CS 221
Logic Design

Fall 2021

By Wessam El-Behaidy & Salwa Osama

K-MAP

Lecture 4

BOOLEAN FUNCTION SIMPLIFICATION

Algebraic method

- Boolean Algebra
- **Disadvantage**
  - It lacks specific rules to predict each succeeding step in the manipulative process
- **Advantage**
  - For complex boolean functions

Graphical method

- Karnaugh Map (k-map)
- **Advantage**
  - Simple straightforward procedure
- **Disadvantage**
  - Functions of up to five variables by the map

KARNAUGH MAP

- Is a diagram made up of squares
- Each square representing one minterm of the function
- The simplified expressions produced by the map are always in one of the two standard forms:
  - SOP and POS
- It is sometimes possible to find two or more expressions that satisfy the minimization. In that case, either solution is satisfactory.

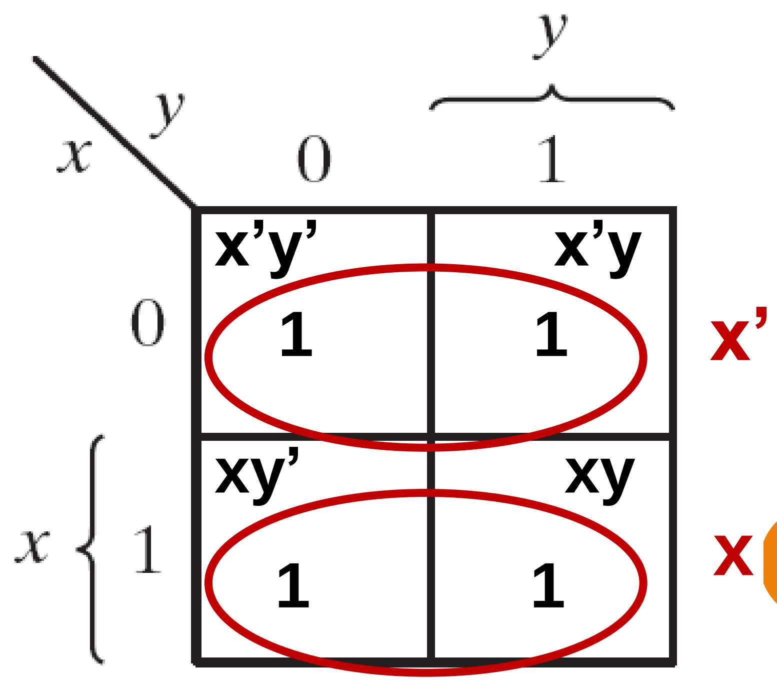
m_0	m_1
m_2	m_3

TWO-VARIABLE MAP

- Upon the Boolean function, the minterms are filled by 1's (Only adjacent 1's (2 or 4) can be grouped)
- Grouping will reduce a literal:

(1) $x'y' + x'y = x'(y' + y) = x'$

(2) $xy' + xy = x(y' + y) = x$

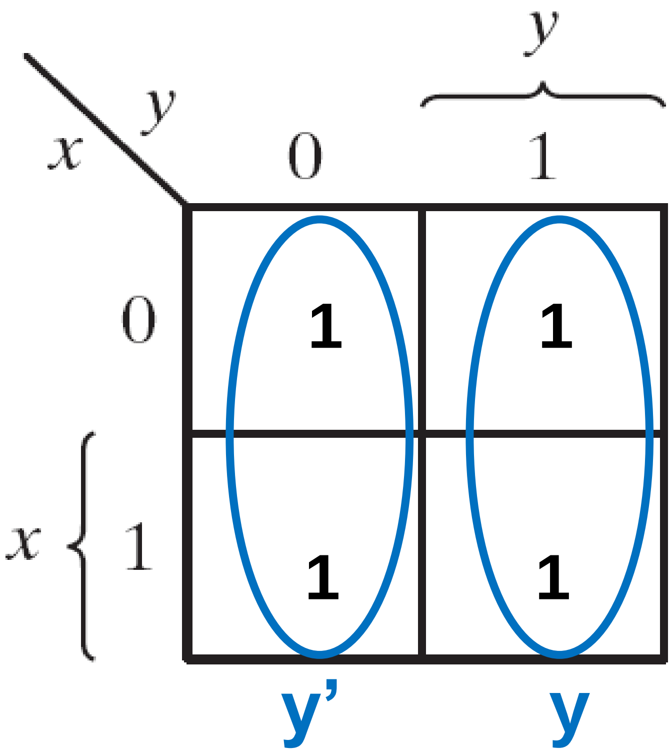


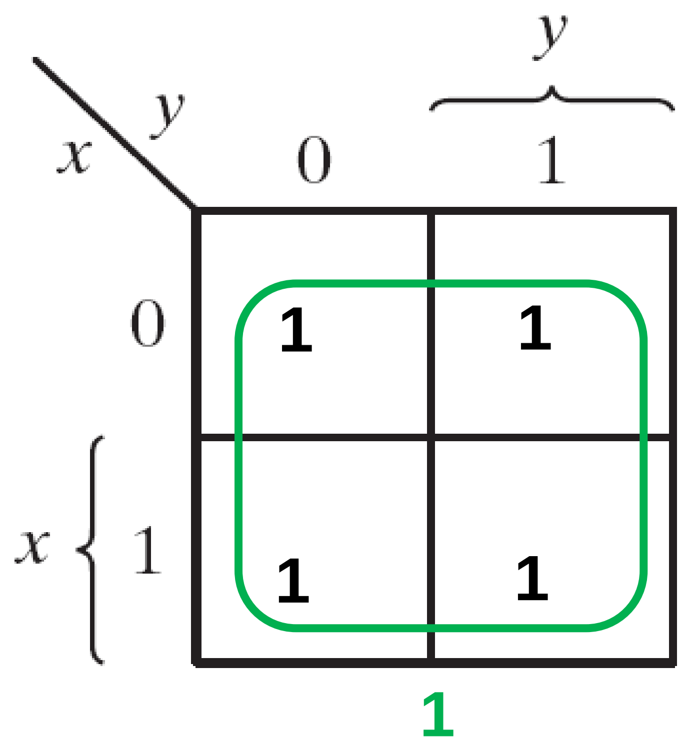
TWO-VARIABLE MAP (CONT.)

(3) x'y'+ xy' = y' (4) x'y+ xy = y

(5) x'y'+xy'+x'y+ xy = ?

- If the four adjacent (the whole map) are grouped, the output is the identity function (i.e. 1).





EXAMPLE:

Boolean Algebra

- $F = x'y + xy' + xy$

  $= x'y + xy' + \overbrace{xy + xy}^{x + x = x}$

  $= (x'y + xy) + (xy' + xy)$

  $= y(x' + x) + x(y + y')$

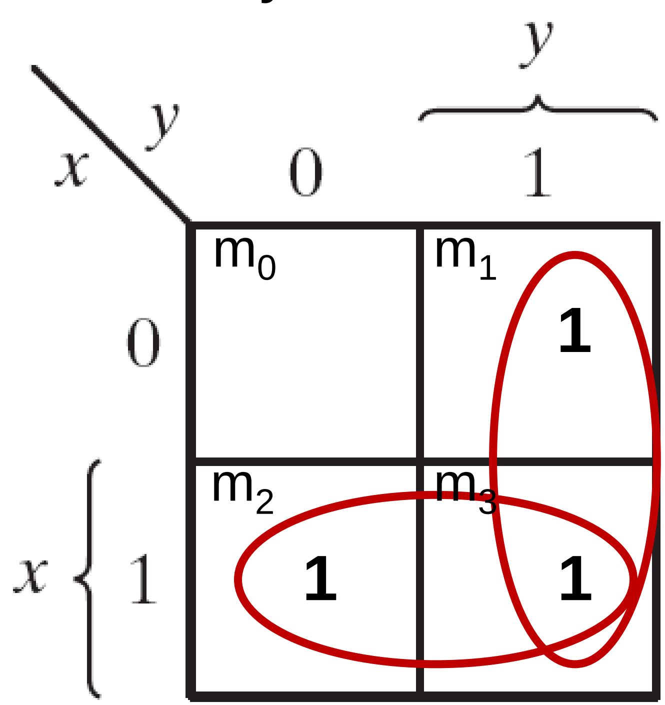
  $= x + y$

A SOP expression can be obtained by ORing all squares that contain a 1.

K-map

- $F = x'y + xy' + xy$

  $= x + y$



THREE-VARIABLE MAP

- 3 variables → 8 squares
- The minterms are arranged, in a sequence similar to ***Gray code***; only one bit changes in value from one adjacent column to the next.

Take care

m_0	m_1	m_3	m_2
m_4	m_5	m_7	m_6

(a)

$x \backslash yz$	00	01	11	10
0	m_0 $x'y'z'$	m_1 $x'y'z$	m_3 $x'yz$	m_2 $x'yz'$
x 1	m_4 $xy'z'$	m_5 $xy'z$	m_7 xyz	m_6 xyz'

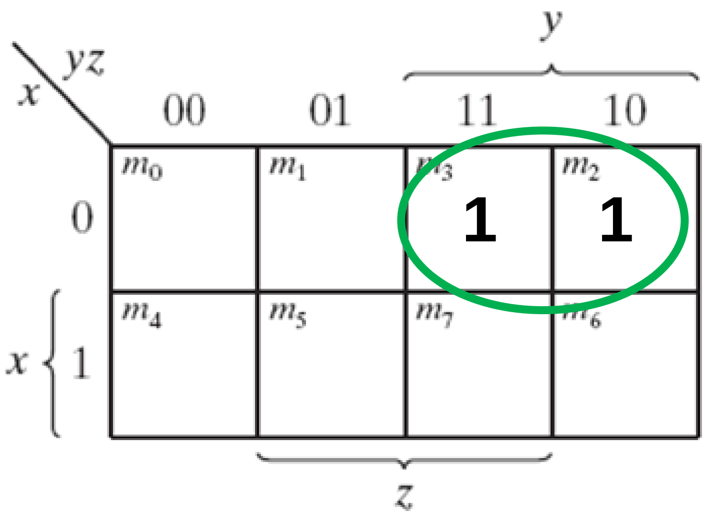
y: columns 11, 10; z: columns 01, 11

(b)

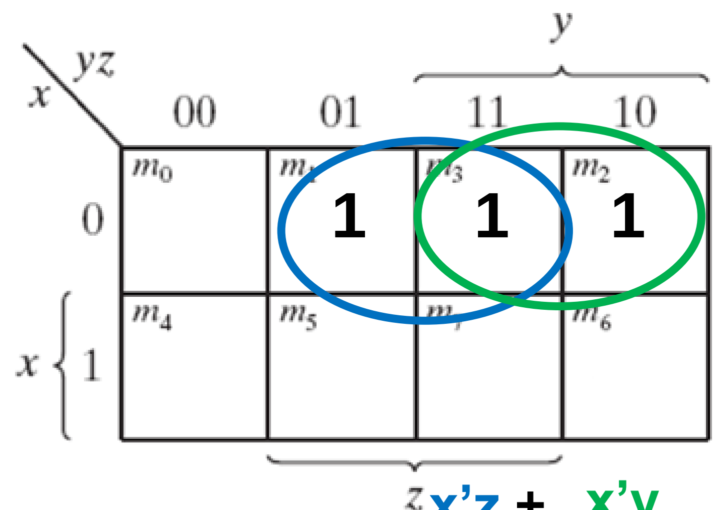
THREE-VARIABLE MAP: ADJACENT SQUARES

- The number of adjacent squares that may be combined must always represent a number that is a power of two; 1,2,4,8,…
- As more adjacent squares combined, we obtain a product term with fewer literals:
  - 1 square → given 3 literals
  - 2 adjacent squares → given 2 literals
  - 4 adjacent squares → given 1 literal
  - 8 adjacent squares (the entire map)→ always 1

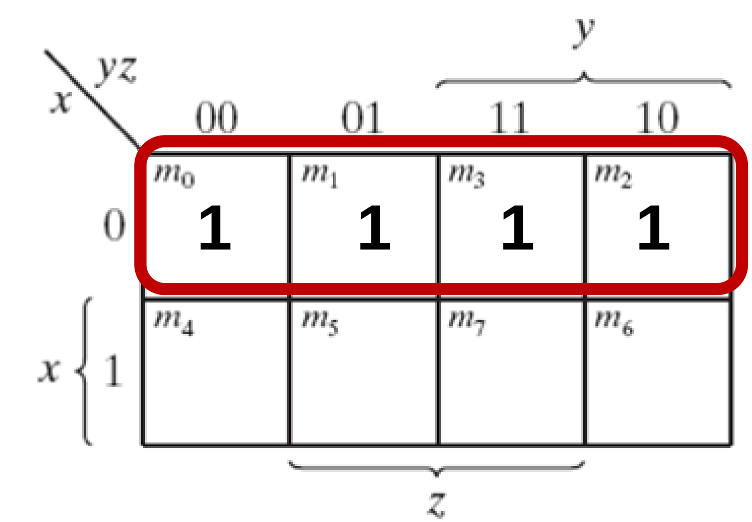
THREE-VARIABLE MAP:
ADJACENT SQUARES (CONT.)



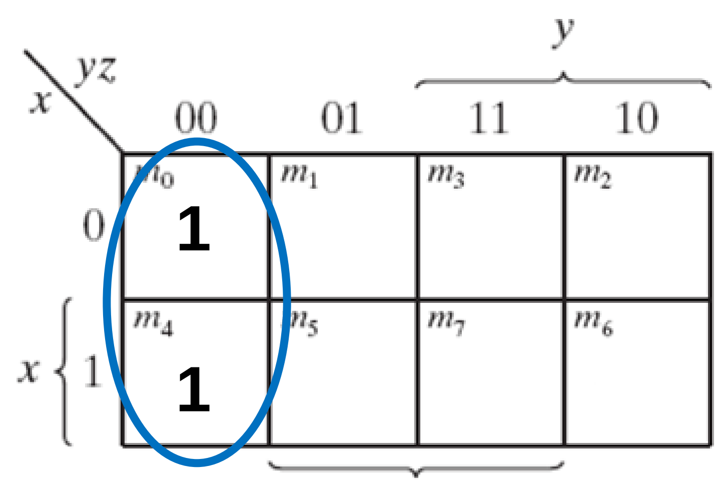
x’y



x’z + x’y

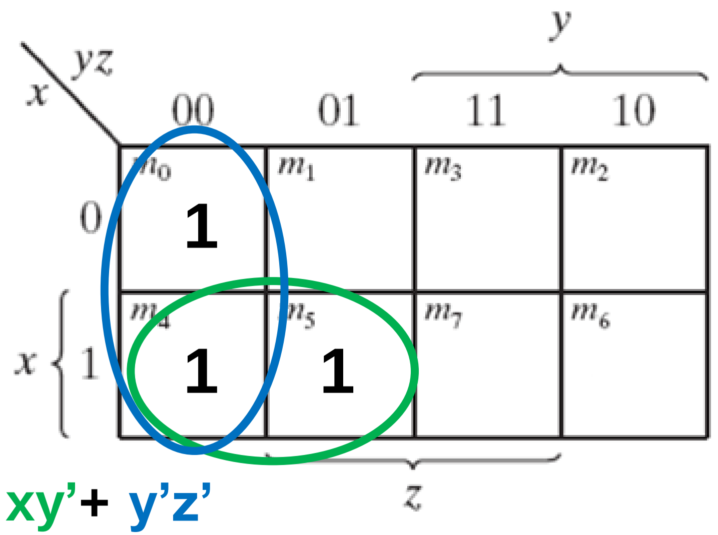


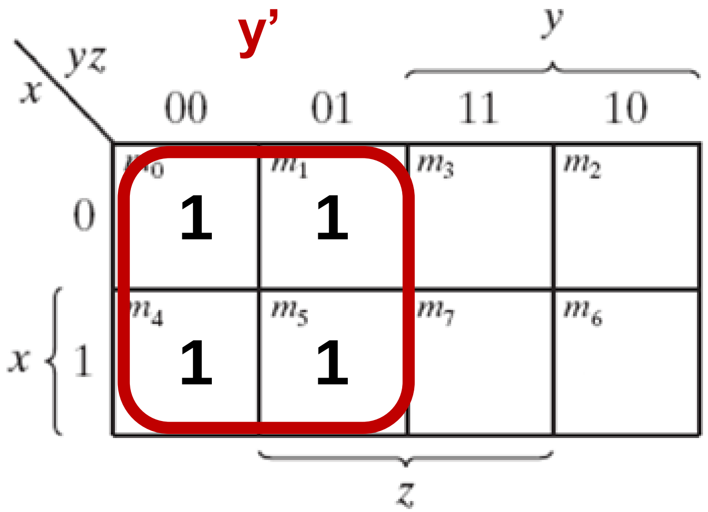
x’

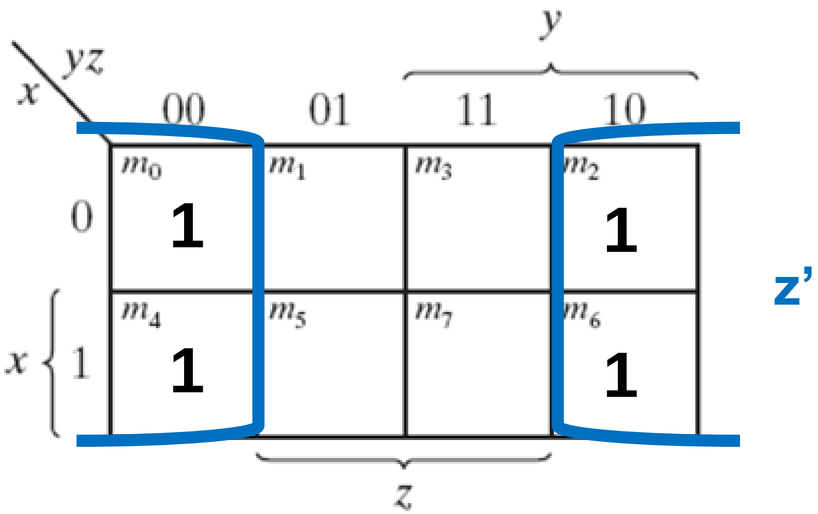


y’z’

Three-variable map: Adjacent squares (Cont.)

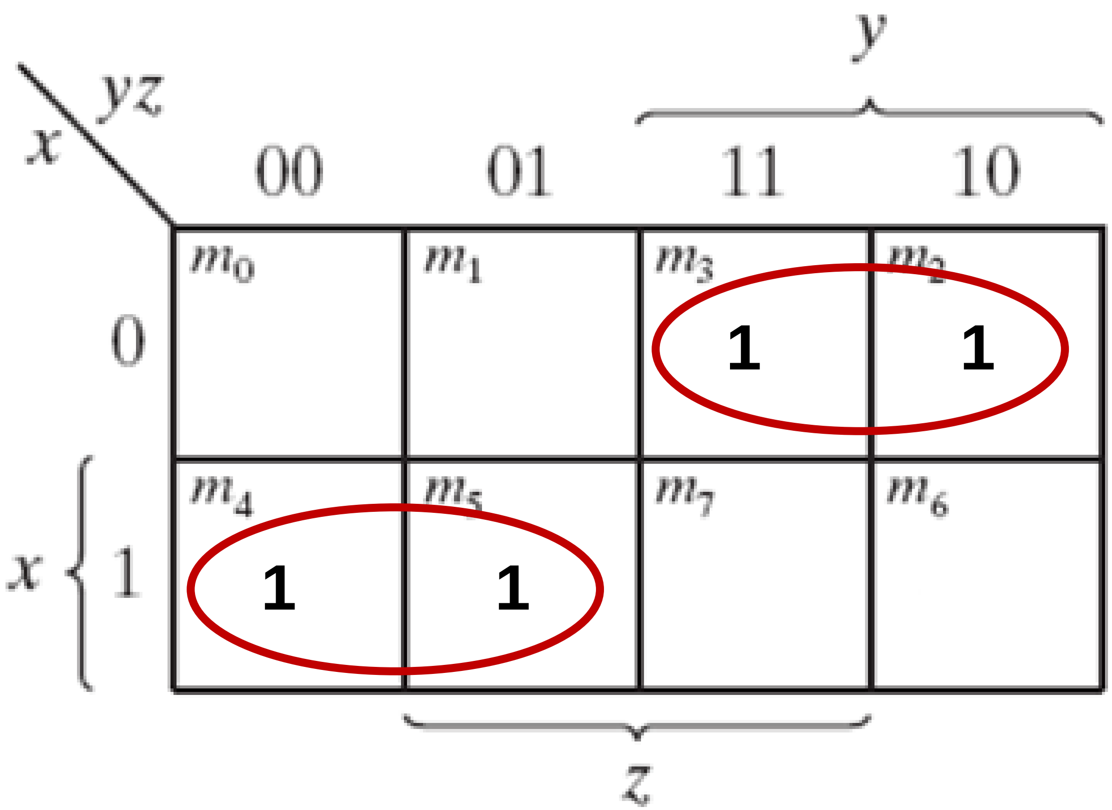






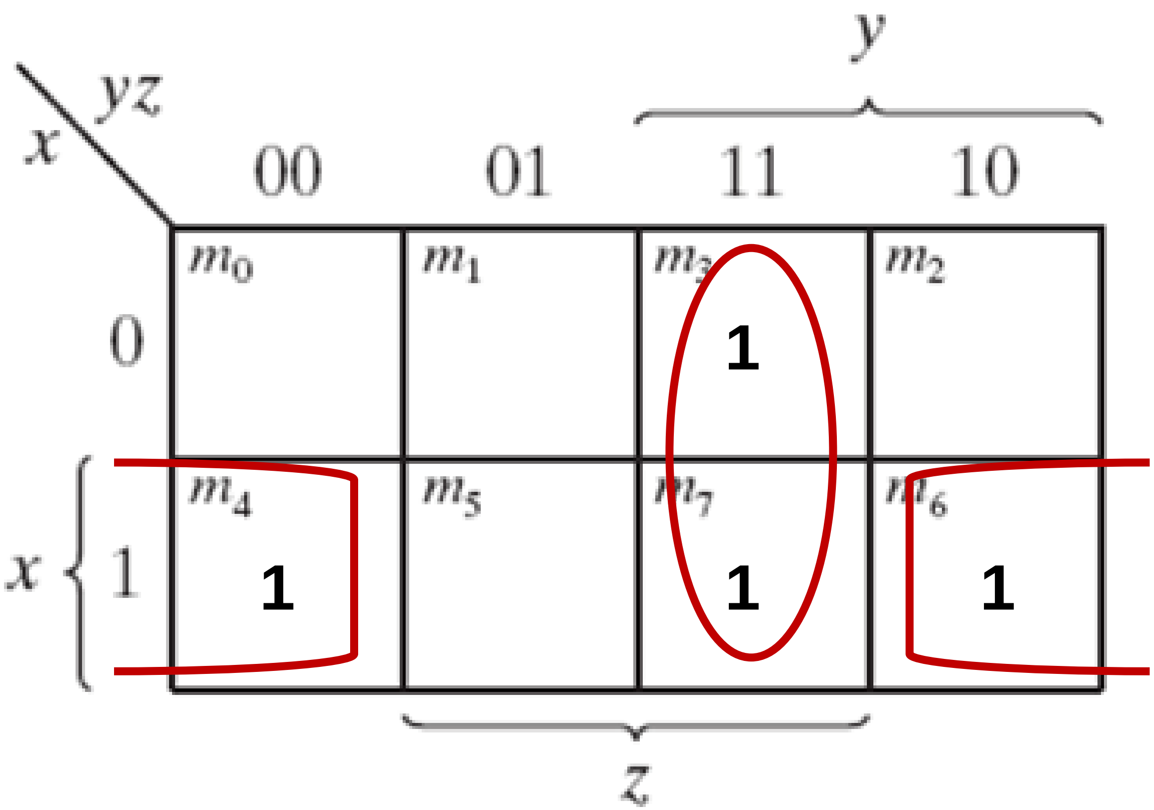
EXAMPLE 3.1 (P.92)

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



EXAMPLE 3.2 (P.93)

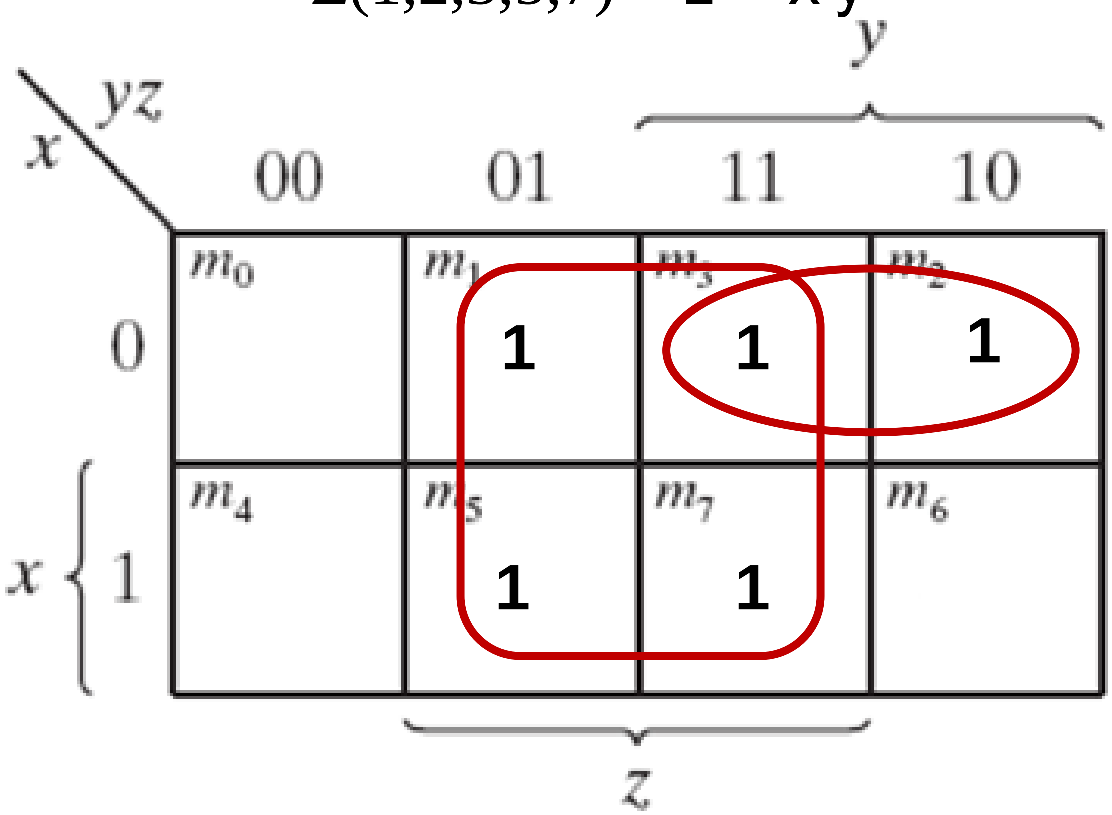
- F(x, y, z) $= \Sigma(3,4,6,7)$
  $= yz + xz'$



EXAMPLE 3.4 (P.95)

- If a function is not expressed in sum of minterms:

$$F = x'z + x'y + xy'z + yz$$
$$= \Sigma(1,2,3,5,7) = z + x'y$$



FOUR-VARIABLE MAP

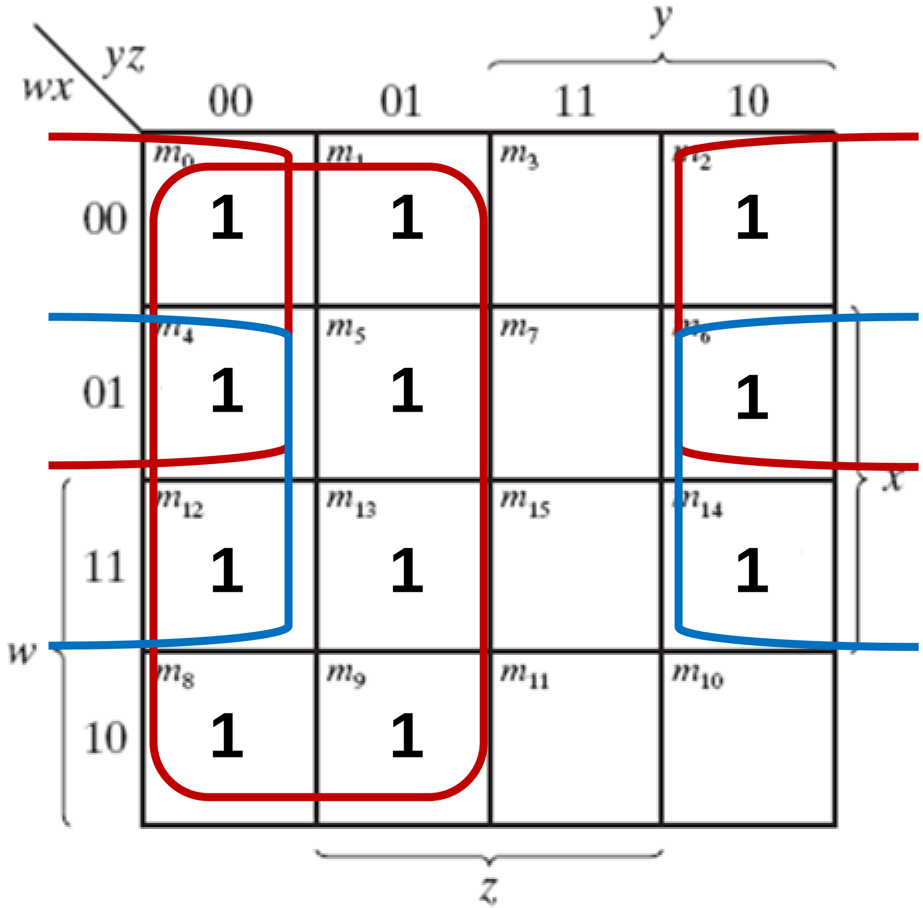
- There are 16 minterms
- Rows and columns are numbered in Gray code sequence
- Its map minimization is similar as the method used in three-variable functions.
- Adjacent squares are defined to be:
  - Squares next to each other
  - The top and bottom edges
  - The right and left edges

$wx \backslash yz$	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} $wx'yz'$

1 square	→ 4 literals
2 adjacent squares	→ 3 literals
4 adjacent squares	→ 2 literal
8 adjacent squares	→ 1 literal
16 adjacent squares	→ always 1

EXAMPLE 3.5 (P.97)

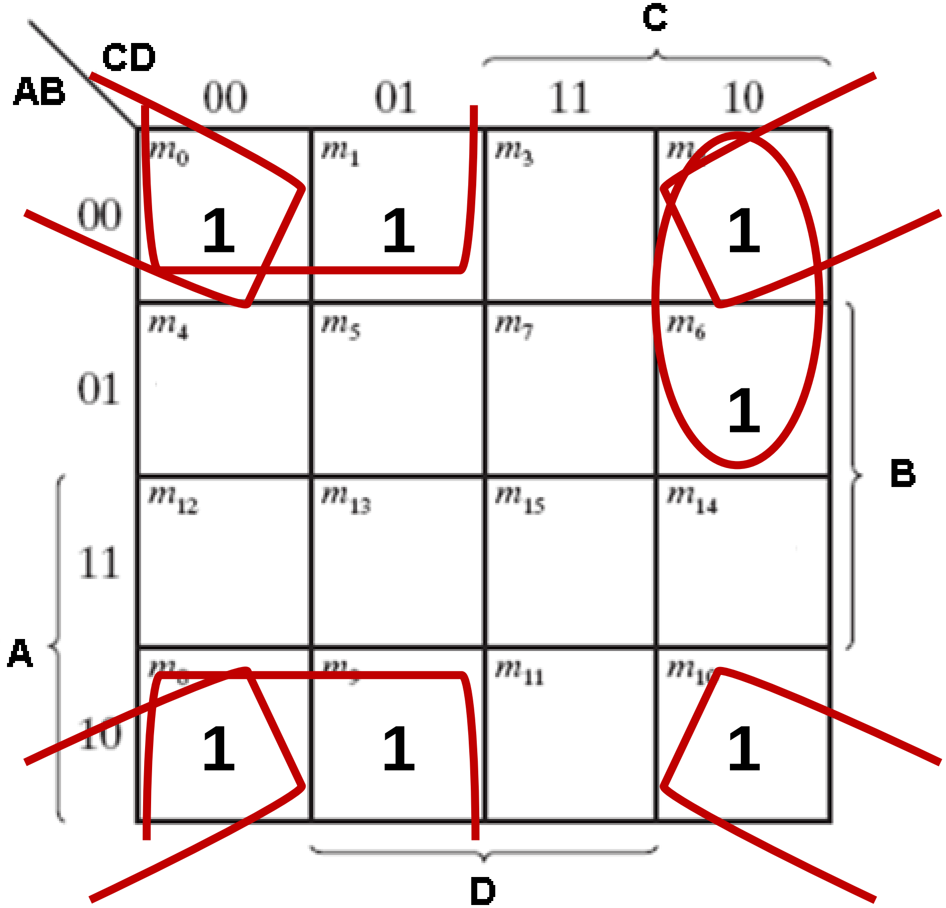
- F(w, x, y, z) = Σ(0,1,2,4,5,6,8,9,12,13,14)
  = y' + w'z' + xz'



EXAMPLE 3.6 (P.97)

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

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



PRIME IMPLICANTS

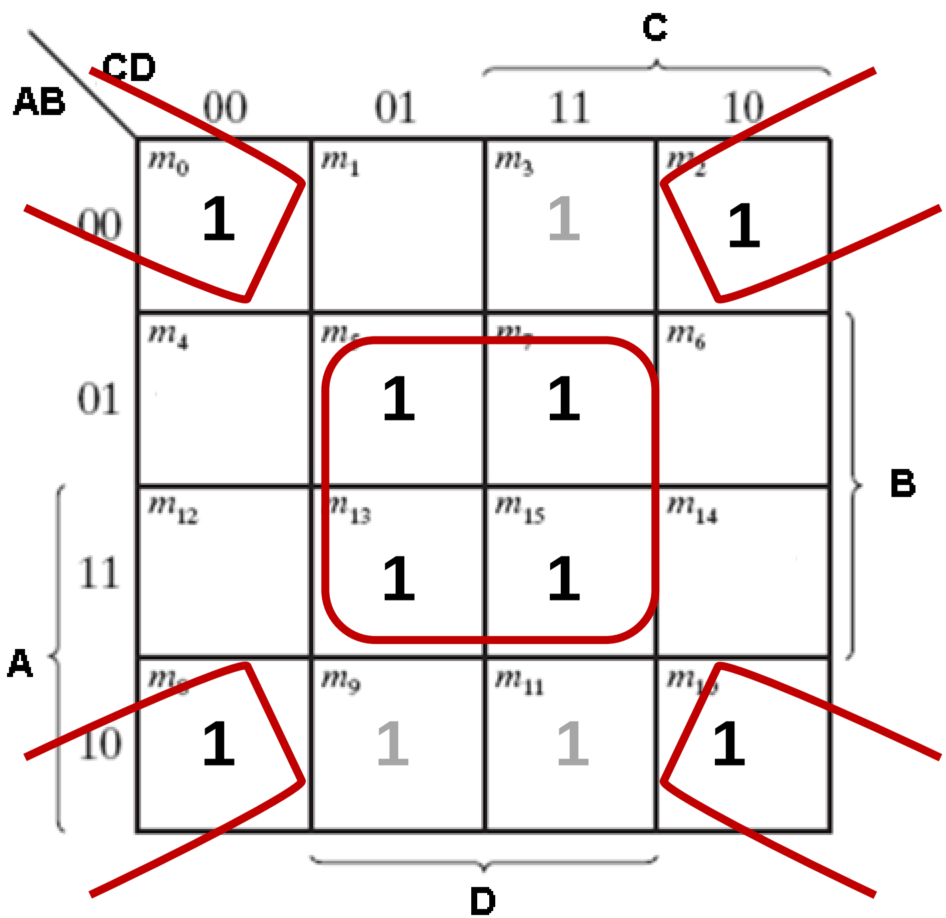
- **Prime implicant:**
  - Is a product term obtained by combining the maximum possible number of adjacent squares in the map

- **Essential prime implicant**
  - If a minterm in a square is covered by only one prime implicant

EXAMPLE

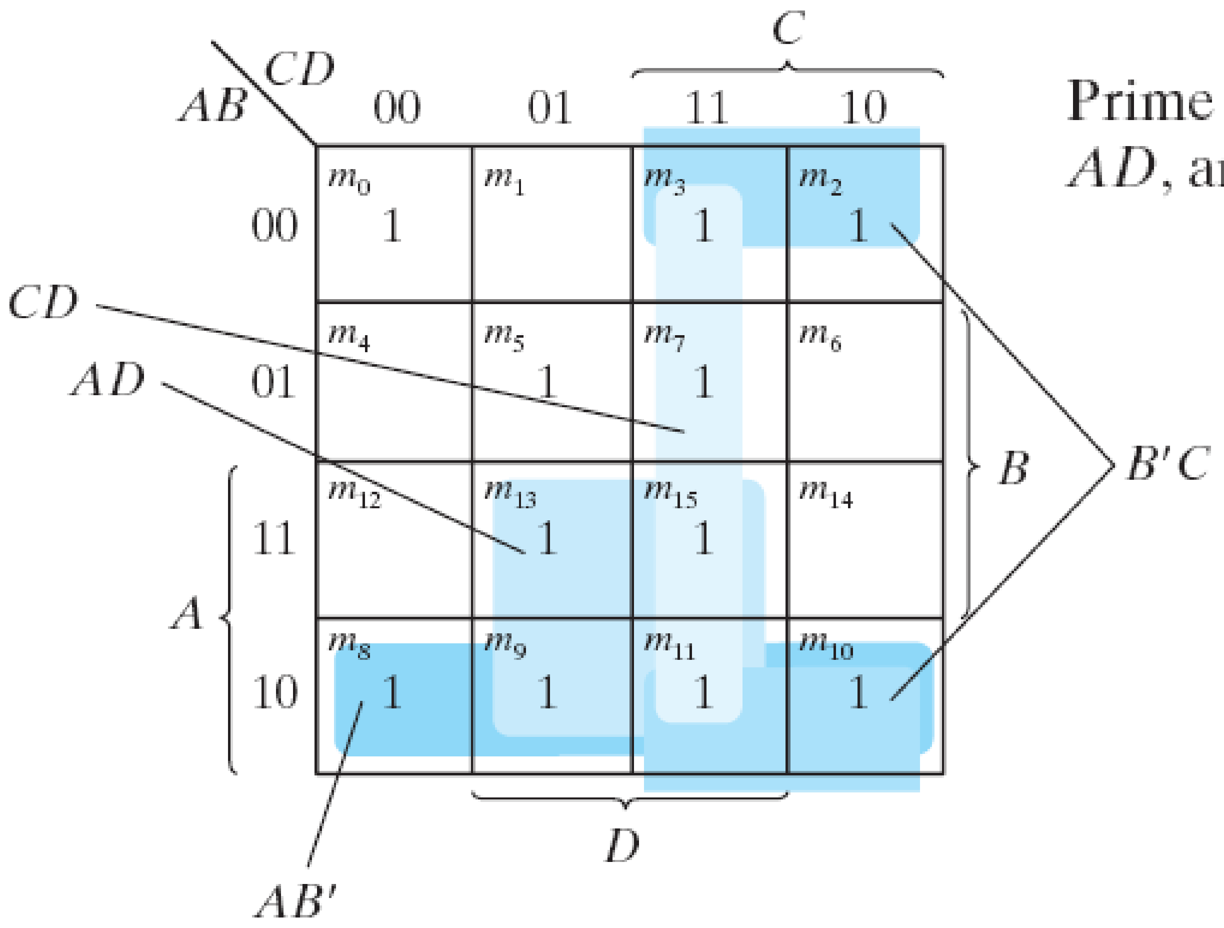
- F(A, B, C, D)= Σ(0,2,3,5,7,8,9,10,11,13,15)

Essential prime implicants: BD and B'D'



EXAMPLE (CONT.)

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



Prime implicants CD, $B'C$, AD, and AB'

Example (Cont.)

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

This function has 4 different simplified expressions:

$$F = BD + B'D' + CD + AD$$
$$= BD + B'D' + CD + AB'$$
$$= BD + B'D' + B'C + AD$$
$$= \underbrace{BD + B'D'}_{\substack{\text{Essential}\\ \text{prime implicants}}} + \underbrace{B'C + AB'}_{\substack{\text{Other}\\ \text{prime implicants}}}$$

- The simplified expression =

  essential implicants + other prime implicants

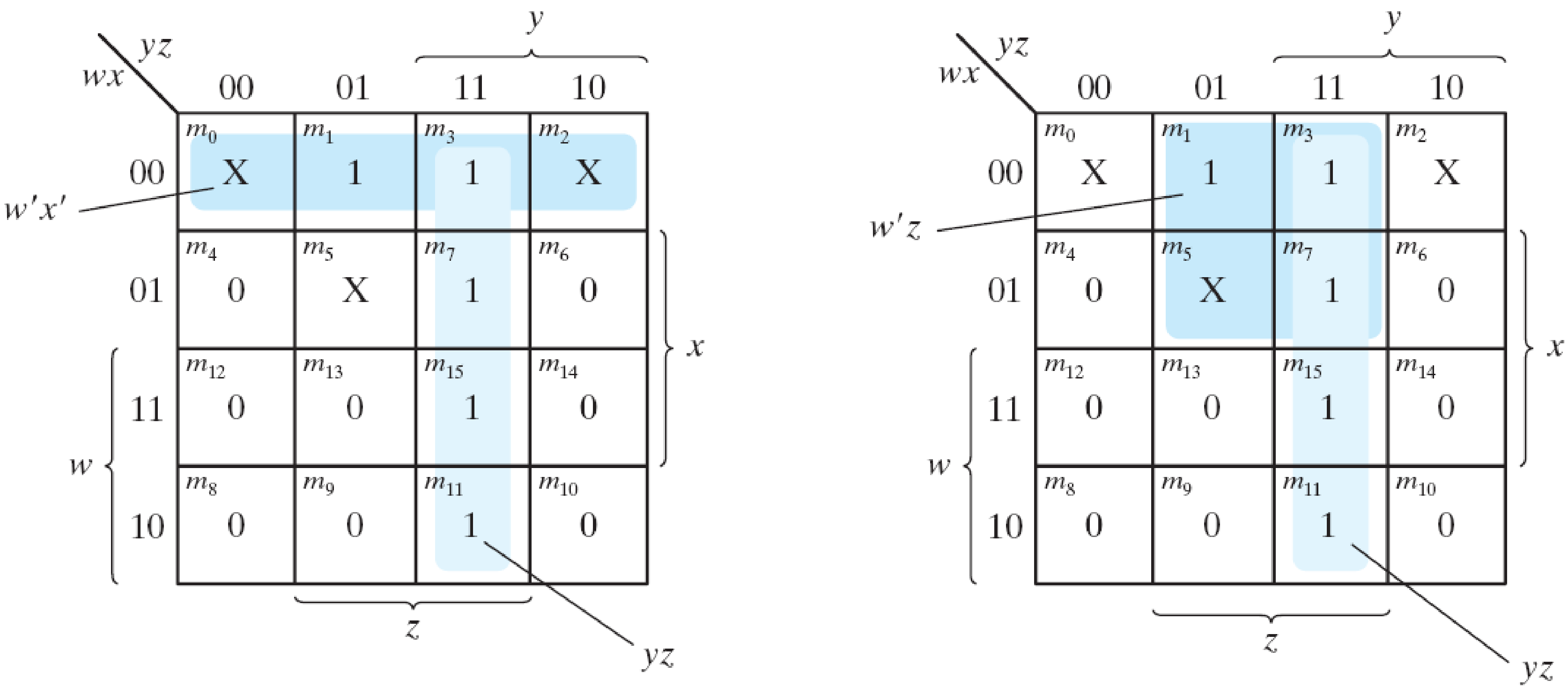
DON'T CARE CONDITIONS

- The Boolean function specifies the conditions under which the function is equal to 1 or 0.
- In practice, in some applications the function is not specified for certain combinations of the variables.
  - ➔ called "***incompletely specified functions***"
  - ➔ we simply don't care what value is
- A ***don't care minterm*** is a combination of variables whose logical value is not specified.
  - It marked as **x**
  - When simplifying the function, we choose to include each don't care minterm with either 1's or 0's, depending on which combination gives the simplest expression.

EXAMPLE 3.9(P. 103)

- F(w,x,y,z) = Σ (1,3,7,11,15) and d(w,x,y,z)= Σ (0,2,5)

Don't care conditions



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

Either expression is accepted

FIVE-VARIABLE MAP

- Maps for more than 4 variables are not simple to use as for four or fewer variables.
- 5 variable needs 32 squares

SIX-VARIABLE MAP AND MORE

- Six-variable map consists of:
  - 4 four-variable maps
  - Require 64 squares
- Maps with six and more variables need to many squares and are impractical to use.
- The alternative is to employ computer programs specifically written to facilitate the simplification of Boolean functions with a large number of variables.

Product of sums simplification

- For obtaining a minimized function in product of sums:

1. Mark the empty squares by 0’s
2. Combine them into valid adjacent squares

➔ we obtain the simplified complemented function

3. Apply DeMorgan’s theorem (or dual and literal complementing)

➔ we obtain the simplified function in product of sums

EXAMPLE 3.7 (P.101)

- F(A,B,C,D)=

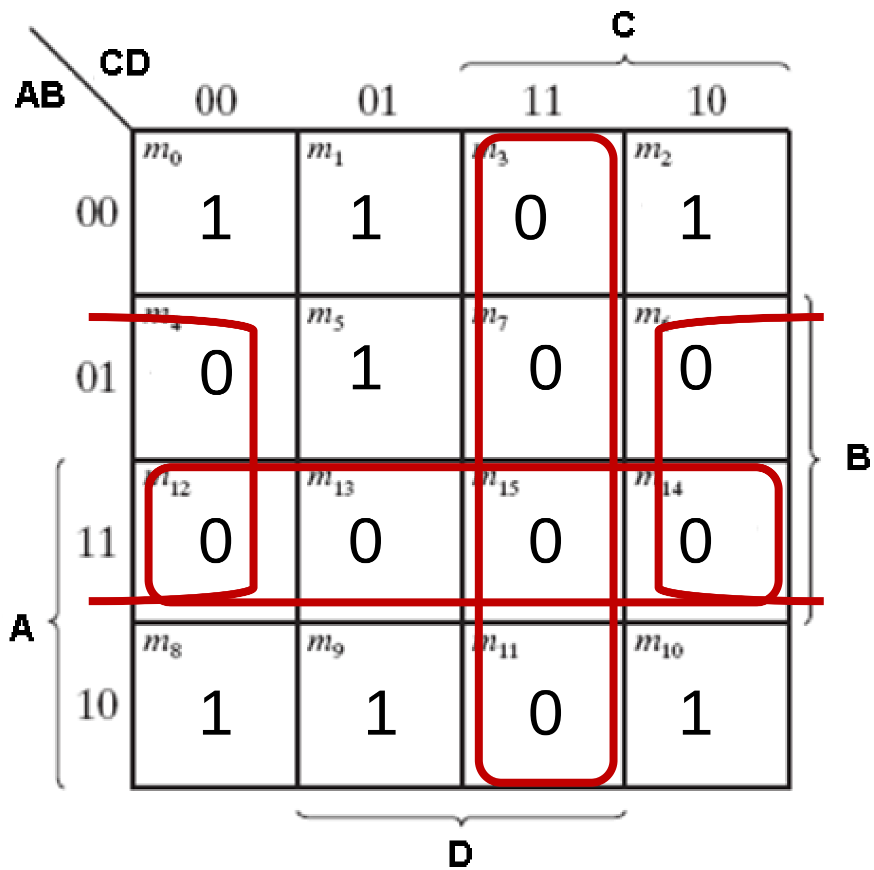
  Σ (0,1,2,5,8,9,10)

Simplify into:

Product of sums

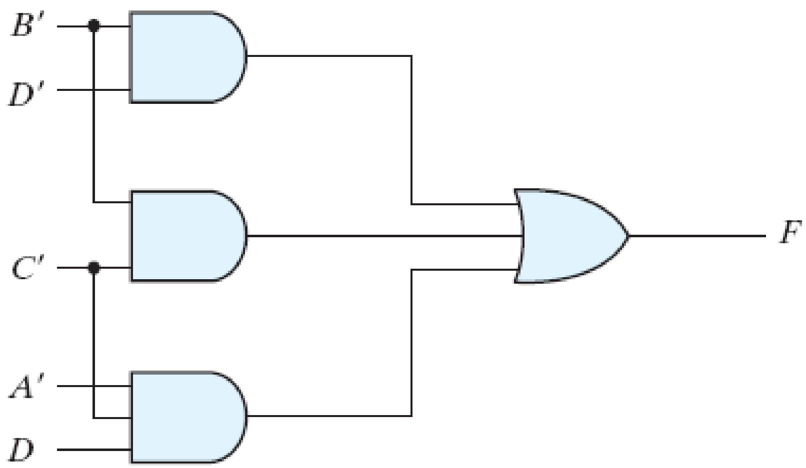
F'= AB + CD + BD'

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



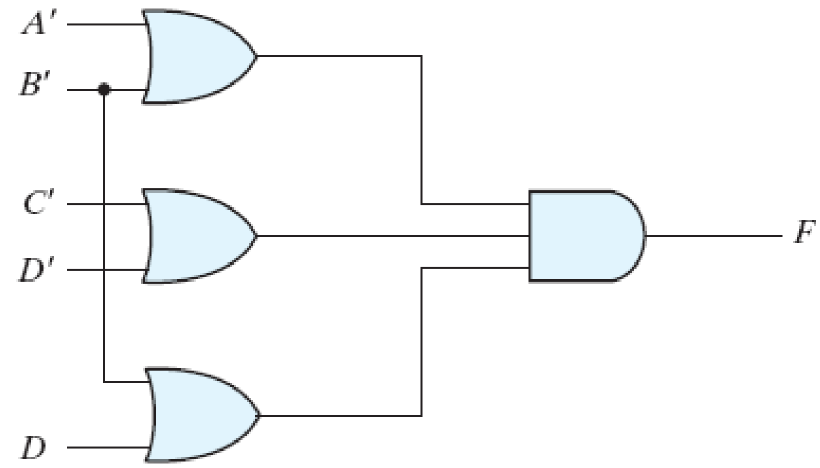
EXAMPLE 3.7 (CONT.)

- Implementation of simplified expressions (two-level implementation):



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

SOP



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

POS

EXAMPLE

Suppose, the function is expressed in:

- The sum of minterms

F(x,y,z)= $\Sigma(1,3,4,6)$

➔ Represent 1's in map

- The product of maxterms

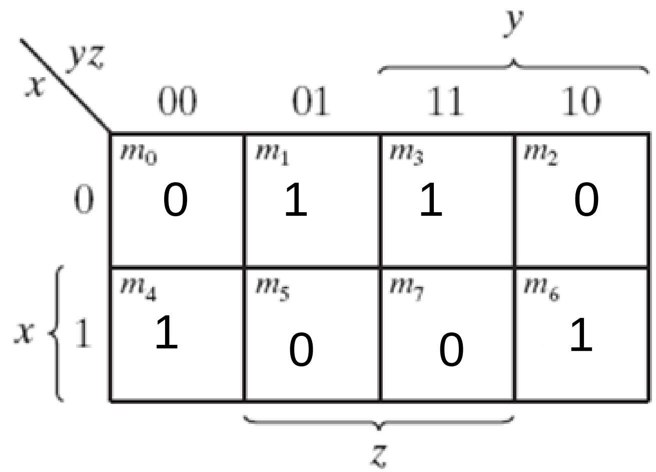
F(x,y,z)=$\Pi(0,2,5,7)$

➔ Represent 0's in map

- The truth table

Table 3.2
Truth Table of Function F

x	y	z	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0



Example

- Given function expressed into product of sums form

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

1) Take its complement

$$F'= ABC + B'D'$$

2) Mark the minterms of F' with 0's and others with 1's

3) Simplify the function by Combining:

- 1's if the required is SOP
- 0's and then complement if the required is POS

AB \ CD	00	01	11	10
00	m_0 0	m_1 1	m_3 1	m_2 0
01	m_4 1	m_5 1	m_7 1	m_6 1
11	m_{12} 1	m_{13} 1	m_{15} 0	m_{14} 0
10	m_8 0	m_9 1	m_{11} 1	m_{10} 0

EXAMPLE:
PRODUCT OF SUMS WITH DON'T CARE

F(w,x,y,z) = Σ (1,3,7,11,15) and d(w,x,y,z)= Σ (0,2,5)

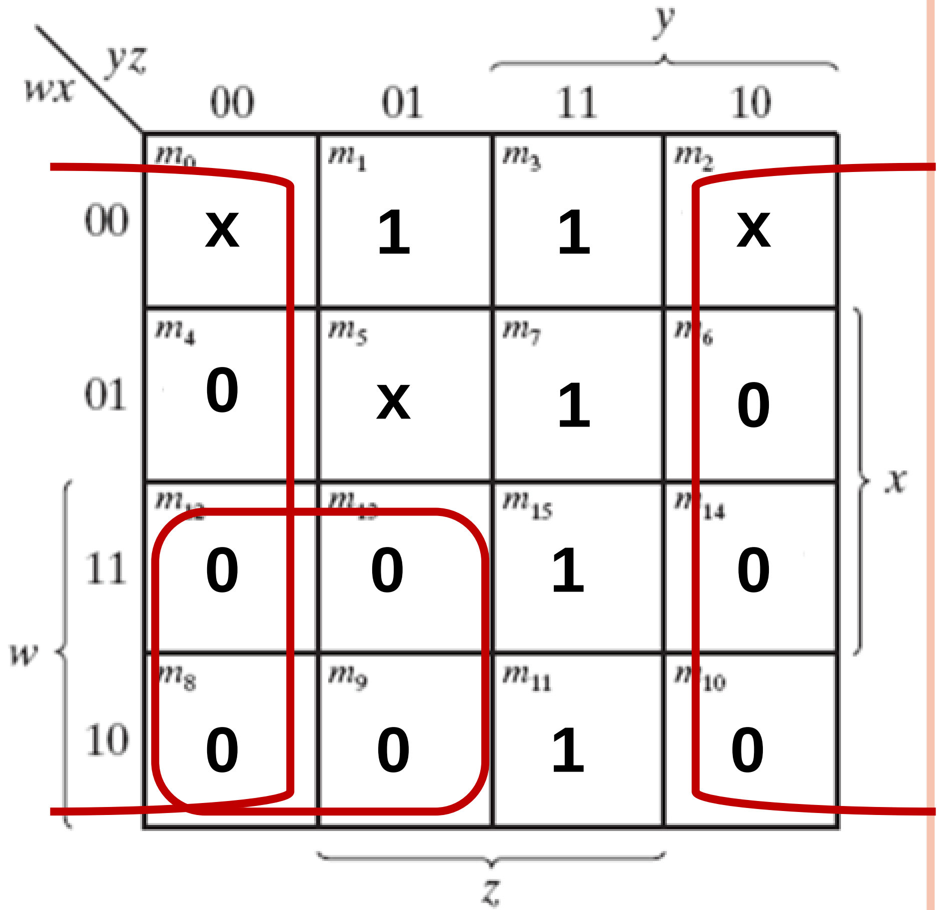
Product of sums=?

- F'= z' + wy'

  F= z(w'+y)

➔F= Π (**0,2**,4,6,8,9,10,12,13)

➔F =Σ (1,3,**5**,7,11,15)



THANKS

We covered from:

Ch.3 (sec. 3.2 → 3.6)

Next week:

Analysis and design of combinational circuit