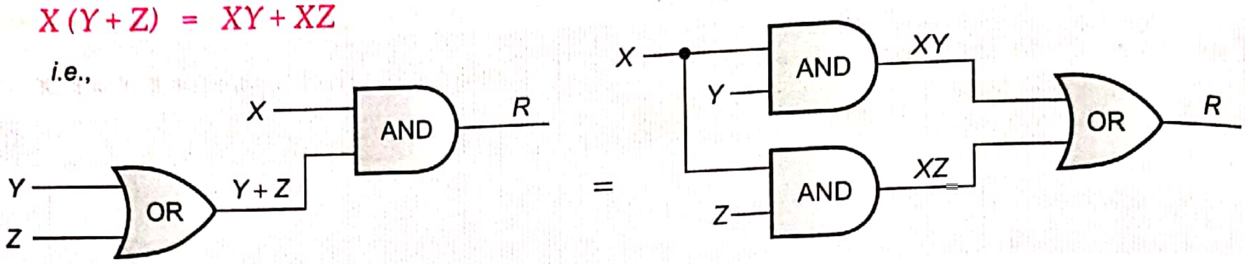
14.7.7 Distributive Law

This law states that

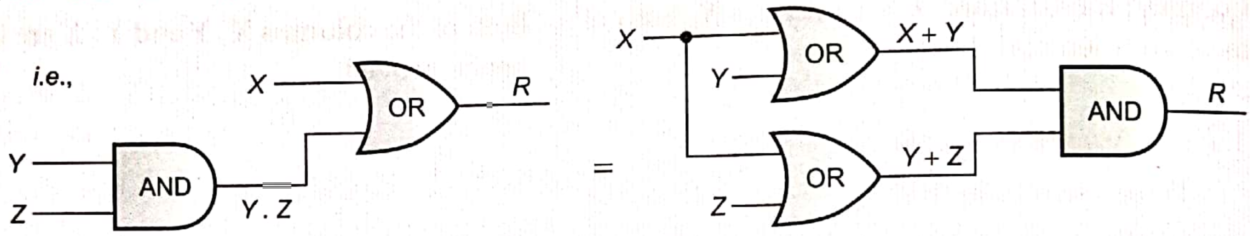
(*a*) $X(Y+Z) = XY+XZ$

i.e.,



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

i.e.,



Proof.

(*a*) Truth table for $X(Y+Z) = XY+XZ$ is given below :

Table 14.27 *Truth Table for $X(Y+Z) = XY+XZ$*

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

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

(*b*) Since rule (*b*) is dual of rule (*a*), hence it is also proved.

However, we are giving the algebraic proof of law $X + YZ = (X+Y)(X+Z)$

$$\text{R.H.S.} = (X+Y)(X+Z) = XX + XZ + XY + YZ$$

$$= X + XZ + XY + YZ \qquad (XX = X,\ \textit{Indempotence law})$$

$$= X + XY + XZ + YZ = X(1+Y) + XZ + YZ$$

$$= X.1 + XZ + YZ \qquad (1 + Y = 1,\ \textit{property of 0 and 1})$$

$$= X + XZ + YZ \qquad (X.1 = X,\ \textit{property of 0 and 1})$$

$$= X(1+Z) + YZ$$

$$= X.1 + YZ \qquad (1 + Z = 1,\ \textit{property of 0 and 1})$$

$$= X + YZ \qquad (X.1 = X,\ \textit{property of 0 and 1})$$

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

Hence proved.

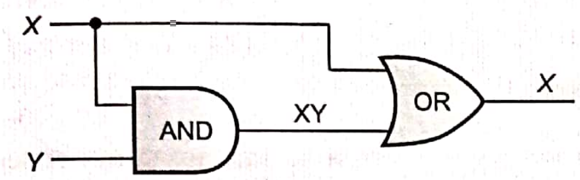
NOTE

$X + YZ$ expression is **sum of two product**-terms ($X.1, YZ$) and $(X+Y)(X+Z)$ is product of sum-terms $(X+Y, X+Z)$. So, this law is a useful one to convert a sum-of-product type expression to product-of-sum type expression and vice-versa.

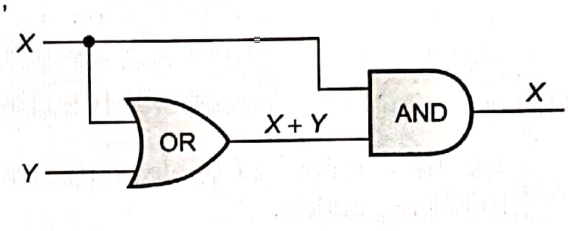
14.7.8 Absorption Law

According to this law

i.e., (*a*) $X + XY = X$



i.e., (*b*) $X(X + Y) = X$



Proof.

(*a*) X + XY = X

Truth table for $X + XY = X$ is given below :

Table 14.28 *Truth Table for X + XY = X*

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

Column X and $X + XY$ are identical. Hence proved. Also it can be proved algebraically as

$$\text{L.H.S.} = X + XY$$
$$= X(1 + Y)$$

Putting $1 + Y = 1$

(*ref. properties of 0, 1 Theorem 1*)

$X.1 = X$ = R.H.S. (*ref. properties of 0, 1*)

Hence proved.

(*b*) X (X + Y) = X

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

$$\text{L.H.S.} = X(X + Y) = X.X + XY$$
$$= X.X + XY$$
$$= X + XY$$

($X.X = X$: *Indempotence Law*)

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

(*using* $1 + Y = 1$, *properties of 0, 1*)

$= X$ ($X.1 = X$, *using property of 0, 1*)

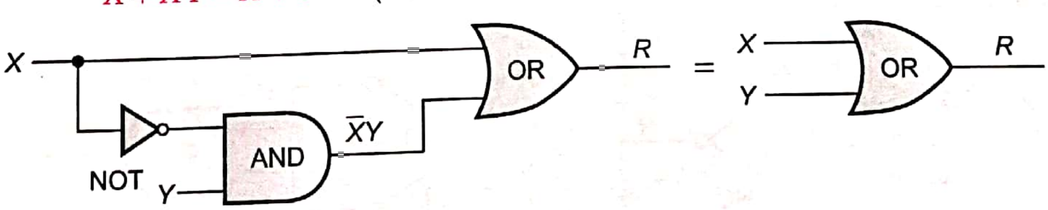
$=$ R.H.S

Hence proved.

14.7.9 Some Other Rules of Boolean Logic Algebra

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

$X + \overline{X}Y = X + Y$ (Sometimes also referred to as the *third distributive law*)



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

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

Proof. $\text{L.H.S} = X + \overline{X}Y$

$= X.1 + \overline{X}Y$ (Putting $X = X.1$, *property of 0 and 1*)

$= X(1+Y) + \overline{X}Y$ (Putting 1 as 1+Y, ∵ 1+Y = 1, property of 0 and 1)

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

$= X + Y(X + \overline{X})$ ($X + \overline{X} = 1$, complementarity law)

$= X + Y.1$ (Y.1 = Y, property of 0 and 1)

$= X + Y$

= R.H.S. Hence proved.

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

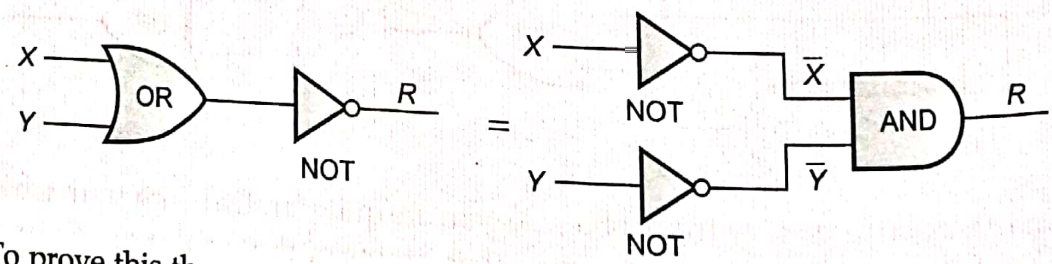
1.	*Properties of 0*	$0 + X = X$; $0 . X = 0$
2.	*Properties of 1*	$1 + X = 1$; $1 . X = X$
3.	*Indempotence law*	$X + X = X$; $X . X = X$
4.	*Involution*	$\overline{\overline{X}} = X$
5.	*Complementarity law*	$X + \overline{X} = 1$; $X . \overline{X} = 0$
6.	*Commutative law*	$X + Y = Y + X$; $X . Y = Y.X$
7.	*Associative law*	$X + (Y + Z) = (X + Y) + Z$; $X(YZ) = (XY)Z$
8.	*Distributive law*	$X(Y + Z) = XY + XZ$; $X + YZ = (X + Y)(X + Z)$
9.	*Absorption law*	$X + XY = X$; $X.(X + Y) = X$
10.	*Other (3rd distributive law)*	$X + \overline{X}Y = X + Y$

14.8 DEMORGAN'S THEOREMS

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

14.8.1 DeMorgan's First Theorem

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



Proof. To prove this theorem, we need to recall complementarity laws, which state that

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

i.e., a logical variable/expression when added with its complement produces the output as 1 and when multiplied with its complement produces the output as 0.

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

Let us assume that $P = X + Y$ where, P, X, Y are Logical/Boolean variables. Then, according to complementation law

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

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

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

and $(X + Y) . \overline{X}\,\overline{Y}$ must be equal to 0. *(as $X . \overline{X} = 0$)*

Let us first prove the first part, *i.e.,*

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

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

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

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

$$= 1.1 \qquad (ref.\ 1 + X = 1)$$

$$= 1$$

So first part is proved.

Now let us prove the second part *i.e.,*

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

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

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

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

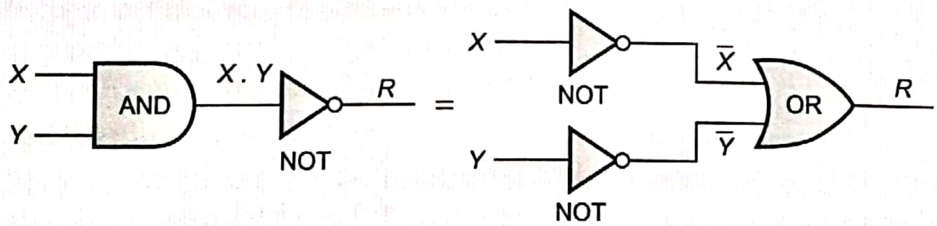
$$= 0 . \overline{Y} + \overline{X} . 0 \qquad (ref.\ X . \overline{X} = 0)$$

$$= 0 + 0 = \mathbf{0}$$

So, second part is also proved, thus : $\overline{X + Y} = \overline{X}\,\overline{Y}$

14.8.2 DeMorgan's Second Theorem

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



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

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

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

(*a*) $XY + (\overline{X} + \overline{Y}) = 1$ and

(*b*) $XY(\overline{X} + \overline{Y}) = 0$

To prove the *first part*

$$\text{L.H.S} = XY + (\overline{X} + \overline{Y})$$
$$= (\overline{X} + \overline{Y}) + XY \quad (ref.\ X + Y = Y + X)$$
$$= (\overline{X} + \overline{Y} + X).(\overline{X} + \overline{Y} + Y) \quad (ref.\ X + YZ = (X + Y)(X + Z))$$
$$= (X + \overline{X} + \overline{Y}).(\overline{X} + Y + \overline{Y})$$
$$= (1 + \overline{Y}).(\overline{X} + 1) \quad (ref.\ X + \overline{X} = 1)$$
$$= 1.1 \quad (ref.\ 1 + X = 1)$$
$$= 1 = \text{R.H.S}$$

Now the *second part* i.e.,

$$XY.(\overline{X} + \overline{Y}) = 0$$
$$\text{L.H.S} = XY.(\overline{X} + \overline{Y})$$
$$= XY\overline{X} + XY\overline{Y} \quad (ref.\ X(Y + Z) = XY + XZ)$$
$$= X\overline{X}Y + XY\overline{Y}$$
$$= 0.Y + X.0 \quad (ref.\ X.\overline{X} = 0)$$
$$= 0 + 0 = 0 = \text{R.H.S.}$$
$$XY.(\overline{X} + \overline{Y}) = 0$$
and $XY + (\overline{X} + \overline{Y}) = 1$

$\Rightarrow$ $\overline{XY} = \overline{X} + \overline{Y}$. Hence the theorem.

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

(i) Complement the entire function

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

(iii) Complement each of the individual variables.

NOTE

'Break the line, change the sign" to demorganize a boolean expression.

This process is called *demorganization*. For example,

$$\overline{\overline{AB} + \overline{A} + AB} = \overline{\overline{AB}}.\overline{\overline{A}}.\overline{AB} \quad \text{[Changed + to . and complemented individual expressions]}$$
$$= AB\ .\ A\ .\ \overline{AB} \quad [\because \overline{\overline{AB}} = AB \text{ and } \overline{\overline{A}} = A]$$
$$= AB.\overline{AB}.A$$
$$= 0.A \quad [AB.\overline{AB} = 0]$$
$$= 0 \quad [\because 0.A = 0]$$

Alternatively, you may solve it as follows :

$$\overline{\overline{AB} + \overline{A} + AB} = \overline{[\overline{A} + \overline{B}] + \overline{A} + AB]}$$
$$(\because \overline{AB} = \overline{A} + \overline{B}\ ;\ \textit{DeMorgan's 2}^{nd}\textit{ theorem})$$
$$= \overline{(\overline{A} + \overline{B}) + (\overline{A} + AB)}$$
$$= \overline{(\overline{A} + \overline{B})} \cdot \overline{(\overline{A} + AB)} \quad (\because \overline{(X + Y)} = \overline{X}.\overline{Y}\ \textit{DeMorgan's 2}^{nd}\textit{ theorem})$$
$$= \overline{\overline{A}}.\overline{\overline{B}}.(\overline{\overline{A}}.\overline{AB})$$
$$= A.B(A.(\overline{A} + \overline{B}))$$
$$= AB(A\overline{A} + A\overline{B})$$
$$= AB(0 + A\overline{B}) \quad (A\overline{A} = 0)$$
$$= AB.0 + ABA\overline{B} = 0 + AAB\overline{B}$$
$$= 0 + A\,A.0 \quad (B\overline{B} = 0)$$
$$= 0 + 0 = 0$$

Check Point 14.4

1. Which of the following Boolean equations is/are incorrect ? Write the correct forms of the incorrect ones :
   (a) $A + A' = 1$ (b) $A + 0 = A$
   (c) $A.1 = A$ (d) $AA' = 1$
   (e) $A + AB = A$ (f) $A(A + B)' = A$
   (g) $(A + B)' = A' + B$ (h) $(AB)' = A'B'$
   (i) $A + 1 = 1$ (j) $A + A = A$
   (k) $A + A'B = A + B$
   (l) $X + YZ = (X + Y)(X + A)$
2. Find the complement of the following functions applying De'Morgan's theorem
   (a) $F(x, y, z) = x'yz' + x'y'z$
   (b) $F(x, y, z) = x(y'z + yz)$
3. What is the logical product of several variables called ? What is the logical sum of several variables called ?
4. What is the procedure "Break the line, change the sign" ?

Basic Duality of Boolean Logic

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

For example, $X + X.Y = X$

Its dual will be $X.(X + Y) = X$ *(Remember change . to + and vice-versa ; complement 0 and 1.)*

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

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

In proving the theorems or rules of boolean algebra, it is then necessary to prove only one theorem, and the dual of the theorem follows necessarily. In effect, *all boolean algebra is predicated on this two-for-one basis.*

EXAMPLE 14.5 *Give the dual of the following result in Boolean algebra :* $X.X' = 0$ *for each X.*

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

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

EXAMPLE 14.6 *Give the dual of* $X + 0 = X$ *for each X.*

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

EXAMPLE 14.7 *State the principle of duality in boolean algebra and give the dual of the boolean expression :*

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

Solution. Principle of duality states that from every boolean relation, another boolean relation can be derived by

(*i*) changing each OR sign (+) to an AND (.) sign

(*ii*) changing each AND (.) sign to an OR (+) sign

(*iii*) replacing each 1 by 0 and each 0 by 1.

The new derived relation is known as the dual of the original relation.

Dual of $(X + Y).(\overline{X} + \overline{Z}).(Y + Z)$ will be : $(X.Y) + (\overline{X}.\overline{Z}) + (Y.Z) = XY + \overline{X}\,\overline{Z} + YZ$

14.9 SIMPLIFYING A BOOLEAN EXPRESSION

Boolean logic or Boolean expressions are practically implemented through logic gates, thus a Boolean expression should always be in minimized form before practically implementing them so as to have a simplified circuitary and less cost. The Boolean expressions can be of one of the following types :

(*i*) **Sum-of-Products type.** Here the product-terms (*e.g.*, ABC, XY, $\overline{Y}Z$, $\overline{Z}\,\overline{X}$ etc.) are added (*i.e.*, ANDed) *e.g.*, $ABC + \overline{B}\,\overline{C}$, $XY + \overline{Y}Z + \overline{Z}\,\overline{X}$ are all sum-of-product type expressions.

(*ii*) **Product-of-Sums type.** Here the sum-terms (*e.g.*, $X + Y$, $\overline{Y} + Z$, $A + \overline{B} + C$ etc.) are multiplied (*i.e.*, ORed), *e.g.*, $(X + Y)(\overline{Y} + Z)$, $(A + B + C)(A + \overline{B} + \overline{C})$, $(P + Q)(\overline{Q} + \overline{P} + S)(\overline{R} + \overline{P} + S)$, are all product-of-sums type expressions.

There are many ways to minimize or simplify Boolean expressions but here we are covering one method of doing so the algebraic method of simplifying a Boolean expression.

14.9.1 Algebraic Method

This method makes use of boolean postulates, rules and theorems to simplify the expressions. Following examples illustrate this way of simplifying Boolean expressions.

EXAMPLE 14.8 *Simplify* $A\overline{B}C\overline{D} + A\overline{B}CD + ABC\overline{D} + ABCD$.

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

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

$= AC(\overline{B} + B) = AC.1 = AC$ $\quad(\overline{B} + B = 1)$

EXAMPLE 14.9 *Reduce the expression* $\overline{XY} + \overline{X} + XY$.

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

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

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

$= \overline{X} + \overline{Y} + XY$ $\quad(\because \overline{X} + \overline{X} = \overline{X}$ as $X + X = X)$

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

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

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

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

$= 1$ (*putting* $\overline{X} + 1 = 1$ as $1 + X = 1$)

EXAMPLE 14.10 *Minimise* $AB + \overline{AC} + A\overline{B}C(AB + C)$.

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

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

$= AB + \overline{AC} + 0 + A\overline{B}CC$ (*putting* $B\overline{B} = 0$)

$= AB + \overline{AC} + A\overline{B}.C$ (*putting* $C.C = C$)

$= AB + \overline{A} + \overline{C} + A\overline{B}C$ (*putting* $\overline{AC} = \overline{A} + \overline{C}$, *DeMorgan's 2nd theorem*)

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

$= \overline{A} + B + \overline{C} + A\overline{B}C$ (*putting* $\overline{A} + AB = A + B$ *because* $X + \overline{X}Y = X + Y$)

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

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

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

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

$= A + \overline{A} + B + \overline{C} = 1 + B + \overline{C}$ (*putting* $A + \overline{A} = 1$)

$= 1$ (*as* $1 + X = 1$ *i.e., anything added to 1 results in 1*)

EXAMPLE 14.11 *Reduce* $\overline{X}\,\overline{Y}\,\overline{Z} + \overline{X}Y\overline{Z} + X\overline{Y}\,\overline{Z} + XY\overline{Z}$.

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

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

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

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

$= \overline{Z}.1 = \overline{Z}$ $\quad(\overline{X} + X = 1)$

14.10 MORE ABOUT LOGIC GATES

We have covered three basic logic gates NOT, OR, AND so far. But there are some more logic gates also which are derived from three basic gates (*i.e.*, AND, OR and NOT). These gates are more popular than NOT, OR and AND and are widely used in industry. This section introduces NOR, NAND, XOR, XNOR gates.

14.10.1 NOR Gate

The **Nor Gate** has two or more input signals but only one output signal. If all the inputs are 0 (*i.e.*, low), then the output signal is 1 (high).

If either of the two inputs is 1 (*high*), the output will be 0 (*low*). NOR *gate is nothing but inverted* OR *gate.*

NOR GATE

The Nor Gate has two or more input signals but only one output signal. If all the inputs are 0 (*i.e.*, low), then the output signal is 1 (high).

The NOR gate can have as many inputs (2 or more inputs) as desired. No matter how many inputs are there, the action of NOR gate is the same *i.e.*, All 0 (*low*) inputs produce output as 1.

Following truth Tables (14.29 and 14.30) illustrate NOR action.

Table 14.29 *2-input NOR gate*

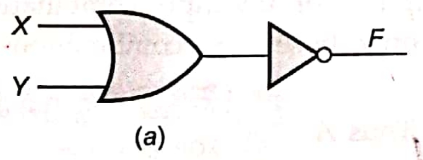
X	Y	F
0	0	1
0	1	0
1	0	0
1	1	0

$F = \overline{X + Y}$

Table 14.30 *3-input NOR gate*

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

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



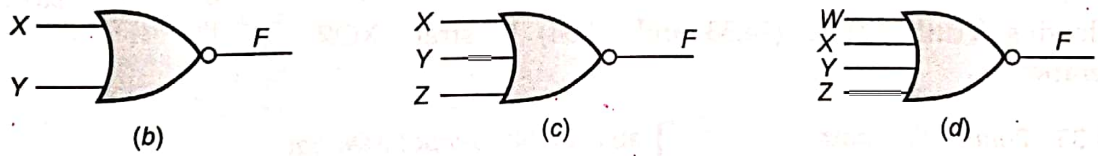


Figure 14.7 (*a*) Logical meaning of NOR gate (*b*) 2 input NOT gate (*c*) 3 input NOR gate (*d*) 4 input NOR gate

14.10.2 NAND Gate

The **NAND Gate** has two or more input signals but only one output signal. If all of the inputs are 1 (high), then the output produced is 0 (low).

NAND *gate is inverted* AND *gate.* Thus, for all 1 (*high*) inputs, it produces 0 (*low*) output, otherwise for any other input combination, it produces a 1 (*high*) output. NAND gate can also have as many inputs as desired.

NAND GATE

The NAND Gate has two or more input signals but only one output signal. If all of the inputs are 1 (high), then the output produced is 0 (low).

NAND action is illustrated in following Truth Tables (14.31 and 14.32).

Table 14.31 *2-input NAND gate*

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

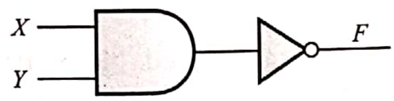
$F = \overline{XY}$

Table 14.32 *3-input NAND gate*

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

$F = \overline{XYZ}$

The logical meaning of NAND gate can be shown as follows :



The symbols of 2, 3, 4 input NAND gates are given below :

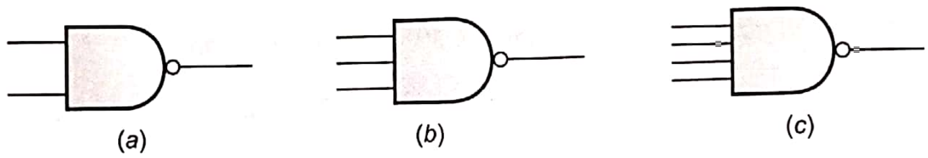


Figure 14.8 (*a*) 2-input NAND gate (*b*) 3-input NAND gate (*c*) 4-input NAND gate.

14.10.3 XOR Gate (Exclusive OR Gate)

The **XOR Gate** can also have two or more inputs but produces one output signal. Exclusive-OR gate is different from OR gate. OR gate produces output 1 for any input combination having one or more 1's, but XOR gate produces output 1 for only those input combinations that have odd number of 1's.

In boolean algebra ⊕ sign stands for XOR operation. Thus A XOR B can be written as $A \oplus B$.

Following Truth Tables (14.33 and 14.34) illustrate XOR operation.

XOR GATE

XOR gate produces output 1 for only those input combinations that have odd number of 1's.

Table 14.33 *2-input XOR gate*

No. of 1's even/odd	X	Y	F
Even	0	0	0
Odd	0	1	1
Odd	1	0	1
Even	1	1	0

Table 14.34 *3-input XOR gate*

No. of 1's	X	Y	Z	F
Even	0	0	0	0
Odd	0	0	1	1
Odd	0	1	0	1
Even	0	1	1	0
Odd	1	0	0	1
Even	1	0	1	0
Even	1	1	0	0
Odd	1	1	1	1

The symbols of XOR gates are given below :

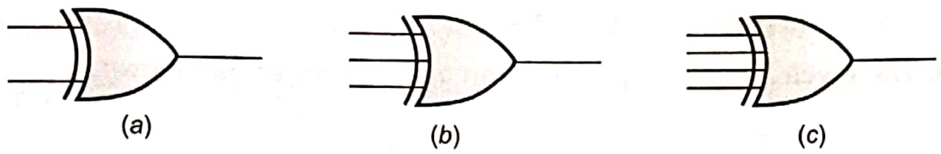


Figure 14.9 (a) 2-input XOR gate (b) 3-input XOR gate (c) 4-input XOR gate.

XOR addition can be summarised as follows :

$0 \oplus 0 = 0, \quad 0 \oplus 1 = 1, \quad 1 \oplus 0 = 1, \quad 1 \oplus 1 = 0$

NOTE

Remember odd number of 1's produce output 1.

14.10.4 XNOR Gate (Exclusive NOR gate)

The **XNOR Gate** is logically equivalent to an inverted XOR *i.e.*, XOR gate followed by a NOT gate (inventor). Thus XNOR produces 1 (high) output when the input combination has even number of 1's. Following tables (14.35 and 14.36) illustrate XNOR action.

XNOR GATE

XNOR gate produces output 1 for only those input combinations that have even number of 1's.

Table 14.35 *2-input XNOR gate*

No. of 1's	X	Y	F
Even	0	0	1
Odd	0	1	0
Odd	1	0	0
Even	1	1	1

Table 14.36 *3-input XNOR gate*

No. of 1's	X	Y	Z	F
Even	0	0	0	1
Odd	0	0	1	0
Odd	0	1	0	0
Even	0	1	1	1
Odd	1	0	0	0
Even	1	0	1	1
Even	1	1	0	1
Odd	1	1	1	0

NOTE

The XNOR Gate is logically equivalent to an inverted XOR *i.e.*, XOR gate followed by a NOT gate (inventor).

Following are the XNOR gate symbols :

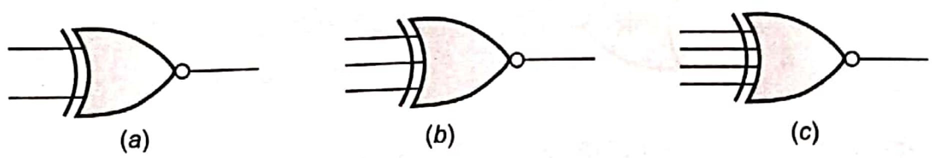


Figure 14.10 (a) 2-input XNOR gate (b) 3-input XNOR gate (c) 4-input XNOR gate.

The bubble (small circle), on the outputs of NAND, NOR, XNOR gates represents complementation.

Now that we are familiar with logic gates, we can use them in designing logic circuits.

EXAMPLE 14.12 *Design a circuit to realise the following :*

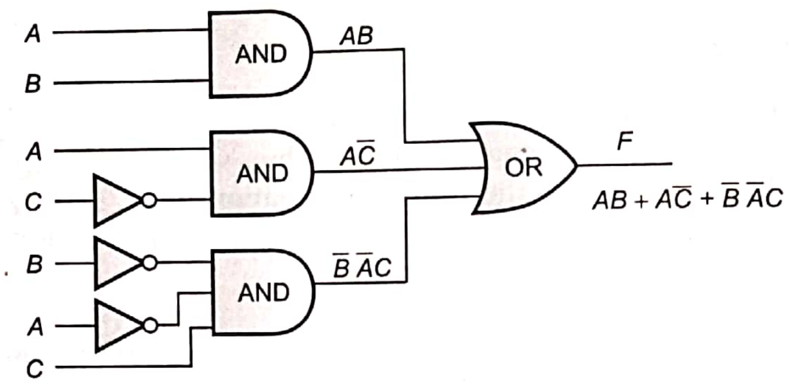
$$F(a, b, c) = AB + A\overline{C} + \overline{B}\,\overline{A}C.$$

Solution. The given boolean expression can also be written as follows :

$$F(a, b, c) = A.B + A.\overline{C} + \overline{B}.\overline{A}.C$$

or $F(a, b, c) =$ (A AND B) OR (A AND (NOT C)) OR ((NOT B) AND (NOT B) AND C)

Now these logical operators can easily be implemented in form of logic gates. Thus circuit diagram for above expression will be as follows :



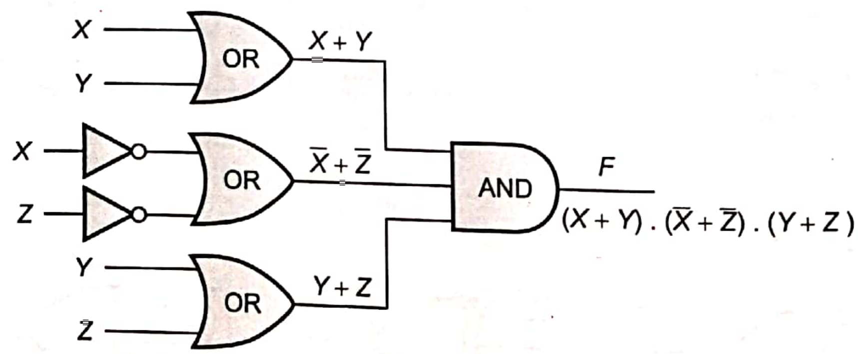
EXAMPLE 14.13 *Draw the diagram of digital circuit for the function :*

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

Solution. Above expression can also be written as :

$F(X, Y, Z) =$ (X OR Y) AND ((NOT X) OR (NOT Z)) AND (Y OR Z)

Thus circuit diagram will be



14.10.5 NAND to NAND and NOR to NOR design

We can design circuits using AND, OR, NOT gates as we have done so far, but NAND and NOR gates are more popular as these are less expensive and easier to design. And also other switching functions (AND, OR) can easily by implemented using NAND/NOR gates. Thus NAND, NOR gates are also referred to as *Universal Gates*.

NAND-to-NAND Logic

AND and OR operations from NAND gates are shown below :

AND operation

AND operation using NAND is

$$X . Y = (X \text{ NAND } Y) \text{ NAND } (X \text{ NAND } Y)$$

Proof. X NAND Y

$= \overline{X . Y}$

$= \overline{X} + \overline{Y}$ *(DeMorgan's Second Theorem)*

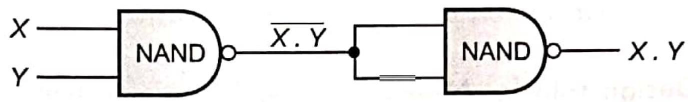


Figure 14.11 AND operation using NAND-NAND logic.

(X NAND Y) NAND (X NAND Y)

$= (\overline{X} + \overline{Y}) \text{ NAND } (\overline{X} + \overline{Y})$

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

$= \overline{(\overline{X} + \overline{Y})} + \overline{(\overline{X} + \overline{Y})}$ *(DeMorgan's Second Theorem)*

$= \overline{\overline{X}} . \overline{\overline{Y}} + \overline{\overline{X}} . \overline{\overline{Y}}$ *(DeMorgan's First Theorem)* $(\overline{\overline{X}} = X)$

$= X . Y + X . Y$

$= XY$ *(i.e., X* **AND** *Y)* $(X + X = X)$

OR operation

OR operation using NAND is

$$X + Y = (X \text{ NAND } X) \text{ NAND } (Y \text{ NAND } Y)$$

Proof. X NAND X

$= \overline{X . X}$

$= \overline{X} + \overline{X}$ *(DeMorgan's Second Theorem)*

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

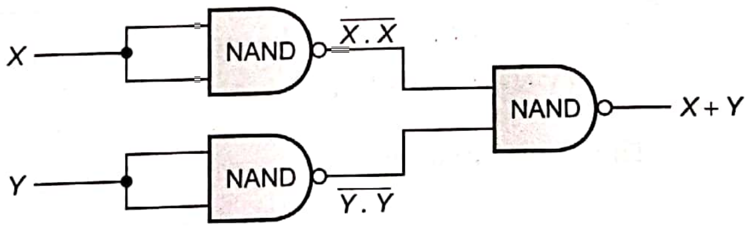


Figure 14.12 OR operation NAND to NAND logic.

Similarly, Y NAND Y $= \overline{Y}$

Therefore, (X NAND X) NAND (Y NAND Y)

$= \overline{X} \text{ NAND } \overline{Y}$

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

$= \overline{\overline{X}} + \overline{\overline{Y}}$ *(De Morgan's Second Theorem)*

$= X + Y$ *(i.e., X* **OR** *Y)* $(\overline{\overline{X}} = X, \overline{\overline{Y}} = Y)$

NOT operation

NOT operation using NAND is

NOT X = X NAND X

Proof. $X \text{ NAND } X = \overline{X.X} = \overline{X}$ $(\because X.X = X)$

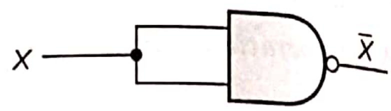


Figure 14.13 NOT operation using NAND gate.

NAND-to-NAND logic is best suited for boolean expression in *Sum-of-Products* form.

Design rule for NAND-TO-NAND logic Network (only for 2 level circuit)

1. Derive simplified sum-of-products expression.
2. Draw a circuit diagram using AND, OR gates
3. Replace all basic gates (AND, OR, NOT) with NAND gates

For example, $XY\overline{Z} + Z\overline{X}$ (it is a sum-of-products expression) can be drawn as follows :

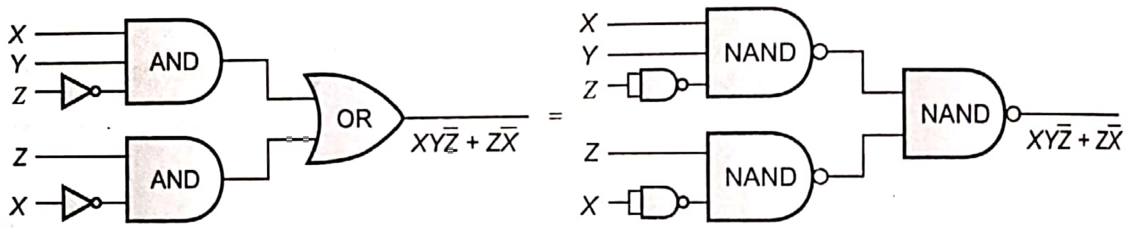


Figure 14.14 (*a*) AND-to-OR circuit (*b*) NAND-to-NAND circuit

EXAMPLE 14.14 *Draw the diagram of a digital circuit for the function*

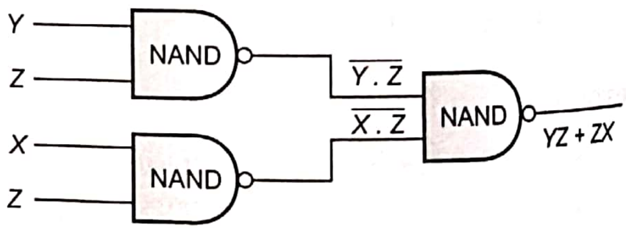
$F(X, Y, Z) = YZ + XZ$ *using NAND gates only.*

Solution.

$F(X, Y, Z) = YZ + XZ$ can be written as

= (Y NAND Z) NAND (X NAND Z)

Thus logic circuit is



EXAMPLE 14.15 *Draw the diagram of digital circuit for*

$F(a, b, c) = AB + BC + CD$ *using NAND-to-NAND logic.*

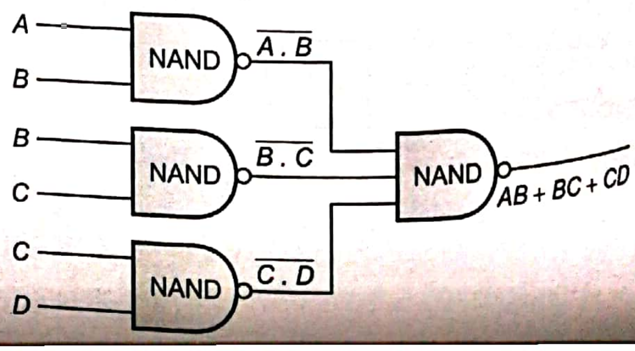
Solution. $F(a, b, c) = AB + BC + CD$

= (*A* NAND *B*) NAND (*B* NAND *C*) NAND (*C* NAND *D*)

Thus logic circuit is

NOTE

NAND-to-NAND logic is best suited for boolean expression in *Sum-of-Products* form.

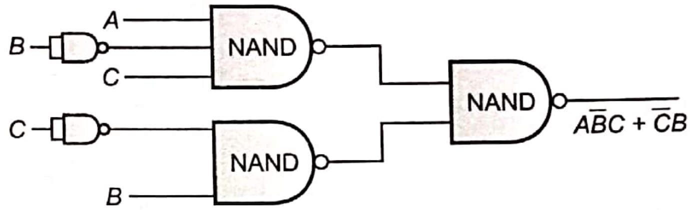


EXAMPLE 14.16 *Draw the circuit diagram for :* $F = A\overline{B}C + \overline{C}B$ *using NAND-to-NAND logic only.*

Solution. $F = A\,\overline{B}\,C + \overline{C}\,B$

$= ((A)\text{ NAND (NOT } B\text{) NAND (C)) NAND ((NOT } C\text{) NAND } B)$

Thus logic circuit is



NOR-to-NOR logic

AND and OR operations can be implemented in NOR-to-NOR form as shown below.

OR operation

$A + B = (A \text{ NOR } B) \text{ NOR } (A \text{ NOR } B)$

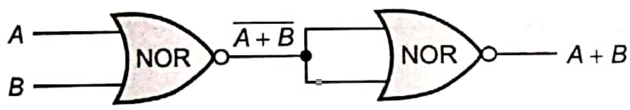


Figure 14.15 OR operation using NOR to NOR logic.

AND operation

$A\,.\,B = (A \text{ NOR } A) \text{ NOR } (B \text{ NOR } B)$

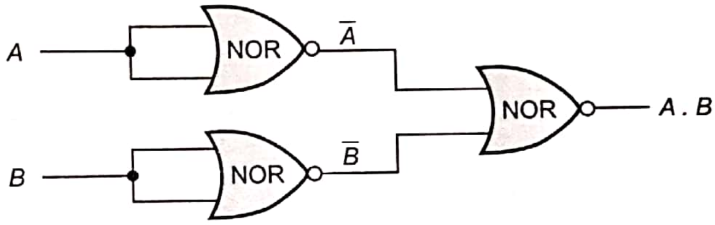


Figure 14.16 AND operation using NOR to NOR logic.

NOT operation

$\text{NOT } X = X \text{ NOR } X$

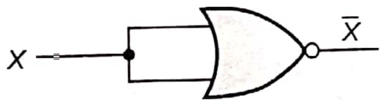


Figure 14.17 NOT operation using NOR .

> **NOTE**
>
> **NOR-to-NOR** logic is best suited for boolean expression in Product-of-Sums form.

Design rule for NOR-to-NOR logic network (only for 2-level circuits)

1. Derive a simplified Product-of-Sums form of the expression.
2. Draw a circuit diagram using OR, AND gates.
3. Finally substitute NOR gates for OR, NOT and AND gates.

For example, $(X + Y)(Y + Z)(Z + X)$ [It is a product-of-sums type expression] can be implemented as follows :

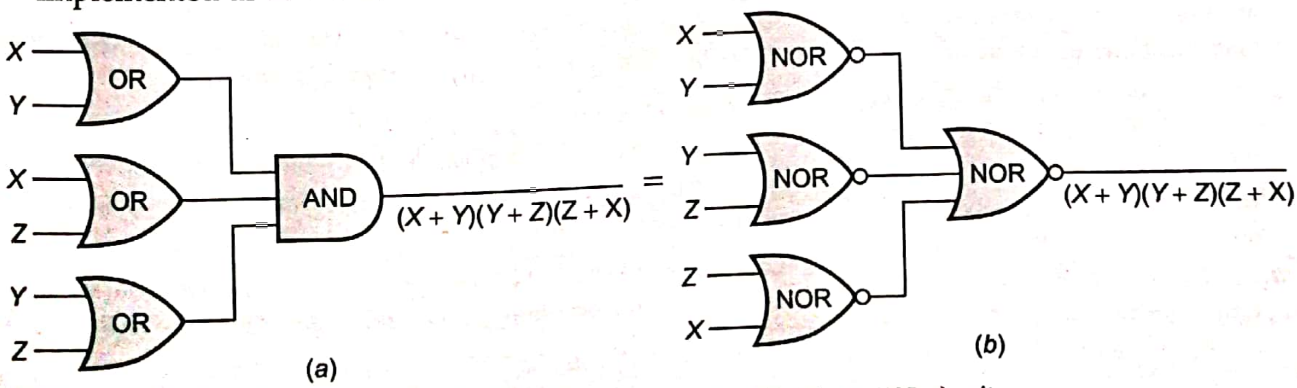
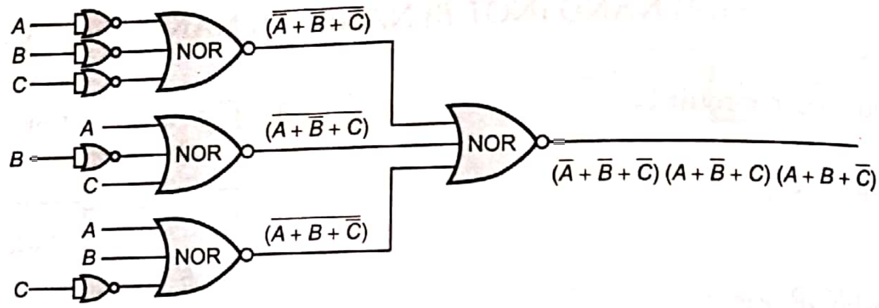


Figure 14.18 (*a*) OR-to-AND circuit (*b*) NOR-to-NOR circuit

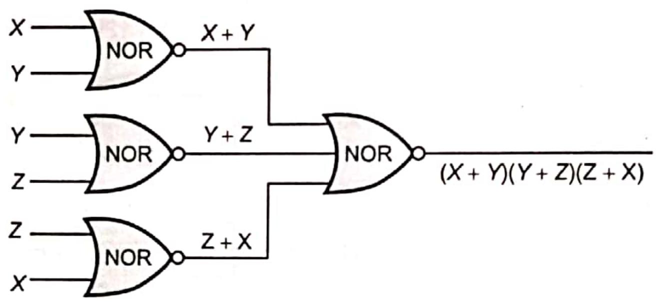
EXAMPLE 14.17 *Represent* $(\overline{A}+\overline{B}+\overline{C})(A+\overline{B}+\overline{C})(A+B+\overline{C})$ *in NOR-to-NOR logic network.*

Solution.



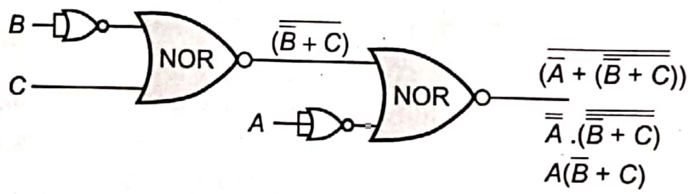
EXAMPLE 14.18 *Represent* $(X+Y)(Y+Z)(Z+X)$ *in NOR-to-NOR form.*

Solution. $(X + Y)(Y + Z)(Z + X) = (X \text{ NOR } Y) \text{ NOR } (Y \text{ NOR } Z) \text{ NOR } (Z \text{ NOR } X)$



EXAMPLE 14.19 *Show* $A(\overline{B}+C)$ *using NOR gates only.*

Solution.



Check Point 14.5

1. Why are NAND and NOR gates called Universal gates ?
2. Which gates are called Universal gates and why ?
3. State the purpose of reducing the switching functions to the minimal form.
4. Draw a logic circuit diagram using NAND or NOR only to implement the Boolean function $F(a, b) = a'b' + ab$.
5. What is inverted AND gate called ? What is inverted OR gate called ?
6. When does an XOR gate produce a high output ? When does an XNOR gate produce a high output ?

LET US REVISE

- *The decision which results into either YES (TRUE) or NO (FALSE) is called a binary decision.*
- *The statements which can be determined to be True or False are called* ***logical statements*** *or* ***truth functions.***
- *Truth values are TRUE and FALSE or 1 and 0.*
- *Truth table is a table which represents all the possible values of logical variables/ statements along with all the possible results of the given combinations of values.*
- *If result of any logical statement or expression is always TRUE or 1, it is called* ***tautology*** *and if the result is always FALSE or 0 it is called* ***fallacy.***