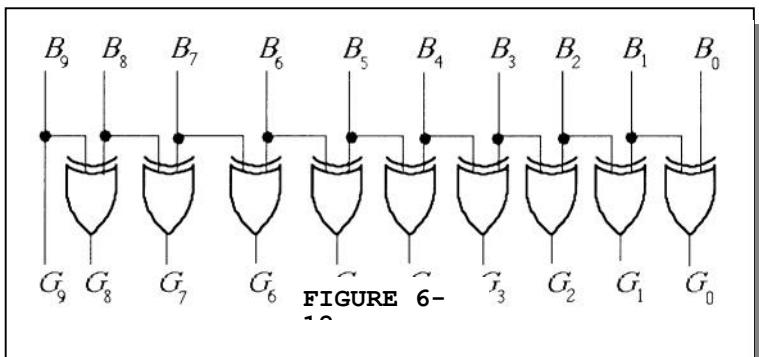
(c)  $13_{10} = \text{00010011}_{\text{BCD}} = \text{1101}_2$

(d)  $26_{10} = \text{00100110}_{\text{BCD}} = \text{11010}_2$

(e)  $33_{10} = \text{00110011}_{\text{BCD}} = \text{100001}_2$

|            |     |                          |                |     |                          |                |
|------------|-----|--------------------------|----------------|-----|--------------------------|----------------|
| <b>24.</b> | (a) | 1010101010<br>1111111111 | binary<br>gray | (b) | 1111100000<br>1000010000 | binary<br>gray |
|            | (c) | 0000001110<br>0000001001 | binary<br>gray | (d) | 1111111111<br>1000000000 | binary<br>gray |

See Figure 6-12.

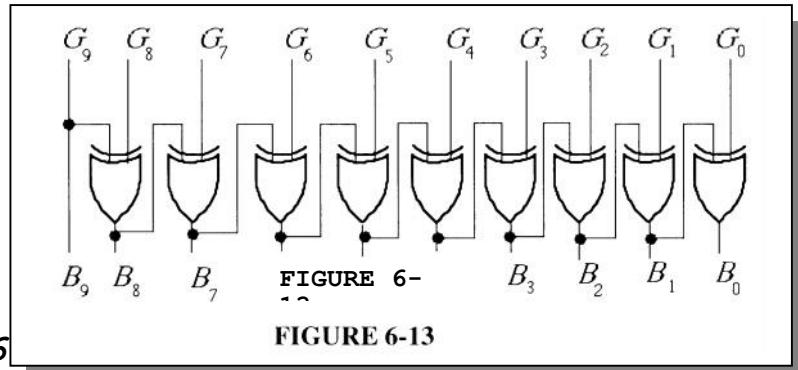


**FIGURE 6-12**

|            |     |                          |                |                                 |
|------------|-----|--------------------------|----------------|---------------------------------|
| <b>25.</b> | (a) | 10<br>110                | 00<br>00       | gray<br>binary                  |
|            | (c) | 1111000111<br>1010000101 | gray<br>binary | (d)<br>0000000001<br>0000000001 |

See Figure 6-13.

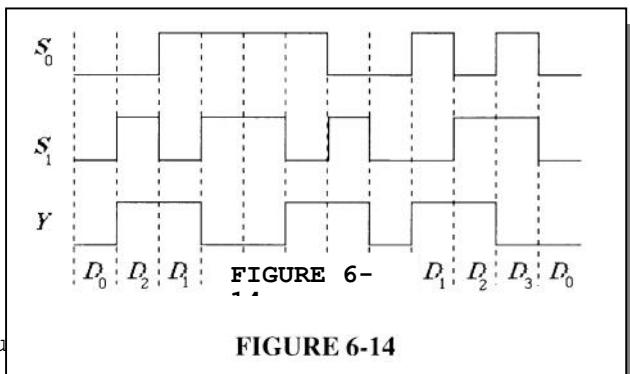
**Section 6**



**FIGURE 6-13**

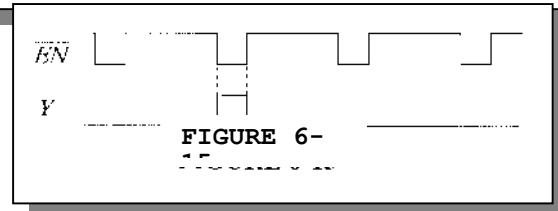
26.  $S_1S_0 = 01$  selects,  $D_1$ , therefore  $Y = 1$ .

27. See Figure 6-14.



**FIGURE 6-14**

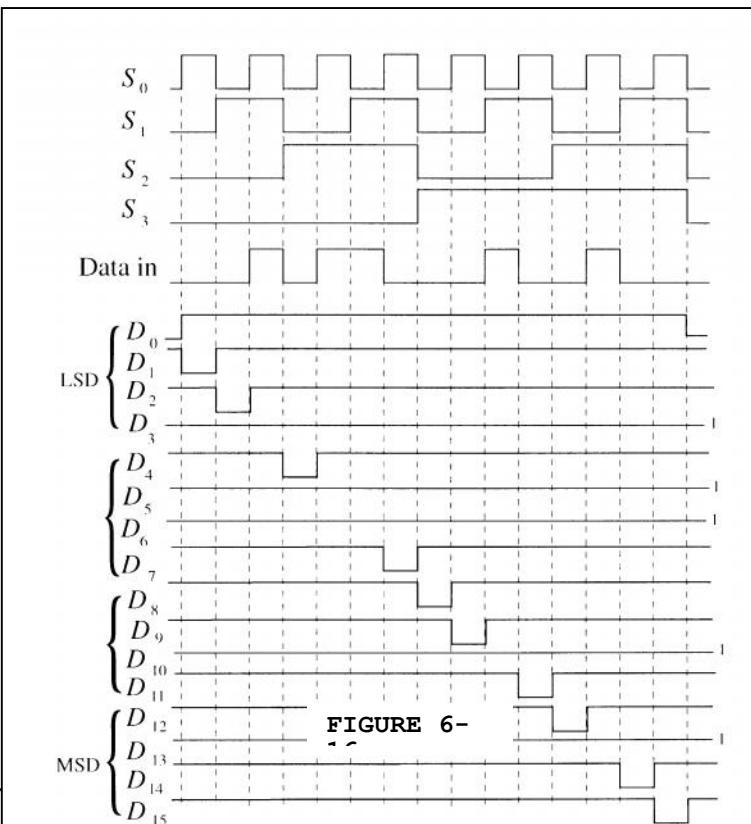
28. See Figure 6-



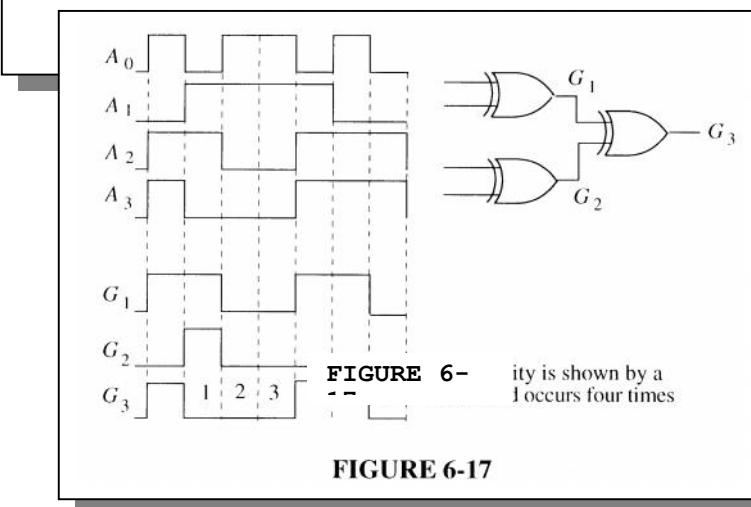
**Section 6-9 Demultiplexers**

29. See Figure 6-16.

**Section 6**

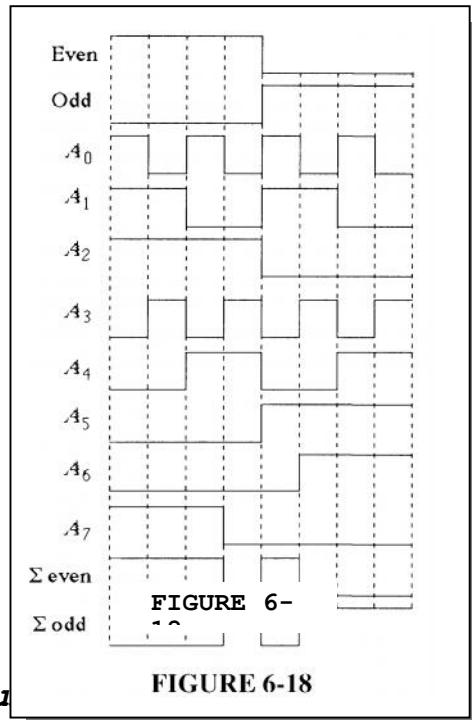


30. See Fig.



**FIGURE 6-17**

31. See Figure 6-18.



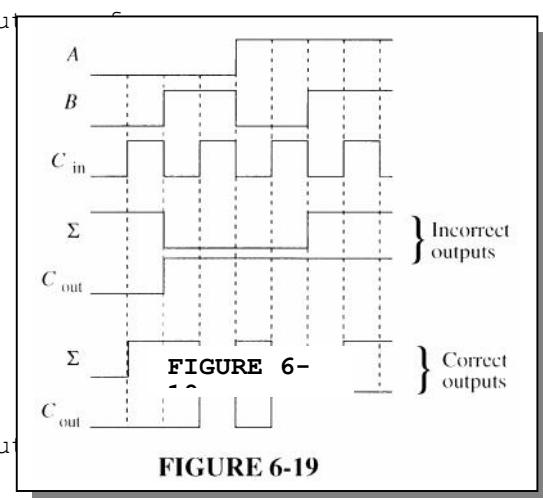
**Section 6-11 Troubleshooting**

**FIGURE 6-18**

32. The outputs given in the problem are incorrect. By observation of these incorrect waveforms, we can conclude that the outputs of the device are not open or shorted because both waveforms are changing.

Observe that at the beginning of the timing diagram all inputs are 0 but the sum is 1. This indicates that an input is stuck HIGH. Start by assuming that  $C_{in}$  is stuck HIGH. This results in  $\Sigma$  and  $C_{out}$  output waveforms that match the waveforms given in the problem, indicating that  $C_{in}$  is indeed stuck HIGH, perhaps shorted to  $V_{cc}$ .

See Figure 6-19 for the correct outputs.



33. (a) OK (b) Segment  $g$  burned out  
Segment  $b$  output stuck LOW

34. Step 1: Verify that the supply voltage is applied.  
Step 2: Go through the key sequence and verify the output code in Table 1.

| Key  | $A_3$ | $A_2$ | $A_1$ | $A_0$ |
|------|-------|-------|-------|-------|
| None | 1     | 1     | 1     | 1     |
| 0    | 1     | 1     | 1     | 1     |

|   |   |   |   |
|---|---|---|---|
| 1 | 1 | 1 | 0 |
| 2 | 1 | 1 | 0 |
| 3 | 1 | 0 | 0 |
| 4 | 1 | 0 | 1 |
| 5 | 1 | 0 | 0 |
| 6 | 1 | 0 | 1 |
| 7 | 1 | 0 | 0 |
| 8 | 0 | 1 | 1 |
| 9 | 0 | 1 | 0 |

**TABLE 1**

*Step 3:* Check for proper priority operation by repeating the key sequence in Table 1 except that for each key closure, hold that key down and depress each lower-valued key as specified in Table 2.

| Hold down keys | Depress keys one at a time           | $A_3$ | $A_2$ | $A_1$ | $A_0$ |
|----------------|--------------------------------------|-------|-------|-------|-------|
| 1              | 0                                    | 1     | 1     | 1     | 0     |
| 2              | 1, 0                                 | 1     | 1     | 0     | 1     |
| 3              | 2, 1, 0                              | 1     | 1     | 0     | 0     |
| 4              | 3, 2, 1, 0                           | 1     | 0     | 1     | 1     |
| 5              | 4, 3, 2, 1, 0                        | 1     | 0     | 1     | 0     |
| 6              | 5, 4, 3, 2, 1, 0                     | 1     | 0     | 0     | 1     |
| 7              | 6, 5, 4, 3, 2, 1, 0                  | 1     | 0     | 0     | 0     |
| 8              | 7, 6, 5, 4, 3, 2,                    | 0     | 1     | 1     | 1     |
| 9              | 1, 0<br>8, 7, 6, 5, 4, 3,<br>2, 1, 0 | 0     | 1     | 1     | 0     |

**TABLE 2**

- 35.** (a) Open  $A_1$  input acts as a HIGH. All binary values corresponding to a BCD number having a 1's value of 0, 1, 4, 5, 8, or 9 will be off by 2. This will first be seen for a BCD value of 00000000.  
 (b) Open  $C_{out}$  of top adder. All values not normally involving a carry out will be off by 32. This will first be seen for a BCD value of 00000000.  
 (c) The  $\Sigma_4$  output of top adder is shorted to ground. Same binary values above 15 will be short by 16. The first BCD value to indicate this will be 00011000.  
 (d)  $\Sigma_3$  of bottom adder is shorted to ground. Every other set of 16 value starting with 16 will be short 16. The first BCD value to indicate this will be 00010110.
- 36.** (a) The 1Y1 output of the 74LS139 is stuck HIGH or open; B cathode open.  
 (b) No power; EN input to the 74LS139 is open.  
 (c) The f output of the 74LS48 is stuck HIGH.  
 (d) The frequency of the data select input is too low.
- 37.** 1. Place a LOW on pin 7 (Enable).  
 2. Apply a HIGH to  $D_0$  and a LOW to  $D_1$  through  $D_7$ .  
 3. Go through the binary sequence on the select inputs and check Y and  $\bar{Y}$  according to Table 3.

| $S_2$ | $S_1$ | $S_0$ | Y | $\bar{Y}$ |
|-------|-------|-------|---|-----------|
| 0     | 0     | 0     | 1 | 0         |
| 0     | 0     | 1     | 0 | 1         |
| 0     | 1     | 0     | 0 | 1         |
| 0     | 1     | 1     | 0 | 1         |
| 1     | 0     | 0     | 0 | 1         |

|   |   |   |   |   |
|---|---|---|---|---|
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 | 1 |

TABLE 3

4. Repeat the binary sequence of select inputs for each set of data inputs listed in Table 4. A HIGH on the  $Y$  output should occur only for the corresponding combinations of select inputs shown.

| $D_0$ | $D_1$ | $D_2$ | $D_3$ | $D_4$ | $D_5$ | $D_6$ | $D_7$ | $Y$ | $\bar{Y}$ | $S_2$ | $S_1$ | $S_0$ |
|-------|-------|-------|-------|-------|-------|-------|-------|-----|-----------|-------|-------|-------|
| L     | H     | L     | L     | L     | L     | L     | L     | 1   | 0         | 0     | 0     | 1     |
| L     | L     | H     | L     | L     | L     | L     | L     | 1   | 0         | 0     | 1     | 0     |
| L     | L     | L     | H     | L     | L     | L     | L     | 1   | 0         | 0     | 1     | 1     |
| L     | L     | L     | L     | H     | L     | L     | L     | 1   | 0         | 1     | 0     | 0     |
| L     | L     | L     | L     | L     | H     | L     | L     | 1   | 0         | 1     | 0     | 1     |
| L     | L     | L     | L     | L     | L     | H     | L     | 1   | 0         | 1     | 1     | 0     |
| L     | L     | L     | L     | L     | L     | L     | H     | 1   | 0         | 1     | 1     | 0     |
| L     | L     | L     | L     | L     | L     | L     | H     | 1   | 0         | 1     | 1     | 1     |

TABLE 4

38. The  $\Sigma$  EVEN output of the 74LS280 should be HIGH and the output of the error gate should be HIGH because of the error condition. Possible faults are:

1.  $\Sigma$  EVEN output of the 74LS280 stuck LOW.
2. Error gate faulty.
3. The ODD input to the 74LS280 is open thus acting as a HIGH.
4. The inverter going to the ODD input of the 74LS280 has an open output or the output is stuck HIGH.

39. Apply a HIGH in turn to each Data input,  $D_0$  through  $D_7$  with LOWs on all the other inputs. For each HIGH applied to a data input, sequence through all eight binary combinations of select inputs ( $S_2S_1S_0$ ) and check for a HIGH on the corresponding data output and LOWs on all the other data outputs.

One possible approach to implementation is to decode the  $S_2S_1S_0$  inputs and generate an inhibit pulse during any given bit time as determined by the settings of seven switches. The inhibit pulse effectively changes a LOW on the  $Y$  serial data line to a HIGH during the selected bit time(s), thus producing a bit error. A basic diagram of this approach is shown in Figure 6-20.

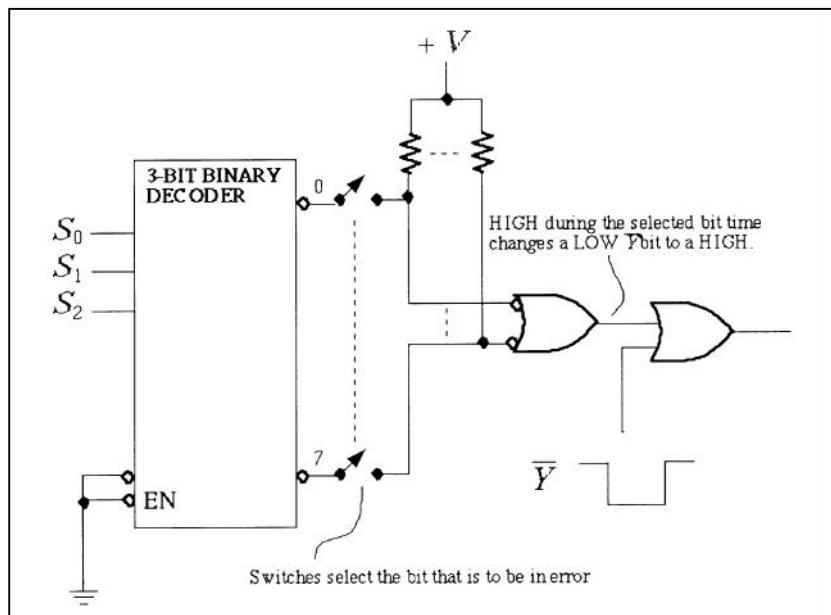
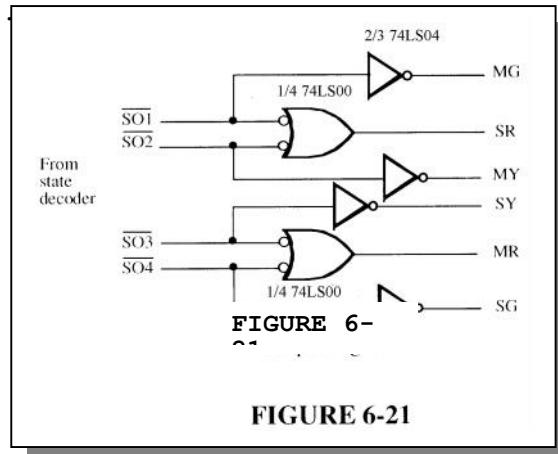


FIGURE 6-20

**FIGURE 6-**  
20

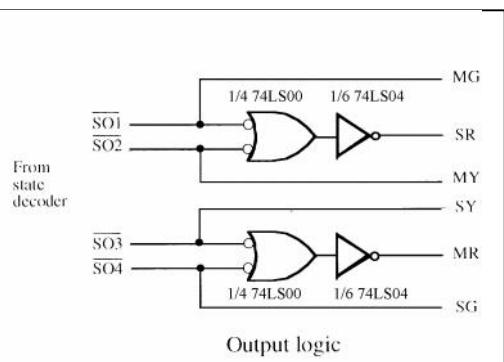
**Digital System Application**

40. See Figure 6-21



**FIGURE 6-21**

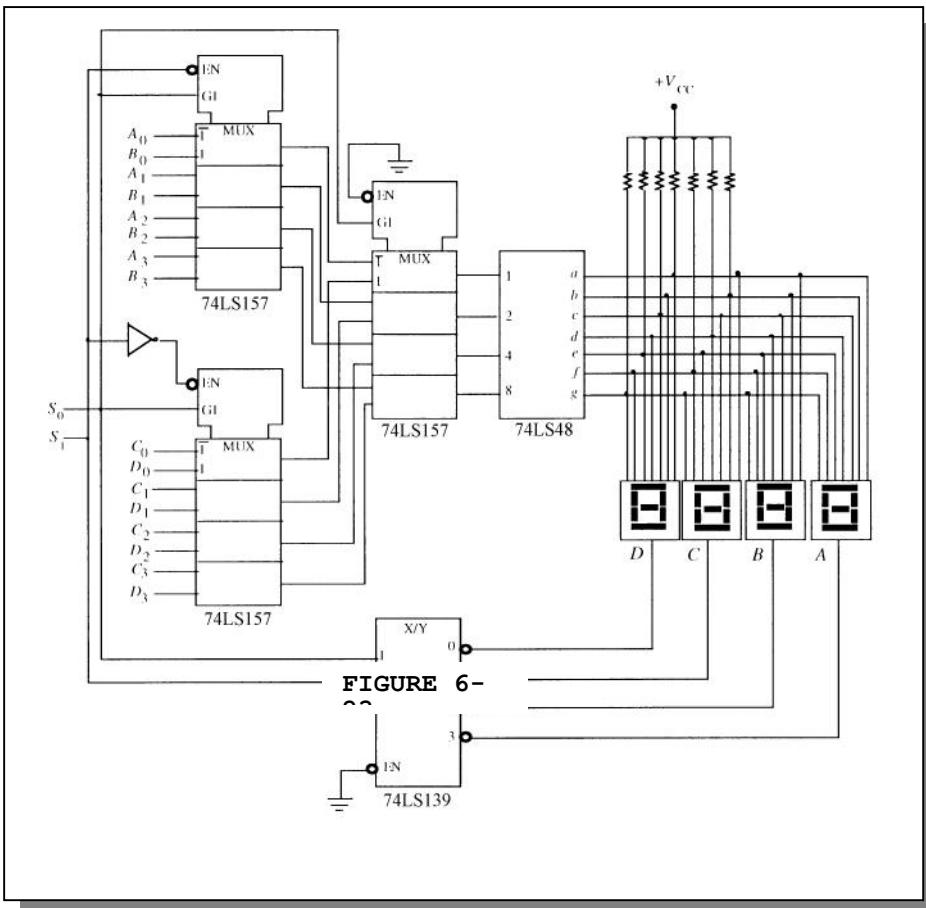
**41.** See Figure 6-2



**Special Design**

**FIGURE 6-**

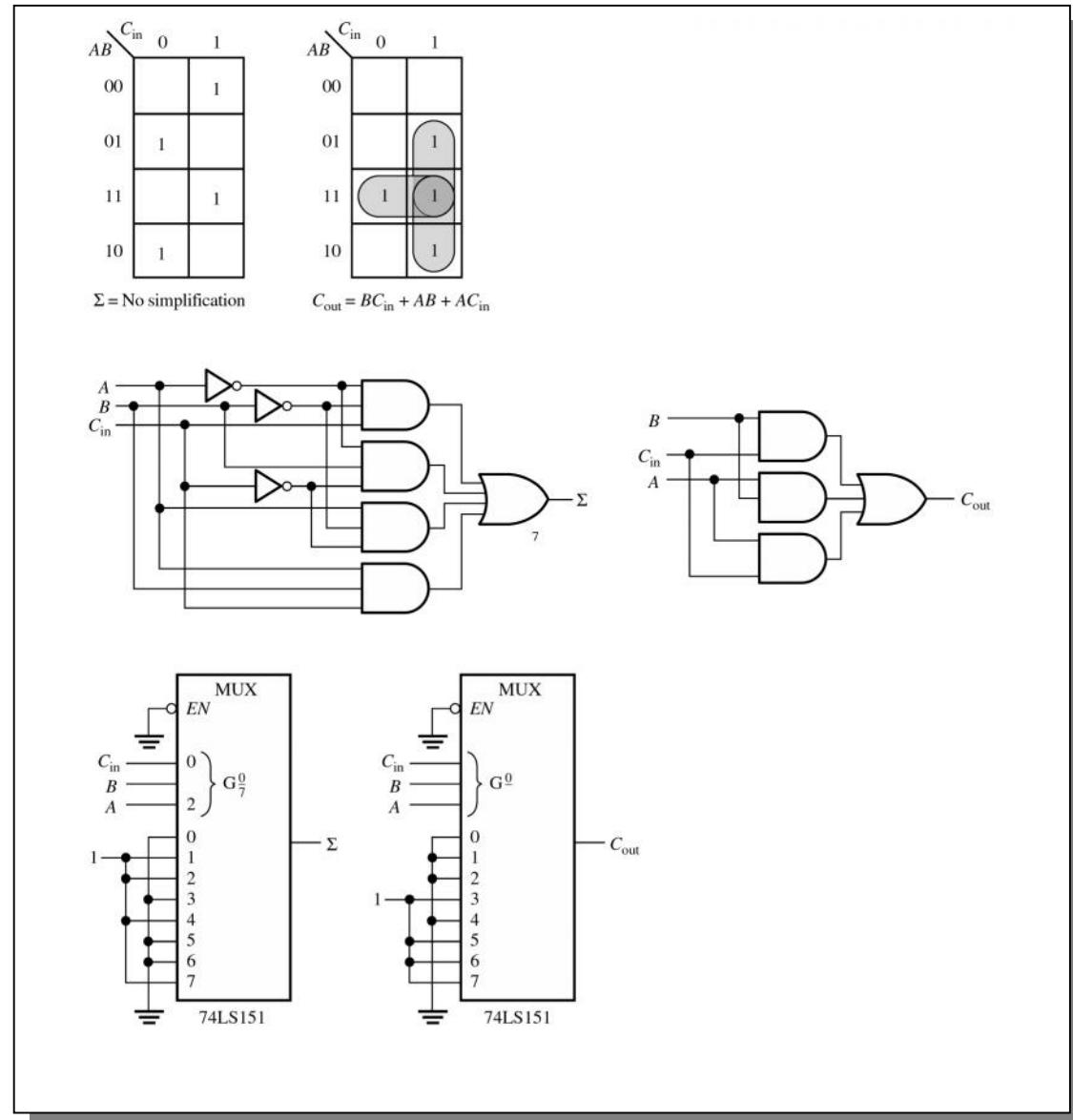
**42.** See Figure 6-23.



$$43. \quad \Sigma = \overline{ABC}_{in} + \overline{AB}\overline{C}_{in} + \overline{A}\overline{BC}_{in} + ABC_{in}$$

$$C_{out} = ABC_{in} + \overline{ABC}_{in} + \overline{AB}\overline{C}_{in} + A\overline{BC}_{in}$$

See Figure 6-24.



$$44. \quad Y = \overline{A_3 A_2 A_1 A_0} + \overline{A_3 A_2} A_1 A_0 + \overline{A_3} A_2 A_1 \overline{A_0} + \overline{A_3} A_2 A_1 A_0 + A_3 \overline{A_2} \overline{A_1} \overline{A_0}$$

$$+ A_3 \overline{A_2} A_1 \overline{A_0} + A_3 \overline{A_2} A_1 A_0 + A_3 A_2 \overline{A_1} \overline{A_0} + A_3 A_2 A_1 A_0$$

See Figure 6-25.

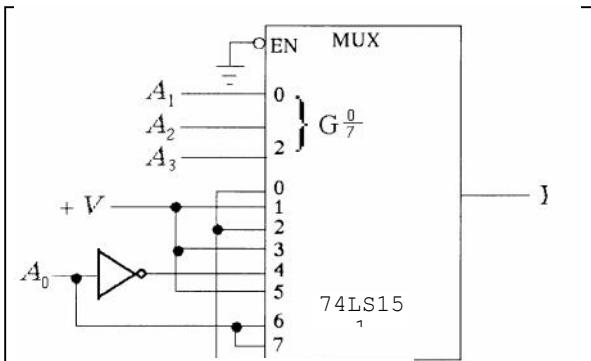


FIGURE 6-25

45. See Figure

FIGURE 6-25

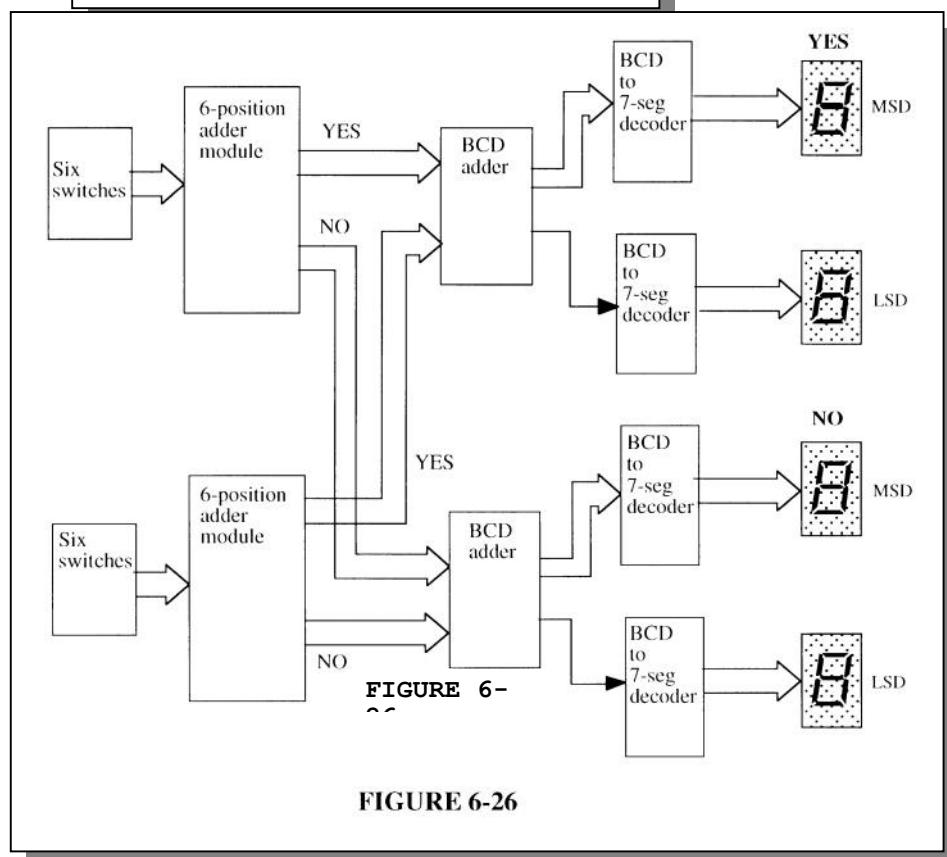
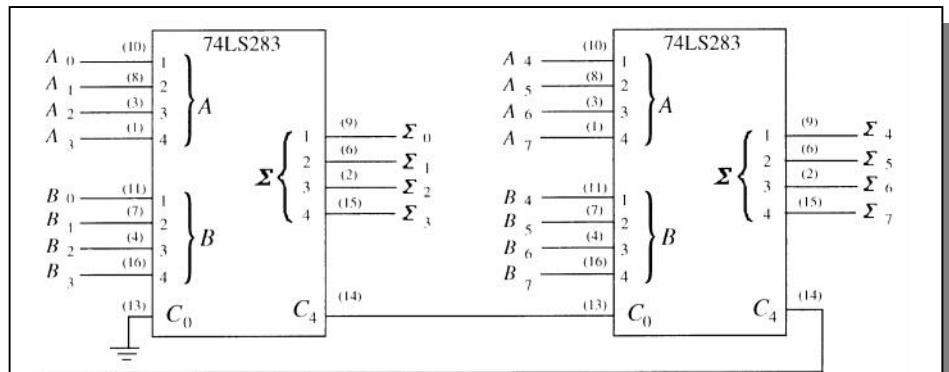
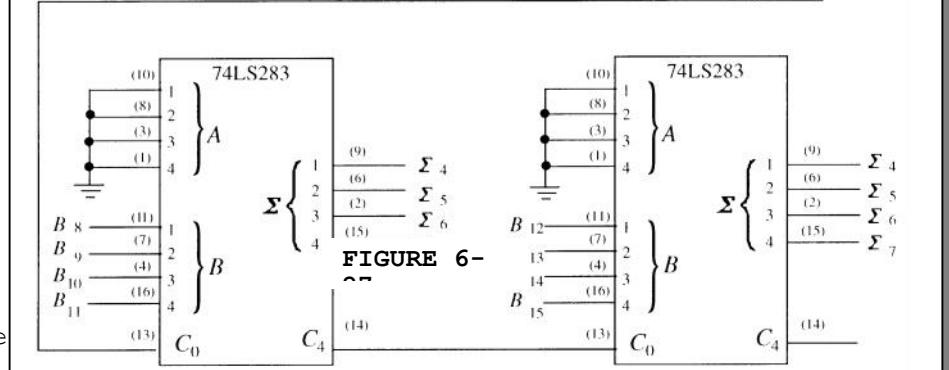


FIGURE 6-26

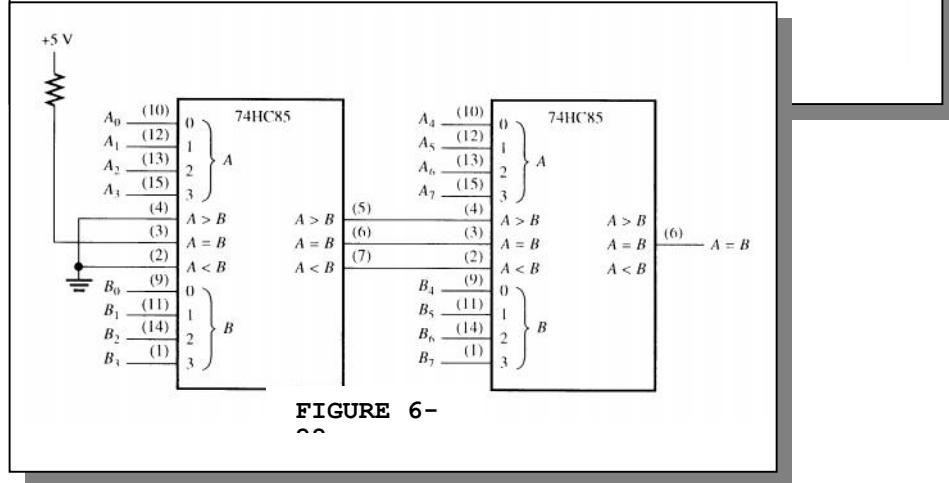
**46.** See Figure 6-27.



**47.** See

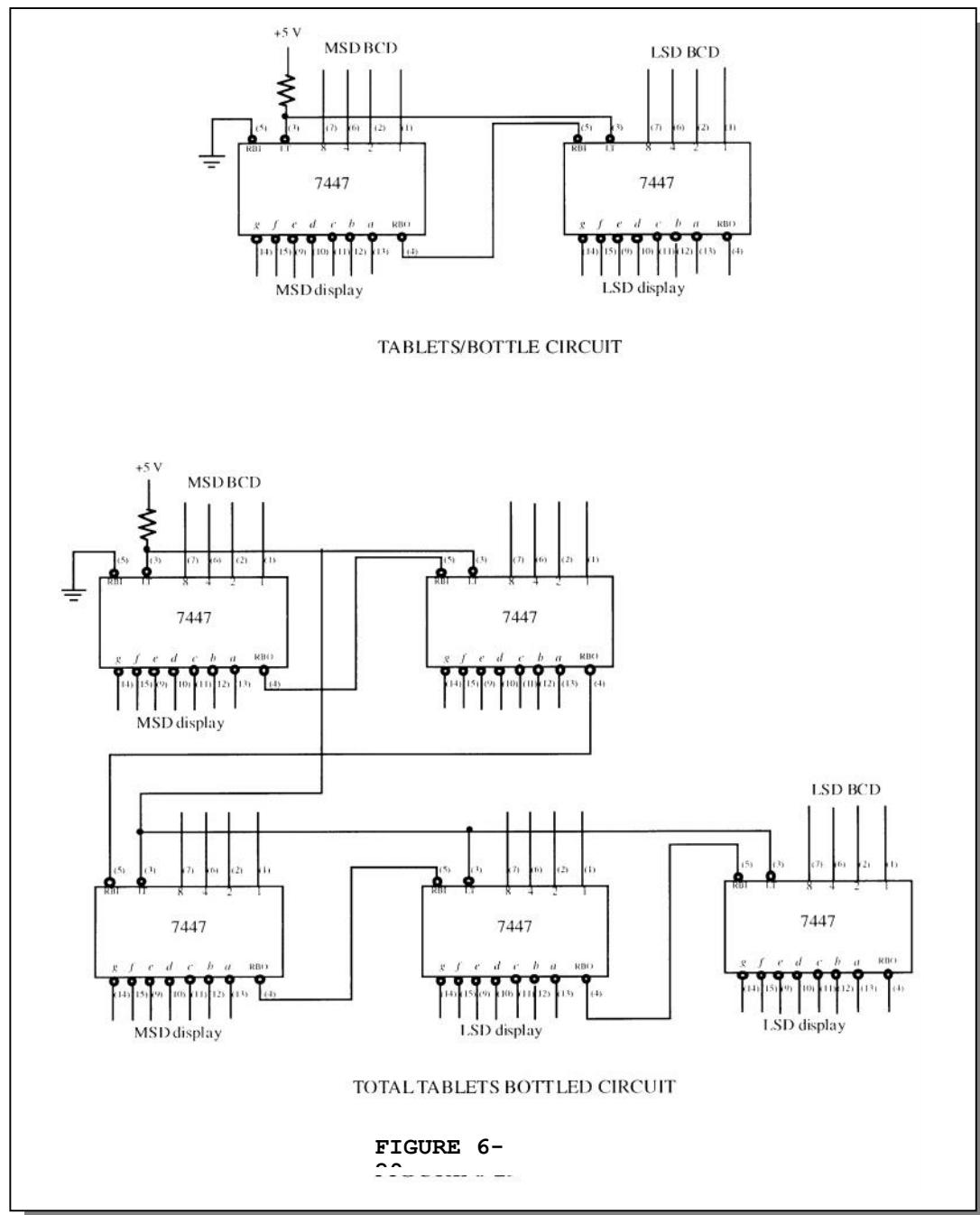


**FIGURE 6-**

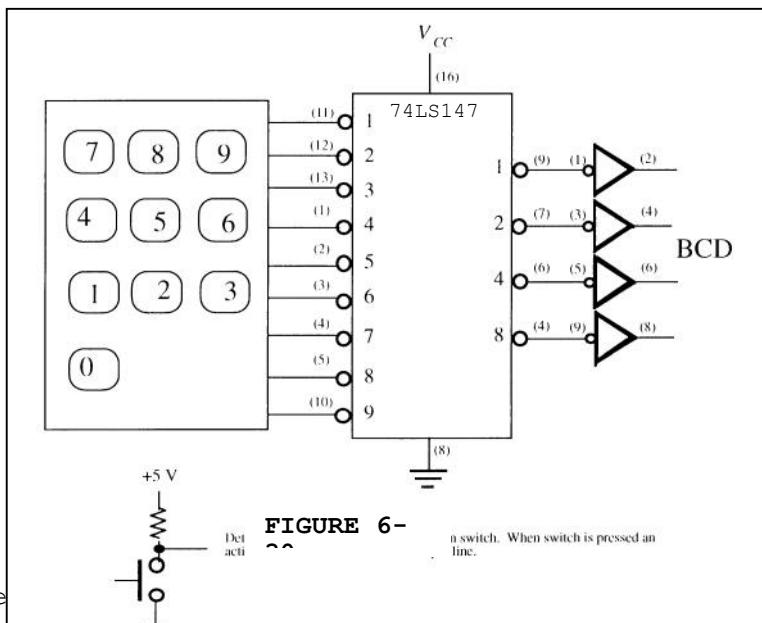


**FIGURE 6-**

**48.** See Figure 6-29.

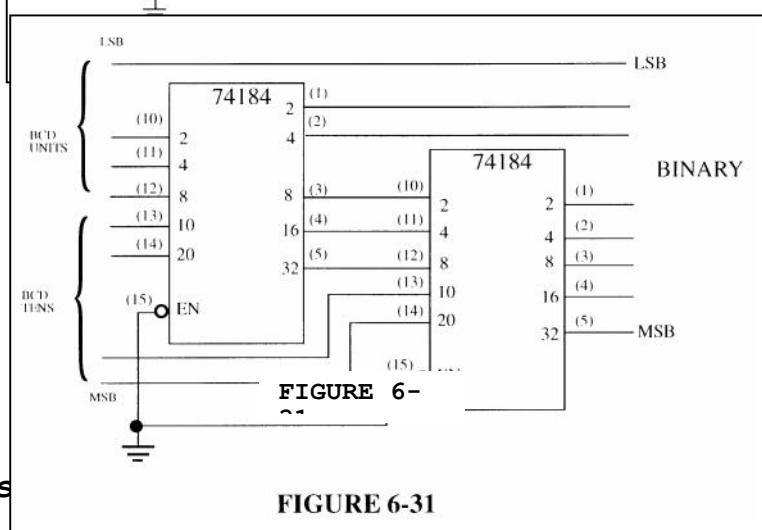


49. See Figure 6-30.



**FIGURE 6-30**

50. See



**FIGURE 6-31**

51. LSB adder carry output open.

52. Pins 4 and 5 shorted together.

53. Pin 12 of upper 74148 open.

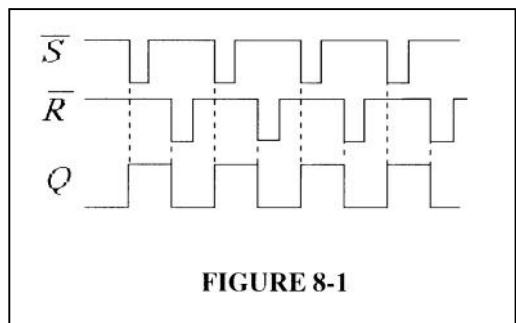
54. Pin 3 of upper 74151 open.

## CHAPTER 7

### LATCHES, FLIP-FLOPS, and TIMERS

#### Section 7-1 Latches

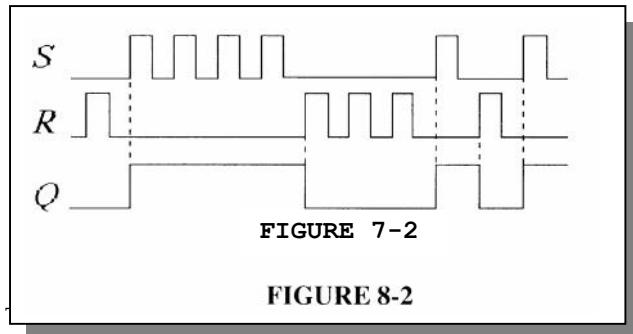
1. See Figure 7-1.



**FIGURE 8-1**

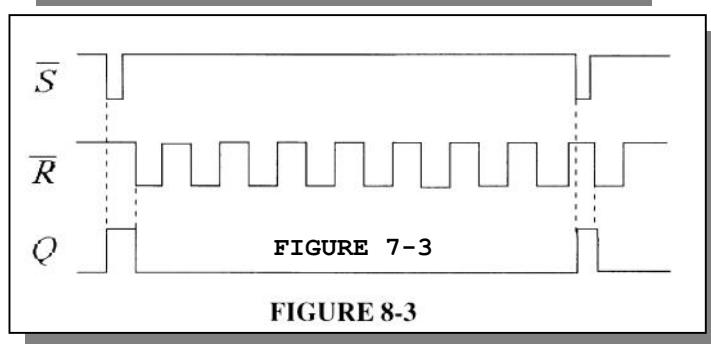
**FIGURE 7-1**

2. See Figure 7-2.



**FIGURE 8-2**

3. See Figure



4. See Figure 7-4.

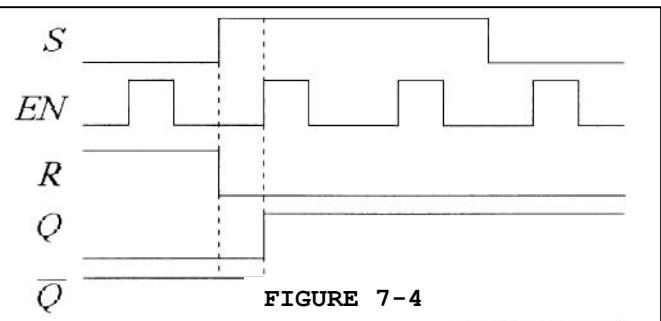


FIGURE 7-4

5. See Figure

FIGURE 8-4

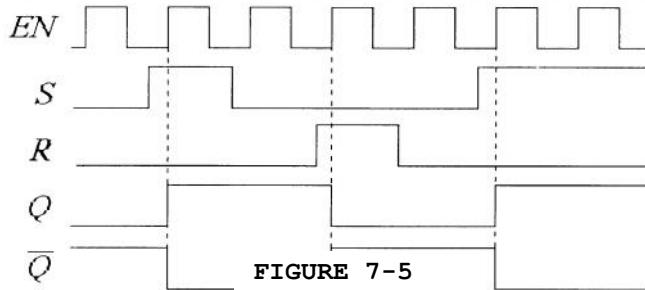


FIGURE 7-5

6. See Figure

FIGURE 8-5

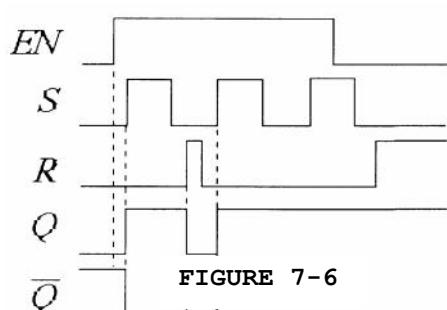
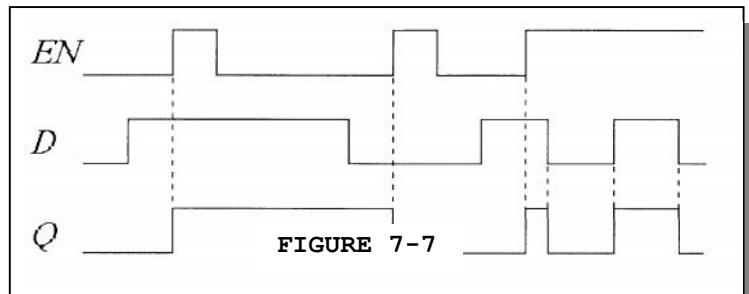


FIGURE 7-6

FIGURE 8-6

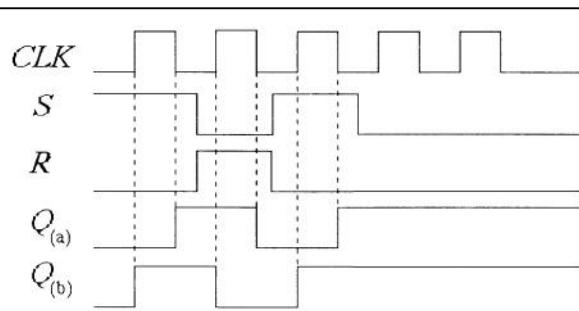
7. See Figure 7-7.



**Section 7**

**FIGURE 8-7**

8. See Figure 7-8.

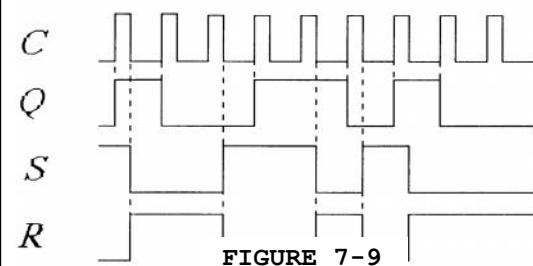


(a) The flip-flop triggers on the negative edge of the clock pulse.

(b) The flip-flop triggers on the positive edge of the clock pulse.

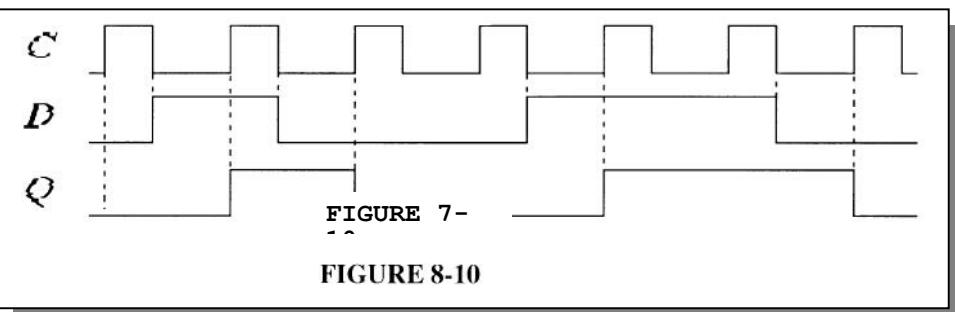
9. See Figure

**FIGURE 8-8**

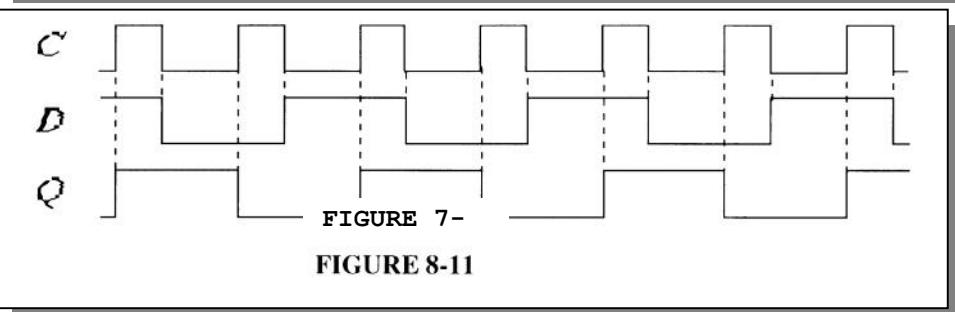


**FIGURE 8-9**

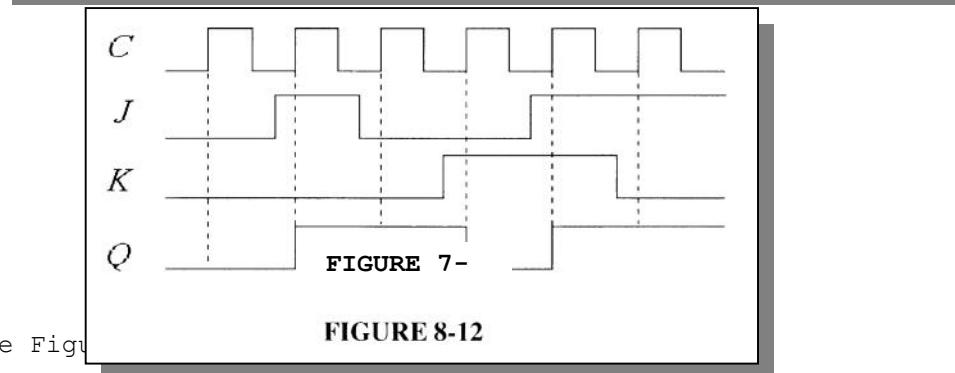
10. See Figure 7-10.



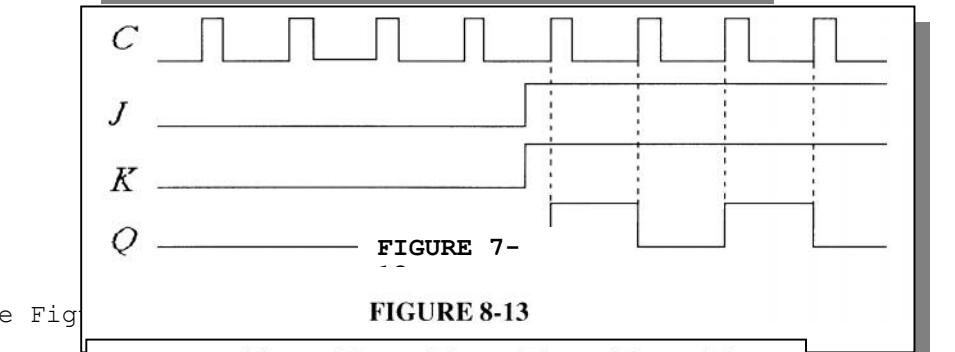
11. See Figure 8-11.



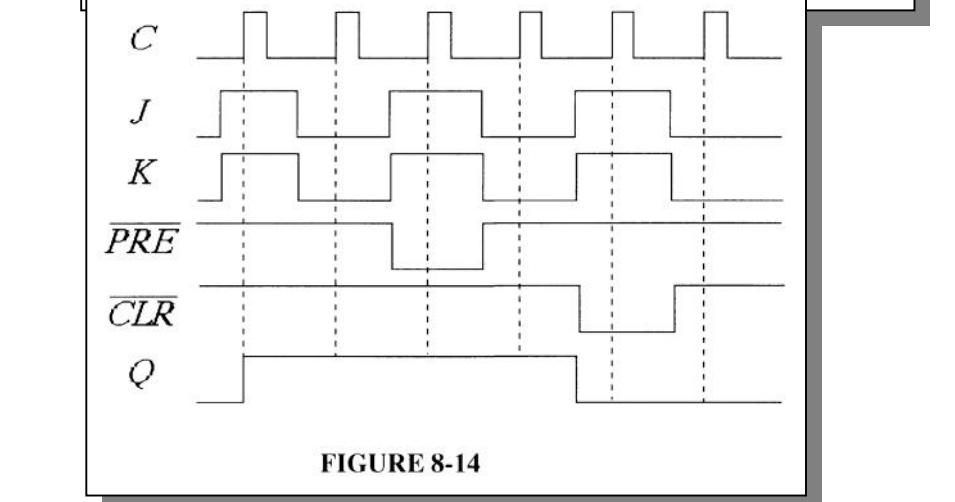
12. See Figure 8-12.



13. See Figure 8-13.

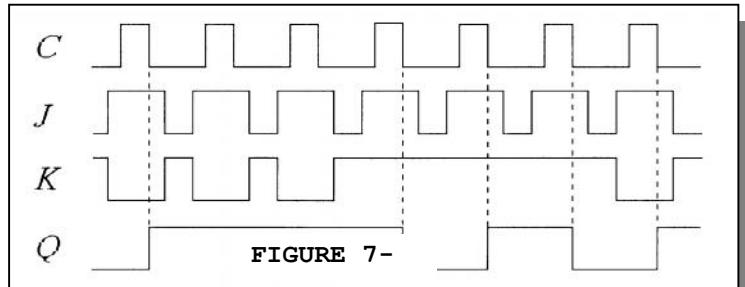


14. See Figure 8-14.



**FIGURE 7-**

15. See Figure 7-15.

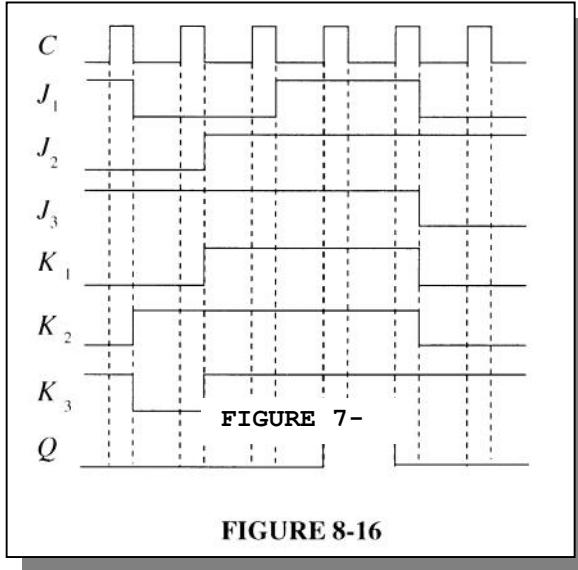


**FIGURE 7-**

16.  $J: 001$   
 $K: 00001$   
 $Q: 0011000$

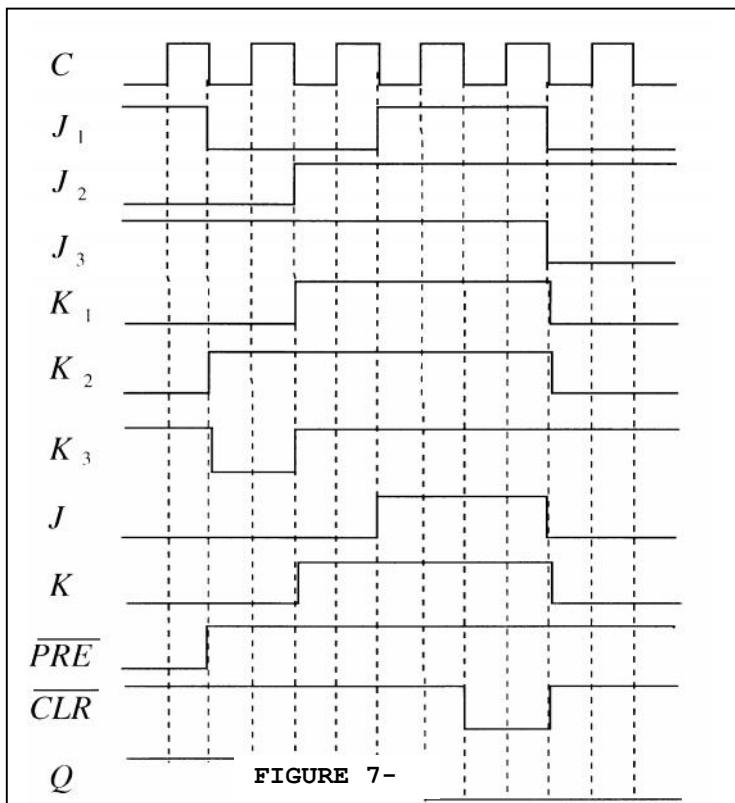
**FIGURE 8-15**

17. See Figure 7-16.



**FIGURE 8-16**

- 18.** See Figure 7-17.



**FIGURE 7-**

## **Section 7**

## **istics**

- 19.** The di:

**FIGURE 8-17**

- 20.**  $t_{PLH}$  ( $\overline{CL}$ ) :

Time from triggering edge of clock to the LOW-to-HIGH transition of the  $Q$  output.

$t_{PHL}$  (Clock to  $Q$ ):

Time from triggering edge of clock to the HIGH-to-LOW transition of the  $Q$  output.

$t_{PLH}$  ( $\overline{PRE}$  to  $Q$ ):

Time from assertion of the Preset input to the LOW-to-HIGH transition of the  $Q$  output.

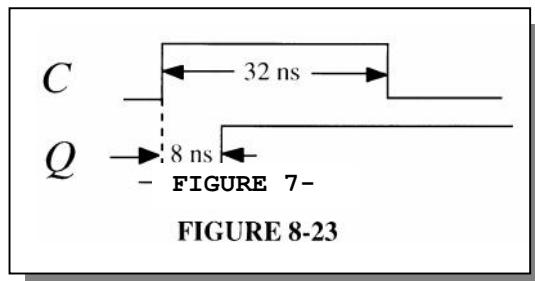
$t_{PHL}$  ( $\overline{CLR}$  to  $Q$ ):

Time from assertion of the clear input to the HIGH-to-LOW transition of the  $Q$  output.

$$21. \quad T_{min} = 30 \text{ ns} + 37 \text{ ns} = 67 \text{ ns}$$

$$f_{max} = \frac{1}{T_{min}} = 14.9 \text{ MHz}$$

22. See Figure 7-18.



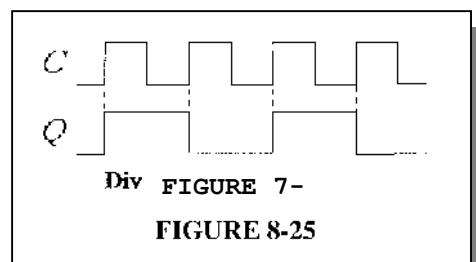
$$23. \quad I_T = 15(10 \text{ mA}) = 150 \text{ mA}$$

$$P_T = (5 \text{ V})(150 \text{ mA}) = 750 \text{ mW}$$

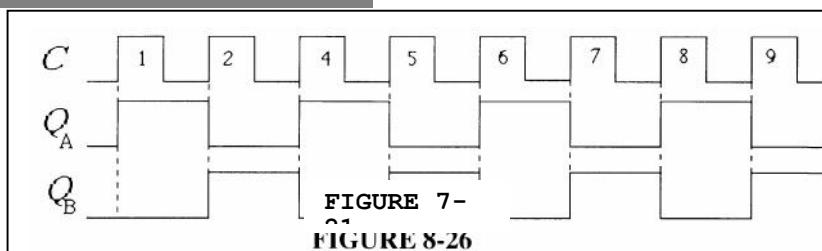
24. See Figure 7-19.

#### **Section 7-4 Flip-Flop Applications**

25. See Figure 7-20.



26.



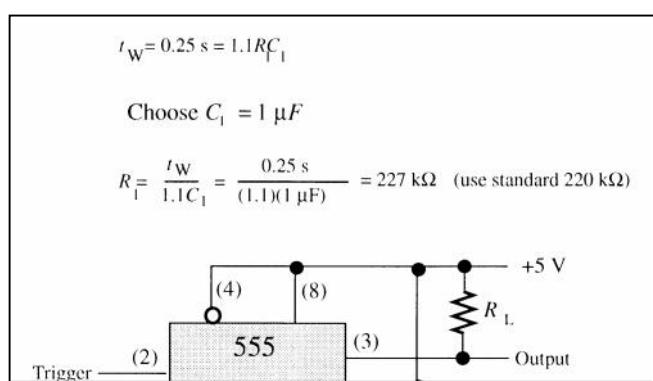
#### **Section**

$$27. \quad t_W = 0.7RC_{\text{EXT}} = 0.7(3.3 \text{ k}\Omega)(2000 \text{ pF}) = 4.62 \mu\text{s}$$

$$28. \quad R_X = \frac{t_W}{RC_{\text{EXT}}} - 0.7 = \frac{5000 \text{ ns}}{0.32 \times 10,000 \text{ pF}} - 0.7 = 1.56 \text{ k}\Omega$$

#### **Section 7-6 The 555 Timer**

29. See Figure 7-22.



**FIGURE 7-**  
**22**

$$30. \quad f = \frac{1}{0.7(R_1 + 2R_2)C_2} = \frac{1}{0.7(1000\Omega + 2200\Omega)(0.01\mu F)} = \mathbf{44.6 \text{ kHz}}$$

$$31. \quad T = \frac{1}{f} = \frac{1}{20 \text{ kHz}} = 50 \text{ } \mu\text{s}$$

For a duty cycle of 75%:

$$t_H = 37.5 \text{ } \mu\text{s} \text{ and } t_L = 12.5 \text{ } \mu\text{s}$$

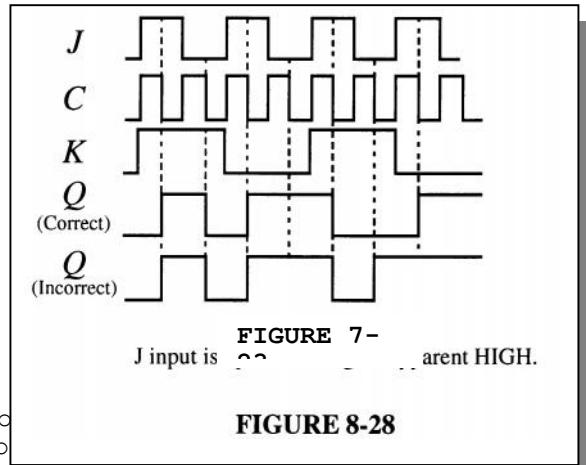
$$R_1 + R_2 = \frac{t_H}{0.7C} = \frac{37.5 \mu\text{s}}{0.7(0.002 \mu\text{F})} = 26,786 \text{ } \Omega$$

$$R_2 = \frac{t_L}{0.7C} = \frac{12.5 \mu\text{s}}{0.7(0.002 \mu\text{F})} = \mathbf{8,929 \text{ } \Omega} \text{ (use } 9.1 \text{ k}\Omega\text{)}$$

$$R_1 = 26,786 \text{ } \Omega - R_2 = 26,786 \text{ } \Omega - 8,929 \text{ } \Omega = \mathbf{17,857 \text{ } \Omega} \text{ (use } 18 \text{ k}\Omega\text{)}$$

### **Section 7-7 Troubleshooting**

32. The flip-flop in Figure 7-90 of the text has an internally open *J* input.
33. The wire from pin 6 to pin 10 and the ground wire are reversed. Pin 7 should be at ground and pin 6 connected to pin 10.
34. See Figure 7-23.



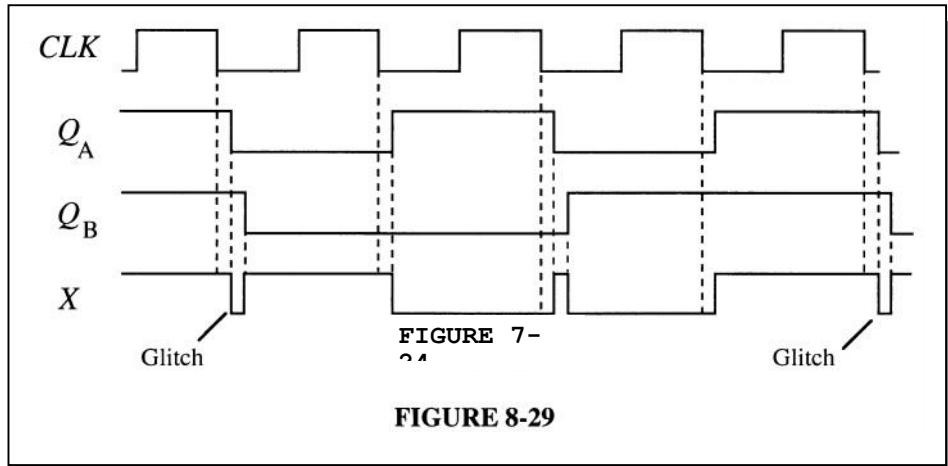
**FIGURE 7-**

J input is open - therefore Q is not reset when C is HIGH.

35. Since none of the inputs affects all of the flip-flops

must be a fault that affects all the flip-flops. One of the clock (CLK) and clear (CLR) inputs must be shorted to ground because a LOW on either one will prevent the flip-flops from changing state. Most likely, the CLR line is shorted to ground because if the clock line were shorted chances are that all of the flip-flops would not have ended up reset when the power was turned on unless an initial LOW was applied to the CLR at power on.

36. Small differences in the switching times of flip-flop A and flip-flop B due to propagation delay cause the glitches as shown in the expanded timing diagram in Figure 7-24. The delays are exaggerated greatly for purposes of illustration. Glitches are eliminated by strobing the output with the clock pulse.

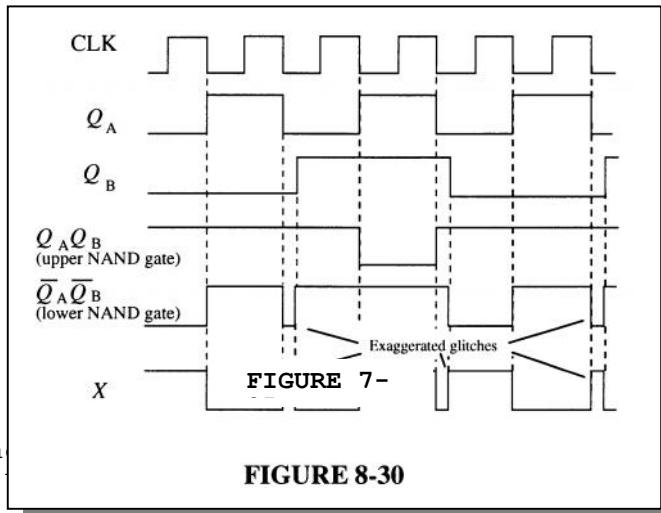


**FIGURE 7-**

24

**FIGURE 8-28**

37. (a) See Figure 7-25.

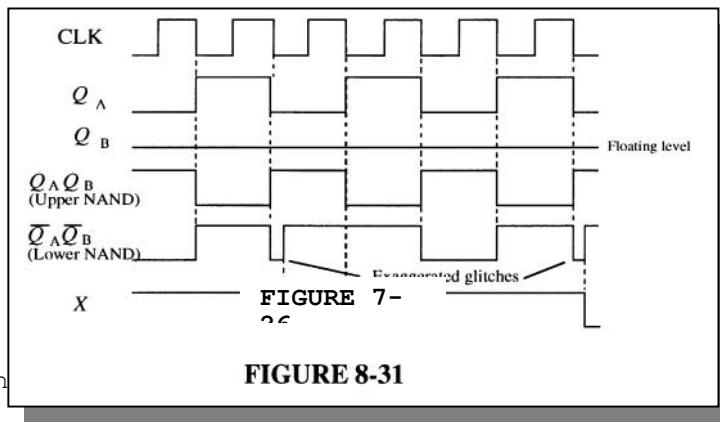


**FIGURE 8-30**

The timing

- (b) K<sub>B</sub> open a  
diagram is

- (c) See Figure 7-26.



**FIGURE 8-31**

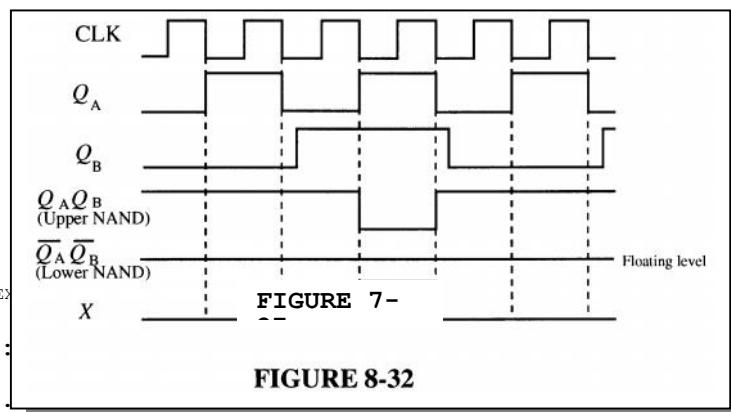
- (d) X remain  
 $(\overline{Q}_B = 1)$ .

- (e) See Figure 7-27.

38.  $t_w = 0.7RC_{ex}$

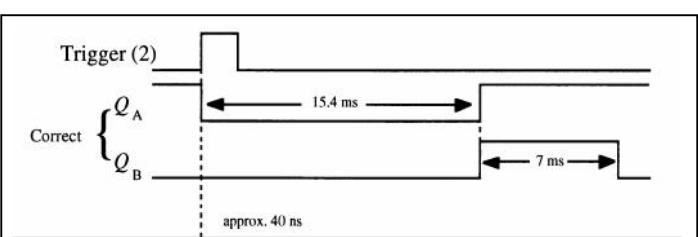
One-shot A:

One-shot B:



**FIGURE 8-32**

The pulse width of one shot A is apparently not controlled by the external components and the one-shot is producing its minimum pulse width of about 40 ns. An open pin 11 would cause this problem. See Figure 7-28.



**FIGURE 7-**

### Digital System Application

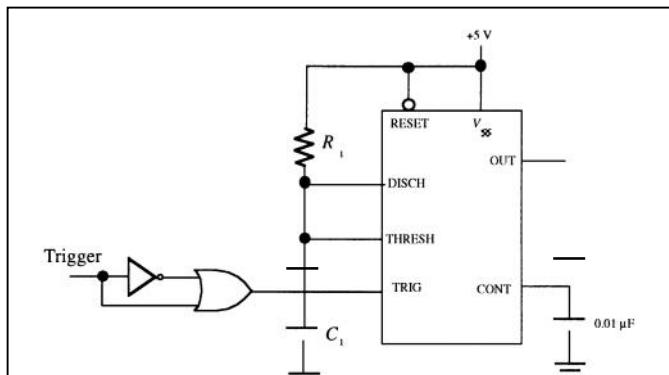
39. For the 4 s timer let  $C_1 = 1 \mu\text{F}$

$$R_1 = \frac{4\text{s}}{(1.1)(1\mu\text{F})} = 3.63 \text{ M}\Omega \text{ (use } 3.9 \text{ M}\Omega)$$

For the 25 s timer let  $C_1 = 2.2 \mu\text{F}$

$$R_1 = \frac{25\text{s}}{(1.1)(2.2\mu\text{F})} = 10.3 \text{ M}\Omega \text{ (use } 10 \text{ M}\Omega)$$

See Figure 7-29.

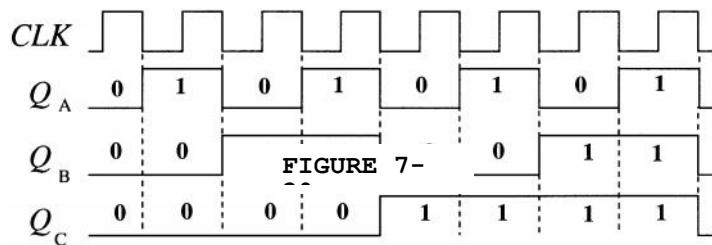
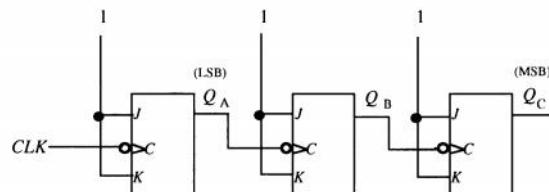


**FIGURE 7-**  
4 s timer a  
except for the  
component values calculated above.

### Special Design

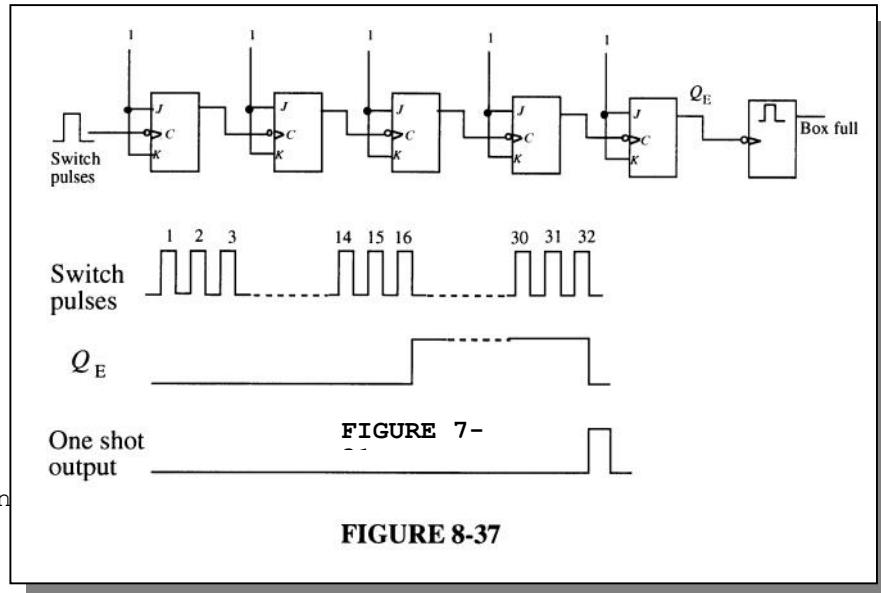
40. See Figure 7

**FIGURE 8-34**



**FIGURE 8-36**

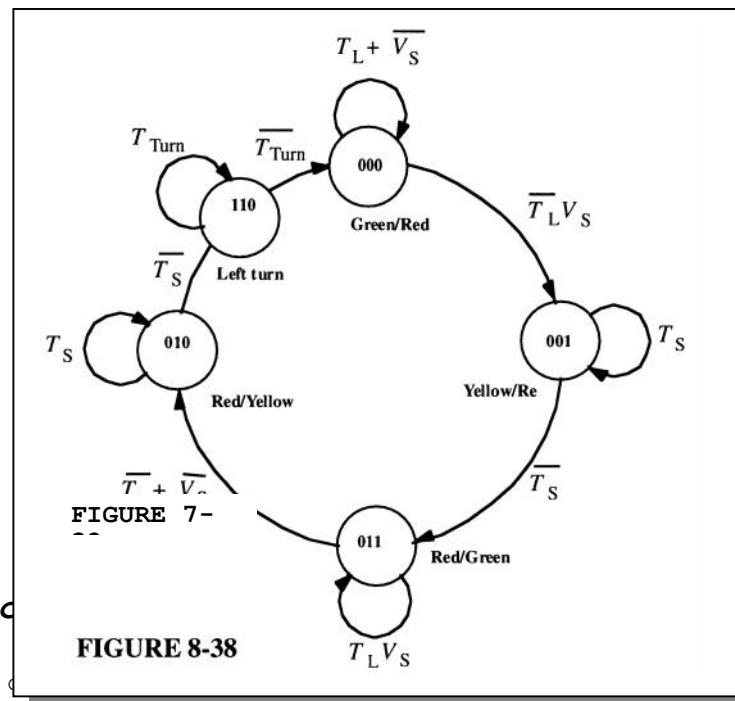
**41.** See Figure 7-31 for one possibility.



**FIGURE 7-**

**42.** Chan  
signal on

- 1.
  2. Add decoding logic to the State Decoder to decode the turn signal state.
  3. Change the Output Logic to incorporate the turn signal output.
  4. Change the Trigger Logic to incorporate a trigger output for the turn signal timer.
  5. Add a 15 second timer.
- See Figure 7-32.



**FIGURE 8-38**

**Multisim Trac**

**43.**  $\bar{Q}$  output o

**44.**  $K$  input of U2 open.

**45.**  $\overline{SET}$  input of U1 open.

- 46.** No fault.
- 47.** K input of U2 open.

---

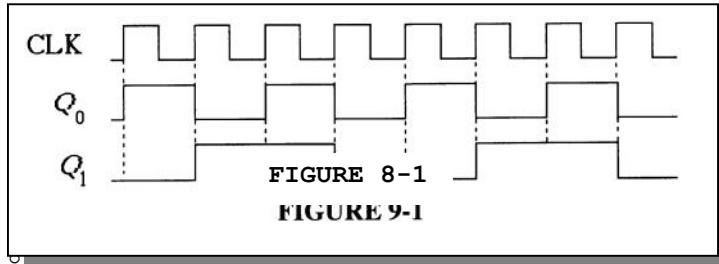
## CHAPTER 8

### COUNTERS

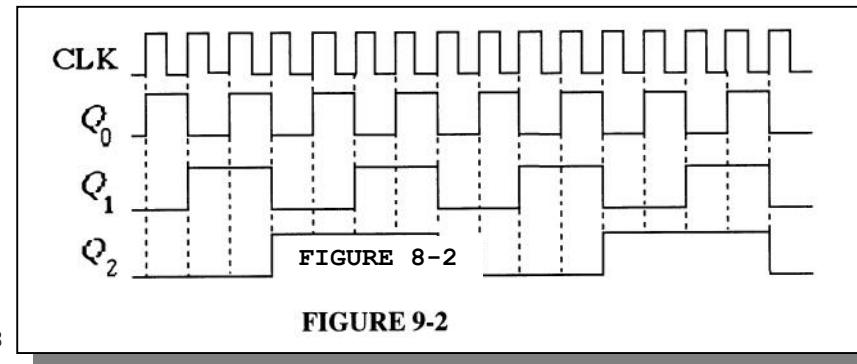
---

#### Section 8-1 Asynchronous Counter Operation

1. See Figure 8-1.



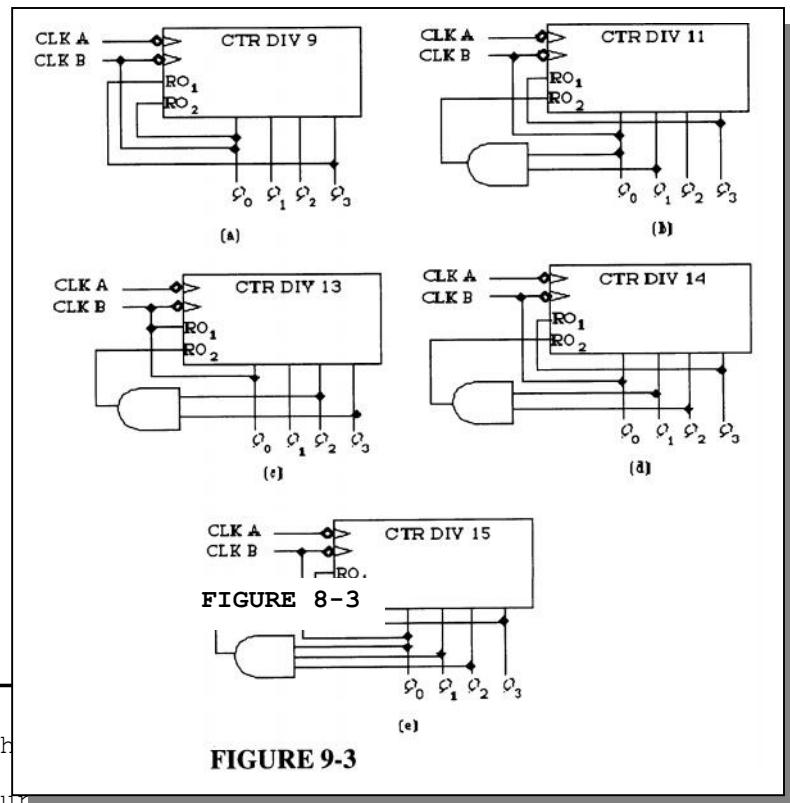
2. See Figure 8-2.



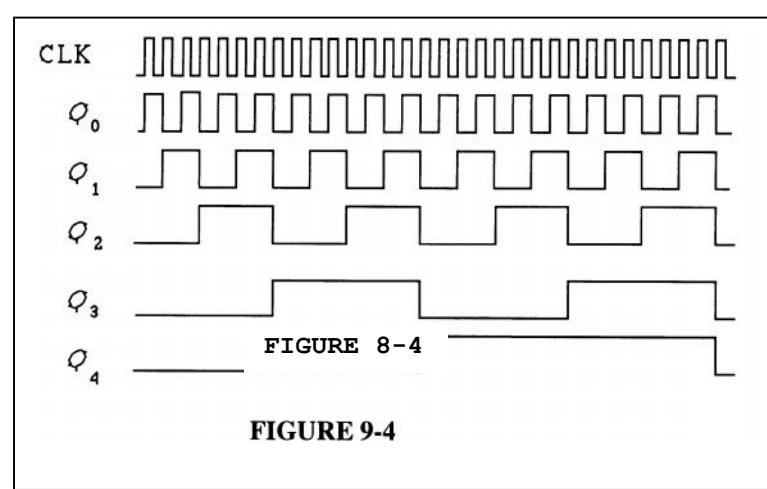
3.  $t_{p(\max)} = 3(8)$

Worst-case delay occurs when all flip-flops change state from 011 to 100 or from 111 to 000.

- 4.** See Figure 8-3.



**FIGURE 9-3**

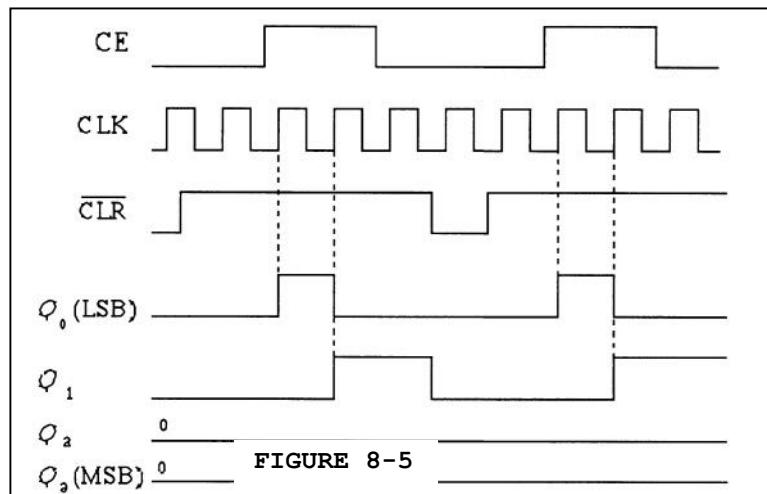


**FIGURE 9-4**

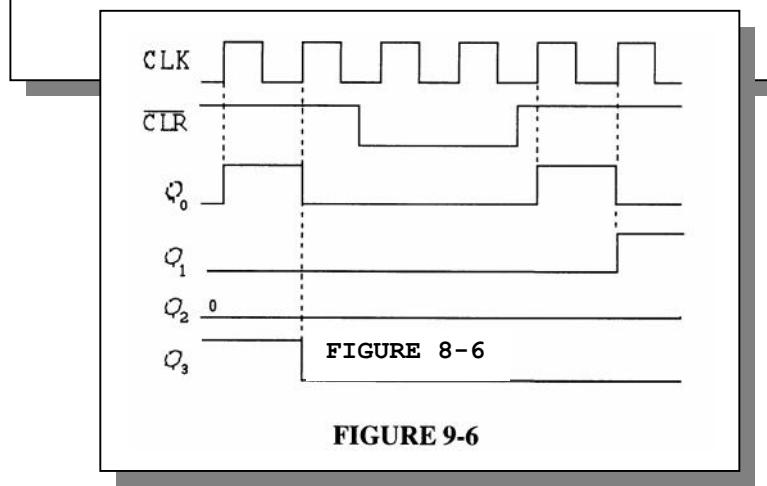
7. Each flip-flop is initially reset.

| CLK | $J_0K$ | $J_1K$ | $J_2K$ | $J_3K$ | $Q_0$ | $Q_1$ | $Q_2$ | $Q_3$ |
|-----|--------|--------|--------|--------|-------|-------|-------|-------|
|     | 0      | 1      | 2      | 3      |       |       |       |       |
| 1   | 1      | 0      | 0      | 0      | 1     | 0     | 0     | 0     |
| 2   | 1      | 1      | 0      | 0      | 0     | 1     | 0     | 0     |
| 3   | 1      | 0      | 0      | 0      | 1     | 1     | 0     | 0     |
| 4   | 1      | 1      | 1      | 0      | 0     | 0     | 1     | 0     |
| 5   | 1      | 0      | 0      | 0      | 1     | 0     | 1     | 0     |
| 6   | 1      | 1      | 0      | 0      | 0     | 1     | 1     | 0     |
| 7   | 1      | 0      | 0      | 0      | 1     | 1     | 1     | 0     |
| 8   | 1      | 1      | 1      | 1      | 0     | 0     | 0     | 1     |
| 9   | 1      | 0      | 0      | 0      | 1     | 0     | 0     | 1     |
| 10  | 1      | 0      | 0      | 1      | 0     | 0     | 0     | 0     |

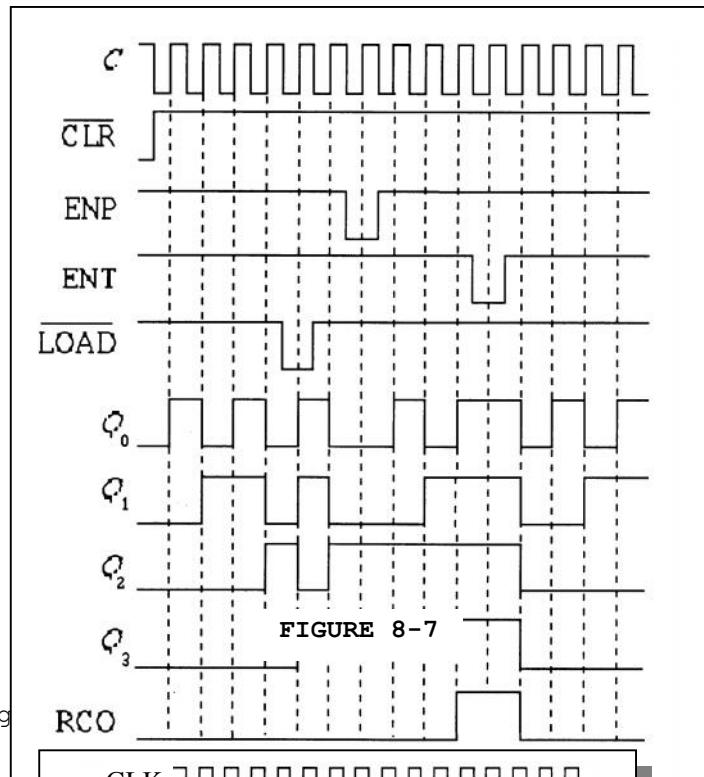
8. See Figure 8-5.



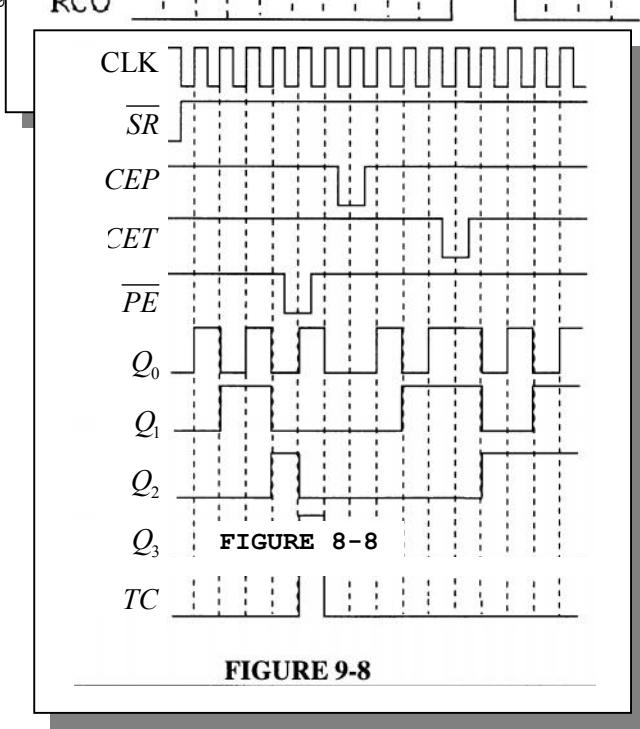
9. See Figure 9-6.



10. See Figure 8-7.

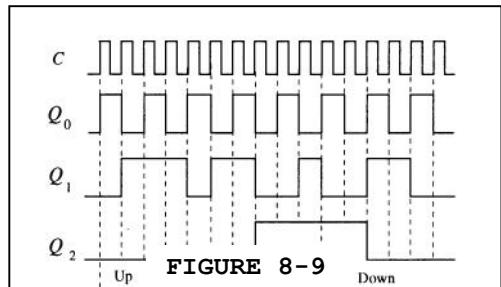


11. See Fig



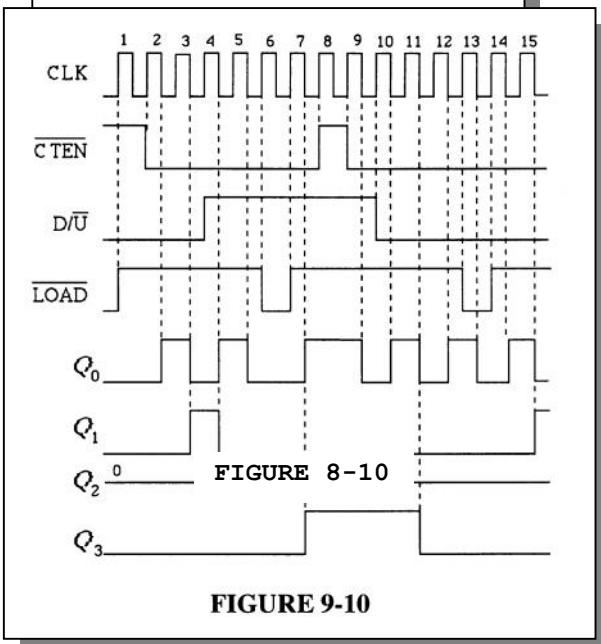
### **Section 8-3 Up/Down Synchronous Counters**

12. See Figure 8-9.



**FIGURE 8-9**

13. See Figure 8-1



**FIGURE 8-10**

### **Section 8-4**

14.

|           | $Q_2$ | $Q_1$ | $Q_0$ | $D_2$ | $D_1$ | $D_0$ |
|-----------|-------|-------|-------|-------|-------|-------|
| Initially | 0     | 0     | 0     | 0     | 0     | 1     |
| At CLK 1  | 0     | 0     | 1     | 0     | 1     | 1     |
| At CLK 2  | 0     | 1     | 1     | 1     | 1     | 1     |
| At CLK 3  | 1     | 1     | 1     | 1     | 1     | 0     |
| At CLK 4  | 1     | 1     | 0     | 1     | 0     | 0     |
| At CLK 5  | 1     | 0     | 0     | 0     | 0     | 1     |
| At CLK 6  | 0     | 0     | 1     | 0     | 1     | 1     |

The sequence is 000 to 001 to 011 to 111 to 110 to 100 and back to 001, etc.

15.

|             | FF3 | FF2 | FF1 | FF0 | $Q_3$ | $Q_2$ | $Q_1$ | $Q_0$ |
|-------------|-----|-----|-----|-----|-------|-------|-------|-------|
| Initially   | Tog | Tog | Tog | Tog | 0     | 0     | 0     | 0     |
| After CLK 1 | NC  | NC  | NC  | Tog | 1     | 1     | 1     | 1     |
| After CLK 2 | NC  | NC  | Tog | Tog | 1     | 1     | 1     | 0     |
| After CLK 3 | NC  | Tog | Tog | Tog | 1     | 1     | 0     | 1     |
| After CLK 4 | Tog | Tog | Tog | Tog | 1     | 0     | 1     | 0     |
| After CLK 5 | Tog | Tog | Tog | Tog | 0     | 1     | 0     | 1     |

|   |  |  |
|---|--|--|
| 5 |  |  |
|---|--|--|

Tog = toggle, NC = no change

The counter locks up in the 1010 and 0101 states, alternating between them.

**16. NEXT-STATE TABLE**

| <b>Present State</b> |       | <b>Next State</b> |       |
|----------------------|-------|-------------------|-------|
| $Q_1$                | $Q_0$ | $Q_1$             | $Q_0$ |
| 0                    | 0     | 1                 | 0     |
| 1                    | 0     | 0                 | 1     |
| 0                    | 1     | 1                 | 1     |
| 1                    | 1     | 0                 | 0     |

**TRANSITION TABLE**

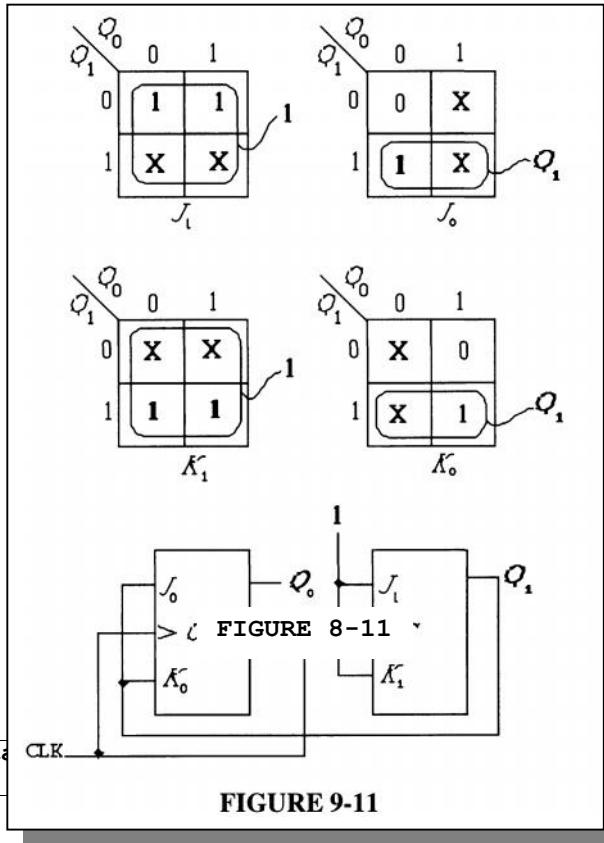
| <b>Output State Transitions</b><br>(Present state to next state) |        | <b>Flip-Flop Inputs</b> |       |       |       |
|------------------------------------------------------------------|--------|-------------------------|-------|-------|-------|
| $Q_1$                                                            | $Q_0$  | $J_1$                   | $K_1$ | $J_0$ | $K_0$ |
| 0 to 1                                                           | 0 to 0 | 1                       | X     | 0     | X     |
| 1 to 0                                                           | 0 to 1 | X                       | 1     | 1     | X     |
| 0 to 1                                                           | 1 to 1 | 1                       | X     | X     | 0     |
| 1 to 0                                                           | 1 to 0 | X                       | 1     | X     | 1     |

See Figure 8-11.

17. NEXT-STATE

| Present State |       | Next State |       |       |
|---------------|-------|------------|-------|-------|
| $Q_2$         | $Q_1$ | $Q_2$      | $Q_1$ | $Q_0$ |
| 0             | 0     | 0          | 0     | 1     |
| 1             | 0     | 1          | 0     | 1     |
| 0             | 1     | 1          | 1     | 0     |
| 1             | 0     | 1          | 1     | 1     |
| 1             | 1     | 1          | 1     | 0     |
| 1             | 1     | 0          | 0     | 1     |
| 0             | 1     | 0          | 0     | 1     |

**FIGURE 9-11**



> FIGURE 8-11 <

TRANSITION TABLE

| Output State<br>Transitions<br>(Present state to<br>next state) |        |        | Flip-flop Inputs |       |       |       |       |       |  |
|-----------------------------------------------------------------|--------|--------|------------------|-------|-------|-------|-------|-------|--|
| $Q_2$                                                           | $Q_1$  | $Q_0$  | $J_2$            | $K_2$ | $J_1$ | $K_1$ | $J_0$ | $K_0$ |  |
| 0 to 0                                                          | 0 to 0 | 1 to 0 | 1                | X     | 0     | X     | X     | 1     |  |
| 1                                                               | 0      | 0 to 1 | X                | 1     | 1     | X     | 1     | X     |  |
| 1 to 0                                                          | 0 to 1 | 1 to 1 | 1                | X     | X     | 1     | X     | 0     |  |
| 0                                                               | 1      | 1 to 1 | X                | 0     | 1     | X     | X     | 0     |  |
| 0 to 1                                                          | 1 to 0 | 1 to 0 | X                | 0     | X     | 0     | X     | 1     |  |
| 1                                                               | 0      | 0 to 1 | 0                | X     | X     | 1     | 1     | X     |  |
| 1 to 0                                                          | 0 to 0 | 0 to 0 | X                | 1     | X     | 0     | 0     | X     |  |
| 1                                                               | 1      |        |                  |       |       |       |       |       |  |
| 1 to 1                                                          | 1 to 0 |        |                  |       |       |       |       |       |  |
| 0 to 1                                                          | 1 to 0 |        |                  |       |       |       |       |       |  |
| 0                                                               | 0      |        |                  |       |       |       |       |       |  |
| 1 to 0                                                          | 1 to 0 |        |                  |       |       |       |       |       |  |
| 0                                                               | 1      |        |                  |       |       |       |       |       |  |

See Figure 8-12.

|                                                                                   |                                                                                   |                                                                                                       |
|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------|
| $Q_2 Q_1 Q_0$<br><p>00 X 1<br/>01 0 1<br/>11 X X<br/>10 X X</p> $J_2 = Q_0$       | $Q_2 Q_1 Q_0$<br><p>00 X 0<br/>01 X X<br/>11 X X<br/>10 1 1</p> $J_1 = Q_2$       | $Q_2 Q_1 Q_0$<br><p>00 X X<br/>01 1 X<br/>11 0 X<br/>10 1 X</p> $J_0 = \bar{Q}_1 + \bar{Q}_2$         |
| $Q_2 Q_1 Q_0$<br><p>00 X X<br/>01 X X<br/>11 1 0<br/>10 1 0</p> $K_2 = \bar{Q}_0$ | $Q_2 Q_1 Q_0$<br><p>00 X X<br/>01 1 1<br/>11 0 0<br/>10 X X</p> $K_1 = \bar{Q}_2$ | $Q_2 Q_1 Q_0$<br><p>00 X 1<br/>01 X 0<br/>11 X 1<br/>10 X 0</p> $K_0 = \bar{Q}_1 \bar{Q}_2 + Q_1 Q_2$ |

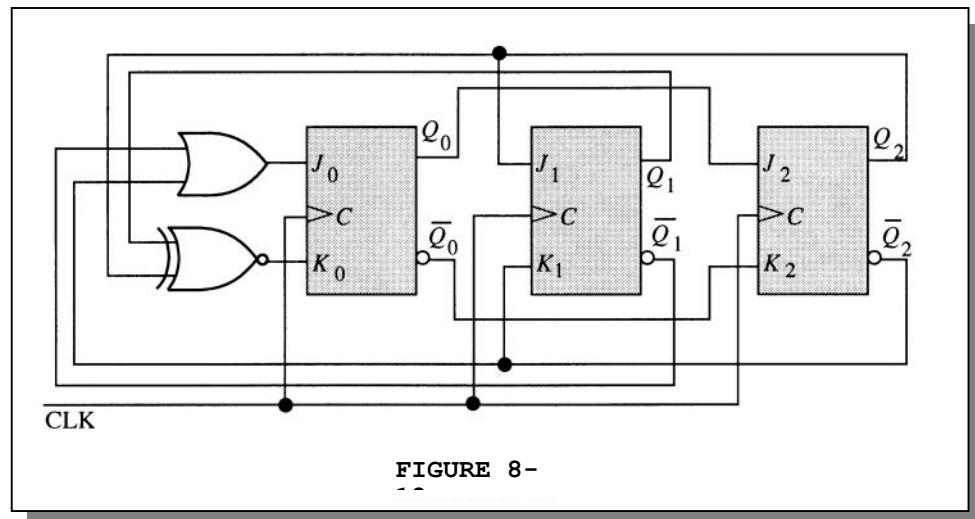


FIGURE 8-

## 18. NEXT-STATE TABLE

| Present State |       |       |       | Next State |       |       |       |
|---------------|-------|-------|-------|------------|-------|-------|-------|
| $Q_3$         | $Q_2$ | $Q_1$ | $Q_0$ | $Q_3$      | $Q_2$ | $Q_1$ | $Q_0$ |
| 0             | 0     | 0     | 0     | 1          | 0     | 0     | 1     |
| 1             | 0     | 0     | 1     | 0          | 0     | 0     | 1     |
| 0             | 0     | 0     | 1     | 1          | 0     | 0     | 0     |
| 1             | 0     | 0     | 0     | 0          | 0     | 1     | 0     |

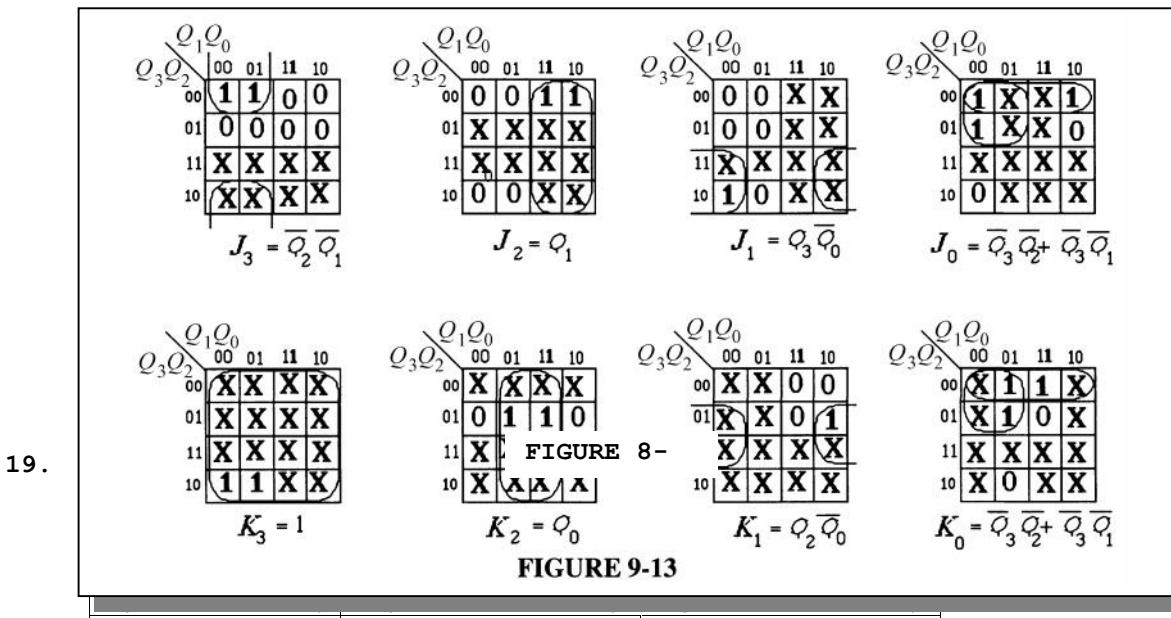
|   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |

TRANSITION TABLE

| Output State Transition<br>(Present State to next state) |        |        |        | Flip-flop Inputs |       |       |       |       |       |       |       |
|----------------------------------------------------------|--------|--------|--------|------------------|-------|-------|-------|-------|-------|-------|-------|
| $Q_3$                                                    | $Q_2$  | $Q_1$  | $Q_0$  | $J_3$            | $K_3$ | $J_2$ | $K_2$ | $J_1$ | $K_1$ | $J_0$ | $K_0$ |
| 0 to 1                                                   | 0 to 0 | 0 to 0 | 0 to 1 | 1                | X     | 0     | X     | 0     | X     | 1     | X     |
| 1 to 0                                                   | 0 to 0 | 0 to 0 | 1      | X                | 1     | 0     | X     | 0     | X     | X     | 0     |
| 1 to 0                                                   | 0 to 0 | 0 to 0 | 0 to 0 | 1                | X     | 0     | X     | 0     | X     | X     | 1     |
| 0 to 0                                                   | 0 to 0 | 0 to 1 | 0 to 1 | X                | 1     | 0     | X     | 1     | X     | 0     | X     |
| 1 to 0                                                   | 0 to 0 | 0 to 0 | 0 to 0 | 0                | X     | 1     | X     | X     | 0     | X     | 0     |
| 1 to 0                                                   | 0 to 0 | 0 to 1 | 0 to 0 | 0                | X     | 1     | X     | X     | 0     | X     | 1     |
| 0 to 0                                                   | 0 to 0 | 1 to 0 | 0 to 0 | 0                | X     | X     | 0     | X     | 1     | 0     | X     |
| 0 to 1                                                   | 1 to 1 | 1 to 1 | 1      | 0                | X     | X     | 1     | 0     | X     | 1     | X     |
| 0 to 0                                                   | 1 to 1 | 1 to 1 | 1      | 0                | X     | X     | 1     | 0     | X     | 1     | X     |
| 0 to 0                                                   | 0 to 1 | 1 to 1 | 0      | 0                | X     | X     | 0     | X     | 1     | 0     | X     |
| 0 to 0                                                   | 1 to 0 | 1 to 0 | 0      | 0                | X     | X     | 0     | X     | 1     | 0     | X     |
| 0 to 0                                                   | 1 to 0 | 0 to 0 | 0 to 0 | 0                | X     | X     | 0     | X     | 1     | 0     | X     |
| 0 to 0                                                   | 1 to 0 | 0 to 0 | 0 to 0 | 0                | X     | X     | 0     | X     | 1     | 0     | X     |
| 0 to 0                                                   | 1 to 0 | 0 to 0 | 1 to 0 | 0                | X     | X     | 1     | 0     | X     | X     | 1     |
| 0 to 0                                                   | 0 to 0 | 0 to 0 | 0 to 0 | 0                | X     | X     | 1     | 0     | X     | X     | 1     |

Binary states for 10, 11, 12, 13, 14, and 15 are unallowed and can be represented by don't cares.

See Figure 8-13. Counter implementation is straightforward from input expressions.



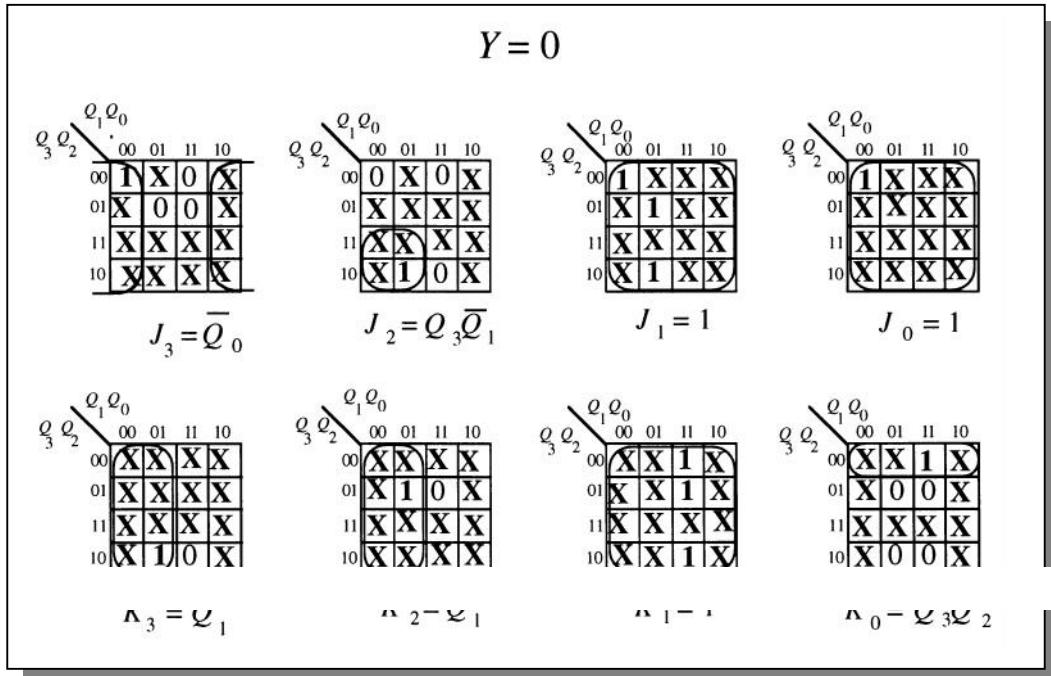
|   |   |   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|

|   |   |   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |

TRANSITION TABLE

| Output State Transitions<br>(Present State to next state) |        |        |        | Y | Flip-flop Inputs |          |          |          |
|-----------------------------------------------------------|--------|--------|--------|---|------------------|----------|----------|----------|
| $Q_3$                                                     | $Q_2$  | $Q_1$  | $Q_0$  |   | $J_3K_3$         | $J_2K_2$ | $J_1K_1$ | $J_0K_0$ |
| 0 to 0                                                    | 0 to 0 | 0 to 0 | 0 to 0 | 0 | 1X               | 0X       | 1X       | 1X       |
| 1                                                         | 0      | 1      | 1      | 1 | 0X               | 0X       | 1X       | 1X       |
| 0 to 0                                                    | 0 to 0 | 0 to 0 | 0 to 0 | 0 | 0X               | 0X       | X1       | X1       |
| 0                                                         | 0      | 1      | 1      | 1 | 0X               | 1X       | X1       | X0       |
| 0 to 0                                                    | 0 to 0 | 1 to 1 | 1 to 1 | 0 | 0X               | X1       | 1X       | X0       |
| 0                                                         | 0      | 0      | 0      | 1 | 0X               | X0       | 1X       | X0       |
| 0 to 0                                                    | 0 to 0 | 1 to 1 | 1 to 1 | 0 | 0X               | X0       | X1       | X0       |
| 0                                                         | 1      | 0      | 1      | 1 | 1X               | X1       | X1       | X0       |
| 0 to 1                                                    | to 0   | 0 to 1 | 1 to 1 | 0 | X1               | 1X       | 1X       | X0       |
| 0                                                         | 0      | 1      | 1      | 1 | X0               | 0X       | 1X       | X0       |
| 0 to 1                                                    | 1 to 0 | 0 to 0 | 1 to 1 | 0 | X0               | 0X       | X1       | X0       |
| 0                                                         | 1      | 1      | 1      | 1 | X1               | 0X       | X1       | X1       |
| 0 to 1                                                    | 1 to 0 | 1 to 1 | 1 to 1 | 0 | 0                | 0        | 0        | 0        |
| 0                                                         | 1      | 0      | 1      | 1 | 0                | 0        | 0        | 0        |
| 0 to 1                                                    | 1 to 0 | 1 to 1 | 1 to 1 | 1 | 0                | 0        | 0        | 0        |
| 1                                                         | 0      | 0      | 1      | 1 | 0                | 0        | 0        | 0        |
| 1 to 0                                                    | 0 to 0 | 0 to 0 | 1 to 1 | 0 | 0                | 0        | 0        | 0        |
| 0                                                         | 1      | 1      | 1      | 1 | 0                | 0        | 0        | 0        |
| 1 to 0                                                    | 0 to 0 | 0 to 0 | 1 to 1 | 0 | 0                | 0        | 0        | 0        |
| 1                                                         | 0      | 1      | 1      | 1 | 0                | 0        | 0        | 0        |
| 1 to 0                                                    | 0 to 0 | 1 to 0 | 1 to 1 | 0 | 0                | 0        | 0        | 0        |
| 1                                                         | 0      | 0      | 1      | 1 | 0                | 0        | 0        | 0        |
| 1 to 0                                                    | 0 to 0 | 1 to 1 | 1 to 1 | 0 | 0                | 0        | 0        | 0        |
| 0                                                         | 0      | 0      | 0      | 0 | 0                | 0        | 0        | 0        |

See Figure 8-14.



| $Y = 1$                                                      |                                                              |                                                                |                                                                |
|--------------------------------------------------------------|--------------------------------------------------------------|----------------------------------------------------------------|----------------------------------------------------------------|
| $J_3 = Q_2 Q_1$                                              | $J_2 = \bar{Q}_3 Q_1$                                        | $J_1 = 1$                                                      | $J_0 = 1$                                                      |
| $K_3 = Q_1$                                                  | $K_2 = Q_1$                                                  | $K_1 = 1$                                                      | $K_0 = \bar{Q}_3 \bar{Q}_2$                                    |
| $Q_3 Q_2 Q_1 Q_0$                                            | $Q_3 Q_2 Q_1 Q_0$                                            | $Q_3 Q_2 Q_1 Q_0$                                              | $Q_3 Q_2 Q_1 Q_0$                                              |
| 00 01 11 10                                                  | 00 01 11 10                                                  | 00 01 11 10                                                    | 00 01 11 10                                                    |
| 00 0 X 0   X<br>01 X 0   1 X<br>11 X X   X X<br>10 X X   X X | 00 0 X 1   X<br>01 X X   X X<br>11 X X   X X<br>10 X 0   0 X | 00 1 X   X X X<br>01 X 1   X X<br>11 X X   X X<br>10 X 1   X X | 00 1 X   X X X<br>01 X X   X X<br>11 X X   X X<br>10 X X   X X |

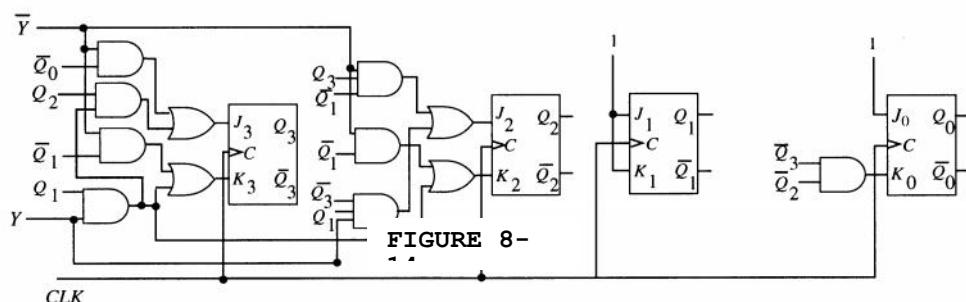


FIGURE 9-14

Section

20. (a)

$$f_1 = \frac{1\text{kHz}}{4} = 250 \text{ Hz}$$

$$f_2 = \frac{250 \text{ Hz}}{8} = 31.25 \text{ Hz}$$

$$f_3 = \frac{31.25 \text{ Hz}}{2} = 15.625 \text{ Hz}$$

(b) Modulus =  $10 \times 10 \times 10 \times 2 = 2000$

$$f_1 = \frac{100 \text{ kHz}}{10} = 10 \text{ kHz}$$

$$f_2 = \frac{10 \text{ kHz}}{10} = 1 \text{ kHz}$$

$$f_3 = \frac{1 \text{ kHz}}{10} = 100 \text{ Hz}$$

$$f_4 = \frac{100 \text{ Hz}}{2} = 50 \text{ Hz}$$

(c) Modulus =  $3 \times 6 \times 8 \times 10 \times 10 = 14400$

$$f_1 = \frac{21 \text{ MHz}}{3} = 7 \text{ MHz}$$

$$f_2 = \frac{7 \text{ MHz}}{6} = 1.167 \text{ MHz}$$

$$f_3 = \frac{1.167 \text{ MHz}}{8} = 145.875 \text{ kHz}$$

$$f_4 = \frac{145.875 \text{ kHz}}{10} = 14.588 \text{ kHz}$$

$$f_5 = \frac{14.588 \text{ kHz}}{10} = 1.459 \text{ kHz}$$

(d) Modulus =  $2 \times 4 \times 6 \times 8 \times 16 = 6144$

$$f_1 = \frac{39.4 \text{ kHz}}{2} = 19.7 \text{ kHz}$$

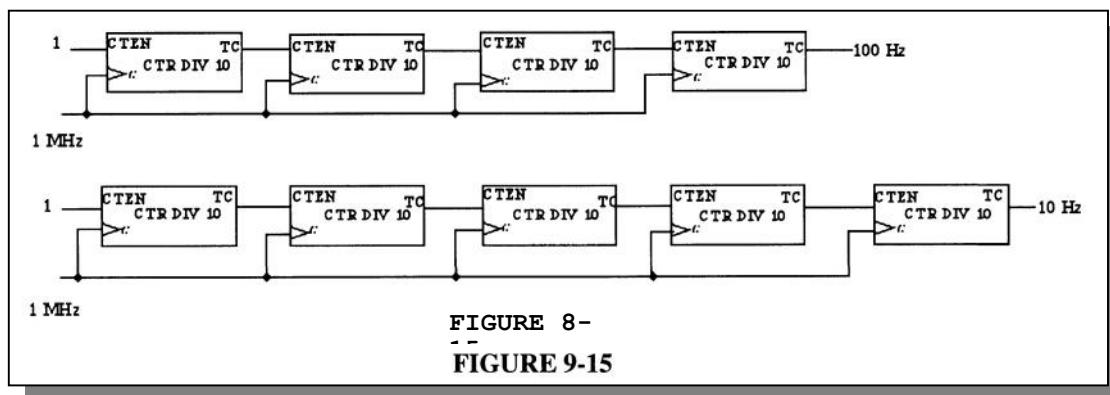
$$f_2 = \frac{19.7 \text{ kHz}}{4} = 4.925 \text{ kHz}$$

$$f_3 = \frac{4.925 \text{ kHz}}{6} = 820.83 \text{ Hz}$$

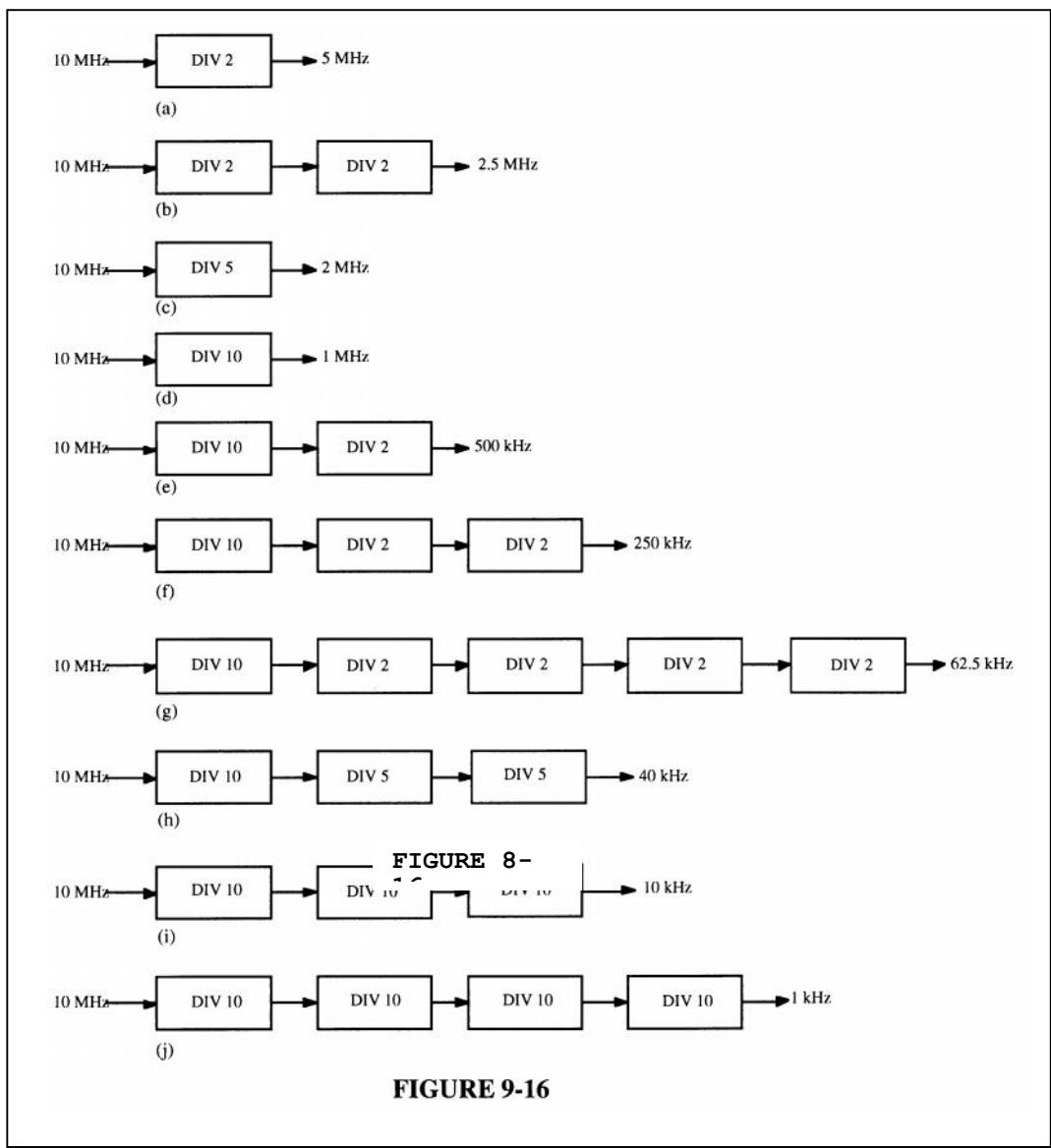
$$f_4 = \frac{820.683}{8} = 102.6 \text{ Hz}$$

$$f_5 = \frac{102.6 \text{ Hz}}{16} = 6.41 \text{ Hz}$$

21. See Figure 8-15.

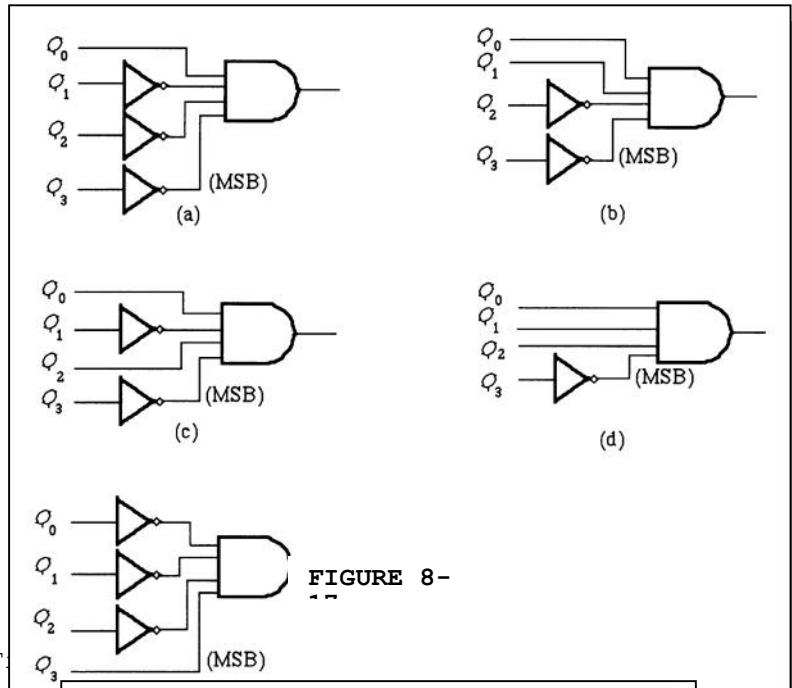


22. See Figure 8-16.

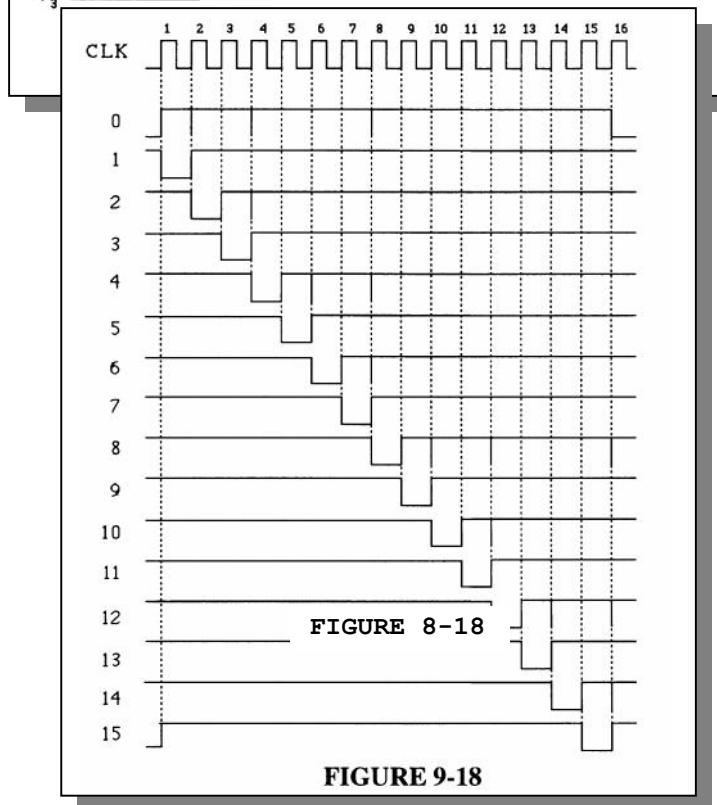


**Section 8-6 Counter Decoding**

23. See Figure 8-17.



24. See Figure 8-18.

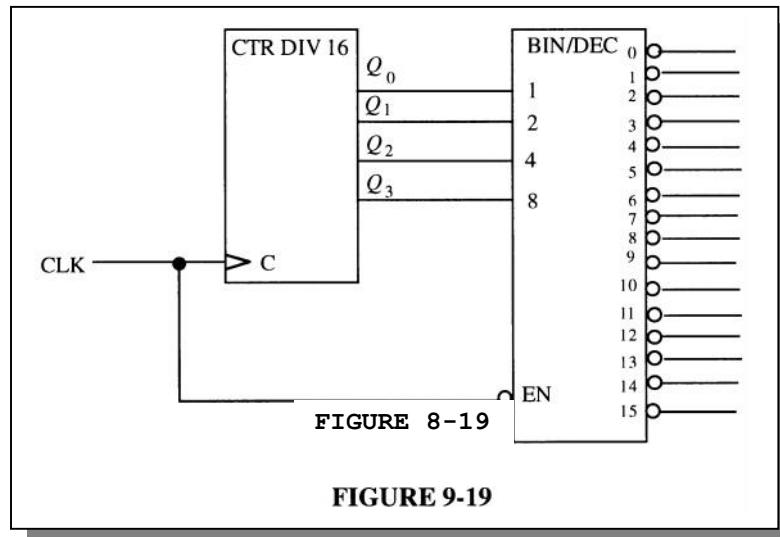


**FIGURE 9-18**

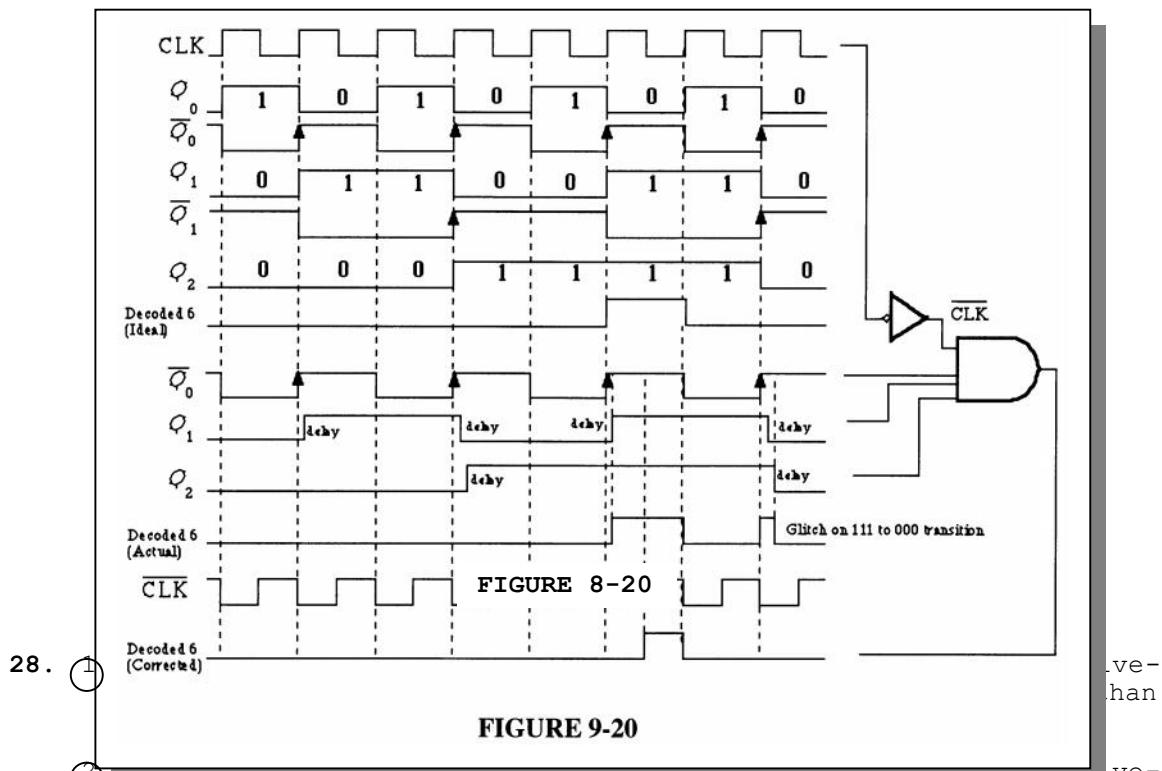
- 25.** The states with an asterisk are the transition states that produce glitches on the decoder outputs. The glitches are indicated on the waveforms in Figure 8-18 (Problem 8-24) by short vertical lines.

|         |        |
|---------|--------|
| Initial | 0000   |
| CLK 1   | 0001   |
| CLK 2   | 0000 * |
|         | 0010   |
| CLK 3   | 0011   |
| CLK 4   | 0010 * |
|         | 0000 * |
|         | 0100   |
| CLK 5   | 0100   |
| CLK 6   | 0100 * |
|         | 0110   |
| CLK 7   | 0111   |
| CLK 8   | 0110 * |
|         | 0100 * |
|         | 0000 * |
|         | 1000   |
| CLK 9   | 1001   |
| CLK 10  | 1000*  |
|         | 1010   |
| CLK 11  | 1011   |
| CLK 12  | 1010 * |
|         | 1000 * |
|         | 1100   |
| CLK 13  | 1101   |
| CLK 14  | 1100 * |
|         | 1110   |
| CLK 15  | 1111   |
| CLK 16  | 1110 * |
|         | 1100 * |
|         | 1000 * |
|         | 0000   |

- 26.** See Figure 8-19.



27. See Figure 8-20.



28.

going edge of CLK 4 if the propagation delay of FF2 is less than FF0 and FF1.

- ③ There is a possibility of a glitch on decode 7 at the positive-going edge of CLK 6 if the propagation delay of FF1 is less than FF0.

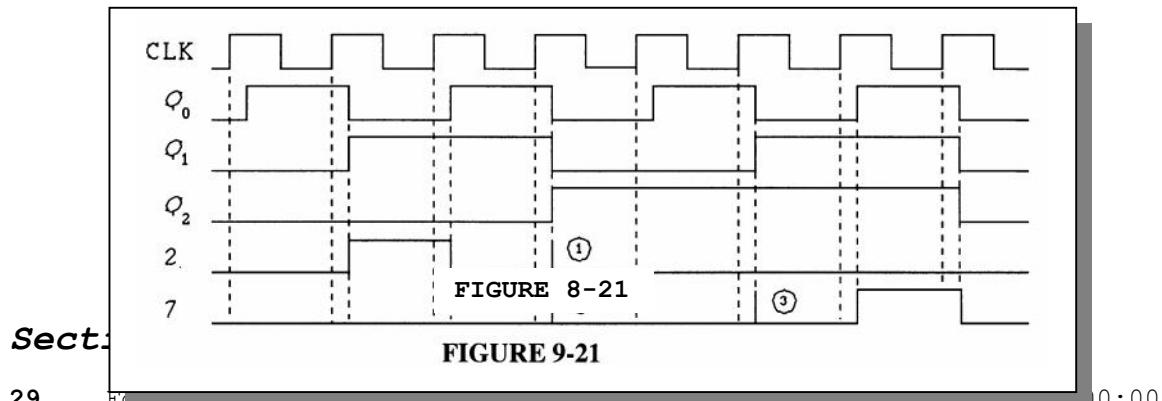
See the timing diagram in Figure 8-21 which is expanded to show the delays.

Any glitches can be prevented by using CLK as an input to both decode gates.

Secti-

29. For a 60-Hz system, the binary state of each counter after 62 60-Hz pulses are:

Hours, tens: 0001  
Hours, units: 0010  
Minutes, tens: 0000



Minutes, units: 0001  
 Seconds, tens: 0000  
 Seconds, units: 0010

30. For the digital clock, the counter output frequencies are:  
**Divide-by-60 input counter:**

$$\frac{60 \text{ Hz}}{60} = 1 \text{ Hz}$$

**Seconds counter:**

$$\frac{1 \text{ Hz}}{60} = 16.7 \text{ mHz}$$

**Minutes counter:**

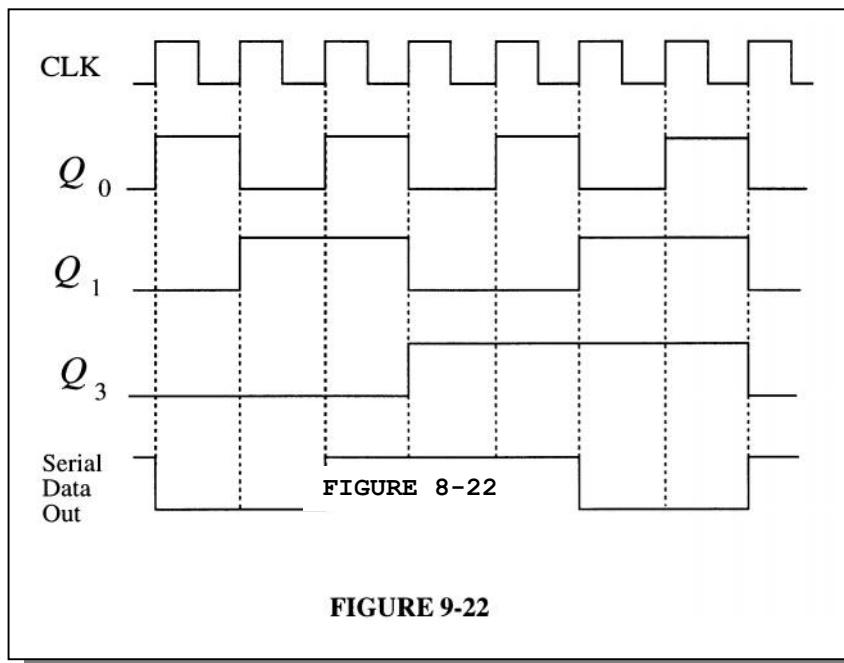
$$\frac{16.7 \text{ mHz}}{60} = 278 \text{ } \mu\text{Hz}$$

**Hours counter:**

$$\frac{278 \text{ } \mu\text{Hz}}{12} = 23.1 \text{ } \mu\text{Hz}$$

31.  $53 + 37 - 26 = 64$

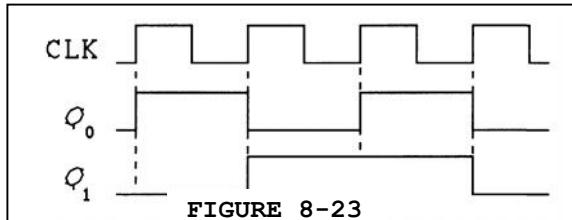
32. See Figure 8-22.



### **Section 8-9 Troubleshooting**

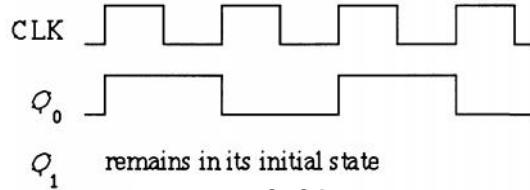
**33.** (a)  $Q_0$  and  $Q_1$  will not change due to the clock shorted to ground at FF0.

(b)  $Q_0$  being open does not affect normal operation. See Figure 8-23.



**FIGURE 8-23**

(c) See Figure 9-23



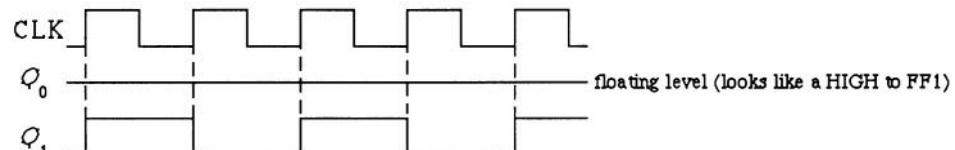
**FIGURE 9-23**

(d) No, the clock acts as a HIGH.

(e) A shorted  $K$  input will pull all  $J$  and  $K$  inputs LOW and the counter will not change from its initial state.

**34.** (a)  $Q_0$  and  $Q_1$  will not change from initial states.

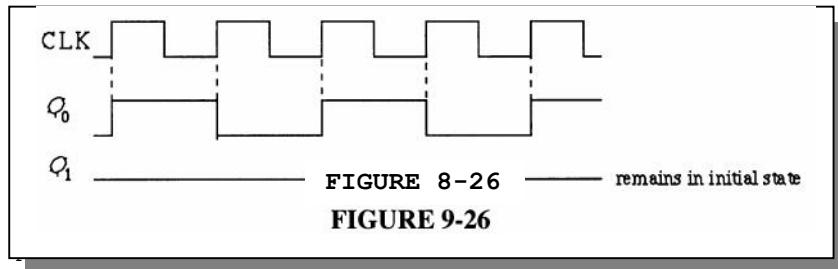
(b) See Figure 8-25.



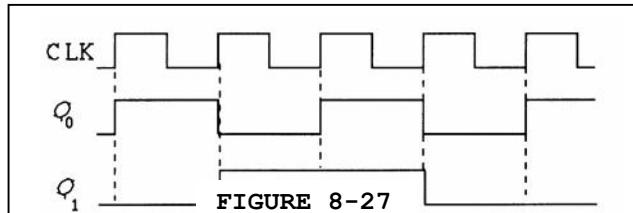
**FIGURE 8-25**

**FIGURE 9-25**

(c) See Figure 8-26.

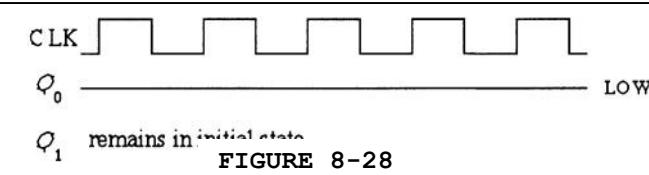


(d)



**FIGURE 9-27**

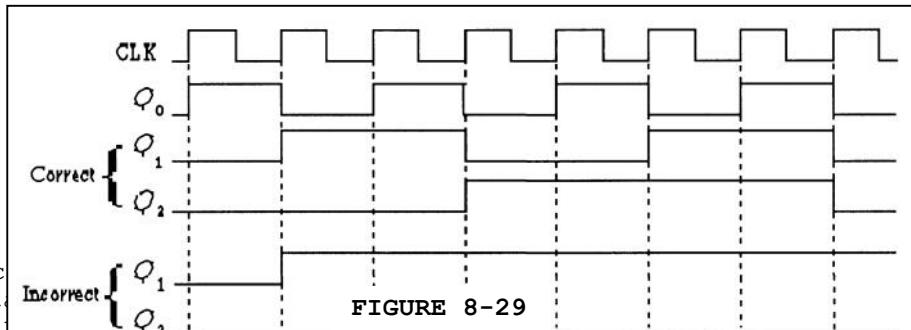
(e) Both **J** and **K** inputs of FF1 were connected to ground, producing a no-change condition. **Q<sub>0</sub>** also grounded. See figure 8-28.



**FIGURE 9-28**

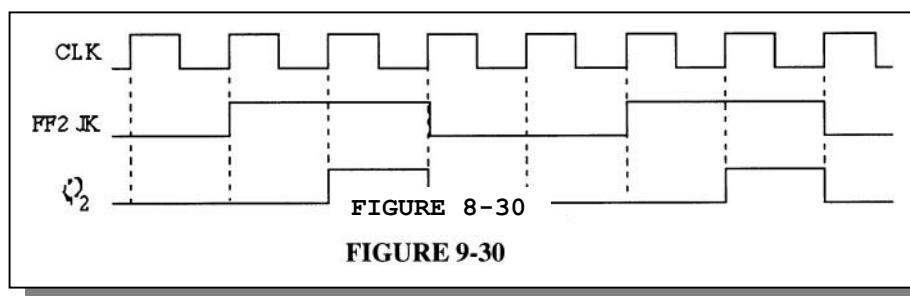
35. First, determine if **Q<sub>0</sub>**, **Q<sub>1</sub>**, and **Q<sub>2</sub>** are correct. If **Q<sub>0</sub>** is correct, then **Q<sub>1</sub>** and **Q<sub>2</sub>** are incorrect in Figure 8-29. In one case, **Q<sub>1</sub>** is correct and **Q<sub>2</sub>** is incorrect. See Figure 8-29.

Since **Q<sub>1</sub>** goes HIGH and stays HIGH, FF1 must be in the SET state (**J** = 1, **K** = 0). There must be a wiring error at the **J** and **K** inputs to FF1; **K** must be connected to ground rather than to the **J** input.



36. Since **Q<sub>0</sub>** is correct, **Q<sub>1</sub>** and **Q<sub>2</sub>** are incorrect. In one case, **Q<sub>1</sub>** is correct and **Q<sub>2</sub>** is incorrect. See Figure 8-29.

37. If the outputs shown in Figure 8-30 are as



**38.** Number of states = 40,000

$$f_{out} = \frac{5 \text{ MHz}}{40,000} = 125 \text{ Hz}$$

76.2939 Hz is not correct. The faulty division factor is

$$\frac{5 \text{ MHz}}{76.2939 \text{ Hz}} = 65,536$$

Obviously, the counter is going through all of its states. This means that the  $63C0_{16}$  on its parallel inputs is not being loaded. Possible faults are:

- Inverter output is stuck HIGH or open.
- RCO output of last counter is stuck LOW.

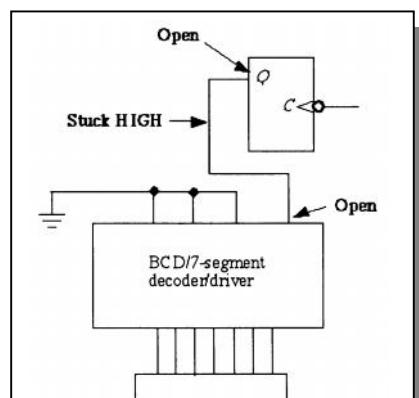
**39.**

| Stage | Open | Loaded Count | $f_{out}$                                                                                  |
|-------|------|--------------|--------------------------------------------------------------------------------------------|
| 1     | 0    | 63C1         | 250.006                                                                                    |
| 1     | 1    | 63C2         | Hz                                                                                         |
| 1     | 2    | 63C4         | 250.012                                                                                    |
| 1     | 3    | 63C8         | Hz                                                                                         |
| 2     | 0    | 63D0         | 250.025                                                                                    |
| 2     | 1    | 63E0         | Hz                                                                                         |
| 2     | 2    | 63C0         | 250.050                                                                                    |
| 2     | 3    | 63C0         | Hz                                                                                         |
| 3     | 0    | 63C0         | 250.100                                                                                    |
| 3     | 1    | 63C0         | Hz                                                                                         |
| 3     | 2    | 67C0         | 250.200                                                                                    |
| 3     | 3    | 6BC0         | Hz                                                                                         |
| 4     | 0    | 73C0         | 250 Hz                                                                                     |
| 4     | 1    | 63C0         | 250 Hz                                                                                     |
| 4     | 2    | 63C0         | 250 Hz                                                                                     |
| 4     | 3    | E3C0         | 250 Hz<br>256.568<br>Hz<br>263.491<br>Hz<br>278.520<br>Hz<br>250 Hz<br>250 Hz<br>1.383 kHz |

**40.** ▪ The flip-flop output is stuck HIGH or open.

- The least significant BCD/7-segment input is open.

See Figure 8-31.



**FIGURE 8-31**

- 41.** Th DIV 6 is the tens of minutes counter.  $Q_1$  open causes a continuous apparent HIGH output to the decode 6 gate and to the BCD/7-segment decoder/driver.

The apparent counter sequence is shown in the table.

| <b>Actual State of Ctr.</b> | <b>Apparent state</b> |       |       |       |
|-----------------------------|-----------------------|-------|-------|-------|
|                             | $Q_3$                 | $Q_2$ | $Q_1$ | $Q_0$ |
| 0                           | 0                     | 0     | 1     | 0     |
| 1                           | 0                     | 0     | 1     | 1     |
| 2                           | 0                     | 0     | 1     | 0     |
| 3                           | 0                     | 0     | 1     | 1     |
| 4                           | 0                     | 1     | 1     | 0     |

The decode 6 gate interprets count 4 as a 6 (0110) and clears the counter back to 0 (actually 0010). Thus, the apparent (not actual) sequence is as shown in the table.

- 42.** There are several possible causes of the malfunction. First check power to all units. Other possible faults are listed below.

- Sensor Latch
  - Action: Disconnect entrance sensor and pulse sensor input.
  - Observation: Latch should SET.
  - Conclusion: If latch does not SET, replace it.
- NOR gate
  - Action: Pulse sensor input.
  - Observation: Pulse on gate output.
  - Conclusion: If there is no pulse, replace gate.
- Counter
  - Action: Pulse sensor input.
  - Observation: Counter should advance.
  - Conclusion: If counter does not advance, replace it.
- Output Interface
  - Action: Pulse sensor input until terminal count is reached.
  - Observation: FULL indication and gate lowered
  - Conclusion: No FULL indication or if gate does not lower, replace interface.
- Sensor/Cable
  - Action: Try to activate sensor.
  - Observation: If all previous checks are OK, sensor or cable is faulty.
  - Conclusion: Replace sensor or cable.

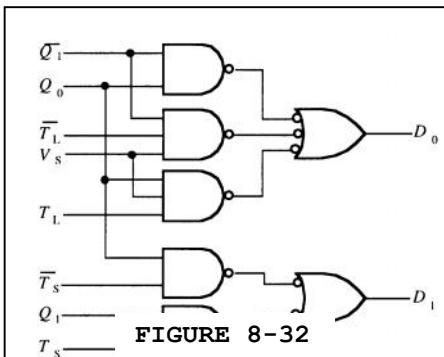
### **Digital System Application**

- 43.** The expressions for the  $D_0$  and the  $D_1$  flip-flop inputs in the sequential logic portion of the system were developed for the System

Assignment Activities 1 and 2. Figure 8-32 shows the NAND implementation.

$$D_0 = \overline{Q_1}Q_0 + Q_1\overline{T_L}V_S + Q_0T_LV_S$$

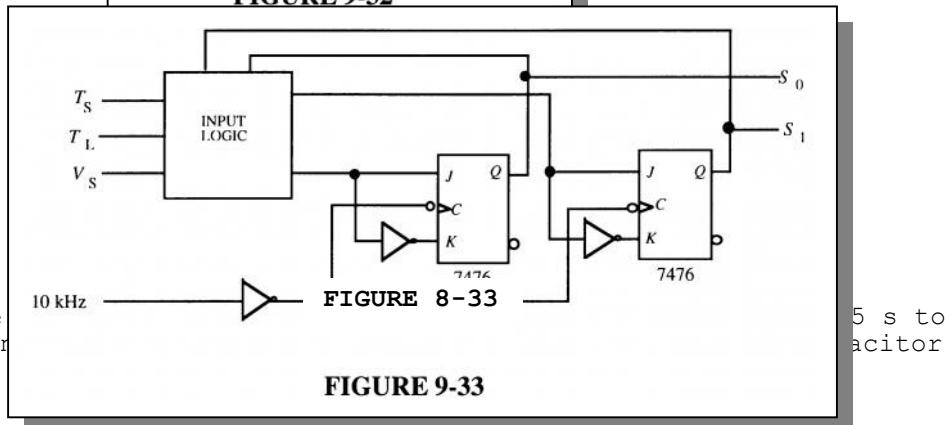
$$D_1 = Q_0\overline{T_L} + Q_1T_S$$



**FIGURE 8-32**

**44.** See Figure 8-33

**FIGURE 9-32**



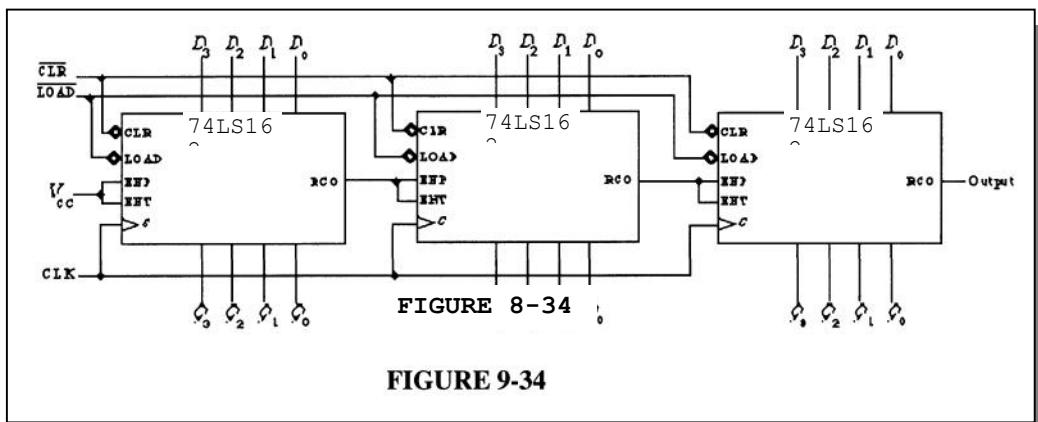
**FIGURE 9-33**

**45.** The time  
60 s by in  
value by

$$\frac{60\text{ s}}{25\text{ s}} = 2.4 \text{ times}$$

### **Special Design Problems**

**46.** See Figure 8-34.



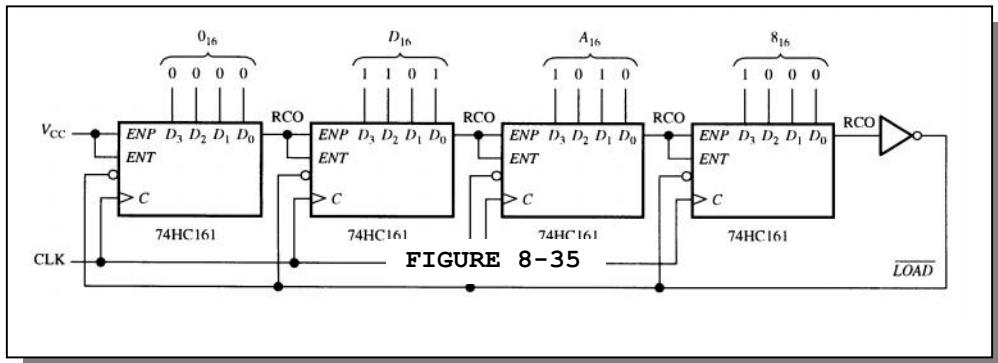
**FIGURE 8-34**

**47.**

Preset the counter to 30,000 so that it counts from 30,000 up to 65,536 on each full cycle, thus producing a sequence of 30,000 states (modulus 30,000).

$$35,536 = 1000101011010000_2 = \mathbf{8AD0}_{16}$$

See Figure 8-35.

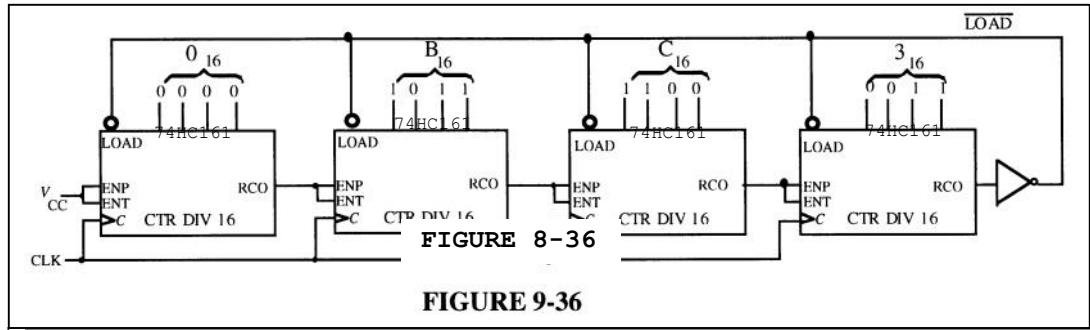


48.

Preset the counter to 10,000 so that it counts from 10,000 up to 65,536 on each full cycle, thus producing a sequence of 50,000 states (modulus 50,000).

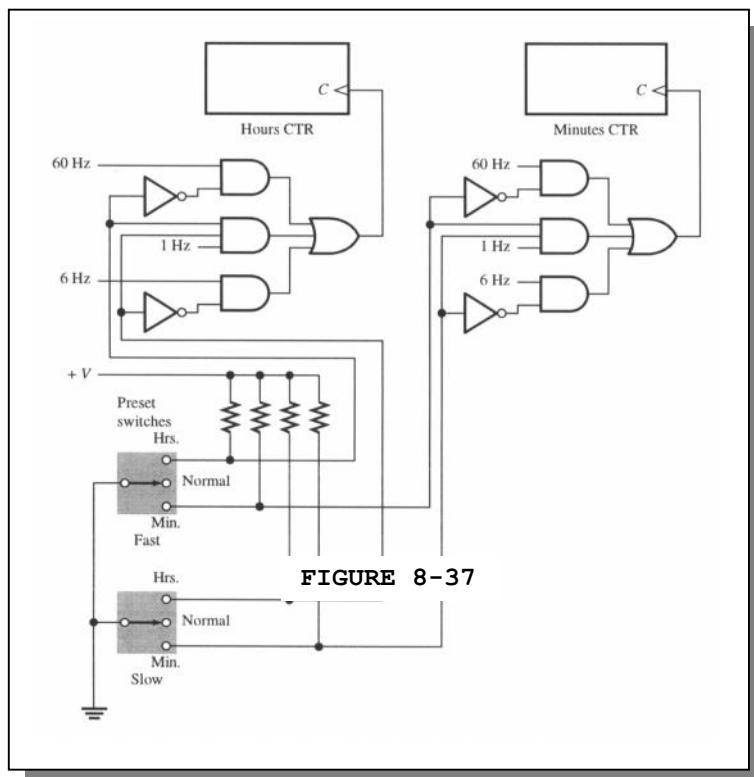
$$15,536 = 11110010110000_2 = \mathbf{3CB0}_{16}$$

See Figure 8-36.

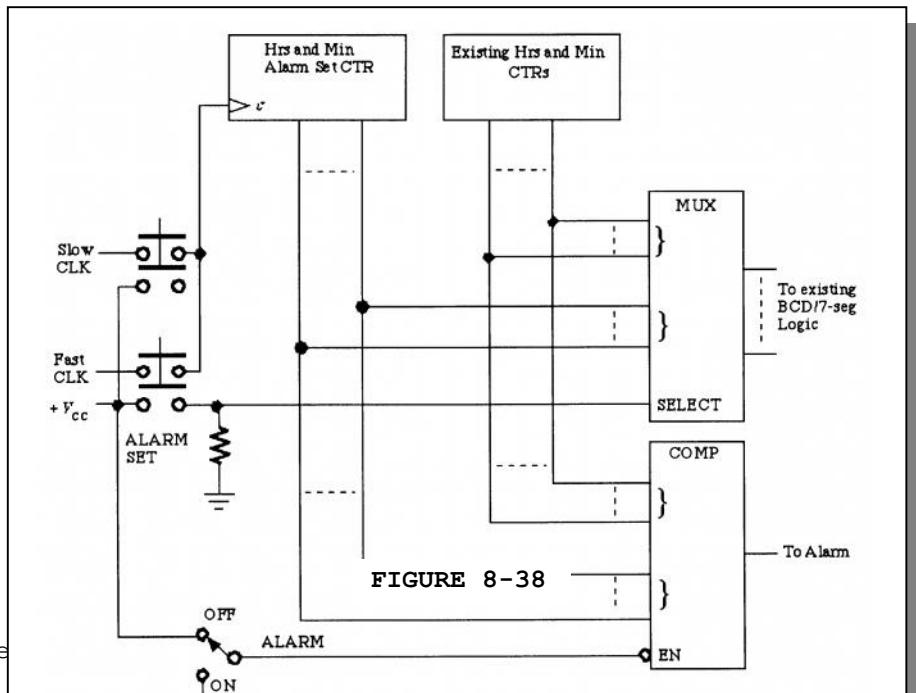


49.

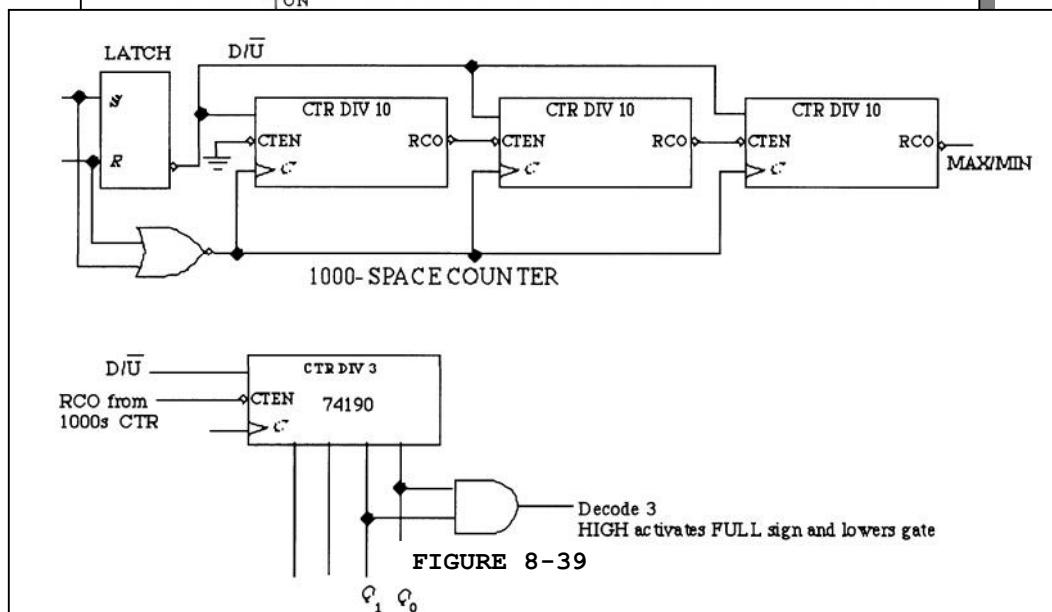
counter is not preset. One possible implementation is shown in Figure 8-37.



50. See Figure 8-38.

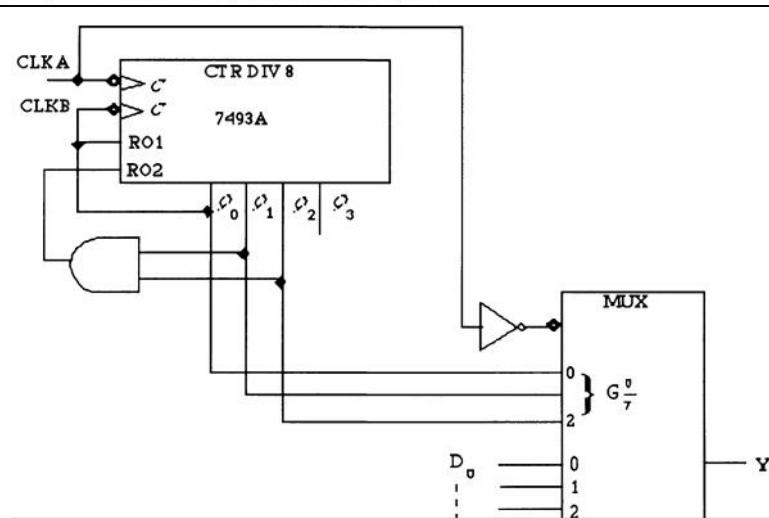


51. See



52.

Add to 1000-space counter for expansion to 3000 spaces



**FIGURE 8-40**

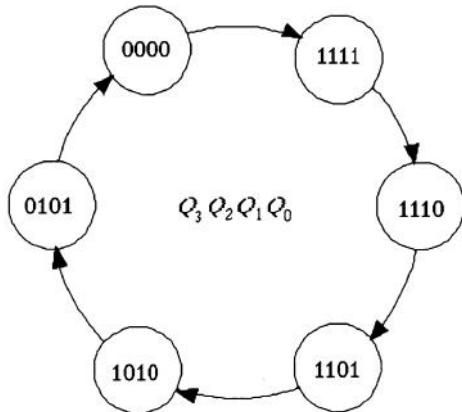
**53. NEXT-STATE TABLE**

| Present State |       |       |       | Next State |       |       |       |
|---------------|-------|-------|-------|------------|-------|-------|-------|
| $Q_3$         | $Q_2$ | $Q_1$ | $Q_0$ | $Q_3$      | $Q_2$ | $Q_1$ | $Q_0$ |
| 0             | 0     | 0     | 0     | 1          | 1     | 1     | 1     |
| 1             | 1     | 1     | 1     | 1          | 1     | 1     | 0     |
| 1             | 1     | 1     | 0     | 1          | 1     | 0     | 1     |
| 1             | 1     | 0     | 1     | 1          | 0     | 1     | 0     |
| 1             | 0     | 1     | 0     | 0          | 1     | 0     | 1     |
| 0             | 1     | 0     | 1     | 0          | 0     | 0     | 0     |

TRANSITION TABLE

| Output State Transitions |        |        |        | Flip-flop Inputs |          |          |          |
|--------------------------|--------|--------|--------|------------------|----------|----------|----------|
| $Q_3$                    | $Q_2$  | $Q_1$  | $Q_0$  | $J_3K_3$         | $J_2K_2$ | $J_1K_1$ | $J_0K_0$ |
| 0 to 0                   | 0 to 0 | 0 to 0 | 0 to 0 | 1X               | 1X       | 1X       | 1X       |
| 1                        | 1      | 1      | 1      | X0               | X0       | X0       | X1       |
| 1 to 1                   | 1 to 1 | 1 to 1 | 1 to 1 | X0               | X0       | X1       | 1X       |
| 1                        | 1      | 1      | 0      | X0               | X1       | 1X       | X1       |
| 1 to 1                   | 1 to 1 | 1 to 0 | 0 to 0 | X1               | 1X       | X1       | 1X       |
| 1                        | 1      | 0      | 1      | 0X               | X1       | 0X       | X1       |
| 1 to 1                   | 1 to 0 | 0 to 1 | 1 to 0 |                  |          |          |          |
| 1                        | 0      | 1      | 0      |                  |          |          |          |
| 1 to 0                   | 0 to 1 | 1 to 0 | 0 to 1 |                  |          |          |          |
| 0 to 0                   | 1 to 0 | 0 to 0 | 1 to 0 |                  |          |          |          |
| 0                        | 0      | 0      | 0      |                  |          |          |          |

See Figure 8-41.



The desired sequence

| $Q_3\ Q_2$ | $\infty$ | 01 | 11 | 10 |
|------------|----------|----|----|----|
| 00         | 1        | X  | X  | X  |
| 01         | X        | 0  | X  | X  |
| 11         | X        | X  | X  | X  |
| 10         | X        | X  | X  | X  |

$$J_3 = \bar{Q}_2$$

| $Q_3\ Q_2$ | $\infty$ | 01 | 11 | 10 |
|------------|----------|----|----|----|
| 00         | 1        | X  | X  | X  |
| 01         | X        | X  | X  | X  |
| 11         | X        | X  | X  | X  |
| 10         | X        | X  | X  | 1  |

$$J_2 = 1$$

| $Q_3\ Q_2$ | $\infty$ | 01 | 11 | 10 |
|------------|----------|----|----|----|
| 00         | 1        | X  | X  | X  |
| 01         | X        | 0  | X  | X  |
| 11         | X        | 1  | X  | X  |
| 10         | X        | X  | X  | 1  |

$$J_1 = \bar{Q}_2 + Q_3$$

$$J_0 = 1$$

| $Q_3\ Q_2$ | $\infty$ | 01 | 11 | 10 |
|------------|----------|----|----|----|
| 00         | X        | X  | X  | X  |
| 01         | X        | X  | X  | X  |
| 11         | X        | 0  | 0  | 0  |
| 10         | X        | X  | X  | 1  |

$$K_3 = \bar{Q}_2$$

| $Q_3\ Q_2$ | $\infty$ | 01 | 11 | 10 |
|------------|----------|----|----|----|
| 00         | X        | X  | X  | X  |
| 01         | X        | 1  | X  | X  |
| 11         | X        | 1  | 0  | 0  |
| 10         | X        | X  | X  | X  |

$$K_2 = \bar{Q}_1$$

| $Q_3\ Q_2$ | $\infty$ | 01 | 11 | 10 |
|------------|----------|----|----|----|
| 00         | X        | X  | X  | X  |
| 01         | X        | X  | X  | X  |
| 11         | X        | X  | 0  | 1  |
| 10         | X        | X  | X  | 1  |

$$K_1 = \bar{Q}_0$$

| $Q_3\ Q_2$ | $\infty$ | 01 | 11 | 10 |
|------------|----------|----|----|----|
| 00         | X        | X  | X  | X  |
| 01         | X        | 1  | X  | X  |
| 11         | X        | 1  | 1  | X  |
| 10         | X        | X  | X  | X  |

$$K_0 = 1$$

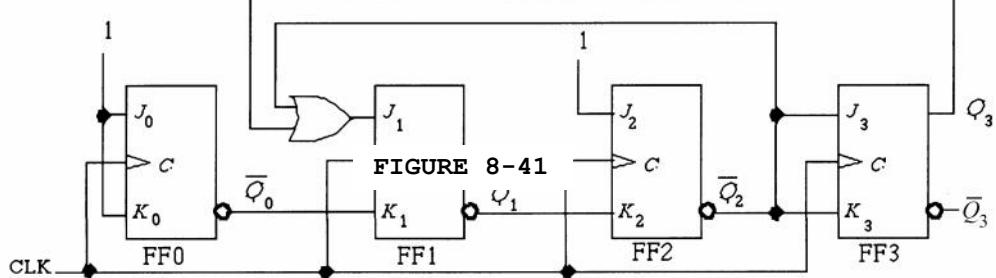
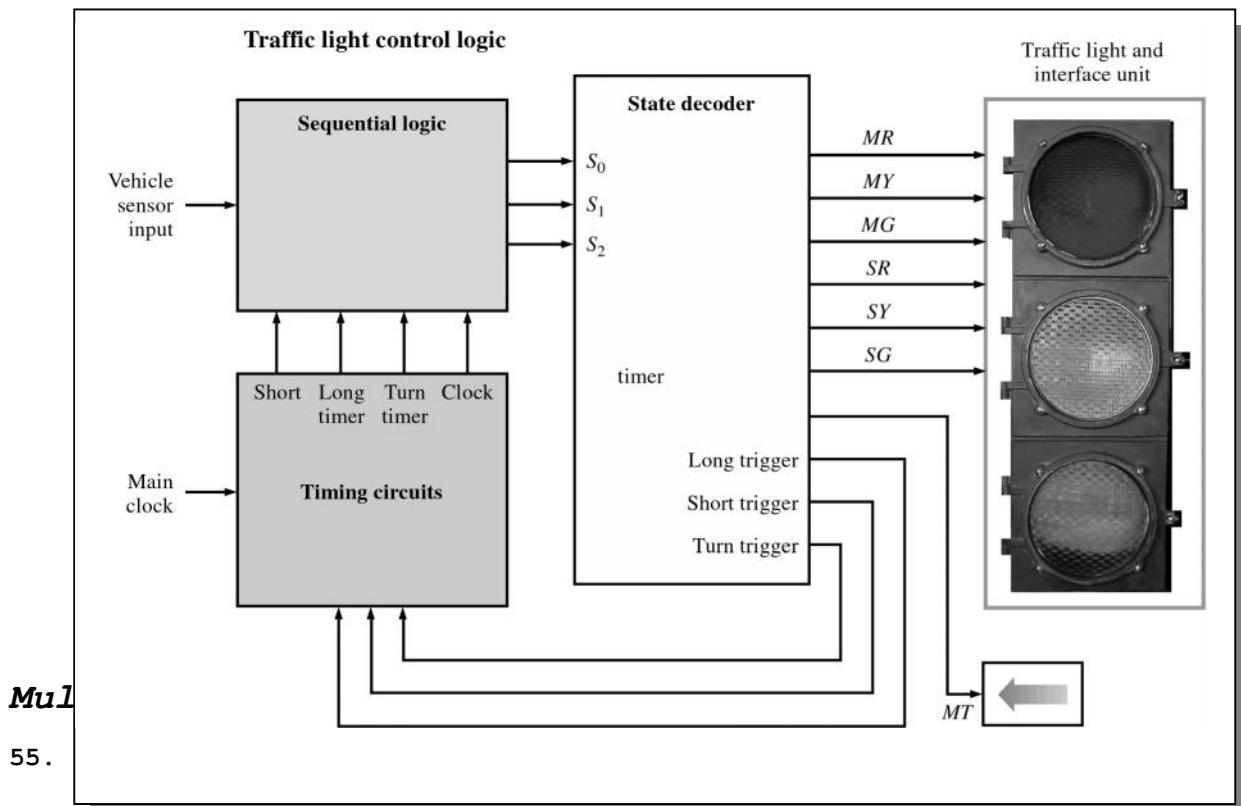


FIGURE 9-41

54. See Figure 8-42.



---

## CHAPTER 9

### SHIFT REGISTERS

---

#### Section 9-1 Basic Shift Register Functions

1. Shift registers store binary data in a series of flip-flops or other storage elements.
2. 1 byte = 8 bits; 2 bytes = 16 bits

#### Section 9-2 Serial In/Serial Out Shift Registers

3. See Figure 9-1.

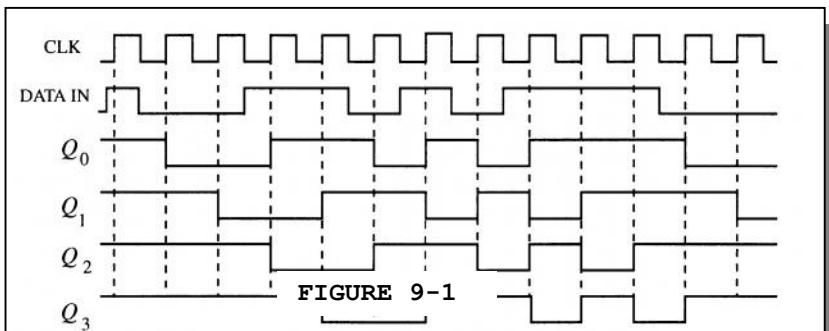


FIGURE 9-1

4. See Figure 10-1.

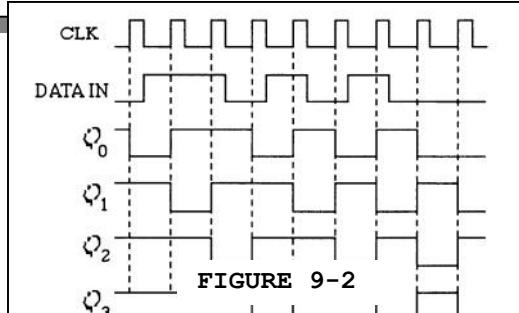


FIGURE 10-1

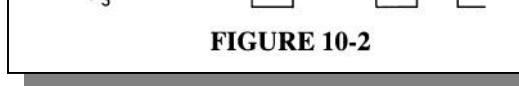
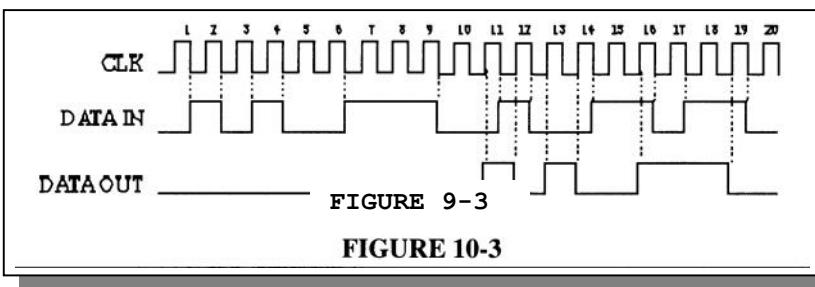


FIGURE 9-2

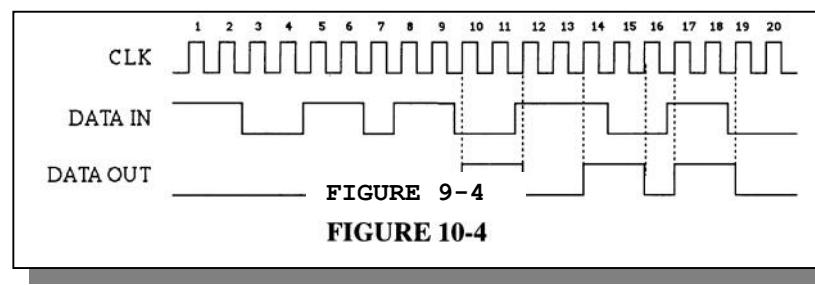
5.

|        |             |
|--------|-------------|
| Initia | 10100111100 |
| lly    | 0           |
| CLK 1  | 01010011110 |
| CLK 2  | 0           |
| CLK 3  | 00101001111 |
| CLK 4  | 0           |
| CLK 5  | 00010100111 |
| CLK 6  | 1           |
| CLK 7  | 00001010011 |
| CLK 8  | 1           |
| CLK 9  | 10000101001 |
| CLK 10 | 1           |
| CLK 11 | 11000010100 |
| CLK 12 | 1           |
|        | 11100001010 |
|        | 0           |
|        | 01110000101 |
|        | 0           |
|        | 00111000010 |
|        | 1           |
|        | 00011100001 |
|        | 0           |
|        | 10001110000 |
|        | 1           |
|        | 11000111000 |
|        | 0           |

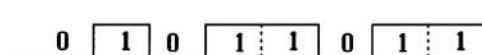
6. See Figure 9-3.



7.



8.



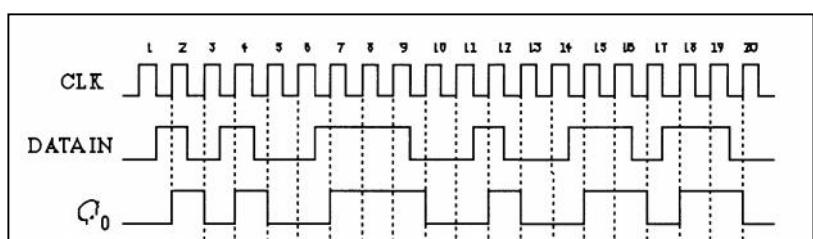
The number binary number 11011010 is stored in the register after eight clock pulses. **FIGURE 9-5**

**FIGURE 10-5**

**Section**

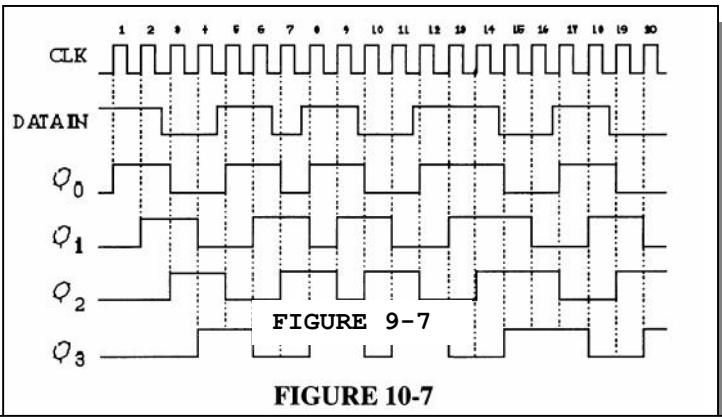
**Registers**

9. See Figure 9-6.



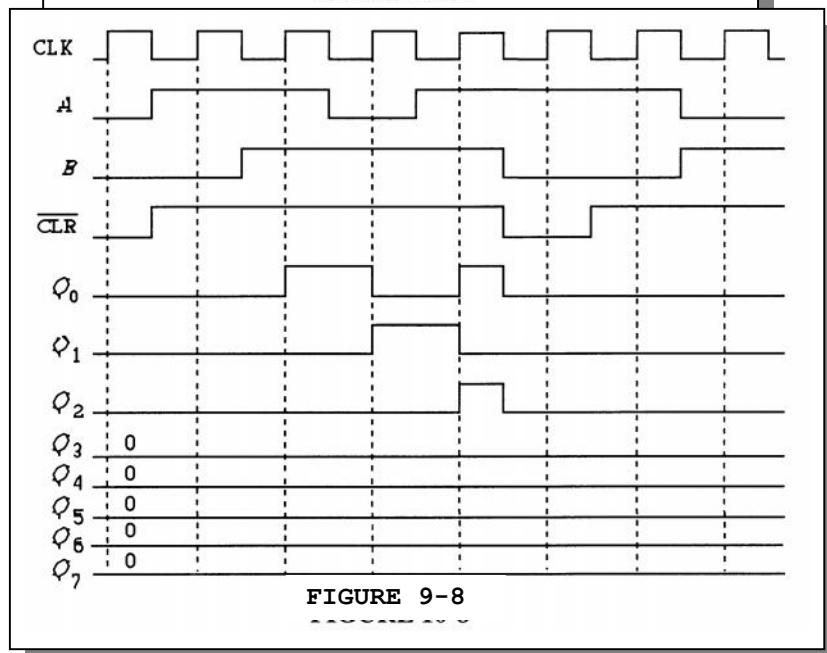
**FIGURE 9-6**

10. See Figure 9-7.



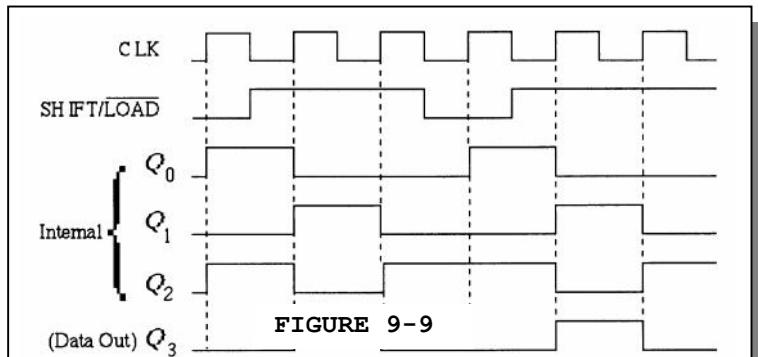
11. See Figure

**FIGURE 10-7**

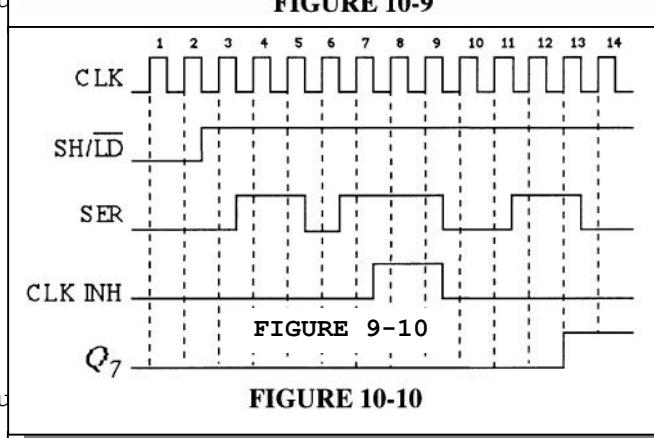


**Section 9-4 Parallel In/Serial Out Shift Registers**

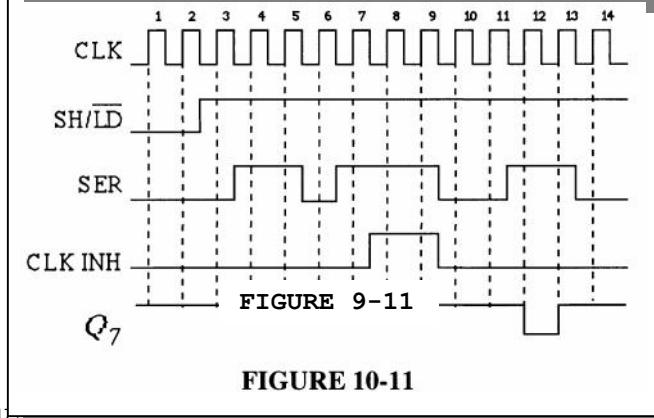
12. See Figure 9-9.



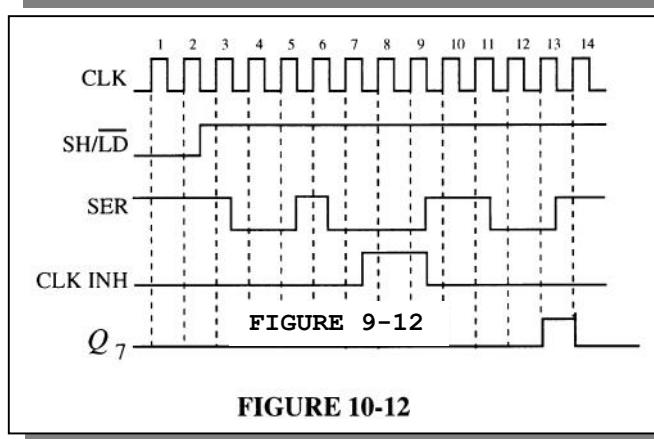
13. See Figure



14. See Figure

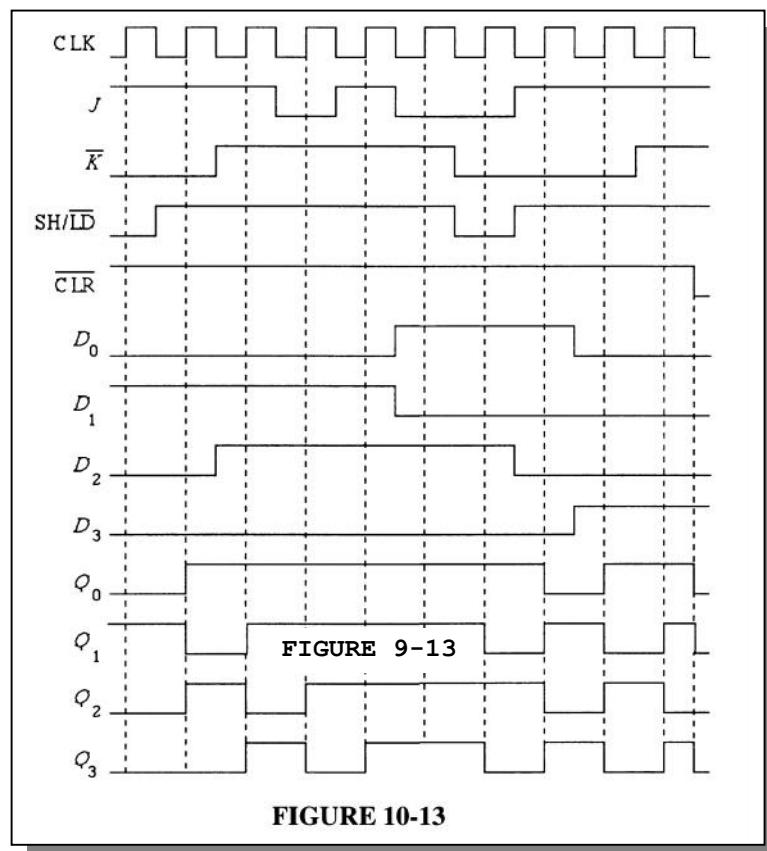


15. See Figure



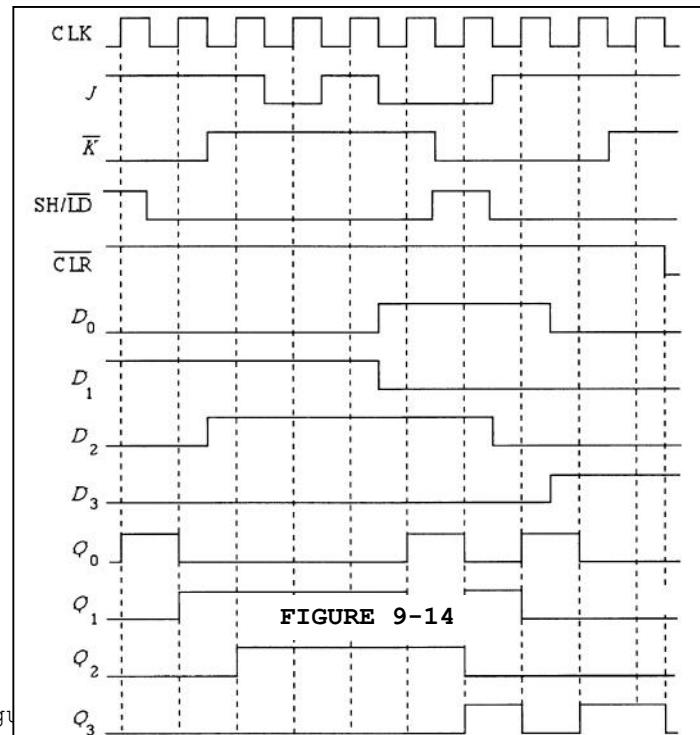
**Section 9-5 Parallel In/Parallel Out Shift Registers**

16. See Figure 9-13.

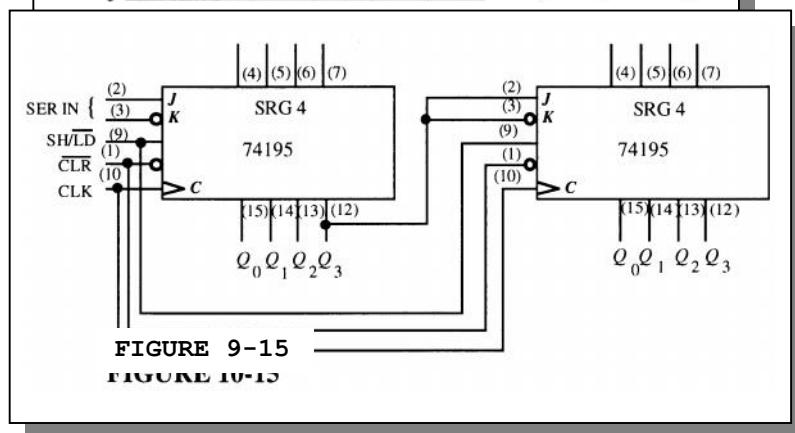


**FIGURE 10-13**

17. See Figure 9-14.



18. See Figure



**Section 9-6 Bidirectional Shift Registers**

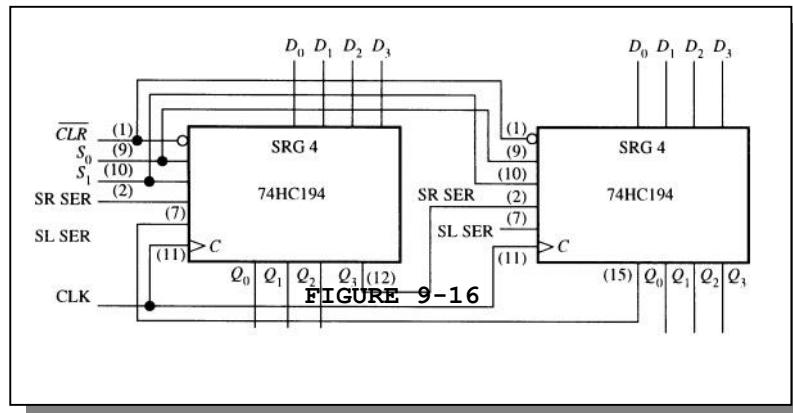
19.

|           |          |            |
|-----------|----------|------------|
| Initially | 01001100 |            |
| (76)      | 10011000 | Shift left |
| CLK 1     | 01001100 | Shift      |
| CLK 2     | 00100110 | right      |
| CLK 3     | 00010011 | Shift      |
| CLK 4     | 00100110 | right      |
| CLK 5     | 01001100 | Shift      |
| CLK 6     | 00100110 | right      |
| CLK 7     | 01001100 | Shift left |
| CLK 8     | 00100110 | Shift left |
| CLK 9     | 01001100 | Shift      |
| CLK 10    | 10011000 | right      |
| CLK 11    |          | Shift left |
|           |          | right      |
|           |          | Shift left |
|           |          | Shift left |

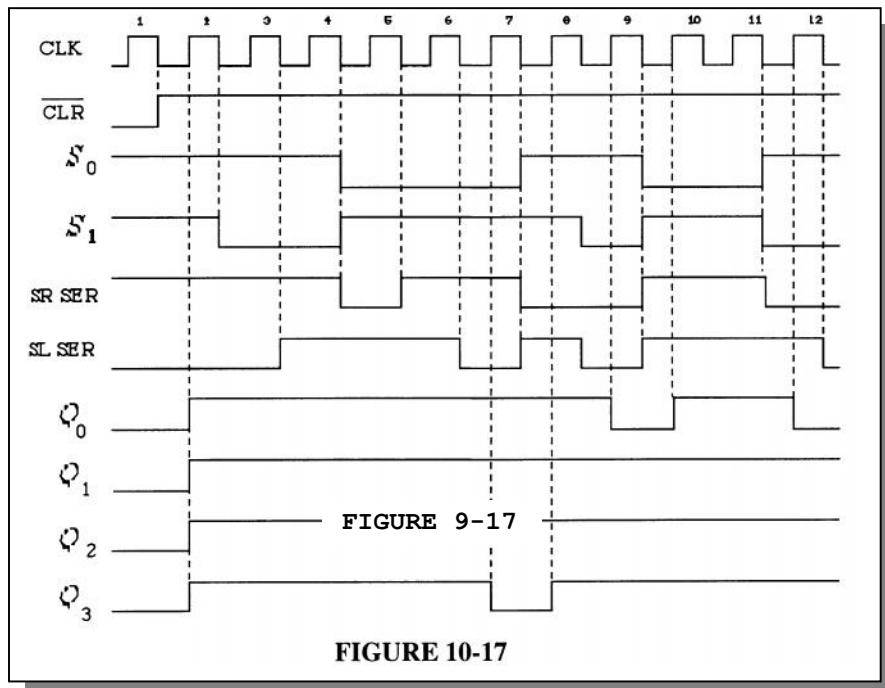
20.

|           |          |             |
|-----------|----------|-------------|
| Initially | 01001100 |             |
| (76)      | 00100110 | Shift       |
| CLK 1     | 00010011 | right       |
| CLK 2     | 00001001 | Shift       |
| CLK 3     | 00010010 | right       |
| CLK 4     | 00100100 | Shift       |
| CLK 5     | 01001000 | right       |
| CLK 6     | 00100100 | Shift left  |
| CLK 7     | 01001000 | Shift left  |
| CLK 8     | 10010000 | Shift left  |
| CLK 9     | 00100000 | Shift       |
| CLK 10    | 00010000 | right       |
| CLK 11    | 00001000 | Shift left  |
| CLK 12    |          | Shift left  |
|           |          | Shift right |
|           |          | Shift right |

21. See Figure 9-16.



22. See Figure 9-17.



**Section**

23. (a)

**FIGURE 10-17**

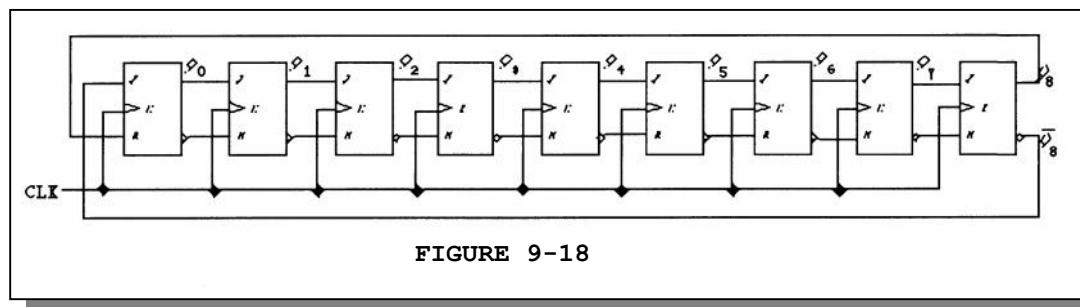
$$(c) \quad 2n = 14 \\ n = 7$$

$$(d) \quad 2n = 16 \\ n = 8$$

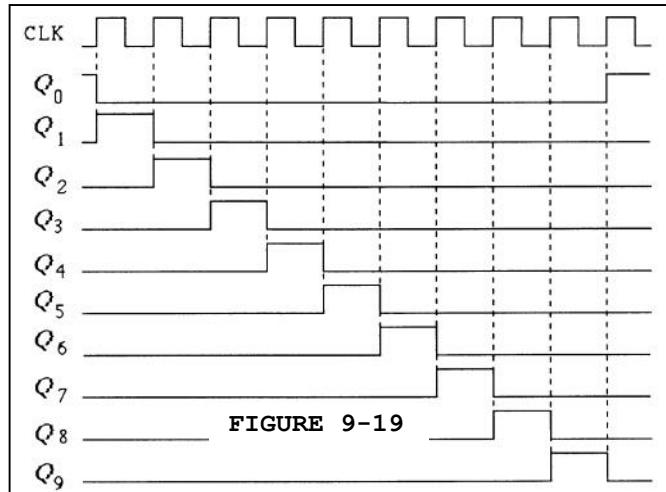
24.  $2n = 18$ ;  $n = 9$  flip-flops

| $Q_0$ | $Q_1$ | $Q_2$ | $Q_3$ | $Q_4$ | $Q_5$ | $Q_6$ | $Q_7$ | $Q_8$ |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |
| 1     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |
| 1     | 1     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |
| 1     | 1     | 1     | 0     | 0     | 0     | 0     | 0     | 0     |
| 1     | 1     | 1     | 1     | 0     | 0     | 0     | 0     | 0     |
| 1     | 1     | 1     | 1     | 1     | 0     | 0     | 0     | 0     |
| 1     | 1     | 1     | 1     | 1     | 1     | 0     | 0     | 0     |
| 1     | 1     | 1     | 1     | 1     | 1     | 1     | 0     | 0     |
| 1     | 1     | 1     | 1     | 1     | 1     | 1     | 1     | 0     |
| 1     | 1     | 1     | 1     | 1     | 1     | 1     | 1     | 1     |
| 0     | 1     | 1     | 1     | 1     | 1     | 1     | 1     | 1     |
| 0     | 0     | 1     | 1     | 1     | 1     | 1     | 1     | 1     |
| 0     | 0     | 0     | 1     | 1     | 1     | 1     | 1     | 1     |
| 0     | 0     | 0     | 0     | 1     | 1     | 1     | 1     | 1     |
| 0     | 0     | 0     | 0     | 0     | 1     | 1     | 1     | 1     |
| 0     | 0     | 0     | 0     | 0     | 0     | 0     | 1     | 1     |
| 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |
| 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |

See Figure 9-18.



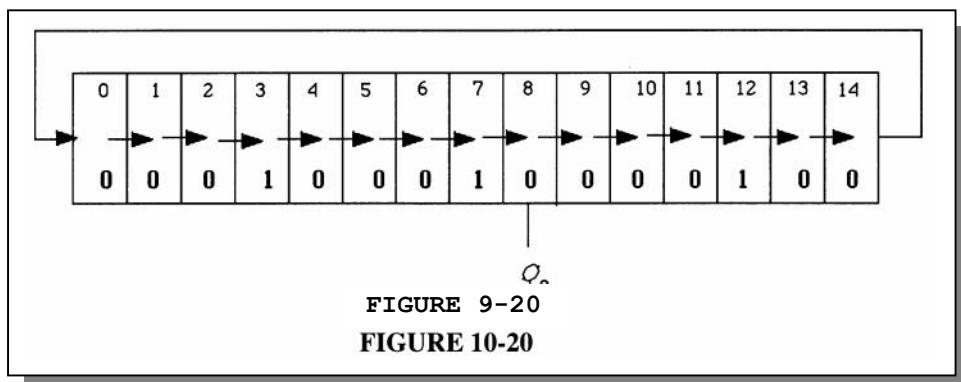
25. See Figure 9-19.



26. A 15-bit  
stages RE.  
See Figure

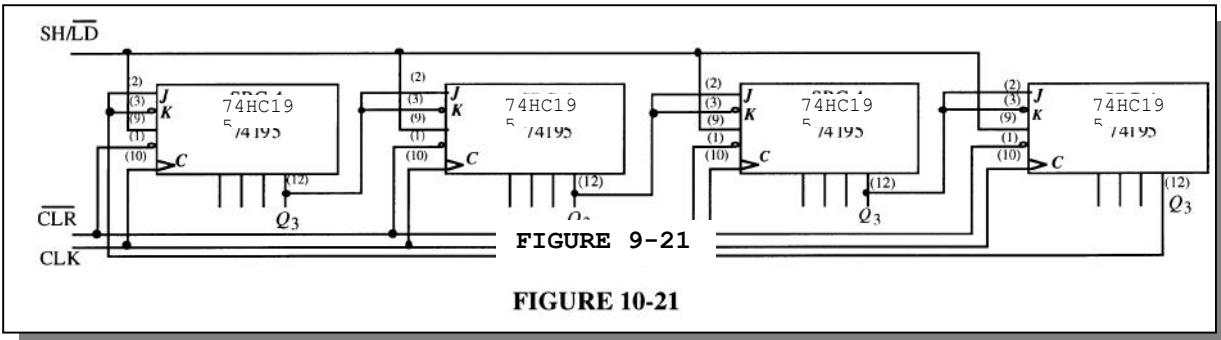
**FIGURE 10-19**

and the remaining



### **Section 9-8 Shift Register Applications**

27. See Figure 9-21.



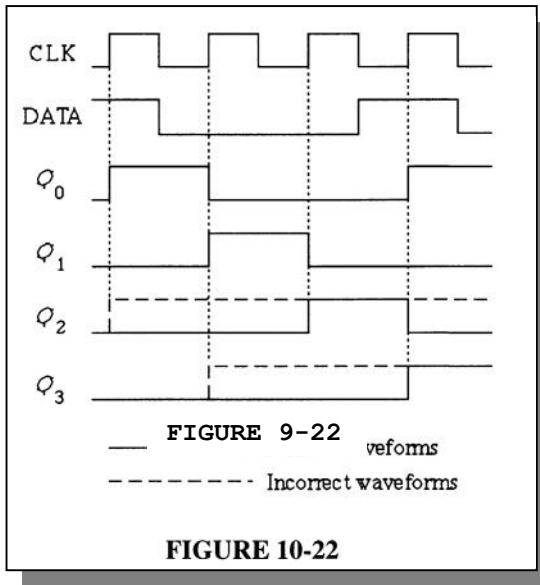
**FIGURE 10-21**

the ring counter when power is turned on.

29. An incorrect code may be produced.

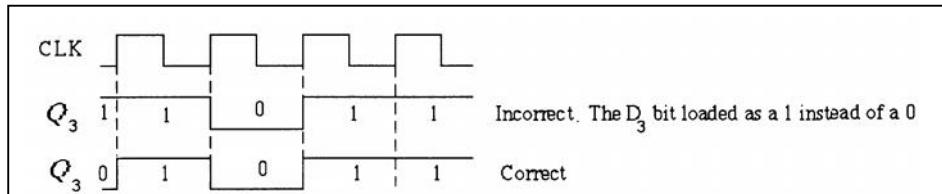
### **Section 9-10 Troubleshooting**

30.  $Q_2$  goes HIGH on the first clock pulse indicating that the  $D$  input is open. See Figure 9-22.



**FIGURE 10-22**

- 31.** Since the LSB flip-flop works during serial shift, the problem is most likely in gate G3. An open  $D_3$  input at G3 will cause the observed waveform. See Figure 9-23.



**FIGURE 9-23**

**FIGURE10-23**

- 32.** It [ ] open inverter input will keep the inverter output LOW thus disabling all of the shift-left control gates G5, G6, G7, and G8.

- 33.** (a) No clock at switch closure due to faulty NAND gate or one-shot; open clock input to key code register; open SH/LD input to key code register.

(b) The diode in the third row is open;  $Q_2$  output of ring counter is open.

(c) The NAND (negative-OR) gate input connected to the first column is shorted to ground or open, preventing a switch closure transition.

(d) The "2" input to the column encoder is open.

- 34.** 1. Number the switches in the matrix according to the following format:

|    |    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|----|
| 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  |
| 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 |
| 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 |
| 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 |
| 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 |
| 49 | 50 | 51 | 52 | 53 | 54 | 55 | 56 |
| 57 | 58 | 59 | 60 | 61 | 62 | 63 | 64 |

2. Depress switches one at a time and observe the key code output according to the following Table 1.

| Switch number | Key Code Register |       |       |       |       |       |
|---------------|-------------------|-------|-------|-------|-------|-------|
|               | $Q_0$             | $Q_1$ | $Q_2$ | $Q_3$ | $Q_4$ | $Q_5$ |
| 1             | 0                 | 1     | 1     | 0     | 1     | 1     |
| 2             | 0                 | 1     | 1     | 1     | 0     | 1     |
| 3             | 0                 | 1     | 1     | 0     | 0     | 1     |
| 4             | 0                 | 1     | 1     | 1     | 1     | 0     |
| 5             | 0                 | 1     | 1     | 0     | 1     | 0     |
| 6             | 0                 | 1     | 1     | 1     | 0     | 0     |
| 7             | 0                 | 1     | 1     | 0     | 0     | 0     |
| 8             | 0                 | 1     | 1     | 1     | 1     | 1     |
| 9             | 1                 | 0     | 1     | 0     | 1     | 1     |
| 10            | 1                 | 0     | 1     | 1     | 0     | 1     |
| 11            | 1                 | 0     | 1     | 0     | 0     | 1     |
| 12            | 1                 | 0     | 1     | 1     | 1     | 0     |
| 13            | 1                 | 0     | 1     | 0     | 1     | 0     |
| 14            | 1                 | 0     | 1     | 1     | 0     | 0     |
| 15            | 1                 | 0     | 1     | 0     | 0     | 0     |
| 16            | 1                 | 0     | 1     | 1     | 1     | 1     |
| 17            | 0                 | 0     | 1     | 0     | 1     | 1     |
| 18            | 0                 | 0     | 1     | 1     | 0     | 1     |
| 19            | 0                 | 0     | 1     | 0     | 0     | 1     |
| 20            | 0                 | 0     | 1     | 1     | 1     | 0     |
| 21            | 0                 | 0     | 1     | 0     | 1     | 0     |
| 22            | 0                 | 0     | 1     | 1     | 0     | 0     |
| 23            | 0                 | 0     | 1     | 0     | 0     | 0     |
| 24            | 0                 | 0     | 1     | 1     | 1     | 1     |
| 25            | 1                 | 1     | 0     | 0     | 1     | 1     |
| 26            | 1                 | 1     | 0     | 1     | 0     | 1     |
| 27            | 1                 | 1     | 0     | 0     | 0     | 1     |
| 28            | 1                 | 1     | 0     | 1     | 1     | 0     |
| 29            | 1                 | 1     | 0     | 0     | 1     | 0     |
| 30            | 1                 | 1     | 0     | 1     | 0     | 0     |
| 31            | 1                 | 1     | 0     | 0     | 0     | 0     |
| 32            | 1                 | 1     | 0     | 1     | 1     | 1     |
| 33            | 0                 | 1     | 0     | 0     | 1     | 1     |
| 34            | 0                 | 1     | 0     | 1     | 0     | 1     |
| 35            | 0                 | 1     | 0     | 0     | 0     | 1     |
| 36            | 0                 | 1     | 0     | 1     | 1     | 0     |
| 37            | 0                 | 1     | 0     | 0     | 1     | 0     |
| 38            | 0                 | 1     | 0     | 1     | 0     | 0     |
| 39            | 0                 | 1     | 0     | 0     | 0     | 0     |
| 40            | 0                 | 1     | 0     | 1     | 1     | 1     |
| 41            | 1                 | 0     | 0     | 0     | 1     | 1     |
| 42            | 1                 | 0     | 0     | 1     | 0     | 1     |
| 43            | 1                 | 0     | 0     | 0     | 0     | 1     |
| 44            | 1                 | 0     | 0     | 1     | 1     | 0     |
| 45            | 1                 | 0     | 0     | 0     | 1     | 0     |
| 46            | 1                 | 0     | 0     | 1     | 0     | 1     |
| 47            | 1                 | 0     | 0     | 0     | 0     | 0     |
| 48            | 1                 | 0     | 0     | 1     | 1     | 1     |
| 49            | 0                 | 0     | 0     | 0     | 1     | 1     |
| 50            | 0                 | 0     | 0     | 1     | 0     | 1     |
| 51            | 0                 | 0     | 0     | 0     | 0     | 1     |
| 52            | 0                 | 0     | 0     | 1     | 1     | 0     |
| 53            | 0                 | 0     | 0     | 0     | 1     | 0     |
| 54            | 0                 | 0     | 0     | 1     | 0     | 0     |
| 55            | 0                 | 0     | 0     | 0     | 0     | 0     |
| 56            | 0                 | 0     | 0     | 1     | 1     | 1     |
| 57            | 1                 | 1     | 1     | 0     | 1     | 1     |
| 58            | 1                 | 1     | 1     | 1     | 0     | 1     |
| 59            | 1                 | 1     | 1     | 0     | 1     | 1     |

|    |   |   |   |   |   |   |
|----|---|---|---|---|---|---|
| 60 | 1 | 1 | 1 | 1 | 1 | 0 |
| 61 | 1 | 1 | 1 | 0 | 1 | 0 |
| 62 | 1 | 1 | 1 | 1 | 0 | 0 |
| 63 | 1 | 1 | 1 | 0 | 0 | 0 |
| 64 | 1 | 1 | 1 | 1 | 1 | 1 |

**TABLE 1**

- 35.** (a) Contents of Data Output Register remain constant.  
 (b) Contents of both registers do not change.  
 (c) Third stage output of Data Output Register remains HIGH.  
 (d) Clock generator is disabled after each pulse by the flip-flop being continuously SET and then RESET.

### **Digital System Application**

**36.** The purpose of the Security Code logic is to accept a 4-digit code, compare it with a stored code, and if the codes match, to disarm the system for entry.

**37.** The states of shift registers A and C after two correct key closures are:

Shift Register A: 1001  
 Shift Register C: 00000100

**38.** The states of shift registers A and B after each key closure when entering 7645 are:

**After key 7 is pressed:**

Shift register A contains 0111  
 Shift register B contains 11000

**After key 6 is pressed:**

Shift register A contains 0110  
 Shift register B contains 11100

**After key 4 is pressed:**

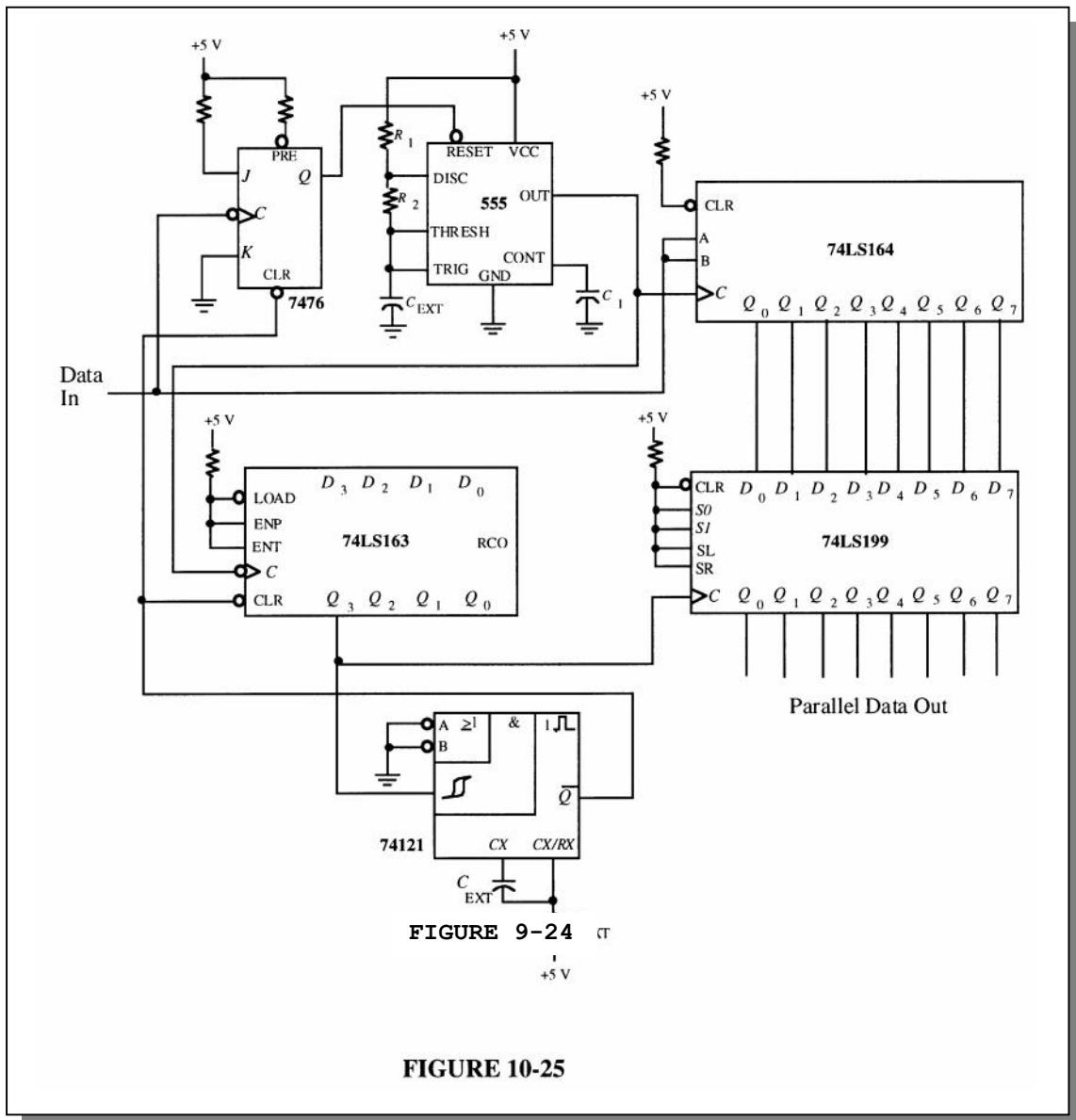
Shift register A contains 0100  
 Shift register B contains 11110

**After key 5 (an incorrect entry) is pressed:**

Shift register A contains 0000  
 Shift register B contains 10000

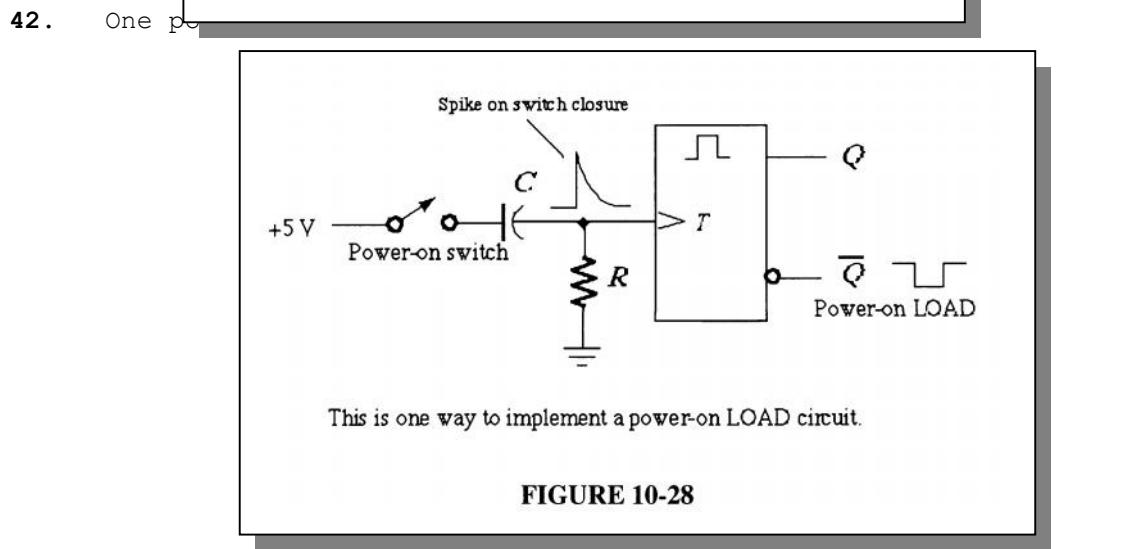
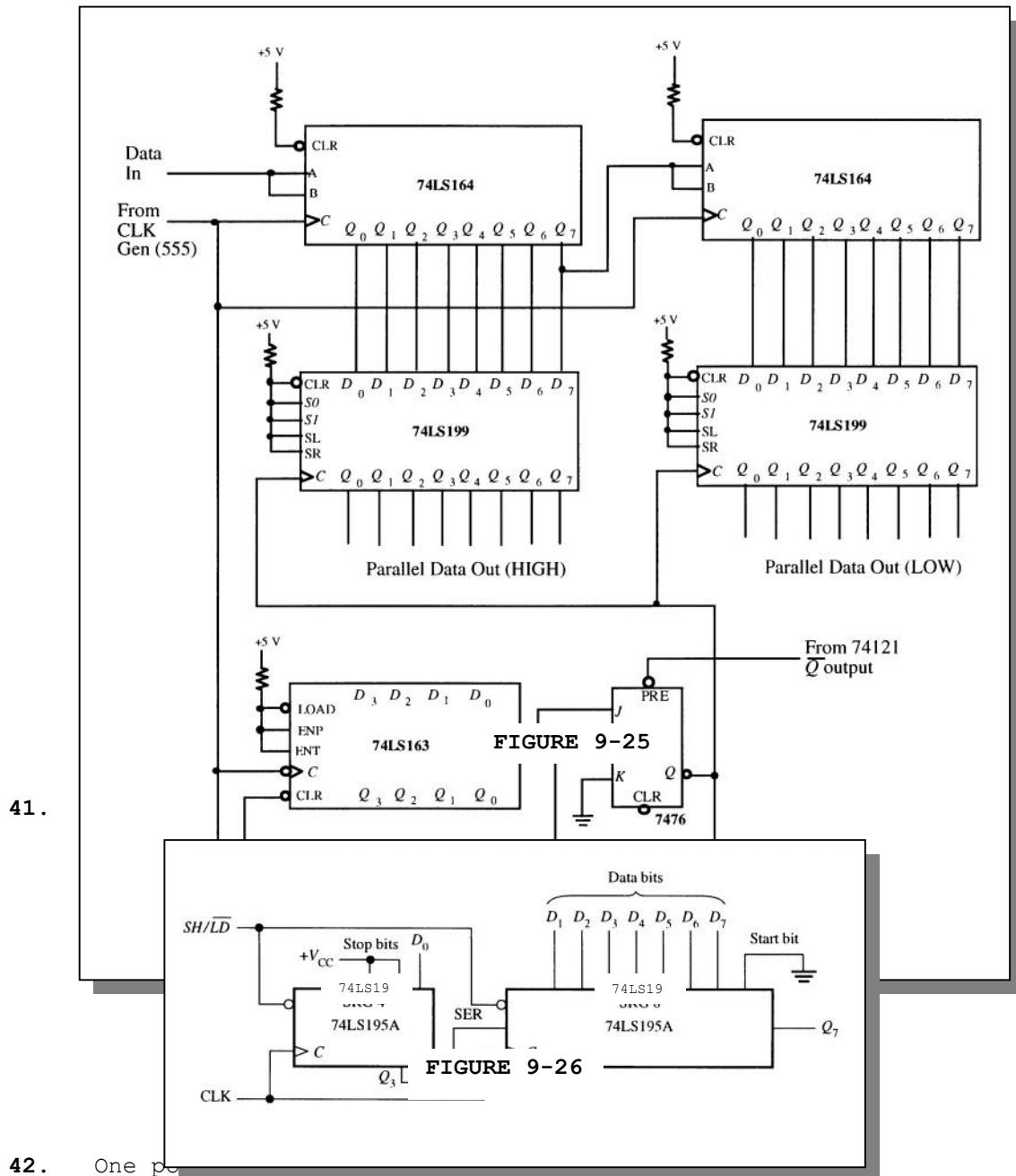
**Special Design Problems**

39. See Figure 9-24.



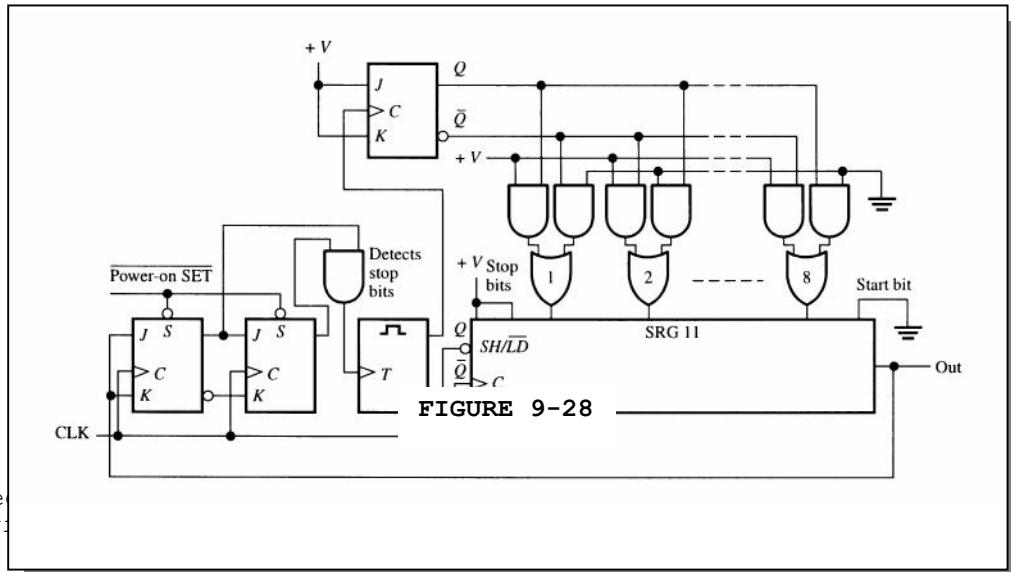
**FIGURE 10-25**

40. Figure 9-25 shows only the 74LS164, 74LS199, and 74LS163 portions of the circuit that require modification for 16-bit conversion.



**FIGURE 9-27**

- 43.** See Figure 9-28.



- 44.** Re  
Reg:

### Multisim Troubleshooting Practice

- 45.** CLK input of U3 open.  
**46.** No fault.  
**47.** Pin 14 open.  
**48.** No fault.  
**49.** CLK input of U6 open.

---

## CHAPTER 10

### MEMORY AND STORAGE

---

#### Section 10-1 Basics of Semiconductor Memory

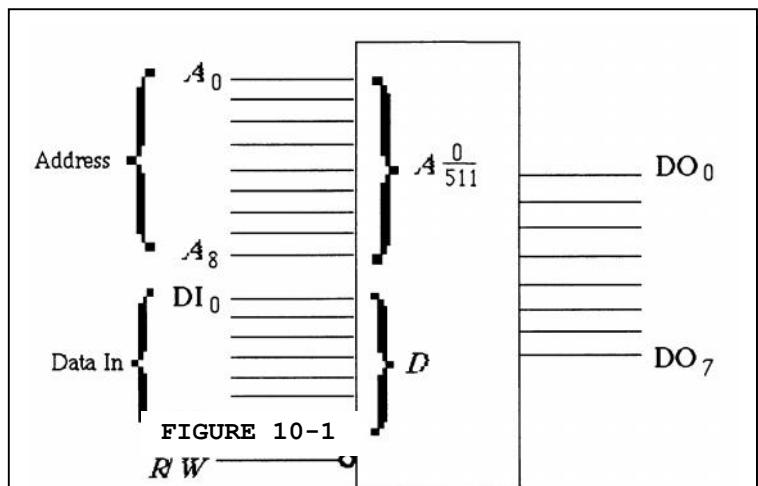
1. (a) ROM: no read/write control  
 (b) RAM
2. They are random access memories because any address can be accessed at any time. You do not have to go through all the preceding addresses to get to a specific address.
3. **Address bus** provides for transfer of address code to memory for accessing any memory location in any order for a read or a write operation.  
**Data bus** provides for transfer of data between the microprocessor and memory or input/output devices.
4. (a)  $0A_{16} = 00001010_2 = 10_{10}$   
 (b)  $3F_{16} = 00111111_2 = 63_{10}$   
 (c)  $CD_{16} = 11001101_2 = 205_{10}$

#### Section 10-2 Random-Access Memories (RAMs)

5.

|       | BIT<br>0 | BIT<br>1 | BIT<br>2 | BIT<br>3 |
|-------|----------|----------|----------|----------|
| ROW 0 | 1        | 0        | 0        | 0        |
| ROW 1 | 0        | 0        | 0        | 0        |
| ROW 2 | 0        | 0        | 1        | 0        |
| ROW 3 | 0        | 0        | 0        | 0        |

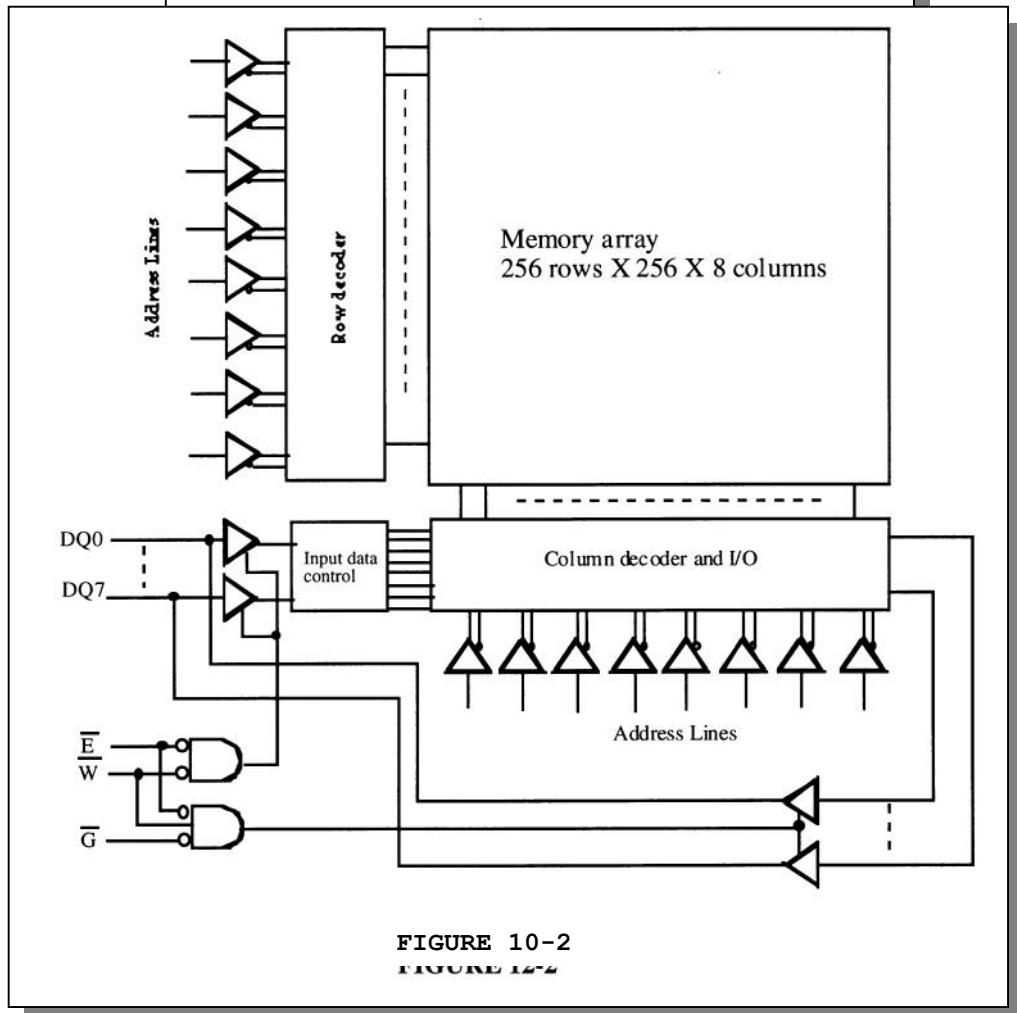
6. See Figure 10-1.



7.  $64k \times 8$

8. See Figu

**FIGURE 12-1**



9. The difference between SRAM and DRAM is that data in a SRAM are stored in latches or flip-flops indefinitely as long as power is applied while data in a DRAM are stored in capacitors which require periodic refreshing to retain the stored data.

10. The bit capacity of a DRAM with 12 address lines is

$$2^{2 \times 12} = 2^{24} = 16,777,216 \text{ bits} = 16 \text{ Mbits}$$

### **Section 10-3 Read-Only Memories (ROMs)**

11.

| <b>Input</b>          |                       | <b>Outputs</b>        |                       |                       |                       |
|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| <b>s</b>              |                       | <i>O</i> <sub>3</sub> | <i>O</i> <sub>2</sub> | <i>O</i> <sub>1</sub> | <i>O</i> <sub>0</sub> |
| <i>A</i> <sub>1</sub> | <i>A</i> <sub>0</sub> |                       |                       |                       |                       |
| 0                     | 0                     | 0                     | 1                     | 0                     | 1                     |
| 0                     | 1                     | 1                     | 0                     | 0                     | 1                     |
| 1                     | 0                     | 1                     | 1                     | 1                     | 0                     |
| 1                     | 1                     | 0                     | 0                     | 1                     | 0                     |

12.

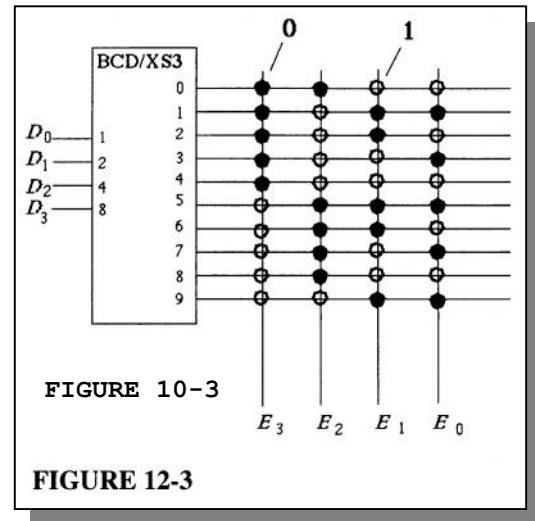
| <b>Inputs</b>         |                       |                       | <b>Outputs</b>        |                       |                       |                       |
|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| <i>A</i> <sub>2</sub> | <i>A</i> <sub>1</sub> | <i>A</i> <sub>0</sub> | <i>O</i> <sub>3</sub> | <i>O</i> <sub>2</sub> | <i>O</i> <sub>1</sub> | <i>O</i> <sub>0</sub> |
| 0                     | 0                     | 0                     | 0                     | 1                     | 0                     | 0                     |
| 0                     | 0                     | 1                     | 1                     | 1                     | 1                     | 1                     |
| 0                     | 1                     | 0                     | 1                     | 0                     | 1                     | 1                     |
| 0                     | 1                     | 1                     | 1                     | 0                     | 0                     | 1                     |
| 1                     | 0                     | 0                     | 1                     | 1                     | 1                     | 0                     |
| 1                     | 0                     | 1                     | 1                     | 0                     | 0                     | 0                     |
| 1                     | 1                     | 0                     | 0                     | 0                     | 1                     | 1                     |
| 1                     | 1                     | 1                     | 0                     | 1                     | 0                     | 1                     |

13.

| <b>BCD</b>            |                       |                       |                       | <b>Excess-3</b>       |                       |                       |                       |
|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| <i>D</i> <sub>3</sub> | <i>D</i> <sub>2</sub> | <i>D</i> <sub>1</sub> | <i>D</i> <sub>0</sub> | <i>E</i> <sub>3</sub> | <i>E</i> <sub>2</sub> | <i>E</i> <sub>1</sub> | <i>E</i> <sub>0</sub> |
| 0                     | 0                     | 0                     | 0                     | 0                     | 0                     | 1                     | 1                     |
| 0                     | 0                     | 0                     | 1                     | 0                     | 1                     | 0                     | 0                     |
| 0                     | 0                     | 1                     | 0                     | 0                     | 1                     | 0                     | 1                     |
| 0                     | 0                     | 1                     | 1                     | 0                     | 1                     | 1                     | 0                     |
| 0                     | 1                     | 0                     | 0                     | 0                     | 1                     | 1                     | 1                     |
| 0                     | 1                     | 0                     | 1                     | 1                     | 0                     | 0                     | 0                     |
| 0                     | 1                     | 1                     | 0                     | 1                     | 0                     | 0                     | 1                     |
| 0                     | 1                     | 1                     | 1                     | 1                     | 0                     | 1                     | 0                     |
| 1                     | 0                     | 0                     | 0                     | 1                     | 0                     | 1                     | 1                     |
| 1                     | 0                     | 0                     | 1                     | 1                     | 1                     | 0                     | 0                     |

See Figure 10-3.

14.  $2^{14} = 16,384$  addresses  
 $16,384 \times 8 \text{ bits} = 131,072 \text{ bits}$



### Section 10-4 Programmable ROMs (PROMs and EPROMs)

15. Blown links: 1 – 17, 19 – 23, 25 – 31, 34, 37, 38, 40 – 47, 53, 55, 58, 61, 62, 63, 65, 67, 69.

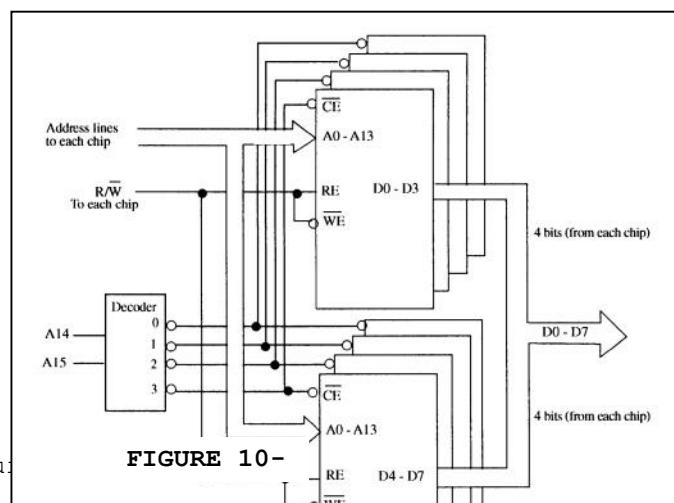
| X Input        |                |                | X <sup>i</sup> | X Output       |                |                |                |                |                |                |                |                |
|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| X <sub>2</sub> | X <sub>1</sub> | X <sub>0</sub> |                | 2 <sup>8</sup> | 2 <sup>7</sup> | 2 <sup>6</sup> | 2 <sup>5</sup> | 2 <sup>4</sup> | 2 <sup>3</sup> | 2 <sup>2</sup> | 2 <sup>1</sup> | 2 <sup>0</sup> |
| 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0              |
| 1              | 0              | 0              | 1              | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 1              |
| 2              | 0              | 1              | 0              | 8              | 0              | 0              | 0              | 0              | 1              | 0              | 0              | 0              |
| 3              | 0              | 1              | 1              | 27             | 0              | 0              | 0              | 0              | 1              | 1              | 0              | 1              |
| 4              | 1              | 0              | 0              | 64             | 0              | 0              | 1              | 0              | 0              | 0              | 0              | 0              |
| 5              | 1              | 0              | 1              | 125            | 0              | 0              | 1              | 1              | 1              | 1              | 0              | 1              |
| 6              | 1              | 1              | 0              | 216            | 0              | 1              | 1              | 0              | 1              | 1              | 0              | 0              |
| 7              | 1              | 1              | 1              | 343            | 1              | 0              | 1              | 0              | 1              | 0              | 1              | 1              |

16.

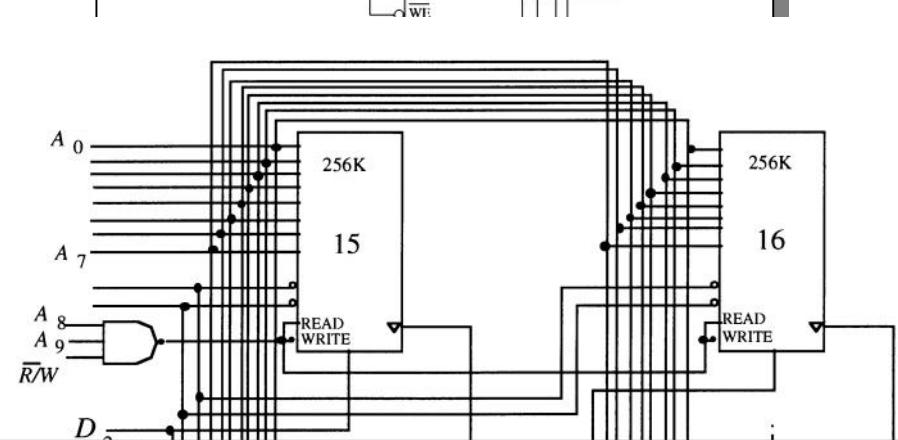
| Address                             | Contents                           |
|-------------------------------------|------------------------------------|
| A <sub>13</sub> -----A <sub>0</sub> | Q <sub>7</sub> -----Q <sub>0</sub> |
| 01001100010011                      | 10101100                           |
| 11011101011010                      | 00100101                           |
| 01011010011001                      | 10110011                           |
| 11010010001110                      | 00101000                           |
| 01010010100101                      | 10001011                           |
| 01010000110100                      | 11010101                           |
| 01001001100001                      | 11001001                           |
| 11011011100100                      | 01001001                           |
| 01101110001111                      | 01010010                           |
| 101111100011010                     | 01001000                           |
| 101011100011010                     | 11001000                           |

### Section 10-6 Memory Expansion

17. 16k × 4 DRAMs can be connected to make a 64k × 8 DRAM as shown in Figure 10-4.



18. See Figur

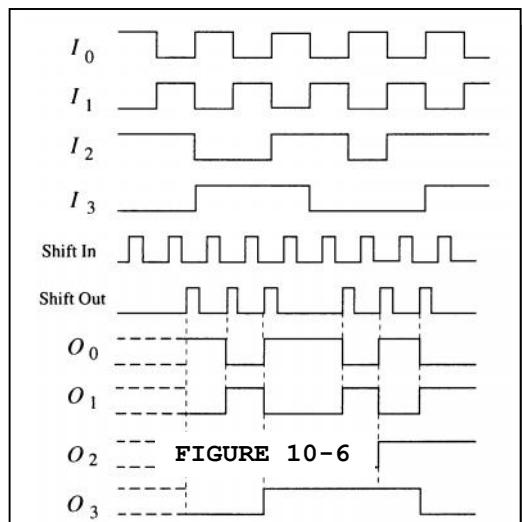


**FIGURE 10-5**

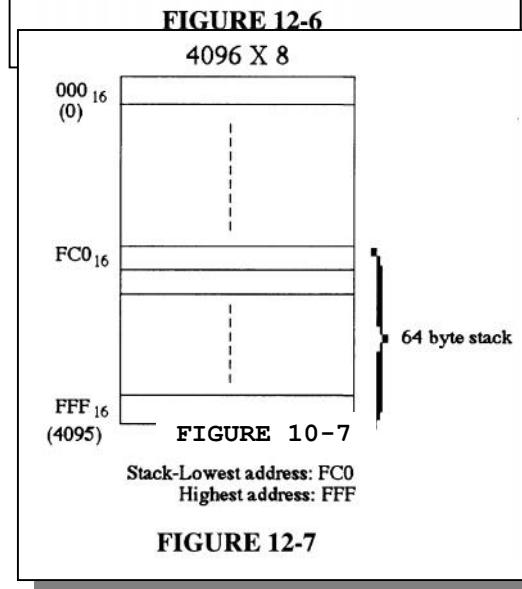
- 19.** Word length = 8 bits, word capacity = **64k words**  
Word length = 4 bits, word capacity = **256k words**

**Section 10-7 Special Types of Memories**

20. See Figure 10-6.

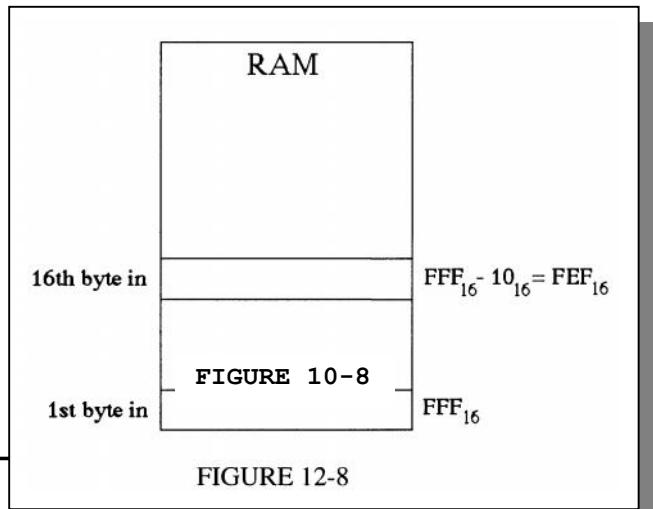


21. See Figure 1



22. The first byte goes into  $\text{FFF}_{16}$ .  
 The last byte (16th) goes into a lower address:  $16_{10} = 10_{16}$   
 $\text{FFF}_{16} - 10_{16} = \text{FEF}_{16}$

See Figure 10-8.



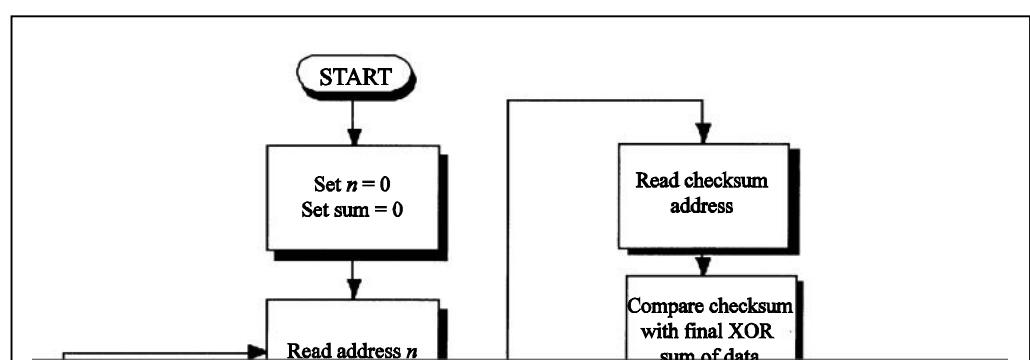
## Section 10-

## age

23. A hard disk Each track is divided into a number of sectors with each sector of a track having a physical address. Hard disks typically have from a few hundred to a few thousand tracks.
24. Seek time is the average time required to position the drive head over the track containing the desired data. The latency period is the average time required for the data to move under the drive head.
25. Magnetic tape has a longer access time than disk because data must be accessed sequentially rather than randomly.
26. A magneto-optic disk is a read/write medium using lasers and magnetic fields.  
 A CD-ROM (compact-disk ROM) is a read-only optical (laser) medium.  
 A WORM (write-once-read-many) is an optical medium in which data can be written once and read many times.

## Section 10-9 Troubleshooting

27. The correct checksum is **00100**.  
 The actual checksum is 01100. The second bit from the left is in error.
28. (a) ROM0: Low address -  $00_{16}$  High address -  $1F_{16}$   
 ROM1: Low address -  $20_{16}$  High address -  $3F_{16}$   
 ROM2: Low address -  $40_{16}$  High address -  $5F_{16}$   
 ROM3: Low address -  $60_{16}$  High address -  $7F_{16}$
- (b) Same as flow chart in Figure 10-68 in text except that the last data address is specified as  $7E_{16}$  ( $7F_{16} - 1$ ).
- (c) See Figure 10-9.

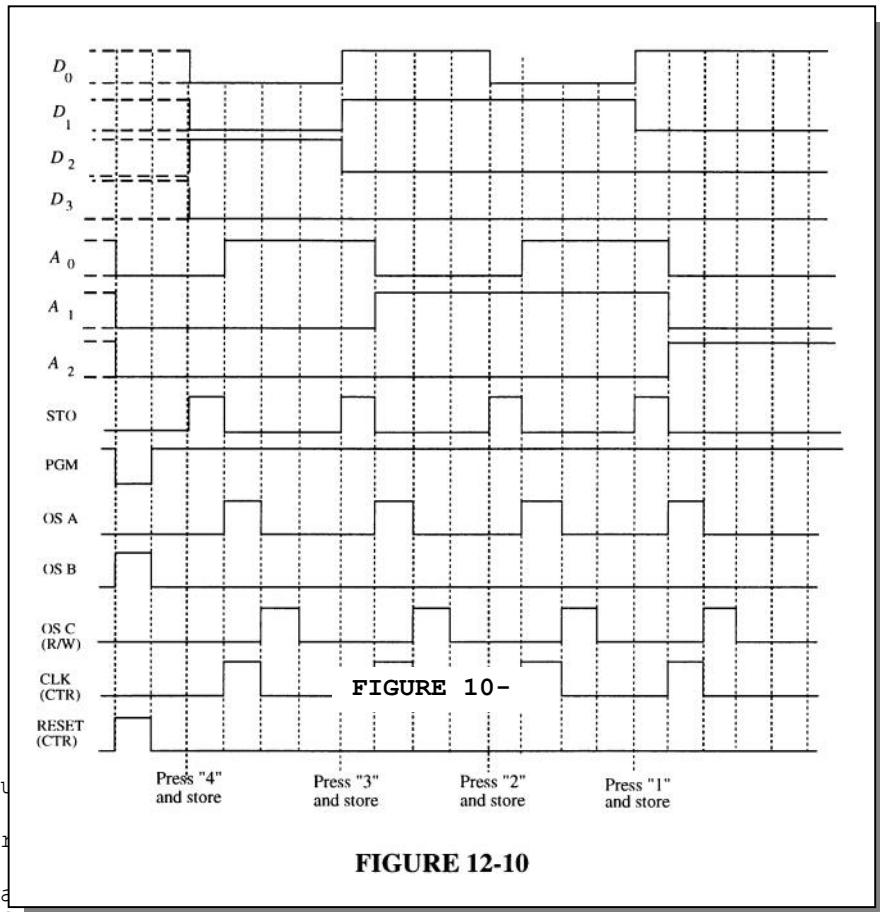


**FIGURE 10-9**

- (d) A single checksum will not isolate the faulty chip. It will only indicate that there is an error in one of the chips.
- 29.** (a)  $40_{16} - 5F_{16}$  is 64 - 95 decimal; ROM 2  
(b)  $20_{16} - 3F_{16}$  is 32 - 63 decimal; ROM 1  
(c)  $00_{16} - 7F_{16}$  is 0 - 127 decimal; All ROMs

**Digital System Application**

30. See Figure 10-10.



31. The cou  
32. The pur  
33. An adva  
A disadvantage is that a PROM cannot be readily reprogrammed with a new entry code.
- code.  
**FIGURE 12-10**

### **Special Design Problems**

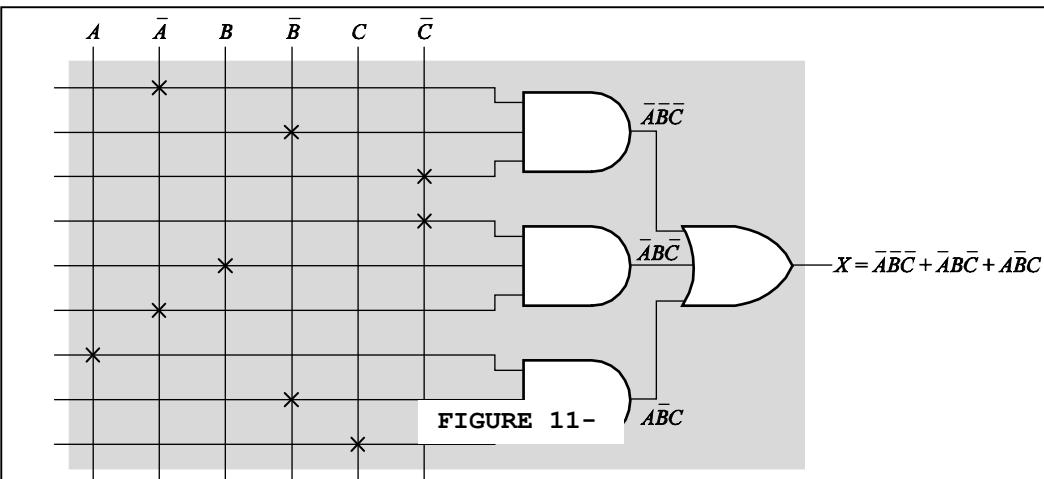
- 34.** 1. Add an additional row of four flip-flops.  
 2. Connect the keypad encoder outputs to the added flip-flops.  
 3. Change the 2-bit counter to a 3-bit counter.  
 4. Replace the four 2-input AND gates in the memory address decoder to 3-input AND gates and add a fifth 3-input AND gate. Modify the decoder to decode 000, 001, 010, 011, 100.  
 5. Change the 4-input OR gates to 5-input OR gates.
- 35.** To accommodate a 5-bit entry code, shift register C must be loaded with five 0s instead of four. The HIGH (1) must be moved left place on the parallel inputs.

## **CHAPTER 11**

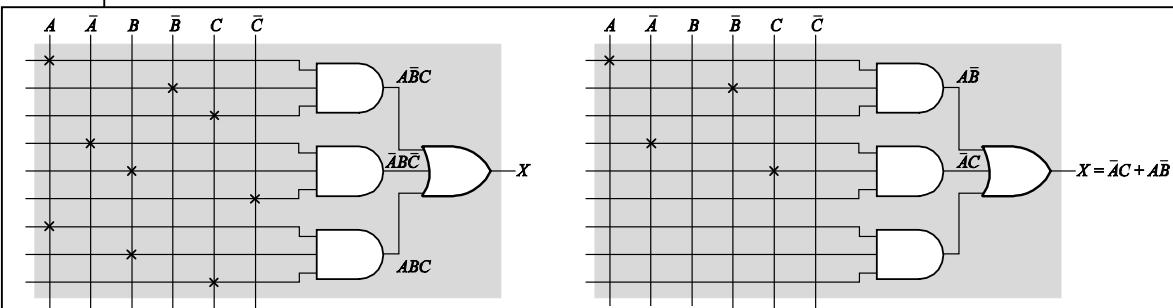
### PROGRAMMABLE LOGIC AND SOFTWARE

#### **Section 11-1 Programmable Logic: SPLDs and CPLDs**

- 1.**  $X = \overline{ABC} + \overline{ABC} + \overline{ABC}$ . See Figure 11-1.



**2.**



**FIGURE 11-** implemented as stated because it has 4 product terms.  
 Simplify to:  $X = \overline{AC} + A\bar{B}$

- (b) PAL12H6 is a programmable array logic device with 12 inputs and 6 active-HIGH outputs.

4. Typically, an exclusive-OR gate is used to determine the polarity of the output. When a 1 is applied to one input of the XOR gate, the output of the XOR is the complement of the signal on the other input. When a 0 is applied to one input of the XOR, the signal on the output of the XOR is the same as the signal on the other input.
5. A CPLD basically consists of multiple SPLDs that can be connected with a programmable interconnect array.

### **Section 11-2 Altera CPLDs**

6. (a) Inputs from PIA to LAB: **36** (b) Outputs from LAB to PIA: **16** (c) Inputs from I/O to PIA: **8 to 16** (d) Outputs from LAB to I/O: **8 to 16**
7. (a)  $\overline{ABCD}$  (b)  $ABC(\overline{DE}) = ABC(\overline{D} + \overline{E}) = ABC\overline{D} + ABC\overline{E}$
8.  $A\overline{BCD} + EFGH + AB\overline{CD} + \overline{ABCD}$

### **Section 11-3 Xilinx CPLDs**

9.  $A\overline{B} + \overline{AB}$
10. (a) Inputs from AIM to FB: **40** (b) Outputs from FB to AIM: **16** (c) Inputs from I/O to AIM: **16** (d) Outputs from FB to I/O: **16**
11.  $X_1 = A\overline{BCD} + \overline{ABCD} + ABC\overline{D}; \quad X_2 = ABCD + A\overline{BCD} + \overline{ABCD} + A\overline{BCD}$

### **Section 11-4 Macrocells**

12. (a) A 0 on the select line selects  $D_0$ . The output is **1**.  
 (b) A 1 on the select line selects  $D_1$ . The output is **0**.
13. (a) Since the  $D_0$  (upper) input of MUX 5 is selected, the macrocell is configured for **combinational** logic. The output of the XOR goes through MUX 5 to the "To I/O" output making it a **1**.  
 (b) Since the  $D_1$  (lower) input of MUX 5 is selected, the macrocell is configured for **registered** logic. The output of the flip-flop goes through MUX 5 to the "To I/O" output making it a **0**.
14. (a) The macrocell is configured for **registered** logic because the  $D_1$  input of MUX 8 is selected, allowing the flip-flop output to pass through.  
 (b) The **GCK1** clock is applied to the flip-flop because the  $D_1$  input of MUX 3 and the  $D_1$  input of MUX 5 are selected.  
 (c) The OR gate output is applied to the XOR which is set for noninversion by MUX 1. The output of the XOR is selected by MUX2 and a **1** is applied to the D/T input of the flip-flop.  
 (d) The output of MUX 8 is a **1** because MUX 8 selects the Q output of the flip-flop (assuming that the S and R inputs are 0).
15. (a) The macrocell is configured for **registered** logic because the  $D_1$

input of MUX 8 is selected, allowing the flip-flop output to pass through.

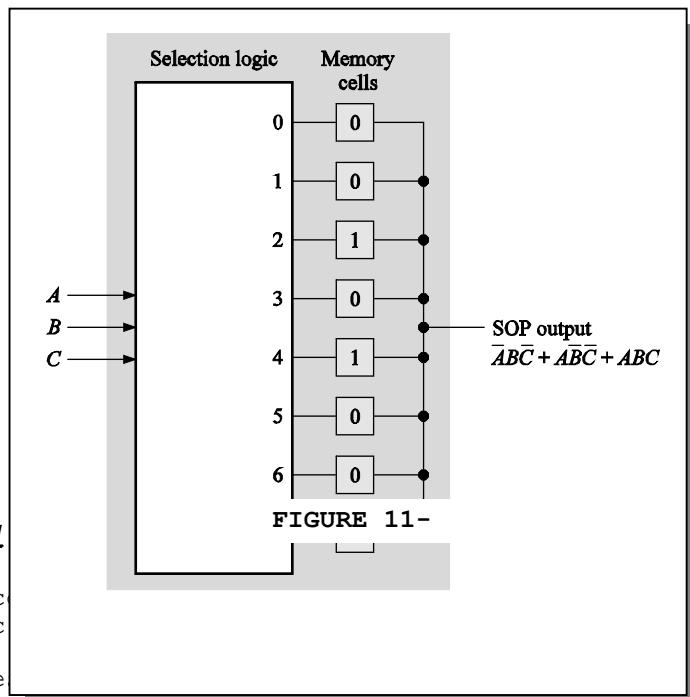
- (b) The **GCK1** clock is applied to the flip-flop because the  $D_1$  input of MUX 3 and the  $D_1$  input of MUX 5 are selected.
- (c) The OR gate output is applied to the XOR which is set for inversion by MUX 1. The output of the XOR is selected by MUX 2 and a **0** is applied to the D/T input of the flip-flop.
- (d) The output of MUX 8 is a **0** because MUX 8 selects the Q output of the flip-flop (assuming that the S and R inputs are 0).

## **Section 11-5 Programmable Logic: FPGAs**

**16.** An FPGA typically consists of configurable logic blocks (CLBs). Each CLB is made up of a number of logic modules with a local interconnect. Each logic module typically consists of a look-up table (LUT) and associated logic. Global column and row interconnects are used to connect the CLBs to I/Os as well as each other.

**17.** SOP output =  $\overline{ABC} + \overline{ABC} + \overline{ABC} + A\overline{BC} + ABC$

**18.** See Figure 11-3.



## **Section 11**

**19.** An ALM consists of a LUT, adder logic, and register logic.

**20.** The mode select switch can be used to select Arithmetic, ANDed LUT,

**21.** See Figure 11-4.

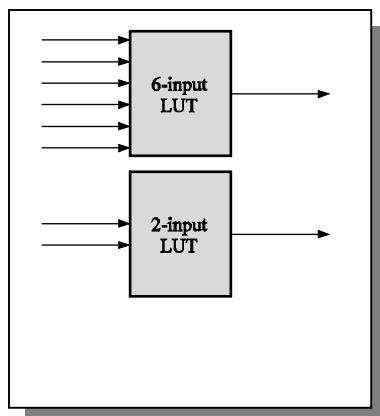
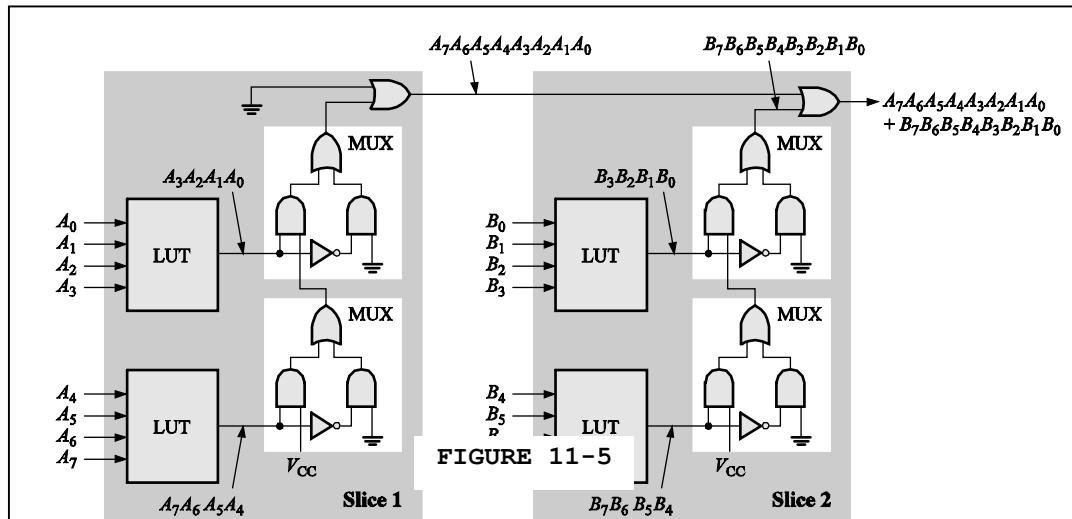


FIGURE 11-

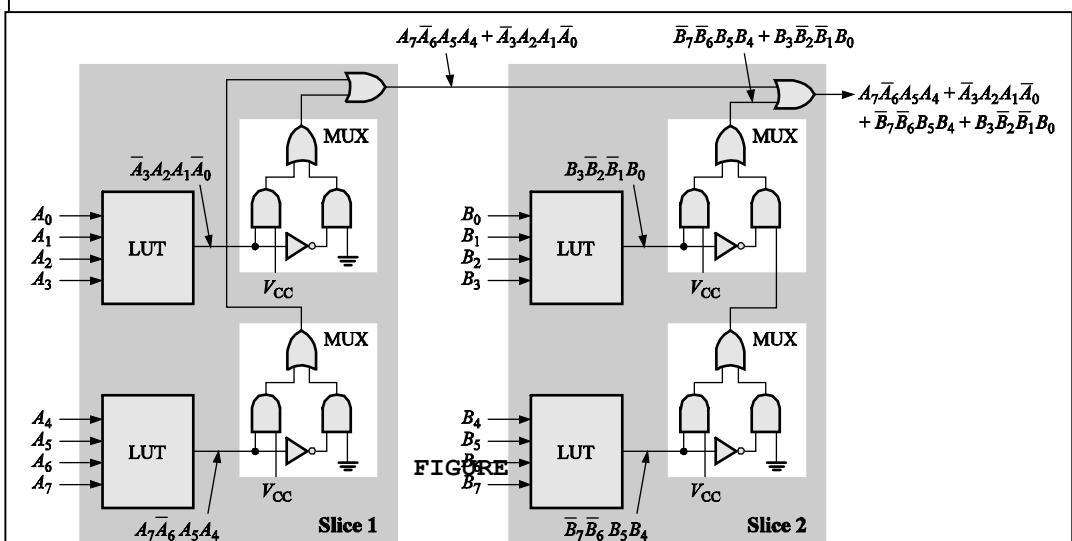
$$\begin{aligned}
 22. \quad & (A_4 A_3 \bar{A}_2 A_1 + \bar{A}_4 \bar{A}_3 \bar{A}_2 A_1) A_0 + (\bar{A}_5 A_3 A_2 A_1 + A_5 \bar{A}_3 A_2 \bar{A}_1 + A_5 A_3 A_2 \bar{A}_1) A_0 \\
 = & A_4 A_3 \bar{A}_2 A_1 A_0 + \bar{A}_4 \bar{A}_3 \bar{A}_2 A_1 A_0 + \bar{A}_5 A_3 A_2 A_1 \bar{A}_0 + A_5 \bar{A}_3 A_2 \bar{A}_1 \bar{A}_0 + A_5 A_3 A_2 \bar{A}_1 \bar{A}_0
 \end{aligned}$$

**Section 11-7 Xilinx FPGAs**

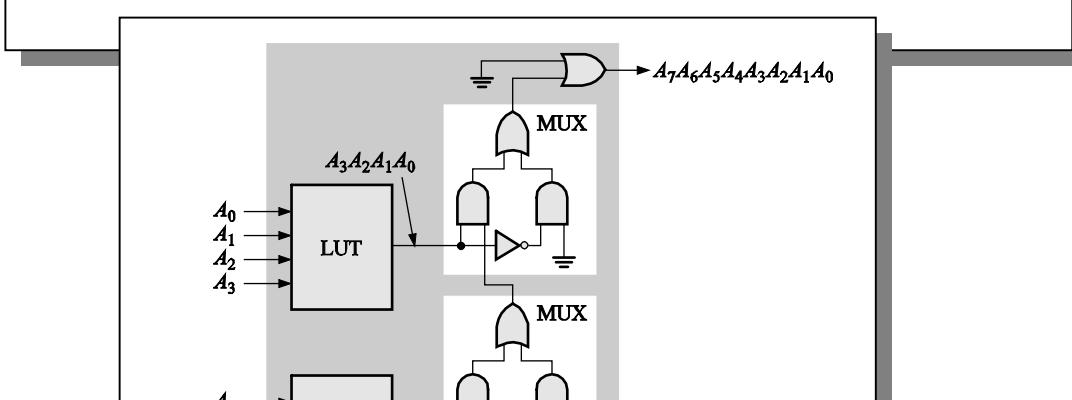
23. See Figure 11-5.



24 .

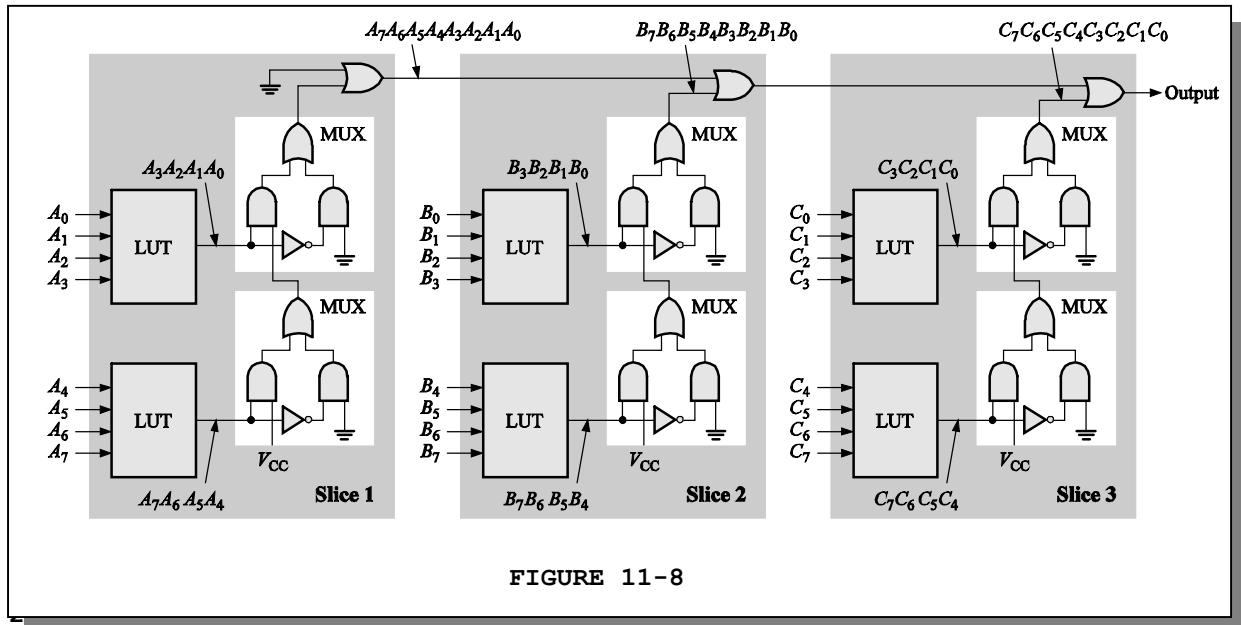


25 .

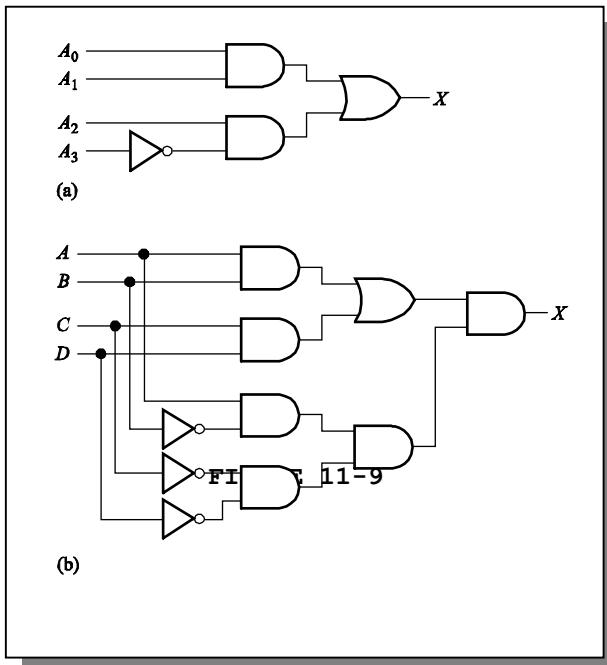


**FIGURE 11-7**

**26.** Three slices are required. See Figure 11-8.

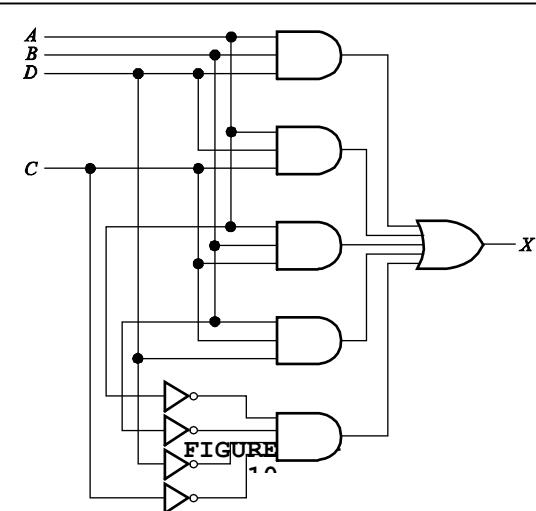


**FIGURE 11-8**

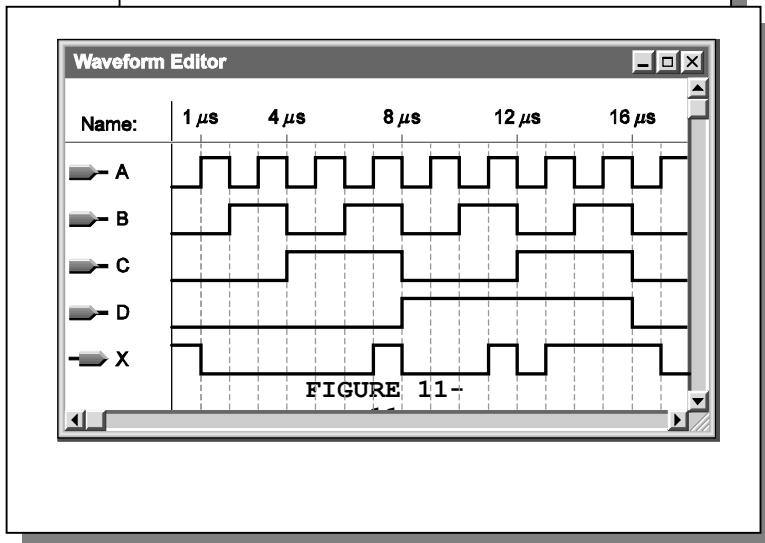


$$\begin{aligned}
 28. \quad X &= \overline{ABCD} + A\overline{BCD} + A\overline{B}CD + ABC\overline{D} + ABCD + \overline{ABC}\overline{D} \\
 &= ABD + ACD + ABC + BCD + \overline{ABC}\overline{D}
 \end{aligned}$$

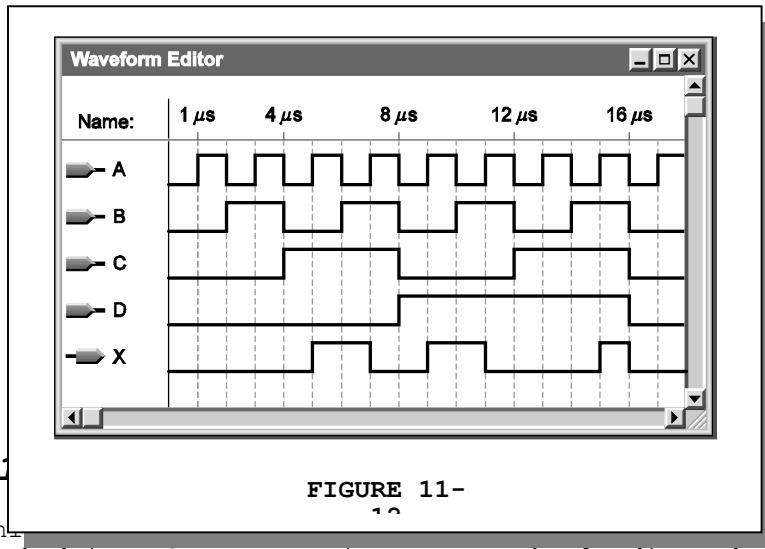
See Figure 11-10.



29. See Figure 11



30.  $X = \overline{ABCD} + A\overline{BCD} + ABCD + A\overline{BCD} + \overline{ABC}\overline{D}$ . See Figure 11-12.



### Section 1

**FIGURE 11-**

31. The Shifted data from the output of the internal programmable logic pass through the MUX, and are clocked into Capture register A on the leading edge of the clock pulse. From the output of Capture register A, the data go through the upper MUX and are clocked into Capture register B on the trailing edge of the clock pulse.
32. PDI/O = 0 and OE = 1. The data from the internal programmable logic pass through the selected MUX and through the output buffer to the pin.
33. PDI/O = 0 and OE = 0. The data are applied to the input pin and go through the selected MUX to the internal programmable logic.
34. SHIFT = 1, PDI/O = 1, and OE = 0. Data are applied to SDI, go through the MUX, and are clocked into Capture register A on the leading edge of the clock pulse. From the output of Capture register A, the data go through the upper MUX and are clocked into Capture register B on the trailing edge of the clock pulse. A pulse on the UPDATE input clocks the data into Update register B. The data on the output of Capture Register B go through the MUX to the internal programmable logic. The data also appear on the SDO.

### **Section 11-10 Troubleshooting**

35. 00001100101000111011 shifted from TDI to TDO, left-most bit first. The bold-faced code will appear on the logic inputs in the sequence shown.

```

0 00001100101000111011
1 00001100101000111011
3 00001100101000111011
6 000001100101000111011
12    000011000101000111011
9 00001100101000111011
2 000011001001000111011
5 00001100101000111011
10    00001100101000111011
4 00001100101000111011
8 00001100101000111011
1 000011001010001111011
3 000011001010001111011
7 000011001010000111011
15    00001100101000111011
14    00001100101000111011
13    00001100101000111011
11    00001100101000111011

```

### **Digital System Application**

36. 11 inverters can be eliminated. Only four are needed to produce the complements of *A*, *B*, *C*, and *D*.

There are three AND gates that produce the product term  $\overline{AC}$ . Two can be eliminated.

There are three AND gates that produce the product term  $\overline{AB}$ . Two can be eliminated.

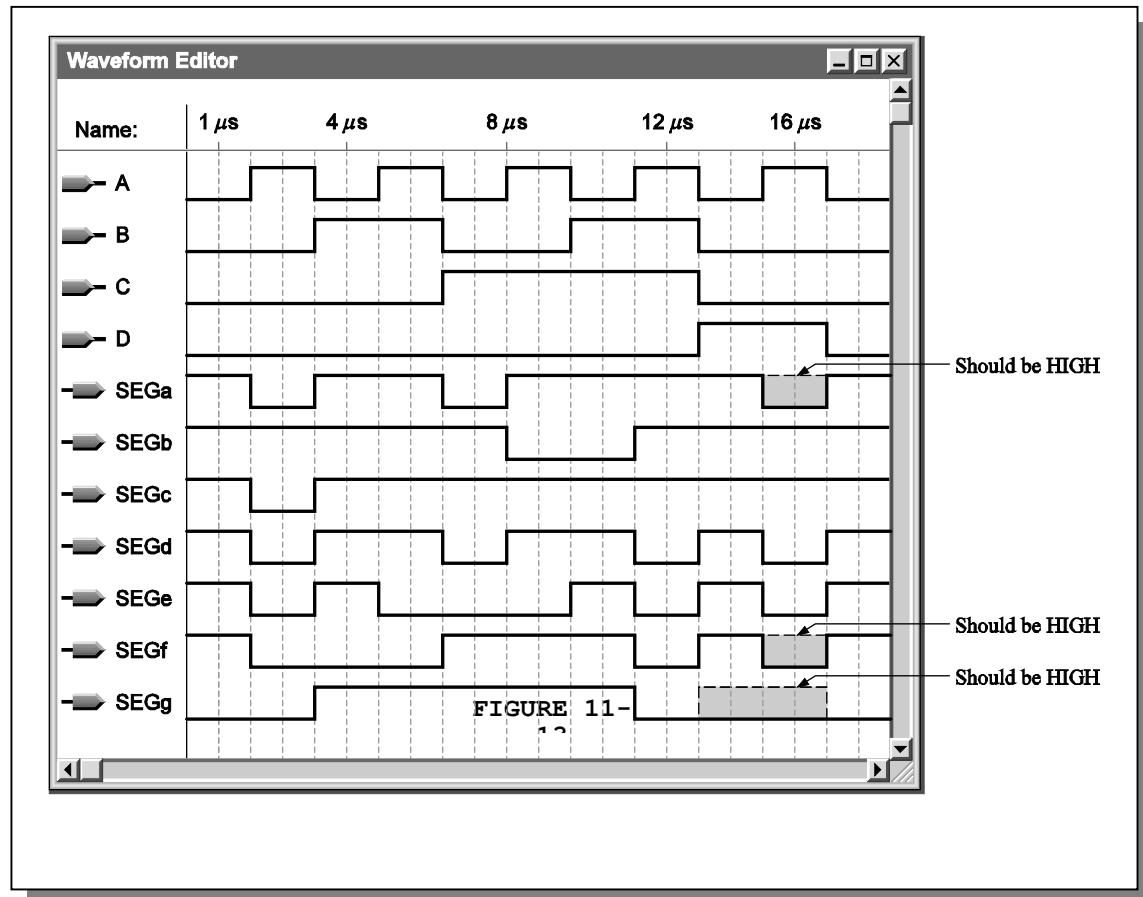
There are two AND gates that produce the product term  $B\overline{C}$ . One can be eliminated.

There are two AND gates that produce the product term  $\overline{BC}$ . One can be eliminated.

There are two AND gates that produce the product term  $\overline{AB}$ . One can be eliminated.

7 AND gates can be eliminated.

37. The D input to the logic is faulty or not connected. See Figure 11-13.



---

## **CHAPTER 12**

### **INTRODUCTION TO COMPUTERS**

---

#### ***Section 12-1 The Basic Computer***

1. The basic elements of a computer are central processing unit (CPU), memory unit, and input/output ports.
2. Two types of software are system and application.
3. A bus is a set of physical connections over which data and other information is transferred in a computer according to a standard set of specifications.
4. A port is a physical interface on a computer through which data is passed to and from peripherals.

#### ***Section 12-2 Microprocessors***

5. The basic elements of a microprocessor are arithmetic logic unit (ALU), instruction decoder, control unit, and register array.
6. A microprocessor performs arithmetic operations, logic operations, data movements, and decision functions.
7. The three microprocessor buses are address, data, and control.
8. Groups of Pentium instructions are: data transfer, arithmetic and logic, bit manipulation, loops and jumps, strings, subroutines and interrupts, and control.

#### ***Section 12-3 A Specific Microprocessor Family***

9. A microprocessor repeatedly cycles through *fetch, decode, execute*.
10. Pipelining is the process of fetching and executing at the same time so that more than one instruction can be processed simultaneously.
11. The six segment registers of the 80386 and above are:  
CS, DS, SS, ES, FS, GS
12. The code segment (CS) register contains 0F05 and the instruction pointer contains 0100. The physical address is

$$0F050 + 0100 = 0F150$$

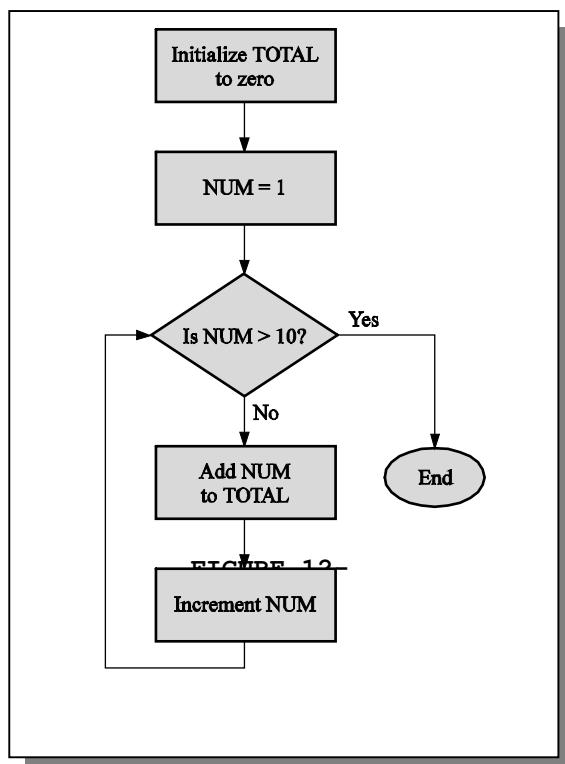
13. AH and AL are 8-bit registers and represent the high and low part of the 16-bit AX register. The EAX is a 32-bit register which includes the AX register as the lower 16 bits.

- 14.** (a) A *flag* is a bit stored in the flag register that is set or cleared by the processor.  
(b) A *flag* indicates a status or a control condition. A *status* flag is an indicator of a condition after an arithmetic or logic operation. A *control* flag alters processor operations under certain conditions.

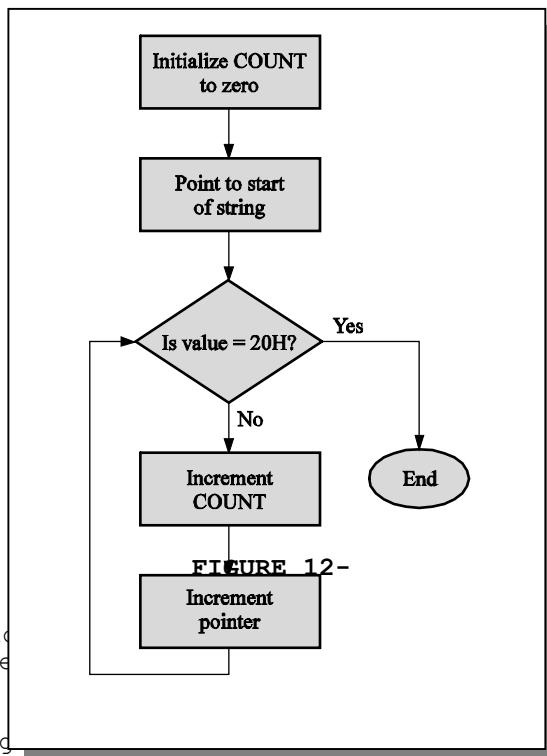
- 15.** Instruction pairing allows two instructions to execute at the same time.

#### **Section 12-4 Computer Programming**

- 16.** An assembler is a program that translates mnemonics and operands into machine code.
- 17.** The flowchart in Figure 12-1 shows the process for adding numbers from one to ten and saving the results in a memory location named TOTAL.



18. The flowchart in Figure 12-2 shows how you can count the number of bytes in a string and place the count in a memory location called COUNT. The string starts at a location named START and uses 20H (space) to indicate the end.



19. When the instruction pointed to by the

word in memory register.

20. translates a program to machine code.

hat compiles or and converts it to

### **Section 12-5 Interrupts**

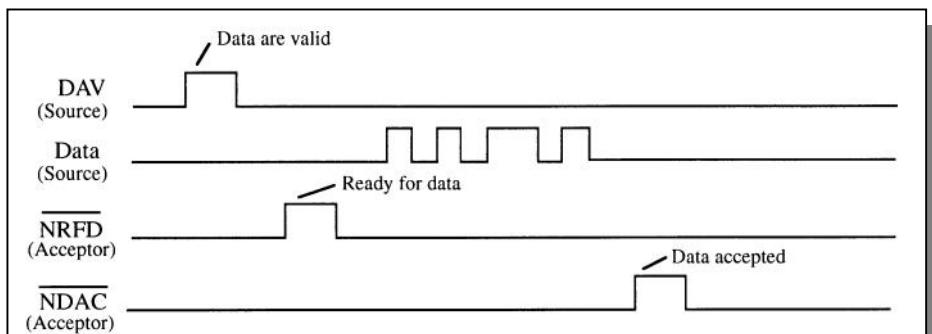
21. In a polled I/O, the CPU polls each device in turn to see if it needs service; in an interrupt-driven system, the peripheral device signals the CPU when it requires service.
22. Vectoring is when the PIC provides a pointer to a service routine.
23. A software interrupt is a program instruction that invokes an interrupt service routine.

### **Section 12-6 Direct Memory Access (DMA)**

24. In a DMA operation, the DMA controller is given control by the CPU and allows data to flow between memory and a peripheral directly, bypassing the CPU.
25. The CPU is bypassed in DMA.

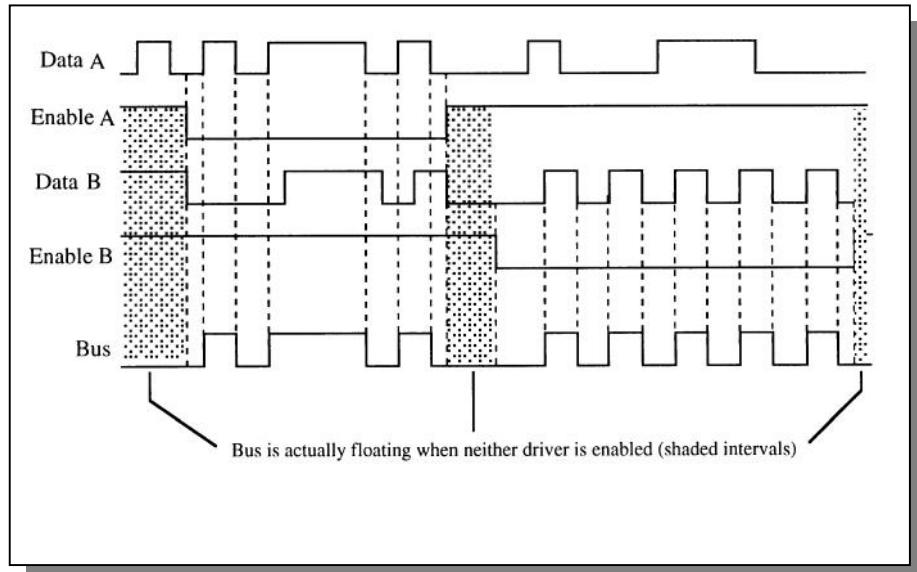
### **Section 12-7 Internal Interfacing**

26. See Figure 12-3.

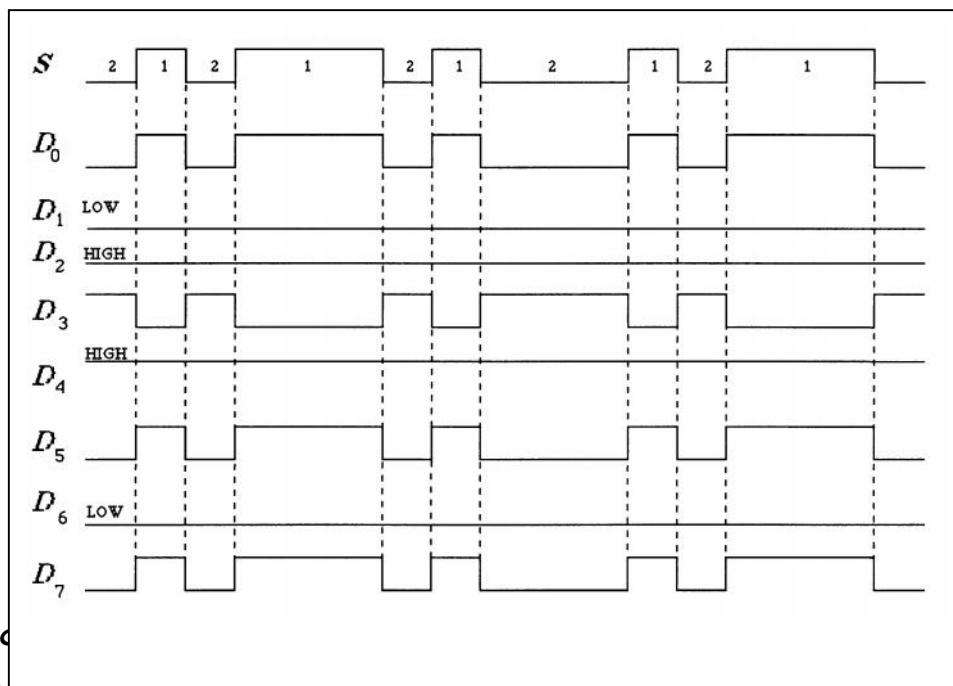


**FIGURE 12-**

27. See Figure 12-4.

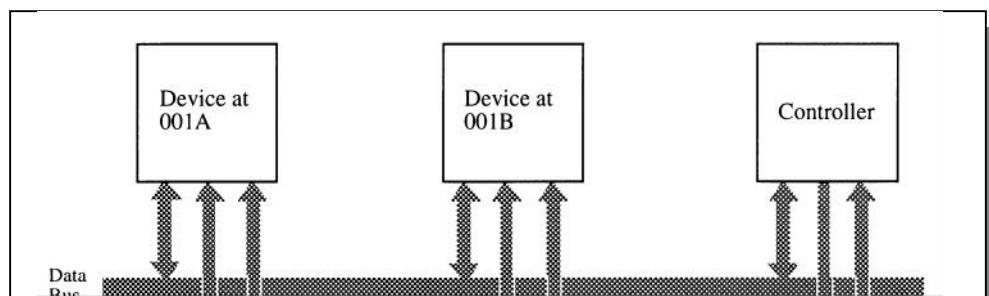


28. See Figure 12-5.



### Section 12-5

29. The **PCI** bus is a local bus that connects expansion cards to the processor. The PCI bus is used for expansion devices and is connected to the local bus through a bus controller.
30. Plug-and-Play refers to self-configuring hardware that can be installed into and used in a computer system without the need for manual installation of jumpers or setting of switches.
31. The PCI bus is a 33 or 66 MHz, 32- or 64-bit, plug-and-play compatible expansion bus. ISA is an 8- or 16-bit 8.33 MHz expansion bus. PCI supports 3.3 V supplies while ISA supports 5 V and 12 V supplies.
32. A shorter RS-232C cable can support faster communication rates.
33. DCE stands for data communications equipment, such as a modem. DTE stands for data terminal equipment, such as a computer. Both acronyms are associated with the RS-232/EIA-232 standard.
34. A USB cable consists of a power line, ground line, and two differential data lines.
35. Since there are eight instruments already on the bus and the limit is fourteen, six more instruments can be connected.
36. Three data bytes are transferred because the NDAC line goes HIGH three times, each time indicating that a data byte is accepted.
37. A controller is sending data to two listeners. The first two bytes of data (3F and 41) go to the listener with address 001A. The second two bytes go to the listener with address 001B. The handshake signals (DAV, NRFD, and NDAC) indicate that the data transfer is successful. See Figure 12-6.



**FIGURE 12-**

- 38.** If a talker sends a data byte to a listener on a GPIB system and a DTE sends a data byte to a DCE on an RS-232C system, the RS-232C system will receive the data first. This is because GPIB requires significantly more setup and handshaking than RS-232C.

---

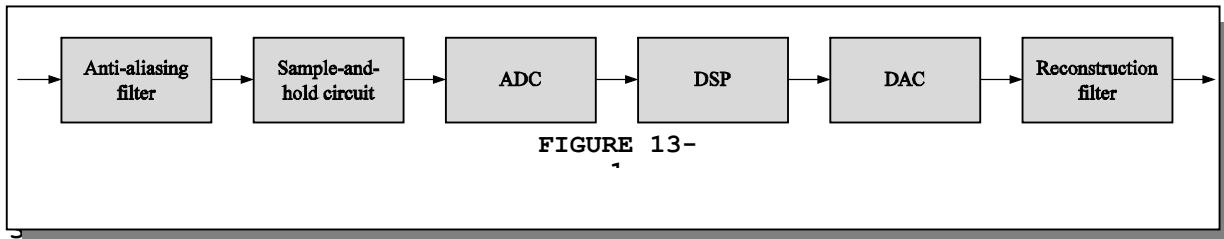
## CHAPTER 13

### INTRODUCTION TO DIGITAL SIGNAL PROCESSING

---

#### **Section 13-1 Digital Signal Processing Basics**

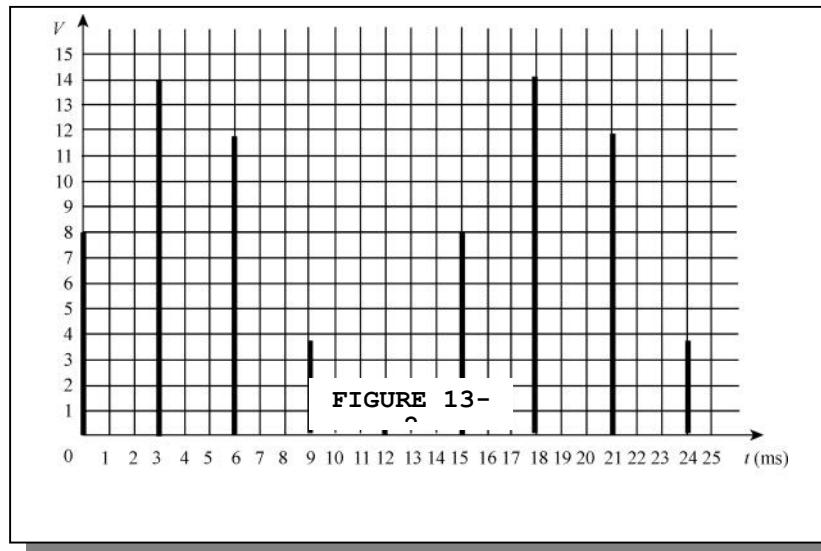
1. The purpose of analog-to-digital conversion is to change an analog signal into a sequence of digital codes that represent the amplitude of the analog signal with respect to time.
2. See Figure 13-1.



of digital codes into an analog signal represented by the digital codes.

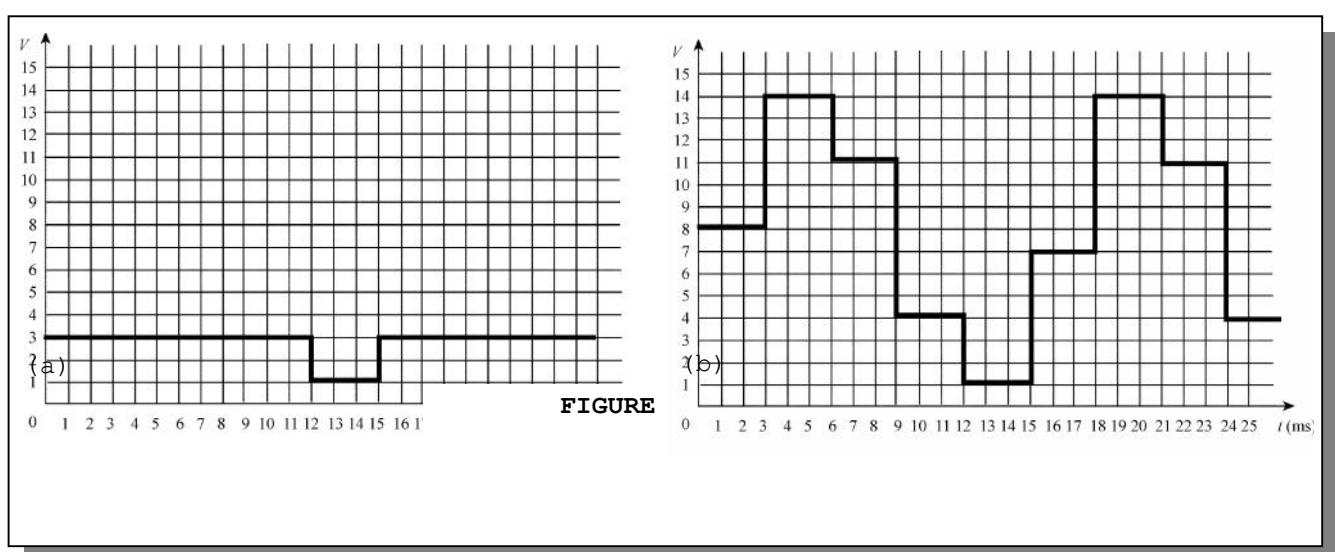
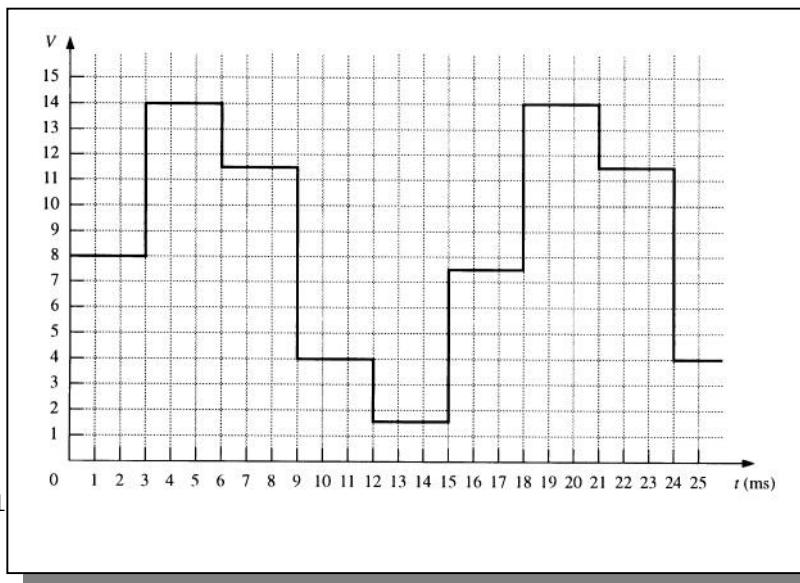
#### **Section 13-2 Converting Analog Signals to Digital**

4. See Figure 13-2.

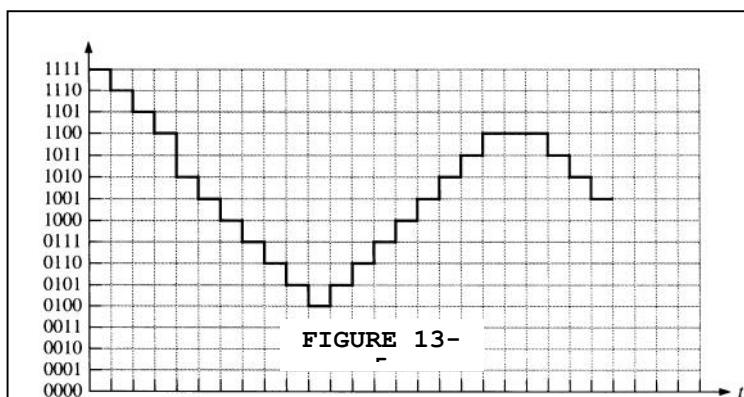


5. See Figure 13-3.

6. 11, 11  
 7. 1000,  
 8. See Figure 13-4.



9. See Figure 13-5.



### Section 13-3

ods

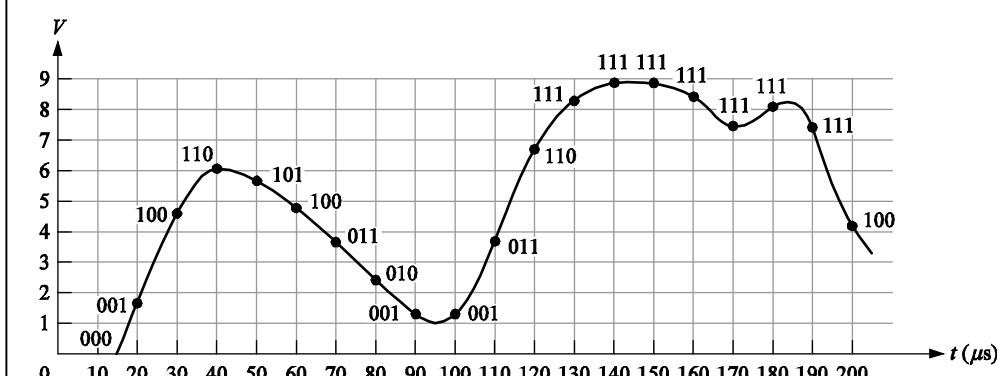
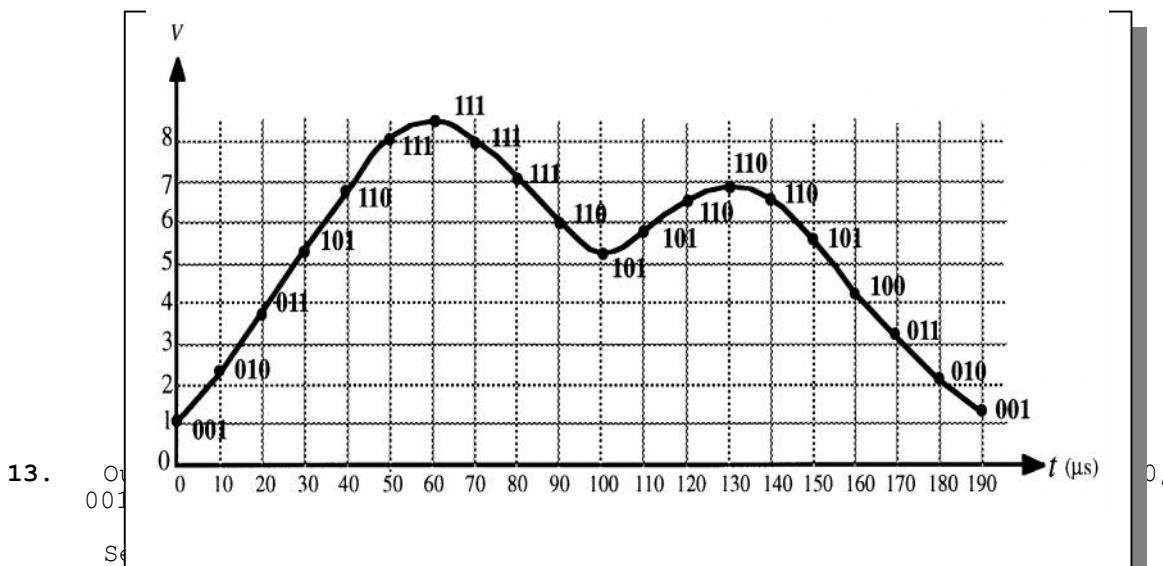
10.  $\frac{V_{\text{out}}}{V_{\text{in}}} = \frac{2 \text{ V}}{10 \text{ mV}} = 200$

11. 
$$\frac{V_{\text{OUT}}}{V_{\text{IN}}} = \frac{R_F}{R_{\text{IN}}}$$

$$R_F = R_{\text{in}} \left( \frac{V_{\text{OUT}}}{V_{\text{IN}}} \right) = 1 \text{ k}\Omega (330) = 330 \text{ k}\Omega$$

12. 001, 010, 011, 101, 110, 111, 111, 111, 111, 110, 101, 101, 110, 110, 101, 100, 011, 010, 001.

See Figure 13-6.



**FIGURE 13-**

**14.**

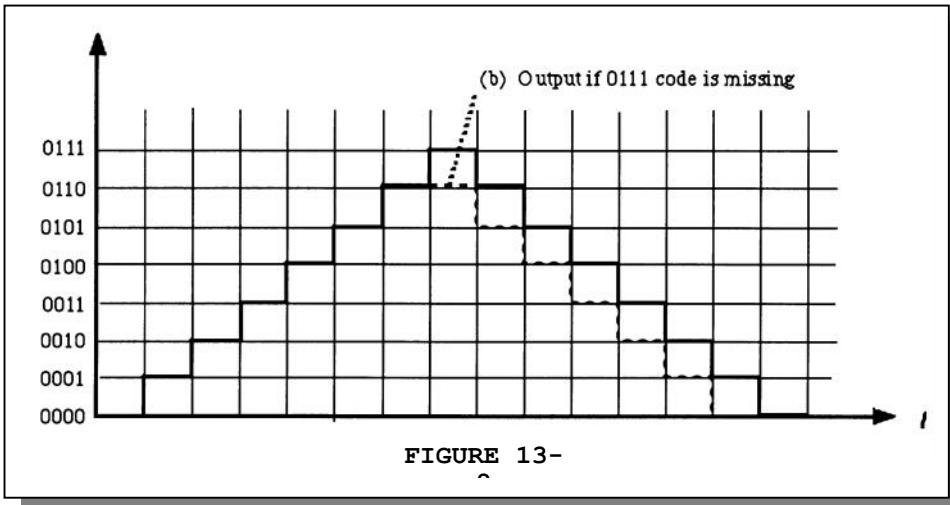
| SAR | Comment                          |
|-----|----------------------------------|
| 11  | Less than $V_{in}$ . Keep the 1. |
| 11  | Less than $V_{in}$ . Keep the 1. |
| 11  | Less than $V_{in}$ . Keep the 1. |
|     | Less than $V_{in}$ . Keep the 1. |

Conversion never terminates since 2 bits cannot represent the input.

**15.**

| SAR  | Comment                                                                             |
|------|-------------------------------------------------------------------------------------|
| 1000 | Greater than $V_{in}$ . Reset MSB.                                                  |
| 0100 |                                                                                     |
| 0110 | Less than $V_{in}$ . Keep the 1.<br>Equal to $V_{in}$ . Keep the 1<br>(final state) |

**16.** See Figure 13-8.



**Section 13-4 The Digital Signal Processor (DSP)**

17.  $2000 \text{ MIPS} \times \frac{32 \text{ bit/instruction}}{8 \text{ bits/byte}}$

$$= 2000 \text{ MIPS} \times 4 \text{ bytes/instruction}$$

$$= 8000 \text{ Mbytes/s}$$

18.  $\frac{400 \text{ Mbits/s}}{32 \text{ bits/instruction}} = 12.5 \text{ million instructions/s}$

19.  $1000 \text{ MFLOPS} = 1,000,000,000 \text{ floating-point operations/s}$

- 20.
1. Program address generate (PG). The program address is generated by the CPU.
  2. Program address send (PS). The program address is sent to the memory.
  3. Program access ready wait (PW). A memory read operation occurs.
  4. Program fetch packet receive (PR). The CPU receives the packet of instructions.

- 21.
1. Instruction dispatch (DP): Instruction packets are split into execute packets and assigned to functional units;
  2. Instruction decode (DC): Instructions are decoded.

**Section 13-5 Digital-to-Analog Conversion Methods**

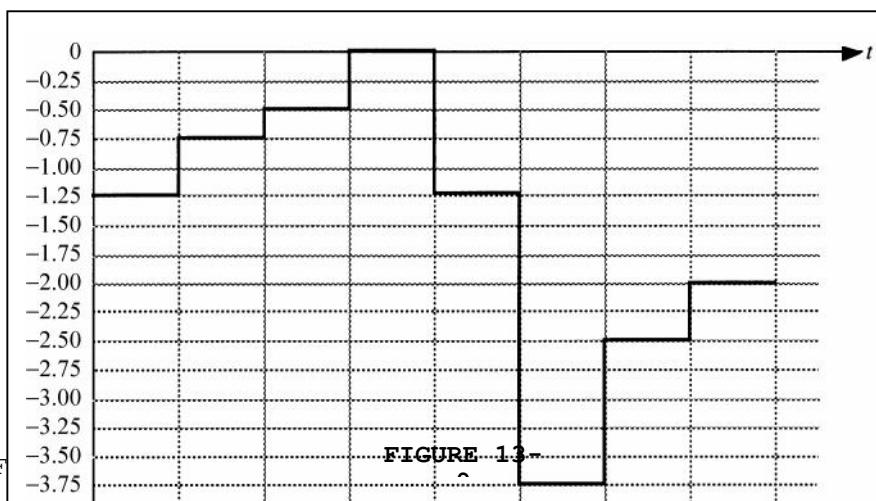
22.  $R_0 = 10 \text{ k}\Omega$

$$R_1 = \frac{R_0}{2} = \frac{10 \text{ k}\Omega}{2} = 5 \text{ k}\Omega$$

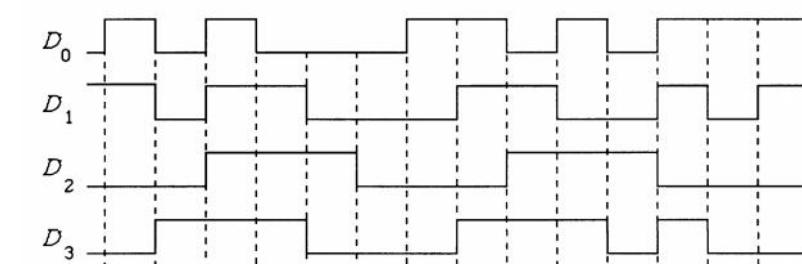
$$R_2 = \frac{R_0}{4} = \frac{10 \text{ k}\Omega}{4} = 2.5 \text{ k}\Omega$$

$$R_3 = \frac{R_0}{8} = \frac{10 \text{ k}\Omega}{8} = 1.25 \text{ k}\Omega$$

23. See Figure 13-9.



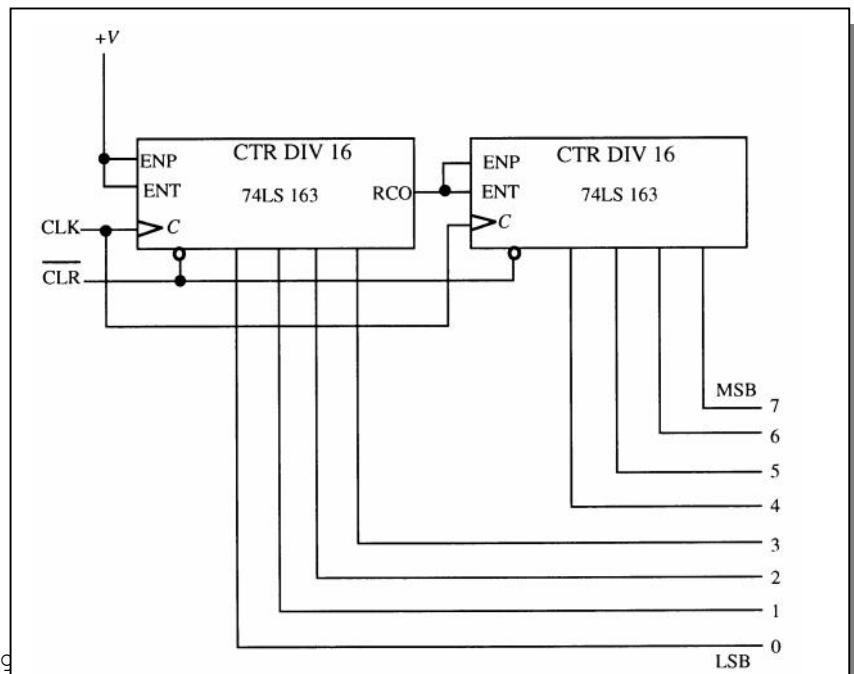
- 24.



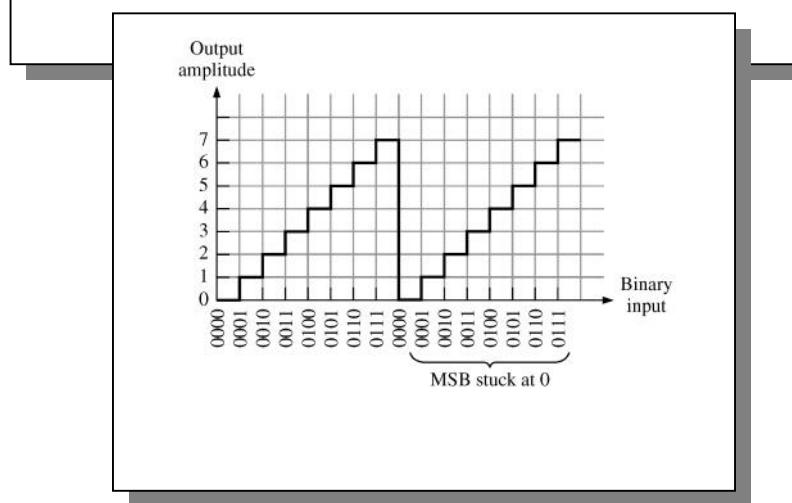
**FIGURE 13-**  
**10**

25. (a)  $\left( \frac{1}{(2^3 - 1)} \right) 100 = 14.3\%$
- (b)  $\left( \frac{1}{2^{10} - 1} \right) 100 = 0.098\%$
- (c)  $\left( \frac{1}{2^{18} - 1} \right) 100 = 0.00038\%$

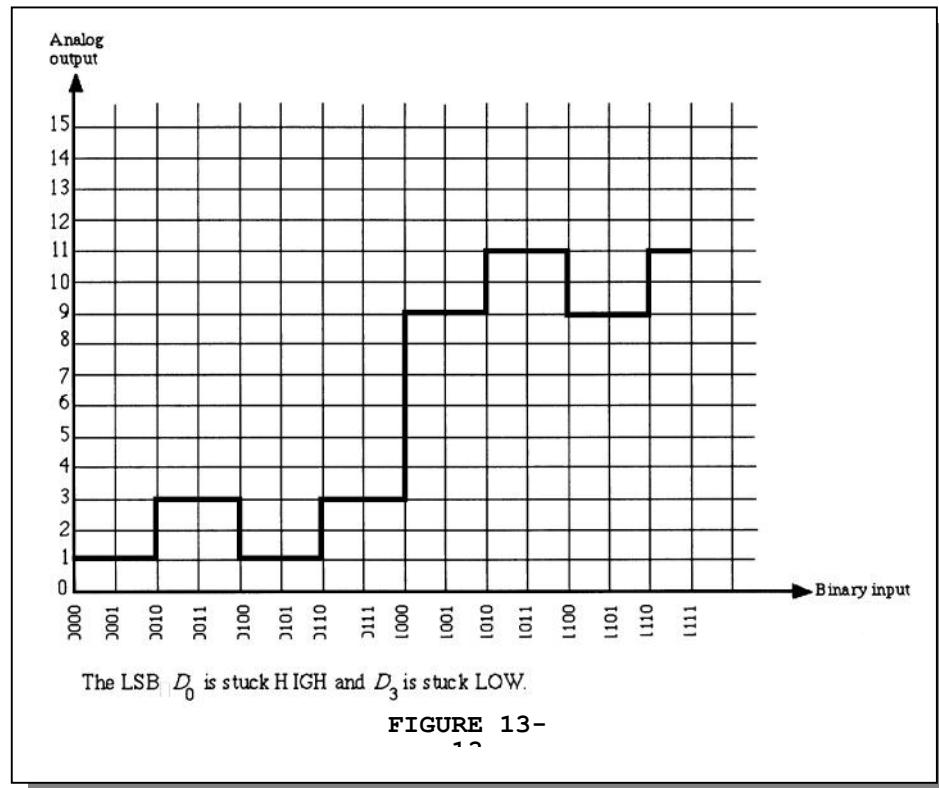
**26.** See Figure 13-11.



**27.** See Fig.



28. See Figure 13-13.



---

## CHAPTER 14

### INTEGRATED CIRCUIT TECHNOLOGIES

---

#### Section 14-1 Basic Operational Characteristics and Parameters

1. No, because the  $V_{OH(min)}$  is less than the  $V_{IH(min)}$ . The gate may interpret 2.2 V as a LOW.
2. Yes, they are compatible because the  $V_{OL(max)}$  is less than the  $V_{IL(max)}$ .
3.  $V_{NH} = V_{OH(min)} - V_{IH(min)} = 2.4 \text{ V} - 2.25 \text{ V} = 0.15 \text{ V}$   
 $V_{NL} = V_{IL(max)} - V_{OL(max)} = 0.65 \text{ V} - 0.4 \text{ V} = 0.25 \text{ V}$
4. The maximum amplitudes equal the noise margins of 0.15 V and 0.25 V.
5. Gate A:  
 $V_{NH} = 2.4 \text{ V} - 2 \text{ V} = 0.4 \text{ V}$   
 $V_{NL} = 0.8 \text{ V} - 0.4 \text{ V} = 0.4 \text{ V}$   
Gate B:  
 $V_{NH} = 3.5 \text{ V} - 2.5 \text{ V} = 1 \text{ V}$   
 $V_{NL} = 0.6 \text{ V} - 0.2 \text{ V} = 0.4 \text{ V}$   
Gate C:  
 $V_{NH} = 4.2 \text{ V} - 3.2 \text{ V} = 1 \text{ V}$   
 $V_{NL} = 0.8 \text{ V} - 0.2 \text{ V} = 0.6 \text{ V}$

**Gate C** has the highest noise margins.

6.  $P_{D(LOW)} = (5 \text{ V}) (2 \text{ mA}) = 10 \text{ mW}$   
 $P_{D(HIGH)} = (5 \text{ V}) (3.5 \text{ mA}) = 17.5 \text{ mW}$   
 $P_{D(avg)} = \frac{P_{D(LOW)} + P_{D(HIGH)}}{2} = \frac{27.5 \text{ mW}}{2} = 13.75 \text{ mW}$
7. The pulse goes through three gates in the shortest path.  
 $3 \times 4 \text{ ns} = 12 \text{ ns}$
8.  $t_{p(avg)} = \frac{t_{PLH} + t_{PHL}}{2} = \frac{2 \text{ ns} + 3 \text{ ns}}{2} = 2.5 \text{ ns}$
9. Gate A average propagation delay:

$$\frac{t_{PLH} + t_{PHL}}{2} = \frac{1 \text{ ns} + 1.2 \text{ ns}}{2} = 1.1 \text{ ns}$$

Speed/Power product =  $(1.1 \text{ ns}) (15 \text{ mW}) = 16.5 \text{ pJ}$

Gate B average propagation delay:

$$\frac{5 \text{ ns} + 4 \text{ ns}}{2} = 4.5 \text{ ns}$$

Speed/Power product =  $(4.5 \text{ ns}) (8 \text{ mW}) = 36 \text{ pJ}$

Gate C average propagation delay:  $\frac{10 \text{ ns} + 10 \text{ ns}}{2} = 10 \text{ ns}$

Speed/Power product =  $(10 \text{ ns}) (0.5 \text{ mW}) = 5 \text{ pJ}$

**Gate C** has the best speed/power product.

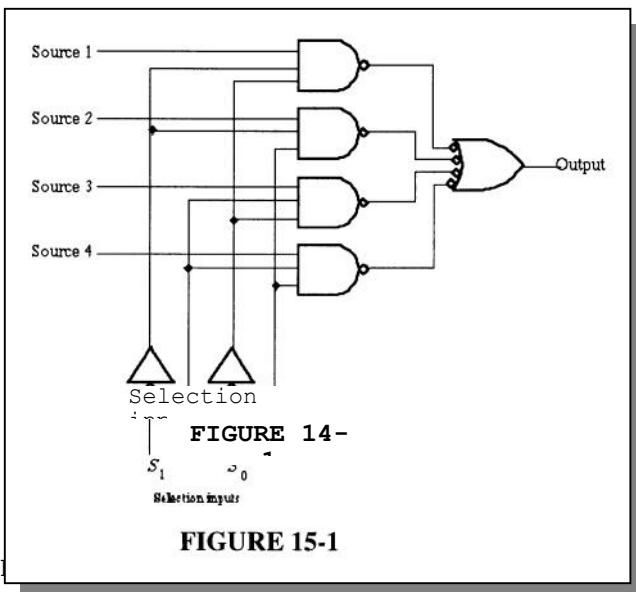
10. Gate A can be operated at the highest frequency because it has the shortest propagation delay.
  11. G2 is overloaded because it has 12 unit loads.
  12. The network in (a) can operate at the highest frequency because the driving gate has fewer loads.

## *Section 14-2 CMOS Circuits*

- 13.** (a) ON                      (b) OFF  
      (c) OFF                      (d) ON

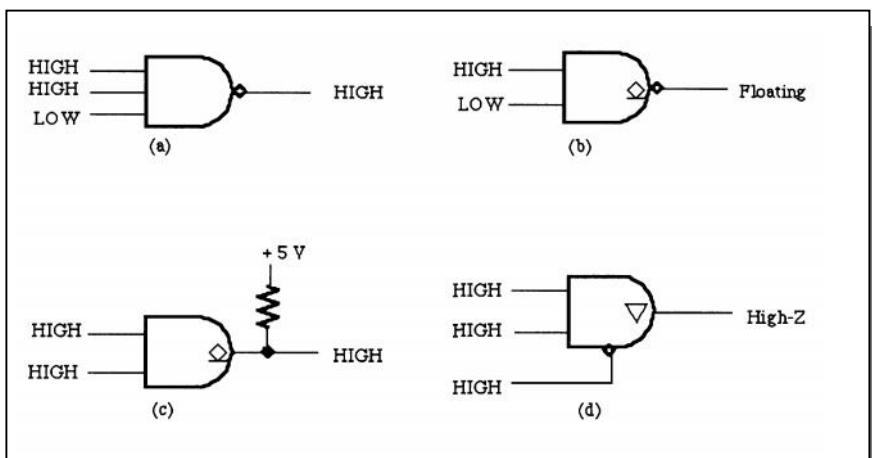
**14.** Unused inputs should be connected as follows:  
      Negative-OR gate (NAND) to  $V_{cc}$   
      NAND gate to +  $V_{cc}$   
      NOR gate to ground

**15.** See Figure 14-1 for another possible approach in addition to circuit given in text answers.



**FIGURE 15-1**

- 16.** (a) ON:  base-emitter junction.  
(b) OFF: insufficient voltage on base to forward-bias the base-emitter junction.  
(c) OFF: emitter is more positive than the base which reverse-biases the base-emitter junction.  
(d) OFF: base and emitter at same voltage. No forward bias.  
**17.** See Figure 14-2.

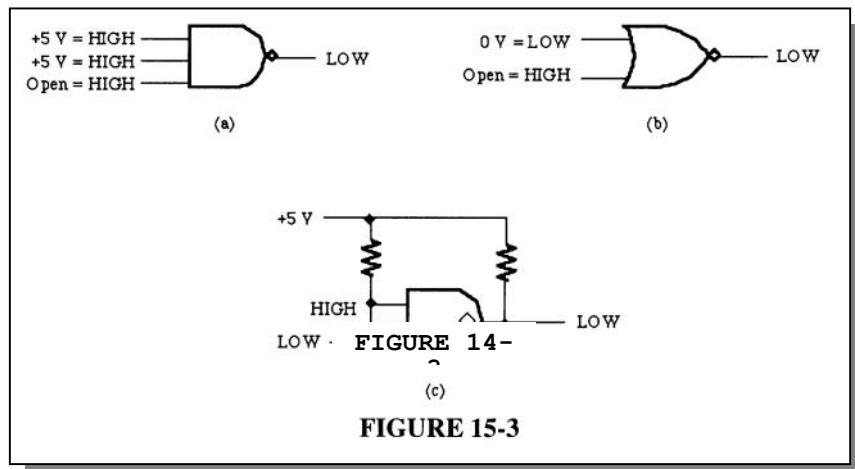


**FIGURE 14-**

- 18.** Connect a  $1\text{ k}\Omega$  pull-up resistor to the unused inputs of the two NAND gates. Connect the unused input of the NOR gate to ground. Connect a pull-up resistor to the open collector of the NOR gate (value depends on load).

#### **Section 14-4 Practical Considerations in the Use of TTL**

- 19.** See Figure 14-3.



- 20.** (a) The driving gate output is HIGH, it is sourcing 3 unit loads.  
 $I_T = 3(40\text{ }\mu\text{A}) = 120\text{ }\mu\text{A}$

- (b) The driving gate output is LOW, it is sinking current from 2 unit loads.

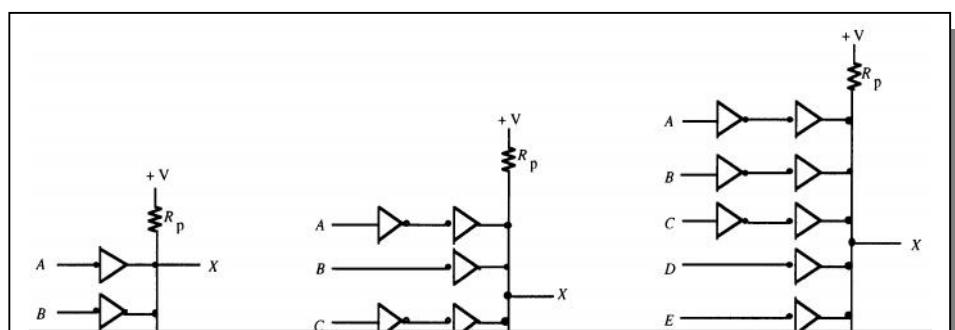
$$I_T = 2(-1.6\text{ mA}) = -3.2\text{ mA}$$

- (c) G1 output is HIGH, it is sourcing 6 unit loads.  
 $I_T = 6(40\text{ }\mu\text{A}) = 240\text{ }\mu\text{A}$

G2 output is LOW, it is sinking current from 2 unit loads.  
 $I_T = 2(-1.6\text{ mA}) = -3.2\text{ mA}$

G3 output is HIGH, it is sourcing 2 unit loads.  
 $I_T = 2(40\text{ }\mu\text{A}) = 80\text{ }\mu\text{A}$

- 21.** See Figure 14-4. Pull-up resistors of second-level inverters are not shown.



**FIGURE 14-4**

22. (a)  $X = ABC\bar{D}$

(b)  $X = (\overline{ABC})(\overline{DE})(\overline{FG})$

(c)  $X = (\overline{A+B})(\overline{C+D})(\overline{E+F})(\overline{G+H}) = \overline{ABCDEFGH}$

23. Worst case for determining minimum  $R_p$  is when only one gate is sinking all of the current (40 mA maximum).

For 10 UL:  $I_L = 10(1.6 \text{ mA}) = 16 \text{ mA}$

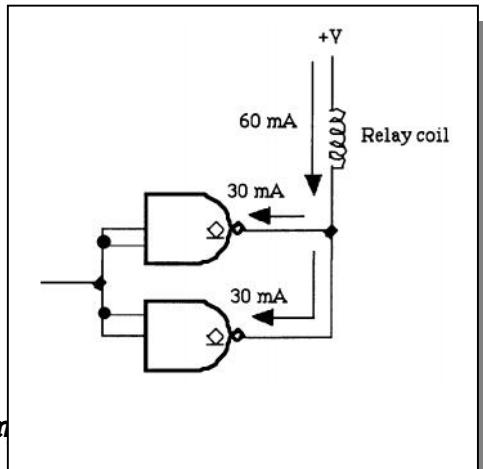
For each gate:  $I_{Rp(max)} = I_{OL(max)} - 16 \text{ mA} = 40 \text{ mA} - 16 \text{ mA} = 24 \text{ mA}$

$$V_{Rp} = 5 \text{ V} - 0.25 \text{ V} = 4.75 \text{ V}$$

$$R_{p(min)} = \frac{V_{Rp}}{I_{Rp(max)}} = \frac{4.75 \text{ V}}{24 \text{ mA}} = 198 \Omega$$

$R_{p(min)}$  for (a), (b), and (c) is the same value.

24. See Figure 14-5.



### Section 14-5 Com

### TL Performance

25. **F series:** SPP =  $3.3 \text{ ns} \times 6 \text{ mW} = 19.8 \text{ pJ}$   
**LS series:** SPP =  $10 \text{ ns} \times 2.2 \text{ mW} = 22 \text{ pJ}$   
**ALS series:** SPP =  $7 \text{ ns} \times 1.4 \text{ mW} = 9.8 \text{ pJ}$   
**ABT series:** SPP =  $3.2 \text{ ns} \times 17 \mu\text{W} = 0.0544 \text{ pJ}$   
**HC series:** SPP =  $7 \text{ ns} \times 2.75 \mu\text{W} = 0.01925 \text{ pJ}$   
**AC series:** SPP =  $5 \text{ ns} \times 0.55 \mu\text{W} = 0.00275 \text{ pJ}$   
**AHC series:** SPP =  $3.7 \text{ ns} \times 2.75 \mu\text{W} = 0.010175 \text{ pJ}$   
**LV series:** SPP =  $9 \text{ ns} \times 1.6 \mu\text{W} = 0.0144 \text{ pJ}$   
**LVC series:** SPP =  $4.3 \text{ ns} \times 0.8 \mu\text{W} = 0.00344 \text{ pJ}$

**ALVC series:** SPP = 3 ns × 0.8 μW = 0.0024 pJ

**ALVC** has the best (lowest value) speed-power product. It is, however, misleading to compare CMOS and TTL in terms of SPP because the power of CMOS goes up with frequency.

- 26.** (a) ALVC  
 (b) AHC  
 (c) AC  
 (d) ALVC
- 27.** (a) A and B to X: 3(3.3 ns) = 9.9 ns  
 C and D to X: 2(3.3 ns) = 6.6 ns
- (b) A to X<sub>1</sub>, X<sub>2</sub>, X<sub>3</sub>: 2(7 ns) = 14 ns  
 B to X<sub>1</sub>: 7 ns  
 C to X<sub>2</sub>: 7 ns  
 D to X<sub>3</sub>: 7 ns
- (c) A, B to X: 3(3.7 ns) = 11.1 ns  
 C, D, to X: 2(3.7 ns) = 7.4 ns
- 28.** (a) HC has an  $f_{\max} = 50$  MHz  

$$f_{\text{clock}} = \frac{1}{50 \text{ ns}} = 20 \text{ MHz}$$
- (b) LS has an  $f_{\max} = 33$  MHz  

$$f_{\text{clock}} = \frac{1}{60 \text{ ns}} = 16.7 \text{ MHz}$$
- (c) AHC has an  $f_{\max} = 170$  MHz  

$$f_{\text{clock}} = \frac{1}{4 \text{ ns}} = 250 \text{ MHz}$$

Since  $f_{\text{clock}} > f_{\max}$  for the AHC flip-flop, the output will be erratic.

### **Section 14-6 Emitter-Coupled Logic (ECL) Circuits**

- 29.** ECL operates with nonsaturated BJTs whereas TTL transistors saturate when turned on.
- 30.** (a) Lowest propagation delay - ECL  
 (b) Lowest power - HCMOS  
 (c) Lowest speed/power product - HCMOS