

**Paper 7, TDC Part-3**  
**Chapter– 4, Combinational Logic Design**  
**Lecture - 4**

**By:**

**Mayank Mausam**

**Assistant Professor (Guest Faculty)**

**Department of Electronics**

**L.S. College, BRA Bihar University,**

**Muzaffarpur, Bihar**

# Combinational Logic Design

## Karnaugh Map Representation of Logic Functions.

As we know that before implementing any logical function (boolean expression) using logic gates, it is required to simplify the boolean expression. Sometimes it is difficult to simplify the boolean expression using Boolean algebraic theorems. Karnaugh map is another graphical technique which provides a systematic method for simplifying Boolean expressions.

# Combinational Logic Design

In this method, the information ~~contained~~ contained in a truth table or available in POS or SOP form is represented on a Map called Karnaugh Map (K-Map).

K-Map contain 'N' no. of cells where 'N' depends on the number of variable present in the Boolean Expression.

If no. of variables in the expression is 'n' then  $N = 2^n$

For example a boolean expression with 3 variable will have  $2^3 = 8 = N$  (number)

# Combinational Logic Design

of cell). The technique of K-Map may be used for any number of variables. In general it is used upto 6 variables, beyond this it becomes very cumbersome.

Figure below shows the K-Map for 2, 3 & 4 variables.

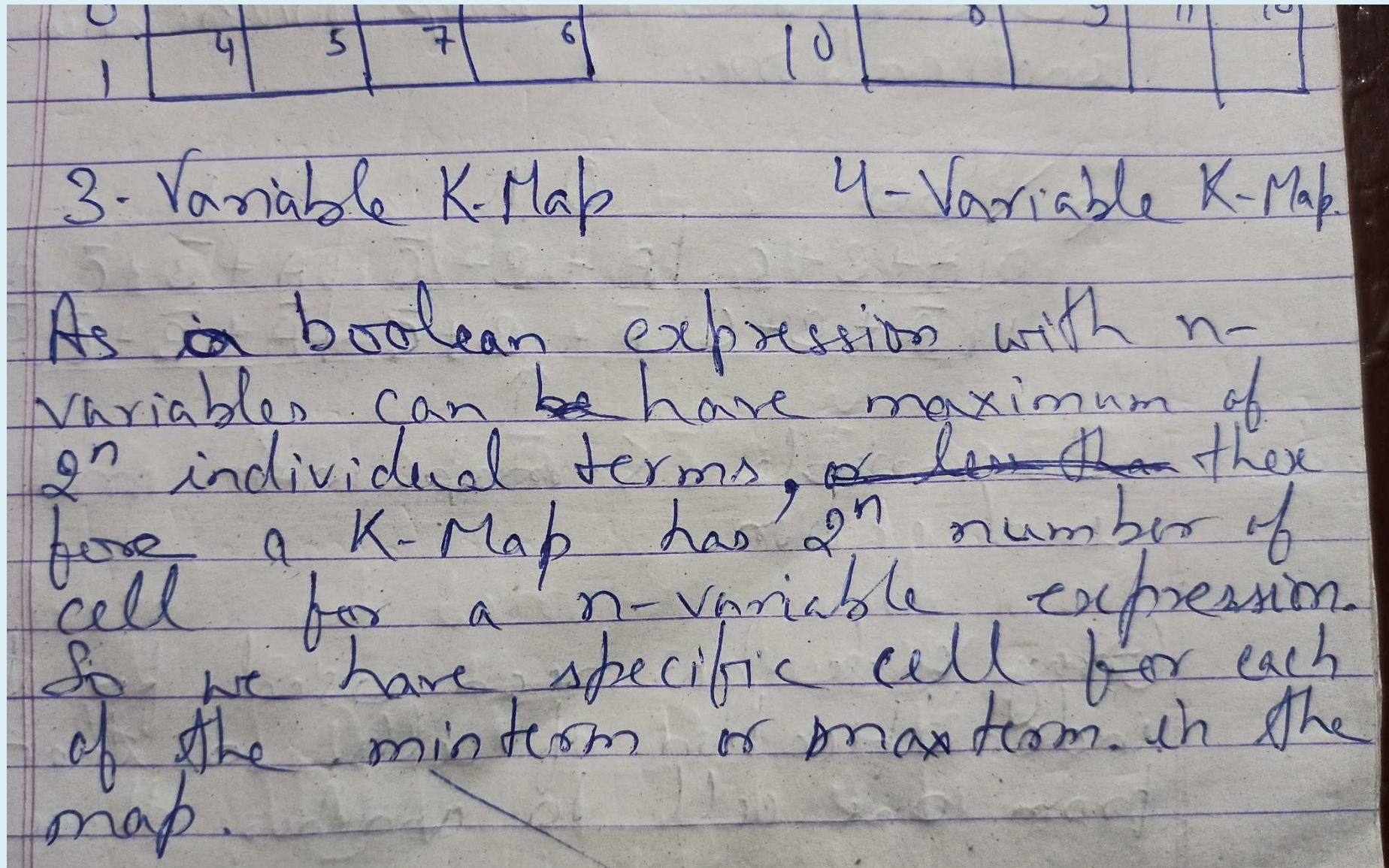
A \ B	0	1
0	0	1
1	2	3

2-Variable K-Map

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

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

# Combinational Logic Design



The image shows handwritten notes on a piece of lined paper. At the top, there are two Karnaugh maps. The first is a 3-variable K-map with 8 cells, labeled 0 through 7. The second is a 4-variable K-map with 16 cells, labeled 0 through 15. Below the maps, the text reads: '3-Variable K-Map' and '4-Variable K-Map.' followed by a paragraph explaining the relationship between the number of variables and the number of cells in a K-map.

3-Variable K-Map

4-Variable K-Map.

As a boolean expression with  $n$ -variables can have maximum of  $2^n$  individual terms, ~~or less than~~ there fore a K-Map has  $2^n$  number of cell for a  $n$ -variable expression. So we have specific cell for each of the min term or max term in the map.

# Combinational Logic Design

Figure below shows the minterm / maxterm corresponding to each cell.

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

↓

2-variable K-map with minterms

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

↓

2-Variable K-map with maxterms

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

↑

3 variable K-map with minterms indication.

# Combinational Logic Design

3 variable K-map with min terms indication.

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

3-variable K-map with max terms indication.

Gray code has been used for the identification of cells, so that moving from one cell to another there should

# Combinational Logic Design

be change of only one variable.

Now we can also write K-Map entry for 4-variables.

Representation of Truth Table on K-Map.

Let us take example of 3-variable logic function (Y) as below

# Combinational Logic Design

Truth table of Function Y.

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

So we can express the logic function Y in equation from the table

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

# Combinational Logic Design

<sup>simplify</sup>  
let us reduce the expression

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

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

$$Y = \bar{A}C + B\bar{C} \quad \text{--- (i)}$$

Looking into the truth table of function  $Y$ , we will now map the function  $Y$  into the K-map and then ~~solve~~ simplify the equation.

# Combinational Logic Design

To map a logical function in Sum of Products form into the K-map we enter '1' in each cell for which the function is true, i.e. 1.

So from the truth table we can see that the function  $Y$  is true for following minterms:  $\rightarrow$

$\bar{A}\bar{B}C$ ,  $\bar{A}B\bar{C}$ ,  $\bar{A}BC$  &  $AB\bar{C}$

Now drawing K-map for logical function  $Y$ .

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

K-Map for logical function  $Y$ .

# Combinational Logic Design

To solve (simplify) the logical function using K-Map, grouping of '1' is done in case of SOP form.

The grouping of '1' is done by taking 2, 4, 8, 16 ---- and so on number's of '1' i.e. dual, quad, octet ---- and so on. This has been discussed later.

## Combinational Logic Design

So after simplifying using K-Map, the logical function  $Y$  can be written as,

$$Y = \bar{A}C + B\bar{C} \quad \text{--- (ii)}$$

~~So (i)~~

So (i) & (ii) are equivalent.

Here we observe that K-map has simplified our function easily.

# Combinational Logic Design

Refer book- Modern Digital Electronics by RP Jain.

***Thank You***