

5

COMBINATIONAL LOGIC

CHAPTER OBJECTIVES

- Analyze basic combinational logic circuits, such as AND-OR, AND-OR-Invert, exclusive-OR, exclusive-NOR, and other general combinational networks
- Use AND-OR and AND-OR-Invert circuits to implement sum-of-products (SOP) and product-of-sums (POS) expressions
- Write the Boolean output expression for any combinational logic circuit
- Develop a truth table from the output expression for a combinational logic circuit
- Use the Karnaugh map to expand an output expression containing terms with missing variables into a full SOP form
- Design a combinational logic circuit for a given Boolean output expression
- Design a combinational logic circuit for a given truth table
- Simplify a combinational logic circuit to its minimum form

- Use NAND gates to implement any combinational logic
- Use NOR gates to implement any combinational logic

INTRODUCTION

In Chapters 3 and 4, logic gates were discussed on an individual basis and in simple combinations. You were introduced to SOP and POS implementations, which are basic forms of combinational logic. When logic gates are connected together to produce a specified output for certain specified combinations of input variables, with no storage involved, the resulting circuit is in the category of **combinational logic**. In combinational logic, the output is at all times dependent on the combination of input levels. This chapter expands on the material introduced in earlier chapters with a coverage of the analysis, design, and troubleshooting of various combinational logic circuits.

5-1

BASIC COMBINATIONAL LOGIC CIRCUITS

In Chapter 4, you learned that SOP expressions are implemented with an AND gate for each product term and one OR gate for summing all of the product terms. This SOP implementation is called AND-OR logic and is the basic form for realizing standard Boolean functions. In this section, the AND-OR and the AND-OR-Invert are examined; and the exclusive-OR and exclusive-NOR gates, which are actually a form of AND-OR logic, are also covered.

After completing this section, you should be able to

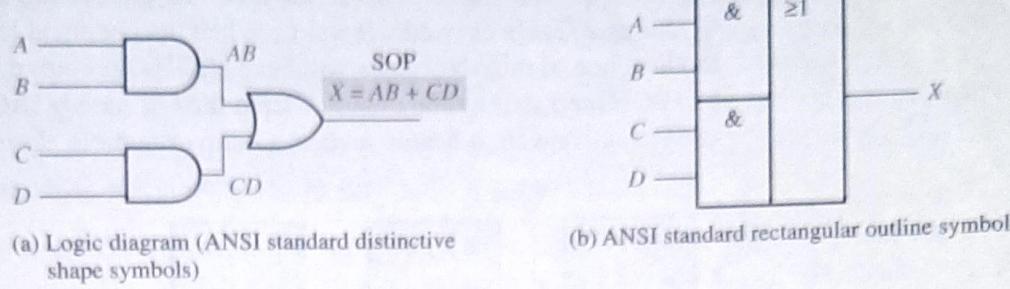
- Analyze and apply AND-OR circuits
- Analyze and apply exclusive-OR gates
- Analyze and apply AND-OR-Invert circuits
- Analyze and apply exclusive-NOR gates

AND-OR Logic

Figure 5-1(a) shows an AND-OR circuit consisting of two 2-input AND gates and one 2-input OR gate; Figure 5-1(b) is the ANSI standard rectangular outline symbol. The Boolean expressions for the AND gate outputs and the resulting SOP expression for the output X are shown on the diagram. In general, an AND-OR circuit can have any number of AND gates each with any number of inputs.

The truth table for a 4-input AND-OR logic circuit is shown in Table 5-1. The intermediate AND gate outputs (the AB and CD columns) are also shown in the table.

► FIGURE 5-1



► TABLE 5-1

INPUTS				AB	CD	OUTPUT X
A	B	C	D			
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	1	0	0	0	0
0	0	1	1	0	1	1
0	1	0	0	0	0	0
0	1	0	1	0	0	0
0	1	1	0	0	0	0
0	1	1	1	0	1	1
1	0	0	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	0	0	0
1	0	1	1	0	1	1
1	1	0	0	1	0	1
1	1	0	1	1	0	1
1	1	1	0	1	0	1
1	1	1	1	1	1	1

An AND-OR circuit directly implements an SOP expression, assuming the complements (if any) of the variables are available. The operation of the AND-OR circuit in Figure 5-1 is stated as follows:

For a 4-input AND-OR logic circuit, the output X is HIGH (1) if both input A and input B are HIGH (1) or both input C and input D are HIGH (1).

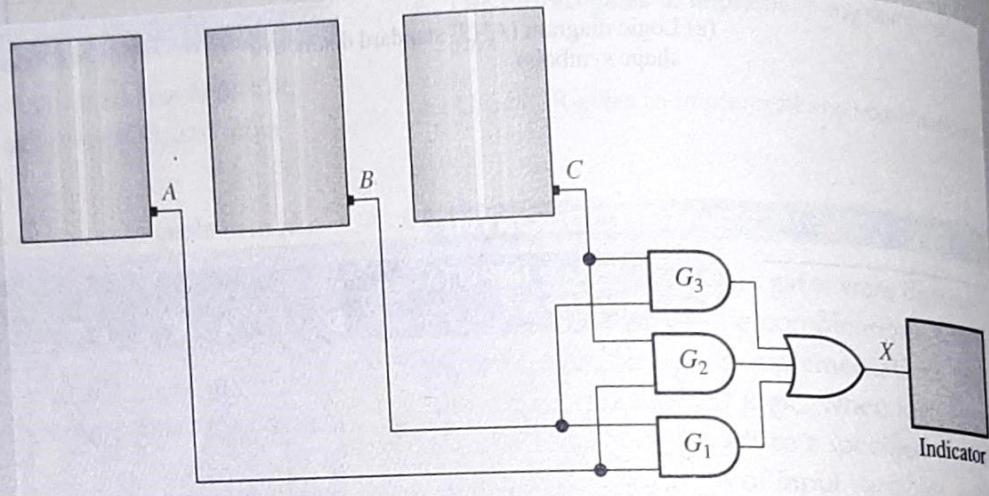
EXAMPLE 5-1

In a certain chemical-processing plant, a liquid chemical is used in a manufacturing process. The chemical is stored in three different tanks. A level sensor in each tank produces a HIGH voltage when the level of chemical in the tank drops below a specified point.

Design a circuit that monitors the chemical level in each tank and indicates when the level in any two of the tanks drops below the specified point.

Solution

The AND-OR circuit in Figure 5-2 has inputs from the sensors on tanks A, B, and C shown. The AND gate G_1 checks the levels in tanks A and B, gate G_2 checks tanks A and C, and gate G_3 checks tanks B and C. When the chemical level in any two of the tanks is too low, one of the AND gates will have HIGHs on both of its inputs causing its output to be HIGH, and so the final output X from the OR gate is HIGH. This HIGH input is then used to activate an indicator such as a lamp or audible alarm, as shown in the figure.

**FIGURE 5-2**

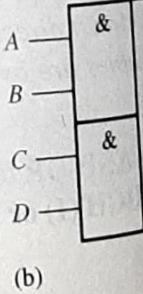
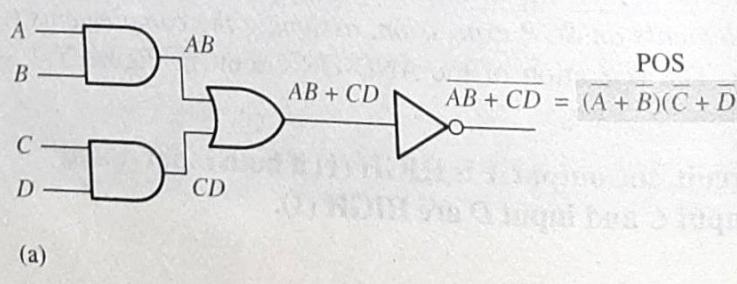
Supplementary Problem Write the Boolean SOP expression for the AND-OR logic in Figure 5-2.

AND-OR-Invert Logic

When the output of an AND-OR circuit is complemented (inverted), it results in an AND-OR-Invert circuit. Recall that AND-OR logic directly implements SOP expressions. POS expressions can be implemented with AND-OR-Invert logic. This is illustrated as follows, starting with a POS expression and developing the corresponding AND-OR-Invert expression.

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

The logic diagram in Figure 5-3(a) shows an AND-OR-Invert circuit and the development of the POS output expression. The ANSI standard rectangular outline symbol is shown in

FIGURE 5-3

part (b). In general, an AND-OR-Invert circuit can have any number of AND gates each with any number of inputs.

The operation of the AND-OR-Invert circuit in Figure 5–3 is stated as follows:

For a 4-input AND-OR-Invert logic circuit, the output X is LOW (0) if both input A and input B are HIGH (1) or both input C and input D are HIGH (1).

A truth table can be developed from the AND-OR truth table in Table 5–1 by simply changing all 1s to 0s and all 0s to 1s in the output column.

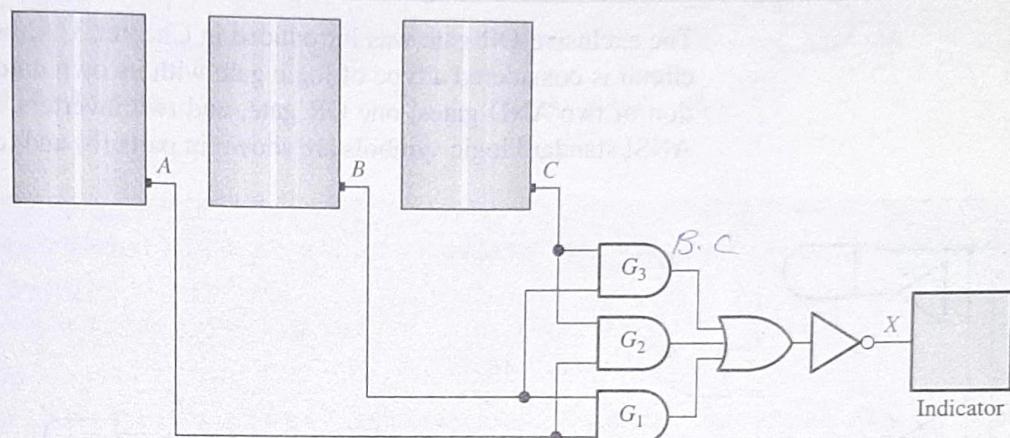
EXAMPLE 5–2

The sensors in the chemical tanks of Example 5–1 are being replaced by a new model that produces a LOW voltage instead of a HIGH voltage when the level of the chemical in the tank drops below a critical point.

Modify the circuit in Figure 5–2 to operate with the different input levels and still produce a HIGH output to activate the indicator when the level in any two of the tanks drops below the critical point. Show the logic diagram.

Solution

The AND-OR-Invert circuit in Figure 5–4 has inputs from the sensors on tanks A , B , and C as shown. The AND gate G_1 checks the levels in tanks A and B , gate G_2 checks tanks A and C , and gate G_3 checks tanks B and C . When the chemical level in any two of the tanks gets too low, each AND gate will have a LOW on at least one input causing its output to be LOW and, thus, the final output X from the inverter is HIGH. This HIGH output is then used to activate an indicator.



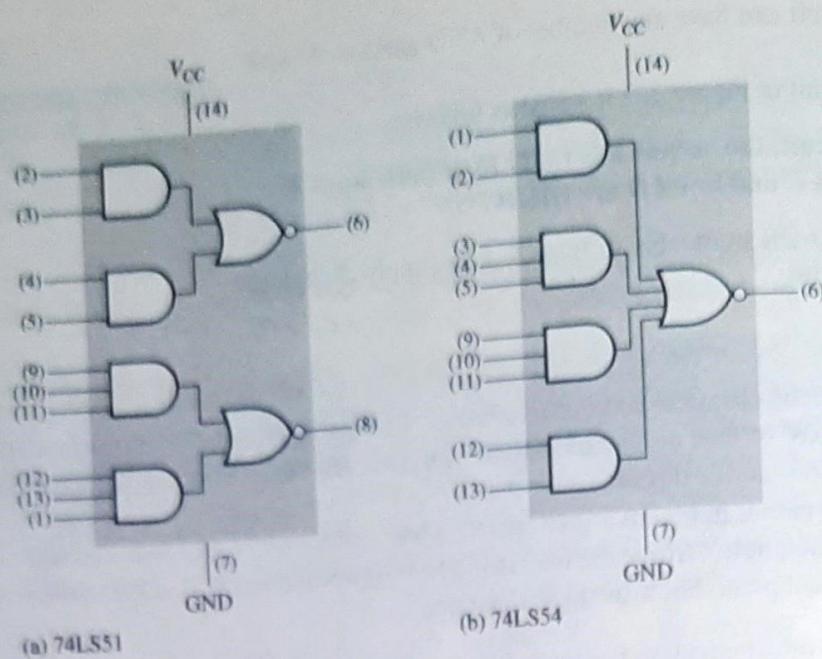
▲ FIGURE 5–4

Supplementary Problem

Write the Boolean expression for the AND-OR-Invert logic in Figure 5–4 and show that the output is HIGH (1) when any two of the inputs A , B , and C are LOW (0).

AND-OR-INVERT INTEGRATED CIRCUITS

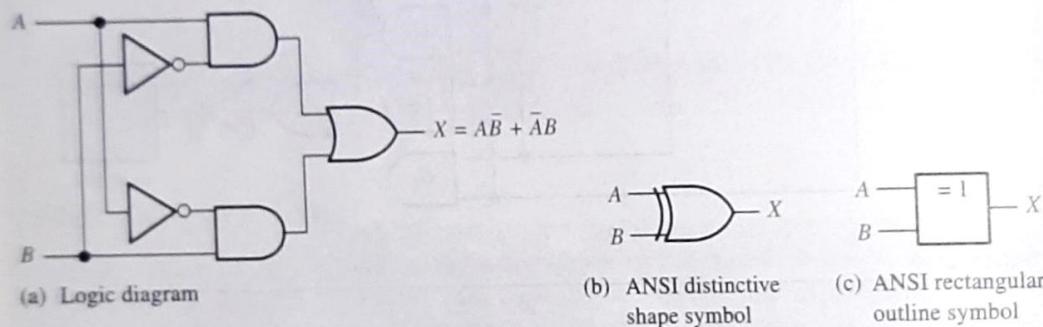
The 74LS51 and the 74LS54 are examples of AND-OR-Invert logic. The 74LS51 includes two separate AND-OR-Invert circuits in a single package. One circuit has two 2-input AND gates and the other circuit has two 3-input AND gates, as shown in Figure 5–5(a). The 74LS54 is a single AND-OR-Invert circuit that has two 2-input AND gates and two 3-input AND gates, as shown in Figure 5–5(b). Notice that the inversion is indicated by a bubble on the output of the OR gates in each case, showing that the OR-Invert part of the circuit is effectively a NOR gate.



▲ FIGURE 5-5

Exclusive-OR Logic

The exclusive-OR gate was introduced in Chapter 3. Although, because of its importance, it is considered a type of logic gate with its own unique symbol, it is actually a combination of two AND gates, one OR gate, and two inverters, as shown in Figure 5-6(a). ANSI standard logic symbols are shown in parts (b) and (c).



▲ FIGURE 5-6

The output expression for the circuit in Figure 5-6 is

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

Evaluation of this expression results in the truth table in Table 5-2. Notice that the output is HIGH only when the two inputs are at opposite levels. A special exclusive-OR operation is often used, so the expression $X = A\bar{B} + \bar{A}B$ can be stated as "X is equal to A exclusive OR B" and can be written as

$$X = A \oplus B$$

► TABLE 5-2
Truth table for exclusive-OR

A	B	X
0	0	0
0	1	1
1	0	1
1	1	0

Exclusive-NOR Logic

As you know, the complement of the exclusive-OR function is the exclusive-NOR, which is derived as follows:

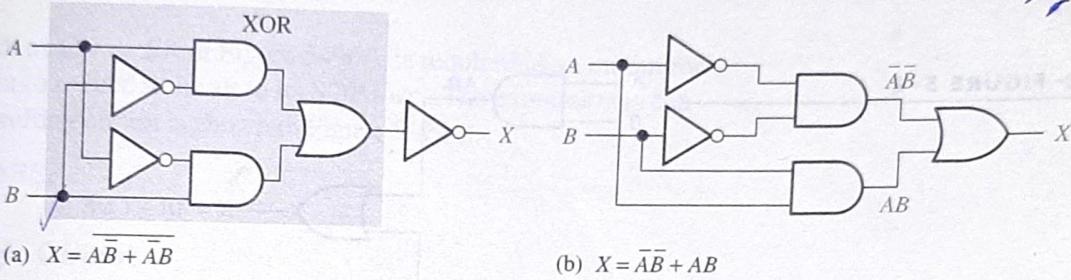
$$X = \overline{AB} + \overline{AB} = \overline{(AB)} \overline{(AB)} = (\overline{A} + B)(A + \overline{B}) = \overline{AB} + AB.$$

Notice that the output X is HIGH only when the two inputs, A and B , are at the same level.

The exclusive-NOR can be implemented by simply inverting the output of an exclusive-OR, as shown in Figure 5-7(a), or by directly implementing the expression $\overline{AB} + AB$, as shown in part (b).

$$\begin{aligned} &(\overline{A} + B) \overline{(\overline{A} + B)} \\ &(A + \overline{B}) \overline{(A + \overline{B})} \end{aligned}$$

► FIGURE 5-7



SECTION 5-1 REVIEW

Answers are at the end of the chapter.

- Determine the output (1 or 0) of a 4-variable AND-OR-Invert circuit for each of the following input conditions:
 - $A = 1, B = 0, C = 1, D = 0$
 - $A = 1, B = 1, C = 0, D = 1$
 - $A = 0, B = 1, C = 1, D = 1$.
- Determine the output (1 or 0) of an exclusive-OR gate for each of the following input conditions:
 - $A = 1, B = 0$
 - $A = 1, B = 1$
 - $A = 0, B = 1$
 - $A = 0, B = 0$.
- Develop the truth table for a certain 3-input logic circuit with the output expression $X = \overline{ABC} + \overline{ABC} + \overline{ABC} + \overline{ABC} + ABC$.
- Draw the logic diagram for an exclusive-NOR circuit.

5-2 IMPLEMENTING COMBINATIONAL LOGIC

In this section, examples are used to illustrate how to implement a logic circuit from a Boolean expression or a truth table. Minimization of a logic circuit using the methods covered in Chapter 4 is also discussed.

After completing this section, you should be able to

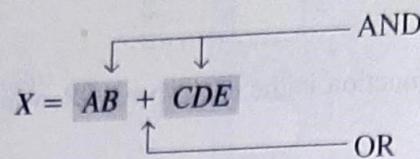
- Implement a logic circuit from a Boolean expression
- Implement a logic circuit from a truth table
- Minimize a logic circuit

From a Boolean Expression to a Logic Circuit

Let us examine the following Boolean expression:

$$X = AB + CDE$$

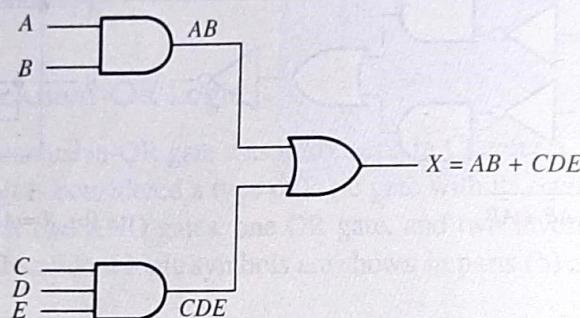
A brief inspection shows that this expression is composed of two terms, AB and CDE , domain of five variables. The first term is formed by ANDing A with B , and the second term is formed by ANDing C , D , and E . The two terms are then ORed to form the output X . The operations are indicated in the structure of the expression as follows:



Note that in this particular expression, the AND operations forming the two individual terms AB and CDE , must be performed *before* the terms can be ORed.

To implement this Boolean expression, a 2-input AND gate is required to form the term AB , and a 3-input AND gate is needed to form the term CDE . A 2-input OR gate is required to combine the two AND terms. The resulting logic circuit is shown in Figure 5-8.

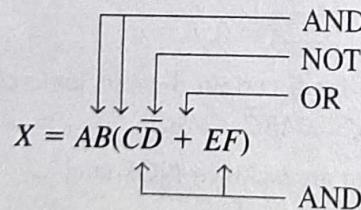
► FIGURE 5-8



As another example, let us implement the following expression:

$$X = AB(\bar{CD} + EF)$$

A breakdown of this expression shows that the terms AB , and $\bar{CD} + EF$ are ANDed. The term $\bar{CD} + EF$ is formed by first ANDing C and D and NOTing the result, and then ORing this result with E and F , and then ORing this result with AB . This structure is indicated in relation to the expression as follows:

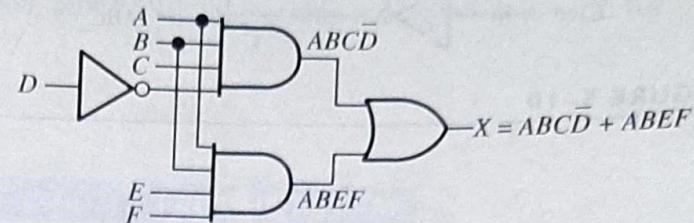
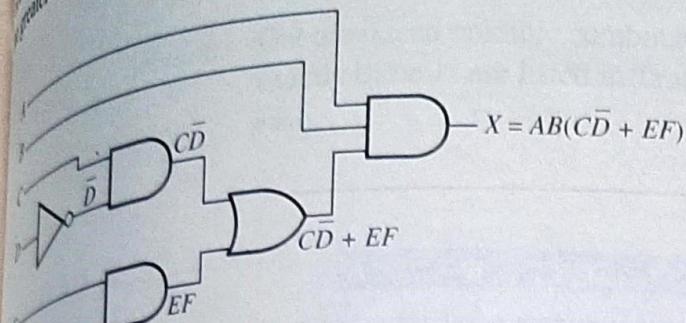


Before the expression can be formed, you must have the term $\bar{CD} + EF$; but before you can get this term, you must have the terms \bar{CD} and EF ; but before you can get the term \bar{CD} , you must have D . So, as you can see, the logic operations must be done in the proper order.

The logic gates required to implement $X = AB(\bar{CD} + EF)$ are as follows:

1. One inverter to form \bar{D}
2. Two 2-input AND gates to form \bar{CD} and EF
3. One 2-input OR gate to form $\bar{CD} + EF$
4. One 3-input AND gate to form X

Often, the total propagation delay time through a logic circuit is a major consideration. Propagation delays are additive, so the more gates or inverters between input and output, the greater the propagation delay time.



(b) Sum-of-products implementation of the circuit in part (a)

FIGURE 5-9

Unless an intermediate term, such as $CD̄ + EF$ in Figure 5-9(a), is required as an output for some other purpose, it is usually best to reduce a circuit to its SOP form. The expression is converted to SOP as follows, and the resulting circuit is shown in Figure 5-9(b).

$$AB(CD̄ + EF) = ABCD̄ + ABEF$$

From a Truth Table to a Logic Circuit

If you begin with a truth table instead of an expression, you can write the SOP expression from the truth table and then implement the logic circuit. Table 5-3 specifies a logic function.

TABLE 5-3

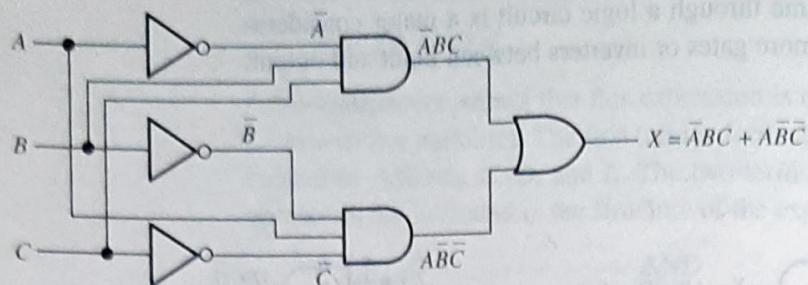
INPUTS			OUTPUT	PRODUCT TERM
A	B	C	X	
0	0	0	0	
0	0	1	0	
0	1	0	0	
0	1	1	1	$\bar{A}BC$
1	0	0	1	$A\bar{B}\bar{C}$
1	0	1	0	
1	1	0	0	
1	1	1	0	

The Boolean SOP expression obtained from the truth table by ORing the product terms for $X = 1$ is

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

The first term in the expression is formed by ANDing the three variables \bar{A} , B , and C . The second term is formed by ANDing the three variables A , \bar{B} , and \bar{C} . The logic gates required to implement this expression are as follows: three inverters to produce \bar{A} , \bar{B} , and \bar{C} variables; two 3-input AND gates to form the terms $\bar{A}BC$ and $A\bar{B}\bar{C}$; and one OR gate to form the final output function, $\bar{A}BC + A\bar{B}\bar{C}$.

The implementation of this logic function is illustrated in Figure 5-10.



▲ FIGURE 5-10

EXAMPLE 5-3

Design a logic circuit to implement the operation specified in the truth table of Table 5-4.

▼ TABLE 5-4

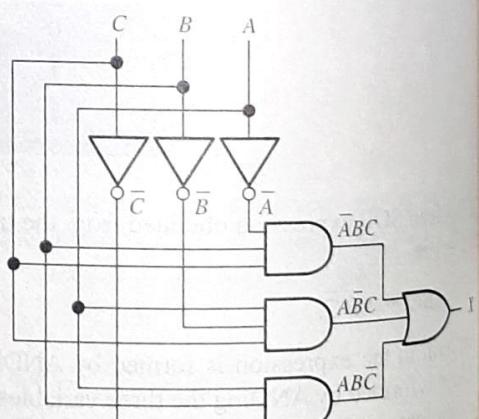
INPUTS			OUTPUT	PRODUCT TERM
A	B	C	X	
0	0	0	0	
0	0	1	0	
0	1	0	0	
0	1	1	1	\bar{ABC}
1	0	0	0	
1	0	1	1	\bar{ABC}
1	1	0	1	\bar{ABC}
1	1	1	0	

Solution Notice that $X = 1$ for only three of the input conditions. Therefore, the logic expression is

$$X = \bar{ABC} + \bar{ABC} + \bar{ABC}.$$

The logic gates required are three inverters, three 3-input AND gates and one 3-input OR gate. The logic circuit is shown in Figure 5-11.

► FIGURE 5-11



Supplementary Problem Determine if the logic circuit of Figure 5-11 can be simplified.

EXAMPLE 5-4

Develop a logic circuit with four input variables that will only produce a 1 output when exactly three input variables are 1s.

Solution

Out of sixteen possible combinations of four variables, the combinations in which there are exactly three 1s are listed in Table 5-5, along with the corresponding product term for each.

TABLE 5-5

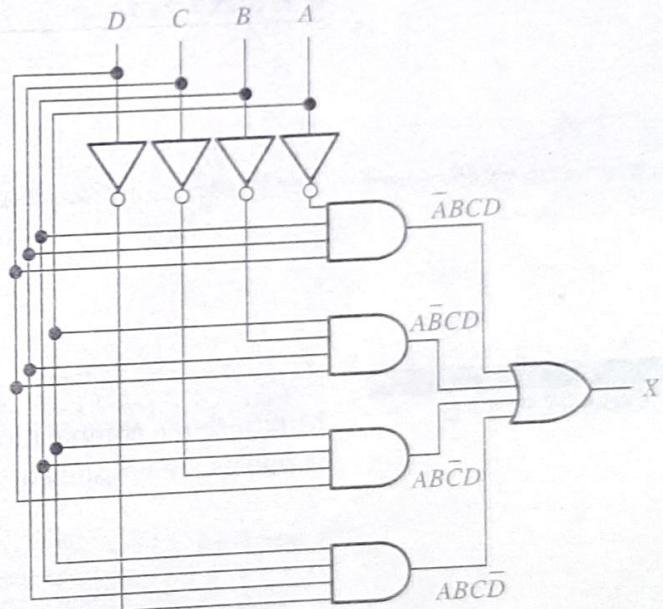
A	B	C	D	PRODUCT TERM
0	1	1	1	$\bar{A}BCD$
1	0	1	1	$A\bar{B}CD$
1	1	0	1	$A\bar{B}\bar{C}D$
1	1	1	0	$AB\bar{C}\bar{D}$

The product terms are ORed to get the following expression:

$$X = \bar{A}BCD + A\bar{B}CD + A\bar{B}\bar{C}D + AB\bar{C}\bar{D}$$

This expression is implemented in Figure 5-12 with AND-OR logic.

► FIGURE 5-12

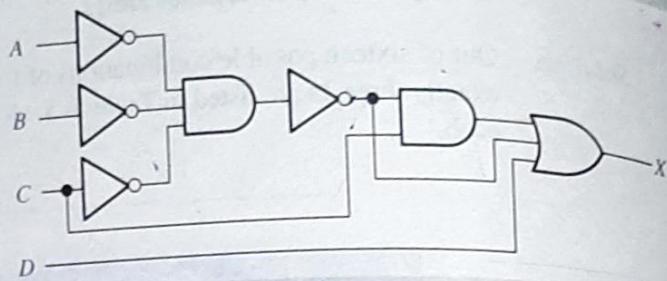


Supplementary Problem Determine if the logic circuit of Figure 5-12 can be simplified.

EXAMPLE 5-5

Reduce the combinational logic circuit in Figure 5-13 to a minimum form.

► FIGURE 5-13



Solution The expression for the output of the circuit is

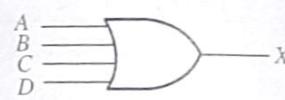
$$X = (\overline{ABC})C + \overline{ABC} + D$$

Applying DeMorgan's theorem and Boolean algebra,

$$\begin{aligned} X &= (\overline{\overline{A}} + \overline{\overline{B}} + \overline{\overline{C}})C + \overline{\overline{A}} + \overline{\overline{B}} + \overline{\overline{C}} + D \\ &= AC + BC + CC + A + B + C + D \\ &= AC + BC + C + A + B + C + D \\ &= C(A + B + 1) + A + B + D \\ X &= A + B + C + D \end{aligned}$$

The simplified circuit is a 4-input OR gate as shown in Figure 5-14.

► FIGURE 5-14

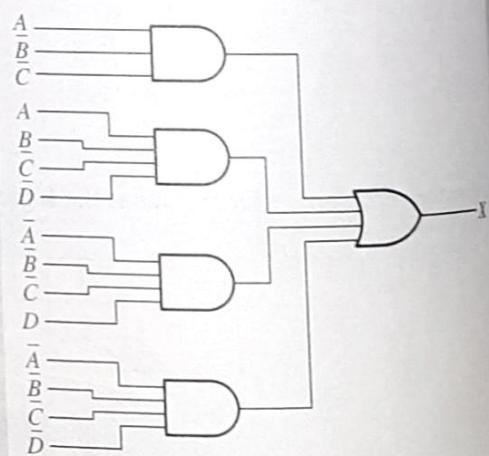


Supplementary Problem Verify the minimized expression $A + B + C + D$ using a Karnaugh map.

EXAMPLE 5-6

Minimize the combinational logic circuit in Figure 5-15. Inverters for the complementary variables are not shown.

► FIGURE 5-15



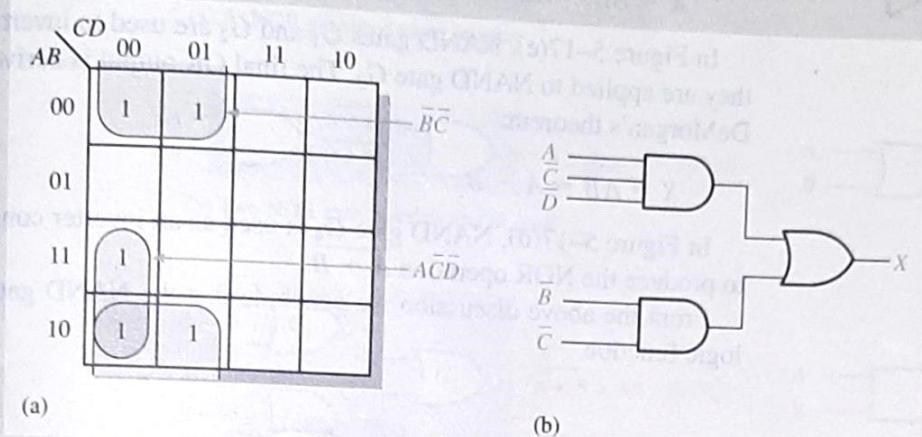
Solution The output expression is

$$X = A\bar{B}\bar{C} + AB\bar{C} + \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}\bar{C}$$

Expanding the first term to include the missing variables D and \bar{D} ,

$$\begin{aligned} X &= A\bar{B}\bar{C}(D + \bar{D}) + AB\bar{C} + \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}\bar{C} \\ &= A\bar{B}\bar{C}D + A\bar{B}\bar{C}\bar{D} + AB\bar{C} + \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}\bar{C} \end{aligned}$$

This expanded SOP expression is mapped and simplified on the Karnaugh map in Figure 5-16(a). The simplified implementation is shown in part (b). Inverters are not shown.



▲ FIGURE 5-16

Supplementary Problem

Develop the POS equivalent of the circuit in Figure 5-16(b).

SECTION 5-2 REVIEW

- Implement the following Boolean expressions as they are stated:
 - $X = ABC + AB + AC$
 - $X = AB(C + DE)$
- Develop a logic circuit that will produce a 1 on its output only when all three inputs are 1s or when all three inputs are 0s.
- Reduce the circuits in Question 1 to minimum SOP form.

5-3 THE UNIVERSAL PROPERTY OF NAND AND NOR GATES

Up to this point, you have studied combinational circuits implemented with AND gates, OR gates, and inverters. In this section, the universal property of the NAND gate and the NOR gate is discussed. The universality of the NAND gate means that it can be used as an inverter and that combinations of NAND gates can be used to implement the AND, OR, and NOR operations. Similarly, the NOR gate can be used to implement the inverter, AND, OR, and NAND operations.

After completing this section, you should be able to

- Use NAND gates to implement the inverter, the AND gate, the OR gate, and the NOR gate
- Use NOR gates to implement the inverter, the AND gate, the OR gate, and the NAND gate

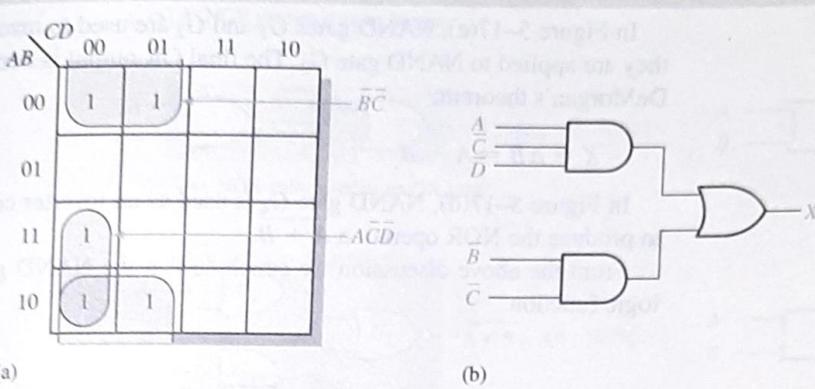
Solution The output expression is

$$X = \bar{A}\bar{B}\bar{C} + A\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}\bar{C}D + \bar{A}\bar{B}CD$$

Expanding the first term to include the missing variables D and \bar{D} ,

$$\begin{aligned} X &= \bar{A}\bar{B}\bar{C}(D + \bar{D}) + A\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}\bar{C}D + \bar{A}\bar{B}CD \\ &= \bar{A}\bar{B}\bar{C}D + \bar{A}\bar{B}\bar{C}\bar{D} + A\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}CD + \bar{A}\bar{B}CD \end{aligned}$$

This expanded SOP expression is mapped and simplified on the Karnaugh map in Figure 5-16(a). The simplified implementation is shown in part (b). Inverters are not shown.



▲ FIGURE 5-16

Supplementary Problem Develop the POS equivalent of the circuit in Figure 5-16(b).

SECTION 5-2 REVIEW

1. Implement the following Boolean expressions as they are stated:
 - (a) $X = ABC + AB + AC$
 - (b) $X = AB(C + DE)$
2. Develop a logic circuit that will produce a 1 on its output only when all three inputs are 1s or when all three inputs are 0s.
3. Reduce the circuits in Question 1 to minimum SOP form.

5-3 THE UNIVERSAL PROPERTY OF NAND AND NOR GATES

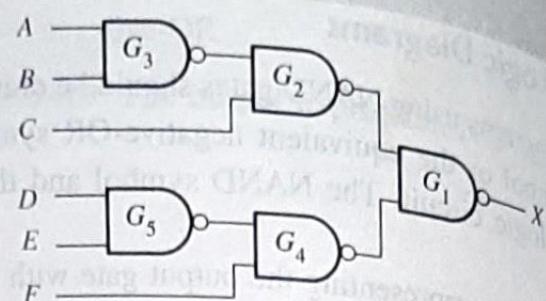
Up to this point, you have studied combinational circuits implemented with AND gates, OR gates, and inverters. In this section, the universal property of the NAND gate and the NOR gate is discussed. The universality of the NAND gate means that it can be used as an inverter and that combinations of NAND gates can be used to implement the AND, OR, and NOR operations. Similarly, the NOR gate can be used to implement the inverter, AND, OR, and NAND operations.

After completing this section, you should be able to

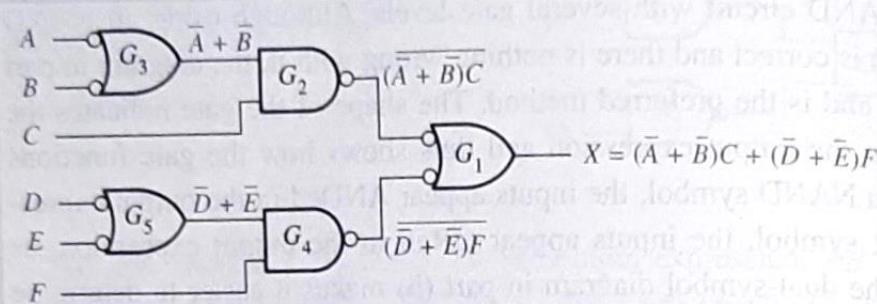
Use NAND gates to implement the inverter, the AND gate, the OR gate, and the NOR gate
Use NOR gates to implement the inverter, the AND gate, the OR gate, and the NAND gate

EXAMPLE 5-7

Redraw the logic diagram and develop the output expression for the circuit in Figure 5-22, using the appropriate dual symbols.

FIGURE 5-22**Solution**

Redraw the logic diagram in Figure 5-22 with the use of equivalent negative-OR symbols as shown in Figure 5-23. Writing the expression for X directly from the indicated logic operation of each gate gives $X = (\bar{A} + \bar{B})C + (\bar{D} + \bar{E})F$.

FIGURE 5-23**Supplementary Problem**

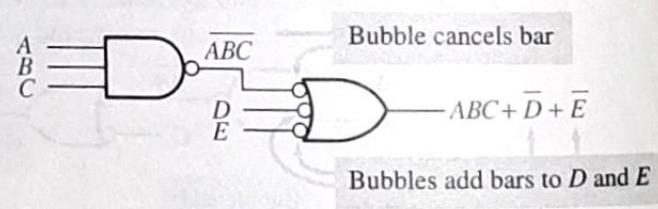
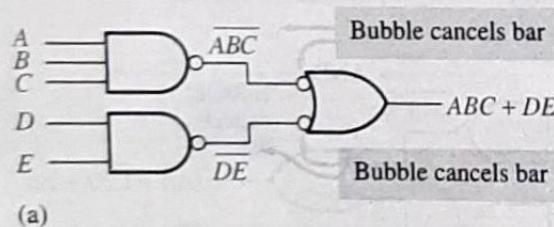
Derive the output expression from Figure 5-22 and show it is equivalent to the expression in the solution.

EXAMPLE 5-8

Implement each expression with NAND logic:

$$(a) ABC + DE \quad (b) ABC + \bar{D} + \bar{E}$$

Solution See Figure 5-24.

**FIGURE 5-24****Supplementary Problem**

Convert the NAND circuits in Figure 5-24(a) and (b) to equivalent AND-OR logic.

NOR Logic
A NOR gate can function as either a NOR or a negative-AND, as shown by DeMorgan's theorem.

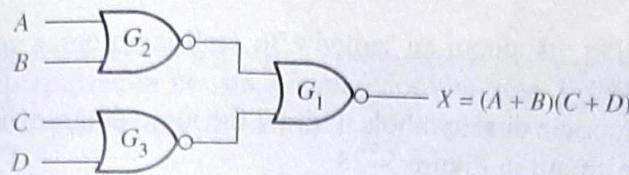
$$\overline{A+B} = \overline{\overline{AB}}$$

NOR negative-AND

Consider the NOR logic in Figure 5-25. The output expression is developed as follows:

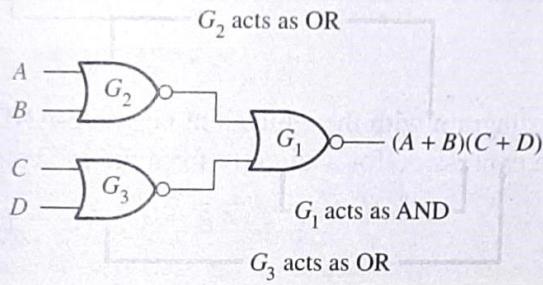
$$X = \overline{\overline{A+B} + \overline{C+D}} = (\overline{\overline{A+B}})(\overline{\overline{C+D}}) = (A+B)(C+D)$$

FIGURE 5-25

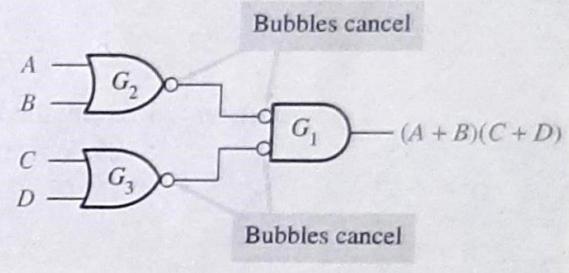


Notice that the output expression $(A+B)(C+D)$ consists of two OR terms ANDed together. This shows that gates G_2 and G_3 act as OR gates and gate G_1 acts as an AND gate, as illustrated in Figure 5-26(a). This circuit is redrawn in part (b) with a negative-AND symbol for gate G_1 .

FIGURE 5-26



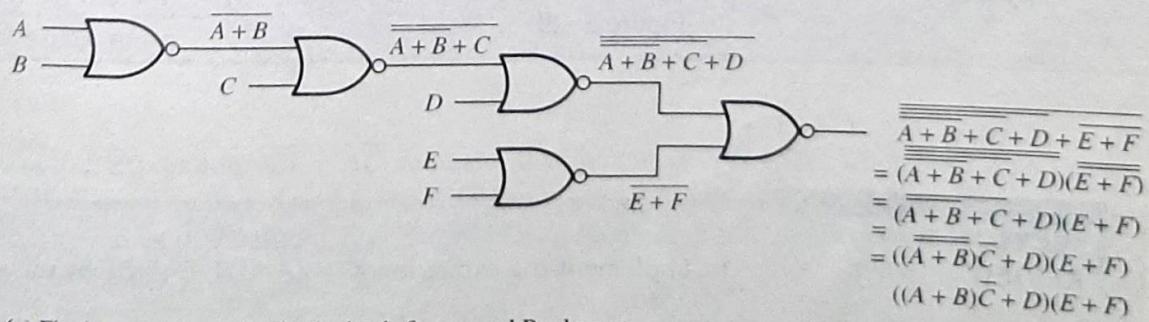
(a)



(b)

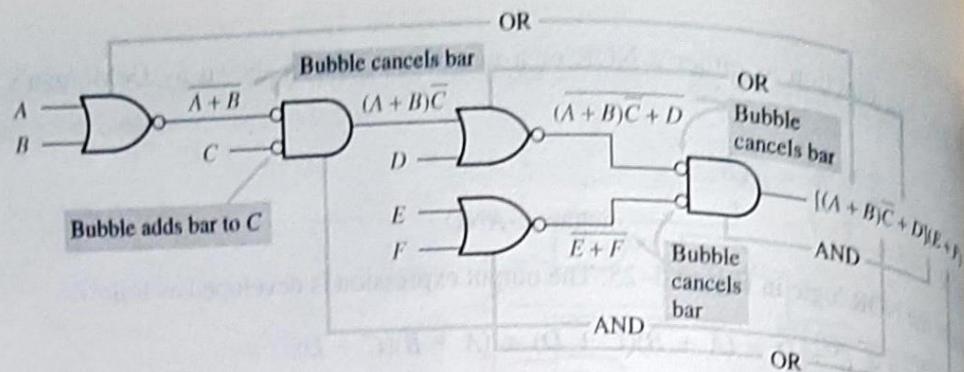
As with NAND logic, the purpose for using the dual symbols is to make the logic diagram easier to read and analyze, as illustrated in the NOR logic circuit in Figure 5-27. When the circuit in part (a) is redrawn with dual symbols in part (b), notice that all input-to-input connections between gates are bubble-to-bubble or nonbubble-to-nonbubble. Again, you can see that the shape of each gate symbol indicates the type of term (AND or OR) that it produces in the output expression, thus making the output expression easier to determine and the logic diagram easier to analyze.

FIGURE 5-27(a)



(a) Final output expression is obtained after several Boolean steps.

► FIGURE 5-27(b)

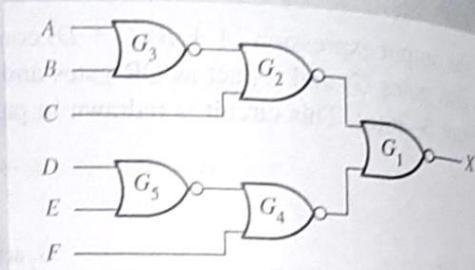


(b) Output expression can be obtained directly from the function of each gate symbol in the diagram.

EXAMPLE 5-9

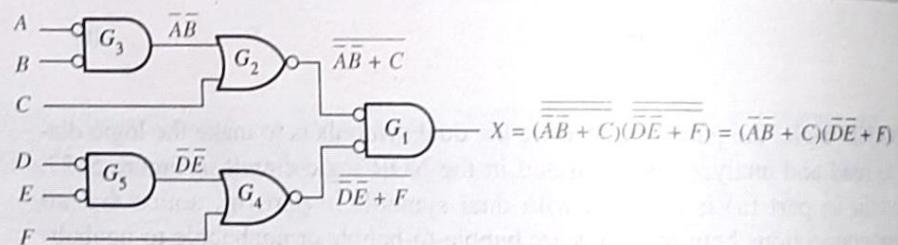
Using appropriate dual symbols, redraw the logic diagram and develop the output expression for the circuit in Figure 5-28.

► FIGURE 5-28



Solution Redraw the logic diagram with the equivalent negative-AND symbols as shown in Figure 5-29. Writing the expression for X directly from the indicated operation of each gate,

$$X = (\overline{AB} + C)(\overline{DE} + F)$$



▲ FIGURE 5-29

Supplementary Problem Prove that the output of the NOR circuit in Figure 5-28 is the same as for the circuit in Figure 5-29.

SECTION 5-4 REVIEW

1. Implement the expression $X = \overline{(\overline{A} + \overline{B} + \overline{C})DE}$ by using NAND logic.
2. Implement the expression $X = \overline{\overline{ABC} + (D + E)}$ with NOR logic.

LOGIC CIRCUIT OPERATION WITH PULSE WAVEFORMS

After completing this section, you should be able to

- Analyze combinational logic circuits with pulse waveform inputs
- Develop a timing diagram for any given combinational logic circuit with specified inputs

The operation of any gate is the same regardless of whether its inputs are pulsed or constant levels. The nature of the inputs (pulsed or constant levels) does not alter the truth table of the gate. The examples in this section illustrate the analysis of combinational logic circuits with pulsed inputs.

The following is a review of the operation of individual gates for use in analyzing combinational circuits with pulse waveform inputs:

1. The output of an AND gate is HIGH only when all inputs are HIGH at the same time.
2. The output of an OR gate is HIGH only when at least one of its inputs is HIGH.
3. The output of a NAND gate is LOW only when all inputs are HIGH at the same time.
4. The output of a NOR gate is LOW only when at least one of its inputs is HIGH.

EXAMPLE 5-10

Determine the final output waveform X for the circuit in Figure 5-30, with input waveforms A , B , and C as shown.

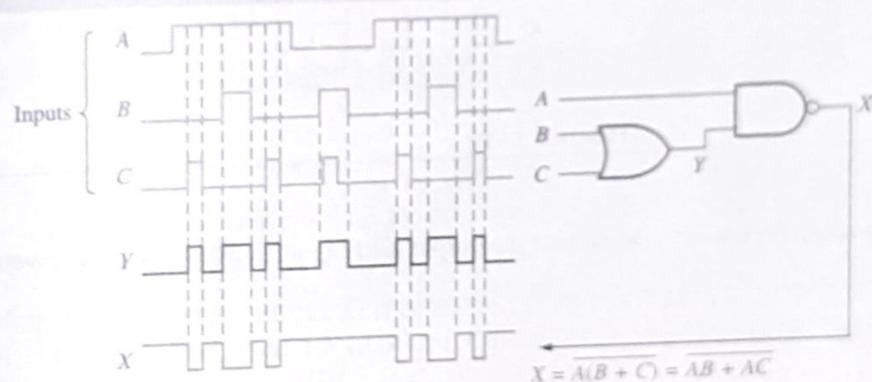


FIGURE 5-30

Solution

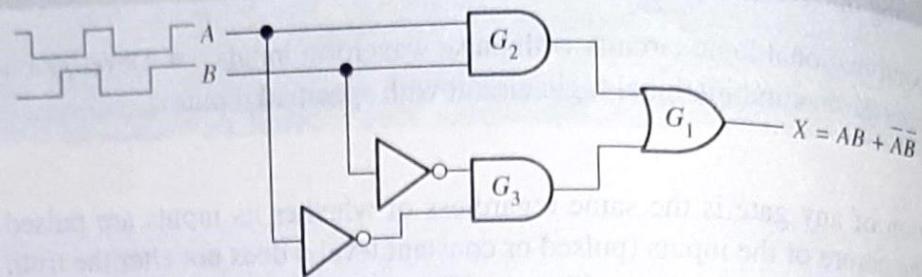
The output expression, $\overline{AB} + \overline{AC}$, indicates that the output X is LOW when both A and B are HIGH or when both A and C are HIGH or when all inputs are HIGH. The output waveform X is shown in the timing diagram of Figure 5-30. The intermediate waveform Y at the output of the OR gate is also shown.

Determine the output waveform if input A is a constant HIGH level.

Supplementary Problem

EXAMPLE 5-11

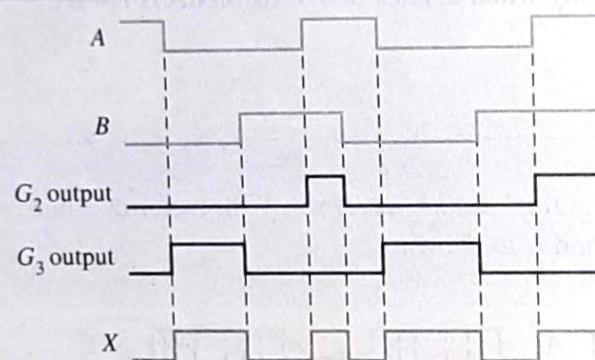
Draw the timing diagram for the circuit in Figure 5-31 showing the outputs of G_1 , G_2 , and G_3 with the input waveforms A , and B , as indicated.



▲ FIGURE 5-31

Solution

When both inputs are HIGH or when both inputs are LOW, the output X is HIGH as shown in Figure 5-32. Notice that this is an exclusive-NOR circuit. The intermediate outputs of gates G_2 and G_3 are also shown in Figure 5-32.



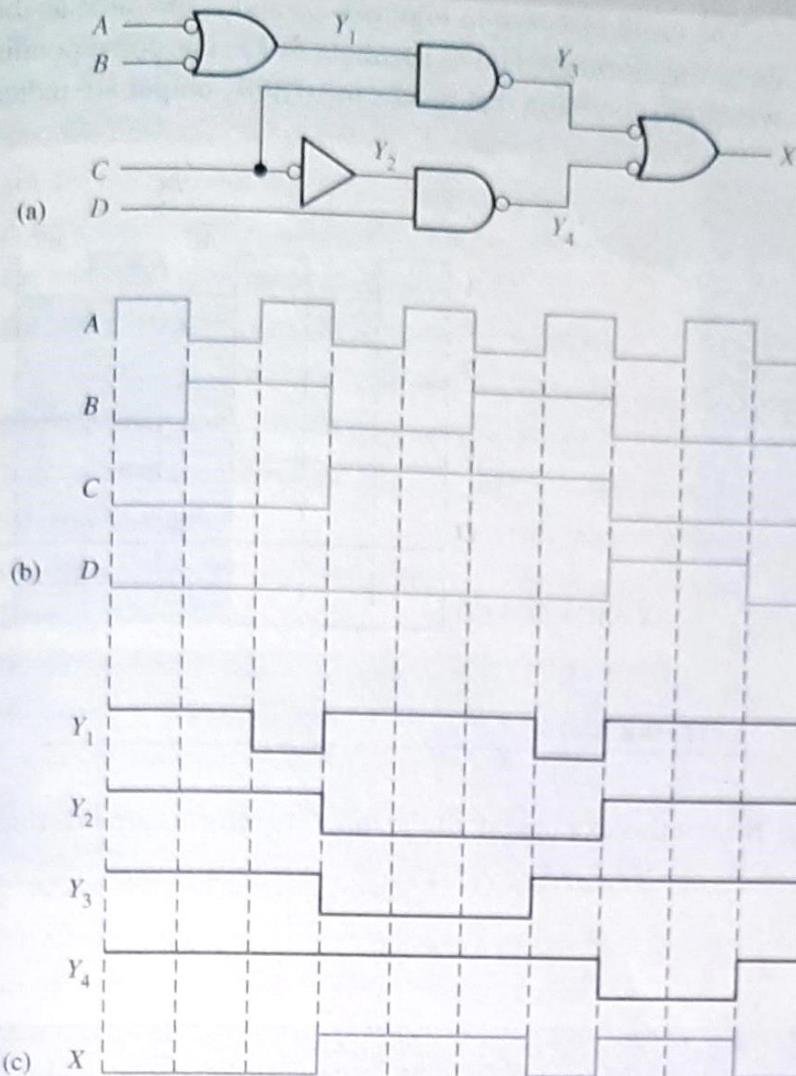
▲ FIGURE 5-32

Supplementary Problem

Determine the output X in Figure 5-31 if input B is inverted.

EXAMPLE 5-12

Determine the output waveform X for the logic circuit in Figure 5-33(a) by first finding intermediate waveform at each of points Y_1 , Y_2 , Y_3 , and Y_4 . The input waveforms are shown in Figure 5-33(b).



▲ FIGURE 5-33

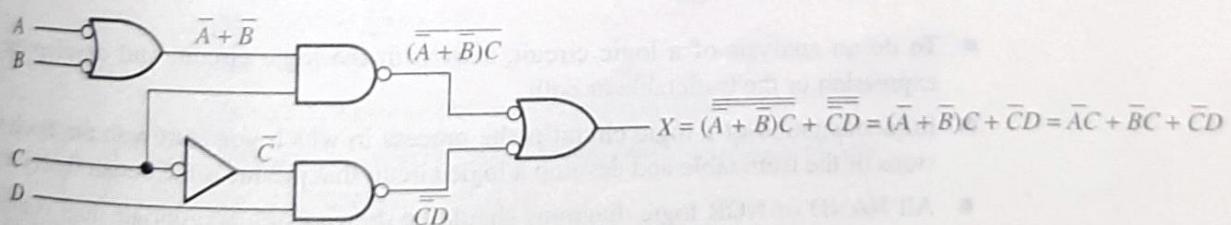
Solution All the intermediate waveforms and the final output waveform are shown in the timing diagram of Figure 5-33(c).

Supplementary Problem Determine the waveforms Y_1, Y_2, Y_3, Y_4 and X if input waveform A is inverted.

EXAMPLE 5-13

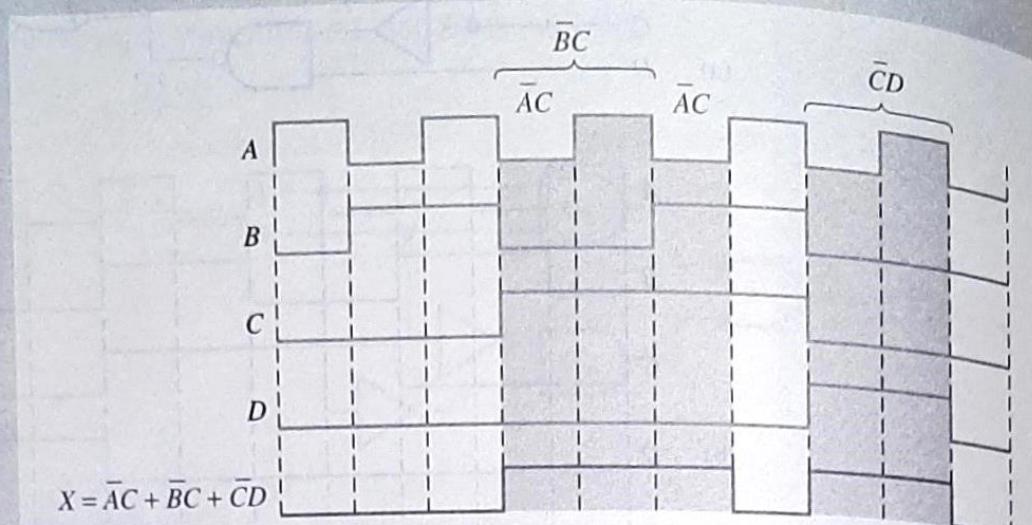
Determine the output waveform X for the circuit in Example 5-12, Figure 5-33(a), directly from the output expression.

Solution The output expression for the circuit is developed in Figure 5-34. The SOP form indicates that the output is HIGH when A is LOW and C is HIGH or when B is LOW and C is HIGH or when C is LOW and D is HIGH.



▲ FIGURE 5-34

The result is shown in Figure 5–35 and is the same as the one obtained by the intermediate-waveform method in Example 5–12. The corresponding product terms for each waveform condition that results in a HIGH output are indicated.



▲ FIGURE 5-35

Supplementary Problem Repeat this example if all the input waveforms are inverted.

SECTION 5-5 REVIEW

- One pulse with $t_w = 50 \mu s$ is applied to one of the inputs of an exclusive-OR circuit. A second positive pulse with $t_w = 10 \mu s$ is applied to the other input beginning $15 \mu s$ after the leading edge of the first pulse. Show the output in relation to the inputs.
- The pulse waveforms A and B in Figure 5–30 are applied to the exclusive-NOR circuit in Figure 5–31. Develop a complete timing diagram.

SUMMARY

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

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

- To do an analysis of a logic circuit, start with the logic circuit, and develop the Boolean expression or the truth table or both.
- Implementation of a logic circuit is the process in which you start with the Boolean output expressions or the truth table and develop a logic circuit that produces the output function.
- All NAND or NOR logic diagrams should be drawn using appropriate dual symbols so that outputs are connected to bubble inputs and nonbubble outputs are connected to nonbubble inputs.
- When two negation indicators (bubbles) are connected, they effectively cancel each other.