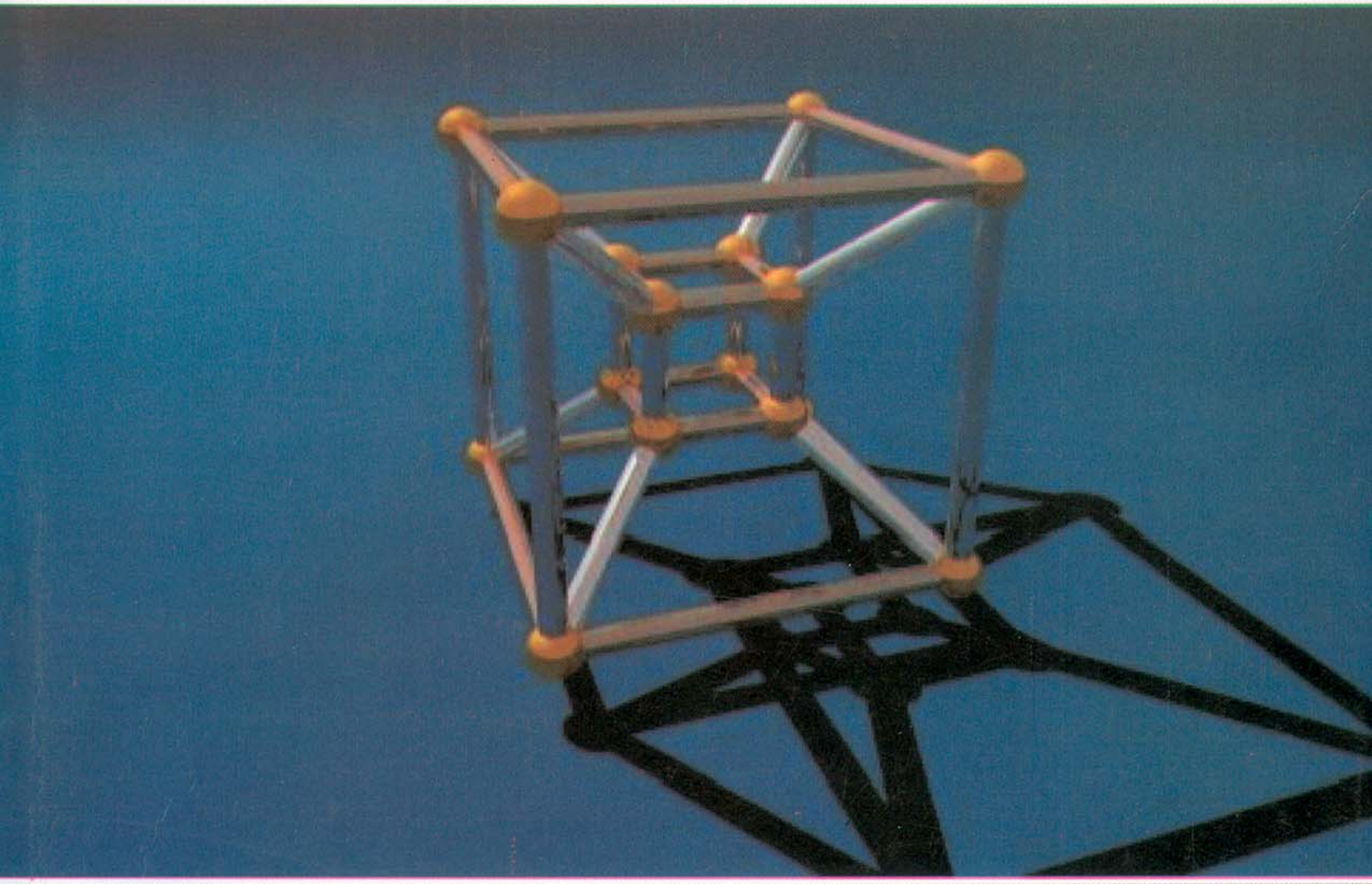
NEW AGE

SECOND EDITION

FUNDAMENTAL APPROACH TO DISCRETE MATHEMATICS



D.P. Acharjya
Sreekumar

NEW AGE INTERNATIONAL PUBLISHERS

10

Boolean Algebra

10.0 INTRODUCTION

For centuries mathematicians felt there was a connection between mathematics and logic, but no one could find this missing link before George Boole. In 1854 he introduced symbolic logic known as Boolean Algebra, Boolean function, Boolean expression, Boolean ring and many more honour the nineteenth century mathematician George Boole. Each variable in Boolean algebra has either of two values: true or false. The purpose of this two - state algebra was to solve logic problems.

Almost after a century of Boole's work, it was observed by C.E. Shannon in 1938, that Boolean algebra could be used to analyze electrical circuits. This was developed by Shannon while he analyzed telephone switching circuits. Because of Shannon's work, engineers realized that Boolean algebra could be applied to Computer electronics.

This chapter introduces the Gate, Combinatorial Circuits, Boolean Expression, Boolean Algebra, Boolean Functions and Various Normal Forms.

10.1 GATES

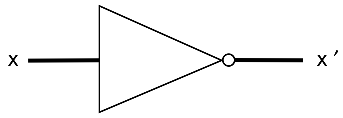
In logic we have discussed about the logical connectives $\neg$, $\wedge$ and $\vee$. The connectives $\wedge$ and $\vee$ can be considered as circuits connected in series and parallel respectively. A circuit with one or more input signals but only one output signal is known as a gate. Gates are digital circuits because of input and output signals, which are either low or high. Gates are also called logical circuits because they can be analyzed with Boolean algebra. In gates, the connectives $\neg$, $\wedge$ and $\vee$ are usually denoted by the symbols $'$, . and + respectively. The block diagrams for different gates are discussed below.

10.1.1 A NOT Gate

A NOT gate receives input x, where x is a bit (binary digit) and produces output x' where

$$x' = \begin{cases} 1 \text{ if } x = 0 \\ 0 \text{ if } x = 1 \end{cases}$$

The output state is always the opposite of the input state. The output is sometimes called the complement of the input. A NOT gate is drawn as shown in the following figure.

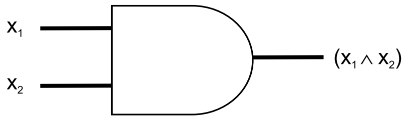


10.1.2 An AND Gate

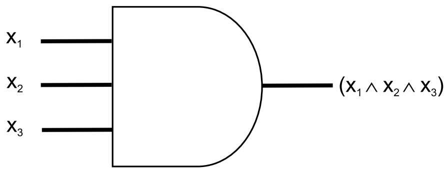
An AND gate receives inputs x_1 and x_2, where x_1 and x_2 are bits, and produces output $(x_1 \wedge x_2)$, where

$$(x_1 \wedge x_2) = \begin{cases} 1 & \text{if } x_1 = x_2 = 1 \\ 0 & \text{otherwise} \end{cases}$$

An AND gate may have more inputs also but the output is always one. An AND gate is drawn as shown in the following figure.



(2 input AND gate)



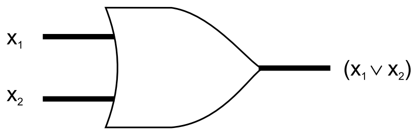
(3 input AND gate)

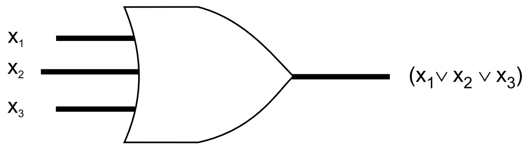
10.1.3 An OR Gate

An OR gate receives inputs x_1 and x_2, where x_1 and x_2 are bits, and produces output $(x_1 \vee x_2)$, where

$$(x_1 \vee x_2) = \begin{cases} 1 & \text{if } x_1 = 1 \text{ or } x_2 = 1 \\ 0 & \text{otherwise} \end{cases}$$

An OR gate may have more inputs also but the output is always one. An OR gate is drawn as shown in the following figure.





(2 input OR gate) (3 input OR gate)

The logic tables for the basic AND, OR and NOT gates are given below.

x_1	x_2	$(x_1 \wedge x_2)$
1	1	1
1	0	0
1	0	0
0	0	0

x_1	x_2	$(x_1 \vee x_2)$
1	1	1
1	0	1
0	1	1
0	0	0

x	x'
1	0
0	1

■ 10.2 MORE LOGIC GATES

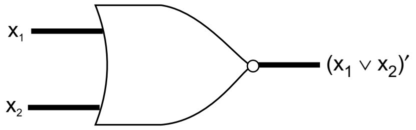
There are some other types of gates which are useful and frequently used in computer science. These are called NAND, NOR, XOR and XNOR gates. The block diagrams for these different gates are given below.

10.2.1 NOR Gate

A NOR gate receives inputs x_1 and x_2 where x_1 and x_2 are bits, and produces output $(x_1 \vee x_2)'$, where

$$(x_1 \vee x_2)' = \begin{cases} 1 & \text{if } x_1 = x_2 = 0 \\ 0 & \text{otherwise} \end{cases}$$

A NOR gate may have more inputs also, but the output is always one. A NOR gate is drawn as shown in the following figure.



(2 input NOR gate)

According to de Morgan's first theorem, we have

$$(x_1 \vee x_2)' = x_1' \wedge x_2' \quad i.e., \quad (x_1 + x_2)' = x_1' . x_2'$$

10.2.2 NAND Gate

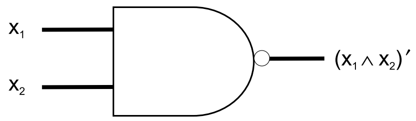
A NAND gate receives inputs x_1 and x_2, where x_1 and x_2 are bits, and produces output $(x_1 \wedge x_2)'$, where

$$(x_1 \wedge x_2)' = \begin{cases} 1 & \text{if } x_1 = 0 \text{ or } x_2 = 0 \\ 0 & \text{otherwise} \end{cases}$$

A NAND gate may have more inputs also, but the output is always one. A NAND gate is drawn as shown in the following figure.

According to the de Morgan's second theorem we have

$$(x_1 \wedge x_2)' = x_1' + x_2' \; i.e., \; (x_1 . x_2)' = x_1' + x_2'$$

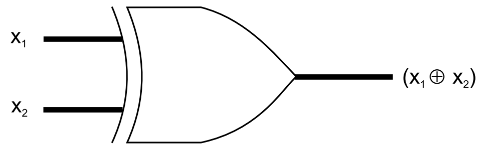


10.2.3 XOR Gate (Exclusive OR Gate)

A XOR gate receives inputs x_1 and x_2, where x_1 and x_2 are bits, and produces output $(x_1 \overline{\vee} x_2)$ or $(x_1 \oplus x_2)$, where

$$(x_1 \oplus x_2) = \begin{cases} 1 & \text{if } x_1 = 1 \text{ or } x_2 = 1 \text{ but not both} \\ 0 & \text{otherwise} \end{cases}$$

From the definition, it is clear that, the Exclusive OR gate, *i.e*. XOR gate produces 1 that have an odd number of 1's. A XOR gate may have more inputs also, but the output is always one. A XOR gate is drawn as shown in the following figure.



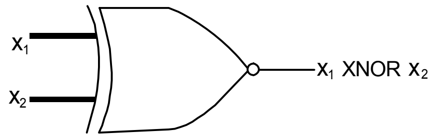
10.2.4 XNOR Gate (Exclusive NOR Gate)

A XNOR gate receives inputs x_1 and x_2, where x_1 and x_2 are bits, and produces output x_1 XNOR x_2 where

$$x_1 \text{ XNOR } x_2 = \begin{cases} 1 & \text{if } x_1 \text{and } x_2 \text{ are same bits} \\ 0 & \text{otherwise} \end{cases}$$

XNOR gate may have more inputs also, but the output is always one. In this case it recognizes even-parity words. Even parity means a word has an even number of 1's. For example 11100111 has even parity because it contains six 1's. Odd parity means a word has an odd number of 1's. For example 1101 has odd parity because it contains three 1's.

A XNOR gate is drawn as shown in the following figure.



The logic tables for the above NOR, NAND, XOR and XNOR gates are given below.

x_1	x_2	$(x_1 \wedge x_2)'$
1	1	0
1	0	1
0	1	1
0	0	1

x_1	x_2	$(x_1 \vee x_2)'$
1	1	0
1	0	0
0	1	0
0	0	1

x_1	x_2	$(x_1 \oplus x_2)$
1	1	0
1	0	1
0	1	1
0	0	0

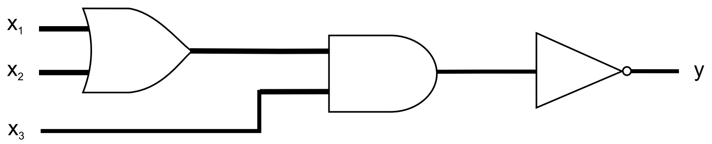
x_1	x_2	$(x_1$ XNOR $x_2)$
1	1	1
1	0	0
0	1	0
0	0	1

■ 10.3 COMBINATORIAL CIRCUIT

In digital computer electronics, there are only two possibilities, *i.e.,* 0 and 1, for the smallest, indivisible object. These 0 and 1 are known as binary digits (bit). A bit in one part of a circuit is transmitted to another part of the circuit as a voltage. Thus two voltage levels are needed. *i.e.*, high voltage level and low voltage level. A high voltage level communicates 1 whereas a low voltage level communicates 0.

A combinatorial circuit is a circuit which produces an unique output for every combination of inputs. A combinatorial circuit has no memory, previous inputs and the state of the system do not affect the output of a combinatorial circuit. These circuits can be constructed using gates which we have already discussed.

Let us consider the circuit

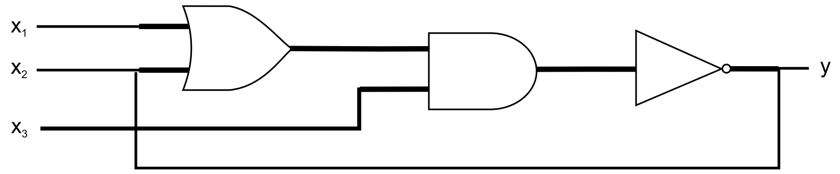


x_1	x_2	x_3	y
1	1	1	0
1	1	0	1
0	1	1	0
1	0	1	0
0	0	1	1
0	1	0	1
1	0	0	1
0	0	0	1

The logic table for the above circuit is given in the above table. From the table it is clear that the output y is uniquely defined for each combination of inputs x_1, x_2 and x_3. Therefore, the circuit is a combinatorial circuit.

If $x_1 = 1$ and $x_2 = 1$, then the output of OR gate is 1. Now the input for AND gate is 1 and 0, so the output of AND gate is 0. Since the input to the NOT gate is 0, the output $y = 1$.

Consider another circuit as

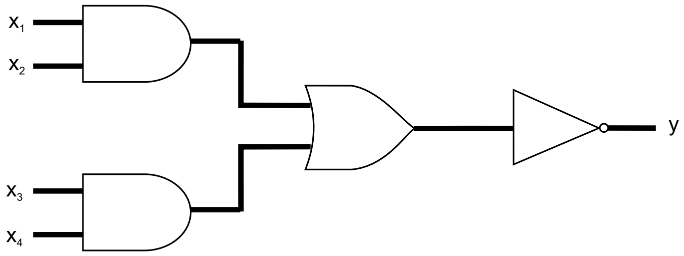


The above circuit is not a combinatorial circuit, as the output y is not defined uniquely for every combination of inputs x_1, x_2 and x_3.

■ 10.4 BOOLEAN EXPRESSION

Any expression built up from the variables $x_1, y_1, z_1, x_2, y_2, z_2, \ldots$ by applying the operations $\wedge$, $\vee$ and $'$ a finite number of times. If X_1 and X_2 are Boolean expressions, then (X_1), X_2', $(X_1 \wedge X_2)$ and $(X_1 \vee X_2)$ are also Boolean expressions. The output of a combinatorial circuit is also a Boolean expression.

Let us consider the combinatorial circuit as



The Boolean expression to the above circuit is given as $((x_1 \wedge x_2) \vee (x_3 \wedge x_4))'$.

10.4.1 Theorem

If $\wedge$, $\vee$ and $'$ are connectives defined earlier, then the following properties hold.

(*i*) Associative Laws: For all $a, b, c \in \{0, 1\}$

$$(a \wedge b) \wedge c = a \wedge (b \wedge c) \text{ and}$$
$$(a \vee b) \vee c = a \vee (b \vee c)$$

(*ii*) Identity Laws: For all $a \in \{0, 1\}$

$$(a \wedge 1) = a \quad \text{and} \quad (a \vee 0) = a$$

(*iii*) Commutative Laws: For all $a, b \in \{0, 1\}$

$$(a \wedge b) = (b \wedge a) \quad \text{and}$$
$$(a \vee b) = (b \vee a)$$

(*iv*) Complement Laws: For all $a \in \{0, 1\}$

$$(a \wedge a') = 0 \quad \text{and}$$
$$(a \vee a') = 1$$

(*v*) Distributive Laws: For all $a, b, c \in \{0, 1\}$

$$a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c) \quad \text{and}$$
$$a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$$

Proof: Proofs of (i), (ii), (iii) and (iv) are immediate consequences of the definitions. We prove only the first distributive law. Here we simply evaluate both sides of law for all possible values of $a, b, c \in \{0, 1\}$ and verify that in each case we obtain the same result.

We must show that $a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$

a	b	c	$(b \wedge c)$	$a \vee (b \wedge c)$	$(a \vee b)$	$(a \vee c)$	$(a \vee b) \wedge (a \vee c)$
1	1	1	1	1	1	1	1
0	0	1	0	0	0	1	0
0	1	0	0	0	1	0	0
1	0	0	0	1	1	1	1
1	1	0	0	1	1	1	1
1	0	1	0	1	1	1	1
0	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0

Therefore, $a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$

10.4.2 de Morgan's Laws

If x_1, x_2 are bits, *i.e.*, $x_1, x_2 \in \{0, 1\}$, then

(i) $(x_1 \wedge x_2)' = x_1' \vee x_2'$

(ii) $(x_1 \vee x_2)' = x_1' \wedge x_2'$

Proof: We prove only the first de Morgan's Law.

i.e., $(x_1 \wedge x_2)' = x_1' \vee x_2'$

Construct the logical table.

x_1	x_2	$(x_1 \wedge x_2)'$	x_1'	x_2'	$x_1' \vee x_2'$
1	1	0	0	0	0
1	0	1	0	1	1
0	1	1	1	0	1
0	0	1	1	1	1

Therefore, $(x_1 \wedge x_2)' = x_1' \vee x_2'$.

■ 10.5 EQUIVALENT COMBINATORIAL CIRCUITS

Two combinatorial circuits, each having inputs $x_1, x_2, \ldots, x_n$ are said to be equivalent if they produce the same outputs for same inputs *i.e.*, the output for both the circuits remains same if the circuits receive same inputs.

Consider the following combinatorial circuits.

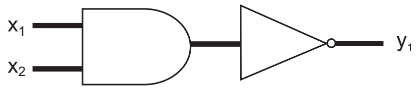


Figure 1

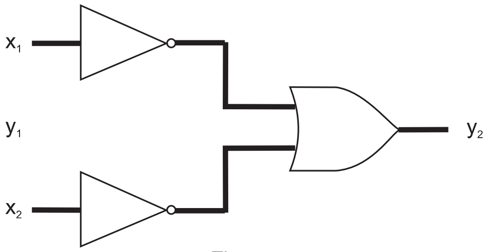


Figure 2

The logic tables for both the circuits are given below, which are identical.

x_1	x_2	y_1
1	1	0
1	0	1
0	1	1
0	0	1

x_1	x_2	y_2
1	1	0
1	0	1
0	1	1
0	0	1

From the logic tables it is clear that both the combinatorial circuits are equivalent.

■ 10.6 BOOLEAN ALGEBRA

A Boolean algebra B consists of a set S together with two binary operations $\wedge$ and $\vee$ on S, a singular operation $'$ on S and two specific elements 0 and 1 of S such that the following laws hold. We write B = {S, $\wedge$, $\vee$, $'$, 0, 1}.

(*a*) Associative Laws: For all $a, b, c \in$ S

$$(a \wedge b) \wedge c = a \wedge (b \wedge c)$$

and $$(a \vee b) \vee c = a \vee (b \vee c)$$

(*b*) Commutative Laws: For all $a, b \in$ S

$$(a \wedge b) = (b \wedge a)$$

and $$(a \vee b) = (b \vee a)$$

(*c*) Distributive Laws: For all $a, b, c \in$ S

$$a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$$

and $$a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$$

(*d*) Identity Laws: For all $a \in$ S

$$(a \wedge 1) = a \quad \text{and} \quad (a \vee 0) = a$$

(*e*) Complement Laws: For all $a \in$ S

$(a \wedge a') = 0$ and $(a \vee a') = 1$

10.6.1 Theorem

In a Boolean algebra; if $(a \vee b) = 1$ and $(a \wedge b) = 0$, then $b = a'$, *i.e.,* the complement is unique.

Proof: Suppose that $(a \vee b) = 1$ and $(a \wedge b) = 0$

Now $b = (b \vee 0)$ [Identity law]

$= b \vee (a \wedge a')$ [Complement law]

$= (b \vee a) \wedge (b \vee a')$ [Distributive law]

	$= (a \vee b) \wedge (b \vee a')$	[Commutative law]
	$= 1 \wedge (b \vee a')$	[Given condition]
	$= (b \vee a')$	[Identity law]
This implies	$b = (b \vee a')$	...(i)
Again a'	$= (a' \vee 0)$	[Identity law]
	$= a' \vee (a \wedge b)$	[Given condition]
	$= (a' \vee a) \wedge (a' \vee b)$	[Distributive law]
	$= 1 \wedge (a' \vee b)$	[Complement law]
	$= (a' \vee b)$	[Identity law]
	$= (b \vee a')$	[Commutative law]
This implies	$a' = (b \vee a') = b$	[Equation 1]

10.6.2 Theorem

In a Boolean algebra B = (S, $\vee$, $\wedge$, $'$, 0, 1) ; the following properties hold.

(a) Idempotent Laws: For all $x \in$ S

$$(x \vee x) = x \quad \text{and} \quad (x \wedge x) = x$$

(b) Bound Laws: For all $x \in$ S

$$(x \vee 1) = 1 \quad \text{and} \quad (x \wedge 0) = 0$$

(c) Absorption Laws: For all $x, y \in$ S

$$x \wedge (x \vee y) = x \quad \text{and} \quad x \vee (x \wedge y) = x$$

(d) Involution Laws: For all $x \in$ S

$$(x')' = x$$

(e) 0 and 1 Laws: $0' = 1$and $1' = 0$

(f) de Morgan's Laws: For all $x, y \in$ S

$$(x \wedge y)' = x' \vee y'$$

and

$$(x \vee y)' = x' \wedge y'$$

Proof: (a)	$x = x \vee 0$	[Identity law]
	$= x \vee (x \wedge x')$	[Complement law]
	$= (x \vee x) \wedge (x \vee x')$	[Distributive law]
	$= (x \vee x) \wedge 1$	[Complement law]
	$= (x \vee x)$	[Identity law]
Therefore,	$(x \vee x) = x$	
Again	$x = x \wedge 1$	[Identity law]
	$= x \wedge (x \vee x')$	[Complement law]
	$= (x \wedge x) \vee (x \wedge x')$	[Distributive law]
	$= (x \wedge x) \vee 0$	[Complement law]
	$= (x \wedge x)$	[Identity law]
Therefore,	$(x \wedge x) = x$	
(b)	$(x \vee 1) = (x \vee 1) \wedge 1$	[Identity law]
	$= (x \vee 1) \wedge (x \vee x')$	[Complement law]
	$= ((x \vee 1) \wedge x) \vee ((x \vee 1) \wedge x')$	[Distributive law]

$$= ((x \wedge x) \vee (1 \wedge x)) \vee ((x \wedge x') \vee (1 \wedge x'))$$
$$= (x \vee (1 \wedge x)) \vee ((x \wedge x') \vee (1 \wedge x')) \quad \text{[Idempotent law]}$$
$$= (x \vee x) \vee ((x \wedge x') \vee x')) \quad \text{[Identity law]}$$
$$= (x \vee x) \vee (0 \vee x') \quad \text{[Complement law]}$$
$$= x \vee (0 \vee x') \quad \text{[Idempotent law]}$$
$$= (x \vee x') \quad \text{[Identity law]}$$
$$= 1 \quad \text{[Complement law]}$$

Therefore, $(x \vee 1) = 1$

Again, $(x \wedge 0) = (x \wedge 0) \vee 0$ [Identity law]

$$= (x \wedge 0) \vee (x \wedge x') \quad \text{[Complement law]}$$
$$= ((x \wedge 0) \vee x) \wedge ((x \wedge 0) \vee x') \quad \text{[Distributive law]}$$
$$= ((x \vee x) \wedge (0 \vee x)) \wedge ((x \vee x') \wedge (0 \vee x'))$$
$$= ((x \vee x) \wedge x) \wedge ((x \vee x') \wedge x') \quad \text{[Identity law]}$$
$$= (x \wedge x) \wedge ((x \vee x') \wedge x') \quad \text{[Idempotent law]}$$
$$= (x \wedge x) \wedge ((x \wedge x') \vee (x' \wedge x')) \quad \text{[Distributive law]}$$
$$= x \wedge ((x \wedge x') \vee (x' \wedge x')) \quad \text{[Idempotent law]}$$
$$= x \wedge (0 \vee (x' \wedge x')) \quad \text{[Complement law]}$$
$$= x \wedge (0 \vee x') \quad \text{[Idempotent law]}$$
$$= x \wedge x' \quad \text{[Identity law]}$$
$$= 0 \quad \text{[Complement law]}$$

Therefore, $(x \wedge 0) = 0$

(*c*) $x \wedge (x \vee y) = (x \vee 0) \wedge (x \vee y)$ [Identity law]

$$= x \vee (0 \wedge y) \quad \text{[Distributive law]}$$
$$= x \vee (y \wedge 0) \quad \text{[Commutative law]}$$
$$= x \vee 0 \quad \text{[Bound law]}$$
$$= x \quad \text{[Identity law]}$$

Therefore, $x \wedge (x \vee y) = x$

Again, $x \vee (x \wedge y) = (x \wedge 1) \vee (x \wedge y)$ [Identity law]

$$= x \wedge (1 \vee y) \quad \text{[Distributive law]}$$
$$= x \wedge (y \vee 1) \quad \text{[Commutative law]}$$
$$= x \wedge 1 \quad \text{[Bound law]}$$
$$= x \quad \text{[Identity law]}$$

Therefore, $x \vee (x \wedge y) = x$

(*d*) $x' \vee x = x \vee x'$ [Commutative law]

$$= 1 \quad \text{[Complement law]}$$

i.e., $x' \vee x = 1$

Also, $x' \wedge x = x \wedge x'$ [Commutative law]

$$= 0 \quad \text{[Complement law]}$$

i.e., $x' \wedge x = 0$

Thus we have $x' \vee x = 1$ and $x' \wedge x = 0$

Therefore, $x = (x')'$ *i.e.*, $(x')' = x$

(*e*) We know that $(0 \vee 1) = (1 \vee 0) = 1$

i.e., $(0 \vee 1) = 1$

Again by Theorem $(0 \wedge 1) = (1 \wedge 0) = 0$

Thus we have $(0 \vee 1) = 1$ and $(0 \wedge 1) = 0$

Therefore, $1 = 0'$ and $0' = 1$

Similarly we also have $(1 \vee 0) = 1$ and $(1 \wedge 0) = 0$

Therefore, $0 = 1'$ and $1' = 0$

(*f*) Let $a = (x \wedge y)$ and $b = (x' \vee y')$

Now
$$\begin{aligned}
(a \vee b) &= (x \wedge y) \vee b \\
&= (x \vee b) \wedge (y \vee b) && \text{[Distributive law]} \\
&= (x \vee (x' \vee y')) \wedge (y \vee (x' \vee y')) \\
&= ((x \vee x') \vee y') \wedge (y \vee (x' \vee y')) && \text{[Associative law]} \\
&= (1 \vee y') \wedge (y \vee (x' \vee y')) && \text{[Complement law]} \\
&= (1 \vee y') \wedge (y \vee (y' \vee x')) && \text{[Commutative law]} \\
&= (1 \vee y') \wedge ((y \vee y') \vee x') && \text{[Associative law]} \\
&= (1 \vee y') \wedge (1 \vee x') && \text{[Complement law]} \\
&= 1 \wedge 1 && \text{[Bound law]} \\
&= 1 && \text{[Idempotent law]}
\end{aligned}$$

Again,
$$\begin{aligned}
(a \wedge b) &= (x \wedge y) \wedge (x' \vee y') \\
&= ((x \wedge y) \wedge x') \vee ((x \wedge y) \wedge y') && \text{[Distributive law]} \\
&= ((y \wedge x) \wedge x') \vee ((x \wedge y) \wedge y') && \text{[Commutative law]} \\
&= (y \wedge (x \wedge x')) \vee (x \wedge (y \wedge y')) && \text{[Associative law]} \\
&= (y \wedge 0) \vee (x \wedge 0) && \text{[Complement law]} \\
&= 0 \vee 0 && \text{[Bound law]} \\
&= 0 && \text{[Idempotent law]}
\end{aligned}$$

Therefore, $(a \vee b) = 1$ and $(a \wedge b) = 0$

This implies that $b = a'$ *i.e.*, $a' = b$

i.e., $(x \wedge y)' = (x' \vee y')$

Similarly the other de Morgan's law $(x \vee y)' = (x' \wedge y')$ can be proved.

■ 10.7 DUAL OF A STATEMENT

The dual of a statement involving Boolean expressions is obtained by replacing 0 by 1, 1 by 0, $\wedge$ by $\vee$, and $\vee$ by $\wedge$. Two Boolean expressions are said to be dual of each other if one expression is obtained from other by replacing 0 by 1, 1 by 0, $\wedge$ by $\vee$, and $\vee$ by $\wedge$.

Consider the statement $(x \wedge y)' = x' \vee y'$. The dual of above statement is $(x \vee y)' = x' \wedge y'$. Similarly the Boolean expressions $(x \wedge 1) = x$ and $(x \vee 0) = x$ are dual of each other.

10.7.1 Theorem

In Boolean algebra, the dual of a theorem is also a theorem.

Proof: Suppose that T is a theorem in Boolean algebra. Then there is a proof P of T involving definitions of a Boolean algebra. Let P_1 be the sequence of statements obtained by replacing 0 by 1, 1 by 0, $\wedge$ by $\vee$ and $\vee$ by $\wedge$. Then P_1 is a proof of the dual of T.

■ 10.8 BOOLEAN FUNCTION

Let B = (S, $\vee$, $\wedge$, ′, 0, 1) be a Boolean algebra and let X $(x_1, x_2, x_3, \ldots, x_n)$ be a Boolean expression in '*n*' variables. A function $f: B^n \to B$ is called a Boolean function if f is of the form

$$f(x_1, x_2, x_3, \ldots, x_n) = X(x_1, x_2, x_3, \ldots, x_n)$$

Let us consider the example of a Boolean function $f: B^3 \to B$; B = {0, 1} defined by

$$f(x_1, x_2, x_3) = x_1 \wedge (x_2 \vee \bar{x}_3)$$

The inputs and outputs are given in the following table.

x_1	x_2	x_3	$f(x_1, x_2, x_3)$
1	1	1	1
1	1	0	1
1	0	1	0
0	1	1	0
1	0	0	1
0	1	0	0
0	0	1	0
0	0	0	0

10.8.1 Representations of Boolean Functions

We have seen that Boolean functions are nothing but the evaluation functions of Boolean expressions. It is also to be noted that two Boolean expressions give rise to the same evaluation function if and only if they are equivalent. Therefore, we identify a Boolean function with any of the equivalent Boolean expressions, whose evaluation function gives it.

This gives rise to the representation of a Boolean function. There are several ways for representing Boolean functions. These are

(*a*) Tabular Representation
(*b*) *n* Space Representation
(*c*) Cube Representation

Here we will discuss only tabular representation.

Tabular Representation: We know that, a Boolean function is completely determined by its evaluation over any Boolean algebra. In tabular representation, the procedure is very clear. We consider a row R of the table where the output is 1. We then form the combination $(x_1 \wedge x_2 \wedge x_3 \wedge \ldots \wedge x_n)$ and place a bar over each x_i whose value is 0 in row R. The combination formed is 1 if and only if x_i have the values given in row R. We thus OR the terms to obtain the Boolean expression.

To clear the procedure let us consider the Boolean function given by the following table.

x_1	x_2	x_3	$f(x_1, x_2, x_3)$
1	1	1	1 ← Row 1
1	1	0	0
1	0	1	1 ← Row 3
0	1	1	0
1	0	0	0
0	1	0	1 ← Row 6
0	0	1	0
0	0	0	0

From the table it is clear that, the output is 1 for the rows 1, 3 and 6. Consider the first row of the table and the combination is $(x_1 \wedge x_2 \wedge x_3)$ as $x_1 = x_2 = x_3 = 1$. Similarly for third row of the table we may construct the combination $(x_1 \wedge \overline{x_2} \wedge x_3)$ as $x_1 = 1, x_2 = 0, x_3 = 1$. Thus for sixth row the combination is $(\overline{x_1} \wedge x_2 \wedge \overline{x_3})$.

Therefore, the Boolean function $f(x_1, x_2, x_3)$ is given as

$$f(x_1, x_2, x_3) = (x_1 \wedge x_2 \wedge x_3) \vee (x_1 \wedge \overline{x_2} \wedge x_3) \vee (\overline{x_1} \wedge x_2 \wedge \overline{x_3}).$$

■ 10.9 VARIOUS NORMAL FORMS

In this section we will discuss about two normal forms *i.e.*, disjunctive normal form and conjunctive normal form.

10.9.1 Disjunctive Normal Form

A Boolean function $f: B^n \to B$ which consists of a sum of elementary products is called the disjunctive normal form of the given function f.

Let $f: B^n \to B$ is a Boolean function. If f is not identically zero, let $A_1, A_2, A_3, \ldots, A_k$ denote the elements A_i of B_2^n, for which $f(A_i) = 1$,

where, $\quad A_i = (a_1, a_2, \ldots, a_n)$.

For each A_i set $\quad m_i = (y_1 \wedge y_2 \wedge y_3 \wedge \ldots \wedge y_n)$

where, $\quad y_i = \begin{cases} x_i & \text{if } a_i = 1 \\ x_i' & \text{if } a_i = 0 \end{cases}$

Then, $f(x_1, x_2, x_3, \ldots, x_n) = m_1 \vee m_2 \vee m_3 \vee \ldots \vee m_k$. This representation of a Boolean function is called the disjunctive normal form.

Let us consider the Boolean function $(x_1 \oplus x_2)$. The truth table for this function is given below.

x_1	x_2	$(x_1 \oplus x_2)$
1	1	0
1	0	1
0	1	1
0	0	0

The disjunctive normal form of this function is given as

$$(x_1 \oplus x_2) = (x_1 \wedge \bar{x}_2) \vee (\bar{x}_1 \wedge x_2)$$

10.9.2 Conjunctive Normal Form

A Boolean function $f: B^n \to B$ which consists of a product of elementary sums is called the conjunctive normal form of the given function f.

Let $f: B^n \to B$ is a Boolean function. If f is not identically one, let $A_1, A_2, A_3, \ldots, A_k$ denote the elements A_i of B_2^n, for which $f(A_i) = 0$,

where, $$A_i = (a_1, a_2, a_3, \ldots, a_n).$$

For each A_i set

$$M_i = (y_1 \vee y_2 \vee y_3 \vee \ldots \vee y_n)$$

where, $$y_i = \begin{cases} x_i & \text{if } a_i = 0 \\ x_i' & \text{if } a_i = 1 \end{cases}$$

Then, $f(x_1, x_2, x_3, \ldots, x_n) = M_1 \wedge M_2 \wedge M_3 \wedge \ldots \wedge M_k$. This representation of a Boolean function is called the conjunctive normal form.

Let us consider the Boolean function $(x_1 \oplus x_2)$. The truth table for this function is given below.

x_1	x_2	$(x_1 \oplus x_2)$
1	1	0
1	0	1
0	1	1
0	0	0

From the table it is clear that, the output is 0 for the rows 1 and 4. Consider the first row of the table and the combination is $(\bar{x}_1 \vee \bar{x}_2)$. Similarly for the fourth row the combination is $(x_1 \vee x_2)$. So the conjunctive normal form for this function is given as

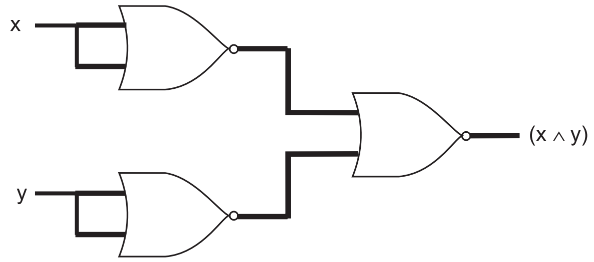
$$(x_1 \oplus x_2) = (\bar{x}_1 \vee \bar{x}_2) \wedge (x_1 \vee x_2)$$

Note: A term of the form $(y_1 \wedge y_2 \wedge y_3 \wedge \ldots \wedge y_n)$, where each y_i is either x_i or $\bar{x}_i$ is called a minterm where as a term of the form $(y_1 \vee y_2 \vee y_3 \vee \ldots \vee y_n)$, where each y_i is either x_i or $\bar{x}_i$ is called a maxterm.

SOLVED EXAMPLES

Example 1 *Construct an AND gate using three NOR gates.*

Solution: The output to an AND gate is $(x \wedge y)$, if the inputs are x and y. The output to a NOR gate is $\left(\overline{x \vee y}\right)$, if the inputs are x and y. The gating network is given further:

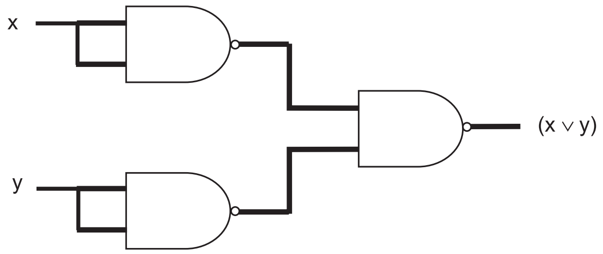


From the diagram given above it is clear that the output to the first NOR gate is $\left(\overline{x \vee x}\right) = \bar{x}$.

Similarly the output to the second NOR gate is $\left(\overline{y \vee y}\right) = \bar{y}$. Therefore, the output to the final NOR gate is $(x \wedge y)$.

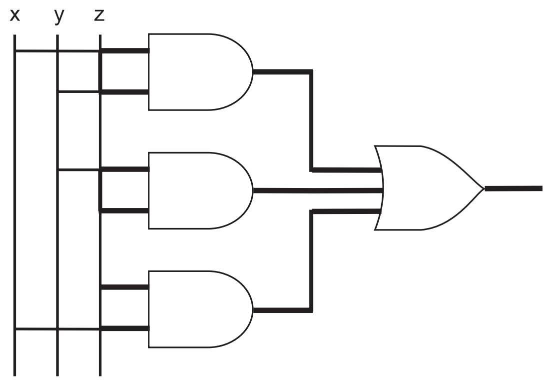
Example 2 *Construct an OR gate using three NAND gates.*

Solution: The output to an OR gate is $(x \vee y)$, if the inputs are x and y. The output to an NAND gate is $\left(\overline{x \wedge y}\right)$, if the inputs are x and y. The gating network is given as below.



Example 3 *Describe a gating network corresponding to the statement (x . y) + (y . z) + (z . x).*

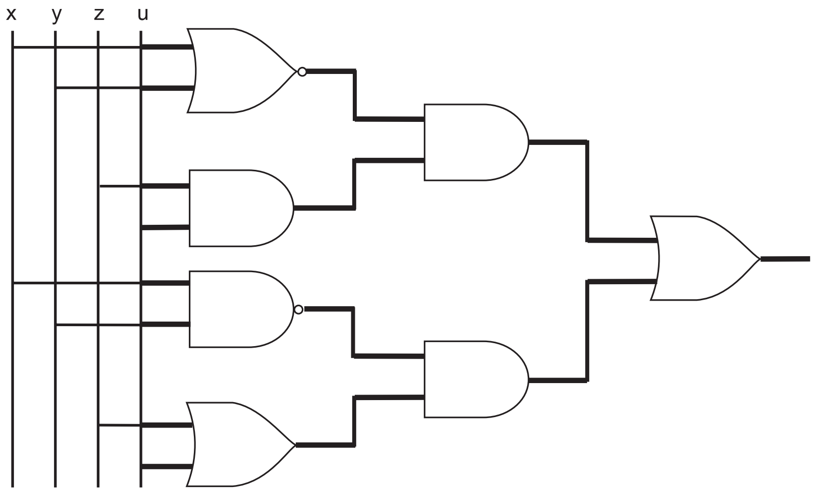
Solution: Given statement is $(x \cdot y) + (y \cdot z) + (z \cdot x)$. The gating network is given as



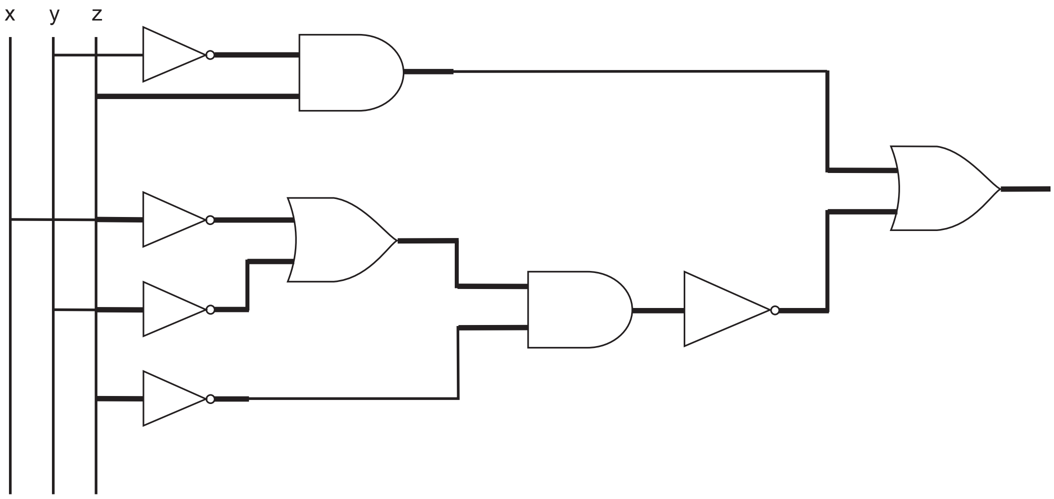
Example 4 *Describe a gating network corresponding to the statement*

$$\left(\overline{x+y}\right)(z \cdot u)+\left(\overline{x \cdot y}\right)(z+u)$$

Solution: Given statement is $\left(\overline{x+y}\right)(z \cdot u)+\left(\overline{x \cdot y}\right)(z+u)$. The gating network is given as below.



Example 5 *Describe the output of the following gating network.*



Solution: Consider the gating network given above. The output to the above gating network is given as

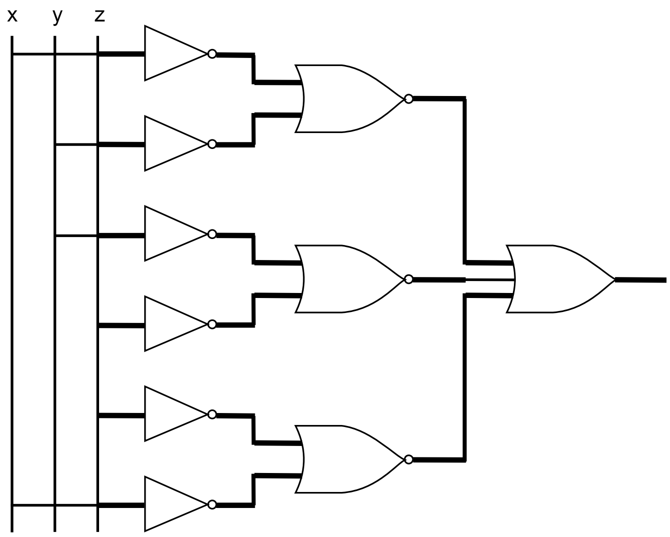
$$(\overline{y}\,.z) + (\overline{(\overline{x}+\overline{y}).\overline{z}}) = \overline{y}z + (\overline{\overline{x}+\overline{y}}) + \overline{\overline{z}} \qquad \text{[de Morgan's law]}$$

$$= \overline{y}z + \overline{\overline{x}}\,.\overline{\overline{y}} + z \qquad \text{[de Morgan's law]}$$

$$= \overline{y}z + xy + z$$

Example 6 *Construct a gating network using inverter and OR gate corresponding to the statement* $(x\,.\,y) + (y\,.\,z) + (z\,.\,x)$.

Solution: Given statement is $(x\,.\,y) + (y\,.\,z) + (z\,.\,x)$. The gating network is given futher.



Example 7 *Find the value of the Boolean expression given below for x = 1, y = 1 and z = 0.*

$$(x \wedge (y \vee (x \wedge \overline{y}))) \vee ((x \wedge \overline{y}) \vee (\overline{x \wedge \overline{z}}))$$

Solution: Given that the value of the inputs are $x = 1$, $y = 1$ and $z = 0$. Now, the value of $(x \wedge \overline{y})$ is 0.

The value of $(y \vee (x \wedge \overline{y}))$ is 1

The value of $(x \wedge (y \vee (x \wedge \overline{y})))$ is 1

Similarly, the value of the $(\overline{x \wedge \overline{z}})$ is 0

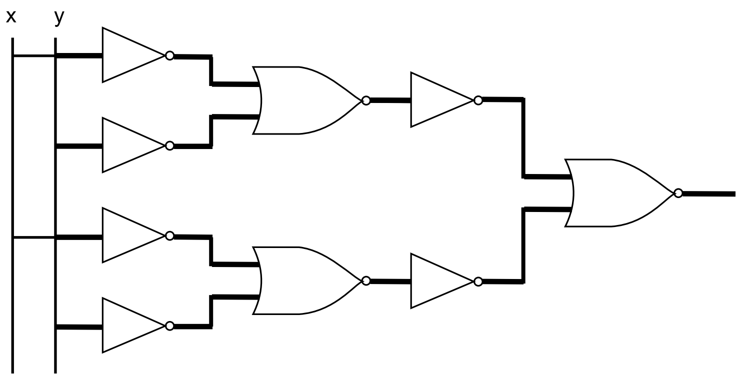
The value of $((x \wedge \overline{y}) \vee (\overline{x \wedge \overline{z}}))$ is 0

So, the value of the Boolean expression

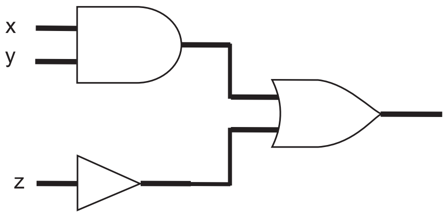
$(x \wedge (y \vee (x \wedge \overline{y}))) \vee ((x \wedge \overline{y}) \vee (\overline{x \wedge \overline{z}}))$ is 1.

Example 8 *Construct an AND gate using inverters and three NOR gates.*

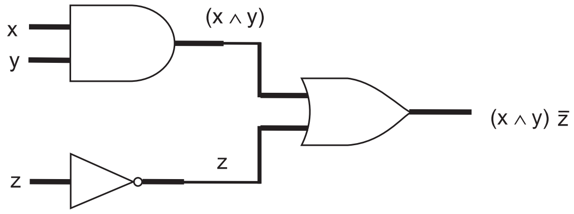
Solution: Output to an AND gate is $(x \wedge y)$ or xy, if the inputs are x and y. The output to a NOR gate is $(\overline{x \vee y})$, if the inputs are x and y. The gating network is given below.



Example 9 *Write the Boolean expression that represents the combinatorial circuit, write the logic table and write the output of each gate symbolically.*



Solution: Given the gating network as below.



The Boolean expression that represents the combinatorial circuit is $((x \wedge y) \vee \bar{z})$. The logic table is given as below.

x	y	z	$(x \wedge y)$	$(x \wedge y) \vee \bar{z}$
1	1	1	1	1
1	1	0	1	1
1	0	1	0	0
0	1	1	0	0
1	0	0	0	1
0	1	0	0	1
0	0	1	0	0
0	0	0	0	1

Example 10 *If $(x + y) = (x + z)$ and $(x' + y) = (x' + z)$, then $y = z$.*

Solution: Given that $(x + y) = (x + z)$ *i.e.,* $(x \vee y) = (x \vee z)$

And $(x' + y) = (x' + z)$ *i.e.,* $(x' \vee y) = (x' \vee z)$

Now,

$$
\begin{aligned}
y &= y \vee 0 && \text{[Identity law]}\\
&= y \vee (x \wedge x') && \text{[Complement law]}\\
&= (y \vee x) \wedge (y \vee x') && \text{[Distributive law]}\\
&= (x \vee y) \wedge (x' \vee y) && \text{[Commutative law]}\\
&= (x \vee z) \wedge (x' \vee \mathrm{z}) && \text{[Given condition]}\\
&= (z \vee x) \wedge (z \vee x') && \text{[Commutative law]}\\
&= z \vee (x \wedge x') && \text{[Distributive law]}\\
&= z \vee 0 && \text{[Complement law]}\\
&= z && \text{[Identity law]}
\end{aligned}
$$

Therefore, $y = z$.

Example 11 *Given the Boolean function f, write f in its disjunctive normal form.*

x	y	z	$f(x, y, z)$
1	1	1	1
1	1	0	1
1	0	1	0
0	1	1	0
1	0	0	0
0	1	0	1
0	0	1	0
0	0	0	1

Solution: From the table given below it is clear that, the output is 1 for the rows 1, 2, 6 and 8. For the first row the combination is $(x \wedge y \wedge z)$. Similarly for rows 2, 6 and 8 the combinations are $(x \wedge y \wedge \bar{z})$, $(\bar{x} \wedge y \wedge \bar{z})$ and $(\bar{x} \wedge \bar{y} \wedge \bar{z})$ respectively.

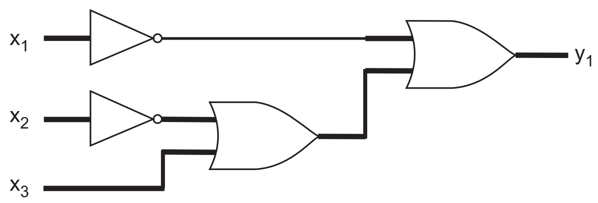
Thus, the disjunctive normal form to the above function f is given as

$$f(x, y, z) = (x \wedge y \wedge z) \vee (x \wedge y \wedge \bar{z}) \vee (\bar{x} \wedge y \wedge \bar{z}) \vee (\bar{x} \wedge \bar{y} \wedge \bar{z})$$

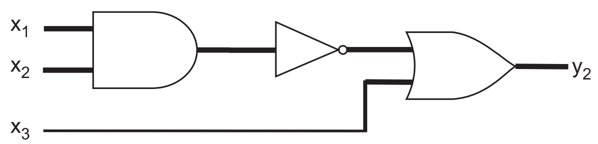
x	y	z	$f(x, y, z)$
1	1	1	1← Row 1
1	1	0	1 ← Row 2
1	0	1	0
0	1	1	0
1	0	0	0
0	1	0	1← Row 6
0	0	1	0
0	0	0	1← Row 8

Example 12 *Show that the combinatorial circuits (a) and (b) are equivalent.*

(*a*)

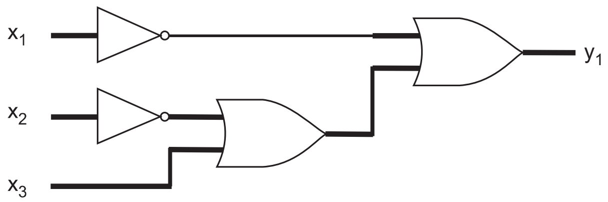


(*b*)

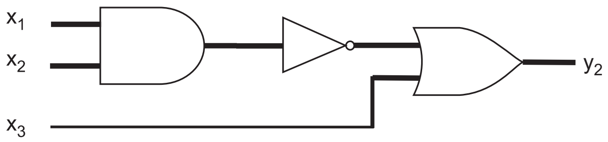


Solution: Given combinatorial circuits are

(*a*)



(*b*)



The output y_1 for combinatorial circuit (*a*) is given as

$$y_1 = \bar{x}_1 \vee (\bar{x}_2 \vee x_3) = (\bar{x}_1 \vee \bar{x}_2) \vee x_3 = \left(\overline{x_1 \wedge x_2}\right) \vee x_3$$

The output y_2 for combinatorial circuit (*b*) is given as $y_2 = \left(\overline{x_1 \wedge x_2}\right) \vee x_3$. Hence, the combinatorial circuits (*a*) and (*b*) are equivalent.

Example 13 *Reduce the following Boolean products to either 0 or a fundamental product.*

(*a*) $x\,y\,x'z$ (*b*) $x\,y\,z'y\,x$

Solution: (*a*)

$$
\begin{aligned}
x\,y\,x'z &= x\,x'\,y\,z && \text{[Commutative law]}\\
&= 0\,y\,z && \text{[Complement law]}\\
&= 0 && \text{[Bound law]}
\end{aligned}
$$

i.e., $x\,y\,x'z = 0$

(*b*)

$$
\begin{aligned}
x\,y\,z'y\,x &= x\,y\,y\,z'\,x && \text{[Commutative law]}\\
&= x\,y\,z'\,x && \text{[Idempotent law]}\\
&= x\,y\,x\,z' && \text{[Commutative law]}\\
&= x\,x\,y\,z' && \text{[Commutative law]}\\
&= x\,y\,z' && \text{[Idempotent law]}
\end{aligned}
$$

i.e., $x\,y\,z'y\,x = x\,y\,z'$

Example 14 *Given the Boolean function f, write f in its conjunctive normal form.*

x	y	z	$f(x, y, z)$
1	1	1	1
1	1	0	1
1	0	1	0
0	1	1	0
1	0	0	0
0	1	0	1
0	0	1	0
0	0	0	1

Solution: Given the Boolean function f as below.

x	y	z	$f(x, y, z)$
1	1	1	1
1	1	0	1
1	0	1	0 ← Row 3
0	1	1	0 ← Row 4

Contd...

1	0	0	0 ← Row 5
0	1	0	1
0	0	1	0 ← Row 7
0	0	0	1

From the table it is clear that, the output is 0 for the rows 3, 4, 5 and 7. For the third row the combination is $(\bar{x} \vee y \vee \bar{z})$. Similarly for rows 4, 5 and 7 the combinations are $(x \vee \bar{y} \vee \bar{z})$, $(\bar{x} \vee y \vee z)$ and $(x \vee y \vee \bar{z})$ respectively.

Thus, the conjunctive normal form to the above function is given as

$$f(x, y, z) = (\bar{x} \vee y \vee \bar{z}) \wedge (x \vee \bar{y} \vee \bar{z}) \wedge (\bar{x} \vee y \vee z) \wedge (x \vee y \vee \bar{z}).$$

Example 15 *Design a combinatorial circuit that computes exclusive OR; XOR of x and y.*

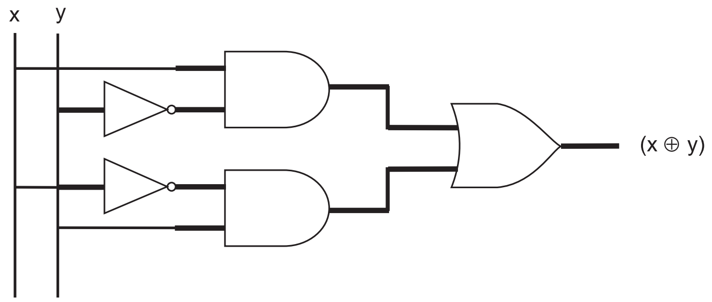
Solution: Let the inputs to the XOR gate be x and y. The logic table for XOR gate is given below.

x	y	$x \oplus y$
1	1	0
1	0	1
0	1	1
0	0	0

So, the disjunctive normal form of this function is given as

$$x \oplus y = (x \wedge y') \vee (x' \wedge y)$$

The combinatorial circuit corresponding to $(x \oplus y)$ is given below.



Example 16 *Find the disjunctive and conjunctive normal form of the given function and draw the combinatorial circuit corresponding to the disjunctive normal form.*

x	y	z	$f(x, y, z)$
1	1	1	0
1	1	0	0
1	0	1	0
0	1	1	1
1	0	0	1
0	1	0	1
0	0	1	1
0	0	0	0

Solution: Given Boolean function is

x	y	z	$f(x, y, z)$
1	1	1	0
1	1	0	0
1	0	1	0
1	0	0	1← Row 4
0	1	1	1← Row 5
0	1	0	1← Row 6
0	0	1	1← Row 7
0	0	0	0

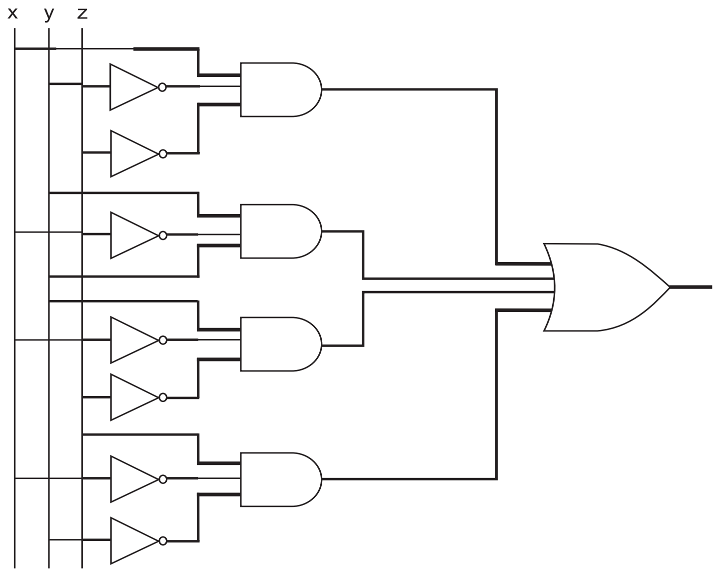
From the table it is clear that the output is 1 for rows 4, 5, 6 and 7. For the fourth row the combination is $(x \wedge y' \wedge z')$. Similarly the combinations $(x' \wedge y \wedge z)$, $(x' \wedge y \wedge z')$, and $(x' \wedge y' \wedge z)$ are for rows 5, 6 and 7 respectively. So, the disjunctive normal form to the above function is given as

$$f(x, y, z) = (x \wedge y' \wedge z') \vee (x' \wedge y \wedge z) \vee (x' \wedge y \wedge z') \vee (x' \wedge y' \wedge z).$$

Similarly, corresponding to the output 0 for rows 1, 2, 3 and 8, the conjunctive normal form to the above function is given as

$$f(x, y, z) = (x' \vee y' \vee z') \wedge (x' \vee y' \vee z) \wedge (x' \vee y \vee z') \wedge (x \vee y \vee z).$$

The combinatorial circuit corresponding to the disjunctive normal form is given below.



Example 17 *Find the disjunctive normal form of the function using algebraic technique.*

$$f(x, y) = (x \vee y) \wedge (x' \vee y')$$

Solution:

$$f(x, y) = (x \vee y) \wedge (x' \vee y')$$

$= (x \wedge (x' \vee y')) \vee (y \wedge (x' \vee y'))$ [Distributive law]

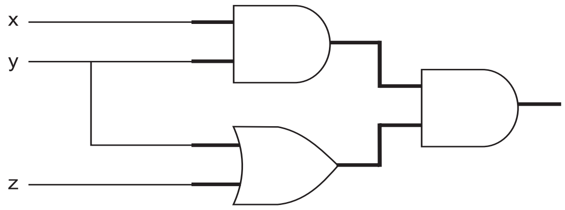
$= (x \wedge x') \vee (x \wedge y') \vee (y \wedge x') \vee (y \wedge y')$ [Distributive law]

$= 0 \vee (x \wedge y') \vee (y \wedge x') \vee 0$ [Complement law]

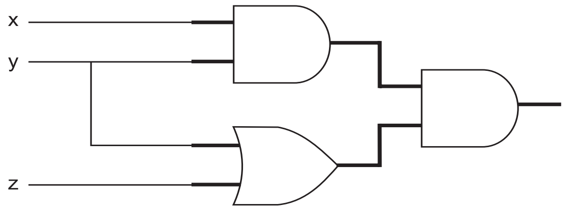
$= (x \wedge y') \vee (y \wedge x')$ [Identity law] *i.e.*, $f(x, y) = (x \wedge y') \vee (y \wedge x')$

Which is the disjunctive normal form of the function $f(x, y)$.

Example 18 *Find the disjunctive normal form for the following combinatorial circuit.*



Solution: Given that the combinatorial circuit as



The output of the above combinatorial circuit is given as $f(x, y, z) = (x \wedge y) \wedge (y \vee z)$. The logic table for the above expression is given below. From the table it is clear that the function has output 1 for rows 1 and 2. For the first row the combination is $(x \wedge y \wedge z)$ whereas for second row the combination is $(x \wedge y \wedge z')$. Thus, the disjunctive normal form for the above function is given as

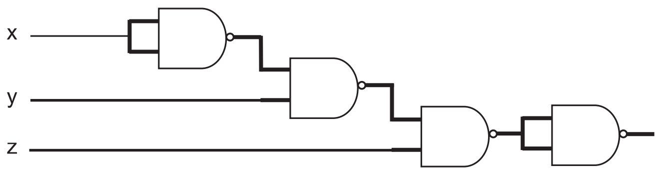
$$f(x, y, z) = (x \wedge y \wedge z) \vee (x \wedge y \wedge z')$$

x	y	z	$(x \wedge y)$	$(y \vee z)$	$(x \wedge y) \wedge (y \vee z)$
1	1	1	1	1	1
1	1	0	1	1	1
1	0	1	0	1	0
0	1	1	0	1	0
1	0	0	0	0	0
0	1	0	0	1	0
0	0	1	0	1	0
0	0	0	0	0	0

EXERCISES

1. Find the disjunctive normal form of each function using algebraic technique.
   (*a*) $f(x, y) = x \vee (x \wedge y)$
   (*b*) $f(x, y, z) = x \vee y \wedge (x \vee z')$
   (*c*) $f(x, y, z) = x \vee (y' \vee (x\,y' \vee x\,z'))$
2. Reduce the following Boolean products to either 0 or a fundamental product.
   (*a*) $x\,y\,z\,y$ (*b*) $x\,y\,z'\,y\,x'\,z'$ (*c*) $x\,y'\,z\,x\,y'$ (*d*) $x\,y\,z'\,t\,y'\,t$
   (*e*) $x\,y'\,x\,z'\,t\,y'$

3. Write the logic table for the circuit given below.

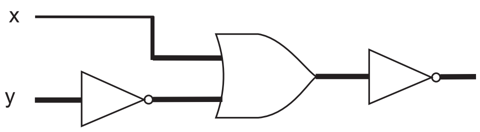


4. Find the disjunctive normal form of a Boolean expression having a logic table the same as the given table and draw the combinatorial circuit corresponding to the disjunctive normal form.

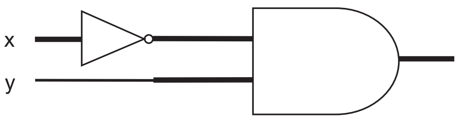
x	y	z	f
1	1	1	1
1	1	0	1
1	0	1	0
0	1	1	0
1	0	0	1
0	1	0	0
0	0	1	1
0	0	0	0

5. Are the combinatorial circuits equivalent? Explain.

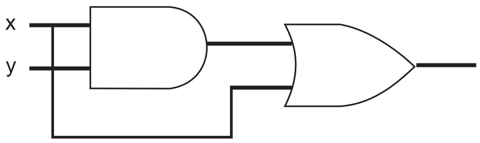
(*i*) (*a*)



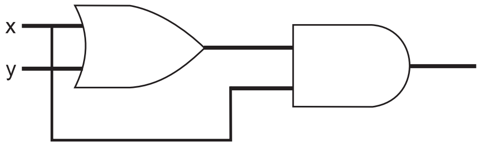
(*b*)



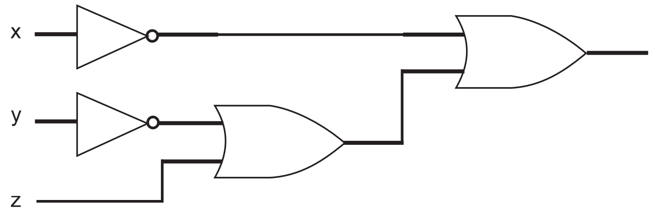
(*ii*) (*a*)



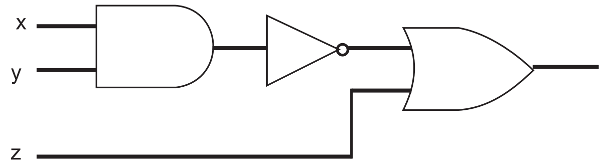
(*b*)



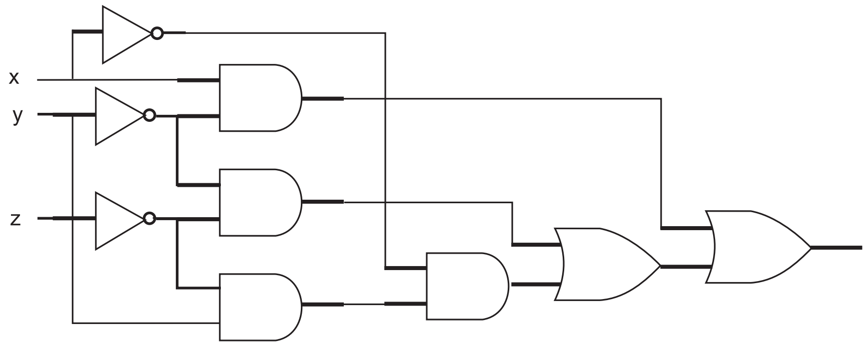
(*iii*) (*a*)



(*b*)



6. Find the Boolean expression in disjunctive normal form for the circuit given below.



7. Find the disjunctive normal form of each function corresponding to the logic tables given below.

(*a*)

x	y	$f(x, y)$
1	1	1
1	0	0
0	1	1
0	0	1

(*b*)

x	y	$f(x, y)$
1	1	0
1	0	1
0	1	0
0	0	1

(*c*)

x	y	z	$f(x, y, z)$
1	1	1	1
1	1	0	0
1	0	1	1
0	1	1	1
1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

(*d*)

x	y	z	$f(x, y, z)$
1	1	1	0
1	1	0	0
1	0	1	1
0	1	1	0
1	0	0	1
0	1	0	1
0	0	1	1
0	0	0	1

8. Find the conjunctive normal form of each function given in question 7.

9. Draw the logic circuit (Combinatorial circuit) with inputs x, y, z and output Y which corresponds to each Boolean expression.

(*i*) $Y = x'y\,z + x'y\,z' + x\,y\,z'$

(*ii*) $Y = x\,y'\,z + x\,z' + y'\,z$

10. Construct a combinatorial circuit that represents the following Boolean function.

x	y	z	$f(x, y, z)$
1	1	1	0
1	1	0	1
1	0	1	1
0	1	1	1
1	0	0	1
0	1	0	1
0	0	1	1
0	0	0	0

11. Write the dual of each Boolean equation.

(*a*) $(a \wedge 1) \vee (0 \vee a') = 0$

(*b*) $a \wedge (a' \vee b) = a \wedge b$

(*c*) $a \vee (a' \wedge b) = a \vee b$

(*d*) $(a \vee 1) \wedge (a \vee 0) = a$

(*e*) $(a \wedge a') \vee (a \wedge 0) = a$

(*f*) $(a \vee b) \wedge (b \vee c) = (a \wedge c) \vee b$

[**Hint:** To obtain the dual equation, interchange $\vee$ and $\wedge$, and interchange 0 and 1]

12. Discuss a XOR gate with four inputs x, y, z and t.

13. Express the following Boolean expression $f(x, y, z)$ as a sum of products and then in its complete sum- of- products form.

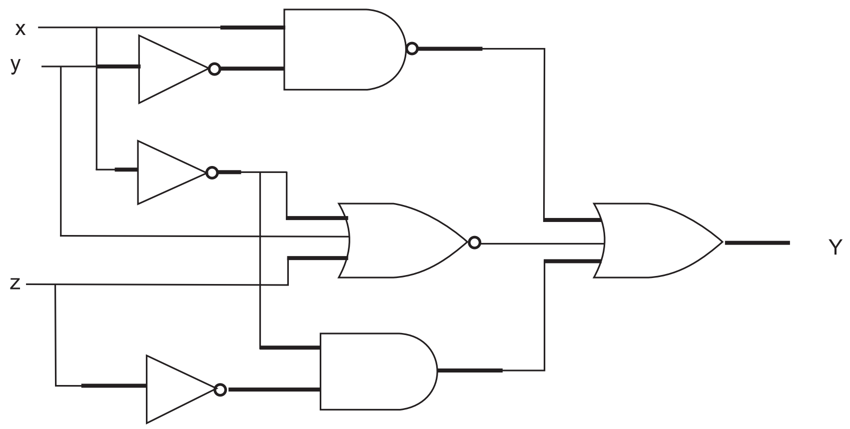
(*a*) $f(x, y, z) = x\,(x\,y' + x'y + y'\,z)$

(*b*) $f(x, y, z) = (x' + y)' + y'\,z$

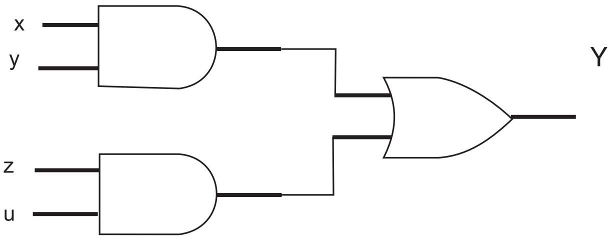
(*c*) $f(x, y, z) = (x + y'\,z)\,(y + z')$

14. Express the output Y as a Boolean expression in the inputs x, y, z, t and u for the logic circuits given below.

(*a*)



(*b*)



(*c*)