

# Logic Gates and Boolean Algebra

## Logic Gates

Digital signal has two discrete levels or values 1 and 0, and are referred as HIGH (ON) and LOW (OFF) states.

A transistor operates very reliably in the switching mode. So transistor is one of the main building blocks of a digital circuit.

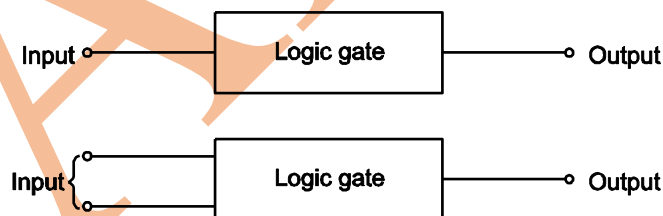
The simplest way to use a transistor is a switch, meaning that we operate it at either saturation or cut-off.

When saturated, a transistor is like a closed switch, whereas cut-off is like an open switch.

A diode can also be used as a switch. In forward bias condition, resistance of diode is zero and it behaves like closed switch. On the other hand, under reverse bias, the resistance is very high and no current can flow, which makes it as an open switch.

The logic gate has one or more inputs.

Depending upon the inputs it produces a HIGH (1) or LOW (0). A gate receives binary input and produces output which depends upon the inputs and the function intended for.





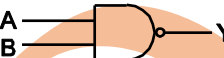




**Fig.: Block diagram of logic gate**

For simplicity in designing and understanding of digital system, each gate has assigned symbol.

There are many types of symbols depending on function they perform.

The most commonly used symbols are given in Table 1:

Logic gates

Sr. No.	Logic gate	Function	Symbol
1.	Inverter	NOT	
2.	AND	AND	
3.	NAND	NAND	
4.	OR	OR	
5.	NOR	NOR	
6.	Exclusive OR	EX-OR	
7.	Exclusive NOR	EX-NOR	

### Positive and Negative Logic

In a digital system there are **two discrete levels** *HIGH (1) and LOW (0)*.

If the higher of the two voltages represents a 1 and lower voltage represents 0, the system is called ***positive logic*** system.

On the other hand, if lower voltage represents a 1 and higher voltage represents 0 we have ***negative logic*** system.

Suppose that +5V and 0V are our logic level voltages.

We will designate +5V as HIGH (1) and 0V as LOW (0).

So positive and negative logic can be defined as :

<b>Positive logic</b>	<b>Negative logic</b>
HIGH = 1	HIGH = 0
LOW = 0	LOW = 1

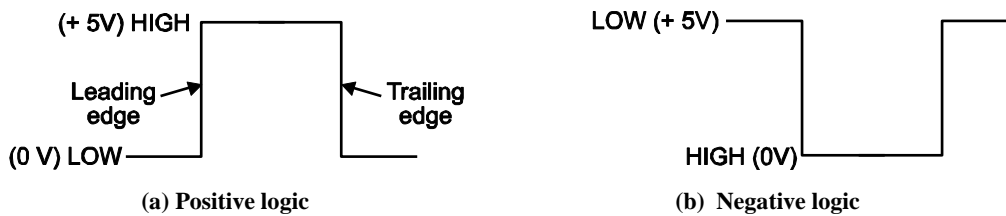


Fig. : Digital signal representation

From Fig.(a), + 5V will be considered as HIGH level in positive logic and LOW level in negative logic system.

Similarly, voltage level 0V will be considered as LOW in the positive logic system and HIGH level in the negative logic system, as in Fig. (b).

### **Basic Logic Gates (Universal and Derived Gates)**

In a digital system, there are only few basic operations performed irrespective of the complexities of the system.

These operations may be required to be performed number of times in a digital computer or digital control system.

The basic logic gates are AND, OR, NOT etc. These basic gates can be combined to perform other important logic operations like NAND, NOR and EX-NOR gates. So these are called as **derived gates**.

Any Boolean (or logic) expression can be realized by using the AND, OR and NOT gates. NAND, NOR operations can be derived from it.

These operations have become very popular and are widely used because either NAND, NOR gates are sufficient for the realization of any logical expression. Because of this reason, NAND and NOR gates are known as **universal gates**.

#### **1. NOT Gate**

The NOT gate is also called **inverter** as it inverts the input signal.

The inverter (NOT circuit) performs a basic logic function called inversion or complementation.

It is called NOT because its output is NOT the same as input.

The purpose of NOT or inverter is to change one logic level to the opposite level.

In terms of bits, it changes 1 to 0 and 0 to 1.



Input A	Output Y = $\bar{A}$
0	1
1	0

(c) Truth table

**Fig.1: NOT gate**

Fig. 1. (a) and (b) shows standard symbols for NOT gate.

As shown in Fig. 1, NOT gate has one input A and one output Y.

Its **logic expression** is,

$$\begin{aligned} Y &= \text{NOT } A \\ &= \bar{A} \end{aligned}$$

and is read as "**Y equals NOT A**" or Y equals complement of A.

$$\text{If } A = 0, \quad Y = \text{NOT } 0 = 1$$

On the other hand,

$$\text{if } A = 1, \quad Y = \text{NOT } 1 = 0$$

The presence of small circle, known as the **bubble**, always denotes inversion in digital circuits.

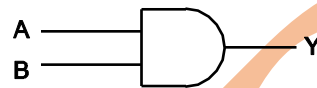
## 2.AND Gate

The AND gate performs **logical multiplication**, more commonly known as **AND** function.

The AND gate is composed of two or more inputs and single output.

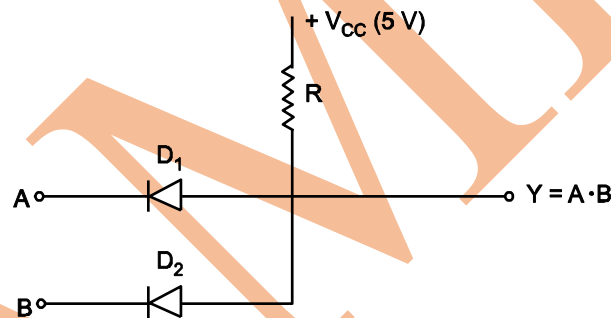
The output of AND gate is high (1) only when all the inputs are high (1).

A logic gate can be used as diode, transistor, FET or combination of these elements.



(a) Symbol

The inputs of the two input AND gates are labelled as A and B and output is labelled as Y.



(b) Simplified circuit with diode

Inputs		Output
A	B	Y = A · B
0	0	0
0	1	0
1	0	0
1	1	1

(c) Truth table

**Fig. 2**

The AND gate can be designed using diodes. Fig. 2 (b) shows two input AND gates using diodes.

There are **only four** possible **input cases**.

**Case I : A is low and B is low.**

With this situation both diodes  $D_1$  and  $D_2$  are forward bias by supply voltage and will conduct. Because of this the output voltage is ideally zero. This means Y is low.

i.e.  $Y = 0$

**Case II : A is low and B is high** since diode  $D_1$  is forward bias pulling the output down to a low voltage.

The diode  $D_2$  is reversed bias. The output  $Y = 0$ .

**Case III : A is high and B is low.** The diode  $D_2$  is forward bias and pulling the output down to a low voltage.

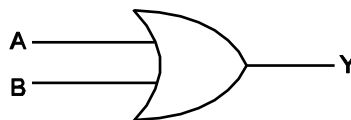
The diode  $D_1$  is reversed bias, therefore  $Y = 0$ .

**Case IV : A is high and B is high** with both inputs at + 5V. Both diodes are non-conducting because the voltage across each is zero.

Therefore, current will not flow through R and output Y is at high (logic 1) i.e.  $Y = 1$ .

### **3. OR Gate**

An OR gate performs logical addition, more commonly known as OR function. It has two or more inputs and one output. The logic symbol of OR gate is shown in Fig. 3 (a).



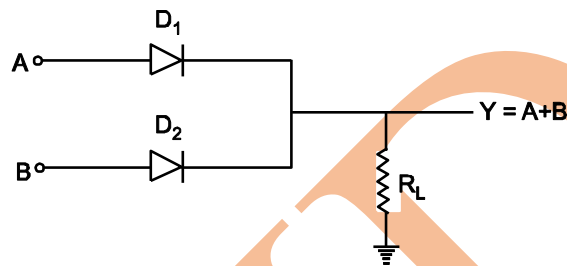
**(a) Symbol**

The inputs of two input OR gates are labelled as A and B. Output is labelled as Y.

The operation of OR gate is such that a HIGH on the output is produced when any of the inputs are HIGH.

Output is LOW only when all of the inputs are LOW.

In certain situation, if we want a output HIGH, when one or more of its inputs is HIGH, OR gate can be used.



(b) Diode circuit of OR gate

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

(c) Truth table

Fig. 3

Let us assume input voltage 1 (High state) or + 5V and 0 (Low state) is 0 V.

For different input combinations, we will have the following cases :

**Case I : A and B both LOW** i.e.  $A = 0$  and  $B = 0$ . In this case, both diodes  $D_1$  and  $D_2$  are non-conducting, therefore, Y output is LOW. i.e.  $Y = 0$ .

**Case II : A is LOW and B is HIGH** i.e.  $A = 0$  and  $B = 1$ .

The high B input voltage of the diode D<sub>2</sub> will conduct as it is under forward bias condition. The conducting current will flow through resistor R<sub>L</sub> and high voltage will develop at the output. This will make Y = 1 (High logic). The diode D<sub>1</sub> is reversed bias.

**Case III :** A is HIGH and B is LOW i.e. A = 1 and B = 0.

The diode D<sub>1</sub> will conduct and D<sub>2</sub> is non-conducting. The conducting current in this case also flows through resistor R<sub>L</sub> and HIGH voltage will develop at the output. This will make Y = 1 (High logic).

**Case IV :** A and B both are HIGH i.e. A = 1 and B = 1. Diodes D<sub>1</sub> and D<sub>2</sub> will conduct and therefore output is at high logic level i.e. Y = 1.

### Universal and Derived Gates

Any Boolean (or logic) expression can be realized by using the AND, OR and NOT gate as discussed before.

From these three operations, two more operations have to be derived : the **NAND and NOR operations**. These operations are very popular and are widely used.

**NAND and NOR gates are known as *universal gates* because it can perform any logical function of AND, OR and NOT gate. NAND or NOR gates are sufficient for the realization of any logical expression.**



## Universal Gate (NAND/NOR)

Function	Symbol	NAND	NOR
1. NOT			
2. AND			
3. OR			

The NAND and NOR gates have universal property and can be used for performing AND, OR and NOT functions, so an AND/OR/NOT logic circuit can be converted to NAND/NOR logic.

### NAND as NOT Gate

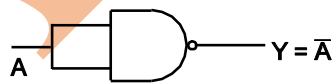


Fig. 4

$$Y = \overline{A \cdot A} = \overline{A}$$

A NOT gate can be obtained from NAND gate **by connecting all the inputs together.**

NAND as AND Gate

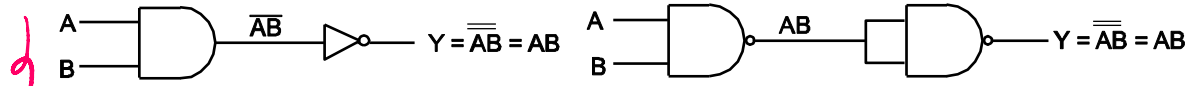


Fig. 5

The AND gate can be obtained by simply inverting output of NAND gate.

**NAND as OR Gate**

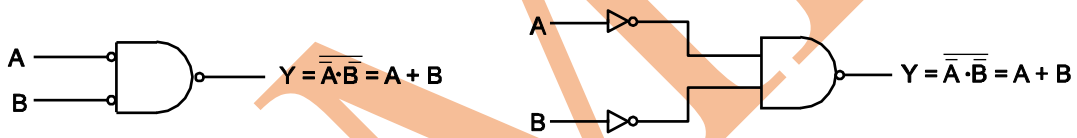
The OR gate can be designed by using NAND gate.

$$Y = A + B$$

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

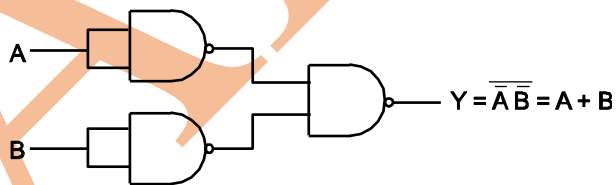
$$Y = \overline{\bar{A} \cdot \bar{B}}$$

Therefore an OR gate operation can be obtained by NANDing  $\bar{A}$  and  $\bar{B}$ .



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

NOT NAND



OR gate using NAND gate

Fig. 6

## NOR as NOT Gate

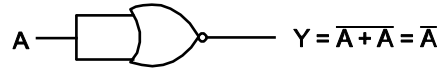


Fig. 7

Two inputs of OR gate are connected together, it gives inverter of input signal i.e. NOT.

NOR as AND Gate

The AND gate can be designed by using NOR gate.

$$Y = A \cdot B$$

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

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

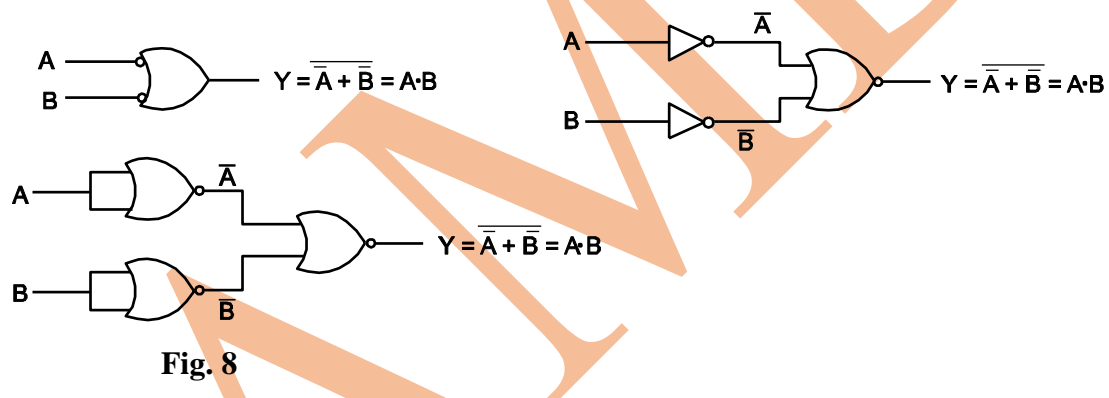


Fig. 8

## **Boolean algebra**

Boolean algebra is the mathematics of digital system.

In fact it is a convenient and systematic way of expressing and analyzing the operation of logic circuits.

It is found that the knowledge of Boolean algebra is essential to study and analysis of logic circuit.

**A mathematical system for formulating logical statements with symbols, so that problems can be written and solved like a ordinary algebra**, was developed by the Irish logician and mathematician George Boole and known as ***Boolean algebra***.

Boolean algebra was introduced in 1854. Boole is one of the persons in a long historical chain who were concerned with formalizing and mechanizing the process of logical thinking.

Boolean algebra is a generalization of set algebra and the algebra of propositions and is a tool for studying and applying logic.

It provides mathematical basis for expressing logic circuit functions, as well as analyzing and designing of the digital system.

### **Boolean Axioms**

The Boolean axioms are **the laws of Boolean algebra for addition, multiplication and for the inversion**.

They are known as axioms because they are the truth which can be verified for different possibilities but cannot be proved.

There are different operators for Boolean algebra.

### **Boolean Operators:**

The operators in Boolean algebra are slightly different than conventional algebra for the basic gates and the following operators are commonly used.

- (i) **Dot sign ( $\cdot$ )**: The dot sign ( $\cdot$ ) which is also expressed as ( $\times$ ), indicates logical product of two terms.

The **logical product of two terms** A and B is expressed as  $A \cdot B$  (or  $A \times B$ ) and is read as "**A AND B**". The inputs A and B are said to be ANDed.

(ii) **Plus sign (+)**: The (+) sign indicates the **logical sum of two terms**.

For example,  $A + B$  represents logical sum of terms A and B and is read as "**A OR B**". The inputs A and B are said to be ORed.

(iii) **Overbar ( $\bar{\phantom{A}}$ )**: A sign of ( $\bar{\phantom{A}}$ ) indicates that the terms which are overbar, are to be complemented.

For example,  $\bar{A}$  represents complementation of term A and is read as "**NOT A**".

### **Boolean Addition and Multiplication:**

Addition in Boolean algebra involves variables having values of either a binary 1 or a binary 0.

Binary 1 will represent a HIGH level and binary 0 will represent a LOW level in Boolean equations.

The basic rules for Boolean addition are as follows :

$$0 + 0 = 0$$

$$0 + 1 = 1$$

$$1 + 0 = 1$$

$$1 + 1 = 1$$

**Boolean addition is the same as the OR.** Notice that, it differs from binary addition in the case where two 1s are added.

Multiplication in Boolean algebra follows the same basic rules governing binary multiplication.

$$0 \cdot 0 = 0$$

$$0 \cdot 1 = 0$$

$$1 \cdot 0 = 0$$

$$1 \cdot 1 = 1$$

**Boolean multiplication is the same as the AND.**

Complementation in Boolean algebra is as follows :

$$\overline{0} = 1$$

$$\overline{1} = 0$$

It is same as NOT.

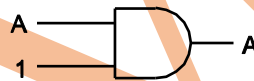
## **2. Boolean Algebra Rules and Laws**

There are certain rules and laws in Boolean algebra.

The alphabets A, B, C and D can be used as variables having values 0 and 1.

### **Laws of Intersection :**

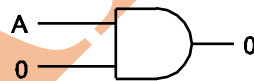
**Law 1 :  $A \cdot 1 = A$**



**Fig. 2.1**

If a logic 1 is applied to one of the two inputs of the AND gate and signal A to the other input, the output will be A.

**Law 2 :  $A \cdot 0 = 0$**

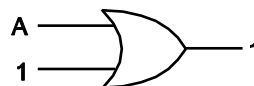


**Fig. 2.2**

If a logic 0 is applied to one of the two inputs of the AND gate and signal A to the other input, the output will be logic 0.

### **Laws of Union :**

**Law 3 :  $A + 1 = 1$**



**Fig. 2.3**

If a logic 1 is applied to the two inputs of OR gate and signal A to the other input, the output will be logic 1.

**Law 4 :  $A + 0 = A$**

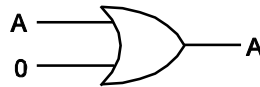


Fig. 2.4

If a logic 0 is applied to one of the two inputs of OR gate and A to the other input, the output will be logic A.

### Laws of Tautology:

**Law 5 :  $A \cdot A = A$**

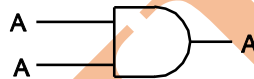


Fig. 2.5

If the same signal A is applied to all the inputs of AND gate, the output will be same as the input.

**Law 6 :  $A + A = A$**



Fig. 2.6

If the same signal A is applied to all the inputs of OR gate, the output will be same as input A.

### Laws of Complement :

**Law 7 :  $A \cdot \bar{A} = 0$**



Fig. 2.7

If a logic signal A and its complement  $\bar{A}$  is applied to an AND gate, the output will be logic 0.

**Law 8 :  $A + \bar{A} = 1$**



Fig. 2.8

If a logic signal A and its complement  $\bar{A}$  is applied to an OR gate, the output will be 1.

### Law of Double negation :

**Law 9 :**  $\overline{\overline{A}} = A$

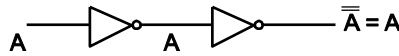


Fig. 2.9

The complement of the complement of A is A.

$\overline{\overline{A}} = A$   
 $\overline{A=0} = 1$      $\overline{A=1} = 0 \Rightarrow \overline{\overline{A}} = A$

### Laws of Commutation :

**Law 10 :**  $A \cdot B = B \cdot A$

The commutative law of multiplication states that order in which variables are ANDed makes no difference at the output.



Fig. 2.10

**Law 11 :**  $A + B = B + A$

This law states that 'the order in which the inputs are given to an OR gate makes no difference at the output'.



Fig. 2.11

### Laws of Association :

**Law 12 :** The associative law of addition for three variables is stated as follows,

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

This law states that 'in the ORing of several variables, the result is same regardless of grouping of variables'.

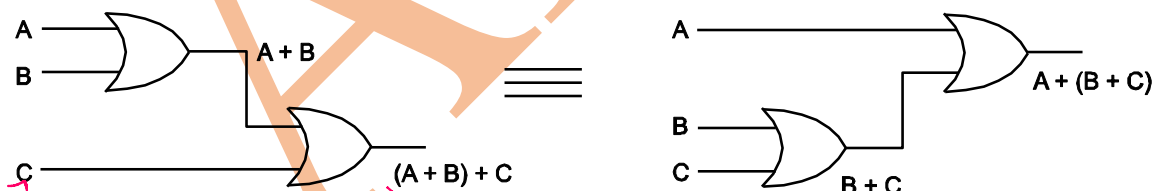


Fig. 2.12



**Law 13 :** The associative law of multiplication is stated as follows for three variables.

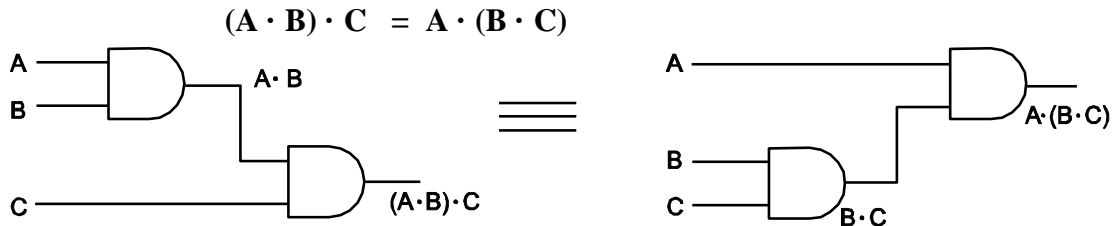


Fig. 2.13

### Laws of Distribution :

**Law 14 :** The distributive law for three variables is written as follows :

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

This law states that 'ORing several variables and ANDing the result with a single variable is equivalent to ANDing the single variable with each of several variables and then ORing the products'.

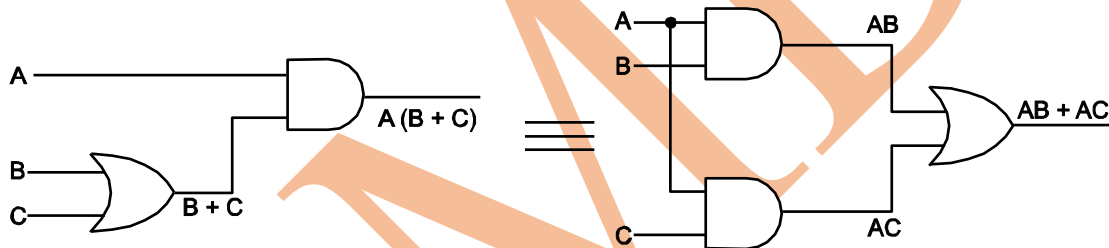


Fig. 2.14

**Law 15 :**  $A + (B \cdot C) = (A + B) \cdot (A + C)$

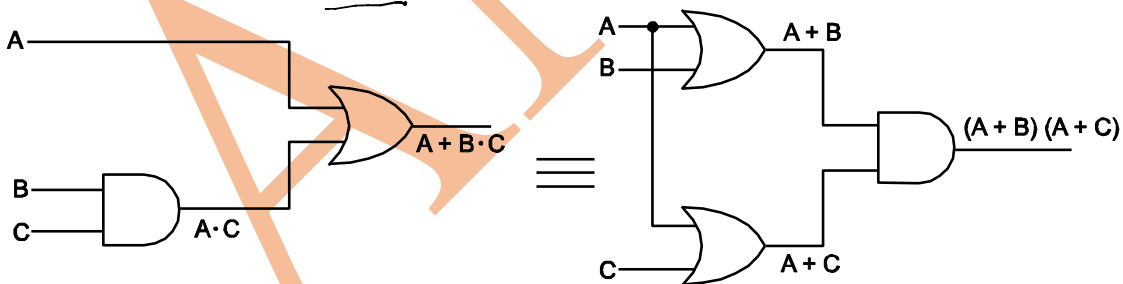


Fig. 2.15

**Proof :**

$$\begin{aligned} (A + B) \cdot (A + C) &= AA + AB + AC + BC \\ &= A(1 + B + C) + BC \end{aligned} \quad \dots (1)$$

Though B and C have any value either 0 or 1,  
 $1 + B + C = 1$

Therefore equation (1) becomes  $= A \cdot 1 + BC$

From law (1),  $A \cdot 1 = A$

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

## Laws of Absorption :

Law 16 :

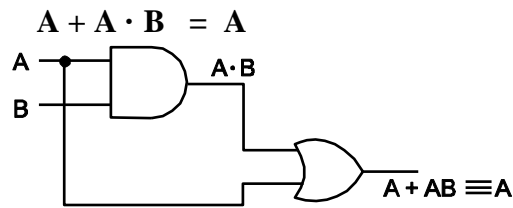


Fig. 2.16

**Proof :** Consider

$$\begin{aligned} A + A \cdot B &= A \cdot (1 + B) \\ &= A \cdot 1 \\ &= A \end{aligned} \quad \begin{aligned} \because 1 + B &= 1 \\ \because A \cdot 1 &= A \end{aligned}$$

Law 17 :

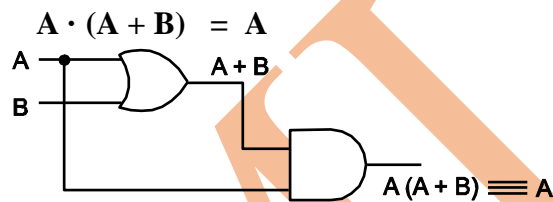


Fig. 2.17

**Proof :**

$$\begin{aligned} A \cdot (A + B) &= A \cdot A + A \cdot B \\ &= A + AB \\ &= A(1 + B) \\ &= A \cdot 1 \\ &= A \end{aligned} \quad \begin{aligned} \because A \cdot A &= A \\ \because 1 + B &= 1 \\ \because A \cdot 1 &= A \end{aligned}$$

Law 18 :

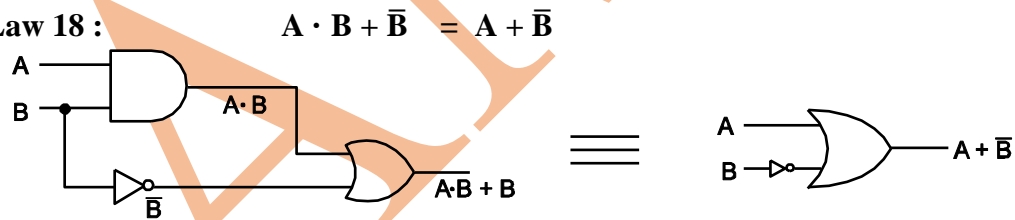


Fig. 2.18

**Proof :**

$$\begin{aligned} A \cdot B + \bar{B} &= A \cdot B + \bar{B} \cdot 1 \\ &= A \cdot B + \bar{B} \cdot (A + 1) \\ &= A \cdot B + \bar{B}A + \bar{B} \cdot 1 \\ &= A \cdot (B + \bar{B}) + \bar{B} \\ &= \underline{A + \bar{B}} \end{aligned} \quad \begin{aligned} \because \bar{B} \cdot 1 &= \bar{B} \\ \because A + 1 &= 1 \\ \because \bar{B} \cdot 1 &= \bar{B} \\ \because B + \bar{B} &= 1 \end{aligned}$$

**Law 19 :**  $A \cdot (\bar{A} + B) = A \cdot B$

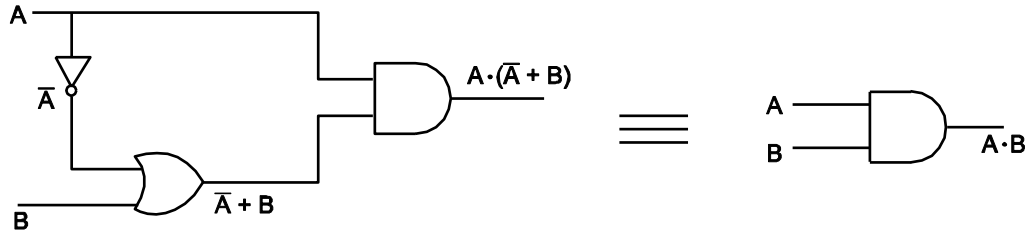


Fig. 2.19

**Proof :**

$$\begin{aligned}
 A \cdot (\bar{A} + B) &= A\bar{A} + AB \\
 &= 0 + AB && \because A \cdot \bar{A} = 0 \\
 &= A \cdot B
 \end{aligned}$$

**Law 20 :**  $A \cdot \bar{B} + B = A + B$

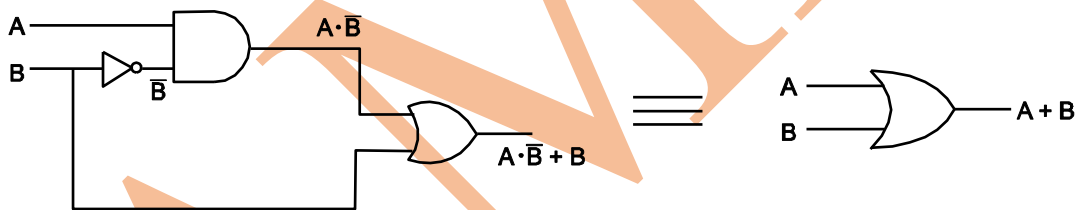


Fig. 2.20

**Proof :** Consider

$$\begin{aligned}
 A \cdot \bar{B} + B &= A \cdot \bar{B} + B \cdot 1 && \because B \cdot 1 = B \\
 &= A \cdot \bar{B} + B \cdot (A + 1) && \because A + 1 = 1 \\
 &= A \cdot \bar{B} + B \cdot A + B \cdot 1 \\
 &= A(\bar{B} + B) + B && \because \bar{B} + B = 1 \\
 &= A \cdot 1 + B && \because A \cdot 1 = A \\
 &= A + B
 \end{aligned}$$

**Table 2. : Boolean algebraic theorems**

<b>Law</b>	<b>Classification</b>	<b>Algebraic definition</b>
1	Laws of intersection	$A \cdot 1 = A$
2		$A \cdot 0 = 0$
3	Laws of union	$A + 1 = 1$
4		$A + 0 = A$
5	Laws of tautology	$A \cdot A = A$
6		$A + A = A$
7	Laws of complement	$A \cdot \bar{A} = 0$
8		$A + \bar{A} = 1$
9	Law of double negation	$\bar{\bar{A}} = A$
10	Laws of commutation	$A \cdot B = B \cdot A$
11		$A + B = B + A$
12	Laws of association	$(A + B) + C = A + (B + C)$
13		$(A \cdot B) \cdot C = A \cdot (B \cdot C)$
14	Laws of distribution	$A(B + C) = A \cdot B + A \cdot C$
15		$A + (B \cdot C) = (A + B) \cdot (A + C)$
16	Laws of absorption	$A + A \cdot B = A$
17		$A \cdot (A + B) = A$
18		$A \cdot B + \bar{B} = A + \bar{B}$
19		$A \cdot (\bar{A} + B) = AB$
20		$A \cdot \bar{B} + B = A + B$

## De Morgan's Theorems

De Morgan, a logician and mathematician proposed two theorems which are important parts of Boolean algebra.

De Morgan's first theorem is,

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

The complement of product is equal to the sum of the complements. The complement of two or more variables ANDed is the same as the OR of the complement of each individual variables.

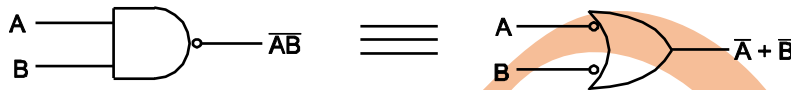


Fig. 1

It can also be constructed with AND, OR and NOT gate as follows :

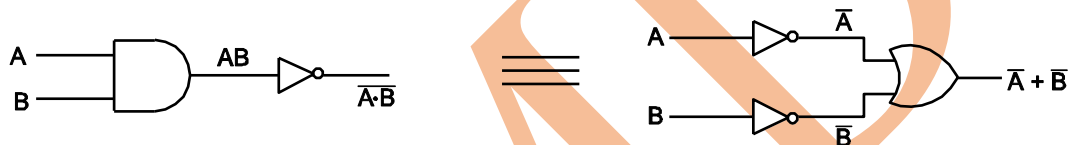


Fig. 2

Table 3 : De Morgan's first theorem

Input		Intermediate value			Output	
A	B	AB	$\overline{A}$	$\overline{B}$	$\overline{A \cdot B}$	$\overline{A} + \overline{B}$
0	0	0	1	1	1	1
0	1	0	1	0	1	1
1	0	0	0	1	1	1
1	1	1	0	0	0	0

De Morgan's Second Theorem

The complement of a sum is equal to the product of the complements. It can be expressed as

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

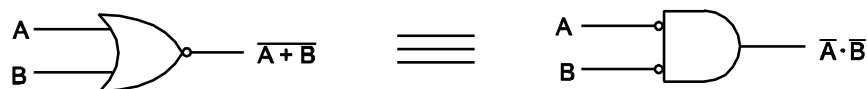


Fig. 3

Using AND, OR and NOT gate, the circuit can be drawn as

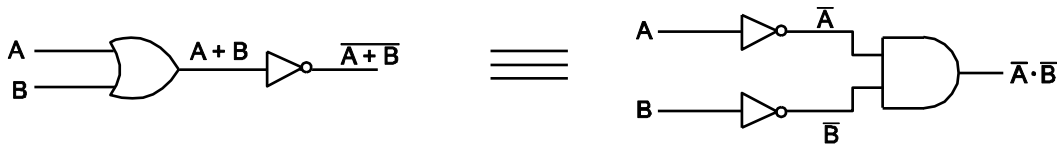


Fig. 4

Table 4 : De Morgan's second theorem

Input		Intermediate value			Output	
A	B	A + B	$\bar{A}$	$\bar{B}$	$\overline{A + B}$	$\bar{A} \cdot \bar{B}$
0	0	0	1	1	1	1
0	1	1	1	0	0	0
1	0	1	0	1	0	0
1	1	1	0	0	0	0

Now consider the NAND operation of three variables.

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

and

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

The above results can be easily extended to any number of variables.

### Applications of De Morgan's Theorems

De Morgan's theorems can be used for simplifying logic function.

**Example 1 :** Simplify the following expressions.

**Solution :** (i) 
$$\begin{aligned} \overline{\overline{A + B + C}} &= \overline{\overline{A + B} \cdot \overline{C}} \quad [\because \overline{\overline{A + B}} = A + B \text{ and } \overline{\overline{C}} = C] \\ &= A + B \cdot C \\ &= (A + B) \cdot C \end{aligned}$$

(ii) 
$$\overline{\overline{A + B + CD}} = \overline{\overline{A + B} \cdot \overline{CD}} = (\bar{A} + \bar{B}) CD$$

(iii) 
$$\begin{aligned} \overline{\overline{AB + \bar{A} + AB}} &= \overline{\overline{A + B + \bar{A} + AB}} \\ &= \overline{\overline{A + \bar{A} + B}} && \because \bar{A} + A = \bar{A} \\ &= \overline{\overline{A + A + B}} && \because AB + \bar{B} = A + \bar{B} \\ &= \overline{1 + B} = \bar{1} = 0 \end{aligned}$$

$$\overline{AB + A + AB} = 0$$

$$\begin{aligned} \text{(iv)} \quad \overline{(\overline{A \cdot B})(B \cdot C)(\overline{CD})} &= \overline{\overline{A \cdot B} + \overline{B \cdot C} + \overline{(\overline{CD})}} = \overline{\overline{A} + \overline{B} + \overline{B} + \overline{C} + \overline{C} + \overline{D}} \\ &= \overline{A + \overline{B} + \overline{B} + \overline{C} + \overline{C} + D} \quad \because \overline{\overline{A}} = A \text{ and } \overline{\overline{D}} = D \\ &= \overline{A + \overline{B} + \overline{C} + D} \quad \because \overline{\overline{B} + \overline{B}} = \overline{B} \text{ and } \overline{\overline{C} + \overline{C}} = \overline{C} \end{aligned}$$

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

$$\text{(v)} \quad y = \overline{\overline{A \cdot B} + \overline{A \cdot B}}$$

Using De Morgan's theorem

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

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

### Simplification of Logic Equations using Laws of Boolean Algebra

Many times it is essential to reduce number of gates required for designing a digital circuit, reduce a particular expression to its simplest form, ultimately which reduces size and price (cost) of circuit.

By applying basic laws, rules and theorems of Boolean algebra, it is possible to implement practically.

Following examples illustrate the technique.

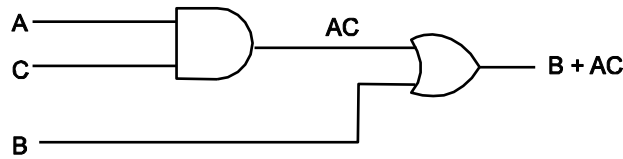
**Example 1 :** Simplify the expression  $AB + A(B + C) + B(B + C)$  using Boolean algebra techniques.

**Solution :** Consider

$$\begin{aligned} AB + A(B + C) + B(B + C) &= AB + AB + AC + BB + BC \\ &= A(B + B) + AC + B(B + C) \quad \because B \cdot B = B \\ &= AB + AC + B \quad \because B + B = B, B(B + C) = B \\ &= AB + B + AC \\ &= B(A + 1) + AC \\ &= B \cdot 1 + AC \quad \because A + 1 = 1 \\ &= B + AC \end{aligned}$$

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

The logic circuit will be



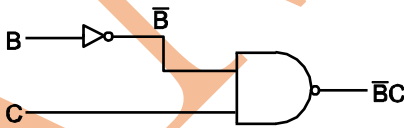
**Example 2 :** Simplify the expression  $[A\bar{B} (C + BD) + \bar{A} \bar{B}] C$  using Boolean algebra techniques.

**Solution :** Consider

$$\begin{aligned}
 [A\bar{B} (C + BD) + \bar{A} \bar{B}] C &= (A\bar{B}C + A\bar{B}BD + \bar{A} \bar{B}) C \\
 &= (A\bar{B}C + A \cdot 0 \cdot D + \bar{A} \bar{B}) C && \because \bar{B} B = 0 \\
 &= (A\bar{B}C + 0 + \bar{A} \bar{B}) C \\
 &= (A\bar{B}C + \bar{A} \bar{B}) C \\
 &= A\bar{B}CC + \bar{A}\bar{B}C \\
 &= A\bar{B}C + \bar{A}\bar{B}C && \because C \cdot C = C \\
 &= \bar{B}C (A + \bar{A}) \\
 &= \bar{B}C \cdot (1) && \because A + \bar{A} = 1 \\
 &= \bar{B}C
 \end{aligned}$$

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

The logic circuit will be



**Example 3 :** Using Boolean algebra techniques, simplify the following expressions as much as possible : (i)  $A(A + B)$ ,

(ii)  $A(\bar{A} + AB)$ ,

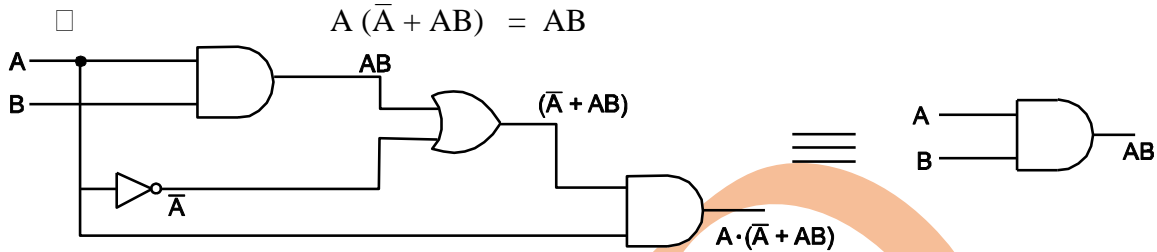
(iii)  $BC + \bar{B}C$ ,

(iv)  $A(A + \bar{A}B)$ .

**Solution :** (i)  $A(A + B) = A \cdot A + A \cdot B$   
 $= A + A \cdot B$   $\because A \cdot A = A$   
 $= A(1 + B)$   
 $= A \cdot (1)$   $\because 1 + B = 1$   
 $= A$   
 $A(A + B) = A$



$$\begin{aligned}
 \text{(ii)} \quad A(\bar{A} + AB) &= A\bar{A} + A \cdot AB \\
 &= 0 + A \cdot AB && \because A\bar{A} = 0 \\
 &= AB && \because A \cdot A = A
 \end{aligned}$$



$$\begin{aligned}
 \text{(iii)} \quad BC + \bar{B}C &= C(B + \bar{B}) \\
 &= C \cdot (1) && \because B + \bar{B} = 1 \\
 &= C
 \end{aligned}$$

$$\begin{aligned}
 \text{(iv)} \quad A(A + \bar{A}B) &= A \cdot A + A \cdot \bar{A}B \\
 &= A + A \cdot \bar{A}B && \because A \cdot A = A \\
 &= A + 0 && \because A \cdot \bar{A} = 0 \\
 &= A && \because A + 0 = A
 \end{aligned}$$

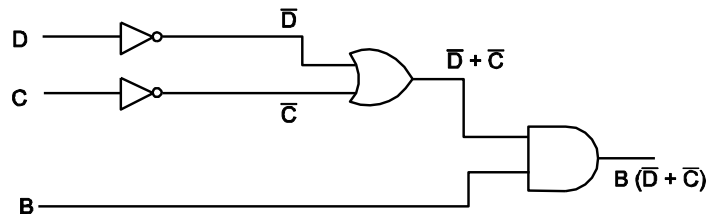
**Example 4 :** Reduce the following Boolean expression and draw the logic diagram :

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

**Solution :** Consider

$$\begin{aligned}
 \bar{A}BC\bar{D} + BC\bar{D} + B\bar{C}\bar{D} + B\bar{C}D &= BC\bar{D}(\bar{A} + 1) + B\bar{C}(\bar{D} + D) \\
 &&& (\because \bar{A} + 1 = 1, D + \bar{D} = 1) \\
 &= BC\bar{D} \cdot (1) + B\bar{C} \cdot (1) = BC\bar{D} + B\bar{C} = B(\bar{C}\bar{D} + \bar{C}) \\
 &= B(\bar{D}\bar{C} + \bar{C}) && (\because \bar{D}\bar{C} = \bar{C}\bar{D}) \\
 &= B(\bar{D} + \bar{C}) && (\because \bar{D}\bar{C} + \bar{C} = \bar{D} + \bar{C})
 \end{aligned}$$

The logic circuit will be



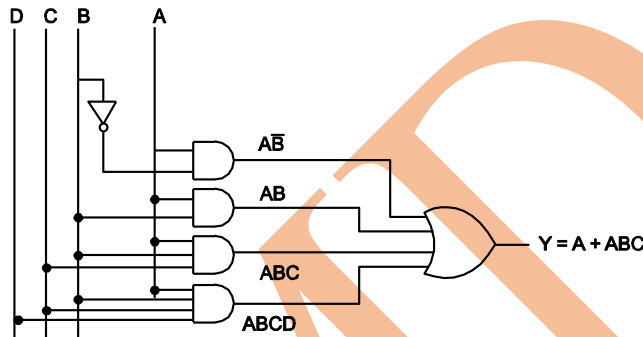
**Example 5 :** Simplify the following equations using laws of Boolean algebra :

(i)  $Y = ABCD + ABC + AB + A\bar{B}$

**Solution :** Consider

$$\begin{aligned} ABCD + ABC + AB + A\bar{B} &= ABC(D + 1) + A(B + \bar{B}) \\ &= ABC \cdot (1) + A \cdot (1) \quad (\because D + 1 = 1, B + \bar{B} = 1) \\ &= ABC + A \end{aligned}$$

Circuit for  $Y = ABCD + ABC + AB + A\bar{B}$  can be constructed as follows :

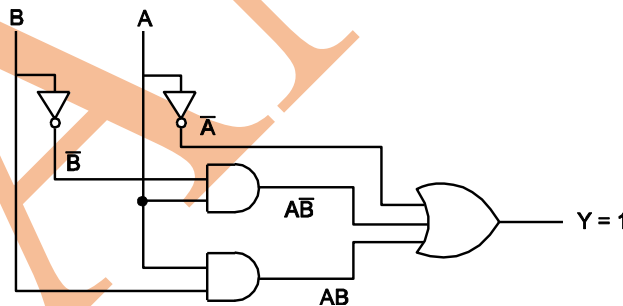


Here four input AND and four input OR gate are used.

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

**Solution :** Consider

$$\begin{aligned} \bar{A} + AB + A\bar{B} &= \bar{A} + A(B + \bar{B}) \\ &= \bar{A} + A(1) = \bar{A} + A = 1 \quad (\because B + \bar{B} = 1) \end{aligned}$$



Here three input OR gate is used.

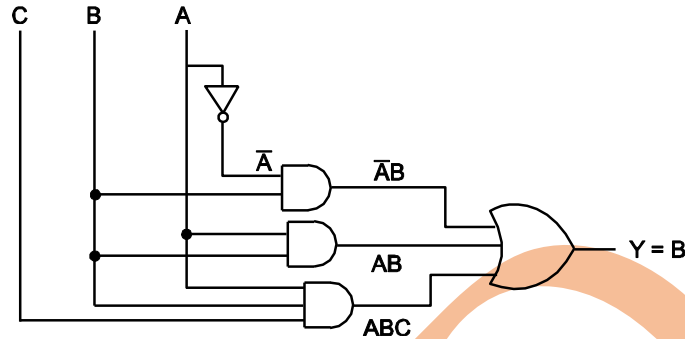
(iii)  $Y = AB + \bar{A}B + ABC$

**Solution :**

$$\begin{aligned} Y &= AB + ABC + \bar{A}B \\ &= AB(1 + C) + \bar{A}B \\ &= AB(1) + \bar{A}B = AB + \bar{A}B \quad (\because 1 + C = 1) \\ &= B(A + \bar{A}) \end{aligned}$$

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

Let us consider the following circuit to solve the given equation.

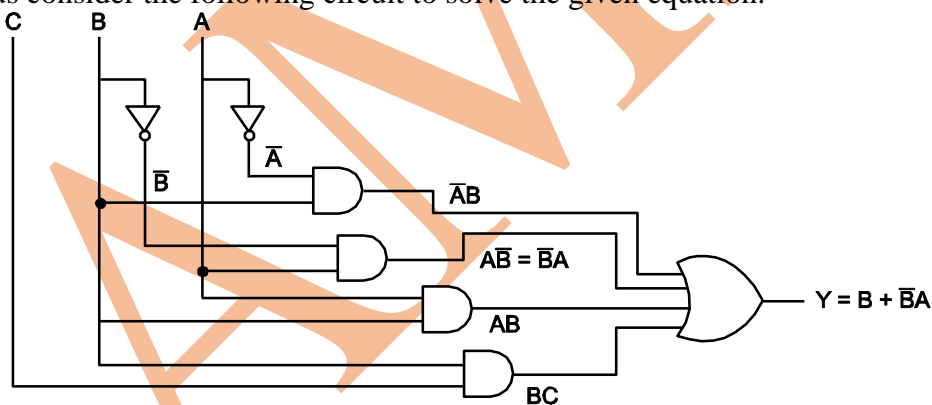


(iv)  $A = AB + BC + \bar{B}A + \bar{A}B$

**Solution :**

$$\begin{aligned} Y &= AB + \bar{A}B + BC + \bar{B}A = B(A + \bar{A}) + BC + \bar{B}A \\ &= B + BC + \bar{B}A \quad (\because A + \bar{A} = 1) \\ &= B(1 + C) + \bar{B}A \\ &= B + \bar{B}A \quad (\because 1 + C = 1) \end{aligned}$$

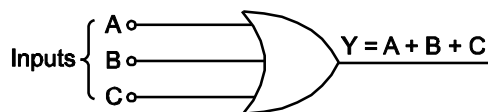
Let us consider the following circuit to solve the given equation.



### Boolean Expression in SOP and POS Form

Boolean expressions can be used to build the logic circuit.

If we have the expression  $Y = A + B + C$  and asked to build a circuit that perform this logic function, we can very easily see that there must be an OR gate having three inputs A, B and C. The circuit can be realized using three input OR gate as shown in Fig. below.



**Fig.:** Logic diagram for Boolean expression  $Y = A + B + C$

It is found that the Boolean expression come in two forms.

All Boolean expressions can be converted into either of two standard forms; the sum of product form or the product of sum forms.

It is found that this standardization makes the evaluation, simplification and implementation of Boolean expressions much more systematic and easier.

### Sum of Product (SOP) Form :

A product term consist of the product of multiplication of variables or their complements (known as *literal*).

When two or more product terms are summed by Boolean addition, the resulting expression is known as sum of product (SOP) form. The examples of SOP are,

$$Y = AC + BC$$

$$Y = AB + ABC$$

$$Y = ABC + BCD + AB\bar{D} \text{ etc.}$$

The sum of product form is called *minterm form* in engineering texts and the product of sum form is called the *maximum term form* by engineers, technicians and scientist.

An SOP expression can contain a single variable term. In an SOP expression a single overbar cannot extend over more than one variable, however more than one variable in a term can have an overbar. Thus,  $\overline{ABC}$  is not valid but  $\overline{A} \overline{B} \overline{C}$  is valid in SOP.

The set of variables contained in the expression in either complemented or uncomplemented form is known as domain of a Boolean expression. For example, the domain of the expression  $\overline{A}B + ABC$  is the set of variable A, B and C.

### Implementation of an SOP Expression :

The sum of product (SOP) expression contains product of sum terms.

A product term is produced by an AND operation and the sum or addition of two or more product terms is produced by an OR operation.

Therefore, an SOP expression is implemented by an AND-OR logic.

Consider the SOP expression  $y = AB + BC + CD$ , it can be implemented as shown in Fig. below:

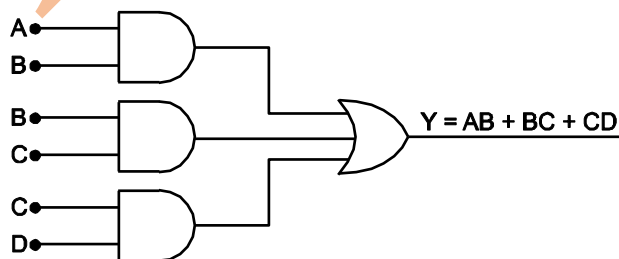


Fig.: SOP of Boolean expression  $Y = AB + BC + CD$

### The Product of Sum (POS) Form :

A sum term consist of the sum i.e. Boolean addition of literals (variables or their complements). When two or more sum terms are multiplied, the resulting expression is known as product of sum (POS) form, for example,

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

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

POS expression can contain a single variable term. In a POS expression, a single overbar cannot extend over more than one variable, however more than one variable in a term can have an overbar. For example, POS expression can have the term  $\bar{A} + \bar{B} + \bar{C}$  but will not have  $\overline{A + B + C}$ .

### Implementation of a POS Expression :

Given Boolean expression, of the POS form can be implemented by ANDing the outputs of two or more OR gates.

A sum term is produced by an OR operation and the product of two or more sum terms is produced by an AND operation.

The expression  $Y = (A + B) (B + C) (C + D)$  can be implemented as shown in Fig. POS form.

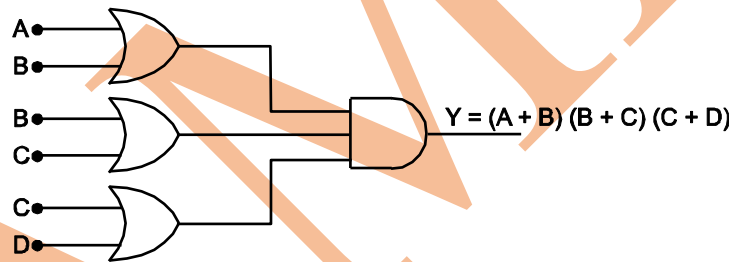


Fig.: POS form of  $Y = (A + B) (B + C) (C + D)$

**Standard or Canonical SOP and POS forms :** A logic expression is said to be in the standard or canonical form.

If each SOP term consists of all the literals/variables in their complemented or uncomplemented form, the standard SOP form is

$$Y = \underbrace{ABC}_{\uparrow} + \underbrace{A\bar{B}C}_{\uparrow} + \underbrace{AB\bar{C}}_{\uparrow}$$

Each product term  
contain all the three variables

The Boolean expression is said to be in standard POS form, if all the terms in POS consist of all the literals in their complemented or uncomplemented form. For example,

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

Each sub term contain all the literals

Each individual term in the standard SOP form is called *minterm*.

e.g. Consider,

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

Each individual terms called minterm

Each individual term in the standard POS form is called *maxterm*.

e.g. Consider,

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

Each individual terms called maxterm

#### Conversion of SOP/POS Expression to its Standard SOP/POS Form

The given Boolean expression can be converted to their corresponding SOP and POS forms. As seen in the standard SOP form each product term consist of all the literals.

e.g.  $Y = AB + A\bar{B} + \bar{A}\bar{B}$  is standard SOP

but  $Y = AB + ABC + ABC\bar{C}$  is not standard SOP

The conversion of expression into standard SOP form is a three step process :

- (1) For each term find the missing literal,
- (2) Then AND the term with the term formed by ORing the missing literal and its complement,
- (3) Simplify the obtained equation.

e.g. Convert the expression

$$Y = AB + A\bar{C} + BC \text{ in the standard SOP form.}$$

**Solution :** Given expression is

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

**Step 1 :** Find the missing literal in each term.

$$Y = \underbrace{AB}_{\substack{\text{Missing} \\ \text{literal} \\ \text{is } C}} + \underbrace{A\bar{C}}_{\substack{\text{Missing} \\ \text{literal} \\ \text{is } B}} + \underbrace{BC}_{\substack{\text{Missing} \\ \text{literal} \\ \text{is } A}}$$

**Step 2 :** AND each term with its (Missing literal + Its complement).

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

Note that  $C + \bar{C} = 1$  then it does not change the value of expression.

**Step 3 :** Simplification of the expression.

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

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

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

Since

$$A + \bar{A} = 1$$

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

Since in the above expression each term contains all the literals it is in the standard SOP form.

Conversion to Standard POS Form

In the standard POS form each sum term consists of all the literals in the complemented or uncomplemented form e.g.

$$Y = (\bar{A} + B) \cdot (\bar{A} + \bar{B}) \cdot (A + \bar{B}) \text{ is in the standard POS}$$

whereas,  $Y = (\bar{A} + \bar{B}) \cdot (A + B + C)$  is in the non-standard POS form.

The given POS expression can be converted into standard POS form using three steps.

**Step 1 :** For each term find the missing literal.

**Step 2 :** Then OR each term with the term formed by ANDing the missing literal in that term with its complement.

**Step 3 :** Simplify the expression to obtain standard POS form.

e.g. Convert the expression

$$Y = (A + B) \cdot (A + \bar{B}) \cdot (B + \bar{C}) \text{ into the standard POS forms.}$$

**Solution :** Find the missing literal in each term

$$Y = \underbrace{(A + B)}_{\substack{\text{Missing} \\ \text{literal} \\ \text{is } C}} \cdot \underbrace{(A + \bar{B})}_{\substack{\text{Missing} \\ \text{literal} \\ \text{is } C}} \cdot \underbrace{(B + \bar{C})}_{\substack{\text{Missing} \\ \text{literal} \\ \text{is } A}}$$

**Step 1 :** OR each term with missing literal and its complement.

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

**Step 2 :** Simplify the expression

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

Let  $p = A + B, \quad q = A + \bar{B}, \quad r = B + \bar{C},$

$\therefore Y = (p + C\bar{C}) \cdot (q + C\bar{C}) \cdot (r + A\bar{A}) \quad \dots (2)$

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

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

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

Since each term of above expression contains all the literals so the equation is standard POS form.

$\therefore y = (\square \bar{A} = A)$

### **Karnaugh Map**

The Karnaugh map or K-map provides a systematic method for simplifying a Boolean expression and can be used as visual display of fundamental products needed for a sum of products solution.

The K-map is composed of an arrangement of adjacent 'cells' each representing one particular combination of variables in product form.

The K-map consists of **2<sup>n</sup> cells, where n is number of variables.**

For example, there are four combinations of the products of two variables A and B and their complements  $\bar{A}\bar{B}, \bar{A}B, A\bar{B}$  and  $AB$ . Therefore, the K-map must have four cells, with each cell representing one of the variable combination.

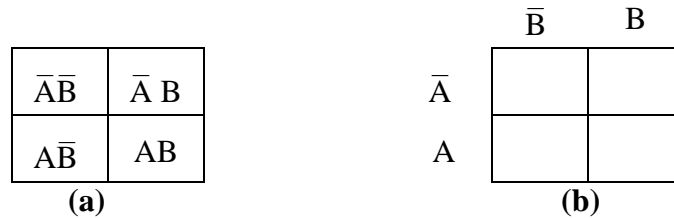
### **Advantages of K-map**

(i) As we have seen the laws of Boolean algebra, but it is difficult to apply these laws when number of variables are large. In such a case, it makes easy by using K-map.

(ii) Use of large number of laws of Boolean algebra increases the chances of error because one have to remember all these laws and has to apply them at correct place. But by using K-map it is not necessary to remember all the laws, according to circuit situation K-map may apply. It provides easiest way to produce simplest Boolean expression with minimum chances of error.



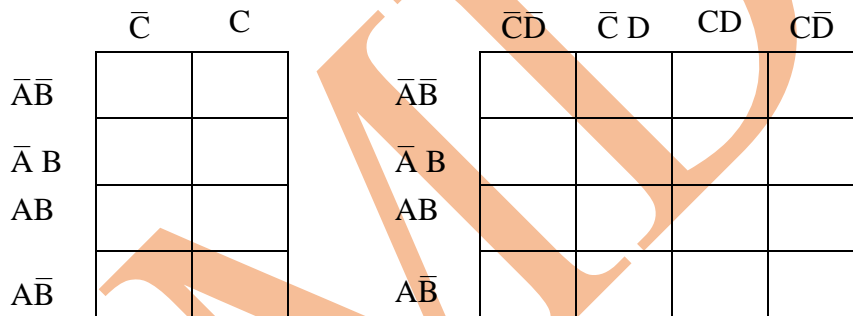
Format of a **two variable K-map** can be represented as follows:



**Fig.**

Variable combination is shown in Fig. (a) and actually how K-map can be arranged with variables outside the cell is shown in Fig. (b).

Extensions of the K-map to **three and four variables** are shown in Fig..



**(a) Three variable map ( $2^3 = 8$  cells)**      **(b) Four variable map ( $2^4 = 16$  cells)**

**Fig.**

Karnaugh maps can be used for **five, six or more variables**.

We see how K-map can be drawn from the following Table

Consider the Table:.

**Table 2.4**

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

Output Y is taken by random selection of 1. First we draw the blank map.

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

**Fig.**

The vertical column is labelled as  $\bar{A}\bar{B}$ ,  $\bar{A}B$ ,  $AB$  and  $A\bar{B}$ .

Output  $Y = 1$  appears for 010, 110 and 111. The fundamental products for these input conditions are  $\bar{A}B\bar{C}$ ,  $AB\bar{C}$ ,  $ABC$  (bar on variables where it is 0).

Now, enter 1s for these products in K-map.

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

**Fig.**

Enter 0s in the remaining spaces. Then final K-map becomes as in Fig..

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

**Fig.**

### Pairs, Quads and Octets

**(a) Pair :** The K map shown in Fig. 2.46 contains a pair of 1s that are horizontally adjacent (next to each other).

	$\bar{C}\bar{D}$	$\bar{C}D$	$CD$	$C\bar{D}$
$\bar{A}\bar{B}$	0	0	0	0
$\bar{A}B$	1	1	0	0
$AB$	0	0	0	0
$A\bar{B}$	0	0	0	0

(a)

	$\bar{C}\bar{D}$	$\bar{C}D$	$CD$	$C\bar{D}$
$\bar{A}\bar{B}$	0	0	0	0
$\bar{A}B$	1	1	0	0
$AB$	0	0	0	0
$A\bar{B}$	0	0	0	0

(b)

**Fig.: Horizontally adjacent ones**

The first 1 represents the product of  $\bar{A} B \bar{C} \bar{D}$  and second 1 the product of  $\bar{A} B C \bar{D}$ . As we move from the first 1 to second 1, only one variable goes from complemented to uncomplemented ( $\bar{D}$  to  $D$ ), the other variables do not change the form ( $\bar{A} B \bar{C}$  remains unchanged). In such case, we can eliminate the variable that changes the form.

The sum of product (SOP) in which each individual term called as minterm represents

$$\begin{aligned}
 Y &= \bar{A} B \bar{C} \bar{D} + \bar{A} B C \bar{D} \\
 &= \bar{A} B \bar{C} (\bar{D} + D) && \because \bar{D} + D = D + \bar{D} \\
 &= \bar{A} B \bar{C} (D + \bar{D}) \\
 &= \bar{A} B \bar{C} && \because D + \bar{D} = 1
 \end{aligned}$$

$$\square \quad Y = \bar{A} B \bar{C}$$

Adjacent 1s as shown in Fig. (a) complements are dropped out. For easy identification we will encircle a pair of adjacent 1s as shown in Fig. (b).

For the pair of horizontally or vertically adjacent 1s, we can eliminate the variable that appears in both complemented and uncomplemented form.

Examples of pairs for 3 variables

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

**Fig. (c)**

B goes from complemented to uncomplemented form ( $\bar{B}$  to  $B$ ).

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

$$= \bar{A}\bar{C}$$

If more than one pair exist on a Karnaugh map, we can OR the simplified products to get the Boolean equation.

For example,

	$\bar{C}\bar{D}$	$\bar{C}D$	$CD$	$C\bar{D}$
$\bar{A}\bar{B}$	1	1	0	0
$\bar{A}B$	0	0	0	0
$AB$	0	0	0	1
$A\bar{B}$	0	0	0	1

Fig. (d)

For upper horizontal pair,

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

□

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

and for the lower vertical pair.

$$Y = ABC\bar{D} + AB\bar{C}\bar{D}$$

□

$$Y = AC\bar{D}$$

The corresponding Boolean equation for this map is

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

**(b) Quad:** A quad is a group of four 1s that are horizontally or vertically adjacent. The 1s may be end to end or in the form of square.

This group can be formed by combining top row, bottom row, left column, right column, just like in pairing. Infact, quad eliminates two variables and their complements.

Consider K-map for three variables.

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

Fig. 2.47

Looking in K-map from Fig. 2.47, we find except for  $\bar{C}$ , other variables AB are changed from complement to uncomplement form and/or vice versa. Therefore, output Y becomes from Boolean algebra,

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

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

$$Y = \bar{A}\bar{C}(B + \bar{B}) + A\bar{C}(B + \bar{B}) \quad (\because B + \bar{B} = 1)$$

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

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

$$Y = \bar{C} \quad (\because \bar{A} + A = 1)$$

The other combinations of quads are

(a)

	$\bar{C}\bar{D}$	$\bar{C}D$	$CD$	$C\bar{D}$
$\bar{A}\bar{B}$	0	0	0	0
$\bar{A}B$	1	1	1	1
$AB$	0	0	0	0
$A\bar{B}$	0	0	0	0

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

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

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

$$(\because D + \bar{D} = 1)$$

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

$$(\because \bar{C} + C = 1)$$

$$= \bar{A}B$$

(b)

	$\bar{C}\bar{D}$	$\bar{C}D$	$CD$	$C\bar{D}$
$\bar{A}\bar{B}$	0	0	0	0
$\bar{A}B$	0	0	1	1
$AB$	1	1	0	0
$A\bar{B}$	1	1	0	0

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

$$\begin{aligned}
 &= ABC\bar{D} + A\bar{B}C(\bar{D} + D) \\
 &= ABC\bar{D} + A\bar{B}C \\
 &= A\bar{C}(B + \bar{B}) \quad (\because B + \bar{B} = 1) \\
 &= A\bar{C}
 \end{aligned}$$

(c)

	$\bar{C}\bar{D}$	$\bar{C}D$	$CD$	$C\bar{D}$
$\bar{A}\bar{B}$	0	0	0	0
$\bar{A}B$	1	0	0	1
$AB$	1	0	0	1
$A\bar{B}$	0	0	0	0

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

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

$$Y = \bar{A}B\bar{D} + AB\bar{D} \quad (\because \bar{C} + C = 1)$$

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

$$Y = B\bar{D}$$

**(c) Octet :** This is a group of eight adjacent 1s. An octet like this eliminates three variables and their complements. Octet can be considered as a pair of quads.

(i) Consider the K-map shown in Fig. 2.51.

	$\bar{C}\bar{D}$	$\bar{C}D$	$CD$	$C\bar{D}$
$\bar{A}\bar{B}$	0	0	0	0
$\bar{A}B$	1	1	1	1
$AB$	1	1	1	1
$A\bar{B}$	0	0	0	0

The Boolean expression will be

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

$$+ AB\bar{C}\bar{D} + AB\bar{C}D + ABC\bar{D} + ABCD$$

$$\square Y = \bar{A}B\bar{C}(\bar{D} + D) + \bar{A}BC(\bar{D} + D) + AB\bar{C}(\bar{D} + D) + ABC(\bar{D} + D)$$

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

$$\begin{aligned}
 &= \bar{A}B(\bar{C} + C) + AB(\bar{C} + C) \\
 &= \bar{A}B + AB = B(\bar{A} + A) = B
 \end{aligned}$$

(ii) Consider the K-map shown in Fig.

	$\bar{C}\bar{D}$	$\bar{C}D$	$CD$	$C\bar{D}$
$\bar{A}\bar{B}$	0	0	0	0
$\bar{A}B$	0	0	0	0
$AB$	1	1	1	1
$A\bar{B}$	1	1	1	1

$$\begin{aligned}
 Y &= ABC\bar{D} + ABC\bar{D} + ABCD + ABC\bar{D} \\
 &\quad + A\bar{B}C\bar{D} + A\bar{B}C\bar{D} + A\bar{B}CD + A\bar{B}C\bar{D}
 \end{aligned}$$

First eliminate D,

$$\begin{aligned}
 Y &= ABC(\bar{D} + D) + ABC(D + \bar{D}) + A\bar{B}C(\bar{D} + D) + A\bar{B}C(D + \bar{D}) \\
 &= ABC + ABC + A\bar{B}C + A\bar{B}C
 \end{aligned}$$

Now eliminate C,

$$\begin{aligned}
 Y &= AB(\bar{C} + C) + A\bar{B}(\bar{C} + C) \\
 &= AB + A\bar{B}
 \end{aligned}$$

Now eliminate B,

$$\begin{aligned}
 Y &= A(B + \bar{B}) \\
 Y &= A
 \end{aligned}$$

In this way, three variables B, C, D and their complements dropout from the corresponding product.

#### Simplification using Karnaugh Map

As we have seen, a pair eliminates one variable, a quad eliminates two variables and their complements and an octate eliminates three variables and their complements. Because of this, after drawing the Karnaugh map, encircle the octet first, the quads second and pairs the last.

For example,

	$\bar{C}\bar{D}$	$\bar{C}D$	$CD$	$C\bar{D}$
$\bar{A}\bar{B}$	0	1	1	1
$\bar{A}B$	0	0	1	1
$AB$	1	1	0	1
$A\bar{B}$	1	1	0	1

In Fig. above there is no octet, but there are two quads and one pair.

Boolean algebra for pair

$$Y_1 = \bar{A}\bar{B}\bar{C}D + \bar{A}\bar{B}CD$$

$$Y_1 = \bar{A}\bar{B}D(\bar{C} + C)$$

$$Y_1 = \bar{A}\bar{B}D$$

The lower quad represents the output

$$Y_2 = ABC\bar{D} + ABCD + AB\bar{C}\bar{D} + AB\bar{C}D$$

$$Y_2 = ABC(\bar{D} + D) + AB\bar{C}(\bar{D} + D)$$

$$Y_2 = ABC + AB\bar{C}$$

$$Y_2 = A\bar{C}(B + \bar{B})$$

$$Y_2 = A\bar{C}$$

and the quad at right represents the output

$$Y_3 = \bar{A}\bar{B}C\bar{D} + \bar{A}BC\bar{D} + ABC\bar{D} + A\bar{B}C\bar{D}$$

$$Y_3 = \bar{A}C\bar{D}(\bar{B} + B) + AC\bar{D}(B + \bar{B})$$

$$Y_3 = \bar{A}C\bar{D} + AC\bar{D}$$

$$Y_3 = C\bar{D}(\bar{A} + A)$$

$$Y_3 = C\bar{D}$$

ORing these simplified products  $Y_1$ ,  $Y_2$ ,  $Y_3$ , we get the Boolean equation corresponding to the entire Karnaugh map.

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

Overlapping Groups



We can use the same 1 more than once.

	$\bar{C}\bar{D}$	$\bar{C}D$	$CD$	$C\bar{D}$
$\bar{A}\bar{B}$	0	0	0	0
$\bar{A}B$	0	1	0	0
$AB$	1	1	1	1
$A\bar{B}$	1	1	1	1

(a)

	$\bar{C}\bar{D}$	$\bar{C}D$	$CD$	$C\bar{D}$
$\bar{A}\bar{B}$	0	0	0	0
$\bar{A}B$	0	1	0	0
$AB$	1	1	1	1
$A\bar{B}$	1	1	1	1

(b)

Fig. (a) shows how the same 1 can be grouped. The 1 representing the fundamental product  $AB\bar{C}D$  is part of pair and part of octet.

Output of lower octet is  $A$  and for pair is  $\bar{B}\bar{C}D$ . Therefore simplified equation for overlapping group is,

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

But it is valid to encircle 1 as shown in Fig. 2.54 (b). Then  $Y$  output becomes

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

where  $A$  is output from octet and  $\bar{A}\bar{B}\bar{C}D$  is from encircled 1.

### Rolling the Map :

Consider the following Fig. 2.55.

	$\bar{C}\bar{D}$	$\bar{C}D$	$CD$	$C\bar{D}$
$\bar{A}\bar{B}$	0	0	0	0
$\bar{A}B$	1	0	0	1
$AB$	1	0	0	1
$A\bar{B}$	0	0	0	0

(a)

	$\bar{C}\bar{D}$	$\bar{C}D$	$CD$	$C\bar{D}$
$\bar{A}\bar{B}$	0	0	0	0
$\bar{A}B$	1	0	0	1
$AB$	1	0	0	1
$A\bar{B}$	0	0	0	0

(b)

The result from pair gives

$$\begin{aligned} Y &= \bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}B\bar{C}\bar{D} + \bar{A}\bar{B}C\bar{D} + \bar{A}B\bar{C}D \\ &= \bar{B}\bar{C}\bar{D}(\bar{A} + A) + \bar{B}C\bar{D}(\bar{A} + A) \\ &= \bar{B}\bar{C}\bar{D} + \bar{B}C\bar{D} \end{aligned}$$

Visualise picking up the Karnaugh map and rolling it so that left side touches the right side, two pairs actually form a quad. To show this, draw half circles around each pair as shown in Fig. 2.52 (b). Then quad output gives,

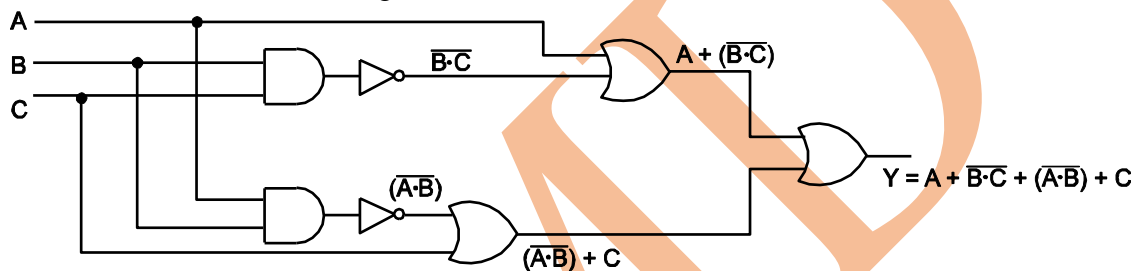
$$Y = B\bar{D}$$

This is output of rolled quad. Therefore 1s on the edges of K-map can be grouped with 1s on opposite edges.

**Example 1 :** Draw a logic circuit and obtain truth table for the following expression

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

**Solution :** Consider the Fig. 2.56.



**Fig.**

**Table: Truth table**

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

Using Boolean expression, we can reduce this circuit as follows.

Consider  $Y = A + \bar{B}\cdot\bar{C} + \bar{A}\cdot\bar{B} + C$

Applying De Morgan's theorem,

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

$$\square \quad Y = (A + \bar{A}) + \bar{B} + \bar{B} + C + \bar{C} \quad (\overline{A\cdot B} = \bar{A} + \bar{B})$$

$$= 1 + \bar{B} + 1$$

$$Y = 1 + \bar{B} \quad (\because A + \bar{A} = 1)$$

$$Y = 1 \quad (\because C + \bar{C} = 1)$$

Thus, output Y is HIGH irrespective of any input H or L as it can be seen from truth table also.

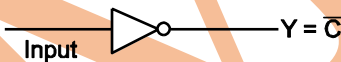
**Example 2. :** Simplify the equation and then draw logic diagram.

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

**Solution :** Consider  $Y = \bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC\bar{C}$

$$\begin{aligned} \square \quad Y &= \bar{A}\bar{C}(\bar{B} + B) + A\bar{C}(\bar{B} + B) \\ &= \bar{A}\bar{C}(1) + A\bar{C}(1) \quad (\because \bar{B} + B = 1) \\ &= \bar{A}\bar{C} + A\bar{C} \\ &= \bar{C}(\bar{A} + A) \quad (\because \bar{A} + A = 1) \\ &= \bar{C} \end{aligned}$$

The logic circuit to solve the above equation is



We can verify the result by considering the input conditions : A = 0, B = 0 and C = 1.

The expected result is  $Y = \bar{C}$  i.e.  $Y = 0$

By applying the input, we get,

$$\begin{aligned} Y &= \bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC\bar{C} \\ &= \bar{0}\bar{0}\bar{1} + \bar{0}0\bar{1} + 0\bar{0}\bar{1} + 00\bar{1} \\ &= 0 + 0 + 0 + 0 \\ &= 0 \end{aligned}$$

i.e.  $Y = \bar{C}$

**Example 3 :** Simplify the following equation and then draw logic diagram and truth table.

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

**Solution :** Consider  $Y = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}\bar{C} + B$

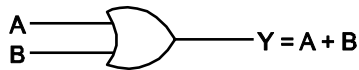
$$\square \quad Y = \bar{A}\bar{B}(C + \bar{C}) + B$$

$$= A\bar{B}(1) + B \quad (\because C + \bar{C} = 1)$$

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

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

The logic circuit is,



**Table: Truth table**

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

**Example 4 :** Simplify the following Boolean equation and then draw logic diagram and truth table :

$$Y = AB\bar{C} + ABC + BC$$

**Solution :**

$$Y = AB\bar{C} + ABC + BC$$

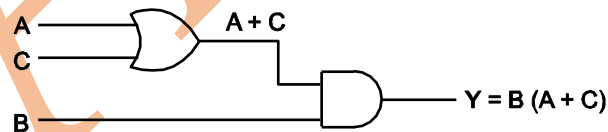
$$Y = AB(\bar{C} + C) + BC \quad (\because C + \bar{C} = 1)$$

$$Y = AB(1) + BC$$

$$Y = AB + BC$$

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

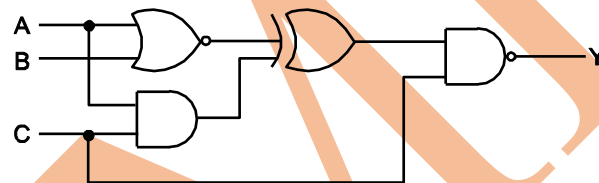
The logic circuit is,



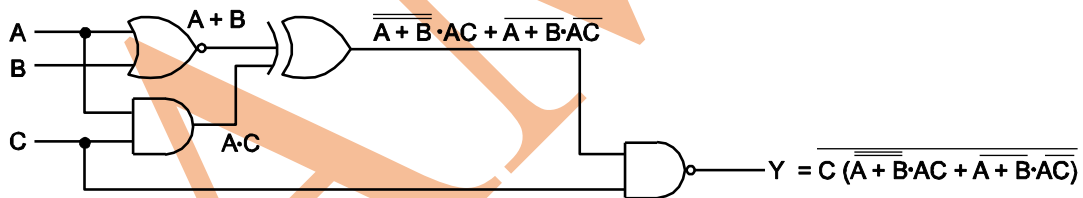
**: Truth table**

Inputs			Output
A	B	C	$Y = B(A + C)$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

**Example 5 :** Write the Boolean expression for the following logic diagram and give its truth table.



**Solution :** Consider the



**Fig. 2.61**

The Boolean expression will be,

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

$$Y = \overline{\overline{C}} + [\overline{\overline{A+B}} \cdot AC + \overline{\overline{A+B}} \cdot AC] \dots \overline{A \cdot B} = \overline{A} + \overline{B}$$

$$Y = \overline{\overline{C}} + [(A+B \cdot AC) + (\overline{A+B} \cdot \overline{AC})] \dots \overline{\overline{A}} = A$$

$$Y = \overline{\overline{C}} + [(A+B \cdot AC) \cdot (\overline{A+B} \cdot \overline{AC})] \dots \overline{\overline{A+B}} = \overline{A} \cdot \overline{B}$$

$$Y = \overline{\overline{C}} + [\overline{\overline{A+B}} + \overline{AC}] \cdot (\overline{A+B} + \overline{AC})$$

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

$$Y = \bar{C} + [(\bar{A} \cdot \bar{B} + \bar{A} + \bar{C}) \cdot (A(1 + C) + B)]$$

$$Y = \bar{C} + [(\bar{A}(1 + \bar{B}) + \bar{C}) \cdot (A(1 + C) + B)]$$

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

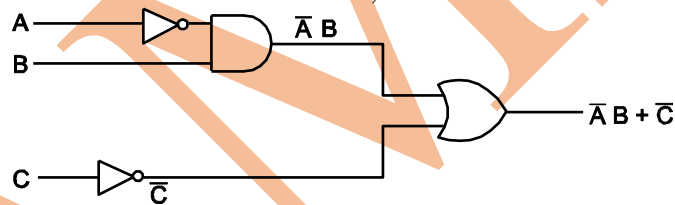
$$Y = \bar{C} + [(\bar{A}A + \bar{A}B + \bar{A}\bar{C} + \bar{C}B)]$$

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

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

$$Y = \bar{C}(1 + A) + \bar{A}B + \bar{C}B = \bar{C} + \bar{A}B + \bar{C}B$$

Circuit can be reduced and can be drawn as,



**Truth table**

Inputs			Output
A	B	C	$Y = \bar{A}B + \bar{C}$
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

**Example 6 :** Minimise the equation  $y = \bar{A}\bar{B}C + \bar{A}BC + A\bar{B}C + ABC$  using Boolean Algebra or K-maps.

**Solution : (i) Using Boolean algebra :** Rearranging the above

$$\begin{aligned} y &= \bar{A}\bar{B}C + \bar{A}BC + A\bar{B}C + ABC \\ &= (\bar{A} + A)\bar{B}C + (\bar{A} + A)BC \quad (\square A + \bar{A} = 1) \\ &= \bar{B}C + BC = C(\bar{B} + B) \\ y &= C \end{aligned}$$

$\therefore$

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

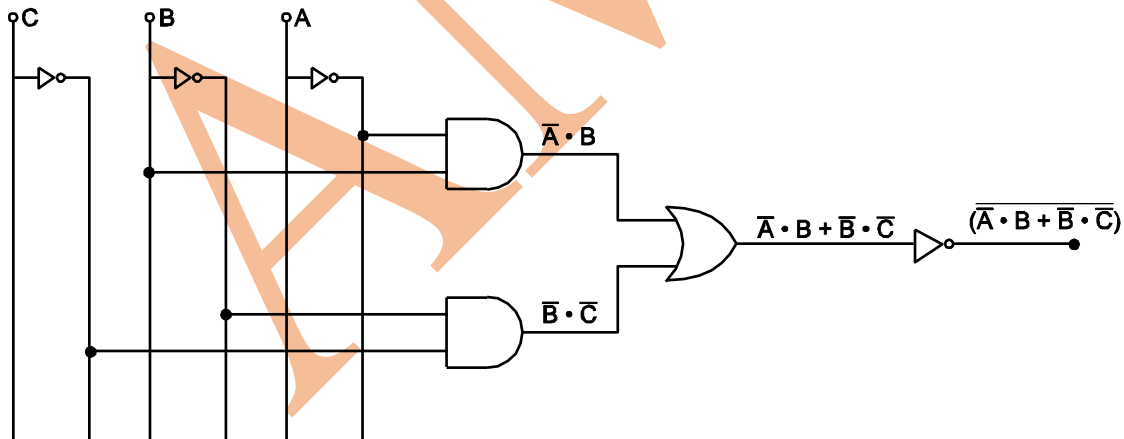
**(ii) Using K-map :** Since equation contains three variables, the K-map will have  $2^3 = 8$  cells.

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

$$y = C$$

**Example 7 :** Draw the logic diagram for the expression  $y = \overline{(\bar{A} \cdot B + \bar{B} \cdot \bar{C})}$ .

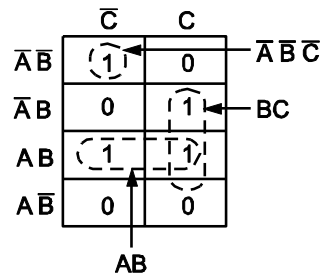
**Solution :**



**Example 8 :** Simplify the following SOP expression using K-map.

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

**Solution :** The K-map for the given equation is

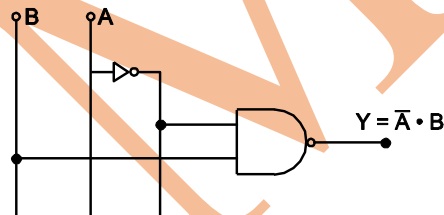


$\therefore y = AB + BC + \bar{A}\bar{B}\bar{C}$

**Example 9 :** Simplify the logic expression.  $y = \bar{A}\bar{B} + \bar{A}\bar{B}\bar{C} + \bar{A}BCD + \bar{A}BC\bar{D}$  using the laws of Boolean algebra. Draw simplified logic diagram.

**Solution :** Consider expression

$$\begin{aligned}
 y &= \bar{A}\bar{B} + \bar{A}\bar{B}\bar{C} + \bar{A}BCD + \bar{A}BC\bar{D} \\
 &= \bar{A}\bar{B}(1 + \bar{C}) + \bar{A}BC(D + \bar{D}) \quad (\square 1 + \bar{C} = 1, D + \bar{D} = 1) \\
 &= \bar{A}\bar{B} + \bar{A}BC \\
 &= \bar{A}\bar{B}(1 + C) \\
 &= \bar{A}\bar{B} \quad (\square 1 + C = 1)
 \end{aligned}$$



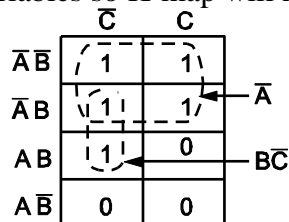
**: Logic diagram**

**Example 10 :** Minimize the following logical expression using K-maps.

$$y = \bar{A}\bar{B}C + \bar{A}\bar{B}\bar{C} + A\bar{B}\bar{C} + \bar{A}B\bar{C} + \bar{A}BC$$

**Solution :** Consider expression  $y = \bar{A}\bar{B}C + \bar{A}\bar{B}\bar{C} + A\bar{B}\bar{C} + \bar{A}BC$

Since expression has three variables so K-map will have  $2^3 = 8$  cells.



$\therefore y = \bar{A} + B\bar{C}$



**Example 11 :** Convert the following SOP expression into standard SOP form :

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

**Solution :** Given expression is

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

**Step 1 :** Find the missing literal in each term

$$Y = AB \quad + \quad AC \quad + \quad B\bar{C}$$

$\uparrow \qquad \qquad \uparrow \qquad \qquad \uparrow$   
 missing missing missing  
 literal literal literal  
 is is is  
 C B A

**Step 2 :** And each term with its (missing literal + its complement)

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

Since  $C + \bar{C} = 1$ , the value of expression does not change as

$$\begin{aligned} Y &= ABC + AB\bar{C} + ABC + A\bar{B}C + AB\bar{C} + \bar{A}B\bar{C} \\ &= ABC + ABC + AB\bar{C} + A\bar{B}C + \bar{A}B\bar{C} + \bar{A}B\bar{C} \quad (\square A + A = A) \\ &= ABC + AB\bar{C} + A\bar{B}C + \bar{A}B\bar{C} \end{aligned}$$

Since in the above expression each term contains all the literals it is in the standard SOP form.

**Example 12 :** Simplify the following expression using K-map :

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

**Solution :** Given expression contains three literals. So the K-map will have  $2^3 = 8$  cells.

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

$$y = \bar{A}\bar{C} + \bar{A}B + BC$$

**Example 13 :** Simplify the following using Boolean algebra.

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

**Solution :**

$$\begin{aligned} Y &= \bar{A}\bar{B}C + \bar{A} \cdot \bar{B} \cdot \bar{\bar{C}} + \bar{A}\bar{B}C\bar{D} \quad \text{Using } \overline{A + BC} = \bar{A} \cdot \bar{B} \cdot \bar{C} \\ &= \bar{A}\bar{B}C + \bar{A} \cdot \bar{B} \cdot C + \bar{A}\bar{B}C\bar{D} \quad \text{Using } \bar{\bar{A}} = A \\ &= \bar{A}\bar{B}C + \bar{A}\bar{B}C\bar{D} \quad (A + A = A) \\ Y &= \bar{A}\bar{B}(C + \bar{C}D) \end{aligned}$$