Lecture No.4

LECTURE OUTLINE

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.

Standard logic symbols for the **inverter** are shown in Figure 3–1.

FIGURE 3–1 Standard logic symbol for the inverter

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 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.

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.

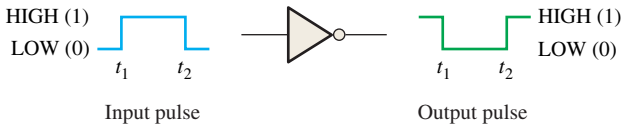


FIGURE 3–2 Inverter operation with a pulse input. Open file F03-02 to verify inverter operation. *A Multisim tutorial is available on the website.*

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.

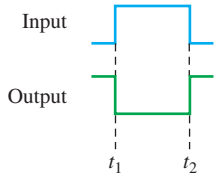


FIGURE 3–3 Timing diagram for the case in Figure 3–2.

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?

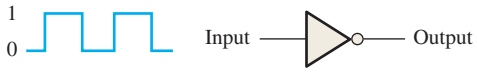


FIGURE 3–4

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**.

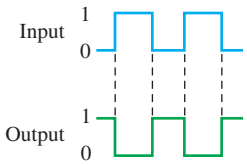


FIGURE 3–5

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 let-ters 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

$$X = \overline{A}$$

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 $\overline{A}$ can be read as "A bar" or "not A."

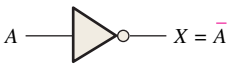


FIGURE 3–6 The inverter complements an input variable.

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.

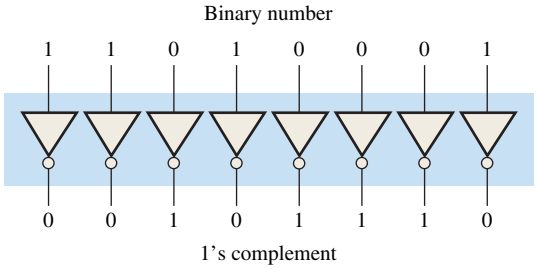


FIGURE 3–7 Example of a 1's complement circuit using inverters.

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.

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 out-put, as indicated by the standard logic symbol 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.

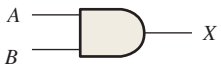


FIGURE 3–8 Standard logic symbol for the AND gate showing two inputs

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 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.

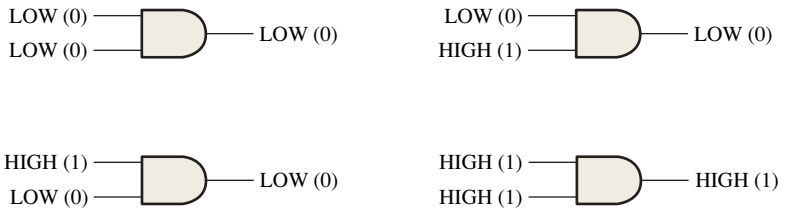


FIGURE 3–9 All possible logic levels for a 2-input AND gate. Open file F03-09 to verify AND gate operation.

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:

$$N = 2^n \qquad \text{Equation 3–1}$$

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.

TABLE 3–2
Truth table for a 2-input AND gate.

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

1 = HIGH, 0 = LOW

TABLE 3–3

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

EXAMPLE 3–2

(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) $N = 2^4 = \mathbf{16}$. There are 16 possible combinations of input bits 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.*

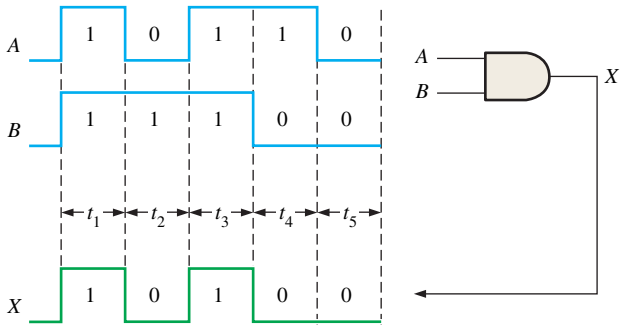
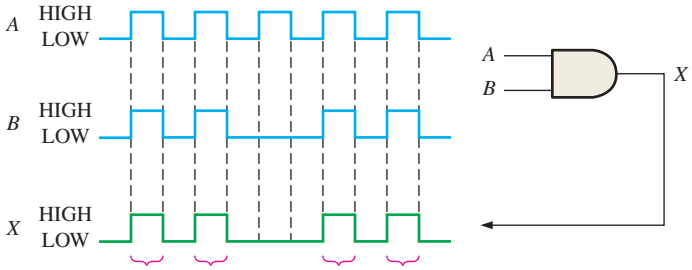


FIGURE 3–10 Example of AND gate operation with a timing diagram showing input and output relationships.

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?



A and *B* are both HIGH during these four time intervals; therefore, *X* is HIGH.

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.

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.

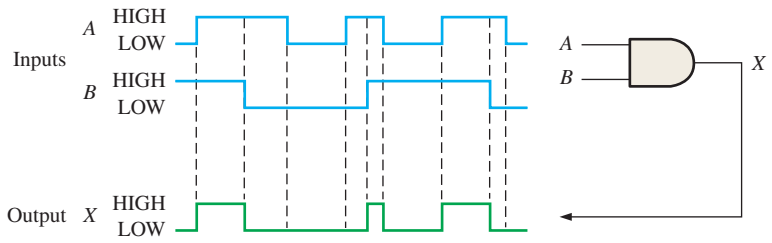


FIGURE 3–12

Solution

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

EXAMPLE 3–5

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

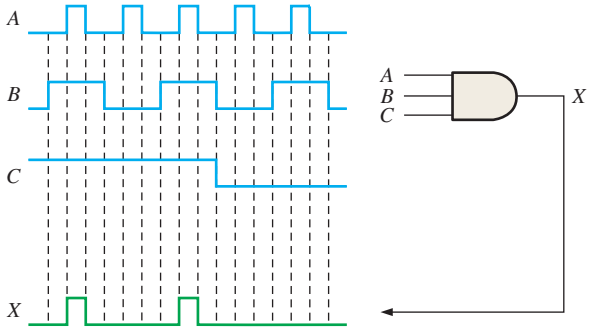


FIGURE 3–13

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.

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.

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

$$0 \cdot 0 = 0$$
$$0 \cdot 1 = 0$$
$$1 \cdot 0 = 0$$
$$1 \cdot 1 = 1$$

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.

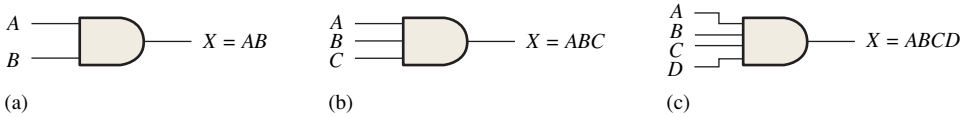


FIGURE 3–15 Boolean expressions for AND gates with two, three, and four inputs.

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

A	B	$AB = X$
0	0	$0 \cdot 0 = 0$
0	1	$0 \cdot 1 = 0$
1	0	$1 \cdot 0 = 0$
1	1	$1 \cdot 1 = 1$

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.

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.

FIGURE 3–18 Standard logic symbol for the OR gate showing two inputs

Operation of an OR Gate

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.

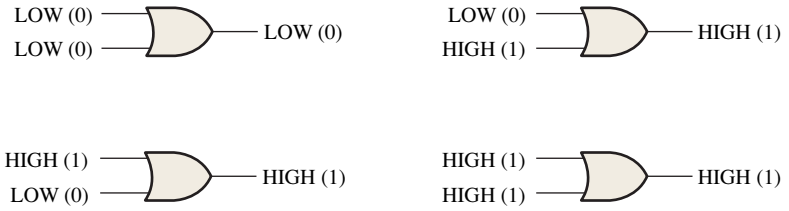


FIGURE 3–19 All possible logic levels for a 2-input OR gate. Open file F03-19 to verify OR gate operation.

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).

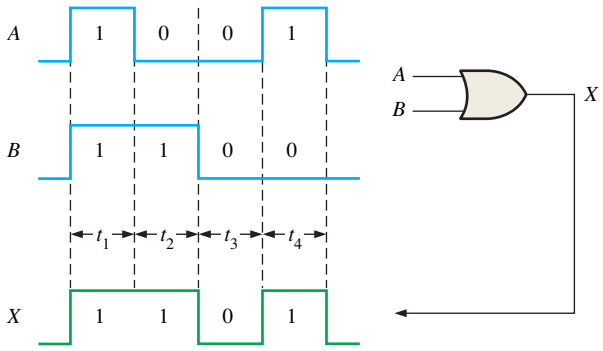


FIGURE 3–20 Example of OR gate operation with a timing diagram showing input and output time relationships.

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?

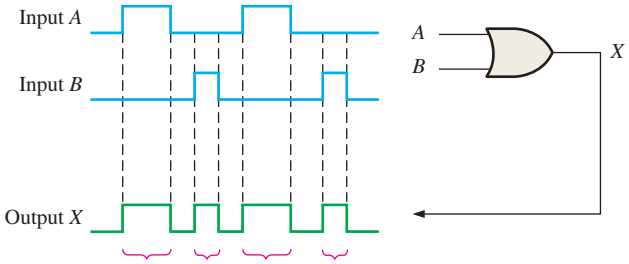


FIGURE 3–21

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.

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.

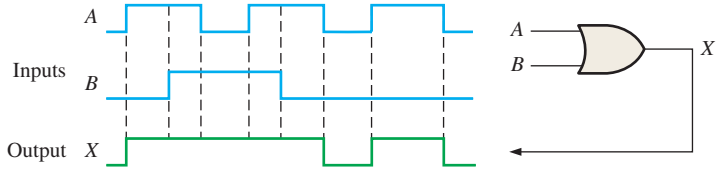


FIGURE 3–22

Solution

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

EXAMPLE 3–9

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

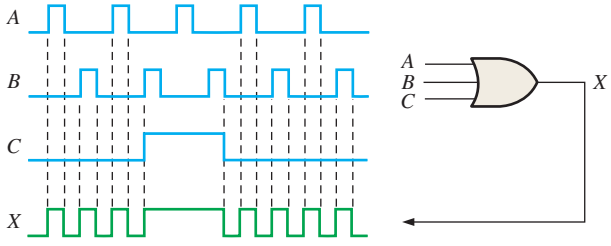


FIGURE 3–23

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.

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:

$$0 + 0 = 0$$
$$0 + 1 = 1$$
$$1 + 0 = 1$$
$$1 + 1 = 1$$

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.

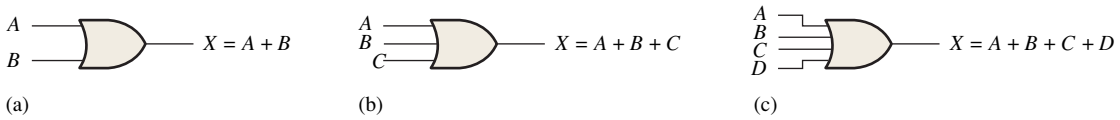


FIGURE 3–24 Boolean expressions for OR gates with two, three, and four inputs.

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.

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

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.

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 is shown in Figure 3–26.

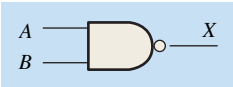


FIGURE 3–26 Standard NAND gate logic symbol

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.

TABLE 3–7

Truth table for a 2-input NAND gate.

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

1 = HIGH, 0 = 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.

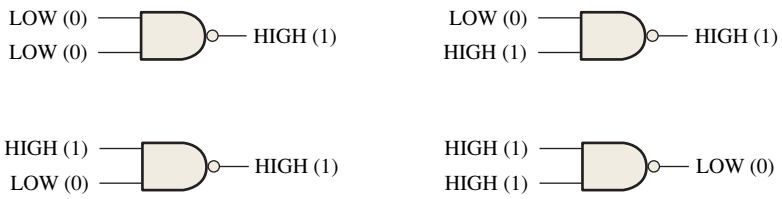


FIGURE 3–27 Operation of a 2-input NAND gate. Open file F03-27 to verify NAND gate operation.

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.

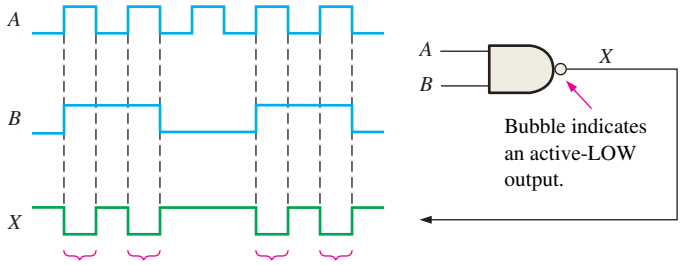


FIGURE 3–28

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.

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.

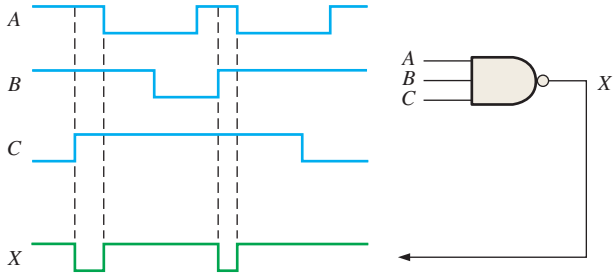


FIGURE 3–29

Solution

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

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.

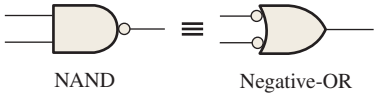


FIGURE 3–30 ANSI/IEEE standard symbols representing the two equivalent operations of a NAND gate.

EXAMPLE 3–14

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

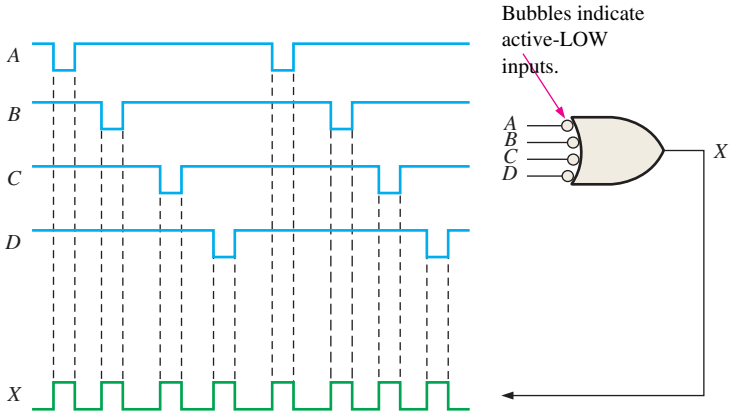


FIGURE 3–33

Solution

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

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.

TABLE 3–8

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

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 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 is shown in Figure 3–34.

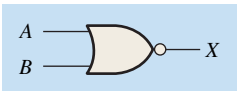


FIGURE 3–34 Standard NOR gate logic symbols

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.

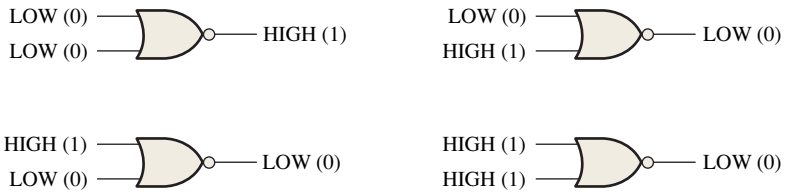


FIGURE 3–35 Operation of a 2-input NOR gate. Open file F03-35 to verify NOR gate operation.

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.

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.

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?

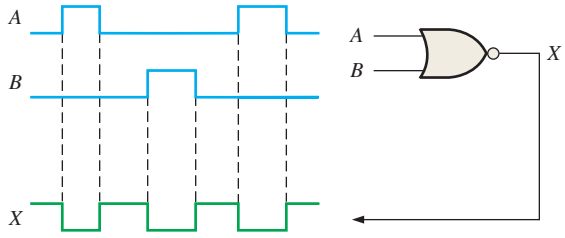


FIGURE 3–36

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.

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.

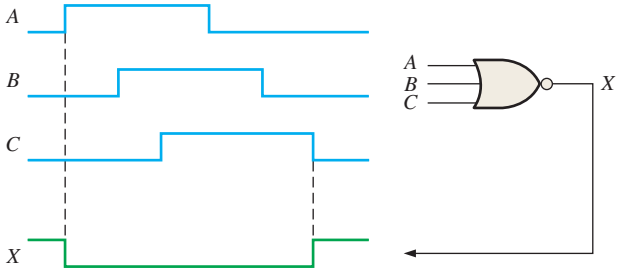


FIGURE 3–37

Solution

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

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.

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

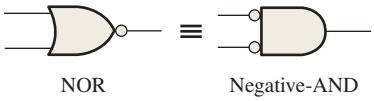


FIGURE 3–38 Standard symbols representing the two equivalent operations of a NOR gate.

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.

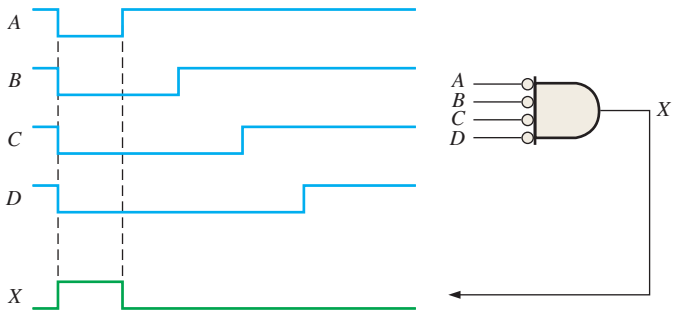


FIGURE 3–41

Solution

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

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

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

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.

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

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.

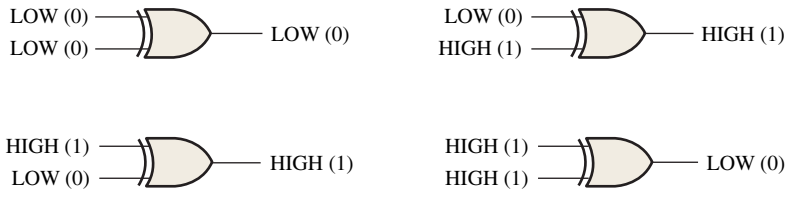


FIGURE 3–43 All possible logic levels for an exclusive-OR gate. Open file F03-43 to verify XOR gate operation.

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.

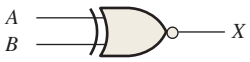


FIGURE 3–45 Standard logic symbol for the exclusive-NOR gate.

TABLE 3–12
Truth table for an exclusive-NOR gate.

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

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.

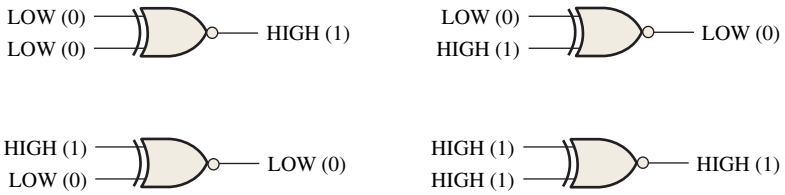


FIGURE 3–46 All possible logic levels for an exclusive-NOR gate. Open file F03-46 to verify XNOR gate operation.

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.

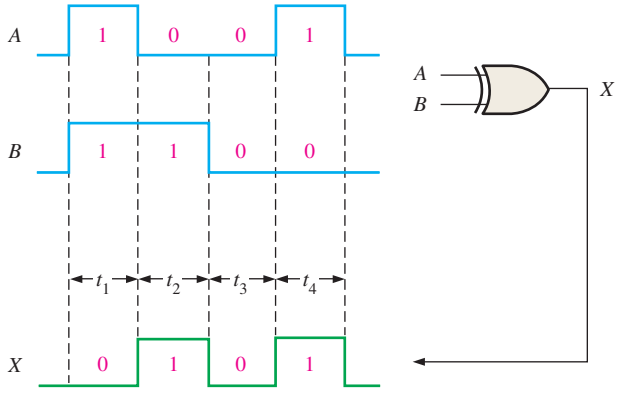


FIGURE 3–47 Example of exclusive-OR gate operation with pulse waveform inputs.

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.

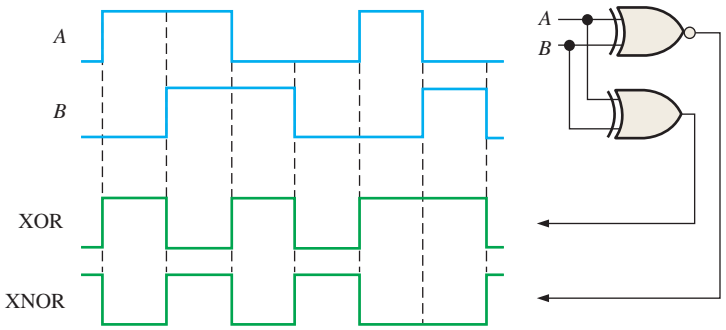


FIGURE 3–48

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.

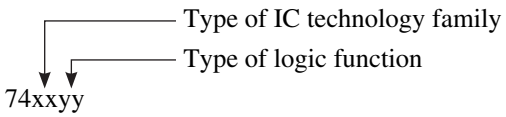
3–8 Fixed-Function Logic Gates

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.

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.

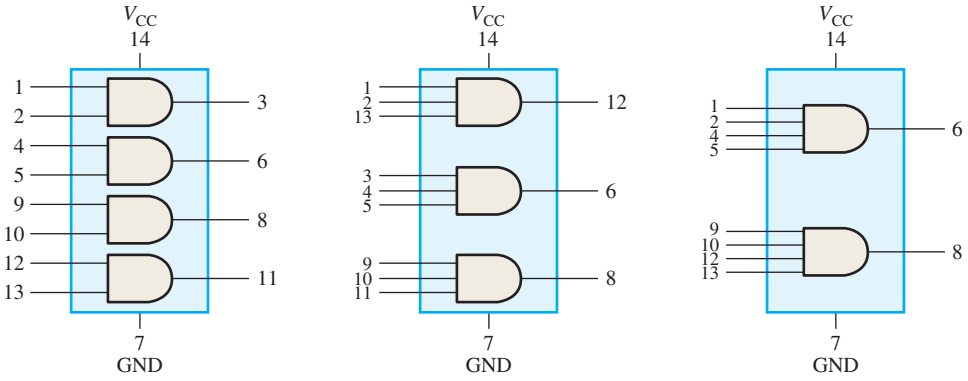
74 Series Logic Gate Functions

The 74 series is the standard fixed-function logic devices. The device label format includes one or more letters that indentify 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,



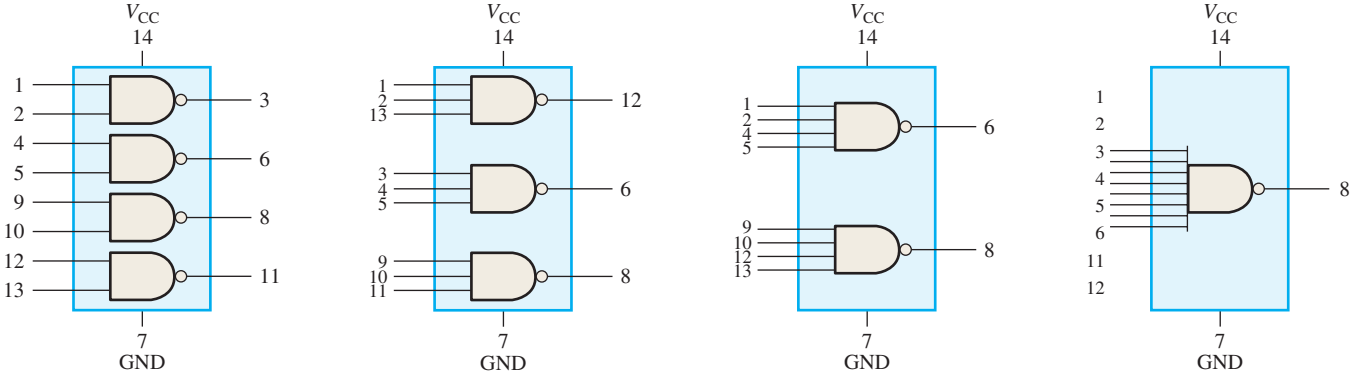
(a) 74xx08 (b) 74xx11 (c) 74xx21

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.



(a) 74xx00 (b) 74xx10 (c) 74xx20 (d) 74xx30

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.

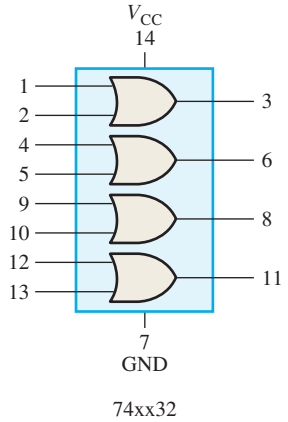


FIGURE 3–61 74 series OR gate device.

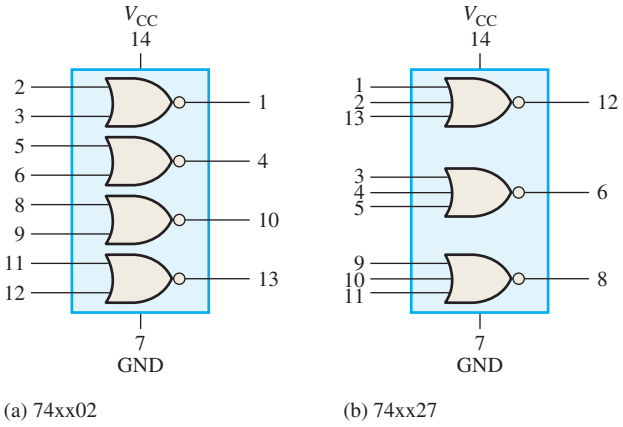


FIGURE 3–62 74 series NOR gate devices.

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.

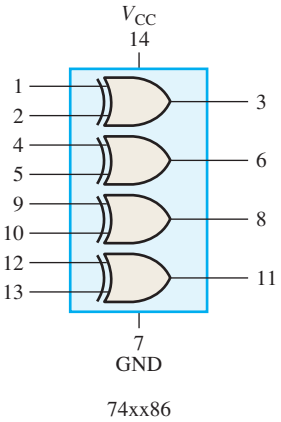


FIGURE 3–63 74 series XOR gate.

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.

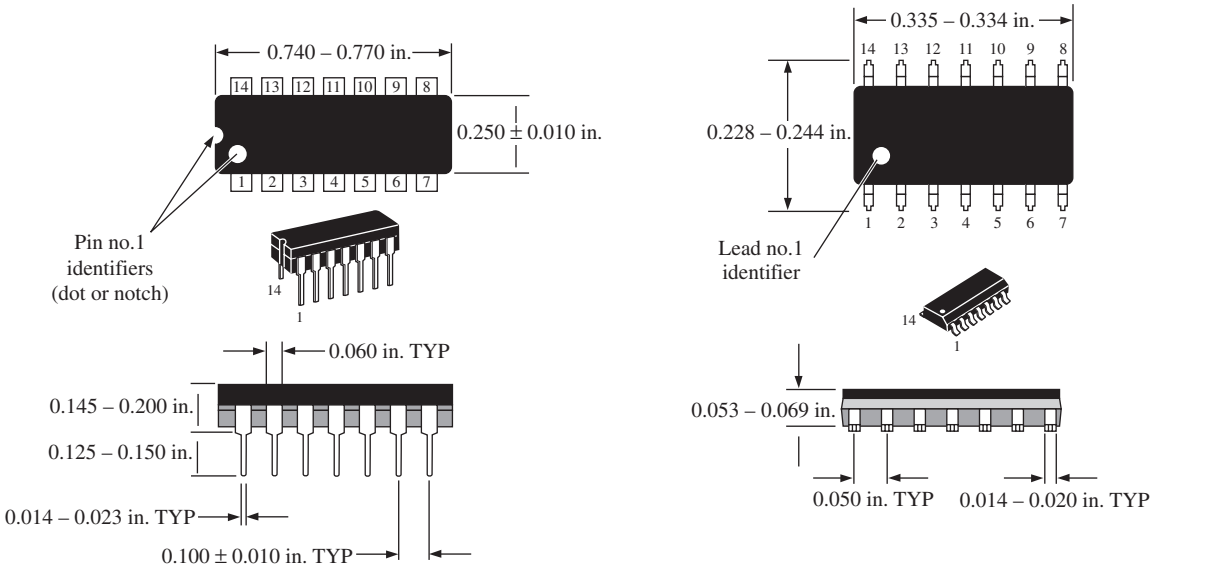


FIGURE 3–64 Typical dual in-line (DIP) and small-outline (SOIC) packages showing pin numbers and basic dimensions.