Boolean Algebra

In 1854 George Boole introduced a systematic approach of logic and developed an algebraic system to treat the logic functions, which is now called Boolean algebra. In 1938 C.E. Shannon developed a two-valued Boolean algebra called Switching algebra, and demonstrated that the properties of two-valued or bistable electrical switching circuits can be represented by this algebra.

Boolean algebra is a mathematical system for the manipulation of variables that can have one of two values. In digital systems, these values are “on” and “off,” 1 and 0, or “high” and “low.”

The following Huntington postulates are satisfied for the definition of Boolean algebra on a set of elements S together with two binary operators (+) and (.).

1. (*a*) Closer with respect to the operator (+).

   (*b*) Closer with respect to the operator (.).
2. (*a*) An identity element with respect to + is designated by 0 *i.e.*, x + 0 = 0 + x = x.

   (*b*) An identity element with respect to . is designated by 1 *i.e.*, x.1 = 1. x= x.

Two-Valued Boolean Algebra

Two-valued Boolean algebra is defined on a set of only two elements, S = {0,1}, with rules for two binary operators (+) and (.) and inversion or complement as shown in the following operator tables, respectively.

A	B	A + B
0	0	0
0	1	1
1	0	1
1	1	1

A	B	A.B
0	0	0
0	1	0
1	0	0
1	1	1

A	A′
0	1
1	0

The basic rules of Boolean algebra are:-

Postulates and Theorems of Boolean Algebra

Postulate 2	(a)	$x + 0 = x$	(b)	$x \cdot 1 = x$
Postulate 5	(a)	$x + x' = 1$	(b)	$x \cdot x' = 0$
Theorem 1	(a)	$x + x = x$	(b)	$x \cdot x = x$
Theorem 2	(a)	$x + 1 = 1$	(b)	$x \cdot 0 = 0$
Theorem 3, involution		$(x')' = x$		
Postulate 3, commutative	(a)	$x + y = y + x$	(b)	$xy = yx$
Theorem 4, associative	(a)	$x + (y + z) = (x + y) + z$	(b)	$x(yz) = (xy)z$
Postulate 4, distributive	(a)	$x(y + z) = xy + xz$	(b)	$x + yz = (x + y)(x + z)$
Theorem 5, DeMorgan	(a)	$(x + y)' = x'y'$	(b)	$(xy)' = x' + y'$
Theorem 6, absorption	(a)	$x + xy = x$	(b)	$x(x + y) = x$

The proving of theorems can be done by using the **Postulates** or the **truth table** as illustrated in the following :

THEOREM 1(a): $x + x = x$.

Statement	Justification
$x + x = (x + x) \cdot 1$	postulate 2(b)
$= (x + x)(x + x')$	5(a)
$= x + xx'$	4(b)
$= x + 0$	5(b)
$= x$	2(a)

THEOREM 1(b): $x \cdot x = x$.

Statement	Justification
$x \cdot x = xx + 0$	postulate 2(a)
$= xx + xx'$	5(b)
$= x(x + x')$	4(a)
$= x \cdot 1$	5(a)
$= x$	2(b)

THEOREM 2(a): $x + 1 = 1$.

Statement	Justification
$x + 1 = 1 \cdot (x + 1)$	postulate 2(b)
$= (x + x')(x + 1)$	5(a)
$= x + x' \cdot 1$	4(b)
$= x + x'$	2(b)
$= 1$	5(a)

THEOREM 6(a): $x + xy = x$.

Statement	Justification
$x + xy = x \cdot 1 + xy$	postulate 2(b)
$= x(1 + y)$	4(a)
$= x(y + 1)$	3(a)
$= x \cdot 1$	2(a)
$= x$	2(b)

THEOREM 6(b): $x(x + y) = x$ by duality.

Example: prove that x(y + z)= xy + xz

Solution:

x	y	z	y + z	x(y + z)	xy	xz	xy + xz
0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	1	0	0	0	0
0	1	1	1	0	0	0	0
1	0	0	0	0	0	0	0
1	0	1	1	1	0	1	1
1	1	0	1	1	1	0	1
1	1	1	1	1	1	1	1

Hence, it is proved because the left side is similar to the right side

Example: prove that x + yz = (x+y)(x + z)

Solution:

x	y	z	yz	x + yz	x + y	x + z	(x+y)(x+z)
0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0
0	1	0	0	0	1	0	0
0	1	1	1	1	1	1	1
1	0	0	0	1	1	1	1
1	0	1	0	1	1	1	1
1	1	0	0	1	1	1	1
1	1	1	1	1	1	1	1

Hence, it is proved because the left side is similar to the right side

Operator Precedence

The operator precedence for evaluating Boolean expressions is (1) parentheses, (2) NOT, (3) AND, and (4) OR. In other words, expressions inside parentheses must be evaluated before all other operations. The next operation that holds precedence is the complement, and then follows the AND and, finally, the OR.

Boolean Function

A Boolean function is a relation between the binary inputs and the binary outputs. The value of a function (output) may be 0 or 1, depending on the values of inputs present in the Boolean function. Boolean Function can be described by:

1- a truth table
2- Boolean equation,
3- a logic diagram

1- Truth table

Truth table for a function is a list of all combinations of 1's and 0's that can be assigned to the binary variables and a list that shows the value of the function for each binary combination.

For n variables, there are 2^n rows (states)

For example, when the number of variables (inputs) n=3, then the number of rows (states) = 2^3 = 8 as shown in this table:

A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

2- Boolean equation (Boolean function form):

Boolean equation consists of a binary variable identifying the function (output) followed by an equal sign and a Boolean expression formed with binary variables, the two binary operators AND and OR, one unary operator NOT, and parentheses. When a Boolean expression is implemented with logic gates, each literal in the function is designated as input to the gate. The literal may be a primed or unprimed variable. For example, the Boolean equation of the truth table above is:

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

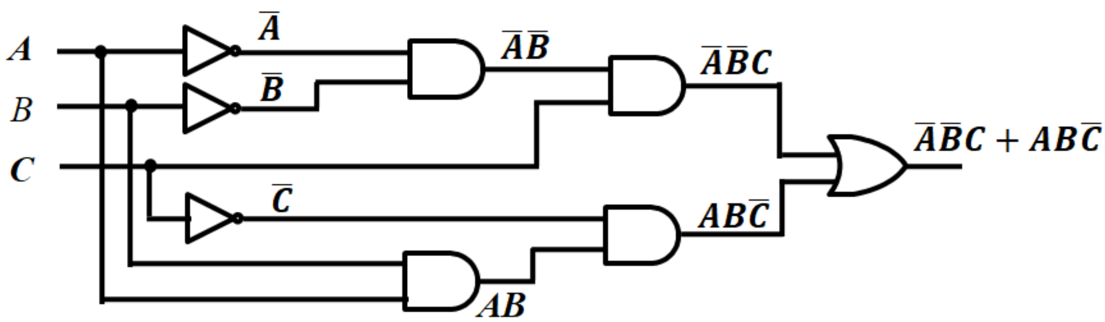
Where F is the function (output)

A, B, C are the input variables (literals)

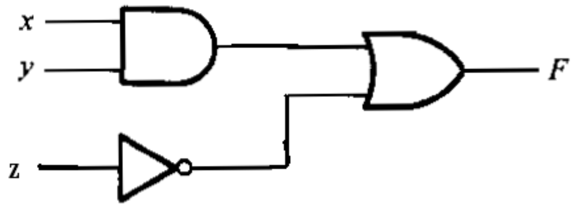
3- Logic diagram (circuit diagram):

The logic diagram composed of logic gates in which are interconnected by wires that carry logic signals. The figure below shows the logic diagram of the Boolean equation

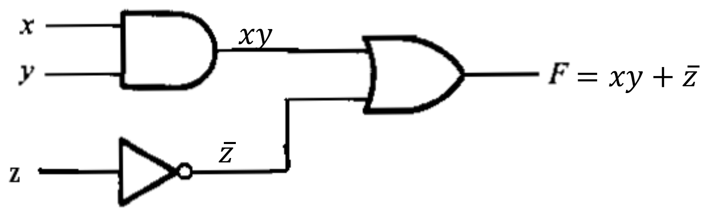
$$F = \bar{A}\bar{B}C + AB\bar{C}$$



Example: Write the Boolean expression for the logic diagram shown.



Solution:



The Boolean expression of this circuit is :

$$\boldsymbol{F = xy + \bar{z}}$$

Simplification using Boolean algebra:

Minimization of the number of literals and the number of terms leads to less complex circuits as well as less number of gates, which should be a designer's aim. There are several methods to minimize the Boolean function such as Boolean algebra and Karnaugh map (K-Map). Here, simplification or minimization of complex algebraic expressions will be shown with the help of postulates and theorems of Boolean algebra.

Example : Simplify the following Boolean expression.

$$\boldsymbol{F = xyz + \bar{x}y + xy\bar{z}}$$

Solution:

$\boldsymbol{F = xy(z + \bar{z}) + \bar{x}y}$

$\boldsymbol{F = xy + \bar{x}y}$

$\boldsymbol{F = y(x + \bar{x})}$

$\boldsymbol{F = y}$

Example : Simplify the following Boolean expression.

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

Solution:

$$\boldsymbol{F = A(B + \overline{B}) + B\overline{A}}$$

$$\boldsymbol{F = A + B\overline{A}} \qquad \boldsymbol{because\ (B + \overline{B}) = 1}$$

$$\boldsymbol{F = (A + B)(A + \overline{A})}$$

$$\boldsymbol{F = A + B} \qquad \boldsymbol{because\ (A + \overline{A}) = 1}$$

Example: Simplify the following Boolean expression.

$$\boldsymbol{F = \overline{((\overline{x} + y).\overline{z})}}$$

Solution: using De Morgan theorem

$$\boldsymbol{F = \overline{(\overline{x} + y)} + \overline{\overline{z}}}$$

$$\boldsymbol{F = \overline{\overline{x}}.\overline{y} + z}$$

$$\boldsymbol{F = x.\overline{y} + z}$$

Example: simplify and draw the logic diagram of

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

Solution:

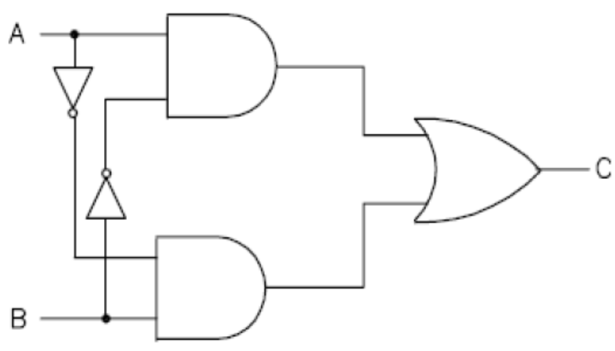
$C = (A + B)\overline{A.B}$ *applying DeMorgan theorem*

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

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

$C = 0 + A\bar{B} + B\bar{A} + 0$ *because* $A\bar{A} = 0$ *and* $B\bar{B} = 0$

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

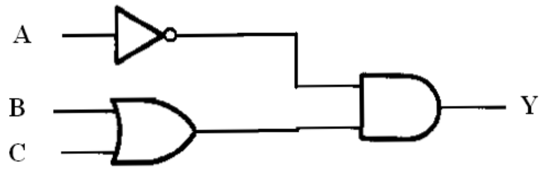


Example : Draw the Boolean expression

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

a) using basic logic gates.
b) using NOR gates only.

Solution :
a)



b)