

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

Online Instructor's Manual
for

Digital Fundamentals

A Systems Approach

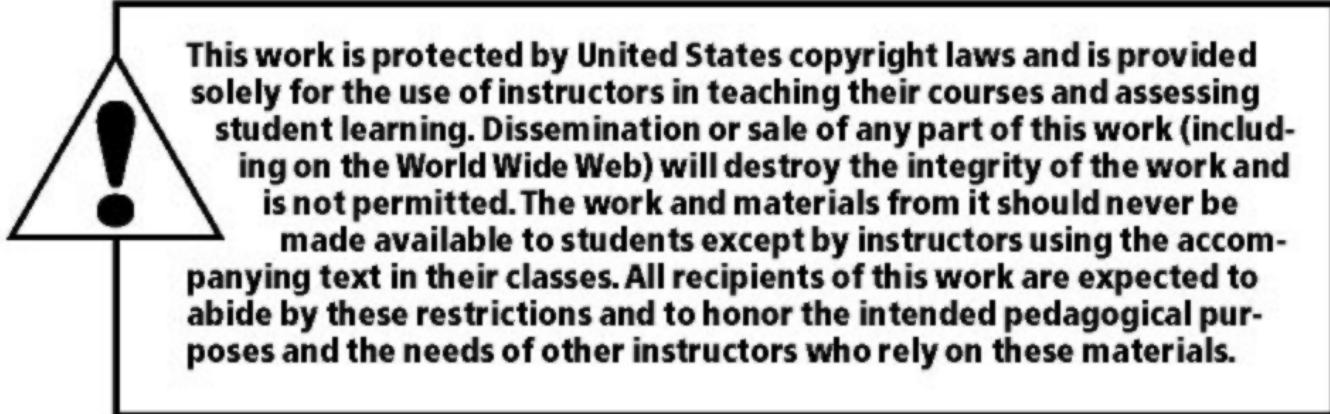
Thomas L. Floyd

PEARSON

Boston Columbus Indianapolis New York San Francisco Upper Saddle River

Amsterdam Cape Town Dubai London Madrid Milan Munich Paris Montreal Toronto

Delhi Mexico City Sao Paulo Sydney Hong Kong Seoul Singapore Taipei Tokyo



Copyright 2013 Pearson Education, Inc., publishing as Prentice Hall, 1 Lake Street, Upper Saddle River, New Jersey, 07458. All rights reserved. Manufactured 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. To obtain permission(s) to use material from this work, please submit a written request to Pearson Education, Inc., Permissions Department, 1 Lake Street, Upper Saddle River, New Jersey 07458.

Many of the designations by manufacturers and seller to distinguish their products are claimed as trademarks. Where those designations appear in this book, and the publisher was aware of a trademark claim, the designations have been printed in initial caps or all caps.

10 9 8 7 6 5 4 3 2 1

PEARSON

ISBN13: 978-0-13-298983-1

ISBN10: 0-13-298983-2

Problem Solutions

CHAPTER 1

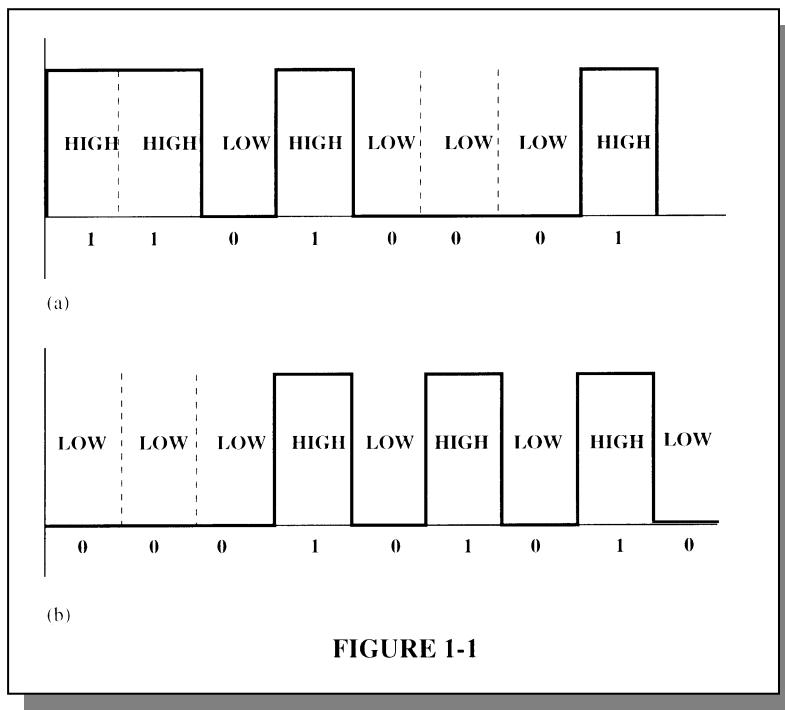
INTRODUCTION TO DIGITAL SYSTEMS

Section 1-1 Digital and Analog Signals and Systems

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.
3. A clock, a thermometer, and a speedometer can have either an analog or a digital output.

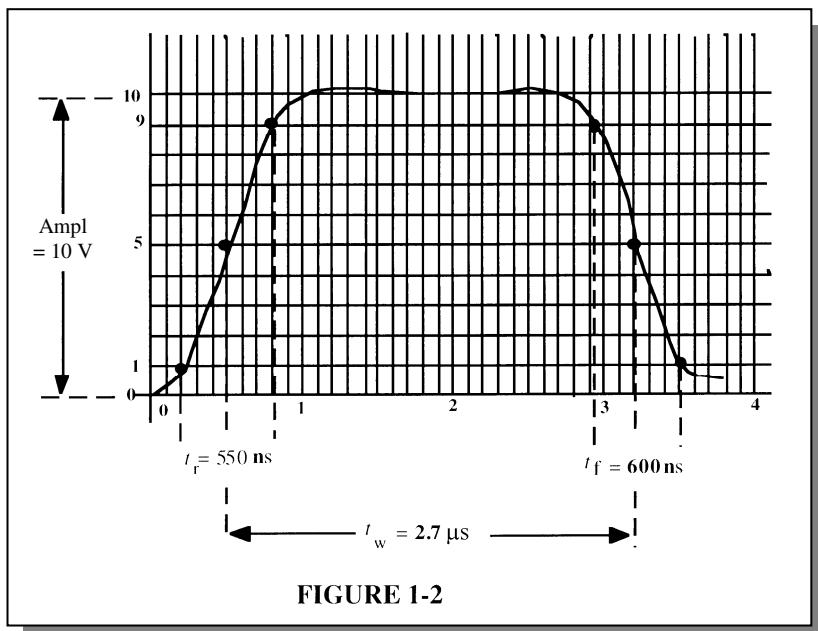
Section 1-2 Binary Digits, Logic Levels, and Digital Waveforms

4. In positive logic, a 1 is represented by a HIGH level and a 0 by a LOW level. In negative logic, a 1 is represented by a LOW level, and a 0 by a HIGH level.
5. HIGH = 1; LOW = 0. See Figure 1-1.

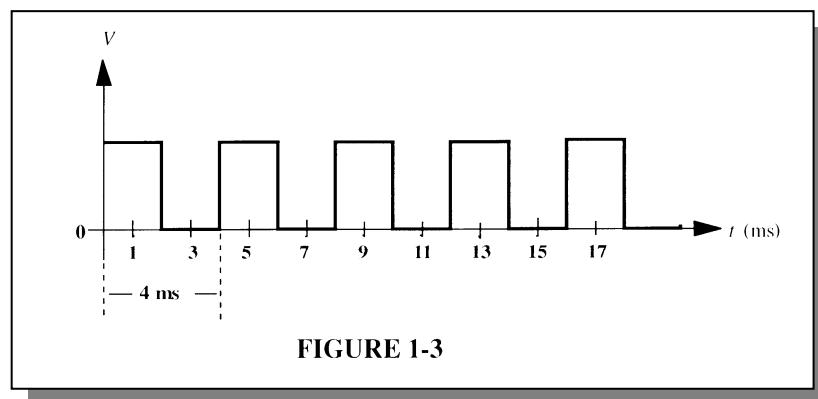


6. A 1 is a HIGH and a 0 is a LOW:
 - (a) HIGH, LOW, HIGH, HIGH, HIGH, LOW, HIGH
 - (b) HIGH, HIGH, HIGH, LOW, HIGH, LOW, LOW, HIGH

7. See Figure 1-2.



8. $T = 4 \text{ ms}$. See Figure 1-3.



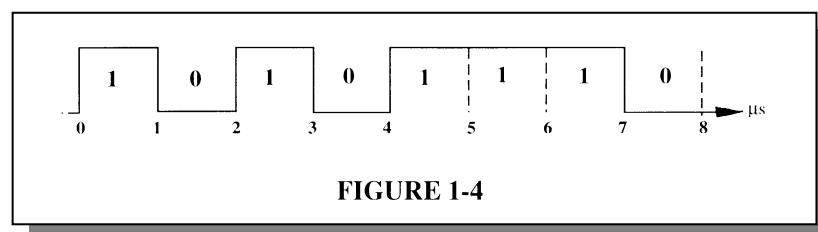
9. $f = \frac{1}{T} = \frac{1}{4 \text{ ms}} = 0.25 \text{ kHz} = 250 \text{ Hz}$

10. The waveform in Figure 1-63 is **periodic** because it repeats at a fixed interval.

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

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

12. See Figure 1-4.



Chapter 1

13. Each bit time = 1 μ s

$$\text{Serial transfer time} = (8 \text{ bits})(1 \mu\text{s}/\text{bit}) = 8 \mu\text{s}$$

Parallel transfer time = 1 bit time = 1 μ s

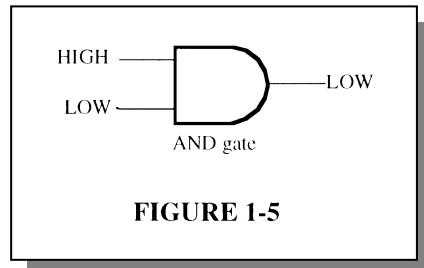
14. $T = \frac{1}{f} = \frac{1}{3.5 \text{ GHz}} = 0.286 \text{ ns}$

Section 1-3 Logic Operations

15. $L_{\text{ON}} = \text{SW1} + \text{SW2} + \text{SW1} \cdot \text{SW2}$

16. An AND gate produces a HIGH output only when *all* of its inputs are HIGH.

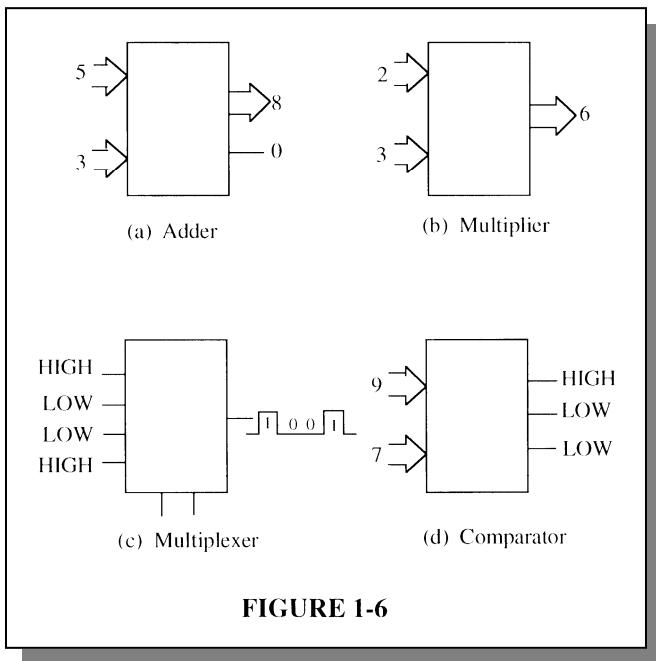
17. AND gate. See Figure 1-5.



18. 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 Combinational and Sequential Logic Functions

19. See Figure 1-6.



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

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

21. See Figure 1-7.



Section 1-5 Programmable Logic

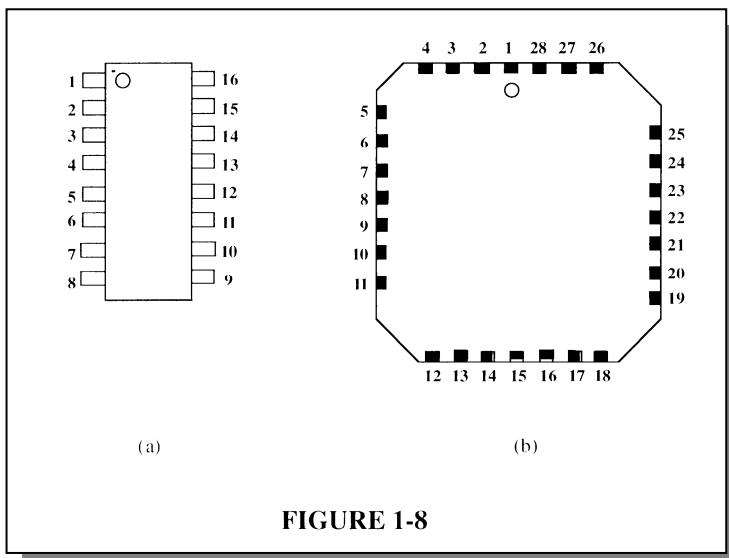
22. The following do not describe PLDs: ABEL, CUPL
23. (a) SPLD: Simple Programmable Logic Device
 (b) CPLD: Complex Programmable Logic Device
 (c) HDL: Hardware Description Language
 (d) FPGA: Field-Programmable Gate Array
 (e) GAL: Generic Array Logic
24. (a) Design entry: The step in a programmable 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.
25. 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 1-6 Fixed-Function Logic Devices

26. Circuits with complexities of from 100 to 10,000 equivalent gates are classified as large scale integration (LSI).
27. 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.

Chapter 1

28. See Figure 1-8.



Section 1-7 A System

29. The system can be implemented with a PLD, a microcontroller, or with fixed-function ICS.
30. The counter counts the tablets and determines when a bottle is full.
31. Register A stores the maximum number of tablets/bottle; Register B stores the cumulative total of tablets bottled.

Section 1-8 Measuring Instruments

32. Amplitude = top of pulse minus base line
 $V = 8 \text{ V} - 1 \text{ V} = 7 \text{ V}$

33. Amplitude = (3 div)(2 V /div) = **6 V**.

34. $T = (4 \text{ div})(2 \text{ ms/div}) = 8 \text{ ms}$

$$f = \frac{1}{T} = \frac{1}{8 \text{ ms}} = 125 \text{ Hz}$$

CHAPTER 2

NUMBER SYSTEMS, OPERATIONS, AND CODES

Section 2-1 The Decimal Number System

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 The Binary Numbers System

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) $1111 = 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 8 + 4 + 2 + 1 = 15$

Chapter 2

- 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

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) $0.0981 \cong 0.0 + 0.0 + 0.0 + 0.0625 + 0.03125 + 0.0 + 0.0 + 0.00390625 = 0.0001101$

Chapter 2

- 13.** (a) $\frac{15}{2} = 7, R = 1$ (LSB) (b) $\frac{21}{2} = 10, R = 1$ (LSB) (c) $\frac{28}{2} = 14, R = 0$ (LSB)
- $\frac{7}{2} = 3, R = 1$ $\frac{10}{2} = 5, R = 0$ $\frac{14}{2} = 7, R = 0$
- $\frac{3}{2} = 1, R = 1$ $\frac{5}{2} = 2, R = 1$ $\frac{7}{2} = 3, R = 1$
- $\frac{1}{2} = 0, R = 1$ (MSB) $\frac{2}{2} = 1, R = 0$ $\frac{3}{2} = 1, R = 1$
- $\frac{1}{2} = 0, R = 1$ (MSB) $\frac{1}{2} = 0, R = 1$ (MSB)
- (d) $\frac{34}{2} = 17, R = 0$ (LSB) (e) $\frac{40}{2} = 20, R = 0$ (LSB) (f) $\frac{59}{2} = 29, R = 1$ (LSB)
- $\frac{17}{2} = 8, R = 1$ $\frac{20}{2} = 10, R = 0$ $\frac{29}{2} = 14, R = 1$
- $\frac{8}{2} = 4, R = 0$ $\frac{10}{2} = 5, R = 0$ $\frac{14}{2} = 7, R = 0$
- $\frac{4}{2} = 2, R = 0$ $\frac{5}{2} = 2, R = 1$ $\frac{7}{2} = 3, R = 1$
- $\frac{2}{2} = 1, R = 0$ $\frac{2}{2} = 1, R = 0$ $\frac{3}{2} = 1, R = 1$
- $\frac{1}{2} = 0, R = 1$ (MSB) $\frac{1}{2} = 0, R = 1$ (MSB) $\frac{1}{2} = 0, R = 1$ (MSB)
- (g) $\frac{65}{2} = 32, R = 1$ (LSB) (h) $\frac{73}{2} = 36, R = 1$ (LSB)
- $\frac{32}{2} = 16, R = 0$ $\frac{36}{2} = 18, R = 0$
- $\frac{16}{2} = 8, R = 0$ $\frac{18}{2} = 9, R = 0$
- $\frac{8}{2} = 4, R = 0$ $\frac{9}{2} = 4, R = 1$
- $\frac{4}{2} = 2, R = 0$ $\frac{4}{2} = 2, R = 0$
- $\frac{2}{2} = 1, R = 0$ $\frac{2}{2} = 1, R = 0$
- $\frac{1}{2} = 0, R = 1$ (MSB) $\frac{1}{2} = 0, R = 1$ (MSB)

- 14.** (a) $0.98 \times 2 = 1.96$ 1 (MSB)
 $0.96 \times 2 = 1.92$ 1
 $0.92 \times 2 = 1.84$ 1
 $0.84 \times 2 = 1.68$ 1
 $0.68 \times 2 = 1.36$ 1
 $0.36 \times 2 = 0.72$ 0
 continue if more accuracy is desired
 0.11110
- (b) $0.347 \times 2 = 0.694$ 0 (MSB)
 $0.694 \times 2 = 1.388$ 1
 $0.388 \times 2 = 0.776$ 0
 $0.776 \times 2 = 1.552$ 1
 $0.552 \times 2 = 1.104$ 1
 $0.104 \times 2 = 0.208$ 0
 $0.208 \times 2 = 0.416$ 0
 continue if 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

- | | | | | | |
|----------------|--|-----|--|-----|---|
| 15. (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) | $\begin{array}{r} 1101 \\ + 1011 \\ \hline 11000 \end{array}$ |
| 16. (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) | $\begin{array}{r} 11010 \\ - 10111 \\ \hline 00011 \end{array}$ |

Chapter 2

17.	(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) $\begin{array}{r} 1001 \\ \times 110 \\ \hline 0000 \end{array}$
	$\begin{array}{r} 11 \\ \times 11 \\ \hline 1001 \end{array}$	$\begin{array}{r} 100 \\ \times 100 \\ \hline 1000 \end{array}$	$\begin{array}{r} 000 \\ \times 111 \\ \hline 111 \end{array}$	$\begin{array}{r} 1001 \\ \times 110 \\ \hline 1001 \end{array}$
			$\begin{array}{r} 111 \\ \times 100 \\ \hline 100011 \end{array}$	$\begin{array}{r} 1001 \\ \times 110 \\ \hline 110110 \end{array}$
	(e) $\begin{array}{r} 1101 \\ \times 1101 \\ \hline 1101 \end{array}$	(f) $\begin{array}{r} 1110 \\ \times 1101 \\ \hline 1110 \end{array}$		
	$\begin{array}{r} 1101 \\ \times 1101 \\ \hline 0000 \end{array}$	$\begin{array}{r} 1110 \\ \times 1101 \\ \hline 0000 \end{array}$		
	$\begin{array}{r} 1101 \\ \times 1101 \\ \hline 1101 \end{array}$	$\begin{array}{r} 1110 \\ \times 1101 \\ \hline 1110 \end{array}$		
	$\begin{array}{r} 1101 \\ \times 1101 \\ \hline 10101001 \end{array}$	$\begin{array}{r} 1110 \\ \times 1101 \\ \hline 10110110 \end{array}$		

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.** Zero is represented in 1's complement as all 0's (for +0) or all 1's (for -0).
- 20.** Zero is represented by all 0's only in 2's complement.
- 21.** (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.
- 22.** Take the 1's complement and add 1:
 - (a) $01 + 1 = 10$
 - (b) $000 + 1 = 001$
 - (c) $0110 + 1 = 0111$
 - (d) $0010 + 1 = 0011$
 - (e) $00011 + 1 = 00100$
 - (f) $01100 + 1 = 01101$
 - (g) $01001111 + 1 = 01010000$
 - (h) $11000010 + 1 = 11000011$

Section 2-6 Signed Numbers

- 23.** (a) Magnitude of 29 = 0011101
 $+ 29 = 00011101$
- (b) Magnitude of 85 = 1010101
 $- 85 = 11010101$
- (c) Magnitude of $100_{10} = 1100100$
 $+ 100 = 01100100$
- (d) Magnitude of 123 = 1111011
 $- 123 = 11111011$

- 24.** (a) Magnitude of 34 = 0100010

$$\begin{array}{r} -34 \\ \hline 11011101 \end{array}$$
- (c) Magnitude of 99 = 1100011

$$\begin{array}{r} -99 \\ \hline 10011100 \end{array}$$
- (b) Magnitude of 57 = 0111001

$$\begin{array}{r} +57 \\ \hline 00111001 \end{array}$$
- (d) Magnitude of 115 = 1110011

$$\begin{array}{r} +115 \\ \hline 01110011 \end{array}$$
- 25.** (a) Magnitude of 12 = 1100

$$\begin{array}{r} +12 \\ \hline 00001100 \end{array}$$
- (c) Magnitude of 101_{10} = 1100101

$$\begin{array}{r} +101_{10} \\ \hline 01100101 \end{array}$$
- (b) Magnitude of 68 = 1000100

$$\begin{array}{r} -68 \\ \hline 10111100 \end{array}$$
- (d) Magnitude of 125 = 1111101

$$\begin{array}{r} -125 \\ \hline 10000011 \end{array}$$
- 26.** (a) $10011001 = -25$ (b) $01110100 = +116$ (c) $10111111 = -63$
- 27.** (a) $10011001 = -(01100110) = -102$
(b) $01110100 = +(1110100) = +116$
(c) $10111111 = -(1000000) = -64$
- 28.** (a) $10011001 = -(1100111) = -103$
(b) $01110100 = +(1110100) = +116$
(c) $10111111 = -(1000001) = -65$
- 29.** (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 = 11000001100000000000000000
11000101011000001100000000000000
- 30.** (a) 110000001010010011100010000000000
Sign = 1
Exponent = 10000001 = $129 - 127 = 2$
Mantissa = $1.01001001110001 \times 2^2 = 101.001001110001$
 $-101.001001110001 = \mathbf{-5.15258789}$
- (b) 01100110010000111101001000000000
Sign = 0
Exponent = 11001100 = $204 - 127 = 77$
Mantissa = 1.10000111101001
1.10000111101001 $\times 2^{77}$

Chapter 2

Section 2-7 Arithmetic Operations with Signed Numbers

31. (a) $33 = 00100001$ 00100001
 $15 = 00001111$ $\underline{+ 00001111}$
 00110000
- (b) $56 = 00111000$ 00111000
 $27 = 00011011$ $\underline{+ 11100101}$
 $-27 = 11100101$
 00011101
- (c) $46 = 00101110$ 11010010
 $-46 = 11010010$ $\underline{+ 00011001}$
 $25 = 00011001$ 11101011
- (d) $110_{10} = 01101110$ 10010010
 $-110_{10} = 10010010$ $\underline{+ 10101100}$
 $84 = 01010100$
 $-84 = 10101100$
 100111110
32. (a) 00010110
 $\underline{+ 00110011}$
 01001001
- (b) 01110000
 $\underline{+ 10101111}$
 100011111
33. (a) 10001100
 $\underline{+ 00111001}$
 11000101
- (b) 11011001
 $\underline{+ 11100111}$
 11000000
34. (a) 00110011
 $\underline{- 00010000}$
 $\cancel{\cancel{A}} 00100011$
- (b) 01100101
 $\underline{- 11101000}$
 $\cancel{\cancel{A}} 01111101$
35. 01101010
 $\times \underline{11110001}$
 01101010
 01101010
 100111110
 01101010
 1011100110
 01101010
 11000110110

Changing to 2's complement with sign: 100111001010

36. $\frac{01000100}{00011001} = 00000010$
 $\frac{68}{25} = 2, \text{ remainder of } 18$

Section 2-8 Hexadecimal Numbers

37. (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) $8A9D_{16} = 1000\ 1010\ 1001\ 1101$

- 38.** (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) $1001\ 1000\ 0010 = 982_{16}$
- 39.** (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) $700_{16} = 7 \times 16^2 = 1792$
- 40.** (a) $\frac{8}{16} = 0$, remainder = 8
 hexadecimal number = 8_{16}
- (b) $\frac{14}{16} = 0$, remainder = 14 = E_{16}
 hexadecimal number = E_{16}
- (c) $\frac{33}{16} = 2$, remainder = 1 (LSD)
 $\frac{2}{16} = 0$, remainder = 2
 hexadecimal number = 21_{16}
- (d) $\frac{52}{16} = 3$, remainder = 4 (LSD)
 $\frac{3}{16} = 0$, remainder = 3
 hexadecimal number = 34_{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 = $11C_{16}$
- (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 = $B4A_{16}$
- (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 = $FB3_{16}$
- (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}
- 41.** (a) $37_{16} + 29_{16} = 60_{16}$
 (b) $A0_{16} + 6B_{16} = 10B_{16}$
 (c) $FF_{16} + BB_{16} = 1BA_{16}$

Chapter 2

- 42.** (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

- 43.** (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) $7765_8 = 7 \times 8^3 + 7 \times 8^2 + 6 \times 8^1 + 5 \times 8^0 = 3584 + 448 + 48 + 5 = 4085$

- 44.** (a) $\frac{15}{8} = 1$, remainder = 7 (LSD)
 $\frac{1}{8} = 0$, remainder = 1
octal number = 17_8
- (b) $\frac{27}{8} = 3$, remainder = 3 (LSD)
 $\frac{3}{8} = 0$, remainder = 3
octal number = 33_8
- (c) $\frac{46}{8} = 5$, remainder = 6 (LSD)
 $\frac{5}{8} = 0$, remainder = 5
octal number = 56_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
- (e) $\frac{100}{8} = 12$, remainder = 4 (LSD)
 $\frac{12}{8} = 1$, remainder = 4
 $\frac{1}{8} = 0$, remainder = 1
octal number = 144_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
- (g) $\frac{219}{8} = 27$, remainder = 3 (LSD)
 $\frac{27}{8} = 3$, remainder = 3
 $\frac{3}{8} = 0$, remainder = 3
octal number = 333_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

- 45.** (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) $100213_8 = 001\ 000\ 000\ 010\ 001\ 011$

- 46.** (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)

- 47.** (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) $156 = 0001\ 0101\ 0110$

- 48.** (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

Chapter 2

49. (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) $1051 = 0001\ 0000\ 0101\ 0001$

50. (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) $1000\ 0111\ 0000 = 870$

51. (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) $0110\ 0110\ 0110\ 0111 = 6667$

52. (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)
$$\begin{array}{r} 10000101 \\ + 01000111 \\ \hline 10000111 \end{array}$$

53. (a)

$$\begin{array}{r}
 1000 \\
 + 0110 \\
 \hline
 1110 \quad \textit{invalid} \\
 + 0110 \\
 \hline
 00010100
 \end{array}$$

(b)

$$\begin{array}{r}
 0111 \\
 + 0101 \\
 \hline
 1100 \quad \textit{invalid} \\
 + 0110 \\
 \hline
 00010010
 \end{array}$$

(c)

$$\begin{array}{r}
 1001 \\
 + 1000 \\
 \hline
 10001 \quad \textit{invalid} \\
 + 0110 \\
 \hline
 00010111
 \end{array}$$

(d)

$$\begin{array}{r}
 1001 \\
 + 0111 \\
 \hline
 10000 \quad \textit{invalid} \\
 + 0110 \\
 \hline
 00010110
 \end{array}$$

(e)

$$\begin{array}{r}
 00100101 \\
 + 00100111 \\
 \hline
 01001100 \quad \textit{invalid} \\
 + 0110 \\
 \hline
 01010010
 \end{array}$$

(f)

$$\begin{array}{r}
 01010001 \\
 + 01011000 \\
 \hline
 10101001 \quad \textit{invalid} \\
 + 0110 \\
 \hline
 000100001001
 \end{array}$$

(g)

$$\begin{array}{r}
 10011000 \\
 + 10010111 \\
 \hline
 10010111 \quad \textit{invalid} \\
 + 01100110 \\
 \hline
 000110010101
 \end{array}$$

(h)

$$\begin{array}{r}
 010101100001 \\
 + 011100001000 \\
 \hline
 110001101001 \quad \textit{invalid} \\
 + 0110 \\
 \hline
 0001001001101001
 \end{array}$$

Chapter 2

54. (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

55. 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 $0000_2 = 0000$

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

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

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

57. (a)
$$\begin{array}{r} 1010 \\ 1100 \end{array} \text{ Gray}$$

(b)
$$\begin{array}{r} 00010 \\ 00011 \end{array} \text{ Gray}$$

(c)
$$\begin{array}{r} 11000010001 \\ 10000011110 \end{array} \text{ Gray}$$

58. (a) $1 \rightarrow 00110001$
 (c) $6 \rightarrow 00110110$
 (e) $18 \rightarrow 0011000100111000$
 (g) $56 \rightarrow 0011010100110110$
 (i) $107 \rightarrow 00110001001100000110111$
- (b) $3 \rightarrow 00110011$
 (d) $10 \rightarrow 0011000100110000$
 (f) $29 \rightarrow 0011001000111001$
 (h) $75 \rightarrow 0011011100110101$

- 59.** (a) $0011000 \rightarrow \text{CAN}$ (b) $1001010 \rightarrow \mathbf{J}$
 (c) $0111101 \rightarrow =$ (d) $0100011 \rightarrow \#$
 (e) $0111110 \rightarrow >$ (f) $1000010 \rightarrow \mathbf{B}$
- 60.** $1001000 \ 1100101 \ 1101100 \ 1101100 \ 1101111 \ 0101110 \ 0100000$
 $\begin{array}{cccccccc} \mathbf{H} & \mathbf{e} & \mathbf{l} & \mathbf{l} & \mathbf{o} & . & \# \\ 1001000 & 1101111 & 1110111 & 0100000 & 1100001 & 1110010 & 1100101 \\ \mathbf{H} & \mathbf{o} & \mathbf{w} & \# & \mathbf{a} & \mathbf{r} & \mathbf{e} \\ 0100000 & 1111001 & 1101111 & 1110101 & 0111111 & & \\ \# & \mathbf{y} & \mathbf{o} & \mathbf{u} & ? & & \end{array}$
- 61.** $1001000 \ 1100101 \ 1101100 \ 1101100 \ 1101111 \ 0101110 \ 0100000$
 $\begin{array}{cccccccc} \mathbf{48} & \mathbf{65} & \mathbf{6C} & \mathbf{6C} & \mathbf{6F} & \mathbf{2E} & \mathbf{20} \\ 1001000 & 1101111 & 1110111 & 0100000 & 1100001 & 1110010 & 1100101 \\ \mathbf{48} & \mathbf{6F} & \mathbf{77} & \mathbf{20} & \mathbf{61} & \mathbf{72} & \mathbf{65} \\ 0100000 & 1111001 & 1101111 & 1110101 & 0111111 & & \\ \mathbf{20} & \mathbf{79} & \mathbf{6F} & \mathbf{75} & \mathbf{3F} & & \end{array}$
- 62.** 30 INPUT A, B
- | | | |
|----|---------|-----------|
| 3 | 0110011 | 33_{16} |
| 0 | 0110000 | 30_{16} |
| SP | 0100000 | 20_{16} |
| I | 1001001 | 49_{16} |
| N | 1001110 | $4E_{16}$ |
| P | 1010000 | 50_{16} |
| U | 1010101 | 55_{16} |
| T | 1010100 | 54_{16} |
| SP | 0100000 | 20_{16} |
| A | 1000001 | 41_{16} |
| , | 0101100 | $2C_{16}$ |
| B | 1000010 | 42_{16} |

Section 2-12 Error Detection and Correction Codes

- 63.** Code (b) 011101010 has five 1s, so it is in error.
- 64.** Codes (a) 11110110 and (c) 010101010101010 are in error because they have an even number of 1s.
- 65.** (a) 1 10100100 (b) 0 00001001 (c) 1 11111110

Chapter 2

66. (a)
$$\begin{array}{r} 1100 \\ + 1011 \\ \hline 0111 \end{array}$$

(b)
$$\begin{array}{r} 1111 \\ + 0100 \\ \hline 1011 \end{array}$$

(c)
$$\begin{array}{r} 100011100 \\ + 10011001 \\ \hline 110000101 \end{array}$$

67. (a)
$$\begin{array}{r} 1100 \\ + 0111 \\ \hline 1011 \end{array}$$

(b)
$$\begin{array}{r} 1111 \\ + 1011 \\ \hline 0100 \end{array}$$

(c)
$$\begin{array}{r} 100011100 \\ + 110000101 \\ \hline 010011001 \end{array}$$

In each case, you get the other number.

68.
$$\begin{array}{r} 101100100000 \\ 1010 \downarrow \downarrow \downarrow | \downarrow \downarrow \downarrow \\ 1001 \downarrow \downarrow \downarrow | \downarrow \downarrow \downarrow \\ 1010 \downarrow \downarrow \downarrow | \downarrow \downarrow \downarrow \\ 1100 \downarrow \downarrow \downarrow | \downarrow \downarrow \downarrow \\ 1010 \downarrow \downarrow \downarrow | \downarrow \downarrow \downarrow \\ 1100 \downarrow \downarrow \downarrow | \downarrow \downarrow \downarrow \\ 1010 \downarrow \downarrow \downarrow | \downarrow \downarrow \downarrow \\ 1100 \downarrow \downarrow \downarrow | \downarrow \downarrow \downarrow \\ 1010 \downarrow \downarrow \downarrow | \downarrow \downarrow \downarrow \\ 1100 \downarrow \downarrow \downarrow | \downarrow \downarrow \downarrow \\ 1010 \downarrow \downarrow \downarrow | \downarrow \downarrow \downarrow \\ \text{Remainder} = 0110 \end{array}$$

$$\begin{array}{r} 101100100110 \\ 1010 \downarrow \downarrow \downarrow | \downarrow \downarrow \downarrow \\ 1001 \downarrow \downarrow \downarrow | \downarrow \downarrow \downarrow \\ 1010 \downarrow \downarrow \downarrow | \downarrow \downarrow \downarrow \\ 1100 \downarrow \downarrow \downarrow | \downarrow \downarrow \downarrow \\ 1010 \downarrow \downarrow \downarrow | \downarrow \downarrow \downarrow \\ 1101 \downarrow \downarrow \downarrow | \downarrow \downarrow \downarrow \\ 1010 \downarrow \downarrow \downarrow | \downarrow \downarrow \downarrow \\ 1111 \downarrow \downarrow \downarrow | \downarrow \downarrow \downarrow \\ 1010 \downarrow \downarrow \downarrow | \downarrow \downarrow \downarrow \\ 0000 \end{array}$$

Append remainder to data. CRC is 101100100110.

69. Error in MSB of transmitted CRC:

$$\begin{array}{r} 001100100110 \\ 1010 \downarrow \downarrow \downarrow | \downarrow \downarrow \downarrow \\ 1001 \downarrow \downarrow \downarrow | \downarrow \downarrow \downarrow \\ 1010 \downarrow \downarrow \downarrow | \downarrow \downarrow \downarrow \\ 1100 \downarrow \downarrow \downarrow | \downarrow \downarrow \downarrow \\ 1010 \downarrow \downarrow \downarrow | \downarrow \downarrow \downarrow \\ 1101 \downarrow \downarrow \downarrow | \downarrow \downarrow \downarrow \\ 1010 \downarrow \downarrow \downarrow | \downarrow \downarrow \downarrow \\ 1110 \downarrow \downarrow \downarrow | \downarrow \downarrow \downarrow \\ 1010 \downarrow \downarrow \downarrow | \downarrow \downarrow \downarrow \\ 1000 \downarrow \downarrow \downarrow | \downarrow \downarrow \downarrow \\ 1010 \downarrow \downarrow \downarrow | \downarrow \downarrow \downarrow \\ 1011 \downarrow \downarrow \downarrow | \downarrow \downarrow \downarrow \\ 1010 \downarrow \downarrow \downarrow | \downarrow \downarrow \downarrow \\ 10 \end{array}$$

Remainder is 10, indicating an error.

CHAPTER 3

LOGIC GATES AND GATE COMBINATIONS

Section 3-1 Introduction to Boolean Algebra

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

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

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

(g) $A \overline{B} \overline{C} = 1$ 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

Chapter 3

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

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

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

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

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

7. (a) Commutative law of addition
 (b) Commutative law of multiplication
 (c) Distributive law

8. Refer to Table 3-1 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: $A + A = A$
 (d) Rule 6: $A + \bar{A} = 1$
 (e) Rule 10: $A + AB = A$
 (f) Rule 11: $A + \bar{A}B = A + B$ (applied to 1st and 3rd terms)

Section 3-2 The Inverter

9. See Figure 3-1.

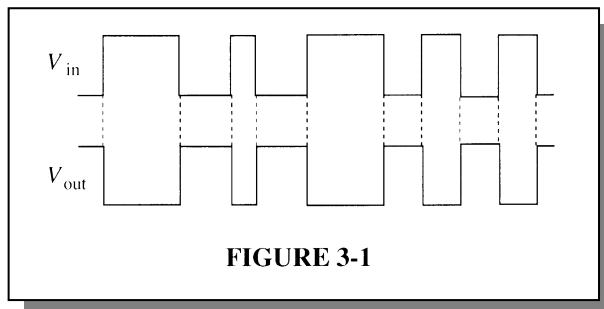


FIGURE 3-1

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

11. See Figure 3-2.

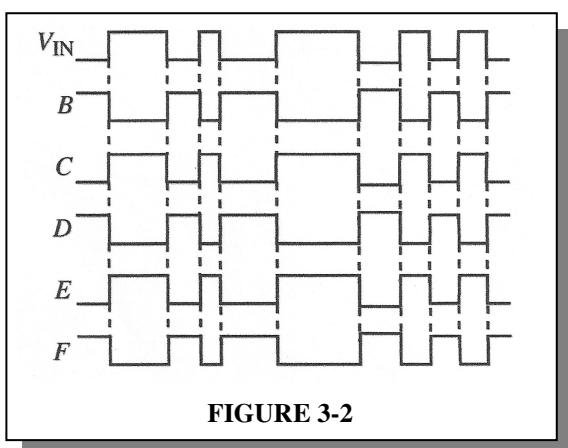


FIGURE 3-2

Section 3-3 The AND Gate

12. See Figure 3-3.

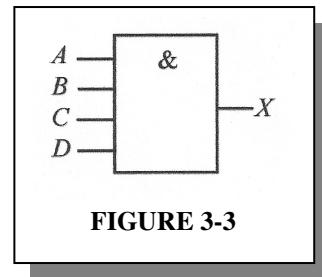


FIGURE 3-3

13. See Figure 3-4.

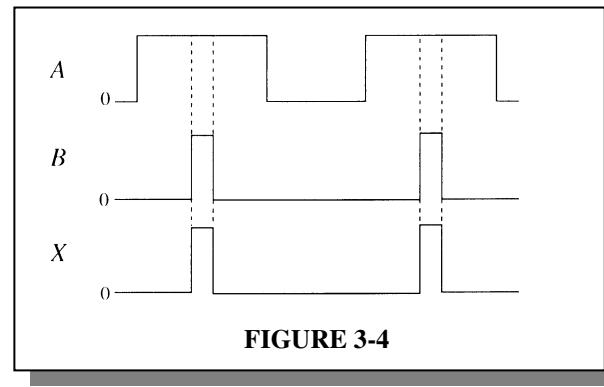
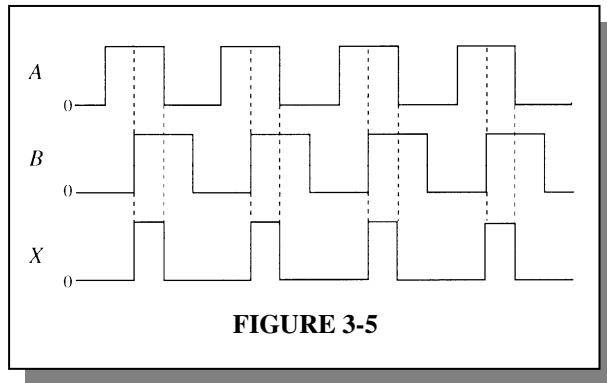


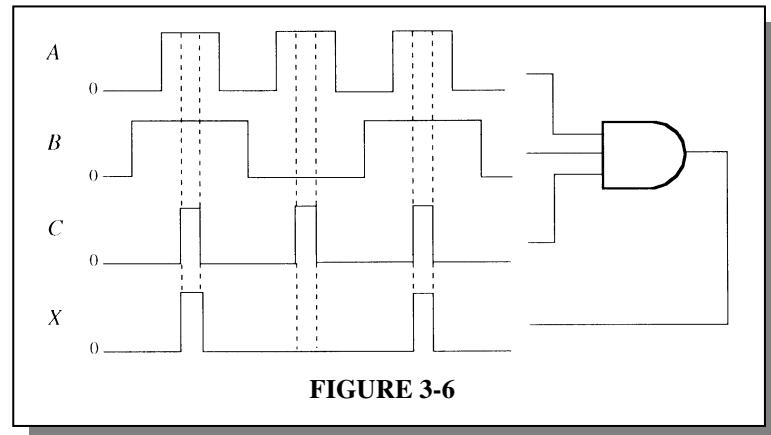
FIGURE 3-4

Chapter 3

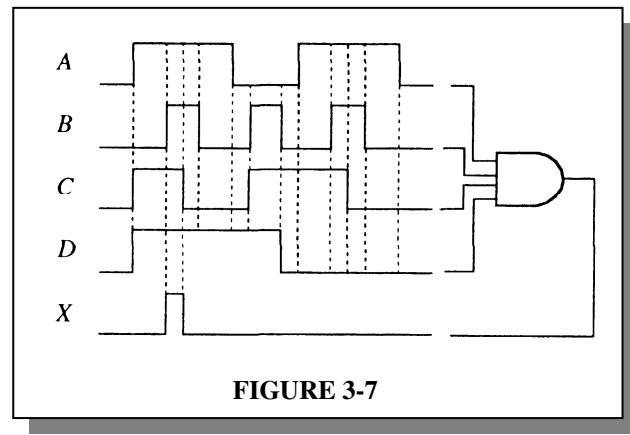
14. See Figure 3-5.



15. See Figure 3-6.



16. See Figure 3-7.



Section 3-4 The OR Gate

17. See Figure 3-8.

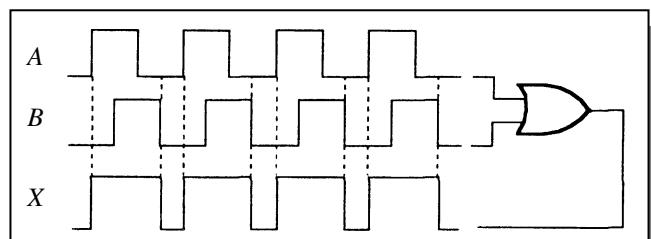


FIGURE 3-7

18. See Figure 3-9.

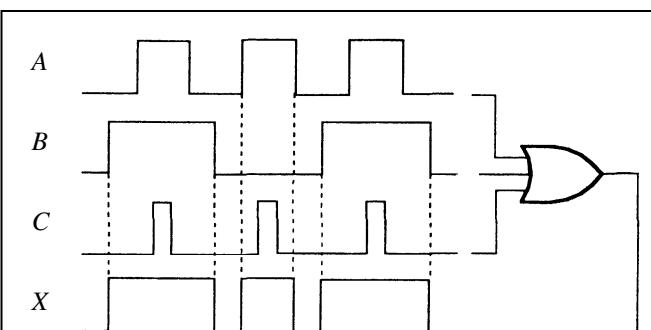


FIGURE 3-7

19. See Figure 3-10.

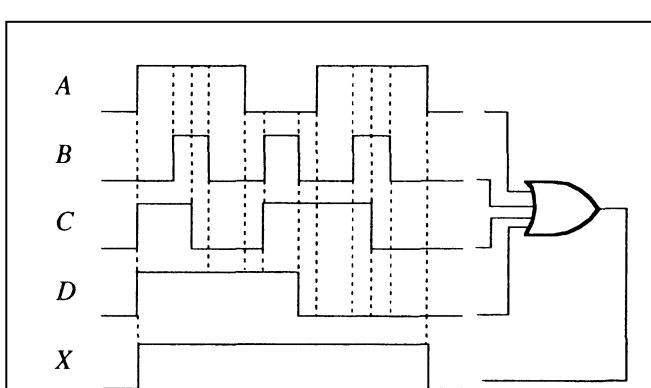


FIGURE 3-10

Chapter 3

20. See Figure 3-11.

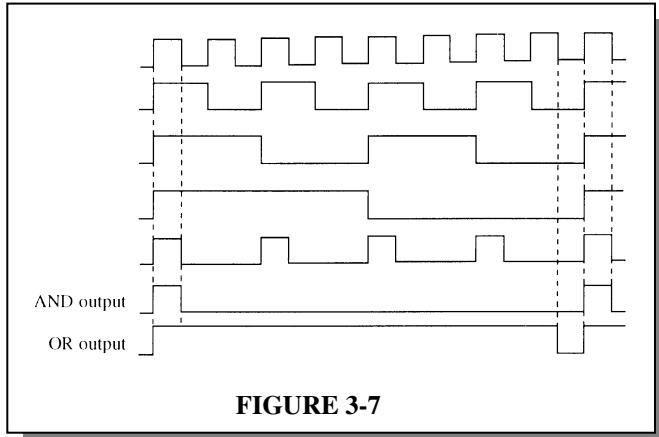


FIGURE 3-7

21. See Figure 3-12.

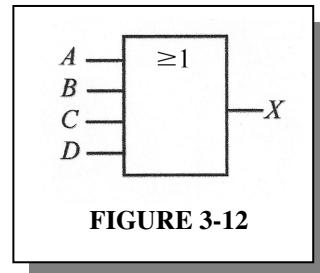


FIGURE 3-12

- 22.

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

Section 3-5 The NAND Gate

23. See Figure 3-13.

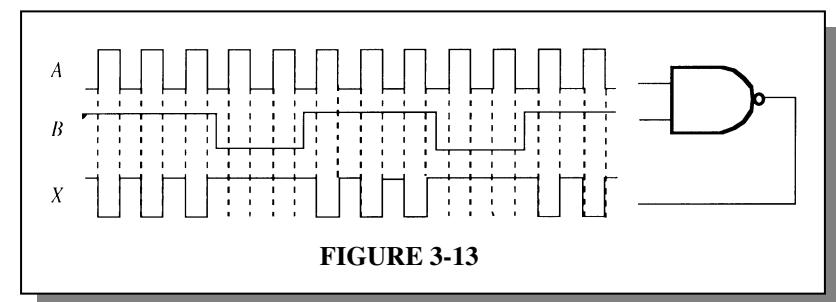


FIGURE 3-13

24. See Figure 3-14.

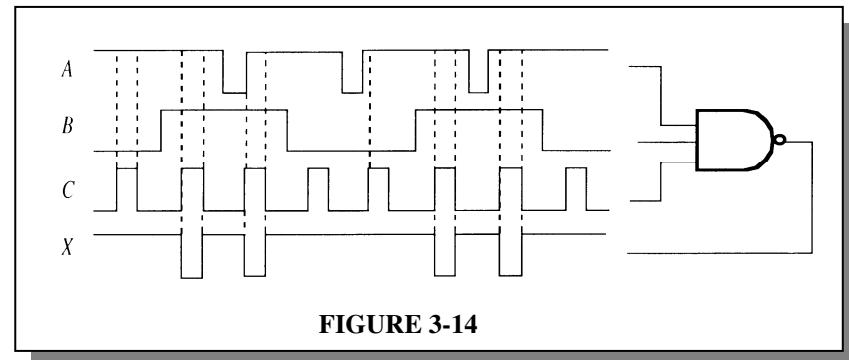


FIGURE 3-14

25. See Figure 3-15.

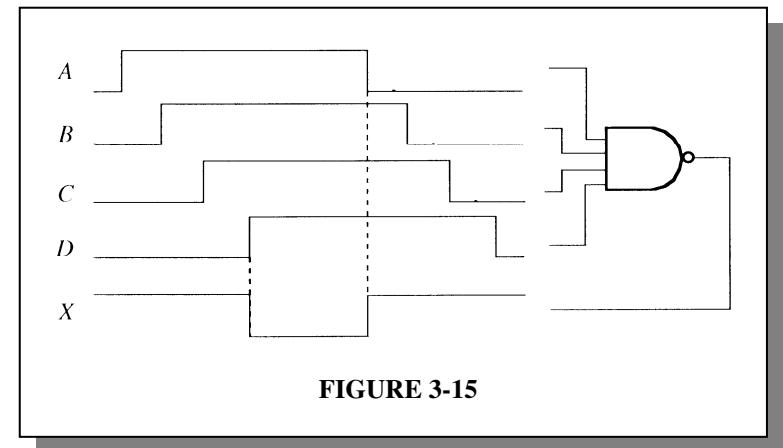


FIGURE 3-15

26. See Figure 3-16.

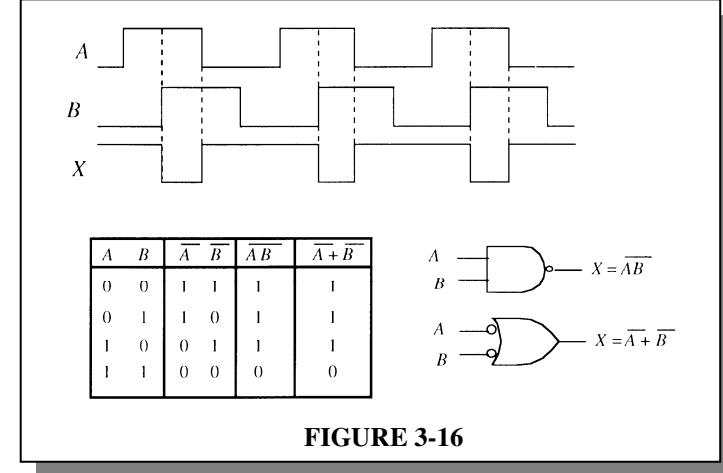
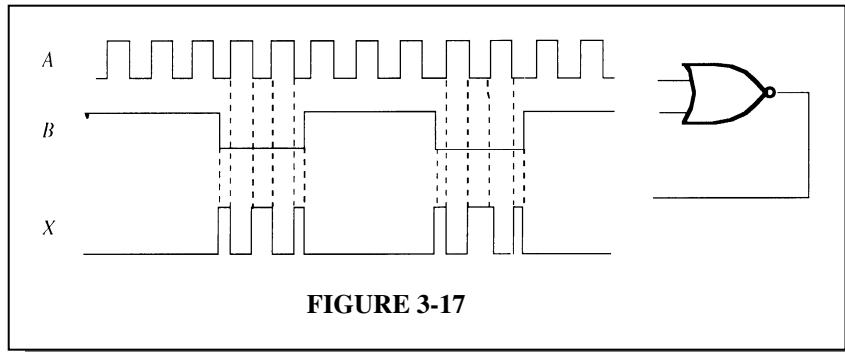


FIGURE 3-16

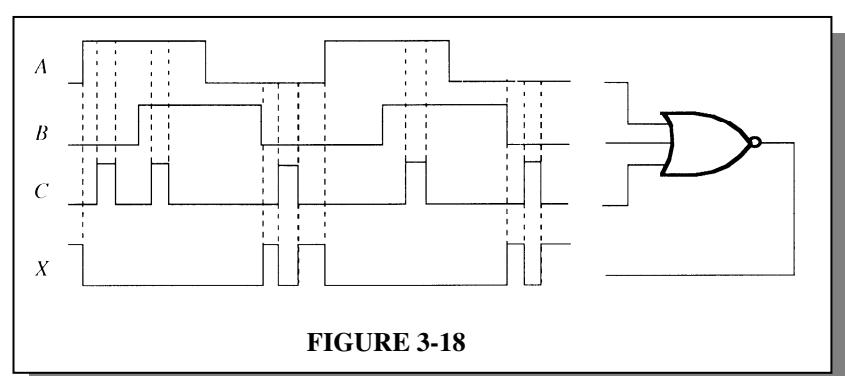
Chapter 3

Section 3-6 The NOR Gate

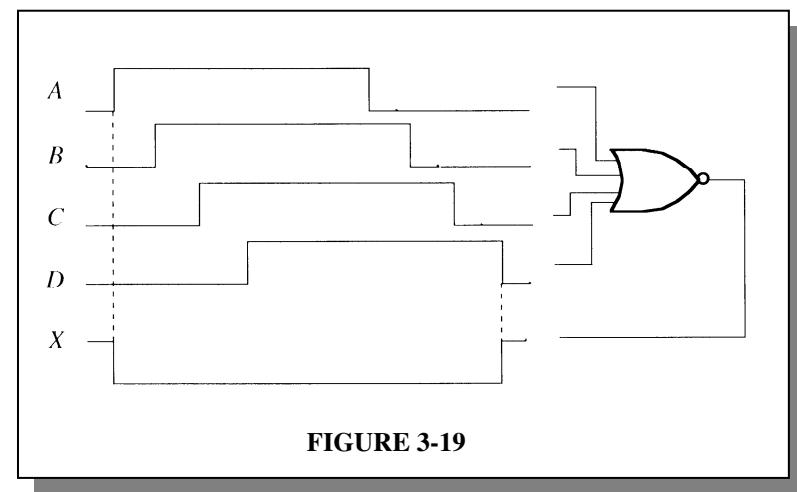
27. See Figure 3-17.



28. See Figure 3-18.



29. See Figure 3-19.



30. See Figure 3-20.

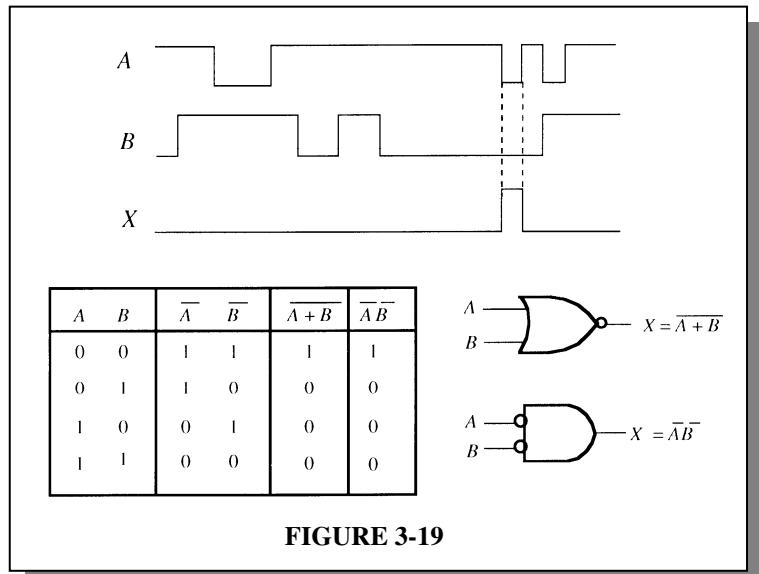


FIGURE 3-19

Section 3-7 The Exclusive-OR and Exclusive-NOR Gates

31. 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.

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

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

32. See Figure 3-21.

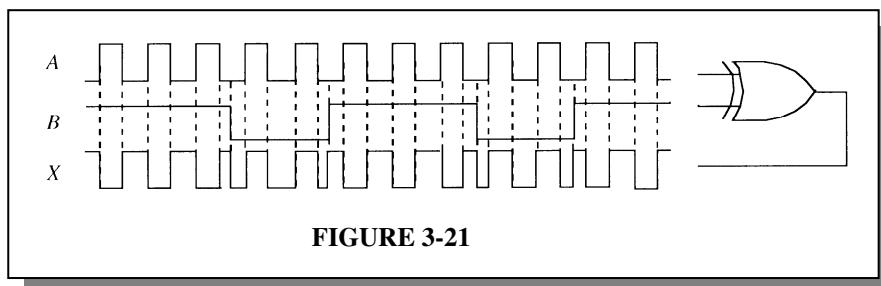


FIGURE 3-21

33. See Figure 3-22.

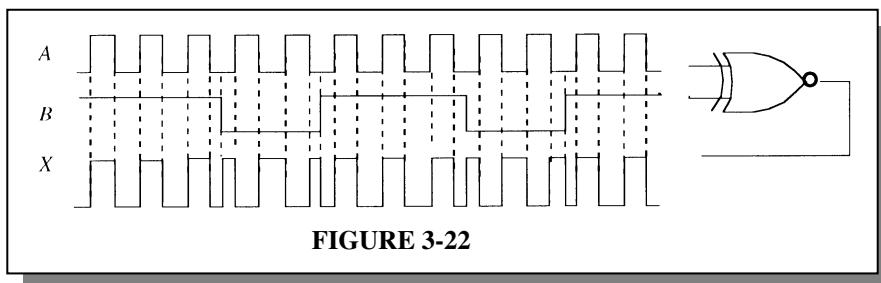
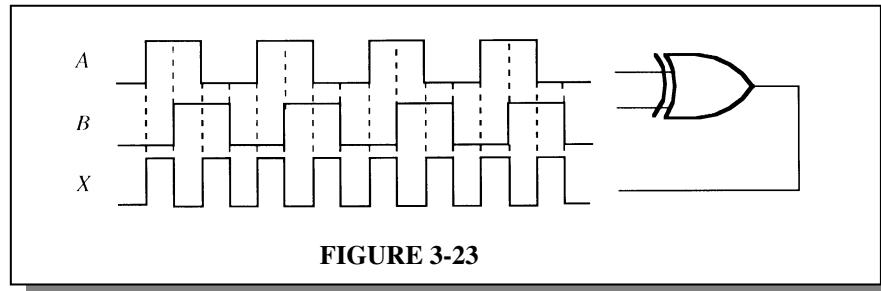


FIGURE 3-22

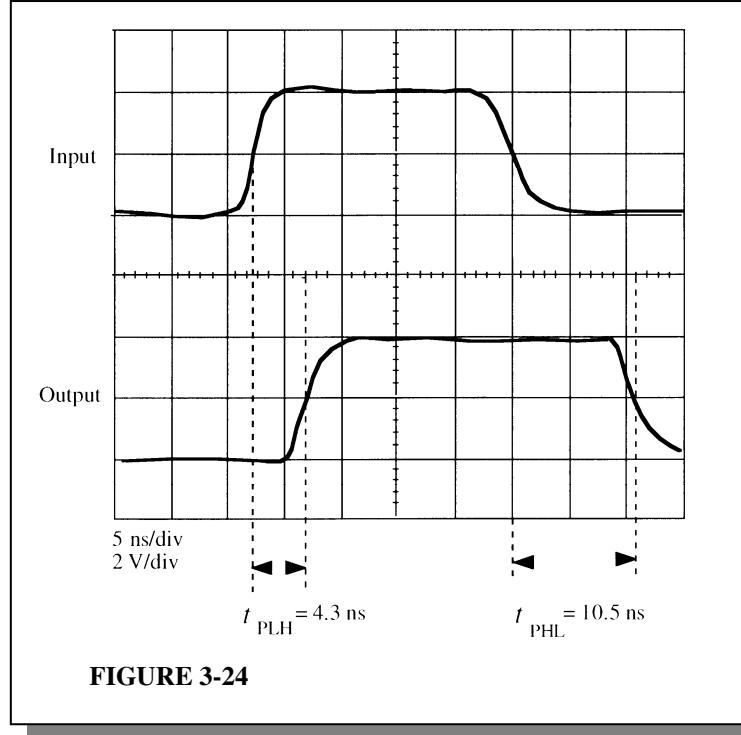
Chapter 3

34. See Figure 3-23.



Section 3-8 Gate Performance Characteristics and Parameters

35. See Figure 3-24.



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

$$37. P_D = V_{CC}I_C = (5 \text{ V})(4 \text{ mA}) = \mathbf{20 \text{ mW}}$$

$$38. P_{AVG} = \left(\frac{I_{CCH} + I_{CCL}}{2} \right) V_{CC} = \left(\frac{0.5 \text{ mA} + 1 \text{ mA}}{2} \right) 5 \text{ V} = \mathbf{3.75 \text{ mW}}$$

Section 3-9 Programmable Logic

39. $X_1 = \overline{AB}$

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

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

40. $X_1 = \overline{ABC}$

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 $A, \overline{A}, \overline{B}, \overline{B}$, and \overline{C} column fuses

$$X_2 = ABC$$

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

$$X_3 = \overline{ABC}$$

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

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

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

41. **entity ANDgate is**

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

```
end entity ANDgate;
```

```
architecture ANDfunction of ANDgate is
```

```
begin
```

```
  X <= A and B and C and D;
```

```
end architecture ANDfunction;
```

42. **module ANDgate (A, B, C, D, X);**

```
input A, B, C, D;
```

```
output X;
```

```
assign X = A && B && C && D;
```

```
endmodule
```

Chapter 3

```
43. entity NORgate is
    port (A, B, C, D, E: in bit; X: out bit);
end entity NORgate;
architecture NORfunction of NORgate is
begin
    X <= not(A or B or C or D or E);
end architecture NORfunction;

44. module NORgate (A, B, C, D, E, X);
    input A, B, C, D, E;
    output X;
    assign X = !(A || B || C || D || E);
endmodule
```

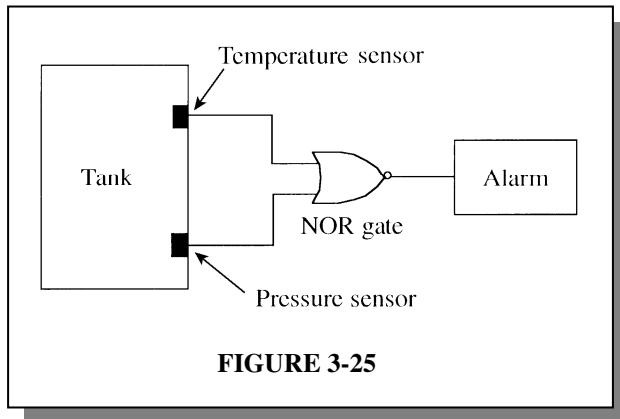
Section 3-10 Troubleshooting

45. Troubleshooting is the process of recognizing, isolating, and correcting a fault or failure in a system.
46. In the half-splitting method, a point half way between the input and output is checked for the presence or absence of a signal.
47. In the signal-tracing method, a signal is tracked as it progresses through a system until a point is found where the signal disappears or is incorrect.
48. In signal substitution, a generated signal replaces the normal input signal of a system or portion of a system. In signal injection a generated signal is injected into the system at a point where the normal signal has been determined to be faulty or missing.
49. When a failure is reported, determine when and how it failed and what are the symptoms.
50. No output signal can be caused by no dc power, no input signal, or a short or open that prevents the signal from getting to the output.
51. An incorrect output can be caused by an incorrect dc supply voltage, improper ground, incorrect component value, or a faulty component.
52. Some types of obvious things that you look for when a system fails are visible faults such as shorted wires, solder splashes, wire clippings bad or open connections, burned components. Also look for a signal that is incorrect in terms of amplitude shape, or frequency or the absence of a signal.
53. To isolate a fault in a system, apply half-splitting or signal tracing.
54. Two common troubleshooting instruments are the oscilloscope and the DMM.

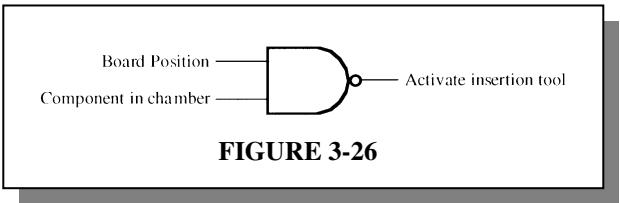
- 55.** When a fault has been isolated to a particular circuit board, the options are to repair the board or replace the board with a known good board.

Special Design Problems

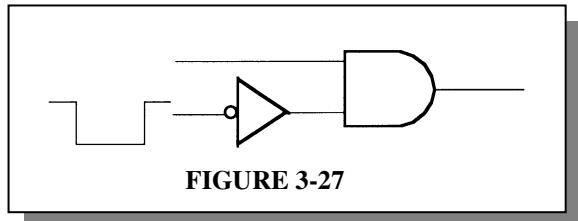
- 56.** See Figure 3-25.



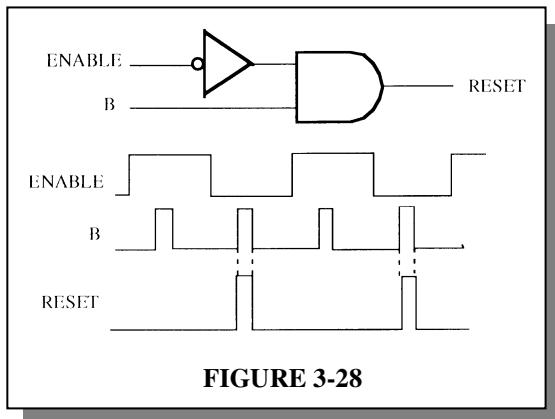
- 57.** See Figure 3-26.



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



- 59.** See Figure 3-28.



Chapter 3

60. See Figure 3-29.

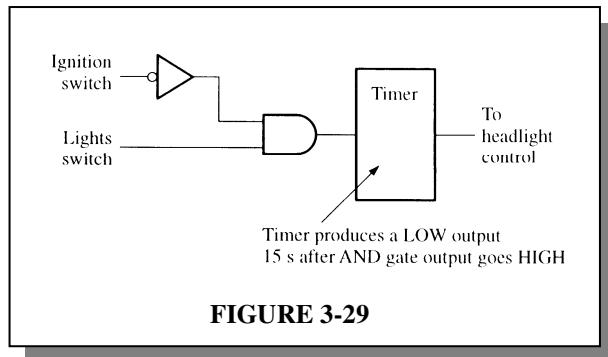


FIGURE 3-29

61. See Figure 3-30.

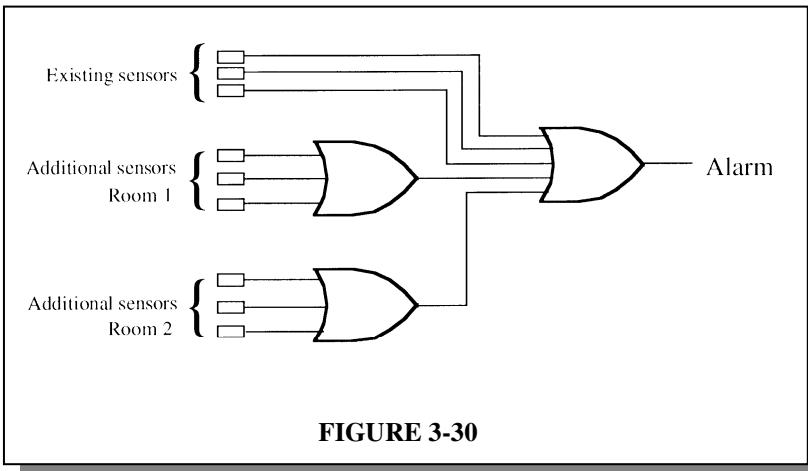


FIGURE 3-30

62. See Figure 3-31.

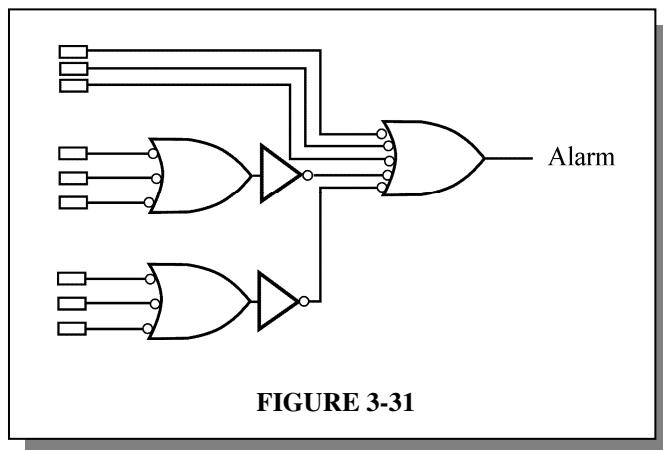


FIGURE 3-31

Multisim Troubleshooting Practice

63. Circuit fault: Input B of AND gate U1 shorted to ground.

Predicted effect of fault: Output X is always LOW.

Observed effect of introduced fault: Output X is always LOW.

64. Circuit fault: Input B of NAND gate shorted to VCC.

Predicted effect of fault: Output X is LOW whenever input A is HIGH.

Observed effect of introduced fault: Output X is LOW whenever input A is HIGH.

65. Observed operation: Output X is LOW whenever input B is HIGH.

Suspected fault: Input A or NOR gate shorted to ground.

Effect of introduced fault: Output X is LOW whenever input B is HIGH.

66. Observed operation: Output X is the inverse of input A.

Suspected fault: Input B of NOR gate shorted to VCC.

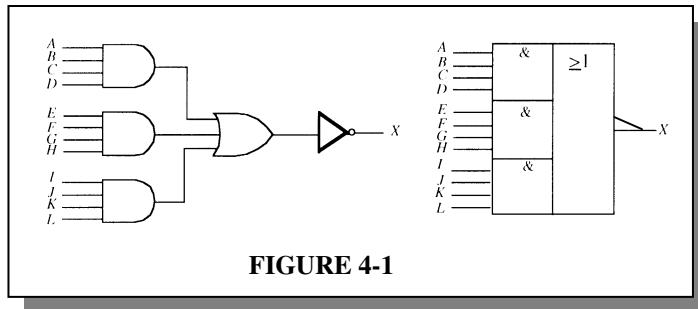
Effect of introduced fault: Output X is the inverse of input A.

CHAPTER 4

COMBINATIONAL LOGIC

Section 4-1 Basic Combinational Logic Circuits

1. See Figure 4-1.



2. (a) $X = \overline{AB} + \overline{A} + AC$
 (b) $X = \overline{\overline{AB} + \overline{ACD} + DB\overline{D}}$

3. (a) $X = ABB$
 (b) $X = AB + B$
 (c) $X = \overline{A} + B$
 (d) $X = (A + B) + AB$
 (e) $X = \overline{\overline{ABC}}$
 (f) $X = (A + B)(\overline{B} + C)$

4. (a) $X = ABB$

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

(b) $X = AB + B$

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

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

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

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

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

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

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

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

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

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

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	1
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	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

Chapter 4

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

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

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

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	X	0	0	X	1
X	X	X	X	0	X	X	0	1
X	X	X	X	0	X	0	X	0

For all other entries $X = 0$.

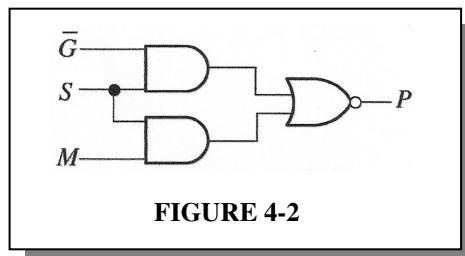
X = don't care

An abbreviated table is shown because there are 256 combinations.

Section 4-2 Boolean Expressions and Truth Tables

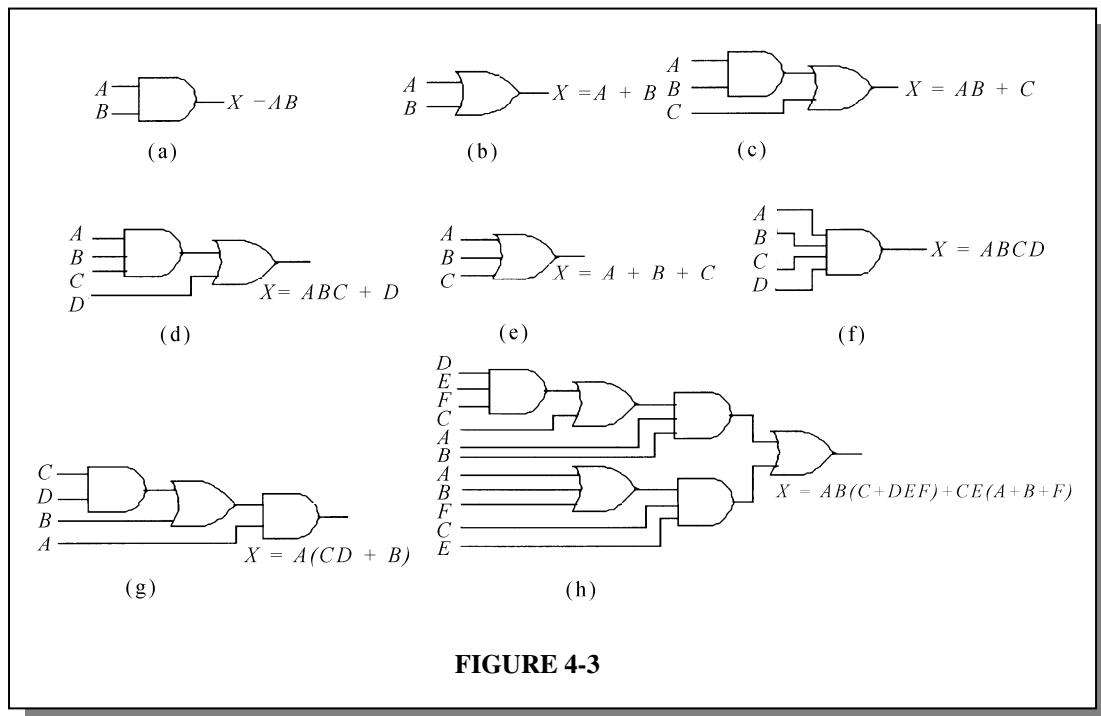
6. Let G = guard, S = switch, M = motor temp, and P = power. See Figure 4-2.

$$P = \overline{\overline{GS} + MS}$$



7. $X = \overline{\overline{ABCD} + EFGH}$

8. See Figure 4-3.



Chapter 4

9. See Figure 4-4.

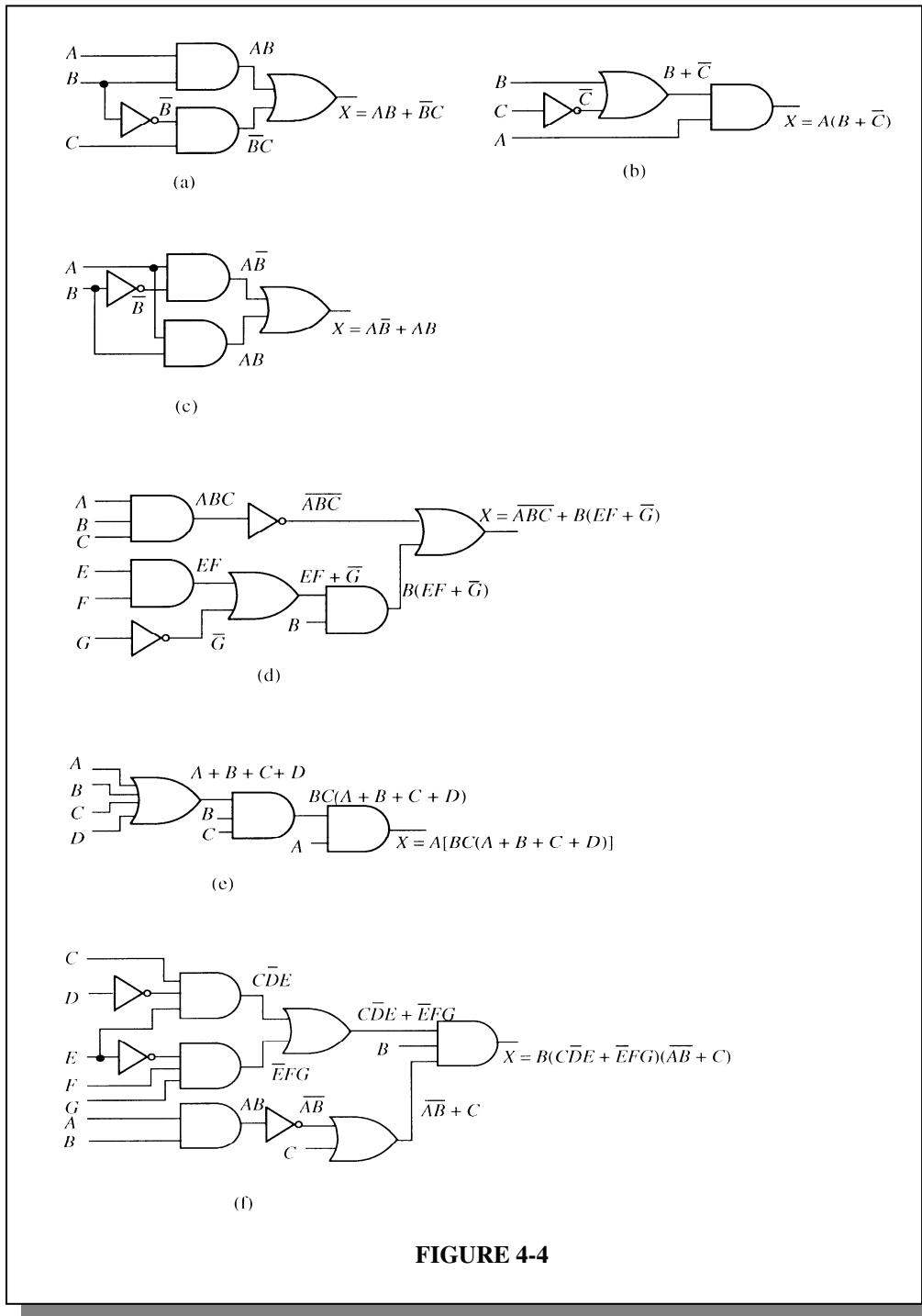


FIGURE 4-4

10. See Figure 4-5.

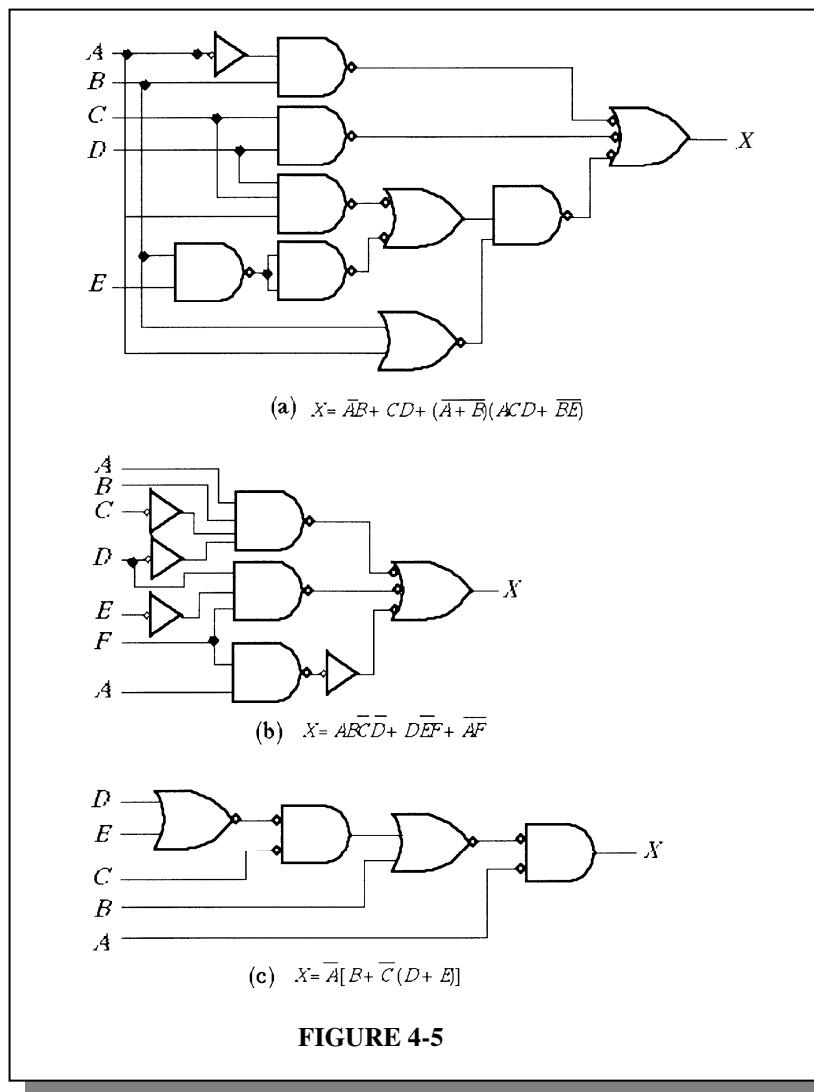


FIGURE 4-5

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

Five 3-input AND gates, one 5-input OR gate, and three inverters.

12. $X = \overline{ABC}\overline{D} + ABCD$

Eight 4-input AND gates, one 8-input OR gates, and four inverters.

Chapter 4

Section 4-7 Boolean Expressions and Truth Tables

13. (a) Table 4-1

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) Table 4-2

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

14. (a) Table 4-3

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) Table 4-4

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

15. (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) $X = \overline{ABC}\overline{D} + \overline{A}\overline{B}\overline{C}D + \overline{A}\overline{B}CD + \overline{ABC}\overline{D} + \overline{ABC}D + A\overline{B}\overline{C}\overline{D} + ABC\overline{D}$

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

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

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

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

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

Section 4-3 DeMorgan's Theorems

16.

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

17.

- (a) $\overline{\overline{AB(C+D)}} = \overline{\overline{AB}} + \overline{\overline{(C+D)}} = \overline{\overline{A}} + \overline{\overline{B}} + \overline{\overline{CD}}$
- (b) $\overline{\overline{AB(CD+EF)}} = \overline{\overline{AB}} + \overline{\overline{(CD+EF)}} = \overline{\overline{A}} + \overline{\overline{B}} + \overline{\overline{(CD)}}(\overline{\overline{EF}})$
 $= \overline{\overline{A}} + \overline{\overline{B}} + (\overline{\overline{C}} + \overline{\overline{D}})(\overline{\overline{E}} + \overline{\overline{F}})$
- (c) $\overline{\overline{(A+B+C+D)}} + \overline{\overline{ABCD}} = \overline{\overline{ABC}}\overline{\overline{D}} + \overline{\overline{A}} + \overline{\overline{B}} + \overline{\overline{C}} + \overline{\overline{D}}$
- (d) $\overline{\overline{(A+B+C+D)(ABCD)}} = (\overline{\overline{ABCD}})(\overline{\overline{A}} + \overline{\overline{B}} + \overline{\overline{C}} + \overline{\overline{D}})$
 $= \overline{\overline{ABCD}} + \overline{\overline{A}} + \overline{\overline{B}} + \overline{\overline{C}} + \overline{\overline{D}} = \overline{\overline{A}} + \overline{\overline{B}} + \overline{\overline{C}} + \overline{\overline{D}} + \overline{\overline{AB}}\overline{\overline{CD}}$
- (e) $\overline{\overline{AB(CD+EF)(AB+CD)}} = \overline{\overline{AB}} + \overline{\overline{(CD+EF)}} + (\overline{\overline{AB}} + \overline{\overline{CD}})$
 $= AB + (\overline{\overline{CD}})(\overline{\overline{EF}}) + (\overline{\overline{AB}})(\overline{\overline{CD}})$
 $= AB + (\overline{\overline{C}} + \overline{\overline{D}})(\overline{\overline{E}} + \overline{\overline{F}}) + ABCD$

18.

- (a) $\overline{\overline{\overline{(ABC)(EFG)}}} + \overline{\overline{\overline{(HIJ)(KLM)}}} = \overline{\overline{\overline{ABC}}} + \overline{\overline{\overline{EFG}}} + \overline{\overline{\overline{HIJ}}} + \overline{\overline{\overline{KLM}}}$
 $= \overline{\overline{\overline{ABC}}} + \overline{\overline{\overline{EFG}}} + \overline{\overline{\overline{HIJ}}} + \overline{\overline{\overline{KLM}}} = (\overline{\overline{\overline{ABC}}})(\overline{\overline{\overline{EFG}}})(\overline{\overline{\overline{HIJ}}})(\overline{\overline{\overline{KLM}}})$
 $= (\overline{\overline{A}} + \overline{\overline{B}} + \overline{\overline{C}})(\overline{\overline{E}} + \overline{\overline{F}} + \overline{\overline{G}})(\overline{\overline{H}} + \overline{\overline{I}} + \overline{\overline{J}})(\overline{\overline{K}} + \overline{\overline{L}} + \overline{\overline{M}})$
- (b) $\overline{\overline{(A+B+C+D)}} + \overline{\overline{BC}} = \overline{\overline{A}}(\overline{\overline{B}}\overline{\overline{C}})(\overline{\overline{CD}}) + BC = \overline{\overline{A}}(\overline{\overline{B}}\overline{\overline{C}})(\overline{\overline{CD}}) + BC$
 $= \overline{\overline{ABC}}(\overline{\overline{C}} + \overline{\overline{D}}) + BC = \overline{\overline{ABC}} + \overline{\overline{ABC}}\overline{\overline{D}} + BC = \overline{\overline{ABC}}(1 + \overline{\overline{D}}) + BC$
 $= \overline{\overline{ABC}} + BC$
- (c) $\overline{\overline{\overline{(A+B)(C+D)(E+F)(G+H)}}}$
 $= (\overline{\overline{A}} + \overline{\overline{B}})(\overline{\overline{C}} + \overline{\overline{D}})(\overline{\overline{E}} + \overline{\overline{F}})(\overline{\overline{G}} + \overline{\overline{H}}) = \overline{\overline{ABC}}\overline{\overline{DEF}}\overline{\overline{GH}}$

Chapter 4

Section 4-3 The Universal Property of NAND and NOR Gates

19. See Figure 4-6.

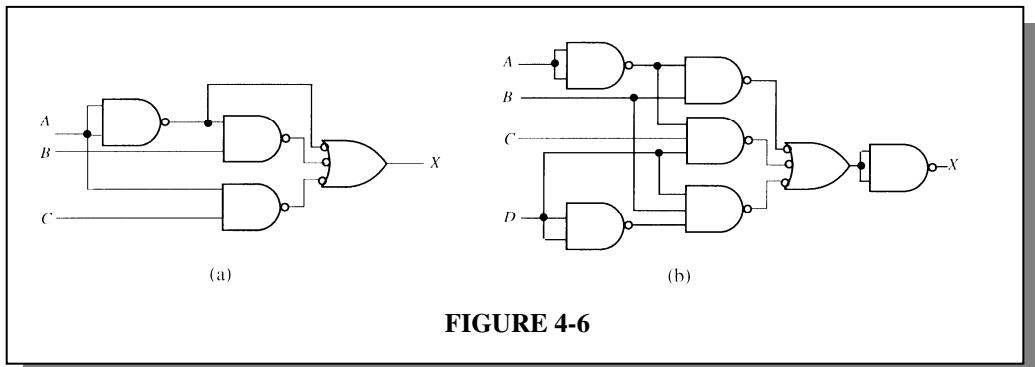


FIGURE 4-6

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

See Figure 4-7.

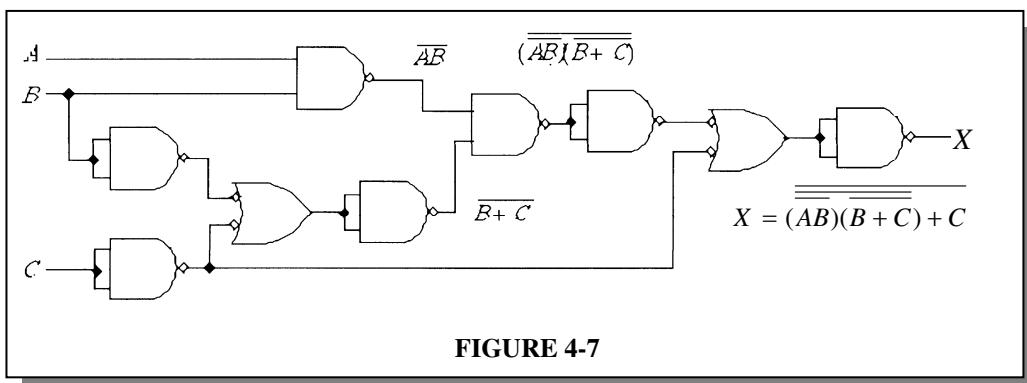


FIGURE 4-7

21. See Figure 4-8.

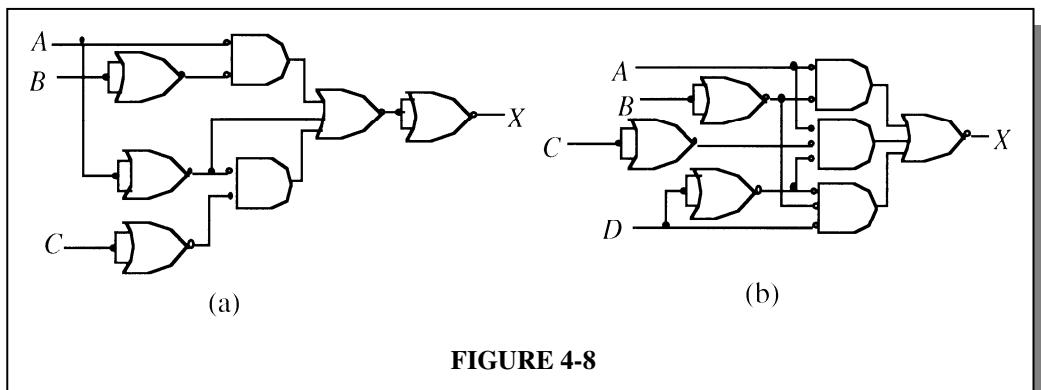
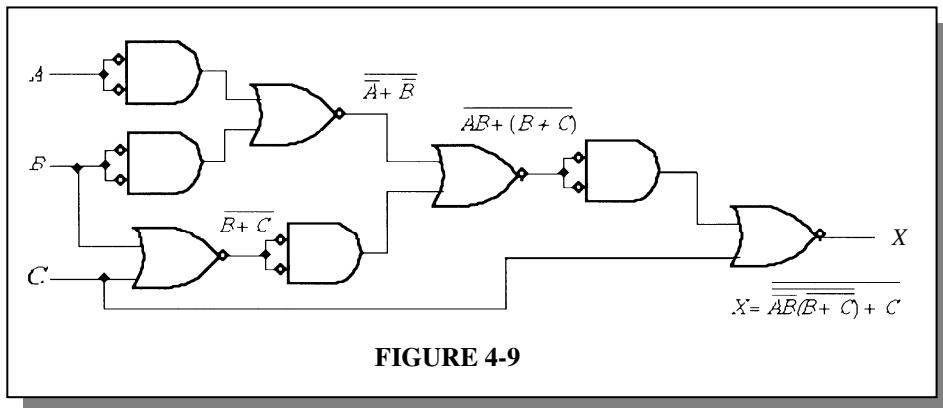


FIGURE 4-8

22. See Figure 4-9.

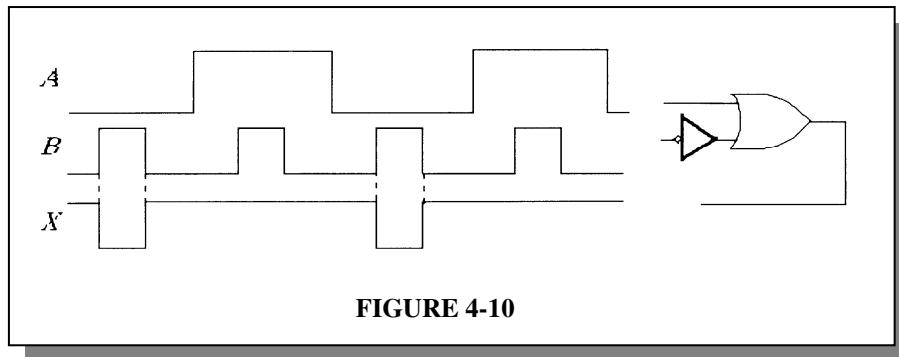


Section 4-5 Pulse Waveform Operation

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

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

See Figure 4-10.



Chapter 4

25. X is HIGH when ABC are all HIGH or when A is HIGH and B is LOW and C 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 4-11.

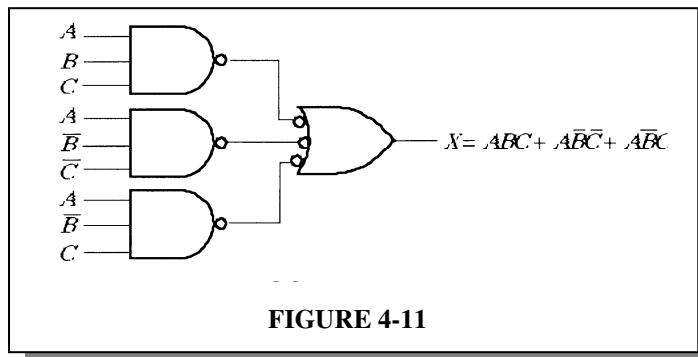


FIGURE 4-11

26. X is HIGH when A is HIGH, B is LOW, and C is LOW. We do not know if X is HIGH when all inputs are HIGH.

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

See Figure 4-12.

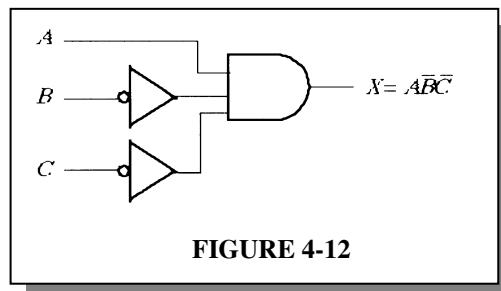
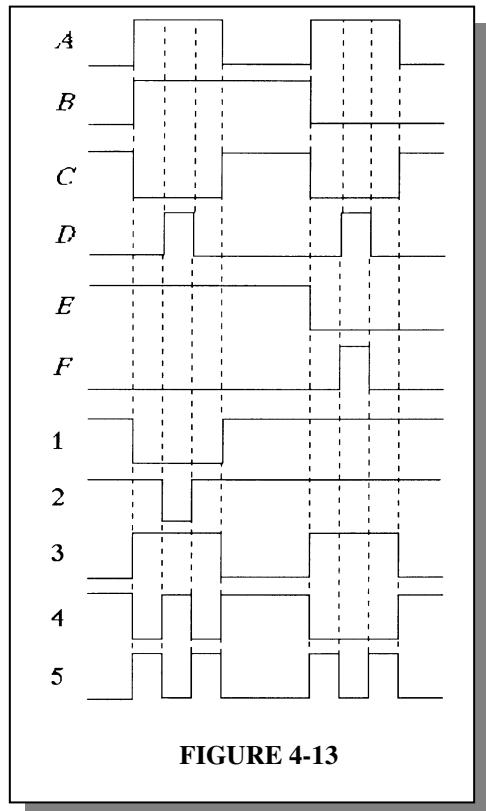
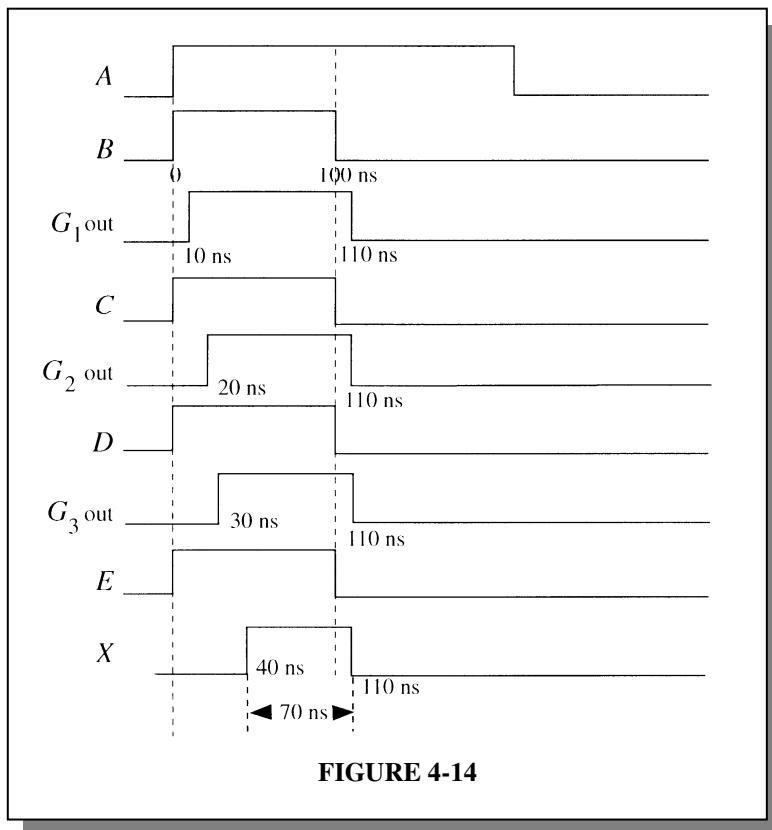


FIGURE 4-12

27. See Figure 4-13.



28. The output pulse is sufficiently wide. It is greater than 25 ns. A maximum is not specified. See Figure 4-14.



Chapter 4

Section 4-6 Combinational Logic with VHDL and Verilog

29. **VHDL:** A and B and C; Verilog: $X = A \&\& B \&\& C;$

30. --The VHDL program for Figure 4-37(b)

```
entity Circuit4_37b is
    port (A, B, C, D: in bit; X: out bit);
end entity Circuit4_37b;
architecture LogicFunction of Circuit4_37b 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;
```

```
//The Verilog program for Figure 4-37(b)
module Figure4_37b (A, B, C, D, X);
    input (A, B, C, D);
    output X;
    assign X = !(A && B) || !(A && C && D) || (D && B && !D);
endmodule
```

31. --The VHDL program for Figure 4-38(e) is

```
end entity Circuit4_38e is
architecture LogicFunction of Circuit4_38e is
begin
    X <= ((A nand B) and B) nand C;
end architecture LogicFunction;
```

```
//The Verilog for Figure 4-38(e)
module Circuit4_38e (A, B, C, X);
    input (A, B, C,);
    output X;
    assign X = !((A && B) && B) && C;
endmodule
```

--The VHDL program for Figure 4-38(f) is

```
entity Fig4_38f is
    port (A, B, C: in bit; X: out bit);
end entity Circuit4_38f;
architecture LogFunction of Circuit4_38f is
begin
    X <= (A or B) and (not B or C);
end architecture LogicFunction;
```

```
//The Verilog for Figure 4-38(f)
```

```
module Circuit4_38f (A, B, C, X);
    input (A, B, C);
    output X;
    X = (A || B) && (!B || C);
endmodule
```

32. See Figure 4-15 for input/output, gate, and signal labeling.

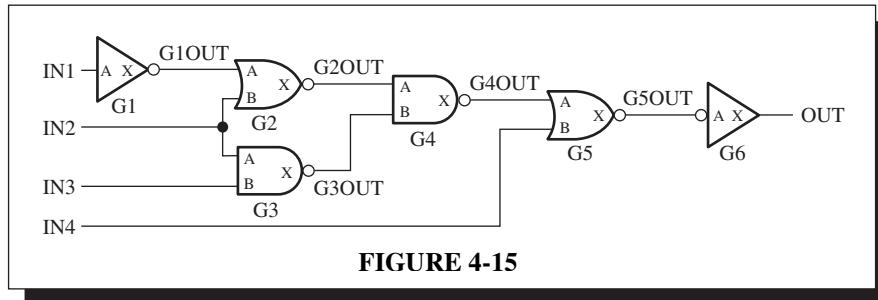


FIGURE 4-15

--VHDL program for the logic circuit in Figure 4-15 (textbook Figure 4-39(d))

```

entity Circuit4_39d is
    port (IN1, IN2, IN3, IN4: in bit; OUT: out bit);
end entity Circuit4_39d;
architecture LogicOperation of Circuit4_39d 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: NANDgate 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;
```

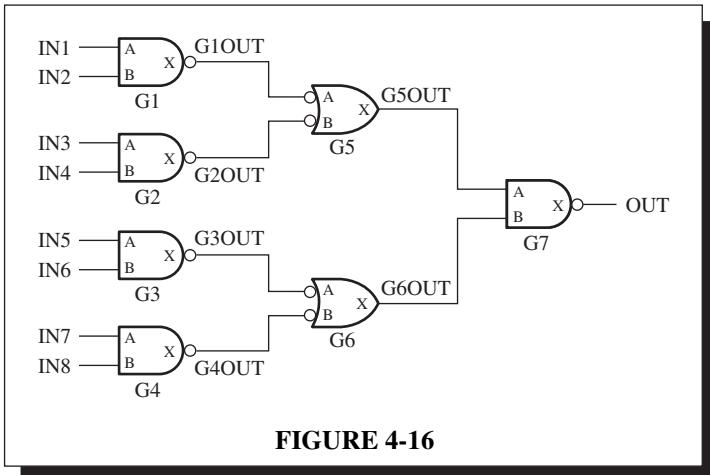
//The Verilog program for Figure 4-39(d) is

```

module Figure4_39d (A, B, C, D, X);
input (A, B, C, D);
output X;
assign X = !(!(!(!A || B)) && !(B && C)) || D;
endmodule
```

Chapter 4

33. See Figure 4-16 for input/output, gate, and signal labeling.



--VHDL program for the logic circuit in Figure 4-16 (textbook Figure 4-39(f))

```

entity Circuit4_39f is
    port (IN1, IN2, IN3, IN4, IN5, IN6, IN7, IN8: in bit; OUT: out bit);
end entity Circuit4_39f;
architecture LogicFunction of Circuit4_39f 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;
```

```

//The Verilog program for Figure 4-39(f) is
module Figure4_39f (A, B, C, D, E, F, G, H, X);
input (A, B, C, D, E, F, G, H);
output X;
assign X = !(((A && B)) || (C && D)) && ((E && F) || (G && H));
endmodule
```

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

This is the SOP expression for the function in Table 4-10 of the textbook. The following program applies the data flow approach for this logic function.

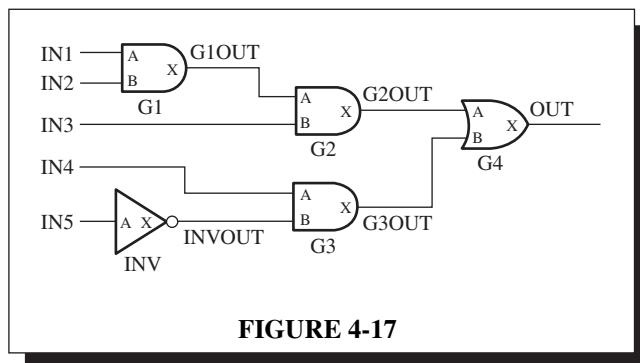
```
--VHDL program for Table4_10 SOP logic
entity Table4_10 is
    port (A, B, C: in bit; X: out bit);
end entity Table4_10;
architecture LogicOperation of Table4_10 is
begin
    X <= (not A and not B and not C) or (not A and B and not C)
        or (A and not B and not C) or (A and B and not C) or (A and B and C);
end architecture LogicOperation;

//The Verilog program for Figure 4-10 is
module Table4_10 (A, B, C, X);
input A, B, C;
output X;
assign X = !(A && !B && !C) || (!A && B && !C) || (A && !B && !C) || (A && B && !C)
    || (A && B && C);
endmodule
```

35. --VHDL program for textbook Figure4_47(a) data flow approach

```
entity Fig4_47a is
    port (A, B, C, D, E: in bit; X: out bit);
end entity Fig4_47a;
architecture DataFlow of Fig4_47a is
begin
    X <= (A and B and C) or (D and not E)
end architecture DataFlow;
```

See Figure 4-17 for the circuit in textbook Figure 4-47(a) modified for the structural approach.



--VHDL program for textbook Figure4_47(a) structural approach

```
entity Fig4_47a is
    port (IN1, IN2, IN3, IN4, IN5: in bit; OUT: out bit);
end entity Fig4_47a;
architecture Structure of Fig4_47a is
```

Chapter 4

```
--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;
```

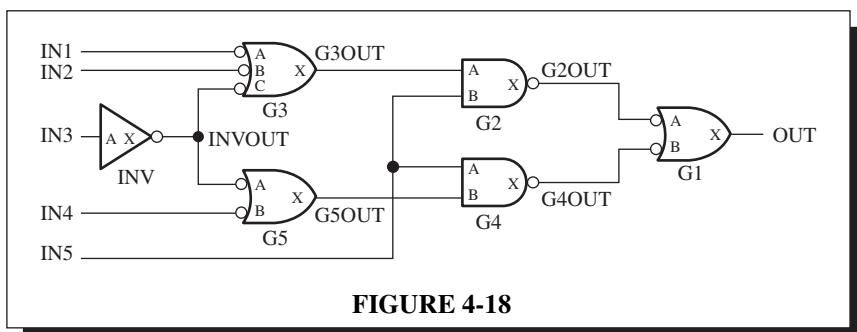
In Verilog, the program is

```
module Figure4_47a (A, B, C, D, E, X);
input (A, B, C, D, E);
output X;
assign X = (A && B && C) || (D && !E);
endmodule
```

36. --VHDL program for textbook Figure4-47(b) data flow approach

```
entity Fig4_47b is
    port (A, B, C, D, E: in bit; X: out bit);
end entity Fig4_47b;
architecture DataFlow of Fig4_47b is
begin
    X <= (not A or not B or C) and E or (C or not D) and E;
end architecture DataFlow;
```

See Figure 4-18 for the circuit in textbook Figure 4-47(b) labeled for the structural approach.



```
--VHDL program for textbook Fig4_47(b) structural approach
entity Fig4_47b is
    port (IN1, IN2, IN3, IN4, IN5: in bit; OUT: out bit);
end entity Fig4_47b;
architecture Structure of Fig4_47b 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;

//The Verilog program for Figure 4-47(b)
module Figure4_47b (A, B, C, D, E, X);
input A, B, C, D, E;
output X;
assign X = (!A || !B || C) && E || (C || D) && E;
endmodule
```

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

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

Chapter 4

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

38. --Program for textbook Figure4-48 data flow approach
entity Fig4_48 **is**

```

port (A1, A2, B1, B2: in bit; X: out bit);
end entity Fig4_48;
architecture LogicCircuit of Fig4_48 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;
```

39. The AND gates are numbered top to bottom G1, G2, G3, G4. The OR gate is G5 and the inverters are, top to bottom. G6 and G7. Change A_1, A_2, B_1, B_2 to IN1, IN2, IN3, IN4 respectively. Change X to OUT.

```

entity Circuit4_48 is
    port (IN1, IN2, IN3, (IN4: in bit; OUT: out bit);
end entity Circuit 4_48;
architecture Logic of Circuit 4_48 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;
```

```

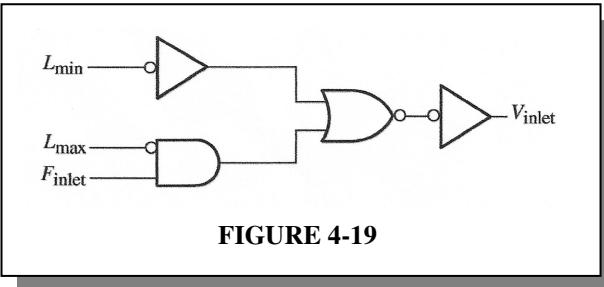
begin
  G1: AND_gate port map (A => IN1, B => IN2, X => G1OUT);
  G2: AND_gate port map (A => IN2, B => G6OUT, X => G2OUT);
  G3: AND_gate port map (A => G6OUT, B => G7OUT, X => G3OUT);
  G4: AND_gate port map (A => G7OUT, B => IN1, X => G4OUT);
  G5: OR_gate port map (A => G1OUT, B => G2OUT, X => G3OUT,
    D => G4OUT, X => OUT);
  G6: Inverter port map (A => IN3, X => G6OUT);
  G7: Inverter port map (A => IN4, X => G7OUT);
end architecture Logic;

```

Section 4-7 A System

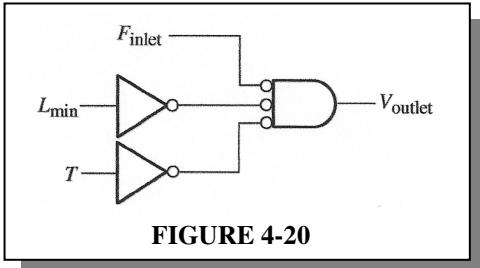
40. $V_{\text{inlet}} = \bar{L}_{\text{min}} + \bar{L}_{\text{max}} F_{\text{inlet}}$

See Figure 4-19.



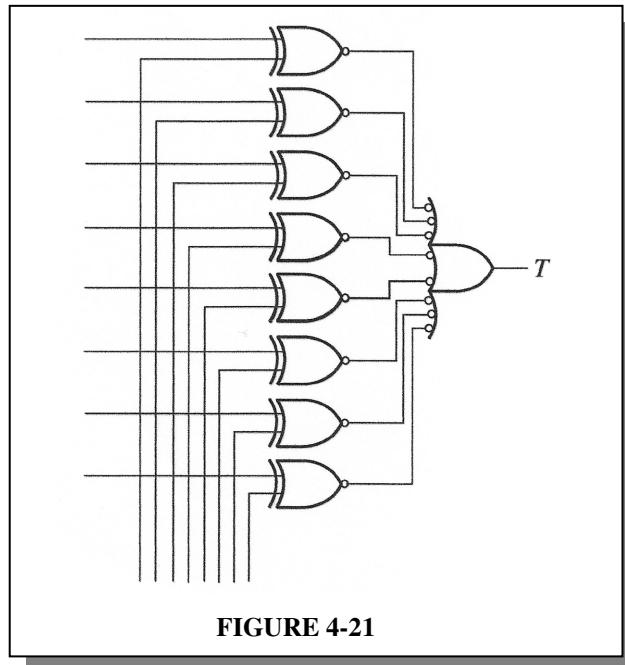
41. $V_{\text{outlet}} = L_{\text{min}} \bar{F}_{\text{inlet}} T$

See Figure 4-20.



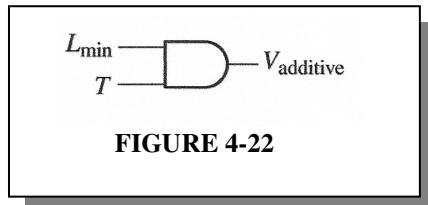
Chapter 4

42. See Figure 4-21



43. $V_{\text{additive}} = TL_{\min}$

See Figure 4-22.



Section 4-8 Troubleshooting

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

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.**

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

Since X is the same as the G_3 output, either G_1 or G_2 has failed with its output *stuck LOW*.

46. $X = AB + CD + EF$

X does not go HIGH when C and D are HIGH. G_2 has failed with the output *open* or *stuck HIGH* or the corresponding input to G_4 is *open*.

47. $X = \overline{\overline{A}\overline{B} + \overline{C}\overline{D} + \overline{E}\overline{F}} = (\overline{A}\overline{B})(\overline{C}\overline{D})(\overline{E}\overline{F}) = (A + B)(C + D)(E + F)$

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

$$48. \quad (a) \quad X = (\bar{A} + \bar{B} + C)E + (C + \bar{D})E = \bar{A}E + \bar{B}E + CE + \bar{C}E + \bar{D}E \\ = \bar{A}E + \bar{B}E + CE + \bar{D}E$$

See Figure 4-23.

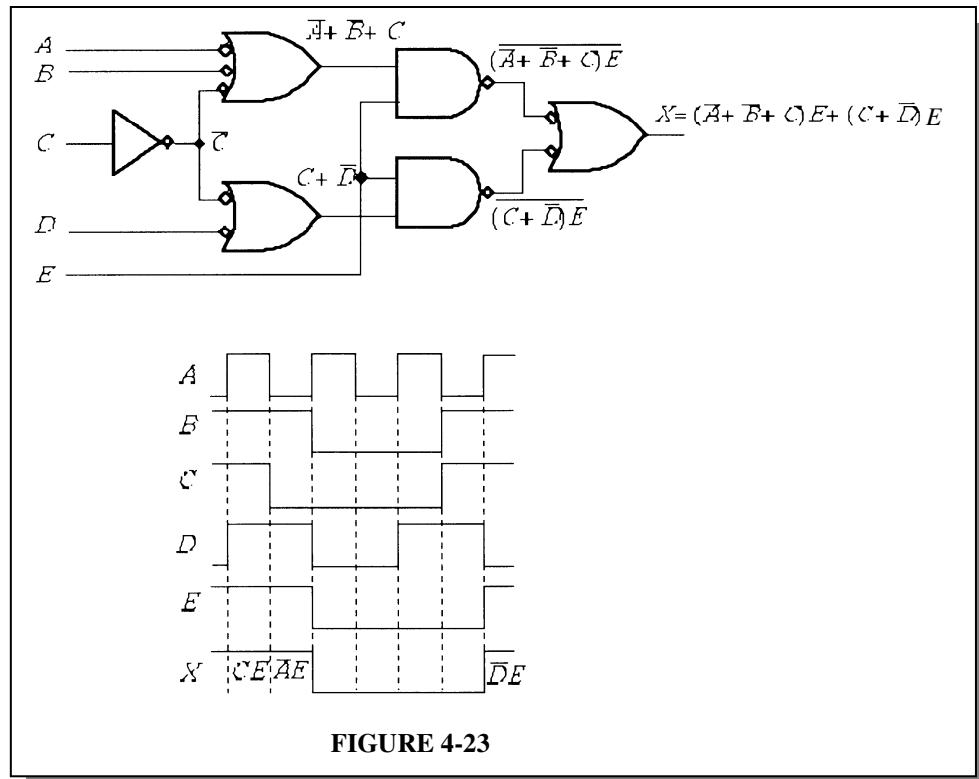


FIGURE 4-23

$$(b) \quad X = E + E(\bar{D} + C) = E(1 + \bar{D} + C) = E$$

Waveform X is the same as waveform E , in Figure 4-23. Since this is the correct waveform, the open output of gate G_3 does not show up for this *particular* set of input waveforms.

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

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.

Chapter 4

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

The output of the $\overline{\overline{CD}}$ gate is *stuck LOW*. See Figure 4-24.

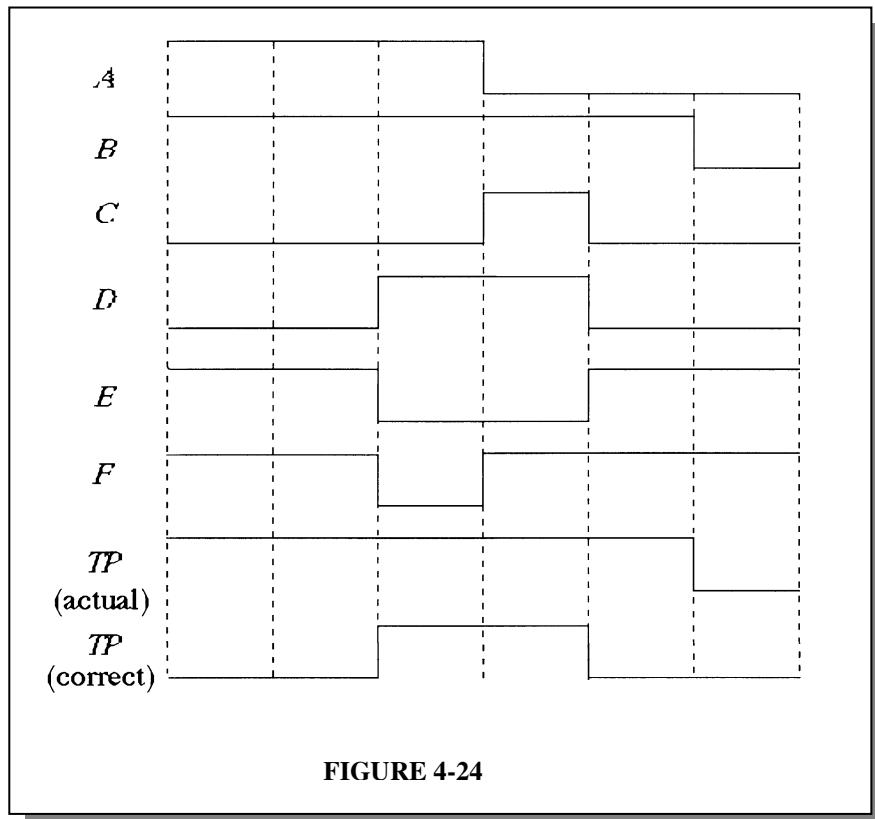


FIGURE 4-24

Special Problems

50.

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 4-25.

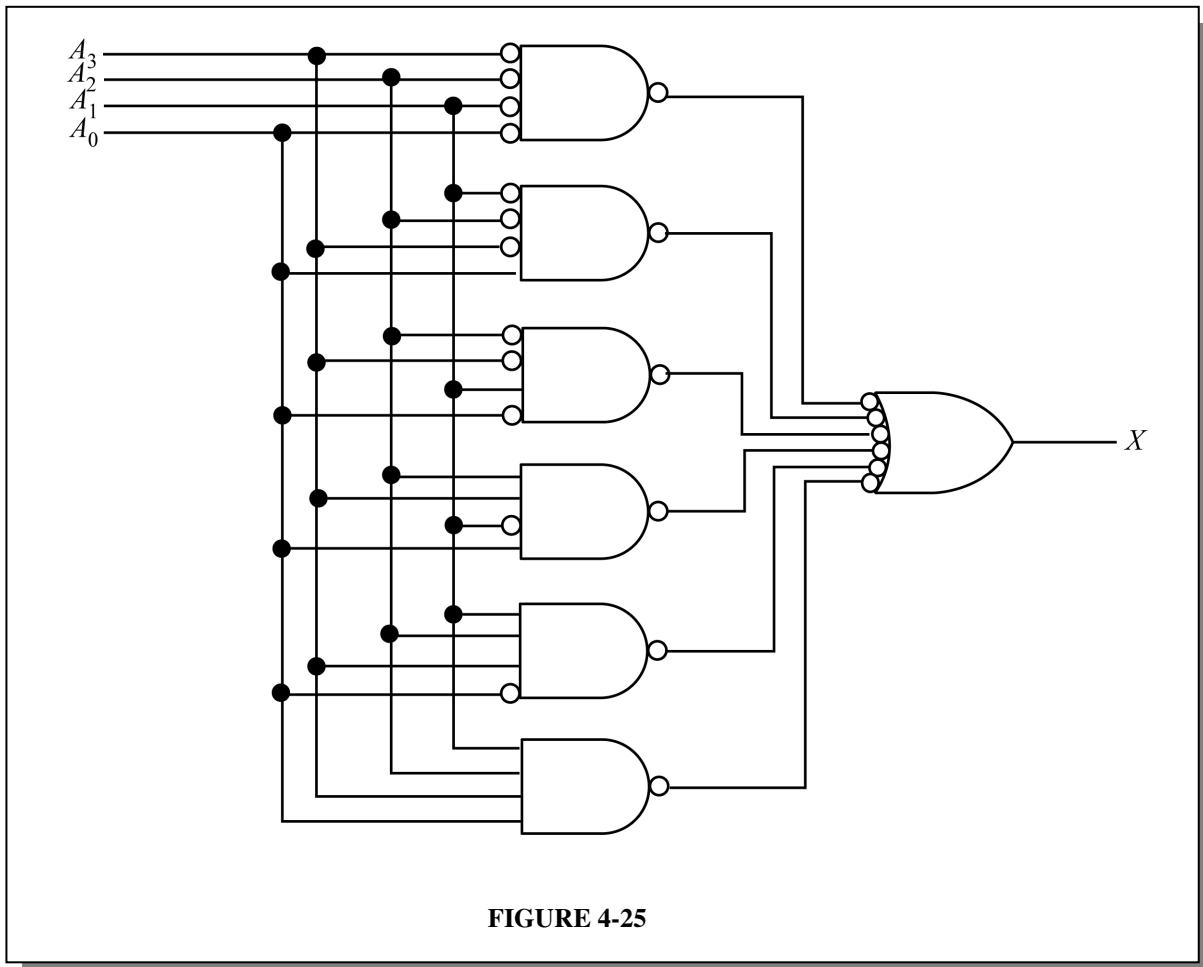


FIGURE 4-25

51. Let

X = Lamp on

A = Front door switch on

\bar{A} = Front door switch off

B = Back door switch on

\bar{B} = Back door switch off

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

See Figure 4-26.

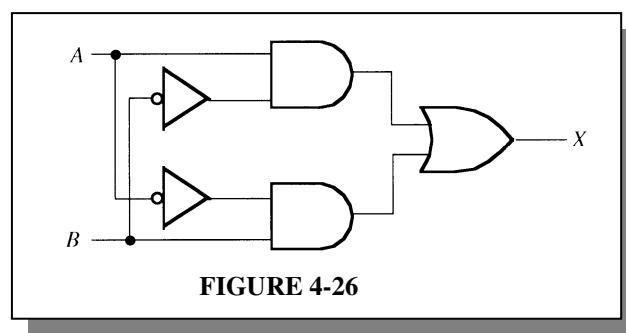


FIGURE 4-26

Chapter 4

52. See Figure 4-27.

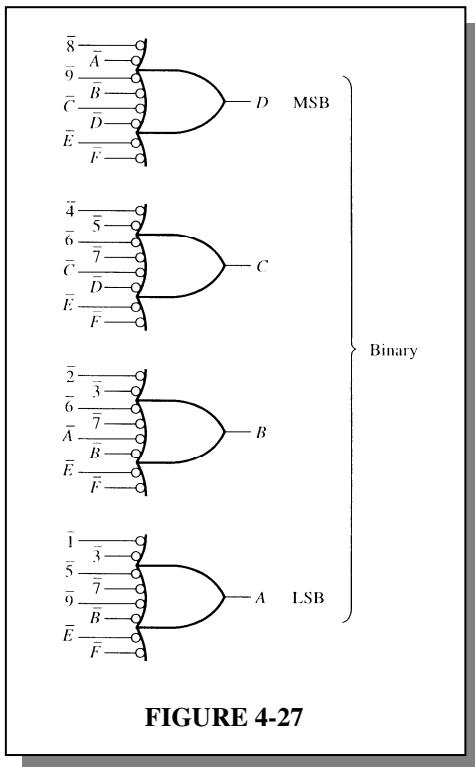


FIGURE 4-27

Multisim Troubleshooting Practice

53. **Circuit fault:** Input C is shorted to ground.

Predicted effect of fault: X1 and X2 are always off X3 is on only when both B and C are HIGH. X4 is on when either B or C is LOW.

Observed effect of introduced fault: X1 and X2 are always off. X3 is on only when both Band C are HIGH (switches Band C are both open).

54. **Circuit fault:** Bottom input U7 is open.

Predicted effect of fault: X1 and X4 are on when A is LOW and Band C are both HIGH. X2 and X4 are on when B is LOW and A and C are both HIGH. X3 is on when C is LOW and both A and B are HIGH.

Observed effect of introduced fault: X1 and X4 are on when A is LOW and Band C are both HIGH. X2 and X4 are on when B is LOW and A and C are both HIGH. X3 is on when C is LOW and both A and B are HIGH.

55. **Observed operation:** X1 is on when either A or B is LOW. X2 is always off. X3 is on when both A and B are HIGH or when C is LOW. X4 is always on. X3 is on when C is HIGH and either A or B is LOW.

Suspected fault: Output of U3 is shorted to ground.

Effect of introduced fault: X1 is on when either A or B is LOW. X2 is always off. X3 is on when both A and B are HIGH or when C is LOW. X4 is always on. X3 is on when C is HIGH and either A or B is LOW.

56. **Observed operation:** X1 is on when A is HIGH. X2 is on when C is LOW. X3 is on when A is HIGH or when both A and C are LOW. X4 is on when D is LOW or when both C and D are HIGH. X5 is on when C is HIGH and A is LOW or when D is HIGH and C is LOW.

Suspected fault: Lower input of U1 is shorted to VCC.

Effect of introduced fault: X1 is on when A is HIGH. X2 is on when C is LOW. X3 is on when A is HIGH or when both A and C are LOW. X4 is on when D is LOW or when both C and D are HIGH. X5 is on when C is HIGH and A is LOW or when D is HIGH and C is LOW.

CHAPTER 5

FUNCTIONS OF COMBINATIONAL LOGIC

Section 5-1 A System

1. Register A stores three BCD digits that represent the number of tablets per bottle. For 36 tablets, Register A stores the following BCD number after five or any number of bottles have been filled:

$$\mathbf{000000110110 = 036}$$

Register B stores the cumulative total number of tablets in binary code. After five bottles have been filled, 180 tablets have been bottled. Register B stores the following binary number representing the total tablets bottled:

$$\mathbf{10110100 = 180}$$

2. After six bottles, the output of the adder is a binary code representing $6 \times 36 = 216$:

$$\mathbf{11011000_2}$$

3. The output of the BCD to binary code converter is the binary code for 36:

$$\mathbf{00100100_2.}$$

4. The input to the BCD-to-segment decoder is the BCD code for 036. The active (1) outputs of the BCD-to-7 segment decoder are as follows:

MSD: **No active outputs** represent 0.

Middle digit: **Active outputs a, b, c, d, g** represent a 3.

LSD: **Active outputs c, d, e, f, g** represent a 6.

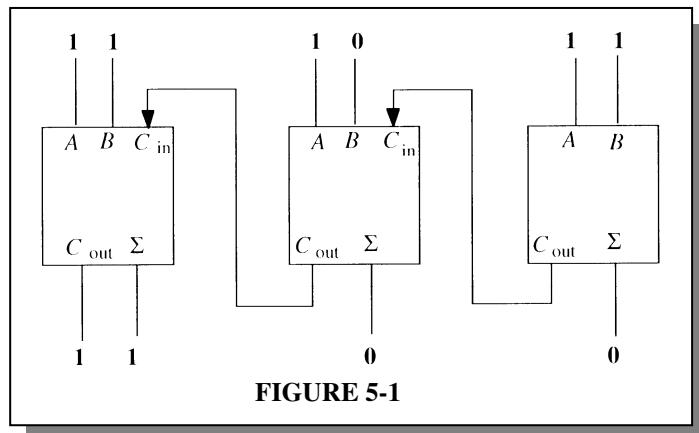
Section 5-2 Basic Adders

5. (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

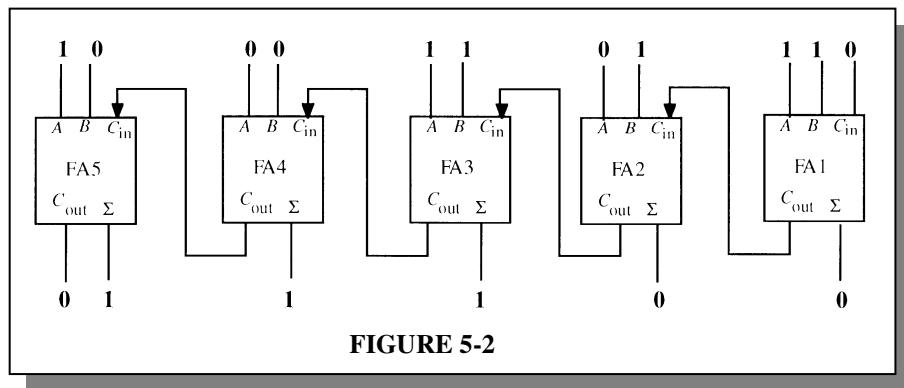
6. (a) $A = 0, B = 0, C_{in} = 0$
 (b) $A = 1, B = 0, C_{in} = 0$ or $A = 0, B = 1, C_{in} = 0$
 or $A = 0, B = 0, C_{in} = 1$
 (c) $A = 1, B = 1, C_{in} = 1$
 (d) $A = 1, B = 1, C_{in} = 0$ or $A = 0, B = 1, C_{in} = 1$
 or $A = 1, B = 0, C_{in} = 1$
7. (a) $\Sigma = 1, C_{out} = 0$ (b) $\Sigma = 1, C_{out} = 0$
 (c) $\Sigma = 0, C_{out} = 1$ (d) $\Sigma = 1, C_{out} = 1$

Section 5-3 Parallel Adders

8.
$$\begin{array}{r} 111 \\ 101 \\ \hline 1100 \end{array}$$
 See Figure 5-1.



9.
$$\begin{array}{r} 10101 \\ 00111 \\ \hline 11100 \end{array}$$
 See Figure 5-2.



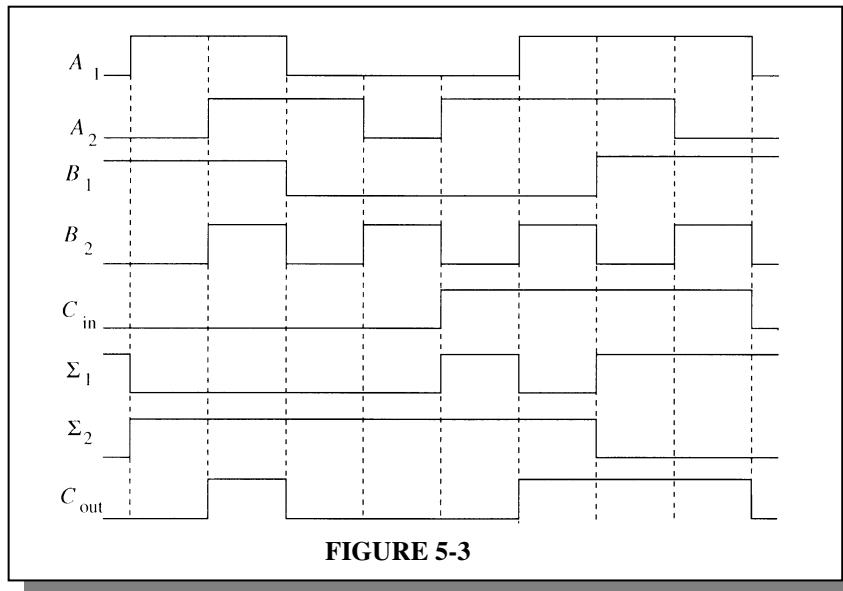
10. (a) When the $\overline{Add/Subt}$ is HIGH, the two numbers are subtracted.
 (b) When the input is LOW, the numbers are added.

Chapter 5

11. $A = 1001 = -7, B = 1100 = -4$

$\begin{array}{r} 1001 \\ 0011 \leftarrow \text{Complement of } B \\ \underline{1101} \leftarrow \text{LSB Carry input} \\ 1101 = -3 \text{ in 2's comp} \end{array}$

12. See Figure 5-3.



13.

A_4	A_3	A_2	A_1	B_4	B_3	B_2	B_1	Σ_5	Σ_4	Σ_3	Σ_2	Σ_1
1	0	0	1	0	0	0	1	0	1	0	1	0
1	0	1	0	1	1	0	1	0	0	1	1	1
0	0	1	0	0	0	1	1	0	0	1	0	1
1	0	1	1	0	1	1	1	1	0	0	1	0

$$\Sigma_1 = 0110$$

$$\Sigma_2 = 1011$$

$$\Sigma_3 = 0110$$

$$\Sigma_4 = 0001$$

$$\Sigma_5 = 1000$$

$$14. \quad \begin{array}{r} 1111 \\ 1001 \\ \hline 11000 \end{array}$$

Σ outputs should be $C_{\text{out}}\Sigma_4\Sigma_3\Sigma_2\Sigma_1 = 11000$.
The Σ_3 output is HIGH and should be LOW.

See Figure 5-4.

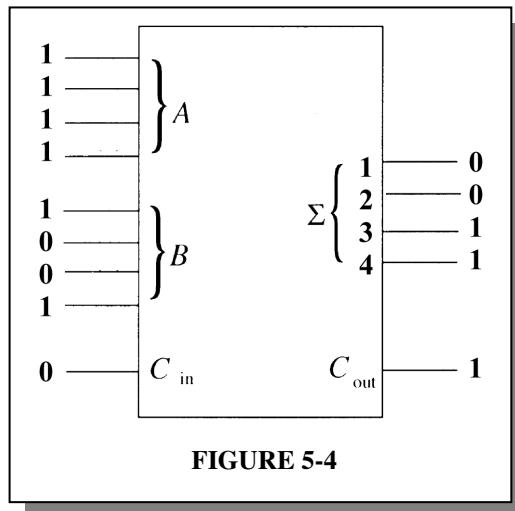


FIGURE 5-4

Section 5-4 Ripple Carry and Look-Ahead Carry Adders

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

16. Full-adder 5:

$$C_{\text{in}5} = C_{\text{out}4}$$

$$C_{\text{out}5} = C_{g5} + C_{p5}C_{g4} + C_{p5}C_{p4}C_{g3} + C_{p5}C_{p4}C_{p3}C_{g2} + C_{p5}C_{p4}C_{p3}C_{g2}C_{g1} + C_{p5}C_{p4}C_{p3}C_{p2}C_{p1}C_{\text{in}1}$$

The logic to be added to text Figure 5-21 is shown in Figure 5-5.

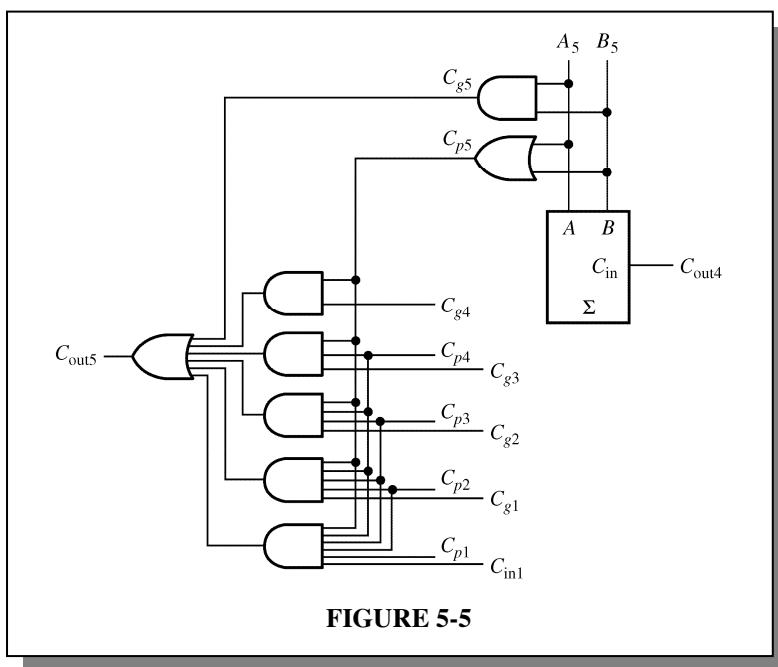


FIGURE 5-5

Chapter 5

Section 5-5 Comparators

17. The $A = B$ output is HIGH when $A_0 = B_0$ and $A_1 = B_1$.

See Figure 5-6.

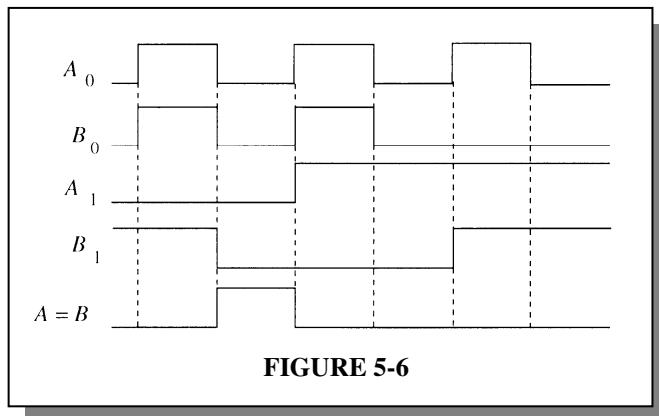


FIGURE 5-6

18. See Figure 5-7.

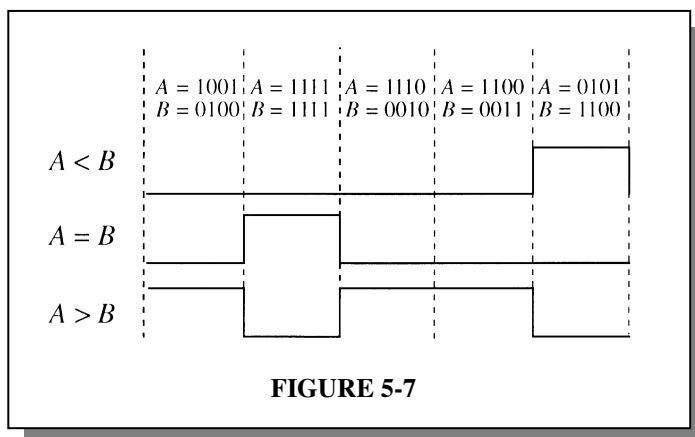


FIGURE 5-7

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

Section 5-6 Decoders

20. (a) $A_3A_2A_1A_0 = 1110$ (b) $A_3A_2A_1A_0 = 1100$
(c) $A_3A_2A_1A_0 = 1111$ (d) $A_3A_2A_1A_0 = 1000$

- 21.** See Figure 5-8.

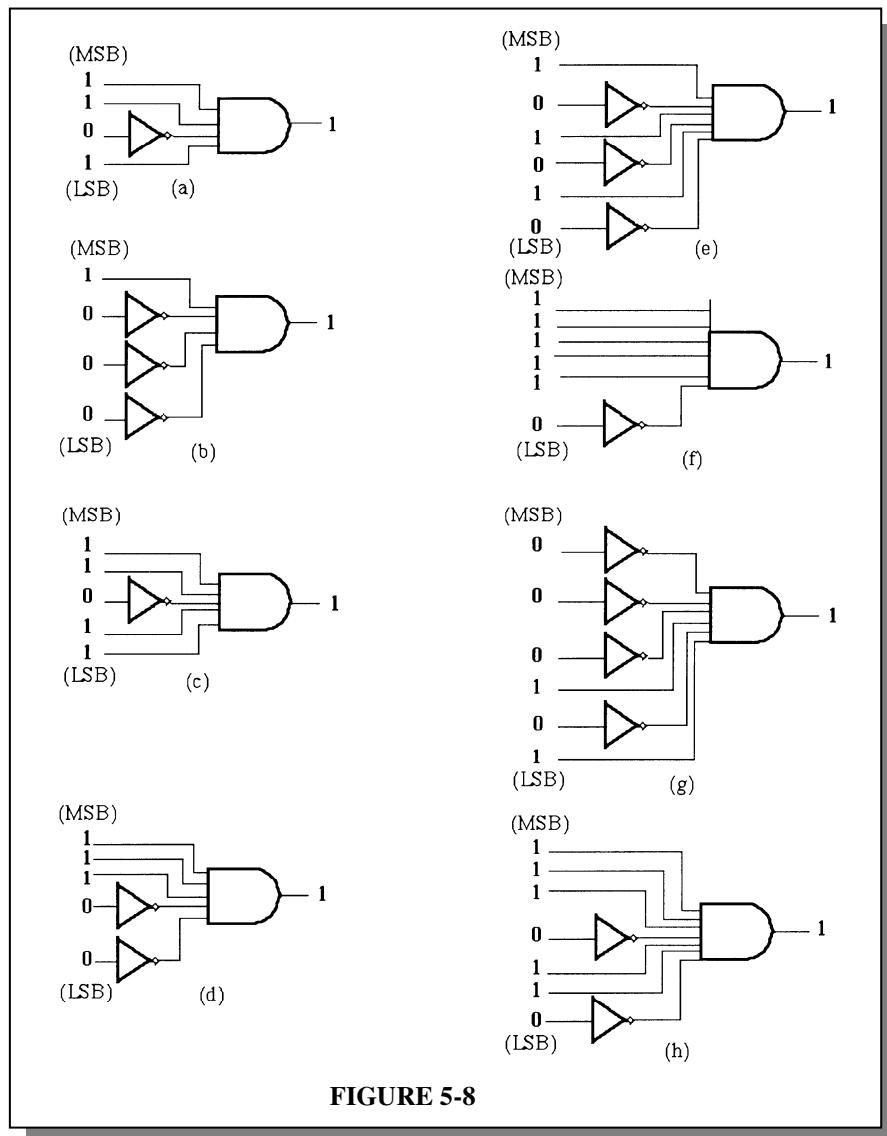


FIGURE 5-8

- 22.** Change the AND gates to NAND gates in Figure 5-8.

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

See Figure 5-9.

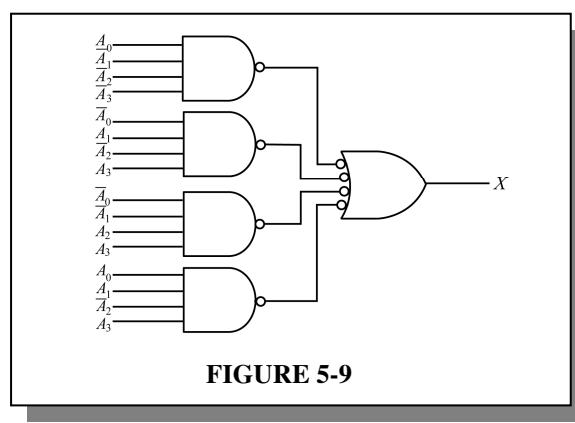


FIGURE 5-9

Chapter 5

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

See Figure 5-10.

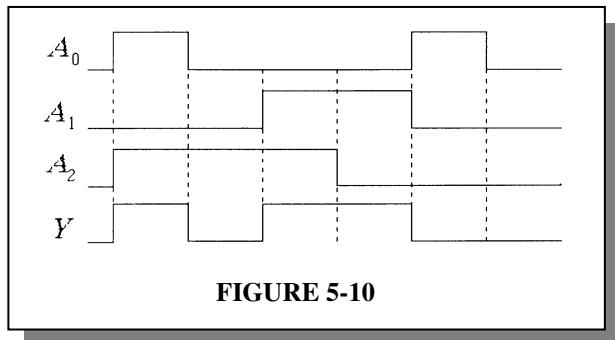


FIGURE 5-10

25. See Figure 5-11.

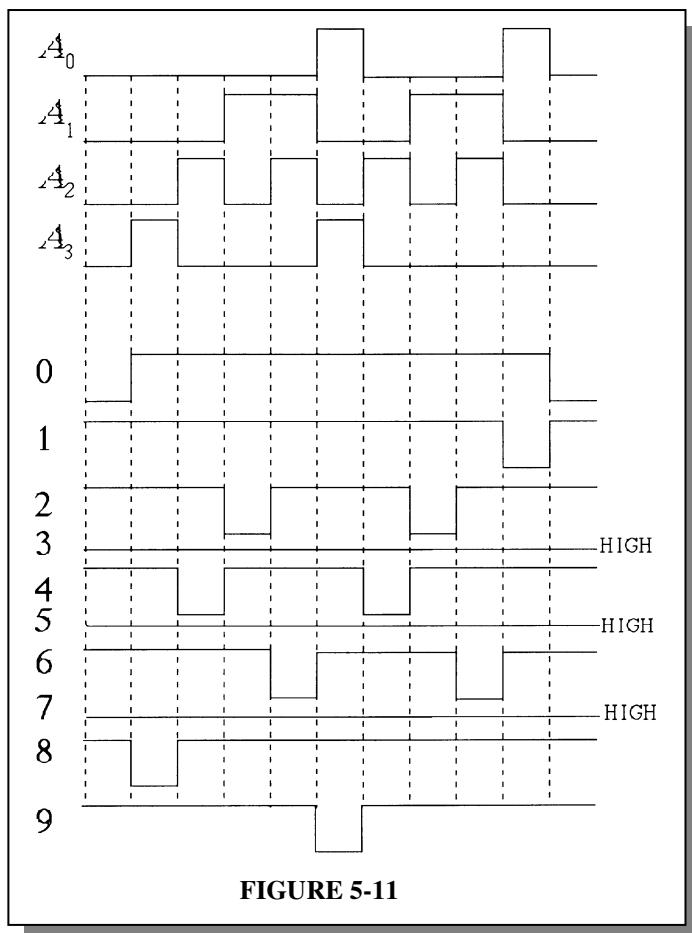


FIGURE 5-11

26. 0 1 6 9 4 4 4 8 0

Section 5-7 Encoders

27. A_0, A_1 , and A_3 are HIGH. $A_3A_2A_1A_0 = 1011$, which is an invalid BCD code.

28. The highest priority input is 9, so the BCD code is 1001.

Section 5-8 Code Converters

29. (a) $2_{10} = \mathbf{0010}_{\text{BCD}} = \mathbf{0010}_2$

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

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

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

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

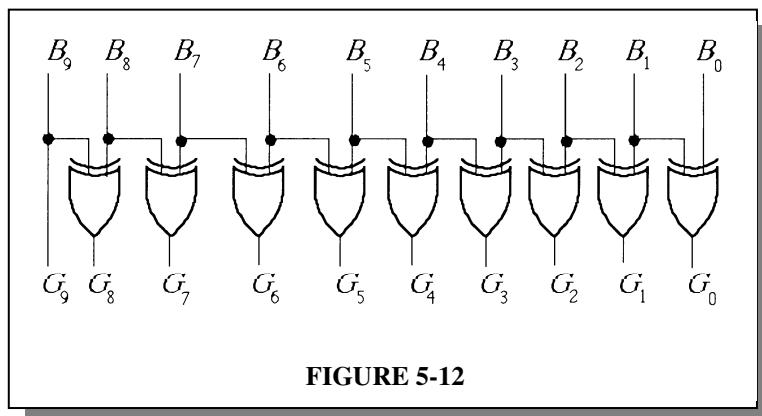
30. (a) 1010101010 binary
 1111111111 gray

(b) 1111100000 binary
 1000010000 gray

(c) 0000001110 binary
 0000001001 gray

(d) 1111111111 binary
 1000000000 gray

See Figure 5-12.



31. (a) 1010000000 gray
 1100000000 binary

(b) 0011001100 gray
 0010001000 binary

(c) 1111000111 gray
 1010000101 binary

(d) 0000000001 gray
 0000000001 binary

See Figure 5-13.

Chapter 5

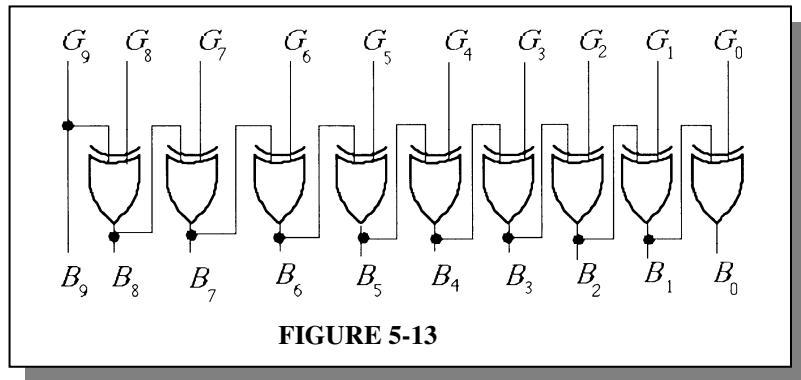


FIGURE 5-13

Section 5-9 Multiplexers (Data Selectors)

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

33. See Figure 5-14.

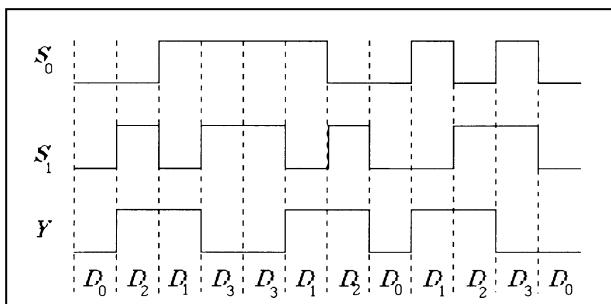


FIGURE 5-14

34. See Figure 5-15.

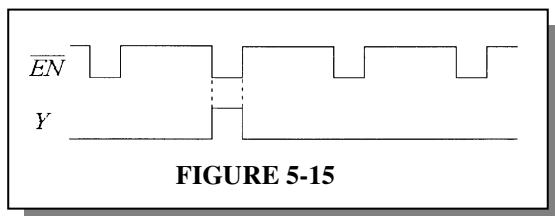
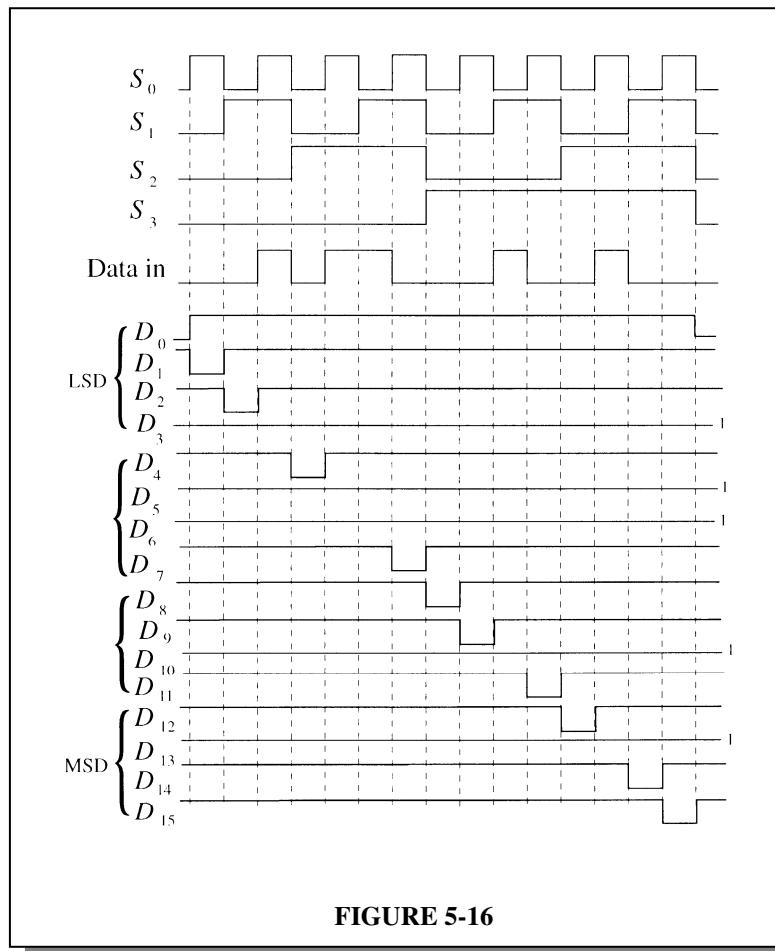


FIGURE 5-15

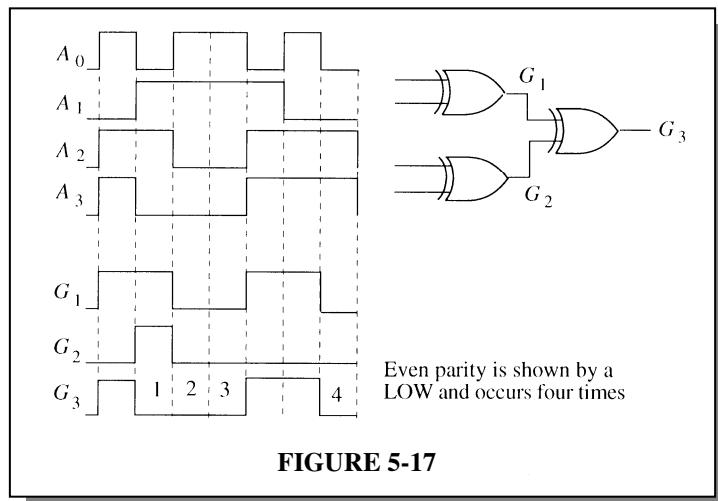
Section 5-10 Demultiplexers

35. See Figure 5-16.



Section 5-11 Parity Generators/Checkers

36. See Figure 5-17.



Chapter 5

33. See Figure 5-18.

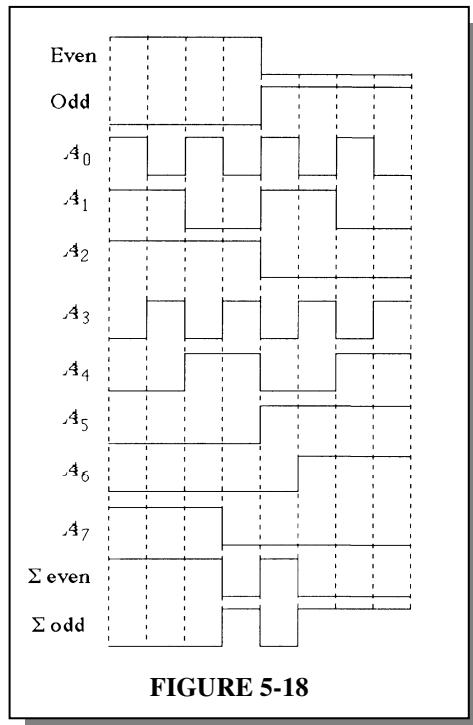


FIGURE 5-18

Section 5-12 Logic Functions with VHDL and Verilog

```

38. entity 8_Bit_Adder is
  port (A, B: in std_logic_vector (7 downto 0); Cin: in std_logic; SUM: out std_logic_vector
(7 downto 0);
      Cout: out std_logic);
end entity 8_Bit_Adder;

architecture Function of 8 Bit Adder is
signal C: std _logic vector (7 downto 1);
component FullAdder is
  port (A, B, Cin: in std_logic; SUM: out std_logic; Cout: out std_logic);
end component FullAdder;
begin
  FA1: FullAdder port map (A=>A(0), B => B(0), Cin => Cin, SUM => SUM(0), Cout =>C(1));
  FA2: FullAdder port map (A=>A(1), B => B(1), Cin => Cin, SUM => SUM(1), Cout =>C(2));
  FA3: FullAdder port map (A=>A(2), B => B(2), Cin => Cin, SUM => SUM(2), Cout =>C(3));
  FA4: FullAdder port map (A=>A(3), B => B(3), Cin => Cin, SUM => SUM(3), Cout =>C(4));
  FA5: FullAdder port map (A=>A(4), B => B(4), Cin => Cin, SUM => SUM(4), Cout =>C(5));
  FA6: FullAdder port map (A=>A(5), B => B(5), Cin => Cin, SUM => SUM(5), Cout =>C(6));
  FA7: FullAdder port map (A=>A(6), B => B(6), Cin => Cin, SUM => SUM(6), Cout =>C(7));
  FA8: FullAdder port map (A=>A(7), B => B(7), Cin => Cin, SUM => SUM(7), Cout =>Cout);
end architecture Function;

```

- ```

39. module 8_Bit_Adder (A, B, C, Cin, SUM, Cout);
 input [7:0] A;
 input [7:0] B;
 input Cin;
 output [7:0] SUM;
 output [7:0] Cout;
 wire C1, C2, C3, C4, C5, C6, C7;
 Full Adder Full Adder 1 (A[0], B[0], Cin, SUM[0], C1);
 Full Adder Full Adder 2 (A[1], B[1], Cin, SUM[1], C2);
 Full Adder Full Adder 3 (A[2], B[2], Cin, SUM[2], C3);
 Full Adder Full Adder 4 (A[3], B[3], Cin, SUM[3], C4);
 Full Adder Full Adder 5 (A[4], B[4], Cin, SUM[4], C5);
 Full Adder Full Adder 6 (A[5], B[5], Cin, SUM[5], C6);
 Full Adder Full Adder 7 (A[6], B[6], Cin, SUM[6], C7);
 Full Adder Full Adder 8 (A[7], B[7], Cin, SUM[7], Cout);
 endmodule 8_Bit_Adder

40. entity Encoder is
 port (D: in bit_vector (0 to 9); X: out bit_vector (0 to 3));
end entity Encoder;
architecture Function of Encoder is
begin
 X(0) <= (D(1) or D(3) or D(5) or D(7) or D(9)) and not (D(0) or D(2) or D(4) or D(6) or D(8));
 X(1) <= (D(2) or D(3) or D(6) or D(7)) and not (D(0) or D(1) or D(4) or D(5) or D(8) or D(9));
 X(2) <= (D(4) or D(5) or D(6) or D(7)) and not (D(0) or D(1) or D(2) or D(3) or D(8) or D(9));
 X(3) <= (D(8) or D(9)) and not (D(0) or D(1) or D(2) or D(3) or D(4) or D(5) or D(6) or D(7));
end architecture Function;

41. module Encoder (D, X);
 input [9:0] D; output [3:0] X;
 X[0] = (D[1] || D[3] || D[5] || D[7] || D[9]) && !(D[0] || D[2] || D[4] || D[6] || D[8]);
 X[1] = (D[2] || D[3] || D[6] || D[7]) && !(D[0] || D[1] || D[4] || D[5] || D[8] || D[9]);
 X[2] = (D[4] || D[5] || D[6] || D[7]) && !(D[0] || D[1] || D[2] || D[3] || D[8] || D[9]);
 X[3] = (D[8] || D[9]) && !(D[0] || D[1] || D[2] || D[3] || D[4] || D[5] || D[6] || D[7]);
 endmodule Encoder

```

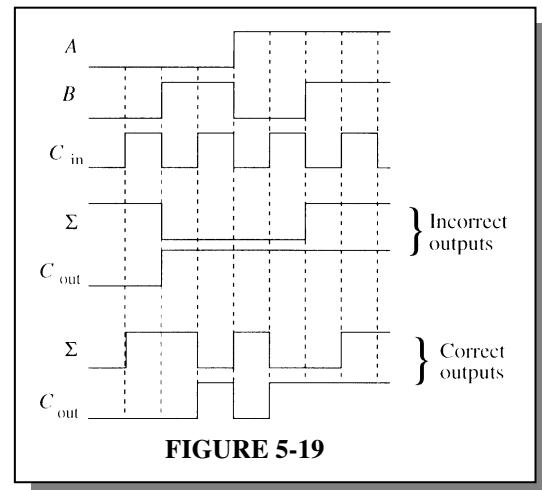
### **Section 5-13 Troubleshooting**

42. 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  $V_{CC}$ .

## Chapter 5

See Figure 5-19 for the correct output waveforms.



**FIGURE 5-19**

43. (a) OK (b) Segment g burned out; output G open (c) Segment b output stuck LOW

44. Step 1: Verify that the supply voltage is applied.

Step 2: Go through the key sequence and verify the output code in Table 5-1.

**TABLE 5-1**

| Key  | $A_3$ | $A_2$ | $A_1$ | $A_0$ |
|------|-------|-------|-------|-------|
| None | 1     | 1     | 1     | 1     |
| 0    | 1     | 1     | 1     | 1     |
| 1    | 1     | 1     | 1     | 0     |
| 2    | 1     | 1     | 0     | 1     |
| 3    | 1     | 1     | 0     | 0     |
| 4    | 1     | 0     | 1     | 1     |
| 5    | 1     | 0     | 1     | 0     |
| 6    | 1     | 0     | 0     | 1     |
| 7    | 1     | 0     | 0     | 0     |
| 8    | 0     | 1     | 1     | 1     |
| 9    | 0     | 1     | 1     | 0     |

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 5-2.

**TABLE 5-2**

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

- 45.** (a) Open  $A_1$  input acts as a HIGH. All binary values corresponding to a BCD number having a 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.
- 46.** (a) The 1Y1 output of the decoder is *stuck HIGH* or *open*; B cathode open.  
 (b) No power; EN input to the decoder is *open*.  
 (c) The  $f$  output of the BCD/7-seg decoder is *stuck HIGH*.  
 (d) The frequency of the data select input is too *low*.
- 47.** 1. Place a LOW on the Enable input.  
 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 5-3.

**TABLE 5-3**

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

4. Repeat the binary sequence of select inputs for each set of data inputs listed in Table 5-4. A HIGH on the  $Y$  output should occur only for the corresponding combinations of select inputs shown.

**TABLE 5-4**

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

## Chapter 5

48. The  $\Sigma$  EVEN output of the parity generator 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 parity generator *stuck LOW*.
  2. Error gate faulty.
  3. The ODD input to the parity generator is *open* thus acting as a HIGH.
  4. The inverter going to the ODD input of the parity generator has an *open* output or the output is *stuck HIGH*.
49. 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 5-20.

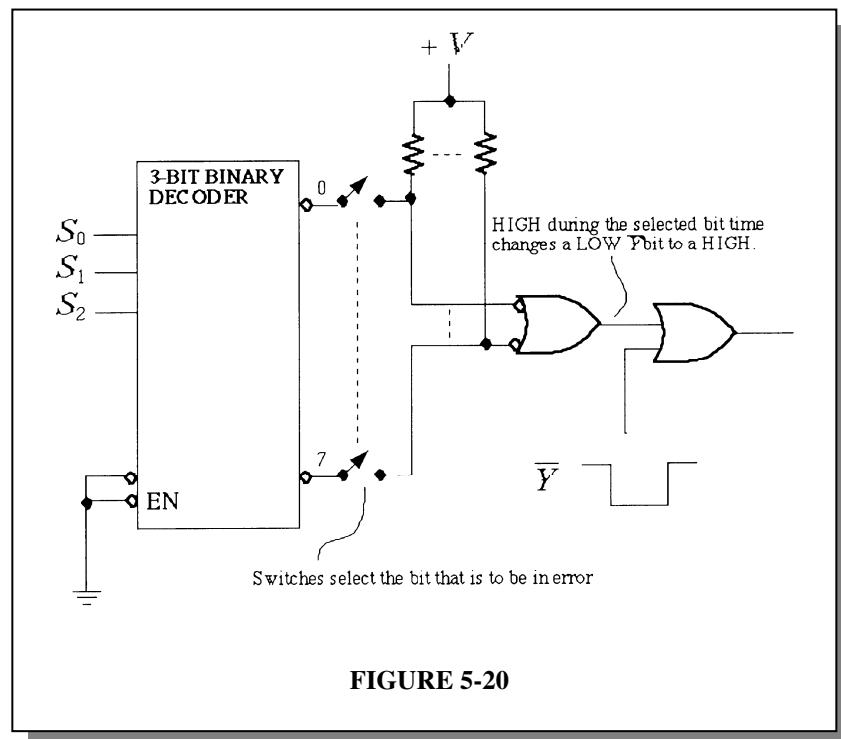
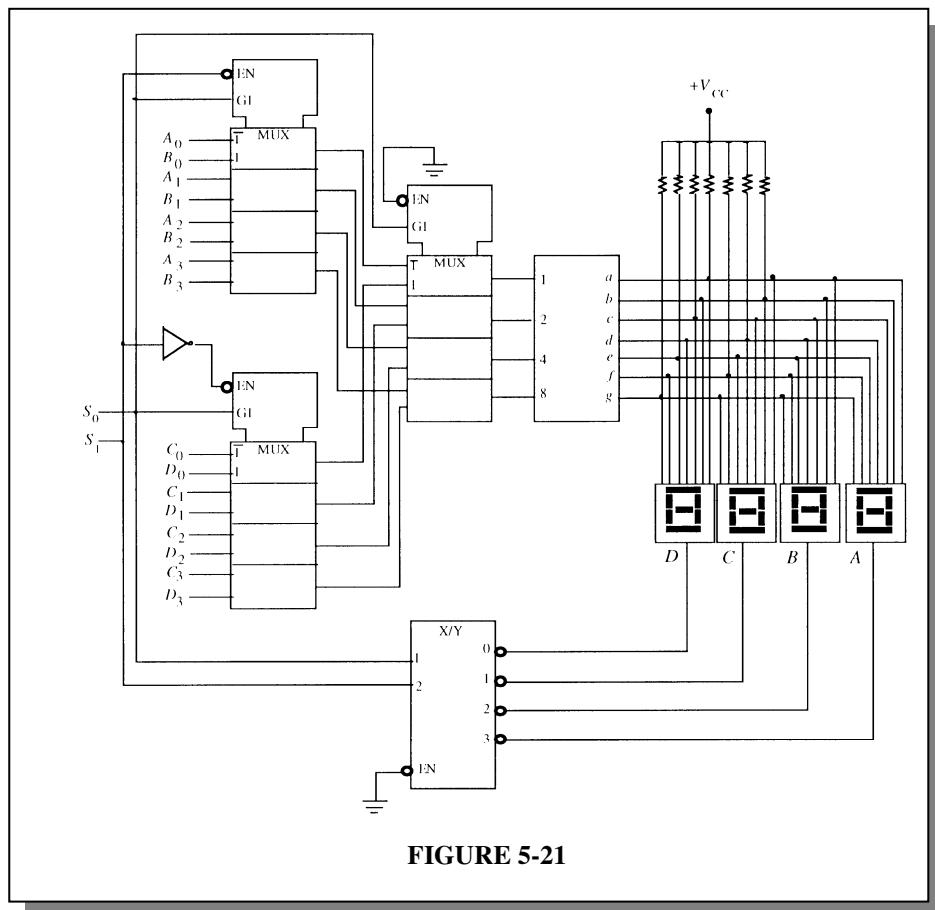


FIGURE 5-20

*Special Problems*

50. See Figure 5-21.



**FIGURE 5-21**

## Chapter 5

51.  $\Sigma = \overline{ABC}_{in} + \overline{AB}\overline{C}_{in} + A\overline{B}\overline{C}_{in} + ABC_{in}$   
 $C_{out} = ABC_{in} + \overline{ABC}_{in} + A\overline{B}\overline{C}_{in} + A\overline{B}\overline{C}_{in}$

See Figure 5-22.

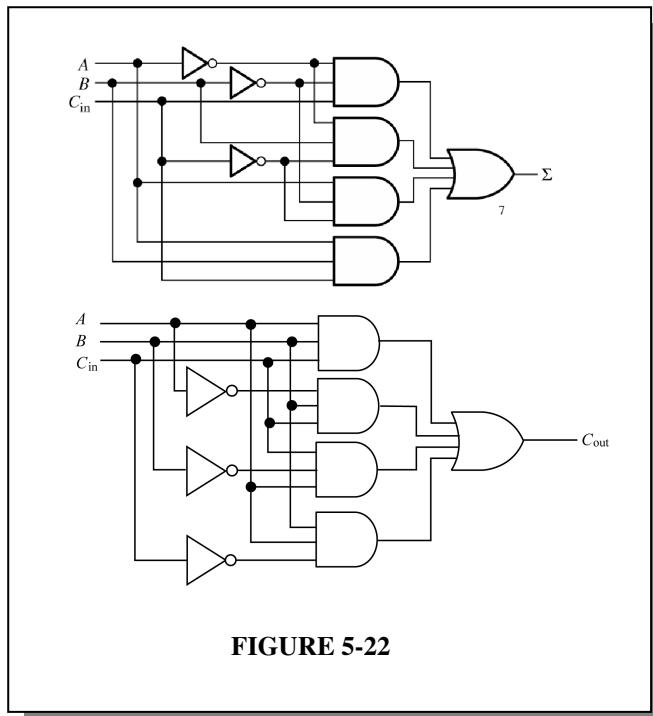


FIGURE 5-22

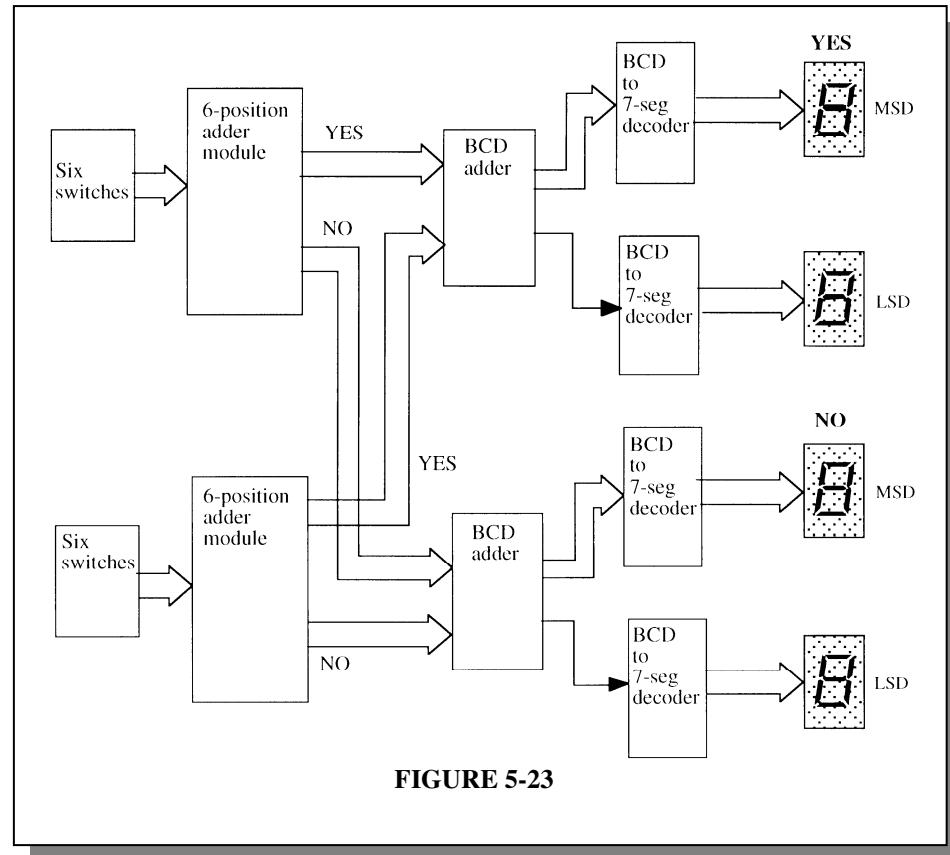
52. 
$$Y = \overline{A_3}\overline{A_2}A_1\overline{A_0} + \overline{A_3}\overline{A_2}A_1A_0 + \overline{A_3}A_2A_1\overline{A_0} + \overline{A_3}A_2A_1A_0 + A_3\overline{A_2}\overline{A_1}\overline{A_0} \frac{n!}{r!(n-r)!}$$
  

$$+ A_3\overline{A_2}A_1\overline{A_0} + A_3\overline{A_2}A_1A_0 + A_3A_2\overline{A_1}\overline{A_0} + A_3A_2A_1A_0$$

Eight 4-input AND gates; one 8-input OR gate; four inverters.

```
entity ANDORLogic is
 port (A3, A2, A1, A0: in bit; Y out bit);
end entity ANDORLogic;
architecture Function of ANDORLogic is
begin
 Y <= (not A3 and not A2 and A1 and not A0) (not A3 and not A2 and A1 and A0) or
 (not A3 and A2 and A1 and not A0) or (not 3 and A3 and A1 and A0) or
 (A3 and not A2 and not A1 and not A0) or (A3 and not A2 and A1 and not A0) or
 (A3 and not A2 and A1 and A0) or (A3 and A2 and not A1 and A0) or
 (A3 and A2 and A1 and A0);
end architecture Function;
```

53. See Figure 5-23.



**FIGURE 5-23**

### *Multisim Troubleshooting Practice*

54. **Circuit fault:** Input CIN of U1 is shorted to VCC.

**Predicted effect of fault:** Displayed sum is always one greater than actual sum of inputs.

**Observed effect of introduced fault:** Displayed sum is always one greater than actual sum of inputs.

55. **Circuit fault:** Input B of 3-to-8 decoder is shorted to VCC.

**Predicted effect of fault:** Inputs are decoded as follows: 0000 = 2, 0001 = 3, 0010 = 2, 0011 = 3, 0100 = 6, 0101 = 7, 0110 = 6, 0111 = 7, 1000 and higher = No decoded output.

**Observed effect of introduced fault:** Inputs are decoded as follows: 0000 = 2, 0001 = 3, 0010 = 3, 0100 = 6, 0101 = 7, 0110 = 6, 0111 = 7, 1000 and higher = No decoded output.

## **Chapter 5**

- 56.** **Observed operation:** Inputs encoded as follows: 0 = 0010, 1 = 0011, 2 = 0010, 3 = 0011, 4 = 0110, 5 = 0111, 6 = 0110, 7 = 0111, 8 = 1010, 9 = 1011, A = 1010, B = 1011, C = 1110, D = 1111, E = 1110, F = 1111.

**Suspected fault:** Output A1 of U2 is shorted to ground.

**Effect of introduced fault:** Inputs encoded as follows: 0 = 0010, 1 = 0011, 2 = 0010, 3 = 0011, 4 = 0110, 5 = 0111, 6 = 0110, 7 = 0111, 8 = 1010, 9 = 1011, A = 1010, B = 1011, C = 1110, D = 1111, E = 1110, F = 1111.

- 57.** **Observed operation:** Data output X is incorrect for binary inputs of 0011, 0111, 1011, and 1111.

**Suspected fault:** Line from input S2 to multiplexer input B is shorted to ground.

**Effect of introduced fault:** Data output X is incorrect for binary inputs of 0011, 0111, 1011, and 1111.

---

## CHAPTER 6

### LATCHES, FLIP-FLOPS, and TIMERS

---

#### Section 6-1 A System

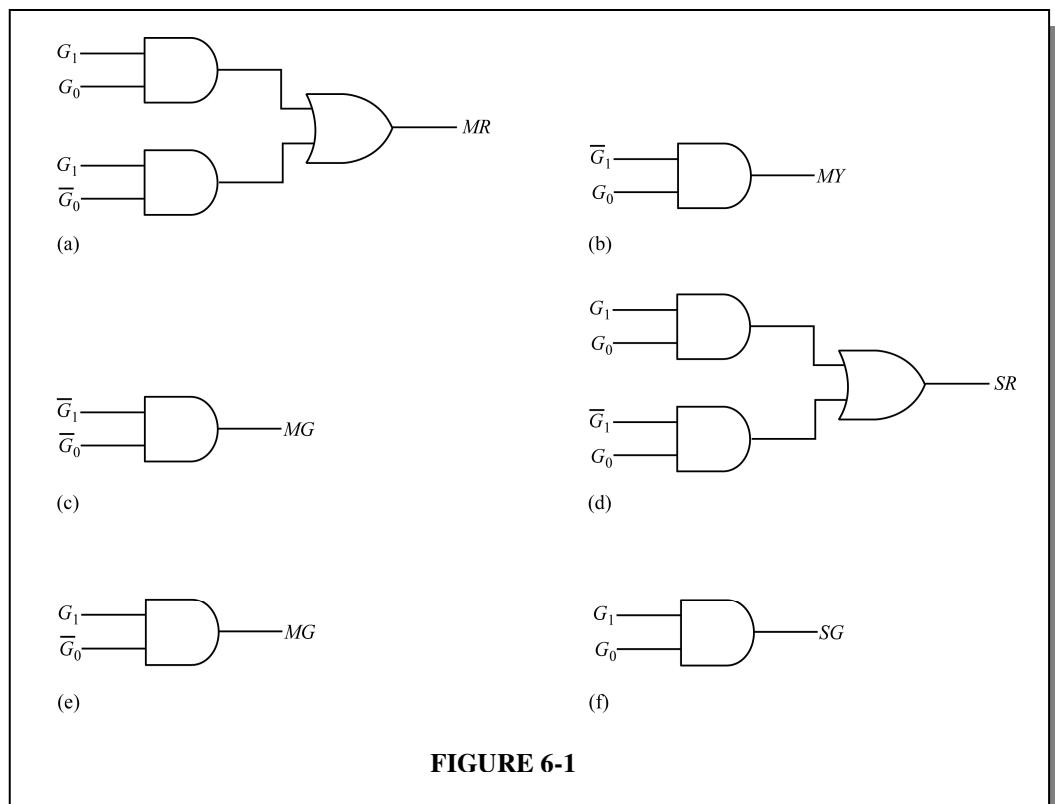
1. The system remains in the first state for 25 s or as long as there is not vehicle on the side street.
2. The system remains in the fourth state for 4 s.

3.  $MR = G_1 G_0 + G_1 \bar{G}_0$        $SR = \bar{G}_1 \bar{G}_0 + \bar{G}_1 G_0$

$$MY = \bar{G}_1 G_0 \quad SY = G_Y \bar{G}_0$$

$$MG = \bar{G}_1 \bar{G}_0 \quad SG = G_1 G_0$$

4. See Figure 6-1.



## *Chapter 6*

### *Section 6-2 Latches*

5. See Figure 6-2.

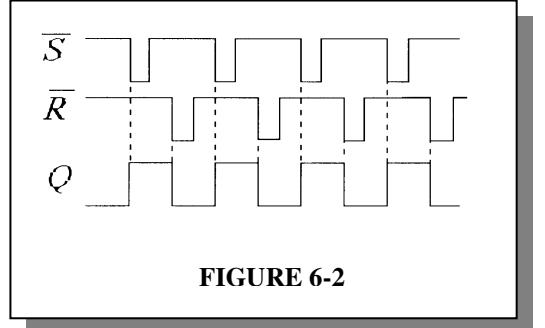


FIGURE 6-2

6. See Figure 6-3.

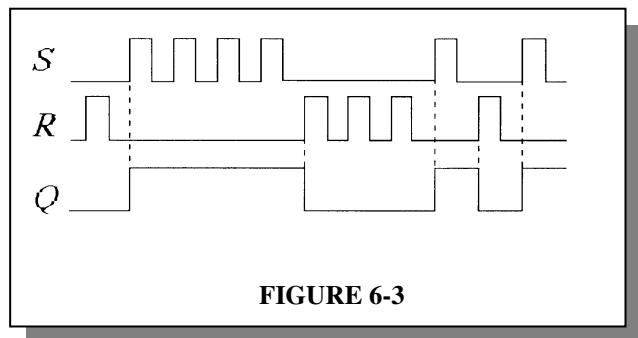


FIGURE 6-3

7. See Figure 6-4.

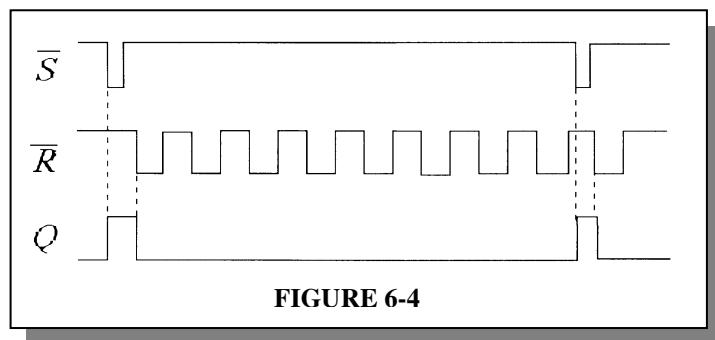
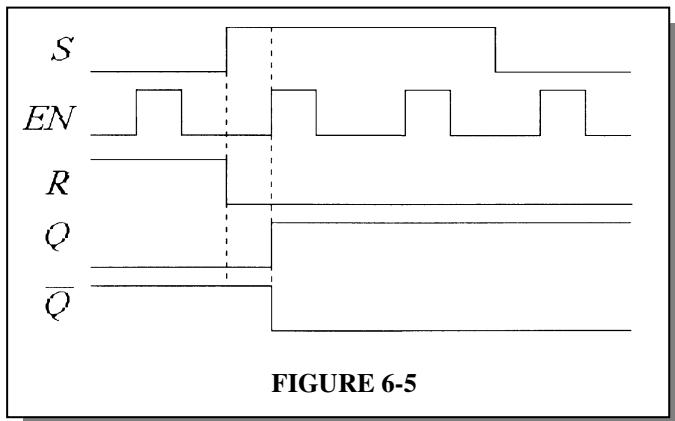


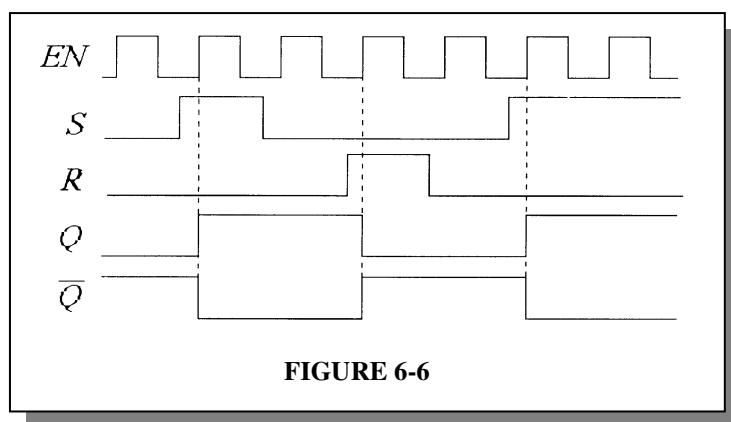
FIGURE 6-4

8. See Figure 6-5.



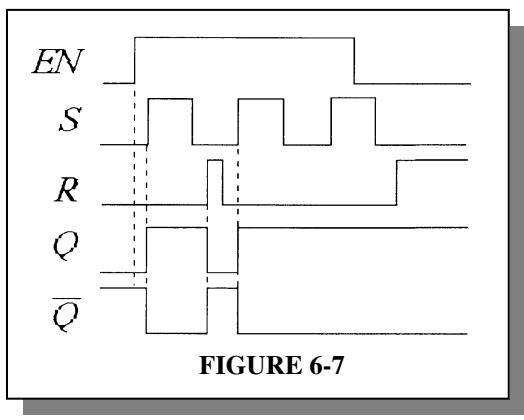
**FIGURE 6-5**

9. See Figure 6-6.



**FIGURE 6-6**

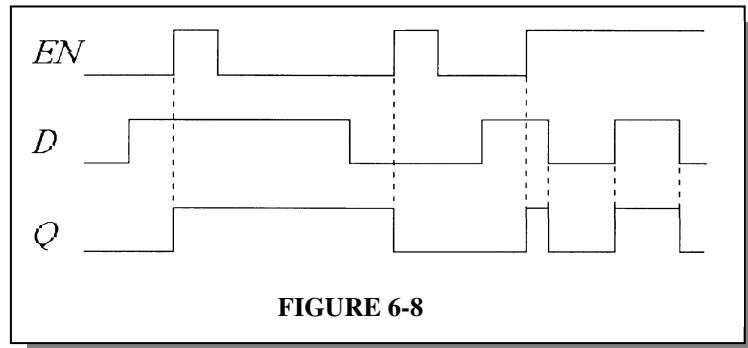
10. See Figure 6-7.



**FIGURE 6-7**

## *Chapter 6*

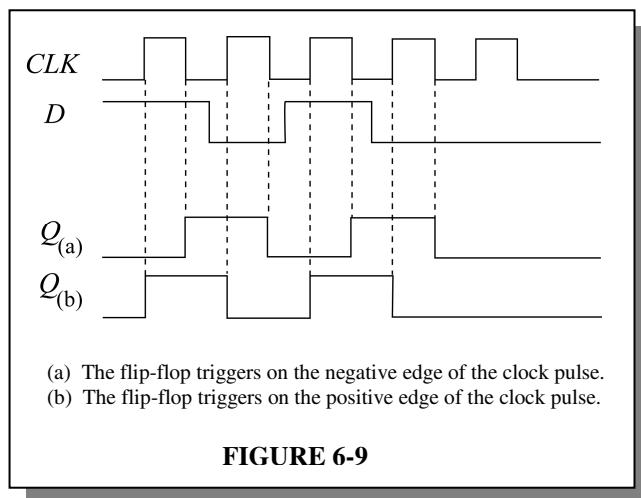
11. See Figure 6-8.



**FIGURE 6-8**

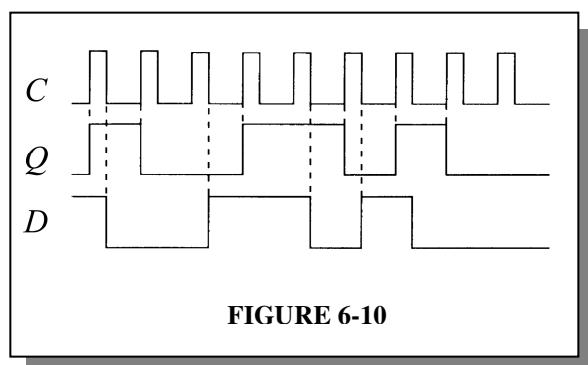
### *Section 6-3 Flip-Flops*

12. See Figure 6-9.



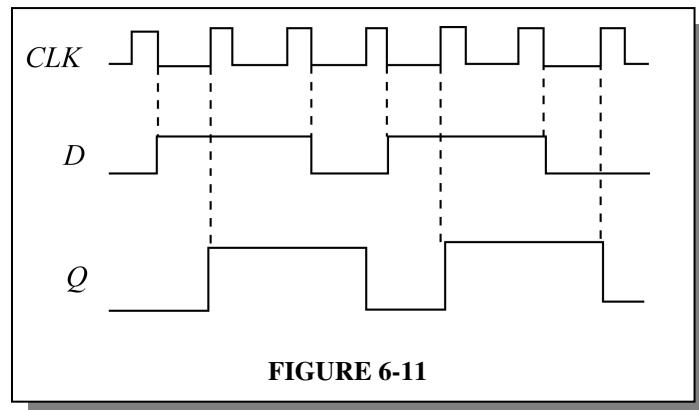
**FIGURE 6-9**

13. See Figure 6-10.



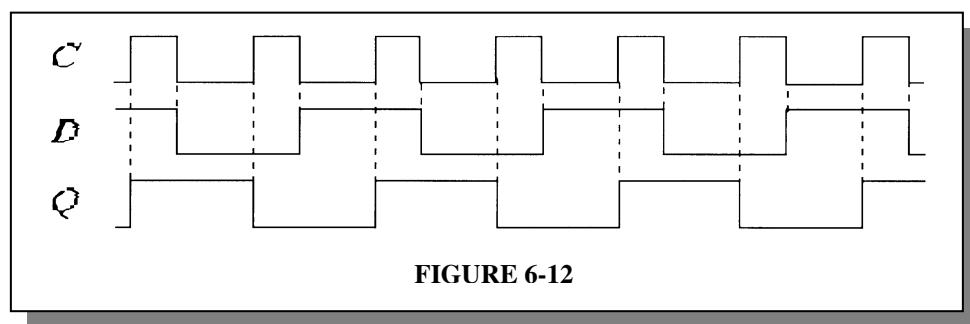
**FIGURE 6-10**

14. See Figure 6-11.



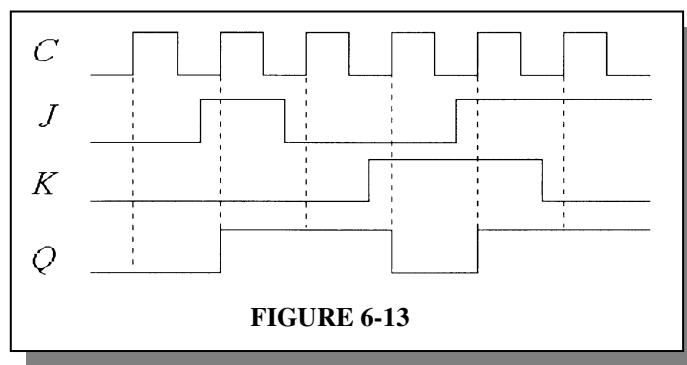
**FIGURE 6-11**

15. See Figure 6-12.



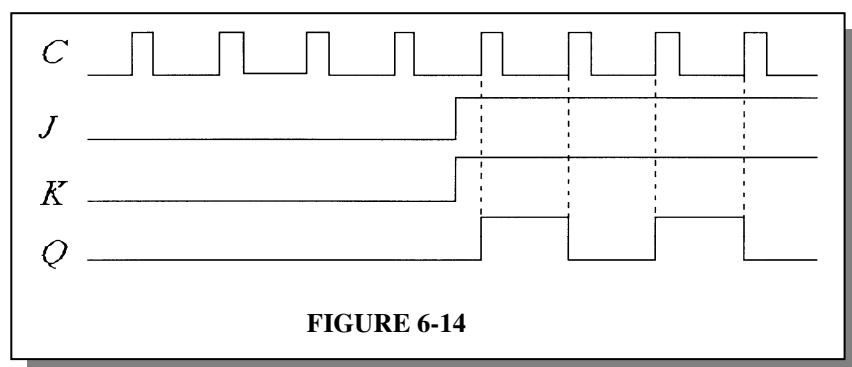
**FIGURE 6-12**

16. See Figure 6-13.



**FIGURE 6-13**

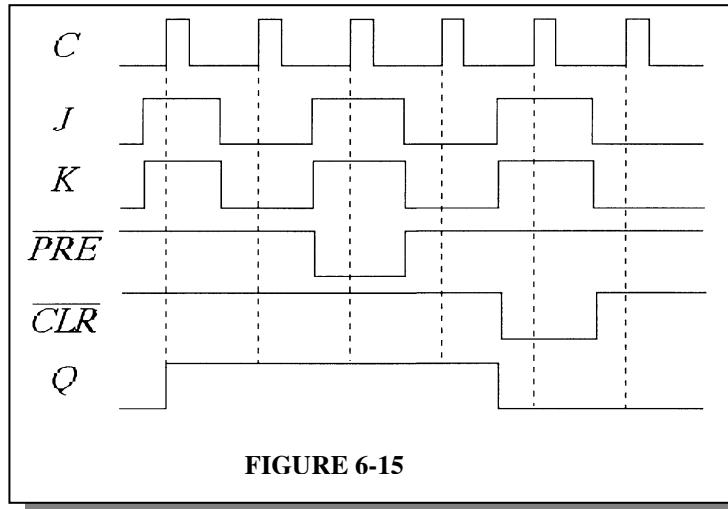
17. See Figure 6-14.



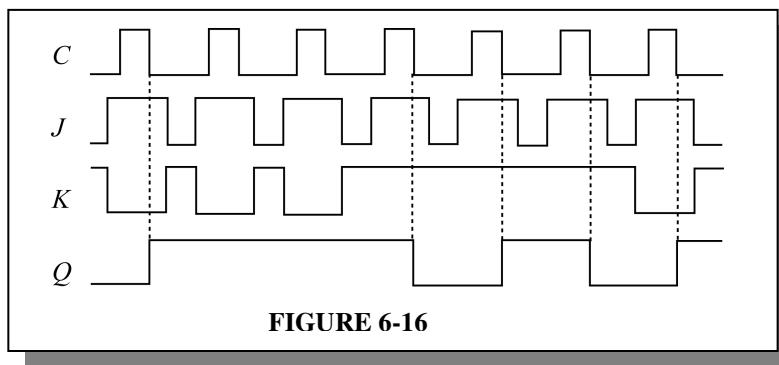
**FIGURE 6-14**

## Chapter 6

18. See Figure 6-15.



19. See Figure 6-16.

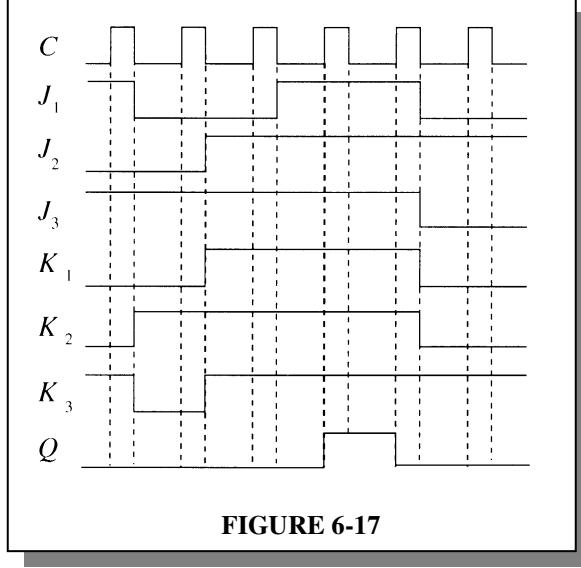


20.  $J: 0010000$

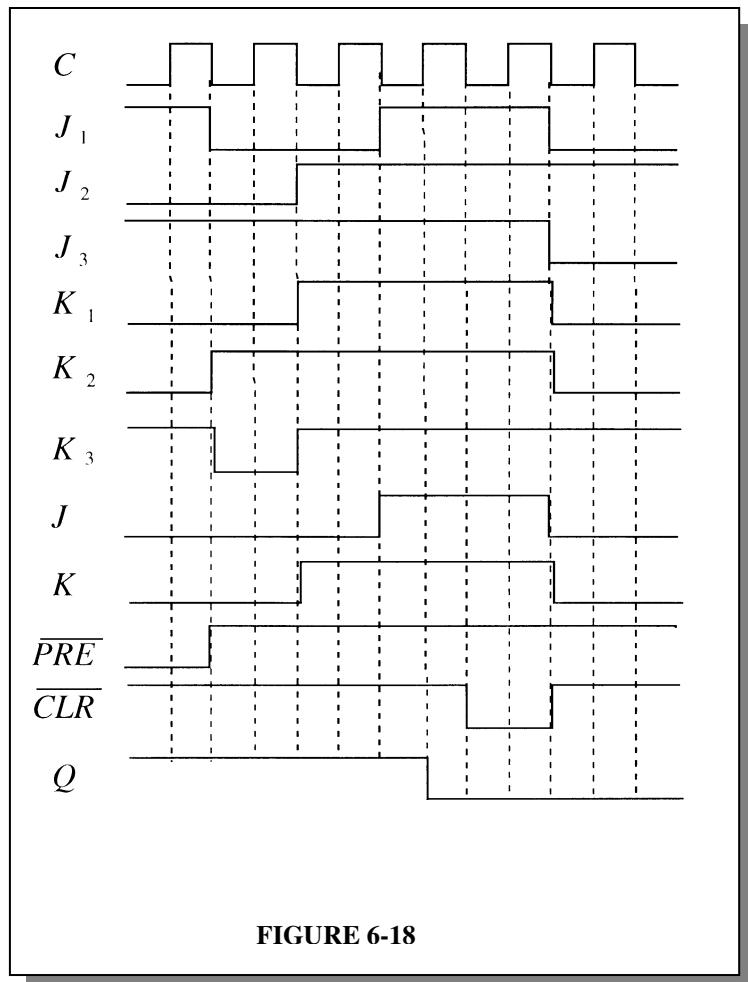
$K: 0000100$

$Q: 0011000$

21. See Figure 6-17.

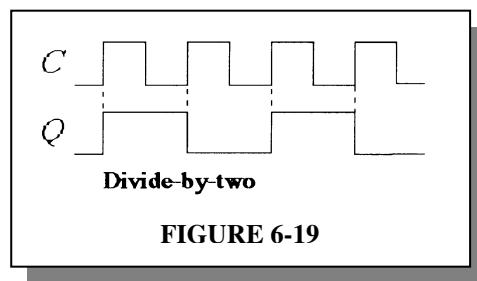


22. See Figure 6-18.



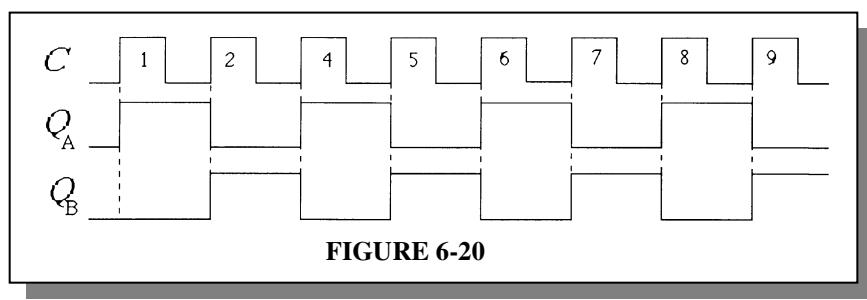
**FIGURE 6-18**

23. See Figure 6-19.



**FIGURE 6-19**

24. See Figure 6-20.



**FIGURE 6-20**

## Chapter 6

### Section 6-4 Flip-Flop Operating Characteristics

25. The direct current and dc supply voltage

26.  $t_{PLH}$  (Clock to  $Q$ ):

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.

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

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

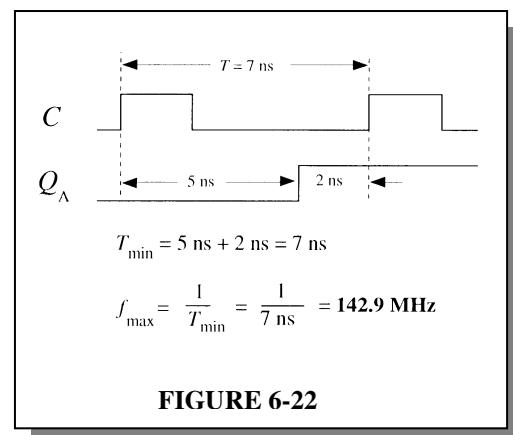
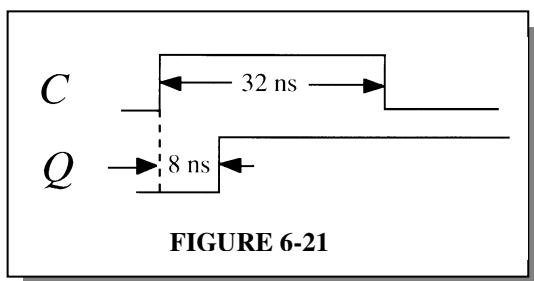
$t_{PHL}$  (CLR to  $Q$ ):

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

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

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

28. See Figure 6-21.



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

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

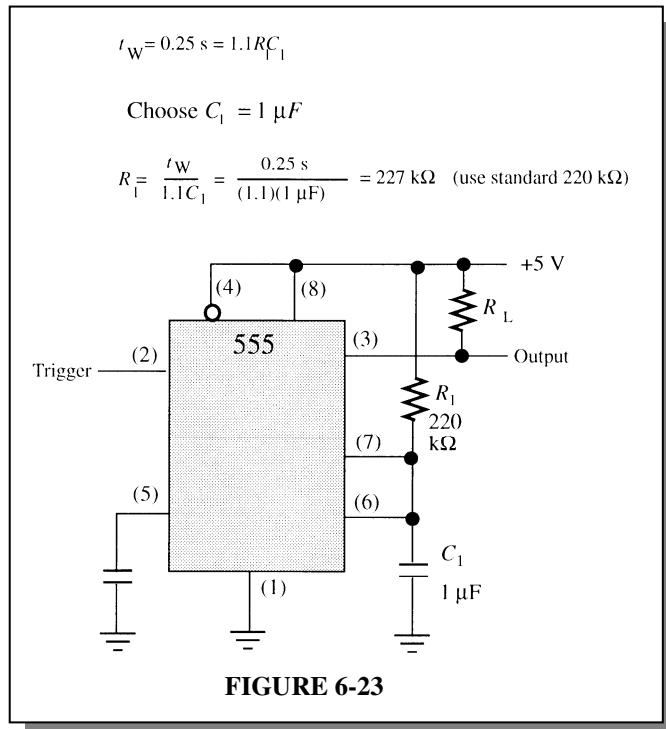
30. See Figure 6-22.

### Section 7-5 Timers

31.  $t_{PW} = 1.1R_X C_X = 1.1(3.3 \text{ k}\Omega)(2000 \text{ pF}) = 7.26 \mu\text{s}$

32.  $R_X = \frac{t_{PW}}{1.1(C_X)} = \frac{5 \mu\text{s}}{1.1(10,000 \text{ pF})} = 4.55 \Omega$

33. See Figure 6-23.



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

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

For a duty cycle of 75%:

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

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

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

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

### Section 6-6 Bistable Logic with VHDL and Verilog

36.  $Q \Leftarrow \text{not}(J1 \text{ and } \text{not}Q)$  is equivalent to  $Q \Leftarrow J1 \text{ nand } \text{not } Q$ ;
37. **signal** is an equivalent **reg** data type.
38. The **always block** code structure performs the functionality in the process block.
39. The keyword **assign** sets the result of a Boolean equation to an output identifier.

## **Chapter 6**

### **Section 6-7 Traffic Signal Control System with VHDL and Verilog**

**40.** The FreqDivide block produces a 1 Hz, 50% duty cycle output from the 24 MHz clock input

**41.** The VHDL statements that result in the 4 s and 25 s time intervals are

SetCountLong <=25;

SetCountShort <= 4;

**42.** The Verilog statements that result in the 4 s and 25 s time intervals are

Wire[8:0] SetCountLong = 25;

Wire[8:0]SetCountShort = 4;

**43.** The lines of code that describe the sequential logic are

VHDL: D1 <= (GO and not TS) or (G1 and TS);

D0 <= (not G1 and not Tl and VS) or (not G1 and G0) or (G0 and TL and VS);

Verilog: assign D1 = (G9 && !TS) || (G1 && TS);

assign D0 = (!G1 && !TS && VS) || (!G1 && G0) || (G0 && TL && VS);

**44.** SetCount is assigned the value 6,000,000.

**45.** Answers may vary.

(1) An additional Gray code bit G3 is added. Additionally, the Gray code input must recycle from count 110 back to 000, requiring a truncated sequence.

(2) Additional output identifier 55 is created.

(3) Combinational logic for S1, S2, S3, S4, and S5 are modified to incorporate the following truncated 3-bit gray code sequence:

State 1 (S1): G2 = 0, G1 = 0, G0 = 0

State 2 (S2): G2 = 0, G1 = 0, G0 = 1

State 3 (S3): G2 = 0, G1 = 1, G0 = 1

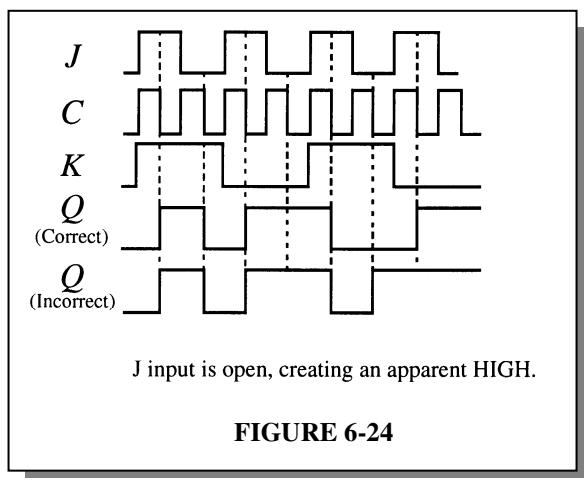
State 4 (S4): G2 = 0, G1 = 1, G0 = 0

State 5 (S5): G2 = 1, G1 = 1, G0 = 0

- 46.** If a 1 Hz clock replaces the 24 MHz system clock, the FreqDivide block can be eliminated and CLKin renamed Clock.
- 47.** Changes are
- (1) The Walk portion of the sign is simply wired to a HIGH.
  - (2) Two new output variables are added: DontMain and DontSide are applied to the Don't input of the sign.
  - (3) Output DontMain is assigned the logic statement for SG. Output DontSide is assigned the logic statement for MG.

### **Section 6-8 Troubleshooting**

- 48.** The flip-flop in Figure 6-89 of the text has an internally open *J* input.
- 49.** 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.
- 50.** See Figure 6-24.



- 51.** Since none of the flip-flops change, the problem must be a fault that affects all of them. The two functions common to all the flip-flops are the clock (CLK) and clear (CLR) inputs. One of these lines 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.

## Chapter 6

52. 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 6-25. The delays are exaggerated greatly for purposes of illustration. Glitches are eliminated by strobing the output with the clock pulse.

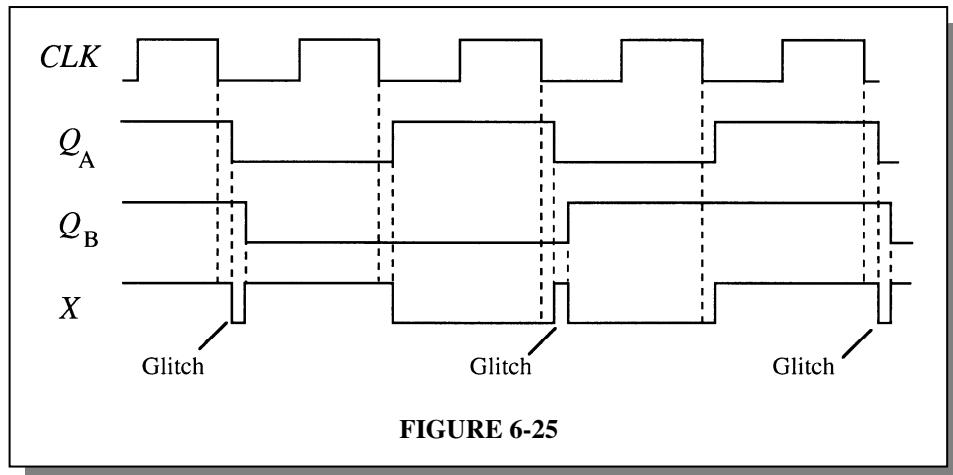


FIGURE 6-25

53. (a) See Figure 6-26.

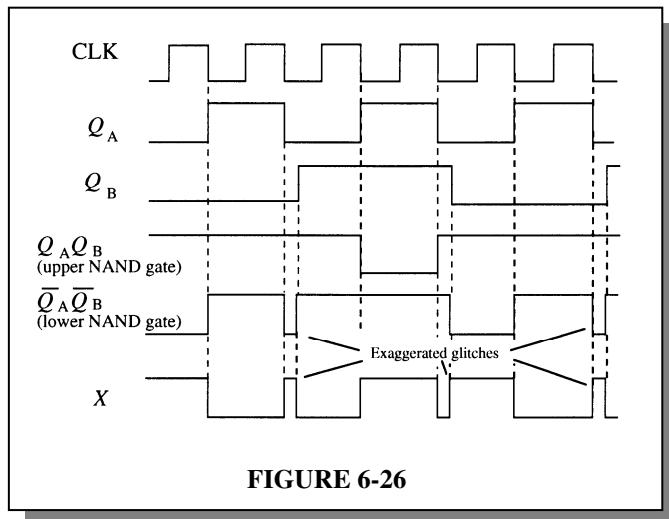
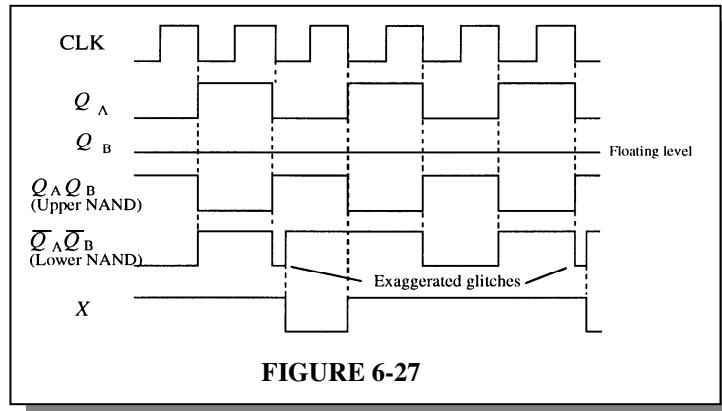


FIGURE 6-26

- (b)  $K_B$  open acts as a HIGH and the operation is normal. The timing diagram is the same as Figure 6-26.

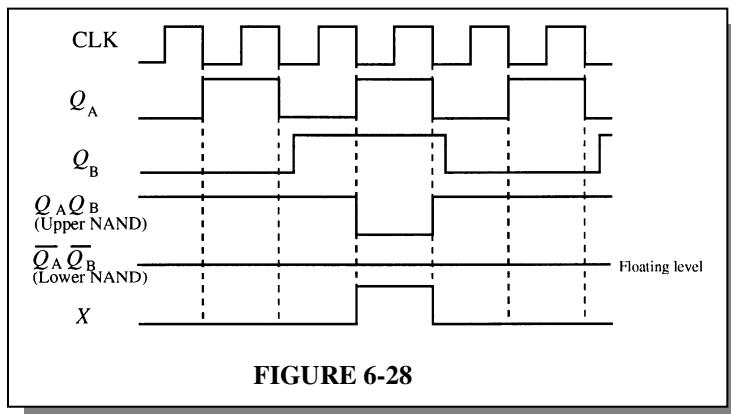
(c) See Figure 6-27.



**FIGURE 6-27**

(d)  $X$  remains LOW if  $Q_B = 1$  ( $\overline{Q_B} = 0$ ).  $X$  follows  $\overline{Q_A}$  if  $Q_B = 0$  ( $\overline{Q_B} = 1$ ).

(e) See Figure 6-28.



**FIGURE 6-28**

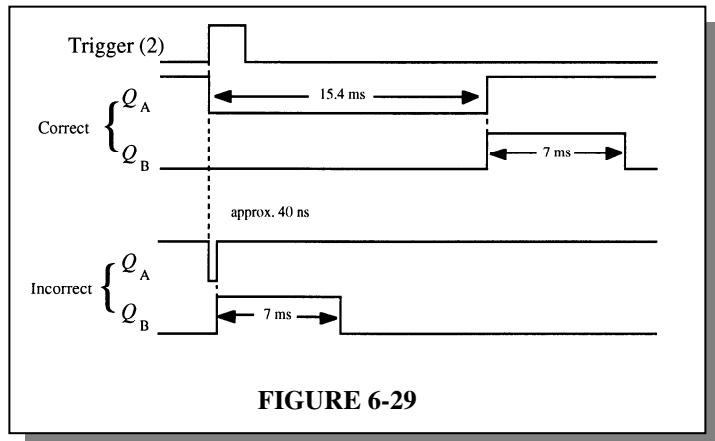
$$54. \quad t_{PW} = 0.7RC_{EXT}$$

$$\text{One-shot A: } t_{PW} = 0.7(0.22 \mu\text{F})(100 \text{ k}\Omega) = 15.4 \text{ ms}$$

$$\text{One-shot B: } t_{PW} = 0.7(0.1 \mu\text{F})(100 \text{ k}\Omega) = 7 \text{ ms}$$

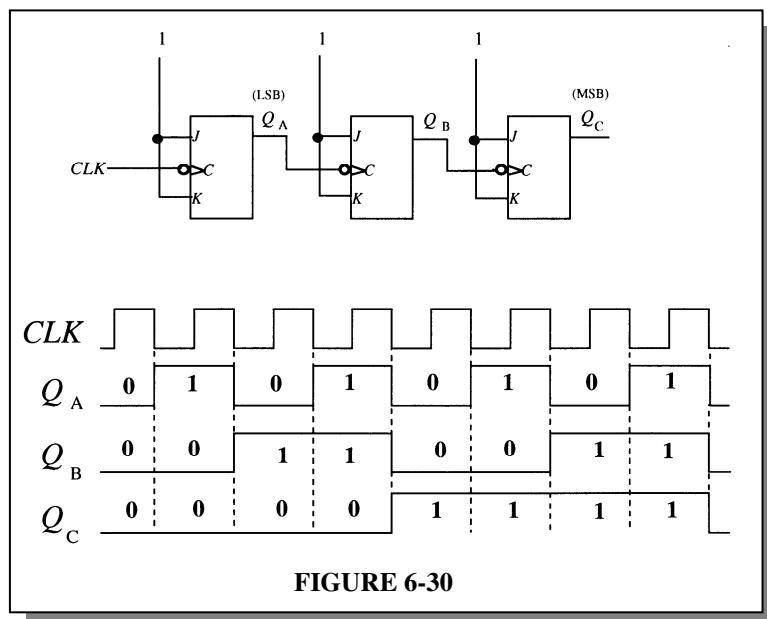
## Chapter 6

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 6-29.

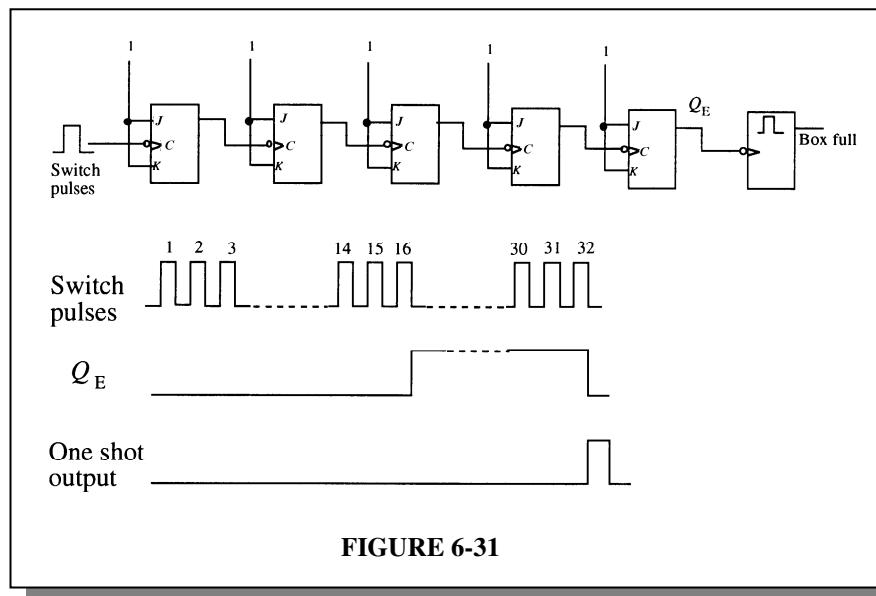


## Special Problems

55. See Figure 6-30.



56. See Figure 6-31 for one possibility.

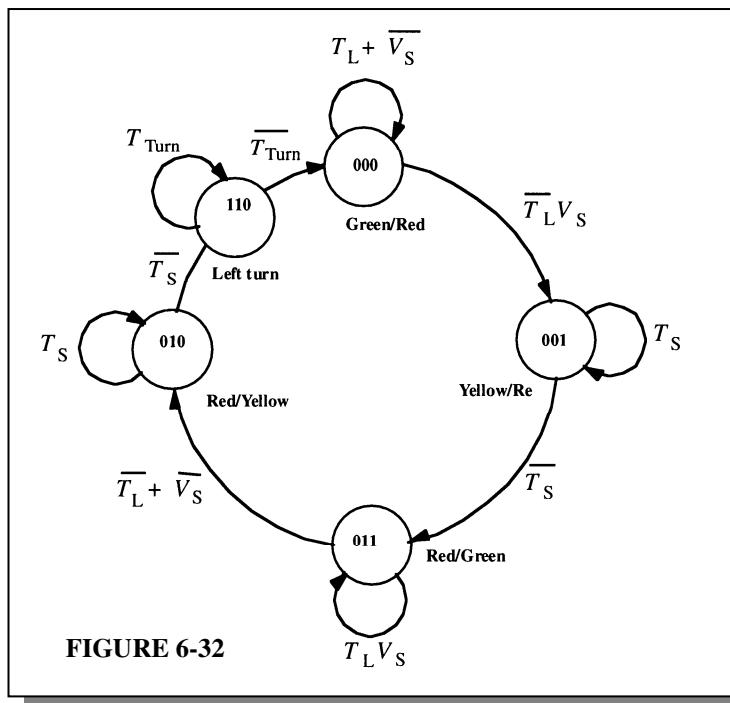


**FIGURE 6-31**

57. Changes required for the system to incorporate a 15 s left turn signal on main:

1. Change the 2-bit gray code sequence to a 3-bit sequence.
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 6-32.



**FIGURE 6-32**

## **Chapter 6**

### **Multisim Troubleshooting Practice**

- 58. Circuit fault:** R input of U1 is shorted to VCC.

**Predicted effect** of fault: The flip-flop will be reset and unable to set.

**Observed effect of introduced fault:** The flip-flop will be reset and unable to set.

- 59. Circuit fault:** Line to K input is shorted to VCC.

**Predicted effect** of fault: If line to J input is LOW, Q output will go and remain LOW and not Q output will go and remain HIGH. If line to J input is HIGH, Q and not Q will toggle.

**Observed effect of introduced fault:** If line to J input is LOW, Q output will go and remain LOW and not Q output will go and remain HIGH. If line to J input is HIGH Q and not Q will toggle.

- 60. Observed operation:** The SET and RESETbar inputs affect the Q and Qbar outputs, but the D input D0 does not affect the Q and Qbar outputs.

**Suspected fault:** Clock input is shorted to VCC or ground.

**Effect of introduced fault:** The SET and RESETbar inputs affect the Q and Qbar outputs, but the D input D0 does not affect the Q and Qbar outputs.

- 61. Observed operation:** Pulse width of one-shot is 690  $\mu$ s rather than 6.9  $\mu$ s.

**Suspected fault:** 1 k $\Omega$  resistor accidentally is used in place of 10 k $\Omega$  timing resistor, or a 0.1  $\mu$ F capacitor is used in place of the 1  $\mu$ F timing capacitor.

**Effect of introduced fault:** Pulse width of one-shot is 690  $\mu$ s rather than 6.9  $\mu$ s.

- 62. Observed operation:** The Q output of U2 toggles as expected but the Q output of U2 remains LOW.

**Suspected fault:** J input of U2 is shorted to ground.

**Effect of introduced fault:** The Q output of U2 toggles as expected but the Q output of U2 remains LOW.

---

## CHAPTER 7

### SHIFT REGISTERS

---

#### **Section 7-1 A System**

1. The BCD code sequence when the access number 4739 entered on the keypad is **1001001101110100**. The most significant digit (4) is entered first and appears as the right-most four bits in the code.
2. The initial state of register C is 00010000. After two correct digit entries, the state of Register C is **00000100**.
3. The states of shift registers A and C after two correct key entries are  
Register A: **0111**; Register C: **00000100**
4. When digit 4 is entered on the keypad, register A contains **0100**.
5. The register outputs are  $Q_0=1$ ,  $Q_1=0$ ,  $Q_2=0$ ,  $Q_3=0$ .
6. After three clock pulses the register states are  $Q_0=0$ ,  $Q_1=0$ ,  $Q_2=0$ ,  $Q_3=1$ .
7. The states of shift registers A and C after each key closure when entering 7645 are:

**After key 7 is pressed:**

Shift register A contains 0111  
Shift register C contains 11000

**After key 6 is pressed:**

Shift register A contains 0110  
Shift register C contains 11100

**After key 4 is pressed:**

Shift register A contains 0100  
Shift register C contains 11110

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

Shift register A contains 0000  
Shift register C contains 1000

8. Changes to the code-selection logic to increase the entry code to 5 digits are
  - (a) Additional stage must be added to the register.
  - (b) Four additional AND gates
  - (c) OR gates changed to 5-inputs.

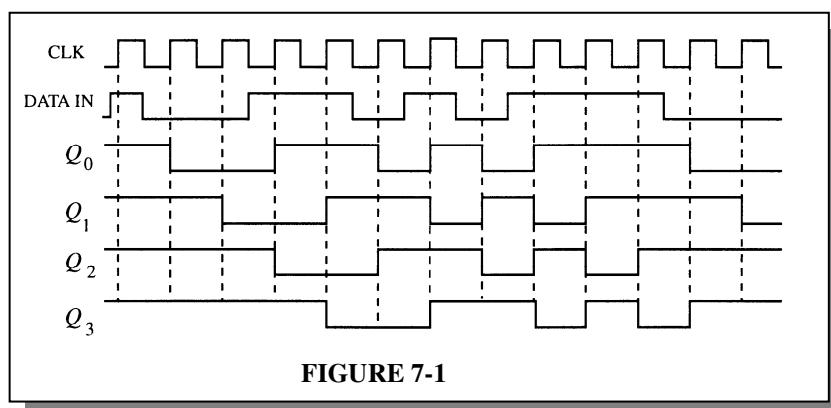
## *Chapter 7*

### *Section 7-2 Basic Shift Register Functions*

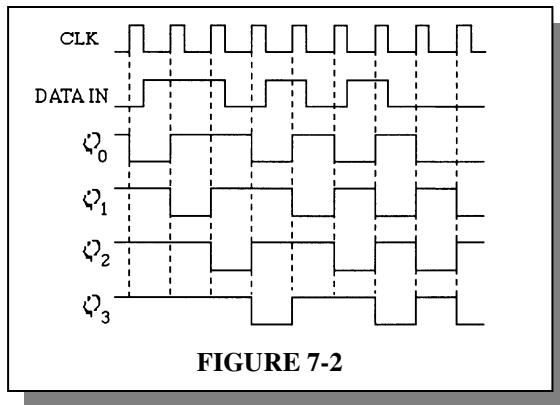
9. Shift registers store binary data in a series of flip-flops or other storage elements.
10. 1 byte = **8 bits**; 2 bytes = **16 bits**
11. Shift data and store data.

### *Section 7-3 Types of Shift Registers*

12. Initially: 0000  
1<sup>st</sup> CLK: 1000  
2<sup>nd</sup> CLK: 1100  
3<sup>rd</sup> CLK: 0110
13. See Figure 7-1.



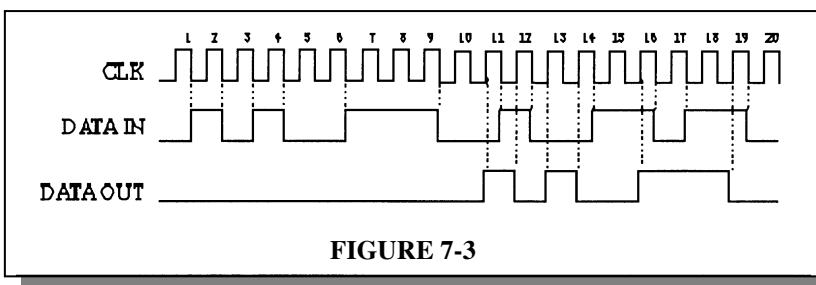
14. See Figure 7-2.



15.

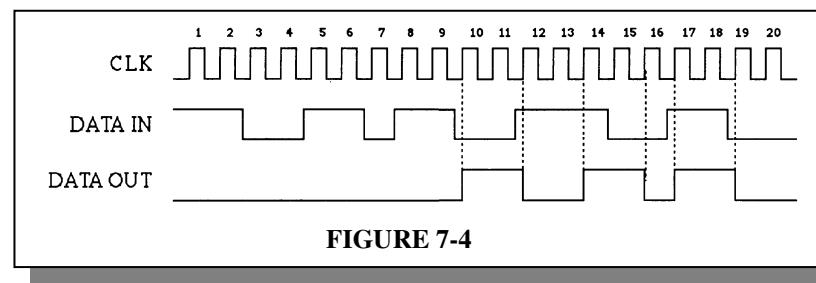
|           |              |
|-----------|--------------|
| Initially | 101001111000 |
| CLK 1     | 010100111100 |
| CLK 2     | 001010011110 |
| CLK 3     | 000101001111 |
| CLK 4     | 000010100111 |
| CLK 5     | 100001010011 |
| CLK 6     | 110000101001 |
| CLK 7     | 111000010100 |
| CLK 8     | 011100001010 |
| CLK 9     | 001110000101 |
| CLK 10    | 000111000010 |
| CLK 11    | 100011100001 |
| CLK 12    | 110001110000 |

16. See Figure 7-3.



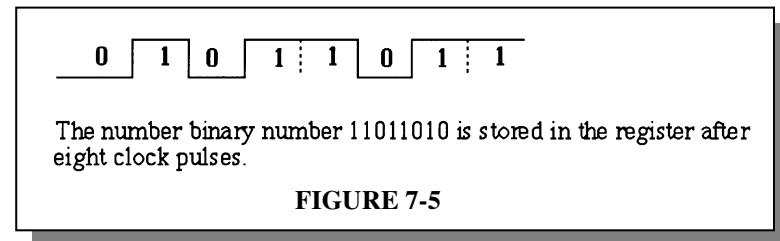
**FIGURE 7-3**

17. See Figure 7-4.



**FIGURE 7-4**

18. See Figure 7-5.



**FIGURE 7-5**

## Chapter 7

19. See Figure 7-6.

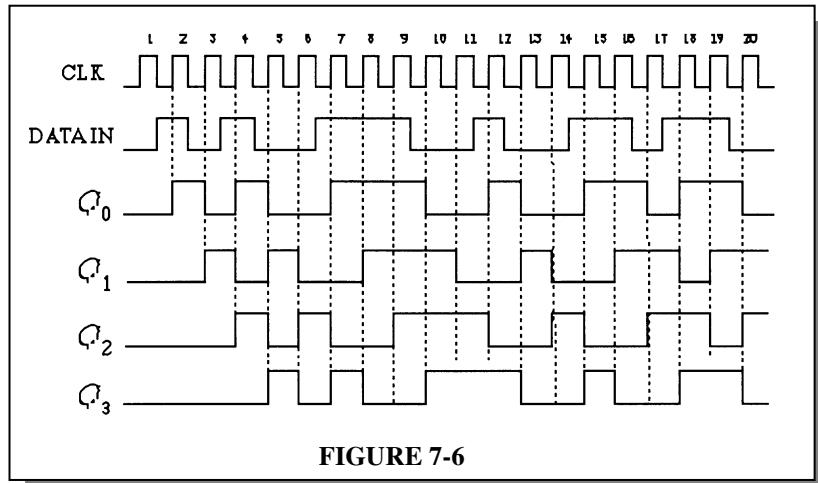


FIGURE 7-6

20. See Figure 7-7.

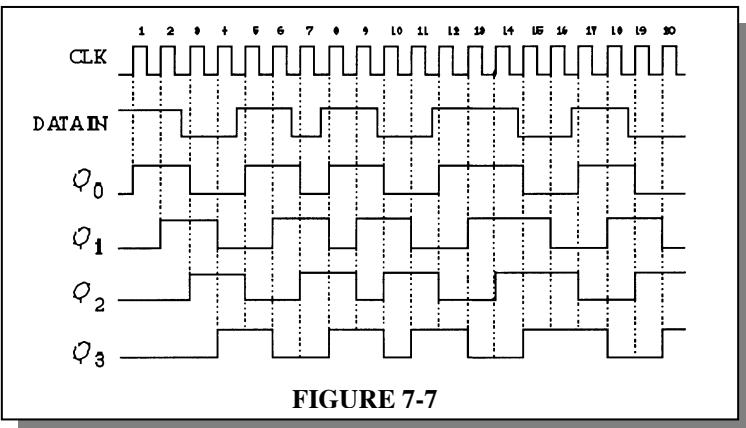


FIGURE 7-7

21. See Figure 7-8.

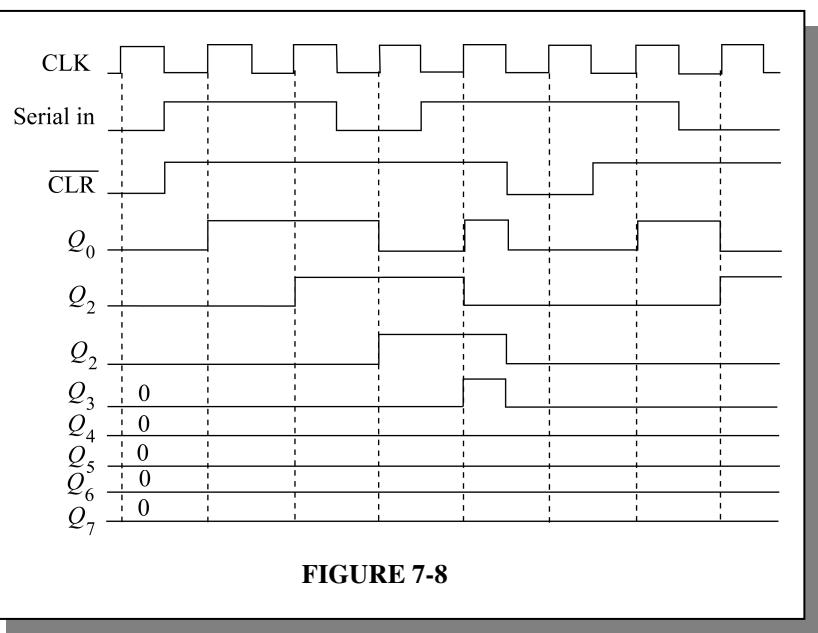
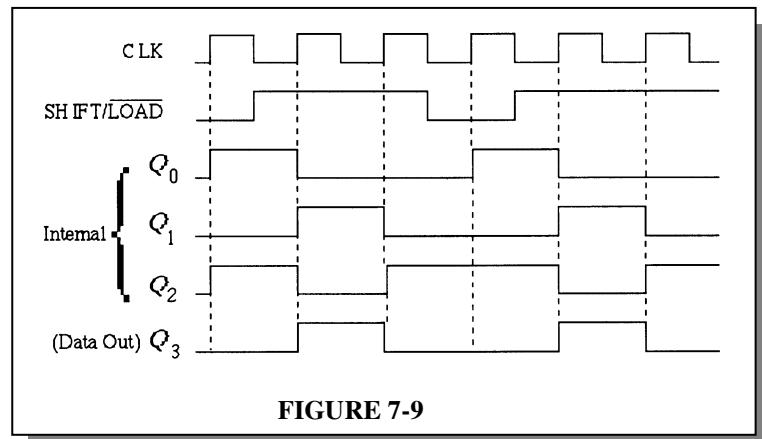


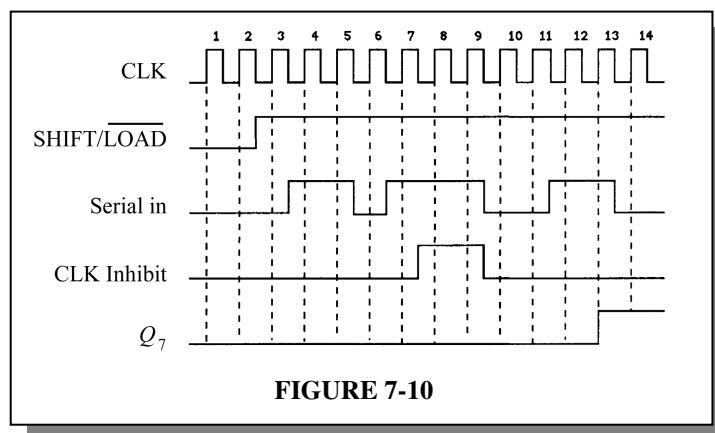
FIGURE 7-8

**22.** See Figure 7-9.



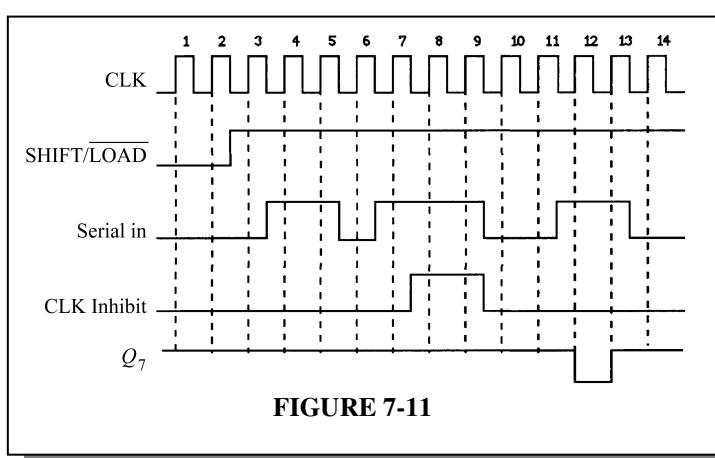
**FIGURE 7-9**

**23.** See Figure 7-10.



**FIGURE 7-10**

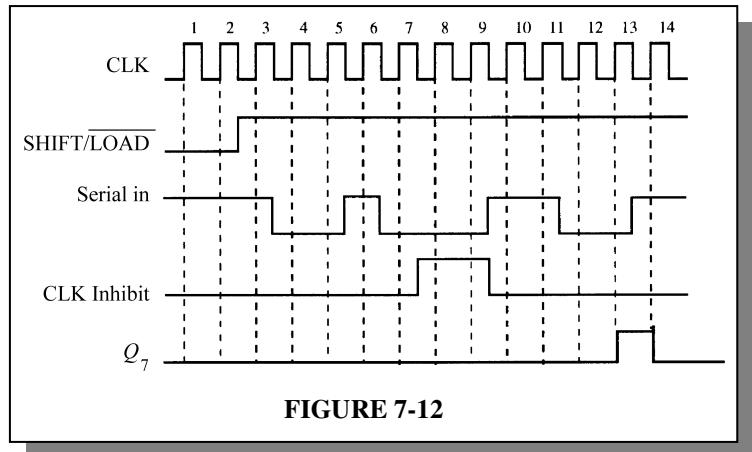
**24.** See Figure 7-11.



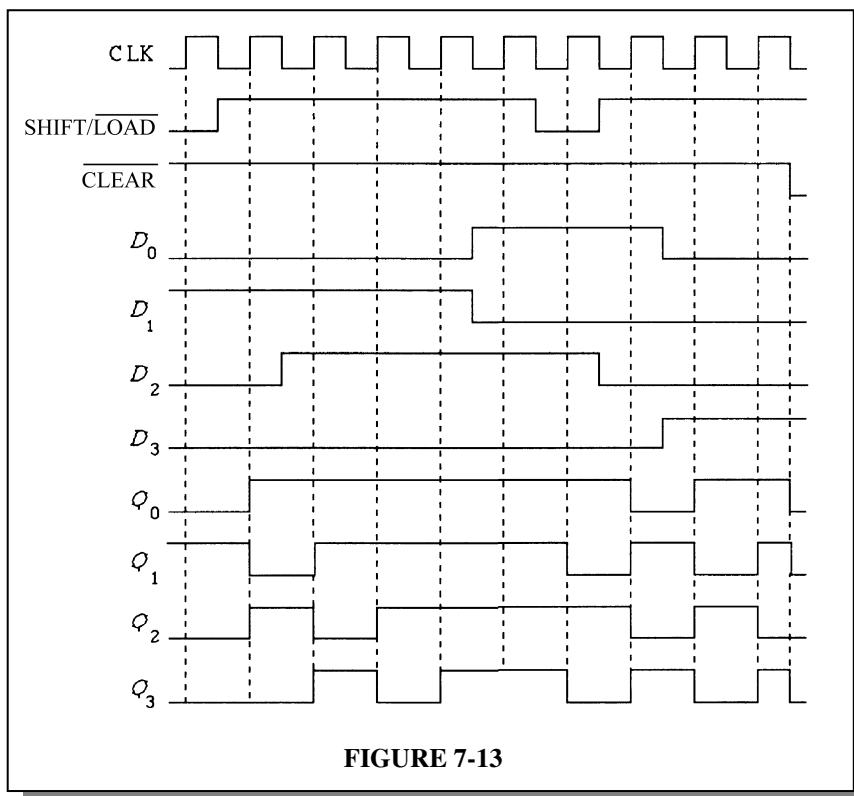
**FIGURE 7-11**

## *Chapter 7*

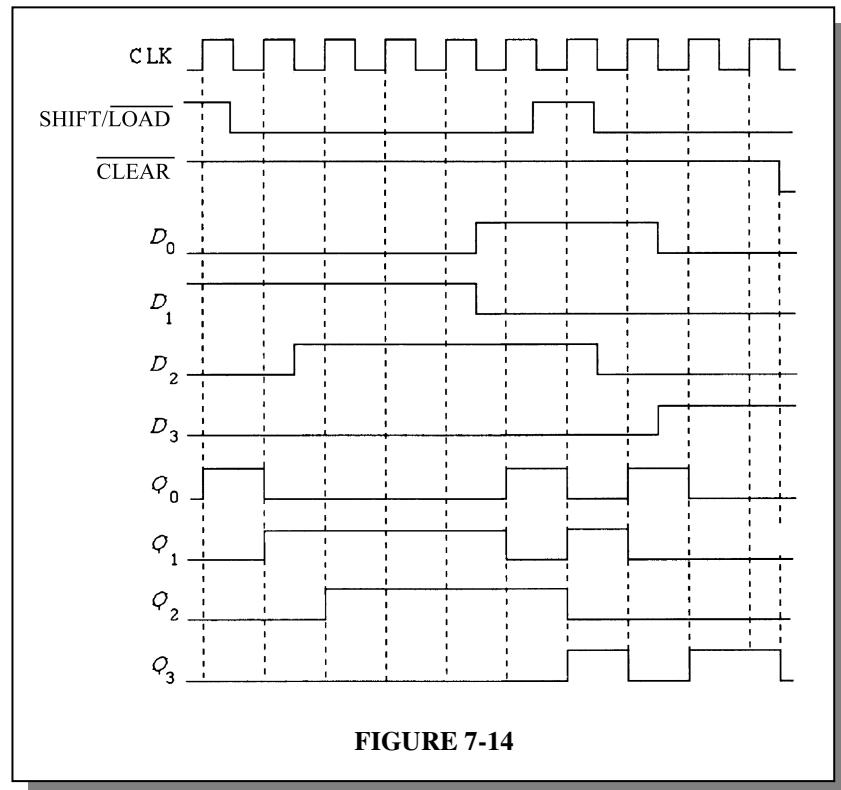
**25.** See Figure 7-12.



**26.** See Figure 7-13.

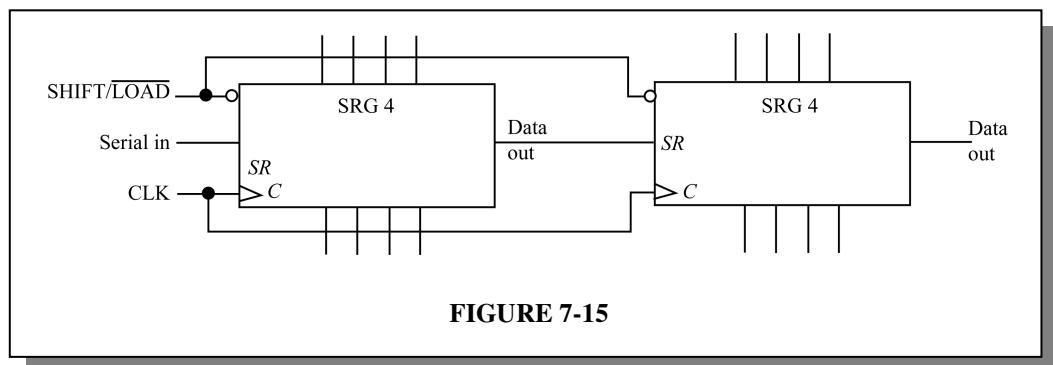


**27.** See Figure 7-14.



**FIGURE 7-14**

**28.** See Figure 7-15. Other configurations are possible.



**FIGURE 7-15**

## Chapter 7

### Section 7-4 Bidirectional Shift Registers

29.

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

30.

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

31. See Figure 7-16. Other configurations are possible.

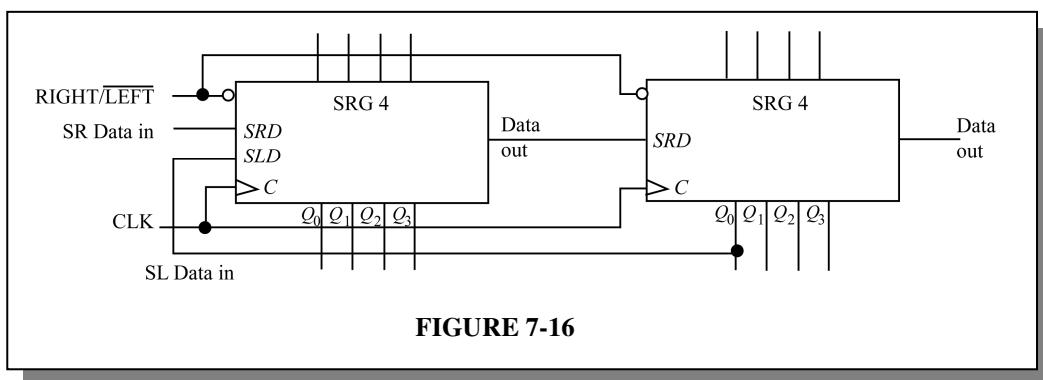
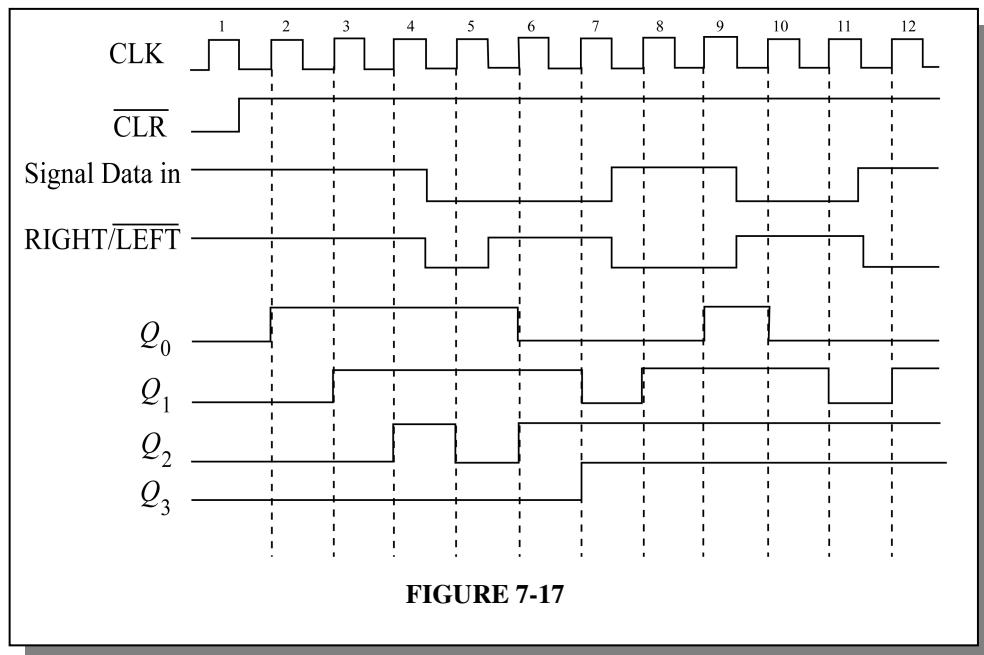


FIGURE 7-16

- 32.** See Figure 7-17.



### **Section 7-5 Shift Register Counters**

- |            |                          |                          |
|------------|--------------------------|--------------------------|
| <b>33.</b> | (a) $2n = 6$<br>$n = 3$  | (b) $2n = 10$<br>$n = 5$ |
|            | (c) $2n = 14$<br>$n = 7$ | (d) $2n = 16$<br>$n = 8$ |

- 34.**  $2n = 18$ ;  $n = 9$  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     |
| 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     | 0     | 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     |

## Chapter 7

See Figure 7-18.

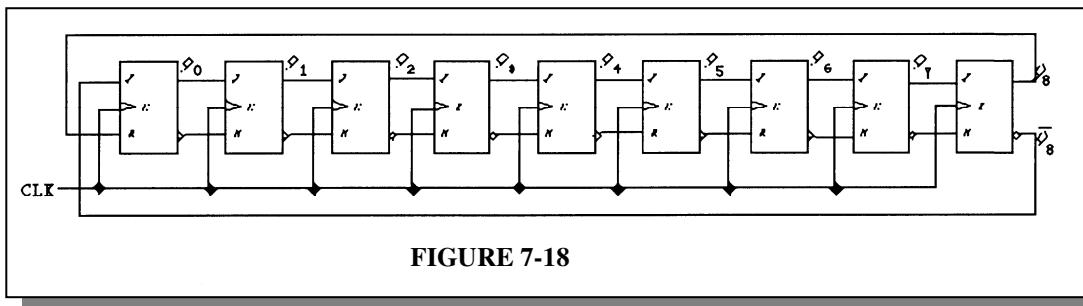


FIGURE 7-18

35. See Figure 7-19.

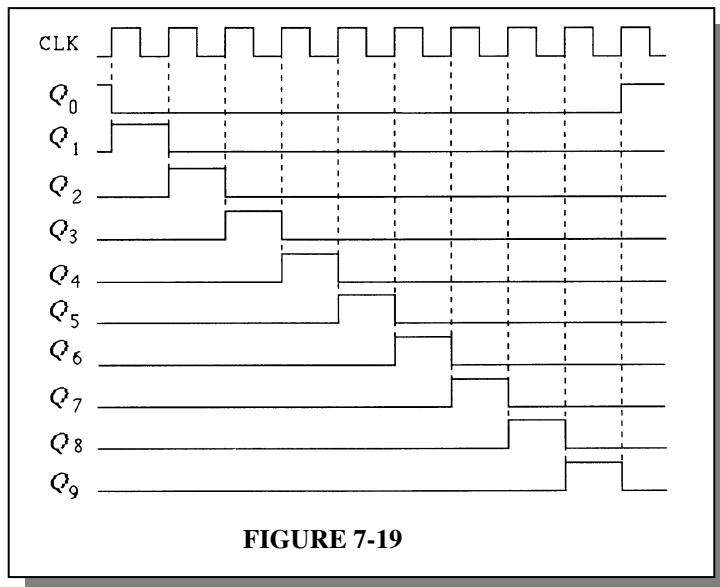


FIGURE 7-19

36. A 15-bit ring counter with stages 3, 7, and 12 SET and the remaining stages RESET. See Figure 7-20.

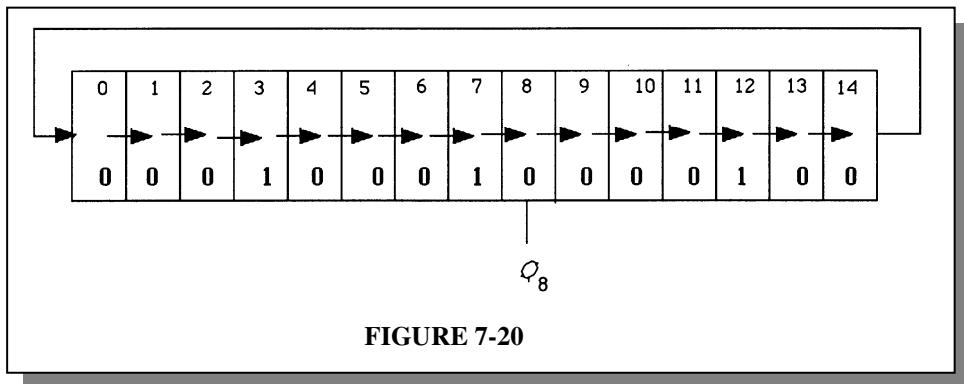


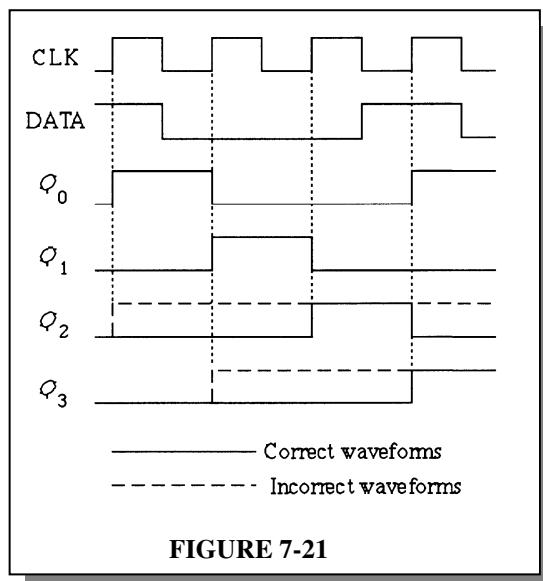
FIGURE 7-20

### Section 7-6 Security System with VHDL and Verilog

37. To accommodate a five-digit code, the hard-coded binary value stored in the 8-bit shift register C component will need to be changed from 00010000 to 00100000.
38. To assure that the 8-bit shift register C shifts correctly, the information from 4-bit shift registers A and B must first be clocked in to the magnitude comparator to avoid a potential race condition. The one-shot B clocks the shift register on stabilized data.
39. If two key presses were read by the code selection component before one-shot A can time out, a stored code value would be skipped. The system would read the second value as an error and the system would not disarm.

### Section 7-7 Troubleshooting

40.  $Q_2$  goes HIGH on the first clock pulse indicating that the  $D$  input is open. See Figure 7-21.



**FIGURE 7-21**

## Chapter 7

41. 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 7-22.

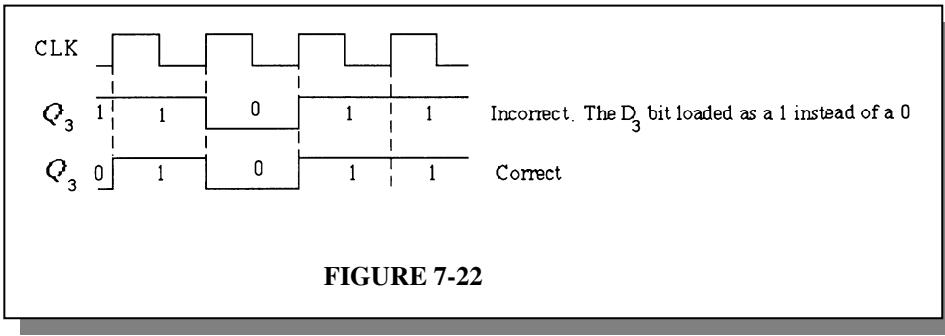


FIGURE 7-22

42. It takes a LOW on the RIGHT/LEFT input to shift data left. An open inverter input will keep the inverter output LOW thus disabling all of the shift-left control gates G5, G6, G7, and G8.
43. (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.
44. 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 7-1.

**Table 7-1**

| 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     |

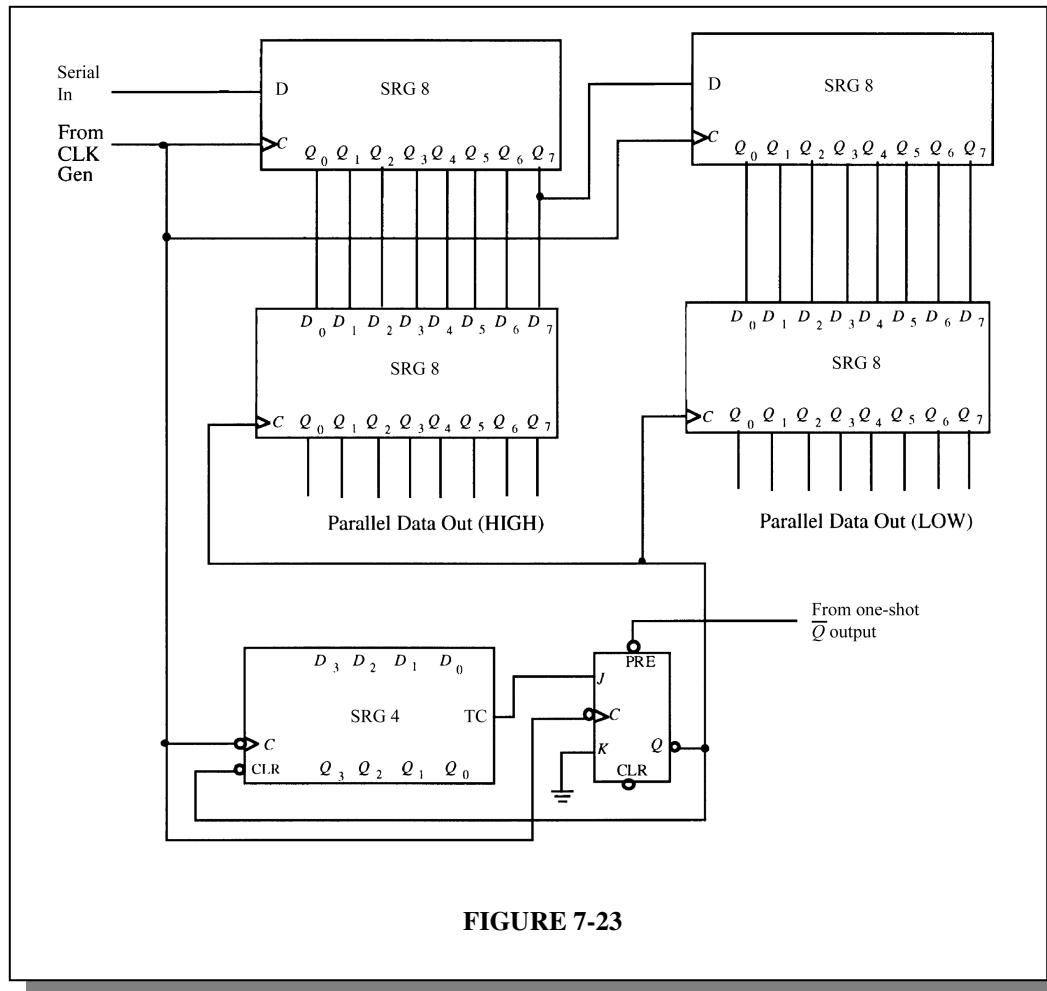
## *Chapter 7*

|    |   |   |   |   |   |   |
|----|---|---|---|---|---|---|
| 46 | 1 | 0 | 0 | 1 | 0 | 0 |
| 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 | 0 | 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 |

- 45.** (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.

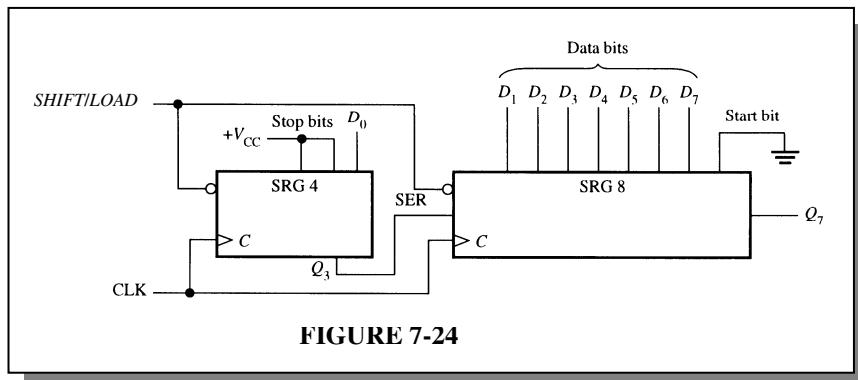
### **Special Problems**

- 46.** Figure 7-23 shows only the shift register, portions of the circuit that require modification for 16-bit conversion.



**FIGURE 7-23**

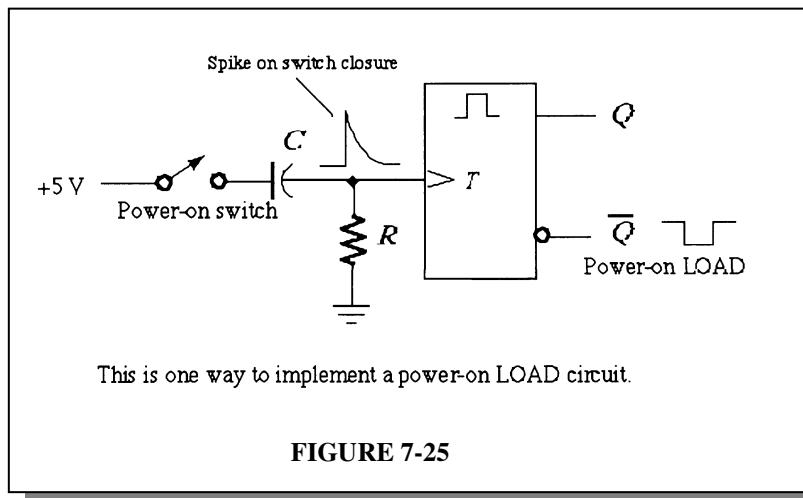
- 47.** See Figure 7-24 for one possible implementation.



**FIGURE 7-24**

## Chapter 7

48. One possible approach is shown in Figure 7-25.



### Multisim Troubleshooting Practice

49. **Circuit fault:** The line to the CLK input of U3 is open.

**Predicted effect of fault:** Q0 and Q1 will sequence normally. Q2 and Q3 will remain LOW.

**Observed effect of introduced fault:** Q0 and Q1 will sequence normally. Q2 and Q3 will remain LOW.

50. **Circuit fault:** The B input of the 74LS164 shift register is shorted to VCC.

**Predicted effect of fault:** The counter is enabled whenever the A input of the 74LS164 counter is HIGH.

**Observed effect of introduced fault:** The counter is enabled whenever the A input of the 74LS164 counter is HIGH.

51. **Circuit fault:** The QH input of the 74LS165 shift register is shorted to ground.

**Predicted effect of fault:** The QH input is always LOW but the QHbar output behaves as expected.

**Observed effect of introduced fault:** The QH input is always LOW but the QHbar output behaves as expected.

52. **Observed operation:** The outputs of the 74LS195 shift register change as expected during shift operations but the QB output is always LOW during load operations.

**Suspected fault:** The B input is shorted to ground.

**Effect of introduced fault:** The outputs of the 74LS195 shift register change as expected during shift operations but the QB output is always LOW during load operations.

**53. Observed operation:** The ring counter can initiate circulation of 1s but not 0s.

**Suspected fault:** The RESETbar line of the first stage is shorted to VCC.

**Effect of introduced fault:** The ring counter can initiate circulation of 1s but not 0s.

---

## CHAPTER 8

### COUNTERS

---

#### Section 8-1 A System

- For the digital clock in Figure 8-1 of the text, reset to 12:00:00, the binary state of each counter after sixty-two 60-Hz pulses are:

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

- 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:**

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

**Hours counter:**

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

#### Section 8-2 Finite State Machines

- The decoded decade counter is a Moore machine. See Figure 8-1.

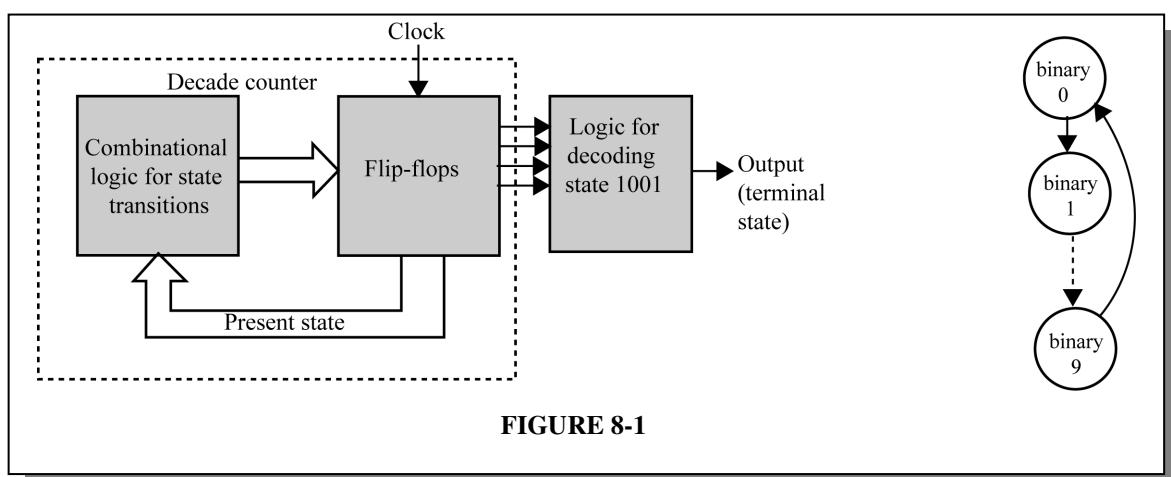
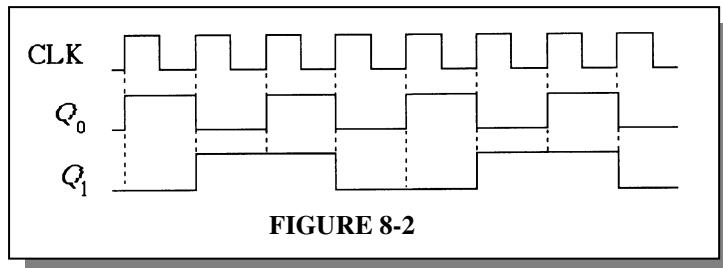


FIGURE 8-1

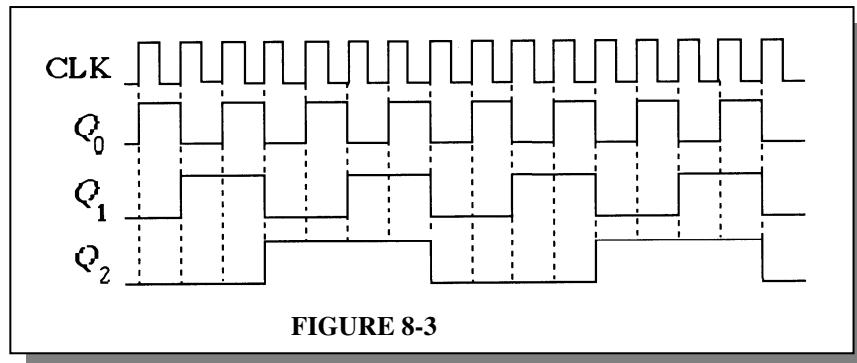
4. The traffic signal control system is a Mealy machine because it has an external input (not including the clock) and the outputs are from the combinational logic portion of the system.

### **Section 8-3 Asynchronous Counter Operation**

5. See Figure 8-2.



6. See Figure 8-3.

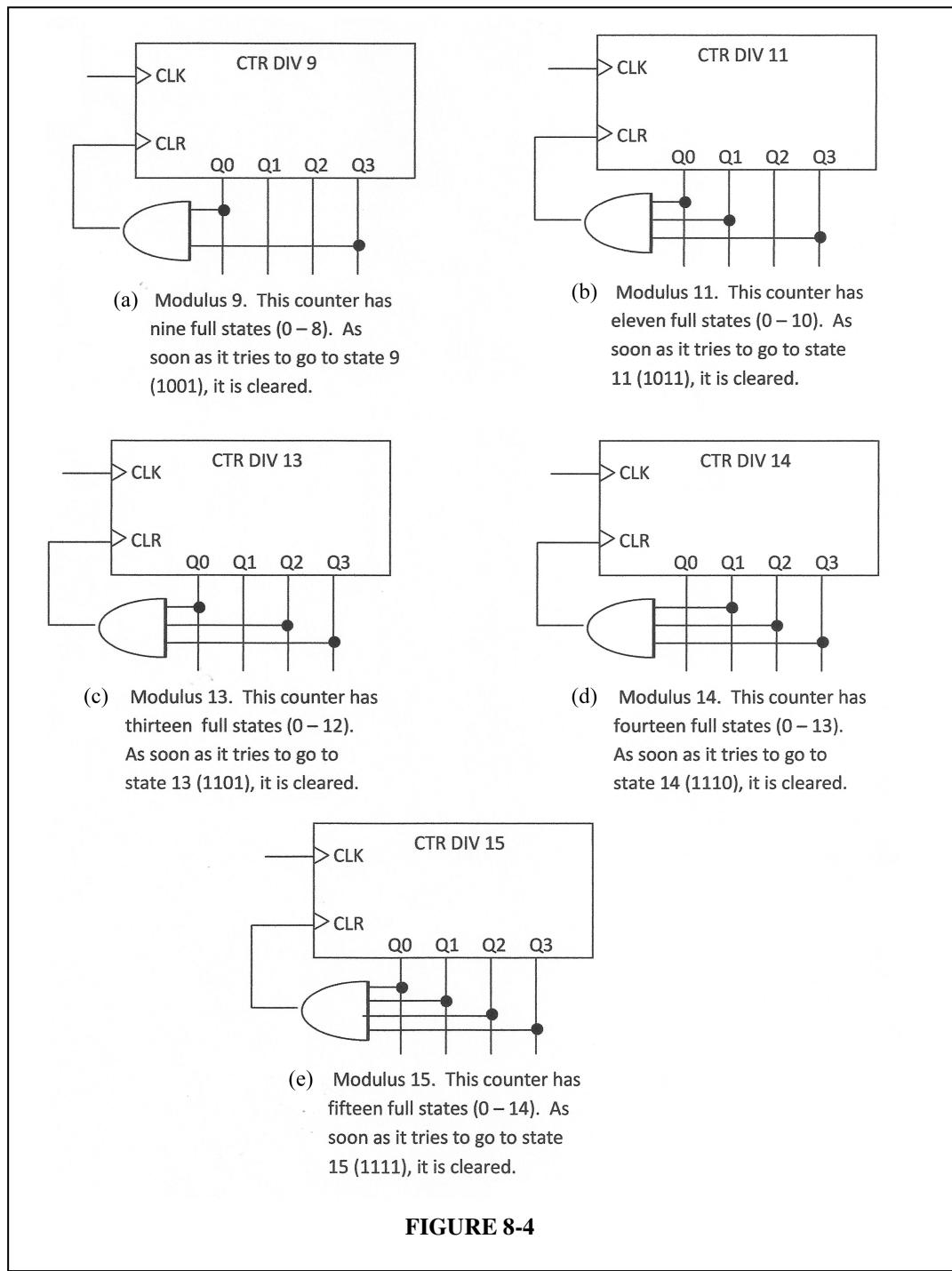


7.  $t_{p(\max)} = 3(8 \text{ ns}) = 24 \text{ ns}$

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

## Chapter 8

8. See Figure 8-4.

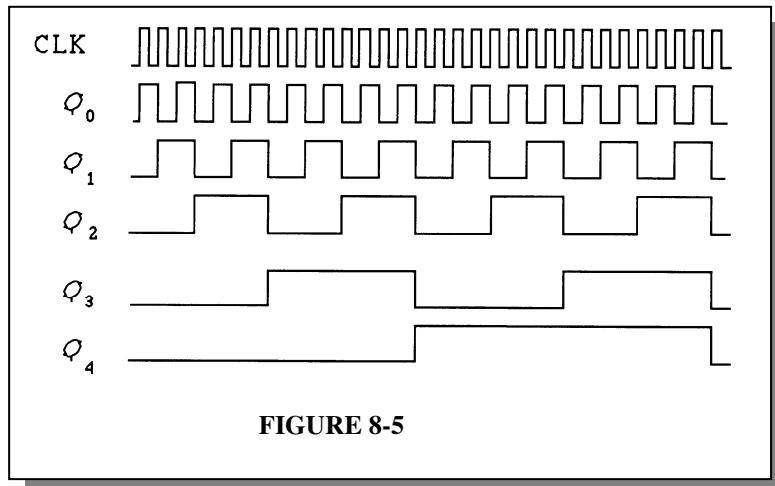


**FIGURE 8-4**

### Section 8-4 Synchronous Counters

9. **8 ns**, the time it takes one flip-flop to change state.

10. See Figure 8-5.

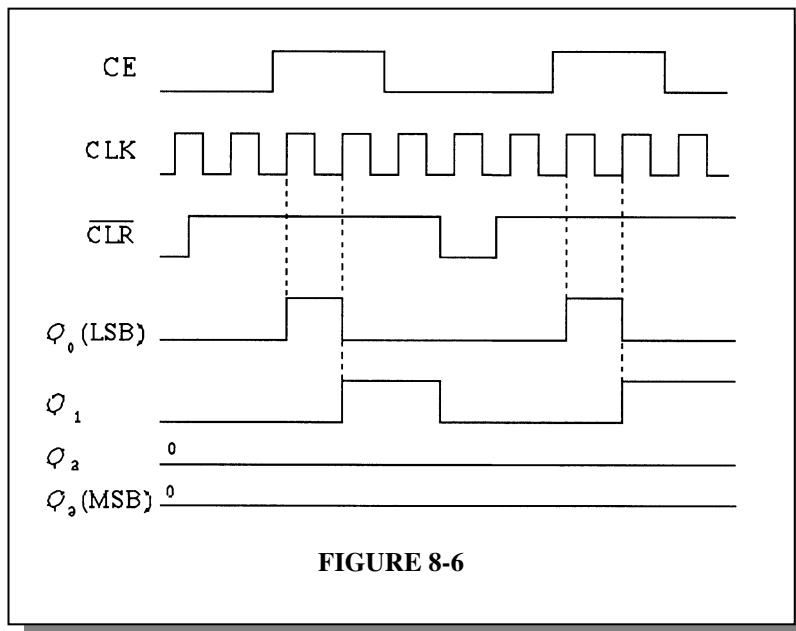


11. Each flip-flop is initially reset.

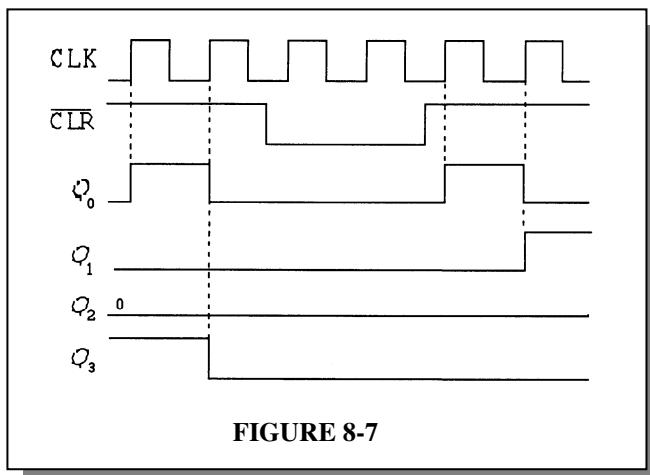
| CLK | $J_0K_0$ | $J_1K_1$ | $J_2K_2$ | $J_3K_3$ | $Q_0$ | $Q_1$ | $Q_2$ | $Q_3$ |
|-----|----------|----------|----------|----------|-------|-------|-------|-------|
| 1   | 1        | 0        | 0        | 0        | 1     | 0     | 0     | 0     |
| 2   | 1        | 1        | 0        | 0        | 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     |

## *Chapter 8*

12. See Figure 8-6.

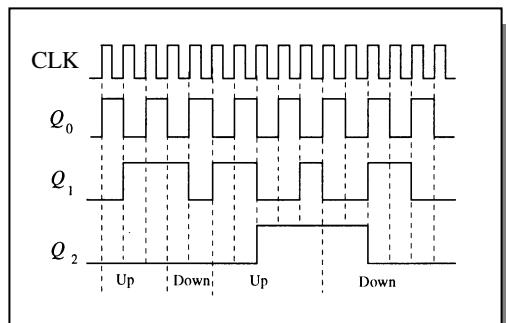


13. See Figure 8-7.



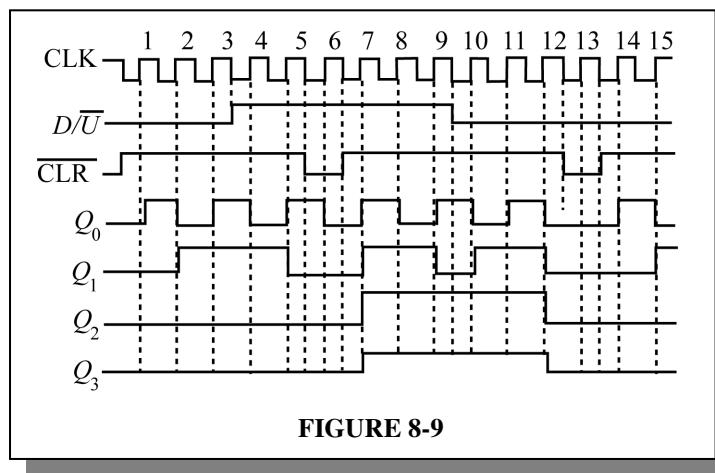
### *Section 8-5 Up/Down Synchronous Counters*

14. See Figure 8-8.



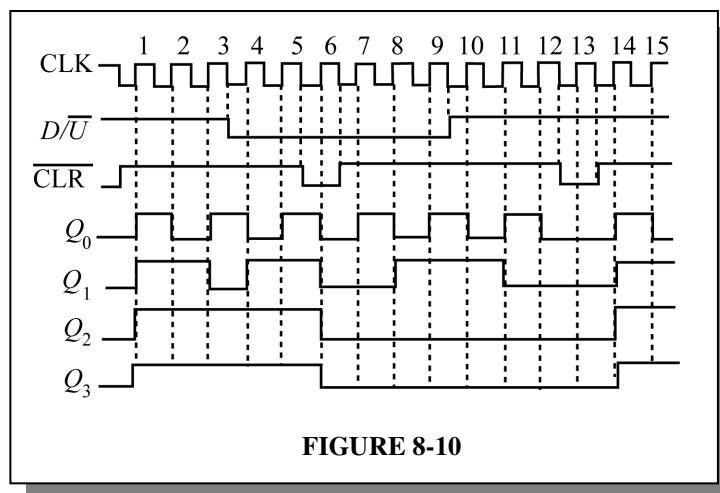
**FIGURE 8-8**

15. See Figure 8-9.



**FIGURE 8-9**

16. See Figure 8-10.



**FIGURE 8-10**

## Chapter 8

17. See Figure 8-11.

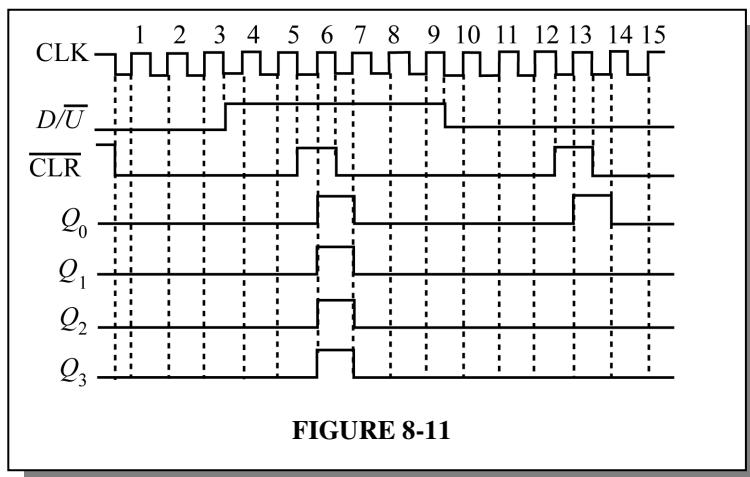


FIGURE 8-11

### Section 8-6 Cascaded Counters

18. (a) Modulus =  $4 \times 8 \times 2 = 64$

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

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

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

(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 =  $2 \times 4 \times 6 \times 8 \times 16 = 6144$

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

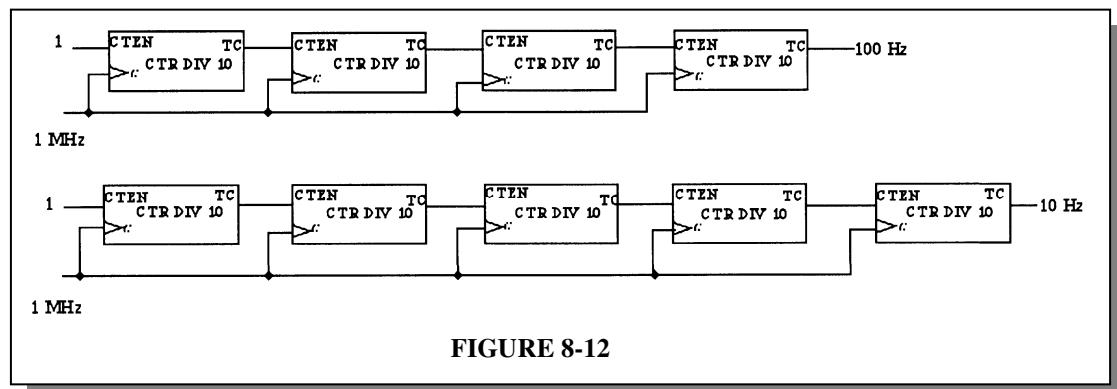
$$f_2 = \frac{19.7 \text{ kHz}}{4} = 4.925 \text{ kHz}$$

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

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

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

19. See Figure 8-12.



## Chapter 8

20. See Figure 8-13.

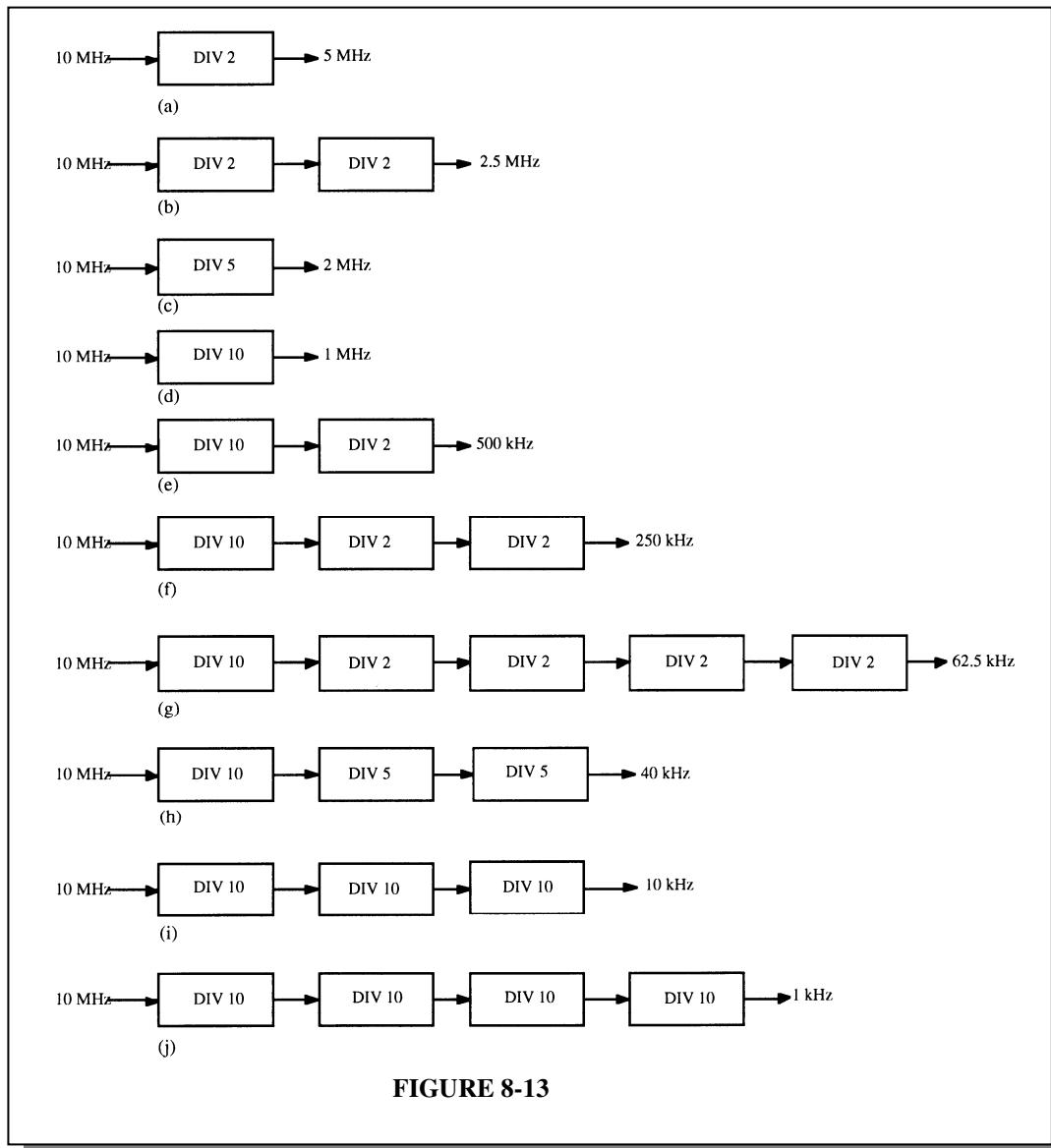
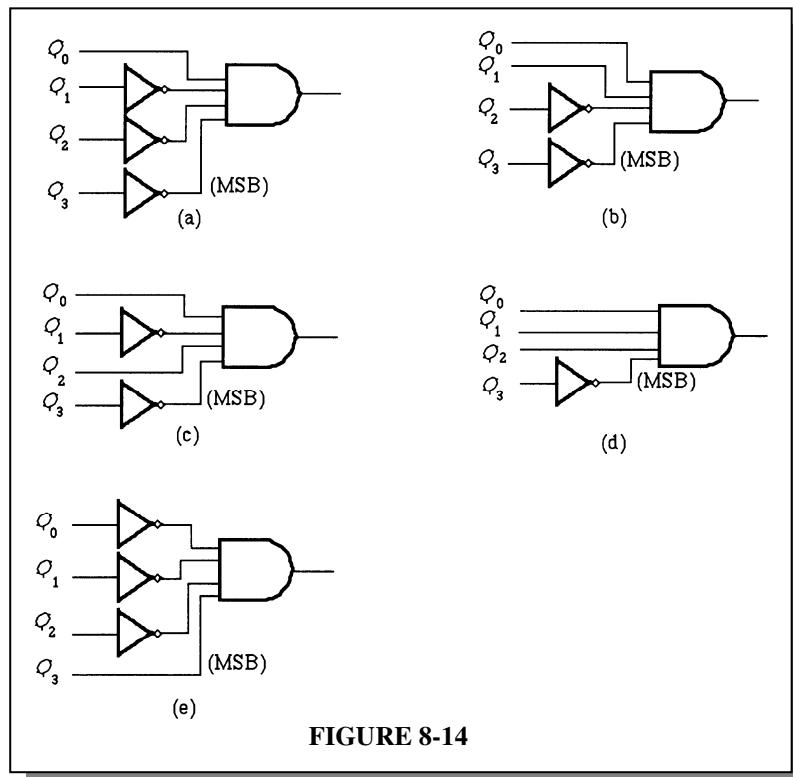


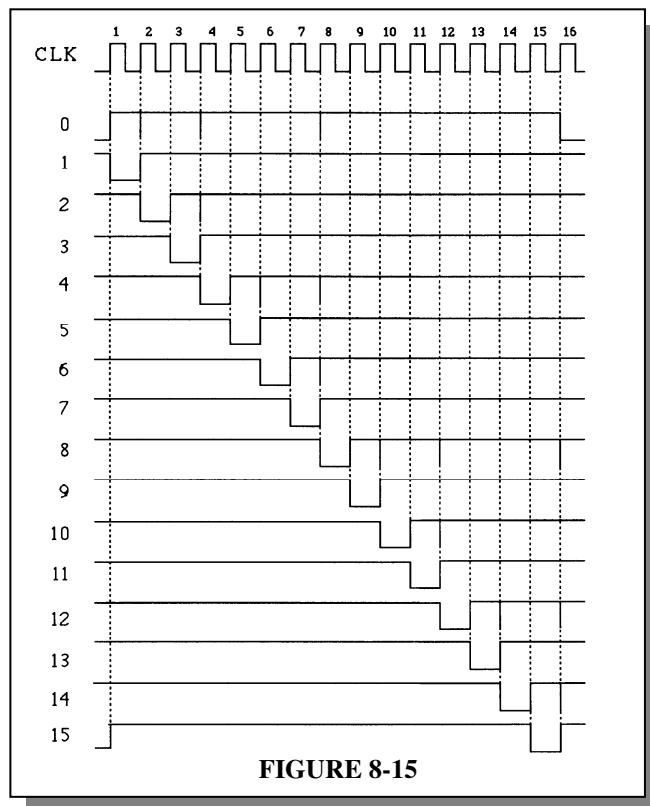
FIGURE 8-13

### Section 8-7 Counter Decoding

21. See Figure 8-14.



22. See Figure 8-15.

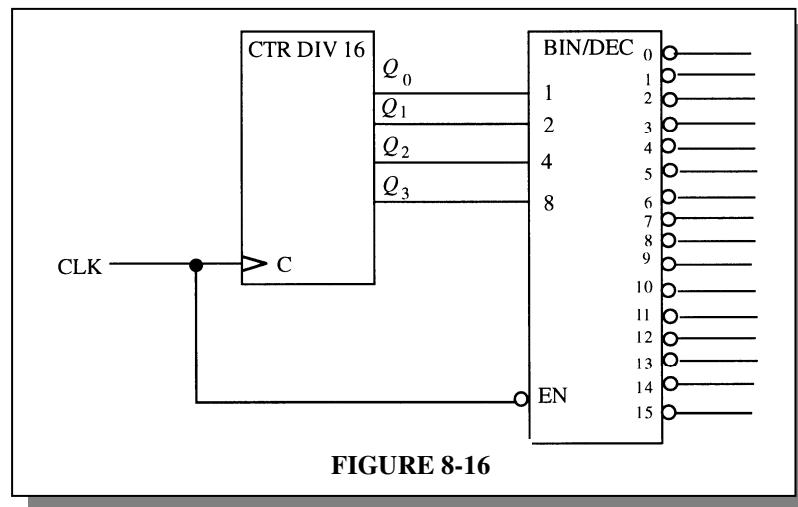


## Chapter 8

- 23.** 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 8-15 (Problem 8-22) by solid short vertical lines.

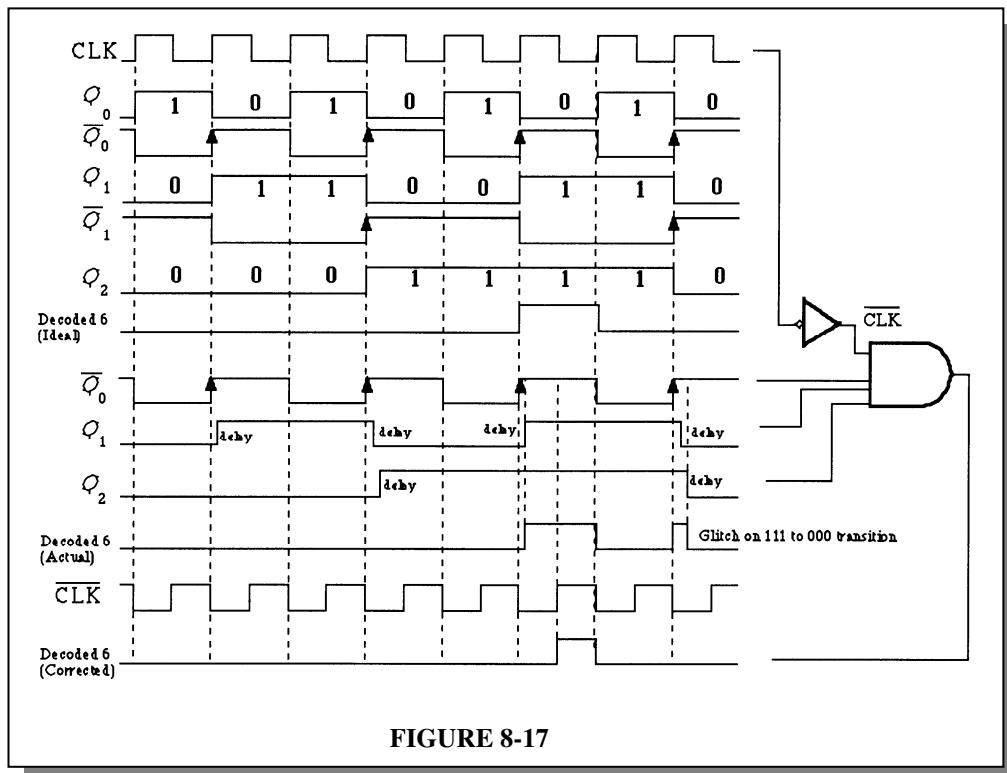
|         |        |
|---------|--------|
| Initial | 0000   |
| CLK 1   | 0001   |
| CLK 2   | 0000 * |
|         | 0010   |
| CLK 3   | 0011   |
| CLK 4   | 0010 * |
|         | 0000 * |
|         | 0100   |
| CLK 5   | 0100   |
| CLK 6   | 0100 * |
|         | 0110   |
| CLK 7   | 0111   |
| CLK 8   | 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   |

- 24.** See Figure 8-16.



**FIGURE 8-16**

25. See Figure 8-17.

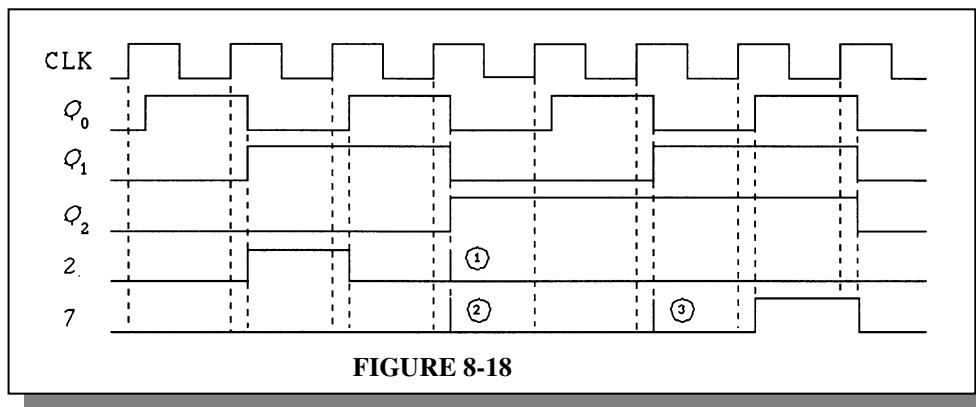


**FIGURE 8-17**

26. ① There is a possibility of a glitch on decode 2 at the positive-going edge of CLK 4 if the propagation delay of FF0 is less than FF1 or FF2.  
 ② There is a possibility of a glitch on decode 7 at the positive-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 8-18 which is expanded to show the delays.

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



**FIGURE 8-18**

## *Chapter 8*

### *Section 8-8 Counters with VHDL and Verilog*

27. The addition AND gate (G3) causes the additional fifth JK flip flop stage (FF4) to change when Q<sub>0</sub>, Q<sub>1</sub>, Q<sub>2</sub>, and Q<sub>3</sub> are HIGH.

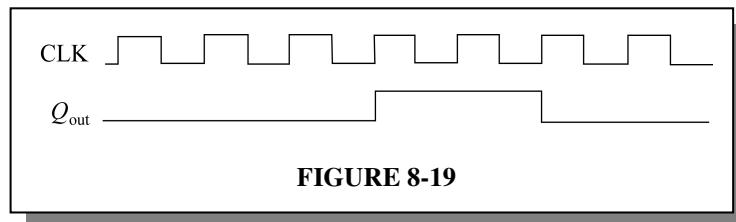
**VHDL**      G3 <= G2 and Q3;

**Verilog** assign G3 = G2 && Q3;

28. Advantages of implementing a counter system using VHDL are (answers may vary)

1. Simulation allows for logic verification.
2. The counter behavior can be modified without changing hardware.
3. Common flip-flop definitions can be used for multiple applications.
4. Reduced hardware.
5. Reusable code for multiple application.

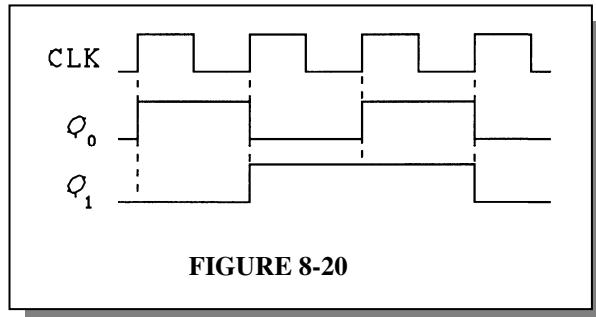
29. The timing diagram for the 4-bit synchronous decade counter is shown in Figure 8-19.



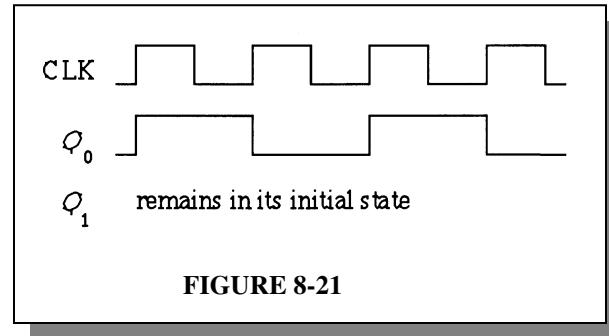
30. The binary value 2 is raised to the second power.

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

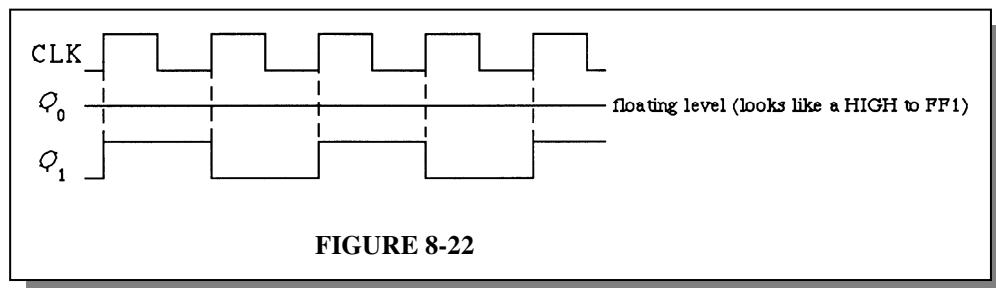
- 31.** (a)  $Q_0$  and  $Q_1$  will not change due to the clock shorted to ground at FF0.
- (b)  $Q_0$  being open does not affect normal operation. See Figure 8-20.



- (c) See Figure 8-21.

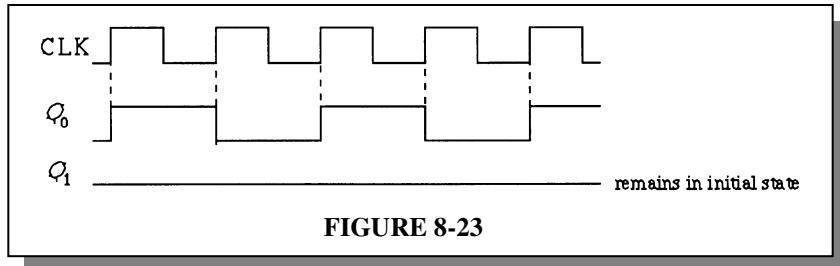


- (d) Normal operation because an open  $J$  input acts as a HIGH.
- (e) A shorted  $K$  input will pull all  $J$  and  $K$  inputs LOW and the counter will not change from its initial state.
- 32.** (a)  $Q_0$  and  $Q_1$  will not change from initial states.
- (b) See Figure 8-22.



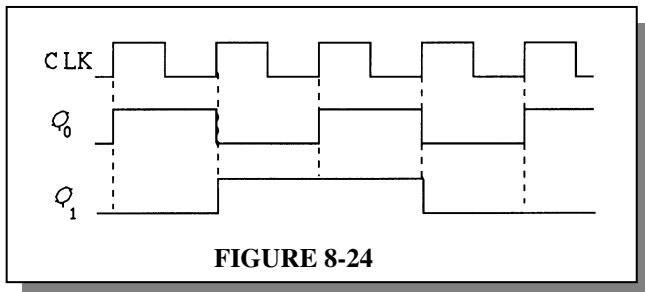
## Chapter 8

- (c) See Figure 8-23.



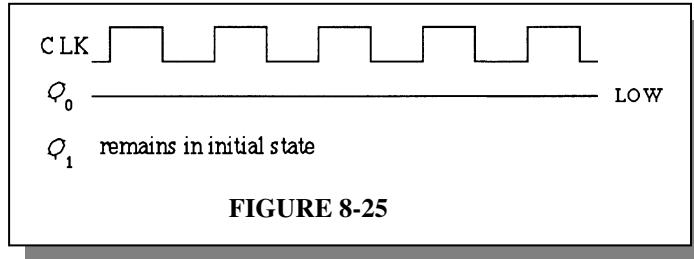
**FIGURE 8-23**

- (d) Normal operation. See Figure 8-24.



**FIGURE 8-24**

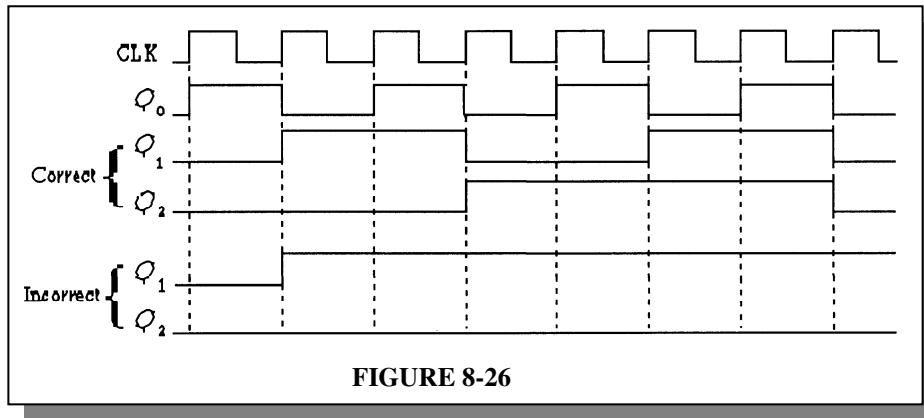
- (e) Both  $J$  and  $K$  of FF1 are pulled LOW if  $K$  is grounded, producing a no-change condition.  $Q_0$  also grounded. See Figure 8-25.



**FIGURE 8-25**

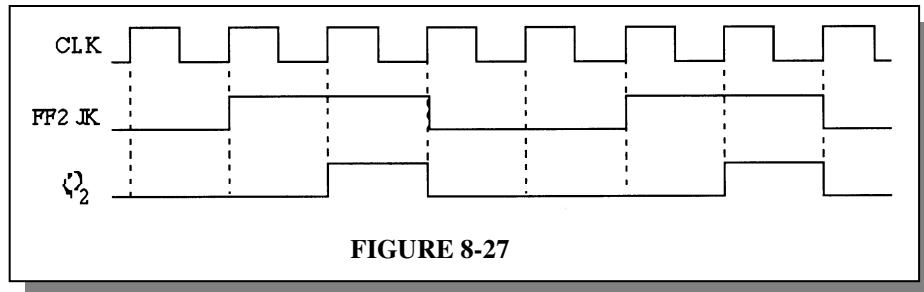
33. First, determine the correct waveforms and observe that  $Q_0$  is correct but  $Q_1$  and  $Q_2$  are incorrect in Figure 8-56 in the text. See Figure 8-26.

Since  $Q_1$  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.



**FIGURE 8-26**

34. Since  $Q_2$  toggles on each clock pulse, its  $J$  and  $K$  inputs must be constantly HIGH. The most probable fault is that the AND gate's output is *open*.
35. If the  $Q_0$  input to the AND gate is *open*, the  $JK$  inputs to FF2 are as shown in Figure 8-27.



36. 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<sub>16</sub> on its parallel inputs is not being loaded. Possible faults are:

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

- 37.

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

## Chapter 8

38. □ The flip-flop output is stuck HIGH or open.  
 □ The least significant BCD/7-segment input is open.

See Figure 8-28.

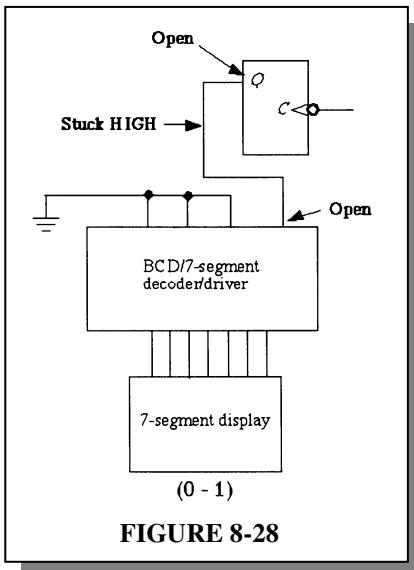


FIGURE 8-28

39. 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.

| Actual State of Ctr. | Apparent state |       |       |       |
|----------------------|----------------|-------|-------|-------|
|                      | $Q_3$          | $Q_2$ | $Q_1$ | $Q_0$ |
| 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.

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

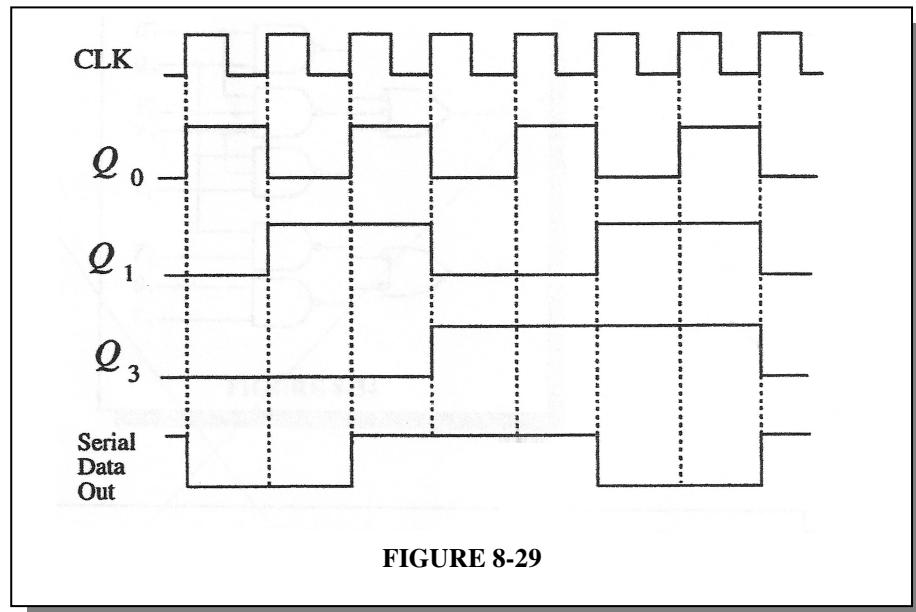
- Sensor Latch
  - Action:* Disconnect entrance sensor and pulse sensor input.
  - Observation:* Latch should SET.
  - Conclusion:* If latch does not SET, replace it.
- NOR gate
  - Action:* Pulse sensor input.
  - Observation:* Pulse on gate output.
  - Conclusion:* If there is no pulse, replace gate.

- 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.

### ***Special Problems***

**41.**  $53 + 37 - 22 = 68$

**42.** See Figure 8-29.



**FIGURE 8-29**

## Chapter 8

43. See Figure 8-30.

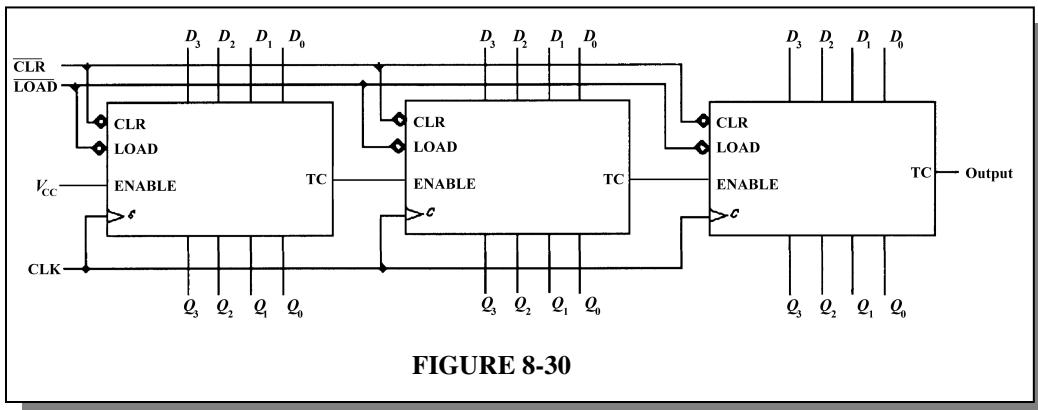


FIGURE 8-30

$$44. \quad 65,536 - 30,000 = 35,536$$

Preset the counter to 35,536 so that it counts from 35,536 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 8-31.

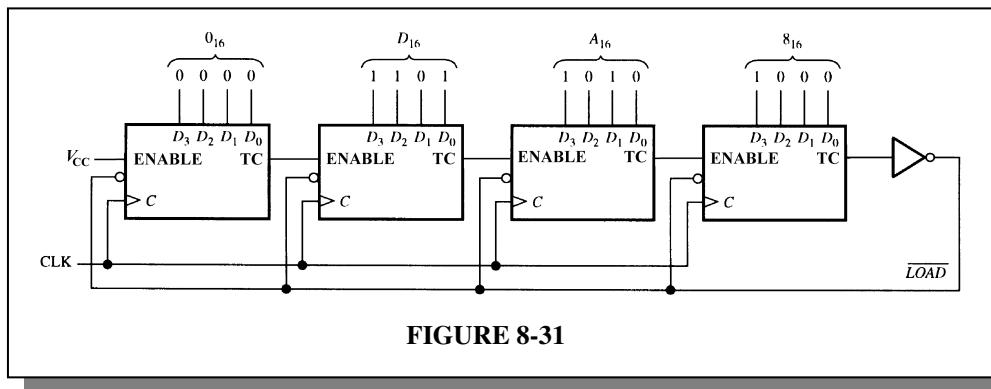


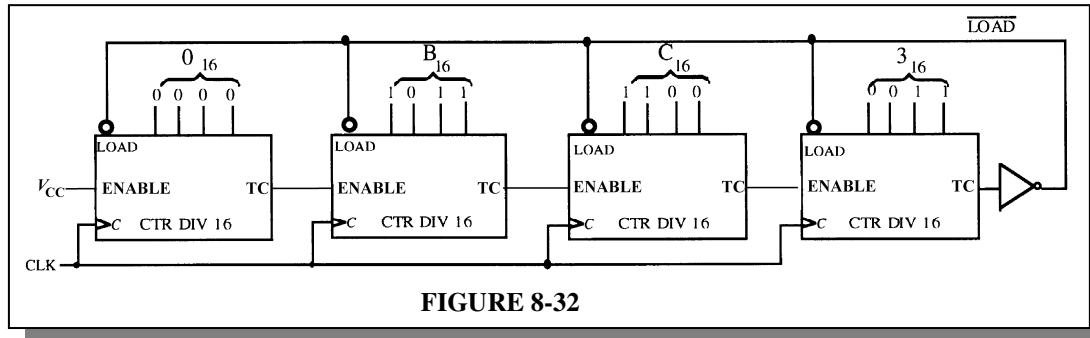
FIGURE 8-31

$$45. \quad 65,536 - 50,000 = 15,536$$

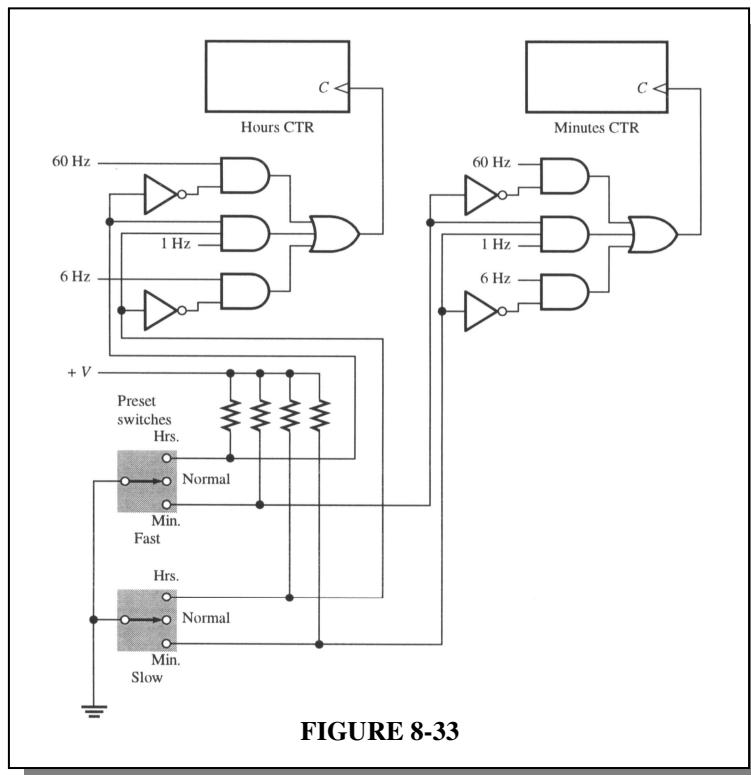
Preset the counter to 15,536 so that it counts from 15,536 up to 65,536 on each full cycle, thus producing a sequence of 50,000 states (modulus 50,000).

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

See Figure 8-32.

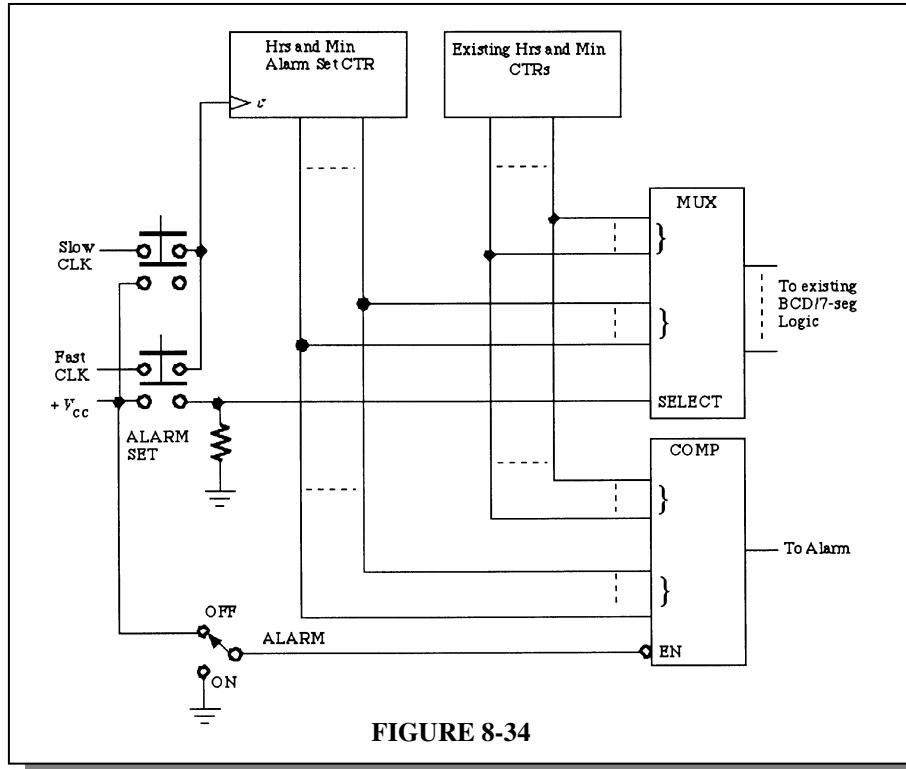


46. The approach is to preset the hours and minutes counters independently, each with a fast or slow preset mode. The seconds counter is not preset. One possible implementation is shown in Figure 8-33.

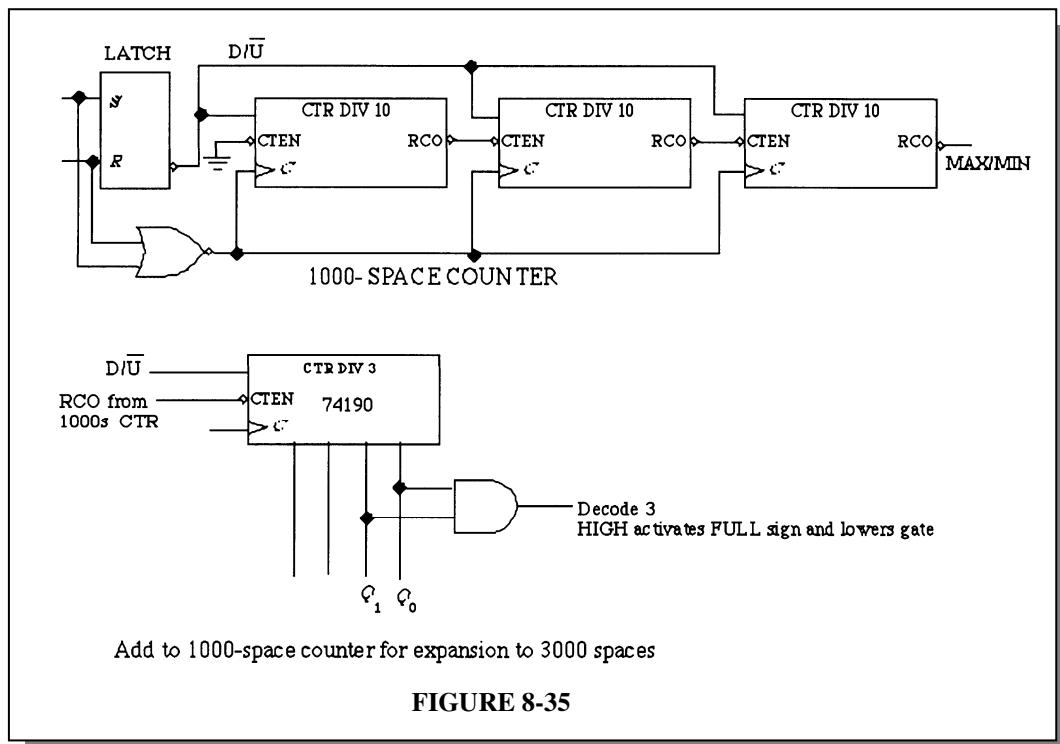


## Chapter 8

47. See Figure 8-34.



48. See Figure 8-35.



## **Multisim Troubleshooting Practice**

- 49. Circuit fault:** The line to the CLK input of U3 is open.

**Predicted effect of fault:** Q0 and Q1 will sequence normally. Q2 and Q3 will remain LOW.

**Observed effect of introduced fault:** Q0 and Q1 will sequence normally. Q2 and Q3 will remain LOW.

- 50. Circuit fault:** The input of the U5 AND gate that connects to the Q output of U2 shorted to VCC.

**Predicted effect of fault:** Q1 is always HIGH. The Q2Q1Q0 count sequence is 000, 001, 110, 000... rather than a 3-bit binary up count.

**Observed effect of introduced fault:** Q1 is always HIGH. The Q2Q1Q0 count sequence is 000, 001, 110, 000... rather than a 3-bit binary up count.

- 51. Circuit fault:** Q output of U3 is shorted to ground.

**Predicted effect of fault:** The Q3Q2Q1Q0 count sequence is 000, 0001, 0010, 0011, 0000... rather than a decade up count.

**Observed effect of introduced fault:** The Q3Q2Q1Q0 count sequence is 000, 0001, 0010, 0011, 0000... rather than a decade up count.

- 52. Observed operation:** Outputs of 74LS163 counter always matches the values on its parallel inputs during synchronous operation.

**Suspected fault:** Line to notLOAD input always LOW.

**Effect of introduced fault:** Outputs of &\$LS163 counter always matches the values on its parallel inputs during synchronous operation.

- 53. Observed operation:** 74LS190 decade counter always functions as a down counter.

**Suspected fault:** Line to notU/D input of counter always HIGH.

**Effect of introduced fault:** 74LS190 decade counter always functions as a down counter.

# CHAPTER 9

## PROGRAMMABLE LOGIC

### Section 9-1 Simple Programmable Logic Devices (SPLDs)

1.  $X = \overline{A}\overline{B}\overline{C} + \overline{A}\overline{B}\bar{C} + A\overline{B}\bar{C}$ . See Figure 9-1.

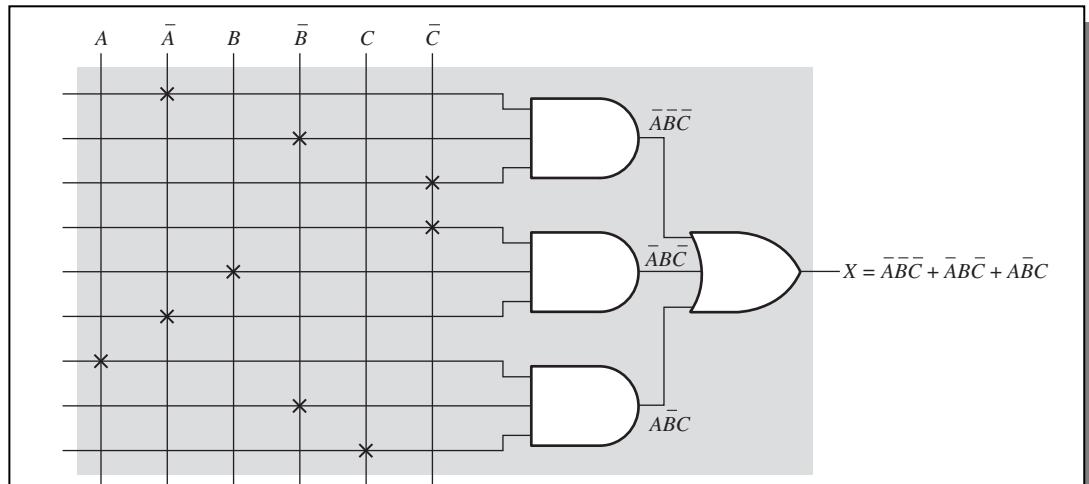


FIGURE 9-1

2. See Figure 9-2.

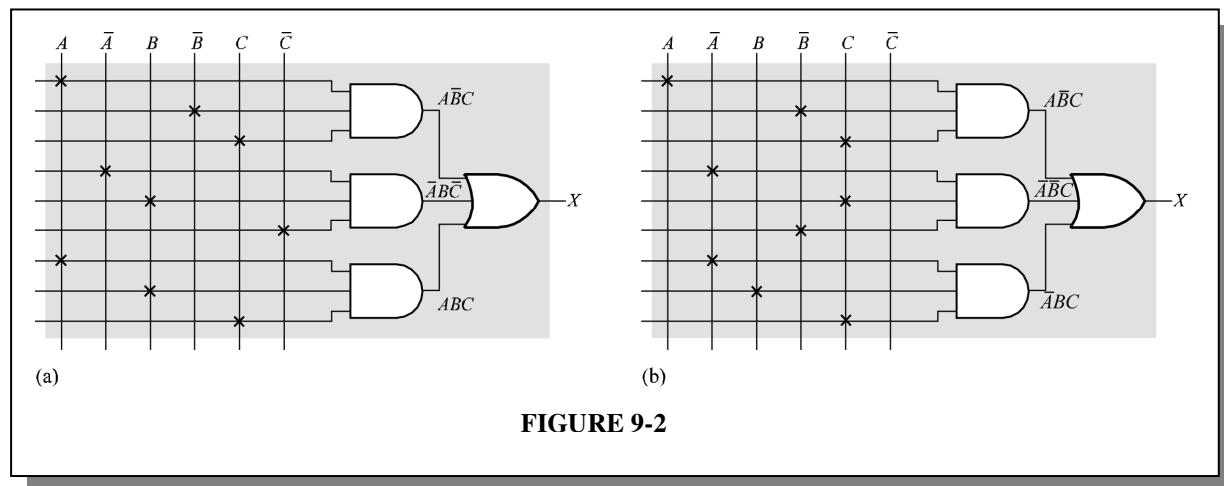
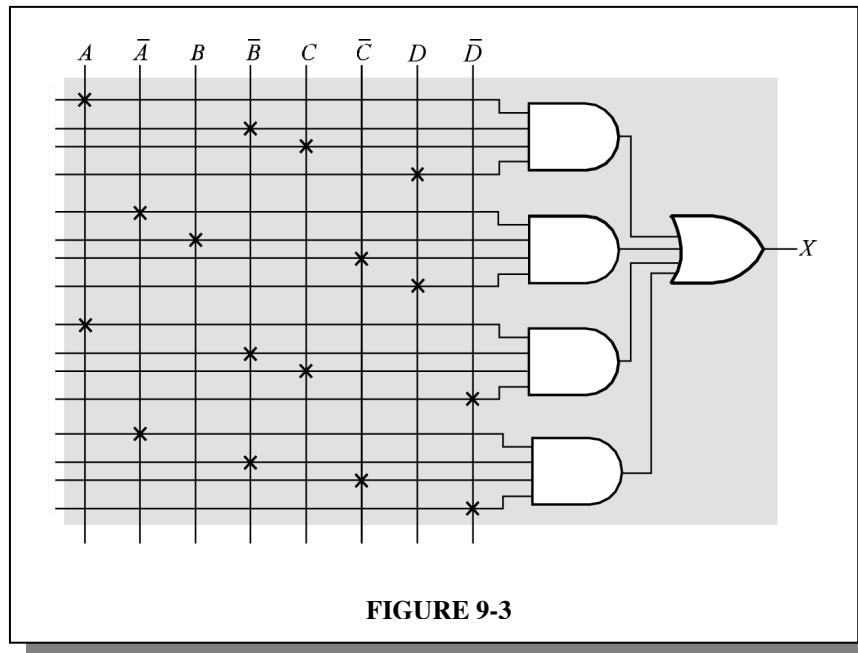


FIGURE 9-2

3. See Figure 9-3.



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.

### **Section 9-2 Complex Programmable Logic Devices (CPLDs)**

5. A CPLD basically consists of multiple SPLDs that can be connected with a programmable interconnect array.
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.  $A\overline{B}\overline{C}\overline{D} + EFGH + AB\overline{C}D + \overline{A}BCD$
9.  $A\overline{B} + \overline{A}B$

## Chapter 9

10. See Figure 9-4.

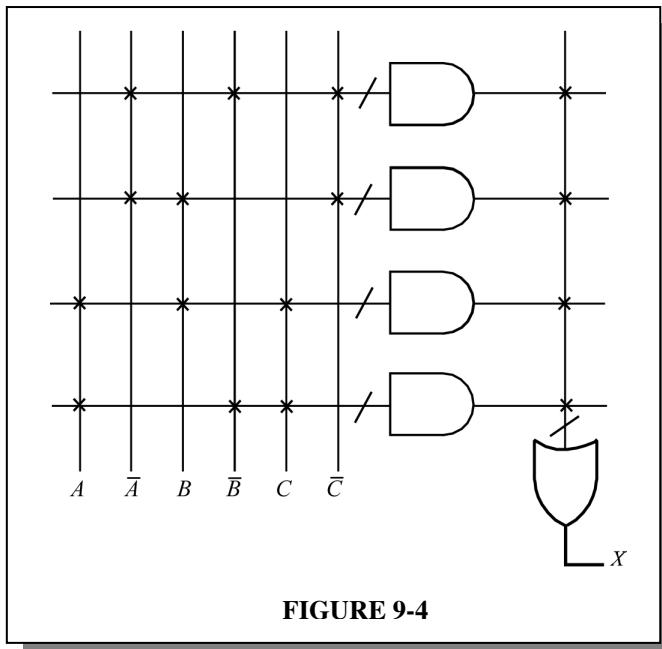


FIGURE 9-4

$$11. \quad X_1 = A\bar{B}\bar{C}\bar{D} + \bar{A}BC\bar{D} + ABC\bar{D}; \quad X_2 = ABCD + AB\bar{C}\bar{D} + \bar{A}BC\bar{D} + A\bar{B}CD$$

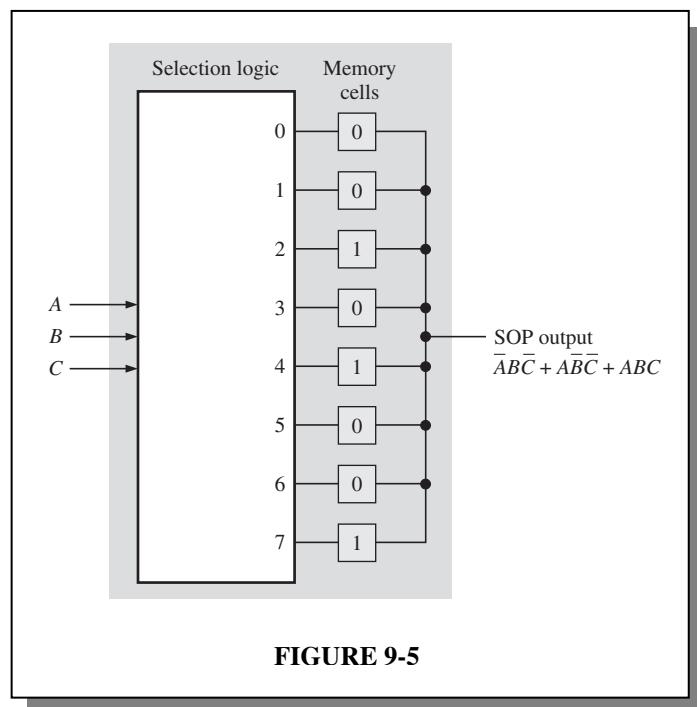
### Section 9-3 Macrocell Modes

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 9-4 Field-Programmable Gate Arrays (FPGAs)**

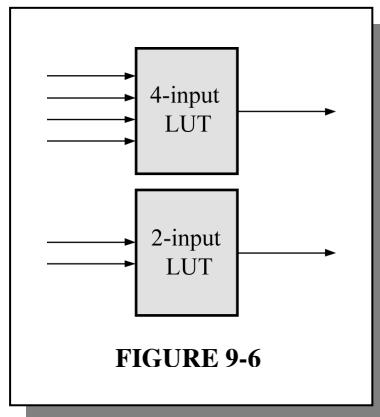
16. An FPGA typically consists of configurable logic blocks (CLBs). Each CLB is made up of a number of logic modules with a local interconnect. Each logic module typically consists of a look-up table (LUT) and associated logic. Global column and row interconnects are used to connect the CLBs to I/Os as well as each other.
17. SOP output =  $\overline{A}\overline{B}\overline{C} + \overline{A}\overline{B}C + \overline{ABC} + A\overline{B}\overline{C} + ABC$
18. See Figure 9-5.



**FIGURE 9-5**

## *Chapter 9*

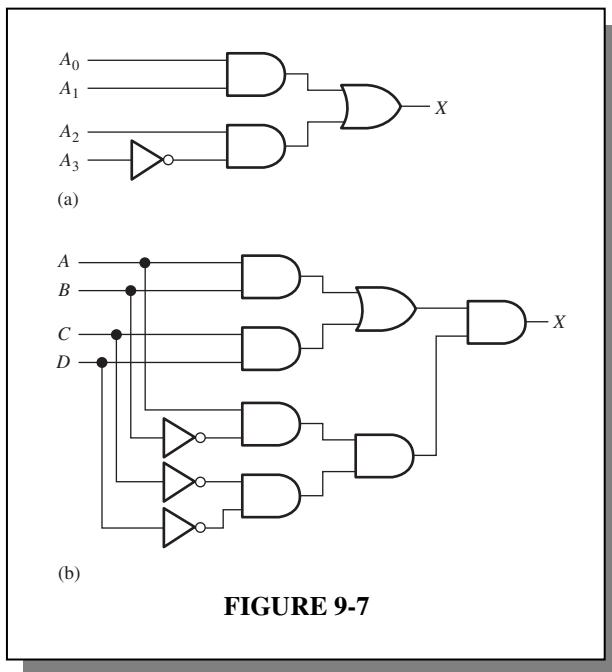
**19.** See Figure 9-6.



$$\begin{aligned} \mathbf{20.} \quad & (A_4 A_3 \bar{A}_2 A_1 + \bar{A}_4 \bar{A}_3 \bar{A}_2 A_1) A_0 + (\bar{A}_5 A_3 A_2 A_1 + A_5 \bar{A}_3 A_2 \bar{A}_1 + A_5 A_3 A_2 \bar{A}_1) A_0 \\ & = A_4 A_3 \bar{A}_2 A_1 A_0 + \bar{A}_4 \bar{A}_3 \bar{A}_2 A_1 A_0 + \bar{A}_5 A_3 A_2 A_1 \bar{A}_0 + A_5 \bar{A}_3 A_2 \bar{A}_1 \bar{A}_0 + A_5 A_3 A_2 \bar{A}_1 \bar{A}_0 \end{aligned}$$

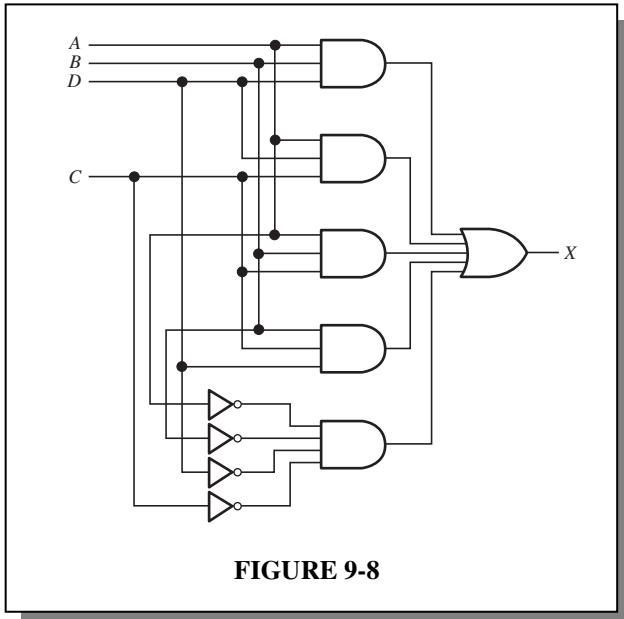
### *Section 9-5 Programmable Logic Software*

**21.** See Figure 9-7.



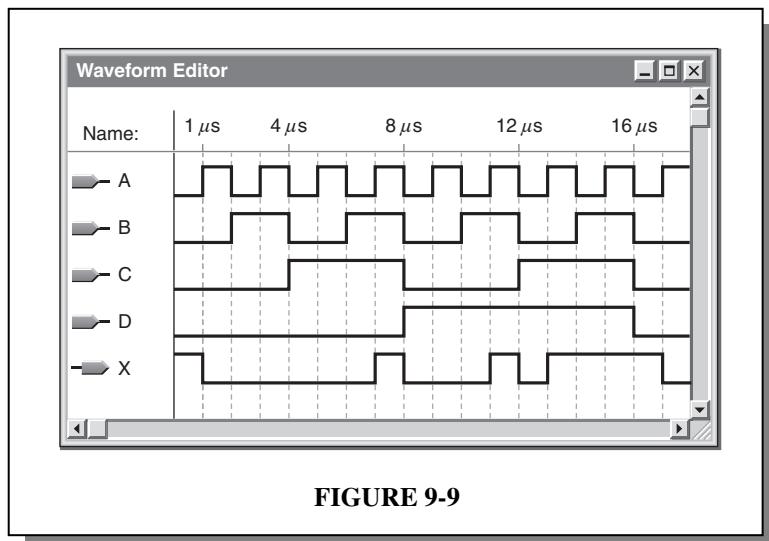
$$\begin{aligned}
 22. \quad X &= \overline{ABCD} + A\overline{BCD} + A\overline{B}\overline{C}D + ABC\overline{D} + ABCD + \overline{A}\overline{B}\overline{C}\overline{D} \\
 &= ABD + ACD + ABC + BCD + \overline{ABC}\overline{D}
 \end{aligned}$$

See Figure 9-8.



**FIGURE 9-8**

23. See Figure 9-9.



**FIGURE 9-9**

## Chapter 9

24.  $X = \overline{ABCD} + A\overline{B}\overline{C}D + ABCD + A\overline{B}C\overline{D} + \overline{A}B\overline{C}D$ . See Figure 9-10.

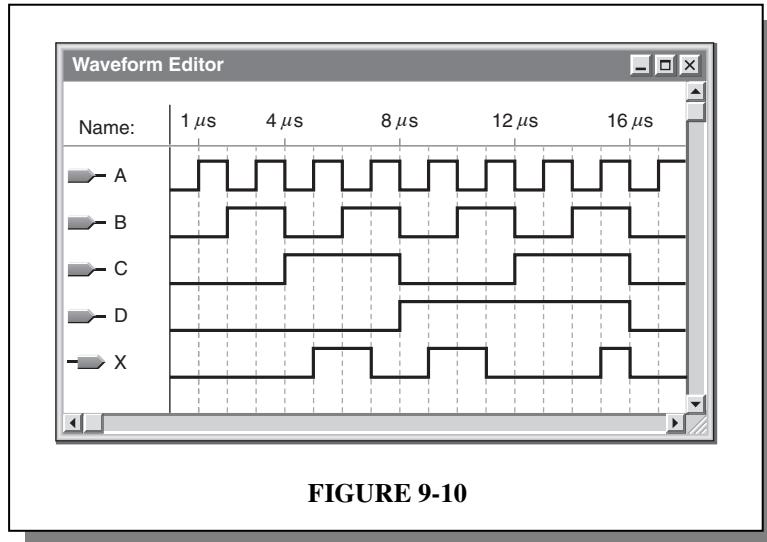


FIGURE 9-10

### Section 9-6 Boundary Scan Logic

25. The Shift input = 1, 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.
26. 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.
27. 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.
28. 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 9-7 Troubleshooting**

- 29.** 000011001010001111011 shifted from TDI to TDO, left-most bit first. The bold-faced code will appear on the logic inputs in the sequence shown.

|    |                              |
|----|------------------------------|
| 0  | <b>000011001010001111011</b> |
| 1  | <b>000011001010001111011</b> |
| 3  | <b>000011001010001111011</b> |
| 6  | <b>000011001010001111011</b> |
| 12 | <b>000011001010001111011</b> |
| 9  | <b>000011001010001111011</b> |
| 2  | <b>000011001010001111011</b> |
| 5  | <b>000011001010001111011</b> |
| 10 | <b>000011001010001111011</b> |
| 4  | <b>000011001010001111011</b> |
| 8  | <b>000011001010001111011</b> |
| 1  | <b>000011001010001111011</b> |
| 3  | <b>000011001010001111011</b> |
| 7  | <b>000011001010001111011</b> |
| 15 | <b>000011001010001111011</b> |
| 14 | <b>000011001010001111011</b> |
| 13 | <b>000011001010001111011</b> |
| 11 | <b>000011001010001111011</b> |

---

## **CHAPTER 10**

### **MEMORY AND STORAGE**

---

#### ***Section 10-1 Memory System Hierarchy***

1. Cache is a small area of fast memory used by the central processing unit. The L1 cache is located in the processor, and the L2 cache is outside of the processor.
2. The main memory generally consists of RAM, which is a volatile memory that temporarily stores less frequently used data and program instructions, and ROM, which is a nonvolatile memory used for permanent storage of frequently used programs and data
3. Hard disk has the highest latency (longer access time).
4.  $150 \text{ PB} (\text{petabytes}) = 150,000,000,000,000,000 = 150 \times 10^{15} \text{ B}$
5. Cache hit: acquisition time = **30 ns**; Cache miss:  $30 \text{ ns} + 75 \text{ ns} = \mathbf{105 \text{ ns}}$
6. Temporal locality refers to a block of data that tends to be referenced soon or frequently.  
Spatial locality refers to a block of data that is referenced and nearby data tend to be referenced soon.

#### ***Section 10-2 Basics of Semiconductor Memory***

7. (a) ROM: no read/write control  
(b) RAM
8. 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.
9. **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.
10. (a)  $0A_{16} = 00001010_2 = \mathbf{10}_{10}$   
(b)  $3F_{16} = 00111111_2 = \mathbf{63}_{10}$   
(c)  $CD_{16} = 11001101_2 = \mathbf{205}_{10}$

### Section 10-3 Random-Access Memories (RAMs)

11.

|       | BIT 0 | BIT 1 | BIT 2 | BIT 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     |

12. See Figure 10-1.

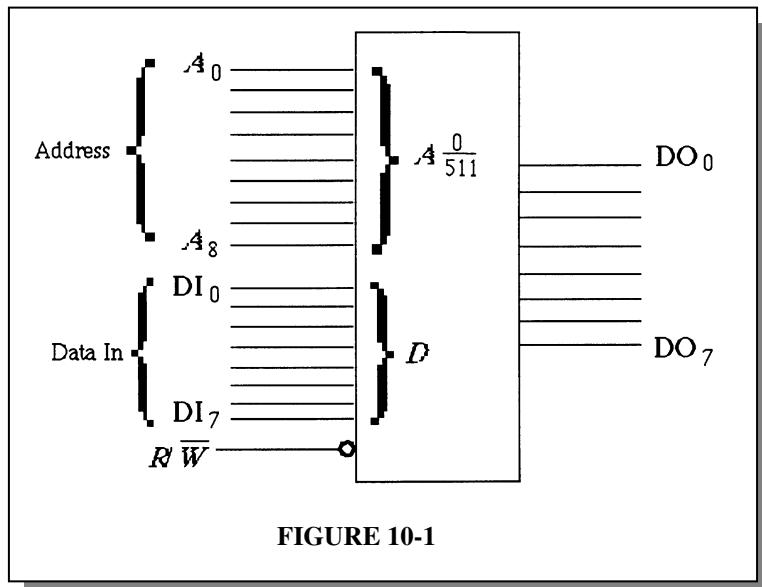


FIGURE 10-1

13.  $64k \times 8 = 512 \times 128 \times 8 = \mathbf{512 \text{ rows} \times 128 \text{ 8-bit columns}}$

## Chapter 10

14. See Figure 10-2.

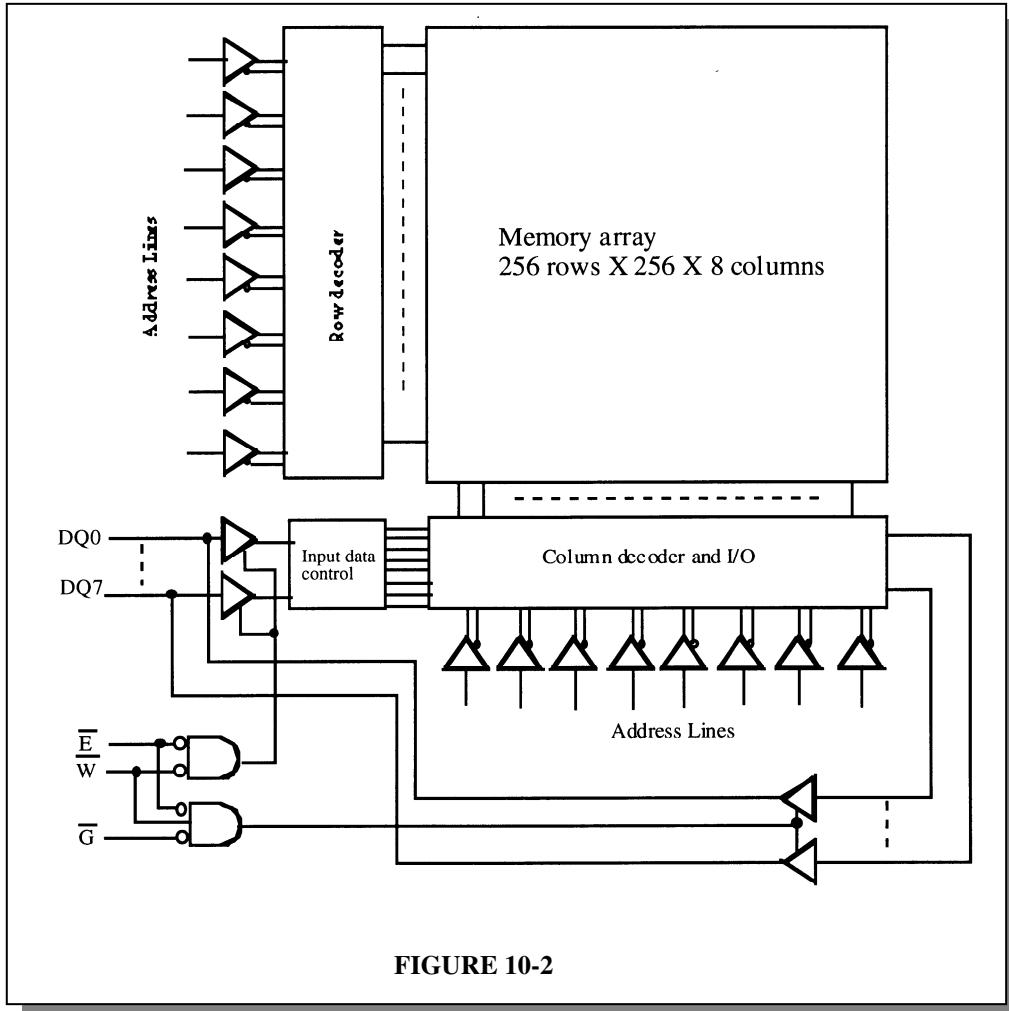


FIGURE 10-2

15. 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.

16. 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 10-4 Read-Only Memories (ROMs)

17.

| Inputs |       | Outputs |       |       |       |
|--------|-------|---------|-------|-------|-------|
| $A_1$  | $A_0$ | $O_3$   | $O_2$ | $O_1$ | $O_0$ |
| 0      | 0     | 0       | 1     | 0     | 1     |
| 0      | 1     | 1       | 0     | 0     | 1     |
| 1      | 0     | 1       | 1     | 1     | 0     |
| 1      | 1     | 0       | 0     | 1     | 0     |

18.

| Inputs |       |       | Outputs |       |       |       |
|--------|-------|-------|---------|-------|-------|-------|
| $A_2$  | $A_1$ | $A_0$ | $O_3$   | $O_2$ | $O_1$ | $O_0$ |
| 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     |

19.

| BCD   |       |       |       | Excess-3 |       |       |       |
|-------|-------|-------|-------|----------|-------|-------|-------|
| $D_3$ | $D_2$ | $D_1$ | $D_0$ | $E_3$    | $E_2$ | $E_1$ | $E_0$ |
| 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.

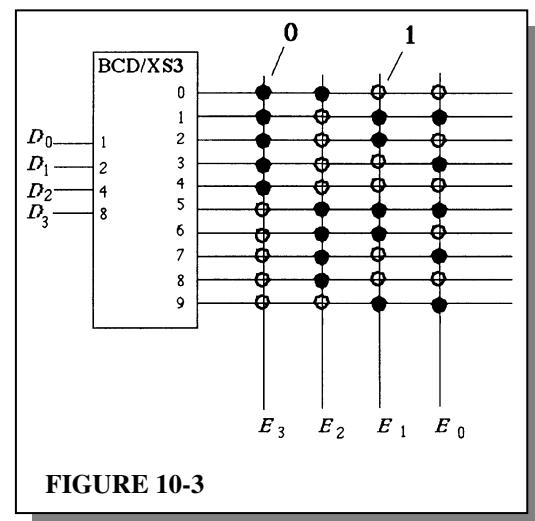


FIGURE 10-3

20.  $2^{14} = 16,384$  addresses  
 $16,384 \times 8 \text{ bits} = 131,072 \text{ bits}$

## Chapter 10

### Section 10-5 Programmable ROMs

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

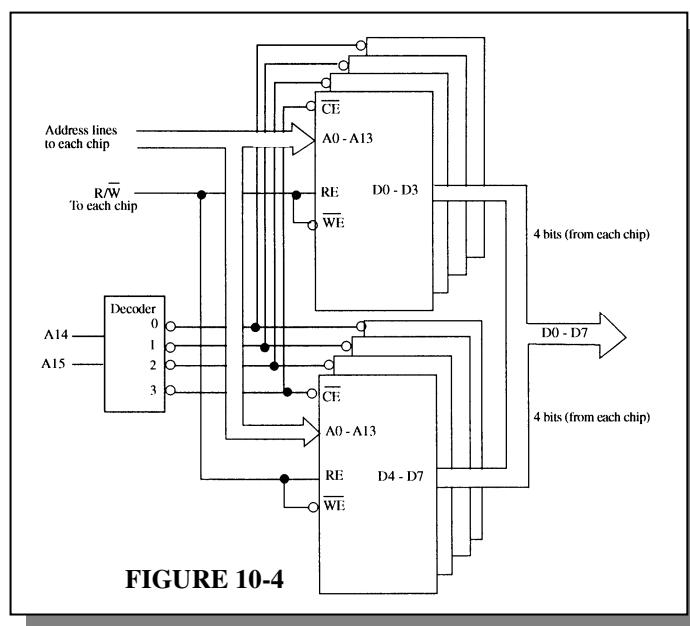
| X Input |       |       | $X^3$ | X Output |       |       |       |       |       |       |       |       |
|---------|-------|-------|-------|----------|-------|-------|-------|-------|-------|-------|-------|-------|
| $X_2$   | $X_1$ | $X_0$ |       | $2^8$    | $2^7$ | $2^6$ | $2^5$ | $2^4$ | $2^3$ | $2^2$ | $2^1$ | $2^0$ |
| 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     |

22.

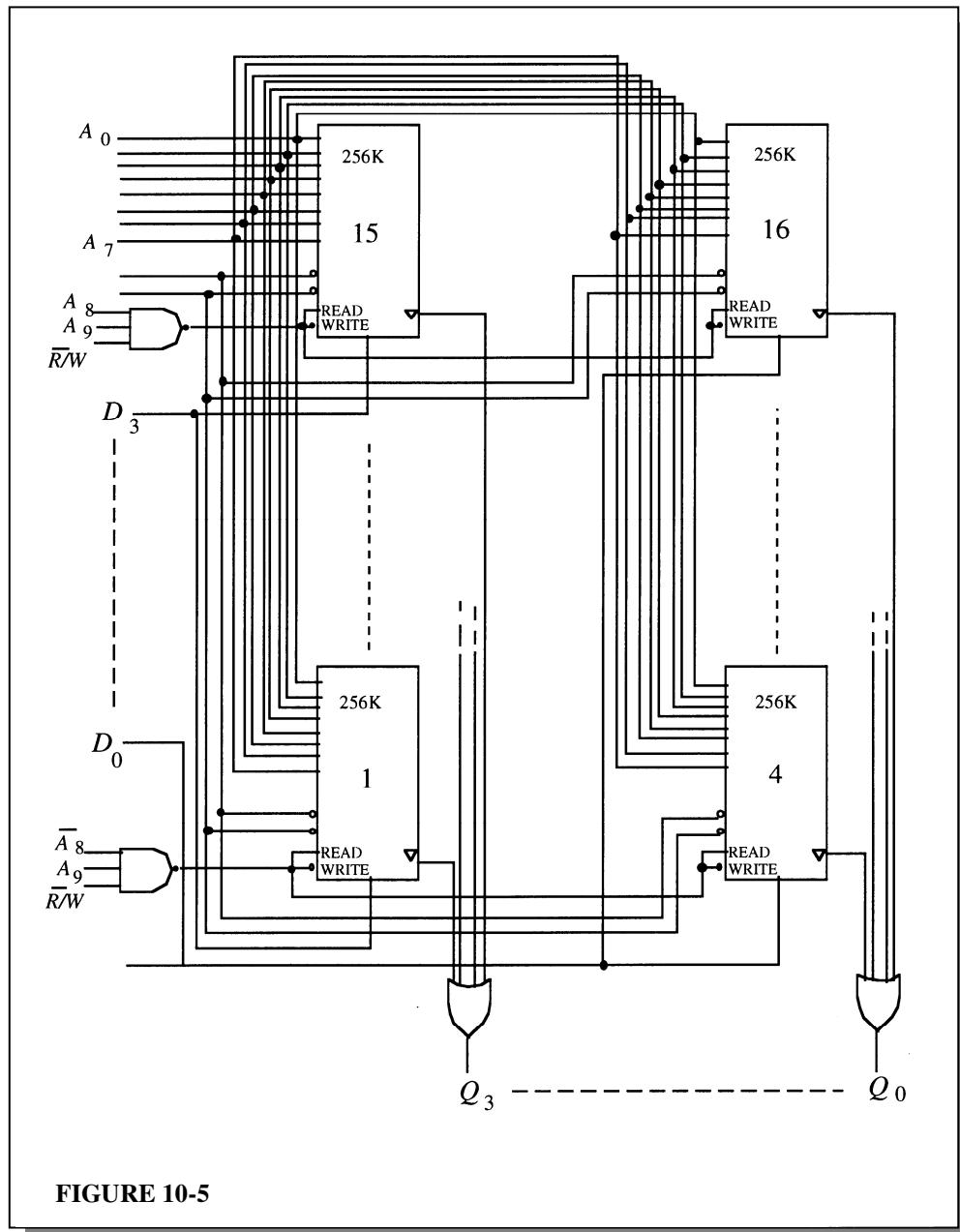
| Address             | Contents         |
|---------------------|------------------|
| $A_{13} \cdots A_0$ | $Q_7 \cdots Q_0$ |
| 01001100010011      | 10101100         |
| 11011101011010      | 00100101         |
| 01011010011001      | 10110011         |
| 11010010001110      | 00101000         |
| 01010010100101      | 10001011         |
| 01010000110100      | 11010101         |
| 01001001100001      | 11001001         |
| 11011011100100      | 01001001         |
| 01101110001111      | 01010010         |
| 10111110011010      | 01001000         |
| 10101110011010      | 11001000         |

### Section 10-7 Memory Expansion

23. 16k  $\times$  4 DRAMs can be connected to make a 64k  $\times$  8 DRAM as shown in Figure 10-4.



24. See Figure 10-5.



**FIGURE 10-5**

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

## Chapter 10

### Section 10-8 Special Types of Memories

26. See Figure 10-6.

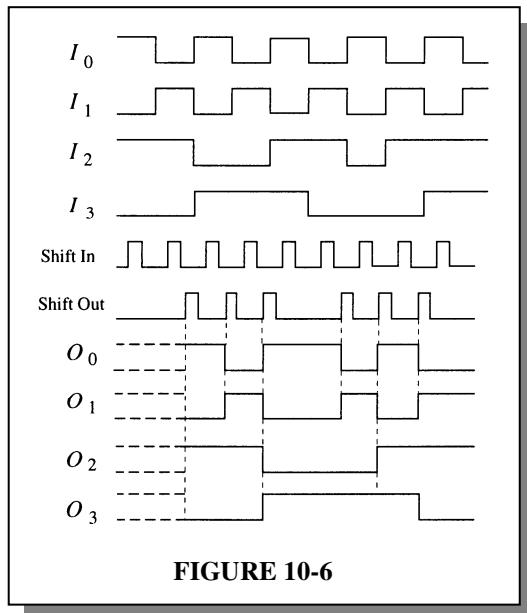


FIGURE 10-6

27. See Figure 10-7. Addresses are in hexadecimal.

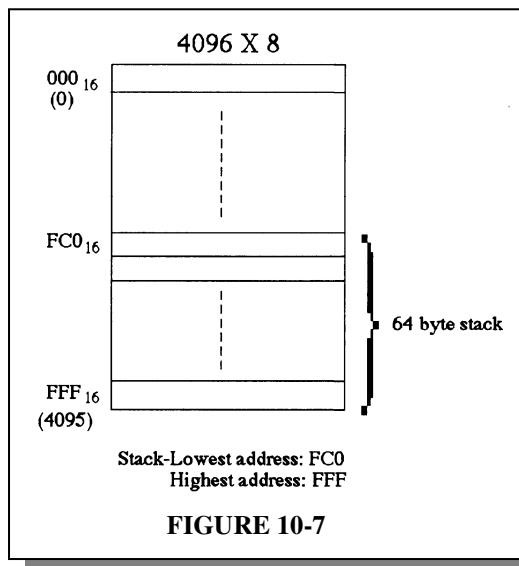
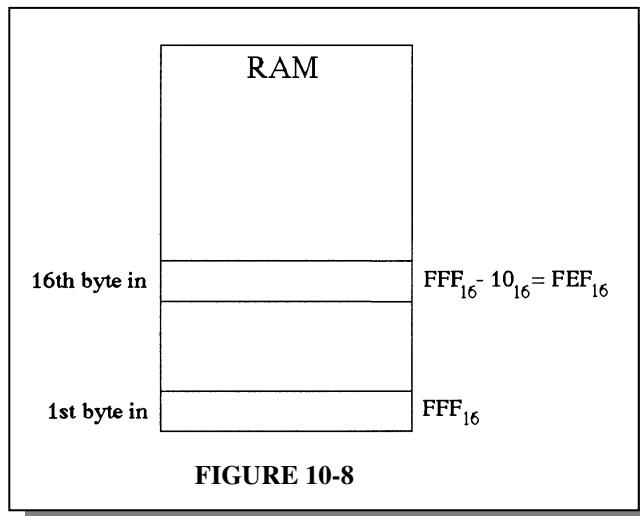


FIGURE 10-7

28. 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 10-9 Magnetic and Optical Storage**

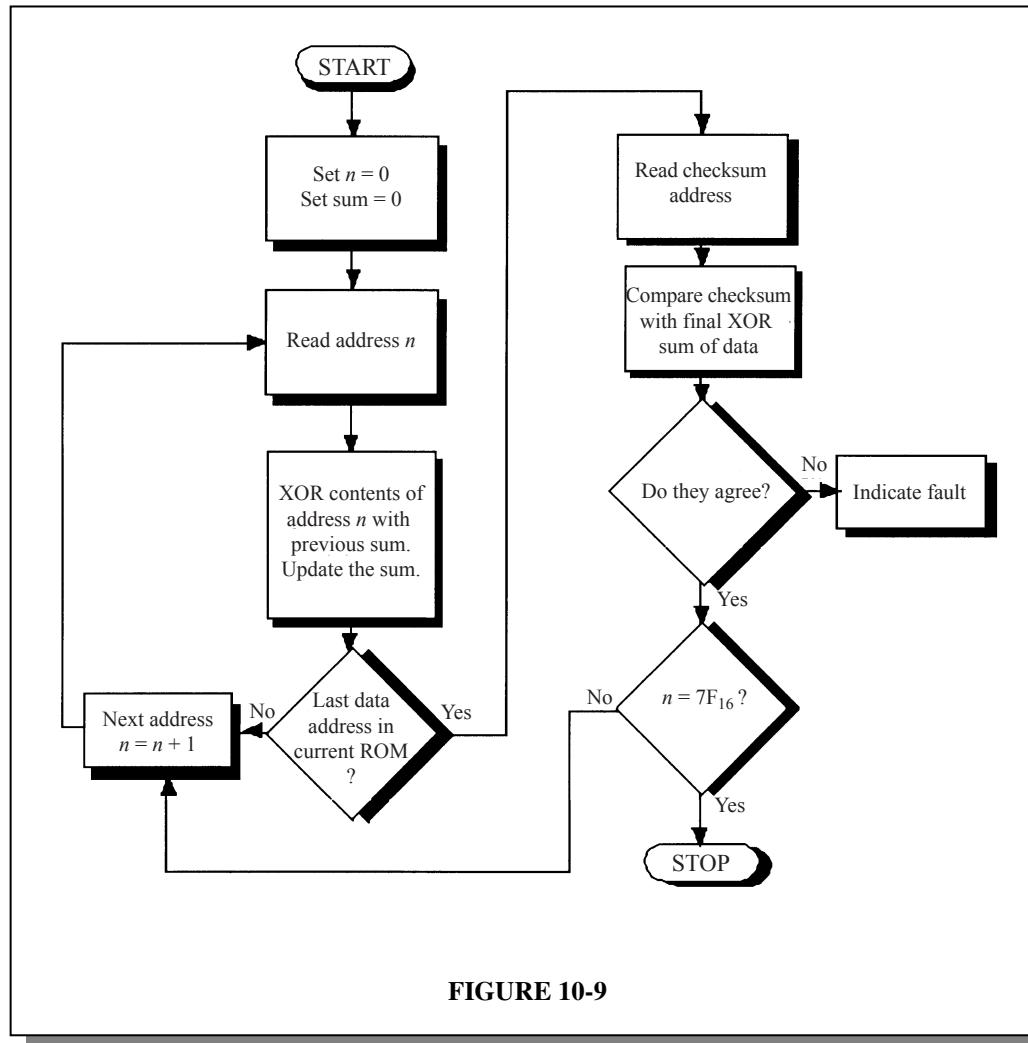
29. A hard disk is formatted into tracks and sectors. Each track is divided into a number of sectors with each sector of a track having a physical address. Hard disks typically have from a few hundred to a few thousand tracks.
30. Seek time is the average time required to position the drive head over the track containing the desired data. The latency period is the average time required for the data to move under the drive head.
31. Magnetic tape has a longer access time than disk because data must be accessed sequentially rather than randomly.
32. 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 10-10 Troubleshooting**

33. The correct checksum is **00100**.  
 The actual checksum is 01100. The second bit from the left is in error.
34. (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 -  $7F_{16}$

## Chapter 10

- (b) Same as flow chart in Figure 10-71 in text except that the last data address is specified as  $7F_{16}$ .
- (c) See 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.
- 35.**
- (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

---

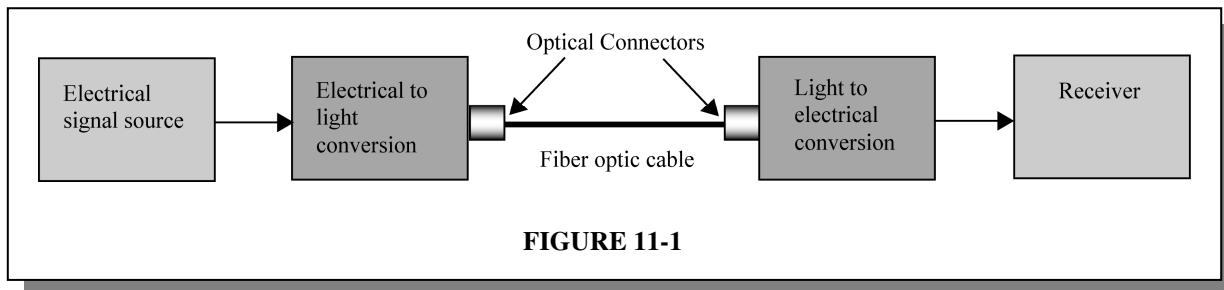
## CHAPTER 11

# DATA TRANSMISSION

---

### Section 11-1 Data Transmission Media

1. A coaxial cable consists of an outer jacket, metallic shield, dielectric, and center conductor.
2. Types of twisted pair cable: UTP: unshielded tested pair; STP: shielded twisted pair
3. Advantages of fiber optics over electrical transmission media are faster data rates, higher signal capacity (more signals at a time), transmission over longer distances, and not susceptible to EMI.
4. Three parts of an optical fiber are the core through which the light travels, the highly reflective cladding surrounding the core, and the protective jacket around the cladding.
5. In multimode, the light entering the optical fiber will tend to propagate through the core in multiple rays (modes), basically due to varying angles as each light ray moves along.
6. The 50/125 and 62.5/125 are multimode fibers. The 8.3/125 is a single-mode fiber.
7. See Figure 11-1.



8. A frequency of 100 MHz is a radio wave.
9. Visible light is in the frequency range  $4 \times 10^{14}$  Hz to  $7.5 \times 10^{14}$  Hz. Figure 11-10 in the textbook shows the range in units of wavelength (nm).

### Section 11-2 Methods and Modes of Data Transmission

10. Number of bits =  $(1 \times 10^6 \text{ bits/s})(1 \times 10^{-3} \text{ s}) = \mathbf{1000 \text{ bits}}$
11.  $\frac{8 \text{ bits}}{1 \mu\text{s}} = \mathbf{8 \text{ Mbps}}$

## Chapter 11

12. Bit rate =  $\frac{(3 \text{ bits/symbol})(12 \text{ symbols})}{0.5 \mu\text{s}} = \frac{36 \text{ bits}}{0.5 \mu\text{s}} = 72 \text{ Mbps}$

$$\text{Baud} = \frac{72 \text{ Mbps}}{(3 \text{ bits/symbol})} = 24 \text{ Mbaud}$$

13. Baud =  $\frac{25 \text{ Mbps}}{(5 \text{ bits/symbol})} = 5 \text{ Mbaud}$

14. Efficiency =  $\frac{\text{Data bits}}{\text{data bits}} = \frac{16}{20} = 0.8$

15. See Figure 11-2.

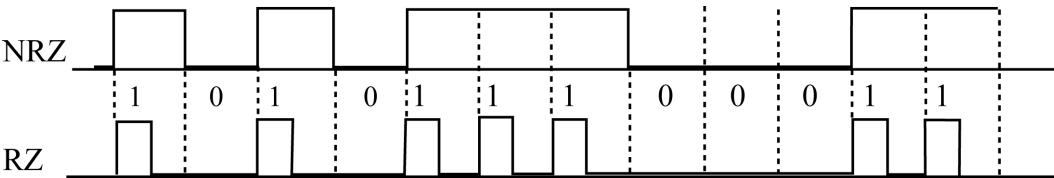


FIGURE 11-2

16. See Figure 11-3.

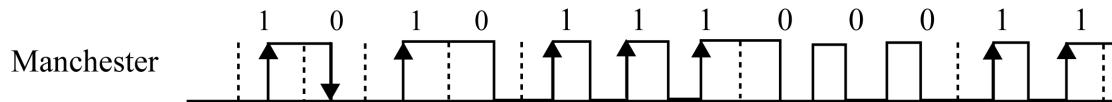


FIGURE 11-3

17. The bit sequence represented by the Manchester code is **001110011**.

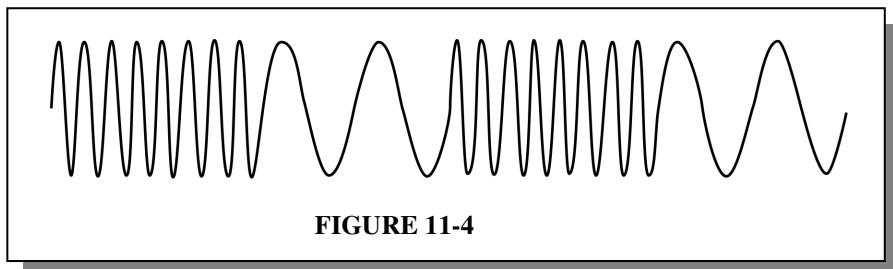
- *Preamble:* A group of bits at the beginning of a frame that is used to alert the receiver that a new frame has arrived and to synchronize the receiver's clock with the transmitted clock.
- *Address Fields:* A group of bits containing the address(s) of the sender and the receiver. One or both addresses may be present in a given protocol.
- *Control Field:* This group of bits identifies the type of data being sent, such as handshaking (establishes a connection), file transfers, and the size of the data.
- *Data Field:* This sequence is the actual information being sent and can be of a fixed length or a variable length. If it is a fixed-length field, a group of bits called a *pad* is used to fill in if the actual data field is less than the fixed field.

- *Frame Check:* This field contains an error check such as parity, CRC (cyclic redundancy check), or checksum, which is a value computed by a simple algorithm of the data bits in the frame.
- *End Frame:* A group of bits that tells the receiver when the end of the frame occurs.

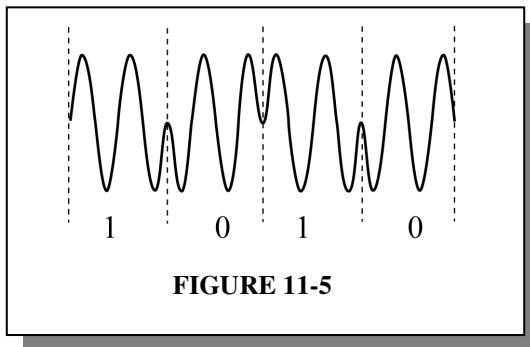
### **Section 11-3 Modulation of Analog Signals with Digital Data**

**19.** The binary code represented by the ASK signal is **11010111110001000001**.

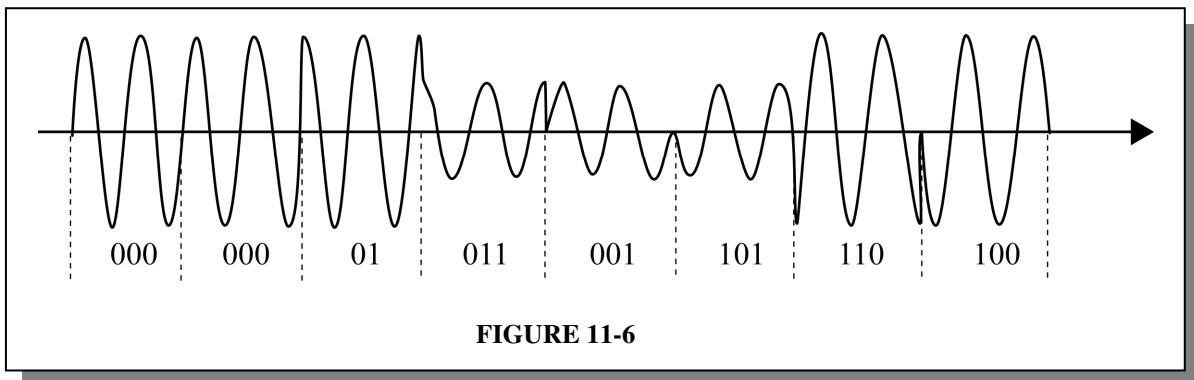
**20.** See Figure 11-4. FSK represents 1010.



**21.** See Figure 11-5.

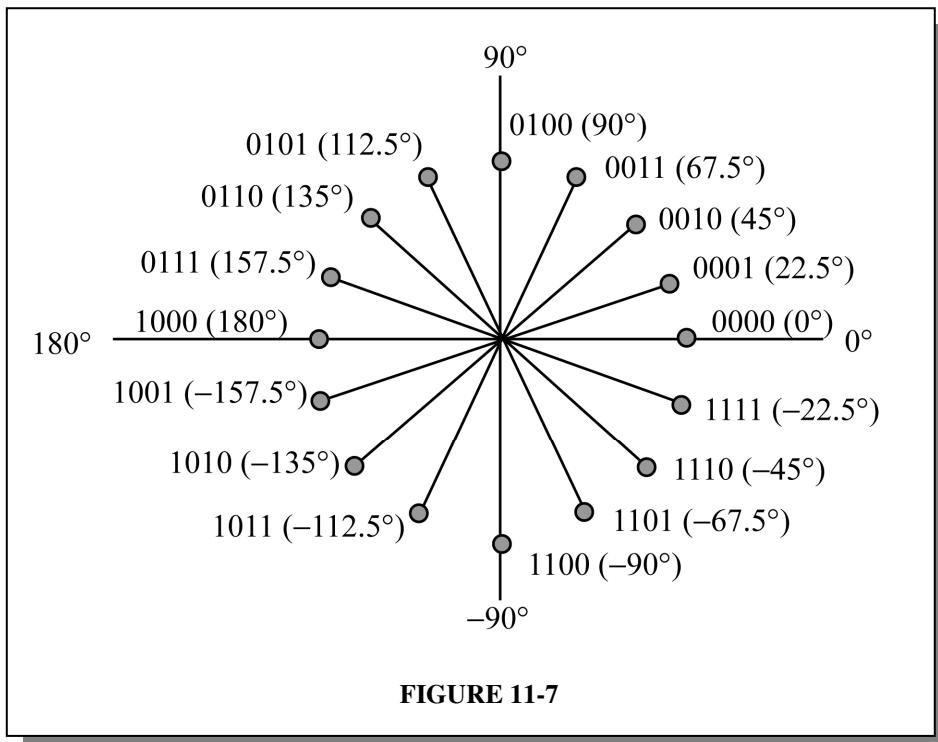


**22.** See Figure 11-6.

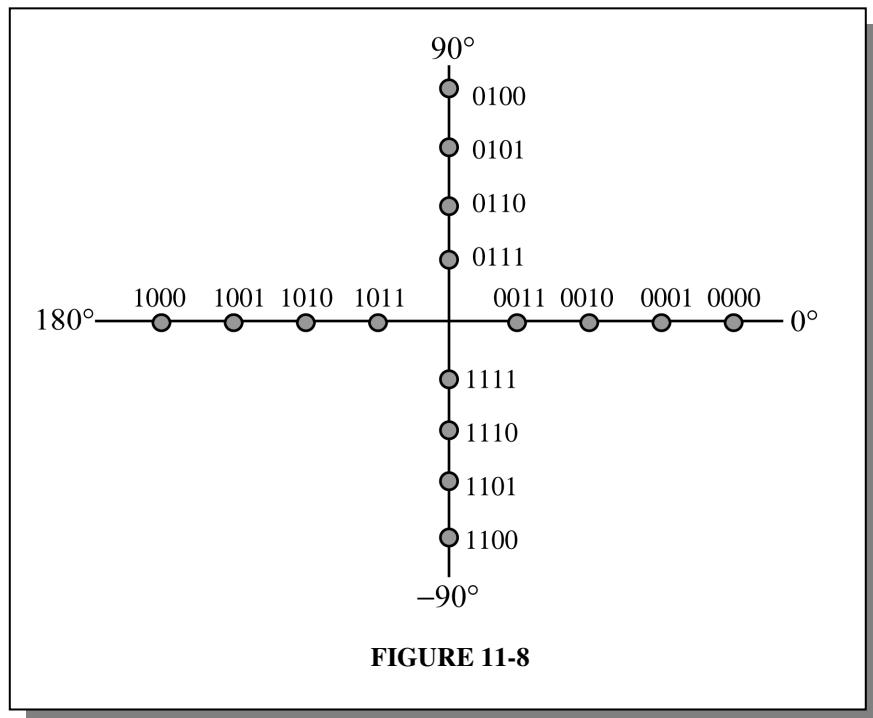


## Chapter 11

23. See Figure 11-7.

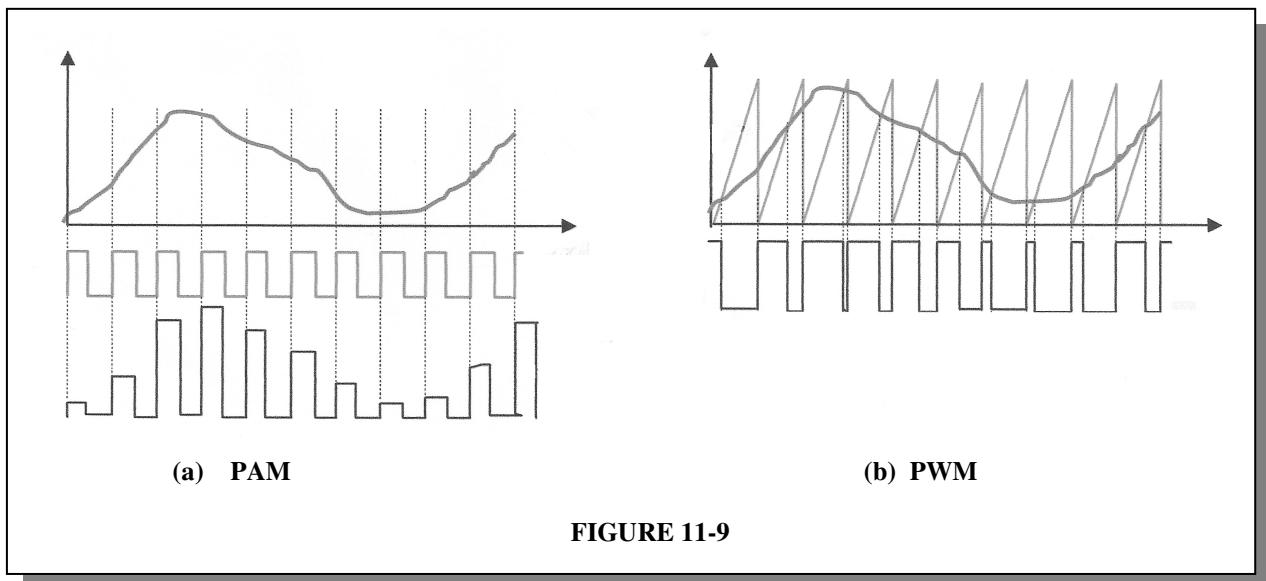


24. Four phases and four amplitudes can be used as shown in Figure 11-8.



### **Section 11-4 Modulation of Digital Signals with Analog Data**

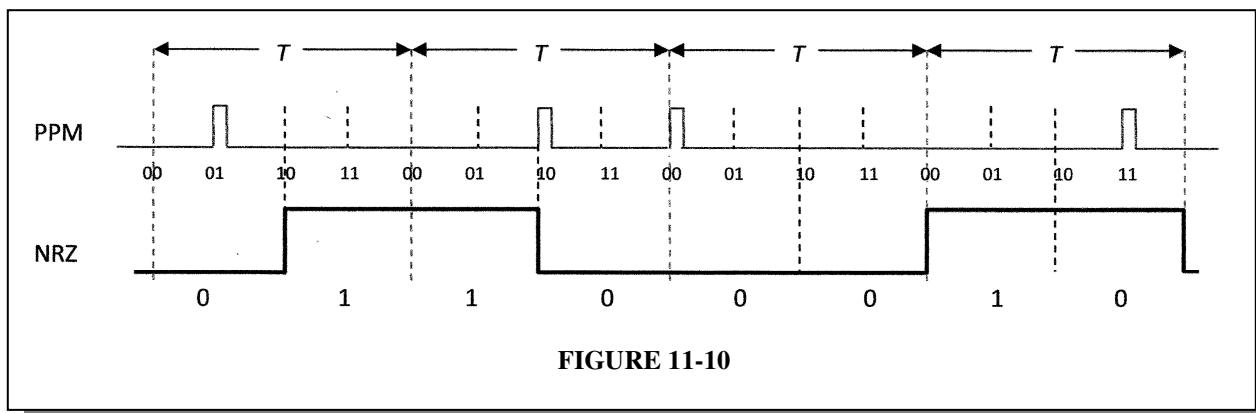
25. In the PWM intersective method, the sawtooth intersects the sinusoidal modulating signal twice during each cycle. The sawtooth is either increasing above the sine wave or decreasing below the sine wave. When the sawtooth is increasing above the sine wave, a low level is generated, and when it is decreasing below the sine wave, a high level is generated. The resulting output is a series of pulses with widths proportional to the amplitude of the sine wave.
26. See Figure 11-9.



27. It takes three bits to describe each of the eight positions.

$$\text{Data rate } \frac{3 \text{ bits}}{10 \text{ ms}} = 300 \text{ bps}$$

28. The bit sequence of the PPM signal is 01100011. See Figure 11-10.



29.  $16 = 2^4$ . Four bits are required to represent 16 voltage levels.

## Chapter 11

30. The sequence of bits is 0000001101010111. The NRZ code is shown in Figure 11-11.

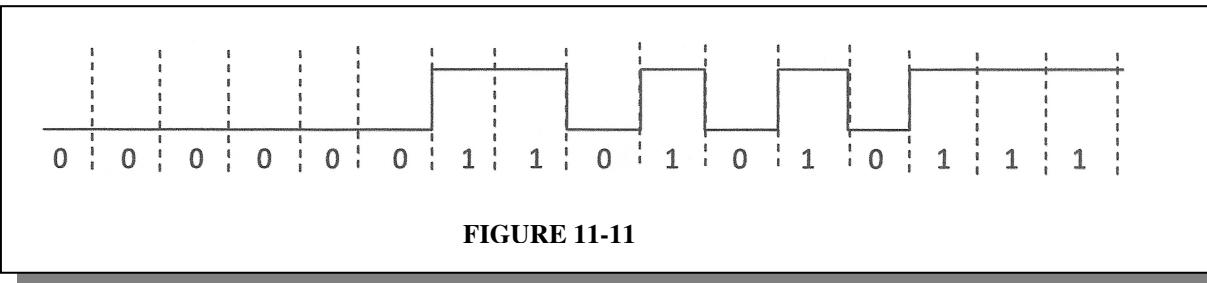


FIGURE 11-11

### Section 11-5 Multiplexing and Demultiplexing

31. Bit interleaved: A single data bit from a source is transmitted on the channel, followed by a data bit from another source, and so on.

Byte interleaved: A byte of data from a source is transmitted on the channel, followed by a byte from another source, and so on.

32. In synchronous TDM, the time slots allotted to each source are fixed, and each time slot is transmitted whether or not the source has data to send. In statistical TDM, the time slot assignments are variable so that only data from active sources are transmitted and no blank time slots are transmitted.

33. In FDM, band-pass filters are used on the receiving end to separate the transmitted signals.
34. The frequency separation between each source in FDM is called the **guard band**.

### Section 11-6 Effects of Transmission Media on Data Quality

35.  $R_{\text{cond}} = \left( \frac{25.67 \Omega}{1000 \text{ ft}} \right) 500 \text{ ft} = 12.84 \Omega$

$$I = \frac{V_s}{R_{\text{load}} + R_{\text{cond}}} = \frac{3 \text{ V}}{612.84 \Omega} = 4.9 \text{ mA}$$

$$A = 1 - \left( \frac{V_s - IR_{\text{cond}}}{V_s} \right) = 1 - \left( \frac{3 \text{ V} - (4.9 \text{ mA})(12.84 \Omega)}{3 \text{ V}} \right) = 0.021$$

$$V_{\text{rec}} = V_s - AV_s = 3 \text{ V} - 0.21(3 \text{ V}) = 2.94 \text{ V}$$

36. Attenuation =  $1 - \left( \frac{\text{Output intensity}}{\text{Input intensity}} \right) = 1 - \left( \frac{92 \text{ lumens}}{100 \text{ lumens}} \right) = 1 - 0.92 = 0.08$

**37.**  $\lambda = c / f = (299,792,458 \text{ m/s})/250 \text{ mHz} = 1.2 \text{ m}$

$$A = 10 \log \left( \frac{4\pi d}{\lambda} \right)^2 = 20 \log \left( \frac{4\pi \cdot 50 \text{ m}}{1.2 \text{ m}} \right) = \mathbf{54.4 \text{ dB}}$$

**38.**  $S/N = 20 \log \left( \frac{V_s}{V_n} \right) = 20 \log \left( \frac{900 \mu V}{100 \mu V} \right) = \mathbf{19.1 \text{ dB}}$

**39.**  $S/N = 20 \log \left( \frac{V_s}{V_n} \right) = 20 \log \left( \frac{100 \mu V}{110 \mu V} \right) = \mathbf{-0.828}$

**40.** Noise margin =  $S_T(\text{dB}) - S_m(\text{dB}) = 20 \text{ dB} - 12 \text{ dB} = \mathbf{8 \text{ dB}}$

---

## CHAPTER 12

### SIGNAL CONVERSION AND PROCESSING

---

#### Section 12-1 A System

1.  $\frac{\Delta V}{2^4} = \frac{5 \text{ V} - 1 \text{ V}}{16} = 0.25 \text{ V}$

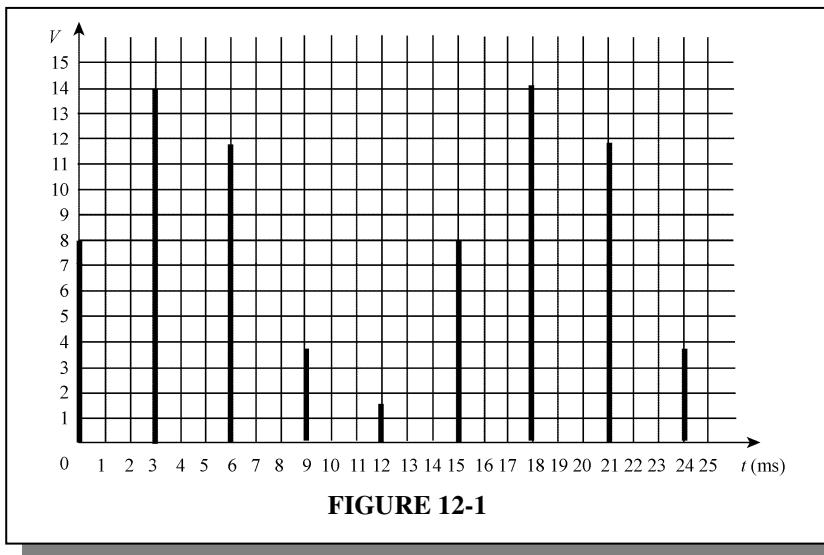
2. 1001 represents  $9(0.25 \text{ V}) = 2.25 \text{ V}$

3.  $\frac{18^\circ}{1.5^\circ} = 12 \text{ cycles}$

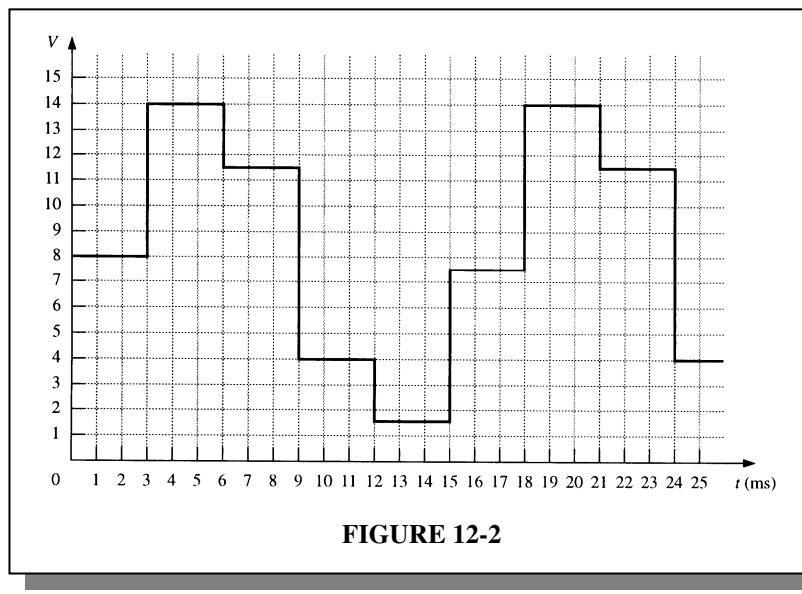
- 4.
- *Sunlight sensors:* The photoresistors sense the sunlight intensity and provide proportional voltage outputs.
  - *ADCs (analog-to-digital converters):* These units convert the voltages from the sunlight sensors to a digital code.
  - *A/D processing:* This block receives the digitized light level inputs from the ADCs and compares the two values to determine the required direction of rotation.
  - *Frequency divider:* This divides (reduces) the system clock frequency to a frequency compatible with the rate at which the stepper motor can be advanced.
  - *Stepper motor control:* This unit accepts the output of the A/D processing block and produces an output that tells the motor which way to turn.
  - *Interface:* This block provides interfacing between the motor control and the stepper motor.

#### Section 12-2 Converting Analog Signals to Digital

5. See Figure 12-1.



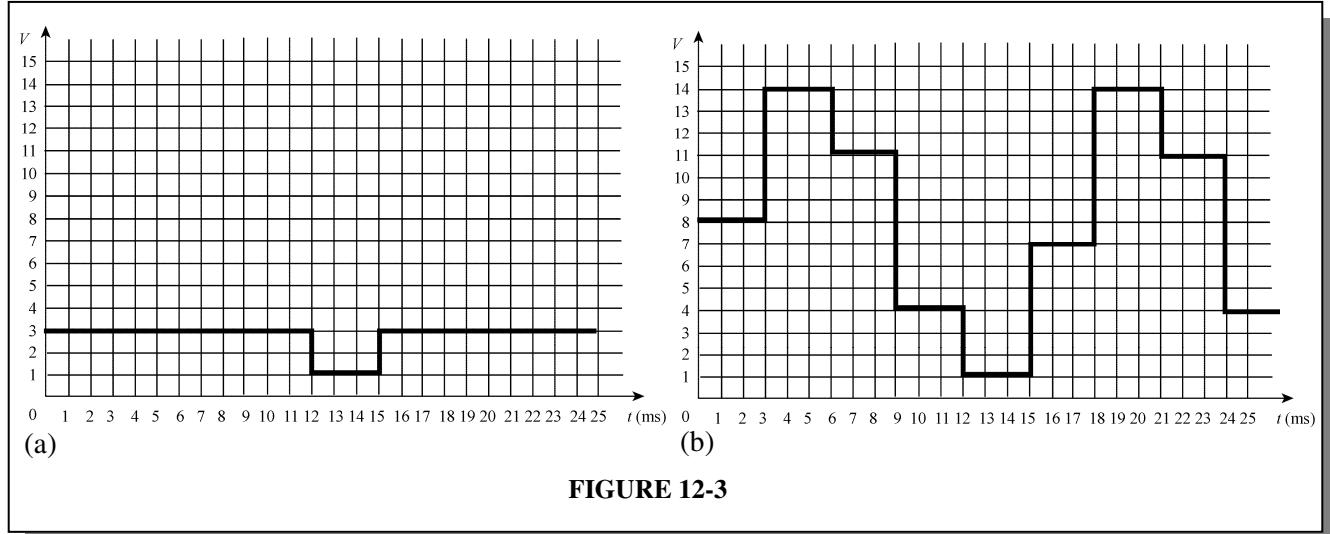
6. See Figure 12-2.



7.  $11, 11, 11, 11, 01, 11, 11, 11, 11$

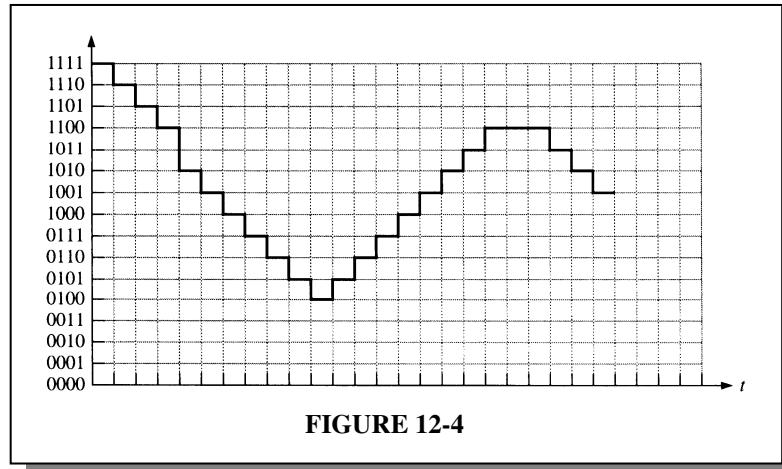
8.  $1000, 1110, 1011, 0100, 0001, 0111, 1110, 1011, 0100$

9. See Figure 12-3.



## Chapter 12

10. See Figure 12-4.



### Section 12-3 Analog-to-Digital Conversion Methods

$$11. \quad \frac{V_{\text{out}}}{V_{\text{in}}} = \frac{2 \text{ V}}{10 \text{ mV}} = 200$$

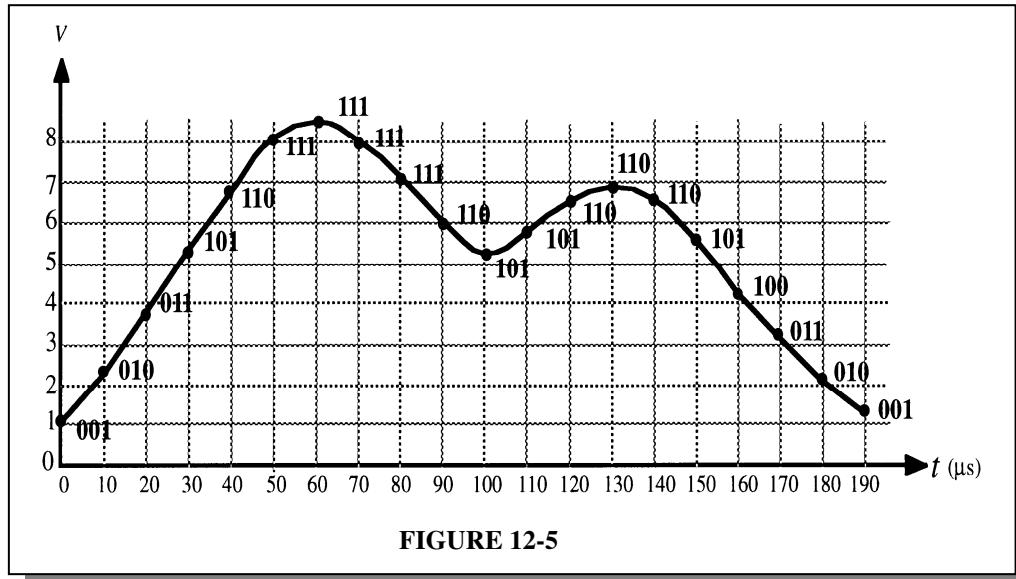
$$12. \quad \frac{V_{\text{OUT}}}{V_{\text{IN}}} = \frac{R_F}{R_{\text{IN}}}$$
$$R_F = R_{\text{in}} \left( \frac{V_{\text{OUT}}}{V_{\text{IN}}} \right) = 1 \text{ k}\Omega (330) = 330 \text{ k}\Omega$$

$$13. \quad A_v = \frac{R_f}{R_i} = \frac{47 \text{ k}\Omega}{2.2 \text{ k}\Omega} = -21.4$$

$$14. \quad \text{Number of comparators} = 2^n - 1 = 2^8 - 1 = 255$$

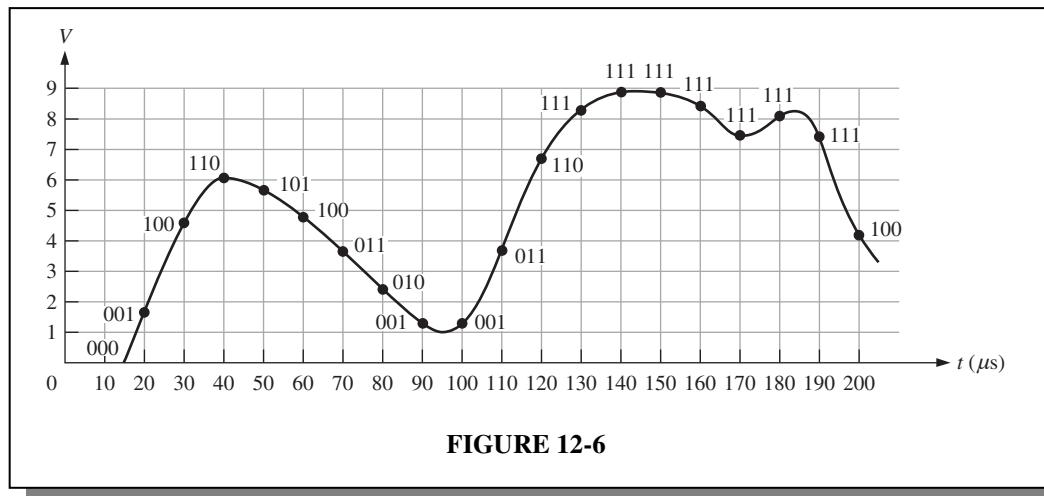
15. 001, 010, 011, 101, 110, 111, 111, 111, 110, 101, 101, 110, 110, 110, 101, 100, 011, 010, 001.

See Figure 12-5.



16. Output of 3-bit converter: 000, 001, 100, 110, 101, 100, 011, 010, 001, 001, 011, 110, 111, 111, 111, 111, 111, 111, 100.

See Figure 12-6.



## Chapter 12

17.

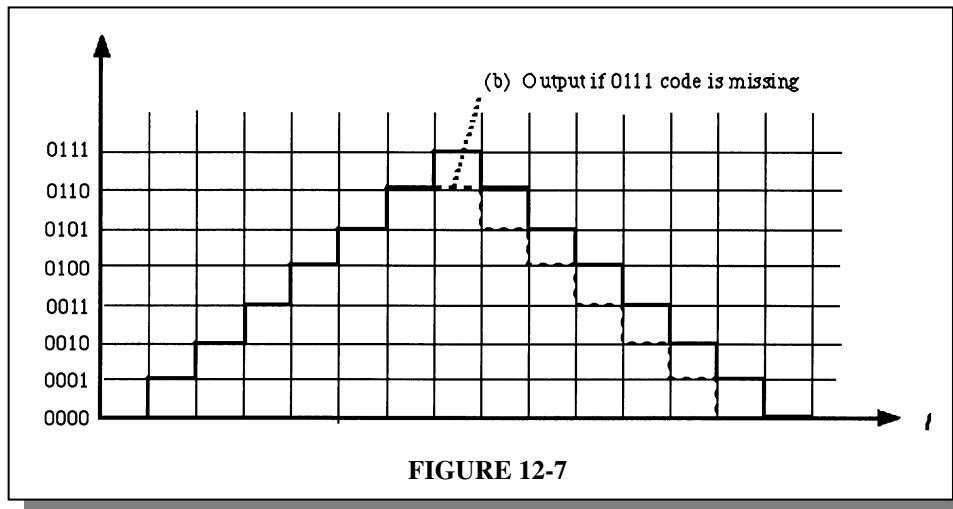
| SAR | Comment                          |
|-----|----------------------------------|
| 11  | Less than $V_{in}$ . Keep the 1. |
| 11  | Less than $V_{in}$ . Keep the 1. |
| 11  | Less than $V_{in}$ . Keep the 1. |

Conversion never terminates since 2 bits cannot represent the input.

18.

| SAR  | Comment                                      |
|------|----------------------------------------------|
| 1000 | Greater than $V_{in}$ . Reset MSB.           |
| 0100 | Less than $V_{in}$ . Keep the 1.             |
| 0110 | Equal to $V_{in}$ . Keep the 1 (final state) |

19. See Figure 12-7.



### Section 12-4 Digital-to-Analog Conversion Methods

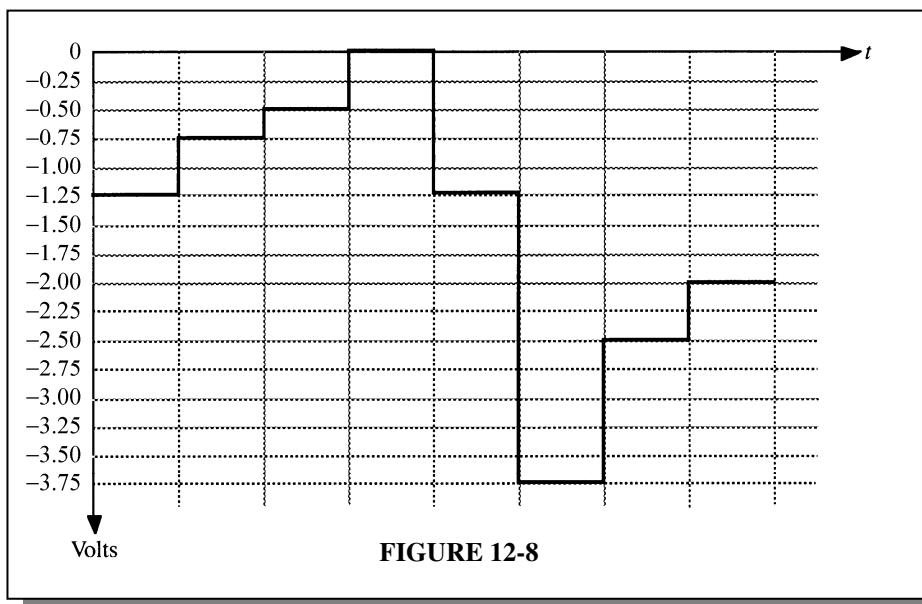
20.  $R_0 = 10 \text{ k}\Omega$

$$R_1 = \frac{R_0}{2} = \frac{10 \text{ k}\Omega}{2} = 5 \text{ k}\Omega$$

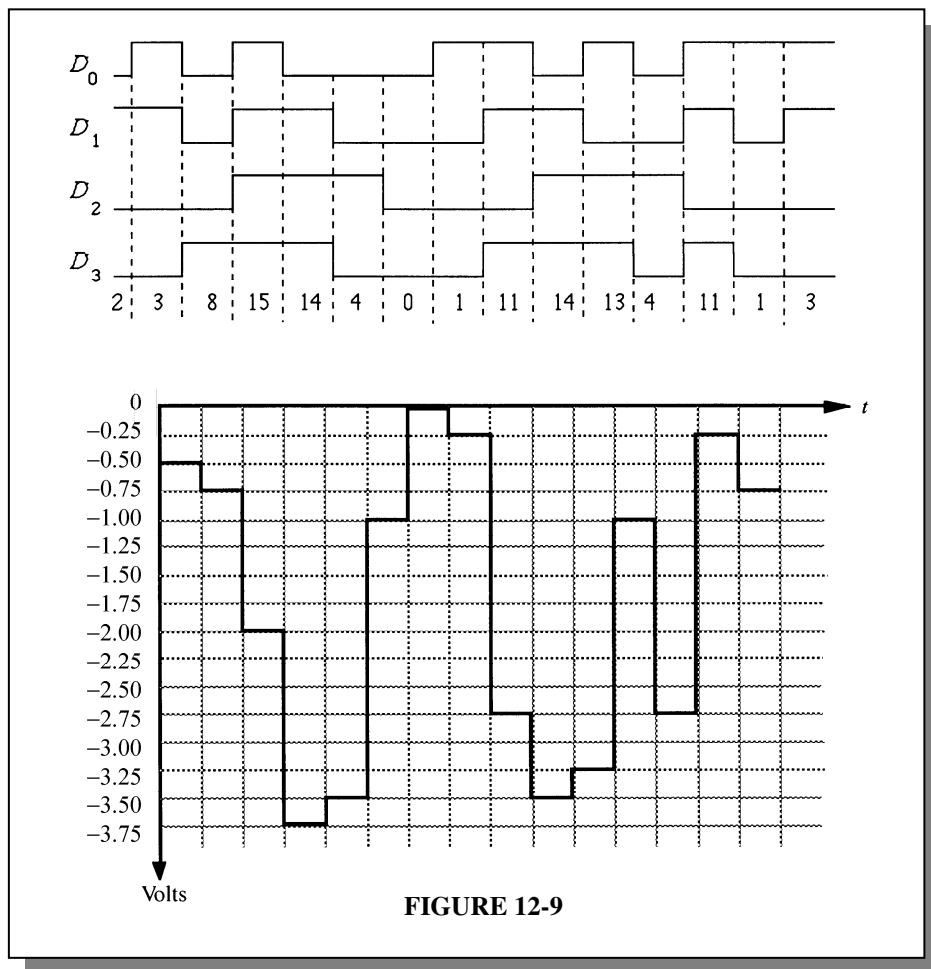
$$R_2 = \frac{R_0}{4} = \frac{10 \text{ k}\Omega}{4} = 2.5 \text{ k}\Omega$$

$$R_3 = \frac{R_0}{8} = \frac{10 \text{ k}\Omega}{8} = 1.25 \text{ k}\Omega$$

**21.** See Figure 12-8.



**22.** See Figure 12-9.



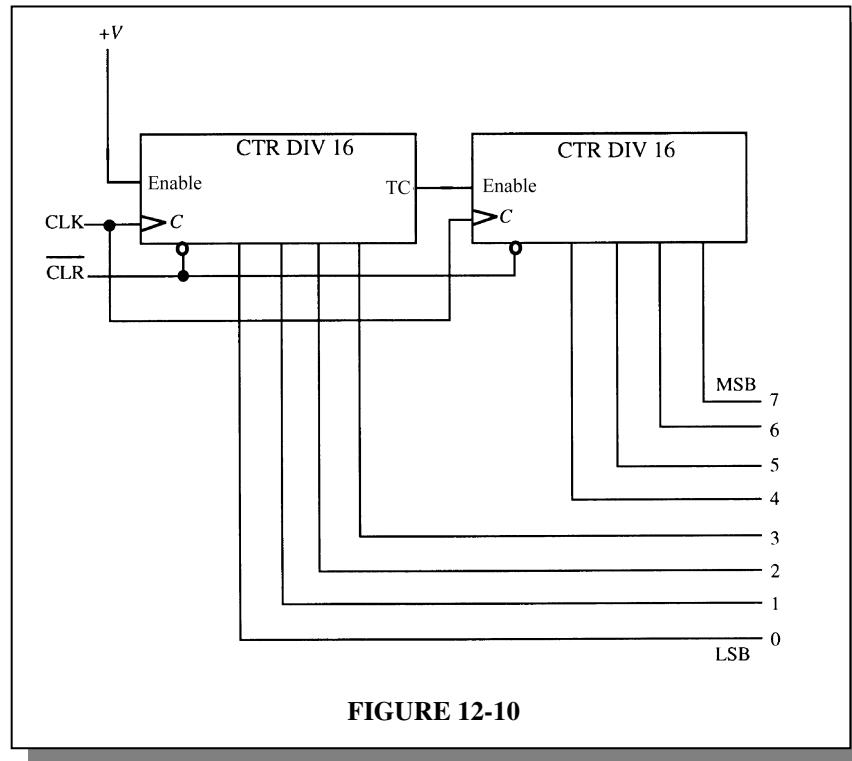
## Chapter 12

23. (a)  $\left(\frac{1}{(2^3-1)}\right)100 = 14.3\%$

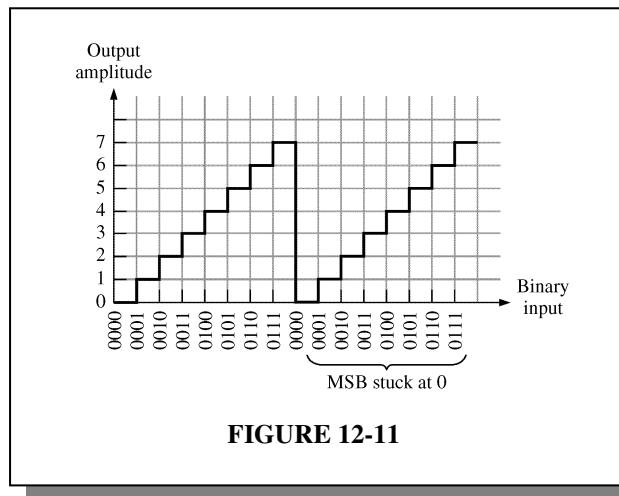
(b)  $\left(\frac{1}{2^{10}-1}\right)100 = 0.098\%$

(c)  $\left(\frac{1}{2^{18}-1}\right)100 = 0.00038\%$

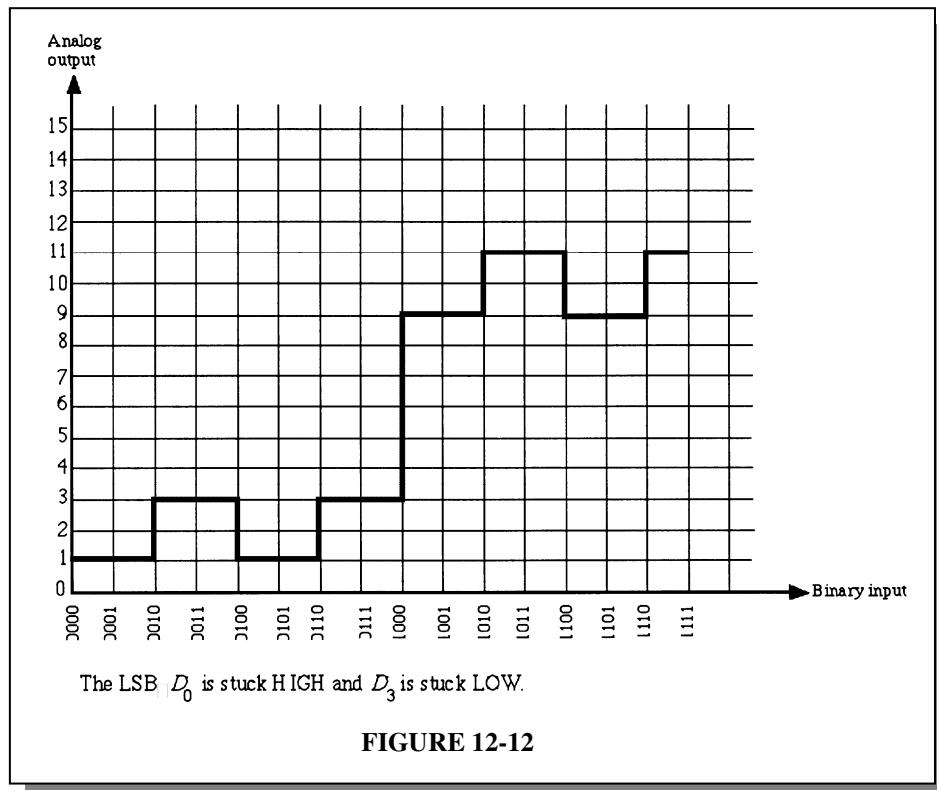
24. See Figure 12-10.



25. See Figure 12-11.

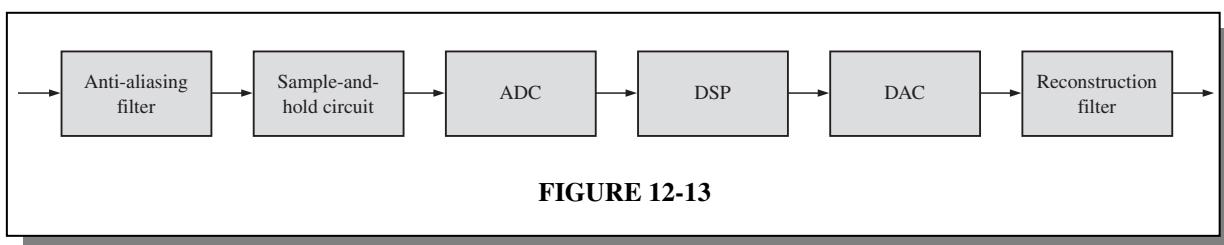


- 26.** See Figure 12-12.



### **Section 12-5 Digital Signal Processing**

- 27.** 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.
- 28.** See Figure 12-13.



- 29.** The purpose of digital-to-analog conversion is to change a sequence of digital codes into an analog signal represented by the digital codes.

## ***Chapter 12***

### ***Section 12-6 The Digital Signal Processor (DSP)***

**30.**  $2000 \text{ MIPS} \times \frac{32 \text{ bit/instruction}}{8 \text{ bits/byte}}$

=  $2000 \text{ MIPS} \times 4 \text{ bytes/instruction}$

=  $8000 \text{ Mbytes/s}$

**31.**  $\frac{400 \text{ Mbits/s}}{32 \text{ bits/instruction}} = 12.5 \text{ million instructions/s}$

**32.**  $1000 \text{ MFLOPS} = 1,000,000,000 \text{ floating-point operations/s}$

- 33.**
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.

- 34.**
1. Instruction dispatch (DP): Instruction packets are split into execute packets and assigned to functional units;
  2. Instruction decode (DC): Instructions are decoded.

---

## **CHAPTER 13**

### **DATA PROCESSING AND CONTROL**

---

#### ***Section 13-1 The Basic Computer System***

1. The basic elements of a computer are the CPU, memory/storage, input/output, and buses.
2. The functional units of a CPU are the arithmetic logic unit (ALU), the instruction decoder, the register set, and timing/control unit.
3. A bus is a conductor or set of conductors for transferring data that meets certain specifications.
4. The control bus signal lines vary in characteristics, nature, and function whereas the address and data bus signals are functionally identical within each bus.

#### ***Section 13-2 Practical Computer System Considerations***

5. Tristate and open-collector outputs are used to connect devices to a bus.
6. Signal loading is caused by the input current requirement or the input capacitance of a device or devices that the signal is driving. If the number of devices exceeds the fan-out specification of the driving device, signal loading occurs. Buffers are used to prevent signal loading.
7. (a) Nine loads      (b) Two loads
8. A decoder is used to select and enable a device based on the address of the device. The chip select input of a given device is enabled.
9. The wait state holds the state of the bus signals for one processor clock to allow the processor to complete an access operation.
10. The basic elements of a microprocessor or CPU are the ALU (arithmetic logic unit), the instruction decoder, the timing/control unit, and register set.

#### ***Section 13-3 The CPU: Basic Operation***

11. The microprocessor controls system hardware, provides hardware support for the operating system, and executes application programs.
12. The three microprocessor buses are the address bus, the data bus, and the control bus.
13. During fetch, an instruction is read from memory and decoded. During execute, the processor carries out the sequence of operations called for by the instruction.
14. Pipelining is a process used by a microprocessor to begin executing the next instruction in a program before the previous instruction has been completed. Several instructions can be in the pipeline simultaneously, each at different processing stages called segments. When a segment

## **Chapter 13**

completes an operation, it passes the result of the operation to the next segment and fetches the next operation from the preceding segment.

### **Section 13-4 The CPU: Addressing Modes**

**15.** Sequence of events in the inherent addressing mode:

- (1) Address of op-code placed on address bus
- (2) Op-code (instruction) placed on data bus and stored in data register
- (3) Instruction decoded
- (4) Instruction carried out

**16.** Sequence of events in the direct addressing mode:

*First fetch/execute cycle:*

- (1) Address of op-code, placed on address bus
- (2) Op-code (instruction) placed on data bus and stored in data register
- (3) Instruction decoded
- (4) Instruction carried out

*Second fetch/execute cycle:*

- (5) Address of operand placed on address bus
- (6) Operand address placed on data bus and stored in data register
- (7) Operand address loaded into address register.

*Third fetch/execute cycle:*

- (8) Address of operand placed on address bus
- (9) Operand address placed on data bus and stored in data register
- (10) Operand loaded into accumulator

**17.** Sequence of events in the indexed addressing mode:

*First fetch/execute cycle:*

- (1) Address of indexed op-code placed on address bus
- (2) Indexed op-code placed on data bus and stored in data register
- (3) Indexed instruction decoded
- (4) Address of operand fetched

*Second fetch/execute cycle:*

- (5) Offset address selected
- (6) Offset address placed on data bus and stored in data register
- (7) Offset address added to contents of index register to produce address of operand.

*Third fetch/execute cycle:*

- (8) Address of operand transferred to address register
- (9) Address of operand placed on address bus
- (10) Operand address placed on data bus and stored in data register
- (11) Operand loaded into accumulator

- 18.** The processor will branch to the address which is the sum of the program count and the relative address in the data register: Address =  $125_{10} + 55_{10} = 180_{10}$  ( $10110100_2$ ).

### **Section 13-5 The CPU: Special Operations**

- 19.** The interrupt vector table is used in auto-vectored interrupts to obtain the starting address for an interrupt service routine (ISR).
- 20.** The interrupt service routine (ISR) is similar to a standard (normal) subroutine with the following exceptions:
1. The processor automatically saves status information about the program that is executing at the time of the interrupt.
  2. The processor obtains the address of the ISR.
  3. The ISR uses an RTI instruction rather than a standard RET to return to the main program.
- 21.** The sequence of events during a bus request operation is as follows:
- (1) The bus master requesting control of the system buses submits a request by asserting the processor's bus request (BR) line.
  - (2) The processor tristates the system buses and signals that it has released control of the buses by asserting the bus grant (BG) line.
  - (3) The requesting bus master uses the system address, data, and control lines to transfer data between system devices.
  - (4) After completing the data transfers, the requesting bus master tristates the system buses and signals the end of the bus request operation by asserting the bus grant acknowledge (BGACK) line.
- 22.** DMA is direct memory access. The purpose of DMA is to transfer large amounts of data between system devices in a fraction of the time required by the system processor.

### **Section 13-6 Operating Systems and Hardware**

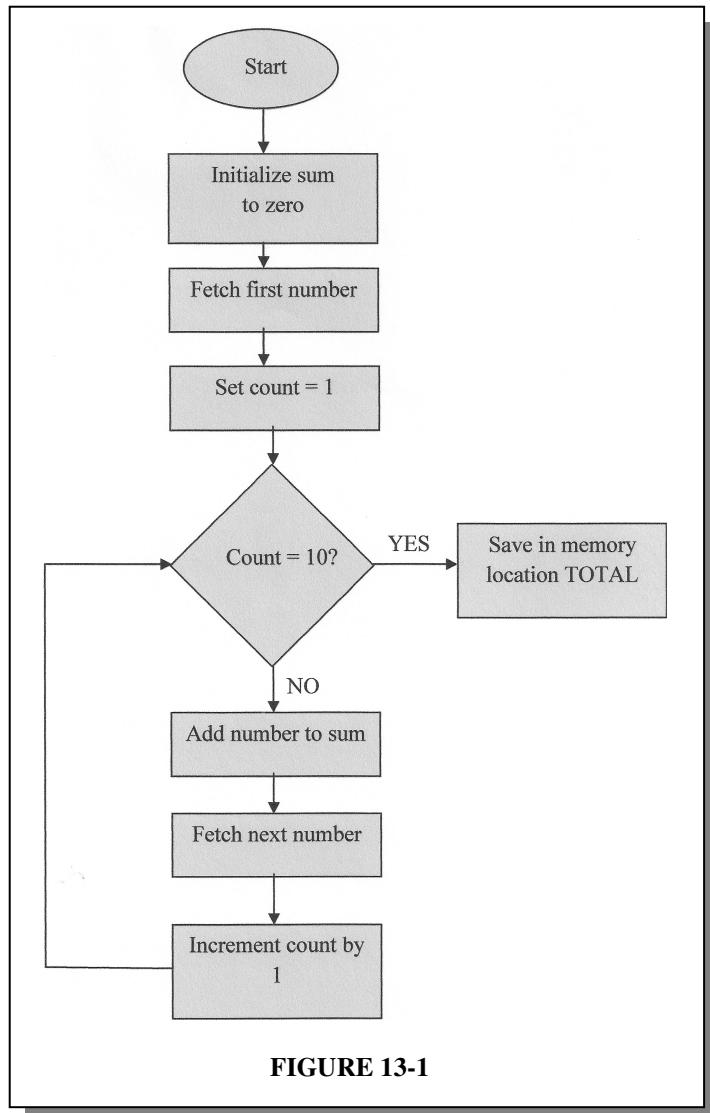
- 23.** The first group consists of application software, which includes word processors, spreadsheets, computer games, and other programs, written to accomplish some specific task. The second group consists of system software, a major portion of which is the operating system. The operating system manages the system hardware, supervises the running of applications software, provides a standard operating environment for programs in which they can run and interacts with the computer hardware.
- 24.** The three basic duties of a multitasking operating system are to allocate system resources, protect processes and system resources, and provide system services.
- 25.** Two ways in which computers execute more than one process are multitasking and multiprocessing.
- 26.** Running multiple processes can result in corruption of shared memory, corruption of one process's data by another, one process blocking another process's access to resources, and one process monopolizing system resources.

## **Chapter 13**

- 27.** MMUs handle memory accessing including memory protection, wait state generation, address translation for virtual memory, and cache control.

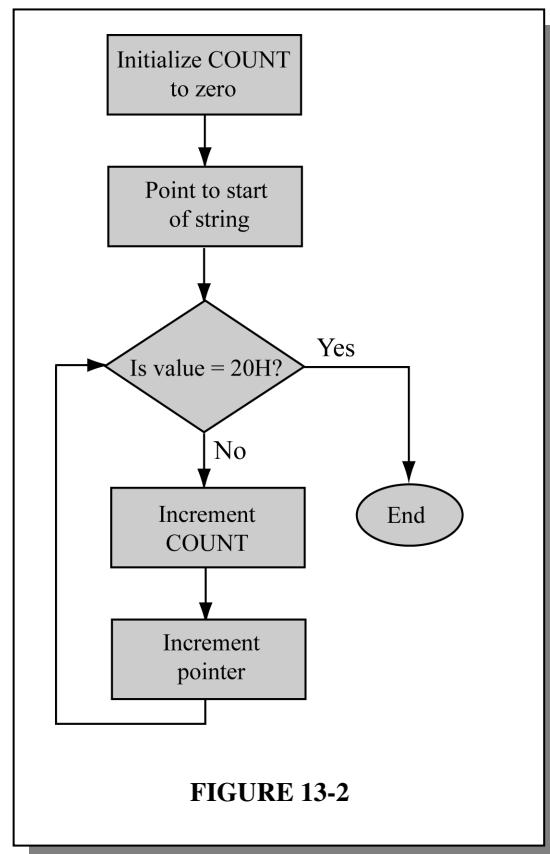
### **Section 13-7 Programming**

- 28.** An assembler is a program that converts an assembly language program into machine language that can be recognized by the microprocessor.
- 29.** One possible flow chart is shown in Figure 13-1.



**FIGURE 13-1**

- 30.** The flowchart in Figure 13-2 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 20H (space) to indicate the end.



**FIGURE 13-2**

- 31.** Move contents of bx register into ax register.
- 32.** A compiler is a program that compiles or translates a program written in a high-level language and converts it into machine code.

### **Section 13-8 Microcontrollers and Embedded Systems**

- 33.** A microcontroller is a device that combines a microprocessor with common peripheral units.

## Chapter 13

34. See Figure 13-3.

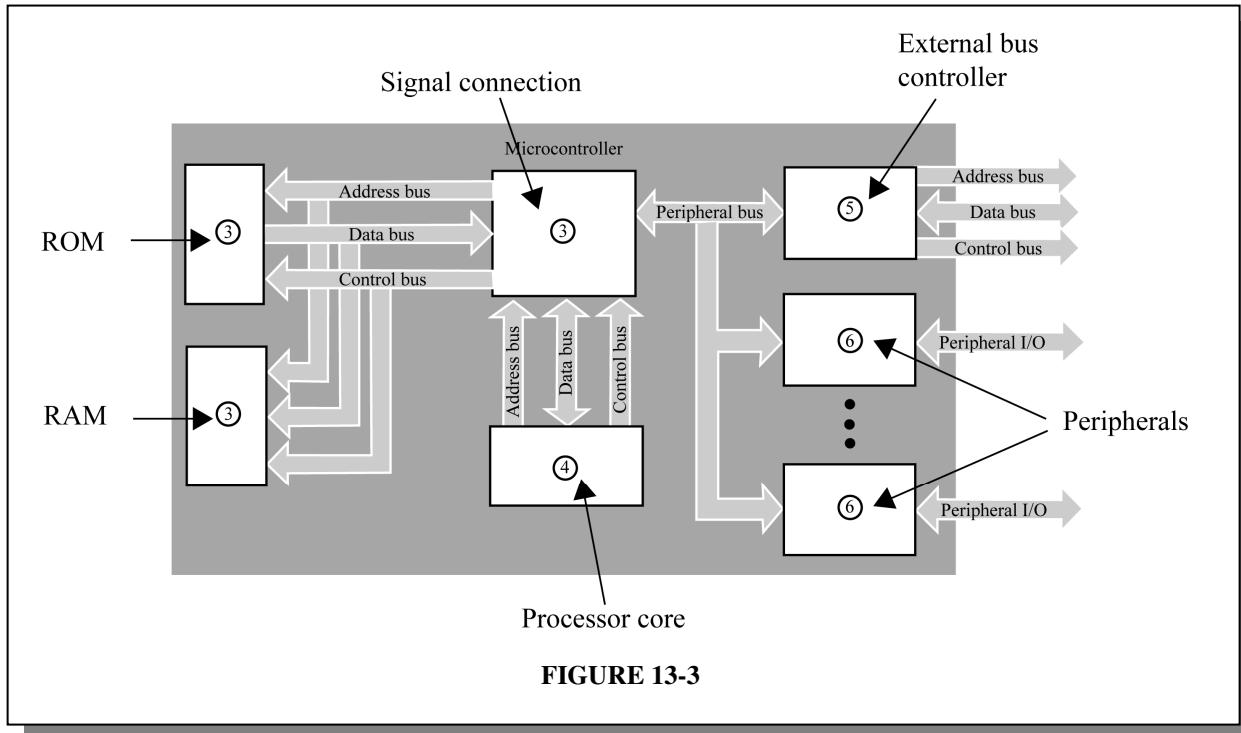


FIGURE 13-3

35. (a) GPIO: General-purpose input/output can be configured for serial data or parallel data input and output.
- (b) and (c) ADC and DAC: Analog-to-digital converter and digital-to-analog converter allow the digital system to interact with the analog world.
- (d) PWM: Pulse width modulation varies pulse width proportional to an analog signal and is commonly used for motor control.
- (e) Quadrature encoder is used to determine the speed, direction, and position of a moving object such as a mouse or stepper motor.
- (f) Timer provides for frequency and timing requirements in a microcontroller.
- (g) Communications controller allows microcontrollers to communicate with other devices using specified protocols.
36. Microcontrollers are widely used in embedded applications because they provide the interface and processing resources required by embedded systems.
37. The microcontroller includes certain peripherals that are external to a microprocessor.

---

## CHAPTER 14

### BUSES, NETWORKS, AND INTERFACING

---

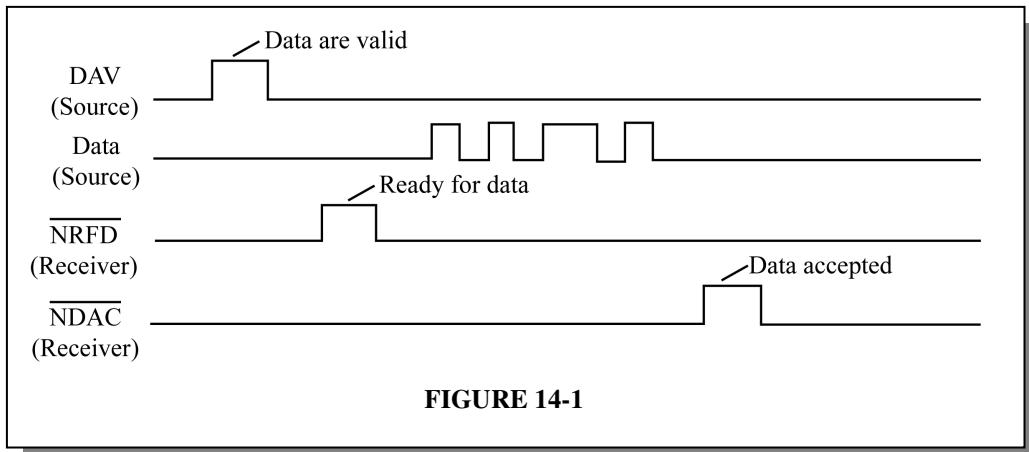
#### Section 14-1 Bus Basics

1. *Physical characteristics of a bus:* Number of conductors, length, configuration (serial or parallel), type of connector, number of connector pins, pin configuration.  
*Electrical/performance characteristics of a bus:* Signal format, signal voltage, clock frequency, transfer speed, bandwidth, data frame format, handshaking protocol, error detection, impedances.
2. A computer system bus consists of the address bus, the data bus, and the control bus.
3. The AGP (accelerated graphics port) bus is an interface for a graphic card. The graphics controller can directly access the main memory through the AGP.
4. Devices that typically connect to a computer via an external bus are the printer, the mouse, the keyboard, the monitor, and the scanner.
5. Bus width is the number of bits that a bus can transmit at one time; bus bandwidth, expressed in MBps, is the number of bytes per clock cycle times the number of clock cycles per second.
6. Bandwidth =  $\frac{(\text{width})(\text{frequency})}{8 \text{ bits/byte}} = \frac{(16 \text{ bits})(100 \text{ MHz})}{8 \text{ bits/byte}} = 200 \text{ MBps}$   
Bandwidth =  $\frac{\frac{(\text{width})(\text{frequency})}{8 \text{ bits/byte}} \cdot 10^6}{2^{20}} = 191 \text{ MBps}$
7. Simple handshake protocol: A requesting device sends a request for data to a responding device. The responding device acknowledges the request and sends the requested data.
8. A differential bus provides much higher data rates and longer transmission distances than does a single-ended bus.

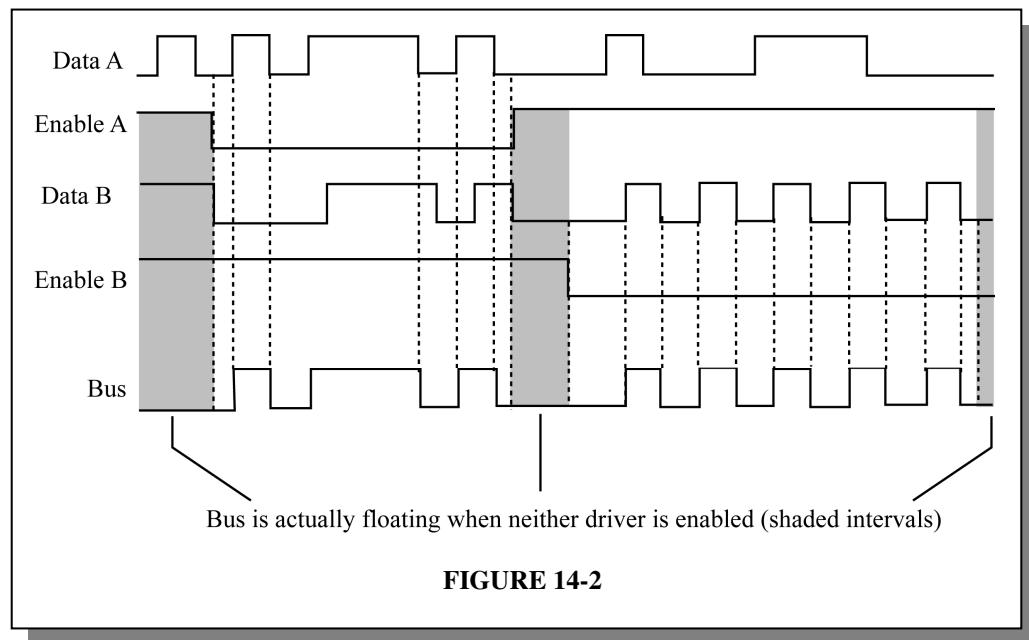
## Chapter 14

### Section 14-2 Bus Interfacing

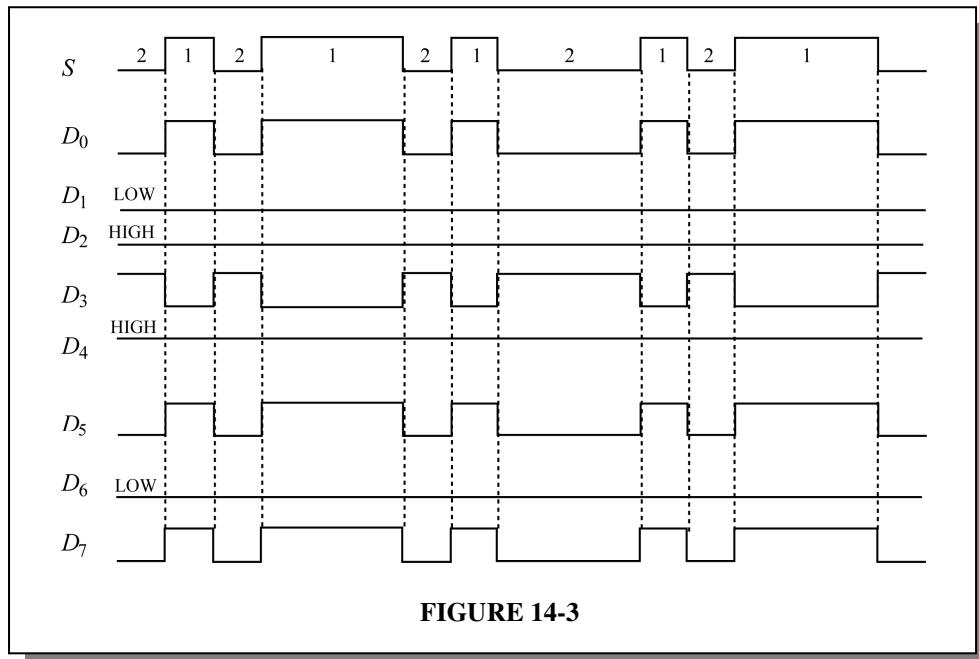
9. See Figure 14-1.



10. See Figure 14-2.



11. See Figure 14-3.



**FIGURE 14-3**

### **Section 14-3 Parallel Buses**

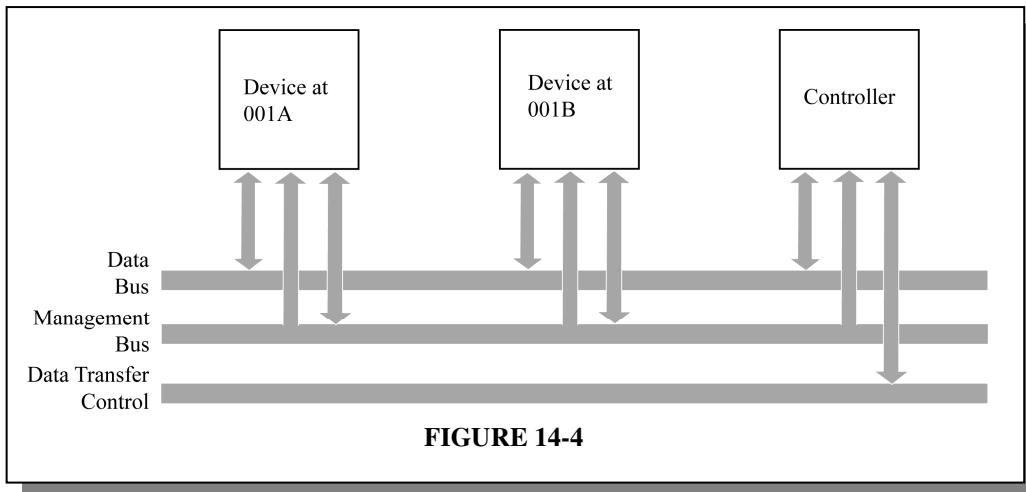
- 12. The northbridge serves as an interface for the AGP and RAM and connects to the southbridge with a PCI bus. The southbridge handles all input/output functions.
- 13. The PCI bus is an internal synchronous bus for interconnecting chips, expansion boards, and processor/memory subsystems. The original PCI bus has a width of 32 bits and a frequency of 33 MHz. Another version has a width of 64 bits and a frequency of 66 MHz. Still later versions enable 64-bit data transfers using up to a 133 MHz clock to enable bandwidths of up to 1066 MBps.

The PCI-X bus is a high-performance enhancement of the PCI and is backward compatible with PCI, although it is a faster bus and has some additional features. It is a 64-bit bus. PIC-X runs at a frequency of 133 MHz, and the PCI-X 2.0 revision supports frequencies of 266 MHz and 533 MHz. Some of the additional features increase system reliability by minimizing errors at high transfer rates.

- 14. The PCI-Express bus does not use a shared bus as PCI and PCI-X do.
- 15. x2 indicates a two-lane bus.
- 16. The terms *talker* and *listener* area associated with the IEE-488 bus ((GPIB)).
- 17. BSY: Busy  
SEL: Select  
RST: Reset  
C/D: Control/Data

|                            |
|----------------------------|
| MSG: Message               |
| REQ: Request               |
| ACK: Acknowledge a request |
| ATN: Attention             |

18. Three data bytes are transferred because the NDAC line goes HIGH three times, each time indicating that a data byte is accepted.
19. 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 14-4.



#### Section 14-4 The Universal Serial Bus (USB)

20. (a) SCSI                (b) USB                (c) Super speed USB (V 3.0)
21. Four types of USB packets: token, data, handshake, and start-of-frame.
22. *Sync Field:* All packets start with a sync (synchronization) field. The sync field consists of 8 bits for low and full speed or 32 bits for high speed and is used to synchronize the receiver clock with that of the transmitter.

*PID Field:* The packet identification field is used to identify the type of packet that is being transmitted. There are 4 bits in the PID; however, to ensure it is received correctly, the 4 bits are complemented and repeated, making an 8-bit PID code.

*Data Field:* Contains up to 1024 bytes of data.

*CRC Field:* Cyclic Redundancy Checks are performed on the data within the packet using from 5 bits to 16 bits, depending on the type of packet.

*EOP Field:* This field signals the end of a packet.

23. USB 3.0 uses **NRZ** encoding.
24. Maximum USB data bytes =  $\frac{8192 \text{ bits}}{8 \text{ bits per byte}} = 1024 \text{ bytes}$

- 25.** From Table 14-7 in textbook, the maximum total length is **82 feet (25 m)**.

### **Section 14-5 Other Serial Buses**

- 26.** RS-232 uses single-ended transmission. RS-422 uses differential transmission.
- 27.** (1) MOSI (master out slave in) is initiated by the master and received by the slave.  
 (2) MISO (master in slave out) is initiated by the slave and received by the master.  
 (3) SCLK (serial clock) is generated by the master for synchronizing data transfers.  
 (4) SS (slave select) is generated by the master to select an individual slave.
- 28.** (a) daisy chain                    (b) Parallel
- 29.** I<sup>2</sup>C is an internal serial bus primarily for connecting ICs on a PC board

**30.**

| SOF     | Arbitration field                   | Control field                    | Data field    | CRC field | ACK      | EOF      |
|---------|-------------------------------------|----------------------------------|---------------|-----------|----------|----------|
| (1 bit) | Identifier (11 bits)<br>RTR (1 bit) | Reserve (2 bits)<br>DLC (4 bits) | (0 – 8 bytes) | (16 bits) | (2 bits) | (7 bits) |

- 31.** Other possible units on an automotive CAN system include wiper control unit, parking control unit, entertainment system unit, tire pressure monitor, seat position unit, heads-up display unit.
- 32.** IEEE-1394 S100 data rate = **98.304 Mbps**; IEEE-1392 S1600 data rate = 1.6 Gbps.

### **Section 14-6 Network Topologies**

- 33.** (a) bus                    (b) ring                    (c) star                    (d) tree                    (e) mesh
- 34.** In shared media topology, all devices can access the physical network at any time as long as no other device is attempting access at the same time. In token-based topology, a code called the *token* travels around the network and if a device needs to send data, it grabs the token and attaches the data packet to it.
- 35.** The physical structure or the way in which devices are connected to the network with actual cables is called physical topology. The way in which data are moved through the network is called logical topology and is related to network protocol.
- 36.** The OSI layers are as follows.
- (1) *Physical:* This layer defines the actual network hardware and physical characteristics such as type of cables or connections, voltage levels, and timing.
  - (2) *Data:* The protocol is assigned to the data and the data packet sequencing. Also, the type of network is specified.

## **Chapter 14**

- (3) *Network*: Logical protocol, addressing, and routing are determined at this layer.
  - (4) *Transport*: This layer provides data flow control, error checking, and data recovery.
  - (5) *Session*: Communication with a receiving device is established, maintained, and ended by this layer.
  - (6) *Presentation*: This layer converts data to a standard format that can be understood by all the other layers.
  - (7) *Application*: This layer interacts with the application or operating system when files are transferred, messages are read, or any other network-related activity is required
- 37.** Ethernet local area network (LAN) technology or protocol uses a process called CSMA/CD (carrier sense multiple access/collision detection) to access the network.
- 38.** The Ethernet data frame has a destination address and a source address.
- 39.** CRC (cyclic redundancy check) is performed in the FCS field of the Ethernet data frame.
- 40.** Data collisions occur when two devices send data at the same time. A collision is handled in the Ethernet system with the CSMA/CD protocol.
- 41.** The Ethernet physical layer describes interconnection methods, cables and connectors, data encoding (Manchester), and electrical properties.
- 42.** In the data link layer, Ethernet
  - Creates a data frame for IP data (Internet protocol)
  - Address devices on the network
  - Passes data within the LAN and to network interface controller (NIC) for IP data
  - Provides CSMA/CD protocol
  - Detects errors
- 43.** TCP/IP stands for Transmission Control Protocol/Internet Protocol and is the communication protocol for the Internet. There are two protocol suites in TCP/IP. TCP provides for communication between applications; IP handles communication between computers and is responsible for sending data packets to the right destination across a network.
- 44.** The binary code for the IP address 162.18.248.1 is  
**10000010.00010010.11111000.00000001**
- 45.** Client/server networking is an application architecture that handles tasks between service providers called *servers* and service requesters called *clients*.

# **PART TWO**

## **Laboratory Solutions for**

### **Experiments in Digital Fundamentals:**

### **A Systems Approach**

**David Buchla and Doug Joksch**

## Experiment 1: Laboratory Instrument Familiarization

### Data and Observations

Data and the number of significant figures will vary according to signal generator and measurement equipment. Sample (measured) data is shown.

**TABLE 1-1**

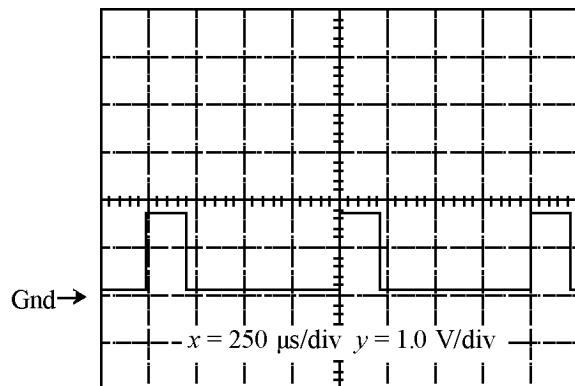
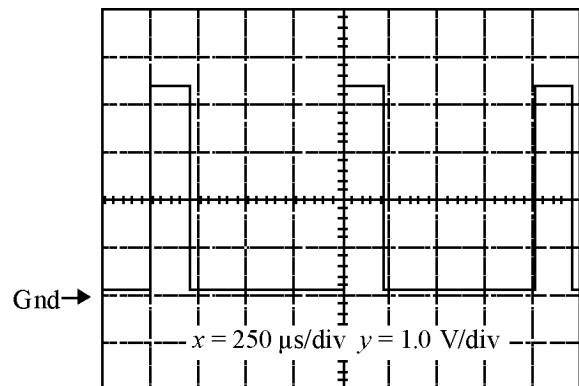
| Voltage Setting = 5.0 V | Voltage Reading |
|-------------------------|-----------------|
| Power Supply meter      | <b>5.0 V</b>    |
| DMM                     | <b>5.02 V</b>   |
| Oscilloscope            | <b>5.04 V</b>   |

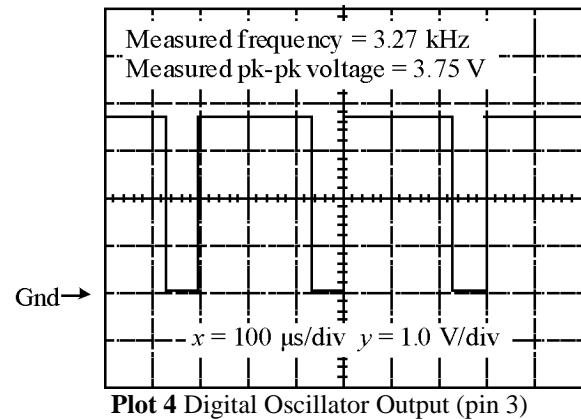
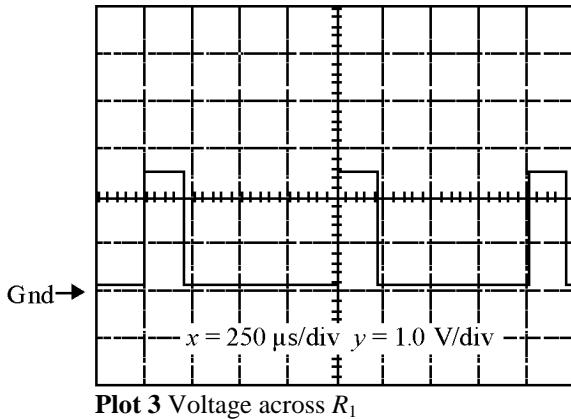
**TABLE 1-2**

| Function Generator Parameters (at 1.0 kHz) | Measured Values              |
|--------------------------------------------|------------------------------|
| Pulse Width                                | <b>197 <math>\mu</math>s</b> |
| Period                                     | <b>1.00 ms</b>               |
| Amplitude                                  | <b>4.3 V</b>                 |

**TABLE 1-3**

| Step | Digital Oscillator Parameters | Measured Values              |
|------|-------------------------------|------------------------------|
| 12   | Period                        | <b>306 <math>\mu</math>s</b> |
|      | Duty cycle                    | <b>78%</b>                   |
|      | Amplitude                     | <b>3.75 V</b>                |
|      | Frequency                     | <b>3.27 kHz</b>              |
| 13   | Period                        | <b>352 ms</b>                |
|      | Frequency                     | <b>2.84 Hz</b>               |





### Further Investigation Results:

With the partial failure consisting of a  $100\ \Omega$  resistor in parallel with the LED, most of the current goes through the resistor (Figure 1-8b). This current can be traced through  $R_1$  and  $R_2$  and back to the supply using a logic pulser and current tracer as shown in Figure 1-9. With the short circuit (Figure 1-10), the current can be traced along the protoboard's bus, through the short and back to the supply with no current in the resistors or the LED.

### Evaluation and Review Questions

1. The circuit can be damaged if the wrong voltage is connected to it.
2. Vertical section: sets the amplitude of the input signal and develops voltages for display section; it sends signals to the trigger section.  
Trigger section: causes the start of the acquisition of the waveform. A stable display requires the trigger to occur at the same point on the waveform for each acquisition.  
Horizontal section: controls the time base.  
Display section: changes the intensity or other display parameters.
3. A probe from each channel is connected across the ungrounded component. All grounds are connected to the circuit ground. The scope is configured to measure the difference between the two channels. (This will vary between scope types; it sometimes requires the channels to be added with one channel inverted).
4. (a) The oscillator works but the LED is off. The voltage at pin 3 and the LED are identical.  
(b) The oscillator works but the frequency is too low (depending on the capacitor).  
(c) Reversing the power and ground leads will result in destructively high current.  
(d) The oscillator works but the LED is off. No voltage appears at the LED (but pin 3 is okay).
5. A digital oscilloscope is typically as accurate as a DMM and allows detection of noise or ripple with a power supply. A DMM is typically more portable. An analog scope is not as accurate as a DMM but does not show potential problems such as noise or ripple. A digital scope may be as accurate as a DMM because of automated measurement capability and gives more information than a DMM about potential problems such as the presence of noise.
6. Pulses from the logic pulser are applied to the circuit board trace that normally carries power but with power removed. The current tracer can be moved along this line, following the path of current until the short is found. (Note that this technique can be used for any board, analog or digital).

## Experiment 2: Constructing a Logic Probe

### Data and Observations

Step 3: Logic thresholds HIGH 1.98 V LOW 0.76 V

**TABLE 2-1**

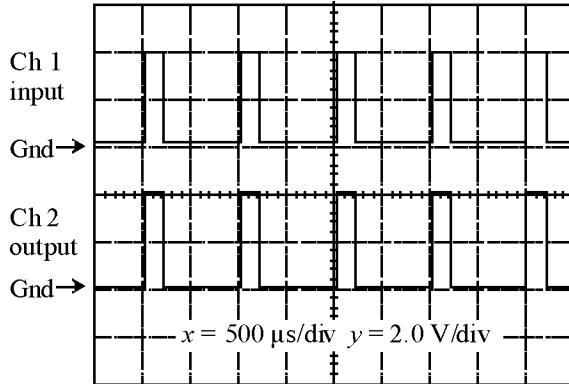
| Step |                      | Output Logic Level |               |               |
|------|----------------------|--------------------|---------------|---------------|
|      |                      | Input is LOW       | Input is OPEN | Input is HIGH |
| 4    | one inverter         | <b>HIGH</b>        | <b>LOW</b>    | <b>LOW</b>    |
| 5    | two series inverters | <b>LOW</b>         | <b>HIGH</b>   | <b>HIGH</b>   |

**TABLE 2-2**

| Step |                                 | Input Logic Level (pin 3) | Output Logic Level (pin 3) | Logic Level (pin 5) | Logic Level (pin 6) |
|------|---------------------------------|---------------------------|----------------------------|---------------------|---------------------|
| 7    | $V_{in}$ momentarily on ground. |                           | <b>HIGH</b>                |                     |                     |
| 8    | $V_{in}$ momentarily on +5.0 V. |                           | <b>LOW</b>                 |                     |                     |
| 9    | Fault condition: open at pin 5  | <b>LOW</b>                | <b>HIGH</b>                | <b>INVALID</b>      | <b>LOW</b>          |
| 10   | Voltages with fault: DMM        | <b>0.06 V</b>             | <b>4.14 V</b>              | <b>1.64 V</b>       | <b>0.06 V</b>       |
| 11   | Voltages with fault: (scope)    | <b>0.06 V</b>             | <b>4.18 V</b>              | <b>1.65 V</b>       | <b>0.06 V</b>       |

### Further Investigation Results:

The signals are nearly identical as shown. In the experimental circuit, a slight difference in amplitude was noted and the output is delayed by 14 ns on the leading edge and 16 ns on the trailing edge. The time difference is due to transition time through the gates.



**Plot 1**

### Evaluation and Review Questions:

1. Increase the value of the  $330\ \Omega$  resistor in the input voltage divider.
2. Two inverters in series form a buffer that can allow additional drive current from the same logic output.
3. (a) The circuits are identical.  
 (b) If the input to the first inverter is open, an invalid level of approximately 1.6 V will be observed.  
 (c) If the input to the first inverter is open, then the output will be LOW.
4. A logic probe can quickly determine if a valid logic HIGH or LOW is present at some point in a circuit whereas a DMM can assign a numeric quantity to the level.
5. (a)  $V_{out}$  will alternate between a HIGH and LOW at a rapid rate.  
 (b) Approximately 50 ns.  
 (c) Since the period will be approximately 100 ns, the oscillation frequency is 10 MHz.
6. A steady state invalid level is an indication of an open input.

## Experiment 3: Number Systems

### Data and Observations

TABLE 3-1

| Inputs        |             | Output                |
|---------------|-------------|-----------------------|
| Binary Number | BCD Number  | Seven-Segment Display |
| 0 0 0 0       | <b>0000</b> | 0                     |
| 0 0 0 1       | <b>0001</b> | 1                     |
| 0 0 1 0       | <b>0010</b> | 2                     |
| 0 0 1 1       | <b>0011</b> | 3                     |
| 0 1 0 0       | <b>0100</b> | 4                     |
| 0 1 0 1       | <b>0101</b> | 5                     |
| 0 1 1 0       | <b>0110</b> | 6                     |
| 0 1 1 1       | <b>0111</b> | 7                     |
| 1 0 0 0       | <b>1000</b> | 8                     |
| 1 0 0 1       | <b>1001</b> | 9                     |
| 1 0 1 0       | INVALID     | 0                     |
| 1 0 1 1       | INVALID     | 1                     |
| 1 1 0 0       | INVALID     | 2                     |
| 1 1 0 1       | INVALID     | 3                     |
| 1 1 1 0       | INVALID     | 4                     |
| 1 1 1 1       | INVALID     | 5                     |

Step 6. Method to cause leading zero suppression: The 7447A's ripple-blanking input for the most significant digit is connected to a LOW. If inputs A, B, C, D are also LOW, the display will be off and the ripple blanking output is LOW. Successive digits are blanked by connecting the ripple blanking output from the most significant digit to the next decoder's ripple-blanking input.

TABLE 3-2

| Trouble Number | Trouble                                           | Observations                                                                                              |
|----------------|---------------------------------------------------|-----------------------------------------------------------------------------------------------------------|
| 1              | LED for the C input is open                       | <i>The "C" switch has no effect on the output. The "C" input on the 7447A is always LOW.</i>              |
| 2              | A input to 7447A is open                          | <i>Only odd numbers can be observed on the display because the open A input is interpreted as a HIGH.</i> |
| 3              | LAMP TEST is shorted to ground                    | <i>All segments are on; switches have no effect.</i>                                                      |
| 4              | Resistor connected to pin 15 of the 7447A is open | <i>Upper left (f) segment is always off. Numbers 0, 4, 5, 6, 8, 9 are shown incorrectly.</i>              |

**Further Investigation Results:**

The circuit needed is the same as Figure 3-2 except the MSB is disconnected and grounded at the BCD input to the 7447A. The LED used to indicate the D input can be connected to an LED that represents the octal MSB or it could be shown on the A input of a second 7447A decoder. The idea of this investigation is to reinforce converting a binary number to an octal number by grouping the binary bits by groups of three.

**Evaluation and Review Questions:**

1. Segment g open on seven segment display, open wire or resistor to g segment, bad 7447A.
2. Leave switches with binary 1000 and test logic at segment g. If the g segment is LOW, the segment is bad; if it is HIGH, move to pin 14 of 7447A and test the logic. If it is LOW, the resistive path is open; if it is HIGH, test inputs to 7447A.
3. All segments will be off.
4. Binary is a base two weighted number system. Column values increase by powers of two. BCD is a code that represents each decimal digit with 4 binary bits.
- 5.

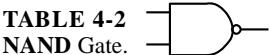
| Binary          | Octal      | Hexadecimal | Decimal    | BCD                   |
|-----------------|------------|-------------|------------|-----------------------|
| <b>01001100</b> | <u>114</u> | <u>4C</u>   | <u>76</u>  | <u>0111 0110</u>      |
| <u>11000100</u> | <b>304</b> | <u>C4</u>   | <u>196</u> | <u>0001 1001 0110</u> |
| <u>11100110</u> | <b>346</b> | <b>E6</b>   | <u>230</u> | <u>0010 0011 0000</u> |
| <u>111001</u>   | <b>71</b>  | <u>39</u>   | <b>57</b>  | <u>0101 0111</u>      |
| <u>110001</u>   | <b>61</b>  | <u>31</u>   | <u>49</u>  | <b>0100 1001</b>      |

6. a. The base can be found algebraically by solving  $1x^2 + 2x^1 + 5 = 85$  ( $x$  represents the base). The positive root (base) is 8.  
b. 16

## Experiment 4: Logic Gates

### Data and Observations

TABLE 4-2  
NAND Gate.



| Inputs |   | Output | Measured Output Voltage |
|--------|---|--------|-------------------------|
| A      | B | X      |                         |
| 0      | 0 | 1      | 3.95 V                  |
| 0      | 1 | 1      | 3.95 V                  |
| 1      | 0 | 1      | 3.95 V                  |
| 1      | 1 | 0      | 0.06 V                  |

TABLE 4-3  
NOR Gate.



| Inputs |   | Output | Measured Output Voltage |
|--------|---|--------|-------------------------|
| A      | B | X      |                         |
| 0      | 0 | 1      | 4.01 V                  |
| 0      | 1 | 0      | 0.06 V                  |
| 1      | 0 | 0      | 0.06 V                  |
| 1      | 1 | 0      | 0.06 V                  |

TABLE 4-4  
Truth table for Figure 4-4.



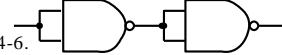
| Input | Output | Measured Output Voltage |
|-------|--------|-------------------------|
| A     | X      |                         |
| 0     | 1      | 3.95 V                  |
| 1     | 0      | 0.06 V                  |

TABLE 4-5  
Truth table for Figure 4-5.



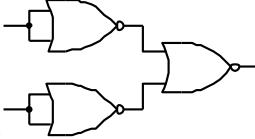
| Input | Output | Measured Output Voltage |
|-------|--------|-------------------------|
| A     | X      |                         |
| 0     | 1      | 4.01 V                  |
| 1     | 0      | 0.06 V                  |

TABLE 4-6  
Truth table for Figure 4-6.



| Input | Output | Measured Output Voltage |
|-------|--------|-------------------------|
| A     | X      |                         |
| 0     | 0      | 0.05 V                  |
| 1     | 1      | 3.97 V                  |

TABLE 4-7  
Truth table for Figure 4-7.



| Inputs |   | Output | Measured Output Voltage |
|--------|---|--------|-------------------------|
| A      | B | X      |                         |
| 0      | 0 | 0      | 0.06 V                  |
| 0      | 1 | 0      | 0.06 V                  |
| 1      | 0 | 0      | 0.06 V                  |
| 1      | 1 | 0      | 3.96 V                  |

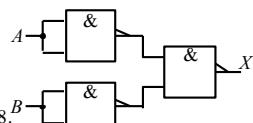
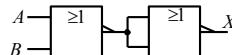


TABLE 4-8  
Truth table for Figure 4-8.

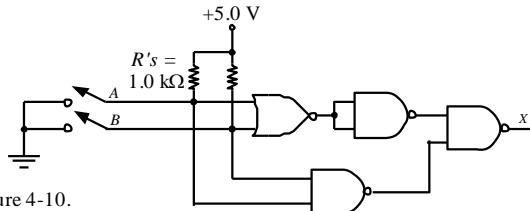
| Inputs |   | Output | Measured Output Voltage |
|--------|---|--------|-------------------------|
| A      | B | X      |                         |
| 0      | 0 | 0      | 0.05 V                  |
| 0      | 1 | 1      | 3.98 V                  |
| 1      | 0 | 1      | 3.98 V                  |
| 1      | 1 | 1      | 3.98 V                  |

TABLE 4-9  
Truth table for Figure 4-9.



| Inputs |   | Output | Measured Output Voltage |
|--------|---|--------|-------------------------|
| A      | B | X      |                         |
| 0      | 0 | 0      | 0.06 V                  |
| 0      | 1 | 1      | 3.96 V                  |
| 1      | 0 | 1      | 3.96 V                  |
| 1      | 1 | 1      | 3.95 V                  |

### Further Investigation Results:



**TABLE 4-10**  
Truth table for Figure 4-10.

| Inputs |   | Output | Measured Output Voltage |
|--------|---|--------|-------------------------|
| A      | B | X      |                         |
| 0      | 0 | 1      | <b>3.97 V</b>           |
| 0      | 1 | 0      | <b>0.06 V</b>           |
| 1      | 0 | 0      | <b>0.06 V</b>           |
| 1      | 1 | 1      | <b>3.97 V</b>           |

The equivalent gate is an XOR gate.

### Evaluation and Review Questions:

1. a. Figures 4-4 and 4-5.  
b. Figure 4-7.  
c. Figures 4-8 and 4-9
2. Betty. A LOW on any input produces a LOW on the output. This describes an AND gate.
3. A HIGH on any input causes a HIGH output. This describes an OR gate.
4.
 

The circuit diagram shows a logic circuit with two inputs, A and B. Input A is connected to the non-inverting input of a first inverter. Input B is connected to the inverting input of the same inverter. The output of this inverter is connected to the inverting input of a second inverter. The non-inverting input of the second inverter is connected to ground through a 1.0 kΩ resistor labeled R's. The output of the second inverter is connected to the inverting input of a third inverter. The non-inverting input of the third inverter is connected to ground through a 1.0 kΩ resistor labeled R's. The output of the third inverter is labeled X. A 5.0 V power source is connected between the output X and ground.
5. When the signal is HIGH, data is transmitted; when it is LOW, data is received.
6. The fact that an output is at a constant level does not in itself indicate a bad gate. If the circuit is not working properly, check the inputs first. If all inputs are HIGH, and the output is HIGH, the gate may be bad, but connections to the output should be checked to see if another component is holding the output HIGH.

## Experiment 5: More Logic Gates: Data and Observations:

**TABLE 5-2**  
OR Gate

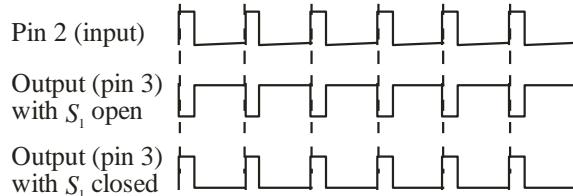


| Inputs |   | Output | Measured Output Voltage |
|--------|---|--------|-------------------------|
| A      | B | X      |                         |
| 0      | 0 | 0      | 0.07 V                  |
| 0      | 1 | 1      | 3.93 V                  |
| 1      | 0 | 1      | 3.93 V                  |
| 1      | 1 | 1      | 3.93 V                  |

**TABLE 5-3**  
XOR Gate



| Inputs |   | Output | Measured Output Voltage |
|--------|---|--------|-------------------------|
| A      | B | X      |                         |
| 0      | 0 | 0      | 0.06 V                  |
| 0      | 1 | 1      | 3.94 V                  |
| 1      | 0 | 1      | 3.94 V                  |
| 1      | 1 | 0      | 0.06 V                  |



**PLOT 1**

**TABLE 5-4**

| $D_3$ | $D_2$ | $D_1$ | $D_0$ | $Q_3$ | $Q_2$ | $Q_1$ | $Q_0$ |
|-------|-------|-------|-------|-------|-------|-------|-------|
| 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |
| 0     | 0     | 0     | 1     | 1     | 1     | 1     | 1     |
| 0     | 0     | 1     | 0     | 1     | 1     | 1     | 0     |
| 0     | 0     | 1     | 1     | 1     | 1     | 0     | 1     |
| 0     | 1     | 0     | 0     | 1     | 1     | 0     | 0     |
| 0     | 1     | 0     | 1     | 1     | 0     | 1     | 1     |
| 0     | 1     | 1     | 0     | 1     | 0     | 1     | 0     |
| 0     | 1     | 1     | 1     | 1     | 0     | 0     | 1     |
| 1     | 0     | 0     | 0     | 1     | 0     | 0     | 0     |
| 1     | 0     | 0     | 1     | 0     | 1     | 1     | 1     |
| 1     | 0     | 1     | 0     | 0     | 1     | 1     | 0     |
| 1     | 0     | 1     | 1     | 0     | 1     | 0     | 1     |
| 1     | 1     | 0     | 0     | 0     | 1     | 0     | 0     |
| 1     | 1     | 0     | 1     | 0     | 0     | 1     | 1     |
| 1     | 1     | 1     | 0     | 0     | 0     | 1     | 0     |
| 1     | 1     | 1     | 1     | 0     | 0     | 0     | 1     |

**TABLE 5-5**

| Symptom Number | Symptom                                                               | Possible Cause                                                                      |
|----------------|-----------------------------------------------------------------------|-------------------------------------------------------------------------------------|
| 1              | None of the LEDs operate; the switches have no effect.                | <i>Power supply not on or open lead from the power or ground line to the board.</i> |
| 2              | LEDs on the output side do not work; those on the input side do work. | <i>Power or ground connection to the 7486 open.</i>                                 |
| 3              | The LED representing $Q_3$ is sometimes on when it should be off.     | <i>An input to the top XOR gate is likely to be open.</i>                           |
| 4              | The Complement switch has no effect on the outputs.                   | <i>Open or shorted connection from switch or a bad switch.</i>                      |

### Further Investigation Results:

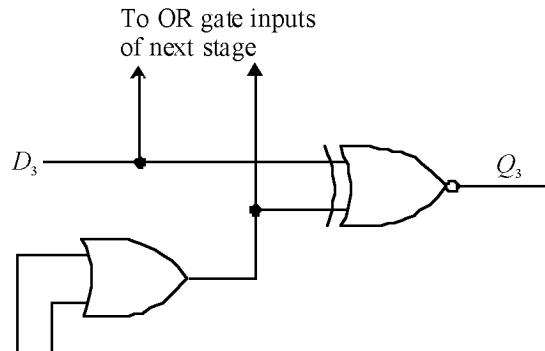
Each time a switch is thrown, no matter which one, the LED will change states (either on or off).

### Evaluation and Review Questions:

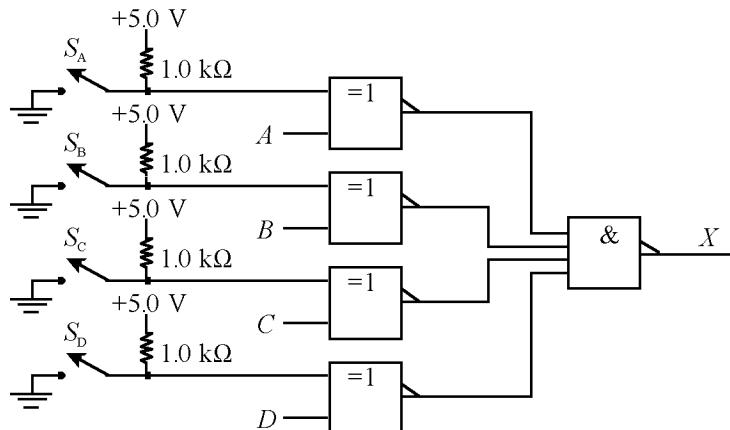
- One input is a control input and the other is the data. The data is unchanged if the control input is LOW but is inverted if the control input is HIGH. This is easily seen on a truth table to the right.

| Control | Data | Output |
|---------|------|--------|
| 0       | 0    | 0      |
| 0       | 1    | 1      |
| 1       | 0    | 1      |
| 1       | 1    | 0      |

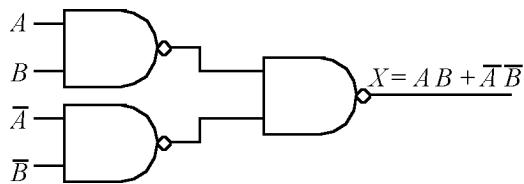
- The circuit can be expanded by “chaining” additional OR gates to D3 and the output of the upper OR gate as shown in the partial schematic to the right. The output of the OR gate is connected to the input of the next XOR gate. The other XOR input is connected to the data line as before.



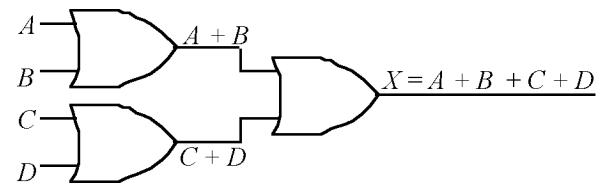
- The output of an XNOR is HIGH when the inputs agree. If the A B C D inputs all agree with their respective switch settings, then all XOR gates have a HIGH output and the NAND gate output (labeled X) will be LOW; otherwise it will be HIGH.
- The comparator schematic, drawn with ANSI/IEEE Std 91-1984 symbols, is shown below.



- XOR function with NAND gates:



- Four input OR function with 2-input gates:



## Experiment 6: Interpreting Manufacturer's Data Sheets: Data and Observations:

**TABLE 6-1**

TTL 7404

| Recommended Operating Conditions                                                                           |                                   |                                                                  |     |      |        |     |      |       |                     |
|------------------------------------------------------------------------------------------------------------|-----------------------------------|------------------------------------------------------------------|-----|------|--------|-----|------|-------|---------------------|
| Symbol                                                                                                     | Parameter                         | DM5405                                                           |     |      | DM7404 |     |      | Units | Measured Value      |
|                                                                                                            |                                   | Min                                                              | Nom | Max  | Min    | Nom | Max  |       |                     |
| $V_{CC}$                                                                                                   | Supply Voltage                    | 4.5                                                              | 5   | 5.5  | 4.75   | 5   | 5.25 | V     |                     |
| $V_{IH}$                                                                                                   | High-Level Input Voltage          | 2                                                                |     |      | 2      |     |      | V     | a. <b>4.98 V</b>    |
| $V_{IL}$                                                                                                   | Low-Level Input Voltage           |                                                                  |     | 0.8  |        |     | 0.8  | V     | b. <b>0.289 V</b>   |
| $I_{OH}$                                                                                                   | High-Level Output Current         |                                                                  |     | -0.4 |        |     | -0.4 | mA    | c. <b>-0.245 mA</b> |
| $I_{OL}$                                                                                                   | Low-Level Output Current          |                                                                  |     | 16   |        |     | 16   | mA    | d. <b>14.5 mA</b>   |
| $T_A$                                                                                                      | Free Air Operating Temperature    | -55                                                              |     | 125  | 0      |     | 70   | °C    |                     |
| <b>Electrical Characteristics</b> Over Recommended Operating Free Air Temperature (unless otherwise noted) |                                   |                                                                  |     |      |        |     |      |       |                     |
| Symbol                                                                                                     | Parameter                         | Conditions                                                       |     |      | Min    | Typ | Max  | Units |                     |
| $V_I$                                                                                                      | Input clamp voltage               | $V_{CC}=\text{Min}$ , $I_I=-12\text{mA}$                         |     |      |        |     | -1.5 | V     |                     |
| $V_{OH}$                                                                                                   | High-level output voltage         | $V_{CC}=\text{Min}$ , $I_{OH}=\text{Max}$<br>$V_{IL}=\text{Max}$ |     |      | 2.4    | 3.4 |      | V     | e. <b>3.68 V</b>    |
| $V_{OL}$                                                                                                   | Low Level Output Voltage          | $V_{CC}=\text{Min}$ , $I_{OL}=\text{Max}$<br>$V_{IH}=\text{Min}$ |     |      |        | 0.2 | 0.4  | V     | f. <b>0.222 V</b>   |
| $I_I$                                                                                                      | Input Current @ max Input Voltage | $V_{CC}=\text{Max}$ , $V_I=5.5\text{ V}$                         |     |      |        |     | 1    | mA    |                     |

| Electrical Characteristics Over Recommended Operating Free Air Temperature (unless otherwise noted) |                                  |                                          |      |     |     |      |      |       |                    |
|-----------------------------------------------------------------------------------------------------|----------------------------------|------------------------------------------|------|-----|-----|------|------|-------|--------------------|
| Symbol                                                                                              | Parameter                        | Conditions                               |      |     | Min | Type | Max  | Units | Measured Value     |
| $I_{IH}$                                                                                            | High-Level Input Current         | $V_{CC}=\text{Max}$ , $V_I=2.4\text{ V}$ |      |     |     |      | 40   | μA    | g. <b>7.4 μA</b>   |
| $I_{IL}$                                                                                            | Low-Level Input Current          | $V_{CC}=\text{Max}$ , $V_I=0.4\text{ V}$ |      |     |     |      | -1.6 | μA    | h. <b>-0.88 mA</b> |
| $I_{OS}$                                                                                            | Short Circuit Output Current     | $V_{CC}=\text{Max}$                      | DM54 | -20 |     | -55  | mA   |       |                    |
|                                                                                                     |                                  |                                          | DM74 | -18 |     | -55  |      |       |                    |
| $I_{CCH}$                                                                                           | Supply Current With Outputs High | $V_{CC}=\text{Max}$                      |      |     |     | 8    | 16   | mA    |                    |
| $I_{CCL}$                                                                                           | Supply Current With Outputs Low  | $V_{CC}=\text{Max}$                      |      |     |     | 14   | 27   | mA    |                    |

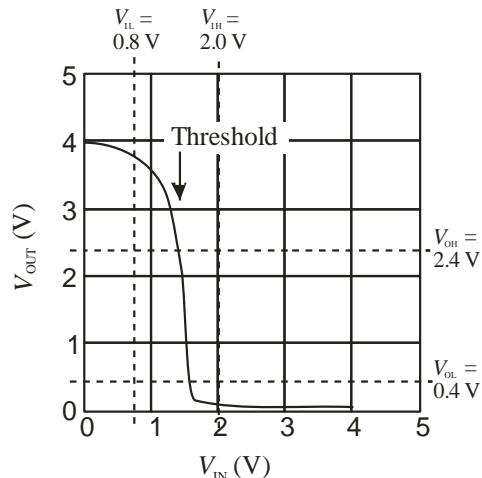
**TABLE 6-2 CMOS 4081**

|     | Quantity                                  | Manufacturer's Specified Value | Measured Value                   |
|-----|-------------------------------------------|--------------------------------|----------------------------------|
| (a) | $V_{OL(max)}$ , low-level output voltage  | <b>0.05 V</b>                  | <b>0.1 mV</b>                    |
| (b) | $V_{OH(min)}$ , high-level output voltage | <b>4.95 V</b>                  | <b>4.99 V</b>                    |
| (c) | $V_{IL(max)}$ , low-level input voltage   | <b>1.5 V</b>                   | <b>1.5 V</b>                     |
| (d) | $V_{IH(min)}$ , high-level output voltage | <b>3.5 V</b>                   | <b>3.5 V</b>                     |
| (e) | $I_{OL(min)}$ , high-level output current | <b>0.51 mA</b>                 | <b>1.10 mA</b>                   |
| (f) | $I_{OH(min)}$ , high-level output current | <b>-0.51 mA</b>                | <b>-1.05 mA</b>                  |
| (g) | $I_{IN(typ)}$ , input current             | $\pm 10^{-5} \mu A$            | <b>0.04 <math>\mu A^*</math></b> |

\* limited by instruments

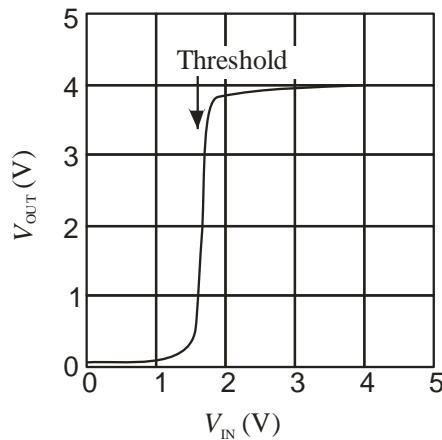
**Further Investigation Results:**

| $V_{in}$ (V) | $V_{out}$ (V) |
|--------------|---------------|
| 0.4          | <b>3.98</b>   |
| 0.8          | <b>3.64</b>   |
| 1.2          | <b>2.88</b>   |
| 1.3          | <b>2.52</b>   |
| 1.4          | <b>1.39</b>   |
| 1.5          | <b>0.03</b>   |
| 1.6          | <b>0.03</b>   |
| 2.0          | <b>0.03</b>   |
| 2.4          | <b>0.03</b>   |
| 2.8          | <b>0.03</b>   |
| 3.2          | <b>0.03</b>   |
| 3.6          | <b>0.03</b>   |
| 4.0          | <b>0.03</b>   |



**Evaluation and Review Questions:**

1. A load causes the output LOW to be higher. (The measured LOW with no load was 30 mV; with the minimum load, the measured LOW was 222 mV).
2.  $R = 2.4 \text{ V} / 0.4 \text{ mA} = 6 \text{ k}\Omega$ .
3.  $V_{\text{NL(LOW)}} = 0.3 \text{ V}$   
 $V_{\text{NL(HIGH)}} = 0.5 \text{ V}$
4. a. Circuit (a) is better.  
b. The  $I_{\text{OH}}$  specification is exceeded in circuit (b). In (a), the LED is turned on with  $I_{\text{OL}}$  and is within the specified limit.
5. Logic levels are specified with reference to ground and can produce incorrect logic if the ground level is wrong. The scope should be dc coupled to assure the troubleshooter knows the ground level of the signal. (Note that in some digital scopes, the ground level is shown with a small arrow to the side of the display.)
6. The transfer curve for an AND gate (with inputs tied together) is shown below. The threshold is expected to occur at +1.6 V.



*Plot for Question 6*

## Experiment 7: Boolean Laws and DeMorgan's Theorems

Data and Observations:

TABLE 7-2

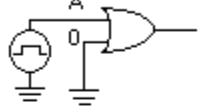
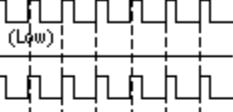
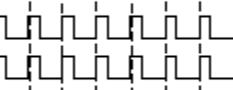
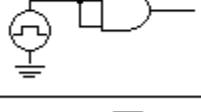
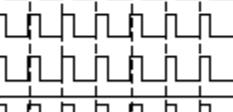
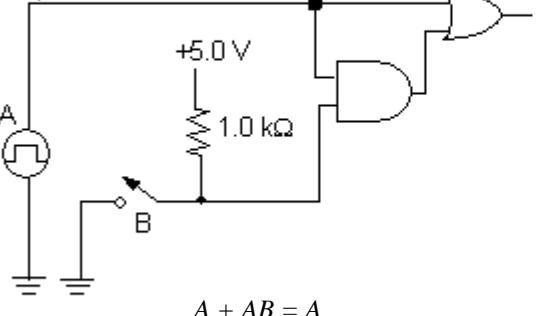
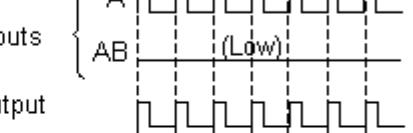
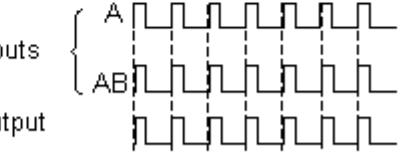
| Schematic                                                                         | Timing Diagram                                                                                                | Boolean Rule          |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------|-----------------------|
|  | Inputs: A, 0<br>Output:      | $A + 0 = A$           |
|  | Inputs: A, A<br>Output:      | $A + A = A$           |
|  | Inputs: A, A-bar<br>Output:  | $A \cdot A = A$       |
|  | Inputs: A, A-bar<br>Output:  | $A \cdot \bar{A} = 0$ |

TABLE 7-3

| Schematic                                                                                                        | Timing Diagram                                                                                                                                                                                                                                                                                                                          |
|------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>Rule 10:</p>  $A + AB = A$ | <p>Timing diagram for <math>B = 0</math>:</p> <p>Inputs: A, AB<br/>Output: </p> <p>Timing diagram for <math>B = 1</math>:</p> <p>Inputs: A, AB<br/>Output: </p> |

**TABLE 7-4**

| Schematic                              | Timing Diagram                                                                              |
|----------------------------------------|---------------------------------------------------------------------------------------------|
| <p>Rule 11:</p> $A + \bar{A}B = A + B$ | <p>Timing diagram for <math>B = 0</math>:</p> <p>Timing diagram for <math>B = 1</math>:</p> |

**Further Investigation Results:****TABLE 7-5**

Truth table for Figure 7-5

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

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

**TABLE 7-6**

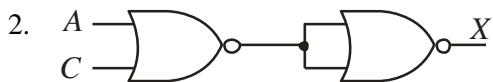
Truth table for Figure 7-6

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

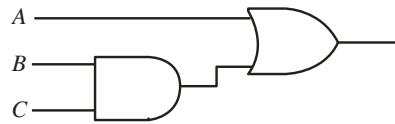
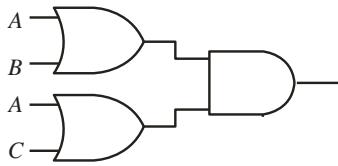
$$X = \bar{A}B + \bar{C}$$

**Evaluation and Review Questions:**

1. 
$$\begin{aligned} A(A + B) + C &= AA + AB + C \text{ (distribution)} \\ &= A + AB + C \text{ (rule 7)} \\ &= A + C \text{ (rule 10)} \end{aligned}$$



3. Rule 12:  $(A + B)(A + C) = A + BC$



4. Because the two circuits have identical truth tables, they perform the same logic.

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

5.  $(\overline{A} \cdot 1) = A + 0 = A$

6. Answers vary but should follow a logical sequence. An example is:

1. Verify that each IC has power and ground connected and that it is the correct voltage.
2. Select an input that should turn on the LED (such as all switches closed to ground.) With this input, verify that the  $A$ ,  $B$ , and  $C$  inputs to their respective gates are all LOW.
3. With all inputs LOW, verify that the output of the 4069 gates are both HIGH and that the output of the 4071 is HIGH. Verify that the output of the 4081 is HIGH.
4. Check that the LED is inserted in the correct direction and that there is a path from the output of the 4081 through a  $1.0\text{ k}\Omega$  resistor through the LED to ground.

## Experiment 8: Logic Circuit Simplification

### Data and Observations

Note: Truth tables 8-2 and 8-3 are identical. Only Table 8-2 is shown.

**TABLE 8-2**

Truth table for BCD invalid code detector

| Inputs |   |   |   | Output |
|--------|---|---|---|--------|
| D      | C | B | A | 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 | 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 | 1      |
| 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      |

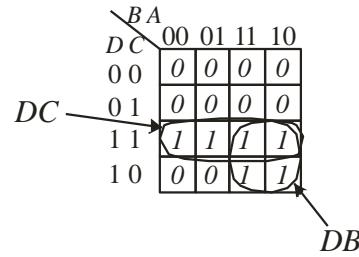


FIGURE 8-3  
Karnaugh Map of truth table for the BCD invalid code detector

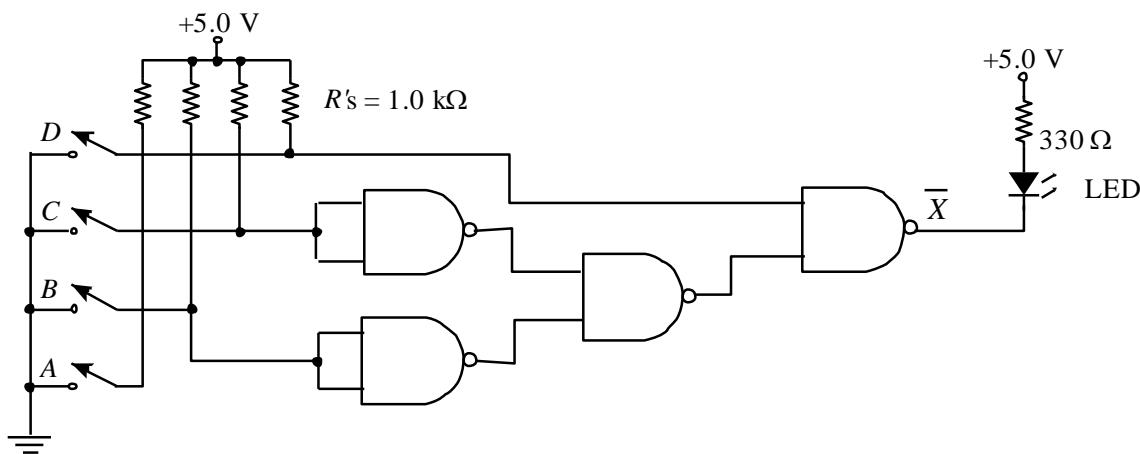
Minimum sum-of-products read from map:

$$X = \underline{DB + DC}$$

Factoring D from both product terms gives:

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

Step 5: Circuit for BCD invalid code detector (replacing OR gate with equivalent NAND gates):



Note that the A switch is not used in the invalid code detector.

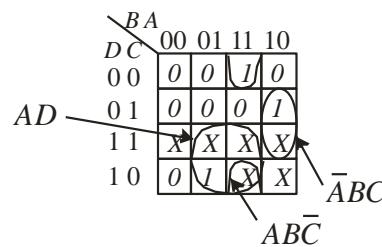
**TABLE 8-4**

| Problem Number | Problem                                                                                | Effect                                                                                                                                  |
|----------------|----------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| 1              | The pull-up resistor for the $D$ switch is open.                                       | <i>The D input will be an invalid HIGH when the D switch is open. The circuit will perform normally except is susceptible to noise.</i> |
| 2              | The ground to the NAND gate in Figure 8-4 is open.                                     | <i>The LED will not turn on under any conditions since there is no return path for LED current.</i>                                     |
| 3              | A 3.3 k $\Omega$ resistor was accidentally used in place of the 330 $\Omega$ resistor. | <i>Circuit operates normally but LED is slightly dimmer (may depend on the LED).</i>                                                    |
| 4              | The LED was inserted backward.                                                         | <i>The LED will not turn on under any conditions.</i>                                                                                   |
| 5              | Switch A is shorted to ground.                                                         | <i>The A input is not used so it will have no effect.</i>                                                                               |

**Further Investigation Results:****TABLE 8-5**

Truth table for BCD numbers divisible by three

| Inputs |     |     |     | Output |
|--------|-----|-----|-----|--------|
| $D$    | $C$ | $B$ | $A$ | $X$    |
| 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   | 1      |
| 0      | 1   | 1   | 1   | 0      |
| 1      | 0   | 0   | 0   | 0      |
| 1      | 0   | 0   | 1   | 1      |
| 1      | 0   | 1   | 0   | X      |
| 1      | 0   | 1   | 1   | X      |
| 1      | 1   | 0   | 0   | X      |
| 1      | 1   | 0   | 1   | X      |
| 1      | 1   | 1   | 0   | X      |
| 1      | 1   | 1   | 1   | X      |

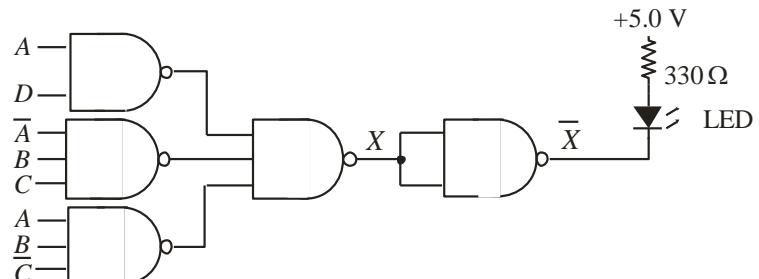


**FIGURE 8-5**  
Karnaugh Map of truth table for BCD numbers divisible by three.

Minimum sum-of-products read from map:

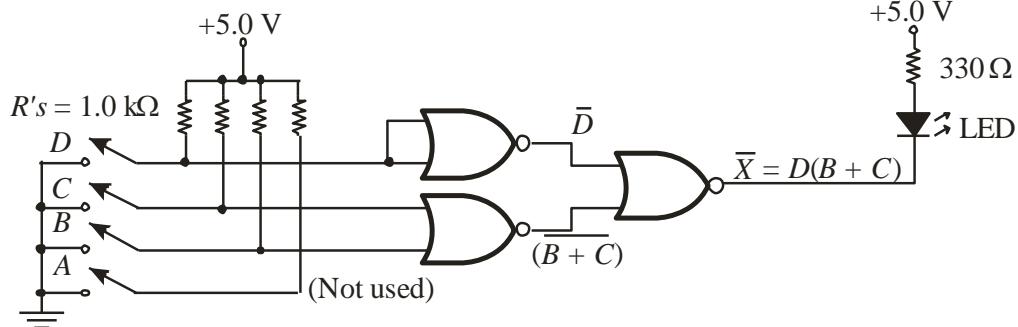
$$X = \underline{AD} + \underline{ABC} + \underline{\bar{A}\bar{B}\bar{C}}$$

Circuit:

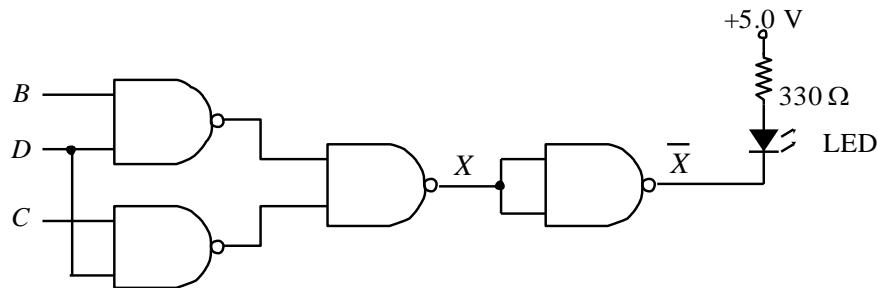


**Evaluation and Review Questions:**

1. The path from the OR gate to the NAND gate may be open or the inputs to the NAND gate may both be shorted to  $D$ . To troubleshoot the problem, put the circuit in state 1000 or 1001 and test the logic at the OR gate output and at the input of the NAND gate.
2. Equivalent circuit for Figure 8-4 using only NOR gates:



3. The  $A$  input was not connected in the circuit in Figure 8-4 because the  $A$  variable was eliminated by Boolean algebra from the output expression.
4.  $\bar{X} = \bar{D}(\bar{B}\bar{C})$   
 $X = \overline{\bar{D}(\bar{B}\bar{C})}$   
 $X = D + BC$
5. Sum of products (SOP) form  $X = BD + CD$ . Product of sums (POS form)  $X = D(B+C)$ . This can be done by factoring or by reading the 0's directly from the map (see Appendix B of text).
6. Implementation from the expression in step 2 ( $X = BD + CD$ ) is shown in the schematic below:



## Experiment 9: The Perfect Pencil Machine

The model-2 of the perfect pencil machine is a combinational logic circuit that delivers a pencil or change for certain combinations of inputs representing coin switches. Coin switches are provided for one nickel, ( $N_1$ ), two dimes ( $D_1$  and  $D_2$ ), and a quarter ( $Q$ ). (Only one nickel switch is used to simplify the problem). The two dime switches are stacked on top of each other such that  $D_1$  is always activated before  $D_2$ . The truth table has a number of don't care (X) entries, considered to be impossible inputs as described in the experiment. For example,  $D_2$  cannot be inserted in the machine until  $D_1$  has activated its switch and more than two coins cannot be entered in the machine. These "don't cares" allow the logic to be simplified considerably.

**TABLE 9-2**

Truth table for model-2 perfect pencil machine

| Inputs |       |       |     | Outputs |      |        |        |
|--------|-------|-------|-----|---------|------|--------|--------|
| $N_1$  | $D_1$ | $D_2$ | $Q$ | $P$     | $NC$ | $DC_1$ | $DC_2$ |
| 0      | 0     | 0     | 0   | 0       | 0    | 0      | 0      |
| 0      | 0     | 0     | 1   | 1       | 0    | 1      | 0      |
| 0      | 0     | 1     | 0   | X       | X    | X      | X      |
| 0      | 0     | 1     | 1   | X       | X    | X      | X      |
| 0      | 1     | 0     | 0   | 0       | 0    | 0      | 0      |
| 0      | 1     | 0     | 1   | 1       | 0    | 1      | 1      |
| 0      | 1     | 1     | 0   | 1       | 1    | 0      | 0      |
| 0      | 1     | 1     | 1   | X       | X    | X      | X      |
| 1      | 0     | 0     | 0   | 0       | 0    | 0      | 0      |
| 1      | 0     | 0     | 1   | 1       | 1    | 1      | 0      |
| 1      | 0     | 1     | 0   | X       | X    | X      | X      |
| 1      | 0     | 1     | 1   | X       | X    | X      | X      |
| 1      | 1     | 0     | 0   | 1       | 0    | 0      | 0      |
| 1      | 1     | 0     | 1   | X       | X    | X      | X      |
| 1      | 1     | 1     | 0   | X       | X    | X      | X      |
| 1      | 1     | 1     | 1   | X       | X    | X      | X      |

$P = \text{Pencil}$   
 $\cancel{D_2} Q$   

|        |    |    |    |    |
|--------|----|----|----|----|
| $ND_1$ | 00 | 01 | 11 | 10 |
| 00     | 0  | 1  | X  | X  |
| 01     | 0  | 1  | X  | 1  |
| 11     | 1  | X  | X  | X  |
| 10     | 0  | 1  | X  | X  |

$NC = \text{Nickel Change}$   
 $\cancel{D_2} Q$   

|        |    |    |    |    |
|--------|----|----|----|----|
| $ND_1$ | 00 | 01 | 11 | 10 |
| 00     | 0  | 0  | X  | X  |
| 01     | 0  | 0  | X  | 1  |
| 11     | 0  | X  | X  | X  |
| 10     | 0  | 1  | X  | X  |

$P = ND_1 + D_2 + Q$

$NC = NQ + D_2$

$DC_1 = \text{First dime change}$   
 $\cancel{D_2} Q$   

|        |    |    |    |    |
|--------|----|----|----|----|
| $ND_1$ | 00 | 01 | 11 | 10 |
| 00     | 0  | 1  | X  | X  |
| 01     | 0  | 1  | X  | 0  |
| 11     | 0  | X  | X  | X  |
| 10     | 0  | 1  | X  | X  |

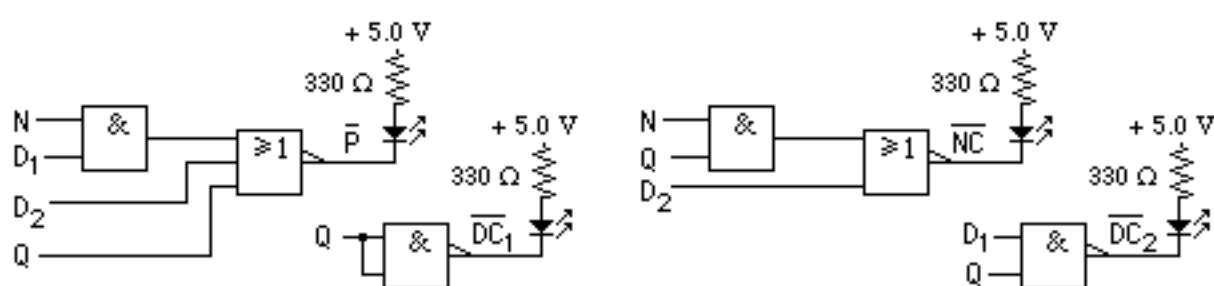
$DC_2 = \text{Second dime change}$   
 $\cancel{D_2} Q$   

|        |    |    |    |    |
|--------|----|----|----|----|
| $ND_1$ | 00 | 01 | 11 | 10 |
| 00     | 0  | 0  | X  | X  |
| 01     | 0  | 1  | X  | 0  |
| 11     | 0  | X  | X  | X  |
| 10     | 0  | 0  | X  | X  |

$DC_1 = Q$

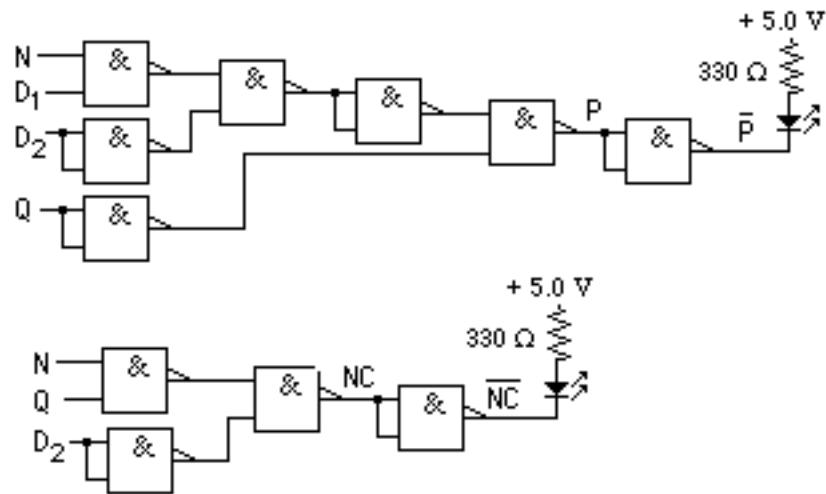
$DC_2 = D_1 Q$

A simplified implementation is shown (switches not shown). The output logic from the maps has been inverted to light LEDs with a LOW rather than a HIGH in keeping with TTL specified  $I_O$  values.



**Further Investigation Results:**

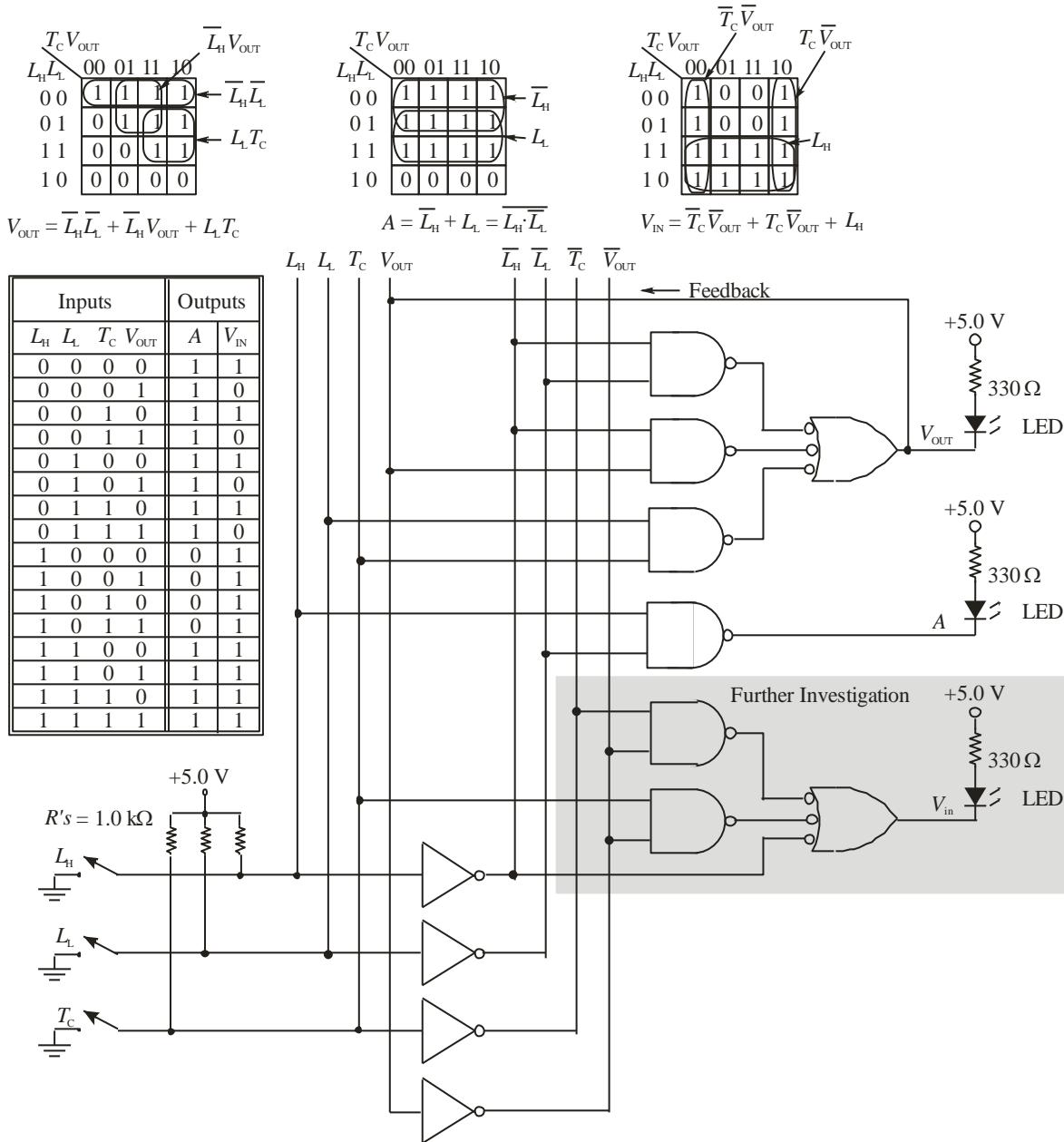
This investigation requires the student to modify his or her experiment to use only 2-input NAND gates. The two circuits for dime change logic already meet that requirement. Modifications for the pencil and nickel change logic are shown below:



## Experiment 10: The Molasses Tank

The model-2 Molasses Tank controller uses standard combinational logic but one of the inputs to the truth table is the output valve logic,  $V_{OUT}$ , to produce latching action using feedback. The truth table for the output valve is given in the experiment and the truth table for the alarm,  $A$ , and the input valve,  $V_{IN}$ , are given here. (The input valve is implemented in the Further Investigation with an extra Karnaugh map provided in the manual for this). Maps for these three outputs are also shown. (The heater output was not required in the experiment, but could be implemented as  $H = T_C \cdot L_L$ .) A circuit implementation using inverters and NAND gates is shown (note that  $A$  is implemented after applying DeMorgan's theorem). Active LOW outputs are used for the LEDs as given in the truth tables. Other implementations are possible.

Multisim files for the Model-1 controller are available on the website that was described previously. The file Exp-10nf illustrates how the feedback avoids the oscillation that could occur if the high level detector was the only control for the outlet valve by not closing the valve until the lower level is sensed.



## Experiment 11: Adder and Magnitude Comparator

### Data and Observations

The connections to the adder needed to complete the circuit in Figure 11-2 for a binary to excess-three conversion circuit are:

| input:         | connect to:      |
|----------------|------------------|
| B <sub>4</sub> | A>B              |
| B <sub>3</sub> | Ground           |
| B <sub>2</sub> | $\overline{A>B}$ |
| B <sub>1</sub> | +5.0 V           |
| C <sub>0</sub> | Ground           |

Results for the binary to excess-3 conversion are given in Table 11-4. The inverters are considered part of the display, therefore, logic shown on Table 11-4 is before the 7404 inverters (a logic 1 turns ON an LED).

### Further Investigation Results:

TABLE 11-4

Truth table for Figure 11-2

| Inputs<br>(Binary) |   |   |   | Outputs<br>(Excess-3) |   |   |   |
|--------------------|---|---|---|-----------------------|---|---|---|
| D                  | C | B | A | A'                    | D | C | B |
| 0                  | 0 | 0 | 0 | 0                     | 0 | 0 | 1 |
| 0                  | 0 | 0 | 1 | 0                     | 0 | 1 | 0 |
| 0                  | 0 | 1 | 0 | 0                     | 0 | 1 | 0 |
| 0                  | 0 | 1 | 1 | 0                     | 0 | 1 | 1 |
| 0                  | 1 | 0 | 0 | 0                     | 0 | 1 | 1 |
| 0                  | 1 | 0 | 1 | 0                     | 1 | 0 | 0 |
| 0                  | 1 | 1 | 0 | 0                     | 1 | 0 | 0 |
| 0                  | 1 | 1 | 1 | 0                     | 1 | 0 | 1 |
| 1                  | 0 | 0 | 0 | 0                     | 1 | 0 | 1 |
| 1                  | 0 | 0 | 1 | 0                     | 1 | 1 | 0 |
| 1                  | 0 | 1 | 0 | 1                     | 0 | 0 | 1 |
| 1                  | 0 | 1 | 1 | 1                     | 0 | 1 | 0 |
| 1                  | 1 | 0 | 0 | 1                     | 0 | 1 | 0 |
| 1                  | 1 | 0 | 1 | 1                     | 0 | 1 | 1 |
| 1                  | 1 | 1 | 0 | 1                     | 0 | 1 | 1 |
| 1                  | 1 | 1 | 1 | 1                     | 1 | 1 | 0 |

TABLE 11-5

Truth table for overflow error.

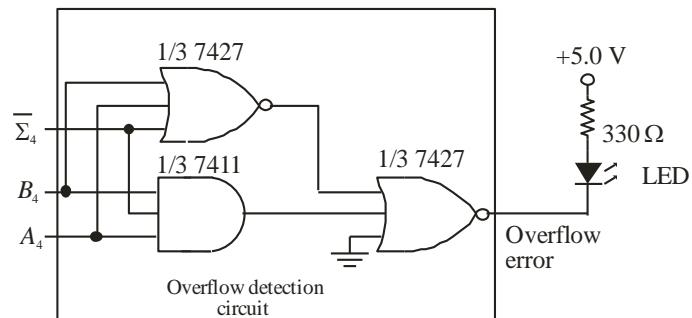
| Sign Bits      |                |            | Error |
|----------------|----------------|------------|-------|
| A <sub>4</sub> | B <sub>4</sub> | $\Sigma_4$ | X     |
| 0              | 0              | 0          | 0     |
| 0              | 0              | 1          | 1     |
| 0              | 1              | 0          | 0     |
| 0              | 1              | 1          | 0     |
| 1              | 0              | 0          | 0     |
| 1              | 0              | 1          | 0     |
| 1              | 1              | 0          | 1     |
| 1              | 1              | 1          | 0     |

| $A_4 B_4 \Sigma_4$ | 0 | 1 |
|--------------------|---|---|
| 0 0                | 0 | 1 |
| 0 1                | 0 | 0 |
| 1 1                | 1 | 0 |
| 1 0                | 0 | 0 |

FIGURE 11-5  
Karnaugh map for overflow error.

$$X = A_4 B_4 \Sigma_4 + \overline{A_4} B_4 \Sigma_4$$

$$X = A_4 B_4 \Sigma_4 + \overline{A_4} \overline{B_4} \Sigma_4$$



### Evaluation and Review Questions:

1. a. The input to the 7404 will see an illegal logic HIGH, causing the A' LED to be on.  
b. Approx 1.6 V.
2. The carry-out ( $C_4$ ) of the lower order adder is connected to the carry-in ( $C_0$ ) of the higher order adder. The schematic is shown in text as Figure 6-12(a).
3. It is the carry-in.
4. Connect the  $B$  inputs of the upper 7485 to 1000 and the lower comparator  $B$  inputs to 1100. Connect cascading inputs on both so that  $A=B$  inputs are HIGH; all other cascading inputs are LOW.
5. To form the two's complement, invert the bits of the subtrahend and add one by connecting the carry-in to a HIGH.
6. An additional voting switch can be connected to the carry-in logic of the parallel adders.

## Experiment 12: Combinational Logic Using Multiplexers

### Data and Observations

**TABLE 12-1**

Truth table for 2-bit comparator  $A \geq B$ .

| Inputs |       |       | Output | Connect Data to: |
|--------|-------|-------|--------|------------------|
| $A_2$  | $A_1$ | $B_2$ | $B_1$  | $X$              |
| 0      | 0     | 0     | 0      | 1                |
| 0      | 0     | 0     | 1      | 0                |
| 0      | 0     | 1     | 0      | 0                |
| 0      | 0     | 1     | 1      | 0                |
| 0      | 1     | 0     | 0      | 1                |
| 0      | 1     | 0     | 1      | 1                |
| 0      | 1     | 1     | 0      | 0                |
| 0      | 1     | 1     | 1      | 0                |
| 1      | 0     | 0     | 0      | 1                |
| 1      | 0     | 0     | 1      | 1                |
| 1      | 0     | 1     | 0      | 1                |
| 1      | 0     | 1     | 1      | 0                |
| 1      | 1     | 0     | 0      | 1                |
| 1      | 1     | 0     | 1      | 1                |
| 1      | 1     | 1     | 0      | 1                |
| 1      | 1     | 1     | 1      | 1                |

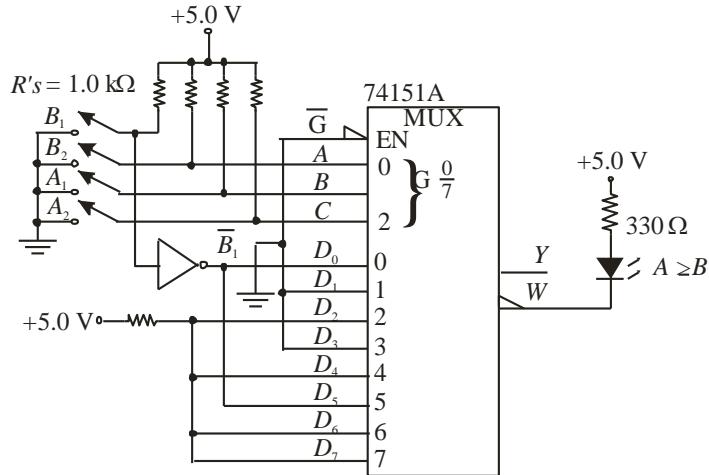


Figure 12-4

### Further Investigation Results:

**TABLE 12-2**

Truth table for even parity generator.

| Inputs |       |       | Output | Connect Data to: |
|--------|-------|-------|--------|------------------|
| $A_3$  | $A_2$ | $A_1$ | $A_0$  | $X$              |
| 0      | 0     | 0     | 0      | 0                |
| 0      | 0     | 0     | 1      | 1                |
| 0      | 0     | 1     | 0      | 1                |
| 0      | 0     | 1     | 1      | 0                |
| 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      | 0                |
| 1      | 1     | 0     | 1      | 1                |
| 1      | 1     | 1     | 0      | 1                |
| 1      | 1     | 1     | 1      | 0                |

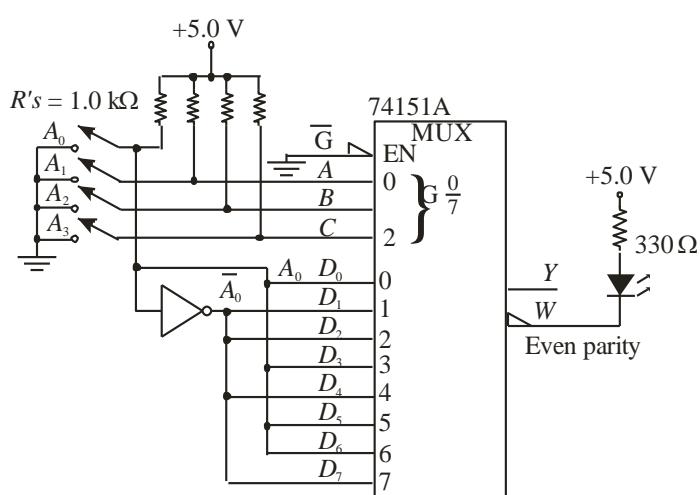
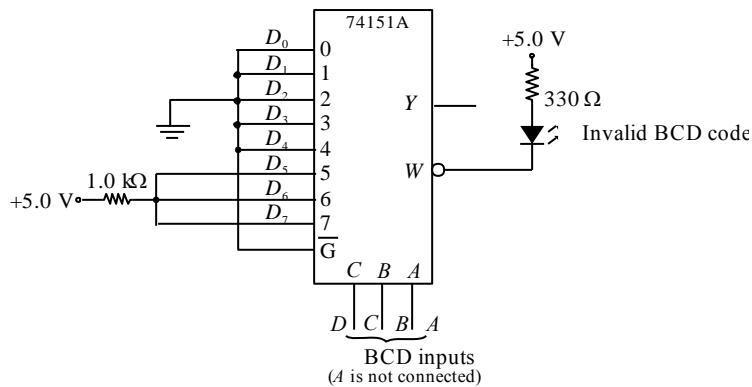


Figure 12-5

**Evaluation and Review Questions:**

1. The BCD invalid code detector can be designed as shown:

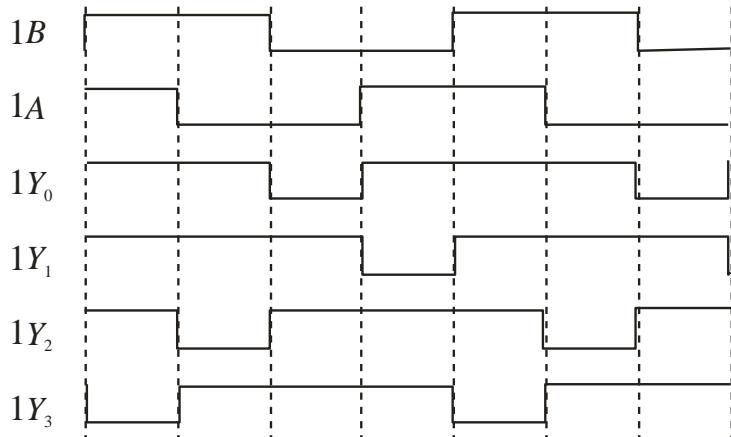
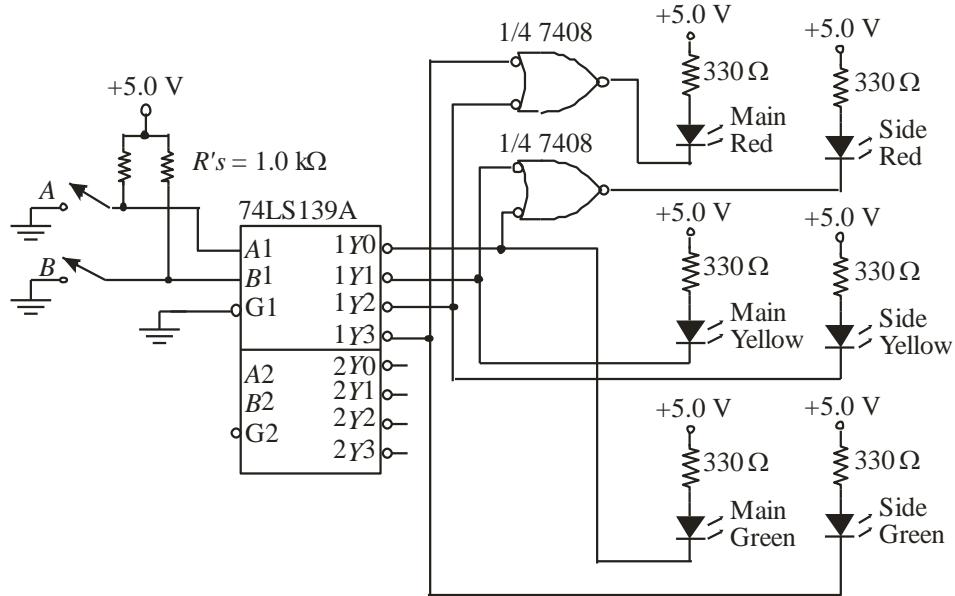


2.  $\bar{A}_0\bar{A}_1A_2 + A_0\bar{A}_1\bar{A}_2 + A_0\bar{A}_1A_2 + \bar{A}_0A_1A_2$
3. The input which affects the second half of the truth table is the  $A_2$  input. Place the circuit in a fault condition and check the inputs to the multiplexer beginning with the  $A_2$  input.
4. The inverter output is used with minterms 0, 1, 10, and 11. However it will be incorrect only when  $B_1$  should be HIGH, which is a fault condition for minterm 0 and 10. Place the circuit in one of these fault conditions and check the appropriate inputs to the multiplexer. Since the multiplexer input is incorrect, the troubleshooter is led to test the inverter and discover the fault.
5. The lines on the truth table that would be incorrect are  $A_2A_1B_2B_1 = 0\ 0\ 0\ 1$  and  $A_2A_1B_2B_1 = 1\ 0\ 1\ 1$ . All other lines will read correctly.
6. Odd parity can be obtained from the  $Y$  output of the 74151A.

## Experiment 13: Combinational Logic Using Demultiplexers

Traffic Light Output logic (Figure 13-6):

Connect  $1Y_1$  to Main Yellow and to Side Red through the 7408 AND gate. Connect  $1Y_3$  to Main Red through a 7408 AND gate and to Side Green. Connect  $1Y_2$  to Main Red through a 7408 AND gate and to Side Yellow. The enable (G1) is wired LOW with switches to the A<sub>1</sub> and B<sub>1</sub> select inputs. The circuit is shown below:



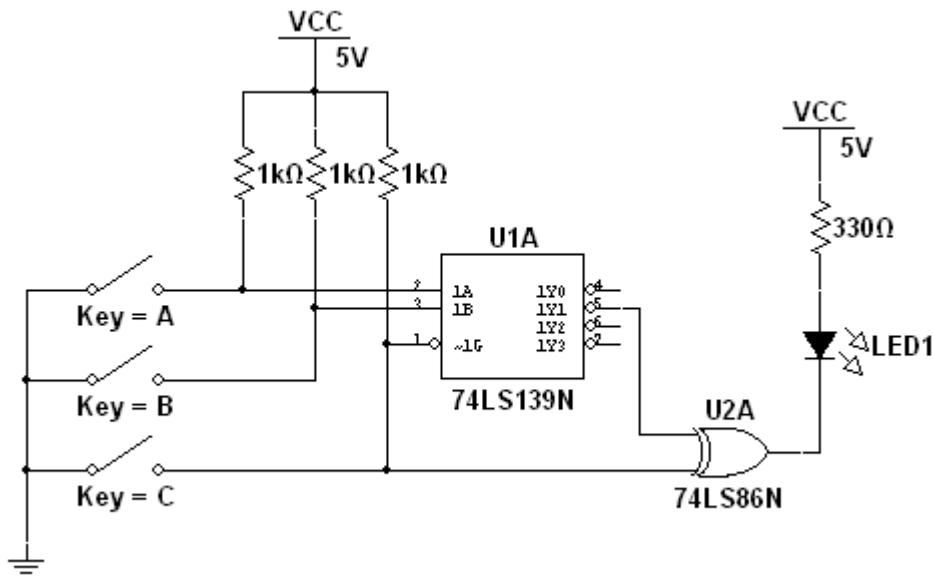
**Figure 13-8**  
Timing diagram for step 5

### Further Investigation Results:

The NAND gate serves as a decoder for a particular output state from the counter. The decoded output then enables the decoder and lights a corresponding LED. As long as the CLK input to the counter is above approximately 100 Hz, the LED appears to be on steady. Different LEDs can be selected by changing the decoder's inputs.

**Evaluation and Review Questions:**

1. Use one-half of the 74LS139A for the least significant four bits and the other half for the most significant four bits. Use the enable input on each decoder for the third select input, inverting it for the most significant four bits.
2. To emphasize the active-LOW inputs and outputs.
3. Gray code is useful to prevent glitches in the decoded outputs.
4. a. The input is an illegal HIGH, preventing the  $1Y_0$  and the  $1Y_1$  outputs from being selected.  
b. The input is LOW, preventing the  $1Y_2$  and the  $1Y_3$  outputs from being selected.  
c. The chip is not selected; therefore all outputs will be HIGH.
5. The sequence of the lights would be incorrect. This can be corrected by reversing the output logic for states 2 and 3. The disadvantage is possible glitches in the decoded outputs.
6. The LED will be on whenever  $C$  is open (HIGH) or if  $C$  and  $B$  are closed (LOW) and  $A$  is open (HIGH). (Multisim file is used for illustration.)



## Experiment 14: The D Latch and D Flip-Flop

Step 3 Observations: The output of the latch changes only when the switch makes initial contact with the A or B terminals. The latch does not change when the switch is "bounced".

Step 5 Observations: As long as the enable is HIGH, the output ( $Q$ ) follows the pulse generator input. When enable is LOW, changes on the D input are not seen on the output, however; the outputs can be latched by grounding either one momentarily, even when enable is not asserted.

Step 6 Observations: The burglar alarm basically illustrates the functions of the D latch. When enable is asserted HIGH, the opening of a door or window switch causes the alarm LED to latch on. Closing the door or window switch has no effect on the output. The alarm LED is turned off by closing all switches (or placing it in standby) and pressing the reset pushbutton.

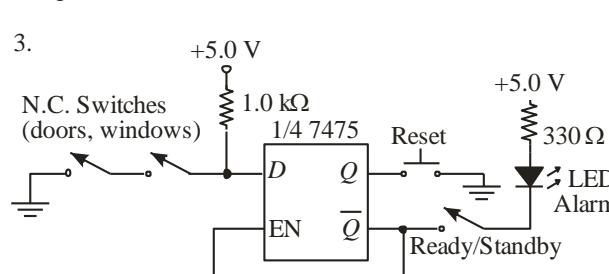
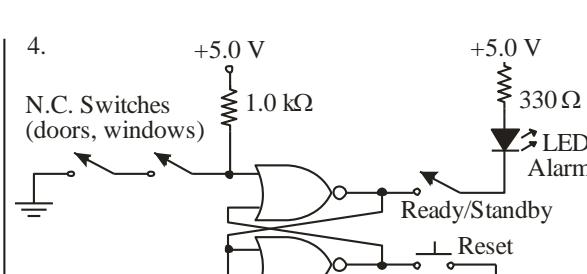
Step 7 and 8 Observations: The setup time specification for a 7474 is 20 ns. The four inverters provide sufficient setup time to read the pulse as a HIGH, causing the output to remain a constant HIGH. When the delay is removed, the output is a constant LOW.

Step 10 Observations: The D flip-flop needs setup time as shown with the clock delay circuit. When sufficient setup time is given, the output will go HIGH; otherwise, it is LOW. The preset and clear inputs are asserted with a LOW input and are observed to be asynchronous inputs. With the clock delay in place and connecting the  $Q$  to D, the scope signal can be made to appear as if timing is changing when in fact it is not. (Trigger from the clock to show the timing problem).

### Further Investigation Results:

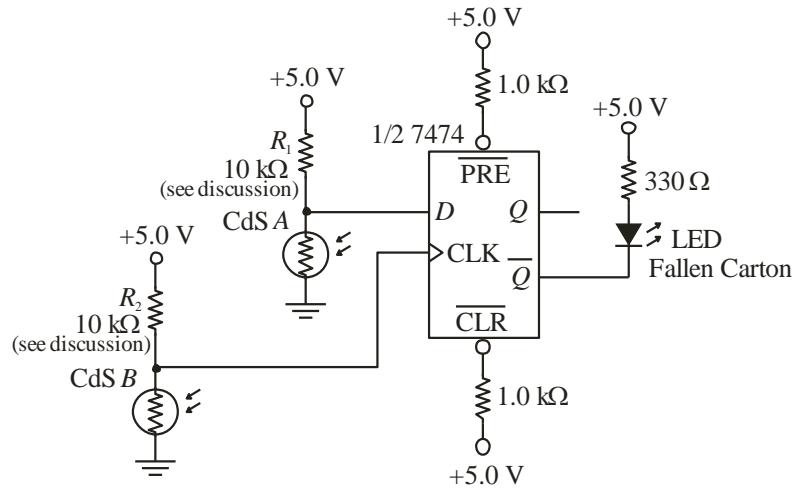
The serial parity test circuit changes the parity every time a logic 1 is received but does not change parity when a logic 0 is received. The circuit has an advantage over the parallel parity test circuit in Floyd's text only when the data is already in serial form and needs to be tested. In this case, the serial tester circuit is simpler and just as fast.

### Evaluation and Review Questions:

1. The switch debounce circuit requires two contacts (throws) because the output will not change if a single contact were used.
2. No. With NOR gates, the output would change as the switch pole is first moved from the contact but the other NOR gate output would be unaffected – causing bouncing to be seen on the output. A similar problem would occur on the other contact.
3. 
4. 
5. There are several possibilities. Among them are:
  1. Reset or enable switch stuck LOW.
  2. Open path to the alarm LED including open resistor or open diode.
  3. Faulty NAND gate (open output on lower right NAND gate, shorted output on top right gate, etc.)  
The most likely fault is an open connection rather than a faulty component.
6. A latch would oscillate very rapidly whenever the data was HIGH.

## Experiment 15: The Fallen Carton Detector

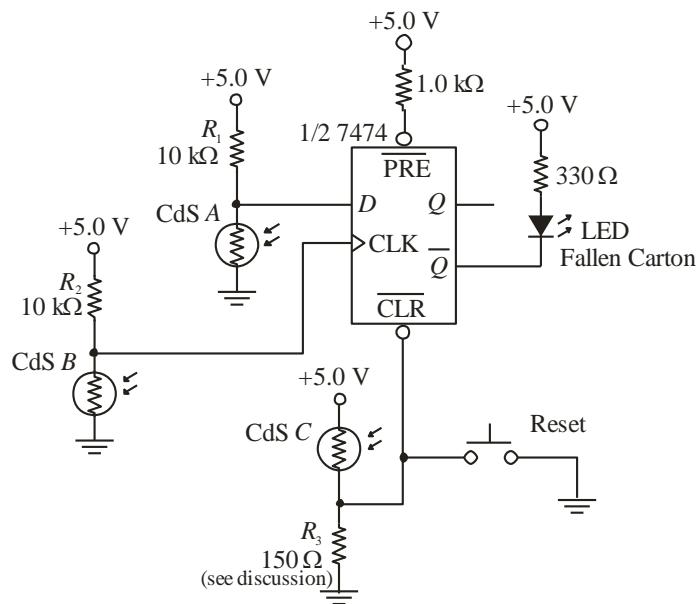
The fallen carton detector is a sequential logic circuit which is designed to detect if a carton, moving on a conveyor belt, is upright or has fallen over. If the carton has fallen, the circuit is to trip a solenoid (indicated by an LED). The circuit can be implemented with a D flip-flop as shown. Student circuits may vary.



Students will need to experiment to determine optimum series resistors (shown as  $R_1$  and  $R_2$ ) to use with the photocells and with the particular room lighting. With a CdS photocell, a value of about  $10\text{ k}\Omega$  will work in normal room light when connected to a 7474 D flip-flop. The voltage will be LOW when uncovered ( $< 0.2\text{ V}$ ) and will be HIGH ( $> 3.0\text{ V}$ ) when covered.

### Further Investigation Results:

As the carton is sent into the reject hopper, a photocell can sense its presence. The photocell output can be connected to the  $\overline{\text{CLR}}$  input of the D flip-flop to produce a LOW input when it is covered by reversing the photocell and the series resistor. Because TTL logic requires more current to pull it down, the value of  $R_3$  must be much smaller than in the first circuit. A value of  $150\text{ }\Omega$  worked well with the photocell and the particular room light tested. A reset button is also connected to the  $\overline{\text{CLR}}$  input of the D flip-flop. The modified circuit is shown below.



## Experiment 16: The J-K Flip-Flop

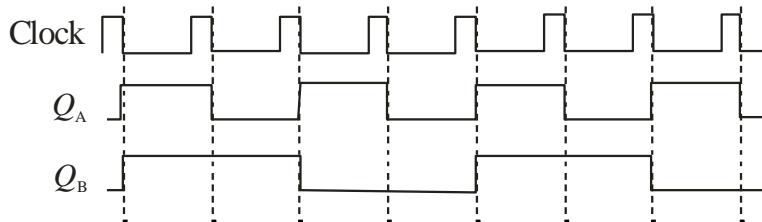
### Data and Observations

Step 1 Observations: The preset and clear inputs are asserted with a LOW input and are asynchronous. When both are LOW at the same time, both  $Q$  and  $\bar{Q}$  are HIGH.

Step 2 Observations: The truth table is verified –  $Q$  follows  $J$  when the  $J$  and  $K$  inputs are different. The output does not change when both inputs are LOW and the output toggles when the inputs are both HIGH. The output is observed to change on the trailing edge of the clock. The output duty cycle is seen to be 50%.

Step 3 Observations: The circuit toggles for each clock pulse.

Step 4 Observations: The ripple counter divides the input frequency in two for each stage as shown in the timing diagram.  $Q_A$  is  $\frac{1}{2}$  the Clock frequency;  $Q_B$  is  $\frac{1}{4}$  the clock frequency.



Plot 1

### Further Investigation Results:

The older 7476 (now obsolete) measured and specified values are:

Measured  $t_{PLH} = 20$  ns. Manufacturer's specified  $t_{PLH} = 16$  ns (typ), 25 ns (max).

Measured  $t_{PHL} = 25$  ns. Manufacturer's specified  $t_{PHL} = 25$  ns (typ), 40 ns (max).

The 74LS76A is faster with specified values as follows:

Manufacturer's specified  $t_{PLH} = 15$  ns (typ), 20 ns (max). (Data is from ON Semiconductor.)

Manufacturer's specified  $t_{PHL} = 15$  ns (typ), 20 ns (max). (Data is from ON Semiconductor.)

### Evaluation and Review Questions:

1. An asynchronous input is independent of the clock and can affect the output without the presence of a clock signal. A synchronous input can only affect the output when a clock signal is present.
2. a. Assert the preset input ( $\overline{PRE}$ ) which is asynchronous.  
b. Clock the flip-flop when  $J$  is HIGH and  $K$  is LOW.
3. The output is latched.
4. If  $Q$  is connected to  $J$  and  $\bar{Q}$  is connected to  $K$ , the clock will not change the output logic.
5. The CLK input is open; the  $Q$  output is shorted LOW; the  $\overline{CLR}$  input is shorted LOW.
6. Green LED could be open,  $390\ \Omega$  resistor open, +5.0 V supply not connected to  $390\ \Omega$  resistor, bad 74LS76A.

## Experiment 17: One Shots and Astable Multivibrators

**TABLE 17-1**

Data for 74121 monostable multivibrator

| Quantity                      | Computed Value | Measured Value |
|-------------------------------|----------------|----------------|
| Timing Resistor, $R_T$        | <b>7.14 kΩ</b> | <b>6.89 kΩ</b> |
| External Capacitor, $C_{EXT}$ | 0.01 μF        | <b>0.01 μF</b> |
| Pulse width, $t_W$            | <b>48.2 μs</b> | <b>46.0 μs</b> |

**TABLE 17-2**

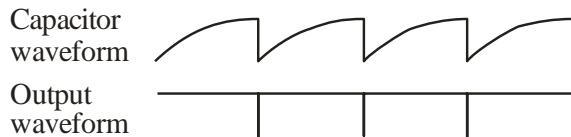
Data for 555 timer as an astable multivibrator

| Quantity         | Computed Value  | Measured Value  |
|------------------|-----------------|-----------------|
| Resistor, $R_1$  | 7.5 kΩ          | <b>7.46 kΩ</b>  |
| Resistor, $R_2$  | 10 kΩ           | <b>10.1 kΩ</b>  |
| Capacitor, $C_1$ | 0.01 μF         | <b>0.01 μF</b>  |
| Frequency        | <b>5.24 kHz</b> | <b>5.50 kHz</b> |
| Duty Cycle       | <b>0.64</b>     | <b>0.66</b>     |

Step 3. Input logic levels and generator connection: To trigger the 74121 with a leading edge clock, either  $A_1$  or  $A_2$  must be held LOW and the signal generator is connected to the  $B$  input.

Step 5. Observations as the frequency is raised to 50 kHz: The pulse width remains the same; it does not stay HIGH when the frequency is increased.

Step 8.



**Plot 1**

Step 9. Observations with a short across  $R_2$ : The output stays HIGH except for a short "spike" that appears each time a new cycle is initiated.

Step 10. Various solutions are possible. One solution is to modify the circuit shown in Figure 17-2 by changing  $R_2$  to 3.3 kΩ. (The calculated value is 3.45 kΩ.)

### Further Investigation Results:

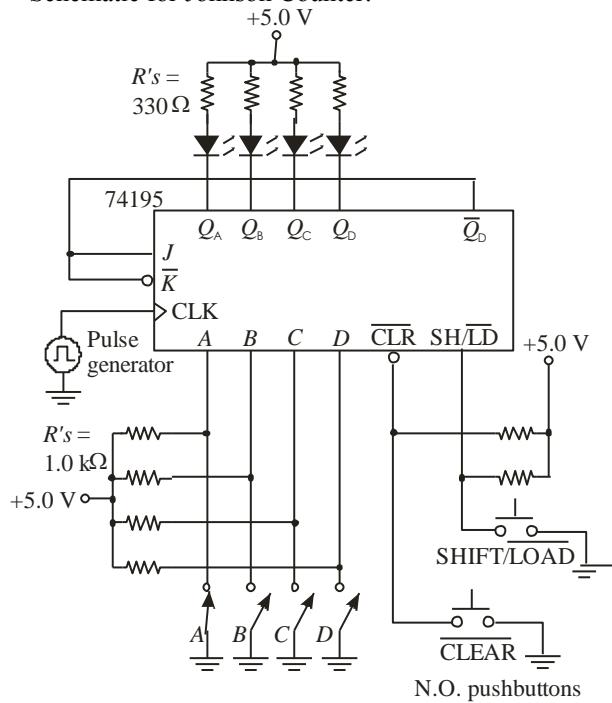
Both timers need trailing edge triggers that can be obtained on the 74121 by connecting  $A_1$  and  $A_2$  together and connecting  $B$  HIGH (see Figure 17-1, 3 lines from the end of the table). The manufacturer specifies that the external resistor cannot be larger than 40 kΩ. An approximate 4 s pulse can be obtained by selecting a 150 μF capacitor and a 39 kΩ resistor as shown in Figure 17-1 (computed = 4.1 s). A 25 s pulse can be obtained by selecting a 1000 μF capacitor and a 36 kΩ resistor (computed = 25.2 s).

### Evaluation and Review Questions:

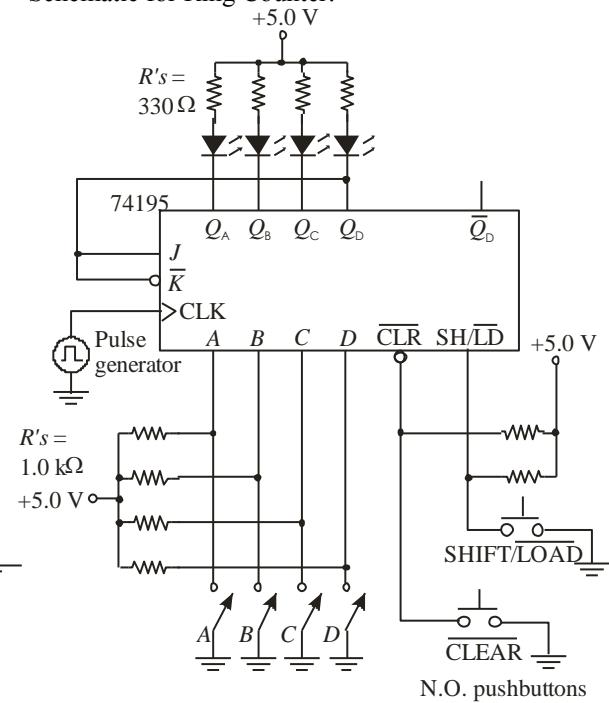
1. A non-retriggerable monostable multivibrator ignores any triggers that occur during the timing cycle.
2. a. 0.036 μF.  
b. Select an external variable resistor. For the computed capacitance, the external resistance needs to vary from 2 kΩ to 10 kΩ.
3. Largest recommended resistor is 40 kΩ. The largest capacitor is 1000 μF. With this combination the pulse width is 28 seconds.
4. Duty cycle = 50.1%; frequency = 400 Hz.
5. The required frequency is 0.083 Hz. The sum of  $R_1 + 2R_2 = 1.72 \text{ M}\Omega$ . From the duty cycle equation, the resistors are found to be:  $R_1 = 0.36 \text{ M}\Omega$  and  $R_2 = 0.68 \text{ M}\Omega$ .
6. The voltage ranges from +5.0 V to +10.0 V.

## Experiment 18: Shift Register Counters

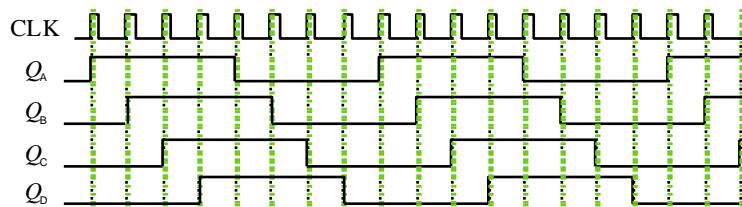
Schematic for Johnson Counter:



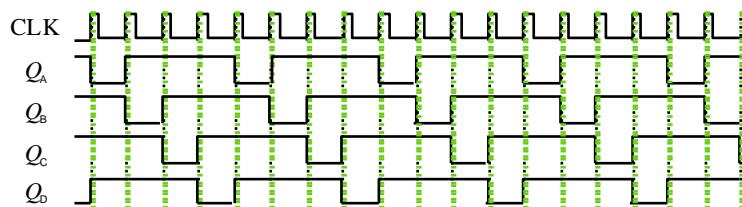
Schematic for Ring Counter:



Timing diagram  
for Johnson counter:

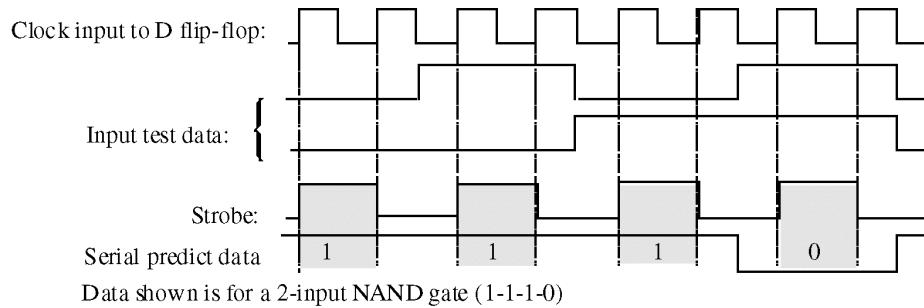


Timing diagram for ring  
counter loaded with 1110:



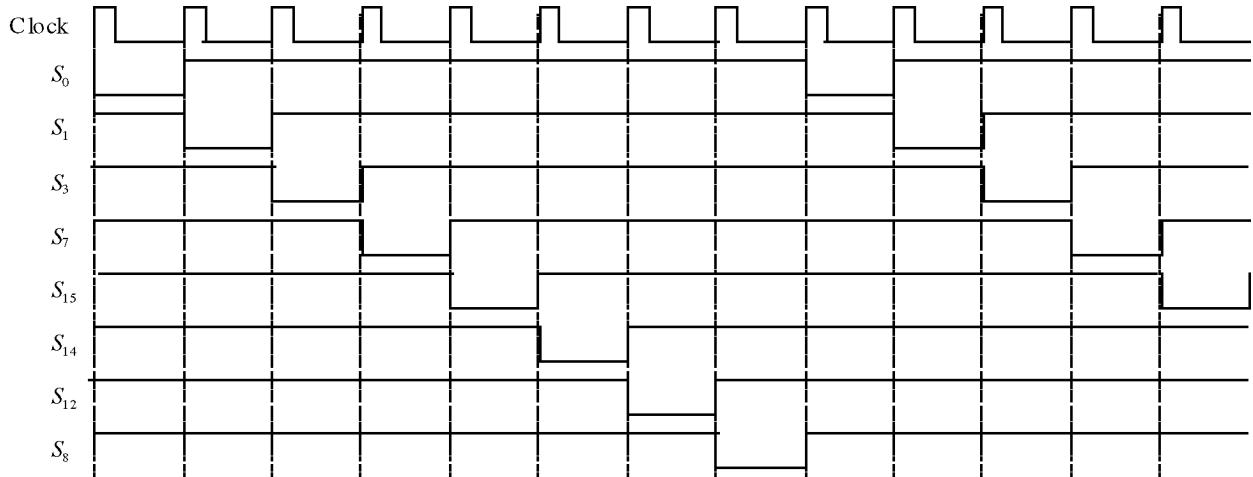
### Further Investigation Results

The **Serial predict data** is compared with the data from the device when the **Strobe** line goes HIGH as shown in the following timing diagram:



### Evaluation and Review Questions:

1. The  $Q_D$  output of the first 74195 is connected to the  $J-K$  inputs of the second. The  $\bar{Q}_D$  output from the second 74195 is connected to the  $J-K$  inputs of the first.
2. To prevent the data from shifting more than one position for each clock pulse.
3. a. 101-110-011. b. 101-010-101. (An interesting but generally not useful result!).
4. After the stored data has been clocked out, the same input data will appear at all outputs.
5. The 8 active states are shown in the diagram. (Repeats after 8 states).



6. Politely tell the boss that a normal ring counter is self-decoding and that a decoder is not necessary.

## Experiment 19: Application of Shift Register Circuits

### Data and Observations:

**Step 2:** After loading the data, the start bit flip-flop is observed to have a LOW (LED on). Data from the shift register can be seen to move to the right at each clock pulse. The data at the  $Q_A$  output moves to the *Serial data out* position after 5 clock pulses. The register is then filled with HIGHs (LEDs all off) until reloaded.

**Step 3:** Sample calculations of the resistors for the 555 configured as an astable multivibrator are given. (See Figure 17-2 for the circuit for the astable configuration.)

$$f = \frac{1}{0.7(R_1 + 2R_2)C} = \frac{1}{1 \text{ ms}} \quad t_H = 0.8 \text{ ms} \quad t_L = 0.2 \text{ ms}$$

Let  $C = 0.1 \mu\text{F}$ ; then

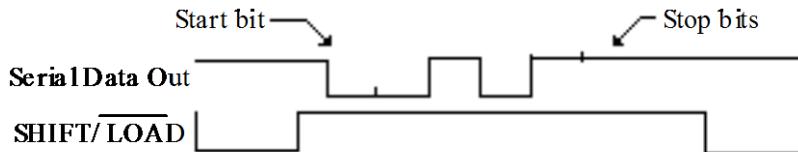
$$R_1 + R_2 = \frac{t_H}{0.7 C} = \frac{0.8 \text{ ms}}{0.7 (0.1 \mu\text{F})} = 11.4 \text{ k}\Omega$$

$$R_2 = \frac{t_L}{0.7 (0.1 \mu\text{F})} = 2.86 \text{ k}\Omega \quad (\text{choose } 2.7 \text{ k}\Omega)$$

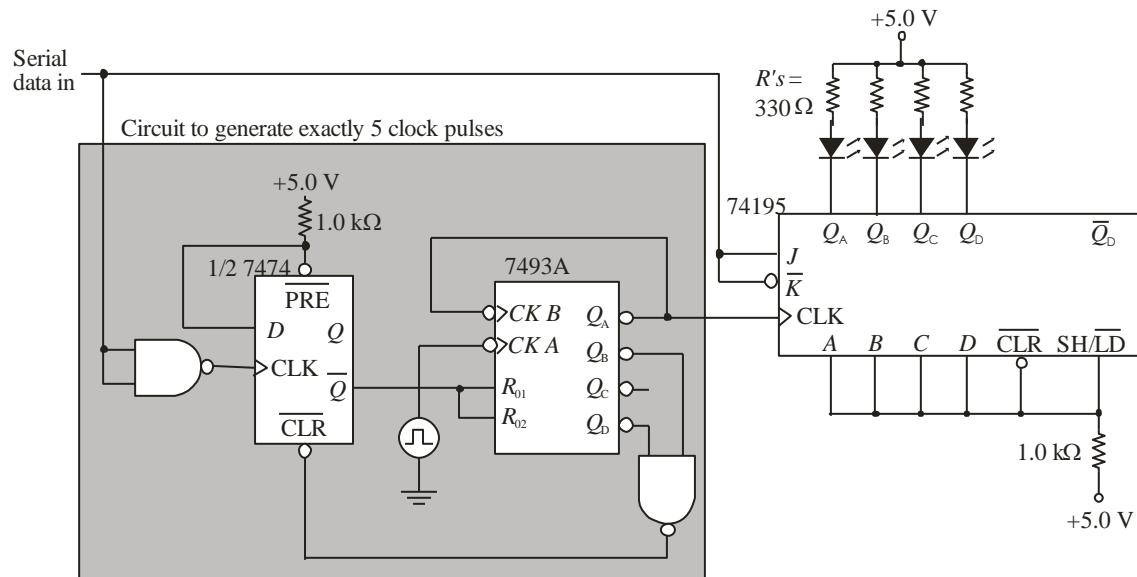
$$R_1 = 11.4 \text{ k}\Omega - 2.7 \text{ k}\Omega \quad (\text{choose } 9.1 \text{ k}\Omega)$$

With the values selected above, the duty cycle is 81% and the frequency is 985 Hz.

**Step 4:** Because of the asynchronous relation between the data clock and the **SHIFT/ LOAD** pulse, the time between the serial data out and the **SHIFT/ LOAD** and the start bit is not fixed.



**Step 5:** The circuit to send exactly five clock pulses is shown in the shaded box. When the start bit is detected, the data is shifted in the 74195. The frequency of the pulse generator is set to twice that of the transmitter in order to generate one complete pulse with every two clocks. (The 7493A counts to ten during this process.)



**Further Investigation Results:**

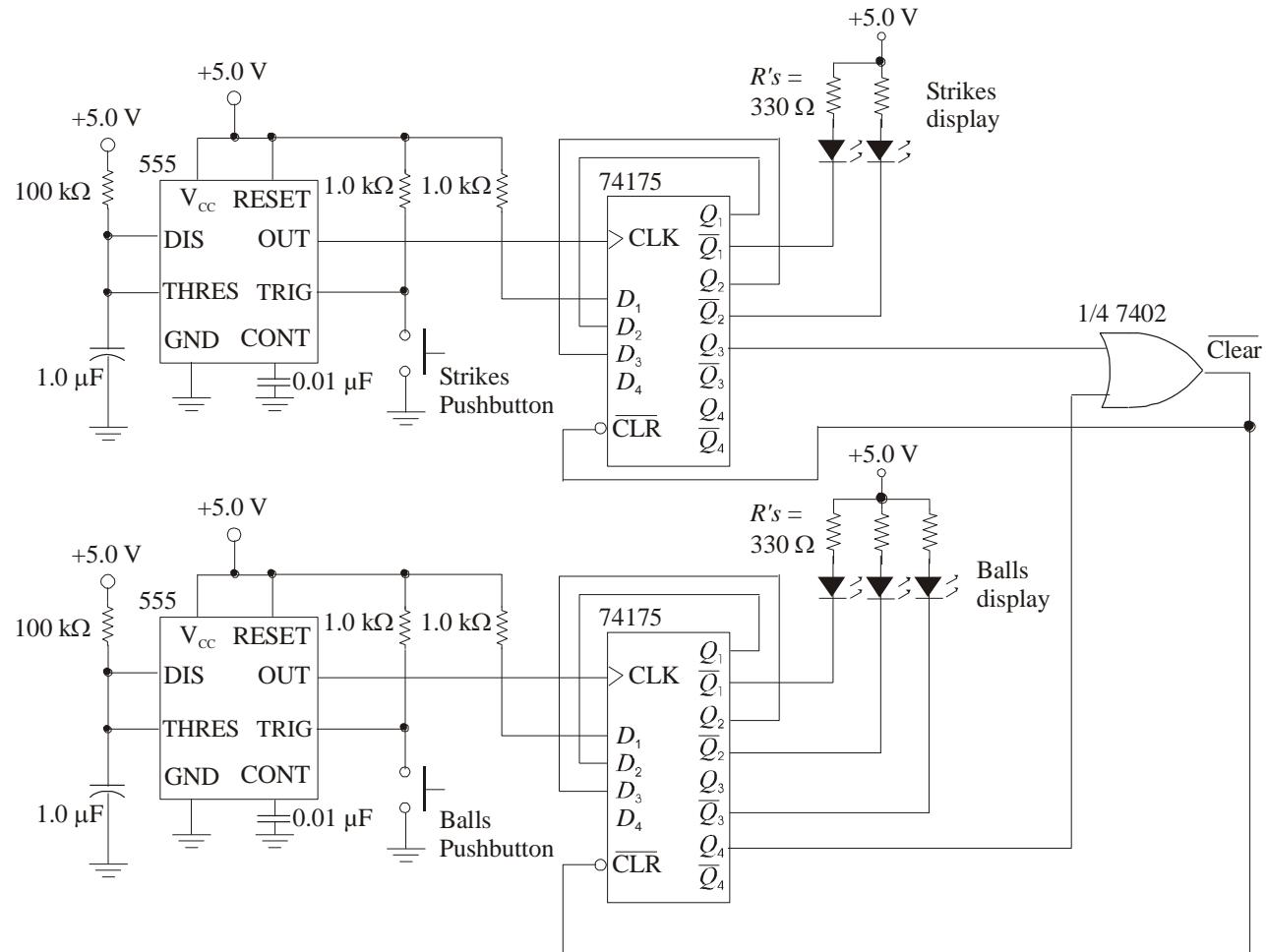
The receiver pulse generator is set twice as fast as the transmitter's pulse generator because the circuit that generates 5 clock pulses divides the receiver frequency by two. The generator frequencies should be close to the 2-1 ratio, but it is not necessary to be exact. Five data pulses are used to send the start bit right through the receiver (into the "bit bucket"), moving only the data that was set at the transmitter into the receiver. The stop bits are represented by HIGHs, and since the line is held HIGH (no transition), the receiver will not be clocked until a new start bit is received.

**Evaluation and Review Questions:**

1. The *J-K* inputs are connected HIGH. This HIGH is shifted into the 74195A at each clock pulse.
2. The first bit is a start bit which is not part of the data. The extra clock pulse causes it to be clocked through the receiver's shift register.
3. The start bit transition is a falling edge (HIGH to LOW). The NAND gate, which acts as an inverter, changes the active edge so that it can trigger the 7474 with a leading edge.
4.
  - a. The numbers to be added are entered into the input shift registers. The full adder produces the sum and the carry bit sequentially as each bit is shifted through the input registers. As the sum is produced, it is moved sequentially through the output shift register. The carry-out flip-flop stores the carry bit to be added to the next most significant bit.
  - b. The output shift register will contain 0011; the carry-out flip-flop will contain 1.
5. A 74195A shift register can be configured as a ring counter. The register is loaded with the pattern 1000 which is recirculated. The outputs are taken from either  $Q_A$  and  $Q_C$  or from  $Q_B$  and  $Q_D$ .
6. If the start bit flip-flop is loaded with a HIGH instead of a LOW, the start bit will not occur. This could occur if the clear input to the start bit flip-flop were open.

## Experiment 20: The Baseball Scoreboard

The baseball scoreboard is a sequential logic circuit that can be implemented with shift registers or with a counter. One possible solution, using 74175 quad D flip-flops connected as shift registers is shown. Switch debounce is provided by the 555 timers.



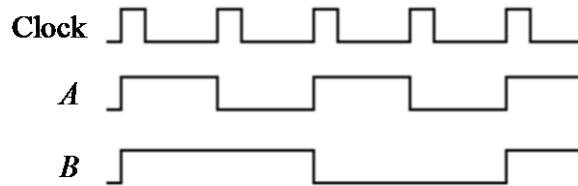
### Further Investigation Results (TTL)

Student answers will vary. The inning display can be set up as a 16-position shift register with a single LED on at any given time. Another approach is to use a 16-bit DMUX driven with a manually clocked 4-bit counter controlling the select inputs of the DMUX. Outs can be connected as a 2-bit shift register made from D flip-flops controlled by a debounced pushbutton, similar to the circuits shown above.

## Experiment 21: Asynchronous Counters

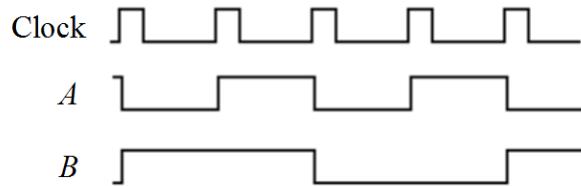
### Data and Observations

Waveforms from step 1:



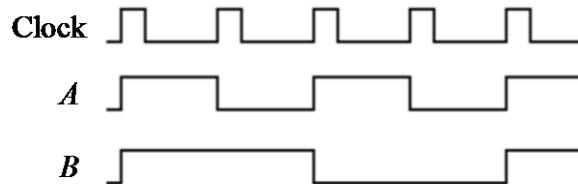
Counter is a down counter.

Waveforms from step 2:



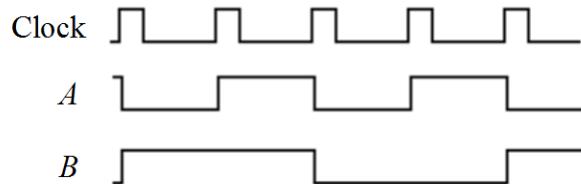
Counter is an up counter.

Waveforms from step 3: (same as step 1)



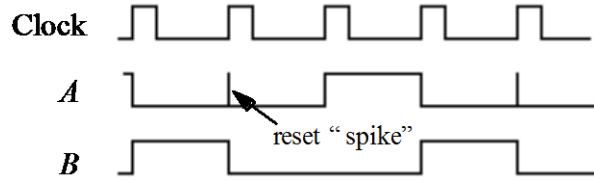
Counter is a down counter.

Waveforms from step 4: (same as step 2)



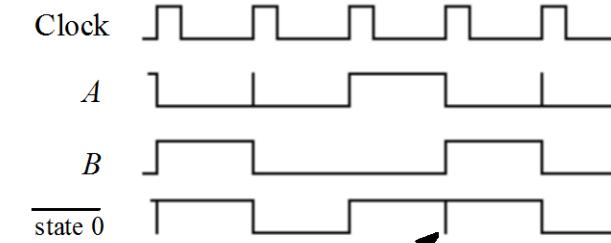
Counter is an up counter.

Waveforms from step 5:



Counter sequence is 0 - 1 - 2 - reset. The short "spike" is caused by the reset in state 3.

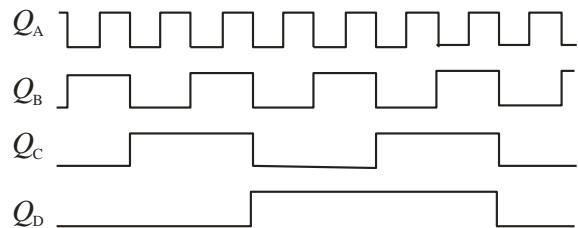
Waveforms from step 6:



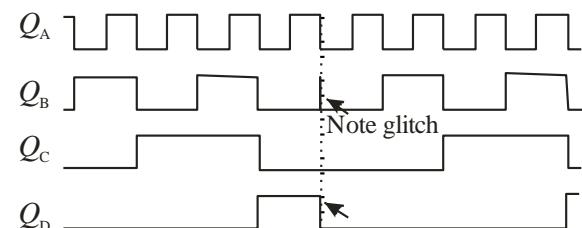
The decoded output shows a "glitch" when both A and B change together.

Notice that the "glitch" in step 5 is a consequence of detecting state 3 and using it to reset the counter. The "glitch" in the state zero decoded output is different. It is caused by the fact that the counter is momentarily in state zero due to propagation delay between the two flip-flops.

Waveforms from step 7:

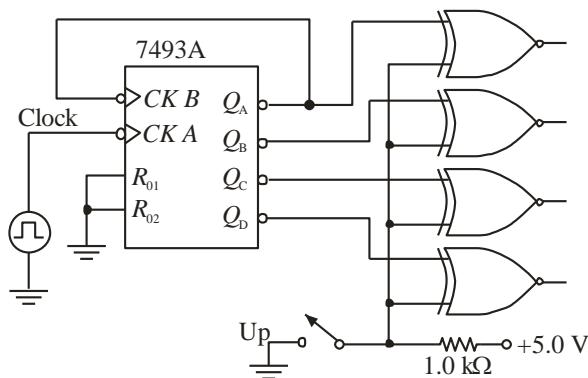


Waveforms from step 8:



The sequence is 0-1-2-3-4-5-6-7-8-9 - reset (momentarily in state 10)

### Further Investigation Results:

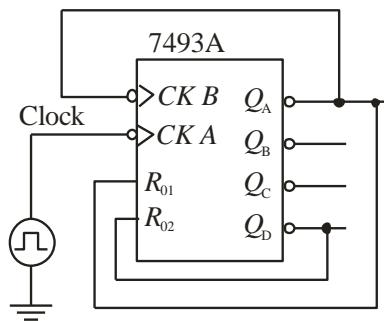


The XOR gates act as inverters when the Up switch is closed and act as buffers when it is open. By inverting or not inverting the waveforms, the counter can appear as an up or a down counter. The disadvantage of the method is that the reversal of the bits when the switch is open or closed will cause the count to change at the switching time.

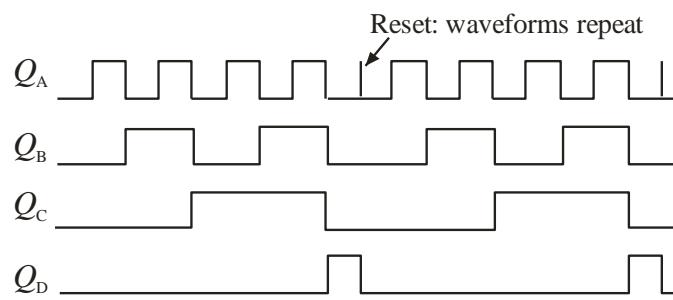
### Evaluation and Review Questions:

1. a. The *A* signal is 4X faster than the clock. Since the measurement shows that it is 4.00 kHz, the clock frequency is 1.00 kHz.  
b. If you want to check the time relationship of any other signal with respect to the displayed signals, it is easier to make the comparison with the slower waveform. Since it is a digital scope, the two channels are digitized together and so this reason is for convenience only. If this were an analog scope, a timing error can be made by triggering on a faster signal to observe (and compare) a slower signal.
2. a. The count sequence is truncated by decoding one number more than the ending number and using the decoded output to reset the counter.  
b. The procedure produces a glitch because the counter is in the decoded state for a very short interval.
3. The clear input may be shorted LOW, producing a HIGH output on both flip-flops; the line from  $\bar{Q}$  to *D* on the *A* flip flop may be open causing an (invalid) HIGH to be clocked into the *A* flip-flop at each clock pulse.
4. The time relationship determined from a two channel oscilloscope could be interpreted incorrectly because the scope can be triggered by *any* arbitrary clock pulse. Relative timing is unambiguous when the various waveforms are compared to the slowest waveform.

5. a. Circuit:



b. Waveforms:



6. The sequence is 0-1-2-3-4-5-8-9-reset (momentarily in state 10)

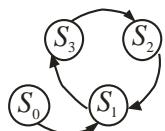
## Experiment 22: Analysis of Synchronous Counters with Decoding

**TABLE 22-2**

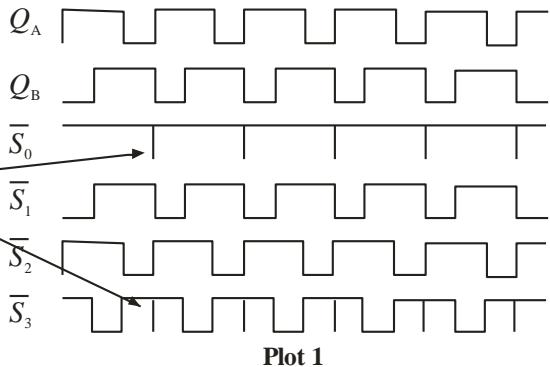
Analysis of synchronous counter  
shown in Figure 22-3

| Outputs | Inputs       |                   |           |             |
|---------|--------------|-------------------|-----------|-------------|
|         | $J_B = Q_A$  | $K_B = \bar{Q}_A$ | $J_A = 1$ | $K_A = Q_B$ |
| 0 0     | 0            | 1                 | 1         | 0           |
| 0 1     | 1            | 0                 | 1         | 0           |
| 1 1     | 1            | 0                 | 1         | 1           |
| 1 0     | 0            | 1                 | 1         | 1           |
| 0 1     | (Repeats...) |                   |           |             |

State diagram:



Note the glitches in the decoded outputs for state 0 and state 3.



Step 4: State diagram:

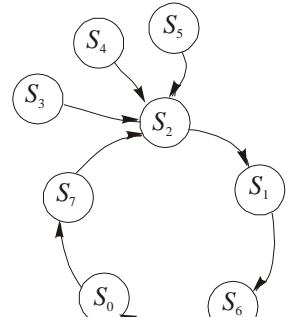


**TABLE 22-3**

Analysis of counter shown in Figure 22-4

Step 6: State diagram:

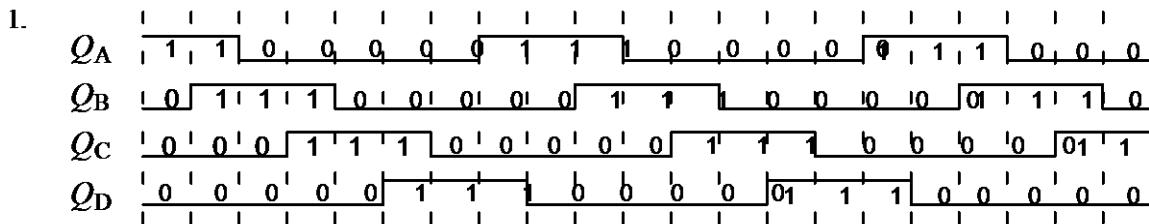
| Outputs | Inputs                                      |           |           |                   |                   |           |
|---------|---------------------------------------------|-----------|-----------|-------------------|-------------------|-----------|
|         | $J_C = \bar{Q}_A$                           | $K_C = 1$ | $J_B = 1$ | $K_B = \bar{Q}_A$ | $J_A = \bar{Q}_C$ | $K_A = 1$ |
| 0 0 0   | 1                                           | 1         | 1         | 1                 | 1                 | 1         |
| 1 1 1   | 0                                           | 1         | 1         | 0                 | 0                 | 1         |
| 0 1 0   | 0                                           | 1         | 1         | 1                 | 1                 | 1         |
| 0 0 1   | 1                                           | 1         | 1         | 0                 | 1                 | 1         |
| 1 1 0   | 0                                           | 1         | 1         | 1                 | 0                 | 1         |
| 0 0 0   | <b>Repeats..test unused states:</b>         |           |           |                   |                   |           |
| 0 1 1   | 0                                           | 1         | 1         | 0                 | 1                 | 1         |
| 0 1 0   | <b>Returns to a tested state (main seq)</b> |           |           |                   |                   |           |
| 1 0 0   | 1                                           | 1         | 1         | 1                 | 0                 | 1         |
| 0 1 0   | <b>Returns to a tested state (main seq)</b> |           |           |                   |                   |           |
| 1 0 1   | 1                                           | 1         | 1         | 0                 | 0                 | 1         |
| 0 1 0   | <b>Returns to a tested state (main seq)</b> |           |           |                   |                   |           |



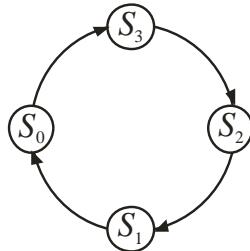
### Further Investigation Results:

The circuit shows the letters for the word C-L-U-E in the seven segment display.

**Evaluation and Review Questions:**



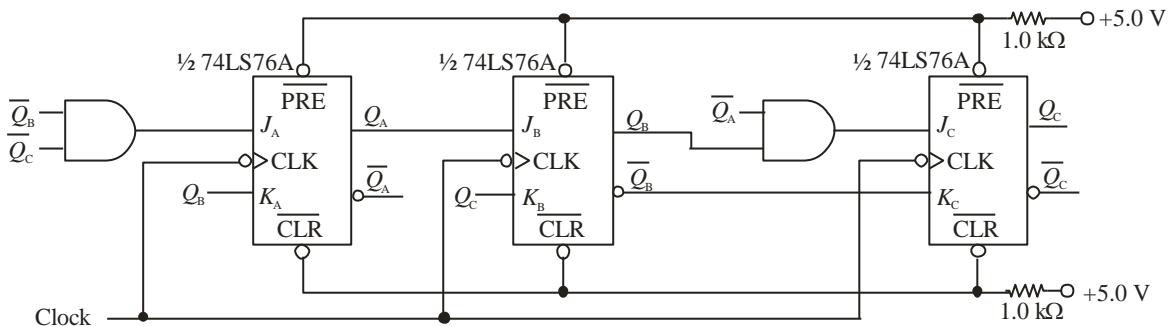
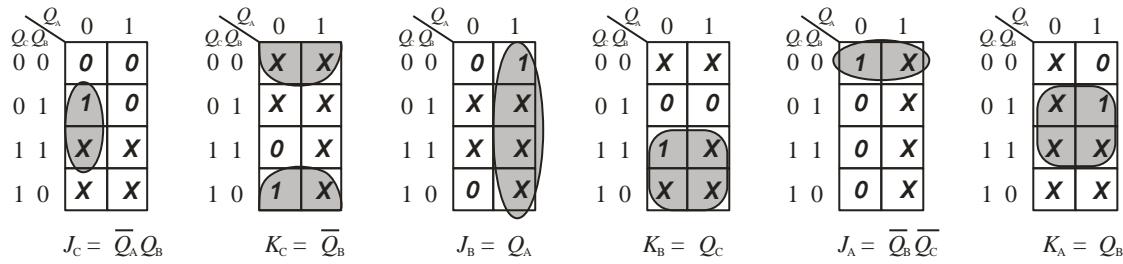
2. The sequence is that of a down counter:



3. Full decoding means all possible states must be decoded. Because there are three outputs, full decoding can be accomplished with a 3-to-8 decoder such as the 74LS138.
4. Connect the pushbutton so that it clears the  $A$  and  $C$  counters and presets the  $B$  counter using the asynchronous clear and preset inputs.
5. The counter can be locked in state 3 if there are no clock pulses or if the common preset line is shorted LOW.
6. State 9 ( $DCBA = 1001$ ) should go to state 1 ( $DCBA = 0001$ ) but isn't changing. By observation, three of the F/F's do not change states for this transition (FFA, FFB, and FFC). FFD should change and isn't changing, so it is the likely culprit.

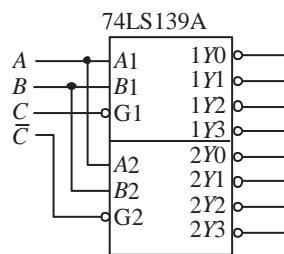
## Experiment 23: Design of Synchronous Counters

| Present State     | Next State        |
|-------------------|-------------------|
| $Q_C$ $Q_B$ $Q_A$ | $Q_C$ $Q_B$ $Q_A$ |
| 0 0 0             | 0 0 1             |
| 0 0 1             | 0 1 1             |
| 0 1 1             | 0 1 0             |
| 0 1 0             | 1 1 0             |
| 1 1 0             | 1 0 0             |
| 1 0 0             | 0 0 0             |



### Further Investigation Results:

The 74LS139A can be connected to form a 3-to-8 decoder as shown below:



### Evaluation and Review Questions:

Note: A variation for drawing Karnaugh maps was presented in the lab book in Figure 23-3 and repeated here for reference. The  $Q_B$  and  $Q_A$  variables are not written side-by-side; rather they are written nearly vertically. Some students find it easier to read the map to relate the number to the variable when they are drawn like this.

1.

| $Q_B$ | $Q_A$ | 0 | 1 | 1 | 1 | 0 |
|-------|-------|---|---|---|---|---|
| 0     | 0     | X | 1 | X | X |   |
| 0     | 1     | 0 | X | X | X |   |
| 1     | 1     | 0 | X | X | X |   |
| 1     | 0     | 0 | X | X | X |   |

$$J_B = Q_A \overline{Q_D}$$

| $Q_B$ | $Q_A$ | 0 | 1 | 1 | 1 | 0 |
|-------|-------|---|---|---|---|---|
| 0     | 0     | X | X | 0 | 0 |   |
| 0     | 1     | X | X | X | T |   |
| 1     | 1     | X | X | X | X |   |
| 1     | 0     | X | X | X | X |   |

$$K_B = Q_C$$

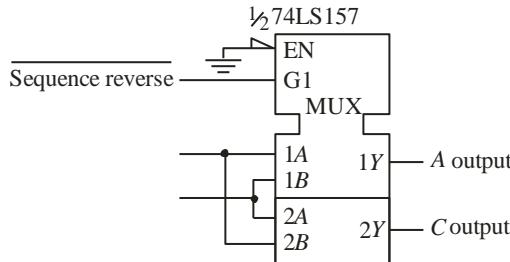
| $Q_B$ | $Q_A$ | 0 | 1 | 1 | 1 | 0 |
|-------|-------|---|---|---|---|---|
| 0     | 0     | X | X | X | 0 |   |
| 0     | 1     | 0 | X | X | 0 |   |
| 1     | 1     | 0 | X | X | X |   |
| 1     | 0     | X | X | X | X |   |

$$J_A = \overline{Q_B} \overline{Q_C}$$

| $Q_B$ | $Q_A$ | 0 | 1 | 1 | 1 | 0 |
|-------|-------|---|---|---|---|---|
| 0     | 0     | X | 0 | 1 | X |   |
| 0     | 1     | X | X | X | X |   |
| 1     | 1     | X | X | X | X |   |
| 1     | 0     | 1 | X | X | X |   |

$$K_A = Q_B$$

2. Since the largest binary number in the circuit for the experiment is less than ten, a BCD to seven-segment decoder (such as the 74LS47A) can be used to convert the binary number into a value that can be shown in a seven-segment display.
3. The reset button is connected between ground and the preset input of FFC and FFB; it is also connected between ground and the clear input of FFA.
4. The count sequence can be reversed by reversing the  $A$  and  $C$  bits from the counter. This can be accomplished by using a 2-input MUX, such as the 74LS157. (Note that only a part of the 74LS157 is shown.)



5. When the counter goes from state 3 to state 2, the  $A$  output is seen to do a 1 to 0 transition. Likewise, when the counter goes from state 6 to state 4, the  $B$  output does a 1 to 0 transition. These transitions are unique, therefore trigger by connecting the  $A$  and  $B$  outputs to the  $A1$  and  $A2$  trigger inputs of the 74121 ( $B$  trigger is HIGH). (See the function table for the 74121, given in Figure 17-1).
6. a. The  $D$  transition table is shown. The  $Q_{N+1}$  output follows  $D$  at the clock edge.

| $Q_N$ | $Q_{N+1}$ | $D$ |
|-------|-----------|-----|
| 0 → 0 | 0         | 0   |
| 0 → 1 | 1         |     |
| 1 → 0 | 0         |     |
| 1 → 1 | 1         |     |

- b. The  $J-K$  flip-flop is more versatile because it has the latch and toggle states. For this reason, there are  $X$ 's on the transition table which aid in designing a simpler circuit for irregular count sequences.

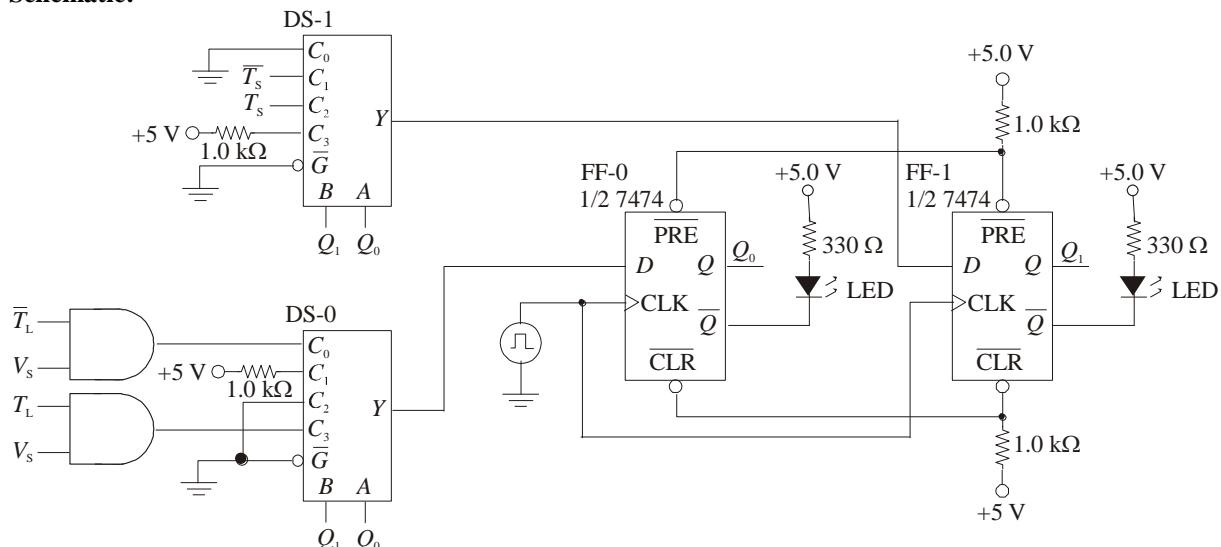
## Experiment 24: The Traffic Signal Controller

Note - Table 24-1 lists present states in the gray-code sequence selected for the controller (0-1-3-2) rather than binary sequence.

**Table 24-1**

| Present State |       | Next State |       | Input Conditions                     | Input Product Term for Data Selector-1 | Input Product Term for Data Selector-0 |
|---------------|-------|------------|-------|--------------------------------------|----------------------------------------|----------------------------------------|
| $Q_1$         | $Q_0$ | $Q_1$      | $Q_0$ |                                      |                                        |                                        |
| 0             | 0     | 0          | 0     | $T_L + \bar{V}_S$<br>$\bar{T}_L V_S$ | 0                                      | $\bar{T}_L V_S$                        |
| 0             | 0     | 0          | 1     |                                      |                                        |                                        |
| 0             | 1     | 0          | 1     | $\bar{T}_S$<br>$\bar{T}_S$           | $\bar{T}_S$                            | 1                                      |
| 0             | 1     | 1          | 1     |                                      |                                        |                                        |
| 1             | 1     | 1          | 1     | $T_L V_S$<br>$\bar{T}_L + \bar{V}_S$ | 1                                      | $T_L V_S$                              |
| 1             | 1     | 1          | 0     |                                      |                                        |                                        |
| 1             | 0     | 1          | 0     | $T_S$<br>$\bar{T}_S$                 | $T_S$                                  | 0                                      |
| 1             | 0     | 0          | 0     |                                      |                                        |                                        |

**Schematic:**



**Figure 24-4**

**Step 6:** To return to the first state, the vehicle sensor must be LOW and remain LOW (no vehicle on side street).

### Further Investigation Results (TTL)

The first idea is sound; the short timer can be eliminated if states change in no less than 4 s. The equations would need to be changed to implement this. The second idea will not allow the state machine to function as designed.

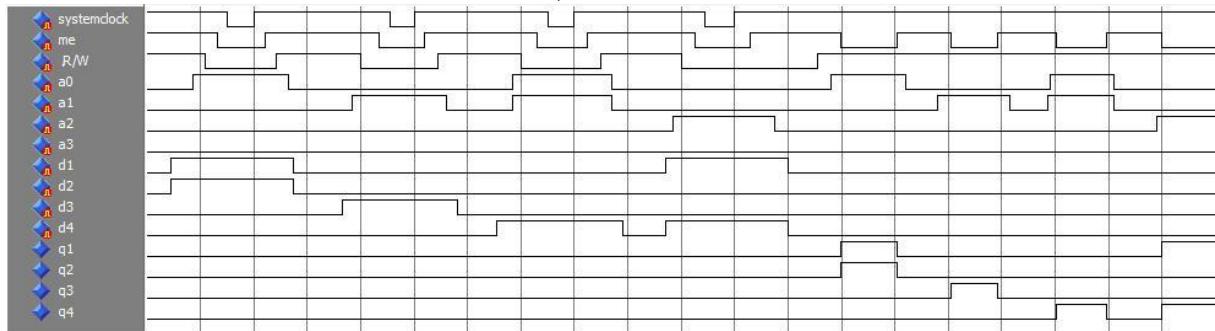
### Evaluation and Review Questions:

- Gray code is selected to avoid false states due to decoding glitches.
- The long timer ( $T_L$ ) and the vehicle sensor ( $V_S$ ) must both be HIGH.
- A third flip-flop is needed for the counter and the MUXs would be changed to 3-to-8 lines.

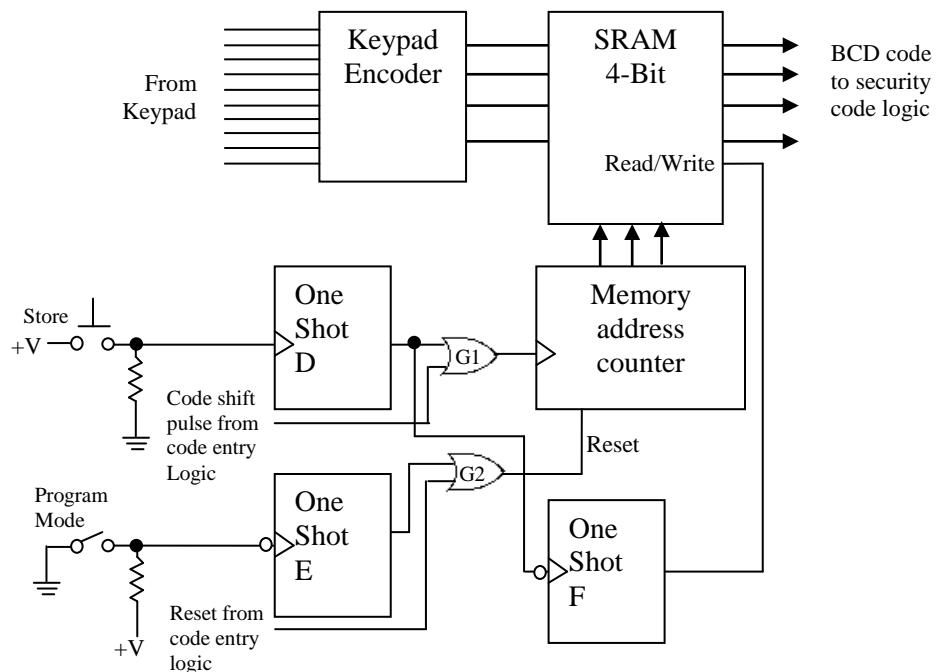
4. Connect the  $Y$  output from the MUXs to  $J$ ; this line is inverted and connected to  $K$ .
5. Refer to the 7th line down on the table in Figure 17-1. In order to have both  $A$  inputs trigger on a trailing edge,  $B$  must be HIGH.
6. The next state will sequence only if the long timer ( $T_L$ ) is LOW and the vehicle sensor ( $V_S$ ) is HIGH. Because the long timer is okay, test  $V_S$ . If it is okay, check the counter for clock pulses and correct inputs on  $D$ . If okay, check asynchronous inputs, power, and grounds.

## Experiment 25: Semiconductor Memories

Student waveforms should look like the results, shown:



**Further Investigation Results:** Design and description may vary.



To initially store a code in the SRAM, the program mode switch is closed, triggering one-shot E, and remains closed during code programming. The pulse output of one-shot E goes through OR gate G2 to reset (clear) the counter. To enter the first code digit, the desired key on the keypad is pressed and held while the store switch is closed momentarily, triggering one-shot D. The pulse output of one-shot D goes through OR gate G1 to clock the counter to the 001 state to select address one in the SRAM. At the end of the pulse from one-shot D, one-shot F is triggered to generate a write pulse to memory, which stores the selected code digit on address one. To enter the second digit, the desired key is pressed and held, again triggering one-shot D and advancing the counter to the 010 state, selecting memory address two. One-shot F is then triggered and generates a write pulse storing the code digit in memory address two. These steps are repeated for the third and fourth digit.

Once the code digits are stored, the program switch is turned off. The memory address counter is reset by the security system code. The security system code sends a shift pulse to increment the memory address counter to the next stored code digit.

**Evaluation and Review Questions:**

1. The basic cell operational component is a flip flop; it can be set or reset for a write operation on input BitIn or tested without changing its state for a read operation. The address select (AddSel) input is set high to select the cell. The bit (Bitin) to be stored sets the JK flip-flop inputs for a set or reset. The Read/Write input is set LOW to enable the input to the JK flip-flop and a clock pulse clocks stores the data. Setting the Read/Write input to a HIGH outputs the stored bit.

2. To decode 8-bits instead of 4-bits, four additional columns of sixteen cells are required. Four additional input and output lines are required to read and write 8-bits.

3. The first printing has an error. The question should read: How can an *open* collector device connect to a common bus? The answer for this question is:

Open collector devices use a single pull-up resistor that is connected to each line. This pull-up resistor keeps the line HIGH unless one of the outputs connected to the line pulls it LOW. Tristate devices do not require a pull-up resistor.

If you are using a PLD trainer, can your PLD device be connected directly to a common bus? (Explain)  
Answers will vary depending on the construction of the trainer board.

4a. The OR gates allow the selected SRAM cell output to be presented on outputs Q1 through Q4 when input R/W is set HIGH.

b. All SRAM cell outputs for each column would be directly connected to each other eliminating isolation from each other. This condition will cause the HIGH output of a select SRAM cell to back feed as an input to all other interconnected cells.

## Experiment 26: Serial-to-Parallel Data Converter

Outputs d0 through d7 follow the serialdatain serial waveform.

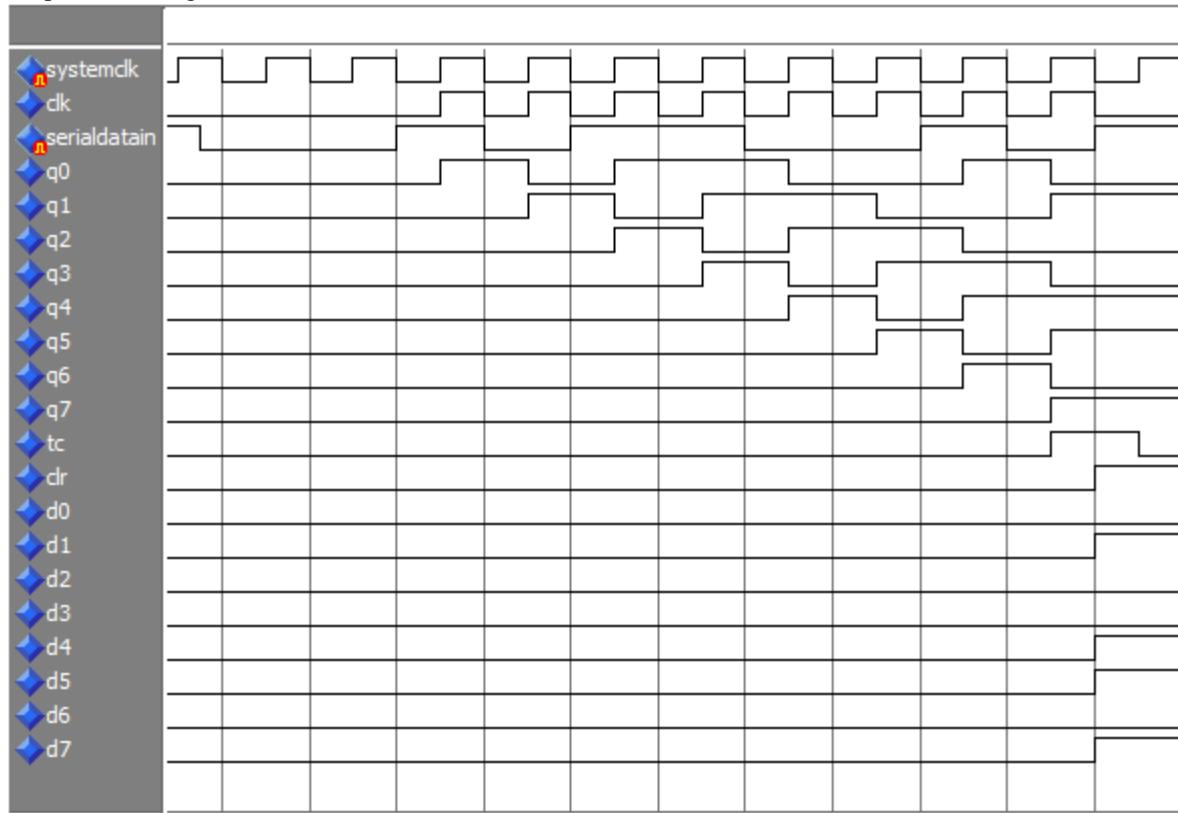
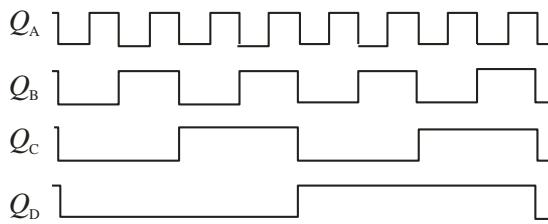


Figure 26-5 Simulation input waveforms.

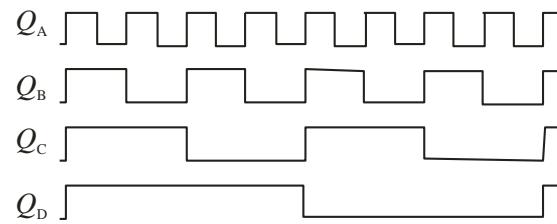
### Evaluation and Review Questions:

1. The EightBitParSftReg and EightBitShiftReg components require eight additional flip-flop stages DFF8 through DFF15 to accommodate the added 8-bits of data. Component CTRDIV10 requires modification to allow a count from 0-16. The output logic TC needs to reset on count 16.
2. The start bit set to LOW indicates that eight data bits will follow. Two stop bits set to HIGH following the data ends the message and resets the count until a LOW input is read as the next start bit and the process repeats.
3. To read data on a falling clock edge the SystemClk input must be inverted and the Serial DataIn input synchronized to the inverted clock.
4. Move CTRDIV10 output Q3 to Q2 to cause the data reset to occur on count 4.
5. The control flip-flop enables the clock generator module sending the serial input data to the data-input register and in turn increments the divide-by-eight counter.
6. The output of the one-shot clears to the control flip-flop and resets the CTR DIV 8 counter. The output of the one-shot also clocks the data from the data-input shift register to the output shift register. Another HIGH-to-LOW transition of the input data stream starts the process over again.

## Experiment 27: D/A and A/D Conversion



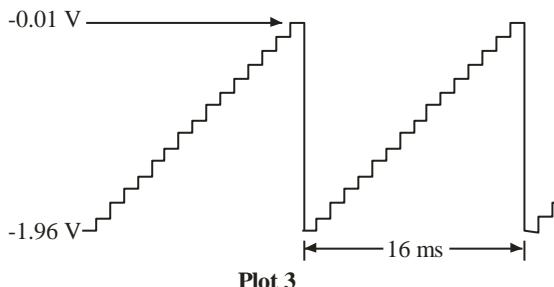
Plot 1



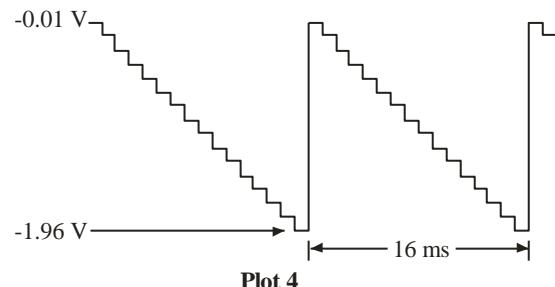
Plot 2

Table 27-1

| Quantity                     | Measured Value |
|------------------------------|----------------|
| DC full-scale output voltage | -1.96 V        |
| Volts/step                   | 0.13 V/step    |

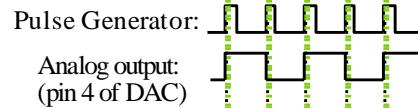


Plot 3



Plot 4

**Steps 7 and 8:** The digital light meter has an oscillating behavior because of the tracking A/D converter as shown. The D/A output waveform increases its dc level from approximately -2 V to approximately -0.2 V as the light level increases.



### Further Investigation Results

Measured data for the circuit in Figure 27-1 is shown on the table. The voltages listed are at the thresholds.

| Voltage | Display | Voltage | Display | Voltage | Display | Voltage | Display |
|---------|---------|---------|---------|---------|---------|---------|---------|
| 0.009 V | 0       | 1.35 V  | 4       | 2.59 V  | 8       | 3.83 V  | █       |
| 0.437 V | 1       | 1.63 V  | 5       | 2.91 V  | 9       | 4.17 V  | ██      |
| 0.675 V | 2       | 1.93 V  | 6       | 3.23 V  | █       | 4.49 V  | ██      |
| 1.009 V | 3       | 2.26 V  | 7       | 3.53 V  | █       | 4.85 V  | blank   |

### Evaluation and Review Questions:

1. a. Binary 1010 is represented by -1.30 V. b. The output frequency is 62.5 Hz.
2.  $(-1.95 \text{ V})/255 \text{ steps} = -7.65 \text{ mV/step}$
3. a. The output voltage is larger because the reference current is larger.  
b. The resolution measured in volts/step increases. (Note that if measured simply by number of bits, the resolution is unchanged.)
4. The oscillation is caused by the quantizing error between the analog voltage from the photocell and the digitized representation from the converter. If the least significant bit is not used, the oscillation can be eliminated.
5. a. Decrease the reference current by increasing the  $5.1 \text{ k}\Omega$  resistor or decrease the  $2.0 \text{ k}\Omega$  load resistor.  
b. No. The change only effects the digital display, not the basic sensitivity of the photocell.
6. A higher clock frequency causes the meter to respond faster to changes but has the disadvantage of causing the display to change rapidly, making it more difficult to read.

## **Experiment 28: Introduction to the Intel Processors**

*Note:* This experiment is an introduction to microprocessor concepts and uses a tutorial format to acquaint students with how computers store data, the Intel register structure, a few assembly language instructions, and an introduction to programming. *Debug* is the program used for the tutorial and is available on any DOS based computer. Debug is limited to the 16-bit registers. No hardware is needed and questions are answered in the Procedure section. Although some steps do not require a student answer, the sequence of the steps is shown here to aid in following the experiment.

### **Procedure:**

**Step 1:** Student enters Debug and observes some Debug commands.

**Step 2:** Answers will vary depending on the computer. The equipment status word 27 C2 given as an example is interpreted as follows:

Number of parallel printer ports attached = 3

Number of serial ports attached = 1

Number of diskette devices = 1

Initial video mode = 80 x 25

Math coprocessor is present? yes

Diskette drive is present? yes

**Step 3:** Answers will vary depending on the computer.

**Step 4:** The right side will be filled with zeros since 30H represents ASCII 0.

**Step 5:** Location DS:20 is changed to 31H (ASCII 1).

**Step 6:** The display will show DS:0020 31 30 DS:0050. (DS will have the current data segment value). This indicates that location DS:20 does not correspond to location DS:50. The two corresponding data points in these locations are 31H and 30H.

**Step 7:** The 16-bit register set is displayed.

**Step 8:** The command sequence is **R BX <cr>** followed by **:200 <cr>** and **R CX <cr>** followed by **:F003 <cr>**.

**Step 9:** The Debug assembler is used to enter a short code.

**Step 10:** The code in step 9 is traced. The *caps lock* is activated (and shown with an LED).

**Step 11:** Each loop toggles the *caps lock* function; the light will turn on and off twice before exiting the loop.

**Step 12:** “Byte-size” data is entered into 17 locations (the 17th location is DS:60 and will have a zero entered as an end of record signal).

**Step 13:** The modified “BIG” code is assembled. After it is executed with the “go” command, AX will have the largest number in the list, BX will contain 0060H (the last location), and CX will contain 0000H (the final value of the loop counter).

**Step 14:** The largest number is observed to have been copied into location DS:60.

### **Further Investigation Results:**

The smallest byte value in the list (from DS:50 to DS:5F will be moved into the AL register and into location DS:60. The AX register will show FFH. BX contains 0060H (as before) and CX contains 0000H (as before).

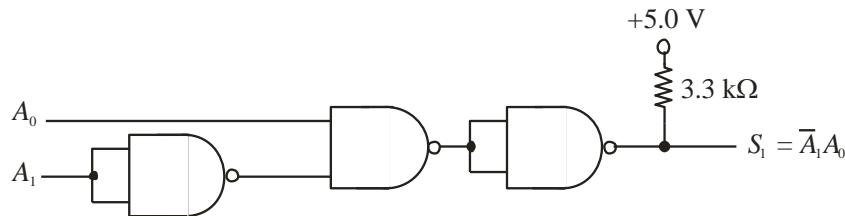
**Evaluation and Review Questions:**

1. The data in step 2 is from the BIOS RAM area, the data from step 3 is in ROM.
2. Locations from DS:1F0 to DS:1FF are filled with the value FAH.
3. **D 0:0 9**
4. The BIOS work area starts at location 40:0. An example of data that is in this area is information about the peripherals connected to the computer.
5. When any number is XORed with itself, the result is always zero because  $0 \text{ XOR } 0 = 0$  and  $1 \text{ XOR } 1 = 0$ .
6. The LOOP instruction decrements the CX register and jumps to the start of the loop if CX is not zero; otherwise it executes the next instruction.

## Experiment 29: Applications of Bus Systems

**Step 3:** With all switches on the 14532B open, the chip outputs are all LOW, including the GS output as shown on the second line of the 14532B truth table. The transistors act as inverters, causing the corresponding inputs on the 14051B to be HIGH. If the inhibit line is tied LOW on the 14051B, the internal FET switch is closed that connects X7 to the common out/in line. This completes the path from the 555 timer to the LED representing the least significant bit, causing it to blink.

Decoder schematic: The decoder schematic depends on the particular address assigned to each student. As an example, a 7400 NAND gate, shown decoding address 01, is drawn below:



### Further Investigation Results:

Answers may vary. A 4-bit counter such as the 7493A may be selected or the student may design a two-bit counter with *J-K* or *D* flip-flops (such as the ripple counter drawn in Figure 18-2). The counter output is connected to a 7447A BCD to seven segment decoder similar to Figure 3-2.

### Evaluation and Review Questions:

1. a. The transistors are used to provide an open-collector connection to the data bus.  
b. The display is an input device. The bus can have multiple input devices connected to it without bus contention problems.
2. Each priority encoder is located at a unique address as determined by the individual decoders.
3. The 14051B can be connected as a MUX by connecting inputs to the *X* lines and taking the output from the common *out/in* line. The select inputs behave the same for both configurations.
4.  $R_{16}$  is a current limiting resistor for the LEDs. If it opens, no LED will be on.
5. a. The constant HIGH is connected to the selected output, causing the selected LED to stay on.  
b. All LEDs will be off.
6. a. The line will be LOW.  
b. A data line can be pulled LOW by any transistor connected to the line that is conducting because it provides a direct path to ground.  
c. Check the address bus to determine which indicator is being addressed.