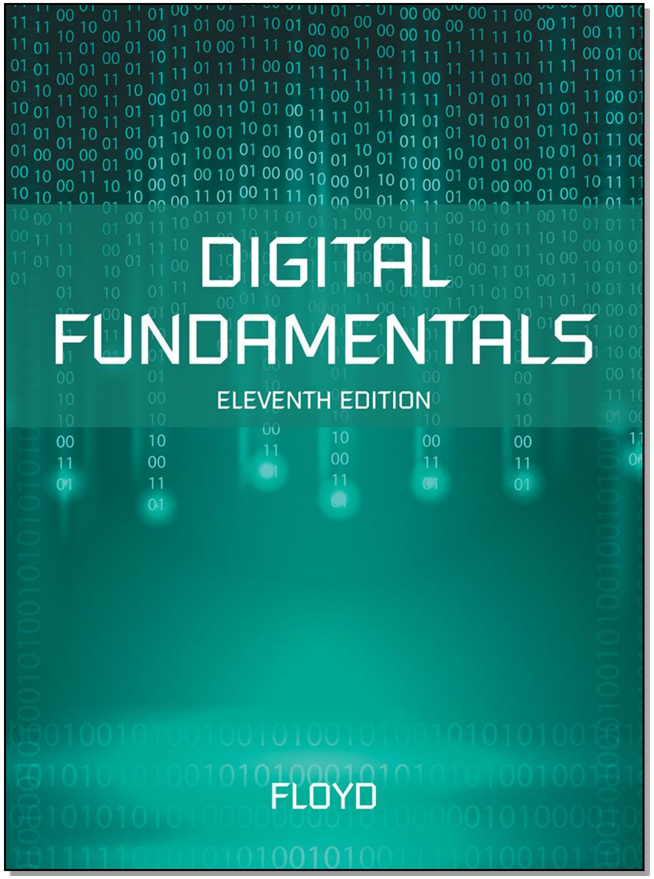
Digital Fundamentals

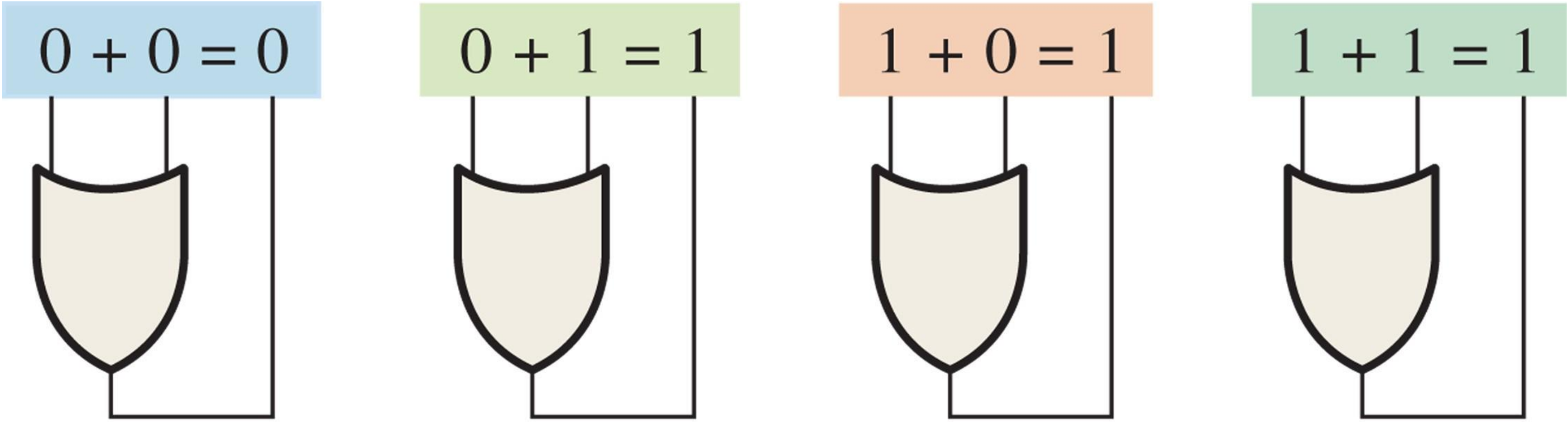
ELEVENTH EDITION



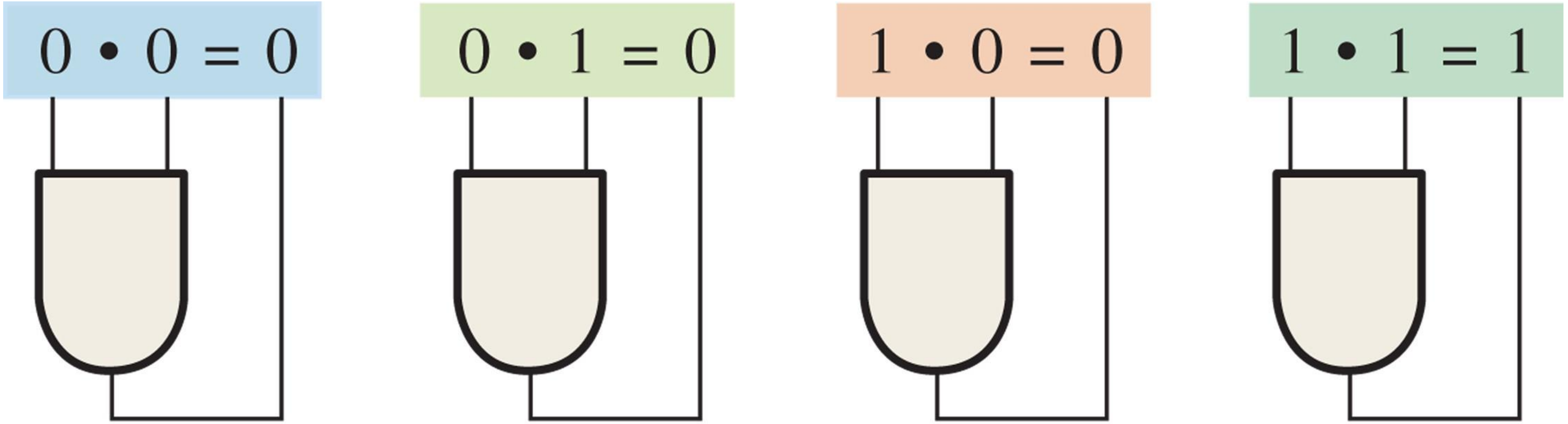
CHAPTER 4

Boolean Algebra and Logic Simplification

Boolean Addition



Boolean Multiplication



Determine the values of A, B, C, and D that make the sum term $A + \overline{B} + C + \overline{D}$ equal to 0.

Solution

For the sum term to be 0, each of the literals in the term must be 0. Therefore, $A = \mathbf{0}$, $B = \mathbf{1}$ so that $\overline{B} = 0$, $C = \mathbf{0}$, and $D = \mathbf{1}$ so that $\overline{D} = 0$.

$$A + \overline{B} + C + \overline{D} = 0 + \overline{1} + 0 + \overline{1} = 0 + 0 + 0 + 0 = 0$$

PEARSON

Determine the values of A, B, C, and D that make the product term $A\overline{B}C\overline{D}$ equal to 1.

Solution

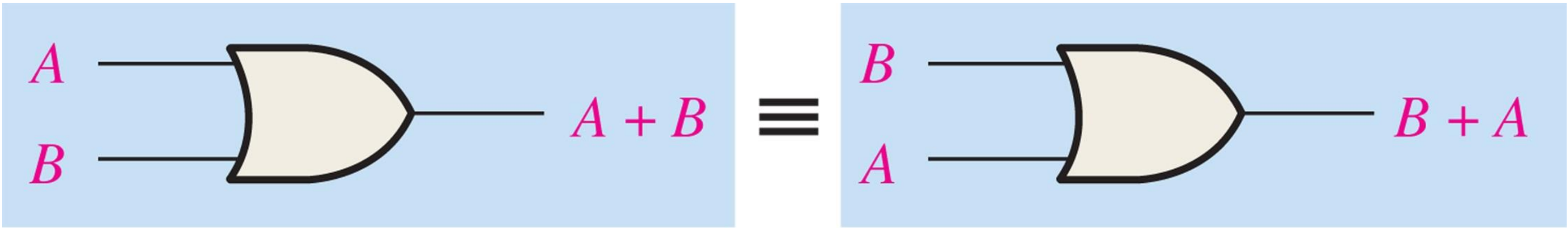
For the product term to be 1, each of the literals in the term must be 1. Therefore, $A = \mathbf{1}$, $B = \mathbf{0}$ so that $\overline{B} = 1$, $C = \mathbf{1}$, and $D = \mathbf{0}$ so that $\overline{D} = 1$.

$$A\overline{B}C\overline{D} = 1 \cdot \overline{0} \cdot 1 \cdot \overline{0} = 1 \cdot 1 \cdot 1 \cdot 1 = 1$$

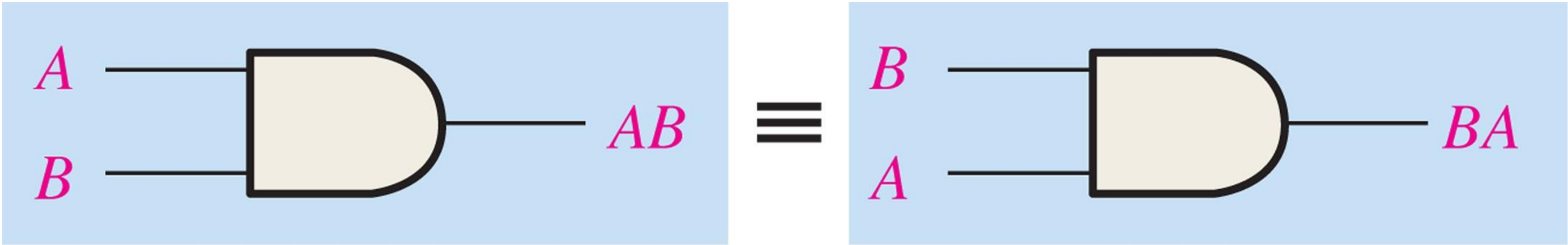
PEARSON

Laws of Boolean Algebra

Commutative law of addition



Commutative law of multiplication



Associative law of addition

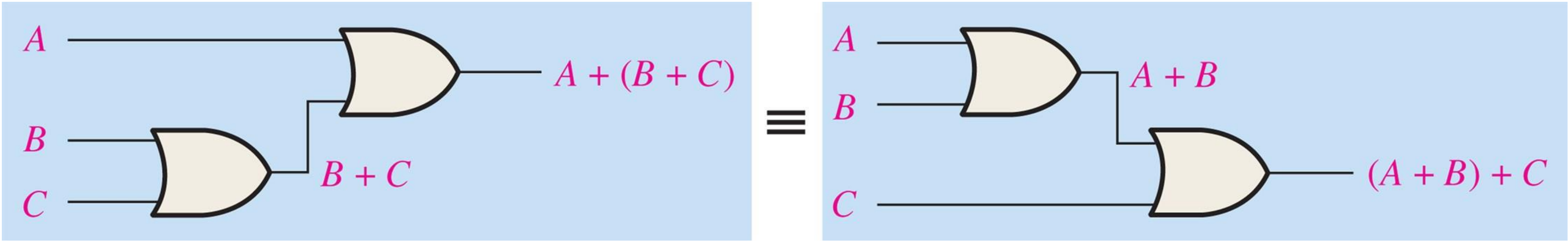
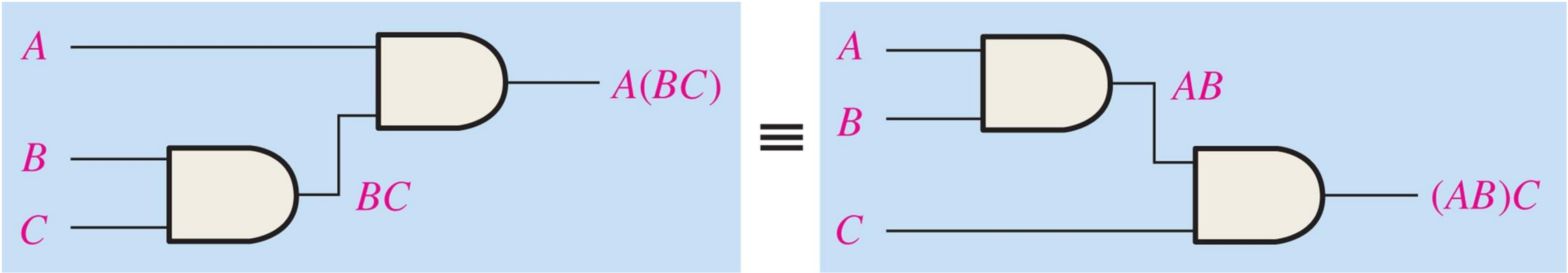
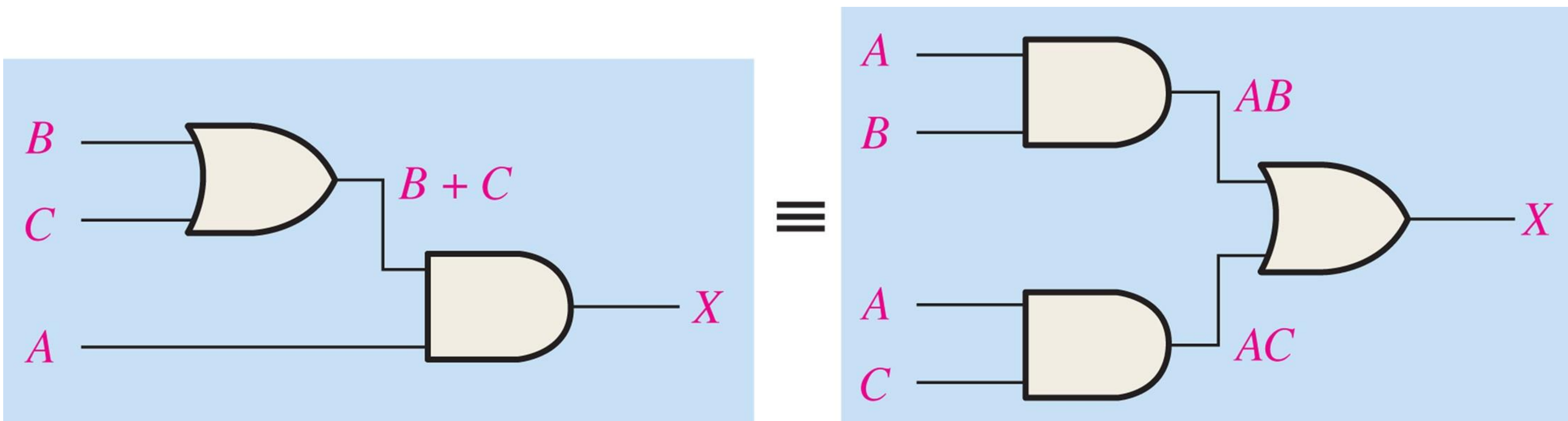


FIGURE 4-6 Application of associative law of multiplication.



Application of distributive law



$X = A(B + C)$ $\qquad$ $X = AB + AC$

TABLE 4–1

Basic rules of Boolean algebra.

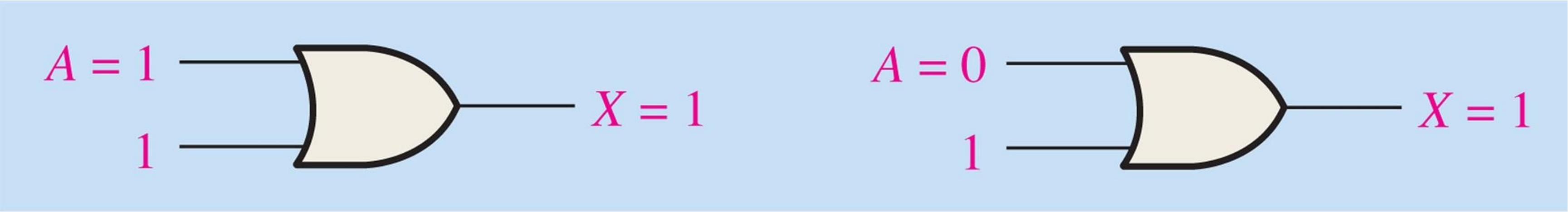
1. $A + 0 = A$	**7.** $A \cdot A = A$
2. $A + 1 = 1$	**8.** $A \cdot \overline{A} = 0$
3. $A \cdot 0 = 0$	**9.** $\overline{\overline{A}} = A$
4. $A \cdot 1 = A$	**10.** $A + AB = A$
5. $A + A = A$	**11.** $A + \overline{A}B = A + B$
6. $A + \overline{A} = 1$	**12.** $(A + B)(A + C) = A + BC$

A, B, or C can represent a single variable or a combination of variables.

Rule 1

$$X = A + 0 = A$$

Rule 2



$$X = A + 1 = 1$$

Rule 3

$$X = A \cdot 0 = 0$$

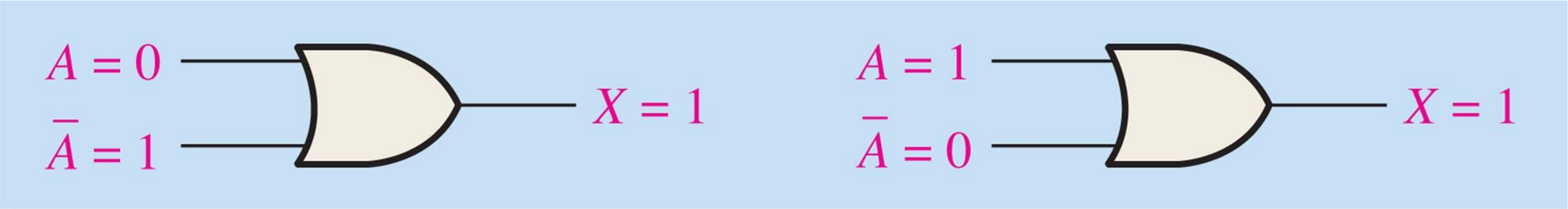
Rule 4

$$X = A \cdot 1 = A$$

Rule 5

$$X = A + A = A$$

Rule 6



$$X = A + \bar{A} = 1$$

PEARSON

Rule 7

$$X = A \cdot A = A$$

Rule 8

$$X = A \cdot \overline{A} = 0$$

Rule 9

$$\overline{\overline{A}} = A$$

PEARSON

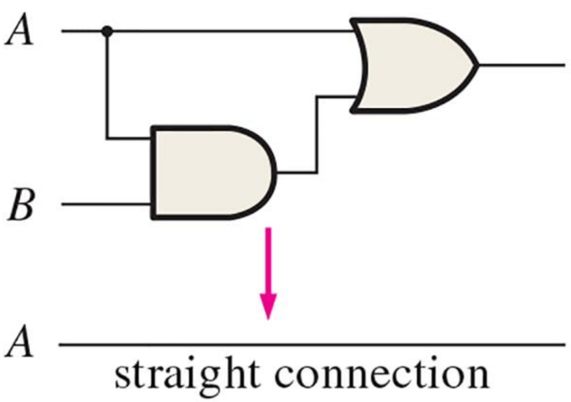
Rule 10

TABLE 4–2

Rule 10: $A + AB = A$. Open file T04-02 to verify.

A	B	AB	$A + AB$
0	0	0	0
0	1	0	0
1	0	0	1
1	1	1	1

equal



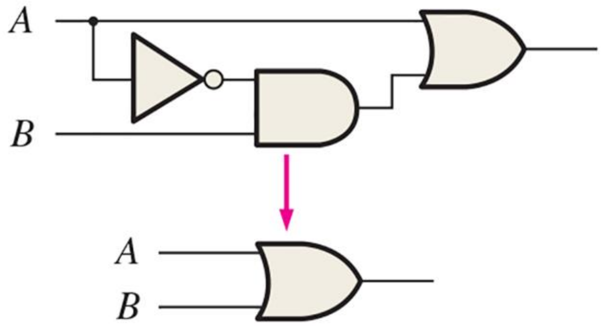
Rule 11

TABLE 4–3

Rule 11: $A + \overline{A}B = A + B$. Open file T04-03 to verify.

A	B	$\overline{A}B$	$A + \overline{A}B$	$A + B$
0	0	0	0	0
0	1	1	1	1
1	0	0	1	1
1	1	0	1	1

equal



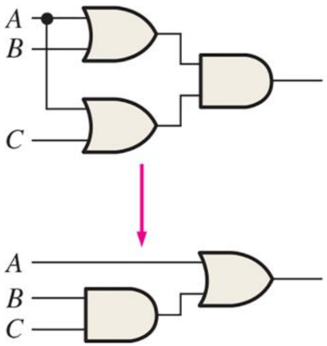
Rule 12

TABLE 4–4

Rule 12: $(A + B)(A + C) = A + BC$. Open file T04-04 to verify.

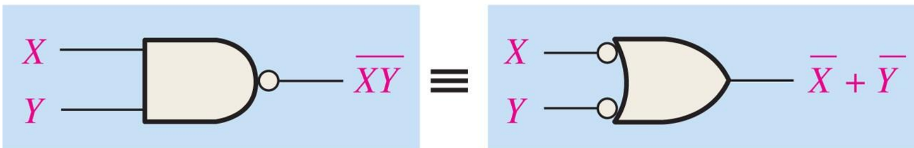
A	B	C	$A + B$	$A + C$	$(A + B)(A + C)$	BC	$A + BC$
0	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0
0	1	0	1	0	0	0	0
0	1	1	1	1	1	1	1
1	0	0	1	1	1	0	1
1	0	1	1	1	1	0	1
1	1	0	1	1	1	0	1
1	1	1	1	1	1	1	1

equal



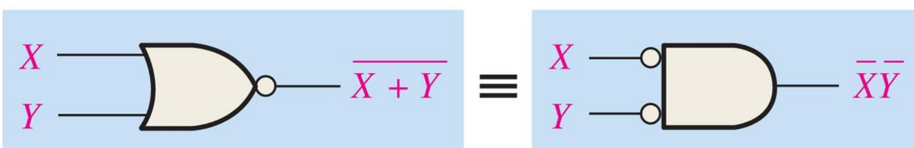
PEARSON

DeMorgan’s theorems



NAND | Negative-OR

Inputs		Output	
X	Y	$\overline{XY}$	$\overline{X} + \overline{Y}$
0	0	1	1
0	1	1	1
1	0	1	1
1	1	0	0



NOR | Negative-AND

Inputs		Output	
X	Y	$\overline{X + Y}$	$\overline{X}\,\overline{Y}$
0	0	1	1
0	1	0	0
1	0	0	0
1	1	0	0

Apply DeMorgan's theorems to the expressions $\overline{XYZ}$ and $\overline{X + Y + Z}$.

Solution

$$\overline{XYZ} = \overline{X} + \overline{Y} + \overline{Z}$$
$$\overline{X + Y + Z} = \overline{X}\,\overline{Y}\,\overline{Z}$$

Apply DeMorgan's theorems to the expressions $\overline{WXYZ}$ and $\overline{W + X + Y + Z}$.

Solution

$$\overline{WXYZ} = \overline{W} + \overline{X} + \overline{Y} + \overline{Z}$$
$$\overline{W + X + Y + Z} = \overline{W}\,\overline{X}\,\overline{Y}\,\overline{Z}$$

PEARSON

Apply DeMorgan's theorems to each of the following expressions:

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

(b) $\overline{ABC + DEF}$

(c) $\overline{A\overline{B} + \overline{C}D + EF}$

(a) Let $A + B + C = X$ and $D = Y$. The expression $\overline{(A + B + C)D}$ is of the form $\overline{XY} = \overline{X} + \overline{Y}$ and can be rewritten as

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

Next, apply DeMorgan's theorem to the term $\overline{A + B + C}$.

$$\overline{A + B + C} + \overline{D} = \overline{A}\,\overline{B}\,\overline{C} + \overline{D}$$

PEARSON

(b) Let $ABC = X$ and $DEF = Y$. The expression $\overline{ABC + DEF}$ is of the form $\overline{X + Y} = \overline{X}\,\overline{Y}$ and can be rewritten as

$$\overline{ABC + DEF} = (\overline{ABC})(\overline{DEF})$$

Next, apply DeMorgan's theorem to each of the terms $\overline{ABC}$ and $\overline{DEF}$.

$$(\overline{ABC})(\overline{DEF}) = (\overline{A} + \overline{B} + \overline{C})(\overline{D} + \overline{E} + \overline{F})$$

(c) Let $A\overline{B} = X$, $\overline{C}D = Y$, and $EF = Z$. The expression $\overline{A\overline{B} + \overline{C}D + EF}$ is of the form $\overline{X + Y + Z} = \overline{X}\,\overline{Y}\,\overline{Z}$ and can be rewritten as

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

Next, apply DeMorgan's theorem to each of the terms $\overline{A\overline{B}}$, $\overline{\overline{C}D}$, and $\overline{EF}$.

$$(\overline{A\overline{B}})(\overline{\overline{C}D})(\overline{EF}) = (\overline{A} + B)(C + \overline{D})(\overline{E} + \overline{F})$$

Boolean Analysis of Logic Circuits

A combinational logic circuit showing the development of the Boolean expression for the output

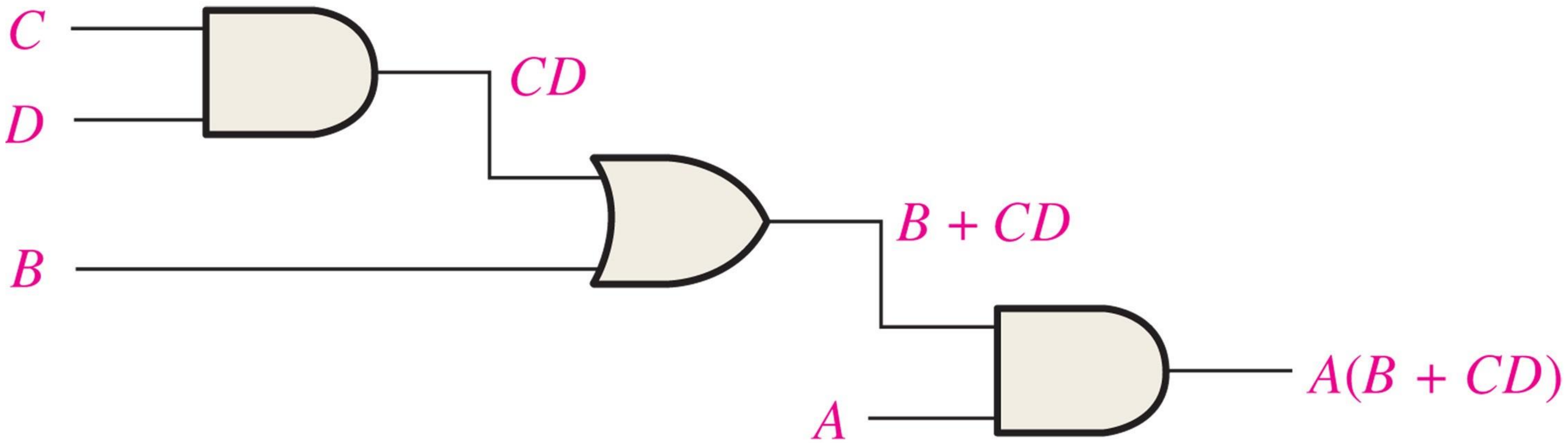


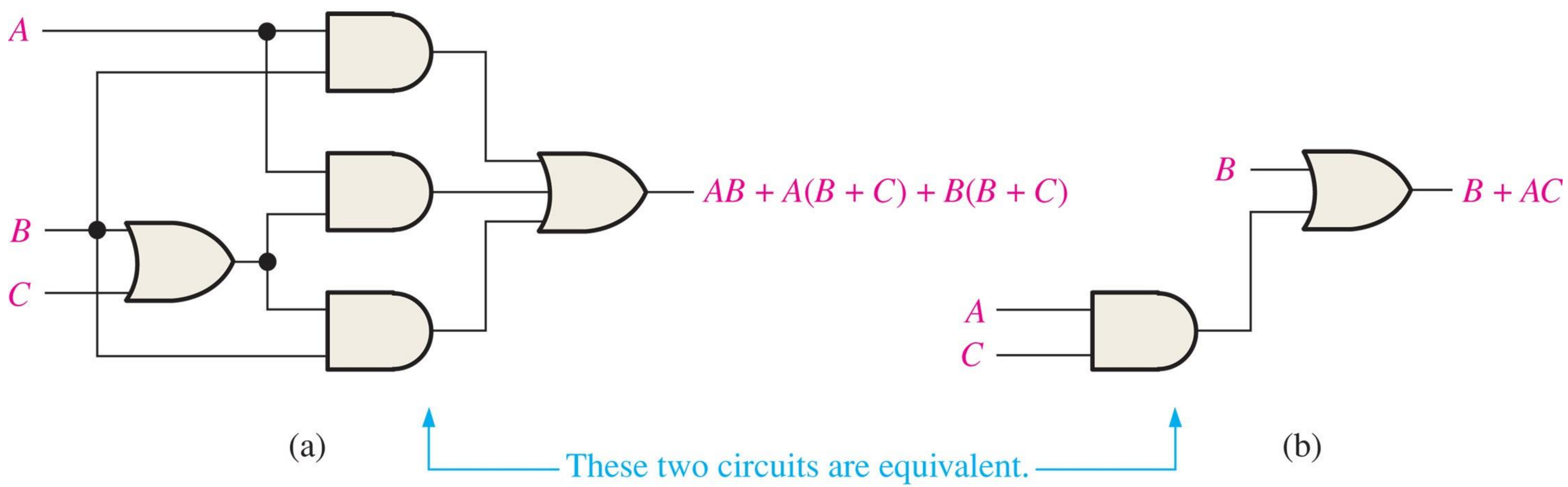
TABLE 4–5

Truth table for the logic circuit in Figure 4–18.

Inputs				Output
A	B	C	D	$A(B + CD)$
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

Using Boolean algebra techniques, simplify this expression:

$$AB + A(B + C) + B(B + C)$$



(a) (b)

Solution

The following is not necessarily the only approach.

Step 1: Apply the distributive law to the second and third terms in the expression, as follows:

$$AB + AB + AC + BB + BC$$

Step 2: Apply rule 7 ($BB = B$) to the fourth term.

$$AB + AB + AC + B + BC$$

Step 3: Apply rule 5 ($AB + AB = AB$) to the first two terms.

$$AB + AC + B + BC$$

Step 4: Apply rule 10 ($B + BC = B$) to the last two terms.

$$AB + AC + B$$

Step 5: Apply rule 10 ($AB + B = B$) to the first and third terms.

$$B + AC$$

Simplify the following Boolean expression:

$$\overline{\overline{AB} + AC} + \overline{A}\,\overline{B}C$$

Solution

Step 1: Apply DeMorgan's theorem to the first term.

$$(\overline{\overline{AB}})(\overline{AC}) + \overline{A}\,\overline{B}C$$

Step 2: Apply DeMorgan's theorem to each term in parentheses.

$$(\overline{A} + \overline{B})(\overline{A} + \overline{C}) + \overline{A}\,\overline{B}C$$

Step 3: Apply the distributive law to the two terms in parentheses.

$$\overline{A}\,\overline{A} + \overline{A}\,\overline{C} + \overline{A}\,\overline{B} + \overline{B}\,\overline{C} + \overline{A}\,\overline{B}C$$

Step 4: Apply rule 7 ($\overline{A}\,\overline{A} = \overline{A}$) to the first term, and apply rule 10 [$\overline{A}\,\overline{B} + \overline{A}\,\overline{B}C = \overline{A}\,\overline{B}(1 + C) = \overline{A}\,\overline{B}$] to the third and last terms.

$$\overline{A} + \overline{A}\,\overline{C} + \overline{A}\,\overline{B} + \overline{B}\,\overline{C}$$

Step 5: Apply rule 10 [$\overline{A} + \overline{A}\,\overline{C} = \overline{A}(1 + \overline{C}) = \overline{A}$] to the first and second terms.

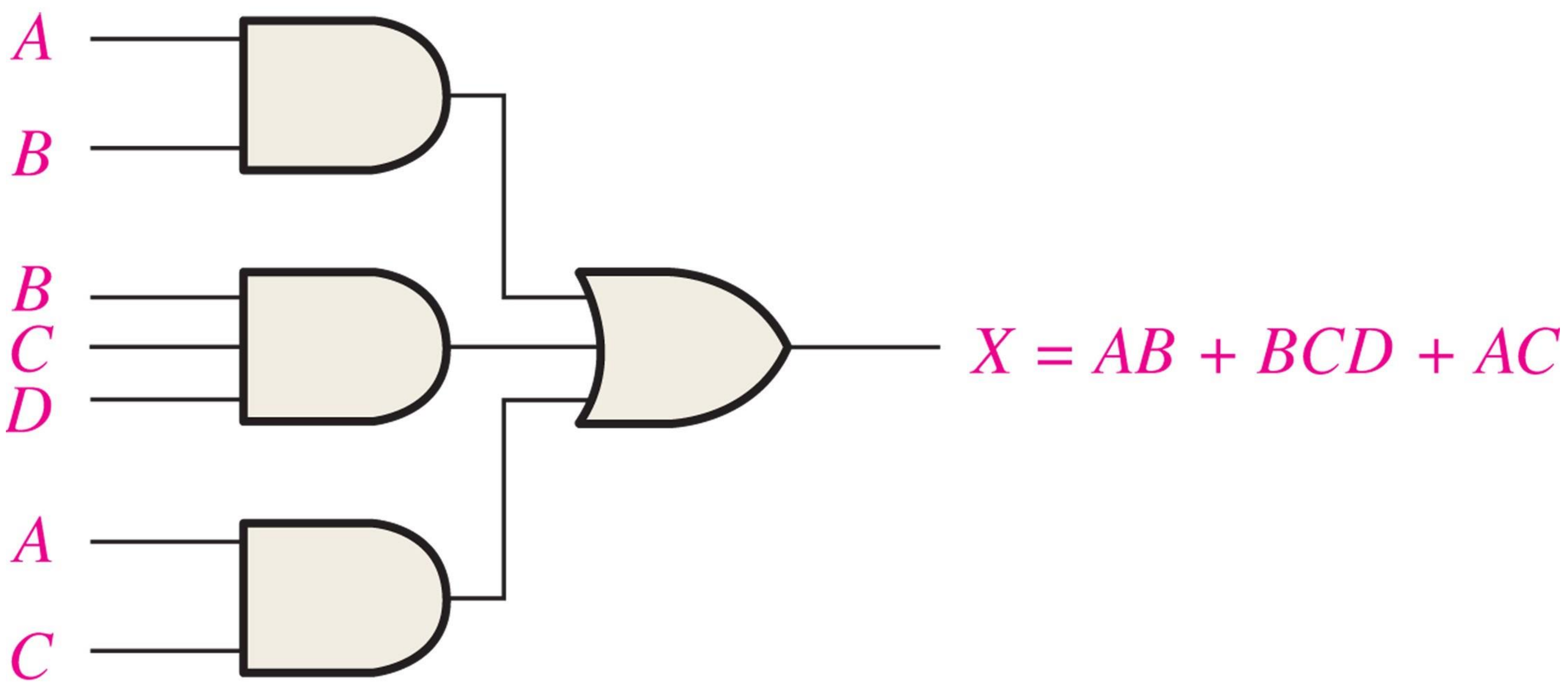
$$\overline{A} + \overline{A}\,\overline{B} + \overline{B}\,\overline{C}$$

Step 6: Apply rule 10 [$\overline{A} + \overline{A}\,\overline{B} = \overline{A}(1 + \overline{B}) = \overline{A}$] to the first and second terms.

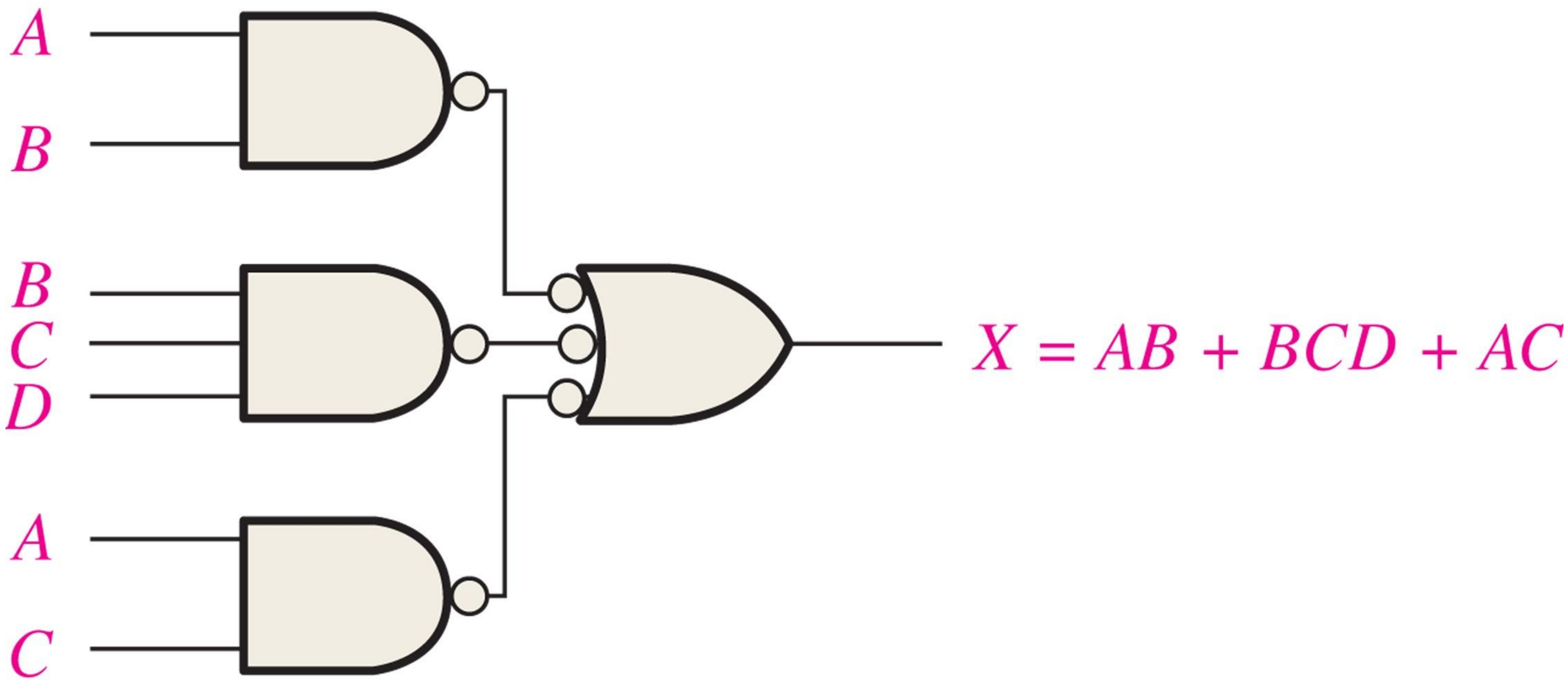
$$\overline{A} + \overline{B}\,\overline{C}$$

Implementation of the SOP expression *AB* + *BCD* + *AC*

AND/OR Implementation of an SOP Expression



This NAND/NAND implementation



Standard Forms of Boolean Expressions

Conversion of a General Expression to SOP Form

Convert the following Boolean expression into standard SOP form:

$$A\overline{B}C + \overline{A}\,\overline{B} + AB\overline{C}D$$

Solution

The domain of this SOP expression is A, B, C, D. Take one term at a time. The first term, $A\overline{B}C$, is missing variable D or $\overline{D}$, so multiply the first term by $D + \overline{D}$ as follows:

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

In this case, two standard product terms are the result.

The second term, $\overline{A}\,\overline{B}$, is missing variables C or $\overline{C}$ and D or $\overline{D}$, so first multiply the second term by $C + \overline{C}$ as follows:

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

The two resulting terms are missing variable D or $\overline{D}$, so multiply both terms by $D + \overline{D}$ as follows:

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

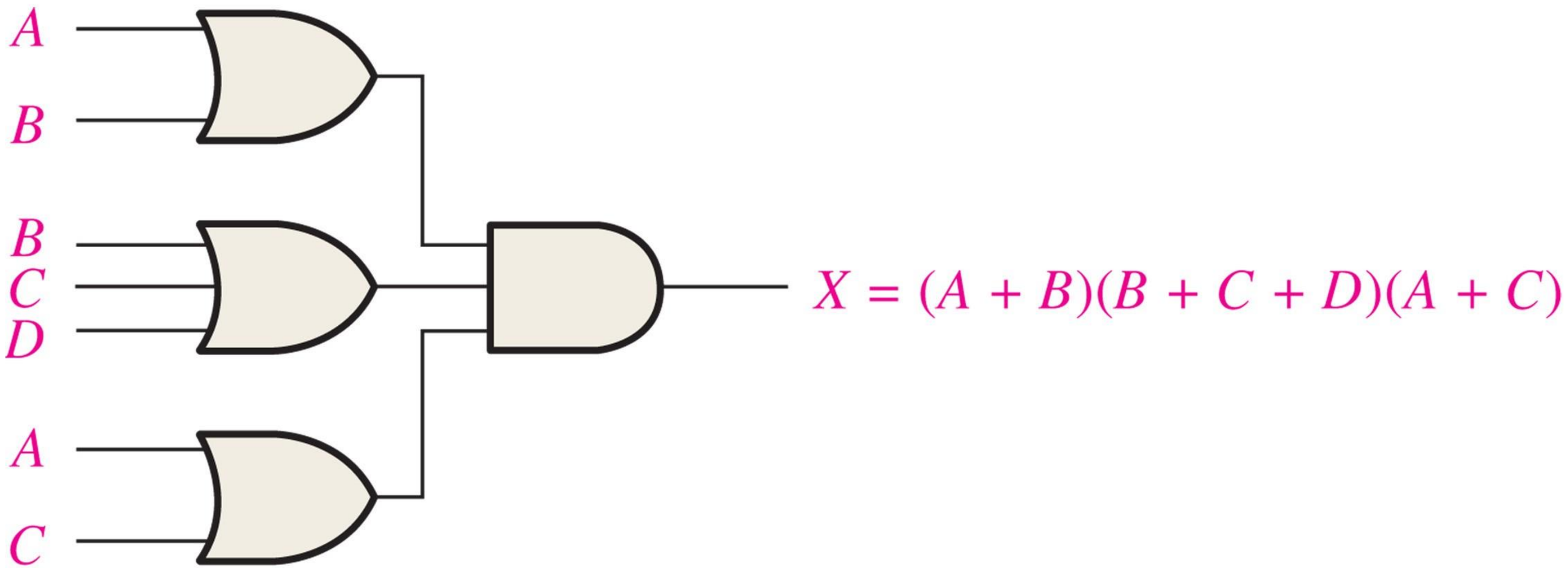
In this case, four standard product terms are the result.

The third term, $AB\overline{C}D$, is already in standard form. The complete standard SOP form of the original expression is as follows:

$$A\overline{B}C + \overline{A}\,\overline{B} + AB\overline{C}D = A\overline{B}CD + A\overline{B}C\overline{D} + \overline{A}\,\overline{B}CD + \overline{A}\,\overline{B}C\overline{D} + \overline{A}\,\overline{B}\,\overline{C}D + \overline{A}\,\overline{B}\,\overline{C}\,\overline{D} + AB\overline{C}D$$

The Product-of-Sums (POS) Form

Implementation of the POS expression $(A + B)(B + C + D)(A + C)$



The Standard POS Form

Convert the following Boolean expression into standard POS form:

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

Solution

The domain of this POS expression is A, B, C, D. Take one term at a time. The first term, $A + \overline{B} + C$, is missing variable D or $\overline{D}$, so add $D\overline{D}$ and apply rule 12 as follows:

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

The second term, $\overline{B} + C + \overline{D}$, is missing variable A or $\overline{A}$, so add $A\overline{A}$ and apply rule 12 as follows:

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

The third term, $A + \overline{B} + \overline{C} + D$, is already in standard form. The standard POS form of the original expression is as follows:

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

Boolean Expressions and Truth Tables

TABLE 4–6

Inputs			Output	
A	B	C	X	Product Term
0	0	0	0	
0	0	1	1	$\overline{A}\,\overline{B}C$
0	1	0	0	
0	1	1	0	
1	0	0	1	$A\overline{B}\,\overline{C}$
1	0	1	0	
1	1	0	0	
1	1	1	1	ABC

TABLE 4–7

Inputs			Output	
A	B	C	X	Sum Term
0	0	0	0	$(A + B + C)$
0	0	1	1	
0	1	0	0	$(A + \overline{B} + C)$
0	1	1	0	$(A + \overline{B} + \overline{C})$
1	0	0	1	
1	0	1	0	$(\overline{A} + B + \overline{C})$
1	1	0	0	$(\overline{A} + \overline{B} + C)$
1	1	1	1	

TABLE 4–8

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

The Karnaugh Map

FIGURE 4-25 A 3-variable Karnaugh map showing Boolean product terms for each cell.

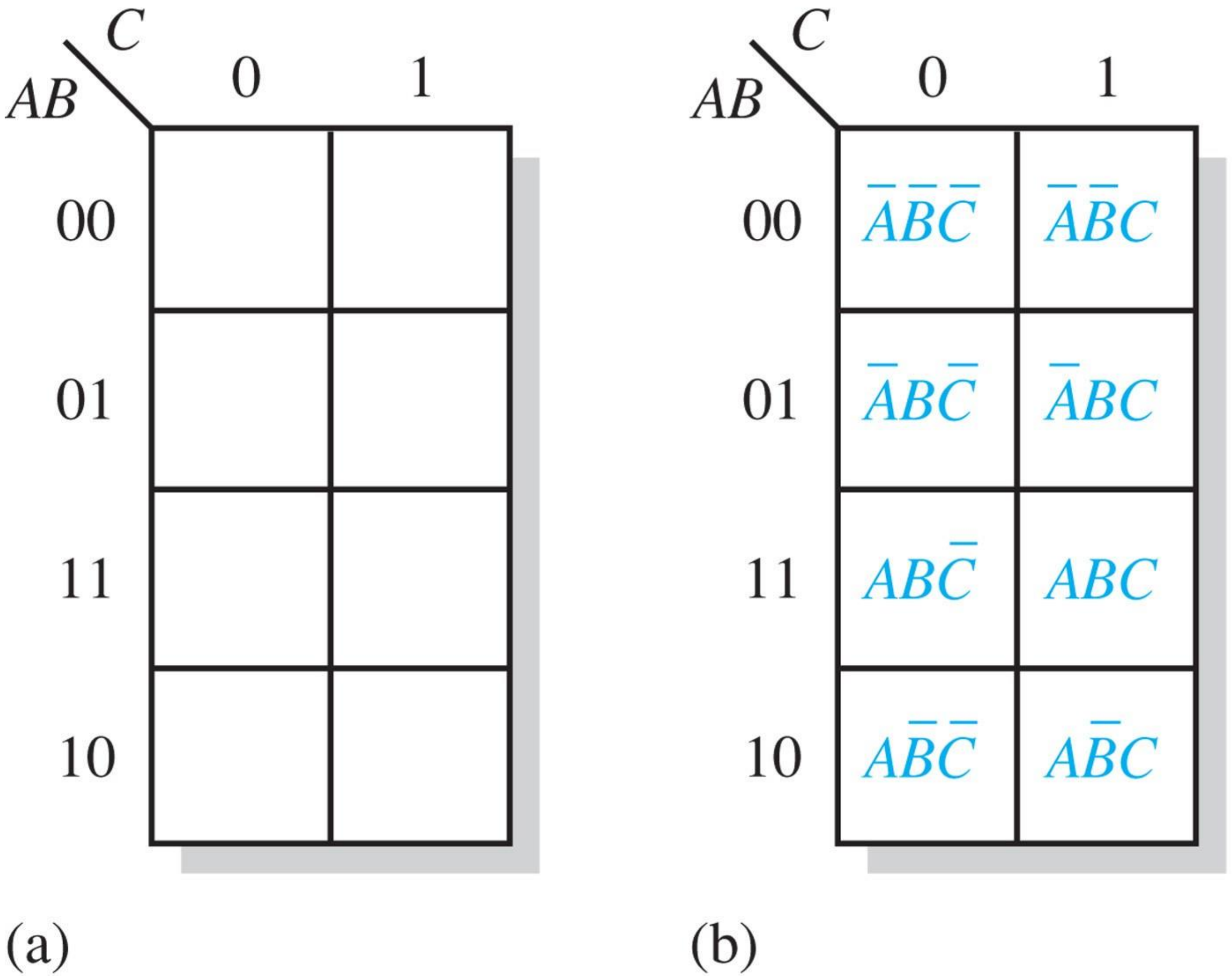
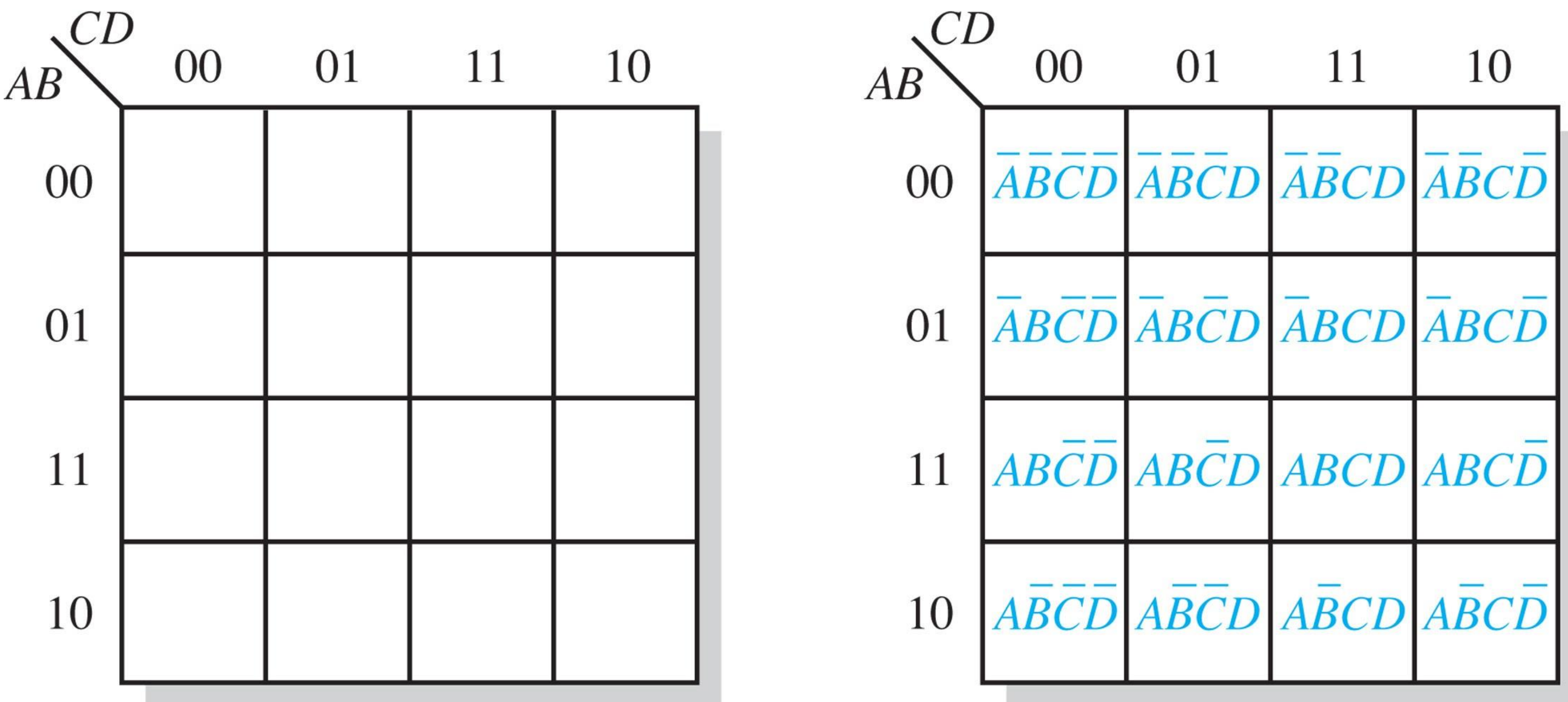


FIGURE 4-26 A 4-variable Karnaugh map.



(a) (b)

FIGURE 4–27 Adjacent cells on a Karnaugh map are those that differ by only one variable. Arrows point between adjacent cells.

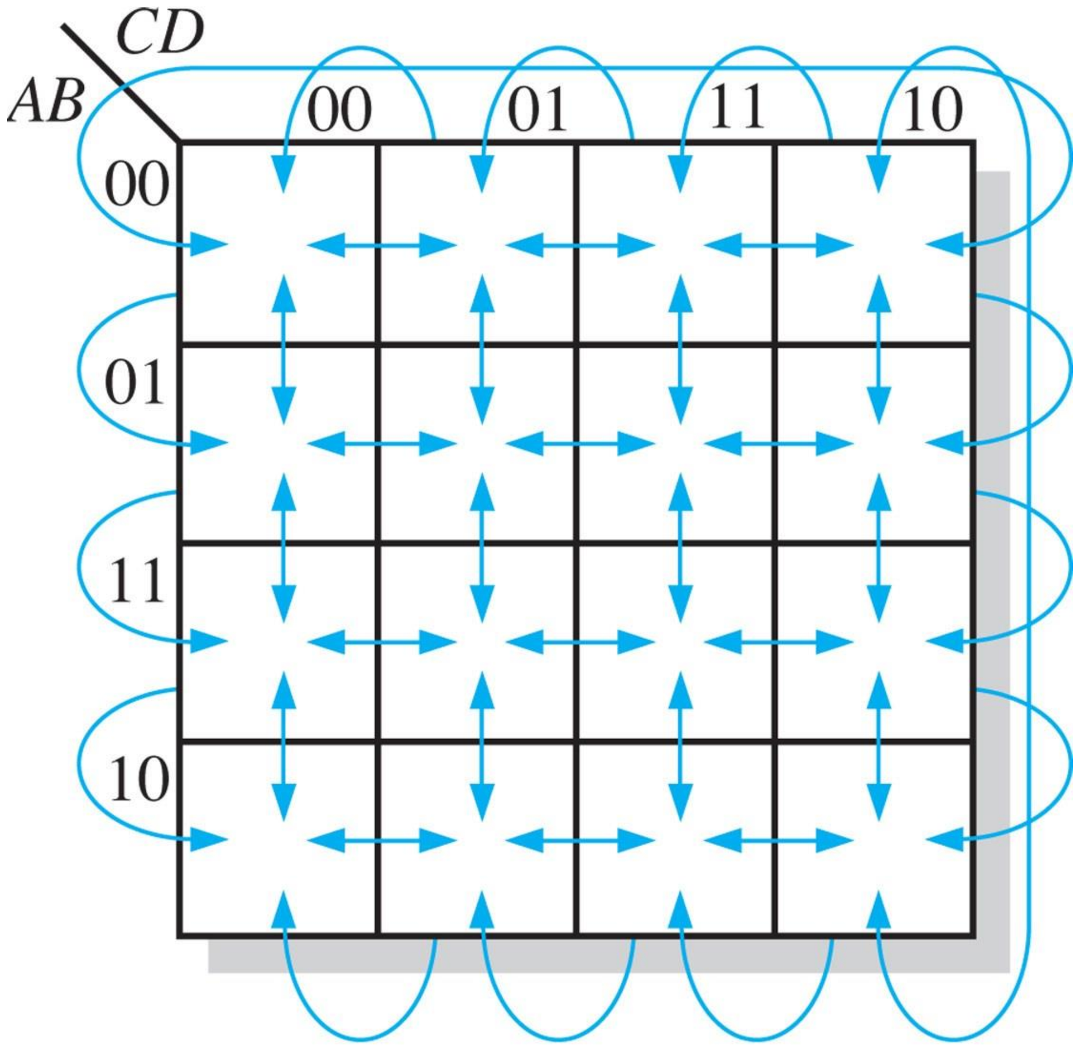


FIGURE 4-28 Example of mapping a standard SOP expression.

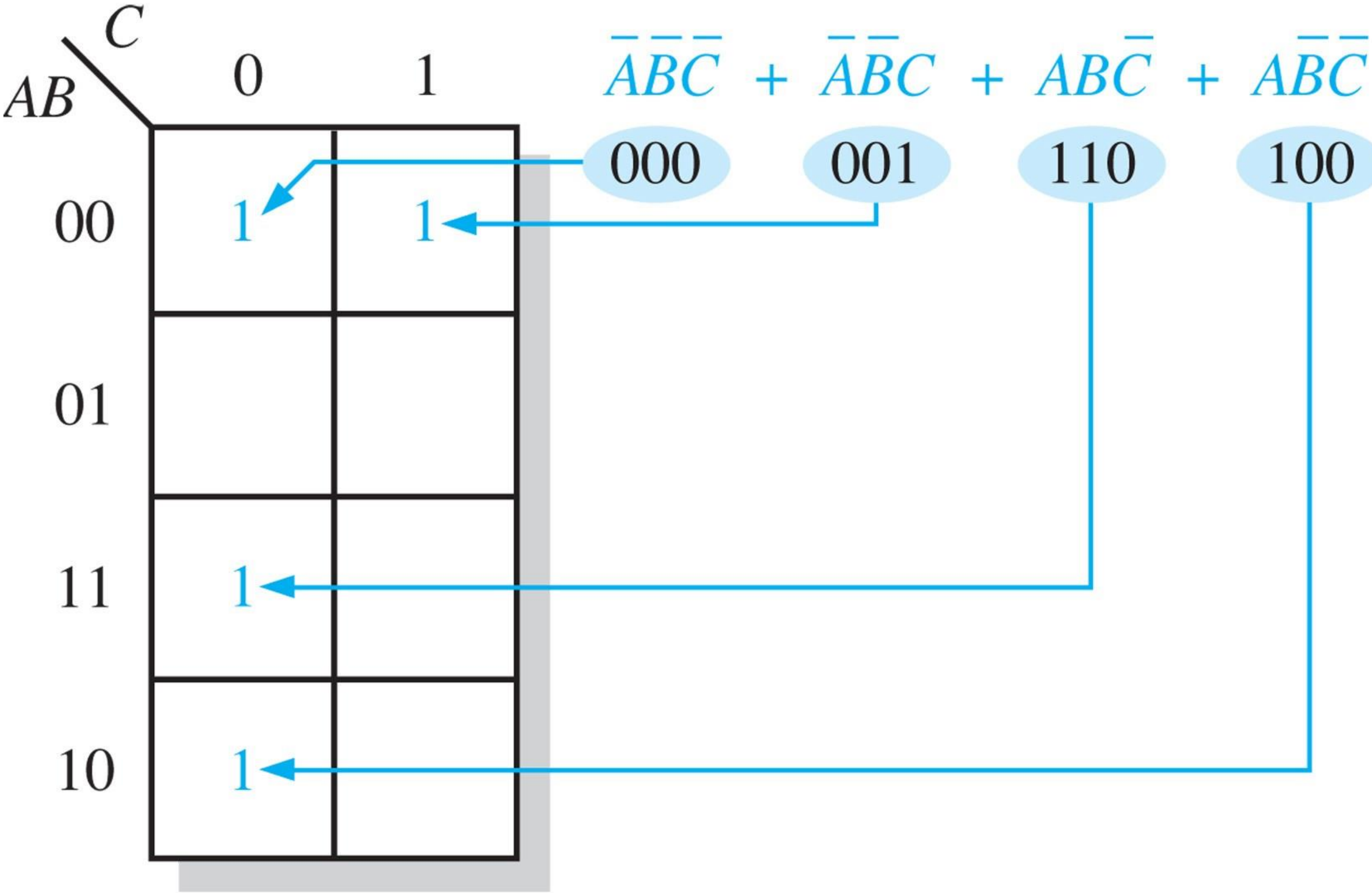
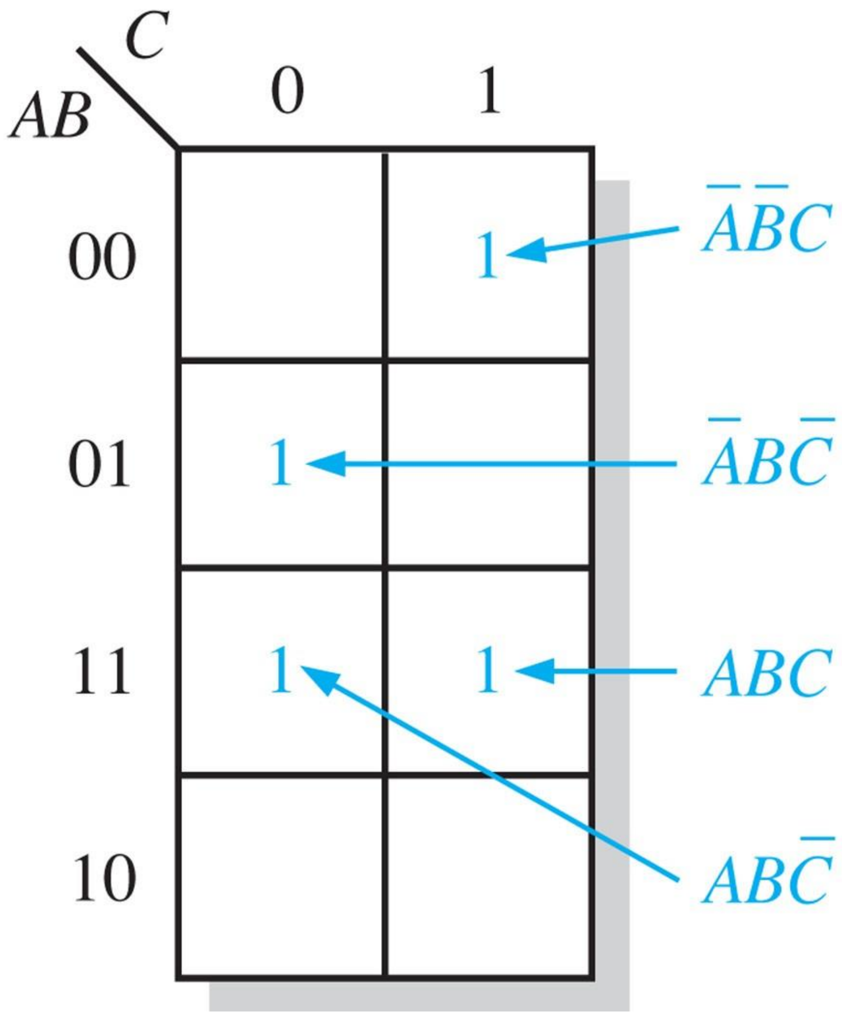
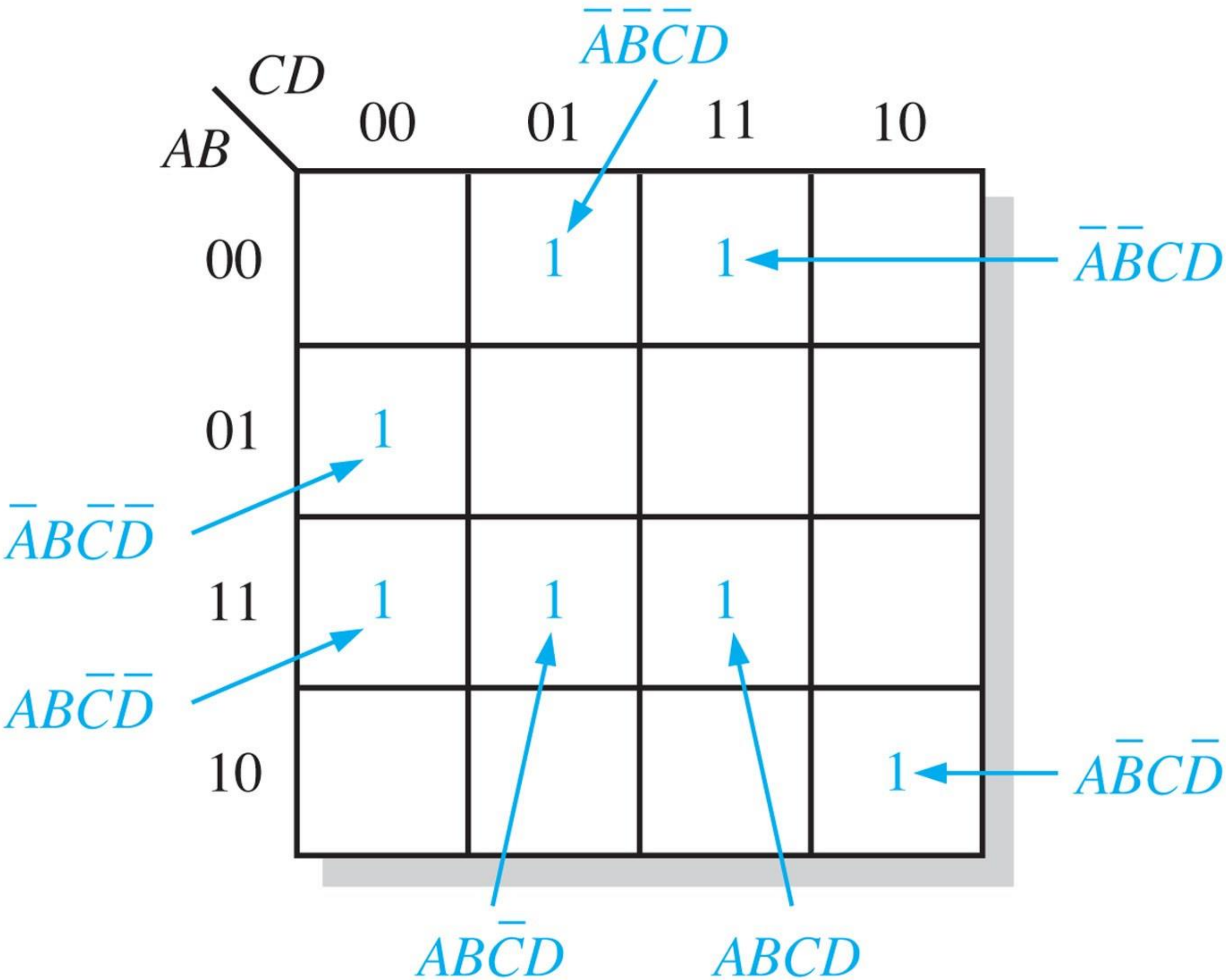


FIGURE 4-29



C
AB
0
1
00
01
11
10
1
1
1
1
$\overline{A}\overline{B}C$
$\overline{A}B\overline{C}$
ABC
$AB\overline{C}$

FIGURE 4-30



CD
AB
00
01
11
10
$\overline{A}\,\overline{B}\,\overline{C}D$
$\overline{A}\,\overline{B}CD$
$\overline{A}B\overline{C}\,\overline{D}$
$AB\overline{C}\,\overline{D}$
$A\overline{B}C\overline{D}$
$AB\overline{C}D$
$ABCD$
1
1
1
1
1
1
1

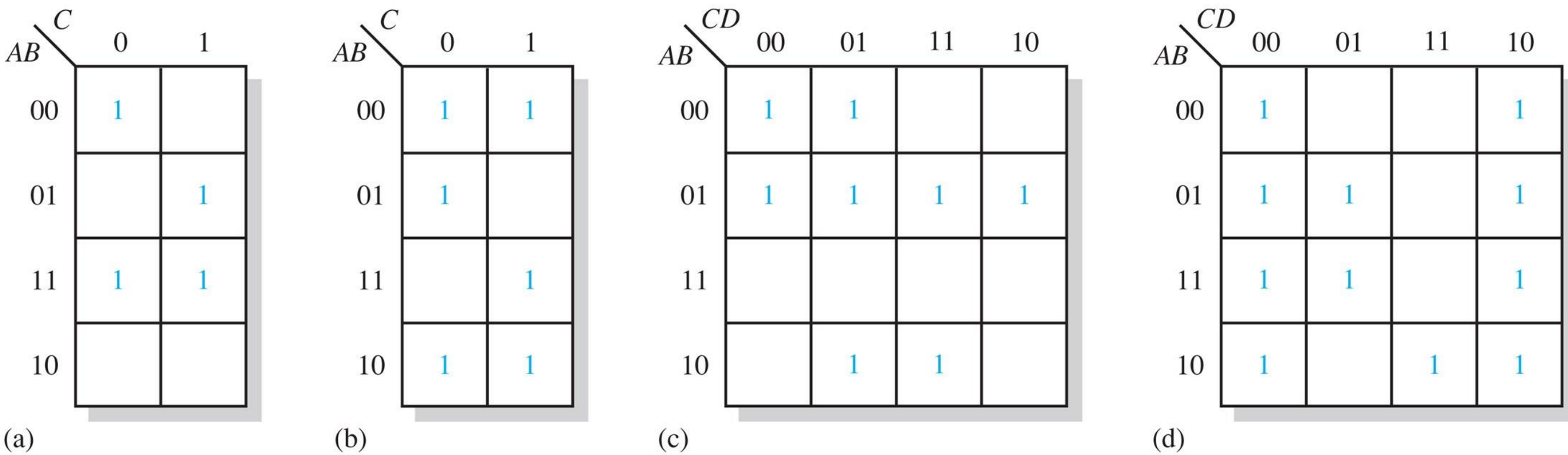
FIGURE 4-31

AB \ C	0	1
00	1	1
01	1	1
11	1	
10	1	1

FIGURE 4-32

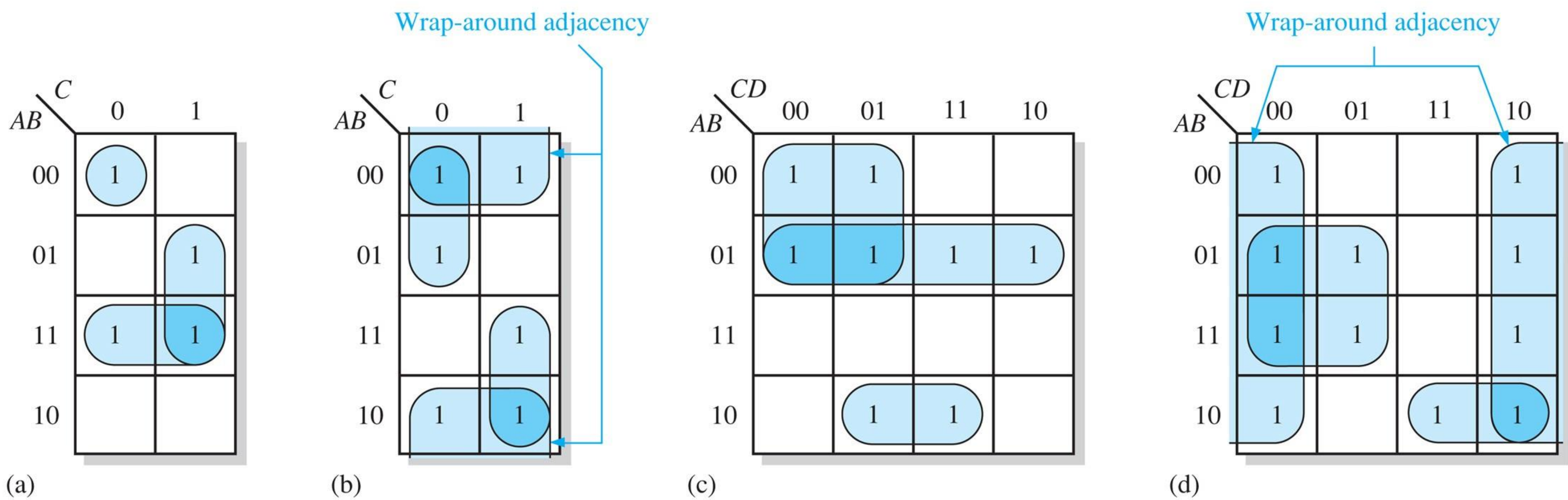
AB \ CD	00	01	11	10
00	1	1		
01				
11	1	1		
10	1	1	1	1

FIGURE 4-33



C
AB
0
1
00
01
11
10
1
1
1
1
(a)
C
AB
0
1
00
01
11
10
1
1
1
1
1
1
(b)
CD
AB
00
01
11
10
00
01
11
10
1
1
1
1
1
1
1
1
(c)
CD
AB
00
01
11
10
00
01
11
10
1
1
1
1
1
1
1
1
1
1
1
(d)

FIGURE 4-34



Wrap-around adjacency
Wrap-around adjacency
AB
C
0
1
00
01
11
10
CD
00
01
11
10
1
(a)
(b)
(c)
(d)

FIGURE 4-35

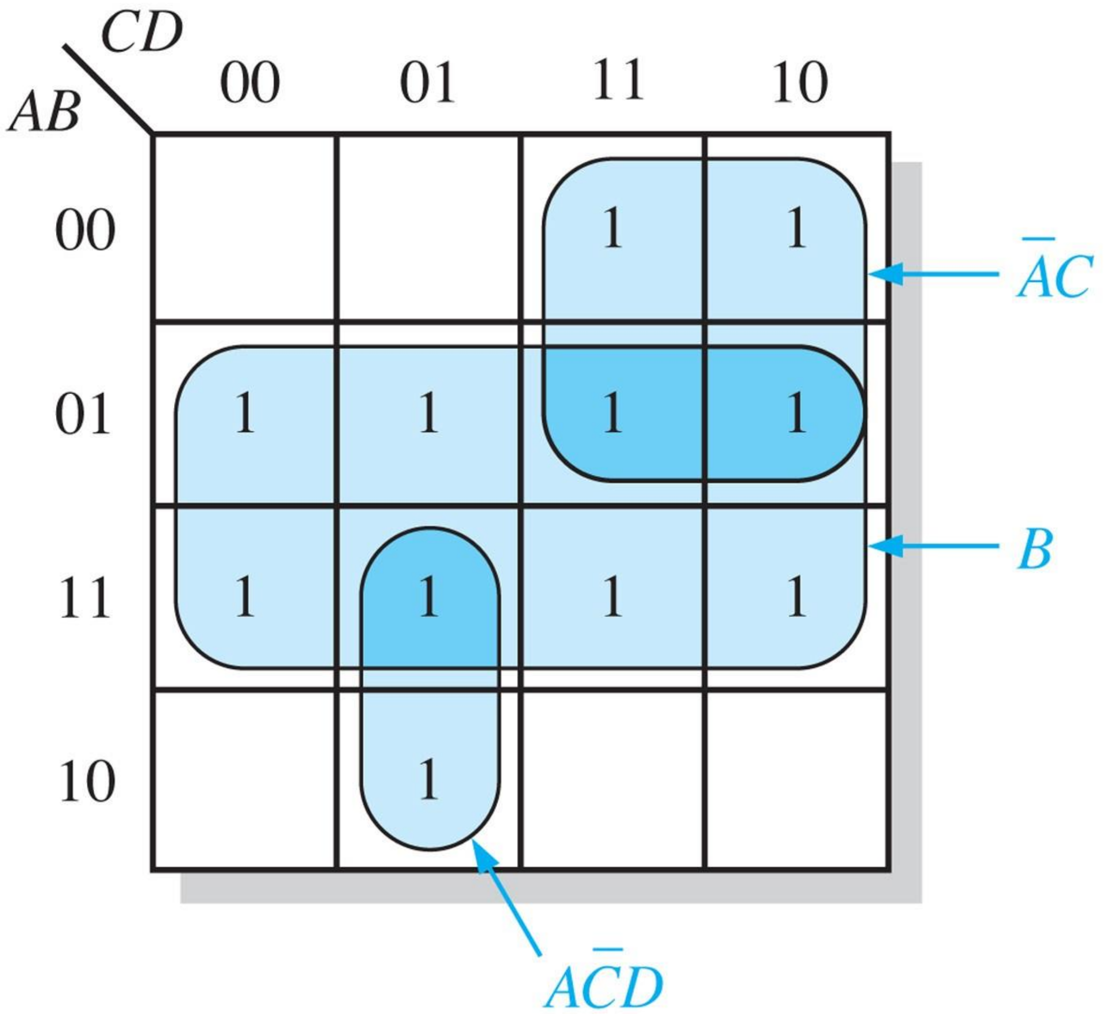
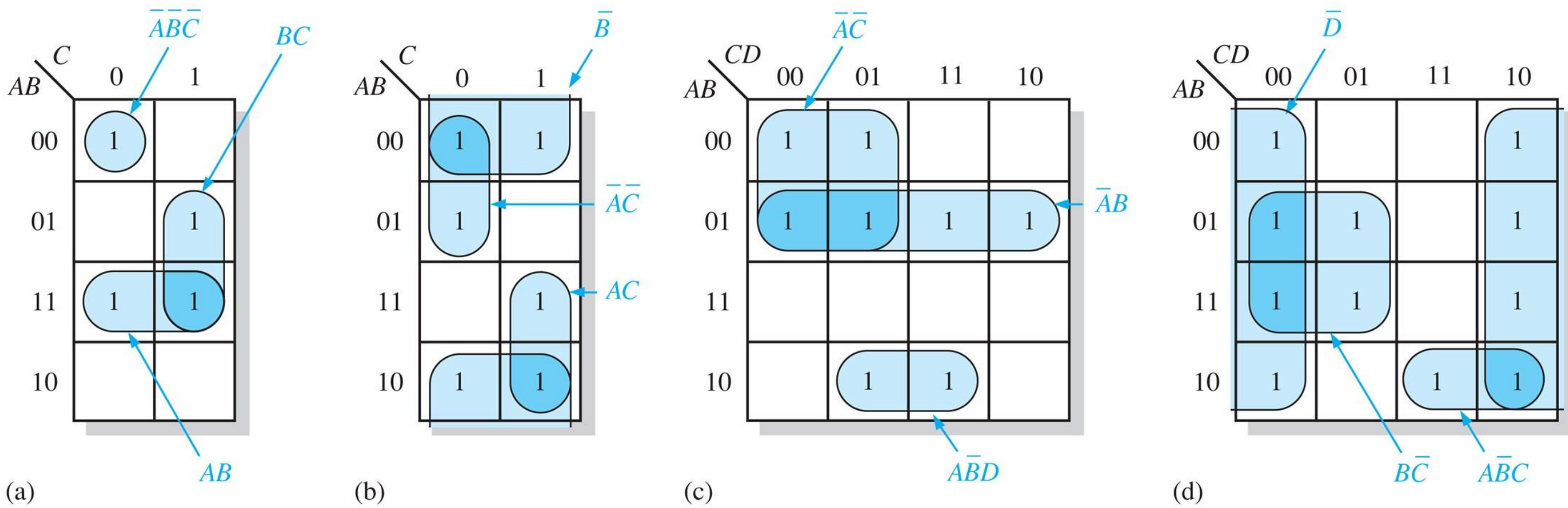


FIGURE 4-36



$\overline{A}\overline{B}\overline{C}$
BC
AB
(a)
$\overline{B}$
$\overline{A}\overline{C}$
AC
(b)
$\overline{A}\overline{C}$
$\overline{A}B$
$A\overline{B}D$
(c)
$\overline{D}$
$B\overline{C}$
$A\overline{B}C$
(d)

FIGURE 4-37

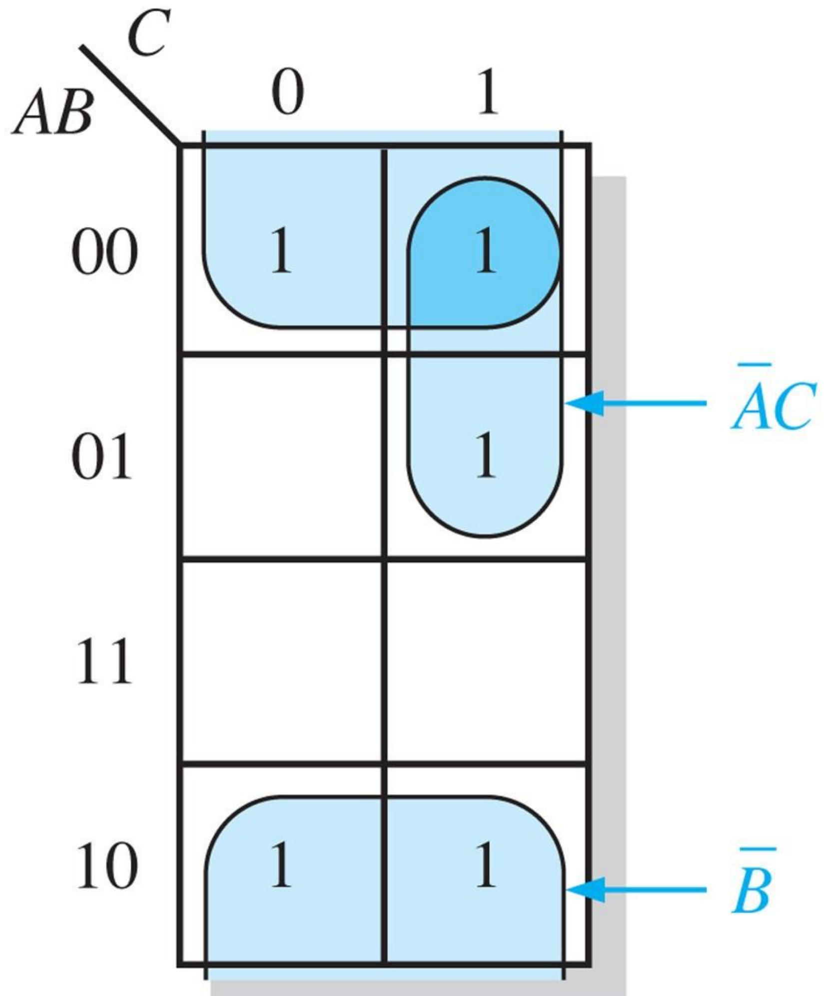
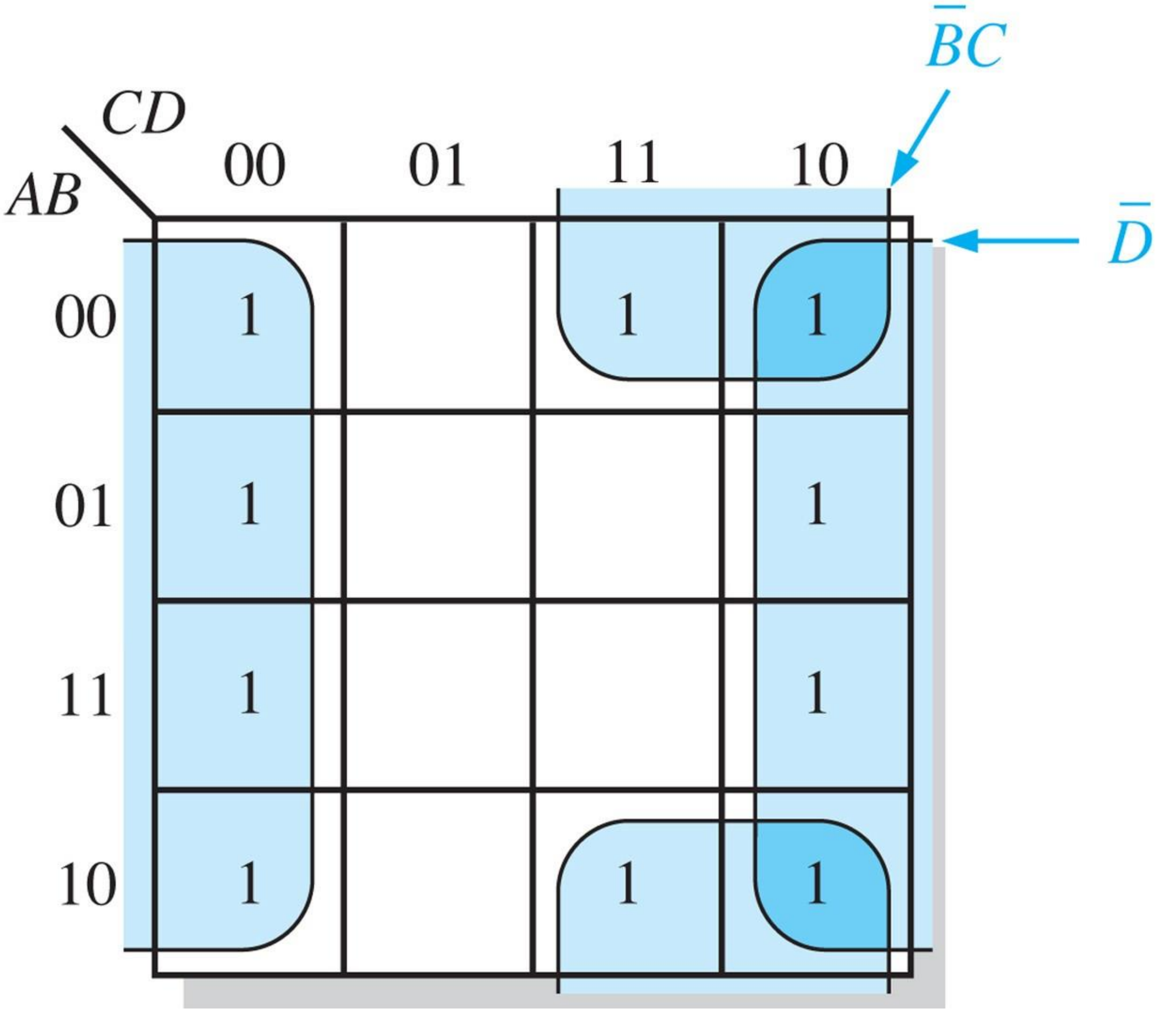


FIGURE 4-38



$\overline{B}C$
CD
AB
00
01
11
10
$\overline{D}$
00
01
11
10
1
1
1
1
1
1
1
1
1
1
1
1

FIGURE 4-39 Example of mapping directly from a truth table to a Karnaugh map.

$$X = \overline{A}\,\overline{B}\,\overline{C} + A\overline{B}\,\overline{C} + AB\overline{C} + ABC$$

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

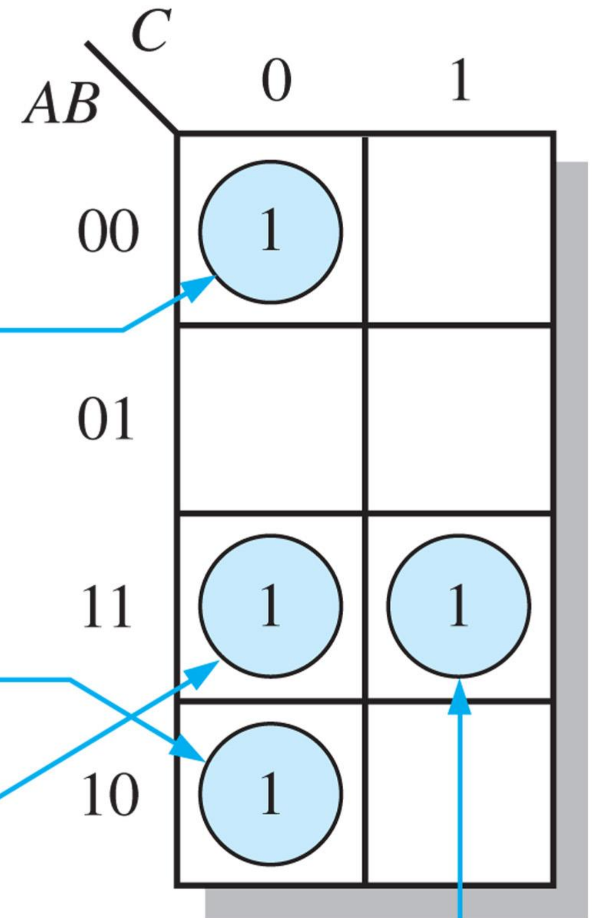
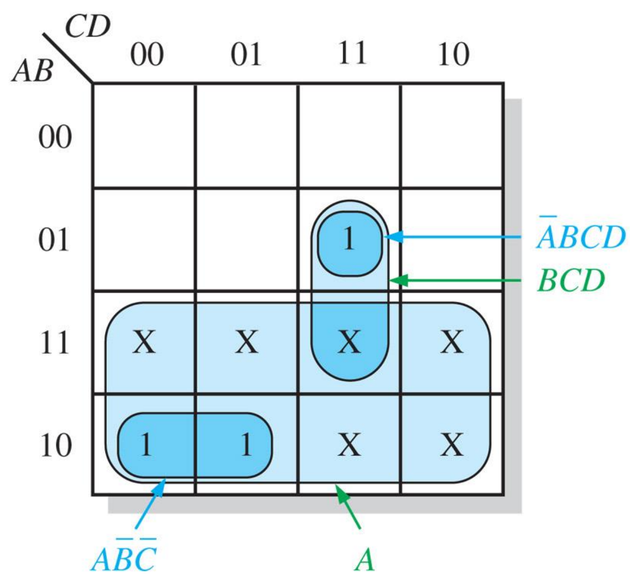


FIGURE 4-40 Example of the use of "don't care" conditions to simplify an expression.

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

Don't cares

(a) Truth table



(b) Without "don't cares" $Y = A\overline{B}\overline{C} + \overline{A}BCD$
With "don't cares" $Y = A + BCD$

FIGURE 4-41 7-segment display.

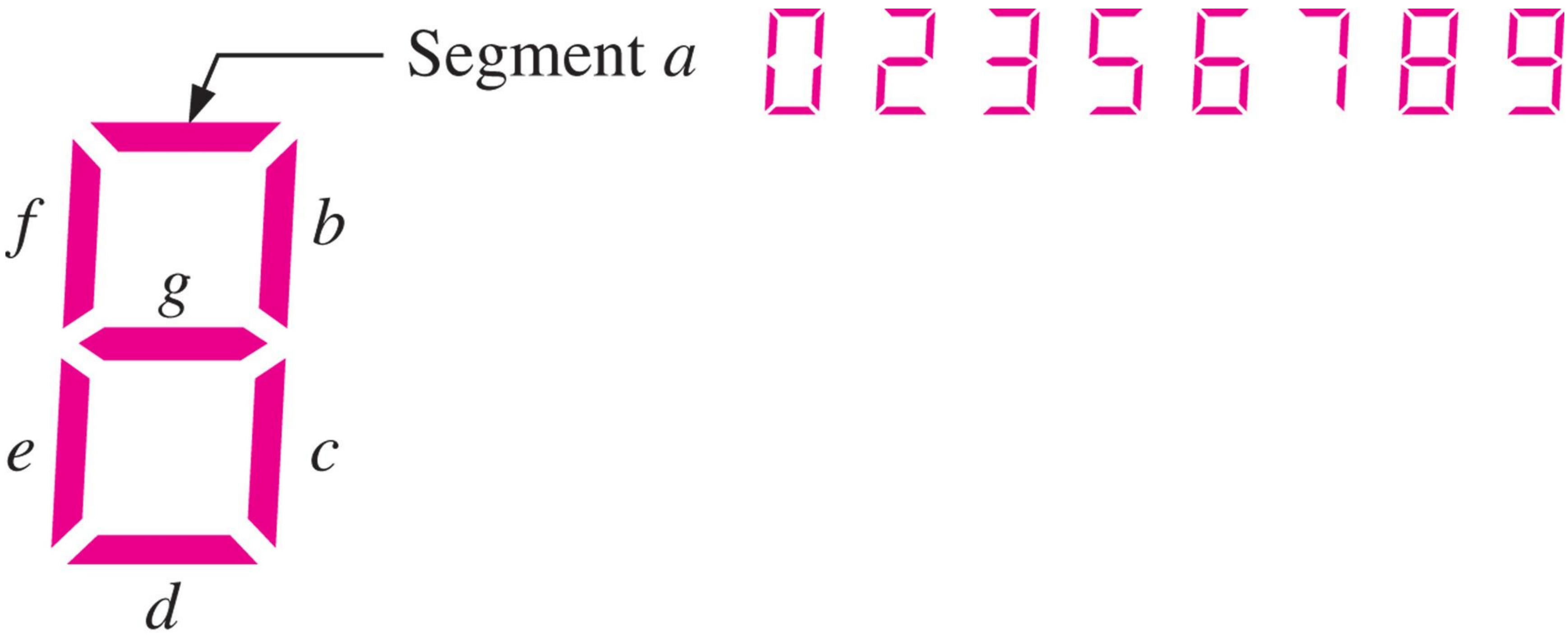


FIGURE 4-42

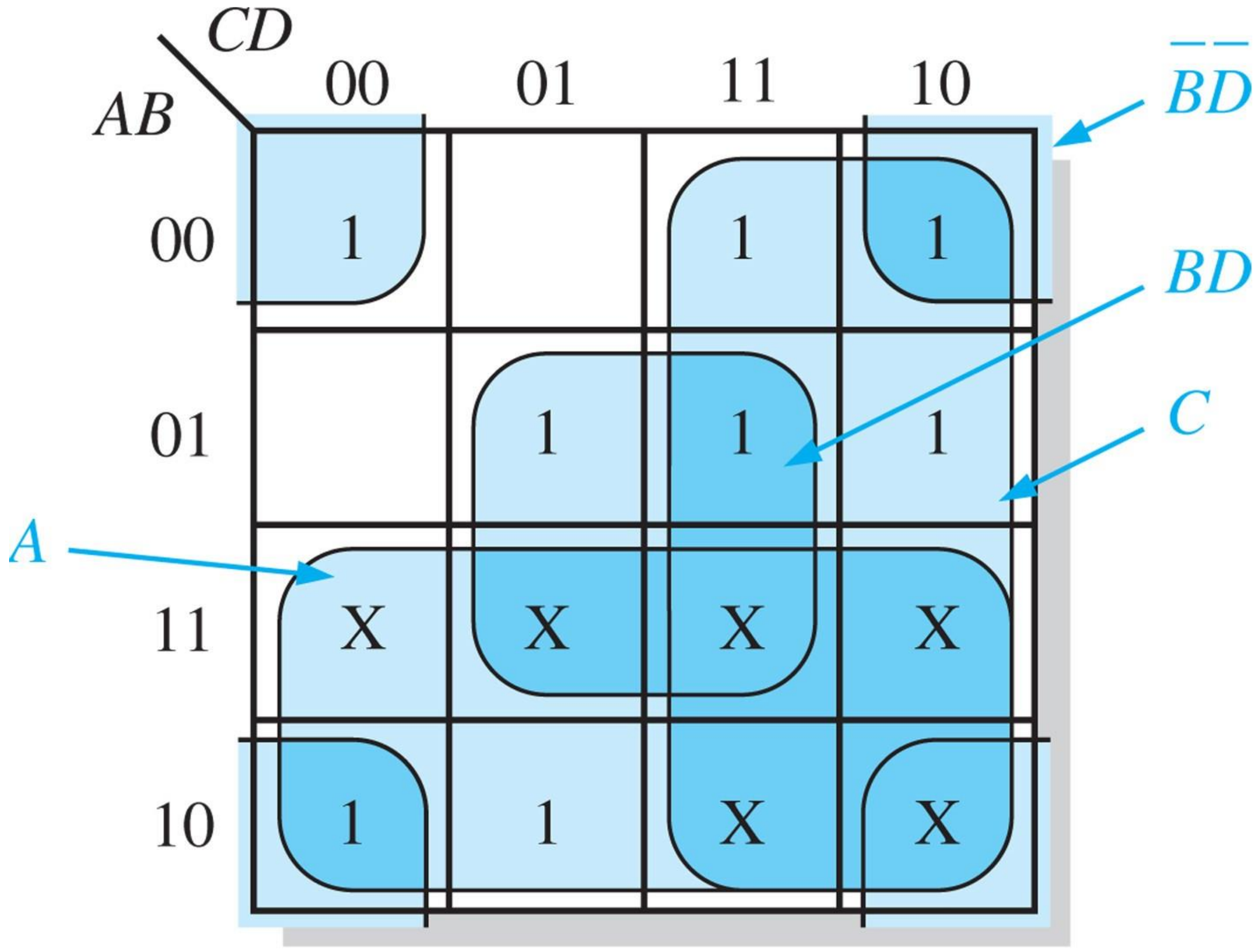
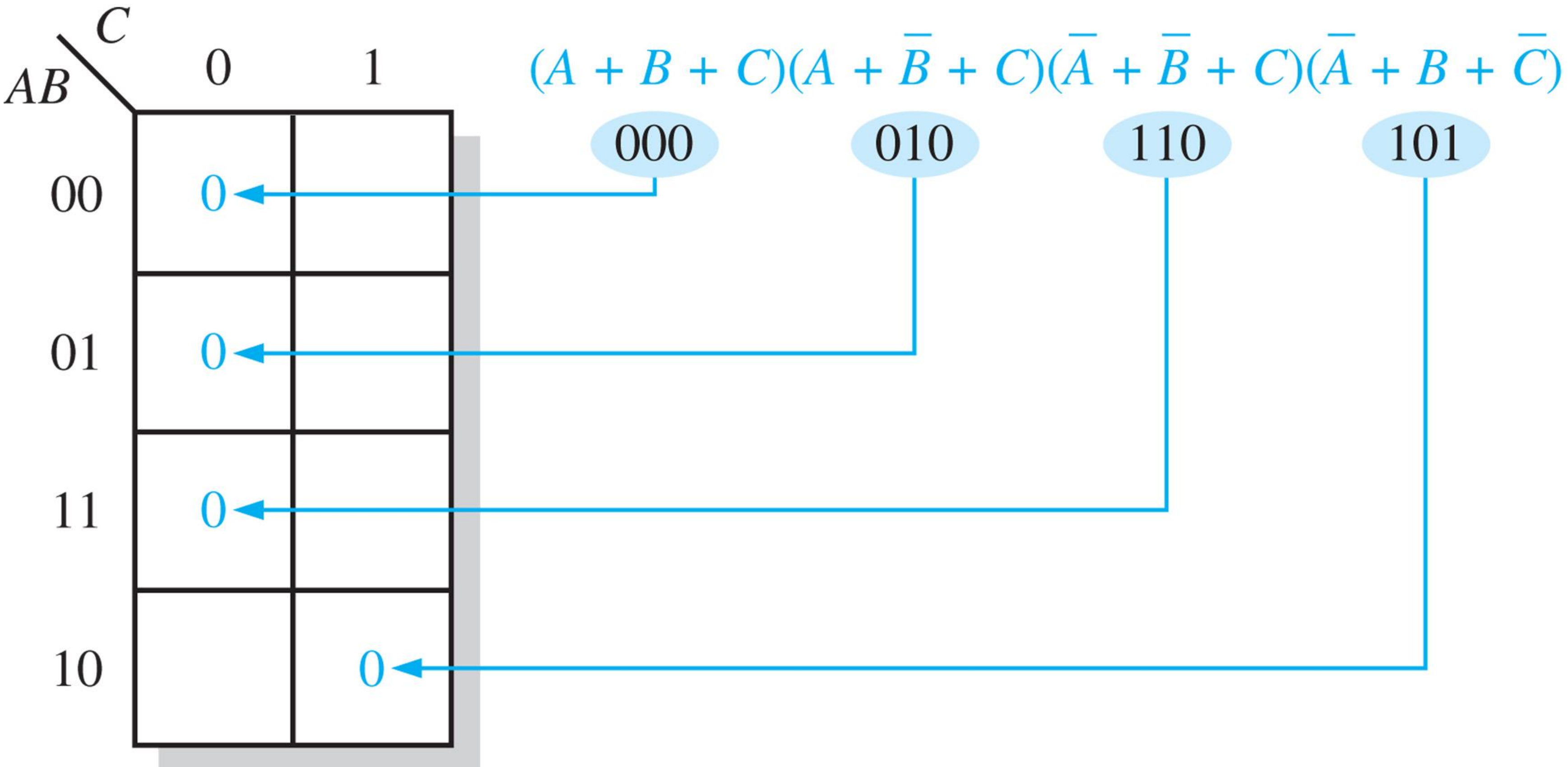
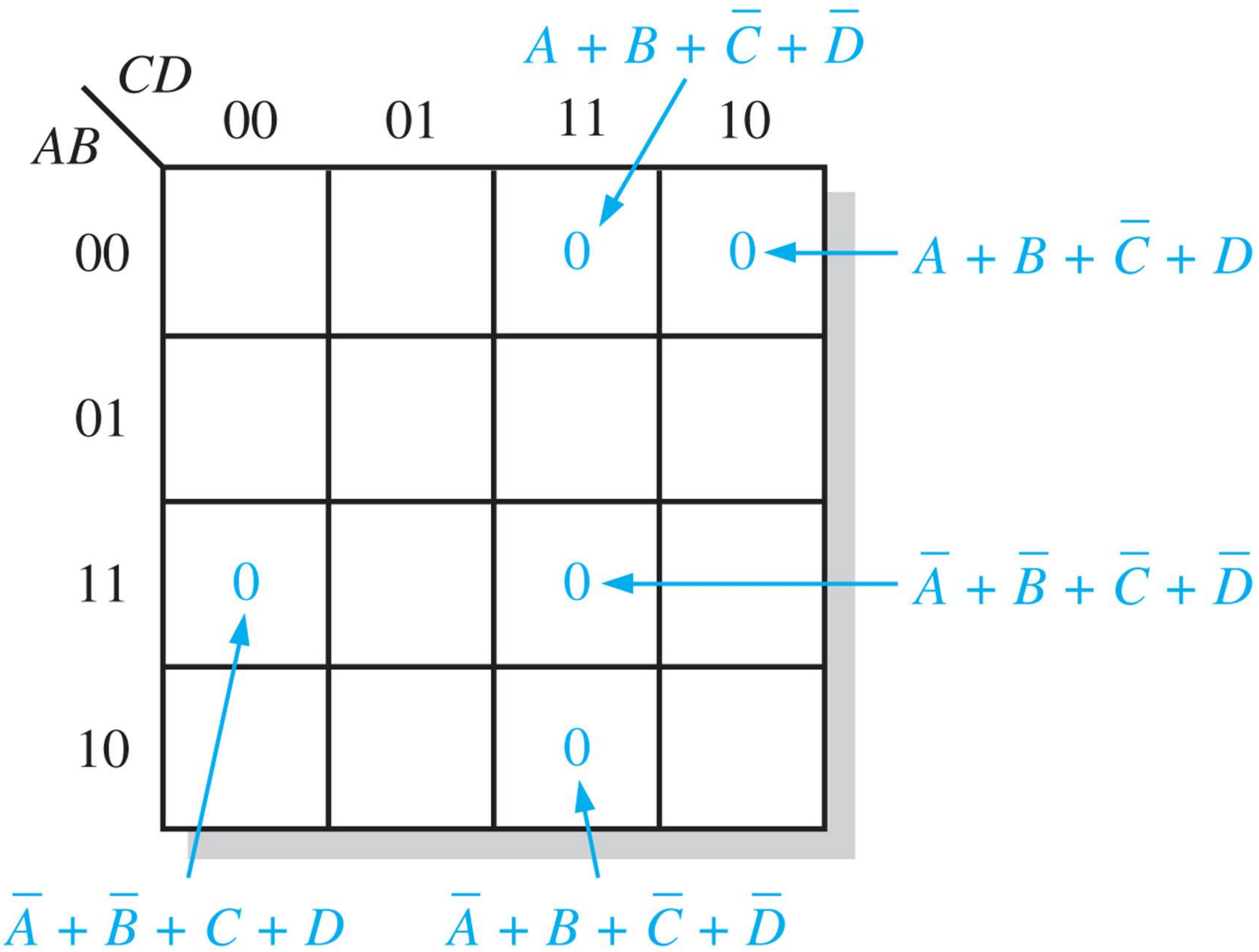

CD
AB
00
01
11
10
00
01
11
10
1
1
1
1
1
1
X
X
X
X
1
1
X
X
$\overline{B}\overline{D}$
BD
C
A

FIGURE 4-43 Example of mapping a standard POS expression.



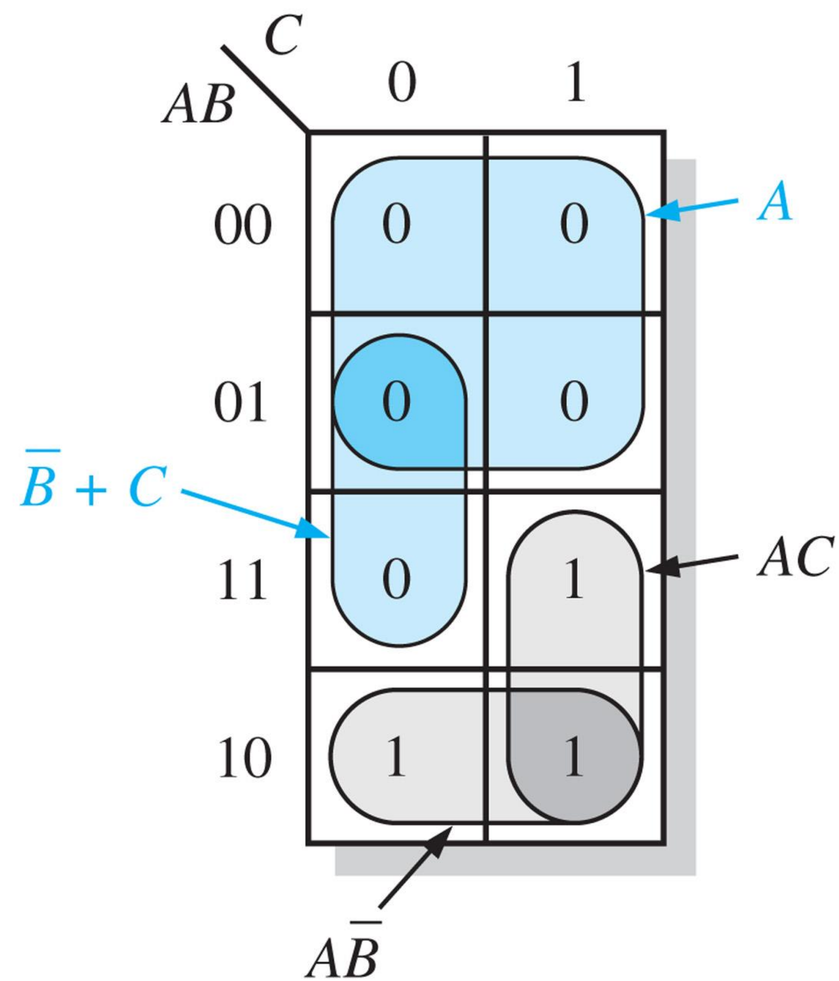
PEARSON

FIGURE 4-44



CD
AB
00
01
11
10
00
01
11
10
0
0
0
0
0
$A + B + \overline{C} + \overline{D}$
$A + B + \overline{C} + D$
$\overline{A} + \overline{B} + \overline{C} + \overline{D}$
$\overline{A} + \overline{B} + C + D$
$\overline{A} + B + \overline{C} + \overline{D}$

FIGURE 4-45



Digital Fundamentals, Eleventh Edition
Thomas L. Floyd

FIGURE 4-46

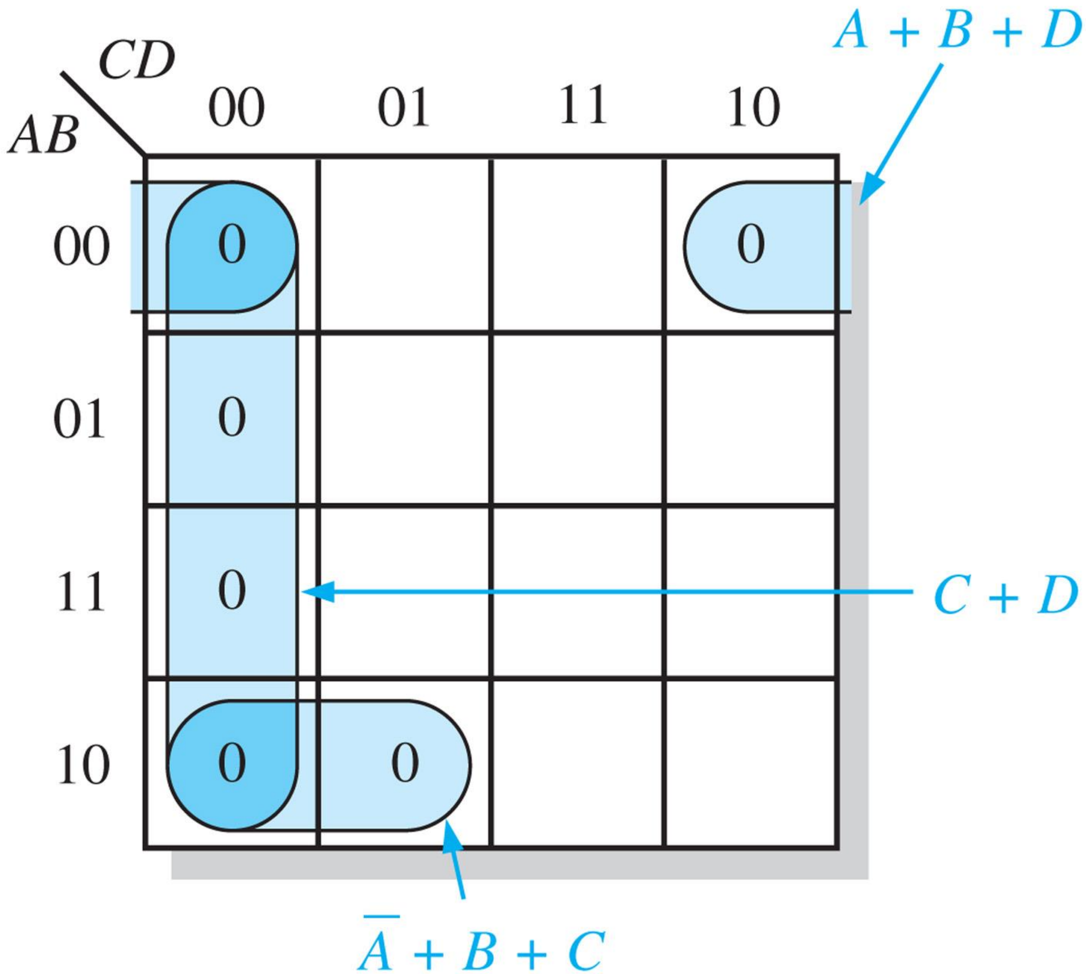
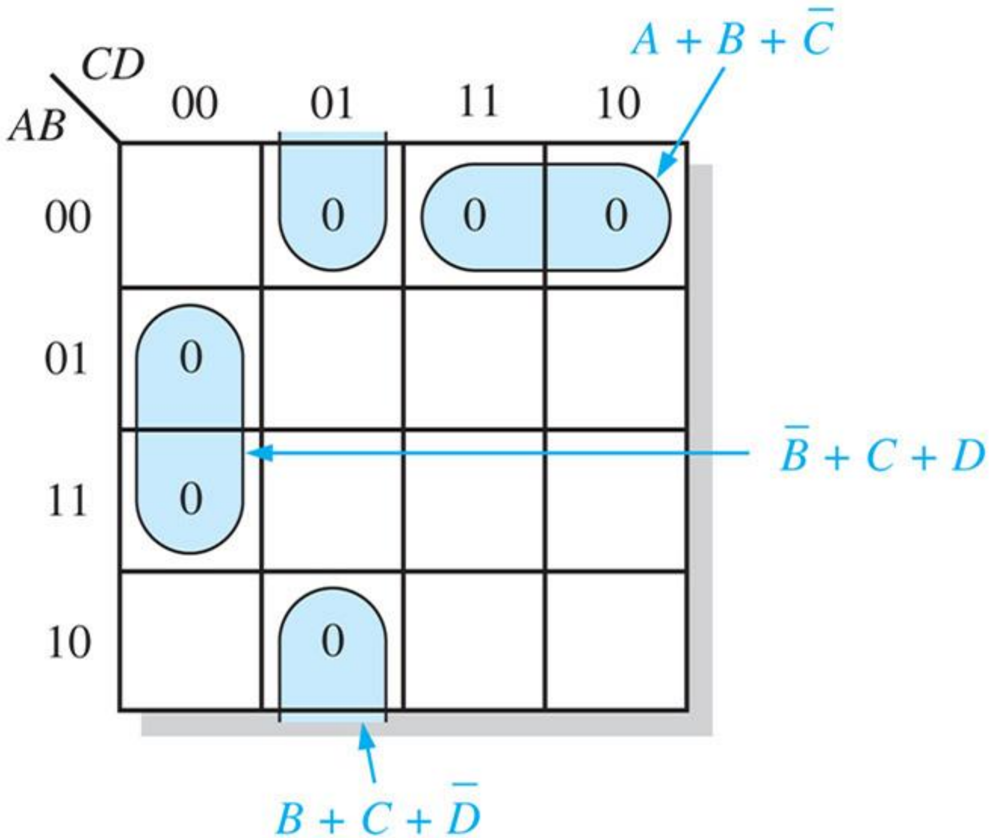
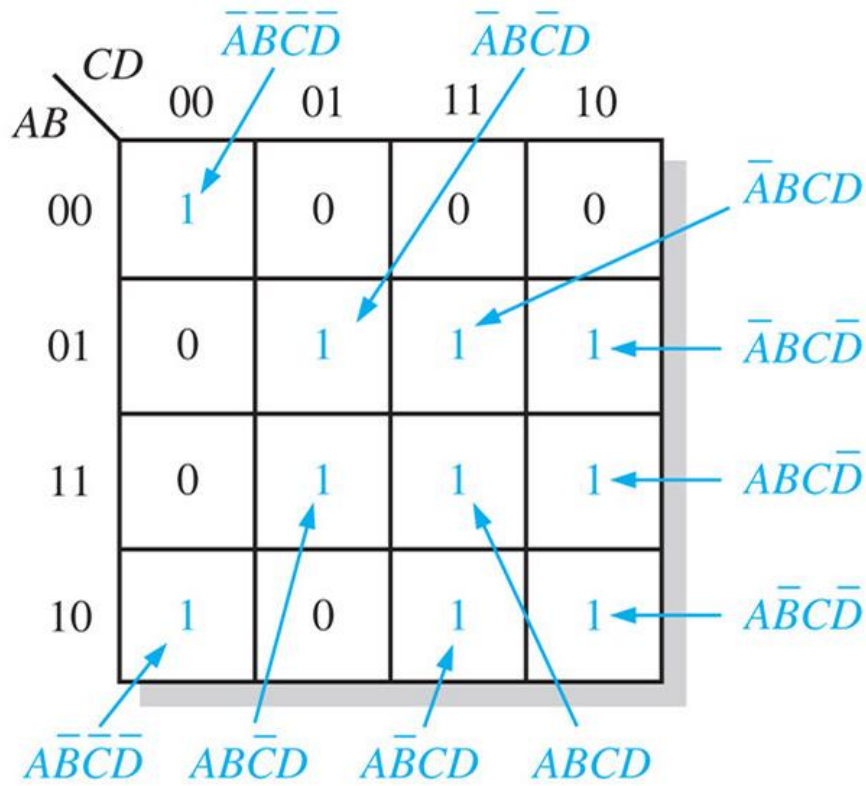


FIGURE 4-47



(a) Minimum POS: $(A + B + C)(\overline{B} + C + D)(B + C + \overline{D})$

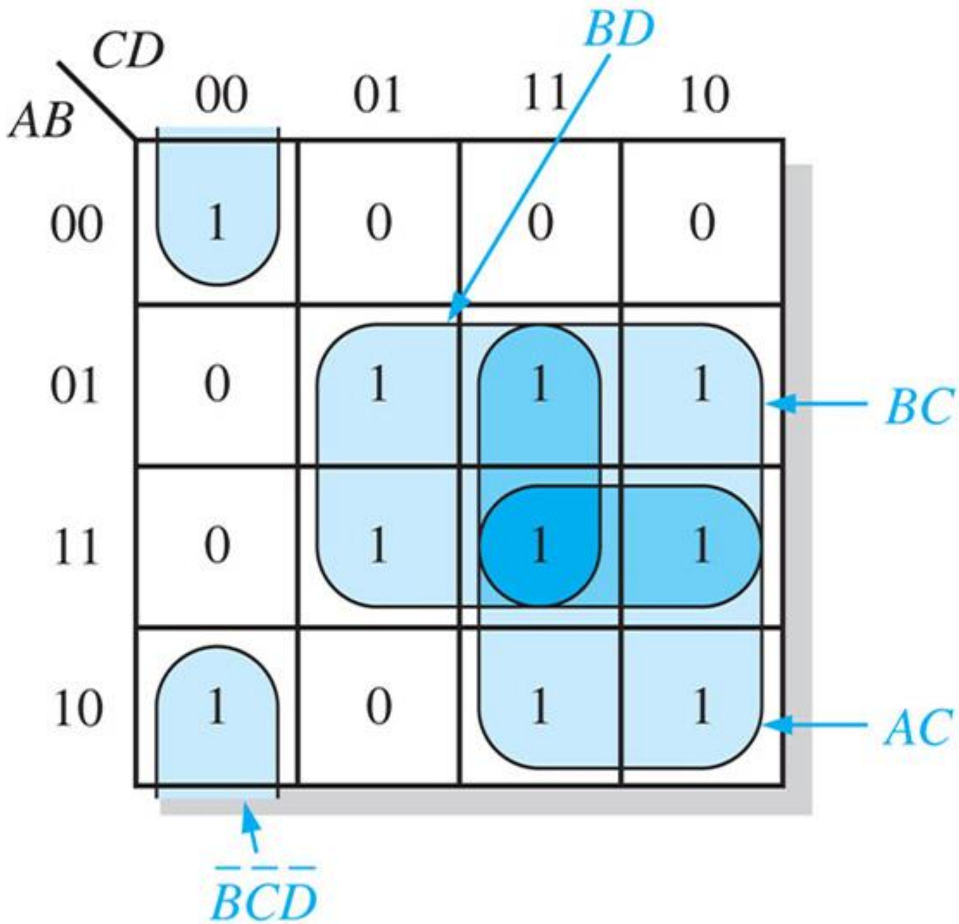
FIGURE 4-47 (continued)



(b) Standard SOP:

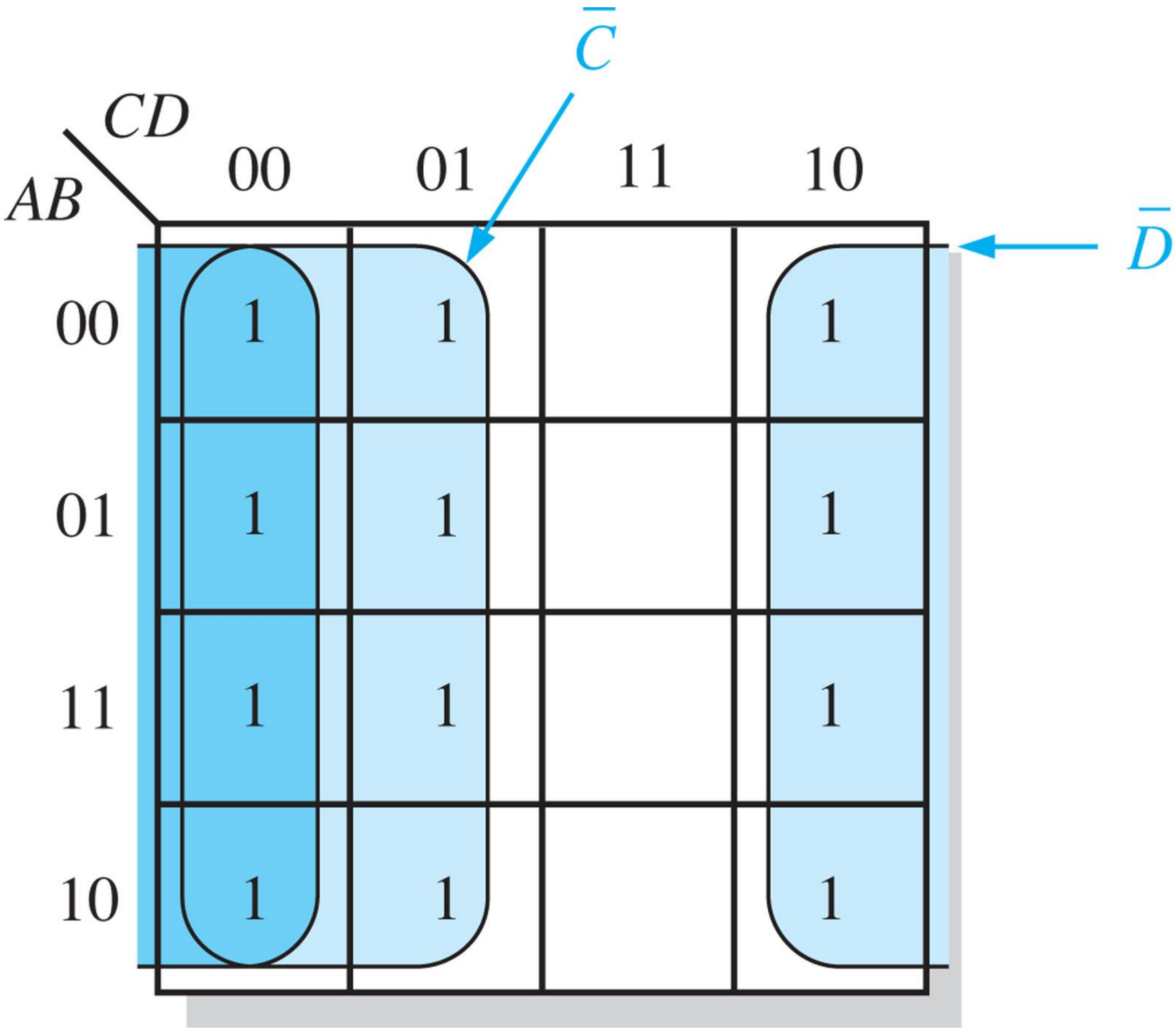
$$\bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}B\bar{C}D + \bar{A}BCD + \bar{A}BC\bar{D} + ABC\bar{D} + A\bar{B}C\bar{D} + A\bar{B}\bar{C}\bar{D} + AB\bar{C}D + A\bar{B}CD + ABCD$$

FIGURE 4-47 (continued)



(c) Minimum SOP: $AC + BC + BD + \overline{B}\,\overline{C}\,\overline{D}$

FIGURE 4-48



CD
AB
00
01
11
10
00
01
11
10
1 1 1
1 1 1
1 1 1
1 1 1
$\overline{C}$
$\overline{D}$

FIGURE 4-49 Illustration of the three levels of abstraction for describing a logic function.

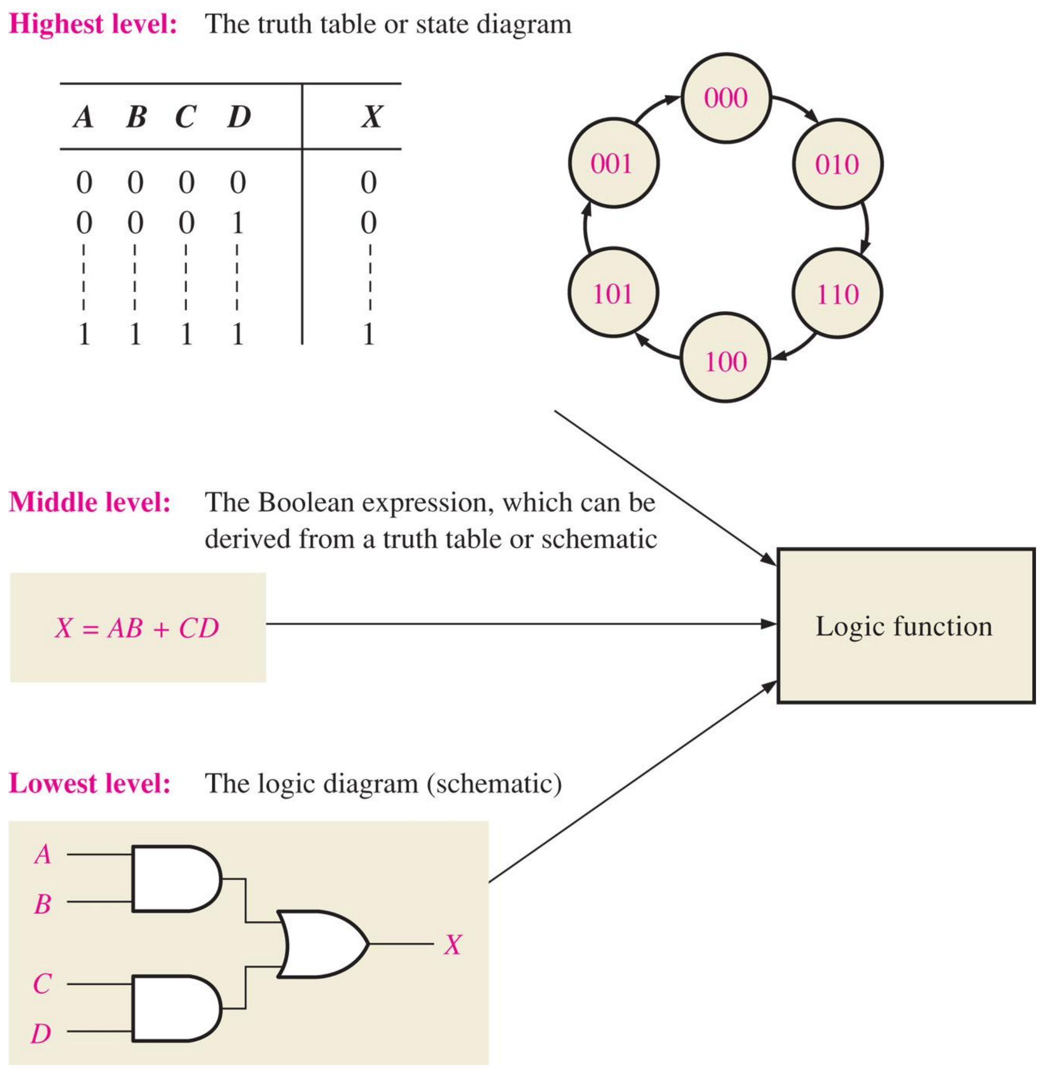


FIGURE 4-50 Seven-segment display.

(a) Segment arrangement

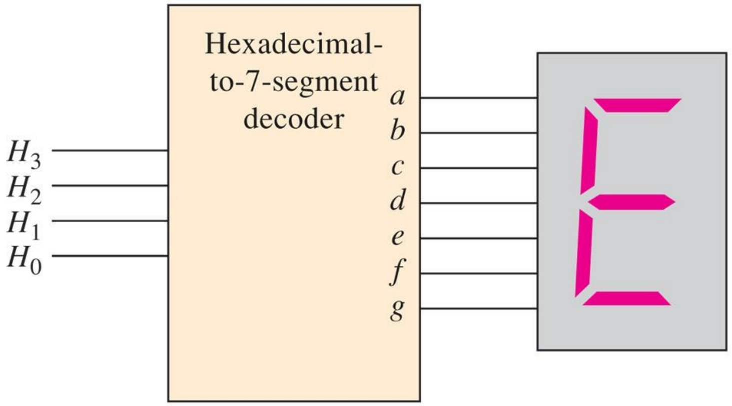
(b) Formation of the ten digits and certain letters

TABLE 4–14

Active segments for each of the five letters used in the system display.

Letter	Segments Activated
A	*a, b, c, e, f, g*
b	*c, d, e, f, g*
C	*a, d, e, f*
d	*b, c, d, e, g*
E	*a, d, e, f, g*

FIGURE 4-51 Hexadecimal-to-7-segment decoder for letters *A* through *E*, used in the system.

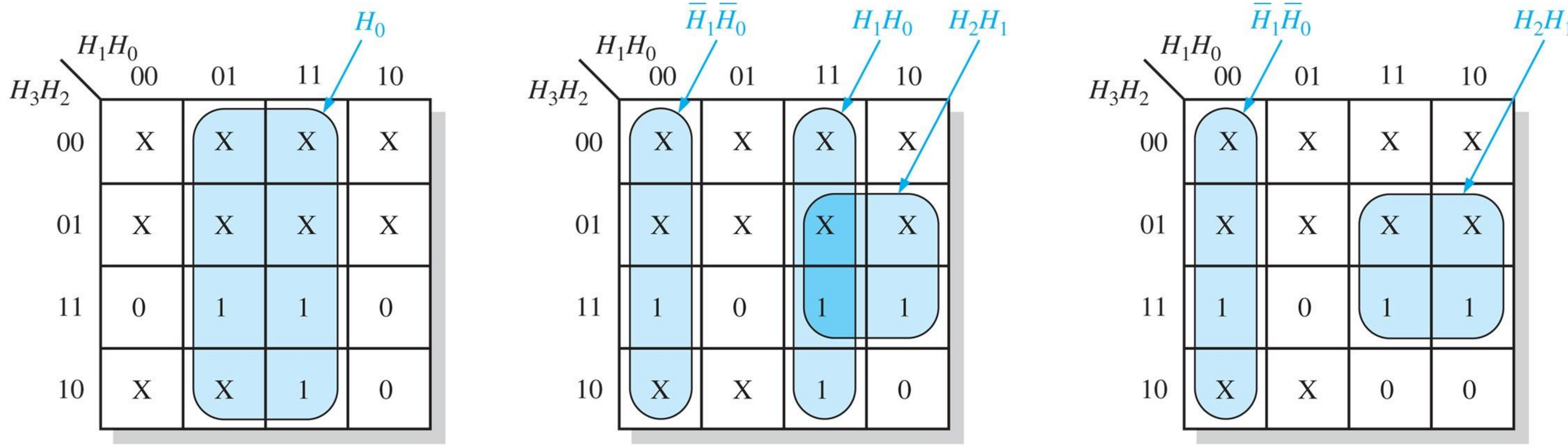


(a)

Letter	Hexadecimal Inputs				Segment Ouputs						
	H_3	H_2	H_1	H_0	*a*	*b*	*c*	*d*	*e*	*f*	*g*
A	1	0	1	0	0	0	0	1	0	0	0
b	1	0	1	1	1	1	0	0	0	0	0
C	1	1	0	0	0	1	1	0	0	0	1
d	1	1	0	1	1	0	0	0	0	1	0
E	1	1	1	0	0	1	1	0	0	0	0
F	1	1	1	1	1	1	1	1	1	1	1

(b)

FIGURE 4-52 Minimization of the expressions for segments *a, b,* and *c.*



$a = H_0$

(a)

$b = \overline{H}_1\overline{H}_0 + H_1H_0 + H_2H_1$

(b)

$c = \overline{H}_1\overline{H}_0 + H_2H_1$

(c)

FIGURE 4-53 Segment-*b* and segment-*c* logic circuits.