Unit 2

Digital Signals:

Digital signal is a signal that is being used to represent data as a sequence of discrete values. In most digital circuits, the signal can have two possible valid values; this is called a **binary signal** or **logic signal**. They are represented by two voltage bands: one near a reference value (typically termed as *ground* or zero volts), and the other a value near the supply voltage. These correspond to the two values "zero" and "one" (or "false" and "true").

Example of digital wave form: 0101100100

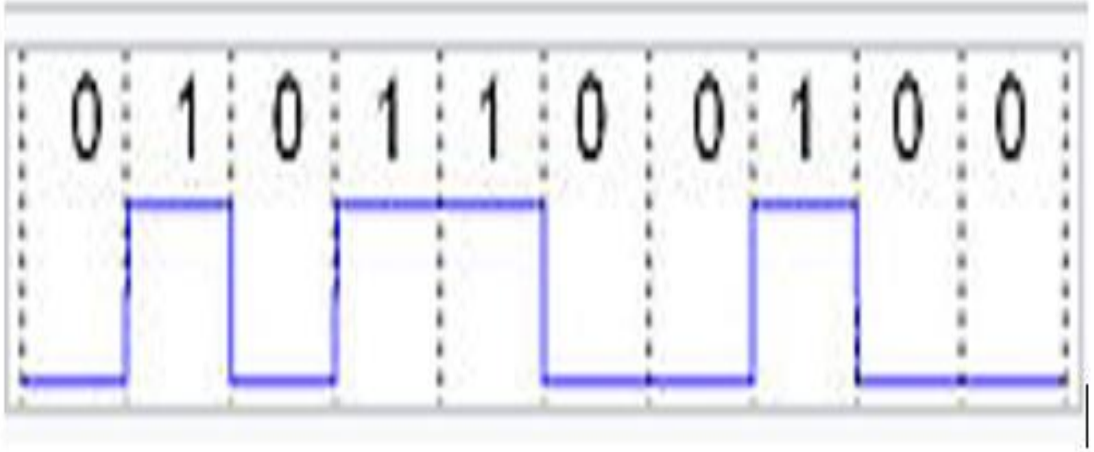


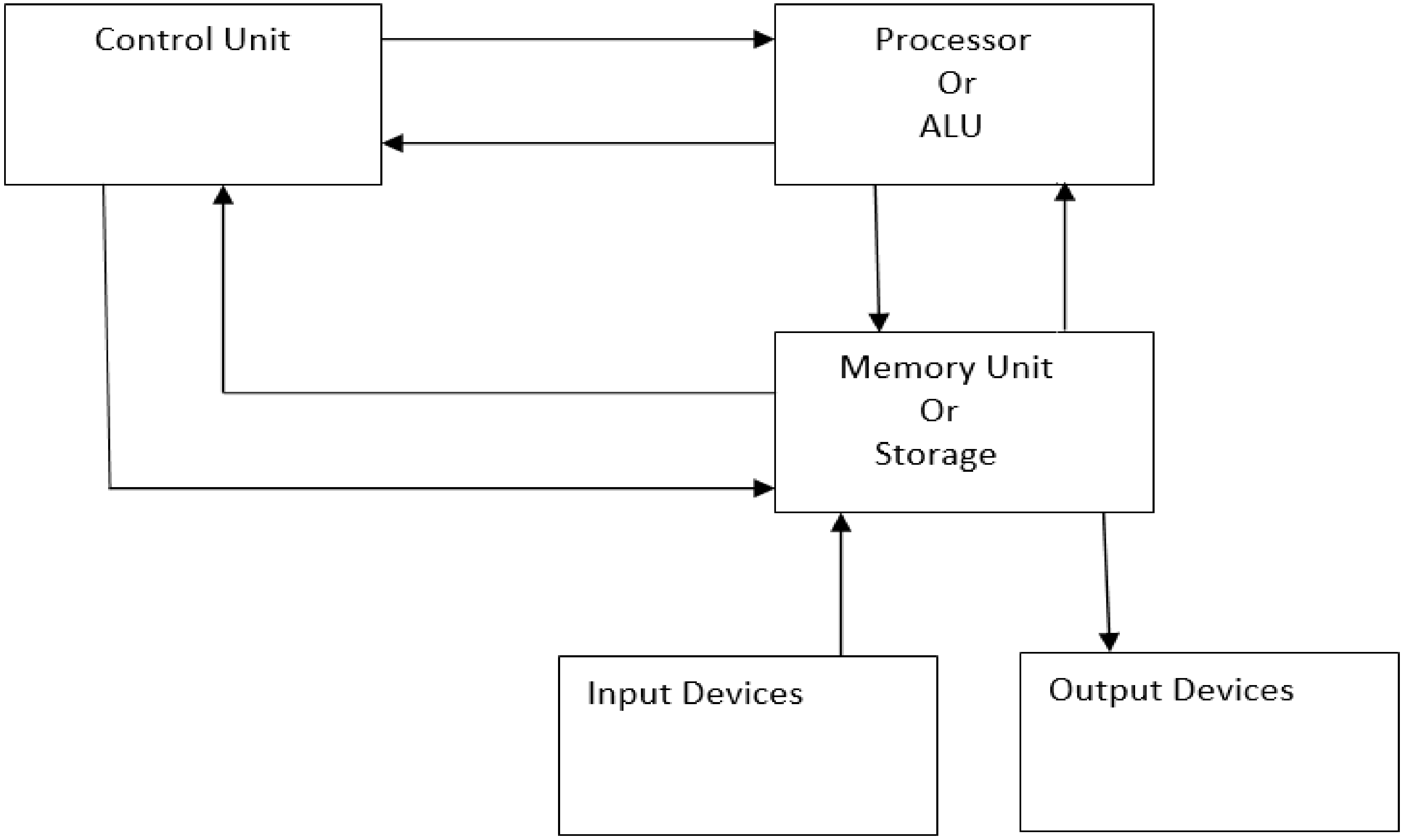
Fig: Digital Wave Form

Analog signal: Analog signal represents continuous values; at any given time it represents a real number within a continuous range of values.

Digital System: Digital system is data technology that uses discrete (discontinuous) values. This system operates on binary digits (signals) 0 and 1. Working of digital computers, communication systems, calculators etc. are based on digital techniques and these systems are known as digital system.

Analog System: An analog system is a system that use analog i.e. continuous range of values to present information. Like voltmeter, automobile speedometer are the example of analog system.

Block diagram of digital computer



Functions or working principle of digital computer:

1. Memory unit stores programs as well as input and output.
2. The control unit supervises the flow of information between various units and retrieves the instructions stored in memory unit.
3. After getting control signal from control unit, memory unit sends the data to the processor.
4. For each instruction, control unit informs the processor to execute the operation according to the instructions.
5. After getting control signal processor sends the processed information to memory unit sends that information to the output unit.

Advantages of digital system:

1. In case of digital system large numbers of ICs are available for performing various operations, hence digital systems are highly reliable, accurate, small in size and speed of operation is very high.
2. Computer controlled digital systems can be controlled by software that allows new functions to be added without changing hardware.

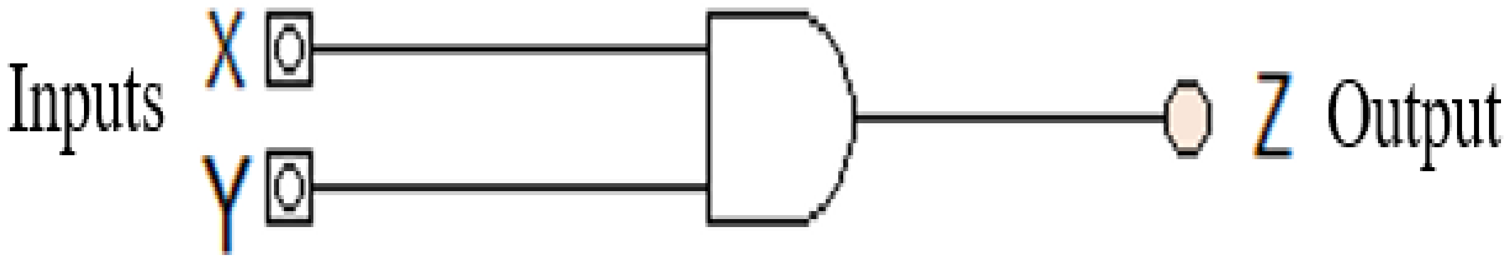
Disadvantages of digital system:

1. It is difficult to install digital system, because it requires many more complex electronic circuits and ICs.
2. In digital systems, if a single piece of digital data lost, large blocks of related data can completely change.

Logic gates: Logic gates are the basic building blocks of digital computer. Logic gates have one or more than one input and only one output. Input and output values are the logical values true (1) and false (0). Logic gates are also called combinational logic circuits. Basic logic gates are AND, OR and NOT.

1. **AND gate:** The output of AND gate is 1 if and only if all of the inputs are 1, otherwise the output value is 0.

Logic diagram of AND gate:

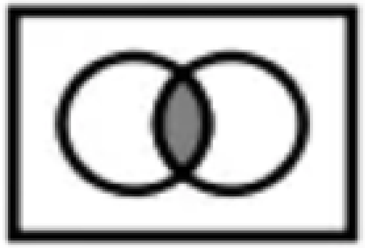


Algebraic Function:

Z = X AND Y

= X.Y

Venn diagram:



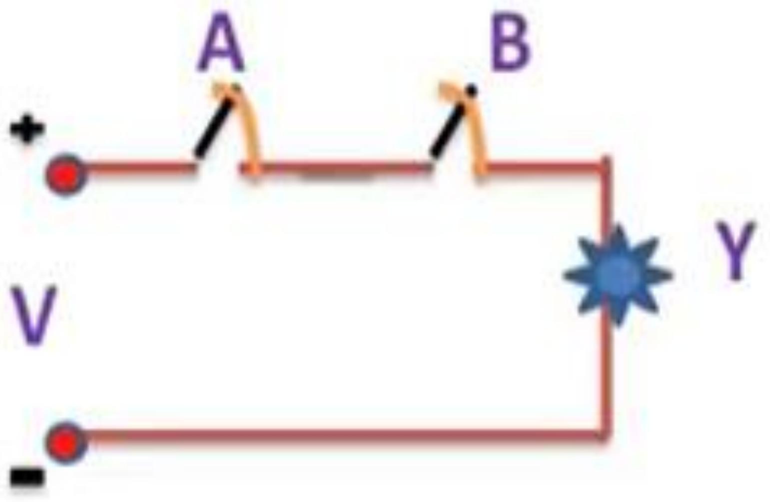
Truth Table:

2^n combination of variables. i.e. $2^2 = 4$ binary Combinations from 0 to 3.

Inputs		**Outputs**
X	**Y**	**Z**
0	0	0
0	1	0
1	0	0
1	1	1

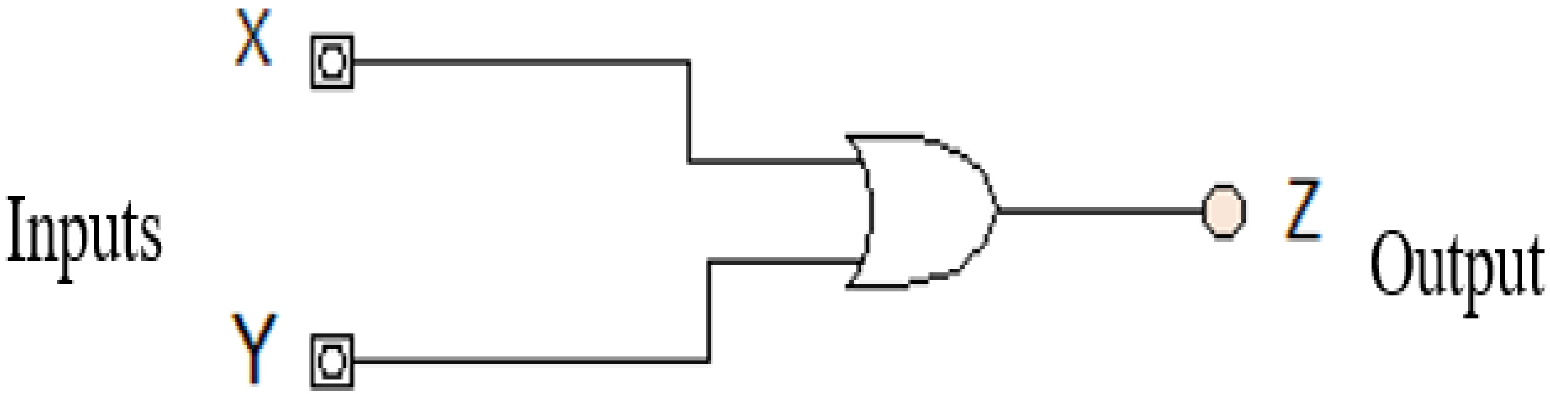
Switching Circuit of AND Gate:

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1



2. OR gate: The output of OR gate is 1 if any one of the input value is 1 and output is 0 if all the inputs are 0.

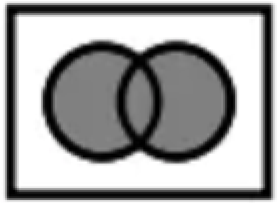
Logic diagram of OR gate:



Algebraic Function:

Z = X OR Y   or X + Y

Venn diagram:



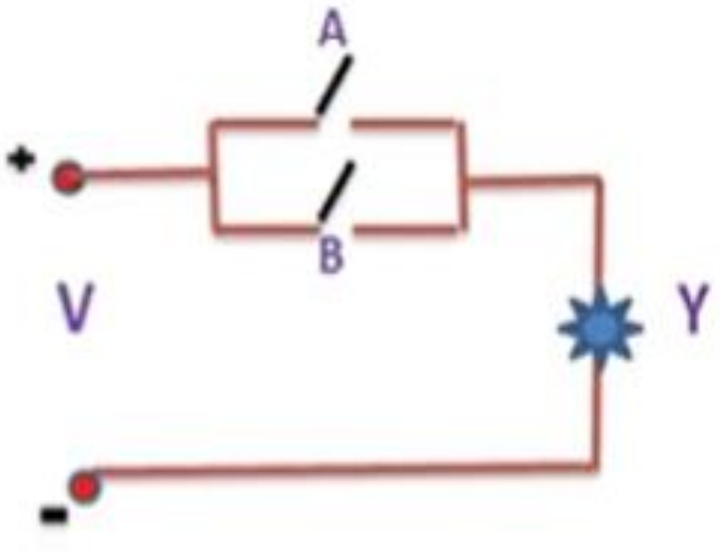
Truth Table:

2^n combination of variables. i.e. $2^2 = 4$ binary Combinations from 0 to 3.

Inputs		Outputs
X	**Y**	**Z**
0	0	0
0	1	1
1	0	1
1	1	1

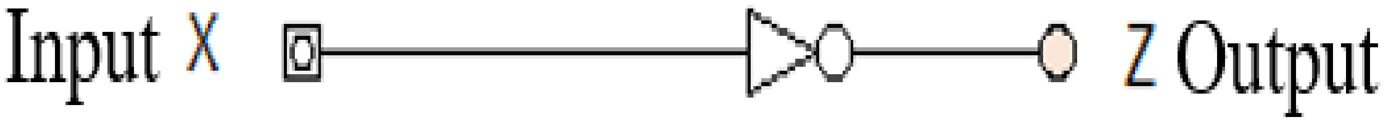
Switching Circuit of OR Gate:

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1



3. NOT gate (inverter): A NOT has only one input and one output the output of NOT gate is the inversion of the input i.e. complement of the input.

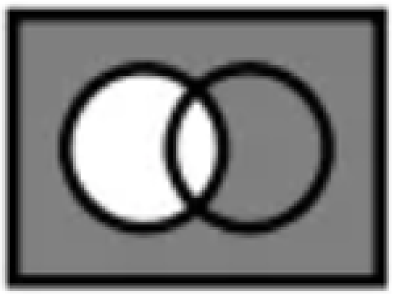
Logic diagram of NOT gate:



Algebraic Function:

$$Z = \text{NOT } X$$
$$= \bar{X}$$

Venn diagram:

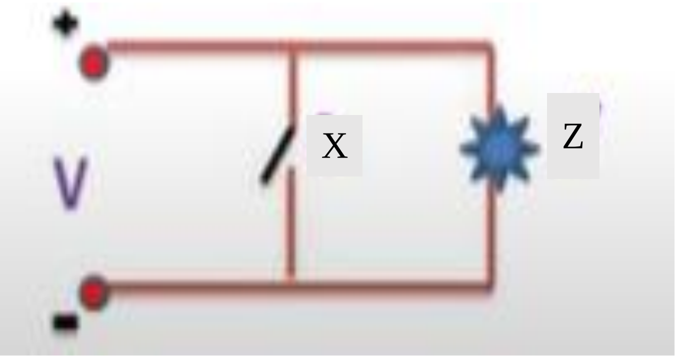


Truth Table:

Input	Output
X	Z
0	1
1	0

Switching Circuit of NOT Gate:

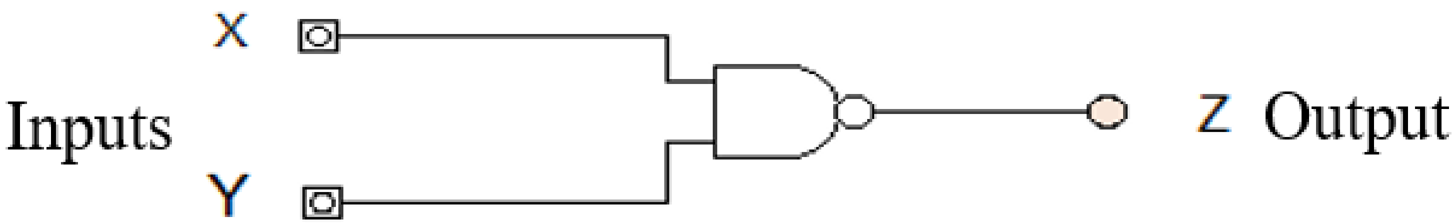
Input	Output
X	Z
0	1
1	0



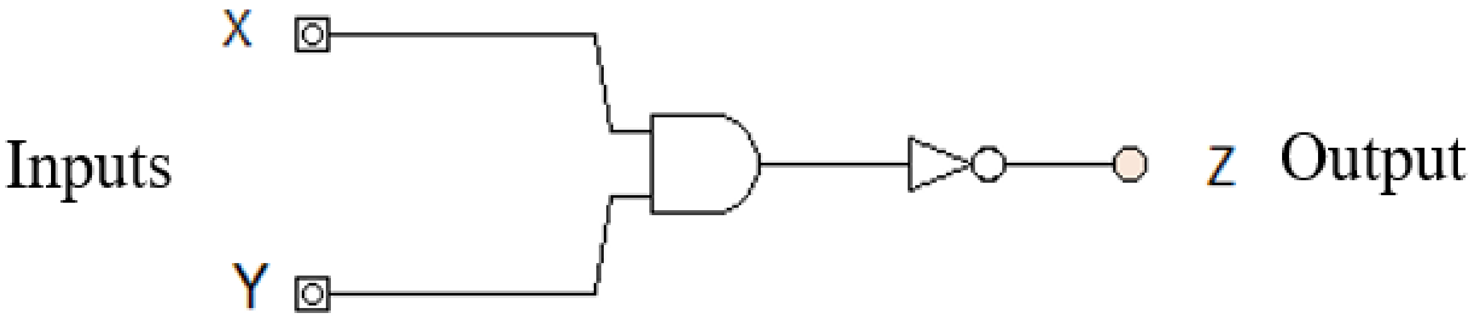
Combined Gates:

NAND gate: NAND gate is the combination of AND and NOT gates. An AND gate with inverter at the output. The output of NAND gate is the complement of AND gate.

Logic Diagram of NAND gate:



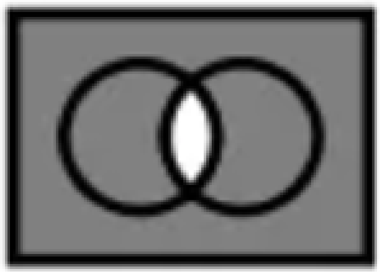
Or



Algebraic Function:

$$Z = \overline{X \text{ AND } Y}$$
$$= (X.Y)'$$

Venn diagram:



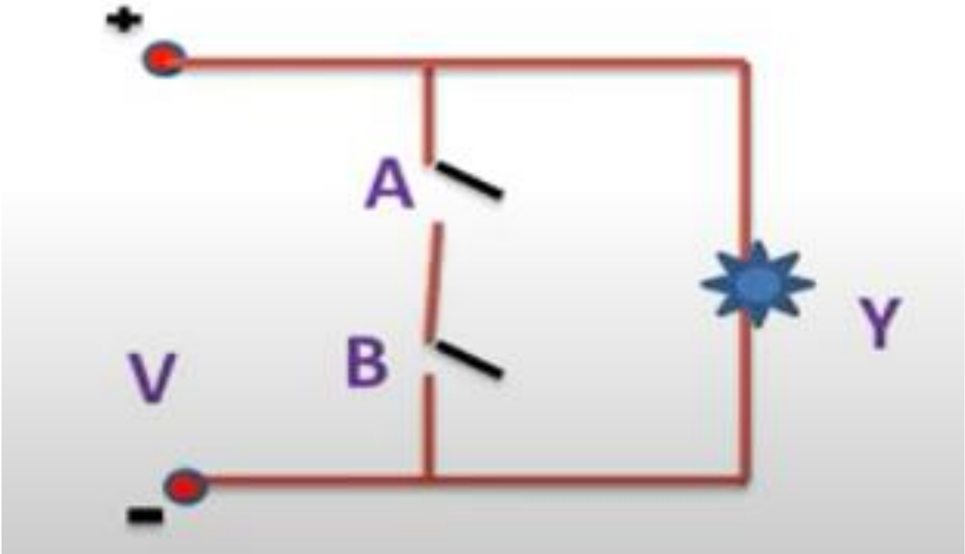
Truth Table:

2^n combination of variables. i.e. $2^2 = 4$ binary Combinations from 0 to 3.

Inputs		**Outputs**
X	**Y**	**Z**
0	0	1
0	1	1
1	0	1
1	1	0

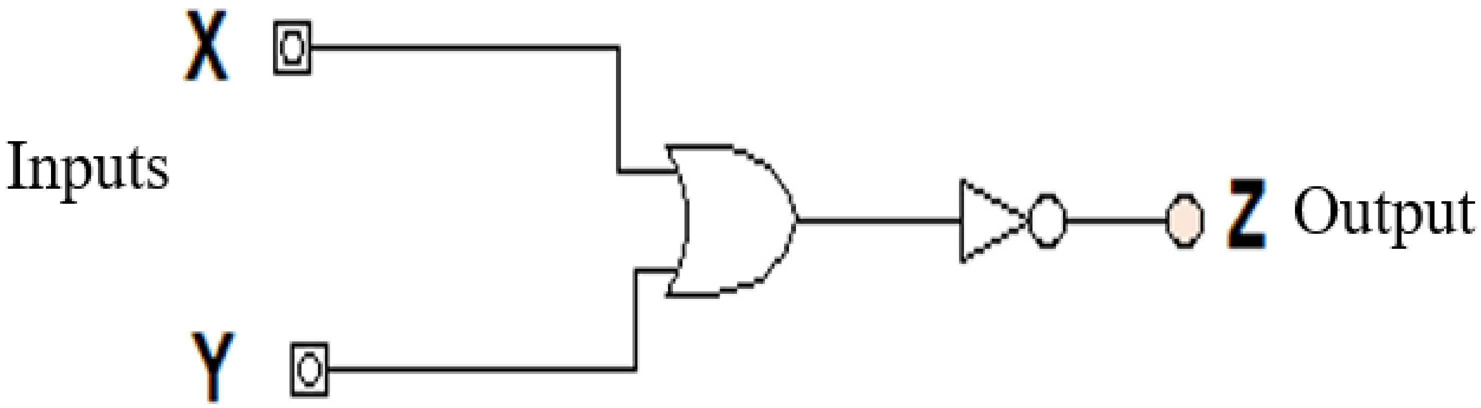
Switching Circuit of NAND Gate:

A	B	$Y = (A.B)'$
0	0	1
0	1	1
1	0	1
1	1	0

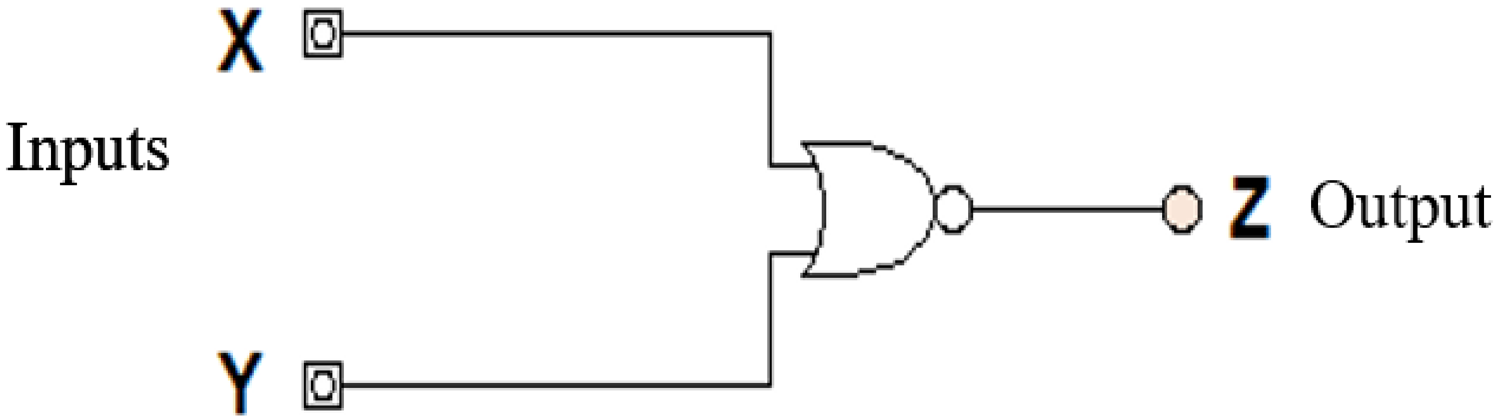


NOR Gate: NOR gate is the combination of OR and NOT gate. The output of NOR gate is the complement of OR gate.

Logic Diagram of NAND gate:



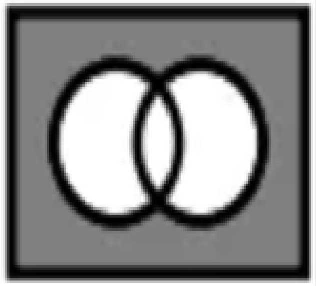
Or



Algebraic Function:

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

Venn diagram:

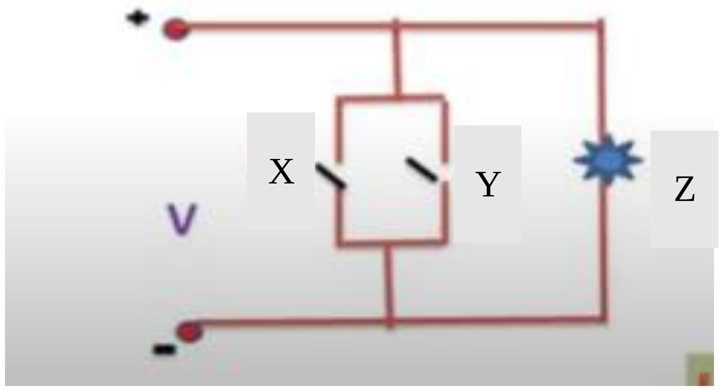


Truth Table:
2^n combination of variables. i.e. $2^2 = 4$ binary Combinations from 0 to 3.

Inputs		Outputs
X	**Y**	**Z**
0	0	1
0	1	0
1	0	0
1	1	0

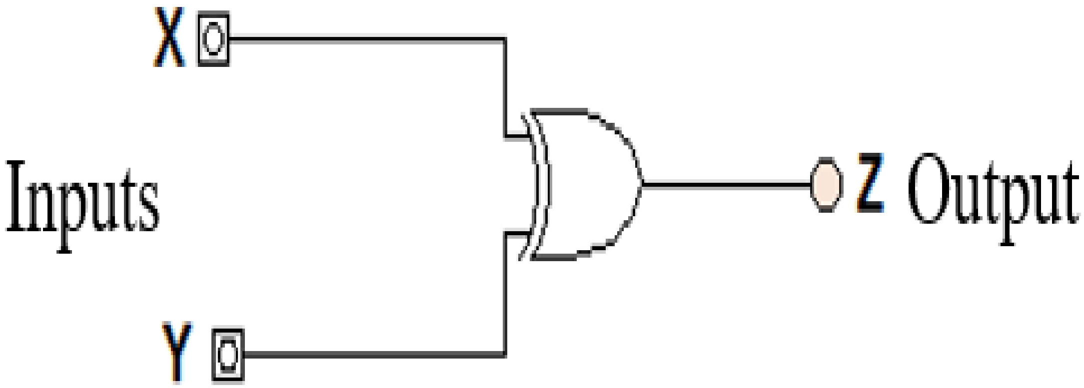
Switching Circuit of NOR Gate:

Inputs		Outputs
X	**Y**	**Z**
0	0	1
0	1	0
1	0	0
1	1	0



Exclusive – OR (XOR) Gate: An XOR gate gives an output value 1 when there are different input values and the output value is low when there are same input values.

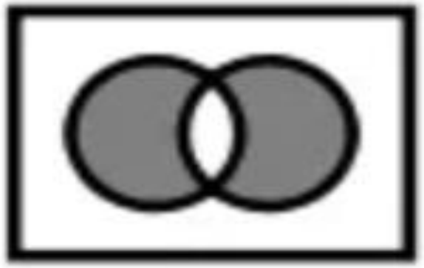
Logic Diagram of XOR gate:



Algebraic Function:

$$Z = X\bar{Y} + \bar{X}Y$$

Venn diagram:



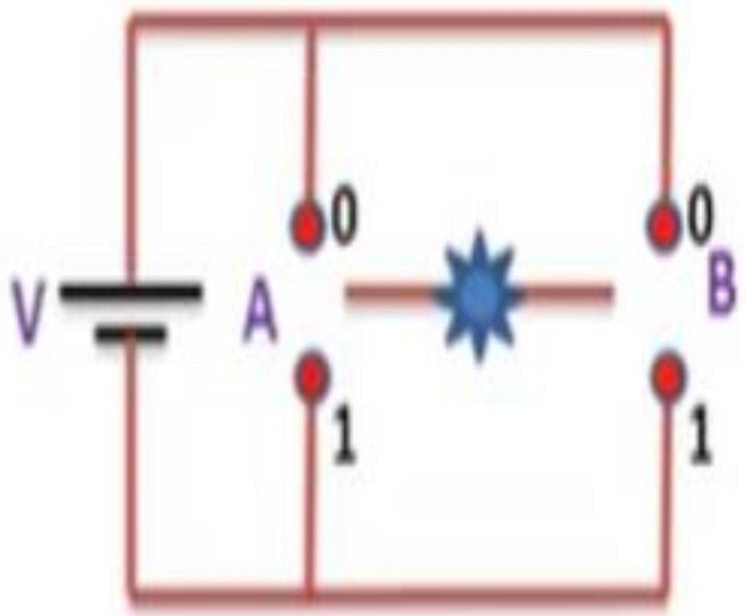
Truth Table:

Inputs		**Outputs**
X	**Y**	**Z**
0	0	0
0	1	1
1	0	1
1	1	0

Switching Circuit of XOR Gate:

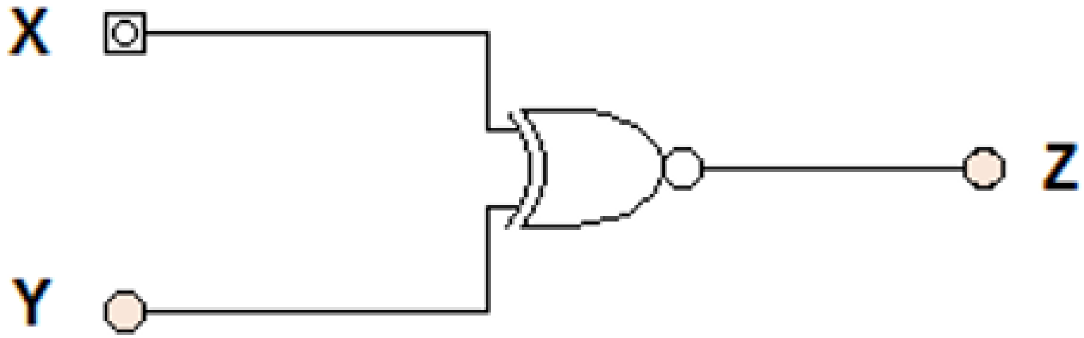
A	B	Y = (A'B+B'A)
0	0	0
0	1	1
1	0	1
1	1	0

odd bit detector



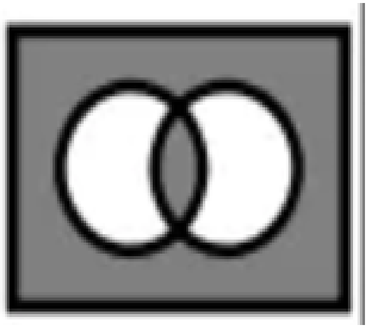
Exclusive – NOR (X-NOR) Gate: An X-NOR gate gives an output value 1 when there are same input values and the output value is low when there are different input values.

Logic Diagram of XNOR gate:



Algebraic Function:

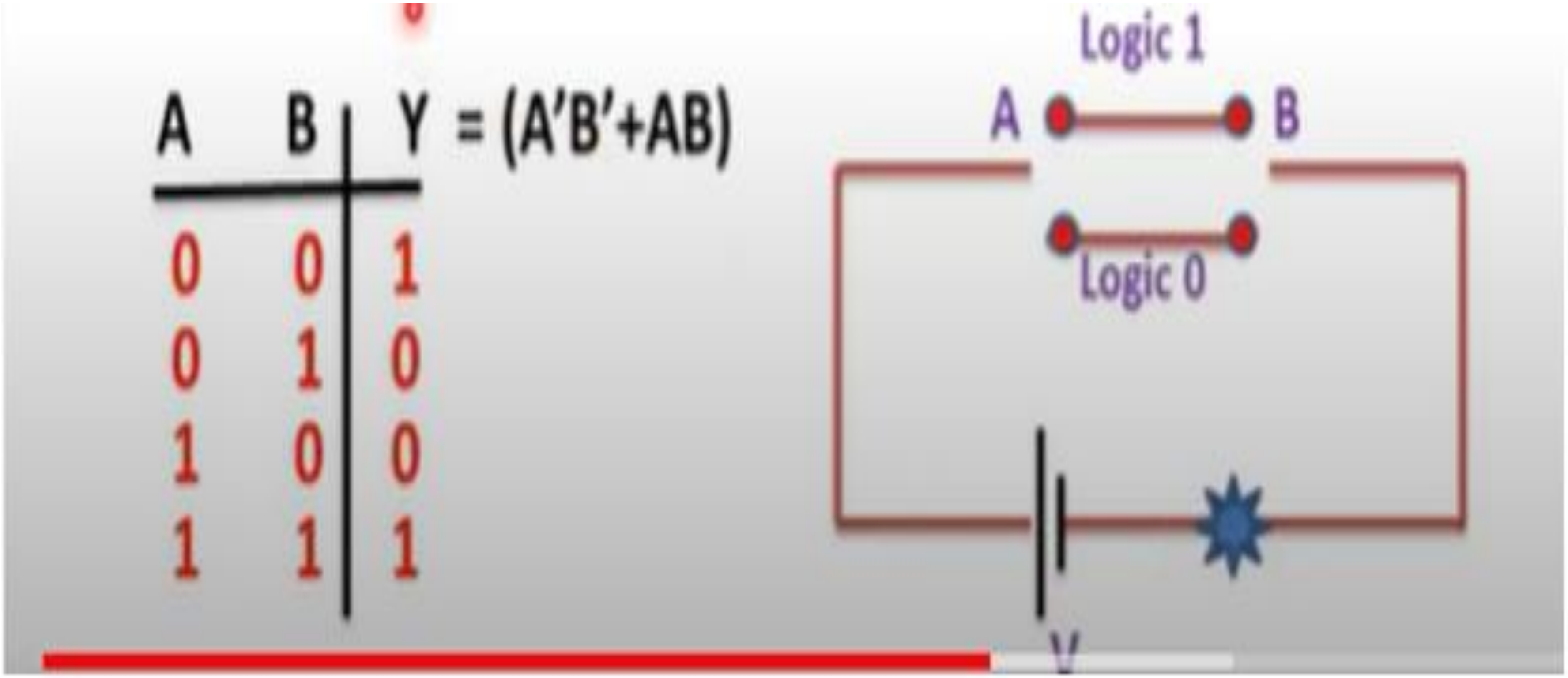
$Z = XY + \bar{X}\bar{Y}$



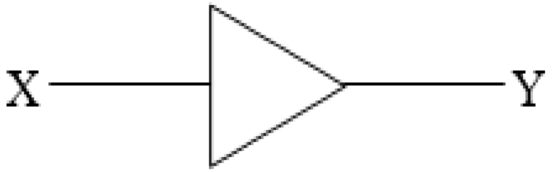
Truth Table:

Inputs		Outputs
X	**Y**	**Z**
0	0	1
0	1	0
1	0	0
1	1	1

Switching Circuit of XNOR Gate:



Buffer: Buffer produces the transfer function but does not produce any particular logic operation, since the binary value of the output is equal to the binary value of the input. Buffer will delay the time between input and output. It can also amplify the signal if the current is too weak.

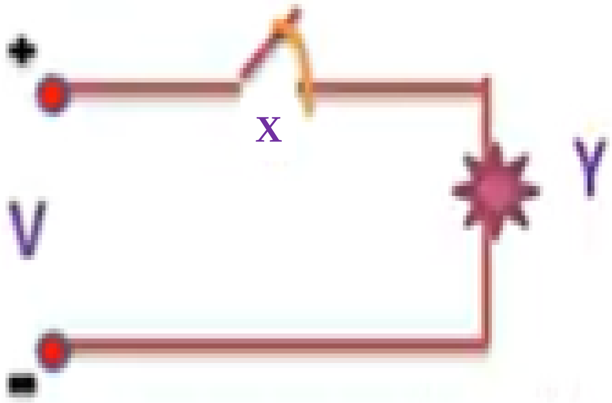


Logic function: $Y = X$

Truth Table:

Input	Output
X	Y
0	0
1	1

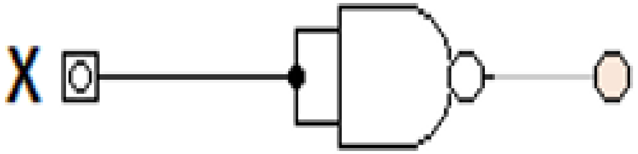
Switching Circuit of Buffer:



Universal Gates: NAND and NOR gates are called universal gates because, we can build any gate using NAND or NOR gates. Basic logic gates AND, OR and NOT can be realized by using only NAND or NOR gates.

NAND as a universal gate:

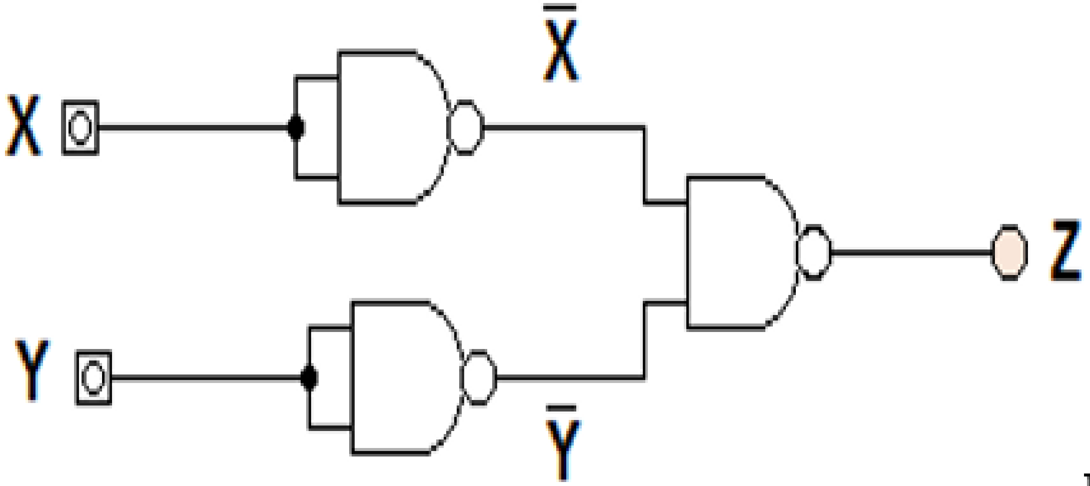
i) **NOT gate:** NOT gate can be realized using single input NAND gate.



$Y = \overline{X}$

Properties of NOT gate.

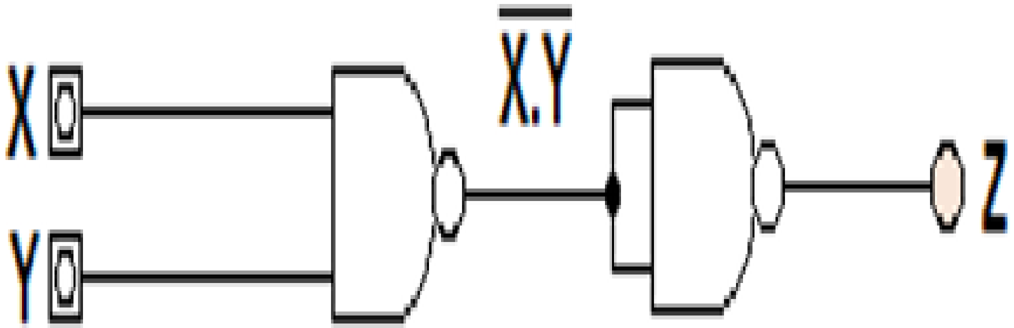
ii) **OR gate:** OR gate can be realized using three NAND gates.



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

Properties of OR gate.

iii) **AND gate:** AND gate can be realized using two NAND gates.

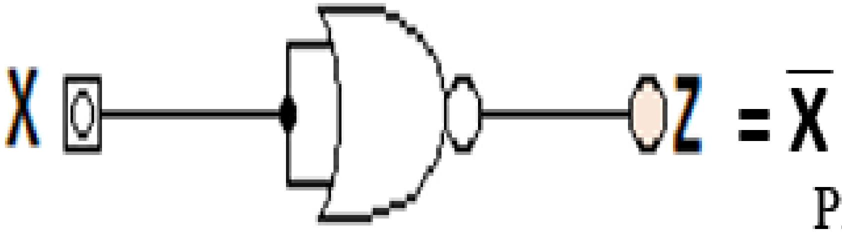


$$= \overline{\overline{X.Y}} = \overline{\overline{X}+\overline{Y}} = \overline{\overline{X}}\,.\,\overline{\overline{Y}} = X\,.\,Y$$

Properties of AND gate.

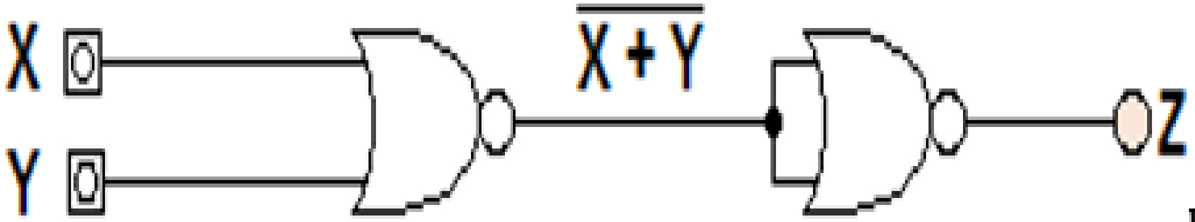
NOR as a universal gate:

i) **NOT gate:** NOT gate can be realized using a single input NOR gate.



Properties of NOT gate.

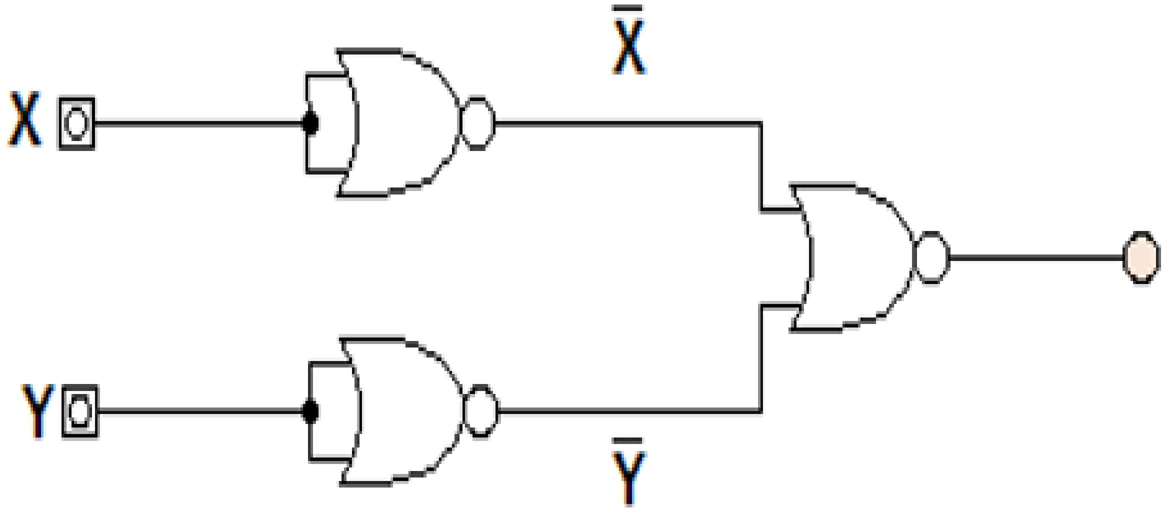
ii) **OR gate:** OR gate can be realized using two NOR gates.



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

Properties of OR gate.

iii) AND gate: AND gate can be realized using three NOR gates.

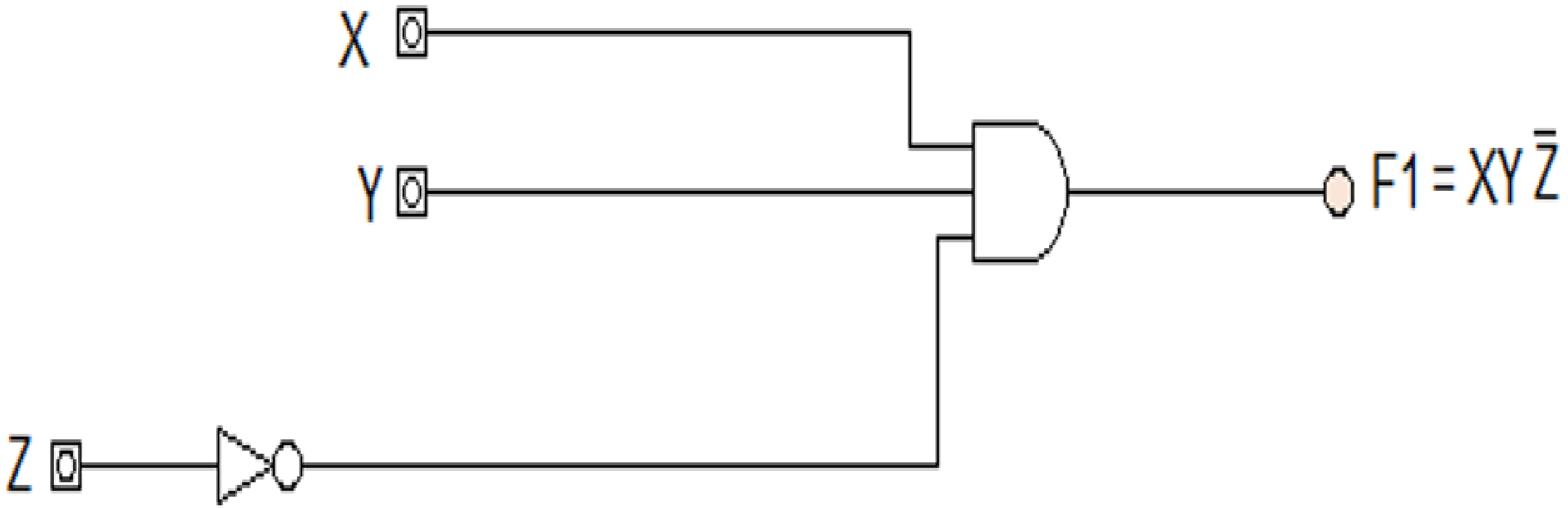


$$Z = \overline{\overline{X} + \overline{Y}} = \overline{\overline{X}}\,.\,\overline{\overline{Y}} = X\,.\,Y$$

Properties of AND gate.

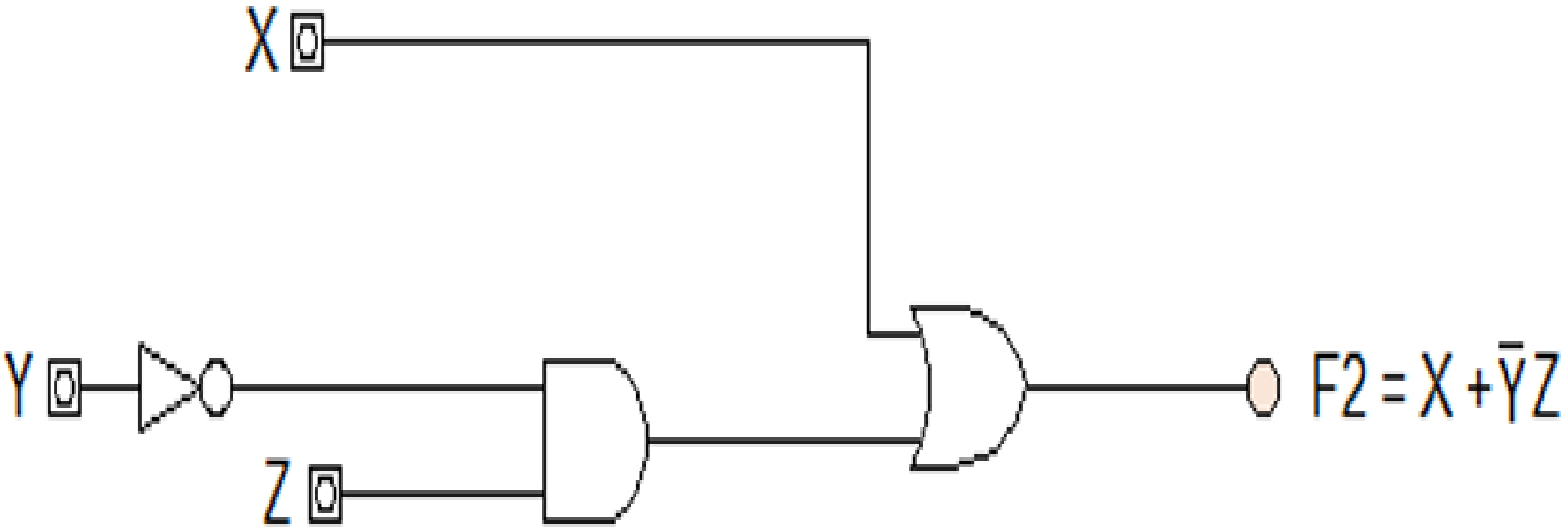
Implementation/realization of Boolean function with logic gates and truth table:

i) $F1 = XY\bar{Z}$



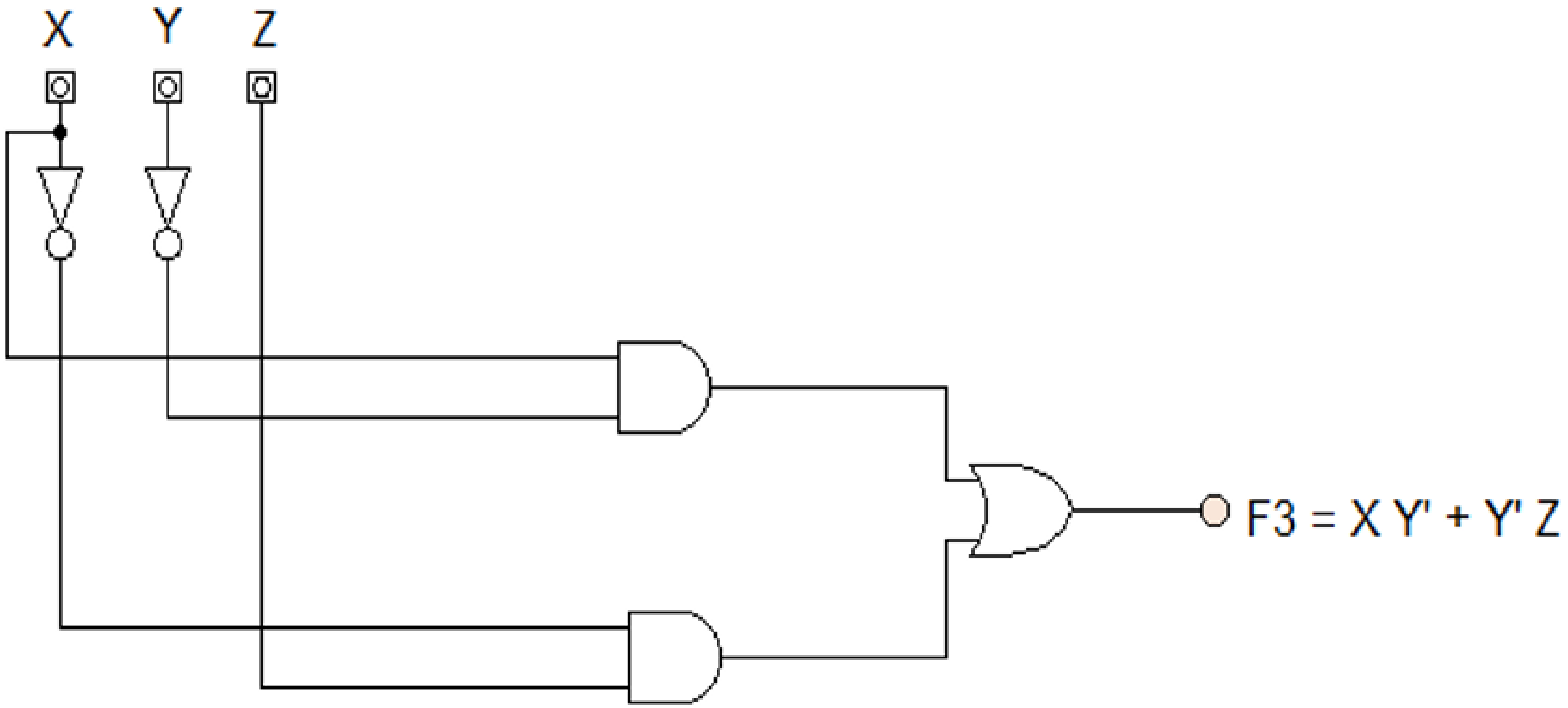
Implementation/realization of Boolean function with logic gates and truth table:

ii) $F2 = X + \bar{Y}Z$



Implementation/realization of Boolean function with logic gates and truth table:

iii) $F3 = X\bar{Y} + \bar{X}Z$



Implementation/realization of Boolean function with logic gates and truth table:

Truth table for F1, F2 and F3:

X	Y	Z	$F1 = XY\bar{Z}$	$F2 = X + \bar{Y}Z$	$F3 = X\bar{Y} + \bar{X}Z$
0	0	0	0	0	0
0	0	1	0	1	1
0	1	0	0	0	0
0	1	1	0	0	1
1	0	0	0	1	1
1	0	1	0	1	1
1	1	0	1	1	0
1	1	1	0	1	0

<u>Boolean algebra</u>: It is a method of expressing logic in a mathematical context. It a form of symbolic logic, in which variables have only the values “true (1)” or “false (0)”. Relationship between two values are expressed by the Boolean operators AND, OR and NOT. Boolean algebra is used to solve and minimize logic equations. Thus a minimize equation uses the minimum number of logic gates and reduces the cost of operation.

<u>Common Postulates of Boolean algebra</u> :- The postulates of Boolean algebra is originated from the three basic logic operations AND, OR and NOT.

$$\left.\begin{array}{ll} 1. & 0.0 = 0 \\ 2. & 0.1 = 0 \\ 3. & 1.0 = 0 \\ 4. & 1.1 = 1 \end{array}\right\} \text{Derived from AND operation}$$

$$\left.\begin{array}{ll} 5. & 0 + 0 = 0 \\ 6. & 0 + 1 = 1 \\ 7. & 1 + 0 = 1 \\ 8. & 1 + 1 = 1 \end{array}\right\} \text{Derived from OR operation}$$

$$\left.\begin{array}{ll} 9. & \overline{0} = 1 \\ 10. & \overline{1} = 0 \end{array}\right\} \text{Derived from NOT operation}$$

Basic theorems of Boolean algebra:

1. $0 \, . \, X = 0$
2. $X \, . \, 0 = 0$
3. $1 \, . \, X = X$
4. $X \, . \, 1 = X$
5. $X + 0 = X$
6. $0 + X = X$
7. $X + 1 = 1$
8. $1 + X = 1$
9. $1 + \overline{X} = 1$
10. $X. \, X = X$
11. $X. \, \overline{X} = 0$
12. $X + X = X$
13. $X + \overline{X} = 1$
14. $\overline{\overline{X}} = X$

Basic theorems of Boolean algebra:

$$\left.\begin{array}{l} 15.\ X + Y = Y + X \\ 16.\ X\ .\ Y = Y\ .\ X \end{array}\right\}\ \text{Commutative Laws}$$

$$\left.\begin{array}{l} 17.\ X.(YZ) = (X.Y).Z \\ 18.\ (X + Y) + Z = X + (Y + Z) \end{array}\right\}\ \text{Associative Laws}$$

$$\left.\begin{array}{l} 19.\ X.(Y + Z) = X.Y + X.Z \\ 20.\ X + YZ = (X + Y)\ (X + Z) \end{array}\right\}\ \text{Distributive Laws}$$

$$\left.\begin{array}{l} 21.\ \overline{X + Y} = \overline{X}\ .\ \overline{Y} \\ 22.\ \overline{X.\ Y} = \overline{X} + \overline{Y} \end{array}\right\}\ \text{De-Morgan's Theorems}$$

$$\left.\begin{array}{l} 23.\ X + XY = X \\ 24.\ X.(X + Y) = X \\ 25.\ X\,Y + X\,Y' = X \end{array}\right\}\ \text{Absorption Laws}$$

Basic theorems of Boolean algebra:

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

Proof of Theorem17.

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

L.H.S. $= X.(Y.Z)$

IF $X=0$,

$0.(YZ) = 0$ By Theorem 1

R.H.S $= (X.Y).Z$

IF $X = 0$,

$(0.Y).Z = 0.Z = 0$ By Theorem 1

Proved

Basic theorems of Boolean algebra:

Proof of Theorem 19.

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

L.H.S.$= X.(Y + Z)$

IF $X = 1$,

$1.(Y + Z) = 1.Y + 1.Z = Y + Z$ By Theorem 3

R.H.S $= X.Y + X.Z$

IF $X = 1$,

$1.Y + 1.Z = Y + Z$ By Theorem 3

Proved

Basic theorems of Boolean algebra:

Proof of Theorem 21 and 22(De-Morgan's theorem).

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

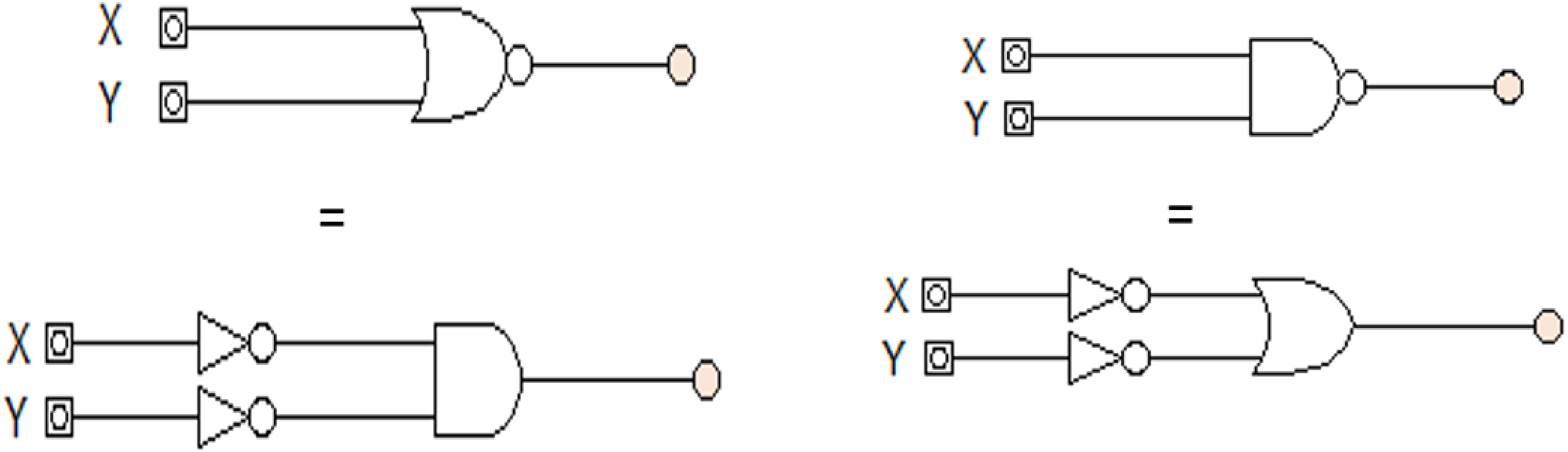
Proof by truth table:

X	Y	$\overline{X+Y}$	$\overline{X}\,.\,\overline{Y}$
0	0	1	1
0	1	0	0
1	0	0	0
1	1	0	0

X	Y	$\overline{X\,.\,Y}$	$\overline{X}+\overline{Y}$
0	0	1	1
0	1	1	1
1	0	1	1
1	1	0	0

Basic theorems of Boolean algebra:

Logic diagram for De-Morgan's theorem:



Hence, De- Morgan's theorems states that the complement of sum is equal to product of the complements and the complement of the product is equal to sum of the complements.

Basic theorems of Boolean algebra:

Proof of Theorem 24.

$X. (X + Y) = X$

$X. (X + Y)$

$= X.X + XY$

$= X + X.Y$ By theorem 10 i.e. $X.X = X$

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

Simplification of Boolean Expression by algebraic method:

1. Simplify:

$$X\overline{Y}\,\overline{Z} + X\overline{Y}\,\overline{Z}W + X\overline{Z}$$

$= x\, y'\, z'(1 + w) + x\, z'$

$= x\, y'\, z' + x\, z'$ As, $1 + w = 1$

$= x\, z'\, (x + 1) = x\, z'.1 = x\, z'$ As, $x + 1 = 1$

2. Simplify : $x + x'\, y + y' + x\, x'\, y + x'\, y'\, y$

$= x + x'\, y + y' + 0 + 0$ As, $x\, x'$ and $y\, y' = 0$

$= x + x'\, y + y' = x + y + y'$ As, $x + x'\, y = x + y$

$= x + 1$ As, $y + y' = 1$

$= 1$

Simplification of Boolean Expression by algebraic method:

3. Simplify: $z(y + z)(x + y + z)$

$= (yz + zz)(x + y + z)$

$= (yz + z)(x + y + z)$ As, $zz = z$

$= z(x + y + z)$ As, $z + yz = z$

$= xz + yz + zz = xz + yz + z = xz + z(y + 1) = xz + z = z(x + 1) = z.1 = z$

Different forms of Boolean Algebra:

1) Sum of Products (SOP): products terms are summed together.
   i) Minimal SOP: Minimum numbers of literals(variable)
      Ex. x y +x’ y +x y’
   ii) Expanded SOP: each products terms consists maximum number of Literal
      Ex. x y z + w x y z +x’ y z +w’ x y z
2) Product of Sums (POS): Summed terms are multiplied together.
   i) Minimal (POS): Minimum number of variables in sum terms.
      Ex. (x + y) (x’ +y) (x + y’)
   ii) Expanded POS: Maximum number of variable in sum terms.
      Ex. (x + y+ z) (w + x+ y +z) (w + x + y +z)

Canonical form of logic expression: When each term of logic expression contains All the variables, then that term is called canonical form. It is also called standard form of logic expression.

Minterm: It is canonical form of SOP expression. It is denoted m.
Maxterm: It is canonical form of POS expression. It is denoted by M.

Minterms and Maxterms for three variables:

X	y	z	Minterms	Notation	Maxterm	Notation
0	0	0	x' y' z'	m_0	(x + y + z)	M_0
0	0	1	x' y' z	m_1	(x + y + z')	M_1
0	1	0	x' y z'	m_2	(x + y'+ z)	M_2
0	1	1	x' y z	m_3	(x + y' + z')	M_3
1	0	0	x y' z'	m_4	(x' + y + z)	M_4
1	0	1	x y' z	m_5	(x'+ y + z')	M_5
1	1	0	x y z'	m_6	(x' + y'+ z)	M_6
1	1	1	x y z	m_7	(x' + y' + z')	M_7

Example: Express the Boolean function F = A + B'C in sum of minters.

First Term: $A = A(B + B') = AB + AB'$

$= AB(C + C') + AB'(C + C')$

$= ABC + ABC' + AB'C + AB'C'$

Second Term: $B'C = B'C(A + A')$

$= AB'C + A'B'C$

Combining all the terms,

$= ABC+ABC'+AB'C+AB'C' +AB'C+A'B'C$

$=ABC+ABC'+AB'C+AB'C'+A'B'C$

$= m7+m6+m5+m4+m1$

$F(A, B, C) = \sum(1,4,5,6,7)$

Example: Express the Boolean function F = x y + x' z in product of maxterm.

F = x y + x' z

= (x y + x') (x y + z)

= (x + x') (x'+ y) (x + z) (y + z) = (x' + y) (x + z) (y + z)

First Term: (x' + y)

x' + y + z z' = (x' + y + z) (x' + y + z')

Second term: (x + z)

x + z + y y' = (x + y +z) (x + y' + z)

Third term: (y + z)

y + z + x x'= (x + y + z) (x' + y + z)

Combining all the terms:

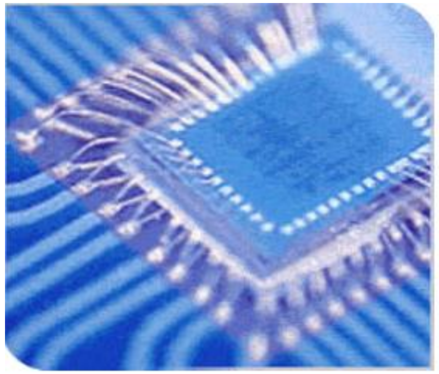
(x' + y + z) (x' + y + z') (x + y + z) (x + y' + z)

= M4 M5 M0 M2

= F(X, Y, Z) = $\pi(0,2,4,5)$

Integrated Circuits

- **Integrated circuit** (also called a *chip*) A piece of silicon on which many gates have been embedded
- An Integrated circuit is an association (or connection) of various electronic devices such as resistors, capacitors and transistors etched (or fabricated) to a semiconductor material such as silicon or germanium.
- It is also called as a **chip** or **microchip**.
- An IC can function as an amplifier, rectifier, oscillator, counter, timer and memory.



Types of ICs

- **Analog or Linear ICs:**
- They produce continuous output depending on the input signal.
- From the name of the IC we can deduce that the output is a linear function of the input signal.
- When the input and output relationship of a circuit is linear, linear IC is used.
- Input and output can take place on a continuous range of values
- Op-amp (operational amplifier) is one of the types of linear ICs which are used in amplifiers, timers and counters, oscillators etc.

Types of ICs

- **Digital ICs**:

- Unlike Analog ICs, Digital ICs never give a continuous output signal. Instead it operates only during defined states.
- When the circuit is either in ON state or OFF state and not in between the two, the circuit is called digital circuit and the IC used in such circuit is called digital IC.
- Digital ICs are used mostly in microprocessor and various memory applications.
- Logic gates are the building blocks of Digital ICs which operate either at 0 or 1.

Types of ICs

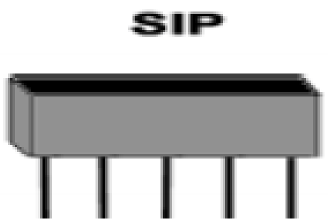
Linear ICs	Digital ICs
Linear ICs (Linear Integrated Circuits) are called as analog IC.	Digital ICs (Digital Integrated Circuits) are also called as non linear IC.
Linear integrated circuits inputs and outputs can take on a continuous range of values and the outputs are generally proportional to the inputs.	Digitals ICs contain circuits whose inputs and outputs voltage are limited to two possible levels low or high.
It is used in aircraft, space, vehicles, radars, PLL, Oscilloscopes etc.	Its used in microprocessor, computers, clocks, digital watches, calculator etc.
The design requirements are more drastic as compared to digital ICs.	The design requirement as less drastic as compare To linear ICs.
It is commercially available as operational amplifiers, voltage multipliers, voltage comparator, regulators, microwave amplifiers Etc.	Its commercially available as microprocessor chips, memory chips, analog to digital chips , digitals to analog chips, logic gates, flip flops, counters, registers etc.
Its consist of very less number of transistor as compared to digital ICs..	Its consist of more number of transistor as compared to linear ICs.

IC-Levels of Integration

- SSI (small-scale integration): Up to 100 electronic components per chip
- MSI (medium-scale integration): From 100 to 3,000 electronic components per chip
- LSI (large-scale integration): From 3,000 to 100,000 electronic components per chip
- VLSI (very large-scale integration): From 100,000 to 1,000,000 electronic components per chi
- ULSI (ultra large-scale integration): More than 1 million electronic components per chip

SIP (Single In-line Package)

- A **single in-line package** is an electronic device package which has one row of connection pins.
- It is not as popular as the dual in-line package (DIP) which contains two rows of pins, but has been used for packaging RAM chips and multiple resistors with a common pin.
- SIPs group RAM chips together on a small board.
- The board itself has a single row of pin-leads that resembles a comb extending from its bottom edge, which plug into a special socket on a system or system-expansion board.
- SIPs are commonly found in memory modules.



Dual in-line package (DIP)

- Dual in-line package (DIP) is a type of semiconductor component packaging.
- It is an electronic device package with a rectangular housing and two parallel rows of electrical connecting pins.
- DIPs can be installed either in sockets or permanently soldered into holes extending into the surface of the printed circuit board.
- DIP is relatively broadly defined as any rectangular package with two uniformly spaced parallel rows of pins pointing downward, whether it contains an IC chip or some other device(s), and whether the pins emerge from the sides of the package and bend downwards.
- A DIP is usually referred to as a **DIP*n***, where *n* is the total number of pins.

Eg IC 741 is DIP8

SIMM (Single In-line Memory Module)

- SIMM has a single line of connector.
- In a SIMM pins on opposite sides of board are tied together to form one electrical path.
- Short for **Single In-line Memory Module, SIMM** is a circuit board that holds six to nine memory chips per board, the ninth chip usually an error checking chip (parity/non parity) and were commonly used with Intel Pentium or Pentium compatible motherboards.
- SIMMs are rarely used today and have been widely replaced by DIMMs.
- SIMM ha 32 bit data path
- SIMMs are available in two flavors: 30 pin and 72 pin.
- 30-pin SIMMs are the older standard, and were popular on third and fourth generation motherboards.
- 72-pin SIMMs are used on fourth, fifth and sixth generation PCs.

DIMM (Dual In-line Memory Module)

- DIMM has two lines of connectors.
- Short for **Dual In-line Memory Module, DIMM** is a circuit board that holds memory chips.
- In a DIMM opposite pins remain electrically isolated to form two separate contact.
- DIMMs have a 64-bit data path because of the Pentium Processor requirements.
- Because of the new bit path, DIMMs can be installed one at a time, unlike SIMMs on a Pentium that would require two to be added

DIMM (Dual In-line Memory Module)

- **SO-DIMM** is short for **Small Outline DIMM** and is available as a 72-pin and 144-pin configuration.
- SO-DIMMs are commonly utilized in laptop computers.
- Advantages DIMMs have over SIMMs:
- DIMMs have separate contacts on each side of the board, thereby providing twice as much data as a single SIMM.
- The command address and control signals are buffered on the DIMMs. With heavy memory requirements this will reduce the loading effort of the memory.