CHAPTER 4

Logic Design

This chapter deals with Boolean algebra, minimization techniques for boolean expressions, combinational and sequential logic circuits.

4.1 BOOLEAN ALGEBRA

Boolean algebra is an algebra of logic. It is one of the most basic tools to analyze and design logic circuits. It is named after George Boole who developed it in 1854. The original purpose of this algebra was to simplify logical statements and solve logic problems. It had no practical application until 1939 when Shannon applied it to telephone switching circuits. Shannon's work gave an idea that boolean algebra could be applied to computer electronics. Today it is the backbone of design and analysis of computer and other digital circuits.

Now consider the statement "The man is tall". This statement may be either true or false. So it is a logical statement. A symbol can be used to represent a logical statement. For example,

X = the man is tall

Further, 1 is used to represent true, and 0 to represent false. If the above mentioned statement is true, we write X = 1. If the above statement is false we write X = 0. Here 1 and 0 represent only truth and falsehood. They do not represent numerical values. 1 and 0 have no arithmetic significance. Thev only have logical significance.

4.1.1 AND Operation

We have already discussed AND operation in Chapter 3. Here the approach is different i.e. it is from logical statement point of view. Now consider the following statements:

The man is tall (= X)
The man is wise (= Y)

These statements can be written as a compound statement given below:

The man is tall and the man is wise

This compound statement can be written in the symbolic form as follows:

X AND Y

X may be true or false, and Y may be true or false. Therefore, the compound statement X AND Y will be true only when both X and Y are true. Table 4.1 shows all possibilities of truth and falsehood of X and Y, and corresponding truth and falsehood of X AND Y. Such a table is called truth table.

Table 4.1 Truth Table for AND Operation

X		Y	X AND Y
True	AND	True	True
True	AND	False	False
False	AND	True	False
False	AND	False	False

A "." is used to represent AND operation. So X AND Y will be represented as X.Y. Representing true by 1, false by 0 and AND by "." the above table can be replaced by Table 4.2.

Other symbols which are used to represent AND operation are $\wedge$ and $\cap$. In ordinary algebra "." represents multiplication but in boolean algebra it represents only logical AND operation. The rules for AND operation are exactly same as those of simple arithmetic multiplication. This is just a coincidence which enabled us to remember these rules without any additional effort, though the AND operation has nothing to do with the arithmetic multiplication.

Table 4.2 Truth Table for AND Operation in Symbolic Form

X Y		X.Y
1.1	=	1
1.0	=	0
0.1	=	0
0.0	=	0

4.1.2. OR Operation

Now consider the following two basic logical statements:

He will give me a pen (= X)
He will give me a pencil (= Y)

There two statements can be written as a compound statement given below:

He will give me a pen or a pencil or both

This compound statement can be written in the symbolic form as shown below:

X OR Y OR both

This is called inclusive OR. Both is not written in the statement. If simply OR is written, it is understood that it is inclusive OR. X OR Y means X OR Y OR both. Therefore, an inclusive OR is simply written as given below:

X OR Y

There is another connective: exclusive OR. It is written as XOR. It means X OR Y but not both.

Here we shall consider inclusive OR. X may be true or false, and Y may be true or false. The compound statement X OR Y will be true when any one or both statements are true. Table 4.3 shows all possibilities of truth and falsehood of X and Y, and corresponding truth and falsehood of X OR Y.

Table 4.3 Truth Table for OR Operation

X		Y	X OR Y
True	OR	True	True
True	OR	False	True
False	OR	True	True
False	OR	False	False

A "+" is used to represent OR operation. So X OR Y can be written as X + Y. Representing true by 1, false by 0 and OR by "+", the above table can be replaced by Table 4.4.

Table 4.4 Truth Table for OR Operation In Symbolic Notation

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

Other symbols used to represent OR operation are $\vee$ and $\cup$. In ordinary algebra "+" means addition, but in boolean algebra it simply represents logical OR operation. The rules for OR operation are not exactly same as those of arithmetic addition. The 1st rule $1 + 1 = 1$ is not valid for addition. It simply indicates that if both statements are true, the compound statement is true. So we have to remember this particular rule for OR operation. The other three rules for OR operation are exactly same as those for arithmetic addition.

4.1.3. NOT Operation

Now consider the following statement:

The man is wise (assume it is = X)

This statement may be true or false. If this statement is true, the statement given below will be false:

The man is NOT wise (= NOT X = $\overline{X}$)

If the statement "The man is wise" is false, the statement "The man is NOT wise" will be true. Table 4.5 is the truth table for NOT operation.

Table 4.5. Truth Table for NOT Operation

X	NOT X
True	False
False	True

NOT X is represented by $\overline{X}$ or X'. $\overline{X}$ is called the **complement** of X.

If $X = 1, \overline{X} = \overline{1} = 0$

If $X = 0, \overline{X} = \overline{0} = 1$

It also follows that

$$\overline{\overline{X}} = X$$

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

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

Table 4.6 shows the truth table for NOT operation in symbolic form.

Table 4.6. Truth Table for NOT Operation in Symbolic Form

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

4.1.4 Examples of Switches to Illustrate Logic Operations

Electrical switches are very good examples to give clear concept of AND, OR operations and many Boolean theorems. A switch has only two states: either closed or open. These are similar to truth and falsehood of a statement. We can assume closed = 1 and open (off) = 0.

Now consider two switches connected in series as shown in Fig. 4.1. It is a very good example to illustrate AND operation. The bulb will glow only when both the switches X and Y are closed. Table 4.7 shows its truth table.

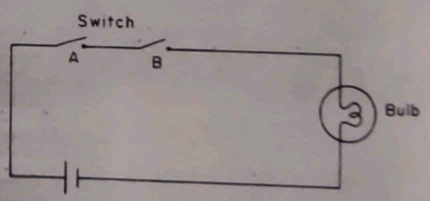

Fig. 4.1 Two switches in series

We can assume closed = 1, open = 0, ON = 1 and OFF = 0, then Table 4.8 will replace the Table 4.7.

Similarly, two switches connected in parallel as shown in Fig. 4.2 is an example to illustrate OR operation. The bulb will glow when either or both switches are on. Table 4.9 shows its truth table.

Table 4.7. The Behaviour of Two Switches Connected in Series

Switch X	Switch Y	Bulb B
Closed	Closed	ON
Closed	Open	OFF
Open	Closed	OFF
Open	Open	OFF

Table 4.8. Truth Table for Two Switches Connected in Series

Switches		Bulb B
X	Y	B = X.Y
1	1	1
1	0	0
0	1	0
0	0	0

Table 4.9. Behaviour of Two Switches connected in Parallel

Switch X	Switch Y	Bulb B
Closed	Closed	ON
Closed	Open	ON
Open	Closed	ON
Open	Open	OFF

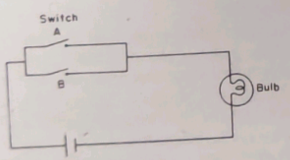

Fig. 4.2 Two switches in parallel

Table 4.10 shows the truth table in symbolic form for two switches in parallel.

Table 4.10. Truth Table for Two Switches in parallel

Switches		Bulb
X	Y	B = X + Y
1	1	1
1	0	1
0	1	1
0	0	0

Now consider the following:

A switch is closed = A
A switch is NOT closed, i.e. open = $\overline{A}$

4.1.5 Boolean Variables, Operations and Functions

A computer is a binary digital system. Such a system operates on electrical signals which have only two possible states: HIGH (5 volts) and LOW (0 volt). These high and low states are represented by 1 and 0 respectively. A signal that does not change its state (or value) in time is called a **constant** signal. The value of a constant signal will always remain same: either 1 or 0. On the other hand, a signal which changes its state in time is known as a **variable** signal. The value of a variable signal may be 1 at some times and 0 at some other times. For design and analysis of digital systems a constant signal and a variable signal will be treated just as a constant and as a variable respectively. Thus the variables which have only two values 1 and 0, are called **Boolean variables** (or logic variables). These variables may be denoted by A, B, C, X, Y, Z etc. There are only two Boolean constants 0 and 1. In ordinary algebra a variable can have very large (or even infinite) number of values, but in Boolean algebra they can have only one of the two possible values, 0 and 1. Thus, Boolean algebra becomes much simpler than ordinary algebra.

The only algebraic operations (or functions) used in Boolean algebra are AND, OR and NOT. All other functions can be expressed in terms of these basic functions. For example, NAND operation is the combination of AND followed by NOT. Similarly, NOR is the combination of OR followed by NOT.

In ordinary algebra we have the concept of expression or function. Similarly, in Boolean algebra, we have the concept of expression or function. A Boolean function or expression consists of Boolean variables. Consider the following example:

$$X = A + B.C + C.(\overline{D} + E)$$

In the above equation the variable X is the function of A, B, C, D, and E. This can be written as

$$X = f(A, B, C, D, E).$$

A, B, C, D and E are Boolean variables. The right hand side of the above equation is known as an *expression*. Each occurrence of a variable or its complement in an expression is called *literal*. In the above expression there are five variables and six literals. Generally, "." is not written in an expression. It can be written where additional clarity is required. Boolean expression (or Boolean function) is also called logic expression (or logic function).

4.1.6. Boolean Postulates

Fundamental conditions or self-evident propositions are called postulates. The postulates for Boolean algebra originate from the three basic logic functions—AND, OR and NOT. The properties of these basic functions as given in Tables 4.2, 4.4 and 4.6 are the postulates for Boolean Algebra. These are called Boolean postulates and they are summarized in Table 4.11. These postulates define the operation of the AND, OR and NOT functions. In other words these are the results of these basic operations.

Table 4.11 Boolean Postulates

(1)	$0.0=0$	Derived from AND operation
(2)	$0.1=0$	
(3)	$1.0=0$	
(4)	$1.1=1$	
(5)	$0+0=0$	Derived from OR operation
(6)	$0+1=1$	
(7)	$1+0=1$	
(8)	$1+1=1$	
(9)	$\overline{0}=1$	Derived from NOT operation
(10)	$\overline{1}=0$	

4.1.7. Boolean Theorems

With the help of Boolean postulates many useful theorms known as Boolean theorems have been derived. These theorems are very useful in simplifying logical expressions or transforming them into other useful equivalent expressions. Table 4.12 presents the list of Boolean theorems.

In Boolean algebra, it is possible to test the validity of the theorems by substituting all possible values of the variables because the variables have only two values, viz. 0 and 1. This technique of proving theorems is known as *proof by perfect induction.*

Table 4.12 Boolean Theorems

(1)	$0.X=0$	Properties of AND operation
(2)	$X.0=0$	
(3)	$1.X=X$	
(4)	$X.1=X$	
(5)	$X+0=X$	Properties of OR operation
(6)	$0+X=X$	
(7)	$X+1=1$	
(8)	$1+X=1$	
(9)	$X.X=X$	Combining a variable with itself or its complement
(10)	$X.\overline{X}=0$	
(11)	$X+X=X$	
(12)	$X+\overline{X}=1$	
(13)	$\overline{\overline{X}}=X$	Double complementation
(14)	$X+Y=Y+X$	Commutative laws
(15)	$X.Y=Y.X$	
(16)	$X.(Y.Z)=(X.Y).Z=X.Y.Z$	Associative laws
(17)	$(X+Y)+Z=X+(Y+Z)=X+Y+Z$	
(18)	$X.(Y+Z)=X.Y+X.Z$	Distributive laws
(19)	$X+Y.Z=(X+Y).(X+Z)$	
(20)	$X+XY=X$	Absorption
(21)	$X(X+Y)=X$	
(22)	$XY+X\overline{Y}=X$	
(23)	$(X+Y)(X+\overline{Y})=X$	
(24)	(a) $X+\overline{X}Y=X+Y$	
	(b) $XZ+Z\overline{X}Y=ZX+ZY$	
(25)	$X(\overline{X}+Y)=XY$	
(26)	$(Z+X)(Z+\overline{X}+Y)=(Z+X)(Z+Y)$	
(27)	$XY+\overline{X}Z+YZ=XY+\overline{X}Z$	
(28)	$(X+Y)(\overline{X}+Z)(Y+Z)=(X+Y)(\overline{X}+Z)$	
(29)	$XY+\overline{X}Z=(X+Z)(\overline{X}+Y)$	
(30)	$(X+Y)(\overline{X}+Z)=XZ+\overline{X}Y$	
(31)	$\overline{X.Y.Z\ldots}=\overline{X}+\overline{Y}+\overline{Z}+\ldots$	DeMorgan's Theorem
(32)	$\overline{X+Y+Z+\ldots}=\overline{X}.\overline{Y}.\overline{Z}\ldots$	

Theorem 27.

$$XY+\overline{X}Z+YZ=XY+\overline{X}Z$$

Proof. $XY+\overline{X}Z+YZ=XY+\overline{X}Z+YZ(X+\overline{X})$

$$=XY+\overline{X}Z+YZX+YZ\overline{X}$$

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

$$=XY+\overline{X}Z$$

Theorem 31.

$\overline{X.Y}=\overline{X}+\overline{Y}$ (Taking only two variables).

Proof. This theorem can easily be proved by truth table approach. For all values of X and Y, we calculate $\overline{X.Y}$ and $\overline{X}+\overline{Y}$, and tabulate the result as shown in Table 4.14.

Table 4.14.

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

From the above truth table we conclude that

$$\overline{X.Y}=\overline{X}+\overline{Y}$$

Theorem 32.

$\overline{X+Y}=\overline{X}.\overline{Y}$ (Takingonly two variables)

Proof. To prove this theorem we can prepare the truth table as shown in Table 4.15.

Table 4.15.

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

From the above truth table we conclude that

$$\overline{X+Y}=\overline{X}.\overline{Y}$$

4.1.8 Simplification of Boolean Expression by Algebraic Method

Boolean theorems are very useful tools for simplifying logical expressions. Some examples of simplification are given below.

Example 1. Simplify the logical expression

$$X\overline{Y}\,\overline{Z}+X\overline{Y}\,\overline{Z}W+X\overline{Z}$$

The above expression can be written as

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

$=X\overline{Y}\,\overline{Z}+X\overline{Z}$, as $1+W=1$ by Theorem 8.

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

$=X\overline{Z}$, as $\overline{Y}+1=1$, by Theorem 8.

Example 2. Simplify the Boolean expression

$$X+\overline{X}Y+\overline{Y}+(X+\overline{Y})\overline{X}Y$$

The above expression can be written as

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

$=X+\overline{X}Y+\overline{Y}$, as $X\overline{X}=0$, $X\overline{X}Y=0$,

as $\overline{Y}Y=0$, $\overline{Y}\,\overline{X}Y=0$.

$=X+Y+\overline{Y}$, as $X+\overline{X}Y=X+Y$ by Theorem 24 (a).

$=X+1$, as $Y+\overline{Y}=1$ by Theorem 12.

$=1$, by Theorem 7.

Example 3. Simplify the logical expression

$$Z(Y+Z)(X+Y+Z)$$

The above expression can be written as

$(ZY+ZZ)(X+Y+Z)$

$=(ZY+Z)(X+Y+Z)$, as $ZZ=Z$ by Theorem 9.

$=Z(X+Y+Z)$, as $Z+ZY=Z$, by Theorem 20

$= ZX + ZY + ZZ$

$= ZX + ZY + Z,$ as $ZZ = Z,$ by Theorem 9.

$= ZX + Z,$ as $Z + ZY = Z,$ by Theorem 20.

$= Z\ ,$ as $Z + ZX = Z,$ by Theorem 20.

Example 4. Simplify the logical expression

$$\overline{X}\,\overline{Y} + \overline{X}Z + YZ + \overline{Y}Z\overline{W}$$

At first glance it looks that it can not be reduced further. But applying a trick it can be reduced as follows:

The above expression can be written as

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

$= \overline{X}\,\overline{Y} + \overline{X}Z(Y + \overline{Y}) + YZ + \overline{Y}Z\overline{W}$

$= \overline{X}\,\overline{Y} + \overline{X}ZY + \overline{X}Z\overline{Y} + YZ + \overline{Y}Z\overline{W}$

$= \overline{X}\,\overline{Y}(1 + Z) + YZ(\overline{X} + 1) + \overline{Y}Z\overline{W}$

$= \overline{X}\,\overline{Y} + YZ + \overline{Y}Z\overline{W}$

Example 5. Simplify the expression $(X + Y)(\overline{X} + Z)(Y + Z)$

The above expression can be written as

$(X\overline{X} + XZ + Y\overline{X} + YZ)(Y + Z)$

$= (XZ + Y\overline{X} + YZ)(Y + Z),$ as $X\overline{X} = 0$

$= XZY + YY\overline{X} + YYZ + XZZ + Y\overline{X}Z + YZZ$

$= XZY + Y\overline{X} + YZ + XZ + Y\overline{X}Z + YZ$

Rearranging the terms we get

$XZY + XZ + Y\overline{X} + Y\overline{X}Z + YZ,$ as $YZ + YZ = YZ$

$= XZ(Y + 1) + Y\overline{X} + YZ(\overline{X} + 1)$

$= XZ + Y\overline{X} + YZ$

Now it seems that it can not be reduced further. But apply the following trick:

The above expression can be written as

$XZ + Y\overline{X} + YZ(X + \overline{X}),$ as $X + \overline{X} = 1$

$= XZ + Y\overline{X} + YZX + YZ\overline{X}$

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

$= XZ + Y\overline{X}$

4.1.9 Dual and Complement of a Boolean Expression

Two expressions are called *equivalent* only when both are equal to 1 or equal to 0. Two expressions are *complements* of each other if one expression is equal to 1 while the other is equal to 0, and vice versa.

To obtain the complement of a Boolean expression the following changes are made:

(i) all . signs are changed to + signs
(ii) all + signs are changed to . signs
(iii) all 1s are changed to 0s
(iv) all 0s are changed to 1s
(v) all literals are complemented.

Example. Find complement of $1.X + \overline{Y}Z + 0$

The complement of the above expression will be

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

The *dual* of a Boolean expression is obtained by performing the following operations:

(i) all . signs are changed to + signs
(ii) all + signs are changed to . signs
(iii) all 1s are changed to 0s
(iv) all 0s are changed to 1s

In finding a dual of an expression literals are not complemented. The following examples illustrate the principle. There is no general rule for the values of dual expressions. Both expressions may be equal to 1 or both may be equal to 0. One may be equal to 1 while the other is equal to 0.

Example 1. Find the dual of the Boolean expression

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