An English Mathematician, GEORGE BOOLE gave the concept of Boolean algebra/algebra of logic in his revolutionary papers titled "The mathematical analysis of logic" and "An investigation of the laws of thought" in years 1847 and 1854 respectively. His work in these two papers was applied by **Claude Shanon** to design electrical circuits. Further development of Boolean algebra led to the birth of Modern High Speed Digital Computers.

Boolean Constant: False (0) or True (1) values are known as Boolean Constants/Logical Constants/Truth Values.

Boolean Statement: A statement is said to be a Boolean/Logical Statement if it has a definite value, which is either false or true.

Examples:

Logical Statement	**Non Logical Statement**
It is raining outside.	What is your name?
The door is open.	Who is Ms. Jamson?
New Delhi is capital of India.	Where is your car parked?

Boolean operators: Operators used in Boolean algebra are known as Boolean/ logical operators. Basic logical operators and their notations are shown below:

	AND	**OR**	**NOT**
	.	+	'
	$\wedge$	$\vee$	~
Examples	X . Y	X + Y	X'
	X $\wedge$ Y	X $\vee$ Y	~X

Boolean Variable: A variable, which holds false/true value, is known as Boolean variable (X, Y, Z etc.).

Boolean Expression: A meaningful combination of Boolean operators (AND/OR/NOT), Boolean operand/variable (X, Y, Z etc.) and Boolean constant (0 or 1) is known as Boolean Expression (Logical Expression).

Examples:

X + Y.Z
A . (B+C) + B.C′
U OR V AND NOT Z

Boolean Algebra: Boolean algebra is an algebraic structure on a set **B** together with Boolean operators · (AND), + (OR) & ' (NOT) with the following postulates satisfied.

- **Closure Property** : If **X** & **Y** are two Boolean variables
  then
  $X+Y \in B$
  $X.Y \in B$
  $X' \in B$
- **Commutative Property:** If **X** & **Y** are two Boolean variables
  then
  $X + Y = Y + X$
  $X . Y = Y . X$
- **Associative Property** : If **X, Y** & **Z** are three Boolean variables
  then
  $X + (Y + Z) = (X + Y) + Z$
  $X . (Y . Z) = (X . Y) . Z$

- **Existence of identity** : For every $X \in B$ there exists
  $0 \in B$ such that $X + 0 = X$
  $1 \in B$ such that $X . 1 = X$
- **Existence of inverse** : For every $X \in B$
  there exists $X' \in B$
  such that $X + X' = 1$
  $X . X' = 0$

Duality Principle: It states that any law/theorem in Boolean algebra remains unchanged if the following changes are done simultaneously.

- Change all + to **.** & vice versa
- Change all **0** to **1** & vice versa

[REMEMBER: Order of execution of various parts of the expression remains unchanged]

For example :

a) $X + Y = Y + X$ **Commutative Law**
Dual $X . Y = Y . X$ **Commutative Law**

b) $X . Y + 0$ is **dual** of $(X + Y) . 1$

c) $(X'+Y).(X+Y')$ is **dual** of $X'.Y + X.Y'$

Laws/Theorems of Boolean Algebra

1. Distributive : For every $X , Y , Z \in B$

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

$X + Y . Z=(X + Y).(X + Z)$ (by **Duality**)

Generalised Form:

$X.(Y1 + Y2 + \ldots +Yn) = X . Y1 + X . Y2 + \ldots + X . Yn$

$X + Y1.Y2. \ldots .Yn =(X + Y1).(X + Y2). \ldots . (X + Yn)$

2. Demorgan's : For every $X,Y \in B$

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

$(X . Y)'= X' + Y'$ (by **Duality**)

Generalised Form:

$(X1 + X2 + X3 + \ldots + Xn)'= X1'.X2'.X3' \ldots .Xn'$

$(X1 . X2 . X3 . \ldots . Xn)'= X1' + X2' + X3'+ \ldots +Xn'$

3. Idempotent : For every $X \in B$

$X + X = X$

$X . X = X$ (by **Duality**)

Generalised Form:

$X + X + X + \ldots + X = X$

$X . X . X . \ldots . X = X$

4. Involution /Complementation Law : For every $X \in B$

$(X')'= X$

5. Absorption Law : For every $X , Y \in B$

i) $X + X . Y = X$
$X . (X + Y)= X$ (by **Duality**)

ii) $X + X' . Y = X + Y$
$X . (X'+ Y)= X . Y$(by **Duality**)

6. **Dominance of 0 & 1** : For every $X \in B$

$X . 0 = 0$

$X + 1 = 1$ (by **Duality)**

Boolean Function: A Function with logical variable, logical constant & logical operator representing a truth value (0 or 1) is known as Boolean Function.

For example :
- i) F = X + Y . Z
- ii) F(X,Y) = X′ + Y′ + Y. X
- iii) F(X,Y) = Σ (0,2)
  the function has same meaning as
  F(X,Y) = X′.Y′ + X.Y′
- iv) F(X,Y,Z) = ∏(1,2,4) the function means same as
  F(X,Y,z) = (X+Y+Z′).(X+Y′+Z).(X′+Y+Z)

Algebraic Verification of laws:

1. **Absorption Laws**

X+X.Y=X	X.(X+Y)=X
L.H.S. =X+X.Y	L.H.S. =X.(X+Y)
=X.1+X.Y	=X.X+X.Y
=X.(1+Y)	=X+X.Y
=X.1	=X.1+X.Y
=X	=X.(1+Y)
=R.H.S.	=X.1
	=X (R.H.S.)

X+X′.Y=X+Y	X.(X′+Y)=X.Y
L.H.S. =X+X′.Y	L.H.S. =X.(X′+Y)
=(X+X′).(X+Y)	=X.X′+X.Y
=1.(X+Y)	=0+X.Y
=X+Y (R.H.S.)	=X.Y (R.H.S.)

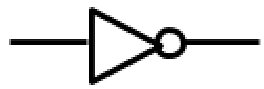
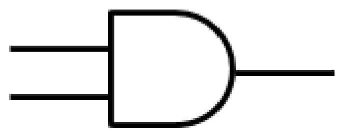
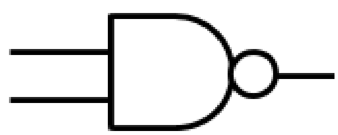
2. **De Morgan's Laws**

(X+Y)′=X′.Y′	(X.Y)′=X′+Y′
(X+Y).(X+Y)′=(X+Y).X′.Y′	(X.Y).(X.Y)′=(X.Y).(X′+Y′)
0 =X.X′.Y′+Y.X′.Y′	0 =X.Y.X′+X.Y.Y′
=0.Y′+Y.Y′.X′	=X.X′.Y+X.0
=0+0.X′	=0.Y+0
=0+0	=0+0
0 =0	0 =0

Truth Table: A truth table is a table containing all possible combinations of truth-values (false/true values) that can be assigned to variables present in an expression and the resultant truth-values of operations.

There will be 2^n combinations for **n** number of variables. Examples shown in Ref. Chart 1.0

Logic Gate: An electronic gadget which can perform some logical operation like AND/NOT /OR etc. is known as Logic Gate.

<table>
<tr><th colspan="3">Reference Chart 1.0</th></tr>
<tr><th>Operator</th><th>Truth Table</th><th>Logic Gate</th></tr>
<tr><td>NOT</td><td><table><tr><th>X</th><th>X′</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table></td><td></td></tr>
<tr><td>AND</td><td><table><tr><th>X</th><th>Y</th><th>X.Y</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table></td><td></td></tr>
<tr><td>OR</td><td><table><tr><th>X</th><th>Y</th><th>X+Y</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table></td><td></td></tr>
<tr><td>NAND</td><td><table><tr><th>X</th><th>Y</th><th>X.Y</th><th>(X.Y)′</th></tr><tr><td>0</td><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td><td>0</td></tr></table></td><td></td></tr>
<tr><td>NOR</td><td><table><tr><th>X</th><th>Y</th><th>X+Y</th><th>(X+Y)′</th></tr><tr><td>0</td><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td><td>0</td></tr></table></td><td></td></tr>
</table>

Verification of Laws using Truth Table

1. **Absorption Law:** X+X'.Y=X+Y

X	Y	X′	X′.Y	X+X′.Y	X+Y
0	0	1	0	0	0
0	1	1	1	1	1
1	0	0	0	1	1
1	1	0	0	1	1

2. **Distributive Law:** X+Y.Z=(X+Y).(X+Z)

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

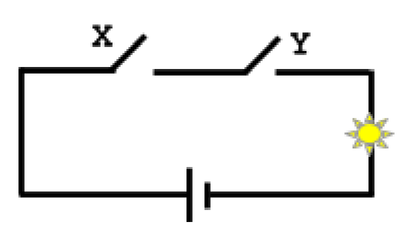
3. **Dominance of 1:** X+1=1

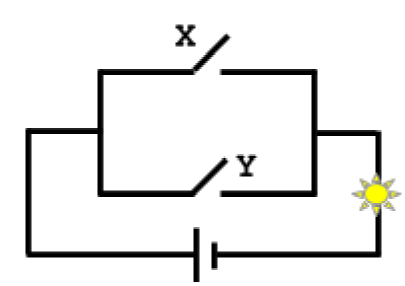
X	1	X+1
0	1	1
1	1	1

4. **Inverse Law:** X+X'=1

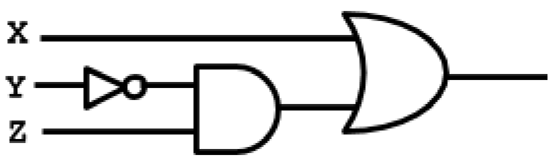
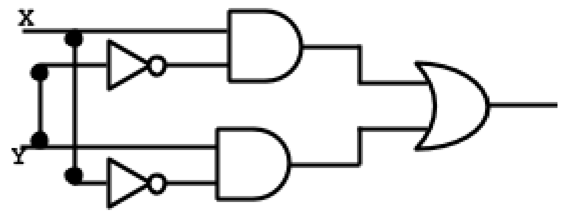
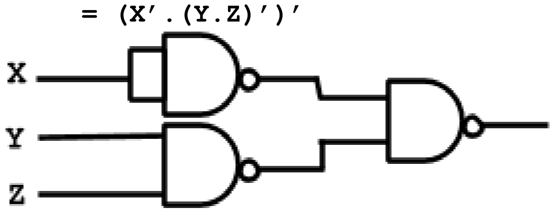
X	X′	X+X′	1
0	1	1	1
1	0	1	1

Equivalent Switching Circuits for **AND** & **OR**





Drawing Logic Circuits for Boolean Expressions

X+Y′.Z	**X.Y′+X′.Y**
	
X+Y.Z using NAND Gate only **X+Y.Z = ((X+Y.Z)′)′** **= (X′.(Y.Z)′)′** 	**X+Y.Z** using NOR Gate only **X+Y.Z = X+((Y.Z)′)′** **= X+(Y′+Z′)′** **= ((X+(Y′+Z′)′)′)′** 

Canonical and Standard Forms: Boolean expressions such as X or Y' or X' containing single variable or it's complement are called Literals. These literals when evaluated as Boolean expressions take the value 1 on half of the combinations of variables.

A **Minterm** in n variables X_1, X_2, …., X_n is defined as a meet (using AND operation) of n literals, where each literal involves a different variable from {X_1,X_2, …., X_n }; i.e., in a minterm, each variable must appear once, either complemented form or otherwise.

Example : For 2 variables
X′.Y′ , X′.Y , X.Y′ , X.Y are Minterms
For 3 variables
X′.Y.Z′ , X.Y′.Z′ , X.Y.Z etc. are the Minterms

A **Maxterm** in n variables is sum (join) of variables or its complement.

Example : For 2 variables
X+Y , X+Y' , X'+Y , X'+Y' etc. are Maxterms
For 3 variables
X+Y+Z , X+Y+Z' , X+Y'+Z etc. are Maxterms

A Boolean expression expressed in a product of maxterms (i.e. Product Of Sum **POS**) or sum of minterms (i.e. Sum Of Product **SOP**) form is known as a Canonical Form. A fully expanded POS /SOP form of an expression contains all the variables either in direct or complemented form.

Example: **POS** Form F (X,Y)= (X + Y) (X + Y′) (X′ + Y)
G (A,B,C)= (A′+B′+C′) (A′+B′+C)
SOP Form F (X,Y) = X′ Y′ + X Y + X′ Y
T (Y,W,Z)= Y′W′Z + Y W Z′ + Y W′Z + YWZ

Minimal Form: This is a form of a Boolean expression with minimum number of literals (expression cannot be reduced further).

Example : **X'+Y.Z ,A′.B + A.B′ , A+B+C.D′** are minimal forms of Boolean expressions.

Truth Table representation for Product Terms and Sum Terms

For 2 Variables

X	Y	Product Term (Min Term)	Sum Term (Max Term)
0	0	X′.Y′	X+Y
0	1	X′.Y	X+Y′
1	0	X.Y′	X′+Y
1	1	X.Y	X′+Y′

For 3 Variables

X	Y	Z	Product Term	Sum Term
0	0	0	X′.Y′.Z′	X+Y+Z
0	0	1	X′.Y′.Z	X+Y+Z′
0	1	0	X′.Y.Z′	X+Y′+Z
0	1	1	X′.Y.Z	X+Y′+Z′
1	0	0	X.Y′.Z′	X′+Y+Z
1	0	1	X.Y′.Z	X′+Y+Z′
1	1	0	X.Y.Z′	X′+Y′+Z
1	1	1	X.Y.Z	X′+Y′+Z′

For 4 Variables

X	Y	Z	W	Product Term	Sum Term
0	0	0	0	X′.Y′.Z′.W′	X+Y+Z+W
0	0	0	1	X′.Y′.Z′.W	X+Y+Z+W′
0	0	1	0	X′.Y′.Z.W′	X+Y+Z′+W
0	0	1	1	X′.Y′.Z.W	X+Y+Z′+W′
0	1	0	0	X′.Y.Z′.W′	X+Y′+Z+W
0	1	0	1	X′.Y.Z′.W	X+Y′+Z+W′
0	1	1	0	X′.Y.Z.W′	X+Y′+Z′+W
0	1	1	1	X′.Y.Z.W	X+Y′+Z′+W′
1	0	0	0	X.Y′.Z′.W′	X′+Y+Z+W
1	0	0	1	X.Y′.Z′.W	X′+Y+Z+W′
1	0	1	0	X.Y′.Z.W′	X′+Y+Z′+W
1	0	1	1	X.Y′.Z.W	X′+Y+Z′+W′
1	1	0	0	X.Y.Z′.W′	X′+Y′+Z+W
1	1	0	1	X.Y.Z′.W	X′+Y′+Z+W′
1	1	1	0	X.Y.Z.W′	X′+Y′+Z′+W
1	1	1	1	X.Y.Z.W	X′+Y′+Z′+W′

Note: To obtain **SOP** form from a Truth Table, Look for 1s in the truth table of given Function
Represent 0 as Complemented form of Variable
1 as Non-complemented form of Variable
To obtain **POS** form from a Truth Table, Look for 0 in the truth table of given Function

Represent 1 as Complemented form of a Variable
0 as Non-complemented form of Variable

Example: Obtain SOP and POS form for Boolean Function F, which is represented using Truth table below:

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

SOP Form: **F(X,Y,Z) = X'.Y'.Z' + X'.Y.Z + X.Y'.Z' + X.Y.Z' + X.Y.Z**
POS Form: **F(X,Y,Z) = (X+Y+Z').(X+Y'+Z).(X'+Y+Z)**

Karnaugh Maps: A Karnaugh Map is a graphical representation of a switching function (Boolean function). Once a switching circuit is represented by a K-Map, then this function can be reduced to the minimal form in a very easy manner. Moreover, this map ensures whether further simplification of the given switching function is possible or not. In brief, a K-Map provides a systematic mathematical method to reduce a switching function to the minimal form.

A Karnaugh Map consists of a set of squares also called cells. The number of cells in this map depends upon the number of variables used in the switching function. A Karnaugh map for a switching function with n variables consists of 2^n cells.

K-Map (2 variables)

	X′	X
Y′	0	2
Y	1	3

K-Map (3 variables)

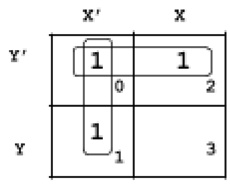
	X′Y′	X′Y	XY	XY′
Z′	0	2	6	4
Z	1	3	7	5

K-Map (4 variables)

	X′Y′	X′Y	XY	XY′
Z′W′	0	4	12	8
Z′W	1	5	13	9
ZW	3	7	15	11
ZW′	2	6	14	10

Reducing a 2 Var. Expression using K-Map

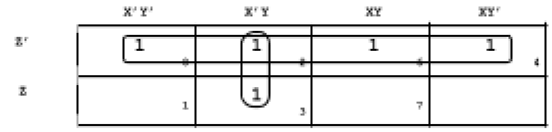
F(X,Y)=(0,1,2)



F(X,Y)=X′+Y′(Minimal Form)

Reducing a 3 Var Expression using K-Map

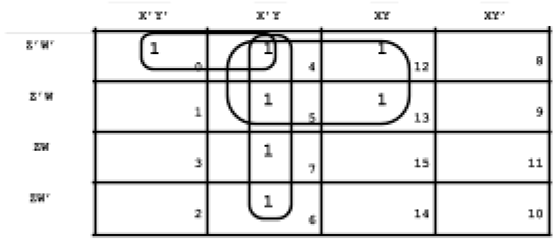
F(X,Y,Z)=(0,2,3,4,6)



F(X,Y,Z)=X′.Y + Z′ (Minimal Form)

Reducing a 4 Var Expression using K-Map

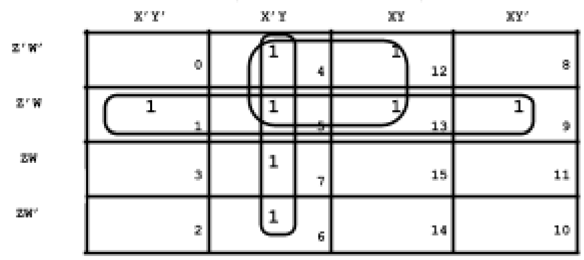
F(X,Y,Z,W)=(0,4,5,6,7,12,13)



F(X,Y,Z,W)=Y.Z′+ X′.Y+ X′.Z′W′

Reducing a 4 Var Expression using K-Map

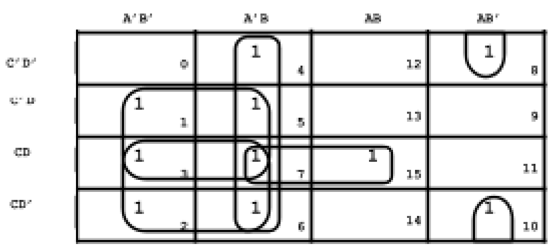
F(X,Y,Z,W)=(1,4,5,6,7,12,13)



F(X,Y,Z,W)=Y.Z′ + X′.Y + Z′W

Reducing a 4 Var Expression using K-Map

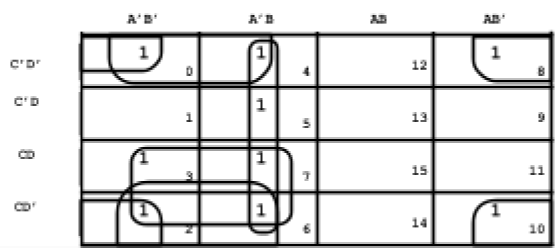
F(A,B,C,D)=S(1,2,3,4,5,6,7,8,10,15)



F(A,B,C,D)= A′B+A′C+A′D+BCD+AB′D′

Reducing a 4 Var Expression using K-Map

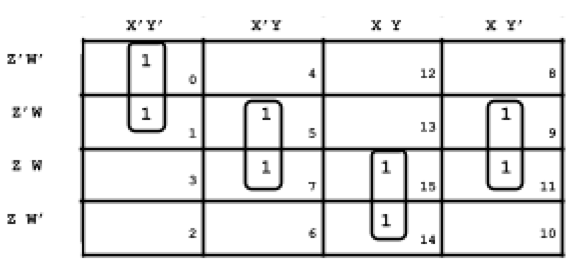
F(A,B,C,D)=S(0,2,3,4,5,6,7,8,10)



F(A,B,C,D)= A′B+A′C+B′D′

Reducing a 4 Var Expression using K-Map

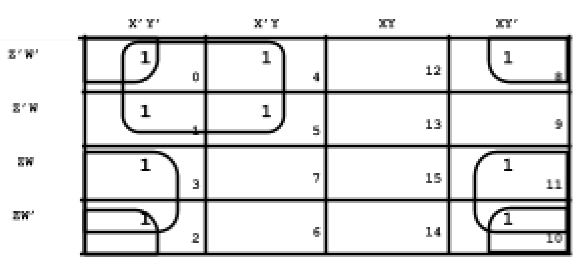
F(X,Y,Z,W)=S(0,1,5,7,9,11,14,15)



F(X,Y,Z,W)= X′Y′Z′+X′YW+XYZ+XY′W

Reducing a 4 Var Expression using K-Map

F(X,Y,Z,W)=S(0,1,2,3,4,5,8,10,11)



F(X,Y,Z,W)= X′Z′+Y′W′+Y′Z

Assignment based on CBSE pattern:

1. Define the following:
   a) Boolean variable b) Boolean expression c) Boolean statement
2. State the Associative property. Verify it using truth table.
3. State the principle of duality. Give suitable examples.
4. State the Inverse law. Verify it using a truth table.
5. State DeMorgan's law. Verify it using a truth table.
6. State Distributive law. Verify it using a truth table.
7. Verify the following laws algebraically:
   a) De Morgan's law b) Absorption law
8. Draw logic circuits for the following Boolean expressions:
   a) X+Y.X'+Z' b) X.Y.Z+X'.Y' c) (X+Y').Z' d) A+B'.B'.C
9. Draw circuits for the following expressions using NAND gate only:
   a) X'.Y+X.Z' b) (X+Y').Z
10. Draw circuits for the following expressions using NOR gate only:
    a) A.B'+B.C b) (A'+B').(A+C')
11. Prove the following algebraically:
    a) X'Y'+XY=(X+Y')(X'+Y) b) X+Y+Z=X+X'Y+X'Z
12. Obtain SOP form for Boolean Functions F and G from the given truth tables:
    a)

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

b)

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

13. Obtain POS form for Boolean Functions F, G and H from the given truth tables:

a)

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

b)

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

c)

A	B	C	H
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

14. Obtain POS form from the following expression:
F(X,Y,Z)= X'.Y'.Z+X.Y.Z'+X.Y'+X.Y.Z

15. Obtain SOP form from the following expression:
F(A,B,C)=(A'+B+C').(A+B'+C').(A'+B'+C').(A+B+C')

17. Obtain minimal forms using K-Maps:

a) **F(X,Y)** = $\sum$(0,1,2)
b) **F(X,Y)** = $\sum$(1,2,3)
c) **F(A,B)** = $\sum$(2,3)
d) **F(X,Y)** = $\sum$(0,2,3)
e) **F(X,Y,Z)** = $\sum$(0,1,3,5,6,7)
f) **F(U,V,W)** = $\sum$(3,5,6,7)
g) **F(X,Y,Z)** = $\sum$(2,3,4,5,6,7)
h) **F(A,B,C)** = $\prod$(1,6)
i) **F(X,Y,Z)** = $\prod$(3,6,7)
j) **F(A,B,C)** = $\sum$(0,2,3,4,5)
k) **F(X,Y,Z,W)** = $\sum$(0,1,2,3,6,9,11,13,15)
l) **F(P,Q,R,S)** = $\sum$(0,2,3,8,10,11,15)
m) **F(X,Y,Z,W)** = $\sum$(0,1,2,4,5,6,8,10)
n) **F(U,V,W,Z)** = $\sum$(0,5,7,8,10,11,13,15)
o) **F(P,Q,R,S)** = $\sum$(0,2,4,7,8,10,12,13,15)
p) **F(X,Y,Z,W)** = $\sum$(4,7,9,11,13,15)
q) **F(A,B,C,D)** = $\sum$(1,4,5,7,9,13,14,15)
r) **F(X,Y,Z,W)** = $\prod$(2,3)
s) **F(X,Y,Z,W)** = $\prod$(0,1,2,3,6,14)
t) **F(A,B,C,D)** = $\prod$(4,6,7)