Chapter3: Gate-Level Minimization

Lecture1- Three and Four-Variables Function Simplification using Map Method

Engr. Arshad Nazir, Asst Prof
Dept of Electrical Engineering
SEECS

Chapter Contents

Karnaugh Maps
SOP and POS Simplifications
Don’t Care Conditions
Quine McCluskey Minimization Algorithm
NAND and NOR Implementations
Parity Generation and Detection

Objectives

- Introduction to Map Method
- Plot and Labeling of minterms on Map
- Functions Simplification in Sum-of-Products (SOP) form using Three and Four-Variables Map

K-Map Method

- The Karnaugh Map (K-Map) method uses a simple procedure for minimizing Boolean functions.
  - The map is a diagram made up of squares with each square representing one minterm of the function.
  - The key is to learn to identify visual patterns.
  - The result is always an expression that is in one of the two standard forms, SOP or POS.
  - Much faster and more efficient than previous minimization techniques with Boolean algebra. It can be used to simplify functions of up to six variables.
  - It is possible to find two or more expressions that satisfy the minimization criteria.
  - Rules to consider
    - Every cell containing a 1 must be included at least once.
    - The largest possible “power of 2 rectangle” must be enclosed.
    - The 1’s must be enclosed in the smallest possible number of rectangles.

Two-Variable Map

- A two-variable map holds four minterms for two variables.
  - We mark the squares of the minterms that belong to a given function.
  - Combine adjacent squares to find minimal expression.

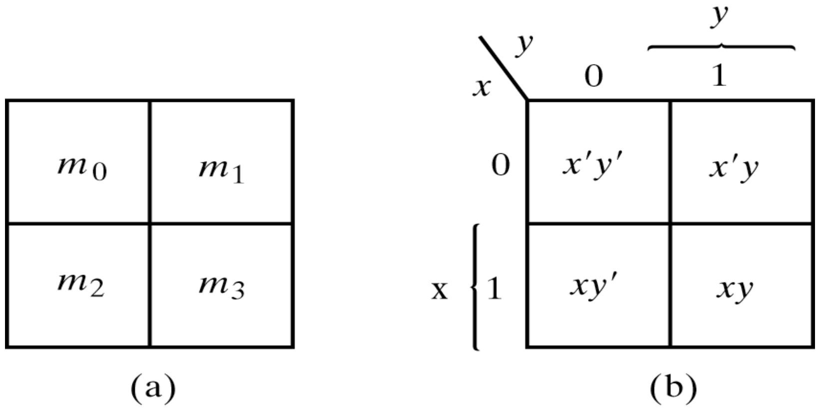


Fig. 3-1 Two-variable Map

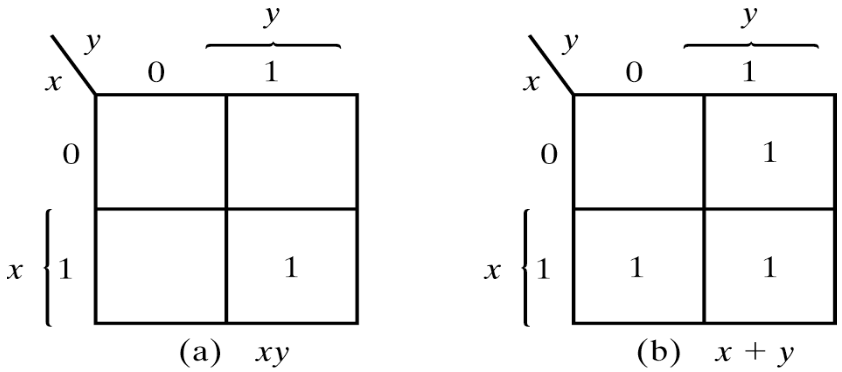


Fig. 3-2 Representation of Functions in the Map

Three-Variable Map

- A three-variable map holds eight minterms for three variables.
  - Again, we mark the squares of the minterms that belong to a given function.
  - Note that the sequence is arranged in Gray code to allow only one bit to change from column to column and row to row.
- Since any two adjacent cells in a 3-variable map represent a change in only a single bit, we use this to do minimization.
  - Consider the two cells for m_0 and m_1 where the difference is the negation of the bit z.
  - $F = m_0 + m_1 = x'y'z' + x'y'z = x'y'(z' + z) = x'y'$

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	$x'y'z'$	$x'y'z$	$x'yz$	$x'yz'$
1	$xy'z'$	$xy'z$	xyz	xyz'

(b)

Fig. 3-3 Three-variable Map

Minimization Example

- Each of the two adjacent pairs of entries can be simplified by eliminating the changing bit (z in both cases).
  - F (x,y,z) = x’y’ + xy

x \ yz	00	01	11	10
0	1	1		
1			1	1

$$F(x, y, z) = \sum(0,1,6,7)$$

x \ yz	00	01	11	10
0	m_0	m_1	m_3	m_2
1	m_4	m_5	m_7	m_6

Notes on Adjacency

- So far, we have assumed that adjacent cells in the map need to touch each other but this is not always the case.
  - ➢ m_0 and m_2 are considered adjacent
    - ○ $m_0 + m_2 = x'y'z' + x'yz' = x'z'(y' + y) = x'z'$
  - ➢ m_4 and m_6 are considered adjacent
    - ○ $m_4 + m_6 = xy'z' + xyz' = xz'(y' + y) = xz'$

x \ yz	00	01	11	10
0	m_0	m_1	m_3	m_2
1	m_4	m_5	m_7	m_6

x \ yz	00	01	11	10
0	$x'y'z'$	$x'y'z$	$x'yz$	$x'yz'$
1	$xy'z'$	$xy'z$	xyz	xyz'

3-Variable Map Patterns

- The number of adjacent squares that may be combined always represent a number that is a power of 2 such as 1, 2, 4, and 8.
  - One square represents one minterm with three literals.
  - Two adjacent squares represents a term of two literals.
  - Four adjacent squares represents a term of one literal.
  - Eight adjacent squares represents the entire map and produces a function that is always equal to 1.

Mapping Functions Example

- Given the function
  - ➢ F = x′z + xy′ + xy′z + yz

    F = ∑(1, 3, 4, 5, 7)

x \ yz	00	01	11	10
0	m_0	m_1	m_3	m_2
1	m_4	m_5	m_7	m_6

- Map the function
- Determine the sum of minterms equation
- Determine the minimum sum of products expression

x \ yz	00	01	11	10
0		1	1	
1	1	1	1	

- The minimum sum-of-Products (SOP) is

  F= = z + xy′

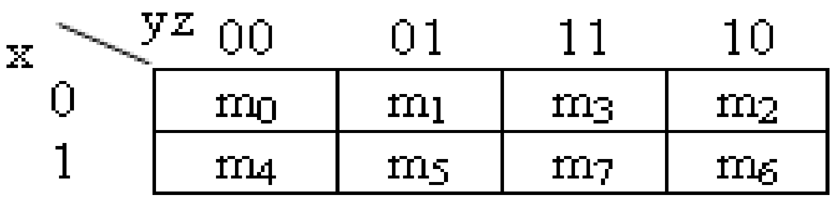
Example 3-1

Simplify the Boolean function F(x,y,z)= ∑(2,3,4,5) using map method

Solution:

- 1 is marked in each minterm that represents the function
- Find the possible adjacent squares and mark them with rectangles
- The upper right rectangle represents the area enclosed closed by x’y (eliminating the changing bit)
- Similarly lower left rectangle represents xy’
- The logical sum of these two terms gives:

F = x'y + xy'



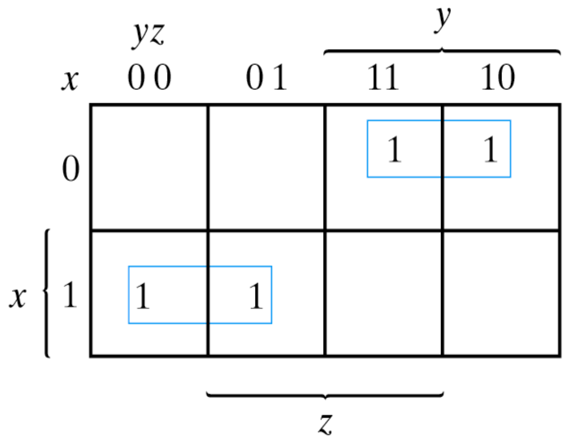


Fig. 3-4 Map for Example 3-1; $F(x, y, z) = \Sigma(2, 3, 4, 5) = x'y + xy'$

Example 3-2

Simply the function $F(x,y,z) = \sum(3,4,6,7)$

Solution:

- 1 is marked in each minterm that represents the function
- Find the possible adjacent squares and mark them with rectangles
- Two adjacent squares are combined in the third column to give a two-literal term yz
- The remaining two squares with 1's are enclosed in half rectangles. This gives two-literal term xz'
- The logical sum of these two terms gives: F = yz + xz'

x \ yz	00	01	11	10
0	m_0	m_1	m_3	m_2
1	m_4	m_5	m_7	m_6

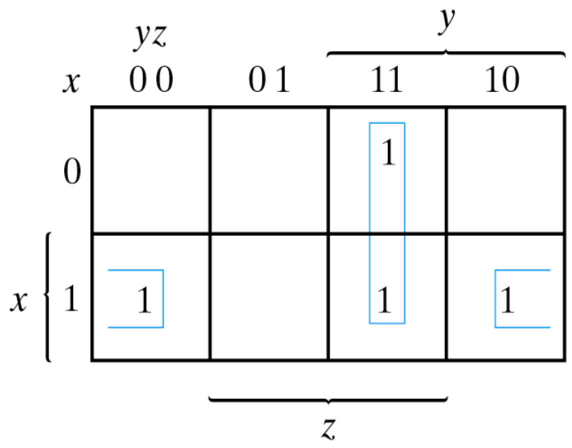


Fig. 3-5 Map for Example 3-2; $F(x, y, z) = \Sigma(3, 4, 6, 7) = yz + xz'$

Example 3-3

Simply the function F(x,y,z) =∑ S(0,2,4,5,6)

Solution:

- 1 is marked in each minterm that represents the function
- Find the possible adjacent squares and mark them with rectangles
- We combine four adjacent squares to get a single literal term z’ as $m_0+m_2+m_4+m_6$

  x'y'z'+x'yz'+xy'z'+xyz’= x'z'(y'+y) +xz'(y'+y)

  = x'z' + xz’ = z'
- The remaining two squares with 1’s are enclosed by a rectangle (with one square that is already used once). This gives two-literal term xy’
- The logical sum of these two terms gives:

  F = z’ + xy’

x \ yz	00	01	11	10
0	m_0	m_1	m_3	m_2
1	m_4	m_5	m_7	m_6

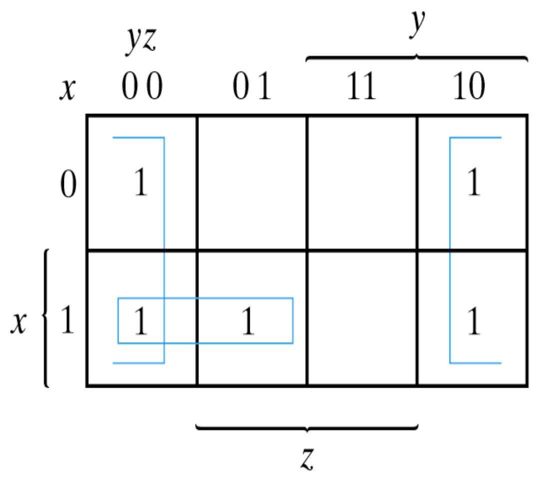


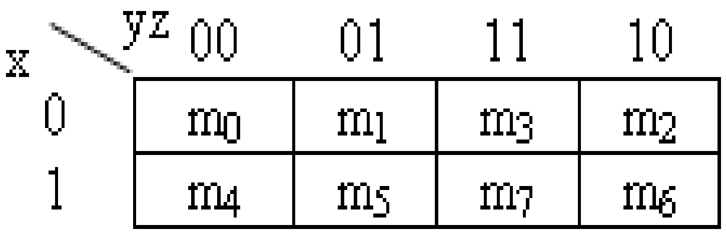
Fig. 3-6 Map for Example 3-3; $F(x, y, z) = \Sigma(0, 2, 4, 5, 6) = z' + xy'$

Example 3-4

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

Solution:

- The two squares corresponding to the first term A'C. (A' first row and C two middle columns)
- A'B has 1's in squares 011 and 010 in the same way. AB'C has 1 square 101 and BC has two 1's in squares 011 and 111
- The function has total of 5 minterms as shown in figure
- Find the possible adjacent squares and mark them with rectangles as shown in the map
- It can be simplified with only two terms giving: F = C + A'B



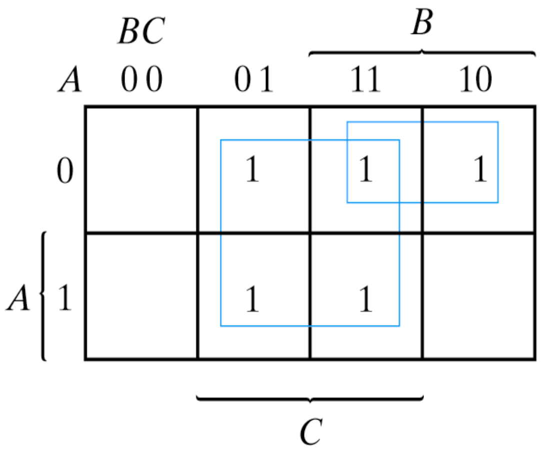


Fig. 3-7 Map for Example 3-4; $A'C + A'B + AB'C + BC = C + A'B$

Four-Variable Map

- A four-variable map holds 16 minterms for four variables.
  - ➢ Again, we mark the squares of the minterms that belong to a given function.
  - ➢ Note that the sequence is not arranged in a binary way.
  - ➢ The sequence used is a Gray code and allows only one bit to change from column to column and row to row.

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)

$wx \backslash yz$	00	01	11	10
00	$w'x'y'z'$	$w'x'y'z$	$w'x'yz$	$w'x'yz'$
01	$w'xy'z'$	$w'xy'z$	$w'xyz$	$w'xyz'$
11	$wxy'z'$	$wxy'z$	$wxyz$	$wxyz'$
10	$wx'y'z'$	$wx'y'z$	$wx'yz$	$wx'yz'$

(b)

Fig. 3-8 Four-variable Map

4-Variable Map Patterns

- The number of adjacent squares that may be combined always represent a number that is a power of 2 such as 1, 2, 4, 8, and 16.
  - One square represents one minterm with four literals.
  - Two adjacent squares represents a term of three literals.
  - Four adjacent squares represents a term of two literals.
  - Eight adjacent squares represents a term of one literal.
  - Sixteen adjacent squares represents the entire map and produces a function that is always equal to 1.

Minimization Example

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

$$F(w, x, y, z) = \sum(2, 3, 6, 7, 8, 10, 11, 12, 14, 15)$$

- The eight adjacent squares can be combined to form the one literal term y.
- Four adjacent squares can be combined by folding property to form the two literal term wz'.
- The simplified expression will be logical sum of two product terms producing the function

F = y+wz'

Another Example

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

$F(w, x, y, z) = \sum(0, 2, 3, 4, 8, 10, 11)$

- Four adjacent corners can be combined to form the two literal term x′z′.
- Four adjacent squares can be combined to form the two literal term x′y.
- The remaining 1 is combined with a single adjacent 1 to obtain the three literal term w′y′z′.

$$F = x'z' + x'y + w'y'z'$$

Another Example

Simplify the function $F=A'BC'+A'CD'+ABC+AB'C'D'+ABC'+AB'C$ using map method

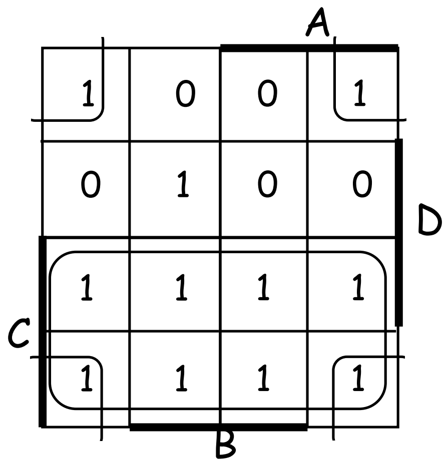
AB \ CD	00	01	11	10
00	0	0	0	1
01	1	1	0	1
11	1	1	1	1
10	1	0	1	1

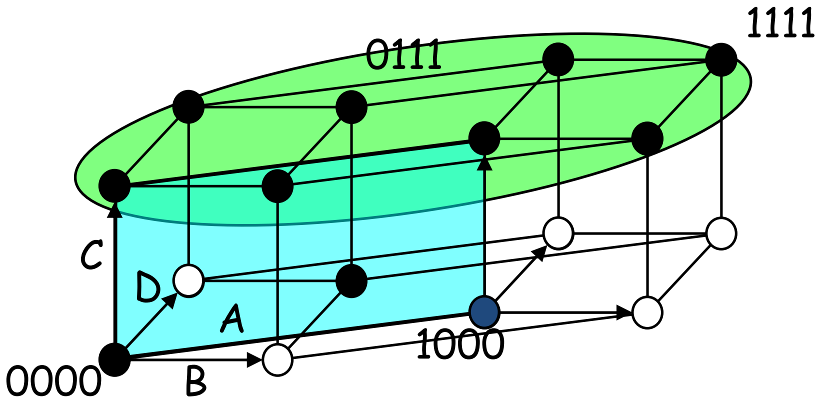
$F=BC'+CD'+AC+AD'$

Another Example

Simplify the function $F(A,B,C,D) = \sum m(0,3,5,8,9,10,11,12,13,14,15)$

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

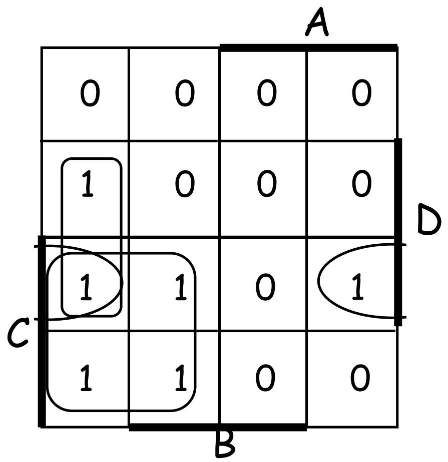




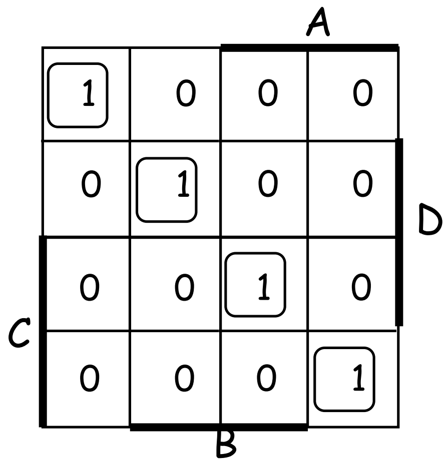
Solution set can be considered as a coordinate System!

Another Example

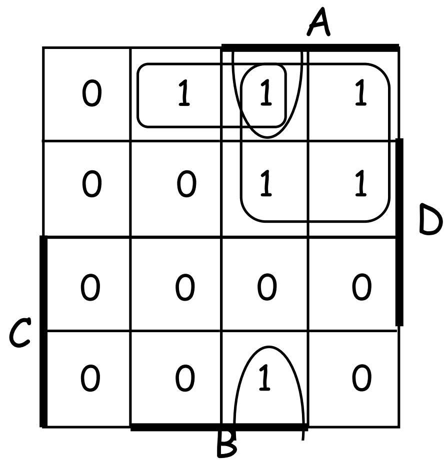
Magnitude Comparator



K-map for LT



K-map for EQ



K-map for GT

LT = A' B' D + A' C + B' C D

EQ = A'B'C'D' + A'BC'D + ABCD + AB'CD'

GT = B C' D' + A C' + A B D'

Example 3-5

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

Solution:

- 1 is marked in each minterm that represents the function
- Find the possible adjacent squares and mark them with rectangles
- We combine eight adjacent squares to get a single literal term y'
- The top two 1's on the right are combined with the top two 1's on the left to give the term w'z'
- We combine the single square left on right with three adjecent squares that are already used to give the term xz'
- The logical sum of these three terms gives:

  $F = y' + w'z' + xz'$

wx \ yz	00	01	11	10
00	m_0	m_1	m_3	m_2
01	m_4	m_5	m_7	m_6
11	m_{12}	m_{13}	m_{15}	m_{14}
10	m_8	m_9	m_{11}	m_{10}

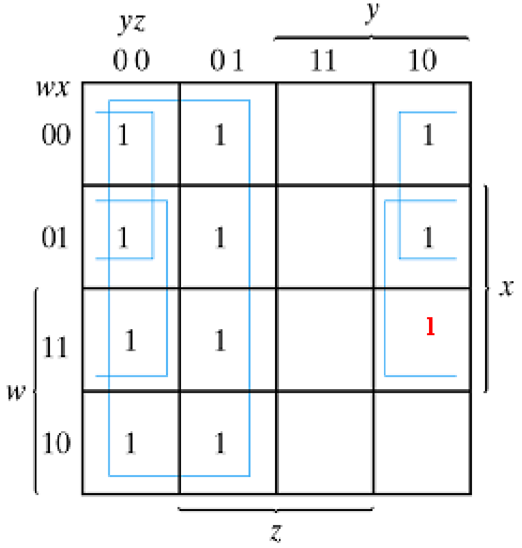


Fig. 3-9 Map for Example 3-5; $F(w, x, y, z) = \Sigma(0, 1, 2, 4, 5, 6, 8, 9, 12, 13, 14) = y' + w'z' + xz'$

Example 3-6

F = A′B′C′+B′CD′+A′BCD′+AB′C′

Solution:

- Each of three literal term in map is represented by two squares and four literal term in map is represented by one square
- We combine the 1's in the four corners to give the term B′D′
- The two left hand 1's in the top row are combined with two 1's in the bottom row to give the term B′C′
- The remaining 1's may be combined in the two-square area to give the term A′CD′
- The logical sum of these three terms gives:

F = B′D′ + B′C′+ A′CD′

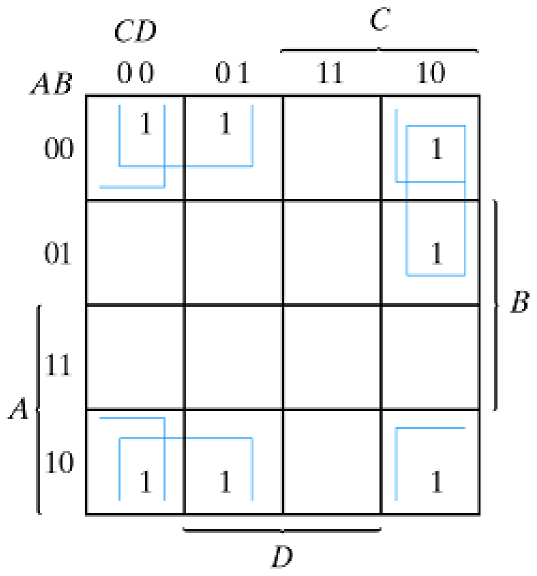
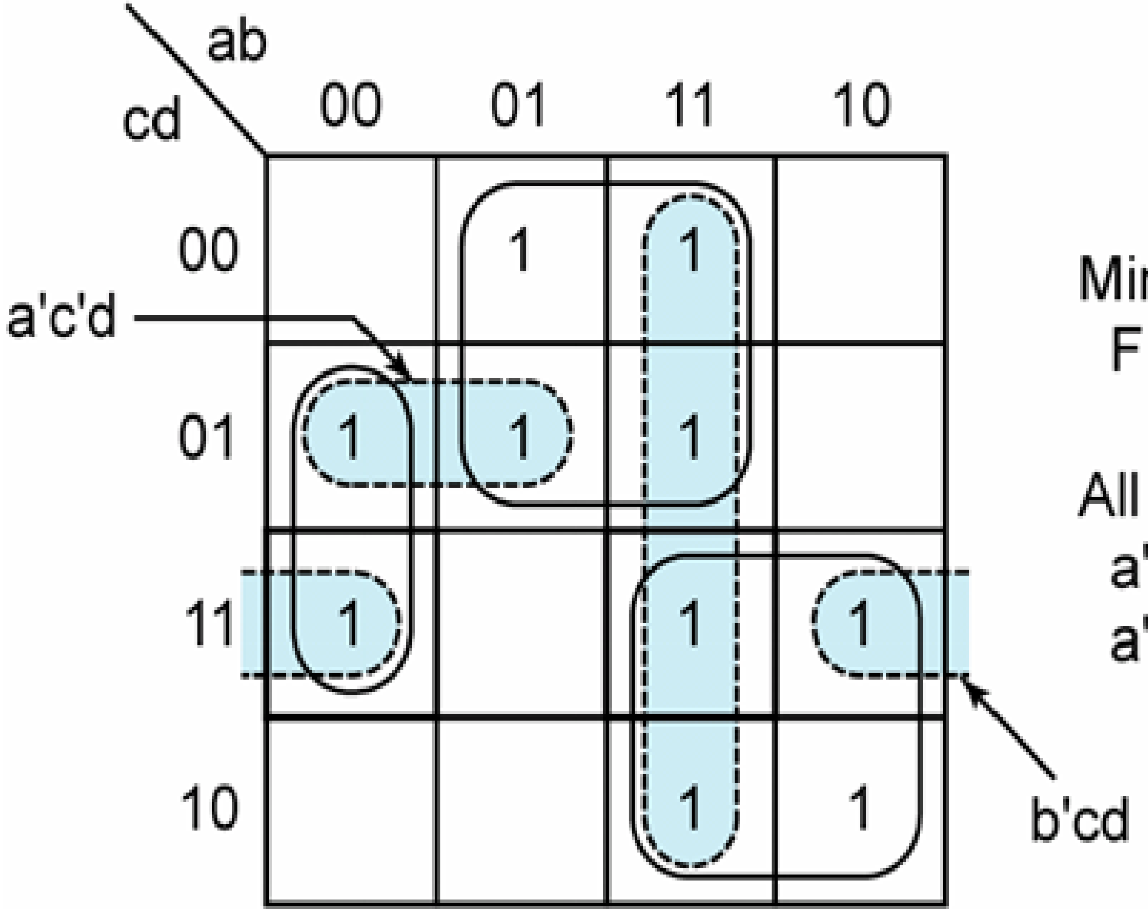


Fig.3-10 Map for Example 3-6; $A'B'C' + B'CD' + A'BCD' + AB'C' = B'D' + B'C' + A'CD'$

Prime Implicants Definitions

- **Implicant of a function F**
  a single 1 (= minterm) or any group of 1's which can be combined together in a K-map (i.e., 1's that are adjacent and which are grouped in a number that is always a power of 2). represents a product term which is called an implicant of a function F. An implicant represents a product term that can be used in a SOP expression for that function, that is, the function is 1 whenever the implicant is 1 (and maybe other times, as well …)
- **Prime Implicant**
  is an implicant that cannot be combined with another one to eliminate a literal. In other word each prime implicant corresponds to a product term in one of the minimum SOP expression for the function. A prime implicant is an implicant that is not fully contained in any other implicant.
- **Essential prime implicant**
  is a prime implicant that includes at least one 1 that is not included in any other prime implicant. In other words if a particular element of the on-set is covered by only one prime implicant, than that implicant is called an essential prime implicant.

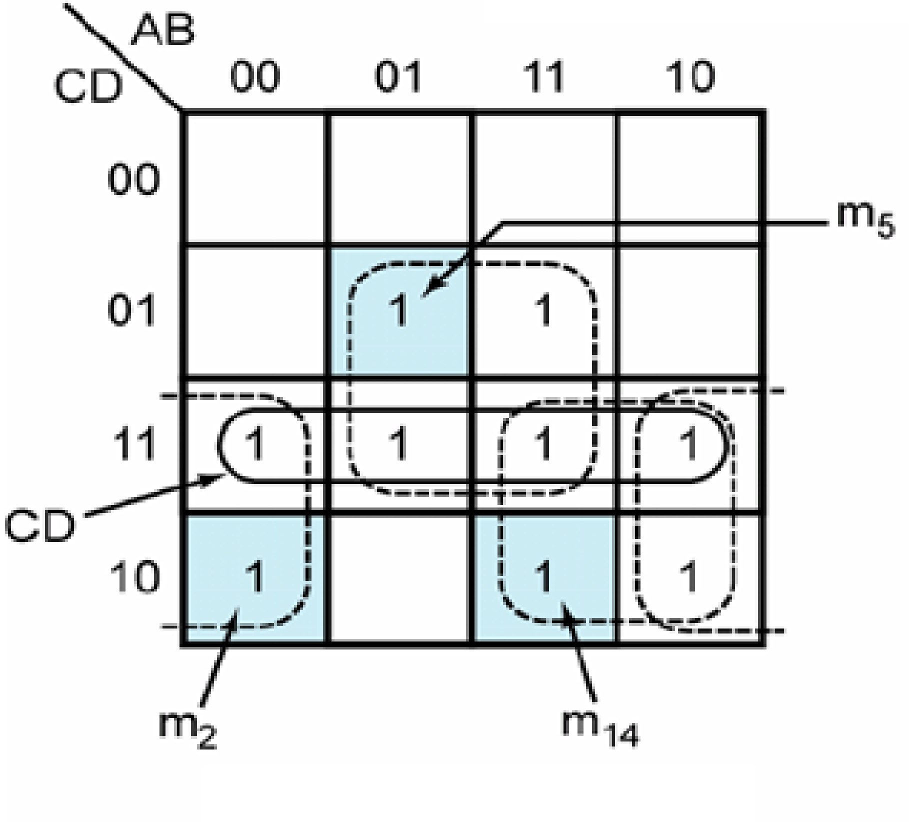
Example of Prime Implicants



Minimum solution:
F = a'b'd + bc' + ac

All prime implicants:
a'b'd, bc', ac,
a'c'd, ab, b'cd

Essential Prime Implicants



Essential Prime Implicants:

BD, AC, B′C

Distinguished 1-cells: m_2, m_5, m_{14}

Other Prime Implicants:

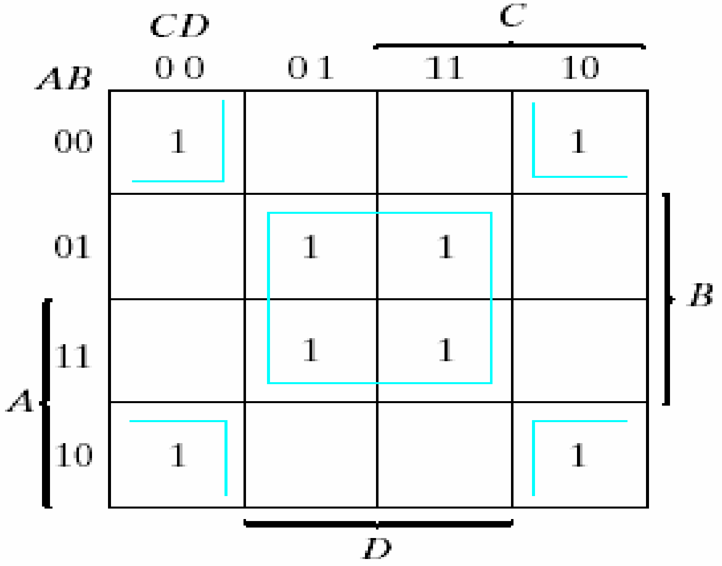
CD

F=BD+AC+B′C

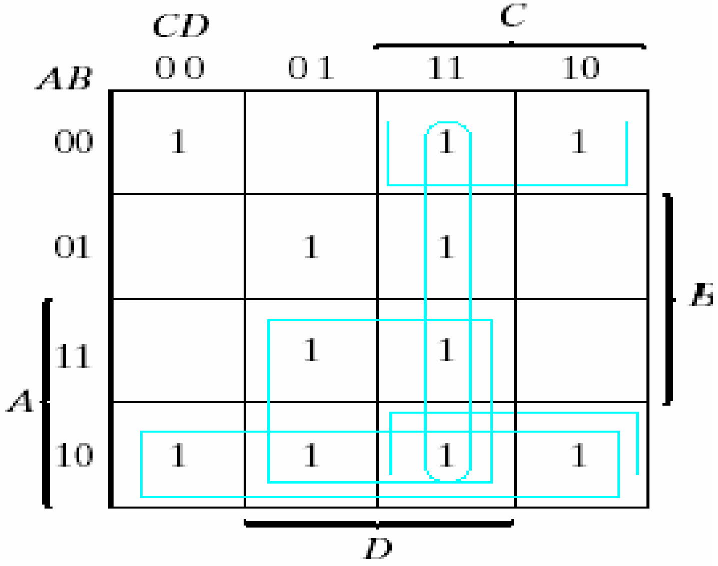
Functions with Multiple Solutions

Example: Find all the possible solutions of the following function using map method:-

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



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



(b) Prime implicants CD, B′C AD, and AB′

Functions with Multiple Solutions Cont…

- All PIs: BD, B′D′,CD, B′C, AD, AB′
- EPIs: BD, B′D′
- All possible solutions are:-

F=BD+B′D′+CD+AD

F=BD+B′D′+CD+AB′

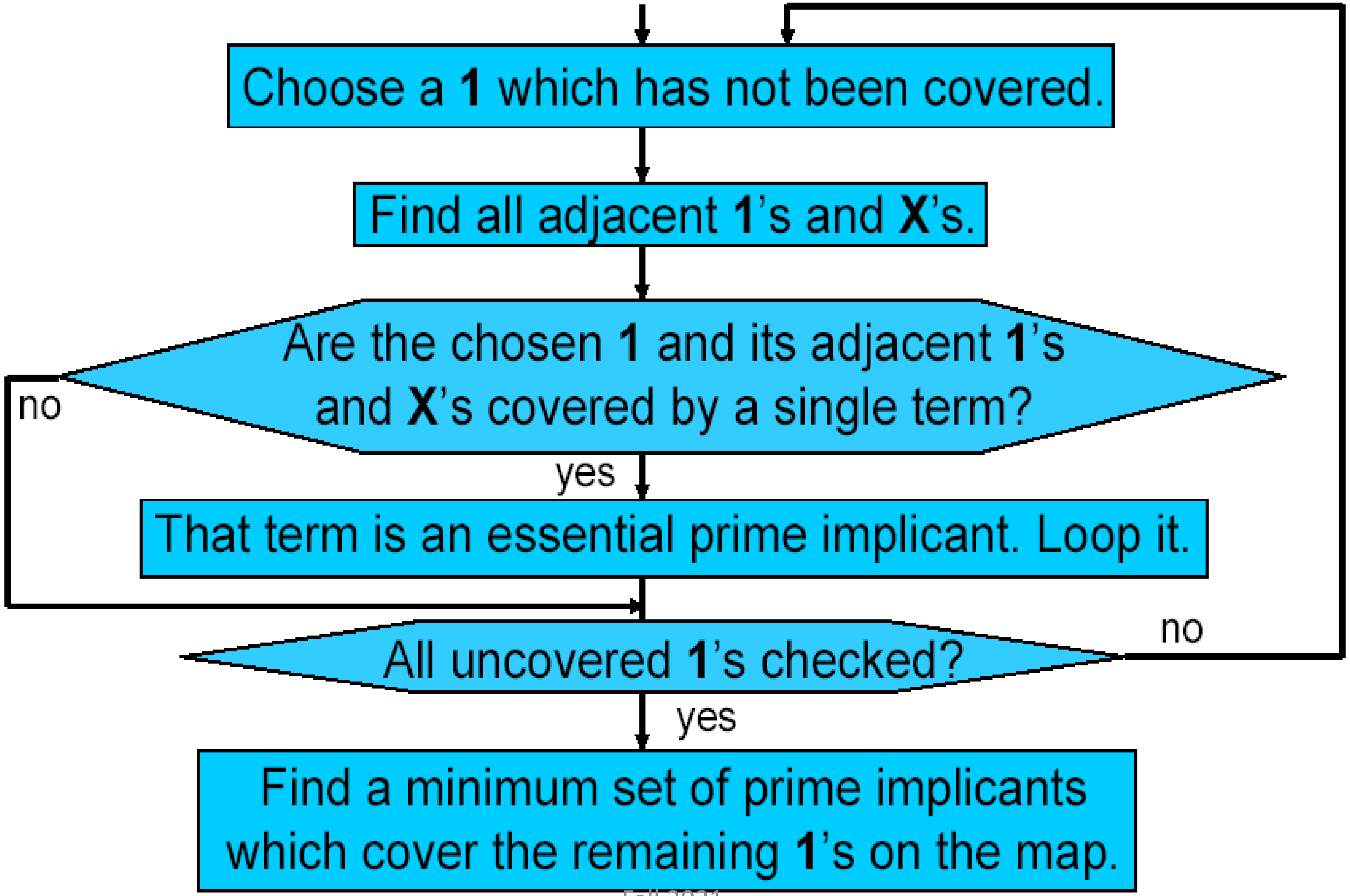
F=BD+B′D′+B′C+AD

F=BD+B′D′+B′C+AB′

OR

F= BD+B′D′+[(CD+AD) or (CD+AB′) or (B′C+AD) or (B′C+AB′)]

Algorithm for determining minimum SOP using a K-map



The End