

Combinational Logic Circuits

Nouman M Durrani

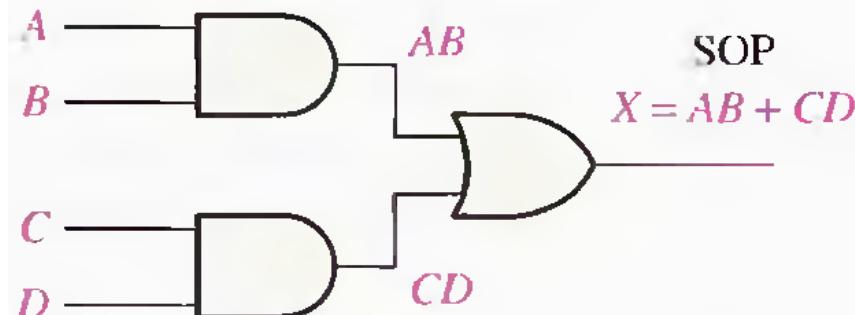
COMBINATIONAL LOGIC

- In combinational logic, the output level is at all times dependent on the combination of input levels.
- In this Chapter, we'll cover the following topics:
 - 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 function
 - Use NOR gates to implement any combinational logic function

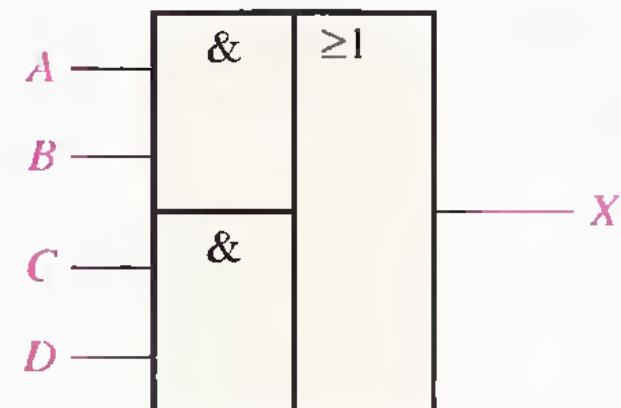
BASIC COMBINATIONAL LOGIC CIRCUITS

An AND-OR circuit directly implements an SOP expression.

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.



(a) Logic diagram (ANSI standard distinctive shape symbols)



(b) ANSI standard rectangular outline symbol

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.

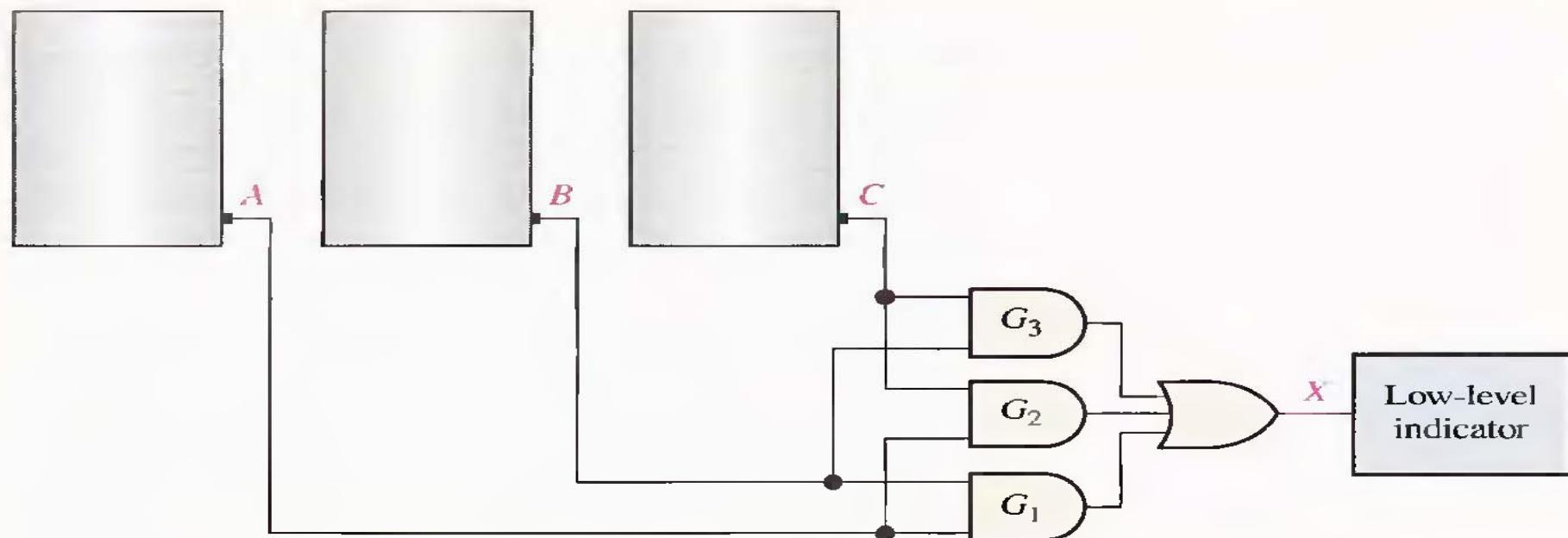
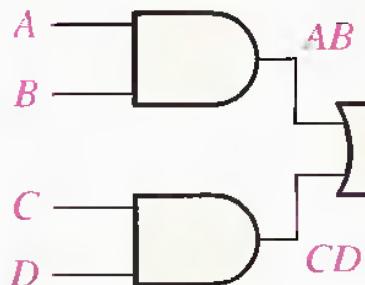


FIGURE 5–2

AND-OR-Invert Logic

- When the output of an AND-OR circuit is complemented (inverted), it results in an AND-OR-Invert circuit.
- POS expressions can be implemented with AND-OR-Invert logic.

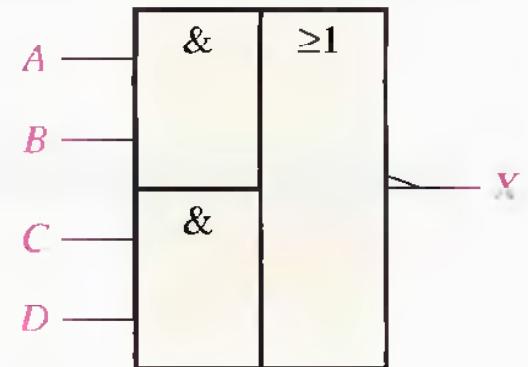
$$X = (\bar{A} + \bar{B})(\bar{C} + \bar{D}) = (\overline{AB})(\overline{CD}) = \overline{\overline{AB}\overline{CD}} = \overline{\overline{AB}} + \overline{\overline{CD}} = \overline{AB + CD}$$



(a)

FIGURE 5–3

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



(b)

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

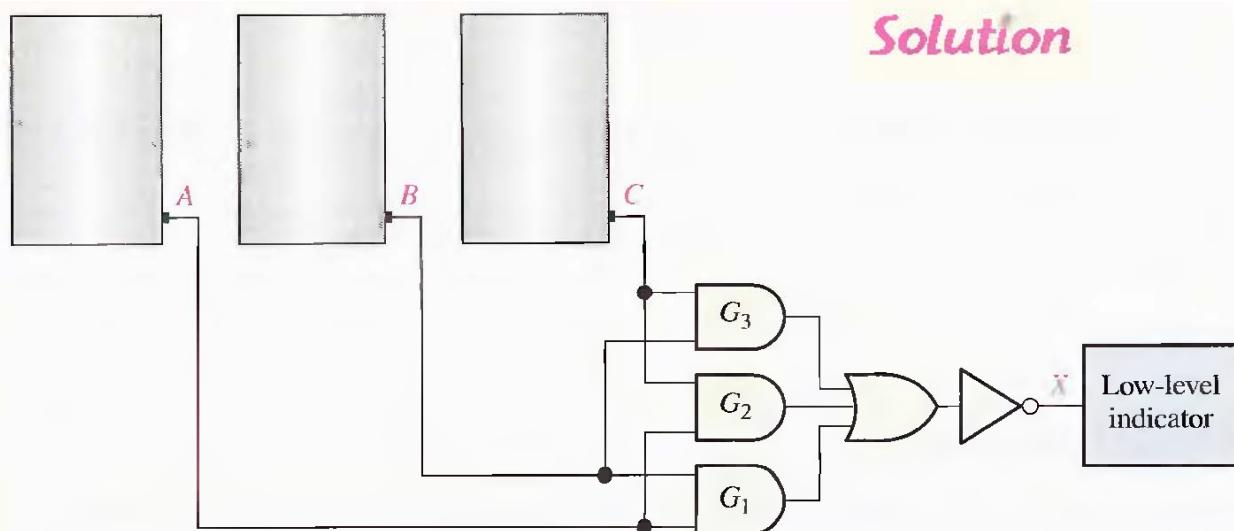
For a 4-input AND-OR-Invert logic circuit, the output X is LOW (0) if both input A and input B are HIGH (1) or both input C and input D are HIGH (1).

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

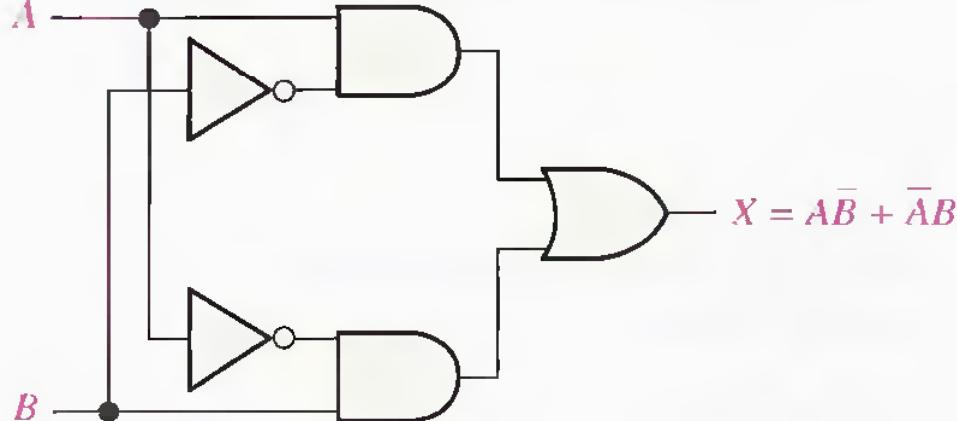


▲ FIGURE 5-4

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.

Exclusive-OR Logic

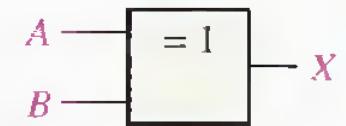
- The exclusive-OR is a combination of two AND gates, one OR gate, and two inverters, as shown in Figure 5-5(a).
- The two ANSI standard logic symbols are shown in parts (b) and (c).



(a) Logic diagram



(b) ANSI distinctive shape symbol



(c) ANSI rectangular outline symbol

FIGURE 5-5

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

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

$$X = A \oplus B$$

Exclusive-NOR Logic

- The complement of the exclusive-OR function is the exclusive-NOR, which is derived as follows:

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

The exclusive-NOR can be implemented by simply inverting the output of an exclusive-OR, as shown in Figure 5–6(a), or by directly implementing the expression $\overline{\bar{A}\bar{B}} + AB$, as shown in part (b).

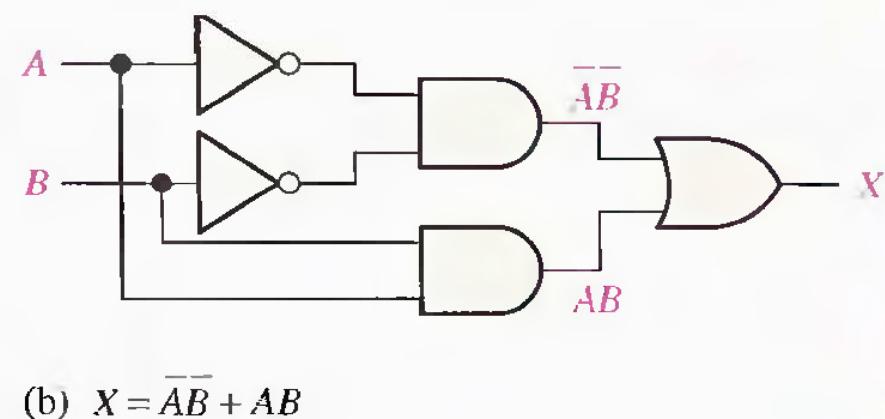
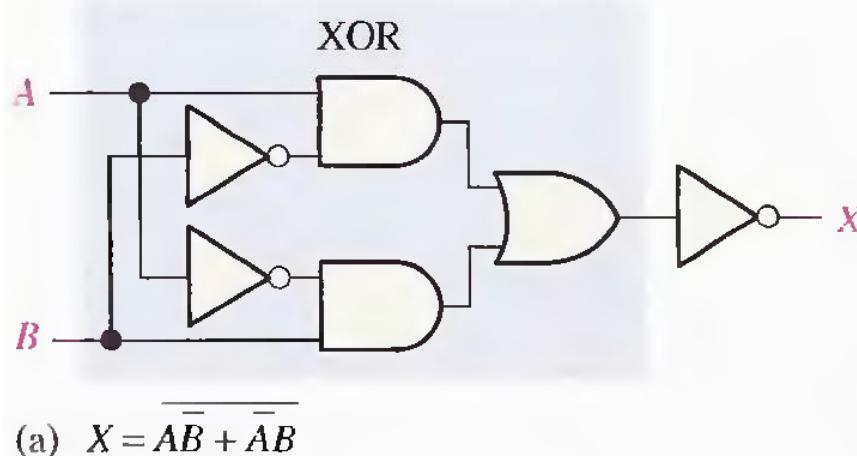
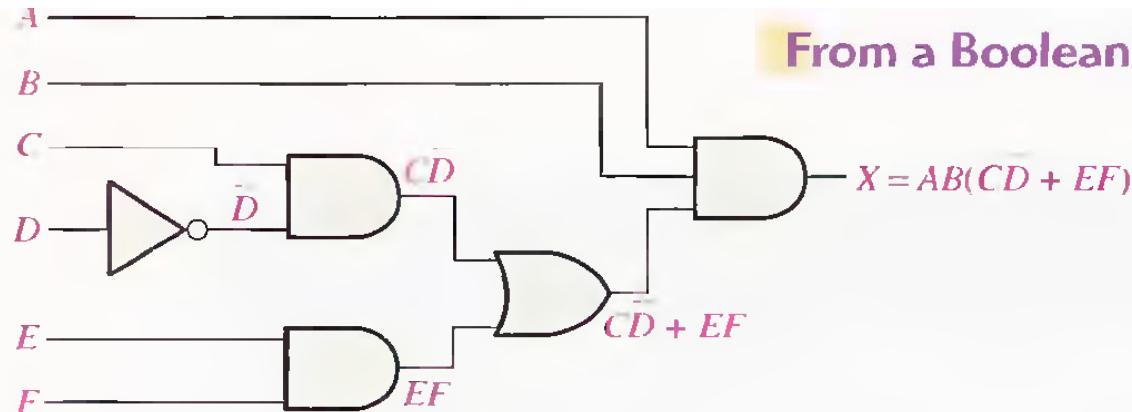


FIGURE 5–6

IMPLEMENTING COMBINATIONAL LOGIC

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

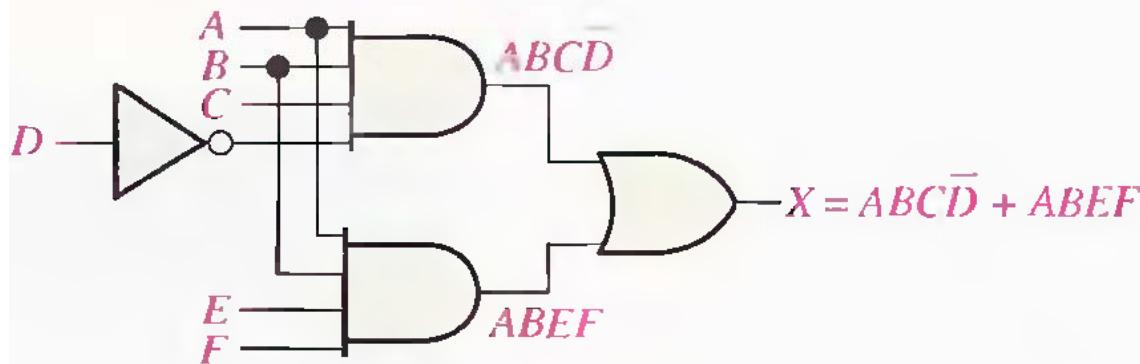
From a Boolean Expression to a Logic Circuit



(a)

FIGURE 5–8

Logic circuits for $X = AB(\bar{CD} + EF) = ABC\bar{D} + ABEF$.



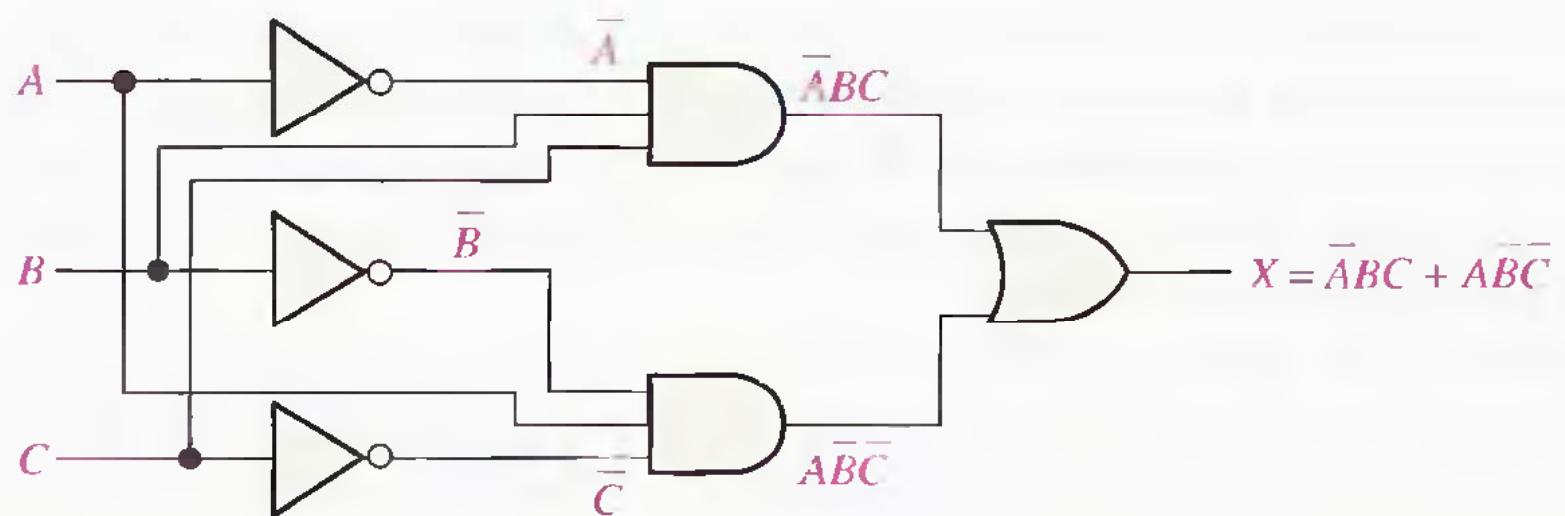
(b) Sum-of-products implementation of the circuit in part (a)

From a Truth Table to a Logic Circuit

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 which $X = 1$ is

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



EXAMPLE 5–3

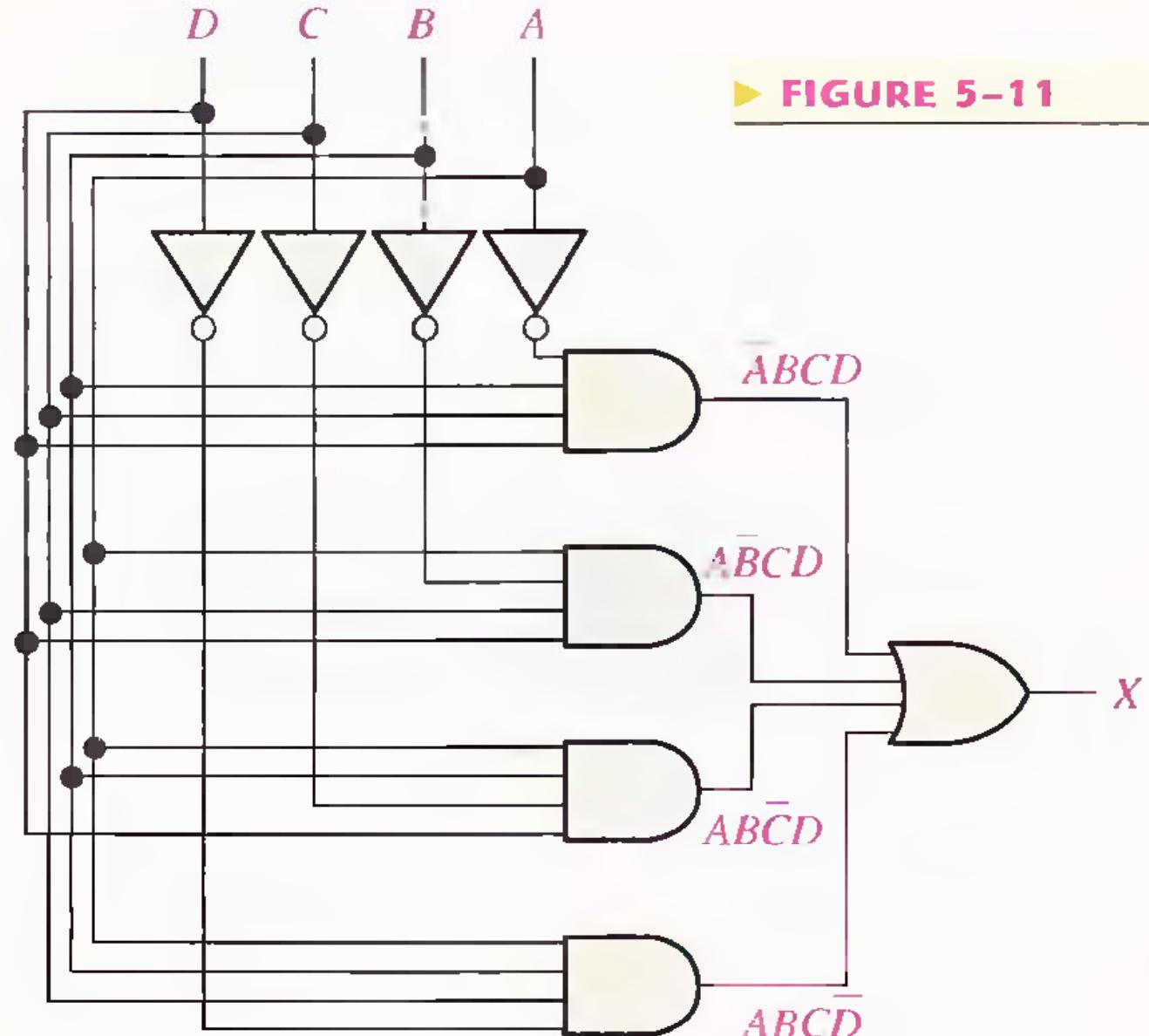
Develop a logic circuit with four input variables that will only produce a 1 output when exactly three input variables are 1s.

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.

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	$AB\bar{C}D$
1	1	1	0	$ABC\bar{D}$

The product terms are ORed to get the following expression:

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

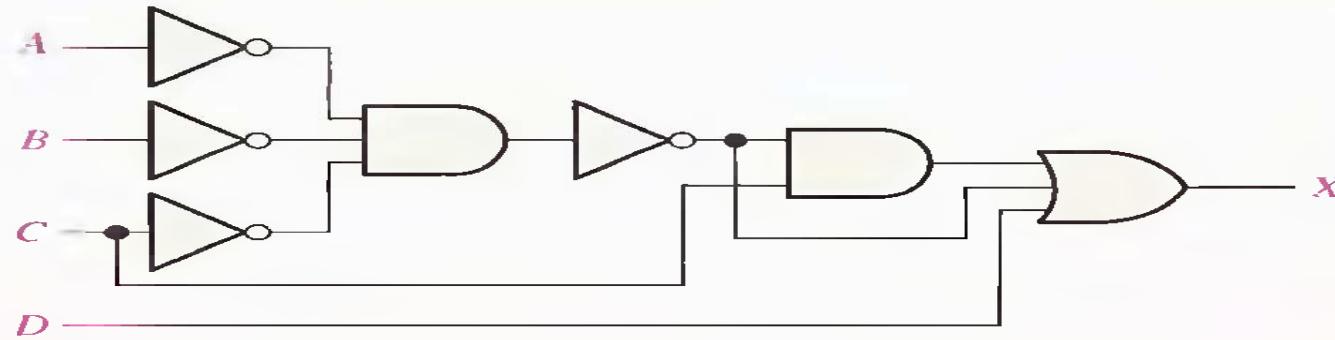


► FIGURE 5–11

Related Problem Determine if the logic circuit of Figure 5–11 can be simplified.

EXAMPLE 5–5

Reduce the combinational logic circuit in Figure 5–12 to a minimum form.



The expression for the output of the circuit is

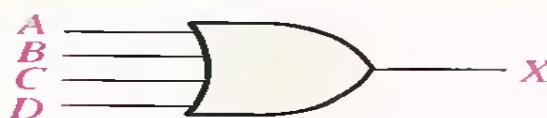
$$X = (\overline{A}\overline{B}\overline{C})C + \overline{A}\overline{B}\overline{C} + D$$

Applying DeMorgan's theorem and Boolean algebra,

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

The simplified circuit is a 4-input OR gate as shown in Figure 5–13.

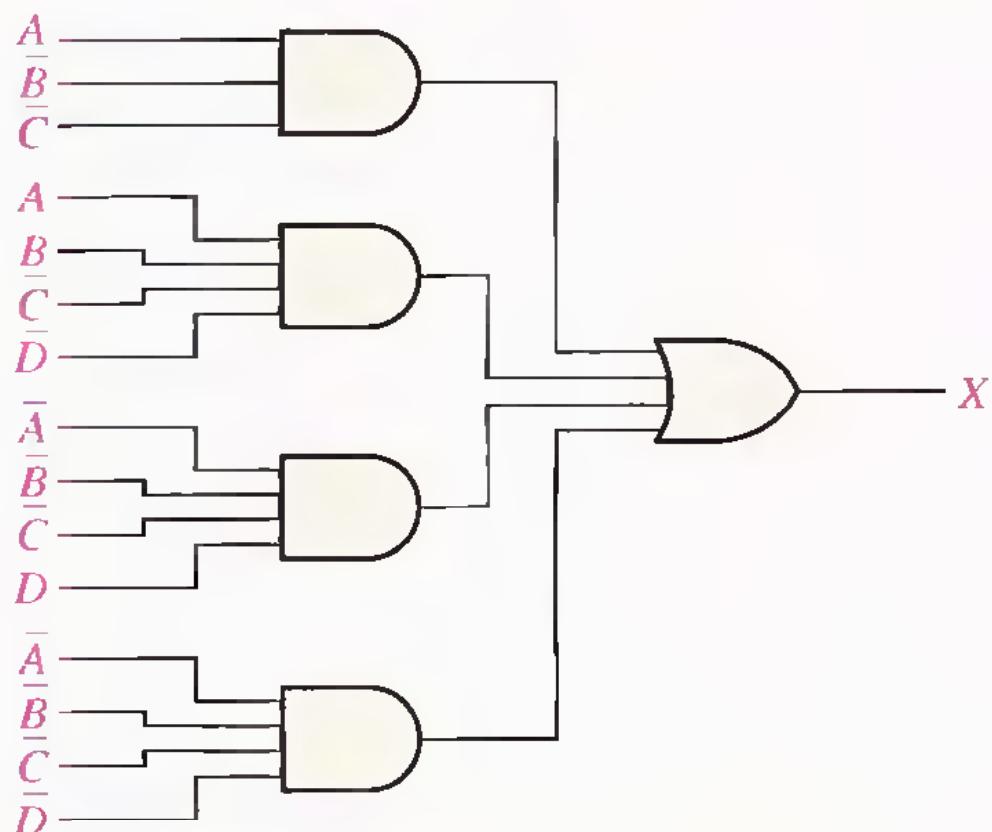
FIGURE 5–13



EXAMPLE 5–6

Minimize the combinational logic circuit in Figure 5–14. Inverters for the complemented variables are not shown.

► FIGURE 5–14



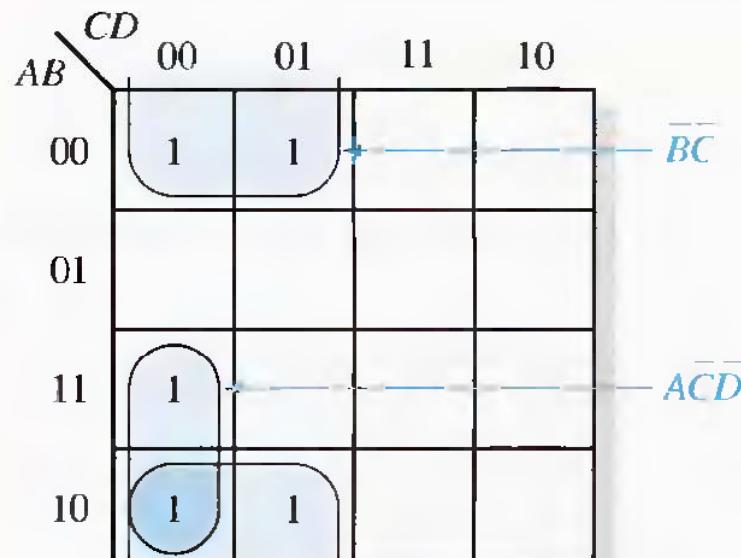
The output expression is

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

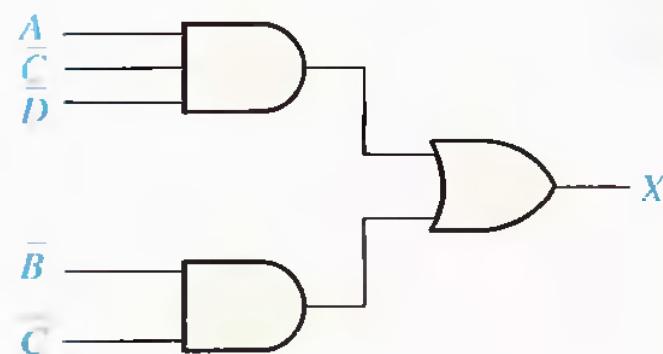
Expanding the first term to include the missing variables D and \bar{D} ,

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

This expanded SOP expression is mapped and simplified on the Karnaugh map in Figure 5–15(a). The simplified implementation is shown in part (b). Inverters are not shown.



(a)



(b)

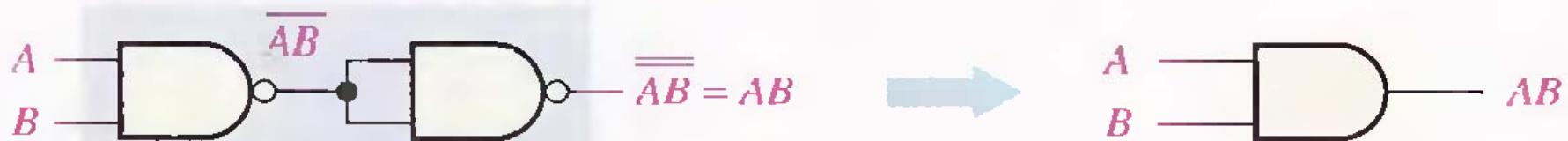
FIGURE 5–15

The NAND Gate as a Universal Logic Element

- The NAND gate is a universal gate because it can be used to produce the NOT, the AND, the OR, and the NOR functions. An inverter can be made from a NAND gate by connecting all of the inputs together and creating, in effect, a single input, as shown in Figure 5-16(a) for a 2-input gate.



(a) One NAND gate used as an inverter



(b) Two NAND gates used as an AND gate

 FIGURE 5-16

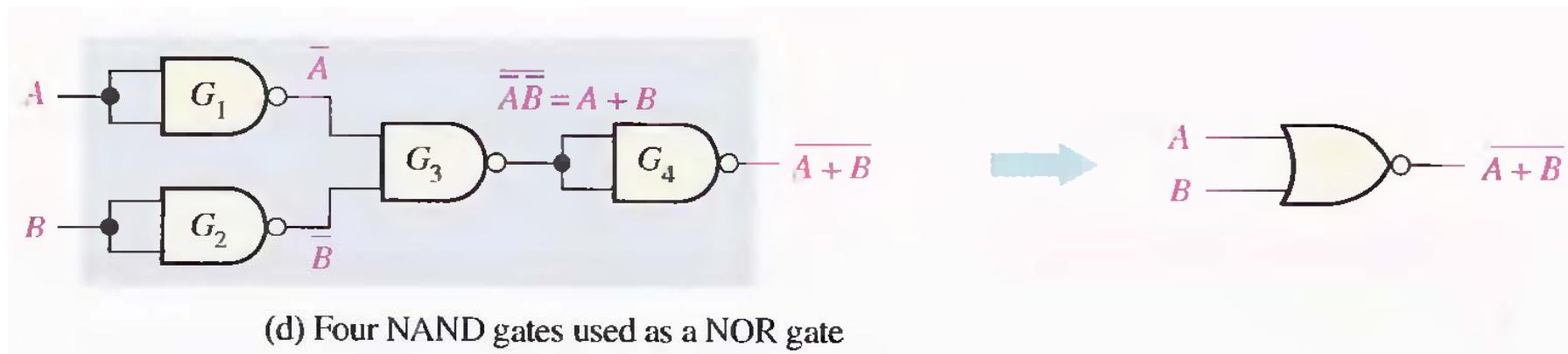
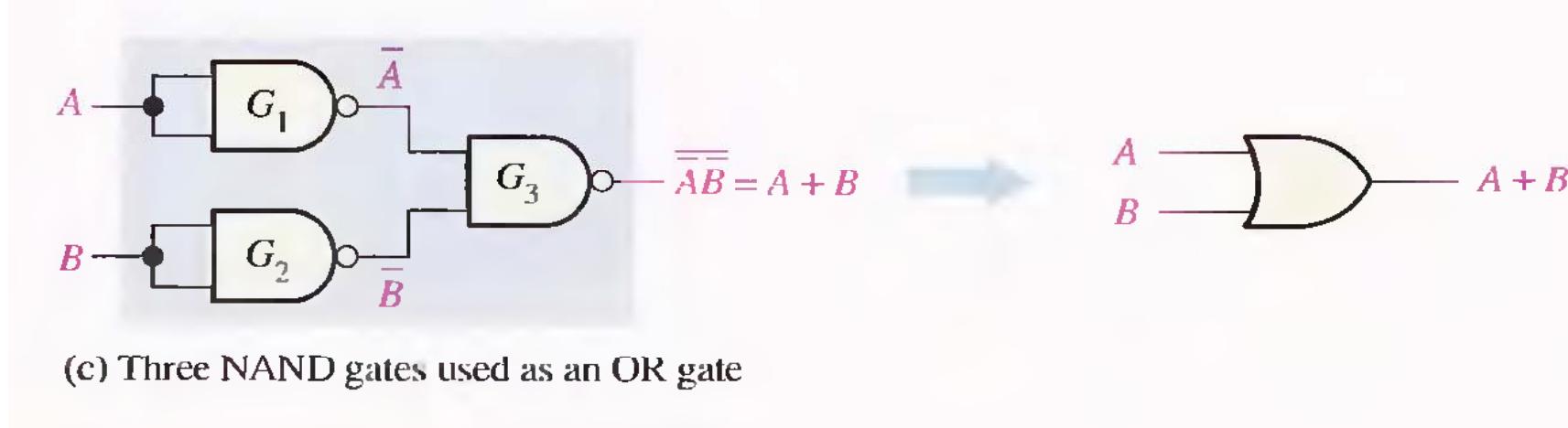


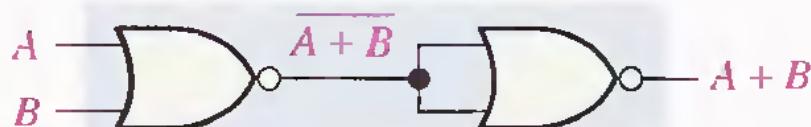
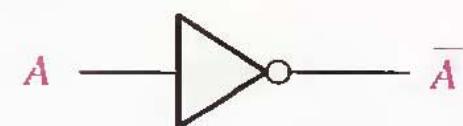
FIGURE 5-16

The NOR Gate as a Universal Logic Element

- NOR gate can be used to produce the NOT, AND, OR and NAND functions.
- A NOT circuit, or inverter, can be made from a NOR gate by connecting all of the inputs together to effectively create a single input, as shown in Figure 5-17



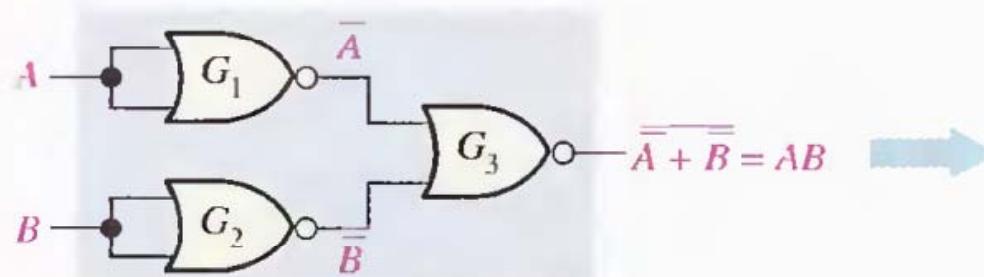
(a) One NOR gate used as an inverter



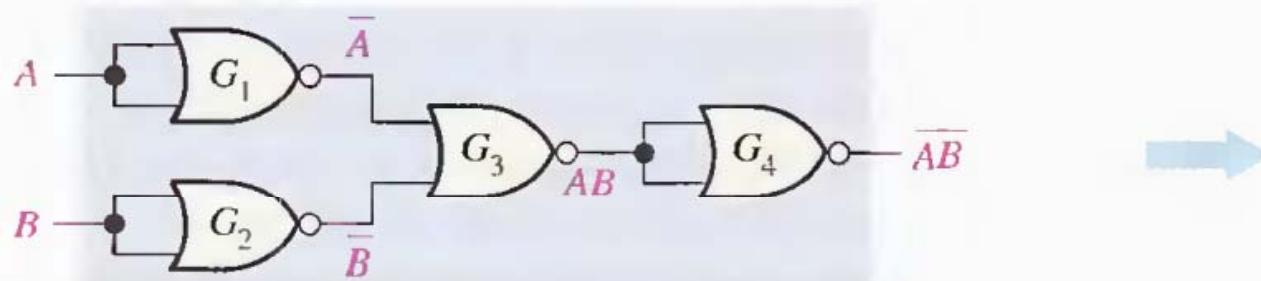
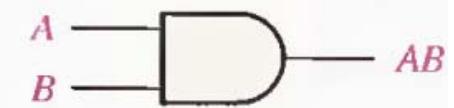
(b) Two NOR gates used as an OR gate



◀ FIGURE 5-17



(c) Three NOR gates used as an AND gate



(d) Four NOR gates used as a NAND gate



FIGURE 5-17

COMBINATIONAL LOGIC USING NAND AND NOR GATES

- NAND gate can function as either a NAND or a negative-OR because, by DeMorgan's theorem,

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

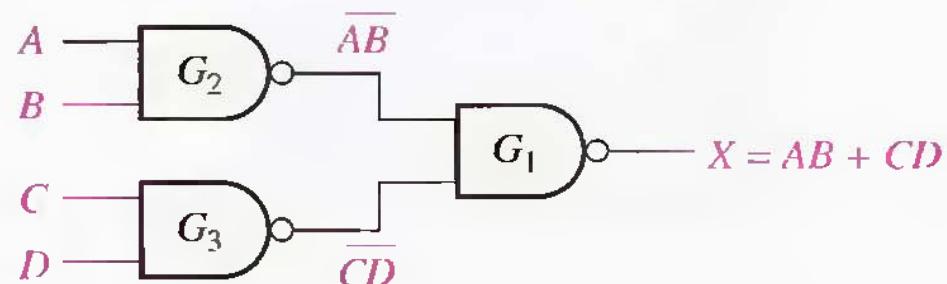
↑ ↑
NAND negative-OR

Consider the NAND logic in Figure 5–18. The output expression is developed in the following steps:

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

► FIGURE 5–18

NAND logic for $X = AB + CD$.

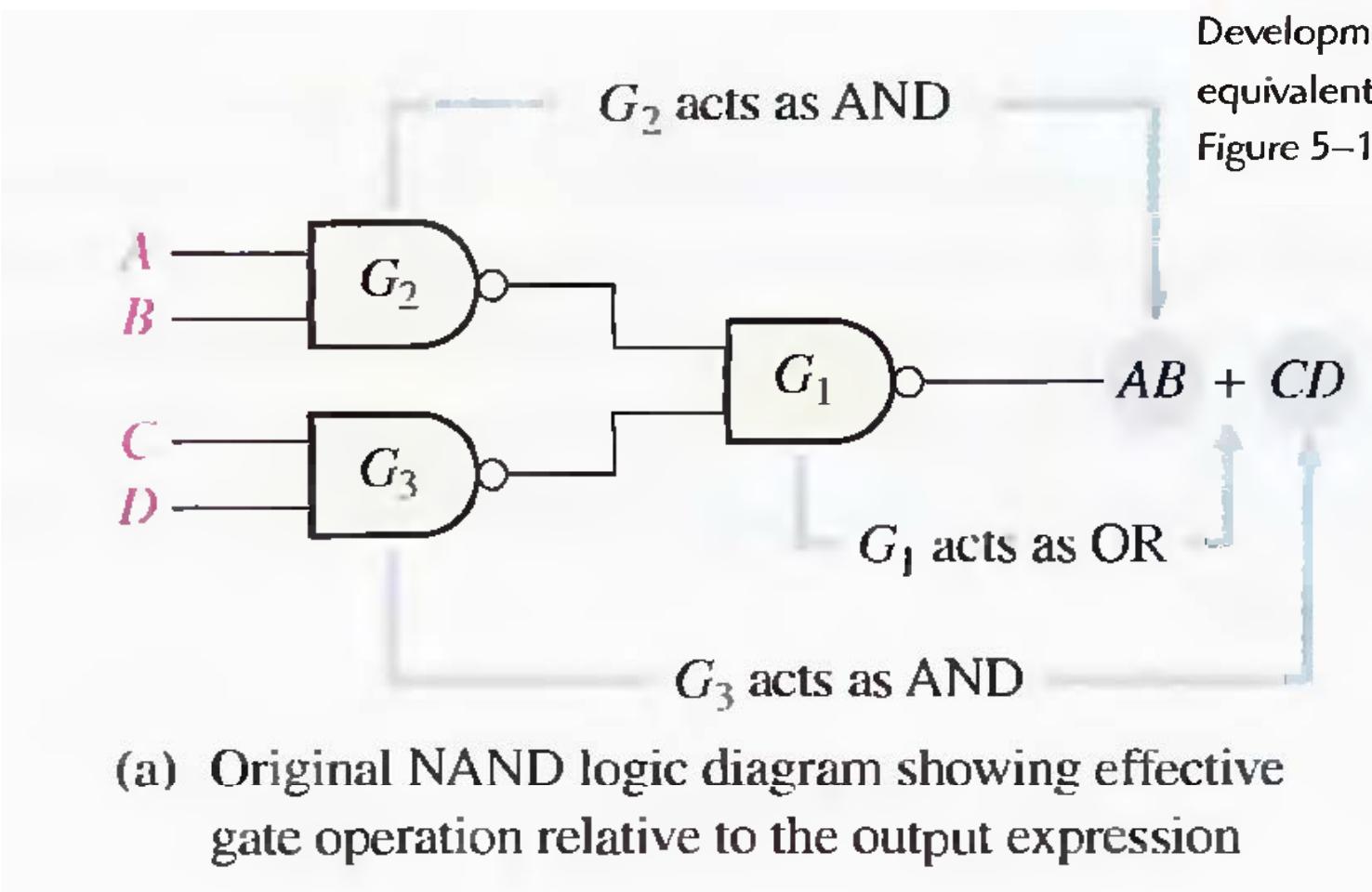


NAND Logic Diagrams Using Dual Symbols

- The NAND symbol and the negative-OR symbol are called dual symbols.

◀ FIGURE 5–19

Development of the AND-OR equivalent of the circuit in Figure 5–18.



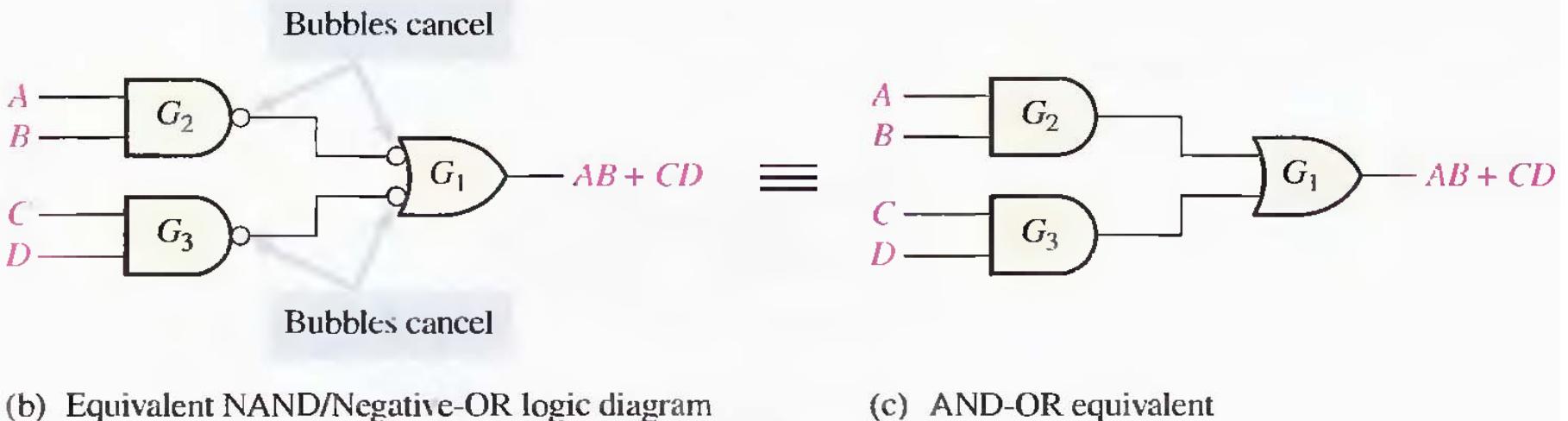
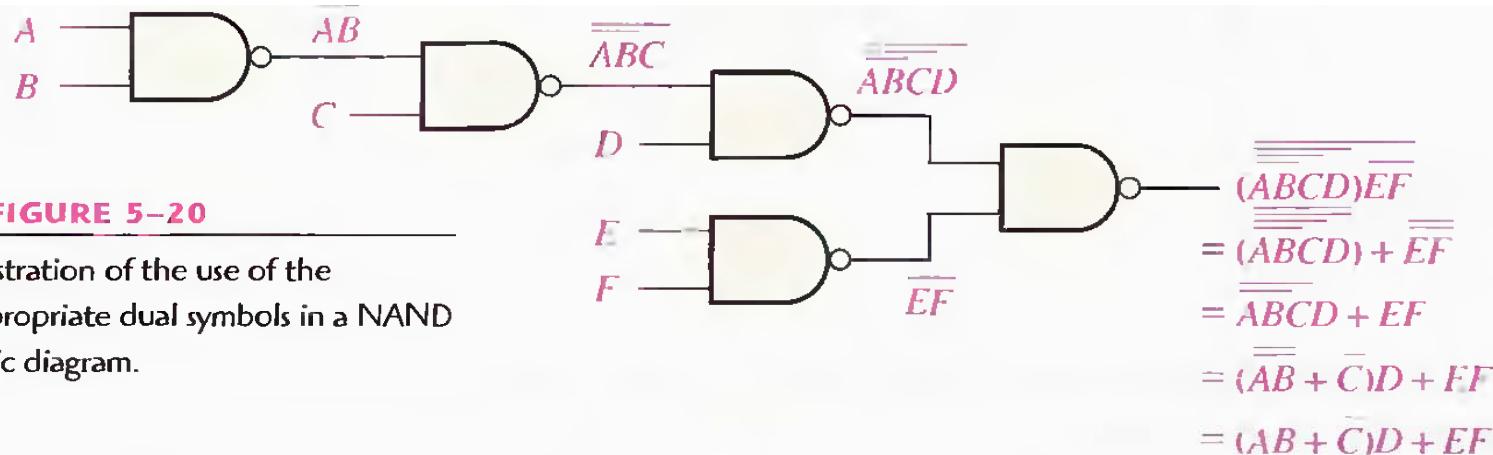
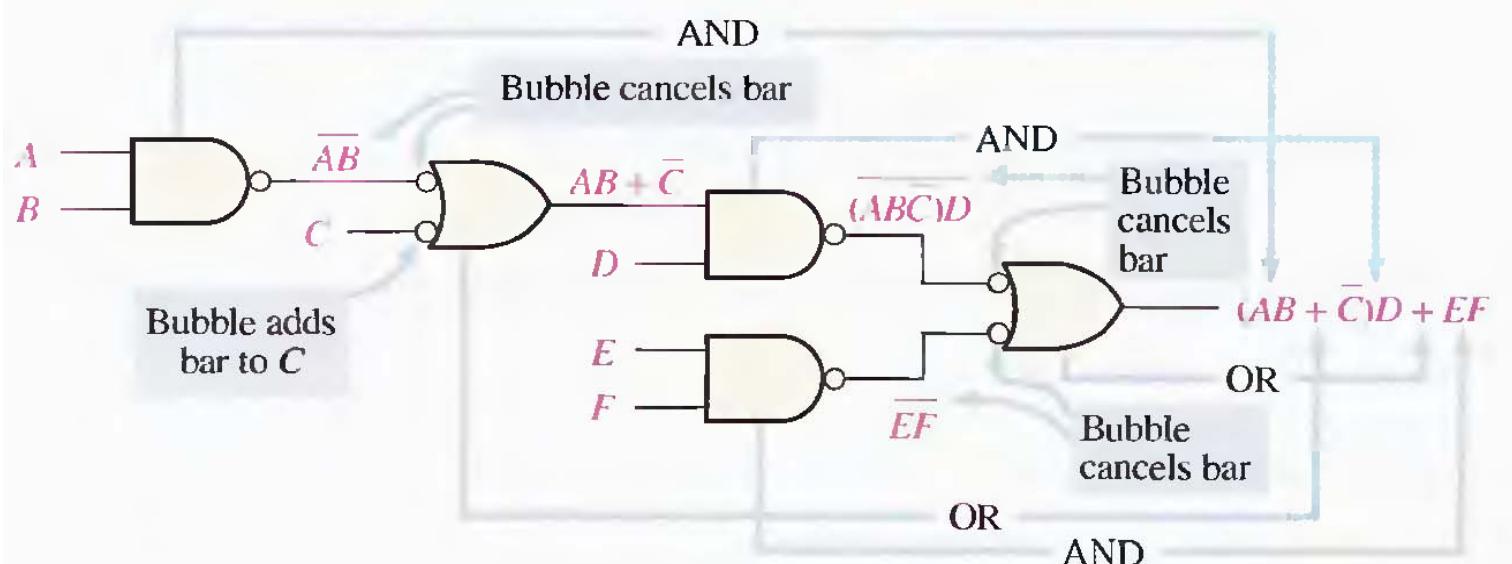


FIGURE 5–19



(a) Several Boolean steps are required to arrive at final output expression.

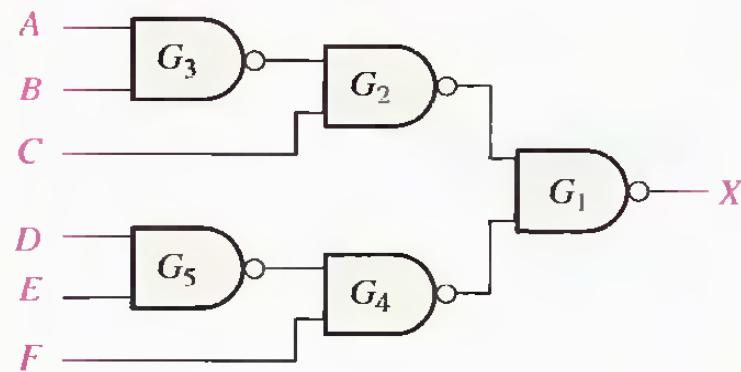


(b) Output expression can be obtained directly from the function of each gate symbol in the diagram.

EXAMPLE 5–7

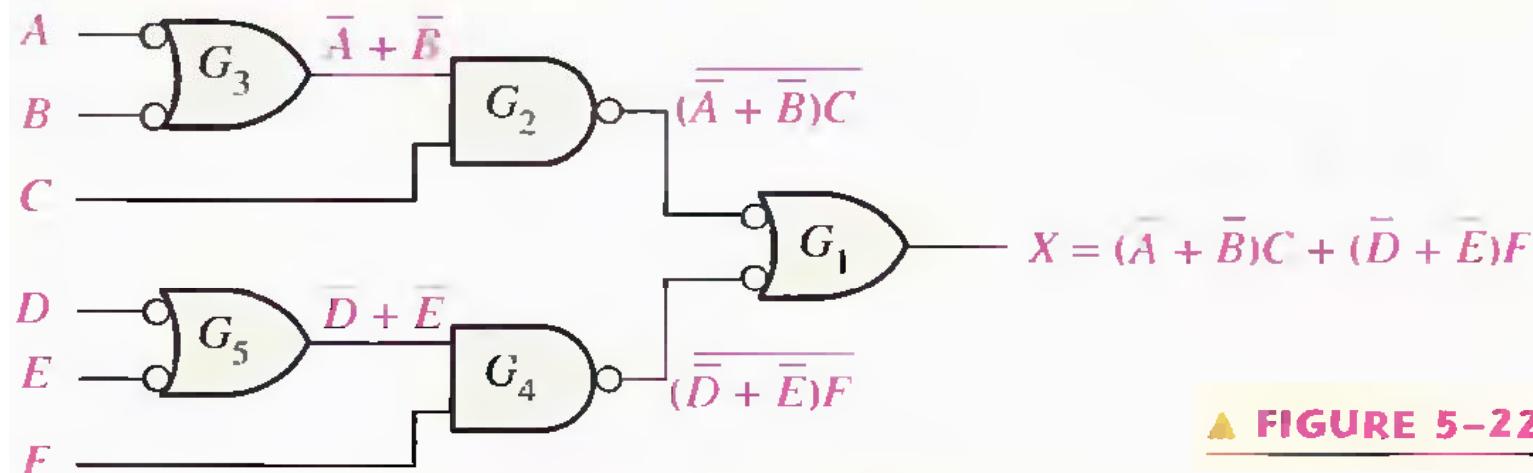
Redraw the logic diagram and develop the output expression for the circuit in Figure 5–21 using the appropriate dual symbols.

► FIGURE 5–21



Solution

Redraw the logic diagram in Figure 5–21 with the use of equivalent negative-OR symbols as shown in Figure 5–22. 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 5–22

EXAMPLE 5–8

Implement each expression with NAND logic using appropriate dual symbols:

(a) $ABC + DE$

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

Solution See Figure 5–23.

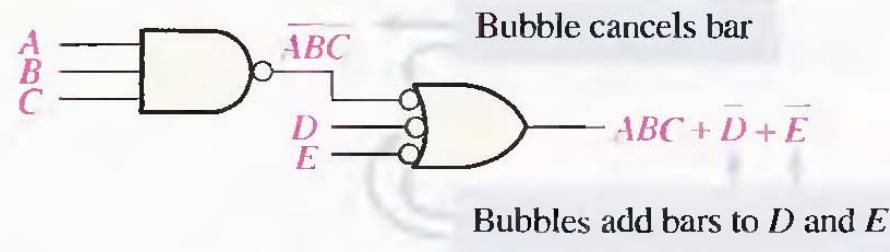
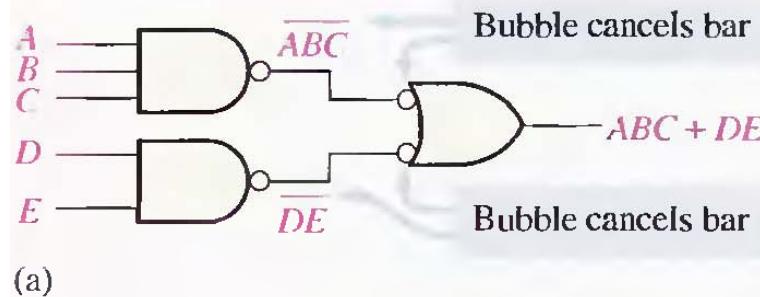


FIGURE 5–23

Related Problem Convert the NAND circuits in Figure 5–23(a) and (b) to equivalent AND-OR logic.

NOR Logic

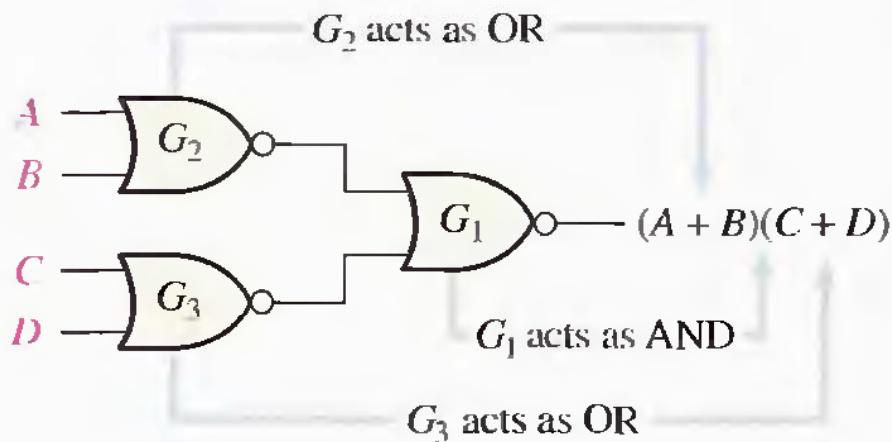
A NOR gate can function as either a NOR or a negative-AND, as shown by DeMorgan's theorem.

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

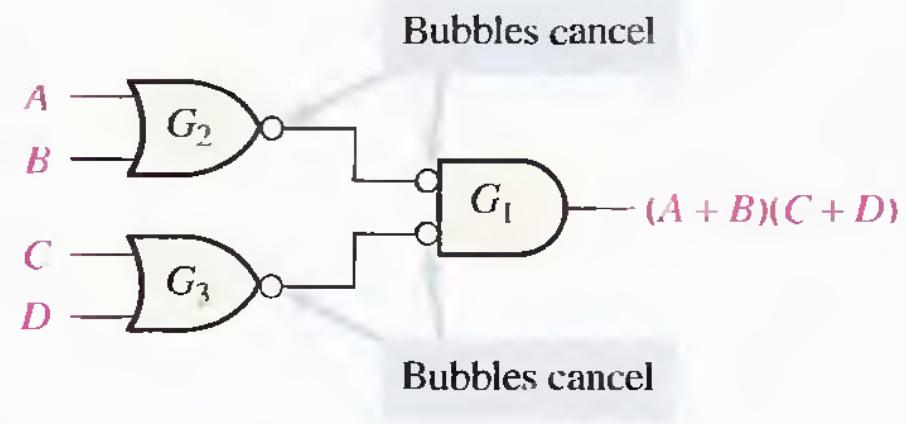
↑ ↑
NOR negative-AND

Consider the NOR logic in Figure 5–24. The output expression is developed as follows:

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



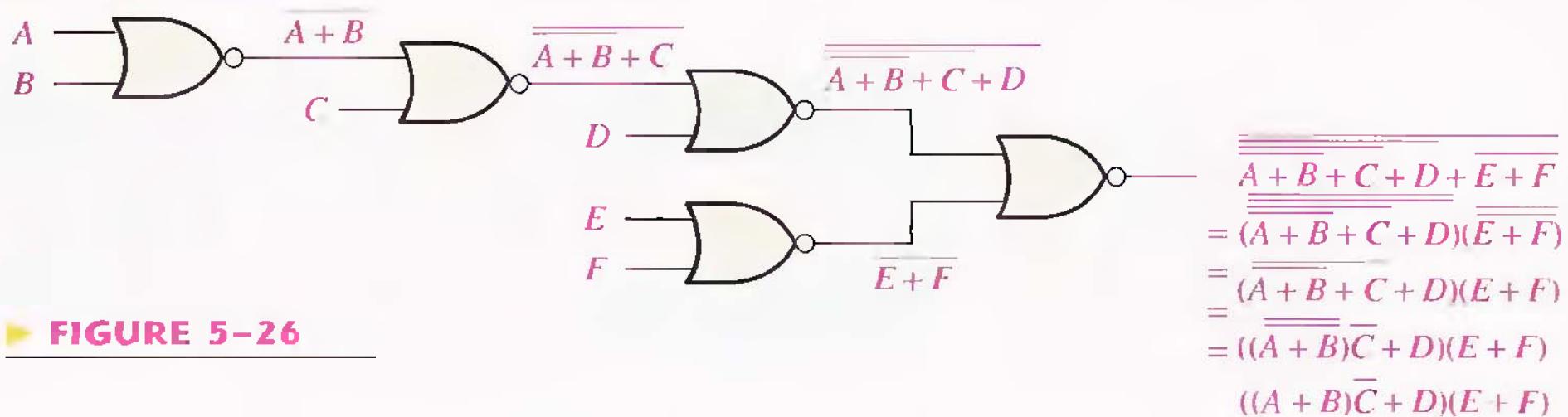
(a)



(b)

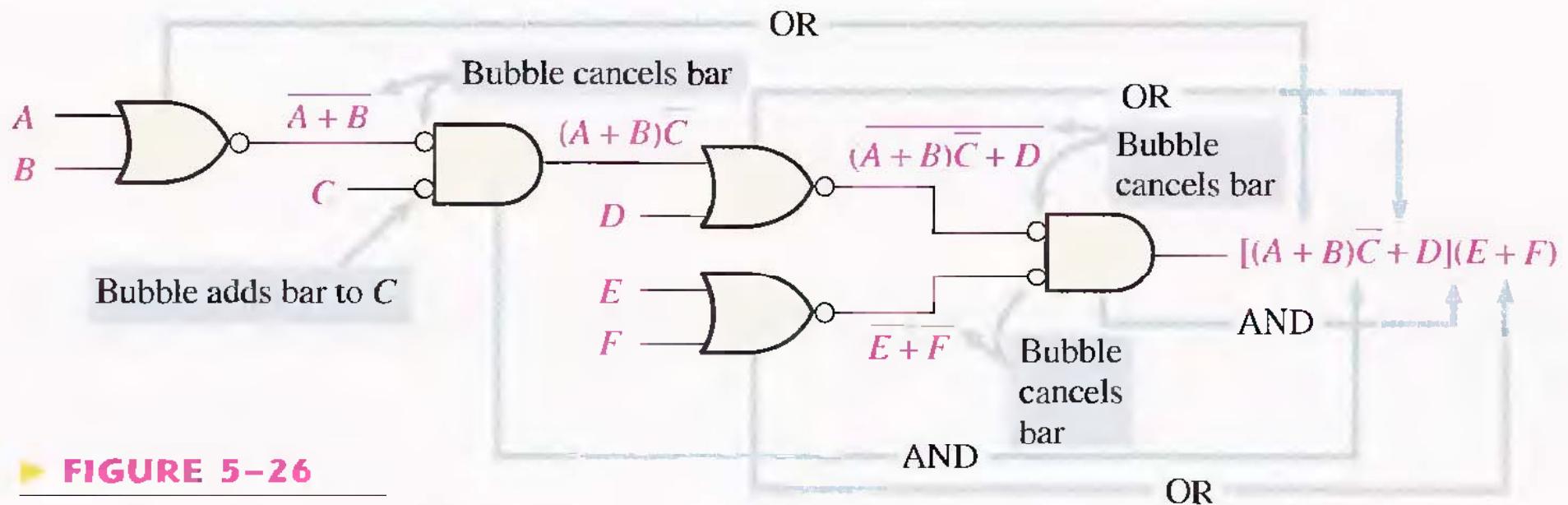
◀ FIGURE 5–24

NOR Logic Diagram Using Dual Symbols



► FIGURE 5–26

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



► FIGURE 5–26

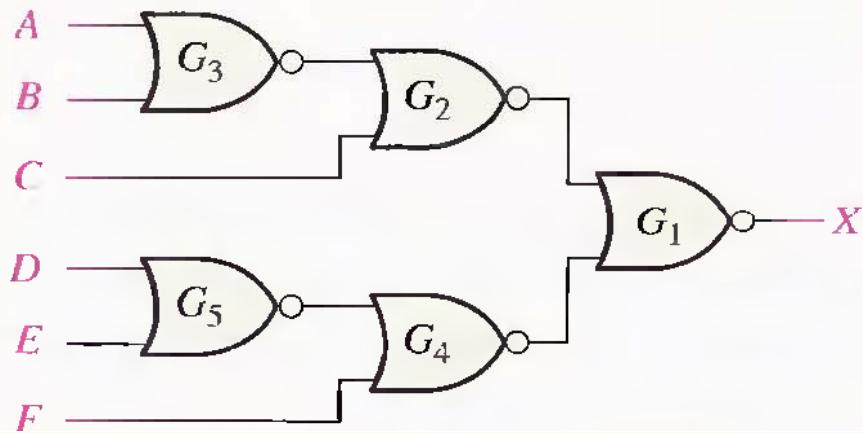
(b) Output expression can be obtained directly from the function of each gate symbol in the diagram.

EXAMPLE 5-9

Using appropriate dual symbols, redraw the logic diagram and develop the output expression for the circuit in Figure 5-27.

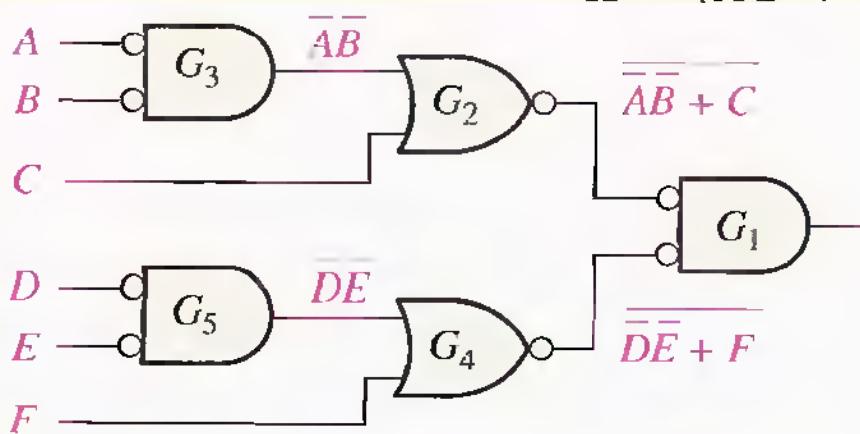
FIGURE 5-27

Solution



Redraw the logic diagram with the equivalent negative-AND symbols as shown in Figure 5-28. Writing the expression for X directly from the indicated operation of each gate,

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

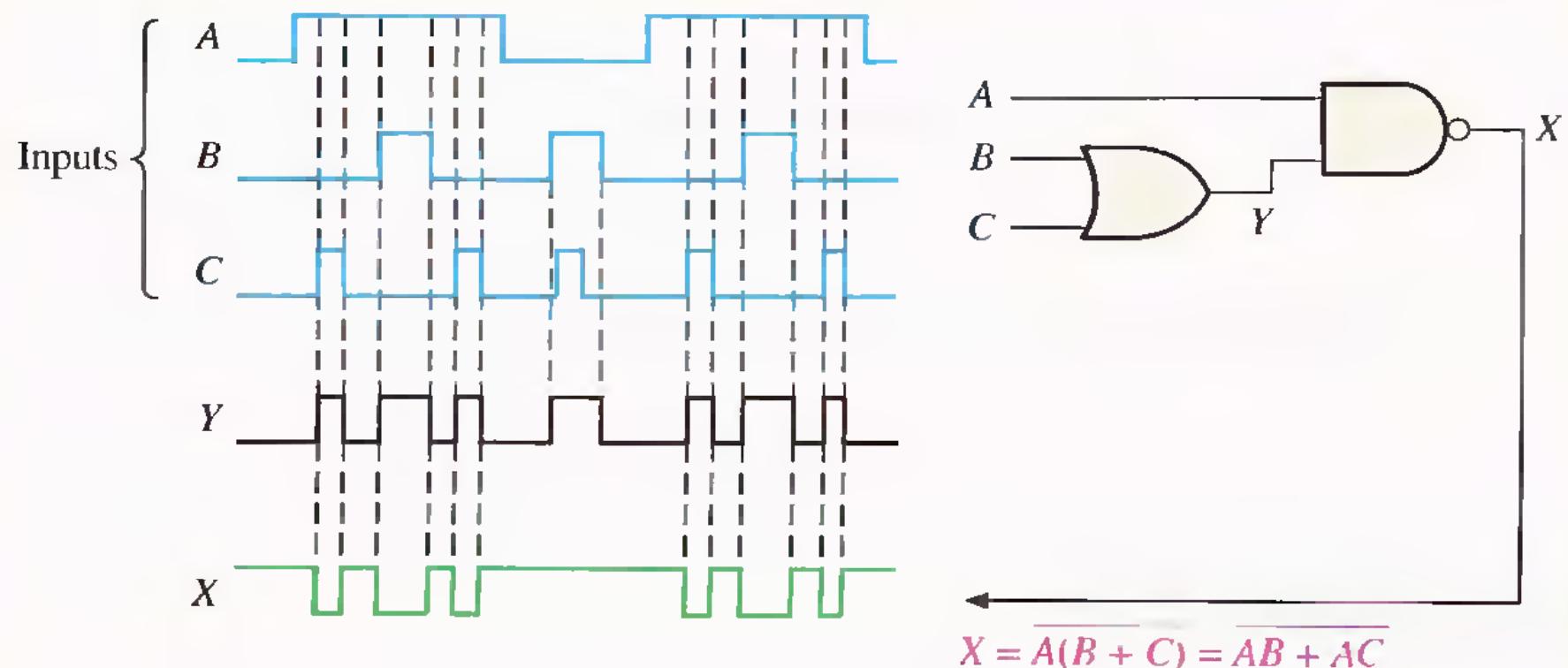


$$\begin{aligned} &= (\overline{\overline{AB}} + \overline{C})(\overline{\overline{DE}} + \overline{F}) \\ &= (\overline{AB} + C)(\overline{DE} + F) = (\overline{AB} + C)(\overline{\overline{DE}} + F) \end{aligned}$$

FIGURE 5-28

LOGIC CIRCUIT OPERATION WITH PULSE WAVEFORM INPUTS

Determine the final output waveform X for the circuit in Figure 5–29, with input waveforms A , B , and C as shown.



▲ FIGURE 5-29

EXAMPLE 5-11

Draw the timing diagram for the circuit in Figure 5–30 showing the outputs of G_1 , G_2 , and G_3 with the input waveforms, A , and B , as indicated.

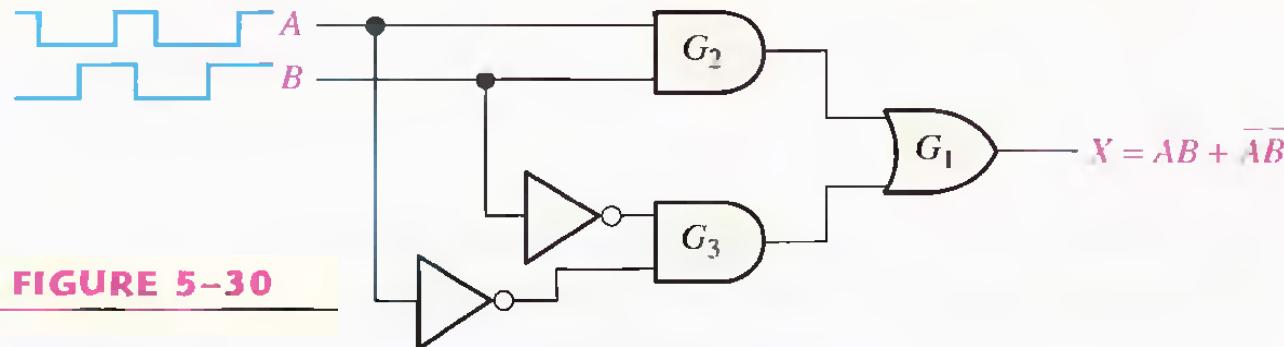


FIGURE 5–30

Solution When both inputs are HIGH or when both inputs are LOW, the output X is HIGH as shown in Figure 5–31. Notice that this is an exclusive-NOR circuit. The intermediate outputs of gates G_2 and G_3 are also shown in Figure 5–31.

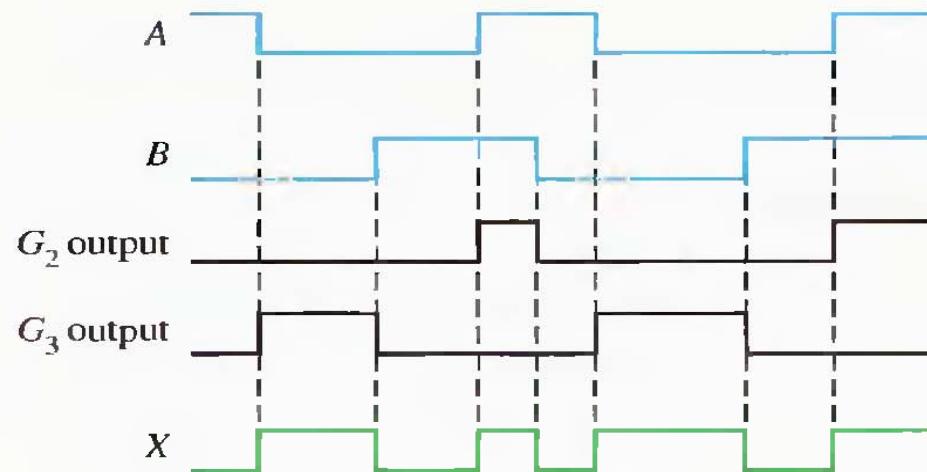
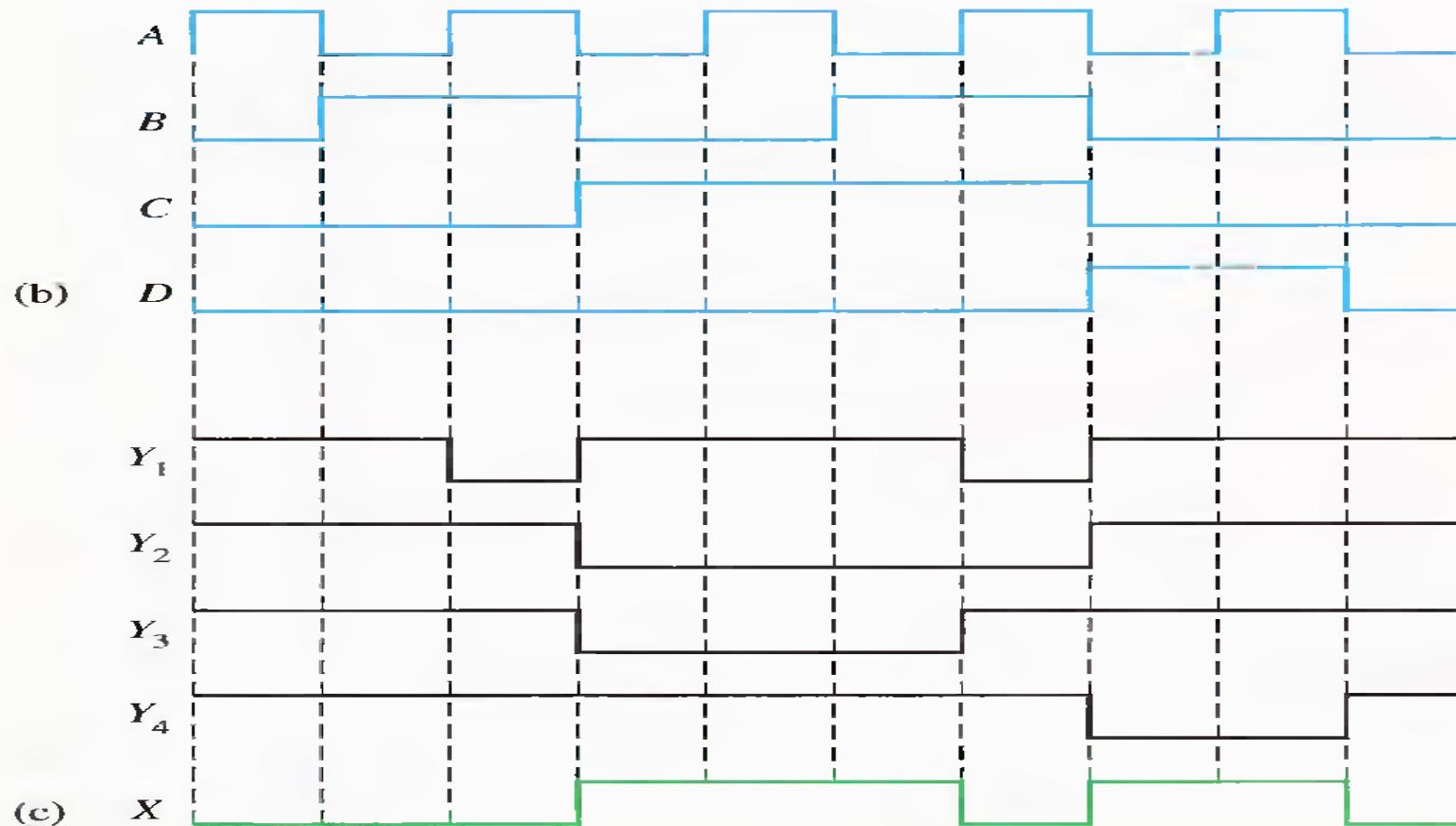
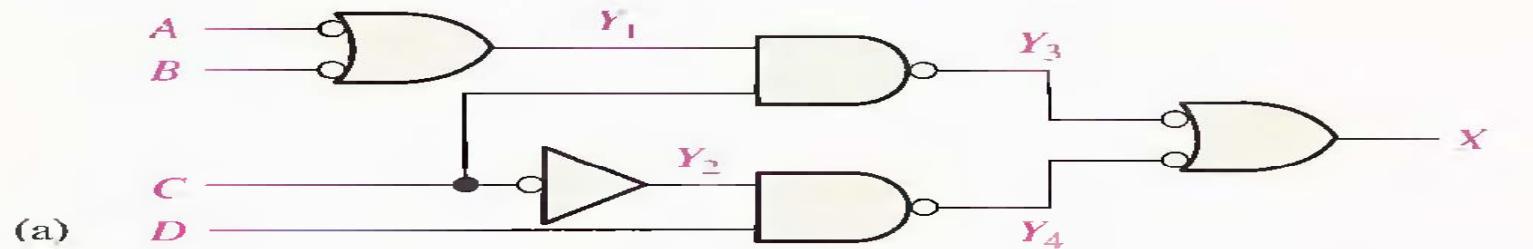


FIGURE 5–31

Determine the output waveform X for the logic circuit in Figure 5–32(a) by first finding the intermediate waveform at each of points Y_1 , Y_2 , Y_3 , and Y_4 . The input waveforms are shown in Figure 5–32(b).



EXAMPLE 5-13

Determine the output waveform X for the circuit in Example 5-12, Figure 5-32(a), directly from the output expression.

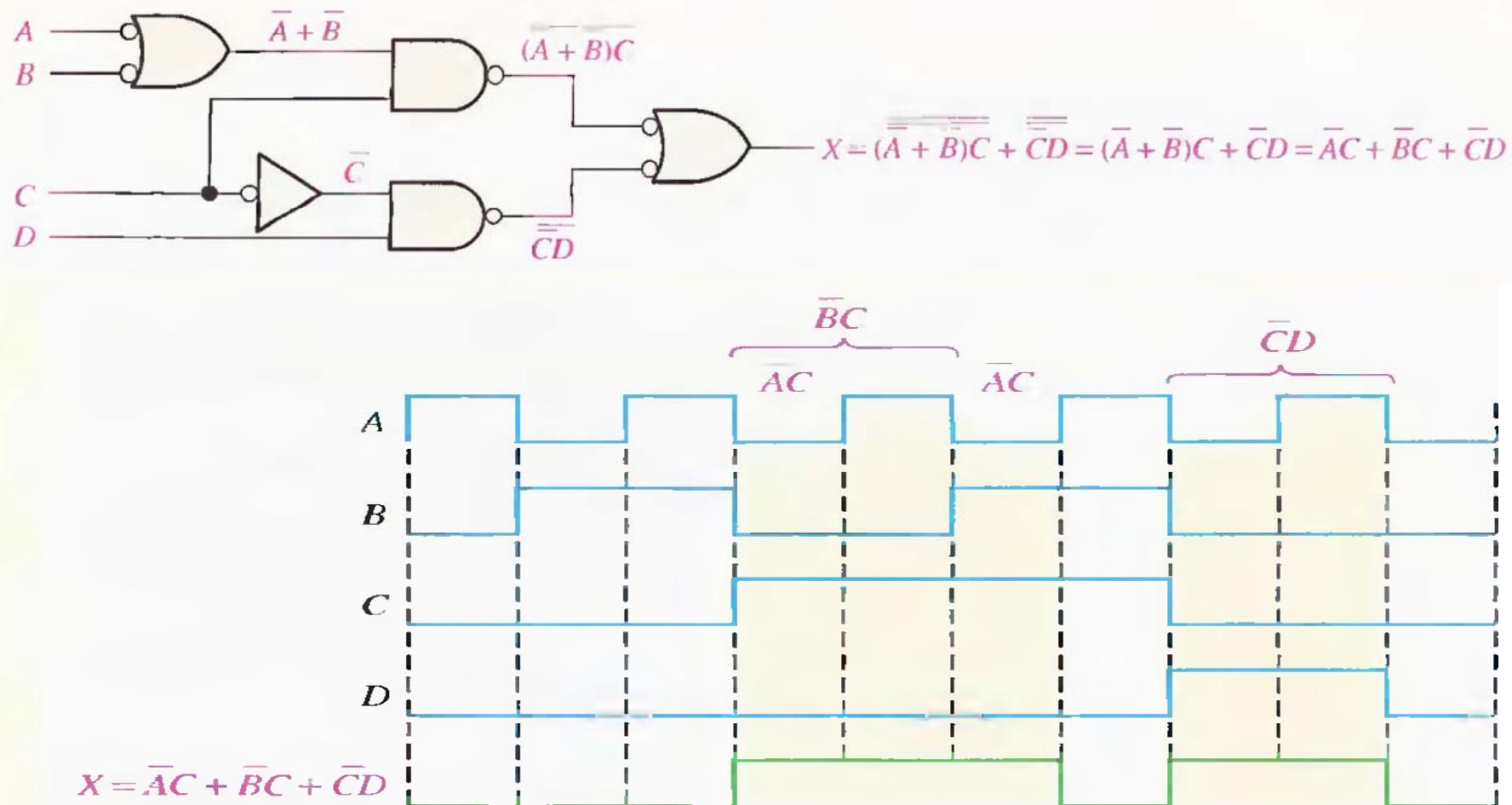
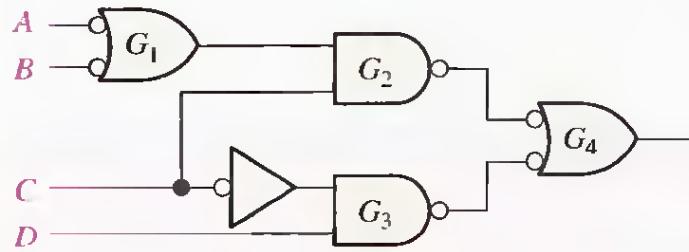


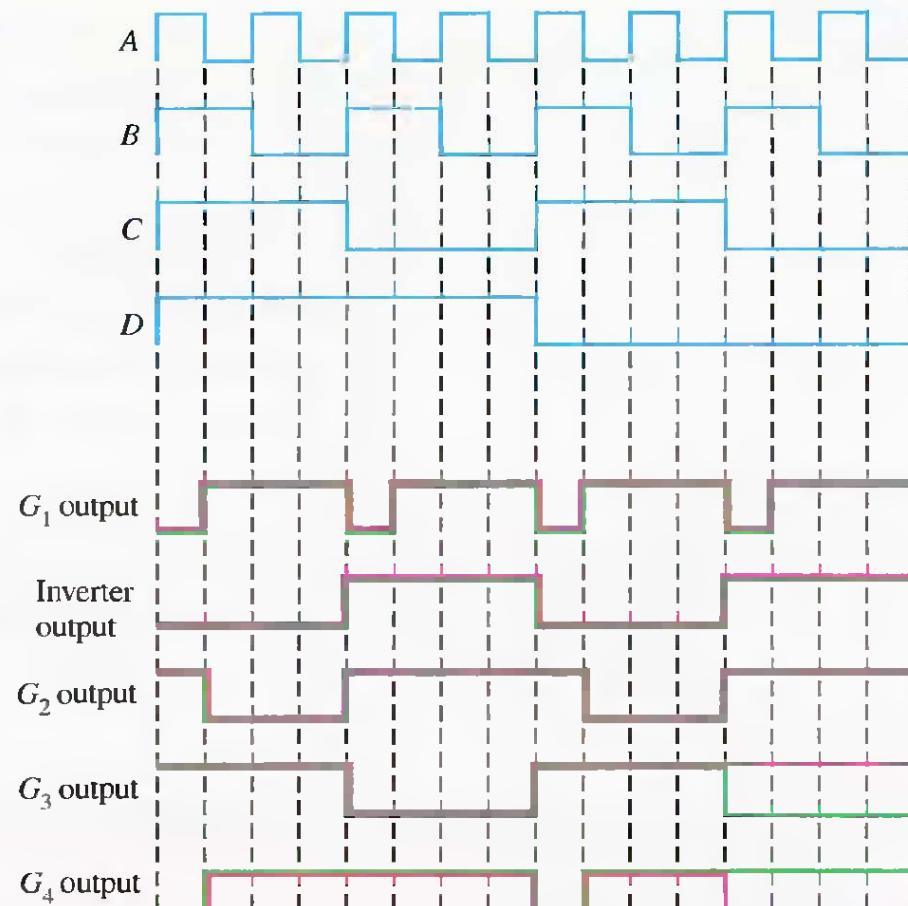
FIGURE 5-34

EXAMPLE 5-15

Determine the fault in the logic circuit of Figure 5-47(a) by using waveform analysis. You have observed the waveforms shown in green in Figure 5-47(b). The red waveforms are correct and are provided for comparison.



(a)



(b)

FIGURE 5-47