6.13 Boolean Algebra

1. **Boolean algebra :** A non empty set with two binary operations '+' and '·', an unary operation '′' and two distinct elements 0 and 1 is called Boolean algebra, denoted by (B, +, ·, ′, 0, 1) iff it satisfies the following properties. Let a, b $\in$ B then

(i) Commutative laws ;

$$a + b = b + a \text{ and } a \cdot b = b \cdot a$$

(ii) Distributive laws :

$a + (b \cdot c) = (a + b) \cdot (a + c)$

and $a \cdot (b + c) = a \cdot b + a \cdot c$

(iii) Identity laws :

$a + 0 = a$ and $a \cdot 1 = a$

(iv) Complement law :

$a + a' = 1$ and $a \cdot a' = 0$

2. **Basic results in Boolean algebra :** Let $a, b, c \in B$ then

   (i) *Idempotent laws :* $a + a = a$ and $a \cdot a = a$

   (ii) *Boundedness laws :* $a + 1 = 1$ and $a \cdot 0 = 0$

   (iii) *Absorption laws :*

   $a + (a \cdot b) = a$ and $a \cdot (a + b) = a$

   (iv) *Associative laws :* $a + (b + c) = (a + b) + c$ and $(a \cdot b) \cdot c = a \cdot (b \cdot c)$

   (v) *Uniqueness and complements :* $a + x = 1$ and $a \cdot x = 0 \Rightarrow x = a'$

   (vi) *Involution laws :* $(a')' = a$, $0' = 1$ and $1' = 0$

   (vii) *Demorgan's laws :* $(a + b)' = a' \cdot b'$ and $(a \cdot b)' = a' + b'$

3. **Complement of a function :** The complement of a function F is $\overline{F}$. We can obtain $\overline{F}$ by interchanging 1's for 0's and 0's for 1's.

   Demorgan's theorem :

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

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

4. **Simplification of Boolean function :**

   (i) *Sum of Products (SOP) :* A Boolean expression E is said to be in a sum of products form if E is a sum of product of variables.

   e.g. $Y = ABC + B\overline{C}D + \overline{A}\,\overline{B}\,\overline{C}$

   (ii) *Product of Sums (POS) :* A Boolean expression E is said to be in a product of sums form if E is a product of sum of terms.

   e.g. $Y = (A+B)+(\overline{A}+B)$

   (iii) *Minterm :* A minterm of n variables is a product of n literals in which each variable appears exactly once in either true or complemented form but not both.

   Minterms for two variables are xy, xy', x'y, x'y'.

   Minterms for 3 variables are xyz, x'yz, xy'z, xyz', x'y'z, x'yz', xy'z', x'y'z'.

   There are 2^n minterms for n variables.

   (iv) *Maxterm :* Maxterm of n variabels is the sum of n literals in which each variable appears exactly once in either true or complemented form but not both.

   Maxterms for two variables are (x + y), (x' + y), (x + y'), (x' + y')

   Maxterms for 3 variables are (x + y + z), (x' + y + z), (x + y' + z), (x + y + z'), (x' + y' + z), (x' + y + z'), (x + y' + z'), (x' + y' + z').

   For n variables there will be 2^n maxterms.

   (v) When a Boolean expression is written in sum of minterms form, it is referred as minterm expansion or disjunctive normal form. It is also called as canonical sum of products.

   (vi) When a Boolean expression is written in product of maxterms form, it is referred as maxterm expansion or conjunction normal form or Canonical product of sums.

Ex.6.13.1 : Prove that : (i) $a + a = a$; (ii) $a + 1 = 1$; (iii) $a + (a \cdot b) = a$

Sol. :

(i) Consider, $a = a + 0$

$a = a + aa'$ ...(By Identity law)

$= (a + a) \cdot (a + a')$ ...(By Complement law)

$= (a + a) \cdot 1$

$a = a + a$

(ii) $a + 1 = a + a + a'$ ...(By Complement law)

$= (a + a) + a'$

$= a + a'$ ($\because$ idempotent)

$= a$

(iii) $a + (a \cdot b) = a \cdot 1 + a \cdot b$

$$= a \cdot (1 + b)$$

$$= a \cdot 1$$

$$= a$$

Ex.6.13.2 : Prove that $(A + B)(A + C) = A + BC$

Sol. : L.H.S. $= (A + B)(A + C)$

$$= AA + AC + BA + BC$$

$$= A + AC + BA + BC \quad ...(\because AA = 1)$$

$$= A(1 + C) + BA + BC$$

$$= A + AB + BC \quad ...(\because 1 + C = 1)$$

$$= A(1 + B) + BC$$

$$= A + BC \quad ...(\because 1 + B = 1)$$

$$= \text{R.H.S.}$$

Ex.6.13.3 : Prove that $(A+\overline{B}+AB)(A+\overline{B})(\overline{A}B) = 0$

Sol. :

L.H.S. $= (A+\overline{B}+AB)(A+\overline{B})(\overline{A}B)$

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

$$(\because A + AB = A)$$

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

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

$$(\because A + A = A)$$

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

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

$$= (A+\overline{B})(\overline{A}B) \quad (\because 1+\overline{B} = 1)$$

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

$$= 0+0 \quad (\because A\overline{A} = 0 \text{ and } B\overline{B} = 0)$$

$$= 0$$

Ex.6.13.4 : Simplify the following expression :

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

Sol. :

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

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

...(De Morgan's first theorem)

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

But $\overline{A}+A = \overline{A}$

$$\therefore \quad y = \overline{(\overline{A}+\overline{B}+AB)}$$

Now use De-Morgan's second theorem which states that,

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

$$\therefore \quad y = \overline{\overline{A}}\cdot\overline{\overline{B}}\cdot\overline{AB}$$

But $\overline{\overline{A}} = A$ and $\overline{\overline{B}} = B$

$$y = A\cdot B\cdot\overline{AB}$$

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

...(De-Morgan's second theorem)

$$\therefore \quad y = A\cdot B(\overline{A}+\overline{B}) = A\overline{A}B+AB\overline{B}$$

But $A\overline{A} = 0$ and $B\overline{B} = 0$

$$\therefore \quad y = 0\cdot B + A\cdot 0 = 0+0 = 0$$

$$\therefore \quad y = 0$$

Ex.6.13.5 : For the given function, $F = x\overline{y}+x\overline{y}$, find the complement of 'F'.

Sol. : $F = x\overline{y}+x\overline{y}$

$$F = x\overline{y} \quad ...(A + A = A)$$

Take the complement of both sides,

$$\therefore \quad \overline{F} = \overline{x\overline{y}}$$

Using De morgan's first law, we get,

$$\overline{F} = \overline{x}+\overline{\overline{y}} \quad ...(\text{as } \overline{A\cdot B} = \overline{A}+\overline{B}\ \overline{\overline{y}} = y)$$

$$\overline{F} = \overline{x}+y$$

Ex.6.13.6 : Simplify :

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

Sol. :

L.H.S. $= \overline{A}\,\overline{B}\,\overline{C}\,\overline{D}+\overline{A}\,\overline{B}\,\overline{C}D+\overline{A}\,\overline{B}CD+\overline{A}\,\overline{B}C\overline{D}$

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

But $\overline{D}+D = 1$

L.H.S. $= \overline{A}\,\overline{B}\,\overline{C}+\overline{A}\,\overline{B}C = \overline{A}\,\overline{B}(\overline{C}+C)$

But $\quad \overline{C}+C = 1$

$$\text{L.H.S.} = \overline{A}\,\overline{B} = \text{R.H.S.}$$

Ex.6.13.7 : Find the complement of the following functions :

$F_1 = \overline{A}\,\overline{B}\,\overline{C}+\overline{A}\,\overline{B}\,C$ and $F_2 = A(\overline{B}\,\overline{C}+BC)$

Sol. : (i) $\quad F_1 = \overline{A}B\overline{C}+\overline{A}\,\overline{B}C$

$\therefore \quad \overline{F_1} = \overline{(\overline{A}B\overline{C})+(\overline{A}\,\overline{B}C)}$

$$= (\overline{\overline{A}B\overline{C}})\cdot(\overline{\overline{A}\,\overline{B}C})$$

$\therefore \quad \overline{F_1} = (A+\overline{B}+C)\cdot(A+B+\overline{C})$

(ii) $\quad F_2 = A(\overline{B}\,\overline{C}+BC)$

$$\overline{F_2} = [\overline{A(\overline{B}\,\overline{C}+BC)}] = [\overline{\overline{A}+(\overline{B}\,\overline{C}+BC)}]$$

$$= \overline{A}(\overline{\overline{B}\,\overline{C}})\cdot(\overline{BC})$$

$\therefore \quad \overline{F_2} = \overline{A} + (B+C)(\overline{B}+\overline{C})$

Ex.6.13.8 : For the logic circuit shown in Fig. 6.13.1 Write the Boolean expression and simplify it.

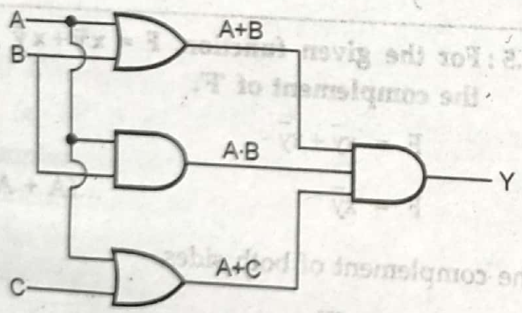


Fig. 6.13.1

Sol. :

Step 1 : Write the Boolean expression.

The expression for output for the given logic circuit is

$$Y = (A+B)\,(AB)\,(A+C)$$

Step 2 : Bring this expression in SOP form.

Multiply the terms to get the expression into SOP form.

$$Y = (A+B)\;(AAB+ABC)$$

$$Y = (A+B)\;(AB+ABC) \quad \ldots (AA = A)$$

$$= AAB + AABC + BAB + BABC$$

$$= AB + ABC + AB + ABC$$

$$= AB + AB + ABC + ABC$$

But $\quad AB + AB = AB$ and $ABC + ABC = ABC$

$\therefore \quad Y = AB + ABC = AB\,(1+C)$

$$Y = AB \quad \ldots (1 + C = 1)$$

This is simplified expression.

Ex.6.13.9 : Convert the following expression into their standard SOP or POS forms.

(a) $Y = AB + AC + BC$

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

(c) $Y = A + BC + ABC$

(d) $(x+y)\,(x'+y')$

Sol. : (a) $\quad Y = AB + AC + BC$

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

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

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

$ABC + ACB + BCA = ABC \quad \ldots(\text{As } A + A = A)$

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

This is the required expression in standard SOP form.

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

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

But $\quad A + BC = (A+B)\,(A+C)$

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

This is in the standard POS form.

(c) $\quad Y = A + BC + ABC$

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

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

$$= (ABC+BCA+ABC)+AB\overline{C}+A\overline{B}C+A\overline{B}\,\overline{C}+BC\overline{A}$$

as $\quad A + A = A$

then $\quad (ABC + BCA + ABC) = ABC$

$\therefore \quad Y = ABC+AB\overline{C}+A\overline{B}C+A\overline{B}\,\overline{C}+BC\overline{A}$

This expression is in the standard SOP form.

d) We have $(x+y)\,(x'+y')$

$$= xx' + xy' + yx' + yy'$$

which is the dnf.

Ex.6.13.10 : Explain how to write maxterms and minterms from truth table for three variables.

Sol. : Let A, B, C be three variables and Y be its output. The concept of maxterms and minterms allow us to introduce a very convenient shorthand notation to express logic functions.

Consider the following table

Variables	Minterms	Maxterms
A B C	m_i	M_i
0 0 0	$m_0 = \overline{ABC}$	$M_0 = A + B + C$
0 0 1	$m_1 = \overline{AB}C$	$M_1 = A + B + \overline{C}$
0 1 0	$m_2 = \overline{A}B\overline{C}$	$M_2 = A + \overline{B} + C$
0 1 1	$m_3 = \overline{A}BC$	$M_3 = A + \overline{B} + \overline{C}$
1 0 0	$m_4 = A\overline{BC}$	$M_4 = \overline{A} + B + C$
1 0 1	$m_5 = A\overline{B}C$	$M_5 = \overline{A} + B + \overline{C}$
1 1 0	$m_6 = AB\overline{C}$	$M_6 = \overline{A} + \overline{B} + C$
1 1 1	$m_7 = ABC$	$M_7 = \overline{A} + \overline{B} + \overline{C}$

Ex.6.13.11 : For the following truth table of 3 variables A, B, C. Write the logic expression in the standard SOP form and POS form.

Sol. : Given truth table is

A	B	C	Y		Product terms
0	0	0	0	→	$M_0 = A + B + C$
0	0	1	1	←	$\overline{A}\,\overline{B}\,C(m_1)$
0	1	0	0	→	$M_2 = A + \overline{B} + C$
0	1	1	0	→	$M_3 = A + \overline{B} + \overline{C}$
1	0	0	1	←	$A\,\overline{B}\,\overline{C}(m_4)$
1	0	1	0		$M_5 = \overline{A} + B + \overline{C}$
1	1	0	0		$M_6 = \overline{A} + \overline{B} + C$
1	1	1	1	←	$ABC(m_7)$

Consider the product terms for which output Y = 1

OR (add) all the product terms

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

which is the requierd logic expression in standard SOP form.

This expression can also be written as,

$$Y = m_1 + m_4 + m_7 = \sum m(1,4,7)$$

Consider the maxterms for which Y = 0

$$\therefore \quad M_0 = A + B + C,\ M_2 = A + \overline{B} + C,$$

$$M_3 = A + \overline{B} + \overline{C}$$

$$M_5 = \overline{A} + B + \overline{C} \text{ and } M_6 = \overline{A} + \overline{B} + C$$

Therefore, the standard POS form is

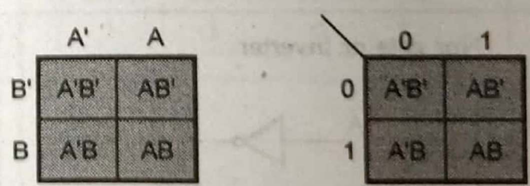
$$Y = (A + B + C)\cdot(A + \overline{B} + C)\cdot(A + \overline{B} + \overline{C})\cdot(\overline{A} + B + \overline{C})\cdot(\overline{A} + \overline{B} + C)$$

6.14 Karnaugh Map (K-map)

AKTU : 2010-11, 2011-12, 2016-17

Karnaught map method is a graphical technique which provides a simple straight forward procedure for simplification of Boolean expression of two, three or four variables. It can also be extended for five, six or more variables.

1. **Two variable Karnaugh maps :** The number of variables are 2 so the map will have $2^2 = 4$ square. In this case four possible minterms with two variables A and B i.e. AB, AB', A'B , A'B' are represented by four squares in the map labelled below.



The expression can be simplified by properly combining those squares in the K-map which contain 1s. The process for combining 1s is called looping.

2. **Three variable K-map :** The number of variables are 3 so map will have $2^3 = 8$ squaers. The eight possible minterms are labelled as shown in Fig. 6.14.1

Given a minterm expansion of a function, it can be plotted on a map by placing 1s in the square which corresponds to minterms present in the expression and 0s in the remaining squares.

	A'B'	A'B	AB	AB'
C'	A'B'C'	A'BC'	ABC'	AB'C'
C	A'B'C	A'BC	ABC	AB'C

C \ AB	00	01	11	10
0	A'B'C'	A'BC'	ABC'	AB'C'
1	A'B'C	A'BC	ABC	AB'C

Fig. 6.14.1

3. **Four variable Karnaught map :** The number of variables are 4. Hence map will be $2^4 = 16$ squares. Fig. 6.14.2 shows the K-map for four variables A, B, C and D and alternative way of representing four variables.

A'B'C'D'	A'BC'D'	ABC'D'	AB'C'D'
A'B'C'D	A'BC'D	ABC'D	AB'C'D
A'B'CD	A'BCD	ABCD	AB'CD
A'B'CD'	A'BCD'	ABCD'	AB'CD'

CD \ AB	00	01	11	10
	A'B'C'D'	A'BC'D'	ABC'D'	AB'C'D'
	A'B'C'D	A'BC'D	ABC'D	AB'C'D
	A'B'CD	A'BCD	ABCD	AB'CD
	A'B'CD'	A'BCD'	ABCD'	AB'CD'

Fig. 6.14.2

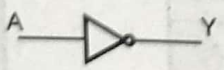
To simplify a sum of product expression in four variable one has to identify groups of minterms of squares 2, 4, 8 of 16 containing 1s that can be combined.

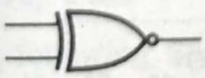
Logic gates :

Logic gates are the devices used as basic building blocks of all the digital circuits. The Boolean algebra developed by Charles Boole way back in 1884 is used for representing, simplifying and analysing the logic circuits. The basic logic gates are NOT, AND and OR along with NOR, NAND, Ex-OR etc.

The relation between the inputs and the outputs of a gate can be expressed mathematically by means of the Boolean expression. In order to understand Boolean algebra, we need to use the gates. So the symbols and Boolean expressions should be known to us which is given as follows :

Various logic gates :

Sr. No	Name of gate	Boolean expression	Truth table	Logical operation
1.	Nor gate or inverter 	$Y = \overline{A}$	A / Y: 0 / 1; 1 / 0	Inversion
2.	AND gate 	$Y = AB$	A / B / Y: 0 / 0 / 0; 0 / 1 / 0; 1 / 0 / 0; 1 / 1 / 1	Logical multiplication

3.	OR gate 	$Y = A + B$	A B Y 0 0 0 0 1 1 1 0 1 1 1 1	Logical additon
4.	NAND gate 	$Y = \overline{AB}$	A B Y 0 0 1 0 1 1 1 0 1 1 1 0	NOT AND
5.	NOR gate 	$Y = \overline{A+B}$	A B Y 0 0 1 0 1 0 1 0 0 1 1 0	NOT OR
6.	Exclusive OR 	$Y = A \oplus B$	A B Y 0 0 0 0 1 1 1 0 1 1 1 0	Addition/ subtraction
7.	Exclusive NOR X-NOR 	$Y = \overline{A \oplus B}$	A B Y 0 0 1 0 1 0 1 0 0 0 1 1	NOT EX-OR

Ex.6.14.1 : Find the K-map and simplify the expression for AB' + A'B'

Sol. : Two adjacent square A'B' and AB' containing 1 have been grouped together. They have been circled. These two terms can be looped that eliminates the A variable since it appears both in complemented and uncomplemented forms.

B \ A	0	1
0	1	1
1	0	0

This can be verified algebraically as follows :

$$AB' + A'B' = (A + A')\, B' = 1 \cdot B' = B'$$

Ex.6.14.2 : Use the K-map to simplify the following :
(i) X = ABC' + ABC ;
(ii) X = A'B'C' + AB'C'

Sol. : (i) The Boolean function

X = ABC' + ABC is shown in Fig. 6.14.3 in the K-map as follows :

C \ AB	00	01	11	10
0	0	0	1	0
1	1	0	1	0

Fig. 6.14.3

The adjacent square representing ABC' and ABC are grouped together. This eliminates the C variable since it appears in both uncomplemented and complemented form. The simplified function will be X = AB.

(ii) The Boolean function X = A'B'C' + AB'C' is shown in Fig. 6.14.4 in the K-map as

	00	01	11	10
0	1	0	0	0
1	0	0	0	0

Fig. 6.14.4

Thus the two 1s in this map can be looped to provide a simplified result X = B'C'.

Ex.6.14.3 : Use the K-map to simplify the following
(i) X = A'B'C' + A'BC' + ABC' + AB'C
(ii) X = A'B'C' + A'B'C + A'BC + A'BC' + AB'C + ABC

Sol. : (i) The Boolean function

X = A'B'C' + A'BC' + ABC' + AB'C and the K-map of three variable is as follows :

C \ AB	00	01	11	10
0	1	1	1	0
1	0	0	0	1

The group of first two horizontal 1 square gives A'C' and the group second and third horizontal 1 square gives BC'.

Hence, the simplfied result is

$$X = A'C' + BC' + AB'C$$

(ii) The K-map of given function

The quad formed by A'B'C', A'BC', A'B'C and A'BC produces the resultant as A' and the quad formed by A'B'C, A'BC, ABC and AB'C produces the resultant as C.

	00	01	11	10
0	1	1	0	0
1	1	1	1	1

Hence final result is ,

$$X = A' + C$$

Ex.6.14.4 : Use the K-map to simplify
X = A'B'C'D' + AB'C'D' + A'B'CD' + AB'CD'

Sol. : K-map of the Boolean expression is

AB \ CD	00	01	11	10
00	1			1
01				
11				
10	1			1

The variable B' and D' remain unchanged as A and C are in complemented and uncomplemented form.

The final result is $X = B'D'$

Ex.6.14.5 : Find the logical networks corresponding to Boolean experssion

(i) $AB + C'$; (ii) $A'B'C + A'BC + AB'$

Sol. : (i) Implementation with AND, OR and NOT gate

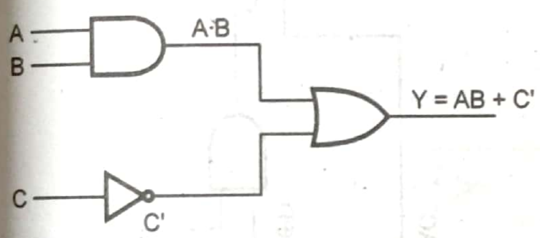


Fig. 6.14.5

(ii)

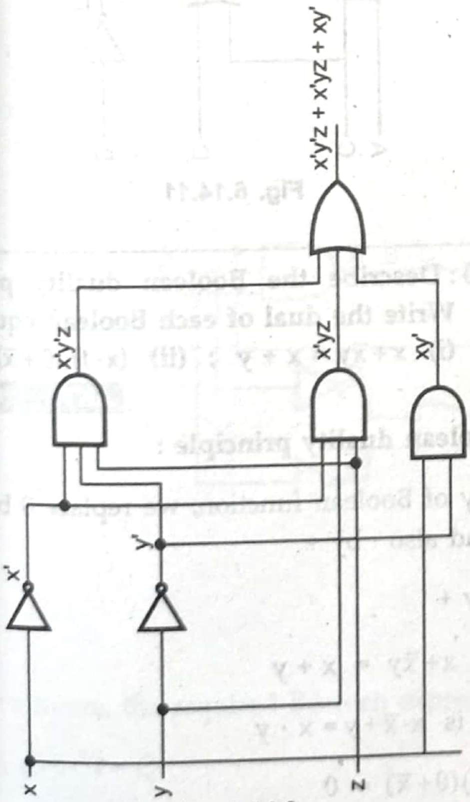


Fig. 6.14.6

Ex.6.14.6 : Realize the following Boolean expersion using only NAND gates.

$Y = (AB + BC)\, C$

Sol. :

$$Y = (AB + BC)\, C$$

$$= ABC + BCC = ABC + BC$$

$$\ldots(\text{as } C \cdot C = C)$$

$$= BC\,(A + 1)$$

$$Y = BC \qquad \ldots(\because A + 1 = 1)$$

$$Y = \overline{\overline{BC}} \qquad \ldots(\text{As } \overline{\overline{A}} = A)$$

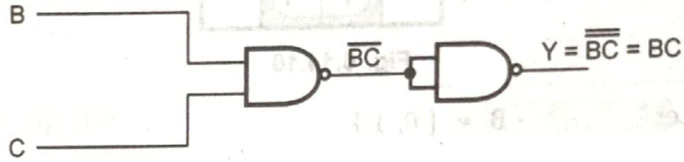


Fig. 6.14.7

Ex.6.14.7 : Realise the logic equation using

(i) OR and AND gates ;

(ii) Only NOR gates

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

Sol. :

Step 1 : Realisation using OR and AND gates.

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

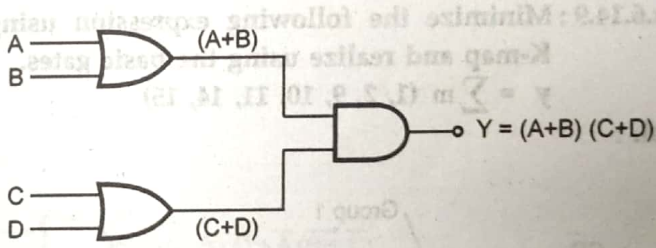


Fig. 6.14.8

Step 2 : Realisation using only NOR gates.

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

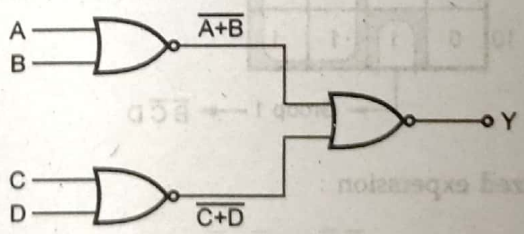


Fig. 6.14.9

Ex.6.14.8 : Define a Boolean function of degree n. Simplify the following Boolean expression using K-maps.

$xyz + xy'z + x'y'z + x'yz + x'y'z'$

AKTU : 2011-12

Sol. : Boolean function of degree n :

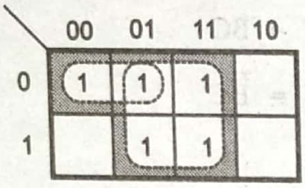


Fig. 6.14.10

Let $\quad B = \{0, 1\}$

Then $B^n = \{(x_1, x_2, x_3 \cdots\cdots x_n) x_i \; B \text{ for } 1 \le i \le n\}$

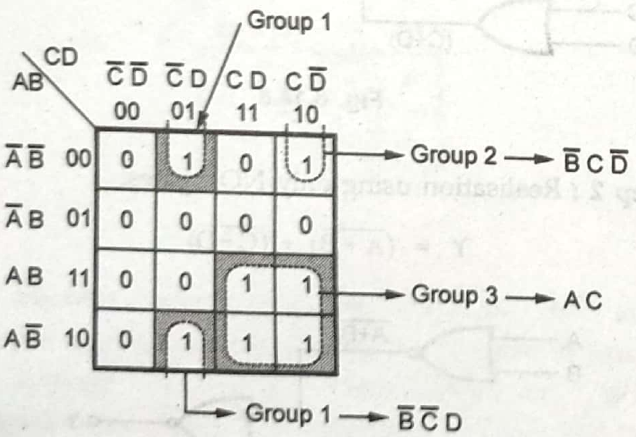
is the set of all possible n-tuples of 0's and 1's. The variable x is called a Boolean variable. B is called Boolean function of degree n.

The K-map for given Boolean function is shown in Fig. 6.14.10

Then the simplified expression is z + x'y'

Ex.6.14.9 : Minimize the following expression using K-map and realize using the basic gates.
$$y = \sum m\,(1, 2, 9, 10, 11, 14, 15)$$

Sol. :



Minimized experssion :

$$y = \bar{B}\bar{C}D + \bar{B}C\bar{D} + AC$$

$$= \bar{B}(\underbrace{\bar{C}D \oplus C\bar{D}}_{\text{EX-OR gate}}) + AC$$

$$= \bar{B}(C \oplus D) + AC$$

Realisation with minimum number of gates :

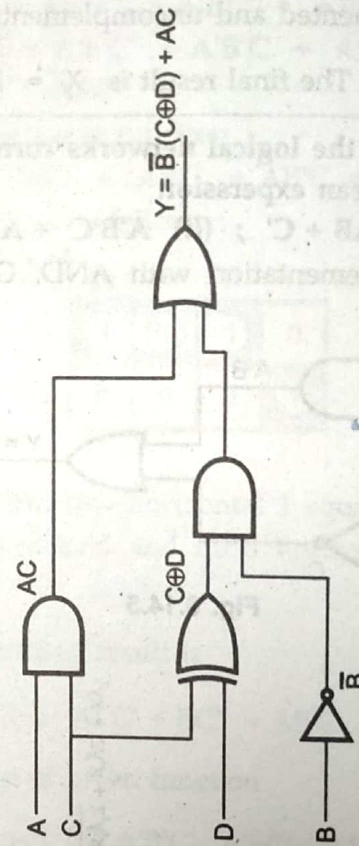


Fig. 6.14.11

Ex.6.14.10 : Describe the Boolean duality principle. Write the dual of each Boolean equations.
(i) $x + \bar{x}y = x + y$; (ii) $(x \cdot 1)(0 + \bar{x}) = 0$

AKTU : 2010-11

Sol. : Boolean duality principle :

In duality of Boolean function, we replace 0 by 1 and 1 by 0 and also · by +

· and · by +

(i) $\quad x + \bar{x}y = x + y$

It's dual is $x \cdot \bar{x} + y = x \cdot y$

(ii) $\quad (x \cdot 1)(0 + \bar{x}) = 0$

It's dual is $\quad (x + 0)(1 \cdot \bar{x}) = 1$

Ex.6.14.11 : Simplify the following Boolean function using K-map f (x, y, z) = S (0, 2, 3, 7)

AKTU : 2010-11

Sol. : The given Boolean function can be represented by K-map as shown in Fig. 6.14.12 on simplification.

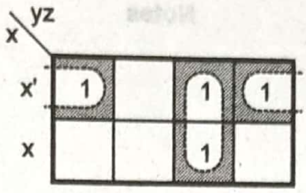


Fig. 6.14.12

The simplified function is $f = yz + x'z'$

Ex.6.14.12 : Find the Boolean algebra expression for the following system **AKTU : 2016-17**

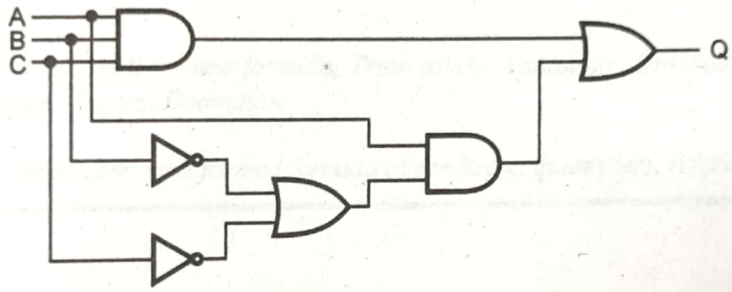


Fig. 6.14.13

Sol. : Given that

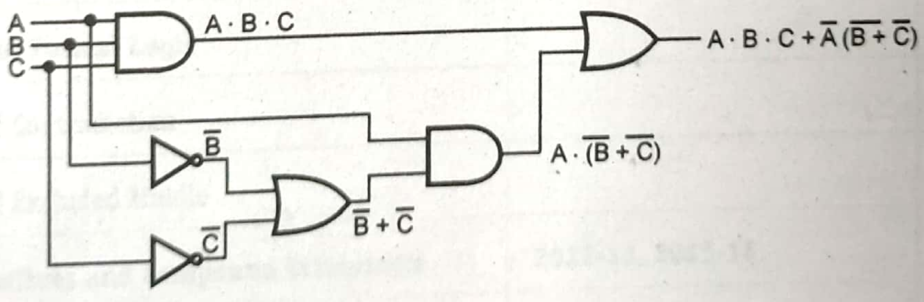


Fig. 6.14.14

From above figure, the required Boolean expression is

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

❑❑❑