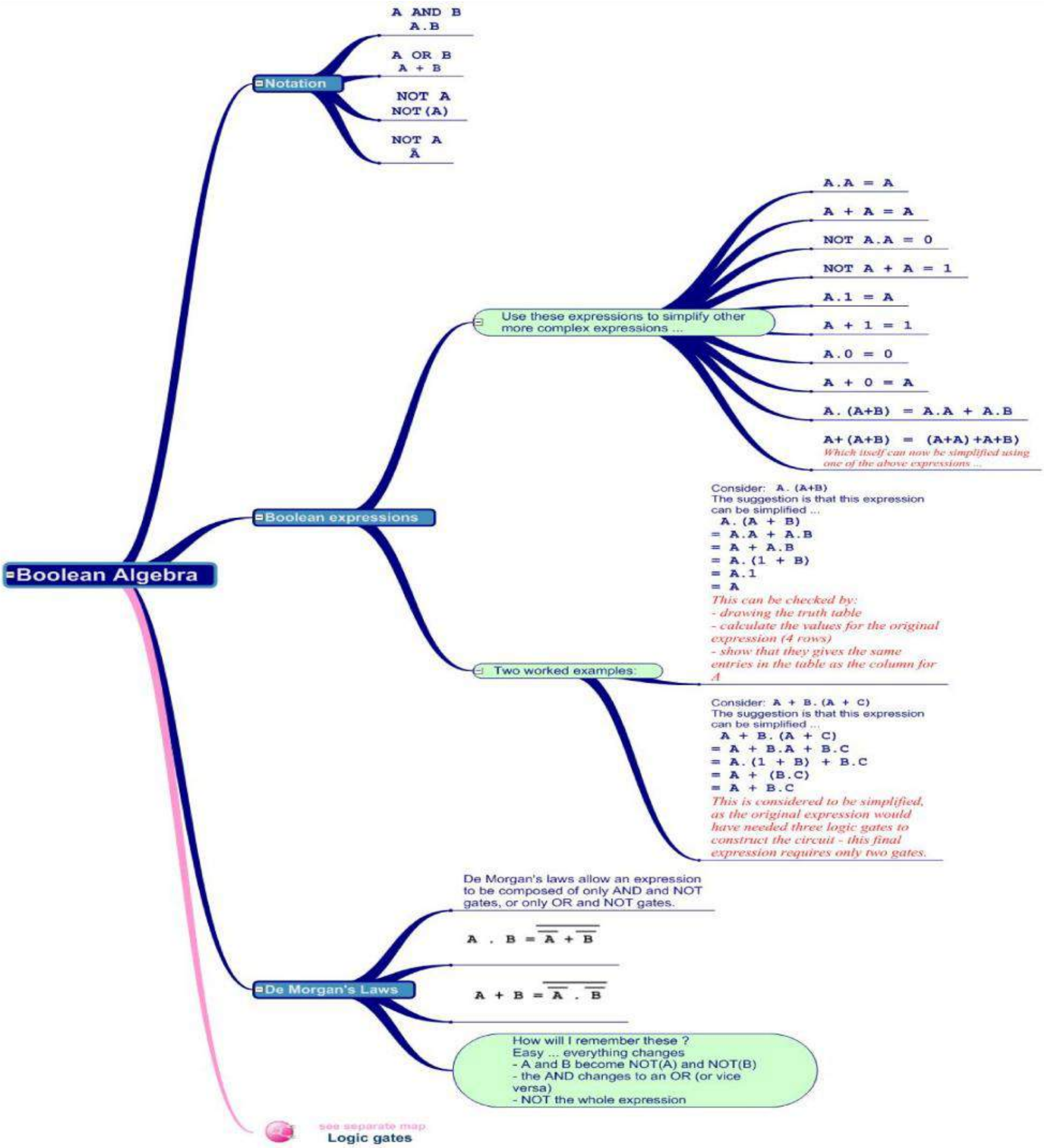
Chatpter 2

Boolean Algebra

Objectives:

- **To understand the concept of boolean algebra**
- **To understand the concept of simplifications of boolean expressions**



2.1 Introducton to Boolean Algebra

In the previous course, we have seen that computers normally use binary numbers. In this chapter, you will learn about an algebra that deals with the binary number system. This algebra, known as Boolean algebra, is very useful in designing logic circuits used by the processors of computer systems. In addition to this, you will also learn about the elementary logic gates that are used to build up circuits of different types to perform the necessary arithmetic operations. These logic gates are the building blocks of all the circuits in a computer. Finally, in this chapter, we will also learn how to use Boolean Algebra to design simple logic circuits frequently used by the arithmetic logic unit of almost all computers.

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'.

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 Algebra to solve relay problems. As logic problems are binary decisions and Boolean Algebra effectively deals with these binary values. Thus it is also called 'Switching Algebra'.

2.2 Binary Valued Quantities - Variable and Constants

Everyday we have to make logic decisions. For example, consider the following questions:

"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 the answer YES or NO. These are the only two possible answers.

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

A variable used in an algebraic formula is generally assumed that the variable may take any numerical value through the entire field of real numbers. However a variable used in Boolean Algebra or Boolean equation can have only one of two possible values. The two values are FALSE (or 0) and TRUE (or 1). 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 respectively. 1 means TRUE and 0 means FALSE. 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 1 or 0 (TRUE or FALSE).

The decision which results into either YES (TRUE or 1) or NO (FALSE or 0) is called a Binary Decision.

Also, if an equation describing logical circuitry has several variables, it is still understood that each of the variables can assume only the values 0 and 1. For instance, in the equation A + B = C , each of the variables A , B and C may have only the values 0 or 1.

2.3.0 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 operations.

2.3.1 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 +, -, x, / to form algebraic expressions.

For example, 2 x A + 3 x B – 6 = (10 x Z) /2 x Y i.e., 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.

These logical operators are also used to combine logical variables and logical constants to form logical expressions.

For example, assuming that x, y and z are logical variables, the logical expressions are

X NOT Y OR Z

Y AND X OR Z

2.3.2 Logical Operators

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

Before we start discussion about logical operators, let us first understand what a Truth Table is ?. Logical statements can have only one of the two values TRUE (YES or 1) or FALSE (NO or 1).

For example, if X and Y are the logical statements and R is the result, then the truth table can be written as follows:

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

Table 1.1

If result of any logical statement or expression is always TRUE or 1, it is called **Tautology** and if the result is always FALSE or 0 it is called **Fallacy**.

1 represents TRUE value and 0 represents FALSE value.

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

Now let us proceed with our discussion about logical operators. There are three logical operators: NOT, OR and AND Operators.

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 $\overline{\ }$ (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.

$\overline{0} = 1$ or NOT (FALSE) = TRUE and

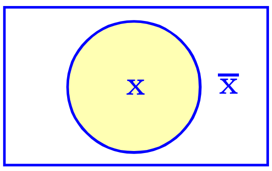
$\overline{1} = 0$ or NOT (TRUE) = FALSE and

The truth table for the NOT operator is

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

Several other symbols like '~' are also used for the complementation symbol. If ~ is used then ~X is read as 'negation of X' and if symbol ' is used then X' is read as complement of X.

Table 1.2 Truth Table for NOT operator



NOT operation is singular or unary operation as it operates on single variable.

Venn diagram for $\overline{x}$ is given above where shaded area depicts $\overline{x}$.

Figure 1.3. Venn diagram for $\overline{x}$

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 mean arithmetic addition, 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 = 1

0 + 1 = 1

1 + 0 = 1

1 + 1 = 1

The truth table of OR operator is given below:

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

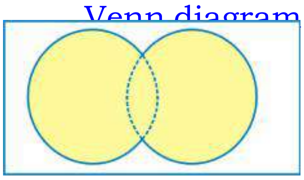
Note that when any one or both X and Y is 1, X + Y is 1.

and both X and Y is 0, X+Y is 0

Table 1.4 : Truth Table for OR operator

To avoid ambiguity, there are other symbols e.g., ∪ and ∨ 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 below where the shaded area depicts X + Y.



Shaded portion shows X + Y

Figure 1.2 : Venn diagram for 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

And the truth table for AND is as follows :

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

Table 1.10 : Truth Table for AND operator

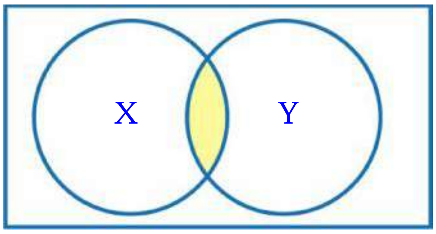


FIGURE 1.3 : Venn Diagram for (X.Y)

Note that only when both X and Y are 1's, X.Y has the result 1. If any one of X and Y is 0, XY result 0. Venn diagram for X.Y is given in the figure above where the shaded area depicts X.Y

2.4.0 Evaluation of Boolean Expressions Using Truth Table

Logical variables are combined by means of logical operators AND, OR and 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 variable can assume the value 0 or 1. The possible combinations of values may be arranged in ascending order as in Table 1.11

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, and Z are the three (3) variables in total. A truth table involving 3 input variables will have 2^3 = 8 rows or combinations in total. The left most column will have half of total entries (4 entries) as zeroes and half as 1's (in total 8). The next column will have number of 0's and 1's halved than first column completing 8 rows and so on. That is why, first column has four 0's and four 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 followed by one 1 completing 8 rows in total.

Table 1.11 Possible Combinations of X, Y and Z

So a column is added to list Y.Z (Table 1.12)

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

AND operation is applied only on columns Y and Z.

Table 1.12 Truth Table for (Y.Z)

One more column is now added to list the values of $\overline{YZ}$ (Table 1.13)

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.

Table 1.13 *truth.table.for.Y.Z.and..*$\overline{YZ}$

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 1.14).

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.

Table 1.14 Truth Table for $X + \overline{YZ}$.

A Boolean expression will be evaluated using precedence rules. The order of evaluation of an expression is called as precedence. The precedence is, firstly NOT, then AND and then OR. If there is parenthesis, then the expression in parenthesis is evaluated first.

Example 1.13: In the Boolean algebra, verify using truth table that X+XY = X for each X, Y in 0 and 1.

As the expression X+XY=X is a two variable expression, so we require four possible combinations 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 1.14: In the Boolean Algebra, verify using truth table that

$\overline{X+Y} = \overline{X} . \overline{Y}$ in 0 and 1.

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

X	Y	X+Y	$\overline{X+Y}$	$\overline{X}$	$\overline{Y}$	$\overline{X}.\overline{Y}$
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 $\overline{X+Y}$ and $\overline{X} . \overline{Y}$ both the columns are identical, hence verified.

Example 1.15: Prepare a table of combinations for the following Boolean algebra 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

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

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

Example 1.16 Prepare truth table for the following Boolean algebra expressions:

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

Solution (a) Truth table for X(+) +X 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 $X\overline{Y}$ $(\overline{Z}+Y)$+ 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

2.4.1 BASIC LOGIC GATES

After Shannon applied Boolean algebra 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?

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 algebra.

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

There are three types of logic gates:

- NOT gate or Inverter
- OR gate
- AND gate

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 complement (opposite) of the input. Following tables summarizes the operation:

X	$\overline{X}$
Low	High
High	Low

Table 1.15 Truth Table for NOT gate

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

Table 1.16 Alternative truth table for NOT gate

A low input or 0 produces high output or 1 and vice versa. The symbol for inverter is given in adjacent Fig. 1.4.

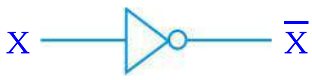


Fig. 1.4. Not gate symbol

NOT Gate is a gate or an electronic circuit that accepts only one input and produces one output signal. The output state is always the complement of the input state.

OR Gate

The OR Gate has two or more input signals, but only one output signal. This gate gives the logical addition of the inputs. If any of the input signals or both is 1 (high), the output signal is 1 (high). The output will be low if all the inputs are low.

An OR gate can have as many inputs as desired. No matter how many inputs are there, the action of OR gate is the same.

The OR gate has two or more input signals, but only one output signal. The out will be the logical addition of the inputs.

Following tables show OR action

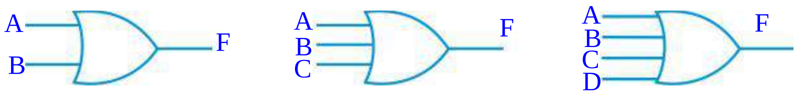
X	Y	F
0	0	0
0	1	1
1	0	1
1	1	1

Table : F=X+Y

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

Table : F=X+Y+Z

The symbol for OR gate is given below:



a) 2 input OR gate b) 3 input OR gate c) 4 input OR gate

Figure 1.5

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 or high only then the output is 1, otherwise output is 0 only.

If any one or all 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 as desired.

The AND Gate has two or more input signals, but only one output signal. The out will be the logical multiplication of the inputs.

Following tables illustrate AND action.

X	Y	A.B
0	0	0
0	1	0
1	0	0
1	1	1

Table 1.19 Two input AND gate

X	Y	Z	X.Y.Z
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

Table 1.20 Three input AND gate

The symbol for AND is

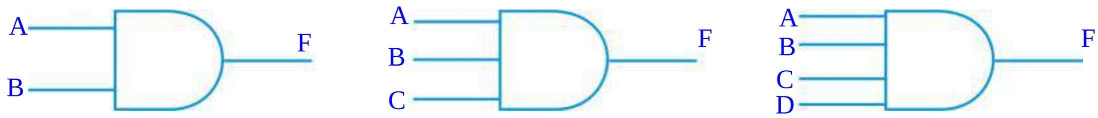


Figure 1.6 (a) 2-input AND gate (b) 3-input AND gate (c) 4 input AND gate

2.5 BASIC POSTULATES OF BOOLEAN ALGEBRA

Boolean algebra is a system of mathematics and consists of fundamental laws. These fundamental laws 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 algebra. These postulates state the basic relations in Boolean algebra:

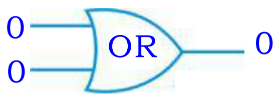
The fundamental laws of the Boolean algebra are called as the postulates of Boolean algebra

The Boolean postulates are:

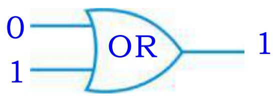
I. If $X \neq 0$ then $X = 1$; and If $X \neq 1$ then $X = 0$

II. OR Relations (Logical Addition)

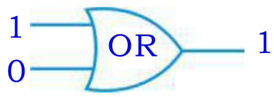
0+0=0



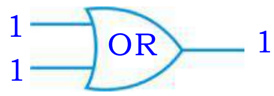
0+1=1



1+0=1



1+1=1



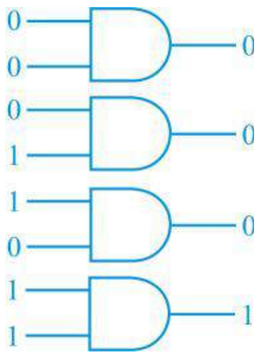
III AND Relations (Logical Multiplication)

$0.0 = 0$

$0.1 = 0$

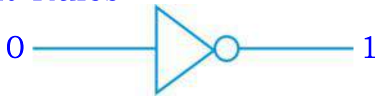
$1.0 = 0$

$1.1 = 1$

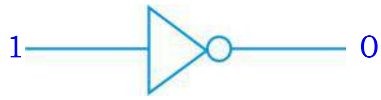


IV Complement Rules

$\overline{0} = 1$



$\overline{1} = 0$



PRINCIPLE OF DUALITY

This is a very important principle used in Boolean algebra. This states that starting with a Boolean relation another Boolean relation can be derived by

i. Changing each OR sign (+) to an AND sign (.)

ii. Changing each AND sign (.) to an OR sign (+)

iii. Changing each 0 by 1 and each 1 by 0.

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

For instance, we take postulates of OR relation, which states that

(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 '.' 0's are replaced by 1's and 1's are replaced by 0's, these equations become

(i) $1.1=1$ (ii) $1.0=0$ (iii) $0.1=0$ (iv) $0.0=0$

These are nothing but postulate III related to AND relations. We'll be applying this duality principle in the theorems of Boolean algebra.

Basic theorems of Boolean algebra

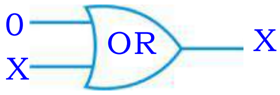
Basic postulates of Boolean algebra are used to define basic theorems of Boolean algebra that provide all the tools necessary for manipulating Boolean expressions. Although simple in appearance, these theorems may be used to construct the Boolean algebra expressions.

Boolean theorems can be proved by substituting all possible values of the variables that are 0 and 1. This technique of proving theorems is called as **proof by perfect induction**. Boolean theorems can also be proved using truth table also.

> Proof by perfect induction is a method of proving Boolean theorems by substituting all possible values of the variables.

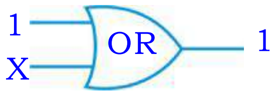
2.5.1 Properties of 0 and 1

a) 0+X=X



(gate representation of (a))

b) 1+X=1

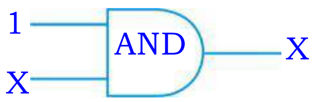


(gate representation of (b))

c) 0.X=0

(gate representation of (c))

d) 1.X=X



(gate representation of (d))

Proof a) 0+x = x

If x = 0, then LHS = 0 + x
= 0 + 0
= 0 { By OR relation }
= x
= RHS

If x = 1, then LHS = 0 + x
= 0 + 1
= 1 { By OR relation }
= x
= RHS

Thus, for every value of x, $0 + x = x$ always.

O	x	R=0+x
0	0	0
0	1	1

Truth table for above expression is given in table 1.21, where R signifies the output.

Table 1.21 Truth Table for 0 + x = x

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

(a) 1 + x = 1

Proof: If $x = 0$, LHS $= 1 + x$

$= 1 + 0$

$= 1$ { By OR relation }

If $x = 1$, LHS $= 1 + x$

$= 1 + 1$

$= 1$ { By OR relation }

Thus, for every value of x, $1 + x = 1$ always.

Truth table for above expression is given below in Table 1.21, where R signifies the output or result.

1	x	1 + x
1	0	1
1	1	1

Table 1.22 Truth Table for 1 + x = 1

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

(a) 0.X = 0

Proof: If $x = 0$, LHS $= 0.x$

$= 0.0$

$= 0$ { By AND relation }

$=$ RHS

If x = 1, LHS = 0.x

= 0.1

= 0 { By AND relation }

= RHS

Thus, for every value of x, 0.x = 0 always.

As both the possible values of X (0 and 1) are to be ANDed with 0, produce the output as 0. The truth table for this expression is as follows:

0	X	R=0.X
0	0	0
0	1	0

Table 1.23 Truth Table for 0.X = 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

Proof: If x = 0, LHS = 1.x

= 1.0

= 0 { By AND relation }

= x

=RHS

If x = 1, LHS = 1.x

= 1.1

= 1 { By AND relation }

= y

=RHS

Thus, for every value of x, 1.x = x always.

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

1	X	1.X
1	0	0
1	1	1

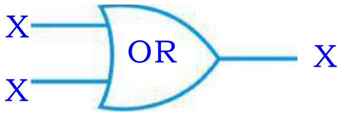
Table 1.24 : Truth Table for 1.X=X

Now observe both the values (0 and 1) when ANDed with 1 produce the same output as that of X. Hence proved.

2.5.2 Indempotence Law

This law states that when a variable is combines with itself using OR or AND operator, the output is the same variable.

a) X + X = X



(gate representation for (a))

b) X . X = X

(gate representation for (b))

Proof :

(a) X + X = X

If x = 0, consider LHS = x + x

= 0 + 0

= 0 { By OR relation }

= x

=RHS

If x = 1, consider LHS = x + x

= 1 + 1

= 1 { By OR relation }

= x

=RHS

Thus, for every value of x, x + x = x always.

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 (0 and 1).

X	X	X+X
0	0	0
1	1	1

Table 1.25 Truth Table for x + x = x

(b) X.X = X

Here X if ANDed with itself.

Proof: If x = 0, consider LHS = x.x

= 0.0

= 0 { By AND relation }

= x

=RHS

If x = 1, LHS = x.x

= 1.1

= 1 { By AND relation }

= x

=RHS

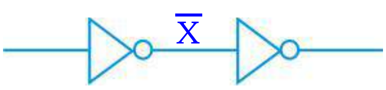
Thus, for every value of x, x + x = x always.

x	x	x.x
0	0	0
1	1	1

Table 1.26 Truth Table for X.X = X

2.5.3 Involution

This law states that the complement of a variable is complemented again, we get the same variable.

$\overline{\overline{(X)}} = X$ ie., X 

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

Proof: If x = 0, then $\overline{x}$ =1 and $\overline{\overline{(x)}} = \overline{1} = 0 = x$

If x = 1, then $\overline{x}$ = 0 and $\overline{\overline{(x)}} = \overline{0} = 1 = x$

Thus, if a variable is complemented twice, we get the same variable.

We'll prepare truth table which is given below:

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

Table 1.27 Truth Table for $\overline{\overline{X}}=X$

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.

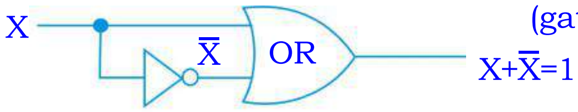
This law is also called double-inversion rule.

2.5.4 Complementarity Laws

Here, we will combine a variable with its complement.

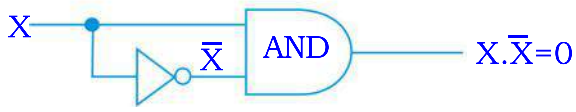
i. These laws states that

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



(gate representation of (a))

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



Proof: If $x = 0$, LHS $= x + \overline{x}$

$= 0 + 1$ $(\overline{x} = 1)$

$= 1$ { By OR relation }

= RHS

If $x = 1$, LHS $= x + \overline{x}$

$= 1 + 0$

$= 1$ { By OR relation }

= RHS

Thus, for every value of x, $x + \overline{x} = 1$ always.

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

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

Table 1.28 Truth Table for $X + \overline{X} = 1$

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 the equation holds true for both possible values of X, it is a **tautology.**

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

Proof: If $x = 0$, LHS $= x . \overline{x}$

$= 0 . 1$ ($\overline{x} = 1$)

$= 0$ { By AND relation }

If $x = 1$, LHS $= x . \overline{x}$

$= 1 . 0$ ($\overline{x} = 0$)

$= 0$ { By AND relation }

Thus, for every value of x, $x . \overline{x} = 0$ always.

Truth table for the expression is as follows:

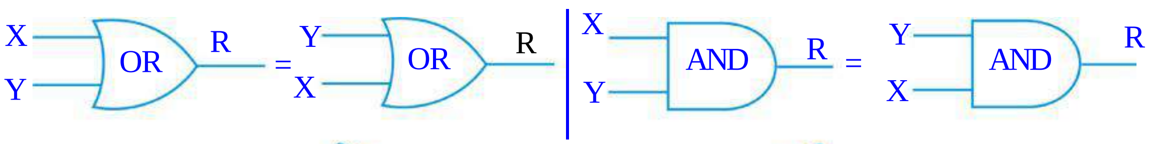
X	$\overline{X}$	$X.\overline{X}$
0	1	0
1	0	0

Table 1.29 Truth table for $X . \overline{X} = 0$

The equations X.X = 0 as it holds true for both the values of X. Hence proved. Observe that X.X=0. It is a fallacy. It is the dual of $X\overline{+}X=1$.

2.5.5 Commutative Law

These laws state that a) $x + y = y + x$ and b) $x . y = y .x$



(R signifies the output)

If x = 0 then LHS = x + y

= 0 + y

= y

RHS = y + x

= y + 0

= y

Therefore, for x = 0, x + y = y + x

If x = 1 then LHS = x + y

= 1 + y

= 1

RHS = y + x

= y + 1

= 1

Therefore, for x = 1, x + y = y + x. Hence the proof.

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

Table 1.30 Truth Table for X + Y = Y +X

Compare the columns X + Y and Y +X, both of these are identical. Hence also proved by truth table.

(b) Truth Table for X . Y = Y . X is given below:

Proof: If x = 0 then LHS = x . y

= 0 . y

= 0

RHS = y . x

= y . 0

= 0

Therefore, for x = 0, x + y = y + x

If x = 1 then LHS = x . y

= 1 . y

= y

Therefore, for x = 1, x + y = y + x. Hence the proof.

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

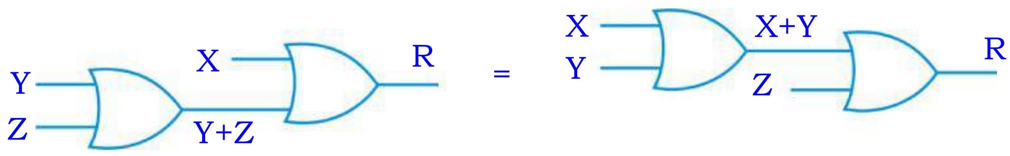
Table 1.31 Truth Table for X . Y = Y . X

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

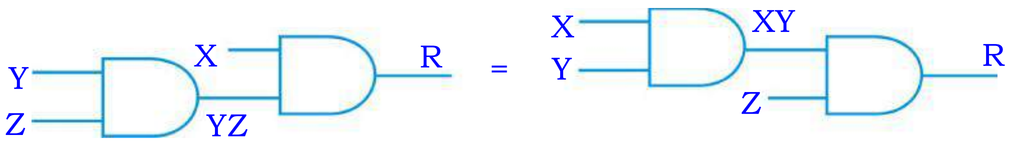
2.5.6 Associative Law

These laws state that

(a) X + (Y + Z) = (X + Y) + Z (associative Law of addition)



(b) X (Y Z) = (X Y) . Z (associative Law of multiplication)



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

Proof: If X = 0 the LHS = X + (Y + Z)

= 0 + (Y+ Z)

= Y+Z

RHS = (X+Y)+Z

= (0+Y)+Z

= Y+Z

Therefore for X=0, X+(Y+Z) = (X+Y)+Z

If X=1, then LHS =X+(Y+Z) RHS = (X+Y)+Z

=1+(Y+Z) =1+(Y+Z)

= 1 =1+Z

Therefore X=1, X+(Y+Z) = (X+Y)+Z = 1

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

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

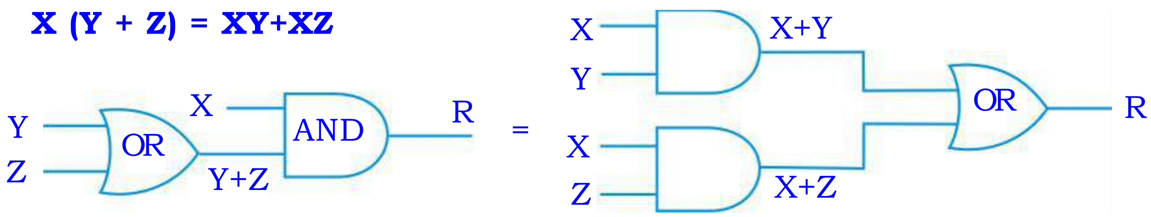
(a) Table 1.32 Truth Table for X + (Y + Z) = (X + Y) + Z

Compare the columns X+(Y+Z) and (X+Y)+Z, both of these are identical. Hence proved. Note : Give proof with table for rule (b). Since rule (b) is a dual of rule (a), hence it is also proved.

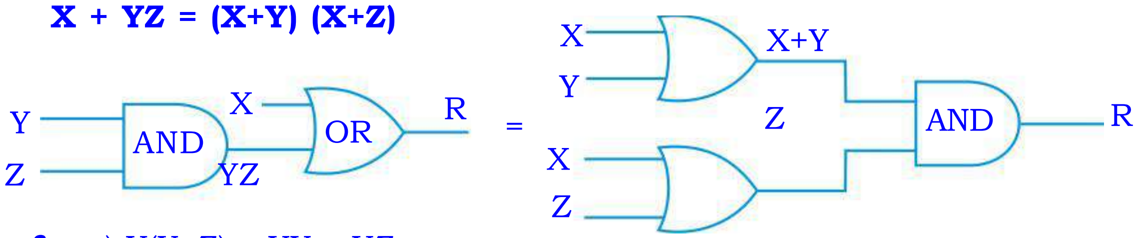
2.5.7 Distributive Law

This law states that

(a) X (Y + Z) = XY+XZ



(a) X + YZ = (X+Y) (X+Z)



Proof: a) X(Y+Z) = XY + XZ

If X=0, LHS = X(Y+Z) = 0(Y+Z) = 0 RHS = XY + XZ = 0.Y + 0.Z = 0 + 0 = 0

If X=1, LHS = X(Y+Z) = 1(Y+Z) = Y+Z RHS = XY + XZ = 1.Y + 1.Z = Y + Z

Therefore, for every value of x, LHS = RHS. i.e., x(y+z) = xz + yz

Truth Table for X (Y + Z) = XY+XZ is given below:

Table 1.33 Truth table for X (Y + Z) = XY+XZ

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

Both the columns X(Y+Z) and XY+XZ are identical, hence proved.

Note : Since rule (b) is dual of rule (a), hence it is also proved

(b) X + YZ = (X+Y) (X+Z)

Proof: RHS = (X+Y) (X+Z)

= XX + XZ +XY + YZ

= X+XZ+XY+YZ ($\therefore$XX=X)

= X(1 + Z + Y) + YZ

= X + YZ (1 + z + y = 1)

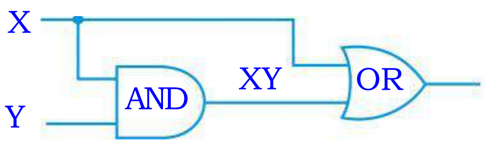
= LHS Hence the proof

Truth table for X+YZ = (X+Y)(X+Z) is given below

2.5.8 Absorption Law

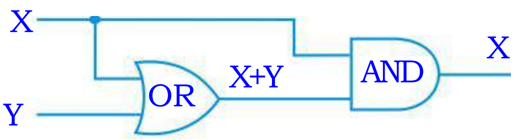
According to this law

a) X+XY=X



Logic diagram (a)

b) X(X+Y)=X



Logic diagram (b)

Proof: a) X+XY = X

LHS = x + xy

= x(1 + y)

= x.1 ($\therefore$ 1+Y=1)

= x ($\therefore$X.1=X)

= RHS

Truth Table for X+XY = X is given below:

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

Table 1.34 : Truth Table for X+XY = X

Column X and X+XY are identical. Hence proved

(b) Since rule (b) is dual of rule (a), it is also proved. However, we are giving the algebraic proof of this law.

L.H.S.= X(X+Y) = X.X + XY
= X.X + XY
= X + XY (X.X = X Indempotence Law)
= X(1+Y)
= X.1 (using 1 + Y = 1 properties of 0, 1)
= X (X . 1 = X using property of 0, 1)
= RHS

Truth table for X(X+Y)=X

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

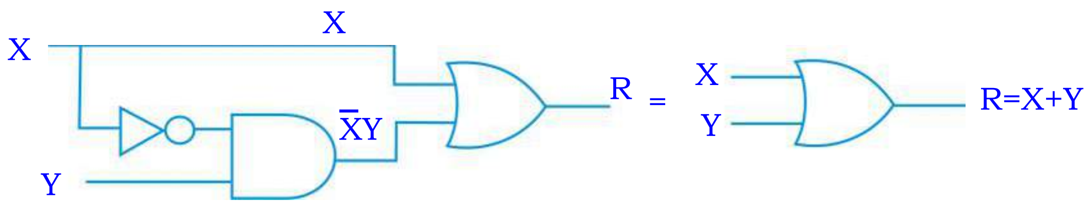
Some Other Rules of Boolean Algebra

There are some more rules of Boolean algebra which are given below:

$X+\bar{X}Y=X+Y$ (This is the third distributive law)

This rule can easily be proved by truth tables. As you are quite familiar with truth tables now, truth table proof is left for you as exercise, the other proofs of these rules are being given here:

$X+\bar{X}Y=X+Y$



Proof : **LHS = X + XY**

= (x+x)(x+y) {x+x= 1}

= 1.(x+y)

= x+y

= RHS

All the theorems of Boolean algebra, which we have been covered so far, are summarized in the following table:

Table 1.35 Boolean algebra rules

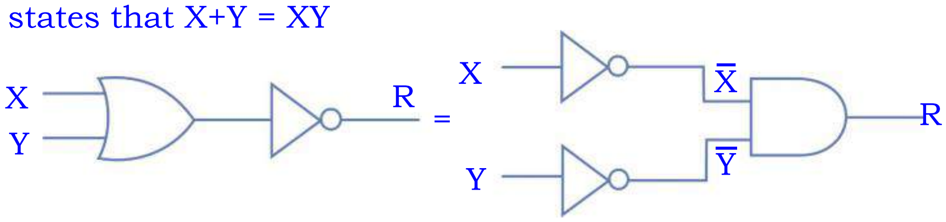
1	0+X=X	Properties of 0
2	0.X = 0	
3	1+X=1	Properties of 1
4	1.X = X	
5	X + X = X	Indempotence law
6	X . X = X	
7	$\overline{\overline{X}}$ = X	Involution
8	X + $\overline{X}$ = 1	Complementarity law
9	X. $\overline{X}$ = 0	
10	X + Y = Y + X	Commutative law
11	X . Y = Y. X	
12	X + (Y + Z) = (X+Y)+Z	Associative law
13	X(YZ) = (XY) Z	
14	X (Y+Z) = XY+XZ	Distributive law
15	X+YZ=(X+Y) (X+Z)	
16	X+XY=X	Absorption law
17	X . (X+Y) = X	
18	X+$\overline{X}$Y=X+Y	

2.6 De Morgan's theorems

One of the most powerful identities used in Boolean algebra is De Morgan's theorem. Augustus De Morgan had paved the way to Boolean algebra by discovering these two important theorems. This section introduces these two theorems of De Morgan.

De Morgan's First Theorem

It states that $\overline{X+Y} = \overline{X}\,\overline{Y}$



Proof: To prove this theorem, we need to recall complementarity laws, which state that $X+\overline{X}=1$ and $X . \overline{X} = 0$ i.e., a logical variable/expression when added with its complement produces the output 1 and when multiplied with its complement produces the output 0.

Now to prove De Morgan's first theorem, we will use complementarity laws.

Let us assume that P=X+Y where, P, X, Y are logical variables. Then, according to complementation law $P + \overline{P} = 1$ and $P.\overline{P} = 0$.

That means, if P, X, Y are Boolean variables then this complementarity law must hold for variable P. i.e., $\overline{P} = \overline{X+Y} = \overline{X}\,\overline{Y}$

Therefore $P+\overline{P}=(X+Y)+\overline{X}\,\overline{Y}$

$(X+Y) + \overline{X}\ \overline{Y}$ must be equal to 1. (As $X+\overline{X} = 1$)

And, $(X+Y).\ \overline{X}\ \overline{Y}$ must be equal to 0 (As $X.\overline{X}= 0$)

Let us first prove the first part, i.e., $(X + Y) + (\overline{X}\,\overline{Y}) = 1$

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

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

$= (1 + Y) . (X + 1)$

$= 1.1$ (ref. $X + \overline{X} = 1$)

$= 1$ (ref. $1 + X = 1$)

Now, let us prove the second part. i.e., $(X + Y) . (\overline{X}\,\overline{Y}) = 0$

$(X + Y) . (\overline{X}\,\overline{Y}) = \overline{X}\,\overline{Y}. (X + Y)$ (ref. X(YZ)=((XY)Z)

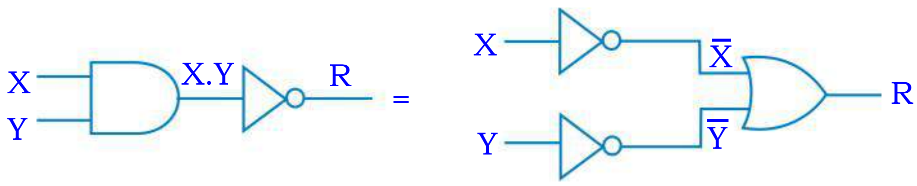
$= X\overline{X}\,\overline{Y} + \overline{X}Y\overline{Y}$

$= 0.Y + X.0$

$= 0+0=0$ (ref. $X . \overline{X} = 0$)

2.6.2 De Morgan's Second Theorem

This theorem states that: $\overline{(X.Y)} = \overline{X} + \overline{Y}$



Proof: Again to prove this theorem, we will make use of complementarity law i.e.,

$$X + \overline{X} = 1 \text{ and } X.\overline{X} = 0$$

If $\overline{XY}$'s complement is $\overline{X} + \overline{Y}$ then it must be true that

(a) $X\,Y + (\overline{X} + \overline{Y}) = 1$ and (b) $X\,Y\,(\overline{X} + \overline{Y}) = 0$

To prove the first part

$$
\begin{aligned}
\text{L.H.S.} &= XY + (\overline{X} + \overline{Y}) \\
&= (\overline{X} + \overline{Y}) + XY \\
&= (\overline{X} + \overline{Y} + X) \,.\, (\overline{X} + \overline{Y} + Y) \\
&= (X + \overline{X} + \overline{Y}) \,.\, (\overline{X} + Y + \overline{Y}) \\
&= (1 + \overline{Y}) \,.\, (\overline{X} + 1) \qquad (\text{ref. } X + \overline{X} = 1) \\
&= 1.1 \qquad (\text{ref. } 1 + X = 1) \\
&= 1 = \text{R.H.S.}
\end{aligned}
$$

Now, the second part. i.e., $XY \,.\, (\overline{X} + \overline{Y}) = 0$

$$
\begin{aligned}
\text{L.H.S.} &= XY.\, (\overline{X} + \overline{Y}) \qquad (\text{ref. } X(Y+Z)=XY+XZ) \\
&= XY\overline{X} + XY\overline{Y} \\
&= X\overline{X}Y + XY\overline{Y} \\
&= 0.Y + X.0 \qquad (\text{ref. } X \,.\, \overline{X} = 0) \\
&= 0+0 \\
&= 0 \\
&= \text{RHS}
\end{aligned}
$$

$XY.(\overline{X} + \overline{Y}) = 0$ and $XY(\overline{X} + \overline{Y}) = 1$

Thus, $\overline{X.Y} = \overline{X}+\overline{Y}$ Hence the theorem.

Although the identities above represent De Morgan's theorem, the transformation is more easily performed by following these steps:

(i) Complement the entire function

(ii) Change all the ANDs (.) to ORs (+) and all the ORs (+) to ANDs (.)

(iii) Complement each of the individual variables.

This process is called De Morganization.

'Break the line, change the sign' to De Morganize a Boolean expression.

a) Solve using De Morgan's Theorem

$\overline{\bar{A}\bar{B}+\bar{A}+AB} = \overline{\bar{A}\bar{B} + \bar{A} + AB}$ (∴$\overline{AB}=\bar{A}+\bar{B}$; Demorgan's 2nd theorem)

=

$= \overline{\overline{AB}} \,.\, \bar{\bar{A}} \,.\, \overline{AB}$ (∴$\overline{X+Y}=\bar{X}.\bar{Y}$) Demorgan's law)

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

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

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

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

$=AB.0+ABA\bar{B}$

$=0+0$

$=0$

2.6.3 Applications of De Morgan's Theorem

1. De Morgan's theorem useful in the implementation of the basic gate operations with alternative gates, particularly with NAND and NOR gates which are readily available in IC form.

2. De Morgan's theorem is used in the simplification of Boolean expressions.

3. De Morgan's laws commonly apply to text searching using Boolean operators AND, OR and NOT. Consider a set of documents containing the words "cars" or "trucks". De Morgan's laws hold that these two searches will return the same set of documents.

4. De Morgan's laws are an example of a more general concept of mathematical duality.

2.6.4 Basic Duality of Boolean algebra

We already have talked about duality principle. If you observe all the theorems and rules covered so far, you'll find a basic duality which underlines all Boolean algebra. The postulates and theorems which have been presented can all be divided into pairs.

For example, X+X.Y = X

Its dual will be X.(X+Y) = X

(Remember, change . to + and vice versa; complement 0 and 1.)

Similarly, (X+Y)+Z = X+(Y+Z) is the dual of (X.Y).Z=X.(Y.Z)

and X+0 = X is dual of X.1 = X

In proving the theorems or rules of Boolean algebra, it is then necessary to prove only one theorem, and the dual of the theorem follows necessarily.

In effect, all Boolean algebra is predicated on this two-for-one basis.

Example 1.17: Give the dual of following result in Boolean algebra:

$$X\overline{X} = 0 \text{ for each } X.$$

Solution: Using duality principle, dual of $X.\overline{X}=0$ is $X+\overline{X}=1$ (By changing (.) to (+) and vice versa and by replacing 1's by 0's and vice versa).

Example 1.18: Give the dual of X+0=X for each X.

Solution: Using duality principle, dual of X+0=X is X.1=X

Example 1.19: State the principle of duality in Boolean algebra and give the dual of the Boolean expression: $(X+Y).(\overline{X}+\overline{Z}).(Y+Z)$

Solution: Principle of duality states that from every Boolean relation, another Boolean relation can be derived by

(i) Changing each OR sign (+) to an AND (.) sign
(ii) Changing each AND (.) sign to an OR (+) sign
(iii) Replacing each 1 by 0 each 0 by 1
The new derived relation is known as the dual of the original relation.
Dual of $(X+Y).(\overline{X}+\overline{Z}).(Y+Z)$ will be

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

2.7 DERIVATION OF BOOLEAN EXPRESSION

Boolean expressions which consist of a single variable or its complement e.g., X or Y or Z are known as literals.

Now before starting derivation of Boolean expression, first we will talk about two very important terms. These are (i) Minterms (ii)Maxterms

2.7.1 Minterms

Minterm is a **product** of all the literals (with or without the bar) within the logic system.

One of the most powerful theorems within Boolean algebra states that any Boolean function can be expressed as the sum of products of all the variables within the system. For example, X+Y can be expressed as the sum of several products, each of the product containing letters X and Y. These products are called Minterms and each product contains all the literals with or without the bar.

Also when values are given for different variables, minterm can easily be formed. E.g., if X=0, Y=1, Z=0 then minterm will be $\overline{X}Y\overline{Z}$ i.e., for variable with a value 0, take its complement and the one with value 1, multiply it as it is. Similarly for X=1, Y=0, Z=0, minterm will be $X\overline{Y}\overline{Z}$.

Steps involved in minterm expansion of expression

1. First convert the given expression in sum of products form.
2. In each term, if any variable is missing (e.g., in the following example Y is missing in first term and X is missing in second term), multiply that term with (missing term+$\overline{\text{missing term}}$) factor, (e.g., if Y is missing multiply with $Y+\overline{Y}$).
3. Expand the expression.
4. Remove all duplicate terms and we will have minterm form of an expression.

Example 1.20: Convert X+Y to minterms.

Solution: $X+Y=X.1+Y.1$

$=X.(Y+\overline{Y})+Y(X+\overline{X})$ $\quad$ ($X+\overline{X}=1$ complementary law)

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

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

$=XY + X\overline{Y} + \overline{X}Y$ $\quad$ (XY + XY = XY Indempotent law)

Note that each term in the above example contains all the letters used: X and Y. The terms XY, X and Y are therefore minterms. This process is called expansion of expression.

Other procedure for expansion could be

1. Write down all the terms
2. Put X's where letters much be inserted to convert the term to a product term.
3. Use all combinations of X's in each term to generate minterms.
4. Drop out duplicate terms.

Example 1.21: Find the minterms for AB+C.

Solution: It is a 3 variable expression, so a product term must have all three letters, A, B and C.

1. Write down all terms AB+C
2. Insert X's where letters are missing ABX+XXC
3. Write all the combinations of X's in first term $AB\overline{C}$, ABC

Write all the combinations of X's in second term $\overline{A}\overline{B}C$, $A\overline{B}C$, ABC, $\overline{A}BC$

4. Add all of them. Therefore, $AB+C= AB\overline{C}+ABC+\overline{A}\overline{B}C+A\overline{B}C+ABC+\overline{A}BC$
5. Now remove all duplicate terms. $AB\overline{C}+ABC+\overline{A}\overline{B}C+A\overline{B}C+\overline{A}BC$

Now to verify, we will prove vice versa

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

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

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

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

$= \overline{A}C.1 + A\overline{B}C+ AB.1$

$= \overline{A}C.1 + A\overline{B}C + AB.1$

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

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

$= \overline{A}C + A(B + C)$ $(B+\overline{B}C=B+C$ Absorption law)

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

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

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

$= C.1 + AB$

$= C + AB$

$= AB + C$

= RHS

Shorthand minterm Notation

Since all the letters (2 in case of 2 variable expression, 3 in case of 3 variable expressions) must appear in every product, a shorthand notation has been developed that saves actually writing down the letters themselves. To form this notation, following steps are to be followed:

1. First of all, copy original terms.
2. Substitute 0's for barred letters and 1's for non-barred letters
3. Express the decimal equivalent of binary word as a subscript of m.

Example 1.22: To find the minterm designation of $X\overline{Y}\overline{Z}$

Solution: 1. Copy original form = $X\overline{Y}\overline{Z}$

2. Substitute 1's for non-barred and 0's for barred letters.

   Binary equivalent = 100

3. Decimal equivalent of 100 = $1x2^2 + 0x2^1 + 0x2^0 = 4 + 0 + 0 = 4$

4. Express as decimal subscript of

   Thus $X\overline{Y}\overline{Z} = m_4$

Similarly, minterm designation of $A\overline{B}C\overline{D}$ would be

Copy Original Term $A\overline{B}C\overline{D}$

Binary equivalent = 1010

Decimal equivalent = $1x2^3 + 0x2^2 + 1x2^1 + 0x2^0 = 8 + 0 + 2 + 0 = 10$

Express as subscript of m = m_{10}

2.7.2 Maxterms

A maxterm is a **sum** of all the literals (with or without the bar) within the logic system.

Trying to be logical about logic, if there is something called minterm, there surely must be one called maxterm and there is.

If the value of a variable is 1, then its complement is added otherwise the variable is added as it is.

Example: If the values of variables are X=0, Y=1 and Z=1 then its Maxterm will be $\overline{X} + \overline{Y} + Z$(Y and Z are 1's, so their complements are taken; X= 0, so it is taken as it is).

Similarly if the given values are X=1, Y=0, Z =0 and W=1 then its Maxterm is

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

Maxterms can also be written as M (Capital M) with a subscript which is decimal equivalent of given input combination e.g., above mentioned Maxterm $\overline{X}+Y+Z+\overline{W}$ whose input combination is 1001 can be written as M_9 as decimal equivalent of 1001 is 9.

2.7.3 Canonical Expression

Canonical expression can be represented in following two forms:

(i) Sum-of-Products (SOP)

(ii) Product-of-sums (POS)

Boolean Expression composed entirely either of minterms or maxterms is referred to as **Canonical Expression**.

Sum-of-Products (SOP)

A logical expression is derived from two sets of known values:

- Various possible input values
- The desired output values for each of the input combinations.

Let us consider a specific problem.

A logical network has two inputs X and Y and an output Z. The relationship between inputs and outputs is to be as follows:

(i) When X=0 and Y=0 then Z=1

(ii) When X =0 and Y=1 then Z=0

(iii) When X =1 and Y=0, then Z=1

(iv) When X=1 and Y=1, then Z=1

We can prepare a truth table from the above relations as follows:

X	Y	Z	Product Terms
0	0	1	$\overline{X}\overline{Y}$
0	1	0	$\overline{X}Y$
1	0	1	$X\overline{Y}$
1	1	1	XY

Table 1.36 truth table for product terms (2-input)

Here, we have added one more column to the table consisting list of product terms or minterms. Adding all the terms for which the output is 1. i.e., Z=1 we get following expression:

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

Now see, it is an expression containing only minterms. This type of expression is called **minterm canonical form of Boolean expression** or **canonical sum-of-products form of expression**.

When a Boolean expression is represented purely as sum of minterms, it said to be in canonical SOP form.

Example 1.23: A Boolean function F defined on three input variables X, Y and Z is 1 if and only if number of 1(one) inputs is odd (e.g., F is 1 if X=1, Y=0,Z=0), Draw the truth table for the above function and express it in canonical sum of products from.

Solution: The output is 1, only if one of the inputs is odd. All the possible combinations when one of inputs is odd are

X=1. Y=0, Z=0

X=0, Y=1, Z=0

X=0, Y=0, Z=1

X=1, Y=1, Z=1

For these combinations output is 1, otherwise output is 0. Preparing the truth table for it we get the following truth table.

X	Y	Z	F	Product Terms/ Minterms
0	0	0	0	$\overline{X}\,\overline{Y}\,\overline{Z}$
0	0	1	1	$\overline{X}\,\overline{Y}Z$
0	1	0	1	$\overline{X}Y\overline{Z}$
0	1	1	0	$\overline{X}YZ$
1	0	0	1	$X\overline{Y}\,\overline{Z}$
1	0	1	0	$X\overline{Y}Z$
1	1	0	0	$XY\overline{Z}$
1	1	1	1	XYZ

Table 1.37 truth table for product terms (3-input)

Adding all the minterms (product terms) for which output is 1, get

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

This is the desired Canonical SOP from

So, deriving SOP expression from truth table can be summarized as follows:

1. For a given expression, prepare a truth table for all possible combinations of inputs.
2. Add a new column for minterms and list the minterms for all the combinations.
3. Add all the minterms for which there is output as 1. This gives you the desired canonical S-O-P expression.

Another method of deriving canonical SOP expression is algebraic method. This is just the same as above. We will take another example here.

Example 1.24: Convert $\overline{X}Y + \overline{X}\,\overline{Z}$ into canonical SOP from.

Solution: Rule 1: Simplify the given expression using appropriate theorems/ rules.

$$\overline{(\overline{X}Y) + (\overline{X}\,\overline{Z})} = (X + \overline{Y})\,(X + Z) \quad \text{using demorgan's law}$$

$$= X + \overline{Y}Z \qquad \text{(Using Distributive law)}$$

Since it is a 3 variable expression, a product term must have all 3 variables.

Rule 2: Wherever a literal is missing, multiply that term with

missing variable + $\overline{\text{missing variable}}$

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

(Y, Z are missing in first term, x is missing in second term)

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

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

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

Rule 3: By removing the duplicate terms, we get $XYZ + X\overline{Y}Z + XY\overline{Z} + \overline{X}\overline{Y}Z + \overline{X}\overline{Y}Z$ This is the desired Canonical SOP from.

Above Canonical SOP expression can also be represented by following shorthand notation. Here F is a variable function and m is a notation for minterm. This specifies that output F is sum of 1st, 4th, 5th, 6th and 7th minterms.

i.e., $F = m_1 + m_4 + m_5 + m_6 + m_7$ or $F=\Sigma(1,4,5,6,7)$

Converting Shorthand notation to minterms

We already have learnt how to represent minterm into shorthand notation. Now we will learn how to convert vice versa.

Rule1: Find binary equivalent of decimal subscript e.g., for m_6 subscript is 6, binary equivalent of 6 is 110.

Rule2: For every 1's write the variable as it is and for 0's write variable's complemented form i.e., for 110 it is $XY\overline{Z}$. $XY\overline{Z}$ is the required minterm for m_6.

Example 1.25: Convert the following three input function F denoted by the expression into its canonical SOP form.

Solution: If three inputs are X, Y and Z then

$F = m_0 + m_1 + m_2 + m_5$

$m_0=000 \Rightarrow \overline{X}\overline{Y}\overline{Z}$

$m_1=001 \Rightarrow \overline{X}\overline{Y}Z$

$m_2=010 \Rightarrow \overline{X}Y\overline{Z}$

$m_5=101 \Rightarrow X\overline{Y}Z$

Canonical SOP form of the expression is

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

Product-of-sum form (POS)

When a Boolean expression is represented purely as product of Maxterms, it is said to be in canonical Product-of-Sum form.

This form of expression is also referred to as Maxterm canonical form of Boolean expression.

Just as any Boolean expression can be transformed into a sum of minterms, it can also be represented as a product of Maxterms.

(a) Truth table method

The truth Table method for arriving at the desired expression is as follows:

1. Prepare a table of inputs and outputs

2. Add one additional column of sum terms. For each row of the table, a sum term is formed by adding all the variables in complemented or uncomplemented form. i.e., if input value for a given variable is 1, variable is complemented and if 0, not complemented.

Example: If X=0, Y=1, Z=1 then Sum term will be $X + \overline{Y} + \overline{Z}$

Now the desired expression is product of the sums from the rows in which the output is 0.

Example 1.26: Express in the product of sums from the Boolean function F(X, Y, Z) and the truth table for which is given below:

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

Solution: Add a new column containing Maxterms. Now the table is as follow:

X	Y	Z	F	Maxterms
0	0	0	1	$X + Y + Z$
0	0	1	0	$X + Y + \overline{Z}$
0	1	0	1	$X + \overline{Y} + Z$
0	1	1	0	$X + \overline{Y} + \overline{Z}$
1	0	0	1	$\overline{X} + Y + Z$
1	0	1	0	$\overline{X} + Y + \overline{Z}$
1	1	0	1	$\overline{X} + \overline{Y} + Z$
1	1	1	1	$\overline{X} + \overline{Y} + \overline{Z}$

Now by multiplying maxterms for the output 0's, we get the desired product of sums expression which is $(X + Y + \bar{Z})\ (X + \bar{Y} + \bar{Z})\ (\bar{X} + Y + \bar{Z})$

(b) Algebraic Method

We will explain this method with the help of an example.

Example 1.27 Express $\bar{X}\,Y + Y(\bar{Z}\,(\bar{Z} + Y))$ into canonical product-of-sums form.

Solution: Rule 1: Simplify the given expression using appropriate theorems/rules:

$$\bar{X}\,Y + Y\,(\bar{Z}\,(\bar{Z} + Y)) = \bar{X}Y + Y(\bar{Z}\,\bar{Z} + Y\bar{Z}) \qquad \{X(Y+Z) = XY+XZ\}$$

$$= \bar{X}Y + Y\,(\bar{Z} + Y\bar{Z}\,) \qquad (Z\,.\,Z = Z \text{ as } X\,.\,X = X)$$

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

$$= \bar{X}\,Y + Y\,\bar{Z}\,.\,1 \qquad \{1 + Y = 1\}$$

$$= \bar{X}\,Y + Y\,\bar{Z}$$

Rule 2: To convert into product of sums form, apply the Boolean algebra rule which states that $X + YZ = (X + Y)\ (X + Z)$

$$\bar{X}Y + Y\bar{Z} = (\bar{X}\,Y + Y)\ (\bar{X}\,Y + \bar{Z}) \qquad (X + Y = Y + X)$$

$$= (Y + \bar{X}\,Y)\ (\bar{Z} + \bar{X}\,Y)$$

$$= (Y + \bar{X})\ (Y + Y)\ (\bar{Z} + \bar{X})\ (\bar{Z} + Y)$$

$$= (\bar{X} + Y)\ Y\ (\bar{X} + \bar{Z})\ (Y + \bar{Z}) \qquad (X + Y = Y)$$

Now, this is in product of sums form but not in canonical product of sums form (In Canonical expression all the sum terms are Maxterms.)

Rule 3: After converting into product of sum terms, in a sum term for a missing variable add (Missing variable . $\overline{\text{missing variable}}$.) e.g., if variable Y is missing add $Y\bar{Y}$.

$$(\bar{X} + Y)\ (Y)(\bar{X} + \bar{Z})(Y + \bar{Z}) = (\bar{X} + Y + Z\bar{Z})\ (X\,\bar{X} + Y + Z\bar{Z})\ (\bar{X} + Y\,\bar{Y} + \bar{Z})\ (X\,\bar{X} + Y + \bar{Z})$$

Rule 4: Keep on simplifying the expression (using the rule, X+YZ=(X+Y)(X+Z)) until you get product of sum terms which are maxterms.

$$(\bar{X} + Y + Z\bar{Z}) = (\bar{X} + Y + Z)\ (\bar{X} + Y + \bar{Z}) = M_4 .\ M_5$$

$$(\,X\,\bar{X} + Y + Z\bar{Z}) = (X\bar{X} + Y + Z)\ (X\bar{X} + Y + \bar{Z})$$

$$= (X + Y + Z)\ (\bar{X} + Y + Z)\ (X + Y + \bar{Z})\ (\bar{X} + Y + \bar{Z}) = M_0.M_4.M_1.M_5$$

$$(\bar{X} + Y\bar{Y} + \bar{Z}) = (\bar{X} + Y + \bar{Z})\ (\bar{X} + \bar{Y} + \bar{Z}) = M_5.M_7$$

$$(\,X\,\bar{X} + Y + \bar{Z}\,) = (X + Y + \bar{Z})\ (\bar{X} + Y + \bar{Z}) = M_1.M_5$$

$$(\bar{X} + Y)\ (Y)\ (\bar{X}{+}Z)\ (Y{+}\bar{Z}) = (\bar{X} + Y + Z)\ (\bar{X} + Y + \bar{Z})\ (X + Y + Z)\ (\bar{X} + Y + Z)\ (X + Y + \bar{Z})\ (\bar{X} + Y + \bar{Z})\ (\bar{X} + Y + \bar{Z})\ (\bar{X} + \bar{Y} + \bar{Z})\ (X + Y + \bar{Z})\ (\bar{X} + Y + \bar{Z})$$

Short hand = M_4, M_5, M_0, M_4, M_1, M_5, M_5, M_7, M_1, M_5 = M(0, 4, 5, 7)

Rule 5: Removing all the duplicate terms, we get

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

This is the desired canonical product of sums form of expression.

Shorthand maxterm notation

Shorthand notation of the above given canonical product of sums expression is

F= Π⊕(0,1,4,5,7) or F = Π M(0,1,4,5,7)

This specifies that output F is product 0^{th}, 1^{st}, 4^{th} and 7^{th} Maxterms

i.e., $F = M_0.M_1.M_5.M_5.M_7$

Here, M_0 means Maxterm for Binary equivalent of 0 i.e., 000 ie., X=0, Y=0, Z=0

And, Maxterm will be (X+Y+Z) (Complemented variable is 1 and uncomplemented variable is 0)

Similarly, M_1 Means 0 0 1 $X+Y+\overline{Z}$

AS $F = M_0 .M_1 .M_{4.} M_5 .M_7$

and M_0 = 000 $X + Y + Z$

M_1 = 001 $X + Y + \overline{Z}$

M_4 =100 $\overline{X} + Y + Z$

M_5 = 101 $\overline{X} + Y + \overline{Z}$

M_7 = 111 $\overline{X} + \overline{Y} + \overline{Z}$

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

Example 1.28: Convert the following function into canonical product of sums form: F(X, Y, Z) = ΠM(0, 2, 4, 5)

Note: To convert an expression from shorthand SOP form to shorthand POS form, just create truth table from given expression. From the created truth table, derive other form of expression. For example, from truth table, you can convert an expression F(X, Y,Z)= Σ(0,1,3,5) to Π⊕M(2,4,6,7)

Solution: F Π(X, Y, Z) = Π⊕M(0, 2,4,5) = $M_0 \, . \, M_2 \, .M_4 \, .M_5$

M_0 = 000 $X + Y + Z$

M_2 = 010 $X + \overline{Y} + Z$

M_4 = 100 $\overline{X} + Y + Z$

M_5 = 101 $\overline{X} + Y + \overline{Z}$

⇒⊕F = $(X + Y + Z)\ (X + \overline{Y} + Z)\ (\overline{X} + Y + Z)\ (\overline{X} + Y + \overline{Z})$

Sum term V/s Maxterm and product term V/s minterm

Sum term means sum of the variables. It does not necessarily mean that all the variables must be included whereas Maxterm means a sum-term having the entire variables.

For Example, for 3 Variables F(X, Y, Z) functions X + Y, X + Z, $\overline{Y}$ + Z etc. are sum terms whereas X + Y + Z, $\overline{X}$ + Y + $\overline{Z}$, X + $\overline{Y}$ + Z etc. are Maxterms.

Similarly, product term means product of the variables, not necessarily all the variables, whereas minterm means product of all the variables.

For A 3 variable (a, b, c) function $\overline{a}bc$, $a\overline{b}c$, $ab\overline{c}$ etc. are minterms whereas ab, bc, bc, ac etc. are product terms only.

Same is the difference between Canonical SOP or POS expression. A Canonical SOP or POS expression must have all the Minterms or Maxterms respectively, whereas a simple SOP or POS expression can just have product terms or sum terms respectively.

2.7.4 Minimization of Boolean expression

After obtaining an SOP or POS expression, the next thing to do is to simplify the Boolean expression, because Boolean operations are practically implemented in the form of gates. A minimized Boolean expression means less number of gates which means simplified circuitry. This section deals with two methods simplification of Boolean expression.

Algebraic Method

This method makes use of Boolean postulates, rules and theorems to simplify the expressions.

Example 1.29 simplify $A\overline{B}C\overline{D} + A\overline{B}CD + ABC\overline{D} + ABCD$

Solution: $A\overline{B}C\overline{D} + A\overline{B}CD + ABC\overline{D} + ABCD$

$A\overline{B}C(\overline{D} + D) + ABC(\overline{D} + D) = A\overline{B}C.1 + ABC.1$ $(\overline{D} + D = 1)$

$= AC(\overline{B}+B) = AC$

Example 1.30: Reduce the expression $\overline{XY} + \overline{X} + XY$

Solution: $\overline{XY} + \overline{X} + XY$

$= (\overline{X} + \overline{Y}) + \overline{X} + XY$ (using Demorgan's 2nd theorem i.e.,)

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

$= \overline{X} + XY + \overline{Y}$ $\{\overline{X} + \overline{X} = \overline{X}$ as $X + X = X\}$

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

$= \overline{X} + Y + \overline{Y}$ $\{Y + \overline{Y} = 1\}$

$= \overline{X} + 1$ {Putting $Y + \overline{Y} = 1$)

$= 1$ {putting $\overline{X} + 1 = 1$ as $1 + X = 1\}$

Example 1.31: Minimize $AB + \overline{AC} + A\overline{B}C\,(AB + C)$

Solution $AB + \overline{AC} + A\overline{B}C\,(AB + C) = AB + \overline{AC} + A\overline{B}CAB + A\overline{B}CC$

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

$= AB + \overline{AC} + A\overline{B}C$ $\{B\overline{B} = 0 \text{ and } CC = C\}$

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

$= \overline{A} + AB + \overline{C} + A\overline{B}C$ {rearranging the terms}

$= \overline{A} + B + \overline{C} + A\overline{B}C$ $\{\overline{A} + AB = \overline{A} + B \text{ because } X + \overline{X}Y = X + Y\}$

$= \overline{A} + \overline{C} + B + AC\overline{B}$ $(B + \overline{B}AC = B + AC \text{ because } X + \overline{X}Y = X + Y)$

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

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

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

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

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

$= 1$ $(1 + X = 1)$

Example 1.32: Reduce $\overline{X}\,\overline{Y}\,\overline{Z} + \overline{X}\,Y\,\overline{Z} + X\,\overline{Y}\,\overline{Z} + X\,Y\,\overline{Z}$

Solution. $\overline{X}\,\overline{Y}\,\overline{Z} + \overline{X}\,Y\,\overline{Z} + X\,\overline{Y}\,\overline{Z} + X\,Y\,\overline{Z} = \overline{X}\,(\overline{Y}\,\overline{Z} + Y\,\overline{Z}) + X\,(\overline{Y}\,\overline{Z} + Y\,\overline{Z})$

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

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

$= \overline{Z}$

2.8 Simplification using Karnaugh Maps

Truth tables provide a nice, natural way to list all values of a function. There are several other ways to represent function values. One of the way is Karnaugh Map (in short K-Map) named after its originator Maurice Karnaugh. These maps are sometimes also called Veitch diagrams.

Karnaugh Map or K-Map is a graphical display of the fundamental product in a truth table.

Karnaugh map is nothing but a rectangle made up of certain number of squares, each square representing a Maxterm or Minterm.

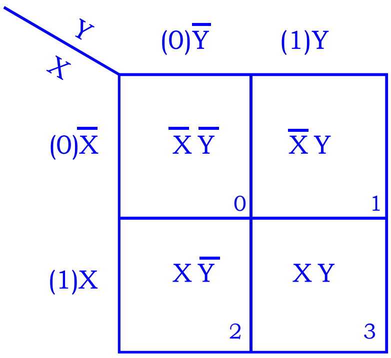
2.8.1 Sum of products Reduction using Karnaugh Map

In S-O-P reduction each square of K-Map represents a minterm of the given function. Thus, for a function of n variables, there would be a map of 2^n squares, each representing a minterm (refer to Fig. 1.7). Given a K-map, for SOP reduction the map is filled in by placing in squares whose minterms lead to a 1 output.

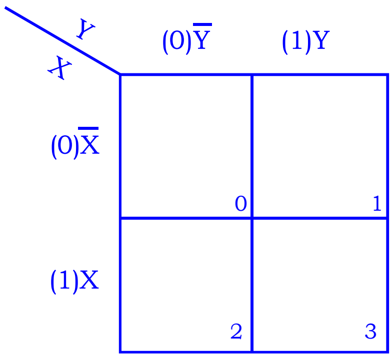
Following are 2,3,4 variable K-maps for SOP reduction. (see fig. 1.7)

Note in every square a number is written. These subscripted numbers denote that this square corresponds to that number's minterm. For example, in 3 variable map X Y Z box has been given number 2 which means this square corresponds to M_2. Similarly, box number 7 means it corresponds to m_7 and so on.

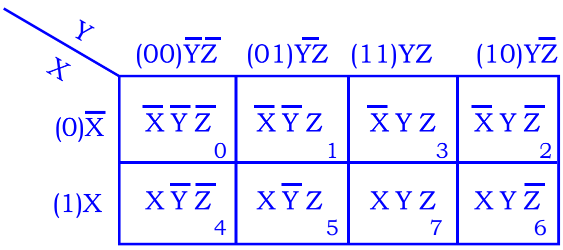
Please notice the numbering scheme here, it is 0, 1, 3, 2 then 4, 5, 7, 6 and so on, always squares are marked using this scheme while making a K-map.



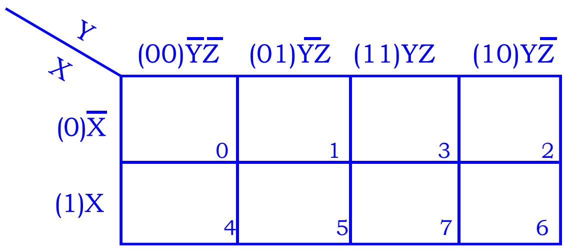
(a)



(a)



(c)



(d)

X \ Y	(00)$\overline{Y}\overline{Z}$	(01)$\overline{Y}Z$	(11)YZ	(10)$Y\overline{Z}$
(00)$\overline{W}\overline{X}$	$\overline{W}\overline{X}\overline{Y}\overline{Z}$ 0	$\overline{W}\overline{X}\overline{Y}Z$ 1	$\overline{W}\overline{X}YZ$ 3	$\overline{W}\overline{X}Y\overline{Z}$ 2
(01)$\overline{W}X$	$\overline{W}X\overline{Y}\overline{Z}$ 4	$\overline{W}X\overline{Y}Z$ 5	$\overline{W}XYZ$ 7	$\overline{W}XY\overline{Z}$ 6
(11)WX	$WX\overline{Y}\overline{Z}$ 12	$WX\overline{Y}Z$ 13	$WXYZ$ 15	$WXY\overline{Z}$ 14
(10)$W\overline{X}$	$W\overline{X}\overline{Y}\overline{Z}$ 8	$W\overline{X}\overline{Y}Z$ 9	$W\overline{X}YZ$ 11	$W\overline{X}Y\overline{Z}$ 10

X \ Y	(00)$\overline{Y}\overline{Z}$	(01)$\overline{Y}Z$	(11)YZ	(10)$Y\overline{Z}$
(00)$\overline{W}\overline{X}$	0	1	3	2
(01)$\overline{W}X$	4	5	7	6
(11)WX	12	13	15	14
(10)$W\overline{X}$	8	9	11	10

4-variable K Map representing minterms.

Observe carefully above given K-map. See the binary numbers at the top of K-map. These do not follow binary progression, instead they differ by only one place when moving from left to right : 00, 01, 11, 10. It is done so that only one variable changes from complemented to un complemented form or vice versa. The terms are $\overline{A}$ $\overline{B}$. $\overline{A}$ B, AB, A $\overline{B}$

This binary code 00, 01, 11, 10 is called Gray code. Gray Code is the binary code in which each successive number differs only in one place. That is why box numbering scheme follows above order only.

How to Map in K-Map?

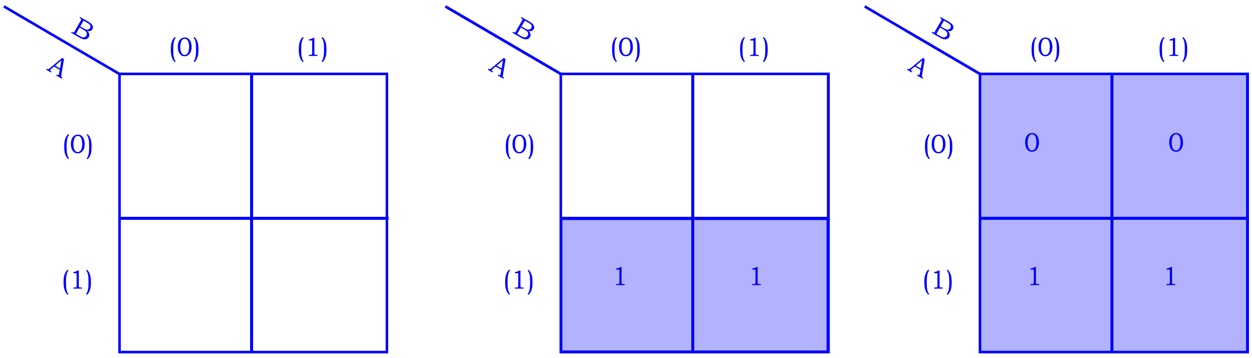
We'll take an example of 2 variable map to be illustrated, with the following truth table for mapping (Table 1.38)

Table 1.38

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

Canonical S-O-P expression for this table is F=AB + AB or F = Σ(2,3).

To map this function first we'll draw an empty 2-variable K-map as shown in Fig. 1.8(a)



(a) (b) (c)

Now look for output 1 in the given truth table (1.38) for a given truth table,.

For minterms M_2 and M_3 the output is 1. Thus mark 1 in the squares for m_2 and m_3 i.e., square numbered as 2 and the one numbered as 3. Now our K-map will look like fig 1.8 (b)

After entering 1's for all 1 outputs, enter 0's in all blank squares. K-map will now look like Fig 1.8 same is the method for mapping 3-variable and 4-varible maps i.e., enter 1's for all 1 outputs in the corresponding squares and then enter 0's in the rest of the squares.

How to reduce Boolean expression in S-O-P form using K-map?

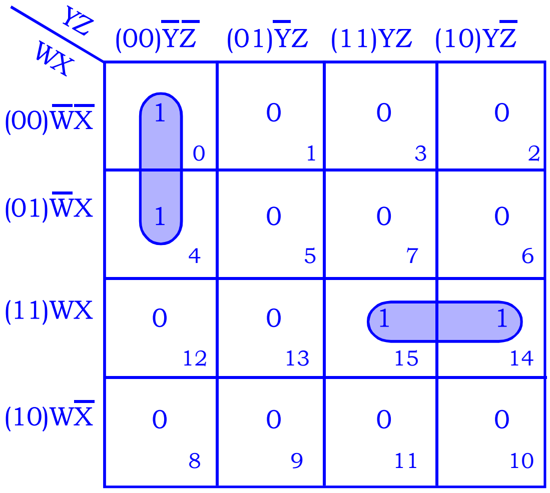
For reducing the expression, first we have to mark pairs, quads and octets.

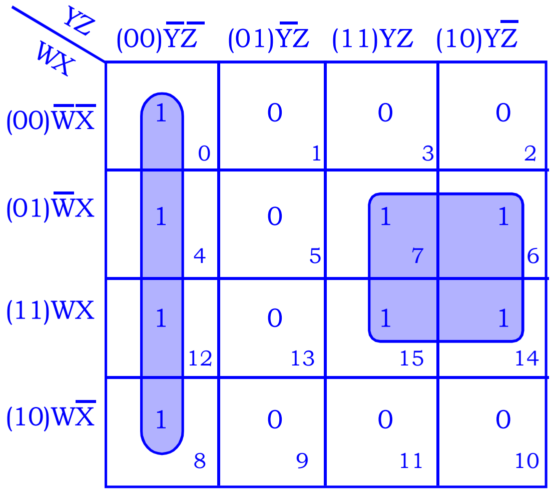
To reduce an expression, adjacent 1's are encircled. If 2 adjacent 1's are encircled, it makes a pair; if 4 adjacent 1's are encircled, it makes a quad; and if 8 adjacent 1's are encircled, it makes an octet.

While encircling groups of 1's, firstly search for octets and mark them, then for quads and lastly go for pairs. This is because a bigger group removes more variables thereby making the resultant expression simpler.

Reduction of a pair : In the K-map in fig. 1.9, after mapping a given function F(W, X, Y, Z) two pairs have been marked. Pair-1 is m_0+m_4 (group of 0^{th} minterm and 4^{th} minterm as these numbers tell us minterm's subscript). Pair-2 is m_{14}+m_{15}.

Observe that Pair-1 is a vertical pair. Moving vertically in pair-1, see one variable X is changing its state from $\overline{X}$ to X as m_0 is $\overline{W}\,\overline{X}\,\overline{Y}\overline{Z}$ and m_4 is $\overline{W}\,X\,\overline{Y}\overline{Z}$. Compare the two and we see $\overline{W}\,\overline{X}\,\overline{Y}\overline{Z}$ changes to $\overline{W}\,X\,\overline{Y}\overline{Z}$. So, the variable X can be removed.





Pair Reduction Rule

Remove the variable which changes its state from complemented to uncomplemented or vice versa. Pair removes one variable only.

Thus reduced expression for Pair-1 is $\overline{W}\,\overline{Y}\,\overline{Z}$ as $\overline{W}\,\overline{X}\,\overline{Y}\,\overline{Z}$ (m_0) changes to $\overline{W}\,X\,\overline{Y}\,\overline{Z}$ (m_4)

We can prove the same algebraically also as follows :

$$\text{Pair-1} = m0 + m4 = \overline{W}\,\overline{X}\,\overline{Y}\,\overline{Z} + \overline{W}\,X\,\overline{Y}\,\overline{Z}$$

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

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

$$= \overline{W}\,\overline{Y}\,\overline{Z}$$

Similarly, reduced expression for Pair-2 (m_{14}+m_{15}) will be WXY as WXYZ (m_{14}) changes to WXY$\overline{Z}$ (m_{15}). Z will be removed as it is changing its state from to $\overline{Z}$.

Reduction of a quad

If we are given with the K-map shown in fig. 1.10 in which two quads have been marked.

Quad-1 is m_0 + m_4 + m_{12} + m_8 and Quad-2 is m_7+m_6+m_{15}+m_{14}. When we move across quad-1, two variables change their states i.e., W and X are changing their states, so these two variables will be removed.

Quad, Reduction Rule

Remove the two variables which change their states. A Quad removes two variables. Thus reduced expression for quad-1 is $\overline{Y}\,\overline{Z}$ as W and X (both) are removed.

Similarly, in Quad -2 ($m_7+m_6+m_{15}+m_{14}$), horizontally moving, variable Z is removed as $\overline{W}$ X Y Z (m_7) changes to $\overline{W}$ X Y $\overline{Z}$ (m_6) and vertically moving, variable W is removed as (m_7) changes to WXYZ. Thus reduced expression for quad-2 is (by removing W and Z) XY. XY+$\overline{Y}\overline{Z}$

Reduction of an octet

Suppose, we have K-map with an octet marked as shown in Fig. 1.11.

WX \ YZ	(00)$\overline{Y}\overline{Z}$	(01)$\overline{Y}Z$	(11)YZ	(10)$Y\overline{Z}$
(00)$\overline{W}\overline{X}$	0 (0)	0 (1)	0 (3)	0 (2)
(01)$\overline{W}X$	0 (4)	0 (5)	0 (7)	0 (6)
(11)WX	1 (12)	1 (13)	1 (15)	1 (14)
(10)$W\overline{X}$	1 (8)	1 (9)	1 (11)	1 (10)

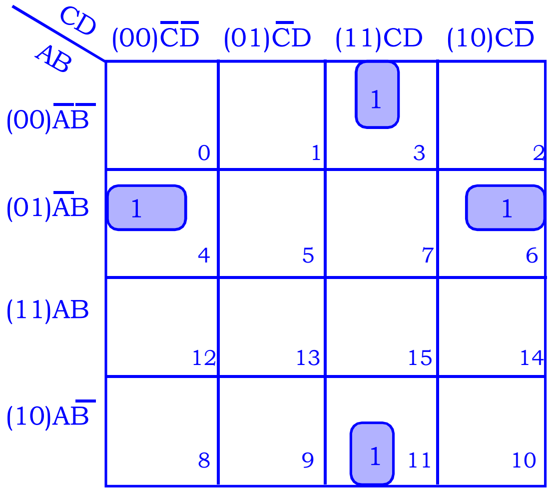
While moving horizontally in the octet two variables Y and Z are removed and moving vertically one variable x is removed. Thus eliminating X, Y and Z, the reduced expression for the octet is W only.

Octet Reduction Rule

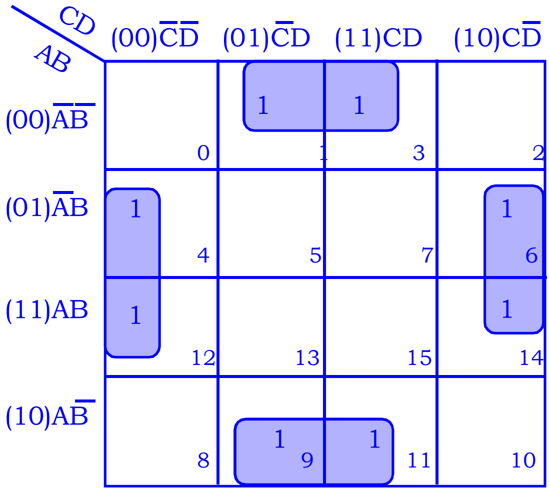
Remove the three variables which change their states. An octet removes 3-variables. But after marking pairs, quads and octets, there are certain other things to be taken care of before arriving at the final expression. These are map rolling, overlapping groups and redundant groups.

Map Rolling

Map Rolling means roll the map i.e., consider the map as if its left edges are touching the right edges and top edges are touching the bottom edges. This is a special property of Karnaugh maps that its opposite edges squares and corner squares are considered contiguous (Just as the world map is treated contiguous at its opposite ends). As in opposite edges squares and in corner squares only one variable changes its state from complemented to uncomplemented state or vice versa. Therefore, while making the pairs, quads and octets, map must be rolled. Following pairs, quads and octets are marking after rolling the map.



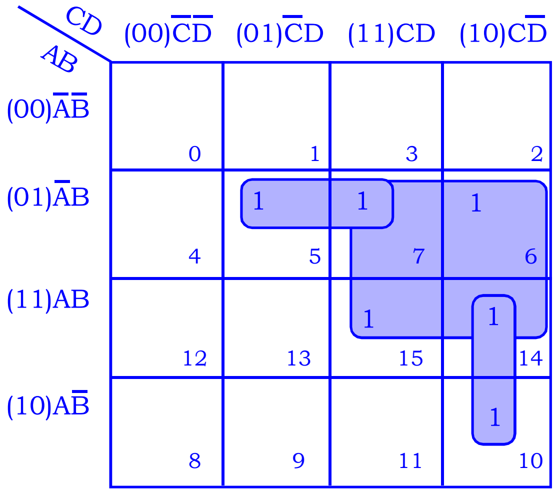
$\overline{A}B\overline{D}+\overline{B}CD$



$B\overline{D}+\overline{B}D$

Overlapping Groups

Overlapping means same 1 can be encircled more than once. For example, if the following K-map is given:



Observe that 1 for m_7 has been encircled twice. Once for Pair-1(m_5+m_7) and again for Quad (m_7+m_6+m_{15}+m_{14}). Also 1 for m_{14} has been encircled twice. For the Quad and for Pair-2 (m_{14}+m_{10}).

Here, reduced expression for Pair-1 is ABD

Reduced expression for Quad is BC

Reduced expression for Pair-2 is ACD

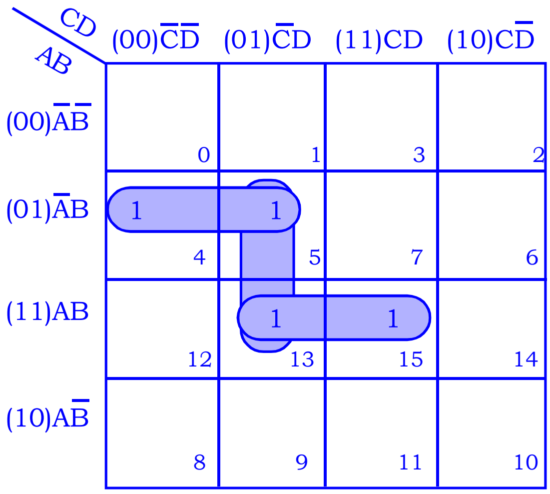
Thus final reduced expression for this map is ABD + BC + ACD

Thus reduced expression for entire K-map is sum of all reduced expressions in the very K-map.

But before writing the final expression we must take care of redundant Groups.

Redundant Groups

Reduntant group is a group whose all 1's are overlapped by other groups (i.e., pairs, quads, octets). Here is an example, given below.



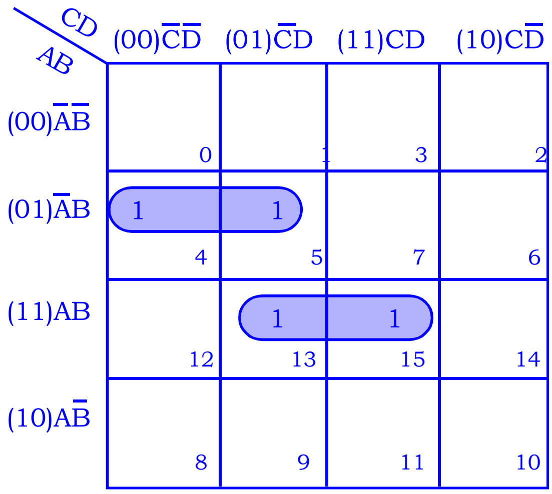


Fig. 1.14(a) has a redundant group. There are three pairs : Pair-1 (m_4+m_5), Pair-2 (m_5+m_{13}), Pair-3 (m_{13}+m_{15}). But Pair-2 is a redundant group as its all 1's are marked by other groups.

With this reduntant group, the reduced expression will be $\bar{A}B\bar{C}$+BD+ABD. For a simpler expression, Redundant Groups must be removed. After removing the redundant group, we get the K-map shown in fig. 1.14 (b).

The reduced expression, for K-map in fig. 1.14 (b), will be

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

Which is much simpler expression Σ.

Thus removal of redundant group leads to much simpler expression.

Summary of all the rules for S-O-P reduction using K-map

1. Prepare the truth table for given function.
2. Draw an empty K-map for the given function (i.e., 2 variable K-map for 2 variable function; 3 variable K-map for 3 variable function, and so on).

3. Map the given function by entering 1's for the outputs as 1 in the corresponding squares.
4. Enter 0's in all left out empty squares.
5. Encircle adjacent 1's in form of octets, quads and pairs. Do not forget to roll the map and overlap.
6. Remove redundant groups, if any.
7. Write the reduced expressions for all the groups and OR (+) them.

Example 1.33 Reduce F (a, b, c, d) = Σm (0,2,7,8,10,15) using Karnaugh map.

Solution: Given F (a, b, c, d) = Σm (0,2,7,8,10,15)

$$= m_0 + m_2 + m_7 + m_8 + m_{10} + m_{15}$$

$m_0 = 0000 = \bar{A}\,\bar{B}\,\bar{C}\,\bar{D}$ $m_2 = 0010 = \bar{A}\,\bar{B}\,C\,\bar{D}$

$m_7 = 0111 = \bar{A}\,B\,C\,D$ $m_8 = 1000 = A\,\bar{B}\,\bar{C}\,\bar{D}$

$m_{10} = 1010 = A\,\bar{B}\,C\,\bar{D}$ $m_{15} = 1111 = A\,B\,C\,D$

Truth table for the given function is as follows :

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

Mapping the given function in a K-map we get

AB \ CD	(00)$\bar{C}\bar{D}$	(01)$\bar{C}D$	(11)CD	(10)$C\bar{D}$
(00)$\bar{A}\bar{B}$	1 (0)	0 (1)	0 (3)	1 (2)
(01)$\bar{A}B$	0 (4)	0 (5)	1 (7)	0 (6)
(11)AB	0 (12)	0 (13)	1 (15)	0 (14)
(10)$A\bar{B}$	1 (8)	0 (9)	1 (11)	1 (10)

In the above K-map two groups have been marked, one Pair and One Quad.

Pair is $m_7 + m_{15}$

And Quad is $m_0 + m_2 + m_8 + m_{10}$

Reduced expression for pair ($m_7 + m_{15}$) is BCD as A is removed. Reduced expression for quad ($m_0 + m_2 + m_8 + m_{10}$) is $\bar{B}\bar{D}$ as for horizontal corners C is removed and for vertical corners A is removed.

Thus final reduced expression is BCD + $\bar{B}\bar{D}$

Example 1.34: What is the simplified Boolean equation for the function?

$$F(A,B,C,D) = \Sigma(7,9,10,11,12,13,14,15)$$

Solution: Completing the given Karnaugh map by entering 0's in the empty squares, by numbering the squares with their minterm's subscripts and then by encircling all possible groups, we get the following K-map.

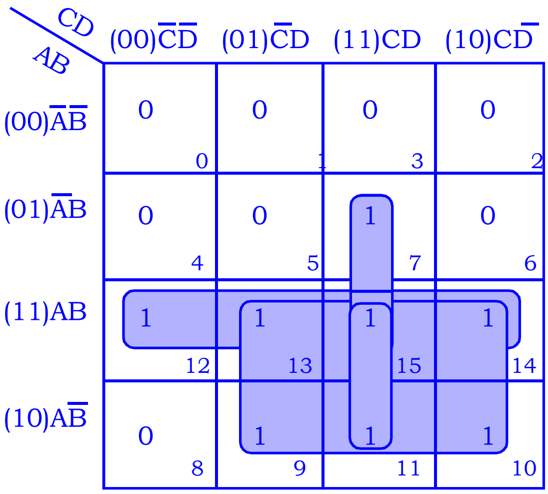
There is one pair, three quads

Pair-1 = $m_7 + m_{15}$

Quad-1 = $m_{12} + m_{13} + m_{14} + m_{15}$

Quad-2 = $m_{13} + m_{15} + m_9 + m_{11}$

Quad-3 = $m_{15} + m_{11} + m_{14} + m_{10}$



Reduced expression for pair-1 ($m_7 + m_{15}$) is BCD, as $\bar{A}BCD$ (m_7) changes to ABCD (m_{15}) eliminating A.

Reduced expression for Quad-1 ($m_{12} + m_{13} + m_{14}$) is AB, as while moving across the Quad, C and D both are removed because both are changing their states from complemented to uncomplemented or vice-versa.

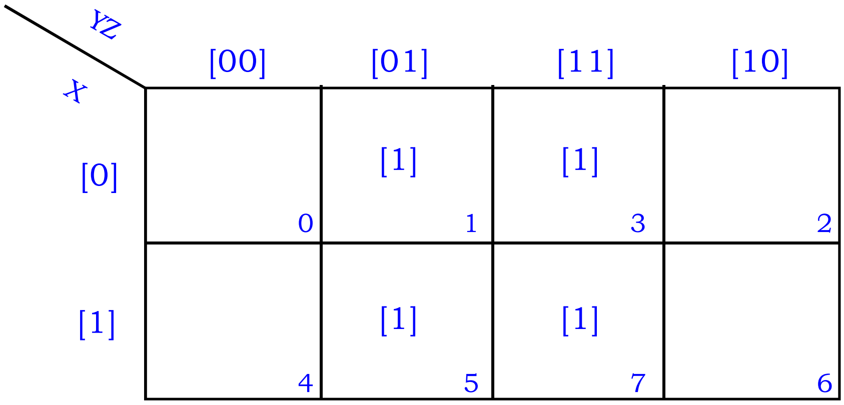
Reduced expression for Quad 2 ($m_{13} + m_{15} + m_9 + m_{11}$) is AD, as moving horizontally, C is removed and moving vertically, B is removed.

Reduced expression of Quad-3 ($m_{15} + m_{11} + m_{14} + m_{10}$) is AC as horizontal movement removes D and vertical movement removes B.

Thus, Pair-1 = BCD, Quad-1 = AB, Quad-2 = AD, Quad-3 = AC

Hence final reduced expression will be BCD+AB+AD+AC.

Example 1.35: Obtain a simplified expression for a Boolean function F (X, Y, A) the Karnaugh map for which is given below:

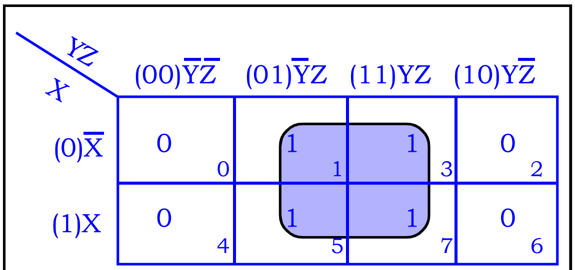


Solution: Completing the given K-map.

We have 1 group which is a Quad i.e.,

$m_1 + m_3 + m_5 + m_7$

Reduced expression for this Quad is Z, as moving horizontally from $\bar{X}\,\bar{Y}\,Z$ (m1) to $\bar{X}\,\bar{Y}\,Z$ (m3), Y is removed (Y changing from $\bar{Y}$ to Y) and moving vertically from m1 to m5 or m3 to m7, $\bar{X}$ changes to X, thus $\bar{X}$ is removed.



(c)

Example 1.36: Minimize the following function using a Karnaugh map:

F (W, X, Y, Z) = Σ(0,4,8,12)

Solution: Given function F (W, X, Y, Z) = Σ(0,4,8,12)

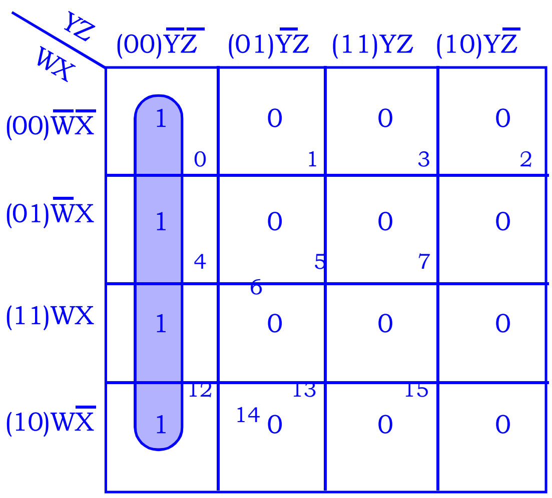
$F = m_0 + m_4 + m_8 + m_{12}$

$m_0 = 0000 = \bar{W}\,\bar{X}\,\bar{Y}\,\bar{Z}$

$m_4 = 0100 = \bar{W}\,X\,\bar{Y}\,\bar{Z}$

$m_8 = 1000 = W\,\overline{X\,Y\,Z}$

$m_{12} = 1100 = W\,X\,\overline{Y\,Z}$

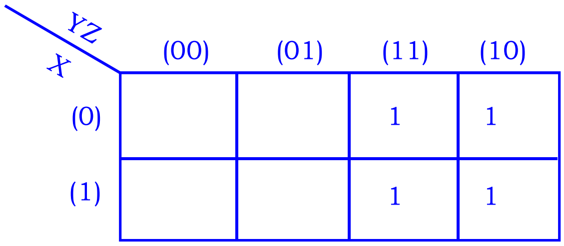


Maping the given function on a K-Map, we get $(m_0 + m_4 + m_8)$

Reduced expression for this quad is $\overline{Y}\overline{Z}$ as while moving across the Quad W and X are removed. Because these are changing their states from complemented to uncomplemented or vice versa.

Thus, final reduced expression is $\overline{Y}\overline{Z}$.

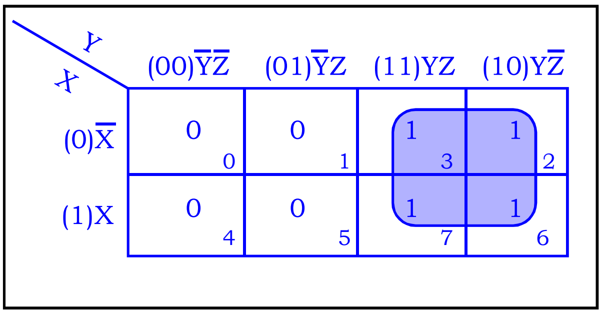
Example 1.37: Using the Karnaugh technique obtain the simplified expression as sum products for the following map:



(d)

Solution: Completing the given K- map, we get one group which is a Quad has been marked.

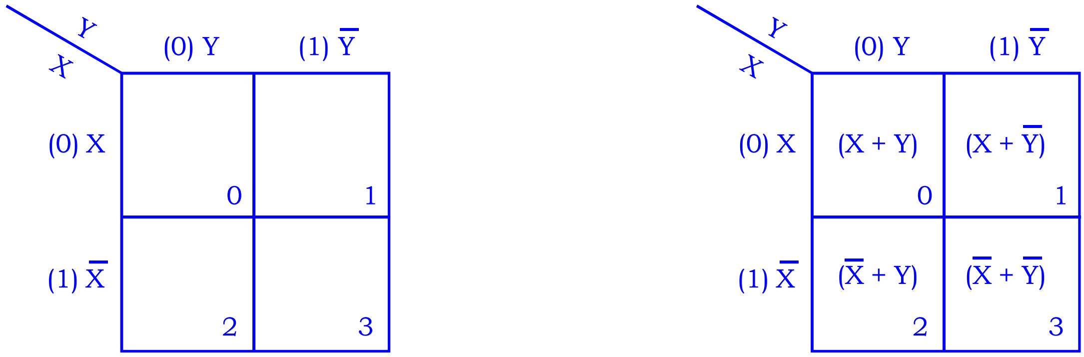
Quad reduces two variables. Moving horizontally, Z is removed as it changes from Z to $\overline{Z}$ and moving vertically, X is removed as it changes from X to $\overline{X}$. Thus only one variable Y is left. Hence Reduced S-O-P expression is Y. Thus F=Y assuming F is the given function.



2.8.2 Product-of-Sum Reduction using Karnaugh Map

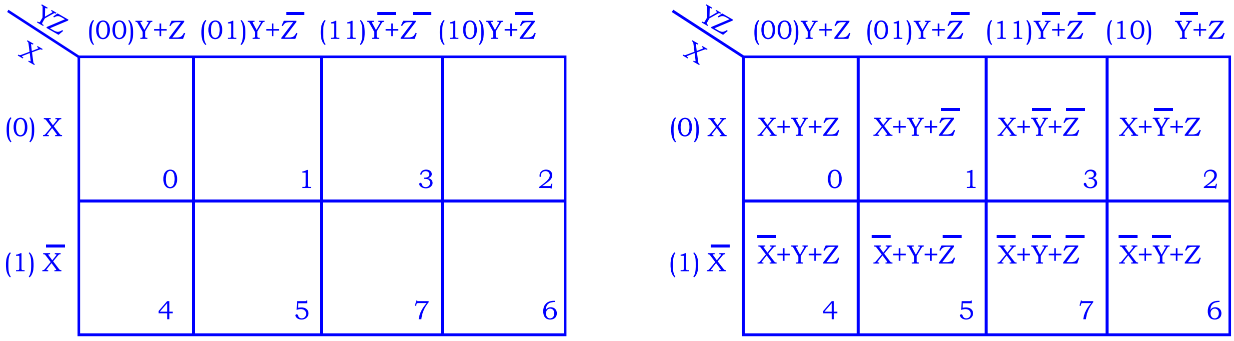
In POS reduction each square of K-map represents a Maxterm. Karnaugh map is just the same as that of the used in S-O-P reduction. For a function of n variables, map would represent 2^n squares, each representing a maxterm.

For POS reduction map is filled by placing 0's in squares whose Maxterms lead to output 0.Following are 2, 3 4 variable K-Maps for POS reduction.



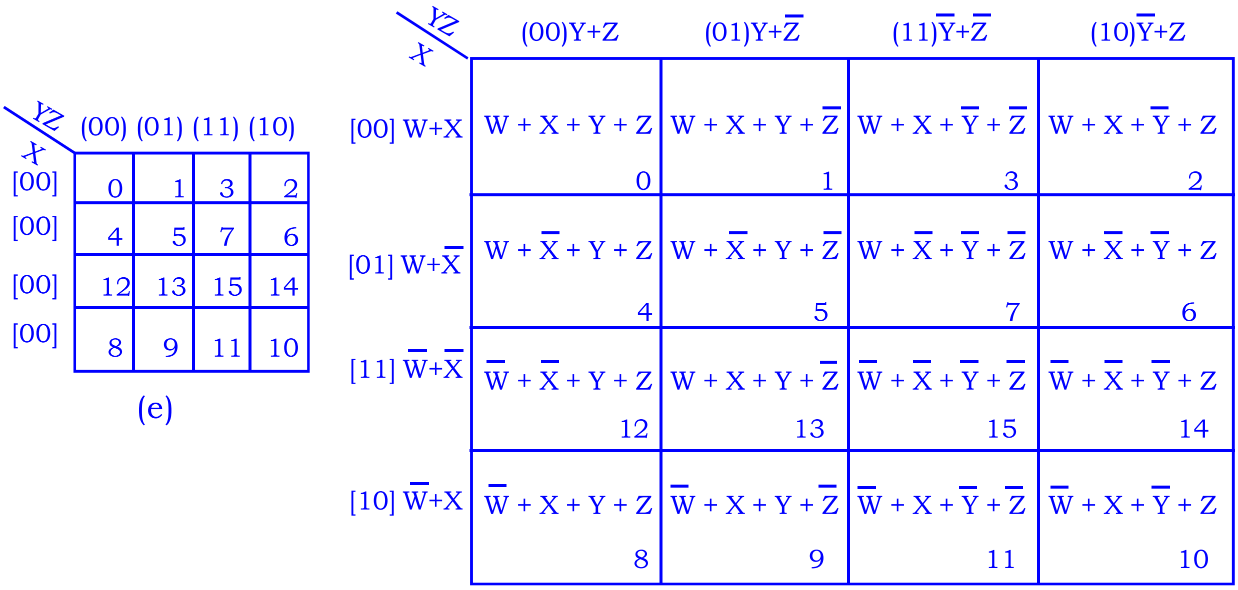
(a) (b)

2- variable K-Map representing Maxterms.



(c) (d)

3-variable K-Map representing Maxterms



(e) (f)

4 varible K–Map representing Minterms

Figure 1.115: 2,3,4 variable K-Maps of POS expression.

Again the numbers in the squares represent Maxterm subscripts. Box with number 1 represent M1, Number 6 represent, M6, and so on. Also notice box numbering scheme is the same i.e., 0, 1, 3, 2 ; 4, 5, 7, 6 ; 12, 13, 15, 14 ; 8, 9, 11, 10.

One more similarity in SOP K-map and POS K-map is that they are binary progression in gray code only. So, here also some Gray Code appears at the top.

But one major difference is that in POS K-Map, complemented letters represent 1's uncomplemented letters represent 0's, whereas it is just the opposite in SOP K-Map. Thus in the fig 1.15 (b), (d), (f) for 0's uncomplemented letters appear and for 1's complemented letters appear.

How to derive POS Boolean expression using K-Map?

Rules for deriving expression are the same except for the thing i.e., POS expression adjacent 0's are encircled in the form of pairs, quads and octets. Therefore, rules for deriving POS Boolean expression can be summarized as follows:

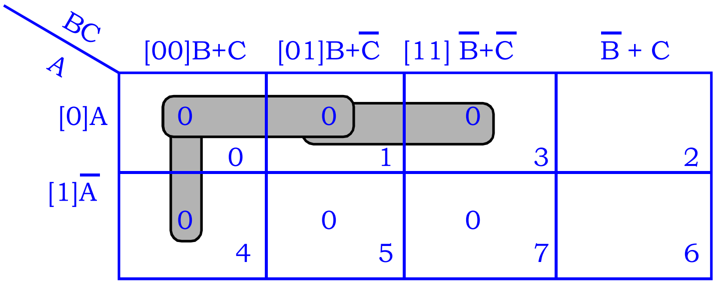
1. Prepare the truth table for a given function.
2. Draw an empty K-map for given function (i.e., 2-variable K-map for 2 variable function; 3 variable K-map for 3 variable function and so on).
3. Map the given function by entering 0's then squares numbered 5 and 13 will be having 0's)
4. Enter 1's in all left out empty squares.
5. Encircle adjacent 0's in the form of octets, quads, and pair. Do not forget to role the map and overlap.
6. Remove redundant groups, if any.
7. Write the reduced expressions for all the groups and AND (.) them.

Example 1.38: Reduce the following Karnaugh map in Product of sums form:

A \ BC	(00)	(01)	(11)	(10)
(0)	0	0	0	1
(1)	0	1	1	1

Solution: To reach at POS expression, we'll have to encircle all possible groups of adjacent 0's encircling we get the following K-map.

There are 3 pairs which are;



Pair1: $M_0 \, . \, M_1$;

Pair 2: $M_0 \, .M_4$;

Pair 3: $M_1 \, . \, M_{3;}$

But there is one redundant group also i.e., Pair-1 (it's all 0's are encircled by other groups). Thus removing this redundant pair-1, we have only two groups now.

Reduced POS expression for Pair-2 is (B+C), as while moving across pair-2, A changes its state from A to A, thus A is removed.

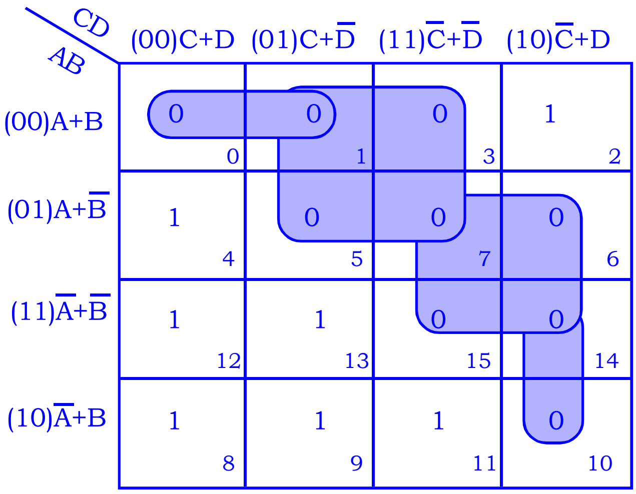
Reduced POS expression for Pair 3 is $(A+\overline{C})$, as while moving across Pair 3 B changes to $\overline{B}$, hence eliminated.

Final POS expression will be $(B+C).(A+\overline{C})$

Example1.39: Find the minimum POS expression of

$$Y(A, B, C, D) = \Pi(0, 1, 3, 5, 6, 7\ 10, 14, 15).$$

Solution: As the given function is 4 variable function, we'll draw 4 variable K-Map and then put 0's for the given Maxterms. i.e., in the squares whose numbers are 0, 1, 3, 5, 6, 7, 10, 14, 15 as each square number represents its Maxterm. So, K-map will be



Encircling adjacent 0's we have following groups:

Pair-1 = $M_0 . M_{1;}$ Pair -2= $M_{14} . M_{10}$;

Quad = $M_1 . M_3 . M_5 . M_7$; Quad-2 = $M_7 . M_6 . M_{15} . M_{14;}$

Reduced expressions are the following:

For Pair -1, (A+B+C) (as D is eliminated: D changes to $\overline{D}$)

For pair-2, $(\overline{A}+\overline{C}+D)$ ($\overline{B}$ Changes to B; hence eliminated)

For Quad-1, $(A+\overline{D})$ (Horizontally C and vertically B is eliminated as C, B are changing their states)

For Quad-2, $(\overline{B}+\overline{C})$ (horizontally D and vertically A is eliminated)

Hence final POS expression will be

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

Review questions:

One mark questions:

1. What is another name of Boolean algebra?
2. What do you understand by logic function?
3. Give examples for logic function.
4. What is meant by tautology and fallacy?
5. Prove the 1+Y is a tautology and 0.Y is a fallacy.
6. State indempotence law.
7. Prove indempotence law using truth table.
8. Draw logic diagram to represent indempotence law.
9. State Involution law.
10. Prove Involution law using truth table.
11. Draw logic diagram to represent Involution law.
12. State Complementarity law.
13. Prove Complementarity law using truth table.
14. Draw logic diagram to represent Complementarity law.
15. State Commutative law.
16. Prove Commutative law using truth table.
17. Draw logic diagram to represent Commutative law.
18. State Associative law.
19. Prove Associative law using truth table.
20. Draw logic diagram to represent Associative law.
21. State Distributive law.
22. Prove Distributive law using truth table.
23. Draw logic diagram to represent Distributive law.
24. Prove that $X+XY = X$ (Absorption law)
25. Prove that $X(X+Y) = X$ (Absorption law)
26. Draw logic diagram to represent Absorption law.
27. Prove that $XY + X\overline{Y} = X$
28. Prove that $(X+Y)(X+\overline{Y}) = X$
29. Prove that $X + \overline{X} Y = X + Y$

30. What is a minterm?

31. Find the minterm for XY + Z.

32. What is a maxterm?

33. Find the maxterm for X + $\bar{Y}$ + Z.

34. What is the canonical form of Boolean expression?

Two marks questions:

1. Prove algebraically that (X + Y) (X + Z) = X + YZ
2. Prove algebraically that X + $\bar{X}$Y = X + Y
3. Use duality theorem to derive another Boolean relation from : A + $\bar{A}$B=A+B
4. What would be complement of the following :

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

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

(c) $XY + \bar{Y}Z + Z\bar{Z}$

(d) $X + \bar{X}Y + \bar{X}\bar{Z}$

5. What are the fundamental products for each of the input words; ABCD = 0010, ABCD = 110, ABCD = 1110. Write SOP expression.
6. A truth table has output 1 for each of these inputs. ABCD = 0011, ABCD = 0101, ABCD = 1000, what are the fundamental products and write minterm expression.
7. Construct a Boolean function of three variables X, Y and Z that has an output 1 when exactly two of X, Y and Z are having values 0, and an output 0 in all other cases.
8. Construct a truth table for three variables A, B and C that will have an output 1 when XYZ = 100, XYZ = 101, XYZ = 110 and XYZ = 111. Write the Boolean expression for logic network in SOP form.
9. Convert the following expressions to canonical Product-of-Sum form:

(a) (A+C)(C+D)

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

(c) (X+Y)(Y+Z)(X+Z)

10. Convert the following expressions to canonical Sum-of-Product form:

(a) $(X+\bar{X}Y+\bar{X}\bar{Z})$

(b) $YZ + \bar{X}Y$

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

11. Draw Karnaugh maps for the following expressions:

(a) $\bar{X}\bar{Y} + \bar{X}Y$

(b) $XY\bar{Z} + \bar{X}\bar{Y}Z$

(c) $\bar{X}\bar{Y}\bar{Z} + X\bar{Y}\bar{Z} + \bar{X}Y\bar{Z}$

12. Draw a general K-map for four variables A, B, C and D.
13. Given the expression in four variables , draw the K – map for the function:

(a) $m_2 + m_3 + m_5 + m_7 + m_9 + m_{11} + m_{13}$

(b) $m_0 + m_2 + m_4 + m_8 + m_9 + m_{10} + m_{11} + m_{12} + m_{13}$

14. Draw the K – map for the function in three variables given below.

(a) $m_0 + m_2 + m_4 + m_6 + m_7$

(b) $m_1 + m_2 + m_3 + m_5 + m_7$

15. Write S-O-P expression corresponding to the function F in the following truth table and draw the logic gate diagram (use OR and AND gates)

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

Three marks questions:

1. State and prove any three theorems of Boolean algebra.
2. State and prove associative law of addition and multiplication.
3. State and prove De Morgam's theorems by the method of perfect induction.
4. Obtain the minterm expression for the Boolean function F = A+BC.
5. Explain with an example how to express a Boolean function in its sum-of-products form.
6. Explain with an example how to express a Boolean function in its product- of-sum form.
7. Construct a truth table for minterms and maxterms for three variables and designate the terms.
8. Using basic gates, construct a logic circuit for the Boolean expression $(\overline{X}+Y).(X+Z).(Y+Z)$
9. Simplify the following Boolean expressions and draw logic circuit diagrams of the simplified expressions using only NAND gates.

(a) $\overline{A}BC + \overline{A}B\overline{C} + A\overline{B}\overline{C} + A\overline{B}C$

(b) $\overline{A}C + \overline{A}B + A\overline{B}\overline{C} + BC$

(c) $(ABC).(AB\overline{C})+A\overline{B}C + \overline{A}BC$

(d) $\overline{A}BC + A\overline{B}\overline{C} + ABC + AB\overline{C}$

(e) $(A+B+C)(A+B+\overline{C})(A+B+\overline{C})(A+\overline{B}+C)$

10. For a four variable map in w,x,y and z draw the subcubes for

(a) $WX\overline{Y}$ (b) WX (c) $XY\overline{Z}$ (d) Y

11. Convert the following product-of-sums form into its corresponding sum-of-products form using write the truth table.
F(x,y,z) = Π(2,46,7)

12. (a) Reduce the following Boolean expression to the simplest form:
$A.[\ B+C.(\overline{AB + AC})$
(b) Given : F(x,y,z) =Σ(1,3,7) then prove that F'(x,y,z) = Π(0,2,4,5,6)

Five marks questions:

1. Using maps, simplify the following expressions in four variables W, X, Y and Z.

(a) $m_1+m_3+m_5+m_6+m_7+m_9+m_{11}+m_{13}$

(b) $m_0+m_2+m_4+m_8+m_9+m_{10}+m_{11}+ m_{12}+m_{13}$

2. For the Boolean function F and F' in the truth table , find the following:

(a) List the minterms of the functions F and F'

(b) Express F and F' in sum of minterms in algebraic form.

(c) Simplify the functions to an expression with a minimum number of literals.

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

3. State and prove De Morgan's theorems algebraically.
4. Find the complement of F = X + YZ, then show that F.F' =0 and F + F' = 1.
5. (a) State the two Absorption laws of Boolean algebra. Verify using truth table.

(b) Simplify using laws of Boolean algebra. At each step state clearly the law used for simplification. F = x.y + x.z + x.y.z

6. Given the Boolean function F (x, y, z) = Σ(0, 2, 4, 5, 6). Reduce it using Karnuagh map method.
7. (a) State the two complement properties of Boolean algebra. Verify using the truth tables. (b) $x.(\overline{yz} + yz)$
8. Given the Boolean function F(A,B,C,D) = Σ(5,6,7,8,9,10.14). Use Karnaugh's map to reduce the function F using SOP form. Write a logic gate diagram for the reduced SOP expression.
9. Given ; F(A,B,C,D) = (0,2,4,6,8,10,14). Use Karnaugh map to reduce the function F using POS form. Write a logic gate diagram for the reduced POS expression.
10. Use Karnaugh map to reduce the given functions using SOP form. Draw the logic gate diagrams for the reduced SOP expression. You may use gates with more than two inputs. Assume that the variables and their complements are available as inputs.

11. Given the Boolean function $F(A,B,C,D)=\Sigma(0,4,8,9,10,11,12,13,15)$.

Reduce it by using Karnaugh map.

Working Sheet: