

FIGURE 1–58 Concept of the signal-tracing method. Input to output is shown. The same applies if you start at the output and go toward the input.

at the output where there is an incorrect or absent signal and go back toward the input from point to point until a correct signal is found. Also, you can begin at the input and check the signal and move toward the output from point to point until the correct signal is lost. In both cases, the fault would be between the point and the output. Of course, you must know what the signal is supposed to look like in order to know if anything is wrong. Figure 1–58 illustrates the concept of signal tracing.

Signal Substitution and Injection

Signal substitution is used when the system being tested has been separated from its signal source. A generator signal is used to replace the normal signal that comes from the source when the system or portion of a system is recombined with the part that normally produces the input signal. Signal injection can be used to insert a signal at certain points in the system using the half-splitting approach.

SECTION 1–8 CHECKUP

1. List five steps in the troubleshooting procedure.
2. Name two troubleshooting methods.
3. List five obvious things to look for in a failed system.
4. Is it important to know about the relationship between a cause and a symptom?

SUMMARY

- An analog quantity has continuous values.
- A digital quantity has a discrete set of values.
- A binary digit is called a bit.
- A pulse is characterized by rise time, fall time, pulse width, and amplitude.
- The frequency of a periodic waveform is the reciprocal of the period. The formulas relating frequency and period are

$$f = \frac{1}{T} \quad \text{and} \quad T = \frac{1}{f}$$

- The duty cycle of a pulse waveform is the ratio of the pulse width to the period, expressed by the following formula as a percentage:

$$\text{Duty cycle} = \left(\frac{t_W}{T} \right) 100\%$$

-  A timing diagram is an arrangement of two or more waveforms showing their relationship with respect to time.
- Three basic logic operations are NOT, AND, and OR. The standard symbols for these are given in Figure 1–59.

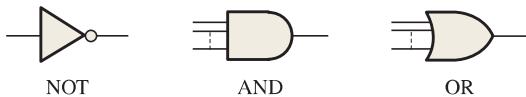


FIGURE 1–59

BCD/Grey

- The basic logic functions are comparison, arithmetic, code conversion, decoding, encoding, data selection, storage, and counting.
- Two types of SPLDs (simple programmable logic devices) are PAL (programmable array logic) and GAL (generic array logic).
- The CPLD (complex programmable logic device) contains multiple SPLDs with programmable interconnections.
- The FPGA (field-programmable gate array) has a different internal structure than the CPLD and is generally used for more complex circuits and systems.
- The two broad physical categories of IC packages are through-hole mounted and surface mounted.
- Three families of fixed-function integrated circuits are CMOS, bipolar, and BiCMOS.
- Bipolar is also known as TTL (transistor-transistor logic).
- The categories of ICs in terms of circuit complexity are SSI (small-scale integration), MSI (medium-scale integration), LSI, VLSI, and ULSI (large-scale, very large-scale, and ultra large-scale integration).
- Common instruments used in testing and troubleshooting digital circuits are the oscilloscope, logic analyzer, arbitrary waveform generator, data pattern generator, function generator, dc power supply, digital multimeter, logic probe, and logic pulser.
- Two basic methods of troubleshooting are the half-splitting method and the signal-tracing method.

KEY TERMS

Key terms and other bold terms in the chapter are defined in the end-of-book glossary.

Analog Being continuous or having continuous values.

AND A basic logic operation in which a true (HIGH) output occurs only when all the input conditions are true (HIGH).

Binary Having two values or states; describes a number system that has a base of two and utilizes 1 and 0 as its digits.

Bit A binary digit, which can be either a 1 or a 0.

Clock The basic timing signal in a digital system; a periodic waveform used to synchronize operation.

Compiler A program that controls the design flow process and translates source code into object code in a format that can be logically tested or downloaded to a target device.

CPLD A complex programmable logic device that consists basically of multiple SPLD arrays with programmable interconnections.

Data Information in numeric, alphabetic, or other form.

Digital Related to digits or discrete quantities; having a set of discrete values.

Duty cycle The ratio of the pulse width to the period of a digital waveform, expressed as a percentage.

01	00	00	00	00
00	00	10	00	00
00	11	11	11	11
11	11	11	11	11
11	11	11	11	11
11	01	01	01	01
01	01	01	01	01
01	10	00	10	00
10	00	01	01	01
01	01	11	00	00
01	00	11	10	00
00	10	11	10	10
10	10	01	11	11
10	00	01	00	00
00	11	10	11	11



Embedded system Generally, a single-purpose system, such as a processor, built into a larger system for the purpose of controlling the system.

✓ Fixed-function logic A category of digital integrated circuits having functions that cannot be altered.

FPGA Field-programmable gate array.

Gate A logic circuit that performs a basic logic operation such as AND or OR.

✓ Input The signal or line going into a circuit.

Integrated circuit (IC) A type of circuit in which all of the components are integrated on a single chip of semiconductive material of extremely small size.

Inverter A NOT circuit; a circuit that changes a HIGH to a LOW or vice versa.

Logic In digital electronics, the decision-making capability of gate circuits, in which a HIGH represents a true statement and a LOW represents a false one.

✓ Microcontroller An integrated circuit consisting of a complete computer on a single chip and used for specified control functions.

NOT A basic logic operation that performs inversions.

OR A basic logic operation in which a true (HIGH) output occurs when one or more of the input conditions are true (HIGH).

✓ Output The signal or line coming out of a circuit.

Parallel In digital systems, data occurring simultaneously on several lines; the transfer or processing of several bits simultaneously.

Programmable logic A category of digital integrated circuits capable of being programmed to perform specified functions.

Pulse A sudden change from one level to another, followed after a time, called the pulse width, by a sudden change back to the original level.

Serial Having one element following another, as in a serial transfer of bits; occurring in sequence rather than simultaneously.

SPLD Simple programmable logic device.

Timing diagram A graph of digital waveforms showing the time relationship of two or more waveforms.

Troubleshooting The technique or process of systematically identifying, isolating, and correcting a fault in a circuit or system.

TRUE/FALSE QUIZ

Answers are at the end of the chapter.

- ✓ 1.** An analog quantity is one having continuous values.
- 2.** A digital quantity has no discrete values.
- ✓ 3.** There are two digits in the binary system.
- ✓ 4.** The term *bit* is short for binary digit.
- 5.** In positive logic, a LOW level represents a binary 1.
- ✓ 6.** A periodic wave repeats itself at a fixed interval.
- ✓ 7.** A timing diagram shows the timing relationship of two or more digital waveforms.
- 8.** An AND function is implemented by a logic circuit known as an inverter.
- 9.** A flip-flop is a bistable logic circuit that can store only two bits at a time.
- ✗ 10.** Two broad types of digital integrated circuits are fixed-function and programmable.

SELF-TEST

Answers are at the end of the chapter.

1. A quantity having discrete numerical values is
 - (a) an analog quantity
 - (b) a digital quantity
 - (c) a binary quantity
 - (d) a natural quantity

2. The term *bit* means
 (a) a small amount of data (b) a 1 or a 0
 (c) binary digit **(d)** both answers (b) and (c)
3. The time interval between the 50% points on the rising and falling edges is
 (a) rise time (b) fall time
 (c) pulse width (d) period
4. A pulse in a certain waveform has a frequency of 50 Hz. It repeats itself every
 (a) 1 ms (b) 20 ms (c) 50 ms (d) 100 ms
5. In a certain digital waveform, the period is four times the pulse width. The duty cycle is
 (a) 25% (b) 50% (c) 75% (d) 100%
6. An inverter
 (a) performs the NOT operation (b) changes a HIGH to a LOW
 (c) changes a LOW to a HIGH (d) does all of the above
7. The output of an OR gate is LOW when
 (a) any input is HIGH (b) all inputs are HIGH
 (c) no inputs are HIGH (d) Both (a) and (b)
8. The output of an AND gate is LOW when
 (a) any input is LOW (b) all inputs are HIGH
 (c) no inputs are HIGH (d) Both (a) and (c)
9. The device used to convert a binary number to a 7-segment display format is the
 (a) multiplexer (b) encoder
 (c) decoder (d) register
10. An example of a data storage device is
 (a) the logic gate (b) the flip-flop (c) the comparator
 (d) the register (e) both answers (b) and (d)
11. VHDL is a
 (a) logic device (b) PLD programming language
 (c) computer language (d) very high density logic
12. A CPLD is a
 (a) controlled program logic device (b) complex programmable logic driver
 (c) complex programmable logic device (d) central processing logic device
13. An FPGA is a
 (a) field-programmable gate array (b) fast programmable gate array
 (c) field-programmable generic array (d) flash process gate application
14. A fixed-function IC package containing four AND gates is an example of
 (a) MSI (b) SMT (c) SOIC (d) SSI
15. An LSI device has a circuit complexity of from
 (a) 10 to 100 equivalent gates (b) more than 100 to 10,000 equivalent gates
 (c) 2000 to 5000 equivalent gates (d) more than 10,000 to 100,000 equivalent gates

PROBLEMS

Answers to odd-numbered problems are at the end of the book.

Section 1–1 Digital and Analog Quantities

1. Name two advantages of digital data as compared to analog data.
2. Which quantities are more affected by noise: analog or digital?
3. List any three common products that measure analog quantities.

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

4. Can a digital system exist over a complete interval of time? Why or why not?
5. Define the sequence of bits (1s and 0s) represented by each of the following sequences of levels:
 (a) HIGH, HIGH, LOW, LOW, LOW, LOW, HIGH, HIGH
 (b) HIGH, LOW, HIGH, LOW, HIGH, LOW, HIGH, LOW

01	00	00	00	00
00	00	10	00	00
00	11	11	11	11
11	11	00	11	11
11	11	11	11	11
11	01	01	01	01
01	01	01	01	01
01	10	00	01	01
01	01	00	01	01
01	01	11	10	00
00	10	11	10	00
10	10	01	11	10
10	00	01	11	10
00	11	10	11	11

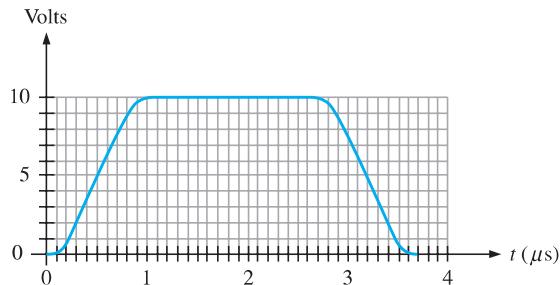


FIGURE 1-60

8. Can the digital waveform in Figure 1–61 be called a pulse train?
 9. What is the frequency of the waveform in Figure 1–61?
 10. Is the pulse waveform in Figure 1–61 periodic or nonperiodic?
 11. Determine the duty cycle of the waveform in Figure 1–61.

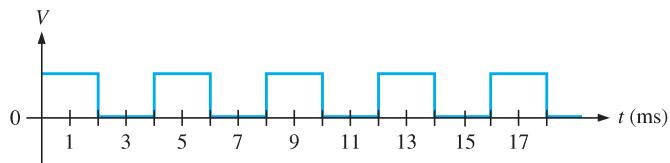


FIGURE 1-61

12. Determine the bit sequence represented by the waveform in Figure 1–62. A bit time is $1 \mu\text{s}$ in this case.
 13. What is the total serial transfer time for the eight bits in Figure 1–62? What is the total parallel transfer time?
 14. What is the period if the clock frequency is 4 kHz?

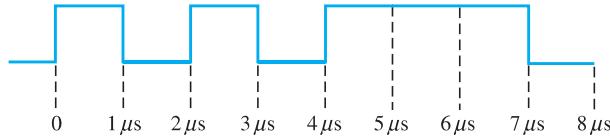


FIGURE 1-62

Section 1–3 Basic Logic Functions

- 15.** Form a single logical statement from the following information:

 - (a) The light is ON if SW1 is closed.
 - (b) The light is ON if SW2 is closed.
 - (c) The light is OFF if both SW1 and SW2 are open.

16. The output of a logic gate is an inversion of the input. What type of logic gate is it?

17. A basic 2-input logic circuit has a HIGH on one input and a LOW on the other, and the output is HIGH. Identify the circuit.

18. A basic 3-input logic circuit has a LOW on one input and a HIGH on the other two inputs, and the output is LOW. What type of logic circuit is it?

Section 1–4 Combinational and Sequential Logic Functions

19. Name the logic function of each block in Figure 1–63 based on your observation of the inputs and outputs.

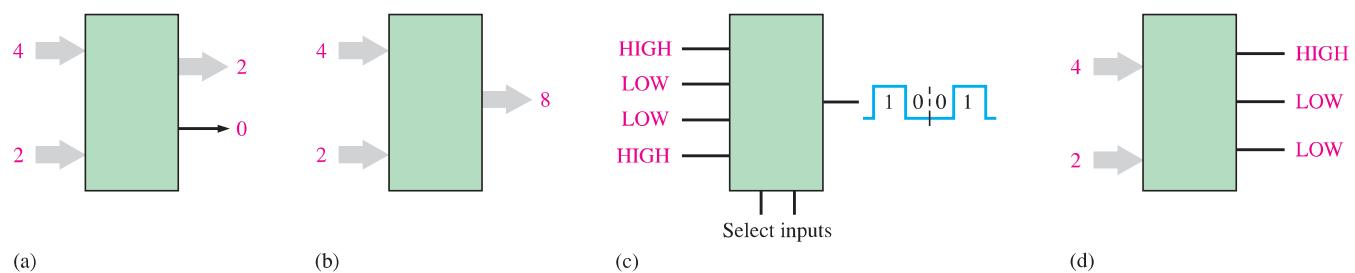


FIGURE 1–63

20. A pulse waveform with a frequency of 20 kHz is applied to the input of a counter. During 40 ms, how many pulses are counted?
21. Consider a register that can store eight bits. Assume that it has been reset so that it contains zeros in all positions. If you transfer four alternating bits (0101) serially into the register, beginning with a 1 and shifting to the right, what will the total content of the register be as soon as the fourth bit is stored?

Section 1–5 Introduction to Programmable Logic

22. Describe each of the following programming steps:
- Synthesis
 - Implementation
 - Compiler
23. What do each of the following stand for?
- SPLD
 - CPLD
 - HDL
 - FPGA
 - GAL
24. Define each of the following PLD programming terms:
- design entry
 - simulation
 - compilation
 - download
25. Describe the process of place-and-route.

Section 1–6 Fixed-Function Logic Devices

26. How are integrated circuit packages classified?
27. What are LSI circuits?
28. Label the pin numbers on the packages in Figure 1–64. Top views are shown.

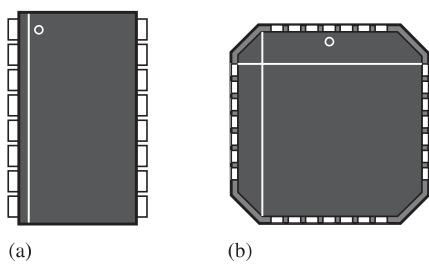


FIGURE 1–64

Section 1–7 Test and Measurement Instruments

29. A pulse is displayed on the screen of an oscilloscope, and you measure the base line as 2 V and the top of the pulse as 10 V. What is the amplitude?
30. A waveform is measured on the oscilloscope and its amplitude covers two vertical divisions. If the vertical control is set at 1 V/div, what is the total amplitude of the waveform?
31. The period of a pulse waveform measures four horizontal divisions on an oscilloscope. If the time base is set at 2 ms/div, what is the frequency of the waveform?

32. What record length is required if an oscilloscope has a sampling rate of 12 Msamples/s and the input waveform is sampled for 2 ms?

Section 1–8 Introduction to Troubleshooting

- 33.** Define troubleshooting.
 - 34.** Explain the half-splitting method of troubleshooting.
 - 35.** Explain the signal-tracing method of troubleshooting.
 - 36.** Discuss signal substitution and injection.
 - 37.** Give some examples of the type of information that you look for when a system is reported to have failed.
 - 38.** If the symptom in a particular system is no output, name two possible general causes.
 - 39.** If the symptom of a particular system is an incorrect output, name two possible causes.
 - 40.** What obvious things should you look for before starting the troubleshooting process?
 - 41.** How would you isolate a fault in a system?
 - 42.** Name two common instruments used in troubleshooting.
 - 43.** Assume that you have isolated the problem down to a specific circuit board. What are your options at this point?

ANSWERS

SECTION CHECKUPS

Section 1–1 Digital and Analog Quantities

1. *Analog* means continuous.
 2. *Digital* means discrete.
 3. A digital quantity has a discrete set of values and an analog quantity has continuous values.
 4. A public address system is analog. A CD player is analog and digital. A computer is all digital.
 5. A mechatronic system consists of both mechanical and electronic components.

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

1. Binary means having two states or values.
 2. A bit is a binary digit.
 3. The bits are 1 and 0.
 4. Rise time: from 10% to 90% of amplitude. Fall time: from 90% to 10% of amplitude.
 5. Frequency is the reciprocal of the period.
 6. A clock waveform is a basic timing waveform from which other waveforms are derived.
 7. A timing diagram shows the time relationship of two or more waveforms.
 8. Parallel transfer is faster than serial transfer.

Section 1–3 Basic Logic Functions

1. When the input is LOW
 2. When all inputs are HIGH
 3. When any or all inputs are HIGH
 4. An inverter is a NOT circuit.
 5. A logic gate is a circuit that performs a logic operation (AND, OR).

Section 1-4 Combinational and Sequential Logic Functions

1. A comparator compares the magnitudes of two input numbers.
 2. Add, subtract, multiply, and divide

3. Encoding is changing a familiar form such as decimal to a coded form such as binary.
4. Decoding is changing a code to a familiar form such as binary to decimal.
5. Multiplexing puts data from many sources onto one line. Demultiplexing takes data from one line and distributes it to many destinations.
6. Flip-flops, registers, semiconductor memories, magnetic disks
7. A counter counts events with a sequence of binary states.

Section 1–5 Introduction to Programmable Logic

1. Simple programmable logic device (SPLD), complex programmable logic device (CPLD), and field-programmable gate array (FPGA)
2. A CPLD is made up of multiple SPLDs.
3. Design entry, functional simulation, synthesis, implementation, timing simulation, and download
4. *Design entry:* The logic design is entered using development software. *Functional simulation:* The design is software simulated to make sure it works logically. *Synthesis:* The design is translated into a netlist. *Implementation:* The logic developed by the netlist is mapped into the programmable device. *Timing simulation:* The design is software simulated to confirm that there are no timing problems. *Download:* The design is placed into the programmable device.
5. The microcontroller has fixed internal circuits and its operation is directed by a program.

Section 1–6 Fixed-Function Logic Devices

1. An IC is an electronic circuit with all components integrated on a single silicon chip.
2. DIP—dual in-line package; SMT—surface-mount technology;
SOIC—small-outline integrated circuit; SSI—small-scale integration; MSI—medium-scale integration; LSI—large-scale integration; VLSI—very large-scale integration; ULSI—ultra large-scale integration
3. (a) SSI
(b) MSI
(c) LSI
(d) VLSI
(e) ULSI

Section 1–7 Test and Measurement Instruments

1. The oscilloscope measures, processes, and displays electrical waveforms.
2. The logic analyzer has more channels than the oscilloscope and has more than one data display format.
3. The volts/div control sets the voltage for each division on the screen.
4. The sec/div control sets the time for each division on the screen.
5. The function generator produces various types of waveforms.
6. The record length is the maximum number of samples that can be acquired during a given time interval.

Section 1–8 Introduction to Troubleshooting

1. Gather information, identify symptoms and possible causes, isolate point(s) of failure, apply proper tools to determine cause, and fix problem.
2. Half-splitting and signal tracing
3. Blown fuse, absence of DC power, loose connections, broken wires, loosely connected circuit board
4. Yes

01 00 00 00
00 00 10 00
00 11 11 11
11 11 00 11
11 11 11 11
11 01 01 01
01 01 01 01
01 10 00 10
10 01 00 01
01 01 11 00
01 00 11 10
00 10 11 10
10 10 01 10
10 00 01 00
00 11 10 11

RELATED PROBLEMS FOR EXAMPLES

1-1 $f = 6.67 \text{ kHz}$; Duty cycle = 16.7%

1-2 Serial transfer: 3.33 ns

1-3 Amplitude = 12 V; $T = 8 \text{ ms}$

TRUE/FALSE QUIZ

1. T **2.** F **3.** T **4.** T **5.** F **6.** T **7.** T **8.** F **9.** F **10.** T

SELF-TEST

1. (b) **2.** (c) **3.** (a) **4.** (b) **5.** (a) **6.** (d) **7.** (b) **8.** (a) **9.** (d)
10. (e) **11.** (c) **12.** (a) **13.** (d) **14.** (d) **15.** (b)

SUMMARY

- A binary number is a weighted number in which the weight of each whole number digit is a positive power of two and the weight of each fractional digit is a negative power of two. The whole number weights increase from right to left—from least significant digit to most significant.
- A binary number can be converted to a decimal number by summing the decimal values of the weights of all the 1s in the binary number.
- A decimal whole number can be converted to binary by using the sum-of-weights or the repeated division-by-2 method.
- A decimal fraction can be converted to binary by using the sum-of-weights or the repeated multiplication-by-2 method.
- The basic rules for binary addition are as follows:

$$0 + 0 = 0$$

$$0 + 1 = 1$$

$$1 + 0 = 1$$

$$1 + 1 = 10$$

- The basic rules for binary subtraction are as follows:

$$0 - 0 = 0$$

$$1 - 1 = 0$$

$$1 - 0 = 1$$

$$10 - 1 = 1$$

- The 1's complement of a binary number is derived by changing 1s to 0s and 0s to 1s.
- The 2's complement of a binary number can be derived by adding 1 to the 1's complement.
- Binary subtraction can be accomplished with addition by using the 1's or 2's complement method.
- A positive binary number is represented by a 0 sign bit.
- A negative binary number is represented by a 1 sign bit.
- For arithmetic operations, negative binary numbers are represented in 1's complement or 2's complement form.
- In an addition operation, an overflow is possible when both numbers are positive or when both numbers are negative. An incorrect sign bit in the sum indicates the occurrence of an overflow.
- The hexadecimal number system consists of 16 digits and characters, 0 through 9 followed by A through F.
- One hexadecimal digit represents a 4-bit binary number, and its primary usefulness is in simplifying bit patterns and making them easier to read.
- A decimal number can be converted to hexadecimal by the repeated division-by-16 method.
- The octal number system consists of eight digits, 0 through 7.
- A decimal number can be converted to octal by using the repeated division-by-8 method.
- Octal-to-binary conversion is accomplished by simply replacing each octal digit with its 3-bit binary equivalent. The process is reversed for binary-to-octal conversion.
- A decimal number is converted to BCD by replacing each decimal digit with the appropriate 4-bit binary code.
- The ASCII is a 7-bit alphanumeric code that is used in computer systems for input and output of information.
- A parity bit is used to detect an error in a code.
- The CRC (cyclic redundancy check) is based on polynomial division using modulo-2 operations.

00 00 00 00
00 00 10 00
00 11 11 11
11 11 11 11
11 11 11 01
11 01 01 01
01 01 01 01
01 10 00 01
10 01 00 01
01 01 11 00
01 00 11 10
00 10 11 10
10 10 01 00
10 00 01 11
00 11 10 11

KEY TERMS

Key terms and other bold terms in the chapter are defined in the end-of-book glossary.

Alphanumeric Consisting of numerals, letters, and other characters.

ASCII American Standard Code for Information Interchange; the most widely used alphanumeric code.

BCD Binary coded decimal; a digital code in which each of the decimal digits, 0 through 9, is represented by a group of four bits.

Byte A group of eight bits.

Cyclic redundancy check (CRC) A type of error detection code.

Floating-point number A number representation based on scientific notation in which the number consists of an exponent and a mantissa.

Hexadecimal Describes a number system with a base of 16.

LSB Least significant bit; the right-most bit in a binary whole number or code.

MSB Most significant bit; the left-most bit in a binary whole number or code.

Octal Describes a number system with a base of eight.

Parity In relation to binary codes, the condition of evenness or oddness of the number of 1s in a code group.

TRUE/FALSE QUIZ

Answers are at the end of the chapter.

1. The octal number system is a weighted system with eight digits.
2. The binary number system is a weighted system with two digits.
3. MSB stands for most significant bit.
4. In hexadecimal, $9 + 1 = 10$.
5. The 1's complement of the binary number 1010 is 0101.
6. The 2's complement of the binary number 1111 is 0000.
7. The right-most bit in a signed binary number is the sign bit.
8. The hexadecimal number system has 16 characters, six of which are alphabetic characters.
9. BCD stands for binary coded decimal.
10. An error in a given code can be detected by verifying the parity bit.
11. CRC stands for cyclic redundancy check.
12. The modulo-2 sum of 11 and 10 is 100.

SELF-TEST

Answers are at the end of the chapter.

1. $3 \times 10^1 + 4 \times 10^0$ is
 (a) 0.34 (b) 3.4 **(c) 34** (d) 340
2. The decimal equivalent of 1000 is
 (a) 2 (b) 4 (c) 6 **(d) 8**
3. The binary number 11011101 is equal to the decimal number
 (a) 121 **(b) 221** (c) 441 (d) 256
4. The decimal number 21 is equivalent to the binary number
(a) 10101 (b) 10001 (c) 10000 (d) 11111
5. The decimal number 250 is equivalent to the binary number
(a) 11111010 (b) 11110110 (c) 11111000 (d) 11111011
6. The sum of 1111 + 1111 in binary equals
 (a) 0000 (b) 2222 **(c) 11110** (d) 11111

7. The difference of $1000 - 100$ equals
 (a) 100 (b) 101 (c) 110 (d) 111
8. The 1's complement of 11110000 is
 (a) 11111111 (b) 11111110 (c) 00001111 (d) 10000001
9. The 2's complement of 11001100 is
 (a) 00110011 (b) 00110100 (c) 00110101 (d) 00110110
10. The decimal number +122 is expressed in the 2's complement form as
 (a) 01111010 (b) 11111010 (c) 01000101 (d) 10000101
11. The decimal number -34 is expressed in the 2's complement form as
 (a) 01011110 (b) 10100010 (c) 11011110 (d) 01011101
12. A single-precision floating-point binary number has a total of
 (a) 8 bits (b) 16 bits (c) 24 bits (d) 32 bits
13. In the 2's complement form, the binary number 10010011 is equal to the decimal number
 (a) -19 (b) +109 (c) +91 (d) -109
14. The binary number 10110011100101010001 can be written in octal as
 (a) 5471230₈ (b) 5471241₈ (c) 2634521₈ (d) 23162501₈
15. The binary number 10001101010001101111 can be written in hexadecimal as
 (a) AD467₁₆ (b) 8C46F₁₆ (c) 8D46F₁₆ (d) AE46F₁₆
16. The binary number for F7A9₁₆ is
 (a) 1111011110101001 (b) 1110111110101001
 (c) 111111010110001 (d) 1111011010101001
17. The BCD number for decimal 473 is
 (a) 110011010 (b) 110001110011 (c) 010001110011 (d) 010011110011
18. Refer to Table 2-7. The command STOP in ASCII is
 (a) 101001110101001001111010000 (b) 1010010100110010011101010000
 (c) 1001010110110011101010001 (d) 1010011101010010011101100100
19. The code that has an even-parity error is
 (a) 1010011 (b) 1101000 (c) 1001000 (d) 1110111
20. In the cyclic redundancy check, the absence of errors is indicated by
 (a) Remainder = generator code (b) Remainder = 0
 (c) Remainder = 1 (d) Quotient = 0

PROBLEMS

Answers to odd-numbered problems are at the end of the book.

Section 2-1 Decimal Numbers

- What is the weight of 7 in each of the following decimal numbers?
 (a) 1947 (b) 1799 (c) 1979
- Express each of the following decimal numbers as a power of ten:
 (a) 1000 (b) 10000000 (c) 1000000000
- Give the value of each digit in the following decimal numbers:
 (a) 263 (b) 5436 (c) 234543
- How high can you count with six decimal digits?

Section 2-2 Binary Numbers

- Convert the following binary numbers to decimal:
 (a) 001 (b) 010 (c) 101 (d) 110
 (e) 1010 (f) 1011 (g) 1110 (h) 1111
- Convert the following binary numbers into decimal:
 (a) 100001 (b) 100111 (c) 101010 (d) 111001
 (e) 1100000 (f) 1111101 (g) 11110010 (h) 11111111

7. Convert each binary number to decimal:

(a) 110011.11 (b) 101010.01 (c) 1000001.111
(d) 1111000.101 (e) 1011100.10101 (f) 1110001.00001
(g) 1011010.1010 (h) 1111111.11111

8. What is the highest decimal number that can be represented by each of the following numbers of binary digits (bits)?

(a) two (b) three (c) four (d) five (e) six
(f) seven (g) eight (h) nine (i) ten (j) eleven

9. How many bits are required to represent the following decimal numbers?

(a) 5 (b) 10 (c) 15 (d) 20
(e) 100 (f) 120 (g) 140 (h) 160

10. Generate the binary sequence for each decimal sequence:

(a) 0 through 7 (b) 8 through 15 (c) 16 through 31
(d) 32 through 63 (e) 64 through 75

Section 2–3 Decimal-to-Binary Conversion

- 11.** Convert each decimal number to binary by using the sum-of-weights method:

(a) 12 (b) 15 (c) 25 (d) 50
(e) 65 (f) 97 (g) 127 (h) 198

12. Convert each decimal fraction to binary using the sum-of-weights method:

(a) 0.26 (b) 0.762 (c) 0.0975

13. Convert each decimal number to binary using repeated division by 2:

(a) 13 (b) 17 (c) 23 (d) 30
(e) 35 (f) 40 (g) 49 (h) 60

14. Convert each decimal fraction to binary using repeated multiplication by 2:

(a) 0.76 (b) 0.456 (c) 0.8732

Section 2–4 Binary Arithmetic

15. Add the binary numbers:

(a) $10 + 10$ (b) $10 + 11$ (c) $100 + 11$
(d) $111 + 101$ (e) $1111 + 111$ (f) $1111 + 1111$

16. Use direct subtraction on the following binary numbers:

(a) $10 - 1$ (b) $100 - 11$ (c) $110 - 100$
(d) $1111 - 11$ (e) $1101 - 101$ (f) $110000 - 1111$

17. Perform the following binary multiplications:

(a) 11×10 (b) 101×11 (c) 111×110
(d) 1100×101 (e) 1110×1110 (f) 1111×1100

18. Divide the binary numbers as indicated:

(a) $110 \div 11$ (b) $1010 \div 10$ (c) $1111 \div 101$

Section 2-5 Complements of Binary Numbers

- 19** What are two ways of representing zero in 1's complement form?

20 How is zero represented in 2's complement form?

21. Determine the 1's complement of each binary number:

(a) 100 (b) 111 (c) 1100
(d) 10111011 (e) 1001010 (f) 10101010

22. Determine the 2's complement of each binary number using either method:

(a) 11 (b) 110 (c) 1010 (d) 1001
(e) 101010 (f) 11001 (g) 11001100 (h) 11000111

Section 2-6 Signed Numbers

23. Express each decimal number in binary as an 8-bit sign-magnitude number:
 (a) +29 (b) -85 (c) +100 (d) -123
24. Express each decimal number as an 8-bit number in the 1's complement form:
 (a) -34 (b) +57 (c) -99 (d) +115
25. Express each decimal number as an 8-bit number in the 2's complement form:
 (a) +12 (b) -68 (c) +101 (d) -125
26. Determine the decimal value of each signed binary number in the sign-magnitude form:
 (a) 10011001 (b) 01110100 (c) 10111111
27. Determine the decimal value of each signed binary number in the 1's complement form:
 (a) 10011001 (b) 01110100 (c) 10111111
28. Determine the decimal value of each signed binary number in the 2's complement form:
 (a) 10011001 (b) 01110100 (c) 10111111
29. Express each of the following sign-magnitude binary numbers in single-precision floating-point format:
 (a) 011110000101011 (b) 100110000011000
30. Determine the values of the following single-precision floating-point numbers:
 (a) 1 10000001 01001001110001000000000
 (b) 0 11001100 10000111101001000000000

01	00	00	00	00	00	00	00
00	00	00	10	00	00	00	00
00	11	11	11	11	11	11	11
11	11	11	00	00	00	00	00
11	11	11	11	11	11	11	11
11	01	01	01	01	01	01	01
11	01	01	01	01	01	01	01
01	01	01	01	01	01	01	01
01	10	00	00	00	00	00	00
10	00	01	00	01	01	01	01
01	01	11	00	00	00	00	00
01	00	11	10	00	00	00	00
00	10	11	10	00	00	00	00
10	10	01	10	00	00	00	00
10	00	01	11	11	11	11	11
00	11	10	11	11	11	11	11

Section 2-7 Arithmetic Operations with Signed Numbers

31. Convert each pair of decimal numbers to binary and add using the 2's complement form:
 (a) 33 and 15 (b) 56 and -27 (c) -46 and 25 (d) -110 and -84
32. Perform each addition in the 2's complement form:
 (a) 00010110 + 00110011 (b) 01110000 + 10101111
33. Perform each addition in the 2's complement form:
 (a) 10001100 + 00111001 (b) 11011001 + 11100111
34. Perform each subtraction in the 2's complement form:
 (a) 00110011 - 00010000 (b) 01100101 - 11101000
35. Multiply 01101010 by 11110001 in the 2's complement form.
36. Divide 10001000 by 00100010 in the 2's complement form.

Section 2-8 Hexadecimal Numbers

37. Convert each hexadecimal number to binary:
 (a) 46_{16} (b) 54_{16} (c) $B4_{16}$ (d) $1A3_{16}$
 (e) FA_{16} (f) ABC_{16} (g) $ABCD_{16}$
38. Convert each binary number to hexadecimal:
 (a) 1111 (b) 1011 (c) 11111
 (d) 10101010 (e) 10101100 (f) 10111011
39. Convert each hexadecimal number to decimal:
 (a) 42_{16} (b) 64_{16} (c) $2B_{16}$ (d) $4D_{16}$
 (e) FF_{16} (f) BC_{16} (g) $6F1_{16}$ (h) ABC_{16}
40. Convert each decimal number to hexadecimal:
 (a) 10 (b) 15 (c) 32 (d) 54
 (e) 365 (f) 3652 (g) 7825 (h) 8925
41. Perform the following additions:
 (a) $25_{16} + 33_{16}$ (b) $43_{16} + 62_{16}$ (c) $A4_{16} + F5_{16}$ (d) $FC_{16} + AE_{16}$
42. Perform the following subtractions:
 (a) $60_{16} - 39_{16}$ (b) $A5_{16} - 98_{16}$ (c) $F1_{16} - A6_{16}$ (d) $AC_{16} - 10_{16}$

Section 2-9 Octal Numbers

43. Convert each octal number to decimal:

- (a) 14_8 (b) 53_8 (c) 67_8 (d) 174_8
 (e) 635_8 (f) 254_8 (g) 2673_8 (h) 7777_8

44. Convert each decimal number to octal by repeated division by 8:

- (a) 23 (b) 45 (c) 65 (d) 84
 (e) 124 (f) 156 (g) 654 (h) 9999

45. Convert each octal number into binary:

- (a) 17_8 (b) 26_8 (c) 145_8 (d) 456_8
 (e) 653_8 (f) 777_8

46. Convert each binary number to octal:

- (a) 100 (b) 110 (c) 1100
 (d) 1111 (e) 11001 (f) 11110
 (g) 110011 (h) 101010 (i) 10101111

Section 2-10 Binary Coded Decimal (BCD)

47. Convert each of the following decimal numbers to 8421 BCD:

- (a) 10 (b) 13 (c) 18 (d) 21 (e) 25 (f) 36
 (g) 44 (h) 57 (i) 69 (j) 98 (k) 125 (l) 156

48. Convert each of the decimal numbers in Problem 47 to straight binary, and compare the number of bits required with that required for BCD.

49. Convert the following decimal numbers to BCD:

- (a) 104 (b) 128 (c) 132 (d) 150 (e) 186
 (f) 210 (g) 359 (h) 547 (i) 1051

50. Convert each of the BCD numbers to decimal:

- (a) 0001 (b) 0110 (c) 1001
 (d) 00011000 (e) 00011001 (f) 00110010
 (g) 01000101 (h) 10011000 (i) 100001110000

51. Convert each of the BCD numbers to decimal:

- (a) 10000000 (b) 001000110111
 (c) 001101000110 (d) 010000100001
 (e) 011101010100 (f) 100000000000
 (g) 100101111000 (h) 0001011010000011
 (i) 1001000000011000 (j) 0110011001100111

52. Add the following BCD numbers:

- (a) 0010 + 0001 (b) 0101 + 0011
 (c) 0111 + 0010 (d) 1000 + 0001
 (e) 00011000 + 00010001 (f) 01100100 + 00110011
 (g) 01000000 + 01000111 (h) 10000101 + 00010011

53. Add the following BCD numbers:

- (a) 1000 + 0110 (b) 0111 + 0101
 (c) 1001 + 1000 (d) 1001 + 0111
 (e) 00100101 + 00100111 (f) 01010001 + 01011000
 (g) 10011000 + 10010111 (h) 010101100001 + 011100001000

54. Convert each pair of decimal numbers to BCD, and add as indicated:

- (a) 4 + 3 (b) 5 + 2 (c) 6 + 4 (d) 17 + 12
 (e) 28 + 23 (f) 65 + 58 (g) 113 + 101 (h) 295 + 157

Section 2-11 Digital Codes

55. In a certain application a 4-bit binary sequence cycles from 1111 to 0000 periodically. There are four bit changes, and because of circuit delays, these changes may not occur at the same

instant. For example, if the LSB changes first, the number will appear as 1110 during the transition from 1111 to 0000 and may be misinterpreted by the system. Illustrate how the Gray code avoids this problem.

56. Convert each binary number to Gray code:

- (a) 11011 (b) 1001010 (c) 1111011101110

57. Convert each Gray code to binary:

- (a) 1010 (b) 00010 (c) 11000010001

58. Convert each of the following decimal numbers to ASCII. Refer to Table 2–7.

- (a) 1 (b) 3 (c) 6 (d) 10 (e) 18
 (f) 29 (g) 56 (h) 75 (i) 107

59. Determine each ASCII character. Refer to Table 2–7.

- (a) 0011000 (b) 1001010 (c) 0111101
 (d) 0100011 (e) 0111110 (f) 1000010

60. Decode the following ASCII coded message:

1001000 1100101 1101100 1101100 1101111 0101110
 0100000 1001000 1101111 1101111 0100000 1100001
 1110010 1100101 0100000 1111001 1101111 1110101
 0111111

61. Write the message in Problem 60 in hexadecimal.

62. Convert the following statement to ASCII:

30 INPUT A, B

Section 2–12 Error Codes

63. Determine which of the following even parity codes are in error:

- (a) 100110010 (b) 011101010 (c) 1011111010001010

64. Determine which of the following odd parity codes are in error:

- (a) 11110110 (b) 00110001 (c) 01010101010101010

65. Attach the proper even parity bit to each of the following bytes of data:

- (a) 10100100 (b) 00001001 (c) 11111110

66. Apply modulo-2 to the following:

- (a) 1100 + 1011 (b) 1111 + 0100 (c) 10011001 + 100011100

67. Verify that modulo-2 subtraction is the same as modulo-2 addition by adding the result of each operation in problem 66 to either of the original numbers to get the other number. This will show that the result is the same as the difference of the two numbers.

68. Apply CRC to the data bits 10110010 using the generator code 1010 to produce the transmitted CRC code.

69. Assume that the code produced in problem 68 incurs an error in the most significant bit during transmission. Apply CRC to detect the error.

ANSWERS

SECTION CHECKUPS

Section 2–1 Decimal Numbers

1. (a) 1370: 10 (b) 6725: 100 (c) 7051: 1000 (d) 58.72: 0.1

2. (a) $51 = (5 \times 10) + (1 \times 1)$

(b) $137 = (1 \times 100) + (3 \times 10) + (7 \times 1)$

(c) $1492 = (1 \times 1000) + (4 \times 100) + (9 \times 10) + (2 \times 1)$

(d) $106.58 = (1 \times 100) + (0 \times 10) + (6 \times 1) + (5 \times 0.1) + (8 \times 0.01)$

Section 2–2 Binary Numbers

1. $2^8 - 1 = 255$
2. Weight is 16.
3. $10111101.011 = 189.375$

Section 2–3 Decimal-to-Binary Conversion

1. (a) $23 = 10111$ (b) $57 = 111001$ (c) $45.5 = 101101.1$
2. (a) $14 = 1110$ (b) $21 = 10101$ (c) $0.375 = 0.011$

Section 2–4 Binary Arithmetic

1. (a) $1101 + 1010 = 10111$ (b) $10111 + 01101 = 100100$
2. (a) $1101 - 0100 = 1001$ (b) $1001 - 0111 = 0010$
3. (a) $110 \times 111 = 101010$ (b) $1100 \div 011 = 100$

Section 2–5 Complements of Binary Numbers

1. (a) 1's comp of $00011010 = 11100101$ (b) 1's comp of $11110111 = 00001000$
(c) 1's comp of $10001101 = 01110010$
2. (a) 2's comp of $00010110 = 11101010$ (b) 2's comp of $11111100 = 00000100$
(c) 2's comp of $10010001 = 01101111$

Section 2–6 Signed Numbers

1. Sign-magnitude: $+9 = 00001001$
2. 1's comp: $-33 = 11011110$
3. 2's comp: $-46 = 11010010$
4. Sign bit, exponent, and mantissa

Section 2–7 Arithmetic Operations with Signed Numbers

1. Cases of addition: positive number is larger, negative number is larger, both are positive, both are negative
2. $00100001 + 10111100 = 11011101$
3. $01110111 - 00110010 = 01000101$
4. Sign of product is positive.
5. $00000101 \times 01111111 = 0100111011$
6. Sign of quotient is negative.
7. $00110000 \div 00001100 = 00000100$

Section 2–8 Hexadecimal Numbers

1. (a) $10110011 = B3_{16}$ (b) $110011101000 = CE8_{16}$
2. (a) $57_{16} = 01010111$ (b) $3A5_{16} = 001110100101$
(c) $F8OB_{16} = 1111100000001011$
3. $9B30_{16} = 39,728_{10}$
4. $573_{10} = 23D_{16}$
5. (a) $18_{16} + 34_{16} = 4C_{16}$ (b) $3F_{16} + 2A_{16} = 69_{16}$
6. (a) $75_{16} - 21_{16} = 54_{16}$ (b) $94_{16} - 5C_{16} = 38_{16}$

Section 2–9 Octal Numbers

1. (a) $73_8 = 59_{10}$ (b) $125_8 = 85_{10}$
2. (a) $98_{10} = 142_8$ (b) $163_{10} = 243_8$

3. (a) $46_8 = 100110$ (b) $723_8 = 111010011$ (c) $5624_8 = 101110010100$
 4. (a) $110101111 = 657_8$ (b) $1001100010 = 1142_8$ (c) $1011111001 = 2771_8$

Section 2-10 Binary Coded Decimal (BCD)

1. (a) 0010: 2 (b) 1000: 8 (c) 0001: 1 (d) 0100: 4
 2. (a) $6_{10} = 0110$ (b) $15_{10} = 00010101$ (c) $273_{10} = 001001110011$
 (d) $849_{10} = 100001001001$
 3. (a) $10001001 = 89_{10}$ (b) $001001111000 = 278_{10}$ (c) $000101010111 = 157_{10}$
 4. A 4-bit sum is invalid when it is greater than 9_{10} .

Section 2-11 Digital Codes

1. (a) $1100_2 = 1010$ Gray (b) $1010_2 = 1111$ Gray (c) $11010_2 = 10111$ Gray
 2. (a) 1000 Gray = 1111_2 (b) 1010 Gray = 1100_2 (c) 11101 Gray = 10110_2
 3. (a) K: $1001011 \rightarrow 4B_{16}$ (b) r: $1110010 \rightarrow 72_{16}$
 (c) \$: $0100100 \rightarrow 24_{16}$ (d) +: $0101011 \rightarrow 2B_{16}$

Section 2-12 Error Codes

1. (c) 0101 has an error.
 2. (d) 11111011 has an error.
 3. (a) 10101001 (b) 01000001 (c) 11101110 (d) 10001101
 4. Cyclic redundancy check
 5. (a) 0 (b) 0 (c) 1 (d) 1

RELATED PROBLEMS FOR EXAMPLES

- 2-1 9 has a value of 900, 3 has a value of 30, 9 has a value of 9.
 2-2 6 has a value of 60, 7 has a value of 7, 9 has a value of $9/10$ (0.9), 2 has a value of $2/100$ (0.02), 4 has a value of $4/1000$ (0.004).
 2-3 $10010001 = 128 + 16 + 1 = 145$
 2-4 $10.111 = 2 + 0.5 + 0.25 + 0.125 = 2.875$
 2-5 $125 = 64 + 32 + 16 + 8 + 4 + 1 = 1111101$
 2-6 $39 = 100111$
 2-7 $1111 + 1100 = 11011$
 2-8 $111 - 100 = 011$
 2-9 $110 - 101 = 001$
 2-10 $1101 \times 1010 = 10000010$
 2-11 $1100 \div 100 = 11$
 2-12 00110101
 2-13 01000000
 2-14 See Table 2-10.

TABLE 2-10

	Sign-Magnitude	1's Comp	2's Comp
+19	00010011	00010011	00010011
-19	10010011	11101100	11101101

- 2-15 $01110111 = +119_{10}$
 2-16 $11101011 = -20_{10}$
 2-17 $11010111 = -41_{10}$

01	00	00	00
00	00	00	00
00	11	11	11
11	11	11	11
11	11	11	11
11	11	11	01
01	01	01	01
01	01	01	10
01	01	00	01
01	01	11	00
01	00	11	10
00	10	11	10
10	10	01	00
10	01	00	01
00	11	10	11

2-18 11000010001010011000000000**2-19** 01010101**2-20** 00010001**2-21** 1001000110**2-22** $(83)(-59) = -4897$ (10110011011111 in 2's comp)**2-23** $100 \div 25 = 4$ (0100)**2-24** 4F79C₁₆**2-25** 0110101111010011₂**2-26** $6BD_{16} = 011010111101 = 2^{10} + 2^9 + 2^7 + 2^5 + 2^4 + 2^3 + 2^2 + 2^0$
 $= 1024 + 512 + 128 + 32 + 16 + 8 + 4 + 1 = 1725_{10}$ **2-27** $60A_{16} = (6 \times 256) + (0 \times 16) + (10 \times 1) = 1546_{10}$ **2-28** 2591₁₀ = A1F₁₆**2-29** 4C₁₆ + 3A₁₆ = 86₁₆**2-30** BCD₁₆ - 173₁₆ = A5A₁₆**2-31** (a) $001011_2 = 11_{10} = 13_8$ (b) $010101_2 = 21_{10} = 25_8$
(c) $001100000_2 = 96_{10} = 140_8$ (d) $111101010110_2 = 3926_{10} = 7526_8$ **2-32** 1250762₈**2-33** 1001011001110011**2-34** 82,276₁₀**2-35** 1001100101101000**2-36** 10000010**2-37** (a) 111011 (Gray) (b) 111010₂**2-38** The sequence of codes for if ($y < 8$) is 69₁₆66₁₆20₁₆28₁₆79₁₆3C₁₆38₁₆29₁₆**2-39** 01001011**2-40** Yes**2-41** A 0 remainder results**2-42** Errors are indicated.

TRUE/FALSE QUIZ

1. T 2. T 3. T 4. F 5. T 6. F 7. F 8. T 9. T 10. T
11. T 12. F

SELF-TEST

1. (c) 2. (d) 3. (b) 4. (a) 5. (a) 6. (c) 7. (a) 8. (c)
9. (b) 10. (a) 11. (c) 12. (d) 13. (d) 14. (b) 15. (c) 16. (a)
17. (c) 18. (a) 19. (b) 20. (b)



Proper grounding is very important when setting up to take measurements or work on a circuit. Properly grounding the oscilloscope protects you from shock and grounding yourself protects your circuits from damage. Grounding the oscilloscope means to connect it to earth ground by plugging the three-prong power cord into a grounded outlet. Grounding yourself means using a wrist-type grounding strap, particularly when you are working with CMOS logic. The wrist strap must have a high-value resistor between the strap and ground for protection against accidental contact with a voltage source.

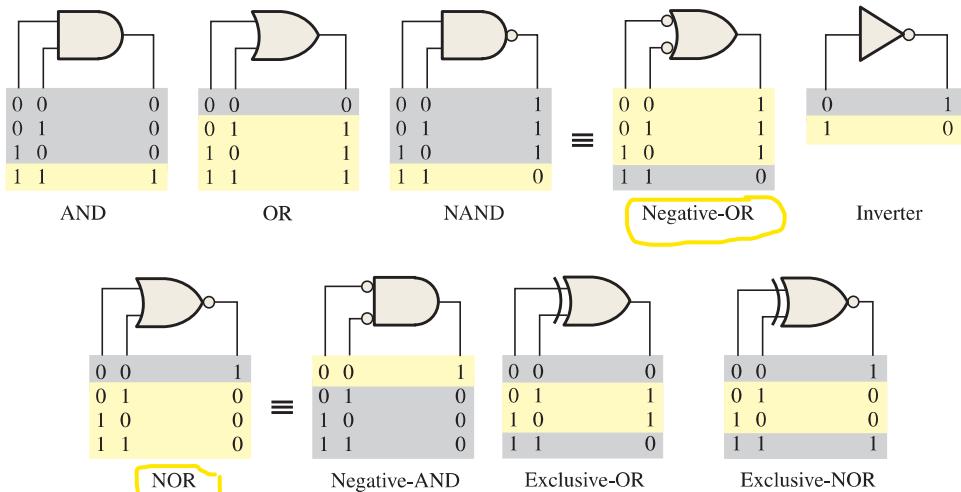
Also, for accurate measurements, make sure that the ground in the circuit you are testing is the same as the scope ground. This can be done by connecting the ground lead on the scope probe to a known ground point in the circuit, such as the metal chassis or a ground point on the PCB. You can also connect the circuit ground to the GND jack on the front panel of the scope.

SECTION 3-9 CHECKUP

1. What are the most common types of failures in ICs?
2. If two different input waveforms are applied to a 2-input bipolar NAND gate and the output waveform is just like one of the inputs, but inverted, what is the most likely problem?
3. Name two characteristics of pulse waveforms that can be measured on the oscilloscope.

SUMMARY

- The inverter output is the complement of the input.
- The AND gate output is HIGH only when all the inputs are HIGH.
- The OR gate output is HIGH when any of the inputs is HIGH.
- The NAND gate output is LOW only when all the inputs are HIGH.
- The NAND can be viewed as a negative-OR whose output is HIGH when any input is LOW.
- The NOR gate output is LOW when any of the inputs is HIGH.
- The NOR can be viewed as a negative-AND whose output is HIGH only when all the inputs are LOW.
- The exclusive-OR gate output is HIGH when the inputs are not the same.
- The exclusive-NOR gate output is LOW when the inputs are not the same.
- Distinctive shape symbols and truth tables for various logic gates (limited to 2 inputs) are shown in Figure 3-75.



Note: Active states are shown in yellow.

FIGURE 3-75

- Most programmable logic devices (PLDs) are based on some form of AND array.
- Programmable link technologies are fuse, antifuse, EEPROM, EEPROM, flash, and SRAM.
- A PLD can be programmed in a hardware fixture called a programmer or mounted on a development printed circuit board.
- PLDs have an associated software development package for programming.
- Two methods of design entry using programming software are text entry (HDL) and graphic (schematic) entry.
- ISP PLDs can be programmed after they are installed in a system, and they can be reprogrammed at any time.
- JTAG stands for Joint Test Action Group and is an interface standard (IEEE Std. 1149.1) used for programming and testing PLDs.
- An embedded processor is used to facilitate in-system programming of PLDs.
- In PLDs, the circuit is programmed in and can be changed by reprogramming.
- The average power dissipation of a logic gate is

$$P_D = V_{CC} \left(\frac{I_{CCH} + I_{CCL}}{2} \right)$$

- The speed-power product of a logic gate is

$$SPP = t_p P_D$$

- As a rule, CMOS has a lower power consumption than bipolar.
- In fixed-function logic, the circuit cannot be altered.

KEY TERMS

Key terms and other bold terms in the chapter are defined in the end-of-book glossary.

AND array An array of AND gates consisting of a matrix of programmable interconnections.

AND gate A logic gate that produces a HIGH output only when all of the inputs are HIGH.

Antifuse A type of PLD nonvolatile programmable link that can be left open or can be shorted once as directed by the program.

Bipolar A class of integrated logic circuits implemented with bipolar transistors; also known as TTL.

Boolean algebra The mathematics of logic circuits.

CMOS Complementary metal-oxide semiconductor; a class of integrated logic circuits that is implemented with a type of field-effect transistor.

Complement The inverse or opposite of a number. LOW is the complement of HIGH, and 0 is the complement of 1.

EEPROM A type of nonvolatile PLD reprogrammable link based on electrically erasable programmable read-only memory cells and can be turned on or off repeatedly by programming.

EPROM A type of PLD nonvolatile programmable link based on electrically programmable read-only memory cells and can be turned either on or off once with programming.

Exclusive-NOR (XNOR) gate A logic gate that produces a LOW only when the two inputs are at opposite levels.

Exclusive-OR (XOR) gate A logic gate that produces a HIGH output only when its two inputs are at opposite levels.

Fan-out The number of equivalent gate inputs of the same family series that a logic gate can drive.

Flash A type of PLD nonvolatile reprogrammable link technology based on a single transistor cell.

Fuse A type of PLD nonvolatile programmable link that can be left shorted or can be opened once as directed by the program.

Inverter A logic circuit that inverts or complements its input.

JTAG Joint Test Action Group; an interface standard designated IEEE Std. 1149.1.

NAND gate A logic gate that produces a LOW output only when all the inputs are HIGH.

00	00	11
10	00	11
11	11	11
00	11	11
11	01	01
00	01	01
11	01	10
01	01	10
00	10	01
00	01	01
11	01	00
11	00	10
11	10	10
01	10	00
01	00	11
01	11	01
10	11	01

NOR gate A logic gate in which the output is LOW when one or more of the inputs are HIGH.

OR gate A logic gate that produces a HIGH output when one or more inputs are HIGH.

Propagation delay time The time interval between the occurrence of an input transition and the occurrence of the corresponding output transition in a logic circuit.

SRAM A type of PLD volatile reprogrammable link based on static random-access memory cells and can be turned on or off repeatedly with programming.

Target device A PLD mounted on a programming fixture or development board into which a software logic design is to be downloaded.

Truth table A table showing the inputs and corresponding output(s) of a logic circuit.

Unit load A measure of fan-out. One gate input represents one unit load to the output of a gate within the same IC family.

VHDL A standard hardware description language that describes a function with an entity/architecture structure.

TRUE/FALSE QUIZ

Answers are at the end of the chapter.

- ✓ 1. An inverter performs a NOT operation.
- ✓ 2. A NOT gate cannot have more than one input.
- ✗ 3. If any input to an OR gate is zero, the output is zero.
- ✗ 4. If all inputs to an AND gate are 1, the output is 0.
- ✓ 5. A NAND gate can be considered as an AND gate followed by a NOT gate.
- ✓ 6. A NOR gate can be considered as an OR gate followed by an inverter.
- ✗ 7. The output of an exclusive-OR is 0 if the inputs are opposite.
- 8. Two types of fixed-function logic integrated circuits are bipolar and NMOS.
- 9. Once programmed, PLD logic can be changed.
- 10. Fan-out is the number of similar gates that a given gate can drive.

SELF-TEST

Answers are at the end of the chapter.

1. When the input to an inverter is LOW (0), the output is
 (a) HIGH or 0 (b) LOW or 0 (c) HIGH or 1 (d) LOW or 1
2. An inverter performs an operation known as
 (a) complementation (b) assertion (c) inversion (d) both answers (a) and (c)
3. The output of an AND gate with inputs A , B and C is 0 (LOW) when
 (a) $A = 0, B = 0, C = 0$ (b) $A = 0, B = 1, C = 1$ (c) both answers (a) and (b)
4. The output of an OR gate with inputs A , B and C is 0 (LOW) when
 (a) $A = 0, B = 0, C = 0$ (b) $A = 0, B = 1, C = 1$ (c) both answers (a) and (b)
5. A pulse is applied to each input of a 2-input NAND gate. One pulse goes HIGH at $t = 0$ and goes back LOW at $t = 1$ ms. The other pulse goes HIGH at $t = 0.8$ ms and goes back LOW at $t = 3$ ms. The output pulse can be described as follows:
 (a) It goes LOW at $t = 0$ and back HIGH at $t = 3$ ms.
 (b) It goes LOW at $t = 0.8$ ms and back HIGH at $t = 3$ ms.
 (c) It goes LOW at $t = 0.8$ ms and back HIGH at $t = 1$ ms.
 (d) It goes LOW at $t = 0.8$ ms and back LOW at $t = 1$ ms.
6. A pulse is applied to each input of a 2-input NOR gate. One pulse goes HIGH at $t = 0$ and goes back LOW at $t = 1$ ms. The other pulse goes HIGH at $t = 0.8$ ms and goes back LOW at $t = 3$ ms. The output pulse can be described as follows:
 (a) It goes LOW at $t = 0$ and back HIGH at $t = 3$ ms.
 (b) It goes LOW at $t = 0.8$ ms and back HIGH at $t = 3$ ms.
 (c) It goes LOW at $t = 0.8$ ms and back HIGH at $t = 1$ ms.
 (d) It goes HIGH at $t = 0.8$ ms and back LOW at $t = 1$ ms.

- 7.** A pulse is applied to each input of an exclusive-OR gate. One pulse goes HIGH at $t = 0$ and goes back LOW at $t = 1$ ms. The other pulse goes HIGH at $t = 0.8$ ms and goes back LOW at $t = 3$ ms. The output pulse can be described as follows:
- It goes HIGH at $t = 0$ and back LOW at $t = 3$ ms.
 - It goes HIGH at $t = 0$ and back LOW at $t = 0.8$ ms.
 - It goes HIGH at $t = 1$ ms and back LOW at $t = 3$ ms.
 - both answers (b) and (c)
- 8.** A positive-going pulse is applied to an inverter. The time interval from the leading edge of the input to the leading edge of the output is 7 ns. This parameter is
- speed-power product
 - propagation delay, t_{PHL}
 - propagation delay, t_{PLH}
 - pulse width
- 9.** Most PLDs utilize an array of
- NOT gates
 - NOR gates
 - OR gates
 - AND gates
- 10.** The rows and columns of the interconnection matrix in an SPLD are connected using
- fuses
 - switches
 - gates
 - transistors
- 11.** An antifuse is formed using
- two insulators separated by a conductor
 - two conductors separated by an insulator
 - an insulator packed beside a conductor
 - two conductors connected in a series
- 12.** An EPROM can be programmed using
- transistors
 - diodes
 - a multiprogrammer
 - a device programmer
- 13.** Two ways to enter a logic design using PLD development software are
- text and numeric
 - text and graphic
 - graphic and coded
 - compile and sort
- 14.** JTAG stands for
- Joint Test Action Group
 - Java Top Array Group
 - Joint Test Array Group
 - Joint Time Analysis Group
- 15.** In-system programming of a PLD typically utilizes
- an embedded clock generator
 - an embedded processor
 - an embedded PROM
 - both (a) and (b)
 - both (b) and (c)
- 16.** To measure the period of a pulse waveform, you must use
- a DMM
 - a logic probe
 - an oscilloscope
 - a logic pulser
- 17.** Once you measure the period of a pulse waveform, the frequency is found by
- using another setting
 - measuring the duty cycle
 - finding the reciprocal of the period
 - using another type of instrument

PROBLEMS

Answers to odd-numbered problems are at the end of the book.

Section 3-1 The Inverter

- 1.** The input waveform shown in Figure 3-76 is applied to a system of two inverters connected in a series. Draw the output waveform across each inverter in proper relation to the input.



FIGURE 3-76

00	00	00	11
10	00	11	11
11	11	11	11
00	11	11	01
11	11	01	01
01	01	01	10
01	01	10	01
00	10	01	01
00	01	01	00
11	01	00	10
11	00	10	10
11	10	10	00
01	10	00	11
01	11	01	01
10	11	01	01

2. A combination of inverters is shown in Figure 3–77. If a LOW is applied to point *A*, determine the net output at points *E* and *F*.

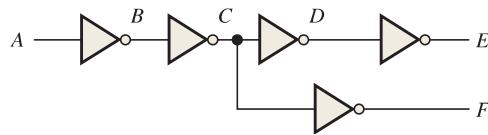


FIGURE 3–77

3. If the waveform in Figure 3–76 is applied to point *A* in Figure 3–77, determine the waveforms at points *B* through *F*.

Section 3–2 The AND Gate

4. Draw the rectangular outline symbol for a 3-input AND gate.
 5. Determine the output, *X*, for a 2-input AND gate with the input waveforms shown in Figure 3–78. Show the proper relationship of output to inputs with a timing diagram.

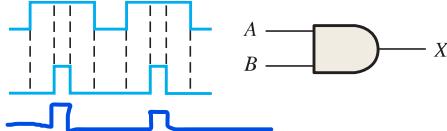


FIGURE 3–78

6. The waveforms in Figure 3–79 are applied to points *A* and *B* of a 2-input AND gate followed by an inverter. Draw the output waveform.

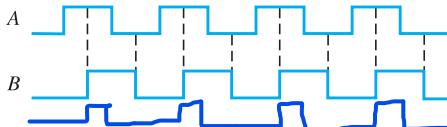


FIGURE 3–79

7. The input waveforms applied to a 3-input AND gate are as indicated in Figure 3–80. Show the output waveform in proper relation to the inputs with a timing diagram.

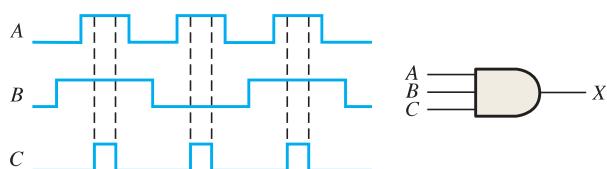


FIGURE 3–80

8. The input waveforms applied to a 4-input AND gate are as indicated in Figure 3–81. The output of the AND gate is fed to an inverter. Draw the net output waveform of this system.

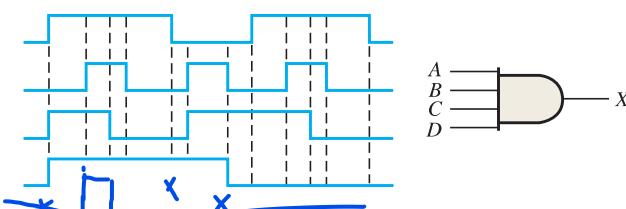


FIGURE 3–81

Section 3-3 The OR Gate

9. Draw the rectangular outline symbol for a 3-input OR gate.
10. Write the expression for a 4-input OR gate with inputs A, B, C, D , and output X .
11. Determine the output for a 2-input OR gate when the input waveforms are as in Figure 3-79 and draw a timing diagram.
12. Repeat Problem 7 for a 3-input OR gate.
13. Repeat Problem 8 for a 4-input OR gate.
14. For the waveforms given in Figure 3-82, A and B are ANDed with output F , D and E are ANDed with output G , and C, F , and G are ORed. Draw the net output waveform.

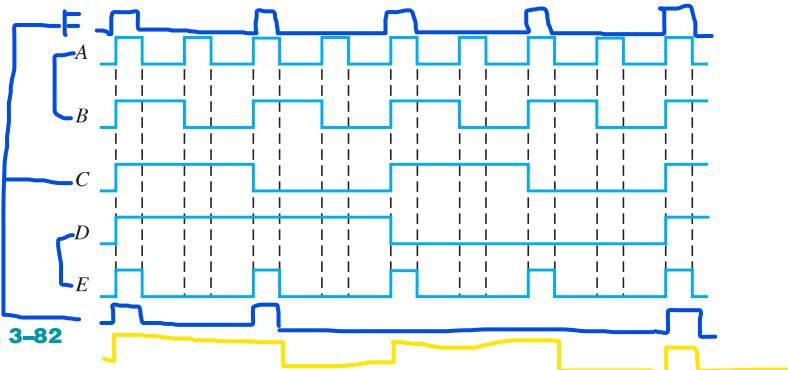


FIGURE 3-82

15. Draw the rectangular outline symbol for a 4-input OR gate.
16. Show the truth table for a system of a 3-input OR gate followed by an inverter.

Section 3-4 The NAND Gate

17. For the set of input waveforms in Figure 3-83, determine the output for the gate shown and draw the timing diagram.

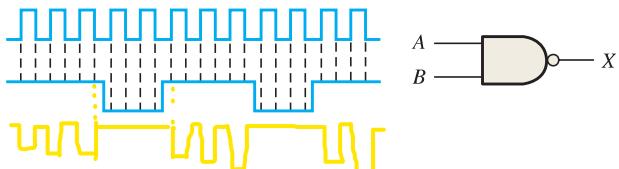


FIGURE 3-83

18. Determine the gate output for the input waveforms in Figure 3-84 and draw the timing diagram.

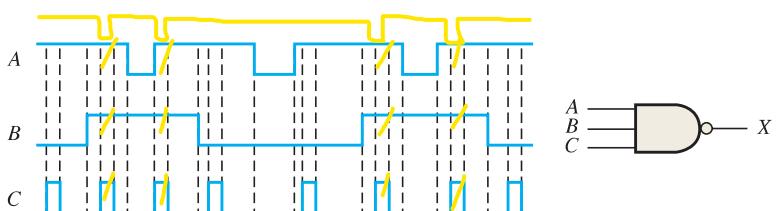


FIGURE 3-84

19. Determine the output waveform in Figure 3-85.

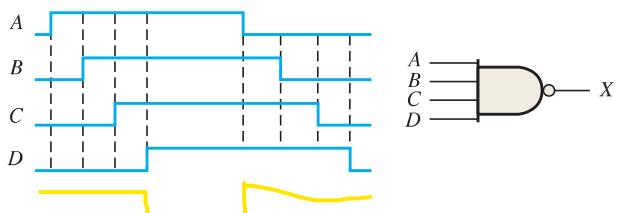


FIGURE 3-85

20. As you have learned, the two logic symbols shown in Figure 3–86 represent equivalent operations. The difference between the two is strictly from a functional viewpoint. For the NAND symbol, look for two HIGHS on the inputs to give a LOW output. For the negative-OR, look for at least one LOW on the inputs to give a HIGH on the output. Using these two functional viewpoints, show that each gate will produce the same output for the given inputs.

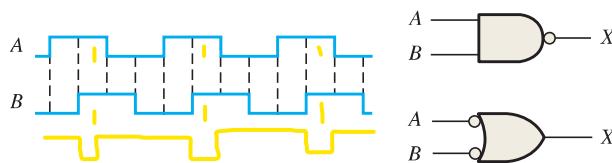


FIGURE 3-86

Section 3-5 The NOR Gate

21. Repeat Problem 17 for a 2-input NOR gate.
22. Determine the output waveform in Figure 3–87 and draw the timing diagram.

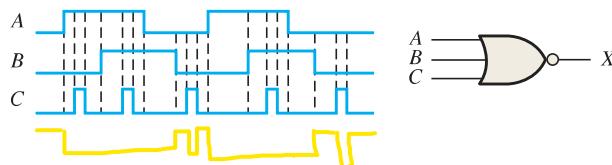


FIGURE 3-87

23. Repeat Problem 19 for a 4-input NOR gate.

24. The NAND and the negative-OR symbols represent equivalent operations, but they are functionally different. For the NOR symbol, look for at least one HIGH on the inputs to give a LOW on the output. For the negative-AND, look for two LOWs on the inputs to give a HIGH output. Using these two functional points of view, show that both gates in Figure 3–88 will produce the same output for the given inputs.

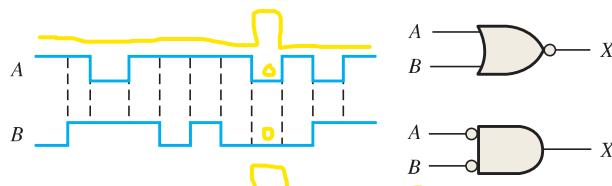


FIGURE 3-88

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

25. How does an exclusive-OR gate differ from an OR gate in its logical operation?
26. Repeat Problem 17 for an exclusive-OR gate.
27. Repeat Problem 17 for an exclusive-NOR gate.
28. Determine the output of an exclusive-NOR gate for the inputs shown in Figure 3–79 and draw a timing diagram.

Section 3-7 Programmable Logic

29. In the simple programmed AND array with programmable links in Figure 3–89, determine the Boolean output expressions.

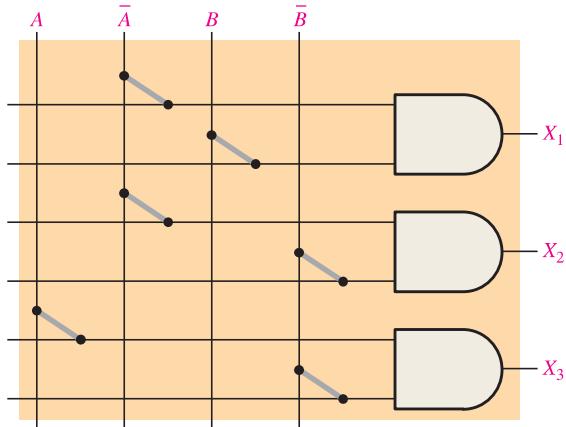


FIGURE 3-89

30. Determine by row and column number which fusible links must be blown in the programmable AND array of Figure 3–90 to implement each of the following product terms:
 $X_1 = \bar{A}BC$, $X_2 = AB\bar{C}$, $X_3 = \bar{A}B\bar{C}$.

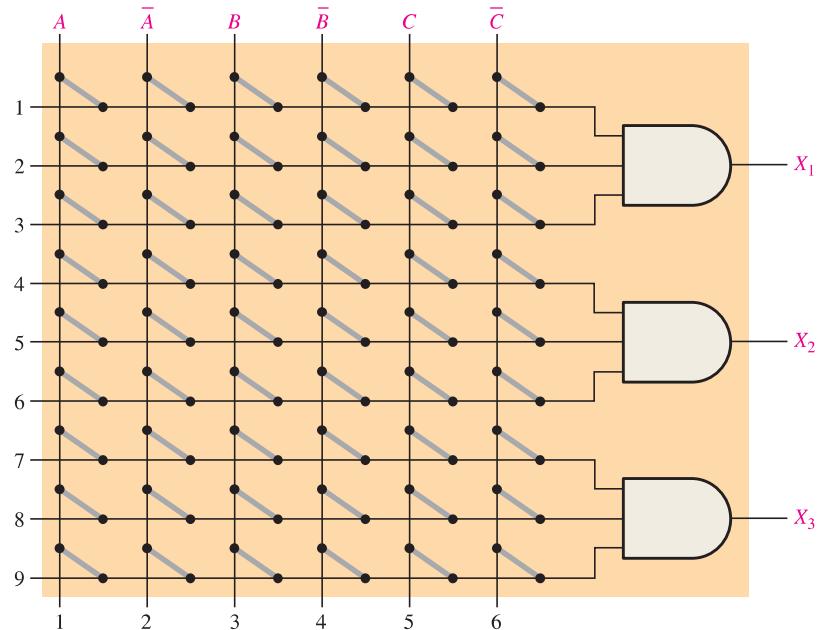


FIGURE 3-90

31. Describe a 4-input AND gate using VHDL.
 32. Describe a 5-input NOR gate using VHDL.

Section 3–8 Fixed-Function Logic Gates

33. In the comparison of certain logic devices, it is noted that the power dissipation for one particular type increases as the frequency increases. Is the device bipolar or CMOS?
34. Using the data sheets in Figures 3–65 and 3–66, determine the following:
- 74LS00 power dissipation at maximum supply voltage and a 50% duty cycle
 - Minimum HIGH level output voltage for a 74LS00
 - Maximum propagation delay for a 74LS00
 - Maximum LOW level output voltage for a 74HC00A
 - Maximum propagation delay for a 74HC00A
35. Determine t_{PLH} and t_{PHL} from the oscilloscope display in Figure 3–91. The readings indicate volts/div and sec/div for each channel.

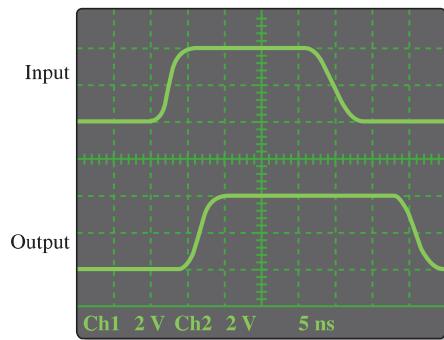


FIGURE 3–91

36. Gate A has $t_{PLH} = t_{PHL} = 6$ ns. Gate B has $t_{PLH} = t_{PHL} = 10$ ns. Which gate can be operated at a higher frequency?
37. If a logic gate operates on a dc supply voltage of +5 V and draws an average current of 4 mA, what is its power dissipation?
38. The variable I_{CCH} represents the dc supply current from V_{CC} when all outputs of an IC are HIGH. The variable I_{CCL} represents the dc supply current when all outputs are LOW. For a 74LS00 IC, determine the typical power dissipation when all four gate outputs are HIGH. (See data sheet in Figure 3–66.)

Section 3–9 Troubleshooting

39. Examine the conditions indicated in Figure 3–92, and identify the faulty gates.

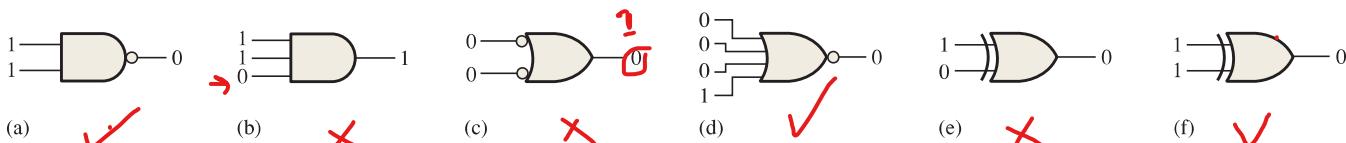


FIGURE 3–92

40. Determine the faulty gates in Figure 3–93 by analyzing the timing diagrams.

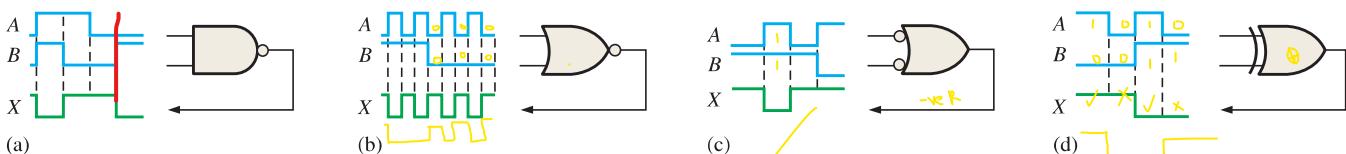


FIGURE 3–93

- 41.** Using an oscilloscope, you make the observations indicated in Figure 3–94. For each observation determine the most likely gate failure.

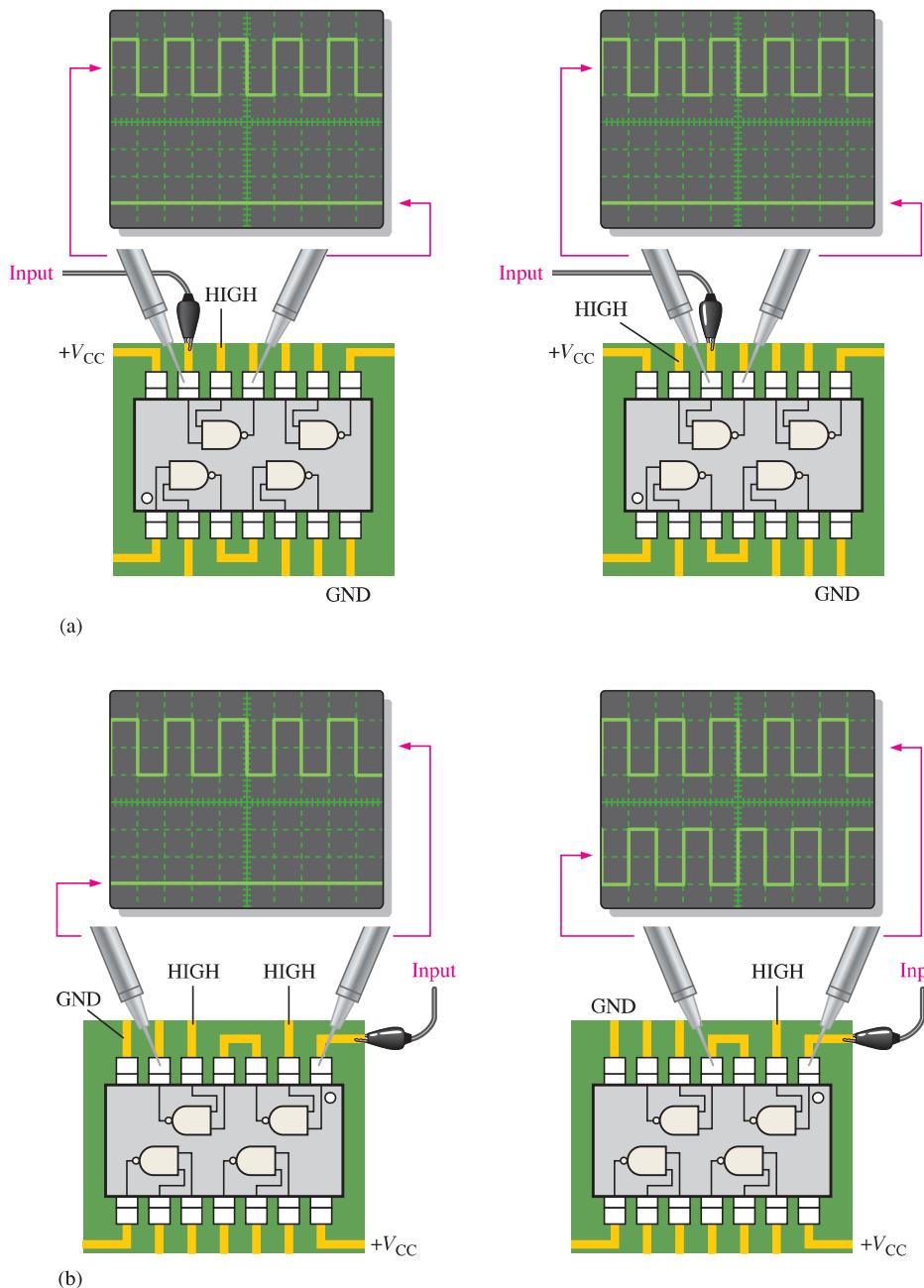


FIGURE 3-94

- 42.** The seat belt alarm circuit in Figure 3–17 has malfunctioned. You find that when the ignition switch is turned on and the seat belt is unbuckled, the alarm comes on and will not go off. What is the most likely problem? How do you troubleshoot it?
- 43.** Every time the ignition switch is turned on in the circuit of Figure 3–17, the alarm comes on for thirty seconds, even when the seat belt is buckled. What is the most probable cause of this malfunction?
- 44.** What failure(s) would you suspect if the output of a 3-input NAND gate stays HIGH no matter what the inputs are?



Special Design Problems

45. Modify the frequency counter in Figure 3–16 to operate with an enable pulse that is active-LOW rather than HIGH during the 1 ms interval.
46. Assume that the enable signal in Figure 3–16 has the waveform shown in Figure 3–95. Assume that waveform *B* is also available. Devise a circuit that will produce an active-HIGH reset pulse to the counter only during the time that the enable signal is LOW.

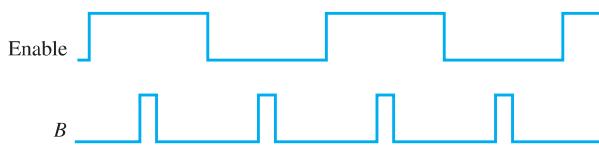


FIGURE 3-95

47. Design a circuit to fit in the beige block of Figure 3–96 that will cause the headlights of an automobile to be turned off automatically 15 s after the ignition switch is turned off, if the light switch is left on. Assume that a LOW is required to turn the lights off.

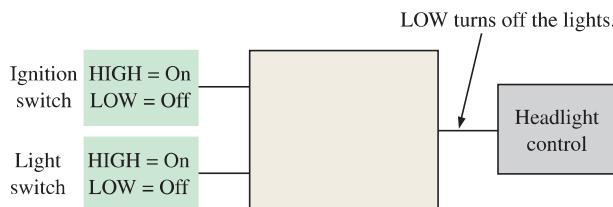


FIGURE 3-96

48. Modify the logic circuit for the intrusion alarm in Figure 3–25 so that two additional rooms, each with two windows and one door, can be protected.
49. Further modify the logic circuit from Problem 48 for a change in the input sensors where Open = LOW and Closed = HIGH.
50. Sensors are used to monitor the pressure and the temperature of a chemical solution stored in a vat. The circuitry for each sensor produces a HIGH voltage when a specified maximum value is exceeded. An alarm requiring a LOW voltage input must be activated when either the pressure or the temperature is excessive. Design a circuit for this application.
51. In a certain automated manufacturing process, electrical components are automatically inserted in a PCB. Before the insertion tool is activated, the PCB must be properly positioned, and the component to be inserted must be in the chamber. Each of these prerequisite conditions is indicated by a HIGH voltage. The insertion tool requires a LOW voltage to activate it. Design a circuit to implement this process.

MultiSim



Multisim Troubleshooting Practice

52. Open file P03-52. For the specified fault, predict the effect on the circuit. Then introduce the fault and verify whether your prediction is correct.
53. Open file P03-53. For the specified fault, predict the effect on the circuit. Then introduce the fault and verify whether your prediction is correct.
54. Open file P03-54. For the observed behavior indicated, predict the fault in the circuit. Then introduce the suspected fault and verify whether your prediction is correct.
55. Open file P03-55. For the observed behavior indicated, predict the fault in the circuit. Then introduce the suspected fault and verify whether your prediction is correct.

ANSWERS

SECTION CHECKUPS

Section 3–1 The Inverter

1. When the inverter input is 1, the output is 0.



- (b) A negative-going pulse is on the output (HIGH to LOW and back HIGH).

Section 3–2 The AND Gate

1. An AND gate output is HIGH only when all inputs are HIGH.
2. An AND gate output is LOW when one or more inputs are LOW.
3. Five-input AND: $X = 1$ when $ABCDE = 11111$, and $X = 0$ for all other combinations of $ABCDE$.

Section 3–3 The OR Gate

1. An OR gate output is HIGH when one or more inputs are HIGH.
2. An OR gate output is LOW only when all inputs are LOW.
3. Three-input OR: $X = 0$ when $ABC = 000$, and $X = 1$ for all other combinations of ABC .

Section 3–4 The NAND Gate

1. A NAND gate output is LOW only when all inputs are HIGH.
2. A NAND gate output is HIGH when one or more inputs are LOW.
3. NAND: active-LOW output for all HIGH inputs; negative-OR: active-HIGH output for one or more LOW inputs. They have the same truth tables.
4. $X = \overline{ABC}$

Section 3–5 The NOR Gate

1. A NOR gate output is HIGH only when all inputs are LOW.
2. A NOR gate output is LOW when one or more inputs are HIGH.
3. NOR: active-LOW output for one or more HIGH inputs; negative-AND: active-HIGH output for all LOW inputs. They have the same truth tables.
4. $X = \overline{A + B + C}$

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

1. An XOR gate output is HIGH when the inputs are at opposite levels.
2. An XNOR gate output is HIGH when the inputs are at the same levels.
3. Apply the bits to the XOR gate inputs; when the output is HIGH, the bits are different.

Section 3–7 Programmable Logic

1. Fuse, antifuse, EEPROM, EEPROM, flash, and SRAM
2. Volatile means that all the data are lost when power is off and the PLD must be reprogrammed; SRAM-based
3. Text entry and graphic entry
4. JTAG is Joint Test Action Group; the IEEE Std. 1149.1 for programming and test interfacing.
5. entity NORgate is


```
port (A, B, C: in bit; X: out bit);
end entity NORgate;
architecture NORfunction of NORgate is
begin
  X <= A nor B nor C;
end architecture NORfunction;
```
6. entity XORgate is


```
port (A, B: in bit; X: out bit);
end entity XORgate;
architecture XORfunction of XORgate is
begin
  X <= A xor B;
end architecture XORfunction;
```

Section 3–8 Fixed-Function Logic Gates

1. Fixed-function logic cannot be changed. PLDs can be programmed for any logic function.
2. CMOS and bipolar (TTL)

01	00	00	00	00	00
00	00	10	00	00	00
00	11	11	11	11	11
11	11	11	00	00	00
11	11	11	11	11	11
11	01	01	01	01	01
01	01	01	01	01	01
01	10	00	00	00	00
10	01	00	01	01	01
01	01	11	00	00	00
01	00	11	10	00	00
00	10	11	10	00	00
10	00	01	10	00	00
10	00	10	11	11	11

- 00 00 00 11
10 00 11 11
11 11 11 11
00 11 11 01
11 01 01 01
01 01 01 10
01 01 10 01
00 10 01 01
00 01 01 00
11 01 00 10
11 10 10 10
11 10 10 00
01 10 00 11
01 00 11 01
10 11 01
3. (a) LS—Low-power Schottky
 - (b) HC—High-speed CMOS
 - (c) HCT—HC CMOS TTL compatible
 4. Lowest power—CMOS
 5. Six inverters in a package; four 2-input NAND gates in a package
 6. $t_{PLH} = 10 \text{ ns}$; $t_{PHL} = 8 \text{ ns}$
 7. 18 pJ
 8. I_{CCL} —dc supply current for LOW output state; I_{CCH} —dc supply current for HIGH output state
 9. V_{IL} —LOW input voltage; V_{IH} —HIGH input voltage
 10. V_{OL} —LOW output voltage; V_{OH} —HIGH output voltage

Section 3–9 Troubleshooting

1. Opens and shorts are the most common failures.
2. An open input which effectively makes input HIGH
3. Amplitude and period

RELATED PROBLEMS FOR EXAMPLES

3–1 The timing diagram is not affected.

3–2 See Table 3–15.

TABLE 3–15

Inputs	Output	Inputs	Output
<i>ABCD</i>	<i>X</i>	<i>ABCD</i>	<i>X</i>
0000	0	1000	0
0001	0	1001	0
0010	0	1010	0
0011	0	1011	0
0100	0	1100	0
0101	0	1101	0
0110	0	1110	0
0111	0	1111	1

3–3 See Figure 3–97.

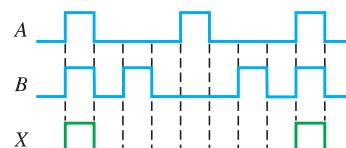


FIGURE 3–97

3–4 The output waveform is the same as input *A*.

3–5 See Figure 3–98.

3–6 Results are the same as example.

3–7 See Figure 3–99.

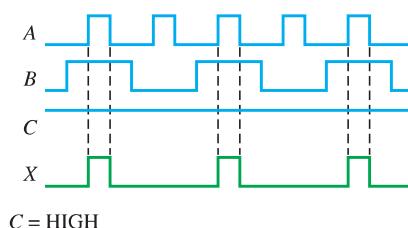


FIGURE 3–98

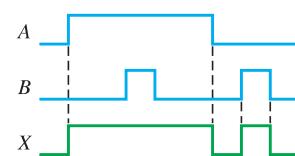


FIGURE 3–99

3-8 See Figure 3-100.

3-9 See Figure 3-101.

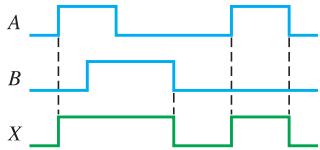


FIGURE 3-100

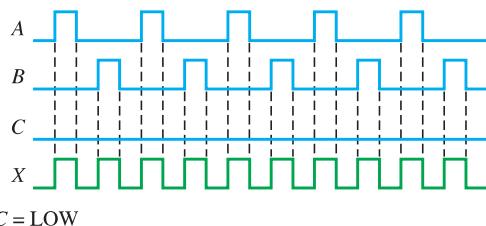


FIGURE 3-101

3-10 See Figure 3-102.

3-11 See Figure 3-103.

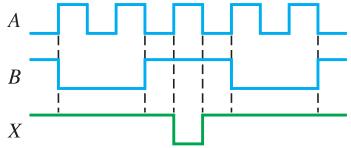


FIGURE 3-102

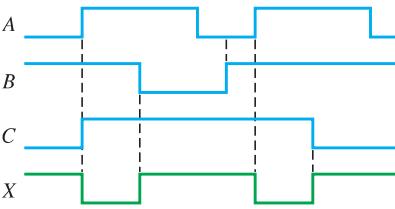


FIGURE 3-103

3-12 Use a 3-input NAND gate.

3-13 Use a 4-input NAND gate operating as a negative-OR gate.

3-14 See Figure 3-104.

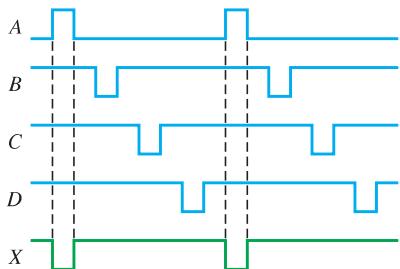


FIGURE 3-104

3-15 See Figure 3-105.

3-16 See Figure 3-106.

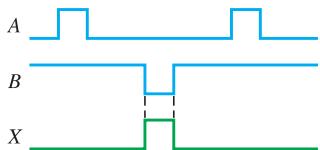


FIGURE 3-105

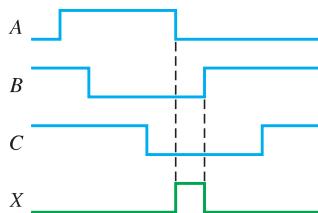


FIGURE 3-106

01	00	00	00	00
00	00	10	00	00
00	11	11	11	11
11	11	00	11	11
11	11	11	11	11
11	01	01	01	01
01	01	01	01	01
01	10	00	10	10
10	01	00	01	01
01	01	11	00	00
01	00	11	10	10
10	10	11	10	00
10	00	01	00	00
00	11	10	11	11

00 00 00 11
10 11 11 11
11 11 11 11
00 11 11 01
11 11 01 01
01 01 01 10
01 01 10 01
00 10 01 01
00 01 01 00
00 01 00 10
11 01 10 10
11 10 10 00
01 10 00 11
01 00 11 01
10 11 01

3-17 Use a 2-input NOR gate.

3-18 A 3-input NAND gate.

3-19 The output is always LOW. The output is a straight line.

3-20 The exclusive-OR gate will not detect simultaneous failures if both circuits produce the same outputs.

3-21 The outputs are unaffected.

3-22 6 columns, 9 rows, and 3 AND gates with three inputs each

3-23 The gate with 4 ns t_{PLH} and t_{PHL} can operate at the highest frequency.

3-24 10 mW

3-25 The gate output or pin 13 input is internally open.

3-26 The display will show an erratic readout because the counter continues until reset.

3-27 The enable pulse is too short or the counter is reset too soon.

TRUE/FALSE QUIZ

1. T 2. T 3. F 4. F 5. T
6. T 7. F 8. F 9. T 10. T

SELF-TEST

1. (c) 2. (d) 3. (c) 4. (a) 5. (c) 6. (a) 7. (d) 8. (b) 9. (d)
10. (a) 11. (b) 12. (d) 13. (b) 14. (a) 15. (d) 16. (c) 17. (c)

Exercise

15. Write the VHDL code for segments *d*, *e*, *f*, and *g*.

Simulation

The decoder simulation using Multisim is shown in Figure 4–54 with the letter E selected. Subcircuits are used for the segment logic to be developed as activities or in the lab. The purpose of simulation is to verify proper operation of the circuit.

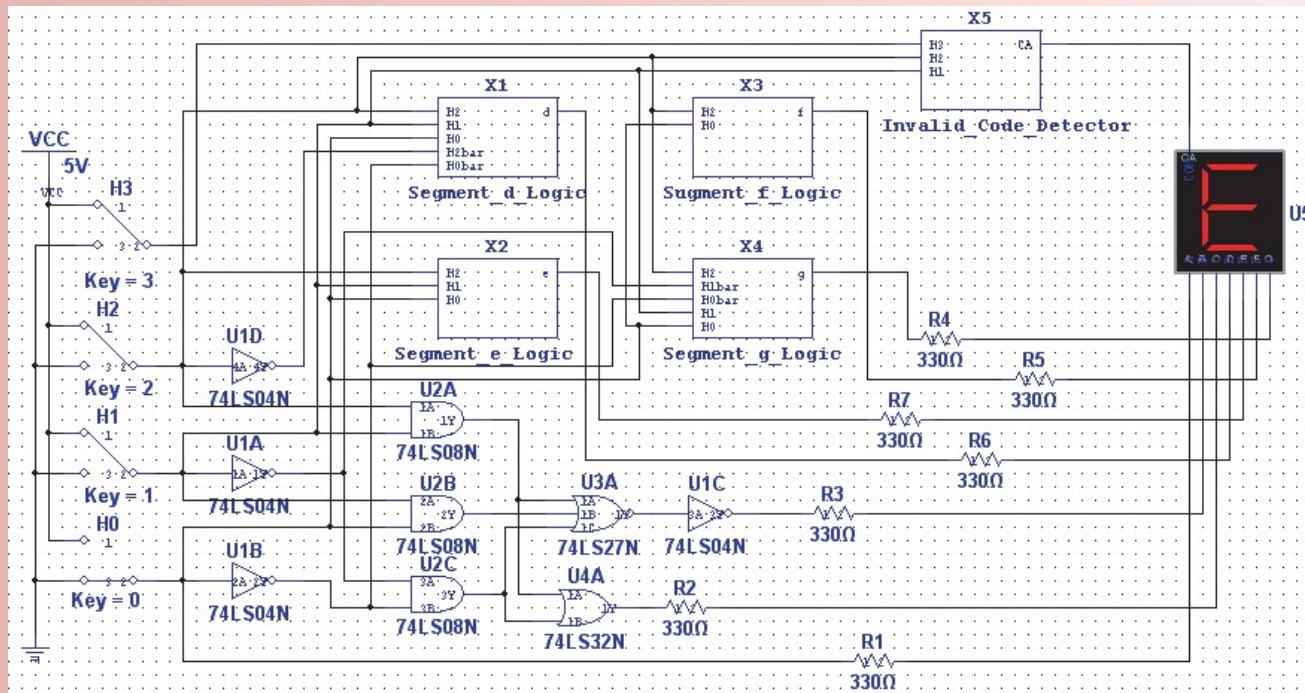


FIGURE 4–54 Multisim circuit screen for decoder and display.



Open file AL04 in the Applied Logic folder on the website. Run the simulation of the decoder and display using your Multisim software. Observe the operation for the specified letters.

Putting Your Knowledge to Work

How would you modify the decoder for a common-cathode 7-segment display?

SUMMARY

- Gate symbols and Boolean expressions for the outputs of an inverter and 2-input gates are shown in Figure 4–55.



FIGURE 4–55

- Commutative laws: $A + B = B + A$
 $AB = BA$
- Associative laws: $A + (B + C) = (A + B) + C$
 $A(BC) = (AB)C$
- Distributive law: $A(B + C) = AB + AC$
- Boolean rules:

1. $A + 0 = A$	7. $A \cdot A = A$
2. $A + 1 = 1$	8. $A \cdot \bar{A} = 0$
3. $A \cdot 0 = 0$	9. $\bar{\bar{A}} = A$
4. $A \cdot 1 = A$	10. $A + AB = A$
5. $A + A = A$	11. $A + \bar{A}B = A + B$
6. $A + \bar{A} = 1$	12. $(A + B)(A + C) = A + BC$
- DeMorgan's theorems:

1. The complement of a product is equal to the sum of the complements of the terms in the product.

$$\overline{XY} = \overline{X} + \overline{Y}$$

2. The complement of a sum is equal to the product of the complements of the terms in the sum.

$$\overline{X + Y} = \overline{X}\overline{Y}$$

- Karnaugh maps for 3 variables have 8 cells and for 4 variables have 16 cells.
- Quinn-McCluskey is a method for simplification of Boolean expressions.
- The three levels of abstraction in VHDL are data flow, structural, and behavioral.

KEY TERMS

Key terms and other bold terms in the chapter are defined in the end-of-book glossary.

Complement The inverse or opposite of a number. In Boolean algebra, the inverse function, expressed with a bar over a variable. The complement of a 1 is 0, and vice versa.

“Don’t care” A combination of input literals that cannot occur and can be used as a 1 or a 0 on a Karnaugh map for simplification.

Karnaugh map An arrangement of cells representing the combinations of literals in a Boolean expression and used for a systematic simplification of the expression.

Minimization The process that results in an SOP or POS Boolean expression that contains the fewest possible literals per term.

Product-of-sums (POS) A form of Boolean expression that is basically the ANDing of ORed terms.

Product term The Boolean product of two or more literals equivalent to an AND operation.

Sum-of-products (SOP) A form of Boolean expression that is basically the ORing of ANDed terms.

Sum term The Boolean sum of two or more literals equivalent to an OR operation.

Variable A symbol used to represent an action, a condition, or data that can have a value of 1 or 0, usually designated by an italic letter or word.

TRUE/FALSE QUIZ

Answers are at the end of the chapter.

1. Variable, complement, and literal are all terms used in Boolean algebra.
2. Addition in Boolean algebra is equivalent to the NOR function.
3. Multiplication in Boolean algebra is equivalent to the AND function.
4. The commutative law, associative law, and distributive law are all laws in Boolean algebra.
5. The complement of 0 is 0 itself.
6. When a Boolean variable is multiplied by its complement, the result is the variable.

01	00	00	00	00
00	00	10	00	00
00	11	11	11	11
11	11	00	00	00
11	11	11	11	11
11	01	01	01	01
01	01	01	01	01
01	10	00	10	10
10	01	00	01	01
01	01	11	00	00
00	10	11	10	10
10	10	01	10	00
10	00	01	11	11
00	11	10	11	11

00	00	11
10	00	11
11	11	11
00	11	11
11	11	01
00	01	01
01	01	10
01	01	10
00	10	01
00	01	01
00	01	00
11	00	10
11	00	10
11	10	10
11	10	00
01	00	11
01	00	11
10	11	01

7. “The complement of a product of variables is equal to the sum of the complements of each variable” is a statement of DeMorgan’s theorem.
8. SOP means sum-of-products.
9. Karnaugh maps can be used to simplify Boolean expressions.
10. A 3-variable Karnaugh map has six cells.
11. VHDL is a type of hardware definition language.
12. A VHDL program consists of an entity and an architecture.

SELF-TEST

Answers are at the end of the chapter.

1. A variable is a symbol in Boolean algebra used to represent
 - (a) data
 - (b) a condition
 - (c) an action
 - (d) answers (a), (b), and (c)
2. The Boolean expression $A + B + C$ is
 - (a) a sum term
 - (b) a literal term
 - (c) an inverse term
 - (d) a product term
3. The Boolean expression \overline{ABCD} is
 - (a) a sum term
 - (b) a literal term
 - (c) an inverse term
 - (d) a product term
4. The domain of the expression $A\overline{B}CD + A\overline{B} + \overline{C}D + B$ is
 - (a) A and D
 - (b) B only
 - (c) A, B, C , and D
 - (d) none of these
5. According to the associative law of addition,
 - (a) $A + B = B + A$
 - (b) $A = A + A$
 - (c) $(A + B) + C = A + (B + C)$
 - (d) $A + 0 = A$
6. According to commutative law of multiplication,
 - (a) $AB = BA$
 - (b) $A = AA$
 - (c) $(AB)C = A(BC)$
 - (d) $A0 = A$
7. According to the distributive law,
 - (a) $A(B + C) = AB + AC$
 - (b) $A(BC) = ABC$
 - (c) $A(A + 1) = A$
 - (d) $A + AB = A$
8. Which one of the following is *not* a valid rule of Boolean algebra?
 - (a) $A + 1 = 1$
 - (b) $A = \overline{A}$
 - (c) $AA = A$
 - (d) $A + 0 = A$
9. Which of the following rules states that if one input of an AND gate is always 1, the output is equal to the other input?
 - (a) $A + 1 = 1$
 - (b) $A + A = A$
 - (c) $A \cdot A = A$
 - (d) $A \cdot 1 = A$
10. According to DeMorgan’s theorems, the complement of a product of variables is equal to
 - (a) the complement of the sum
 - (b) the sum of the complements
 - (c) the product of the complements
 - (d) answers (a), (b), and (c)
11. The Boolean expression $X = (A + B)(C + D)$ represents
 - (a) two ORs ANDed together
 - (b) two ANDs ORed together
 - (c) A 4-input AND gate
 - (d) a 4-input OR gate
12. An example of a sum-of-products expression is
 - (a) $A + B(C + D)$
 - (b) $\overline{AB} + A\overline{C} + A\overline{B}C$
 - (c) $(\overline{A} + B + C)(A + \overline{B} + C)$
 - (d) both answers (a) and (b)
13. An example of a product-of-sums expression is
 - (a) $A(B + C) + A\overline{C}$
 - (b) $(A + B)(\overline{A} + B + \overline{C})$
 - (c) $\overline{A} + \overline{B} + BC$
 - (d) both answers (a) and (b)
14. An example of a standard SOP expression is
 - (a) $\overline{AB} + A\overline{B}C + A\overline{B}\overline{D}$
 - (b) $A\overline{B}C + A\overline{C}D$
 - (c) $\overline{AB} + \overline{A}\overline{B} + AB$
 - (d) $A\overline{B}\overline{C}\overline{D} + \overline{AB} + \overline{A}$



PROBLEMS

Answers to odd-numbered problems are at the end of the book.

Section 4–1 Boolean Operations and Expressions

- Using Boolean notation, write an expression that is a 0 only when all of its variables (A , B , C , and D) are 0s.
 - Write an expression that is a 1 when one or more of its variables (A , B , C , D , and E) are 0s.
 - Write an expression that is a 0 when one or more of its variables (A , B , and C) are 0s.
 - Evaluate the following operations:
(a) $0 + 0 + 0 + 0$ **(b)** $0 + 0 + 0 + 1$ **(c)** $1 + 1 + 1 + 1$
(d) $1 \cdot 1 + 0 \cdot 0 + 1$ **(e)** $1 \cdot 0 \cdot 1 \cdot 0$ **(f)** $1 \cdot 0 + 1 \cdot 0 + 0 \cdot 1 + 0 \cdot 1$
 - Find the values of the variables that make each product term 1 and each sum term 0.
(a) ABC **(b)** $A + B + C$ **(c)** $\bar{A}\bar{B}C$ **(d)** $\bar{A} + \bar{B} + C$
(e) $A + \bar{B} + \bar{C}$ **(f)** $\bar{A} + \bar{B} + \bar{C}$
 - Find the value of X for all possible values of the variables.
(a) $X = A + B + C$ **(b)** $X = (A + B)C$ **(c)** $X = (A + B)(\bar{B} + \bar{C})$
(d) $X = (A + B) + (\bar{A}B + BC)$ **(e)** $X = (\bar{A} + \bar{B})(A + B)$

Section 4–2 Laws and Rules of Boolean Algebra

7. Identify the law of Boolean algebra upon which each of the following equalities is based:

 - $A + AB + ABC + \overline{ABCD} = \overline{ABCD} + ABC + AB + A$
 - $A + \overline{AB} + ABC + \overline{ABCD} = \overline{DCBA} + CBA + \overline{BA} + A$
 - $AB(CD + \overline{CD} + EF + \overline{EF}) = ABCD + AB\overline{CD} + AB EF + AB\overline{EF}$

8. Identify the Boolean rule(s) on which each of the following equalities is based:

 - $\overline{AB + CD} + \overline{EF} = AB + CD + \overline{EF}$
 - $A\bar{A}B + A\bar{B}\bar{C} + A\bar{B}\bar{B} = ABC$
 - $A(BC + BC) + AC = A(BC) + AC$
 - $AB(C + \bar{C}) + AC = AB + AC$
 - $\bar{A}\bar{B} + A\bar{B}\bar{C} = \bar{A}\bar{B}$
 - $ABC + \overline{AB} + \overline{ABCD} = ABC + \overline{AB} + D$

Section 4–3 DeMorgan’s Theorems

9. Apply DeMorgan's theorems to each expression:

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

00	00	00	11
10	00	11	11
11	11	11	11
00	11	11	01
11	01	01	01
01	01	01	10
01	01	10	01
00	10	10	01
00	01	01	00
00	00	01	00
11	01	00	10
11	10	10	00
01	10	00	11
01	11	01	01
10	11	01	01

10. Apply DeMorgan's theorems to each expression:

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

(b) $\overline{AB(CD + EF)}$

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

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

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

11. Apply DeMorgan's theorems to the following:

(a) $\overline{(ABC)\overline{(EFG)}} + \overline{(H\bar{I}J)(K\bar{L}M)}$

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

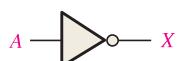
(c) $\overline{(A + B)(C + D)(E + F)(G + H)}$

Section 4-4 Boolean Analysis of Logic Circuits

12. Write the Boolean expression for each of the logic gates in Figure 4-56.



(a)



(b)



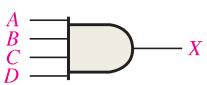
(c)



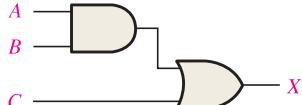
(d)

FIGURE 4-56

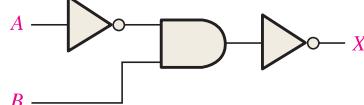
13. Write the Boolean expression for each of the logic circuits in Figure 4-57.



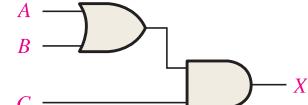
(a)



(b)



(c)



(d)

FIGURE 4-57

14. Draw the logic circuit represented by each of the following expressions:

(a) $A + B + C + D$

(b) $ABCD$

(c) $A + BC$

(d) $ABC + D$

15. Draw the logic circuit represented by each expression:

(a) $AB + \overline{A}\bar{B}$

(b) $ABCD$

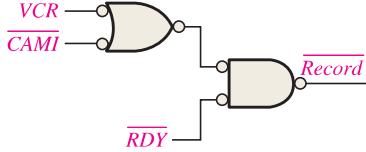
(c) $A + BC$

(d) $ABC + D$

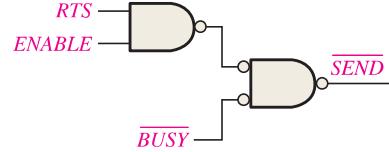
16. (a) Draw a logic circuit for the case where the output, ENABLE, is HIGH only if the inputs, ASSERT and READY, are both LOW.

(b) Draw a logic circuit for the case where the output, HOLD, is HIGH only if the input, LOAD, is LOW and the input, READY, is HIGH.

17. Develop the truth table for each of the circuits in Figure 4-58.



(a)



(b)

FIGURE 4-58

18. Construct a truth table for each of the following Boolean expressions:

(a) $A + B + C$

(b) ABC

(c) $AB + BC + CA$

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

(e) $\overline{AB} + \overline{BC} + \overline{CA}$

Section 4-5 Logic Simplification Using Boolean Algebra

19. Using Boolean algebra techniques, simplify the following expressions as much as possible:

(a) $A(A + B)$

(b) $A(\overline{A} + AB)$

(c) $BC + \overline{BC}$

(d) $A(A + \overline{AB})$

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

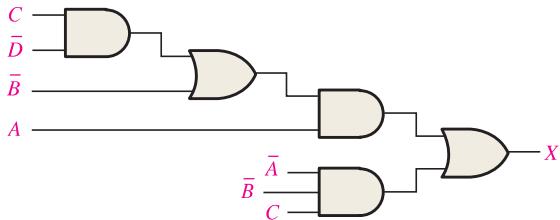
20. Using Boolean algebra, simplify the following expressions:

- (a) $(\bar{A} + B)(A + C)$ (b) $A\bar{B} + A\bar{B}C + A\bar{B}CD + A\bar{B}CDE$
 (c) $BC + \bar{B}CD + B$ (d) $(B + \bar{B})(BC + B\bar{C}\bar{D})$
 (e) $BC + (\bar{B} + \bar{C})D + BC$

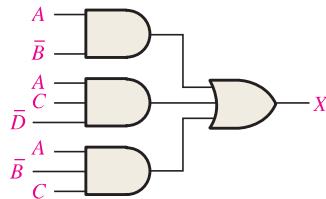
21. Using Boolean algebra, simplify the following expressions:

- (a) $CE + C(E + F) + \bar{E}(E + G)$ (b) $\bar{B}\bar{C}D + (\bar{B} + C + \bar{D}) + \bar{B}\bar{C}\bar{D}E$
 (c) $(C + CD)(C + \bar{C}D)(C + E)$ (d) $BCDE + BC(\bar{D}\bar{E}) + (\bar{B}\bar{C})DE$
 (e) $BCD[BC + \bar{D}(CD + BD)]$

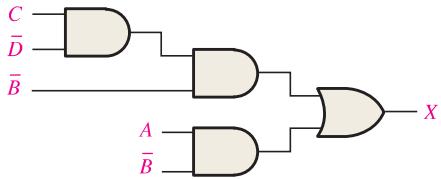
22. Determine which of the logic circuits in Figure 4–59 are equivalent.



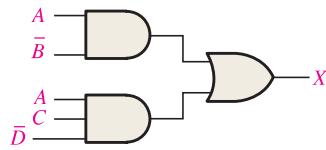
(a)



(b)



(c)



(d)

FIGURE 4–59

Section 4–6 Standard Forms of Boolean Expressions

23. Convert the following expressions to sum-of-product (SOP) forms:

- (a) $(C + D)(A + \bar{D})$ (b) $A(A\bar{D} + C)$ (c) $(A + C)(CD + AC)$

24. Convert the following expressions to sum-of-product (SOP) forms:

- (a) $BC + DE(B\bar{C} + DE)$ (b) $BC(\bar{C}\bar{D} + CE)$ (c) $B + C[BD + (C + \bar{D})E]$

25. Define the domain of each SOP expression in Problem 23 and convert the expression to standard SOP form.

26. Convert each SOP expression in Problem 24 to standard SOP form.

27. Determine the binary value of each term in the standard SOP expressions from Problem 25.

28. Determine the binary value of each term in the standard SOP expressions from Problem 26.

29. Convert each standard SOP expression in Problem 25 to standard POS form.

30. Convert each standard SOP expression in Problem 26 to standard POS form.

Section 4–7 Boolean Expressions and Truth Tables

31. Develop a truth table for each of the following standard SOP expressions:

- (a) $ABC + \bar{A}\bar{B}C + AB\bar{C}$ (b) $\bar{X}\bar{Y}\bar{Z} + \bar{X}Y\bar{Z} + X\bar{Y}Z + \bar{X}YZ$

32. Develop a truth table for each of the following standard SOP expressions:

- (a) $A\bar{B}C\bar{D} + AB\bar{C}\bar{D} + \bar{A}\bar{B}CD + \bar{A}\bar{B}\bar{C}\bar{D}$
 (b) $WXYZ + \bar{W}XY\bar{Z} + W\bar{X}YZ + \bar{W}\bar{X}Y\bar{Z} + WX\bar{Y}\bar{Z}$

33. Develop a truth table for each of the SOP expressions:

- (a) $\bar{A}B + ABC + \bar{A}\bar{C} + A\bar{B}C$ (b) $\bar{X} + Y\bar{Z} + WZ + X\bar{Y}\bar{Z}$

01	00	00	00	00
00	11	11	11	11
11	11	00	11	11
11	11	11	11	11
01	01	01	01	01
01	01	01	01	01
01	10	00	10	10
01	01	01	01	01
01	01	11	00	00
00	10	11	10	10
10	10	01	00	00
10	00	01	11	11
00	11	10	11	11

- 34.** Develop a truth table for each of the standard POS expressions:
- $(\bar{A} + \bar{B} + \bar{C})(A + B + C)(A + B + \bar{C})$
 - $(A + \bar{B} + C + \bar{D})(\bar{A} + B + \bar{C} + D)(A + B + \bar{C} + \bar{D})(\bar{A} + \bar{B} + C + D)$
- 35.** Develop a truth table for each of the standard POS expressions:
- $(A + B)(A + C)(A + B + C)$
 - $(A + \bar{B})(A + \bar{B} + \bar{C})(B + C + \bar{D})(\bar{A} + B + \bar{C} + D)$
- 36.** For each truth table in Table 4–15, derive a standard SOP and a standard POS expression.

TABLE 4–15

$ABCD$		X	$ABCD$		X
ABC	X	ABC	X	ABC	X
0 0 0	1	0 0 0	0	0 0 0	0
0 0 1	1	0 0 1	0	0 0 1	0
0 1 0	0	0 1 0	1	0 0 1	0
0 1 1	1	0 1 1	0	0 1 1	0
1 0 0	0	1 0 0	0	0 1 0	1
1 0 1	1	1 0 1	1	0 1 0	1
1 1 0	0	1 1 0	1	0 1 1	0
1 1 1	1	1 1 1	1	0 1 1	1
(a)		(b)		(c)	
				(d)	

Section 4–8 The Karnaugh Map

- 37.** Draw a 3-variable Karnaugh map and label each cell according to its binary value.
- 38.** Draw a 4-variable Karnaugh map and label each cell according to its binary value.
- 39.** Write the standard product term for each cell in a 3-variable Karnaugh map.

Section 4–9 Karnaugh Map SOP Minimization

- 40.** Use a Karnaugh map to find the minimum SOP form for each expression:
- $\bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}\bar{C} + A\bar{B}C$
 - $AC(\bar{B} + C)$
 - $\bar{A}(BC + \bar{B}\bar{C}) + A(BC + B\bar{C})$
 - $\bar{A}\bar{B}\bar{C} + A\bar{B}\bar{C} + \bar{A}\bar{B}C + ABC$
- 41.** Use a Karnaugh map to simplify each expression to a minimum SOP form:
- $\bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + ABC$
 - $AC[\bar{B} + B(B + \bar{C})]$
 - $DEF + \bar{D}\bar{E}\bar{F} + \bar{D}\bar{E}\bar{F}$
- 42.** Expand each expression to a standard SOP form:
- $AB + A\bar{B}C + ABC$
 - $A + BC$
 - $\bar{A}\bar{B}\bar{C}D + A\bar{C}\bar{D} + B\bar{C}D + \bar{A}\bar{B}CD$
 - $\bar{A}\bar{B} + A\bar{B}\bar{C}D + CD + B\bar{C}D + ABCD$
- 43.** Minimize each expression in Problem 42 with a Karnaugh map.
- 44.** Use a Karnaugh map to reduce each expression to a minimum SOP form:
- $A + B\bar{C} + CD$
 - $\bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}\bar{C}D + ABCD + ABC\bar{D}$
 - $\bar{A}\bar{B}(\bar{C}\bar{D} + \bar{C}D) + AB(\bar{C}\bar{D} + \bar{C}D) + A\bar{B}\bar{C}\bar{D}$
 - $(\bar{A}\bar{B} + A\bar{B})(CD + \bar{C}D)$
 - $\bar{A}\bar{B} + A\bar{B} + \bar{C}\bar{D} + CD$

45. Reduce the function specified in truth Table 4–16 to its minimum SOP form by using a Karnaugh map.
46. Use the Karnaugh map method to implement the minimum SOP expression for the logic function specified in truth Table 4–17.
47. Solve Problem 46 for a situation in which the last six binary combinations are not allowed.

TABLE 4–16

Inputs	Output
A B C	X
0 0 0	1
0 0 1	1
0 1 0	0
0 1 1	1
1 0 0	1
1 0 1	1
1 1 0	0
1 1 1	1

TABLE 4–17

Inputs	Output
A B C D	X
0 0 0 0	0
0 0 0 1	1
0 0 1 0	1
0 0 1 1	0
0 1 0 0	0
0 1 0 1	0
0 1 1 0	1
0 1 1 1	1
1 0 0 0	0
1 0 0 1	0
1 0 1 0	1
1 0 1 1	0
1 1 0 0	1
1 1 0 1	1
1 1 1 0	0
1 1 1 1	1

Section 4–10 Karnaugh Map POS Minimization

48. Use a Karnaugh map to find the minimum POS for each expression:
- $(A + B + C)(\bar{A} + \bar{B} + \bar{C})(A + \bar{B} + C)$
 - $(X + \bar{Y})(\bar{X} + Z)(X + \bar{Y} + \bar{Z})(\bar{X} + \bar{Y} + Z)$
 - $A(B + \bar{C})(\bar{A} + C)(A + \bar{B} + C)(\bar{A} + B + \bar{C})$
49. Use a Karnaugh map to simplify each expression to minimum POS form:
- $(A + \bar{B} + C + \bar{D})(\bar{A} + B + \bar{C} + D)(\bar{A} + \bar{B} + \bar{C} + \bar{D})$
 - $(X + \bar{Y})(W + \bar{Z})(\bar{X} + \bar{Y} + \bar{Z})(W + X + Y + Z)$
50. For the function specified in Table 4–16, determine the minimum POS expression using a Karnaugh map.
51. Determine the minimum POS expression for the function in Table 4–17.
52. Convert each of the following POS expressions to minimum SOP expressions using a Karnaugh map:
- $(A + \bar{B})(A + \bar{C})(\bar{A} + \bar{B} + C)$
 - $(\bar{A} + B)(\bar{A} + \bar{B} + \bar{C})(B + \bar{C} + D)(A + \bar{B} + C + \bar{D})$

Section 4–11 The Quine-McCluskey Method

53. List the minterms in the expression
- $$X = ABC + \bar{A}\bar{B}C + A\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}BC$$
54. List the minterms in the expression
- $$X = \bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}\bar{C}D + \bar{A}\bar{B}\bar{C}\bar{D} + A\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}\bar{C}\bar{D}$$
55. Create a table for the number of 1s in the minterms for the expression in Problem 54 (similar to Table 4–10).
56. Create a table of first level minterms for the expression in Problem 54 (similar to Table 4–11).

00 00 00 00
00 00 10 00
00 11 11 11
11 11 11 11
11 11 11 11
11 01 01 01
01 01 01 01
01 10 00 10
10 01 00 01
01 01 11 00
01 00 11 10
10 10 01 10
10 00 01 00
00 11 10 11

- 00 00 00 11
10 00 11 11
11 11 11 11
00 11 11 01
11 11 01 01
01 01 01 10
01 01 10 01
00 10 01 01
00 01 01 00
11 01 00 10
11 00 10 10
11 10 10 00
01 10 00 11
01 00 11 01
10 11 01
57. Create a table of second level minterms for the expression in Problem 54 (similar to Table 4–12).
 58. Create a table of prime implicants for the expression in Problem 54 (similar to Table 4–13).
 59. Determine the final reduced expression for the expression in Problem 54.

Section 4–12 Boolean Expressions with VHDL

60. Write a VHDL program for the logic circuit in Figure 4–60.

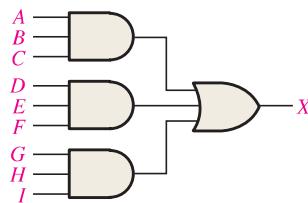


FIGURE 4–60

61. Write a program in VHDL for the expression

$$Y = A\bar{B}C + \bar{A}\bar{B}C + A\bar{B}\bar{C} + \bar{A}BC$$

Applied Logic

62. If you are required to choose a type of digital display for low light conditions, will you select LED or LCD 7-segment displays? Why?
63. Explain the purpose of the invalid code detector.
64. For segment *c*, how many fewer gates and inverters does it take to implement the minimum SOP expression than the standard SOP expression?
65. Repeat Problem 64 for the logic for segments *d* through *g*.

Special Design Problems

66. The logic for segments *b* and *c* in Figure 4–53 produces LOW outputs to activate the segments. If a type of 7-segment display is used that requires a HIGH to activate a segment, modify the logic accordingly.
67. Redesign the logic for segment *a* in the Applied Logic to include the letter F in the display.
68. Repeat Problem 67 for segments *b* through *g*.
69. Design the invalid code detector.

MultiSim



Multisim Troubleshooting Practice

70. Open file P04-70. For the specified fault, predict the effect on the circuit. Then introduce the fault and verify whether your prediction is correct.
71. Open file P04-71. For the specified fault, predict the effect on the circuit. Then introduce the fault and verify whether your prediction is correct.
72. Open file P04-72. For the observed behavior indicated, predict the fault in the circuit. Then introduce the suspected fault and verify whether your prediction is correct.

ANSWERS

SECTION CHECKUPS

Section 4–1 Boolean Operations and Expressions

1. $\bar{A} = \bar{\bar{0}} = 1$
2. $A = 1, B = 1, C = 0; \bar{A} + \bar{B} + C = \bar{1} + \bar{1} + 0 = 0 + 0 + 0 = 0$
3. $A = 1, B = 0, C = 1; A\bar{B}C = 1 \cdot \bar{0} \cdot 1 = 1 \cdot 1 \cdot 1 = 1$

Section 4–2 Laws and Rules of Boolean Algebra

1. $A + (B + C + D) = (A + B + C) + D$
2. $A(B + C + D) = AB + AC + AD$

Section 4–3 DeMorgan’s Theorems

1. (a) $\overline{ABC} + \overline{(D+E)} = \overline{A} + \overline{B} + \overline{C} + D\overline{E}$
- (b) $\overline{(A+B)C} = \overline{A}\overline{B} + \overline{C}$
- (c) $\overline{A+B+C} + \overline{DE} = \overline{A}\overline{B}\overline{C} + D + \overline{E}$

Section 4–4 Boolean Analysis of Logic Circuits

1. $(C+D)B + A$
2. Abbreviated truth table: The expression is a 1 when A is 1 or when B and C are 1s or when B and D are 1s. The expression is 0 for all other variable combinations.

Section 4–5 Logic Simplification Using Boolean Algebra

1. (a) $A + AB + A\overline{BC} = A$
- (b) $(\overline{A} + B)C + ABC = C(\overline{A} + B)$
- (c) $A\overline{B}C(BD + CDE) + A\overline{C} = A(\overline{C} + \overline{B}DE)$
2. (a) Original: 2 AND gates, 1 OR gate, 1 inverter; Simplified: No gates (straight connection)
- (b) Original: 2 OR gates, 2 AND gates, 1 inverter; Simplified: 1 OR gate, 1 AND gate, 1 inverter
- (c) Original: 5 AND gates, 2 OR gates, 2 inverters; Simplified: 2 AND gates, 1 OR gate, 2 inverters

Section 4–6 Standard Forms of Boolean Expressions

1. (a) SOP (b) standard POS (c) standard SOP (d) POS
2. (a) $AB\overline{CD} + A\overline{B}CD + ABC\overline{D} + ABCD + \overline{A}\overline{B}CD + \overline{A}\overline{B}C\overline{D} + \overline{A}\overline{B}C\overline{D}$
- (c) Already standard
3. (b) Already standard
- (d) $(A + \overline{B} + \overline{C})(A + \overline{B} + C)(A + B + \overline{C})(A + B + C)$

Section 4–7 Boolean Expressions and Truth Tables

1. $2^5 = 32$
2. $0110 \longrightarrow \overline{W}\overline{X}YZ$
3. $1100 \longrightarrow \overline{W} + \overline{X} + Y + Z$

Section 4–8 The Karnaugh Map

1. (a) upper left cell: 000 (b) lower right cell: 101
(c) lower left cell: 100 (d) upper right cell: 001
2. (a) upper left cell: $\overline{X}\overline{Y}\overline{Z}$ (b) lower right cell: $X\overline{Y}Z$
(c) lower left cell: $X\overline{Y}\overline{Z}$ (d) upper right cell: $\overline{X}\overline{Y}Z$
3. (a) upper left cell: 0000 (b) lower right cell: 1010
(c) lower left cell: 1000 (d) upper right cell: 0010
4. (a) upper left cell: $\overline{W}\overline{X}\overline{Y}\overline{Z}$ (b) lower right cell: $W\overline{X}YZ$
(c) lower left cell: $W\overline{X}\overline{Y}\overline{Z}$ (d) upper right cell: $\overline{W}\overline{X}YZ$

Section 4–9 Karnaugh Map SOP Minimization

1. 8-cell map for 3 variables; 16-cell map for 4 variables
2. $AB + B\overline{C} + \overline{A}\overline{B}C$
3. (a) $\overline{A}\overline{B}\overline{C} + \overline{A}\overline{B}C + ABC + A\overline{B}\overline{C}$
(b) $\overline{A}\overline{B}\overline{C} + \overline{A}\overline{B}C + \overline{A}\overline{B}\overline{C} + ABC + A\overline{B}\overline{C} + A\overline{B}C$
(c) $\overline{A}\overline{B}\overline{C}\overline{D} + \overline{A}\overline{B}\overline{C}D + \overline{A}\overline{B}C\overline{D} + \overline{A}\overline{B}CD + \overline{A}BC\overline{D} + \overline{A}BCD + A\overline{B}\overline{C}\overline{D} + A\overline{B}\overline{C}D + A\overline{B}C\overline{D} + A\overline{B}CD + ABC\overline{D} + ABCD$
(d) $\overline{A}\overline{B}\overline{C}\overline{D} + \overline{A}\overline{B}\overline{C}D + \overline{A}\overline{B}C\overline{D} + \overline{A}\overline{B}CD + \overline{A}B\overline{C}\overline{D} + \overline{A}B\overline{C}D + A\overline{B}\overline{C}\overline{D} + A\overline{B}\overline{C}D + A\overline{B}C\overline{D} + A\overline{B}CD + ABC\overline{D} + ABCD$

Section 4–10 Karnaugh Map POS Minimization

1. In mapping a POS expression, 0s are placed in cells whose value makes the standard sum term zero; and in mapping an SOP expression 1s are placed in cells having the same values as the product terms.

01	00	00	00	00
00	11	11	11	11
00	11	11	11	11
11	11	11	11	11
11	11	11	11	11
11	01	01	01	01
01	01	01	01	01
01	10	00	01	10
10	01	00	01	01
01	01	11	00	00
01	00	11	10	10
10	10	01	10	10
10	00	01	00	00
00	11	10	11	11

00	00	11
10	11	11
11	11	11
00	11	01
11	11	01
01	01	01
01	01	10
01	10	01
00	01	01
00	01	00
11	00	10
11	00	10
11	10	10
11	10	00
01	10	00
01	00	11
01	00	11
10	11	01

2. 0 in the 1011 cell: $\bar{A} + B + \bar{C} + \bar{D}$

3. 1 in the 0010 cell: $\bar{A}\bar{B}\bar{C}\bar{D}$

Section 4-11 The Quine-McCluskey Method

1. A minterm is a product term in which each variable appears once, either complemented or uncomplemented.
2. An essential prime implicant is a product term that cannot be further simplified by combining with other terms.

Section 4-12 Boolean Expressions with VHDL

1. Simplification can make a VHDL program shorter, easier to read, and easier to modify.
2. Code simplification results in less space used in a target device, thus allowing capacity for more complex circuits.
3. Truth table: Behavioral
Boolean expression: Data flow
Logic diagram: Structural

RELATED PROBLEMS FOR EXAMPLES

4-1 $\bar{A} + B = 0$ when $A = 1$ and $B = 0$.

4-2 $\bar{A}\bar{B} = 1$ when $A = 0$ and $B = 0$.

4-3 XYZ

4-4 $W + X + Y + Z$

4-5 $ABC\bar{D}\bar{E}$

4-6 $(A + \bar{B} + \bar{C}D)\bar{E}$

4-7 $\overline{ABCD} = \bar{A} + \bar{B} + \bar{C} + \bar{D}$

4-8 Results should be same as example.

4-9 $A\bar{B}$

4-10 CD

4-11 $AB\bar{C} + \bar{A}C + \bar{A}\bar{B}$

4-12 $\bar{A} + \bar{B} + \bar{C}$

4-13 Results should be same as example.

4-14 $\bar{A}\bar{B}\bar{C} + AB + A\bar{C} + A\bar{B} + \bar{B}\bar{C}$

4-15 $W\bar{X}YZ + W\bar{X}Y\bar{Z} + W\bar{X}\bar{Y}Z + \bar{W}\bar{X}YZ + WX\bar{Y}Z + WXY\bar{Z}$

4-16 011, 101, 110, 010, 111. Yes

4-17 $(A + \bar{B} + C)(A + \bar{B} + \bar{C})(A + B + C)(\bar{A} + B + C)$

4-18 010, 100, 001, 111, 011. Yes

4-19 SOP and POS expressions are equivalent.

4-20 See Table 4-18.

4-21 See Table 4-19.

TABLE 4-18

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	0

TABLE 4-19

A	B	C	X
0	0	0	1
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	0

4-22 The SOP and POS expressions are equivalent.

4-23 See Figure 4-61.

4-24 See Figure 4-62.

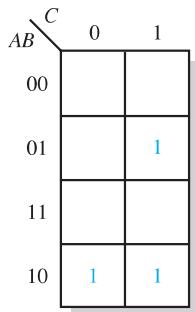


FIGURE 4-61

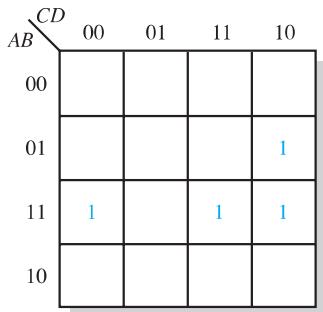


FIGURE 4-62

4-25 See Figure 4-63.

4-26 See Figure 4-64.

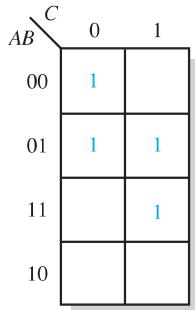


FIGURE 4-63

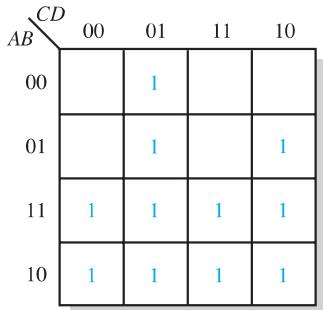


FIGURE 4-64

4-27 No other ways

4-28 $X = B + \bar{A}C + A\bar{C}D + C\bar{D}$

4-29 $X = \bar{D} + A\bar{B}C + B\bar{C} + \bar{A}B$

4-30 $Q = X + Y$

4-31 $Q = \bar{X}\bar{Y}\bar{Z} + W\bar{X}Z + \bar{W}YZ$

4-32 See Figure 4-65.

4-33 See Figure 4-66.

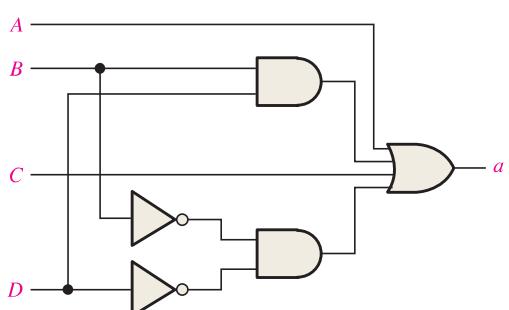


FIGURE 4-65

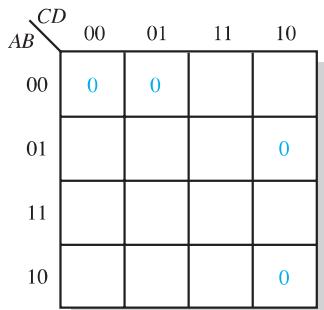


FIGURE 4-66

4-34 $(X + \bar{Y})(X + \bar{Z})(\bar{X} + Y + Z)$
4-35 $(\bar{X} + \bar{Y} + Z)(\bar{W} + \bar{X} + Z)(W + X + Y + Z)(W + \bar{X} + Y + \bar{Z})$
4-36 $\bar{Y}\bar{Z} + \bar{X}\bar{Z} + \bar{W}Y + \bar{X}\bar{Y}Z$
4-37 architecture RelProb_1 of Example4_37 is
begin
 X <= (not A or B or C) and D;
end architecture RelProb_1;
architecture RelProb_2 of Example4_37 is
begin
 X <= (not A and D or B and D or C and D);
end architecture RelProb_2;
4-38 architecture RelProb of Example4_38 is
begin
 X <= not(A and ((B and C) or not D))
end architecture RelProb;

TRUE/FALSE QUIZ

1. T 2. F 3. T 4. T 5. F 6. F
 7. T 8. T 9. T 10. F 11. F 12. T

SELF-TEST

1. (d) 2. (a) 3. (d) 4. (c) 5. (c) 6. (a) 7. (a)
 8. (b) 9. (d) 10. (b) 11. (a) 12. (b) 13. (b) 14. (c)
 15. (c) 16. (c) 17. (c) 18. (b) 19. (c) 20. (c)

Open file AL05 in the Applied Logic folder on the website. Run the simulation of the valve-control logic using your Multisim software and observe the operation. Create a new Multisim file, connect the temperature control logic, and run the simulation.



Putting Your Knowledge to Work

If the temperature of the syrup can never be more than 9°C below the specified value, can the temperature control circuit be simplified? If so, how?

SUMMARY

- AND-OR logic produces an output expression in SOP form.
- AND-OR-Invert logic produces a complemented SOP form, which is actually a POS form.
- The operational symbol for exclusive-OR is \oplus . An exclusive-OR expression can be stated in two equivalent ways:

$$A\bar{B} + \bar{A}B = A \oplus B$$

- To do an analysis of a logic circuit, start with the logic circuit, and develop the Boolean output expression or the truth table or both.
- Implementation of a logic circuit is the process in which you start with the Boolean output expressions or the truth table and develop a logic circuit that produces the output function.
- All NAND or NOR logic diagrams should be drawn using appropriate dual symbols so that bubble outputs are connected to bubble inputs and nonbubble outputs are connected to nonbubble inputs.
- When two negation indicators (bubbles) are connected, they effectively cancel each other.
- A VHDL component is a predefined logic function stored for use throughout a program or in other programs.
- A component instantiation is used to call for a component in a program.
- A VHDL signal effectively acts as an internal interconnection in a VHDL structural description.

KEY TERMS

Key terms and other bold terms in the chapter are defined in the end-of-book glossary.

Component A VHDL feature that can be used to predefine a logic function for multiple use throughout a program or programs.

Negative-AND The dual operation of a NOR gate when the inputs are active-LOW.

Negative-OR The dual operation of a NAND gate when the inputs are active-LOW.

Node A common connection point in a circuit in which a gate output is connected to one or more gate inputs.

Signal A waveform; a type of VHDL object that holds data.

Signal tracing A troubleshooting technique in which waveforms are observed in a step-by-step manner beginning at the input and working toward the output or vice versa. At each point the observed waveform is compared with the correct signal for that point.

Universal gate Either a NAND gate or a NOR gate. The term *universal* refers to the property of a gate that permits any logic function to be implemented by that gate or by a combination of that kind.

00 00 00	00
00 00 10	11
00 11 11	11
11 11 00	11
11 11 11	01
11 01 01	01
01 01 01	01
01 10 00	01
10 01 00	01
01 01 11	00
01 00 11	10
00 10 11	10
10 10 01	00
10 00 01	01
00 11 10	11

TRUE/FALSE QUIZ

Answers are at the end of the chapter.

1. AND-OR logic can have only two 2-input AND gates.
2. AOI is an acronym for AND-OR-Invert.
3. If the inputs of an exclusive-OR gate are the same, the output is LOW (0).
4. If the inputs of an exclusive-NOR gate are different, the output is HIGH (1).
5. A parity generator cannot be implemented using exclusive-OR gates.
6. NAND gates can be used to produce the AND functions.
7. NOR gates cannot be used to produce the OR functions.
8. Any SOP expression can be implemented using only NAND gates.
9. The dual symbol for a NAND gate is a negative-AND symbol.
10. Negative-OR is equivalent to NAND.

SELF-TEST

Answers are at the end of the chapter.

1. The output expression for an AND-OR circuit having one AND gate with inputs A, B and C and one AND gate with inputs D, E and F is

(a) $ABCDEF$	(b) $A + B + C + D + E + F$
(c) $ABC + DEF$	(d) $(A + B + C)(D + E + F)$
2. A logic circuit with an output $X = \overline{AB} + \overline{ABC}$ consists of

(a) two AND gates and one OR gate	(b) two AND gates, one OR gate and an inverter
(c) two AND gates, two OR gates and two inverters	(d) two AND gates, one OR gate and three inverters
3. To implement the expression $\overline{XYZ} + \overline{XY}\overline{Z} + X\overline{Y}\overline{Z} + \overline{X}\overline{YZ} + XY\overline{Z}$, it takes

(a) five AND gates, one OR gate, and eight inverters	(b) four AND gates, two OR gates, and six inverters
(c) five AND gates, three OR gates, and seven inverters	(d) five AND gates, one OR gate, and seven inverters
4. The expression $\overline{ABCD} + ABC\overline{D} + A\overline{B}\overline{C}\overline{D}$

(a) cannot be simplified	(b) can be simplified to $\overline{ABC} + A\overline{B}$
(c) can be simplified to $ABC\overline{D} + A\overline{B}\overline{C}$	(d) None of these answers is correct.
5. The output expression for an AND-OR-Invert circuit having one AND gate with inputs A, B, C and another AND gate with inputs D, E, F is

(a) $ABC + DEF$	(b) $(A + B + C)(D + E + F)$
(c) $(\overline{A} + \overline{B} + \overline{C})(\overline{D} + \overline{E} + \overline{F})$	(d) $\overline{A} + \overline{B} + \overline{C} + \overline{D} + \overline{E} + \overline{F}$
6. An exclusive-NOR function is expressed as

(a) $\overline{AB} + AB$	(b) $\overline{AB} + A\overline{B}$
(c) $(\overline{A} + B)(A + \overline{B})$	(d) $(\overline{A} + \overline{B})(A + B)$
7. The AND operation can be produced with

(a) two NAND gates	(b) three NAND gates
(c) one NOR gate	(d) three NOR gates
8. The OR operation can be produced with

(a) two NOR gates	(b) three NAND gates
(c) four NAND gates	(d) both answers (a) and (b)
9. When using dual symbols in a logic diagram,

(a) bubble outputs are connected to bubble inputs	(b) the NAND symbols produce the AND operations
(c) the negative-OR symbols produce the OR operations	(d) All of these answers are true.
(e) None of these answers is true.	

- 10.** All Boolean expressions can be implemented with
- NAND gates only
 - NOR gates only
 - combinations of NAND and NOR gates
 - combinations of AND gates, OR gates, and inverters
 - any of these
- 11.** A VHDL component
- can be used once in each program
 - is a predefined description of a logic function
 - can be used multiple times in a program
 - is part of a data flow description
 - answers (b) and (c)
- 12.** A VHDL component is called for use in a program by using a
- signal
 - variable
 - component instantiation
 - architecture declaration

PROBLEMS

Answers to odd-numbered problems are at the end of the book.

Section 5–1 Basic Combinational Logic Circuits

- Draw the ANSI distinctive shape logic diagram for a 4-wide, 3-input AND-OR-Invert circuit.
Also draw the ANSI standard rectangular outline symbol.
- Write the output expression for each circuit in Figure 5–54.

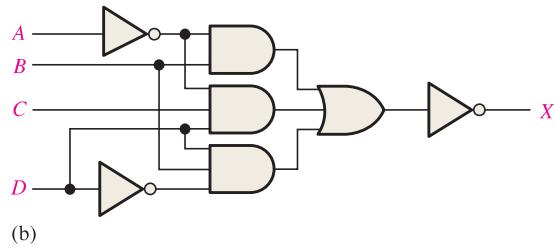
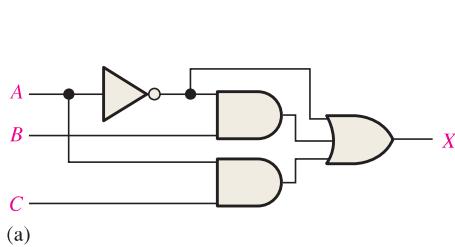


FIGURE 5–54

- Write the output expression for each circuit as it appears in Figure 5–55.

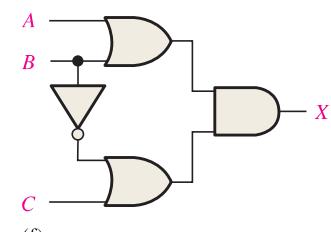
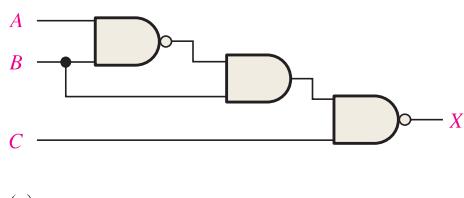
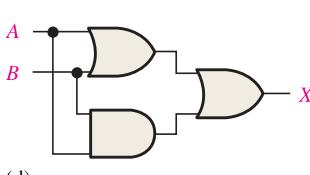
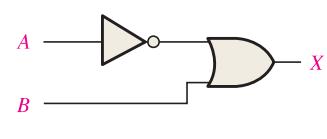
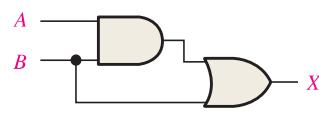
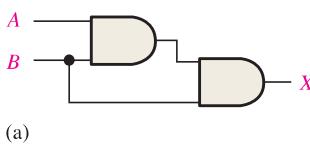


FIGURE 5–55

- 00 00 00 11
10 00 11 11
11 11 11 11
00 11 11 01
11 11 01 01
01 01 01 10
01 01 10 01
00 10 01 01
00 01 01 00
11 00 00 10
11 00 10 10
11 10 10 00
01 10 00 11
01 00 11 01
10 11 01
4. Write the output expression for each circuit as it appears in Figure 5–56 and then change each circuit to an equivalent AND-OR configuration.
 5. Develop the truth table for each circuit in Figure 5–55.
 6. Develop the truth table for each circuit in Figure 5–56.
 7. Show that an exclusive-NOR circuit produces a POS output.

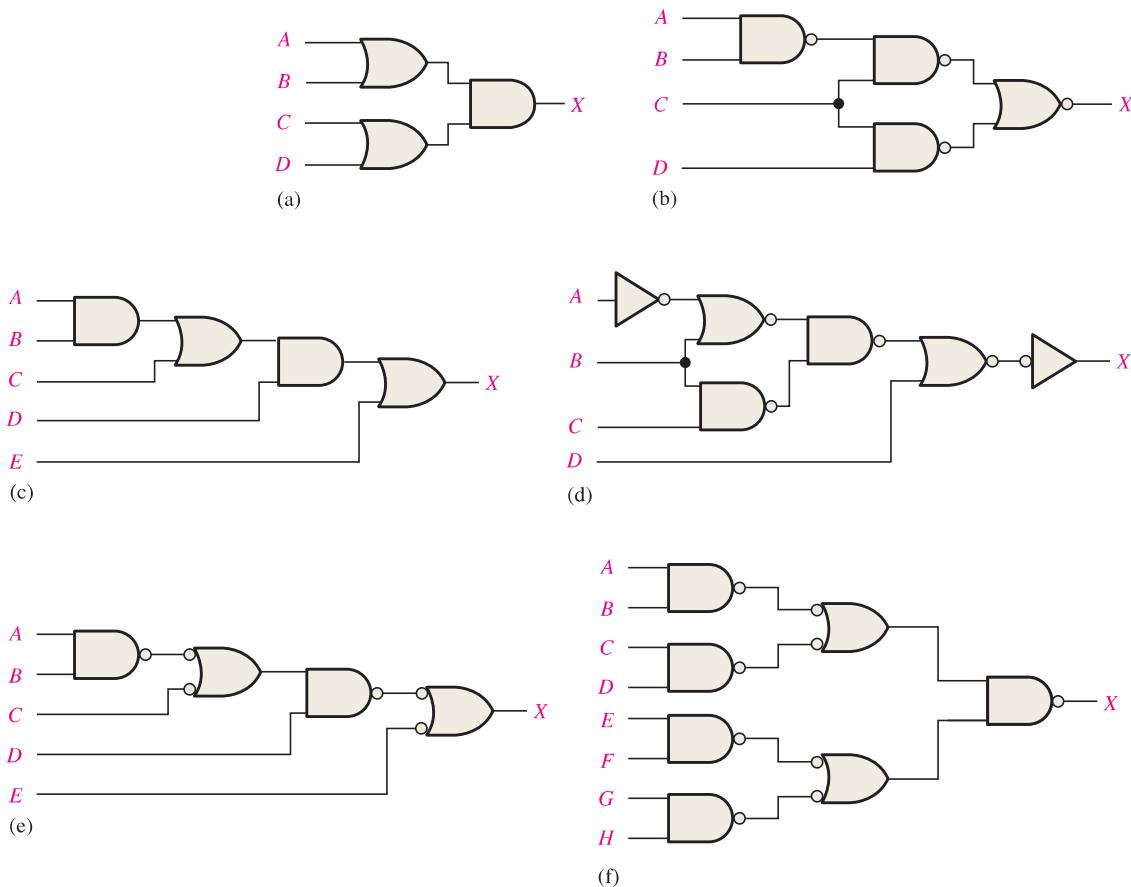


FIGURE 5-56

Section 5–2 Implementing Combinational Logic

8. Develop an AND-OR-Invert logic circuit for a power drive which switches *on* (logic 1) when the guard is in place (logic 1) and switches *off* (logic 0) when the motor is too hot (logic 0).
9. An AOI (AND-OR-Invert) logic chip has two 4-input AND gates connected to a 2-input NOR gate. Write the Boolean expression for the circuit (assume the inputs are labeled A through H).
10. Use AND gates, OR gates, or combinations of both to implement the following logic expressions as stated:
 - (a) $X = A + B + C$
 - (b) $X = ABC$
 - (c) $X = A + BC$
 - (d) $X = AB + CD$
 - (e) $X = (A + B)(C + D)$
 - (f) $X = A + BCD$
 - (g) $X = ABC + BCD + DEF$
 - (h) $X = ABC(D + E + F) + AC(C + D + E)$

11. Use AND gates, OR gates, and inverters as needed to implement the following logic expressions as stated:

- (a) $X = AB + \bar{B}C$
- (b) $X = A(B + \bar{C})$
- (c) $X = A\bar{B} + AB$
- (d) $X = \bar{A}\bar{B}\bar{C} + B(EF + \bar{G})$
- (e) $X = A[BC(A + B + C + D)]$
- (f) $X = B(C\bar{D}E + \bar{E}FG)(\bar{A}\bar{B} + C)$

12. Use NAND gates, NOR gates, or combinations of both to implement the following logic expressions as stated:

- (a) $X = \bar{A}\bar{B} + CD + (\bar{A} + \bar{B})(ACD + \bar{B}\bar{E})$
- (b) $X = AB\bar{C}\bar{D} + D\bar{E}F + \bar{A}\bar{F}$
- (c) $X = \bar{A}[B + \bar{C}(D + E)]$

13. Implement a logic circuit for the truth table in Table 5–8.

TABLE 5–8

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

14. Implement a logic circuit for the truth table in Table 5–9.

TABLE 5–9

Inputs				Output
<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>X</i>
0	0	0	0	0
0	0	0	1	0
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	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	0
1	1	1	0	0
1	1	1	1	1

01	00	00	00
00	00	10	00
00	11	11	11
11	11	00	11
11	11	11	11
11	01	01	01
01	01	01	01
01	10	00	10
10	01	00	01
01	01	11	00
01	00	11	10
00	10	11	10
10	10	01	10
10	00	01	00
00	11	10	11

- 00 00 00 11
10 00 11 11
11 11 11 11
00 11 11 01
11 11 01 01
01 01 01 10
01 01 10 01
00 10 01 01
00 01 01 00
11 01 00 10
11 10 10 10
11 10 10 00
01 10 00 11
01 00 11 01
10 11 01
15. Simplify the circuit in Figure 5–57 as much as possible, and verify that the simplified circuit is equivalent to the original by showing that the truth tables are identical.
 16. Repeat Problem 15 for the circuit in Figure 5–58.

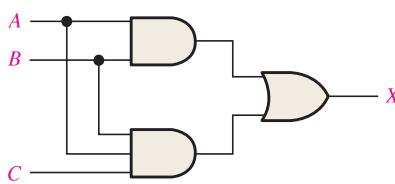


FIGURE 5-57

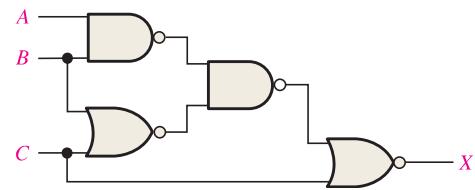


FIGURE 5-58

17. Minimize the gates required to implement the functions in each part of Problem 11 in SOP form.
18. Minimize the gates required to implement the functions in each part of Problem 12 in SOP form.
19. Minimize the gates required to implement the function of the circuit in each part of Figure 5–56 in SOP form.

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

20. Implement the logic circuits in Figure 5–54 using only NAND gates.
21. Implement the logic circuit in Figure 5–58 using only NAND gates.
22. Repeat Problem 20 using only NOR gates.
23. Repeat Problem 21 using only NOR gates.

Section 5-4 Combinational Logic Using NAND and NOR Gates

24. Show how the following expressions can be implemented as stated using only NOR gates:

(a) $X = ABC$	(b) $X = \overline{ABC}$	(c) $X = A + B$
(d) $X = A + B + \overline{C}$	(e) $X = \overline{AB} + \overline{CD}$	(f) $X = (A + B)(C + D)$
(g) $X = AB[\overline{C}\overline{DE} + \overline{AB}] + \overline{BCE}$		
25. Repeat Problem 24 using only NAND gates.
26. Implement each function in Problem 10 by using only NAND gates.
27. Implement each function in Problem 11 by using only NAND gates.

Section 5-5 Pulse Waveform Operation

28. The output of the logic circuit and input waveforms in Figure 5–59 is passed through an inverter. Draw the output waveform.

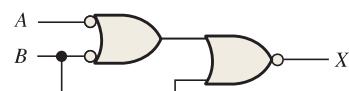


FIGURE 5-59

29. For the logic circuit in Figure 5–60, draw the output waveform in proper relationship to the inputs.

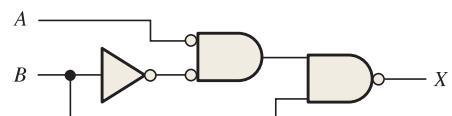


FIGURE 5-60

30. For the input waveforms in Figure 5–61, what logic circuit will generate the output waveform shown?

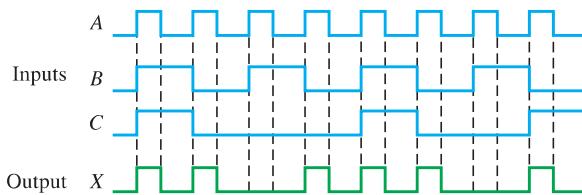


FIGURE 5–61

31. Repeat Problem 30 for the waveforms in Figure 5–62.

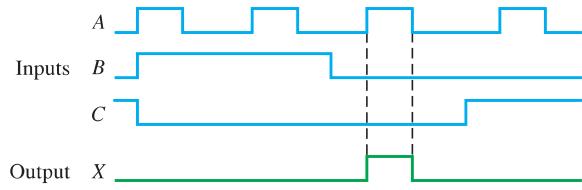


FIGURE 5–62

32. For the circuit in Figure 5–63, draw the waveforms at the numbered points in the proper relationship to each other.

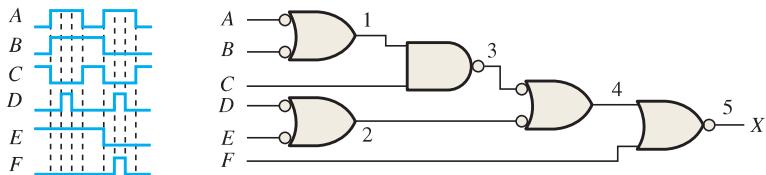


FIGURE 5–63

33. Assuming a propagation delay through each gate of 10 nanoseconds (ns), determine if the *desired* output waveform X in Figure 5–64 (a pulse with a minimum $t_W = 25$ ns positioned as shown) will be generated properly with the given inputs.

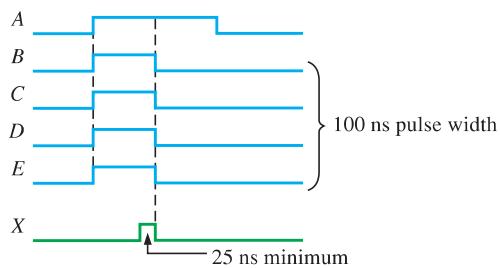
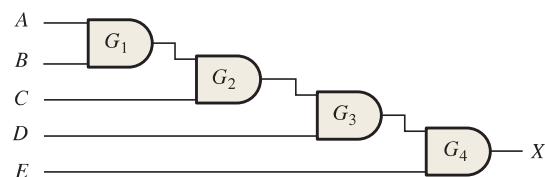


FIGURE 5–64



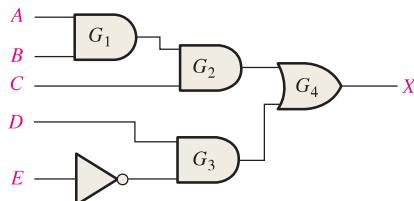
Section 5–6 Combinational Logic with VHDL

34. Describe a 2-input NAND gate with VHDL.
35. Describe a 3-input AND gate with VHDL.
36. Write a VHDL program using the data flow approach (Boolean expressions) to describe the logic circuit in Figure 5–54(b).
37. Write VHDL programs using the data flow approach (Boolean expressions) for the logic circuits in Figure 5–55(e) and (f).

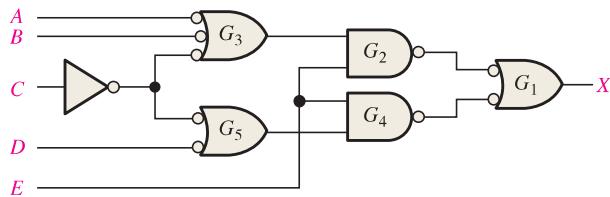
01	00	00	00	00	00	00
00	00	10	11	11	11	11
00	11	11	11	11	11	11
11	11	11	11	11	11	11
11	11	11	11	11	11	11
11	01	01	01	01	01	01
01	01	01	01	01	01	01
01	10	00	01	01	01	01
10	01	00	01	01	01	01
01	01	11	00	01	01	01
01	00	11	10	00	01	01
00	10	11	10	10	01	01
10	00	01	10	00	00	00
10	00	01	11	11	11	11

00	00	11
10	00	11
11	11	11
00	11	01
11	11	01
01	01	10
01	01	10
00	10	01
00	01	01
00	01	00
11	01	00
11	10	10
11	10	00
01	10	00
01	00	11
01	11	01
10	11	01

38. Write a VHDL program using the structural approach for the logic circuit in Figure 5–56(d). Assume component declarations for each type of gate are already available.
39. Repeat Problem 38 for the logic circuit in Figure 5–56(f).
40. Describe the logic represented by the truth table in Table 5–8 using VHDL by first converting it to SOP form.
41. Develop a VHDL program for the logic in Figure 5–65, using both the data flow and the structural approach. Compare the resulting programs.

**FIGURE 5–65**

42. Develop a VHDL program for the logic in Figure 5–66, using both the data flow and the structural approach. Compare the resulting programs.

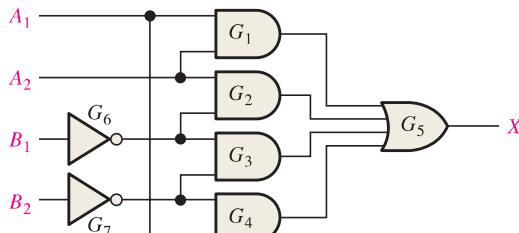
**FIGURE 5–66**

43. Given the following VHDL program, create the truth table that describes the logic circuit.

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

44. Describe the logic circuit shown in Figure 5–67 with a VHDL program, using the data flow approach.

45. Repeat Problem 44 using the structural approach.

**FIGURE 5–67**

Section 5–7 Troubleshooting

46. For the logic circuit and the input waveforms in Figure 5–68, the indicated output waveform is observed. Determine if this is the correct output waveform.

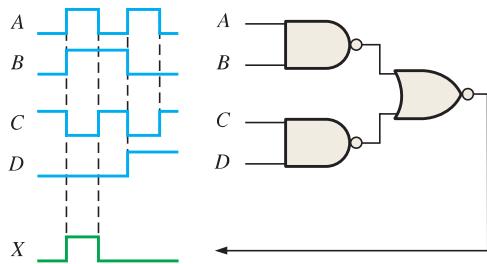


FIGURE 5–68

47. The output waveform in Figure 5–69 is incorrect for the inputs that are applied to the circuit. Assuming that one gate in the circuit has failed, with its output either an apparent constant HIGH or a constant LOW, determine the faulty gate and the type of failure (output open or shorted).

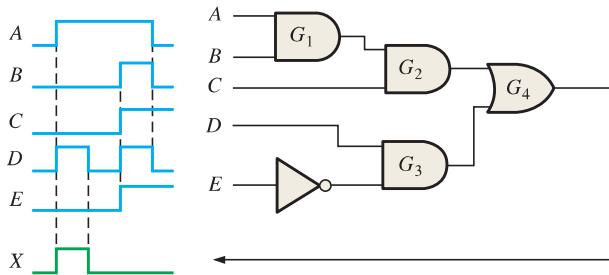


FIGURE 5–69

48. Repeat Problem 47 for the circuit in Figure 5–70, with input and output waveforms as shown.

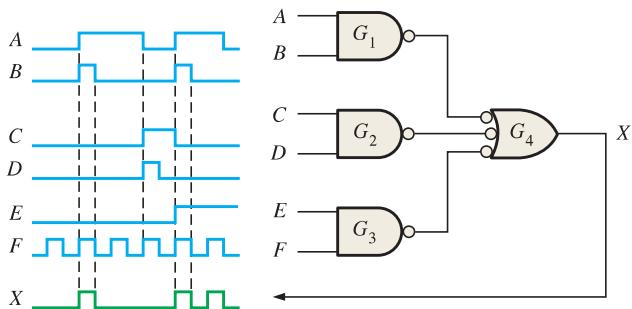


FIGURE 5–70

49. By examining the connections in Figure 5–71, determine the driving gate and load gate(s). Specify by device and pin numbers.

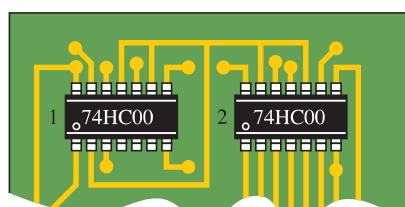
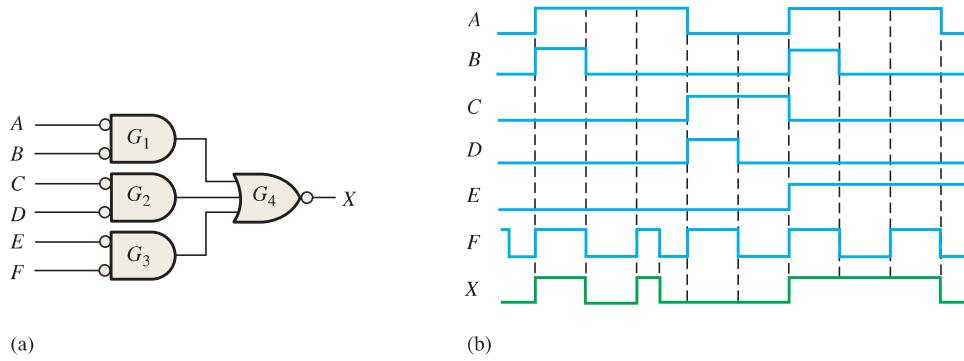


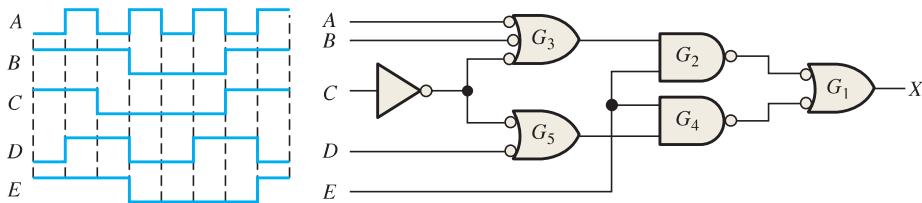
FIGURE 5–71

50. Figure 5–72(a) is a logic circuit under test. Figure 5–72(b) shows the waveforms as observed on a logic analyzer. The output waveform is incorrect for the inputs that are applied to the circuit. Assuming that one gate in the circuit has failed, with its output either an apparent constant HIGH or a constant LOW, determine the faulty gate and the type of failure.

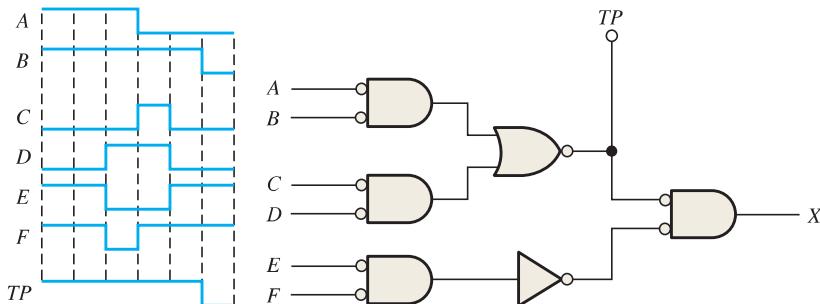
**FIGURE 5-72**

51. The logic circuit in Figure 5–73 has the input waveforms shown.

- (a) Determine the correct output waveform in relation to the inputs.
- (b) Determine the output waveform if the output of gate G_3 is open.
- (c) Determine the output waveform if the upper input to gate G_3 is shorted to ground.

**FIGURE 5-73**

52. The logic circuit in Figure 5–74 has only one intermediate test point available besides the output, as indicated. For the inputs shown, you observe the indicated waveform at the test point. Is this waveform correct? If not, what are the possible faults that would cause it to appear as it does?

**FIGURE 5-74**

Applied Logic

- 53. Describe the function of each of the three sensors in the tank.
- 54. Implement the inlet valve logic using NOR gates and inverters.
- 55. Repeat Problem 54 for the outlet valve logic.
- 56. Implement the temperature control logic using XNOR gates.
- 57. Design a circuit to enable an additive to be introduced into the syrup through another inlet only when the temperature is at the specified value and the syrup level is at the low-level sensor.

01	00	00	00	00
00	00	10	00	00
00	11	11	11	11
11	11	00	11	11
11	11	11	11	11
11	01	01	01	01
01	01	01	01	01
01	10	00	01	10
10	01	00	01	01
01	01	11	00	00
00	10	11	10	10
10	10	01	00	00
10	00	01	11	11
00	11	10	11	11

Special Design Problems

58. (a) Design a logic circuit to produce a HIGH output only if the input, represented by a 4-bit binary number, is greater than twelve or less than three. First develop the truth table and then draw the logic diagram.
 (b) Describe the logic using VHDL.

59. (a) Develop the logic circuit necessary to meet the following requirements:

A battery-powered lamp in a room is to be operated from two switches, one at the back door and one at the front door. The lamp is to be on if the front switch is on and the back switch is off, or if the front switch is off and the back switch is on. The lamp is to be off if both switches are off or if both switches are on. Let a HIGH output represent the on condition and a LOW output represent the off condition.

- (b) Describe the logic using VHDL.
 60. (a) Develop the NAND logic for a hexadecimal keypad encoder that will convert each key closure to binary.
 (b) Describe the logic using VHDL.

Multisim Troubleshooting Practice



61. Open file P05-61. For the specified fault, predict the effect on the circuit. Then introduce the fault and verify whether your prediction is correct.
 62. Open file P05-62. For the specified fault, predict the effect on the circuit. Then introduce the fault and verify whether your prediction is correct.
 63. Open file P05-63. For the observed behavior indicated, predict the fault in the circuit. Then introduce the suspected fault and verify whether your prediction is correct.
 64. Open file P05-64. For the observed behavior indicated, predict the fault in the circuit. Then introduce the suspected fault and verify whether your prediction is correct.

ANSWERS

SECTION CHECKUPS

Section 5-1 Basic Combinational Logic Circuits

1. (a) $\overline{AB + CD} = \overline{1 \cdot 0 + 1 \cdot 0} = 1$ (b) $\overline{AB + CD} = \overline{1 \cdot 1 + 0 \cdot 1} = 0$
 (c) $\overline{AB + CD} = \overline{0 \cdot 1 + 1 \cdot 1} = 0$
2. (a) $\overline{A\bar{B}} + \overline{\bar{A}B} = 1 \cdot \overline{0} + \overline{1} \cdot 0 = 1$ (b) $\overline{A\bar{B}} + \overline{\bar{A}B} = 1 \cdot \overline{1} + \overline{1} \cdot 1 = 0$
 (c) $\overline{A\bar{B}} + \overline{\bar{A}B} = 0 \cdot \overline{1} + \overline{0} \cdot 1 = 1$ (d) $\overline{A\bar{B}} + \overline{\bar{A}B} = 0 \cdot \overline{0} + \overline{0} \cdot 0 = 0$
3. $X = 1$ when $ABC = 000, 011, 101, 110$, and 111 ; $X = 0$ when $ABC = 001, 010$, and 100
4. $X = AB + \overline{A}\overline{B}$; the circuit consists of two AND gates, one OR gate, and two inverters. See Figure 5-6(b) for diagram.

Section 5-2 Implementing Combinational Logic

1. (a) $X = ABC + AB + AC$: three AND gates, one OR gate
 (b) $X = AB(C + DE)$: three AND gates, one OR gate
2. $X = ABC + \overline{A}\overline{B}\overline{C}$; two AND gates, one OR gate, and three inverters
3. (a) $X = AB(C + 1) + AC = AB + AC$
 (b) $X = AB(C + DE) = ABC + ABDE$

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

1. (a) $X = \overline{A} + B$: a 2-input NAND gate with A and \overline{B} on its inputs.
 (b) $X = \overline{AB}$: a 2-input NAND with A and B on its inputs, followed by one NAND used as an inverter.
2. (a) $X = \overline{\overline{A} + B}$: a 2-input NOR with inputs \overline{A} and B , followed by one NOR used as an inverter.
 (b) $X = \overline{A}\overline{B}$: a 2-input NOR with \overline{A} and B on its inputs.

00	00	11
10	00	11
11	11	11
00	11	01
11	11	01
01	01	01
01	01	10
00	10	01
00	01	01
00	01	00
11	01	00
11	10	10
11	10	10
01	10	00
01	00	11
01	11	01
10	11	01

Section 5–4 Combinational Logic Using NAND and NOR Gates

1. $X = \overline{(\overline{A} + \overline{B} + \overline{C})DE}$: a 3-input NAND with inputs A , B , and C , with its output connected to a second 3-input NAND with two other inputs, D and E
2. $X = \overline{\overline{A}\overline{B}\overline{C}} + (D + E)$: a 3-input NOR with inputs A , B , and C , with its output connected to a second 3-input NOR with two other inputs, D and E

Section 5–5 Pulse Waveform Operation

1. The exclusive-OR output is a $15\ \mu s$ pulse followed by a $25\ \mu s$ pulse, with a separation of $10\ \mu s$ between the pulses.
2. The output of the exclusive-NOR is HIGH when both inputs are HIGH or when both inputs are LOW.

Section 5–6 Combinational Logic with VHDL

1. A VHDL component is a predefined program describing a specified logic function.
2. A component instantiation is used to call for a specified component in a program architecture.
3. Interconnections between components are made using VHDL signals.
4. Components are used in the structural approach.

Section 5–7 Troubleshooting

1. Common gate failures are input or output open; input or output shorted to ground.
2. Input shorted to V_{CC} causes output to be stuck LOW.
3. (a) G_4 output is HIGH until rising edge of seventh pulse, then it goes LOW.
 (b) G_4 output is the same as input D .
 (c) G_4 output is the inverse of the G_2 output shown in Figure 5–49(b).

RELATED PROBLEMS FOR EXAMPLES

5–1 $X = AB + AC + BC$

5–2 $X = \overline{AB + AC + BC}$

If $A = 0$ and $B = 0$, $X = \overline{0 \cdot 0 + 0 \cdot 1 + 0 \cdot 1} = \overline{0} = 1$

If $A = 0$ and $C = 0$, $X = \overline{0 \cdot 1 + 0 \cdot 0 + 1 \cdot 0} = \overline{0} = 1$

If $B = 0$ and $C = 0$, $X = \overline{1 \cdot 0 + 1 \cdot 0 + 0 \cdot 0} = \overline{0} = 1$

- 5–3 Determine the even-parity output for all 16 input combinations. Each combination should have an even number of 1s including the parity bit.

- 5–4 Apply codes with odd number of 1s and verify output is 1.

- 5–5 Cannot be simplified

- 5–6 Cannot be simplified

- 5–7 $X = A + B + C + D$ is valid.

- 5–8 See Figure 5–75.

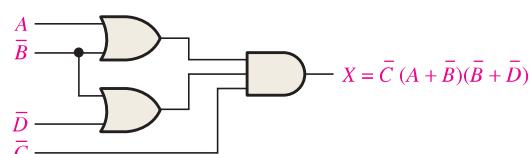


FIGURE 5–75

$$5-9 \quad X = \overline{\overline{(ABC)}\overline{\overline{DEF)}}} = (\overline{AB})C + (\overline{DE})F = (\overline{A} + \overline{B})C + (\overline{D} + \overline{E})F$$

5–10 See Figure 5–76.

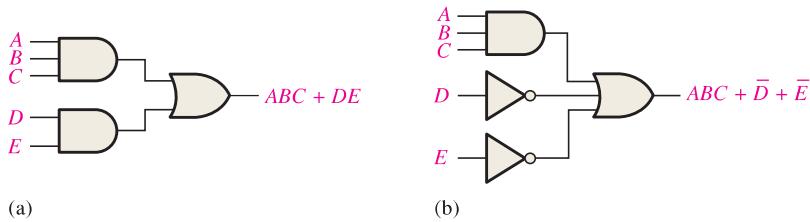


FIGURE 5-76

$$5-11 \quad X = \overline{\overline{(A + B + C)}} + \overline{\overline{(D + E + F)}} = \overline{(A + B + C)}\overline{(D + E) + F} = (\overline{AB} + C)(\overline{DE} + F)$$

5-12 See Figure 5-77.

5-13 See Figure 5-78.

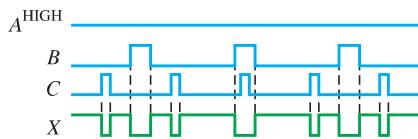


FIGURE 5-77

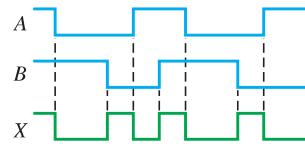


FIGURE 5-78

5-14 See Figure 5-79.

5–15 See Figure 5–80.

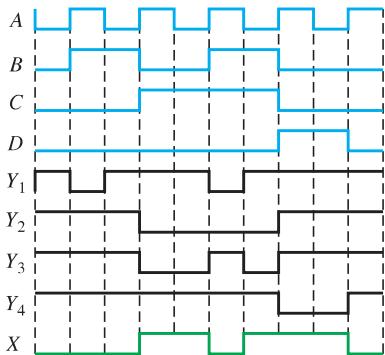


FIGURE 5-79

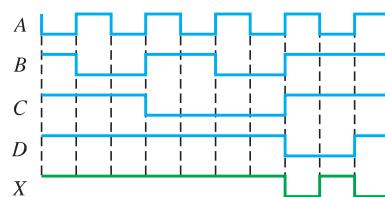


FIGURE 5-80

5-16 G5: NAND_gate2 port map (A => IN9, B => IN10, X => OUT5);

5-17 See Figure 5-81.

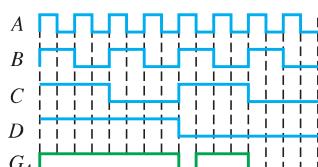


FIGURE 5-81

Simulation

Open Multisim file AL06 in the Applied Logic folder on the website. Run the simulation for the combinational logic unit of the traffic signal controller and observe the operation for each of the four states in the light sequence.

Putting Your Knowledge to Work

There is a requirement for a pedestrian push button that would activate the yellow caution light for 4 s and the red light for 15 s on both the main street and the side street. (a) Modify the state diagram for this additional feature. (b) Develop the additional logic required.

SUMMARY

- Half-adder and full-adder operations are summarized in truth Tables 6–12 and 6–13.

TABLE 6–12

Inputs		Carry Out	Sum
A	B	C_{out}	Σ
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

TABLE 6–13

Inputs		Carry In	Carry Out	Sum
A	B	C_{in}	C_{out}	Σ
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

- Combination logic functions include comparators, decoders, encoders, code converters, multiplexers, demultiplexers, and parity generators/checkers.
- Software versions of standard logic functions from the 74XX series are available for use in a programmable logic design.

KEY TERMS

Key terms and other bold terms in the chapter are defined in the end-of-book glossary.

Cascading Connecting two or more similar devices in a manner that expands the capability of one device.

Comparator A digital circuit that compares the magnitudes of two quantities and produces an output indicating the relationship of the quantities.

Decoder A digital circuit that converts coded information into a familiar or noncoded form.

Demultiplexer (DEMUX) A circuit that switches digital data from one input line to several output lines in a specified time sequence.

Encoder A digital circuit that converts information to a coded form.

Full-adder A digital circuit that adds two bits and an input carry to produce a sum and an output carry.

Glitch A voltage or current spike of short duration, usually unintentionally produced and unwanted.



Half-adder A digital circuit that adds two bits and produces a sum and an output carry. It cannot handle input carries.

Look-ahead carry A method of binary addition whereby carries from preceding adder stages are anticipated, thus eliminating carry propagation delays.

Multiplexer (MUX) A circuit that switches digital data from several input lines onto a single output line in a specified time sequence.

Parity bit A bit attached to each group of information bits to make the total number of 1s odd or even for every group of bits.

Priority encoder An encoder in which only the highest value input digit is encoded and any other active input is ignored.

Ripple carry A method of binary addition in which the output carry from each adder becomes the input carry of the next higher-order adder.

TRUE/FALSE QUIZ

Answers are at the end of the chapter.

1. A half-adder adds two binary bits.
2. A half-adder has a carry output only.
3. A full adder adds two bits and produces two outputs.
4. A full-adder can be realized only by using 2-input XOR gates.
5. When the input bits are both 1 and the input carry bit is 1, the sum output of a full adder is 1.
6. The output of a comparator is 0 when the two binary inputs given are equal.
7. A decoder detects the presence of a specified combination of input bits.
8. The 4-line-to-10-line decoder and the 1-of-10 decoder are two different types.
9. An encoder essentially performs a reverse decoder function.
10. A multiplexer is a logic circuit that allows digital information from a single source to be routed onto several lines.

SELF-TEST

Answers are at the end of the chapter.

1. A half-adder is characterized by

(a) two inputs and two outputs	(b) three inputs and two outputs
(c) two inputs and three outputs	(d) two inputs and one output
2. A full-adder is characterized by

(a) two inputs and two outputs	(b) three inputs and two outputs
(c) two inputs and three outputs	(d) two inputs and one output
3. The inputs to a full adder are $A = 1, B = 0, C_{in} = 1$. The outputs are

(a) $\Sigma = 0, C_{out} = 1$	(b) $\Sigma = 1, C_{out} = 0$
(c) $\Sigma = 0, C_{out} = 0$	(d) $\Sigma = 1, C_{out} = 1$
4. A 3-bit parallel adder can add

(a) three 2-bit binary numbers	(b) two 3-bit binary numbers
(c) three bits at a time	(d) three bits in sequence
5. To expand a 2-bit parallel adder to a 4-bit parallel adder, you must

(a) use two 2-bit adders with no interconnections	(b) use two 2-bit adders and connect the sum outputs of one to the bit inputs of the other
(c) use four 2-bit adders with no interconnections	(d) use two 2-bit adders with the carry output of one connected to the carry input of the other
6. If a 74HC85 magnitude comparator has $A = 1000$ and $B = 1010$, the outputs are

(a) $A > B = 0, A < B = 0, A = B = 0$	(b) $A > B = 0, A < B = 0, A = B = 1$
(c) $A > B = 0, A < B = 1, A = B = 0$	(d) $A > B = 0, A < B = 1, A = B = 1$

7. If a 1-of-16 decoder with active-LOW outputs exhibits a LOW on the decimal 12 output, what are the inputs?
- (a) $A_3A_2A_1A_0 = 1010$ (b) $A_3A_2A_1A_0 = 1110$
 (c) $A_3A_2A_1A_0 = 1100$ (d) $A_3A_2A_1A_0 = 0100$
8. A BCD-to-7 segment decoder has 0100 on its inputs. The active outputs are
- (a) a, c, f, g (b) b, c, f, g
 (c) b, c, e, f (d) b, d, e, g
9. If an octal-to-binary priority encoder has its 0, 2, 5, and 6 inputs at the active level, the active-HIGH binary output is
- (a) 110 (b) 010
 (c) 101 (d) 000
10. In general, a multiplexer has
- (a) one data input, several data outputs, and selection inputs
 (b) one data input, one data output, and one selection input
 (c) several data inputs, several data outputs, and selection inputs
 (d) several data inputs, one data output, and selection inputs
11. Data distributors are basically the same as
- (a) decoders (b) demultiplexers
 (c) multiplexers (d) encoders
12. Which of the following codes exhibit even parity?
- (a) 10011000 (b) 01111000
 (c) 11111111 (d) 11010101
 (e) all (f) both answers (b) and (c)

PROBLEMS

Answers to odd-numbered problems are at the end of the book.

Section 6-1 Half and Full Adders

- For the full-adder of Figure 6-4, determine the outputs for each of the following inputs
- (a) $A = 0, B = 1, C_{in} = 0$ (b) $A = 1, B = 0, C_{in} = 1$
 (c) $A = 0, B = 0, C_{in} = 0$
- What are the half-adder inputs that will produce the following outputs:
- (a) $\Sigma = 0, C_{out} = 0$ (b) $\Sigma = 1, C_{out} = 0$
 (c) $\Sigma = 0, C_{out} = 1$
- Determine the outputs of a full-adder for each of the following inputs:
- (a) $A = 1, B = 0, C_{in} = 0$ (b) $A = 0, B = 0, C_{in} = 1$
 (c) $A = 0, B = 1, C_{in} = 1$ (d) $A = 1, B = 1, C_{in} = 1$

Section 6-2 Parallel Binary Adders

- For the parallel adder in Figure 6-69, determine the complete sum by analysis of the logical operation of the circuit. Verify your result by longhand addition of the two input numbers.

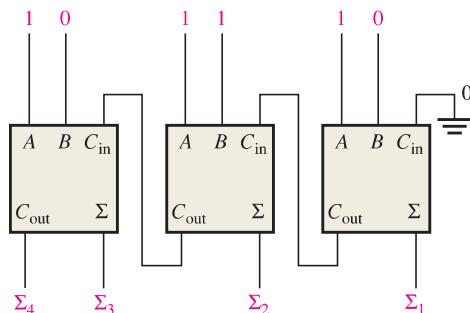


FIGURE 6-69

5. Repeat Problem 4 for the circuit and input conditions in Figure 6–70.

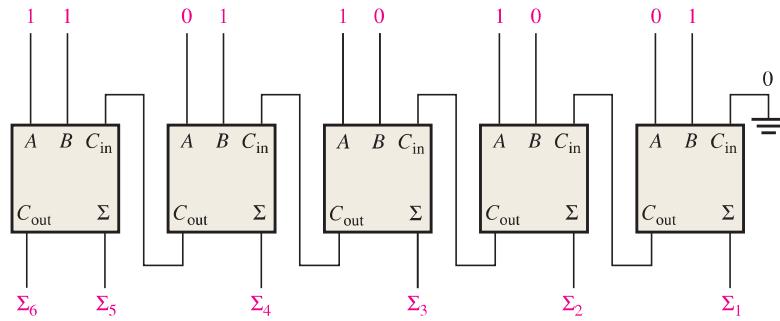


FIGURE 6-70

6. The circuit shown in Figure 6–71 is a 4-bit circuit that can add or subtract numbers in a form used in computers (positive numbers in true form; negative numbers in complement form). (a) Explain what happens when the $\overline{Add/Subt.}$ input is HIGH. (b) What happens when $\overline{Add/Subt.}$ is LOW?

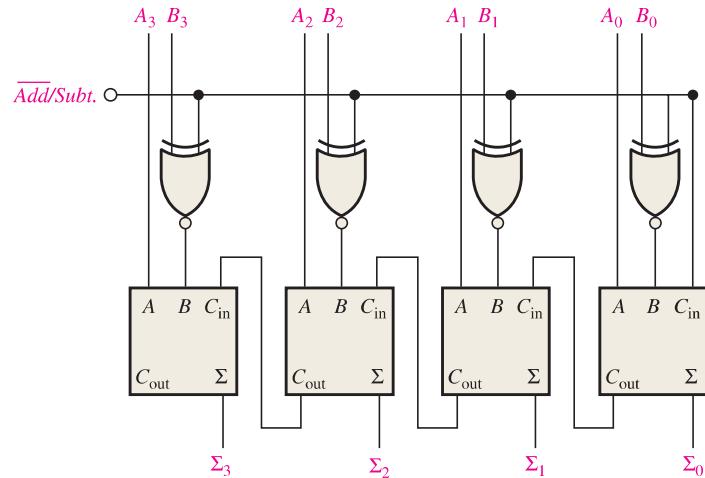


FIGURE 6-71

7. For the circuit in Figure 6–71, assume the inputs are $\overline{Add/Subt.} = 1$, $A = 1010$, and $B = 1101$. What is the output?
 8. The input waveforms in Figure 6–72 are applied to a 2-bit adder. Determine the waveforms for the sum and the output carry in relation to the inputs by constructing a timing diagram.

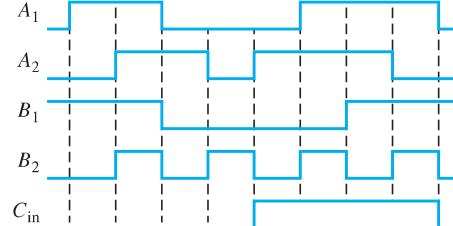


FIGURE 6-72

9. The following sequences of bits (right-most bit first) appear on the inputs to a 4-bit parallel adder. Determine the resulting sequence of bits on each sum output.

A_1	1010
A_2	1100
A_3	0101
A_4	1101
B_1	1001
B_2	1011
B_3	0000
B_4	0001

10. In the process of checking a 74HC283 4-bit parallel adder, the following logic levels are observed on its pins: 1-HIGH, 2-HIGH, 3-HIGH, 4-HIGH, 5-LOW, 6-LOW, 7-LOW, 9-HIGH, 10-LOW, 11-HIGH, 12-LOW, 13-HIGH, 14-HIGH, and 15-HIGH. Determine if the IC is functioning properly.

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

11. Each of the eight full-adders in an 8-bit parallel ripple carry adder exhibits the following propagation delay:

A to Σ and C_{out} :	20 ns
B to Σ and C_{out} :	20 ns
C_{in} to Σ :	30 ns
C_{in} to C_{out} :	25 ns

Determine the maximum total time for the addition of two 8-bit numbers.

12. Show the additional logic circuitry necessary to make the 4-bit look-ahead carry adder in Figure 6-17 into a 5-bit adder.

Section 6-4 Comparators

13. The waveforms in Figure 6-73 are applied to the comparator as shown. Determine the output ($A = B$) waveform.

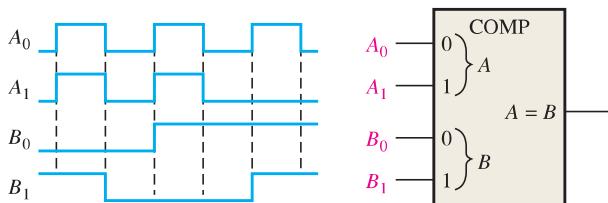


FIGURE 6-73

14. For the 4-bit comparator in Figure 6-74, plot each output waveform for the inputs shown. The outputs are active-HIGH.

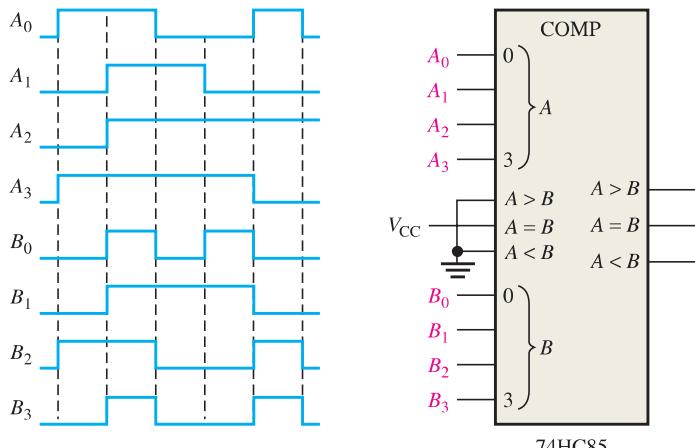


FIGURE 6-74

01	00	00	00	00
00	00	00	00	00
00	11	11	11	11
11	11	11	11	11
11	11	11	01	01
11	01	01	01	01
01	01	01	01	01
01	10	00	01	01
10	00	01	01	01
01	01	01	01	01
01	01	11	00	00
00	10	11	10	00
10	10	01	10	00
10	00	01	00	00
00	11	10	11	11

00	00	00	11
10	00	11	11
11	11	11	11
00	11	11	01
11	11	01	01
00	01	01	01
11	01	01	10
01	01	10	01
00	10	01	01
00	01	01	00
11	01	00	10
11	10	10	00
01	10	00	11
01	11	01	01
10	11	01	01

15. For each set of binary numbers, determine the output states for the comparator of Figure 6–21.

(a) $A_3A_2A_1A_0 = 1010$ (b) $A_3A_2A_1A_0 = 1101$ (c) $A_3A_2A_1A_0 = 1001$
 $B_3B_2B_1B_0 = 1101$ $B_3B_2B_1B_0 = 1101$ $B_3B_2B_1B_0 = 1000$

Section 6–5 Decoders

16. When a LOW is on the output of each of the decoding gates in Figure 6–75, what is the binary code appearing on the inputs? The MSB is A_3 .

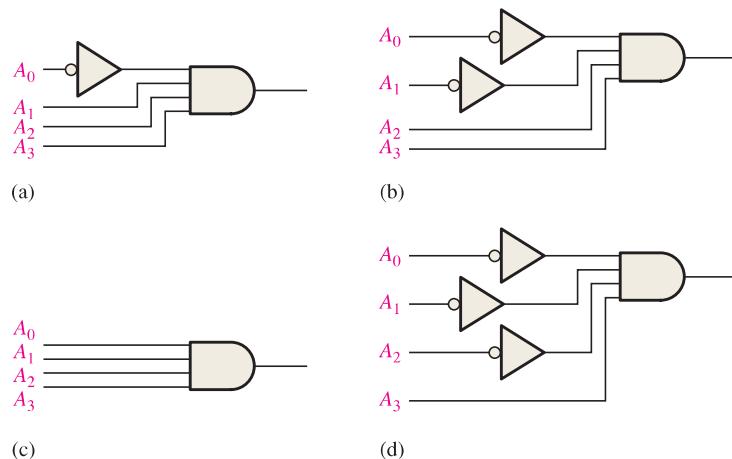


FIGURE 6–75

17. Show the decoding logic for each of the following codes if an active-HIGH (1) output is required:

(a) 1101 (b) 1000 (c) 11011 (d) 11100
 (e) 101010 (f) 111110 (g) 000101 (h) 1110110

18. Solve Problem 17, given that an active-LOW (0) output is required.

19. You wish to detect only the presence of the codes 1010, 1100, 0001, and 1011. An active-HIGH output is required to indicate their presence. Develop the minimum decoding logic with a single output that will indicate when any one of these codes is on the inputs. For any other code, the output must be LOW.

20. If the input waveforms are applied to the decoding logic as indicated in Figure 6–76, sketch the output waveform in proper relation to the inputs.

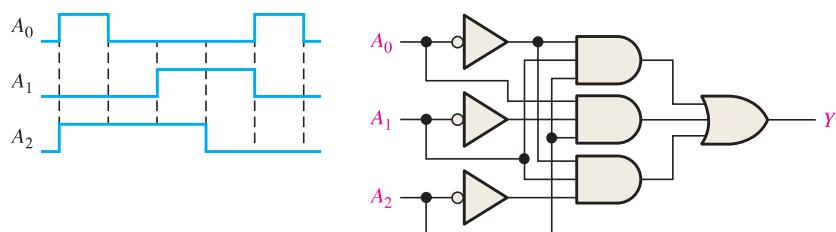


FIGURE 6–76

21. BCD numbers are applied sequentially to the BCD-to-decimal decoder in Figure 6–77. Draw a timing diagram, showing each output in the proper relationship with the others and with the inputs.

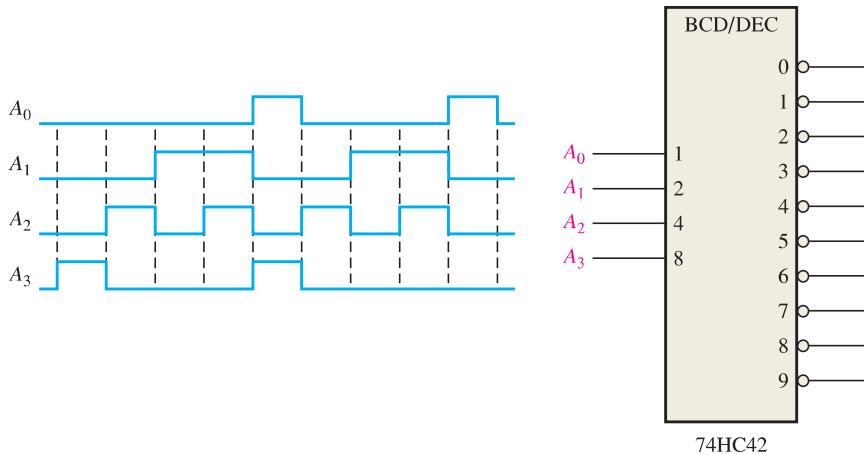


FIGURE 6-77

22. A 7-segment decoder/driver drives the display in Figure 6–78. If the waveforms are applied as indicated, determine the sequence of digits that appears on the display.

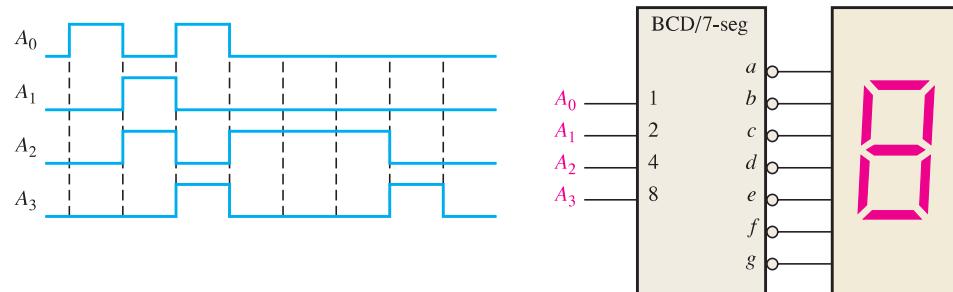


FIGURE 6-78

Section 6-6 Encoders

23. For the decimal-to-BCD encoder logic of Figure 6–37, assume that the 9 input and the 3 input are both HIGH. What is the output code? Is it a valid BCD (8421) code?
24. A 74HC147 encoder has LOW levels on pins 2, 5, and 12. What BCD code appears on the outputs if all the other inputs are HIGH?

Section 6-7 Code Converters

25. Convert each of the following decimal numbers to BCD and then to binary.
- (a) 4 (b) 7 (c) 12 (d) 23 (e) 34
26. Show the logic required to convert a 10-bit binary number to Gray code and use that logic to convert the following binary numbers to Gray code:
- (a) 1010111100 (b) 1111000011 (c) 1011110011 (d) 1000000001
27. Show the logic required to convert a 10-bit Gray code to binary and use that logic to convert the following Gray code words to binary:
- (a) 1010111100 (b) 1111000011 (c) 1011110011 (d) 1000000001

Section 6–8 Multiplexers (Data Selectors)

28. For the multiplexer in Figure 6–79, determine the output for the following input states: $D_0 = 1$, $D_1 = 0$, $D_2 = 0$, $D_3 = 1$, $S_0 = 0$, $S_1 = 1$.

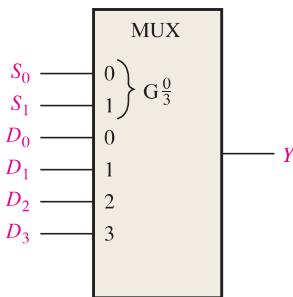


FIGURE 6–79

29. If the data-select inputs to the multiplexer in Figure 6–79 are sequenced as shown by the waveforms in Figure 6–80, determine the output waveform with the data inputs specified in Problem 28.

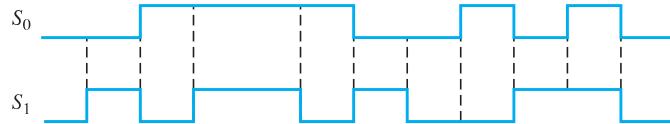


FIGURE 6–80

30. The waveforms in Figure 6–81 are observed on the inputs of a 74HC151 8-input multiplexer. Sketch the Y output waveform.

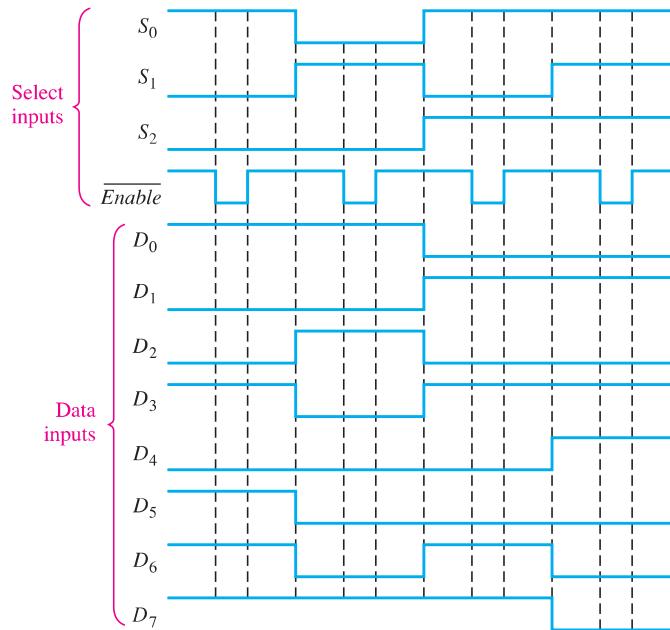


FIGURE 6–81

Section 6–9 Demultiplexers

31. Develop the total timing diagram (inputs and outputs) for a 74HC154 used in a demultiplexing application in which the inputs are as follows: The data-select inputs are repetitively sequenced through a straight binary count beginning with 0000, and the data input is a serial data stream carrying BCD data representing the decimal number 2468. The least significant digit (8) is first in the sequence, with its LSB first, and it should appear in the first 4-bit positions of the output.

Section 6-10 Parity Generators/Checkers

32. The waveforms in Figure 6-82 are applied to the 4-bit parity logic. Determine the output waveform in proper relation to the inputs. For how many bit times does even parity occur, and how is it indicated? The timing diagram includes eight bit times.

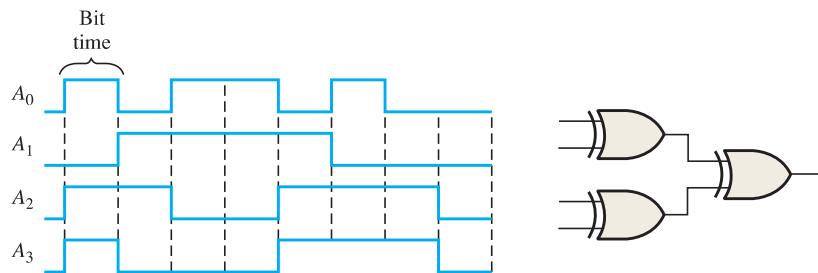


FIGURE 6-82

33. Determine the Σ Even and the Σ Odd outputs of a 74HC280 9-bit parity generator/checker for the inputs in Figure 6-83. Refer to the function table in Figure 6-56.

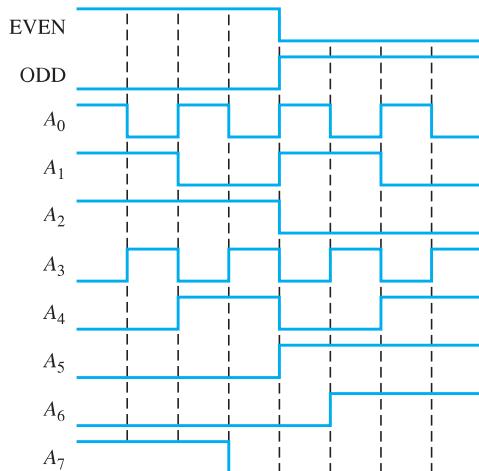


FIGURE 6-83

Section 6-11 Troubleshooting

34. The full-adder in Figure 6-84 is tested under all input conditions with the input waveforms shown. From your observation of the Σ and C_{out} waveforms, is it operating properly, and if not, what is the most likely fault?

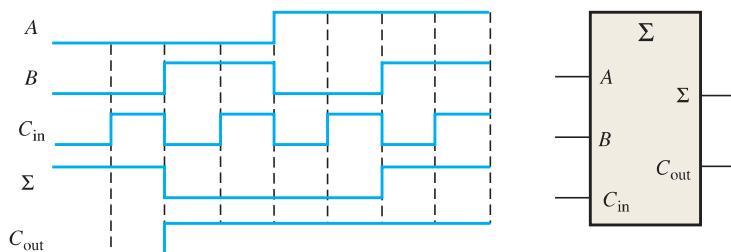


FIGURE 6-84

35. List the possible faults for each decoder/display in Figure 6–85.

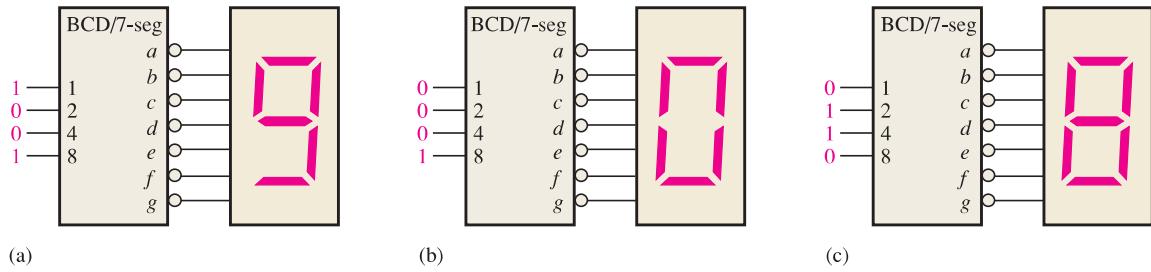


FIGURE 6-85

36. Develop a systematic test procedure to check out the complete operation of the keyboard encoder in Figure 6–39.
37. You are testing a BCD-to-binary converter consisting of 4-bit adders as shown in Figure 6–86. First verify that the circuit converts BCD to binary. The test procedure calls for applying BCD numbers in sequential order beginning with 0_{10} and checking for the correct binary output. What symptom or symptoms will appear on the binary outputs in the event of each of the following faults? For what BCD number is each fault *first* detected?
- (a) The A_1 input is open (top adder).
 - (b) The C_{out} is open (top adder).
 - (c) The Σ_4 output is shorted to ground (top adder).
 - (d) The 32 output is shorted to ground (bottom adder).

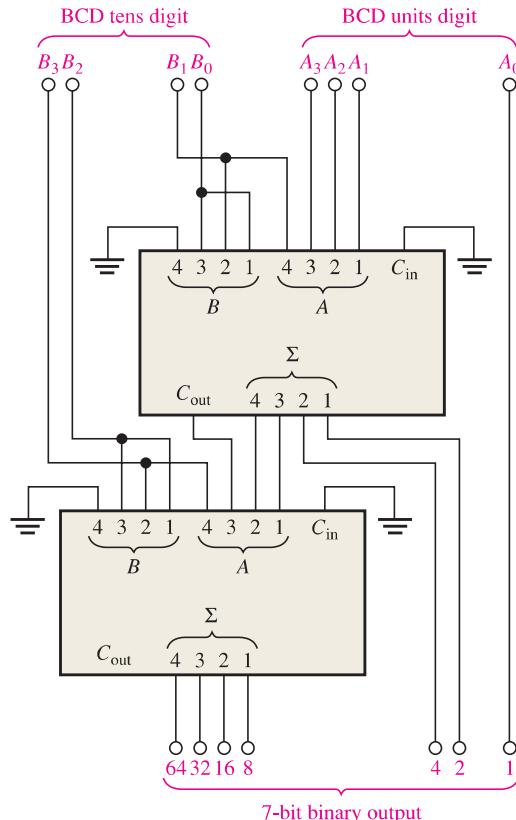


FIGURE 6-86

38. For the 7-segment display multiplexing system in Figure 6–49, determine the most likely cause or causes for each of the following symptoms:

- (a) The B -digit (MSD) display does not turn on at all.
- (b) Neither 7-segment display turns on.
- (c) The f -segment of both displays appears to be on all the time.
- (d) There is a visible flicker on the displays.

39. Develop a systematic procedure to fully test the 74HC151 data selector IC.

40. During the testing of the data transmission system in Figure 6–58, a code is applied to the D_0 through D_6 inputs that contains an odd number of 1s. A single bit error is deliberately introduced on the serial data transmission line between the MUX and the DEMUX, but the system does not indicate an error (error output = 0). After some investigation, you check the inputs to the even parity checker and find that D_0 through D_6 contain an even number of 1s, as you would expect. Also, you find that the D_7 parity bit is a 1. What are the possible reasons for the system not indicating the error?

41. In general, describe how you would fully test the data transmission system in Figure 6–58, and specify a method for the introduction of parity errors.

Applied Logic

42. Use a 74HC00 (quad NAND gates) and any other devices that may be required to produce active-HIGH outputs for the given inputs of the state decoder.

43. Implement the light output logic with the 74HC00 if active-LOW outputs are required.

Special Design Problems

44. Modify the design of the 7-segment display multiplexing system in Figure 6–49 to accommodate two additional digits.

45. Using Table 6–2, write the SOP expressions for the Σ and C_{out} of a full-adder. Use a Karnaugh map to minimize the expressions and then implement them with inverters and AND-OR logic. Show how you can replace the AND-OR logic with 74HC151 data selectors.

46. Implement the logic function specified in Table 6–14 by using a 74HC151 data selector.

TABLE 6–14

Inputs				Output
A_3	A_2	A_1	A_0	Y
0	0	0	0	0
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	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	1	0	0
1	1	1	1	1

47. Using two of the 6-position adder modules from Figure 6–13, design a 12-position voting system.

48. The adder block in the tablet-bottling system in Figure 6–87 performs the addition of the 8-bit binary number from the counter and the 16-bit binary number from Register B. The result from

01	00	00	00	00
00	00	00	10	00
00	00	11	11	11
11	11	11	11	11
11	11	11	01	01
11	01	01	01	01
01	01	01	01	01
01	10	00	10	10
10	01	00	01	01
01	01	11	00	00
01	00	11	10	10
10	10	01	10	10
10	00	01	00	00
00	11	10	11	11

Functions of Combinational Logic

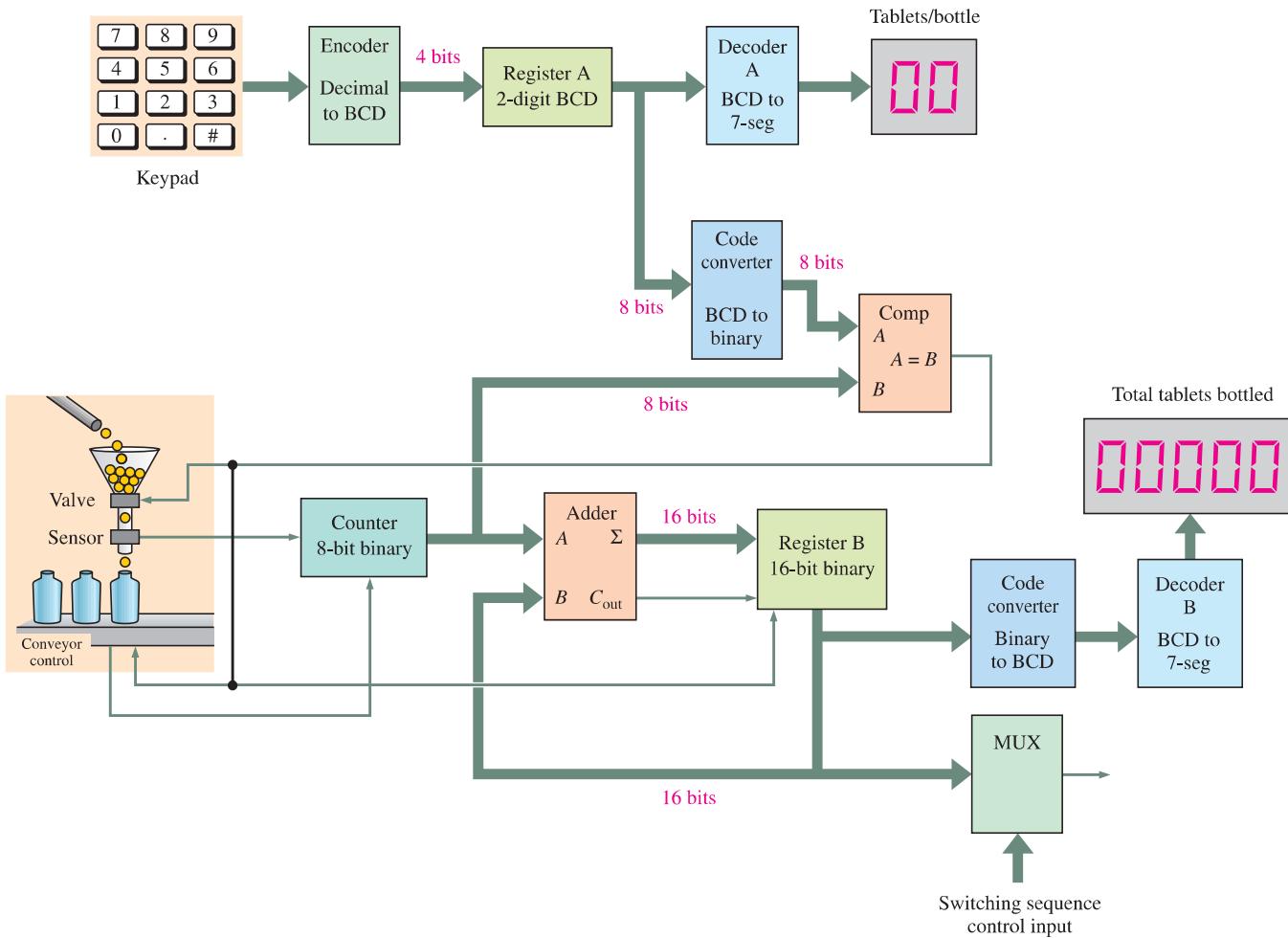


FIGURE 6–87

the adder goes back into Register B. Use 74HC283s to implement this function and draw a complete logic diagram including pin numbers. This is similar to the system in Section 1–4.

49. Use 74HC85s to implement the comparator block in the tablet-bottling system in Figure 6–87 and draw a complete logic diagram including pin numbers. The comparator compares the 8-bit binary number (actually only seven bits are required) from the BCD-to-binary converter with the 8-bit binary number from the counter.
50. Two BCD-to-7-segment decoders are used in the tablet-bottling system in Figure 6–87. One is required to drive the 2-digit *tablets/bottle* display and the other to drive the 5-digit *total tablets bottled* display. Use 74HC47s to implement each decoder and draw a complete logic diagram including pin numbers.
51. The encoder shown in the system block diagram of Figure 6–87 encodes each decimal key closure and converts it to BCD. Use a 74HC147 to implement this function and draw a complete logic diagram including pin numbers.
52. The system in Figure 6–87 requires two code converters. The BCD-to-binary converter changes the 2-digit BCD number in Register A to an 8-bit binary code (actually only 7 bits are required because the MSB is always 0). Use appropriate fixed-function IC code converters to implement the BCD-to-binary converter function and draw a complete logic diagram including pin numbers.



Multisim Troubleshooting Practice

53. Open file P06-53. For the specified fault, predict the effect on the circuit. Then introduce the fault and verify whether your prediction is correct.

54. Open file P06-54. For the specified fault, predict the effect on the circuit. Then introduce the fault and verify whether your prediction is correct.
55. Open file P06-55. For the observed behavior indicated, predict the fault in the circuit. Then introduce the suspected fault and verify whether your prediction is correct.
56. Open file P06-56. For the observed behavior indicated, predict the fault in the circuit. Then introduce the suspected fault and verify whether your prediction is correct.

ANSWERS

SECTION CHECKUPS

Section 6-1 Half and Full Adders

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

Section 6-2 Parallel Binary Adders

1. $C_{\text{out}} \Sigma_4 \Sigma_3 \Sigma_2 \Sigma_1 = 11001$
2. Three 74HC283s are required to add two 10-bit numbers.

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

1. $C_g = 0, C_p = 1$
2. $C_{\text{out}} = 1$

Section 6-4 Comparators

1. $A > B = 1, A < B = 0, A = B = 0$ when $A = 1011$ and $B = 1010$
2. Right comparator: $A < B = 1; A = B = 0; A > B = 0$
Left comparator: $A < B = 0; A = B = 0; A > B = 1$

Section 6-5 Decoders

1. Output 5 is active when 101 is on the inputs.
2. Four 74HC154s are used to decode a 6-bit binary number.
3. Active-HIGH output drives a common-cathode LED display.

Section 6-6 Encoders

1. (a) $A_0 = 1, A_1 = 1, A_2 = 0, A_3 = 1$
 (b) No, this is not a valid BCD code.
 (c) Only one input can be active for a valid output.
2. (a) $\bar{A}_3 = 0, \bar{A}_2 = 1, \bar{A}_1 = 1, \bar{A}_0 = 1$
 (b) The output is 0111, which is the complement of 1000 (8).

Section 6-7 Code Converters

1. 10000101 (BCD) = 1010101_2
2. An 8-bit binary-to-Gray converter consists of seven exclusive-OR gates in an arrangement like that in Figure 6-40 but with inputs B_0-B_7 .

Section 6-8 Multiplexers (Data Selectors)

1. The output is 0.
2. (a) 74HC153: Dual 4-input data selector/multiplexer
 (b) 74HC151: 8-input data selector/multiplexer

01	00	00	00	00	00	00	00
00	00	10	00	00	00	00	00
00	11	11	11	11	11	11	11
11	11	00	00	00	00	00	00
11	11	11	11	11	11	11	11
11	01	01	01	01	01	01	01
01	01	01	01	01	01	01	01
01	10	00	00	00	00	00	00
10	01	00	01	01	01	01	01
01	01	11	01	01	01	01	00
01	00	11	10	00	00	00	00
00	10	11	10	10	01	01	00
10	10	01	10	00	00	00	00
00	11	10	11	11	11	11	11

00	00	00	11
10	00	11	11
11	11	11	11
11	11	11	01
01	01	01	01
01	01	01	10
01	01	10	01
00	01	01	01
00	01	01	00
11	01	00	10
11	10	10	00
01	10	00	11
01	00	11	01
10	11	01	01

3. The data output alternates between LOW and HIGH as the data-select inputs sequence through the binary states.
4. (a) The 74HC157 multiplexes the two BCD codes to the 7-segment decoder.
 (b) The 74HC47 decodes the BCD to energize the display.
 (c) The 74HC139 enables the 7-segment displays alternately.

Section 6–9 Demultiplexers

1. A decoder can be used as a multiplexer by using the input lines for data selection and an Enable line for data input.
2. The outputs are all HIGH except D_{10} , which is LOW.

Section 6–10 Parity Generators/Checkers

1. (a) Even parity: 110100 (b) Even parity: 001100011
2. (a) Odd parity: 11010101 (b) Odd parity: 11000001
3. (a) Code is correct, four 1s. (b) Code is in error, seven 1s

Section 6–11 Troubleshooting

1. A glitch is a very short-duration voltage spike (usually unwanted).
2. Glitches are caused by transition states.
3. Strobe is the enabling of a device for a specified period of time when the device is not in transition.

RELATED PROBLEMS FOR EXAMPLES

- 6–1** $\Sigma = 1, C_{\text{out}} = 1$
6–2 $\Sigma_1 = 0, \Sigma_2 = 0, \Sigma_3 = 1, \Sigma_4 = 1$
6–3 $1011 + 1010 = 10101$
6–4 See Figure 6–88.

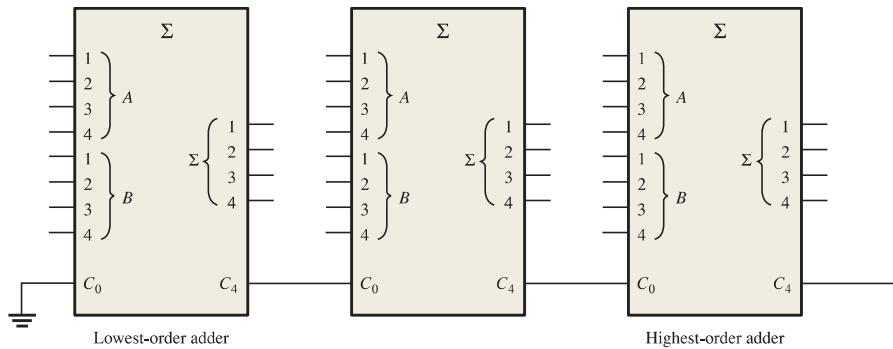


FIGURE 6–88

- 6–5** See Figure 6–89.

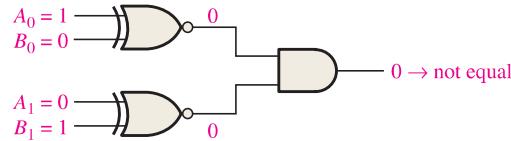


FIGURE 6–89

- 6–6** $A > B = 0, A = B = 0, A < B = 1$

6-7 See Figure 6-90.

6-8 See Figure 6-91.

6-9 Output 22

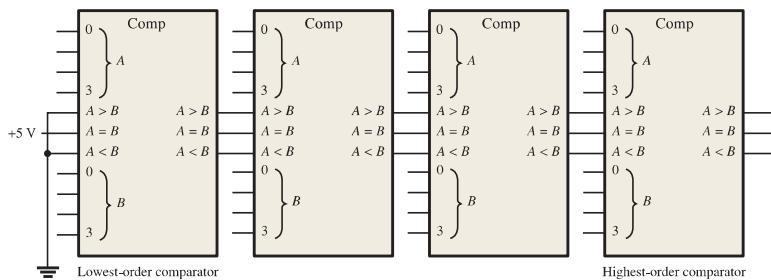


FIGURE 6-90

6-10 See Figure 6-92.

6-11 All inputs LOW: $\bar{A}_0 = 0, \bar{A}_1 = 1, \bar{A}_2 = 1, \bar{A}_3 = 0$

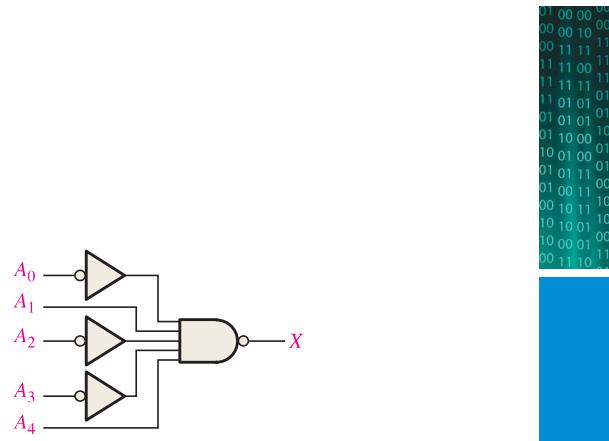


FIGURE 6-91

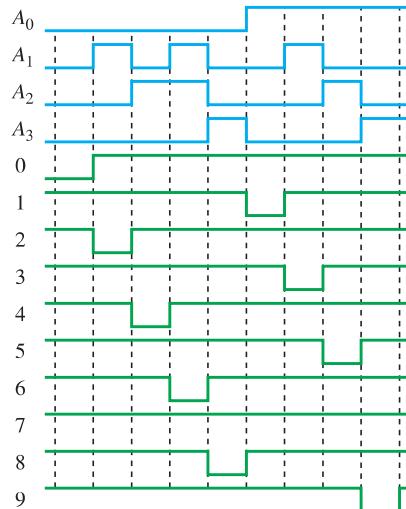


FIGURE 6-92

All inputs HIGH: All outputs HIGH.

6-12 BCD 01000001

$$\begin{array}{r}
 \xrightarrow{\quad} 00000001 \quad 1 \\
 \xrightarrow{\quad} 00101000 \quad 40 \\
 \text{Binary} \quad 00101001 \quad 41
 \end{array}$$

6-13 Seven exclusive-OR gates

6-14 See Figure 6-93.

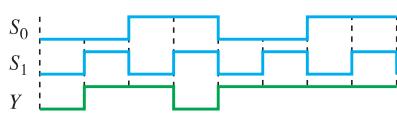


FIGURE 6-93

00 00 00 11
10 11 11 11
11 11 11 11
00 11 11 01
11 11 01 01
01 01 01 10
01 01 10 01
00 10 01 01
00 01 01 00
11 01 00 10
11 10 10 10
11 10 10 00
01 10 00 11
01 00 11 01
10 11 01

- 6-15** $D_0: S_3 = 0, S_2 = 0, S_1 = 0, S_0 = 0$
 $D_4: S_3 = 0, S_2 = 1, S_1 = 0, S_0 = 0$
 $D_8: S_3 = 1, S_2 = 0, S_1 = 0, S_0 = 0$
 $D_{13}: S_3 = 1, S_2 = 1, S_1 = 0, S_0 = 1$

6-16 See Figure 6-94.

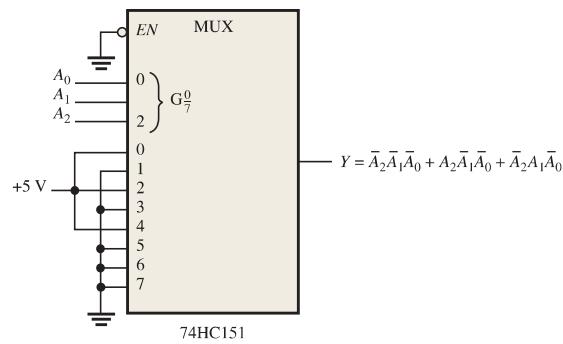


FIGURE 6-94

6-17 See Figure 6-95.

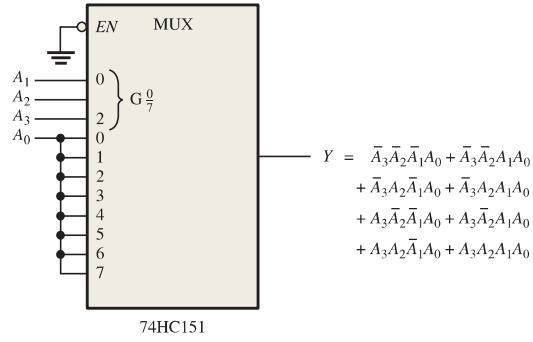


FIGURE 6-95

6-18 See Figure 6-96.

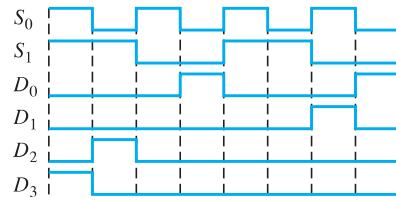


FIGURE 6-96

TRUE/FALSE QUIZ

1. T 2. F 3. F 4. F 5. T
 6. F 7. T 8. F 9. T 10. F

SELF-TEST

1. (a) 2. (b) 3. (a) 4. (b) 5. (d) 6. (c)
 7. (c) 8. (b) 9. (a) 10. (d) 11. (b) 12. (f)

SUMMARY

- Latches are bistable devices whose state normally depends on asynchronous inputs.
- Edge-triggered flip-flops are bistable devices with synchronous inputs whose state depends on the inputs only at the triggering transition of a clock pulse. Changes in the outputs occur at the triggering transition of the clock.
- Monostable multivibrators (one-shots) have one stable state. When the one-shot is triggered, the output goes to its unstable state for a time determined by an RC circuit.
- Astable multivibrators have no stable states and are used as oscillators to generate timing waveforms in digital systems.

KEY TERMS

Key terms and other bold terms in the chapter are defined in the end-of-book glossary.

Astable Having no stable state. An astable multivibrator oscillates between two quasi-stable states.

Bistable Having two stable states. Flip-flops and latches are bistable multivibrators.

Clear An asynchronous input used to reset a flip-flop (make the Q output 0).

Clock The triggering input of a flip-flop.

D flip-flop A type of bistable multivibrator in which the output assumes the state of the D input on the triggering edge of a clock pulse.

Edge-triggered flip-flop A type of flip-flop in which the data are entered and appear on the output on the same clock edge.

Hold time The time interval required for the control levels to remain on the inputs to a flip-flop after the triggering edge of the clock in order to reliably activate the device.

J-K flip-flop A type of flip-flop that can operate in the SET, RESET, no-change, and toggle modes.

Latch A bistable digital circuit used for storing a bit.

Monostable Having only one stable state. A monostable multivibrator, commonly called a *one-shot*, produces a single pulse in response to a triggering input.

One-shot A monostable multivibrator.

Power dissipation The amount of power required by a circuit.

Preset An asynchronous input used to set a flip-flop (make the Q output 1).

Propagation delay time The interval of time required after an input signal has been applied for the resulting output change to occur.

RESET The state of a flip-flop or latch when the output is 0; the action of producing a RESET state.

SET The state of a flip-flop or latch when the output is 1; the action of producing a SET state.

Set-up time The time interval required for the control levels to be on the inputs to a digital circuit, such as a flip-flop, prior to the triggering edge of a clock pulse.

Synchronous Having a fixed time relationship.

Timer A circuit that can be used as a one-shot or as an oscillator.

Toggle The action of a flip-flop when it changes state on each clock pulse.

TRUE/FALSE QUIZ

Answers are at the end of the chapter.

1. A latch has one stable state.
2. A latch is considered to be in the RESET state when the Q output is low.

01	00	00	00	00
00	00	10	00	00
00	11	11	11	11
11	11	11	11	11
11	11	11	01	01
11	01	01	01	01
01	01	01	01	10
01	10	00	01	01
10	01	00	01	01
01	01	11	00	00
01	00	11	10	00
00	10	11	10	00
10	10	01	10	00
10	00	01	11	11
00	11	10	11	11

- 00 00 00 11
10 11 11 11
11 11 11 11
00 11 11 01
11 11 01 01
01 01 01 10
01 01 10 01
00 10 01 01
00 01 01 00
11 01 00 10
11 10 10 10
11 10 10 00
01 00 00 11
01 00 11 01
10 11 01
3. A gated D latch cannot be used to change state.
 4. Flip-flops and latches are both bistable devices.
 5. An edge-triggered D flip-flop changes state whenever the D input changes.
 6. A clock input is necessary for an edge-triggered flip-flop.
 7. When both the J and K inputs are HIGH, an edge-triggered J-K flip-flop changes state on each clock pulse.
 8. A one-shot is also known as an astable multivibrator.
 9. When triggered, a one-shot produces a single pulse.
 10. The 555 timer cannot be used as a pulse oscillator.

SELF-TEST

Answers are at the end of the chapter.

1. An active HIGH input S-R latch is formed by the cross-coupling of
 - (a) two NOR gates
 - (b) two NAND gates
 - (c) two OR gates
 - (d) two AND gates
2. Which of the following is not true for an active LOW input $\bar{S}-\bar{R}$ latch?
 - (a) $S = 1, R = 1, Q = NC, \bar{Q} = NC$
 - (b) $S = 0, R = 1, Q = 1, \bar{Q} = 0$
 - (c) $S = 1, R = 0, Q = 1, \bar{Q} = 0$
 - (d) $S = 0, R = 0, Q = 1, \bar{Q} = 1$
3. For what combinations of the inputs D and EN will a D latch reset?
 - (a) $D = \text{LOW}, EN = \text{LOW}$
 - (b) $D = \text{LOW}, EN = \text{HIGH}$
 - (c) $D = \text{HIGH}, EN = \text{LOW}$
 - (d) $D = \text{HIGH}, EN = \text{HIGH}$
4. A flip-flop changes its state during the
 - (a) complete operational cycle
 - (b) falling edge of the clock pulse
 - (c) rising edge of the clock pulse
 - (d) both answers (b) and (c)
5. The purpose of the clock input to a flip-flop is to
 - (a) clear the device
 - (b) set the device
 - (c) always cause the output to change states
 - (d) cause the output to assume a state dependent on the controlling (J - K or D) inputs.
6. For an edge-triggered D flip-flop,
 - (a) a change in the state of the flip-flop can occur only at a clock pulse edge
 - (b) the state that the flip-flop goes to depends on the D input
 - (c) the output follows the input at each clock pulse
 - (d) all of these answers
7. A feature that distinguishes the J-K flip-flop from the D flip-flop is the
 - (a) toggle condition
 - (b) preset input
 - (c) type of clock
 - (d) clear input
8. A flip-flop is SET when
 - (a) $J = 0, K = 0$
 - (b) $J = 0, K = 1$
 - (c) $J = 1, K = 0$
 - (d) $J = 1, K = 1$
9. A J-K flip-flop with $J = 1$ and $K = 1$ has a 10 kHz clock input. The Q output is
 - (a) constantly HIGH
 - (b) constantly LOW
 - (c) a 10 kHz square wave
 - (d) a 5 kHz square wave
10. A one-shot is a type of
 - (a) monostable multivibrator
 - (b) astable multivibrator
 - (c) timer
 - (d) answers (a) and (c)
 - (e) answers (b) and (c)

01 00 00 00
00 00 10 00
00 11 11 11
11 11 00 11
11 11 11 11
11 01 01 01
01 01 01 10
01 10 00 10
10 01 00 01
01 01 11 01
01 00 11 00
00 10 11 10
10 10 01 10
10 00 01 00
00 11 10 11

PROBLEMS

Answers to odd-numbered problems are at the end of the book.

Section 7-1 Latches

1. If the waveforms in Figure 7–70 are applied to an active-HIGH S-R latch, draw the resulting Q output waveform in relation to the inputs. Assume that Q starts LOW.

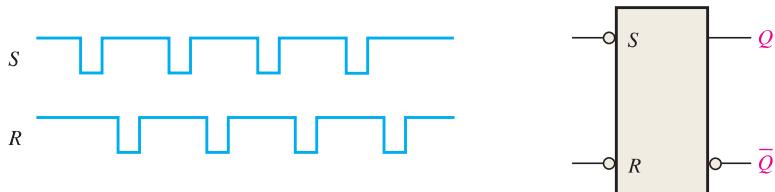


FIGURE 7-70

2. Solve Problem 1 for the input waveforms in Figure 7–71 applied to an active-LOW $\bar{S} - \bar{R}$ latch.

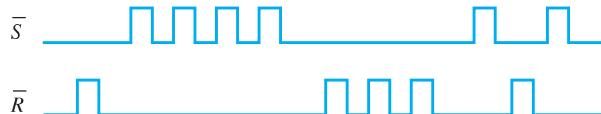


FIGURE 7-71

3. Solve Problem 1 for the input waveform in Figure 7-72.



FIGURE 7-72

4. For a gated S-R latch, determine the Q and \bar{Q} outputs for the inputs in Figure 7–73. Show them in proper relation to the enable input. Assume that Q starts LOW.

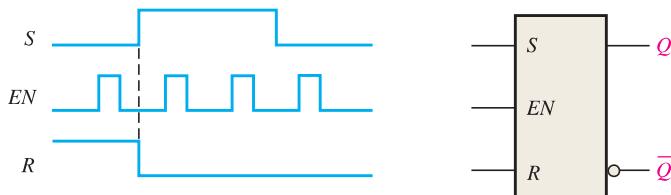


FIGURE 7-73

5. Determine the output of a gated D latch for the inputs in Figure 7–74.



FIGURE 7-74

6. Determine the output of a gated D latch for the inputs in Figure 7–75.



FIGURE 7-75

7. For a gated D latch, the waveforms shown in Figure 7–76 are observed on its inputs. Draw the timing diagram showing the output waveform you would expect to see at Q if the latch is initially RESET.

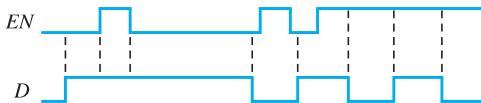


FIGURE 7-76

Section 7-2 Flip-Flops

8. Two edge-triggered J-K flip-flops are shown in Figure 7–77. If the inputs are as shown, draw the Q output of each flip-flop relative to the clock, and explain the difference between the two. The flip-flops are initially RESET.

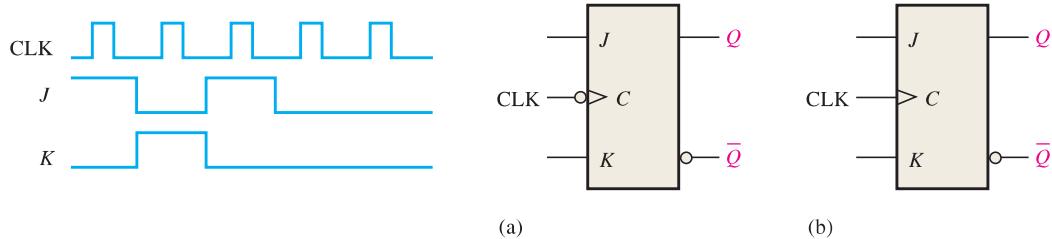


FIGURE 7-77

9. The Q output of an edge-triggered D flip-flop is shown in relation to the clock signal in Figure 7–78. Determine the input waveform on the D input that is required to produce this output if the flip-flop is a positive edge-triggered type.

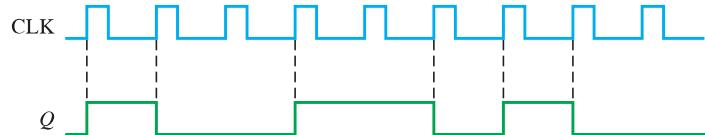


FIGURE 7-78

10. Draw the Q output relative to the clock for a D flip-flop with the inputs as shown in Figure 7–79. Assume positive edge-triggering and Q initially LOW.

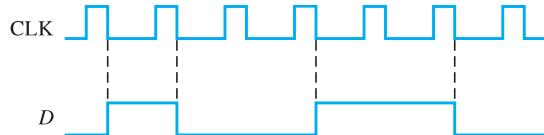


FIGURE 7-79

11. Solve Problem 10 for the inputs in Figure 7–80.

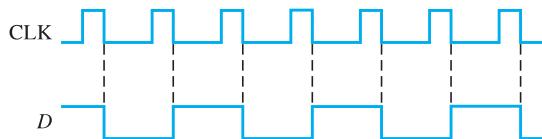


FIGURE 7-80

12. For a positive edge-triggered D flip-flop with the input as shown in Figure 7–81, determine the Q output relative to the clock. Assume that Q starts LOW.

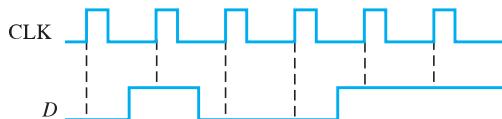


FIGURE 7-81

13. Solve Problem 12 for the input in Figure 7–82.

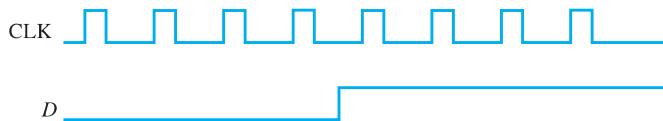


FIGURE 7-82

14. Determine the Q waveform relative to the clock if the signals shown in Figure 7–83 are applied to the inputs of the J-K flip-flop. Assume that Q is initially LOW.

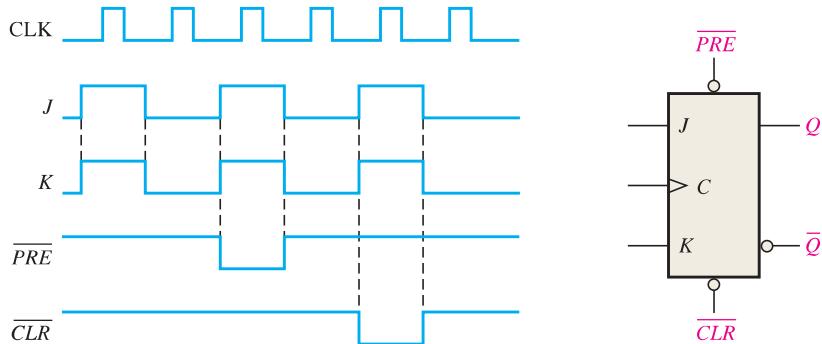


FIGURE 7-83

15. For a negative edge-triggered J-K flip-flop with the inputs in Figure 7–84, develop the Q output waveform relative to the clock. Assume that Q is initially LOW.

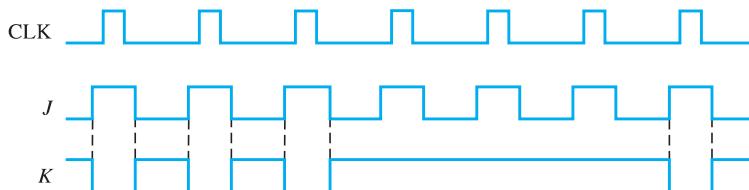


FIGURE 7-84

00 00 00 11
10 11 11 11
11 11 11 11
00 11 11 01
11 11 01 01
01 01 01 10
01 01 10 01
00 10 01 01
00 01 01 00
11 00 10 10
11 10 10 00
01 10 00 11
01 00 11 01
10 11 01

- 16.** The following serial data are applied to the flip-flop through the AND gates as indicated in Figure 7–85. Determine the resulting serial data that appear on the Q output. There is one clock pulse for each bit time. Assume that Q is initially 0 and that \overline{PRE} and \overline{CLR} are HIGH. Right-most bits are applied first.

J_1 : 1 0 1 0 0 1 1; J_2 : 0 1 1 1 0 1 0; J_3 : 1 1 1 1 0 0 0; K_1 : 0 0 0 1 1 1 0; K_2 : 1 1 0 1 1 0 0;
 K_3 : 1 0 1 0 1 0 1

- 17.** For the circuit in Figure 7–85, complete the timing diagram in Figure 7–86 by showing the Q output (which is initially LOW). Assume \overline{PRE} and \overline{CLR} remain HIGH.

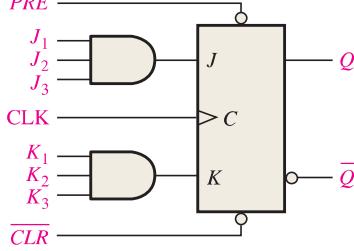


FIGURE 7–85

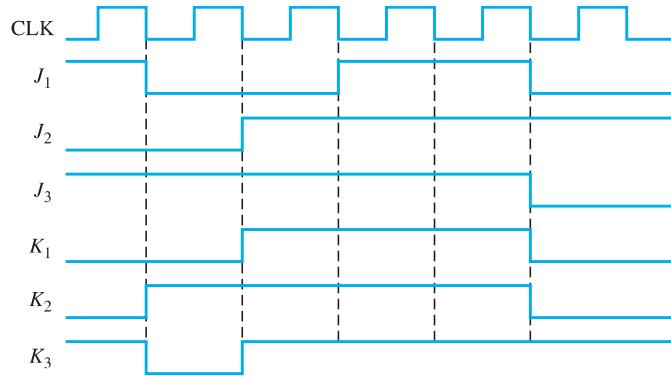


FIGURE 7–86

- 18.** Solve Problem 17 with the same J and K inputs but with the \overline{PRE} and \overline{CLR} inputs as shown in Figure 7–87 in relation to the clock.

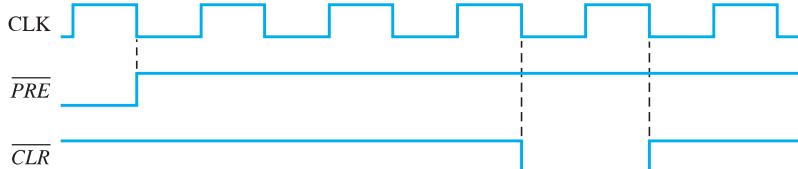


FIGURE 7–87

Section 7–3 Flip-Flop Operating Characteristics

- 19.** What determines the power dissipation of a flip-flop?
- 20.** Typically, a flip-flop is limited in its operation due to hold time and setup time. Explain how.
- 21.** The datasheet of a certain flip-flop specified that the minimum HIGH time for the clock pulse is 20 ns and the minimum LOW time is 40 ns. What is the maximum operating frequency?
- 22.** The flip-flop in Figure 7–88 is initially RESET. Show the relation between Q output and the clock pulse if the propagation delay t_{PLH} (clock to Q) is 5 ns.

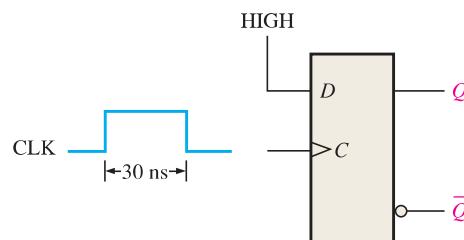


FIGURE 7–88

- 23.** The direct current required by a particular flip-flop that operates on a +4 V dc source is found to be 8 mA. A certain digital device uses 16 of these flip-flops. Determine the current capacity required for the +4 V dc supply and the total power dissipation of the system.

- 24.** For the circuit in Figure 7–89, determine the maximum frequency of the clock signal for reliable operation if the set-up time for each flip-flop is 3 ns and the propagation delays (t_{PLH} and t_{PHL}) from clock to output are 6 ns for each flip-flop.

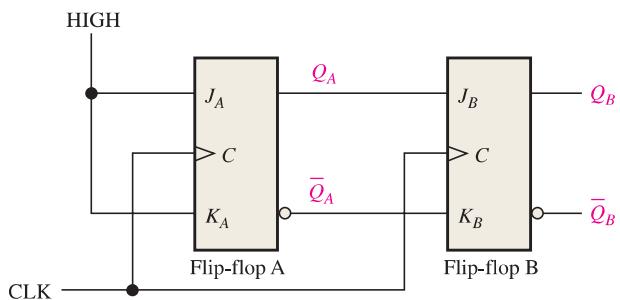


FIGURE 7-89

Section 7-4 Flip-Flop Applications

- 25.** A D flip-flop is connected as shown in Figure 7–90. Determine the Q output in relation to the clock. What specific function does this device perform?

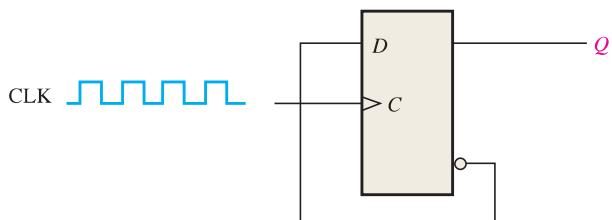


FIGURE 7-90

- 26.** For the circuit in Figure 7–89, develop a timing diagram for eight clock pulses, showing the Q_A and Q_B outputs in relation to the clock.

Section 7-5 One-Shots

- 27.** Determine the pulse width of a 74121 one-shot if the external resistor is $1\text{ k}\Omega$ and the external capacitor is 1 pF .
- 28.** An output pulse of $3\text{ }\mu\text{s}$ duration is to be generated by a 74LS122 one-shot. Using a capacitor of $50,000\text{ pF}$, determine the value of external resistance required.
- 29.** Create a one-shot using a 555 timer that will produce a 0.5 s output pulse.

Section 7-6 The Astable Multivibrator

- 30.** A 555 timer is configured to run as an astable multivibrator as shown in Figure 7–91. Determine its frequency.

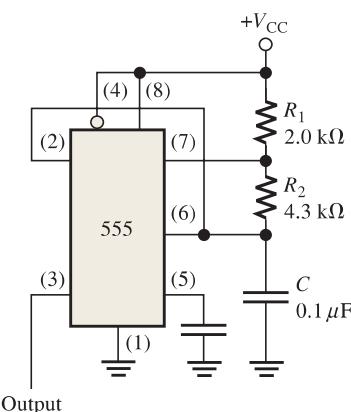


FIGURE 7-91

31. Determine the values of the external resistors for a 555 timer used as an astable multivibrator with an output frequency of 10 kHz, if the external capacitor C is $0.004 \mu\text{F}$ and the duty cycle is to be approximately 80%.

Section 7-7 Troubleshooting

32. The flip-flop in Figure 7-92 is tested under all input conditions as shown. Is it operating properly? If not, what is the most likely fault?

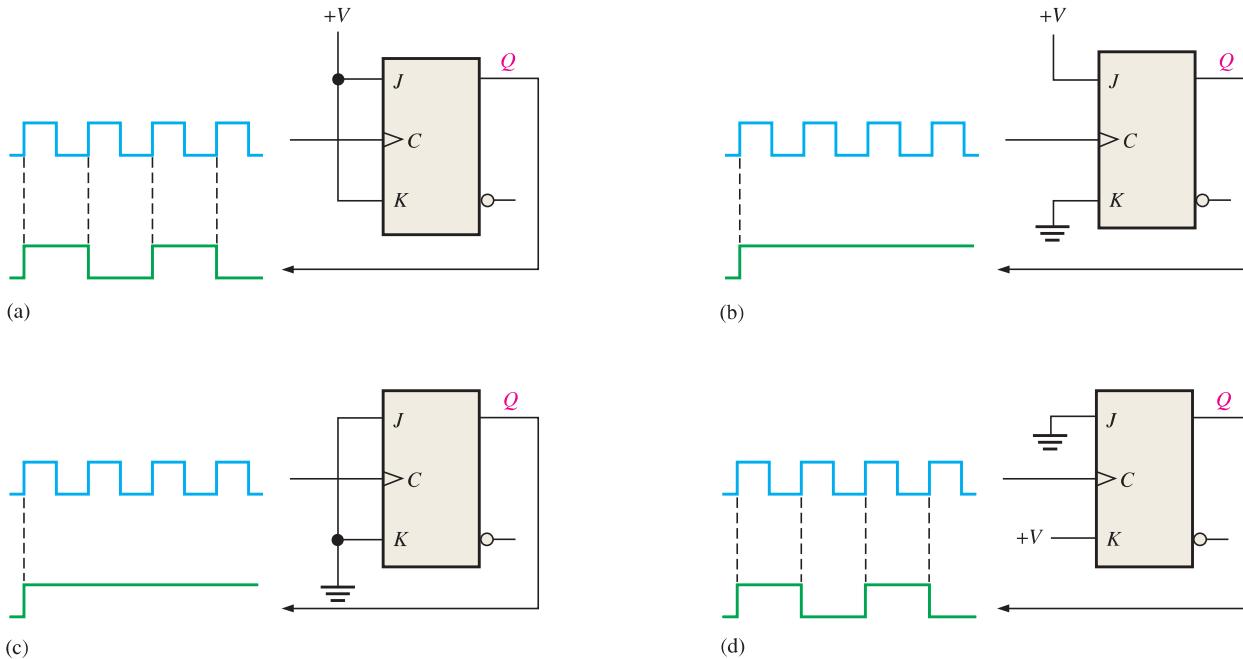


FIGURE 7-92

33. A 74HC00 quad NAND gate IC is used to construct a gated S-R latch on a protoboard in the lab as shown in Figure 7-93. The schematic in part (a) is used to connect the circuit in part (b). When you try to operate the latch, you find that the Q output stays HIGH no matter what the inputs are. Determine the problem.

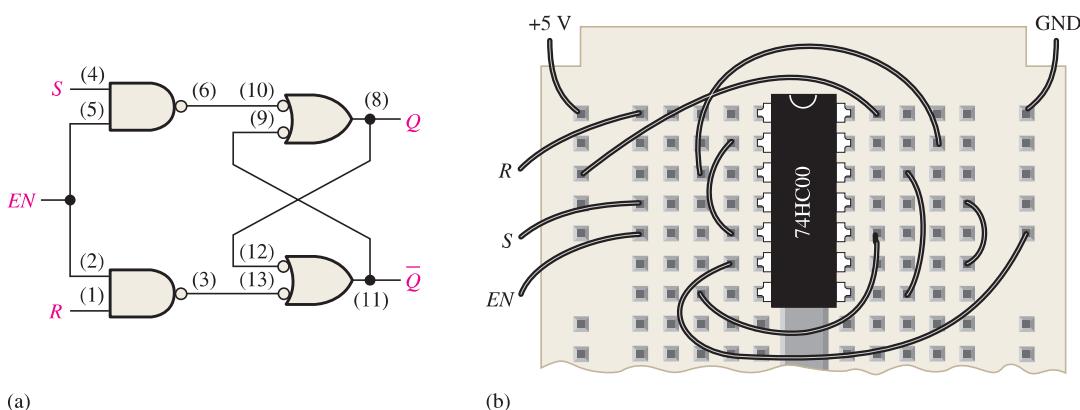


FIGURE 7-93

- 34.** Determine if the flip-flop in Figure 7–94 is operating properly, and if not, identify the most probable fault.

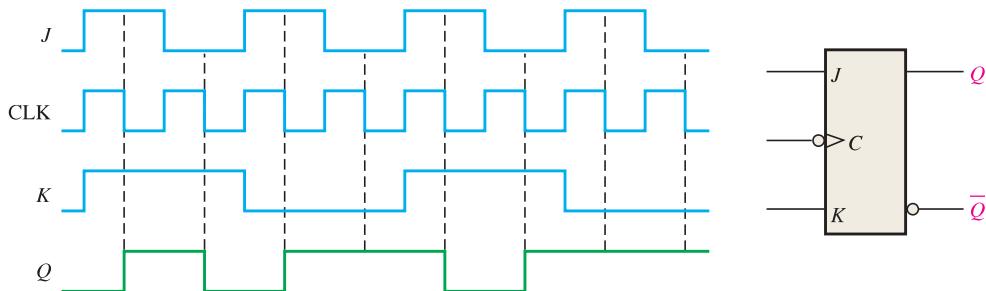


FIGURE 7-94

- 35.** The parallel data storage circuit in Figure 7–35 does not operate properly. To check it out, you first make sure that V_{CC} and ground are connected, and then you apply LOW levels to all the *D* inputs and pulse the clock line. You check the *Q* outputs and find them all to be LOW; so far, so good. Next you apply HIGHs to all the *D* inputs and again pulse the clock line. When you check the *Q* outputs, they are still all LOW. What is the problem, and what procedure will you use to isolate the fault to a single device?
- 36.** The flip-flop circuit in Figure 7–95(a) is used to generate a binary count sequence. The gates form a decoder that is supposed to produce a HIGH when a binary zero or a binary three state occurs (00 or 11). When you check the Q_A and Q_B outputs, you get the display shown in part (b), which reveals glitches on the decoder output (*X*) in addition to the correct pulses. What is causing these glitches, and how can you eliminate them?

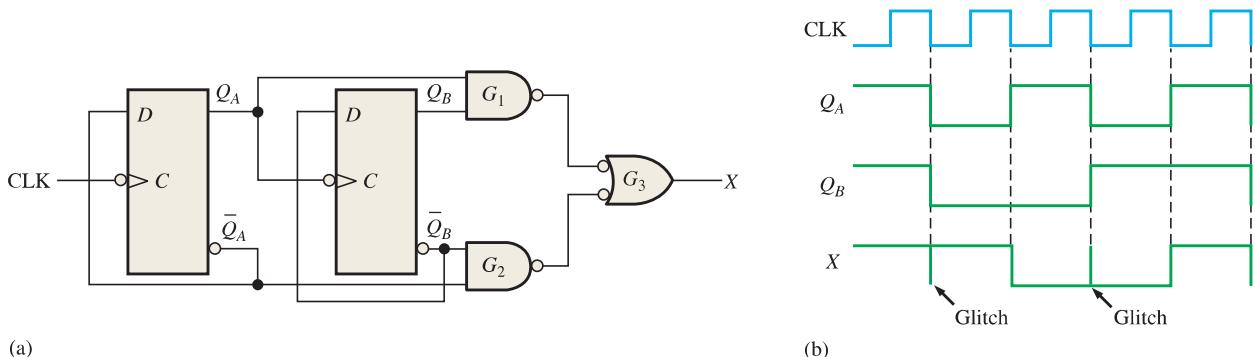


FIGURE 7-95

- 37.** Determine the Q_A , Q_B and X outputs over six clock pulses in Figure 7–95(a) for each of the following faults in the bipolar (TTL) circuits. Start with both Q_A and Q_B LOW.
- D* input open
 - Q_B output open
 - clock input to flip-flop B shorted
 - gate G_2 output open
- 38.** Two 74121 one-shots are connected on a circuit board as shown in Figure 7–96. After observing the oscilloscope display, do you conclude that the circuit is operating properly? If not, what is the most likely problem?

Applied Logic

- Using 555 timers, redesign the timing circuits portion of the traffic signal controller for an approximate 5 s caution light and 30 s red and green lights.
- Repeat Problem 39 using 74121 one-shots.
- Repeat Problem 39 using 74122 one-shots.
- Implement the input logic in the sequential circuit unit of the traffic signal controller using only NAND gates.
- Specify how you would change the time interval for the green light from 25 s to 60 s.

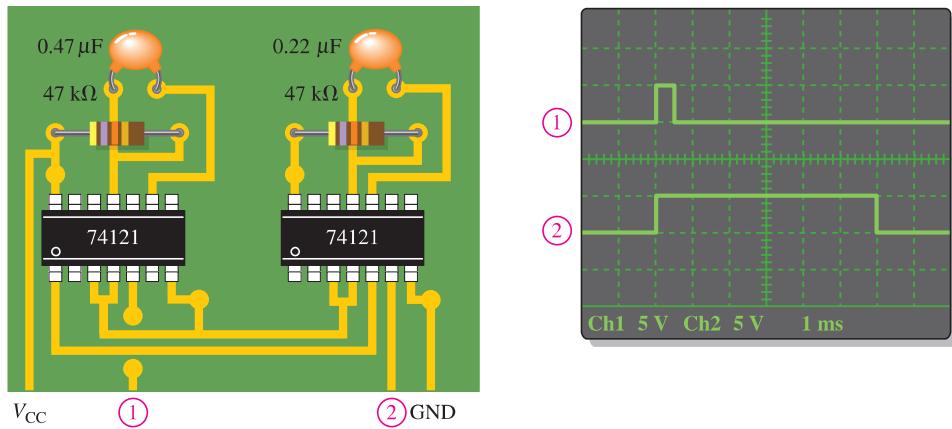


FIGURE 7-96

Special Design Problems

- 44.** Design a basic counting circuit that produces a binary sequence from zero through seven by using negative edge-triggered J-K flip-flops.

45. In the shipping department of a softball factory, the balls roll down a conveyor and through a chute single file into boxes for shipment. Each ball passing through the chute activates a switch circuit that produces an electrical pulse. The capacity of each box is 32 balls. Design a logic circuit to indicate when a box is full so that an empty box can be moved into position.

46. List the design changes that would be necessary in the traffic signal controller to add a 15 s left turn arrow for the main street. The turn arrow will occur after the red light and prior to the green light. Modify the state diagram from Chapter 6 to show these changes.



Multisim Troubleshooting Practice

47. Open file P07-47. For the specified fault, predict the effect on the circuit. Then introduce the fault and verify whether your prediction is correct.
 48. Open file P07-48. For the specified fault, predict the effect on the circuit. Then introduce the fault and verify whether your prediction is correct.
 49. Open file P07-49. For the observed behavior indicated, predict the fault in the circuit. Then introduce the suspected fault and verify whether your prediction is correct.
 50. Open file P07-50. For the observed behavior indicated, predict the fault in the circuit. Then introduce the suspected fault and verify whether your prediction is correct.
 51. Open file P07-51. For the observed behavior indicated, predict the fault in the circuit. Then introduce the suspected fault and verify whether your prediction is correct.

ANSWERS

SECTION CHECKUPS

Section 7-1 Latches

1. Three types of latches are S-R, gated S-R, and gated D.
 2. $SR = 00$, NC; $SR = 01$, $Q = 0$; $SR = 10$, $Q = 1$; $SR = 11$, invalid
 3. $Q \equiv 1$

Section 7-2 Flip-Flops

1. The output of a gated D latch can change any time the gate enable (EN) input is active. The output of an edge-triggered D flip-flop can change only on the triggering edge of a clock pulse.
 2. The output of a J-K flip-flop is determined by the state of its two inputs whereas the output of a D flip-flop follows the input.
 3. Output Q goes HIGH on the trailing edge of the first clock pulse, LOW on the trailing edge of the second pulse, HIGH on the trailing edge of the third pulse, and LOW on the trailing edge of the fourth pulse.

Section 7-3 Flip-Flop Operating Characteristics

1. (a) Set-up time is the time required for input data to be present before the triggering edge of the clock pulse.
- (b) Hold time is the time required for data to remain on the inputs after the triggering edge of the clock pulse.
2. The 74AHC74 can be operated at the highest frequency, according to Table 7-4.

01	00	00	00	00
00	00	10	00	00
00	11	11	11	11
11	11	11	11	11
11	11	11	01	01
11	01	11	01	01
01	01	01	01	01
01	10	00	01	01
01	01	01	01	01
01	01	11	00	00
01	00	11	10	10
00	10	11	10	10
10	10	01	01	00
10	01	10	11	11

Section 7-4 Flip-Flop Applications

1. A group of data storage flip-flops is a register.
2. For divide-by-2 operation, the flip-flop must toggle ($D = \bar{Q}$).
3. Six flip-flops are used in a divide-by-64 device.

Section 7-5 One-Shots

1. A nonretriggerable one-shot times out before it can respond to another trigger input. A retriggerable one-shot responds to each trigger input.
2. Pulse width is set with external R and C components.
3. 11 ms.

Section 7-6 The Astable Multivibrator

1. An astable multivibrator has no stable state. A monostable multivibrator has one stable state.
2. Duty cycle = $(15 \text{ ms}/20 \text{ ms})100\% = 75\%$

Section 7-7 Troubleshooting

1. Yes, a negative edge-triggered J-K flip-flop can be used.
2. An astable multivibrator using a 555 timer can be used to provide the clock.

RELATED PROBLEMS FOR EXAMPLES

7-1 The Q output is the same as shown in Figure 7-5(b).

7-2 See Figure 7-97.

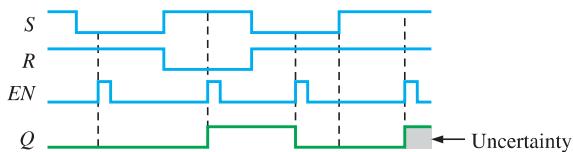


FIGURE 7-97

7-3 See Figure 7-98.

7-4 See Figure 7-99.

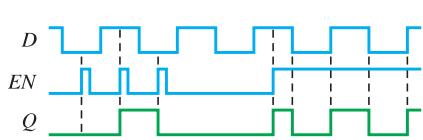


FIGURE 7-98

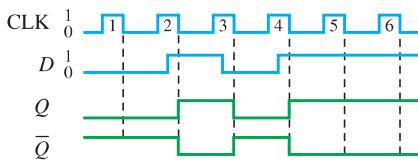


FIGURE 7-99

7-5 See Figure 7-100.

7-6 See Figure 7-101.

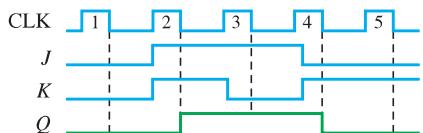


FIGURE 7-100

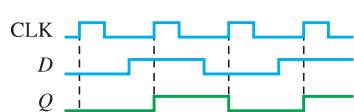


FIGURE 7-101

00 00 00 11
10 11 11 11
11 11 11 11
00 11 11 01
11 11 01 01
01 01 01 10
01 01 10 01
00 10 01 01
00 01 01 00
11 01 00 10
11 10 10 10
11 10 10 00
01 00 00 11
01 00 11 01
10 11 01

7-7 See Figure 7-102.

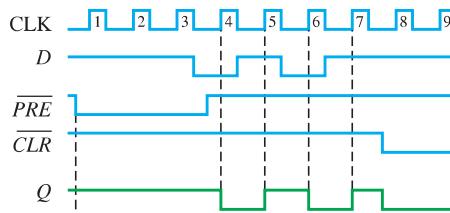


FIGURE 7-102

7-8 See Figure 7-103.

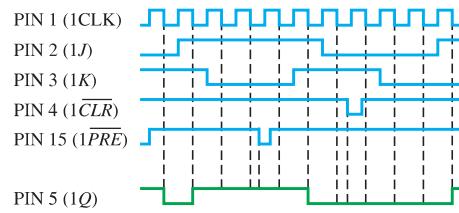


FIGURE 7-103

7-9 $2^5 = 32$. Five flip-flops are required.

7-10 Sixteen states require four flip-flops ($2^4 = 16$).

7-11 $C_{EXT} = 7143 \text{ pF}$ connected from CX to RX/CX of the 74121 with no external resistor.

7-12 $C_{EXT} = 560 \text{ pF}$, $R_{EXT} = 27 \text{ k}\Omega$. See Figure 7-104.

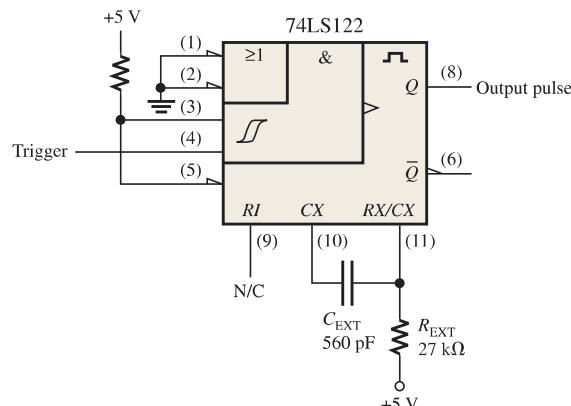


FIGURE 7-104

7-13 $R_1 = 91 \text{ k}\Omega$

7-14 Duty cycle $\approx 32\%$

TRUE/FALSE QUIZ

1. F 2. T 3. F 4. T 5. F 6. T 7. T 8. F 9. T 10. F

SELF-TEST

1. (a) 2. (c) 3. (b) 4. (d) 5. (d) 6. (d)
7. (a) 8. (c) 9. (d) 10. (d) 11. (c) 12. (f)

floor where the elevator is located. As a result of this comparison, either an UP command, a DOWN command, or an OPEN command is issued to the elevator motor control. As the elevator moves toward the desired floor, the floor counter is either incremented at each floor as it goes up or decremented at each floor as it goes down. Once the elevator reaches the desired floor, a STOP/OPEN command is issued to the elevator motor control and to the door control. After a preset time, the delay timer issues a CLOSE signal to the elevator door control. As mentioned, this elevator design is limited to one floor call and one floor request per cycle.

Initialization The initial one-time setup requires that the elevator be placed at the basement level and the floor counter be preset to 000. After this, the counter will automatically move through the sequence of states determined by the elevator position.

Exercise

1. Explain the purpose of the floor counter.
2. Describe what happens during the WAIT mode.
3. How does the system know when the desired floor has been reached?
4. Discuss the limitations of the elevator design in Figure 9–64.

Implementation

The elevator controller can be implemented using fixed-function logic devices, a PLD programmed with a VHDL (or Verilog) code, or a programmed microcontroller or microprocessor. In the Chapter 10 Applied Logic, the VHDL program code for the elevator controller is presented. You will see how to program a PLD step by step.

Putting Your Knowledge to Work

What changes are required in the logic diagram of Figure 9–64 to upgrade the elevator controller for a ten-story building?

SUMMARY

- Asynchronous and synchronous counters differ only in the way in which they are clocked. The first stage of an asynchronous counter is driven by a clock pulse. Each succeeding stage is clocked by the output of the previous stage. In a synchronous counter, all stages are clocked by the same clock pulse. Synchronous counters can run at faster clock rates than asynchronous counters.
- The maximum modulus of a counter is the maximum number of possible states and is a function of the number of stages (flip-flops). Thus,

$$\text{Maximum modulus} = 2^n$$

where n is the number of stages in the counter. The modulus of a counter is the *actual* number of states in its sequence and can be equal to or less than the maximum modulus.

- The overall modulus of cascaded counters is equal to the product of the moduli of the individual counters.

KEY TERMS

Key terms and other bold terms in the chapter are defined in the end-of-book glossary.

Asynchronous Not occurring at the same time.

Cascade To connect “end-to-end” as when several counters are connected from the terminal count output of one counter to the enable input of the next counter.

00 00 00 00
00 00 00 10 00
00 11 11 11
11 11 11 11
11 11 11 11
11 11 11 11
11 11 00 01
11 01 01 01
01 01 01 01
01 10 00 10
10 01 00 01
01 01 11 00
01 00 11 00
00 10 11 10
10 10 01 00
10 00 00 01
00 11 10 11



Decade Characterized by ten states or values.

Modulus The number of unique states through which a counter will sequence.

Recycle To undergo transition (as in a counter) from the final or terminal state back to the initial state.

State diagram A graphic depiction of a sequence of states or values.

State machine A logic system or circuit exhibiting a sequence of states conditioned by internal logic and external inputs; any sequential circuit exhibiting a specified sequence of states. Two types of state machine are Moore and Mealy.

Synchronous Occurring at the same time.

Terminal count The final state in a counter's sequence.

TRUE/FALSE QUIZ

Answers are at the end of the chapter.

1. A state machine is a sequential circuit having a limited number of states occurring in a prescribed order.
2. Synchronous counters cannot be realized using J-K flip-flops.
3. An asynchronous counter is also known as a ripple counter.
4. A decade counter has twelve states.
5. A counter with four stages has a maximum modulus of sixteen.
6. To achieve a maximum modulus of 32, sixteen stages are required.
7. If the present state is 1000, the next state of a 4-bit up/down counter in the DOWN mode is 0111.
8. Two cascaded decade counters divide the clock frequency by 10.
9. A counter with a truncated sequence has less than its maximum number of states.
10. To achieve a modulus of 100, ten decade counters are required.

SELF-TEST

Answers are at the end of the chapter.

1. A Moore state machine consists of combinational logic circuits that determine

(a) sequences	(b) memory
(c) both (a) and (b)	(d) neither (a) nor (b)
2. The output of a Mealy machine depends on its

(a) inputs	(b) next state
(c) present state	(d) answers (a) and (c)
3. The maximum cumulative delay of an asynchronous counter must be

(a) more than the period of the clock waveform	(b) less than the period of the clock waveform
(c) equal to the period of the clock waveform	(d) both (a) and (c)
4. A decade counter with a count of zero (0000) through nine (1001) is known as

(a) an ASCII counter	(b) a binary counter
(c) A BCD counter	(d) a decimal counter
5. The modulus of a counter is

(a) the number of flip-flops	(b) the actual number of states in its sequence
(c) the number of times it recycles in a second	(d) the maximum possible number of states
6. A 3-bit binary counter has a maximum modulus of

(a) 3	(b) 6
(c) 8	(d) 16
7. A 5-bit binary counter has a maximum modulus of

(a) 4	(b) 8
(c) 16	(d) 32
8. A modulus-12 counter must have

(a) 12 flip-flops	(b) 3 flip-flops
(c) 4 flip-flops	(d) synchronous clocking

9. Which one of the following is an example of a counter with a truncated modulus?
- Modulus 8
 - Modulus 14
 - Modulus 16
 - Modulus 32
10. A 4-bit ripple counter consists of flip-flops that each have a propagation delay from clock to Q output of 12 ns. For the counter to recycle from 1111 to 0000, it takes a total of
- 12 ns
 - 24 ns
 - 48 ns
 - 36 ns
11. A BCD counter is an example of
- a full-modulus counter
 - a decade counter
 - a truncated-modulus counter
 - answers (b) and (c)
12. Which of the following is a valid state in an 8421 BCD counter?
- 1010
 - 1011
 - 1111
 - 1000
13. Three cascaded modulus-10 counters have an overall modulus of
- 30
 - 100
 - 1000
 - 10,000
14. A 10 MHz clock frequency is applied to a cascaded counter consisting of a modulus-5 counter, a modulus-8 counter, and two modulus-10 counters. The lowest output frequency possible is
- 10 kHz
 - 2.5 kHz
 - 5 kHz
 - 25 kHz
15. A 4-bit binary up/down counter is in the binary state of zero. The next state in the DOWN mode is
- 0001
 - 1111
 - 1000
 - 1110
16. The initial count of a modulus-13 binary counter is
- 0000
 - 1111
 - 1101
 - 1100

PROBLEMS

Answers to odd-numbered problems are at the end of the book.

Section 9–1 Finite State Machines

- Represent a decade counter with the terminal state decoded as a state machine. Identify the type and show the block diagram and the state diagram.
- Identify the type of state machine for the traffic signal controller in Chapter 6. State the reason why it is the type you specified.

Section 9–2 Asynchronous Counters

- For the ripple counter shown in Figure 9–65, show the complete timing diagram for eight clock pulses, showing the clock, Q_0 , and Q_1 waveforms.

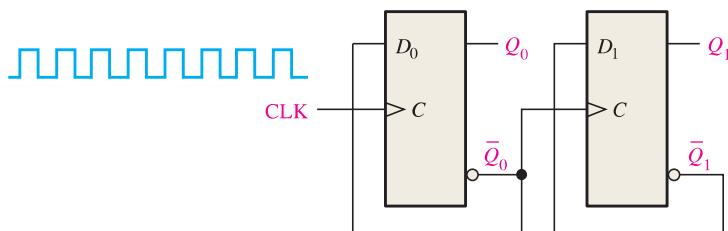


FIGURE 9–65

00	00	11
10	00	11
11	11	11
11	11	00
00	11	01
11	01	01
01	01	01
01	10	01
00	01	01
00	01	00
11	00	10
11	10	10
01	10	00
01	00	11
01	11	01
10	11	01

4. For the ripple counter in Figure 9–66, show the complete timing diagram for sixteen clock pulses. Show the clock, Q_0 , Q_1 , and Q_2 waveforms.

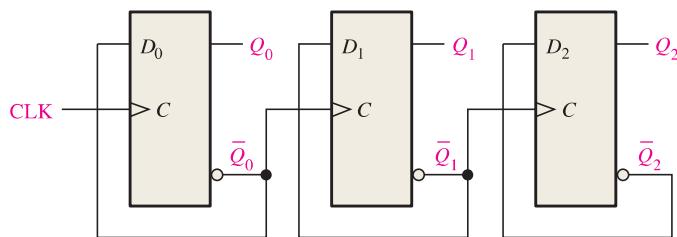


FIGURE 9–66

5. In the counter of Problem 4, assume that each flip-flop has a propagation delay from the triggering edge of the clock to a change in the Q output of 8 ns. Determine the worst-case (longest) delay time from a clock pulse to the arrival of the counter in a given state. Specify the state or states for which this worst-case delay occurs.
 6. Show how to connect a 74HC93 4-bit asynchronous counter for each of the following moduli:
 (a) 9 (b) 11 (c) 13 (d) 14 (e) 15

Section 9–3 Synchronous Counters

7. If the counter of Problem 5 were synchronous rather than asynchronous, what would be the longest delay time?
 8. Show the complete timing diagram for the 5-stage synchronous binary counter in Figure 9–67. Verify that the waveforms of the Q outputs represent the proper binary number after each clock pulse.

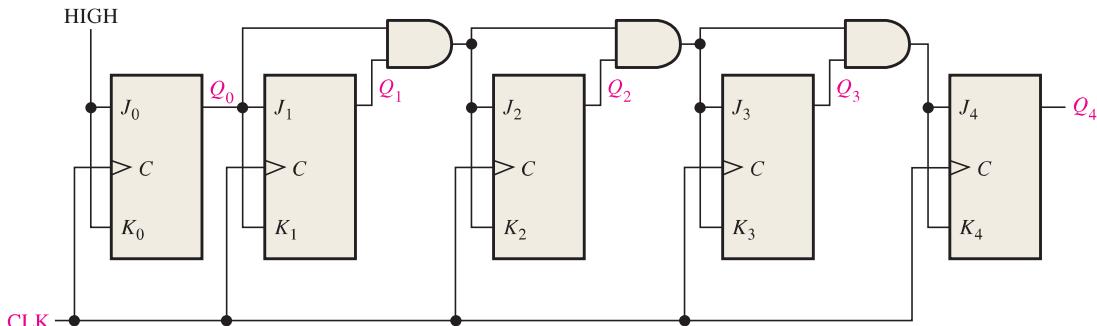


FIGURE 9–67

9. By analyzing the J and K inputs to each flip-flop prior to each clock pulse, prove that the decade counter in Figure 9–68 progresses through a BCD sequence. Explain how these conditions in each case cause the counter to go to the next proper state.

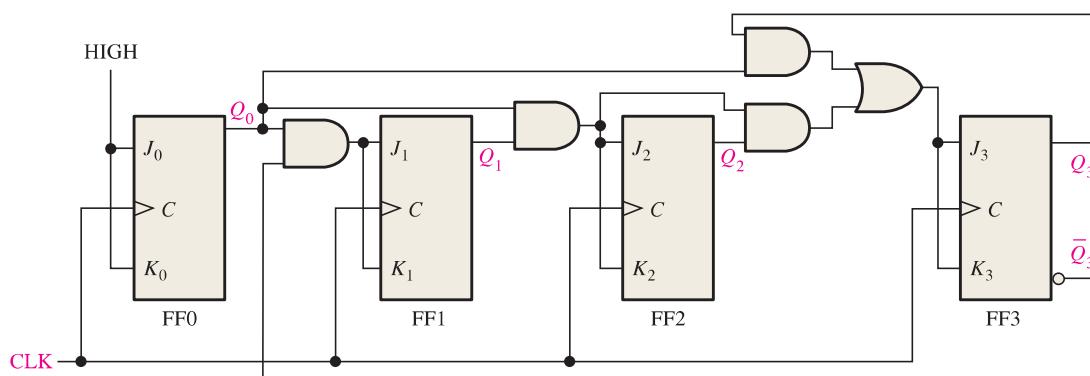


FIGURE 9–68

10. The waveforms in Figure 9–69 are applied to the count enable, clear, and clock inputs as indicated. Show the counter output waveforms in proper relation to these inputs. The clear input is asynchronous.

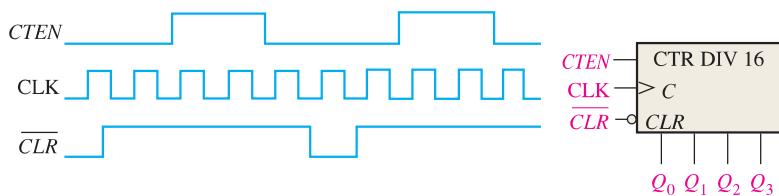


FIGURE 9-69

11. A BCD decade counter is shown in Figure 9–70. The waveforms are applied to the clock and clear inputs as indicated. Determine the waveforms for each of the counter outputs (*Q₀*, *Q₁*, *Q₂*, and *Q₃*). The clear is synchronous, and the counter is initially in the binary 1000 state.

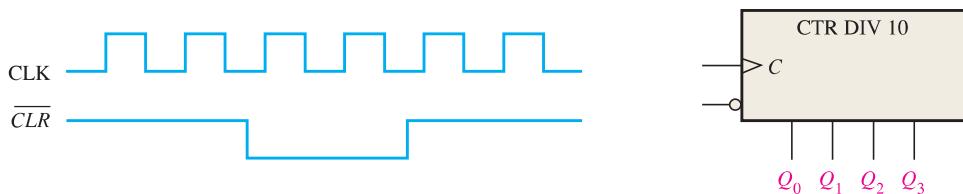


FIGURE 9-70

12. The waveforms in Figure 9–71 are applied to a 74HC163 binary counter. Determine the *Q* outputs and the *RCO*. The inputs are *D₀* = 1, *D₁* = 1, *D₂* = 0, and *D₃* = 1.

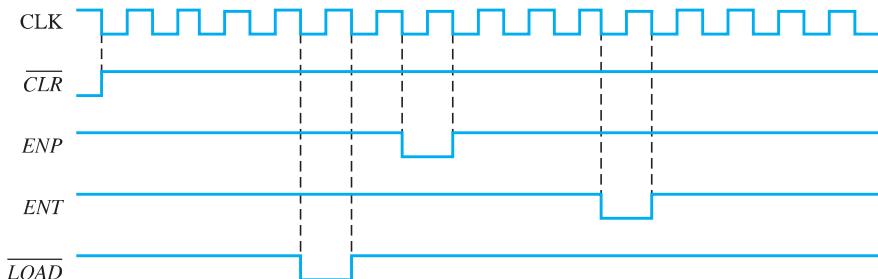


FIGURE 9-71

13. The waveforms in Figure 9–71 are applied to a 74HC161 counter. Determine the *Q* outputs and the *RCO*. The inputs are *D₀* = 1, *D₁* = 0, *D₂* = 0, and *D₃* = 1.

Section 9-4 Up/Down Synchronous Counters

14. Show a complete timing diagram for a 3-bit up/down counter that goes through the following sequence. Indicate when the counter is in the UP mode and when it is in the DOWN mode. Assume positive edge-triggering.

0, 1, 2, 3, 2, 1, 2, 3, 4, 5, 6, 5, 4, 3, 2, 1, 0

15. Develop the *Q* output waveforms for a 74HC190 up/down counter with the input waveforms shown in Figure 9–72. A binary 0 is on the data inputs. Start with a count of 0000.

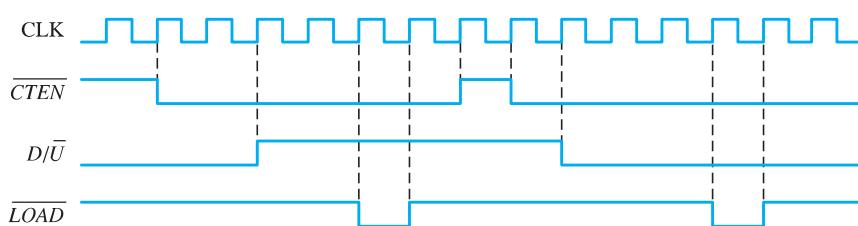


FIGURE 9-72

16. Repeat Problem 15 if the D/\bar{U} input signal is inverted with the other inputs the same.
 17. Repeat Problem 15 if the \overline{CTEN} is inverted with the other inputs the same.

Section 9–5 Design of Synchronous Counters

18. Determine the sequence of the counter in Figure 9–73.

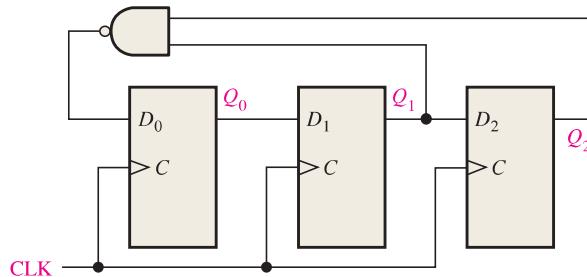


FIGURE 9–73

19. Determine the sequence of the counter in Figure 9–74. Begin with the counter cleared.

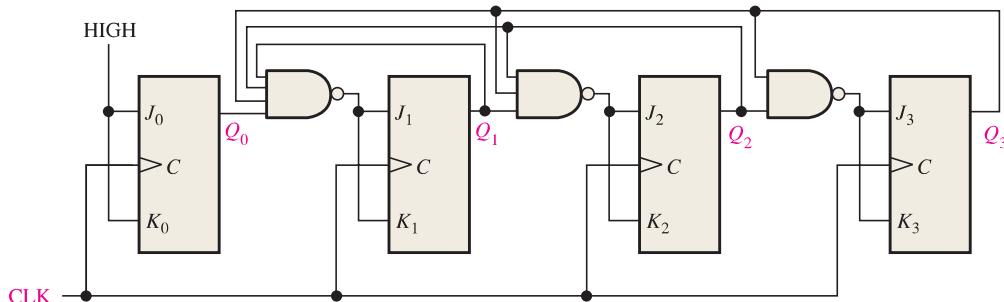


FIGURE 9–74

20. Design a counter to produce the following sequence. Use J-K flip-flops.

00, 10, 01, 11, 00, ...

21. Design a counter to produce the following binary sequence. Use J-K flip-flops.

1, 4, 3, 5, 7, 6, 2, 1, ...

22. Design a counter to produce the following binary sequence. Use J-K flip-flops.

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

23. Design a binary counter with the sequence shown in the state diagram of Figure 9–75.

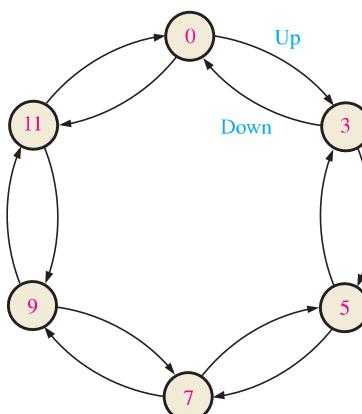


FIGURE 9–75

Section 9–6 Cascaded Counters

24. For each of the cascaded counter configurations in Figure 9–76, determine the frequency of the waveform at each point indicated by a circled number, and determine the overall modulus.

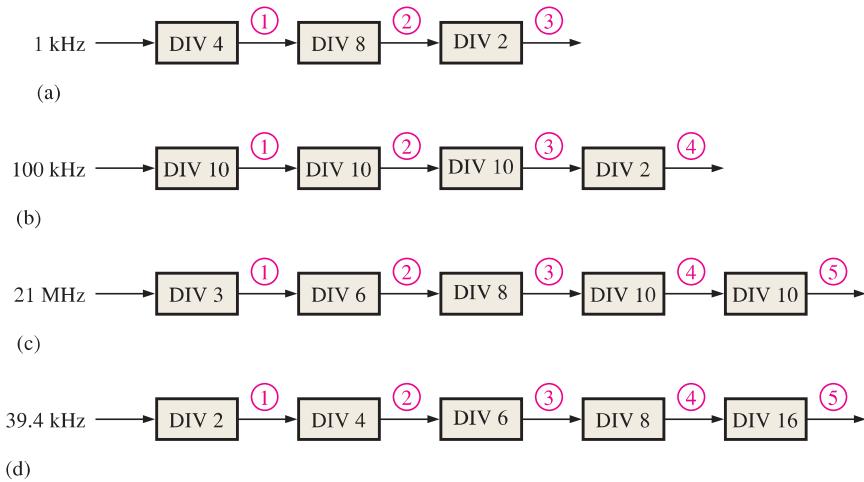


FIGURE 9–76

25. Expand the counter in Figure 9–38 to create a divide-by-10,000 counter and a divide-by-100,000 counter.
26. With general block diagrams, show how to obtain the following frequencies from a 10 MHz clock by using single flip-flops, modulus-5 counters, and decade counters:
- | | | | | |
|-------------|--------------|------------|------------|-------------|
| (a) 5 MHz | (b) 2.5 MHz | (c) 2 MHz | (d) 1 MHz | (e) 500 kHz |
| (f) 250 kHz | (g) 62.5 kHz | (h) 40 kHz | (i) 10 kHz | (j) 1 kHz |

Section 9–7 Counter Decoding

27. Given a BCD decade counter with only the Q outputs available, show what decoding logic is required to decode each of the following states and how it should be connected to the counter. A HIGH output indication is required for each decoded state. The MSB is to the left.
- | | | | | |
|----------|----------|----------|----------|----------|
| (a) 0001 | (b) 0011 | (c) 0101 | (d) 0111 | (e) 1000 |
|----------|----------|----------|----------|----------|
28. For the 4-bit binary counter connected to the decoder in Figure 9–77, determine each of the decoder output waveforms in relation to the clock pulses.

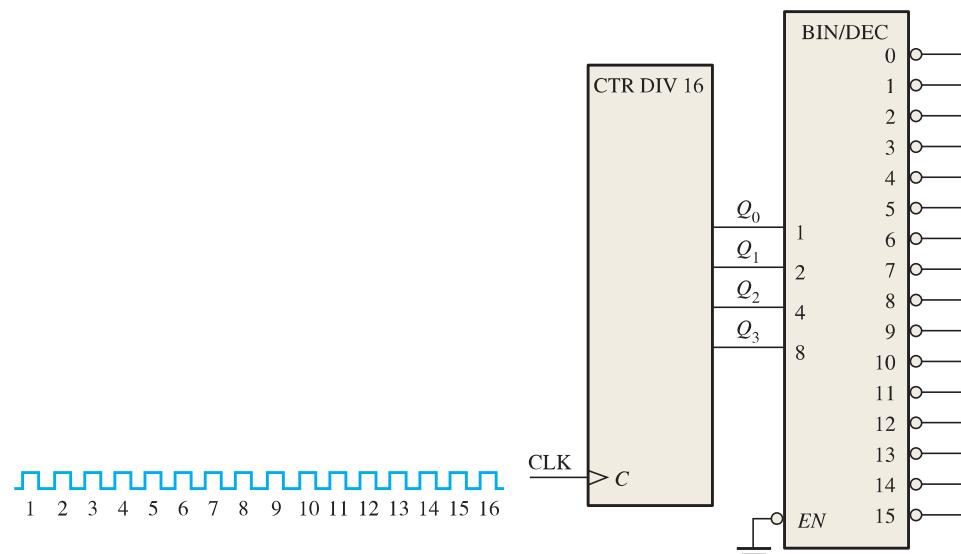


FIGURE 9–77

00	00	11
10	00	11
11	11	11
00	11	11
11	11	01
01	01	10
01	01	10
00	10	01
00	01	01
01	01	00
11	00	10
11	10	10
01	10	00
01	00	11
01	11	01
10	11	01

29. If the counter in Figure 9–77 is asynchronous, determine where the decoding glitches occur on the decoder output waveforms.
30. Modify the circuit in Figure 9–77 to eliminate decoding glitches.
31. Analyze the counter in Figure 9–42 for the occurrence of glitches on the decode gate output. If glitches occur, suggest a way to eliminate them.
32. Analyze the counter in Figure 9–43 for the occurrence of glitches on the outputs of the decoding gates. If glitches occur, make a design change that will eliminate them.

Section 9–8 Counter Applications

33. Assume that the digital clock of Figure 9–48 is initially reset to 12 o'clock. Determine the binary state of each counter after sixty-two 60 Hz pulses have occurred.
34. What is the output frequency of each counter in the digital clock circuit of Figure 9–48?
35. For the automobile parking control system in Figure 9–51, a pattern of entrance and exit sensor pulses during a given 24-hour period are shown in Figure 9–78. If there were 53 cars already in the garage at the beginning of the period, what is the state of the counter at the end of the 24 hours?

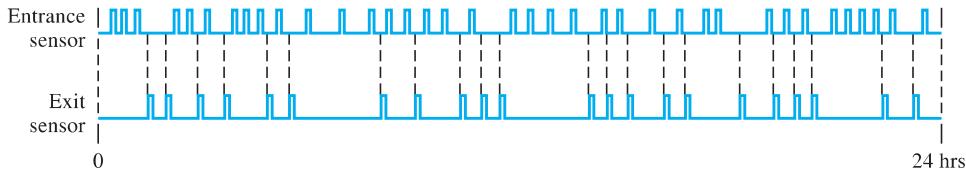


FIGURE 9–78

36. The binary number for decimal 57 appears on the parallel data inputs of the parallel-to-serial converter in Figure 9–53 (D_0 is the LSB). The counter initially contains all zeros and a 10 kHz clock is applied. Develop the timing diagram showing the clock, the counter outputs, and the serial data output.

Section 9–10 Troubleshooting

37. For the counter in Figure 9–4, show the timing diagram for the Q_0 and Q_1 waveforms for each of the following faults (assume Q_0 and Q_1 are initially LOW):
 - clock input to FF0 shorted to ground
 - Q_0 output open
 - clock input to FF1 open
 - D input to FF0 open
 - D input to FF1 shorted to ground
38. Solve Problem 37 for the counter in Figure 9–12(b).
39. Isolate the fault in the counter in Figure 9–6 by analyzing the waveforms in Figure 9–79.
40. From the waveform diagram in Figure 9–80, determine the most likely fault in the counter of Figure 9–15.

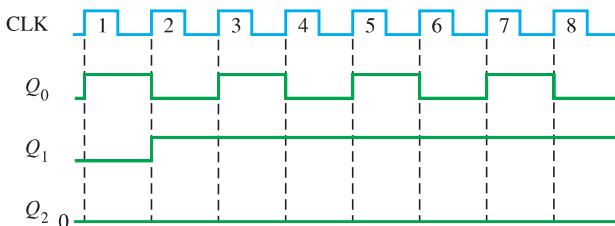


FIGURE 9–79

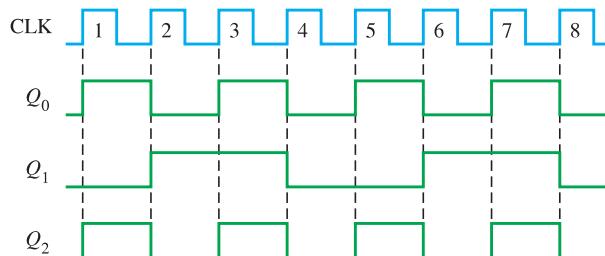


FIGURE 9–80

- 41.** Solve Problem 40 if the Q_2 output has the waveform observed in Figure 9–81. Outputs Q_0 and Q_1 are the same as in Figure 9–80.

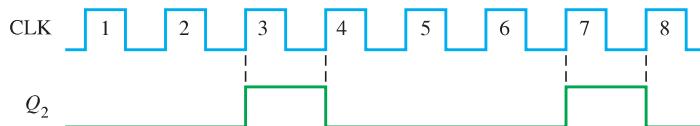


FIGURE 9-81

- 42.** You apply a 5 MHz clock to the cascaded counter in Figure 9–41 and measure a frequency of 76.2939 Hz at the last RCO output. Is this correct, and if not, what is the most likely problem?
- 43.** Develop a table for use in testing the counter in Figure 9–41 that will show the frequency at the final RCO output for all possible open failures of the parallel data inputs (D_0 , D_1 , D_2 , and D_3) taken one at a time. Use 10 MHz as the test frequency for the clock.
- 44.** The tens-of-hours 7-segment display in the digital clock system of Figure 9–48 continuously displays a 1. All the other digits work properly. What could be the problem?
- 45.** What would be the visual indication of an open Q_1 output in the tens portion of the minutes counter in Figure 9–48? Also see Figure 9–49.
- 46.** One day (perhaps a Monday) complaints begin flooding in from patrons of a parking garage that uses the control system depicted in Figures 9–51 and 9–52. The patrons say that they enter the garage because the gate is up and the FULL sign is off but that, once in, they can find no empty space. As the technician in charge of this facility, what do you think the problem is, and how will you troubleshoot and repair the system as quickly as possible?

Applied Logic

- 47.** Propose a general design for generation of the 3-bit FLRCALL code and the Call pulse by the pressing of a single button.
- 48.** Propose a general design for generation of the 3-bit FLRREQ code and the Request pulse by the pressing of one of seven buttons.
- 49.** What changes are required to the logic diagram in Figure 9–64 to modify the elevator controller for a four-story building?

Special Design Problems

- 50.** Design a modulus-1000 counter by using decade counters.
- 51.** Modify the design of the counter in Figure 9–41 to achieve a modulus of 30,000.
- 52.** Repeat Problem 51 for a modulus of 50,000.
- 53.** Modify the digital clock in Figures 9–48, 9–49, and 9–50 so that it can be preset to any desired time.
- 54.** Design an alarm circuit for the digital clock that can detect a predetermined time (hours and minutes only) and produce a signal to activate an audio alarm.
- 55.** Modify the design of the circuit in Figure 9–52 for a 1000-space parking garage and a 3000-space parking garage.
- 56.** Implement the parallel-to-serial data conversion logic in Figure 9–53 with specific fixed-function devices.
- 57.** In Problem 19 it was found that the counter locks up and alternates between two states. It turns out that this operation is the result of a design flaw. Redesign the counter so that when it goes into the second of the lock-up states, it will recycle to the all-0s state on the next clock pulse.

Multisim Troubleshooting Practice

- 58.** Open file P09-58. For the specified fault, predict the effect on the circuit. Then introduce the fault and verify whether your prediction is correct.
- 59.** Open file P09-59. For the specified fault, predict the effect on the circuit. Then introduce the fault and verify whether your prediction is correct.
- 60.** Open file P09-60. For the specified fault, predict the effect on the circuit. Then introduce the fault and verify whether your prediction is correct.



01	00	00	00	00
00	00	10	00	00
00	11	11	11	11
11	11	11	11	11
11	11	11	11	11
11	01	01	01	01
01	01	01	01	10
01	01	00	01	01
10	10	00	01	01
01	01	11	00	01
01	00	11	10	00
00	10	11	10	10
10	10	01	10	10
10	00	01	00	00
00	11	10	11	11



ANSWERS

61. Open file P09-61. For the observed behavior indicated, predict the fault in the circuit. Then introduce the suspected fault and verify whether your prediction is correct.
62. Open file P09-62. For the observed behavior indicated, predict the fault in the circuit. Then introduce the suspected fault and verify whether your prediction is correct.

SECTION CHECKUPS

Section 9–1 Checkup

1. A finite state machine is a sequential circuit having a finite number of states that occur in a specified order.
2. Moore state machine and Mealy state machine
3. The Moore state machine has an output(s) that is dependent on the present internal state only. The Mealy state machine has an output(s) that is dependent on both the present internal state and the value of the inputs.

Section 9–2 Asynchronous Counters

1. Asynchronous means that each flip-flop after the first one is enabled by the output of the preceding flip-flop.
2. A modulus-14 counter has fourteen states requiring four flip-flops.

Section 9–3 Synchronous Counters

1. All flip-flops in a synchronous counter are clocked simultaneously.
2. The counter can be preset (initialized) to any given state.
3. Counter is enabled when *ENP* and *ENT* are both HIGH; *RCO* goes HIGH when final state in sequence is reached.

Section 9–4 Up/Down Synchronous Counters

1. The counter goes to 1001.
2. UP: 1111; DOWN: 0000; the next state is 1111.

Section 9–5 Design of Synchronous Counters

1. $J = 1, K = X$ (“don’t care”)
2. $J = X$ (“don’t care”), $K = 0$
3. (a) The next state is 1011.
 (b) Q_3 (MSB): no-change or SET; Q_2 : no-change or RESET; Q_1 : no change or SET;
 Q_0 (LSB): SET or toggle

Section 9–6 Cascaded Counters

1. Three decade counters produce $\div 1000$; 4 decade counters produce $\div 10,000$.
2. (a) $\div 20$: flip-flop and DIV 10
 (b) $\div 32$: flip-flop and DIV 16
 (c) $\div 160$: DIV 16 and DIV 10
 (d) $\div 320$: DIV 16 and DIV 10 and flip-flop

Section 9–7 Counter Decoding

1. (a) No transitional states because there is a single bit change
 (b) 0000, 0001, 0010, 0101, 0110, 0111
 (c) No transitional states because there is a single bit change
 (d) 0001, 0010, 0011, 0100, 0101, 0110, 0111, 1000, 1001, 1010, 1011, 1100, 1101, 1110

Section 9–8 Counter Applications

- Gate G_1 resets flip-flop on first clock pulse after count 9. Gate G_2 decodes count 12 to preset counter to 0001.
- The hours decade counter advances through each state from zero to nine, and as it recycles from nine back to zero, the flip-flop is toggled to the SET state. This produces a ten (10) on the display. When the hours decade counter is in state 12, the decode NAND gate causes the counter to recycle to state 1 on the next clock pulse. The flip-flop resets. This results in a one (01) on the display.

01	00	00	00
00	00	10	00
00	11	11	11
11	11	00	11
11	11	11	11
11	01	11	01
01	01	01	01
01	10	00	01
10	01	00	01
01	01	11	00
01	00	11	10
00	10	11	10
10	10	01	00
10	01	10	11
00	11	10	11

Section 9–9 Logic Symbols with Dependency Notation

- C : control, usually clock; M : mode; G : AND
- D indicates data storage.

Section 9–10 Troubleshooting

- No pulses on TC outputs: $CTEN$ of first counter shorted to ground or to a LOW; clock input of first counter open; clock line shorted to ground or to a LOW; TC output of first counter shorted to ground or to a LOW.
- With inverter output open, the counter does not recycle at the preset count but acts as a full-modulus counter.

RELATED PROBLEMS FOR EXAMPLES

9–1 See Figure 9–82.

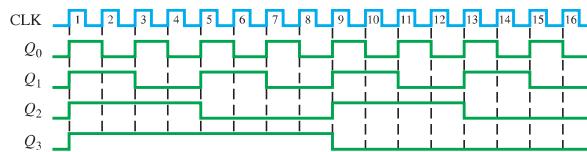


FIGURE 9–82

9–2 Connect Q_0 to the NAND gate as a third input (Q_2 and Q_3 are two of the inputs). Connect the \overline{CLR} line to the \overline{CLR} input of FF0 as well as FF2 and FF3.

9–3 See Figure 9–83.

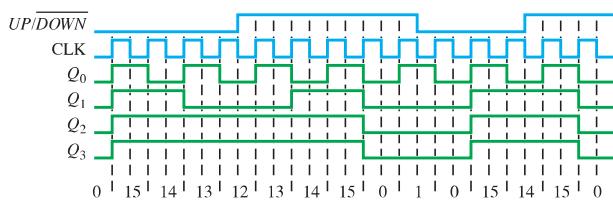


FIGURE 9–83

9–4 See Table 9–14.

TABLE 9–14

Present Invalid State			D Inputs			Next State		
Q_2	Q_1	Q_0	D_2	D_1	D_0	Q_2	Q_1	Q_0
0	0	0	1	1	1	1	1	1 valid state
0	1	1	0	0	0	0	0	0
1	0	0	1	1	1	1	1	1
1	1	0	1	0	1	1	0	1 valid state

000 → 111

011 → 000 → 111

100 → 111

110 → 101

```

00 00 00 11
10 11 11 11
11 11 11 11
00 11 11 01
11 11 01 01
01 01 01 10
01 01 10 01
00 10 01 01
00 01 01 01
11 01 01 00
11 00 10 10
11 10 10 00
01 10 00 11
01 00 11 01
10 11 01

```

9-5 Three flip-flops, sixteen 3-input AND gates, two 4-input OR gates, four 2-input OR gates, and one inverter

9-6 Five decade counters are required. $10^5 = 100,000$

9-7 $f_{Q0} = 1 \text{ MHz}/[(10)(2)] = 50 \text{ kHz}$

9-8 See Figure 9-84.

9-9 8AC0₁₆ would be loaded. $16^4 - 8AC0_{16} = 65,536 - 32,520 = 30,016$

$$f_{TC4} = 10 \text{ MHz}/30,016 = 333.2 \text{ Hz}$$

9-10 See Figure 9-85.

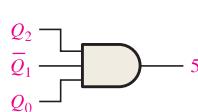


FIGURE 9-84

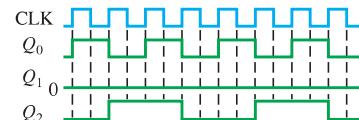


FIGURE 9-85

TRUE/FALSE QUIZ

1. T 2. F 3. T 4. F 5. T 6. F 7. T 8. F 9. T 10. F

SELF-TEST

- | | | | | | | | |
|--------|---------|---------|---------|---------|---------|---------|---------|
| 1. (c) | 2. (a) | 3. (b) | 4. (c) | 5. (b) | 6. (c) | 7. (d) | 8. (c) |
| 9. (b) | 10. (c) | 11. (d) | 12. (d) | 13. (c) | 14. (b) | 15. (b) | 16. (a) |