Online Instructor's Manual

to accompany

Digital Fundamentals

Tenth Edition

Thomas L. Floyd

PEARSON
Prentice Hall

Upper Saddle River, New Jersey
Columbus, Ohio

10 9 8 7 6 5 4 3 2 1

ISBN-13: 978-0-13-712960-7
ISBN-10: 0-13-712960-2

CONTENTS

To access supplementary materials online, instructors need to request an instructor access code. Go to **www.pearsonhighered.com/irc**, where you can register for an access code. Within 48 hours after registering you will receive a confirming e-mail including an instructor access code. Once you have received your code, go the site and log on for full instructions on downloading the materials you wish to use.

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

PART 1

Problem Solutions

CHAPTER 1
INTRODUCTORY CONCEPTS

Section 1-1 Digital and Analog Quantities

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

2. Pressure is an analog quantity.

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.

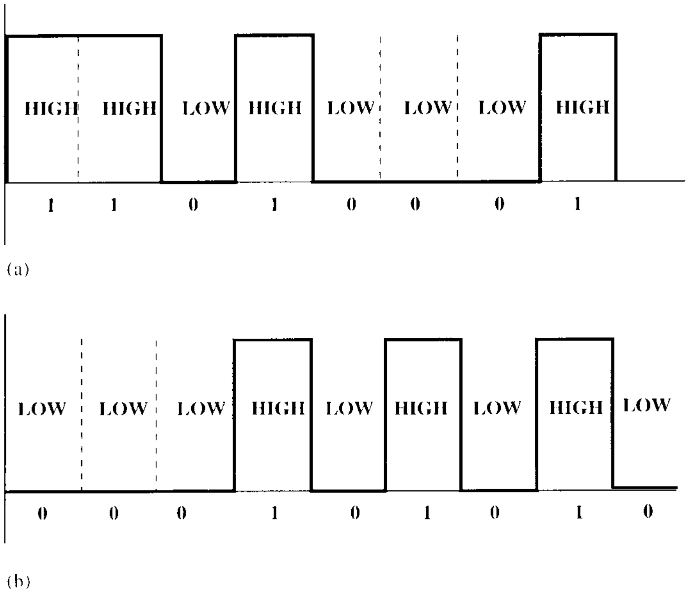


FIGURE 1-1

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. $T = 4$ 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 1-61 is **periodic** because it repeats at a fixed interval.

11. $t_W = 2$ ms; $T = 4$ 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.

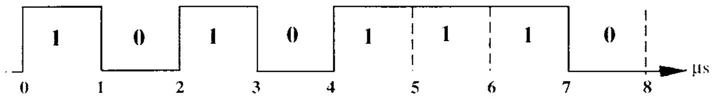


FIGURE 1-4

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

Parallel transfer time = 1 bit time = 1 μs

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

Section 1-3 Basic Logic Operations

15. L_{ON} = SW1 + SW2 + SW1 · SW2

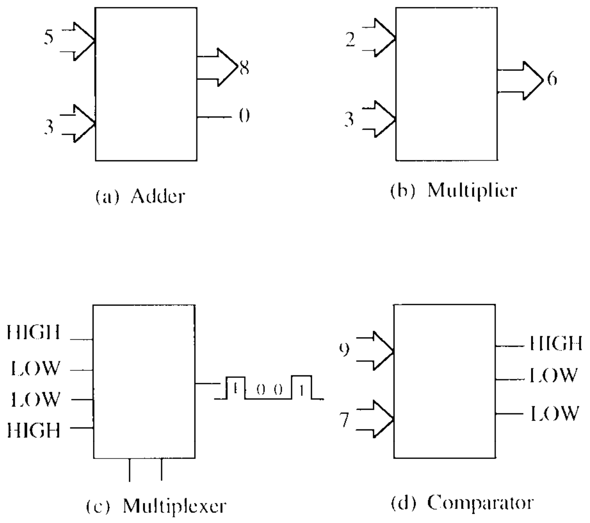
16. An AND gate produces a HIGH output only when *all* of its inputs are HIGH.

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 Introduction to the System Concept

19. See Figure 1-6.



(a) Adder

(b) Multiplier

(c) Multiplexer

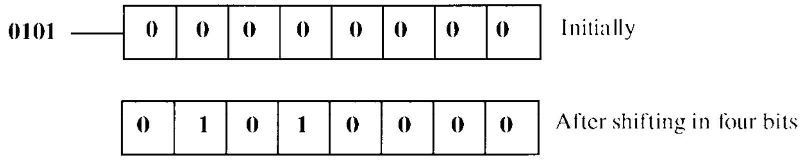
(d) Comparator

FIGURE 1-6

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

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

21. See Figure 1-7.



0101	0	0	0	0	0	0	0	0	Initially
	0	1	0	1	0	0	0	0	After shifting in four bits

FIGURE 1-7

Section 1-5 Fixed-Function Integrated Circuits

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

23. 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.

24. See Figure 1-8.

Section 1-6 Test and Measurement Instruments

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

26. A flashing probe lamp indicates a continuous sequence of pulses (pulse train).

Section 1-7 Introduction to Programmable Logic

27. The following do not describe PLDs: VHDL, AHDL

28. SPLD: Simple Programmable Logic Device
CPLD: Complex Programmable Logic Device
HDL: Hardware Description Language
FPGA: Field-Programmable Gate Array
GAL: Generic Array Logic

29. (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.

30. 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.

CHAPTER 2
NUMBER SYSTEMS, OPERATIONS, AND CODES

Section 2-1 Decimal Numbers

1. (a) $1386 = 1 \times 10^3 + 3 \times 10^2 + 8 \times 10^1 + 6 \times 10^0$
$= 1 \times 1000 + 3 \times 100 + 8 \times 10 + 6 \times 1$
The digit 6 has a weight of $10^0 = 1$

(b) $54{,}692 = 5 \times 10^4 + 4 \times 10^3 + 6 \times 10^2 + 9 \times 10^1 + 2 \times 10^0$
$= 5 \times 10{,}000 + 4 \times 1000 + 6 \times 100 + 9 \times 10 + 2 \times 1$
The digit 6 has a weight of $10^2 = 100$

(c) $671{,}920 = 6 \times 10^5 + 7 \times 10^4 + 1 \times 10^3 + 9 \times 10^2 + 2 \times 10^1 + 0 \times 10^0$
$= 6 \times 100{,}000 + 7 \times 10{,}000 + 1 \times 1000 + 9 \times 100 + 2 \times 10 + 0 \times 1$
The digit 6 has a weight of $10^5 = 100{,}000$

2. (a) $10 = 10^1$ (b) $100 = 10^2$
(c) $10{,}000 = 10^4$ (d) $1{,}000{,}000 = 10^6$

3. (a) $471 = 4 \times 10^2 + 7 \times 10^1 + 1 \times 10^0$
$= 4 \times 100 + 7 \times 10 + 1 \times 1$
$= 400 + 70 + 1$

(b) $9{,}356 = 9 \times 10^3 + 3 \times 10^2 + 5 \times 10^1 + 6 \times 10^0$
$= 9 \times 1000 + 3 \times 100 + 5 \times 10 + 6 \times 1$
$= 9{,}000 + 300 + 50 + 6$

(c) $125{,}000 = 1 \times 10^5 + 2 \times 10^4 + 5 \times 10^3$
$= 1 \times 100{,}000 + 2 \times 10{,}000 + 5 \times 1000$
$= 100{,}000 + 20{,}000 + 5{,}000$

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

Section 2-2 Binary Numbers

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

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)

$$\begin{aligned}
\frac{15}{2} &= 7,\ R = 1\text{(LSB)}\\
\frac{7}{2} &= 3,\ R = 1\\
\frac{3}{2} &= 1,\ R = 1\\
\frac{1}{2} &= 0,\ R = 1\text{ (MSB)}
\end{aligned}$$

(b)

$$\begin{aligned}
\frac{21}{2} &= 10,\ R = 1\text{ (LSB)}\\
\frac{10}{2} &= 5,\quad R = 0\\
\frac{5}{2} &= 2,\quad R = 1\\
\frac{2}{2} &= 1,\quad R = 0\\
\frac{1}{2} &= 0,\quad R = 1\text{ (MSB)}
\end{aligned}$$

(c)

$$\begin{aligned}
\frac{28}{2} &= 14,\quad R = 0\text{ (LSB)}\\
\frac{14}{2} &= 7,\quad R = 0\\
\frac{7}{2} &= 3,\quad R = 1\\
\frac{3}{2} &= 1,\quad R = 1\\
\frac{1}{2} &= 0,\quad R = 1\text{ (MSB)}
\end{aligned}$$

(d)

$$\begin{aligned}
\frac{34}{2} &= 17,\ R = 0\text{ (LSB)}\\
\frac{17}{2} &= 8,\ R = 1\\
\frac{8}{2} &= 4,\quad R = 0\\
\frac{4}{2} &= 2,\quad R = 0\\
\frac{2}{2} &= 1,\quad R = 0\\
\frac{1}{2} &= 0,\quad R = 1\text{ (MSB)}
\end{aligned}$$

(e)

$$\begin{aligned}
\frac{40}{2} &= 20,\ R = 0\text{ (LSB)}\\
\frac{20}{2} &= 10,\ R = 0\\
\frac{10}{2} &= 5,\quad R = 0\\
\frac{5}{2} &= 2,\quad R = 1\\
\frac{2}{2} &= 1,\quad R = 0\\
\frac{1}{2} &= 0,\quad R = 1\text{ (MSB)}
\end{aligned}$$

(f)

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

(g)

$$\begin{aligned}
\frac{65}{2} &= 32,\ R = 1\text{ (LSB)}\\
\frac{32}{2} &= 16,\ R = 0\\
\frac{16}{2} &= 8,\ R = 0\\
\frac{8}{2} &= 4,\quad R = 0\\
\frac{4}{2} &= 2,\quad R = 0\\
\frac{2}{2} &= 1,\quad R = 0\\
\frac{1}{2} &= 0,\quad R = 1\text{(MSB)}
\end{aligned}$$

(h)

$$\begin{aligned}
\frac{73}{2} &= 36,\ R = 1\text{ (LSB)}\\
\frac{36}{2} &= 18,\ R = 0\\
\frac{18}{2} &= 9,\quad R = 0\\
\frac{9}{2} &= 4,\quad R = 1\\
\frac{4}{2} &= 2,\quad R = 0\\
\frac{2}{2} &= 1,\quad R = 0\\
\frac{1}{2} &= 0,\quad R = 1\text{ (MSB)}
\end{aligned}$$

14. (a)

0.98 × 2 = 1.96	1 (MSB)
0.96 × 2 = 1.92	1
0.92 × 2 = 1.84	1
0.84 × 2 = 1.68	1
0.68 × 2 = 1.36	1
0.36 × 2 = 0.72	0

continue if more accuracy is desired
0.111110

(b)

0.347 × 2 = 0.694	0 (MSB)
0.694 × 2 = 1.388	1
0.388 × 2 = 0.776	0
0.776 × 2 = 1.552	1
0.552 × 2 = 1.104	1
0.104 × 2 = 0.208	0
0.208 × 2 = 0.416	0

continue if more accuracy is desired
0.0101100

(c)

0.9028 × 2 = 1.8056	1 (MSB)
0.8056 × 2 = 1.6112	1
0.6112 × 2 = 1.2224	1
0.2224 × 2 = 0.4448	0
0.4448 × 2 = 0.8896	0
0.8896 × 2 = 1.7792	1
0.7792 × 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) $\begin{array}{r} 11010 \\ -\,10111 \\ \hline 00011 \end{array}$

17. (a)
$$\begin{array}{r} 11 \\ \times 11 \\ \hline 11 \\ 11 \\ \hline 1001 \end{array}$$

(b)
$$\begin{array}{r} 100 \\ \times 10 \\ \hline 000 \\ 100 \\ \hline 1000 \end{array}$$

(c)
$$\begin{array}{r} 111 \\ \times 101 \\ \hline 111 \\ 000 \\ 111 \\ \hline 100011 \end{array}$$

(d)
$$\begin{array}{r} 1001 \\ \times 110 \\ \hline 0000 \\ 1001 \\ 1001 \\ \hline 110110 \end{array}$$

(e)
$$\begin{array}{r} 1101 \\ \times 1101 \\ \hline 1101 \\ 0000 \\ 1101 \\ 1101 \\ \hline 10101001 \end{array}$$

(f)
$$\begin{array}{r} 1110 \\ \times 1101 \\ \hline 1110 \\ 0000 \\ 1110 \\ 1110 \\ \hline 10110110 \end{array}$$

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

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

19. 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
–34 = 11011101

(b) Magnitude of 57 = 0111001
+57 = 00111001

(c) Magnitude of 99 = 1100011
–99 = 10011100

(d) Magnitude of 115 = 1110011
+115 = 01110011

25. (a) Magnitude of 12 = 1100
+12 = 00001100

(b) Magnitude of 68 = 1000100
–68 = 10111100

(c) Magnitude of 101_{10} = 1100101
+101_{10} = 01100101

(d) Magnitude of 125 = 1111101
–125 = 10000011

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 → sign = 0
1.11110000101011 × 2^{14} → exponent = 127 + 14 + 141 = 10001101
Mantissa = 11110000101011000000000
01000110111110000101011000000000

(b) 100110000011000 → sign = 1
1.10000011000 × 2^{11} → exponent = 127 + 11 = 138 = 10001010
Mantissa = 10000011000000000000000
11000101010000011000000000000000

30. (a) 11000000101001001110001000000000
Sign = 1
Exponent = 10000001 = 129 – 127 = 2
Mantissa = 1.01001001110001 × 2^2 = 101.001001110001
–101.001001110001 = **–5.15258789**

(b) 01100110010000111110100100000000
Sign = 0
Exponent = 11001100 = 204 – 127 = 77
Mantissa = 1.100001111101001
1.100001111101001 × 2^{77}

Section 2-7 Arithmetic Operations with Signed Numbers

31. (a) 33 = 00100001
15 = 00001111

$$\begin{array}{r} 00100001 \\ +\,00001111 \\ \hline 00110000 \end{array}$$

(b) 56 = 00111000
27 = 00011011
–27 = 11100101

$$\begin{array}{r} 00111000 \\ +\,11100101 \\ \hline 00011101 \end{array}$$

(c) 46 = 00101110
–46 = 11010010
25 = 00011001

$$\begin{array}{r} 11010010 \\ +\,00011001 \\ \hline 11101011 \end{array}$$

(d) 110_{10} = 01101110
-110_{10} = 10010010
84 = 01010100
–84 = 10101100

$$\begin{array}{r} 10010010 \\ +\,10101100 \\ \hline 100111110 \end{array}$$

32. (a)

$$\begin{array}{r} 00010110 \\ +\,00110011 \\ \hline 01001001 \end{array}$$

(b)

$$\begin{array}{r} 01110000 \\ +\,10101111 \\ \hline 100011111 \end{array}$$

33. (a)

$$\begin{array}{r} 10001100 \\ +\,00111001 \\ \hline 11000101 \end{array}$$

(b)

$$\begin{array}{r} 11011001 \\ +\,11100111 \\ \hline 11000000 \end{array}$$

34. (a)

$$\begin{array}{r} 00110011 \\ -\,00010000 \\ \hline \end{array} \qquad \begin{array}{r} 00110011 \\ +\,11110000 \\ \hline \cancel{1}\;00100011 \end{array}$$

(b)

$$\begin{array}{r} 01100101 \\ -\,11101000 \\ \hline \end{array} \qquad \begin{array}{r} 01100101 \\ +\,00011000 \\ \hline 01111101 \end{array}$$

35.

$$\begin{array}{r} 01101010 \\ \times\,11110001 \\ \hline \end{array} \qquad \begin{array}{r} 01101010 \\ \times\,00001111 \\ \hline 01101010 \\ 01101010 \\ \hline 100111110 \\ 01101010 \\ \hline 1011100110 \\ 01101010 \\ \hline 11000110110 \end{array}$$

Changing to 2’s complement with sign: 100111001010

36. $\dfrac{01000100}{00011001} = 00000010$

$\dfrac{68}{25}$ = 2, 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) $FF_{16} + BB_{16} = 1BA_{16}$

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 ibts BCD
	(l)	156 = 10011100_2	8 bits binary, 12 bits BCD

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)
$$\begin{array}{r} 10000101 \\ +\ 00010011 \\ \hline 10000111 \end{array}$$

53. (a)

$$\begin{array}{rl} 1000 & \\ +\,0110 & \\ \hline 1110 & \textit{invalid} \\ +\,0110 & \\ \hline 00010100 & \end{array}$$

(b)

$$\begin{array}{rl} 0111 & \\ +\,0101 & \\ \hline 1100 & \textit{invalid} \\ +\,0110 & \\ \hline 00010010 & \end{array}$$

(c)

$$\begin{array}{rl} 1001 & \\ +\,1000 & \\ \hline 10001 & \textit{invalid} \\ +\,0110 & \\ \hline 00010111 & \end{array}$$

(d)

$$\begin{array}{rl} 1001 & \\ +\,0111 & \\ \hline 10000 & \textit{invalid} \\ +\,0110 & \\ \hline 00010110 & \end{array}$$

(e)

$$\begin{array}{rl} 00100101 & \\ +\,00100111 & \\ \hline 01001100 & \textit{invalid} \\ +\,0110 & \\ \hline 01010010 & \end{array}$$

(f)

$$\begin{array}{rl} 01010001 & \\ +\,01011000 & \\ \hline 10101001 & \textit{invalid} \\ +\,0110 & \\ \hline 000100001001 & \end{array}$$

(g)

$$\begin{array}{rl} 10011000 & \\ +\,10010111 & \\ \hline 100101111 & \textit{invalid} \\ +\,01100110 & \\ \hline 000110010101 & \end{array}$$

(h)

$$\begin{array}{rl} 010101100001 & \\ +\,011100001000 & \\ \hline 110001101001 & \textit{invalid} \\ +\,0110 & \\ \hline 0001001001101001 & \end{array}$$

54. (a) 4 + 3

$$\begin{array}{r} 0100 \\ +\,0011 \\ \hline 0111 \end{array}$$

(b) 5 + 2

$$\begin{array}{r} 0101 \\ +\,0010 \\ \hline 0111 \end{array}$$

(c) 6 + 4

$$\begin{array}{r} 0110 \\ +\,0100 \\ \hline 1010 \\ +\,0110 \\ \hline 00010000 \end{array}$$

(d) 17 + 12

$$\begin{array}{r} 00010111 \\ +\,00100010 \\ \hline 00101001 \end{array}$$

(e) 28 + 23

$$\begin{array}{r} 00101000 \\ +\,00100011 \\ \hline 01001011 \\ +\,0110 \\ \hline 01010001 \end{array}$$

(f) 65 + 58

$$\begin{array}{r} 01100101 \\ +\,01011000 \\ \hline 10111101 \\ +\,01100110 \\ \hline 000100100011 \end{array}$$

(g) 113 + 101

$$\begin{array}{r} 000100010011 \\ +\,000100000001 \\ \hline 001000010100 \end{array}$$

(h) 295 + 157

$$\begin{array}{r} 001010010101 \\ +\,000101010111 \\ \hline 001111101100 \\ +\,01100110 \\ \hline 010001010010 \end{array}$$

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)
1 + 1 + 0 + 1 + 1 *Binary*
1 0 1 1 0 *Gray*

(b)
1 + 0 + 0 + 1 + 0 + 1 + 0 *Binary*
1 1 0 1 1 1 1 *Gray*

(c)
1 + 1 + 1 + 1 + 0 + 1 + 1 + 1 + 0 + 1 + 1 + 1 + 0 *Binary*
1 0 0 0 1 1 0 0 1 1 0 0 1 *Gray*

57. (a)
1 0 1 0 *Gray*
1 1 0 0 *Binary*

(b)
0 0 0 1 0 *Gray*
0 0 0 1 1 *Binary*

(c)
1 1 0 0 0 0 1 0 0 0 1 *Gray*
1 0 0 0 0 0 1 1 1 1 0 *Binary*

58. (a) 1 → 00110001
(b) 3 → 00110011
(c) 6 → 00110110
(d) 10 → 0011000100110000
(e) 18 → 0011000100111000
(f) 29 → 0011001000111001
(g) 56 → 0011010100110110
(h) 75 → 0011011100110101
(i) 107 → 001100010011000000110111

59. (a) 0011000 → CAN (b) 1001010 → **J**
(c) 0111101 → = (d) 0100011 → #
(e) 0111110 → > (f) 1000010 → **B**

60.

1001000	1100101	1101100	1101100	1101111	0101110	0100000
H	**e**	**l**	**l**	**o**	**.**	**#**
1001000	1101111	1110111	0100000	1100001	1110010	1100101
H	**o**	**w**	**#**	**a**	**r**	**e**
0100000	1111001	1101111	1110101	0111111		
#	**y**	**o**	**u**	**?**		

61.

1001000	1100101	1101100	1101100	1101111	0101110	0100000
48	**65**	**6C**	**6C**	**6F**	**2E**	**20**
1001000	1101111	1110111	0100000	1100001	1110010	1100101
48	**6F**	**77**	**20**	**61**	**72**	**65**
0100000	1111001	1101111	1110101	0111111		
20	**79**	**6F**	**75**	**3F**		

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 Codes

63. Code (b) 011101010 has five 1s, so it is in error.

64. Codes (a) 11110110 and (c) 01010101010101010 are in error because they have an even number of 1s.

65. (a) 1 10100100 (b) 0 00001001 (c) 1 11111110

66. (a)

```
  1100
+1011
  0111
```

(b)

```
  1111
+0100
  1011
```

(c)

```
  100011100
+  10011001
  110000101
```

67. (a)

```
  1100
+0111
  1011
```

(b)

```
  1111
+1011
  0100
```

(c)

```
  100011100
+110000101
  010011001
```

In each case, you get the other number.

68.

```
101100100000
1010
   1001
   1010
     1100
     1010
      1100
      1010
       1100
       1010
        1100
        1010
Remainder = 0110
```

```
101100100110
1010
   1001
   1010
     1100
     1010
      1101
      1010
       1111
       1010
        1010
        1010
        0000
```

Append remainder to data. CRC is 101100100110.

69. Error in MSB of transmitted CRC:

```
001100100110
1010
1001
1010
  1100
  1010
   1101
   1010
    1110
    1010
     1000
     1010
       1011
       1010
          10
```

Remainder is 10, indicating an error.

CHAPTER 3
LOGIC GATES

Section 3-1 The Inverter

1. See Figure 3-1.

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

3. See Figure 3-2.

FIGURE 3-2

Section 3-2 The AND Gate

4. See Figure 3-3.

FIGURE 3-3

5. See Figure 3-4.

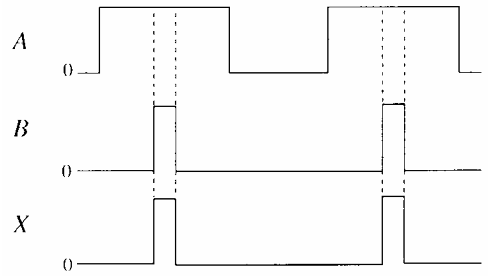


FIGURE 3-4

6. See Figure 3-5.

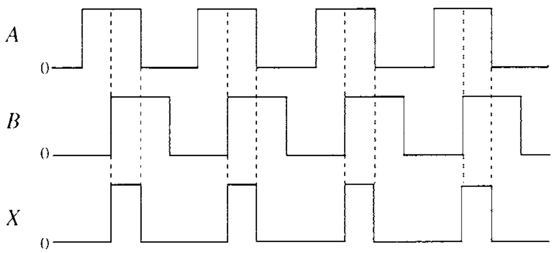


FIGURE 3-5

7. See Figure 3-6.

FIGURE 3-6

8. See Figure 3-7.

FIGURE 3-7

Section 3-3 The OR Gate

9. See Figure 3-8.

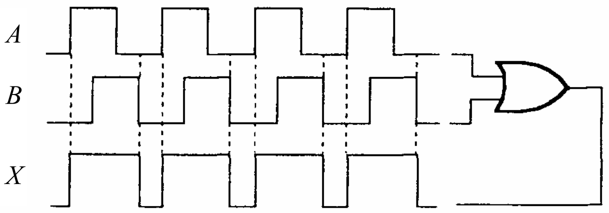


FIGURE 3-8

10. See Figure 3-9.

FIGURE 3-9

11. See Figure 3-10.

FIGURE 3-10

12. See Figure 3-11.

FIGURE 3-11

13. See Figure 3-12.

FIGURE 3-12

14.

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

Section 3-4 The NAND Gate

15. See Figure 3-13.

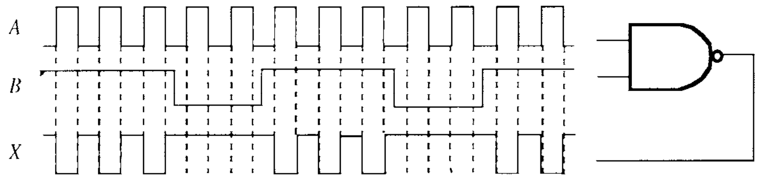


FIGURE 3-13

16. See Figure 3-14.

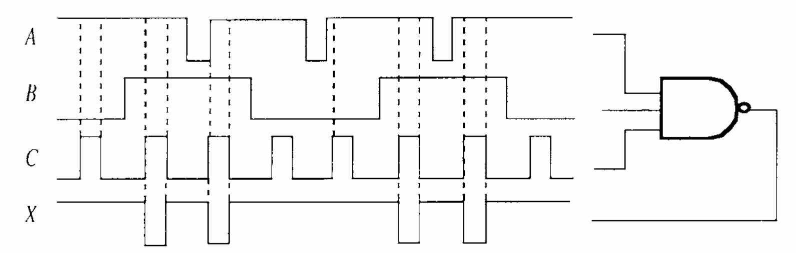


FIGURE 3-14

17. See Figure 3-15.

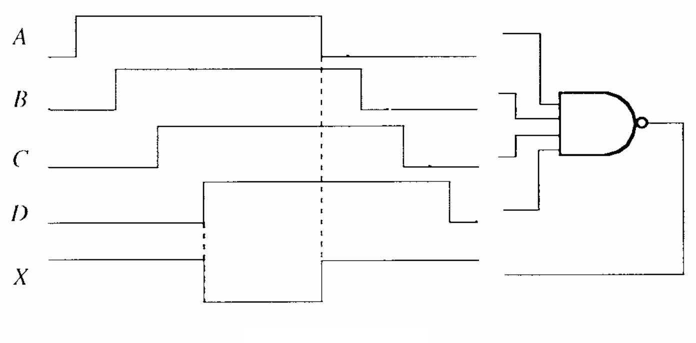
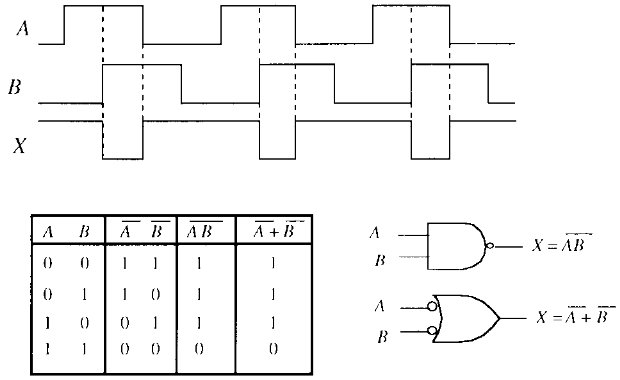


FIGURE 3-15

18. See Figure 3-16.



A	B	$\overline{A}$	$\overline{B}$	$\overline{AB}$	$\overline{A}+\overline{B}$
0	0	1	1	1	1
0	1	1	0	1	1
1	0	0	1	1	1
1	1	0	0	0	0

FIGURE 3-16

Section 3-5 The NOR Gate

19. See Figure 3-17.

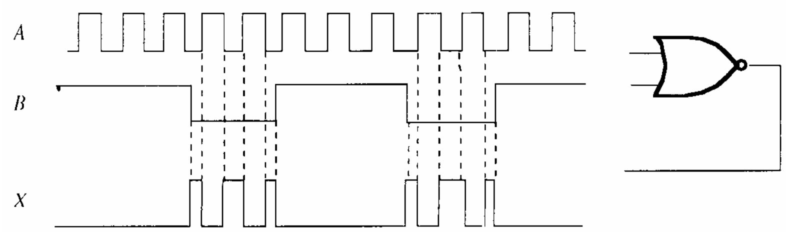


FIGURE 3-17

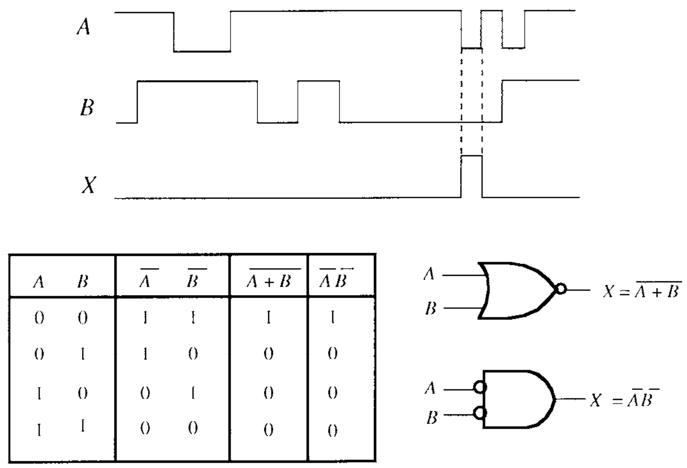
20. See Figure 3-18.

FIGURE 3-18

21. See Figure 3-19.

FIGURE 3-19

22. See Figure 3-20.



A	B	$\overline{A}$	$\overline{B}$	$\overline{A+B}$	$\overline{A}\,\overline{B}$
0	0	1	1	1	1
0	1	1	0	0	0
1	0	0	1	0	0
1	1	0	0	0	0

FIGURE 3-20

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

23. 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.

XOR = $A\overline{B}+\overline{A}B$
OR = $A+B$

24. See Figure 3-21.

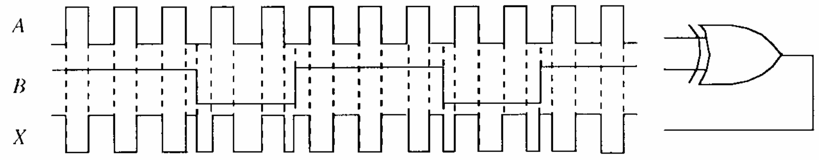


FIGURE 3-21

25. See Figure 3-22.

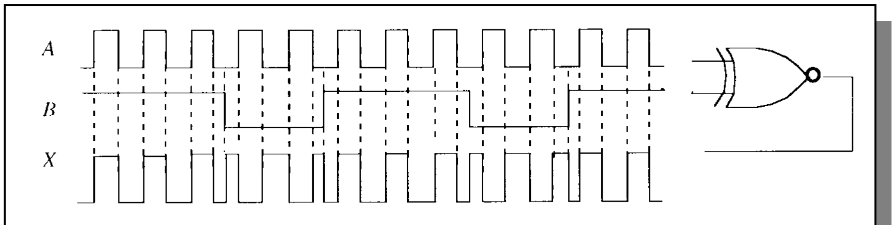


FIGURE 3-22

26. See Figure 3-23.

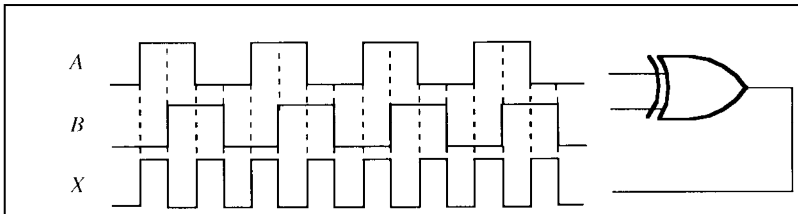


FIGURE 3-23

Section 3-7 Fixed-Function Logic

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

28. (a) $P = \left(\frac{I_{\text{CCH}} + I_{\text{CCL}}}{2}\right)V_{\text{CC}} = \left(\frac{1.6\text{ mA} + 4.4\text{ mA}}{2}\right)5.5\text{ V} = 16.5\text{ mW}$

(b) $V_{\text{OH(min)}} = 2.7$ V

(c) $t_{PLH} = T_{PHL} = 15$ ns

(d) $V_{\text{OL}} = 0.4$ V (max)

(e) @ $V_{\text{CC}} = 2$ V, $t_{PHL} = t_{PLH} = 75$ ns; @ $V_{\text{CC}} = 6$ V, $t_{PHL} = t_{PLH} = 13$ ns

29. See Figure 3-24.

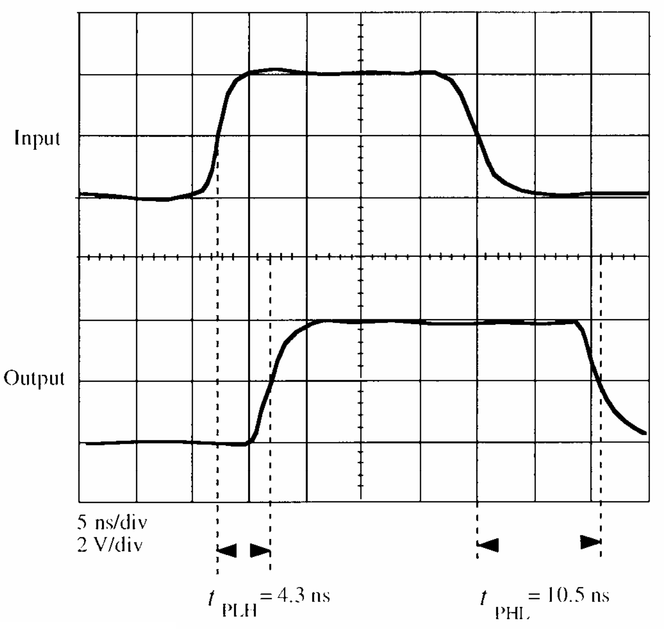


FIGURE 3-24

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

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

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

Section 3-8 Troubleshooting

33. (a) NAND gate OK
(b) AND gate faulty
(c) NAND gate faulty
(d) NOR gate OK
(e) XOR gate faulty
(f) XOR gate OK

34. (a) NAND gate faulty. Input A open.
(b) NOR gate faulty. Input B shorted to ground.
(c) NAND gate OK
(d) XOR gate faulty. Input A open.

35. (a) The gate does not respond to pulses on either input when the other input is HIGH. It is unlikely that both inputs are open. The most probable fault is that the output is stuck in the LOW state (shorted to ground, perhaps) although it could be open.

(b) Pin 4 input or pin 6 output internally open.

36. The timer input to the AND gate is open. Check for 30-second HIGH level on this input when ignition is turned on.

37. An open seat-belt input to the AND gate will act like a constant HIGH just as if the seat belt were unbuckled.

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

Section 3-9 Programmable Logic

39. $X_1 = \overline{A}B$
$X_2 = \overline{A}\,\overline{B}$
$X_3 = A\overline{B}$

40. $X_1 = \overline{A}BC$
Row 1: blow A, B, $\overline{B}$, C, and $\overline{C}$ column fuses
Row 2: blow A, $\overline{A}$, $\overline{B}$, C, and $\overline{C}$ column fuses
Row 3: blow A, $\overline{A}$, B, $\overline{B}$, and $\overline{C}$ column fuses
$X_2 = AB\overline{C}$
Row 4: blow $\overline{A}$, B, $\overline{B}$, C, and $\overline{C}$ column fuses
Row 5: blow A, $\overline{A}$, $\overline{B}$, C, and $\overline{C}$ column fuses
Row 6: blow A, $\overline{A}$, B, $\overline{B}$, and C column fuses
$X_3 = \overline{A}B\overline{C}$
Row 7: blow A, B, $\overline{B}$, C, and $\overline{C}$ column fuses
Row 8: blow A, $\overline{A}$, $\overline{B}$, C, and $\overline{C}$ column fuses
Row 9: blow A, $\overline{A}$, B, $\overline{B}$, and C column fuses

Special Design Problems

41. See Figure 3-25.

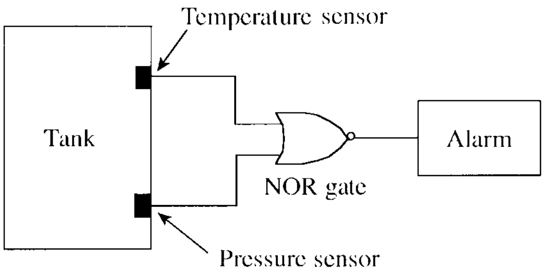


FIGURE 3-25

42. See Figure 3-26.

FIGURE 3-26

43. Add an inverter to the Enable input line of the AND gate as shown in Figure 3-27.

FIGURE 3-27

44. See Figure 3-28.

FIGURE 3-28

45. See Figure 3-29.

FIGURE 3-29

46. See Figure 3-30.

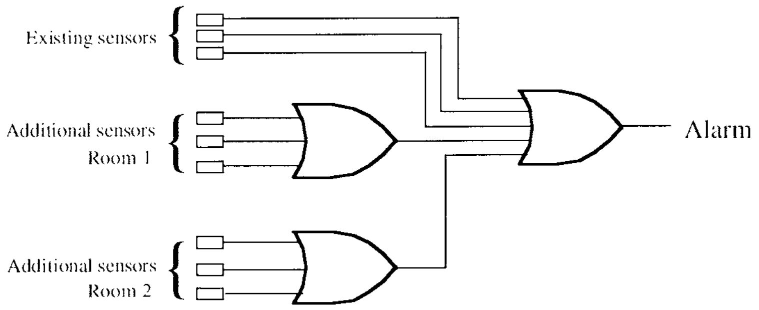


FIGURE 3-30

47. See Figure 3-31.

FIGURE 3-31

Multisim Troubleshooting Practice

48. Input A shorted to output.

49. Inputs shorted together.

50. No fault.

51. Output open.

CHAPTER 4
BOOLEAN ALGEBRA AND LOGIC SIMPLIFICATION

Section 4-1 Boolean Operations and Expressions

1. $X = A + B + C + D$
This is an OR configuration.

2. $Y = ABCDE$

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

4.

(a)	$0 + 0 + 1 = 1$	(b)	$1 + 1 + 1 = 1$
(c)	$1 \cdot 0 \cdot 0 = 1$	(d)	$1 \cdot 1 \cdot 1 = 1$
(e)	$1 \cdot 0 \cdot 1 = 0$	(f)	$1 \cdot 1 + 0 \cdot 1 \cdot 1 = 1 + 0 = 1$

5.
(a) $AB = 1$ when $A = 1$, $B = 1$
(b) $A\overline{B}C = 1$ 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

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

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

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

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

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

Section 4-2 Laws and Rules of Boolean Algebra

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

8. Refer to Table 4-1 in the textbook.

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

Section 4-3 DeMorgan's Theorems

9. (a) $\overline{A+\overline{B}} = \overline{A}\,\overline{\overline{B}} = \overline{\boldsymbol{A}}\boldsymbol{B}$

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

(c) $\overline{A+B+C} = \overline{\boldsymbol{A}}\,\overline{\boldsymbol{B}}\,\overline{\boldsymbol{C}}$

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

(e) $\overline{A(B+C)} = \overline{A}+\overline{(B+C)} = \overline{\boldsymbol{A}}+\overline{\boldsymbol{B}}\,\overline{\boldsymbol{C}}$

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

(g) $\overline{AB+CD} = \overline{(AB)}\,\overline{(CD)} = (\overline{\boldsymbol{A}}+\overline{\boldsymbol{B}})(\overline{\boldsymbol{C}}+\overline{\boldsymbol{D}})$

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

10. (a) $\overline{A\overline{B}(C+\overline{D})} = \overline{A\overline{B}}+\overline{(C+\overline{D})} = \overline{\boldsymbol{A}}+\boldsymbol{B}+\overline{\boldsymbol{C}}\boldsymbol{D}$

(b) $\overline{AB(CD+EF)} = \overline{AB}+\overline{(CD+EF)} = \overline{A}+\overline{B}+\overline{(CD)}\,\overline{(EF)}$

$= \overline{\boldsymbol{A}}+\overline{\boldsymbol{B}}+(\overline{\boldsymbol{C}}+\overline{\boldsymbol{D}})(\overline{\boldsymbol{E}}+\overline{\boldsymbol{F}})$

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

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

$= \overline{A\overline{B}\,\overline{C}\,\overline{D}}+\overline{\overline{A}+B+C+\overline{D}} = \overline{\boldsymbol{A}}+\boldsymbol{B}+\boldsymbol{C}+\boldsymbol{D}+\boldsymbol{A}\overline{\boldsymbol{B}}\,\overline{\boldsymbol{C}}\boldsymbol{D}$

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

$= AB+(\overline{CD})(\overline{\overline{E}F})+(\overline{\overline{AB}})(\overline{\overline{CD}})$

$= \boldsymbol{AB}+(\overline{\boldsymbol{C}}+\overline{\boldsymbol{D}})(\boldsymbol{E}+\overline{\boldsymbol{F}})+\boldsymbol{ABCD}$

11. (a) $\overline{\overline{(\overline{ABC})(\overline{EFG})}+\overline{(\overline{HIJ})(\overline{KLM})}} = \overline{\overline{\overline{ABC}}+\overline{\overline{EFG}}+\overline{\overline{HIJ}}+\overline{\overline{KLM}}}$

$= \overline{ABC+EFG+HIJ+KLM} = (\overline{ABC})(\overline{EFG})(\overline{HIJ})(\overline{KLM})$

$= (\overline{\boldsymbol{A}}+\overline{\boldsymbol{B}}+\overline{\boldsymbol{C}})(\overline{\boldsymbol{E}}+\overline{\boldsymbol{F}}+\overline{\boldsymbol{G}})(\overline{\boldsymbol{H}}+\overline{\boldsymbol{I}}+\overline{\boldsymbol{J}})(\overline{\boldsymbol{K}}+\overline{\boldsymbol{L}}+\overline{\boldsymbol{M}})$

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

$= \overline{A}B\overline{C}(\overline{C}+\overline{D})+BC = \overline{A}B\overline{C}+\overline{A}B\overline{C}\,\overline{D}+BC = \overline{A}B\overline{C}(1+\overline{D})+BC$

$= \overline{\boldsymbol{A}}\boldsymbol{B}\overline{\boldsymbol{C}}+\boldsymbol{BC}$

(c) $\overline{\overline{\overline{(A+B)}\,\overline{(C+D)}\,\overline{(E+F)}\,\overline{(G+H)}}}$

$= (\overline{A+B})(\overline{C+D})(\overline{E+F})(\overline{G+H}) = \overline{\boldsymbol{A}}\,\overline{\boldsymbol{B}}\,\overline{\boldsymbol{C}}\,\overline{\boldsymbol{D}}\,\overline{\boldsymbol{E}}\,\overline{\boldsymbol{F}}\,\overline{\boldsymbol{G}}\,\overline{\boldsymbol{H}}$

Section 4-4 Boolean Analysis of Logic Circuits

12. (a) $AB = X$

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

(c) $A + B = X$

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

13. See Figure 4-1.

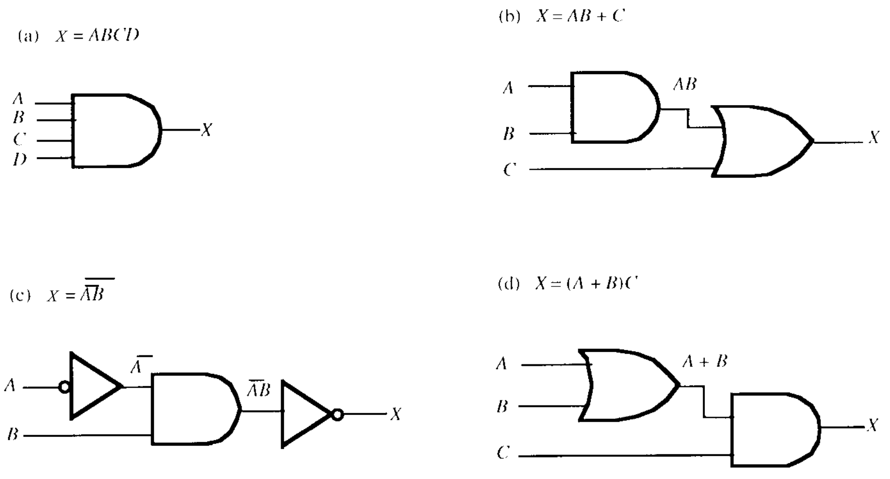
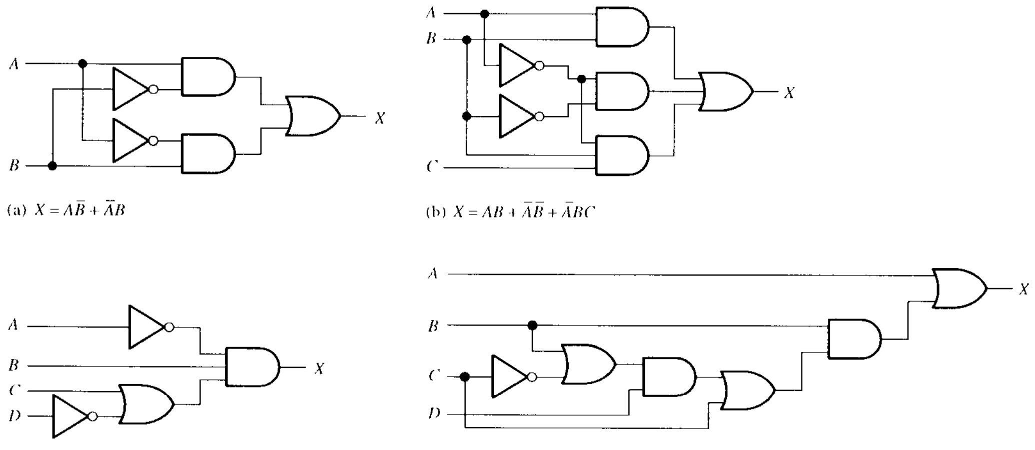


FIGURE 4-1

14. See Figure 4-2.

15. See Figure 4-3.



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

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

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

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

FIGURE 4-3

16. (a) See Figure 4-4(a).
(b) See figure 4-4(b).

FIGURE 4-3

17. See Tables 4-1 and 4-2.

Table 4-1

INPUTS			OUTPUT
$\overline{VCR}$	$\overline{CAMI}$	$\overline{RDY}$	$\overline{RECORD}$
0	0	0	0
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	1

Table 4-2

INPUTS			OUTPUT
RTS	*ENABLE*	$\overline{BUSY}$	$\overline{SEND}$
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

18. (a) $X = A + B$

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

(b) $X = AB$

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

(c) $X = AB + BC$

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

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

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

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

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

Section 4-5 Simplification Using Boolean Algebra

19. (a) $A(A + B) = AA + BB = A + AB = A(1 + B) = \boldsymbol{A}$

(b) $A(\overline{A} + AB) = A\overline{A} + AAB = 0 + AB = \boldsymbol{AB}$

(c) $BC + \overline{B}C = C(B + \overline{B}) = C(1) = \boldsymbol{C}$

(d) $A(A + \overline{A}B) = AA + A\overline{A}B = A + (0)B = A + 0 = \boldsymbol{A}$

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

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

20. (a) $(A+\overline{B})(A+C) = AA + AC + A\overline{B} + \overline{B}C = A + AC + A\overline{B} + \overline{B}C$

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

(b) $\overline{A}B + \overline{A}B\overline{C} + \overline{A}BCD + \overline{A}B\overline{C}\,\overline{D}E = \overline{A}B(1 + \overline{C} + CD + \overline{C}\,\overline{D}E) = \overline{A}B(1)$

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

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

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

$= \boldsymbol{A + C}$

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

$= AB + AB\overline{C} + 0 + 0 = AB(1 + \overline{C}) = \boldsymbol{AB}$

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

$= AB + \overline{AB}C = \boldsymbol{AB} + \boldsymbol{C}$

21. (a) $BD + B(D+E) + \overline{D}(D+F) = BD + BD + BE + \overline{D}D + \overline{D}F$

$= BD + BE + 0 + \overline{D}F = \boldsymbol{BD} + \boldsymbol{BE} + \boldsymbol{\overline{D}F}$

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

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

(c) $(B + BC)(B + \overline{B}C)(B + D) = B(1 + C)(B + C)(B + D)$

$= B(B + C)(B + D) = (BB + BC)(B + D) = (B + BC)(B + D)$

$= B(1 + C)(B + D) = B(B + D) = BB + BD = B + BD = B(1 + D) = \boldsymbol{B}$

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

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

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

$= CD(1 + \overline{A}) + AB\overline{C} + AB\overline{D} = CD + AB\overline{C} + AB\overline{D} = CD + AB(\overline{CD}) = \boldsymbol{CD} + \boldsymbol{AB}$

(e) $ABC[AB + \overline{C}(BC + AC)] = ABABC + ABC\overline{C}(BC + AC)$

$= ABC + 0(BC + AC) = \boldsymbol{ABC}$

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

(a) See Figure 4-5.

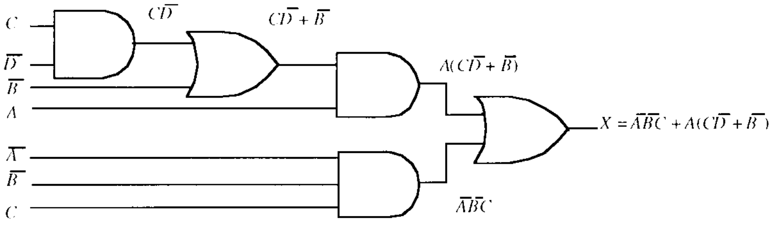

C
$\overline{D}$
$\overline{B}$
A
$\overline{A}$
$\overline{B}$
C
$C\overline{D}$
$C\overline{D}+\overline{B}$
$A(C\overline{D}+\overline{B})$
$\overline{A}\overline{B}C$
$X=\overline{A}\overline{B}C+A(C\overline{D}+\overline{B})$

FIGURE 4-5

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

$$=\overline{B}(A+C)+ACD=\boldsymbol{A\overline{B}}+\boldsymbol{\overline{B}C}+\boldsymbol{AC\overline{D}}$$

(b) See Figure 4-6.

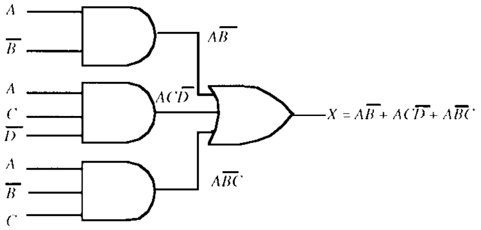


FIGURE 4-6

$$X=A\overline{B}+AC\overline{D}+A\overline{B}C=A\overline{B}(1+C)+AC\overline{D}=\boldsymbol{A\overline{B}}+\boldsymbol{AC\overline{D}}$$

(c) See Figure 4-7.

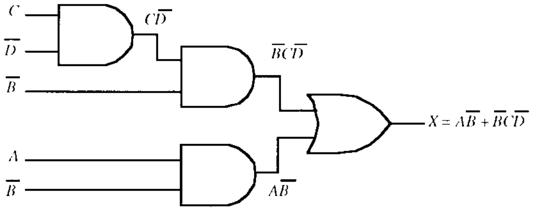


FIGURE 4-7

$X=\boldsymbol{A\overline{B}}+\boldsymbol{\overline{B}C\overline{D}}$ No further simplification is possible.

(d) See Figure 4-8.

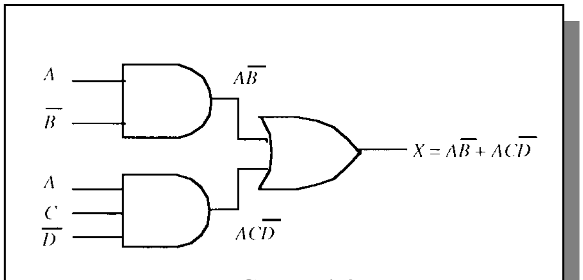


FIGURE 4-8

$X = \boldsymbol{A\overline{B} + AC\overline{D}}$ No further simplification is possible.

Section 4-6 Standard Forms of Boolean Expressions

23. (a) $(A+B)(C+\overline{B}) = AC + BC + B\overline{B} + A\overline{B} = \boldsymbol{AC + BC + A\overline{B}}$

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

(c) $(A + C)(AB + AC) = AAB + AAC + ABC + ACC = AB + AC + ABC + ACC$
$= (AB + AC)(1 + C) = \boldsymbol{AB + AC}$

24. (a) $AB + CD(A\overline{B} + CD) = AB + A\overline{B}CD + CDCD = AB + A\overline{B}CD + CD$

$= AB(A\overline{B} + 1)CD = \boldsymbol{AB + CD}$

(b) $AB(\overline{B}\overline{C} + BD) = AB\overline{B}\overline{C} + ABBD = 0 + ABD = \boldsymbol{ABD}$

(c) $A + B[AC + (B + \overline{C})D] = A + ABC + (B + \overline{C})BD$

$= A + ABC + BD + B\overline{C}D = A(1 + BC) + BD + B\overline{C}D = A + BD(1 + \overline{C})$

$= \boldsymbol{A + BD}$

25. (a) The domain is A, B, C
The standard SOP is: $A\overline{B}C + A\overline{B}\overline{C} + ABC + \overline{A}BC$

(b) The domain is A, B, C
The standard SOP is: $ABC + A\overline{B}C + \overline{A}\overline{B}C$

(c) The domain is A, B, C
The standard SOP is: $ABC + AB\overline{C} + A\overline{B}C$

26. (a) $AB + CD = ABCD + ABC\overline{D} + AB\overline{C}D + AB\overline{C}\overline{D} + \overline{A}\overline{B}CD + \overline{A}BCD + A\overline{B}CD$

(b) $ABD = ABCD + AB\overline{C}D$

(c) $A + BD = A\overline{B}\overline{C}\overline{D} + A\overline{B}\overline{C}D + A\overline{B}C\overline{D} + A\overline{B}CD + AB\overline{C}\overline{D} + AB\overline{C}D$
$+ ABC\overline{D} + ABCD + \overline{A}B\overline{C}D + \overline{A}BCD$

27. (a) $A\overline{B}C + A\overline{B}\,\overline{C} + ABC + \overline{A}BC$: 101 + 100 + 111 + 011

(b) $ABC + A\overline{B}C + \overline{A}\,\overline{B}C$: 111 + 101 + 001

(c) $ABC + AB\overline{C} + A\overline{B}C$: 111 + 110 + 101

28. (a) $ABCD + ABC\overline{D} + AB\overline{C}D + AB\overline{C}\,\overline{D} + \overline{A}\,\overline{B}CD + \overline{A}BCD + A\overline{B}CD$:
1111 + 1110 + 1101 + 1100 + 0011 + 0111 + 1011

(b) $ABCD + AB\overline{C}D$: 1111 + 1101

(c) $A\overline{B}\,\overline{C}\,\overline{D} + A\overline{B}\,\overline{C}D + A\overline{B}C\overline{D} + A\overline{B}CD + AB\overline{C}\,\overline{D} + AB\overline{C}D$
$+\ ABC\overline{D} + ABCD + \overline{A}B\overline{C}D + \overline{A}BCD$:
1000 + 1001 + 1010 + 1011 + 1100 + 1101 + 1110 + 1111 + 0101 + 0111

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

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

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

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

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

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

Section 4-7 Boolean Expressions and Truth Tables

31. (a) Table 4-3

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 4-4

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

32. (a) Table 4-5

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 4-6

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

33. (a) $\overline{A}B + AB\overline{C} + \overline{A}\,\overline{C} + A\overline{B}C = \overline{A}BC + \overline{A}B\overline{C} + AB\overline{C} + \overline{A}\,\overline{B}\,\overline{C} + A\overline{B}C$

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

Table 4-7

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

Table 4-8

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

34. (a) Table 4-9

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

(b) Table 4-10

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

35. (a) Table 4-11

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

(b) Table 4-12

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

36. (a) $X = \overline{A}\,\overline{B}C + A\overline{B}\,\overline{C} + A\overline{B}C + ABC$

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

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

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

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

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

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

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

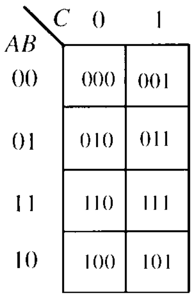
$(\overline{A}+B+C+\overline{D})(\overline{A}+B+\overline{C}+D)(\overline{A}+\overline{B}+C+\overline{D})(\overline{A}+\overline{B}+\overline{C}+D)$

Section 4-8 The Karnaugh Map

37. See Figure 4-9.

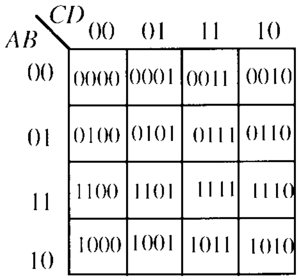
38. See Figure 4-10.

39. See Figure 4-11.



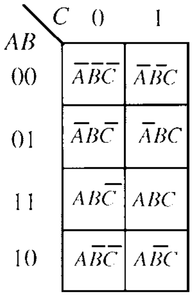
AB \ C	0	1
00	000	001
01	010	011
11	110	111
10	100	101

FIGURE 4-9



AB \ CD	00	01	11	10
00	0000	0001	0011	0010
01	0100	0101	0111	0110
11	1100	1101	1111	1110
10	1000	1001	1011	1010

FIGURE 4-10



AB \ C	0	1
00	$\overline{A}\,\overline{B}\,\overline{C}$	$\overline{A}\,\overline{B}C$
01	$\overline{A}B\overline{C}$	$\overline{A}BC$
11	$AB\overline{C}$	ABC
10	$A\overline{B}\,\overline{C}$	$A\overline{B}C$

FIGURE 4-11

Section 4-9 Karnaugh Map SOP Minimization

40. See Figure 4-12.

FIGURE 4-12

41. See Figure 4-13.

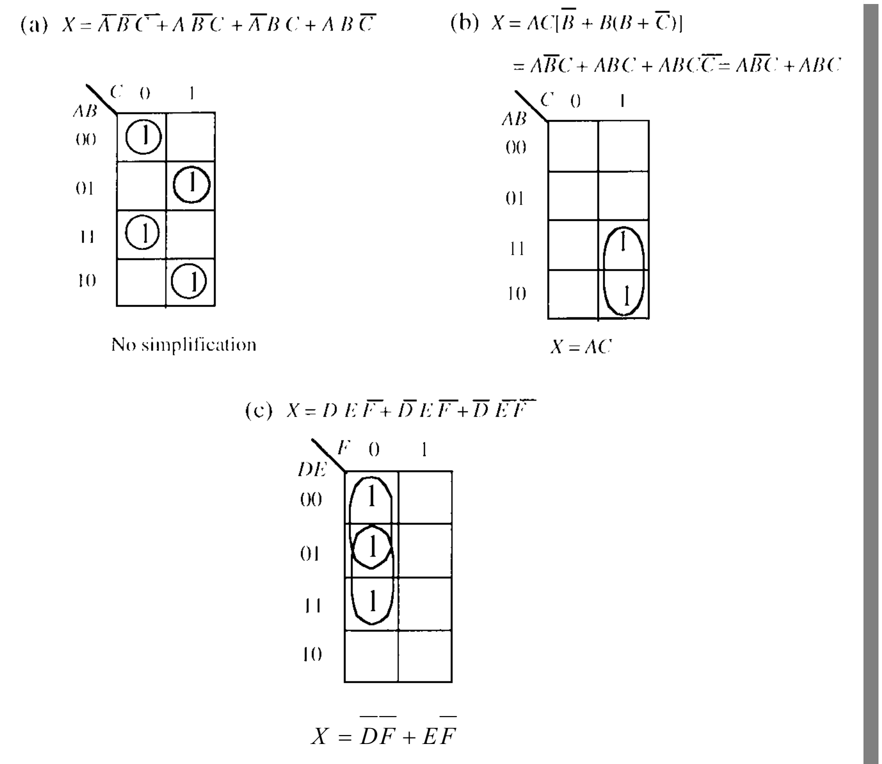

(a) $X = \bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + A\bar{B}C + AB\bar{C}$
C 0 1
AB
00
01
11
10
1
1
1
1
No simplification
(b) $X = AC[\bar{B} + B(B + \bar{C})]$
$= A\bar{B}C + ABC + ABC\bar{C} = A\bar{B}C + ABC$
C 0 1
AB
00
01
11
10
1
1
$X = AC$
(c) $X = \bar{D}\bar{E}\bar{F} + \bar{D}E\bar{F} + DE\bar{F}$
F 0 1
DE
00
01
11
10
1
1
1
$X = \bar{D}\bar{F} + E\bar{F}$

FIGURE 4-13

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

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

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

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

$= ABC + AB\bar{C} + A\bar{B}C + A\bar{B}\bar{C} + \bar{A}BC + ABC$

$= ABC + AB\bar{C} + A\bar{B}C + A\bar{B}\bar{C} + \bar{A}BC$

(c) $A\bar{B}\bar{C}D + AC\bar{D} + B\bar{C}D + \bar{A}BC\bar{D}$

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

$= A\bar{B}\bar{C}D + ABC\bar{D} + A\bar{B}C\bar{D} = AB\bar{C}D + \bar{A}B\bar{C}D + \bar{A}BC\bar{D}$

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

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

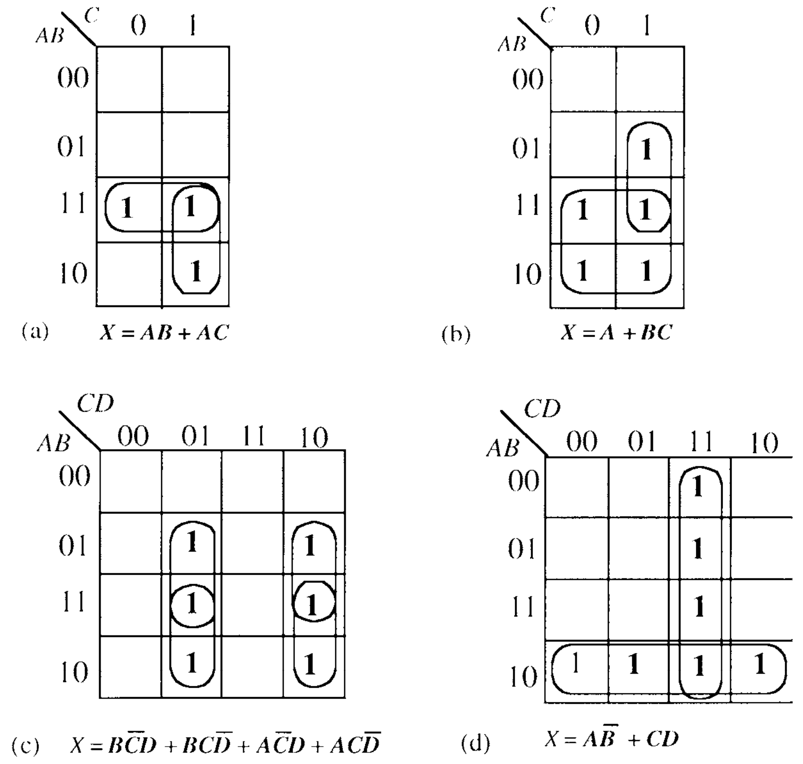
$= A\bar{B}\bar{C}\bar{D} + A\bar{B}\bar{C}D + ABC\bar{D} + ABCD + A\bar{B}\bar{C}D + ABCD + A\bar{B}CD + \bar{A}BCD$

$+ \bar{A}\bar{B}CD + AB\bar{C}D + \bar{A}B\bar{C}D + ABCD$

$= A\bar{B}\bar{C}\bar{D} + A\bar{B}\bar{C}D + ABC\bar{D} + ABCD + A\bar{B}CD + \bar{A}BCD + \bar{A}\bar{B}CD + AB\bar{C}D + \bar{A}B\bar{C}D$

$= \bar{A}\bar{B}CD + \bar{A}B\bar{C}D + \bar{A}BCD + A\bar{B}C\bar{D} + A\bar{B}\bar{C}D + A\bar{B}CD + AB\bar{C}D + ABCD + ABC\bar{D}$

43. See Figure 4-14.



(a) $X = AB + AC$

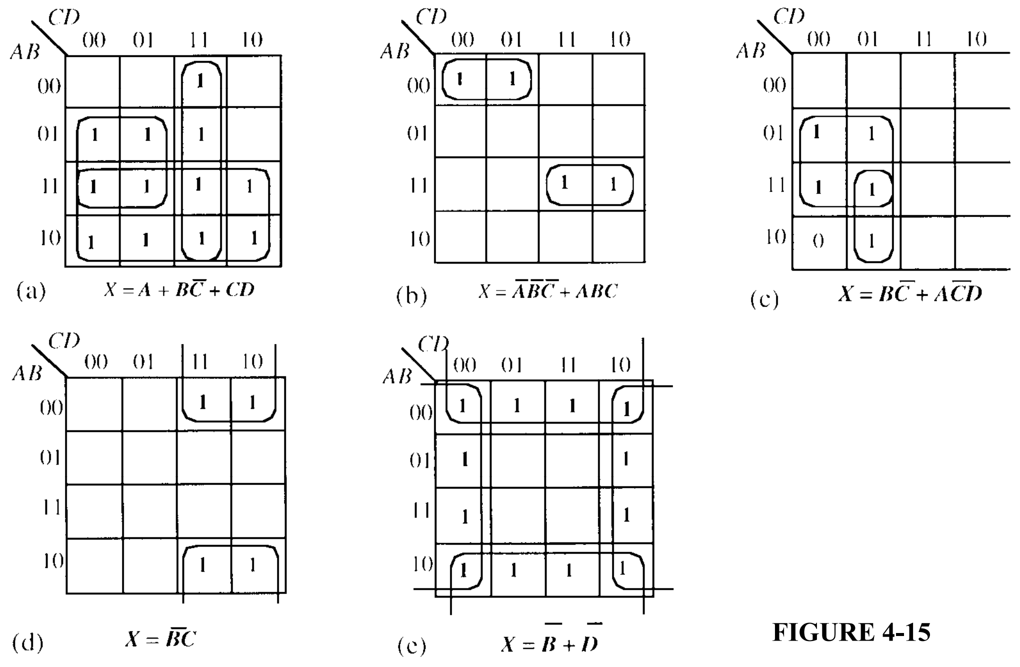
(b) $X = A + BC$

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

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

FIGURE 4-14

44. See Figure 4-15.



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

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

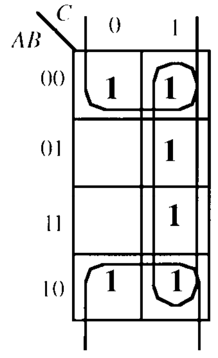
(c) $X = B\overline{C} + A\overline{C}D$

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

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

FIGURE 4-15

45. Plot the 1’s from Table 4-11 in the text on the map as shown in Figure 4-16 and simplify.



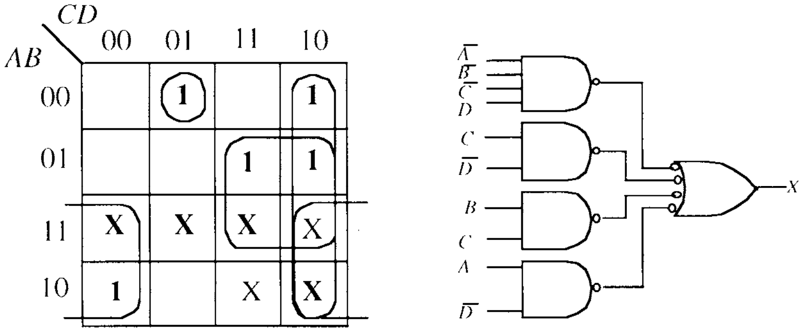
$X = \overline{B} + C$

FIGURE 4-16

46. Plot the 1’s from Table 4-12 in the text on the map as shown in Figure 4-17 and simplify.

FIGURE 4-17

47. See Figure 4-18.



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

FIGURE 4-18

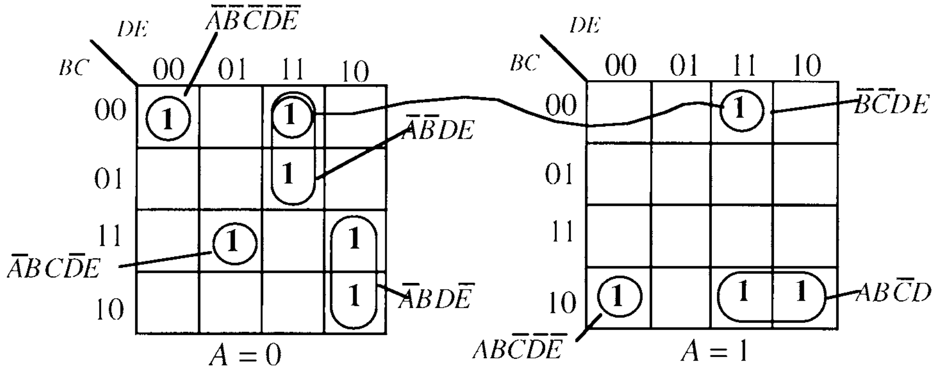
Section 4-10 Five-Variable Karnaugh Maps

48. $X = A\bar{B}C\bar{D}E + \bar{A}\,\bar{B}C\bar{D}E.$

See Figure 4-19.

FIGURE 4-19

49. See Figure 4-20.



$$X = \bar{A}\bar{B}\bar{C}\bar{D}\bar{E} + \bar{A}BC\bar{D}E + AB\bar{C}\bar{D}\bar{E} + \bar{A}\bar{B}DE + \bar{A}BD\bar{E} + \bar{B}\bar{C}DE + AB\bar{C}D$$

FIGURE 4-20

50. See Figure 4-21.

FIGURE 4-21

Section 4-11 Describing Logic with an HDL

51.

```
entity AND_OR is
        port (A, B, C, D, E, F, G, H, I: in bit; X: out bit);
end entity AND_OR;
architecture Logic of AND_OR is
begin
        X <= (A and B and C) or (D and E and F) or (G and H and I);
end architecture Logic;
```

52. The VHDL program:

```
entity SOP is
        port (A, B, C: in bit; X: out bit);
end entity SOP;
architecture Logic of SOP is
begin
        Y <= (A and not B and C) or (not A and not B and C) or
                (A and not B and not C) or (not A and B and C);
end architecture Logic;
```

System Application Activity

53. An LED display is more suitable for low-light conditions because LEDs emit light and LCDs do not.

54. The purpose of the invalid code detector is to detect the codes 1010, 1011, 1100, 1101, 1110, and 1111 to activate the display for letters.

55. The standard SOP expression for segment *c* is:
$c = H_3\bar{H}_2H_1\bar{H}_0 + H_3\bar{H}_2H_1H_0 + H_3H_2\bar{H}_1H_0$
This expression is minimized in Figure 4-22.

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

FIGURE 4-22

56. The standard SOP expression for segment *d* is:
$d = H_3\bar{H}_2H_1H_0 + H_3H_2\bar{H}_1\bar{H}_0 + H_3H_2\bar{H}_1H_0 + H_3H_2H_1\bar{H}_0$
This expression is minimized in Figure 4-23.

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

FIGURE 4-23

The standard SOP expression for segment *e* is:
$e = H_3\overline{H}_2H_1\overline{H}_0 + H_3\overline{H}_2H_1H_0 + H_3H_2\overline{H}_1\overline{H}_0 + H_3H_2\overline{H}_1H_0 + H_3H_2H_1\overline{H}_0$
This expression is minimized in Figure 4-24.

The standard expression requires five 4-input AND gates, one 5-input OR gate, and 3 inverters.
The minimum expression requires one 3-input AND gate.

FIGURE 4-24

The standard SOP expression for segment *f* is:
$f = H_3\overline{H}_2H_1\overline{H}_0 + H_3\overline{H}_2H_1H_0 + H_3H_2\overline{H}_1\overline{H}_0 + H_3H_2H_1\overline{H}_0$
This expression is minimized in Figure 4-25.

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

FIGURE 4-25

The standard SOP expression for segment *g* is:
$g = H_3\overline{H}_2H_1\overline{H}_0 + H_3\overline{H}_2H_1H_0 + H_3H_2\overline{H}_1H_0 + H_3H_2H_1\overline{H}_0$
This expression is minimized in Figure 4-26.

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

FIGURE 4-26

Special Design Problems

57. Connect the OR gate output for each segment to an inverter and then use the inverter output to drive the segment with a HIGH.

58. See Figure 4-27. *F* = 1111
The expression for segment *a* to include the letter *F* is:
$a = H_3\overline{H}_2H_1\overline{H}_0 + H_3H_2\overline{H}_1\overline{H}_0 + H_3H_2H_1\overline{H}_0 + H_3H_2H_1H_0$
The expression is minimized in Figure 4-27.

FIGURE 4-27

59. See Figure 4-28. Segment *b* is used for letters *A* and *d*.
$b = H_3\bar{H}_2H_1\bar{H}_0 + H_3H_2\bar{H}_1H_0$

FIGURE 4-28

See Figure 4-29. Segment *c* is used for letters *A*, *b*, and *d*.
$c = H_3\bar{H}_2H_1\bar{H}_0 + H_3\bar{H}_2H_1H_0 + H_3H_2\bar{H}_1H_0$

FIGURE 4-29

See Figure 4-30. Segment *d* is used for *b*, *C*, *d*, and *E*.
$d = H_3\bar{H}_2H_1H_0 + H_3H_2\bar{H}_1\bar{H}_0 + H_3H_2\bar{H}_1H_0 + H_3H_2H_1\bar{H}_0$

FIGURE 4-30

See Figure 4-31. Segment *e* is used for *A*, *b*, *C*, *d*, *E*, and *F*.
$e = H_3\bar{H}_2H_1\bar{H}_0 + H_3\bar{H}_2H_1H_0 + H_3H_2\bar{H}_1\bar{H}_0 + H_3H_2\bar{H}_1H_0 + H_3H_2H_1\bar{H}_0 + H_3H_2H_1H_0$

Since segment *e* is active-LOW for all letters, *e* = 0.

FIGURE 4-31

See Figure 4-32. Segment *f* is used for *A*, *b*, *C*, *E*, and *F*.
$f = H_3\bar{H}_2H_1\bar{H}_0 + H_3\bar{H}_2H_1H_0 + H_3H_2\bar{H}_1\bar{H}_0 + H_3H_2H_1\bar{H}_0 + H_3H_2H_1H_0$

FIGURE 4-32

See Figure 4-33. Segment *g* is used in *A*, *b*, *d*, *E*, and *F*.
$g = H_3\bar{H}_2H_1\bar{H}_0 + H_3\bar{H}_2H_1H_0 + H_3H_2\bar{H}_1H_0 + H_3H_2H_1\bar{H}_0 + H_3H_2H_1H_0$

FIGURE 4-33

60. The invalid code detector must disable the display when any numerical input (0-9) occurs. A HIGH enables the display and a LOW disables it. A circuit that detects the numeric codes and produces a LOW is shown in Figure 4-34.

FIGURE 4-34

Multisim Troubleshooting Practice

61. Input *A* inverter output open.

62. Input *A* of segment *e* OR gate open.

63. Segment *b* OR gate output open.

CHAPTER 5
COMBINATIONAL LOGIC ANALYSIS

Section 5-1 Basic Combinational Logic Circuits

1. See Figure 5-1.

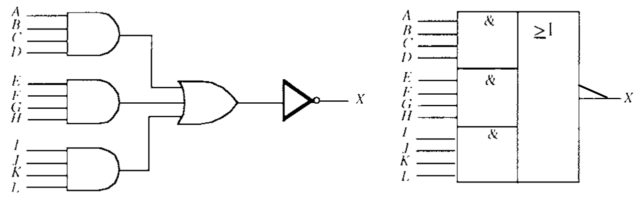


FIGURE 5-1

2. (a) $X = \overline{A}B + \overline{A} + AC$

(b) $X = \overline{\overline{A}B + \overline{A}CD + DB\overline{D}}$

3. (a) $X = ABB$

(b) $X = AB + B$

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

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

(e) $X = \overline{\overline{A}BC}$

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

4. See Figure 5-2 for the circuit corresponding to each expression.

(a) $X = (A + B)(C + D) = AC + AD + BC + BD$

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

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

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

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

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

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

$= (\overline{A} + \overline{B})(\overline{C} + \overline{D}) + (\overline{E} + \overline{F})(\overline{G} + \overline{H}) = \overline{A}\,\overline{C} + \overline{B}\,\overline{C} + \overline{A}\,\overline{D} + \overline{B}\,\overline{D} + \overline{E}\,\overline{G} + \overline{F}\,\overline{G} + \overline{E}\,\overline{H} + \overline{F}\,\overline{H}$

5. (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) $X = \overline{\overline{A}BC}$

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

Chapter 5

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

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

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

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	1
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	1	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) $X = \overline{\overline{\overline{\overline{(\overline{A}+B)}(\overline{BC})}+D}}$

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

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

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	1	1

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

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

For all other entries $X = 0$.
X = *don't care*
An abbreviated table is shown because there are 256 combinations.

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

Section 5-2 Implementing Combinational Logic

8. Let *G* = guard, *S* = switch, *M* = motor temp, and *P* = power. See Figure 5-3.
$P=\overline{\overline{G}S+MS}$

9. $\boldsymbol{X} = \overline{ABCD + EFGH}$

10. See Figure 5-4.

FIGURE 5-4

11. See Figure 5-5.

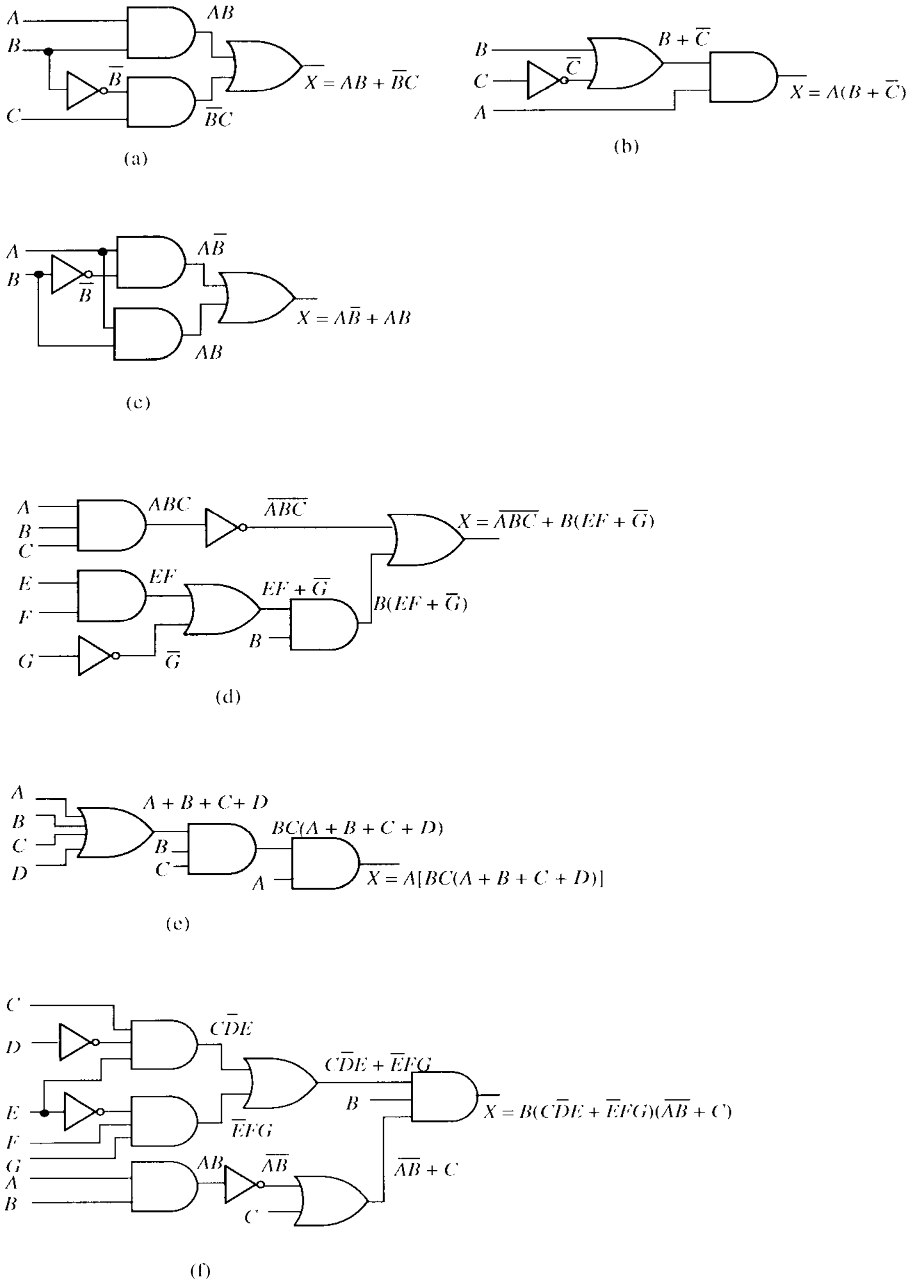
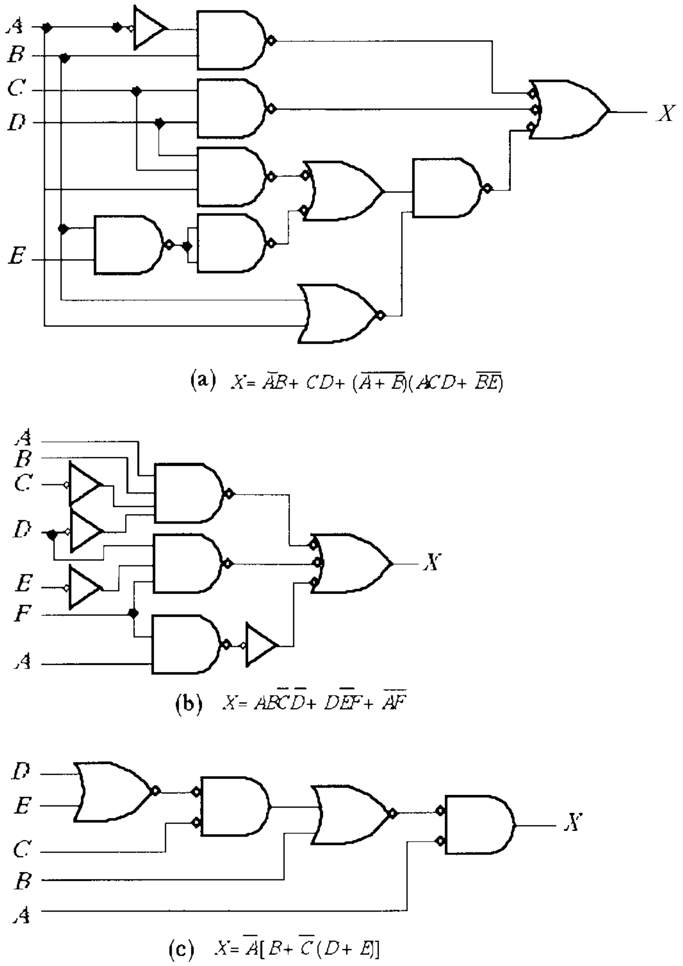


FIGURE 5-5

12. See Figure 5-6.



(a) $X = \overline{A}B + CD + (\overline{A + B})(ACD + \overline{BE})$

(b) $X = AB\overline{C}\overline{D} + D\overline{E}F + \overline{AF}$

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

FIGURE 5-6

13. $X = \overline{A}\overline{B}\overline{C} + \overline{A}B\overline{C} + A\overline{B}\overline{C} + AB\overline{C} + ABC$
See Figure 5-7.

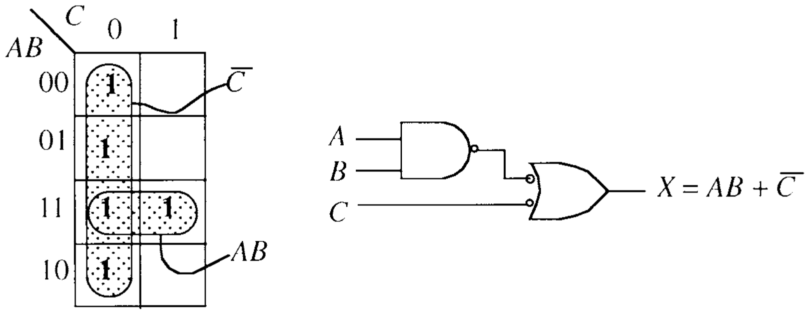


FIGURE 5-7

14. $X = \overline{A}\overline{B}C\overline{D} + \overline{A}\overline{B}CD + \overline{A}B\overline{C}\overline{D} + A\overline{B}\overline{C}\overline{D} + A\overline{B}\overline{C}D + A\overline{B}C\overline{D} + A\overline{B}CD + ABCD$
See Figure 5-8.

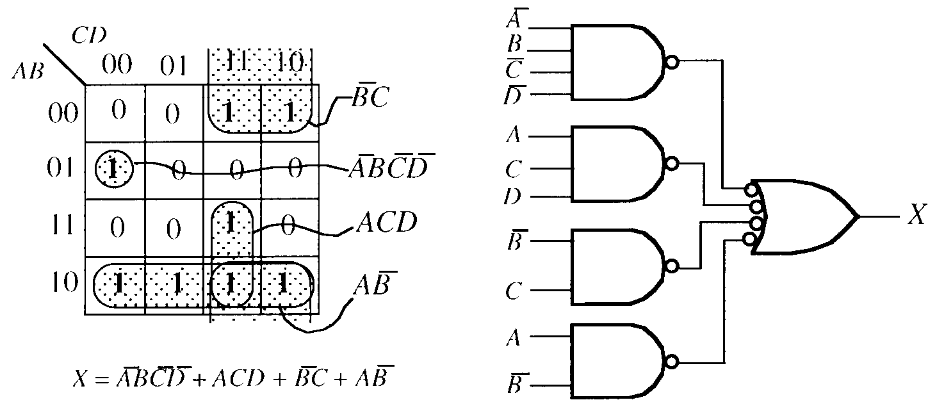


FIGURE 5-8

15. $X = AB + ABC = AB(1 + C) = \boldsymbol{AB}$

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

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

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

Since C is a don't care variable, the output depends only on A and B as shown by the two-variable truth table above which is implemented with the AND gate in Figure 5-9.

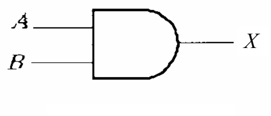


FIGURE 5-9

16. $X = \overline{\overline{\overline{AB}\,\overline{B+C}} + C} = \overline{\overline{\overline{AB}\,\overline{B+C}}}\,\overline{C} = (\overline{AB})(\overline{B+C})\overline{C} = (\overline{A}+\overline{B})(\overline{B}\,\overline{C})\overline{C}$

$= (\overline{A}\,\overline{B}\,\overline{C} + \overline{B}\,\overline{C})\overline{C} = \overline{A}\,\overline{B}\,\overline{C} + \overline{B}\,\overline{C} = \overline{B}\,\overline{C}(A+1) = \boldsymbol{\overline{B}\,\overline{C}}$

See Figure 5-10.

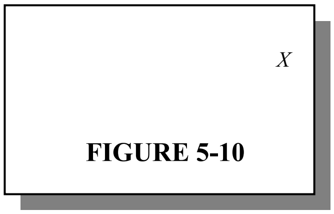


FIGURE 5-10

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

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

No simplification. See Figure 5-11.

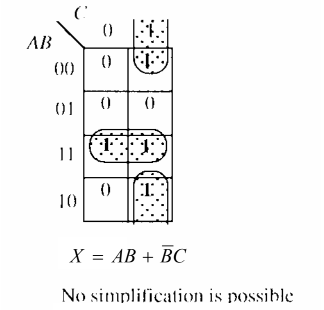


FIGURE 5-11

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

No simplification. Equation can be expressed in another form, as indicated in Figure 5-12.

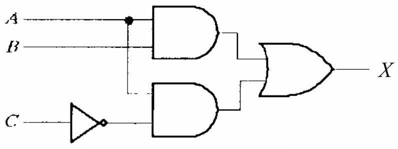


FIGURE 5-12

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

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

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

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

See Figure 5-13.

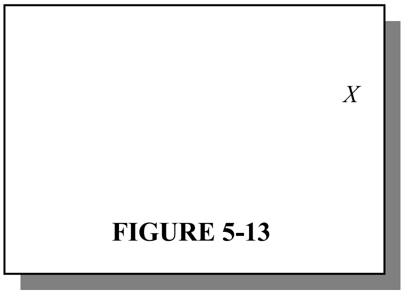


FIGURE 5-13

(e) $X = A(BC(A + B + C + D)) = ABCA + ABCB + ABCC + ABCD$

$= ABC + ABC + ABC + ABCD = ABC + ABC(1 + D)$

$= ABC + ABC = \boldsymbol{ABC}$

See Figure 5-14.

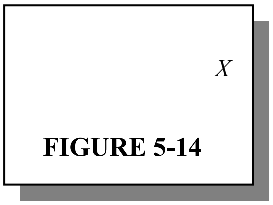


FIGURE 5-14

(f) $X = B(C\overline{DE} + \overline{E}FG)(\overline{AB} + C) = (BC\overline{DE} + B\overline{E}FG)(\overline{A} + \overline{B} + C)$

$= \overline{A}BC\overline{DE} + \overline{A}B\overline{E}FG + BC\overline{DE} + BC\overline{E}FG$

$= BC\overline{DE}(\overline{A} + 1) + \overline{A}B\overline{E}FG + BC\overline{E}FG$

$= \boldsymbol{BC\overline{DE}} + \boldsymbol{\overline{A}B\overline{E}FG} + \boldsymbol{BC\overline{E}FG}$

See Figure 5-15.

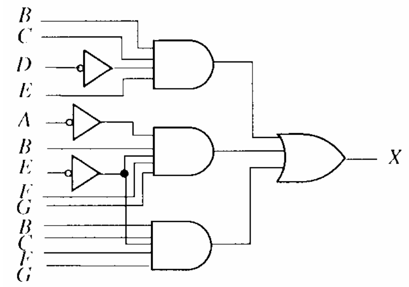


FIGURE 5-15

18. (a) $X = \overline{A}B + CD + (\overline{A+B})(ACD + \overline{BE}) = \overline{A}B + CD + \overline{A}\,\overline{B}(ACD + \overline{B} + \overline{E})$

$= \overline{A}B + CD + \overline{A}\,\overline{B} + \overline{A}\,\overline{B}\,\overline{E} = \overline{A}(B + \overline{B}) + CD + \overline{A}\,\overline{B}\,\overline{E}$

$= \overline{A} + \overline{A}\,\overline{B}\,\overline{E} + CD = \overline{A}(1 + \overline{B}\,\overline{E}) + CD = \boldsymbol{\overline{A} + CD}$

See Figure 5-16.

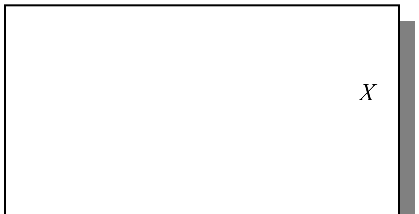


FIGURE 5-16

(b) $X = AB\overline{C}\,\overline{D} + D\overline{E}F + \overline{AF} = AB\overline{C}\,\overline{D} + \overline{D}EF + \overline{A} + \overline{F}$

$= \boldsymbol{\overline{A} + B\overline{C}\,\overline{D} + \overline{F} + D\overline{E}}$

See Figure 5-17.

FIGURE 5-17

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

See Figure 5-18.

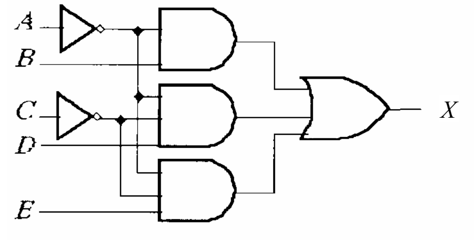


FIGURE 5-18

19. The SOP expressions are developed as follows and the resulting circuits are shown in Figure 5-19.

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

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

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

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

$= \overline{A} + B(1 + C) + D = \overline{\boldsymbol{A}} + \boldsymbol{B} + \boldsymbol{D}$

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

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

$= (\overline{AB})(\overline{CD}) + (\overline{EF})(\overline{GH}) = (\overline{A} + \overline{B})(\overline{C} + \overline{D}) + (\overline{E} + \overline{F})(\overline{G} + \overline{H})$

$= \overline{\boldsymbol{A}}\,\overline{\boldsymbol{C}} + \overline{\boldsymbol{B}}\,\overline{\boldsymbol{C}} + \overline{\boldsymbol{A}}\,\overline{\boldsymbol{D}} + \overline{\boldsymbol{B}}\,\overline{\boldsymbol{D}} + \overline{\boldsymbol{E}}\,\overline{\boldsymbol{G}} + \overline{\boldsymbol{F}}\,\overline{\boldsymbol{G}} + \overline{\boldsymbol{E}}\,\overline{\boldsymbol{H}} + \overline{\boldsymbol{F}}\,\overline{\boldsymbol{H}}$

FIGURE 5-19

Chapter 5

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

20. See Figure 5-20.

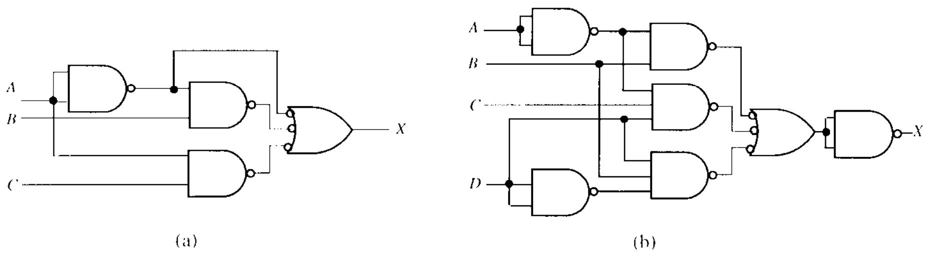


FIGURE 5-20

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

See Figure 5-21.

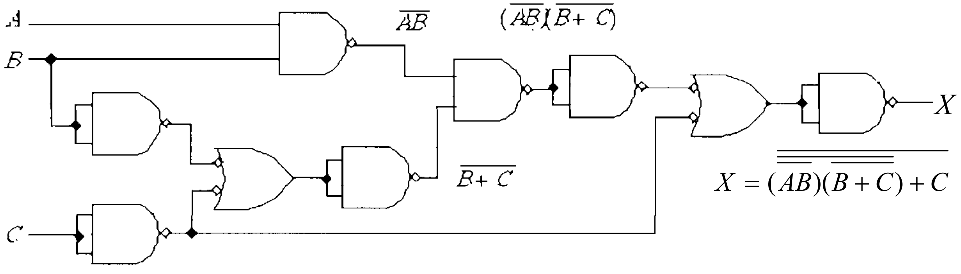

A
B
C
$\overline{AB}$
$(\overline{\overline{AB})(\overline{B+C}})$
$\overline{B+C}$
X
$X = \overline{\overline{(\overline{AB})(\overline{B+C})}+C}$

FIGURE 5-21

22. See Figure 5-22.

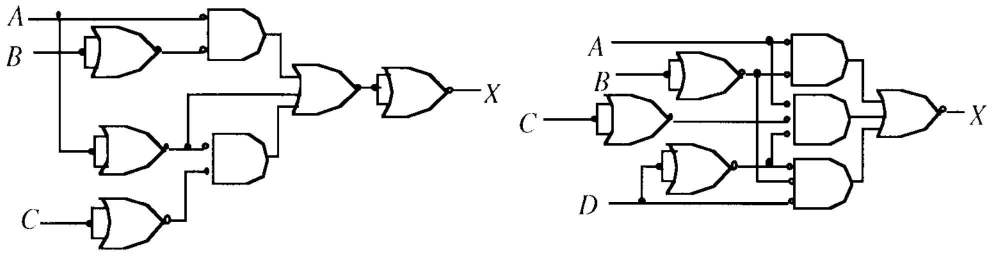


FIGURE 5-22

23. See Figure 5-23.

FIGURE 5-23

Section 5-4 Combinational Logic Using NAND and NOR Gates

24. (a) $X = ABC$

See Figure 5-24.

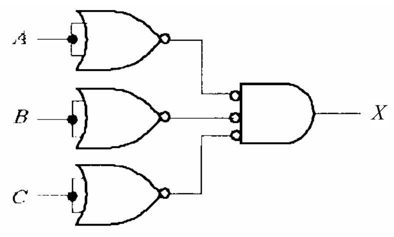


FIGURE 5-24

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

See Figure 5-25.

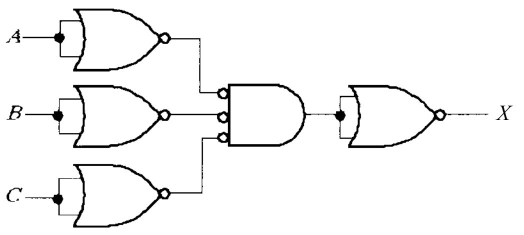


FIGURE 5-25

(c) $X = A + B$

See Figure 5-26.

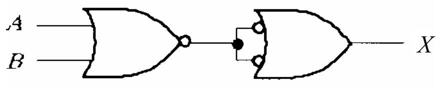


FIGURE 5-26

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

See Figure 5-27.

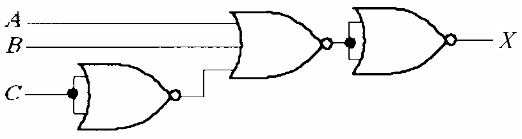


FIGURE 5-27

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

See Figure 5-28.

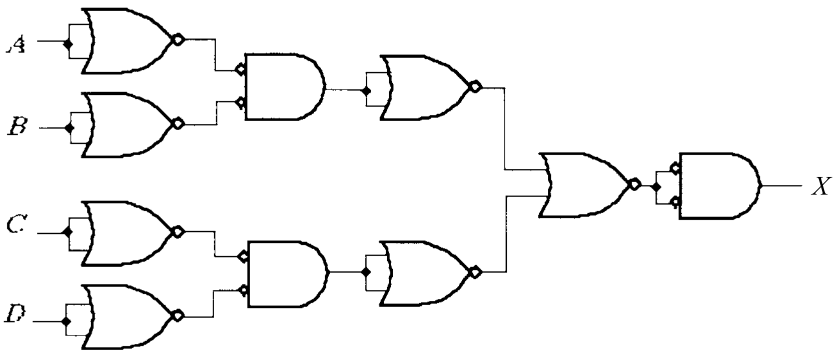


FIGURE 5-28

(f) $X = (A + B)(C + D)$

See Figure 5-29.

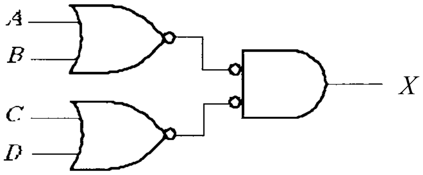


FIGURE 5-29

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

See Figure 5-30.

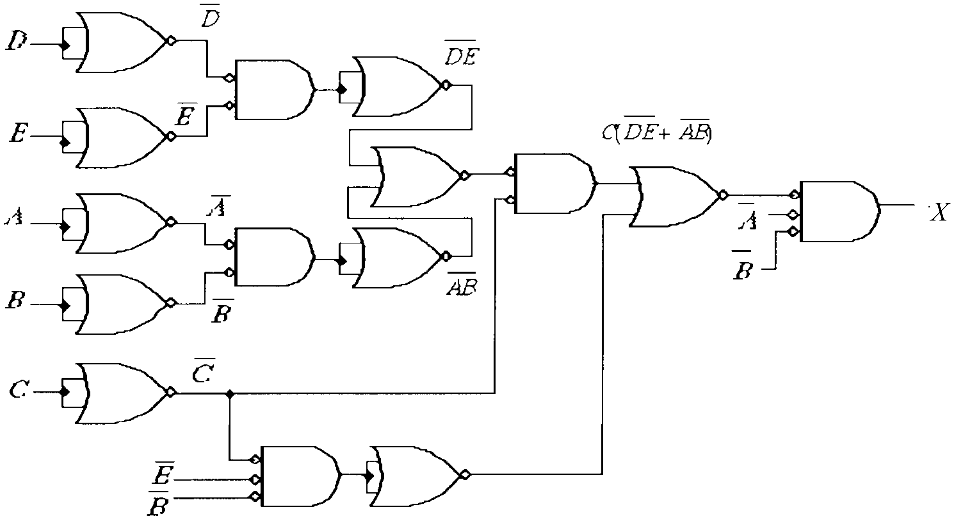


FIGURE 5-30

25. (a) $X = ABC$

See Figure 5-31.

FIGURE 5-31

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

See Figure 5-32.

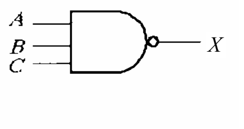


FIGURE 5-32

(c) $X + A + B$

See Figure 5-33.

FIGURE 5-33

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

See Figure 5-34.

FIGURE 5-34

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

See Figure 5-35.

FIGURE 5-35

(f) $X = (A + B)(C + D)$

See Figure 5-36.

FIGURE 5-36

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

See Figure 5-37.

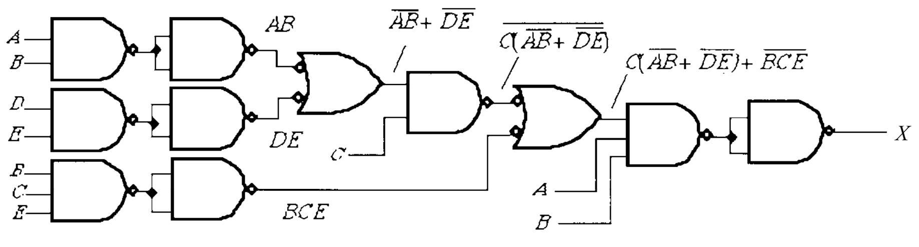

A
B
D
E
E
C
E
AB
$\overline{AB}+\overline{DE}$
$\overline{C(\overline{AB}+\overline{DE})}$
$C(\overline{AB}+\overline{DE})+\overline{BCE}$
DE
C
BCE
A
B
X

FIGURE 5-37

26. (a) $X = AB$

See Figure 5-38.

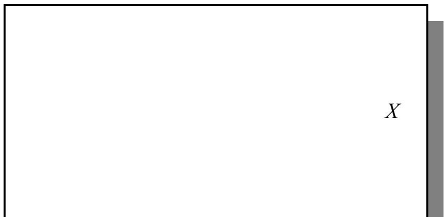


FIGURE 5-38

(b) $X = A + B$

See Figure 5-39.

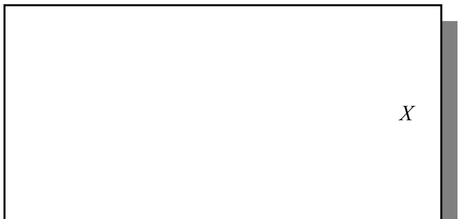


FIGURE 5-39

(c) $X = AB + C$

See Figure 5-40.

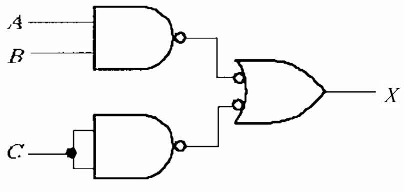


FIGURE 5-40

(d) $X = ABC + D$

See Figure 5-41.

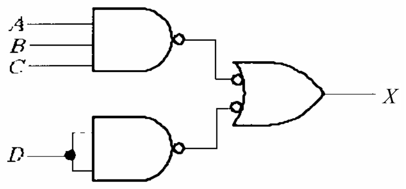


FIGURE 5-41

(e) $X = A + B + C$
See Figure 5-42.

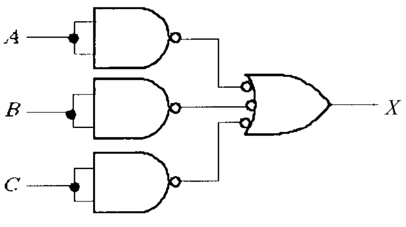


FIGURE 5-42

(f) $X = ABCD$
See Figure 5-43.

FIGURE 5-43

(g) $X = A(CD + B) = \boldsymbol{ACD + AB}$

See Figure 5-44.

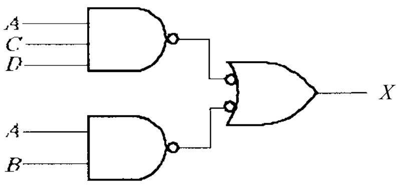


FIGURE 5-44

(h) $X = AB(C + DEF) + CE(A + B + F)$
$= \boldsymbol{ABC + ABDEF + CEA + CEB + CEF}$

See Figure 5-45.

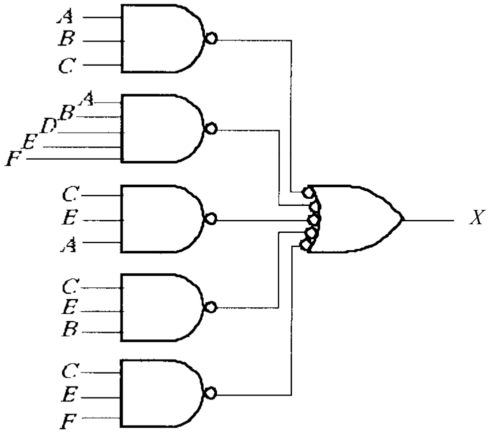


FIGURE 5-45

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

See Figure 5-46.

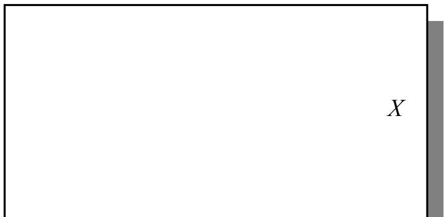


FIGURE 5-46

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

See Figure 5-47.

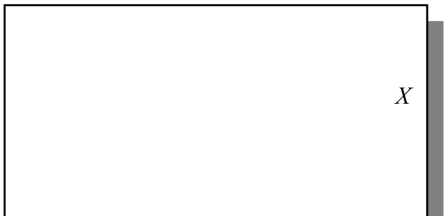


FIGURE 5-47

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

See Figure 5-48.

FIGURE 5-48

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

See Figure 5-49.

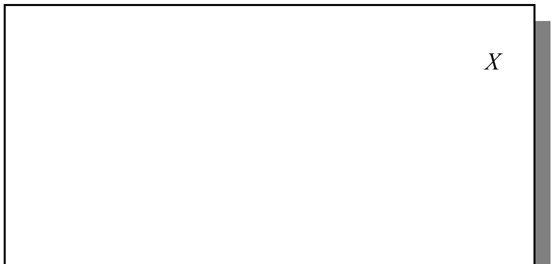


FIGURE 5-49

(e) $X = A[BC(A + B + C + D)] = ABCA + ABCB + ABCC + ABCD$

$= ABC + ABC + ABC + ABCD + ABC(1 + D) = \boldsymbol{ABC}$

See Figure 5-50.

FIGURE 5-50

(f) $X = B(C\overline{D}E + \overline{E}FG)(\overline{AB} + C) = B(C\overline{D}E + \overline{E}FG)(\overline{A} + \overline{B} + C)$

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

$= \overline{A}B\overline{E}FG + B\overline{B}\overline{E}FG + BC\overline{D}E + BC\overline{E}FG$

$= \boldsymbol{\overline{A}B\overline{E}FG + BC\overline{D}E + BC\overline{E}FG}$

See Figure 5-51.

FIGURE 5-51

Section 5-5 Logic Circuit Operation with Pulse Waveform Inputs

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

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

See Figure 5-52.

FIGURE 5-52

30. *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.

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

See Figure 5-53.

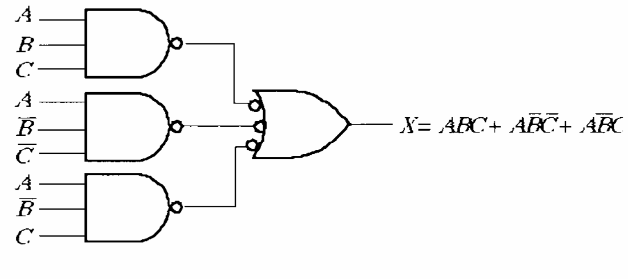


FIGURE 5-53

31. *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.

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

See Figure 5-54.

FIGURE 5-54

32. See Figure 5-55.

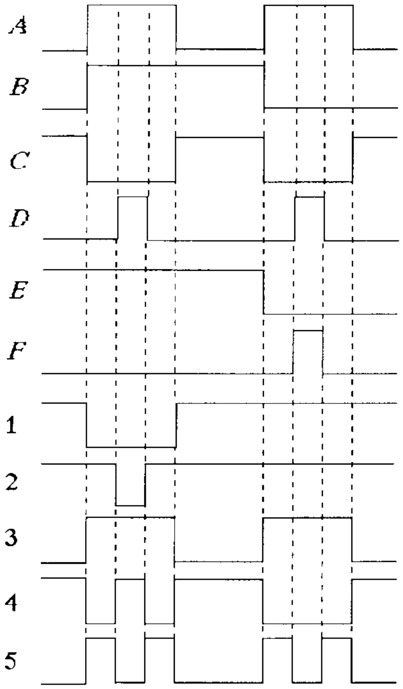


FIGURE 5-55

33. The output pulse is sufficiently wide. It is greater than 25 ns. A maximum is not specified. See Figure 5-56.

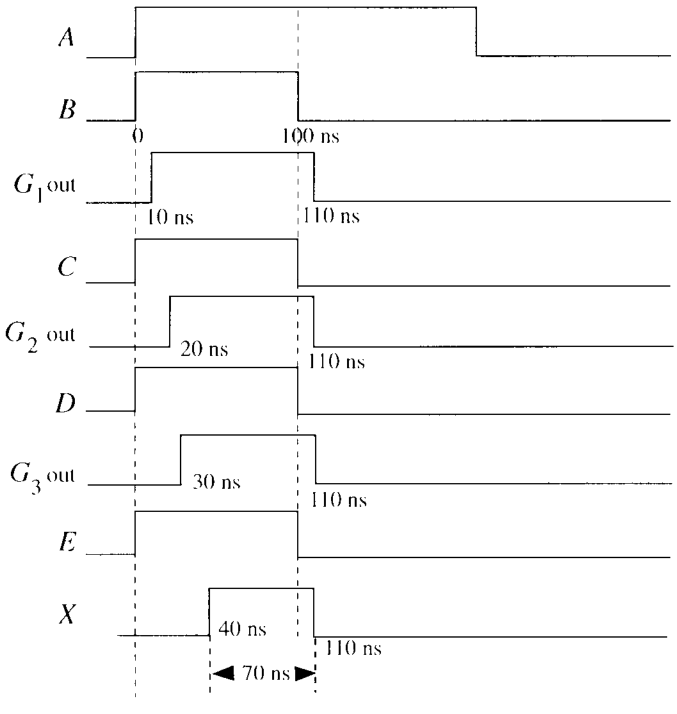


FIGURE 5-56

Section 5-6 Troubleshooting

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

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.**

35. $X = ABC + D\overline{E}$

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

36. $X = AB + CD + EF$

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*.

37. See Figure 5-57.

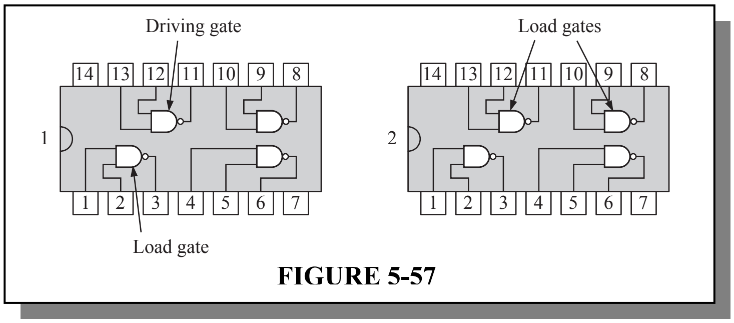


FIGURE 5-57

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

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

39. (a) $X = (\overline{A} + \overline{B} + C)E + (C + \overline{D})E = \overline{A}E + \overline{B}E + CE + \cancel{CE} + \overline{D}E$

$= \overline{A}E + \overline{B}E + CE + \overline{D}E$

See Figure 5-58.

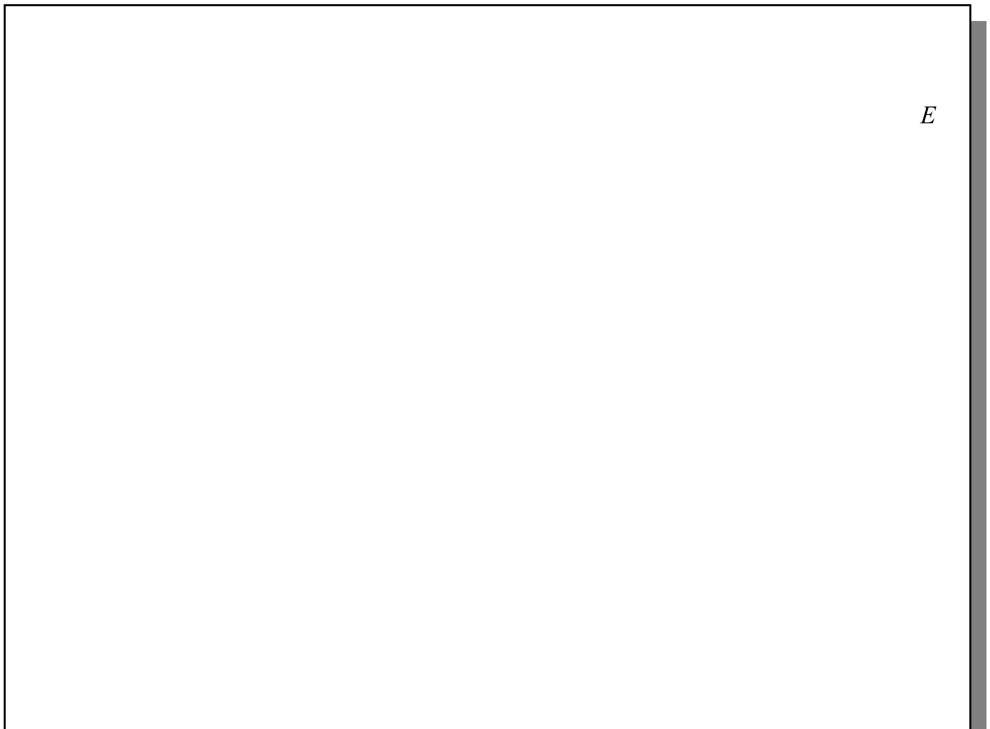


FIGURE 5-58

(b) $X = E + E(\overline{D} + C) = E(1 + \overline{D} + C) = \boldsymbol{E}$

Waveform *X* is the same as waveform *E*, in Figure 5-58. 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) $X = E + E(\overline{A} + \overline{B} + C) = E(1 + \overline{A} + \overline{B} + C) = \boldsymbol{E}$

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.

40. TP = $\overline{\overline{A}\,\overline{B}+\overline{C}\,\overline{D}}$

The output of the $\overline{C}\,\overline{D}$ gate is *stuck LOW*. See Figure 5-59.

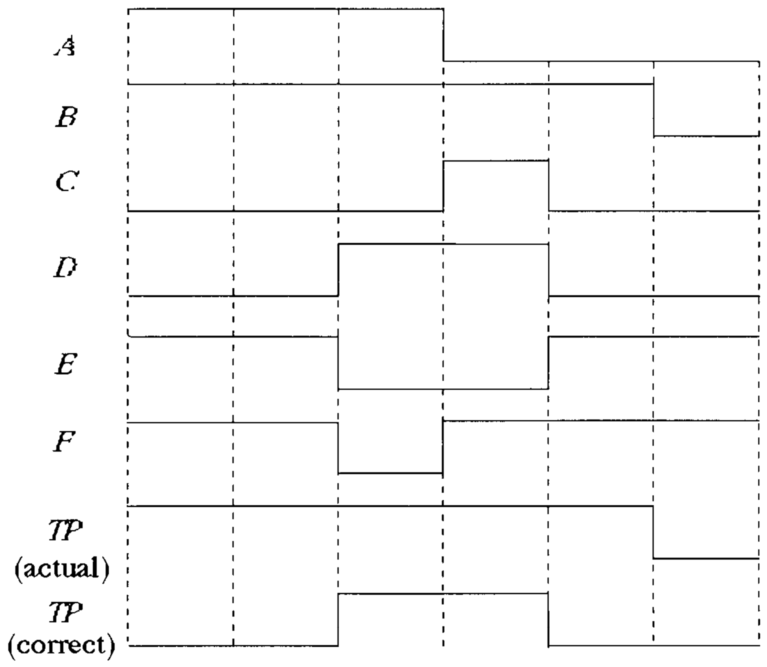


FIGURE 5-59

Section 5-7 Combinational Logic with VHDL

41. X <= A **and** B **and** C

42.

```
entity Circuit5_54b is
        port (A, B, C, D: in bit; X: out bit);
end entity Circuit5_54b;
architecture LogicFunction of Circuit5_54b 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;
```

43.(e)

```
entity Circuit5_55e is
        port (A, B, C: in bit; X: out bit);
end entity Circuit5_55e;
architecture LogicFunction of Circuit5_55e is
begin
        X <= (not A and B) or B or (B and not C) or (not A and not C) or (B and not C) or not C;
        end architecture LogicFunction;
```

(f)

```
entity Circuit5_55f is
        port (A, B, C: in bit;  X: out bit);
end entity Circuit5_55f;
architecture LogicFunction of Circuit5_55f is
begin
        X <= (A or B) and (not B or C);
end architecture Logic Function;
```

44. See Figure 5-60 for input/output, gate, and signal labeling.

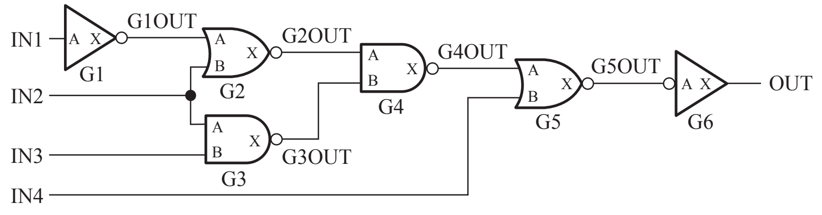


FIGURE 5-60

```
--Program for the logic circuit in Figure 5-60 (textbook Figure 5-56(d))
entity (Circuit5_56d is
        port (IN1, IN2, IN3, IN4: in bit; OUT: out bit);
end entity Circuit5_56d;
architecture LogicOperation of Circuit5_56d 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 NOR gate;
--Component declaration for NAND gate
component NANDgate is
        port (A, B: in bit; X: out bit);
end component NANDgate;
signal G1OUT, G2OUT, G3OUT, G4OUT, G5OUT: bit;
begin
        G1:  Inverter port map (A => IN1, X => G1OUT);
        G2:  NORgate port map (A => G1OUT, B => IN2, X => G2OUT);
        G3:  NAND gate port map (A => IN2, B => IN3, X => G3OUT);
        G4:  NANDgate port map (A => G2OUT, B => G3OUT, X => G4OUT);
        G5:  NORgate port map (A => G4OUT, B => IN4, X => G5OUT);
        G6:  Inverter port map  (A => G5OUT, X => OUT);
end architecture  LogicOperation;
```

45. See Figure 5-61 for input/output, gate, and signal labeling.

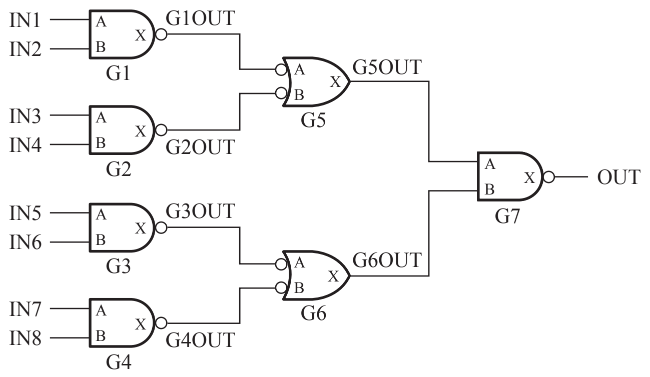


FIGURE 5-61

```
--Program for the logic circuit in Figure 5-61 (textbook Figure 5-56(f))
entity  Circuit5_56f is
        port (IN1, IN2, IN3, IN4, IN5, IN6, IN7, IN8:  in bit; OUT:  out bit);
end entity Circuit5_56f;
architecture LogicFunction of Circuit5_56f 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;
```

46. $X = \overline{A}\,\overline{B}\,\overline{C} + \overline{A}B\overline{C} + A\overline{B}\,\overline{C} + AB\overline{C} + ABC$

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

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

47.

```
--Program for textbook Figure 5-66 data flow approach
entity Fig5_66 is
        port (A, B, C, D, E:  in bit; X: out bit);
end entity Fig5_66;
architecture DataFlow of Fig5_66 is
begin
        X <= (A and B and C) or (D and not E)
end architecture DataFlow;
```

See Figure 5-62 for the circuit in textbook Figure 5-66 modified for the structural approach.

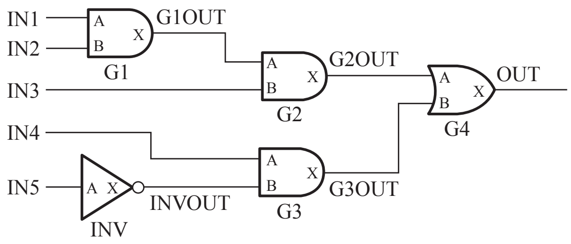


FIGURE 5-62

```
--Program for textbook Figure 5-66 structural approach
entity Fig5_66 is
        port (IN1, IN2, IN3, IN4, IN5:  in bit; OUT: out bit);
end entity Fig5_66;
architecture Structure of Fig5_66 is
--Component declaration for AND gate
component AND_gate is
        port (A, B: in bit; X: out bit);
end component AND_gate;
--Component declaration for OR gate
component OR_gate is
        port (A, B: in bit; X: out bit);
end component OR_gate;
--Component declaration for Inverter
component Inverter is
        port (A: in bit; X: out bit);
end component Inverter;
signal G1OUT, G2OUT, G3OUT, INVOUT: bit;
begin
        G1:  AND_gate port map (A => IN1, B => IN2, X => G1OUT);
        G2:  AND_gate port map (A => G1OUT, B => IN3, X => G2OUT);
        INV:  Inverter port map (A => IN5, X => INVOUT);
        G3:  AND_gate port map (A => IN4, B => INVOUT, X => G3OUT);
        G4:  OR_gate port map (A => G2OUT, B => G3OUT, X => OUT);
end architecture Structure;
```

48.

```
--Program for textbook Figure 5-70 data flow approach
entity Fig5_70 is
        port (A, B, C, D, E:  in bit; X: out bit);
end entity Fig5_70;
architecture DataFlow of Fig5_70 is
begin
        X <= (not A or not B or C) and E or (C or not D) and E;
end architecture DataFlow;
```

See Figure 5-63 for the circuit in textbook Figure 5-70 labeled for the structural approach.

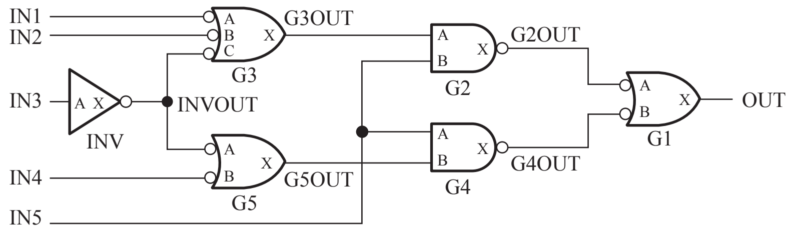


FIGURE 5-63

```
--Program for textbook Figure 5-70 structural approach
entity Fig5_70 is
        port (IN1, IN2, IN3, IN4, IN5:  in bit; OUT: out bit);
end entity Fig5_70;
architecture Structure of Fig5_70 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;
```

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

$$\begin{aligned} X &= (\overline{\overline{A}\,\overline{B} + \overline{A}\,\overline{C} + \overline{A}\,\overline{D} + \overline{B}\,\overline{C} + \overline{B}\,\overline{D} + \overline{D}\,\overline{C}}) \\ &= ((A+B)(A+C)(A+D)(B+C)(B+D)(D+C)) \\ &= (A+B)(A+C)(A+D)(B+C)(B+D)(D+C) \end{aligned}$$

The truth table is:

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

50.

```
--Program for textbook Figure 5-72 data flow approach
entity Fig5_72 is
        port (A1, A2, B1, B2:  in bit; X: out bit);
end entity Fig5_72;
architecture LogicCircuit of Fig5_72 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;
```

51. 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 Circuit5_72 is
        port (IN1, IN2, IN3, (IN4: in bit; OUT: out bit);
end entity Circuit 5_72;
architecture Logic of Circuit 5_72 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;
```

System Application Activity

52. $V_{\text{inlet}} = \overline{L}_{\text{min}} + \overline{L}_{\text{max}} F_{\text{inlet}}$

See Figure 5-64.

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

See Figure 5-65.

FIGURE 5-65

54. See Figure 5-66.

FIGURE 5-66

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

See Figure 5-67.

FIGURE 5-67

Chapter 5

Special Design Problems

56.

A_3	A_2	A_1	A_0	X
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

See Figure 5-68.

FIGURE 5-68

57. Let
X = Lamp on
A = Front door switch on
$\overline{A}$ = Front door switch off
B = Back door switch on
$\overline{B}$ = Back door switch off
$X = A\overline{B} + \overline{A}B$. This is an XOR operation.

See Figure 5-69.

FIGURE 5-69

58. See Figure 5-70.

FIGURE 5-70

Chapter 5

Multisim Troubleshooting Practice

59. Pin B of G1 open.

60. Pin C of OR gate open.

61. Inverter input open.

62. No fault.

CHAPTER 6
FUNCTIONS OF COMBINATIONAL LOGIC

Section 6-1 Basic Adders

1. (a) XOR (upper) output = 0, Sum output = 1, AND (upper) output = 0, AND (lower) output = 1, Carry output = 1
(b) XOR (upper) output = 1, Sum output = 0, AND (upper) output = 1, AND (lower) output = 0, Carry output = 1
(c) XOR (upper) output = 1, Sum output = 1, AND (upper) output = 0, AND (lower) output = 0, Carry output = 0

2. (a) $A = 0$, $B = 0$, $C_{\text{in}} = 0$
(b) $A = 1$, $B = 0$, $C_{\text{in}} = 0$ or $A = 0$, $B = 1$, $C_{\text{in}} = 0$
or $A = 0$, $B = 0$, $C_{\text{in}} = 1$
(c) $A = 1$, $B = 1$, $C_{\text{in}} = 1$
(d) $A = 1$, $B = 1$, $C_{\text{in}} = 0$ or $A = 0$, $B = 1$, $C_{\text{in}} = 1$
or $A = 1$, $B = 0$, $C_{\text{in}} = 1$

3. (a) $\Sigma = 1$, $C_{\text{out}} = 0$ (b) $\Sigma = 1$, $C_{\text{out}} = 0$
(c) $\Sigma = 0$, $C_{\text{out}} = 1$ (d) $\Sigma = 1$, $C_{\text{out}} = 1$

Section 6-2 Parallel Binary Adders

4.
$$\begin{array}{r} 111 \\ \underline{101} \\ 1100 \end{array}$$

See Figure 6-1.

FIGURE 6-1

5.

$$\begin{array}{r} 10101 \\ \underline{00111} \\ 11100 \end{array}$$

See Figure 6-2.

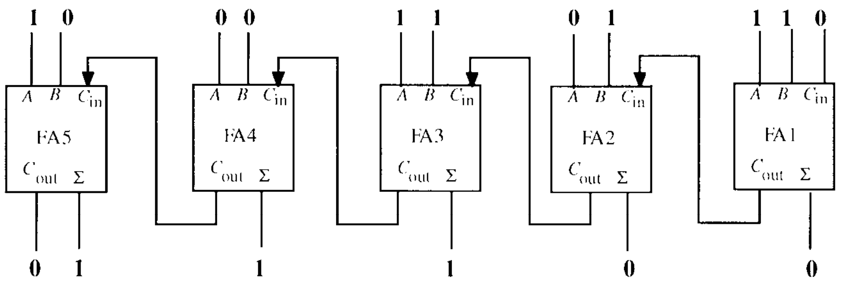


FIGURE 6-2

6. (a) When the $\overline{Add}/Subt$ is HIGH, the two numbers are subtracted.

(b) When the input is LOW, the numbers are added.

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

$$\begin{array}{rl} 1001 & \\ 0011 & \leftarrow \text{Complement of } B \\ \underline{\quad 1} & \leftarrow \text{LSB Carry input} \\ 1101 & = -3 \text{ in 2's comp} \end{array}$$

8. See Figure 6-3.

FIGURE 6-3

9.

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

$\Sigma_1 = 0110$
$\Sigma_2 = 1011$
$\Sigma_3 = 0110$
$\Sigma_4 = 0001$
$\Sigma_5 = 1000$

10.

$$\begin{array}{r} 0100 \\ \underline{1110} \\ 10010 \end{array}$$

Σ outputs should be $C_{out}\Sigma_4\Sigma_3\Sigma_2\Sigma_1 = 10010$.
The Σ_3 output (pin 2) is HIGH and should be LOW.

See Figure 6-4.

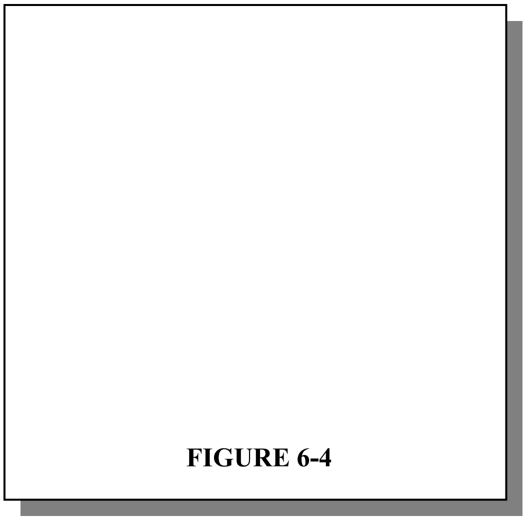


FIGURE 6-4

Section 6-3 Ripple Carry Versus Look-Ahead Carry Adders

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

12. 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 6-18 is shown in Figure 6-5.

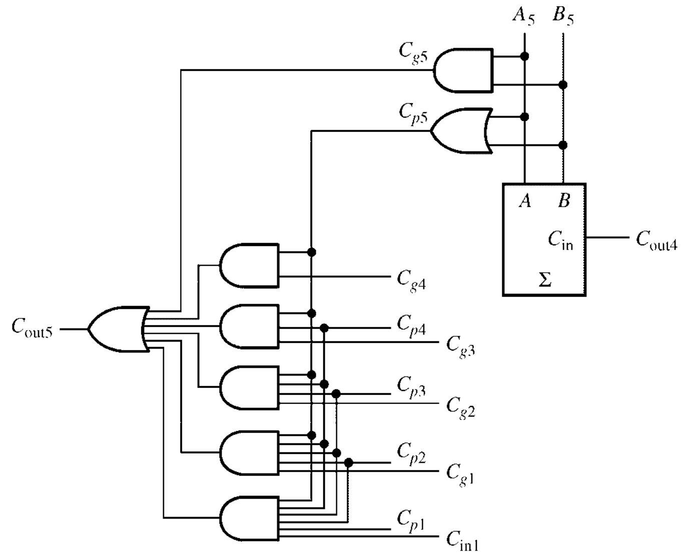


FIGURE 6-5

Section 6-4 Comparators

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

See Figure 6-6.

FIGURE 6-6

14. See Figure 6-7.

FIGURE 6-7

15. (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 6-5 Decoders

16. (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**

17. See Figure 6-8.

FIGURE 6-8

18. Change the AND gates to NAND gates in Figure 6-8.

19. $X = \overline{A_3}\,\overline{A_2}\,\overline{A_1}A_0 + A_3\overline{A_2}A_1\overline{A_0} + A_3A_2\overline{A_1}\,\overline{A_0} + A_3\overline{A_2}A_1A_0$

See Figure 6-9.

FIGURE 6-9

20. $Y = A_2A_1\overline{A_0} + A_2\overline{A_1}A_0 + \overline{A_2}A_1\overline{A_0}$

See Figure 6-10.

FIGURE 6-10

21. See Figure 6-11.

FIGURE 6-11

22. **0 1 6 9 4 4 4 8 0**

Section 6-6 Encoders

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

24. Pin 2 is for decimal 5, pin 5 is for decimal 8, and pin 12 is for decimal 2. The highest priority input is pin 5.

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

Section 6-7 Code Converters

25. (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} = $\mathbf{00110011}_{\text{BCD}}$ = $\mathbf{100001}_2$

26.

(a)	1010101010	binary	(b)	1111100000	binary
	1111111111	gray		1000010000	gray
(c)	0000001110	binary	(d)	1111111111	binary
	0000001001	gray		1000000000	gray

See Figure 6-12.

FIGURE 6-12

27.

(a)	1010000000	gray	(b)	0011001100	gray
	1100000000	binary		0010001000	binary
(c)	1111000111	gray	(d)	0000000001	gray
	1010000101	binary		0000000001	binary

See Figure 6-13.

FIGURE 6-13

Section 6-8 Multiplexers (Data Selectors)

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

29. See Figure 6-14.

FIGURE 6-14

30. See Figure 6-15.

FIGURE 6-15

Section 6-9 Demultiplexers

31. See Figure 6-16.

FIGURE 6-16

Section 6-10 Parity Generators/Checkers

32. See Figure 6-17.

FIGURE 6-17

33. See Figure 6-18.

FIGURE 6-18

Section 6-11 Troubleshooting

34. 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 Σ and C_{out} output waveforms that match the waveforms given in the problem, indicating that C_{in} is indeed stuck HIGH, perhaps shorted to V_{CC}.

See Figure 6-19 for the correct output waveforms.

FIGURE 6-19

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

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

Key	A_3	A_2	A_1	A_0
None	1	1	1	1
0	1	1	1	1
1	1	1	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

TABLE 1

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

Hold down keys	Depress keys one at a time	A_3	A_2	A_1	A_0
1	0	1	1	1	0
2	1, 0	1	1	0	1
3	2, 1, 0	1	1	0	0
4	3, 2, 1, 0	1	0	1	1
5	4, 3, 2, 1, 0	1	0	1	0
6	5, 4, 3, 2, 1, 0	1	0	0	1
7	6, 5, 4, 3, 2, 1, 0	1	0	0	0
8	7, 6, 5, 4, 3, 2, 1, 0	0	1	1	1
9	8, 7, 6, 5, 4, 3, 2, 1, 0	0	1	1	0

TABLE 2

37. (a) Open A_1 input acts as a HIGH. All binary values corresponding to a BCD number having a 1's value of 0, 1, 4, 5, 8, or 9 will be off by 2. This will first be seen for a BCD value of 00000000.

(b) Open C_{out} of top adder. All values not normally involving a carry out will be off by 32. This will first be seen for a BCD value of 00000000.

(c) The Σ_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) Σ_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.

38. (a) The 1Y1 output of the 74LS139 is *stuck HIGH* or *open*; B cathode open.

(b) No power; EN input to the 74LS139 is *open*.

(c) The *f* output of the 74LS47 is *stuck HIGH*.

(d) The frequency of the data select input is too *low*.

39. 1. Place a LOW on pin 7 (Enable).
2. Apply a HIGH to D_0 and a LOW to D_1 through D_7.
3. Go through the binary sequence on the select inputs and check Y and $\overline{Y}$ according to Table 3.

S_2	S_1	S_0	Y	$\overline{Y}$
0	0	0	1	0
0	0	1	0	1
0	1	0	0	1
0	1	1	0	1
1	0	0	0	1
1	0	1	0	1
1	1	0	0	1
1	1	1	0	1

TABLE 3

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

D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	Y	$\overline{Y}$	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

TABLE 4

40. The Σ EVEN output of the 74LS280 should be HIGH and the output of the error gate should be HIGH because of the error condition. Possible faults are:

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

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

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

FIGURE 6-20

System Application Activity

42. See Figure 6-21.

FIGURE 6-21

43. See Figure 6-22.

FIGURE 6-22

Special Design Problems

44. See Figure 6-23.

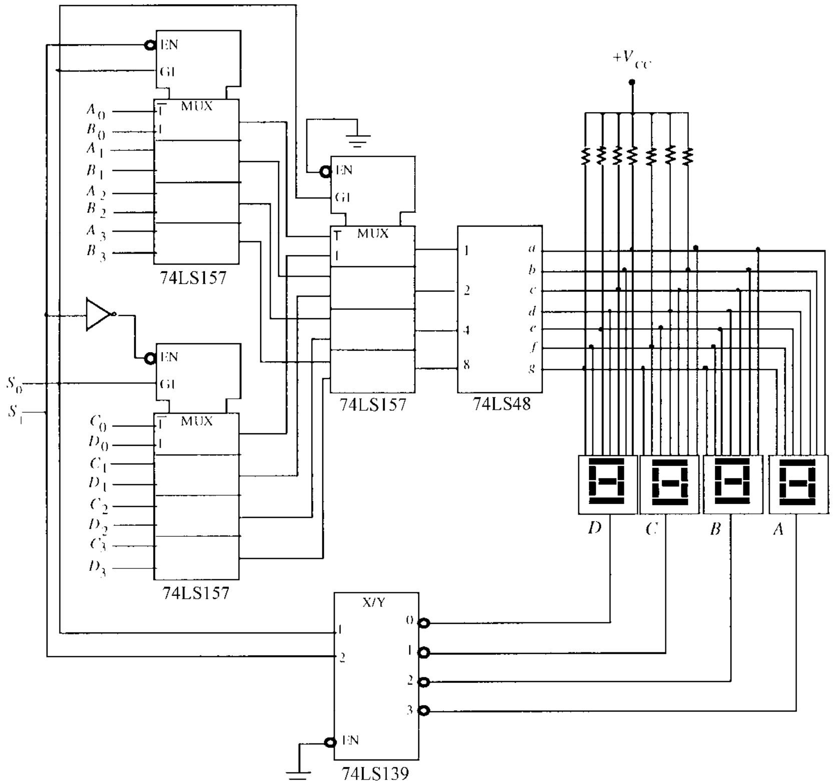


FIGURE 6-23

45. $\Sigma = \overline{A}\,\overline{B}C_{\text{in}} + \overline{A}B\overline{C}_{\text{in}} + A\overline{B}\,\overline{C}_{\text{in}} + ABC_{\text{in}}$

$C_{\text{out}} = ABC_{\text{in}} + \overline{A}BC_{\text{in}} + A\overline{B}C_{\text{in}} + AB\overline{C}_{\text{in}}$

See Figure 6-24.

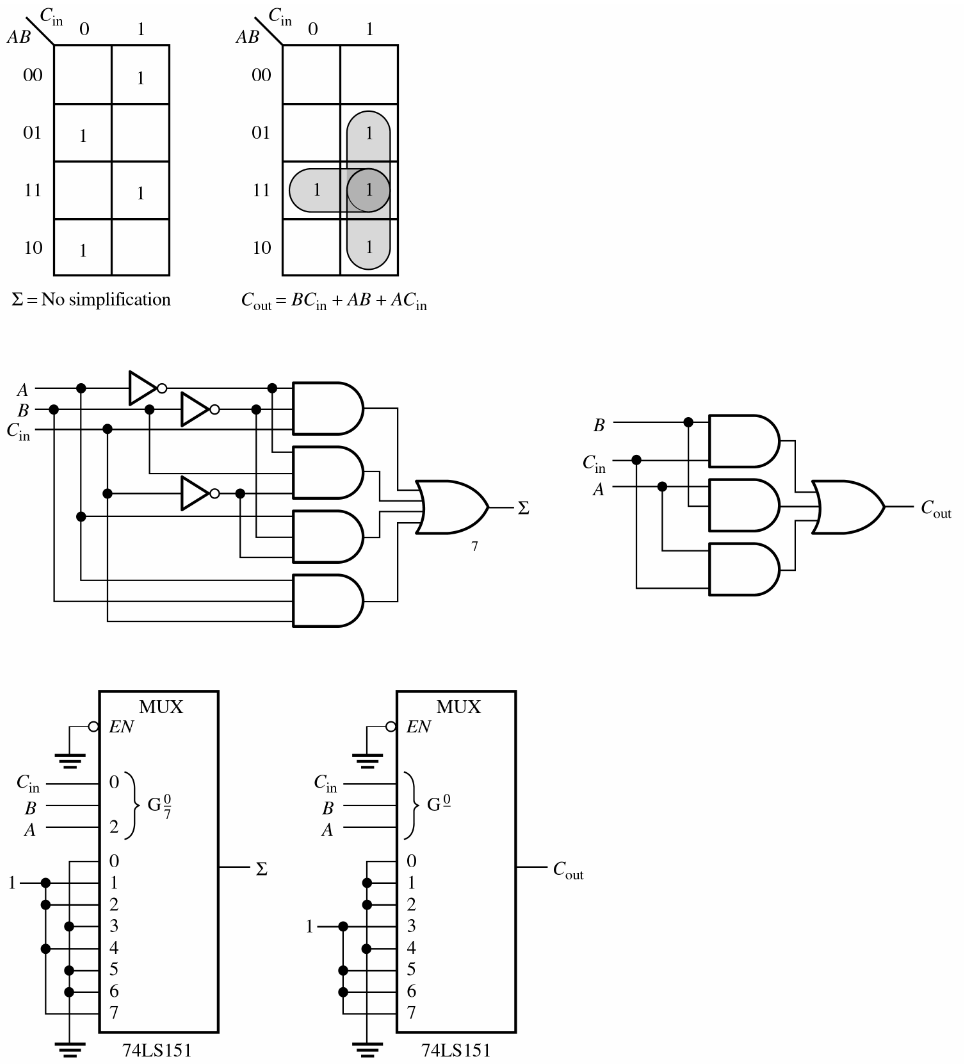


FIGURE 6-24

46. $$Y = \overline{A_3}\,\overline{A_2}A_1\overline{A_0} + \overline{A_3}\,\overline{A_2}A_1A_0 + \overline{A_3}A_2A_1\overline{A_0} + \overline{A_3}A_2A_1A_0 + A_3\overline{A_2}\,\overline{A_1}\,\overline{A_0}$$
$$+\ A_3\overline{A_2}A_1\overline{A_0} + A_3\overline{A_2}A_1A_0 + A_3A_2\overline{A_1}A_0 + A_3A_2A_1A_0$$

See Figure 6-25.

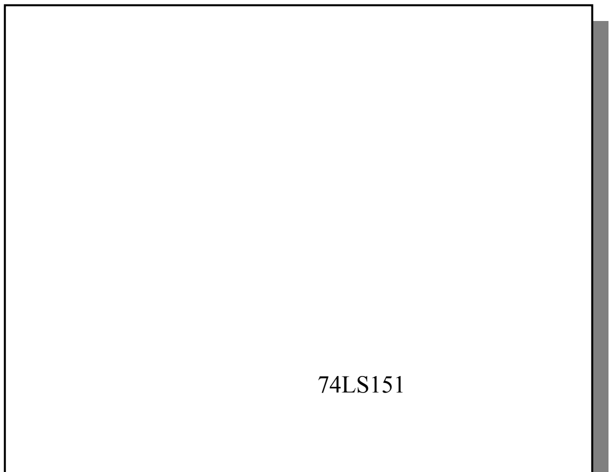


FIGURE 6-25

47. See Figure 6-26.

FIGURE 6-26

48. See Figure 6-27.

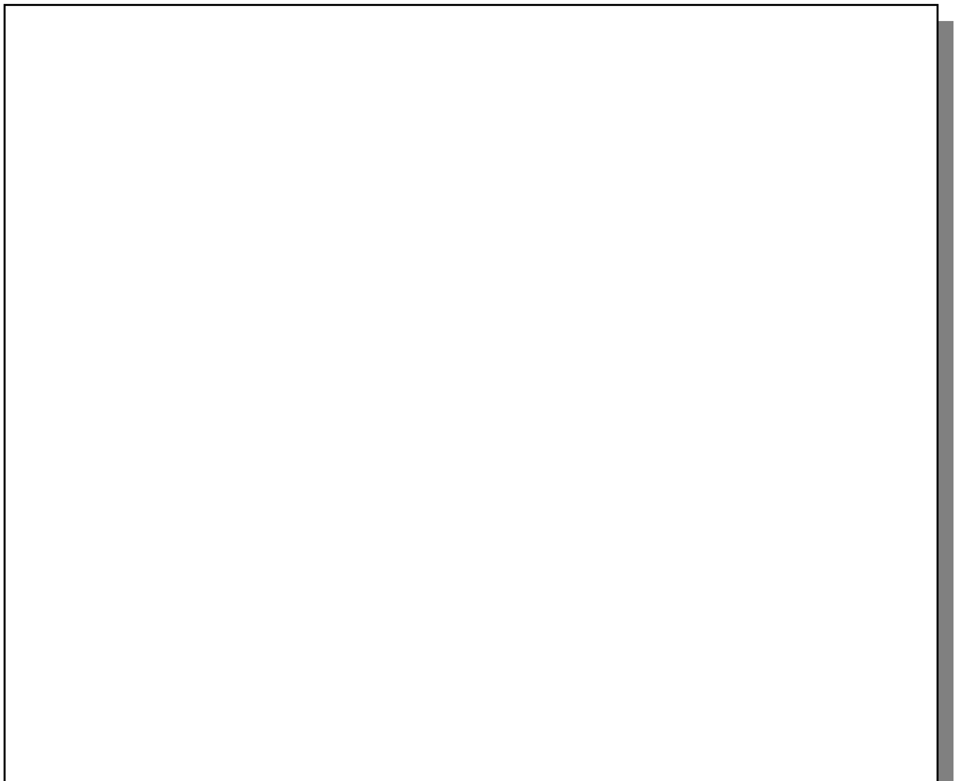


FIGURE 6-27

49. See Figure 6-28.

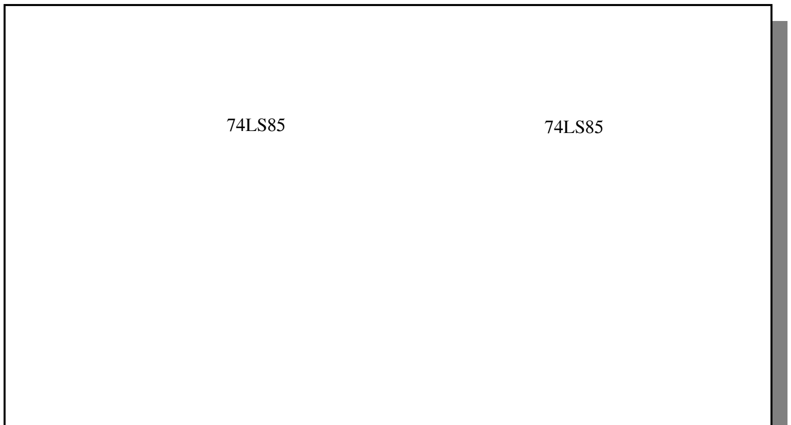


FIGURE 6-28

50. See Figure 6-29.

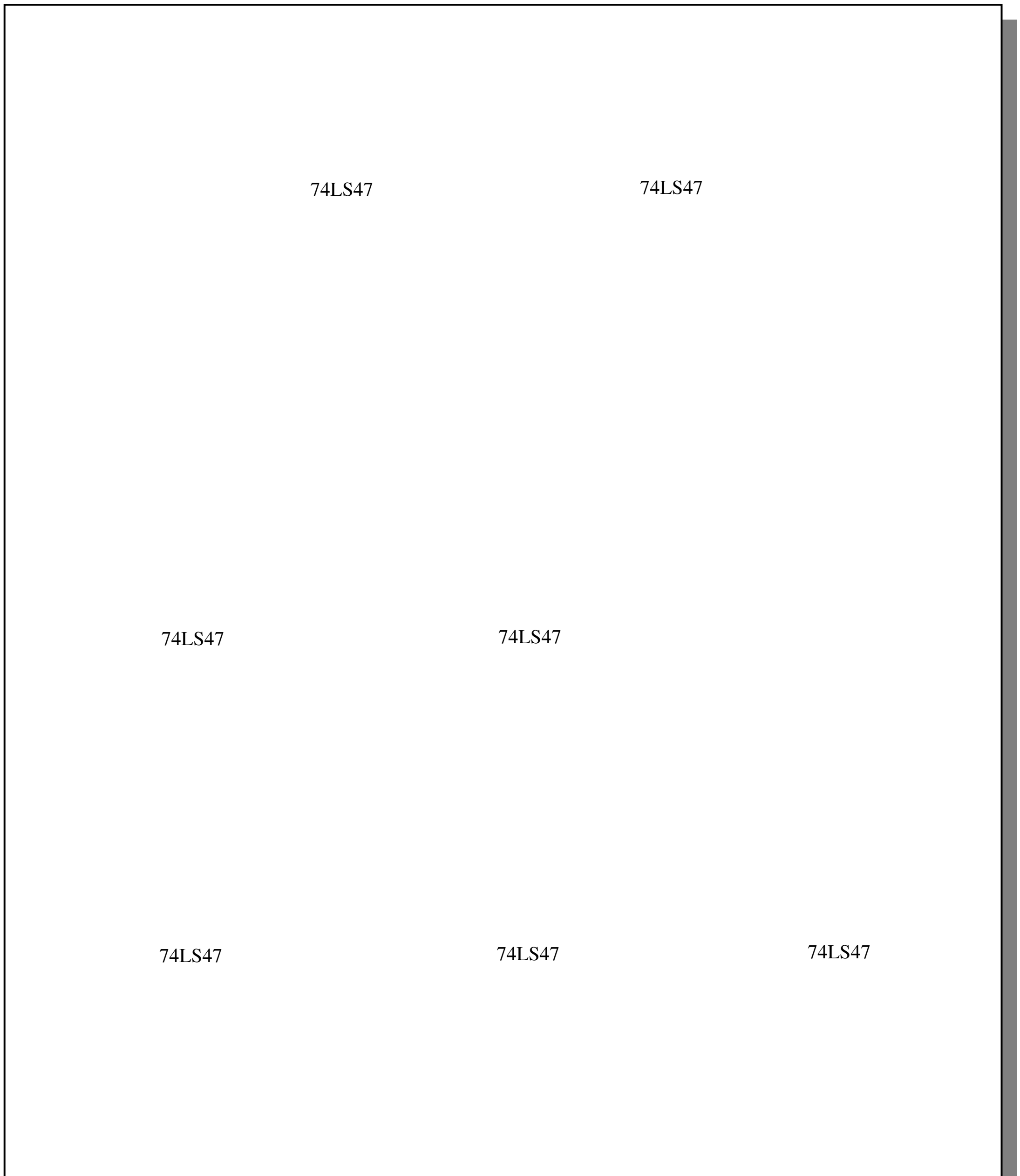


FIGURE 6-29

51. See Figure 6-30.

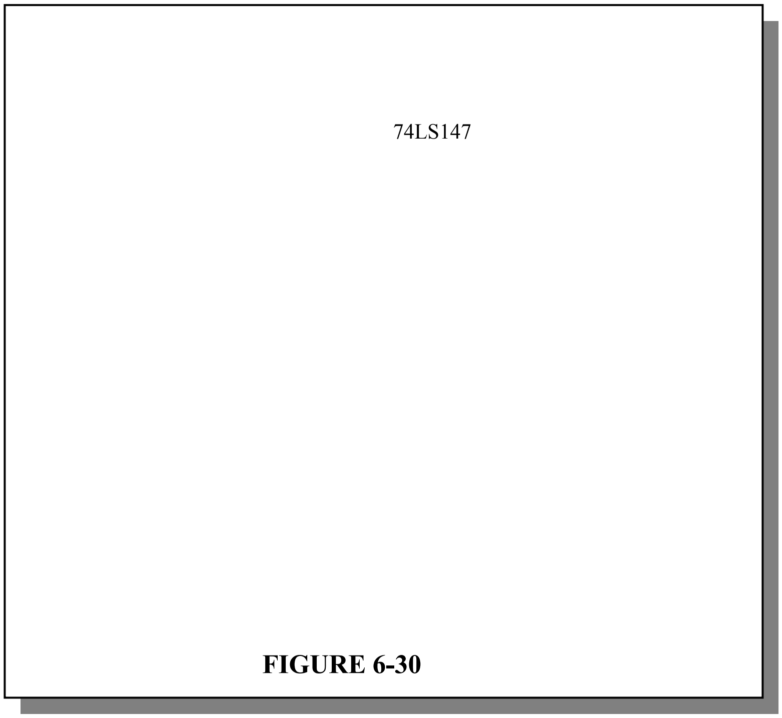


FIGURE 6-30

52. See Figure 6-31.

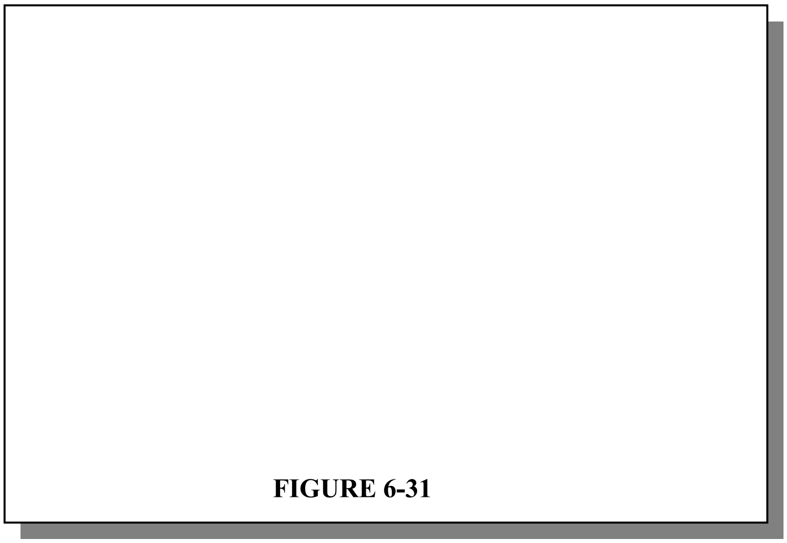


FIGURE 6-31

Multisim Troubleshooting Practice

53. LSB adder carry output open.

54. Pins 4 and 5 shorted together.

55. Pin 12 of upper 74148 open.

56. Pin 3 of upper 74151 open.

CHAPTER 7
LATCHES, FLIP-FLOPS, and TIMERS

Section 7-1 Latches

1. See Figure 7-1.

FIGURE 7-1

2. See Figure 7-2.

FIGURE 7-2

3. See Figure 7-3.

FIGURE 7-3

4. See Figure 7-4.

FIGURE 7-4

5. See Figure 7-5.

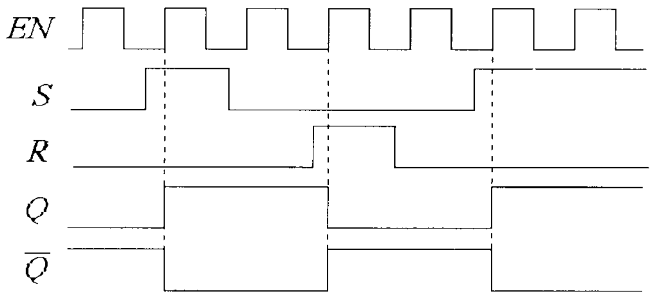


FIGURE 7-5

6. See Figure 7-6.

FIGURE 7-6

7. See Figure 7-7.

FIGURE 7-7

Section 7-2 Edge-Triggered Flip-Flops

8. See Figure 7-8.

FIGURE 7-8

9. See Figure 7-9.

FIGURE 7-9

10. See Figure 7-10.

FIGURE 7-10

11. See Figure 7-11.

FIGURE 7-11

12. See Figure 7-12.

FIGURE 7-12

13. See Figure 7-13.

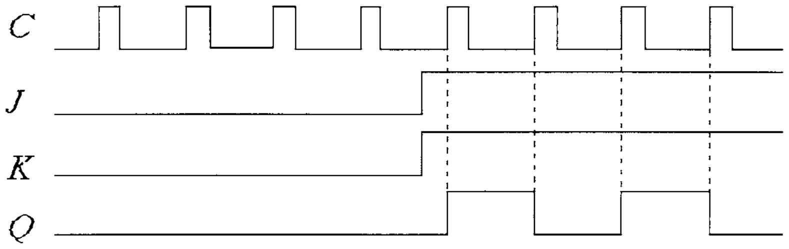


FIGURE 7-13

14. See Figure 7-14.

FIGURE 7-14

15. See Figure 7-15.

FIGURE 7-15

16. J: 0010000
K: 0000100
Q: 0011000

17. See Figure 7-16.

FIGURE 7-16

18. See Figure 7-17.

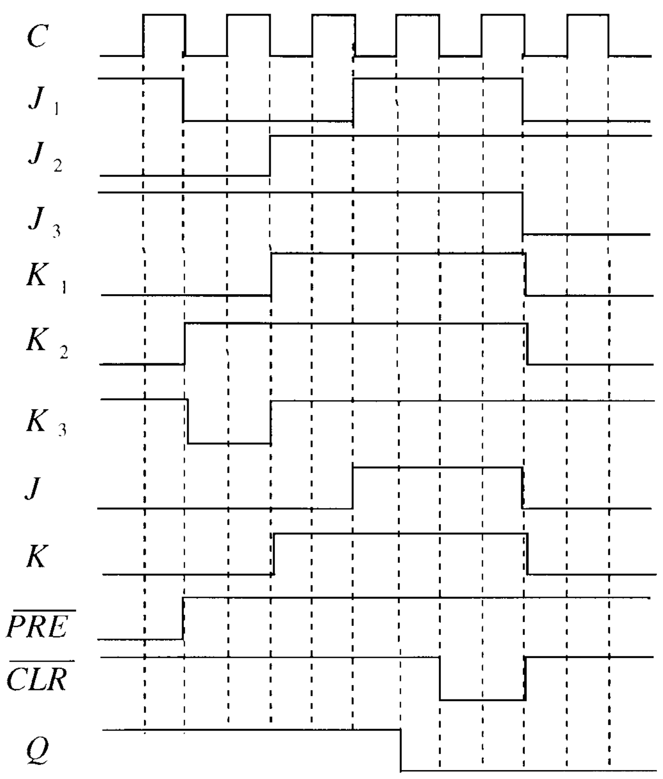


FIGURE 7-17

Section 7-3 Flip-Flop Operating Characteristics

19. The direct current and dc supply voltage

20. 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} ($\overline{\text{PRE}}$ to Q):
Time from assertion of the Preset input to the LOW-to-HIGH transition of the Q output.
t_{PHL} ($\overline{\text{CLR}}$ to Q):
Time from assertion of the clear input to the HIGH-to-LOW transition of the Q output.

21. T_{min} = 30 ns + 37 ns = 67 ns

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

22. See Figure 7-18.

FIGURE 7-18

FIGURE 7-19

23. I_T = 15(10 mA) = **150 mA**
P_T = (5 V)(150 mA) = **750 mW**

24. See Figure 7-19.

Section 7-4 Flip-Flop Applications

25. See Figure 7-20.

FIGURE 7-20

26. See Figure 7-21.

FIGURE 7-21

Section 7-5 One-Shots

27. $t_W = 0.7RC_{\text{EXT}}$ = 0.7(3.3 kΩ)(2000 pF) = **4.62 μs**

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

29. See Figure 7-22.

FIGURE 7-22

Section 7-6 Astable Multivibrator

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

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

For a duty cycle of 75%:

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

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

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

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

Section 7-7 Troubleshooting

32. The flip-flop in Figure 7-94 of the text has an internally open *J* input.

33. The wire from pin 6 to pin 10 and the ground wire are reversed. Pin 7 should be at ground and pin 6 connected to pin 10.

34. See Figure 7-23.

FIGURE 7-23

35. 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 ($\overline{\text{CLR}}$) 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 $\overline{\text{CLR}}$ 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 $\overline{\text{CLR}}$ at power on.

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

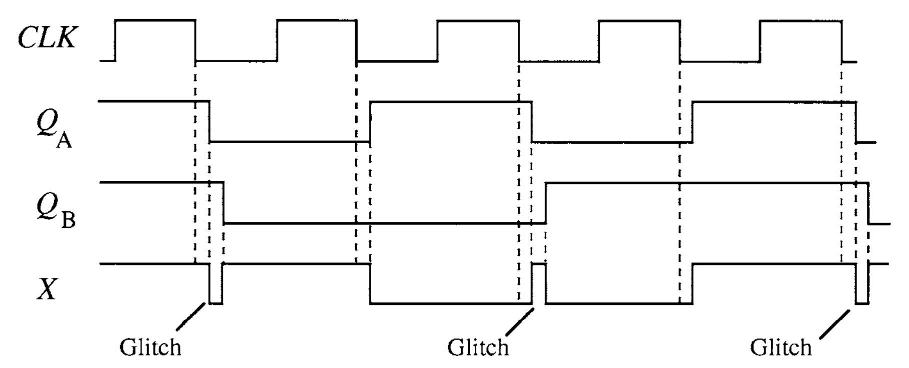


FIGURE 7-24

37. (a) See Figure 7-25.

FIGURE 7-25

(b) K_B open acts as a HIGH and the operation is normal. The timing diagram is the same as Figure 7-25.

(c) See Figure 7-26.

FIGURE 7-26

(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 7-27.

FIGURE 7-27

38. $t_W = 0.7RC_{\text{EXT}}$

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

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

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

FIGURE 7-28

System Application Activity

39. For the 6 s timer let $C_1 = 1\ \mu\text{F}$

$$R_1 = \frac{6\text{ s}}{(1.1)(1\ \mu\text{F})} = 5.5\ \text{M}\Omega \text{ (use } 5.6\ \text{M}\Omega)$$

For the 40 s timer let $C_1 = 2.2\ \mu\text{F}$

$$R_1 = \frac{40\text{ s}}{(1.1)(2.2\ \mu\text{F})} = 16.5\ \text{M}\Omega \ \text{ (use } 15\ \text{M}\Omega)$$

See Figure 7-29.

6 s timer and 40 s timer are the same except for the component values calculated above.

FIGURE 7-29

40. $t_W = 6$ s. Let $C_{\text{EXT}} = 1$ μF.

$t_{\text{W}} = 0.7RC_{\text{EXT}}$

$$R = \frac{t_W}{0.7C_{\text{EXT}}} = \frac{6\text{ s}}{0.7(1\ \mu\text{F})} = 8.6\ \text{M}\Omega$$

$t_W = 40$ s. Let $C_{\text{EXT}} = 10$ μF.

$$R = \frac{t_W}{0.7C_{\text{EXT}}} = \frac{40\text{ s}}{0.7(10\ \mu\text{F})} = \ = 5.7\ \text{M}\Omega$$

See Figure 7-30.

FIGURE 7-30

41. $t_W = 6$ s. Let $C_{\text{EXT}} = 1$ μF.

$$t_W = 0.32R_{\text{EXT}}C_{\text{EXT}}\left(1 + \frac{0.7}{R_{\text{EXT}}}\right) = 0.32R_{\text{EXT}}C_{\text{EXT}} + (0.7)(0.32)C_{\text{EXT}}$$

$$R_{\text{EXT}} = \frac{t_W - (0.7)(0.32)C_{\text{EXT}}}{0.32C_{\text{EXT}}} = \frac{6\text{ s} - (0.224)(1\ \mu\text{F})}{0.32\ (1\ \mu\text{F})} = 18.8\ \text{M}\Omega$$

$t_W = 40$ s. Let $C_{\text{EXT}} = 10$ μF.

$$R_{\text{EXT}} = \frac{40\text{ s} - (0.224)10\ \mu\text{F}}{0.32(10\ \mu\text{F})} = \ = 12.5\ \text{M}\Omega$$

See Figure 7-31.

FIGURE 7-31

Special Design Problems

42. See Figure 7-32.

FIGURE 7-32

43. See Figure 7-33 for one possibility.

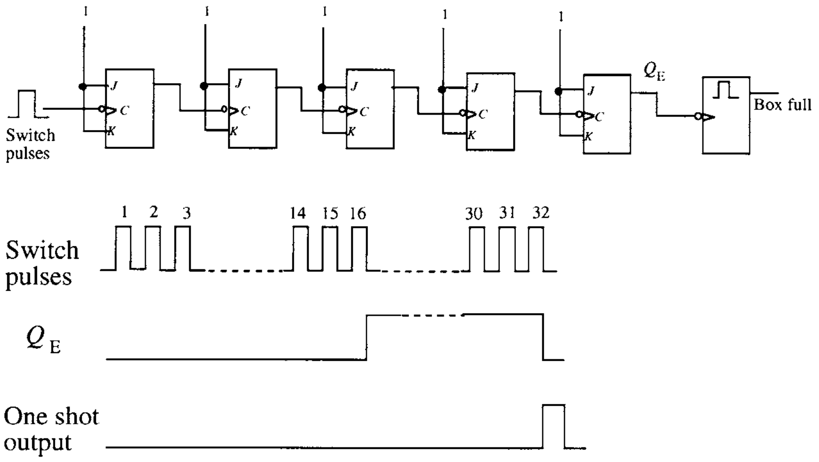


FIGURE 7-33

44. 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 7-34.

FIGURE 7-34

Multisim Troubleshooting Practice

45. $\overline{Q}$ output of U1 open.

46. *K* input of U2 open.

47. $\overline{SET}$ input of U1 open.

48. No fault.

49. *K* input of U2 open.

CHAPTER 8
COUNTERS

Section 8-1 Asynchronous Counters

1. See Figure 8-1.

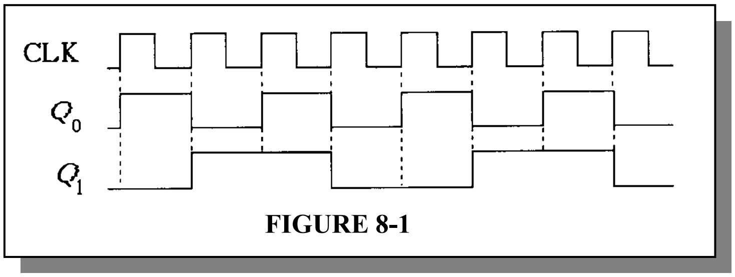


FIGURE 8-1

2. See Figure 8-2.

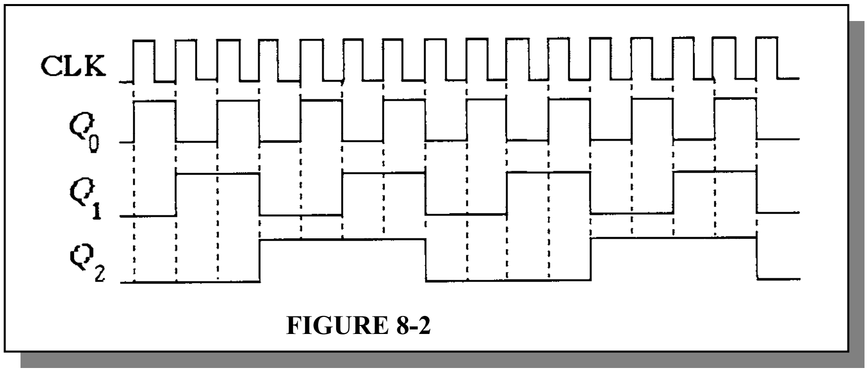


FIGURE 8-2

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

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

4. See Figure 8-3.

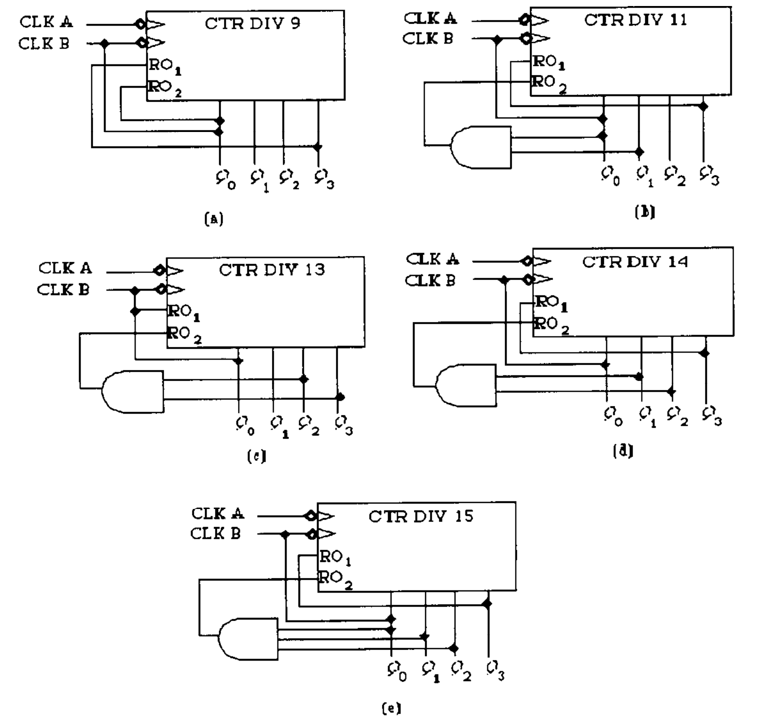


FIGURE 8-3

Section 8-2 Synchronous Counters

5. **8 ns**, the time it takes one flip-flop to change state.

6. See Figure 8-4.

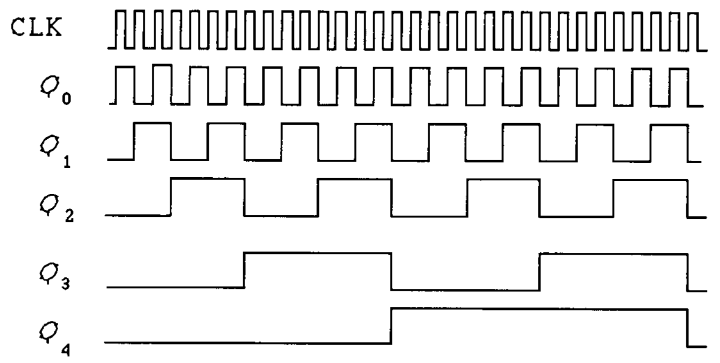


FIGURE 8-4

7. 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

8. See Figure 8-5.

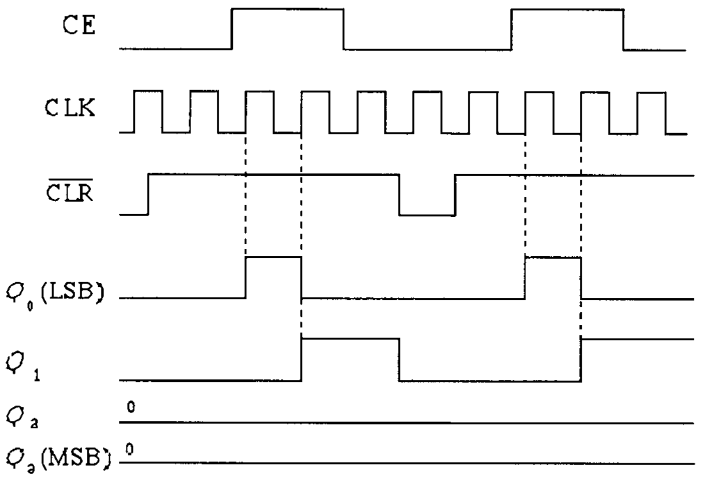


FIGURE 8-5

9. See Figure 8-6.

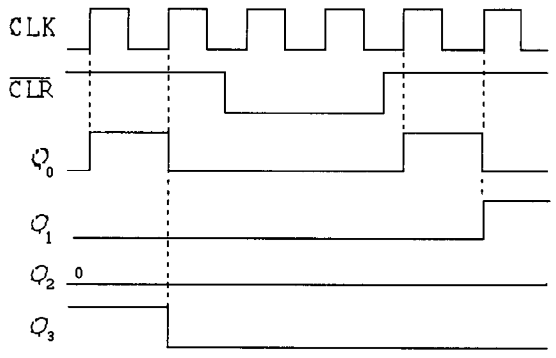


FIGURE 8-6

10. See Figure 8-7.

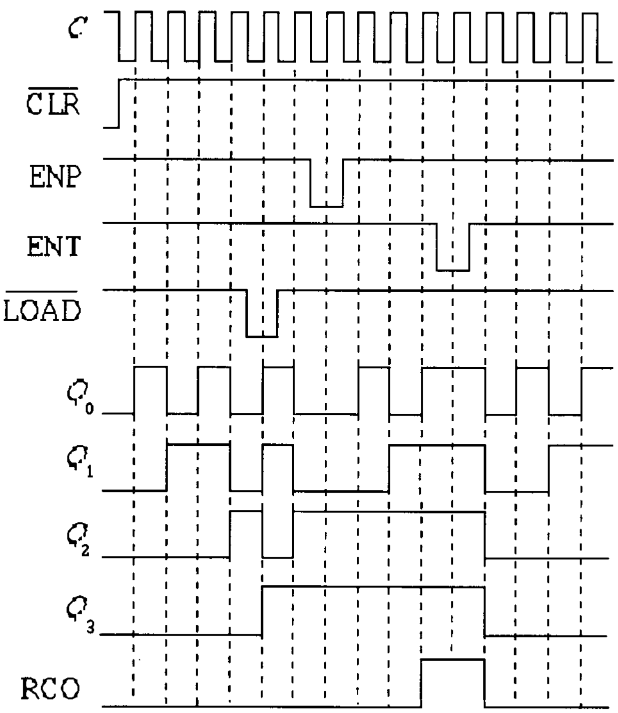


FIGURE 8-7

11. See Figure 8-8.

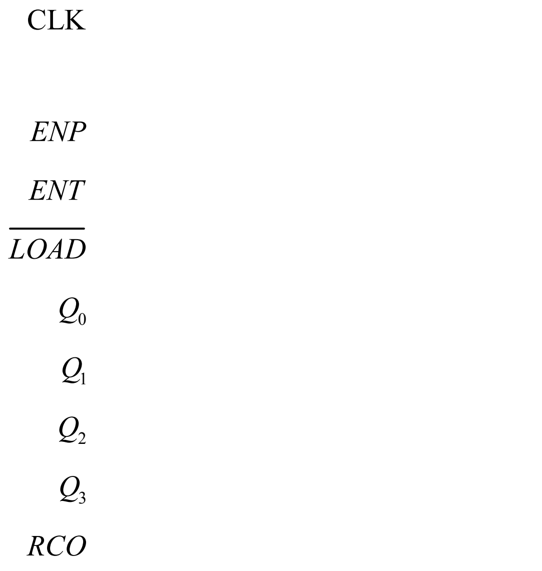


FIGURE 8-8

Section 8-3 Up/Down Synchronous Counters

12. See Figure 8-9.

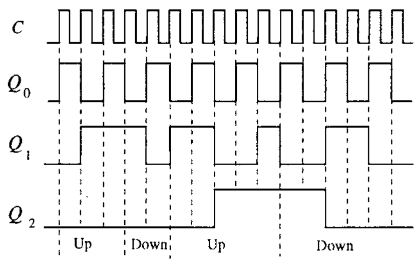


FIGURE 8-9

13. See Figure 8-10.

FIGURE 8-10

14. See Figure 8-11.

FIGURE 8-11

15. See Figure 8-12.
(*NOTE:* The text answer, Figure P-64, is incorrect in the first printing. It will be corrected to match Figure 8-12 in the 2nd printing.)

FIGURE 8-12

Section 8-4 Design of Synchronous Counters

16.

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

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

17.

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

Tog = toggle, NC = no change

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

18. NEXT-STATE TABLE

Present State		**Next State**	
Q_1	Q_0	Q_1	Q_0
0	0	1	0
1	0	0	1
0	1	1	1
1	1	0	0

TRANSITION TABLE

Output State Transitions (Present state to next state)		**Flip-Flop Inputs**			
Q_1	Q_0	J_1	K_1	J_0	K_0
0 to 1	0 to 0	1	X	0	X
1 to 0	0 to 1	X	1	1	X
0 to 1	1 to 1	1	X	X	0
1 to 0	1 to 0	X	1	X	1

See Figure 8-13.

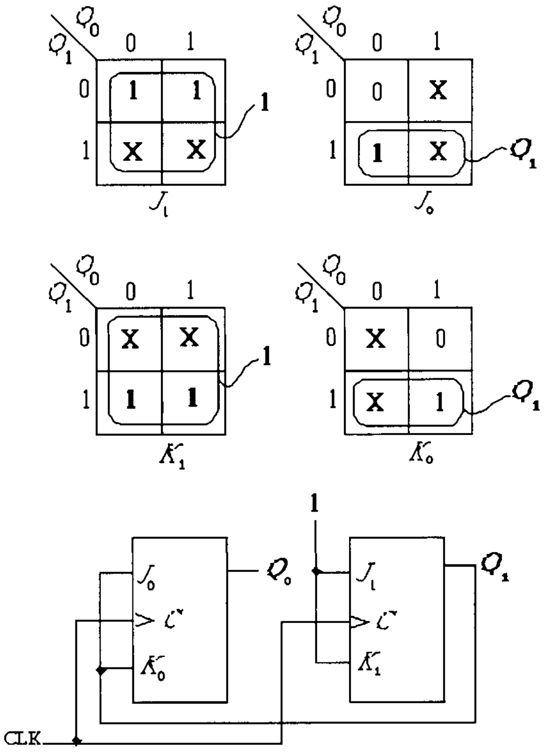


FIGURE 8-13

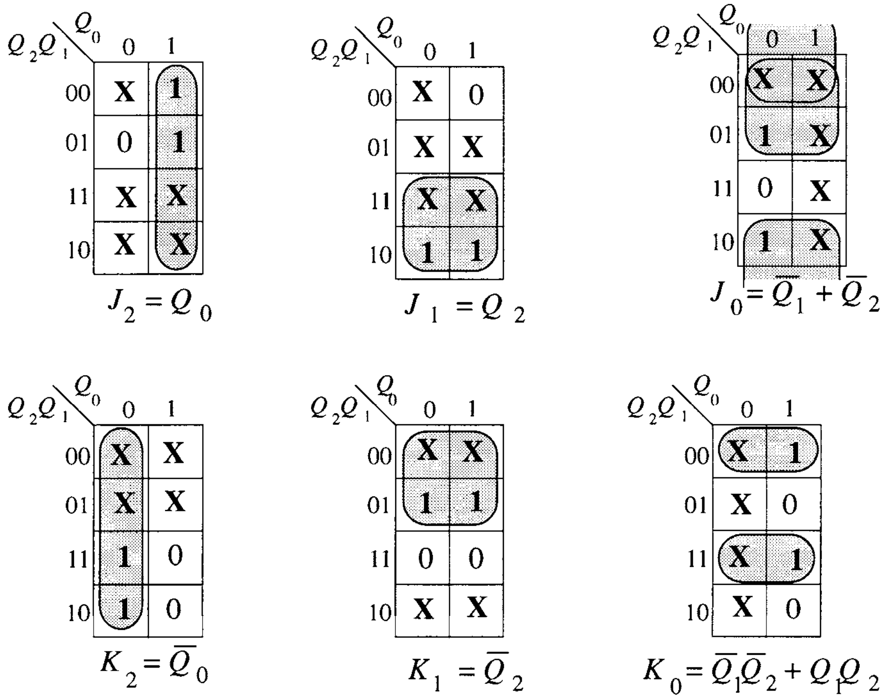
19. NEXT-STATE TABLE

Present State			**Next State**		
Q_2	Q_1	Q_0	Q_2	Q_1	Q_0
0	0	1	1	0	0
1	0	0	0	1	1
0	1	1	1	0	1
1	0	1	1	1	1
1	1	1	1	1	0
1	1	0	0	1	0
0	1	0	0	0	1

TRANSITION TABLE

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

See Figure 8-14.



$Q_2Q_1 \backslash Q_0$	0	1
00	X	1
01	0	1
11	X	X
10	X	X

$J_2 = Q_0$

$Q_2Q_1 \backslash Q_0$	0	1
00	X	0
01	X	X
11	X	X
10	1	1

$J_1 = Q_2$

$Q_2Q_1 \backslash Q_0$	0	1
00	X	X
01	1	X
11	0	X
10	1	X

$J_0 = \overline{Q}_1 + \overline{Q}_2$

$Q_2Q_1 \backslash Q_0$	0	1
00	X	X
01	X	X
11	1	0
10	1	0

$K_2 = \overline{Q}_0$

$Q_2Q_1 \backslash Q_0$	0	1
00	X	X
01	1	1
11	0	0
10	X	X

$K_1 = \overline{Q}_2$

$Q_2Q_1 \backslash Q_0$	0	1
00	X	1
01	X	0
11	X	1
10	X	0

$K_0 = \overline{Q}_1\overline{Q}_2 + Q_1Q_2$

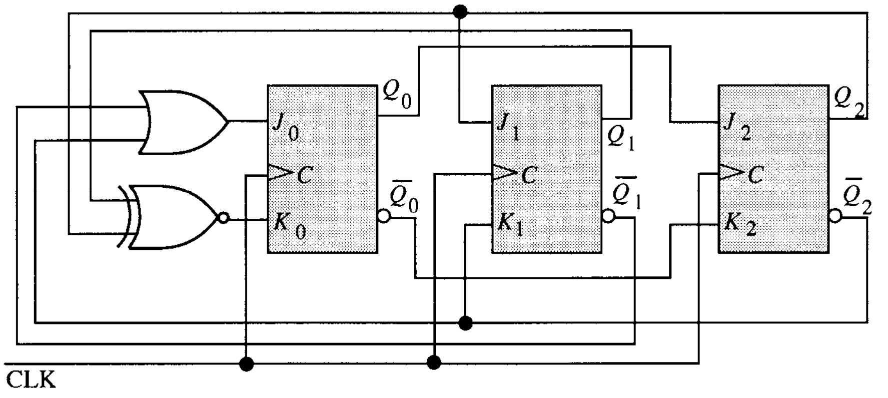


FIGURE 8-14

20. NEXT-STATE TABLE

Present State				**Next State**			
Q_3	Q_2	Q_1	Q_0	Q_3	Q_2	Q_1	Q_0
0	0	0	0	1	0	0	1
1	0	0	1	0	0	0	1
0	0	0	1	1	0	0	0
1	0	0	0	0	0	1	0
0	0	1	0	0	1	1	1
0	1	1	1	0	0	1	1
0	0	1	1	0	1	1	0
0	1	1	0	0	1	0	0
0	1	0	0	0	1	0	1
0	1	0	1	0	0	0	0

TRANSITION TABLE

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

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

See Figure 8-15. Counter implementation is straightforward from input expressions.

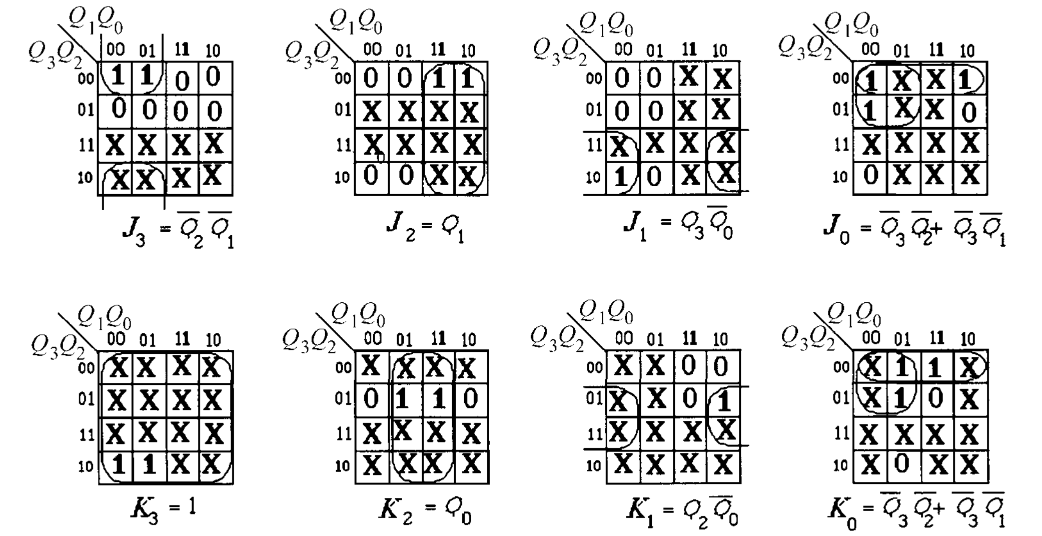

Q_1Q_0
Q_3Q_2
00 01 11 10
$J_3 = \overline{Q}_2\overline{Q}_1$
$J_2 = Q_1$
$J_1 = Q_3\overline{Q}_0$
$J_0 = \overline{Q}_3\overline{Q}_2 + \overline{Q}_3\overline{Q}_1$
$K_3 = 1$
$K_2 = Q_0$
$K_1 = Q_2\overline{Q}_0$
$K_0 = \overline{Q}_3\overline{Q}_2 + \overline{Q}_3\overline{Q}_1$

FIGURE 8-15

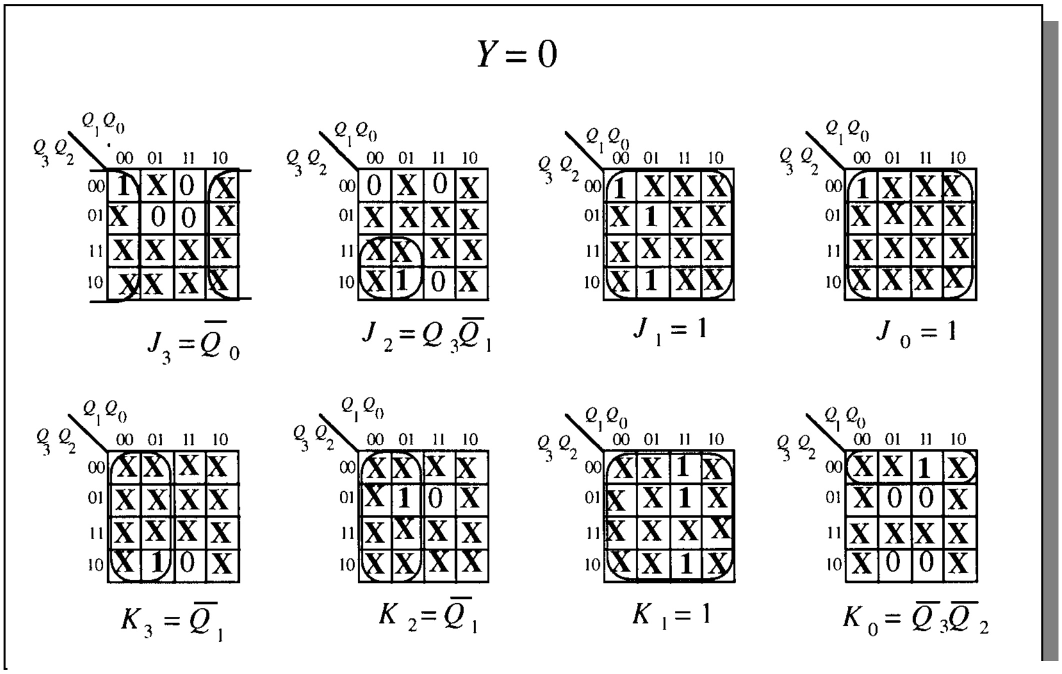
21. NEXT-STATE TABLE

Present State				**Next State**							
				$Y = 1$ (Up)				$Y = 0$ (Down)			
Q_3	Q_2	Q_1	Q_0	Q_3	Q_2	Q_1	Q_0	Q_3	Q_2	Q_1	Q_0
0	0	0	0	0	0	1	1	1	0	1	1
0	0	1	1	0	1	0	1	0	0	0	0
0	1	0	1	0	1	1	1	0	0	1	1
0	1	1	1	1	0	0	1	0	1	0	1
1	0	0	1	1	0	1	1	0	1	1	1
1	0	1	1	0	0	0	0	1	0	0	1

TRANSITION TABLE

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

See Figure 8-16.



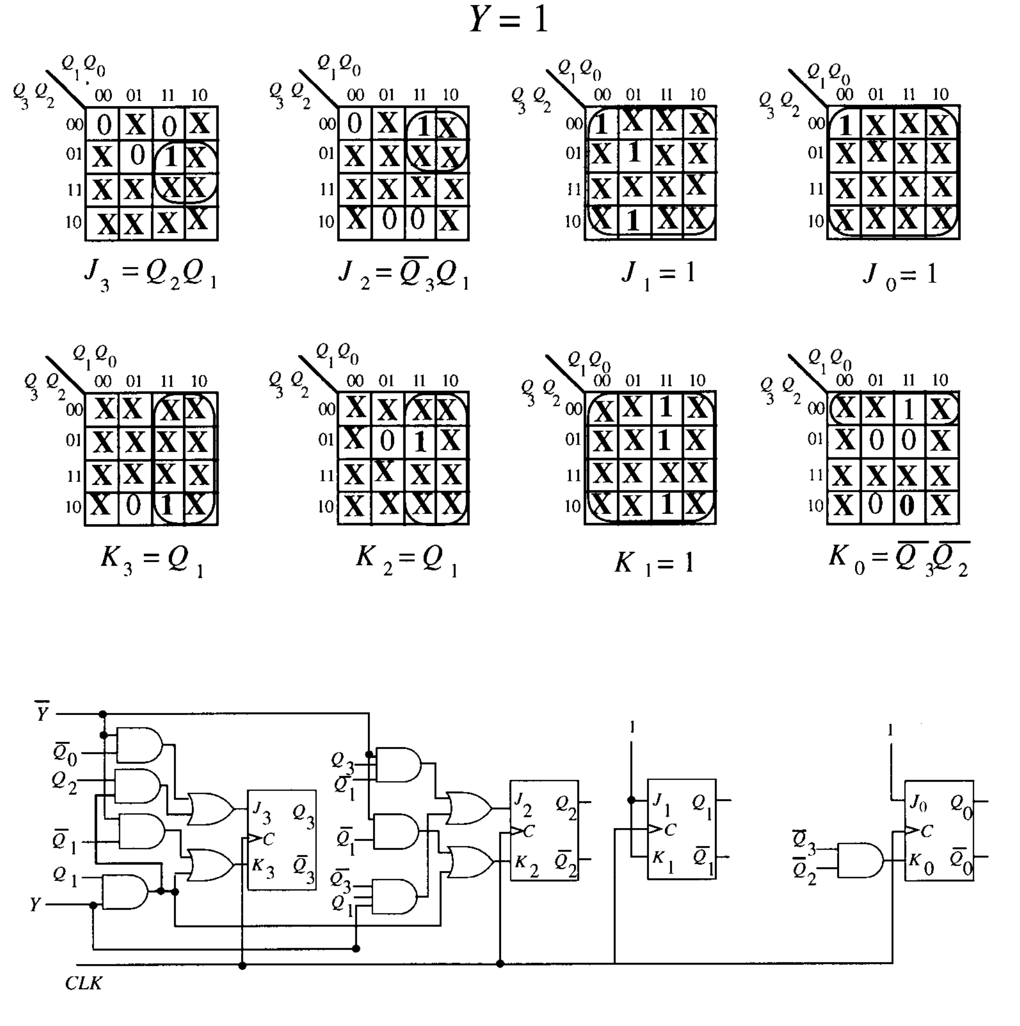


FIGURE 8-16

Section 8-5 Cascaded Counters

22. (a) Modulus = 4 × 8 × 2 = **64**

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

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

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

(b) Modulus = 10 × 10 × 10 × 2 = **2000**

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

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

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

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

(c) Modulus = 3 × 6 × 8 × 10 × 10 = **14400**

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

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

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

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

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

(d) Modulus = 2 × 4 × 6 × 8 × 16 = **6144**

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

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

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

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

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

23. See Figure 8-17.

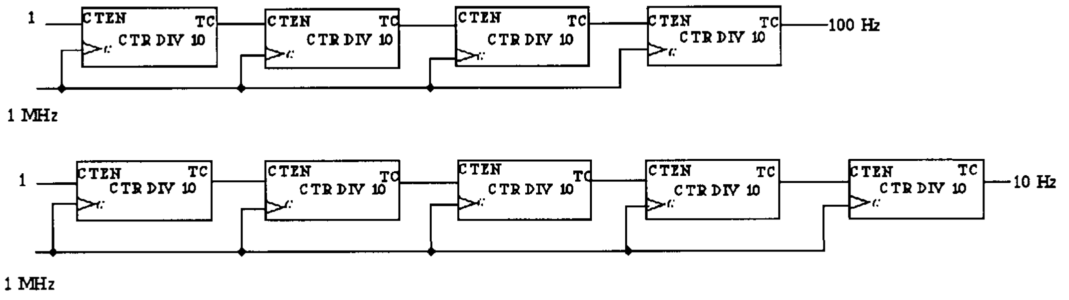


FIGURE 8-17

24. See Figure 8-18.

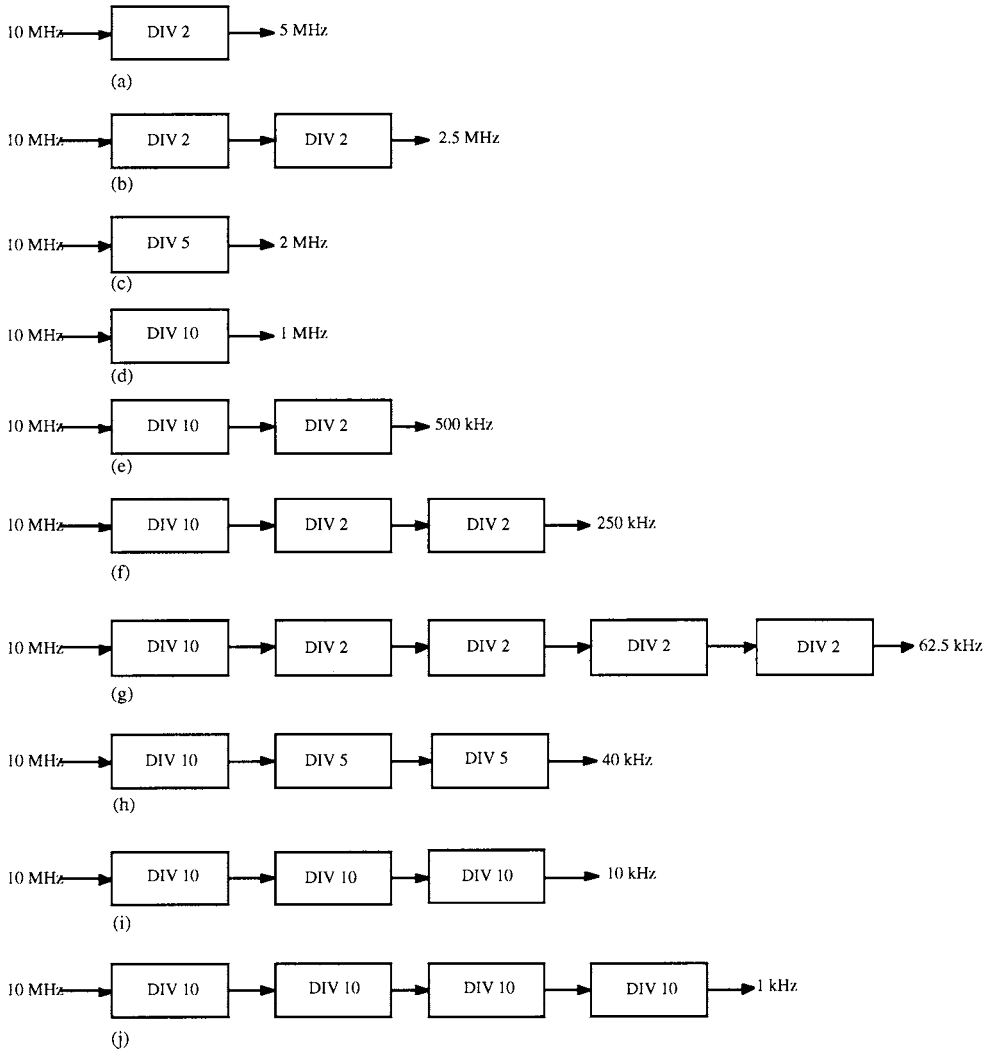


FIGURE 8-18

Section 8-6 Counter Decoding

25. See Figure 8-19.

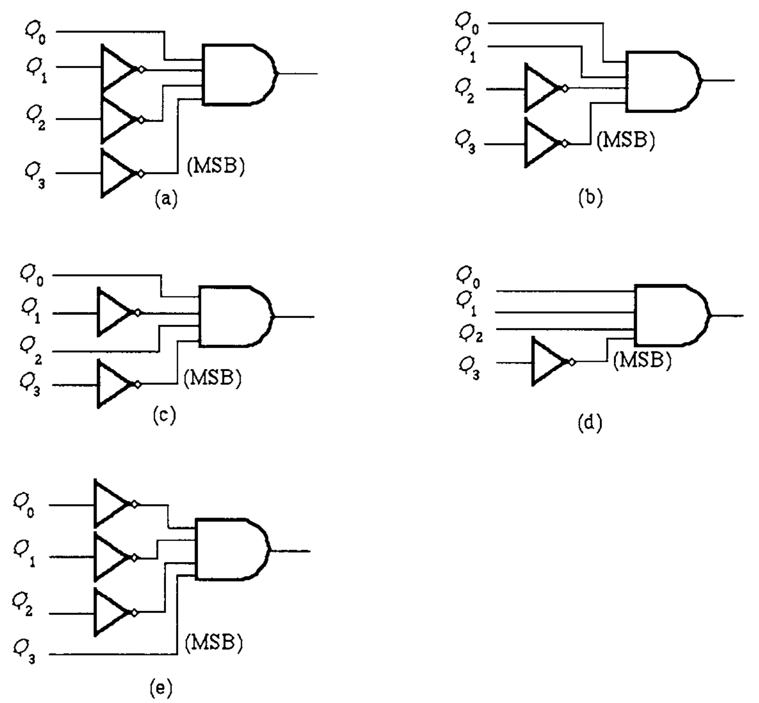


FIGURE 8-19

26. See Figure 8-20.

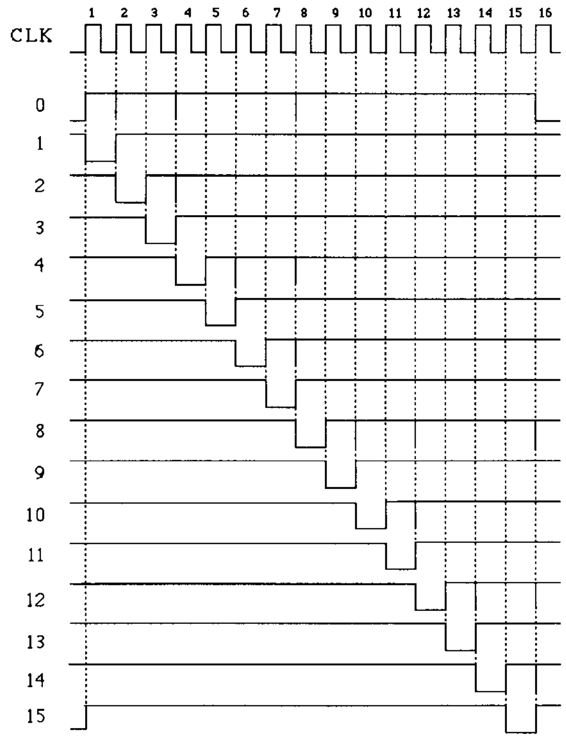


FIGURE 8-20

27. 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 8-20 (Problem 8-26) by short vertical lines.

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

28. See Figure 8-21.

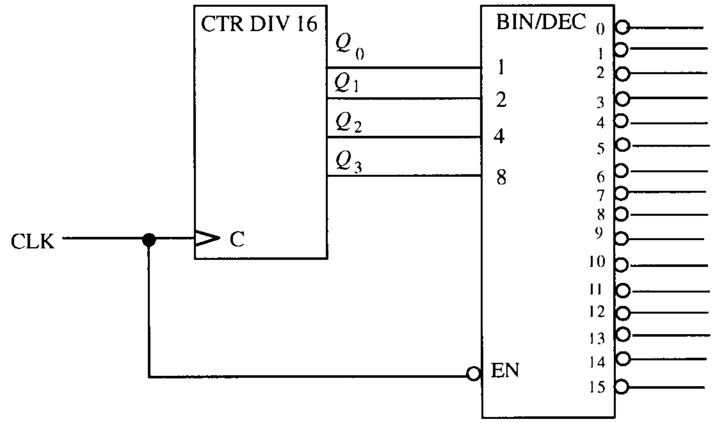


FIGURE 8-21

29. See Figure 8-22.

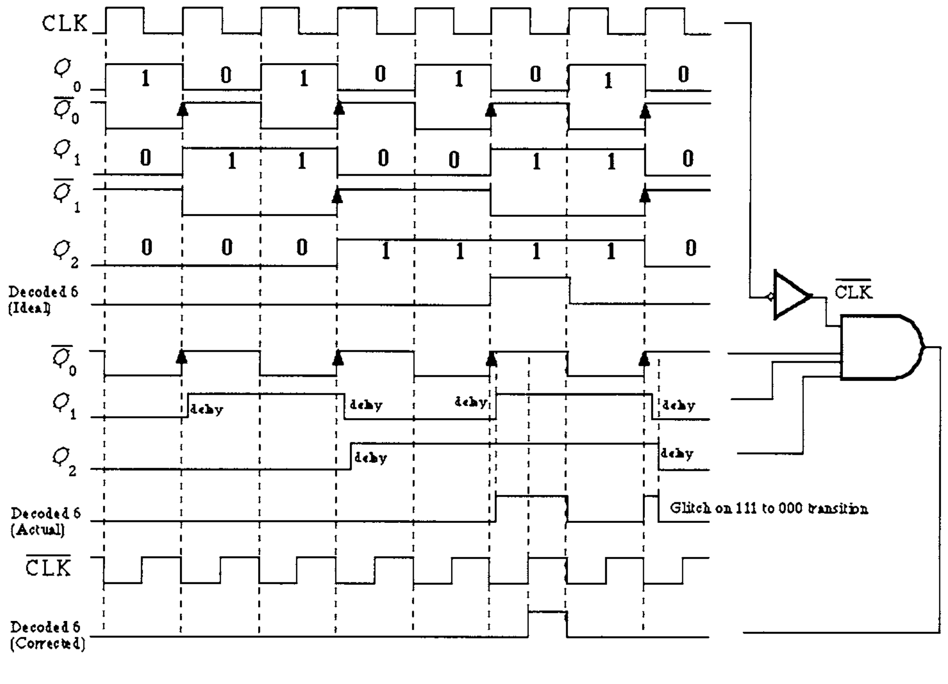


FIGURE 8-22

30. 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 8-23 which is expanded to show the delays.

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

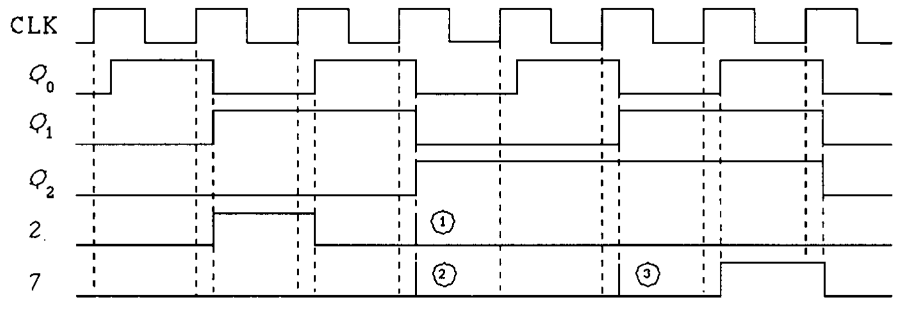


FIGURE 8-23

Section 8-7 Counter Applications

31. For the digital clock in Figure 8-49 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**

32. 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:

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

33. 53 + 37 – 22 = **68**

34. See Figure 8-24.

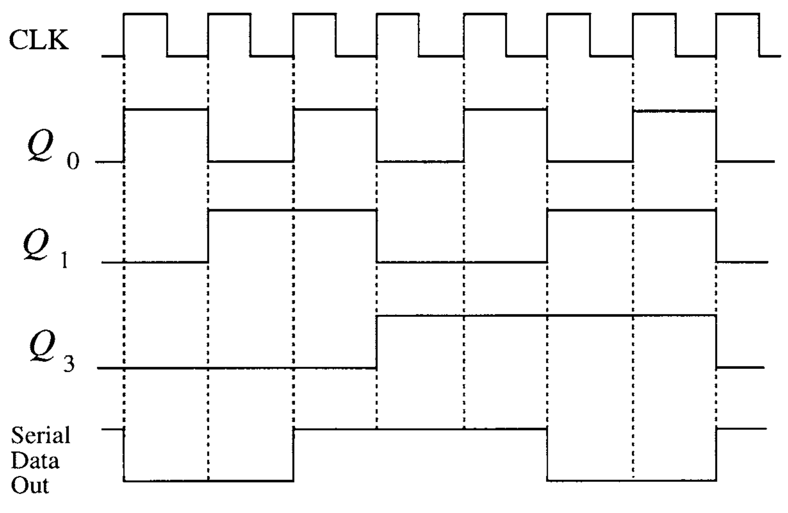


FIGURE 8-24

Section 8-9 Troubleshooting

35. (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 8-25.

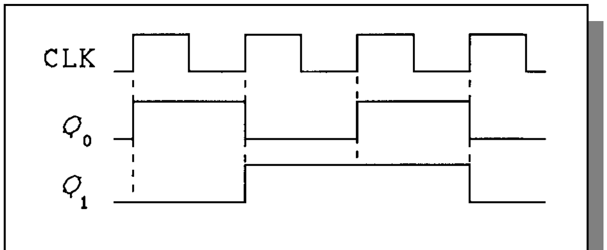


FIGURE 8-25

(c) See Figure 8-26.

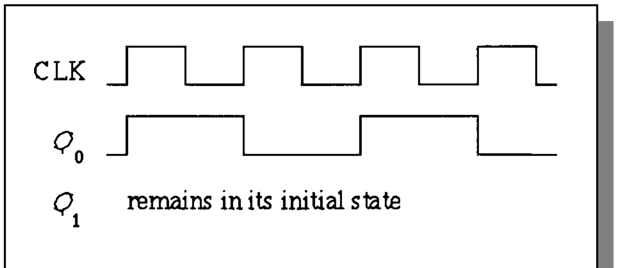


FIGURE 8-26

(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.

36. (a) Q_0 and Q_1 will not change from initial states.

(b) See Figure 8-27.

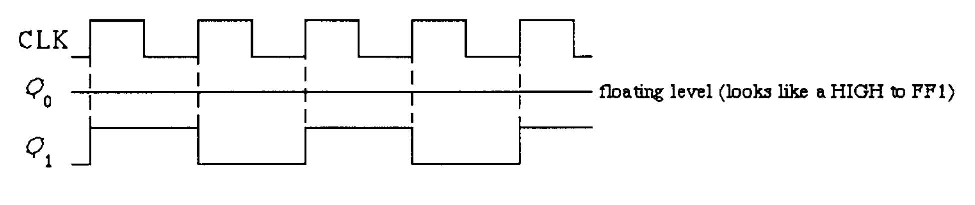


FIGURE 8-27

(c) See Figure 8-28.

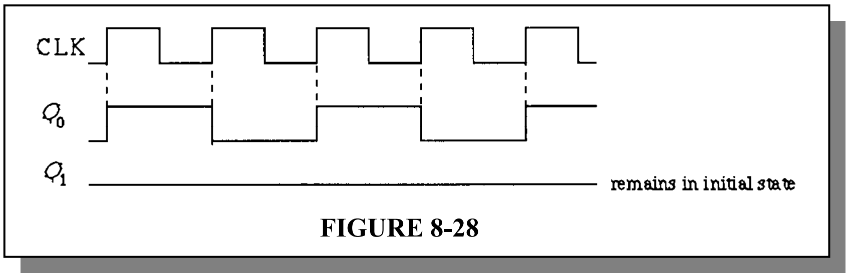


FIGURE 8-28

(d) Normal operation. See Figure 8-29.

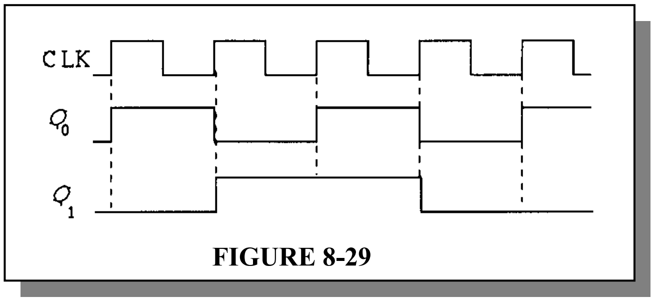


FIGURE 8-29

(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 8-30.

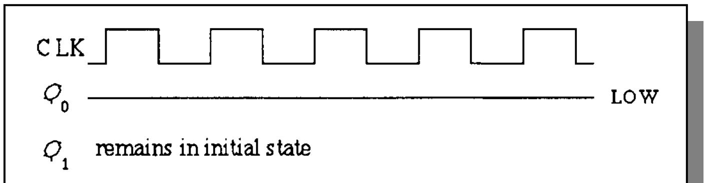


FIGURE 8-30

37. First, determine the correct waveforms and observe that Q_0 is correct but Q_1 and Q_2 are incorrect in Figure 8-83 in the text. See Figure 8-31.

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.

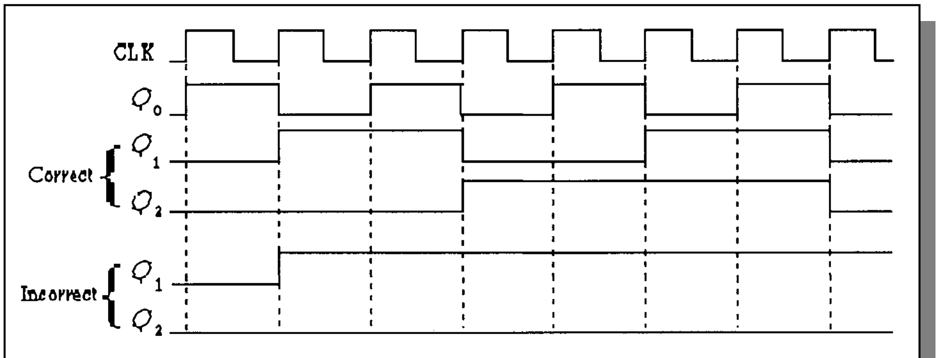


FIGURE 8-31

38. 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*.

39. If the Q_0 input to the AND gate is *open*, the *JK* inputs to FF2 are as shown in Figure 8-32.

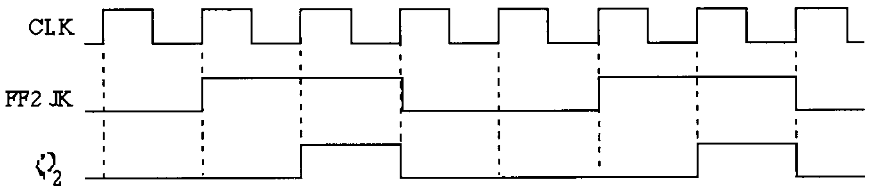


FIGURE 8-32

40. Number of states = 40,000

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

76.2939 Hz is not correct. The faulty division factor is

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

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

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

41.

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

42.
- The flip-flop output is stuck HIGH or open.
- The least significant BCD/7-segment input is open.

See Figure 8-33.

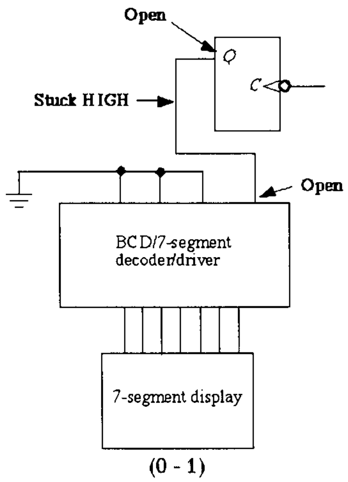


FIGURE 8-33

43. 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.

44. 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.

System Application Activity

45. The expressions for the D_0 and the D_1 flip-flop inputs in the sequential logic portion of the system were developed in the System Application Activity. Figure 8-34 shows the NAND implementation.

$$D_0 = \overline{Q}_1 Q_0 + \overline{Q}_1 \overline{T}_L V_S + Q_0 T_L V_S$$

$$D_1 = Q_0 \overline{T}_L + Q_1 T_S$$

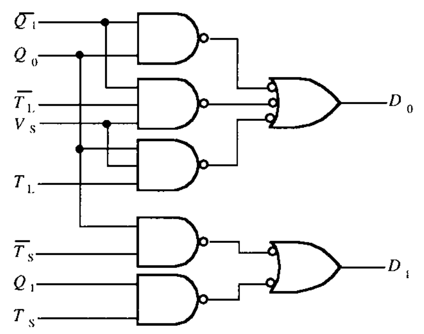


FIGURE 8-34

46. See Figure 8-35.

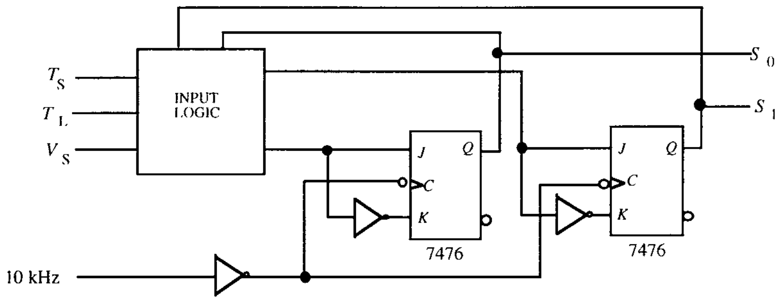


FIGURE 8-35

47. The time interval for the green light can be increased from 25 s to 60 s by increasing the value of either the resistor or the capacitor value by

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

Special Design Problems

48. See Figure 8-36.

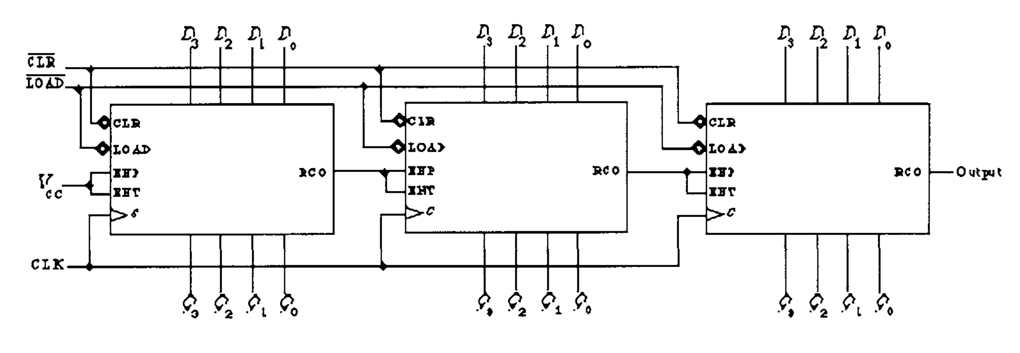


FIGURE 8-36

49. 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 8-37.

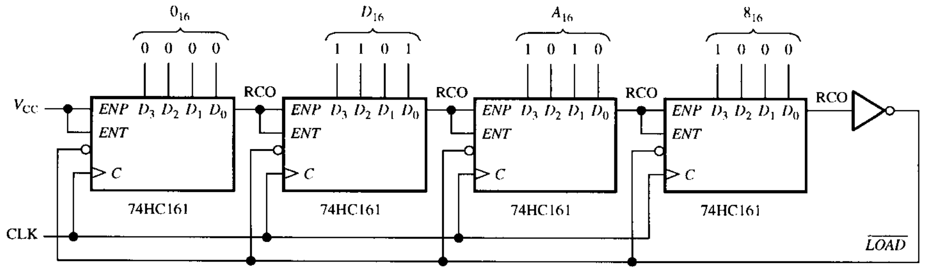


FIGURE 8-37

50. 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 8-38.

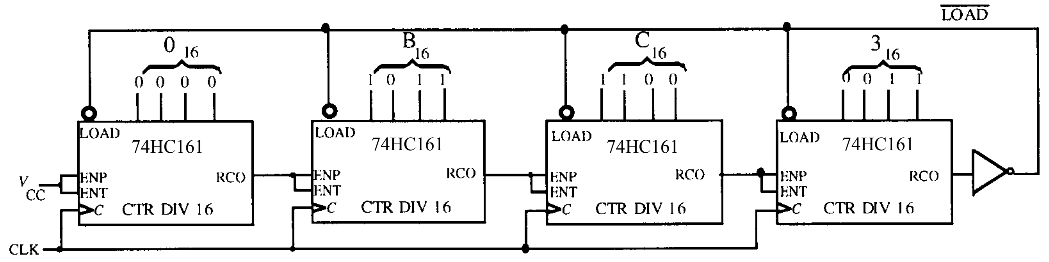


FIGURE 8-38

51. 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 8-39.

FIGURE 8-39

52. See Figure 8-40.

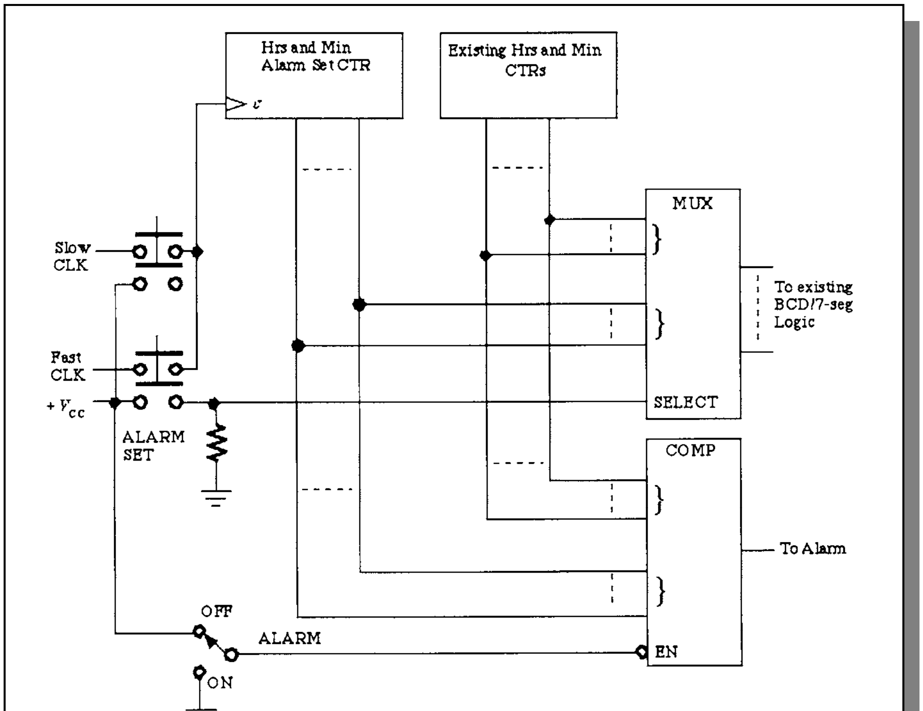
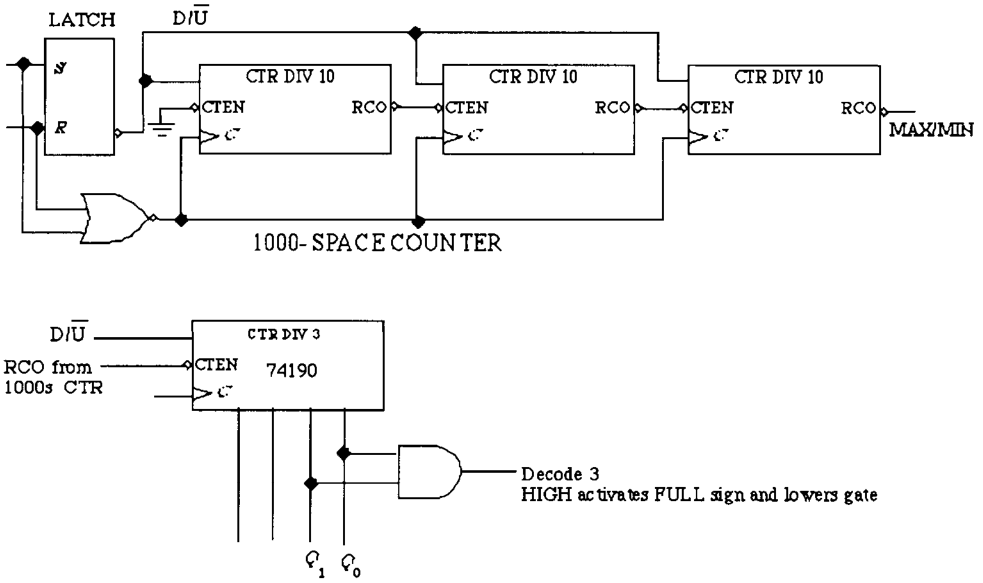


FIGURE 8-40

53. See Figure 8-41.



Add to 1000-space counter for expansion to 3000 spaces

FIGURE 8-41

54. See Figure 8-42.

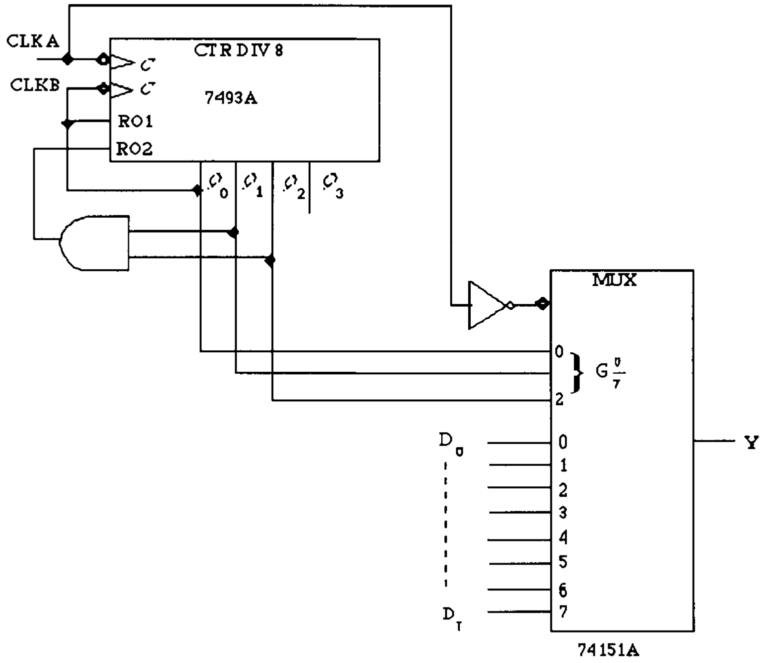


FIGURE 8-42

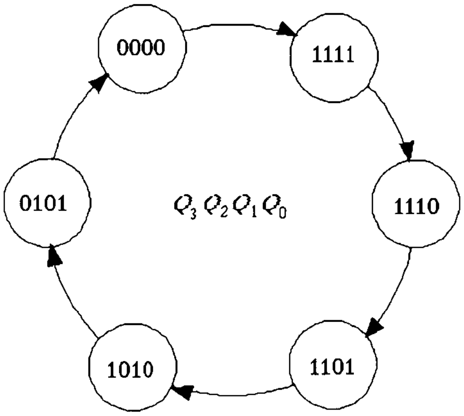
55. NEXT-STATE TABLE

Present State				**Next State**			
Q_3	Q_2	Q_1	Q_0	Q_3	Q_2	Q_1	Q_0
0	0	0	0	1	1	1	1
1	1	1	1	1	1	1	0
1	1	1	0	1	1	0	1
1	1	0	1	1	0	1	0
1	0	1	0	0	1	0	1
0	1	0	1	0	0	0	0

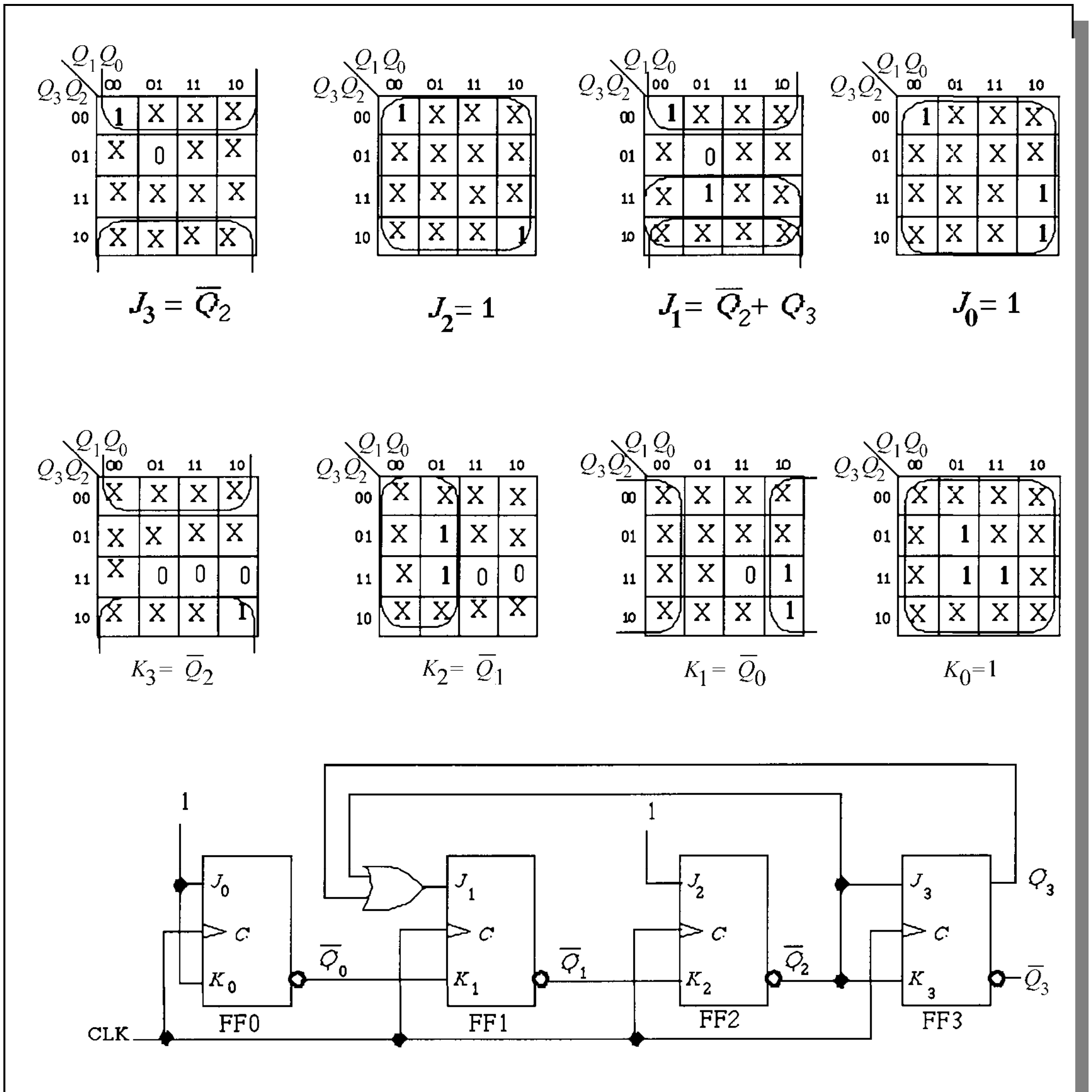
TRANSITION TABLE

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

See Figure 8-43.



The desired sequence



Q_3Q_2 \ Q_1Q_0	00	01	11	10
00	1	X	X	X
01	X	0	X	X
11	X	X	X	X
10	X	X	X	X

$J_3 = \overline{Q}_2$

Q_3Q_2 \ Q_1Q_0	00	01	11	10
00	1	X	X	X
01	X	X	X	X
11	X	X	X	X
10	X	X	X	1

$J_2 = 1$

Q_3Q_2 \ Q_1Q_0	00	01	11	10
00	1	X	X	X
01	X	0	X	X
11	X	1	X	X
10	X	X	X	X

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

Q_3Q_2 \ Q_1Q_0	00	01	11	10
00	1	X	X	X
01	X	X	X	X
11	X	X	X	1
10	X	X	X	1

$J_0 = 1$

Q_3Q_2 \ Q_1Q_0	00	01	11	10
00	X	X	X	X
01	X	X	X	X
11	X	0	0	0
10	X	X	X	1

$K_3 = \overline{Q}_2$

Q_3Q_2 \ Q_1Q_0	00	01	11	10
00	X	X	X	X
01	X	1	X	X
11	X	1	0	0
10	X	X	X	X

$K_2 = \overline{Q}_1$

Q_3Q_2 \ Q_1Q_0	00	01	11	10
00	X	X	X	X
01	X	X	X	X
11	X	X	0	1
10	X	X	X	1

$K_1 = \overline{Q}_0$

Q_3Q_2 \ Q_1Q_0	00	01	11	10
00	X	X	X	X
01	X	1	X	X
11	X	1	1	X
10	X	X	X	X

$K_0 = 1$

FIGURE 8-43

56. See Figure 8-44.

FIGURE 8-44

Multisim Troubleshooting Practice

57. Q output of U3 open.

58. $\overline{SET}$ input of U1 open.

59. Pin A of G3 open.

60. No fault.

61. Pin 9 open.

CHAPTER 9
SHIFT REGISTERS

Section 9-1 Basic Shift Register Operations

1. Shift registers store binary data in a series of flip-flops or other storage elements.

2. 1 byte = **8 bits**; 2 bytes = **16 bits**

3. Shift data and store data

Section 9-2 Serial In/Serial Out Shift Registers

4. Initially: 0000
1st CLK: 1000
2nd CLK: 1100
3rd CLK: 0110

5. See Figure 9-1.

FIGURE 9-1

6. See Figure 9-2.

FIGURE 9-2

7.

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

8. See Figure 9-3.

FIGURE 9-3

9. See Figure 9-4.

FIGURE 9-4

10. See Figure 9-5.

FIGURE 9-5

Section 9-3 Serial In/Parallel Out Shift Registers

11. See Figure 9-6.

FIGURE 9-6

12. See Figure 9-7.

FIGURE 9-7

13. See Figure 9-8.

FIGURE 9-8

Section 9-4 Parallel In/Serial Out Shift Registers

14. See Figure 9-9.

FIGURE 9-9

15. See Figure 9-10.

FIGURE 9-10

16. See Figure 9-11.

FIGURE 9-11

17. See Figure 9-12.

FIGURE 9-12

Section 9-5 Parallel In/Parallel Out Shift Registers

18. See Figure 9-13.

FIGURE 9-13

19. See Figure 9-14.

FIGURE 9-14

20. See Figure 9-15.

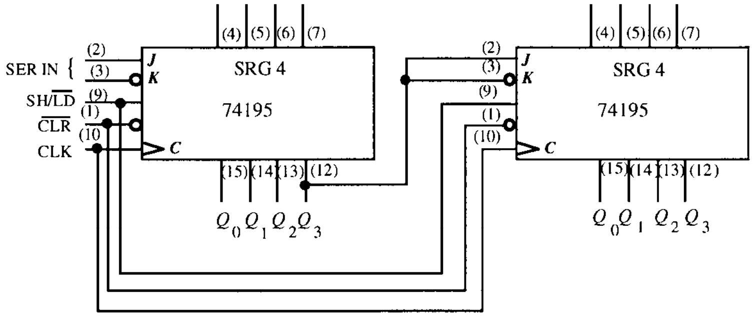


FIGURE 9-15

Section 9-6 Bidirectional Shift Registers

21.

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

22.

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

23. See Figure 9-16.

FIGURE 9-16

24. See Figure 9-17.

FIGURE 9-17

Section 9-7 Shift Register Counters

25. (a) $2n = 6$ (b) $2n = 10$
$n =$ **3** $n =$ **5**

(c) $2n = 14$ (d) $2n = 16$
$n =$ **7** $n =$ **8**

26. $2n = 18$; $n =$ **9** flip-flops

Q_0	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6	Q_7	Q_8
0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0
1	1	1	1	1	0	0	0	0
1	1	1	1	1	1	0	0	0
1	1	1	1	1	1	1	0	0
1	1	1	1	1	1	1	1	0
1	1	1	1	1	1	1	1	1
0	1	1	1	1	1	1	1	1
0	0	1	1	1	1	1	1	1
0	0	0	1	1	1	1	1	1
0	0	0	0	1	1	1	1	1
0	0	0	0	0	1	1	1	1
0	0	0	0	0	0	1	1	1
0	0	0	0	0	0	0	1	1
0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

See Figure 9-18.

FIGURE 9-18

27. See Figure 9-19.

FIGURE 9-19

28. A 15-bit ring counter with stages **3**, **7**, and **12** SET and the remaining stages RESET. See Figure 9-20.

FIGURE 9-20

Section 9-8 Shift Register Applications

29. See Figure 9-21.

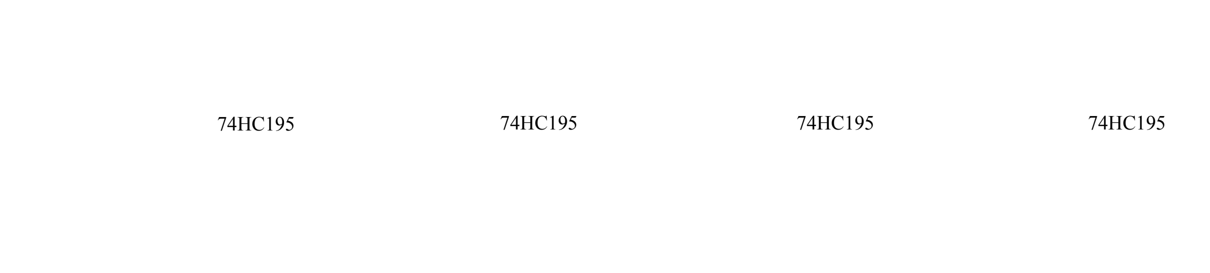


FIGURE 9-21

30. The power-on $\overline{\text{LOAD}}$ input provides a momentary LOW to parallel load the ring counter when power is turned on.

31. An incorrect code may be produced.

Section 9-10 Troubleshooting

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

FIGURE 9-22

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

FIGURE 9-23

34. 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.

35. (a) No clock at switch closure due to faulty NAND gate or one-shot; open clock input to key code register; open $\mathrm{SH}/\overline{\mathrm{LD}}$ 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.

36. *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

Chapter 9

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

Switch number	Key Code Register					
	Q_0	Q_1	Q_2	Q_3	Q_4	Q_5
1	0	1	1	0	1	1
2	0	1	1	1	0	1
3	0	1	1	0	0	1
4	0	1	1	1	1	0
5	0	1	1	0	1	0
6	0	1	1	1	0	0
7	0	1	1	0	0	0
8	0	1	1	1	1	1
9	1	0	1	0	1	1
10	1	0	1	1	0	1
11	1	0	1	0	0	1
12	1	0	1	1	1	0
13	1	0	1	0	1	0
14	1	0	1	1	0	0
15	1	0	1	0	0	0
16	1	0	1	1	1	1
17	0	0	1	0	1	1
18	0	0	1	1	0	1
19	0	0	1	0	0	1
20	0	0	1	1	1	0
21	0	0	1	0	1	0
22	0	0	1	1	0	0
23	0	0	1	0	0	0
24	0	0	1	1	1	1
25	1	1	0	0	1	1
26	1	1	0	1	0	1
27	1	1	0	0	0	1
28	1	1	0	1	1	0
29	1	1	0	0	1	0
30	1	1	0	1	0	0
31	1	1	0	0	0	0
32	1	1	0	1	1	1
33	0	1	0	0	1	1
34	0	1	0	1	0	1
35	0	1	0	0	0	1
36	0	1	0	1	1	0
37	0	1	0	0	1	0
38	0	1	0	1	0	0
39	0	1	0	0	0	0
40	0	1	0	1	1	1
41	1	0	0	0	1	1
42	1	0	0	1	0	1
43	1	0	0	0	0	1
44	1	0	0	1	1	0
45	1	0	0	0	1	0
46	1	0	0	1	0	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

TABLE 1

37. (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.

System Application Activity

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

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

Shift Register A: 1001
Shift Register C: 00000100

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

After key 7 is pressed:
Shift register A contains 0111
Shift register B contains 11000
After key 6 is pressed:
Shift register A contains 0110
Shift register B contains 11100
After key 4 is pressed:
Shift register A contains 0100
Shift register B contains 11110
After key 5 (an incorrect entry) is pressed:
Shift register A contains 0000
Shift register B contains 10000

Chapter 9

Special Design Problems

41. See Figure 9-24.

FIGURE 9-24

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

FIGURE 9-25

43. See Figure 9-26 for one possible implementation.

FIGURE 9-26

44. One possible approach is shown in Figure 9-27.

FIGURE 9-27

45. See Figure 9-28.

FIGURE 9-28

46. Register A requires 8 bits and can be implemented with one 74199. Register B requires 16 bits and can be implemented with two 74199s.

Multisim Troubleshooting Practice

47. CLK input of U3 open.

48. No fault.

49. Pin 14 open.

50. No fault.

51. CLK input of U6 open.

CHAPTER 10
MEMORY AND STORAGE

Section 10-1 Memory Basics

1. (a) ROM: no read/write control
(b) RAM

2. They are random access memories because any address can be accessed at any time. You do not have to go through all the preceding addresses to get to a specific address.

3. ***Address bus*** provides for transfer of address code to memory for accessing any memory location in any order for a read or a write operation.
Data bus provides for transfer of data between the microprocessor and memory or input/output devices.

4. (a) $0A_{16} = 00001010_2 = \mathbf{10_{10}}$
(b) $3F_{16} = 00111111_2 = \mathbf{63_{10}}$
(c) $CD_{16} = 11001101_2 = \mathbf{205_{10}}$

Section 10-2 The Random-Access Memory (RAM)

5.

	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

6. See Figure 10-1.

FIGURE 10-1

7. 64k × 8 = 512 × 128 × 8 = **512 rows × 128 8-bit columns**

8. See Figure 10-2.

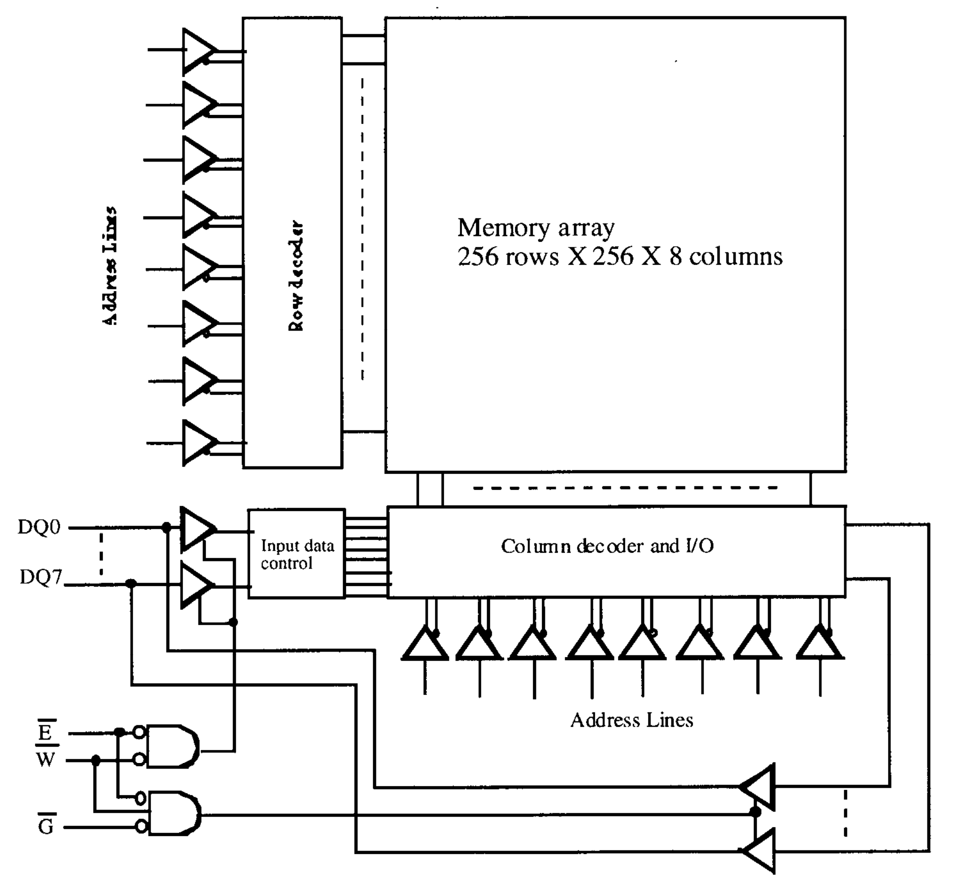


FIGURE 10-2

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

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

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

Section 10-3 The Read-Only Memory (ROM)

11.

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

12.

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

13.

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.

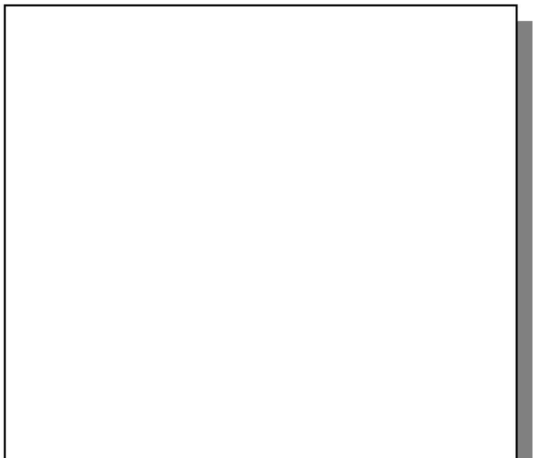


FIGURE 10-3

14. 2^{14} = 16,384 addresses
16,384 × 8 bits = **131,072 bits**

Section 10-4 Programmable ROMs

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

X Input					**X Output**								
	X_2	X_1	X_0	X^3	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	0
1	0	0	1	1	0	0	0	0	0	0	0	0	1
2	0	1	0	8	0	0	0	0	0	1	0	0	0
3	0	1	1	27	0	0	0	0	1	1	0	1	1
4	1	0	0	64	0	0	1	0	0	0	0	0	0
5	1	0	1	125	0	0	1	1	1	1	1	0	1
6	1	1	0	216	0	1	1	0	1	1	0	0	0
7	1	1	1	343	1	0	1	0	1	0	1	1	1

16.

Address	**Contents**
A_{13}--------------A_0	Q_7-------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 10-6 Memory Expansion

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

FIGURE 10-4

18. See Figure 10-5.

FIGURE 10-5

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

Section 10-7 Special Types of Memories

20. See Figure 10-6.

FIGURE 10-6

21. See Figure 10-7.

FIGURE 10-7

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

See Figure 10-8.

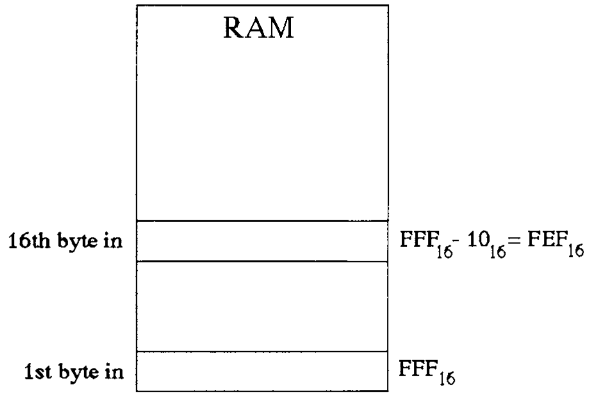


FIGURE 10-8

Section 10-8 Magnetic and Optical Storage

23. 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.

24. Seek time is the average time required to position the drive head over the track containing the desired data. The latency period is the average time required for the data to move under the drive head.

25. Magnetic tape has a longer access time than disk because data must be accessed sequentially rather than randomly.

26. A magneto-optic disk is a read/write medium using lasers and magnetic fields.
A CD-ROM (compact-disk ROM) is a read-only optical (laser) medium.
A WORM (write-once-read-many) is an optical medium in which data can be written once and read many times.

Section 10-9 Troubleshooting

27. The correct checksum is **00100**.
The actual checksum is 01100. The second bit from the left is in error.

28. (a)

ROM0:	Low address - 00_{16}	High address - $1F_{16}$
ROM1:	Low address - 20_{16}	High address - $3F_{16}$
ROM2:	Low address - 40_{16}	High address - $5F_{16}$
ROM3:	Low address - 60_{16}	High address - $7F_{16}$

(b) Same as flow chart in Figure 10-68 in text except that the last data address is specified as **7E**$_{16}$ (7F$_{16}$ – 1).

(c) See Figure 10-9.

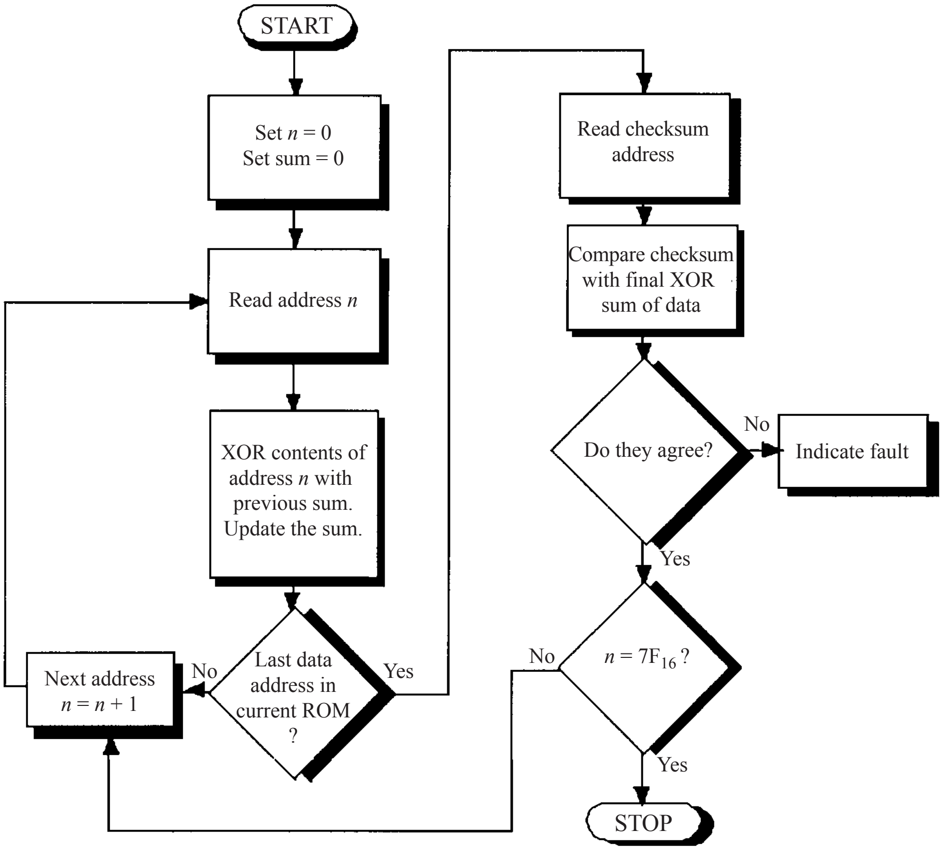


FIGURE 10-9

(d) A single checksum will not isolate the faulty chip. It will only indicate that there is an error in one of the chips.

29. (a) $40_{16} – 5F_{16}$ is 64 – 95 decimal; ROM 2
(b) $20_{16} – 3F_{16}$ is 32 – 63 decimal; ROM 1
(c) $00_{16} – 7F_{16}$ is 0 – 127 decimal; All ROMs

System Application Activity

30. See Figure 10-10.

FIGURE 10-10

31. On first digit entry, the register state is 0001. On second digit entry the register state is 0010.

32. The purpose of the switch memory is to store a 4-digit security code and permit easy code change.

33. A new code can be entered by simply moving the DIP switch settings. No power is required.

Special Design Problems

34. NAND gates U1A-U4D: four 74HC00
NAND gates U5A-U6B: two 74HC20
Shift register U7: 74HC195

35. See Figure 10-11.

FIGURE 10-11

CHAPTER 11
PROGRAMMABLE LOGIC AND SOFTWARE

Section 11-1 Programmable Logic: SPLDs and CPLDs

1. $X = \overline{A}\,\overline{B}\,\overline{C} + \overline{A}B\overline{C} + A\overline{B}C$. See Figure 11-1.

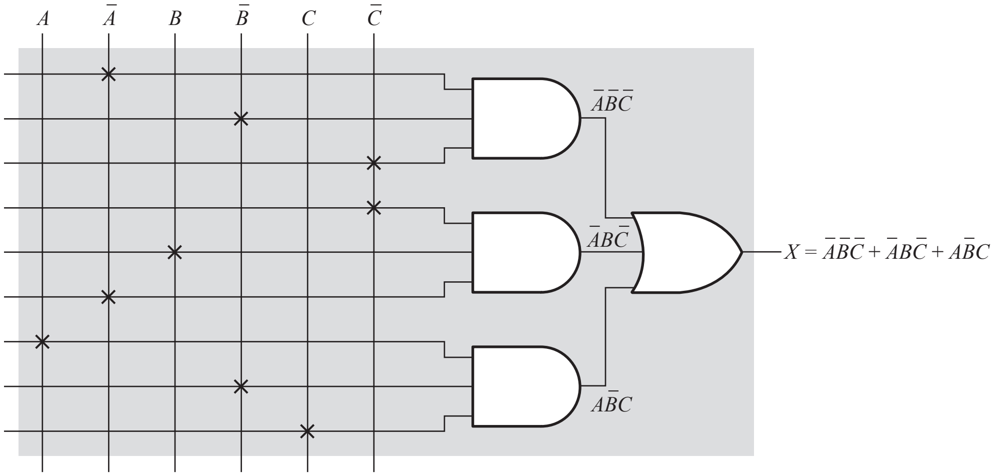
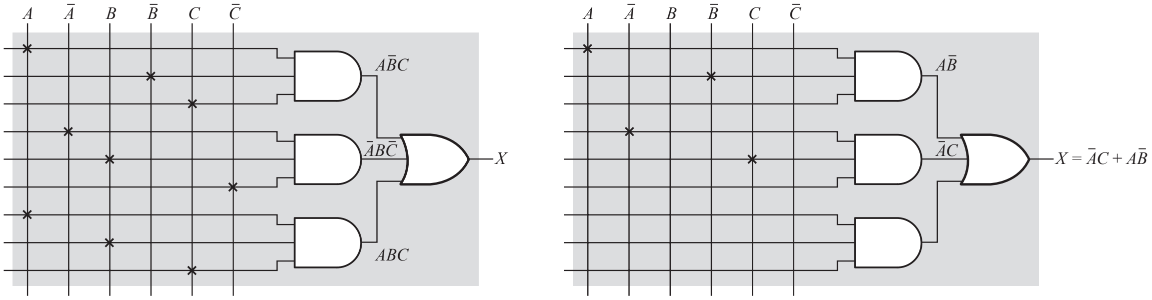

A
$\overline{A}$
B
$\overline{B}$
C
$\overline{C}$
$\overline{A}\,\overline{B}\,\overline{C}$
$\overline{A}B\overline{C}$
$A\overline{B}C$
$X = \overline{A}\,\overline{B}\,\overline{C} + \overline{A}B\overline{C} + A\overline{B}C$

FIGURE 11-1

2. See Figure 11-2.



(a)

(b) Cannot be implemented as stated because it has 4 product terms.
Simplify to: $X = \overline{A}C + A\overline{B}$

FIGURE 11-2

3. (a) PAL16L2 is a programmable array logic device with 16 inputs and two active-LOW outputs.

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

4. Typically, an exclusive-OR gate is used to determine the polarity of the output. When a 1 is applied to one input of the XOR gate, the output of the XOR is the complement of the signal on the other input. When a 0 is applied to one input of the XOR, the signal on the output of the XOR is the same as the signal on the other input.

5. A CPLD basically consists of multiple SPLDs that can be connected with a programmable interconnect array.

Section 11-2 Altera CPLDs

6. (a) Inputs from PIA to LAB: **36** (b) Outputs from LAB to PIA: **16**
(c) Inputs from I/O to PIA: **8 to 16** (d) Outputs from LAB to I/O: **8 to 16**

7. (a) $\overline{A}BC\overline{D}$ (b) $ABC(\overline{DE}) = ABC(\overline{D} + \overline{E}) = ABC\overline{D} + ABC\overline{E}$

8. $A\overline{B}C\overline{D} + EFGH + AB\overline{C}D + \overline{A}BCD$

Section 11-3 Xilinx CPLDs

9. $A\overline{B} + \overline{A}B$

10. (a) Inputs from AIM to FB: **40** (b) Outputs from FB to AIM: **16**
(c) Inputs from I/O to AIM: **16** (d) Outputs from FB to I/O: **16**

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

Section 11-4 Macrocells

12. (a) A 0 on the select line selects D_0. The output is **1**.

(b) A 1 on the select line selects D_1. The output is **0**.

13. (a) Since the D_0 (upper) input of MUX 5 is selected, the macrocell is configured for **combinational** logic**.** The output of the XOR goes through MUX 5 to the “To I/O” output making it a **1**.

(b) Since the D_1 (lower) input of MUX 5 is selected, the macrocell is configured for **registered** logic**.** The output of the flip-flop goes through MUX 5 to the “To I/O” output making it a **0**.

14. (a) The macrocell is configured for **registered** logic because the D_1 input of MUX 8 is selected, allowing the flip-flop output to pass through.

(b) The **GCK1** clock is applied to the flip-flop because the D_1 input of MUX 3 and the D_1 input of MUX 5 are selected.

(c) The OR gate output is applied to the XOR which is set for noninversion by MUX 1. The output of the XOR is selected by MUX2 and a **1** is applied to the D/T input of the flip-flop.

(d) The output of MUX 8 is a **1** because MUX 8 selects the Q output of the flip-flop (assuming that the S and R inputs are 0).

15. (a) The macrocell is configured for **registered** logic because the D_1 input of MUX 8 is selected, allowing the flip-flop output to pass through.

(b) The **GCK1** clock is applied to the flip-flop because the D_1 input of MUX 3 and the D_1 input of MUX 5 are selected.

(c) The OR gate output is applied to the XOR which is set for inversion by MUX 1. The output of the XOR is selected by MUX 2 and a **0** is applied to the D/T input of the flip-flop.

(d) The output of MUX 8 is a **0** because MUX 8 selects the Q output of the flip-flop (assuming that the S and R inputs are 0).

Section 11-5 Programmable Logic: FPGAs

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

17. SOP output = $\overline{A}\,\overline{B}\,\overline{C} + \overline{A}\,\overline{B}C + \overline{A}BC + A\overline{B}C + AB\overline{C}$

18. See Figure 11-3.

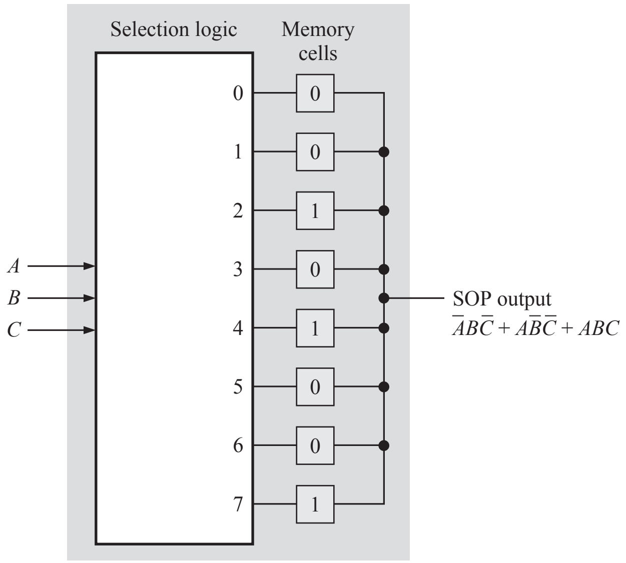

Selection logic
Memory cells
0
1
2
3
4
5
6
7
0
0
1
0
1
0
0
1
A
B
C
SOP output
$\overline{A}B\overline{C} + A\overline{B}\,\overline{C} + ABC$

FIGURE 11-3

Section 11-6 Altera FPGAs

19. An ALM consists of an LUT for combinational logic, adder logic, and register logic.

20. The modes of operation of an ALM are: Normal, Extended LUT, Arithmetic, and Shared arithmetic.

21. See Figure 11-4.

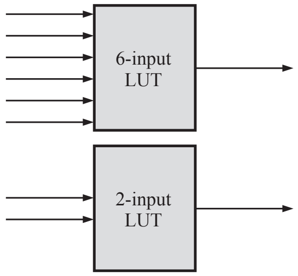


FIGURE 11-4

22.
$$(A_4A_3\overline{A}_2A_1 + \overline{A}_4\overline{A}_3\overline{A}_2A_1)A_0 + (\overline{A}_5A_3A_2A_1 + A_5\overline{A}_3A_2\overline{A}_1 + A_5A_3A_2\overline{A}_1)A_0$$
$$= A_4A_3\overline{A}_2A_1A_0 + \overline{A}_4\overline{A}_3\overline{A}_2A_1A_0 + \overline{A}_5A_3A_2A_1\overline{A}_0 + A_5\overline{A}_3A_2\overline{A}_1\overline{A}_0 + A_5A_3A_2\overline{A}_1\overline{A}_0$$

Section 11-7 Xilinx FPGAs

23. See Figure 11-5.

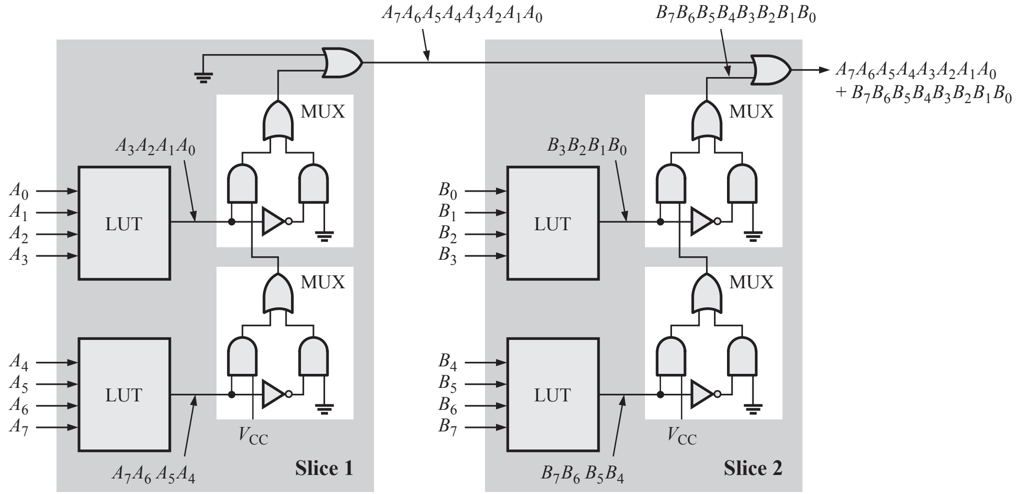


FIGURE 11-5

24. See Figure 11-6.

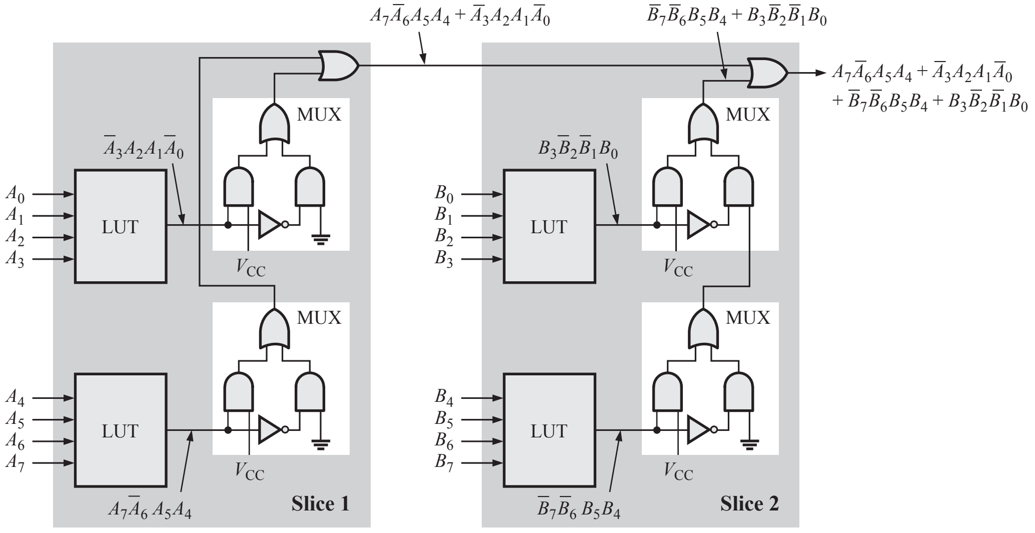

$A_7\overline{A}_6A_5A_4 + \overline{A}_3A_2A_1\overline{A}_0$
$\overline{B}_7\overline{B}_6B_5B_4 + B_3\overline{B}_2\overline{B}_1B_0$
$A_7\overline{A}_6A_5A_4 + \overline{A}_3A_2A_1\overline{A}_0$
$+ \overline{B}_7\overline{B}_6B_5B_4 + B_3\overline{B}_2\overline{B}_1B_0$
MUX
MUX
$\overline{A}_3A_2A_1\overline{A}_0$
$B_3\overline{B}_2\overline{B}_1B_0$
A_0
A_1
A_2
A_3
LUT
V_{CC}
B_0
B_1
B_2
B_3
LUT
V_{CC}
MUX
MUX
A_4
A_5
A_6
A_7
LUT
V_{CC}
B_4
B_5
B_6
B_7
LUT
V_{CC}
$A_7\overline{A}_6\ A_5A_4$
Slice 1
$\overline{B}_7\overline{B}_6\ B_5B_4$
Slice 2

FIGURE 11-6

25. One slice. See Figure 11-7.

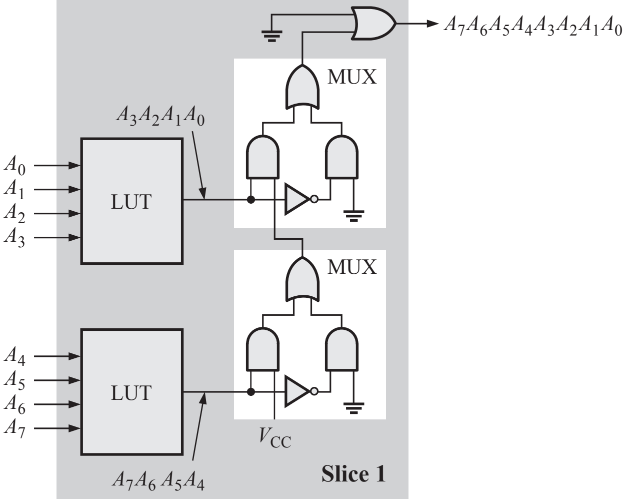


FIGURE 11-7

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

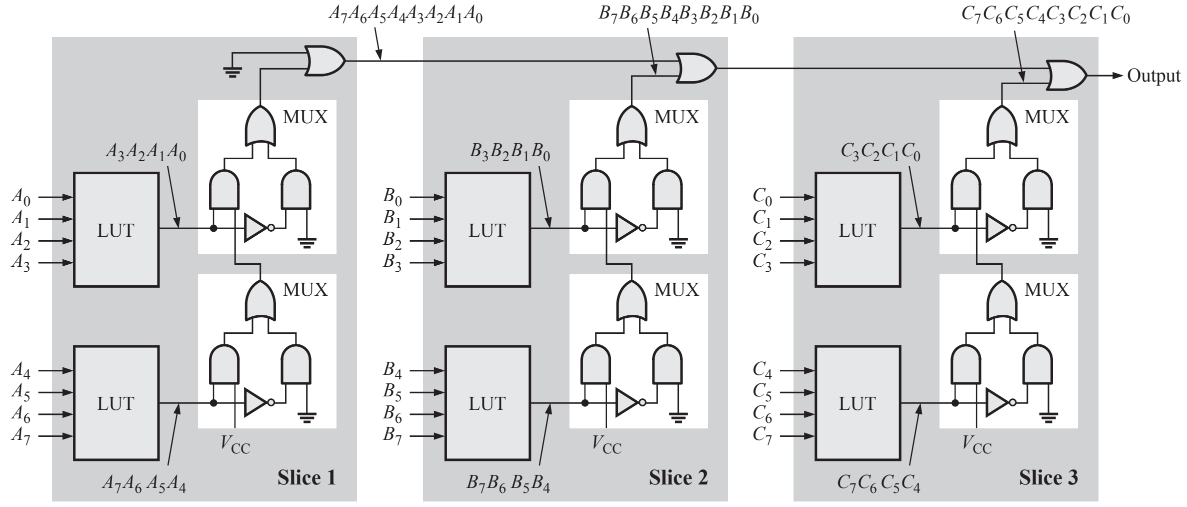


FIGURE 11-8

Section 11-8 Programmable Logic Software

27. See Figure 11-9.

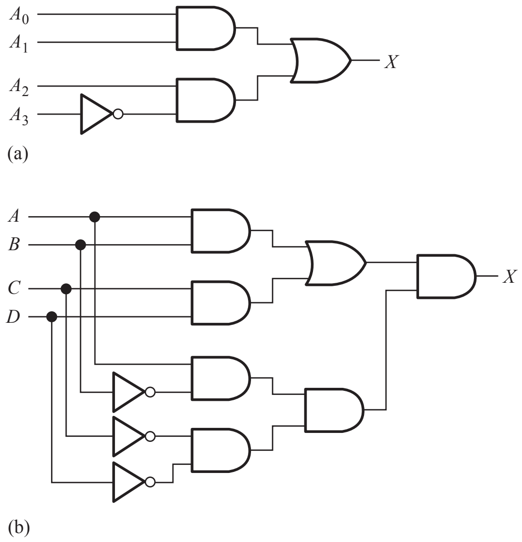


FIGURE 11-9

28. $X = \overline{A}BCD + A\overline{B}CD + AB\overline{C}D + ABC\overline{D} + ABCD + \overline{A}\,\overline{B}\,\overline{C}\,\overline{D}$
$= ABD + ACD + ABC + BCD + \overline{A}\,\overline{B}\,\overline{C}\,\overline{D}$

See Figure 11-10.

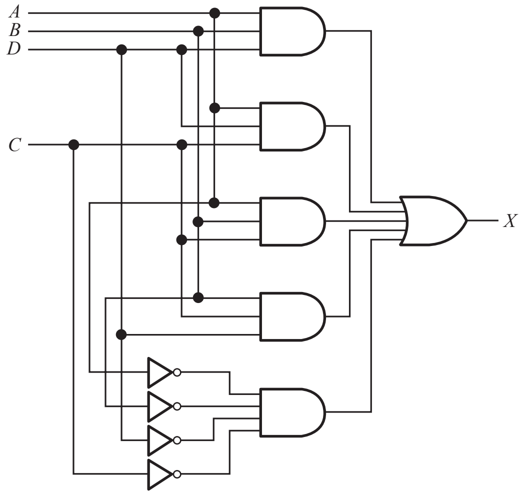


FIGURE 11-10

29. See Figure 11-11.

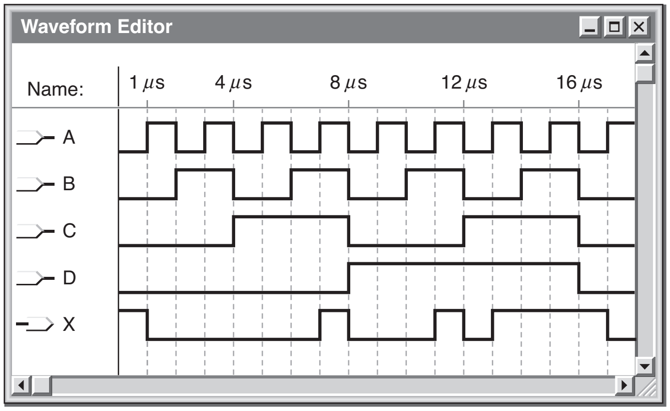


FIGURE 11-11

30. $X = \overline{A}BC\overline{D} + A\overline{B}\overline{C}D + ABCD + A\overline{B}C\overline{D} + \overline{A}B\overline{C}D$. See Figure 11-12.

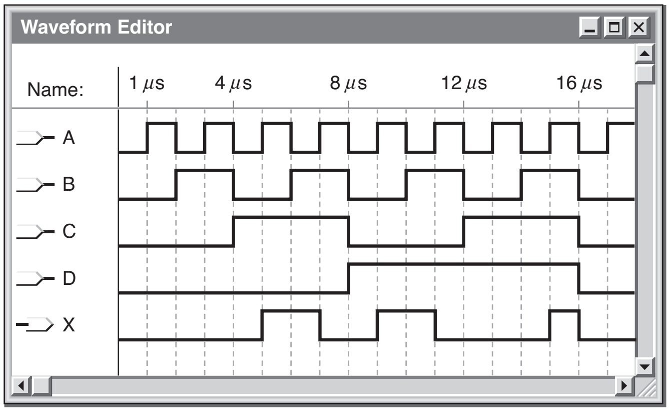


FIGURE 11-12

Section 11-9 Boundary Scan Logic

31. 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 clock into Capture register B on the trailing edge of the clock pulse.

32. PDI/O = 0 and OE = 1. The data from the internal programmable logic pass through the selected MUX and through the output buffer to the pin.

33. PDI/O = 0 and OE = 0. The data are applied to the input pin and go through the selected MUX to the internal programmable logic.

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

Section 11-10 Troubleshooting

35. 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	**0000**11001010001111011
1	0**0001**1001010001111011
3	00**0011**001010001111011
6	000**0110**01010001111011
12	0000**1100**1010001111011
9	00001**1001**010001111011
2	000011**0010**10001111011
5	0000110**0101**0001111011
10	00001100**1010**001111011
4	000011001**0100**01111011
8	0000110010**1000**1111011
1	00001100101**0001**111011
3	000011001010**0011**11011
7	0000110010100**0111**1011
15	00001100101000**1111**011
14	000011001010001**1110**11
13	0000110010100011**1101**1
11	00001100101000111**1011**

System Application Activity

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

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

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

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

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

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

7 AND gates can be eliminated.

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

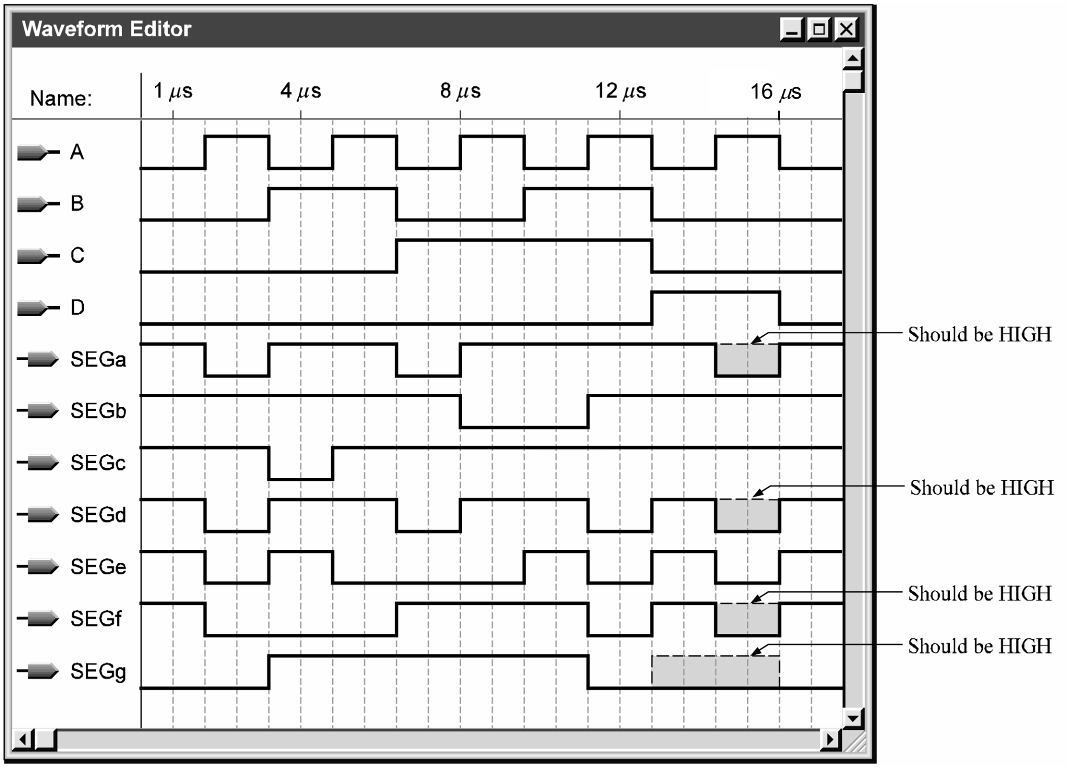


FIGURE 11-13