Chapter 1: Boolean Algebra

The topics:

1. Boolean Algebra

Studying logic is important as because it provides the output for our assumption as true or false. We can define logic in many ways – it can be said as logic is all about arguments or it is a formal method of reasoning.

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

2. Proposition Logic

In Boolean Algebra, the variables can assume only one out of the two values '0' and '1'. Hence, those variables are called binary valued quantities. Boolean Algebra is thus compatible with Binary Arithmetic.

A declarative statement which can be either 'True' or 'False' is said to be a Proposition. A proposition cannot be at True or False states simultaneously. At the same time, it cannot be as Non True and Non False states simultaneously. Thus proposition can be stated as an elementary atomic statement that may take either True value or False value but may not take any other value.

For example consider the following statements:-

It is raining.	[It is a proposition as it may either be true or false depending upon whether it is raining or not raining]
Today is Monday.	[It is a proposition as it will be true when the day is Monday and on other days it will be false]

Similarly,

- It is Hot today
- I like Tea
- I will go to New Delhi

All the above statements are propositions as they may be true or false.

3. Compound Proposition (Logical Operation)

A compound proposition is formed when two or more component propositions are connected to produce a new proposition. A component of a compound proposition is any whole proposition that is part of a larger proposition. The component propositions are connected using one or more of the binary connectives 'OR', 'AND', 'IF THEN', and 'IF AND ONLY IF'.

Let us assume two propositions are :-
x = "It is hot today", here x is termed as Boolean literals or variables
y = "It is raining outside"

Different types of connectives used in propositional logic are as given below:

1. **Disjunction (also called OR).** It is represented by + or $\vee$. Disjunction means one of the two arguments must be true or both to give the output of the compound proposition as true. E.g – x OR y, represented by $x + y$ or $x \vee y$, it means "either it is hot today or it is raining outside".
2. **Conjunction (also called AND).** It is represented by . (dot) or $\wedge$. Conjunction means both of the two arguments must be true to give the output of the compound proposition as

true. E.g – x AND y, represented by x . y or x ∧ y, it means "It is hot today and it is raining outside".

3. **Conditional (also called IF..THEN or Implication).** It is represented by ⇒ or ⊃. Conditional or Implication means if one argument is true then other argument is true. E.g – IF x THEN y, represented by x ⇒ y or x ⊃ y, it means "if it is hot today then it is raining outside". [It is to be noted that in this case if x is false but y is true then also the compound proposition x ⇒ y holds true and if x is true but y is false the final answer will be false.]
4. **Bi-conditional (also called IF and ONLY IF or Equivalence).** It is represented by ⇔ or ≡. Bi-conditional or equivalence means if and only if one argument is true then other argument is true. E.g – IF AND ONLY IF x THEN y, represented by x ⇔ y or x ≡ y, it means "if and only if it is hot today then it is raining outside". [in this case y is fully dependence upon x i.e. if x is true y must be true and if x is false y must be false.]
5. **Negation (also called NOT).** It is represented by ′ or ~ or – It is just a unary operator that inverse the proposition. E.g – NOT x, represented by x′ or ~x or $\bar{x}$. [It is an inverter that inverts the result of the statement.]

Proposition may also termed as well formed-formula (wff).

x = I like tea
y = I like coffee
z = I like biscuit
x+y = I like tea or I like coffee

x' = NOT(I like tea) = I don't like tea

x'.y' = I don't like tea and I don't like coffee
= Neither I like tea nor I like coffee

x'+y' = I don't like tea or I don't like coffee

x->y = If I like Tea then I like Coffee
x<->y = If and only If I like tea then I like Coffee

4. Truth Table

As every simple or compound proposition may take an either true value (also denoted by 1) or false value (also denoted by 0), these values are also called truth values that defines the truth or falsity of a proposition. Thus a truth table is the tabular representation of all the possible outcomes of a simple or compound proposition. When a compound statement or proposition has only two arguments, then there can be $2^2 = 4$ possible states in which they may assume various truth values. Similarly for three arguments statements, the possible outputs will be $2^3 = 8$ and for four arguments statements, the possible outputs will be $2^4 = 16$.
Thus a truth table is a tabular representation of all the possible outputs against a set of given inputs for a particular Propositional statement or Boolean expression.

4.1 Truth table for OR

INPUT		OUTPUT
X	Y	X+Y
0	0	0
0	1	1
1	0	1
1	1	1

4.2 Truth table for AND

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

4.3 Truth table for IF THEN

X	Y	$X \Rightarrow Y$
0	0	1
0	1	1
1	0	0
1	1	1

4.4 Truth table for IF AND ONLY IF

X	Y	$X \Leftrightarrow Y$
0	0	1
0	1	0
1	0	0
1	1	1

4.5 Truth table for NOT

X	X′
0	1
1	0

4.6 Some related terms in Truth Table

- Contingencies - The propositions that have some combination of 1's and 0's in their truth table output column are called contingencies.
- Contradiction - The propositions that have all combination of 0's in their truth table output.
- Tautologies - The propositions that have all combination of 1's in their truth table output.
- Converse– The converse of a conditional proposition is determined by interchanging the antecedent and consequent of the given condition. E.g – Converse of $p \Rightarrow q$ is $q \Rightarrow p$.
  'If I like Coffee then I like Tea' will become 'If I like Tea then I like Coffee'
- Inverse – The inverse of a conditional proposition is another conditional having negated antecedent and consequent. E.g – Inverse of $p \Rightarrow q$ is $\bar{p} \Rightarrow \bar{q}$
  'If I like Coffee then I like Tea' will become 'If I don't like Coffee then I don't like Tea'
- Contrapositive – It is formed by creating another conditional that takes its antecedent as negated consequent of earlier conditional and consequent as negated antecedent of earlier conditional. Contrapositive of $p \Rightarrow q$ is $q' \Rightarrow p'$
  'If I like Coffee then I like Tea' will become 'If I don't like Tea then I don't like Coffee'
- Consistent – Two statements are consistent if and only if their conjunction is not a contradiction. If X one statement and Y another, then X.Y is not a contradiction then the statements are said to be consistent.

5. Principle of Duality

It is a very important principle used in Boolean Algebra. By using this principle, we can derive another Boolean relation from an existing Boolean relation with the help of following changes:-

1. Changing each OR sign (+) to an AND sign (.) and vice-versa
2. Replacing each 0 to 1 and each 1 to 0
3. Complements remain unchanged
4. Priority of the literals in the original expression should not be changed. [NOT>AND>OR]

For example: Dual of (X+Y) .1 = X+Y will be (X.Y) + 0 = X.Y

Some other examples

F(A,B,C) = A.B + B'.C'

Dual of F = (A+B) . (B'+C') …. [not the A+B.B'+C]

Priority of Operators in Boolean Algebra: NOT>AND>OR () overrules the priority

F(A,B,C) = A+B.C'

Dual of F = A . (B+C') otherwise if you write F=A.B+C' it is a wrong answer.

6. Properties and Theorems of Boolean Algebra .

Name of the law	Boolean form	Dual form
1. Commutative law	**X + Y = Y + X**	**X . Y = Y . X**

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

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

2. Associative law	**(X + Y) + Z = X + (Y + Z)**	**(X . Y) . Z = X . (Y . Z)**

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

3. Distributive Law **	**X + Y . Z = (X + Y) . (X + Z)**	**X . (Y + Z) = X .Y + X . Z**

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

4. Idempotent law	**X + X = X**	**X . X = X**

X	X	**X+X**
0	0	**0**
1	1	**1**

X	X	**X.X**
0	0	**0**
1	1	**1**

5. Absorption Law **	**X + X .Y = X**	**X . (X + Y) = X**

X	Y	X.Y	**X+X.Y**
0	0	0	**0**
0	1	0	**0**
1	0	0	**1**
1	1	1	**1**

X	Y	X+Y	**X.(X+Y)**
0	0	0	**0**
0	1	1	**0**
1	0	1	**1**
1	1	1	**1**

6. De Morgans Law **

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

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

Break the bar and change the sign

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

7. Complements law

$X+\overline{X} = 1$ (Tautology)

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

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

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

8. Property of 1

X + 1 = 1

X	1	**X+1**
0	1	**1**
1	1	**1**

X. 1 = X

X	1	**X.1**
0	1	**0**
1	1	**1**

9. Property of 0

X + 0 = X

X	0	**X+0**
0	0	**0**
1	0	**1**

X. 0 = 0

X	0	**X.0**
0	0	**0**
1	0	**0**

10. Involution rule

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

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

11. Conditional elimination

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

X	Y	**X⇒Y**	$\overline{X}$	$\overline{X}+Y$
0	0	**1**	1	**1**
0	1	**1**	1	**1**
1	0	**0**	0	**0**
1	1	**1**	0	**1**

12. Bi-conditional discriminate

$X \Leftrightarrow Y = (X \Rightarrow Y).(Y \Rightarrow X)$ = X.Y + X'.Y'

X	Y	**X⇔Y**	X⇒Y	Y⇒X	**(X ⇒ Y).(Y ⇒ X)**
0	0	**1**	1	1	**1**
0	1	**0**	1	0	**0**
1	0	**0**	0	1	**0**
1	1	**1**	1	1	**1**

6.1 Practice Time Problems

A. Simplify the given expression to the lowest possible form using Boolean laws :

Example 1:

$\boxed{F(A,B,C) = B.(B+A).C.(B'+C)}$

Solution:

= (B.(B+A)) . (C.(B'+C))
= (B.B + A.B) . (B'.C + C.C)
= B.B.B'.C+A.B.B'.C+B.B.C.C+A.B.C.C
= 0 + 0 + B.C + A.B.C (by complement's law and idempotent law)
= B.C.(1+A)
= B.C.1 (by property of 1)
= B.C (Ans)

Alternative method:

= (B.(B+A)).(C.(B'+C))
= (B.B+B.A).(C.B'+C.C)
= (B+A.B).(B'.C+C) … by Idempotent law (B.B=B and C.C=C)
= B.B'.C+A.B.B'.C+B.C+A.B.C
= 0+0+B.C + A.B.C … by complement's law (B.B'=0)
= B.C(1+A)
= B.C (Ans)

Example 2:

$\boxed{F(X,Y) = (X+X.Y).(Y'+Y'.X')}$

Solution:

= X.Y'+X.Y'.X'+X.Y.Y'+X.Y.Y'.X'
= X.Y' + 0 + 0 + 0 (by complement's law)
= X.Y' (Ans)

Alternative method:

= X.Y' (by Absorption law... X+X.Y=X)

B. Draw the truth table for the following Boolean expressions:

Example 1:

F(A,B) = A.B’ + A’.B

Solution:

A	B	A’	B’	A.B’	A’.B	F
0	0	1	1	0	0	0
0	1	1	0	0	1	1
1	0	0	1	1	0	1
1	1	0	0	0	0	0

Example 2:

F(X,Y) = (X+Y).(X’+Y’)

Solution:

X	Y	X’	Y’	X+Y	X’+Y’	F
0	0	1	1	0	1	0
0	1	1	0	1	1	1
1	0	0	1	1	1	1
1	1	0	0	1	0	0

Example 3:

F(A,B,C) = A.B’ + B.C’ +A’.C

Solution:

A	B	C	A’	B’	C’	A.B’	B.C’	A’.C	F
0	0	0	1	1	1	0	0	0	0
0	0	1	1	1	0	0	0	1	1
0	1	0	1	0	1	0	1	0	1
0	1	1	1	0	0	0	0	1	1
1	0	0	0	1	1	1	0	0	1
1	0	1	0	1	0	1	0	0	1
1	1	0	0	0	1	0	1	0	1
1	1	1	0	0	0	0	0	0	0

Example 4:

F(X,Y,Z) = (X+Y)’.(X+Z)’

Solution:

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

Example 5:

$$F(A,B,C,D) = \left(\overline{A+B}\right).\overline{B}.\overline{C}.(A+D)$$

Solution:

A	B	C	D	B’	C’	(A+B)’	A+D	F
0	0	0	0	1	1	1	0	0
0	0	0	1	1	1	1	1	1
0	0	1	0	1	0	1	0	0
0	0	1	1	1	0	1	1	0
0	1	0	0	0	1	0	0	0
0	1	0	1	0	1	0	1	0
0	1	1	0	0	0	0	0	0
0	1	1	1	0	0	0	1	0
1	0	0	0	1	1	0	1	0
1	0	0	1	1	1	0	1	0
1	0	1	0	1	0	0	1	0
1	0	1	1	1	0	0	1	0
1	1	0	0	0	1	0	1	0
1	1	0	1	0	1	0	1	0
1	1	1	0	0	0	0	1	0
1	1	1	1	0	0	0	1	0

Example 6:

$$F(A,B,C,D) = \left(\overline{A+B}\right).\overline{B}.\overline{C}.(A+D)$$

Solution:

A	B	C	D	(A+B)’	B’	C’	A+D	F
0	0	0	0	1	1	1	0	0
0	0	0	1	1	1	1	1	1
0	0	1	0	1	1	0	0	0
0	0	1	1	1	1	0	1	0
0	1	0	0	0	0	1	0	0
0	1	0	1	0	0	1	1	0
0	1	1	0	0	0	0	0	0
0	1	1	1	0	0	0	1	0
1	0	0	0	0	1	1	1	0
1	0	0	1	0	1	1	1	0
1	0	1	0	0	1	0	1	0
1	0	1	1	0	1	0	1	0
1	1	0	0	0	0	1	1	0
1	1	0	1	0	0	1	1	0
1	1	1	0	0	0	0	1	0
1	1	1	1	0	0	0	1	0

C. Answer the following questions:

Example 1:

State the Absorption law and prove both the form of the law by using truth table.

Solution:

Absorption law states the following propositions: (i) A+A.B = A and (ii) A.(A+B) = A

A	B	A.B	A+A.B	A+B	A.(A+B)
0	0	0	0	0	0
0	1	0	0	1	0
1	0	0	1	1	1
1	1	1	1	1	1

Example 2:

State the Distributive law and prove any one form by using truth table.

Solution:

Distributive law states the following propositions: (i) A+B.C = (A+B).(A+C) (ii) A.(B+C) = A.B + A.C

A	B	C	B.C	A+B.C	A+B	A+C	(A+B).(A+C)	B+C	A.(B+C)	A.B	A.C	A.B+A.C
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0	1	0	0	0	0
0	1	0	0	0	1	0	0	1	0	0	0	0
0	1	1	1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	0	0	0	0	0
1	0	1	0	1	1	1	1	1	1	0	1	1
1	1	0	0	1	1	1	1	1	1	1	0	1
1	1	1	1	1	1	1	1	1	1	1	1	1

Example 3:

Write down the dual form of the given Boolean expression: $F(A,B,C) = (\overline{A + B + C}).A.\overline{B}.(\overline{A} + \overline{C})$

Solution: Dual of $F = (\overline{A.B.C}) + A + \overline{B} + (\overline{A}.\overline{C})$

Example 4:

Write down the dual form of the given Boolean expression: $F(A,B,C) = (A + B + C).(A + \overline{B}).(\overline{A}.\overline{C})$

Solution: Dual of $F = (A.B.C) + (A.\overline{B}) + (\overline{A} + \overline{C})$

Example 5:

Find the complement of $X.\left(Y.\overline{Z} + \overline{Y}.X + Y\right)$ using De Morgan's law. Show the relevant reasoning.

Solution:

F(X,Y,Z)' = (X.(Y.Z' + Y'.X + Y))'
= X' + (Y.Z' + Y'.X + Y)'
= X' + (Y.Z')' . (Y'.X)' . (Y)'
= X' + (Y' + Z).(Y + X'). Y'

Example 6:

1. State the output values and the Boolean expression of the followings:
   i. A(XOR)B(XOR)C, when A = 1, B = 0, C =1
   ii. NOT(X(OR)Y) AND NOT(Y(OR)Z), when X=1, Y=1, Z=1
   iii. A(AND)B(OR)C, when A=1, B = 0, C = 1
   iv. NOT(X(AND)Y) AND NOT(Y(OR)Z), when X = 1, Y = 1, Z = 1.

Solution:

i. 0 (For even no. of 1 $\rightarrow$ 0)
ii. 0 (NOT(1) AND NOT(1) $\rightarrow$ 0)
iii. A.B+C = 1
iv. $(\overline{X.Y}).(\overline{Y+Z}) = 0$

Example 7:

Give one similarity and one difference between XOR gate and OR gate.

Solution:

Similarity - Both the gates have two or more inputs and only one output.
Difference – XOR gate produces output 1 for those input combinations that have odd number 1's where as OR gate produces output 1 when all or any inputs are 1

Example 8:

Give one similarity and one difference between XOR gate and XNOR gate.

Solution:

The X in the XOR gate stands for "exclusive." This means that the output from this gate will be a 1 ONLY when one or the other of the inputs is a 1. In other words, this is an either-or gate The XNOR gate is a digital logic gate whose function is the inverse of the exclusive OR (XOR) gate. The two-input version implements logical equality, behaving according to the truth table to the right. A HIGH output (1) results if both of the inputs to the gate are the same. If one but not both inputs are HIGH (1), a LOW output (0) results.

D. Problems for practice

Example 1:

Simplify $(p+q)(\bar{p}+q)(\bar{p}+\bar{q})$ using the laws of Boolean Algebra. At each step state clearly the law used for simplification.

Solution:

(p+q).(p'+q).(p'+q')
($\underline{p.p'}$+p.q+p'.q+q.q).(p'+q')
(p.q+p'.q+q).(p'+q')
$\underline{p.q.p'}$+p'.q.p'+p'.q+$\underline{p.q.q'}$+$\underline{p'.q.q'}$+$\underline{q.q'}$ [underlined terms are removed by complement's law]
p'.q (Ans)

Example 2:

Determine if the following wff is valid, satisfiable or unsatisfiable: (P ⇒R).(Q⇒R) = (P+Q) ⇒R

Solution:

L.H.S. ⇨	R.H.S. ⇨
(p'+r).(q'+r)	(p+q)'+r
p'.q'+q'.r+p'.r+r	p'.q'+r
p'.q'+r(q'+p'+1)	
p'.q'+r	

Example 3:

Determine if the following wff is valid, satisfiable or unsatisfiable: (A+B)(A'+B') = A'B + AB'

Solution:

L.H.S. ⇨ (A+B)(A'+B')
A.A' + A.B' + B.A' B.B'
A.B' + A'.B ⇨ R.H.S.

Example 4:

Reduce the given Boolean expression F(X,Y,Z) = $X.\left(Y.\bar{Z}+\bar{Y}.X+Y\right)$ using Boolean laws. Mention the laws at each stage, wherever used.

X.(Y.Z'+Y'.X+Y)
= X.Y.Z'+X.Y'.X+X.Y
= X.Y.Z'+X.Y'+X.Y
= X.Y.Z'+X.(Y'+Y) = X.Y.Z'+X = X.(Y'Z+1) = X

Example 5:

Simplify the given Boolean expression by using Boolean laws. Also mention them at every stage.

Solution:

$F(a,b)=(a+b).(\bar{a}+b).(\bar{a}+\bar{b})$
= (a.a' + a.b + b.a' + b.b).(a' + b') …… by Complement's law a.a' = 0 and Idempotent law, b.b = b
= (a.b + a'.b + b).(a' + b')
= b.(a'+b') …….. by Absorption law, b+b.a+b.a' = b
= b.a' + b.b'
= a'.b

Example 6:

Simplify A'B + ABC' + ABC using the laws of Boolean Algebra. At each step state clearly the law used for simplification.

Solution:

F(A,B,C) = A'B + AB.(C+C')
= A'B + AB
= B(A'+A)
= B

Example 7:

Simplify the given Boolean expression by using Boolean laws. Also mention them at every stage.

Solution:

$F(A,B,C) = A.C + B.(\overline{B} + C).(A + B.C)$
= AC + (B.B' + B.C).(A+BC)
= AC + ABC + BC
= AC + BC (A+ 1)
= AC + BC

Solved examples

Q1. Write down the dual form of the given Boolean expression: $F(A,B,C) = (\overline{A+B+C}).A.\overline{B}.(\overline{A}+\overline{C})$

Dual of F = $(\overline{A.B.C}) + A + \overline{B} + (\overline{A}.\overline{C})$ **(Ans)**

Q2. Write down the dual form of the given Boolean expression: $F(A,B,C) = (A+B+C).(A+\overline{B}).(\overline{A}.\overline{C})$

Dual of F = $(A.B.C) + (A.\overline{B}) + (\overline{A}+\overline{C})$ **(Ans)**

Q3. Find the complement of $X.\left(Y.\overline{Z} + \overline{Y}.X + Y\right)$ using De Morgan's law. Show the relevant reasoning. F(X,Y,Z)' = (X.(Y.Z' + Y'.X + Y))' = X' + (Y.Z' + Y'.X + Y)'
= X' + (Y.Z')' . (Y'.X)' . (Y)'
= **X' + (Y' + Z).(Y + X'). Y' (Ans)**

Q4. State the output values and the Boolean expression of the followings:

i. A(XOR)B(XOR)C, when A=1, B=0, C=1
0 (For even no. of 1 $\rightarrow$ 0)

ii. NOT(X(OR)Y) AND NOT(Y(OR)Z), when X=1, Y=1, Z=1
0 (NOT(1) AND NOT(1) $\rightarrow$ 0)

iii. A(**AND**)B(**OR**)C, when A=1, B=0, C=1
A.B+C = 1

iv. **NOT**(X(**AND**)Y) **AND** NOT(Y(**OR**)Z), when X=1, Y=1, Z=1.
$(\overline{X.Y}).(\overline{Y+Z})$ = **0**

Practice problems:

1) What do you mean by Contingency, Tautology and Contradiction?
2) Find out the equivalent expression for (p⇔q) + (p=>q).
3) From premises p=>q and q=>p. Conclude q’ + p.q algebraically.
4) From premises p=>q and q=>r. Show (p=>q).(q=>r)=> (p=>r).
5) Draw the truth table the given Boolean functions:
   - (i) F (A, B, C) = AB+BC+CD
   - (ii) F (A, B, C) = (A’+B’+C’).(A+B’+C).(A+B+C’)
   - (iii) F (x,y) = x.y + x’.y’
   - (iv) F (x,y) = x.y’+ x’.y
6) Prove the Following algebraically:
   - (i) (A+B)(B+C)(C+A) = AB+BC+CA
   - (ii) (X+XY).(Y’+Y’X’) = XY’

7. Derivation of Boolean Expression

It is the process of converting a logical proposition into an expression containing some Boolean variables, their complements whenever necessary and the two binary operations i.e. logical addition (OR) and logical multiplication (AND).

7.1 Canonical Form of Boolean Expression

A canonical form of a Boolean expression is the logical sum of some min terms or logical product of some max terms.

7.2 Minterm

It is a Boolean function that is expressed as Sum of Product (SOP) of all variables. Min term is a product of all literals or their complements, present in the Boolean Function. In a min term, the value of a variable is taken to be 1 and its complement is 0.
Steps for finding Min term

1. Identify the missing literals and insert 1 for that term
2. Replace 1 by $X+\overline{X}$ by Complements law
3. Apply Distributive law
4. Apply Idempotent law

Let us take some examples –

Example 1:

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

Solution:

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

Step 2 = $X.(Y+\overline{Y})+(X+\overline{X}).\overline{Y}$ [by Complements law]

Step 3 = $X.Y+X.\overline{Y}+X.\overline{Y}+\overline{X}.\overline{Y}$ [by Distributive law]

Step 4 = $X.Y+X.\overline{Y}+\overline{X}.\overline{Y}$ [by Idempotent law]. This is the complete SOP expression.

Now if we follow the truth table, we will get the min term value

X	Y	MINTERM	DESIGNATION
0	0	$\overline{X}.\overline{Y}$	0
0	1	$\overline{X}.Y$	1
1	0	$X.\overline{Y}$	2
1	1	$X.Y$	3

= $\sum(0, 2, 3)$

Example 2:

$F(X,Y,Z) = X.Y+Z$

Solution:

Step 1 = $X.Y.1+1.1.Z$

Step 2 = $X.Y.(Z+\overline{Z})+(X+\overline{X}).(Y+\overline{Y}).Z$

Step 3 = $X.Y.Z+X.Y.\overline{Z}+X.Y.Z+\overline{X}.Y.Z+X.\overline{Y}.Z+\overline{X}.\overline{Y}.Z$

Step 4 = $X.Y.Z+X.Y.\overline{Z}+\overline{X}.Y.Z+X.\overline{Y}.Z+\overline{X}.\overline{Y}.Z$

X	Y	Z	MINTERM	DESIGNATION
0	0	0	$\overline{X}.\overline{Y}.\overline{Z}$	0
0	0	1	$\overline{X}.\overline{Y}.Z$	1
0	1	0	$\overline{X}.Y.\overline{Z}$	2
0	1	1	$\overline{X}.Y.Z$	3
1	0	0	$X.\overline{Y}.\overline{Z}$	4
1	0	1	$\overline{X}.Y.\overline{Z}$	5
1	1	0	$\overline{X}.\overline{Y}.Z$	6
1	1	1	$X.Y.Z$	7

= $\sum$(1, 3, 5, 6, 7)

7.3 Maxterm

It is a Boolean function that is expressed as Product of Sum (POS) of all variables. Max term is a sum of all literals or their complements, present in the Boolean Function. In a max term, the value of a variable is taken to be 0 and its complement is 1.

Steps for finding Max term

1. Identify the missing literals and insert 0 for that term
2. Replace 0 by $X.\overline{X}$ by Complements law
3. Apply Distributive law
4. Apply Idempotent law

Example 1:

F(X,Y) = X.Y

Solution:

Step 1 = (X+0).(0+Y)
Step 2 = (X+Y.Y').(X.X'+Y)
Step 3 = (X+Y).(X+Y').(X+Y).(X'+Y)
Step 4 = (X+Y).(X+Y').(X'+Y)

X	Y	MAXTERM	DESIGNATION
0	0	$X+Y$	0
0	1	$X+\overline{Y}$	1
1	0	$\overline{X}+Y$	2
1	1	$\overline{X}+\overline{Y}$	3

= $\prod$(0, 1, 2)

Example 2 :

F(X,Y,Z) = X.Y + Z

Solution:

= (X+Z).(Y+Z) , by applying Distributive law [this is because the initial expression must also be in the POS form]
= (X+0+Z).(0+Y+Z)
= (X+Y.Y'+Z).(X.X'+Y+Z)

= (X+Y+Z).(X+Y’+Z).(X+Y+Z).(X’+Y+Z)
= (X+Y+Z).(X+Y’+Z).(X’+Y+Z)

X	Y	Z	MINTERM	DESIGNATION
0	0	0	$X+Y+Z$	0
0	0	1	$\overline{X}+\overline{Y}+Z$	1
0	1	0	$\overline{X}+Y+\overline{Z}$	2
0	1	1	$X+\overline{Y}+\overline{Z}$	3
1	0	0	$\overline{X}+Y+Z$	4
1	0	1	$\overline{X}+Y+\overline{Z}$	5
1	1	0	$\overline{X}+\overline{Y}+Z$	6
1	1	1	$\overline{X}+\overline{Y}+\overline{Z}$	7

= $\prod$(0, 2, 4)

8. Karnaugh Map (K-MAP)

K-map is a graphical display of the fundamental products in a truth table. It is a representation or display of the information derived from a truth table of a Boolean function in a tabular or graphical form.

A reduced K-Map is obtained by filling up the cells by 1 or 0 as the resultant output value of the corresponding min term or max term.

In SOP form, each cell in K-map stands for a particular min term and in POS form each cell represents a particular max term.

Number of cells in a K-map depends up on the number of variables used. It is 2^n for n number of variables in the Boolean function.

A Karnaugh map (K-map) is a pictorial method used to minimize Boolean expressions without having to use Boolean algebra theorems and equation manipulations. A K-map can be thought of as a special version of a truth table.

Using a K-map, expressions with two to four variables are easily minimized. Expressions with five to six variables are more difficult but achievable, and expressions with seven or more variables are extremely difficult (if not impossible) to minimize using a K-map.

Characteristics of K-map:

- K-map represents the truth table or canonical forms.
- Its squares or cells are labeled such that adjacent squares differ by one variable change only.
- K-map squares are filled with 1s or 0s either from the truth table output values or min terms (max terms).
- SOP expression for the output is obtained by OR-ing the squares containing 1s and then simplifying by looping.
- POS expression for the output is obtained by AND-ing the squares containing 0s and then simplifying by looping.

8.1 Two-variables K-Map

X \ Y	$\overline{Y}$	Y
$\overline{X}$	$\overline{X}.\overline{Y}$ 0	$\overline{X}.Y$ 1
X	$X.\overline{Y}$ 2	$X.Y$ 3

2 variables K-map for min term

X \ Y	Y	$\overline{Y}$
X	$X+Y$ 0	$X+\overline{Y}$ 1
$\overline{X}$	$\overline{X}+Y$ 2	$\overline{X}+\overline{Y}$ 3

2 variables K-map for max term

8.2 Three-variables K-Map

X \ YZ	$\overline{Y}.\overline{Z}$	$\overline{Y}.Z$	$Y.Z$	$Y.\overline{Z}$
$\overline{X}$	$\overline{X}.\overline{Y}.\overline{Z}$ 0	$\overline{X}.\overline{Y}.Z$ 1	$\overline{X}.Y.Z$ 3	$\overline{X}.Y.\overline{Z}$ 2
X	$X.\overline{Y}.\overline{Z}$ 4	$X.\overline{Y}.Z$ 5	$X.Y.Z$ 7	$X.Y.\overline{Z}$ 6

3 variables K-map for min term

X \ YZ	$Y+Z$	$Y+\overline{Z}$	$\overline{Y}+\overline{Z}$	$\overline{Y}+Z$
X	$X+Y+Z$ 0	$X+Y+\overline{Z}$ 1	$X+\overline{Y}+\overline{Z}$ 3	$X+\overline{Y}+Z$ 2
$\overline{X}$	$\overline{X}+Y+Z$ 4	$\overline{X}+Y+\overline{Z}$ 5	$\overline{X}+\overline{Y}+\overline{Z}$ 7	$\overline{X}+\overline{Y}+Z$ 6

3 variables K-map for max term

8.3 Four-variables K-Map

XY \ ZW	$\overline{Z}.\overline{W}$	$\overline{Z}.W$	$Z.W$	$Z.\overline{W}$
$\overline{X}.\overline{Y}$	$\overline{X}.\overline{Y}.\overline{Z}.\overline{W}$ 0	$\overline{X}.\overline{Y}.\overline{Z}.W$ 1	$\overline{X}.\overline{Y}.Z.W$ 3	$\overline{X}.\overline{Y}.Z.\overline{W}$ 2
$\overline{X}.Y$	$\overline{X}.Y.\overline{Z}.\overline{W}$ 4	$\overline{X}.Y.\overline{Z}.W$ 5	$\overline{X}.Y.Z.W$ 7	$\overline{X}.Y.Z.\overline{W}$ 6
$X.Y$	$X.Y.\overline{Z}.\overline{W}$ 12	$X.Y.\overline{Z}.W$ 13	$X.Y.Z.W$ 15	$X.Y.Z.\overline{W}$ 14
$X.\overline{Y}$	$X.\overline{Y}.\overline{Z}.\overline{W}$ 8	$X.\overline{Y}.\overline{Z}.W$ 9	$X.\overline{Y}.Z.W$ 11	$X.\overline{Y}.Z.\overline{W}$ 10

4 variables K-map for min term

XY \ ZW	$Z+W$	$Z+\overline{W}$	$\overline{Z}+\overline{W}$	$\overline{Z}+W$
$X+Y$	$X+Y+Z+W$ 0	$X+Y+Z+\overline{W}$ 1	$X+Y+\overline{Z}+\overline{W}$ 3	$X+Y+\overline{Z}+W$ 2
$X+\overline{Y}$	$X+\overline{Y}+Z+W$ 4	$X+\overline{Y}+Z+\overline{W}$ 5	$X+\overline{Y}+\overline{Z}+\overline{W}$ 7	$X+\overline{Y}+\overline{Z}+W$ 6
$\overline{X}+\overline{Y}$	$\overline{X}+\overline{Y}+Z+W$ 12	$\overline{X}+\overline{Y}+Z+\overline{W}$ 13	$\overline{X}+\overline{Y}+\overline{Z}+\overline{W}$ 15	$\overline{X}+\overline{Y}+\overline{Z}+W$ 14
$\overline{X}+Y$	$\overline{X}+Y+Z+W$ 8	$\overline{X}+Y+Z+\overline{W}$ 9	$\overline{X}+Y+\overline{Z}+\overline{W}$ 11	$\overline{X}+Y+\overline{Z}+W$ 10

4 variables K-map for max term

9. Minimizing Boolean Expression through K-Map

For reducing the Boolean expression, we need to mark pairs, quads, and octets.

9.1 How to mark pair, quad, octet?

Pair is two adjacent cells in the K-map horizontally or vertically having same value (i.e 0 or 1).

XY \ ZW	$\bar{Z}.\bar{W}$	$\bar{Z}.W$	$Z.W$	$Z.\bar{W}$
$\bar{X}.\bar{Y}$	1			
$\bar{X}.Y$	1		1	1
$X.Y$				
$X.\bar{Y}$				

Quad is group of four adjacent cells in the K-map horizontally or vertically or four cells forming a square with 1 or 0.

XY \ ZW	$\bar{Z}.\bar{W}$	$\bar{Z}.W$	$Z.W$	$Z.\bar{W}$
$\bar{X}.\bar{Y}$	1			
$\bar{X}.Y$	1	1	1	1
$X.Y$	1	1	1	
$X.\bar{Y}$	1			

Octet is a group of eight cells in the K-map with either 1 or 0.

XY \ ZW	$\bar{Z}.\bar{W}$	$\bar{Z}.W$	$Z.W$	$Z.\bar{W}$
$\bar{X}.\bar{Y}$	1	1		
$\bar{X}.Y$	1	1	1	1
$X.Y$	1	1	1	1
$X.\bar{Y}$	1	1		

9.1.1 Rule for reducing a pair

Step 1 - Mark all the terms of a pair
Step 2 - Remove the variable which changes its state from complemented to un-complemented or vice versa. Here we can see, a pair can remove only 1 variable from the term.

9.1.2 Rule for reducing a quad

Step 1 - Mark all the terms of a quad
Step 2 - Remove the variable which changes its state from complemented to un-complemented or vice versa. Here we can see, a quad can remove 2 variables from the term.

9.1.3 Rule for reducing a octet

Step 1 - Mark all the terms of a octet
Step 2 - Remove the variable which changes its state from complemented to un-complemented or vice versa. Here we can see, a octet can remove 3 variables from the term.

9.2 Map Rolling

Map rolling means roll the map horizontally or vertically by considering the left edge of the map is touching the right edge or top edge is touching the bottom edge. This is a special property of K-map that its opposite edges squares and corner square are considered contiguous.

Rolling pairs (Horizontal & Vertical) - F(X,Y,Z,W)

XY \ ZW	$\overline{Z}.\overline{W}$	$\overline{Z}.W$	$Z.W$	$Z.\overline{W}$
$\overline{X}.\overline{Y}$		1		
$\overline{X}.Y$	1			1
$X.Y$				
$X.\overline{Y}$		1		

Rolling quads (Horizontal & Vertical)

XY \ ZW	$\overline{Z}.\overline{W}$	$\overline{Z}.W$	$Z.W$	$Z.\overline{W}$
$\overline{X}.\overline{Y}$		1	1	
$\overline{X}.Y$	1			1
$X.Y$	1			1
$X.\overline{Y}$		1	1	

Rolling quads (Four corners)

XY \ ZW	$\overline{Z}.\overline{W}$	$\overline{Z}.W$	$Z.W$	$Z.\overline{W}$
$\overline{X}.\overline{Y}$	1			1
$\overline{X}.Y$				
$X.Y$				
$X.\overline{Y}$	1			1

Rolling octet (Horizontal)

XY \ ZW	$\overline{Z}.\overline{W}$	$\overline{Z}.W$	$Z.W$	$Z.\overline{W}$
$\overline{X}.\overline{Y}$	1	1	1	1
$\overline{X}.Y$				
$X.Y$				
$X.\overline{Y}$	1	1	1	1

Rolling octet (Vertical)

XY \ ZW	$\overline{Z}.\overline{W}$	$\overline{Z}.W$	$Z.W$	$Z.\overline{W}$
$\overline{X}.\overline{Y}$	1			1
$\overline{X}.Y$	1			1
$X.Y$	1			1
$X.\overline{Y}$	1			1

9.3 Overlapping Groups

Overlapping means 1 can be encircled more than once in two or more pairs or quads or octet. Overlapping always leads to similar expression but to reduce the Boolean expression into its lowest form, we sometimes take the overlapping groups in to consideration.

9.4 Redundant Group

Redundant group is a group whose all 1's or 0's are overlapped by other (unavoidable) groups (i.e. pairs, quads, octets). We must remove the redundant groups from the expression to get correct Boolean expression as an answer.

9.5 To Summarize the rules for reducing S-O-P expression using K-Map

Step 1. Prepare the truth table for the given function
Step 2. Draw the K-map for the given function according to the number of variables in the expression (i.e. 2-variable K-map or 3-variable K-map or 4-variable K-map).
Step 3. Map the given function by entering 1's for the output as 1 in the corresponding cells.
Step 4. Encircle adjacent 1's in form of octet, quad or pairs as per the priorities. (Take the map rolling and overlapping groups in to consideration also).
Step 5. Remove the redundant group (if any).
Step 6. If any cell remains alone leave that as it is.
Step 7. Write the reduced expression for all the groups and OR (+) them together.

9.6 Some examples on S-O-P reduction:

(a) Given the following Boolean function :- **F (a, b, c, d) = Σ (0, 2, 6, 8, 10, 11, 14, 15)** Use Karnaugh's map to reduce this function F using the given SOP form.

	c'.d'	c'.d	c.d	c.d'
a'.b'	1			1
a'.b				1
a.b			1	1
a.b'	1		1	1

Quad(m_0,m_2,m_8,m_{10}) = b'.d' **[a and c are changing their state]**
Quad(m_2,m_6,m_{14},m_{10}) = c.d' **[a and b are changing their state]**
Quad(m_{11},m_{15},m_{14},m_{10}) = a.c **[b and d are changing their state]**

F = b'.d' + c.d' + a.c (Ans)

(b) Given the following Boolean function : F(a, b, c, d) = a'b'c'd' + a'b'c'd + ab'c'd' + ab'c'd + a'bc'd' + a'bcd'. Write down the Minterm value for the above function. Use Karnaugh's map to reduce this function.

cd / ab	c'd'	c'd	cd	cd'
a'b'	1	1		
a'b	1			1
ab				
ab'	1	1		

Min term of the given function: Σ **(0,1,4,6,8,9)**

Quad(m_0,m_1,m_8,m_9) = b'.c' **[a and d are changing their state]**
Pair(m_4,m_7) = a'.b.d' **[c is changing its state]**

$\boxed{\text{F(a,b,c,d) = b'.c' + a'.b.d' (Ans)}}$

9.6 To Summarize the rules for reducing P-O-S expression using K-Map

Step 1. Prepare the truth table for the given function
Step 2. Draw the K-map for the given function according to the number of variables in the expression (i.e. 2-variable K-map or 3-variable K-map or 4-variable K-map).
Step 3. Map the given function by entering 0's for the output as 0 in the corresponding cells.
Step 4. Encircle adjacent 1's in form of octet, quad or pairs as per the priorities. (Take the map rolling and overlapping groups in to consideration also).
Step 5. Remove the redundant group (if any).
Step 6. If any cell remains alone leave that as it is.
Step 7. Write the reduced expression for all the groups and AND (.) them together.

9.7 Some examples on P-O-S reduction

(a) Given the following Boolean function :- **F (a, b, c, d) = Π (5, 7, 8, 10, 12, 14, 15)**
Use Karnaugh's map to reduce this function F using the given POS form.

	c+d	c+d'	c'+d'	c'+d
a+b				
a+b'		0	0	
a'+b'	0		0	0
a'+b	0			0

Quad(m_8,m_{12},m_{14},m_{10}) = a'+d **[b and c are changing their state]**
Pair(m_5,m_7) = a+b'+d' **[c is changing its state]**
Pair(m_7,m_{15}) = b'+c'+d' **[a is changing its state]**

F = (a'+c').(a+b'+d').(b'+c'+d') (Ans)

(b) Given the following Boolean function :- F (a, b, c, d) = Π (2, 3, 6, 7, 9, 11, 12, 13, 14, 15). Write down the expanded Boolean expression for the above function. Use Karnaugh's map to reduce this function F.

ab \ cd	c+d	c+d'	c'+d'	c'+d
a+b			0	0
a+b'			0	0
a'+b'	0	0	0	0
a'+b		0	0	

Quad(m_{12},m_{13},m_{15},m_{14}) = (a'+b') **[c and d are changing their state]**
Quad(m_3,m_2,m_7,m_6) = (a+c') **[b and d are changing their state]**
Quad(m_{13},m_{15},m_9,m_{11}) = (a'+d') **[b and c are changing their state]**

$\boxed{\text{F(a,b,c,d) = (a'+b').(a+c').(a'+d') (Ans)}}$

Practice problems

1. *Given the following Boolean function :- F (a, b, c, d) = Σ (5, 6, 7, 8, 9, 10, 14) , Use Karnaugh's map to reduce this function F using the given SOP form.*

   Answer:

ab\cd	*c'd'*	*c'd*	*cd*	*cd'*
a'b'				
a'b		*1*	*1*	*1*
ab				*1*
ab'	*1*	*1*		*1*

   F(a,b,c,d) = a'bd + a'bc + ab'c' + acd'

2. *Given the following Boolean function :- F (a, b, c, d) = Π (0, 2, 6, 8, 10, 14), Use Karnaugh's map to reduce this function F using the given POS form.*

   Answer:

ab\cd	c+d	c+d'	c'+d'	c'+d
a+b	0			0
a+b'				0
a'+b'				0
a'+b	0			0

   F(a,b,c,d) = (b+d).(c'+d)

3. Given the following Boolean function :- F (a, b, c, d) = Σ (1, 2, 3, 5, 6, 10, 11), Use Karnaugh's map to reduce this function F using the given SOP form.

   Answer:

ab\cd	c'd'	c'd	cd	cd'
a'b'		1	1	1
a'b		1		1
ab				
ab'			1	1

   F(a,b,c,d) = a'c'd + a'cd' + b'c

4. *Given* the following Boolean function :- **F (a, b, c, d) = Π (2, 3, 4, 5, 8, 9, 12, 14, 15),** Use Karnaugh's map to reduce this function F using the given POS form.

   Answer:

ab\cd	c+d	c+d’	c'+d’	c'+d
a+b			0	0
a+b’	0	0		
a'+b'	0		0	0
a'+b	0	0		

F(a,b,c,d) = (a+b+c’) . (a+b’+c) . (a’+b’+c’) . (a’+b+c) . (b’+c+d) . (a’+c+d)

5. *Given* the following Boolean function :- F (a, b, c, d) = Σ (0,5,7,8,10,12,13,14,15). Use Karnaugh’s map to reduce this function F using the given SOP form. Draw the logic gate diagram for the reduced expression using AND & OR gates only.

Answer:

AB \ CD	C’D’	C’D	CD	CD’
A’B’	1			
A’B		1	1	
AB	1	1	1	1
AB’	1			1

quad1 = B.D
quad2 = A.D’
pair = B’.C’.D’
F(A,B,C,D) = B.D + A.D’ + B’.C’.D’

6. *Given* the following Boolean function :- F (a, b, c, d) = Π (4,6,7,10,11,12,14,15). Use Karnaugh’s map to reduce this function F using the given POS form. Draw the logic gate diagram for the reduced expression using AND & OR gates only.

Answer:

A+B \ C+D	C+D	C+D’	C’+D’	C’+D
A+B				
A+B’	0		0	0
A’+B’	0		0	0
A’+B			0	0

quad1 = B’+D
quad2 = B’+C’
quad3 = A’+C’
F(A,B,C,D) = (B’+D).(B’+C’).(A’+C’)

7. A factory needs a minimum of 1200 tons of raw material and at least 100 workers to start its production. There are 3 suppliers each agreeing to supply 600, 800 and 1250 tons of raw materials respectively.
A = 1 if the first supplier supplies else it is 0 - 600 tons

B = 1 if the second supplier supplies else it is 0 - 800 tons
C = 1 if the third supplier supplies else it is 0 - 1250 tons
D = 1 if 100 workers are available else it is 0
R = 1 if production starts else it is 0

(a) Draw truth table for the problem stated above and derive its SOP expression.

(b) Reduce the above SOP expression using the K- Map. Also draw logic circuit diagram for the reduced SOP expression using NAND gates only.

Answer:

Truth Table for the problem in as given below

A	B	C	D	R	Minterm	Desig
0	0	0	0			0
0	0	0	1			1
0	0	1	0			2
0	0	1	1	1	A'B'CD	3
0	1	0	0			4
0	1	0	1			5
0	1	1	0			6
0	1	1	1	1	A'BCD	7
1	0	0	0			8
1	0	0	1			9
1	0	1	0			10
1	0	1	1	1	AB'CD	11
1	1	0	0			12
1	1	0	1	1	ABC'D	13
1	1	1	0			14
1	1	1	1	1	ABCD	15

R(A,B,C,D) = A'B'CD + A'BCD + AB'CD + ABC'D + ABCD
= $\sum$(3, 7, 11, 13, 15)

K-Map representation of the above minterm

AB/CD	C'D'	C'D	CD	CD'
A'B'			1	
A'B			1	
AB		1	1	
AB'		1	1	

quad1 = CD
quad2 = AD

R(A,B,C,D) = AD+CD

8. In a local survey it was found that a person cannot accept a mobile phone call when –
- He is in office and is in a meeting
- He is not in office but is driving
- He is driving and is having a cake
- He is not in office but having a cake and is in a meeting

Inputs are: F : The person is in office
D : The person is driving
M: The person is in a meeting

C : The person is having a cake [1 indicates yes, 0 indicates no.]

Output is: N: The person cannot accept the mobile phone call. [1 indicates yes, 0 indicates no.]

(a) Draw the truth table and write the Sum Of Product expression for N(F, D, M, C).

(b) Reduce N(F, D, M, C) using Karnaugh's map. Draw the logic gate diagram for the reduced expression for N(F, D, M, C) using AND, OR, NOT gates.

Answer:

Truth Table for the problem in as given below

F	D	M	C	N	Minterm
0	0	0	0		
0	0	0	1		
0	0	1	0		
0	0	1	1	1	F'D'MC
0	1	0	0	1	F'DM'C'
0	1	0	1	1	F'DMC'
0	1	1	0	1	F'DMC'
0	1	1	1	1	F'DMC
1	0	0	0		
1	0	0	1		
1	0	1	0	1	FD'MC'
1	0	1	1	1	FD'MC
1	1	0	0		
1	1	0	1	1	FDM'C
1	1	1	0	1	FDMC'
1	1	1	1	1	FDMC

N(F,D,M,C) = ∑(3,4,5,6,7,10,11,13,14,15)

K-Map representation of the above minterm

FD\MC	M'C'	M'C	MC	MC'
F'D'			1	
F'D	1	1	1	1
FD		1	1	1
FD'			1	1

quad1 = F'D
quad2 = MC
quad3 = DC
quad4 = DM
quad5 = FM

N(F,D,M,C) = F'D + MC + DC + DM + FM

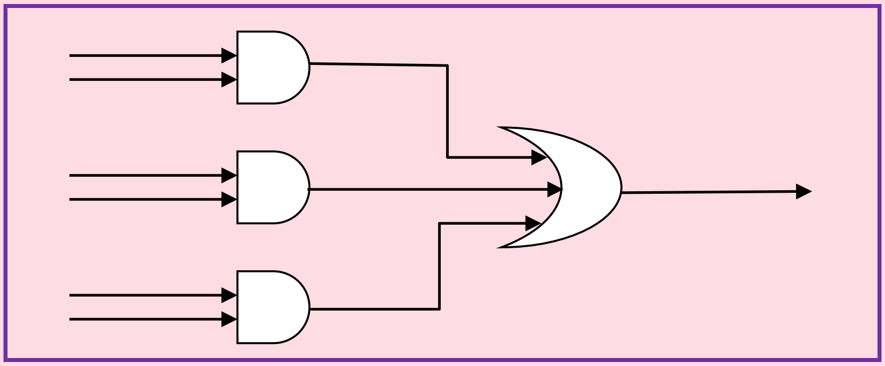
9. F(A,B,C,D) = ∑(4,6,7,10,11,12,14,15)

FD\MC	M'C'	M'C	MC	MC'
F'D'			1	
F'D	1	1	1	1
FD		1	1	1
FD'			1	1

There are 3 quads –

Q1 (m4,m6,m12,m14)= BD’
Q2 (m6,m7,m14,m15) = BC
Q3 (m10,m11,m14,m15) = AC

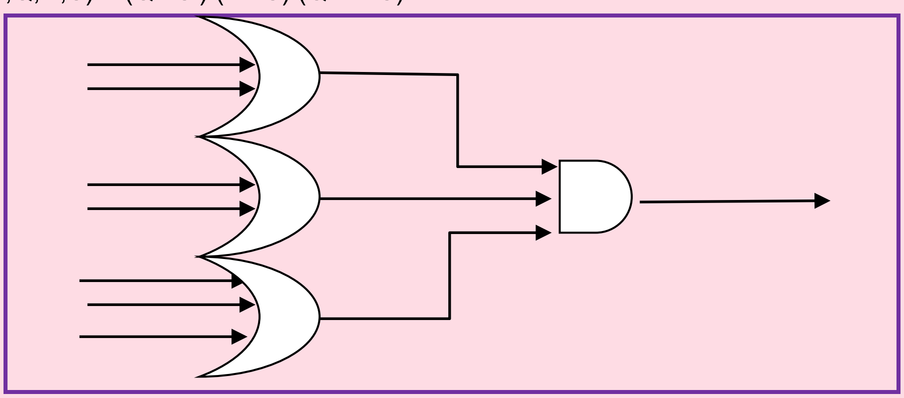
F(A,B,C,D) = BD’+BC+AC



10. F(P,Q,R,S) = ∏(0,5,7,8,10,12,13,14,15)

	R+S	R+S’	R’+S’	R’+S
P+Q	0			
P+Q’		0	0	
P’+Q’	0	0	0	0
P’+Q	0			0

There are 2 quads and a pair –
Q1(m5,m7,m13,m5)= Q’+S’
Q2 (m8,m10,m12,m14) = P’+S
P1(m0,m8) = Q+R+S
F(P,Q,R,S) = (Q’+S’).(P’+S).(Q+R+S)



11.

S	P	C	T	X	Min term	Desig
0	0	0	0	0		0
0	0	0	1	1	S'P'C'T	1
0	0	1	0	0		2
0	0	1	1	1	S'P'CT	3
0	1	0	0	1	S'PC'T'	4
0	1	0	1	1	S'PC'T	5
0	1	1	0	1	S'PCT'	6
0	1	1	1	1	S'PCT	7
1	0	0	0	0	SP'C'T'	8
1	0	0	1	0	SP'C'T	9
1	0	1	0	1	SP'CT'	10
1	0	1	1	1	SP'CT	11
1	1	0	0	1	SPC'T'	12
1	1	0	1	1	SPC'T	13
1	1	1	0	1	SPCT'	14
1	1	1	1	1	SPCT	15

X(S,P,C,T)=∑(1,3,4,5,6,7,9,10,11,12,13,14,15)

	C'T'	C'T	CT	CT'
S'P'		1	1	
S'P	1	1	1	1
SP	1	1	1	1
SP'			1	1

There are 1 octet and 2 quads –
O1(m4,m5,m7,m6,m12,m13,m15,m14) = P
Q1(m1,m3,m5,m7) = S'T
Q2(m10,m11,m14,m15) = SC
X(S,P,C,T) = P+S'T+SC

MCQ on Boolean Algebra and Logic Gates

1. Boolean algebra is also called
A. switching algebra
B. arithmetic algebra
C. linear algebra
D. algebra

2. To perform product of max-terms Boolean function must be brought into
A. AND terms
B. OR terms
C. NOT terms
D. NAND terms

3. x.(x+y) equal to
A. x'
B. 1
C. x
D. 0

4. A Boolean function may be transformed into
A. logical diagram
B. logical graph
C. map
D. matrix

5. e*x=x*e=x is the
A. commutative property
B. inverse property
C. associative property
D. identity element

6. Min-terms are also called
A. standard sum
B. standard product
C. standard division
D. standard subtraction

7. Max-terms are also called
A. standard sum
B. standard product
C. standard division
D. standard subtraction

8. x+x.y=x is known as
A. inverse law
B. commutative law
C. distributive law
D. absorption law

9. A two-variable Boolean algebra is defined as a set of
A. three values
B. two values
C. four values
D. five values

10. Most preceded operator is
A. parenthesis
B. AND
C. OR
D. NOT

11. (a+b+c)' = ?
A. a'.b'.c'
B. a'+b'+c'
C. a.b.c
D. a+b+c

12. x+x'y = ?
A. x
B. y
C. x-y
D. x+y

13. The primed or unprimed variable is
A. map
B. logic gates
C. literal
D. graph

14. A binary variable can take the values
A. 0 only
B. 0 and -1
C. 0 and 1
D. 1 and 2

15. According to the Boolean algebra theorems x.x is equal to
A. x
B. 1
C. 0
D. x'

16. One that is not the postulate (law) of Boolean algebra
A. commutative
B. duality
C. associative
D. identity element

17. Symbol representing AND operation
A. (+)
B. (.)
C. (-)
D. (/)

18. Boolean algebra is an algebraic structure with two arithmetic operations
A. addition and subtraction
B. subtraction and multiplication
C. addition and multiplication
D. addition and division

19. 2^3 would have
A. three values
B. four values
C. six values
D. eight values

20. Is it possible to find two algebraic expressions that specify the same function?
A. no
B. yes
C. maybe
D. never

21. x+x.y = ?
A. y
B. 1
C. 0
D. x

22. x+y=y+x is the
A. commutative property
B. inverse property
C. associative property
D. identity element

23. Boolean algebra is the collection of objects having
A. positive properties
B. negative properties
C. common properties
D. different properties

A B C A D B A D B A A D C C A B B C D B D A C

POINTS TO MEMORIES:

Q1. What are the different types of connectives used in propositional logic?

- **Disjunction (also called OR).** It is represented by + or $\vee$. Disjunction means one of the two arguments must be true or both to give the output of the compound proposition as true. E.g – x OR y, represented by x + y or $x \vee y$, it means "either it is hot today or it is raining outside".
- **Conjunction (also called AND).** It is represented by . or $\wedge$. Conjunction means both of the two arguments must be true to give the output of the compound proposition as true. E.g – x AND y, represented by x . y or $x \wedge y$, it means "It is hot today and it is raining outside".
- **Conditional (also called IF..THEN or Implication).** It is represented by $\Rightarrow$ or $\supset$. Conditional or Implication means if one argument is true then other argument is true. E.g – IF x THEN y, represented by $x \Rightarrow y$ or $x \supset y$, it means "if it is hot today then it is raining outside". [It is to be noted that in this case if x is false but y is true then also the compound proposition $x \Rightarrow y$ holds true]
- **Bi-conditional (also called IF and ONLY IF or Equivalence).** It is represented by $\Leftrightarrow$ or $\equiv$. Bi-conditional or equivalence means if and only if one argument is true then other argument is true. E.g – IF AND ONLY IF x THEN y, represented by $x \Leftrightarrow y$ or $x \equiv y$, it means "if and only if it is hot today then it is raining outside".
- **Negation (also called NOT).** It is represented by $'$ or ~ or – It is just a unary operator that inverse the proposition. E.g – NOT x, represented by x' or ~x or $\overline{x}$.

Q2. Define the following terms in Truth Table :

- **CONTIGENCIES** - The propositions that have some combination of 1's and 0's in their truth table output column are called contingencies..
- **CONTRADICTION** - The propositions that have all combination of 0's in their truth table output.
- **TAUTOLOGIES** - The propositions that have all combination of 1's in their truth table output.
- **CONVERSE** – The converse of a conditional proposition is determined by interchanging the antecedent and consequent of the given condition. E.g – Converse of $p \Rightarrow q$ is $q \Rightarrow p$.
- **INVERSE** – The inverse of a conditional proposition is another conditional having negated antecedent and consequent. E.g – Inverse of $p \Rightarrow q$ is $\overline{p} \Rightarrow \overline{q}$.
- **CONTRAPOSITIVE** – It is formed by creating another conditional that takes its antecedent as negated consequent of earlier conditional and consequent as negated antecedent of earlier conditional.
- **CONSISTENT** – Two statements are consistent if and only if their conjunction is not a contradiction.

What is satisfiable and unsatisfiable propositional logic?

A propositional logic is said to be satisfiable if it is either toutology or contingency and a logic which is only contradiction is said to be unsatisfiable.

Q3. What do you mean by Principle of duality?

Principle of duality states that in an identity,

i. if all the ANDs are replaced with OR and vice versa or
ii. all the 1s are replaced with 0s and vice versa, the identity remains unchanged.

Q4. What is Karnaugh Map (K-MAP) ?

K-map is a graphical display of the fundamental products in a truth table. It is a representation or display of the information derived from a truth table of a Boolean function in a tabular or graphical form.
A reduced K-Map is obtained by filling up the cells by 1 or 0 as the resultant output value of the corresponding min term or max term.
In SOP form, each cell in K-map stands for a particular min term and in POS form each cell represents a particular max term.

Q5. How to mark pair, quad, octet in a K-Map?
Pair is two adjacent cells in the K-map horizontally or vertically having same value (i.e 0 or 1).
Quad is group of four adjacent cells in the K-map horizontally or vertically or four cells forming a square with 1 or 0.
Octet is a group of eight cells in the K-map with either 1 or 0.

Q6. What do you mean by Map Rolling?
Map rolling means roll the map horizontally or vertically by considering the left edge of the map is touching the right edge or top edge is touching the bottom edge. This is a special property of K-map that its opposite edges squares and corner square are considered contiguous.

Q7. Give one similarity and one difference between XOR gate and OR gate.
Ans: Similarity - Both the gates have two or more inputs and only one output.
Difference – XOR gate produces output 1 for those input combinations that have odd number 1's where as OR gate produces output 1 only when all inputs are 1

Q8. Simplify $(p+q)(\overline{p}+q)(\overline{p}+\overline{q})$ using the laws of Boolean Algebra. At each step state clearly the law used for simplification.
Solution: (p+q).(p'+q).(p'+q')
$(\underline{p.p'}+p.q+p'.q+q.q).(p'+q')$
(p.q+p'.q+q).(p'+q')
$\underline{p.q.p'}+p'.q.p'+p'.q+\underline{p.q.q'}+\underline{p'.q.q'}+\underline{q.q'}$ [underlined terms are removed by complement's property]
p'.q (Ans)

Q9.Determine if the following wff is valid, satisfiable or unsatisfiable: $(P \Rightarrow R).(Q \Rightarrow R) = (P+Q) \Rightarrow R$

Solution: L.H.S. ⇨ (p'+r).(q'+r)
p'.q'+q'.r+p'.r+r
p'.q'+r(q'+p'+1)
p'.q'+r
R.H.S. ⇨ (p+q)'+r
p'.q'+r

UNSOLVED EXERSICE
1. Short Answer Questions
1. Simplify A.B+A'.C+B.C using the laws of Boolean Algebra. At each step state clearly the law used for simplification.
2. Why is the NOR gate regarded as Universal Gate? Draw the logic gate symbol and make the truth table for the two input NOR gate.
3. State Absorption Laws. Verify one of the Absorption laws using a truth table.

4. Using Boolean Algebra show that the dual of exclusive OR is equivalent to the complement of exclusive OR.
5. Find the complement of X. (Y.Z' + Y'.X) using De Morgan's law. Show the relevant reasoning. Do not reduce the function.
6. Define Absorption Law. Prove it with help of truth table.
7. Use De Morgan's Law to find the compliment of the following. Can it be represented by a single gate? If yes, name it.
8. State De Morgan's Law of Boolean Algebra and verify one of the laws using truth tables.
9. Explain XOR Gate with help of truth table of three inputs.
10. Obtain the simplified form for a Boolean Expression.
    F(a,b,c,d)=Σ(1,2,3,11,12,14,15) using Karnaugh Maps.
11. Write the dual of **(A'+B.1).((A.B'+C).D)+A'B**
12. Find the dual of $p'qr'+pq'r+p'q'r$ is equal to the complement of $pq'r+q.(p'r'+pr')$ or not. Show working.
13. Prove the dual of XNOR gate is equivalent to XOR gate by using truth table.
14. Draw the logic diagram for F using NAND gate only, where F(A,B,C,D) = $\left(\overline{A.B}\right)+\overline{B}.\overline{C}+\left(A.D\right)$
15. Draw a truth table with a 3 input combination which outputs 1 if there are odd number of 1's. Also derive an SOP expression for the same.

2. Solve using Karnaugh Map

1. Given the following Boolean function :- F (a, b, c, d) = Σ (7, 9, 10, 11, 12, 13, 14, 15). Use Karnaugh's map to reduce this function F using the given SOP form. Draw the logic gate diagram for the reduced SOP expression using AND and OR gates with two or more inputs.
2. Given the following Boolean function :- F (a, b, c, d) = Π (0, 1, 3, 5, 6, 7, 10, 14, 15). Use Karnaugh's map to reduce this function F using the given POS form. Draw the logic gate diagram for the reduced POS expression using AND and OR gates with two or more inputs.
3. Given the Boolean function F(A,B,C,D) = Σ (1,6,7,8,9,10,14,15). Use Karnaugh's map to reduce this function F, using the given SOP form. Draw the logic gate diagram for the reduced SOP form. You may use gates with more than two inputs Assume that variables and their complements are available as inputs.
4. Given D(A,B,C,D) = π (0,2,5,7,8,10,13,15). Use Karnaugh's map to reduce this function D using the given POS form. Draw the logic gate diagram for the reduced POS form. You may use gates with more than two inputs. Assume that variables and their complements are available as inputs
5. Given the following Boolean function :- **F (a, b, c, d) = Σ (1, 2, 3, 5, 6, 10, 11).**
   Use Karnaugh's map to reduce this function F using the given SOP form. Draw the logic gate diagram for the reduced expression using NAND gates only.
6. Given the following Boolean function :- **F (a, b, c, d) = Π (2, 3, 4, 5, 8, 9, 12, 14, 15).**
   Use Karnaugh's map to reduce this function F using the given POS form. Draw the logic gate diagram for the reduced expression using NOR gates only.
7. Given the following Boolean function :- **F (a, b, c, d) = Σ (0,5,7,8,10,12,13,14,15).**
   Use Karnaugh's map to reduce this function F using the given SOP form. Draw the logic gate diagram for the reduced expression using AND & OR gates only.

8. Given the following Boolean function :- **F (a, b, c, d) = Π (4,6,7,10,11,12,14,15).** Use Karnaugh's map to reduce this function F using the given POS form. Draw the logic gate diagram for the reduced expression using AND & OR gates only.

3. Karnaugh Map problems: -

1. The inaugural function of the newly constructed flyover has been organized by the Public Works Department. Apart from a few special invitees, entry is permitted only if:

   The person is an employee of the PWD of Class I category with more that 10 years of working experience

   OR

   The person is an employee of any other government or authorized private organization either at the Managerial level or with more than 10 years of working experience.

   Inputs are:- A: The person is a Class I employee of PWD

   B: The person is an employee of any other government or authorized private organization

   C: The person has more than 10 years of working experience

   D: The person is holding a Managerial post.

   Output:- R – denotes eligible for entry [1 indicates Yes and 0 indicates No]

   a. Draw the truth table for the inputs and outputs given above and write the SOP expression for the result of the motion R (A, B, C, D).

   b. Reduce R using Karnaugh's map. Draw the logic gate diagram of the above result R using the NAND gate and show how it is equivalent to the original circuit.

2. An Insurance company issues a policy to an applicant only when the applicant satisfies at least one of the following conditions:

   The applicant is a married male of age 25 years or above.

   The applicant is a female who never had a car accident.

   The applicant is a married female had has had a car accident.

   The applicant is a male below 25 years.

   The applicant is not below 25 years and has never had a car accident.

INPUTS ARE:

M:- The applicant is married	(1 indicates yes and 0 indicates no)
S:- The applicant is a male	(1 indicates yes and 0 indicates no)
C:- The applicant has had a car accident.	(1 indicates yes and 0 indicates no)
Y:- The applicant is below 25 years	(1 indicates yes and 0 indicates no)

OUTPUT IS:

I:- Denotes, Issue the Policy (1 indicates it is issued and 0 indicates it is not issued)

a. Draw the truth table for the inputs and outputs given above. Write the SOP expression for I(M, S, C, Y).

b. Reduce I(M, S, C, Y) using Karnaugh's map.

c. Draw the logic gate diagram for the reduced SOP expression for I(M, S, C, Y) using AND & OR gates. You may use gates with two or more inputs. Assume that variable and their complements are available as inputs.

3. **A** factory needs a minimum of 1200 tons of raw material and at least 100 workers to start its production. There are 3 suppliers each agreeing to supply 600, 800 and 1250 tons of raw materials respectively.

A = 1 if the first supplier supplies else it is 0

B = 1 if the second supplier supplies else it is 0

C = 1 if the third supplier supplies else it is 0

D = 1 if 100 workers are available else it is 0

R = 1 if production starts else it is 0

Taking A,B,C,D as inputs and R as output (i) Draw truth table for the problem stated above and derive its SOP expression. (ii) Reduce the above SOP expression using the K-Map. Also draw logic circuit diagram for the reduced SOP expression using NAND gates only.

Some extra problems:

Q1. Explain the following terms:
(i) Proposition logic
(ii) Truth Table
(iii) Logical Function
(iv) Principle of Duality
(v) Grey Code
(vi) Pairing in K-Map
(vii) Map-rolling in K-Map
(viii) Karnaugh map
(ix) Map rolling
(x) Universal gates

Q2. Prove the following by applying Boolean theorems:
(i) (x + y + z) . (x' + y + z) = y + z
(ii) ((p'+q) . (q'+r))' + p' + r =1

Q3.Write the complement of the following using De Morgan's theorem
(i) F(a,b,c,d) = ab'c + bc'd' + a'c'
(ii) F(A,B,C.D) = $A.B + \left(A + \overline{B}\right).B$

Q4. Write the dual of the following Boolean results:
(i) A.B(A+B') = AB
(ii) (X+1)(0+Y)(Z+1+0) = Y

Q5. Find the Min term/Max term for the corresponding SOP/POS expression and draw the Karnaugh map for each of them:

(i) F(A,B,C,D) = $\left(\overline{(A+B).(\overline{B}+C).D}\right)$ (find min term)

(ii) F(A,B,C) = $\left(A+\overline{B}\right)\left(\overline{A}+C\right)B$ (find max term)

Q6. By using Karnaugh map, obtain the simplified form of the following functions and draw the logic diagram of the reduced expression using AND and OR gates only

(i) F(x,y,z,w) = $\sum$ (0,2,4,6,8,9,12,13)

(ii) F(w,x,y,z) = Π (0,1,3,5,6,7,10,14,15)

(iii) F(A,B,C,D) = Σ (1,6,7,8,9,10,14,15)

(iv) F(A,B,C,D) = $\prod$ (0,2,5,7,8,10,13,15)

Q7. Draw a logic diagram of the following Boolean expression using NAND and NOR gates only

(i) F(a,b,c,d) = a.b + b.c + d (using NAND gate)

(ii) F(x,y,z) = (x+y).(y+z') (using NOD gate)

Q8. (i) Show that the dual of the exclusive OR is equal to its complement.

(ii) Find the complement of F= X + Y.Z; then show that F.F'=0 and F+F'=1