1.1 Introduction to Analog and Digital signals

Electronic systems usually deal with information. Representation of information is called a **signal.** Signal in electronics is generally in form of voltage or current. Value of a signal is proportional to some physical quantity and it gives information about it. For example, temperature represented in terms of voltage signal.

There are two types of signals which are different in terms of their characteristics with respect to time and value.

1. Analog Signals
2. Digital Signals

A signal whose value is defined at all instances of time is called continuous time signal. On the other hand signal whose values are defined only at discrete instances of time is called discrete time signal. Most of the signals that occur in nature are analog in form. A discrete time signal can be obtained from continuous time signal by process called sampling. This has been illustrated in Fig. 1.1.

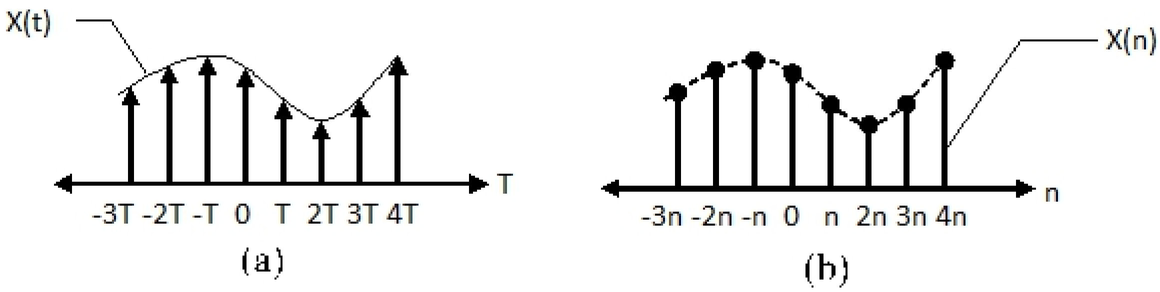


Fig. 1.1: (a) Continuous time signal x(t) sampled at every T interval, (b) Resulting discrete time signal x(n)

Similarly if a signal can take any value in a given range between some minimum and maximum value then the signal is called continuous value signal. On the other hand if a signal takes only certain fixed values in a given range then it is called discrete value signal. The process of converting a continuous value signal to a discrete value signal is called quantization. This is illustrated in Fig. 1.2.

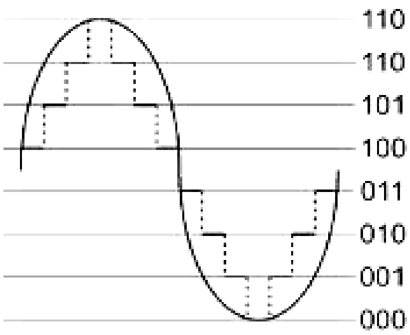


Fig. 1.2: Continuous value signal (solid line) and discrete value signal (dotted line)

Analog signal: Signals that are continuous in time and continuous in value are called analog signal.

Digital signal: Signals that are discrete in time and discrete in values are called digital signals. Digital signals are generally processed by digital systems like computers and hence their values are represented in terms of binary as shown in Fig. 1.2.

Analog signal being continuous in time will have infinite values in any given period of time. Practically a digital system like computer cannot handle infinite values due to limited physical resources and processing power. This is the reason why a continuous time signal has to be sampled and converted to discrete time signal.

Again analog signals are continuous in value and hence can take any value in a given range. Now ideally number of values in any given range will be infinite which cannot be represented by finite number of bits on a computer. For example, as shown in Fig. 1.2, with three bits used for representing values only eight different values can be represented. Thus a continuous value signal has to be quantized and converted to discrete value signal.

1.2 Introduction to Digital System.

A digital system uses a building blocks approach. Many small operational units are interconnected to make up the overall system. The most basic logical unit system is gate circuit. There are several different types of gates with each perform differently from other logic gates.

Digital signal consist of only two values, ‘0’ and ‘1’. These two values have logical meaning i.e. ‘1’ represents the existence of particular condition and ‘0’ represents the absence of condition.

There are two types of tables used in digital system:

(i) Truth Table:
Truth table plots inputs and outputs in terms of 1s and 0s.
(ii) Function Table:
Function table plots inputs and outputs in term of HIGH and LOW voltage levels.

The design of digital system may be roughly divided into three stages;

(i) System Design:
It involves breaking the overall system into subsystem and specifying the characteristics of each subsystem. For example, the system design of a digital computers involves specifying the number and type of memory, ALU and i/p – o/p devices.
(ii) Logic Design:
It involves how to interconnect basic logic building blocks to perform specific function. For example, to make a flip flop different logic gates need to be connected in specific manner.
(iii) Circuit Design:
It involves specifying the interconnection of specific components like resistors, transistors, diodes, CMOS etc. to create a logic gates.

1.3 Comparison between Analog and Digital Systems

	Analog Systems	Digital Systems
1	Analog systems operate on continuous time and continuous value signals.	Digital systems operate on discrete time and discrete value signals generally represented in binary.
2	Analog systems are difficult to design.	Digital systems are easy to design as most of the components are in form of Integrated circuits (IC).
3	Analog systems are mostly custom made and less flexibility.	Digital systems have high degree of flexibility.
4	Less efficient in storage of information.	More efficient in storage of information.
5	Analog signal processed by these systems are affected by noise very easily.	Digital signal are less affected by noise compared to analog signals.
6	Relatively costly compared to digital system	Low cost due to mass production of components.
7	Analog systems are more sensitive to parameter variation.	Digital systems are less sensitive to parameter variation
8	No conversion of input signals are required before processing	Input signals are converted from analog to digital form before it is processed
9	As no conversion of input signal is required, there is no loss of information.	Due to process of sampling and quantization, there is loss of information.
10	Analog systems are more efficient for real time processing	Digital systems may offer limitations for real time processing
11	Probability of error is more in Analog system	Probability of error is less in Analog system

1.4 Advantages of Digital Systems.

(i) Digital systems are easier to design
(ii) Digital systems have high degree of flexibility.
(iii) Information storage is easy
(iv) Digital circuits are less affected by noise
(v) Probability of error is less
(vi) Low cost and more reliable
(vii) More digital circuitry can be fabricated on IC chips
(viii) Digital systems are less sensitive to parameter variation
(ix) Accuracy and precision are greater
(x) Digital systems are based on Boolean algebra which is easy.
(xi) Digital system concerned with only two logic levels.
(xii) There are very few basic operations in digital and are very easy.

1.5 Logic Levels and Different types of Logics.

Digital system use the binary number system. Therefore, two-state devices are used to represent the two binary digits 1s & 0s by two different voltage levels, called HIGH and LOW. Normally, the binary 0 and 1 are represented by the logic voltage levels 0 V and +5 V.

Usually any voltage between 0 V to 0.8 V represents the logic 0 and any voltage between 2 V to 5 V represents the logic 1. This voltage levels can be varies according to the different logical systems.

There are three types of logics available in digital systems.
1. Positive Logic
2. Negative Logic
3. Mixed Logic

1. Positive Logic:
In positive logic high voltage level represents as logic 1 and low voltage level represents as logic 0.

Fig. 1.3: Illustration of positive logic

2. Negative Logic:
In positive logic high voltage level represent as logic 0 and low voltage level represents as logic 1.

Fig. 1.4: Illustration of negative logic

3. Mixed Logic:
This scheme uses positive logic in some portions (e.g inputs) of the system while applying negative logic (e.g. outputs) in other portion of the system.

Suppose some function X = AB’ + A’B for this function the representation of all the logics are as follow;

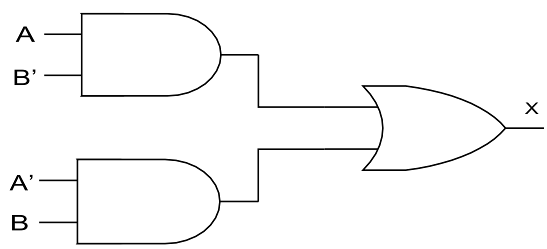


Fig. 1.5: Representation of function X = AB’ + A’B

Truth table of the given function for all the logics is shown as follow;

A	B	X
0	0	0
0	1	1
1	0	1
1	1	0

Positive Logic

A	B	X
1	1	1
1	0	0
0	1	0
0	0	1

Negative Logic

A	B	X
0	0	1
0	1	0
1	0	0
1	1	1

Mixed Logic

1.6 Boolean Algebra

Boolean Algebra is the mathematics we use to analyze digital logic gates and circuits. We can use these "Laws of Boolean" to both reduce and simplify a complex Boolean expression in an attempt to reduce the number of logic gates required in digital system design.

Boolean Algebra is therefore a system of mathematics based on logic that has its own set of rules or laws which are used to define and reduce Boolean expressions.

The variables used in **Boolean Algebra** only have one of two possible values, a logic "0" and a logic "1". *In Boolean algebra, variable do not represent numerical value but it has logical value*. In Boolean algebra an expression given can also be converted into a logic diagram using different logic gates like AND gate, OR gate and NOT gate, NOR gates, NAND gates, XOR gates, XNOR gates etc.

Some basic logical Boolean operations:

OR relations (Logical Addition)

$A + A = A$ $\qquad$ $0 + 0 = 0$

$A + \bar{A} = 1$ $\qquad$ $0 + 1 = 1$

$A + 1 = 1$ $\qquad$ $1 + 0 = 1$

$A + 0 = A$ $\qquad$ $1 + 1 = 1$

AND relations (Logical Multiplication)

$A . A = A$ $\qquad$ $0 . 0 = 0$

$A . \bar{A} = 0$ $\qquad$ $0 . 1 = 0$

$A . 1 = A$ $\qquad$ $1 . 0 = 0$

$A . 0 = 0$ $\qquad$ $1 . 1 = 1$

Complement Rule (Inversion)

$A + \bar{A} = 1$ $\qquad$ $\bar{0} = 1$

$A . \bar{A} = 0$ $\qquad$ $\bar{1} = 0$

$A = A$

De-Morgan's Theorem:

(i) $\overline{A+B} = \bar{A}\,\bar{B}$

A	B	A + B	$\overline{A+B}$	$\bar{A}$	$\bar{B}$	$\bar{A}\,\bar{B}$
0	0	0	**1**	1	1	**1**
0	1	1	**0**	1	0	**0**
1	0	1	**0**	0	1	**0**
1	1	1	**0**	0	0	**0**

Hence, from Truth Table it has been observed that LH.S. = R.H.S.

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

A	B	A . B	$\overline{AB}$	$\bar{A}$	$\bar{B}$	$\bar{A}+\bar{B}$
0	0	0	**1**	1	1	**1**
0	1	0	**1**	1	0	**1**
1	0	0	**1**	0	1	**1**
1	1	1	**0**	0	0	**0**

Hence, from Truth Table it has been observed that LH.S. = R.H.S

1.7 Logic Gates

Logic gates are the fundamental building blocks of digital systems. They are the physical devices that performs the basic Boolean operations of AND, OR and NOT. Input and outputs of logic gates (that is basically a voltage signal) can occur only in two levels. These two levels are termed as :

Logic 1	Logic 0
High Level	Low Level
True	False
ON	OFF

If in representation of higher of the two voltage levels are symbolized as 1 and lower symbolized as 0 then gate is said to be positive logic gate. However, if higher of the two voltage levels is symbolized as 0 and lower as 1 then it is said to be negative logic gate.

Input output behavior of a gate is generally represented using **truth table**. It is a table that lists output for all possible combinations of inputs.

There are total seven logic gates in which three are **basic logic gates** (AND, OR, NOT) and two are **universal logic gates** (NAND, NOR).

Various basic gates are discussed as follows;

1. NOT Gate:

NOT gate has one input and one output. The output becomes logic 1 when input is at logic 0 and output becomes logic 0 when the input is at logic 1. Thus it inverts or complements the logic available at input and hence called an **inverter or complement.**

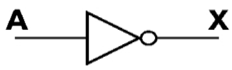
It is represented by a bar over the variable “¯”or with a symbol “ ’ ”. Thus, for example, X = A′ or X = $\overline{A}$ read as “X is equal to Not A or A bar or A complement”. NOT gate and its truth table are shown in fig. 1.6.

IC 7404 is Hex Inverter, consists of six NOT gates on a single chip.

Boolean Expression

X = A'

Logic Diagram Symbol



Truth Table

A	X
0	1
1	0

Fig. 1.6: Illustration of NOT gate

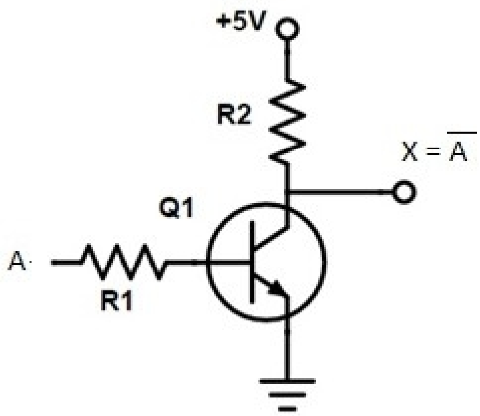


Fig. 1.7: Circuit diagram of NOT gate

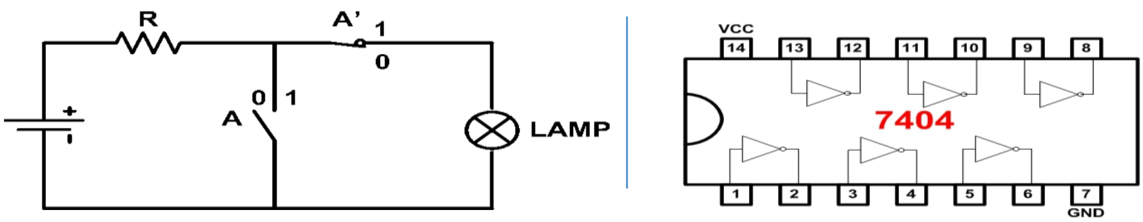


Fig. 1.8: NOT gate switching diagram and IC

2. AND Gate:

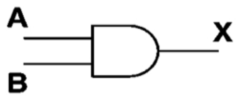
AND gate means all or nothing logic. AND gate has two or more inputs and one output. The output becomes logic 1 only when each one of its input is at logic 1. For all other input combinations it gives output logic 0. It is represented by a symbol •. Thus, for example, X = A

· B (also written simply as X = AB) is read as "X is equal to A AND B". Two input AND gate and its truth table is shown in fig. 1.9.

Boolean Expression

$X = A \cdot B$

Logic Diagram Symbol



Truth Table

A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

Fig. 1.9: Illustration of AND gate

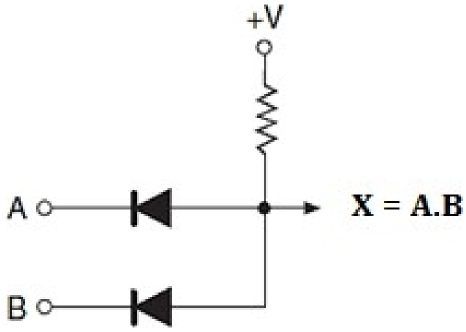


Fig. 1.10: Circuit diagram of AND gate

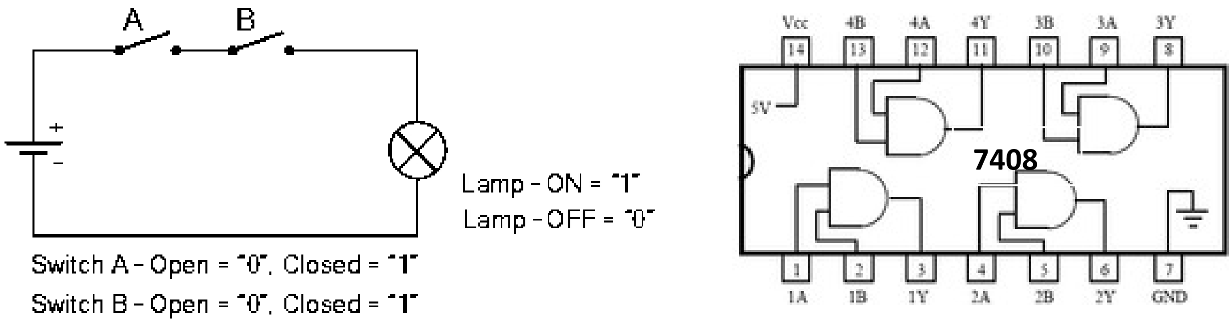


Fig. 1.11: AND gate switching diagram and IC

Working of AND gate circuit using Diode:

When A=0 and B=0, both the diodes are in forward biased condition. Hence current will flow through resistor and diode. So voltage will drop across resistor and output will be Zero, resulting in output Logic-0

When A=0 and B=1, or A=1 and B=0, one of the diodes is in forward biased condition. Which allows the current to pass through resistor and diode. So voltage will drop across resistor again and output will be Zero, resulting in output Logic-0.

When A=1 and B=1, both the diodes are in Reverse biased condition. Hence current will not flow through resistor and diode. So there is no voltage will drop across resistor and this voltage will be available at output, resulting in output Logic-1.

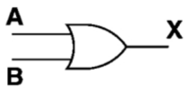
3. OR Gate:

OR gate means any or all logic. OR gate has two or more inputs and one output. The output becomes logic 1 when at least (minimum) one of the inputs is at logic 1. It is represented by a symbol **+**. Thus, for example, X = A + B is read as "X is equal to A OR B". Two input AND gate and its truth table is shown in fig. 1.12.

Boolean Expression

X = A + B

Logic Diagram Symbol



Truth Table

A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

Fig. 1.12: Illustration of AND gate

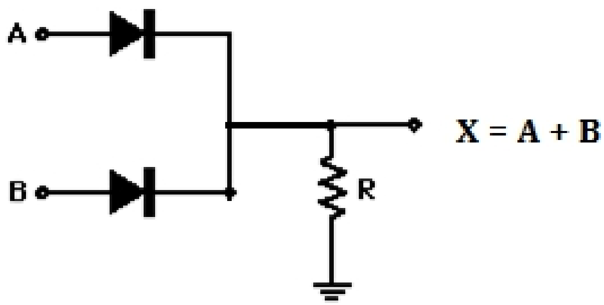


Fig. 1.13: Circuit diagram of OR gate

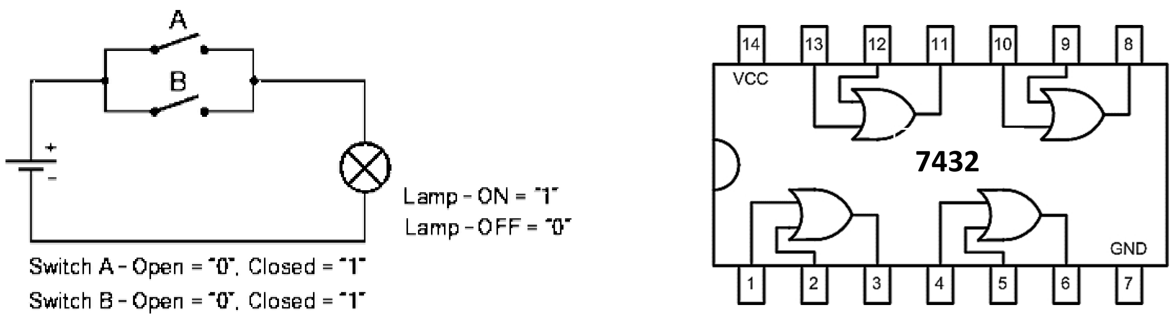


Fig. 1.14: OR gate switching diagram and IC

Working of OR gate circuit using Diode:

When A=0 and B=0, both the diodes are in reverse biased condition. Hence diode behaves as an OFF switch and current will not flow through diode and resistor. So output will be Zero, (Logic-0).

When A=0 and B=1, or A=1 and B=0, one of the diodes is in forward biased condition, which allows the current to pass through diode and resistor. Hence diode behaves as an ON switch. So there will be voltage drop across resistor which will be available as an Output, resulting in output Logic-1.

When A=1 and B=1, both the diodes are in forward biased condition. which allows the current to pass through diode and resistor. Hence diode behaves as an ON switch. So there will be voltage drop across resistor which will be available as an Output, resulting in output Logic-1.

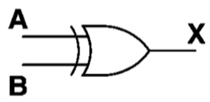
4. Exclusive OR Gate (EX-OR):

It also means ***Inequality detector*** because it gives output high when both inputs are different. Exclusive OR gate give output equal to 1 when the two inputs are exclusively different. This is the reason why it is also known as inequality gate. The schematic symbol and truth table of the gate is shown in fig. 1.15. It is represented by a symbol $\oplus$. Thus, for example, $X = A \oplus B$ is read as "X is equal to A XOR B." The logic expression this gate in terms of AND, OR and NOT operation is $X = A \oplus B = \overline{A}B + A\overline{B}$.

Boolean Expression

$X = A \oplus B$

Logic Diagram Symbol



Truth Table

A	B	X
0	0	0
0	1	1
1	0	1
1	1	0

Fig. 1.15: Illustration of EX-OR gate

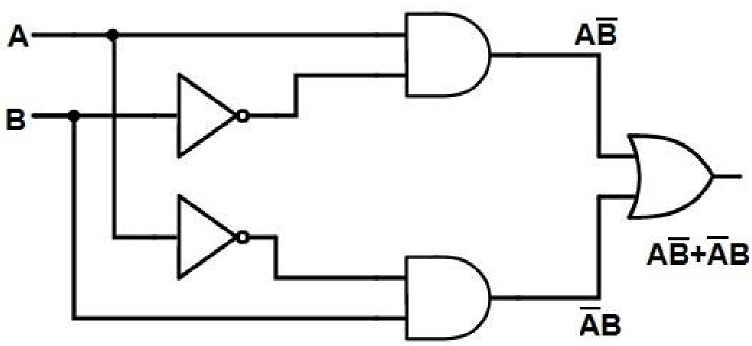
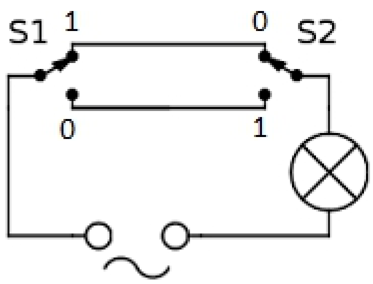


Fig. 1.16: Logic Circuit diagram of EX-OR gate



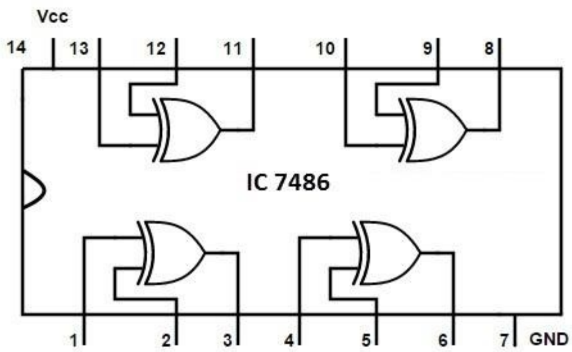


Fig. 1.17: EX-OR gate switching diagram and IC

5. Exclusive NOR Gate (EX-NOR):

It also means ***equality detector*** because it gives output high when both inputs are same. Exclusive NOR gate is XOR gate followed by inverter. Thus it is complement of XOR gate. This is the reason why it is also known as equality gate.

The schematic symbol and truth table of the gate is shown in fig. 1.18. It is represented by a symbol ⊙. Thus, for example, X = A ⊙ B is read as "X is equal to A XNOR B."

The logic expression this gate in terms of AND, OR and NOT operation is

$$X = A \odot B = A\,B + \overline{A}B$$

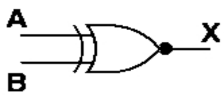
or

$$\mathbf{X = AB + A'B'}$$

Boolean Expression

$X = A \odot B$

Logic Diagram Symbol



Truth Table

A	B	X
0	0	1
0	1	0
1	0	0
1	1	1

Fig. 1.18: Illustration of EX-NOR gate

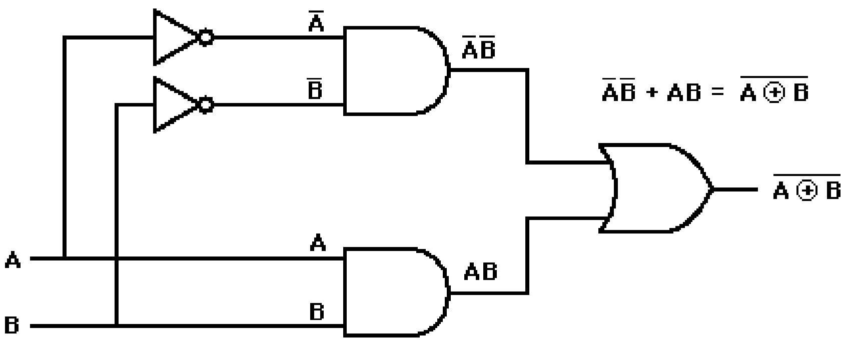
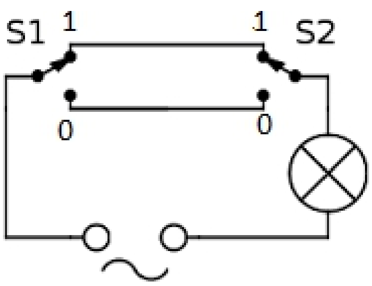


Fig. 1.19: Logic Circuit diagram of EX-NOR gate



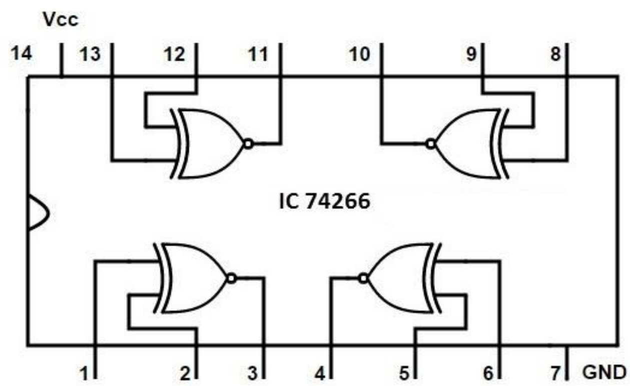


Fig. 1.20: EX-NOR gate switching diagram and IC

Universal Gates:

NAND and NOR gates are known as a universal gates because from this two gates all other gates can be constructed.

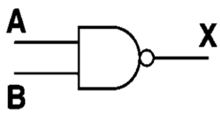
6. NAND Gate:

NAND gate represents combination of AND gate followed by NOT gate. It represents complement of AND operation. Schematic symbol of NAND gate and its truth table are shown in fig. 1.15. The logic expression is given as $X = \overline{(A \cdot B)}$ or X = (A·B)'.

Boolean Expression

X = (A · B)'

Logic Diagram Symbol



Truth Table

A	B	X
0	0	1
0	1	1
1	0	1
1	1	0

Fig. 1.21: Illustration of NAND gate

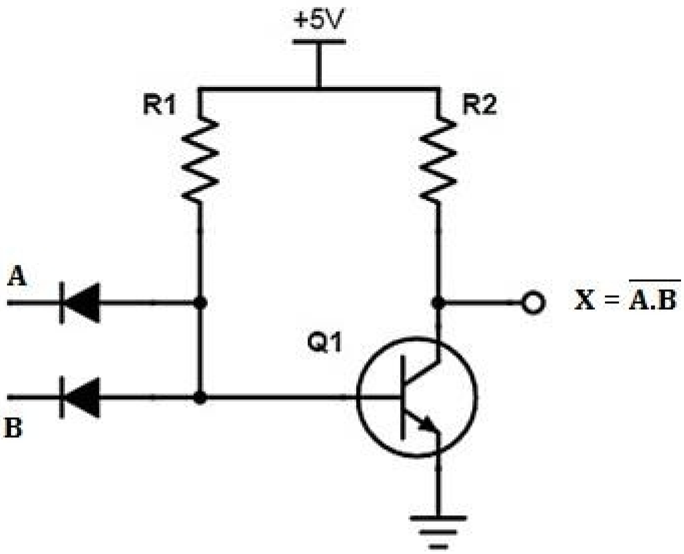


Fig. 1.22: Logic Circuit diagram of NAND gate

Working of NAND gate circuit:

When either A=0 or B=0, diode will be in forward biased condition. Hence diode behaves as an ON switch and current will flow through diode and Resistor R1. Which results in voltage drop across Resistor R1. So Base current to Transistor is zero and Output of Transistor will be Logic 1.

When A=1 and B=1, both the diodes are in reverse biased condition. Which do not allow the current to pass through diode and Resistor R1. Hence diode behaves as an OFF switch. So there will be no voltage drop across resistor. Hence Base current is maximum (Logic-1), making Transistor ON switch and output of Transistor will be Logic-0.

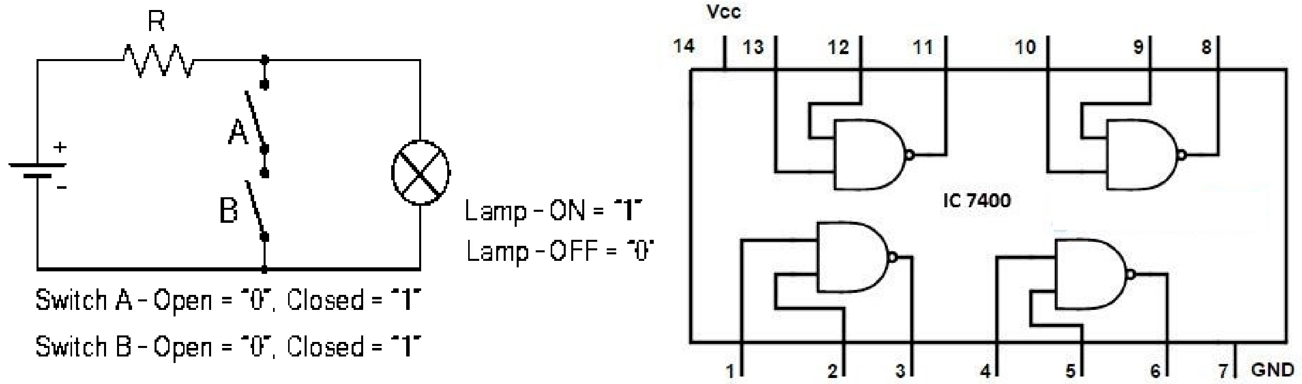
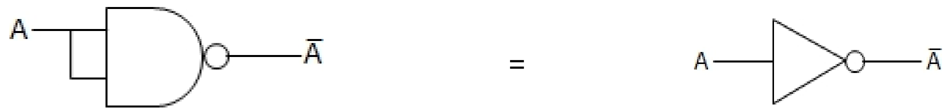


Fig. 1.23: NAND gate switching diagram and IC

NAND gate as Universal gate

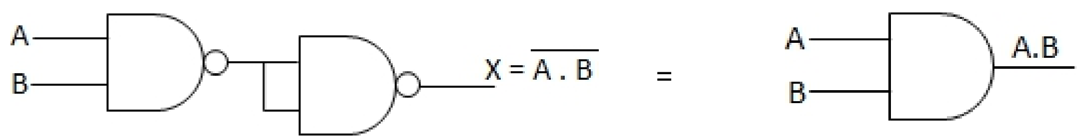
i. Implementing NOT gate

All NAND input pins connect to the input signal A gives an output A'.



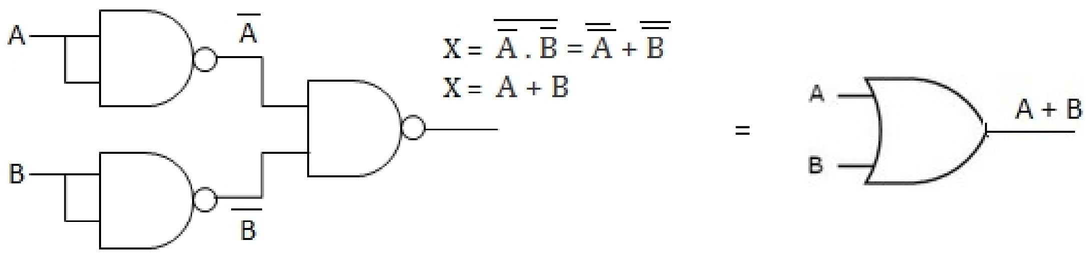
ii. Implementing AND gate

The AND is replaced by a NAND gate with its output complemented by a NAND gate inverter.



iii. Implementing OR gate

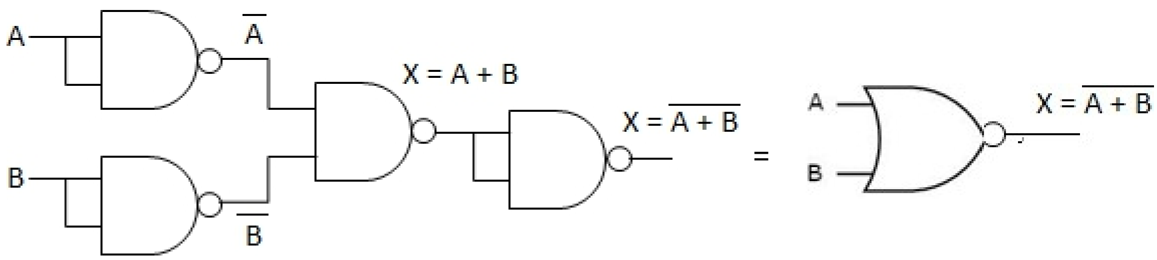
The OR gate is replaced by a NAND gate with all its inputs complemented by NAND gate inverters.



As per De-Morgan's Theorem $\overline{AB} = \bar{A} + \bar{B}$

For above diagram output $X = \overline{\bar{A}\bar{B}} = \bar{\bar{A}} + \bar{\bar{B}} = \mathbf{A + B}$ (which equivalent to OR gate)

iv. Implementing NOR gate



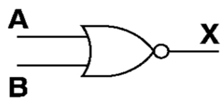
7. NOR Gate:

NOR gate represents combination of OR gate followed by NOT gate. It represents complement of OR operation. Schematic symbol of NOR gate and its truth table are shown in fig. 4.16. The logic expression is given as $X = \overline{(A + B)}$ or $X = (A+B)'$.

Boolean Expression

X = (A + B)'

Logic Diagram Symbol



Truth Table

A	B	X
0	0	1
0	1	0
1	0	0
1	1	0

Fig. 1.24: Illustration of NOR gate

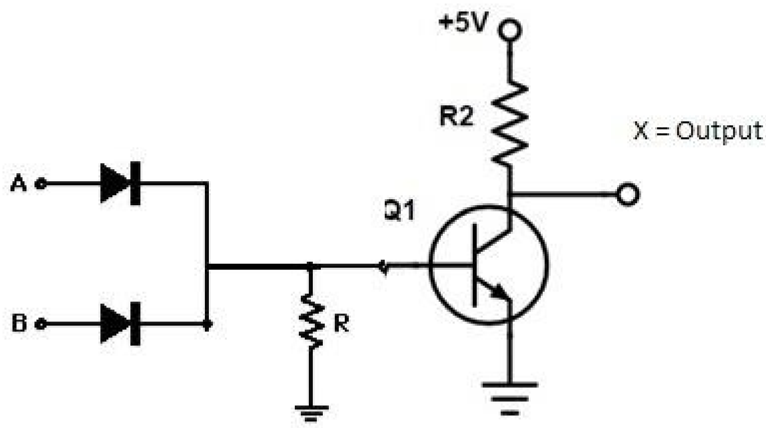


Fig. 1.25: Logic circuit diagram of NOR gate

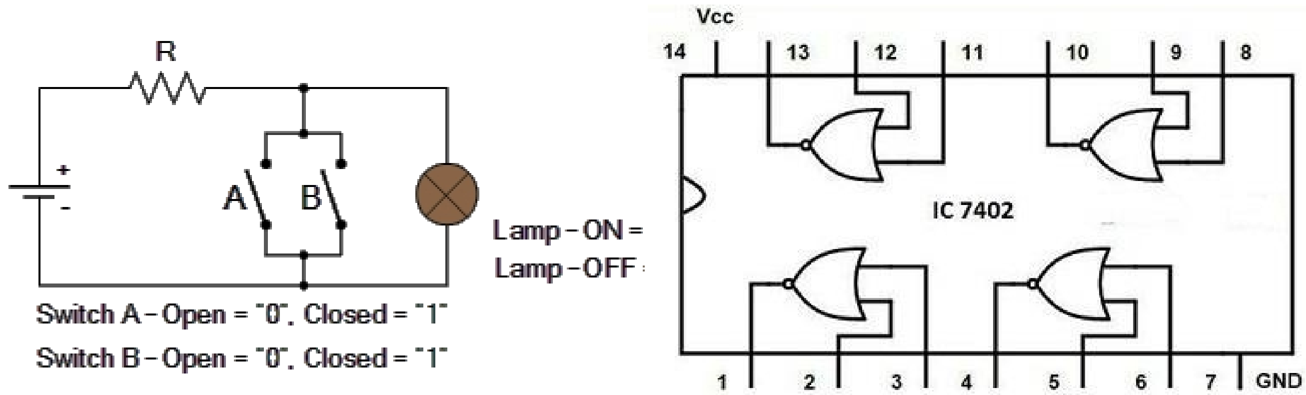
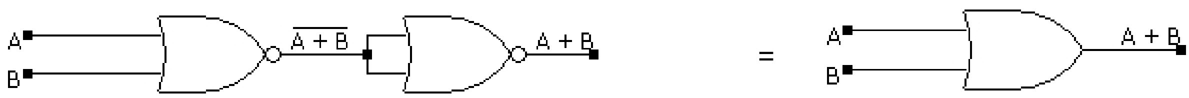


Fig. 1.26: NOR gate switching diagram and IC

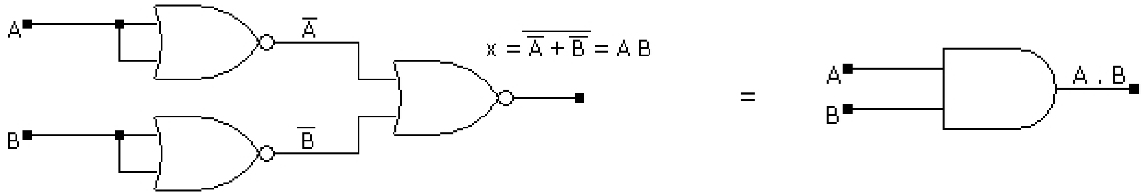
NOR gate as Universal gate.

Implementation of ***NOT*** *Gate:*

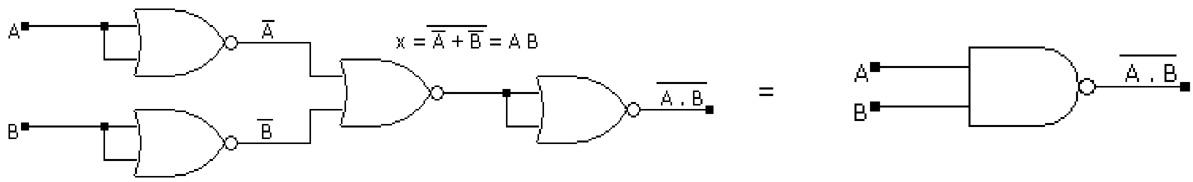
Implementation of ***OR*** *Gate:*



Implementation of ***AND*** *Gate:*



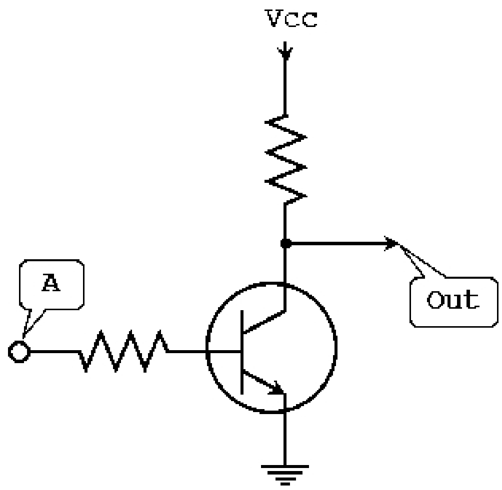
Implementation of ***NAND*** *Gate:*



1.8 Construct Logic Gates using Transistors.

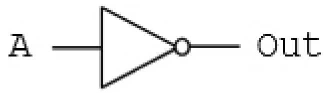
A transistor is a device that acts, depending on the voltage level of an input signal, either as a wire that conducts electricity or as a resistor that blocks the flow of electricity. A transistor acts like a switch.

*Implementation of **NOT** Gate:*

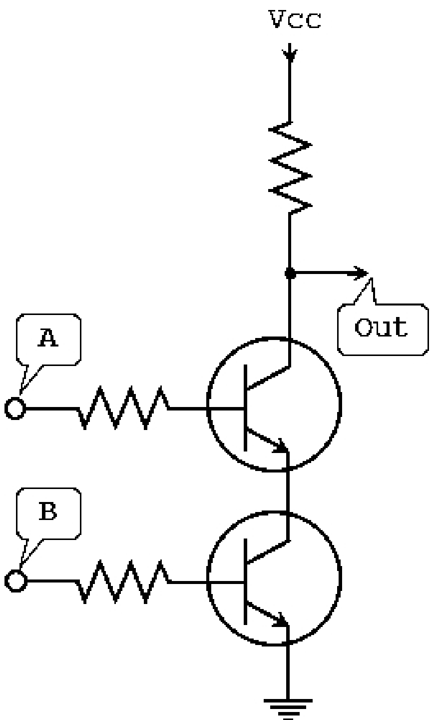


=

Truth Table	
A	**X**
0	1
1	0



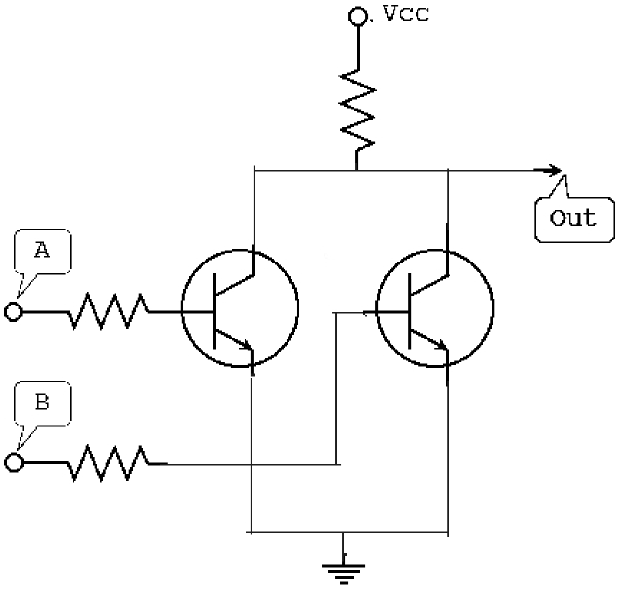
*Implementation of **NAND** Gate:*



=

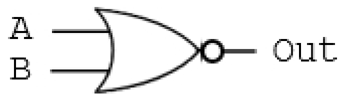
Truth Table		
A	**B**	**X**
0	0	1
0	1	1
1	0	1
1	1	0

*Implementation of **NOR** Gate:*

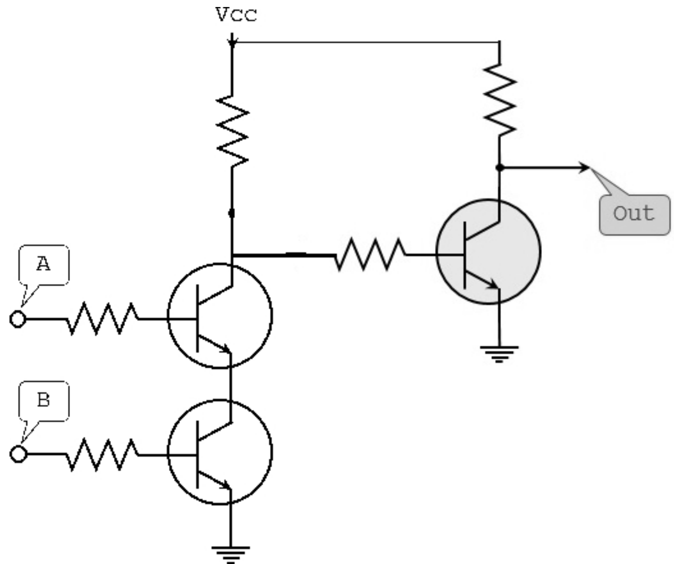


=

Truth Table		
A	**B**	**X**
0	0	1
0	1	0
1	0	0
1	1	0

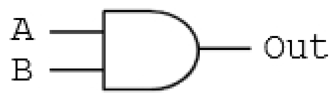


*Implementation of **AND** Gate:*

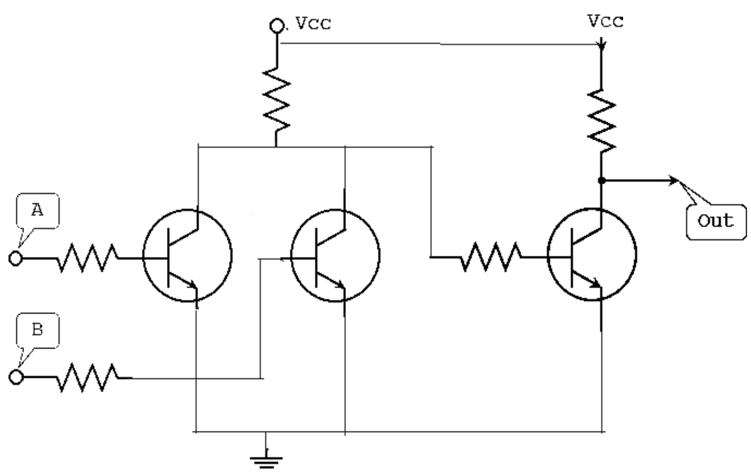


=

Truth Table		
A	**B**	**X**
0	0	0
0	1	0
1	0	0
1	1	1

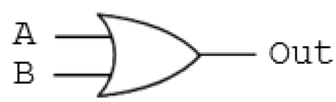


*Implementation of **OR** Gate*



=

Truth Table		
A	**B**	**X**
0	0	0
0	1	1
1	0	1
1	1	1



1.8 Introduction to Logic Families:

It is a circuit configuration or approach used to produce a type of digital integrated circuit. The entire range of digital ICs is fabricated using either bipolar devices or MOS devices or a combination of the two.

Different logic families falling in the first category are called bipolar families, and these include

- Diode Logic (DL)
- Resistor Transfer Logic (RTL)
- Diode Transistor Logic (DTL)
- Transistor Transistor Logic (TTL)
- Emitter Coupled Logic (ECL)
- Integrated Injection Logic (I^2L)

The logic families that use MOS devices as their basis are known as MOS families, and the prominent members belonging to this category are

- PMOS Family (using P-channel MOSFETs)
- NMOS Family (using N-channel MOSFETs)
- CMOS Family (using both N- and P-channel devices)
- Bi-MOS Logic Family uses both Bipolar and MOS devices

From all the logic families listed above, the first three, i.e. DL, RTL and DTL have become obsolete

- Diode Logic used diode & resistors and in fact was never implemented in integrated circuits
- RTL Family used resistors and bipolar transistors
- DTL Family used resistors, diodes and bipolar transistors.

PMOS and I^2L logic families, which were mainly intended for use in custom large-scale integrated (LSI) circuit devices, have also become more or less obsolete.

Logic families that are still in widespread used include TTL, CMOS, ECL, NMOS and Bi-CMOS.

(1) Diode Logic (DL)

Some logic gates can be produced with just Diodes and Resistors are called Diode resistor logic or Diode Logic. Diode logic was used extensively in the construction of early computers.

While diode logic has the advantage of simplicity, the lack of an amplifying stage in each gate limits its application. Not all logical functions can be implemented in diode logic alone. Only the logical AND and logical OR functions can be realized by diode gate.

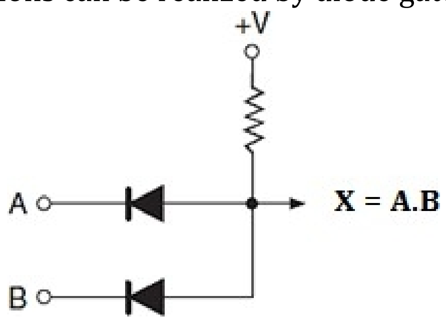


Fig. 1.27: Diode Logic for AND function

(2) Resistor Transistor Logic (RTL)

Digital circuits built using resistors as the input network and bipolar junction transistors (BJTs) as switching devices is known as **Resistor Transistor Logic (RTL)**.

A bipolar transistor switch is the simplest RTL gate (inverter or NOT gate). With two or more base resistors instead of one, the inverter becomes a two-input RTL NOR gate as shown in fig.1.28.

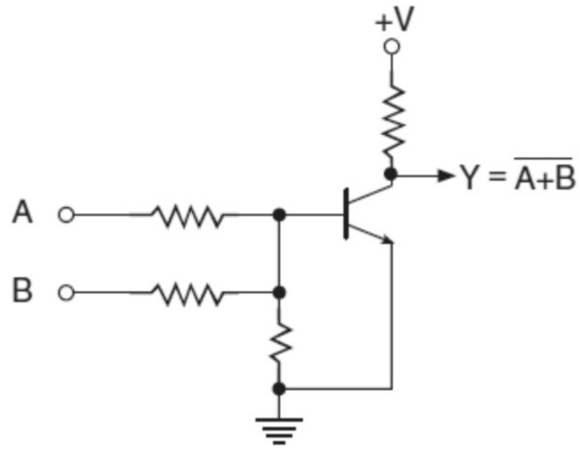


Fig. 1.28: Resistor Transistor Logic for NOR function

The multi-transistor RTL implementation is shown in Fig.1.29. It consists a set of parallel connected transistor switches driven by the logic inputs.

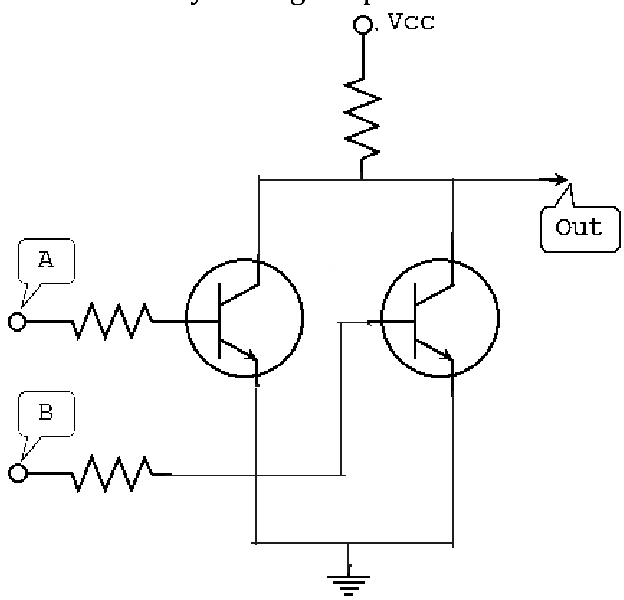


Fig. 1.29: Resistor Transistor Logic for NOR function

The primary advantage of RTL technology was that it used a minimum number of transistors. The disadvantage of RTL is its high power dissipation.

(3) Diode Transistor Logic (DTL)

Diode-Transistor Logic, or DTL, refers to the technology for designing and fabricating digital circuits in which logic gates employ both diodes and transistors.

DTL offers better noise margins and greater fan-outs than RTL, but suffers from low speed, especially in comparison to TTL

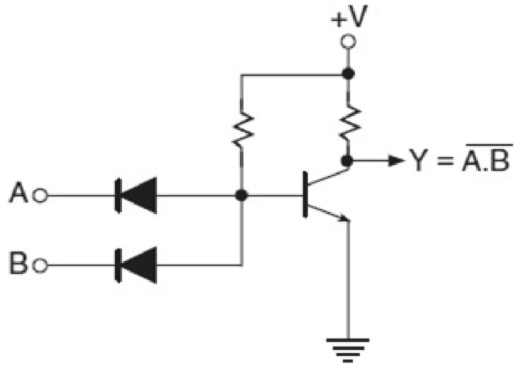


Fig. 1.30: Diode Transistor Logic for NAND function

(4) Transistor Transistor Logic (TTL)

Transistor Transistor Logic (TTL) is a logic family built from bipolar junction transistors. TTL integrated circuits (ICs) were widely used in applications such as computers, industrial controls, test equipment and instrumentation, consumer electronics, and synthesizer.

The multiple-emitter BJT can be used to implement NAND logic. The multiple-emitter BJT forms an integral part of the TTL NAND input circuitry (e.g. the 7400 series of integrated circuits).

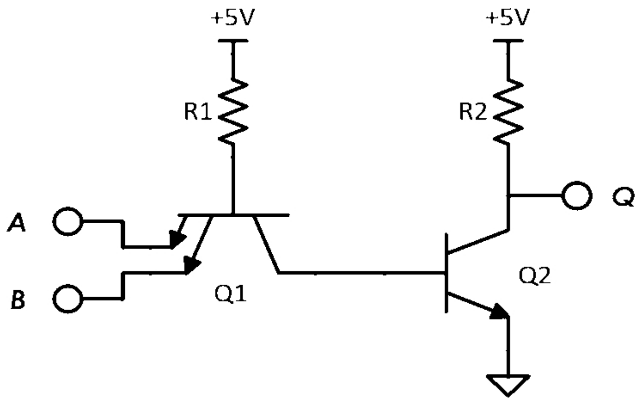


Fig. 1.31: Basic two input TTL NAND function

A two input Basic TTL NAND is shown in Fig.1.31 and two input TTL NAND gate with TOTEM POLE output is shown in Fig.1.32.

Operation of the TTL NAND gate with TOTEM POLE output:

When ***A and B both low***: both B-E junctions of T1 are forward biased. Hence BE junction of transistor Q1 will conduct and force the voltage at collector of Q1 to 0.7V. This voltage is insufficient to forward bias B-E junction of Q2 transistor. Hence Q2 remains OFF. Therefore its collector voltage rises to VCC. As Q3 is operating in emitter follower mode, output Y will be pulled up to high voltage Y= 1

When ***Either A or B low***: If any one input is connected to ground with other left open or connected to logic 1, the corresponding BE junction of transistor Q1 will conduct. This will pull down voltage at Collector of Q2 to 0.7V. This voltage is insufficient to turn on Q2 so it remains OFF. So collector voltage of Q2 will be equal to VCC. This voltage acts as base voltage for Q3. As Q3 acts as an emitter follower, output Y will be pulled to high voltage Y= 1

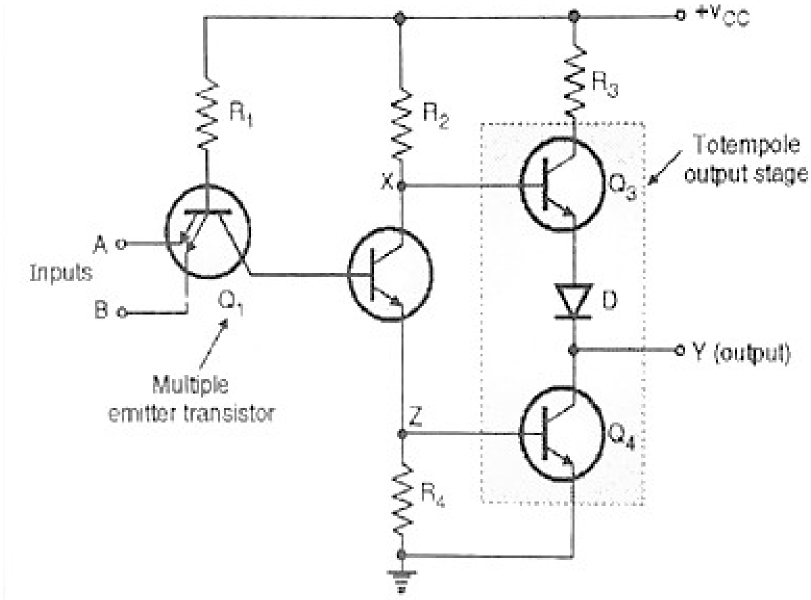


Fig. 1.32: TTL NAND function with TOTEM POPLE output

When ***A and B both high***: If both A and B are connected to logic 1 then both BE junction of transistor Q1 will be reverse biased and do not conduct. Therefore BC junction of Q1 is forward biased and base current is supplied to transistor Q2. As Q2 conducts, the voltage at X will drop down and Q3 will be OFF, whereas voltage at Z will increase to turn ON Q4. As Q4 goes into saturation, the output voltage Y will be pulled down to logic 0. Y = 0.

Diode D prevents conduction of Q3 when output is low. Voltage drop across D keeps emitter of Q3 reverse biased.

(5) Integration Injection Logic (I^2 L)

Integrated injection logic (IIL, I^2L, or I2L) is a class of digital circuits built with multiple collector bipolar junction transistors (BJT). When introduced it had speed comparable to TTL yet was almost as low power as CMOS, making it ideal for use in VLSI (and larger) integrated circuits.

The heart of an I2L circuit is the common emitter open collector inverter. Typically, an inverter consists of an NPN transistor with the emitter connected to ground and the base biased with a forward current.

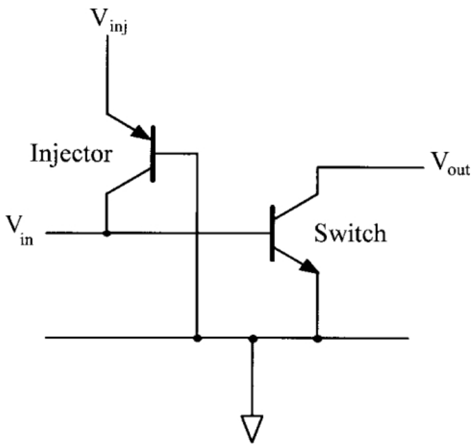


Fig. 1.33: Integration Injection Logic for NOT function

(6) Emitter Coupled Logic (ECL)

Emitter-coupled logic (ECL) is a BJT-based logic family which is generally considered as the fastest logic available. ECL achieves its high-speed operation by employing a relatively small voltage swing and preventing the transistors from entering the saturation region. It has low noise margin. It is preferred for high frequency application.

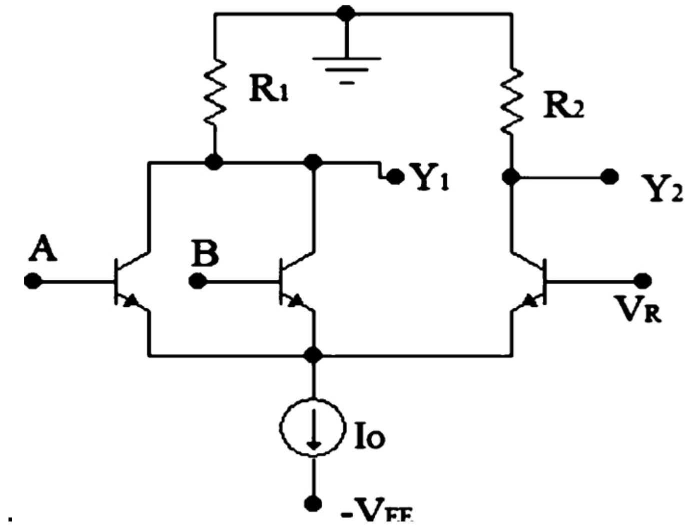
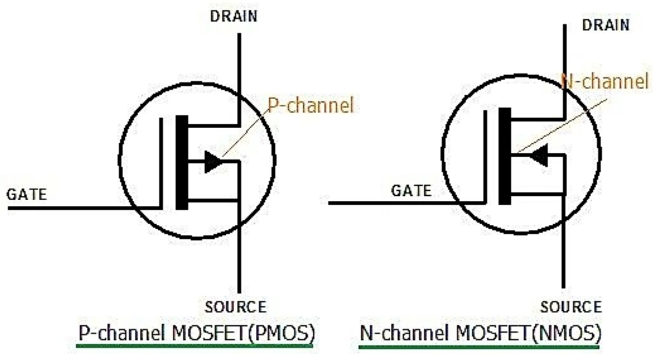


Fig. 1.34: Emitter coupled Logic for NOR/OR function

(7) CMOS (Complementary Metal Oxide Semiconductor) Logic

Complementary metal–oxide semiconductor CMOS is a technology for constructing integrated circuits. CMOS technology is used in microprocessors, microcontrollers, static RAM, and other digital logic circuits. CMOS technology is also used for several analog circuits such as image sensors (CMOS sensor), data converters, and highly integrated transceivers for many types of communication.

Two important characteristics of CMOS devices are high noise immunity and low static power consumption. CMOS became the most used technology to be implemented in very-large-scale integration (VLSI) chips. (e.g. the 4000 series of integrated circuits).



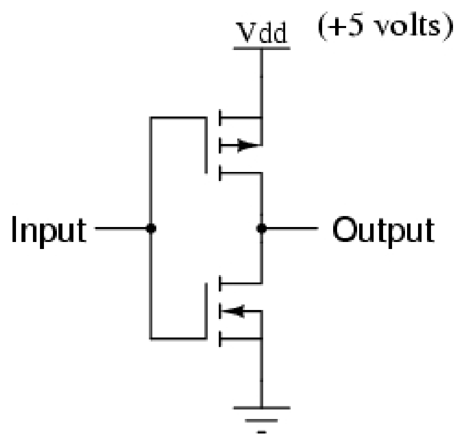


Fig. 1.35: CMOS Inverter/NOT function

1.9 Important terms related to Logic Families

(1) Fan-In:

Maximum number of inputs that can be applied to a single logic gate is known as it's Fan-In. OR it is maximum number of inputs a gate can handle.

(2) Fan-Out

It is the number of similar Gates which can be driven by a single logic gate. High Fan-out is advantageous/preferred.

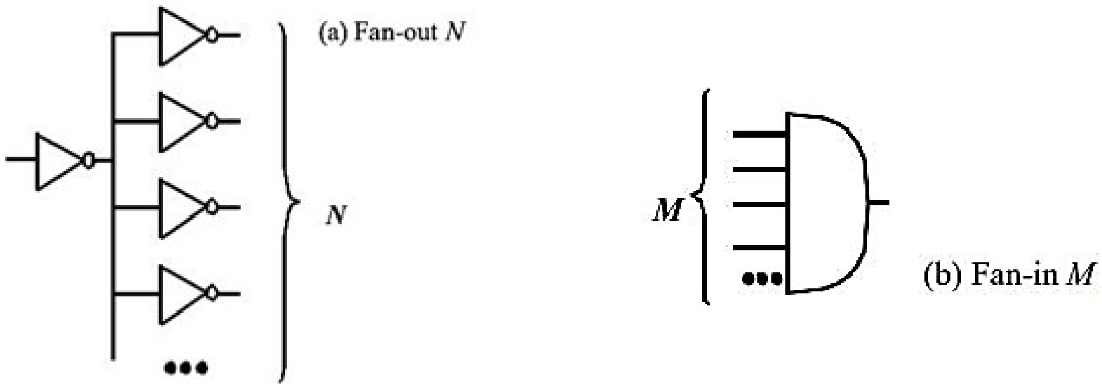


Fig. 1.36: Fan-out and fan-in

(3) Noise Margin

The circuit's ability to tolerate noise signal is referred to as the noise immunity. A quantitative measure of noise immunity is known as Noise Margin.

Noise margin specifies the maximum amplitude of noise pulse that will not change the state of the driven stage/logic gate. Noise margin can be evaluated from a consideration of the voltage levels V_{IHmin}, V_{ILmax}, V_{OHmin} and V_{OLmax}

$V_{ILmax} - V_{OLmax}$ is called **Low Level Noise Margin**
$V_{OHmin} - V_{IHmin}$ is called **High Level Noise Margin**

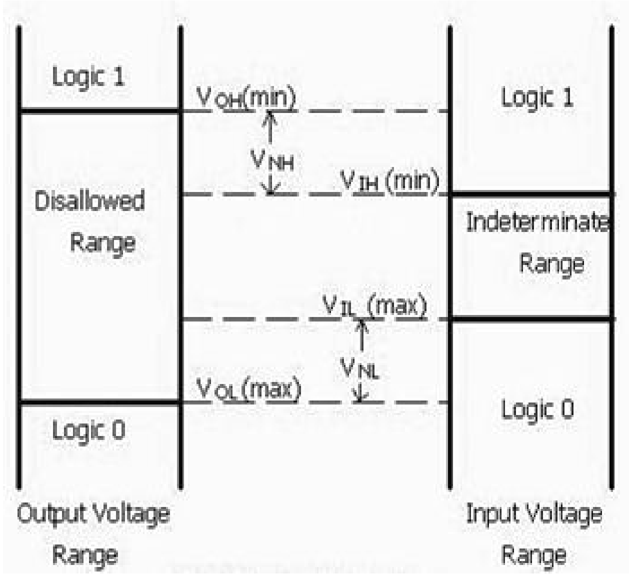


Fig. 1.36: Noise Margin Definition

(4) Propagation Delay Time

It is the time interval between application of data input signal and resulting change in output signal.

(5) Setup Time

It is the minimum time for which the control levels (Input signals) need to be maintained constant at input terminal of logic circuit prior to the arrival of triggering edge of clock pulse.

(6) Hold Time

It is the minimum time for which the control levels (Input signals) need to be maintained constant at input terminal of logic circuit after the arrival of triggering edge of clock pulse.

(7) Switching Time

Another quantity that is used to characterize switching circuits is the speed with which the device responds to the input changes.

For switching circuits, the graph of Fig. 1.37 shows different delay times.
Different delay times are of an inverter gate.

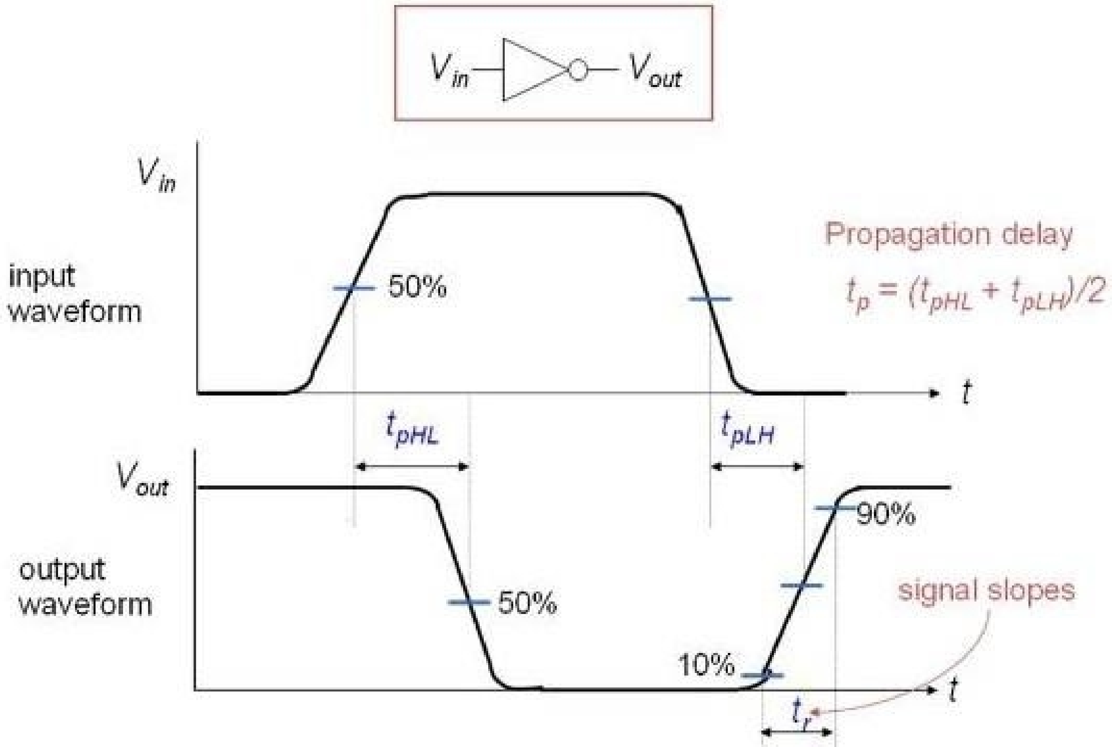


Fig. 1.37: Representation of switching time

Different definitions are as follows:

- **t_{pHL}** (propagation delay time for high to low transition): Difference in time between the point where V_{in} rises to 50% of its final value and the time when V_{out} falls to 50% level from its maximum value.

- **t_{pLH}** (propagation delay time for low to high transition): Time difference between 50% points of the trailing edges of the input and output signals
- **Propagation Delay** : The average of t_{pHL} and t_{pLH}, or

$$t_{pd} = \frac{t_{pHL} + t_{pLH}}{2}$$

- **Fall Time, t_f** : It is the time during which output voltage swing between upper and lower voltage level from 90% to 10% value. (OR time during which signal falls from 90% to 10% value)
- **Rise Time, t_r** : It is the time during which output voltage swing between lower and upper voltage level from 10% to 90% of its maximum value. (OR time during which signal rises from 10% to 90% value)

1.10 comparison of different Logic Families

Parameter	**RTL**	**DTL**	**TTL**	**I²L**	**ECL**	**CMOS**
Basic Gate	NOR	NAND	NAND	NOR	NOR/OR	NOR/NAND
Power dissipation in mW per gate	12	8-12	10	6 nW – 70 µW	40-55	1
Fan Out	5	8	10	Depends on injector current	25	50
Noise Immunity	Normal	Good	Very Good	Poor	Poor	Very Good
Propagation delay in nS	12	30	10	25-250	2	70

Exercise: Question Bank_Unit-6

		Marks
Q:1	Enlist the difference between Analog and Digital Systems	**4**
Q:2	Enlist the advantages of Digital system over Analog system	**4**
Q:3	State & Prove De-Morgan's Theorem.	**4**
Q:4	***Explain Basic Logic Gates with necessary Symbol, Truth Table, Logic circuit diagram and it's Boolean Equation.*** ***OR*** ***Draw Symbol, Truth Table and write Boolean equation of NOT gate, OR gate as well as AND gate.***	***7***
Q:5	***Draw the circuit diagram of AND gate using Diode and explain it's working with necessary truth table.***	***4***
Q:6	***Draw the circuit diagram of OR gate using Diode and explain it's working with necessary truth table.***	***4***
Q:7	***Draw the Logic circuit diagram of EX-OR and EX-NOR gate using Basic Logic gates and explain it's working with necessary Symbol, Truth table and Boolean equation***	***7***
Q:8	Explain Universal Logic Gates in detail with necessary circuit diagram, Symbol, Truth Table and its Boolean equation. **OR** Explain NAND as well as NOR gate in detail with necessary circuit diagram, Symbol, Truth Table and its Boolean equation.	**7**
Q:9	***Prove that NAND gate and NOR gate are Universal Logic gates.*** ***OR*** ***Design NOT gate, AND gate, OR gate as well as NOR gate using NAND gate only.*** ***OR*** ***Using NOR gate only, derive Basic Logic Gates.***	***7***
Q:10	Draw the circuit diagram of NOT, AND, OR, NAND, NOR gate using Transistor only.	**7**
Q:11	Explain different Logic Families in detail **OR** Enlist the name of different Logic Families and explain any two in detail. **OR** Give comparison between Logic Families	**7**
Q:12	***Write short note on TTL logic family.*** ***OR*** ***Draw TTL NAND gate and explain it.***	***7***
Q:13	***Draw the circuit diagram of DTL NAND gate and explain.***	***4***

14	Define following terms (a) Fan-out (b) Fan-in (c) Noise Margin (d) Propagation Delay Time	**4**