Lecture 3

Minterm and Maxterm

Minterm and Maxterm

- To simplify the Boolean functions using k-map, it is necessary to understand the concept of sum of products (SoP), products of sum (PoS), minterms and maxterms.

SoP:- This form consists of two or more AND terms that are ORed together.

Ex:- $$f(A,B,C) = \underbrace{A\overline{B}}_{\text{product term}} + \underbrace{AB\overline{C}}_{\text{product term}} + \underbrace{B}_{\text{product term}}$$

In canonical SoP, each product terms must contain all the variable (literals) either in complemented form or in un-complemented form. In order to obtain the canonical SoP, multiply the first product term with $(C+\overline{C})$ and third product term with $(A+\overline{A})(C+\overline{C})$.

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

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

The terms $AB\bar{C}$ is appearing twice $\Rightarrow$ one can be neglected

since $AB\bar{C} + AB\bar{C} = AB\bar{C}$

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

- Each product term in canonical SOP form is call a minterm.

  So, above Boolean expression contains 6 min terms.

  The numbering of the minterm is performed as follows:

  $A\bar{B}C = 1\ 0\ 1$ — Assign '1' to variable without complement and '0' to variable with complement

Convert binary number 101 to decimal form

$$\underset{\text{Weight } 2^2}{1}\ \ \underset{\text{Weight } 2^1}{0}\ \ \underset{\text{Weight } 2^0}{1} = 1\times 2^0 + 0\times 2^1 + 1\times 2^2 = 1+4 = 5$$

$\therefore\ A\bar{B}C = m_5$ (lower case 'm' is used to represent min term)

Similarly, $A\overline{B}\,\overline{C} = m_4; AB\overline{C} = m_6; ABC = m_7; \overline{A}BC = m_3$ and $\overline{A}BC = m_3$

$$\therefore f(A,B,C) = m_2 + m_3 + m_4 + m_5 + m_6 + m_7 = \sum m(2,3,4,5,6,7)$$

Minterm can also be defined from truth table. Consider the truth table of Boolean function $f(A,B,C) = A\overline{B} + AB\overline{C} + B$

A	B	C	f(A,B,C)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Minterm is defined as the combination of inputs for which the function *f(A,B,C)* is having truth value of *1*.

$$\therefore f(A,B,C) = \overline{A}B\overline{C} + \overline{A}BC + A\overline{B}\,\overline{C} + A\overline{B}C + AB\overline{C} + ABC = \sum m(2,3,4,5,6,7)$$

Maxterm:

PoS:-This form consists of two or more OR terms which are ANDed together .

Ex:- f(A,B,C)= $(A+\overline{B})(A+B+\overline{C})B$

- In canonical PoS, each sum term must contain all the variables either in complemented form or un-complemented form.

Ex:- f(A,B,C)

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

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

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

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

canonical PoS form

In canonical PoS form, each product term is called maxterm.

(A+B+C)= 0 0 0 = M_0 ; '0' is assigned to variable without complement and '1' is assigned to variable with complement. Upper case letter 'M' is used to represent maxterm.

$$\therefore\ f(A,B,C) = M_0 M_1 M_2 M_3 M_4 M_5$$
$$= \prod M(0,1,2,3,4,5)$$

Maxterm can also be defined from truth table. Consider the truth table of Boolean function $f(A,B,C) = (A+\overline{B})(A+B+\overline{C})B$

A	B	C	f(A,B,C)
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Maxterm is defined as the combination of inputs for which the function *f(A,B,C)* is having truth value of *0*.

$$\therefore\ f(A,B,C) = (A+B+C)(A+B+\overline{C})(A+\overline{B}+C)(A+\overline{B}+\overline{C})(\overline{A}+B+C)(\overline{A}+B+\overline{C})$$
$$= \prod M(0,1,2,3,4,5)$$

- Gray code is used to number the rows and columns of k-map
  - In gray code any two consecutive code words differs by only one bit position.

Ex:- 2 bit binary code: 00,01,10,11

2 bit gray code : 00,01,11,10

<u>Binary to gray code conversion</u>

B=$B_3B_2B_1B_0$

$G = G_3G_2G_1G_0$

$G_3 = B_3;\ G_2 = B_3 \oplus B_2;\ G_1 = B_2 \oplus B_1;\ G_0 = B_1 \oplus B_0$

<u>Binary to gray code conversion</u>

$G = G_3G_2G_1G_0$

$B = B_3B_2B_1B_0$

$B_3 = G_3;\ B_2 = B_3 \oplus G_2;\ B_1 = B_2 \oplus G_1; B_0 = B_1 \oplus G_0$