Combinational Logic Design

k-MAP

			Minterms		Maxterms	
X	Y	Z	Term	Designation	Term	Designation
0	0	0	$\bar{X}\bar{Y}\bar{Z}$	mo	$X+Y+Z$	Mo
0	0	1	$\bar{X}\bar{Y}Z$	m1	$X+Y+\bar{Z}$	M1
0	1	0	$\bar{X}Y\bar{Z}$	m2	$X+\bar{Y}+Z$	M2
0	1	1	$\bar{X}YZ$	m3	$X+\bar{Y}+\bar{Z}$	M3
1	0	0	$X\bar{Y}\bar{Z}$	m4	$\bar{X}+Y+Z$	M4
1	0	1	$X\bar{Y}Z$	m5	$\bar{X}+Y+\bar{Z}$	M5
1	1	0	$XY\bar{Z}$	m6	$\bar{X}+\bar{Y}+Z$	M6
1	1	1	XYZ	m7	$\bar{X}+\bar{Y}+\bar{Z}$	M7

Decimal Equivalent	Binary			Gray Code		
	B2	**B1**	**B0**	**G2**	**G1**	**G0**
0	0	0	0	**0**	**0**	**0**
1	0	0	1	**0**	**0**	**1**
2	0	1	0	**0**	**1**	**1**
3	0	1	1	**0**	**1**	**0**
4	1	0	0	**1**	**1**	**0**
5	1	0	1	**1**	**1**	**1**
6	1	1	0	**1**	**0**	**1**
7	1	1	1	**1**	**0**	**0**

K-Map

K-Map

A **Karnaugh map** is similar to a truth table because it presents all of the possible values of input variables and the resulting output for each value. Instead of being organized into columns and rows like a truth table, the Karnaugh map is an array of **cells** in which each cell represents a binary value of the input variables. The cells are arranged in a way so that simplification of a given expression is simply a matter of properly grouping the cells. Karnaugh maps can be used for expressions with two, three, four, and five variables, but we will discuss only 3-variable and 4-variable situations to illustrate the principles. *A discus-*

The number of cells in a Karnaugh map, as well as the number of rows in a truth table, is equal to the total number of possible input variable combinations. For three variables, the number of cells is $2^3 = 8$. For four variables, the number of cells is $2^4 = 16$.

The purpose of a Karnaugh map is to simplify a Boolean expression.

Rules of Boolean Algebra

Boolean Algebra:

A \ B	$\bar{B}$ 0	B 1
$\bar{A}$ 0	$\bar{A}\bar{B}$ 00 0	$\bar{A}B$ 01 1
A 1	$A\bar{B}$ 10 2	AB 11 3

3- Varibale k-map

Rules of Boolean Algebra

Boolean Algebra:

A \ BC	$\bar{B}\bar{C}$ 00	$\bar{B}C$ 01	BC 11	$B\bar{C}$ 10
$\bar{A}$ 0	$\bar{A}\bar{B}\bar{C}$ 000 0	$\bar{A}\bar{B}C$ 001 1	$\bar{A}BC$ 011 3	$\bar{A}B\bar{C}$ 010 2
A 1	$A\bar{B}\bar{C}$ 100 4	$A\bar{B}C$ 101 5	ABC 111 7	$AB\bar{C}$ 110 6

4- Varibale k-map

k-Map

AB \ CD	$\bar{C}\bar{D}$ 00	$\bar{C}D$ 01	CD 11	$C\bar{D}$ 10
$\bar{A}\bar{B}$ 00	0000 0	0001 1	0011 3	0010 2
$\bar{A}B$ 01	0100 4	0101 5	0111 7	0110 6
AB 11	1100 12	1101 13	1111 15	1110 14
$A\bar{B}$ 10	1000 8	1001 9	1011 11	1010 10

Grouping of Cells for Simplification

Grouping of Cells for Simplification

The grouping of cells for simplification is done in the following ways:

1. **Pairs:** If two adjacent cells either in row or in column contain 1's, they are said to be in pairs.

Grouping of two adjacent cells removes one variable and output will contain common variables in two terms.

Grouping of Cells for Simplification

The grouping of cells for simplification is done in the following ways:

2. **Quads:** We can group four adjacent cells in K-map forming a quad. Grouping a quad eliminates the two variables that appear in both normal and complemented form. Resultant terms contain $(n - 2)$ variables, where n is the number of variables.

Grouping of Cells for Simplification

The grouping of cells for simplification is done in the following ways:

3. **Octet:** A group of eight adjacent cells is called octet. Grouping of 1's eliminates three variables. Resultant terms contain $(n - 3)$ variables, where n is the number of variables.

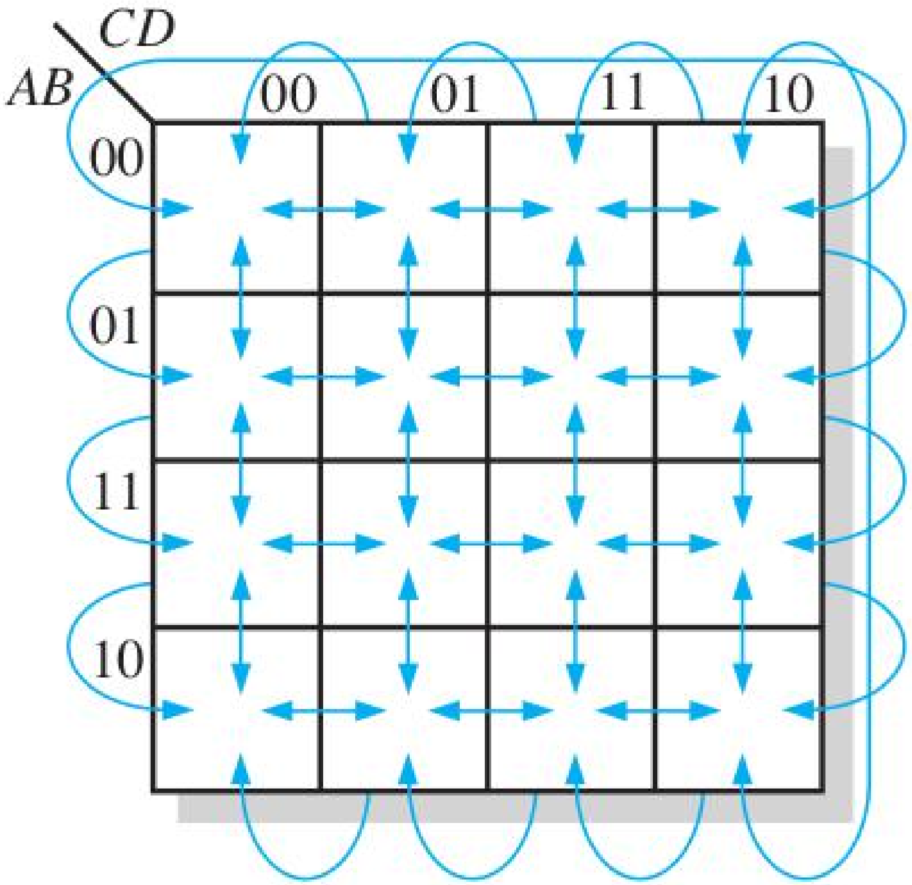
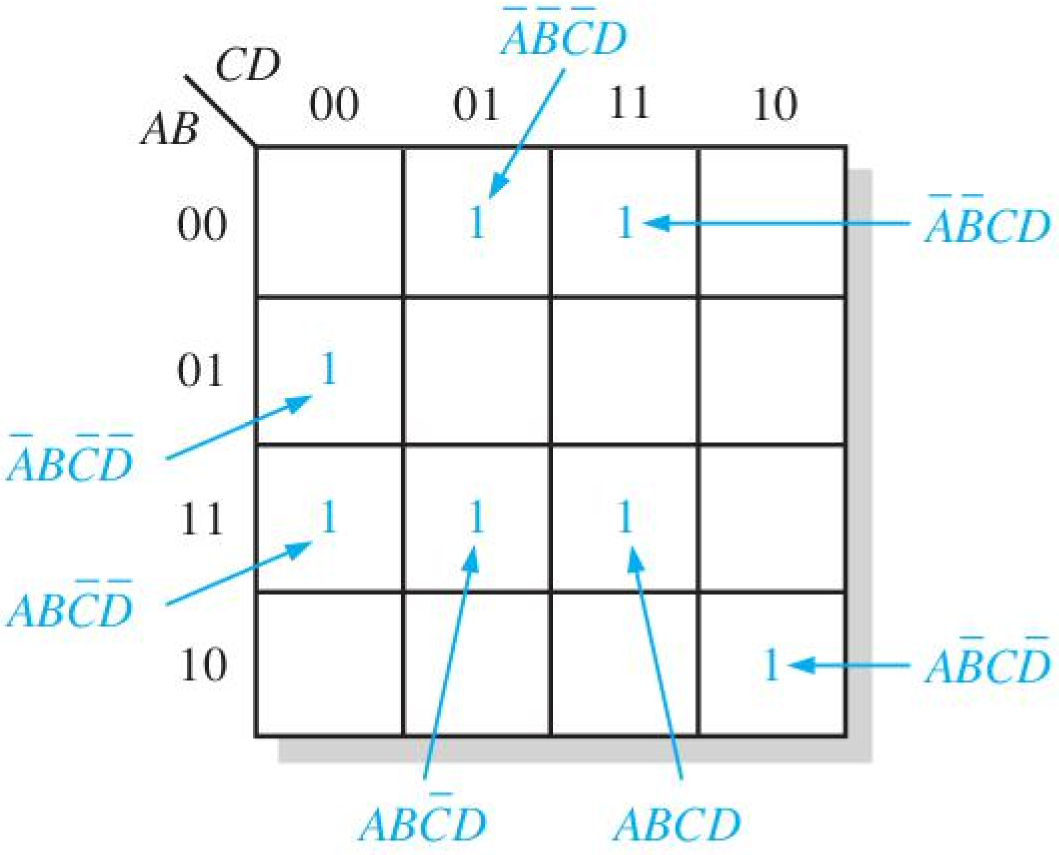


FIGURE 4–27 Adjacent cells on a Karnaugh map are those that differ by only one variable. Arrows point between adjacent cells.

k-Map

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

0011 0100 1101 1111 1100 0001 1010



Group the 1s in each of the Karnaugh maps

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

(c)

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

(d)

Group the 1s in each of the Karnaugh maps

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

(c)

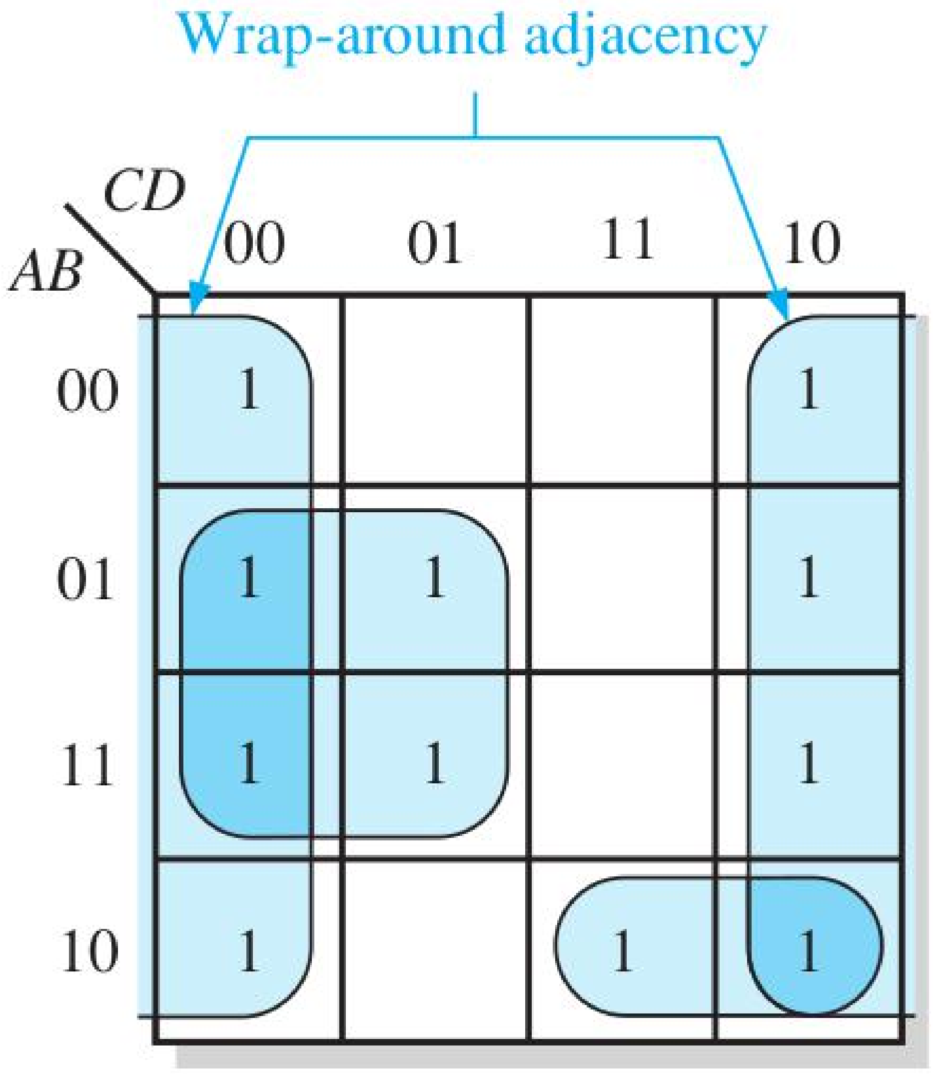
AB \ CD	00	01	11	10
00	1	1		
01	1	1	1	1
11				
10		1	1	

(c)

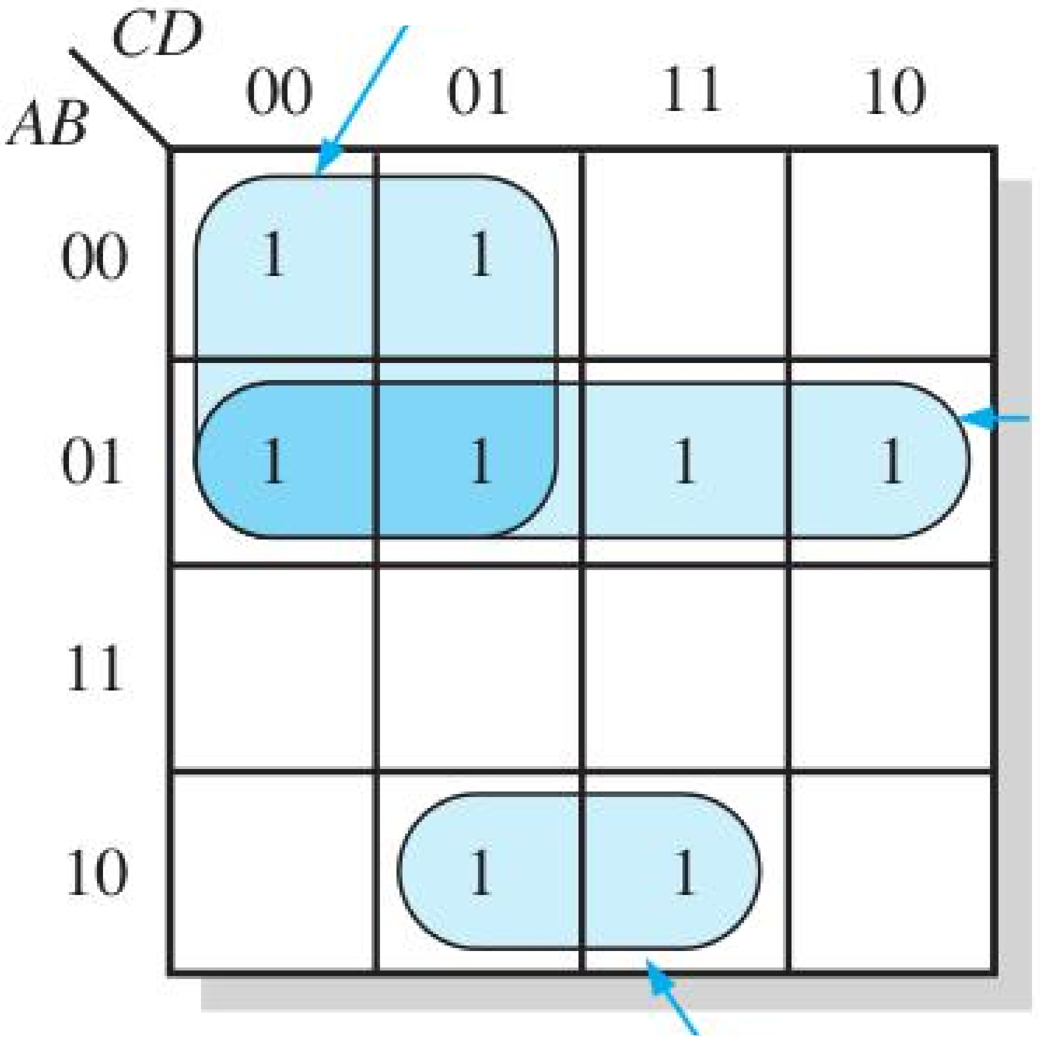
Group the 1s in each of the Karnaugh maps

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

(d)



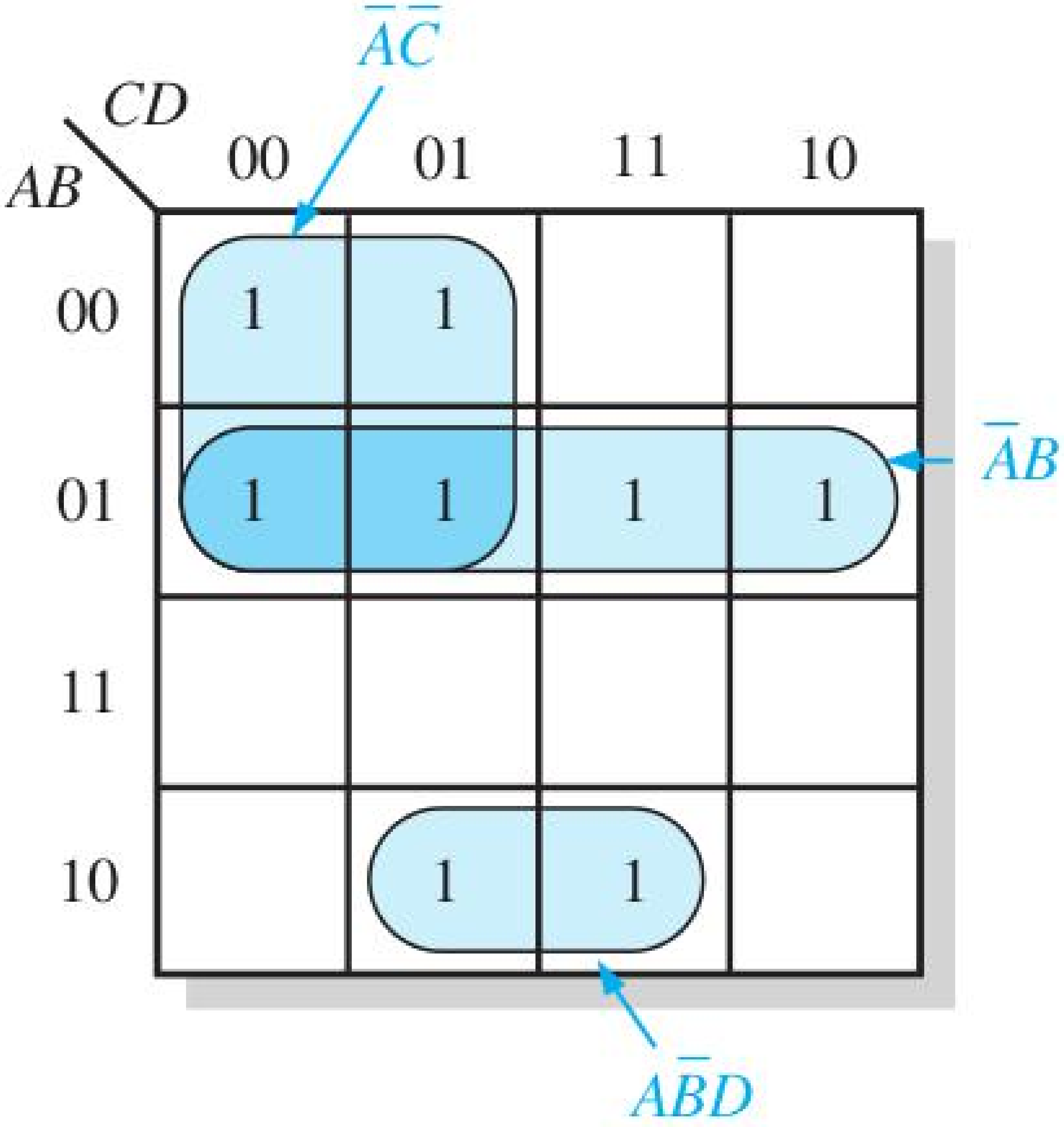
(d)



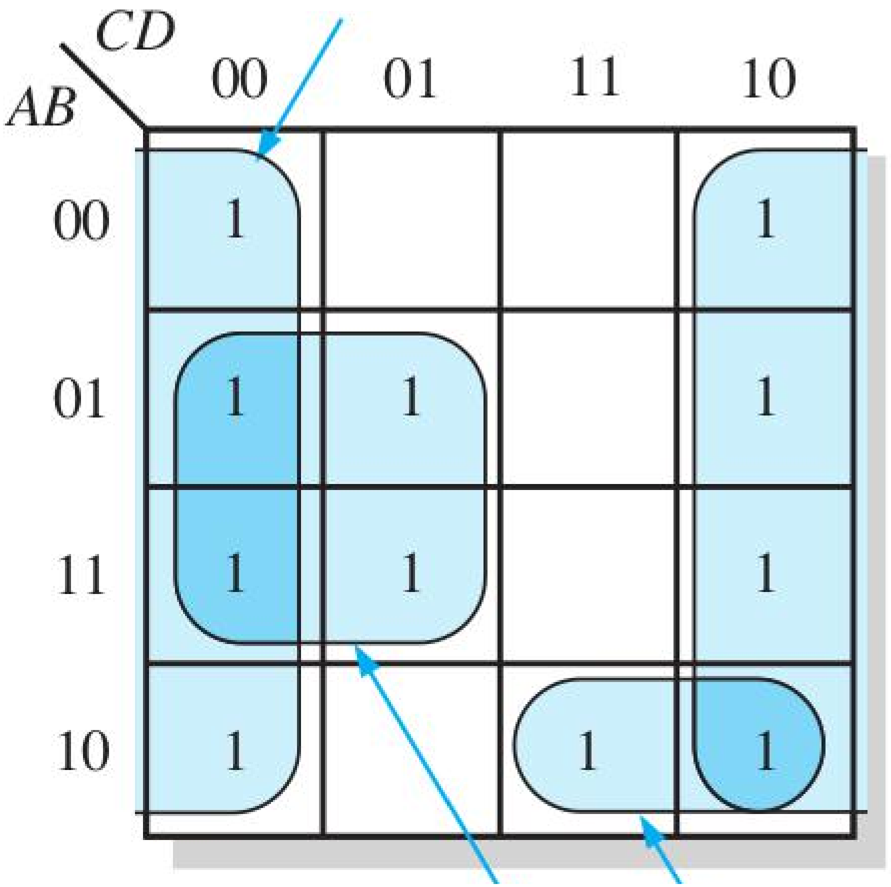
TRY THIS THEN

जुन **BIT** change हुन्छ , फालिदिने |

k-Map

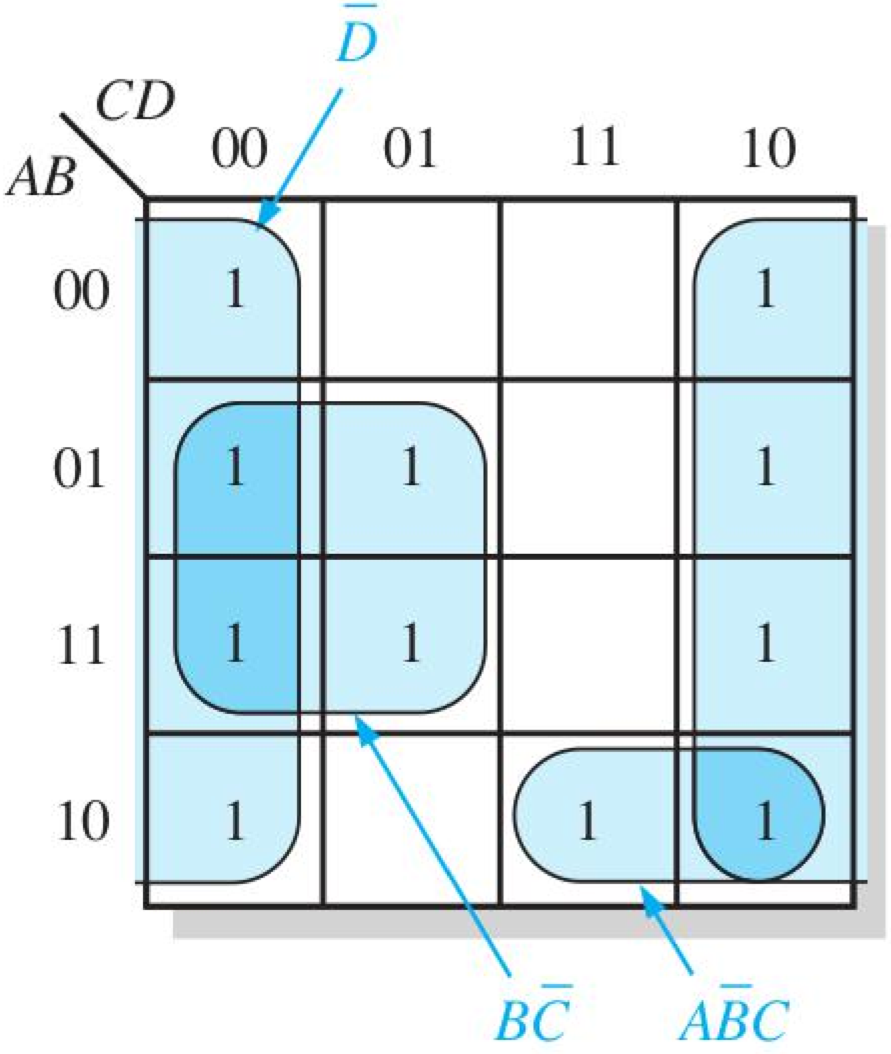


जुन **BIT** change हुन्छ , फालिदिने |



TRY THIS NOW

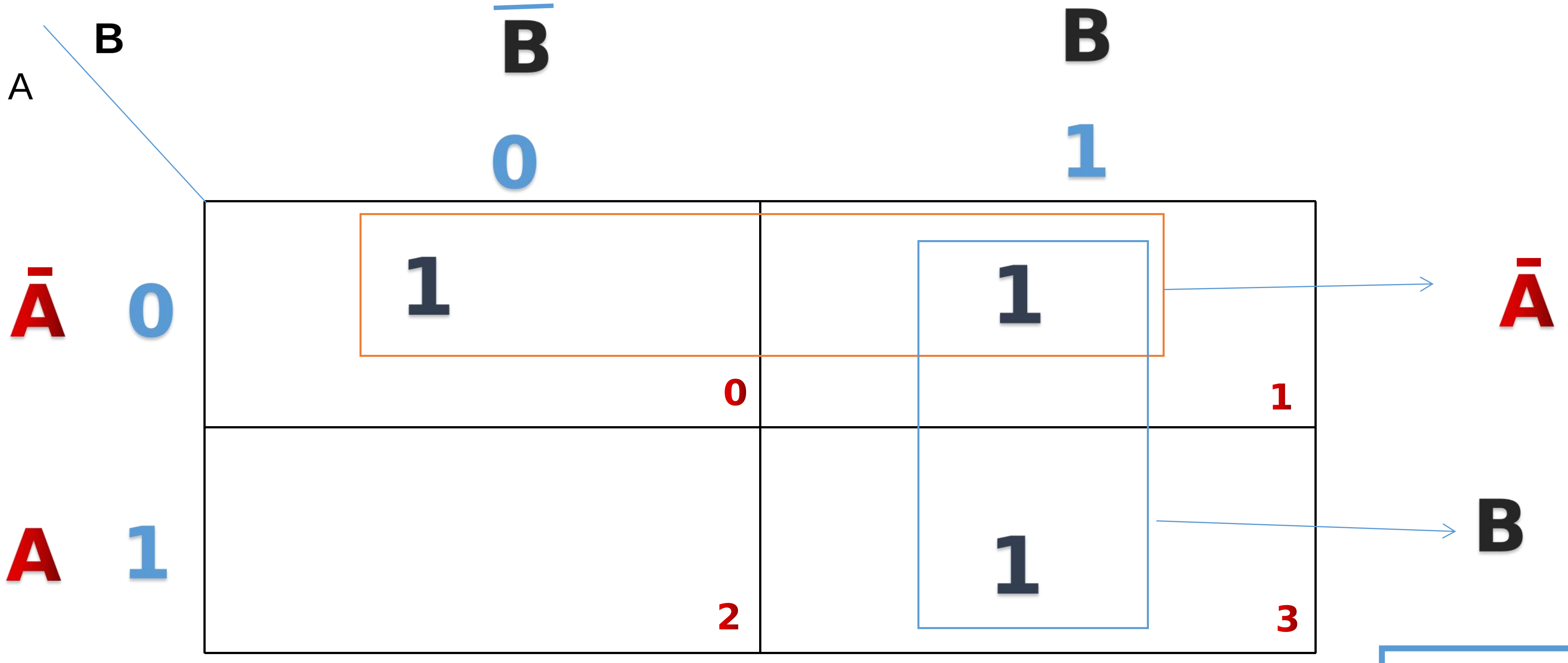
जुन **BIT** chnage हुन्छ , फालिदिने |



ANSWER

जुन **BIT** chnage हुन्छ , फालिदिने |

Example 1: $f(A,B) = \Sigma m(0,1,3)$



$\boxed{\therefore \bar{A} + B}$

Example2: $f(A,B) = \Sigma m(0,1,2,3)$

Example 2: $f(A,B) = \Sigma m(0,1,2,3)$

A \ B		$\bar{B}$ 0	B 1
$\bar{A}$	0	1 (0)	1 (1)
A	1	1 (2)	1 (3)

1

Example 3: f(A,B,C) $=\Sigma m(0,1,3,6,7)$

k-Map

Example4: $f(A,B,C) = \sum m(0,1,3,6,7)$

A \ BC	$\bar{B}\bar{C}$ 00	$\bar{B}C$ 01	BC 11	$B\bar{C}$ 10
$\bar{A}$ 0	1 (0)	1 (1)	1 (3)	(2)
A 1	(4)	(5)	1 (7)	1 (6)

$\bar{A}\bar{B}$

BC

AB

$\therefore \bar{A}\bar{B} + AB + BC$

Example5

Use a Karnaugh map to minimize the following standard SOP expression:

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

Solution

The binary values of the expression are

$$101 + 011 + 001 + 000 + 100$$

Example5: f(A,B,C) $=\Sigma m(5,3,1,0,4)$

E3: $f(A,B,C) = \Sigma m(0,1,3,4,5)$

A \ BC	$\bar{B}\bar{C}$ 00	$\bar{B}C$ 01	BC 11	$B\bar{C}$ 10
$\bar{A}$ 0	1 0	1 1	1 3	 2
A 1	1 4	1 5	 7	 6

$\rightarrow \bar{A}C$

$\bar{B}$

$\boxed{\therefore \bar{B} + \bar{A}C}$

4 Variable k-Map Examples

k-Map

Use a Karnaugh map to minimize the following SOP expression:

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

4 **Solution** 12 3 11 2 6 14 10

The first term $\overline{B}\,\overline{C}\,\overline{D}$ must be expanded into $A\overline{B}\,\overline{C}\,\overline{D}$ and $\overline{A}\,\overline{B}\,\overline{C}\,\overline{D}$ to get the standard SOP expression, which is then mapped; the cells are grouped as shown in Figure 4–38. 0

8

E6: f(A,B,C,D) =Σm(0,2,3,4,6,8,10,11,12,14)

E6: $f(A,B,C,D) = \Sigma m(0,2,3,4,6,8,10,11,12,14)$

CD / AB	$\bar{C}\bar{D}$ 00	$\bar{C}D$ 01	CD 11	$C\bar{D}$ 10
$\bar{A}\bar{B}$ 00	1 0	1	1 3	1 2
$\bar{A}B$ 01	1 4	5	7	1 6
AB 11	1 12	13	15	1 14
$A\bar{B}$ 10	1 8	9	1 11	1 10

$\bar{B}C$

$\bar{D}$

E6: f(A,B,C,D) =Σm(0,2,3,4,6,8,10,11,12,14)

Use a Karnaugh map to minimize the following SOP expression:

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

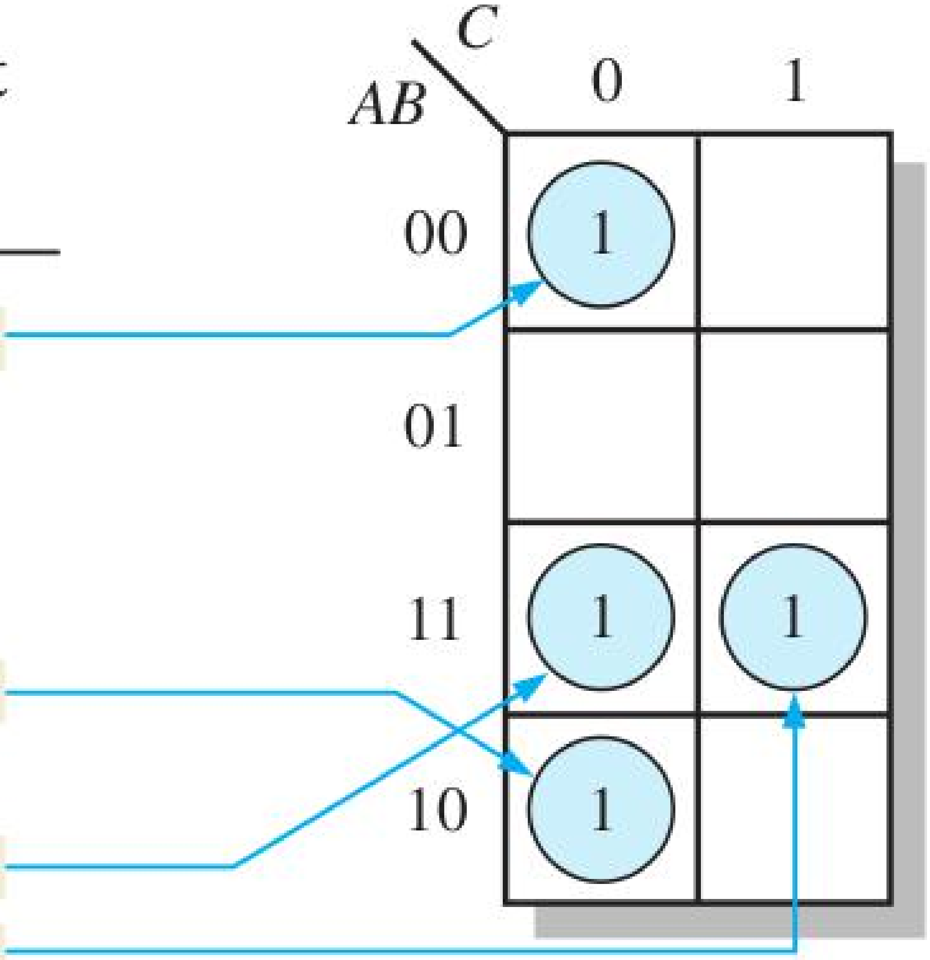
Notice that both groups exhibit "wrap around" adjacency. The group of eight is formed because the cells in the outer columns are adjacent. The group of four is formed to pick up the remaining two 1s because the top and bottom cells are adjacent. The product term for each group is shown. The resulting minimum SOP expression is

$$\overline{D} + \overline{B}C$$

Keep in mind that this minimum expression is equivalent to the original standard expression.

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

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



Example of mapping directly from a truth table to a Karnaugh map.

“Don’t Care” Conditions

“Don’t Care” Conditions

Sometimes a situation arises in which some input variable combinations are not allowed. For example, recall that in the BCD code covered in Chapter 2, there are six invalid combinations: 1010, 1011, 1100, 1101, 1110, and 1111. Since these unallowed states will never occur in an application involving the BCD code, they can be treated as **“don’t care”** terms with respect to their effect on the output. That is, for these “don’t care” terms either a 1 or a 0 may be assigned to the output; it really does not matter since they will never occur.

The “don’t care” terms can be used to advantage on the Karnaugh map. Figure 4–40 shows that for each “don’t care” term, an X is placed in the cell. When grouping the 1s, the Xs can be treated as 1s to make a larger grouping or as 0s if they cannot be used to advantage. The larger a group, the simpler the resulting term will be.

Without "don't cares" $Y = A\bar{B}\bar{C} + \bar{A}BCD$

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

AB \ CD	$\bar{C}\bar{D}$ 00	$\bar{C}D$ 01	CD 11	$C\bar{D}$ 10
$\bar{A}\bar{B}$ 00	0	1	3	2
$\bar{A}B$ 01	4	5	1 7	6
AB 11	12	13	15	14
$A\bar{B}$ 10	1 8	1 9	11	10

Lets Fill Dont Care Conditions Now

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

AB \ CD	$\bar{C}\bar{D}$ 00	$\bar{C}D$ 01	CD 11	$C\bar{D}$ 10
$\bar{A}\bar{B}$ 00	0	1	3	2
$\bar{A}B$ 01	4	5	1 7	6
AB 11	X 12	X 13	X 15	X 14
$A\bar{B}$ 10	1 8	1 9	X 11	X 10

With “don’t cares” $Y = A + BCD$

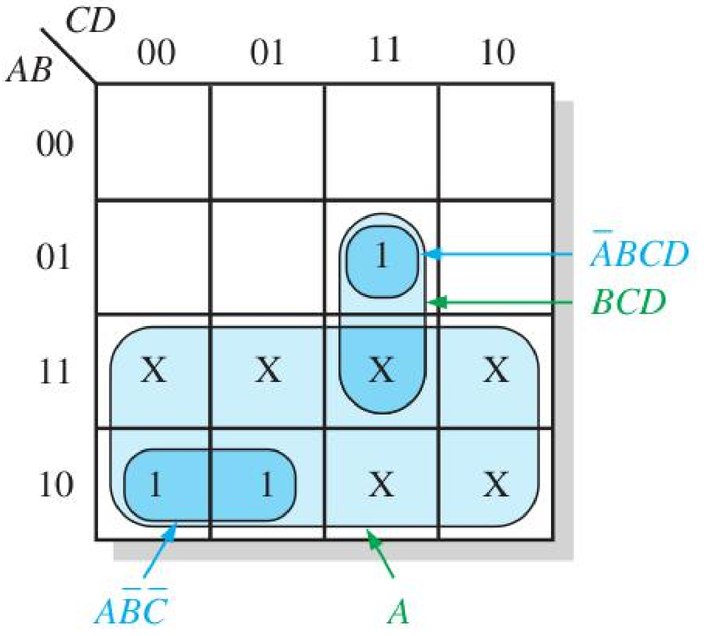
Inputs				Output
A	B	C	D	Y
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	X
1	0	1	1	X
1	1	0	0	X
1	1	0	1	X
1	1	1	0	X
1	1	1	1	X

Don’t cares

AB \ CD	$\bar{C}\bar{D}$ 00	$\bar{C}D$ 01	CD 11	$C\bar{D}$ 10
$\bar{A}\bar{B}$ 00	0	1	3	2
$\bar{A}B$ 01	4	5	1 7	6
AB 11	X 12	X 13	X 15	X 14
$A\bar{B}$ 10	1 8	1 9	X 11	X 10

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

(a) Truth table



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

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

The truth table in Figure 4–40(a) describes a logic function that has a 1 output only when the BCD code for 7, 8, or 9 is present on the inputs. If the "don't cares" are used as 1s, the resulting expression for the function is $A + BCD$, as indicated in part (b). If the "don't cares" are not used as 1s, the resulting expression is $A\bar{B}\,\bar{C} + \bar{A}BCD$; so you can see the advantage of using "don't care" terms to get the simplest expression.

In a 7-segment display, each of the seven segments is activated for various digits. For example, segment *a* is activated for the digits 0, 2, 3, 5, 6, 7, 8, and 9, as illustrated in Figure 4–41. Since each digit can be represented by a BCD code, derive an SOP expression for segment *a* using the variables *ABCD* and then minimize the expression using a Karnaugh map.

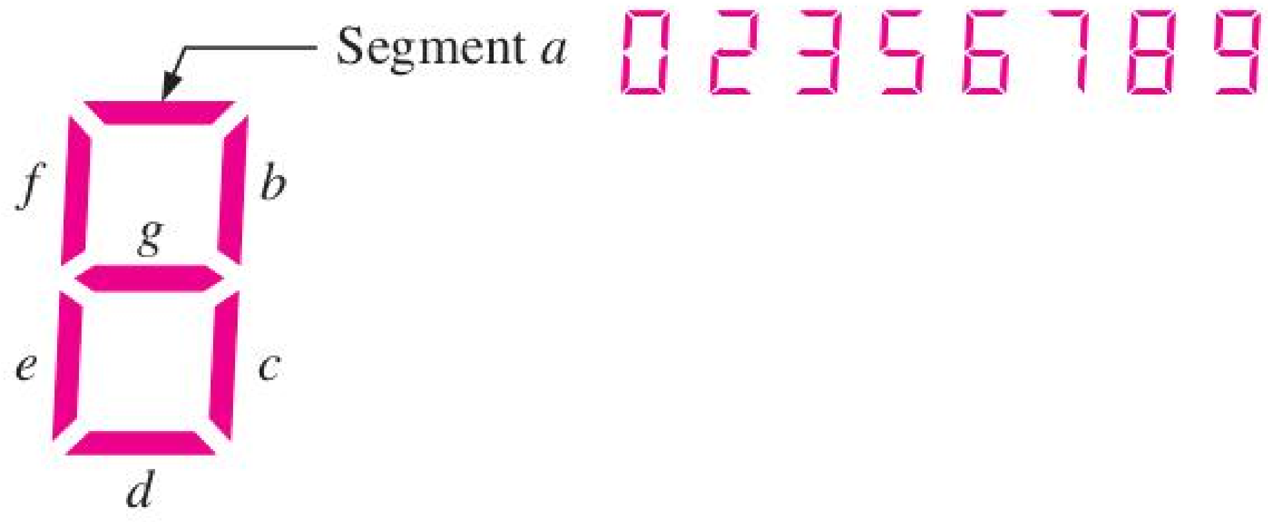


FIGURE 4–41 7-segment display.

In a 7-segment display, each of the seven segments is activated for various digits. For example, segment a is activated for the digits 0, 2, 3, 5, 6, 7, 8, and 9, as illustrated in Figure 4–41. Since each digit can be represented by a BCD code, derive an SOP expression for segment a using the variables $ABCD$ and then minimize the expression using a Karnaugh map.

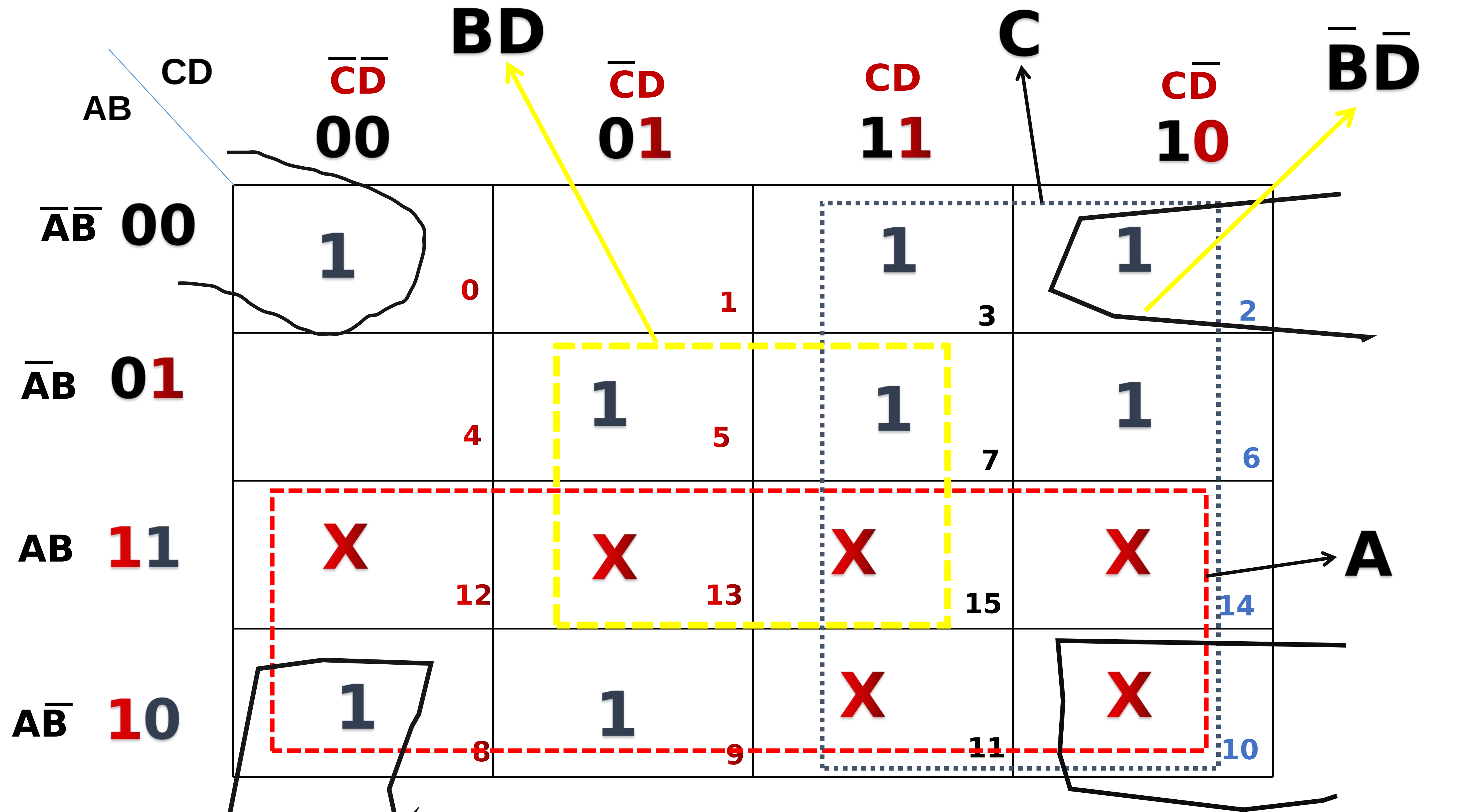
FIGURE 4–41 7-segment display.

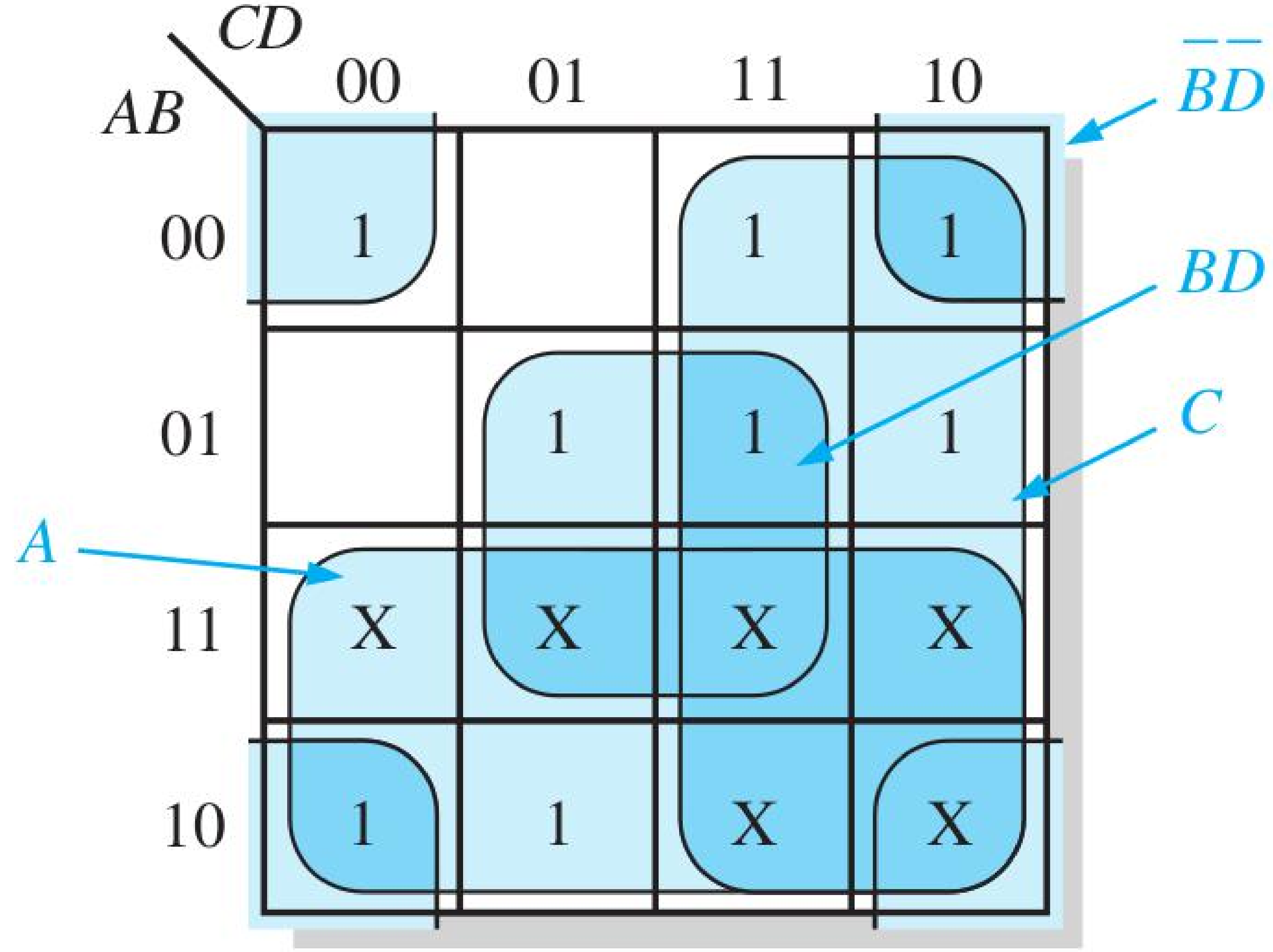
Solution

The expression for segment a is

$$a = \overline{A}\,\overline{B}\,\overline{C}\,\overline{D} + \overline{A}\,\overline{B}C\overline{D} + \overline{A}\,\overline{B}CD + \overline{A}B\overline{C}D + \overline{A}BC\overline{D} + \overline{A}BCD + A\overline{B}\,\overline{C}\,\overline{D} + A\overline{B}\,\overline{C}D$$

Each term in the expression represents one of the digits in which segment a is used. The Karnaugh map minimization is shown in Figure 4–42. X's (don't cares) are entered for those states that do not occur in the BCD code.


CD
AB
$\bar{C}\bar{D}$ 00
BD
$\bar{C}D$ 01
CD 11
C
$C\bar{D}$ 10
$\bar{B}\bar{D}$
$\bar{A}\bar{B}$ 00
$\bar{A}B$ 01
AB 11
$A\bar{B}$ 10
A
1 0
1
1 3
1 2
4
1 5
1 7
1 6
X 12
X 13
X 15
X 14
1 8
1 9
X 11
X 10



From the Karnaugh map, the minimized expression for segment *a* is

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

From the Karnaugh map, the minimized expression for segment a is

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

Related Problem

Draw the logic diagram for the segment-a logic.

Karnaugh Map POS Minimization

Karnaugh Map POS Minimization

In the last section, you studied the minimization of an SOP expression using a Karnaugh map. In this section, we focus on POS expressions. The approaches are much the same except that with POS expressions, 0s representing the standard sum terms are placed on the Karnaugh map instead of 1s.

Mapping a Standard POS Expression

Mapping a Standard POS Expression

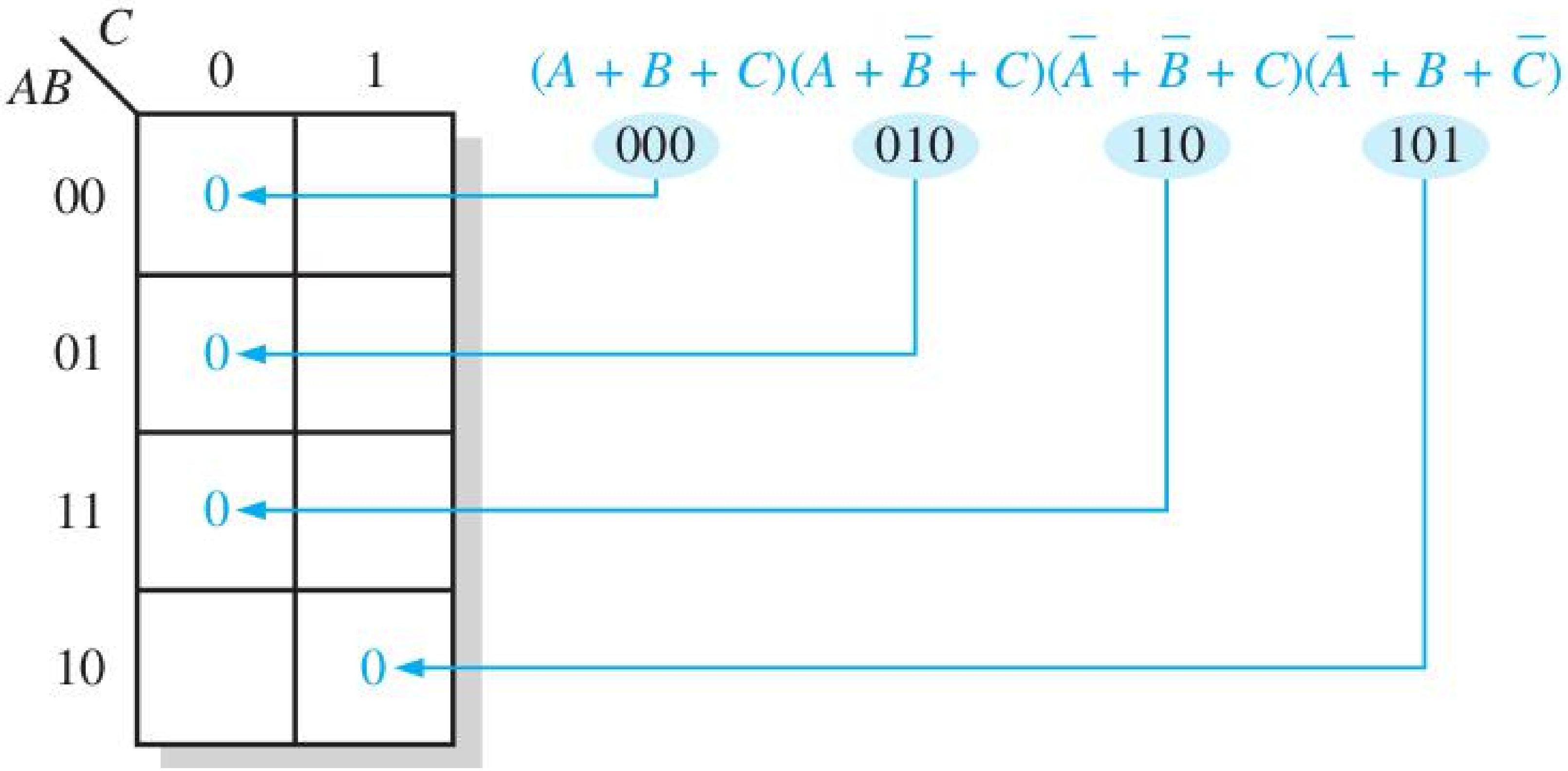
For a POS expression in standard form, a 0 is placed on the Karnaugh map for each sum term in the expression. Each 0 is placed in a cell corresponding to the value of a sum term. For example, for the sum term $A + \overline{B} + C$, a 0 goes in the 010 cell on a 3-variable map.

When a POS expression is completely mapped, there will be a number of 0s on the Karnaugh map equal to the number of sum terms in the standard POS expression. The cells that do not have a 0 are the cells for which the expression is 1. Usually, when working with POS expressions, the 1s are left off. The following steps and the illustration in Figure 4–43 show the mapping process.

Mapping a Standard POS Expression

Step 1: Determine the binary value of each sum term in the standard POS expression. This is the binary value that makes the term equal to 0.

Step 2: As each sum term is evaluated, place a 0 on the Karnaugh map in the corresponding cell.



Mapping a Standard POS Expression

Map the following standard POS expression on a Karnaugh map:

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

Solution

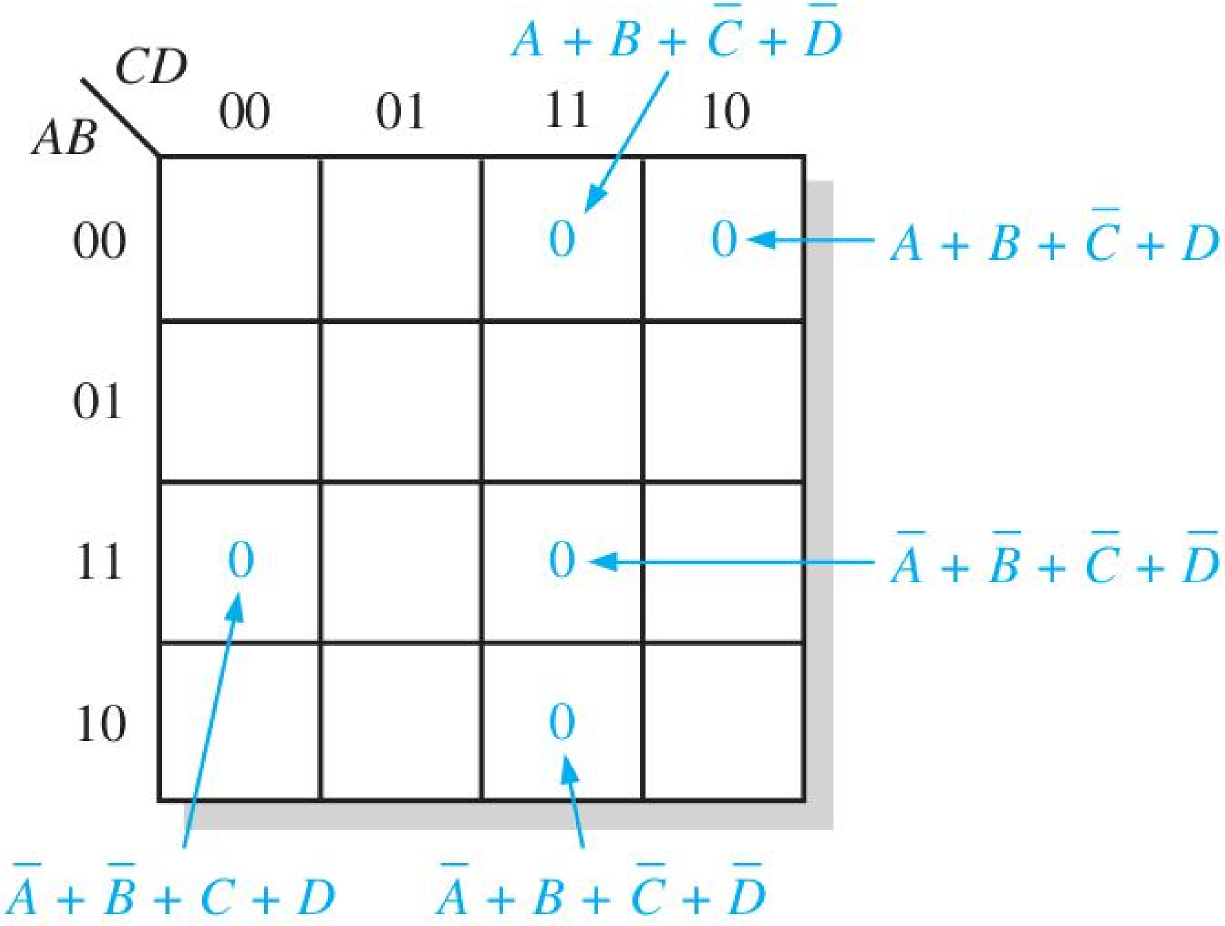
$$\underset{1100}{(\overline{A} + \overline{B} + C + D)}\underset{1011}{(\overline{A} + B + \overline{C} + \overline{D})}\underset{0010}{(A + B + \overline{C} + D)}\underset{1111}{(\overline{A} + \overline{B} + \overline{C} + \overline{D})}\underset{0011}{(A + B + \overline{C} + \overline{D})}$$

Map the following standard POS expression on a Karnaugh map:

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

Solution

$$\underset{1100}{(\overline{A} + \overline{B} + C + D)}\underset{1011}{(\overline{A} + B + \overline{C} + \overline{D})}\underset{0010}{(A + B + \overline{C} + D)}\underset{1111}{(\overline{A} + \overline{B} + \overline{C} + \overline{D})}\underset{0011}{(A + B + \overline{C} + \overline{D})}$$



Mapping a Standard POS Expression

Use a Karnaugh map to minimize the following POS expression:

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

के Standard फर्म मा छ ?

The first term must be expanded into $\overline{A} + B + C + D$ and $A + B + C + D$ to get a standard POS expression,

$$\prod M(0,2,4,12,8,9)$$

$\prod M(0,2,4,12,8,9)$

C+D

$\therefore (C+D)(A+B+D)(\bar{A}+B+C)$

AB \ CD	00	01	11	10
00	0 0	1	3	0 2
01	0 4	5	7	6
11	0 12	13	15	14
10	0 8	0 9	11	10

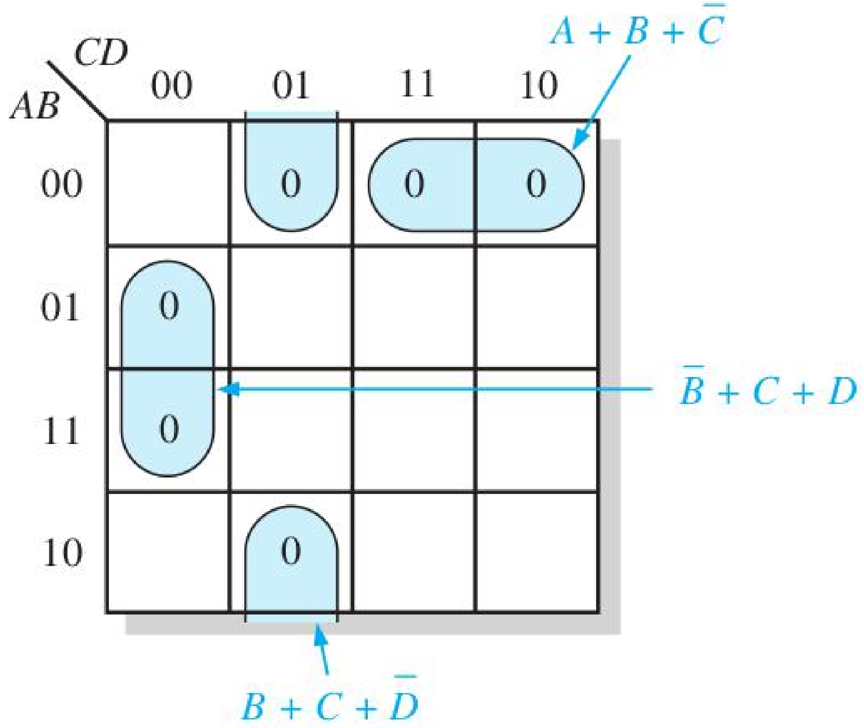
A+B+D

$\bar{A}+B+C$

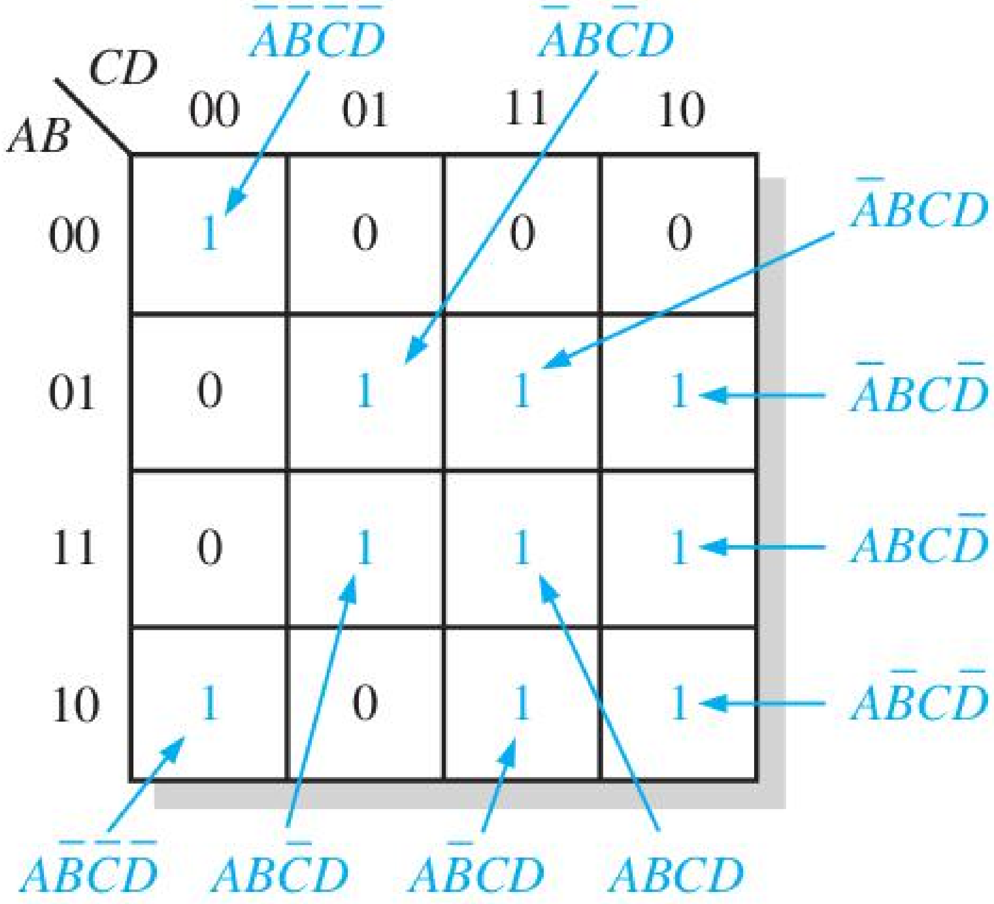
Converting Between POS and SOP Using the Karnaugh Map

Using a Karnaugh map, convert the following standard POS expression into a minimum POS expression, a standard SOP expression, and a minimum SOP expression.

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



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

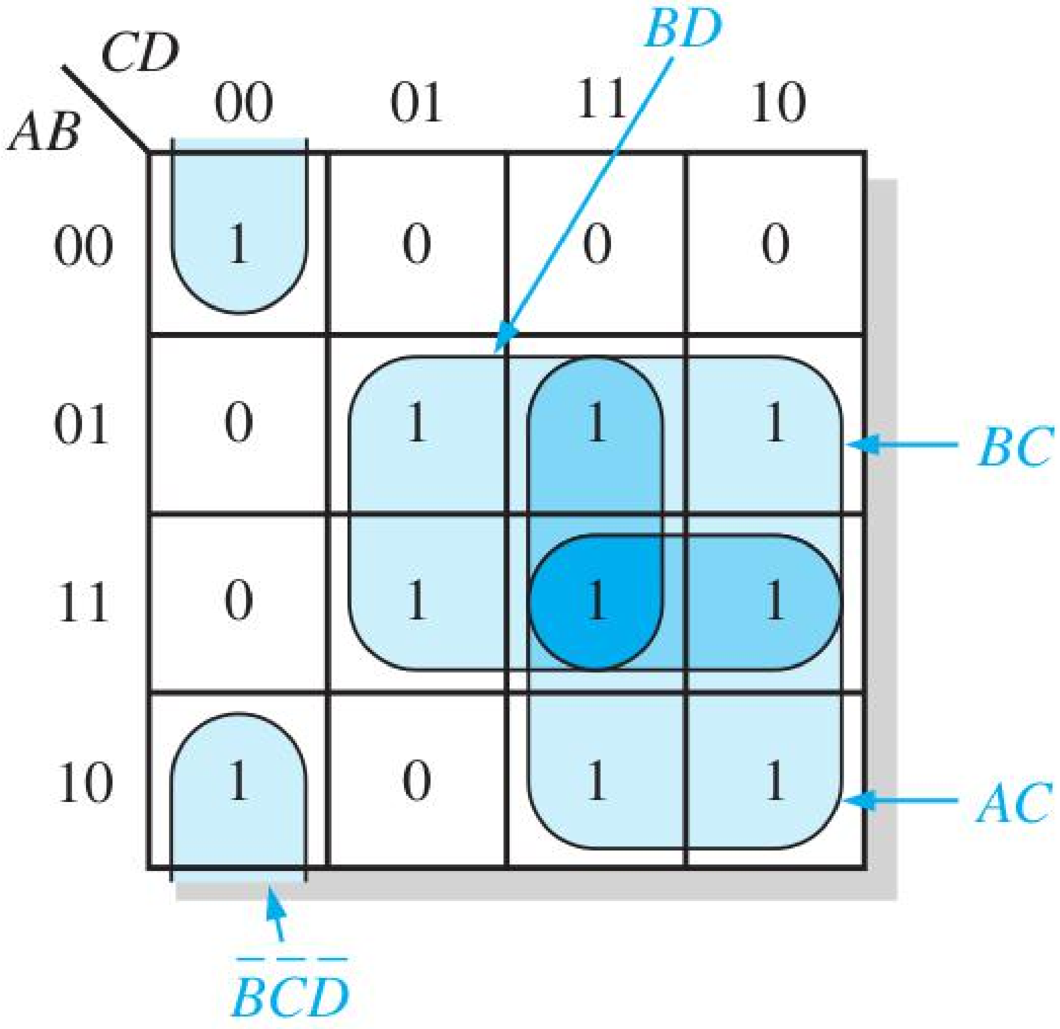


(b) Standard SOP:

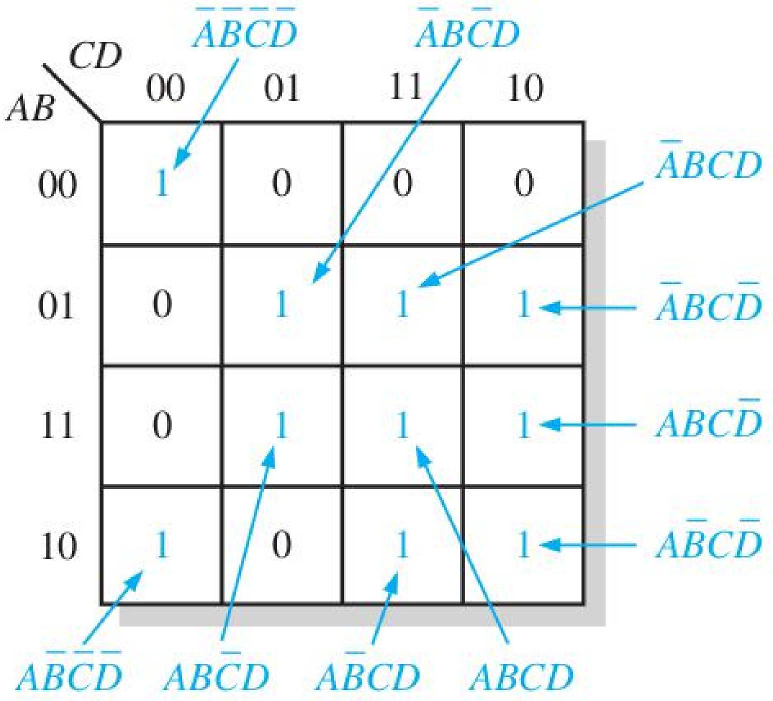
$$\overline{A}\overline{B}\overline{C}\overline{D} + \overline{A}B\overline{C}D + \overline{A}BCD + \overline{A}BC\overline{D} + ABC\overline{D} + A\overline{B}C\overline{D} + A\overline{B}\overline{C}\overline{D} + AB\overline{C}D + A\overline{B}CD + ABCD$$

Using a Karnaugh map, convert the following standard POS expression into a minimum POS expression, a standard SOP expression, and a minimum SOP expression.

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



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



(b) Standard SOP:
$\overline{A}\overline{B}\overline{C}\overline{D} + \overline{A}B\overline{C}D + \overline{A}BCD + \overline{A}BC\overline{D} + ABC\overline{D} + A\overline{B}C\overline{D} +$
$A\overline{B}\overline{C}\overline{D} + AB\overline{C}D + A\overline{B}CD + ABCD$

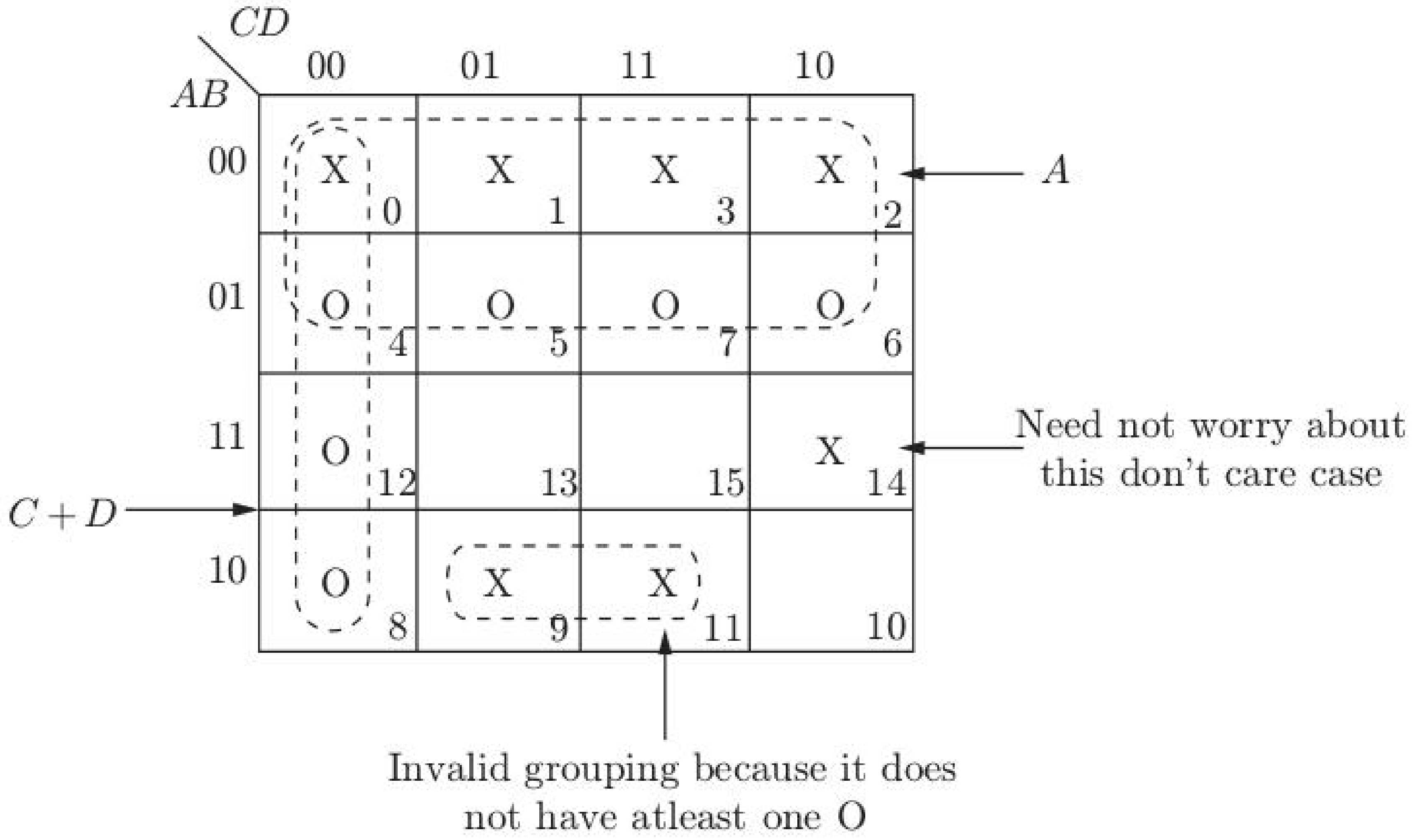
AB \ CD	00	01	11	10
00		1		
01		1		1
11	1	1	1	1
10	1	1	1	1

GET MINIMUM EXPRESSION BASED ON PROVIDED k-Map

ASSIGNMENT

Simplify $F(A, B, C, D) = \Pi M(4, 5, 6, 7, 8, 12) \cdot d(0, 1, 2, 3, 9, 11, 14)$

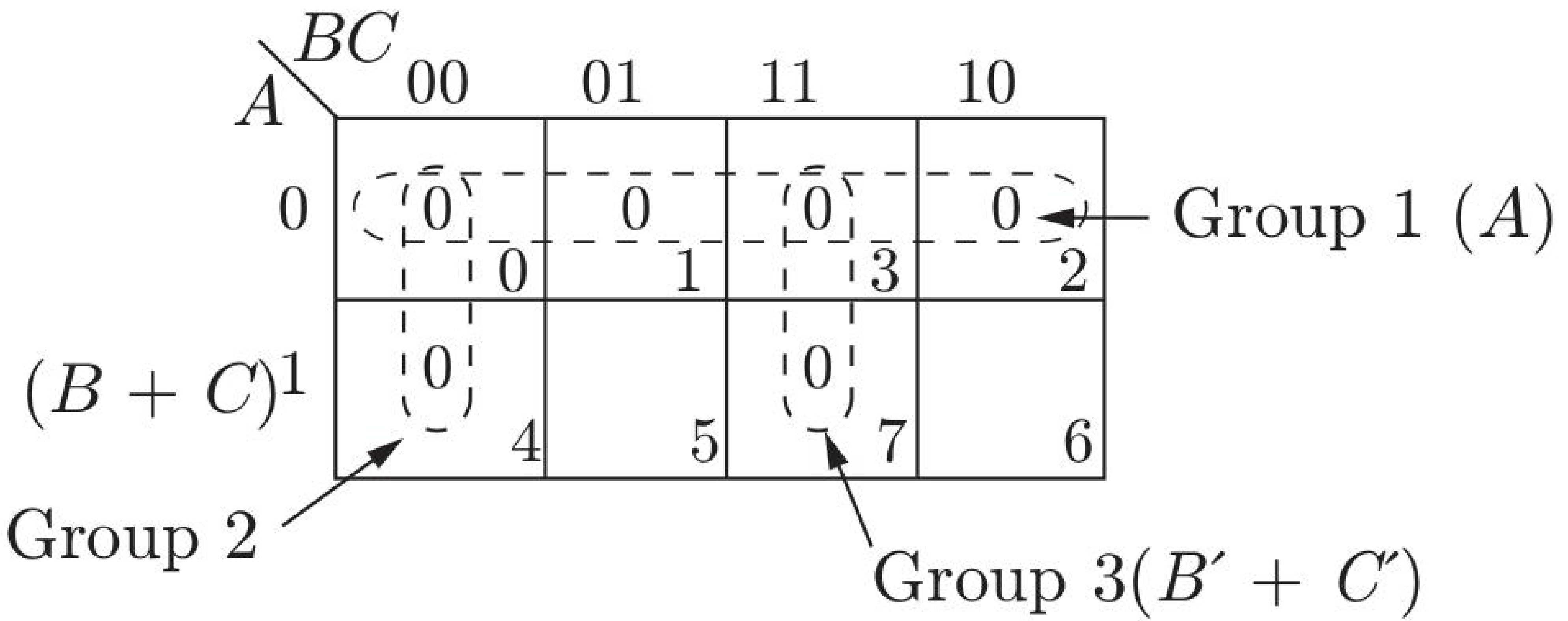
Solution: A logic circuit output may be either 0 or 1 but some circuit has output that will never occur for certain combinations of inputs. These conditions are don't care conditions and represented by X in K-map.



The simplified Boolean expression is $F = A(C + D)$.

Simplify $S = \Pi M(0,1,2,3,4,7)$

Solution:



The simplified Boolean expression is $S = A \cdot (B + C) \cdot (B' + C')$