Representation of a Boolean function:

1) Every Boolean function can be expressed by an algebraic expression, such as $x'y+z$, xyz' etc.

ie $F(x,y,z) = x'y+z$

$F(x,y,z) = xyz'$

$F(x,y,z,t) = x.[y+(z.t')]$ etc.

2) Another method of representation of Boolean function is in terms of a truth table, which list exhaustively all possible combination of the input variables and which for each such input configuration, record a functional value.

for example

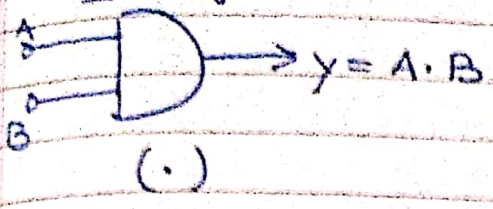
$F(x,y,z) = x'y+z$

x	y	z	x'	x'y	x'y+z
0	0	0	1	0	0
0	0	1	1	0	1
0	1	0	1	1	1
0	1	1	1	1	1
1	0	0	0	0	0
1	0	1	0	0	1
1	1	0	0	0	0
1	1	1	0	0	1

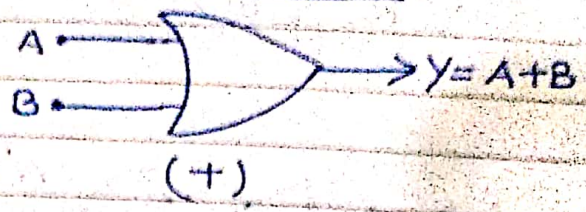
3) The other method for representing Boolean function is Circuit diagram, composed of AND, OR, NOT GATES

(BY Anita Sharma)

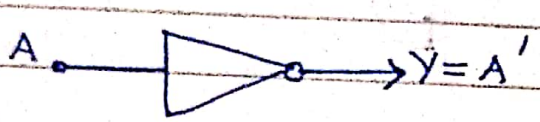
AND gate



OR GATE



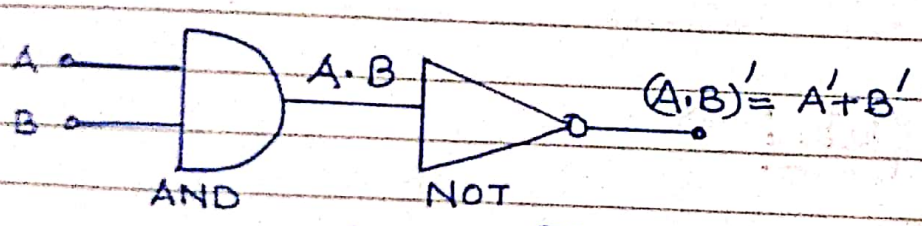
NOT gate

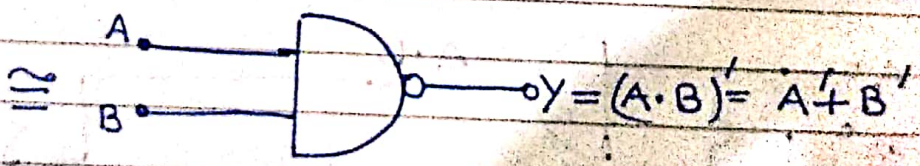


(By Anita Sharma)

Other standard gates are NAND, NOR, XOR, XNOR GATE

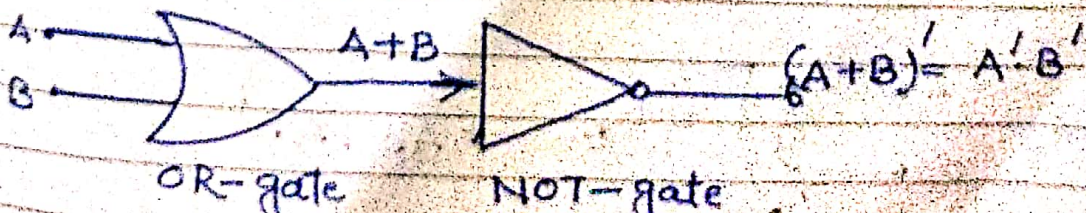
NAND gate :→

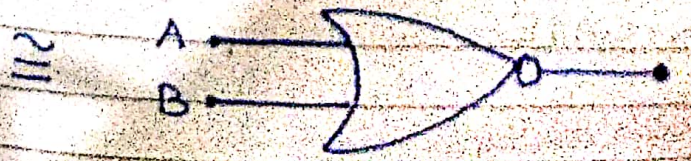




NAND GATE

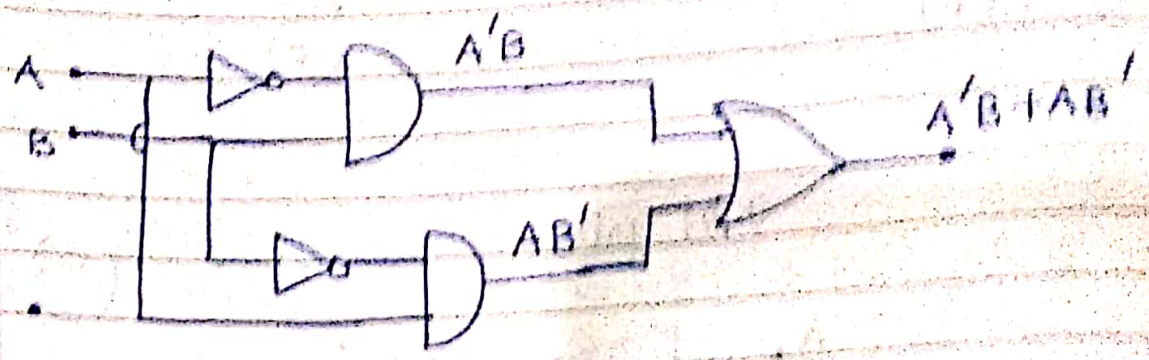
NOR gate :→

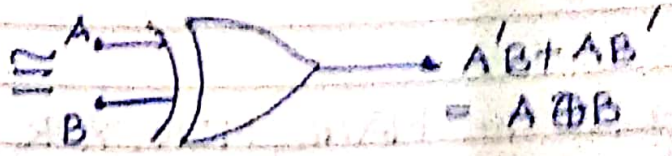




NoR-gate

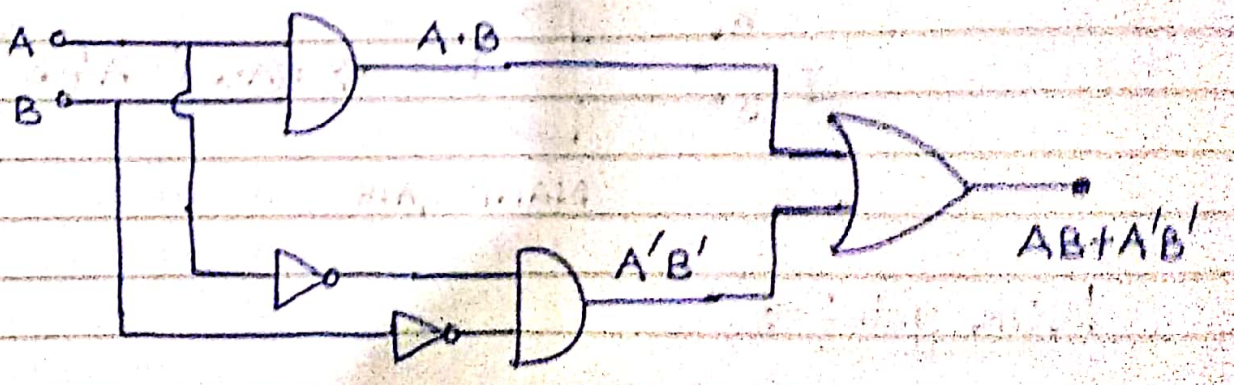
XOR gate :→

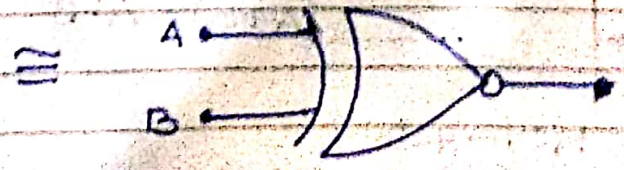




XOR ⊕

XNOR- gate :→

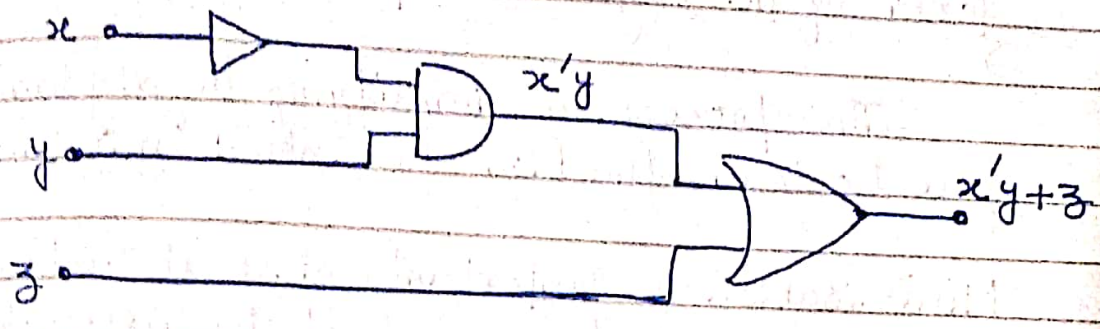




XNOR-GATE.

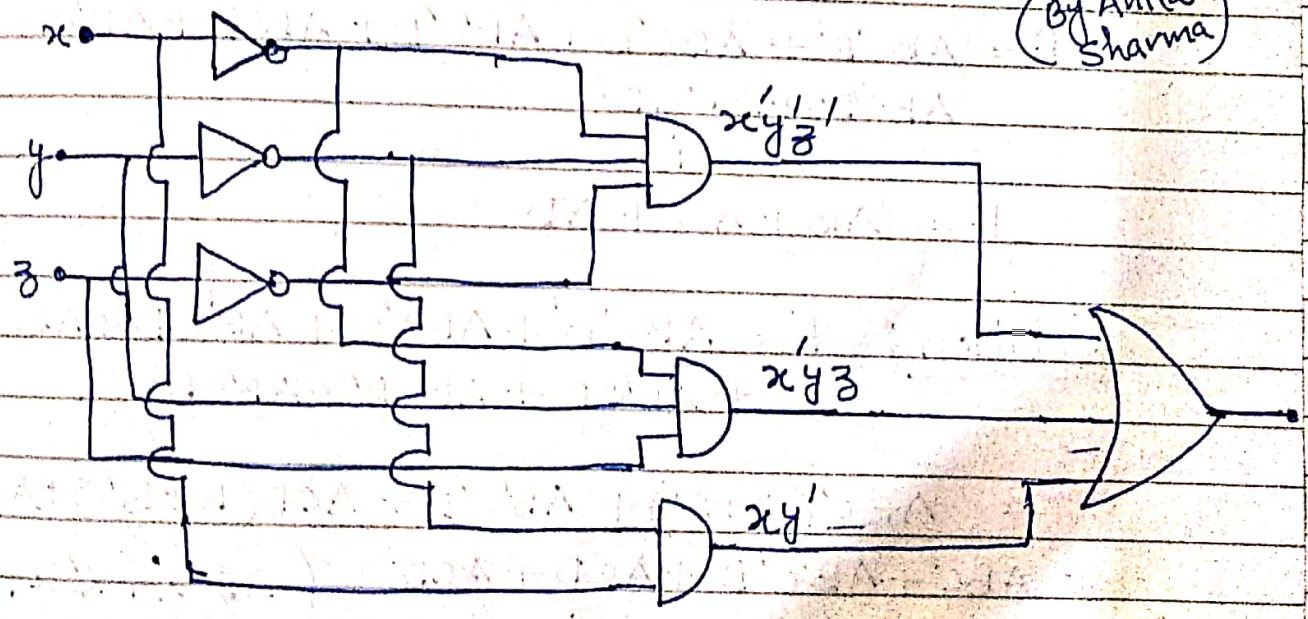
So, a Boolean function can be transformed from an algebraic expression into a circuit diagram composed of logic gates connected in a particular structure.

For example: $F(x,y,z) = x'y + z$, the circuit diagram for this function is as under



(ii) Circuit diagram for the function $F_1(x,y,z) = x'y'z' + x'yz + xy'$ is as follows:

(By Anita Sharma)



$F_2(x,y,z) = x'y'z' + x'yz + xy'$

Minimisation of Boolean Functions:

As discussed earlier, every boolean function can be expressed as a product of maxterms or a sum of minterms. These expression contains a number of literals (variables). When these expressions are directly represented through a circuit, using logic gates, the implementation becomes very complex. So, it is preferable to have the most simplified form of the algebraic expression

The process of simplifying the algebraic expression of a boolean function is called 'minimisation'.

Minimisation is important since it reduces the complexity and hence cost of the associated circuit, which can be done by using boolean algebra or K-map.

For example: The minimised form of boolean function

$F = ABC'D' + ABC'D + AB'C'D + ABCD + AB'CD + ABCD' + AB'CD'$

is

$F = AB + AC + AD$

Solution: $F = ABC'D' + ABC'D + AB'C'D' + ABCD + AB'CD + ABCD' + AB'CD'$

$= ABC'(D+D') + AB'C'D + ACD(B+B') + ACD'(B+B')$

$= ABC' + AB'C'D + ACD + ACD'$ (As $X + X' = 1$ & X)

$= ABC' + AB'C'D + AC(D+D')$

$= ABC' + AB'C'D + AC$

$= AC'(B + B'D) + AC$ $\quad \{ \because X + X'y = X + y \}$

$= AC'(B + D) + AC$

$= AC'B + AC'D + AC$

$= AC'B + A(C'D + C)$

$= AC'B + A(D + C)$

$= AC'B + AD + AC$

$= AD + A(C'B + C)$

$= AD + A(B + C)$

$= AD + AB + AC$

or $AB + AC + AD$

By using K-map

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

G_1, G_2, G_3

As the above expression is in sum of product form so make group of ones in (1, 2, 4 or 8 ones)

The above three groups G_1, G_2 & G_3 will give the three prime implicants (All groups are of four 1's)

The group G_1, spans the whole third row and gives AB

(by Anita Sharma)

The group G_2 of four 1's will give $\bar{A}D$

The group G_3 of four ones will give $\bar{A}G$

So, the minimised boolean expression is

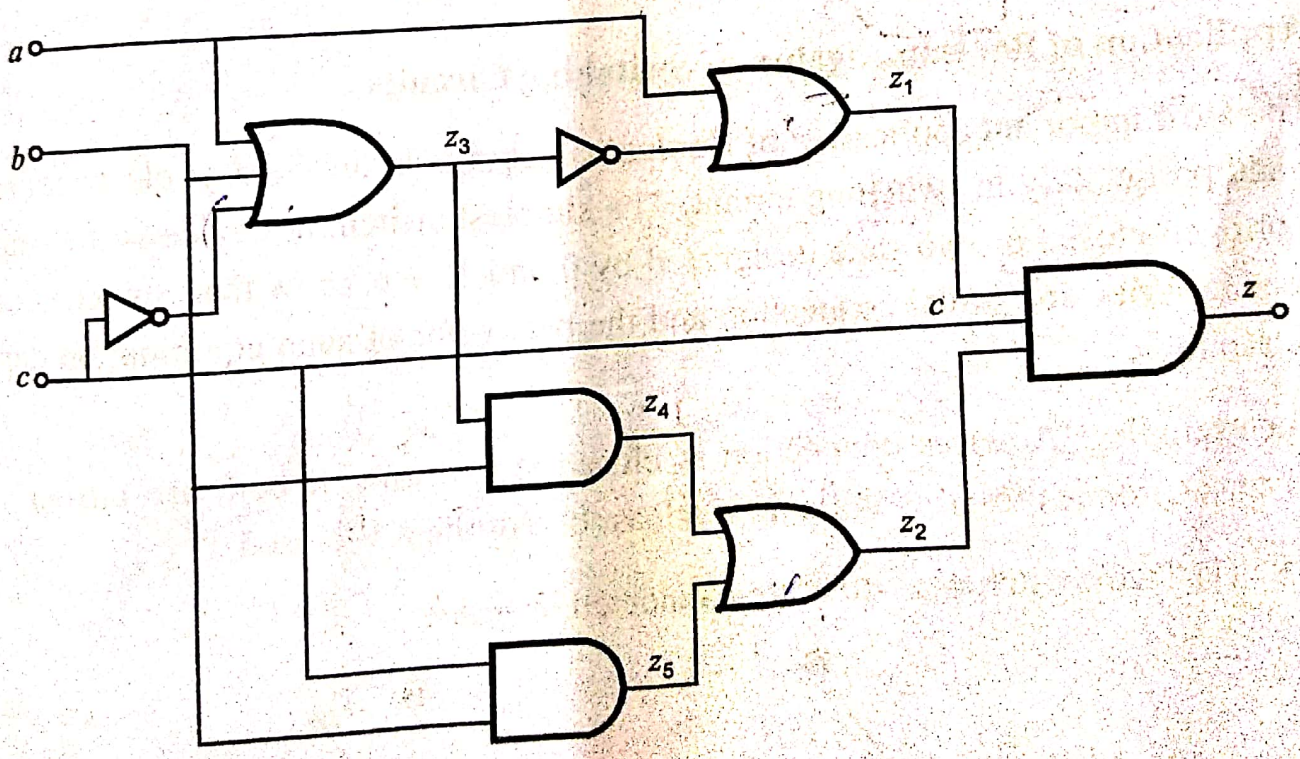
$$F = AB + \bar{A}C + AD$$

D

2.5.2 Minimisation of Boolean Functions :

Here are primarily concern with the problem of obtaining a Boolean function which is minimal according to some criterion, such as the minimum numbers of gates and/or inputs, and which is equivalent to a given Boolean function. Such Problems arise in the design of switching circuits. In a typical situation, the operational requirements of the circuit are stated in verbal form. These statements are subsequently transformed into a Boolean equation for the required output. The next step is to obtain an equation which is logically equivalent to the output equation which will result in a least expensive physical realisation. The desired circuit is finally obtained by replacing the Boolean connectives in the minimal equation by appropriate logic blocks such as OR gate, AND gates, inverter, NOR etc.

For example : Consider the circuit as shown below :



Here we wish to determine the output for z starts with that output gate and express as the inputs to this gate as outputs of earlier subcircuits. These intermediate outputs are treated in a similar fashion, until all the path to the original inputs have been traced.

Using this, we can write the equation for z as

$$z = z_1 \,.\, c \,.\, z_2 = (a + \bar{z}_3) \,.\, c \,.\, (z_4 + z_5)$$

$$= (a + \overline{a+b+\bar{c}}) \,.\, c \,.\, (b\, z_3 + b\, c)$$

$$= (a + \overline{a+b+\bar{c}}) . c . [b(a+b+\bar{c}) + bc]$$

We can simplify this equation considerably by applying certain Boolean identities

$$\therefore \quad z = [a + \overline{(a+b+\bar{c})}]\; c\; [b(a+b+\bar{c}) + bc]$$

$$= (a + \bar{a}\,\bar{b}\,c)\, c\, b\, (a+b+\bar{c}+c) = (a\,c\,b + \bar{a}\,b\,\bar{b}\,c)\,(a+b+1)$$

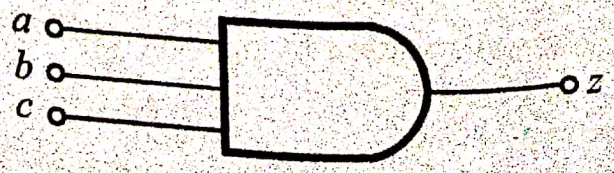
$$= (a\,c\,b + 0)(a+b) \qquad [\because\; x\,\bar{x} = 0]$$

$$= a\,a\,b\,c + a\,c\,b\,b \qquad [\because\; x\,x = x]$$

$$= a\,b\,c + a\,b\,c \qquad [\because\; x + x = x]$$

$$= a\,b\,c.$$

The use of Boolean equation as shown above is obvious. These provide not only a precise discription of the functioning of each subcircuity, but also permit the reduction of the original circuit to one three-input AND gate, *i.e.*, It is equivalent to the following circuit as shown above.



2.5.3 Application of Boolean Alg