CSE1003
Digital Logic and Design

Module 2
BOOLEAN ALGEBRA L4

Dr. S. Hemamalini
Professor
School of Electrical Engineering
VIT Chennai

Module 2
BOOLEAN ALGEBRA 8 hrs

Boolean algebra

- Properties of Boolean algebra
- Boolean functions
- Canonical and Standard forms
- Logic gates - Universal gates
- Karnaugh map - Don’t care conditions
- Tabulation Method

Simplification Techniques

- Primary objective - to obtain an expression that has the minimum number of terms.
- Secondary objective - obtaining an expression with the minimum number of literals
- If there is more than one possible solution with the same number of terms, the one having the minimum number of literals is the choice.

Minimization of Boolean expressions

- The minimization will result in reduction of
  - the number of gates (resulting from less number of terms)
  - the number of inputs per gate (resulting from less number of variables per term)
- Reduced cost
- Improved efficiency
- Reduced power consumption

Sum-of-Products Boolean Expressions

- A sum-of-products expression contains the sum of different terms, with each term being either a single literal or a product of more than one literal.
- It can be obtained from the truth table directly by considering those input combinations that produce a logic '1' at the output.
- Each such input combination produces a term.
- Different terms are given by the product of the corresponding literals.
- The sum of all terms gives the expression.

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

$$Y = \overline{A}\,.\overline{B}.\overline{C} + \overline{A}.B.C + A.B.\overline{C} + A.\overline{B}.C$$

Minimum SOP

- The minimum sum of products (MSOP) of a function, f, is a SOP representation of f that contains the fewest number of product terms and fewest number of literals of any SOP representation of f.
- f= (xyz +x’yz+ xy’z+ …..) is called sum of products.
- The + is sum operator which is an OR gate.
- The product such as *xy* is an AND gate for the two inputs x and y.
- A minimum SOP expression can be implemented with fewer logic gates than a standard expression.

Minimum product of sums (MPOS)

- The minimum product of sums (MPOS) of a function, f, is a POS representation of f that contains the fewest number of sum terms and the fewest number of literals of any POS representation of f.
- The zeros are considered exactly the same as ones in the case of sum of product (SOP)

Simplification using Boolean Algebra

- **Algebraic method**
- **Karnaugh Map**
- **Tabulation Method**

- **Algebraic method**
  - Put the given expression in the SOP form repeatedly using DeMorgan's theorems and multiplication of terms.
  - Check the product terms for common variables and perform factoring wherever possible. This mostly results in elimination of one or more terms.

  - No guarantee that the reduced expression will be a minimal.

Karnaugh Map (K-Map)

- Used to simplify a Boolean expression to their minimal form.
- K-Map method is a pictorial arrangement of the truth table which allows minimal literals to express the function algebraically.
- K-Map method can be applied to functions up to 6 variables only.
- For more than 6 variables, Quine- McClusky method is used.
- K-Map is an array of cells in which each cell represents a binary value of the input variables.
- The cells are arranged in such a way that simplification of a given expression is based on properly grouping the cells.
- K-Map for an n-variable Boolean function is a rectangular diagram in which the 2^n minterms of the given function are arranged in 2^n squares in such an order that the minterms in the adjacent squares differ only by one variable, which is complemented in one square and uncomplemented in the other.

Karnaugh Map

- Is a graphical or pictorial representation of a truth table.
- A ***minterm*** of *n* variables is a product of *n* literals in which each variable appears exactly once in either true or complemented form, but not both.
- A **maxterm** of *n* variables is a sum of *n* literals in which each variable appears exactly once in either true or complemented form, but not both.

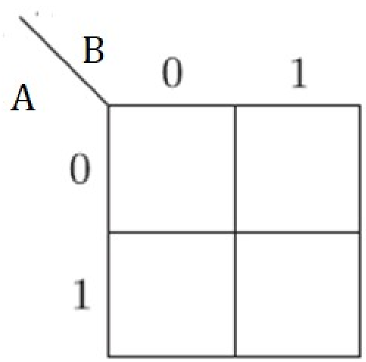
Row No.	A B C	Minterms	Maxterms
0	0 0 0	$A'B'C' = m_0$	$A + B + C = M_0$
1	0 0 1	$A'B'C = m_1$	$A + B + C' = M_1$
2	0 1 0	$A'BC' = m_2$	$A + B' + C = M_2$
3	0 1 1	$A'BC = m_3$	$A + B' + C' = M_3$
4	1 0 0	$AB'C' = m_4$	$A' + B + C = M_4$
5	1 0 1	$AB'C = m_5$	$A' + B + C' = M_5$
6	1 1 0	$ABC' = m_6$	$A' + B' + C = M_6$
7	1 1 1	$ABC = m_7$	$A' + B' + C' = M_7$

- If m_i is a minterm of *f*, then place a 1 in cell i of the K-map.
- If M_i is a maxterm of *f*, then place a 0 in cell i.

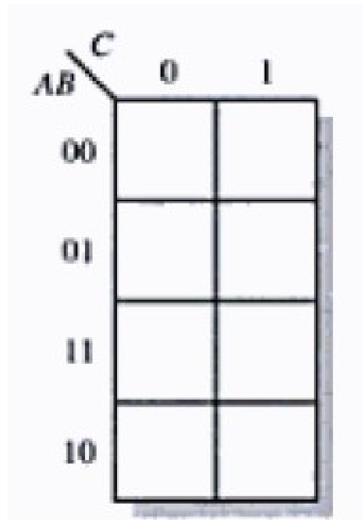
K-Map Procedure

- Prepare a blank map with 2^n cells (n- no. of variables) labelled with the variables appropriately.
- Enter a '1' in each cell (or square) of those minterms for which F=1.
- Enter a '0' in each of the remaining cells.
- Identify the possible pairs of adjacent 1's (horizontally or vertically) and group them together, so that maximum 1's can be grouped into pairs.
  - Bigger groups with eight 1's called octets
  - Bigger groups with four 1's called quads
  - Leave the isolated 1's
  - Any 1 can be reused in this process and the groups can overlap.
- Write down the terms related with each group of 1's as well as each isolated 1, by multiplying the variables and ordinate of that group or isolated 1 and ignoring the variable which appear in both the complemented and uncomplemented forms (A & A')
- Sum up all these terms to get the minimal expression of the function.

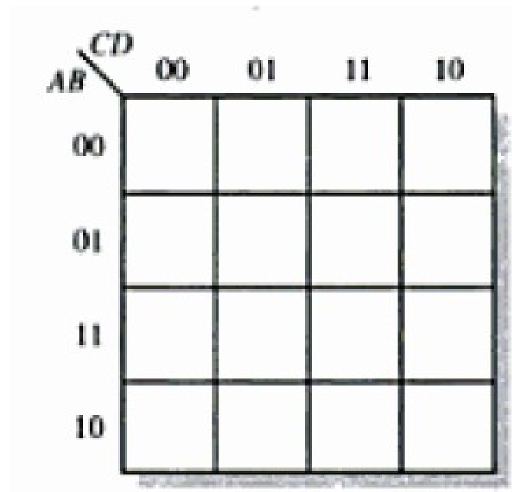
General structure of K-maps



(a) Two variable K-map

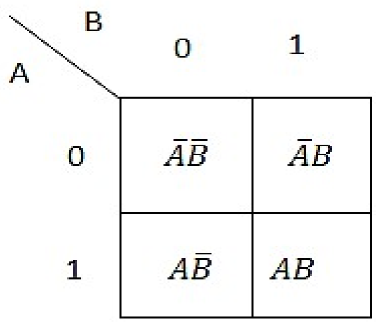


(b) Three variable K-Map

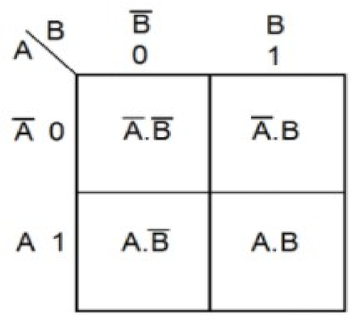


(c) Four variable K-map

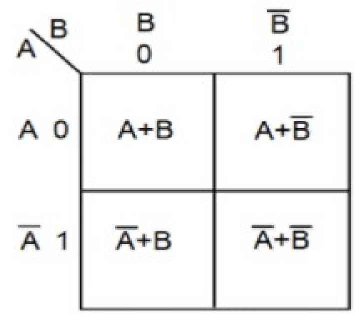
K-Map for 2 variables



SOP form

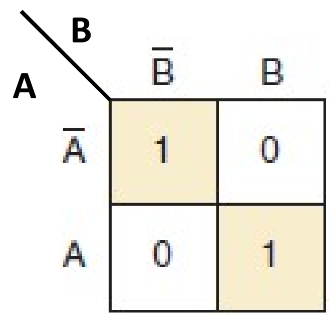


POS form



Two variable K-map f(A,B)=Σm(0,1,3)=A’B’+A’B+AB

A	B	X	
0	0	1	$\rightarrow \overline{A}\,\overline{B}$
0	1	0	
1	0	0	
1	1	1	$\rightarrow AB$



$x = \overline{A}\,\overline{B} + AB$

K-Map for 3 variables

SOP

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

A	B	C	X
0	0	0	1 $\rightarrow \overline{A}\overline{B}\overline{C}$
0	0	1	1 $\rightarrow \overline{A}\overline{B}C$
0	1	0	1 $\rightarrow \overline{A}B\overline{C}$
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1 $\rightarrow AB\overline{C}$
1	1	1	0

$$X = \overline{A}\overline{B}\overline{C} + \overline{A}\overline{B}C + \overline{A}B\overline{C} + AB\overline{C}$$

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

K-Map for 4 variables

AB \ CD	00	01	11	10
00	**A'B'C'D'**	A'B'C'D	A'B'CD	A'B'CD'
01	A'BC'D'	A'BC'D	A'BCD	A'BCD'
11	ABC'D'	ABC'D	ABCD	ABCD'
10	AB'C'D'	AB'C'D	AB'CD	AB'CD'

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

K-Map for 4 variables

A	B	C	D	X	
0	0	0	0	0	
0	0	0	1	1	$\rightarrow \overline{A}\,\overline{B}\,\overline{C}D$
0	0	1	0	0	
0	0	1	1	0	
0	1	0	0	0	
0	1	0	1	1	$\rightarrow \overline{A}B\overline{C}D$
0	1	1	0	0	
0	1	1	1	0	
1	0	0	0	0	
1	0	0	1	0	
1	0	1	0	0	
1	0	1	1	0	
1	1	0	0	0	
1	1	0	1	1	$\rightarrow AB\overline{C}D$
1	1	1	0	0	
1	1	1	1	1	$\rightarrow ABCD$

$$X = \overline{A}\,\overline{B}\,\overline{C}D + \overline{A}B\overline{C}D + AB\overline{C}D + ABCD$$

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

Simplification Guidelines for K-maps

- Always combine as many cells in a group as possible. This will result in the fewest number of literals in the term that represents the group.
- Make as few groupings as possible to cover all minterms. This will result in the fewest product terms.
- Always begin with the largest group, which means if you can find eight members group is better than two four groups and one four group is better than pair of two-group.

Adjacency

- Cell adjacency
  - Only one variable changes from complemented to uncomplemented form or vice-versa moving from one square to the other in the rows or columns.
- Wrap around adjacency
  - The cells in the top row are adjacent to the corresponding cells in the bottom row and the cells in the outer left column are adjacent to the corresponding cells in the outer right column.

K-map method for simplifying a Boolean expression – step by step procedure

1. Construct the K-map as discussed. Enter 1 in those cells corresponding to the minterms for which function value is 1. Place 0's in other cells.

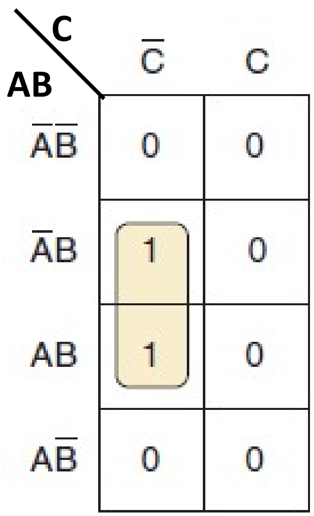
2. Form the groups of possible 1s as pair, quad and octet. There can be overlapping of groups if they include common cells. While doing this make sure that there are minimum number of groups.

3. Encircle the cells which contain 1s and are not adjacent to any other cell. These are known as isolated minterms and they appear in the expression in same form.

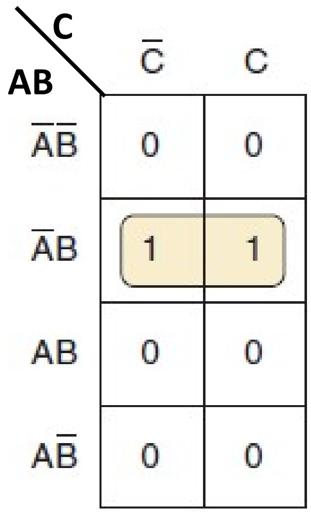
4. Avoid any redundant group.

5. Write the Boolean term for each group and obtain the minimized expression by summing product terms of all the groups.

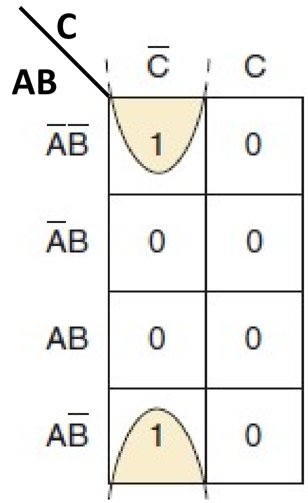
Grouping Cells (Pair)



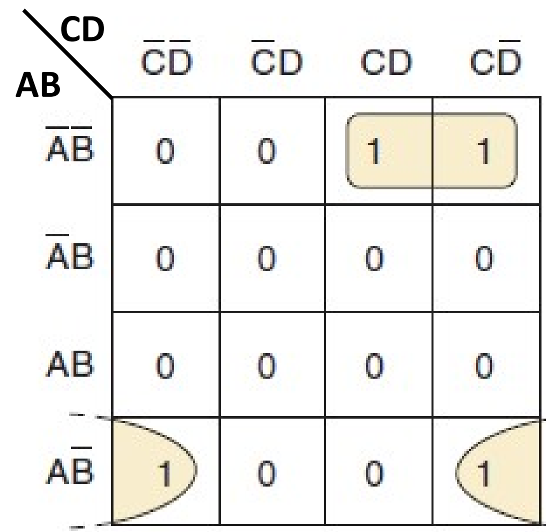
$$X = \bar{A}B\bar{C} + AB\bar{C} = B\bar{C}$$



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



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



$$X = \bar{A}\bar{B}CD + \bar{A}\bar{B}C\bar{D} + A\bar{B}\bar{C}\bar{D} + A\bar{B}C\bar{D} = \bar{A}\bar{B}C + A\bar{B}\bar{D}$$

Looping a pair of adjacent 1s in a K map eliminates the variable that appears in complemented and uncomplemented form.

Groups of Four (Quads)

A K map may contain a group of four 1s that are adjacent to each other. This group is called a *quad.*

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

$X = C$

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

$X = AB$

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

$X = BD$

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

$X = A\bar{D}$

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

$X = \bar{B}\bar{D}$

Looping a quad of adjacent 1s eliminates the two variables that appear in both complemented and uncomplemented form.

Groups of Eight (Octets)

A group of eight 1s that are adjacent to one another is called an *octet.*

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

$X = B$

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

$X = \overline{C}$

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

$X = \overline{B}$

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

$X = \overline{D}$

Looping an octet of adjacent 1s eliminates the three variables that appear in both complemented and uncomplemented form.

When a variable appears in both complemented and uncomplemented form within a loop, that variable is eliminated from the expression.
Variables that are the same for all squares of the loop must appear in the final expression.