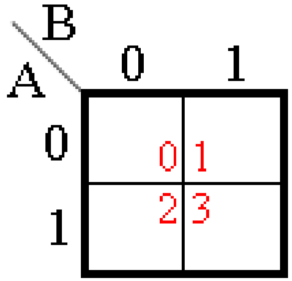
CHAPTER 4

THE KARNAUGH-VEITCH-MAP

Karnaugh Maps (K-Maps)

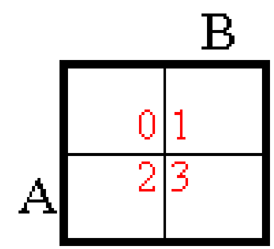
- **Alternate representation of a truth table**
  - ➢ Red decimal = minterm value
    - Note that A is the MSB for this minterm numbering
  - ➢ Adjacent squares have distance = 1
- **Valuable tool for logic minimization**
  - ➢ Applies most Boolean theorems & postulates automatically (when procedure is followed)

form 1



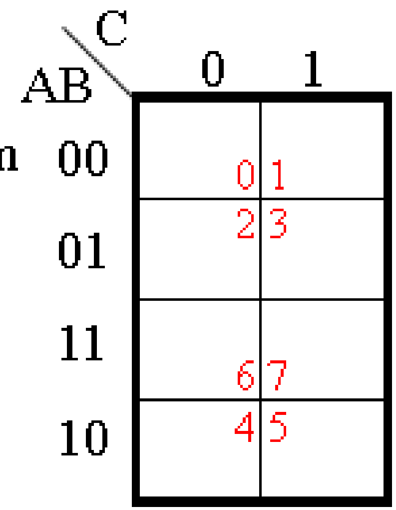
2-variable K-Maps

form 2

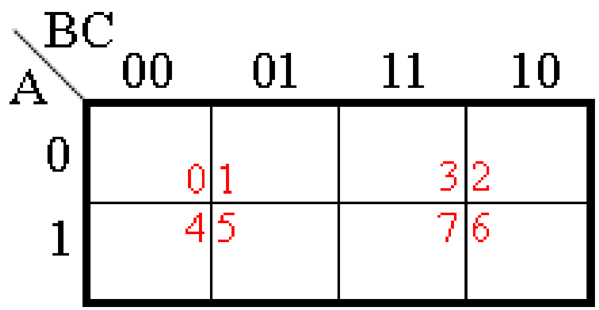


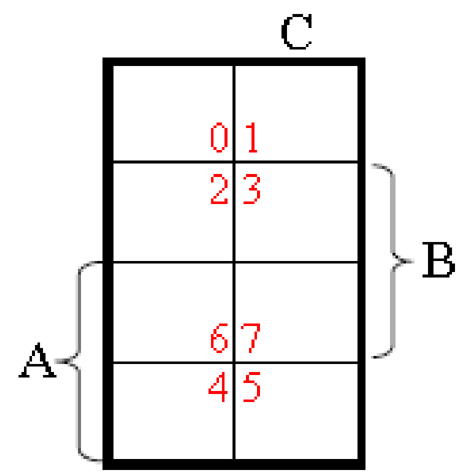
K-maps

- Alternate forms of 3-variable K-maps
  - ➢ Note end-around adjacency
    - Distance = 1
    - Note: A is MSB, C is LSB for minterm numbering

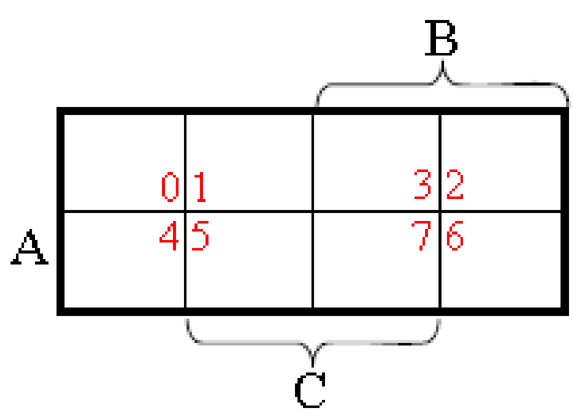


form 1





form 2



K-Maps

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

K-mapping and Minimization steps

Step 1: generate K-map

- Put a 1 in all specified minterms
- Put a 0 in all other boxes (optional)

Step 2: group all adjacent 1s without including any 0s

- All groups (aka *prime implicants*) must be rectangular and contain a "power-of-2" number of 1s
  - 1, 2, 4, 8, 16, 32, …
- An essential group (aka *essential prime implicant*) contains at least 1 minterm not included in any other groups
  - A given minterm may be included in multiple groups

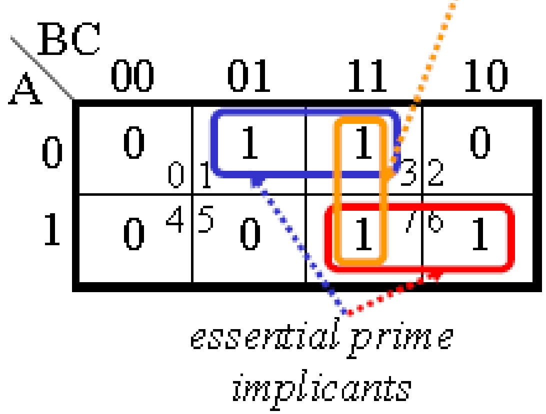
Step 3: define product terms using variables common to all minterms in group

Step 4: sum all essential groups plus a minimal set of remaining groups to obtain a minimum SOP

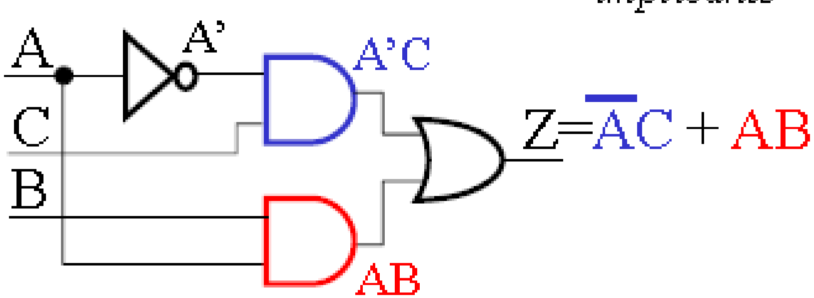
K-mapping example

- $Z=\Sigma_{A,B,C}(1,3,6,7)$
  - Recall SOP minterm implementation
    - 8 gates
    - 27 gate I/O
  - K-map results
    - 4 gates
    - 11 gate I/O

Note: this group not needed since 1s are already covered



A	B	C	Z	Row value
0	0	0	0	0
0	0	1	1	1
0	1	0	0	2
0	1	1	1	3
1	0	0	0	4
1	0	1	0	5
1	1	0	1	6
1	1	1	1	7

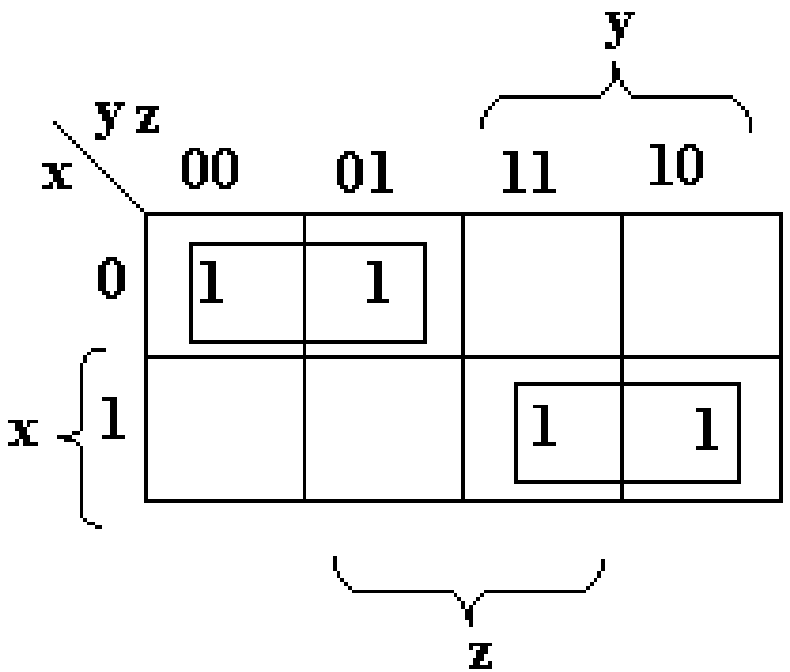


Determining the Minimum SOP Expression from the Map

- 1. Group the cells that have 1s.
- 2. Determine the minimum product terms for each group.
- For a 3-variable map:
  - (1) A 1-cell group yields a 3-variable product term
  - (2) A 2-cell group yields a 2-variable product term
  - (3) A 4-cell group yields a 1-variable term
  - (4) An 8-cell group yields a value of 1 for the expression
- For a 4-variable map
  - ?
- For a 5-variable map
  - ?
- 3. When all the minimum product terms are derived from the Karnaugh map, they are summed to form the minimum SOP expression.

EXAMPLE

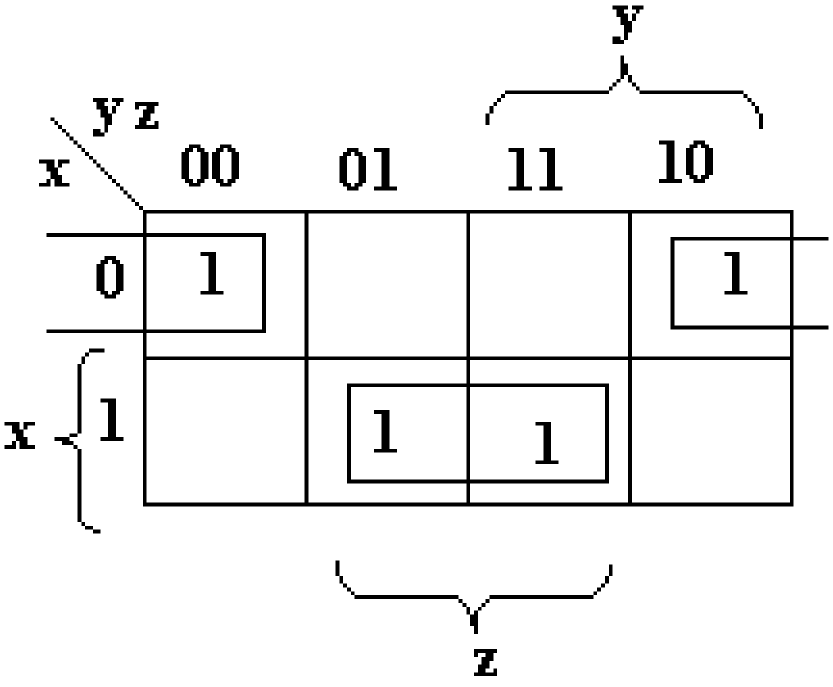
- Simplify the Boolean expression:
  - $F(x,y,z) = \Sigma\ (0,1,6,7)$



$F = x'y' + xy$

EXAMPLE:

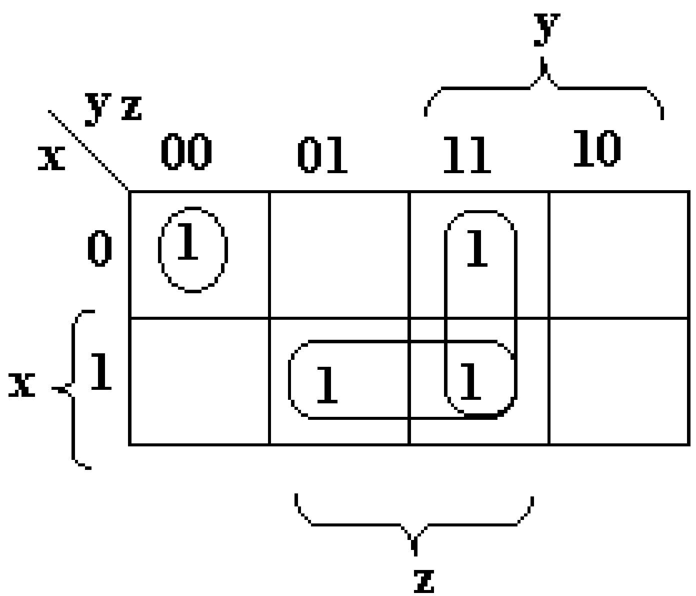
- Simplify the Boolean expression:
  $F(x,y,z) = \Sigma\,(0,2,5,7)$



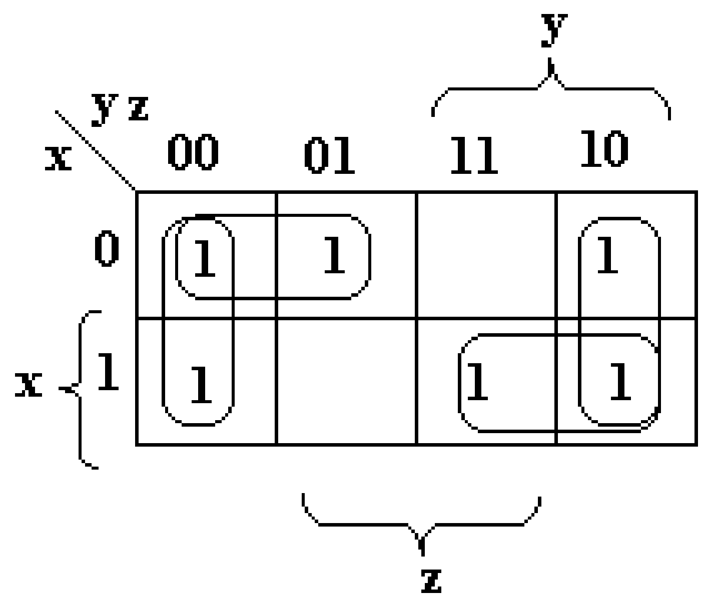
$F = x'z' + xz$

EXAMPLE:

- Group the 1's in each of the following Karnaugh maps
- The groups y’z’+ yz’ should be added to be one group z’:



$F = x'y'z' + xz + yz$



$F = y'z' + x'y' + xy + yz'$

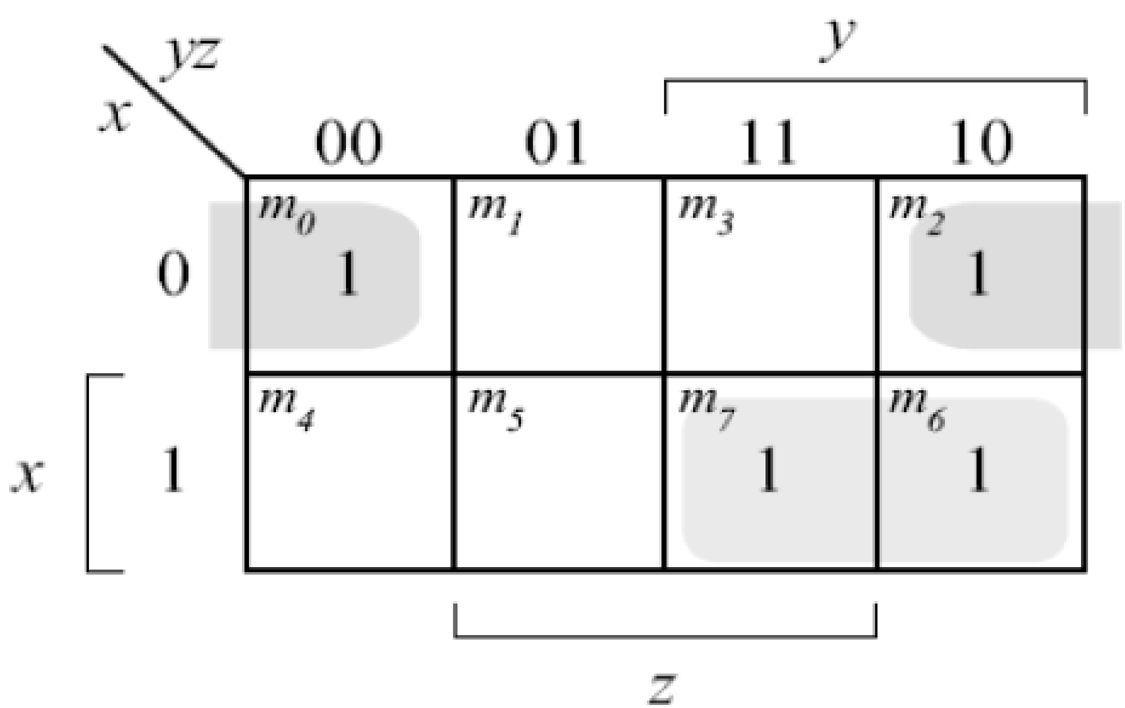
More Examples

- Minimize the following SOP expression,
  - $F(x, y, z) = \Sigma(0,2,6,7)$ → 1
  - $F(x, y, z) = \Sigma(0,2,3,4,6)$ → 2
  - $F(x, y, z) = \Sigma(0,1,2,3,7)$ → 3
  - $F(x, y, z) = \Sigma(3,5,6,7)$ → 4
  - $F(x, y, z) = \Sigma(0,1,5,7)$ → 5
  - $F(x, y, z) = \Sigma(0,1,6,7)$ → 6
  - $F(x, y, z) = \Sigma(1,2,3,6,7)$ → 7

Solutions

Solution

- $F(x, y, z) = \Sigma(0,2,6,7)$ → 1

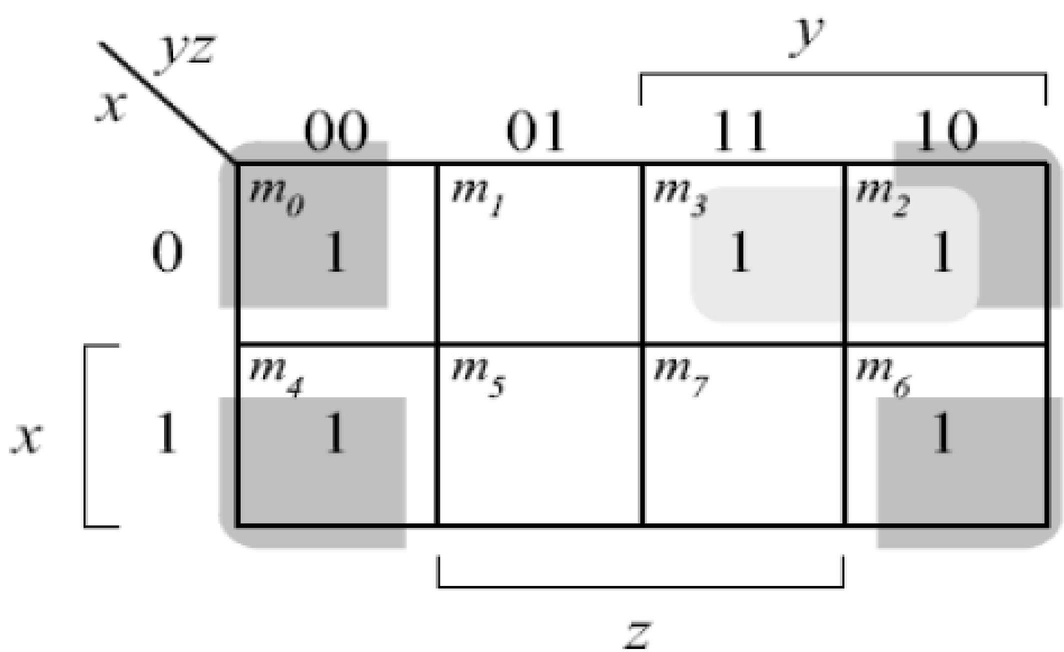


F = x y + x' z'

Solutions

- **Solution**

  $F(x, y, z) = \Sigma(0,2,3,4,6)$ → 2

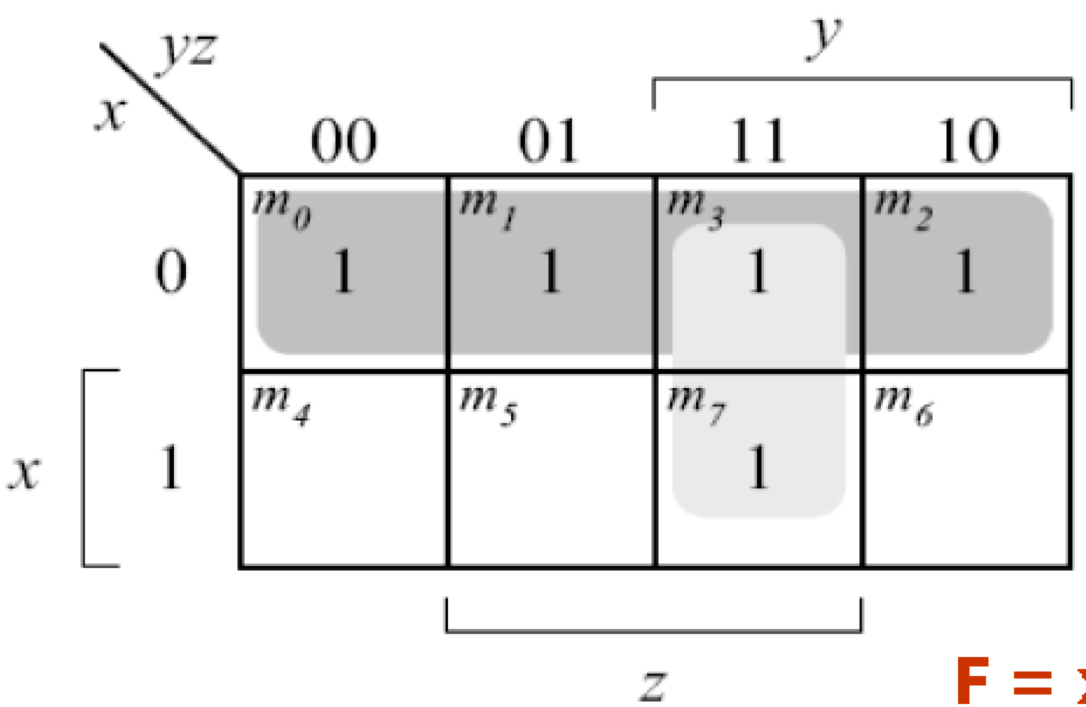


F = z' + x'y

Solutions

- **Solution**

  F(x, y, z) = Σ(0,1,2,3,7) ➔ 3

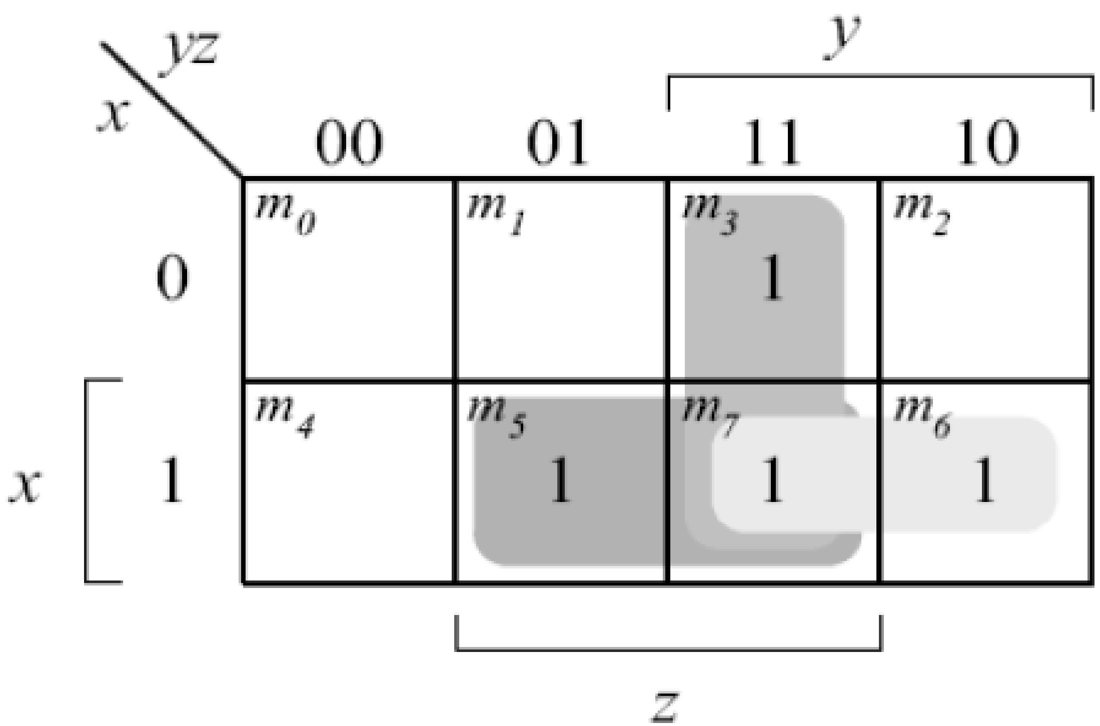


F = x' + y z

Solutions

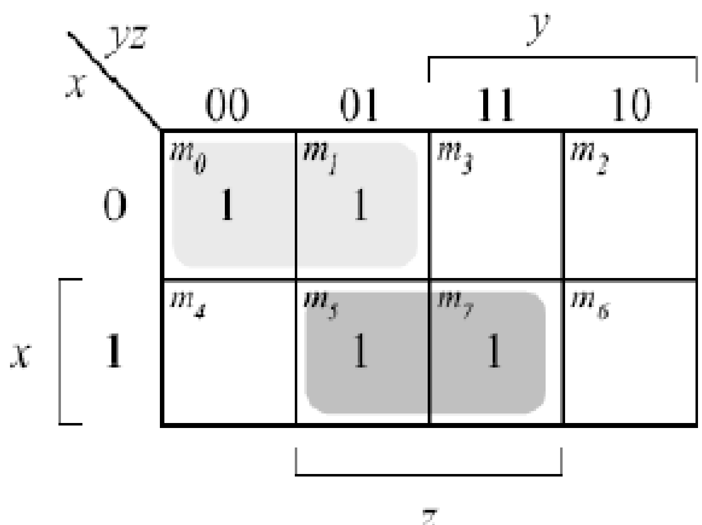
- **Solution**

F(x, y, z) = Σ(3,5,6,7) → 4

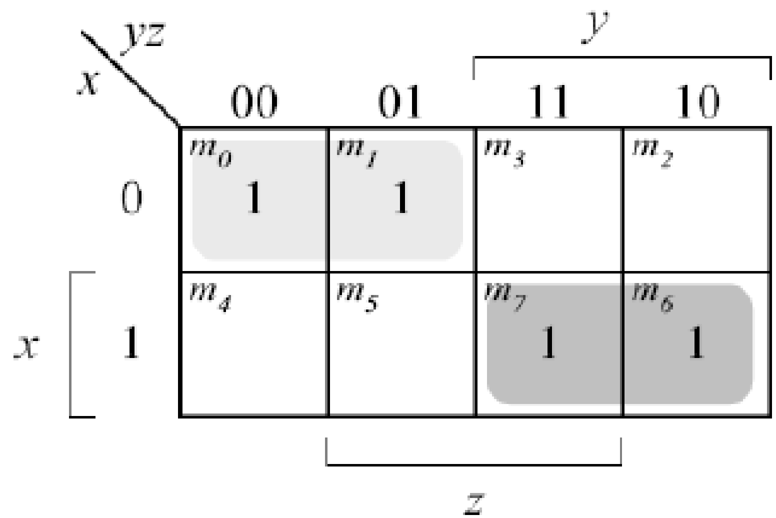


F = xy + xz + yz

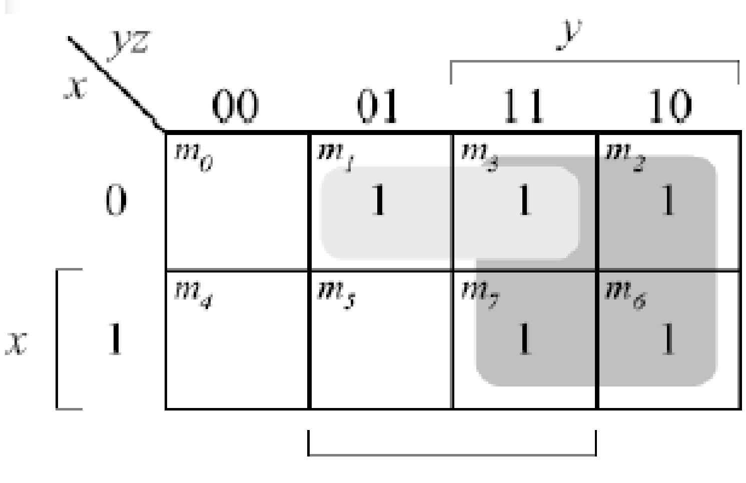
Solutions



5. F = x' y' + x z



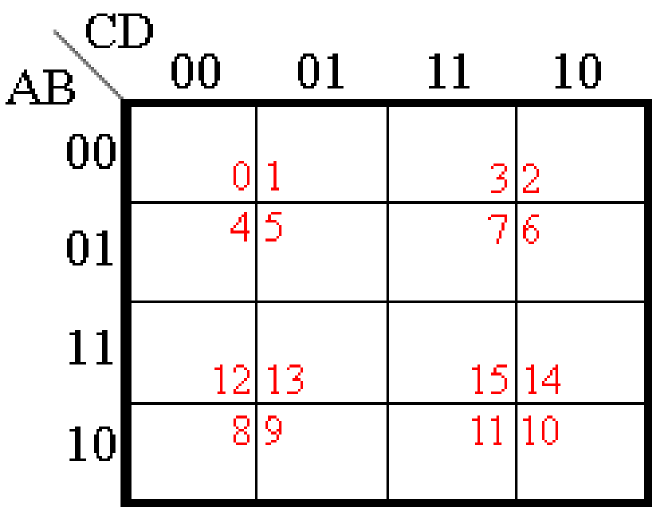
6. F = x' y' + x y



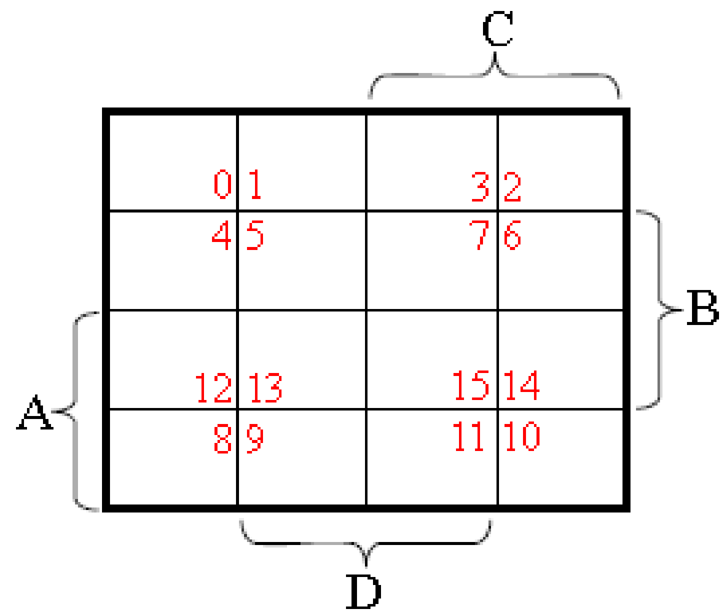
7. F = y + x' z

4-variable K-Map

- Note adjacency of 4 corners as well as sides
- Variable ordering for this minterm numbering: ABCD

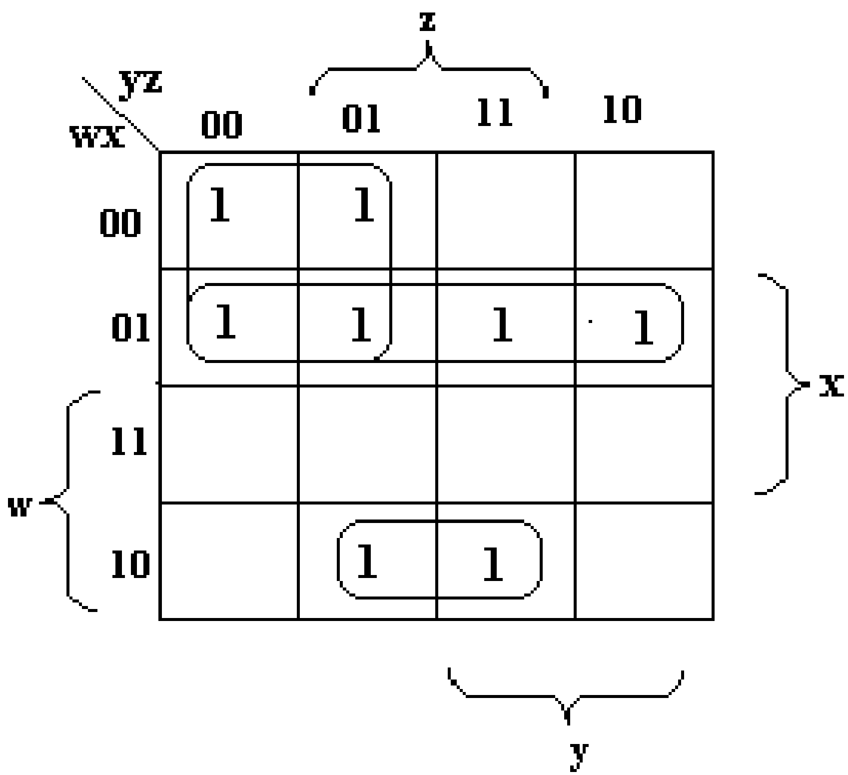


form 1

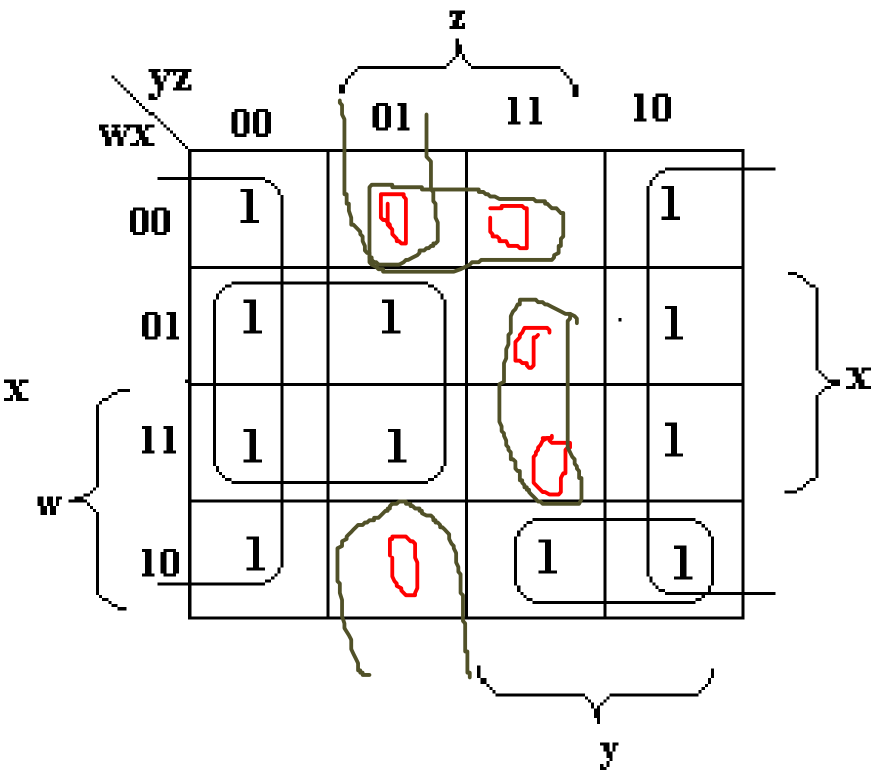


form 2

EXAMPLE:



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



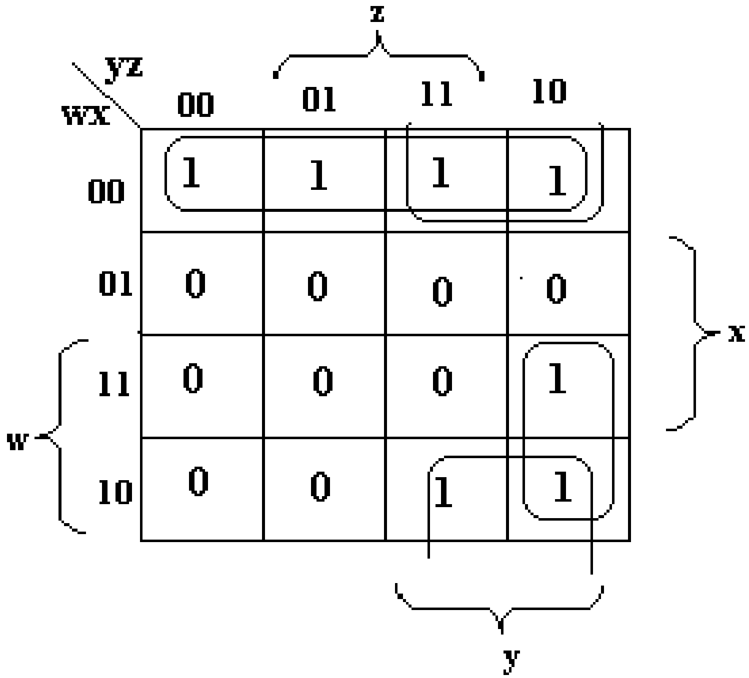
F = xy' + z' +wx'y

KARNAUGH MAP PRODUCT OF SUM (POS) SIMPLIFICATION

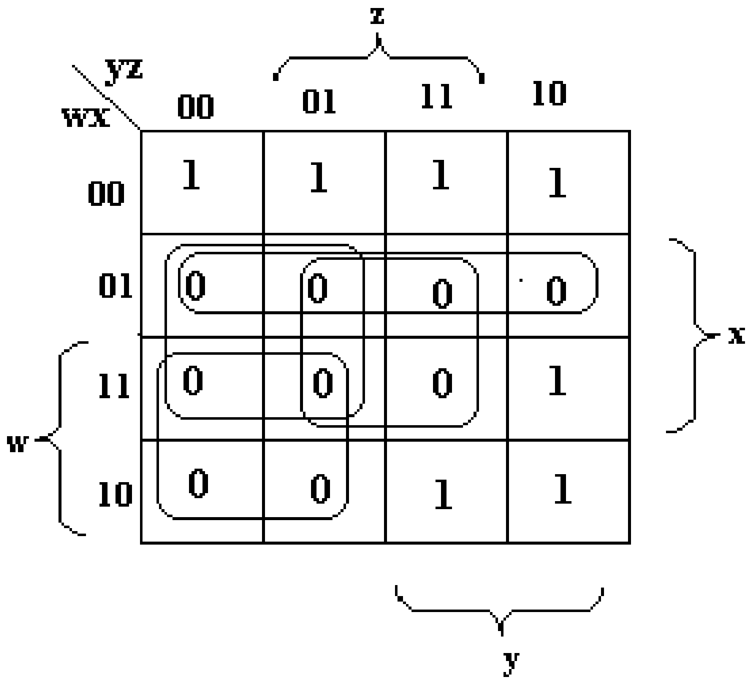
- If we mark the empty squares by 0's and combine them into valid adjacent squares, we obtain a simplified expression of the complement of the function, i.e., of F'.
- The complement of *F′* gives us back the function F.
- Because of Demorgan's theorem, the function so obtained is automatically in the product of sums form.

EXAMPLE

- Simplify the following Boolean function in (a) sum of products and (b) product of sums.
- $F(w,x,y,z) = \Sigma\,(0,1,2,3,10,11,14)$
- The group xy' is redundant and should be removed



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

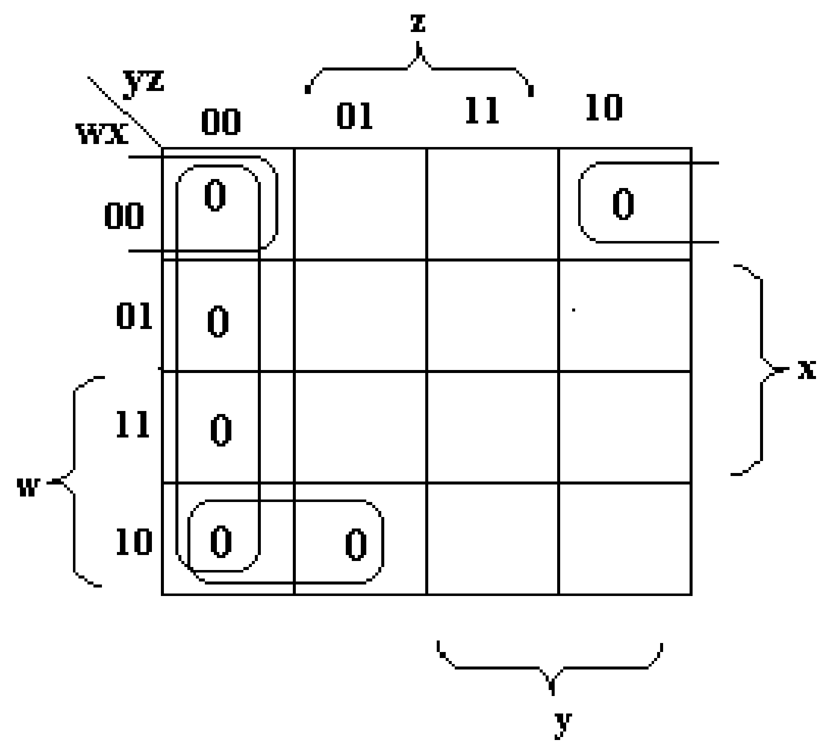


$F' = wy' + xy' + xz + w'x$

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

EXAMPLE

- Use a Karnaugh map to minimize the following POS expression.
- (x+y+z)(w+x+y'+z)
  (w'+x+y+z')
  (w+x'+y+z)
  (w'+x'+y+z)
- (w+x+y+z)(w'+x+y+z)(w+x+y'+z)(w'+x+y+z')
  (w+x'+y+z)(w'+x'+y+z
)
- =Π(0,8,2,9,4,12)

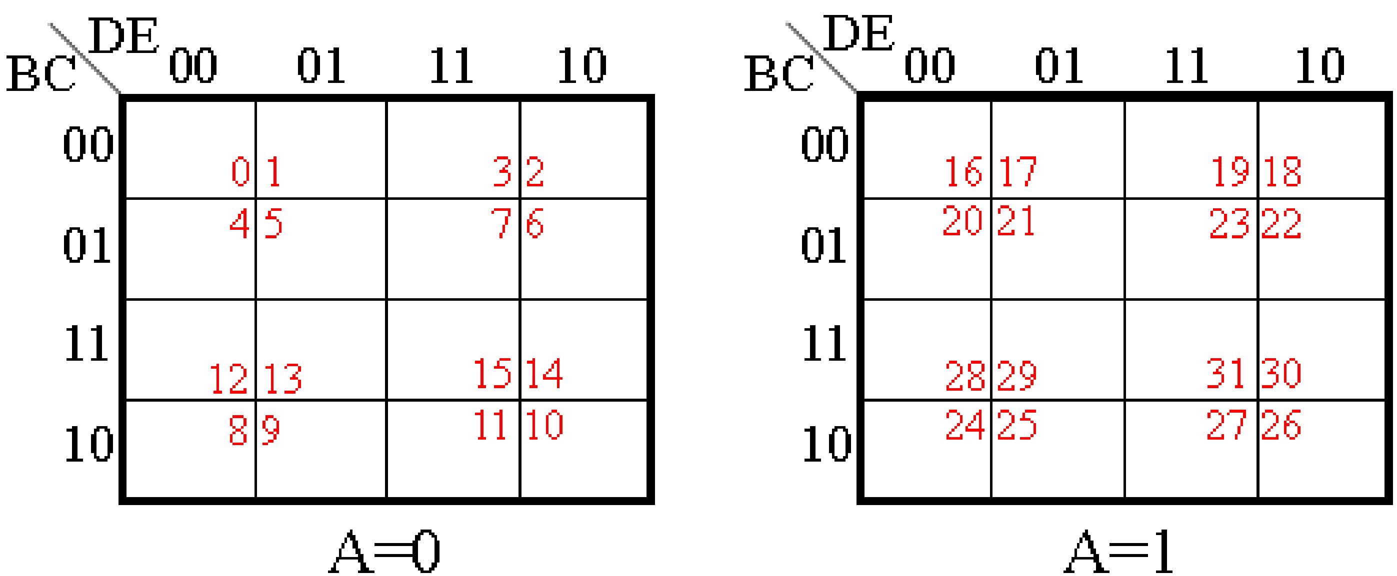


$F' = y'z' + wx'y' + w'x'z'$

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

5-Variable K-Map

- Note adjacency between maps when overlayed
  - distance=1
- Variable order for this minterm numbering:
  - A,B,C,D,E (A is MSB, E is LSB)



DON'T CARE CONDITIONS

- Sometimes input combinations are of no concern
  - Because they may not exist
    - Example: BCD uses only 10 of possible 16 input combinations
  - Since we “don’t care” what the output, we can use these “don’t care” conditions for logic minimization
    - The output for a don’t care condition can be either 0 or 1
      - ✓ WE DON’T CARE!!!
- Don’t Care conditions denoted by:
  - X, -, d, 2
    - X is probably the most often used
- Can also be used to denote inputs
  - Example: ABC = 1X1 = AC
    - B can be a 0 or a 1

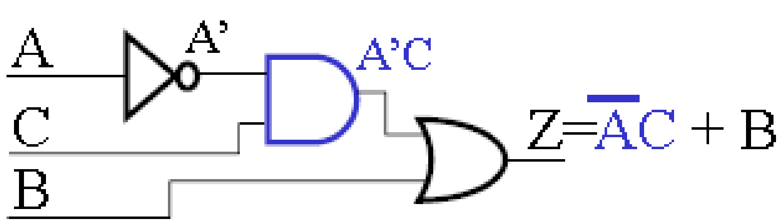
Example

- Truth Table
- K-map
- Minterm
  - ➢ $Z=\Sigma_{A,B,C}(1,3,6,7)+d(2)$
- Maxterm
  - ➢ $Z=\Pi_{A,B,C}(0,4,5)+d(2)$
    - Notice Don't Cares are same for both minterm & maxterm

A \ BC	00	01	11	10
0	0 (0)	1 (1)	1 (3)	X (2)
1	0 (4)	0 (5)	1 (7)	1 (6)

Z=B+A'C

A	B	C	Z
0	0	0	0
0	0	1	1
0	1	0	X
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1



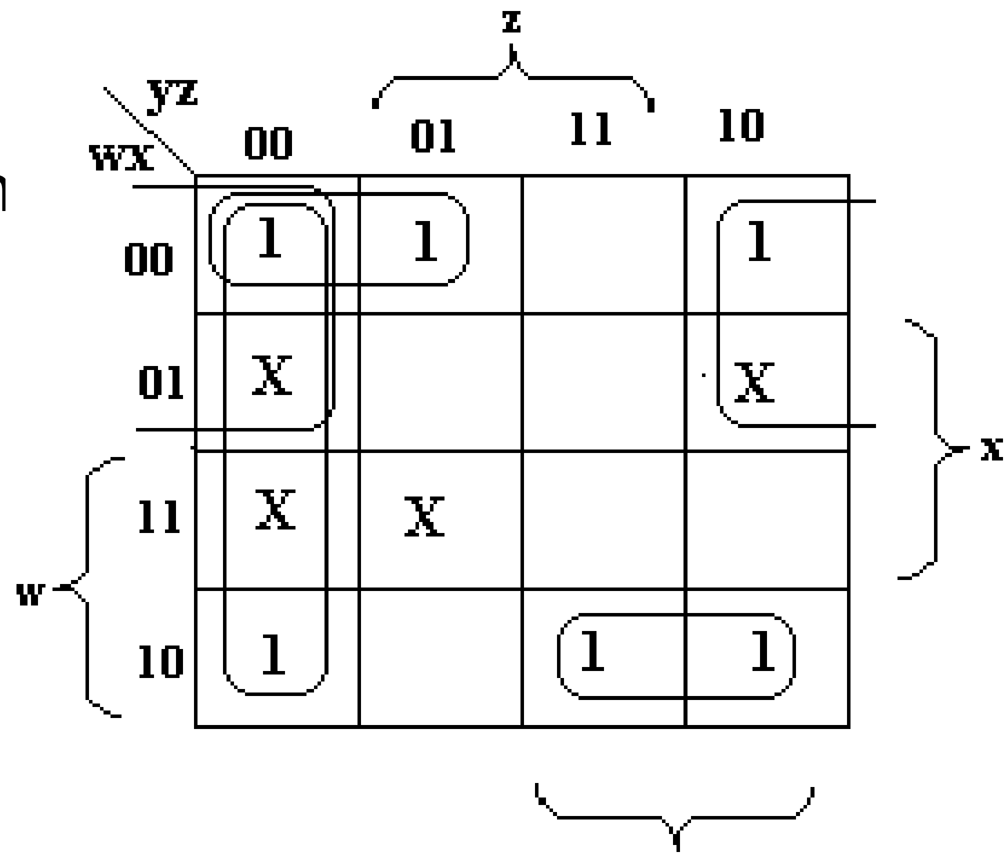
Circuit analysis:

$G=3$ $G_{IO}=8$

(compared to $G=4$ & $G_{IO}=11$ w/o don't care)

Example

- Simplify the following Boolean function F, where d represents th set of do not care conditions
- F(w,x,y,z) = Σ (0,1,2,8,10,11)
- d(w,x,y,z) = Σ (4,6,12,13)



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

QUESTIONS?