Digital Design

CS302

Ref: Digital Design. M. Morris Mano , and Michael D. Ciletti. Pearson, FIFTH EDITION, 2013

Gate-Level Minimization

CHAPTER 3 (PART 1)

Introduction

Gate-level minimization : finding an optimal gate-level implementation of the Boolean functions describing a digital circuit.

Remember

Table 2.3
Minterms and Maxterms for Three Binary Variables

			Minterms		Maxterms	
x	**y**	**z**	**Term**	**Designation**	**Term**	**Designation**
0	0	0	$x'y'z'$	m_0	$x + y + z$	M_0
0	0	1	$x'y'z$	m_1	$x + y + z'$	M_1
0	1	0	$x'yz'$	m_2	$x + y' + z$	M_2
0	1	1	$x'yz$	m_3	$x + y' + z'$	M_3
1	0	0	$xy'z'$	m_4	$x' + y + z$	M_4
1	0	1	$xy'z$	m_5	$x' + y + z'$	M_5
1	1	0	xyz'	m_6	$x' + y' + z$	M_6
1	1	1	xyz	m_7	$x' + y' + z'$	M_7

- $M_j = m'_j$

3.2 THE MAP METHOD

- The map method provides a simple, straightforward procedure for minimizing Boolean functions.

- **The map method is also known as the *Karnaugh map* *or* *K-map*.**

- A **K-map** is a diagram made up of squares, with each square representing one minterm of the function that is to be minimized.

- **The simplest expression is not unique**: It is sometimes possible to find two or more expressions that satisfy the minimization criteria.

Two-Variable K-Map

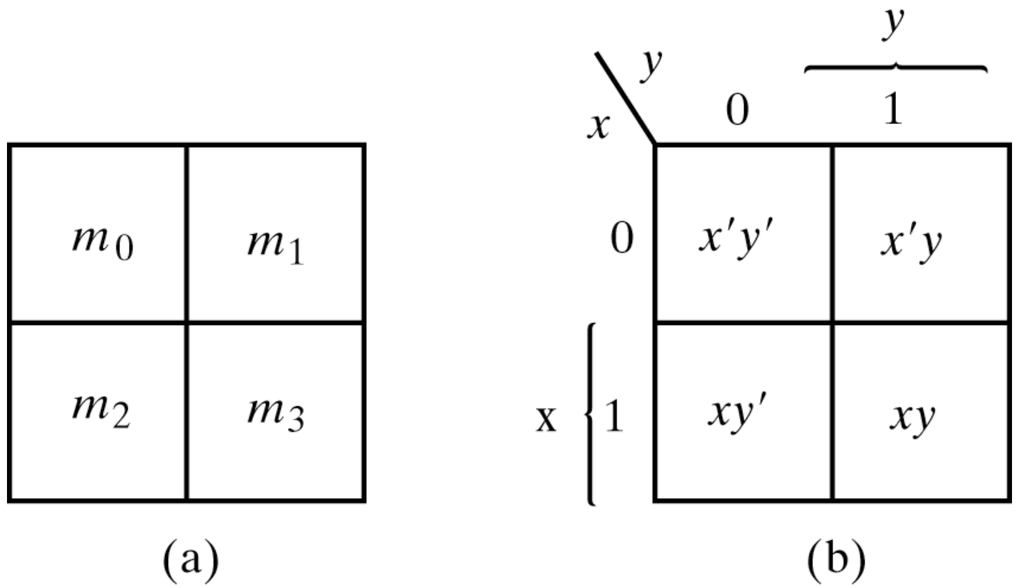


Fig. 3-1 Two-variable Map

One square → one minterm with two literals.
Two adjacent squares → a term with one literal.
Four adjacent squares → a function that is always equal to 1.

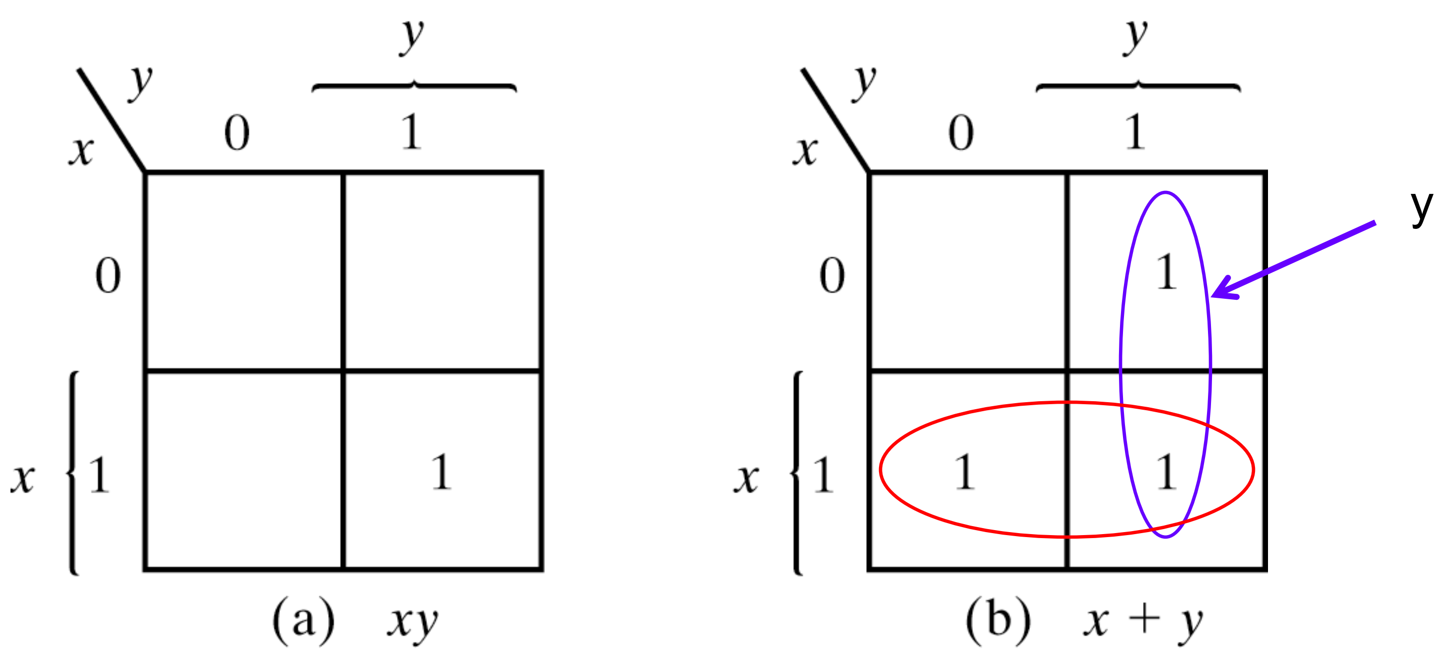


Fig. 3-2 Representation of Functions in the Map

Three-Variable K-Map

The combination of adjacent squares that is useful is determined from the three-variable map:

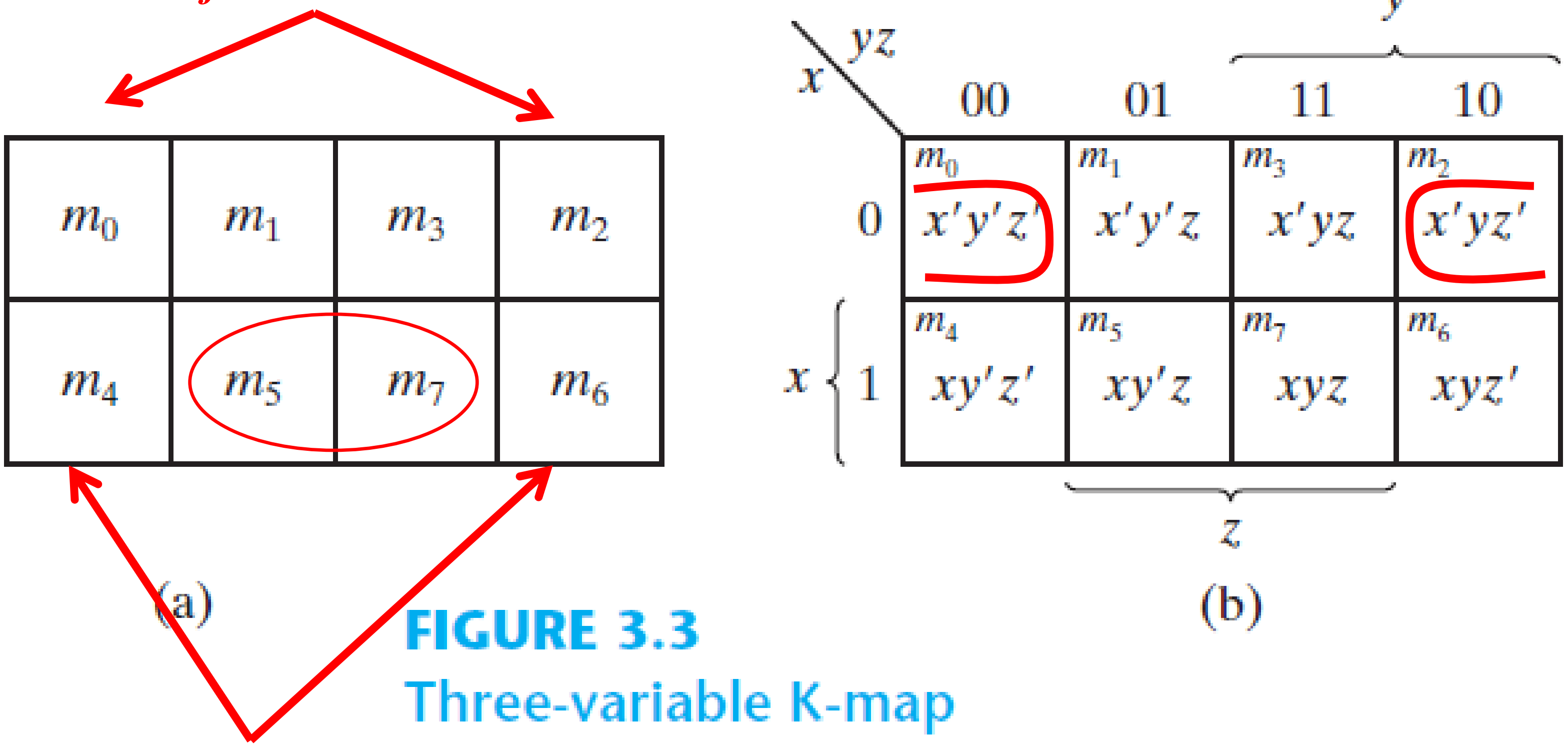
- One square $\rightarrow$ one minterm with three literals.
- Two adjacent squares $\rightarrow$ a term with two literals.
- Four adjacent squares $\rightarrow$ a term with one literal.
- Eight adjacent squares $\rightarrow$ a function that is always equal to 1.

The characteristic of this sequence is that only one bit changes in value from one adjacent column to the next.

Any two adjacent squares in the map differ by only one variable.

e.g. $m_5 + m_7 = xy'z + xyz = xz\,(y + y') = xz$

m_0 is adjacent to m_2



m_4 is adjacent to m_6

FIGURE 3.3
Three-variable K-map

EXAMPLE 3.1

Simplify the Boolean function

$F(x, y, z) = \Sigma(2, 3, 4, 5)$

EXAMPLE 3.1

Simplify the Boolean function

$F(x, y, z) = \Sigma(2, 3, 4, 5)$

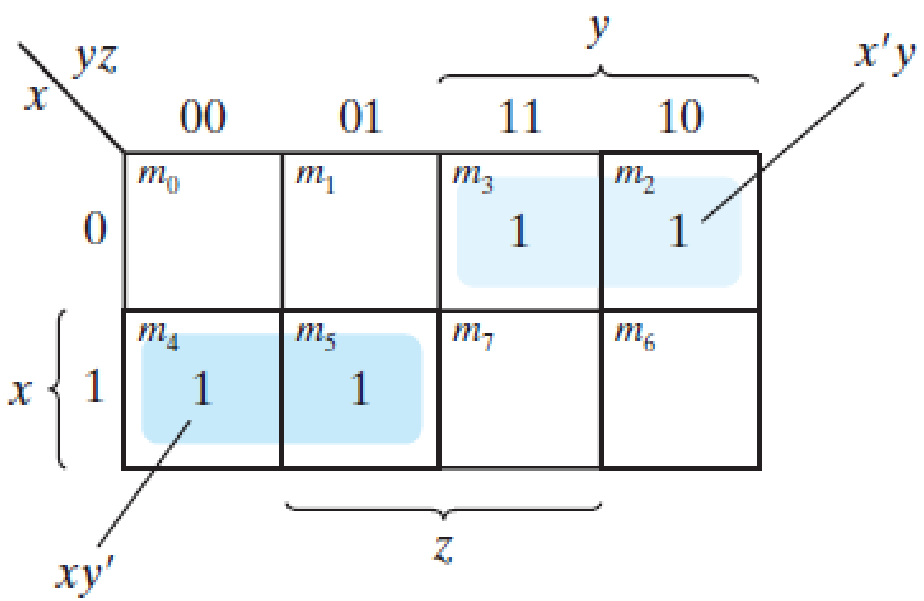


FIGURE 3.4

$F(x, y, z) = \Sigma(2, 3, 4, 5) = x'y + xy'$

EXAMPLE 3.2

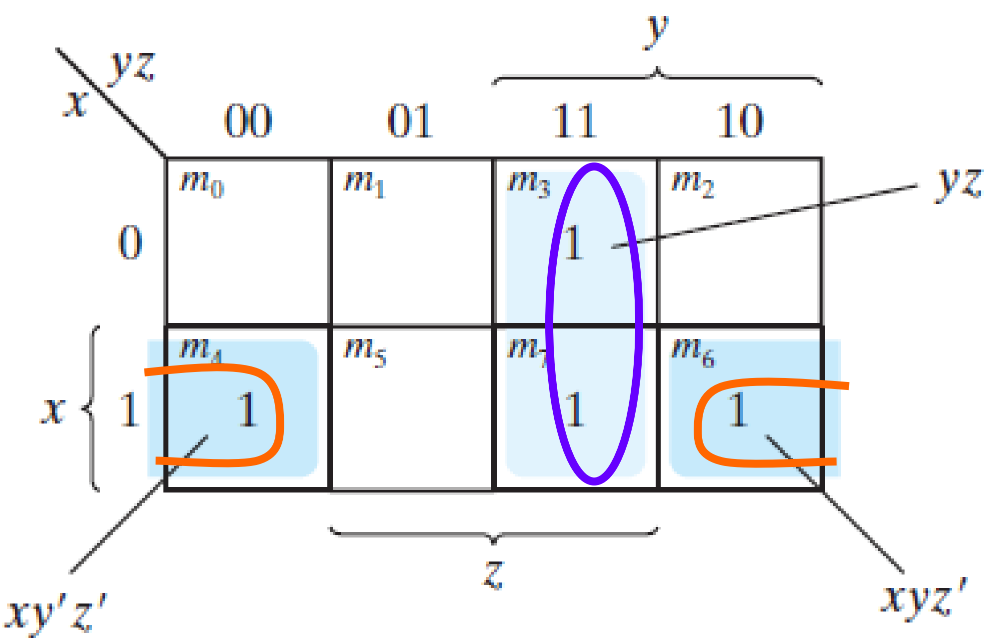
Simplify the Boolean function

$F(x, y, z) = \Sigma(3, 4, 6, 7)$

EXAMPLE 3.2

Simplify the Boolean function

$F(x, y, z) = \Sigma(3, 4, 6, 7)$



Note: $xy'z' + xyz' = xz'$

FIGURE 3.5
Map for Example 3.2, $F(x, y, z) = \Sigma(3, 4, 6, 7) = yz + xz'$

EXAMPLE 3.3

Simplify the Boolean function

$$F(x, y, z) = \Sigma(0, 2, 4, 5, 6)$$

EXAMPLE 3.3

Simplify the Boolean function

$$F(x, y, z) = \Sigma(0, 2, 4, 5, 6)$$

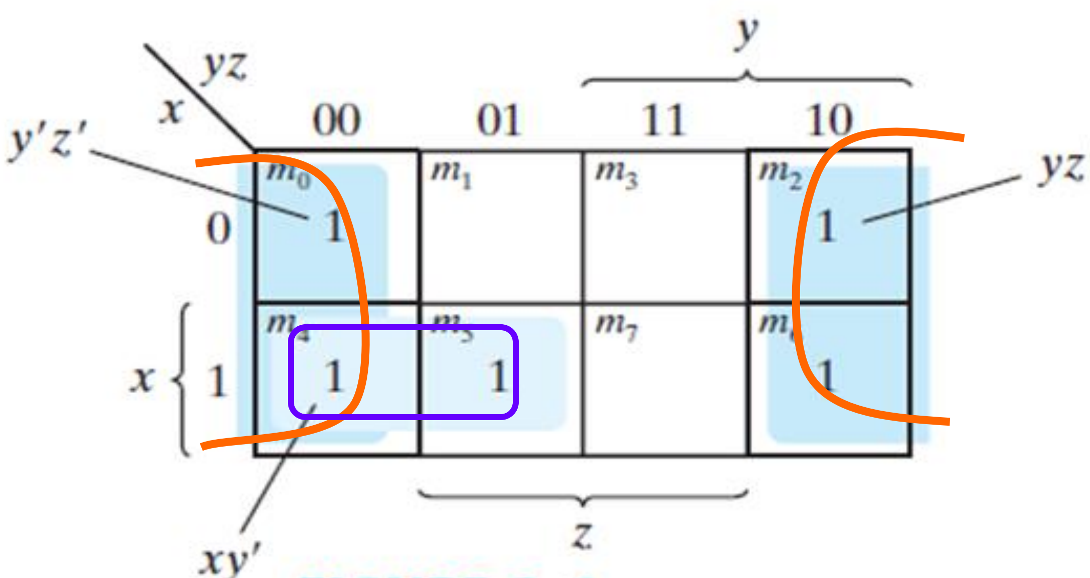


FIGURE 3.6
$F(x, y, z) = z' + xy'$

$m_0 + m_2 + m_4 + m_5 + m_6 = x'y'z' + x'yz' + xy'z' + xy'z + xyz'$

EXAMPLE 3.4

For the Boolean function

$F = A'C + A'B + AB'C + BC$

(a) Express this function as a sum of minterms.

(b) Find the minimal sum-of-products expression.

EXAMPLE 3.4

For the Boolean function

$F = A'C + A'B + AB'C + BC$

(a) Express this function as a sum of minterms.

(b) Find the minimal sum-of-products expression.

(a) $F(A, B, C) = \Sigma(1, 2, 3, 5, 7)$

A \ BC	00	01	11	10
0		1	1	1
1		1	1	

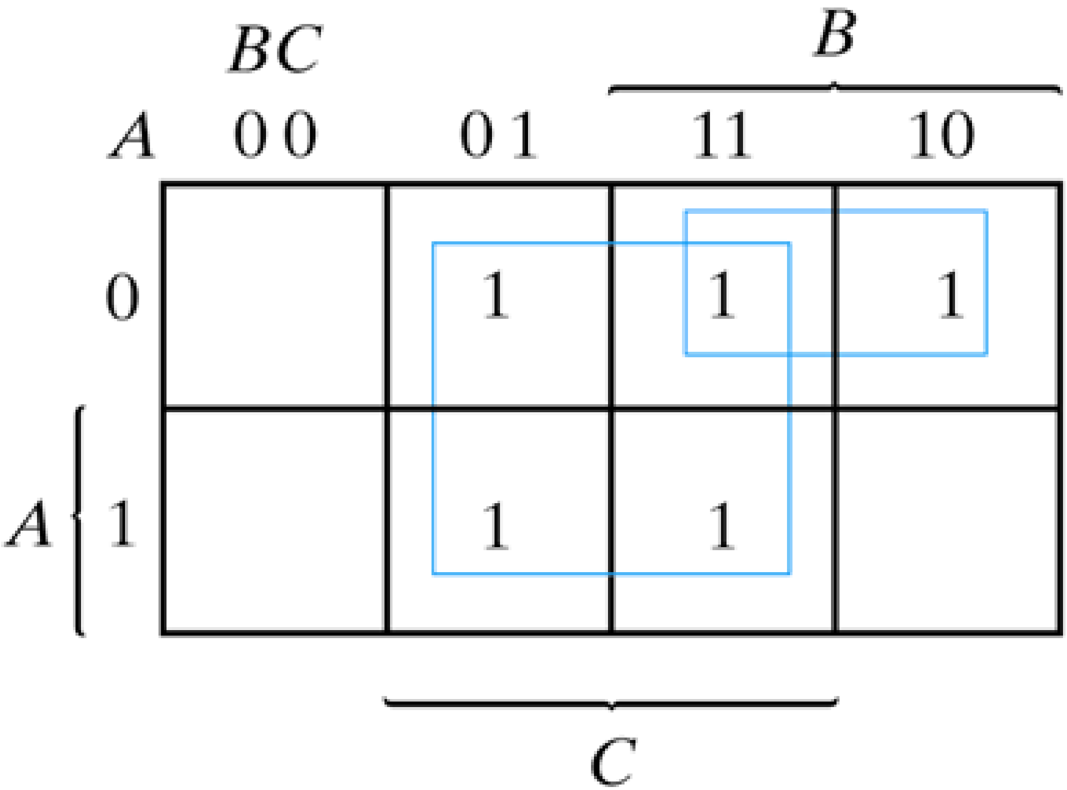


Fig. 3-7 Map for Example 3-4;

$$A'C + A'B + AB'C + BC = C + A'B$$

3.3 Four-Variable K-Map

The combination of adjacent squares that is useful during the simplification process is determined from the four-variable map:

- One square $\rightarrow$ one minterm with four literals.
- Two adjacent squares $\rightarrow$ a term with three literals.
- Four adjacent squares $\rightarrow$ a term with two literals.
- Eight adjacent squares $\rightarrow$ a term with one literal.
- Sixteen adjacent squares $\rightarrow$ a function that is always equal to 1.

3.3 Four-Variable K-Map

m_0	m_1	m_3	m_2
m_4	m_5	m_7	m_6
m_{12}	m_{13}	m_{15}	m_{14}
m_8	m_9	m_{11}	m_{10}

(a)

y

$wx \backslash yz$	00	01	11	10
00	m_0 $w'x'y'z'$	m_1 $w'x'y'z$	m_3 $w'x'yz$	m_2 $w'x'yz'$
01	m_4 $w'xy'z'$	m_5 $w'xy'z$	m_7 $w'xyz$	m_6 $w'xyz'$
11	m_{12} $wxy'z'$	m_{13} $wxy'z$	m_{15} $wxyz$	m_{14} $wxyz'$
10	m_8 $wx'y'z'$	m_9 $wx'y'z$	m_{11} $wx'yz$	m_{10} $wx'yz'$

x

w

z

(b)

FIGURE 3.8
Four-variable map

EXAMPLE 3.5

Simplify the Boolean function

$$F(w, x, y, z) = \Sigma(0, 1, 2, 4, 5, 6, 8, 9, 12, 13, 14)$$

EXAMPLE 3.5

Simplify the Boolean function

$$F(w, x, y, z) = \Sigma(0, 1, 2, 4, 5, 6, 8, 9, 12, 13, 14)$$

wx \ yz	00	01	11	10
00	1	1		1
01	1	1		1
11	1	1		1
10	1	1		

m_0	m_1	m_3	m_2
m_4	m_5	m_7	m_6
m_{12}	m_{13}	m_{15}	m_{14}
m_8	m_9	m_{11}	m_{10}

$$F(w, x, y, z) = y' + xz' + w'z'$$

EXAMPLE 3.6

Simplify the Boolean function

$$F = A'B'C' + B'CD' + A'BCD' + AB'C'$$

23

EXAMPLE 3.6

Simplify the Boolean function

$$F = A'B'C' + B'CD' + A'BCD' + AB'C'$$

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

m_0	m_1	m_3	m_2
m_4	m_5	m_7	m_6
m_{12}	m_{13}	m_{15}	m_{14}
m_8	m_9	m_{11}	m_{10}

$F = B'C' + A'CD' + B'D'$

24

Prime Implicants

A **prime implicant** is a product term obtained by combining the maximum possible number of adjacent squares in the map.

If a minterm in a square is covered by only one prime implicant, that prime implicant is said to be ***essential***.

Prime Implicants

A prime implicant is a product term obtained by combining the maximum possible number of adjacent squares in the map.

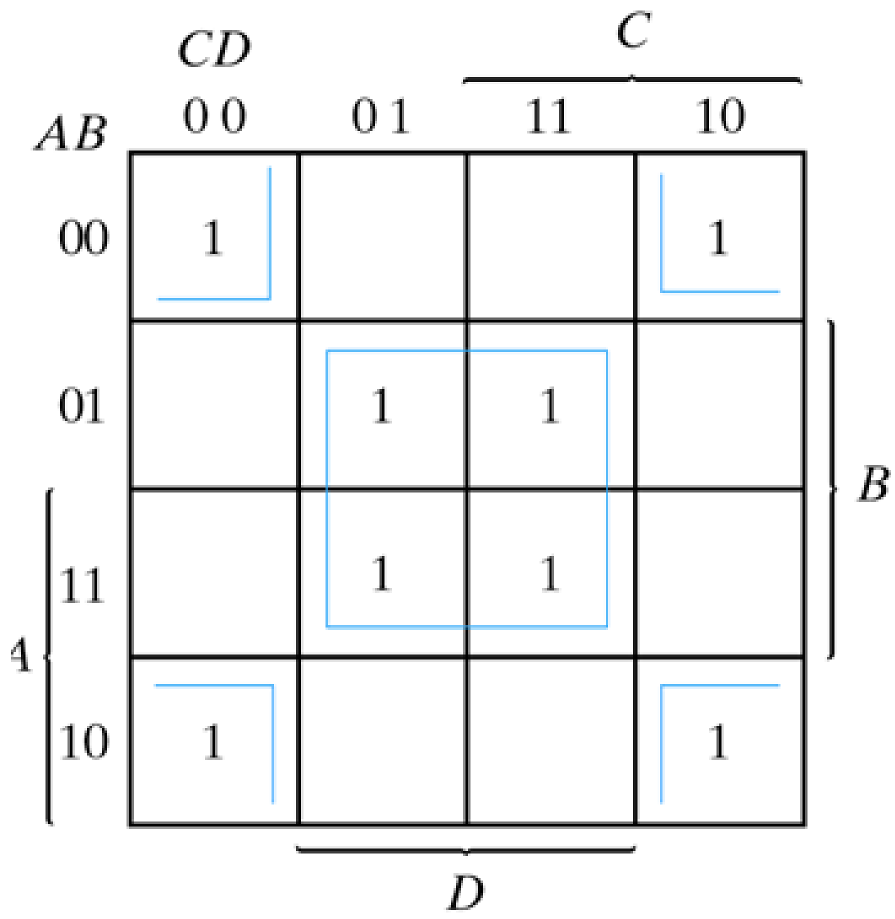
If a minterm in a square is covered by only one prime implicant, that prime implicant is said to be ***essential.***

The prime implicants of a function can be obtained from the map by combining all possible maximum numbers of squares.

The simplified expression

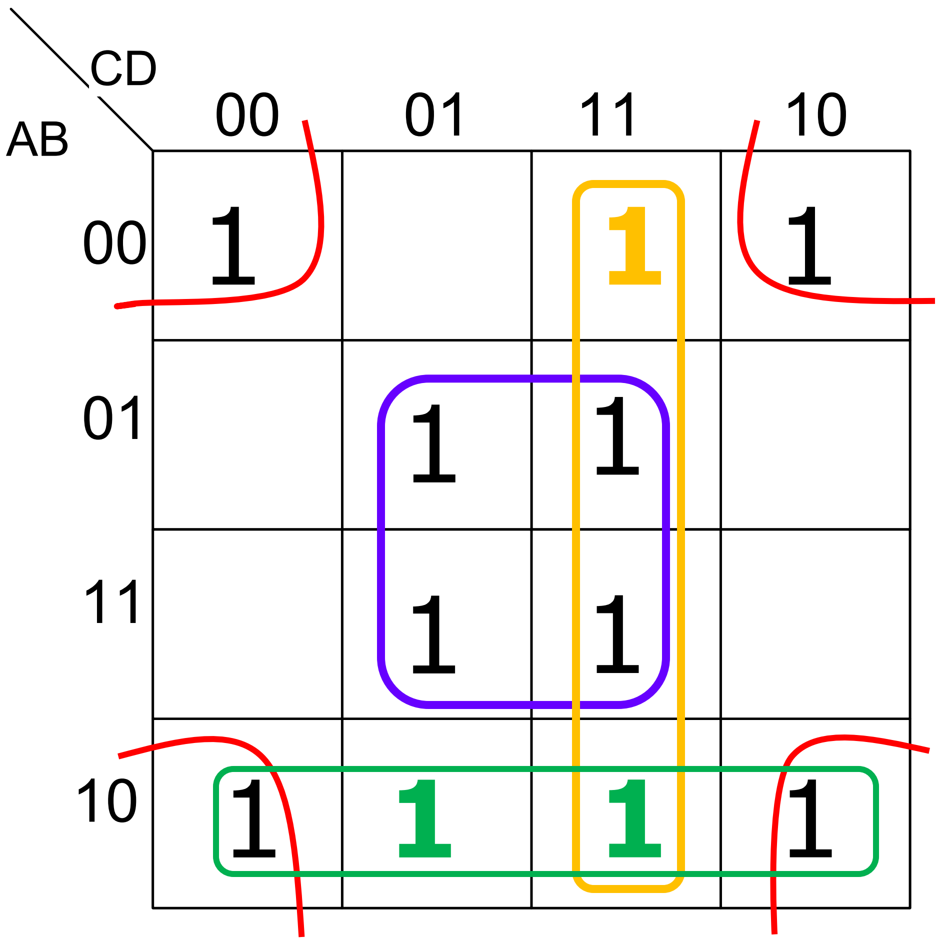
Finding the simplified expression :

1. Find all prime implicants.
2. Determine all the essential prime implicants.
3. Sum all the essential prime implicants, plus other prime that may be needed to cover any remaining minterms not covered by the essential prime implicants.



(a) Essential prime implicants
BD and B′D′

Essential *Prime Implicant* B'D', BD



B'D' +

BD +

CD +

AB'

F(A, B, C, D) = $\sum$ (0, 2, 3, 5, 7, 8, 9, 10, 11, 13, 15)

Essential *Prime Implicants* $B'D'$, BD

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

$B'D'$ +
BD +
CD +
AD

F(A, B, C, D) = $\sum$ (0, 2, 3, 5, 7, 8, 9, 10, 11, 13, 15)

There are four possible ways that the function can be expressed with four product terms of two literals each:

$$\begin{aligned} F &= BD + B'D' + CD + AD \\ &= BD + B'D' + CD + AB' \\ &= BD + B'D' + B'C + AD \\ &= BD + B'D' + B'C + AB' \end{aligned}$$

3.4 PRODUCT-OF-SUMS SIMPLIFICATION

The complement of a function is represented in the map by the squares not marked by 1’s.

If we mark the empty squares by 0’s and combine them into valid adjacent squares, we obtain a simplified sum-of-products expression of the complement of the function (i.e., of F’).

The complement of F’ gives us back the Function F in product-of-sums form (a consequence of DeMorgan’s theorem).

EXAMPLE 3.7

Simplify the following Boolean function into

(a) sum-of-products form and

(b) product-of-sums form:

$$F(A, B, C, D) = \Sigma(0, 1, 2, 5, 8, 9, 10)$$

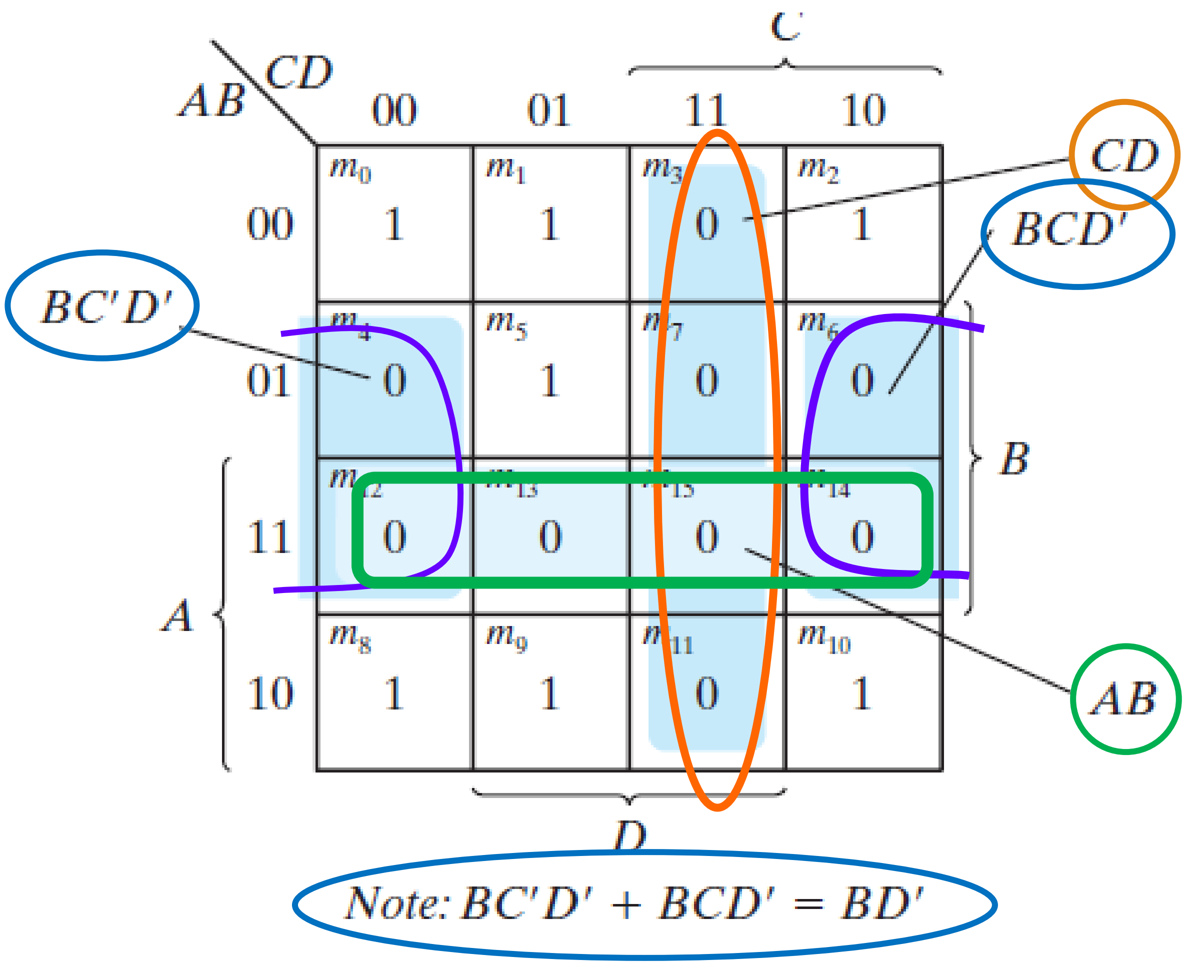
a) sum-of-products form

$$F = B'D' + B'C' + A'C'D$$

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

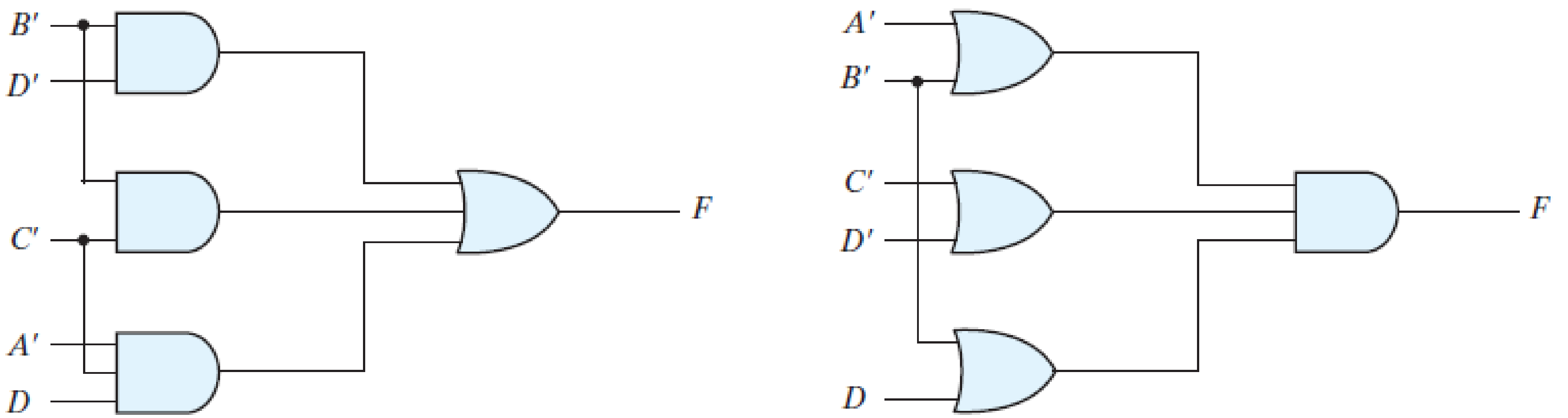
Figure 3.12

b) Product-of-sums form



Consider the squares marked with 0's:

$F' = AB + CD + BD'$ $\xrightarrow{\text{DeMorgan's}}$ $F = (A' + B')(C' + D')(B' + D)$



(a) $F = B'D' + B'C' + A'C'D$

(b) $F = (A' + B')(C' + D')(B' + D)$

FIGURE 3.13
Gate implementations of the function of Example 3.7

Example:

Consider the product-of-sums:

$F = (A' + B' + C')(B + D)$

F can be entered into the map by first taking its complement, namely,

$F' = ABC + B'D'$

and then marking 0's in the squares representing the minterms of F'. The remaining squares are marked with 1's.

3.5 DON’T-CARE CONDITIONS

In practice, in some applications the function is not specified for certain combinations of the variables.

the unspecified minterms of a function *are called* ***don’t-care conditions*** *.*

To distinguish the don’t-care condition from 1’s and 0’s, an **X** is used.

The don’t-care minterms may be assumed to be either 0 or 1.

A *don’t-care minterm* is a combination of variables whose logical value is not specified.

EXAMPLE 3.8

Simplify the Boolean function

$F(w, x, y, z) = \Sigma(1, 3, 7, 11, 15)$

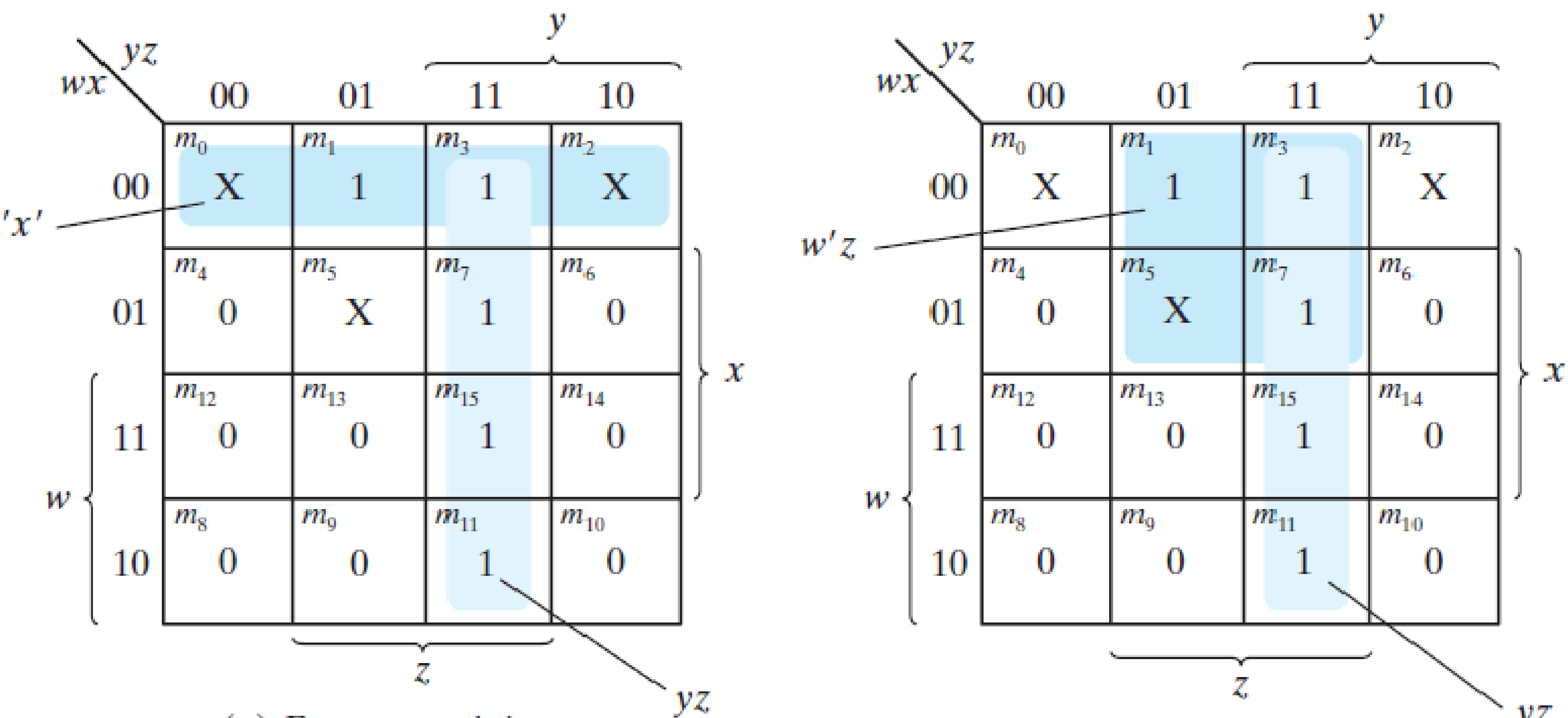
which has the don’t-care conditions

$d(w, x, y, z) = (0, 2, 5)$

into

(a) **sum-of-products** form and

(b) **product-of-sums** form:

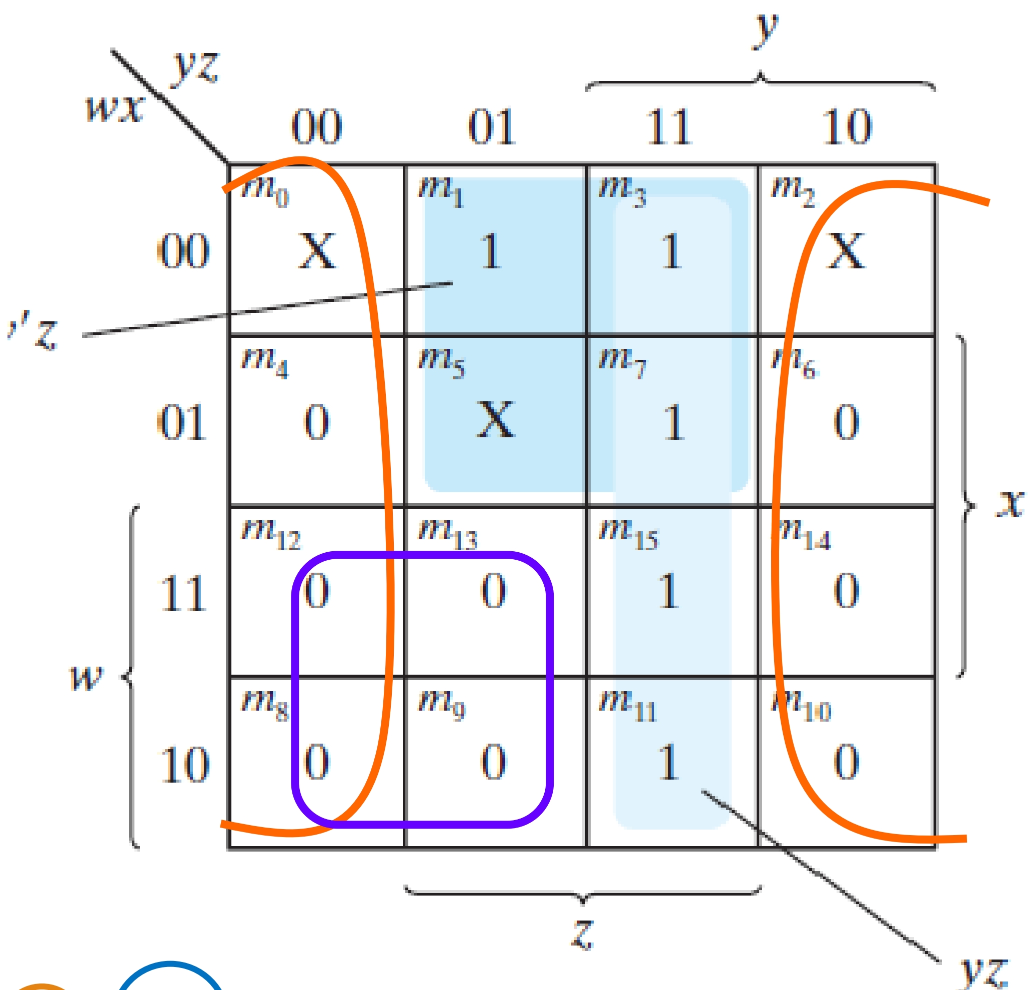


(a) $F = yz + w'x'$

(b) $F = yz + w'z$

FIGURE 3.15
Example with don't-care conditions

To obtain a simplified product-of-sums expression



$F' = z' + wy'$

$F(w, x, y, z) = z(w' + y)$

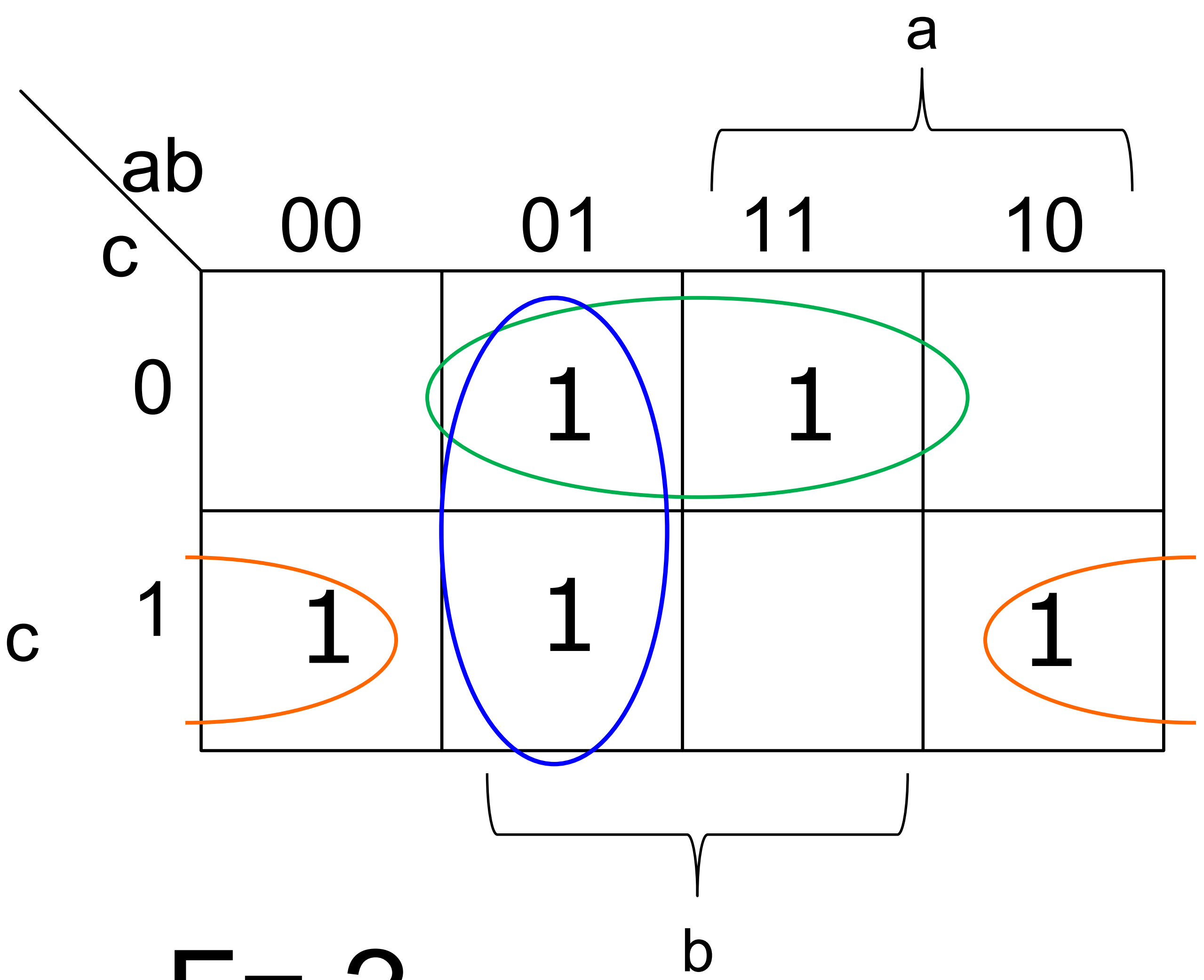
Rules for the derivation of minimal expressions:

1- Begin with cells that are adjacent to no other cells. The minterms in these cells cannot be shortened.

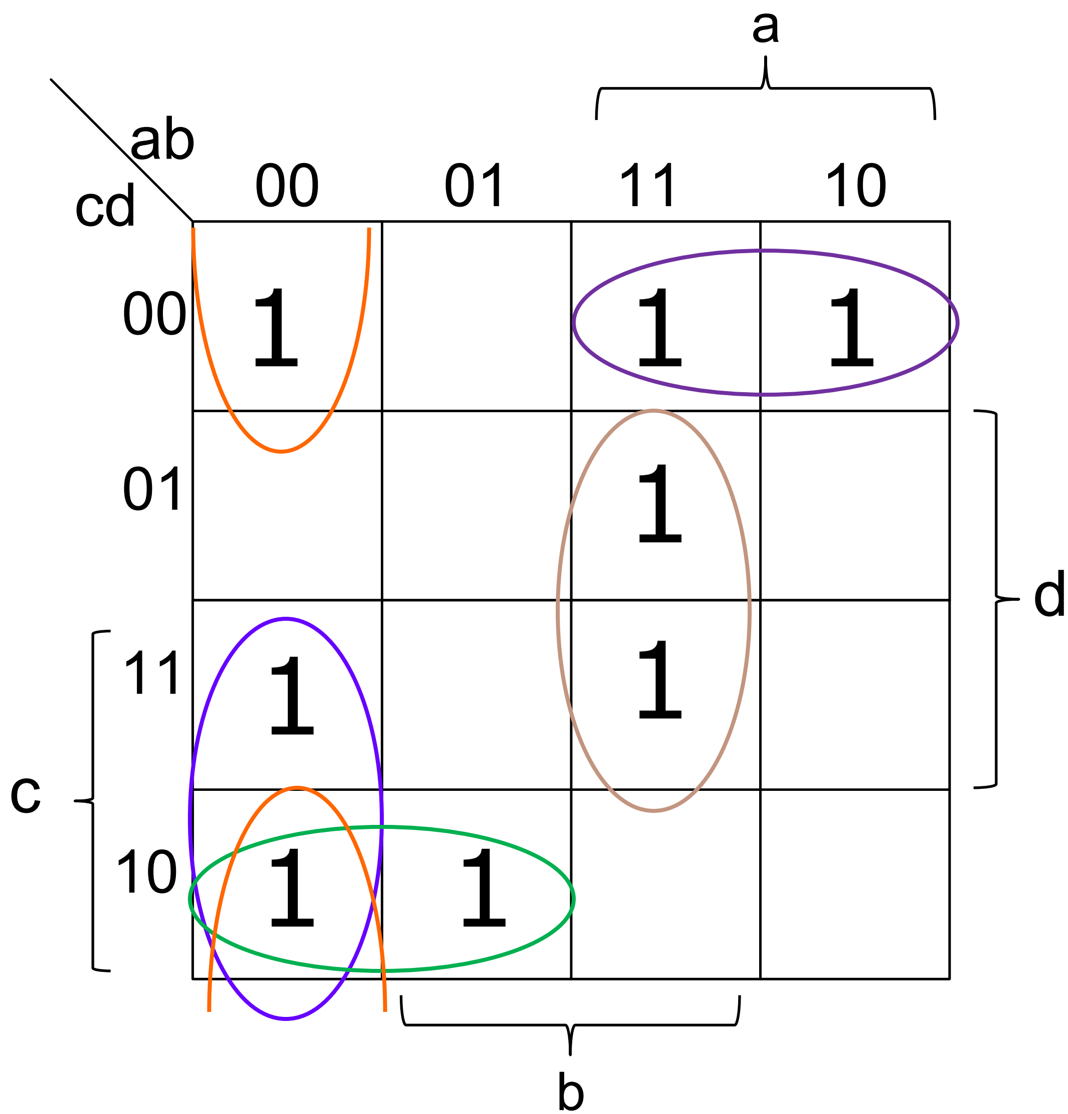
2- Find all cells that are adjacent to only one other cell these form is subcubes of two cells.

3- Find those cells that lead to maximal subcubes of four cells. Then find subcubes of 8 cells, and so on

4-The minimal expression is formed from a collection of as few cubes as possible, each of which is a maximal subcube.

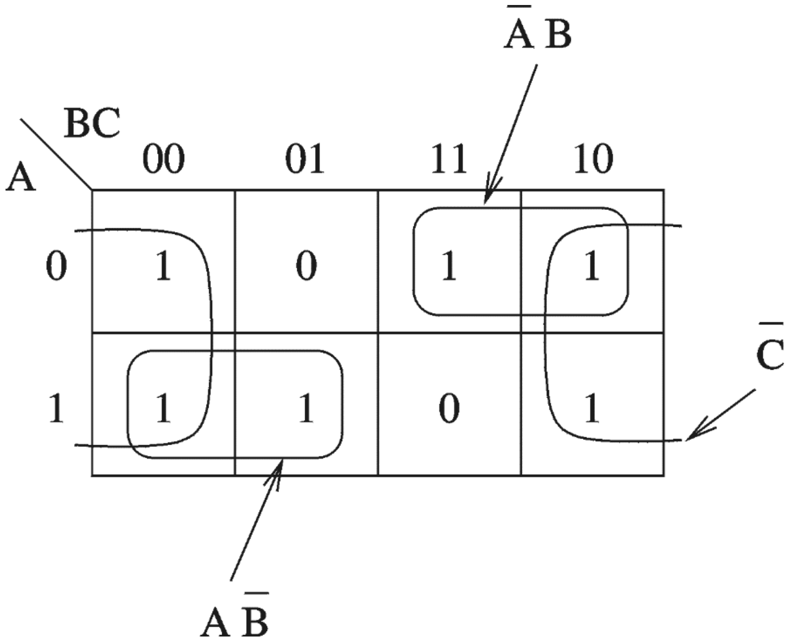


F= ?



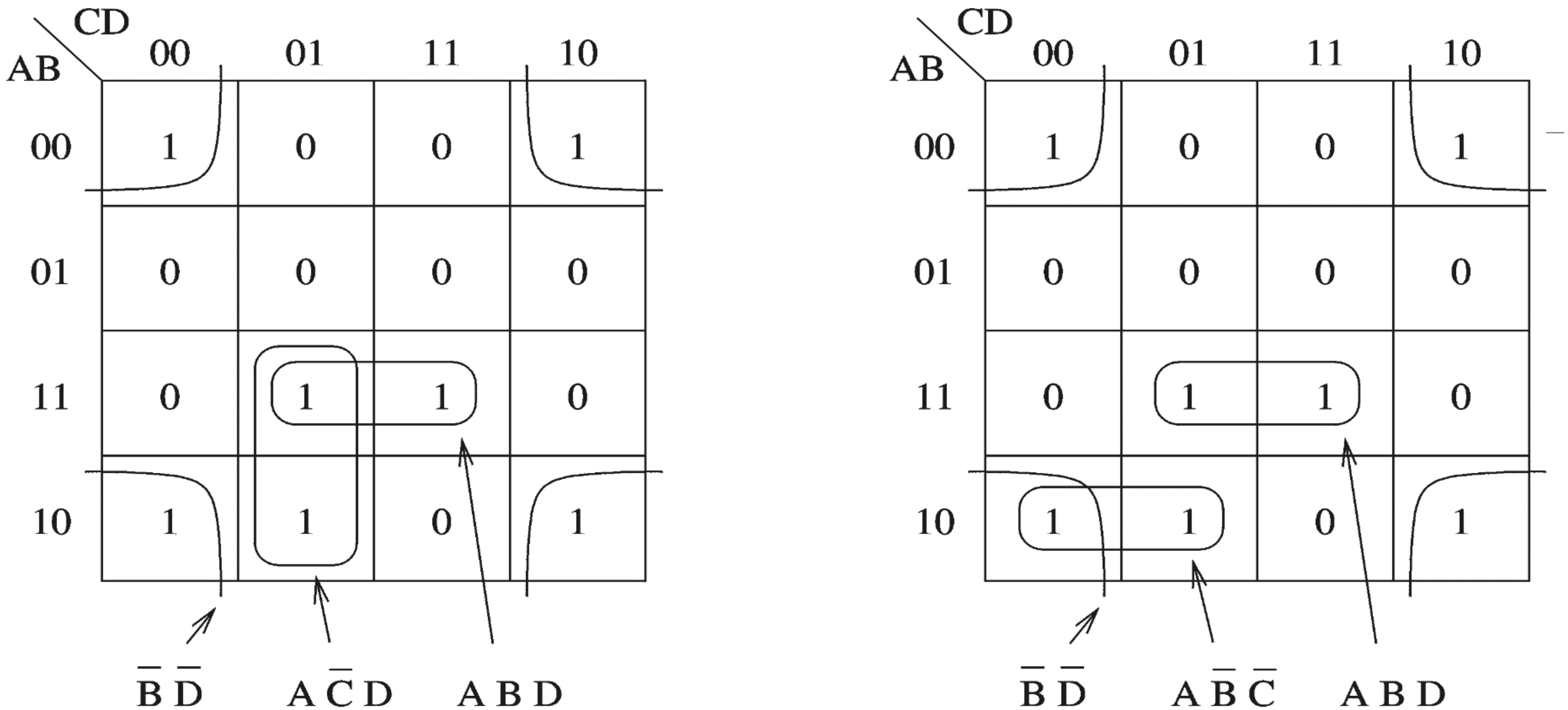
$$F = \overline{a}\,\overline{b}\,\overline{d} + \overline{a}\,\overline{b}c + \overline{a}c\overline{d} + ab\overline{d} + a\overline{c}\,\overline{d}$$

Quiz:

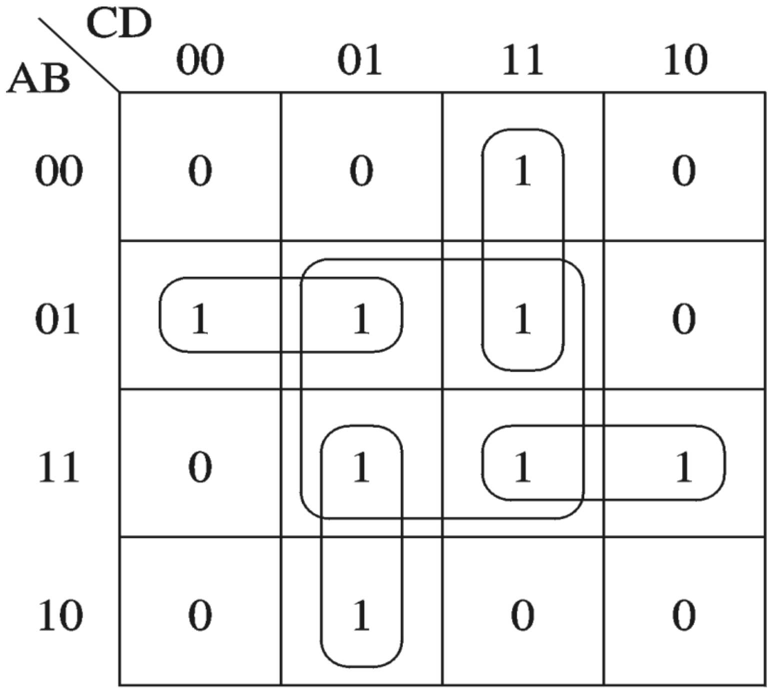


Find the function in terms:

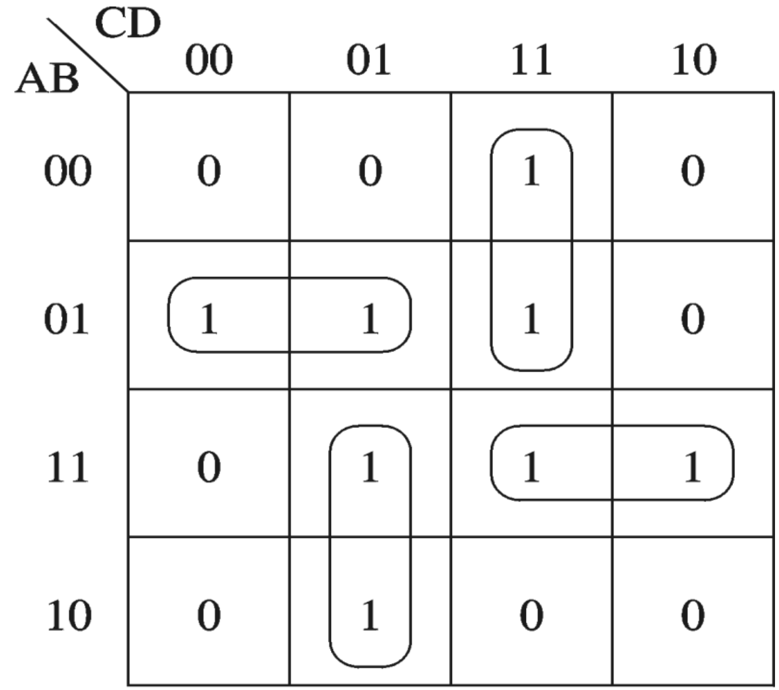
a) Minterms
b) Sum of products
c) Product of sums
d) Prime implicants
e) Essential prime implicants
f) Minimal expressions



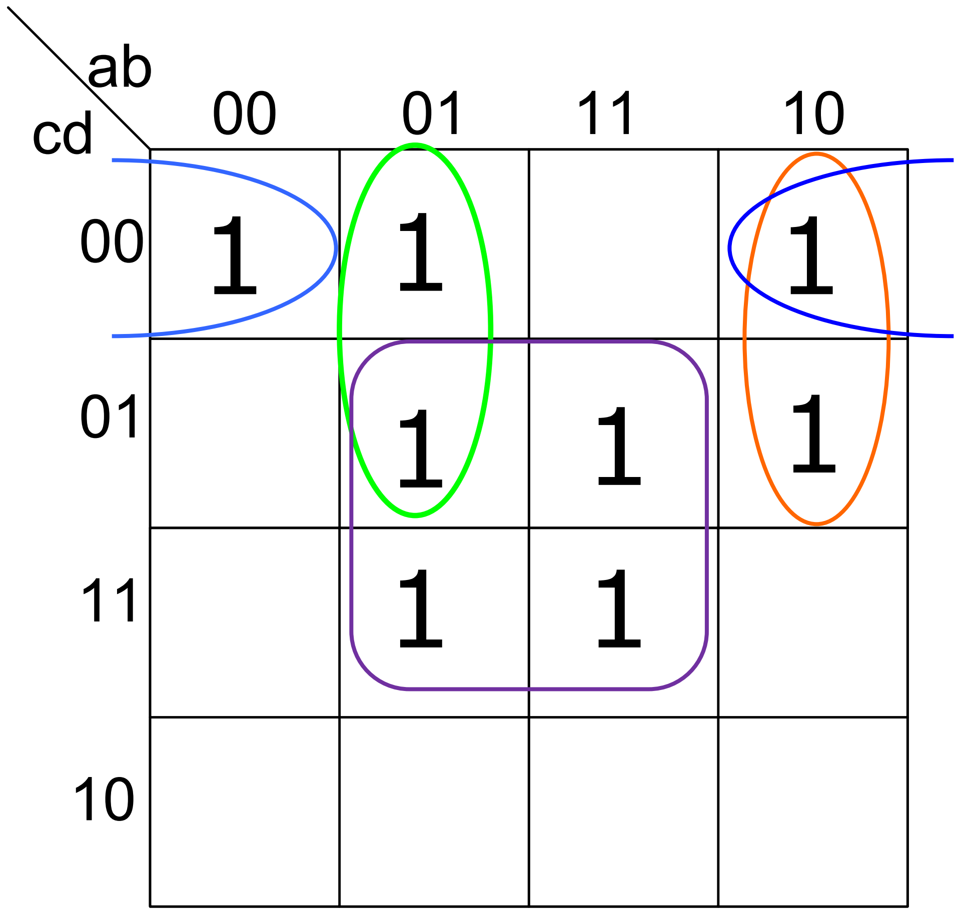
(a) (b)



(a) Nonminimal simplification



(b) Minimal simplification

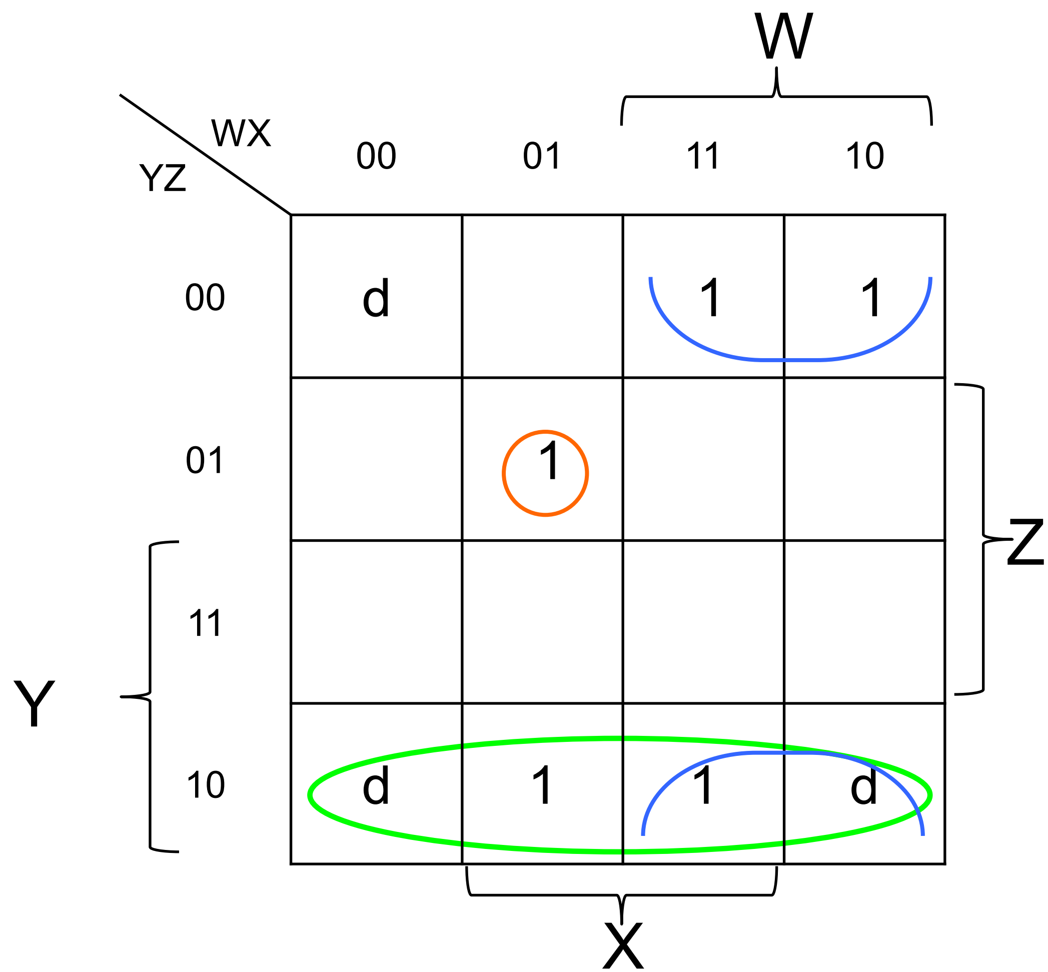


4 maximal subcubes(nonminimal simplification)

cd \ ab	00	01	11	10
00	1	1		1
01		1	1	1
11		1	1	
10				

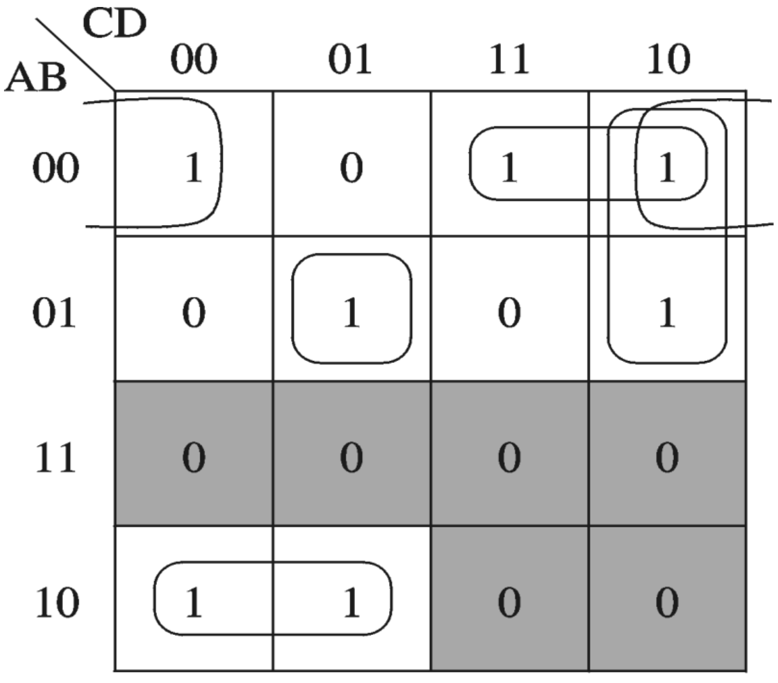
3 maximal subcubes

Don’t cares may simplify the expression

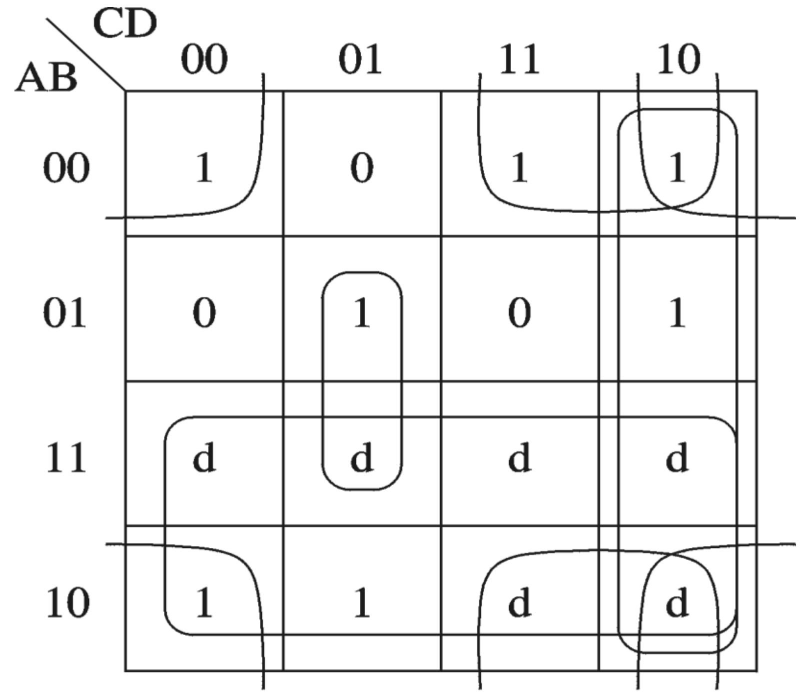


F=Y’WZ’+XYZ’+XY’ZW’

F=WZ’+YZ’+XY’ZW’



(a) Simplification with no don't cares



(b) Simplification with don't cares

F= AC'D'+A'B'C+A'CD'+A'B'D'+A'BC'D

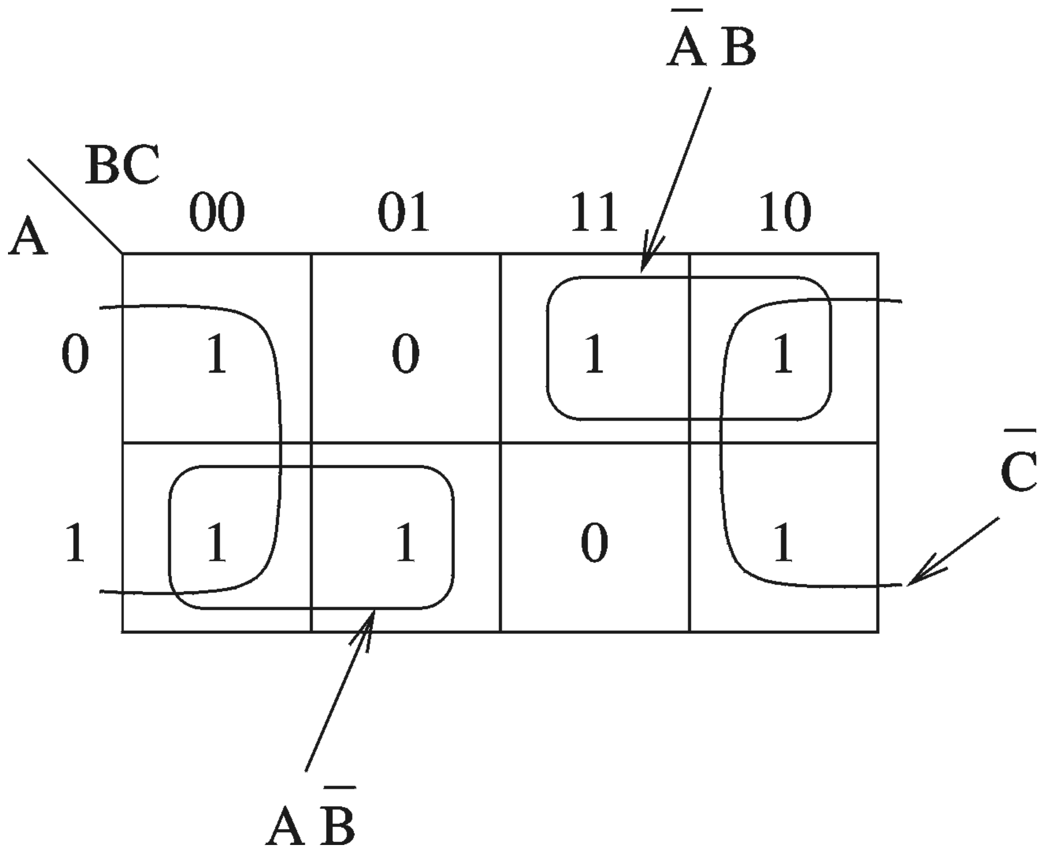
F= A+B'D'+B'C+CD'+BC'D

Subcubes and Covering

- Subcube : set of exactly 2^m adjacent cells containing 1's

m	subcube	
0	$2^0 = 1$ cells	
1	$2^1 = 2$ cells	Pair
2	$2^2 = 4$ cells	Quad
3	$2^3 = 8$ cells	Octet

Using Zeros



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