UNIT - II

GATE - LEVEL MINIMIZATION

* During the process of simplification of Boolean expression we have to predict each successive step.
* We can never be absolutely true that an expression simplified by boolean algebra alone is the simplest possible expression

⇒ Map method:-

* This gives raise to adoption of new technique called, Map method, this gives a systematic approach for simplifying a boolean expression.

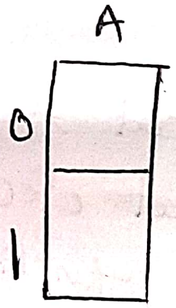
→ The karnaugh map (K-map) is a visual method used to simplify the algebraic expressions in Boolean functions without having to resort to complex theorems (or) equation manipulations.

→ The karnaugh Map (or) K-map is graphical representation of Boolean function. It is used in digital electronics and computer science to simplify Boolean function and minimize the no. of gates required to implement a logic circuit.

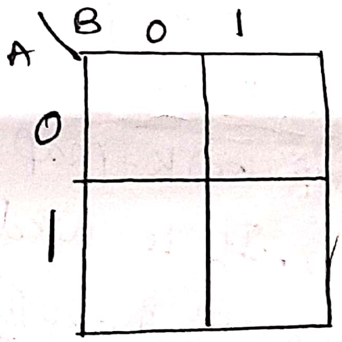
* The Map method, first proposed by veitch and modified by karnaugh, hence it is known as Veitch diagram (or) karnaugh Map.

* One-variable, Two variable, three variable and four-variable maps :-

* The basis of this method is a graphical ~~chart~~ chart known as Karnaugh map (simply k-map)
* It contains boxes called cells.
* Each of the cell represents one of the 2^n possible products that can be formed from 'n' variables. Thus, a 2n-variable map contains $2^2 = 4$ cells. 3-variable map contains $2^3 = 8$ cells and so on.
* Product terms are assigned to the cells of a Karnaugh map by labelling each row and each column of the map with a variable, with it's complement, (or) with a combination of variables and complement.

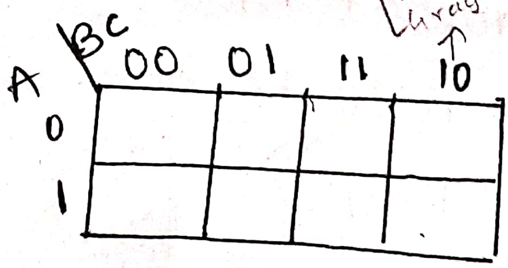


1 Variable map
(2-cells)



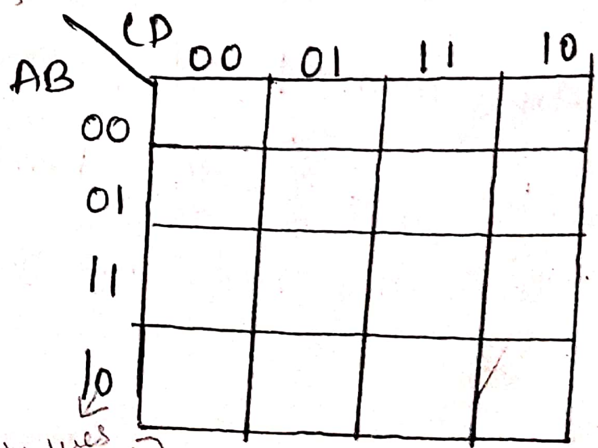
2-Variable map
(4 cells)

[values in graycode sequence]



3-Variable
(8 cells)

[Values in Graycode]



4-Variable map
(16-cells)

A = 0

BC \ DE	00	01	11	10
00				
01				
11				
10				

A = 1

BC \ DE	00	01	11	10
00				
01				
11				
10				

5 Variable map
(32 cells)

* The product term corresponding to a given cell is then the product of all variables in the row and column where the cell is located.

* When we move from one cell to the next cell along any row (or) from one cell to the next along any column, one and only one variable in the product term changes.

A	
0	$\bar{A}$ 0
1	A 1

(a) 1- Variable map

A \ B	0	1
0	$\bar{A}\bar{B}$ 0	$\bar{A}B$ 1
1	$A\bar{B}$ 2	AB 3

(b) 2-Variable map

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

(c) 3-Variable map

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

(d) 4-Variable K-map

$A=0$

$BC \backslash DE$	00	01	11	10
00	$\bar B\bar C\bar D\bar E$ 0	$\bar B\bar C\bar D E$ 1	$\bar B\bar C DE$ 3	$\bar B\bar C D\bar E$ 2
01	$\bar B C\bar D\bar E$ 4	$\bar B C\bar D E$ 5	$\bar B CDE$ 7	$\bar B CD\bar E$ 6
11	$BC\bar D\bar E$ 12	$BC\bar D E$ 13	$BCDE$ 15	$BCD\bar E$ 14
10	$B\bar C\bar D\bar E$ 8	$B\bar C\bar D E$ 9	$B\bar C DE$ 11	$B\bar C D\bar E$ 10

$A=1$

$BC \backslash DE$	00	01	11	10
00	$\bar B\bar C\bar D\bar E$ 16	$\bar B\bar C\bar D E$ 17	$\bar B\bar C DE$ 19	$\bar B\bar C D\bar E$ 18
01	$\bar B C\bar D\bar E$ 20	$\bar B C\bar D E$ 21	$\bar B CDE$ 23	$\bar B CD\bar E$ 22
11	$BC\bar D\bar E$ 28	$BC\bar D E$ 29	$BCDE$ 31	$BCD\bar E$ 30
10	$B\bar C\bar D\bar E$ 24	$B\bar C\bar D E$ 25	$B\bar C DE$ 27	$B\bar C D\bar E$ 26

* for example in 2-variable map, the only change occurs in moving along the bottom row from $A\bar B$ to AB is the change from $\bar B$ to B similarly, the only change that occurs in moving down the right column from $\bar A B$ to AB is the change from $\bar A$ to A. Irrespective of variables the labels along each row and column must conform to the single-change rule

* The Gray code has same properties, where only one variable change when we proceed to next number (or) previous number hence gray code is used to label the rows and columns of K-map.

→ In SOP the mapping variable is '1'

→ In POS the mapping variable is '0'

→ In Simplification of Boolean expression is two types

① Mapping ② grouping

① Mapping for K-map.

Inner – Binary

outer – gray code

* for 2-Variable

→

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

		SOP	POS
0 →	00	$\bar{A}\bar{B}$	$(A+B)$
1 →	01	$\bar{A}B$	$(A+\bar{B})$
2 →	10	$A\bar{B}$	$(\bar{A}+B)$
3 →	11	AB	$(\bar{A}+\bar{B})$

* for 3-Variable

A \ BC	00	01	11	10
0	000 0	001 1	011 3	010 2
1	100 4	101 5	111 7	110 6

	4 2 1	SOP	POS
0 →	0 0 0	$\bar{A}\bar{B}\bar{C}$	$(A+B+C)$
1 →	0 0 1	$\bar{A}\bar{B}C$	$(A+B+\bar{C})$
2 →	0 1 0	$\bar{A}B\bar{C}$	$(A+\bar{B}+C)$
3 →	0 1 1	$\bar{A}BC$	$(A+\bar{B}+\bar{C})$
4 →	1 0 0	$A\bar{B}\bar{C}$	$(\bar{A}+B+C)$
5 →	1 0 1	$A\bar{B}C$	$(\bar{A}+B+\bar{C})$
6 →	1 1 0	$AB\bar{C}$	$(\bar{A}+\bar{B}+C)$
7 →	1 1 1	ABC	$(\bar{A}+\bar{B}+\bar{C})$

* for 4-Variable

AB \ CD	00	01	11	10
00	0000 0	0001 1	0011 3	0010 2
01	0100 4	0101 5	0111 7	0110 6
11	1100 12	1101 13	1111 15	1110 14
10	1000 8	1001 9	1011 11	1010 10

	8 4 2 1	SOP	POS
0 →	0 0 0 0 →	$\bar{A}\bar{B}\bar{C}\bar{D}$	— $(A+B+C+D)$
1 →	0 0 0 1 →	$\bar{A}\bar{B}\bar{C}D$	— $(A+B+C+\bar{D})$
2 →	0 0 1 0 →	$\bar{A}\bar{B}C\bar{D}$	— $(A+B+\bar{C}+D)$
3 →	0 0 1 1 →	$\bar{A}\bar{B}CD$	— $(A+B+\bar{C}+\bar{D})$
4 →	0 1 0 0 →	$\bar{A}B\bar{C}\bar{D}$	— $(A+\bar{B}+C+D)$
5 →	0 1 0 1 →	$\bar{A}B\bar{C}D$	— $(A+\bar{B}+C+\bar{D})$
6 →	0 1 1 0 →	$\bar{A}BC\bar{D}$	— $(A+\bar{B}+\bar{C}+D)$
7 →	0 1 1 1 →	$\bar{A}BCD$	— $(A+\bar{B}+\bar{C}+\bar{D})$
8 →	1 0 0 0 →	$A\bar{B}\bar{C}\bar{D}$	— $(\bar{A}+B+C+D)$
9 →	1 0 0 1 →	$A\bar{B}\bar{C}D$	— $(\bar{A}+B+C+\bar{D})$
10 →	1 0 1 0 →	$A\bar{B}C\bar{D}$	— $(\bar{A}+B+\bar{C}+D)$
11 →	1 0 1 1 →	$A\bar{B}CD$	— $(\bar{A}+B+\bar{C}+\bar{D})$
12 →	1 1 0 0 →	$AB\bar{C}\bar{D}$	— $(\bar{A}+\bar{B}+C+D)$
13 →	1 1 0 1 →	$AB\bar{C}D$	— $(\bar{A}+\bar{B}+C+\bar{D})$
14 →	1 1 1 0 →	$ABC\bar{D}$	— $(\bar{A}+\bar{B}+\bar{C}+D)$
15 →	1 1 1 1 →	$ABCD$	

⇒ In case of SOP form:

A	
0	m_0
1	m_1

A \ B	0	1
0	m_0	m_1
1	m_2	m_3

(a) 1 Variable map (b) 2-variable map

A \ BC	00	01	11	10
0	m_0	m_1	m_3	m_2
1	m_4	m_5	m_7	m_6

(c) 3-Variable map

Gray code sequence

AB \ CD	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_{16}	m_{15}
10	m_8	m_9	m_{11}	m_{10}

(d) 4-Variable map

* Instead of writing actual product terms, corresponding short hand minterm notations are written in the cell & row and columns are marked with gray code instead of variables.

⇒ In case of POS:-

A	
0	M_0
1	M_1

(a) 1 Variable map

A \ B	0	1
0	M_0	M_1
1	M_2	M_3

A \ BC	00	01	11	10
0	M_0	M_1	M_3	M_2
1	M_4	M_5	M_7	M_6

AB \ CD	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}

Scanned with OKEN Scanner

Plotting a Karnaugh map:

logic function can be represented in various forms such as truth table, SOP boolean expression, POS boolean expression. let us see, how to plot the given boolean function which is in truth table form into K-map.

Representation of truth table on karnaugh map:

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

A \ B	0	1
0	0	1
1	1	0

} cell address '01'

(a) Representation of 2-variable truth table on K-map.

In the given truth table input variables are the cell's co-ordinates, the O/P variables is the cell contents.

(b) Representation of 3-variable truth table on K-map

	A (4)	B (2)	C (1)	Y
0	0	0	0	0
1	0	0	1	1
2	0	1	0	1
3	0	1	1	0
4	1	0	0	1
5	1	0	1	0
6	1	1	0	0
7	1	1	1	1

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

Representing standard SOP on k-map

→ A Boolean expression in the sum of products form can be plotted on the Karnaugh map by placing a '1' in each cell corresponding to a term (minterm) in the sum of products expression. Remaining cells are filled with zeros.

eg: Plot the given boolean expression

$$Y = AB\bar{C} + ABC + \bar{A}\bar{B}C \text{ on k-map}$$

Sol: The expression has 3 variables, hence it can be plotted using 3-variable k-map.

A \ BC	00	01	11	10
0				
1				

A \ BC	00	01	11	10
0				
1				

$\bar{A}\bar{B}C$

A \ BC	00	01	11	10
0	0	1	0	0
1	0	0	1	1

→ $AB\bar{C}$

ABC

$F = A\bar{B} + \bar{A}B$: map the variable in to k-map

(5)

Representing Standard POS on - K-map :

* A Boolean expression in the product of sums can be plotted on the K-map by placing a '0' in each cell corresponding to a term (maxterm) in the expression. Remaining cells are filled with ones.

) Plot the boolean expression $Y = (A+\bar{B}+C)(A+\bar{B}+\bar{C})(\bar{A}+\bar{B}+C)(A+B+\bar{C})$ on K-map

Sol) The expression has 3-variables and hence it can be ploted using 3-variable map

A \ BC	00	01	11	10
0	$A+B+C$	$A+B+\bar{C}$	$A+\bar{B}+\bar{C}$	
1				

A \ BC	00	01	11	10
0	0	0^{1}	0^{3}	0^{2}
1	4	5	7	0^{6}

$A+\bar{B}+C = M_2$, $A+\bar{B}+\bar{C} = M_3$, $\bar{A}+\bar{B}+C = M_6$

$A+B+\bar{C} = M_1$

① $f = A\bar{B} + \bar{A}B$ map the variable in to k-map

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

10 01

$\Sigma m(1,2)$

A \ B	0	1
0	0	1 1
1	1 2	3

② $Y = A + BC$ map variable in to k-map

$F = A + BC$

By using canonical form

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

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

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

111 110 101 100 111 011

m_7 m_6 m_5 m_4 m_7 m_3

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

$\Sigma m(3,4,5,6,7)$

③ $F = (A+B+C)(\bar{A}+\bar{B}+C)(\bar{A}+\bar{B}+\bar{C})$ map variable into k-map

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

0 0 0   1 1 0   1 1 1

M_0 M_6 M_7

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

Grouping : either horizontal (or) vertical ⑥

They are mainly 4 types of grouping

1. Octet :- grouping of 8 adjacent cells
2. Quard :- " " 4 " "
3. pair :- " " 2 " "
4. isolate :- group of single cells

-> Large group minimization of variables is small.
In digital electronics, grouping is a process to simplify the Boolean expression.

1. Octet grouping :-

* Horizontal octet grouping
* vertical octet grouping
* horizontal edge octet grouping
  (1st Row is adjacent to 4th row)
  (1st coloumn is adjacent to 4th coloumn)
* Vertical edge octet grouping

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

A B C D

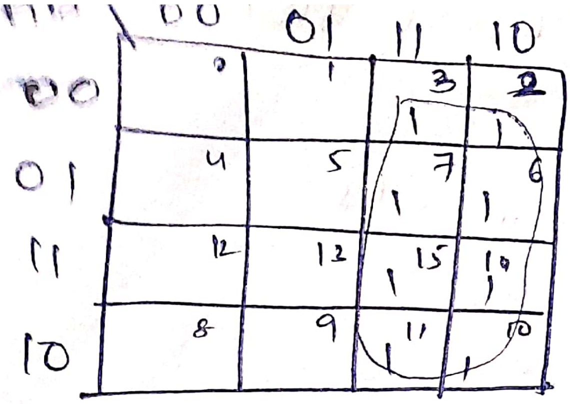
0 1 0 0

1 1 0 1

1 ✗

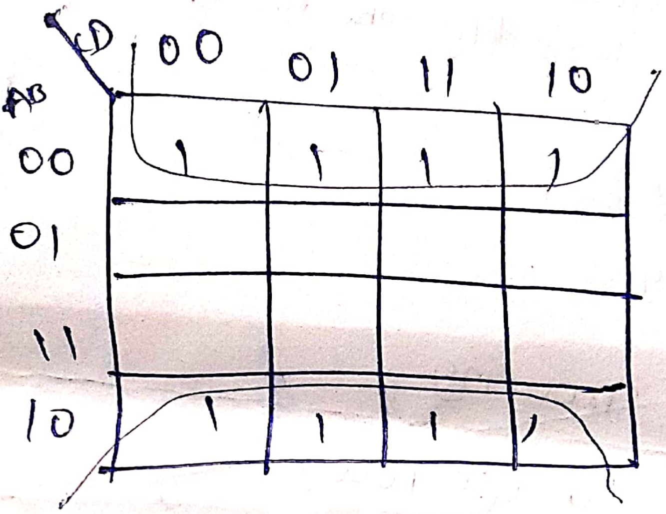
✗ 0 ✓

B



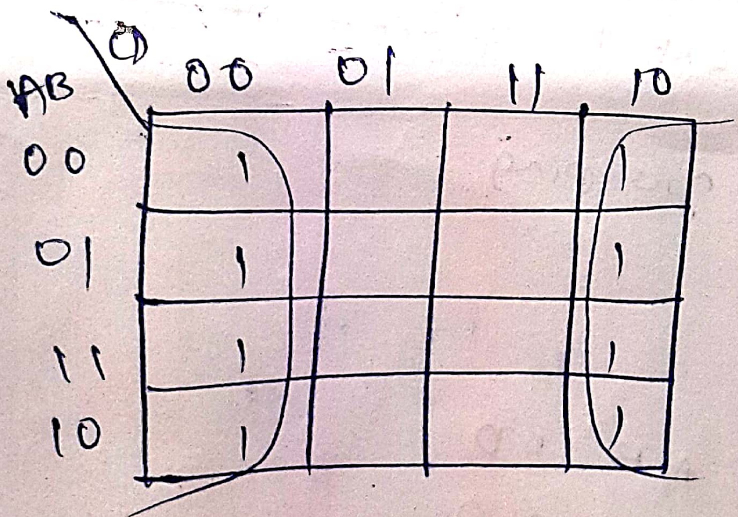
A B CD
0 0 1 1
0 1 1 0
1 1
1 0

C



A B CD
0 0 0 0
1 0 0 1
1 1
1 0

$\bar{B}$



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

$\bar{D}$

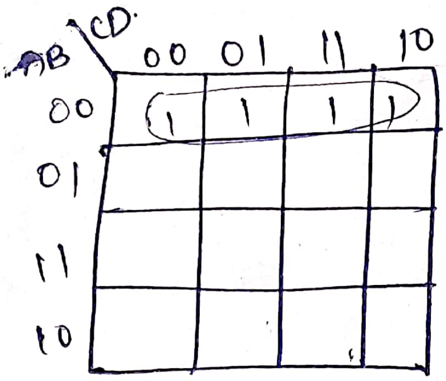
(2) Quad grouping :

→ Horizontal

→ Vertical

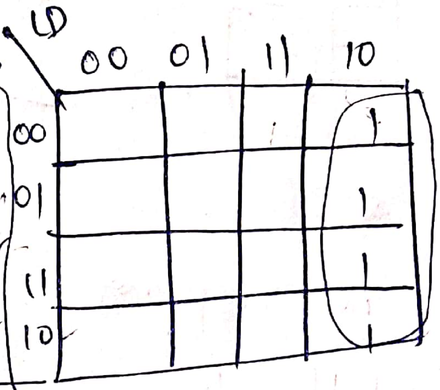
→ Square type

→ ~~xxxxxx~~ edge - corner



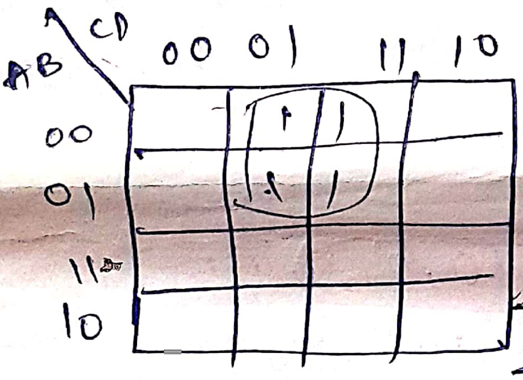
A B C D
0 0 0 0
0 1
1 1
1 0

$\bar{A}\bar{B}$



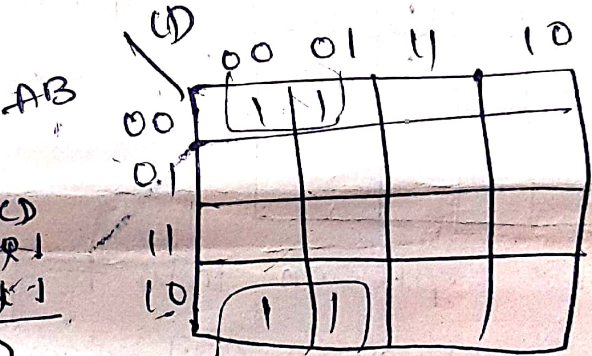
A B CD
0 0 10
0 1
1 1
1 0

$C\bar{D}$



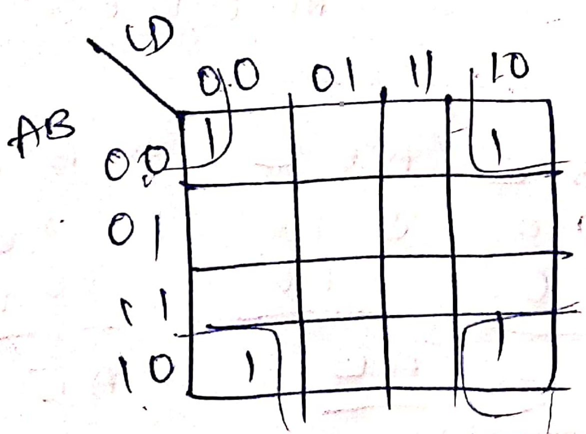
A B C D
0 0 0 1
0 1 0 1

$\bar{A}D$



A B CD
0 0 0 0
1 0 0 1

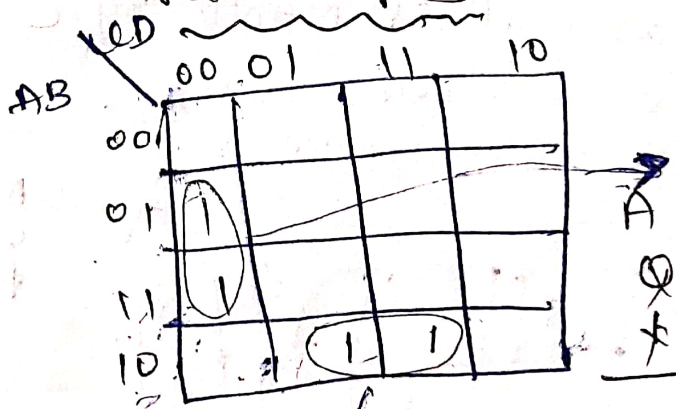
$\bar{B}\bar{C}$



A B CD
0 0 0 0
1 0 1 0

$\bar{B}\bar{D}$

Pair Grouping



A B CD
0 1 0 0
1 1

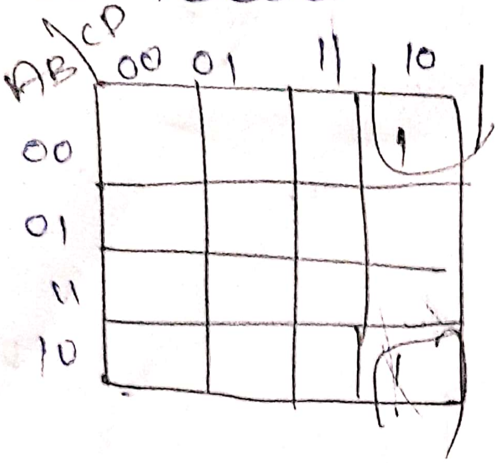
$B\bar{C}\bar{D}$

A B CD
1 0 0 1
1 1

$A\bar{B}D$

(7)

* Pair grouping *

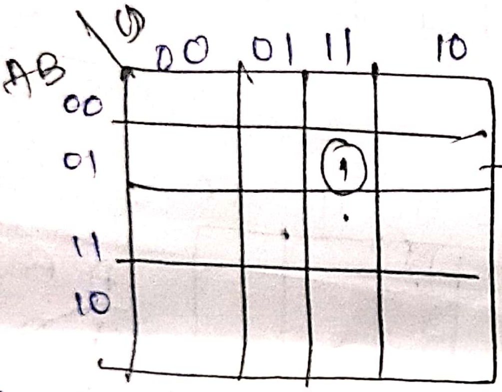


A	B	C	D
~~1~~	0	1	0
~~1~~	0		

$\bar{B} C \bar{D}$

* Isolate *

There is no grouping



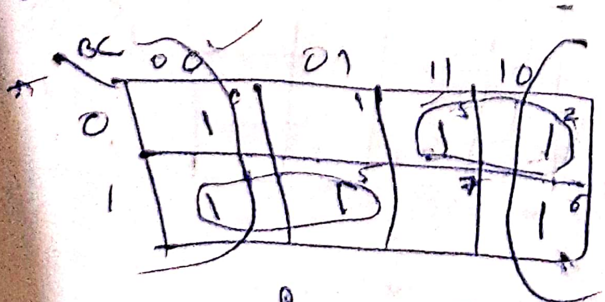
$\rightarrow \bar{A} B C D$

(3) Reduce the expression into $F = \Sigma m(0,2,3,4,5,6)$ using mapping and Implement it AOI logic as well as NAND logic.

AOI (AND, OR, Invert)

→ The expression in SOP.

→ so mapping variable is '1'.



A	B	C
~~0~~	~~0~~	0
~~1~~	~~1~~	0

$\bar{C}$

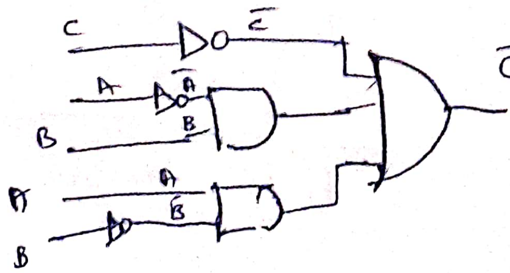
A	B	C
0	1	~~1~~
	1	~~0~~

$\bar{A} B$

A	B	C
1	0	~~0~~
	0	~~1~~

$A \bar{B}$

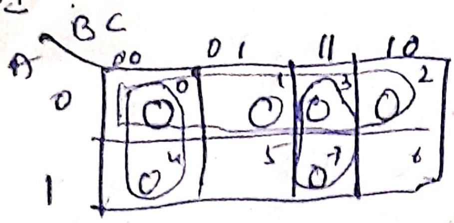
$f = \bar{C} + \bar{A} B + A \bar{B}$



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

$f = \Pi M(0,1,2,3,4,7)$ Simplify the given expression diagram ⑧.

AOI



A	B	C
0	0	0
1		

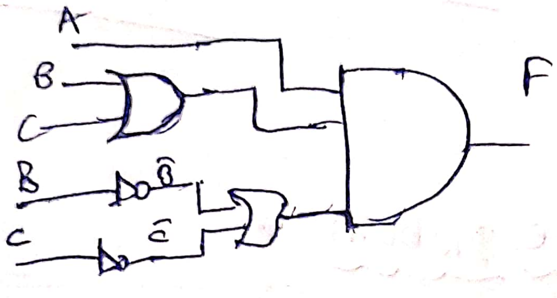
$(B+C)$

A	B	C
0	1	1
1		

$\bar{B}+\bar{C}$

A

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

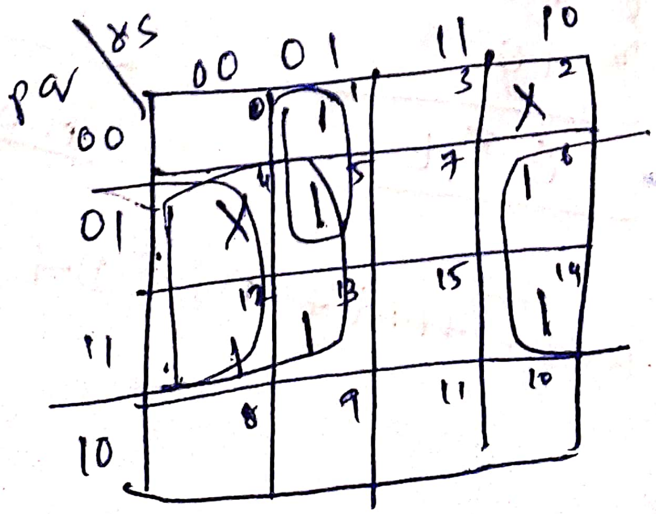


Don't care combinations :-

the "Don't care" condition says that we can use the blank cells of K-map to make a group of variables. To make "don't care" cells as a group of cells, we can use the "don't care" cells as a group of cells, we can use the don't care either 0 (or) 1, and if required, we can also ignore that cell.

→ We mainly use the "don't care" cell to make a large group of cells.

Reduce the expression $f(p,q,r,s) = \Sigma m(1,5,6,12,13,14) + d(2,4)$



p	q	r	s
0	1	0	0
1	1	0	1

$q\bar{r}$

p	q	r	s
0	1	0	0
1	1	0	0

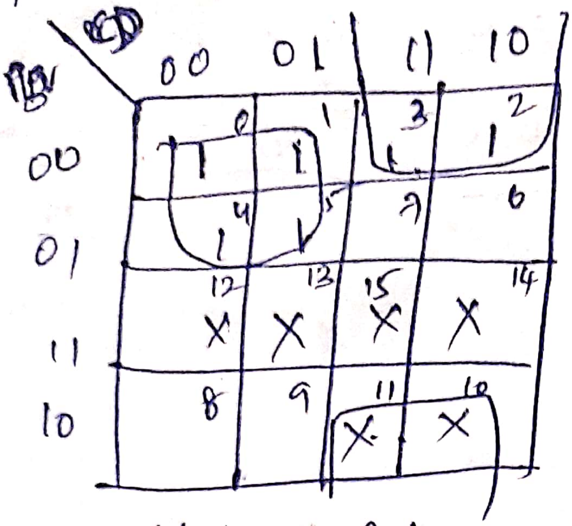
$q\bar{s}$

p	q	r	s
0	0	0	1
0	1		

$\bar{p}\bar{r}s$

$f = \Sigma(0,1,2,3,4,5) + d(10,11,12,13,14,15)$

$f =$



A	B	C	D
0	$\cancel{0}$	0	$\cancel{0}$
0	$\cancel{1}$	0	$\cancel{1}$

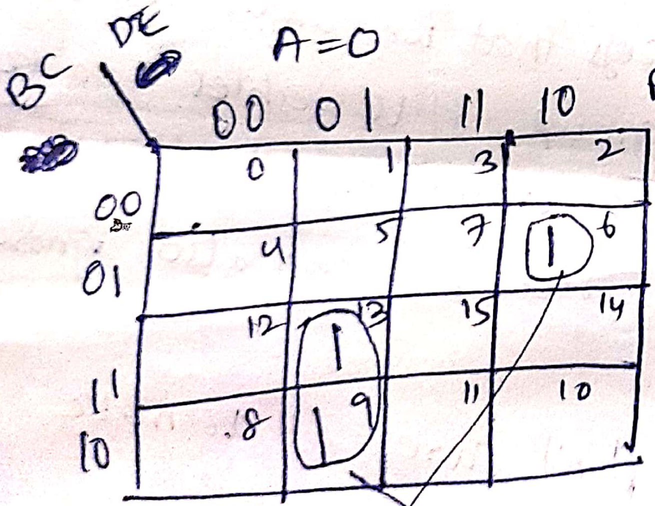
$\bar{A}\,\bar{C}$

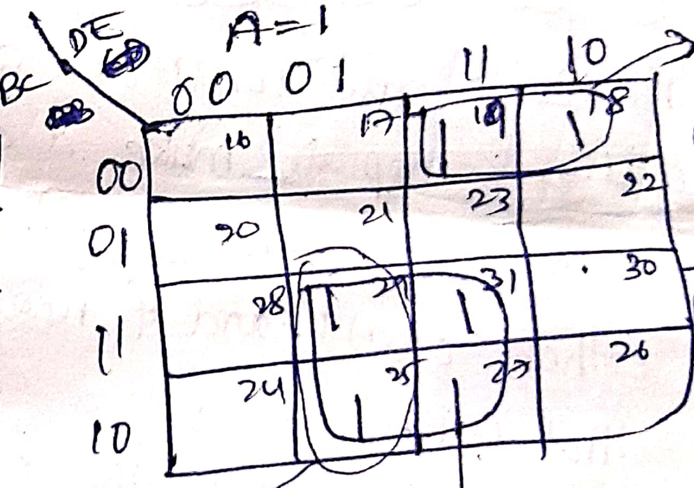
A	B	C	D
$\cancel{0}$	0	1	$\cancel{1}$
$\cancel{1}$	0	1	$\cancel{0}$

$\bar{B}C$

* 5-variable K-map Example problem

$f = \Sigma m(6,9,13,18,19,25,27,29,31)$





A	B	C	D	E
1	0	0	1	

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

A	B	C	D	E
0	0	1	1	0

$\bar{A}\bar{B}CD\bar{E}$

A	B	C	D	E
$\cancel{0}$	1	$\cancel{1}$	0	1
$\cancel{1}$	1	$\cancel{0}$		

$B\bar{D}E$

A	B	C	D	E
1	1	$\cancel{1}$	$\cancel{0}$	1
	1	$\cancel{0}$	$\cancel{1}$	1

ABE

$f = \bar{A}\bar{B}CD\bar{E} + ABE + A\bar{B}\bar{C}D + B\bar{D}E$

* Reduce the following function using K-map (9)

$f(A,B,C,D) = \Pi M(0,2,3,8,9,12,13,15)$

Sol Step 1 :- Given expression contains, 4 variables, hence four variable map ($2^4 = 16$ cells map) is required

Step 2 :- Plot the expression in to K-map

Step 3 :- Group the cells as adjacent cells. as pairs, quads and octets if possible

Step 4: Form the POS Expression

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

In pos form
If $A=0 \Rightarrow A$
If $A=1 \Rightarrow \bar{A}$.

A	B	C	D
0	0	1	~~1~~
		1	~~0~~

$A+B+\bar{C}$

A	B	C	D
1	~~1~~	0	~~0~~
1	~~0~~	0	~~1~~

$(\bar{A}+C)$

A	B	C	D
1	1	~~0~~	1
		~~1~~	1

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

A	B	C	D
~~0~~	0	0	0
~~1~~	0		

$B+C+D$

$\therefore f = (A+B+\bar{C})(\bar{A}+C)(\bar{A}+\bar{B}+\bar{D})(B+C+D)$

Properties of EX-OR Gates;

① $A \oplus A = 0$; output is logic zero, when inputs are same

② $A \oplus \bar{A} = 1$; output is logic one, when inputs are different

③ $A \oplus 1 = \bar{A}$; EX-OR as inverter

④ $A \oplus 0 = A$; EX-OR as non-inverter

⑤ EX-OR as modulo '2' adder

⑥ $A' \oplus B = A \oplus B' = (A \oplus B)'$

⑦ $A \oplus B = B \oplus A$

⑧ $A \oplus (B \oplus C) = (A \oplus B) \oplus C = A \oplus B \oplus C$

EX-OR Truth table:

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