21/06/24

Unit → 2

PAGE____
DATE__/__/__

Basic Logic circuits

BOOLEAN ALGEBRA :~

Mathematician George Boole invented a new kind of algebra - The algebra of logic in the year 1854 properly known as boolean algebra or switching algebra.

He states that symbol can be used to represent the structure of logical thought.

The boolean expression are basically defined by stating that first a constant is a boolean expression and second a variable is a boolean expression.

For example :~ 1) A is a boolean expression, so is $\bar{A}$. The combination of variables such as $\boxed{\bar{A}B \;\&\; \bar{A}B + C}$ are also boolean expression.

However $\boxed{A-B}$ is not a boolean expression.

* Boolean Logic operations :~

A Boolean function is an algebric expressions fared using binary constant binary variables and basic logical operation symbol. Basic logical operations includes the AND function

(Logical multiplication). the OR function (logical addition), and the NOT function (Logical complementation).

A Boolean function can be converted into a logic diagram composed of the AND, OR and NOT (inverter) gates.

* Logic AND Operation :-

The Logical AND operation of two Boolean variable A and B. given as y = A.B is represented by Table 2.1.

The common symbol for this operation is the multiplication sign (.). The Table shows that the result of the AND operation on the variables A and B is Logical 0 for all cases. except when both A and B are logical 1, usually, the dot denoting the AND function is omitted and A.B is written as AB.

Table 2.1 Logical AND operation

Inputs		Outputs
A	B	Y = A.B
0	0	0
0	1	0
1	0	0
1	1	1

PAGE______
DATE__/__/__

* Logical OR operation :~

The logical OR operation between Two Boolean variable A and B, given as $Y = A+B$ is represented by Table 2.2. This Table shows that the result of the OR operation on the variables A and B is logical 1 when A or B (or both) are logical.

The common symbol used for this logical addition operation is the plus sign (+).

Logical OR operation

Inputs		Outputs
A	B	$Y = A+B$
0	0	0
0	1	1
1	0	1
1	1	1

* Logical complementation (Inversion) :~ NOT Gate

The logical Inverse operation Converts the logical 1 to the logical 0 and vice versa. This method is also called the NOT operation. The symbol used for this operation is a bar over the function or the variable.

Serval notations such as a adding on asterisk, a star, prime etc. Over the variable, are used to include the NOT operation. "NOT A" or the complement of A is represented by $\bar{A}$.

PAGE____
DATE__/__/__

* Basic Laws of Boolean Algebra.

Logical operation can be expressed and minimized mathematically using the rules & Laws and theorem of Boolean algebra. It is convenient and systematic method of expressing and analying the operation of digital circuit and system.

Boolean algebra uses binary arithmetic variables which have two distinct symbol 0 and 1. These are called levels or states of logic.

For example, a binary 1 represents a high level and a binary 0 represents a low level.

* Boolean Addition :-

Addition by the Boolean method involves variables having value of either a binary 1 or a 0. The basic rules of Boolean addition are given below.

$$0 + 0 = 0$$
$$0 + 1 = 1$$
$$1 + 0 = 1$$
$$1 + 1 = 1$$

Boolean addition is same as the logical "OR" operation

PAGE ____
DATE __/__/__

* Boolean Multiplication :-

The Basic rules of the boolean multiplication method are as follows –

$$0 \cdot 0 = 0$$
$$0 \cdot 1 = 0$$
$$1 \cdot 0 = 0$$
$$1 \cdot 1 = 1$$

Boolean multiplication is same as the logical And operation.

* Properties of Boolean Algebra :-

Boolean Algebra is a mathematical system consisting of a set of two or more distinct elements. Two binary operators denoted by the symbols (+) and (·) and one unary operator denoted by the symbol either bar (¯) or prime (')

They satisfy the commutative, associative, distribute, absorption, consensus and idempotency properties of the Boolean Algebra.

Commutative property :-

Boolean addition is commutative given by

$$A + B = B + A \qquad (a)$$

PAGE___
DATE__/__/__

According to this property, the order of the OR operation conducted on the variables makes no difference. Boolean algebra is also commutative over multiplication, given by

$$A \cdot B = B \cdot A$$

This means that the order of the AND operations conducted on the variables makes no difference.

Associative property :-

The associative property of addition is given by :-

$$A + (B+C) = (A+B) + C$$

The OR operations of several variables results in the same, regardless of the grouping of the variables.

The associative low of multiplication is given by :

$$A \cdot (B \cdot C) = (A \cdot B) \cdot C$$

According to this low, it makes no difference in what order the variables are grouped during the AND operation of several variables.

PAGE____
DATE__/__/__

Distributive property :-

The boolean addition is distributive over Boolean multiplication given by :-

$$A + BC = (A+B)\ (A+C)$$

This property states that the AND operations (multiplication) of several variables and then the OR operation (addition) of the result with a single variable with each of the several variables and then the AND operation of the sums.

Proof.

$$A + BC = A\cdot 1 + BC \quad (\because A\cdot 1 = A)$$
$$= A\ (1+B) + BC \quad (\because 1+B = 1)$$
$$= A\cdot 1 + AB + BC \quad (\because A\ (B+C) = AB+BC)$$
$$= A\cdot (1\cdot C) + AB + BC \quad (\because 1+C = 1)$$

$$= A\cdot 1 + AC + AB + BC$$
$$= A\cdot A + AC + AB + BC \quad (\because A\cdot A = A)$$
$$= A\ (A+C) + B\ (A+C)$$
$$= (A+B)\ (A+C)$$

(ii) Boolean multiplication is also distribus over Boolean addition given by :-

$$A\cdot (B+C) = A\cdot B + A\cdot C$$

PAGE____
DATE__/__/__

According to this property, the OR operation of several variables and then the AND operation of the result with a single variable is equivalent to the AND operation of the single variable with each of the several variables and then the OR operation of the products.

Absorption law :-

(i) $\boxed{A + AB = A}$ (4a)

Proof

$$A + AB = A\cdot 1 + AB$$
$$= A(1+B)$$
$$= A\cdot 1 = A$$

(ii) $\boxed{A\cdot(A+B) = A}$ (4b)

Proof

$$A(A+B) = A\cdot A + A\cdot B$$
$$= A + AB$$
$$= A(1+B)$$
$$= A\cdot 1$$
$$= A$$

(iii) $\boxed{A + \overline{A}B = A+B}$ (5a)

Proof

$$A + \overline{A}B = (A+\overline{A})(A+B) \quad [\because A+BC = (A+B)(A+C)]$$
$$= 1\cdot(A+B) \quad [\because A+\overline{A} = 1]$$

PAGE______
DATE__/__/__

(iv) $\boxed{A\cdot(\bar{A}+B) = AB}$ (5b)

Proof
$$A\cdot(\bar{A}+B) = A\cdot\bar{A} + AB$$
$$= AB \quad (\because A\bar{A}=0)$$

Consensus Laws –

(i) $\boxed{AB+\bar{A}C+BC = AB+\bar{A}C}$ (6a)

Proof
$$AB+\bar{A}C+BC = AB+\bar{A}C+BC\cdot 1$$
$$= AB+\bar{A}C+BC(A+\bar{A})$$
$$(\because A+\bar{A}=1)$$
$$= AB+\bar{A}C+ABC+\bar{A}BC$$
$$= AB(1+C)+\bar{A}C(1+B)$$
$$(\because 1+B = 1 = 1+C)$$
$$= AB+\bar{A}C$$

(ii) $\boxed{(A+B)(\bar{A}+C)(B+C) = (A+B)(\bar{A}+C)}$ (6b)

Proof
$$(A+B)(\bar{A}+C)(B+C) = (A+B)(\bar{A}+C)(B+C+0)$$
$$= (A+B)(\bar{A}+B)(B+C+A\bar{A})$$
$$= (A+B)(\bar{A}+C)(B+C+A)(B+C+\bar{A})$$
$$= [\because A+BC = (A+B)(A+C)]$$
$$= (A+B)(A+B+C)(\bar{A}+C)(\bar{A}+C+B)$$
$$= (A+B)(\bar{A}+C) \quad [\because A(A+B)=A]$$

The order basic Laws (theorom) of Boolean algebra are given in Table 2.3.

These theorems can be proved early by adopting the truth table method or by using algebraic manipulation.

Table 2.3 other laws of boolean algebra

No.	Boolean laws	
7	(a) $A + 0 = A$ (b) $A \cdot 1 = A$	–
8	(a) $A + 1 = 1$ (b) $A \cdot 0 = 0$	
9	(a) $A + A = A$ (b) $A \cdot A = A$	Idempotency
10	(a) $A + \bar{A} = 1$ (b) $A \cdot \bar{A} = 0$	Fullset null
11	$\bar{\bar{A}} = \bar{\bar{A}} = A$	Double inch or involution

Principle of Duality

From the above properties and laws of Boolean algebra, it is evident that they are grouped in pairs as (a) and (b). One expression can be obtained from the order in each pair by replacing every 0 with 1. every 1 with 0, every (+) with (·) and every (·) with (+). Any pair of expresion satisfying this property is called dual expression. This characteristic of Boolean algebra is called the principle of duality.

PAGE____
DATE__/__/__

* DEMORGAN'S THEOREMS

Two theorems that are in important part of Boolean algebra were proposed by ~~demography~~ demoorgan. The first theorem states that the complement of a product is equal to the sum of the complements. That is, if the variables are A and B, then

$$\overline{AB} = \bar{A} + \bar{B}$$

The second theoram states that, the complement of a sum is equal to the product of the complement. In equation form, this can be written as

$$\overline{A+B} = \bar{A} \cdot \bar{B}$$

The complement of Boolean logic function or a logic expression may be expressed or simplified by following the steps of Demorgan's theorem.

(i) Replace the symbol (+) with symbol (·). the symbol (·) with symbol (+) given in the expression.

(ii) Complement each of the terms or variables in the given expression.

De Morgan's theorems can be proved for any number of variables, proof of these two theorems for 2-input variables can be found in table 2.4.

PAGE ___
DATE _/_/_

Proof for DeMorgan's theorem by perfect induction method.

1	2	3	4	5	6	7	8	9	10
A	B	$\bar{A}$	$\bar{B}$	$A+B$	$A \cdot B$	$\overline{A+B}$	$\bar{A} \cdot \bar{B}$	$\overline{A \cdot B}$	$\bar{A} \cdot \bar{B}$
0	0	1	1	0	0	1	1	1	1
0	1	1	0	1	0	0	0	1	1
1	0	0	1	1	0	0	0	1	1
1	1	0	0	1	1	0	0	0	0

A study of Table 2.4 makes clear that columns 7 and 8 are equal.
Therefore,

$$\overline{A+B} = \bar{A} \cdot \bar{B}$$

Similarly, columns 9 and 10 are equal.
Therefore,

$$\overline{A \cdot B} = \bar{A} + \bar{B}$$

Also, De-Morgan's theorem can be proved by algebric method as follow:-

According to the first theorem, $(\bar{A}+\bar{B})$ is the complement of AB. From the table 2.3, the Boolean laws 10 (a) and 10 (b) are given as,

$$A + \bar{A} = 1 \quad \text{and} \quad A\bar{A} = 0$$

Substituting AB for A and $(\bar{A}+\bar{B})$ for $\bar{A}$ in the above expressions.

$$AB + \bar{A} + \bar{B} = 1 \quad \text{and} \quad AB\,(\bar{A}+\bar{B}) = 0$$

$A + B + \overline{B} = 1$ and $AB\overline{A} + AB\overline{B} = 0$

$A + 1 = 1$ and $0 + 0 = 0$

$1 = 1$ $\qquad$ $0 = 0$

Thus De-Morgan's second theorem is proved algebrically.

Minimization (Simplification) of Boolean Expression using algebraic method.

The switching or Boolean expressions can be simplified by applying properties, laws and theorems of Boolean algebra. The simplification of different Boolean expressions are demonstrated in the following examples.

Example 2.1 :- Prove that $AB + BC + \overline{B}C = AB + C$

Solⁿ ⇒ $AB + BC + \overline{B}C = AB + C(B + \overline{B})$

$= AB + C \cdot 1$

$= AB + C$

Example 2.2 :- Simplify the expression $\overline{A} \cdot B + A \cdot B + \overline{A} \cdot \overline{B}$

Solⁿ ⇒ $\overline{A} \cdot B + A \cdot B + \overline{A} \cdot \overline{B} = (\overline{A} + A) \cdot B + (\overline{A} \cdot \overline{B})$

$= 1 \cdot B + \overline{A} \cdot \overline{B}$

$= B + \overline{A} \cdot \overline{B}$

$= B + \overline{A}$ $\quad (\because A + \overline{A} \cdot B = A + B)$

Example 2.3 :- Simplify the given expression $A + A\cdot\overline{B} + \overline{A}\cdot B$.

Sol^n ⇒ $A + A\cdot\overline{B} + \overline{A}\cdot B = A\,(1+\overline{B}) + \overline{A}\cdot B$

$= A\cdot 1 + \overline{A}\cdot B$

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

- Example 2.4 :- Complement the expression $\overline{A}B + C\overline{D}$

Sol^n ⇒ $\overline{\overline{A}B + C\overline{D}} = (\overline{\overline{A}B})\cdot(\overline{C\overline{D}})$

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

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

- Example 2.5 :- Simplify the expression $AB + \overline{AC} + A\overline{B}C\,(AB+C)$.

Sol^n ⇒ $AB + \overline{AC} + A\overline{B}C\,(AB+C)$

$= AB + \overline{AC} + A\overline{B}C\cdot AB + A\overline{B}C\cdot C$

$= AB + \overline{AC} + A\overline{B}C \quad [\because C\cdot C = C \text{ and } \overline{B}\cdot B = 0]$

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

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

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

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

$= \overline{A} + \overline{C} + 1 \quad [\because B+\overline{B} = 1]$

$= \overline{A} + 1$

$= 1$

PAGE ____
DATE __/__/__

Example :- 2.6 : Simplify the expression $Y = (\bar{A}+B)(A+B)$

Soln => $Y = (\bar{A}+B)(A+B)$

$= A\bar{A} + \bar{A}B + AB + BB$

$= 0 + B(\bar{A} + A + B)$

$= B(1+B)$

$= B \cdot 1$

$= B$

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

$= B + A\bar{A}$ $[\because (A+B)(A+C) = A+BC]$

$= B + 0$

$= B$

- Example : 2.7 : Simplify the expression $\overline{A\bar{B} + ABC} + A(B+A\bar{B})$.

Soln => $\overline{A\bar{B} + ABC} + A(B + A\bar{B})$

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

$=$

Example : 2.8 : Simplify $Y = ABC + A\bar{B}C + AB\bar{C}$ to $Y = A(B+C)$

Soln ⇒

$$Y = ABC + A\bar{B}C + AB\bar{C}$$
$$= AC(B+\bar{B}) + AB\bar{C}$$
$$= AC\cdot 1 + AB\bar{C} \quad \text{-----} \quad (B+\bar{B}=1)$$
$$= AC + AB\bar{C}$$
$$= A(C + B\bar{C})$$
$$= A(B+C)$$

Example : 2.9 : Simplify the given Boolean expression
$$Y = \bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + A\bar{B}\bar{C} + AB\bar{C}$$

Soln ⇒

$$Y = \bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + A\bar{B}\bar{C} + AB\bar{C}$$
$$= \bar{A}\bar{C}(\bar{B}+B) + A\bar{C}(\bar{B}+B)$$
$$= \bar{A}\bar{C}\cdot 1 + A\bar{C}\cdot 1 \quad \text{------} \quad (\bar{B}+B=1)$$
$$= \bar{A}\bar{C} + A\bar{C}$$
$$= \bar{C}(\bar{A}+A)$$
$$= \bar{C}\cdot 1 \quad \text{------} \quad (\bar{A}+A=1)$$
$$= \bar{C}$$

Example : 2.10 : Simplify the expression $Y = \overline{(AB+\bar{C})(\overline{A+B}+C)}$

Soln ⇒

$$Y = \overline{(AB+\bar{C})(\overline{A+B}+C)}$$
$$= \overline{(AB+\bar{C})(\bar{A}\cdot\bar{B}+C)}$$
$$= \overline{AB\cdot\bar{A}\cdot\bar{B} + ABC + \bar{A}\bar{B}\bar{C} + C\bar{C}}$$
$$= \overline{0 + ABC + \bar{A}\bar{B}\bar{C} + 0} \quad \text{------} \quad (A\cdot\bar{A}=0)$$
$$= \overline{ABC + \bar{A}\bar{B}\bar{C}}$$
$$= \overline{ABC}\cdot\overline{\bar{A}\bar{B}\bar{C}}$$
$$= (\bar{A}+\bar{B}+\bar{C})\cdot(\bar{\bar{A}}+\bar{\bar{B}}+\bar{\bar{C}}) \quad \text{--------} \quad (\overline{A+B} = \bar{A}\cdot\bar{B})$$
$$= (\bar{A}+\bar{B}+\bar{C})\cdot(A+B+C)$$

Example : 2.11 : Simplify the expression $Y = \bar{A}C[\overline{\bar{A}BD}] + \bar{A}B\bar{C}\bar{D} + A\bar{B}C$

Sol^n ⇒

$$Y = \bar{A}C[\overline{\bar{A}BD}] + \bar{A}B\bar{C}\bar{D} + A\bar{B}C$$

$$= \bar{A}C[A + \bar{B} + \bar{D}] + \bar{A}B\bar{C}\bar{D} + A\bar{B}C$$

$$= \bar{A}CA + \bar{A}C\bar{B} + \bar{A}C\bar{D} + \bar{A}B\bar{C}\bar{D} + A\bar{B}C$$

$$= 0 + \bar{A}\bar{B}C + \bar{A}C\bar{D} + \bar{A}B\bar{C}\bar{D} + A\bar{B}C \qquad [\because A \cdot \bar{A} = 0]$$

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

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

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

$$= \bar{B}C + \bar{A}\bar{D}(B + C) \qquad \because [A + \bar{A}B = A + B]$$

- **Example : 2.12 : Prove the following boolean expression:-**

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

Sol^n ⇒

$$L.H.S = (A+B)(\bar{A}\bar{C} + C)(\overline{\bar{B} + AC})$$

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

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

$$= [A\bar{A}\bar{C} + AC + \bar{A}B\bar{C} + BC][B(\bar{A} + \bar{C})]$$

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

$$= AC \cdot B\bar{A} + AC \cdot B\bar{C} + \bar{A}B\bar{C} \cdot B\bar{A} + \bar{A}B\bar{C} \cdot B\bar{C} + BC \cdot B\bar{A} + BC \cdot B\bar{C}$$

$$= 0 + 0 + \bar{A}B\bar{C} \cdot B\bar{A}$$

PAGE____
DATE__/__/__

Sum of Products and Products of sum

Logical functions are generally expressed in terms of logical variables. Values taken on by the logical functions and logical variables are in the binary forms. An arbitrary logic functions can be expressed in the following forms:

(i) Sum of Products (SOP)
(ii) Product of sums (POS)

* **Product term** :~ The AND function is referred to as product. In Boolean algebra, the word "product" loses it original meaning but serves to indicate an AND function. The logical product of several variable on a which function depends is considered to be a product term. The variables in a product term can appear either is complemented or uncomplemented form. $AB\bar{C}$, for example is a product term.

* **Sum term** :~ An OR function (+sign) is generally used to refer a sum. The logical sum of several variables on which a function depends is considered to be a sum term. Variables in a sum term can appear either in complemented or uncomplemented form $A+\bar{B}+C$, for example.

→ Sum of Products :~ (SOP)

The logical sum of two or more logical product terms is called a sum of products expression.

It is basically an OR operation of AND operated variable such as:

i) $Y = AB + BC + AC$

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

→ Product of sum (POS) :-

A product of sums expression is a logical product of two or more logical sum terms. It is basically an AND operation of OR operated variables such as:

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

ii) $Y = (A+B+C)\,(A+C)$

* Minterm :- A product term containing all the k variables of the function is either complimented or uncomplemented form is called a Minterm.

A 2-variable function has four possible combinations, viz. $\bar{A}\bar{B}$, $\bar{A}B$, $A\bar{B}$ and AB. These product terms are called minterms or standard products or fundamental products. For a 3-binary input variable function, there are 8 minterms as shown in the table given below. Each minterm can be obtained by the AND operation of all the variables of the function. In the minterm, a variable appears either in uncomplemented form, if it poses a value of 1 in the corresponding combination or in complemented form, if it contains

PAGE____
DATE__/__/__

the value 0. The minterm of a 3-variable function can be represented by $m_0, m_1, m_2, m_3, m_4, m_5, m_6$ and m_7 the suffix indicates the decimal code corresponding to the minterm combination.

A	B	C	Minterm
0	0	0	$\bar{A}\bar{B}\bar{C}$
0	0	1	$\bar{A}\bar{B}C$
0	1	0	$\bar{A}B\bar{C}$
0	1	1	$\bar{A}BC$
1	0	0	$A\bar{B}\bar{C}$
1	0	1	$A\bar{B}C$
1	1	0	$AB\bar{C}$
1	1	1	ABC

The main property of a minterm is that posses the value 1 for only one combination of k input variables; i.e, for a k variable function of the 2k minterms only one minterm will have the value 1, while the remaining $2^k - 1$ minterms will posses the value 0 for an arbitrary input combination 010, i.e, for A = 0, B = 1 and C = 0, only the minterm $\bar{A}B\bar{C}$ will have the value 1.
While the remaining seven minterms will have the value 0.

Cononical Sum of Product expression

It is defined as the logical sum of all the minterms derived from the rows of a truth table, for

PAGE____
DATE__/__/__

which the value of the function is 1. It is also called a minterm cononical form. The conohoical sum of product expression can be given in a compact form by listing the decimal codes in correspondence with the minterm containing a function value of 1. For example, if the cononical sum of product form of a 3-variable logic function Y has three minterms $\bar{A}\bar{B}\bar{C}$, $A\bar{B}C$ and $AB\bar{C}$, this can be expressed as the sum of the decimal codes corresponding to these minterms as started below:

$$Y = \Sigma m\ (0,5,6)$$
$$= M_0 + M_5 + M_6$$
$$= \bar{A}\bar{B}\bar{C} + A\bar{B}C + AB\bar{C}$$

Where Σm (0,5,6) represents the summation of minterms corresponding to the decimal codes 0,5 and 6.

Using the following procedure, the cononical sum of product form of a logic function can be obtained :-

1. Examine each term in the given logic function. Retain if it is a minterm; continue to examine the reset term in the same manner.

2. Check for variables that are missing in each product which is not a minterm. Multiply the product by $(x+\bar{x})$, for each variable x that is missing

3. Multiply all the products and omit the redundant terms.

→ The above procedures can be explained with the following examples :-

1. Obtain the canonical sum of product form of the function $Y(A, B, C) = A+BC$.

Sol^n => The given function containing the two variables.

A and B has the variable B missing in the first term and the variable A missing in the second. Therefore the first term has to be multiplied by $(B+\bar{B})$, the second term by $(A+\bar{A})$.

$$A+B = A\cdot 1 + B\cdot 1$$
$$= A\cdot(B+\bar{B}) + B\cdot(A+\bar{A})$$
$$= AB + A\bar{B} + BA + B\bar{A}$$
$$= AB + A\bar{B} + \bar{A}B \quad (\because AB + AB = AB)$$

$$Y = (A, B) = A+B = AB + A\bar{B} + \bar{A}B.$$

* Maxterm :- A sum term containing all the k variables of the function in either complimented or uncomplemented form is called Maxterm. A 2 variable function has four possible combinations, viz $A+B$, $A+\bar{B}$, $\bar{A}+B$ and $\bar{A}+\bar{B}$. These sum terms are called maxterm.

So also, a 3-binary input variable function has

8 maxterms. Each maxterm can be obtained by the OR operation of all the variables of the function. In the maxterm, a variable appear either in uncomplemented form if it possesses the value 0 in the corresponding combination, or in complemented, or in uncompleted form if it contains the value 1. The maxterms of a 3-variables function can be represented by M_0, M_1, M_2, M_3, M_4, M_5, M_6, and M_7. if the suffix indicates the decimal code corresponding to the maxterm combination.

A	B	C	Maxterm
0	0	0	$A+B+C$
0	0	1	$A+B+\overline{C}$
0	1	0	$A+\overline{B}+C$
0	1	1	$A+\overline{B}+\overline{C}$
1	0	0	$\overline{A}+B+C$
1	0	1	$\overline{A}+B+\overline{C}$
1	1	0	$\overline{A}+\overline{B}+C$
1	1	1	$\overline{A}+\overline{B}+\overline{C}$

The most important property of a maxterm is that it possesses the value 0 for only one combination of k input variables, i.e, for a k variables function of the 2^k maxterms, only one maxterm will have the value 0, while all the remaining 2^k-1 maxterms will have the value 1 for an example arbitary input combination 100, i.e. for A = 1, B = 0, C = 1, only maxterm $(\overline{A}+B+\overline{C})$ will have the value 0, while the remaining seven maxterms will have the value 1.

→ Conoonical Product of sum expression

This is defined as the logical product of all the maxterms derived from the rows of truth table, for which the value of function is 0. It is also known as the maxterm canonical form. The canonical product of sum expression can be given in a compact form by listing to the maxterms containing a function value of 0.

For example, if the canonical product of sum form of a 3-variable logic function Y has four maxterms $(A+B+C)$, $(A+\bar{B}+C)$, $(\bar{A}+B+C)$ and $(\bar{A}+\bar{B}+\bar{C})$ then it can be expressed as the product of decimal codes as given below:-

$$Y = \Pi(0, 2, 4, 7)$$
$$= M_0, M_2, M_4, M_7$$
$$= (A+B+C)\ (A+\bar{B}+C)\ (\bar{A}+B+C)\ (\bar{A}+\bar{B}+\bar{C})$$

* The following procedure can be used to obtain the conoonical product of the sum form of a logic function.

1. Examine each term in the given logic function. Retain it if it is a maxterm continue to examine the next term in the same manner.

PAGE______
DATE__/__/__

2. Check for variables that are missing in each sum, which is not a maxterm. Add $(X\bar{X})$ to the sum term, for each variable X that is missing.

3. Expand the expression using the distributive property and eliminate the redundant term.

Example :- Canonical product of sum form

$$Y = A + \bar{B}C$$
$$= (A+\bar{B}).(A+C)$$
$$= (A+\bar{B}+\bar{C}) + (A+C+B\bar{B})$$
$$= (A+\bar{B}+C)\ (A+\bar{B}+\bar{C})\ (A+B+C)\ (A+\bar{B}+C)$$
$$[\because (A+B)(A+\bar{B}) = A]$$

$$Y = (A+\bar{B}+C)\ (A+\bar{B}+\bar{C})\ (A+B+C)$$
$$Y = M_2\ M_3\ M_0 \quad \text{OR} \quad Y = M_0\ M_2\ M_3$$
$$\therefore Y = \Pi(0,2,3)$$

Driving Sum of Product (SOP) Expression from a Truth table

The sum of product (SOP) expression for a boolean expression function can be derived from its truth table by summing (OR) operation the product terms that correspond to the combinational containing a function value 1. In the product term, the input variable appears either in uncomplemented form of it possesses the value 1, or in complement form if it contains the value 0.

Now, consider a truth table, for a 3-input function Y, Here the value is 1 for the input combination 010, 011, 101 and 111 and their corresponding product terms are $\bar{A}B\bar{C}$, $\bar{A}BC$, $A\bar{B}C$ and ABC respectively.

INPUTS			OUTPUT	PRODUCT	SUM
A	B	C	Y	TERMS	TERMS
0	0	0	0 ~~maxterm~~		$(A+B+C)$
0	0	1	0 minterm		$(A+B+\bar{C})$
0	1	0	1 — ~~Minterm~~ Maxterm	$\bar{A}B\bar{C}$	
0	1	1	1 —	$\bar{A}BC$	
1	0	0	0		$(\bar{A}+B+C)$
1	0	1	1 —	$A\bar{B}C$	[illegible]
1	1	0	0		$(\bar{A}+\bar{B}+C)$
1	1	1	1 —	ABC	

~~$\Sigma(1,2,\ldots)$~~

Sum of product $\Sigma(0,1,4,6)$

Product of sum $(2,3,5,7)$

Now, the final SOP expression for the output 'Y' is obtained by summing (OR operation) of the four product terms as follows:

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

The procedure for obtaining the output expression is SOP form a truth table can be summarised, in general as follows.

PAGE____
DATE__/__/__

1.) Give a product term for each input combination in the table, containing an output value of 1.

2.) Each product term contains its input variables in either complemented or uncomplemented form. If an input variable is 0 its appear in complemented form; if the input variable is 1, it appears in uncomplemented form.

3.) All the product terms are OR operated together in order to produce the final SOP expression of the output.

Deriving Product of sum (POS) Expression from a Truth table

The product of sum (POS) expression for a Boolean (switching) function can also be obtained of the SUM terms corresponding to the combination for which the function assumes the value 0 In the sum term, the input variables appears in an uncomplemented form if it has the value 0 in the corresponding combination and in the compliments form if it has the value 1

Studying the truth table, for a 3-input function Y, we find that the Y value is 0 for the input combinations 000, 001, 100 and 110 and that their corresponding sum terms are $(A+B+C)$, $(A+B+\overline{C})$, $(\overline{A}+B+C)$ and $(A+\overline{B}+C)$

PAGE______
DATE__/__/__

Now the final POS expression for the output Y is obtained by the AND operation of the four sum terms as follows:

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

The procedure for obtaining the output expression in POS form from a truth table can be summarised in general, as follow:

1.) Give a sum term for each input combination in the table, which has an output value of 0.

2.) Each sum term contains all its input variables in complemented and uncomplemented form.
If the input variable is 0, then it appear in a uncomplemented form; if the input variable is 1, it appears in the complemented form.

3.) All the sum terms are AND operated together to obtain the final POS expression of the output.

The POS expression for a Boolean function can also be obtained from its SOP expression using Y.Y as given in the following example:-

Consider a function,

$$Y = \overline{A}B\overline{C} + \overline{A}BC + A\overline{B}C + ABC$$

$\overline{Y} =$ ________________

PAGE ____
DATE __/__/__

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

The complement $\bar{Y}$ can be obtained by the OR operation by the minterms which are not available in Y.

Therefore,

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

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

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

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

Karnaugh Map :-

The simplification of the switching function using Boolean laws and theorems becomes complex with the increase in the number of variables and terms. The karnaugh map technique provides a systematic method for implifying and manipulating switching operations.

In this technique the information contained in a truth table or variable in the POS or SOP form is represented on the karnaugh map.

The K-map is actually a modified form of a Truth table. Here, the combination are convenintly arranged to aid the simplification process by applying the rule $Ax + A\bar{x} = A$.

In an n variable K-map, there are 2^n cells. Each cell corresponds to one combination of n variables. Therefore, for each row of truth table, i.e. for each row of truth table, i.e. for each maxterm and minterm, there is one specific cell in the k-map.

The k-map for 2, 3 and 4 variables are shown in the figure 2.1. The decimal codes corresponding to the combination of variables have been marked as A, B, C and D and the binary number formed by them are taken as AB, ABC and ABCD for 2, 3 and 4 variables.

(a) Two variables :~

B \ A	0	1
0	0	2
1	1	3

(b) Three variables :~

C \ AB	00	01	11	10
0	0	2	6	4
1	1	3	7	5

(c) Four variables :~

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

PAGE ____
DATE __/__/__

The 3 and 4 variables k-maps show that the column and row headings used in representing the cells. are cyclic or unit distance code which result in adjacent cells, differing in just one variable. This helps the grouping of the adjacent cells and their in simplification by the application of the rule

Ax + Ax' = A. In addition the left and right most cells of the 3-variables k-map are adjacent.

For example, the cells 0 and 4 are adjacent. This is because each pair differs in just a single variable. In that 4 variable k-Map, the cells to the extreme left and right as well as those at the top and bottom-most position are adjacent.

A collection or group of 2^m cells. Each adjacent to m cells is called group. This group can be expressed by a product containing n-m variables where n is the number of variables in the k-map.

For example, in a 4-variable k map (i.e n = 4), if a group of 4 (i.e $2^m = 4$; m = 2) 1s is formed then this group can be expressed by 4-2=2 variables.

Similarly, if a group of eight 1s are combined then this group can be expressed by 4-3 = 1 variables. This can be better understood by the examples given latter.

This entries in a truth table can be represented in a k-map give below.

INPUTS			OUTPUTS
A	B	C	Y
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Truth Table of a digital system

Here, the output Y can be written as

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

$$Y\ (A,B,C) = m_1 + m_2 + m_4 + m_4$$

→ The K map for the above three - variable expression is shown below:

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

The value of the output variables Y (0 or 1) for each row of the truth table is entered in the corresponding cells of the k-map.

PAGE______
DATE__/__/__

Simplification is based on the principle of combining the terms present in adjacent cells. The 1s in the adjacent cells and be grouped by drawing a loop around those cells following the given rules.

1. Construct the k-map and enter the 1s in those cells corresponding to the combination for which function value is 1, then enter the 0's in the other cells.