14

Boolean Logic

In This Chapter

14.1 DEVELOPMENT OF BOOLEAN LOGIC

Long ago Aristotle constructed a complete system of formal logic and wrote six famous works on the subject, contributing greatly to the organization of man's reasoning. For centuries afterward, mathematicians kept on trying to solve these logic problems using conventional algebra but only *George Boole* could manipulate these symbols successfully to arrive at a solution with his own mathematical system of logic. Boole's revolutionary paper *'An Investigation of the laws of the thought'* was published in 1854 which led to the development of new system, the *algebra of logic,* 'BOOLEAN ALGEBRA' or 'BOOLEAN LOGIC'.

Boole's work remained confined to papers only until 1938 when *Claude E. Shannon* wrote a paper titled *'A Symbolic Analysis of Relay Switching Circuits'*. In this paper he applied Boolean Logic to solve relay logic problems. As logic problems are binary decisions and Boolean logic effectively deals with these binary values. Thus it is also called *'Switching Algebra'*.

14.2 BINARY VALUED QUANTITIES

Everyday we have to make logic decisions : "Should I carry the book or not ?" "Should I use calculator or not ?" ; "Should I miss TV Programme or not ?". Each of these questions requires a YES or NO answer as there are only these two possible answers.

BINARY DECISION

The decision which results into either YES (TRUE) or NO (FALSE) is called a Binary Decision.

Therefore, each of the above mentioned is a binary decision. Binary decision making also applies to formal logic.

For example, let us consider the following :

1. Indira Gandhi was the only woman Prime Minister of India.
2. 13 – 2 = 11.
3. Delhi is the biggest state in India.
4. What do you say ?
5. What did I tell you yesterday ?

Values *true* and *false* are called Truth values.

1st and 2nd sentences are TRUE but 3rd is FALSE; 4th and 5th are questions which cannot be answered in TRUE and FALSE.

Thus, sentences which can be determined to be *true* or *false* are called *logical statements or truth functions* and the results TRUE or FALSE are called *truth values*. The truth values are depicted by *logical constants* TRUE and FALSE or 1 and 0. 1 means TRUE and 0 means FALSE. And the variables which can store these truth values are called *logical variables* or ***binary valued variables*** as these can store one of the two values TRUE or FALSE.

NOTE

Boolean variables can have value either as 1 (True) or as 0 (zero)

14.3 LOGICAL OPERATIONS

There are some specific operations that can be applied on truth functions. Before learning about these operations, you must know about compound logical functions and logical operators.

14.3.1 Logical Function or Compound Statement

Algebraic variables like *a, b, c* or *x, y, z* etc. are combined with the help of *mathematical operators* like +, –, ×, / to form algebraic expressions *e.g.,*

$$2 \times A + 3 \times B - 6 \times C = (10 \times Z)/2 \times Y \quad \textit{i.e.,} \quad 2A + 3B - 6C = 10Z/2Y$$

Similarly, logic statements or truth functions are combined with the help of *Logical Operators* like AND, OR and NOT to form a *Compound statement or Logical function. e.g.,*

He prefers tea *not* coffee.

He plays guitar *and* she plays sitar.

I watch TV on Sundays *or* I go for swimming.

These logical operators are also used to combine logical variables and logical constants to form *logical expressions e.g.,* assuming *x, y* are logical variables

X NOT Y OR Z

Y AND X OR Z

14.3.2 Logical Operators

TRUTH TABLE

A Truth Table is a table which represents all the possible values of logical variables / statements along with all the possible results of the given combinations of values.

Before we start discussion about logical operators, let us first understand *what a Truth Table is.*

For example, following logical statements can have only one of the *two* values (TRUE (YES) or FALSE (NO))

1. I want to have tea. 2. Tea is readily available.

Let us represent all the possible combinations of values these statements can have in the tabular form :

I want to have tea	T	T	F	F
Tea is readily available	T	F	T	F
(Result) I'll have tea	T	F	F	F

T represents True
F represents False

Or If we represent first statement as *X* and second statement as *Y* and result as *R* then the above table can also be written as follows :

Table 14.1

1 represents TRUE value and
0 represents FALSE value

X	Y	R
1	1	1
1	0	0
0	1	0
0	0	0

TAUTOLOGY

If the result of any logical statement or expression is always TRUE or 1 for all input combinations, it is called Tautology.

FALLACY

If the result of any logical statement or expression is always FALSE or 0 for all input combinations, it is called Fallacy.

This is a truth table *i.e., table of truth values of truth functions.*

Now let us proceed with our discussion about logical operators *i.e.,*

- NOT Operator
- OR Operator
- AND Operator

NOT Operator

This **operator operates on single variable** and operation performed by NOT operator is called *complementation* and the symbol we use for it is (bar). Thus $\overline{X}$ means complement of X and $\overline{YZ}$ means complement of YZ. As we know, the variables used in boolean equations have a unique characteristic that they may assume only one of two possible values 0 and 1, where 0 denotes FALSE and 1 denotes TRUE value. Thus the complement operation can be defined quite simply.

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

Table 14.2 Truth Table for NOT Operators

X	$\overline{X}$ (i.e., NOT X)
0	1
1	0

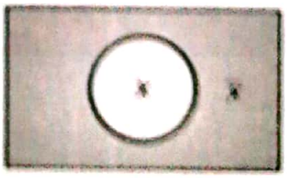


Figure 14.1 Venn diagram for $\overline{X}$

Scanned with CamScanner

Several other symbols *e.g.* ~, ′ are also used for the complementation symbol. If ~ is used then ~ X is read is 'negation of X' and if ′ is used then X′ is read as complement of X. NOT operation is singular or unary operation as it operates on single variable. Venn diagram for $\overline{X}$ is given in Fig. 14.1 where shaded area depicts $\overline{X}$.

OR Operator

A second important operator in boolean algebra is OR operator which denotes operation called *logical addition* and the symbol we use for it is +. The + symbol, therefore, does not have the 'normal' meaning, but is a *logical addition* or logical OR symbol. Thus $X + Y$ can be read as **X OR Y**. For OR operation the possible input and output combinations are as follows :

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

Shaded Portion shows $X + Y$

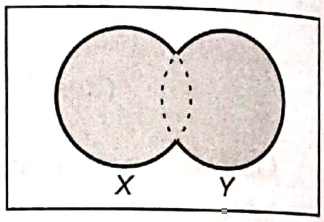


Figure 14.2 Venn diagram for X + Y.

And the truth table of OR operator is given below :

Table 14.3 *Truth Table for OR Operator*

X	Y	X + Y (i.e., X OR Y)
0	0	0
0	1	1
1	0	1
1	1	1

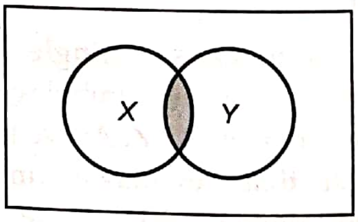
Note that when any one of X and Y is 1, X + Y is 1.

To avoid ambiguity, there are other symbols *e.g.*, *U*, *v*, and *V* have been recommended as replacements for the + sign. Computer people still use the + sign, however, which was the symbol originally proposed by Boole.Venn diagram for $X + Y$ is given (Fig. 14.2), where shaded area depicts $X + Y$.

AND Operator

AND operator performs another important operation of boolean algebra called *logical multiplication* and the symbol for AND operation is (.) dot. Thus X . Y will be read as X AND Y. The rules for AND operation are :

$$0 . 0 = 0$$
$$0 . 1 = 0$$
$$1 . 0 = 0$$
$$1 . 1 = 1$$



Shaded Portion shows $X \cdot Y$

X.Y

Figure 14.3 Venn diagram for (X . Y).

and the truth table for AND is as follows :

Table 14.4 *Truth Table for AND Operator*

X	Y	X . Y (i.e., X AND Y)
0	0	0
0	1	0
1	0	0
1	1	1

Note that only when both X and Y are 1's, then XY has the result 1.

Venn diagram for X . Y is given in Fig. 14.3, where shaded area depicts (X . Y).

14.3.3 Evaluation of Boolean Expressions Using Truth Table

Logical variables are combined by means of logical operators (AND, OR, NOT) to form a boolean expression.

For example, $X + \overline{Y.Z} + \overline{Z}$ is a boolean expression.

It is often convenient to shorten X . Y . Z to XYZ, and using this convention, above expression can be written as $X + \overline{YZ} + \overline{Z}$

To study a boolean expression, it is very useful to construct a table of values for the variables and then to evaluate the expression for each of the possible combinations of variables in turn.

Consider the expression $X + \overline{YZ}$. Here three variables X, Y, Z are forming the expression, each of the variables can assume the value 0 or 1. The possible combinations of values may be arranged in ascending order as in Table 14.5.

NOTE

A truth table of *n* input variables with have 2^n input combinations *i.e.*, 2^n rows *e.g.*, a 4-variable truth table will have 2^4 *i.e.*, 16 rows in it.

Table 14.5 *Possible Combinations of X, Y and Z*

X	Y	Z
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

Since X, Y, Z are three (3) variables in total a truth table involving 3 input variables will have 2^3 i.e., 8 rows in total. The left most column will have half of total entries (i.e., 4 entries) as zeros and half as 1's (in total 8).

The next column will have no of zero's and 1's halved than first column completing 8 rows and so on. That is why, first column has 4 0's and 4 1's, next column has two 0's followed by two 1's completing 8 rows in total and the last column has one 0's followed by one 1's completing 8 rows

So a column is added to list Y . Z (Table 14.6)

Table 14.6 *Truth Table for (Y . Z)*

X	Y	Z	Y . Z
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	0
1	1	1	1

Here AND operation is applied only on columns Y and Z

One more column is now added to list the values of $\overline{YZ}$ (Table 14.7)

Table 14.7 *Truth Table for Y . Z and $\overline{YZ}$.*

X	Y	Z	Y . Z	$\overline{YZ}$
0	0	0	0	1
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	1
1	1	0	0	1
1	1	1	1	0

Note that $\overline{YZ}$ contains complemented values of YZ

Now values of X are ORed (*logical addition*) to the values of $\overline{YZ}$ and the resultant values are contained in the last column (Table 14.8).

Table 14.8 *Truth Table for $X + \overline{YZ}$.*

X	Y	Z	Y . Z	$\overline{YZ}$	$X + \overline{YZ}$
0	0	0	0	1	1
0	0	1	0	1	1
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	0	1	1
1	0	1	0	1	1
1	1	0	0	1	1
1	1	1	1	0	1

Now observe the expression $X + \overline{YZ}$, after ANDing Y and Z, the result has been complemented and then ORed with X.

Here the result is 0 only when both the columns X and $\overline{YZ}$ have 0, otherwise if there is 1 in any of the two columns X and $\overline{YZ}$, the result is 1.

Please note here, while evaluating boolean expression there is a precedence order which is to be taken care of always. Always the order of evaluation of logical operators is **firstly NOT then AND** and **then OR.** If there are parenthesis, then the expression in parenthesis is evaluated first.

Check Point 14.1

1. Name the person who developed boolean logic.
2. What is the other name of boolean logic ? In which year was the boolean logic/algebra developed ?
3. What is a binary decision ? What do you mean by a binary valued variable ?
4. What do you mean by tautology and fallacy ?
5. What is a logic gate ? Name the three basic logic gates.

EXAMPLE 14.1 *Using Boolean logic, verify using truth table that $X + XY = X$ for each X, Y in {0, 1}.*

Solution. As the expression $X + XY = X$ is a two-variable expression, so we require possible combination of values of X, Y. Truth Table will be as follows :

X	Y	XY	X + XY
0	0	0	0
0	1	0	0
1	0	0	1
1	1	1	1

Comparing the columns $X + XY$ and X, we find, contents of both the columns are identical, hence verified.

EXAMPLE 14.2 *Using Boolean logic, verify using truth table that $(X + Y)' = X'Y'$ for each X, Y in {0, 1}.*

Solution. As it is a 2-variable expression, truth table will be as follows :

X	Y	X + Y	(X + Y)'	X'	Y'	X'Y'
0	0	0	1	1	1	1
0	1	1	0	1	0	0
1	0	1	0	0	1	0
1	1	1	0	0	0	0

Comparing the columns $(X + Y)'$ and $X'Y'$, both the columns are identical, hence verified.

EXAMPLE 14.3 *Prepare a table of combinations for the following boolean logic expressions :*

(a) $\overline{X}\,\overline{Y} + \overline{X}Y$ (b) $XY\overline{Z} + \overline{X}\,\overline{Y}Z$ (c) $\overline{X}Y\overline{Z} + X\overline{Y}$

Solution. (a) As $\overline{X}\overline{Y} + \overline{X}Y$ is a 2-variable expression, its truth table is as follows :

X	Y	$\overline{X}$	$\overline{Y}$	$\overline{X}\,\overline{Y}$	$\overline{X}\,Y$	$\overline{X}\,\overline{Y} + \overline{X}\,Y$
0	0	1	1	1	0	1
0	1	1	0	0	1	1
1	0	0	1	0	0	0
1	1	0	0	0	0	0

(b) Truth table for this 3 variable expression is as follows :

X	Y	Z	$\overline{X}$	$\overline{Y}$	$\overline{Z}$	$XY\overline{Z}$	$\overline{X}\,\overline{Y}Z$	$XY\overline{Z} + \overline{X}\,\overline{Y}Z$
0	0	0	1	1	1	0	0	0
0	0	1	1	1	0	0	1	1
0	1	0	1	0	1	0	0	0
0	1	1	1	0	0	0	0	0
1	0	0	0	1	1	0	0	0
1	0	1	0	1	0	0	0	0
1	1	0	0	0	1	1	0	1
1	1	1	0	0	0	0	0	0

(c) Truth table for $\overline{X}Y\overline{Z} + X\overline{Y}$ is as follows :

X	Y	Z	$\overline{X}$	$\overline{Y}$	$\overline{Z}$	$\overline{X}Y\overline{Z}$	$X\,\overline{Y}Z$	$\overline{X}Y\overline{Z} + X\,\overline{Y}Z$
0	0	0	1	1	1	0	0	0
0	0	1	1	1	0	0	0	0
0	1	0	1	0	1	1	0	1
0	1	1	1	0	0	0	0	0
1	0	0	0	1	1	0	1	1
1	0	1	0	1	0	0	1	1
1	1	0	0	0	1	0	0	0
1	1	1	0	0	0	0	0	0

Check Point 14.2

1. Which gates implement logical addition, logical multiplication and complementation ?
2. What is the other name of NOT gate ?
3. What is a truth table ? What is the other name of truth table ?
   (i) $A + 0 = ?$ (ii) $A + 1 = ?$
   (iii) $A . 0 = ?$ (iv) $A . 1 = ?$
4. How many input combinations can be there in the truth table of a logic system having (N) input binary variables ?

EXAMPLE 14.4 *Prepare truth table for the following boolean expressions :*

(*a*) $X(\overline{Y}+\overline{Z})+X\overline{Y}$ (*b*) $X\overline{Y}(Z+Y\overline{Z})+\overline{Z}$ (*c*) $A[(\overline{B}+C)+\overline{C}]$

Solution. (*a*) Truth table for $X(\overline{Y}+\overline{Z})+X\overline{Y}$ is as follows :

X	Y	Z	$\overline{Y}$	$\overline{Z}$	$(\overline{Y}+\overline{Z})$	$X(\overline{Y}+\overline{Z})$	$X\overline{Y}$	$X(\overline{Y}+\overline{Z})+X\overline{Y}$
0	0	0	1	1	1	0	0	0
0	0	1	1	0	1	0	0	0
0	1	0	0	1	1	0	0	0
0	1	1	0	0	0	0	0	0
1	0	0	1	1	1	1	1	1
1	0	1	1	0	1	1	1	1
1	1	0	0	1	1	1	0	1
1	1	1	0	0	0	0	0	0

(*b*) Truth Table for $X\overline{Y}(Z+Y\overline{Z})+\overline{Z}$ is as follows :

X	Y	Z	$\overline{Y}$	$\overline{Z}$	$Y\overline{Z}$	$Z+Y\overline{Z}$	$X\overline{Y}$	$X\overline{Y}(Z+Y\overline{Z})$	$X\overline{Y}(Z+Y\overline{Z})+\overline{Z}$
0	0	0	1	1	0	0	0	0	1
0	0	1	1	0	0	1	0	0	0
0	1	0	0	1	1	1	0	0	1
0	1	1	0	0	0	1	0	0	0
1	0	0	1	1	0	0	1	0	1
1	0	1	1	0	0	1	1	1	1
1	1	0	0	1	1	1	0	0	1
1	1	1	0	0	0	1	0	0	0

(*c*) Truth Table for $A[(\overline{B}+C)+\overline{C}]$ is as follows :

A	B	C	$\overline{B}$	$\overline{C}$	$(\overline{B}+C)$	$(\overline{B}+C)+\overline{C}$	$A[(\overline{B}+C)+\overline{C}]$
0	0	0	1	1	1	1	0
0	0	1	1	0	1	1	0
0	1	0	0	1	0	1	0
0	1	1	0	0	1	1	0
1	0	0	1	1	1	1	1
1	0	1	1	0	1	1	1
1	1	0	0	1	0	1	1
1	1	1	0	0	1	1	1

14.4 BASIC LOGIC GATES

After *Shannon* applied boolean logic in telephone switching circuits, engineers realized that boolean algebra could be applied to computer electronics as well.

In the computers, these boolean operations are performed by logic gates.

What is a Logic Gate ?

A Gate is a basic electronic circuit which operates on one or more signals to produce an output signal.

Gates are digital (two-state) circuits because the input and output signals are either low voltage (denotes 0) or high voltage (denotes 1). Gates are often called *logic circuits* because they can be analyzed with boolean logic. There are *three* types of logic gates :

- Inverter (NOT gate)
- OR gate
- AND gate

GATE

A Gate is a basic electronic circuit which operates on one or more signals to produce an output signal.

14.4.1 Inverter (NOT Gate)

An **Inverter (Not Gate)** is a gate with only one input signal and one output signal; the output state is always the opposite of the input state.

An inverter is also called a NOT gate because the output is not the same as the input. The output is sometimes called the *complement* (opposite) of the input.

Following tables summarise the operation :

INVERTER (NOT GATE)

An Inverter (Not Gate) is a gate with only one input signal and one output signal; the output state is always the opposite of the input state.

Table 14.9 *Truth Table for NOT gate*

X	$\bar{X}$
Low	High
High	Low

Table 14.10 *Alternative Truth Table for NOT Gate*

X	$\bar{X}$
0	1
1	0

A low input *i.e.*, 0 produces high output *i.e.*, 1, and vice versa. The symbol for inverter is given in adjacent Fig. 14.4.

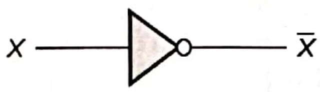


Figure 14.4 NOT gate symbol

14.4.2 OR Gate

The **OR Gate** has two or more input signals but only one output signal. If any of the input signals is 1 (high), the output signal is 1 (high).

If all inputs are 0 then output is also 0. If one or more inputs are 1, the output is 1.

An OR gate can have as many inputs (2 or more inputs) as desired. No matter how many inputs are there, the action of OR gate is the same : one or more 1 (high) inputs produce output as 1.

OR GATE

The **OR Gate** has two or more input signals but only one output signal. If any of the input signals is 1 (high), the output signal is 1 (high).

Following tables show OR action

Table 14.11 *Two Input OR Gate*

X	Y	F
0	0	0
0	1	1
1	0	1
1	1	1

F = X + Y

Table 14.12 *Three Input OR Gate*

X	Y	Z	F
0	0	0	0
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

F = X + Y + Z

The symbol for OR gate is given below :

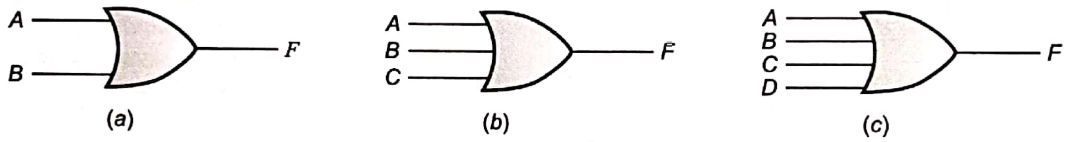


Figure 14.5 (*a*) Two input OR gate (*b*) Three input OR gate (*c*) Four input OR gate.

14.4.3 AND gate

The **AND Gate** can have two or more than two input signals and produce an output signal. When all the inputs are 1 *i.e.*, high then the output is 1 otherwise output is 0 only.

If any of the inputs is 0, the output is 0. To obtain output as 1, all inputs must be 1.

An AND gate can have as many inputs (2 or more inputs) as desired. Following tables illustrate AND action.

> **AND GATE**
>
> The **AND Gate** can have two or more than two input signals and produce one output signal. When all the inputs are 1 *i.e.*, high then the output is 1 otherwise output is 0 only.

Table 14.13 *Two Input AND Gate*

X	Y	F
0	0	0
0	1	0
1	0	0
1	1	1

Here, F = X . Y

Table 14.14 *Three Input AND Gate*

X	Y	Z	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

Here, F = X . Y . Z

The symbol for AND is

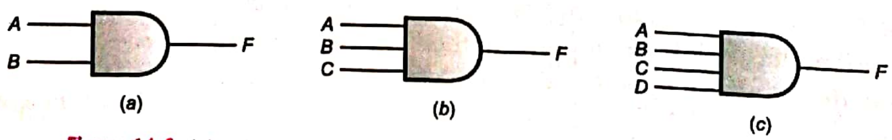


Figure 14.6 (*a*) 2-input AND gate (*b*) 3-input AND gate (*c*) 4-input AND gate

14.5 BASIC POSTULATES OF BOOLEAN LOGIC

Boolean logic algebra, being a system of mathematics, consists of *fundamental laws* that are used to build a workable, cohesive framework upon which are based the theorems of boolean algebra. These fundamental laws are known as *Basic postulates of boolean logic algebra.* These postulates state basic relations in boolean algebra, that follow :

I. If $X \neq 0$ then $X = 1$; and If $X \neq 1$ then $X = 0$

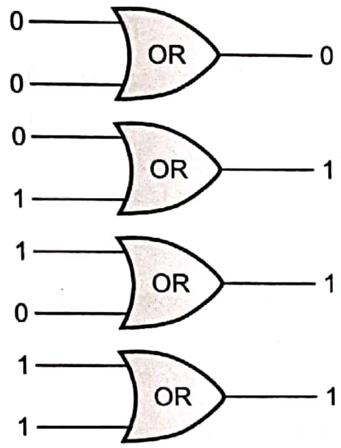
II. OR Relations (*Logical Addition*)

$0 + 0 = 0$

$0 + 1 = 1$

$1 + 0 = 1$

$1 + 1 = 1$



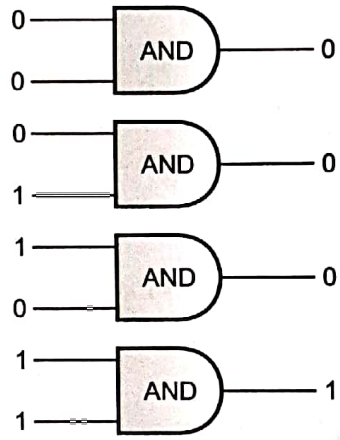
III. AND Relations (*Logical Multiplication*)

$0 . 0 = 0$

$0 . 1 = 0$

$1 . 0 = 0$

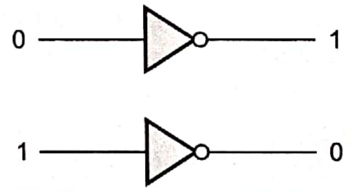
$1 . 1 = 1$



IV. Complement Rules :

$\bar{0} = 1$

$\bar{1} = 0$



14.6 PRINCIPLE OF DUALITY

This is a very important principle used in boolean logic. This states that *starting with a boolean relation, another boolean relation can be derived by* :

1. changing each OR sign (+) to an AND sign (.)
2. changing each AND sign (.) to an OR sign (+)
3. replacing each 0 by 1 and each 1 by 0.

Check Point 14.3

1. Write the dual of : $1 + 1 = 1$
2. Give the dual of the following in Boolean algebra :
   (i) $X . X' = 0$ for each X
   (ii) $X + 0 = X$ for each X.
3. What is the significance of Principle of Duality ?
4. Write dual of the following Boolean Expression :
   (a) $(x + y')$ (b) $xy + xy' + x'y$
   (c) $a + a'b + b'$ (d) $(x + y' + z)(x + y)$

The derived relation using duality principle is called *dual of original expression.*

For instance, we take *postulate II* related to logical addition, which states :

(*a*) $0 + 0 = 0$ (*b*) $0 + 1 = 1$ (*c*) $1 + 0 = 1$ (*d*) $1 + 1 = 1$

Now working according to above guidelines, + is changed to (.) and 0's are replaced by 1's, these become

(*i*) $1 . 1 = 1$ (*ii*) $1 . 0 = 0$ (*iii*) $0 . 1 = 0$ (*iv*) $0 . 0 = 0$

which are nothing but *same as that of postulate III* related to logical multiplication. So *i, ii, iii, iv* are the duals of *a, b, c* & *d*. We'll be applying this duality principle in the theorems of boolean algebra which is our next topic.

14.7 BASIC THEOREMS OF BOOLEAN ALGEBRA/LOGIC

Basic postulates of boolean algebra are used to define *basic theorems of boolean algebra* that provide all the tools necessary for manipulating boolean expressions. Although simple in appearance, these theorems may be used to construct the boolean algebra.

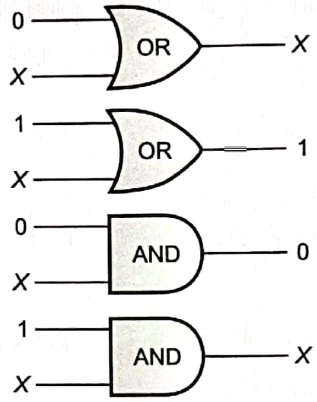
14.7.1 Properties of 0 and 1

(*a*) 0 + X = X (*gate representation of* (*a*))

(*b*) 1 + X = 1 (*gate representation of* (*b*))

(*c*) 0 . X = 0 (*gate representation of* (*c*))

(*d*) 1 . X = X (*gate representation of* (*d*))



Proof.

(a) 0 + X = X

Truth table for above expression is given below in Table 14.15, where *R* signifies the output

Table 14.15 *Truth Table for* $0 + X = X$.

0	X	R
0	0	0
0	1	1

as *X* can have values either 0 or 1 (*postulate I*) both the values ORed with 0 produce the *same output as that of* X. Hence proved

(b) 1 + X = 1

Truth table for this expression is given below in Table 14.16, where *R* signifies the output.

Table 14.16 *Truth Table for* $1 + X = 1$

1	X	R
1	0	1
1	1	1

Again X can have values 0 or 1. Both the values (0 and 1) ORed with 1 produce the output as 1. Hence proved. Therefore $1 + X = 1$ *is a* ***tautology***.

(c) 0 . X = 0

As both the possible values of X (0 and 1) are to be ANDed with 0, so, the truth table for this expression is as follows where (*R* signifies the output)

Table 14.17 *Truth Table for* $0 . X = 0$.

0	X	R
0	0	0
0	1	0

Both the values of X(0 and 1) when ANDed with produce the *output as* 0. Hence proved. Therefore, 0 . X = 0 is a ***fallacy***.

(d) 1 . X = X

Now both the possible values of *X* (0 and 1) are to be ANDed with 1. Thus the truth table for it will be as follows :

Table 14.18 *Truth Table for* $1.X = X$

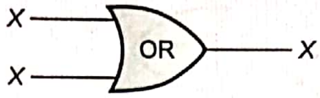
1	X	R
1	0	0
1	1	1

Now observe both the values (0 and 1) when ANDed with 1 produce the *same output as that of* X. Hence proved. Here properties *b* and *c* are duals of each other and properties *a* and *d* are duals of each other.

14.7.2 Indempotence Law

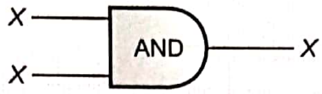
This law states that

(*a*) $X + X = X$ *i.e.,*



(*gate representation for (a)*)

(*b*) $X . X = X$ *i.e.,*



(*gate representation of (b)*)

Proof.

(a) $X + X = X$

To prove this law, we will make truth table for above expression. As X is to be ORed with itself only, we will prepare truth table with the two possible values of X (*i.e.,* 0 and 1).

Table 14.19 *Truth Table for* $X + X = X$

X	X	R
0	0	0
1	1	1

$0 + 0 = 0$ (*ref. postulate II*)

and $1 + 1 = 1$ (*ref. postulate II*)

$\Rightarrow X + X = X$, as it holds true for both values of X. Hence proved.

(b) $X . X = X$

Here X is ANDed with itself. Again we will prepare truth table for this expression taking 2 possible values of X (0 and 1)

Table 14.20 *Truth Table for* $X . X = X$

X	X	R
0	0	0
1	1	1

$0 . 0 = 0$ (*ref. postulate III*)

and $1 . 1 = 1$ (*ref. postulate III*)

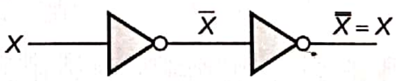
$\Rightarrow X . X = X$, as it holds true for both values of X. Hence proved.

(*a*) and (*b*) are duals of each other.

14.7.3 Involution

This law states that

$(\overline{\overline{X}}) = X$ *i.e.,*



To prove this, again we'll prepare truth table which is given below.

Table 14.21 *Truth Table for* $\overline{\overline{X}} = X$

X	$\overline{X}$	$\overline{\overline{X}}$
0	1	0
1	0	1

First column represents possible values of X, second column represents complement of X (*i.e.,* $\overline{X}$) and the third column represents complement of $\overline{X}$ (*i.e.,* $\overline{\overline{X}}$) which is same as that of X. Hence proved.

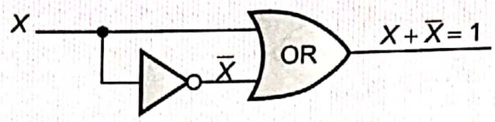
This law is also called *double-inversion rule.*

In Boolean Algebra, if an expression holds *true* then its dual is also *true* and vice-versa.

14.7.4 Complementarity Law

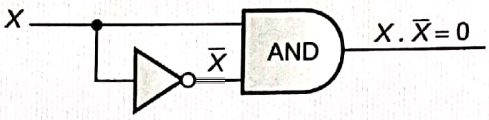
These laws state that

(*a*) $X+\overline{X}=1$



(*gate representation of* (*a*))

and (*b*) $X.\overline{X}=0$



(*gate representation of* (*b*))

Proof.

(*a*) $X+\overline{X}=1$

We will prove $X+\overline{X}=1$ with the help of truth table which is given below :

Table 14.22 *Truth Table for* $X+\overline{X}=1$

X	$\overline{X}$	$X+\overline{X}$
0	1	1
1	0	1

Here, in the first column possible values of X have been taken, second column consists of $\overline{X}$ values (*complement values of* X), X and $\overline{X}$ values are ORed and the output is shown in third column as

$0+1=1$, (*ref. postulate II*)

$1+0=1$ (*ref. postulate II*)

$\Rightarrow X+\overline{X}=1$, as it holds true for both possible values of X.

Hence proved.

It is a *tautology*.

(*b*) $X.\overline{X}=0$

Truth table for this expression is as follows :

Table 14.23 *Truth Table for* $X.\overline{X}=0$

X	$\overline{X}$	$X.\overline{X}$
0	1	0
1	0	0

as $0.1=0$ (*ref. postulate III*)

and $1.0=0$ (*ref. postulate III*)

$\Rightarrow X.\overline{X}=0$, as it holds true for both the values of X. Hence proved. *Observe here* $X.\overline{X}=0$ *is dual of* $X+\overline{X}=1$.

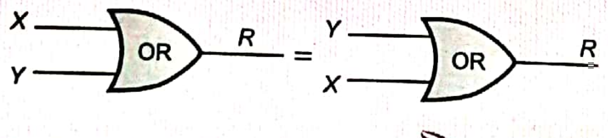
Changing (+) to (.) and 1 to 0, and we get $X.\overline{X}=0$.

It is a *fallacy*.

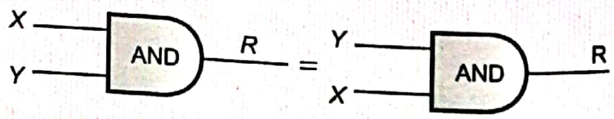
14.7.5 Commutative Law

These laws state that

(*a*) $X+Y=Y+X$



(*b*) $X.Y=Y.X$



(R signifies the output)

Proof.

(a) X + Y = Y + X

Truth Table for $X + Y = Y + X$ is given below :

Table 14.24 *Truth Table for X + Y = Y + X*

X	Y	X+Y	Y+X
0	0	0	0
0	1	1	1
1	0	1	1
1	1	1	1

Compare the columns $X + Y$ and $Y + X$, both of these are identical. Hence proved.

(b) X . Y = Y . X

Truth table for $X . Y = Y . X$ is given below :

Table 14.25 *Truth Table for X .Y = Y .X*

X	Y	X . Y	Y . X
0	0	0	0
0	1	0	0
1	0	0	0
1	1	1	1

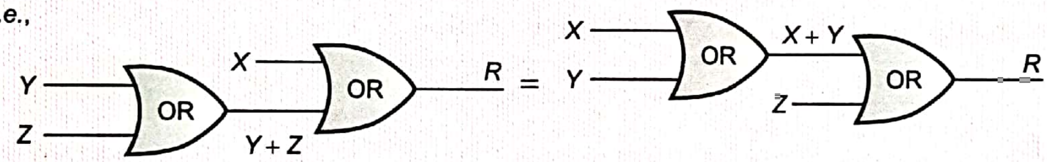
Both of the columns $X . Y$ and $Y . X$ are identical, hence proved.

14.7.6 Associative Law

These laws state that

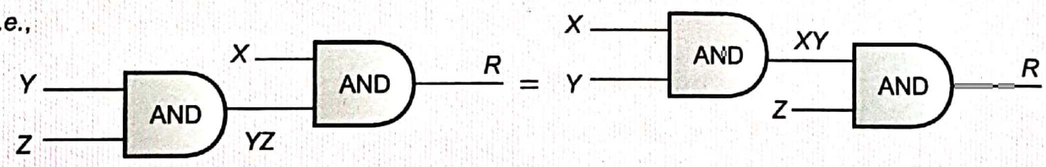
(*a*) $X + (Y + Z) = (X + Y) + Z$

i.e.,



(*b*) $X\,(YZ) = (XY)\,Z$

i.e.,



Proof. (*a*) Truth table for $X + (Y + Z) = (X + Y) + Z$ is given below :

Table 14.26 *Truth Table for X + (Y + Z) = (X + Y) + Z*

X	Y	Z	Y + Z	X + Y	X + (Y + Z)	(X + Y) + Z
0	0	0	0	0	0	0
0	0	1	1	0	1	1
0	1	0	1	1	1	1
0	1	1	1	1	1	1
1	0	0	0	1	1	1
1	0	1	1	1	1	1
1	1	0	1	1	1	1
1	1	1	1	1	1	1

Compare the columns $X + (Y + Z)$ and $(X + Y) + Z$, both of these are identical. Hence proved[1]. Since rule (*b*) is dual of rule (*a*), hence it is also proved.

1. Recall that if a boolean expression is true then its dual is also true.