

# Logic Gates

## CHAPTER OUTLINE

- 3-1 The Inverter
- 3-2 The AND Gate
- 3-3 The OR Gate
- 3-4 The NAND Gate
- 3-5 The NOR Gate
- 3-6 The Exclusive-OR and Exclusive-NOR Gates
- 3-7 Programmable Logic
- 3-8 Fixed-Function Logic Gates
- 3-9 Troubleshooting

## CHAPTER OBJECTIVES

- Describe the operation of the inverter, the AND gate, and the OR gate
- Describe the operation of the NAND gate and the NOR gate
- Express the operation of NOT, AND, OR, NAND, and NOR gates with Boolean algebra
- Describe the operation of the exclusive-OR and exclusive-NOR gates
- Use logic gates in simple applications
- Recognize and use both the distinctive shape logic gate symbols and the rectangular outline logic gate symbols of ANSI/IEEE Standard 91-1984/Std. 91a-1991
- Construct timing diagrams showing the proper time relationships of inputs and outputs for the various logic gates
- Discuss the basic concepts of programmable logic
- Make basic comparisons between the major IC technologies—CMOS and bipolar (TTL)
- Explain how the different series within the CMOS and bipolar (TTL) families differ from each other
- Define *propagation delay time*, *power dissipation*, *speed-power product*, and *fan-out* in relation to logic gates

- List specific fixed-function integrated circuit devices that contain the various logic gates
- Troubleshoot logic gates for opens and shorts by using the oscilloscope

## KEY TERMS

Key terms are in order of appearance in the chapter.

- |                      |                          |
|----------------------|--------------------------|
| ■ Inverter           | ■ EPROM                  |
| ■ Truth table        | ■ EEPROM                 |
| ■ Boolean algebra    | ■ Flash                  |
| ■ Complement         | ■ SRAM                   |
| ■ AND gate           | ■ Target device          |
| ■ OR gate            | ■ JTAG                   |
| ■ NAND gate          | ■ VHDL                   |
| ■ NOR gate           | ■ CMOS                   |
| ■ Exclusive-OR gate  | ■ Bipolar                |
| ■ Exclusive-NOR gate | ■ Propagation delay time |
| ■ AND array          | ■ Fan-out                |
| ■ Fuse               | ■ Unit load              |
| ■ Antifuse           |                          |

## VISIT THE WEBSITE

Study aids for this chapter are available at <http://www.pearsonglobaleditions.com/floyd>

## INTRODUCTION

The emphasis in this chapter is on the operation, application, and troubleshooting of logic gates. The relationship of input and output waveforms of a gate using timing diagrams is thoroughly covered.

Logic symbols used to represent the logic gates are in accordance with ANSI/IEEE Standard 91-1984/Std. 91a-1991. This standard has been adopted by private industry and the military for use in internal documentation as well as published literature.

Both fixed-function logic and programmable logic are discussed in this chapter. Because integrated circuits (ICs) are used in all applications, the logic function of a device is generally of greater importance to the technician or technologist than the details of the component-level circuit operation within the IC package. Therefore, detailed cover-

age of the devices at the component level can be treated as an optional topic. Digital integrated circuit technologies are discussed in Chapter 15 on the website, all or parts of which may be introduced at appropriate points throughout the text.

*Suggestion:* Review Section 1–3 before you start this chapter.

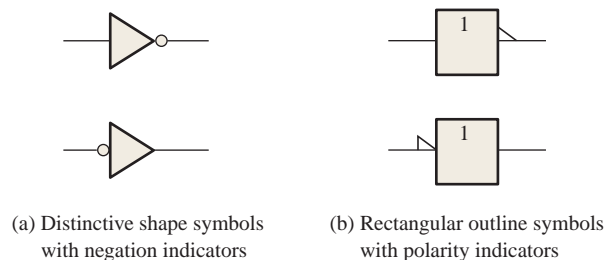
## 3–1 The Inverter

The inverter (NOT circuit) performs the operation called *inversion* or *complementation*. The inverter changes one logic level to the opposite level. In terms of bits, it changes a 1 to a 0 and a 0 to a 1.

After completing this section, you should be able to

- ◆ Identify negation and polarity indicators
- ◆ Identify an inverter by either its distinctive shape symbol or its rectangular outline symbol
- ◆ Produce the truth table for an inverter
- ◆ Describe the logical operation of an inverter

Standard logic symbols for the **inverter** are shown in Figure 3–1. Part (a) shows the *distinctive shape* symbols, and part (b) shows the *rectangular outline* symbols. In this textbook, distinctive shape symbols are generally used; however, the rectangular outline symbols are found in many industry publications, and you should become familiar with them as well. (Logic symbols are in accordance with ANSI/IEEE Standard 91-1984 and its supplement Standard 91a-1991.)



**FIGURE 3–1** Standard logic symbols for the inverter (ANSI/IEEE Std. 91-1984/Std. 91a-1991).

### The Negation and Polarity Indicators

The negation indicator is a “bubble” (○) that indicates **inversion** or *complementation* when it appears on the input or output of any logic element, as shown in Figure 3–1(a) for the inverter. Generally, inputs are on the left of a logic symbol and the output is on the right. When appearing on the input, the bubble means that a 0 is the active or *asserted* input state, and the input is called an active-LOW input. When appearing on the output, the bubble means that a 0 is the active or asserted output state, and the output is called an active-LOW output. The absence of a bubble on the input or output means that a 1 is the active or asserted state, and in this case, the input or output is called active-HIGH.

The polarity or level indicator is a “triangle” ( $\blacktriangle$ ) that indicates inversion when it appears on the input or output of a logic element, as shown in Figure 3–1(b). When appearing on the input, it means that a LOW level is the active or asserted input state. When appearing on the output, it means that a LOW level is the active or asserted output state.

Either indicator (bubble or triangle) can be used both on distinctive shape symbols and on rectangular outline symbols. Figure 3–1(a) indicates the principal inverter symbols used in this text. Note that a change in the placement of the negation or polarity indicator does not imply a change in the way an inverter operates.

### Inverter Truth Table

When a HIGH level is applied to an inverter input, a LOW level will appear on its output. When a LOW level is applied to its input, a HIGH will appear on its output. This operation is summarized in Table 3–1, which shows the output for each possible input in terms of levels and corresponding bits. A table such as this is called a **truth table**.

**TABLE 3–1**

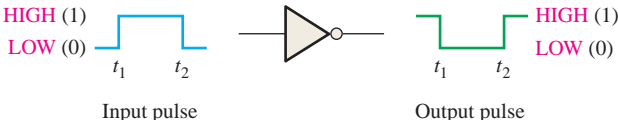
Inverter truth table.

| Input    | Output   |
|----------|----------|
| LOW (0)  | HIGH (1) |
| HIGH (1) | LOW (0)  |

### Inverter Operation

Figure 3–2 shows the output of an inverter for a pulse input, where  $t_1$  and  $t_2$  indicate the corresponding points on the input and output pulse waveforms.

**When the input is LOW, the output is HIGH; when the input is HIGH, the output is LOW, thereby producing an inverted output pulse.**



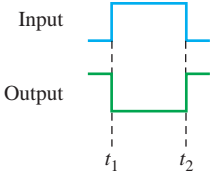
**FIGURE 3–2** Inverter operation with a pulse input. Open file F03-02 to verify inverter operation. A *Multisim* tutorial is available on the website.



### Timing Diagrams

Recall from Chapter 1 that a *timing diagram* is basically a graph that accurately displays the relationship of two or more waveforms with respect to each other on a time basis. For example, the time relationship of the output pulse to the input pulse in Figure 3–2 can be shown with a simple timing diagram by aligning the two pulses so that the occurrences of the pulse edges appear in the proper time relationship. The rising edge of the input pulse and the falling edge of the output pulse occur at the same time (ideally). Similarly, the falling edge of the input pulse and the rising edge of the output pulse occur at the same time (ideally). This timing relationship is shown in Figure 3–3. In practice, there is a very small delay from the input transition until the corresponding output transition. Timing diagrams are especially useful for illustrating the time relationship of digital waveforms with multiple pulses.

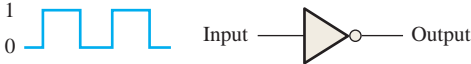
A timing diagram shows how two or more waveforms relate in time.



**FIGURE 3–3** Timing diagram for the case in Figure 3–2.

### EXAMPLE 3–1

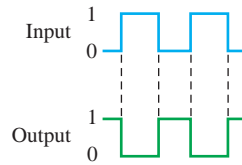
A waveform is applied to an inverter in Figure 3–4. Determine the output waveform corresponding to the input and show the timing diagram. According to the placement of the bubble, what is the active output state?



**FIGURE 3–4**

**Solution**

The output waveform is exactly opposite to the input (inverted), as shown in Figure 3–5, which is the basic timing diagram. The active or asserted output state is **0**.



**FIGURE 3-5**

**Related Problem\***

If the inverter is shown with the negative indicator (bubble) on the input instead of the output, how is the timing diagram affected?

\*Answers are at the end of the chapter.

Boolean algebra uses variables and operators to describe a logic circuit.

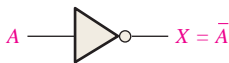
**Logic Expression for an Inverter**

In **Boolean algebra**, which is the mathematics of logic circuits and will be covered thoroughly in Chapter 4, a variable is generally designated by one or two letters although there can be more. Letters near the beginning of the alphabet usually designate inputs, while letters near the end of the alphabet usually designate outputs. The **complement** of a variable is designated by a bar over the letter. A variable can take on a value of either 1 or 0. If a given variable is 1, its complement is 0 and vice versa.

The operation of an inverter (NOT circuit) can be expressed as follows: If the input variable is called *A* and the output variable is called *X*, then

$$X = \bar{A}$$

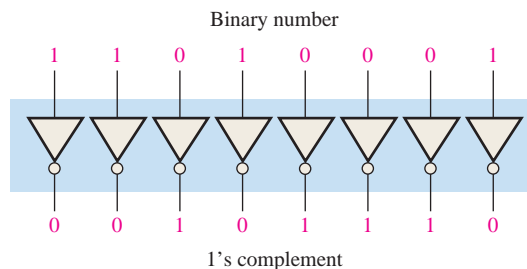
This expression states that the output is the complement of the input, so if  $A = 0$ , then  $X = 1$ , and if  $A = 1$ , then  $X = 0$ . Figure 3–6 illustrates this. The complemented variable  $\bar{A}$  can be read as “A bar” or “not A.”



**FIGURE 3-6** The inverter complements an input variable.

**An Application**

Figure 3–7 shows a circuit for producing the 1’s complement of an 8-bit binary number. The bits of the binary number are applied to the inverter inputs and the 1’s complement of the number appears on the outputs.



**FIGURE 3-7** Example of a 1’s complement circuit using inverters.

**SECTION 3-1 CHECKUP**

Answers are at the end of the chapter.

1. When a 1 is on the input of an inverter, what is the output?
2. An active-HIGH pulse (HIGH level when asserted, LOW level when not) is required on an inverter input.
  - (a) Draw the appropriate logic symbol, using the distinctive shape and the negation indicator, for the inverter in this application.
  - (b) Describe the output when a positive-going pulse is applied to the input of an inverter.

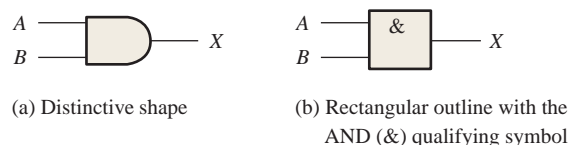
**3-2 The AND Gate**

The AND gate is one of the basic gates that can be combined to form any logic function. An AND gate can have two or more inputs and performs what is known as logical multiplication.

After completing this section, you should be able to

- ◆ Identify an AND gate by its distinctive shape symbol or by its rectangular outline symbol
- ◆ Describe the operation of an AND gate
- ◆ Generate the truth table for an AND gate with any number of inputs
- ◆ Produce a timing diagram for an AND gate with any specified input waveforms
- ◆ Write the logic expression for an AND gate with any number of inputs
- ◆ Discuss examples of AND gate applications

The term *gate* was introduced in Chapter 1 and is used to describe a circuit that performs a basic logic operation. The AND gate is composed of two or more inputs and a single output, as indicated by the standard logic symbols shown in Figure 3-8. Inputs are on the left, and the output is on the right in each symbol. Gates with two inputs are shown; however, an AND gate can have any number of inputs greater than one. Although examples of both distinctive shape symbols and rectangular outline symbols are shown, the distinctive shape symbol, shown in part (a), is used predominantly in this book.



**FIGURE 3-8** Standard logic symbols for the AND gate showing two inputs (ANSI/IEEE Std. 91-1984/Std. 91a-1991).

**Operation of an AND Gate**

An **AND gate** produces a HIGH output *only* when *all* of the inputs are HIGH. When any of the inputs is LOW, the output is LOW. Therefore, the basic purpose of an AND gate is to determine when certain conditions are simultaneously true, as indicated by HIGH levels on all of its inputs, and to produce a HIGH on its output to indicate that all these conditions are

**InfoNote**

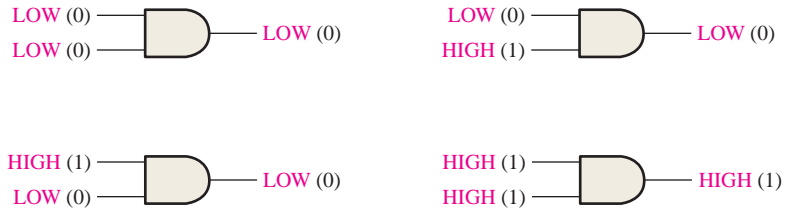
Logic gates are one of the fundamental building blocks of digital systems. Most of the functions in a computer, with the exception of certain types of memory, are implemented with logic gates used on a very large scale. For example, a microprocessor, which is the main part of a computer, is made up of hundreds of thousands or even millions of logic gates.

An AND gate can have more than two inputs.

true. The inputs of the 2-input AND gate in Figure 3–8 are labeled *A* and *B*, and the output is labeled *X*. The gate operation can be stated as follows:

**For a 2-input AND gate, output *X* is HIGH only when inputs *A* and *B* are HIGH; *X* is LOW when either *A* or *B* is LOW, or when both *A* and *B* are LOW.**

Figure 3–9 illustrates a 2-input AND gate with all four possibilities of input combinations and the resulting output for each.



**FIGURE 3–9** All possible logic levels for a 2-input AND gate. Open file F03-09 to verify AND gate operation.

### AND Gate Truth Table

The logical operation of a gate can be expressed with a truth table that lists all input combinations with the corresponding outputs, as illustrated in Table 3–2 for a 2-input AND gate. The truth table can be expanded to any number of inputs. Although the terms HIGH and LOW tend to give a “physical” sense to the input and output states, the truth table is shown with 1s and 0s; a HIGH is equivalent to a 1 and a LOW is equivalent to a 0 in positive logic. For any AND gate, regardless of the number of inputs, the output is HIGH *only* when *all* inputs are HIGH.

The total number of possible combinations of binary inputs to a gate is determined by the following formula:

$$N = 2^n \quad \text{Equation 3–1}$$

where *N* is the number of possible input combinations and *n* is the number of input variables. To illustrate,

- For two input variables:  $N = 2^2 = 4$  combinations
- For three input variables:  $N = 2^3 = 8$  combinations
- For four input variables:  $N = 2^4 = 16$  combinations

You can determine the number of input bit combinations for gates with any number of inputs by using Equation 3–1.

For an AND gate, all HIGH inputs produce a HIGH output.

**TABLE 3–2**

Truth table for a 2-input AND gate.

| Inputs   |          | Output   |
|----------|----------|----------|
| <i>A</i> | <i>B</i> | <i>X</i> |
| 0        | 0        | 0        |
| 0        | 1        | 0        |
| 1        | 0        | 0        |
| 1        | 1        | 1        |

1 = HIGH, 0 = LOW

### EXAMPLE 3–2

**TABLE 3–3**

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

- (a) Develop the truth table for a 3-input AND gate.
- (b) Determine the total number of possible input combinations for a 4-input AND gate.

**Solution**

- (a) There are eight possible input combinations ( $2^3 = 8$ ) for a 3-input AND gate. The input side of the truth table (Table 3–3) shows all eight combinations of three bits. The output side is all 0s except when all three input bits are 1s.
- (b)  $N = 2^4 = 16$ . There are 16 possible combinations of input bits for a 4-input AND gate.

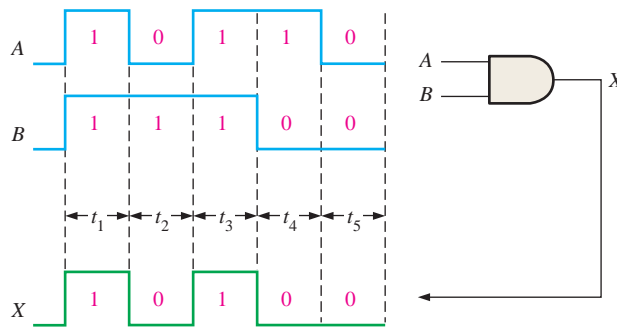
**Related Problem**

Develop the truth table for a 4-input AND gate.

## AND Gate Operation with Waveform Inputs

In most applications, the inputs to a gate are not stationary levels but are voltage waveforms that change frequently between HIGH and LOW logic levels. Now let's look at the operation of AND gates with pulse waveform inputs, keeping in mind that an AND gate obeys the truth table operation regardless of whether its inputs are constant levels or levels that change back and forth.

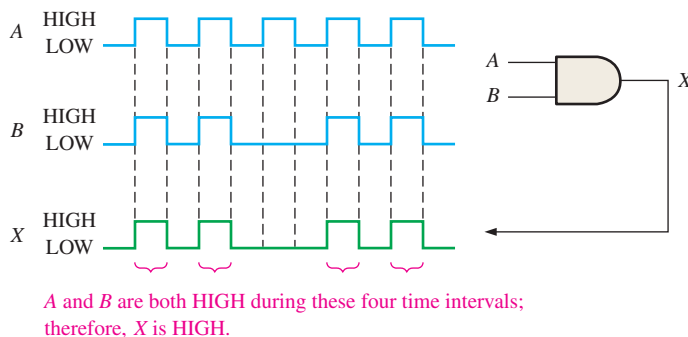
Let's examine the waveform operation of an AND gate by looking at the inputs with respect to each other in order to determine the output level at any given time. In Figure 3–10, inputs  $A$  and  $B$  are both HIGH (1) during the time interval,  $t_1$ , making output  $X$  HIGH (1) during this interval. During time interval  $t_2$ , input  $A$  is LOW (0) and input  $B$  is HIGH (1), so the output is LOW (0). During time interval  $t_3$ , both inputs are HIGH (1) again, and therefore the output is HIGH (1). During time interval  $t_4$ , input  $A$  is HIGH (1) and input  $B$  is LOW (0), resulting in a LOW (0) output. Finally, during time interval  $t_5$ , input  $A$  is LOW (0), input  $B$  is LOW (0), and the output is therefore LOW (0). As you know, a diagram of input and output waveforms showing time relationships is called a *timing diagram*.



**FIGURE 3-10** Example of AND gate operation with a timing diagram showing input and output relationships.

### EXAMPLE 3-3

If two waveforms,  $A$  and  $B$ , are applied to the AND gate inputs as in Figure 3–11, what is the resulting output waveform?



**FIGURE 3-11**

### Solution

The output waveform  $X$  is HIGH only when both  $A$  and  $B$  waveforms are HIGH as shown in the timing diagram in Figure 3–11.

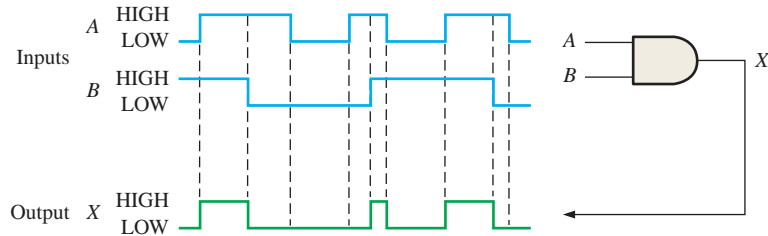
### Related Problem

Determine the output waveform and show a timing diagram if the second and fourth pulses in waveform  $A$  of Figure 3–11 are replaced by LOW levels.

Remember, when analyzing the waveform operation of logic gates, it is important to pay careful attention to the time relationships of all the inputs with respect to each other and to the output.

**EXAMPLE 3-4**

For the two input waveforms, *A* and *B*, in Figure 3-12, show the output waveform with its proper relation to the inputs.



**FIGURE 3-12**

**Solution**

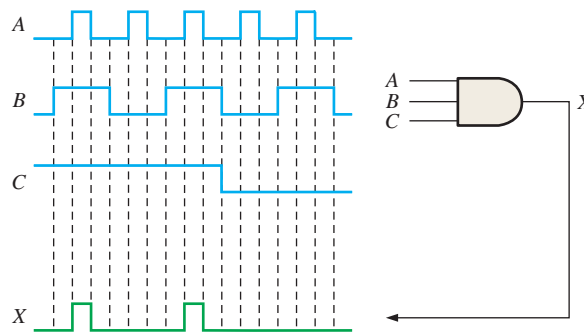
The output waveform is HIGH only when both of the input waveforms are HIGH as shown in the timing diagram.

**Related Problem**

Show the output waveform if the *B* input to the AND gate in Figure 3-12 is always HIGH.

**EXAMPLE 3-5**

For the 3-input AND gate in Figure 3-13, determine the output waveform in relation to the inputs.



**FIGURE 3-13**

**Solution**

The output waveform *X* of the 3-input AND gate is HIGH only when all three input waveforms *A*, *B*, and *C* are HIGH.

**Related Problem**

What is the output waveform of the AND gate in Figure 3-13 if the *C* input is always HIGH?

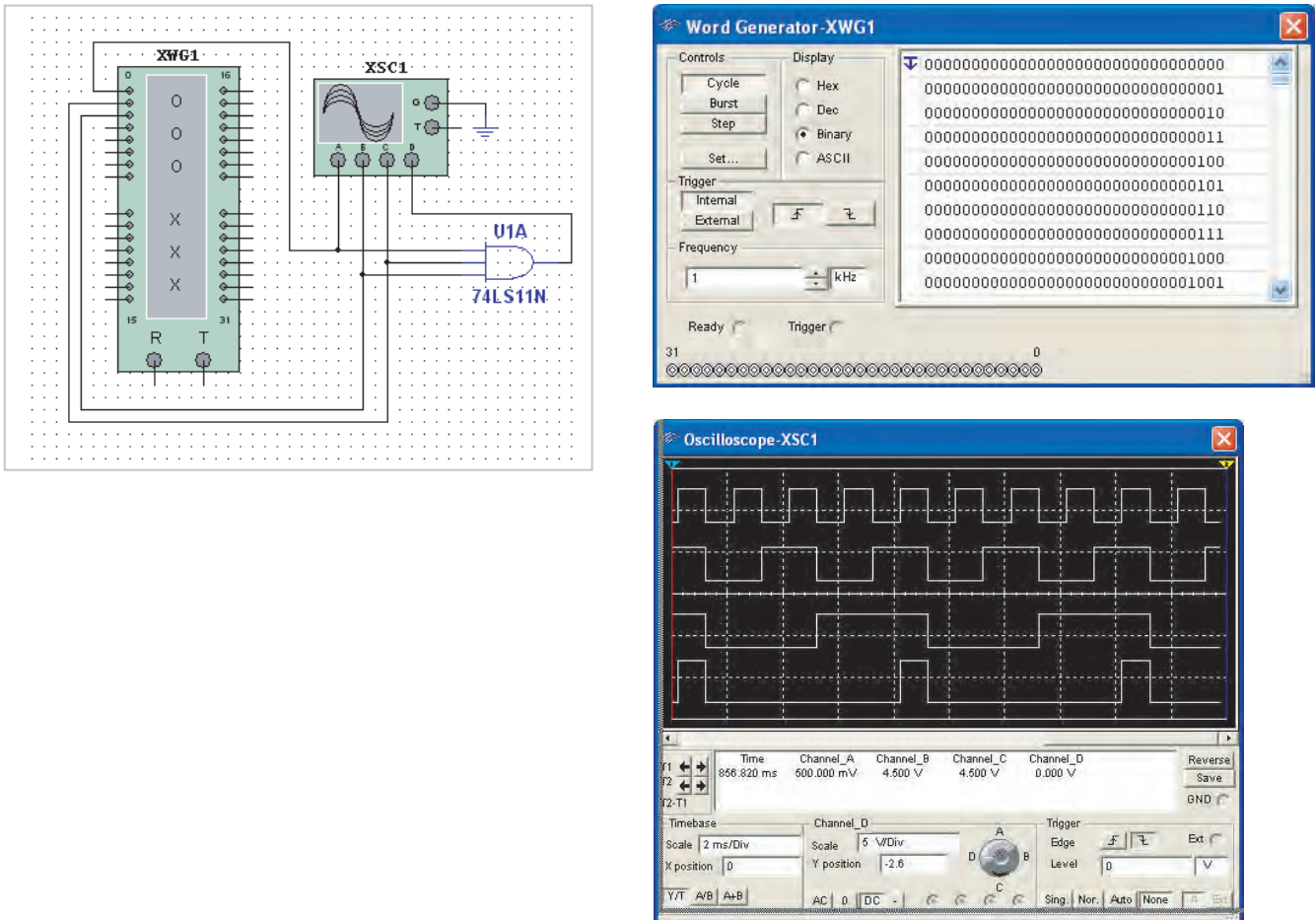


**EXAMPLE 3-6**

Use Multisim to simulate a 3-input AND gate with input waveforms that cycle through binary numbers 0 through 9.

**Solution**

Use the Multisim word generator in the up counter mode to provide the combination of waveforms representing the binary sequence, as shown in Figure 3-14. The first three waveforms on the oscilloscope display are the inputs, and the bottom waveform is the output.



**FIGURE 3-14**

**Related Problem**

Use Multisim software to create the setup and simulate the 3-input AND gate as illustrated in this example.



**Logic Expressions for an AND Gate**

The logical AND function of two variables is represented mathematically either by placing a dot between the two variables, as  $A \cdot B$ , or by simply writing the adjacent letters without the dot, as  $AB$ . We will normally use the latter notation.

**InfoNote**

Processors can utilize all of the basic logic operations when it is necessary to selectively manipulate certain bits in one or more bytes of data. Selective bit manipulations are done with a *mask*. For example, to clear (make all 0s) the right four bits in a data byte but keep the left four bits, ANDing the data byte with 11110000 will do the job. Notice that any bit ANDed with zero will be 0 and any bit ANDed with 1 will remain the same. If 10101010 is ANDed with the mask 11110000, the result is 10100000.

When variables are shown together like *ABC*, they are ANDed.

**Boolean multiplication** follows the same basic rules governing binary multiplication, which were discussed in Chapter 2 and are as follows:

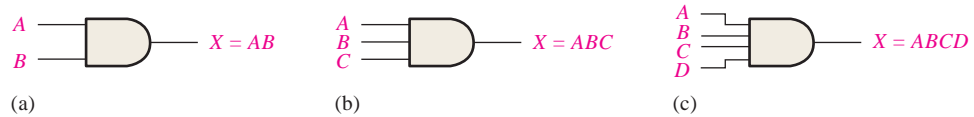
$$\begin{aligned} 0 \cdot 0 &= 0 \\ 0 \cdot 1 &= 0 \\ 1 \cdot 0 &= 0 \\ 1 \cdot 1 &= 1 \end{aligned}$$

**Boolean multiplication is the same as the AND function.**

The operation of a 2-input AND gate can be expressed in equation form as follows: If one input variable is *A*, if the other input variable is *B*, and if the output variable is *X*, then the Boolean expression is

$$X = AB$$

Figure 3–15(a) shows the AND gate logic symbol with two input variables and the output variable indicated.



**FIGURE 3-15** Boolean expressions for AND gates with two, three, and four inputs.

To extend the AND expression to more than two input variables, simply use a new letter for each input variable. The function of a 3-input AND gate, for example, can be expressed as  $X = ABC$ , where *A*, *B*, and *C* are the input variables. The expression for a 4-input AND gate can be  $X = ABCD$ , and so on. Parts (b) and (c) of Figure 3–15 show AND gates with three and four input variables, respectively.

You can evaluate an AND gate operation by using the Boolean expressions for the output. For example, each variable on the inputs can be either a 1 or a 0; so for the 2-input AND gate, make substitutions in the equation for the output,  $X = AB$ , as shown in Table 3–4. This evaluation shows that the output *X* of an AND gate is a 1 (HIGH) only when both inputs are 1s (HIGHS). A similar analysis can be made for any number of input variables.

**TABLE 3-4**

| <i>A</i> | <i>B</i> | $AB = X$        |
|----------|----------|-----------------|
| 0        | 0        | $0 \cdot 0 = 0$ |
| 0        | 1        | $0 \cdot 1 = 0$ |
| 1        | 0        | $1 \cdot 0 = 0$ |
| 1        | 1        | $1 \cdot 1 = 1$ |

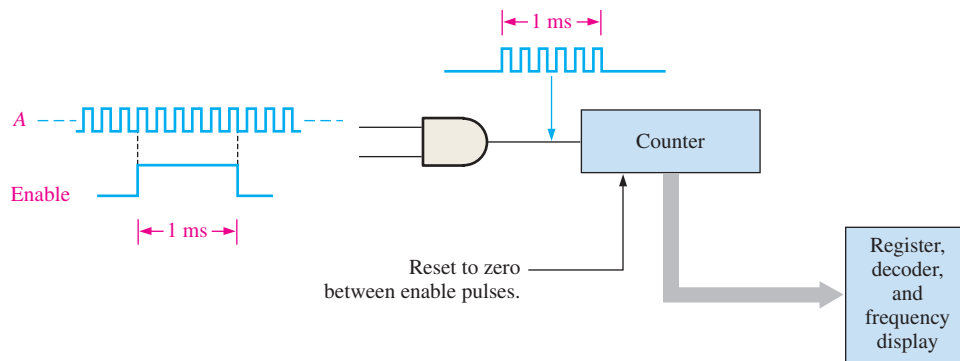
**Applications**

**The AND Gate as an Enable/Inhibit Device**

A common application of the AND gate is to **enable** (that is, to allow) the passage of a signal (pulse waveform) from one point to another at certain times and to **inhibit** (prevent) the passage at other times.

A simple example of this particular use of an AND gate is shown in Figure 3–16, where the AND gate controls the passage of a signal (waveform *A*) to a digital counter. The purpose of this circuit is to measure the frequency of waveform *A*. The enable pulse has a width of precisely 1 ms. When the enable pulse is HIGH, waveform *A* passes through the gate to the counter; and when the enable pulse is LOW, the signal is prevented from passing through the gate (inhibited).

During the 1 millisecond (1 ms) interval of the enable pulse, pulses in waveform *A* pass through the AND gate to the counter. The number of pulses passing through during the 1 ms interval is equal to the frequency of waveform *A*. For example, Figure 3–16 shows six pulses in one millisecond, which is a frequency of 6 kHz. If 1000 pulses pass through the gate in the 1 ms interval of the enable pulse, there are 1000 pulses/ms, or a frequency of 1 MHz.

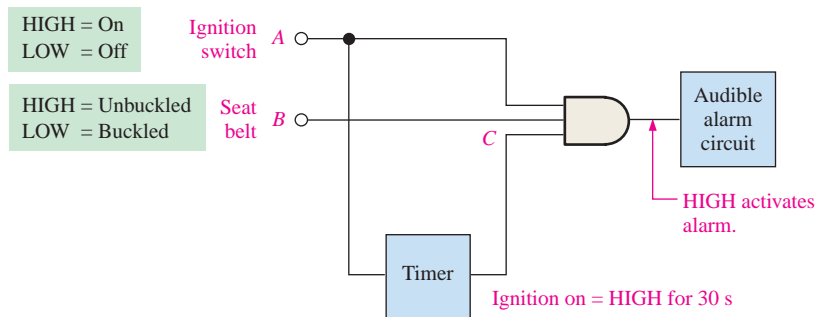


**FIGURE 3-16** An AND gate performing an enable/inhibit function for a frequency counter.

The counter counts the number of pulses per second and produces a binary output that goes to a decoding and display circuit to produce a readout of the frequency. The enable pulse repeats at certain intervals and a new updated count is made so that if the frequency changes, the new value will be displayed. Between enable pulses, the counter is reset so that it starts at zero each time an enable pulse occurs. The current frequency count is stored in a register so that the display is unaffected by the resetting of the counter.

### A Seat Belt Alarm System

In Figure 3-17, an AND gate is used in a simple automobile seat belt alarm system to detect when the ignition switch is on *and* the seat belt is unbuckled. If the ignition switch is on, a HIGH is produced on input *A* of the AND gate. If the seat belt is not properly buckled, a HIGH is produced on input *B* of the AND gate. Also, when the ignition switch is turned on, a timer is started that produces a HIGH on input *C* for 30 s. If all three conditions exist—that is, if the ignition is on *and* the seat belt is unbuckled *and* the timer is running—the output of the AND gate is HIGH, and an audible alarm is energized to remind the driver.



**FIGURE 3-17** A simple seat belt alarm circuit using an AND gate.

#### SECTION 3-2 CHECKUP

1. When is the output of an AND gate HIGH?
2. When is the output of an AND gate LOW?
3. Describe the truth table for a 5-input AND gate.

### 3-3 The OR Gate

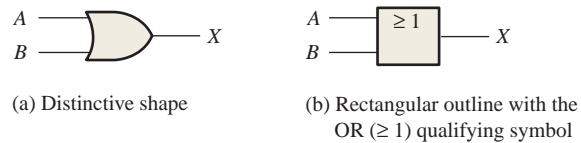
The OR gate is another of the basic gates from which all logic functions are constructed. An OR gate can have two or more inputs and performs what is known as logical addition.

After completing this section, you should be able to

- ◆ Identify an OR gate by its distinctive shape symbol or by its rectangular outline symbol
- ◆ Describe the operation of an OR gate
- ◆ Generate the truth table for an OR gate with any number of inputs
- ◆ Produce a timing diagram for an OR gate with any specified input waveforms
- ◆ Write the logic expression for an OR gate with any number of inputs
- ◆ Discuss an OR gate application

An OR gate can have more than two inputs.

An **OR gate** has two or more inputs and one output, as indicated by the standard logic symbols in Figure 3-18, where OR gates with two inputs are illustrated. An OR gate can have any number of inputs greater than one. Although both distinctive shape and rectangular outline symbols are shown, the distinctive shape OR gate symbol is used in this textbook.



**FIGURE 3-18** Standard logic symbols for the OR gate showing two inputs (ANSI/IEEE Std. 91-1984/Std. 91a-1991).

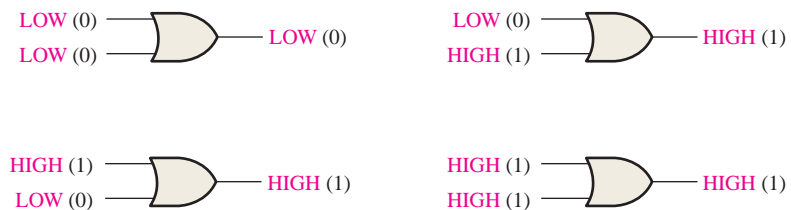
#### Operation of an OR Gate

For an OR gate, at least one HIGH input produces a HIGH output.

An OR gate produces a HIGH on the output when *any* of the inputs is HIGH. The output is LOW only when all of the inputs are LOW. Therefore, an OR gate determines when one or more of its inputs are HIGH and produces a HIGH on its output to indicate this condition. The inputs of the 2-input OR gate in Figure 3-18 are labeled *A* and *B*, and the output is labeled *X*. The operation of the gate can be stated as follows:

**For a 2-input OR gate, output *X* is HIGH when either input *A* or input *B* is HIGH, or when both *A* and *B* are HIGH; *X* is LOW only when both *A* and *B* are LOW.**

The HIGH level is the active or asserted output level for the OR gate. Figure 3-19 illustrates the operation for a 2-input OR gate for all four possible input combinations.



**FIGURE 3-19** All possible logic levels for a 2-input OR gate. Open file F03-19 to verify OR gate operation.

## OR Gate Truth Table

The operation of a 2-input OR gate is described in Table 3–5. This truth table can be expanded for any number of inputs; but regardless of the number of inputs, the output is HIGH when one or more of the inputs are HIGH.

**TABLE 3–5**

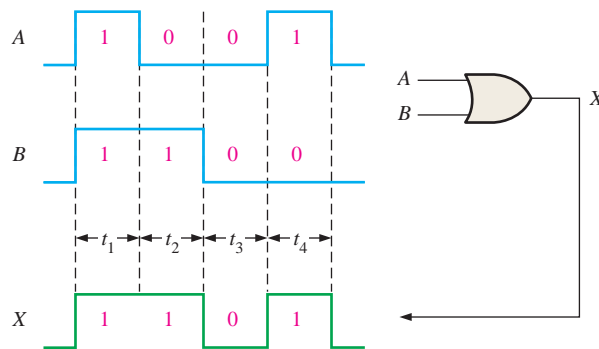
Truth table for a 2-input OR gate.

| Inputs |   | Output |
|--------|---|--------|
| A      | B | X      |
| 0      | 0 | 0      |
| 0      | 1 | 1      |
| 1      | 0 | 1      |
| 1      | 1 | 1      |

1 = HIGH, 0 = LOW

## OR Gate Operation with Waveform Inputs

Now let's look at the operation of an OR gate with pulse waveform inputs, keeping in mind its logical operation. Again, the important thing in the analysis of gate operation with pulse waveforms is the time relationship of all the waveforms involved. For example, in Figure 3–20, inputs *A* and *B* are both HIGH (1) during time interval  $t_1$ , making output *X* HIGH (1). During time interval  $t_2$ , input *A* is LOW (0), but because input *B* is HIGH (1), the output is HIGH (1). Both inputs are LOW (0) during time interval  $t_3$ , so there is a LOW (0) output during this time. During time interval  $t_4$ , the output is HIGH (1) because input *A* is HIGH (1).

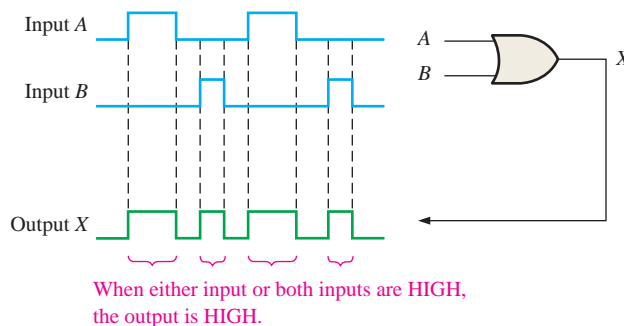


**FIGURE 3–20** Example of OR gate operation with a timing diagram showing input and output time relationships.

In this illustration, we have applied the truth table operation of the OR gate to each of the time intervals during which the levels are nonchanging. Examples 3–7 through 3–9 further illustrate OR gate operation with waveforms on the inputs.

### EXAMPLE 3–7

If the two input waveforms, *A* and *B*, in Figure 3–21 are applied to the OR gate, what is the resulting output waveform?



**FIGURE 3–21**

**Solution**

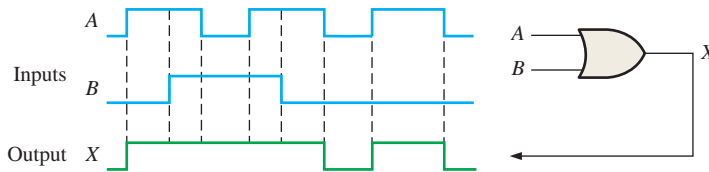
The output waveform  $X$  of a 2-input OR gate is HIGH when either or both input waveforms are HIGH as shown in the timing diagram. In this case, both input waveforms are never HIGH at the same time.

**Related Problem**

Determine the output waveform and show the timing diagram if input  $A$  is changed such that it is HIGH from the beginning of the existing first pulse to the end of the existing second pulse.

**EXAMPLE 3-8**

For the two input waveforms,  $A$  and  $B$ , in Figure 3-22, show the output waveform with its proper relation to the inputs.



**FIGURE 3-22**

**Solution**

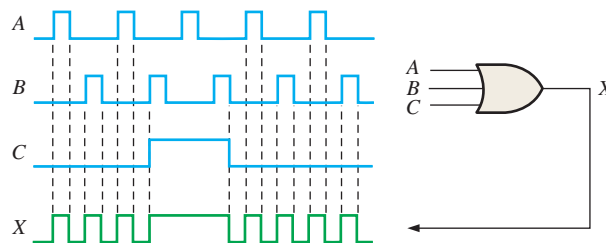
When either or both input waveforms are HIGH, the output is HIGH as shown by the output waveform  $X$  in the timing diagram.

**Related Problem**

Determine the output waveform and show the timing diagram if the middle pulse of input  $A$  is replaced by a LOW level.

**EXAMPLE 3-9**

For the 3-input OR gate in Figure 3-23, determine the output waveform in proper time relation to the inputs.



**FIGURE 3-23**

**Solution**

The output is HIGH when one or more of the input waveforms are HIGH as indicated by the output waveform  $X$  in the timing diagram.

**Related Problem**

Determine the output waveform and show the timing diagram if input  $C$  is always LOW.

## Logic Expressions for an OR Gate

The logical OR function of two variables is represented mathematically by a + between the two variables, for example,  $A + B$ . The plus sign is read as “OR.”

Addition in Boolean algebra involves variables whose values are either binary 1 or binary 0. The basic rules for **Boolean addition** are as follows:

$$\begin{aligned} 0 + 0 &= 0 \\ 0 + 1 &= 1 \\ 1 + 0 &= 1 \\ 1 + 1 &= 1 \end{aligned}$$

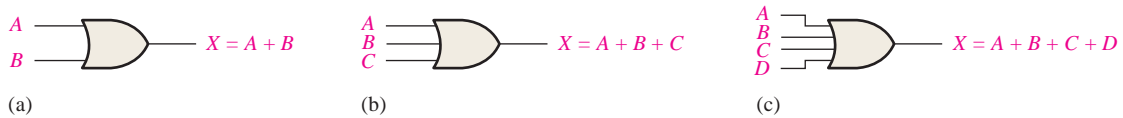
**Boolean addition is the same as the OR function.**

Notice that Boolean addition differs from binary addition in the case where two 1s are added. There is no carry in Boolean addition.

The operation of a 2-input OR gate can be expressed as follows: If one input variable is  $A$ , if the other input variable is  $B$ , and if the output variable is  $X$ , then the Boolean expression is

$$X = A + B$$

Figure 3–24(a) shows the OR gate logic symbol with two input variables and the output variable labeled.



**FIGURE 3-24** Boolean expressions for OR gates with two, three, and four inputs.

To extend the OR expression to more than two input variables, a new letter is used for each additional variable. For instance, the function of a 3-input OR gate can be expressed as  $X = A + B + C$ . The expression for a 4-input OR gate can be written as  $X = A + B + C + D$ , and so on. Parts (b) and (c) of Figure 3–24 show OR gates with three and four input variables, respectively.

OR gate operation can be evaluated by using the Boolean expressions for the output  $X$  by substituting all possible combinations of 1 and 0 values for the input variables, as shown in Table 3–6 for a 2-input OR gate. This evaluation shows that the output  $X$  of an OR gate is a 1 (HIGH) when any one or more of the inputs are 1 (HIGH). A similar analysis can be extended to OR gates with any number of input variables.

## An Application

A simplified portion of an intrusion detection and alarm system is shown in Figure 3–25. This system could be used for one room in a home—a room with two windows and a door. The sensors are magnetic switches that produce a HIGH output when open and a LOW output when closed. As long as the windows and the door are secured, the switches are closed and all three of the OR gate inputs are LOW. When one of the windows or the door is opened, a HIGH is produced on that input to the OR gate and the gate output goes HIGH. It then activates and latches an alarm circuit to warn of the intrusion.

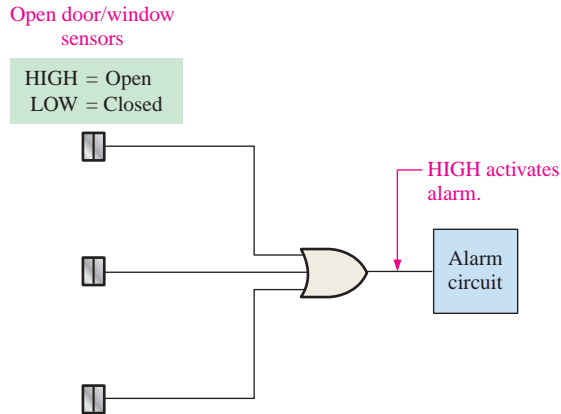
When variables are separated by +, they are ORed.

### InfoNote

A mask operation that is used in computer programming to selectively make certain bits in a data byte equal to 1 (called setting) while not affecting any other bit is done with the OR operation. A mask is used that contains a 1 in any position where a data bit is to be set. For example, if you want to force the sign bit in an 8-bit signed number to equal 1, but leave all other bits unchanged, you can OR the data byte with the mask 10000000.

**TABLE 3-6**

| A | B | $A + B = X$ |
|---|---|-------------|
| 0 | 0 | $0 + 0 = 0$ |
| 0 | 1 | $0 + 1 = 1$ |
| 1 | 0 | $1 + 0 = 1$ |
| 1 | 1 | $1 + 1 = 1$ |



**FIGURE 3-25** A simplified intrusion detection system using an OR gate.

**SECTION 3-3 CHECKUP**

1. When is the output of an OR gate HIGH?
2. When is the output of an OR gate LOW?
3. Describe the truth table for a 3-input OR gate.

**3-4 The NAND Gate**

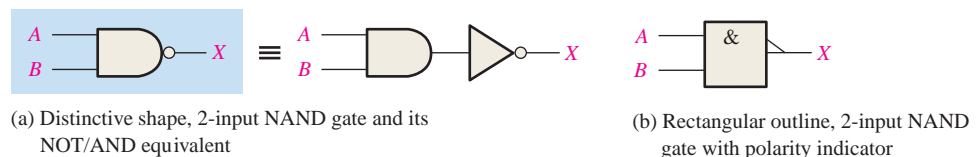
The NAND gate is a popular logic element because it can be used as a universal gate; that is, NAND gates can be used in combination to perform the AND, OR, and inverter operations. The universal property of the NAND gate will be examined thoroughly in Chapter 5.

After completing this section, you should be able to

- ♦ Identify a NAND gate by its distinctive shape symbol or by its rectangular outline symbol
- ♦ Describe the operation of a NAND gate
- ♦ Develop the truth table for a NAND gate with any number of inputs
- ♦ Produce a timing diagram for a NAND gate with any specified input waveforms
- ♦ Write the logic expression for a NAND gate with any number of inputs
- ♦ Describe NAND gate operation in terms of its negative-OR equivalent
- ♦ Discuss examples of NAND gate applications

The NAND gate is the same as the AND gate except the output is inverted.

The term *NAND* is a contraction of NOT-AND and implies an AND function with a complemented (inverted) output. The standard logic symbol for a 2-input NAND gate and its equivalency to an AND gate followed by an inverter are shown in Figure 3-26(a), where the symbol  $\equiv$  means equivalent to. A rectangular outline symbol is shown in part (b).



**FIGURE 3-26** Standard NAND gate logic symbols (ANSI/IEEE Std. 91-1984/Std. 91a-1991).

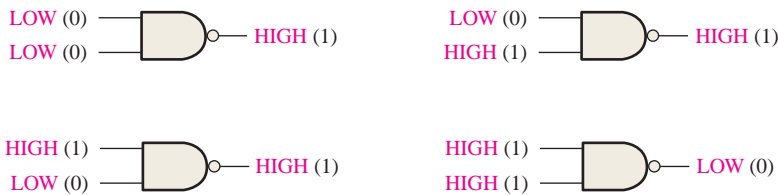


## Operation of a NAND Gate

A **NAND gate** produces a LOW output only when all the inputs are HIGH. When any of the inputs is LOW, the output will be HIGH. For the specific case of a 2-input NAND gate, as shown in Figure 3–26 with the inputs labeled *A* and *B* and the output labeled *X*, the operation can be stated as follows:

**For a 2-input NAND gate, output *X* is LOW only when inputs *A* and *B* are HIGH; *X* is HIGH when either *A* or *B* is LOW, or when both *A* and *B* are LOW.**

This operation is opposite that of the AND in terms of the output level. In a NAND gate, the LOW level (0) is the active or asserted output level, as indicated by the bubble on the output. Figure 3–27 illustrates the operation of a 2-input NAND gate for all four input combinations, and Table 3–7 is the truth table summarizing the logical operation of the 2-input NAND gate.



**FIGURE 3–27** Operation of a 2-input NAND gate. Open file F03-27 to verify NAND gate operation.

**TABLE 3–7**

Truth table for a 2-input NAND gate.

| Inputs   |          | Output   |
|----------|----------|----------|
| <i>A</i> | <i>B</i> | <i>X</i> |
| 0        | 0        | 1        |
| 0        | 1        | 1        |
| 1        | 0        | 1        |
| 1        | 1        | 0        |

1 = HIGH, 0 = LOW.

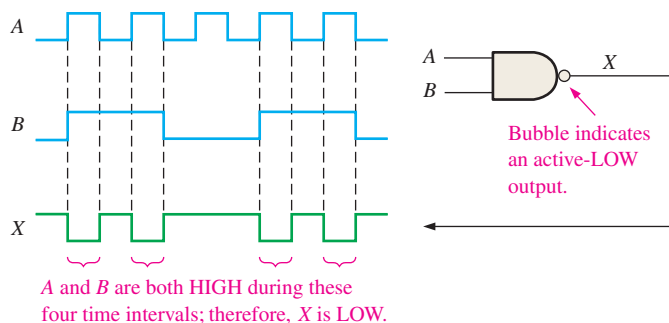


## NAND Gate Operation with Waveform Inputs

Now let's look at the pulse waveform operation of a NAND gate. Remember from the truth table that the only time a LOW output occurs is when all of the inputs are HIGH.

### EXAMPLE 3–10

If the two waveforms *A* and *B* shown in Figure 3–28 are applied to the NAND gate inputs, determine the resulting output waveform.



**FIGURE 3–28**

### Solution

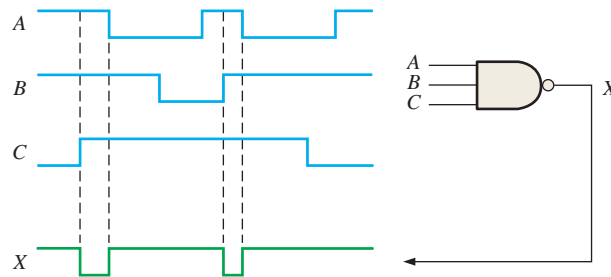
Output waveform *X* is LOW only during the four time intervals when both input waveforms *A* and *B* are HIGH as shown in the timing diagram.

### Related Problem

Determine the output waveform and show the timing diagram if input waveform *B* is inverted.

**EXAMPLE 3-11**

Show the output waveform for the 3-input NAND gate in Figure 3-29 with its proper time relationship to the inputs.



**FIGURE 3-29**

**Solution**

The output waveform *X* is LOW only when all three input waveforms are HIGH as shown in the timing diagram.

**Related Problem**

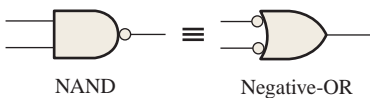
Determine the output waveform and show the timing diagram if input waveform *A* is inverted.

**Negative-OR Equivalent Operation of a NAND Gate**

Inherent in a NAND gate’s operation is the fact that one or more LOW inputs produce a HIGH output. Table 3-7 shows that output *X* is HIGH (1) when any of the inputs, *A* and *B*, is LOW (0). From this viewpoint, a NAND gate can be used for an OR operation that requires one or more LOW inputs to produce a HIGH output. This aspect of NAND operation is referred to as **negative-OR**. The term *negative* in this context means that the inputs are defined to be in the active or asserted state when LOW.

**For a 2-input NAND gate performing a negative-OR operation, output *X* is HIGH when either input *A* or input *B* is LOW, or when both *A* and *B* are LOW.**

When a NAND gate is used to detect one or more LOWs on its inputs rather than all HIGHs, it is performing the negative-OR operation and is represented by the standard logic symbol shown in Figure 3-30. Although the two symbols in Figure 3-30 represent the same physical gate, they serve to define its role or mode of operation in a particular application, as illustrated by Examples 3-12 and 3-13.



**FIGURE 3-30** ANSI/IEEE standard symbols representing the two equivalent operations of a NAND gate.

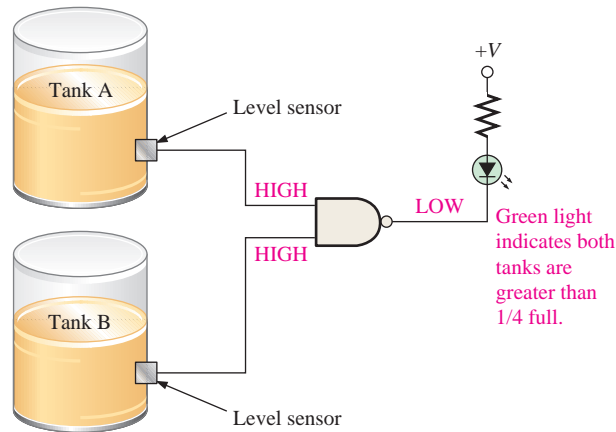
**EXAMPLE 3-12**

Two tanks store certain liquid chemicals that are required in a manufacturing process. Each tank has a sensor that detects when the chemical level drops to 25% of full. The sensors produce a HIGH level of 5 V when the tanks are more than one-quarter full. When the volume of chemical in a tank drops to one-quarter full, the sensor puts out a LOW level of 0 V.

It is required that a single green light-emitting diode (LED) on an indicator panel show when both tanks are more than one-quarter full. Show how a NAND gate can be used to implement this function.

**Solution**

Figure 3-31 shows a NAND gate with its two inputs connected to the tank level sensors and its output connected to the indicator panel. The operation can be stated as follows: If tank *A* and tank *B* are above one-quarter full, the LED is on.



**FIGURE 3-31**

As long as both sensor outputs are HIGH (5 V), indicating that both tanks are more than one-quarter full, the NAND gate output is LOW (0 V). The green LED circuit is connected so that a LOW voltage turns it on. The resistor limits the LED current.

#### Related Problem

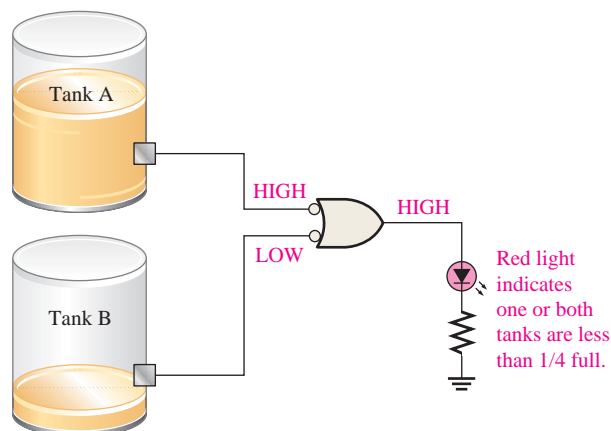
How can the circuit of Figure 3-31 be modified to monitor the levels in three tanks rather than two?

#### EXAMPLE 3-13

For the process described in Example 3-12 it has been decided to have a red LED display come on when at least one of the tanks falls to the quarter-full level rather than have the green LED display indicate when both are above one quarter. Show how this requirement can be implemented.

#### Solution

Figure 3-32 shows a NAND gate operating as a negative-OR gate to detect the occurrence of at least one LOW on its inputs. A sensor puts out a LOW voltage if the volume in its tank goes to one-quarter full or less. When this happens, the gate output goes HIGH. The red LED circuit in the panel is connected so that a HIGH voltage turns it on. The operation can be stated as follows: If tank *A* or tank *B* or both are below one-quarter full, the LED is on.



**FIGURE 3-32**

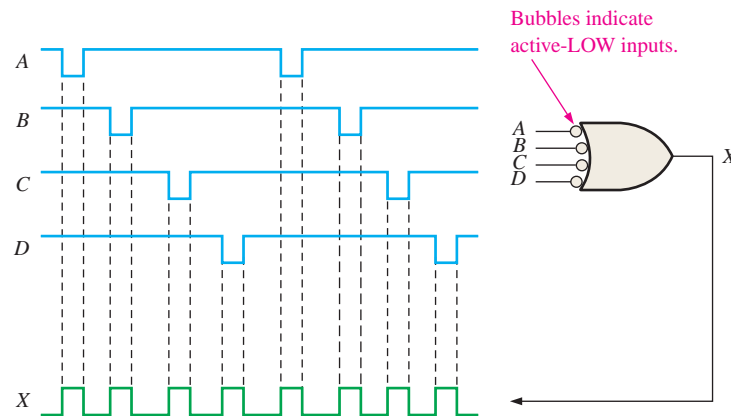
Notice that, in this example and in Example 3–12, the same 2-input NAND gate is used, but in this example it is operating as a negative-OR gate and a different gate symbol is used in the schematic. This illustrates the different way in which the NAND and equivalent negative-OR operations are used.

**Related Problem**

How can the circuit in Figure 3–32 be modified to monitor four tanks rather than two?

**EXAMPLE 3-14**

For the 4-input NAND gate in Figure 3–33, operating as a negative-OR gate, determine the output with respect to the inputs.



**FIGURE 3-33**

**Solution**

The output waveform *X* is HIGH any time an input waveform is LOW as shown in the timing diagram.

**Related Problem**

Determine the output waveform if input waveform *A* is inverted before it is applied to the gate.

**Logic Expressions for a NAND Gate**

The Boolean expression for the output of a 2-input NAND gate is

$$X = \overline{AB}$$

This expression says that the two input variables, *A* and *B*, are first ANDed and then complemented, as indicated by the bar over the AND expression. This is a description in equation form of the operation of a NAND gate with two inputs. Evaluating this expression for all possible values of the two input variables, you get the results shown in Table 3–8.

Once an expression is determined for a given logic function, that function can be evaluated for all possible values of the variables. The evaluation tells you exactly what the output of the logic circuit is for each of the input conditions, and it therefore gives you a complete description of the circuit’s logic operation. The NAND expression can be extended to more than two input variables by including additional letters to represent the other variables.

A bar over a variable or variables indicates an inversion.

| TABLE 3-8 |          |   |
|-----------|----------|---|
| <i>A</i>  | <i>B</i> | $\overline{AB} = X$                       |
| 0         | 0        | $\overline{0 \cdot 0} = \overline{0} = 1$ |
| 0         | 1        | $\overline{0 \cdot 1} = \overline{0} = 1$ |
| 1         | 0        | $\overline{1 \cdot 0} = \overline{0} = 1$ |
| 1         | 1        | $\overline{1 \cdot 1} = \overline{1} = 0$ |

## SECTION 3-4 CHECKUP

1. When is the output of a NAND gate LOW?
2. When is the output of a NAND gate HIGH?
3. Describe the functional differences between a NAND gate and a negative-OR gate. Do they both have the same truth table?
4. Write the output expression for a NAND gate with inputs  $A$ ,  $B$ , and  $C$ .

## 3-5 The NOR Gate

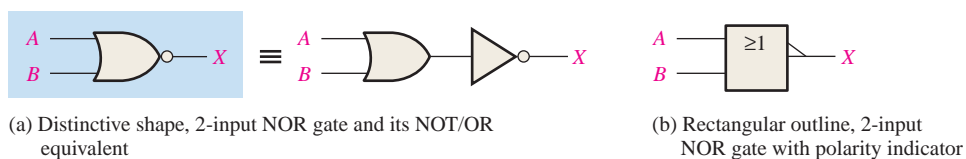
The NOR gate, like the NAND gate, is a useful logic element because it can also be used as a universal gate; that is, NOR gates can be used in combination to perform the AND, OR, and inverter operations. The universal property of the NOR gate will be examined thoroughly in Chapter 5.

After completing this section, you should be able to

- ◆ Identify a NOR gate by its distinctive shape symbol or by its rectangular outline symbol
- ◆ Describe the operation of a NOR gate
- ◆ Develop the truth table for a NOR gate with any number of inputs
- ◆ Produce a timing diagram for a NOR gate with any specified input waveforms
- ◆ Write the logic expression for a NOR gate with any number of inputs
- ◆ Describe NOR gate operation in terms of its negative-AND equivalent
- ◆ Discuss examples of NOR gate applications

The term *NOR* is a contraction of NOT-OR and implies an OR function with an inverted (complemented) output. The standard logic symbol for a 2-input NOR gate and its equivalent OR gate followed by an inverter are shown in Figure 3-34(a). A rectangular outline symbol is shown in part (b).

The NOR is the same as the OR except the output is inverted.

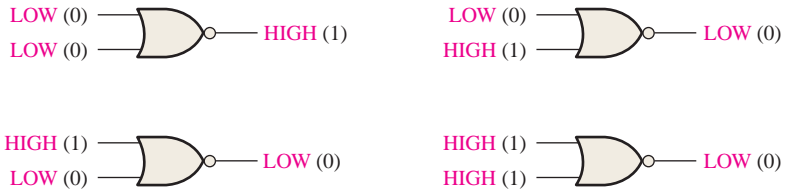


**FIGURE 3-34** Standard NOR gate logic symbols (ANSI/IEEE Std. 91-1984/Std. 91a-1991).

## Operation of a NOR Gate

A **NOR gate** produces a LOW output when *any* of its inputs is HIGH. Only when all of its inputs are LOW is the output HIGH. For the specific case of a 2-input NOR gate, as shown in Figure 3-34 with the inputs labeled  $A$  and  $B$  and the output labeled  $X$ , the operation can be stated as follows:

**For a 2-input NOR gate, output  $X$  is LOW when either input  $A$  or input  $B$  is HIGH, or when both  $A$  and  $B$  are HIGH;  $X$  is HIGH only when both  $A$  and  $B$  are LOW.**



**FIGURE 3-35** Operation of a 2-input NOR gate. Open file F03-35 to verify NOR gate operation.

**TABLE 3-9**

Truth table for a 2-input NOR gate.

| Inputs |   | Output |
|--------|---|--------|
| A      | B | X      |
| 0      | 0 | 1      |
| 0      | 1 | 0      |
| 1      | 0 | 0      |
| 1      | 1 | 0      |

1 = HIGH, 0 = LOW.

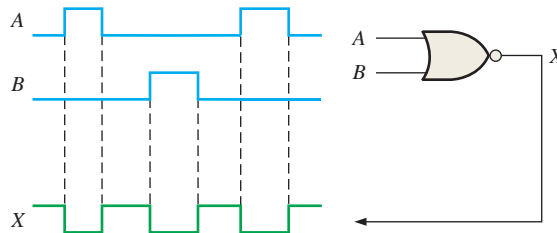
This operation results in an output level opposite that of the OR gate. In a NOR gate, the LOW output is the active or asserted output level as indicated by the bubble on the output. Figure 3-35 illustrates the operation of a 2-input NOR gate for all four possible input combinations, and Table 3-9 is the truth table for a 2-input NOR gate.

### NOR Gate Operation with Waveform Inputs

The next two examples illustrate the operation of a NOR gate with pulse waveform inputs. Again, as with the other types of gates, we will simply follow the truth table operation to determine the output waveforms in the proper time relationship to the inputs.

#### EXAMPLE 3-15

If the two waveforms shown in Figure 3-36 are applied to a NOR gate, what is the resulting output waveform?



**FIGURE 3-36**

#### Solution

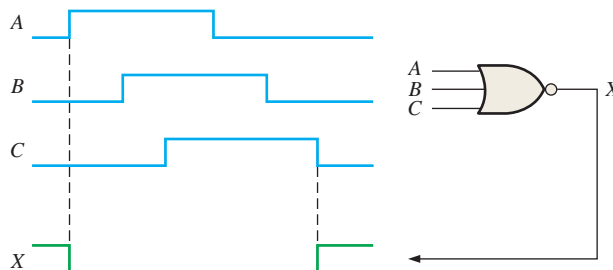
Whenever any input of the NOR gate is HIGH, the output is LOW as shown by the output waveform X in the timing diagram.

#### Related Problem

Invert input B and determine the output waveform in relation to the inputs.

#### EXAMPLE 3-16

Show the output waveform for the 3-input NOR gate in Figure 3-37 with the proper time relation to the inputs.



**FIGURE 3-37**

**Solution**

The output  $X$  is LOW when any input is HIGH as shown by the output waveform  $X$  in the timing diagram.

**Related Problem**

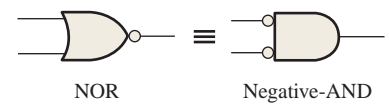
With the  $B$  and  $C$  inputs inverted, determine the output and show the timing diagram.

**Negative-AND Equivalent Operation of the NOR Gate**

A NOR gate, like the NAND, has another aspect of its operation that is inherent in the way it logically functions. Table 3–9 shows that a HIGH is produced on the gate output only when all of the inputs are LOW. From this viewpoint, a NOR gate can be used for an AND operation that requires all LOW inputs to produce a HIGH output. This aspect of NOR operation is called **negative-AND**. The term *negative* in this context means that the inputs are defined to be in the active or asserted state when LOW.

**For a 2-input NOR gate performing a negative-AND operation, output  $X$  is HIGH only when both inputs  $A$  and  $B$  are LOW.**

When a NOR gate is used to detect all LOWs on its inputs rather than one or more HIGHs, it is performing the negative-AND operation and is represented by the standard symbol in Figure 3–38. Remember that the two symbols in Figure 3–38 represent the same physical gate and serve only to distinguish between the two modes of its operation. The following three examples illustrate this.



**FIGURE 3–38** Standard symbols representing the two equivalent operations of a NOR gate.

**EXAMPLE 3–17**

A device is needed to indicate when two LOW levels occur simultaneously on its inputs and to produce a HIGH output as an indication. Specify the device.

**Solution**

A 2-input NOR gate operating as a negative-AND gate is required to produce a HIGH output when both inputs are LOW, as shown in Figure 3–39.



**FIGURE 3–39**

**Related Problem**

A device is needed to indicate when one or two HIGH levels occur on its inputs and to produce a LOW output as an indication. Specify the device.

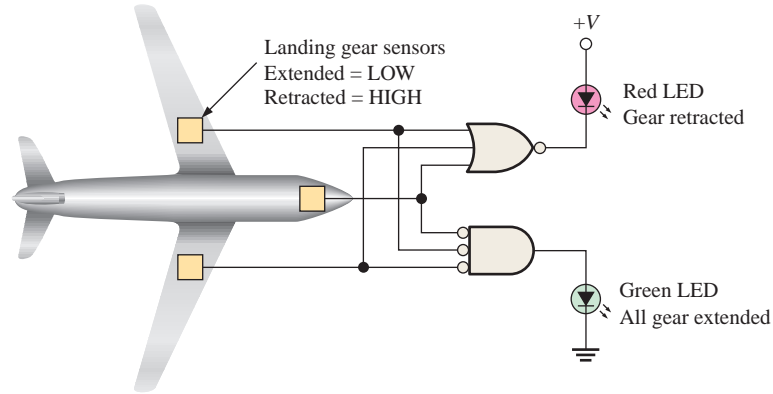
**EXAMPLE 3–18**

As part of an aircraft’s functional monitoring system, a circuit is required to indicate the status of the landing gears prior to landing. A green LED display turns on if all three gears are properly extended when the “gear down” switch has been activated in preparation for landing. A red LED display turns on if any of the gears fail to extend properly prior to landing. When a landing gear is extended, its sensor produces a LOW voltage. When a landing gear is retracted, its sensor produces a HIGH voltage. Implement a circuit to meet this requirement.

**Solution**

Power is applied to the circuit only when the “gear down” switch is activated. Use a NOR gate for each of the two requirements as shown in Figure 3–40. One NOR gate operates as a negative-AND to detect a LOW from each of the three landing gear sensors. When all three of the gate inputs are LOW, the three landing gears are properly extended and the

resulting HIGH output from the negative-AND gate turns on the green LED display. The other NOR gate operates as a NOR to detect if one or more of the landing gears remain retracted when the “gear down” switch is activated. When one or more of the landing gears remain retracted, the resulting HIGH from the sensor is detected by the NOR gate, which produces a LOW output to turn on the red LED warning display.



**FIGURE 3-40**

**Related Problem**

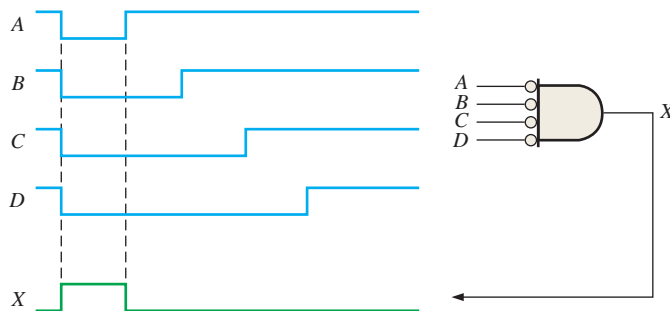
What type of gate should be used to detect if all three landing gears are retracted after takeoff, assuming a LOW output is required to activate an LED display?



When driving a load such as an LED with a logic gate, consult the manufacturer's data sheet for maximum drive capabilities (output current). A regular IC logic gate may not be capable of handling the current required by certain loads such as some LEDs. Logic gates with a buffered output, such as an open-collector (OC) or open-drain (OD) output, are available in many types of IC logic gate configurations. The output current capability of typical IC logic gates is limited to the  $\mu\text{A}$  or relatively low mA range. For example, standard TTL can handle output currents up to 16 mA but only when the output is LOW. Most LEDs require currents in the range of about 10 mA to 50 mA.

**EXAMPLE 3-19**

For the 4-input NOR gate operating as a negative-AND in Figure 3-41, determine the output relative to the inputs.



**FIGURE 3-41**



**Solution**

Any time all of the input waveforms are LOW, the output is HIGH as shown by output waveform *X* in the timing diagram.

**Related Problem**

Determine the output with input *D* inverted and show the timing diagram.

**Logic Expressions for a NOR Gate**

The Boolean expression for the output of a 2-input NOR gate can be written as

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

This equation says that the two input variables are first ORed and then complemented, as indicated by the bar over the OR expression. Evaluating this expression, you get the results shown in Table 3–10. The NOR expression can be extended to more than two input variables by including additional letters to represent the other variables.

**TABLE 3–10**

| <i>A</i> | <i>B</i> | $\overline{A + B} = X$                |
|----------|----------|---------------------------------------|
| 0        | 0        | $\overline{0 + 0} = \overline{0} = 1$ |
| 0        | 1        | $\overline{0 + 1} = \overline{1} = 0$ |
| 1        | 0        | $\overline{1 + 0} = \overline{1} = 0$ |
| 1        | 1        | $\overline{1 + 1} = \overline{1} = 0$ |

**SECTION 3–5 CHECKUP**

1. When is the output of a NOR gate HIGH?
2. When is the output of a NOR gate LOW?
3. Describe the functional difference between a NOR gate and a negative-AND gate. Do they both have the same truth table?
4. Write the output expression for a 3-input NOR with input variables *A*, *B*, and *C*.

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

Exclusive-OR and exclusive-NOR gates are formed by a combination of other gates already discussed, as you will see in Chapter 5. However, because of their fundamental importance in many applications, these gates are often treated as basic logic elements with their own unique symbols.

After completing this section, you should be able to

- ◆ Identify the exclusive-OR and exclusive-NOR gates by their distinctive shape symbols or by their rectangular outline symbols
- ◆ Describe the operations of exclusive-OR and exclusive-NOR gates
- ◆ Show the truth tables for exclusive-OR and exclusive-NOR gates
- ◆ Produce a timing diagram for an exclusive-OR or exclusive-NOR gate with any specified input waveforms
- ◆ Discuss examples of exclusive-OR and exclusive-NOR gate applications

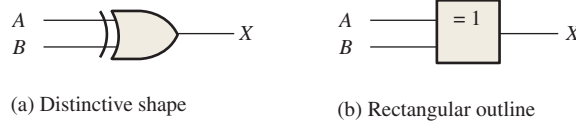
**The Exclusive-OR Gate**

Standard symbols for an exclusive-OR (XOR for short) gate are shown in Figure 3–42. The XOR gate has only two inputs. The **exclusive-OR gate** performs modulo-2 addition (introduced in Chapter 2). The output of an exclusive-OR gate is HIGH *only* when the two

**InfoNote**

Exclusive-OR gates connected to form an adder circuit allow a processor to perform addition, subtraction, multiplication, and division in its Arithmetic Logic Unit (ALU). An exclusive-OR gate combines basic AND, OR, and NOT logic.

For an exclusive-OR gate, opposite inputs make the output HIGH.



**FIGURE 3-42** Standard logic symbols for the exclusive-OR gate.

**TABLE 3-11**

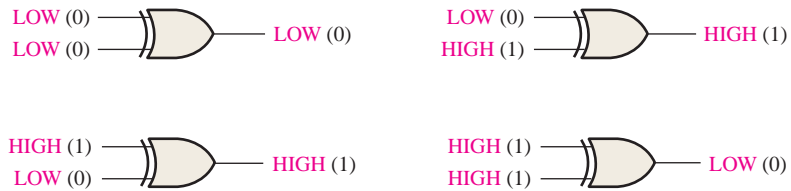
Truth table for an exclusive-OR gate.

| Inputs |   | Output |
|--------|---|--------|
| A      | B | X      |
| 0      | 0 | 0      |
| 0      | 1 | 1      |
| 1      | 0 | 1      |
| 1      | 1 | 0      |

inputs are at opposite logic levels. This operation can be stated as follows with reference to inputs *A* and *B* and output *X*:

**For an exclusive-OR gate, output *X* is HIGH when input *A* is LOW and input *B* is HIGH, or when input *A* is HIGH and input *B* is LOW; *X* is LOW when *A* and *B* are both HIGH or both LOW.**

The four possible input combinations and the resulting outputs for an XOR gate are illustrated in Figure 3-43. The HIGH level is the active or asserted output level and occurs only when the inputs are at opposite levels. The operation of an XOR gate is summarized in the truth table shown in Table 3-11.



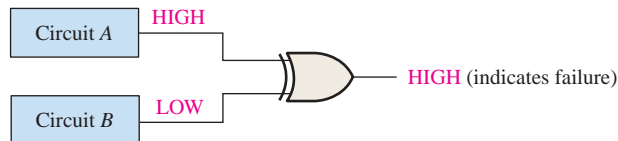
**FIGURE 3-43** All possible logic levels for an exclusive-OR gate. Open file F03-43 to verify XOR gate operation.

**EXAMPLE 3-20**

A certain system contains two identical circuits operating in parallel. As long as both are operating properly, the outputs of both circuits are always the same. If one of the circuits fails, the outputs will be at opposite levels at some time. Devise a way to monitor and detect that a failure has occurred in one of the circuits.

**Solution**

The outputs of the circuits are connected to the inputs of an XOR gate as shown in Figure 3-44. A failure in either one of the circuits produces differing outputs, which cause the XOR inputs to be at opposite levels. This condition produces a HIGH on the output of the XOR gate, indicating a failure in one of the circuits.



**FIGURE 3-44**

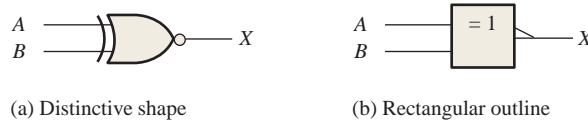
**Related Problem**

Will the exclusive-OR gate always detect simultaneous failures in both circuits of Figure 3-44? If not, under what condition?

### The Exclusive-NOR Gate

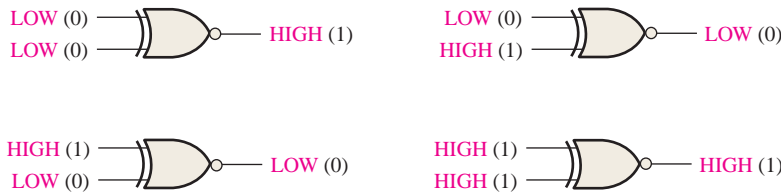
Standard symbols for an **exclusive-NOR (XNOR) gate** are shown in Figure 3–45. Like the XOR gate, an XNOR has only two inputs. The bubble on the output of the XNOR symbol indicates that its output is opposite that of the XOR gate. When the two input logic levels are opposite, the output of the exclusive-NOR gate is LOW. The operation can be stated as follows (*A* and *B* are inputs, *X* is the output):

**For an exclusive-NOR gate, output *X* is LOW when input *A* is LOW and input *B* is HIGH, or when *A* is HIGH and *B* is LOW; *X* is HIGH when *A* and *B* are both HIGH or both LOW.**



**FIGURE 3-45** Standard logic symbols for the exclusive-NOR gate.

The four possible input combinations and the resulting outputs for an XNOR gate are shown in Figure 3–46. The operation of an XNOR gate is summarized in Table 3–12. Notice that the output is HIGH when the same level is on both inputs.



**FIGURE 3-46** All possible logic levels for an exclusive-NOR gate. Open file F03-46 to verify XNOR gate operation.

**TABLE 3-12**

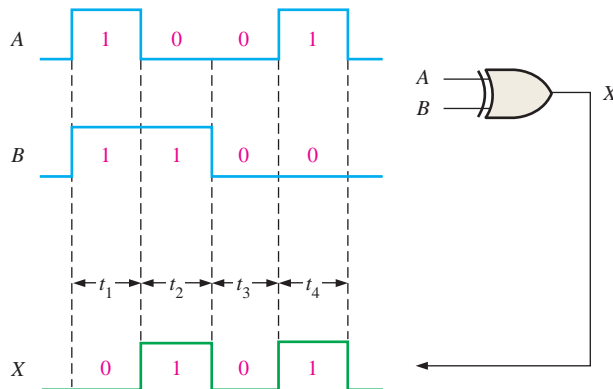
Truth table for an exclusive-NOR gate.

| Inputs   |          | Output   |
|----------|----------|----------|
| <i>A</i> | <i>B</i> | <i>X</i> |
| 0        | 0        | 1        |
| 0        | 1        | 0        |
| 1        | 0        | 0        |
| 1        | 1        | 1        |



### Operation with Waveform Inputs

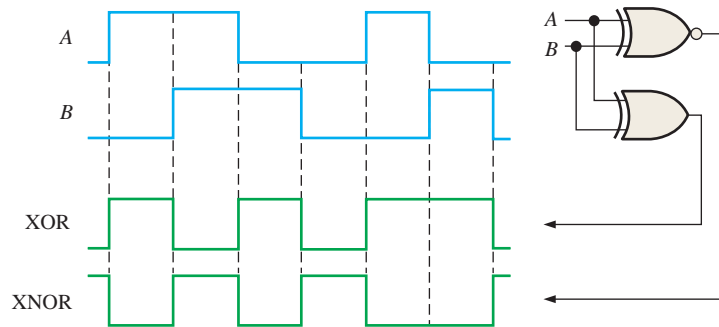
As we have done with the other gates, let’s examine the operation of XOR and XNOR gates with pulse waveform inputs. As before, we apply the truth table operation during each distinct time interval of the pulse waveform inputs, as illustrated in Figure 3–47 for an XOR gate. You can see that the input waveforms *A* and *B* are at opposite levels during time intervals  $t_2$  and  $t_4$ . Therefore, the output *X* is HIGH during these two times. Since both inputs are at the same level, either both HIGH or both LOW, during time intervals  $t_1$  and  $t_3$ , the output is LOW during those times as shown in the timing diagram.



**FIGURE 3-47** Example of exclusive-OR gate operation with pulse waveform inputs.

**EXAMPLE 3-21**

Determine the output waveforms for the XOR gate and for the XNOR gate, given the input waveforms, *A* and *B*, in Figure 3-48.



**FIGURE 3-48**

**Solution**

The output waveforms are shown in Figure 3-48. Notice that the XOR output is HIGH only when both inputs are at opposite levels. Notice that the XNOR output is HIGH only when both inputs are the same.

**Related Problem**

Determine the output waveforms if the two input waveforms, *A* and *B*, are inverted.

**An Application**

An exclusive-OR gate can be used as a two-bit modulo-2 adder. Recall from Chapter 2 that the basic rules for binary addition are as follows:  $0 + 0 = 0$ ,  $0 + 1 = 1$ ,  $1 + 0 = 1$ , and  $1 + 1 = 10$ . An examination of the truth table for an XOR gate shows that its output is the binary sum of the two input bits. In the case where the inputs are both 1s, the output is the sum 0, but you lose the carry of 1. In Chapter 6 you will see how XOR gates are combined to make complete adding circuits. Table 3-13 illustrates an XOR gate used as a modulo-2 adder. It is used in CRC systems to implement the division process that was described in Chapter 2.

**TABLE 3-13**

An XOR gate used to add two bits.

| Input Bits |          | Output (Sum)                |
|------------|----------|-----------------------------|
| <i>A</i>   | <i>B</i> | $\Sigma$                    |
| 0          | 0        | 0                           |
| 0          | 1        | 1                           |
| 1          | 0        | 1                           |
| 1          | 1        | 0 (without the 1 carry bit) |

The diagram shows a logic circuit of an XOR gate with two inputs and one output.

**SECTION 3-6 CHECKUP**

1. When is the output of an XOR gate HIGH?
2. When is the output of an XNOR gate HIGH?
3. How can you use an XOR gate to detect when two bits are different?

**3-7 Programmable Logic**

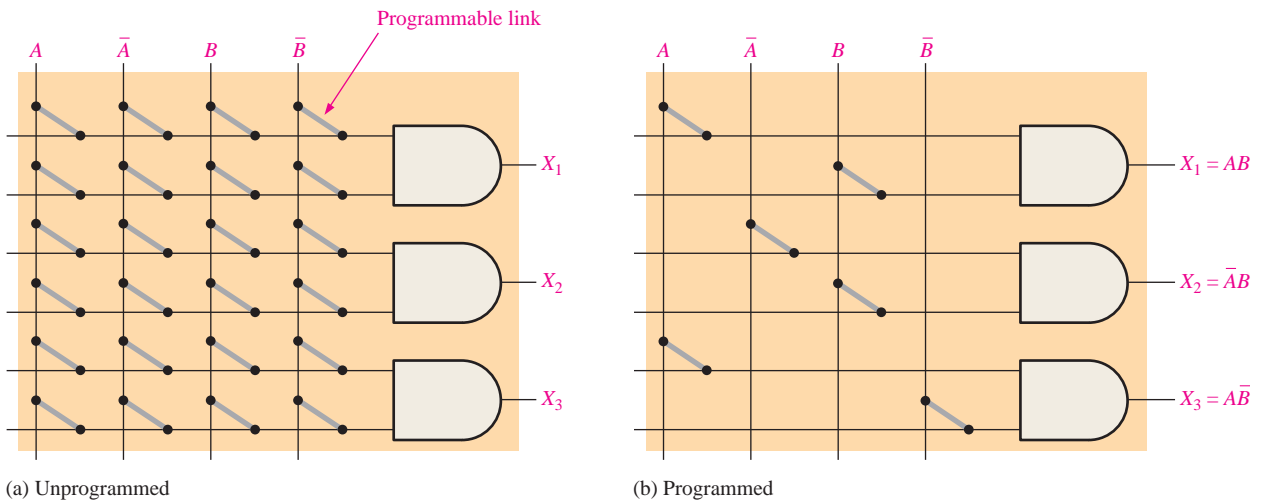
Programmable logic was introduced in Chapter 1. In this section, the basic concept of the programmable AND array, which forms the basis for most programmable logic, is discussed, and the major process technologies are covered. A programmable logic device (PLD) is one that does not initially have a fixed-logic function but that can be programmed to implement just about any logic design. As you have learned, two types of PLD are the SPLD and CPLD. In addition to the PLD, the other major category of programmable logic is the FPGA. Also, basic VHDL programming is introduced.

After completing this section, you should be able to

- ◆ Describe the concept of a programmable AND array
- ◆ Discuss various process technologies for programming a PLD
- ◆ Discuss downloading a design to a programmable logic device
- ◆ Discuss text entry and graphic entry as two methods for programmable logic design
- ◆ Explain in-system programming
- ◆ Write VHDL descriptions of logic gates

**The AND Array**

Most types of PLDs use some form of **AND array**. Basically, this array consists of AND gates and a matrix of interconnections with a programmable link at each cross point, as shown in Figure 3-49(a). Programmable links allow a connection between a row line and a column line in the interconnection matrix to be opened or left intact. For each input to an AND gate, only one programmable link is left intact in order to connect the desired variable to the gate input. Figure 3-49(b) illustrates an array after it has been programmed.



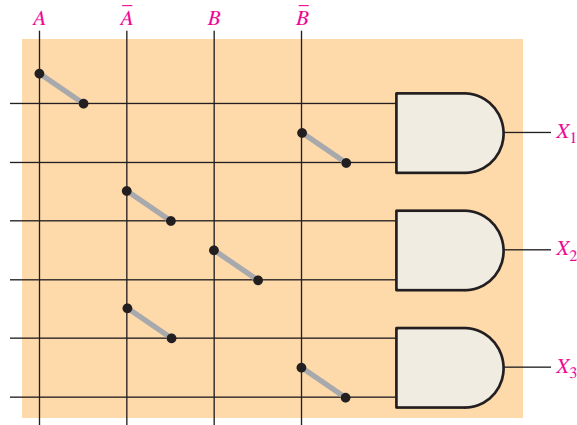
**FIGURE 3-49** Concept of a programmable AND array.

**EXAMPLE 3-22**

Show the AND array in Figure 3-49(a) programmed for the following outputs:  
 $X_1 = A\bar{B}$ ,  $X_2 = \bar{A}B$ , and  $X_3 = \bar{A}\bar{B}$

**Solution**

See Figure 3-50.



**FIGURE 3-50**

**Related Problem**

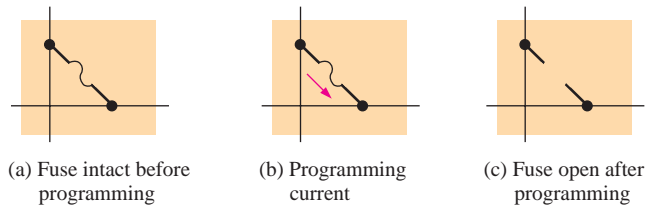
How many rows, columns, and AND gate inputs are required for three input variables in a 3-AND gate array?

**Programmable Link Process Technologies**

A process technology is the physical method by which a link is made. Several different process technologies are used for programmable links in PLDs.

**Fuse Technology**

This was the original programmable link technology. It is still used in some SPLDs. The **fuse** is a metal link that connects a row and a column in the interconnection matrix. Before programming, there is a fused connection at each intersection. To program a device, the selected fuses are opened by passing a current through them sufficient to “blow” the fuse and break the connection. The intact fuses remain and provide a connection between the rows and columns. The fuse link is illustrated in Figure 3-51. Programmable logic devices that use fuse technology are one-time programmable (**OTP**).

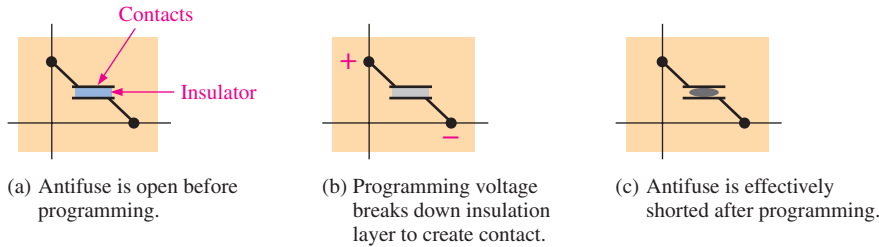


**FIGURE 3-51** The programmable fuse link.

**Antifuse Technology**

An **antifuse** programmable link is the opposite of a fuse link. Instead of breaking the connection, a connection is made during programming. An antifuse starts out as an open circuit

whereas the fuse starts out as a short circuit. Before programming, there are no connections between the rows and columns in the interconnection matrix. An antifuse is basically two conductors separated by an insulator. To program a device with antifuse technology, a programmer tool applies a sufficient voltage across selected antifuses to break down the insulation between the two conductive materials, causing the insulator to become a low-resistance link. The antifuse link is illustrated in Figure 3–52. An antifuse device is also a one-time programmable (OTP) device.

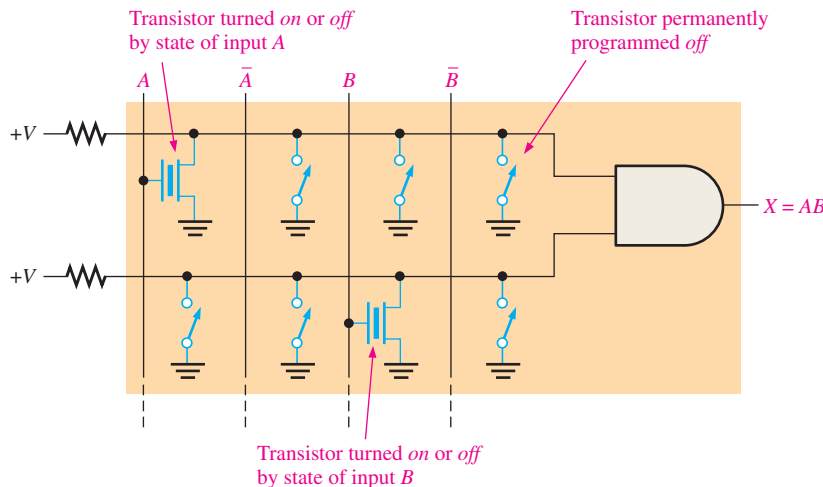


**FIGURE 3-52** The programmable antifuse link.

### EPROM Technology

In certain programmable logic devices, the programmable links are similar to the memory cells in **EPROMs** (electrically programmable read-only memories). This type of PLD is programmed using a special tool known as a device programmer. The device is inserted into the programmer, which is connected to a computer running the programming software. Most EPROM-based PLDs are one-time programmable (OTP). However, those with windowed packages can be erased with UV (ultraviolet) light and reprogrammed using a standard PLD programming fixture. EPROM process technology uses a special type of MOS transistor, known as a floating-gate transistor, as the programmable link. The floating-gate device utilizes a process called Fowler-Nordheim tunneling to place electrons in the floating-gate structure.

In a programmable AND array, the floating-gate transistor acts as a switch to connect the row line to either a HIGH or a LOW, depending on the input variable. For input variables that are not used, the transistor is programmed to be permanently *off* (open). Figure 3–53 shows one AND gate in a simple array. Variable *A* controls the state of the transistor in the first column, and variable *B* controls the transistor in the third column. When a transistor is *off*, like an open switch, the input line to the AND gate is at +*V* (HIGH). When a transistor is *on*, like a closed switch, the input line is connected to ground (LOW). When variable *A*



**FIGURE 3-53** A simple AND array with EPROM technology. Only one gate in the array is shown for simplicity.

or  $B$  is 0 (LOW), the transistor is *on*, keeping the input line to the AND gate LOW. When  $A$  or  $B$  is 1 (HIGH), the transistor is *off*, keeping the input line to the AND gate HIGH.

### EEPROM Technology

Electrically erasable programmable read-only memory technology is similar to EPROM because it also uses a type of floating-gate transistor in E<sup>2</sup>CMOS cells. The difference is that **EEPROM** can be erased and reprogrammed electrically without the need for UV light or special fixtures. An E<sup>2</sup>CMOS device can be programmed after being installed on a printed circuit board (PCB), and many can be reprogrammed while operating in a system. This is called **in-system programming (ISP)**. Figure 3–53 can also be used as an example to represent an AND array with EEPROM technology.

### Flash Technology

**Flash** technology is based on a single transistor link and is both nonvolatile and reprogrammable. Flash elements are a type of EEPROM but are faster and result in higher density devices than the standard EEPROM link. A detailed discussion of the flash memory element can be found in Chapter 11.

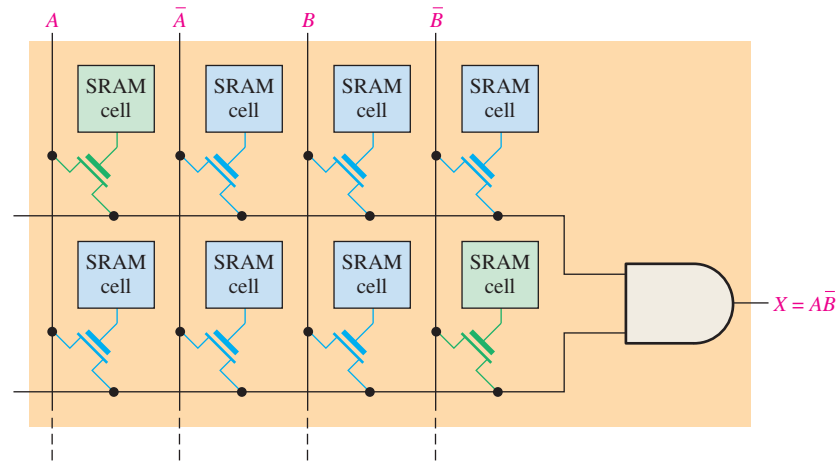
### SRAM Technology

Many FPGAs and some CPLDs use a process technology similar to that used in **SRAMs** (static random-access memories). The basic concept of SRAM-based programmable logic arrays is illustrated in Figure 3–54(a). A SRAM-type memory cell is used to turn a transistor *on* or *off* to connect or disconnect rows and columns. For example, when the memory cell contains a 1 (green), the transistor is *on* and connects the associated row and column lines, as shown in part (b). When the memory cell contains a 0 (blue), the transistor is *off* so there is no connection between the lines, as shown in part (c).

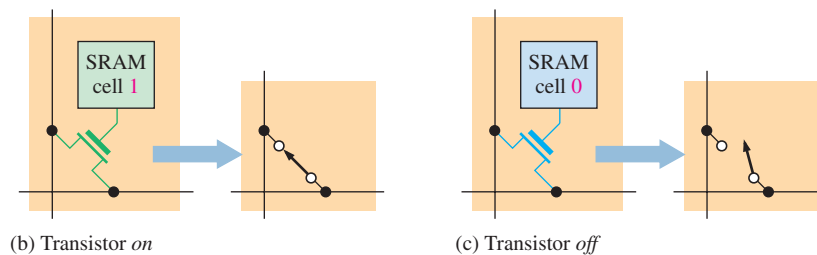
**InfoNote**

Most system-level designs incorporate a variety of devices such as RAMs, ROMs, controllers, and processors that are interconnected by a large quantity of general-purpose logic devices often referred to as “glue” logic. PLDs have come to replace many of the SSI and MSI “glue” devices. The use of PLDs provides a reduction in package count.

For example, in memory systems, PLDs can be used for memory address decoding and to generate memory write signals as well as other functions.



(a) SRAM-based programmable array



(b) Transistor *on*

(c) Transistor *off*

**FIGURE 3–54** Concept of an AND array with SRAM technology.



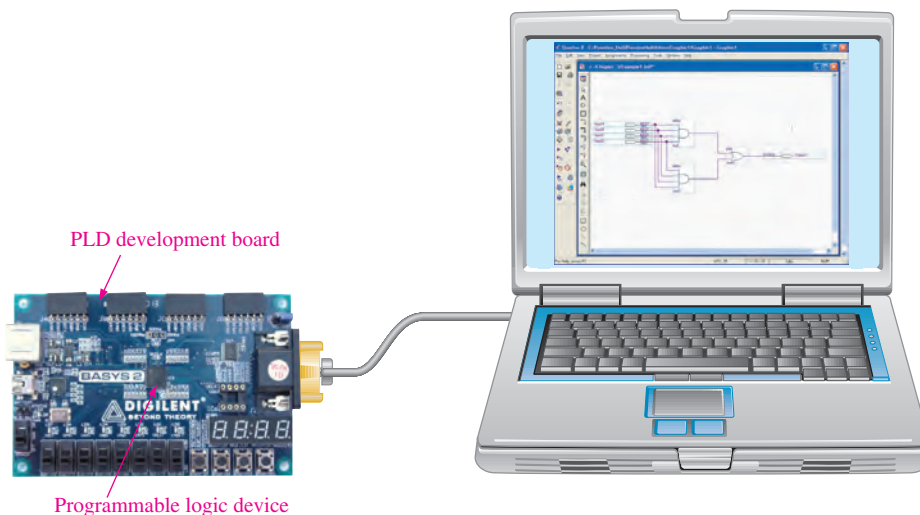
SRAM technology is different from the other process technologies discussed because it is a volatile technology. This means that a SRAM cell does not retain data when power is turned *off*. The programming data must be loaded into a memory; and when power is turned *on*, the data from the memory reprograms the SRAM-based PLD.

The fuse, antifuse, EPROM, EEPROM, and flash process technologies are nonvolatile, so they retain their programming when the power is *off*. A fuse is permanently open, an antifuse is permanently closed, and floating-gate transistors used in EPROM and EEPROM-based arrays can retain their *on* or *off* state indefinitely.

## Device Programming

The general concept of programming was introduced in Chapter 1, and you have seen how interconnections can be made in a simple array by opening or closing the programmable links. SPLDs, CPLDs, and FPGAs are programmed in essentially the same way. The devices with OTP (one-time programmable) process technologies (fuse, antifuse, or EPROM) must be programmed with a special hardware fixture called a *programmer*. The programmer is connected to a computer by a standard interface cable. Development software is installed on the computer, and the device is inserted into the programmer socket. Most programmers have adapters that allow different types of packages to be plugged in.

EEPROM, flash, and SRAM-based programmable logic devices are reprogrammable and can be reconfigured multiple times. Although a device programmer can be used for this type of device, it is generally programmed initially on a PLD development board, as shown in Figure 3–55. A logic design can be developed using this approach because any necessary changes during the design process can be readily accomplished by simply reprogramming the PLD. A PLD to which a software logic design can be downloaded is called a **target device**. In addition to the target device, development boards typically provide other circuitry and connectors for interfacing to the computer and other peripheral circuits. Also, test points and display devices for observing the operation of the programmed device are included on the development board.



**FIGURE 3–55** Programming setup for reprogrammable logic devices. (Photo courtesy of Digilent, Inc.)

## Design Entry

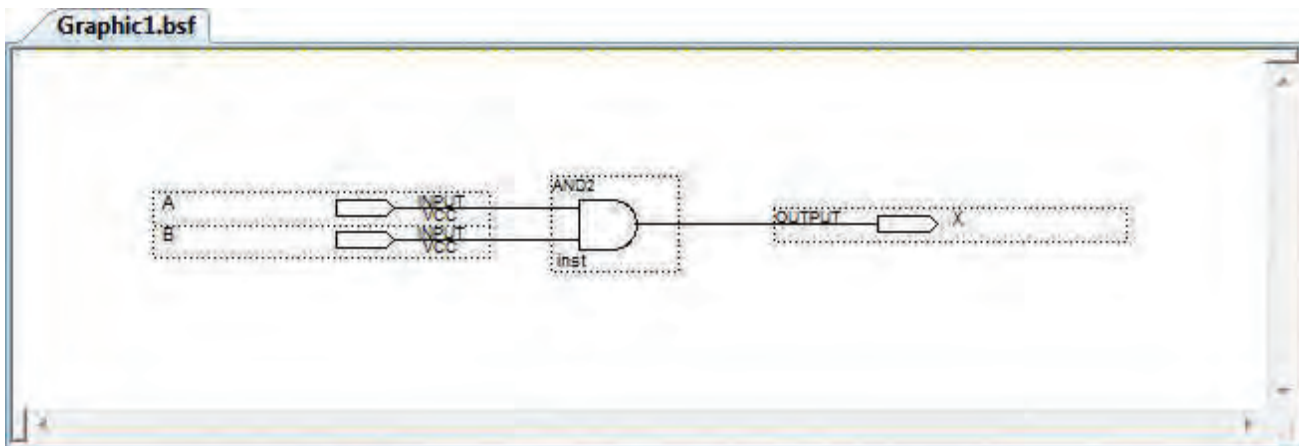
As you learned in Chapter 1, design entry is where the logic design is programmed into the development software. The two main ways to enter a design are by text entry or graphic (schematic) entry, and manufacturers of programmable logic provide software packages to support their devices that allow for both methods.

**Text entry** in most development software, regardless of the manufacturer, supports two or more hardware development languages (HDLs). For example, all software packages support both IEEE standard HDLs, VHDL, and Verilog. Some software packages also support certain proprietary languages such as AHDL.

In **graphic (schematic) entry**, logic symbols such as AND gates and OR gates are placed on the screen and interconnected to form the desired circuit. In this method you use the familiar logic symbols, but the software actually converts each symbol and interconnections to a text file for the computer to use; you do not see this process. A simple example of both a text entry screen and a graphic entry screen for an AND gate is shown in Figure 3–56. As a general rule, graphic entry is used for less-complex logic circuits and text entry, although it can also be used for very simple logic, is used for larger, more complex implementation.

```
Vhdl1.vhd
entity VHDL1 is
  port(A, B: in bit; X: out bit);
end entity VHDL1;
architecture ANDfunction of VHDL1 is
begin
  X <= A and B;
end architecture ANDfunction;
```

(a) VHDL text entry



(b) Equivalent graphic (schematic) entry

**FIGURE 3-56** Examples of design entry of an AND gate.

### In-System Programming (ISP)

Certain CPLDs and FPGAs can be programmed after they have been installed on a system printed circuit board (PCB). After a logic design has been developed and fully tested on a development board, it can then be programmed into a “blank” device that is already soldered onto a system board in which it will be operating. Also, if a design change is required, the device on the system board can be reconfigured to incorporate the design modifications.

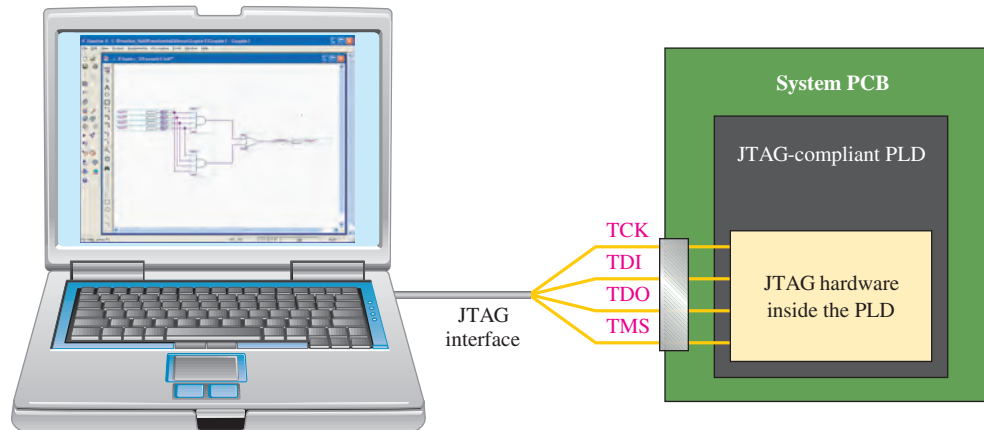
In a production situation, programming a device on the system board minimizes handling and eliminates the need for keeping stocks of preprogrammed devices. It also rules out the possibility of wrong parts being placed in a product. Unprogrammed (blank) devices can

be kept in the warehouse and programmed on-board as needed. This minimizes the capital a business needs for inventories and enhances the quality of its products.

## JTAG

The standard established by the Joint Test Action Group is the commonly used name for IEEE Std. 1149.1. The **JTAG** standard was developed to provide a simple method, called boundary scan, for testing programmable devices for functionality as well as testing circuit boards for bad connections—shorted pins, open pins, bad traces, and the like. Also, JTAG has been used as a convenient way of configuring programmable devices in-system. As the demand for field-upgradable products increases, the use of JTAG as a convenient way of reprogramming CPLDs and FPGAs increases.

JTAG-compliant devices have internal dedicated hardware that interprets instructions and data provided by four dedicated signals. These signals are defined by the JTAG standard to be TDI (Test Data In), TDO (Test Data Out), TMS (Test Mode Select), and TCK (Test Clock). The dedicated JTAG hardware interprets instructions and data on the TDI and TMS signals, and drives data out on the TDO signal. The TCK signal is used to clock the process. A JTAG-compliant PLD is represented in Figure 3–57.



**FIGURE 3–57** Simplified illustration of in-system programming via a JTAG interface.

## Embedded Processor

Another approach to in-system programming is the use of an embedded microprocessor and memory. The processor is embedded within the system along with the CPLD or FPGA and other circuitry, and it is dedicated to the purpose of in-system configuration of the programmable device.

As you have learned, SRAM-based devices are volatile and lose their programmed data when the power is turned *off*. It is necessary to store the programming data in a PROM (programmable read-only memory), which is nonvolatile. When power is turned *on*, the embedded processor takes control of transferring the stored data from the PROM to the CPLD or FPGA.

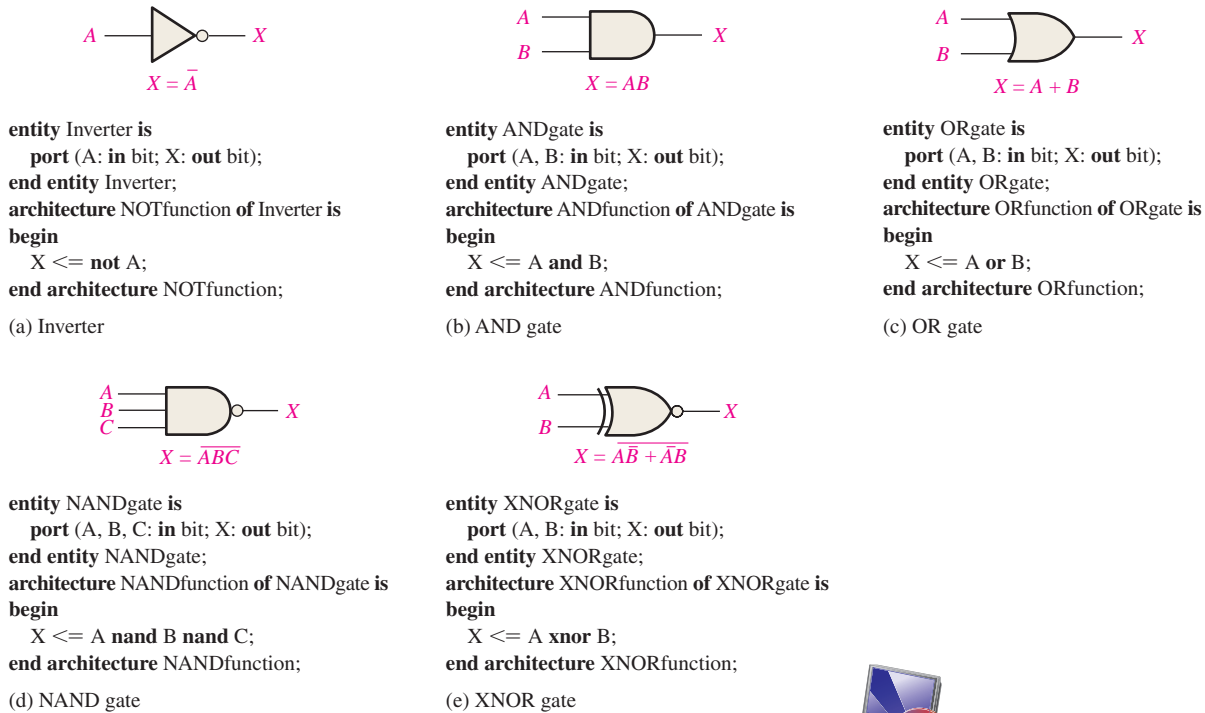
Also, an embedded processor is sometimes used for reconfiguration of a programmable device while the system is running. In this case, design changes are done with software, and the new data are then loaded into a PROM without disturbing the operation of the system. The processor controls the transfer of the data to the device “on-the-fly” at an appropriate time.

## VHDL Descriptions of Logic Gates

Hardware description languages (HDLs) differ from software programming languages because HDLs include ways of describing logic connections and characteristics. An HDL implements a logic design in hardware (PLD), whereas a software programming language, such as C or BASIC, instructs existing hardware what to do. The two standard HDLs used for programming

PLDs are VHDL and Verilog. Both of these HDLs have their advocates, but VHDL will be used in this textbook. A VHDL tutorial is available on the website.

Figure 3–58 shows **VHDL** programs for gates described in this chapter. Two gates are left as Checkup exercises. VHDL has an *entity/architecture* structure. The **entity** defines the logic element and its inputs/outputs or ports; the **architecture** describes the logic operation. Keywords that are part of the VHDL syntax are shown bold for clarity.



**FIGURE 3-58** Logic gates described with VHDL.



**SECTION 3-7 CHECKUP**

1. List six process technologies used for programmable links in programmable logic.
2. What does the term *volatile* mean in relation to PLDs and which process technology is volatile?
3. What are two design entry methods for programming PLDs and FPGAs?
4. Define JTAG.
5. Write a VHDL description of a 3-input NOR gate.
6. Write a VHDL description of an XOR gate.

**3-8 Fixed-Function Logic Gates**

Fixed-function logic integrated circuits have been around for a long time and are available in a variety of logic functions. Unlike a PLD, a fixed-function IC comes with logic functions that cannot be programmed in and cannot be altered. The fixed-function logic is on a much smaller scale than the amount of logic that can be programmed into a PLD. Although the trend in technology is definitely toward programmable logic, fixed-function logic is used in specialized applications where PLDs are not the optimum choice. Fixed-

function logic devices are sometimes called “glue logic” because of their usefulness in tying together larger units of logic such as PLDs in a system.

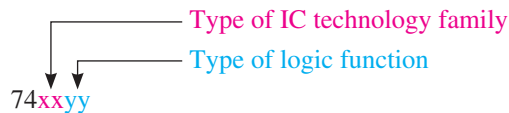
After completing this section, you should be able to

- ◆ List common 74 series gate logic functions
- ◆ List the major integrated circuit technologies and name some integrated circuit families
- ◆ Obtain data sheet information
- ◆ Define *propagation delay time*
- ◆ Define *power dissipation*
- ◆ Define *unit load* and *fan-out*
- ◆ Define *speed-power product*

All of the various fixed-function logic devices currently available are implemented in two major categories of circuit technology: **CMOS** (complementary metal-oxide semiconductor) and **bipolar** (also known as **TTL**, transistor-transistor logic). A type of bipolar technology that is available in very limited devices is ECL (emitter-coupled logic). BiCMOS is another integrated circuit technology that combines both bipolar and CMOS. CMOS is the most dominant circuit technology.

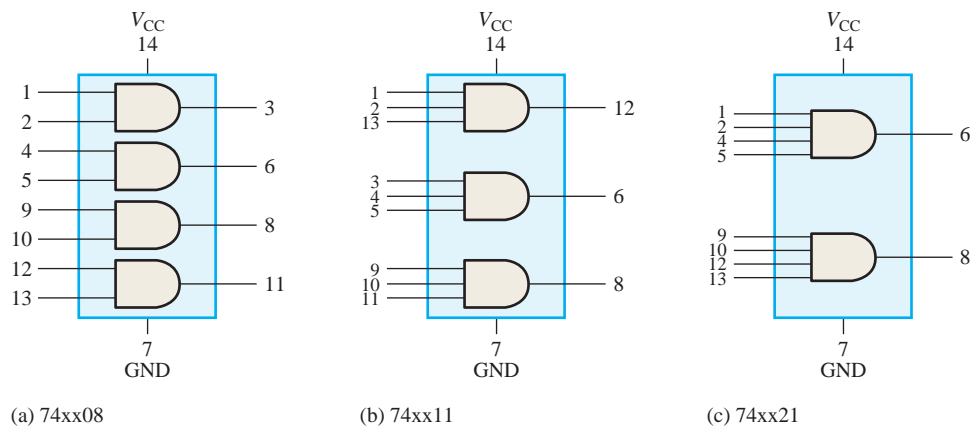
## 74 Series Logic Gate Functions

The 74 series is the standard fixed-function logic devices. The device label format includes one or more letters that identify the type of logic circuit technology family in the IC package and two or more digits that identify the type of logic function. For example, 74HC04 is a fixed-function IC that has six inverters in a package as indicated by 04. The letters, HC, following the prefix 74 identify the circuit technology family as a type of CMOS logic.



### AND Gate

Figure 3–59 shows three configurations of fixed-function AND gates in the 74 series. The 74xx08 is a quad 2-input AND gate device, the 74xx11 is a triple 3-input AND gate device,

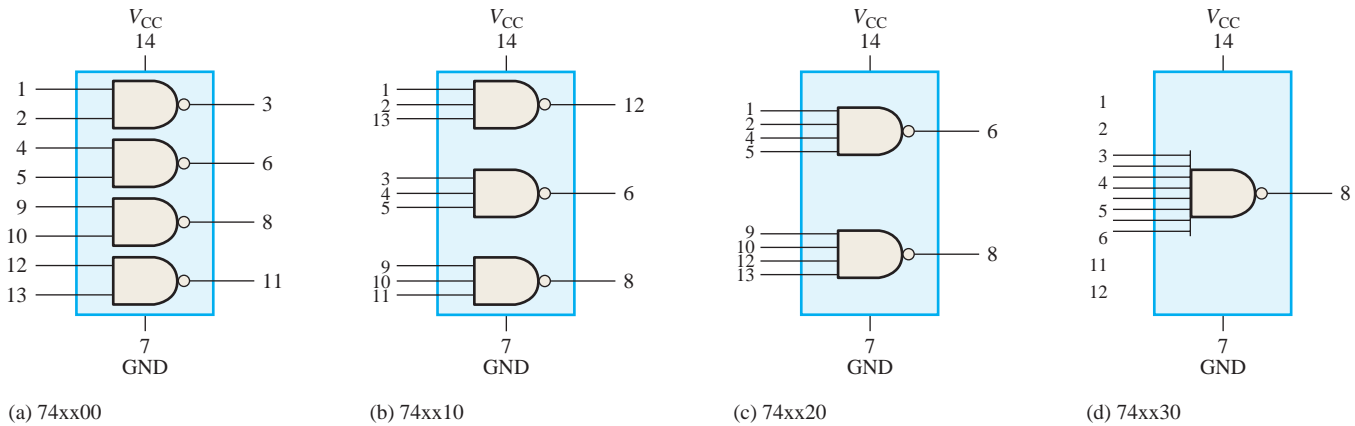


**FIGURE 3–59** 74 series AND gate devices with pin numbers.

and the 74xx21 is a dual 4-input AND gate device. The label xx can represent any of the integrated circuit technology families such as HC or LS. The numbers on the inputs and outputs are the IC package pin numbers.

### NAND Gate

Figure 3–60 shows four configurations of fixed-function NAND gates in the 74 series. The 74xx00 is a quad 2-input NAND gate device, the 74xx10 is a triple 3-input NAND gate device, the 74xx20 is a dual 4-input NAND gate device, and the 74xx30 is a single 8-input NAND gate device.



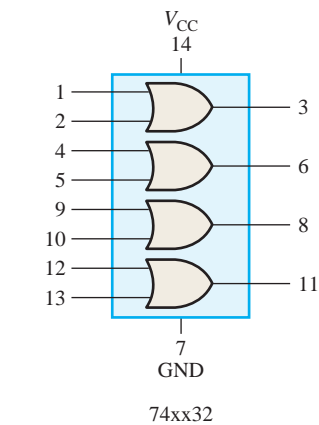
**FIGURE 3–60** 74 series NAND gate devices with package pin numbers.

### OR Gate

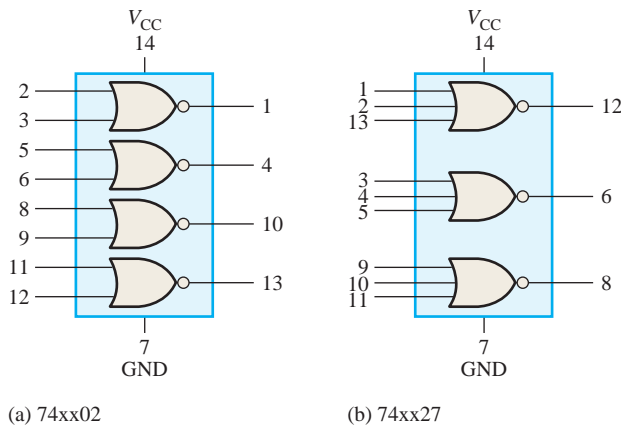
Figure 3–61 shows a fixed-function OR gate in the 74 series. The 74xx32 is a quad 2-input OR gate device.

### NOR Gate

Figure 3–62 shows two configurations of fixed-function NOR gates in the 74 series. The 74xx02 is a quad 2-input NOR gate device, and the 74xx27 is a triple 3-input NOR gate device.



**FIGURE 3–61** 74 series OR gate device.



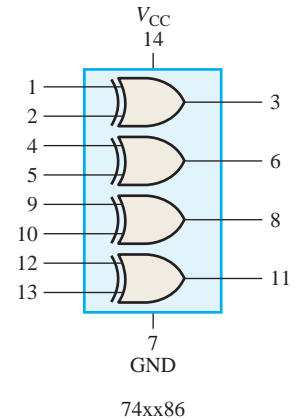
**FIGURE 3–62** 74 series NOR gate devices.

## XOR Gate

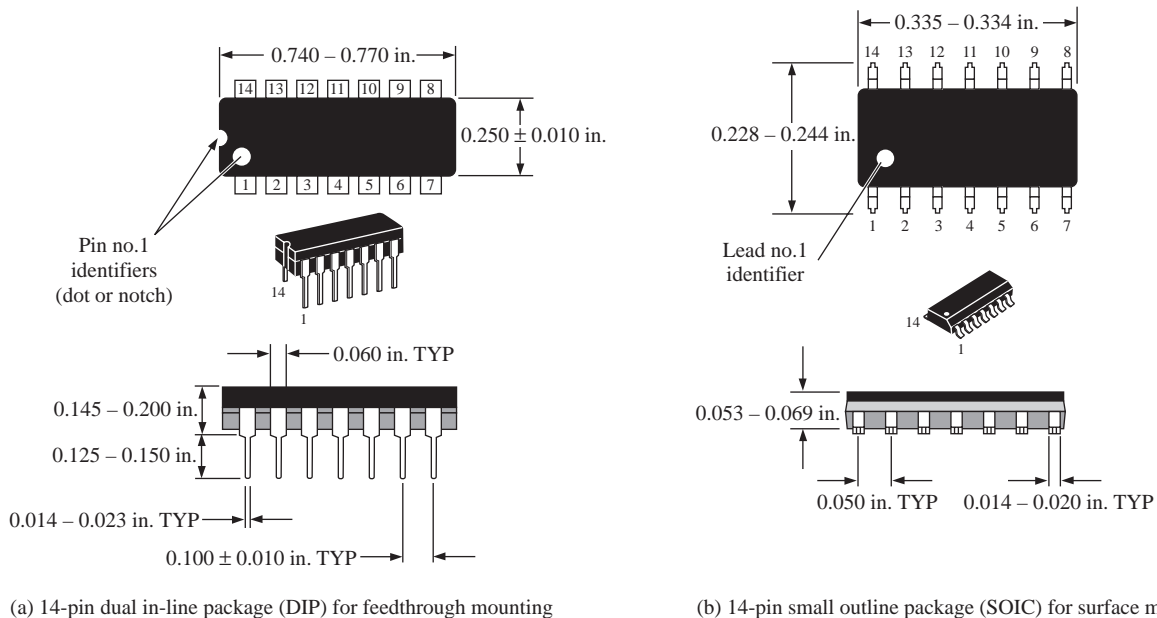
Figure 3–63 shows a fixed-function XOR (exclusive-OR) gate in the 74 series. The 74xx86 is a quad 2-input XOR gate.

## IC Packages

All of the 74 series CMOS are pin-compatible with the same types of devices in bipolar. This means that a CMOS digital IC such as the 74HC00 (quad 2-input NAND), which contains four 2-input NAND gates in one IC package, has the identical package pin numbers for each input and output as does the corresponding bipolar device. Typical IC gate packages, the dual in-line package (DIP) for plug-in or feedthrough mounting and the small-outline integrated circuit (SOIC) package for surface mounting, are shown in Figure 3–64. In some cases, other types of packages are also available. The SOIC package is significantly smaller than the DIP. Packages with a single gate are known as *little logic*. Most logic gate functions are available and are implemented in a CMOS circuit technology. Typically, the gates have only two inputs and have a different designation than multigate devices. For example, the 74xx1G00 is a single 2-input NAND gate.



**FIGURE 3-63** 74 series XOR gate.



**FIGURE 3-64** Typical dual in-line (DIP) and small-outline (SOIC) packages showing pin numbers and basic dimensions.



## Handling Precautions for CMOS

CMOS logic is very sensitive to static charge and can be damaged by ESD (electrostatic discharge) if not handled properly as follows:

1. Store and ship in conductive foam.
2. Connect instruments to earth ground.
3. Connect wrist to earth ground through a large series resistor.
4. Do not remove devices from circuit with power on.
5. Do not apply signal voltage when power is off.

## 74 Series Logic Circuit Families

Although many logic circuit families have become obsolete and some are rapidly on the decline, others are still very active and available. CMOS is the most available and most popular type of logic circuit technology, and the HC (high-speed CMOS) family is the most recommended for new projects. For bipolar, the LS (low-power schottky) family is the most widely used. The HCT, which a variation of the HC family, is compatible with bipolar devices such as LS.

Table 3–14 lists many logic circuit technology families. Because the active status of any given logic family is always in flux, check with a manufacturer, such as Texas Instruments, for information on active/nonactive status and availability for a logic function in a given circuit technology.

**TABLE 3–14**

74 series logic families based on circuit technology.

| Circuit Type | Description                     | Circuit Technology |
|--------------|---------------------------------|--------------------|
| ABT          | Advanced BiCMOS                 | BiCMOS             |
| AC           | Advanced CMOS                   | CMOS               |
| ACT          | Bipolar compatible AC           | CMOS               |
| AHC          | Advanced high-speed CMOS        | CMOS               |
| AHCT         | Bipolar compatible AHC          | CMOS               |
| ALB          | Advanced low-voltage BiCMOS     | BiCMOS             |
| ALS          | Advanced low-power Schottky     | Bipolar            |
| ALVC         | Advanced low-voltage CMOS       | CMOS               |
| AUC          | Advanced ultra-low-voltage CMOS | CMOS               |
| AUP          | Advanced ultra-low-power CMOS   | CMOS               |
| AS           | Advanced Schottky               | Bipolar            |
| AVC          | Advanced very-low-power CMOS    | CMOS               |
| BCT          | Standard BiCMOS                 | BiCMOS             |
| F            | Fast                            | Bipolar            |
| FCT          | Fast CMOS technology            | CMOS               |
| HC           | High-speed CMOS                 | CMOS               |
| HCT          | Bipolar compatible HC           | CMOS               |
| LS           | Low-power Schottky              | Bipolar            |
| LV-A         | Low-voltage CMOS                | CMOS               |
| LV-AT        | Bipolar compatible LV-A         | CMOS               |
| LVC          | Low-voltage CMOS                | CMOS               |
| LVT          | Low-voltage biCMOS              | BiCMOS             |
| S            | Schottky                        | Bipolar            |

The type of integrated circuit technology has nothing to do with the logic function itself. For example, the 74HC00, 74HCT00, and 74LS00 are all quad 2-input NAND gates with identical package pin configurations. The differences among these three logic devices are in the electrical and performance characteristics such as power consumption, dc supply voltage, switching speed, and input/output voltage levels. CMOS and bipolar circuits are implemented with two different types of transistors. Figures 3–65 and 3–66 show partial data sheets for the 74HC00A quad 2-input NAND gate in CMOS and in bipolar technologies, respectively.

## Performance Characteristics and Parameters

Several things define the performance of a logic circuit. These performance characteristics are the switching speed measured in terms of the propagation delay time, the power

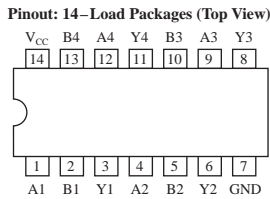
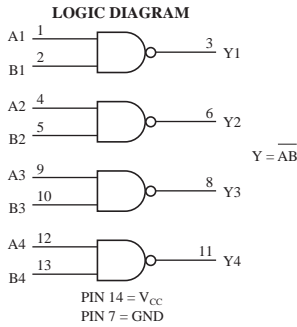
High-speed logic has a short propagation delay time.




### Quad 2-Input NAND Gate High-Performance Silicon-Gate CMOS

The MC54/74HC00A is identical in pinout to the LS00. The device inputs are compatible with Standard CMOS outputs; with pullup resistors, they are compatible with LSTTL outputs.


- Output Drive Capability: 10 LSTTL Loads
- Outputs Directly Interface to CMOS, NMOS and TTL
- Operating Voltage Range: 2 to 6 V
- Low Input Current: 1  $\mu$ A
- High Noise Immunity Characteristic of CMOS Devices
- In Compliance With the JEDEC Standard No. 7A Requirements
- Chip Complexity: 32 FETs or 8 Equivalent Gates




**MC54/74HC00A**




J SUFFIX  
CERAMIC PACKAGE  
CASE 632-08



N SUFFIX  
PLASTIC PACKAGE  
CASE 646-06



D SUFFIX  
SOIC PACKAGE  
CASE 751A-03



DT SUFFIX  
TSSOP PACKAGE  
CASE 946G-01

**ORDERING INFORMATION**

|             |         |
|-------------|---------|
| MC54HCXXAJ  | Ceramic |
| MC74HCXXAN  | Plastic |
| MC74HCXXAD  | SOIC    |
| MC74HCXXADT | TSSOP   |

**FUNCTION TABLE**

| Inputs |   | Output |
|--------|---|--------|
| A      | B | Y      |
| L      | L | H      |
| L      | H | H      |
| H      | L | H      |
| H      | H | L      |

**MAXIMUM RATINGS\***

| Symbol    | Parameter   | Value                  | Unit         |
|-----------|---|------------------------|--------------|
| $V_{CC}$  | DC Supply Voltage (Referenced to GND)                   | -0.5 to +7.0           | V            |
| $V_{in}$  | DC Input Voltage (Referenced to GND)                    | -0.5 to $V_{CC} + 0.5$ | V            |
| $V_{out}$ | DC Output Voltage (Referenced to GND)                   | -0.5 to $V_{CC} + 0.5$ | V            |
| $I_{in}$  | DC Input Current, per Pin                               | $\pm 20$               | mA           |
| $I_{out}$ | DC Output Current, per Pin                              | $\pm 25$               | mA           |
| $I_{CC}$  | DC Supply Current, $V_{CC}$ and GND Pins                | $\pm 50$               | mA           |
| $P_D$     | Power Dissipation in Still Air, Plastic or Ceramic DIP† | 750                    | mW           |
|           |   | 500                    |              |
|           |   | 450                    |              |
| $T_{stg}$ | Storage Temperature                                     | -65 to +150            | $^{\circ}$ C |
| $T_L$     | Lead Temperature, 1 mm from Case for 10 Seconds         | 260                    | $^{\circ}$ C |
|           |   | 300                    |              |

\* Maximum Ratings are those values beyond which damage to the device may occur. Functional operation should be restricted to the Recommended Operating Conditions.  
 † Derating — Plastic DIP: - 10 mW/ $^{\circ}$ C from 65 $^{\circ}$  to 125 $^{\circ}$  C  
 Ceramic DIP: - 10 mW/ $^{\circ}$ C from 100 $^{\circ}$  to 125 $^{\circ}$  C  
 SOIC Package: - 7 mW/ $^{\circ}$ C from 65 $^{\circ}$  to 125 $^{\circ}$  C  
 TSSOP Package: - 6.1 mW/ $^{\circ}$ C from 65 $^{\circ}$  to 125 $^{\circ}$  C

**RECOMMENDED OPERATING CONDITIONS**

| Symbol            | Parameter  | in               | Max      | Unit         |
|-------------------|--|------------------|----------|--------------|
| $V_{CC}$          | DC Supply Voltage (Referenced to GND)                | 2.0              | 6.0      | V            |
| $V_{in}, V_{out}$ | DC Input Voltage, Output Voltage (Referenced to GND) | 0                | $V_{CC}$ | V            |
| $T_A$             | Operating Temperature, All Package Types             | -55              | +125     | $^{\circ}$ C |
| $t_r, t_f$        | Input Rise and Fall Time                             | $V_{CC} = 2.0$ V | 0        | 1000         |
|                   |  | $V_{CC} = 4.5$ V | 0        | 500          |
|                   |  | $V_{CC} = 6.0$ V | 0        | 400          |

**DC CHARACTERISTICS (Voltages Referenced to GND)**

MC54/74HC00A

| Symbol                 | Parameter                                      | Condition   | $V_{CC}$<br>V                 | Guaranteed Limit       |                     |                      | Unit    |
|------------------------|--|---|-------------------------------|------------------------|---------------------|----------------------|---------|
|                        |  |   |                               | -55 to 25 $^{\circ}$ C | $\leq 85^{\circ}$ C | $\leq 125^{\circ}$ C |         |
| $V_{IH}$               | Minimum High-Level Input Voltage               | $V_{out} = 0.1V$ or $V_{CC} - 0.1V$<br>$ I_{out}  \leq 20\mu A$ | 2.0                           | 1.50                   | 1.50                | 1.50                 | V       |
|                        |  |   | 3.0                           | 2.10                   | 2.10                | 2.10                 |         |
|                        |  |   | 4.5                           | 3.15                   | 3.15                | 3.15                 |         |
|                        |  |   | 6.0                           | 4.20                   | 4.20                | 4.20                 |         |
| $V_{IL}$               | Maximum Low-Level Input Voltage                | $V_{out} = 0.1V$ or $V_{CC} - 0.1V$<br>$ I_{out}  \leq 20\mu A$ | 2.0                           | 0.50                   | 0.50                | 0.50                 | V       |
|                        |  |   | 3.0                           | 0.90                   | 0.90                | 0.90                 |         |
|                        |  |   | 4.5                           | 1.35                   | 1.35                | 1.35                 |         |
|                        |  |   | 6.0                           | 1.80                   | 1.80                | 1.80                 |         |
| $V_{OH}$               | Minimum High-Level Output Voltage              | $V_{in} = V_{IH}$ or $V_{IL}$<br>$ I_{out}  \leq 20\mu A$       | 2.0                           | 1.9                    | 1.9                 | 1.9                  | V       |
|                        |  |   | 4.5                           | 4.4                    | 4.4                 | 4.4                  |         |
|                        |  |   | 6.0                           | 5.9                    | 5.9                 | 5.9                  |         |
|                        |  |   | $V_{in} = V_{IH}$ or $V_{IL}$ | $ I_{out}  \leq 2.4mA$ | 3.0                 | 2.48                 |         |
| $ I_{out}  \leq 4.0mA$ | 4.5  | 3.98  |                               | 3.84                   | 3.70                |                      |         |
| $ I_{out}  \leq 5.2mA$ | 6.0  | 5.48  |                               | 5.34                   | 5.20                |                      |         |
| $V_{OL}$               | Maximum Low-Level Output Voltage               | $V_{in} = V_{IH}$ or $V_{IL}$<br>$ I_{out}  \leq 20\mu A$       | 2.0                           | 0.1                    | 0.1                 | 0.1                  | V       |
|                        |  |   | 4.5                           | 0.1                    | 0.1                 | 0.1                  |         |
|                        |  |   | 6.0                           | 0.1                    | 0.1                 | 0.1                  |         |
|                        |  |   | $V_{in} = V_{IH}$ or $V_{IL}$ | $ I_{out}  \leq 2.4mA$ | 3.0                 | 0.26                 |         |
| $ I_{out}  \leq 4.0mA$ | 4.5  | 0.26  |                               | 0.33                   | 0.40                |                      |         |
| $ I_{out}  \leq 5.2mA$ | 6.0  | 0.26  |                               | 0.33                   | 0.40                |                      |         |
| $I_{in}$               | Maximum Input Leakage Current                  | $V_{in} = V_{CC}$ or GND  | 6.0                           | $\pm 0.1$              | $\pm 1.0$           | $\pm 1.0$            | $\mu A$ |
| $I_{CC}$               | Maximum Quiescent Supply Current (per Package) | $V_{in} = V_{CC}$ or GND<br>$I_{out} = 0\mu A$                  | 6.0                           | 1.0                    | 10                  | 40                   | $\mu A$ |

**AC CHARACTERISTICS ( $C_L = 50$  pF, Input  $t_r = t_f = 6$  ns)**

| Symbol             | Parameter   | $V_{CC}$<br>V | Guaranteed Limit       |                     |                      | Unit |
|--------------------|---|---------------|------------------------|---------------------|----------------------|------|
|                    |   |               | -55 to 25 $^{\circ}$ C | $\leq 85^{\circ}$ C | $\leq 125^{\circ}$ C |      |
| $t_{PLH}, t_{PHL}$ | Maximum Propagation Delay, Input A or B to Output Y | 2.0           | 75                     | 95                  | 110                  | ns   |
|                    |   | 3.0           | 30                     | 40                  | 55                   |      |
|                    |   | 4.5           | 15                     | 19                  | 22                   |      |
|                    |   | 6.0           | 13                     | 16                  | 19                   |      |
| $t_{TLH}, t_{THL}$ | Maximum Output Transition Time, Any Output          | 2.0           | 75                     | 95                  | 110                  | ns   |
|                    |   | 3.0           | 27                     | 32                  | 36                   |      |
|                    |   | 4.5           | 15                     | 19                  | 22                   |      |
|                    |   | 6.0           | 13                     | 16                  | 19                   |      |
| $C_{in}$           | Maximum Input Capacitance                           |               | 10                     | 10                  | 10                   | pF   |

|          |  |   |    |
|----------|--|---|----|
| $C_{PD}$ | Power Dissipation Capacitance (Per Buffer) | Typical @ 25 $^{\circ}$ C, $V_{CC} = 5.0$ V, $V_{EE} = 0$ V |    |
|          |  | 22  | pF |

**FIGURE 3-65** CMOS logic. Partial data sheet for a 54/74HC00A quad 2-input NAND gate. The 54 prefix indicates military grade and the 74 prefix indicates commercial grade.

QUAD 2-INPUT NAND GATE

• ESD > 3500 Volts

SN54/74LS00

DC CHARACTERISTICS OVER OPERATING TEMPERATURE RANGE (unless otherwise specified)

| Symbol   | Parameter                                  | Limits |       |      | Unit          | Test Conditions   |
|----------|--|--------|-------|------|---------------|---|
|          |  | Min    | Typ   | Max  |               |   |
| $V_{IH}$ | Input HIGH Voltage                         | 2.0    |       |      | V             | Guaranteed Input HIGH Voltage for All Inputs  |
| $V_{IL}$ | Input LOW Voltage                          | 54     |       | 0.7  | V             | Guaranteed Input LOW Voltage for All Inputs   |
|          |  | 74     |       | 0.8  |               |   |
| $V_{IK}$ | Input Clamp Diode Voltage                  |        | -0.65 | -1.5 | V             | $V_{CC} = \text{MIN}, I_{IN} = -18 \text{ mA}$  |
| $V_{OH}$ | Output HIGH Voltage                        | 54     | 2.5   | 3.5  | V             | $V_{CC} = \text{MIN}, I_{OH} = \text{MAX}, V_{IN} = V_{IH}$ or $V_{IL}$ per Truth Table                 |
|          |  | 74     | 2.7   | 3.5  | V             |   |
| $V_{OL}$ | Output LOW Voltage                         | 54, 74 | 0.25  | 0.4  | V             | $I_{OL} = 4.0 \text{ mA}$ or $V_{CC} = V_{CC \text{ MIN}}, V_{IN} = V_{IL}$ or $V_{IH}$ per Truth Table |
|          |  | 74     | 0.35  | 0.5  | V             |   |
| $I_{IH}$ | Input HIGH Current                         |        |       | 20   | $\mu\text{A}$ | $V_{CC} = \text{MAX}, V_{IN} = 2.7 \text{ V}$   |
|          |  |        |       | 0.1  | $\text{mA}$   | $V_{CC} = \text{MAX}, V_{IN} = 7.0 \text{ V}$   |
| $I_{IL}$ | Input LOW Current                          |        |       | -0.4 | $\text{mA}$   | $V_{CC} = \text{MAX}, I_N = 0.4 \text{ V}$  |
| $I_{OS}$ | Short Circuit Current (Note 1)             |        | -20   | -100 | $\text{mA}$   | $V_{CC} = \text{MAX}$   |
| $I_{CC}$ | Power Supply Current<br>Total, Output HIGH |        |       | 1.6  | $\text{mA}$   | $V_{CC} = \text{MAX}$   |
|          |  |        |       | 4.4  | $\text{mA}$   |   |

NOTE 1: Not more than one output should be shorted at a time, nor for more than 1 second.

AC CHARACTERISTICS ( $T_A = 25^\circ\text{C}$ )


| Symbol    | Parameter                       | Limits |     |     | Unit | Test Conditions                                   |
|-----------|---------------------------------|--------|-----|-----|------|---|
|           |                                 | Min    | Typ | Max |      |   |
| $t_{PLH}$ | Turn-Off Delay, Input to Output |        | 9.0 | 15  | ns   | $V_{CC} = 5.0 \text{ V}$<br>$C_L = 15 \text{ pF}$ |
| $t_{PHL}$ | Turn-On Delay, Input to Output  |        | 10  | 15  | ns   |   |

GUARANTEED OPERATING RANGES

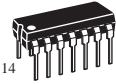
| Symbol   | Parameter                           |        | Min  | Typ | Max  | Unit             |
|----------|-------------------------------------|--------|------|-----|------|------------------|
| $V_{CC}$ | Supply Voltage                      | 54     | 4.5  | 5.0 | 5.5  | V                |
|          |                                     | 74     | 4.75 | 5.0 | 5.25 |                  |
| $T_A$    | Operating Ambient Temperature Range | 54     | -55  | 25  | 125  | $^\circ\text{C}$ |
|          |                                     | 74     | 0    | 25  | 70   |                  |
| $I_{OH}$ | Output Current — High               | 54, 74 |      |     | -0.4 | $\text{mA}$      |
| $I_{OL}$ | Output Current — Low                | 54     |      |     | 4.0  | $\text{mA}$      |
|          |                                     | 74     |      |     | 8.0  |                  |

**SN54/74LS00**


**QUAD 2-INPUT NAND GATE  
LOW POWER SCHOTTKY**



J SUFFIX  
CERAMIC  
CASE 632-08



N SUFFIX  
PLASTIC  
CASE 646-06



D SUFFIX  
SOIC  
CASE 751A-02

ORDERING INFORMATION  
SN54L.SXXJ Ceramic  
SN74L.SXXN Plastic  
SN74L.SXXD SOIC

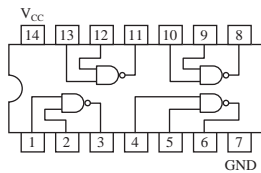


FIGURE 3-66 Bipolar logic. Partial data sheet for a 54/74LS00 quad 2-input NAND gate.

dissipation, the fan-out or drive capability, the speed-power product, the dc supply voltage, and the input/output logic levels.

Propagation Delay Time

This parameter is a result of the limitation on switching speed or frequency at which a logic circuit can operate. The terms *low speed* and *high speed*, applied to logic circuits, refer to the propagation delay time. The shorter the propagation delay, the higher the switching speed of the circuit and thus the higher the frequency at which it can operate.

**Propagation delay time,  $t_p$** , of a logic gate is the time interval between the transition of an input pulse and the occurrence of the resulting transition of the output pulse. There are two different measurements of propagation delay time associated with a logic gate that apply to all the types of basic gates:

- $t_{PHL}$ : The time between a specified reference point on the input pulse and a corresponding reference point on the resulting output pulse, with the output changing from the HIGH level to the LOW level (HL).
- $t_{PLH}$ : The time between a specified reference point on the input pulse and a corresponding reference point on the resulting output pulse, with the output changing from the LOW level to the HIGH level (LH).

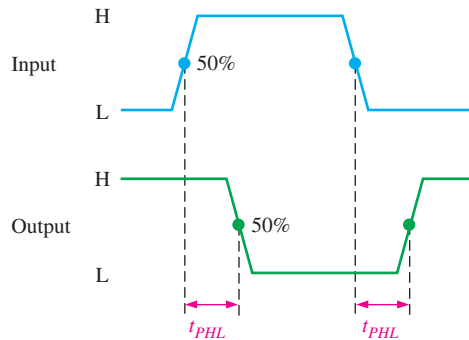
For the HCT family CMOS, the propagation delay is 7 ns, for the AC family it is 5 ns, and for the ALVC family it is 3 ns. For standard-family bipolar (TTL) gates, the typical propagation delay is 11 ns and for F family gates it is 3.3 ns. All specified values are dependent on certain operating conditions as stated on a data sheet.

**EXAMPLE 3-23**

Show the propagation delay times of an inverter.

**Solution**

An input/output pulse of an inverter is shown in Figure 3-67, and the propagation delay times,  $t_{PHL}$  and  $t_{PLH}$ , are indicated. In this case, the delays are measured between the 50% points of the corresponding edges of the input and output pulses. The values of  $t_{PHL}$  and  $t_{PLH}$  are not necessarily equal but in many cases they are the same.



**FIGURE 3-67**

**Related Problem**

One type of logic gate has a specified maximum  $t_{PLH}$  and  $t_{PHL}$  of 10 ns. For another type of gate the value is 4 ns. Which gate can operate at the highest frequency?

**DC Supply Voltage ( $V_{CC}$ )**

The typical dc supply voltage for CMOS logic is either 5 V, 3.3 V, 2.5 V, or 1.8 V, depending on the category. An advantage of CMOS is that the supply voltages can vary over a wider range than for bipolar logic. The 5 V CMOS can tolerate supply variations from 2 V to 6 V and still operate properly although propagation delay time and power dissipation are significantly affected. The 3.3 V CMOS can operate with supply voltages from 2 V to 3.6 V. The typical dc supply voltage for bipolar logic is 5.0 V with a minimum of 4.5 V and a maximum of 5.5 V.

**Power Dissipation**

The **power dissipation**,  $P_D$ , of a logic gate is the product of the dc supply voltage and the average supply current. Normally, the supply current when the gate output is LOW is greater than when the gate output is HIGH. The manufacturer's data sheet usually designates the supply current for the LOW output state as  $I_{CCL}$  and for the HIGH state as  $I_{CCH}$ . The average supply current is determined based on a 50% duty cycle (output LOW half the time and HIGH half the time), so the average power dissipation of a logic gate is

$$P_D = V_{CC} \left( \frac{I_{CCH} + I_{CCL}}{2} \right) \quad \text{Equation 3-2}$$

CMOS gates have very low power dissipations compared to the bipolar family. However, the power dissipation of CMOS is dependent on the frequency of operation. At zero frequency the quiescent power is typically in the microwatt/gate range, and at the maximum operating frequency it can be in the low milliwatt range; therefore, power is sometimes specified at a given frequency. The HC family, for example, has a power of 2.75  $\mu\text{W}$ /gate at 0 Hz (quiescent) and 600  $\mu\text{W}$ /gate at 1 MHz.

A lower power dissipation means less current from the dc supply.

Power dissipation for bipolar gates is independent of frequency. For example, the ALS family uses 1.4 mW/gate regardless of the frequency and the F family uses 6 mW/gate.

### Input and Output Logic Levels

$V_{IL}$  is the LOW level input voltage for a logic gate, and  $V_{IH}$  is the HIGH level input voltage. The 5 V CMOS accepts a maximum voltage of 1.5 V as  $V_{IL}$  and a minimum voltage of 3.5 V as  $V_{IH}$ . Bipolar logic accepts a maximum voltage of 0.8 V as  $V_{IL}$  and a minimum voltage of 2 V as  $V_{IH}$ .

$V_{OL}$  is the LOW level output voltage and  $V_{OH}$  is the HIGH level output voltage. For 5 V CMOS, the maximum  $V_{OL}$  is 0.33 V and the minimum  $V_{OH}$  is 4.4 V. For bipolar logic, the maximum  $V_{OL}$  is 0.4 V and the minimum  $V_{OH}$  is 2.4 V. All values depend on operating conditions as specified on the data sheet.

### Speed-Power Product (SPP)

This parameter (**speed-power product**) can be used as a measure of the performance of a logic circuit taking into account the propagation delay time and the power dissipation. It is especially useful for comparing the various logic gate series within the CMOS and bipolar technology families or for comparing a CMOS gate to a TTL gate.

The SPP of a logic circuit is the product of the propagation delay time and the power dissipation and is expressed in joules (J), which is the unit of energy. The formula is

$$SPP = t_p P_D \quad \text{Equation 3-3}$$

#### EXAMPLE 3-24

A certain gate has a propagation delay of 5 ns and  $I_{CCH} = 1$  mA and  $I_{CCL} = 2.5$  mA with a dc supply voltage of 5 V. Determine the speed-power product.

#### Solution

$$P_D = V_{CC} \left( \frac{I_{CCH} + I_{CCL}}{2} \right) = 5 \text{ V} \left( \frac{1 \text{ mA} + 2.5 \text{ mA}}{2} \right) = 5 \text{ V}(1.75 \text{ mA}) = 8.75 \text{ mW}$$

$$SPP = (5 \text{ ns})(8.75 \text{ mW}) = \mathbf{43.75 \text{ pJ}}$$

#### Related Problem

If the propagation delay of a gate is 15 ns and its SPP is 150 pJ, what is its average power dissipation?

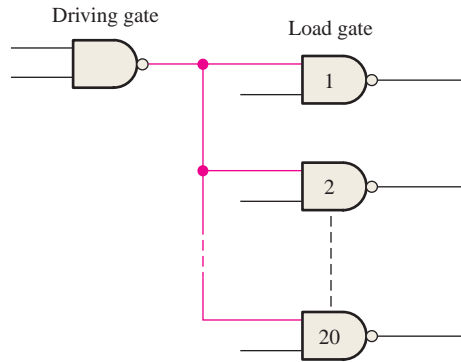
### Fan-Out and Loading

The **fan-out** of a logic gate is the maximum number of inputs of the same series in an IC family that can be connected to a gate's output and still maintain the output voltage levels within specified limits. Fan-out is a significant parameter only for bipolar logic because of the type of circuit technology. Since very high impedances are associated with CMOS circuits, the fan-out is very high but depends on frequency because of capacitive effects.

Fan-out is specified in terms of **unit loads**. A unit load for a logic gate equals one input to a like circuit. For example, a unit load for a 74LS00 NAND gate equals *one* input to another logic gate in the 74LS family (not necessarily a NAND gate). Because the current from a LOW input ( $I_{IL}$ ) of a 74LS00 gate is 0.4 mA and the current that a LOW output ( $I_{OL}$ ) can accept is 8.0 mA, the number of unit loads that a 74LS00 gate can drive in the LOW state is

$$\text{Unit loads} = \frac{I_{OL}}{I_{IL}} = \frac{8.0 \text{ mA}}{0.4 \text{ mA}} = 20$$

A higher fan-out means that a gate output can be connected to more gate inputs.

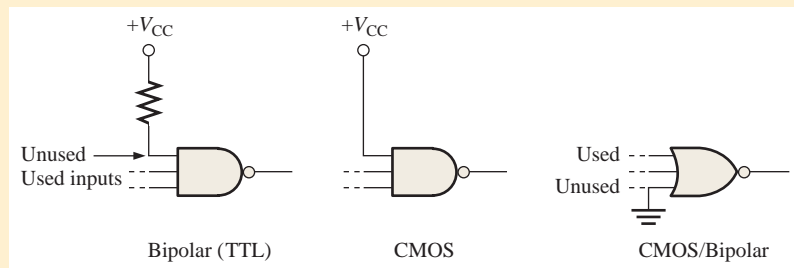


**FIGURE 3–68** The LS family NAND gate output fans out to a maximum of 20 LS family gate inputs.

Figure 3–68 shows LS logic gates driving a number of other gates of the same circuit technology, where the number of gates depends on the particular circuit technology. For example, as you have seen, the maximum number of gate inputs (unit loads) that a 74LS family bipolar gate can drive is 20.



Unused gate inputs for bipolar (TTL) and CMOS should be connected to the appropriate logic level (HIGH or LOW). For AND/NAND, it is recommended that unused inputs be connected to  $V_{CC}$  (through a 1.0 k $\Omega$  resistor with bipolar) and for OR/NOR, unused inputs should be connected to ground.



### SECTION 3–8 CHECKUP

- How is fixed-function logic different than PLD logic?
- List the two types of IC technologies that are the most widely used.
- Identify the following IC logic designators:
  - LS
  - HC
  - HCT
- Which IC technology generally has the lowest power dissipation?
- What does the term *hex inverter* mean? What does *quad 2-input NAND* mean?
- A positive pulse is applied to an inverter input. The time from the leading edge of the input to the leading edge of the output is 10 ns. The time from the trailing edge of the input to the trailing edge of the output is 8 ns. What are the values of  $t_{PLH}$  and  $t_{PHL}$ ?
- A certain gate has a propagation delay time of 6 ns and a power dissipation of 3 mW. Determine the speed-power product?
- Define  $I_{CCL}$  and  $I_{CCH}$ .
- Define  $V_{IL}$  and  $V_{IH}$ .
- Define  $V_{OL}$  and  $V_{OH}$ .

## 3-9 Troubleshooting



Troubleshooting is the process of recognizing, isolating, and correcting a fault or failure in a circuit or system. To be an effective troubleshooter, you must understand how the circuit or system is supposed to work and be able to recognize incorrect performance. For example, to determine whether or not a certain logic gate is faulty, you must know what the output should be for given inputs.

After completing this section, you should be able to

- ◆ Test for internally open inputs and outputs in IC gates
- ◆ Recognize the effects of a shorted IC input or output
- ◆ Test for external faults on a PCB board
- ◆ Troubleshoot a simple frequency counter using an oscilloscope

### Internal Failures of IC Logic Gates

Opens and shorts are the most common types of internal gate failures. These can occur on the inputs or on the output of a gate inside the IC package. *Before attempting any troubleshooting, check for proper dc supply voltage and ground.*

#### Effects of an Internally Open Input

An internal open is the result of an open component on the chip or a break in the tiny wire connecting the IC chip to the package pin. An open input prevents a signal on that input from getting to the output of the gate, as illustrated in Figure 3-69(a) for the case of a 2-input NAND gate. An open TTL (bipolar) input acts effectively as a HIGH level, so pulses applied to the good input get through to the NAND gate output as shown in Figure 3-69(b).



(a) Application of pulses to the open input will produce no pulses on the output.

(b) Application of pulses to the good input will produce output pulses for bipolar NAND and AND gates because an open input typically acts as a HIGH. It is uncertain for CMOS.

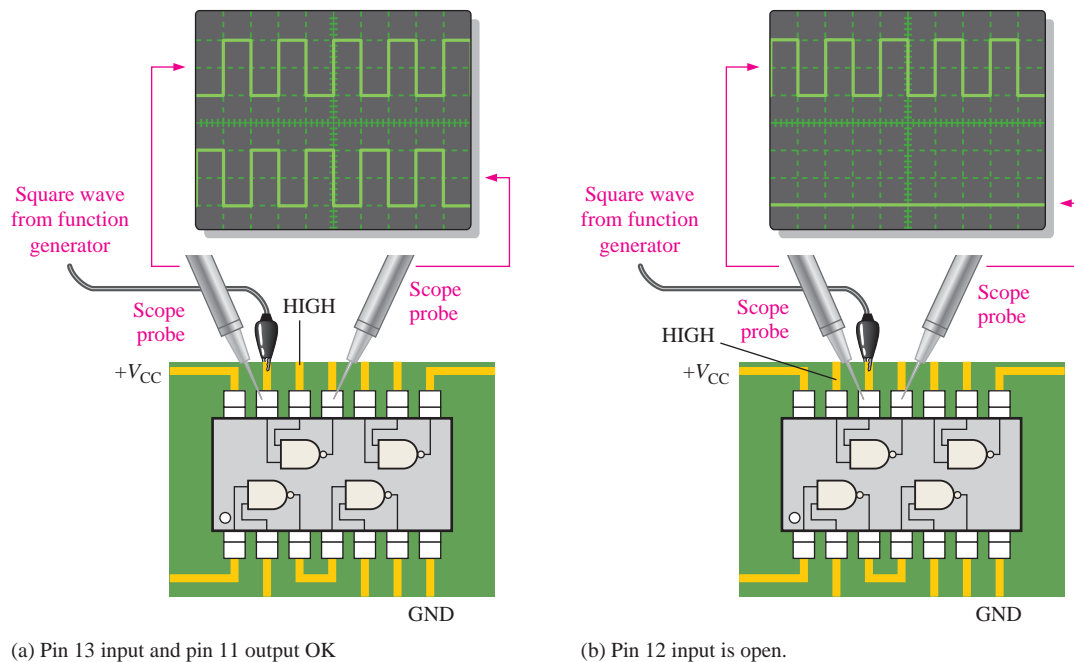
**FIGURE 3-69** The effect of an open input on a NAND gate.

### Conditions for Testing Gates

When testing a NAND gate or an AND gate, always make sure that the inputs that are not being pulsed are HIGH to enable the gate. When checking a NOR gate or an OR gate, always make sure that the inputs that are not being pulsed are LOW. When checking an XOR or XNOR gate, the level of the nonpulsed input does not matter because the pulses on the other input will force the inputs to alternate between the same level and opposite levels.

### Troubleshooting an Open Input

Troubleshooting this type of failure is easily accomplished with an oscilloscope and function generator, as demonstrated in Figure 3-70 for the case of a quad 2-input NAND gate package. When measuring digital signals with a scope, always use dc coupling.



**FIGURE 3-70** Troubleshooting a NAND gate for an open input.

The first step in troubleshooting an IC that is suspected of being faulty is to make sure that the dc supply voltage ( $V_{CC}$ ) and ground are at the appropriate pins of the IC. Next, apply continuous pulses to one of the inputs to the gate, making sure that the other input is HIGH (in the case of a NAND gate). In Figure 3-70(a), start by applying a pulse waveform to pin 13, which is one of the inputs to the suspected gate. If a pulse waveform is indicated on the output (pin 11 in this case), then the pin 13 input is not open. By the way, this also proves that the output is not open. Next, apply the pulse waveform to the other gate input (pin 12), making sure the other input is HIGH. There is no pulse waveform on the output at pin 11 and the output is LOW, indicating that the pin 12 input is open, as shown in Figure 3-70(b). The input not being pulsed must be HIGH for the case of a NAND gate or AND gate. If this were a NOR gate, the input not being pulsed would have to be LOW.

### Effects of an Internally Open Output

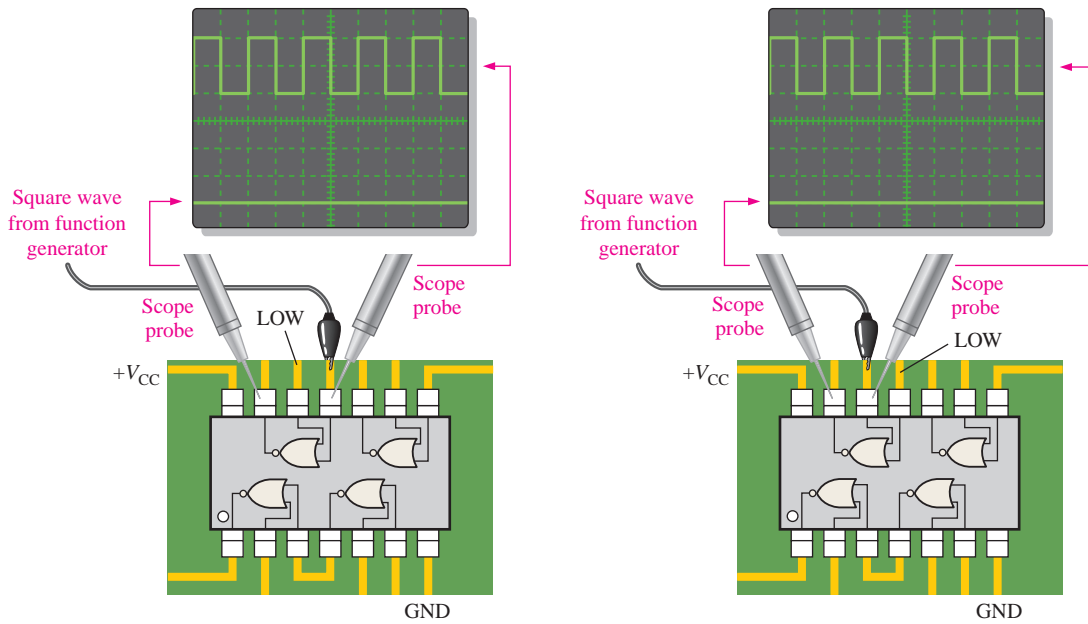
An internally open gate output prevents a signal on any of the inputs from getting to the output. Therefore, no matter what the input conditions are, the output is unaffected. The level at the output pin of the IC will depend upon what it is externally connected to. It could be either HIGH, LOW, or floating (not fixed to any reference). In any case, there will be no signal on the output pin.

### Troubleshooting an Open Output

Figure 3-71 illustrates troubleshooting an open NOR gate output. In part (a), one of the inputs of the suspected gate (pin 11 in this case) is pulsed, and the output (pin 13) has no pulse waveform. In part (b), the other input (pin 12) is pulsed and again there is no pulse waveform on the output. Under the condition that the input that is not being pulsed is at a LOW level, this test shows that the output is internally open.

### Shorted Input or Output

Although not as common as an open, an internal short to the dc supply voltage, ground, another input, or an output can occur. When an input or output is shorted to the supply voltage, it will be stuck in the HIGH state. If an input or output is shorted to ground, it will be



(a) Pulse input on pin 11. No pulse output.

(b) Pulse input on pin 12. No pulse output.

**FIGURE 3-71** Troubleshooting a NOR gate for an open output.

stuck in the LOW state (0 V). If two inputs or an input and an output are shorted together, they will always be at the same level.

### External Opens and Shorts

Many failures involving digital ICs are due to faults that are external to the IC package. These include bad solder connections, solder splashes, wire clippings, improperly etched printed circuit boards (PCBs), and cracks or breaks in wires or printed circuit interconnections. These open or shorted conditions have the same effect on the logic gate as the internal faults, and troubleshooting is done in basically the same ways. A visual inspection of any circuit that is suspected of being faulty is the first thing a technician should do.

#### EXAMPLE 3-25

You are checking a 74LS10 triple 3-input NAND gate IC that is one of many ICs located on a PCB. You have checked pins 1 and 2 and they are both HIGH. Now you apply a pulse waveform to pin 13, and place your scope probe first on pin 12 and then on the connecting PCB trace, as indicated in Figure 3-72. Based on your observation of the scope screen, what is the most likely problem?

#### Solution

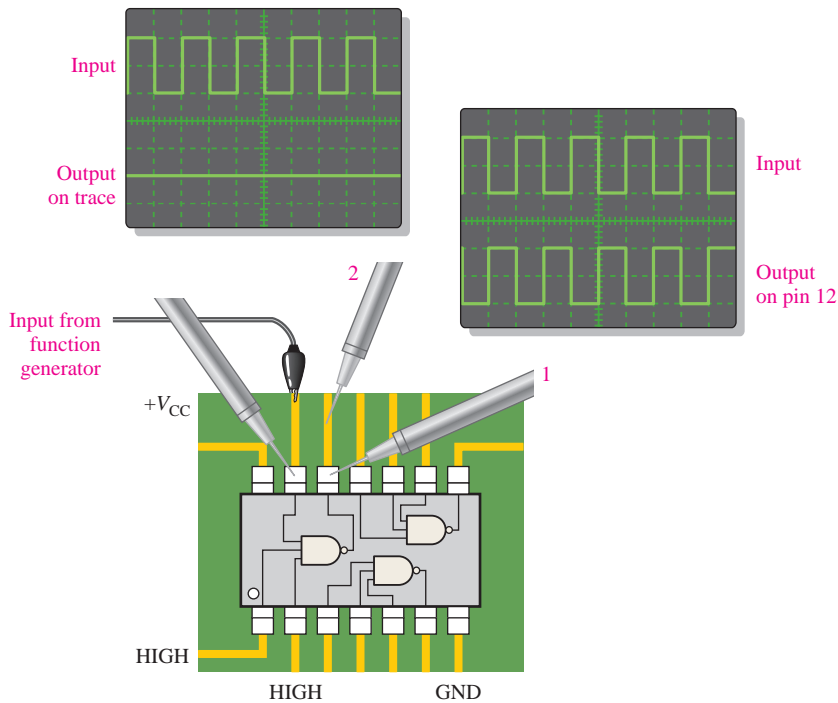
The waveform with the probe in position 1 shows that there is pulse activity on the gate output at pin 12, but there are no pulses on the PCB trace as indicated by the probe in position 2. The gate is working properly, but the signal is not getting from pin 12 of the IC to the PCB trace.

Most likely there is a bad solder connection between pin 12 of the IC and the PCB, which is creating an open. You should resolder that point and check it again.

#### Related Problem

If there are no pulses at either probe position 1 or 2 in Figure 3-72, what fault(s) does this indicate?





**FIGURE 3-72**

In most cases, you will be troubleshooting ICs that are mounted on PCBs or prototype assemblies and interconnected with other ICs. As you progress through this book, you will learn how different types of digital ICs are used together to perform system functions. At this point, however, we are concentrating on individual IC gates. This limitation does not prevent us from looking at the system concept at a very basic and simplified level.

To continue the emphasis on systems, Examples 3-26 and 3-27 deal with troubleshooting the frequency counter that was introduced in Section 3-2.

**EXAMPLE 3-26**

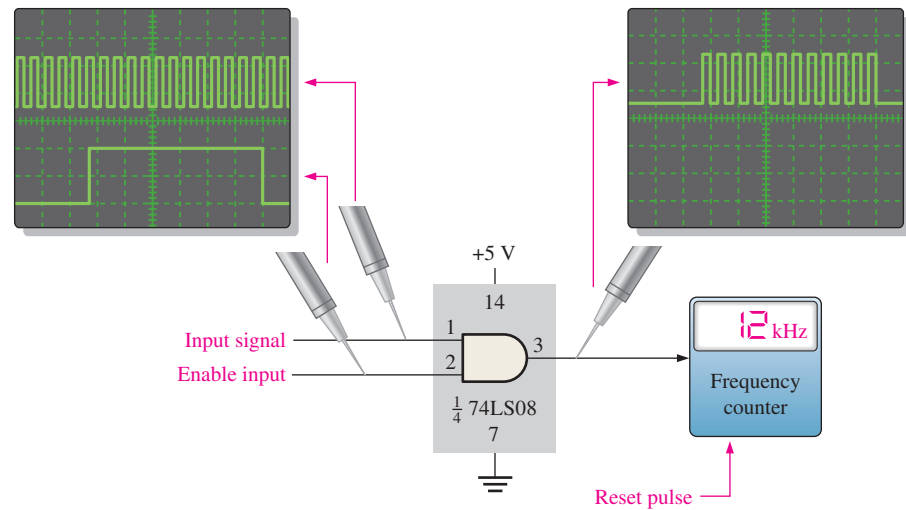
After trying to operate the frequency counter shown in Figure 3-73, you find that it constantly reads out all 0s on its display, regardless of the input frequency. Determine the cause of this malfunction. The enable pulse has a width of 1 ms.

Figure 3-73(a) gives an example of how the frequency counter should be working with a 12 kHz pulse waveform on the input to the AND gate. Part (b) shows that the display is improperly indicating 0 Hz.

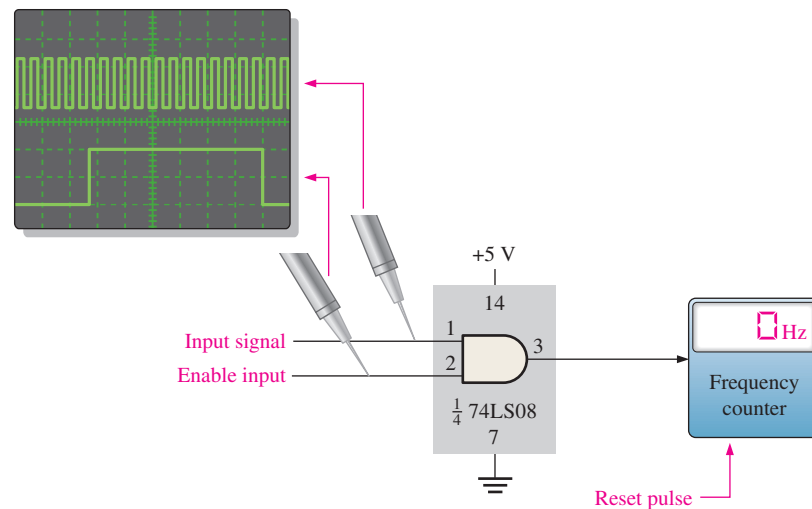
**Solution**

Three possible causes are

1. A constant active or asserted level on the counter reset input, which keeps the counter at zero.
2. No pulse signal on the input to the counter because of an internal open or short in the counter. This problem would keep the counter from advancing after being reset to zero.



(a) The counter is working properly.



(b) The counter is not measuring a frequency.

**FIGURE 3-73**

- No pulse signal on the input to the counter because of an open AND gate output or the absence of input signals, again keeping the counter from advancing from zero.

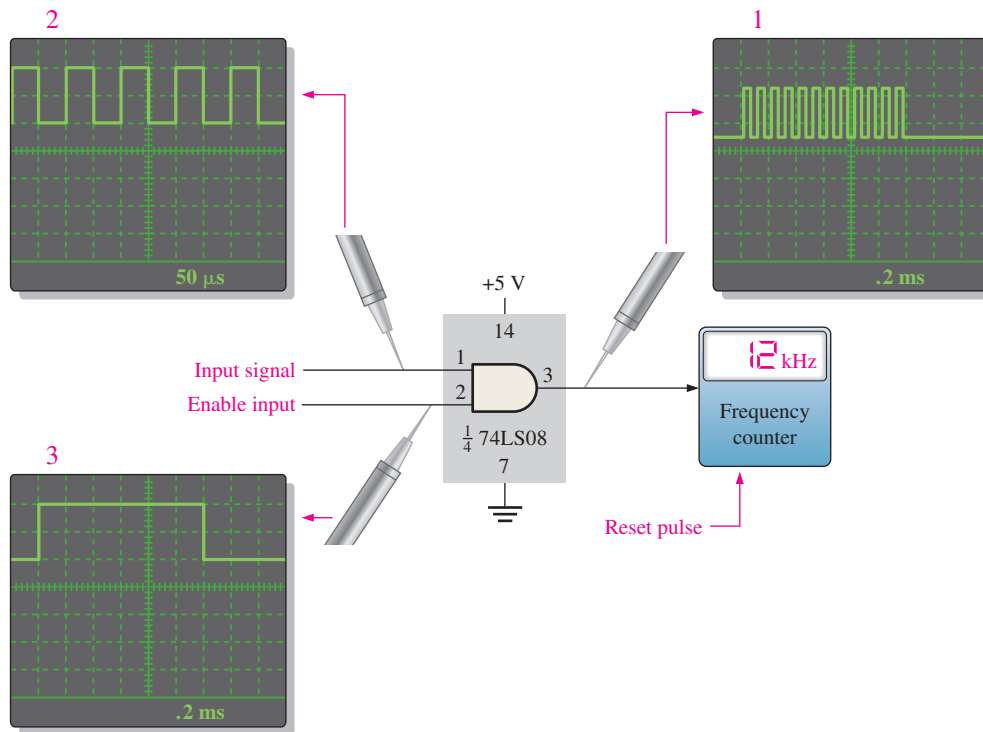
The first step is to make sure that  $V_{CC}$  and ground are connected to all the right places; assume that they are found to be okay. Next, check for pulses on both inputs to the AND gate. The scope indicates that there are proper pulses on both of these inputs. A check of the counter reset shows a LOW level which is known to be the unasserted level and, therefore, this is not the problem. The next check on pin 3 of the 74LS08 shows that there are no pulses on the output of the AND gate, indicating that the gate output is open. Replace the 74LS08 IC and check the operation again.

**Related Problem**

If pin 2 of the 74LS08 AND gate is open, what indication should you see on the frequency display?

**EXAMPLE 3-27**

The frequency counter shown in Figure 3-74 appears to measure the frequency of input signals incorrectly. It is found that when a signal with a precisely known frequency is applied to pin 1 of the AND gate, the oscilloscope display indicates a higher frequency. Determine what is wrong. The readings on the screen indicate time per division.


**FIGURE 3-74**
**Solution**

Recall from Section 3-2 that the input pulses were allowed to pass through the AND gate for exactly 1 ms. The number of pulses counted in 1 ms is equal to the frequency in hertz. Therefore, the 1 ms interval, which is produced by the enable pulse on pin 2 of the AND gate, is very critical to an accurate frequency measurement. The enable pulses are produced internally by a precision oscillator circuit. The pulse must be exactly 1 ms in width and in this case it occurs every 3 ms to update the count. Just prior to each enable pulse, the counter is reset to zero so that it starts a new count each time.

Since the counter appears to be counting more pulses than it should to produce a frequency readout that is too high, the enable pulse is the primary suspect. Exact time-interval measurements must be made on the oscilloscope.

An input pulse waveform of exactly 10 kHz is applied to pin 1 of the AND gate and the frequency counter incorrectly shows 12 kHz. The first scope measurement, on the output of the AND gate, shows that there are 12 pulses for each enable pulse. In the second scope measurement, the input frequency is verified to be precisely 10 kHz (period = 100  $\mu$ s). In the third scope measurement, the width of the enable pulse is found to be 1.2 ms rather than 1 ms.

The conclusion is that the enable pulse is out of calibration for some reason.

**Related Problem**

What would you suspect if the readout were indicating a frequency less than it should be?



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

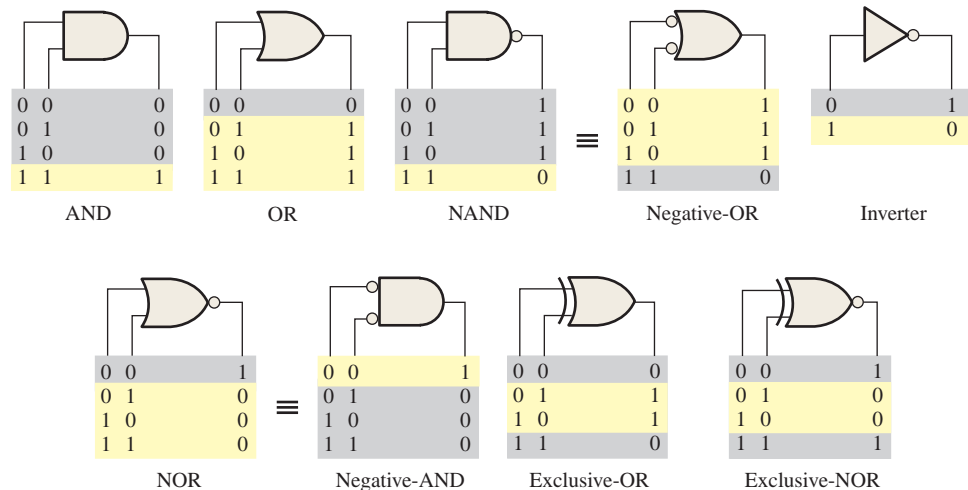
Also, for accurate measurements, make sure that the ground in the circuit you are testing is the same as the scope ground. This can be done by connecting the ground lead on the scope probe to a known ground point in the circuit, such as the metal chassis or a ground point on the PCB. You can also connect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

**SECTION 3-9 CHECKUP**

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

**SUMMARY**

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



Note: Active states are shown in yellow.

**FIGURE 3-75**