Karnaugh Maps (K-maps) and Don’t Care Condition

Contents:

- **Introduction: Karnaugh Maps**
- **2,3, and 4 variables K-maps**
- **Examples**
- **Don’t care conditions**

Karnaugh Maps (K maps)

Introduction: Karnaugh Maps

What are Karnaugh maps?

- Karnaugh maps provide an alternative way of simplifying logic circuits.
- Instead of using Boolean algebra simplification techniques, you can transfer logic values from a Boolean statement or a truth table into a Karnaugh map.
- The arrangement of 0's and 1's within the map helps you to visualise the logic relationships between the variables and leads directly to a simplified Boolean statement.

Named for the American electrical engineer Maurice Karnaugh.

Karnaugh maps

- Karnaugh maps, or K-maps, are often used to simplify logic problems with 2, 3 or 4 variables.

Cell = 2^n ,where n is a number of variables

For the case of 2 variables, we form a map consisting of 2^2=4 cells as shown in Figure

B \ A	0	1
0	$A+B$	$\overline{A}+B$
1	$A+\overline{B}$	$\overline{A}+\overline{B}$

B \ A	0	1
0	00 (0)	10 (2)
1	01 (1)	11 (3)

B \ A	0	1
0	$\overline{A}\overline{B}$	$A\overline{B}$
1	$\overline{A}B$	AB

Maxterm Minterm

2- variables Karnaugh Maps

Karnaugh maps

- 3 variables Karnaugh map

Cell = 2^3=8

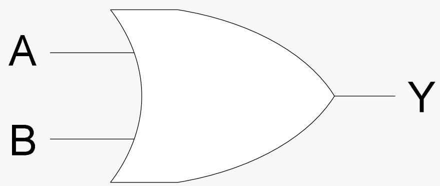
C \ AB	00	01	11	10
0	0 $\bar{A}\bar{B}\bar{C}$	2 $\bar{A}B\bar{C}$	6 $AB\bar{C}$	4 $A\bar{B}\bar{C}$
1	1 $\bar{A}\bar{B}C$	3 $\bar{A}BC$	7 ABC	5 $A\bar{B}C$

4 variables Karnaugh map

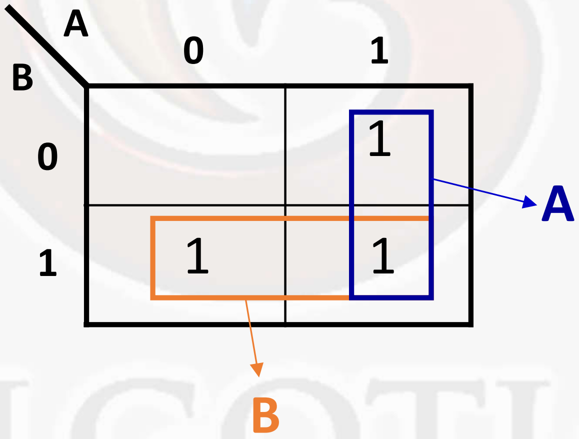
CD \ AB	00	01	11	10
00	0	4	12	8
01	1	5	13	9
11	3	7	15	11
10	2	6	14	10

Example

2-variable Karnaugh maps are trivial but can be used to introduce the methods you need to learn. The map for a 2-input OR gate looks like this:



A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1



A+B

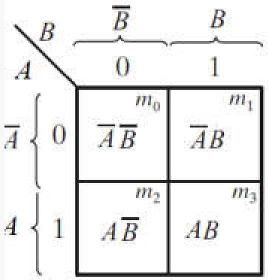
The Karnaugh map is completed by entering a '1'(or '0') in each of the appropriate cells.

Within the map, adjacent cells containing 1's (or 0's) are grouped together in twos, fours, or eights.

Single 1's, corner 1's are also grouped.

Horizontal and vertical variables which are constant throughout the group are considered for the result.
Finally all the results obtained by solving the all possible groups will be summed for the output result

Two-Variable Karnaugh Map-Problems



$Y = F(A,B) = \sum m(2,3)$

A	B	Y
0	0	0
0	1	0
1	0	1
1	1	1

	$\overline{B}$	B
$\overline{A}$	0	0
A	1	1

Y = A

***Y = F(A,B)* = Σ m(1,2,3)**

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

	$\overline{B}$	B
$\overline{A}$	0	1
A	1	1

Y = A + B

Three Variable Karnaugh Map- Problem

	AB \ C	$\overline{C}$ 0	C 1
$\overline{A}\,\overline{B}$	00	0	1
$\overline{A}B$	01	2	3
AB	11	6	7
$A\overline{B}$	10	4	5

F(A,B,C)
minterm numbers

A	B	C	Y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

$Y = F(A,B,C)$
$= \Sigma\, m(2,4,5,6,7)$

K-Map and solution

	$\overline{C}$	C
$\overline{A}\,\overline{B}$	0	0
$\overline{A}B$	1	0
AB	1	1
$A\overline{B}$	1	1

$Y = A + B.C'$

Four-Variable Karnaugh Map

	$\bar{C}\bar{D}$ 00	$\bar{C}D$ 01	CD 11	$C\bar{D}$ 10
$\bar{A}\bar{B}$ 00	0	1	3	2
$\bar{A}B$ 01	4	5	7	6
AB 11	12	13	15	14
$A\bar{B}$ 10	8	9	11	10

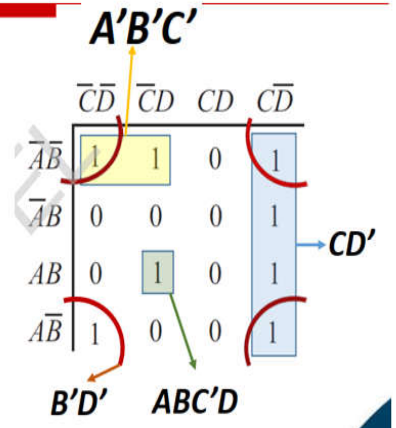
F(A,B,C,D)
minterm numbers

***Y* = *F*(*A*,*B*,*C*,*D*)**
= Σ m(0,1,2,6,8,10,13,14)

A	B	C	D	Y
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

Solution

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	1	1	0	1
$\bar{A}B$	0	0	0	1
AB	0	1	0	1
$A\bar{B}$	1	0	0	1



Y= *B'D' + CD' + A'B'C' + ABC'D*

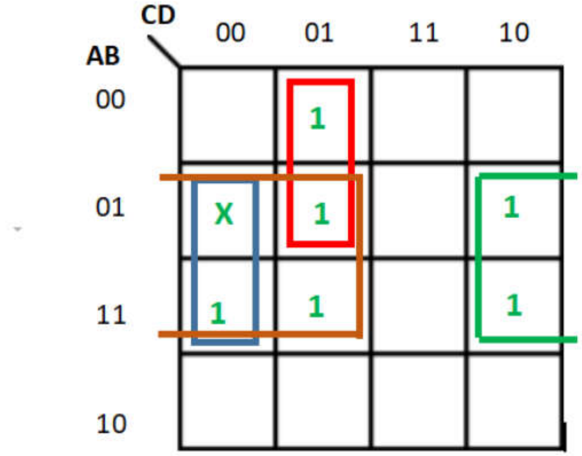
Don’t Care (X) Conditions in K-Maps

The “Don’t Care” conditions allow us to replace the empty cell of a K-Map to form a grouping of the variables. While forming groups of cells, we can consider a “Don’t Care” cell as either 1 or 0 or we can simply 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:

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

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

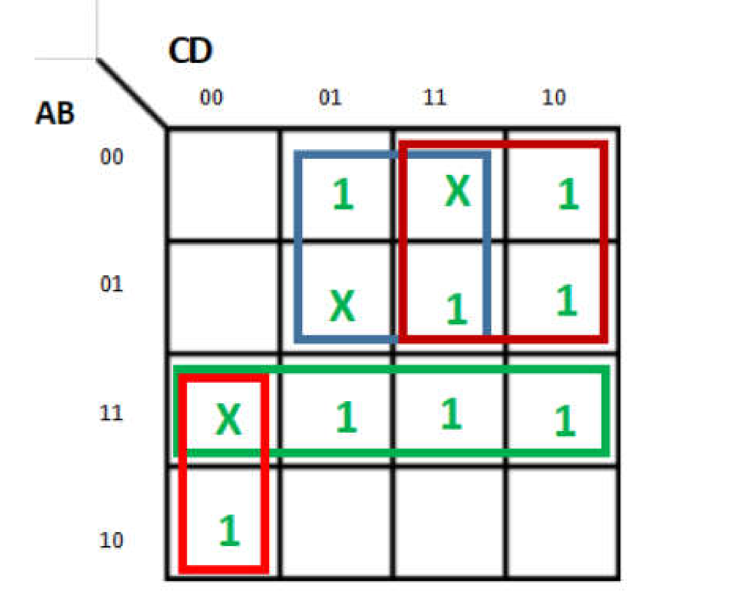


Therefore, SOP minimal is,

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

Example-2:

Minimize 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(3, 5, 12)



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

Significance of "Don't Care" Conditions:

Don't Care conditions has the following significance with respect to the digital circuit design:

Simplification:
These conditions denotes the set of inputs which never occurs for a given digital circuits. Thus, they are being used to further simplify the Boolean output expression.

Lesser number of gates:
Simplification 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 required memory space which in turn results in less power consumption.

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:
Don't cares also prevents hazards in digital systems.

References:

- Digital Principles and Applications, A.P. Malvino, D. P. Leach and Saha, 7th Ed., 2011, Tata McGraw Hill
- Digital Fundamentals, Thomas L. Floyd, 11th Ed., 2015, Pearson Education Limited
- Modern Digital Electronics, R P Jain, 4th Ed., 2010, Tata McGraw Hill