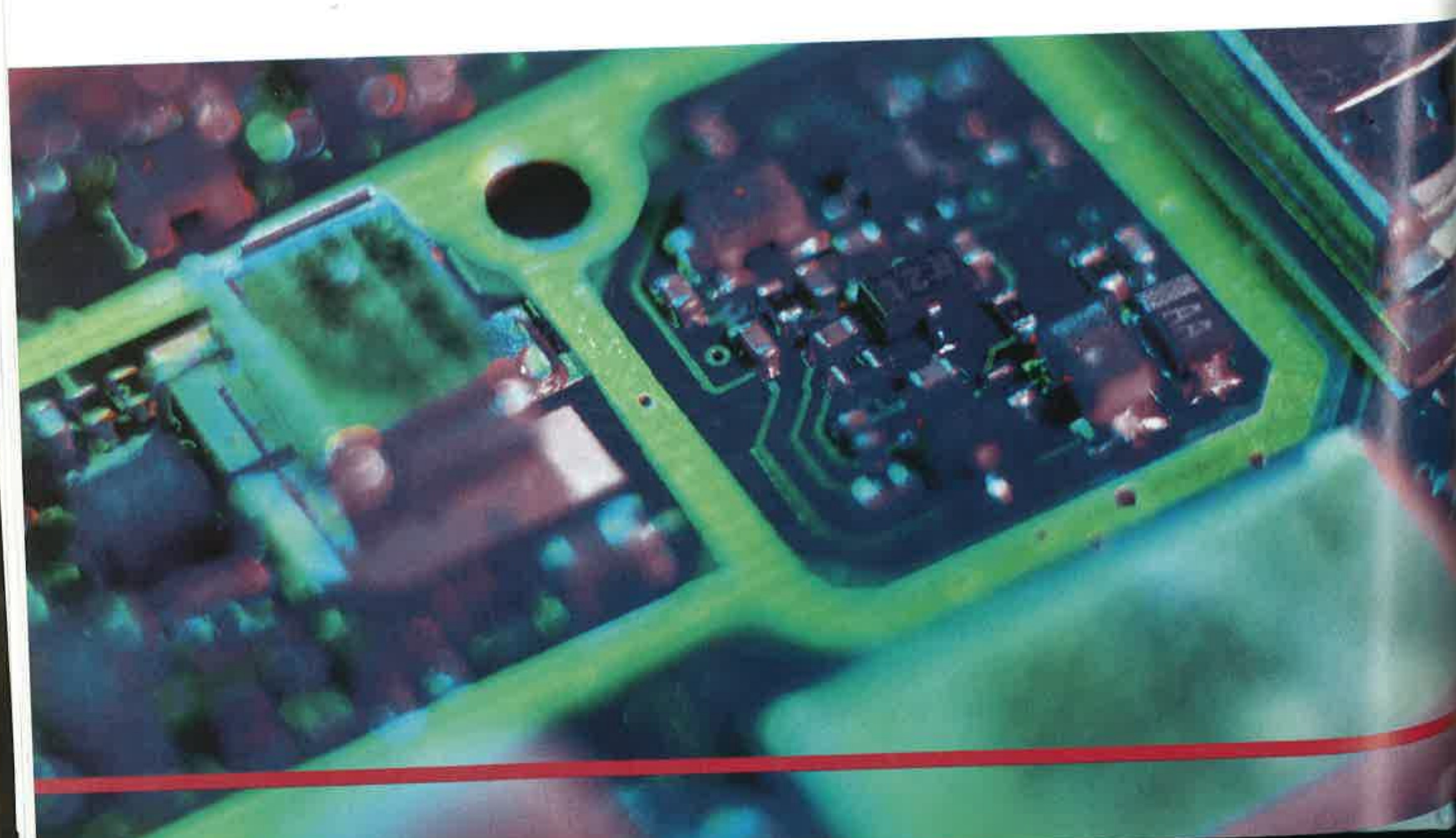
1

Introductory Digital Concepts

CHAPTER OUTLINE

CHAPTER OBJECTIVES

- Explain the basic differences between digital and analog quantities
- Show how voltage levels are used to represent digital quantities
- Describe various parameters of a pulse waveform such as rise time, fall time, pulse width, frequency, period, and duty cycle
- Explain the basic logic operations of NOT, AND, and OR
- Describe the logic functions of the comparator, adder, code converter, encoder, decoder, multiplexer, demultiplexer, counter, and register
- Identify fixed-function digital integrated circuits according to their complexity and the type of circuit packaging
- Identify pin numbers on integrated circuit packages
- Describe the PLD, discuss the various types, and state how PLDs are programmed
- Recognize digital test instruments and understand how they are used in troubleshooting digital circuits and systems
- Show how a complete digital system is formed from the basic functions in a practical application



KEY TERMS

Key terms are in order of appearance in the chapter.

- Analog
- Digital
- Binary
- Bit
- Pulse
- Clock
- Timing diagram
- Data
- Serial
- Parallel
- Logic
- Input
- Output
- Gate
- NOT
- Inverter
- AND
- OR
- Integrated circuit (IC)
- PLD
- Troubleshooting

INTRODUCTION

The term *digital* is derived from the way computers perform operations, by counting digits. For many years, applications of digital electronics were confined to computer systems. Today, digital technology is applied in a wide range of areas in addition to computers. Such applications as television, communications systems, radar, navigation and guidance systems, military systems, medical instrumentation, industrial process control, and consumer electronics use digital techniques. Digital technology has progressed from vacuum-tube circuits to discrete transistors to complex integrated circuits, some of which contain millions of transistors.

This chapter introduces you to digital electronics and provides a broad overview of many important concepts, components, and tools.

■■■ DIGITAL SYSTEM APPLICATION PREVIEW

The last feature in many chapters of this textbook uses a system application to bring together the principal topics covered in the chapter. Each system is designed to fit the particular chapter to illustrate how the theory and devices can be used. Throughout the book, four different systems are introduced, some covering two or more chapters.

All of the systems are simplified to make them manageable in the context of the chapter material. Although they are based on actual system requirements, they are designed to accommodate the topical coverage of the chapter and are not intended to necessarily represent the most efficient or ultimate approach in a given application.

This chapter introduces the first system, which is an industrial control system for counting and controlling items for packaging on a conveyor line. It is designed to incorporate all of the logic functions that are introduced in this chapter so that you can see how they are used and how they work together to achieve a useful objective.

WWW. VISIT THE COMPANION WEBSITE
Study aids for this chapter are available at
http://www.prenhall.com/floyd

1–1 DIGITAL AND ANALOG QUANTITIES

Electronic circuits can be divided into two broad categories, digital and analog. Digital electronics involves quantities with discrete values, and analog electronics involves quantities with continuous values. Although you will be studying digital fundamentals in this book, you should also know about analog because many applications require both.

After completing this section, you should be able to

- Define *analog*
- Define *digital*
- Explain the difference between digital and analog quantities
- State the advantages of digital over analog
- Give examples of how digital and analog quantities are used in electronics

An **analog*** quantity is one having continuous values. A **digital** quantity is one having a discrete set of values. Most things that can be measured quantitatively appear in nature in analog form. For example, the air temperature changes over a continuous range of values. During a given day, the temperature does not go from, say, 70° to 71° instantaneously; it takes on all the infinite values in between. If you graphed the temperature on a typical summer day, you would have a smooth, continuous curve similar to the curve in Figure 1–1. Other examples of analog quantities are time, pressure, distance, and sound.

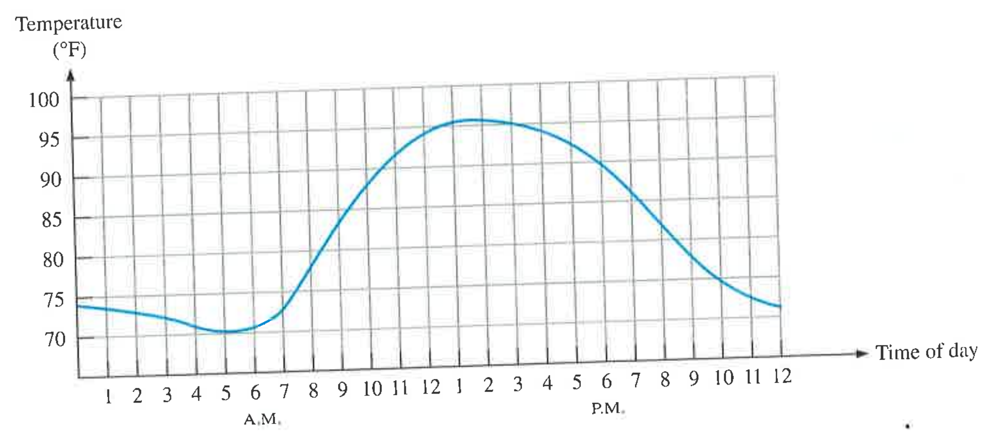


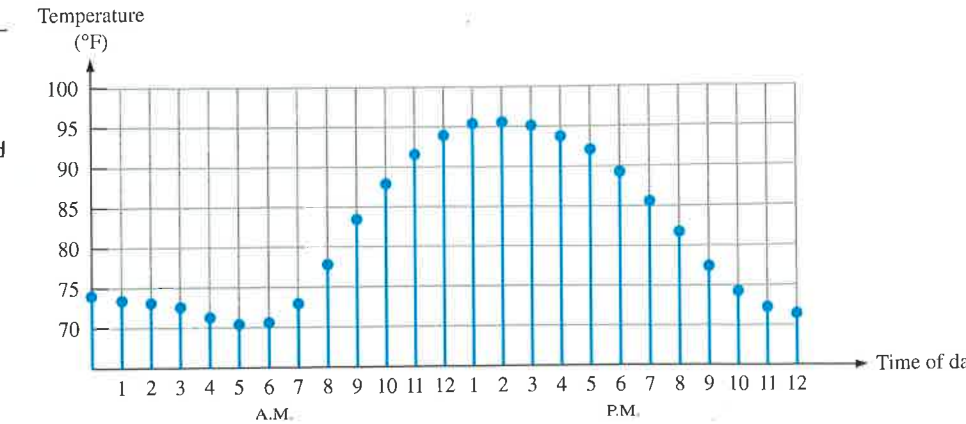
FIGURE 1–1
Graph of an analog quantity (temperature versus time).

Rather than graphing the temperature on a continuous basis, suppose you just take a temperature reading every hour. Now you have sampled values representing the temperature at discrete points in time (every hour) over a 24-hour period, as indicated in Figure 1–2. You have effectively converted an analog quantity to a form that can now be digitized by representing each sampled value by a digital code. It is important to realize that Figure 1–2 itself is not the digital representation of the analog quantity.

The Digital Advantage Digital representation has certain advantages over analog representation in electronics applications. For one thing, digital data can be processed and transmitted more efficiently and reliably than analog data. Also, digital data has a great advantage when storage is necessary. For example, music when converted to digital form can be stored more compactly and reproduced with greater accuracy and clarity than is possible when it is in analog form. Noise (unwanted voltage fluctuations) does not affect digital data nearly as much as it does analog signals.

*All bold terms are important and are defined in the end-of-book glossary. The bold terms in color are key terms and are included in a Key Term glossary at the end of each chapter.

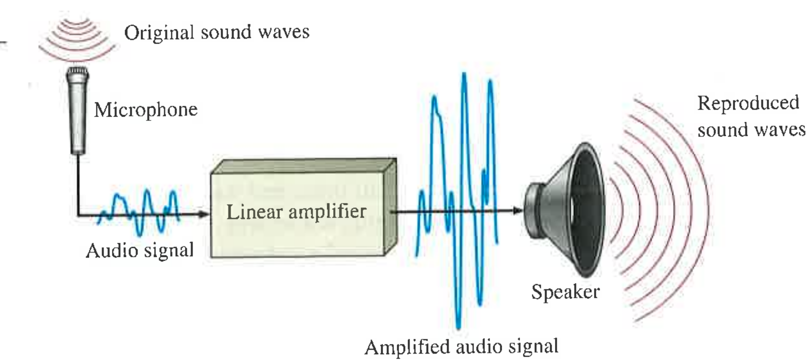
FIGURE 1–2
Sampled-value representation (quantization) of the analog quantity in Figure 1–1. Each value represented by a dot can be digitized by representing it as a digital code that consists of a series of 1s and 0s.



An Analog Electronic System

A public address system, used to amplify sound so that it can be heard by a large audience, is one example of an application of analog electronics. The basic diagram in Figure 1–3 illustrates that sound waves, which are analog in nature, are picked up by a microphone and converted to a small analog voltage called the audio signal. This voltage varies continuously as the volume and frequency of the sound changes and is applied to the input of a linear amplifier. The output of the amplifier, which is an increased reproduction of input voltage, goes to the speaker(s). The speaker changes the amplified audio signal back to sound waves that have a much greater volume than the original sound waves picked up by the microphone.

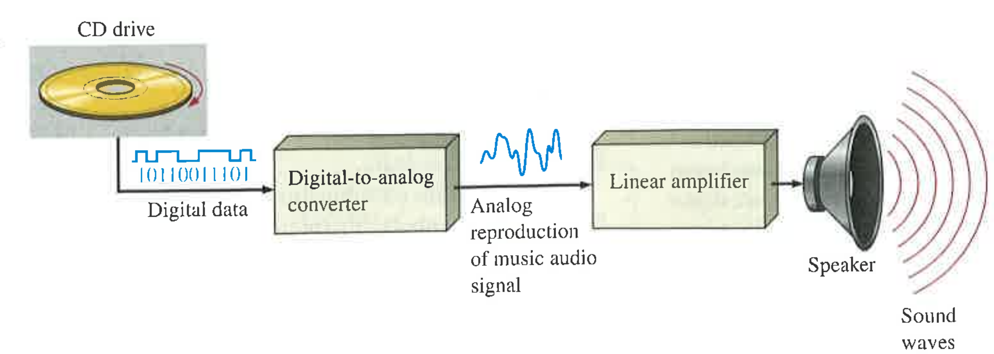
FIGURE 1–3
A basic audio public address system.



A System Using Digital and Analog Methods

The compact disk (CD) player is an example of a system in which both digital and analog circuits are used. The simplified diagram in Figure 1–4 illustrates the basic principle.

FIGURE 1–4
Basic principle of a CD player. Only one channel is shown.



Music in digital form is stored on the compact disk. A laser diode optical system picks up the digital data from the rotating disk and transfers it to the **digital-to-analog converter (DAC)**. The DAC changes the digital data into an analog signal that is an electrical reproduction of the original music. This signal is amplified and sent to the speaker for you to enjoy. When the music was originally recorded on the CD, a process, essentially the reverse of the one described here, using an **analog-to-digital converter (ADC)** was used.

SECTION 1-1 REVIEW

Answers are at the end of the chapter.

1. Define *analog*.
2. Define *digital*.
3. Explain the difference between a digital quantity and an analog quantity.
4. Give an example of a system that is analog and one that is a combination of both digital and analog. Name a system that is entirely digital.

1-2 BINARY DIGITS, LOGIC LEVELS, AND DIGITAL WAVEFORMS

Digital electronics involves circuits and systems in which there are only two possible states. These states are represented by two different voltage levels: A HIGH and a LOW. The two states can also be represented by current levels, open and closed switches, or lamps turned on and off. In digital systems such as computers, combinations of the two states, called *codes*, are used to represent numbers, symbols, alphabetic characters, and other types of information. The two-state number system is called *binary*, and its two digits are 0 and 1. A binary digit is called a *bit*.

After completing this section, you should be able to

■ Define *binary* ■ Define *bit* ■ Name the bits in a binary system ■ Explain how voltage levels are used to represent bits ■ Explain how voltage levels are interpreted by a digital circuit ■ Describe the general characteristics of a pulse ■ Determine the amplitude, rise time, fall time, and width of a pulse ■ Identify and describe the characteristics of a digital waveform ■ Determine the amplitude, period, frequency, and duty cycle of a digital waveform ■ Explain what a timing diagram is and state its purpose ■ Explain serial and parallel data transfer and state the advantage and disadvantage of each

COMPUTER NOTE

The concept of a digital computer can be traced back to Charles Babbage, who developed a crude mechanical computation device in the 1830s. The first functioning digital computer was built in 1944 at Harvard University, but it was electromechanical, not electronic. Modern digital electronics began in 1946 with an electronic digital computer called ENIAC, which was implemented with vacuum-tube circuits. Even though it took up an entire room, ENIAC didn't have the computing power that your hand-held calculator does.

Binary Digits

The two digits in the **binary** system, 1 and 0, are called **bits**, which is a contraction of the words *binary digit*. In digital circuits, two different voltage levels are used to represent the two bits. Generally, 1 is represented by the higher voltage, which we will refer to as a HIGH, and a 0 is represented by the lower voltage level, which we will refer to as a LOW. This is called **positive logic** and will be used throughout the book.

HIGH = 1 and LOW = 0

Another system in which a 1 is represented by a LOW and a 0 is represented by a HIGH is called *negative logic*.

Groups of bits (combinations of 1s and 0s), called *codes*, are used to represent numbers, letters, symbols, instructions, and anything else required in a given application.

Logic Levels

The voltages used to represent a 1 and a 0 are called *logic* levels. Ideally, one voltage level represents a HIGH and another voltage level represents a LOW. In a practical digital circuit, however, a HIGH can be any voltage between a specified minimum value and a specified maximum value. Likewise, a LOW can be any voltage between a specified minimum and a specified maximum. There can be no overlap between the accepted HIGH levels and the accepted LOW levels.

FIGURE 1-5

Logic level ranges of voltage for a digital circuit.

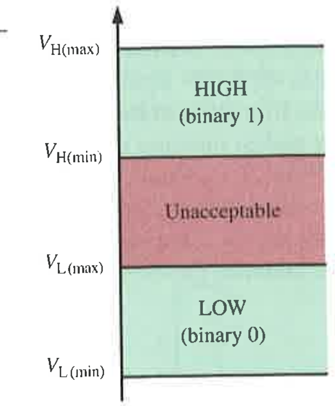


Figure 1–5 illustrates the general range of LOWs and HIGHs for a digital circuit. The variable $V_{H(max)}$ represents the maximum HIGH voltage value, and $V_{H(min)}$ represents the minimum HIGH voltage value. The maximum LOW voltage value is represented by $V_{L(max)}$, and the minimum LOW voltage value is represented by $V_{L(min)}$. The voltage values between $V_{L(max)}$ and $V_{H(min)}$ are unacceptable for proper operation. A voltage in the unacceptable range can appear as either a HIGH or a LOW to a given circuit. Therefore, these unacceptable values are never used. For example, the HIGH values for a certain type of digital circuit called TTL may range from 2 V to 5 V and the LOW values may range from 0 V to 0.8 V. So, for example, if a voltage of 3.5 V is applied, the circuit will accept it as a HIGH or binary 1. If a voltage of 0.5 V is applied, the circuit will accept it as a LOW or binary 0. For this type of circuit, voltages between 0.8 V and 2 V are unacceptable and are never used.

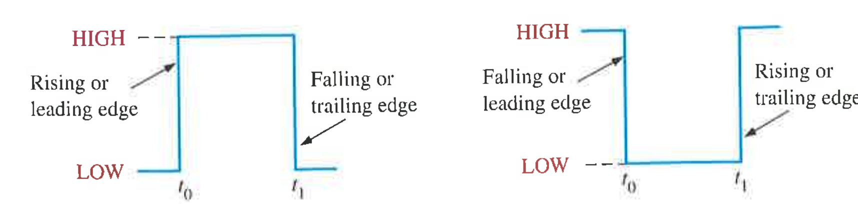
Digital Waveforms

Digital waveforms consist of voltage levels that are changing back and forth between the HIGH and LOW levels or states. Figure 1–6(a) shows that a single positive-going **pulse** is generated when the voltage (or current) goes from its normally LOW level to its HIGH level and then back to its LOW level. The negative-going pulse in Figure 1–6(b) is generated when the voltage goes from its normally HIGH level to its LOW level and back to its HIGH level. A digital waveform is made up of a series of pulses.

The Pulse As indicated in Figure 1–6, the pulse has two edges: a **leading edge** that occurs first at time t_0 and a **trailing edge** that occurs last at time t_1. For a positive-going pulse, the leading edge is a rising edge, and the trailing edge is a falling edge. The pulses in Figure 1–6 are ideal because the rising and falling edges are assumed to change in zero time (instantaneously). In practice, these transitions never occur instantaneously, although for most digital work you can assume ideal pulses.

FIGURE 1-6

Ideal pulses.



(a) Positive–going pulse

(b) Negative–going pulse

Figure 1–7 shows a nonideal pulse. The time required for the pulse to go from its LOW level to its HIGH level is called the **rise time** (t_r) and the time required for the transition from the HIGH level to the LOW level is called the **fall time** (t_f). In practice, it is common to measure rise time from 10% of the pulse **amplitude** (height from baseline) to 90% of the pulse amplitude and to measure the fall time from 90% to 10% of the pulse amplitude, as indicated in Figure 1–7. The bottom 10% and the top 10% of the pulse are not included in the rise and fall times because of the nonlinearities in the waveform in these areas. The **pulse width** (t_W) is a measure of the duration of the pulse and is often defined as the time interval between the 50% points on the rising and falling edges, as indicated in Figure 1–7.

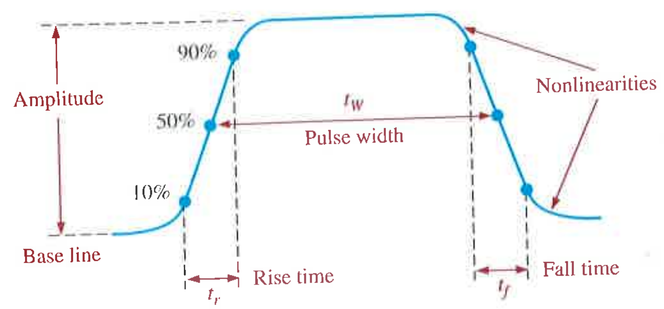


FIGURE 1–7

Nonideal pulse characteristics.

Waveform Characteristics Most waveforms encountered in digital systems are composed of series of pulses, sometimes called *pulse trains*, and can be classified as either periodic or nonperiodic. A **periodic** pulse waveform is one that repeats itself at a fixed interval, called a **period** (T). The **frequency** (f) is the rate at which it repeats itself and is measured in hertz (Hz). A nonperiodic pulse waveform, of course, does not repeat itself at fixed intervals and may be composed of pulses of randomly differing pulse widths and/or randomly differing time intervals between the pulses. An example of each type is shown in Figure 1–8.

The frequency (f) of a pulse (digital) waveform is the reciprocal of the period. The relationship between frequency and period is expressed as follows:

Equation 1–1

$$f = \frac{1}{T}$$

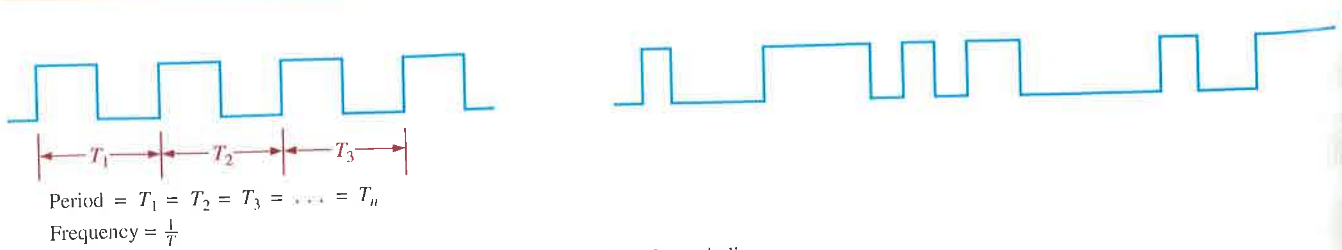
Equation 1–2

$$T = \frac{1}{f}$$

An important characteristic of a periodic digital waveform is its duty cycle. The **duty cycle** is the ratio of the pulse width (t_W) to the period (T) and can be expressed as a percentage.

Equation 1–3

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



(a) Periodic (square wave)

(b) Nonperiodic

FIGURE 1–8

Examples of digital waveforms.

EXAMPLE 1–1

A portion of a periodic digital waveform is shown in Figure 1–9. The measurements are in milliseconds. Determine the following:

(a) period **(b)** frequency **(c)** duty cycle

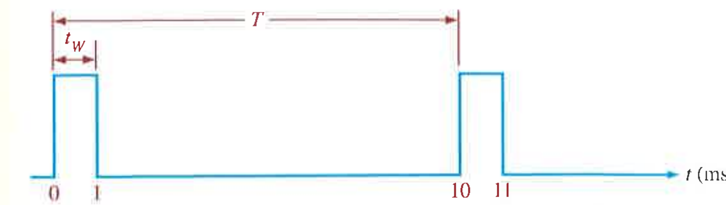


FIGURE 1–9

Solution **(a)** The period is measured from the edge of one pulse to the corresponding edge of the next pulse. In this case T is measured from leading edge to leading edge, as indicated. T equals **10 ms**.

(b) $f = \dfrac{1}{T} = \dfrac{1}{10\text{ ms}} = \mathbf{100\text{ Hz}}$

(c) $\text{Duty cycle} = \left(\dfrac{t_W}{T}\right)100\% = \left(\dfrac{1\text{ ms}}{10\text{ ms}}\right)100\% = \mathbf{10\%}$

Related Problem* A periodic digital waveform has a pulse width of 25 μs and a period of 150 μs. Determine the frequency and the duty cycle.

*Answers are at the end of the chapter.

A Digital Waveform Carries Binary Information

Binary information that is handled by digital systems appears as waveforms that represent sequences of bits. When the waveform is HIGH, a binary 1 is present; when the waveform is LOW, a binary 0 is present. Each bit in a sequence occupies a defined time interval called a **bit time**.

The Clock In digital systems, all waveforms are synchronized with a basic timing waveform called the **clock.** The clock is a periodic waveform in which each interval between pulses (the period) equals the time for one bit.

An example of a clock waveform is shown in Figure 1–10. Notice that, in this case, each change in level of waveform A occurs at the leading edge of the clock waveform.

COMPUTER NOTE

The speed at which a computer can operate depends on the type of microprocessor used in the system. The speed specification, for example 1 GHz, of a computer is the maximum clock frequency at which the microprocessor can run.

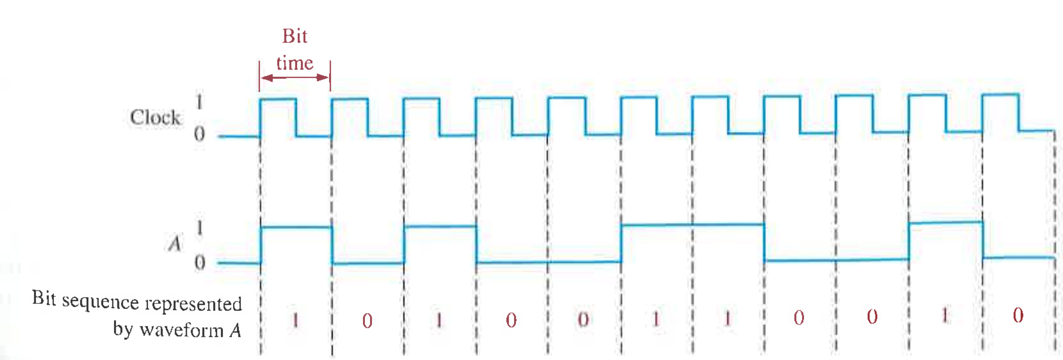


FIGURE 1–10

Example of a clock waveform synchronized with a waveform representation of a sequence of bits.

In other cases, level changes occur at the trailing edge of the clock. During each bit time of the clock, waveform A is either HIGH or LOW. These HIGHs and LOWs represent a sequence of bits as indicated. A group of several bits can be used as a piece of binary information, such as a number or a letter. The clock waveform itself does not carry information.

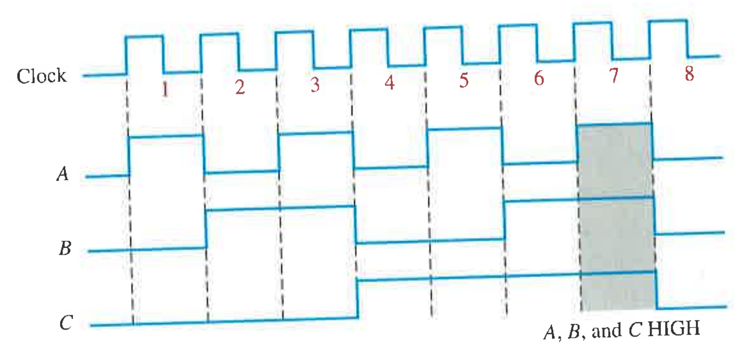
Timing Diagrams

A **timing diagram** is a graph of digital waveforms showing the actual time relationship of two or more waveforms and how each waveform changes in relation to the others. Figure 1–10 is an example of a simple timing diagram that shows how the clock waveform and waveform A are related on a time base.

By looking at a timing diagram, you can determine the states (HIGH or LOW) of all the waveforms at any specified point in time and the exact time that a waveform changes state relative to the other waveforms. Figure 1–11 is an example of a timing diagram made up of four waveforms. From this timing diagram you can see, for example, that the three waveforms A, B, and C are HIGH only during bit time 7 and they all change back LOW at the end of bit time 7 (shaded area).

▶ **FIGURE 1–11**

Example of a timing diagram.



COMPUTER NOTE

Serial data are transferred through a computer's serial port, also known as COM port. The most common uses for a serial port are with a mouse or modem. Parallel data are transferred through a computer's parallel port, also known as LPT port. The most common uses for a parallel port are with a printer or disk drive.

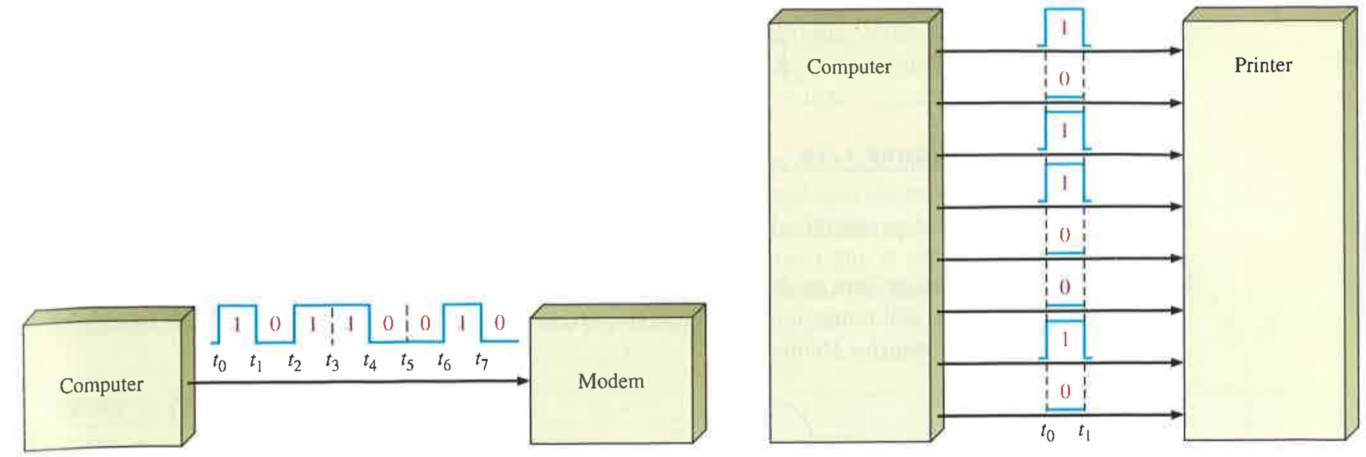
Data Transfer

Data refers to groups of bits that convey some type of information. Binary data, which are represented by digital waveforms, must be transferred from one circuit to another within a digital system or from one system to another in order to accomplish a given purpose. For example, numbers stored in binary form in the memory of a computer must be transferred to the computer's central processing unit in order to be added. The sum of the addition must then be transferred to a monitor for display and/or transferred back to the memory. In computer systems, as illustrated in Figure 1–12, binary data are transferred in two ways: serial and parallel.

When bits are transferred in **serial** form from one point to another, they are sent one bit at a time along a single conductor, as illustrated in Figure 1–12(a) for the case of a computer-to-modem transfer. During the time interval from t_0 to t_1, the first bit is transferred. During the time interval from t_1 to t_2, the second bit is transferred, and so on. To transfer eight bits in series, it takes eight time intervals.

When bits are transferred in **parallel** form, all the bits in a group are sent out on separate lines at the same time. There is one line for each bit, as shown in Figure 1–12(b) for the example of eight bits being transferred from a computer to a printer. To transfer eight bits in parallel, it takes one time interval compared to eight time intervals for the serial transfer.

To summarize, an advantage of serial transfer of binary data is that a minimum of only one line is required. In parallel transfer, a number of lines equal to the number of bits to



(a) Serial transfer of 8 bits of binary data from computer to modem. Interval t_0 to t_1 is first.

(b) Parallel transfer of 8 bits of binary data from computer to printer. The beginning time is t_0.

▲ **FIGURE 1–12**

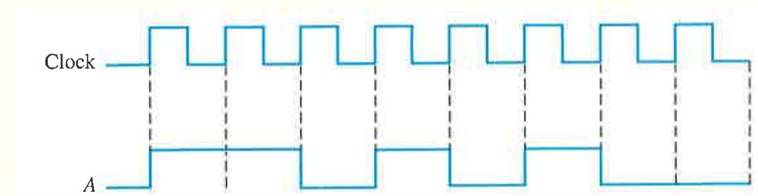
Illustration of serial and parallel transfer of binary data. Only the data lines are shown.

be transferred at one time is required. A disadvantage of serial transfer is that it takes longer to transfer a given number of bits than with parallel transfer. For example, if one bit can be transferred in 1 μs, then it takes 8 μs to serially transfer eight bits but only 1 μs to parallel transfer eight bits. A disadvantage of parallel transfer is that it takes more lines.

EXAMPLE 1–2

(a) Determine the total time required to serially transfer the eight bits contained in waveform A of Figure 1–13, and indicate the sequence of bits. The left-most bit is the first to be transferred. The 100 kHz clock is used as reference.

(b) What is the total time to transfer the same eight bits in parallel?



▲ **FIGURE 1–13**

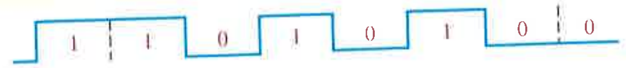
Solution **(a)** Since the frequency of the clock is 100 kHz, the period is

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

It takes 10 μs to transfer each bit in the waveform. The total transfer time for 8 bits is

$$8 \times 10\ \mu s = \mathbf{80\ \mu s}$$

To determine the sequence of bits, examine the waveform in Figure 1–13 during each bit time. If waveform A is HIGH during the bit time, a 1 is transferred. If waveform A is LOW during the bit time, a 0 is transferred. The bit sequence is illustrated in Figure 1–14. The left-most bit is the first to be transferred.



▲ **FIGURE 1–14**

(b) A parallel transfer would take **10 μs** for all eight bits.

Related Problem If binary data are transferred at the rate of 10 million bits per second (10 Mbits/s), how long will it take to parallel transfer 16 bits on 16 lines? How long will it take to serially transfer 16 bits?

SECTION 1–2 REVIEW

1. Define *binary*.
2. What does *bit* mean?
3. What are the bits in a binary system?
4. How are the rise time and fall time of a pulse measured?
5. Knowing the period of a waveform, how do you find the frequency?
6. Explain what a clock waveform is.
7. What is the purpose of a timing diagram?
8. What is the main advantage of parallel transfer over serial transfer of binary data?

1–3 INTRODUCTION TO BASIC LOGIC OPERATIONS

In its basic form, logic is the realm of human reasoning that tells you a certain proposition (declarative statement) is true if certain conditions are true. Propositions can be classified as true or false. Many situations and processes that you encounter in your daily life can be expressed in the form of propositional, or logic, functions. Since such functions are true/false or yes/no statements, digital circuits with their two-state characteristics are applicable.

After completing this section, you should be able to

■ List three basic logic operations ■ Define the NOT operation ■ Define the AND operation ■ Define the OR operation

Several propositions, when combined, form propositional, or logic, functions. For example, the propositional statement "The light is on" will be true if "The bulb is not burned out" is true and if "The switch is on" is true. Therefore, this logical statement can be made: *The light is on only if the bulb is not burned out and the switch is on.* In this example the first statement is true only if the last two statements are true. The first statement ("The light is on") is then the basic proposition, and the other two statements are the conditions on which the proposition depends.

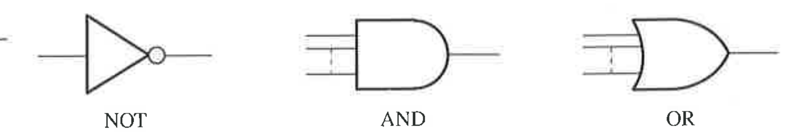
In the 1850s, the Irish logician and mathematician George Boole developed a mathematical system for formulating logic statements with symbols so that problems can be written and solved in a manner similar to ordinary algebra. Boolean algebra, as it is known today, is applied in the design and analysis of digital systems and will be covered in detail in Chapter 4.

The term **logic** is applied to digital circuits used to implement logic functions. Several kinds of digital logic **circuits** are the basic elements that form the building blocks for such complex digital systems as the computer. We will now look at these elements and discuss their functions in a very general way. Later chapters will cover these circuits in detail.

Three basic logic operations (NOT, AND, and OR) are indicated by standard distinctive shape symbols in Figure 1–15. Other standard symbols for these logic operations will be introduced in Chapter 3. The lines connected to each symbol are the **inputs** and **outputs**. The inputs are on the left of each symbol and the output is on the right. A circuit that performs a specified logic operation (AND, OR) is called a logic **gate**. AND and OR gates can have any number of inputs, as indicated by the dashes in the figure.

▶ **FIGURE 1–15**

The basic logic operations and symbols.



In logic operations, the true/false conditions mentioned earlier are represented by a HIGH (true) and a LOW (false). Each of the three basic logic operations produces a unique response to a given set of conditions.

NOT

The **NOT** operation changes one logic level to the opposite logic level, as indicated in Figure 1–16. When the input is HIGH (1), the output is LOW (0). When the input is LOW, the output is HIGH. In either case, the output is *not* the same as the input. The NOT operation is implemented by a logic circuit known as an **inverter**.

▶ **FIGURE 1–16**

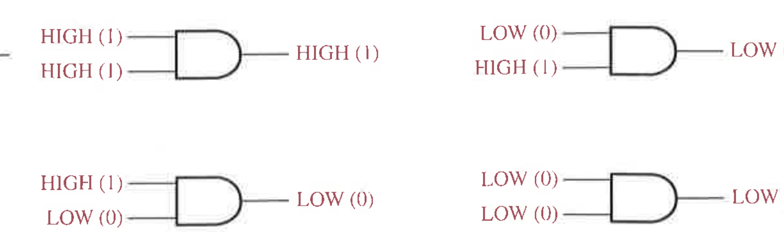
The NOT operation.

AND

The **AND** operation produces a HIGH output only if all the inputs are HIGH, as indicated in Figure 1–17 for the case of two inputs. When one input is HIGH *and* the other input is HIGH, the output is HIGH. When any or all inputs are LOW, the output is LOW. The AND operation is implemented by a logic circuit known as an *AND gate*.

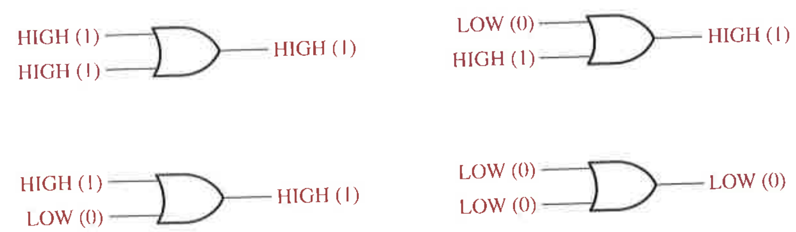
▶ **FIGURE 1–17**

The AND operation.



OR

The **OR** operation produces a HIGH output when any of the inputs is HIGH, as indicated in Figure 1–18 for the case of two inputs. When one input is HIGH *or* the other input is HIGH *or* both inputs are HIGH, the output is HIGH. When both inputs are LOW, the output is LOW. The OR operation is implemented by a logic circuit known as an *OR gate*.



▶ **FIGURE 1–18**

The OR operation.

SECTION 1–3 REVIEW

1. When does the NOT operation produce a HIGH output?
2. When does the AND operation produce a HIGH output?
3. When does the OR operation produce a HIGH output?
4. What is an inverter?
5. What is a logic gate?

1–4 BASIC OVERVIEW OF LOGIC FUNCTIONS

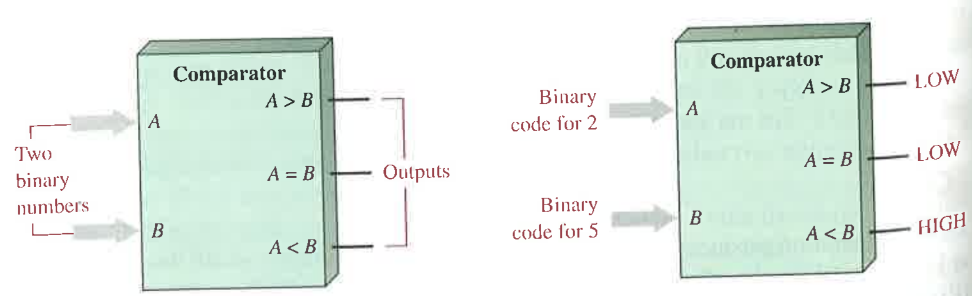
The three basic logic elements AND, OR, and NOT can be combined to form more complex logic circuits that perform many useful operations and that are used to build complete digital systems. Some of the common logic functions are comparison, arithmetic, code conversion, encoding, decoding, data selection, storage, and counting. This section provides a general overview of these important functions so that you can begin to see how they form the building blocks of digital systems such as computers. Each of the basic logic functions will be covered in detail in later chapters.

After completing this section, you should be able to

■ Identify eight basic types of logic functions ■ Describe a basic magnitude comparator ■ List the four arithmetic functions ■ Describe a basic adder ■ Describe a basic encoder ■ Describe a basic decoder ■ Define multiplexing and demultiplexing ■ State how data storage is accomplished ■ Describe the function of a basic counter

The Comparison Function

Magnitude comparison is performed by a logic circuit called a **comparator**, covered in Chapter 6. A comparator compares two quantities and indicates whether or not they are equal. For example, suppose you have two numbers and wish to know if they are equal or not equal and, if not equal, which is greater. The comparison function is represented in Figure 1–19. One number in binary form (represented by logic levels) is applied to input A, and the other number in binary form (represented by logic levels) is applied to input *B*. The outputs indicate the relationship of the two numbers by producing a HIGH level on the proper output line. Suppose that a binary representation of the number 2 is applied to input *A* and a binary representation of the number 5 is applied to input *B*. (We discuss the binary representation of numbers and symbols in Chapter 2.) A HIGH level will appear on the $A < B$ (*A* is less than *B*) output, indicating the relationship between the two numbers (2 is less than 5). The wide arrows represent a group of parallel lines on which the bits are transferred.



(a) Basic magnitude comparator

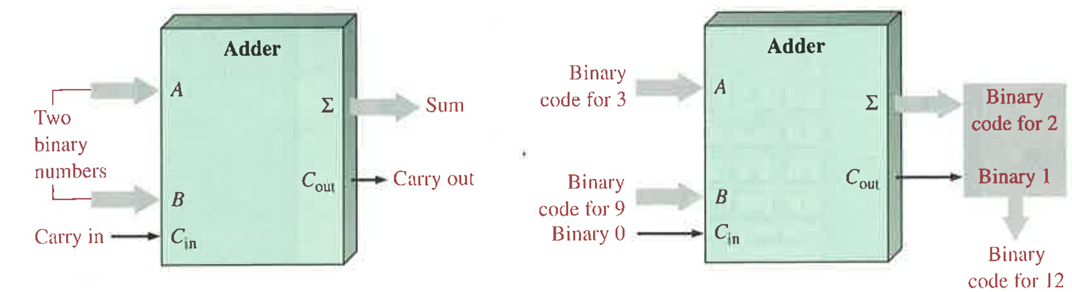
(b) Example: *A* is less than *B* ($2 < 5$) as indicated by the HIGH output ($A < B$)

▶ **FIGURE 1–19**

The comparison function.

The Arithmetic Functions

Addition Addition is performed by a logic circuit called an **adder**, covered in Chapter 6. An adder adds two binary numbers (on inputs *A* and *B* with a carry input C_{in}) and generates a sum (Σ) and a carry output (C_{out}), as shown in Figure 1–20(a). Figure 1–20(b) illustrates the addition of 3 and 9. You know that the sum is 12; the adder indicates this result by producing 2 on the sum output and 1 on the carry output. Assume that the carry input in this example is 0.



(a) Basic adder

(b) Example: *A* plus *B* ($3 + 9 = 12$)

▲ **FIGURE 1–20**

The addition function.

Subtraction Subtraction is also performed by a logic circuit. A **subtracter** requires three inputs: the two numbers that are to be subtracted and a borrow input. The two outputs are the difference and the borrow output. When, for instance, 5 is subtracted from 8 with no borrow input, the difference is 3 with no borrow output. You will see in Chapter 2 how subtraction can actually be performed by an adder because subtraction is simply a special case of addition.

> **COMPUTER NOTE**
>
> In a microprocessor, the arithmetic logic unit (ALU) performs the operations of add, subtract, multiply, and divide as well as the logic operations on digital data as directed by a series of instructions. A typical ALU is constructed of many thousands of logic gates.

Multiplication Multiplication is performed by a logic circuit called a *multiplier*. Numbers are always multiplied two at a time, so two inputs are required. The output of the multiplier is the product. Because multiplication is simply a series of additions with shifts in the positions of the partial products, it can be performed by using an adder in conjunction with other circuits.

Division Division can be performed with a series of subtractions, comparisons, and shifts, and thus it can also be done using an adder in conjunction with other circuits. Two inputs to the divider are required, and the outputs generated are the quotient and the remainder.

The Code Conversion Function

A **code** is a set of bits arranged in a unique pattern and used to represent specified information. A code converter changes one form of coded information into another coded form. Examples are conversion between binary and other codes such as the binary coded decimal (BCD) and the Gray code. Various types of codes are covered in Chapter 2, and code converters are covered in Chapter 6.

The Encoding Function

The encoding function is performed by a logic circuit called an **encoder**, covered in Chapter 6. The encoder converts information, such as a decimal number or an alphabetic character, into some coded form. For example, one certain type of encoder converts each of the decimal digits, 0 through 9, to a binary code. A HIGH level on the input corresponding to a specific decimal digit produces logic levels that represent the proper binary code on the output lines.

Figure 1–21 is a simple illustration of an encoder used to convert (encode) a calculator keystroke into a binary code that can be processed by the calculator circuits.

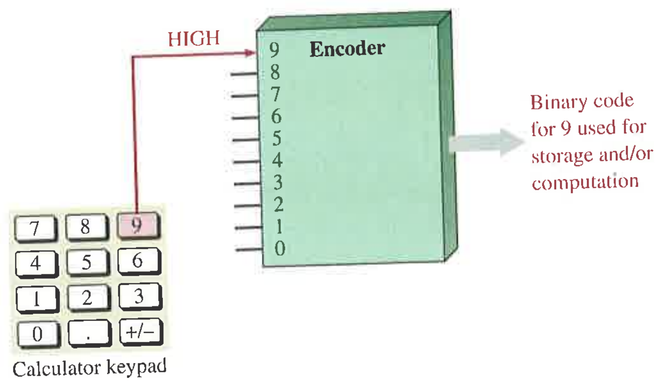


FIGURE 1–21
An encoder used to encode a calculator keystroke into a binary code for storage or for calculation.

The Decoding Function

The decoding function is performed by a logic circuit called a **decoder**, covered in Chapter 6. The decoder converts coded information, such as a binary number, into a noncoded form, such as a decimal form. For example, one particular type of decoder converts a 4-bit binary code into the appropriate decimal digit.

Figure 1–22 is a simple illustration of one type of decoder that is used to activate a 7-segment display. Each of the seven segments of the display is connected to an output line from the decoder. When a particular binary code appears on the decoder inputs, the appropriate output lines are activated and light the proper segments to display the decimal digit corresponding to the binary code.

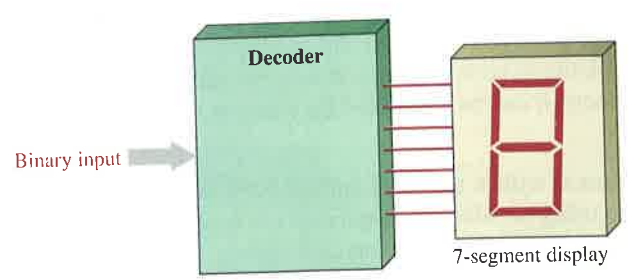


FIGURE 1–22
A decoder used to convert a special binary code into a 7-segment decimal readout.

The Data Selection Function

Two types of circuits that select data are the multiplexer and the demultiplexer. The **multiplexer**, or mux for short, is a logic circuit that switches digital data from several input lines onto a single output line in a specified time sequence. Functionally, a multiplexer

can be represented by an electronic switch operation that sequentially connects each of the input lines to the output line. The **demultiplexer** (demux) is a logic circuit that switches digital data from one input line to several output lines in a specified time sequence. Essentially, the demux is a mux in reverse.

Multiplexing and demultiplexing are used when data from several sources are to be transmitted over one line to a distant location and redistributed to several destinations. Figure 1–23 illustrates this type of application where digital data from three computers are sent out along a single line to three other computers at another location.

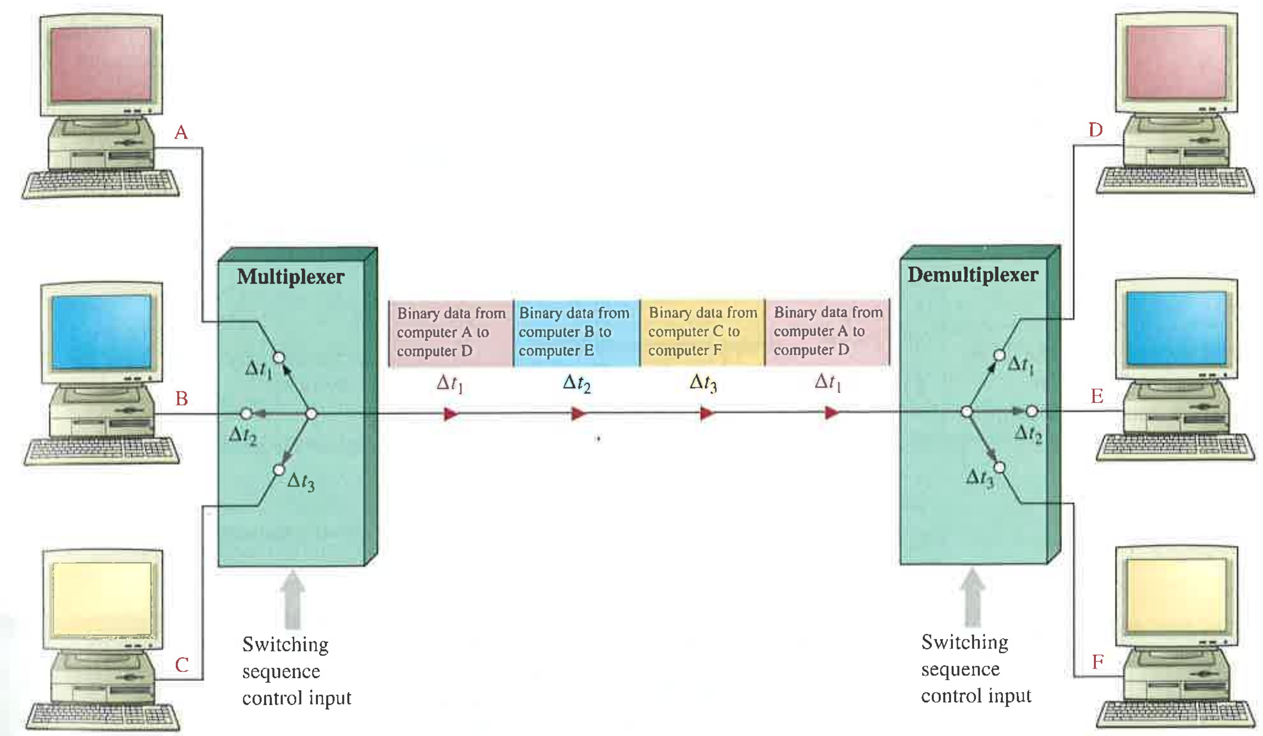


FIGURE 1–23
Illustration of a basic multiplexing/demultiplexing application.

In Figure 1–23, binary data from computer A are connected to the output line during time interval Δt_1 and transmitted to the demultiplexer that connects them to computer D. Then, during interval Δt_2, the multiplexer switches to the input from computer B and the demultiplexer switches the output to computer E. During interval Δt_3, the multiplexer switches to the input from computer C and the demultiplexer switches the output to computer F.

To summarize, during the first time interval, computer A sends data to computer D. During the second time interval, computer B sends data to computer E. During the third time interval, computer C sends data to computer F. After this, the first two computers again communicate and the sequence repeats. Because the time is divided up among several sets of systems where each has its turn to send and receive data, this process is called *time division multiplexing* (TDM).

The Storage Function

Storage is a function that is required in most digital systems, and its purpose is to retain binary data for a period of time. Some storage devices are used for short-term storage and some are used for long-term storage. A storage device can "memorize" a bit or a group of

COMPUTER NOTE

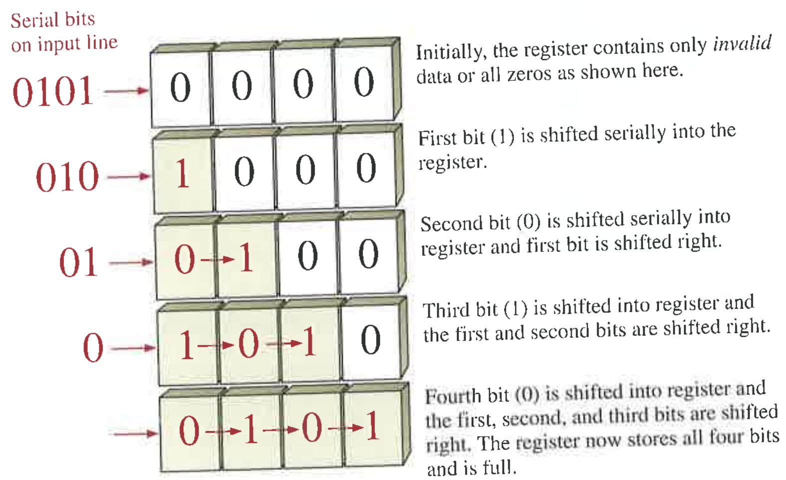
The internal computer memories, RAM and ROM, as well as the smaller caches are semiconductor memories. The registers in a microprocessor are constructed of semiconductor flip-flops. Magnetic disk memories are used in the internal hard drive, the floppy drive, and for the CD-ROM. Magnetic tape memories are often used for data backup.

bits and retain the information as long as necessary. Common types of storage devices are flip-flops, registers, semiconductor memories, magnetic disks, magnetic tape, and optical disks (CDs).

Flip-flops The **flip-flop** is a bistable (two stable states) logic circuit that can store only one bit at a time, either a 1 or a 0. The output of a flip-flop indicates which bit it is storing. A HIGH output indicates that a 1 is stored and a LOW output indicates that a 0 is stored. Flip-flops are implemented with logic gates and are covered in Chapter 8.

Registers A **register** is formed by combining several flip-flops so that groups of bits can be stored. For example, an 8-bit register is constructed from eight flip-flops. In addition to storing bits, registers can be used to shift the bits from one position to another within the register or out of the register to another circuit; therefore, these devices are known as *shift registers*. Shift registers are covered in Chapter 10.

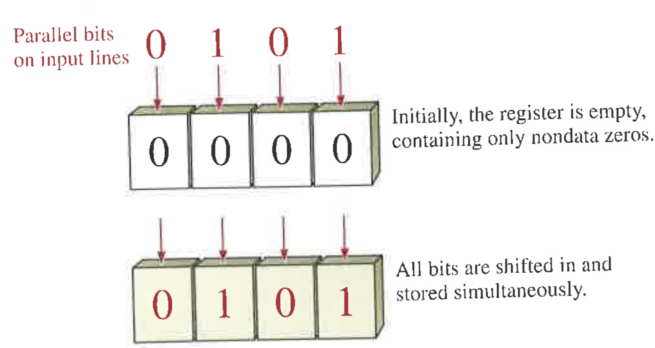
The two basic types of shift registers are serial and parallel. The bits are stored in a serial shift register one at a time, as illustrated in Figure 1–24. A good analogy to the serial shift register is loading passengers onto a bus single file through the door. They also exit the bus single file.



▶ **FIGURE 1–24**

Example of the operation of a 4-bit serial shift register. Each block represents one storage "cell" or flip-flop.

The bits are stored in a parallel register simultaneously from parallel lines, as shown in Figure 1–25. For this case, a good analogy is loading passengers on a roller coaster where they enter all of the cars in parallel.



▶ **FIGURE 1–25**

Example of the operation of a 4-bit parallel shift register.

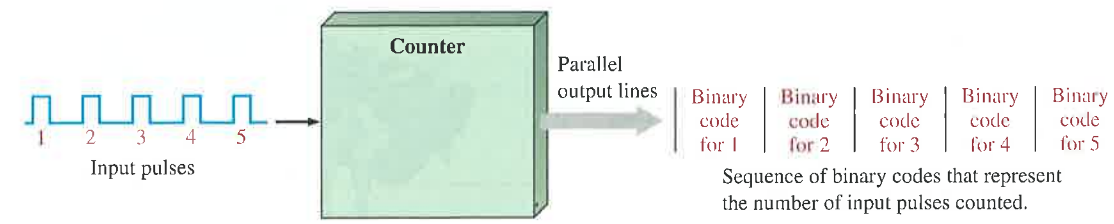
Semiconductor Memories Semiconductor memories are devices typically used for storing large numbers of bits. In one type of memory, called the *read-only memory* or ROM, the binary data are permanently or semipermanently stored and cannot be readily

changed. In the *r*andom-*a*ccess *m*emory or RAM, the binary data are temporarily stored and can be easily changed. Memories are covered in Chapter 12.

Magnetic Memories Magnetic disk memories are used for mass storage of binary data. Examples are the so-called floppy disks used in computers and the computer's internal hard disk. Magneto-optical disks use laser beams to store and retrieve data. Magnetic tape is still used in memory applications and for backing up data from other storage devices.

The Counting Function

The counting function is very important in digital systems. There are many types of digital **counters**, but their basic purpose is to count events represented by changing levels or pulses. To count, the counter must "remember" the present number so that it can go to the next proper number in sequence. Therefore, storage capability is an important characteristic of all counters, and flip-flops are generally used to implement them. Figure 1–26 illustrates the basic idea of counter operation. Counters are covered in Chapter 9.



▲ **FIGURE 1–26**

Illustration of basic counter operation.

SECTION 1–4 REVIEW

1. What does a comparator do?
2. What are the four basic arithmetic operations?
3. Describe encoding and give an example.
4. Describe decoding and give an example.
5. Explain the basic purpose of multiplexing and demultiplexing.
6. Name four types of storage devices.
7. What does a counter do?

1–5 FIXED-FUNCTION INTEGRATED CIRCUITS

All the logic elements and functions that have been discussed—and many more—are available in integrated circuit (IC) form. Digital systems have incorporated ICs for many years because of their small size, high reliability, low cost, and low power consumption. It is important to be able to recognize the IC packages and to know how the pin connections are numbered, as well as to be familiar with the way in which circuit complexities and circuit technologies determine the various IC classifications.

After completing this section, you should be able to

- Recognize the difference between through-hole devices and surface-mount fixed-function devices
- Identify dual in-line packages (DIP)
- Identify small-outline integrated circuit packages (SOIC)
- Identify plastic leaded chip carrier packages

(PLCC) ■ Identify leadless ceramic chip carrier packages (LCCC) ■ Identify flat packs (FP) ■ Determine pin numbers on various types of IC packages ■ Explain the complexity classifications for fixed-function ICs

A monolithic **integrated circuit (IC)** is an electronic circuit that is constructed entirely on a single small chip of silicon. All the components that make up the circuit—transistors, diodes, resistors, and capacitors—are an integral part of that single chip. Fixed-function logic and programmable logic are two broad categories of digital ICs. In fixed-function logic, the logic functions are set by the manufacturer and cannot be altered.

Figure 1–27 shows a cutaway view of one type of fixed-function IC package, with the circuit chip shown within the package. Points on the chip are connected to the package pins to allow input and output connections to the outside world.

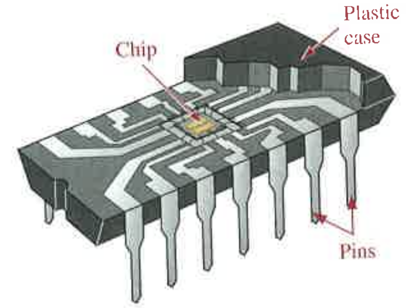


FIGURE 1–27

Cutaway view of one type of fixed-function IC package showing the chip mounted inside, with connections to input and output pins.

IC Packages

Integrated circuit (IC) packages are classified according to the way they are mounted on printed circuit (PC) boards as either through-hole mounted or surface mounted. The through-hole type packages have pins (leads) that are inserted through holes in the PC board and can be soldered to conductors on the opposite side. The most common type of through-hole package is the dual in-line package (**DIP**) shown in Figure 1–28(a).

(a) Dual in-line package (DIP)

(b) Small-outline IC (SOIC)

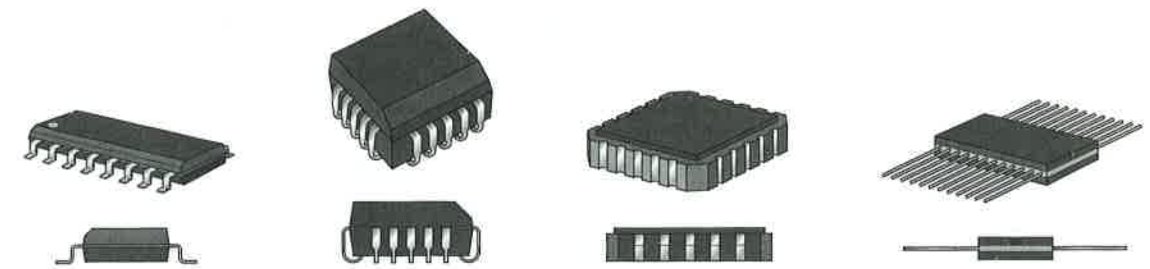
FIGURE 1–28

Examples of through-hole and surface-mounted devices. The DIP is larger than the SOIC with the same number of leads. This particular DIP is approximately 0.785 in. long, and the SOIC is approximately 0.385 in. long.

Another type of IC package uses surface-mount technology (**SMT**). Surface mounting is a space-saving alternative to through-hole mounting. The holes through the PC board are unnecessary for SMT. The pins of surface-mounted packages are soldered directly to conductors on one side of the board, leaving the other side free for additional circuits. Also, for a circuit with the same number of pins, a surface-mounted package is much smaller than a dual in-line package because the pins are placed closer together. An example of a surface-mounted package is the small-outline integrated circuit (SOIC) shown in Figure 1–28(b).

Four common types of SMT packages are the **SOIC** (small-outline IC), the **PLCC** (plastic leaded chip carrier), the **LCCC** (leadless ceramic chip carrier), and the **flat pack** (FP). These types of packages are available in various sizes depending on the number of

leads (more leads are required for more complex circuits). Examples of each type are shown in Figure 1–29. As you can see, the leads of the SOIC are formed into a "gull-wing" shape. The leads of the PLCC are turned under the package in a J-type shape. Instead of leads, the LCCC has metal contacts molded into its ceramic body. The leads of the flat pack extend straight out from the body. Other variations of SMT packages include **SSOP** (shrink small-outline package), **TSSOP** (thin shrink small-outline package), and **TVSOP** (thin very small-outline package).



(a) SOIC with "gull-wing" leads

(b) PLCC with J-type leads

(c) LCCC with no leads (contacts are part of case)

(d) Flat pack (FP) with straight leads

FIGURE 1–29

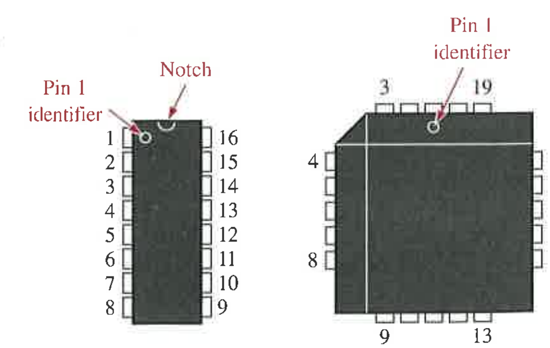
Examples of SMT package configurations.

Pin Numbering

All IC packages have a standard format for numbering the pins (leads). The dual in-line packages (DIPs), the small-outline IC packages (SOICs), and the flat packs have the numbering arrangement illustrated in Figure 1–30(a) for a 16-pin package. Looking at the top of the package, pin 1 is indicated by an identifier that can be either a small dot, a notch, or a beveled edge. The dot is always next to pin 1. Also, with the notch oriented upward, pin 1 is always the top left pin, as indicated. Starting with pin 1, the pin numbers increase as you go down, then across and up. The highest pin number is always to the right of the notch or opposite the dot.

FIGURE 1–30

Pin numbering for two standard types of IC packages. Top views are shown.



(a) DIP or SOIC

(b) PLCC or LCCC

The PLCC and LCCC packages have leads arranged on all four sides. Pin 1 is indicated by a dot or other index mark and is located at the center of one set of leads. The pin numbers increase going counterclockwise as viewed from the top of the package. The highest pin number is always to the right of pin 1. Figure 1–30(b) illustrates this format for a 20-pin PLCC package.

Complexity Classifications for Fixed-Function ICs

Fixed-function digital ICs are classified according to their complexity. They are listed here from the least complex to the most complex. The complexity figures stated here for SSI, MSI, LSI, VLSI, and ULSI are generally accepted, but definitions may vary from one source to another.

- **Small-scale integration (SSI)** describes fixed-function ICs that have up to twelve equivalent gate circuits on a single chip, and they include basic gates and flip-flops.
- **Medium-scale integration (MSI)** describes integrated circuits that have from 12 to 99 equivalent gates on a chip. They include logic functions such as encoders, decoders, counters, registers, multiplexers, arithmetic circuits, small memories, and others.
- **Large-scale integration (LSI)** is a classification of ICs with complexities of 100 to 9999 equivalent gates per chip, including memories.
- **Very large-scale integration (VLSI)** describes integrated circuits with complexities of 10,000 to 99,999 equivalent gates per chip.
- **Ultra large-scale integration (ULSI)** describes very large memories, larger **microprocessors**, and larger single-chip computers. Complexities of 100,000 equivalent gates and greater are classified as ULSI.

Integrated Circuit Technologies

The types of transistors with which all integrated circuits are implemented are either bipolar junction transistors or MOSFETs (metal-oxide semiconductor field-effect transistors). Two types of digital circuit technology that use bipolar junction transistors are TTL (transistor-transistor logic) and ECL (emitter-coupled logic). Of these two, TTL is more widely used. The major circuit technologies that use MOSFETs are CMOS (complementary MOS) and NMOS (n-channel MOS). Microprocessors use MOS technology.

All gates and other functions can be implemented with either type of circuit technology. SSI and MSI circuits are generally available in both TTL and CMOS. LSI, VLSI, and ULSI are generally implemented with CMOS or NMOS because it requires less area on a chip and consumes less power. There is more on these integrated technologies in Chapter 3. In addition, Chapter 15 provides a complete circuit-level coverage that can be introduced at various points throughout the book. Suggested points are indicated by the icon 15. Chapter 15 can also be omitted without affecting any other material.

HANDS ON TIP

Because of their particular structure, MOS devices are very sensitive to static charge and can be damaged by electrostatic discharge (ESD) if not handled properly. The following precautions should be taken when you work with MOS devices:

- MOS devices should be shipped and stored in conductive foam.
- All instruments and metal benches used in testing should be connected to earth ground.
- The handler's wrist should be connected to earth ground with a length of wire and high-value series resistor.
- Do not remove a MOS device (or any device for that matter) from a circuit while the dc power is on.
- Do not connect ac or signal voltages to a MOS device while the dc power supply is off.

SECTION 1-5 REVIEW

1. What is an integrated circuit?
2. Define the terms DIP, SMT, SOIC, SSI, MSI, LSI, VLSI and ULSI.
3. Generally, in what classification does a fixed-function IC with the following number of equivalent gates fall?
   (a) 75 (b) 500 (c) 10 (d) 15,000 (e) 200,000

1-6 PROGRAMMABLE LOGIC DEVICES (PLDs)

In the last section, you learned the various categories (e.g., SSI and MSI) of certain fixed-function logic circuits that are available and you saw some of the packaging configurations. In fixed-function devices, a specific logic function is contained in the IC package when it is purchased and it can never be changed.

Another category of logic device is one in which the logic function is programmed by the user and, in some cases, can be reprogrammed many times. These devices are called *programmable logic devices* or PLDs. This section is a brief introduction to PLDs and how they compare to fixed-function logic devices. In the following chapters, PLDs are covered in detail.

After competing this section, you should be able to

- State the major types of PLDs
- Explain the difference between PLDs and fixed-function devices (SSI and MSI)
- State the advantages of PLDs over SSI and MSI logic devices (fixed-function)
- Describe two ways in which a logic function can be programmed into a PLD

In many applications the **PLD** has replaced the hard-wired fixed-function logic device. You can expect to see a continued growth in PLDs. However, fixed-function logic is still important and will be around for a long time but in more limited applications. One area in which fixed-function logic is very effective is in the laboratory for teaching basic concepts.

One advantage of PLDs over fixed-function logic devices is that many more logic circuits can be "stuffed" into a much smaller area with PLDs. A second advantage is that, with certain PLDs, logic designs can be readily changed without rewiring or replacing components. A third advantage is that, generally, a PLD design can be implemented faster than one using fixed-function ICs once the required programming language is mastered.

Types of PLDs

The three major types of programmable logic are SPLD, CPLD, and FPGA. Each major type generally has several manufacturer-specific subcategories.

SPLDs (simple programmable logic devices) are the least complex form of PLDs. An SPLD can typically replace several fixed-function SSI or MSI devices and their interconnections. The SPLD was the first type of programmable logic available. A few categories of SPLD are listed below, some of which are unique to a specific manufacturer. A typical package has 24 to 28 pins, and one is shown in Figure 1–31.

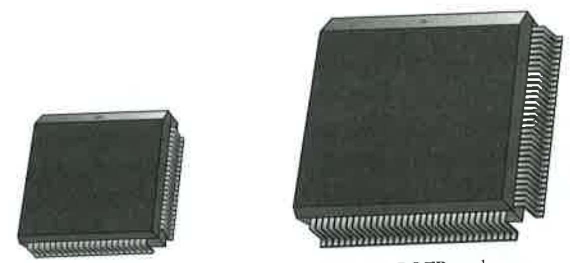
- PAL (programmable array logic)
- GAL (generic array logic)
- PLA (programmable logic array)
- PROM (programmable read-only memory)

CPLDs (complex programmable logic devices) have a much higher capacity than SPLDs, permitting more complex logic circuits to be programmed into them. A typical

FIGURE 1–31
Typical SPLD package.

CPLD is the equivalent of from two to sixty-four SPLDs. The development of these devices followed the SPLD as advances in technology permitted higher-density chips to be implemented. There are several forms of CPLD, which vary in complexity and programming capability. CPLDs typically come in 44-pin to 160-pin packages depending on the complexity. Examples of CPLD packages are shown in Figure 1–32.

FIGURE 1–32
Typical CPLD packages.



(a) 84-pin PLCC package (b) 128-pin PQFP package

FPGAs (field-programmable gate arrays) are different from SPLDs and CPLDs in their internal organization and have the greatest logic capacity. FPGAs consist of an array of anywhere from sixty-four to thousands of logic-gate groups that are sometimes called *logic blocks*. Two basic classes of FPGA are *course-grained* and *fine-grained*. The course-grained FPGA has large logic blocks, and the fine-grained FPGA has much smaller logic blocks. FPGAs come in packages ranging up to 1000 pins or more.

PLD Programming

A logic circuit design for a PLD is entered using one of two basic methods: schematic entry or text-based entry. Sometimes, a combination of both methods is used.

In the schematic entry method, the software allows the user to enter a logic design using logic components (e.g., logic gates, flip-flops) and to interconnect them on the computer screen to form a schematic diagram.

In the text-based entry method, also known as *language-based* entry, the software allows the user to enter a logic design in the form of text using a **hardware description language (HDL)**. Several HDLs are available, such as ABEL and CUPL that were originally developed for programming SPLDs and are still widely used. (ABEL will be covered later in this textbook).

An HDL that is becoming widely used, especially for programming CPLDs and FPGAs, is VHDL, a standard developed by the Department of Defense and adopted by the IEEE (Institute of Electrical and Electronics Engineers). The latest version of VHDL is IEEE std. 1076-1993. Verilog is another popular HDL for programming CPLDs and FPGAs. In addition, there are several proprietary HDLs provided by various manufacturers for their products.

SECTION 1–6 REVIEW

1. What does PLD stand for?
2. What does SPLD stand for?
3. What does CPLD stand for?
4. What does FPGA stand for?

5. Basically, how does a CPLD differ from an SPLD?
6. List two ways in which a logic design can be entered for PLD programming.
7. What does HDL mean?
8. Name two HDLs that were originally developed for SPLDs.
9. Name an important IEEE standard HDL.

1–7 INTRODUCTION TO TEST INSTRUMENTS

Troubleshooting is the technique of systematically isolating, identifying, and correcting a fault in a circuit or system. A variety of instruments are available for use in digital troubleshooting and testing. Some typical equipment is presented in this section.

After completing this section, you should be able to

■ Recognize common oscilloscope controls ■ Distinguish between an analog and digital oscilloscope ■ Determine the amplitude, period, frequency, and duty cycle of a digital waveform with an oscilloscope. ■ Discuss what a logic analyzer does ■ List common logic analyzer formats ■ Explain how the logic probe, pulser, and current probe are used ■ State the purposes of the dc power supply, function generator, and digital multimeter

The Oscilloscope

The oscilloscope (scope for short) is one of the most widely used instruments for general testing and **troubleshooting**. It is basically a graph-displaying device that traces a graph of an electrical signal on its screen. In most applications, the graphs show how signals change over time. The vertical axis of the screen represents voltage and the horizontal axis represents time. In digital applications, pulse waveforms can be displayed and parameters such as amplitude, period, frequency, rise and fall times, and duty cycle can be measured. Also, two pulse waveforms (with 2-channel oscilloscopes) can be displayed simultaneously so that their time relationship can be determined.

There are two basic types of oscilloscope, analog and digital. Both types can be used to view pulse waveforms found in digital circuits. An analog scope works by applying the signal voltage being measured to an electron beam that is sweeping across the screen. The measured voltage deflects the beam up and down proportional to the amount of voltage, immediately tracing the waveform on the screen. A digital scope samples the signal voltage and uses an analog-to-digital converter (ADC) to convert the voltage being measured into digital information. The digital information is then used to reconstruct the waveform on the screen. Typical bench type and handheld oscilloscopes are shown in Figure 1–33.

Although there are still some analog scopes being manufactured and many older ones that are still in use, the digital scope is widely used and is becoming the instrument of choice to replace older analog instruments.

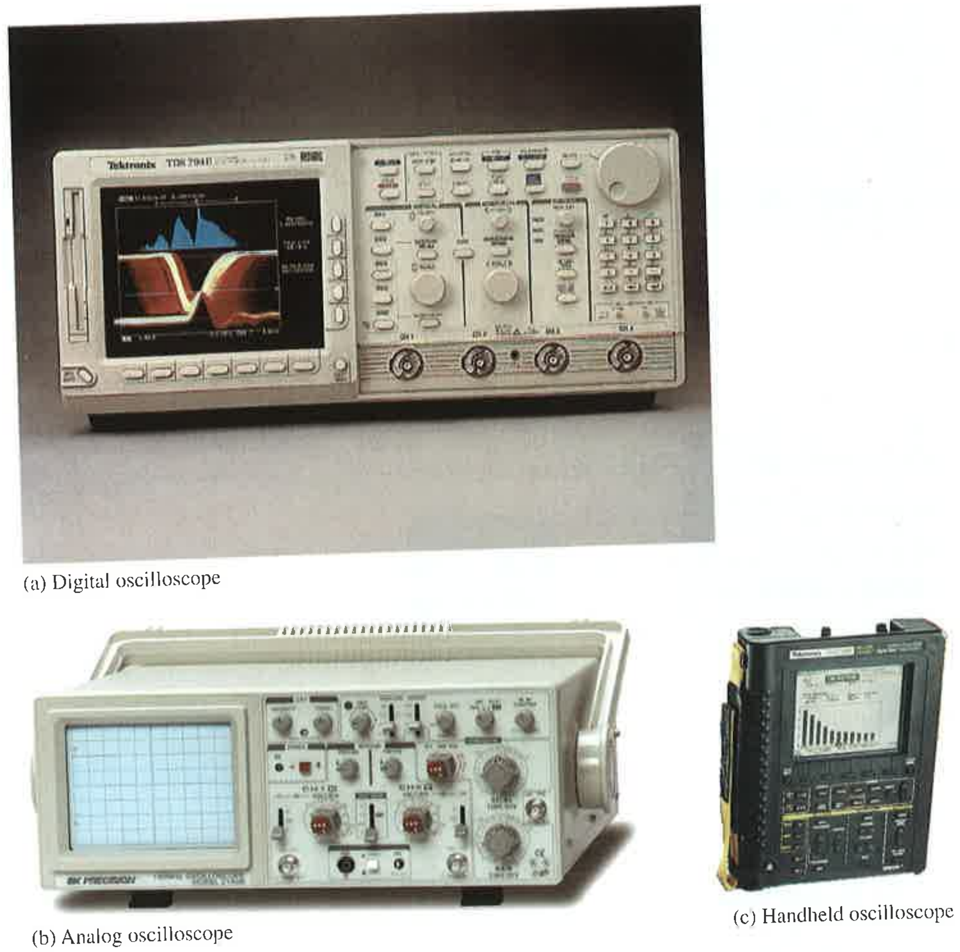
Digital Oscilloscope

A front-panel view of a typical dual-channel digital oscilloscope is shown in Figure 1–34. Instruments vary depending on model and manufacturer, but most have certain common features. Notice that the two Vertical sections and the Horizontal section each contain a Position control, a channel Menu button, and a Volts/Div or Sec/Div control. Most digital scopes have automated measurements, menu selections, screen displays of parameter settings, and other features not found on analog scopes.

We will discuss some of the controls here. Refer to the user manual for a complete coverage of the details of your particular oscilloscope.

▶ FIGURE 1–33

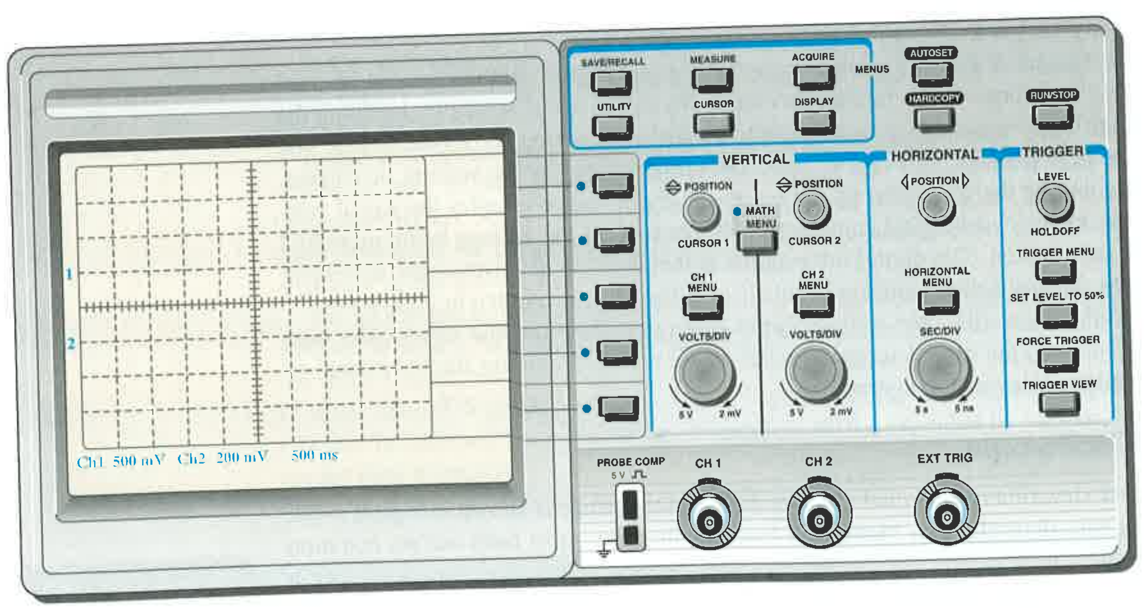
Typical oscilloscopes. (Parts (a) and (c) copyright Tektronix, Inc. All rights reserved. Reproduced by permission. Part (b) courtesy of B&K Precision Corp.)



(a) Digital oscilloscope

(b) Analog oscilloscope

(c) Handheld oscilloscope



▲ FIGURE 1–34

A typical dual-channel digital oscilloscope. Numbers below screen are arbitrary and are shown for illustration only.

In the Vertical section, there are identical controls for each of the two channels (CH1 and CH2). The Position control lets you move a displayed waveform up or down vertically on the screen. The channel Menu button provides for the selection of several items that appear on the screen, such as the coupling (dc, ac, or ground), course or fine adjustment for the Volts/Div, probe attenuation (1×, 10×, for example), and other parameters. The Volts/Div control adjusts the number of volts represented by each vertical division on the screen. For this type of scope, the Volts/Div setting for each channel is displayed on the screen instead of having to read it from a dial setting, as on many analog scopes. The Math Menu button provides a selection of operations that can be performed on the input waveforms, such as subtraction, addition, and inversion.

In the Horizontal section, the controls apply to both channels. The Position control lets you move a displayed waveform left or right horizontally on the screen. The Horizontal Menu button provides for the selection of several items that appear on the screen, such as main time base, expanded view of a portion of a waveform, and other parameters. The Sec/Div control adjusts the time represented by each horizontal division or main time base. For this type of scope, the Sec/Div setting is displayed on the screen instead of having to read it from a dial setting, as on many analog scopes.

In the Trigger section, the Level control determines the point on the triggering waveform where triggering occurs to initiate the display sweep for the input channel waveforms. The Trigger Menu button provides for the selection of several items that appear on the screen, including edge or slope triggering, trigger source, trigger mode, and other parameters.

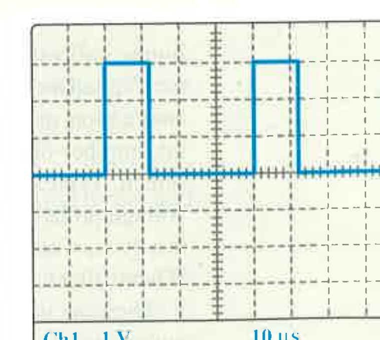
There are four input connectors on the front panel. The two marked CH1 and CH2 are for connecting the input signals for display on the screen using voltage probes. The EXT TRIG connector is for connecting an external trigger source. The Probe Comp is used to electrically match the voltage probe to the input circuit.

Probe Attenuation All scope **probes** have an attenuation factor that indicates by how much they reduce the input signal. A 1× probe does not reduce the input signal at all, while a 10× probe reduces the signal by a factor of 10. The larger the attenuation factor, the less the oscilloscope will load the circuit under test. Less loading means more accuracy. When you use a voltage probe to measure voltage in a circuit, you should always make sure that it is compatible with the scope in terms of the attenuation factor. Use the channel menu in the Vertical section to select the proper probe attenuation to match your probe. For example, if you are using a 10× probe, the 10× attenuation should be selected from the menu.

EXAMPLE 1–3

Based on the readouts, determine the amplitude and the period of the pulse waveform on the screen of a digital oscilloscope as shown in Figure 1–35. Also, calculate the frequency.

▶ FIGURE 1–35



Solution The V/div setting is 1 V. The pulses are three divisions high. Since each division represents 1 V, the pulse amplitude is

$$\text{Amplitude} = (3\text{ div})(1\text{ V/div}) = \mathbf{3\ V}$$

The sec/div setting is 10 μs. A full cycle of the waveform (from beginning of one pulse to the beginning of the next) covers four divisions; therefore, the period is

$$\text{Period} = (4\text{ div})(10\ \mu\text{s/div}) = \mathbf{40\ \mu s}$$

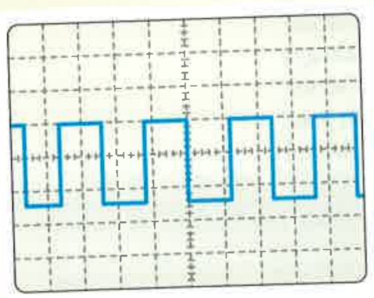
The frequency is calculated as

$$f = \frac{1}{T} = \frac{1}{40\ \mu\text{s}} = \mathbf{25\ kHz}$$

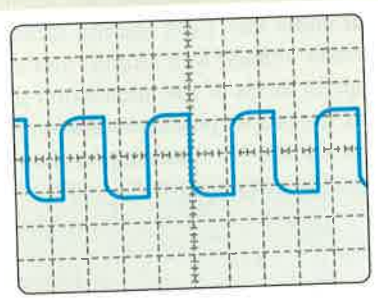
Related Problem For a V/div setting of 4 V and sec/div setting of 2 ms, determine the amplitude and period of the pulse shown on the screen in Figure 1–35.

HANDS ON TIP

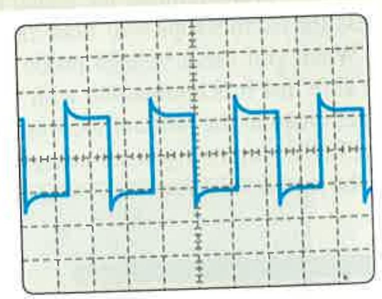
Most measurements you make with an oscilloscope require a 10× attenuator probe that reduces the signal voltage and circuit loading by a factor of 10. Before making any measurement, you should make sure the probe is compensated because an improperly compensated probe can distort the waveforms that you see on the scope screen. On a probe, there is a screwdriver adjustment or other means for adjusting compensation and there is normally a calibrated square wave output available on the front panel of the oscilloscope. Scope waveforms are shown here for three probe conditions: *properly compensated, undercompensated,* and *overcompensated.*



Properly compensated



Undercompensated



Overcompensated

Analog Oscilloscope

Figure 1–36 shows a generic front panel view of a typical analog oscilloscope. There are two Volts/Div controls, one for each of the two channels. These controls determine the attenuation or amplification of the signal, and each of the settings on the dial indicate the number of volts (V) or millivolts (mV) represented by each vertical division on the screen. Typically, there are two brackets on the Volts/Div dial, one for a times 10 (10×) voltage probe and one for a times 1 (1×) voltage probe. The most commonly used probe is a 10×, which attenuates the input signal by ten to reduce loading effects on the circuit. When using a 10× probe, you must read the Volts/Div setting in the 10× bracket.

The time base or Sec/Div control sets the number of seconds (s), milliseconds (ms), or microseconds (μs) represented by each horizontal division as indicated by the dial settings. You can measure the period of a pulse waveform on the horizontal scale and then calculate the frequency.

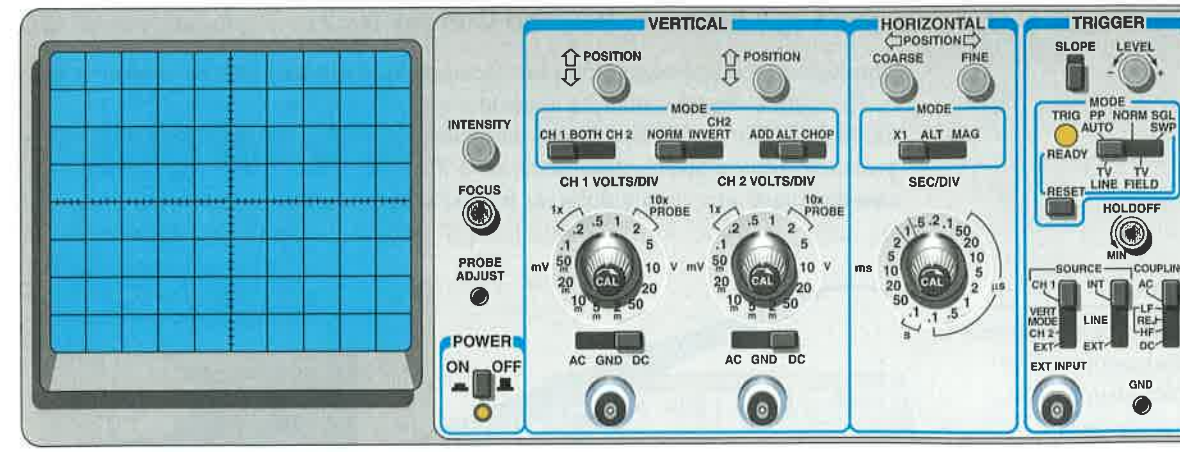


FIGURE 1–36

A typical dual-channel analog oscilloscope.

The Logic Analyzer

Typical logic analyzers are shown in Figure 1–37. This type of instrument can detect and display digital data in several formats.

FIGURE 1–37

Typical logic analyzers. (Copyright Tektronix, Inc. All rights reserved. Reproduced by permission.)

Oscilloscope Format The logic analyzer can be used to display single or dual waveforms on the screen so that characteristics of individual pulses or waveform parameters can be measured.

Timing Diagram Format The logic analyzer can display typically up to thirty-two waveforms in proper time relationship so that you can analyze sets of waveforms and determine how they change in time with respect to each other.

State Table Format The logic analyzer can display binary data in tabular form. For example, various memory locations in a microprocessor-based system can be examined to determine the contents. The data can be displayed in a variety of number systems and codes such as binary, hexadecimal, octal, binary coded decimal (BCD), and ASCII. These number systems and codes are the topics of the next chapter.

The Logic Probe, Pulser, and Current Tracer

The logic probe is a convenient, inexpensive handheld tool that provides a means of troubleshooting a digital circuit by sensing various conditions at a point in a circuit, as illustrated in Figure 1–38. The probe can detect high-level voltage, low-level voltage, single pulses, repetitive pulses, and opens on a PC board. The probe lamp indicates the condition that exists at a certain point, as indicated in the figure.

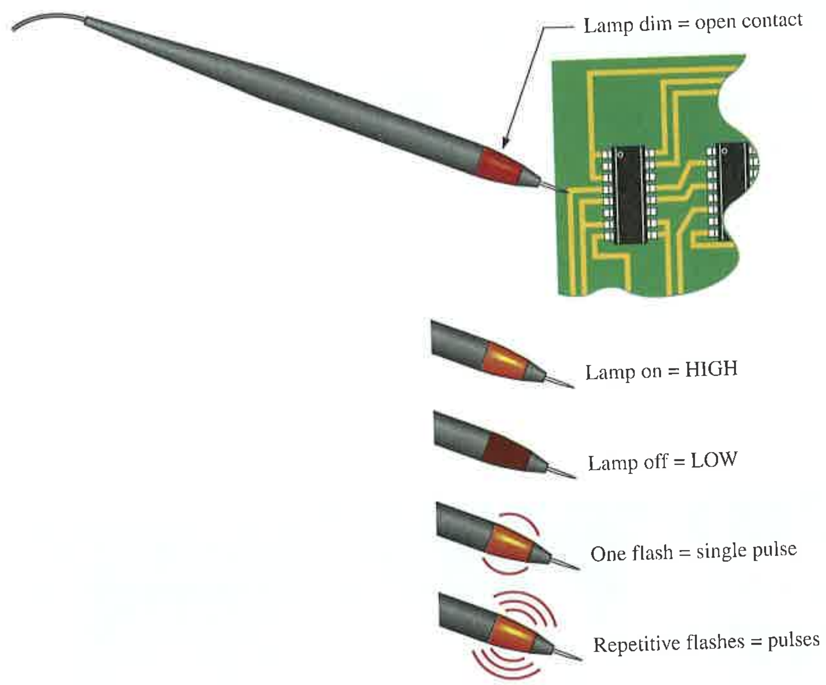


FIGURE 1–38

Illustration of how a logic probe is used to detect various voltage conditions at a given point in a circuit.

The logic pulser produces a repetitive pulse waveform that can be applied to any point in a circuit. You can apply pulses at one point in a circuit with the pulser and check another point for resulting pulses with a logic probe. Also, the pulser can be used in conjunction with the current tracer, as indicated in Figure 1–39. The current tracer, also known as a current probe, senses when there is pulsating current in a line and is particularly useful for locating shorts on a PC board.

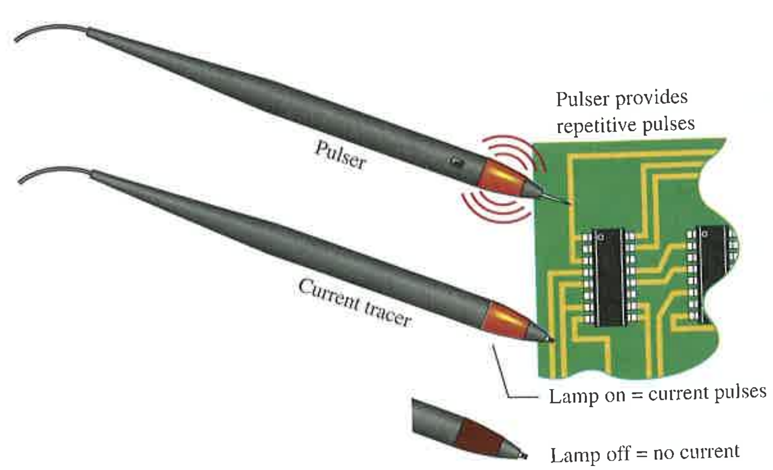


FIGURE 1–39

Illustration of how a logic pulser and a current tracer can be used to apply a pulse to a given point and check for resulting current in another part of the circuit.

The DC Power Supply

The dc power supply is an indispensable instrument on any test bench. The power supply converts ac power from the standard wall outlet into regulated dc voltage. All digital circuits require dc voltage to operate. For example, TTL circuits and many CMOS circuits require approximately +5 V. The power supply is used when a new circuit is breadboarded or when a PC board is pulled from a system for testing and is no longer operating from the internal system power supply. Typical test bench dc power supplies are shown in Figure 1–40.

FIGURE 1–40

Typical test instruments. (Copyright Tektronix, Inc. All rights reserved. Reproduced by permission.)

The Function Generator

The function generator is a versatile signal source that provides pulse waveforms, as well as sine wave and triangular waveforms. Many function generators have logic-compatible outputs to provide proper level waveforms as inputs to digital circuits in order to check the operation. One type of function generator is shown in Figure 1–40.

The Digital Multimeter

No test bench is complete without a digital multimeter (DMM). This instrument is used for measuring dc and ac voltage, dc and ac current, and resistance. Figure 1–40 shows typical test bench and handheld DMMs.

SECTION 1–7 REVIEW

1. How many major horizontal divisions are there on an oscilloscope screen?
2. How many major vertical divisions are there on an oscilloscope screen?
3. What type of instrument can display up to sixteen waveforms on its screen?
4. What is the purpose of a function generator?

DIGITAL SYSTEM APPLICATION

In this digital system application (DSA), a simplified system application of the logic elements and functions that were discussed in Section 1-4 is presented. It is important that an electronic technician or technologist understand how various digital functions can operate together as a total system to perform a specified task. It is also important to begin to think in terms of system-level operation because, in practice, a large part of your work will involve systems rather than individual functions. Of course, to understand systems, you must first understand the basic elements and functions that make up a system.

This DSA introduces you to the system concept. The example here will show you how logic functions can work together to perform a higher-level task and will get you started thinking at the system level. The specific system used here to illustrate the system concept serves as an instructional model and is not necessarily the approach that would be used in practice, although it could be. In modern industrial applications like the one discussed here, instruments known as programmable controllers are often used.

About the System

Let's imagine that an aspirin factory uses the system shown in the simplified block diagram of Figure 1-41 for automatically counting and bottling tablets. The tablets are fed into a large funnel-like hopper. The narrow neck of the funnel allows only one tablet at a time to fall into a bottle on the conveyor belt below.

The digital system controls the number of tablets going into each bottle and displays a continually updated total near the conveyor line as well as at a remote location in another part of the plant. This system utilizes all the basic logic functions that were introduced in Section 1-4, and its only purpose is to show you how these functions may be combined to achieve a desired result.

The general operation is as follows. An optical sensor at the bottom of the funnel neck detects each tablet that passes and produces an electrical pulse. This pulse goes to the counter and advances it by one count; thus, at any time during the filling of a bottle, the counter contains the binary representation of the number of tablets in the bottle. The binary count is transferred from the counter on parallel lines to the *B* input of the comparator (comp). A preset binary number equal to the number of tablets that are to go into each bottle is placed on the *A* input of the comparator. The preset number comes from the keypad and the associated circuits, which include the encoder, register A, and code converter A. When the desired number of tablets is entered on the keypad, it is encoded and then stored by parallel register A until a change in the quantity of tablets per bottle is required.

Suppose, for example, that each bottle is to hold fifty tablets. When the number in the counter reaches 50, the $A = B$ output of the comparator goes HIGH, indicating that the bottle is full.

The HIGH output of the comparator immediately closes the valve in the neck of the funnel to stop the flow of tablets, and at the same time it activates the conveyor to move the next bottle into place under the funnel. When the next bottle is positioned properly under the neck of the funnel, the conveyor control circuit produces a pulse that resets the counter to zero. The $A = B$ output of the comparator goes back LOW, opening the funnel valve to restart the flow of tablets.

In the display portion of the system, the number in the counter is transferred in parallel to the *A* input of the adder. The *B* input of the adder comes from parallel register B that holds the total number of tablets bottled, up through the last bottle filled. For example, if ten bottles have been filled and each bottle holds fifty tablets, register B contains the binary representation for 500. Then, when the next bottle has been filled, the binary number for 50 appears on the *A* input of the adder, and the binary number for 500 is on the *B* input. The adder produces a new sum of 550, which is stored in register B, replacing the previous sum of 500.

The binary number in register B is transferred in parallel to the code converter and decoder, which changes it from binary form to decimal form for display on a readout near the conveyor line. The binary number in the register is also transferred to a multiplexer (mux) so that it can be converted from parallel to series form and transmitted along a single line to a remote location some distance away. It is more economical to run a single line than to run several parallel lines when significant distances are involved, and speed of data transmission is not a factor in this application. At the remote location, the serial data are demultiplexed and sent to register C. From there the data are then decoded for display on the remote readout.

Keep in mind that this system is purely an instructional model and does not represent the ultimate or most efficient way to implement this hypothetical process. Although there are certainly other approaches, this particular approach has been used in order to illustrate all of the logic functions that were introduced in Section 1-4 and that will be covered in detail in future chapters. It shows you one application of the various functional devices at the system level and how they can be connected to accomplish a specific objective. You will see this system again in the next chapter.

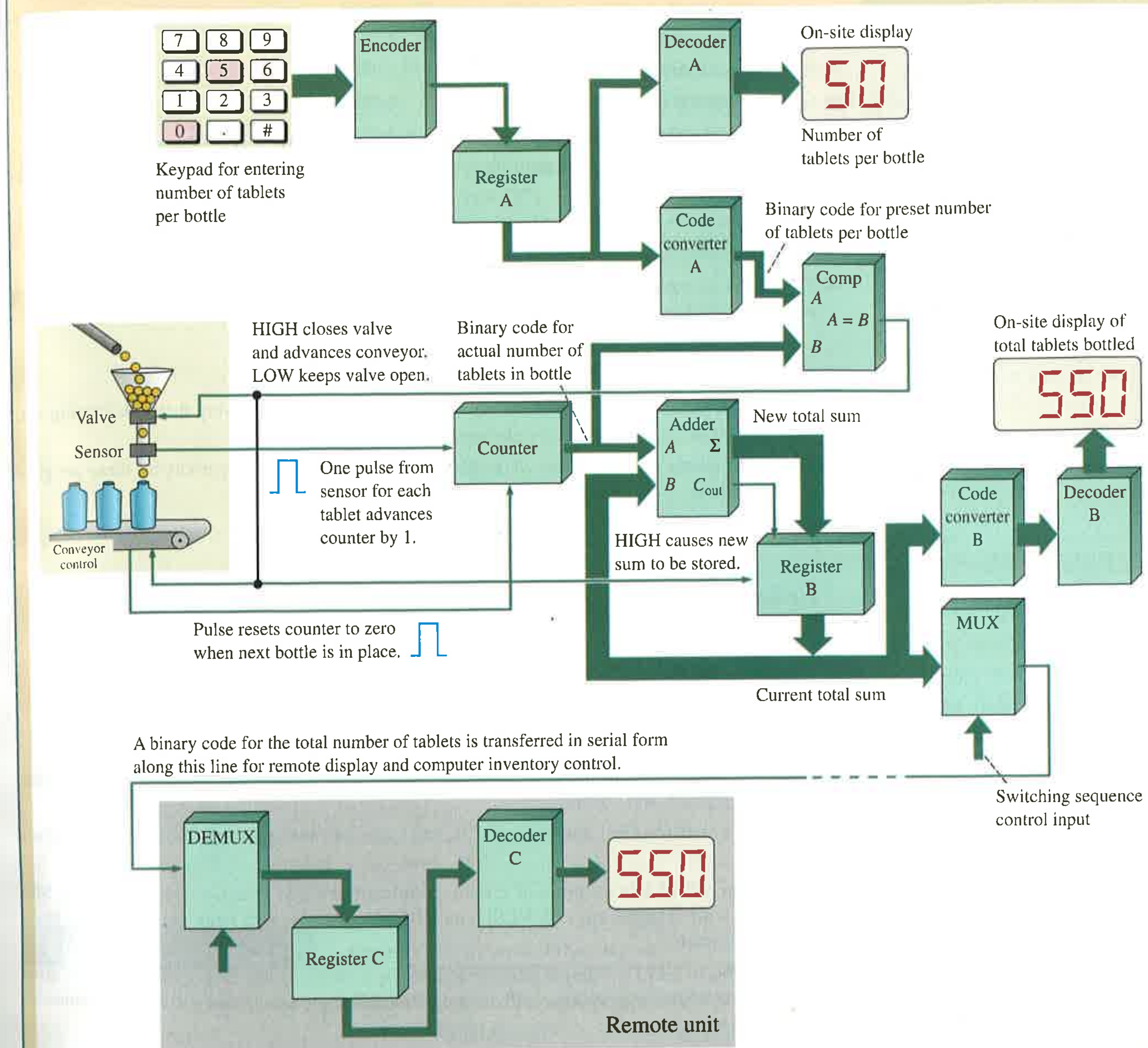


FIGURE 1-41

Simplified basic block diagram for a tablet-counting and bottling control system.

Digital System Application Review

1. Explain the purpose of the comparator (comp) in the system in Figure 1-41.
2. What actions take place when the $A = B$ output of the comparator goes HIGH?
3. What is the content of each register at any given time?

SUMMARY

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

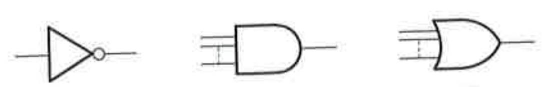
$$f = \frac{1}{T} \text{ and } T = \frac{1}{f}$$

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

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

- A timing diagram is an arrangement of two or more waveforms showing their relationship with respect to time.
- Three basic logic operations are NOT, AND, and OR. The standard symbols for these are given in Figure 1–42.

▶ FIGURE 1–42



- The basic logic functions are comparison, arithmetic, code conversion, decoding, encoding, data selection, storage, and counting.
- The two broad physical categories of IC packages are through-hole mounted and surface mounted.
- The categories of ICs in terms of circuit complexity are SSI (small-scale integration), MSI (medium-scale integration), LSI, VLSI, and ULSI (large-scale, very large-scale, and ultra large-scale integration).
- Basic types of SPLDs (simple programmable logic devices) are PAL (programmable array logic), GAL (generic array logic), PLA (programmable logic array), and PROM (programmable read-only memory).
- Common instruments used in testing and troubleshooting digital circuits are the oscilloscope, logic analyzer, logic probe, pulser, current tracer, dc power supply, function generator and digital multimeter.

KEY TERMS

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

Analog Being continuous or having continuous values.

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

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

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

Clock The basic timing signal in a digital system; a periodic waveform in which each interval between pulses equals the time for one bit.

Data Information in numeric, alphabetic, or other form.

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

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

Input The signal or line going into a circuit.

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

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

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

NOT A basic logic operation that performs inversions.

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

Output The signal or line coming out of a circuit.

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

PLD Programmable logic device.

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

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

Timing diagram A graph of digital waveforms showing time relationships of the waveforms.

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

SELF-TEST

Answers are at the end of the chapter.

1. A quantity having continuous values is
   (a) a digital quantity (b) an analog quantity
   (c) a binary quantity (d) a natural quantity
2. The term *bit* means
   (a) a small amount of data (b) a 1 or a 0
   (c) binary digit (d) both answers (b) and (c)
3. The time interval on the leading edge of a pulse between 10% and 90% of the amplitude is the
   (a) rise time (b) fall time (c) pulse width (d) period
4. A pulse in a certain waveform occurs every 10 ms. The frequency is
   (a) 1 kHz (b) 1 Hz (c) 100 Hz (d) 10 Hz
5. In a certain digital waveform, the period is twice the pulse width. The duty cycle is
   (a) 100% (b) 200% (c) 50%
6. An inverter
   (a) performs the NOT operation (b) changes a HIGH to a LOW
   (c) changes a LOW to a HIGH (d) does all of the above

7. The output of an AND gate is HIGH when
   (a) any input is HIGH (b) all inputs are HIGH
   (c) no inputs are HIGH (d) both answers (a) and (b)
8. The output of an OR gate is HIGH when
   (a) any input is HIGH (b) all inputs are HIGH
   (c) no inputs are HIGH (d) both answers (a) and (b)
9. The device used to convert a binary number to a 7-segment display format is
   (a) multiplexer (b) encoder (c) decoder (d) register
10. An example of a data storage device is
   (a) the logic gate (b) the flip-flop (c) the comparator
   (d) the register (e) both answers (b) and (d)
11. A fixed-function IC package containing four AND gates in an example of
   (a) MSI (b) SMT (c) SOIC (d) SSI
12. An LSI device has a circuit complexity of
   (a) 12 to 99 equivalent gates (b) 100 to 9999 equivalent gates
   (c) 2000 to 5000 equivalent gates (d) 10,000 to 99,999 equivalent gates
13. A CPLD is a
   (a) CMOS programmable logic device (b) Capacitive programmable logic device
   (c) Complex programmable logic device (d) Complementary process latching device
14. VHDL is a
   (a) logic device (b) PLD programming language
   (c) computer language (d) very high density logic

PROBLEMS

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

SECTION 1-1 Digital and Analog Quantities

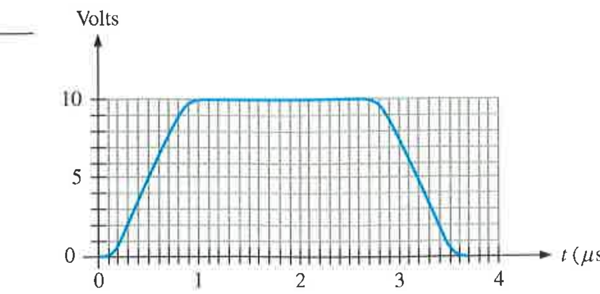
1. Name two advantages of digital data as compared to analog data.
2. Name an analog quantity other than temperature and sound.

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

3. Define the sequence of bits (1s and 0s) represented by each of the following sequences of levels:
   (a) HIGH, HIGH, LOW, HIGH, LOW, LOW, LOW, HIGH
   (b) LOW, LOW, LOW, HIGH, LOW, HIGH, LOW, HIGH, LOW
4. List the sequence of levels (HIGH and LOW) that represent each of the following bit sequences:
   (a) 1 0 1 1 1 0 1 (b) 1 1 1 0 1 0 0 1

5. For the pulse shown in Figure 1–43, graphically determine the following:
   (a) rise time (b) fall time (c) pulse width (d) amplitude

FIGURE 1–43



6. Determine the period of the digital waveform in Figure 1–44.
7. What is the frequency of the waveform in Figure 1–44?
8. Is the pulse waveform in Figure 1–44 periodic or nonperiodic?
9. Determine the duty cycle of the waveform in Figure 1–44.

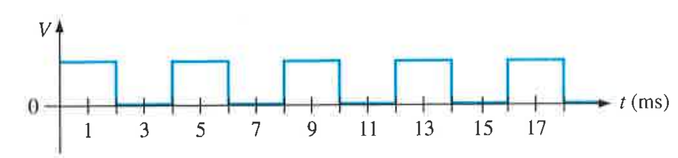


FIGURE 1–44

10. Determine the bit sequence represented by the waveform in Figure 1–45. A bit time is 1 μs this case.
11. What is the total serial transfer time for the eight bits in Figure 1–45? What is the total parallel transfer time?

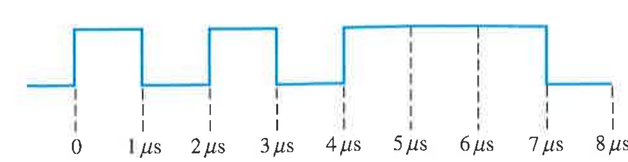


FIGURE 1–45

SECTION 1-3 **Introduction to Basic Logic Operations**

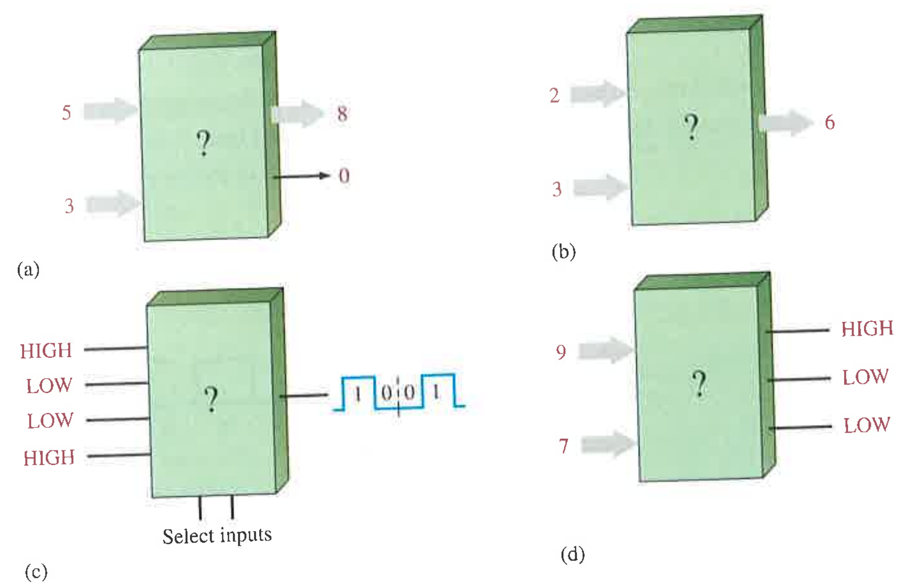
12. A logic circuit requires HIGHs on all its inputs to make the output HIGH. What type of logic circuit is it?

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

14. A basic 2-input logic circuit has a HIGH on one input and a LOW on the other input, and the output is HIGH. What type of logic circuit is it?

SECTION 1-4 **Basic Overview of Logic Functions**

15. Name the logic function of each block in Figure 1–46 based on your observation of the inputs and outputs.



▲ FIGURE 1–46

16. A pulse waveform with a frequency of 10 kHz is applied to the input of a counter. During 100 ms, how many pulses are counted?

17. Consider a register that can store eight bits. Assume that it has been reset so that it contains zeros in all positions. If you transfer four alternating bits (0101) serially into the register, beginning with a 1 and shifting to the right, what will the total content of the register be as soon as the fourth bit is stored?

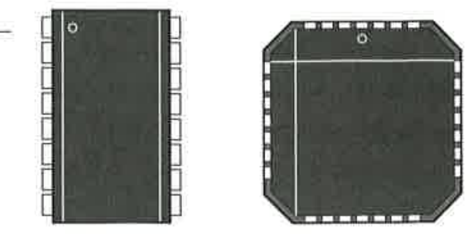
SECTION 1-5 **Fixed-Function Integrated Circuits**

18. A fixed-function digital IC chip has a complexity of 200 equivalent gates. How is it classified?

19. Explain the main difference between the DIP and SMT packages.

20. Label the pin numbers on the packages in Figure 1–47. Top views are shown.

► FIGURE 1–47



SECTION 1-6 **Programmable Logic Devices (PLDs)**

21. Which of the following acronyms do not describe a PLD?
PAL, GAL, SPLD, ABEL, CPLD, CUPL, EPLD, EEPLD, FPGA

22. What do each of the following stand for?
(a) SPLD **(b)** CPLD **(c)** HDL **(d)** FPGA **(e)** GAL

SECTION 1-7 **Introduction to Test Instruments**

23. A pulse is displayed on the screen of an oscilloscope, and you measure the base line as 1 V and the top of the pulse as 8 V. What is the amplitude?

24. A logic probe is applied to a contact point on an IC that is operating in a system. The lamp on the probe flashes repeatedly. What does this indicate?

Digital System Application

25. Define the term *system.*

26. In the system depicted in Figure 1–41, why are the multiplexer and demultiplexer necessary?

27. What action can be taken to change the number of tablets per bottle in the system of Figure 1–41?

ANSWERS

SECTION REVIEWS

SECTION 1-1 **Digital and Analog Quantities**

1. *Analog* means continuous.

2. *Digital* means discrete.

3. A digital quantity has a discrete set of values and an analog quantity has continuous values.

4. A public address system is analog. A CD player is analog and digital. A computer is all digital.

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

1. Binary means having two states or values.

2. A bit is a binary digit.

3. The bits are 1 and 0.

4. Rise time: from 10% to 90% of amplitude. Fall time: from 90% to 10% of amplitude.

5. Frequency is the reciprocal of the period.

6. A clock waveform is a basic timing waveform from which other waveforms are derived.

7. A timing diagram shows time relationships of waveforms.

8. Parallel transfer is faster than serial transfer.

SECTION 1-3 Introduction to Basic Logic Operations

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

SECTION 1-4 Basic Overview of Logic Functions

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

SECTION 1-5 Fixed-Function Integrated Circuits

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

SECTION 1-6 Programmable Logic Devices (PLDs)

1. Programmable logic device
2. Simple PLD
3. Complex PLD
4. Field-programmable gate array
5. A CPLD has a much higher logic capacity than an SPLD.
6. Schematic entry, text-based entry
7. Hardware description language
8. ABEL, CUPL
9. VHDL

SECTION 1-7 Introduction to Test Instruments

1. Ten
2. Eight
3. Logic analyzer
4. To provide pulse, side wave, and triangular waveforms.

Digital System Application

1. The comparator determines when the tablet count reaches the preset number of tablets per bottle.
2. The dispenser valve is closed, the next bottle is moved into place by the conveyor, and the new sum is stored in register B.
3. Register A stores the preset number of tablets per bottle. Register B stores the total number of tablets bottled.

RELATED PROBLEMS FOR EXAMPLES

1–1 $f = 6.67$ kHz; Duty cycle = 16.7%

1–2 Parallel transfer: 100 ns; Serial transfer: 1.6 μs

1–3 Amplitude = 12 V; $T = 8$ ms

SELF-TEST

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