# Chapter 04:Simplification of Boolean function using K-MAP

Er. Hari K.C.

Department of software Engineering

Gandaki college of Engineering and science

# K -Map

- In previous chapters, we have simplified the Boolean functions using Boolean postulates and theorems.
- It is a time consuming process and we have to re-write the simplified expressions after each step.
- To overcome this difficulty, **Karnaugh** introduced a method for simplification of Boolean functions in an easy way.
- This method is known as Karnaugh map method or K-map method. It is a graphical method, which consists of $2^n$ cells for ‘n’ variables.
- The adjacent cells are differed only in single bit position.

# K-Map

$Y = \bar{A}B + A\bar{B} + \bar{A}\bar{B}$

SOP

Sum of products

minterms

- A Karnaugh map provides a pictorial method of grouping together expressions with common factors and therefore eliminating unwanted variables.
- The Karnaugh map can also be described as a special arrangement of a truth table.
- It is used for simplification of Boolean expression.

POS

$Y = (A+B) \cdot (\bar{A}+\bar{B}) \cdot (\bar{A}+B)$

maxterms

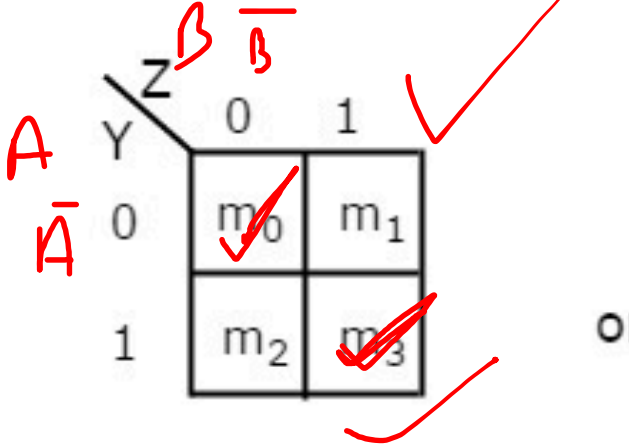
# Two variable K-Map

$y = \bar{A}\bar{B} + AB = m_0 + m_3$

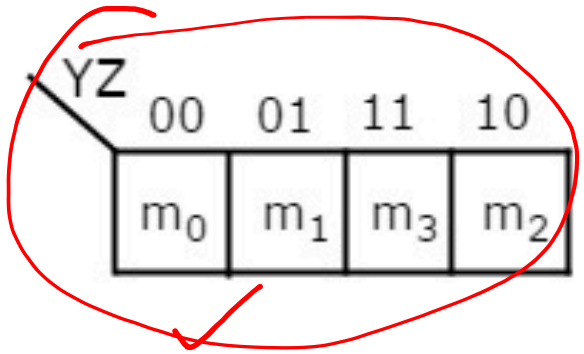
$y = \sum m(0,3)$

## 2 Variable K-Map

The number of cells in 2 variable K-map is four, since the number of variables is two. The following figure shows **2 variable K-Map**.




or




| | | minterm | |
|---|---|---|---|
| 00 | A'B' | mo | |
| 01 | A'B | m1 | |
| 10 | AB' | m2 | |
| 11 | AB | m3 | |

- There is only one possibility of grouping 4 adjacent min terms.
- The possible combinations of grouping 2 adjacent min terms are {($m_0$, $m_1$), ($m_2$, $m_3$), ($m_0$, $m_2$) and ($m_1$, $m_3$)}.

# Three variable K- Map

$y = \bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + ABC$

$m_1 + m_3 + m_7$

$y = \sum m(1,3,7)$

## 3 Variable K-Map

The number of cells in 3 variable K-map is eight, since the number of variables is three. The following figure shows **3 variable K-Map**.

| X \ YZ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | $m_0$ | $m_1$ | $m_3$ | $m_2$ |
| 1 | $m_4$ | $m_5$ | $m_7$ | $m_6$ |

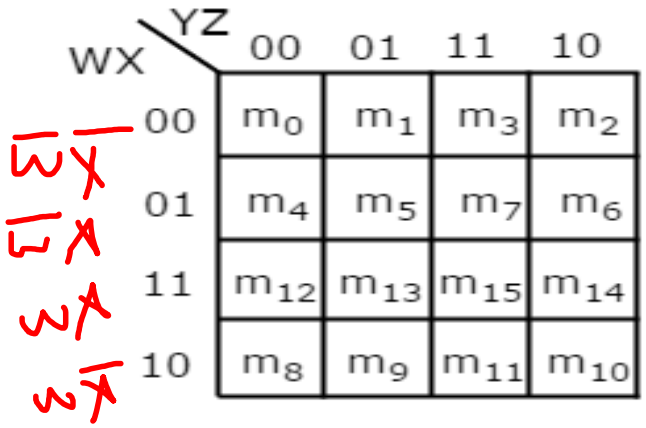
- There is only one possibility of grouping 8 adjacent min terms.
- The possible combinations of grouping 4 adjacent min terms are {($m_0$, $m_1$, $m_3$, $m_2$), ($m_4$, $m_5$, $m_7$, $m_6$), ($m_0$, $m_1$, $m_4$, $m_5$), ($m_1$, $m_3$, $m_5$, $m_7$), ($m_3$, $m_2$, $m_7$, $m_6$) and ($m_2$, $m_0$, $m_6$, $m_4$)}.
- The possible combinations of grouping 2 adjacent min terms are {($m_0$, $m_1$), ($m_1$, $m_3$), ($m_3$, $m_2$), ($m_2$, $m_0$), ($m_4$, $m_5$), ($m_5$, $m_7$), ($m_7$, $m_6$), ($m_6$, $m_4$), ($m_0$, $m_4$), ($m_1$, $m_5$), ($m_3$, $m_7$) and ($m_2$, $m_6$)}.
- If x=0, then 3 variable K-map becomes 2 variable K-map.

| | | minterm |
|---|---|---|
| 000 | A'B'C' | mo |
| 001 | A'B'C | m1 |
| 010 | A'BC' | m2 |
| 011 | A'BC | m3 |
| 100 | AB'C' | m4 |
| 101 | AB'C | m5 |
| 110 | ABC' | m6 |
| 111 | ABC | m7 |

# Four variable K - Map

## 4 Variable K-Map

The number of cells in 4 variable K-map is sixteen, since the number of variables is four. The following figure shows **4 variable K-Map**.

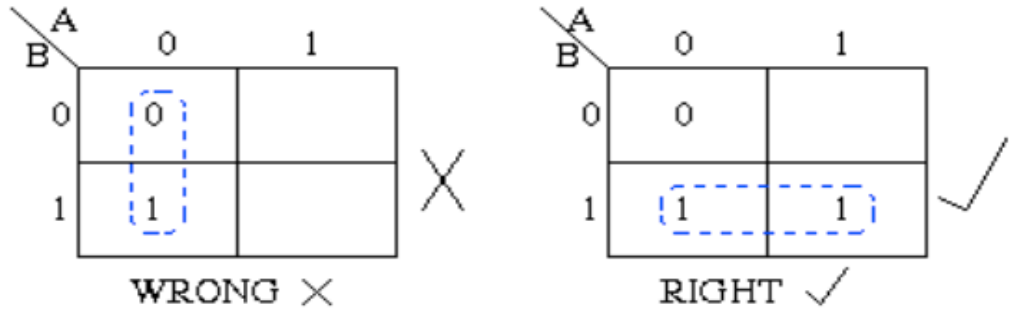



- There is only one possibility of grouping 16 adjacent min terms.
- Let $R_1$, $R_2$, $R_3$ and $R_4$ represents the min terms of first row, second row, third row and fourth row respectively. Similarly, $C_1$, $C_2$, $C_3$ and $C_4$ represents the min terms of first column, second column, third column and fourth column respectively. The possible combinations of grouping 8 adjacent min terms are {($R_1$, $R_2$), ($R_2$, $R_3$), ($R_3$, $R_4$), ($R_4$, $R_1$), ($C_1$, $C_2$), ($C_2$, $C_3$), ($C_3$, $C_4$), ($C_4$, $C_1$)}.
- If w=0, then 4 variable K-map becomes 3 variable K-map.

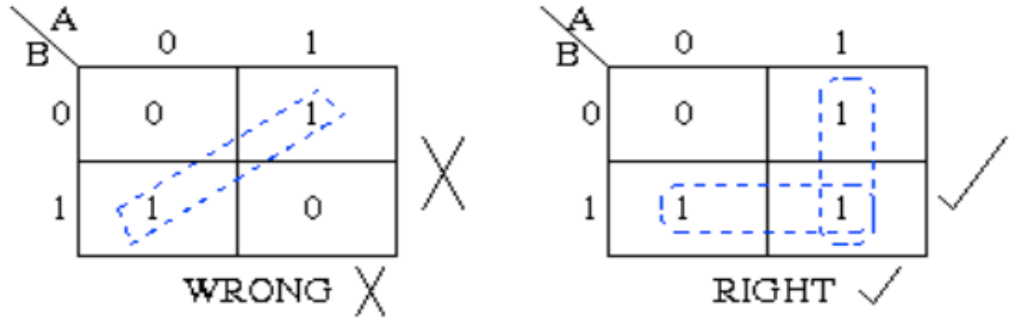
| | | minterm |
|---|---|---|
| 0000 | A’B’C’D’ | mo |
| 0001 | A’B’C’D | m1 |
| 0010 | A’B’CD’ | m2 |
| 0011 | A’B’CD | m3 |
| 0100 | A’BC’D’ | m4 |
| 0101 | A’BC’D | m5 |
| 0110 | A’BCD’ | m6 |
| 0111 | A’BCD | m7 |
| 1000 | AB’C’D’ | m8 |
| 1001 | AB’C’D | m9 |
| 1010 | AB’CD’ | m10 |
| 1011 | AB’CD | m11 |
| 1100 | ABC’D’ | m12 |
| 1101 | ABC’D | m13 |
| 1110 | ABCD’ | m14 |
| 1111 | ABCD | m15 |

# Rules for simplification in K map

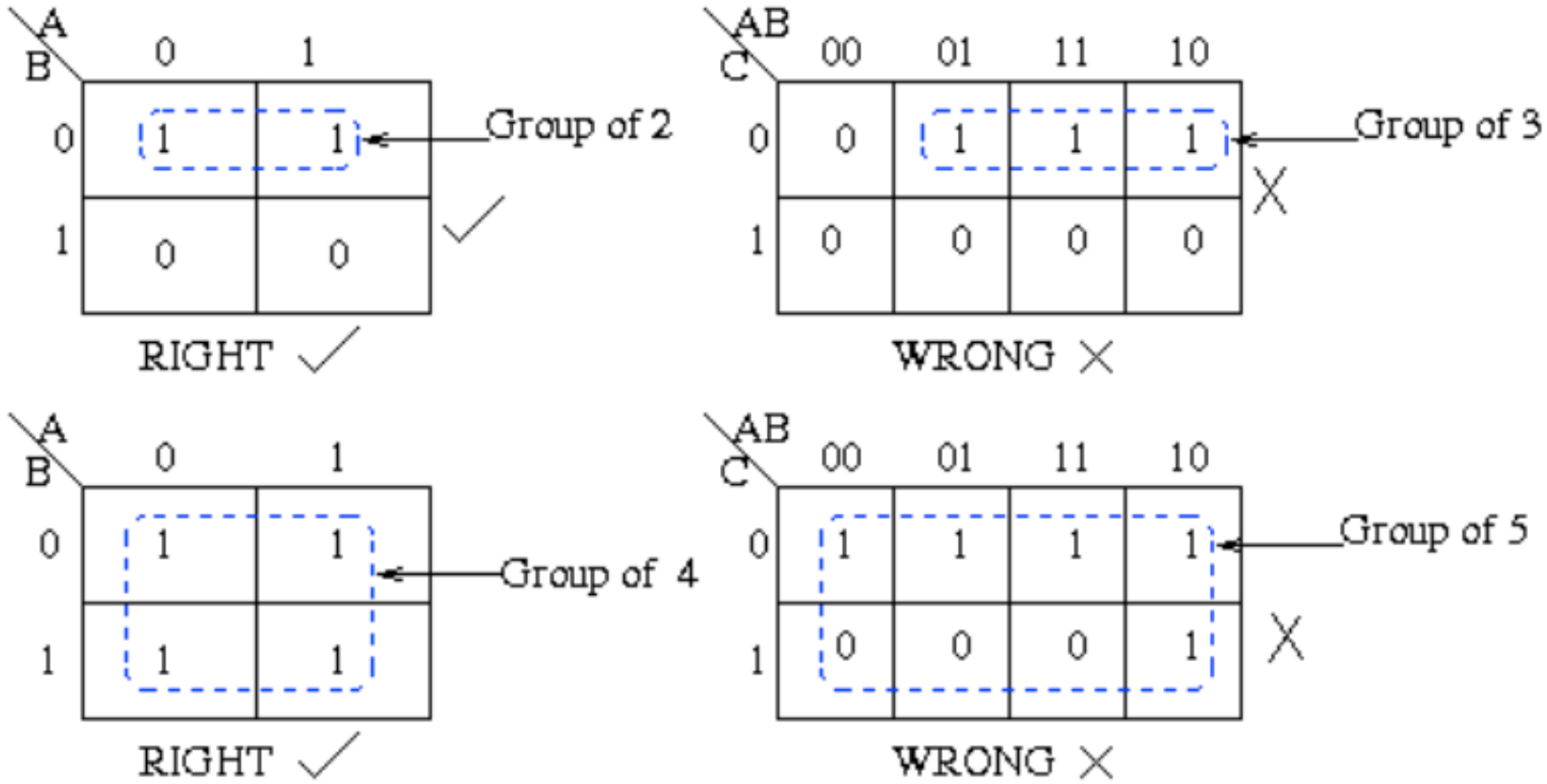
- The Karnaugh map uses the following rules for the simplification of expressions by grouping together adjacent cells containing ones.
-
  - **Groups may not include any cell containing a zero**




  - **Groups may be horizontal or vertical, but not diagonal.**

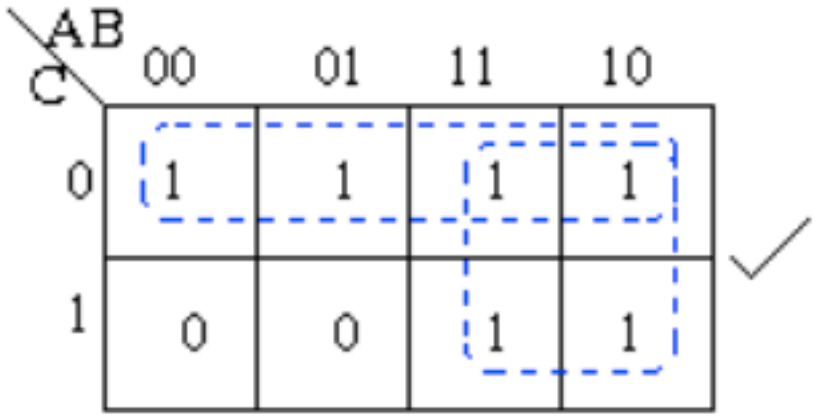


- **Groups must contain 1, 2, 4, 8, or in general $2^n$ cells.**
  **That is if n = 1, a group will contain two 1's since $2^1 = 2$.**
  **If n = 2, a group will contain four 1's since $2^2 = 4$.**

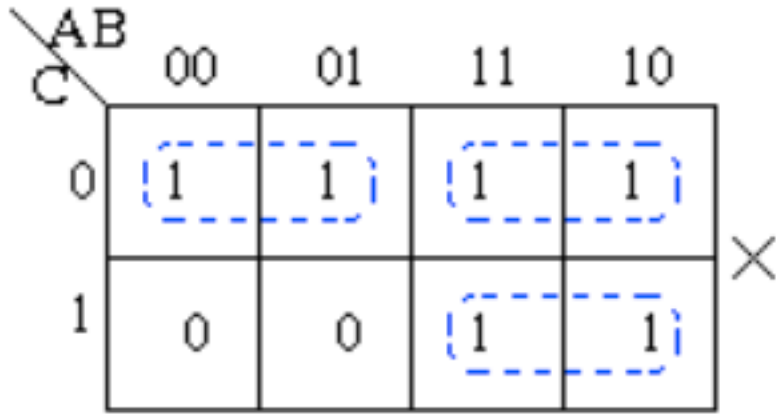



- **Each group should be as large as possible.**

## Karnaugh Maps - Rules of Simplification



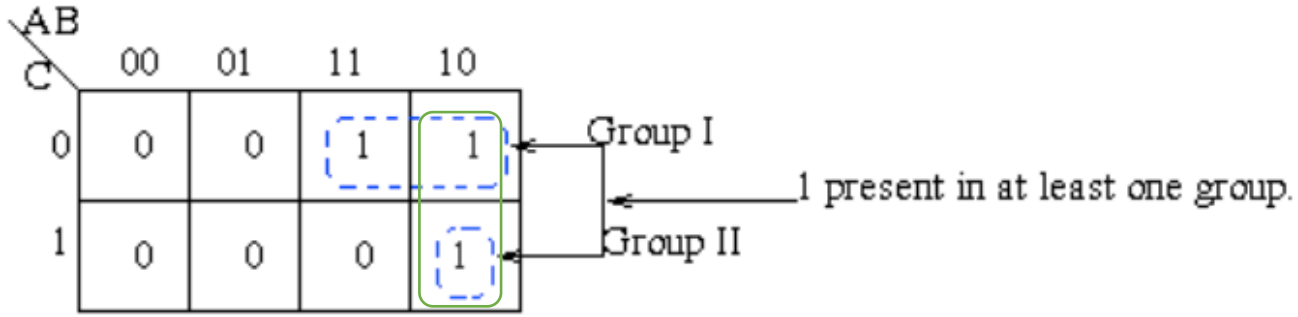

RIGHT ✓



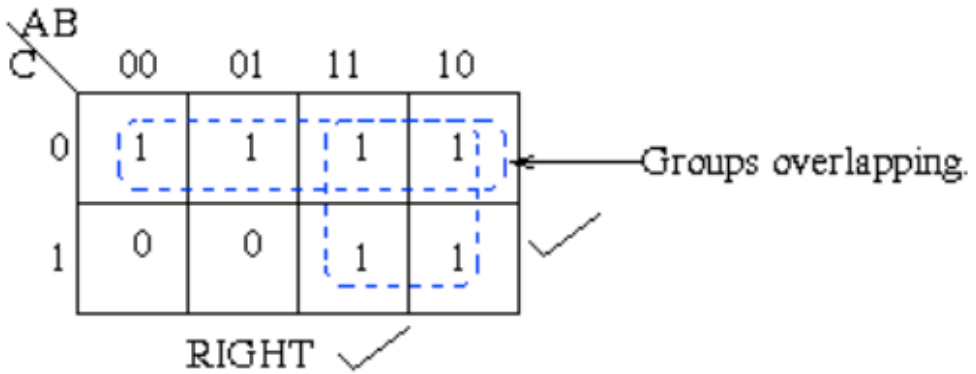

WRONG ×

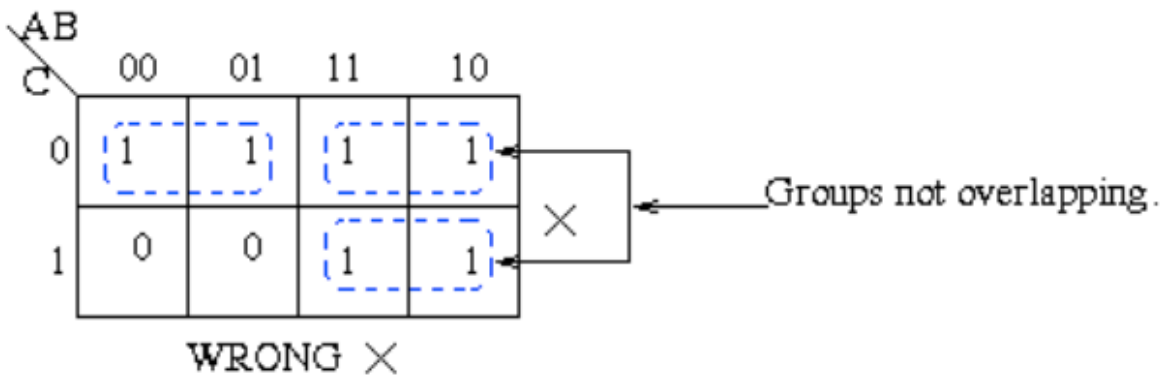
(Note that no Boolean laws broken, but not sufficiently minimal)

- **Each cell containing a *one* must be in at least one group.**

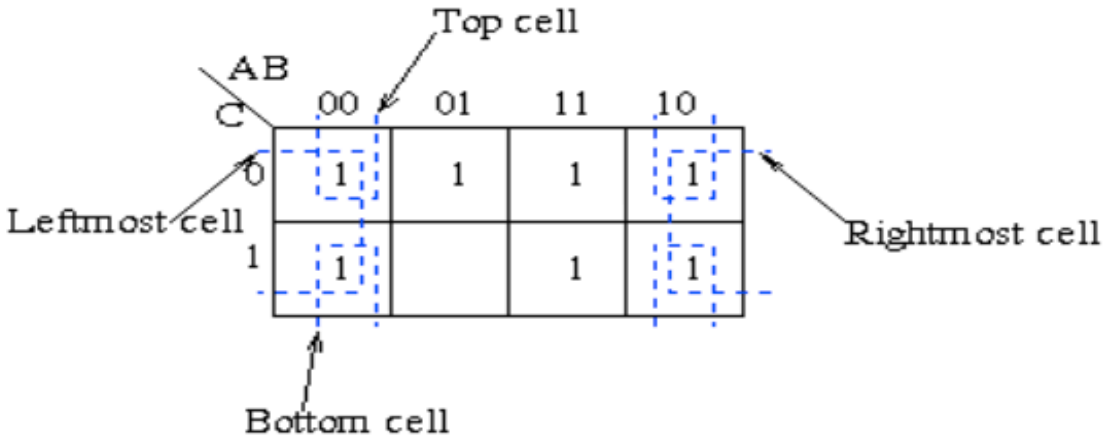



- **Groups may overlap.**

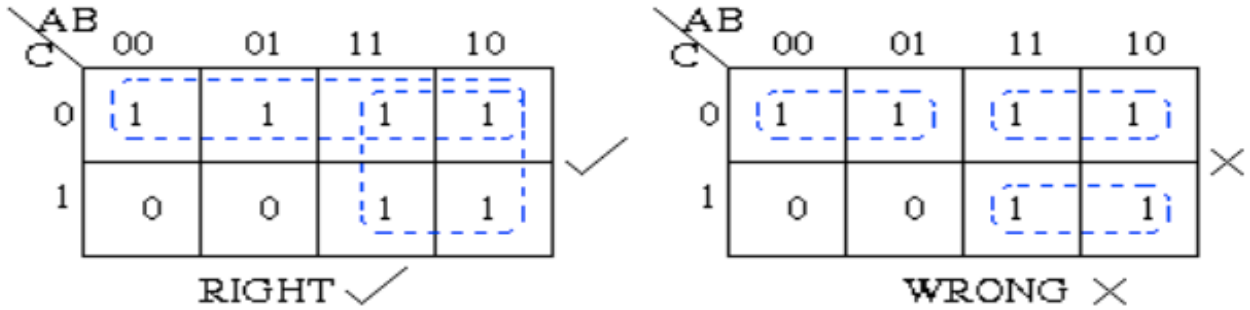


- Groups may wrap around the table. The leftmost cell in a row may be grouped with the rightmost cell and the top cell in a column may be grouped with the bottom cell.




- **There should be as few groups as possible, as long as this does not contradict any of the previous rules.**



**Summmary:**

1. No zeros allowed.
2. No diagonals.
3. Only power of 2 number of cells in each group.
4. Groups should be as large as possible.
5. Every one must be in at least one group.
6. Overlapping allowed.
7. Wrap around allowed.
8. Fewest number of groups possible.

The diagram below illustrates the correspondence between the Karnaugh map and the truth table for the general case of a two variable problem.

| A | B | F | |
|---|---|---|---|
| 0 | 0 | a | $m_0$ |
| 0 | 1 | b | $m_1$ |
| 1 | 0 | c | $m_2$ |
| 1 | 1 | d | $m_3$ |

Truth Table.

| B \ A | 0 | 1 |
|---|---|---|
| 0 | a | b |
| 1 | c | d |

F.

$$F = \bar{A}B + A\bar{B} + AB$$

| A | B | F |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

Truth Table.

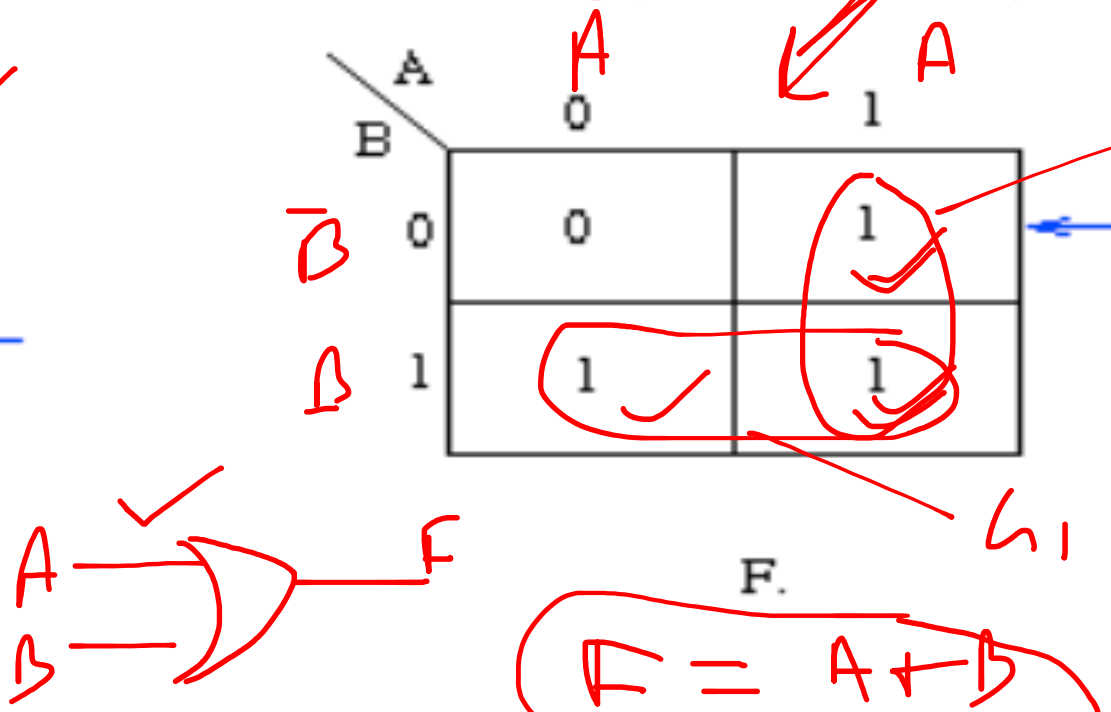
| B \ A | 0 ($\bar{A}$) | 1 (A) |
|---|---|---|
| 0 ($\bar{B}$) | 0 | 1 |
| 1 (B) | 1 | 1 |

F.

$G_1$, $G_2$

$$G_2 = A\bar{B} + AB = A$$

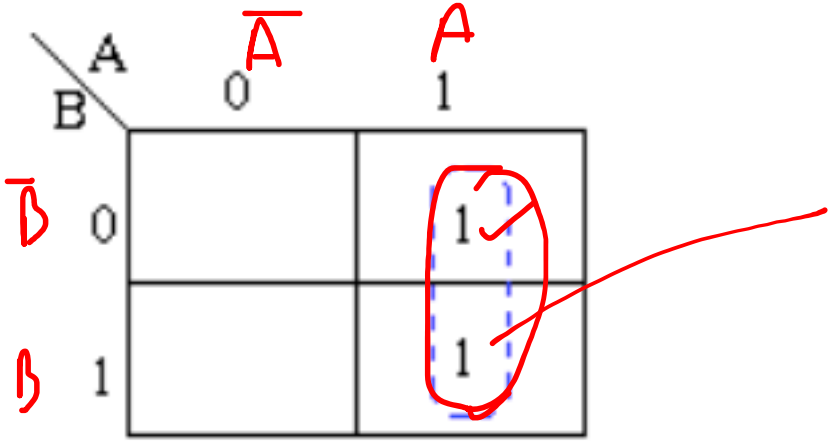
$$G_1 = \bar{A}B + AB = B$$




$$F = A + B$$

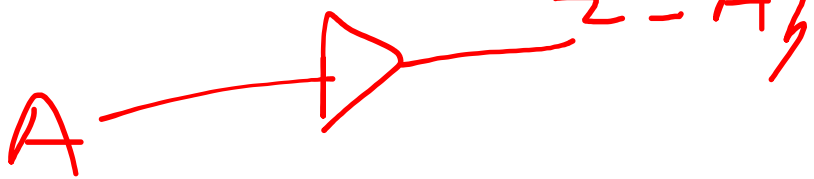
**Example 1:**

Consider the following map. The function plotted is: $Z = f(A,B) = A\overline{B} + AB$

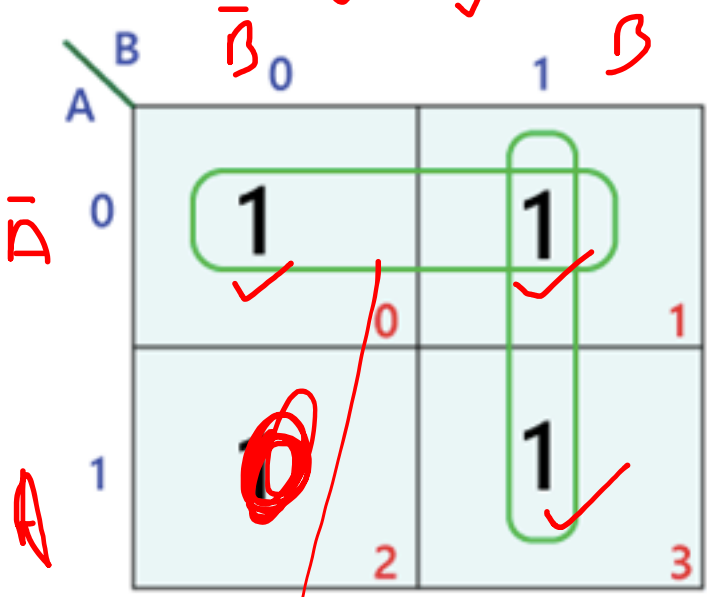



Group 1 : AB' + A B = A

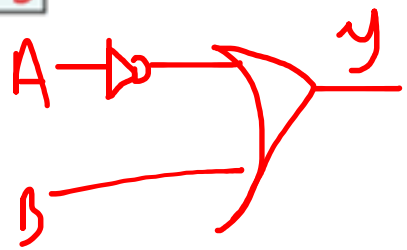
Z = A



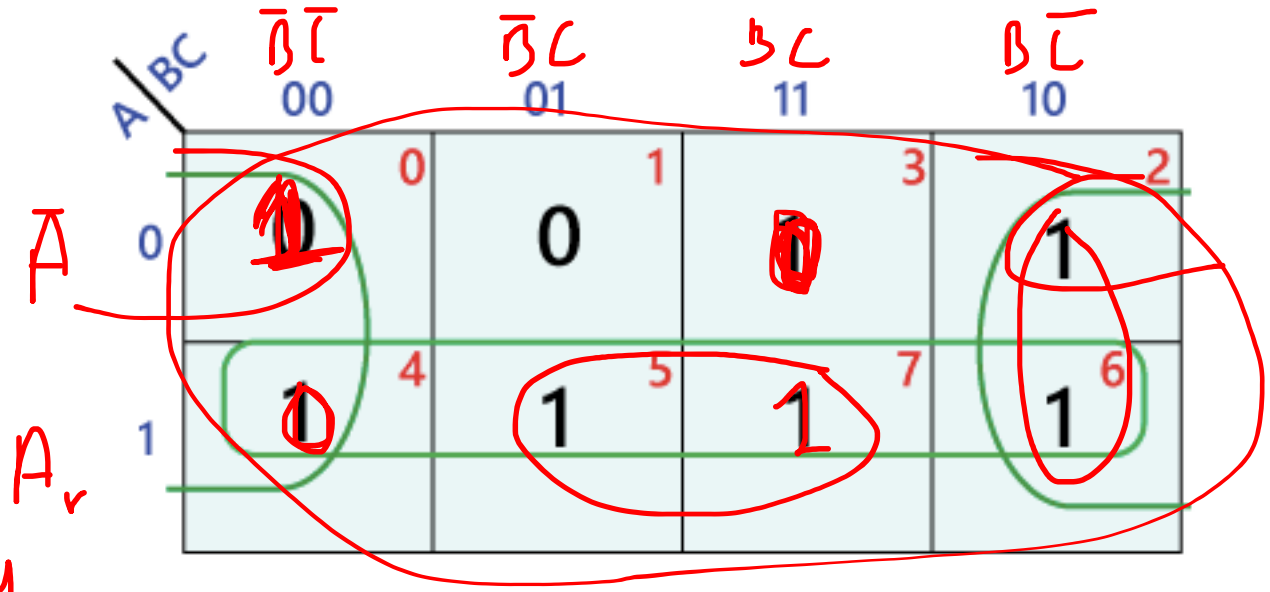
**Example 1: Y=A'B' + A'B+AB**




**Simplified expression: Y=A'+B**



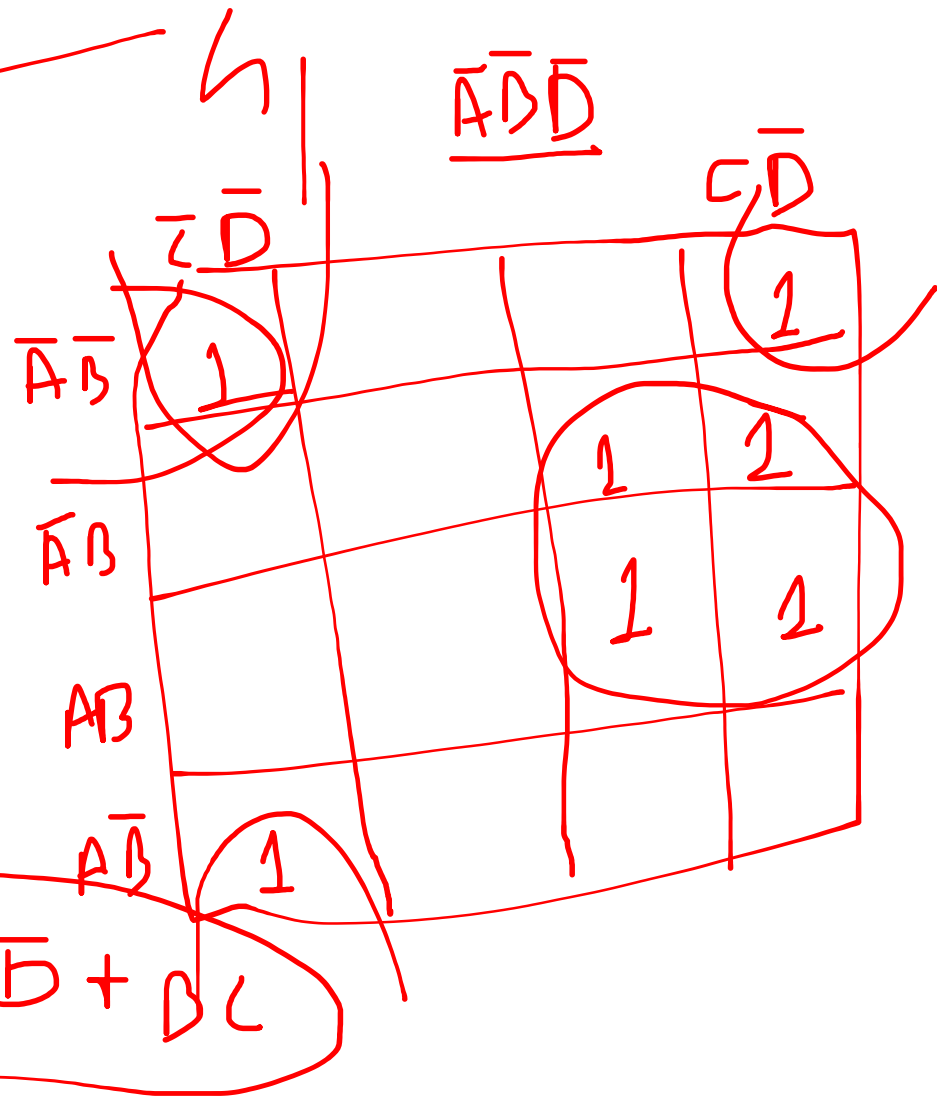

**Example 2: Y=A'B'C'+A' BC'+AB' C'+AB' C+ABC'+ABC**




**Simplified expression: Y=A+C'**

**Example 3: Y=A'B'C' D'+A' B' CD'+A' BCD'+A' BCD+AB' C' D'+ABCD'+ABCD**

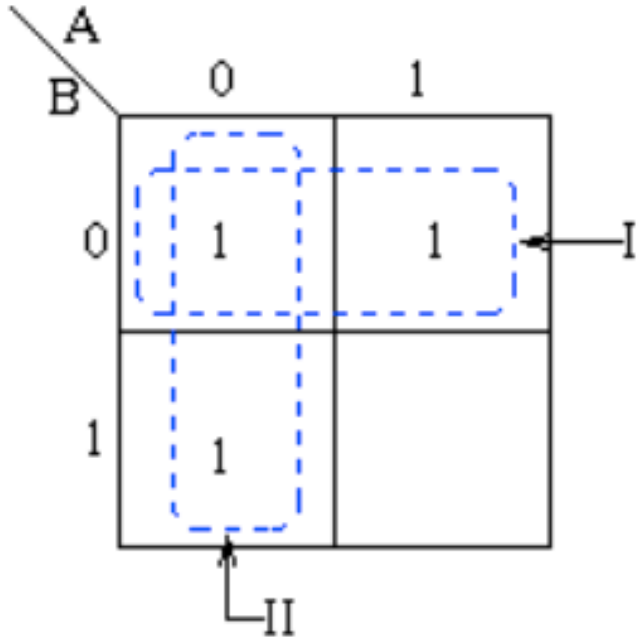
| CD / AB | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 1 (0) | 0 (1) | 0 (3) | 1 (2) |
| 01 | 0 (4) | 1 (5) | 1 (7) | 0 (6) |
| 11 | 0 (12) | 1 (13) | 1 (15) | 0 (14) |
| 10 | 1 (8) | 0 (9) | 0 (11) | 0 (10) |




**Simplified expression: Y=BD+B'D'**

$A\bar{D}\bar{D} + B\bar{C}\bar{D} + DC$

Consider the expression $Z = f(A,B) = \bar{A}\bar{B} + A\bar{B} + \bar{A}B$ plotted on the Karnaugh map:




Group 1 : A' B' + AB'. = B'

Group 2 : A' B' + A'B = A'

Therefor

Z = B' + A'

Minimise the following problems using the Karnaugh maps method.

$Z = f(A,B,C) = \bar{A}\bar{B}\,\bar{C} + \bar{A}B + AB\bar{C} + AC$

$Z = f(A,B,C) = \bar{A}B + B\bar{C} + BC + A\bar{B}\bar{C}$

Minimise the following problems using the Karnaugh maps method.

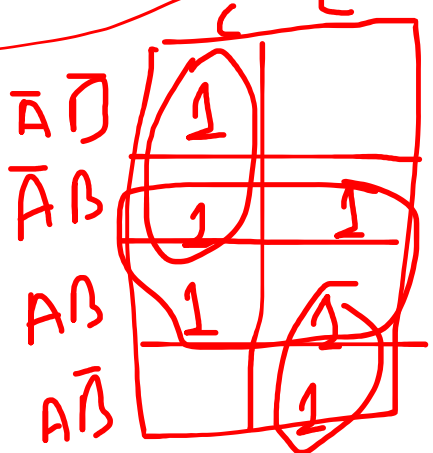
1. $Z = f(A,B,C) = \bar{A}\bar{B}\bar{C} + \bar{A}B + AB\bar{C} + AC$

2. $Z = f(A,B,C) = \bar{A}B + B\bar{C} + BC + A\bar{B}\bar{C}$

$G_2 = A\bar{B}C + ABC = AC$

$G_3 = \bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} = \bar{A}\bar{C}$

$Z = B + AC + \bar{A}\bar{C}$

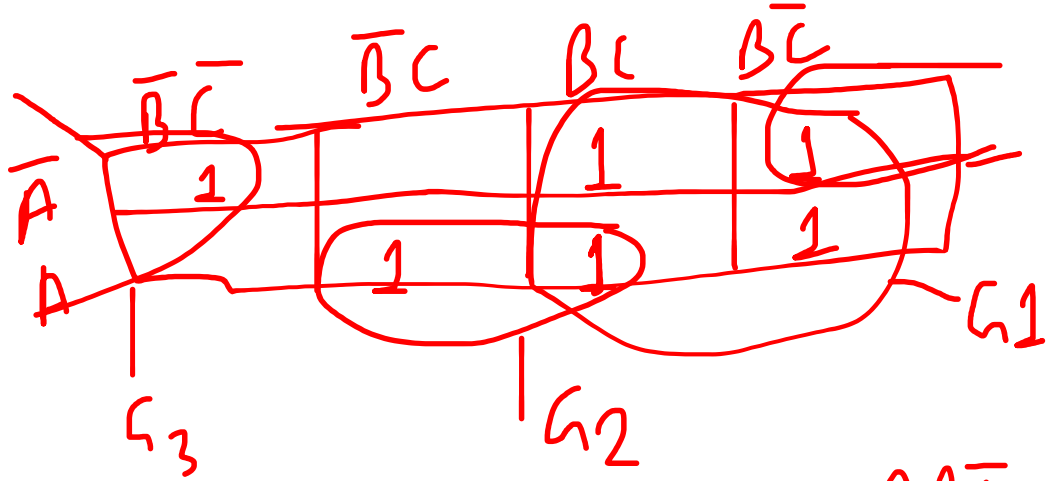



SOP

1) $\bar{A}B(C+\bar{C}) = \bar{A}BC + \bar{A}B\bar{C}$

$AC(B+\bar{B}) = ABC + A\bar{B}C$

$Z = \bar{A}\bar{B}\bar{C} + \bar{A}BC + \bar{A}B\bar{C} + AB\bar{C} + ABC + A\bar{B}C$



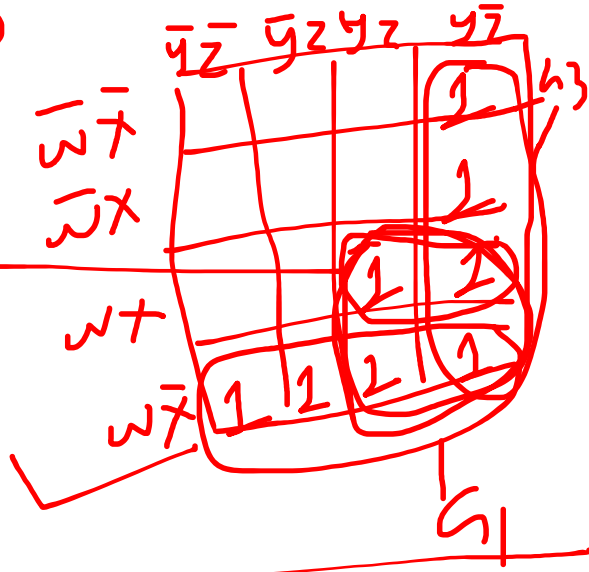

$G_1 = \bar{A}BC + ABC + \bar{A}B\bar{C} + AB\bar{C}$

$= B$

$f = \Sigma m(0,1,3,5,7)$ $G_1 = WX' + WY + YZ$ NAND

# Lets do it.....

$f = \Sigma m(0,1,3)$ $f(\quad) = \Sigma(0,1,3,4,5)$ $G_2$




1. **simplify** the following Boolean function, **f** $W, X, Y, Z$ **= WX'Y' + WY + W'YZ'** using K-map.

2. **simplify** the following Boolean function, $f(X, Y, Z) = \Sigma(0, 1, 2, 4)$ using K-map.

$f = w\bar{x}\bar{y} + wy + \bar{w}yz$

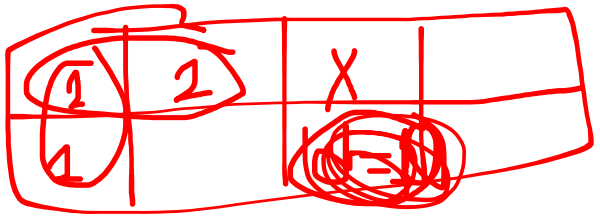
$= w\bar{x}\bar{y}(z+\bar{z}) + wy(x+\bar{x})(z+\bar{z}) + \bar{w}yz(x+\bar{x})$

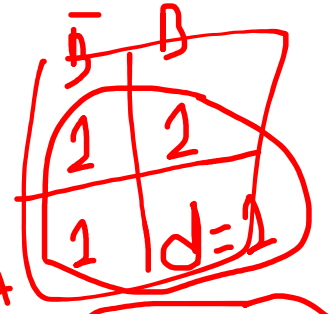
$= w\bar{x}\bar{y}z + w\bar{x}\bar{y}\bar{z} + wxy(z+\bar{z}) + wxyz + wxy\bar{z}$

$= w\bar{x}y(z+\bar{z}) + \bar{w}xyz + \bar{w}\bar{x}yz + w\bar{x}yz + w\bar{x}y\bar{z}$

$= w\bar{x}\bar{y}\bar{z} + w\bar{x}\bar{y}z + wx\bar{y}z + \bar{w}x\bar{y}z + \bar{w}\bar{x}\bar{y}z$

# Don’t Care States



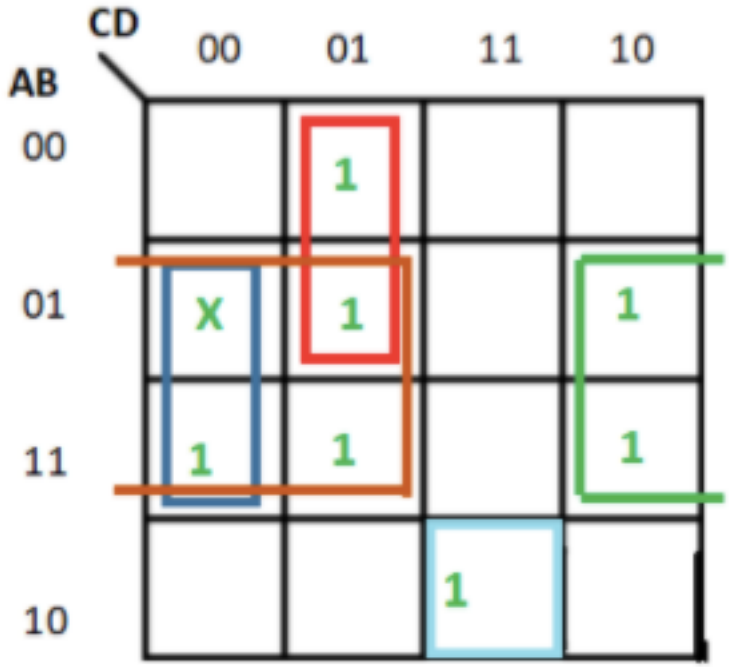


- One of the very significant and useful concept in simplifying the output expression using K-Map is the concept of “Don’t Cares”.
- The “Don’t Care” conditions allow us to replace the empty cell of a K-Map to form a grouping of the variables which is larger than that of forming groups without don’t cares.
- While forming groups of cells, we can consider a “Don’t Care” cell as 1 or 0 or we can also ignore that cell.
- Therefore, “Don’t Care” condition can help us to form a larger group of cells.
- A Don’t Care cell can be represented by a cross(X) in K-Maps representing a invalid combination.

# Example-1:

Minimise the following function in SOP minimal form using K-Maps:

f = m(1, 5, 6, 11, 12, 13, 14) + d(4)



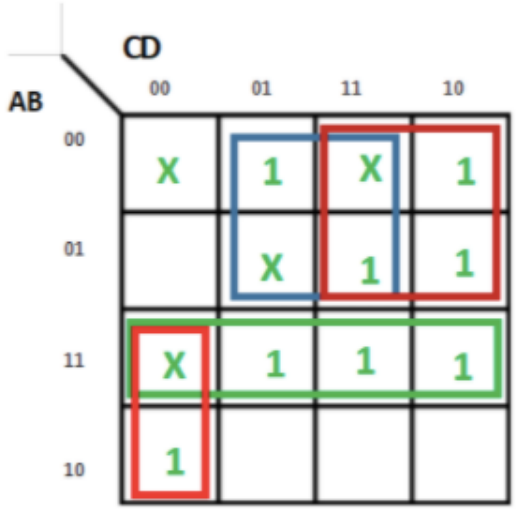
- Therefore, SOP minimal is,

f = BC' + BD' + A'C'D + AB'CD

Realize using Basic gates only:

# Minimise the following function in SOP minimal form using K-Maps:
# F(A, B, C, D) = m(1, 2, 6, 7, 8, 13, 14, 15)
# d(0, 3, 5, 12)

The SOP K-map for the given expression is:




Therefore,

f = AC'D' + A'D + A'C + AB

# Significance of “Don’t Care” Conditions

- **Simplification of the output:**
  These conditions denotes inputs that are invalid for a given digital circuit. Thus, they can used to further simplify the boolean output expression of a digital circuit.

- **Reduction in number of gates required:**
  Simplification of the expression reduces the number of gates to be used for implementing the given expression. Therefore, don’t cares make the digital circuit design more economical.

- **Reduced Power Consumption:**
  While grouping the terms long with don’t cares reduces switching of the states. This decreases the memory space that is required to represent a given digital circuit which in turn results in less power consumption.

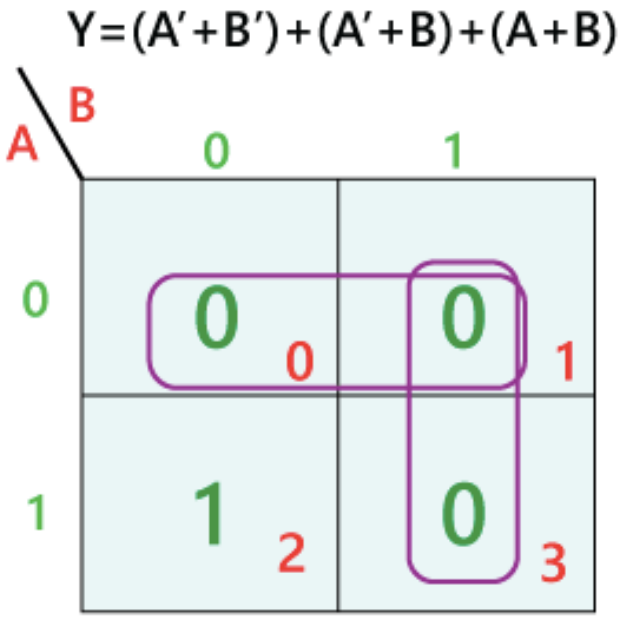
- **Represent Invalid States in Code Converters:**
  These are used in code converters. For example- In design of 4-bit BCD-to-XS-3 code converter, the input combinations 1010, 1011, 1100, 1101, 1110, and 1111 are don’t cares.

- **Prevention of Hazards in Digital Circuits:**
  Don’t cares also prevents hazards in digital systems.

# K map for Max terms

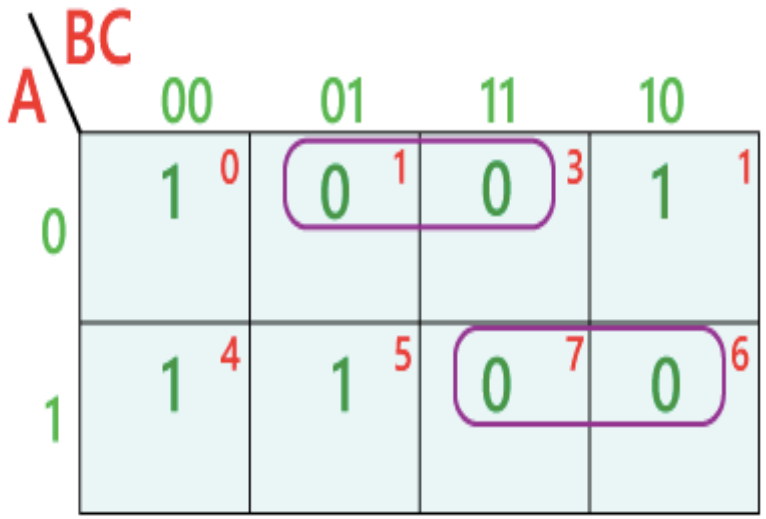
- Product of sum (POS)
- To find the simplified maxterm solution using K-map is the same as to find for the minterm solution. There are some minor changes in the maxterm solution, which are as follows:
- We will populate the K-map by entering the value of 0 to each sum-term into the K-map cell and fill the remaining cells with one's.
- We will make the groups of 'zeros' not for 'ones'.
- Now, we will define the boolean expressions for each group as sum-terms.
- At last, to find the simplified boolean expression in the POS form, we will combine the sum-terms of all individual groups.

**Example 1: Y=(A'+B')(A'+B)(A+B)**



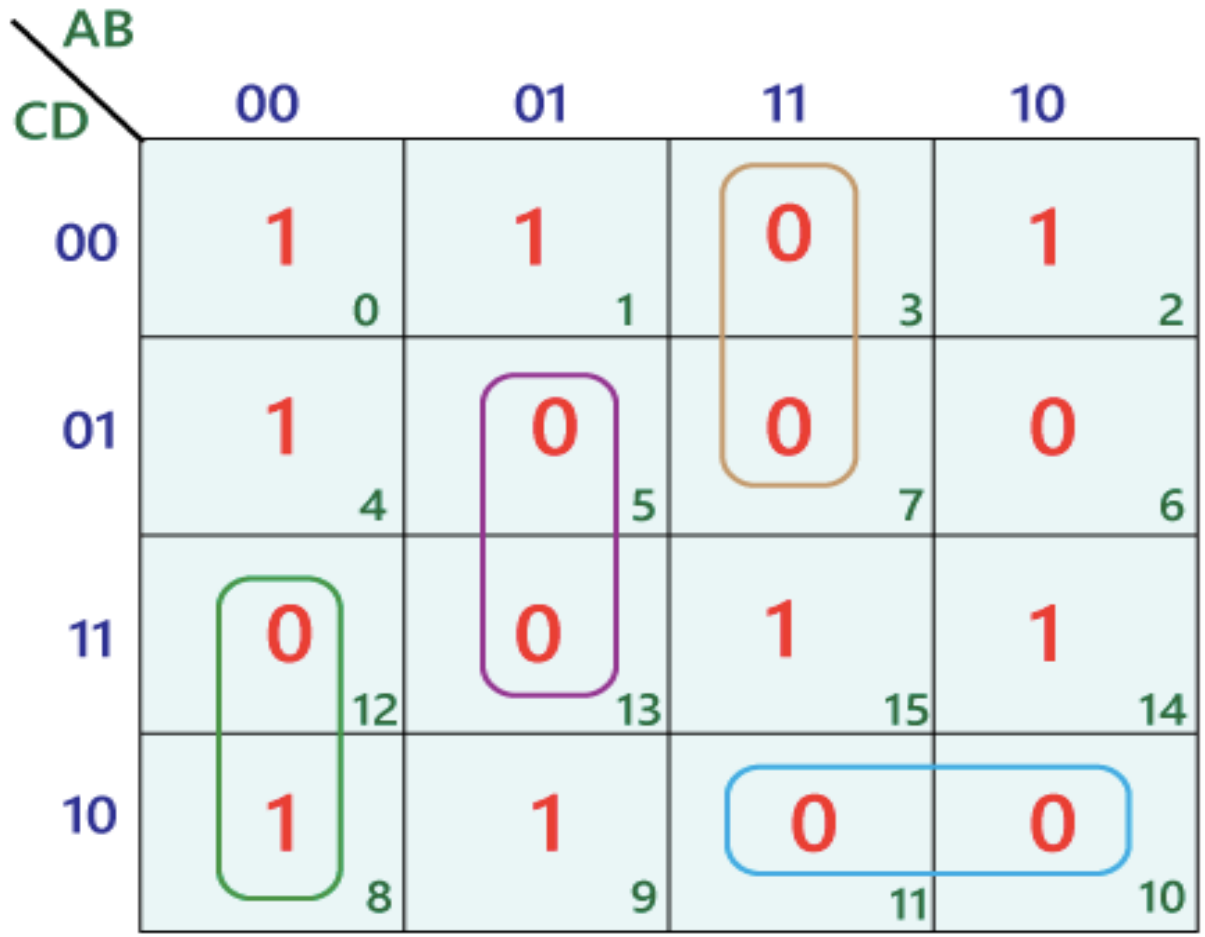

**Simplified expression: A'B**

**Example 2: Y=(A + B + C')(A + B' + C')(A' + B' + C)(A' + B' + C')**




**Simplified expression: Y=(A + C') .(A' + B')**

**Example 3: F(A,B,C,D)=π(3,5,7,8,10,11,12,13)**




**Simplified expression: Y=(A + C') .(A' + B')**

# Using K map solve this

1) F= $\sum$m( 1,3,4,6,9,11,13,15)
   d= m( 0,10,14)

2) F = $\pi$( 0,3,4,5,6,9,10,12,13)
   d= $\pi$( 1, 7,14,15)

$F = \sum m(1,3,4,6,9,11,13,15)$

$d = m(0,10,14)$

$$y = \bar{A}BC + A\bar{B}C + \bar{A}\bar{B}\bar{C}$$

$$\bar{y} = \overline{\bar{A}BC} \cdot \overline{A\bar{B}C} \cdot \overline{\bar{A}\bar{B}\bar{C}}$$

$$= (A+\bar{B}+\bar{C}) \cdot (\bar{A}+\bar{B}+\bar{C}) \cdot (A+B+C)$$

F = $\pi$( 0,3,4,5,6,9,10,12,13)

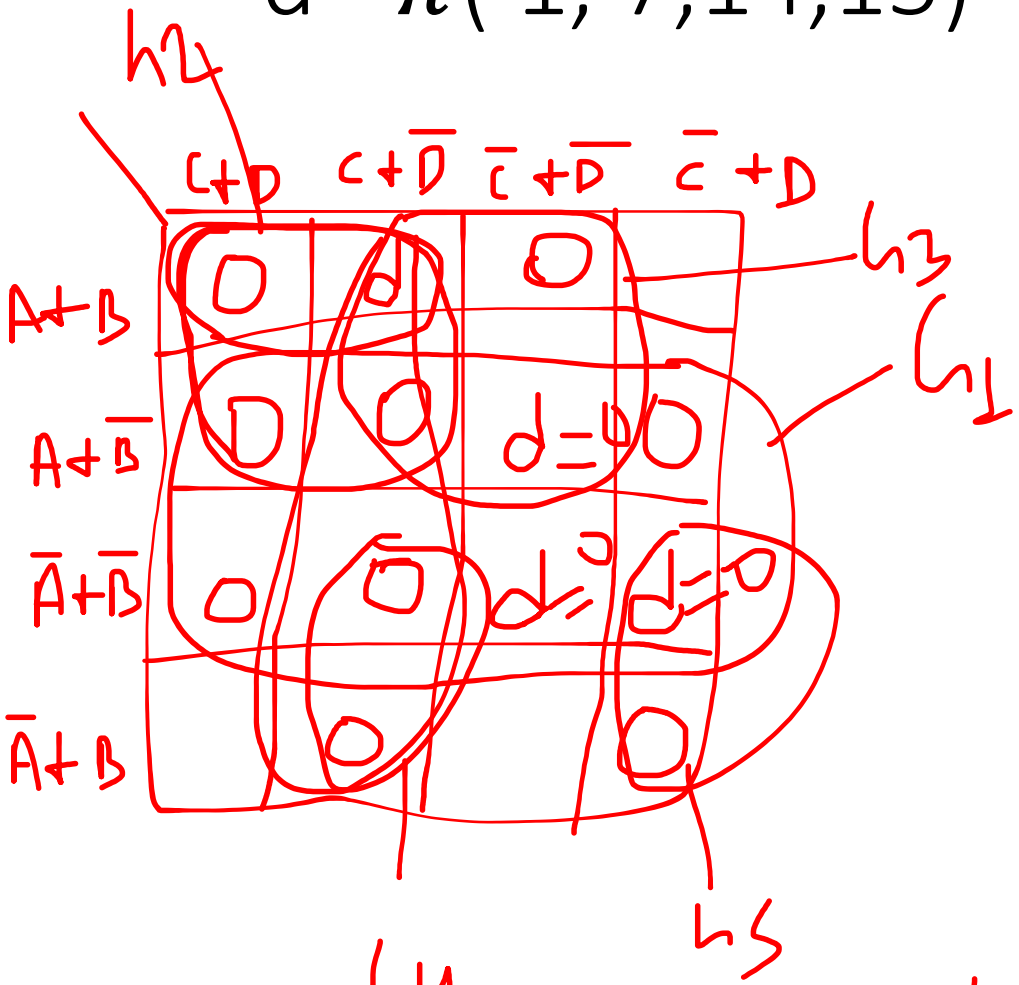
d= $\pi$( 1, 7,14,15)

| Variables | | | Min terms | Max terms |
|---|---|---|---|---|
| A | B | C | $m_i$ | $M_i$ |
| 0 | 0 | 0 | A' B' C' = m 0 | A + B + C = M 0 |
| 0 | 0 | 1 | A' B' C = m 1 | A + B + C' = M 1 |
| 0 | 1 | 0 | A' B C' = m 2 | A + B' + C = M 2 |
| 0 | 1 | 1 | A' B C = m 3 | A + B' + C' = M 3 |
| 1 | 0 | 0 | A B' C' = m 4 | A' + B + C = M 4 |
| 1 | 0 | 1 | A B' C = m 5 | A' + B + C' = M 5 |
| 1 | 1 | 0 | A B C' = m 6 | A' + B' + C = M 6 |
| 1 | 1 | 1 | A B C = m 7 | A' + B' + C' = M 7 |

POS

$F = \pi(0,3,4,5,6,9,10,12,13)$

$d = \pi(1,7,14,15)$



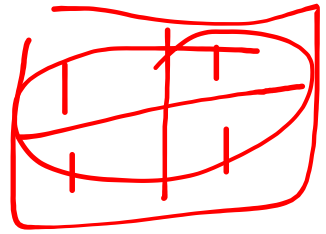

$G_1 = \bar{B}$

$G_2 = A + C$

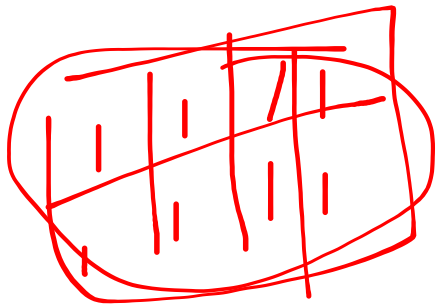
$G_3 = A + \bar{D}$

$G_4 = C + \bar{D}$

$G_5 = \bar{A} + \bar{C} + D$



F = 1



F = 1

$F = (\bar{B})\cdot(A+C)(A+\bar{D})(C+\bar{D})(\bar{A}+\bar{C}+D)$

# The End