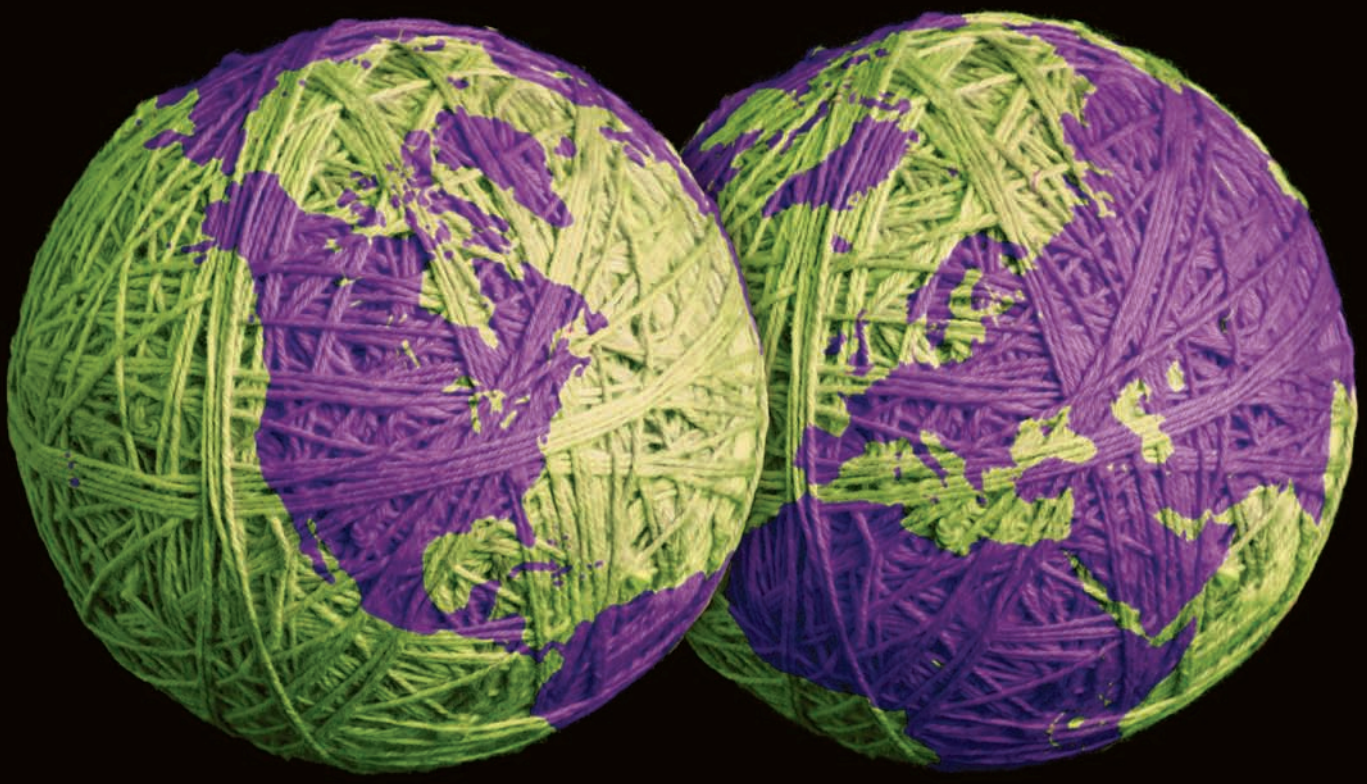
Pearson New International Edition

Digital Electronics
A Practical Approach with VHDL

William Kleitz

Ninth Edition



ALWAYS LEARNING

PEARSON

Pearson New International Edition

Digital Electronics
A Practical Approach with VHDL
William Kleitz
Ninth Edition

PEARSON

Pearson Education Limited
Edinburgh Gate
Harlow
Essex CM20 2JE
England and Associated Companies throughout the world

Visit us on the World Wide Web at: www.pearsoned.co.uk

ISBN 10: 1-292-02561-1
ISBN 13: 978-1-292-02561-2

British Library Cataloguing-in-Publication Data
A catalogue record for this book is available from the British Library

Printed in the United States of America

Table of Contents

Number Systems and Codes

OUTLINE

1 Digital versus Analog
2 Digital Representations of Analog Quantities
3 Decimal Numbering System (Base 10)
4 Binary Numbering System (Base 2)
5 Decimal-to-Binary Conversion
6 Octal Numbering System (Base 8)
7 Octal Conversions
8 Hexadecimal Numbering System (Base 16)
9 Hexadecimal Conversions
10 Binary-Coded-Decimal System
11 Comparison of Numbering Systems
12 The ASCII Code
13 Applications of the Numbering Systems

OBJECTIVES

Upon completion of this chapter, you should be able to do the following:

- Determine the weighting factor for each digit position in the decimal, binary, octal, and hexadecimal numbering systems.
- Convert any number in one of the four number systems (decimal, binary, octal, and hexadecimal) to its equivalent value in any of the remaining three numbering systems.
- Describe the format and use of binary-coded decimal (BCD) numbers.
- Determine the ASCII code for any alphanumeric data by using the ASCII code translation table.

The companion website for this text is www.pearsonhighered.com/kleitz

INTRODUCTION

Digital circuitry is the foundation of digital computers and many automated control systems. In a modern home, digital circuitry controls the appliances, alarm systems, and heating systems. Under the control of digital circuitry and microprocessors, newer automobiles have added safety features, are more energy efficient, and are easier to diagnose and correct when malfunctions arise.

Other uses of digital circuitry include the areas of automated machine control, energy monitoring and control, inventory management, medical electronics, and music. For example, the numerically controlled (NC) milling machine can be programmed by a production engineer to mill a piece of stock material to prespecified dimensions with very accurate repeatability, within 0.01% accuracy. Another use is energy monitoring and control. With the high cost of energy, it is very important for large industrial and commercial users to monitor the energy flows within their buildings. Effective control of heating, ventilating, and air-conditioning can reduce energy bills significantly. More and more grocery stores are using the universal product code (UPC) to check out and total the sale of grocery orders as well as to control inventory and replenish stock automatically. The area of medical electronics uses digital thermometers, life-support systems, and monitors. We have also seen more use of digital electronics in the reproduction of music. Digital reproduction is less susceptible to electrostatic noise and therefore can reproduce music with greater fidelity.

Digital electronics evolved from the principle that transistor circuitry could easily be fabricated and designed to output one of two voltage levels based on the levels placed at its inputs. The two distinct levels (usually +5 volts [V] and 0 V) are HIGH and LOW and can be represented by 1 and 0.

The binary numbering system is made up of only 1s and 0s and is therefore used extensively in digital electronics. The other numbering systems and codes covered in this chapter represent groups of binary digits and therefore are also widely used.

1 Digital versus Analog

Digital systems operate on discrete digits that represent numbers, letters, or symbols. They deal strictly with ON and OFF states, which we can represent by 0s and 1s. **Analog** systems measure and respond to continuously varying electrical or physical magnitudes. Analog devices are integrated electronically into systems to continuously monitor and control such quantities as temperature, pressure, velocity, and position and to provide automated control based on the levels of these quantities. Figure 1 shows some examples of digital and analog quantities.

Review Questions

1. List three examples of *analog* quantities.

2. Why do computer systems deal with *digital* quantities instead of *analog* quantities?

2 Digital Representations of Analog Quantities

Most naturally occurring physical quantities in our world are analog in nature. An analog signal is a continuously variable electrical or physical quantity. Think about a mercury-filled tube thermometer; as the temperature rises, the mercury expands in

Figure 1 Analog versus digital: (a) analog waveform; (b) digital waveform; (c) analog watch; (d) digital watch.

analog fashion and makes a smooth, continuous motion relative to a scale measured in degrees. A baseball player swings a bat in an analog motion. The velocity and force with which a musician strikes a piano key are analog in nature. Even the resulting vibration of the piano string is an analog, sinusoidal vibration.

So why do we need to use digital representations in a world that is naturally analog? The answer is that if we want an electronic machine to interpret, communicate, process, and store analog information, it is much easier for the machine to handle it if we first convert the information to a digital format. A digital value is represented by a combination of ON and OFF voltage levels that are written as a string of 1s and 0s.

For example, an analog thermometer that registers 72°F can be represented in a digital circuit as a series of ON and OFF voltage levels. (We'll learn later that the number 72 converted to digital levels is 0100 1000.) The convenient feature of using ON/OFF voltage levels is that the circuitry used to generate, manipulate, and store them is very simple. Instead of dealing with the infinite span and intervals of analog voltage levels, all we need to use is ON or OFF voltages (usually +5 V = ON and 0 V = OFF).

A good example of the use of a digital representation of an analog quantity is the audio recording of music. Compact disks (CDs) and digital versatile disks (DVDs) are commonplace and are proving to be superior means of recording and playing back music. Musical instruments and the human voice produce analog signals, and the human ear naturally responds to analog signals. So, where does the digital format fit in? Although the process requires what appears to be extra work, the recording industries convert analog signals to a digital format and then store the information on a CD or DVD. The CD or DVD player then converts the digital levels back to their corresponding analog signals before playing them back for the human ear.

To accurately represent a complex musical signal as a digital string (a series of 1s and 0s), several samples of an analog signal must be taken, as shown in

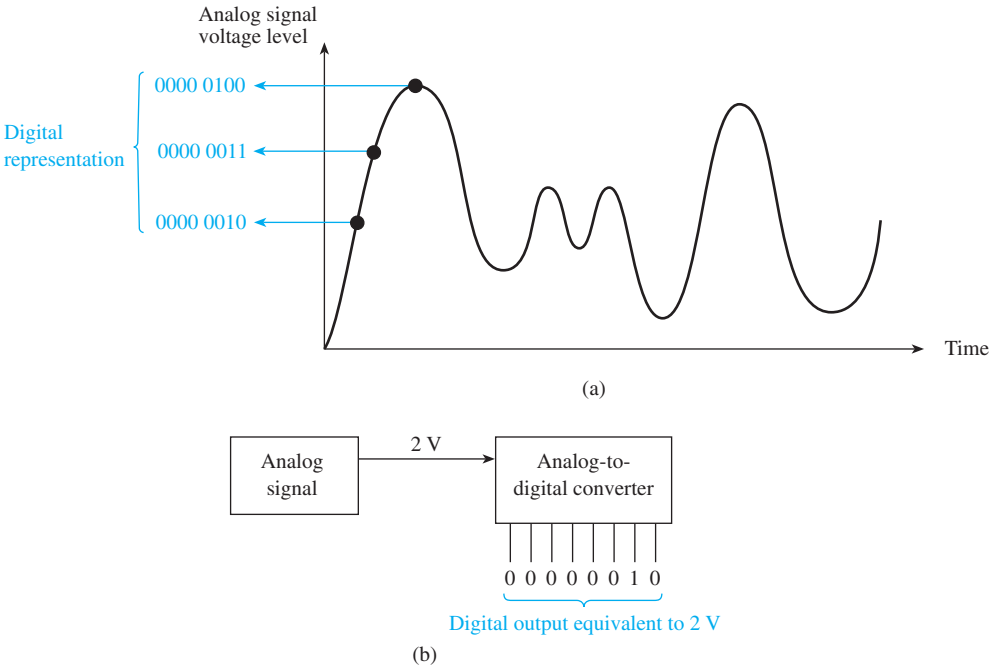


Figure 2 (a) Digital representation of three data points on an analog waveform; (b) converting a 2-V analog voltage into a digital output string.

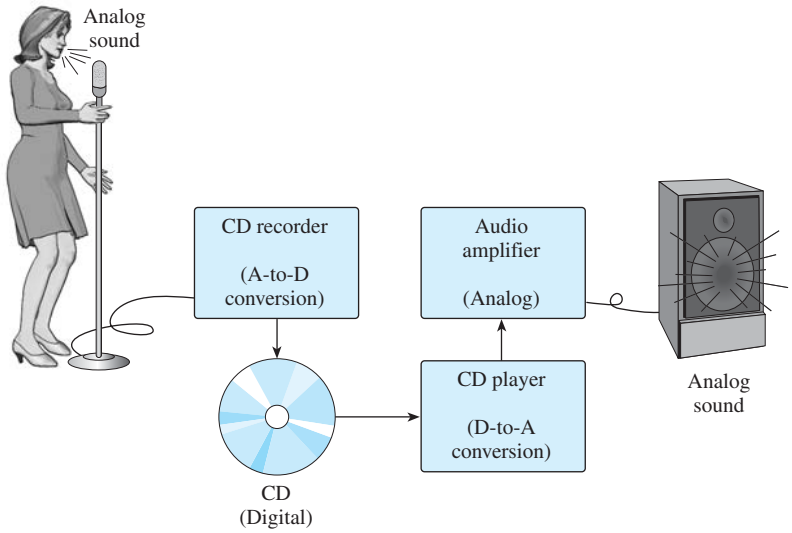


***Figure 3** The process of converting analog sound to digital and then back to analog.

Figure 2(a). The first conversion illustrated is at a point on the rising portion of the analog signal. At that point, the analog voltage is 2 V. Two volts are converted to the digital string 0000 0010, as shown in Figure 2(b). The next conversion is taken as the analog signal in Figure 2(a) is still rising, and the third is taken at its highest level. This process continues throughout the entire piece of music to be recorded. To play back the music, the process is reversed. Digital-to-analog conversions are made to recreate the original analog signal (see Figure 3). If a high-enough number of samples are taken of the original analog signal, an almost-exact reproduction of the original music can be made.

*For additional information on A-to-D and D-to-A be sure to view the podcasts provided on the text website www.pearsonhighered.com/kleitz.

Helpful Hint

One of the more interesting uses of analog-to-digital (A-to-D) and digital-to-analog (D-to-A) conversion is in CD audio systems.

Inside Your PC

A typical 4-minute song requires as many as 300 million ON/OFF digital levels (bits) to be represented accurately. To be transmitted efficiently over the Internet, data compression schemes such as the MP3 standard are employed to reduce the number of bits 10-fold.

Inside Your PC

The CD player uses the optics of a laser beam to look for pits or nonpits on the CD as it spins beneath it. These pits, which are burned into the CD by the CD recorder, represent the 1s and 0s of the digital information the player needs to recreate the original data. A CD contains up to 650 million bytes of digital 1s and 0s (1 byte = 8 bits).

Another optical storage medium is the digital versatile disk (DVD). A DVD is much denser than a CD. It can hold up to 17 billion bytes of data!

It certainly is extra work, but digital recordings have virtually eliminated problems such as electrostatic noise and the magnetic tape hiss associated with earlier methods of audio recording. These problems have been eradicated because, when imperfections are introduced to a digital signal, the slight variation in the digital level does not change an ON level to an OFF level, whereas a slight change in an analog level is easily picked up by the human ear as shown in Figure 4.

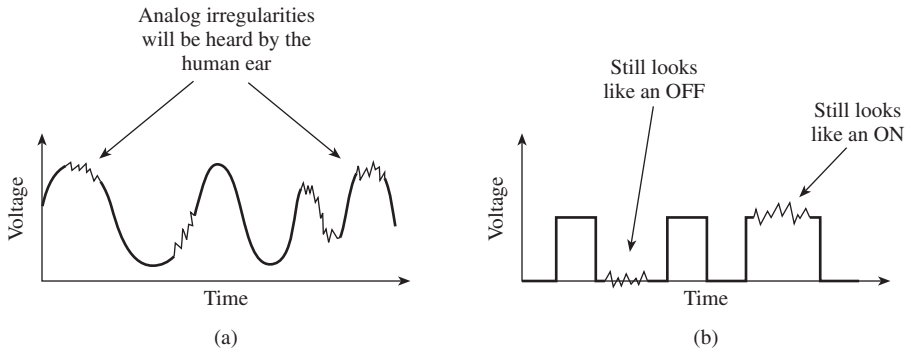


Figure 4 Adding unwanted electrostatic noise to (a) an analog waveform and (b) a digital waveform.

Another application of digital representations of analog quantities is data logging of alternative energy sources. It is very important for energy technicians to keep track of the efficiency of their energy-collection systems. In the case of the solar-collection system shown in Figures 5(a) and (b), system efficiency can be determined by dividing the number of watts produced by the solar photovoltaic (PV) panels by the total solar energy (irradiance) striking the panels. However, since all naturally occurring quantities like solar, wind, temperature, and pressure are analog values, we need to convert them to a digital representation before they can be understood by a computer system.

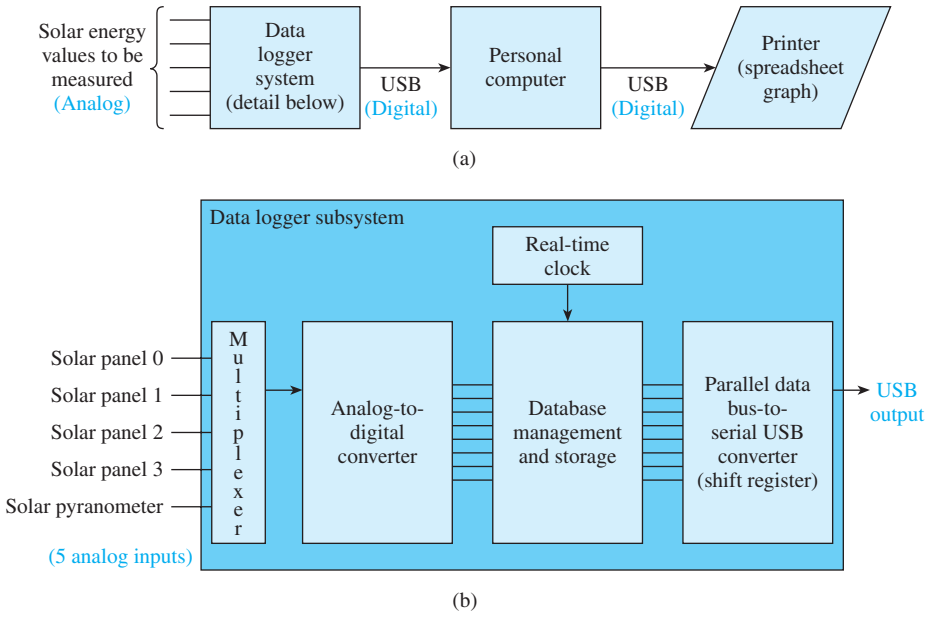


Figure 5 Solar radiation data-logger system: (a) system block diagram; (b) data logger subsystem.

In Figure 5(a) there are five analog solar quantities input to a data-logging system. The data logger digitizes these values and outputs them as a data stream in the USB (Universal Serial Bus) format to a personal computer, which can then be used to analyze the data via a spreadsheet to determine efficiency.

The details of the data-logging system are shown in Figure 5(b). It shows the input to the system as four solar PV panels and one solar pyranometer. The pyranometer is used to measure the solar energy striking the earth at that location in watts-per-meter^2. As the solar PV panels convert sunlight to power (watts), each panel also provides an analog voltage that is proportional to the watts produced. These four analog values are connected to a multiplexer, which alternately routes each of the analog quantities, one at a time, to the analog-to-digital converter (ADC). As each value is received, the ADC outputs its equivalent as an 8-bit digital number (8-, 10-, 12- and higher-bit ADC converters are available). These data need to be time-stamped to help the technician keep track of efficiency at different times of the day and other modifications he or she may have made to the panels during the day. A digital real-time clock circuit provides this time stamp.

Finally, before the data logger can communicate to the PC, the digital data which are now in "parallel" format must be converted to "serial" format to comply with the USB standard used by PCs. This parallel-to-serial conversion is made by a shift register similar to those discussed in a separate chapter. The following sections teach you how to develop and interpret these binary codes that are used in digital systems.

Review Questions

3. Complete the following sentences with the word *analog* or *digital*:

a) Wind speed is an example of a(an) _____ quantity?

b) A music CD contains _____ information?

c) A USB connector transmits _____ data?

d) Hourly outdoor air temperatures exhibit _____ variations?

4. An automobile speedometer display is (digital, analog, or could be either)

5. An analog-to-digital converter outputs an analog voltage. True or false?

6. A music CD player is an example of a(n) (ADC or DAC) process?

7. Electrostatic noise causes more of a problem with which type of signal (analog or digital). Why?

8. Figure 5 implies that the internal circuitry of a PC can only work on (digital, analog) signals?

9. What is the purpose of the multiplexer in Figure 5(b)?

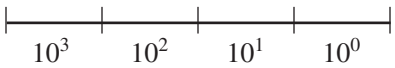
10. What is the purpose of the shift register in Figure 5(b)?

3 Decimal Numbering System (Base 10)

In the **decimal** numbering system, each position contains 10 different possible digits. These digits are 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9. Each position in a multidigit number will have a weighting factor based on a power of 10.

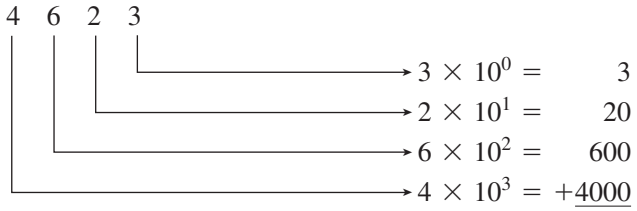
EXAMPLE 1

In a four-digit decimal number, the least significant position (rightmost) has a weighting factor of 10^0; the most significant position (leftmost) has a weighting factor of 10^3:



where $10^3 = 1000$
$10^2 = 100$
$10^1 = 10$
$10^0 = 1$

To evaluate the decimal number 4623, the digit in each position is multiplied by the appropriate weighting factor:



Answer

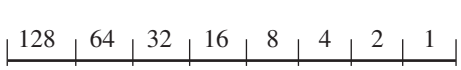
Example 1 illustrates the procedure used to convert from some number system to its decimal (base 10) equivalent. (In the example, we converted a base 10 number to a base 10 answer.) Now let's look at base 2 (binary), base 8 (octal), and base 16 (hexadecimal).

4 Binary Numbering System (Base 2)

Digital electronics use the **binary** numbering system because it uses only the digits 0 and 1, which can be represented simply in a digital system by two distinct voltage levels, such as +5 V = 1 and 0 V = 0.

The weighting factors for binary positions are the powers of 2 shown in Table 1.

TABLE 1 Powers-of-2 Binary Weighting Factors



$2^0 = 1$
$2^1 = 2$
$2^2 = 4$
$2^3 = 8$
$2^4 = 16$
$2^5 = 32$
$2^6 = 64$
$2^7 = 128$

EXAMPLE 2

Convert the binary number 01010110_2 to decimal. (Notice the subscript 2 used to indicate that 01010110 is a base 2 number. A capital letter B can also be used, i.e., 01010110B.)

Solution: Multiply each binary digit by the appropriate weight factor and total the results.

128	64	32	16	8	4	2	1
0	1	0	1	0	1	1	0

$0 \times 2^0 = 0$
$1 \times 2^1 = 2$
$1 \times 2^2 = 4$
$0 \times 2^3 = 0$
$1 \times 2^4 = 16$
$0 \times 2^5 = 0$
$1 \times 2^6 = 64$
$0 \times 2^7 = 0$

86_{10} ***Answer***

Although seldom used in digital systems, binary weighting for values less than 1 is possible (fractional binary numbers). These factors are developed by successively dividing the weighting factor by 2 for each decrease in the power of 2. This is also useful to illustrate why 2^0 is equal to 1, not zero (see Figure 6).

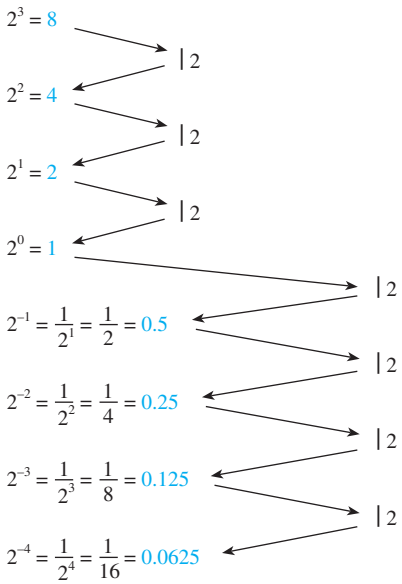
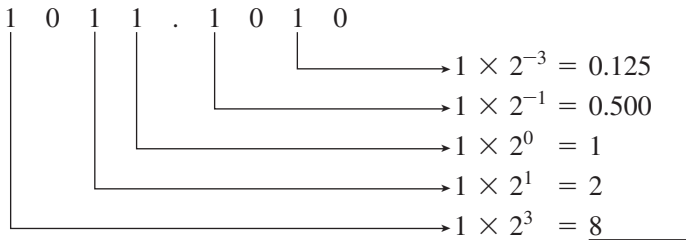


Figure 6 Successive division by 2 to develop fractional binary weighting factors and show that 2^0 is equal to 1.

EXAMPLE 3

Convert the fractional binary number 1011.1010_2 to decimal.

Solution: Multiply each binary digit by the appropriate weighting factor given in Figure 6, and total the results. (We skip the multiplication for the binary digit 0 because it does not contribute to the total.)



11.625_{10} ***Answer***

Review Questions

11. Why is the binary numbering system commonly used in digital electronics?

12. How are the weighting factors determined for each binary position in a base 2 number?

13. Convert $0110\ 1100_2$ to decimal.

14. Convert 1101.0110_2 to decimal.

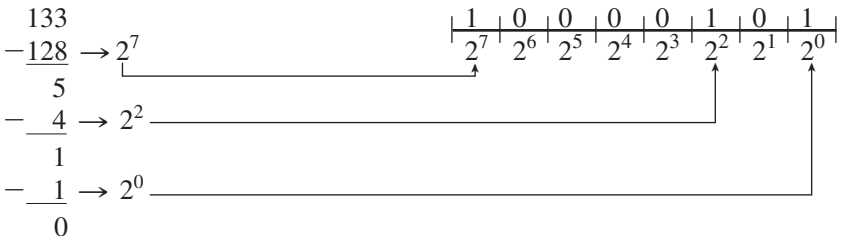
5 Decimal-to-Binary Conversion

The conversion from binary to decimal is usually performed by the digital computer for ease of interpretation by the person reading the number. Conversely, when a person enters a decimal number into a digital computer, that number must be converted to binary before it can be operated on. Let's look at *decimal-to-binary* conversion.

EXAMPLE 4

Convert 133_{10} to binary.

Solution: Referring to Table 1, we can see that the largest power of 2 that will fit into 133 is 2^7 ($2^7 = 128$), but that will still leave the value $5(133 - 128 = 5)$ to be accounted for. Five can be taken care of by 2^2 and 2^0 ($2^2 = 4$, $2^0 = 1$). So the process looks like this:



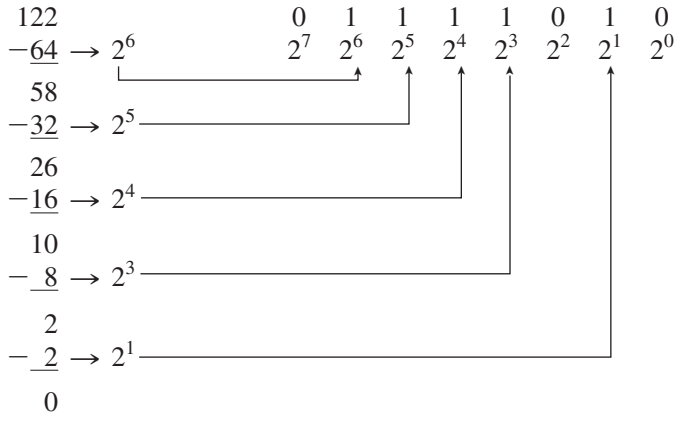
Answer: $1\ 0\ 0\ 0\ 0\ 1\ 0\ 1_2$

Note: The powers of 2 needed to give the number 133 were first determined. Then all other positions were filled with zeros.

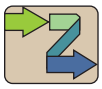
EXAMPLE 5

Convert 122_{10} to binary.

Solution:



Answer: $0\ 1\ 1\ 1\ 1\ 0\ 1\ 0_2$



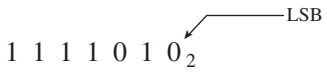
Helpful Hint

This is a good time to realize that a useful way to learn new material like this is to re-solve the examples with the solutions covered up. That way, when you have a problem, you can uncover the solution and see the correct procedure.

Another method of converting decimal to binary is by *successive division.* Successive division involves dividing repeatedly by the number of the base to which you are converting. Continue the process until the answer is 0. For example, to convert 122_{10} to base 2, use the following procedure:

$$\begin{aligned}
122 \div 2 &= 61 \quad \text{with a remainder of } 0 \quad \text{(LSB)}\\
61 \div 2 &= 30 \quad \text{with a remainder of } 1\\
30 \div 2 &= 15 \quad \text{with a remainder of } 0\\
15 \div 2 &= 7 \quad \text{with a remainder of } 1\\
7 \div 2 &= 3 \quad \text{with a remainder of } 1\\
3 \div 2 &= 1 \quad \text{with a remainder of } 1\\
1 \div 2 &= 0 \quad \text{with a remainder of } 1 \quad \text{(MSB)}
\end{aligned}$$

The first remainder, 0, is the **least significant bit (LSB)** of the answer; the last remainder, 1, is the **most significant bit (MSB)** of the answer. Therefore, the answer is as follows:



However, because most computers or digital systems deal with groups of 4, 8, 16, or 32 **bits** (**b**inary dig**it**s), we should keep all our answers in that form. Adding a leading zero to the number $1\ 1\ 1\ 1\ 0\ 1\ 0_2$ will not change its numeric value; therefore, the 8-bit answer is as follows:

$$1\ 1\ 1\ 1\ 0\ 1\ 0_2 = 0\ 1\ 1\ 1\ 1\ 0\ 1\ 0_2$$

Remember not to reverse the LSB and MSB when listing the binary answer.

EXAMPLE 6

Convert 152_{10} to binary using successive division.

Solution:

152 ÷ 2 = 76	remainder 0	(LSB)
76 ÷ 2 = 38	remainder 0	
38 ÷ 2 = 19	remainder 0	
19 ÷ 2 = 9	remainder 1	
9 ÷ 2 = 4	remainder 1	
4 ÷ 2 = 2	remainder 0	
2 ÷ 2 = 1	remainder 0	
1 ÷ 2 = 0	remainder 1	(MSB)

Answer: $1\ 0\ 0\ 1\ 1\ 0\ 0\ 0_2$

Review Questions

15. Convert 43_{10} to binary.

16. Convert 170_{10} to binary.

6 Octal Numbering System (Base 8)

The **octal** numbering system is a method of grouping binary numbers in groups of three. The eight allowable digits are 0, 1, 2, 3, 4, 5, 6, and 7.

The octal numbering system is used by manufacturers of computers that utilize 3-bit codes to indicate instructions or operations to be performed. By using the octal representation instead of binary, the user can simplify the task of entering or reading computer instructions and thus save time.

In Table 2, we see that when the octal number exceeds 7, the least significant octal position resets to zero and the next most significant position increases by 1.

TABLE 2 Octal Numbering System

Decimal	Binary	Octal
0	000	0
1	001	1
2	010	2
3	011	3
4	100	4
5	101	5
6	110	6
7	111	7
8	1000	10
9	1001	11
10	1010	12

7 Octal Conversions

Converting from *binary to octal* is simply a matter of grouping the binary positions in groups of three (starting at the least significant position) and writing down the octal equivalent.

EXAMPLE 7

Convert $0\ 1\ 1\ 1\ 0\ 1_2$ to octal.

Solution:

$$\underbrace{0\ 1\ 1}_{3}\quad\underbrace{1\ 0\ 1}_{5} = 35_8 \quad \textit{Answer}$$

EXAMPLE 8

Convert $1\ 0\ 1\ 1\ 1\ 0\ 0\ 1_2$ to octal.

Solution:

add a leading zero

$$\underbrace{0\ 1\ 0}_{2}\quad\underbrace{1\ 1\ 1}_{7}\quad\underbrace{0\ 0\ 1_2}_{1} = 271_8 \quad \textit{Answer}$$

To convert *octal to binary,* you reverse the process.

EXAMPLE 9

Convert $6\ 2\ 4_8$ to binary.

Solution:

$$\underbrace{1\ 1\ 0}_{6}\quad\underbrace{0\ 1\ 0}_{2}\quad\underbrace{1\ 0\ 0}_{4} = 1\ 1\ 0\ 0\ 1\ 0\ 1\ 0\ 0_2 \quad \textit{Answer}$$

To convert from *octal to decimal*, follow a process similar to that in Section 3 (multiply by weighting factors).

EXAMPLE 10

Convert $3\ 2\ 6_8$ to decimal.

Solution:

$$\begin{aligned} 6 \times 8^0 &= 6 \times 1 = 6 \\ 2 \times 8^1 &= 2 \times 8 = 16 \\ 3 \times 8^2 &= 3 \times 64 = 192 \\ & \qquad\qquad 214_{10} \end{aligned}$$

Answer

Helpful Hint

When converting from octal to decimal, some students find it easier to convert to binary first and then convert binary to decimal.

To convert from *decimal to octal*, the successive-division procedure can be used.

EXAMPLE 11

Convert $4\ 8\ 6_{10}$ to octal.

Solution:

$$\begin{aligned} 486 \div 8 &= 60 \text{ remainder } 6 \\ 60 \div 8 &= 7 \text{ remainder } 4 \\ 7 \div 8 &= 0 \text{ remainder } 7 \end{aligned} \Big\} \ 746_8$$

$486_{10} = 746_8$ ***Answer***

Check:

$7\ 4\ 6$

$6 \times 8^0 = 6$

$4 \times 8^1 = 32$

$7 \times 8^2 = 448$

486 √

Review Questions

17. The only digits allowed in the octal numbering system are 0 to 8. True or false?

18. Convert 111011_2 to octal.

19. Convert 263_8 to binary.

20. Convert 614_8 to decimal.

21. Convert 90_{10} to octal.

8 Hexadecimal Numbering System (Base 16)

The **hexadecimal** numbering system, like the octal system, is a method of grouping bits to simplify entering and reading the instructions or data present in digital computer systems. Hexadecimal uses 4-bit groupings; therefore, instructions or data used in 8-, 16-, or 32-bit computer systems can be represented as a two-, four-, or eight-digit hexadecimal code instead of using a long string of binary digits (see Table 3).

Hexadecimal (hex) uses 16 different digits and is a method of grouping binary numbers in groups of four. Because hex digits must be represented by a single character, letters are chosen to represent values greater than 9. The 16 allowable hex digits are 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, and F.

To signify a hex number, a subscript 16 or the letter H is used (that is, $A7_{16}$ or A7H). Two hex digits are used to represent 8 bits (also known as a *byte*). Four bits (one hex digit) are sometimes called a *nibble.*

TABLE 3 Hexadecimal Numbering System

Decimal	Binary	Hexadecimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F
16	0001 0000	1 0
17	0001 0001	1 1
18	0001 0010	1 2
19	0001 0011	1 3
20	0001 0100	1 4

9 Hexadecimal Conversions

To convert from *binary to hexadecimal*, group the binary number in groups of four (starting in the least significant position) and write down the equivalent hex digit.

EXAMPLE 12

Convert 01101101_2 to hex.

Solution:

$$\underbrace{0\ 1\ 1\ 0}_{6}\ \ \underbrace{1\ 1\ 0\ 1}_{D}{}_2 = 6D_{16}$$ ***Answer***

To convert *hexadecimal to binary*, use the reverse process.

EXAMPLE 13

Convert $A9_{16}$ to binary.

Solution:

$$\overbrace{1\ 0\ 1\ 0}^{A}\ \ \overbrace{1\ 0\ 0\ 1}^{9} = 10101001_2$$ ***Answer***

Helpful Hint

When converting from hexadecimal to decimal, some students find it easier to convert to binary first and then to convert binary to decimal.

To convert *hexadecimal to decimal*, use a process similar to that in Section 3.

EXAMPLE 14

Convert $2\ A\ 6_{16}$ to decimal.

Solution:

$$\begin{array}{cccc} 2 & A & 6 & \\ & & \longrightarrow & 6 \times 16^0 = 6 \times 1 = 6 \\ & \longrightarrow & & A \times 16^1 = 10 \times 16 = 160 \\ \longrightarrow & & & 2 \times 16^2 = 2 \times 256 = \underline{512} \\ & & & 678_{10} \end{array}$$

Answer

EXAMPLE 15

Redo Example 14 by converting first to binary and then to decimal.

Solution:

$$\overbrace{0010}^{2}\ \overbrace{1010}^{A}\ \overbrace{0110}^{6} = 2 + 4 + 32 + 128 + 512 = 678_{10}$$

Answer

To convert from *decimal to hexadecimal*, use successive division. (*Note:* Successive division can always be used when converting from base 10 to any other base numbering system.)

Helpful Hint

At this point, you may be asking if you can use your hex calculator key instead of the hand procedure to perform these conversions. It is important to master these conversion procedures before depending on your calculator so that you understand the concepts involved.

EXAMPLE 16

Convert 151_{10} to hex.

Solution:

$$\begin{array}{rcrcll} 151 & \div & 16 & = 9 & \text{remainder } 7 & \text{(LSD)} \\ 9 & \div & 16 & = 0 & \text{remainder } 9 & \text{(MSD)} \end{array}$$

$$151_{10} = 97_{16}$$

Answer

Check:

$$\begin{array}{cc} 97_{16} & \\ \longrightarrow & 7 \times 16^0 = 7 \\ \longrightarrow & 9 \times 16^1 = \underline{144} \\ & 151 \end{array}$$

√

Team Discussion

Which is the largest number—142_8, 142_{10}, or 142_{16}?

EXAMPLE 17

Convert 498_{10} to hex.

Solution:

$$\begin{array}{rcrcll} 498 & \div & 16 & = 31 & \text{remainder } 2 & \text{(LSD)} \\ 31 & \div & 16 & = 1 & \text{remainder } 15\ (=\text{F}) & \\ 1 & \div & 16 & = 0 & \text{remainder } 1 & \text{(MSD)} \end{array}$$

$$498_{10} = 1\ \text{F}\ 2_{16}$$

Answer

Check:

$$\begin{array}{lrcl} 1\ \mathrm{F}\ 2_{16} & 2 \times 16^0 = & 2 \times & 1 = 2 \\ & \mathrm{F} \times 16^1 = & 15 \times & 61 = 240 \\ & 1 \times 16^2 = & 1 \times & 256 = \underline{256} \\ & & & 498 \ \surd \end{array}$$

Review Questions

22. Why is hexadecimal used instead of the octal numbering system when working with 8- and 16-bit digital computers?

23. The *successive-division* method can be used whenever converting from base 10 to any other base numbering system. True or false?

24. Convert $0110\ 1011_2$ to hex.

25. Convert $\mathrm{E7}_{16}$ to binary.

26. Convert $\mathrm{16C}_{16}$ to decimal.

27. Convert 300_{10} to hex.

10 Binary-Coded-Decimal System

The binary-coded-decimal **(BCD)** system is used to represent each of the 10 decimal digits as a 4-bit binary code. This code is useful for outputting to displays that are always numeric (0 to 9), such as those found in digital clocks or digital voltmeters.

To form a BCD number, simply convert each decimal digit to its 4-bit binary code.

EXAMPLE 18

Convert $4\ 9\ 6_{10}$ to BCD.

Solution:

$$\overbrace{0100}^{4}\ \overbrace{1001}^{9}\ \overbrace{0110}^{6} = 0100\ \ 1001\ \ 0110_{\mathrm{BCD}}$$ ***Answer***

To convert *BCD to decimal,* just reverse the process.

EXAMPLE 19

Convert $0111\ \ 0101\ \ 1000_{\mathrm{BCD}}$ to decimal.

Solution:

$$\overbrace{0111}^{7}\ \overbrace{0101}^{5}\ \overbrace{1000}^{8} = 758_{10}$$ ***Answer***

EXAMPLE 20

Convert $0110\ 0100\ 1011_{BCD}$ to decimal.

Solution:

0110	0100	1011
6	4	*

*This conversion is impossible because 1011 is not a valid binary-coded decimal. It is not in the range 0 to 9.

11 Comparison of Numbering Systems

Table 4 compares numbers written in the five number systems commonly used in digital electronics and computer systems.

TABLE 4 Comparison of Numbering Systems

Decimal	Binary	Octal	Hexadecimal	BCD
0	0000	0	0	0000
1	0001	1	1	0001
2	0010	2	2	0010
3	0011	3	3	0011
4	0100	4	4	0100
5	0101	5	5	0101
6	0110	6	6	0110
7	0111	7	7	0111
8	1000	1 0	8	1000
9	1001	1 1	9	1001
10	1010	1 2	A	0001 0000
11	1011	1 3	B	0001 0001
12	1100	1 4	C	0001 0010
13	1101	1 5	D	0001 0011
14	1110	1 6	E	0001 0100
15	1111	1 7	F	0001 0101
16	0001 0000	2 0	1 0	0001 0110
17	0001 0001	2 1	1 1	0001 0111
18	0001 0010	2 2	1 2	0001 1000
19	0001 0011	2 3	1 3	0001 1001
20	0001 0100	2 4	1 4	0010 0000

12 The ASCII Code

To get information into and out of a computer, we need more than just numeric representations; we also have to take care of all the letters and symbols used in day-to-day processing. Information such as names, addresses, and item descriptions must be input and output in a readable format. But remember that a digital system can deal only with 1s and 0s. Therefore, we need a special code to represent all **alphanumeric** data (letters, symbols, and numbers).

Most industry has settled on an input/output (I/O) code called the American Standard Code for Information Interchange (ASCII). The **ASCII code** uses 7 bits to represent all the alphanumeric data used in computer I/O. Seven bits will yield 128 different code combinations, as listed in Table 5.

TABLE 5 American Standard Code for Information Interchange

LSB \ MSB	000	001	010	011	100	101	110	111
0000	NUL	DLE	SP	0	@	P	′	p
0001	SOH	DC_1	!	1	A	Q	a	q
0010	STX	DC_2	”	2	B	R	b	r
0011	ETX	DC_3	#	3	C	S	c	s
0100	EOT	DC_4	$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	′	7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[	k	{
1100	FF	FS	,	<	L	\	l	|
1101	CR	GS	-	=	M	]	m	}
1110	SO	RS	.	>	N	↑	n	~
1111	SI	US	/	?	O	—	o	DEL

Definitions of control abbreviations:

ACK	Acknowledge	FS	Form separator
BEL	Bell	GS	Group separator
BS	Backspace	HT	Horizontal tab
CAN	Cancel	LF	Line feed
CR	Carriage return	NAK	Negative acknowledge
DC_1–DC_4	Direct control	NUL	Null
DEL	Delete idle	RS	Record separator
DLE	Data link escape	SI	Shift in
EM	End of medium	SO	Shift out
ENQ	Enquiry	SOH	Start of heading
EOT	End of transmission	SP	Space
ESC	Escape	STX	Start text
ETB	End of transmission block	SUB	Substitute
ETX	End text	SYN	Synchronous idle
FF	Form feed	US	Unit separator
		VT	Vertical tab

Team Discussion

Have you ever tried displaying non-ASCII data to your PC screen using a disk utility program? If you were to read a file created by the IRS for your tax return, which fields would be ASCII?

Each time a key is depressed on an ASCII keyboard, that key is converted into its ASCII code and processed by the computer. Then, before outputting the computer contents to a display terminal or printer, all information is converted from ASCII into standard English.

To use the table, place the 4-bit group in the least significant positions and the 3-bit group in the most significant positions.

EXAMPLE 21

100 0111 is the code for G.

3-bit group 4-bit group

EXAMPLE 22

Using Table 5, determine the ASCII code for the lowercase letter *p*.

Solution: 1110000 (*Note:* Often, a leading zero is added to form an 8-bit result, making $p = 0111\ 0000$.)

Review Questions

28. How does BCD differ from the base 2 binary numbering system?

29. Why is ASCII code required by digital computer systems?

30. Convert 947_{10} to BCD.

31. Convert $1000\ \ 0110\ \ 0111_{BCD}$ to decimal.

32. Determine the ASCII code for the letter E.

13 Applications of the Numbering Systems

Because digital systems work mainly with 1s and 0s, we have spent considerable time working with the various number systems. Which system is used depends on how the data were developed and how they are to be used. In this section, we work with several applications that depend on the translation and interpretation of these digital representations.

APPLICATION 1

A geothermal electricity generation facility uses a computer to monitor the temperature and pressure of four liquid storage tanks, as shown in Figure 7(a). Whenever a temperature or a pressure exceeds the danger limit, an internal tank sensor applies a 1 to its corresponding output to the computer. If all conditions are OK, then all outputs are 0.

Helpful Hint

This and the following five applications illustrate the answer to the common student question, "Why are we learning this stuff?"

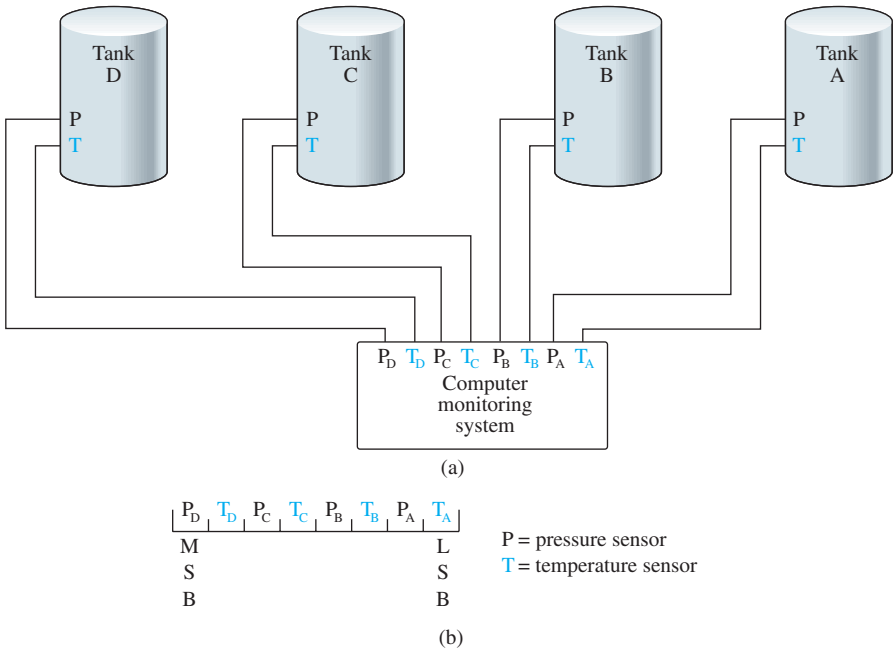


Figure 7 (a) Circuit connections for temperature and pressure monitors at a geothermal electricity generation facility; (b) layout of binary data read by the computer monitoring system.

(a) If the computer reads the binary string 0010 1000, what problems exist?

Solution: Entering that binary string into the chart of Figure 7(b) shows us that the pressure in tanks C and B is dangerously high.

(b) What problems exist if the computer is reading 55H (55 hex)?

Solution: 55H = 0101 0101, meaning that all temperatures are too high

(c) What hexadecimal number is read by the computer if the temperature and pressure in both tanks D and B are high?

Solution: CCH (1100 1100 = CCH)

(d) Tanks A and B are taken out of use, and their sensor outputs are connected to 1s. A computer programmer must write a program to ignore these new circuit conditions. The computer program must check that the value read is always less than what decimal equivalent when no problem exists?

Solution: $<31_{10}$, because, with the 4 low-order bits HIGH, if TC goes HIGH, then the binary string will be 0001 1111, which is equal to 31_{10}.

(e) In another area of the plant, only three tanks (A, B, and C) have to be monitored. What octal number is read if tank B has a high temperature and pressure?

Solution: 14_8 ($001\ 100_2 = 14_8$)

APPLICATION 2

A particular brand of CD player has the capability of converting 12-bit signals from a CD into their equivalent analog values.

(a) What are the largest and smallest hex values that can be used in this CD system?

Solution: Largest: FFF_{16}; smallest: 000_{16}

(b) How many different analog values can be represented by this system?

Solution: FFF_{16} is equivalent to 4095 in decimal. Including 0, this is a total of 4096 unique representations.

APPLICATION 3

Typically, digital thermometers use BCD to drive their digit displays.

(a) How many BCD bits are required to drive a 3-digit thermometer display?

Solution: 12; 4 bits for each digit

(b) What 12 bits are sent to the display for a temperature of 147°F?

Solution: 0001 0100 0111

APPLICATION 4

Most PC-compatible computer systems use a 20-bit address code to identify each of over 1 million memory locations.

(a) How many hex characters are required to identify the address of each memory location?

Solution: Five (Each hex digit represents 4 bits.)

Common Misconception

You may have a hard time visualizing why we add or subtract 1 to determine memory locations. Answer this question: How many problems must you solve if your teacher assigns problems 5 through 10? (You would subtract 5 from 10 and then add 1.) How about if you solve 8 problems starting with 10: Would the last problem be 18 or 17?

(b) What is the 5-digit hex address of the 200th memory location?

Solution: 000C7H (200_{10} = C8H; but the first memory location is 00000H, so we have to subtract 1).

(c) If 50 memory locations are used for data storage starting at location 00C8H, what is the location of the last data item?

Solution: 000F9H (000C8H = 200_{10}, 200 + 50 = 250_{10}, 250 − 1 = 249_{10}, 249_{10} = F9H [We had to subtract 1 because location C8H (200_{10}) received the first data item, so we needed only 49 more memory spaces.])

Inside Your PC

The address settings of your PC I/O devices are given as hexadecimal numbers. They can be determined on a Windows-based machine by pressing the sequence: My Computer > Control Panel > System > Device Manager > Properties > I/O. Determine from the list on your screen what the address settings are for your keyboard, printer, and floppy disk.

APPLICATION 5

If the part number 651-M is stored in ASCII in a computer memory, list the binary contents of its memory locations.

Solution:

6 = 011 0110
5 = 011 0101
1 = 011 0001
− = 010 1101
M = 100 1101

Because most computer memory locations are formed by groups of 8 bits, let's add a zero to the leftmost position to fill each 8-bit memory location. (The leftmost position is sometimes filled by a parity bit.)

Therefore, the serial number, if strung out in five memory locations, would look like the following:

0011 0110 0011 0101 0011 0001 0010 1101 0100 1101

If you look at these memory locations in hexadecimal, they will read as follows:

36 35 31 2D 4D

APPLICATION 6

To look for an error in a BASIC program, a computer programmer uses a debugging utility to display the ASCII codes of a particular part of her program. The codes are displayed in hex as 474F5430203930. Assume that the leftmost bit of each ASCII string is padded with a 0.

(a) Translate the program segment that is displayed.

Solution: GOT0 90.

(b) If you know anything about programming in BASIC, try to determine what the error is.

Solution: Apparently a number zero was typed in the GOTO statement instead of the letter *O*. Change it, and the error should go away.

Summary

In this chapter, we have learned the following:

1. Numeric quantities occur naturally in analog form but must be converted to digital form to be used by computers or digital circuitry.

2. The binary numbering system is used in digital systems because the 1s and 0s are easily represented by ON or OFF transistors, which output 0 V for 0 and 5 V for 1.

3. Any number system can be converted to decimal by multiplying each digit by its weighting factor.

4. The weighting factor of the least significant digit in any numbering system is always 1.

5. Binary numbers can be converted to octal by forming groups of 3 bits and to hexadecimal by forming groups of 4 bits, beginning with the LSB. Each group is then converted to an octal or hex digit.

6. The successive-division procedure can be used to convert from decimal to binary, octal, or hexadecimal.

7. The binary-coded-decimal system uses groups of 4 bits to drive decimal displays such as those in a calculator.

8. ASCII is used by computers to represent all letters, numbers, and symbols in digital form.

Glossary

Alphanumeric: Characters that contain alphabet letters as well as numbers and symbols.

Analog: A system that deals with continuously varying physical quantities such as voltage, temperature, pressure, or velocity. Most quantities in nature occur in analog, yielding an infinite number of different levels.

ASCII Code: American Standard Code for Information Interchange. ASCII is a 7-bit code used in digital systems to represent all letters, symbols, and numbers to be input or output to the outside world.

BCD: Binary-coded decimal. A 4-bit code used to represent the 10 decimal digits 0 to 9.

Binary: The base 2 numbering system. Binary numbers are made up of 1s and 0s, each position being equal to a different power of 2 (2^3, 2^2, 2^1, 2^0, and so on).

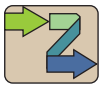
Bit: A single binary digit. The binary number 1101 is a 4-bit number.

Decimal: The base 10 numbering system. The 10 decimal digits are 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9. Each decimal position is a different power of 10 (10^3, 10^2, 10^1, 10^0, and so on).

Digital: A system that deals with discrete digits or quantities. Digital electronics deals exclusively with 1s and 0s or ONs and OFFs. Digital codes (such as ASCII) are then used to convert the 1s and 0s to a meaningful number, letter, or symbol for some output display.

Hexadecimal: The base 16 numbering system. The 16 hexadecimal digits are 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, and F. Each hexadecimal position represents a different power of 16 (16^3, 16^2, 16^1, 16^0, and so on).

Least Significant Bit (LSB): The bit having the least significance in a binary string. The LSB will be in the position of the lowest power of 2 within the binary number.



Helpful Hint

Skimming through the glossary terms is a good way to review the chapter. You should also feel that you have a good understanding of all the topics listed in the objectives at the beginning of the chapter.

Most Significant Bit (MSB): The bit having the most significance in a binary string. The MSB will be in the position of the highest power of 2 within the binary number.

Octal: The base 8 numbering system. The eight octal numbers are 0, 1, 2, 3, 4, 5, 6, and 7. Each octal position represents a different power of 8 (8^3, 8^2, 8^1, 8^0, and so on).

Problems

Section 4

1. Convert the following binary numbers to decimal.

(a) 0110 **(b)** 1011 **(c)** 1001 **(d)** 0111
(e) 1100 **(f)** 0100 1011 **(g)** 0011 0111
(h) 1011 0101 **(i)** 1010 0111 **(j)** 0111 0110

Section 5

2. Convert the following decimal numbers to 8-bit binary.

(a) 186_{10} **(b)** 214_{10} **(c)** 27_{10} **(d)** 251_{10} **(e)** 146_{10}

Sections 6 and 7

3. Convert the following binary numbers to octal.

(a) 011001 **(b)** 11101 **(c)** 1011100
(d) 01011001 **(e)** 1101101

4. Convert the following octal numbers to binary.

(a) 46_8 **(b)** 74_8 **(c)** 61_8 **(d)** 32_8 **(e)** 57_8

5. Convert the following octal numbers to decimal.

(a) 27_8 **(b)** 37_8 **(c)** 14_8 **(d)** 72_8 **(e)** 51_8

6. Convert the following decimal numbers to octal.

(a) 126_{10} **(b)** 49_{10} **(c)** 87_{10} **(d)** 94_{10} **(e)** 108_{10}

Sections 8 and 9

7. Convert the following binary numbers to hexadecimal.

(a) 1011 1001 **(b)** 1101 1100 **(c)** 0111 0100
(d) 1111 1011 **(e)** 1100 0110

8. Convert the following hexadecimal numbers to binary.

(a) $C5_{16}$ **(b)** FA_{16} **(c)** $D6_{16}$ **(d)** $A94_{16}$ **(e)** 62_{16}

9. Convert the following hexadecimal numbers to decimal.

(a) 86_{16} **(b)** $F4_{16}$ **(c)** 92_{16} **(d)** AB_{16} **(e)** $3C5_{16}$

10. Convert the following decimal numbers to hexadecimal.

(a) 127_{10} **(b)** 68_{10} **(c)** 107_{10} **(d)** 61_{10} **(e)** 29_{10}

Section 10

11. Convert the following BCD numbers to decimal.

(a) $1001\ 1000_{BCD}$ **(b)** $0110\ 1001_{BCD}$ **(c)** $0111\ 0100_{BCD}$
(d) $0011\ 0110_{BCD}$ **(e)** $1000\ 0001_{BCD}$

12. Convert the following decimal numbers to BCD.

(a) 87_{10} **(b)** 142_{10} **(c)** 94_{10} **(d)** 61_{10} **(e)** 44_{10}

13. Fill in all of the empty cells in Table P13 by performing the indicated conversion as shown in the row labeled "sample."

14. Fill in all of the empty cells in Table P14 by performing the indicated conversion as shown in the row labeled "sample."

TABLE P13

	Decimal	Binary	Octal	BCD	Hexadecimal
Sample	16	0001 0000	020	0001 0110	10
(a)	35				
(b)		0010 1001			
(c)			053		
(d)				0111 1000	
(e)					3A

TABLE P14

	Decimal	Binary	Octal	BCD	Hexadecimal
Sample	59	0011 1011	073	0101 1001	3B
(a)					44
(b)				1001 1000	
(c)			127		
(d)		0011 0100			
(e)	45				

Section 12

15. Use Table 5 to convert the following letters, symbols, and numbers to ASCII.

(a) % **(b)** $14 **(c)** N-6 **(d)** CPU **(e)** Pg

16. Insert a zero in the MSB of your answers to Problem 13, and list your answers in hexadecimal.

Section 13

C* **17.** The computer monitoring system at the geothermal facility shown in Figure 7 is receiving the following warning codes. Determine the problems that exist for each code (H stands for hex).

(a) $0010\ 0001_2$ **(b)** $C0_{16}$ **(c)** 88H **(d)** 024_8 **(e)** 48_{10}

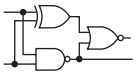
C **18.** What is the BCD representation that is sent to a three-digit display on a voltmeter that is measuring 120 V?

C **19.** A computer programmer observes the following hex string when looking at a particular section of computer memory: 736B753433.

(a) Assume that the memory contents are ASCII codes with leading zeros and translate this string into its alphanumeric equivalent.

(b) The programmer realizes that the program recognizes only capital (uppercase) letters. Convert all letters in the alphanumeric equivalent to capital letters, and determine the new hex string.

*The letter C signifies problems that are more Challenging and thought provoking.



Schematic Interpretation Problems

(*Note:* Appendix: Schematic Diagrams for Chapter-End Problems contains four schematic diagrams of actual digital systems. At the end of this chapter, you will have the opportunity to work with these diagrams to gain experience with real-world circuitry and observe the application of digital logic that was presented in the chapter.)

S* **20.** Locate the HC11D0 master board schematic in Appendix: Schematic Diagrams for Chapter-End Problems. Determine the component name and grid coordinates of the following components. (*Example:* Q3 is a 2N2907 located at A3.)

(a) U1 **(b)** U16 **(c)** Q1 **(d)** P2

S **21.** Find the date and revision number for the HC11D0 master board schematic.

S **22.** Find the quantity of the following devices that are used on the watchdog timer schematic.

(a) 74HC85 **(b)** 74HC08 **(c)** 74HC74 **(d)** 74HC32

MultiSIM® Exercises

MultiSIM is a software simulation tool that is used to reinforce the theory presented in this chapter. It provides an accurate simulation of digital and analog circuit operation along with a simulation of instruments used by a technician to measure IC, component, and circuit characteristics. With this software, you have the ability to build and test most of the circuits presented in this text. The data files for all MultiSIM examples and problems in this chapter are provided on the text Web site.

The problems at the end of this chapter are based on the circuits and theory presented in the section corresponding to the file name. Before attempting any MultiSIM problems, you must thoroughly understand the material presented in that chapter section. The problem definition for each MultiSIM circuit is fully explained in the Description Window that appears in each MultiSIM file.

The problems are basically of three types: (1) *circuit interaction problems* require the student to change input values and take measurements at the outputs to verify circuit operation; (2) *design problems* require the student to design, or modify, a circuit to perform a particular task; and (3) *troubleshooting problems* require the student to find and fix the fault that exists in the circuit that is given.

You will notice that the MultiSIM problems use a slightly different notation to represent certain variables. For example, $\overline{A}$ is represented by A′, C_p is represented by Cp, and 2^0 is represented by 2^0.

E1. (*Note:* You need to understand binary to hexadecimal conversions [Section 8] before attempting this exercise.) Load the circuit file for **Section 08.** This circuit is used to demonstrate the conversion between the binary and hexadecimal numbering systems similar to Examples 12 and 13. The Word Generator is used to drive eight binary lights and two hexadecimal displays. Read the instructions for the circuit in the *Description* window at the bottom of the screen.

(a) What 8-bit binary number will you see on the lights if you press *Step* five times? (An ON light is a 1.) Try it.

(b) How many times must you press *Step* to get the binary number 0000 1011? Try it.

*The letter S designates Schematic interpretation problem.

(c) What hexadecimal number will you see if you press *Step* 14 times? Try it.

(d) How many times must you press *Step* to see the hexadecimal number 1b? Try it.

E2. (*Note:* You need to understand the operation of the geothermal facility monitoring system presented in Figure 7 before attempting this exercise.) Load the circuit file for **Section 13.** Turn the power switch ON. The hex display should read 00H, which indicates that there are no high temperature or pressure levels.

(a) Read the instructions for the circuit in the *Description* window at the bottom of the screen. What would you expect the hex display to read if there is a high temperature in Tank D? To check your answer, raise the temperature in Tank D by pressing the indicated key several times. Return the temperature to a low level by holding the *Ctrl* key as you press 2 repeatedly.

(b) What would you expect the display to read if all temperatures are high? Check your answer, then return the levels to a low state.

(c) What levels are too high if the hex display reads 0CH? Check your answer by raising the levels on the appropriate tank(s). Return all levels to a low state.

(d) What levels are too high if the hex display reads AAH? Check your answer by raising the levels on the appropriate tanks(s). Return all levels to a low state.

Answers to Review Questions

1. Temperature, pressure, velocity, weight, sound

2. Because digital quantities are easier for a computer system to store and interpret

3. **(a)** Analog **(b)** Digital **(c)** Digital **(d)** Analog

4. Could be either

5. False

6. DAC

7. Analog, because small irregularities in the waveform will be heard, but a digital signal with noise still looks like a HIGH or LOW (1 or 0) voltage level.

8. Digital

9. To route just one input at a time to the ADC

10. To convert the parallel data into serial before outputting to the USB connector

11. Because it uses only two digits, 0 and 1, which can be represented by using two distinct voltage levels

12. By powers of 2

13. 108_{10}

14. 13.375_{10}

15. $0010\ 1011_2$

16. $1010\ 1010_2$

17. False

18. 73_8

19. $010\ 110\ 011_2$ or $1011\ 0011_2$

20. 396_{10}

21. 132_8

22. Because hexadecimal uses 4-bit groupings

23. True

24. $6B_{16}$

25. $1110\ 0111_2$

26. 364_{10}

27. $12C_{16}$

28. BCD is used only to represent decimal digits 0 to 9 in 4-bit groupings.

29. To get alphanumeric data into and out of a computer

30. $1001\ 0100\ 0111_{BCD}$

31. 867_{10}

32. $0100\ 0101_{ASCII}$

Answers to Odd-Numbered Problems

1. **(a)** 6_{10} **(b)** 11_{10} **(c)** 9_{10} **(d)** 7_{10}
(e) 12_{10} **(f)** 75_{10} **(g)** 55_{10} **(h)** 181_{10}
(i) 167_{10} **(j)** 118_{10}

3. **(a)** 31_8 **(b)** 35_8 **(c)** 134_8 **(d)** 131_8
(e) 155_8

5. **(a)** 23_{10} **(b)** 31_{10} **(c)** 12_{10} **(d)** 58_{10}
(e) 41_{10}

7. **(a)** $B9_{16}$ **(b)** DC_{16} **(c)** 74_{16} **(d)** FB_{16}
(e) $C6_{16}$

9. **(a)** 134_{10} **(b)** 244_{10} **(c)** 146_{10}
(d) 171_{10} **(e)** 965_{10}

11. **(a)** 98_{10} **(b)** 69_{10} **(c)** 74_{10} **(d)** 36_{10}
(e) 81_{10}

13.

	Decimal	Binary	Octal	BCD	Hexadecimal
(a)	35	0010 0011	043	0011 0101	23
(b)	41	0010 1001	051	0100 0001	29
(c)	43	0010 1011	053	0100 0011	2B
(d)	78	0100 1110	116	0111 1000	4E
(e)	58	0011 1010	072	0101 1000	3A

15. **(a)** 010 0101
(b) 0100100 0110001 0110100
(c) 1001110 0101101 0110110
(d) 1000011 1010000 1010101
(e) 1010000 1100111

17. **(a)** Tank A, temperature high; tank C, pressure high
(b) Tank D, temperature and pressure high
(c) Tanks B and D, pressure high
(d) Tanks B and C, temperature high
(e) Tank C, temperature and pressure high

19. **(a)** sku43 **(b)** $534B553433_{16}$

21. 16-MAR 1995 Revision A

E1. **(a)** 0000 0101 **(b)** Eleven **(c)** 0E **(d)** 2

Digital Electronic Signals and Switches

OUTLINE

1 Digital Signals
2 Clock Waveform Timing
3 Serial Representation
4 Parallel Representation
5 Switches in Electronic Circuits
6 A Relay as a Switch
7 A Diode as a Switch
8 A Transistor as a Switch
9 The TTL Integrated Circuit
10 MultiSIM® Simulation of Switching Circuits
11 The CMOS Integrated Circuit
12 Surface-Mount Devices

OBJECTIVES

Upon completion of this chapter, you should be able to do the following:

- Describe the parameters associated with digital voltage-versus-time waveforms.
- Convert between frequency and period for a periodic clock waveform.
- Sketch the timing waveform for any binary string in either the serial or parallel representation.
- Discuss the application of manual switches and electromechanical relays in electric circuits.
- Explain the basic characteristics of diodes and transistors when they are forward biased and reverse biased.
- Calculate the output voltage in an electric circuit containing diodes or transistors operating as digital switches.
- Perform input/output timing analysis in electric circuits containing electro-mechanical relays or transistors.
- Explain the operation of a common-emitter transistor circuit used as a digital inverter switch.

The companion website for this text is www.pearsonhighered.com/kleitz

INTRODUCTION

Digital electronics deals with 1s and 0s. These logic states will typically be represented by a high and a low voltage level (usually 1 = 5 V and 0 = 0 V).

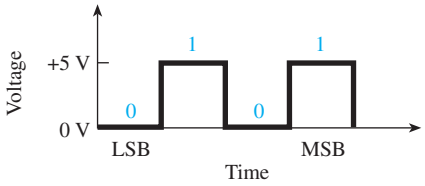
In this chapter, we see how these logic states can be represented by means of a timing diagram and how electronic switches are used to generate meaningful digital signals.

1 Digital Signals

A digital signal is made up of a series of 1s and 0s that represent numbers, letters, symbols, or control signals. Figure 1 shows the **timing diagram** of a typical digital signal. Timing diagrams are used to show the HIGH and LOW (1 and 0) levels of a digital signal as it changes relative to time. In other words, it is a plot of *voltage versus time.* The *y* axis of the plot displays the voltage level and the *x axis*, the time. Digital systems respond to the digital state (0 or 1), not the actual voltage levels. For example, if the voltage levels in Figure 1(a) were not exactly 0 V and +5 V, the digital circuitry would still interpret it as the 0 state and 1 state and respond identically.

Figure 1(a) is a timing diagram showing the bit configuration 1 0 1 0 as it would appear on an **oscilloscope**. Notice in the figure that the LSB comes first in time. In this case, the LSB is transmitted first. The MSB could have been transmitted first as long as the system on the receiving end knows which method is used.

Figure 1(b) is a photograph of an oscilloscope, which is a very important test instrument for making accurate voltage versus time measurements.



(a)

(b)

Figure 1 (a) Typical digital signal; (b) an oscilloscope displaying the digital waveform from a clock generator instrument.

2 Clock Waveform Timing

Most digital signals require precise timing. Special clock and timing circuits are used to produce clock waveforms to trigger the digital signals at precise intervals.

Figure 2 shows a typical *periodic clock waveform* as it would appear on an oscilloscope displaying voltage versus time. The term *periodic* means that the waveform is repetitive, at a specific time interval, with each successive pulse identical to the previous one.

Figure 2 shows eight clock pulses, which we label 0, 1, 2, 3, 4, 5, 6, and 7. The **period** of the clock waveform is defined as the length of time from the falling edge of one pulse to the falling edge of the next pulse (or rising edge to rising edge) and is abbreviated t_p in Figure 2. The **frequency** of the clock waveform is defined as the reciprocal of the clock period. Written as a formula,

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

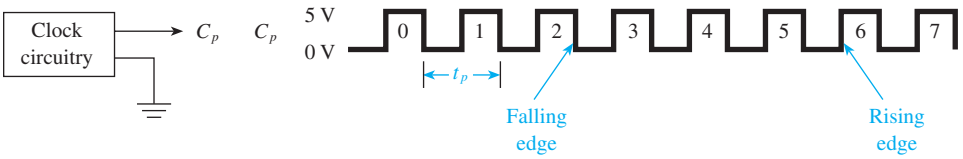


Figure 2 Periodic clock waveform as seen on an oscilloscope displaying voltage versus time.

The basic unit for frequency is *hertz* (Hz), and the basic unit for period is *seconds* (s). Frequency is often referred to as cycles per second (cps) or pulses per second (pps).

Team Discussion

An interesting exercise is to sketch the waveform from a 10-cps clock that is allowed to run for 1s. How long did it take to complete one cycle? How did you find that time? Next, repeat for a 1-MHz clock.

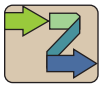
EXAMPLE 1

What is the frequency of a clock waveform whose period is 2 microseconds (μs)?

Solution:

$$f = \frac{1}{t_p} = \frac{1}{2\ \mu\text{s}} = 0.5 \text{ megahertz} \quad (0.5 \text{ MHz or } 500 \text{ kHz})$$

Hint: To review engineering notation, see Table 1.



Helpful Hint

Frequency and time calculations can often be made without a calculator if you realize some of the common reciprocal relationships (e.g., 1/milli = kilo, 1/micro = mega). When using a calculator, if the result is not a power of 3, 6, 9, or 12, then the answer must be converted to one of these common engineering prefixes using algebra or, if available, the *ENG* key on your calculator.

TABLE 1 Common Engineering Prefixes

Prefix	Abbreviation	Power of 10
Tera	T	10^{12}
Giga	G	10^{9}
Mega	M	10^{6}
Kilo	k	10^{3}
Milli	m	10^{-3}
Micro	μ	10^{-6}
Nano	n	10^{-9}
Pico	p	10^{-12}

EXAMPLE 2

A PC manufacturer specifies a microprocessor speed of 4 GHz (Gigahertz). What is the period of the microprocessor's waveform?

Solution:

$$t_p = \frac{1}{f} = \frac{1}{4\ \text{GHz}} = 250\ \text{pS}$$

Digital communications concerns itself with the transmission of bits (1s and 0s). The rate, or frequency, at which they are transmitted is given in bits-per-second (bps). Common transmission rates for a PC connected to the Internet via a telephone line are 28.8 kilobits-per-second (28.8 kbps) and 56 kbps.

EXAMPLE 3

Sketch and label the x and y axis representing a 56 kbps (kilobits per second) clock waveform transmitted between a PC and a peripheral device. (Assume that the voltage levels were measured on an oscilloscope at LOW = 0.2 V and HIGH = 4.5 V.)

Solution:

$$t_p = \frac{1}{f} = \frac{1}{56 \text{ kbps}} = 17.9\ \mu\text{s}$$

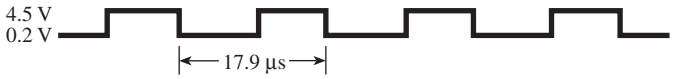


Figure 3 Solution to Example 3.

Team Discussion

For those students who have a PC: Do you know (or could you find out) at what frequency and period your internal microprocessor operates?

EXAMPLE 4

Determine the frequency of the waveform in Figure 4.

Solution:

$$f = \frac{1}{t_p} = \frac{1}{34.7\ \mu\text{s}} = 28.8 \text{ kHz (or 28.8 kbps)}$$

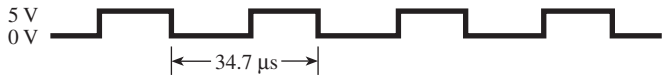


Figure 4 Waveform for Example 4.

Common Misconception

The period is labeled from rising edge to rising edge (or falling edge to falling edge) and is not just the positive pulse.

Review Questions

1. What are the labels on the x axis and y axis of a digital signal mea-sured on an oscilloscope?

2. What is the relationship between clock frequency and clock period?

3. What is the time period from the rising edge of one pulse to the rising edge of the next pulse on a waveform whose frequency is 8 MHz?

4. What is the frequency of a periodic waveform having a period of 50 ns?

5. Repeat Example 1 for a period of 200 ns.

6. Repeat Example 2 for a frequency of 2.6 GHz.

7. Repeat Example 3 for a waveform frequency of 2.8 Mbps and voltage of 0.4 and 4.8 V.

8. Repeat Example 4 for a period of 17.1 μs.

3 Serial Representation

Binary information to be transmitted from one location to another will be in either **serial** or **parallel** format. The serial format uses a single electrical conductor (and a common ground) for the data to travel on. The serial format is inexpensive because it only uses a single conductor and one set of input/output circuitry, but it is slow because it can only transmit 1 bit for each clock period. Communication over telephone lines (like the Internet) and computer-to-computer communication (like office networks) use serial communication (see Figure 5). The ports labels COM on a PC are most often used for the serial communication connection to telephone lines. A plug-in card is used in a PC to provide network serial communication (e.g., Ethernet).

Serial communication can be sped up by using extremely high-speed clock signals. Modern Internet connections and office networks communicate at speeds exceeding 1 million bps. Several standards have been developed for high-speed serial communications, the most common of which are V.90, ISDN, T1, T2, T3, Universal Serial Bus (USB), Ethernet, 10baseT, 100baseT, 1000baseT, cable, and DSL.

Helpful Hint

Although this is too complicated to detail here, you should realize that often there are other *handshaking* signals involved in serial communication (i.e., ready to receive, ready to transmit, start bits, stop bits, parity, and so on).

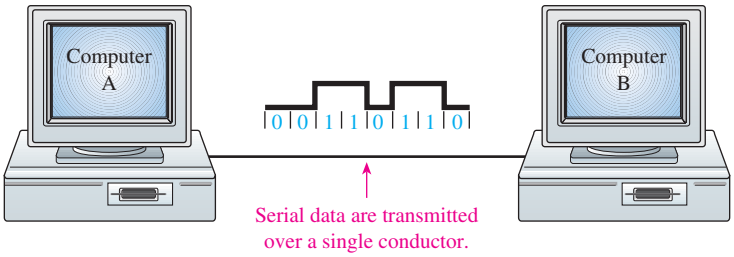


Figure 5 Serial communication between computers.

Let's use Figure 6 to illustrate the serial representation of the binary number 0 1 1 0 1 1 0 0. The serial representation (S_o) is shown with respect to some clock waveform (C_p), and its LSB is drawn first. Each bit from the original binary number occupies a separate clock period, with the change from one bit to the next occurring at each *falling* edge of C_p (C_p is drawn just as a reference).

Inside Your PC

Standard transmission speed for a PC's serial port (labeled COM on Windows-based machines) is 115 kbps. Much higher serial speeds are achieved using the newer (USB) standard. The original version 1.1 standard called for 12 Mbps transmission speeds. Version 2.0 specifies 480 Mbps and version 3.0 can transmit at speeds up to 5 Gbps!

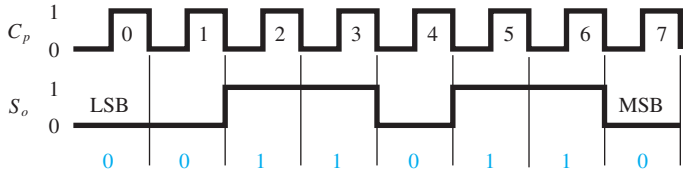


Figure 6 Serial representation of the binary number 0110110.

4 Parallel Representation

The parallel format uses a separate electrical conductor for each bit to be transmitted (and a common ground). For example, if the digital system is using 8-bit numbers, eight lines are required (see Figure 7). This tends to be expensive, but the entire 8-bit number can be transmitted in one clock period, making it very fast.

Inside a computer, binary data are almost always transmitted on parallel channels (collectively called the PCI *data bus*). Two parallel data techniques previously used by computers to communicate to external devices were the Centronics printer interface (port LPT1) and the Small Computer Systems Interface (SCSI, pronounced *scuzzy*).

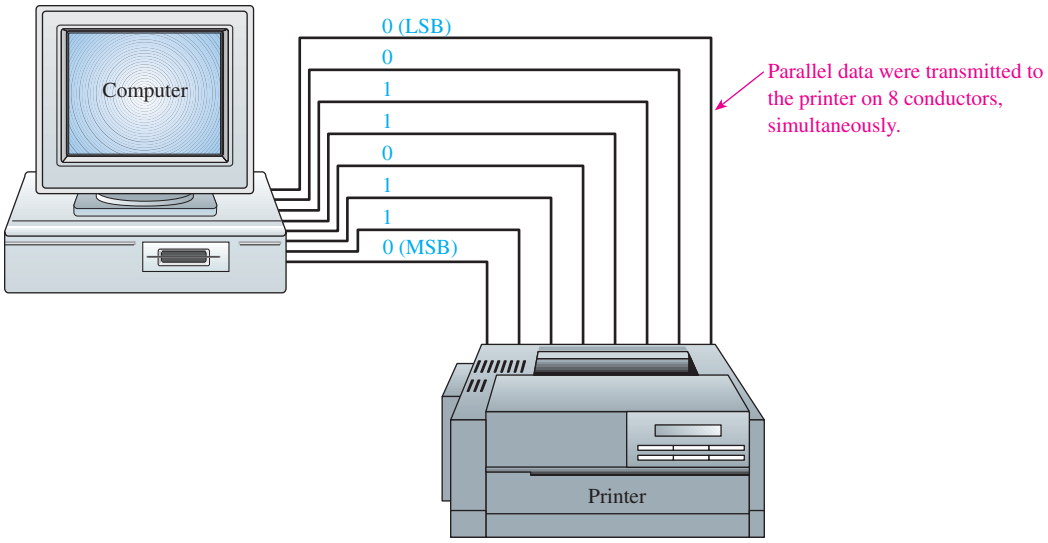


Figure 7 Original parallel communication between a computer and a printer.

Figure 8 illustrates the same binary number that was used in Figure 6 (01101100), this time in the parallel representation.

If the clock period were 2 μs, it would take 2 μs $\times$ 8 periods $=$ 16 μs to transmit the number in serial and only 2 μs $\times$ 1 period $=$ 2 μs to transmit the same 8-bit number in parallel. Thus, you can see that when speed is important, parallel transmission is preferred over serial transmission.

The following examples further illustrate the use of serial and parallel representations.

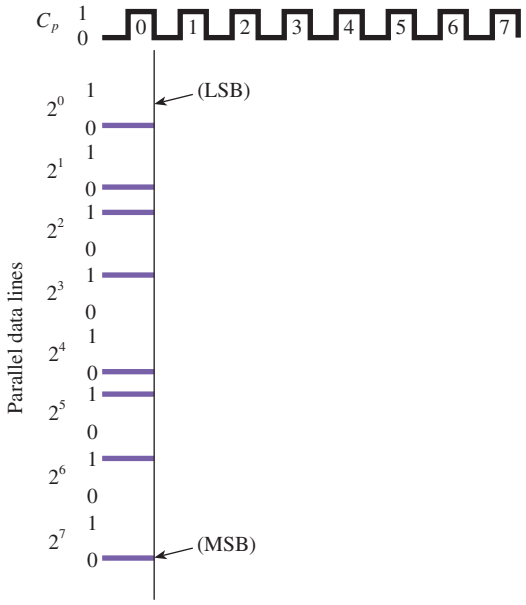


Figure 8 Parallel representation of the binary number 01101100.

Team Discussion

What other devices might use parallel communication? How about serial communication?

Inside Your PC

Most communication inside of a modern PC uses a parallel connection scheme. The newest internal parallel standard is called PCI (Peripheral Component Interconnect) and PCI-Express. These busses range anywhere from 1 to 32 bits in width and can transmit at speeds up to 16 Gbps!

EXAMPLE 5

Sketch the serial and parallel representations of the 4-bit number 0 1 1 1. If the clock frequency is 5 MHz, find the time to transmit using each method.

Solution: Figure 9 shows the representation of the 4-bit number 0 1 1 1.

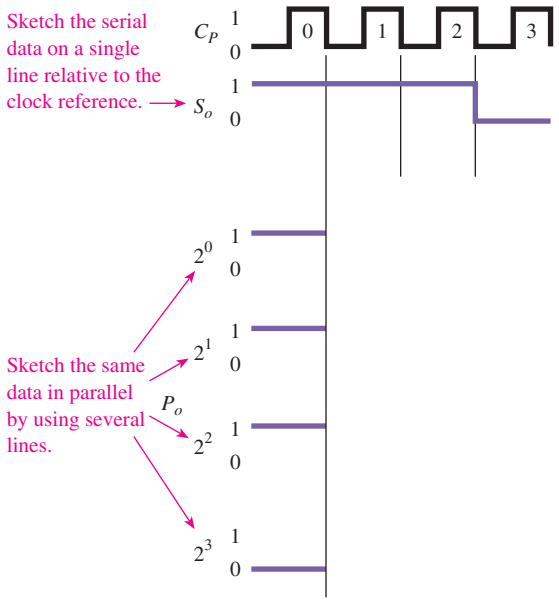


Figure 9

$$t_p = \frac{1}{f} = \frac{1}{5\text{ MHz}} = 0.2\,\mu\text{s}$$

$$t_{\text{serial}} = 4 \times 0.2\,\mu\text{s} = 0.8\,\mu\text{s}$$

$$t_{\text{parallel}} = 1 \times 0.2\,\mu\text{s} = 0.2\,\mu\text{s}$$

EXAMPLE 6

Sketch the serial and parallel representations of the decimal number 74. (Assume a clock frequency of 4 kHz.) Also, what is the state (1 or 0) of the serial line 1.2 ms into the transmission?

Solution: $74_{10} = 0\ 1\ 0\ 0\ 1\ 0\ 1\ 0_2$.

$$t_p = \frac{1}{f} = \frac{1}{4\text{ kHz}} = 0.25\text{ ms}$$

Therefore, the increment of time at each falling edge increases by 0.25 ms. Because each period is 0.25 ms, 1.2 ms will occur within the number 4 period, which, on the S_o line, is a ***0* logic state** (see Figure 10).

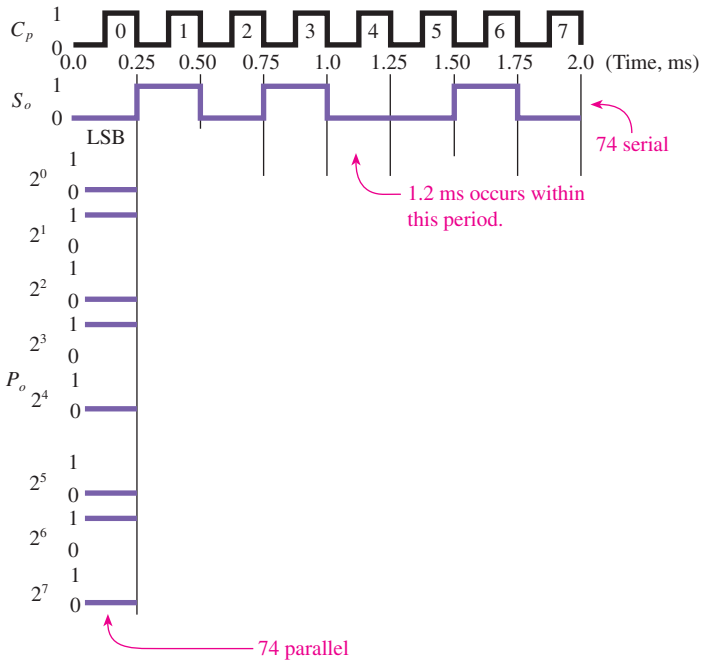


Figure 10

SERIAL TRANSMISSION SIMULATION

Figure 11 shows a MultiSIM simulation of the transmission of the three ASCII characters MP3 from a transmitting device (the Word Generator) to a receiving device (the Logic Analyzer). ASCII characters are generally transmitted most significant character first (but with the LSB of each 8-bit code coming first). The top trace in the Logic Analyzer displays a clock reference waveform (CP) of 24

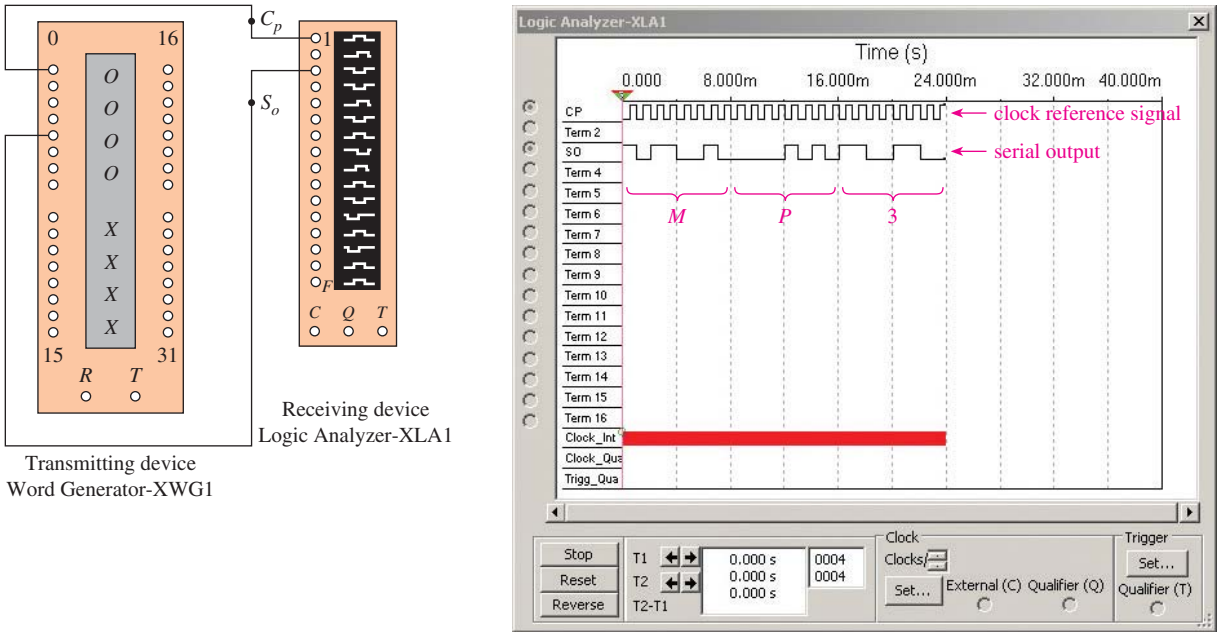


Figure 11 A MultiSIM simulation of the serial transmission of the ASCII characters *MP3*.

clock periods, each period lasting 1 ms. The third trace shows the serial output data (SO). Since ASCII is a 7-bit code, and since digital systems work in 8-bit groupings, a leading zero is added to the MSB of each ASCII code. Also, since the LSB of each character is output first (on the left), the bits read from 8 ms back to 0 ms are 01001101, which is the ASCII code for the letter *M*. Look up the next two 8-bit groupings in an ASCII chart and you will see that it is transmitting the letters *MP3*.

Exercise: (a) On graph paper, draw a 24-cycle CP reference waveform and then the 24-bit serial waveform for the ASCII letters *USB*. (b) Repeat for the letters *jpg*.

PARALLEL TRANSMISSION SIMULATION

Figure 12 shows a MultiSIM simulation of the transmission of the three parallel ASCII characters *Y2K* from a transmitting device (the Word Generator) to a receiving device (the Logic Analyzer). The top trace in the Logic Analyzer displays a clock reference waveform (CP) of 3 clock periods, each period lasting 1 ms. The next eight traces show the parallel output data (PO-P7). Since ASCII is a 7-bit code and since digital systems work in 8-bit groupings, a leading zero is added to the MSB of each ASCII code. During the first period (the first column), the parallel data lines contain the code 0101 1001, which is the ASCII code for the letter *Y*. Look up the next two 8-bit columns in an ASCII chart and you will see that it is transmitting the letters *Y2K*.

Exercise: (a) On graph paper, draw a 3-period CP reference waveform and then the 3-bit parallel waveforms for the ASCII letters *ATM*. (b) Repeat for the letters *CDR*.

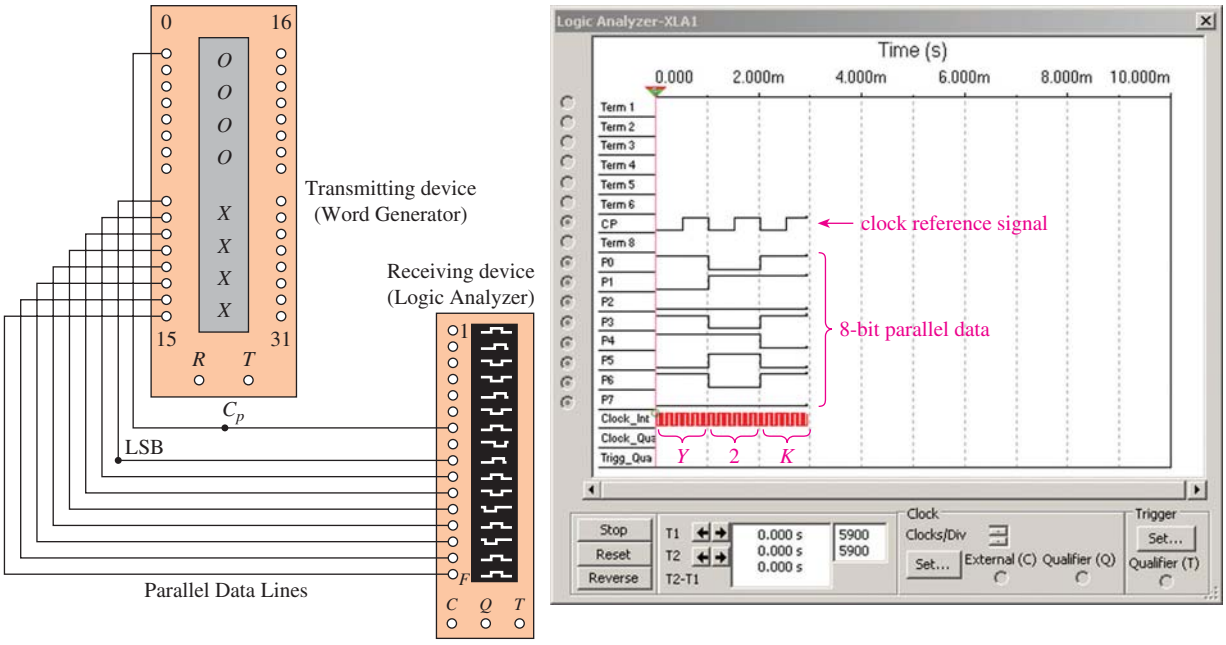


Figure 12 A MultiSIM simulation of the parallel transmission of the ASCII characters *Y2K*.

Review Questions

9. What advantage does parallel have over serial in the transmission of digital signals?

10. Which system requires more electrical conductors and circuitry, serial or parallel?

11. How long will it take to transmit three 8-bit binary strings in serial if the clock frequency is 5 MHz?

12. Repeat Question 11 for an 8-bit parallel system.

5 Switches in Electronic Circuits*

The transitions between 0 and 1 digital levels are caused by switching from one voltage level to another (usually 0 V to +5 V). One way that switching is accomplished is to make and break a connection between two electrical conductors by way of a manual switch or an electromechanical relay. Another way to switch digital levels is by use of semiconductor devices such as **diodes** and **transistors**.

Manual switches and relays have almost *ideal* ON and OFF resistances in that when their contacts are closed (ON) the resistance (measured by an ohmmeter) is 0 ohms (Ω) and current is allowed to flow. When their contacts are open (OFF), the resistance is infinite and no current can flow. Figures 13(a) and (b) show the single-pole, single-throw manual switch. When used in a digital circuit, a single-pole, double-throw manual switch can produce 0 and 1 states at some output terminal, as shown in Figures 13(c) and 13(d), by moving the switch (SW) to the up or down position.

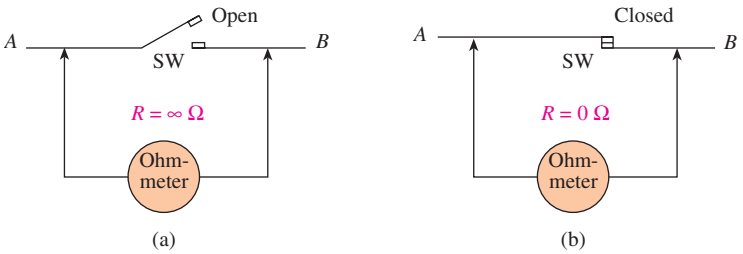


Figure 13 Manual switch: (a) switch open, $R = \infty$ ohms; (b) switch closed, $R = 0$ ohms.

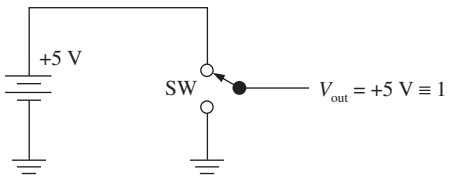


Figure 13(c) 1-Level output.

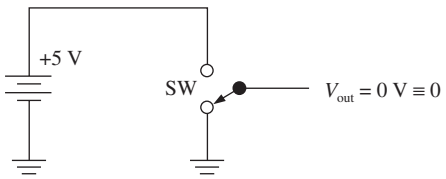
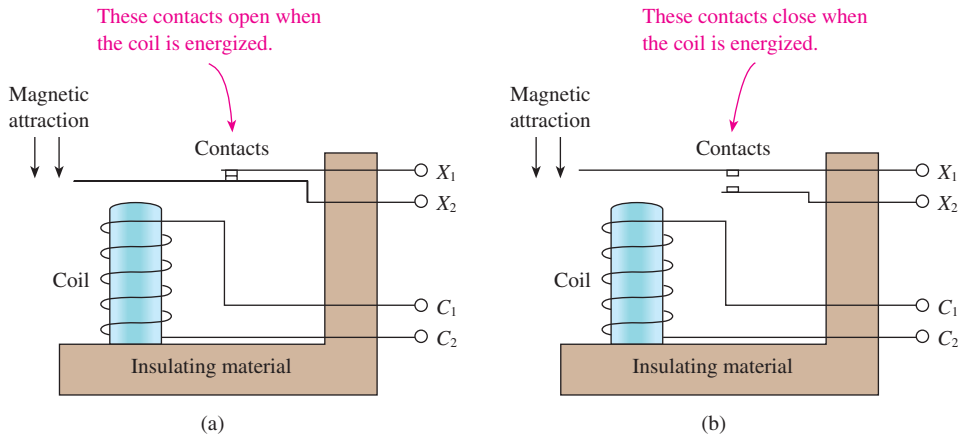


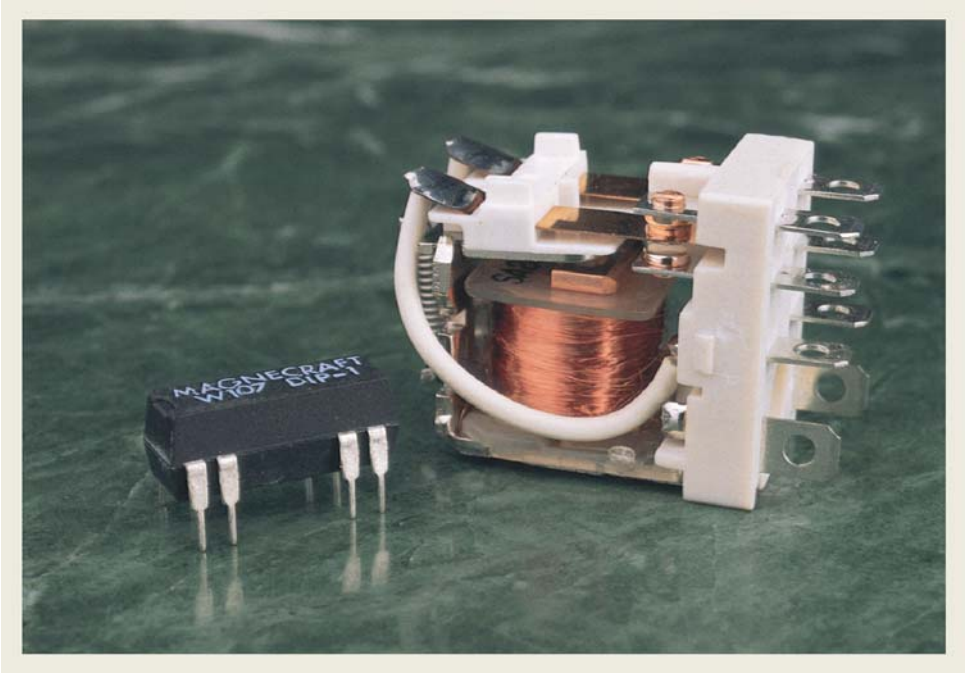
Figure 13(d) 0-Level output.

*The fundamentals of basic electricity are provided in Appendix: Review of Basic Electricity Principles. Ohm's law, simple series circuits, open circuits, and short circuits are explained to help you understand the electrical principles used in the remainder of this chapter.

6 A Relay as a Switch*

An electromechanical **relay** has contacts like a manual switch, but it is controlled by external voltage instead of being operated manually. They are often used to deliver HIGH/LOW digital levels to a high power load like a motor or a high-wattage lamp. Figure 14 shows the physical layout of an electromechanical relay. In Figure 14(a) the magnetic coil is energized by placing a voltage at terminals C_1–C_2; this will cause the lower contact to bend downward, opening the contact between X_1 and X_2. This relay is called *normally closed* (NC) because, at rest, the contacts are touching, or closed. In Figure 14(b), when the coil is energized, the upper contact will be





(c)

Figure 14 Physical representation of an electromechanical relay: (a) normally closed (NC) relay; (b) normally open (NO) relay; (c) photograph of actual relays.

*Systems requiring complex relay switching schemes are generally implemented using programmable logic controllers (PLCs). PLCs are microprocessor-based systems that are programmed to perform complex logic operations, usually to control electrical processes in manufacturing and industrial facilities. They use a programming technique called *ladder logic* to monitor and control several processes, eliminating the need for individually wired relays. PLC is a registered trademark of Allen-Bradley Corporation.

attracted downward, making a connection between X_1 and X_2. This is called a *normally open* (NO) relay because at rest, the contacts are not touching, they are open.

A relay provides total isolation between the triggering source applied to C_1-C_2 and the output X_1-X_2. This total isolation is important in many digital applications, and it is a feature that certain semiconductor switches (e.g., transistors, diodes, and integrated circuits) cannot provide. Also, the contacts are normally rated for currents much higher than the current rating of semiconductor switches.

There are several disadvantages, however, of using a relay in electronic circuits. To energize the relay coil, the triggering device must supply several milliamperes, whereas a semiconductor requires only a few microamperes to operate. A relay is also much slower than a semiconductor. It will take several milliseconds to switch, compared to microseconds (or nanoseconds) for a semiconductor switch.

In Figure 15 a relay is used as a shorting switch in an electric circuit. The +5-V source is used to energize the coil, and the +12-V source is supplying the external electric circuit. When the switch (SW) in Figure 15(a) is closed, the relay coil will become energized, causing the relay contacts to open, which will make V_{out} change from 0 V to 6 V with respect to ground. The voltage-divider equation is used to calculate V_{out} as follows:

$$V_{out} = \frac{12\ \text{V} \times 5\ \text{k}\Omega}{5\ \text{k}\Omega + 5\ \text{k}\Omega} = 6\ \text{V}$$

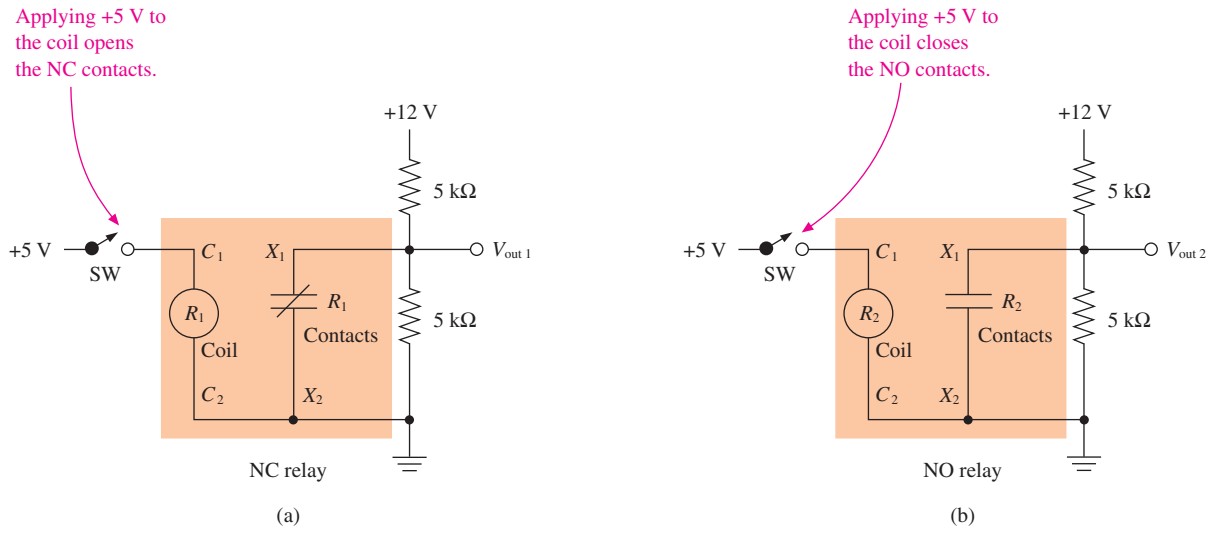


Figure 15 Symbolic representation of an electromechanical relay: (a) NC relay used in a circuit and (b) NO relay used in a circuit.

When the switch in Figure 15(b) is closed, the relay coil becomes an **energized relay coil**, causing the relay contacts to close, changing V_{out2} from 6 V to 0 V.

Now, let's go a step further and replace the 5-V battery and switch with a clock oscillator and use a timing diagram to analyze the results. In Figure 16, the relay is triggered by the clock waveform, C_p. The diode D_1 is placed across the relay coil to protect it from arcing each time the coil is deenergized. Timing diagrams are very useful for comparing one waveform to another because the waveform changes states (1 or 0) relative to time. The timing diagram in Figure 17 shows that when the clock goes HIGH (1), the relay is energized, causing V_{out3} to go LOW (0). When C_p goes LOW (0), the relay is deenergized, causing V_{out3} to go to +5 V (using the voltage divider equation, $V_{out} = [10\ \text{V} \times 5\ \text{k}\Omega]/[5\ \text{k}\Omega + 5\ \text{k}\Omega] = 5\ \text{V}$).

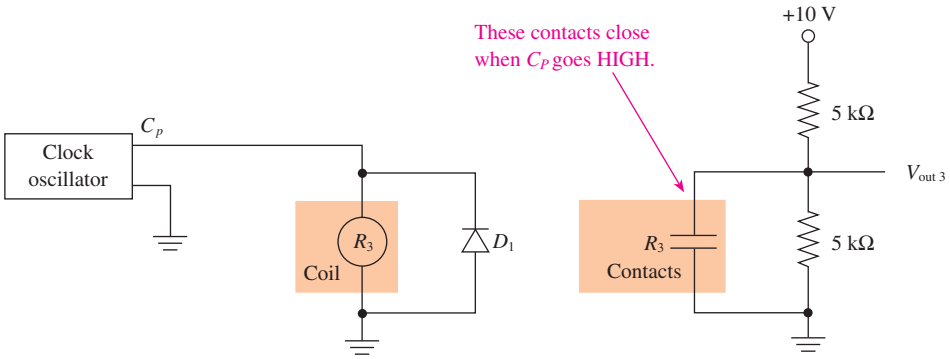


Figure 16 Relay used in a digital circuit.

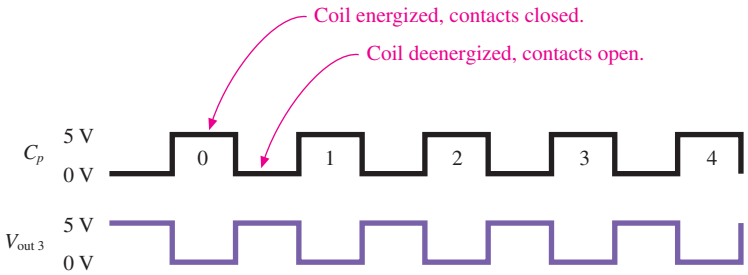


Figure 17 Timing diagram for Figure 16.

The following examples illustrate electronic switching and will help to prepare you for more complex timing analysis.

Common Misconception

The effects of opens and shorts are often miscalculated. Occasionally, it is instructive to assume that an open is equivalent to a 10-MΩ resistor and calculate the voltage across it using the voltage divider equation.

EXAMPLE 7

Draw a timing diagram for the circuit shown in Figure 18, given the C_p waveform in Figure 19.

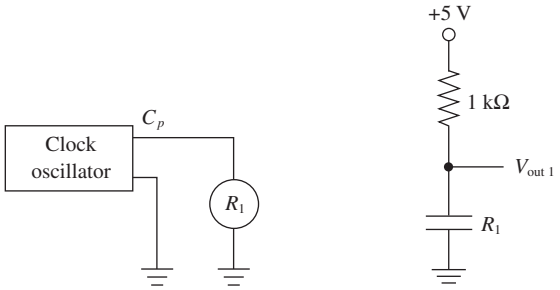


Figure 18

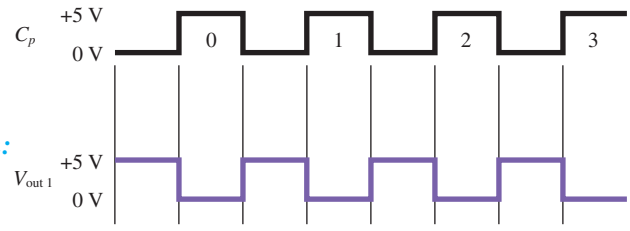


Figure 19

Explanation: When C_p is LOW, the R_1 coil is deenergized, the R_1 contacts are open, $I_{1\,k\Omega} = 0\,A$, $V_{drop1k\Omega} = I \times R = 0\,V$, and $V_{out1} = 5\,V - 0_{V\,drop} = 5\,V$. When C_p is HIGH, the R_1 coil is energized, the R_1 contacts are closed, and $V_{out1} = 0\,V$.

EXAMPLE 8

Draw a timing diagram for the circuit shown in Figure 20(a), given the C_p waveform in Figure 20(b).

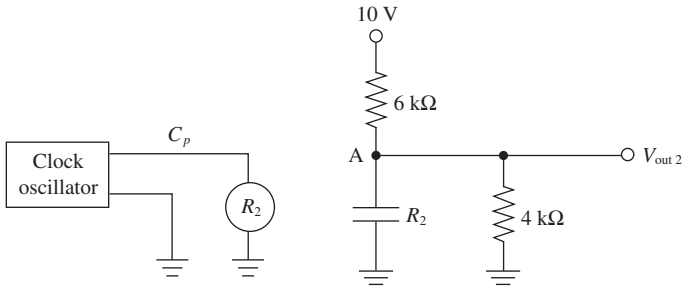
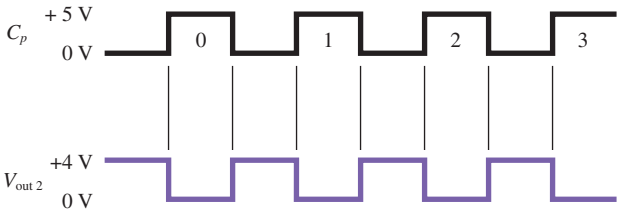


Figure 20(a)



Solution:

Figure 20(b)

Explanation: When the R_2 contacts are closed (R_2 is energized), the voltage at point A is 0 V, making V_{out2} equal to 0 V. When the R_2 contacts are open (R_2 is deenergized), the voltage at point A is $V_A = \dfrac{10\,V \times 4\,k\Omega}{6\,k\Omega + 4\,k\Omega} = 4\,V$ and $V_{out2} = V_A = 4\,V$.

Helpful Hint

Remember that V_{out} is the voltage measured from the point in question to ground.

ELECTRO-MECHANICAL RELAY SWITCHING SIMULATION

Figure 21 shows a MultiSIM simulation of a relay connected in a voltage-divider circuit. As the clock energizes/de-energizes the relay coil, the relay contacts repeatedly short the 8 k resistor, causing the V_{out} waveform (Channel_B) to change from 0 V (3.000 μV) to 8 V repeatedly as shown in the oscilloscope display.

MultiSIM Exercise: Use MultiSIM to open file *fig02_21* from the text website. Run the simulation to create the waveforms shown in Figure 21. Move the measurement cursers '1' and '2' to display the voltage levels shown. Make the following changes, predict the new values for V_{out} and rerun the simulation:

(a) Change the 4 k to 8 k and the 8 k to 4 k.

(b) Change the top resistor to 20 k and the bottom resistor to 4 k.

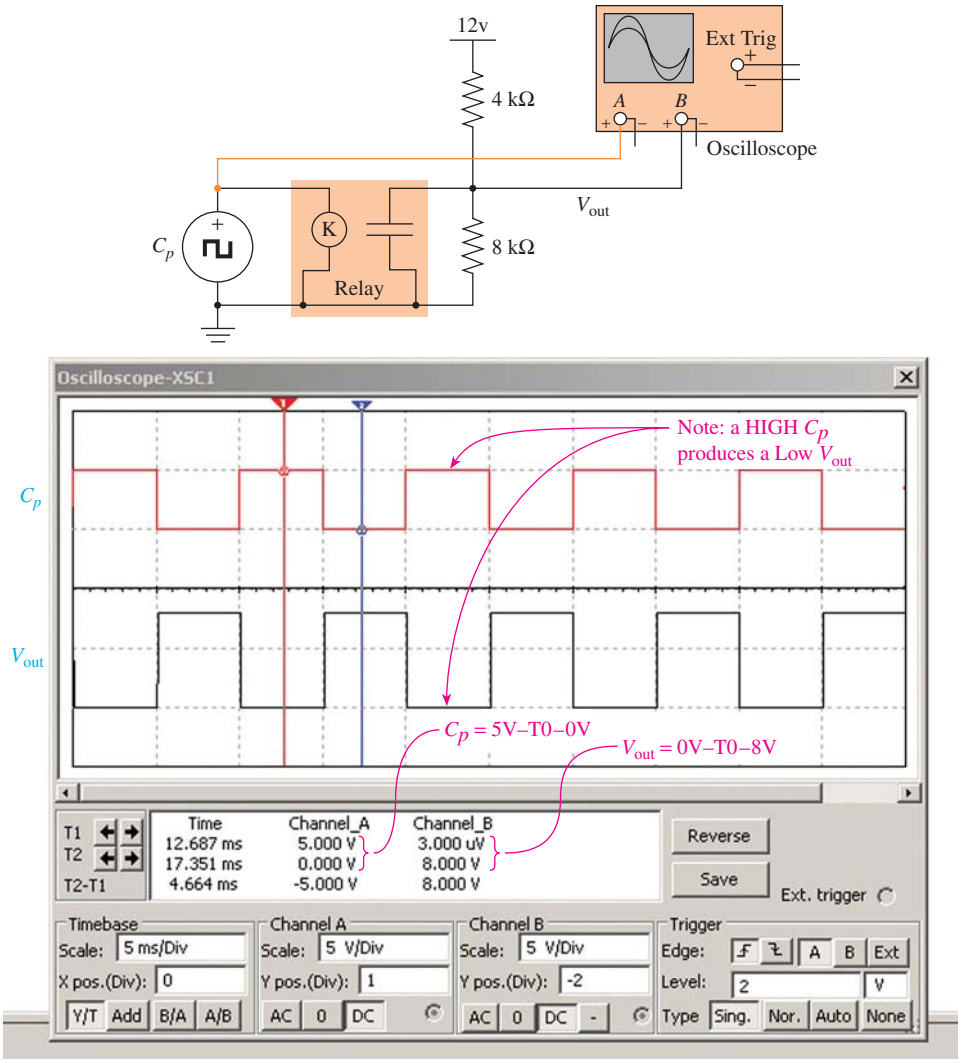


Figure 21 A MultiSIM simulation of an electro-mechanical relay switching circuit.

Review Questions

13. Describe the operation of a relay coil and relay contacts.

14. How does a normally open relay differ from a normally closed relay?

7 A Diode as a Switch

Manual switches and electromechanical relays have limited application in today's digital electronic circuits. Most digital systems are based on semiconductor technology, which uses diodes and transistors. Most electronics students should also take a separate course in electronic devices to cover the in-depth theory of the operation of diodes and transistors. However, without getting into a lot of detail, let's look at how a diode and a transistor can operate as a simple ON/OFF switch.

A diode is a semiconductor device that allows current to flow in one direction but not the other. Figure 22 shows a diode in both the conducting and nonconducting

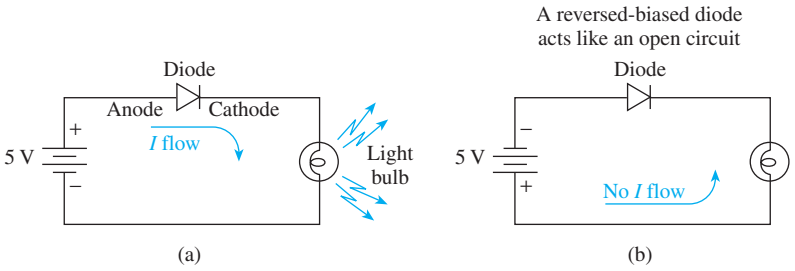


Figure 22 Diode in a series circuit: (a) forward biased and (b) reverse biased.

states. The term *forward biased* refers to a diode whose anode voltage is *more positive* than its cathode, thus allowing current flow in the direction of the arrow. (**Bias** is the voltage necessary to cause a semiconductor device to conduct or cut off current flow.) A reverse-biased diode will not allow current flow because its anode voltage is *equal to* or *more negative* than its cathode. A diode is analogous to a check valve in a water system (see Figure 23).

A diode is not a perfect short in the forward-biased condition, however. The voltage-versus-current curve shown in Figure 24 shows the characteristics of a diode. Notice in the figure that for the reverse-biased condition, as V_{rev} becomes more negative, there is still practically zero current flow.

In the forward-biased condition, as V_{forw} becomes more positive, no current flows until a 0.7-V cut-in voltage is reached.* After that point, the voltage across the diode (V_{forw}) will remain at approximately 0.7 V, and I_{forw} will flow, limited only by the external resistance of the circuit and the 0.7-V internal voltage drop.

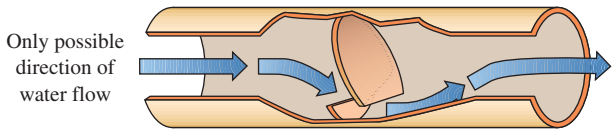


Figure 23 Water system check valve.

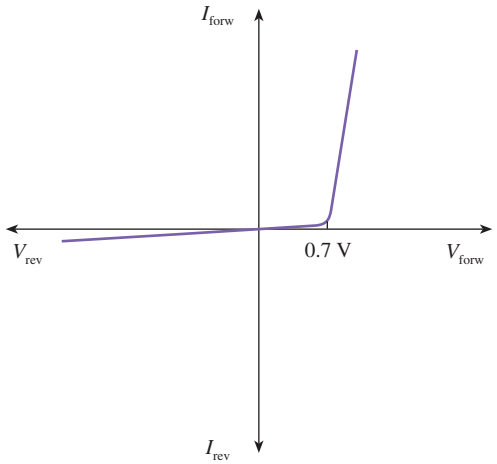


Figure 24 Diode voltage versus current characteristic curve.

*0.7 V is the typical cut-in voltage of a silicon diode, whereas 0.3 V is typical for a germanium diode. We use the silicon diode because it is most commonly used in digital circuitry.

What this means is that current will flow only if the anode is more positive than the cathode, and under those conditions, the diode acts like a short circuit except for the 0.7 V across its terminals. This fact is better illustrated in Figure 25.

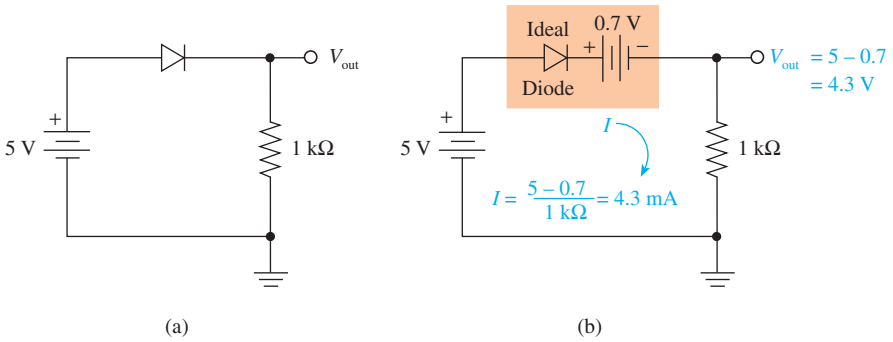


Figure 25 Forward-biased diode in an electric circuit: (a) original circuit and (b) equivalent circuit showing the diode voltage drop and $V_{out} = 5 - 0.7 = 4.3$ V.

The following examples and the problems at the end of the chapter demonstrate the effect that diodes have on electric circuits.

EXAMPLE 9

Determine if the diodes shown in Figure 26 are forward or reverse biased.

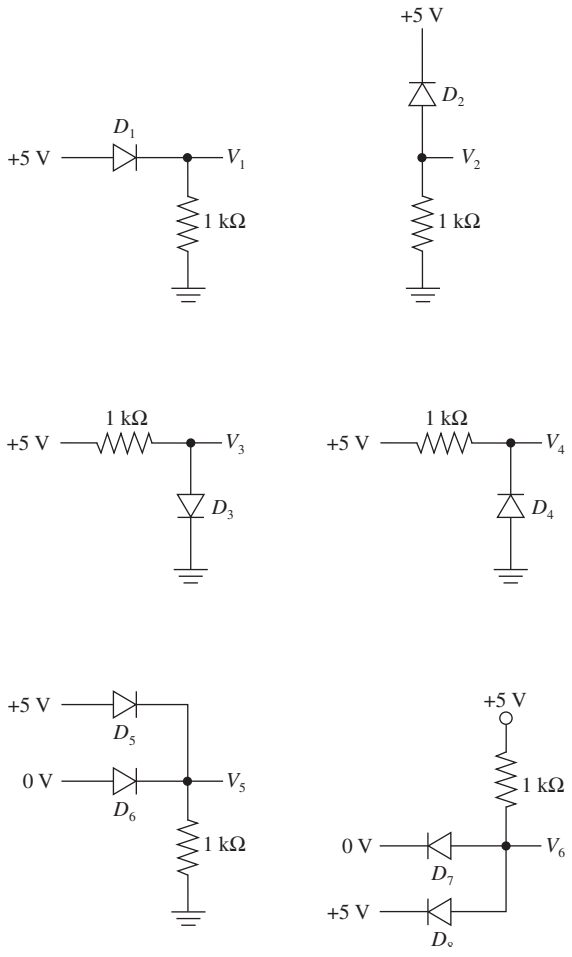


Figure 26

Solution: The diode is forward biased if the anode is more positive than the cathode.

D_1 is forward biased.
D_2 is reverse biased.
D_3 is forward biased.
D_4 is reverse biased.
D_5 is forward biased.
D_6 is reverse biased.
D_7 is forward biased.
D_8 is reverse biased.

EXAMPLE 10

Determine V_1, V_2, V_3, and V_4 (with respect to ground) for the circuits in Example 9.

Solution: V_1: D_1 is forward biased, dropping 0.7 V across its terminals. Therefore, $V_1 = 4.3$ V (5.0 − 0.7).

V_2: D_2 is reverse biased. No current will flow through the 1-kΩ resistor, so $V_2 = 0$ V.

V_3: D_3 is forward biased, dropping 0.7 V across its terminals, making $V_3 = 0.7$ V.

V_4: D_4 is reverse biased, acting like an open. Therefore, $V_4 = 5$ V.

V_5: Because D_6 is reverse biased (open), it has no effect on the circuit. D_5 is forward biased, dropping 0.7 V, making $V_5 = 4.3$ V.

V_6: D_8 is reverse biased (open), so it has no effect on the circuit. D_7 is forward biased, so it has +0.7 V on its anode side, which is +0.7 above the 0-V ground level, making $V_6 = +0.7$ V.

Review Questions

15. To forward bias a diode, the anode is made more _____ (positive/negative) than the cathode.

16. A forward-biased diode has how many volts across its terminals?

8 A Transistor as a Switch

The bipolar transistor is a very commonly used switch in digital electronic circuits. It is a three-terminal semiconductor component that allows an input signal at one of its terminals to cause the other two terminals to become a short or an open circuit. The transistor is most commonly made of silicon that has been altered into *N*-type material and *P*-type material. *N*-type silicon is made by bombarding pure silicon with atoms having structures with *one more* electron than silicon does. *P*-type silicon is made by bombarding pure silicon with atoms having structures with *one less* electron than silicon does.

Three distinct regions make up a bipolar transistor: *emitter, base,* and *collector.* They can be a combination of *N*-*P*-*N*-type material or *P*-*N*-*P*-type material bonded together as a three-terminal device. Figure 27 shows the physical layout and symbol for an *NPN* transistor. (In a *PNP* transistor, the emitter arrow points the other way.)

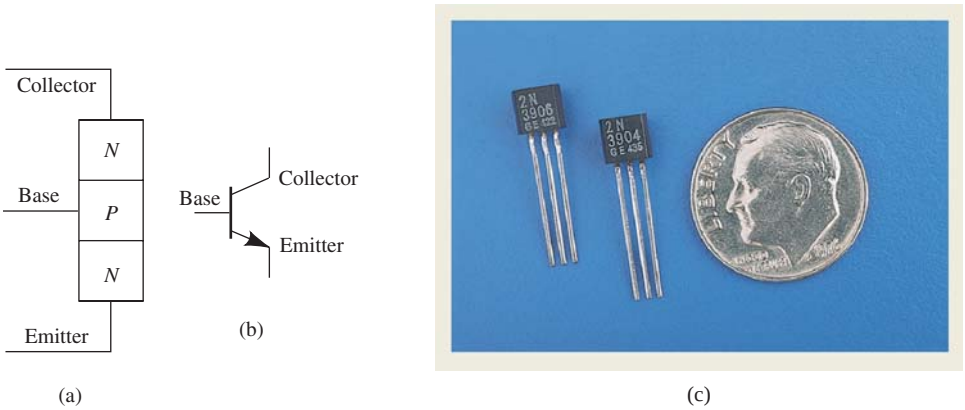


Figure 27 The *NPN* bipolar transistor: (a) physical layout; (b) symbol; (c) photograph.

In an electronic circuit, the input signal (1 or 0) is usually applied to the base of the transistor, which causes the collector–emitter junction to become a short or an open circuit. The rules of transistor switching are as follows:

1. In an *NPN* transistor, applying a positive voltage from base to emitter causes the collector-to-emitter junction to short (this is called "turning the transistor ON"). Applying a negative voltage or 0 V from base to emitter causes the collector-to-emitter junction to open (this is called "turning the transistor OFF").
2. In a *PNP** transistor, applying a negative voltage from base to emitter turns it ON. Applying a positive voltage or 0 V from base to emitter turns it OFF.

Figure 28 shows how an *NPN* transistor functions as a switch in an electronic circuit. In the figure, resistors R_B and R_C are used to limit the base current and the collector current. In Figure 28(a), the transistor is turned ON because the base is more positive than the emitter (input signal = +2 V). This causes the collector-to-emitter junction to short, placing ground potential at V_{out} (V_{out} = 0 V).

Common Misconception

Students often think that the input signal to the base of a transistor must somehow be part of the output at the collector or emitter, but it is not. Once you determine if the *C*-to-*E* is a short or an open, you can ignore the base circuit altogether.

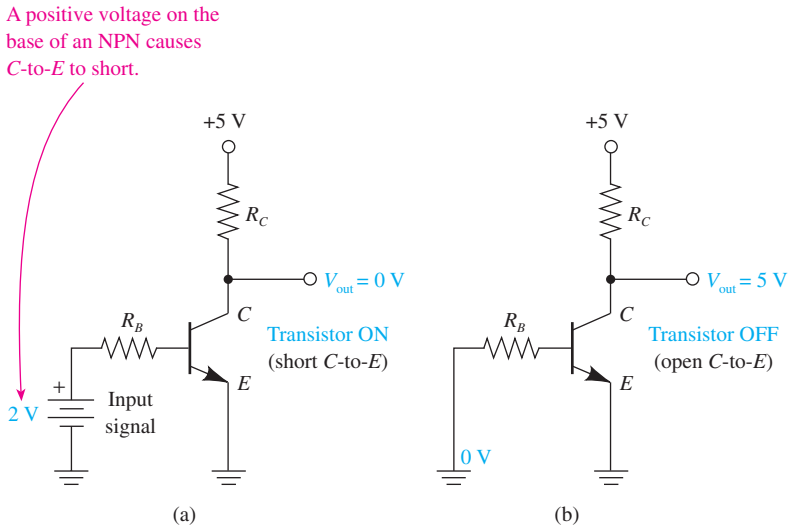


Figure 28 *NPN* transistor switch: (a) transistor ON and (b) transistor OFF.

*PNP transistor circuits are analyzed in the same way as *NPN* circuits except that all voltage and current polarities are reversed. *NPN* circuits are much more common in industry and will be used most often in this text.

In Figure 28(b), the input signal is removed, making the base-to-emitter junction 0 V, turning the transistor OFF. With the transistor OFF, there is no current (0 amps) through R_C, so $V_{out} = 5\text{ V} - (0\text{ A} \times R_C) = 5\text{ V}$.

Digital input signals are usually brought in at the base of the transistor, and the output is taken off the collector or emitter. The following examples use timing analysis to compare the input and output waveforms.

EXAMPLE 11

Sketch the waveform at V_{out} in the circuit shown in Figure 29, given the input signal C_p in Figure 30.

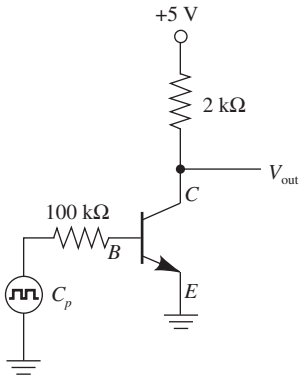


Figure 29

Solution:

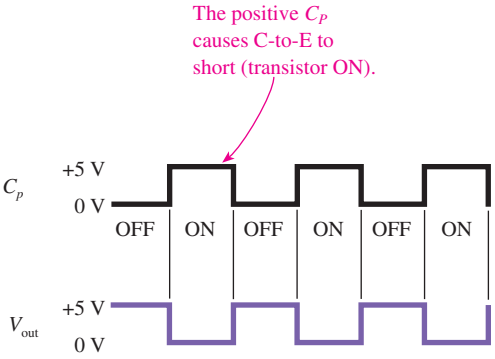


Figure 30

Explanation: When $C_p = 0\text{ V}$, the transistor is OFF and the equivalent circuit is as shown in Figure 31(a).

$$I_C = 0\text{ A}$$

Therefore,

$$V_C = 5\text{ V} - (0\text{ A} \times 2\text{ k}\Omega) = 5\text{ V}$$

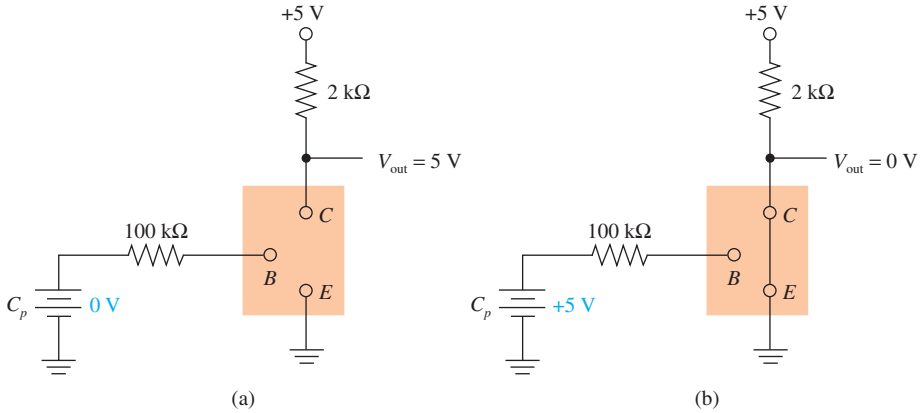


Figure 31 Equivalent circuits: (a) transistor OFF and (b) transistor ON.

When $C_p = +5\text{ V}$, the transistor is ON and the equivalent circuit is as shown in Figure 31(b). The collector is shorted directly to ground; therefore, $V_{out} = 0\text{ V}$.

EXAMPLE 12

Sketch the waveform at V_{out} in the circuit shown in Figure 32, given the input signal C_p in Figure 33.

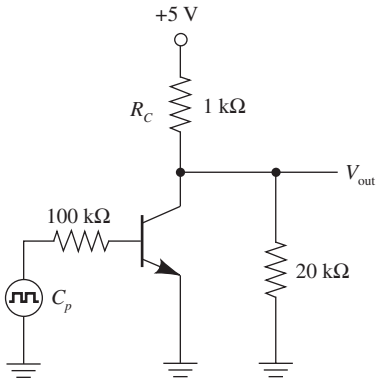


Figure 32

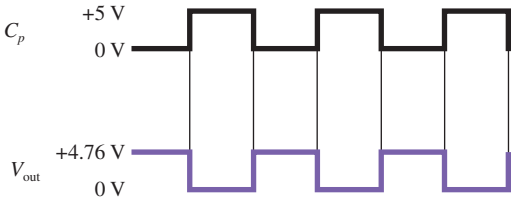


Figure 33

Solution:

Explanation: When $C_p = 0$ V, the transistor is OFF and the equivalent circuit is as shown in Figure 34(a). From the voltage-divider equation,

$$V_{out} = \frac{5\text{ V} \times 20\text{ k}\Omega}{20\text{ k}\Omega + 1\text{ k}\Omega} = 4.76\text{ V}$$

Next, when $C_p = +5$ V, the transistor is ON and the equivalent circuit is as shown in Figure 34(b). Now the collector is shorted to ground, making $V_{out} = 0$ V. Notice the difference in V_{out} as compared to Example 11, which had no load resistor connected to V_{out}.

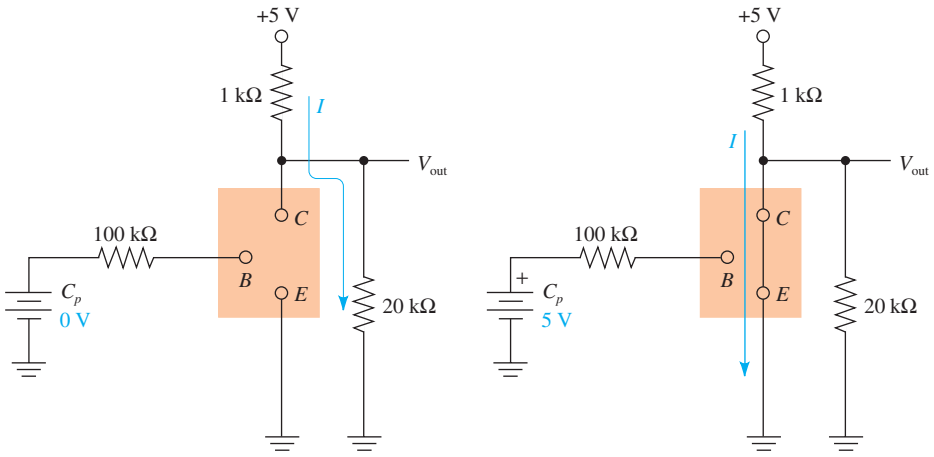


Figure 34 Equivalent circuits: (a) transistor OFF and (b) transistor ON.

Review Questions

17. Name the three pins on a transistor.

18. To turn ON an *NPN* transistor, a _____ (positive/negative) voltage is applied to the base.

19. When a transistor is turned ON, its collector-to-emitter becomes a _____ (short/open).

9 The TTL Integrated Circuit

Transistor–transistor logic **(TTL)** is one of the most widely used integrated-circuit technologies. TTL integrated circuits use a combination of several transistors, diodes, and resistors integrated together in a single package.

One basic function of a TTL integrated circuit is as a complementing switch, or **inverter**. The inverter is used to take a digital level at its input and complement it to the opposite state at its output (1 becomes 0, 0 becomes 1). Figure 35 shows how a common-emitter-connected transistor switch can be used to perform the same function.

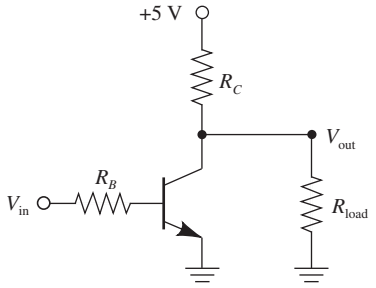


Figure 35 Common-emitter transistor circuit operating as an inverter.

When V_{in} equals 1 (+5 V), the transistor is turned on (called **saturation**) and V_{out} equals 0 (0 V). When V_{in} equals 0 (0 V), the transistor is turned off (called **cutoff**) and V_{out} equals 1 (approximately 5 V), assuming that R_L is much greater than R_C ($R_L \gg R_C$).

EXAMPLE 13

Let's assume that $R_C = 1\,\text{k}\Omega$, $R_L = 10\,\text{k}\Omega$, and $V_{in} = 0$ in Figure 35. V_{out} will equal 4.55 V:

$$\frac{5\text{ V} \times 10\,\text{k}\Omega}{1\,\text{k}\Omega + 10\,\text{k}\Omega} = 4.55\text{ V}$$

But if R_L decreases to $1\,\text{k}\Omega$ by adding more loads in parallel with it, V_{out} will drop to 2.5 V:

$$\frac{5\text{ V} \times 1\,\text{k}\Omega}{1\,\text{k}\Omega + 1\,\text{k}\Omega} = 2.5\text{ V}$$

We can see from Example 13 that the 1-level output of the inverter is very dependent on the size of the load resistor (R_L), which can typically vary by a factor of 10. So right away you might say, "Let's keep R_C very small so that R_L is always much

greater than R_C" ($R_L \gg R_C$). Well, that's fine for the case when the transistor is cut off ($V_{out} = 1$), but when the transistor is saturated ($V_{out} = 0$), the transistor collector current will be excessive if R_C is very small ($I_C = 5\ V/R_C$; see Figure 36).

Helpful Hint

If you understand the idea that V_{out} varies depending on the size of the connected load, it will help you understand why gate outputs are not exactly 0 V and 5 V.

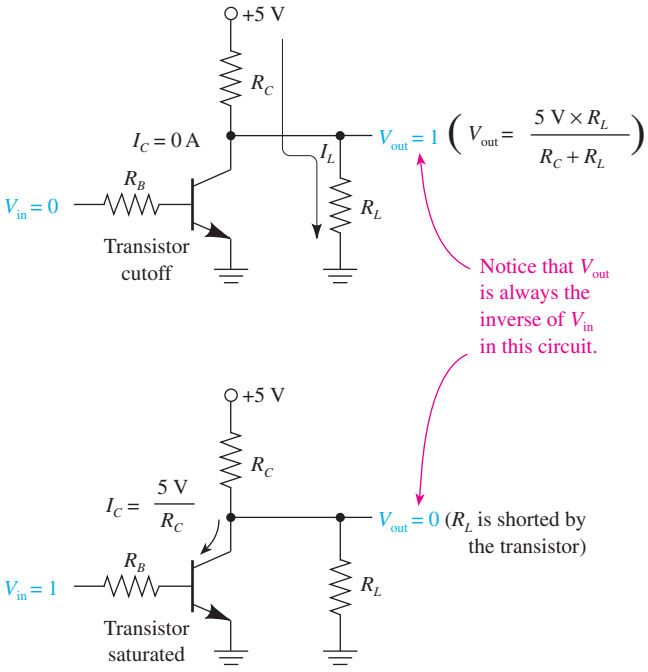


Figure 36 Common-emitter calculations.

Therefore, it seems that when the transistor is cut off ($V_{out} = 1$), we want R_C to be small to ensure that V_{out} is close to 5 V, but when the transistor is saturated, we want R_C to be large to avoid excessive collector current.

This idea of needing a variable R_C resistance is accommodated by the TTL **integrated circuit** (Figure 37). It uses another transistor (Q_4) in place of R_C to act like a varying resistance. Q_4 is cut off (acts like a high R_C) when the output transistor (Q_3) is saturated, and then Q_4 is saturated (acts like a low R_C) when Q_3 is cut off. (In other words, when one transistor is ON, the other one is OFF.) This combination of Q_3 and Q_4 is referred to as the **totem-pole** arrangement.

Transistor Q_1 is the input transistor used to drive Q_2, which is used to control Q_3 and Q_4. Diode D_1 is used to protect Q_1 from negative voltages that might inadvertently be placed at the input. D_2 is used to ensure that when Q_3 is saturated, Q_4 will be cut off totally. V_{CC} is the abbreviation used to signify the power supply to the integrated circuit.

TTL is a very popular family of integrated circuits. It is much more widely used than RTL (resistor–transistor logic) or DTL (diode–transistor logic) circuits, which were the forerunners of TTL. Note that V_{out} is not exactly 0 V and 5 V (it is more typically 0.2 V and 3.4 V).

A single TTL integrated-circuit (IC) package such as the 7404 has six complete logic circuits fabricated into a single silicon **chip**, each logic circuit being the equivalent of Figure 37. The 7404 has 14 metallic pins connected to the outside of a plastic case containing the silicon chip. The 14 pins, arranged 7 on a side, are aligned on 14 holes of a printed-circuit board, where they are then soldered. The 7404 is called a 14-pin **DIP** (dual-in-line package) and costs less than 24 cents. Figure 38 shows a sketch of a 14-pin DIP IC.

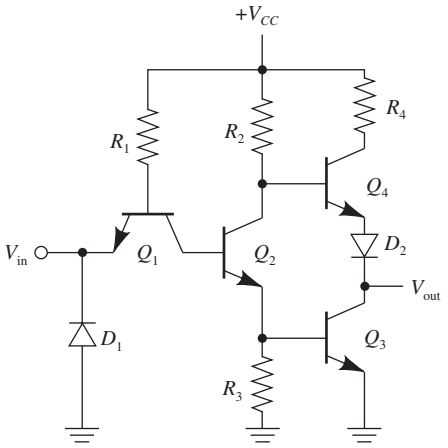


Figure 37 Schematic of a TTL inverter circuit.

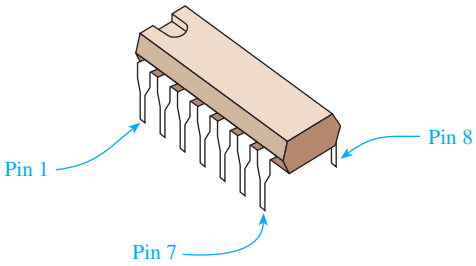


Figure 38 A 7404 TTL IC chip.

ICs are configured as DIPs to ensure that the mechanical stress exerted on the pins when being inserted into a socket is equally distributed and that, although most of these pins serve as conductors to either the gates' inputs or outputs, some simply provide structural support and are simply anchored to the IC casing. These latter pins are denoted by the letters NC, meaning that they are *not* physically or electrically *connected* to an internal component.

The pin configuration of the 7404 is shown in Figure 39. The power supply connections to the IC are made to pin 14 (+5 V) and pin 7 (ground), which supplies power to all six logic circuits. In the case of the 7404, the logic circuits are called *inverters*. The symbol for each inverter is a triangle with a circle at the output. The circle is used to indicate the inversion function. Although *never* shown in the pin configuration top view of digital ICs, each gate is electrically tied internally to both V_{CC} and ground. The entire circuit shown in Figure 37 is contained inside each of the six inverters.

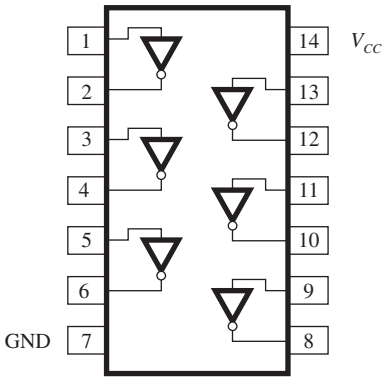


Figure 39 A 7404 hex inverter pin configuration.

Figure 40 shows three different ICs next to a pencil to give you an idea of their size.

10 MultiSIM® Simulation of Switching Circuits

The MultiSIM® software is useful for designing and simulating digital logic before building the actual circuits in the lab. Figure 41 shows four switching circuits that employ switches, transistors, inverter gates, and light-emitting diodes (LEDs). LEDs are

Figure 40 Photograph of three commonly used ICs: the 74HC00, 74ACT244, and 74150.

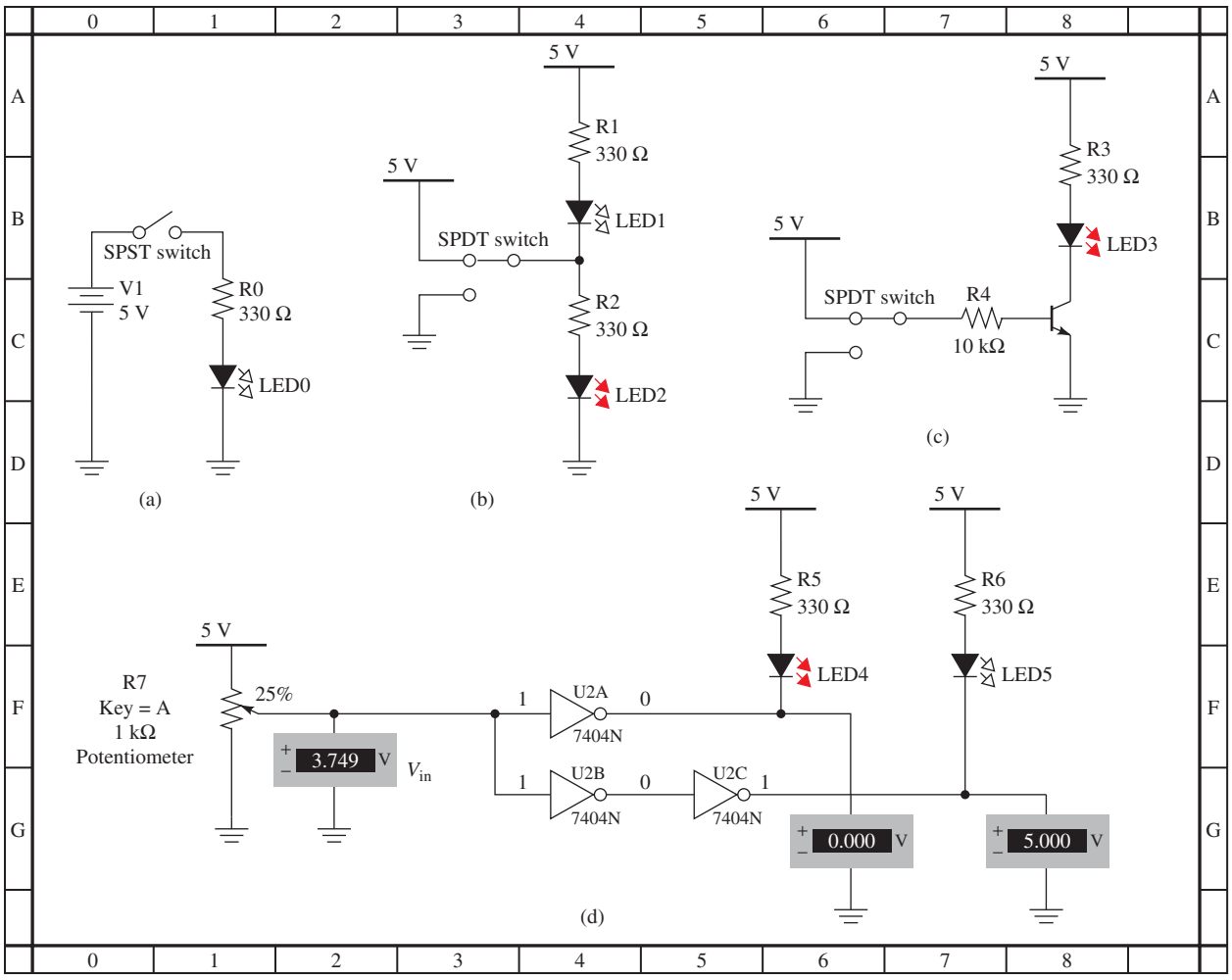


Figure 41 MultiSIM® simulation of switching circuits.

special diodes that illuminate when forward biased. They are often used in digital circuitry to indicate HIGH/LOW logic levels.

If you have already installed MultiSIM on your computer, you can load the circuit file named *fig2_41* from the text website and run the simulation shown in Figure 41. In Figure 41(a), if the single-pole single-throw (SPST) switch is in the UP position, no current can flow and the LED will not illuminate. With the switch thrown DOWN, 5 V are applied to the circuit, which forward biases the LED and makes it illuminate. (You can simulate this action by repeatedly pressing the space bar on your computer to make the switch go DOWN and UP. Notice that MultiSIM designates an ON LED by making the LED arrows RED.)

Figure 41(b) uses a single-pole double-throw (SPDT) switch to input HIGH/LOW levels into the circuit. With the switch in the UP position the current flows through the lower circuit, illuminating LED2. With the switch DOWN, current is instead allowed to flow down through LED1 via the 5-V supply and R_1. Run the simulation and watch the active LED as you throw the switch by pressing the space bar.

Figure 41(c) uses an NPN transistor to supply the current for the LED. In the previous circuits, all of the LED current was funneled through the switch. In this circuit the switch is used to "turn ON" or "turn OFF" the transistor, which in turn provides a path for the current to flow to ground through the collector to the emitter. (The transistor base current required to turn ON a transistor is typically 0.5 mA, whereas the LED current is typically 10 mA.) This is important because the switches in Figure 41 (a) and (b) are replaced by digital logic ICs that may not be able to pass 10 mA as the transistor can. Run the simulation and watch the active LED as you throw the switch by pressing the space bar.

Figure 41(d) uses digital logic (inverters in this case) to turn ON the LEDs. One advantage of using logic gates is that you do not need to provide 5-V and 0-V levels as the input to the circuit as we did above. You need only to provide a voltage that looks HIGH to the input of the gate to make the gate's output go to 5 V. (It gets its 5-V output voltage from the V_{CC} supply connected to pin 14 of the 7404 IC shown in Figure 39.) In this illustration, the R_7 potentiometer (variable resistor) is set to its top 25% point, which drops the 5-V supply by 25%, equaling approximately 3.75 V. This is definitely a HIGH input (1) to the inverters, making them output a LOW (0), which provides a path for the current to turn ON LED4. The current actually flows through LED4 into the output pin of U2A (pin 2 of the 7404 shown in Figure 39) and then down into ground via the ground pin 7 shown in Figure 39. At the same time, inverter U2C will output a HIGH (1) keeping LED5 OFF. The three voltmeters in the circuit show the voltage levels at various stages.

Turn ON the MultiSIM® simulation and decrease the voltage into the gates by repeatedly pressing the *A* key on your keyboard. Notice that when the voltage drops below half, the LEDs switch states. Increase the voltage back up by repeatedly pressing *Shift-A*. Keep in mind that a HIGH into an inverter produces a LOW output and vice versa.

11 The CMOS Integrated Circuit

Another common IC technology used in digital logic is the **CMOS** (complementary metal oxide semiconductor). CMOS uses a complementary pair of metal oxide semiconductor field-effect transistors (MOSFETs) instead of the bipolar transistors used in TTL chips.

The major advantage of using CMOS is its low power consumption. Because of that, it is commonly used in battery-powered devices such as handheld calculators and digital thermometers. The disadvantage of using CMOS is that generally its switching speed is slower than TTL and it is susceptible to burnout due to electrostatic charges if not handled properly. Figure 42 shows the pin configuration for a 4049 CMOS **hex inverter**.

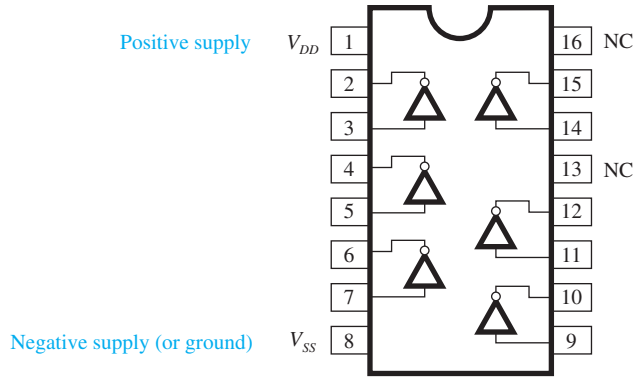
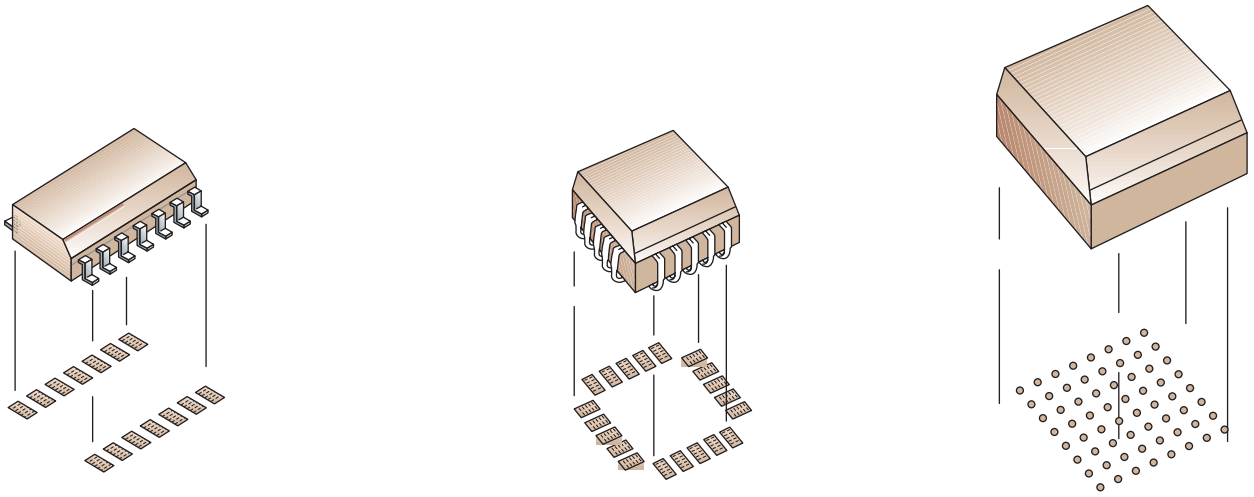
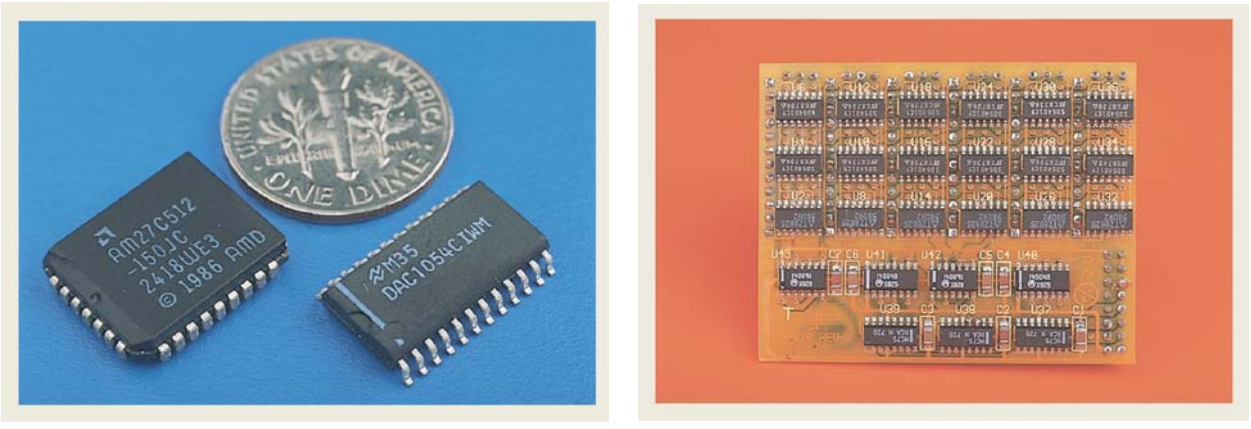


Figure 42 A 4049 CMOS hex inverter pin configuration.



(a) (b) (c)



(d) (e)

Figure 43 Typical surface-mount devices (SMDs) and their footprints: (a) small outline (SO); (b) plastic leaded chip carrier (PLCC); (c) ball grid array (BGA); (d) photograph of actual SMDs; (e) photograph of SMDs mounted on a printed-circuit board.

12 Surface-Mount Devices

The future of modern electronics depends on the ability to manufacture smaller, more dense components and systems. **Surface-mount devices (SMDs)** have fulfilled this need. They have reduced the size of DIP-style logic by as much as 70% and reduced their weight by as much as 90%. To illustrate the size difference, a 7400 IC in the DIP style measures 19.23 mm by 6.48 mm, whereas the equivalent 7400 SMD is only 8.75 mm by 6.20 mm.

SMDs have also significantly lowered the cost of manufacturing printed-circuit boards. This reduction occurs because SMDs are soldered directly to a metalized footprint on the surface of a PC board, whereas holes must be drilled for each leg of a DIP. Also, SMDs can use the faster pick-and-place machines instead of the autoinsertion machines required for "through-hole" mounting of DIP ICs. (Removal of defective SMDs from PC boards is more difficult, however. Special desoldering tools and techniques are required because of the SMD's small size.)

Complete system densities can increase using SMDs because they can be placed closer together and can be mounted to both sides of a printed-circuit board. This also tends to decrease the capacitive and inductive problems that occur in digital systems operating at higher frequencies.

The most popular SMD package styles are the SO (small outline), the PLCC (plastic leaded chip carrier), and the ball grid array (BGA) shown in Figure 43. The SO is a dual-in-line plastic package with leads spaced 0.050 in. apart and bent down and out in a gull-wing format. The PLCC is the most common SMD for ICs requiring a higher pin count (those having more than 28 pins). The PLCC is square, with leads on all four sides. They are bent down and under in a J-bend configuration. They, too, are soldered directly to the metalized footprint on the surface of the circuit board. For even higher pin counts, the BGA uses an array of round solder tabs on the underside of the package. Another version of the grid array is the pin grid array (PGA), which has pins extending from the bottom. It is soldered in holes in a circuit board or placed in a socket for easy removal. Large-scale microprocessors like the Pentium are usually PGA ICs.

The SO package is available for the most popular lower-complexity TTL and CMOS digital logic and analog IC devices. PLCCs, BGAs, and PGAs are available to implement more complex logic, such as microprocessors, microcontrollers, and large memories.

Review Questions

20. In a common-emitter transistor circuit, when V_{out} is 0, R_C should be _____ (small/large), and when V_{out} is 1, R_C should be _____ (small/large).

21. Which transistor in the schematic of the TTL circuit in Figure 37 serves as a variable R_C resistance?

Summary

In this chapter, we have learned that

1. The digital level for 1 is commonly represented by a voltage of 5 V in digital systems. A voltage of 0 V is used for the 0 level.

2. An oscilloscope can be used to observe the rapidly changing voltage-versus-time waveform in digital systems.

3. The frequency of a clock waveform is equal to the reciprocal of the waveform's period.

4. The transmission of binary data in the serial format requires only a single conductor with a ground reference. The parallel format requires several conductors but is much faster than the serial format.

5. Electromechanical relays are capable of forming shorts and opens in circuits requiring high current values but not high speed.

6. Diodes are used in digital circuitry whenever there is a requirement for current to flow in one direction but not in the other.

7. The transistor is the basic building block of the modern digital IC. It can be switched on or off by applying the appropriate voltage at its base connection.

8. TTL and CMOS ICs are formed by integrating thousands of transistors in a single package. They are the most popular ICs used in digital circuitry today.

9. SMD-style ICs are gaining popularity over the through-hole style DIP ICs because of their smaller size and reduced manufacturing costs.

Glossary

Bias: The voltage necessary to cause a semiconductor device to conduct or cut off current flow. A device can be forward or reverse biased, depending on what action is desired.

Chip: The term given to an integrated circuit. It comes from the fact that each integrated circuit comes from a single chip of silicon crystal.

CMOS: Complementary metal oxide semiconductor. A family of integrated circuits used to perform logic functions in digital circuits. The CMOS is noted for its low power consumption but sometimes slow speed.

Cutoff: A term used in transistor switching signifying that the collector-to-emitter junction is turned off or is not allowing current flow.

Diode: A semiconductor device used to allow current flow in one direction but not the other. As an electronic switch, it acts like a short in the forward-biased condition and like an open in the reverse-biased condition.

DIP: Dual-in-line packages. The most common pin layout for integrated circuits. The pins are aligned in two straight lines, one on each side.

Energized Relay Coil: By applying a voltage to the relay coil, a magnetic force is induced within it; this is used to attract the relay contacts away from their resting positions.

Frequency: A measure of the number of cycles or pulses occurring each second. Its unit is the hertz (Hz), and it is the reciprocal of the period.

Hex Inverter: An integrated circuit containing six inverters on a single DIP package.

Integrated Circuit: The fabrication of several semiconductor and electronic devices (transistors, diodes, and resistors) onto a single piece of silicon crystal. Integrated circuits are being used to perform the functions that once required several hundred discrete semiconductors.

Inverter: A logic circuit that changes its input into the opposite logic state at its output (0 to 1 and 1 to 0).

Logic State: A 1 or 0 digital level.

Oscilloscope: An electronic measuring device used in design and troubleshooting to display a waveform of voltage magnitude (*y* axis) versus time (*x* axis).

Parallel: A digital signal representation that uses several lines or channels to transmit binary information. The parallel lines allow for the transmission of an entire multibit number with each clock pulse.

Period: The measurement of time from the beginning of one periodic cycle or clock pulse to the beginning of the next. Its unit is the second(s), and it is the reciprocal of frequency.

Relay: An electric device containing an electromagnetic coil and normally open or normally closed contacts. It is useful because, by supplying a small triggering current to its coil, the contacts will open or close, switching a higher current on or off.

Saturation: A term used in transistor switching that signifies that the collector-to-emitter junction is turned on, or conducting current heavily.

Serial: A digital signal representation that uses one line or channel to transmit binary information. The binary logic states are transmitted 1 bit at a time, with the LSB first.

Surface-Mounted Device: A newer style of integrated circuit, soldered directly to the surface of a printed circuit board. They are much smaller and lighter than the equivalent logic constructed in the DIP through-hole-style logic.

Timing Diagram: A diagram used to display the precise relationship between two or more digital waveforms as they vary relative to time.

Totem Pole: The term used to describe the output stage of most TTL integrated circuits. The totem-pole stage consists of one transistor in series with another, configured in such a way that when one transistor is saturated, the other is cut off.

Transistor: A semiconductor device that can be used as an electronic switch in digital circuitry. By applying an appropriate voltage at the base, the collector-to-emitter junction will act like an open or a shorted switch.

TTL: Transistor–transistor logic. The most common integrated circuit used in digital electronics today. A large family of different TTL integrated circuits is used to perform all the logic functions necessary in a complete digital system.

Problems

Sections 1 and 2

1. Determine the period of a clock waveform whose frequency is

(a) 2 MHz **(b)** 500 kHz **(c)** 4.27 MHz **(d)** 17 MHz

Determine the frequency of a clock waveform whose period is

(e) 2 μs **(f)** 100 μs **(g)** 0.75 ms **(h)** 1.5 μs

Sections 3 and 4

2. Sketch the serial and parallel representations (similar to Figure 10) of the following numbers, and calculate how long they will take to transmit (clock frequency = 2 MHz).

(a) 99_{10} **(b)** 124_{10}

3. **(a)** How long will it take to transmit the number 33_{10} in serial if the clock frequency is 3.7 MHz? (Transmit the number as an 8-bit binary number.)

(b) Is the serial line HIGH or LOW at 1.21 μs?

4. **(a)** How long will it take to transmit the three ASCII-coded characters \$14 in 8-bit parallel if the clock frequency is 8 MHz?

(b) Repeat for \$78.18 at 4.17 MHz.

Sections 5 and 6

C **5.** Draw the timing diagram for V_{out1}, V_{out2}, and V_{out3} in Figure P5.

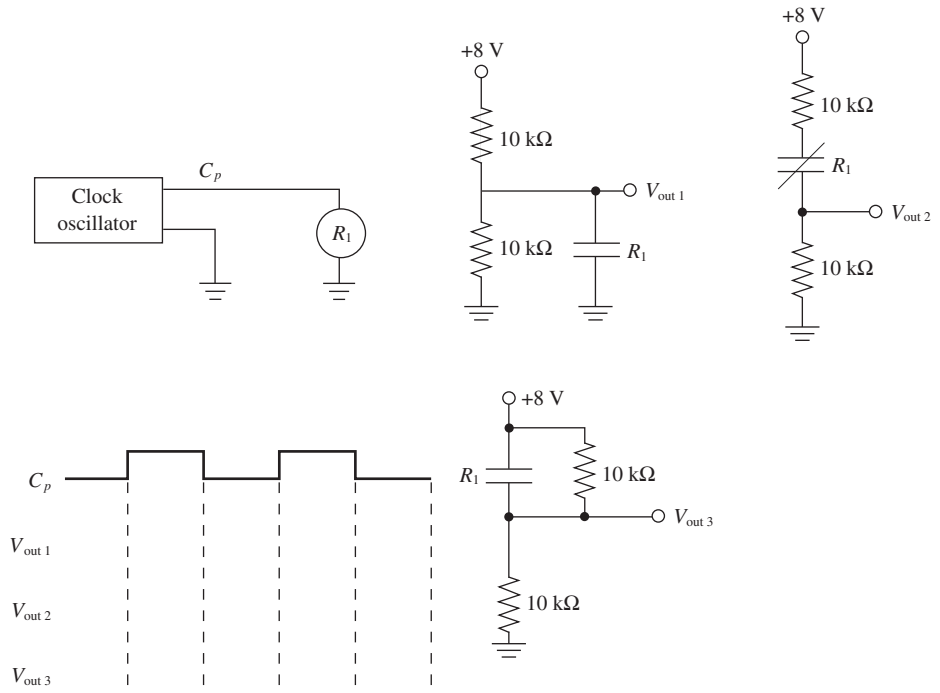


Figure P5

Section 7

6. Determine if the diodes in Figure P6 are reverse or forward biased.

C **7.** Determine V_1, V_2, V_3, V_4, V_5, V_6, and V_7 in the circuits of Figure P6.

8. In Figure P6, if the cathode of any one of the diodes D_8, D_9, or D_{10} is connected to 0 V instead of +5 V, what happens to V_6?

9. In Figure P6, if the anode of any of the diodes D_{11}, D_{12}, or D_{13} is connected to +5 V instead of 0 V, what happens to V_7?

Section 8

10. Find V_{out1} and V_{out2} for the circuits of Figure P10.

11. Sketch the waveforms at V_{out} in the circuit of Figure 32 using $R_C = 6\,k\Omega$.

Section 9

12. To use a common-emitter transistor circuit as an inverter, the input signal is connected to the _____ (base, collector, or emitter) and the output signal is taken from the _____ (base, collector, or emitter).

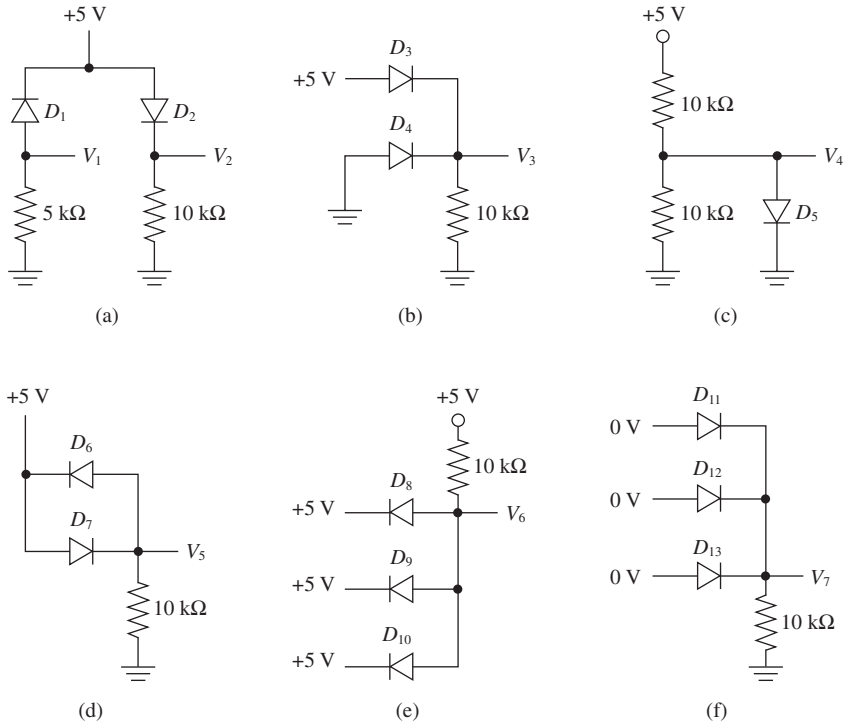


Figure P6

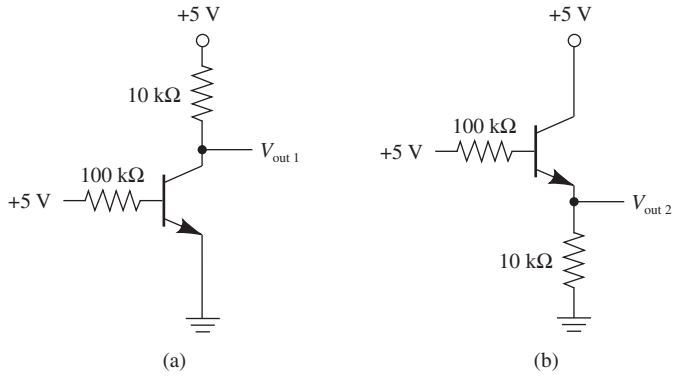


Figure P10

C **13.** Determine V_{out} for the common-emitter transistor inverter circuit of Figure 35 using $V_{in} = 0\,V$, $R_B = 1\,M\Omega$, $R_C = 330\,\Omega$, and $R_{load} = 1\,M\Omega$.

C **14.** If the load resistor (R_{load}) used in Problem 13 is changed to 470 Ω, describe what happens to V_{out}.

C **15.** In the circuit of Figure 35 with $V_{in} = 0\,V$, V_{out} will be almost 5 V as long as R_{load} is much greater than R_C. Why not make R_C very small to ensure that the circuit will work for all values of R_{load}?

C **16.** In Figure 35, if $R_C = 100\,\Omega$, find the collector current when $V_{in} = +5\,V$.

C **17.** Describe how the totem-pole output arrangement in a TTL circuit overcomes the problems faced when using the older common-emitter transistor inverter circuit.

18. Sketch the waveform at C_p and V_{out} for Figure P18.

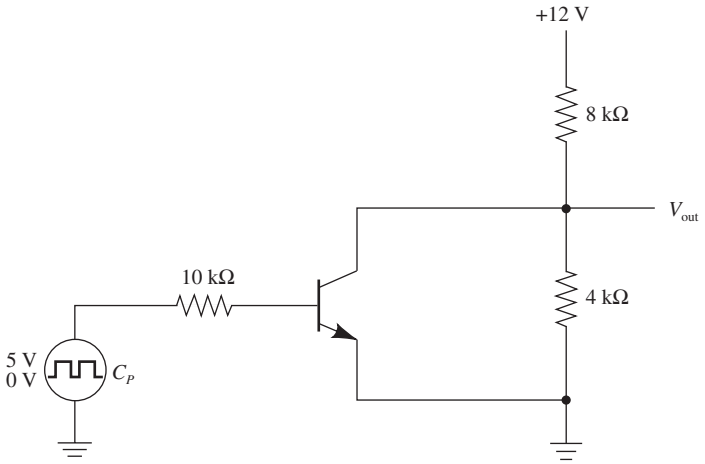
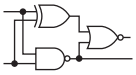


Figure P18



Schematic Interpretation Problems

See Appendix: Schematic Diagrams for Chapter-End Problems for the schematic diagrams.

S **19.** Y1 in the 4096/4196 control card schematic sheet 1 is a crystal used to generate a very specific frequency.

(a) What is its rated frequency?

(b) What time period does that create?

S **20.** Repeat Problem 19 for the crystal *X*1 in the HC11D0 master board schematic.

S **21.** The circuit on the HC11D0 schematic is capable of parallel as well as serial communication via connectors P_3 and P_2. Which is parallel, and which is serial? (*Hint:* TX stands for transmit, RX stands for receive.)

S **22.** Is diode D_1 of the HC11D0 schematic forward or reverse biased? (*Hint:* $V_{CC} = 5$ V.)

S **23.** The transistor Q_1 in the HC11D0 schematic is turned ON and OFF by the level of pin 2 on U3:A. At what level must pin 2 be to turn Q_1 ON, and what will happen to the level on the line labeled RESET B when that happens?

MultiSIM® Exercises

E1. Load the circuit file for **Section 3.** Read the instructions in the *Description* window.

(a) Determine the three ASCII characters that are transmitted in serial.

(b) Determine the number of serial bits transmitted.

E2. Load the circuit file for **Section 4.** Read the instructions in the *Description* window.

(a) Determine the three ASCII characters that are transmitted in parallel.

(b) How many clock pulses did it take to complete the transmission?

E3. Load the circuit file for **Section 6a.** Read the instructions in the *Description* window. The normally open relay contacts are used to create a short across the lower 5-kΩ resistor when C_p goes HIGH.

(a) Measure the voltage levels of C_p and V_{out3} with the oscilloscope. Note the relationship between the two waveforms.

(b) Change the upper resistor to 2 kΩ and the lower resistor to 8 kΩ. Predict the new voltage levels, then measure them with the oscilloscope.

(c) If the normally closed relay contacts were used, what change would you expect in the V_{out3} waveform? Try it.

C **E4.** Load the circuit file for **Section 6b.** Read the instructions in the *Description* window. The normally closed relay contacts are used to create an open between the two resistors when C_p goes HIGH.

(a) Measure the voltage levels of C_p and V_{out4} with the oscilloscope. Note the relationship between the two waveforms. (The top waveform is V_{out4}).

(b) Change the upper resistor to 2 kΩ and the lower resistor to 8 kΩ. Predict the new voltage levels, then measure them with the oscilloscope.

(c) If the normally open relay contacts were used instead of the normally closed contacts, what change would you expect in the V_{out4} waveform? Try it.

E5. Load the circuit file for **Section 7.** Read the instructions in the *Description* window. Before turning the power switch ON, predict the voltage V1, V2, V3, and V4.

(a) Turn the switch ON and check your answers.

(b) Reverse all six diodes, and predict what V1, V2, V3, and V4 will become. Turn the power switch ON, and check your answers.

E6. Load the circuit file for **Section 8.** Read the instructions in the *Description* window.

(a) Measure the voltage levels of C_p and V_{out} with the oscilloscope. Note the relationship between the two waveforms.

(b) Change the upper resistor to 2 kΩ and the lower resistor to 8 kΩ. Predict the new voltage levels, then measure them with the oscilloscope.

Answers to Review Questions

1. *x* axis, time; *y* axis, voltage

2. The clock frequency is the reciprocal of the clock period.

3. 125 ns

4. 20 MHz

5. 5 MHz

6. 385 ps

7. Frequency = 357 ns

8. 58.5 kHz

9. It is faster.

10. Parallel

11. 4.80 μs

12. 600 ns

13. The relay coil is energized by placing a voltage at its terminals. The contacts will either make a connection (NO relay) or break a connection (NC relay) when the coil is energized.

14. An NO relay makes connection when energized. An NC relay breaks connection when energized.

15. Positive

16. Approximately 0.7 V

17. Emitter, base, collector

18. Positive

19. Short

20. Large, small

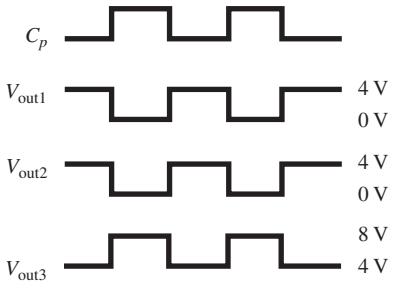
21. Q_4

Answers to Odd-Numbered Problems

1. **(a)** 0.5 μs **(b)** 2 μs **(c)** 0.234 μs
(d) 58.8 ns **(e)** 500 kHz **(f)** 10 kHz
(g) 1.33 kHz **(h)** 0.667 MHz

3. **(a)** 2.16 μs **(b)** LOW

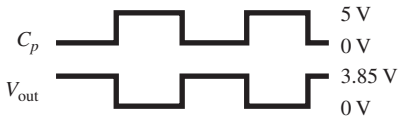
5.



7. $V_1 = 0$ V
$V_2 = 4.3$ V
$V_3 = 4.3$ V
$V_4 = 0.7$ V
$V_5 = 4.3$ V
$V_6 = 5.0$ V
$V_7 = 0$ V

9. That diode will conduct raising V_7 to 4.3 V ("OR").

11.



13. $V_{out} = 4.998$ V

15. Because, when the transistor is turned on (saturated), the collector current will be excessive ($I_C = 5\text{ V}/R_C$)

17. The totem-pole output replaces R_C with a transistor that acts like a variable resistor. The transistor prevents excessive collector current when it is cut off and provides a high-level output when turned on.

19. **(a)** 8.0 MHz **(b)** 125 ns

21. P3 parallel, P2 serial

23. A HIGH on pin 2 will turn Q1 on, making RESET_B approximately zero.

E1. **(a)** Let **(b)** 24

E3. **(a)** $C_p = 5\text{ V}/0\text{ V}$, $V_{out3} = 0\text{ V}/5\text{ V}$, inverse of each other
(b) $C_p = 5\text{ V}/0\text{ V}$, $V_{out3} = 0\text{ V}/8\text{ V}$
(c) C_p and V_{out3} are in phase.

E5. **(a)** V1 = 4.3 V, V2 = 0 V, V3 = 4.3 V, V4 = 0.7 V **(b)** V1 = 0 V, V2 = 4.3 V, V3 = 0 V, V4 = 5.0 V (Both diodes are reverse biased.)

Basic Logic Gates

OUTLINE

OBJECTIVES

Upon completion of this chapter, you should be able to do the following:

- Describe the operation and use of AND gates and OR gates.
- Construct truth tables for two-, three-, and four-input AND and OR gates.
- Draw timing diagrams for AND and OR gates.
- Describe the operation, using timing analysis, of an ENABLE function.
- Sketch the external connections to integrated-circuit chips to implement AND and OR logic circuits.
- Explain how to use a logic pulser and a logic probe to troubleshoot digital integrated circuits.
- Describe the operation and use of inverter, NAND, and NOR gates.
- Construct truth tables for two-, three-, and four-input NAND and NOR gates.
- Draw timing diagrams for inverter, NAND, and NOR gates.
- Use the outputs of a Johnson shift counter to generate specialized waveforms utilizing various combinations of the five basic gates.
- Develop a comparison of the Boolean equations and truth tables for the five basic gates.

The companion website for this text is www.pearsonhighered.com/kleitz

From Chapter 3 of *Digital Electronics: A Practical Approach with VHDL*, Ninth Edition.

INTRODUCTION

Logic **gates** are the basic building blocks for forming digital electronic circuitry. A logic gate has one output terminal and one or more input terminals. Its output will be HIGH (1) or LOW (0) depending on the digital level(s) at the input terminal(s). Through the use of logic gates, we can design digital systems that will evaluate digital input levels and produce a specific output response based on that particular logic circuit design. The five basic logic gates are the AND, OR, NAND, NOR, and inverter.

1 The AND Gate

Let's start by looking at the two-input AND gate whose schematic symbol is shown in Figure 1. The operation of the AND gate is simple and is defined as follows: *The output, X, is HIGH if input A AND input B are both HIGH.* In other words, if $A = 1$ *AND* $B = 1$, then $X = 1$. If either A or B or both are LOW, the output will be LOW.

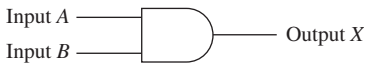


Figure 1 Two-input AND gate symbol.

The best way to illustrate how the output level of a gate responds to all the possible input-level combinations is with a **truth table**. Table 1 is a truth table for a two-input AND gate. On the left side of the truth table, all possible input-level combinations are listed, and on the right side, the resultant output is listed.

TABLE 1 Truth Table for a Two-Input AND Gate

Inputs		Output
A	***B***	***X = AB***
0	0	0
0	1	0
1	0	0
1	1	1

From the truth table, we can see that the output at X is HIGH *only* when *both A* AND B are HIGH. If this AND gate is a TTL integrated circuit, HIGH means +5 V and LOW means 0 V (i.e., 1 is defined as +5 V and 0 is defined as 0 V).

One example of how an AND gate might be used is in a bank burglar alarm system. The output of the AND gate will go HIGH to turn on the alarm if the alarm activation key is in the ON position *AND* the front door is opened. This setup is illustrated in Figure 2(a). Figure 2(b) shows the result for every combination of Key (K) and Door (D).

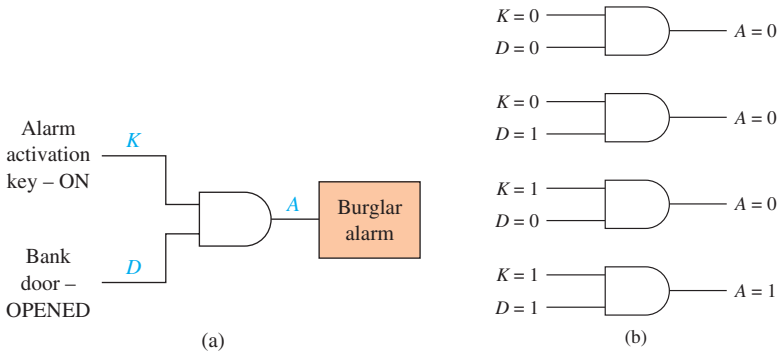


Figure 2 AND gate: (a) Used to activate a burglar alarm; (b) all combinations of key ON (K) and door OPEN (D).

Another way to illustrate the operation of an AND gate is by use of a series electric circuit. In Figure 3, using manual and transistor switches, the output at X is HIGH if *both* switches *A AND B* are HIGH (1).

Figure 3 also shows what is known as the **Boolean equation** for the AND function, $X = A$ and B, which can be thought of as *X equals 1 if A AND B both equal 1.*

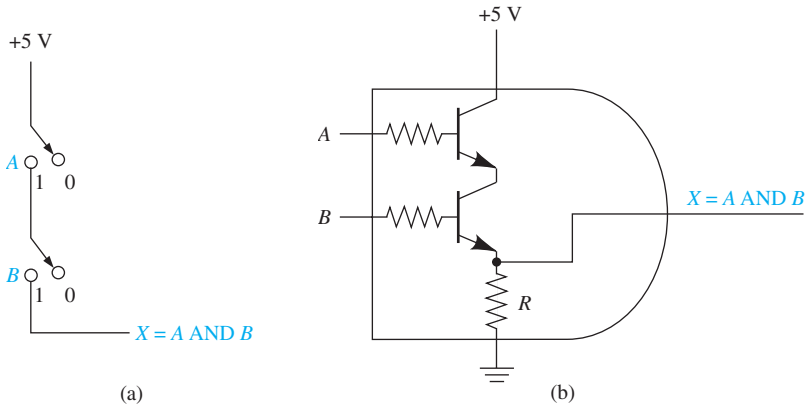


Figure 3 Electrical analogy for an AND gate: (a) using manual switches; (b) using transistor switches.

The Boolean equation for the AND function can more simply be written as $X = A \cdot B$ or just $X = AB$ (which is read as "X equals A AND B"). Boolean equations will be used in this text to depict algebraically the operation of a logic gate or a combination of logic gates.

AND gates can have more than two inputs. Figure 4 shows a four-input, a three-input, and an eight-input AND gate. The truth table for an AND gate with four inputs

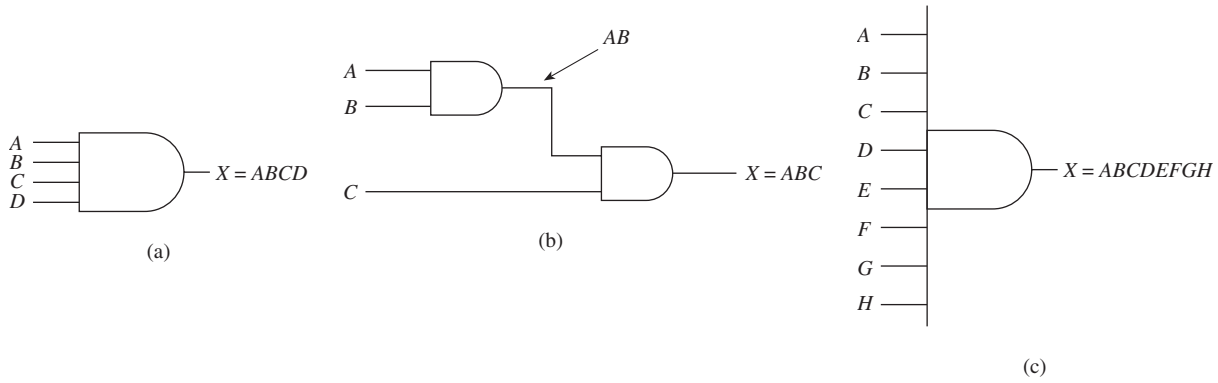


Figure 4 Multiple-input AND gate symbols: (a) 4-input; (b) 3-input formed with two 2-input gates; (c) 8-input.

is shown in Table 2. To determine the total number of different combinations to be listed in the truth table, use the equation

$$\text{number of combinations} = 2^N, \quad \text{where } N = \text{number of inputs} \tag{1}$$

Therefore, in the case of a four-input AND gate, the number of possible input combinations is $2^4 = 16$.

When building the truth table, be sure to list all 16 *different* combinations of input levels. One easy way to ensure that you do not inadvertently overlook a combination of these variables or duplicate a combination is to list the inputs in the order of a binary counter (0000, 0001, 0010, . . . , 1111). Also notice in Table 2 that the A column lists eight 0s, then eight 1s; the B column lists four 0s, four 1s, four 0s, four 1s; the C column lists two 0s, two 1s, two 0s, two 1s, and so on; and the D column lists one 0, one 1, one 0, one 1, and so on.

TABLE 2 Truth Table for a Four-Input AND Gate

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

The output at *X* is HIGH only if *all* inputs are HIGH.

Common Misconception

When you build a truth table, you might mistakenly omit certain input combinations if you don't set the variables up as a binary counter.

2 The OR Gate

The OR gate also has two or more inputs and a single output. The symbol for a two-input OR gate is shown in Figure 5. The operation of the two-input OR gate is defined as follows: *The output at X will be HIGH whenever input A OR input B is HIGH or both are HIGH.* As a Boolean equation, this can be written $X = A + B$ (which is read as "*X* equals *A* OR *B*"). Notice the use of the $+$ symbol to represent the OR function.

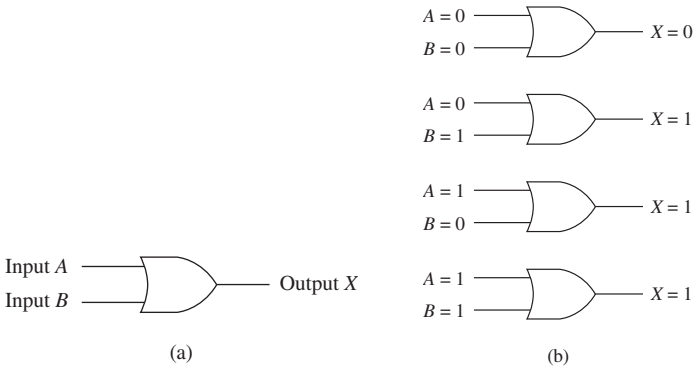


Figure 5 Two-input OR gate: (a) symbol; (b) all input combinations.

The truth table for a two-input OR gate is shown in Table 3.

TABLE 3 Truth Table for a Two-Input OR Gate

Inputs		Output
A	*B*	$X = A + B$
0	0	0
0	1	1
1	0	1
1	1	1

From the truth table you can see that X is 1 whenever A *OR* B is 1 or if *both* A and B are 1. Using manual or transistor switches in an electric circuit, as shown in Figure 6, we can observe the electrical analogy to an OR gate. From the figure, we see that the output at X will be 1 if A *or* B, or *both*, are HIGH (1).

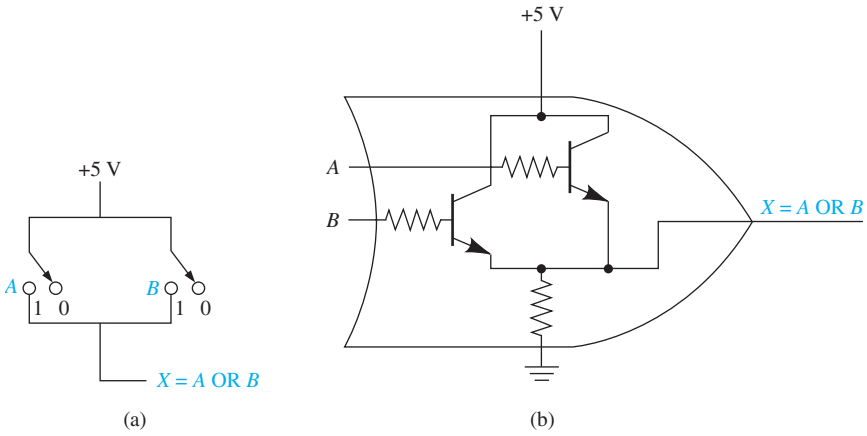


Figure 6 Electrical analogy for an OR gate: (a) using manual switches; (b) using transistor switches.

OR gates can also have more than two inputs. Figure 7 shows three-input OR gates and Figure 8 shows an eight-input OR gate. The truth table for the three-input OR gate will have eight entries ($2^3 = 8$), and the eight-input OR gate will have 256 entries ($2^8 = 256$).

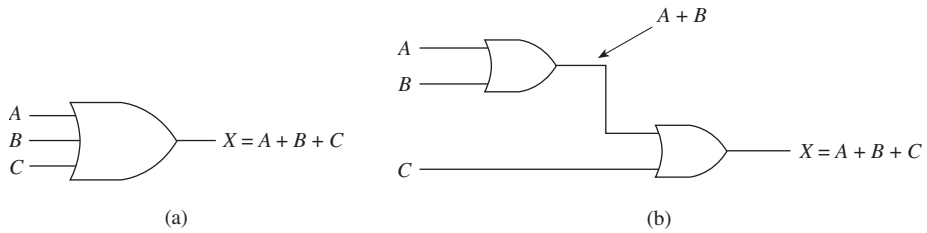


Figure 7 Three-input OR gate: (a) symbol; (b) three inputs formed with two 2-input gates.

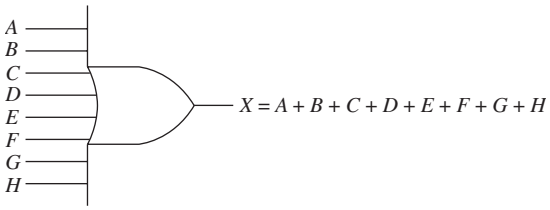


Figure 8 Eight-input OR gate symbol.

Let's build a truth table for the three-input OR gate.

The truth table of Table 4 is built by first using Equation 1 to determine that there will be eight entries, then listing the eight combinations of inputs in the order of a binary counter (000 to 111), and then filling in the output column (X) by realizing that X will always be HIGH as long as at least one of the inputs is HIGH. When you look at the completed truth table, you can see that the only time the output is LOW is when *all* the inputs are LOW.

TABLE 4 Truth Table for a Three-Input OR Gate

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

The output at *X* is HIGH if *any* input is HIGH.

EXAMPLE 1

Determine the output at *U*, *V*, *W*, *X*, *Y*, and *Z* in Figure 9.

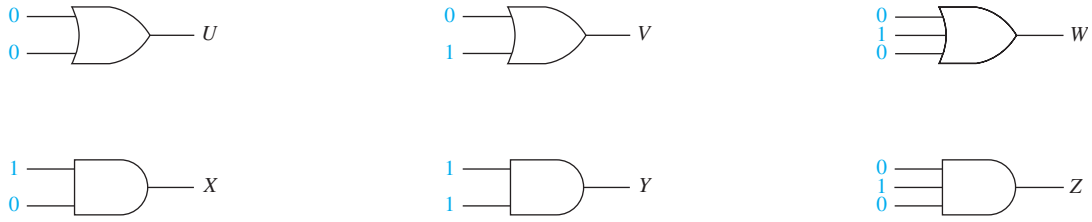


Figure 9 Basic AND and OR gate operation.

Solution:

$U = 0$ (0 OR 0 = 0)
$V = 1$ (0 OR 1 = 1)
$W = 1$ (0 OR 1 OR 0 = 1)
$X = 0$ (1 AND 0 = 0)
$Y = 1$ (1 AND 1 = 1)
$Z = 0$ (0 AND 1 AND 0 = 0)

Review Questions

1. All inputs to an AND gate must be HIGH for it to output a HIGH. True or false?

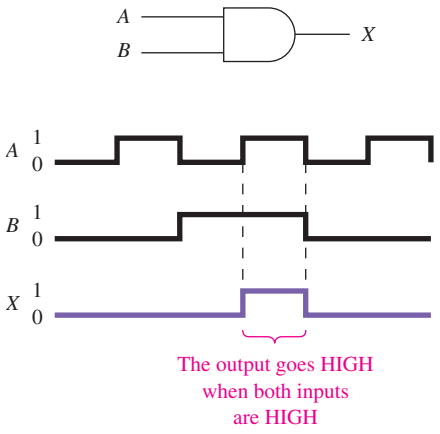
2. What is the purpose of a truth table?

3. What is the purpose of a Boolean equation?

4. What input conditions must be satisfied for the output of an OR gate to be LOW?

3 Timing Analysis

Another useful means of analyzing the output response of a gate to varying input-level changes is by means of a *timing diagram.* A timing diagram is used to illustrate graphically how the output levels change in response to input-level changes.



(a)

(b)

Figure 10 Timing analysis of an AND gate: (a) waveform sketch; (b) actual logic analyzer display.

The timing diagram in Figure 10 shows the two input waveforms (*A* and *B*) that are applied to a two-input AND gate and the *X* output that results from the AND operation. (For TTL and most CMOS logic gates, 1 = +5 V and 0 = 0 V.) As you can see, timing analysis is very useful for visually illustrating the level at the output for varying input-level changes.

Timing waveforms can be observed on an *oscilloscope* or a *logic analyzer*. A dual-trace oscilloscope can display *two* voltage-versus-time waveforms on the same *x* axis. That is ideal for comparing the relationship of one waveform relative to another. The other timing analysis tool is the logic analyzer. Among other things, it can display up to 16 voltage-versus-time waveforms on the same *x* axis (see Figure 10[b]). It can also display the levels of multiple digital signals in a *state table*, which lists the binary levels of all the waveforms, at predefined intervals, in binary, hexadecimal, or octal. Timing analysis of 8 or 16 channels concurrently is very important when analyzing advanced digital and microprocessor systems in which the interrelationship of several digital signals is critical for proper circuit operation.

AND-GATE SIMULATION

The MultiSIM® analysis of the same two-input AND gate circuit is shown in Figure 11. The Four-Channel Oscilloscope is chosen because we can observe both the *A* and *B* inputs and the *X* output simultaneously. Different colors are chosen for the three signals so that they can be distinguished on the oscilloscope display. Also, the *Y* position of the *A* input and *X* output are adjusted so that the waveforms don't overlay on each other. The Word Generator is set up as an up counter to create the combination of waveforms required for *A* and *B*. (Choose ***Set…***, then ***UP Counter***, ***Display • Hex***.)

MultiSIM exercise: Use MultiSIM to open the file *fig3_11* from the text website. Run the simulation to create the waveforms shown in Figure 11. Make the following changes to the gate (U1) and rerun the simulation:

(a) Change U1 to a two-input OR gate (OR2).

(b) Change U1 to a three-input AND gate (AND3) and add the third input waveform.

(c) Change U1 to a three-input OR gate (OR3) and add the third input waveform.

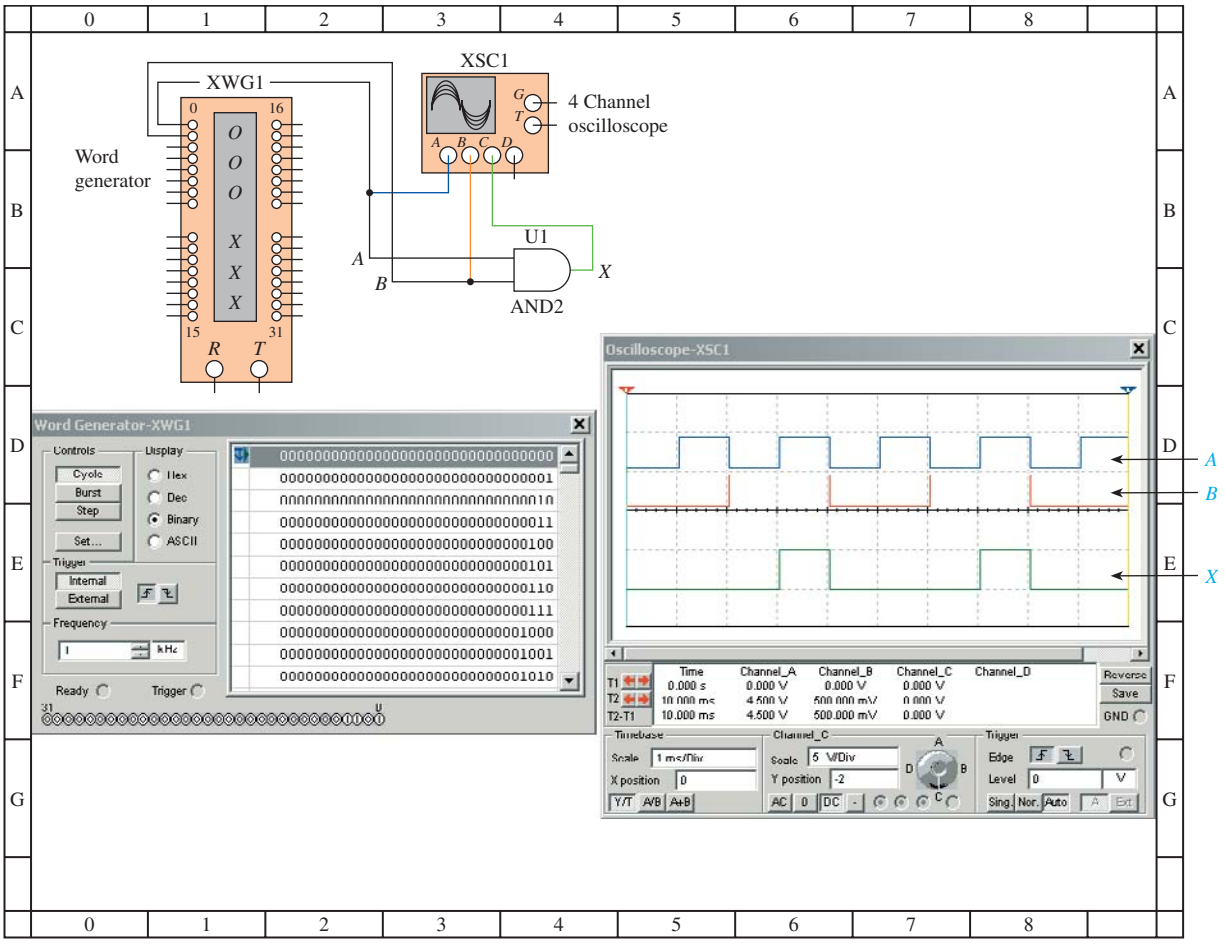
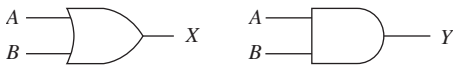


Figure 11 Using the MultiSIM® Four-Channel Oscilloscope to monitor the input and output waveforms of a two-input AND gate.

EXAMPLE 2

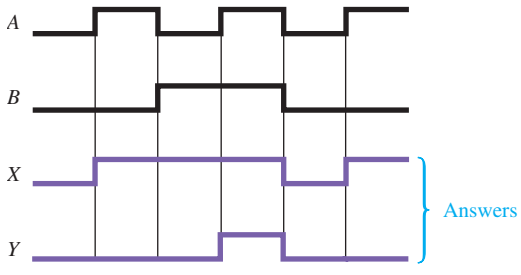
Sketch the output waveform at *X* and *Y* for the two-input OR gate and AND gate shown in Figure 12(a), with the given *A* and *B* input waveforms in Figure 12(b).



(a)

Figure 12(a)

Solution:



(b)

Figure 12(b) Solution to Example 2.

EXAMPLE 3

Sketch the output waveform at *X* for the three-input AND gate and OR gate shown in Figure 13, with the given *A*, *B*, and *C* input waveforms in Figure 14.

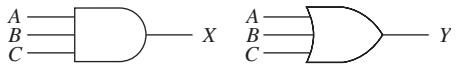


Figure 13

Solution:

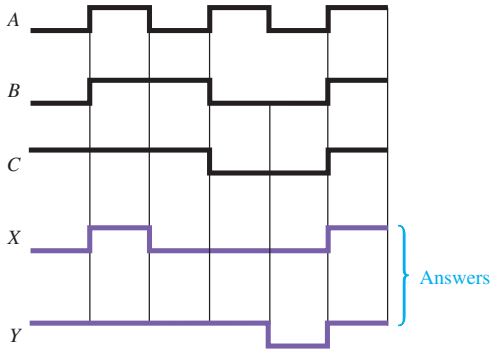


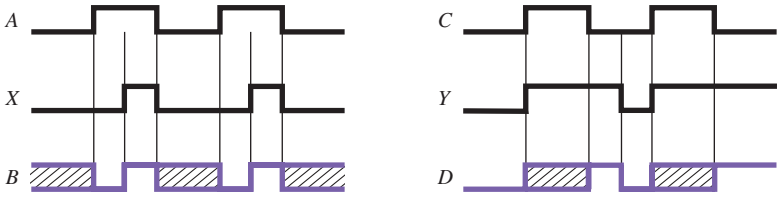
Figure 14 Solution to Example 3.

EXAMPLE 4

The input waveform at *A* and the output waveform at *X* are given for the AND gate in Figure 15(a). Sketch the input waveform that is required at *B* to produce the output at *X* in Figure 15(b). Repeat for the OR gate.

Figure 15(a)

Solution:



= Don't care (*B* can be HIGH or LOW to get the same output at *X*.)

Figure 15(b) Solution to Example 4.

4 Enable and Disable Functions

AND and OR gates can be used to **enable** or **disable** a waveform from being transmitted from one point to another. For example, let's say that you wanted a 1-MHz clock oscillator to transmit only four pulses to some receiving device. You would want to *enable* four clock pulses to be transmitted and then *disable* the transmission from then on.

The clock frequency of 1 MHz converts to 1 μs (1/1 MHz) for each clock period. Therefore, to transmit four clock pulses, we have to provide an *enable* signal for 4 μs. Figure 16 shows the circuit and waveforms to *enable* four clock pulses. For the HIGH clock pulses to get through the AND gate to point *X*, the second input to the AND gate (enable signal input) must be HIGH; otherwise, the output of the AND gate will be LOW. Therefore, when the enable signal is HIGH for 4 μs, four clock pulses pass through the AND gate. When the enable signal goes LOW, the AND gate *disables* any further clock pulses from reaching the receiving device.

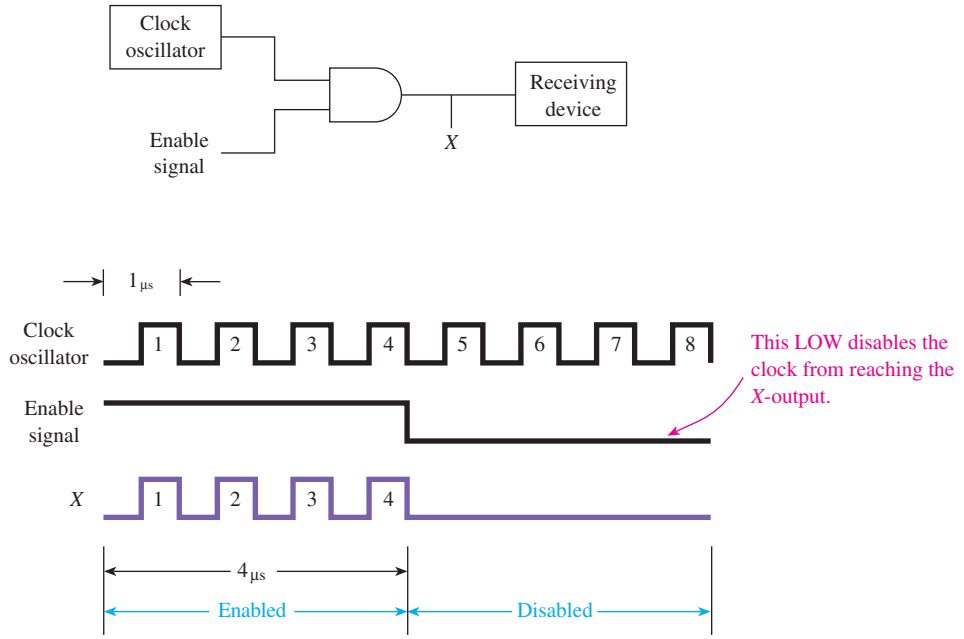


Figure 16 Using an AND gate to enable/disable a clock oscillator.

An OR gate can also be used to disable a function. The difference is that the enable signal input is made HIGH to disable, and that the output of the OR gate goes HIGH when it is disabled, as shown in Figure 17.

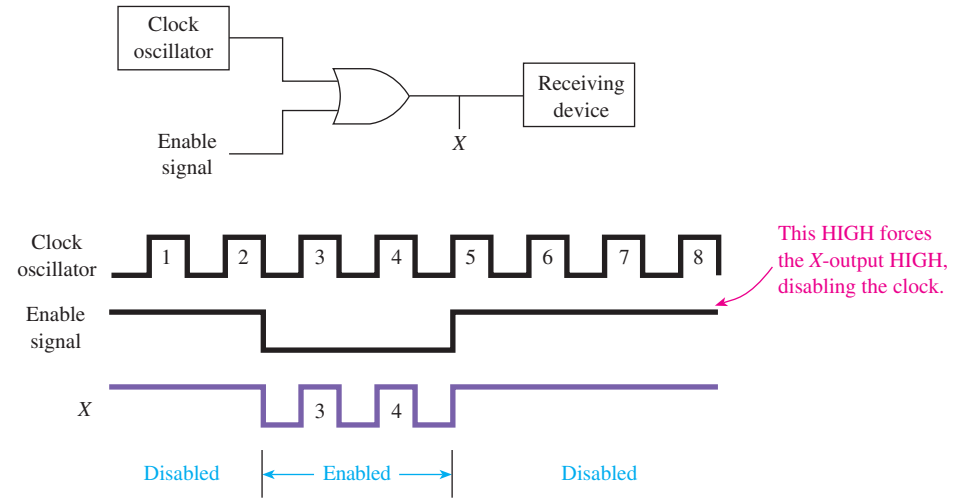


Figure 17 Using an OR gate to enable/disable a clock oscillator.

ENABLE AND DISABLE SIMULATION

Figure 18 shows a MultiSIM simulation of enabling and disabling functions. The word generator is used to create the enable signal (En) and the clock oscillator (Cp). Notice that whenever En is HIGH, the AND gate passes Cp to the output at *X*. When En is LOW, the OR gate passes Cp to the output at *Y*, otherwise *Y* is HIGH.

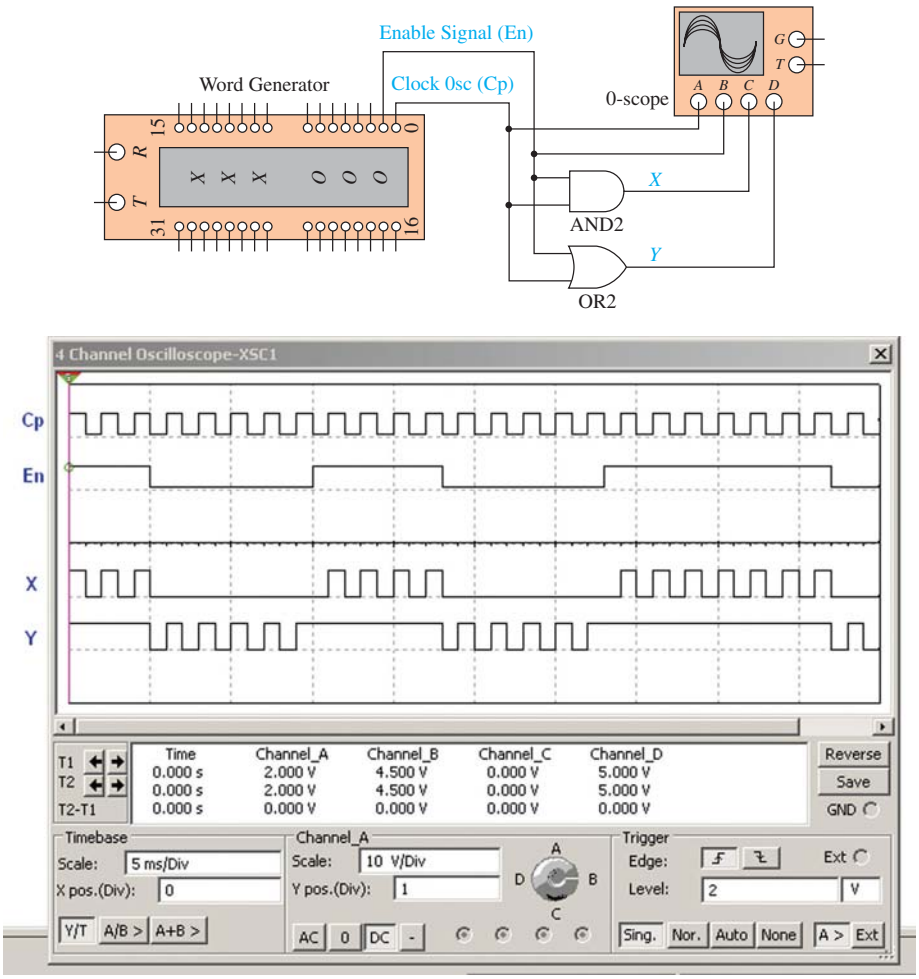


Figure 18 A MultiSIM simulation of enable/disable functions.

Review Questions

5. Describe the purpose of a *timing diagram.*

6. Under what circumstances would diagonal "don't care" hash marks be used in a timing diagram?

7. A _____ (HIGH/LOW) level is required at the input to an AND gate to *enable* the signal at the other input to pass to the output.

5 Using IC Logic Gates

AND and OR gates are available as ICs. The IC pin layout, logic gate type, and technical specifications are all contained in the logic data manual supplied by the manufacturer of the IC. For example, referring to a TTL or a CMOS logic data manual, we can see that there are several AND and OR gate ICs. To list just a few:

1. The 7408 (74LS08, 74HC08) is a quad two-input AND gate.
2. The 7411 (74LS11, 74HC11) is a triple three-input AND gate.
3. The 7421 (74LS21, 74HC21) is a dual four-input AND gate.
4. The 7432 (74LS32, 74HC32) is a quad two-input OR gate.

In each case, the letters *LS* stand for the Low-Power Schottky TTL family and the letters *HC* stand for the High-Speed CMOS family. For example, the basic part number 7408 refers to an AND gate IC with four (quad) internal AND gates each having two inputs. The most common TTL version is the 74LS08, and the most common CMOS version is the 74HC08. They both have exactly the same pin layout and function. In this text, the basic part number is usually given, and it depends on the particular application as to which family is used to implement the design based on IC availability and speed and power considerations.

Besides the family designation (LS, HC, etc.), most ICs will have a prefix that specifies the manufacturer. Two examples of this are SN for Texas Instruments—SN7400 and DM for Fairchild—DM7400. Also, a suffix is added to the end of the part number to specify the package style. Two examples of this are N for Plastic Dual-In-Line Package (P-DIP)—SN7400N and M for Small-Outline Integrated Circuit (SOIC)—DM7400M.

Let's look in more detail at one of these ICs, the 7408 (see Figure 19). The 7408 is a 14-pin DIP IC. The power supply connections are made to pins 7 and 14. This supplies the operating voltage for all four AND gates on the IC. Pin 1 is identified by a small indented circle next to it or by a notch cut out between pin 1 and 14 (see Figure 19). Let's make the external connections to the IC to form a clock oscillator enable circuit similar to Figure 17.

In Figure 20, the first AND gate in the IC was used and the other three are ignored. The IC is powered by connecting pin 14 to the positive power supply and pin 7 to ground. The other connections are made by following the original design from

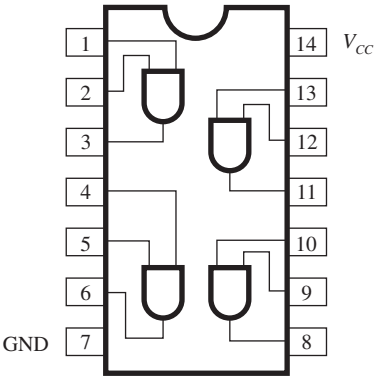
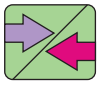


Figure 19 The 7408 quad two-input AND gate IC pin configuration.

Helpful Hint

For example, the basic part number 7408 would become SN74LS08N if it were manufactured by Texas Instruments (SN) as a Low-Power Schottky family (LS) in a plastic DIP (N) package.



Common Misconception

Students often think that a gate output receives its HIGH or LOW voltage level from its input pin. You need to be reminded that each gate has its own totem-pole output arrangement and receives its voltage from V_{CC} or ground.

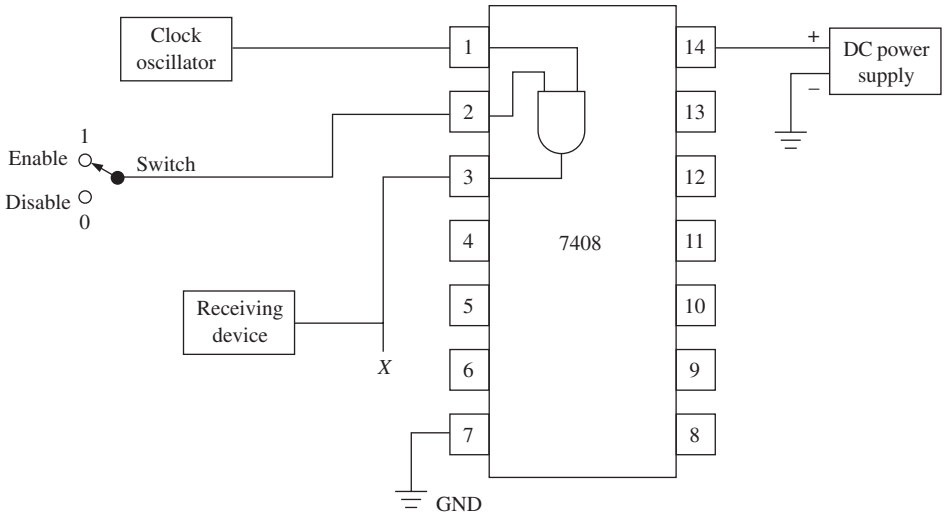


Figure 20 Using the 7408 TTL IC in the clock enable circuit of Figure 17.

Figure 17. The clock oscillator signal passes on to the receiving device when the switch is in the *enable* (1) position, and it stops when in the *disable* (0) position.

The pin configurations for some other logic gates are shown in Figure 21.

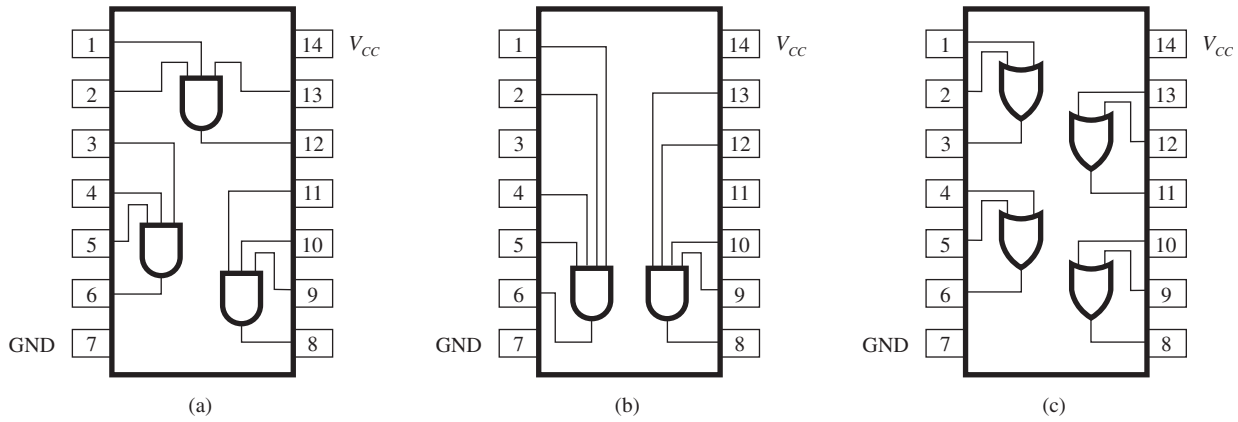


Figure 21 Pin configurations for other popular TTL and CMOS AND and OR gate ICs: (a) 7411 (74HC11); (b) 7421 (74HC21); (c) 7432 (74HC32).

6 Introduction to Troubleshooting Techniques

Like any other electronic device, ICs and digital electronic circuits can go bad. **Troubleshooting** is the term given to the procedure used to find the **fault**, or *trouble*, in the circuits.

To be a good troubleshooter, you must first *understand the theory and operation* of the circuit, devices, and ICs that are suspected to be bad. If you understand how a particular IC is *supposed* to operate, it is a simple task to put the IC through a test or to exercise its functions to see if it operates as you expect.

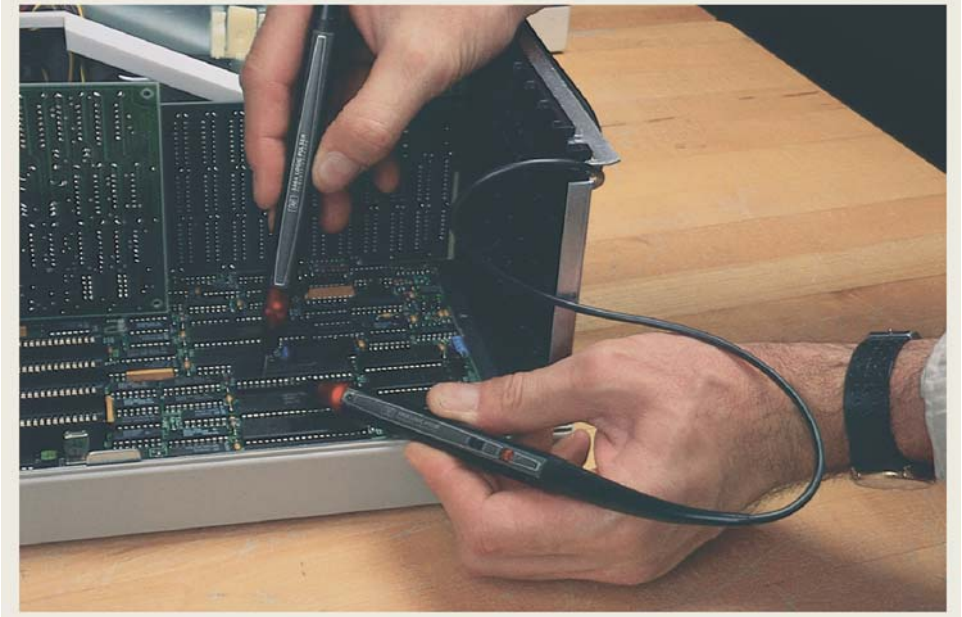


Figure 22 Logic pulser and logic probe.

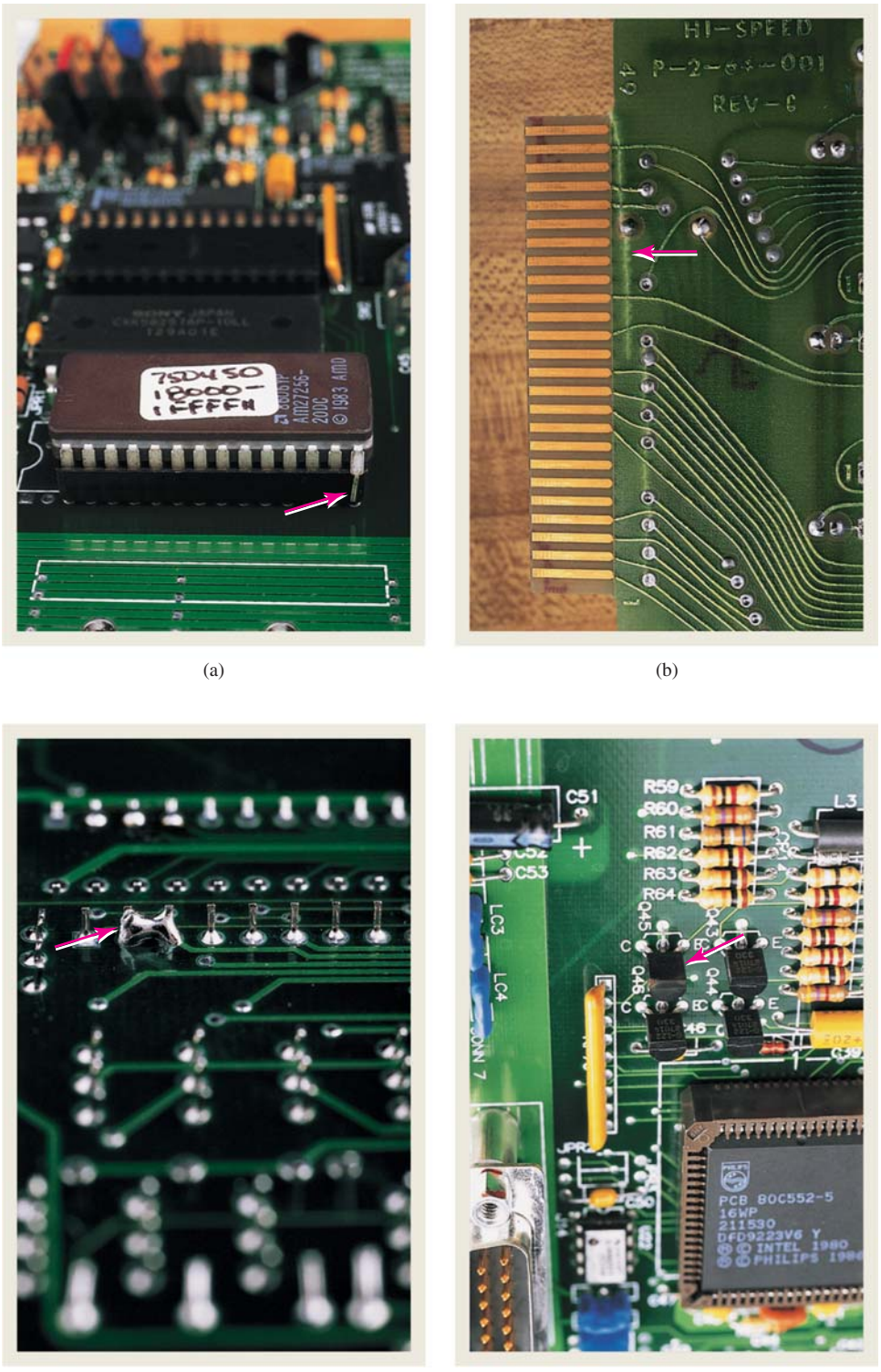
There are two simple tools that we will start with to test the ICs and digital circuits. They are the logic pulser and logic probe (see Figure 22). The **logic probe** has a metal tip that is placed on the IC pin, printed-circuit board trace, or device lead that you want to test. It also has an indicator lamp that glows, telling you the digital level at that point. If the level is HIGH (1), the lamp glows brightly. If the level is LOW (0), the lamp goes out. If the level is **floating** (open circuit, neither HIGH nor LOW), the lamp is dimly lit. Table 5 summarizes the states of the logic probe.

TABLE 5 Logic Probe States

Logic Level	Indicator Lamp
HIGH (1)	On
LOW (0)	Off
Float	Dim

The **logic pulser** is used to provide digital pulses to a circuit being tested. By applying a pulse to a circuit and simultaneously observing a logic probe, you can tell if the pulse signal is getting through the IC or device as you would expect. As you become more and more experienced at troubleshooting, you will find that most IC and device faults are due to an open or short at the input or output terminals.

Figure 23 shows four common problems that you will find on printed-circuit boards that will cause opens or shorts. Figure 23(a) shows an IC that was inserted into its socket carelessly, causing pin 14 to miss its hole and act like an open. In Figure 23(b), the printed-circuit board is obviously cracked, which causes an open circuit across each of the copper traces that used to cross over the crack. Poor soldering results in the *solder bridge* evident in Figure 23(c). In the center of this photo, you can see where too much solder was used, causing an electrical bridge between two adjacent IC pins and making them a short. Experienced troubleshooters will also visually inspect printed-circuit boards for components that may appear to



(a) (b)

(c) (d)

Figure 23 Four common printed-circuit faults: (a) misalignment of pin 14; (b) cracked board; (c) solder bridge; (d) burned transistor.

be darkened from excessive heat. Notice the four transistors in the middle of Figure 23(d). The one on the lower left looks charred and is probably burned out, thus acting like an open.

The following troubleshooting examples will illustrate some basic troubleshooting techniques using the logic probe and pulser.

EXAMPLE 5

The IC AND gate in Figure 24 is suspected of having a fault and you want to test it. What procedure should you follow?

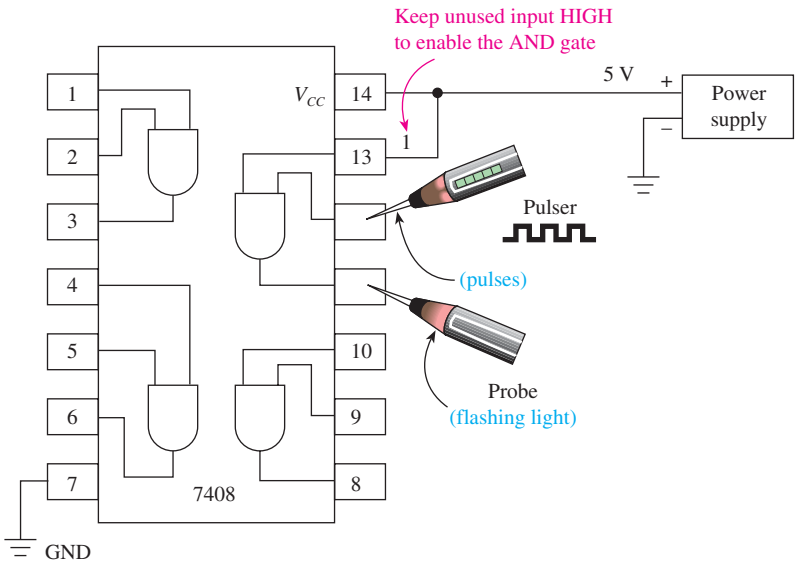


Figure 24 Connections for troubleshooting one gate of a quad AND IC.

Solution: First you apply power to V_{CC} (pin 14) and GND (pin 7). Next you want to check each AND gate with the pulser/probe. Because it takes a HIGH (1) on *both* inputs to an AND gate to make the output go HIGH, if we put a HIGH (+5 V) on one input and pulse the other, we would expect to get pulses at the output of the gate. Figure 24 shows the connections to test one of the gates of a quad AND IC. When the pulser is put on pin 12, the light in the end of the probe flashes at the same speed as the pulser, indicating that the AND gate is passing the pulses through the gate (similar in operation to the clock enable circuit of Figure 16).

The next check is to reverse the connections to pins 12 and 13 and check the probe. If the probe still flashes, that gate is okay. Proceed to the other three gates and follow the same procedure. When one of the gate outputs does not flash, you have found the fault.

Helpful Hint

You should be aware that these troubleshooting examples assume that the IC is removed from the circuit board. In-circuit testing will often give false readings because of the external circuitry connected to the IC. In that case, the circuit schematic must be studied to determine how the other ICs may be affecting the readings.

As mentioned earlier, *the key to troubleshooting an IC is understanding how the IC works.*

Common Misconception

You may mistakenly think that if you want a pin to be LOW (like pin 1), you can just leave it unconnected and it will assume a LOW level. That is not true. All inputs must be tied HIGH or LOW to have predictable results.

EXAMPLE 6

Sketch the connections for troubleshooting the first gate of a 7432 quad OR gate.

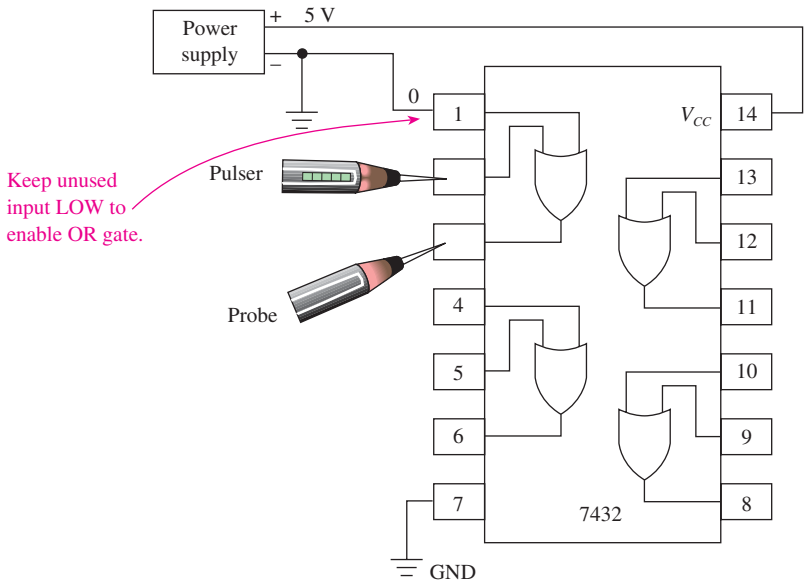


Figure 25 Connections for troubleshooting one OR gate of a 7432 IC.

Solution: The connections are shown in Figure 25. The probe should be flashing if the gate is good. Notice that the second input to the OR gate being checked is connected to a LOW (0) instead of a HIGH. The reason for this is that the output would *always* be HIGH if one input were connected HIGH. Because one input is connected LOW instead, the output will flash together with the pulses from the logic pulser if the gate is good.

EXAMPLE 7

Assume that you used a logic probe to record the levels shown in Figures 26 (a), (b), (c), and (d). Determine which gate is faulty in each IC.

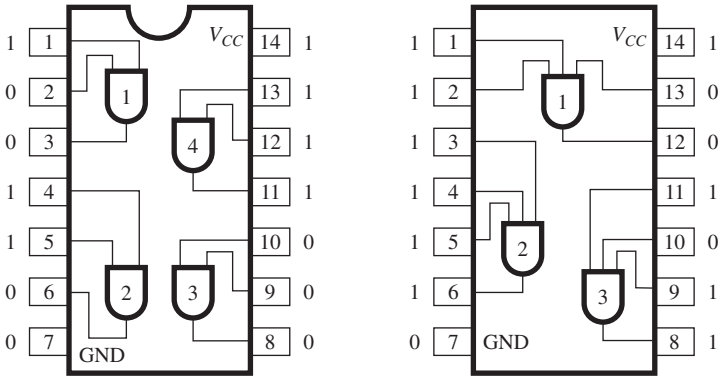


Figure 26 Troubleshooting integrated circuit AND and OR gates.

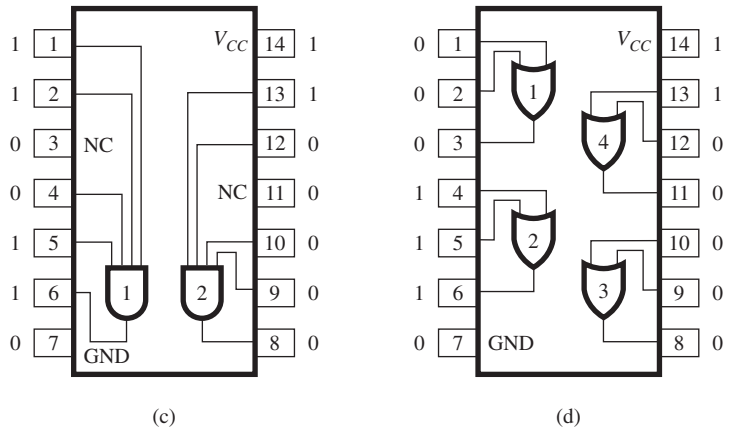


Figure 26 (*Continued*)

Answers: Figure 26(a) Gate 2
Figure 26(b) Gate 3
Figure 26(c) Gate 1
Figure 26(d) Gate 4

Review Questions

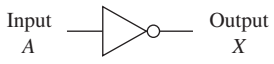
8. Which pins on the 7408 AND IC are used for power supply connections, and what voltage levels are placed on those pins?

9. How is a *logic probe* used to troubleshoot digital ICs?

10. How is a *logic pulser* used to troubleshoot digital ICs?

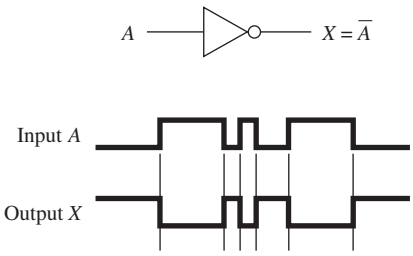
7 The Inverter

The inverter is used to complement, or invert, a digital signal. It has a single input and a single output. If a HIGH level (1) comes in, it produces a LOW-level (0) output. If a LOW level (0) comes in, it produces a HIGH-level (1) output. The symbol and truth table for the inverter gate are shown in Figure 27. (*Note:* The circle is the part of the symbol that indicates inversion. The inversion circle will be used on other gates in upcoming sections.)

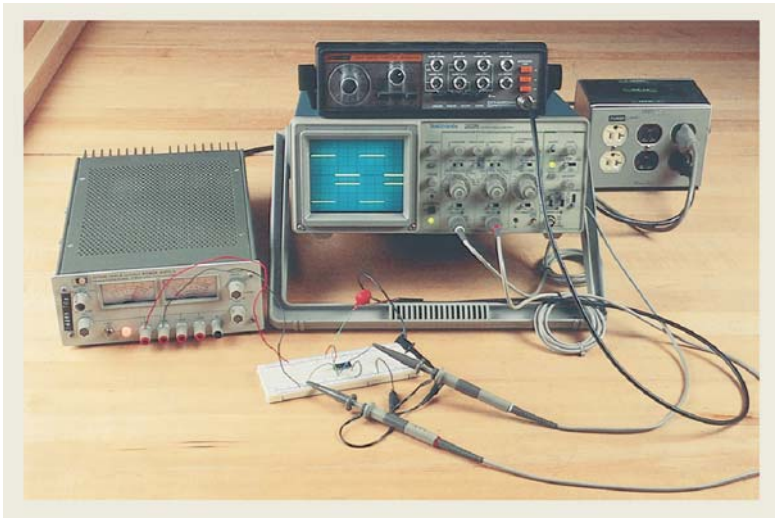


Input *A*	Output *X*
0	1
1	0

Figure 27 Inverter symbol and truth table.



(a)



(b)

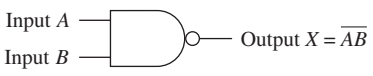
Figure 28 Timing analysis of an inverter gate: (a) waveform sketch and (b) oscilloscope display.

The operation of the inverter is very simple and can be illustrated further by studying the timing diagram of Figure 28. The timing diagram graphically shows us the operation of the inverter. When the input is HIGH, the output is LOW, and when the input is LOW, the output is HIGH. The output waveform is, therefore, the exact complement of the input.

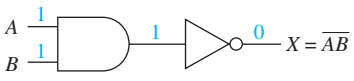
The Boolean equation for an inverter is written $X = \overline{A}(X = \text{NOT } A)$. The *bar* over the *A* is an **inversion bar**, used to signify the **complement**. The inverter is sometimes referred to as the NOT gate.

8 The NAND Gate

The operation of the NAND gate is the same as the AND gate except that its output is inverted. You can think of a NAND gate as an AND gate with an inverter at its output. The symbol for a NAND gate is made from an AND gate with the inversion circle (bubble) at its output, as shown in Figure 29(a).



(a)



(b)

Figure 29 NAND gate: (a) symbol; (b) AND–INVERT equivalent of a NAND gate with $A = 1, B = 1$.

In digital circuit diagrams, you will find the small circle used whenever complementary action (inversion) is to be indicated. The circle at the output acts just like an inverter, so a NAND gate can be drawn symbolically as an AND gate with an inverter connected to its output, as shown in Figure 29(b).

The TTL form of a NAND is the 7400 IC (or the 74LS00 or 74HC00, etc.) Figure 30 shows the output results for all possible input combinations applied to a 7400 quad NAND.

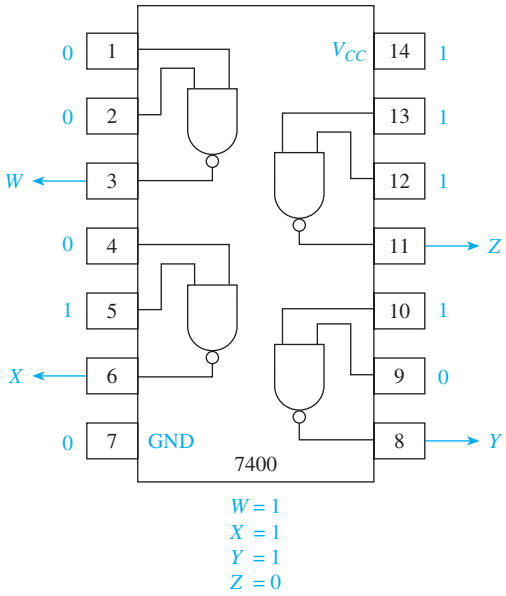


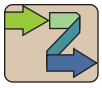
Figure 30 Inputs and outputs of a 7400 quad NAND IC.

The Boolean equation for the NAND gate is written $X = \overline{AB}$. The inversion bar is drawn over (*A* and *B*), meaning that the output of the NAND is the complement of (*A* and *B*) [**NOT** (*A* and *B*)]. Because we are inverting the output, the truth table outputs in Table 6 will be the complement of the AND gate truth table outputs. The easy way to construct the truth table is to think of how an AND gate would respond to the inputs and then invert your answer. From Table 6, we can see that the output is LOW when *both* inputs *A* and *B* are HIGH (just the opposite of an AND gate). Also, the output is HIGH whenever either input is LOW.

TABLE 6 Two-Input NAND Gate Truth Table

A	*B*	$X = \overline{AB}$
0	0	1
0	1	1
1	0	1
1	1	0

Output is always HIGH unless both inputs are HIGH.



Helpful Hint

Some students find it easier to analyze a NAND gate by solving it as an AND gate and then inverting the result.

NAND gates can also have more than two inputs. Figure 31 shows three- and eight-input NAND gate symbols. The truth table for a three-input NAND gate (see Table 7) shows that the output is always HIGH unless *all* inputs go HIGH.

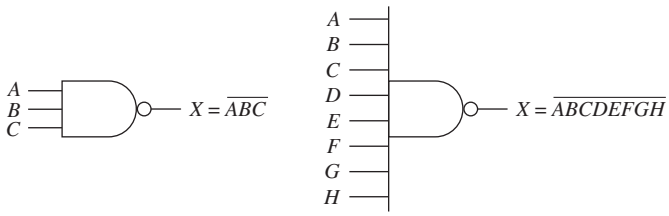


Figure 31 Symbols for three- and eight-input NAND gates.

TABLE 7 Truth Table for a Three-Input NAND Gate

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

Timing analysis can also be used to illustrate the operation of NAND gates. The following examples will contribute to your understanding.

EXAMPLE 8

Sketch the output waveform at *X* for the NAND gate shown in Figure 32, with the given input waveforms in Figure 33.

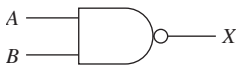


Figure 32

Solution:

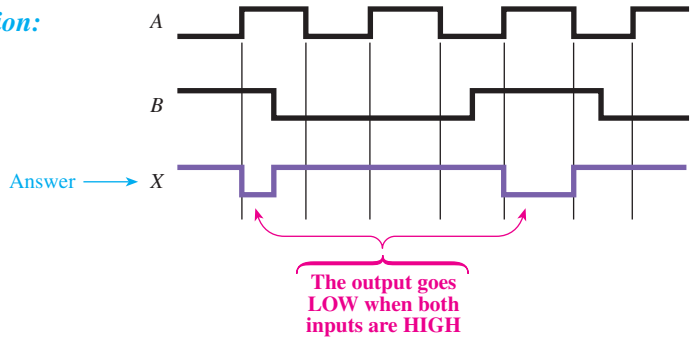
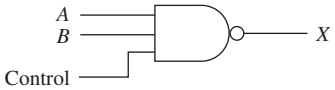


Figure 33 Timing analysis of a NAND gate.

EXAMPLE 9

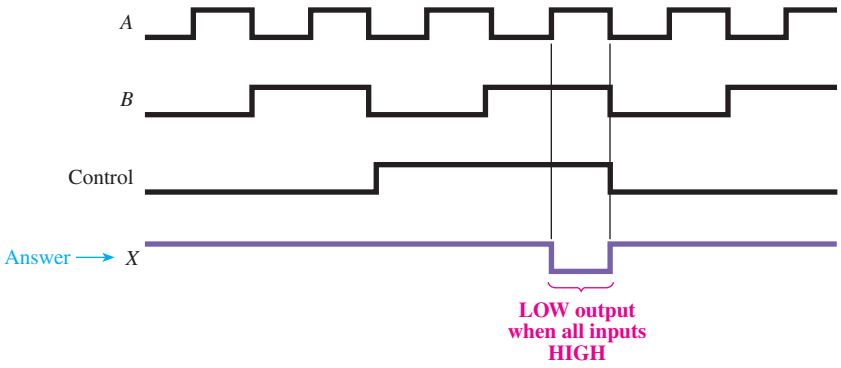
Sketch the output waveform at *X* for the NAND gate shown in Figure 34(a), with the given input waveforms at *A*, *B*, and Control.



(a)

Figure 34(a) Timing analysis of a NAND gate with a Control input: (a) logic symbol; (b) waveforms.

Solution: In Figure 34(b), the Control input waveform is used to *enable/disable* the NAND gate. When it is LOW, the output is stuck HIGH. When it goes HIGH, the output will respond LOW when *A* and *B* go HIGH.



(b)

Figure 34(b)

9 The NOR Gate

The operation of the NOR gate is the same as that of the OR gate except that its output is inverted. You can think of a NOR gate as an OR gate with an inverter at its output. The symbol for a NOR gate and its equivalent OR–INVERT symbol are shown in Figure 35.

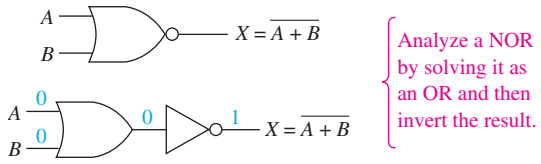


Figure 35 NOR gate symbol and its OR–INVERT equivalent with $A = 0, B = 0$.

The TTL form of a NOR is the 7402 IC (or the 74LS02 or 74HC02, etc.) Figure 36 shows the output results for all possible input combinations applied to a 7402 quad NOR.

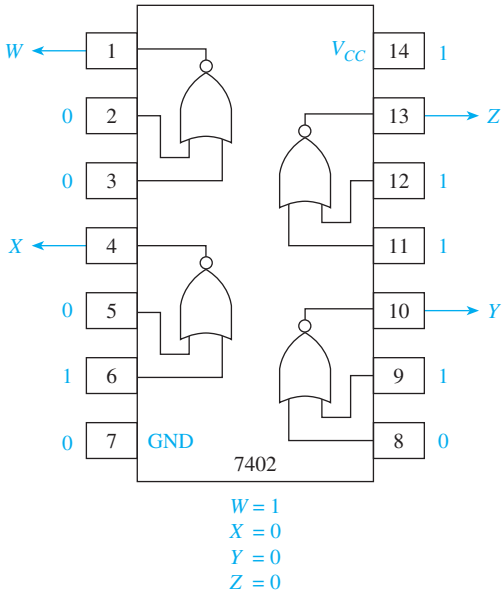


Figure 36 Inputs and outputs of a 7402 quad NOR IC.

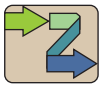
The Boolean equation for the NOR function is $X = \overline{A + B}$. The equation is stated "X equals *not* (A or B)." In other words, X is LOW if A or B is HIGH. The truth table for a NOR gate is given in Table 8. Notice that the output column is the complement of the OR gate truth table output column.

TABLE 8 Truth Table for a NOR Gate

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

Output is always LOW unless both inputs are LOW.

Now let's study some timing analysis examples to get a better grasp of NOR gate operation.



Helpful Hint

To solve a timing analysis problem, it is useful to look at the gate's truth table to see what the *unique* occurrence is for that gate. In the case of the NOR, the odd occurrence is when the output goes HIGH due to all LOW inputs.

EXAMPLE 10

Sketch the output waveform at X for the NOR gate shown in Figure 37, with the given input waveforms in Figure 38.

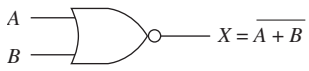


Figure 37

Solution:

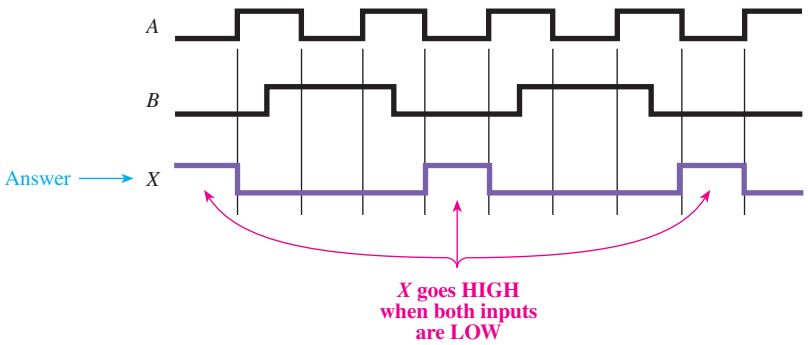


Figure 38 NOR gate timing analysis.

EXAMPLE 11

Sketch the output waveform at X for the NOR gate shown in Figure 39, with the given input waveforms in Figure 40.

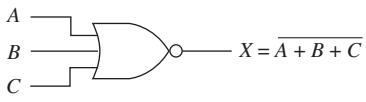


Figure 39

Solution:

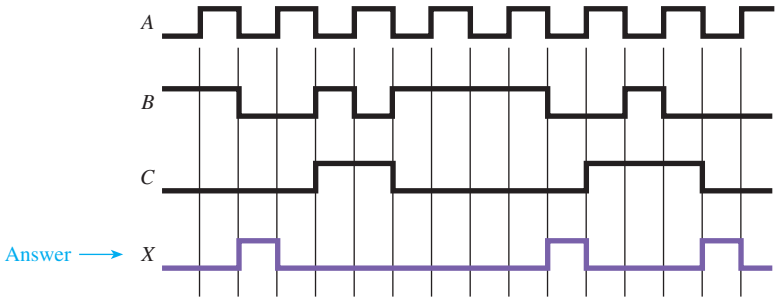


Figure 40 Three-input NOR gate timing analysis.

EXAMPLE 12

Sketch the waveform at the *B* input of the gate shown in Figure 41 that will produce the output waveform shown in Figure 42 for *X*. Repeat for the NAND gate.

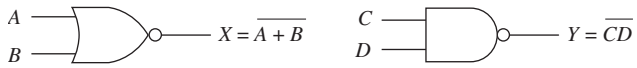


Figure 41

Solution:

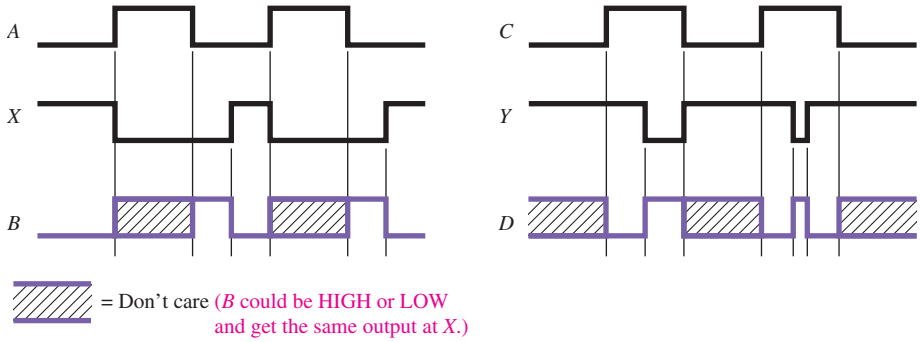


Figure 42 Input waveform requirement to produce a specific output.

Review Questions

11. What is the purpose of an inverter in a digital circuit?

12. How does a NAND gate differ from an AND gate?

13. The output of a NAND gate is always HIGH unless *all* inputs are made _____ (HIGH/LOW).

14. Write the Boolean equation for a three-input NOR gate.