SOP & POS Canonical Forms, Karnaugh Map Method

terms where all the variables complemented or uncomplemented

m.

evaluates to 1. The product terms ab an assigned value of 1

The boolean expression written clirectly truth table as f(a,b) = ab +ab is

Expressions constructed using such standard maxterns are said to be in Maxtern Canonical Forms or standard pos form.

While expressing functions in maxterm canonical form, we look out of rows for which the function evaluates to 'O' rather than '1'.

Consider the same function truth table used in Minterm Canonical form. The maxterm corresponding to row'o' for which the function evaluates to 'o' is at b Observe that a=b=0, a+b evaluates to 'o'. Similarly, for row 2 the maxterm $a+\overline{b}$ evaluates: to 'o' for a=0 b=1

Exercise 1

Write the Boolean expression in minterm canonical form for the function represented by the Itruth table below.

Row No.	a	Ь	C	19	1 terms
0	0	0	0	0	ābē
1	0	0	1	1	abc
2	0	1	0	0	abe
3	0	1	1	0	аьс
4	1	0	0,	1	abc
5	1	0	ı.	0	abc
6	1	1	0	1	abī
7	1	1	1	0	abc

expression in Minterm Canonical form by collecting the terms corresponding to which the function evaluates:

 $f(a,b,c) = \overline{ab}c + a\overline{bc} + ab\overline{c}$

Exercise 2

T

Write the Buolean expression in maxterm canonical form for the function represented by the truth table below.

Row No.	la	Ь	C	1 f	1 terms
O	0	0	0	1	a+b+c
1	0	0	1	1	a+b+c
2	0	1	0	0	a+5+c
3	0	1	1	1	a+5+c
4	1	0	0	1	a+b+c
5	1	O	1	0	ā+b+c
6	1	1	0	0	
7	1	1	1	1	a+b+c
			- 1		1

Solo: The function can be represented by a boolean expression in Mouterm Canonical form by collecting the terms corresponding to which the function evaluates to 'o'.

$$f(a,b,c) = (a+\overline{b}+c)(\overline{a}+b+\overline{c})(\overline{a}+\overline{b}+c)$$

m-Notation & M-Notation

The m-notation is used to simplify writing functions in minterms canonical forms. Minterms are represented as m; , where i stands for the row number for which the function evaluates to 1'.

From exercise 1, $f(a,b,c) = \overline{ab}c + \overline{abc} + \overline{abc}$ can be represented in m-notation as $f = m_1 + m_4 + m_6 \quad \text{or by using summation}$ notation, $f = \overline{\sum} m(1,4,6)$

The M-notation is used to simplify writing in marterm canonical form. Maxterms are represented as Mi, where it stands for the row not for which the function evaluates to o'

From exercise 2:

can be represented in M-notation as

or by using a product notation,
$$f = TIM(2,5,6)$$

Exercise 3: Convert the following to its minterm Canonical form. $f = \overline{a(b+c)} + \overline{c}$

$$\frac{501^{20}}{5} \cdot f = \overline{a}(\overline{b}+c)+\overline{c}$$

$$= \overline{a}\overline{b}+\overline{a}c+\overline{c}$$

$$= \overline{a}\overline{b}(c+\overline{c})+\overline{a}c(b+\overline{b})+\overline{c}(a+\overline{a})(b+\overline{b})$$

$$= \overline{a}\overline{b}c+\overline{a}\overline{b}\overline{c}+\overline{a}\overline{b}c+\overline{a}\overline{b}c+\overline{a}\overline{b}\overline{c}$$

$$f = \overline{abc} + \overline{abc} + \overline{abc} + \overline{abc} + \overline{abc} + \overline{abc} + \overline{abc}$$

$$= m_1 + m_0 + m_3 + m_6 + m_4 + m_2$$

Note: To make std. minterm, multiply (true + true) of missing variable to the non-std term.

Encercise 4. Convert the following to its maxterm Canonical form $f(a,b,c) = (b+\overline{c})(a\overline{b}+c)$

$$\frac{Sol2}{f} = (b+\overline{c})(a\overline{b}+c)$$

Applying clistributive law for second Lerw, we get

$$f = (b+\bar{c})(a+c)(\bar{b}+c)$$

$$= (b+\bar{c}+a\bar{a})(a+c+b\bar{b})(\bar{b}+c+a\bar{a})$$

$$= (a+b+\bar{c})(\bar{a}+b+\bar{c})(a+b+c)(a+\bar{b}+c)$$

$$(a+\bar{b}+c)(\bar{a}+\bar{b}+c)$$

$$f = (a+b+\bar{c})(\bar{a}+b+\bar{c})(a+b+c)(a+\bar{b}+c)(\bar{a}+\bar{b}+c)$$

$$= M_1 \cdot M_5 \cdot M_0 \cdot M_2 \cdot M_6$$

Note: To make std. Maxterm, add (true. true) of missing variable to the mon-std. term.

Minterms & Manterms for Three Variables

Row No.	Α	B	C	Minterms	Manterma
0 1 2 3 4 5 6 7	0000111	0 0 1 1 0 0 1 1	0 - 0 - 0 - 0 -	$\overline{A}\overline{B}\overline{C} = m_0$ $\overline{A}\overline{B}C = m_1$ $\overline{A}\overline{B}C = m_2$ $\overline{A}\overline{B}C = m_3$ $\overline{A}\overline{B}\overline{C} = m_4$ $\overline{A}\overline{B}C = m_5$ $\overline{A}\overline{B}C = m_6$ $\overline{A}\overline{B}C = m_7$	$A + B + C = M_0$ $A + B + \overline{C} = M_1$ $A + \overline{B} + C = M_2$ $A + \overline{B} + \overline{C} = M_3$ $\overline{A} + B + \overline{C} = M_3$ $\overline{A} + B + \overline{C} = M_3$ $\overline{A} + \overline{B} + \overline{C} = M_3$ $\overline{A} + \overline{B} + \overline{C} = M_3$

Complements of Canonical forms

Looking at the truth table of a given function, we can immediately say that the complement of the function will have is where the original functional values were 0 and vice versa

Term.	1 a	t	o c	18	1 7
mo	0	С	0	i	0
mı	0	0	T	0	
m_2	0	1	0		
m ₃	0	U	1	0	
m4	- [0	0	0	1 1 2 2
ms	[0	1	1	f(a,b,c) = 2m(1,3,4,6,7)
m6	1	1	0	0	in sop form.
m7	1	1	1	0	

Now, consider a function f = abc = ms Computing & using DeMorgan's Theorem gives

$$\vec{f} = \vec{abc} = \vec{a} + \vec{b} + \vec{c} = \vec{a} + \vec{b} + \vec{c} = Ms$$

Thus, a function can be complemented by simply changing the notation from m to M.

Consider $f(a,b,c) = mo + m_1 + m_5$

$$f(a,b,c) = \overline{m_0 + m_1 + m_5}$$

$$= \overline{m_0 \cdot \overline{m_1} \cdot \overline{m_5}}$$

$$f(a,b,c) = M_0 \cdot M_1 \cdot M_5$$
 in Posform.

Exercise 5: Find the complement of the following in sop and Pos form

(a)
$$f(abc) = \sum_{m} (0,3,7)$$

Sold For obtaining the complement in SUP form, Consider minterms not listed in f'

For obtaining the complement in POS form, complement f' and apply DeMorgan's thm.

$$f(abc) = \overline{m_0 + m_3 + m_7}$$

$$= \overline{m_0} \cdot \overline{m_3} \cdot \overline{m_7}$$

$$= M_0 \cdot M_3 \cdot M_7$$

$$f(abc) = \overline{II}M(0,3,7)$$
 Posform.

(b)
$$f(abcd) = Im(0,1,2,3,5,8,10,12,13,14)$$
.
(Sop form.

Exercise 6: Find the complement of the following function in SOP & pos form f(abc) = TIM (1,4,6,7)

Soly: Pos form
$$\overline{f}(a,b,c) = \overline{M}M(0,2,3,5)$$

SOP form
$$f(a,b,c) = M_1 \cdot M_4 \cdot M_6 \cdot M_7$$

= $M_1 + M_4 + M_6 + M_7$

Incomplete Boolean Functions & Don't Care Conditions

A PO

all.

allo

N-

1

34

Consider the following design specification:

→ Design a system which accepts 3 ilps a, b, & c and generates an output 'O', when the mo-of 1's in the input is one and generates an o/p'1' when the mo-of 1's in the i/p is ±wo.

S012

Row No.	a	Ь	C	18
0	0	0	0	×
1	0	0	1	0
2	O	l	0	0
3	0	1	1	1
4	1	0	0	0
5	l	0	1	1
6	1	1	0	1
7	l	1	1	X

The O/P is not specified for the first and last row of table. This table describes an incomplete Boolean function.

Truth tables whose OIP for all i/p combinations is either a '1' or a '0' describes a complete boolean funco.

The below table shows the complement of the incomplete boolean function of above table.

Row No.	a	Ь	C	18
0	0	0	0	X
ı	0	0	1	1
2	0	1	0	1
3	0	1	1	0
4	1	0	0	I
56	1	0	1	0
6	1	1	0	0
7	1	1	1	X

Observe that undefined olps continue to remain undefined other an incomplete boolean function is complemented.

All defined olps are however complemented

The unspecified olps in the truth table of an incomplete boolean function is called "Don't-CARE" conditions. Minterms or Maxterms are written for the specified olps are grouped together as clost-care terms.

The dont-care terms in above table's are written as dc(0,7).

The boolean function in minterm canonical form for the above function can be written as

On the same lines, the same boolean function can be expressed in maxterm canonical form as

In practical implementation, these don't cares may be taken either as 1'or o'

Don't - Care Conditions in Logic Design.

Dont - Care conclitions could arise in two scenarios.

- In that i/p condition for don't cares do occur, but opps are not defined.
- 2) Second Scenario is that ilp conditions for dent care olps
 never occurs eg: BCD number as it ranges only from
 0 to 9 taking 4 bits to represent, after 9 rest 6 olps are dont cares

Simplification of Boolean Expression Using Karnaugh Maps.

The complexity of the digital logic gates that implement a boolean function is directly related to the algebraic expression from which the function is implemented. Boolean expressions may be simplified by algebraic manipulation using set of rules & theorems. However this procedure of simplification is awkward, because it lacks specific rules to predict each succeeding! Step in the manipulative process and it is difficult to determine whether the simplest expression has been acheived.

By contrast, the map method provides a straightforward proceduce for simplifying boolean functions of up to four variables.

The map is also known as KARNAUGH Map or K-Map. Boolean functions can be diagrammatically represented by means of Karnaugh Maps.

A 3 variable function has $2^3 = 8$ rows in the truth table. A 4 variable function has $2^4 = 16$ rows in the truth table. Similarly a 'n' variable function would have $2^{n'}$ rows in the truth table.

A Karnaugh map is so drawn such that each row of the truth table corresponds to cell (square) in the map. A '1' is placed in the cell corresponding to a row for which the function evaluates to '1' and '0' is placed in the cells corresponding to a row for which the function evaluates '0'.

The cells (squares) are so arranged that physically adjacent cells are also logically adjacent.

Two terms are logically adjacent if they differ only in one literal (variable).

eg: abc is adjacent to abc, abc and abc.

ab is adjacent to ab and ab

abcd is adjacent to abcd, abcd, abcd &
abcd

Observe that a two variable term is logically adjacent to 2 terms, a three variable term is logically adjacent to 3 terms and an 'n' variable term is logically adjacent to 'n' terms.

On the map, a square (cell) corresponding to a 'n' variable term would be logically adjacent to 'n' other cells.

Two-Variable Map.

9

0

There are 4 minterms for a boolean function with two variables. Hence the two-variable map consists of 4 Squares, one of each minterm as shown below.

		1 B 0 1;		IA	В	Terms
mo mi	1	O ABO AB	0	0	0	ĀB
m ₂ m ₃	•	0 a 2 A 3	1	0	0	A B
P- 4 [0]		1 [48 [43]	3	1	1	AB
fig. (a)		fig. (b)	,			

In flo (b), map is redrawn to show the relationship between the two squares and two variable A & B.

The O and 1' marked on the left side and the top of map designate the values of the variables.

A function of two variables can be represented in a map by making the squares that correspond to the minterms of the function as shown below.

ow/square n	0.) a	16	1 f			
0	0	0	0	a \	0	1
1	0	1	1	^	O°	1
2	1	0	L	0	2	1
3	1	1	0	1	1	0

Note: To represent any function into the K-map, the function should be a standard function i.e., all its minterms must contain all the variables either in complemented or uncomplemented form. If not, convert the non-standard function to standard function and then map it into K-Map.

es:
$$f(a_1b) = a + b\overline{a}$$
 $\leftarrow non-s+d$ function.
 $= a(b+\overline{b}) + \overline{a}b$,
 $f(a_1b) = ab + a\overline{b} + \overline{a}b$ $\leftarrow s+d$ function.

corresponding to minterms of function.

Three-Variable Map.

There are eight minterms for 3 binary variables. Therefore, a three variable map consists of eight squares shown below, each square corresponding to particular i/p variable combinations.

				A Y	30	01	11	10
mo	m1	m ₃	m 2	0	ABC	AGC	ĀBC	ABC 2
m4	ms	m7	m ₆	1	ABZ	ABC 5	ABC 7	AG C
fi	'ğ' (α	(ن				fig (b)		

The map drawn in fig(b) is marked with binary no.s for each row and each column to shown the binary values of the minterms.

Note that the numbers along the columns do not follow the boncomy count sequence.

The characteristic of the listed sequence is that, only one bit changes in value from one adjacent column to the next:

The function of three variables can be represented in a map by marking the squares that corresponds to the minterms of the function.

eg: f(abc) = abc + abc + abc + abc + abc + abc = m1 + m0 + m3 + m6 + m4 + m2

Four-Variable Map.

There are 16 minterms for four binary variables, and therefore, a four-variable map consists of 16 squares. Shown below

mo	mı	ma	102
ma	m ₅	m7	ma
11/2	m,3	mis	mia
me	mq	7111	mic

AB V	00	01	1)	10
00	ĀĒCŌ	AGCD	3 ĀĒCO	2 Ā <u>ā</u> c <u>ō</u>
01	ABCD	ĀBĪD	- 7 ABCD	ABCD
U	12 4BCD	ABCD	15 46(D	14 ABCD
10	ABCD	480	ABLD 11	10 ABCD

The rows and columns are numbered so that only one bit of the binary number change in value between any two adjacent columns or rows.

The minterms corresponding to each square can be optained by combining the row number with the column number.

05/	cd				
00	00	01	11	10	
00	o°	1	1	3 2	
01	1 4	15	0 7	06	
11	12	13	0 15	0	
10	0 8	09	0"	1 10	

from the K-map given below.

a /6	/pc 00		1.1	10
0	1 °	0	03	1 2
1	1 4	15	07	1

SolT

 $f(a,b,c) = \overline{a}\overline{b}\overline{c} + \overline{a}b\overline{c} + a\overline{b}\overline{c} + a\overline{b}c + a\overline{b}c + a\overline{b}c$ The truth table is as follows.

Cell no.	1 a	Ь	C	1 f
0	0	0	0	1
1	0	0	1	0
2	0	1	0	1
3	0	1	1	0
4	L	0	0	1
5	1	0	1	1
6	1	1	0	0
7	1	1	1	1

Exercise: Construct the K-map for the following function f (abcd) = abcd + abcd + abcd + abcd + abcd

sol™: Let us locate the terms in truth table.

cell no.	10	ì	Ь	c d	minterms	J.
0	C) (0	0	abza	0
1	0	0	0	1	abad	0
2	0	0	-1	0	ābccī	1
3	0	C	1	1	ābcd	0
4	0	1	0	0	ābēcī	0
4	0	1	0	1	abical	1
6	0	1	1	0	abccī	0
7	0	1	1	1	abcol	0
8	1	0	0	0	abcd	1
9	l	0	0	1	abcd	0
10	1	0	1	0	abcot	0
11	1	0	1	1	abcd	0
12	1	1	0	0	abēcī	1
13	1	1	0	1	a b ccl	1
14	1	1	1 (0		1
15	1	1	1	1	abca	1
1					abcd,	0

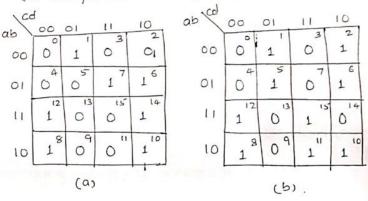
The Karnaugh Map can be drawn as follows.

100	1			
ab Y	00	01	11	10
	0	1	3	2
00	0	10	0	1
01	04	15	07	0
11	1 12	1 13	1 15	0
10	18	09	0"	010

Exercise: Write the Karnaugh maps for the following functions.

SolD: Place a 1' corresponding to each minterm in (a) and fill the remaining cells with 0's.

Place a 'O' corresponding to each monterm in (b) and fill the remaining cells with 1's.



Exercise: Write the k-Map for the function $f(abc) = a\overline{b} + \overline{a}b + \overline{b}c$

$$\frac{5012}{}$$
. $f(abc) = ab + ab + bc$

form of minterms are not having all the variables of function. So first we have to convert it to std. sop.

:
$$f(abc) = a\overline{b}(c+\overline{c}) + \overline{a}b(c+\overline{c}) + \overline{b}c(a+\overline{a})$$

 $= a\overline{b}c + a\overline{b}\overline{c} + \overline{a}bc + \overline{a}b\overline{c} + a\overline{b}c + \overline{a}\overline{b}c$
: $f(abc) = a\overline{b}c + a\overline{b}\overline{c} + \overline{a}bc + \overline{a}b\overline{c} + \overline{a}\overline{b}c$

P.T-0

0	0	0	ābē	10					
0	0	1	ābc	1					
0	l	0	ābē	1	abc	00	01	11	10
0	1	1	ābc	1	1	0	1	3	2
1	0	0	abe	1	0	0	1	1	1
	0	1	_	1	. 1	4	5	7	-
		01	a 6 c	1	1	1	1	0	0
	10	. /	abel	0	Į.		-		

Exercise: Write the K-map for the function.

$$\frac{Sol^2}{2}$$
 $f(abc) = (a+c)(b+c)(\bar{b}+\bar{c})$

As the above equ is not in standard POS. So we have convert it to std. Pos.

$$f(abc) = (a+b\overline{b}+c)(a\overline{a}+b+c)(a\overline{a}+\overline{b}+\overline{c})$$

$$= (a+b+c)(a+\overline{b}+c)(a+b+c)(\overline{a}+b+c)$$

$$(a+\overline{b}+\overline{c})(\overline{a}+\overline{b}+\overline{c})$$

$$(abc) = 6$$

$$f(abc) = (a+b+c)(a+\overline{b}+c)(\overline{a}+b+c)(a+\overline{b}+\overline{c})(\overline{a}+\overline{b}+\overline{c})$$

$$a b c | maxterm | f | a bc | a$$

Use of Kamaugh Maps to Simplify/Minimize Boolean Expressions

The simplified expressions produced by the map are always in SUP or POS form.

The simplest algebraic expression is one with a minimum mo. of terms and with the fewest possible no. of literals (variables) in each term.

This produces a two-level implementation having a logic circuit diagram with a minimum no of gates and the minimum no of ilps to the gates.

Rules for Grouping together adjacent cells/ squares containing 1's.

- 1) Groups must contain 1, 2, 4, 8, 16 ... 2 squares.
- 2) Groups must contain only 1 (and X' if dont-core is allowed).
- 3) Groups may be horizontal or vertical, but not diagonal.
- 4) Groups should be as large as possible
- 5) Each square/cell containing a'1' must be in atleast one group.
- 6) Groups may overlap
- Cell in a row may be grouped with the rightmost cell and top cell in the column may be grouped with the prouped with the bottom cell.
- 8) These should be as few groups as possible.

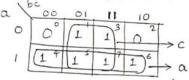
Note: Grouping '1' in map leads to minimal sop form. Grouping '0' in the map leads to minimal Pos form.

Rules for grouping O's in the map is same as for 1's explained above. Only difference is groups must contain only 'O' (and 'x' if dont-care is allowed)

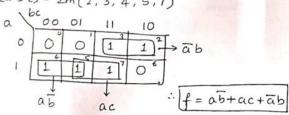
Example: Find the minimal sums for the following boolean functions

Sola

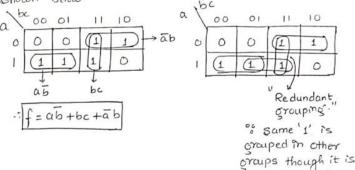
(a)
$$f(abc) = \sum_{m(1,3,4,5,6,7)}$$



in minimal sum of f(abc) = Zm(1,3,4,5,6,7) is f = a+c



The above example can also be solved in other way as shown below.

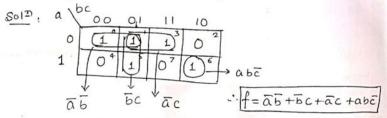


already grouped.
By this we get more terms
& that will not said to be
minimal sum expression.
(OR) efficient expression.

Note: Try to make as less as possible no of groups to get an efficient minimal sum expression.

(ii) Avoid more overlappings of the groups as it leads to Redundant Grouping as shown above.

Example: Find minimal sum for the following function f(abc) = IIM(2,4,7).

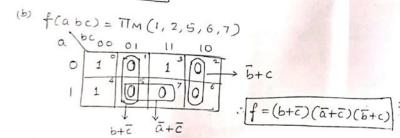


Here we can observe that '1' present in square of abc=001 is getting grouped thrice just to make bigger group. It is not called Redundant Grouping.

In the above eg the given function is POS function, but asked to get minimal sum ie, interms of SOP form. Bo, what we did is we mapped those numbers of TIM in K-map: by O's and rest all squares by 1's.
To get minimal sum, we grouped 1's instead of 0's.
If they asked to get minimal product account for the contract of the second of the se

If they asked to get minimal product expression for the same, function, we would have grouped 0's of the numbers of TIM. and get minimal product expression i.e., POS form.

Exercise: Find the minimal products of boolean. function f = Im(1, 3, 4, 5, 6, 7).



Here each group terms are written in the form of sums by taking variable value o' as true & variable value i' as complement & finally all sums are ANDed to get pos form.

Example: Write all the traininal sums and minimal products of the following Boolean functions.

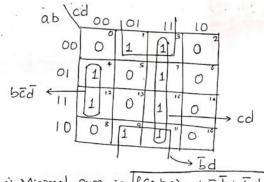
(b)
$$f(abcd) = (a+\overline{b})(a+c+d)(\overline{a}+\overline{b}+\overline{d})(a+\overline{c}+d)$$

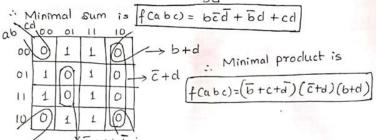
as the given function is not in Standard form. ie; all terms are not containing all the ilp variables.

So we have to convert it to std. form.

$$f(abcd) = \overline{abd}(c+\overline{c}) + bcd(a+\overline{a}) + a\overline{bd}(c+\overline{c}) + b\overline{cd}(a+\overline{a}).$$

= \$\bar{a}\bar{b}\cd + \ar{a}\bar{b}\cd + \ar{a}\ba





(b)
$$f(abcd) = (a+\overline{b})(a+c+d)(\overline{a}+\overline{b}+\overline{d})(a+\overline{c}+d)$$

Its not in Standard Pos, so we have to convert it to Std enpression.

$$= (a+\overline{b}+c\overline{c}+d\overline{d})(a+b\overline{b}+c+d)(\overline{a}+\overline{b}+c\overline{c}+\overline{d})(a+b\overline{b}+\overline{c}+\overline{d})$$

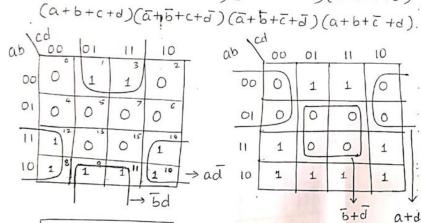
$$= (a+\overline{b}+c\overline{c}+d\overline{d})(a+b\overline{b}+c+d)(\overline{a}+\overline{b}+c\overline{c}+\overline{d})(a+b\overline{b}+\overline{c}+\overline{d})(a+b\overline{b}$$

$$= (a+\overline{b}+c+d)(a+\overline{b}+c+\overline{d})(a+\overline{b}+\overline{c}+d)(a+\overline{b}+\overline{c}+\overline{d})$$

$$(a+b+c+d)(a+\overline{b}+\overline{c}+d)(\overline{a}+\overline{b}+c+\overline{d})(\overline{a}+\overline{b}+\overline{c}+\overline{d})$$

 $(a+b+\overline{c}+d)(a+\overline{b}+\overline{c}+d)$. * Cancel the repeated terms

$$= (a+\overline{b}+c+d)(a+\overline{b}+c+\overline{d})(a+\overline{b}+\overline{c}+d)(a+\overline{b}+\overline{c}+\overline{d})$$



$$f(abcd) = ad + bd$$

Minimal Sum.

-: (f(abcd) = (b+d)(a+d)

Minimal Product.

Note: To get minimal sum, group 1's. To get minimal product, group 0's.)

Minimal Expressions of Incomplete Boolean functions.

Incomplete boolean expressions are those which evaluates to a clon't care conclition for some ilp combinations. It is represented as 'X' or' -' in the truth table.

Karnaugh maps of incomplete boolean expressions would have cells with 'X' entries besicles cells with 1's and O's. Such 'X' cells are called clon't care cells.

Minimal Sum expression is obtained by grouping 1's along with 'x' cells by considering 'x' as 1.

Similarly Minimal product expression is obtained by grouping 0's along with 'x' cells by considering 'x' us 0.

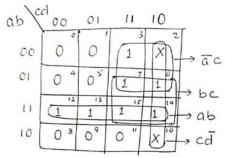
Note: Rules for grouping is as same as what we followed earlier.

Exercise: Find minimal sum of the following incomplete boolean functions using K-maps.

- (a) f(a b cd) = 2m (0, 2, 5, 7, 8, 10, 13, 15) + dc (1, 4, 11, 14)
- (b) f(abcd) = TIM (0,1,4,5,8,9,11)+c/c(2,10)

(a) f(a b cd) = 2m(0,2,5,7,8,10,13,15)+dc(1,4,11,14) : minimal sum is f(abcd) = ac+bd+bd+ac

(b) +(a b cd) = TIM (0,1,4,5,8,9,11) + dc(2,10).



· minimal sum is fabed) = ac+bc+ab+cd

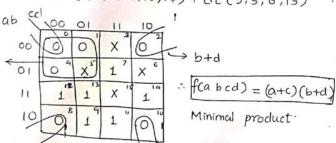
Here the given expression is POS form, so for the numbers of TIM we substituted 'O' in K map & dont care: as 'x'. Remaining cells will be filled by '1's. So to get minimal sum, group 1's along with x's.

incomplete boolean functions using K-maps.

(a)
$$f(a b ccl) = Im(7, 9, 11, 12, 13, 14) + dc (3, 5, 6, 15).$$

8012. fabca = Im (7,9,11,12,13,14) +dc (3,5,6,15)

Q+c



We can observe from the above example, that all'x's are not grouped. It is not necessary, because once all 0's are grouped with x's by considering those grouped x's as 0's its: clone. So no need to consider ungrouped x's as 0's and grouping. Because it violates the logic of given function.

i(b) f(a b cd) = TIM(2,8,11,15)+dc(3,12,14)

