ANALOG AND DIGITAL ELECTRONICS

II B.TECH I SEMESTER

(COMMON FOR CSE & IT)

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

MALLA REDDY COLLEGE OF ENGINEERING & TECHNOLOGY

Autonomous Institution – UGC, Govt. of India

(Affiliated to JNTU, Hyderabad, Approved by AICTE - Accredited by NBA & NAAC – 'A' Grade - ISO 9001:2008 Certified)

Maisammaguda, Dhulapally (Post Via Hakimpet), Secunderabad – 500100

PREPARED BY

Mr E.MAHENDAR REDDY, Mr V SHIVA RAJ KUMAR, Ms K.BHAVANA, Mr M ANANTHA GUPTHA

MALLA REDDY COLLEGE OF ENGINEERING AND TECHNOLOGY

II Year B.Tech IT–I Sem

L	T/P/D	C
4	-/-/-	4

(R18A0401) ANALOG & DIGITAL ELECTRONICS

OBJECTIVES

The main objectives of the course are:

1. To familiarize the student with the principal of operation, analysis and design of junction diode and BJT.
2. To understand basic number systems codes and logical gates.
3. To introduce the methods for simplifying Boolean expressions
4. To outline the formal procedures for the analysis and design of combinational circuits and sequential circuits

UNIT-I

P-N Junction diode: Qualitative Theory of P-N Junction, P-N Junction as a diode , diode equation, volt-ampere characteristics temperature dependence of V-I characteristic , ideal versus practical, diode equivalent circuits,, Zener diode characteristics.

UNIT-II

BIPOLAR JUNCTION TRANSISTOR: The Junction transistor, Transistor construction ,Transistor current components, Transistor as an amplifier, Input and Output characteristics of transistor in Common Base, Common Emitter, and Common collector configurations. α and β Parameters and the relation between them, BJT Specifications.

UNIT-III

Number System and Boolean Algebra: Number Systems, Base Conversion Methods, Complements of Numbers, Codes- Binary Codes, Binary Coded Decimal, Unit Distance Code,Digital Logic Gates (AND, NAND, OR, NOR, EX-OR, EX-NOR), Properties of XOR Gates, Universal Gates, Basic Theorems and Properties, Switching Functions, Canonical and Standard Form.

UNIT-IV

Minimization Techniques:

The Karnaugh Map Method, Three, Four and Five Variable Maps, Prime and Essential Implications, Don’t Care Map Entries, Using the Maps for Simplifying, Multilevel NAND/NOR realizations.

UNIT-V

Combinational Circuits:

Design procedure – Half adder, Full Adder, Half subtractor, Full subtractor, Multiplexer/Demultiplexer, decoder, encoder, Code converters, Magnitude Comparator.

Sequential circuits:

Latches, Flip-Flops-SR, JK, D, T and master slave, characteristic tables and equations, Conversion from one type of Flip-Flop to another.

TEXT BOOKS:

1. "Electronic Devices & Circuits", Special Edition – MRCET, McGraw Hill Publications, 2017.
2. Integrated Electronics Analog Digital Circuits, Jacob Millman and D. Halkias, McGraw Hill.
3. Electronic Devices and Circuits, S.Salivahanan,N.Suresh kumar, McGraw Hill.
4. M. Morris Mano, Digital Design, 3rd Edition, Prentice Hall of India Pvt. Ltd., 2003 /Pearson Education (Singapore) Pvt. Ltd., New Delhi, 2003.
5. Switching and Finite Automata Theory- Zvi Kohavi & Niraj K. Jha, 3rd Edition, Cambridge.

REFERENCE BOOKS:

1. Electronic Devices and Circuits,K.Lal Kishore B.S Publications
2. Electronic Devices and Circuits, G.S.N. Raju, I.K. International Publications, New Delhi, 2006.
3. John F.Wakerly, Digital Design, Fourth Edition, Pearson/PHI, 2006
4. John.M Yarbrough, Digital Logic Applications and Design, Thomson Learning, 2002.
5. Charles H.Roth. Fundamentals of Logic Design, Thomson Learning, 2003.

OUTCOMES:

After completion of the course, the student will be able to:

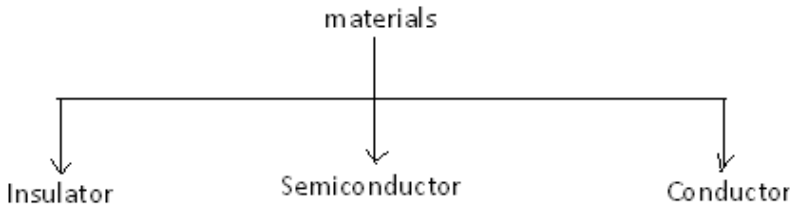
1. Understand and Analyze the different types of diodes, operation and its characteristics
2. Understand and analyze the BJT Transistor.
3. Understand the basic postulates of Boolean algebra and shows the correlation between Boolean expressions
4. Learn the methods for simplifying Boolean expressions
5. Understand the formal procedures for the analysis and design of combinational circuits and sequential circuits

UNIT-I

PN JUNCTION DIODE

1.0 INTRODUCTON

Based on the electrical conductivity all the materials in nature are classified as insulators, semiconductors, and conductors.

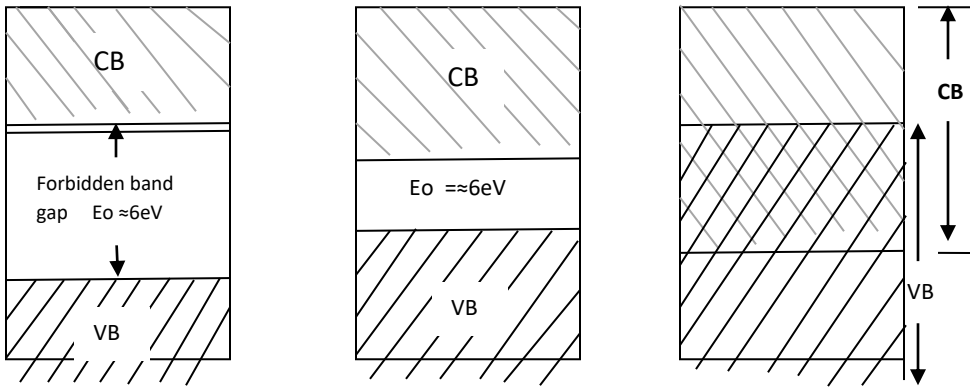


Insulator: An insulator is a material that offers a very low level (or negligible) of conductivity when voltage is applied. Eg: Paper, Mica, glass, quartz. Typical resistivity level of an insulator is of the order of 10^{10} to 10^{12} Ω-cm. The energy band structure of an insulator is shown in the fig.1.1. Band structure of a material defines the band of energy levels that an electron can occupy. Valance band is the range of electron energy where the electron remain bended too the atom and do not contribute to the electric current. Conduction bend is the range of electron energies higher than valance band where electrons are free to accelerate under the influence of external voltage source resulting in the flow of charge.

The energy band between the valance band and conduction band is called as forbidden band gap. It is the energy required by an electron to move from balance band to conduction band i.e. the energy required for a valance electron to become a free electron.

$$1 \text{ eV} = 1.6 \times 10^{-19} \text{ J}$$

For an insulator, as shown in the fig.1.1 there is a large forbidden band gap of greater than 5Ev. Because of this large gap there a very few electrons in the CB and hence the conductivity of insulator is poor. Even an increase in temperature or applied electric field is insufficient to transfer electrons from VB to CB.



Insulator Semiconductor Conductor

FiG:1.1 Energy band diagrams insulator, semiconductor and conductor

Conductors: A conductor is a material which supports a generous flow of charge when a voltage is applied across its terminals. i.e. it has very high conductivity. Eg: Copper, Aluminum, Silver, Gold. The resistivity of a conductor is in the order of 10^{-4} and 10^{-6} Ω-cm. The Valance and conduction bands overlap (fig1.1) and there is no energy gap for the electrons to move from valance band to conduction band. This implies that there are free electrons in CB even at absolute zero temperature (0K). Therefore at room temperature when electric field is applied large current flows through the conductor.

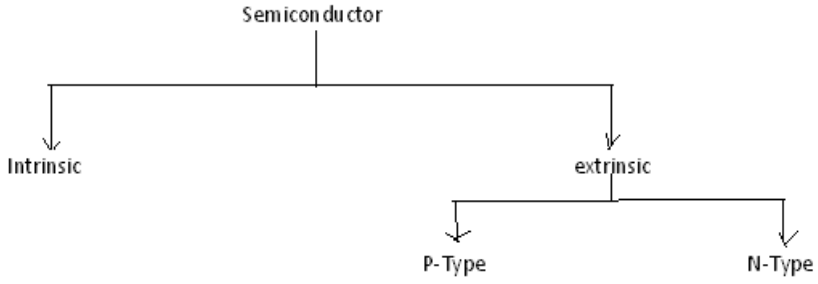
Semiconductor: A semiconductor is a material that has its conductivity somewhere between the insulator and conductor. The resistivity level is in the range of 10 and 10^4 Ω-cm. Two of the most commonly used are Silicon (Si=14 atomic no.) and germanium (Ge=32 atomic no.). Both have 4 valance electrons. The forbidden band gap is in the order of 1eV. For eg., the band gap energy for Si, Ge and GaAs is 1.21, 0.785 and 1.42 eV, respectively at absolute zero temperature (0K). At 0K and at low temperatures, the valance band electrons do not have sufficient energy to move from V to CB. Thus semiconductors act a insulators at 0K. as the temperature increases, a large number of valance electrons acquire sufficient energy to leave the VB, cross the forbidden band gap and reach CB. These are now free electrons as they can move freely under the influence of electric field. At room temperature there are sufficient electrons in the CB and hence the semiconductor is capable of conducting some current at room temperature.

Inversely related to the conductivity of a material is its resistance to the flow of charge or current. Typical resistivity values for various materials' are given as follows.

Insulator	Semiconductor	Conductor
10^{-6} Ω-cm (Cu)	50Ω-cm (Ge)	10^{12} Ω-cm (mica)
	$50x10^3$ Ω-cm (Si)	

Typical resistivity values

1.0.1 Semiconductor Types



A pure form of semiconductors is called as intrinsic semiconductor. Conduction in intrinsic sc is either due to thermal excitation or crystal defects. Si and Ge are the two most important semiconductors used. Other examples include Gallium arsenide GaAs, Indium Antimonide (InSb) etc.

Let us consider the structure of Si. A Si atomic no. is 14 and it has 4 valance electrons. These 4 electrons are shared by four neighboring atoms in the crystal structure by means of covalent bond. Fig. 1.2a shows the crystal structure of Si at absolute zero temperature (0K). Hence a pure SC acts has poor conductivity (due to lack of free electrons) at low or absolute zero temperature.

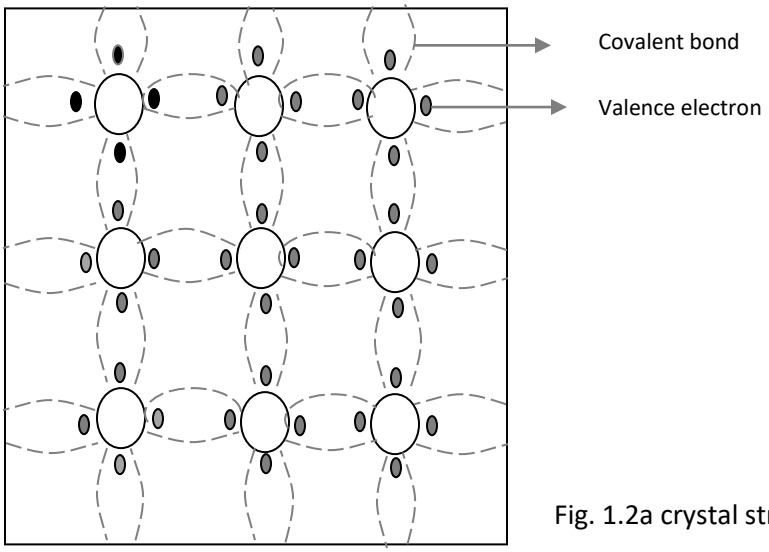


Fig. 1.2a crystal structure of Si at 0K

At room temperature some of the covalent bonds break up to thermal energy as shown in fig 1.2b. The valance electrons that jump into conduction band are called as free electrons that are available for conduction.

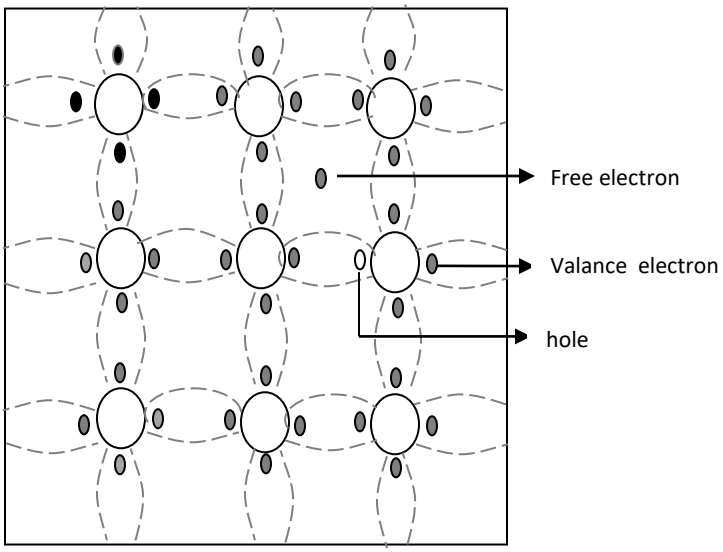
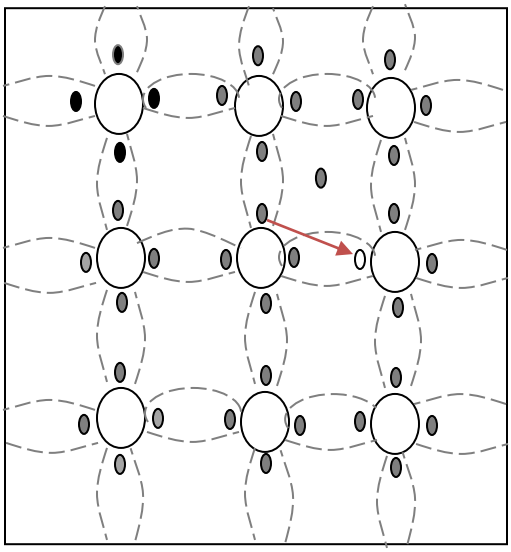


Fig. 1.2b crystal structure of Si at room temperature0K

The absence of electrons in covalent bond is represented by a small circle usually referred to as hole which is of positive charge. Even a hole serves as carrier of electricity in a manner similar to that of free electron.

The mechanism by which a hole contributes to conductivity is explained as follows:

When a bond is in complete so that a hole exists, it is relatively easy for a valance electron in the neighboring atom to leave its covalent bond to fill this hole. An electron moving from a bond to fill a hole moves in a direction opposite to that of the electron. This hole, in its new position may now be filled by an electron from another covalent bond and the hole will correspondingly move one more step in the direction opposite to the motion of electron. Here we have a mechanism for conduction of electricity which does not involve free electrons. This phenomenon is illustrated in fig1.3



Electron movement

Hole movement

Fig. 1.3a

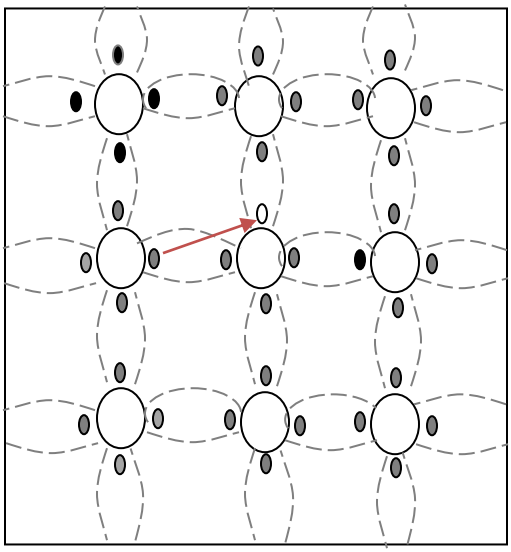


Fig. 1.3b

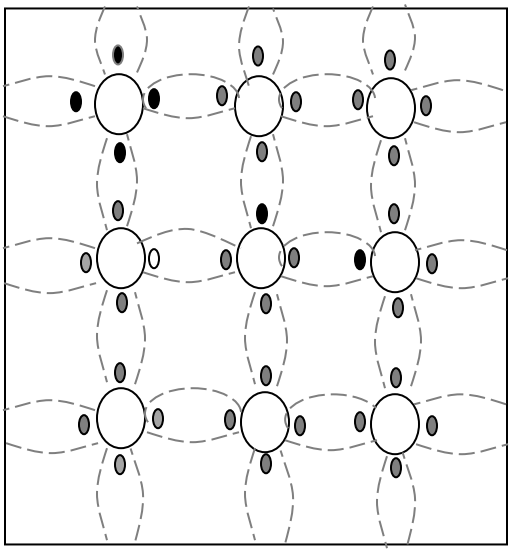


Fig. 1.3c

Fig 1.3a show that there is a hole at ion 6.Imagine that an electron from ion 5 moves into the hole at ion 6 so that the configuration of 1.3b results. If we compare both fig1.3a &fig 1.3b, it appears as if the hole has moved towards the left from ion6 to ion 5. Further if we compare fig 1.3b and fig 1.3c, the hole moves from ion5 to ion 4. This discussion indicates the motion of hole is in a direction opposite to that of motion of electron. Hence we consider holes as physical entities whose movement constitutes flow of current.

In a pure semiconductor, the number of holes is equal to the number of free electrons.

1.0.2 EXTRINSIC SEMICONDUCTOR

Intrinsic semiconductor has very limited applications as they conduct very small amounts of current at room temperature. The current conduction capability of intrinsic semiconductor can be increased significantly by adding a small amounts impurity to the intrinsic semiconductor. By adding impurities it becomes impure or extrinsic semiconductor. This process of adding impurities is called as doping. The amount of impurity added is 1 part in 10^6 atoms.

N type semiconductor: If the added impurity is a pentavalent atom then the resultant semiconductor is called N-type semiconductor. Examples of pentavalent impurities are Phosphorus, Arsenic, Bismuth, Antimony etc.

A pentavalent impurity has five valance electrons. Fig 1.4a shows the crystal structure of N-type semiconductor material where four out of five valance electrons of the impurity atom(antimony) forms covalent bond with the four intrinsic semiconductor atoms. The fifth electron is loosely bound to the impurity atom. This loosely bound electron can be easily

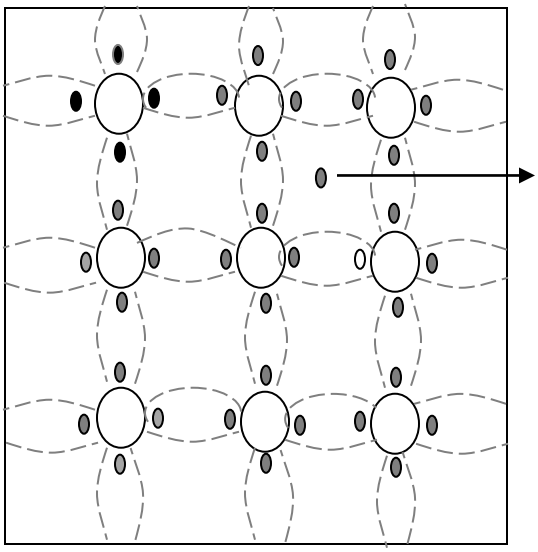


Fig. 1.4a crystal structure of N type SC

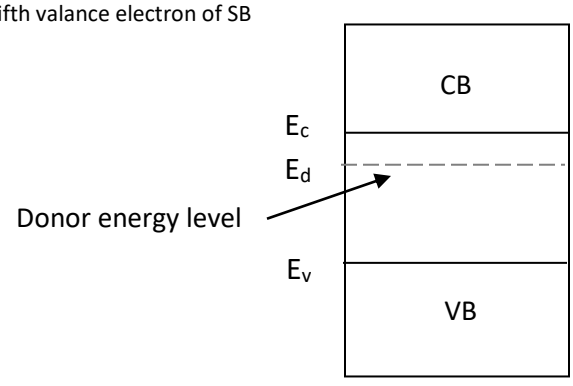


Fig. 1.4bEnergy band diagram of N type

Excited from the valance band to the conduction band by the application of electric field or increasing the thermal energy. The energy required to detach the fifth electron form the impurity atom is very small of the order of 0.01ev for Ge and 0.05 eV for Si.

The effect of doping creates a discrete energy level called donor energy level in the forbidden band gap with energy level E_d slightly less than the conduction band (fig 1.4b). The difference between the energy levels of the conducting band and the donor energy level is the energy required to free the fifth valance electron (0.01 eV for Ge and 0.05 eV for Si). At room temperature almost all the fifth electrons from the donor impurity atom are raised to conduction band and hence the number of electrons in the conduction band increases significantly. Thus every antimony atom contributes to one conduction electron without creating a hole.

In the N-type sc the no. of electrons increases and the no. of holes decreases compared to those available in an intrinsic sc. The reason for decrease in the no. of holes is that the larger no. of electrons present increases the recombination of electrons with holes. Thus current in N type sc is dominated by electrons which are referred to as majority carriers. Holes are the minority carriers in N type sc

P type semiconductor: If the added impurity is a trivalent atom then the resultant semiconductor is called P-type semiconductor. Examples of trivalent impurities are Boron, Gallium , indium etc.

The crystal structure of p type sc is shown in the fig1.5a. The three valance electrons of the impurity (boon) forms three covalent bonds with the neighboring atoms and a vacancy exists in the fourth bond giving rise to the holes. The hole is ready to accept an electron from the neighboring atoms. Each trivalent atom contributes to one hole generation and thus introduces a large no. of holes in the valance band. At the same time the no. electrons are decreased compared to those available in intrinsic sc because of increased recombination due to creation of additional holes.

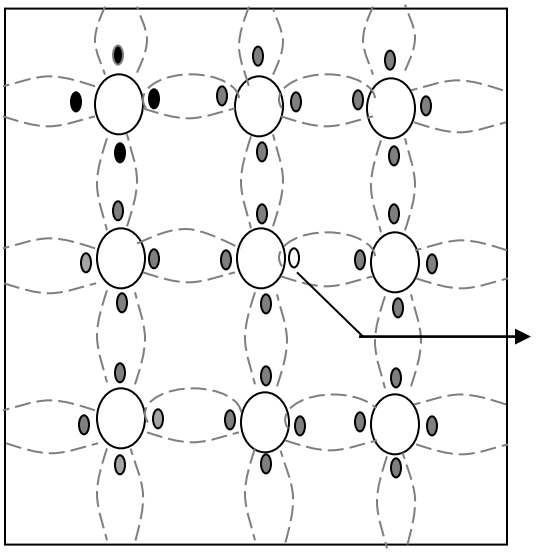


Fig. 1.5a crystal structure of P type sc

Thus in P type sc , holes are majority carriers and electrons are minority carriers. Since each trivalent impurity atoms are capable accepting an electron, these are called as acceptor atoms. The following fig 1.5b shows the pictorial representation of P type sc

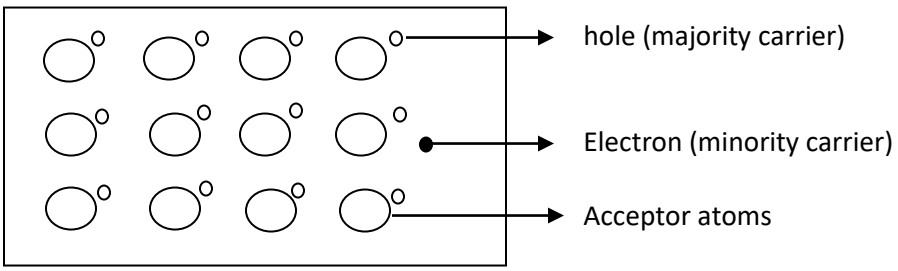


Fig. 1.5b crystal structure of P type sc

- The conductivity of N type sc is greater than that of P type sc as the mobility of electron is greater than that of hole.
- For the same level of doping in N type sc and P type sc, the conductivity of an Ntype sc is around twice that of a P type sc

1.0.3 CONDUCTIVITY OF SEMICONDUCTOR

In a pure sc, the no. of holes is equal to the no. of electrons. Thermal agitation continue to produce new electron- hole pairs and the electron hole pairs disappear because of recombination. with each electron hole pair created , two charge carrying particles are formed . One is negative which is a free electron with mobility μ_n . The other is a positive i.e., hole with mobility μ_p . The electrons and hole move in opppsitte direction in a an electric field E, but since they are of opposite sign, the current due to each is in the same direction. Hence the total current density J within the intrinsic sc is given by

$J = J_n + J_p$

$=q\, n\, \mu_n E + q\, p\, \mu_p E$

$= (n\, \mu_n + p\, \mu_p) qE$

$=\sigma E$

Where n=no. of electrons / unit volume i.e., concentration of free electrons

P= no. of holes / unit volume i.e., concentration of holes

E=applied electric field strength, V/m

q= charge of electron or hole I n Coulombs

Hence, σ is the conductivity of sc which is equal to $(n\, \mu_n + p\, \mu_p)q$. he resistivity of sc is reciprocal of conductivity.

$P = 1/\sigma$

It is evident from the above equation that current density with in a sc is directly proportional to applied electric field E.

For pure sc, n=p= n_i where n_i = intrinsic concentration. The value of n_i is given by

$n_i^2=AT^3 \exp(-E_{GO}/KT)$

therefore, $J= n_i (\mu_n + \mu_p) q E$

Hence conductivity in intrinsic sc is $\sigma i= n_i (\mu_n + \mu_p) q$

Intrinsic conductivity increases at the rate of 5% per ^{o} C for Ge and 7% per ^{o} C for Si.

Conductivity in extrinsic sc (N Type and P Type):

The conductivity of intrinsic sc is given by $\sigma i= n_i (\mu_n + \mu_p) q = (n \mu_n + p \mu_p)q$

For N type , n>>p

Therefore $\sigma= q n \mu_n$

For P type ,p>>n

Therefore $\sigma= q p \mu_p$

1.0.4 CHARGE DENSITIES IN P TYPE AND N TYPE SEMICONDUCTOR:

Mass Action Law:

Under thermal equilibrium for any semiconductor, the product of the no. of holes and the concentration of electrons is constant and is independent of amount of donor and acceptor impurity doping.

$n.p= n_i^2$

where n= eleetron concentration

p = hole concentration

n_i^2= intrinsic concentration

Hence in N type sc , as the no. of electrons increase the no. of holes decreases. Similarly in P type as the no. of holes increases the no. of electrons decreases. Thus the product is constant and is equal to n_i^2 in case of intrinsic as well as extrinsic sc.

The law of mass action has given the relationship between free electrons concentration and hole concentration. These concentrations are further related by the law of electrical neutrality as explained below.

Law of electrical neutrality:

Sc materials are electrically neutral. According to the law of electrical neutrality, in an electrically neutral material, the magnitude of positive charge concentration is equal to tat of negative charge concentration. Let us consider a sc that has N_D donor atoms per cubic centimeter and N_A acceptor atoms per cubic centimeter i.e., the concentration of donor and acceptor atoms are N_D and N_A respectively. Therefore N_D positively charged ions per cubic centimeter are contributed by donor atoms and N_A negatively charged ions per cubic centimeter are contributed by the acceptor atoms. Let n, p is concentration of free electrons and holes respectively. Then according to the law of neutrality

$N_D + p = N_A + n$ ……eq 1.1

For N type sc, $N_A = 0$ and $n >> p$. Therefore $N_D \approx n$ ……eq 1.2

Hence for N type sc the free electron concentration is approximately equal to the concentration of donor atoms. In later applications since some confusion may arise as to which type of sc is under consideration a the given moment, the subscript n or p is added for Ntype or P type respectively. Hence eq1.2 becomes $N_D \approx n_n$

Therefore current density in N type sc is $J = N_D \mu_n q E$

And conductivity $\sigma = N_D \mu_n q$

For P type sc, $N_D = 0$ and $p >> n$. Therefore $N_A \approx p$

Or $N_A \approx p_p$

Hence for P type sc the hole concentration is approximately equal to the concentration of acceptor atoms.

Therefore current density in N type sc is $J = N_A \mu_p q E$

And conductivity $\sigma = N_A \mu_p q$

Mass action law for N type, $n_n p_n = n_i^2$

$p_n = n_i^2 / N_D$ since ($n_n \approx N_D$)

Mass action law for P type, $n_p p_p = n_i^2$

$n_p = n_i^2 / N_A$ since ($p_p \approx N_A$)

1.1 QUANTITATIVE THEORY OF PN JUNCTION DIODE

1.1.1 PN JUNCTION WITH NO APPLIED VOLTAGE OR OPEN CIRCUIT CONDITION:

In a piece of sc, if one half is doped by p type impurity and the other half is doped by n type impurity, a PN junction is formed. The plane dividing the two halves or zones is called PN junction. As

shown in the fig the n type material has high concentration of free electrons, while p type material has high concentration of holes. Therefore at the junction there is a tendency of free electrons to diffuse over to the P side and the holes to the N side. This process is called diffusion. As the free electrons move across the junction from N type to P type, the donor atoms become positively charged. Hence a positive charge is built on the N-side of the junction. The free electrons that cross the junction uncover the negative acceptor ions by filing the holes. Therefore a negative charge is developed on the p –side of the junction..This net negative charge on the p side prevents further diffusion of electrons into the p side. Similarly the net positive charge on the N side repels the hole crossing from p side to N side. Thus a barrier sis set up near the junction which prevents the further movement of charge carriers i.e. electrons and holes. As a consequence of induced electric field across the depletion layer, an electrostatic potential difference is established between P and N regions, which are called the potential barrier, junction barrier, diffusion potential or contact potential, Vo. The magnitude of the contact potential Vo varies with doping levels and temperature. Vo is 0.3V for Ge and 0.72 V for Si.

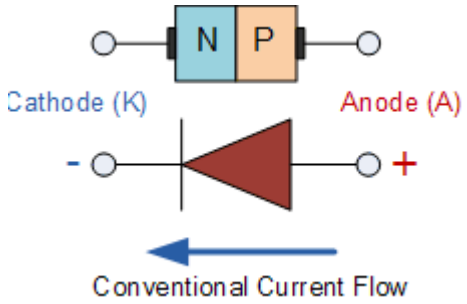


Fig 1.6: Symbol of PN Junction Diode

The electrostatic field across the junction caused by the positively charged N-Type region tends to drive the holes away from the junction and negatively charged p type regions tend to drive the electrons away from the junction. The majority holes diffusing out of the P region leave behind negatively charged acceptor atoms bound to the lattice, thus exposing a negatives pace charge in a previously neutral region. Similarly electrons diffusing from the N region expose positively ionized donor atoms and a double space charge builds up at the junction as shown in the fig. 1.7a

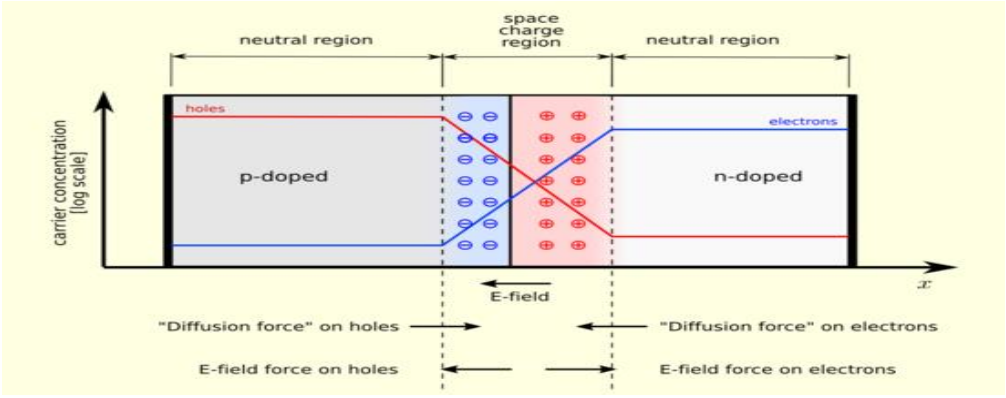


Fig 1.7a

It is noticed that the space charge layers are of opposite sign to the majority carriers diffusing into them, which tends to reduce the diffusion rate. Thus the double space of the layer causes an electric field to be set up across the junction directed from N to P regions, which is in such a direction to inhibit the diffusion of majority electrons and holes as illustrated in fig 1.7b. The shape of the charge density, ρ, depends upon how diode id doped. Thus the junction region is depleted of mobile charge carriers. Hence it is called depletion layer, space region, and transition region. The depletion region is of the order of 0.5µm thick. There are no mobile carriers in this narrow depletion region. Hence no current flows across the junction and the system is in equilibrium. To the left of this depletion layer, the carrier concentration is p= N_A and to its right it is n= N_D.

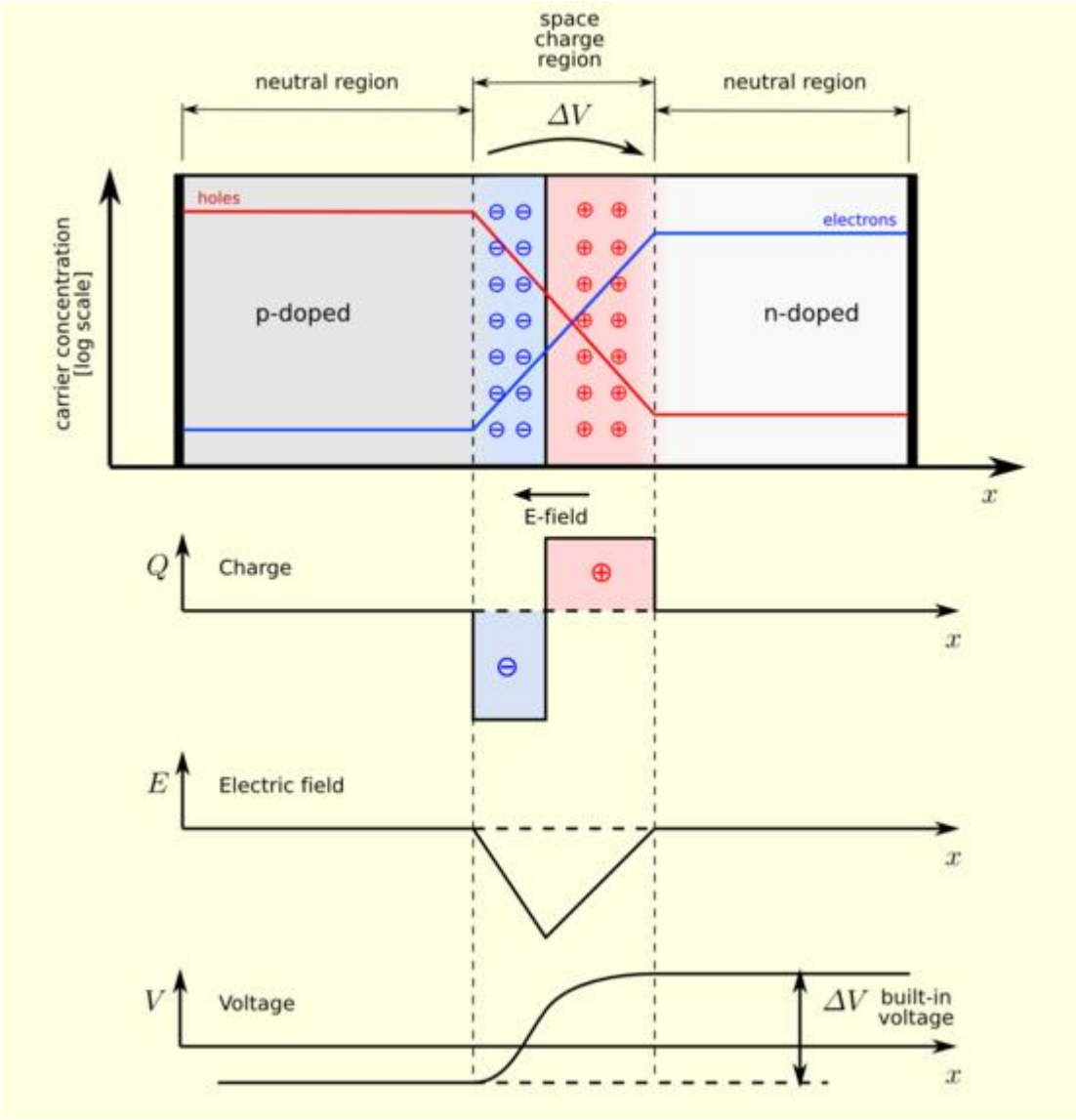


Fig 1.7b

1.1.2 FORWARD BIASED JUNCTION DIODE

When a diode is connected in a **Forward Bias** condition, a negative voltage is applied to the N-type material and a positive voltage is applied to the P-type material. If this external voltage becomes greater than the value of the potential barrier, approx. 0.7 volts for silicon and 0.3 volts for germanium, the potential barriers opposition will be overcome and current will start to flow. This is because the negative voltage pushes or repels electrons towards the junction giving them the energy to cross over and combine with the holes being pushed in the opposite direction towards the junction by the positive voltage. This results in a characteristics curve of zero current flowing up to this voltage point, called the "knee" on the static curves and then a high current flow through the diode with little increase in the external voltage as shown below.

Forward Characteristics Curve for a Junction Diode

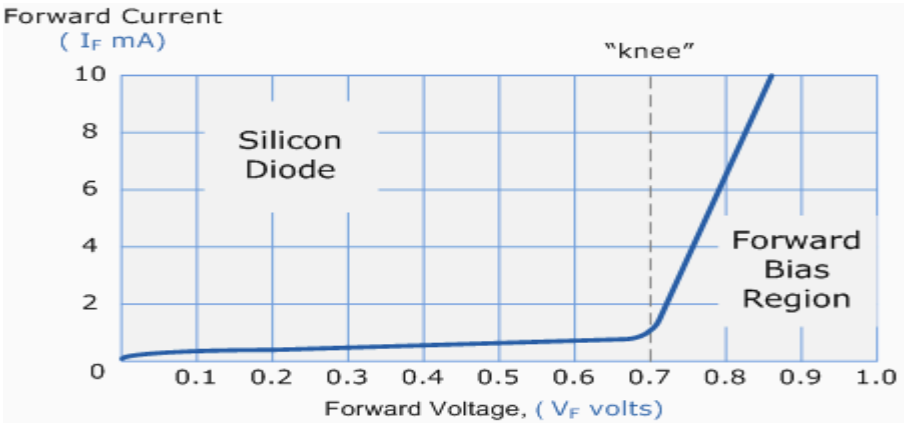


Fig 1.8a: Diode Forward Characteristics

The application of a forward biasing voltage on the junction diode results in the depletion layer becoming very thin and narrow which represents a low impedance path through the junction thereby allowing high currents to flow. The point at which this sudden increase in current takes place is represented on the static I-V characteristics curve above as the "knee" point.

Forward Biased Junction Diode showing a Reduction in the Depletion Layer

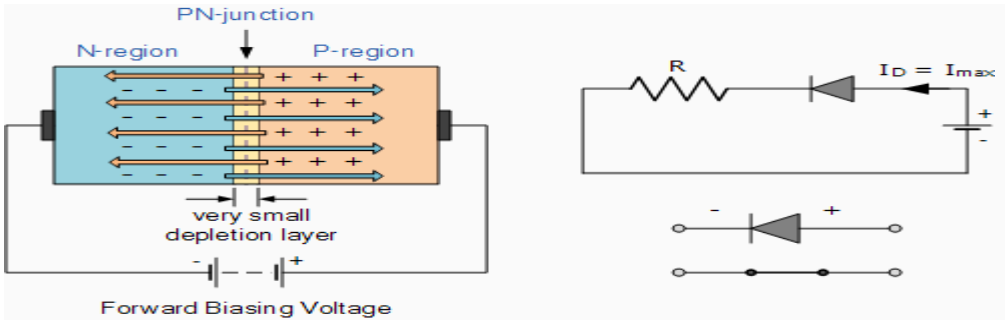


Fig 1.8b: Diode Forward Bias

This condition represents the low resistance path through the PN junction allowing very large currents to flow through the diode with only a small increase in bias voltage. The actual potential difference across the junction or diode is kept constant by the action of the depletion layer at approximately 0.3v for germanium and approximately 0.7v for silicon junction diodes. Since the diode can conduct "infinite" current above this knee point as it effectively becomes a short circuit, therefore resistors are used in series with the diode to limit its current flow. Exceeding its maximum forward current specification causes the device to dissipate more power in the form of heat than it was designed for resulting in a very quick failure of the device.

1.1.2 PN JUNCTION UNDER REVERSE BIAS CONDITION:

Reverse Biased Junction Diode

When a diode is connected in a **Reverse Bias** condition, a positive voltage is applied to the N-type material and a negative voltage is applied to the P-type material. The positive voltage applied to the N-type material attracts electrons towards the positive electrode and away from the junction, while the holes in the P-type end are also attracted away from the junction towards the negative electrode. The net result is that the depletion layer grows wider due to a lack of electrons and holes and presents a high impedance path, almost an insulator. The result is that a high potential barrier is created thus preventing current from flowing through the semiconductor material.

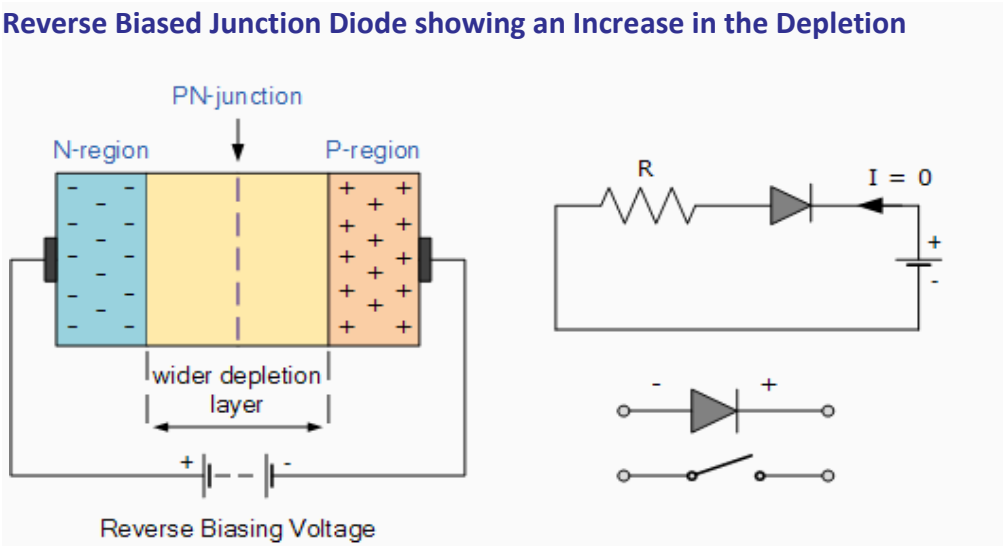


Fig 1.9a: Diode Reverse Bias

This condition represents a high resistance value to the PN junction and practically zero current flows through the junction diode with an increase in bias voltage. However, a very small **leakage current** does flow through the junction which can be measured in microamperes, (μA). One final point, if the reverse bias voltage Vr applied to the diode is increased to a sufficiently high enough value, it will cause the PN junction to overheat and fail due to the avalanche effect around the junction. This may

cause the diode to become shorted and will result in the flow of maximum circuit current, and this shown as a step downward slope in the reverse static characteristics curve below.

Reverse Characteristics Curve for a Junction Diode

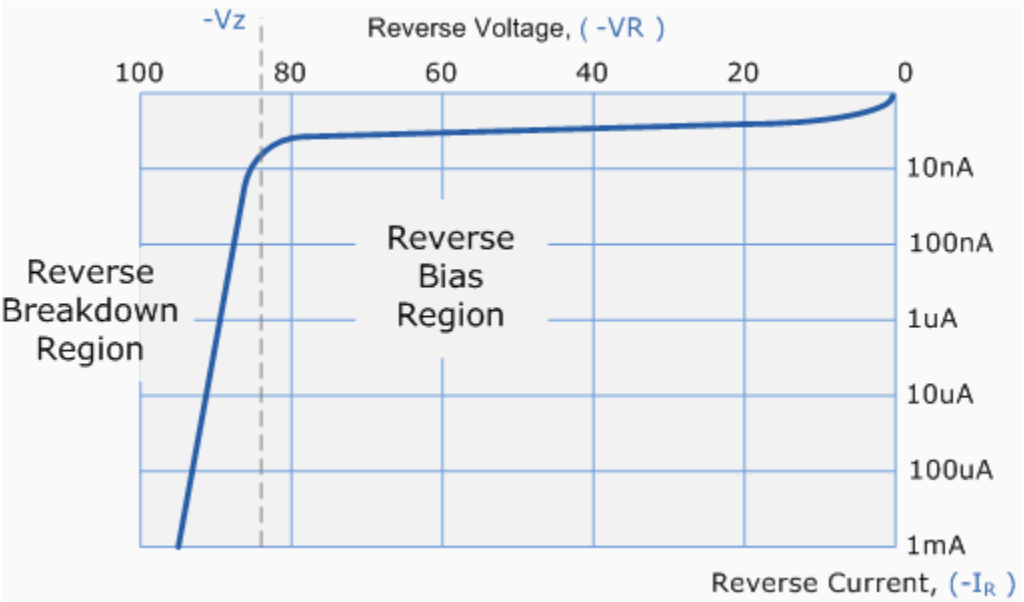


Fig 1.9b: Diode Reverse Characteristics

Sometimes this avalanche effect has practical applications in voltage stabilizing circuits where a series limiting resistor is used with the diode to limit this reverse breakdown current to a preset maximum value thereby producing a fixed voltage output across the diode. These types of diodes are commonly known as **Zener Diodes**

1.2 VI CHARACTERISTICS AND THEIR TEMPERATURE DEPENDENCE

Diode terminal characteristics equation for diode junction current:

$$I_D = I_0 (e^{\frac{v}{\eta v_T}} - 1)$$

Where V_T = KT/q;

V_D_ diode terminal voltage, Volts

I_o _ temperature-dependent saturation current, µA

T _ absolute temperature of p-n junction, K

K _ Boltzmann's constant 1.38x 10 -23J/K)

q _ electron charge 1.6x10-19 C

η = empirical constant, 1 for Ge and 2 for Si

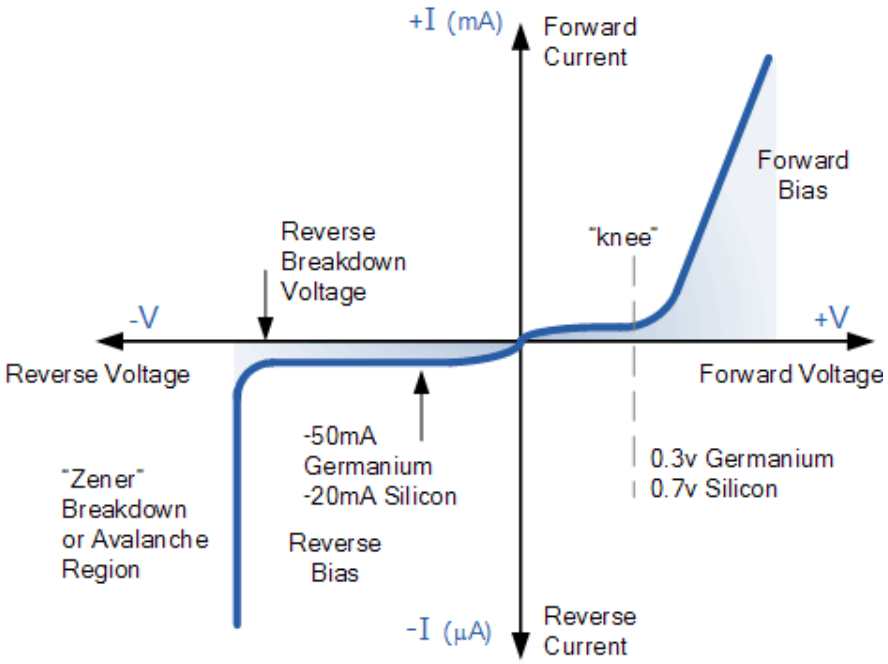


Fig 1.10: Diode Characteristics

Temperature Effects on Diode

Temperature can have a marked effect on the characteristics of a silicon semiconductor diode as shown in Fig. 11 It has been found experimentally that the reverse saturation current Io will just about double in magnitude for every 10°C increase in temperature.

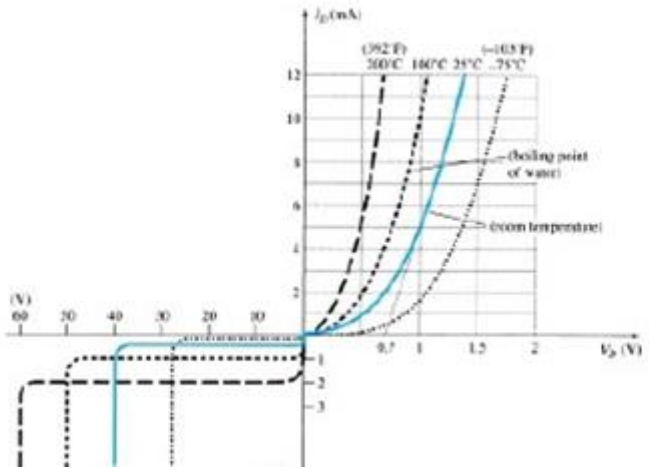


Fig 1.11 Variation in Diode Characteristics with temperature change

It is not uncommon for a germanium diode with an I_o in the order of 1 or 2 A at 25°C to have a leakage current of 100 A - 0.1 mA at a temperature of 100°C. Typical values of I_o for silicon are much lower than

that of germanium for similar power and current levels. The result is that even at high temperatures the levels of I_o for silicon diodes do not reach the same high levels obtained. For germanium—a very important reason that silicon devices enjoy a significantly higher level of development and utilization in design. Fundamentally, the open-circuit equivalent in the reverse bias region is better realized at any temperature with silicon than with germanium. The increasing levels of I_o with temperature account for the lower levels of threshold voltage, as shown in Fig. 1.11. Simply increase the level of I_o in and not rise in diode current. Of course, the level of TK also will be increase, but the increasing level of I_o will overpower the smaller percent change in TK. As the temperature increases the forward characteristics are actually becoming more "ideal,"

1.3 IDEAL VERSUS PRACTICAL RESISTANCE LEVELS

DC or Static Resistance

The application of a dc voltage to a circuit containing a semiconductor diode will result in an operating point on the characteristic curve that will not change with time. The resistance of the diode at the operating point can be found simply by finding the corresponding levels of VD and ID as shown in Fig. 1.12 and applying the following Equation:

$$R_D = \frac{V_D}{I_D}$$

The dc resistance levels at the knee and below will be greater than the resistance levels obtained for the vertical rise section of the characteristics. The resistance levels in the reverse-bias region will naturally be quite high. Since ohmmeters typically employ a relatively constant-current source, the resistance determined will be at a preset current level (typically, a few mill amperes).

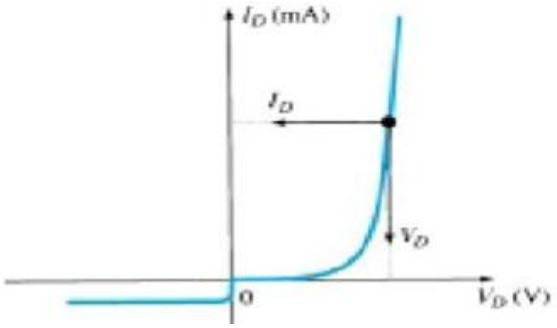


Fig 1.12 Determining the dc resistance of a diode at a particular operating point.

AC or Dynamic Resistance

It is obvious from Eq. 1.3 that the dc resistance of a diode is independent of the shape of the characteristic in the region surrounding the point of interest. If a sinusoidal rather than dc input is applied, the situation will change completely. The varying input will move the instantaneous operating point up and down a region of the characteristics and thus defines a specific change in current and voltage as shown in Fig. 1.13. With no applied varying signal, the point of operation would be the Q-point appearing on Fig. 1.13 determined by the applied dc levels. The designation Q-point is derived from the word quiescent, which means "still or unvarying." A straight-line drawn tangent to the curve through the Q-point as shown in Fig. 1.13 will define a particular change in voltage and current that can be used to determine the ac or dynamic resistance for this region of the diode characteristics. In equation form,

$$r_d = \frac{\Delta V_d}{\Delta I_d}$$

Where Δ Signifies a finite change in the quantity

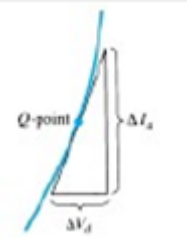


Fig 1.13: Determining the ac resistance of a diode at a particular operating point.

1.4 DIODE EQUIVALENT CIRCUITS

An equivalent circuit is a combination of elements properly chosen to best represent the actual terminal characteristics of a device, system, or such in a particular operating region. In other words, once the equivalent circuit is defined, the device symbol can be removed from a schematic and the equivalent circuit inserted in its place without severely affecting the actual behavior of the system. The result is often a network that can be solved using traditional circuit analysis techniques.

Piecewise-Linear Equivalent Circuit

One technique for obtaining an equivalent circuit for a diode is to approximate the characteristics of the device by straight-line segments, as shown in Fig. 1.31. The resulting equivalent

circuit is naturally called the piecewise-linear equivalent circuit. It should be obvious from Fig. 1.31 that the straight-line segments do not result in an exact duplication of the actual characteristics, especially in the knee region. However, the resulting segments are sufficiently close to the actual curve to establish an equivalent circuit that will provide an excellent first approximation to the actual behaviour of the device. The ideal diode is included to establish that there is only one direction of conduction through the device, and a reverse-bias condition will result in the open- circuit state for the device. Since a silicon semiconductor, diode does not reach the conduction state until VD reaches 0.7 V with a forward bias (as shown in Fig. 1.14a), a battery V_T opposing the conduction direction must appear in the equivalent circuit as shown in Fig. 1.14b. The battery simply specifies that the voltage across the device must be greater than the threshold battery voltage before conduction through the device in the direction dictated by the ideal diode can be established. When conduction is established, the resistance of the diode will be the specified value of r_{av}.

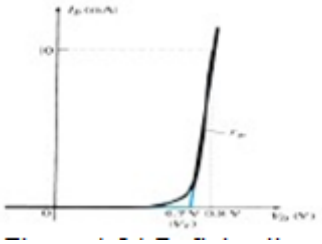


Fig: 1.14aDiode piecewise-linear model characteristics

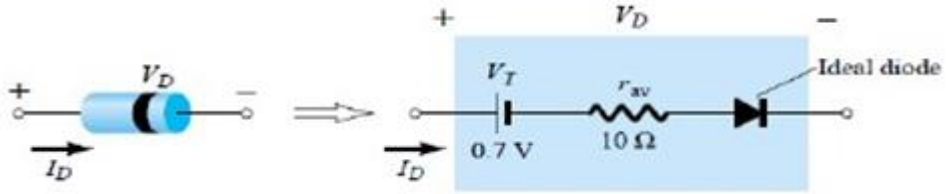


Fig: 1.14b Diode piecewise-linear model equivalent circuit

The approximate level of r_{av} can usually be determined from a specified operating point on the specification sheet. For instance, for a silicon semiconductor diode, if IF _ 10 mA (a forward conduction current for the diode) at VD _ 0.8 V, we know for silicon that a shift of 0.7 V is required before the characteristics rise.

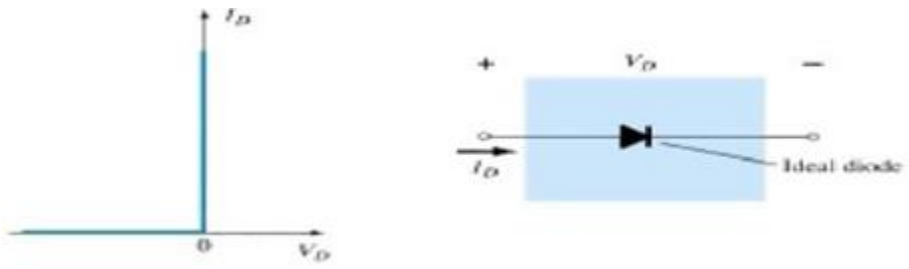


Fig 1.15 Ideal Diode and its characteristics

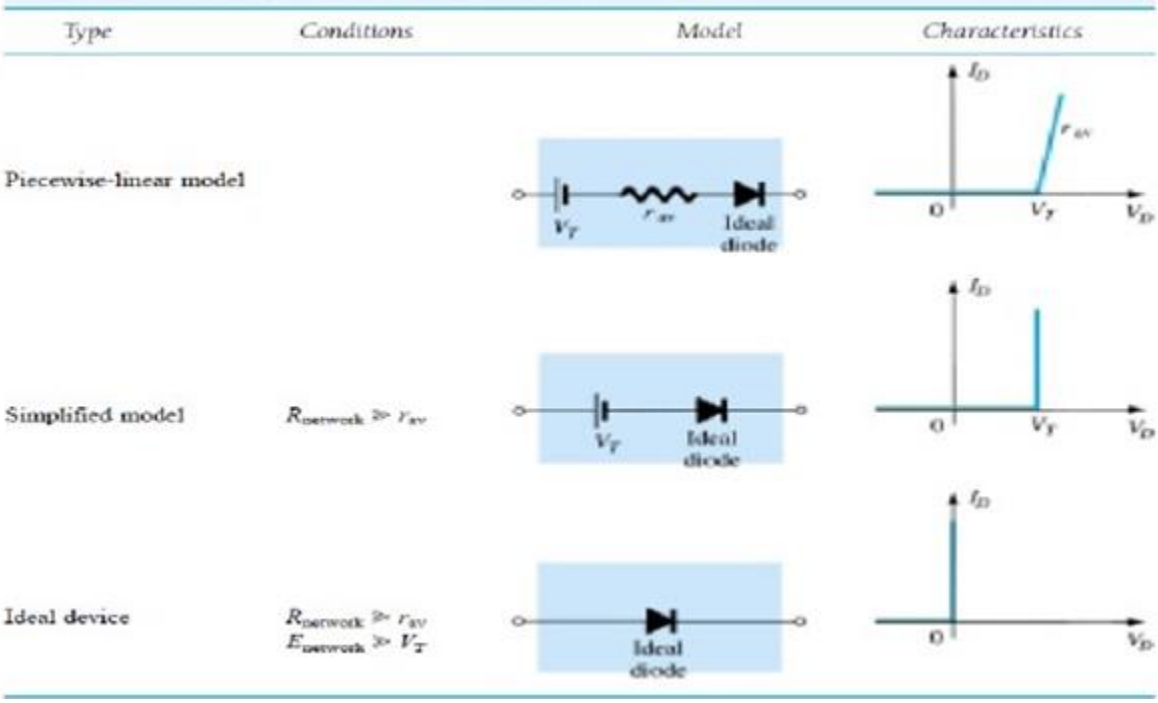


Fig 1.16: Diode equivalent circuits(models)

1.5 TRANSITION AND DIFFUSION CAPACITANCE

Electronic devices are inherently sensitive to very high frequencies. Most shunt capacitive effects that can be ignored at lower frequencies because the reactance XC=1/2πfC is very large (open-circuit equivalent). This, however, cannot be ignored at very high frequencies. XC will become sufficiently small due to the high value of f to introduce a low-reactance “shorting” path. In the p-n semiconductor diode, there are two capacitive effects to be considered. In the reverse-bias region we have the transition- or depletion region capacitance (CT), while in the forward-bias region we have the diffusion (CD) or storage capacitance. Recall that the basic equation for the capacitance of a parallel-plate capacitor is defined by C=€A/d, where € is the permittivity of the dielectric (insulator) between the plates of area A separated by a distance d. In the reverse-, bias region there is a depletion region (free of carriers) that behaves essentially like an insulator between the layers of opposite charge. Since

the depletion width (d) will increase with increased reverse-bias potential, the resulting transition capacitance will decrease. The fact that the capacitance is dependent on the applied reverse-bias potential has application in a number of electronic systems. Although the effect described above will also be present in the forward-bias region, it is over shadowed by a capacitance effect directly dependent on the rate at which charge is injected into the regions just outside the depletion region. The capacitive effects described above are represented by a capacitor in parallel with the ideal diode, as shown in Fig. 1.38. For low- or mid-frequency applications (except in the power area), however, the capacitor is normally not included in the diode symbol.

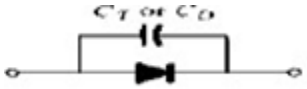


Fig 1.17: Including the effect of the transition or diffusion capacitance on the semiconductor diode

Diode capacitances: The diode exhibits two types of capacitances transition capacitance and diffusion capacitance.

- Transition capacitance: The capacitance which appears between positive ion layer in n-region and negative ion layer in p-region.
- Diffusion capacitance: This capacitance originates due to diffusion of charge carriers in the opposite regions.

The transition capacitance is very small as compared to the diffusion capacitance.

In reverse bias transition, the capacitance is the dominant and is given by:

$$C_T = \epsilon A/W$$

where C_T - transition capacitance

A - diode cross sectional area

W - depletion region width

In forward bias, the diffusion capacitance is the dominant and is given by:

$$C_D = dQ/dV = \tau * dI/dV = \tau * g = \tau/r \text{ (general)}$$

where C_D - diffusion capacitance

dQ - change in charge stored in depletion region

V - change in applied voltage

τ - time interval for change in voltage

g - diode conductance

r - diode resistance

The diffusion capacitance at low frequencies is given by the formula:

$$C_D = \tau * g/2 \text{ (low frequency)}$$

The diffusion capacitance at high frequencies is inversely proportional to the frequency and is given by the formula:

$$C_D = g(\tau/2\omega)^{1/2}$$

Note: The variation of diffusion capacitance with applied voltage is used in the design of varactor.

1.6 BREAK DOWN MECHANISMS

When an ordinary **P-N junction diode** is reverse biased, normally only very small reverse saturation current flows. This current is due to movement of minority carriers. It is almost independent of the voltage applied. However, if the reverse bias is increased, a point is reached when the junction breaks down and the reverse current increases abruptly. This current could be large enough to destroy the junction. If the reverse current is limited by means of a suitable series resistor, the power dissipation at the junction will not be excessive, and the device may be operated continuously in its breakdown region to its normal (reverse saturation) level. It is found that for a suitably designed diode, the breakdown voltage is very stable over a wide range of reverse currents. This quality gives the breakdown diode many useful applications as a voltage reference source.

The critical value of the voltage, at which the breakdown of a P-N junction diode occurs, is called the *breakdown voltage.* The breakdown voltage depends on the width of the depletion region, which, in turn, depends on the doping level. The junction offers almost zero resistance at the breakdown point.

There are two mechanisms by which breakdown can occur at a reverse biased P-N junction:

1. ***avalanche breakdown*** **and**
2. ***Zener breakdown.***

Avalanche breakdown

The minority carriers, under reverse biased conditions, flowing through the junction acquire a kinetic energy which increases with the increase in reverse voltage. At a sufficiently high reverse voltage (say 5 V or more), the kinetic energy of minority carriers becomes so large that they knock out electrons from the covalent bonds of the semiconductor material. As a result of collision, the liberated

electrons in turn liberate more electrons and the current becomes very large leading to the breakdown of the crystal structure itself. This phenomenon is called the avalanche breakdown. The breakdown region is the knee of the characteristic curve. Now the current is not controlled by the junction voltage but rather by the external circuit.

Zener breakdown

Under a very high reverse voltage, the depletion region expands and the potential barrier increases leading to a very high electric field across the junction. The electric field will break some of the covalent bonds of the semiconductor atoms leading to a large number of free minority carriers, which suddenly increase the reverse current. This is called the Zener effect. The breakdown occurs at a particular and constant value of reverse voltage called the breakdown voltage, it is found that Zener breakdown occurs at electric field intensity of about 3 x 10^7 V/m.

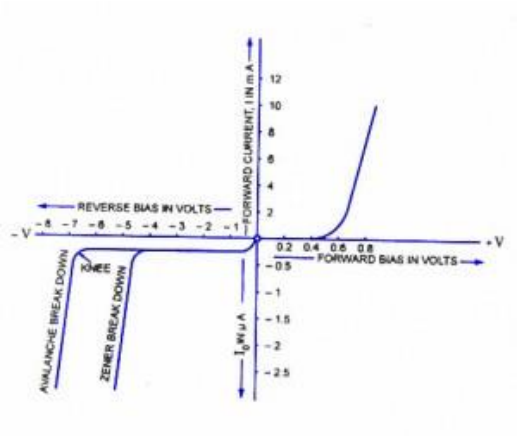


Fig 1.18: Diode characteristics with breakdown

Either of the two (Zener breakdown or avalanche breakdown) may occur independently, or both of these may occur simultaneously. Diode junctions that breakdown below 5 V are caused by Zener effect. Junctions that experience breakdown above 5 V are caused by avalanche effect. Junctions that breakdown around 5 V are usually caused by combination of two effects. The Zener breakdown occurs in heavily doped junctions (P-type semiconductor moderately doped and N-type heavily doped), which produce narrow depletion layers. The avalanche breakdown occurs in lightly doped junctions, which produce wide depletion layers. With the increase in junction temperature Zener breakdown voltage is reduced while the avalanche breakdown voltage increases. The Zener diodes have a negative temperature coefficient while avalanche diodes have a positive temperature coefficient. Diodes that have breakdown voltages around 5 V have zero temperature coefficient. The breakdown phenomenon is reversible and harmless so long as the safe operating temperature is maintained.

1.7 ZENER DIODES

The **Zener diode** is like a general-purpose signal diode consisting of a silicon PN junction. When biased in the forward direction it behaves just like a normal signal diode passing the rated current, but as soon as a reverse voltage applied across the zener diode exceeds the rated voltage of the device, the diodes breakdown voltage V_B is reached at which point a process called *Avalanche Breakdown* occurs in the semiconductor depletion layer and a current starts to flow through the diode to limit this increase in voltage.

The current now flowing through the zener diode increases dramatically to the maximum circuit value (which is usually limited by a series resistor) and once achieved this reverse saturation current remains fairly constant over a wide range of applied voltages. This breakdown voltage point, V_B is called the "zener voltage" for zener diodes and can range from less than one volt to hundreds of volts.

The point at which the zener voltage triggers the current to flow through the diode can be very accurately controlled (to less than 1% tolerance) in the doping stage of the diodes semiconductor construction giving the diode a specific *zener breakdown voltage*, (V_Z) for example, 4.3V or 7.5V. This zener breakdown voltage on the I-V curve is almost a vertical straight line.

Zener Diode I-V Characteristics

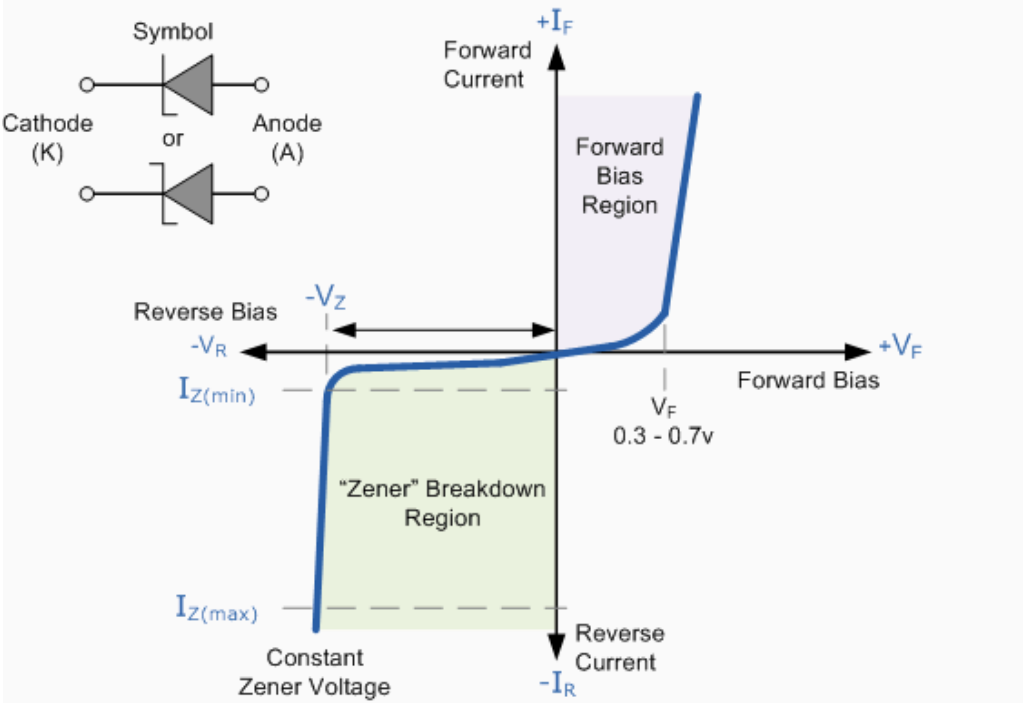


Fig 1.19 : Zener diode characteristics

The **Zener Diode** is used in its "reverse bias" or reverse breakdown mode, i.e. the diodes anode connects to the negative supply. From the I-V characteristics curve above, we can see that the zener

diode has a region in its reverse bias characteristics of almost a constant negative voltage regardless of the value of the current flowing through the diode and remains nearly constant even with large changes in current as long as the zener diodes current remains between the breakdown current $I_{Z(min)}$ and the maximum current rating $I_{Z(max)}$.

This ability to control itself can be used to great effect to regulate or stabilize a voltage source against supply or load variations. The fact that the voltage across the diode in the breakdown region is almost constant turns out to be an important application of the zener diode as a voltage regulator. The function of a regulator is to provide a constant output voltage to a load connected in parallel with it in spite of the ripples in the supply voltage or the variation in the load current and the zener diode will continue to regulate the voltage until the diodes current falls below the minimum $I_{Z(min)}$ value in the reverse breakdown region.

UNIT II

BIPOLAR JUNCTION TRANSISTOR

2.1 INTRODUCTION

A bipolar junction transistor (BJT) is a three terminal device in which operation depends on the interaction of both majority and minority carriers and hence the name bipolar. The BJT is analogues to vacuum triode and is comparatively smaller in size. It is used as amplifier and oscillator circuits, and as a switch in digital circuits. It has wide applications in computers, satellites and other modern communication systems.

A **Transistor** is a three terminal semiconductor device that regulates current or voltage flow and acts as a switch or gate for signals.

Why Do We Need Transistors?

Suppose that you have a FM receiver which grabs the signal you want. The received signal will obviously be weak due to the disturbances it would face during its journey. Now if this signal is read as it is, you cannot get a fair output. Hence we need to amplify the signal. **Amplification** means increasing the signal strength.

This is just an instance. Amplification is needed wherever the signal strength has to be increased. This is done by a transistor. A transistor also acts as a **switch** to choose between available options. It also **regulates** the incoming **current and voltage** of the signals.

Constructional Details of a Transistor

The Transistor is a three terminal solid state device which is formed by connecting two diodes back to back. Hence it has got **two PN junctions**. Three terminals are drawn out of the three semiconductor materials present in it. This type of connection offers two types of transistors. They are **PNP** and **NPN** which means an N-type material between two Ptypes and the other is a P-type material between two N-types respectively.

The construction of transistors is as shown in the following figure which explains the idea discussed above.

Construction of PNP & NPN Transistors

The three terminals drawn from the transistor indicate Emitter, Base and Collector terminals. They have their functionality as discussed below.

Emitter

- The left hand side of the above shown structure can be understood as **Emitter**.
- This has a **moderate size** and is **heavily doped** as its main function is to **supply** a number of **majority carriers**, i.e. either electrons or holes.
- As this emits electrons, it is called as an Emitter.
- This is simply indicated with the letter **E**.

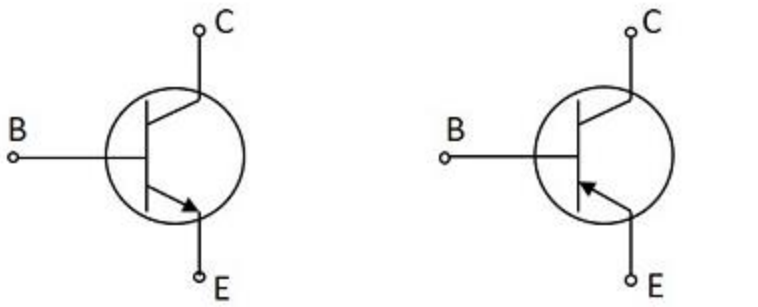
Base

- The middle material in the above figure is the **Base**.
- This is **thin** and **lightly doped**.
- Its main function is to **pass** the majority carriers from the emitter to the collector.
- This is indicated by the letter **B**.

Collector

- The right side material in the above figure can be understood as a **Collector**.
- Its name implies its function of **collecting the carriers**.
- This is **a bit larger** in size than emitter and base. It is **moderately doped**.
- This is indicated by the letter **C**.

The symbols of PNP and NPN transistors are as shown below.

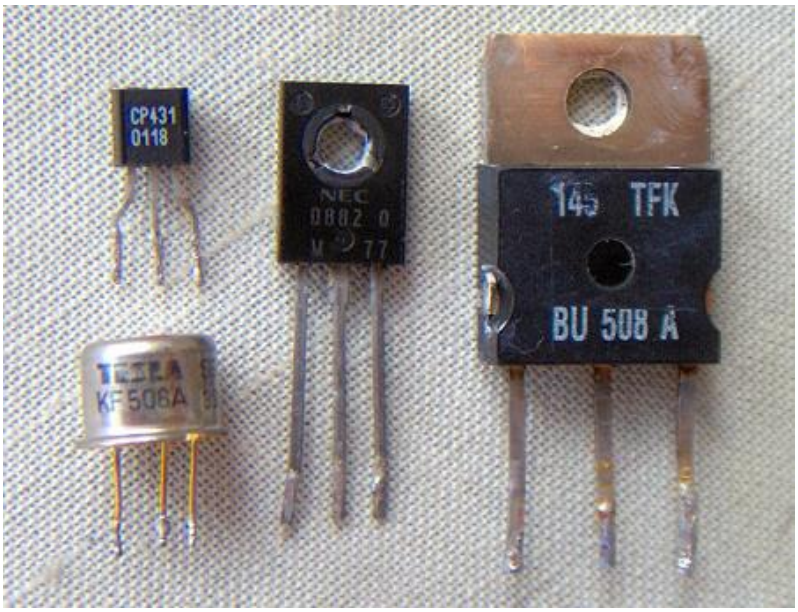


Symbol of NPN transistor

Symbol of PNP transistor

The **arrow-head** in the above figures indicated the **emitter** of a transistor. As the collector of a transistor has to dissipate much greater power, it is made large. Due to the specific functions of emitter and collector, they are **not interchangeable**. Hence the terminals are always to be kept in mind while using a transistor.

In a Practical transistor, there is a notch present near the emitter lead for identification. The PNP and NPN transistors can be differentiated using a Multimeter. The following figure shows how different practical transistors look like.

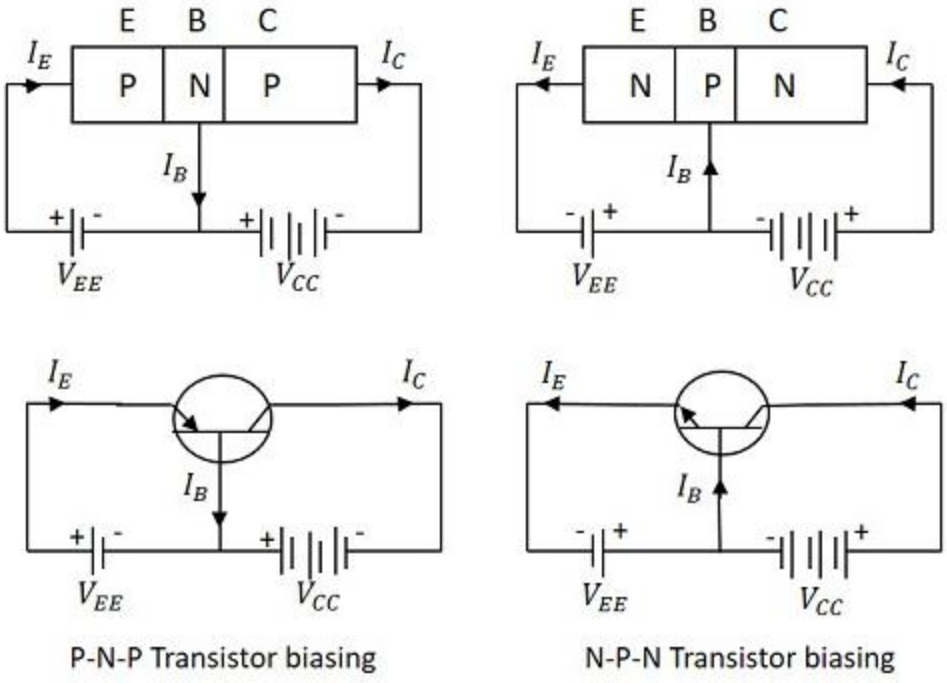


We have so far discussed the constructional details of a transistor, but to understand the operation of a transistor, first we need to know about the biasing.

Transistor Biasing

As we know that a transistor is a combination of two diodes, we have two junctions here. As one junction is between the emitter and base, that is called as **Emitter-Base junction** and likewise, the other is **Collector-Base junction**.

Biasing is controlling the operation of the circuit by providing power supply. The function of both the PN junctions is controlled by providing bias to the circuit through some dc supply. The figure below shows how a transistor is biased.



P-N-P Transistor biasing

N-P-N Transistor biasing

By having a look at the above figure, it is understood that

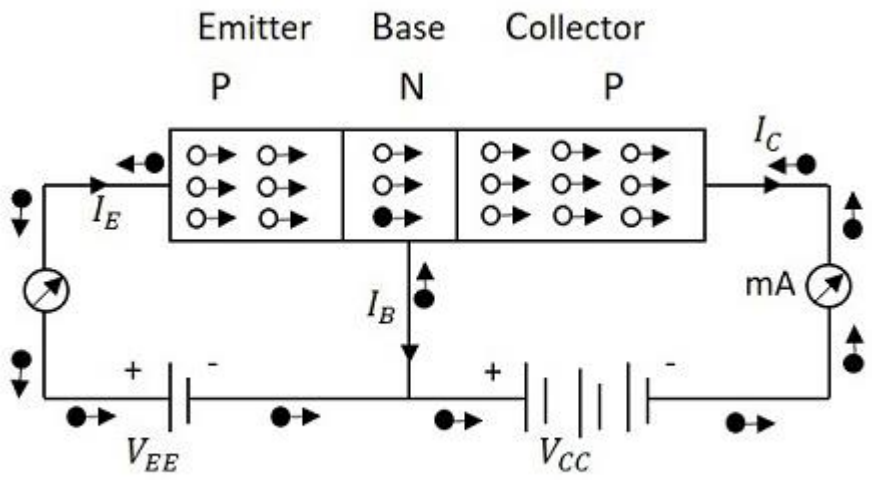
- The N-type material is provided negative supply and P-type material is given positive supply to make the circuit **Forward bias**.
- The N-type material is provided positive supply and P-type material is given negative supply to make the circuit **Reverse bias**.

By applying the power, the **emitter base junction** is always **forward biased** as the emitter resistance is very small. The **collector base junction** is **reverse biased** and its resistance is a bit higher. A small forward bias is sufficient at the emitter junction whereas a high reverse bias has to be applied at the collector junction.

The direction of current indicated in the circuits above, also called as the **Conventional Current**, is the movement of hole current which is **opposite to the electron current**.

Operation PNP Transistor

The operation of a PNP transistor can be explained by having a look at the following figure, in which emitter-base junction is forward biased and collector-base junction is reverse biased.



Operation of a PNP transistor

The voltage $\mathbf{V_{EE}}$ provides a positive potential at the emitter which repels the holes in the P-type material and these holes cross the emitter-base junction, to reach the base region. There a very low percent of holes recombine with free electrons of N-region. This provides very low current which constitutes the base current $\mathbf{I_B}$. The remaining holes cross the collector-base junction, to constitute collector current $\mathbf{I_C}$, which is the hole current.

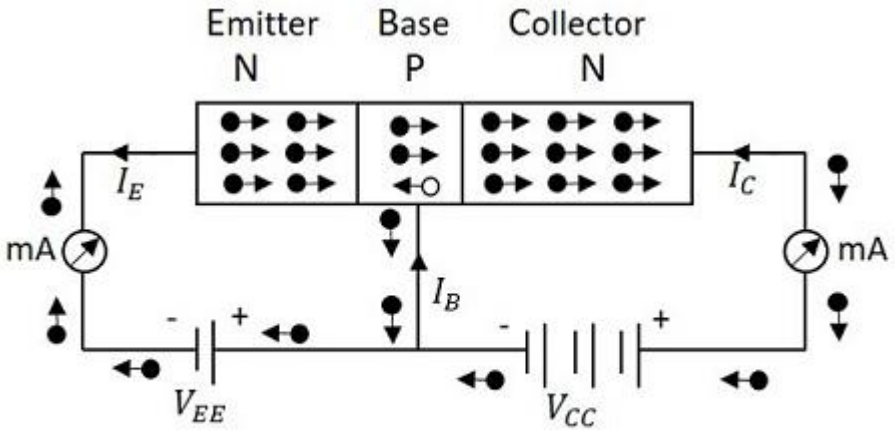
As a hole reaches the collector terminal, an electron from the battery negative terminal fills the space in the collector. This flow slowly increases and the electron minority current flows through the emitter, where each electron entering the positive terminal of $\mathbf{V_{EE}}$, is replaced by a hole by moving towards the emitter junction. This constitutes emitter current $\mathbf{I_E}$.

Hence we can understand that −

- The conduction in a PNP transistor takes place through holes.
- The collector current is slightly less than the emitter current.
- The increase or decrease in the emitter current affects the collector current.

Operation NPN Transistor

The operation of an NPN transistor can be explained by having a look at the following figure, in which emitter-base junction is forward biased and collector-base junction is reverse biased.



Operation of a NPN transistor

The voltage $\mathbf{V_{EE}}$ provides a negative potential at the emitter which repels the electrons in the N-type material and these electrons cross the emitter-base junction, to reach the base region. There a very low percent of electrons recombine with free holes of P-region. This provides very low current which constitutes the base current $\mathbf{I_B}$. The remaining holes cross the collector-base junction, to constitute the collector current $\mathbf{I_C}$.

As an electron reaches out of the collector terminal, and enters the positive terminal of the battery, an electron from the negative terminal of the battery $\mathbf{V_{EE}}$ enters the emitter region. This flow slowly increases and the electron current flows through the transistor.

Hence we can understand that −

- The conduction in a NPN transistor takes place through electrons.
- The collector current is higher than the emitter current.
- The increase or decrease in the emitter current affects the collector current.

Advantages

There are many advantages of a transistor such as −

- High voltage gain.
- Lower supply voltage is sufficient.
- Most suitable for low power applications.
- Smaller and lighter in weight.
- Mechanically stronger than vacuum tubes.
- No external heating required like vacuum tubes.
- Very suitable to integrate with resistors and diodes to produce ICs.

There are few disadvantages such as they cannot be used for high power applications due to lower power dissipation. They have lower input impedance and they are temperature dependent.

2.3 TRANSISTOR CURRENT COMPONENTS:

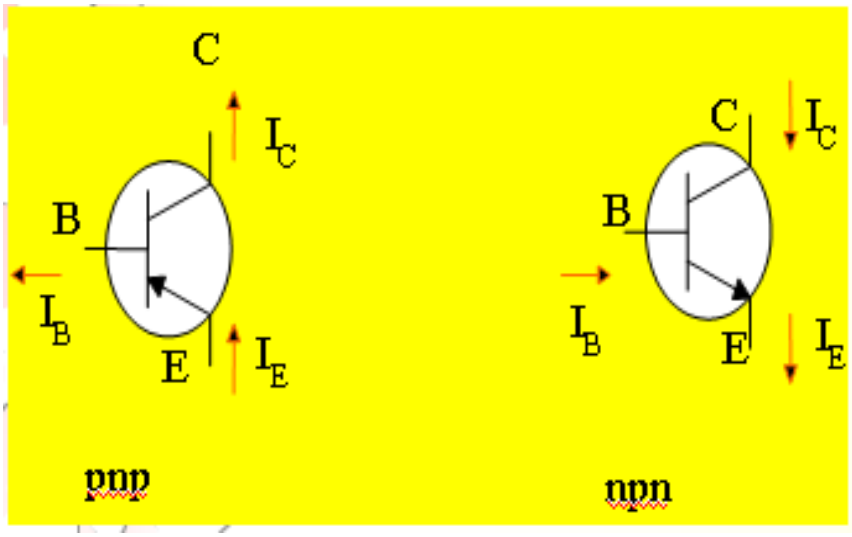


Fig 2.2 Bipolar Junction Transistor Current Components

The above fig 3.2 shows the various current components, which flow across the forward biased emitter junction and reverse- biased collector junction. The emitter current I_E consists of hole current I_{PE} (holes crossing from emitter into base) and electron current I_{nE} (electrons crossing from base into emitter).The ratio of hole to electron currents, I_{pE} / I_{nE} , crossing the emitter junction is proportional to the ratio of the conductivity of the p material to that of the n material. In a transistor, the doping of that of the emitter is made much larger than the doping of the base. This feature ensures (in p-n-p transistor) that the emitter current consists an almost entirely of holes. Such a situation is desired since the current which results from electrons crossing the emitter junction from base to emitter do not contribute carriers, which can reach the collector.

Not all the holes crossing the emitter junction J_E reach the the collector junction J_C

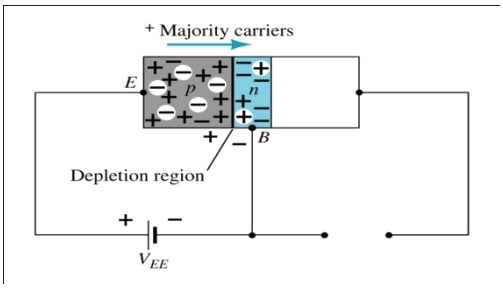
Because some of them combine with the electrons in n-type base. If I_{pC} is hole current at junction J_C there must be a bulk recombination current ($I_{PE-} I_{pC}$) leaving the base.

Actually, electrons enter the base region through the base lead to supply those charges, which have been lost by recombination with the holes injected in to the base across J_E. If the emitter were open circuited so that $I_E=0$ then I_{pC} would be zero. Under these circumstances, the base and collector current I_C would equal the reverse saturation current I_{CO}. If $I_E \neq 0$ then

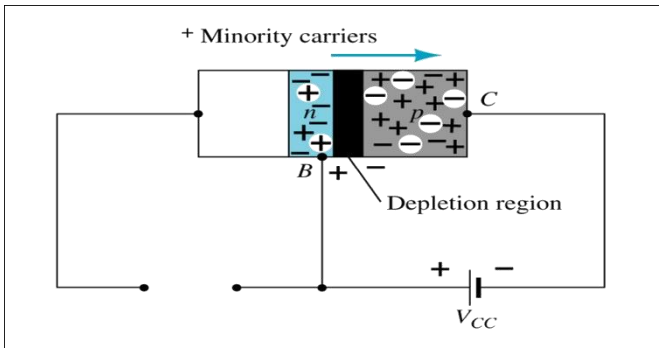
$I_C = I_{CO} - I_{pC}$

For a p-n-p transistor, I_{CO} consists of holes moving across J_C from left to right (base to collector) and electrons crossing J_C in opposite direction. Assumed referenced direction for I_{CO} i.e. from right to left, then for a p-n-p transistor, I_{CO} is negative. For an n-p-n transistor, I_{CO} is positive.The basic operation will be described using the pnp transistor. The operation of the pnp transistor is exactly the same if the roles played by the electron and hole are interchanged.

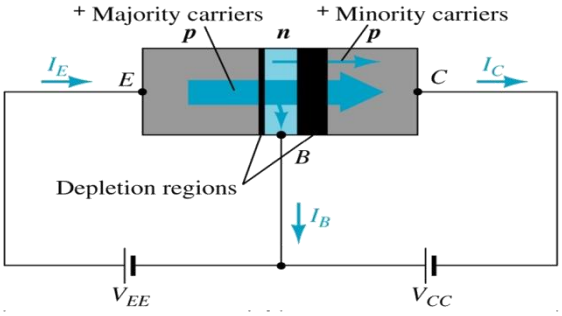
One p-n junction of a transistor is reverse-biased, whereas the other is forward-biased.



3.3a Forward-biased junction of a pnp transistor



2.3b Reverse-biased junction of a pnp transistor



2.3c Both biasing potentials have been applied to a pnp transistor and resulting majority and minority carrier flows indicated.

Majority carriers (+) will diffuse across the forward-biased p-n junction into the n-type material.

A very small number of carriers (+) will through n-type material to the base terminal. Resulting IB is typically in order of microamperes.

The large number of majority carriers will diffuse across the reverse-biased junction into the p-type material connected to the collector terminal

Applying KCL to the transistor :

$$I_E = I_C + I_B$$

The comprises of two components – the majority and minority carriers

$$I_C = I_{Cmajority} + I_{COminority}$$

I_{CO} – I_C current with emitter terminal open and is called leakage current

Various parameters which relate the current components is given below

Emitter efficiency:

$$\gamma = \frac{currentofinjectedcarriersatJ_E}{totalemittercurrent}$$

$$\gamma = \frac{I_{PE}}{I_{pE} + I_{nE}} = \frac{I_{pE}}{I_{nE}}$$

Transport Factor:

$$\beta^* = \frac{injectedcarriercurrentreachingJ_C}{injectedcarrierncurrentatJ_E}$$

$$\beta^* = \frac{I_{pC}}{I_{nE}}$$

Large signal current gain:

The ratio of the negative of collector current increment to the emitter current change from zero (cut-off)to I_E the large signal current gain of a common base transistor.

$$\alpha = \frac{-(I_C - I_{CO})}{I_E}$$

Since I_C and I_E have opposite signs, then α, as defined, is always positive. Typically numerical values of α lies in the range of 0.90 to 0.995

$$\alpha = \frac{I_{pC}}{I_E} = \frac{I_{pC}}{I_{nE}} * \frac{I_{pE}}{I_E} \quad \alpha = \beta * \gamma$$

The transistor alpha is the product of the transport factor and the emitter efficiency. This statement assumes that the collector multiplication ratio α^* is unity. α^* is the ratio of total current crossing J_C to hole arriving at the junction.

2.4 Bipolar Transistor Configurations

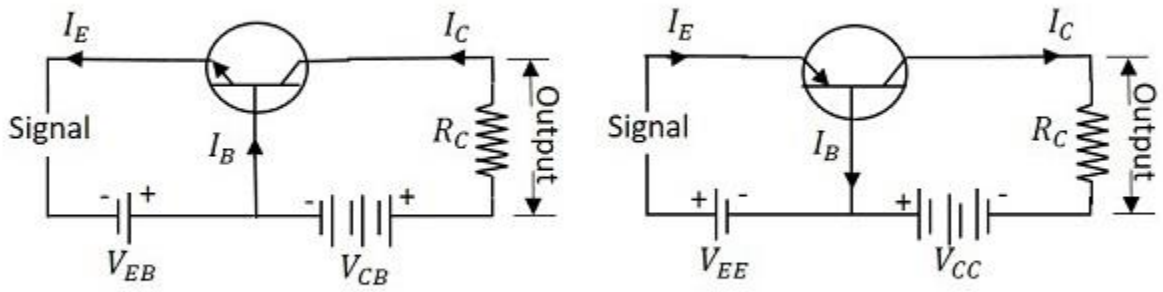
A Transistor has 3 terminals, the emitter, the base and the collector. Using these 3 terminals the transistor can be connected in a circuit with one terminal common to both input and output in a 3 different possible configurations.

The three types of configurations are **Common Base, Common Emitter** and **Common Collector** configurations. In every configuration, the emitter junction is forward biased and the collector junction is reverse biased.

Common Base (CB) Configuration

The name itself implies that the Base terminal is taken as common terminal for both input and output of the transistor. The common base connection for both NPN and PNP transistors is as shown in the following figure.

Common Base Connection



Using NPN transistor

Using PNP transistor

For the sake of understanding, let us consider NPN transistor in CB configuration. When the emitter voltage is applied, as it is forward biased, the electrons from the negative terminal repel the emitter electrons and current flows through the emitter and base to the collector to contribute collector current. The collector voltage $\mathbf{V_{CB}}$ is kept constant throughout this.

In the CB configuration, the input current is the emitter current $\mathbf{I_E}$ and the output current is the collector current $\mathbf{I_C}$.

Current Amplification Factor (α)

The ratio of change in collector current (ΔICΔIC) to the change in emitter current (ΔIEΔIE) when collector voltage $\mathbf{V_{CB}}$ is kept constant, is called as **Current amplification factor**. It is denoted by α.

$$\alpha = \frac{\Delta I_C}{\Delta I_E} \; at\; constant\; V_{CB}$$

Expression for Collector current

With the idea above, let us try to draw some expression for collector current. Along with the emitter current flowing, there is some amount of base current IB which flows through the base terminal due to electron hole recombination. As collector-base junction is reverse biased, there is another current which is flown due to minority charge carriers. This is the leakage current which can be understood as $\mathbf{I_{leakage}}$. This is due to minority charge carriers and hence very small.

The emitter current that reaches the collector terminal is

$$\alpha I_E$$

Total collector current

$$I_C = \alpha I_E + I_{leakage}$$

If the emitter-base voltage V_{EB} = 0, even then, there flows a small leakage current, which can be termed as I_{CBO} (collector-base current with output open).

The collector current therefore can be expressed as

$$I_C = \alpha I_E + I_{CBO}$$

$$I_E = I_C + I_B$$

$$I_C = \alpha(I_C + I_B) + I_{CBO}$$

$$I_C(1 - \alpha) = \alpha I_B + I_{CBO}$$

$$I_C = \left(\frac{\alpha}{1 - \alpha}\right) I_B + \left(\frac{I_{CBO}}{1 - \alpha}\right)$$

$$I_C = \left(\frac{\alpha}{1 - \alpha}\right) I_B + \left(\frac{1}{1 - \alpha}\right) I_{CBO}$$

Hence the above derived is the expression for collector current. The value of collector current depends on base current and leakage current along with the current amplification factor of that transistor in use.

Characteristics of CB configuration

- This configuration provides voltage gain but no current gain.
- Being $\mathbf{V_{CB}}$ constant, with a small increase in the Emitter-base voltage V_{EB}, Emitter current $\mathbf{I_E}$ gets increased.
- Emitter Current $\mathbf{I_E}$ is independent of Collector voltage $\mathbf{V_{CB}}$.
- Collector Voltage $\mathbf{V_{CB}}$ can affect the collector current $\mathbf{I_C}$only at low voltages, when V_{EB} is kept constant.
- The input resistance ri is the ratio of change in emitter-base voltage (ΔVEB) to the change in emitter current (ΔIE) at constant collector base voltage $\mathbf{V_{CB}}$.

$$= \frac{\Delta V_{EB}}{\Delta I_E} \text{ at constant } V_{CB}$$

- As the input resistance is of very low value, a small value of V_{EB} is enough to produce a large current flow of emitter current $\mathbf{I_E}$.

- The output resistance r_o is the ratio of change in the collector base voltage (ΔVCB) to the change in collector current (ΔIC) at constant emitter current I_E.

$$r_o = \frac{\Delta V_{CB}}{\Delta I_C} \; at \; constant \; l_E$$

- As the output resistance is of very high value, a large change in V_{CB} produces a very little change in collector current I_C.
- This Configuration provides good stability against increase in temperature.
- The CB configuration is used for high frequency applications.

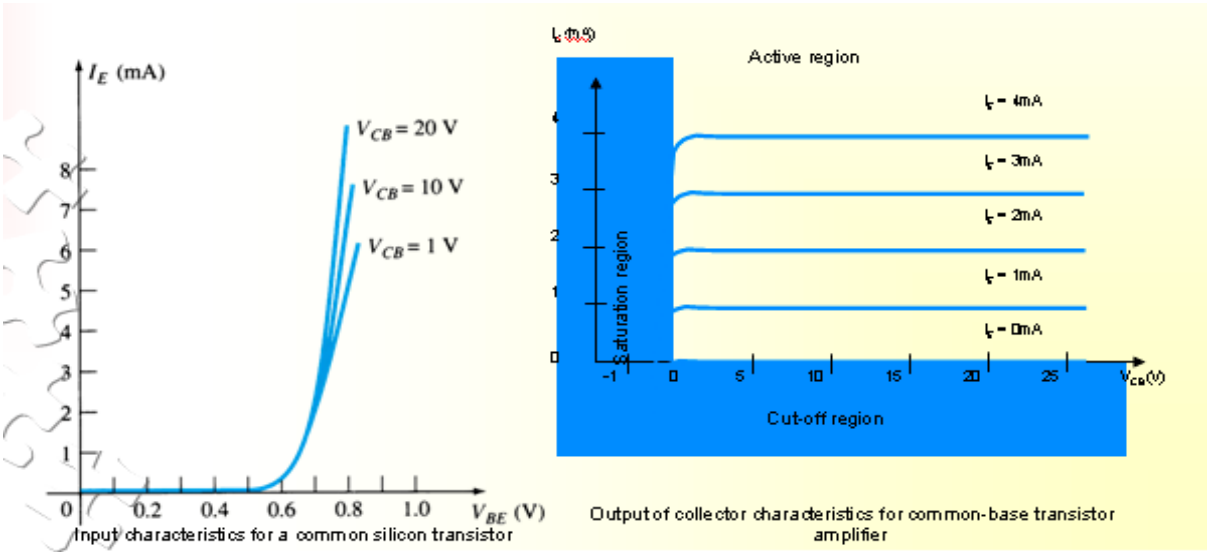
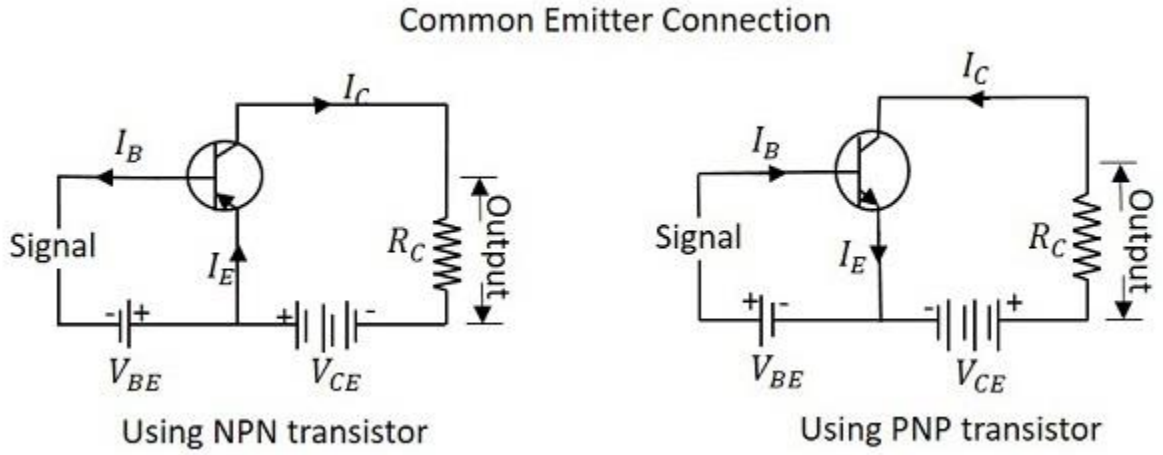


Fig 2.5 CB Input-Output Characteristics

Active region	Saturation region	Cut-off region
• IE increased, Ic increased • BE junction forward bias and CB junction reverse bias • Refer to the graf, Ic ≈ IE • Ic not depends on VCB • Suitable region for the transistor working as amplifier	• BE and CB junction is forward bias • Small changes in VCB will cause big different to Ic • The allocation for this region is to the left of VCB= 0 V.	• Region below the line of IE=0 A • BE and CB is reverse bias • no current flow at collector, only leakage current

Common Emitter (CE) Configuration

The name itself implies that the **Emitter** terminal is taken as common terminal for both input and output of the transistor. The common emitter connection for both NPN and PNP transistors is as shown in the following figure.



Just as in CB configuration, the emitter junction is forward biased and the collector junction is reverse biased. The flow of electrons is controlled in the same manner. The input current is the base current I_B and the output current is the collector current I_C here.

Base Current Amplification factor (β)

The ratio of change in collector current (ΔICΔIC) to the change in base current (ΔIBΔIB) is known as **Base Current Amplification Factor**. It is denoted by β

$$\beta = \frac{\Delta I_C}{\Delta I_B}$$

Relation between β and α

Let us try to derive the relation between base current amplification factor and emitter current amplification factor.

$$\beta = \frac{\Delta I_C}{\Delta I_B}$$

$$\alpha = \frac{\Delta I_C}{\Delta I_E}$$

$$I_E = I_B + I_C$$

$$\Delta I_E = \Delta I_B + \Delta I_C$$

$$\Delta I_B = \Delta I_E - \Delta I_C$$

We can write

$$\beta = \frac{\Delta I_C}{\Delta I_E - \Delta I_C}$$

$$\beta = \frac{\frac{\Delta I_C}{\Delta I_E}}{\frac{\Delta I_E}{\Delta I_E} - \frac{\Delta I_C}{\Delta I_E}}$$

$$\alpha = \frac{\Delta I_C}{\Delta I_E}$$

We have

$$\alpha = \frac{\Delta I_C}{\Delta I_E}$$

Therefore,

$$\beta = \frac{\alpha}{1-\alpha}$$

From the above equation, it is evident that, as α approaches 1, β reaches infinity.

Hence, **the current gain in Common Emitter connection is very high**. This is the reason this circuit connection is mostly used in all transistor applications.

Expression for Collector Current

In the Common Emitter configuration, $\mathbf{I_B}$ is the input current and $\mathbf{I_C}$ is the output current.

We know

IE=IB+IC

And

$$I_C = \alpha I_E + I_{CBO}$$

$$= \alpha(I_B + I_C) + I_{CBO}$$

$$I_C(1 - \alpha) = \alpha I_B + I_{CBO}$$

$$I_C = \frac{\alpha}{1-\alpha} I_B + \frac{1}{1-\alpha} I_{CBO}$$

If base circuit is open, i.e. if $I_B = 0$,

The collector emitter current with base open is I_{CEO}

$$I_{CEO} = \frac{1}{1-\alpha} I_{CBO}$$

Substituting the value of this in the previous equation, we get

$$I_C = \frac{\alpha}{1-\alpha} I_B + I_{CEO}$$

$$I_C = \beta I_B + I_{CEO}$$

Hence the equation for collector current is obtained.

Knee Voltage

In CE configuration, by keeping the base current I_B constant, if V_{CE} is varied, I_C increases nearly to 1v of V_{CE} and stays constant thereafter. This value of V_{CE} up to which collector current I_C changes with V_{CE} is called the **Knee Voltage**. The transistors while operating in CE configuration, they are operated above this knee voltage.

Characteristics of CE Configuration

- This configuration provides good current gain and voltage gain.
- Keeping V_{CE} constant, with a small increase in V_{BE} the base current I_B increases rapidly than in CB configurations.
- For any value of V_{CE} above knee voltage, I_C is approximately equal to βI_B.
- The input resistance r_i is the ratio of change in base emitter voltage (ΔVBE) to the change in base current (ΔIB) at constant collector emitter voltage V_{CE}.

$$r_i = \frac{\Delta V_{BE}}{\Delta I_B} \text{ at constant } V_{CE}$$

- As the input resistance is of very low value, a small value of V_{BE} is enough to produce a large current flow of base current I_B.
- The output resistance r_o is the ratio of change in collector emitter voltage (ΔVCEΔVCE) to the change in collector current (ΔICΔIC) at constant I_B.

$$r_o = \frac{\Delta V_{CE}}{\Delta I_C} \text{ at constant } I_B$$

- As the output resistance of CE circuit is less than that of CB circuit.

- This configuration is usually used for bias stabilization methods and audio frequency applications.

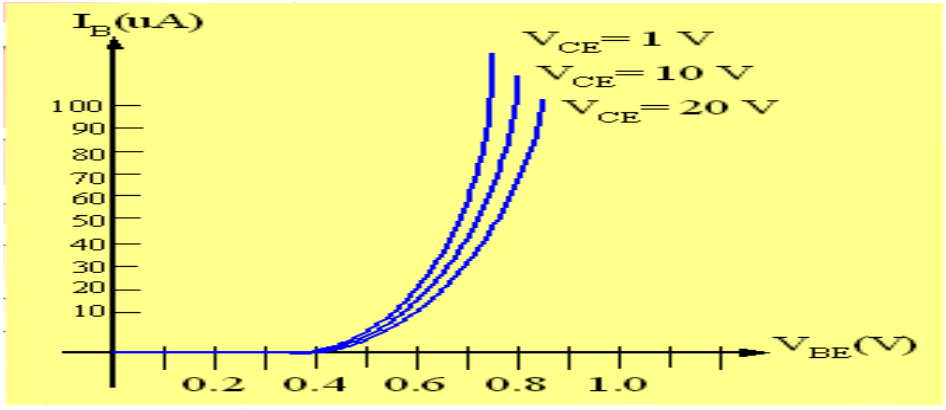


Fig 3.9a Input characteristics for common-emitter npn transistor

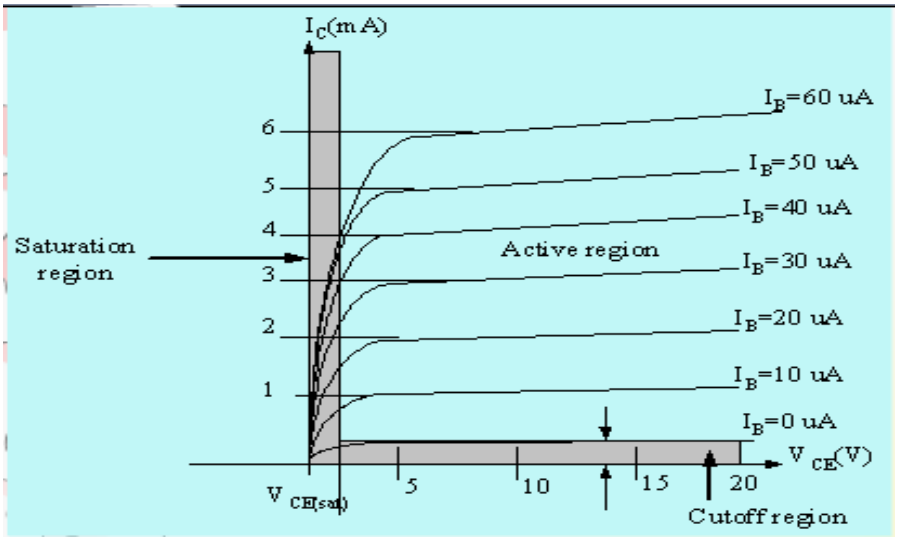


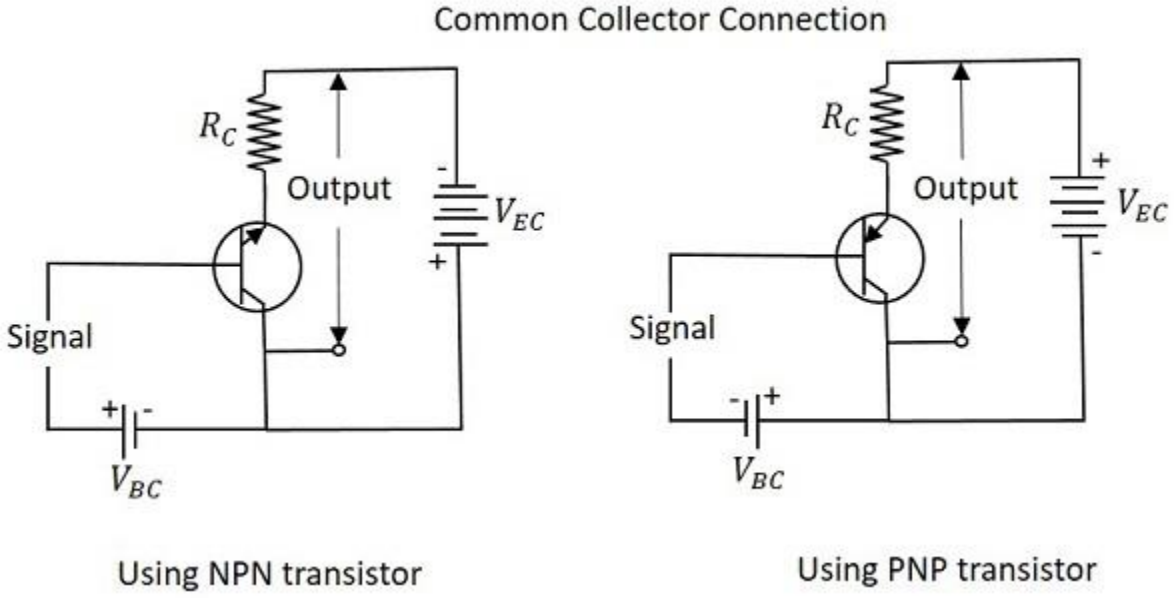
Fig 2.9b Output characteristics for common-emitter npn transistor

Active region	Saturation region	Cut-off region
• B-E junction is forward bias • C-B junction is reverse bias • can be employed for voltage, current and power amplification	• B-E and C-B junction is forward bias, thus the values of I_B and I_C is too big. • The value of V_{CE} is so small. • Suitable region when the transistor as a logic switch. • NOT and avoid this region when the transistor as an amplifier.	• region below I_B=0μA is to be avoided if an undistorted o/p signal is required • B-E junction and C-B junction is reverse bias • I_B=0, I_C not zero, during this condition I_C=I_{CEO} where is this current flow when B-E is reverse bias.

2.7 COMMON – COLLECTOR CONFIGURATION

Common Collector (CC) Configuration

The name itself implies that the **Collector** terminal is taken as common terminal for both input and output of the transistor. The common collector connection for both NPN and PNP transistors is as shown in the following figure.



Using NPN transistor

Using PNP transistor

Just as in CB and CE configurations, the emitter junction is forward biased and the collector junction is reverse biased. The flow of electrons is controlled in the same manner. The input current is the base current $\mathbf{I_B}$ and the output current is the emitter current $\mathbf{I_E}$ here.

Current Amplification Factor (γ)

The ratio of change in emitter current (ΔIEΔIE) to the change in base current (ΔIBΔIB) is known as **Current Amplification factor**in common collector (CC) configuration. It is denoted by **γ**.

$$\gamma = \frac{\Delta I_E}{\Delta I_B}$$

- The current gain in CC configuration is same as in CE configuration.
- The voltage gain in CC configuration is always less than 1.

Relation between γ and α

Let us try to draw some relation between γ and α

$$\gamma = \frac{\Delta I_E}{\Delta I_B}$$

$$\alpha = \frac{\Delta I_C}{\Delta I_E}$$

$$I_E = I_B + I_C$$

$$\Delta I_E = \Delta I_B + \Delta I_C$$

$$\Delta I_B = \Delta I_E - \Delta I_C$$

Substituting the value of I_B, we get

$$\gamma = \frac{\Delta I_E}{\Delta I_E - \Delta I_C}$$

Dividing by ΔIE

$$\gamma = \frac{\frac{\Delta I_E}{\Delta I_E}}{\frac{\Delta I_E}{\Delta I_E} - \frac{\Delta I_C}{\Delta I_E}}$$

$$\frac{1}{1 - \alpha}$$

$$\gamma = \frac{1}{1 - \alpha}$$

Expression for collector current

We know

$$I_C = \alpha I_E + I_{CBO}$$

$$I_E = I_B + I_C = I_B + (\alpha I_E + I_{CBO})$$

$$I_E(1 - \alpha) = I_B + I_{CBO}$$

$$I_E = \frac{I_B}{1 - \alpha} + \frac{I_{CBO}}{1 - \alpha}$$

$$I_C \cong I_E = (\beta + 1)I_B + (\beta + 1)I_{CBO}$$

The above is the expression for collector current.

Characteristics of CC Configuration

- This configuration provides current gain but no voltage gain.
- In CC configuration, the input resistance is high and the output resistance is low.
- The voltage gain provided by this circuit is less than 1.
- The sum of collector current and base current equals emitter current.
- The input and output signals are in phase.
- This configuration works as non-inverting amplifier output.
- This circuit is mostly used for impedance matching. That means, to drive a low impedance load from a high impedance source.

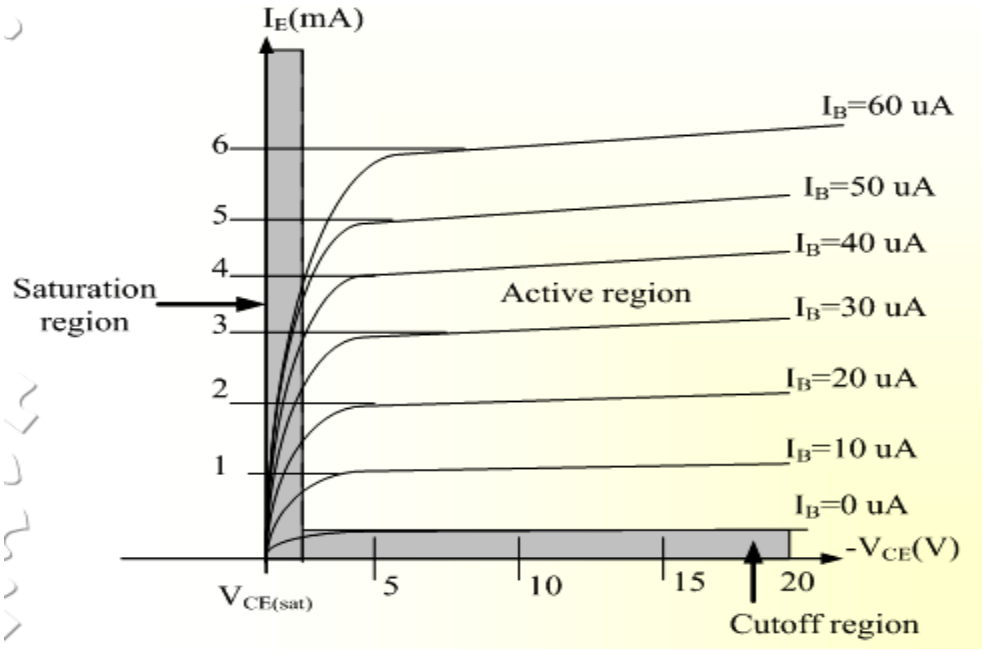


Fig 2.11 Output Characteristics of CC Configuration for npn Transistor

UNIT III

Number System and Boolean Algebra

If base or radix of a number system is 'r', then the numbers present in that number system are ranging from zero to r-1. The total numbers present in that number system is 'r'. So, we will get various number systems, by choosing the values of radix as greater than or equal to two.

In this chapter, let us discuss about the **popular number systems** and how to represent a number in the respective number system. The following number systems are the most commonly used.

- Decimal Number system
- Binary Number system
- Octal Number system
- Hexadecimal Number system

Decimal Number System

The **base** or radix of Decimal number system is **10**. So, the numbers ranging from 0 to 9 are used in this number system. The part of the number that lies to the left of the **decimal point**is known as integer part. Similarly, the part of the number that lies to the right of the decimal point is known as fractional part.

In this number system, the successive positions to the left of the decimal point having weights of 10^0, 10^1, 10^2, 10^3 and so on. Similarly, the successive positions to the right of the decimal point having weights of 10^{-1}, 10^{-2}, 10^{-3} and so on. That means, each position has specific weight, which is **power of base 10**

Example

Consider the **decimal number 1358.246**. Integer part of this number is 1358 and fractional part of this number is 0.246. The digits 8, 5, 3 and 1 have weights of 100, 101, 10^2 and 10^3respectively. Similarly, the digits 2, 4 and 6 have weights of 10^{-1}, 10^{-2} and 10^{-3} respectively.

Mathematically, we can write it as

$1358.246 = (1 \times 10^3) + (3 \times 10^2) + (5 \times 10^1) + (8 \times 10^0) + (2 \times 10^{-1}) + (4 \times 10^{-2}) + (6 \times 10^{-3})$

After simplifying the right hand side terms, we will get the decimal number, which is on left hand side.

Binary Number System

All digital circuits and systems use this binary number system. The **base** or radix of this number system is **2**. So, the numbers 0 and 1 are used in this number system.

The part of the number, which lies to the left of the **binary point** is known as integer part. Similarly, the part of the number, which lies to the right of the binary point is known as fractional part.

In this number system, the successive positions to the left of the binary point having weights of 2^0, 2^1, 2^2, 2^3 and so on. Similarly, the successive positions to the right of the binary point having weights of 2^{-1}, 2^{-2}, 2^{-3} and so on. That means, each position has specific weight, which is **power of base 2**.

Example

Consider the **binary number 1101.011**. Integer part of this number is 1101 and fractional part of this number is 0.011. The digits 1, 0, 1 and 1 of integer part have weights of 2^0, 2^1, 2^2, 2^3respectively. Similarly, the digits 0, 1 and 1 of fractional part have weights of 2^{-1}, 2^{-2}, 2^{-3} respectively.

Mathematically, we can write it as

$1101.011 = (1 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (1 \times 2^0) + (0 \times 2^{-1}) + (1 \times 2^{-2}) + (1 \times 2^{-3})$

After simplifying the right hand side terms, we will get a decimal number, which is an equivalent of binary number on left hand side.

Octal Number System

The **base** or radix of octal number system is **8**. So, the numbers ranging from 0 to 7 are used in this number system. The part of the number that lies to the left of the **octal point** is known as integer part. Similarly, the part of the number that lies to the right of the octal point is known as fractional part.

In this number system, the successive positions to the left of the octal point having weights of 8^0, 8^1, 8^2, 8^3 and so on. Similarly, the successive positions to the right of the octal point having weights of 8^{-1}, 8^{-2}, 8^{-3} and so on. That means, each position has specific weight, which is **power of base 8**.

Example

Consider the **octal number 1457.236**. Integer part of this number is 1457 and fractional part of this number is 0.236. The digits 7, 5, 4 and 1 have weights of 8^0, 8^1, 8^2 and 8^3respectively. Similarly, the digits 2, 3 and 6 have weights of 8^{-1}, 8^{-2}, 8^{-3} respectively.

Mathematically, we can write it as

$1457.236 = (1 \times 8^3) + (4 \times 8^2) + (5 \times 8^1) + (7 \times 8^0) + (2 \times 8^{-1}) + (3 \times 8^{-2}) + (6 \times 8^{-3})$

After simplifying the right hand side terms, we will get a decimal number, which is an equivalent of octal number on left hand side.

Hexadecimal Number System

The **base** or radix of Hexa-decimal number system is **16**. So, the numbers ranging from 0 to 9 and the letters from A to F are used in this number system. The decimal equivalent of Hexa-decimal digits from A to F are 10 to 15.

The part of the number, which lies to the left of the **hexadecimal point** is known as integer part. Similarly, the part of the number, which lies to the right of the Hexa-decimal point is known as fractional part.

In this number system, the successive positions to the left of the Hexa-decimal point having weights of 16^0, 16^1, 16^2, 16^3 and so on. Similarly, the successive positions to the right of the Hexa-decimal point having weights of 16^{-1}, 16^{-2}, 16^{-3} and so on. That means, each position has specific weight, which is **power of base 16**.

Example

Consider the **Hexa-decimal number 1A05.2C4**. Integer part of this number is 1A05 and fractional part of this number is 0.2C4. The digits 5, 0, A and 1 have weights of 16^0, 16^1, 16^2 and 16^3respectively. Similarly, the digits 2, C and 4 have weights of 16^{-1}, 16^{-2} and 16^{-3} respectively.

Mathematically, we can write it as

$1A05.2C4 = (1 \times 16^3) + (10 \times 16^2) + (0 \times 16^1) + (5 \times 16^0) + (2 \times 16^{-1}) + (12 \times 16^{-2}) + (4 \times 16^{-3})$

After simplifying the right hand side terms, we will get a decimal number, which is an equivalent of Hexa-decimal number on left hand side.

In previous chapter, we have seen the four prominent number systems. In this chapter, let us convert the numbers from one number system to the other in order to find the equivalent value.

Decimal Number to other Bases Conversion

If the decimal number contains both integer part and fractional part, then convert both the parts of decimal number into other base individually. Follow these steps for converting the decimal number into its equivalent number of any base 'r'.

- Do **division** of integer part of decimal number and **successive quotients** with base 'r' and note down the remainders till the quotient is zero. Consider the remainders in reverse order to get the integer part of equivalent number of base 'r'. That means, first and last remainders denote the least significant digit and most significant digit respectively.
- Do **multiplication** of fractional part of decimal number and **successive fractions** with base 'r' and note down the carry till the result is zero or the desired number of equivalent digits is obtained. Consider the normal sequence of carry in order to get the fractional part of equivalent number of base 'r'.

Decimal to Binary Conversion

The following two types of operations take place, while converting decimal number into its equivalent binary number.

- Division of integer part and successive quotients with base 2.
- Multiplication of fractional part and successive fractions with base 2.

Example

Consider the **decimal number 58.25**. Here, the integer part is 58 and fractional part is 0.25.

Step 1 − Division of 58 and successive quotients with base 2.

Operation	Quotient	Remainder
58/2	29	**0 (LSB)**
29/2	14	**1**
14/2	7	**0**
7/2	3	**1**
3/2	1	**1**
1/2	0	**1(MSB)**

$$\Rightarrow (58)_{10} = (111010)_2$$

Therefore, the **integer part** of equivalent binary number is **111010**.

Step 2 − Multiplication of 0.25 and successive fractions with base 2.

Operation	Result	Carry
0.25 x 2	0.5	0
0.5 x 2	1.0	1
-	0.0	-

$$\Rightarrow (.25)_{10} = (.01)_2$$

Therefore, the **fractional part** of equivalent binary number is **.01**

$$\Rightarrow (58.25)_{10} = (111010.01)_2$$

Therefore, the **binary equivalent** of decimal number 58.25 is 111010.01.

Decimal to Octal Conversion

The following two types of operations take place, while converting decimal number into its equivalent octal number.

- Division of integer part and successive quotients with base 8.
- Multiplication of fractional part and successive fractions with base 8.

Example

Consider the **decimal number 58.25**. Here, the integer part is 58 and fractional part is 0.25.

Step 1 − Division of 58 and successive quotients with base 8.

Operation	Quotient	Remainder
58/8	7	**2**
7/8	0	**7**

$\Rightarrow (58)_{10} = (72)_8$

Therefore, the **integer part** of equivalent octal number is **72**.

Step 2 − Multiplication of 0.25 and successive fractions with base 8.

Operation	Result	Carry
0.25 x 8	2.00	2
-	0.00	-

$\Rightarrow (.25)_{10} = (.2)_8$

Therefore, the **fractional part** of equivalent octal number is .2

$\Rightarrow (58.25)_{10} = (72.2)_8$

Therefore, the **octal equivalent** of decimal number 58.25 is 72.2.

Decimal to Hexa-Decimal Conversion

The following two types of operations take place, while converting decimal number into its equivalent hexa-decimal number.

- Division of integer part and successive quotients with base 16.

- Multiplication of fractional part and successive fractions with base 16.

Example

Consider the **decimal number 58.25**. Here, the integer part is 58 and decimal part is 0.25.

Step 1 − Division of 58 and successive quotients with base 16.

Operation	Quotient	Remainder
58/16	3	10=A
3/16	0	3

$\Rightarrow (58)_{10} = (3A)_{16}$

Therefore, the **integer part** of equivalent Hexa-decimal number is 3A.

Step 2 − Multiplication of 0.25 and successive fractions with base 16.

Operation	Result	Carry
0.25 x 16	4.00	4
-	0.00	-

$\Rightarrow (.25)_{10} = (.4)_{16}$

Therefore, the **fractional part** of equivalent Hexa-decimal number is .4.

$$\mathbf{\Rightarrow (58.25)_{10} = (3A.4)_{16}}$$

Therefore, the **Hexa-decimal equivalent** of decimal number 58.25 is 3A.4.

Binary Number to other Bases Conversion

The process of converting a number from binary to decimal is different to the process of converting a binary number to other bases. Now, let us discuss about the conversion of a binary number to decimal, octal and Hexa-decimal number systems one by one.

Binary to Decimal Conversion

For converting a binary number into its equivalent decimal number, first multiply the bits of binary number with the respective positional weights and then add all those products.

Example

Consider the **binary number 1101.11**.

Mathematically, we can write it as

$(1101.11)_2 = (1 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (1 \times 2^0) + (1 \times 2^{-1}) + (1 \times 2^{-2})$

$\Rightarrow (1101.11)_2 = 8 + 4 + 0 + 1 + 0.5 + 0.25 = 13.75$

$\Rightarrow (1101.11)_2 = (13.75)_{10}$

Therefore, the **decimal equivalent** of binary number 1101.11 is 13.75.

Binary to Octal Conversion

We know that the bases of binary and octal number systems are 2 and 8 respectively. Three bits of binary number is equivalent to one octal digit, since $2^3 = 8$.

Follow these two steps for converting a binary number into its equivalent octal number.

- Start from the binary point and make the groups of 3 bits on both sides of binary point. If one or two bits are less while making the group of 3 bits, then include required number of zeros on extreme sides.
- Write the octal digits corresponding to each group of 3 bits.

Example

Consider the **binary number 101110.01101**.

Step 1 − Make the groups of 3 bits on both sides of binary point.

101 110.011 01

Here, on right side of binary point, the last group is having only 2 bits. So, include one zero on extreme side in order to make it as group of 3 bits.

$\Rightarrow$ 101 110.011 010

Step 2 − Write the octal digits corresponding to each group of 3 bits.

$\Rightarrow (101\ 110.011\ 010)_2 = (56.32)_8$

Therefore, the **octal equivalent** of binary number 101110.01101 is 56.32.

Binary to Hexa-Decimal Conversion

We know that the bases of binary and Hexa-decimal number systems are 2 and 16 respectively. Four bits of binary number is equivalent to one Hexa-decimal digit, since $2^4 = 16$.

Follow these two steps for converting a binary number into its equivalent Hexa-decimal number.

- Start from the binary point and make the groups of 4 bits on both sides of binary point. If some bits are less while making the group of 4 bits, then include required number of zeros on extreme sides.
- Write the Hexa-decimal digits corresponding to each group of 4 bits.

Example

Consider the **binary number 101110.01101**

Step 1 − Make the groups of 4 bits on both sides of binary point.

10 1110.0110 1

Here, the first group is having only 2 bits. So, include two zeros on extreme side in order to make it as group of 4 bits. Similarly, include three zeros on extreme side in order to make the last group also as group of 4 bits.

⇒ 0010 1110.0110 1000

Step 2 − Write the Hexa-decimal digits corresponding to each group of 4 bits.

$$\Rightarrow (0010\ 1110.0110\ 1000)_2 = (2E.68)_{16}$$

Therefore, the **Hexa-decimal equivalent** of binary number 101110.01101 is (2E.68).

Octal Number to other Bases Conversion

The process of converting a number from octal to decimal is different to the process of converting an octal number to other bases. Now, let us discuss about the conversion of an octal number to decimal, binary and Hexa-decimal number systems one by one.

Octal to Decimal Conversion

For converting an octal number into its equivalent decimal number, first multiply the digits of octal number with the respective positional weights and then add all those products.

Example

Consider the **octal number 145.23**.

Mathematically, we can write it as

$(145.23)_8 = (1 \times 8^2) + (4 \times 8^1) + (5 \times 8^0) + (2 \times 8^{-1}) + (3 \times 8^{-2})$

$\Rightarrow (145.23)_8 = 64 + 32 + 5 + 0.25 + 0.05 = 101.3$

$\Rightarrow (145.23)_8 = (101.3)_{10}$

Therefore, the **decimal equivalent** of octal number 145.23 is 101.3.

Octal to Binary Conversion

The process of converting an octal number to an equivalent binary number is just opposite to that of binary to octal conversion. By representing each octal digit with 3 bits, we will get the equivalent binary number.

Example

Consider the **octal number 145.23**.

Represent each octal digit with 3 bits.

$$(145.23)_8 = (001\ 100\ 101.010\ 011)_2$$

The value doesn't change by removing the zeros, which are on the extreme side.

$$\Rightarrow (145.23)_8 = (1100101.010011)_2$$

Therefore, the **binary equivalent** of octal number 145.23 is 1100101.010011.

Octal to Hexa-Decimal Conversion

Follow these two steps for converting an octal number into its equivalent Hexa-decimal number.

- Convert octal number into its equivalent binary number.
- Convert the above binary number into its equivalent Hexa-decimal number.

Example

Consider the **octal number 145.23**

In previous example, we got the binary equivalent of octal number 145.23 as 1100101.010011.

By following the procedure of binary to Hexa-decimal conversion, we will get

$$(1100101.010011)_2 = (65.4C)16$$

$$\Rightarrow (145.23)_8 = (65.4C)_{16}$$

Therefore, the **Hexa-decimal equivalent** of octal number 145.23 is 65.4*C*.

Hexa-Decimal Number to other Bases Conversion

The process of converting a number from Hexa-decimal to decimal is different to the process of converting Hexa-decimal number into other bases. Now, let us discuss about the conversion of Hexa-decimal number to decimal, binary and octal number systems one by one.

Hexa-Decimal to Decimal Conversion

For converting Hexa-decimal number into its equivalent decimal number, first multiply the digits of Hexa-decimal number with the respective positional weights and then add all those products.

Example

Consider the **Hexa-decimal number 1A5.2**

Mathematically, we can write it as

$$(1A5.2)_{16} = (1 \times 16^2) + (10 \times 16^1) + (5 \times 16^0) + (2 \times 16^{-1})$$

$$\Rightarrow (1A5.2)_{16} = 256 + 160 + 5 + 0.125 = 421.125$$

$$\Rightarrow (1A5.2)_{16} = (421.125)_{10}$$

Therefore, the **decimal equivalent** of Hexa-decimal number 1A5.2 is 421.125.

Hexa-Decimal to Binary Conversion

The process of converting Hexa-decimal number into its equivalent binary number is just opposite to that of binary to Hexa-decimal conversion. By representing each Hexa-decimal digit with 4 bits, we will get the equivalent binary number.

Example

Consider the **Hexa-decimal number 65.4C**

Represent each Hexa-decimal digit with 4 bits.

$$(65.4C)_6 = (0110\ 0101.0100\ 1100)_2$$

The value doesn't change by removing the zeros, which are at two extreme sides.

$\Rightarrow (65.4C)_{16} = (1100101.010011)_2$

Therefore, the **binary equivalent** of Hexa-decimal number 65.4C is 1100101.010011.

Hexa-Decimal to Octal Conversion

Follow these two steps for converting Hexa-decimal number into its equivalent octal number.

- Convert Hexa-decimal number into its equivalent binary number.
- Convert the above binary number into its equivalent octal number.

Example

Consider the **Hexa-decimal number 65.4C**

In previous example, we got the binary equivalent of Hexa-decimal number 65.4C as 1100101.010011.

By following the procedure of binary to octal conversion, we will get

$$(1100101.010011)_2 = (145.23)_8$$

$$\Rightarrow (65.4C)_{16} = (145.23)_8$$

Therefore, the **octal equivalent** of Hexa-decimal number 65.4*C*is 145.23.

We can make the binary numbers into the following two groups − **Unsigned numbers** and **Signed numbers**.

Unsigned Numbers

Unsigned numbers contain only magnitude of the number. They don't have any sign. That means all unsigned binary numbers are positive. As in decimal number system, the placing of positive sign in front of the number is optional for representing positive numbers. Therefore, all positive numbers including zero can be treated as unsigned numbers if positive sign is not assigned in front of the number.

Signed Numbers

Signed numbers contain both sign and magnitude of the number. Generally, the sign is placed in front of number. So, we have to consider the positive sign for positive numbers and negative sign for negative numbers. Therefore, all numbers can be treated as signed numbers if the corresponding sign is assigned in front of the number.

If sign bit is zero, which indicates the binary number is positive. Similarly, if sign bit is one, which indicates the binary number is negative.

Representation of Un-Signed Binary Numbers

The bits present in the un-signed binary number holds the **magnitude** of a number. That means, if the un-signed binary number contains **'N'** bits, then all **N** bits represent the magnitude of the number, since it doesn't have any sign bit.

Example

Consider the **decimal number 108**. The binary equivalent of this number is **1101100**. This is the representation of unsigned binary number.

$$(108)_{10} = (1101100)_2$$

It is having 7 bits. These 7 bits represent the magnitude of the number 108.

Representation of Signed Binary Numbers

The Most Significant Bit (MSB) of signed binary numbers is used to indicate the sign of the numbers. Hence, it is also called as **sign bit**. The positive sign is represented by placing '0' in the sign bit. Similarly, the negative sign is represented by placing '1' in the sign bit.

If the signed binary number contains 'N' bits, then (N-1) bits only represent the magnitude of the number since one bit (MSB) is reserved for representing sign of the number.

There are three **types of representations** for signed binary numbers

- Sign-Magnitude form
- 1's complement form
- 2's complement form

Representation of a positive number in all these 3 forms is same. But, only the representation of negative number will differ in each form.

Example

Consider the **positive decimal number +108**. The binary equivalent of magnitude of this number is 1101100. These 7 bits represent the magnitude of the number 108. Since it is positive number, consider the sign bit as zero, which is placed on left most side of magnitude.

$$(+108)_{10} = (01101100)_2$$

Therefore, the **signed binary representation** of positive decimal number +108 is $\mathbf{01101100}$. So, the same representation is valid in sign-magnitude form, 1's complement form and 2's complement form for positive decimal number +108.

Sign-Magnitude form

In sign-magnitude form, the MSB is used for representing **sign**of the number and the remaining bits represent the **magnitude**of the number. So, just include sign bit at the left most side of unsigned binary number. This representation is similar to the signed decimal numbers representation.

Example

Consider the **negative decimal number -108**. The magnitude of this number is 108. We know the unsigned binary representation of 108 is 1101100. It is having 7 bits. All these bits represent the magnitude.

Since the given number is negative, consider the sign bit as one, which is placed on left most side of magnitude.

$$(-108)_{10} = (11101100)_2$$

Therefore, the sign-magnitude representation of -108 is **11101100**.

1's complement form

The 1's complement of a number is obtained by **complementing all the bits** of signed binary number. So, 1's complement of positive number gives a negative number. Similarly, 1's complement of negative number gives a positive number.

That means, if you perform two times 1's complement of a binary number including sign bit, then you will get the original signed binary number.

Example

Consider the **negative decimal number -108**. The magnitude of this number is 108. We know the signed binary representation of 108 is 01101100.

It is having 8 bits. The MSB of this number is zero, which indicates positive number. Complement of zero is one and vice-versa. So, replace zeros by ones and ones by zeros in order to get the negative number.

$$(-108)_{10} = (10010011)_2$$

Therefore, the **1's complement of $(108)_{10}$** is **$(10010011)_2$**.

2's complement form

The 2's complement of a binary number is obtained by **adding one to the 1's complement** of signed binary number. So, 2's complement of positive number gives a negative number. Similarly, 2's complement of negative number gives a positive number.

That means, if you perform two times 2's complement of a binary number including sign bit, then you will get the original signed binary number.

Example

Consider the **negative decimal number -108**.

We know the 1's complement of $(\mathbf{108})_{\mathbf{10}}$ is $(\mathbf{10010011})_{\mathbf{2}}$

2's compliment of $(108)_{10}$ = 1's compliment of $(108)_{10}$ + 1.

= 10010011 + 1

= 10010100

Therefore, the **2's complement of $(108)_{10}$** is **$(10010100)_2$**.

In this chapter, let us discuss about the basic arithmetic operations, which can be performed on any two signed binary numbers using 2's complement method. The **basic arithmetic operations** are addition and subtraction.

Addition of two Signed Binary Numbers

Consider the two signed binary numbers A & B, which are represented in 2's complement form. We can perform the **addition** of these two numbers, which is similar to the addition of two unsigned binary numbers. But, if the resultant sum contains carry out from sign bit, then discard (ignore) it in order to get the correct value.

If resultant sum is positive, you can find the magnitude of it directly. But, if the resultant sum is negative, then take 2's complement of it in order to get the magnitude.

Example 1

Let us perform the **addition** of two decimal numbers **+7 and +4** using 2's complement method.

The **2's complement** representations of +7 and +4 with 5 bits each are shown below.

$$(+7)_{10} = (00111)_2$$

$$(+4)_{10} = (00100)_2$$

The addition of these two numbers is

$$(+7)_{10} + (+4)_{10} = (00111)_2 + (00100)_2$$

$$\Rightarrow (+7)_{10} + (+4)_{10} = (01011)_2.$$

The resultant sum contains 5 bits. So, there is no carry out from sign bit. The sign bit '0' indicates that the resultant sum is **positive**. So, the magnitude of sum is 11 in decimal number system. Therefore, addition of two positive numbers will give another positive number.

Example 2

Let us perform the **addition** of two decimal numbers **-7** and **-4**using 2's complement method.

The **2's complement** representation of -7 and -4 with 5 bits each are shown below.

$$(-7)_{10} = (11001)_2$$

$$(-4)_{10} = (11100)_2$$

The addition of these two numbers is

$$(-7)_{10} + (-4)_{10} = (11001)_2 + (11100)_2$$

$$\Rightarrow (-7)_{10} + (-4)_{10} = (110101)_2.$$

The resultant sum contains 6 bits. In this case, carry is obtained from sign bit. So, we can remove it

Resultant sum after removing carry is $(-7)_{10} + (-4)_{10} = \mathbf{(10101)_2}$.

The sign bit '1' indicates that the resultant sum is **negative**. So, by taking 2's complement of it we will get the magnitude of resultant sum as 11 in decimal number system. Therefore, addition of two negative numbers will give another negative number.

Subtraction of two Signed Binary Numbers

Consider the two signed binary numbers A & B, which are represented in 2's complement form. We know that 2's complement of positive number gives a negative number. So, whenever we have to subtract a number B from number A, then take 2's complement of B and add it to A. So, **mathematically**we can write it as

A - B = A + ***(2's complement of B)***

Similarly, if we have to subtract the number A from number B, then take 2's complement of A and add it to B. So, **mathematically** we can write it as

B - A = B + ***(2's complement of A)***

So, the subtraction of two signed binary numbers is similar to the addition of two signed binary numbers. But, we have to take 2's complement of the number, which is supposed to be subtracted. This is the **advantage** of 2's complement technique. Follow, the same rules of addition of two signed binary numbers.

Example 3

Let us perform the **subtraction** of two decimal numbers **+7 and +4** using 2's complement method.

The subtraction of these two numbers is

$$(+7)_{10} - (+4)_{10} = (+7)_{10} + (-4)_{10}.$$

The **2's complement** representation of +7 and -4 with 5 bits each are shown below.

$$(+7)_{10} = (00111)_2$$

$$(+4)_{10} = (11100)_2$$

$\Rightarrow (+7)_{10} + (+4)_{10} = (00111)_2 + (11100)_2 = (00011)_2$

Here, the carry obtained from sign bit. So, we can remove it. The resultant sum after removing carry is

$(+7)_{10} + (+4)_{10} = \mathbf{(00011)_2}$

The sign bit '0' indicates that the resultant sum is **positive**. So, the magnitude of it is 3 in decimal number system. Therefore, subtraction of two decimal numbers +7 and +4 is +3.

Example 4

Let us perform the **subtraction of** two decimal numbers **+4** and **+7** using 2's complement method.

The subtraction of these two numbers is

$$(+4)_{10} - (+7)_{10} = (+4)_{10} + (-7)_{10}.$$

The **2's complement** representation of +4 and -7 with 5 bits each are shown below.

$$(+4)_{10} = (00100)_2$$

$$(-7)_{10} = (11001)_2$$

$\Rightarrow (+4)_{10} + (-7)_{10} = (00100)_2 + (11001)_2 = (11101)_2$

Here, carry is not obtained from sign bit. The sign bit '1' indicates that the resultant sum is **negative**. So, by taking 2's complement of it we will get the magnitude of resultant sum as 3 in decimal number system. Therefore, subtraction of two decimal numbers +4 and +7 is -3.

In the coding, when numbers or letters are represented by a specific group of symbols, it is said to be that number or letter is being encoded. The group of symbols is called as **code**. The digital data is represented, stored and transmitted as group of bits. This group of bits is also called as **binary code**.

Binary codes can be classified into two types.

- Weighted codes
- Unweighted codes

If the code has positional weights, then it is said to be **weighted code**. Otherwise, it is an unweighted code. Weighted codes can be further classified as positively weighted codes and negatively weighted codes.

Binary Codes for Decimal digits

The following table shows the various binary codes for decimal digits 0 to 9.

Decimal Digit	8421 Code	2421 Code	84-2-1 Code	Excess 3 Code
0	0000	0000	0000	0011
1	0001	0001	0111	0100
2	0010	0010	0110	0101
3	0011	0011	0101	0110
4	0100	0100	0100	0111
5	0101	1011	1011	1000
6	0110	1100	1010	1001
7	0111	1101	1001	1010
8	1000	1110	1000	1011
9	1001	1111	1111	1100

We have 10 digits in decimal number system. To represent these 10 digits in binary, we require minimum of 4 bits. But, with 4 bits there will be 16 unique combinations of zeros and ones. Since, we have only 10 decimal digits, the other 6 combinations of zeros and ones are not required.

8 4 2 1 code

- The weights of this code are 8, 4, 2 and 1.
- This code has all positive weights. So, it is a **positively weighted code**.
- This code is also called as **natural BCD** (Binary Coded Decimal) **code**.

Example

Let us find the BCD equivalent of the decimal number 786. This number has 3 decimal digits 7, 8 and 6. From the table, we can write the BCD (8421) codes of 7, 8 and 6 are 0111, 1000 and 0110 respectively.

$$\therefore (786)_{10} = (011110000110)_{BCD}$$

There are 12 bits in BCD representation, since each BCD code of decimal digit has 4 bits.

2 4 2 1 code

- The weights of this code are 2, 4, 2 and 1.
- This code has all positive weights. So, it is a **positively weighted code**.
- It is an **unnatural BCD** code. Sum of weights of unnatural BCD codes is equal to 9.
- It is a **self-complementing** code. Self-complementing codes provide the 9's complement of a decimal number, just by interchanging 1's and 0's in its equivalent 2421 representation.

Example

Let us find the 2421 equivalent of the decimal number 786. This number has 3 decimal digits 7, 8 and 6. From the table, we can write the 2421 codes of 7, 8 and 6 are 1101, 1110 and 1100 respectively.

Therefore, the 2421 equivalent of the decimal number 786 is **110111101100**.

8 4 -2 -1 code

- The weights of this code are 8, 4, -2 and -1.
- This code has negative weights along with positive weights. So, it is a **negatively weighted code**.
- It is an **unnatural BCD** code.
- It is a **self-complementing** code.

Example

Let us find the 8 4-2-1 equivalent of the decimal number 786. This number has 3 decimal digits 7, 8 and 6. From the table, we can write the 8 4 -2 -1 codes of 7, 8 and 6 are 1001, 1000 and 1010 respectively.

Therefore, the 8 4 -2 -1 equivalent of the decimal number 786 is **100110001010**.

Excess 3 code

- This code doesn't have any weights. So, it is an **un-weighted code**.
- We will get the Excess 3 code of a decimal number by adding three (0011) to the binary equivalent of that decimal number. Hence, it is called as Excess 3 code.
- It is a **self-complementing** code.

Example

Let us find the Excess 3 equivalent of the decimal number 786. This number has 3 decimal digits 7, 8 and 6. From the table, we can write the Excess 3 codes of 7, 8 and 6 are 1010, 1011 and 1001 respectively.

Therefore, the Excess 3 equivalent of the decimal number 786 is **101010111001**

Gray Code

The following table shows the 4-bit Gray codes corresponding to each 4-bit binary code.

Decimal Number	Binary Code	Gray Code
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101

7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

- This code doesn't have any weights. So, it is an **un-weighted code**.
- In the above table, the successive Gray codes are differed in one bit position only. Hence, this code is called as **unit distance** code.

Binary code to Gray Code Conversion

Follow these steps for converting a binary code into its equivalent Gray code.

- Consider the given binary code and place a zero to the left of MSB.
- Compare the successive two bits starting from zero. If the 2 bits are same, then the output is zero. Otherwise, output is one.
- Repeat the above step till the LSB of Gray code is obtained.

Example

From the table, we know that the Gray code corresponding to binary code 1000 is 1100. Now, let us verify it by using the above procedure.

Given, binary code is 1000.

Step 1 − By placing zero to the left of MSB, the binary code will be 01000.

Step 2 − By comparing successive two bits of new binary code, we will get the gray code as **1100**.

Boolean Algebra is an algebra, which deals with binary numbers & binary variables. Hence, it is also called as Binary Algebra or logical Algebra. A mathematician, named George Boole had developed this algebra in 1854. The variables used in this algebra are also called as Boolean variables.

The range of voltages corresponding to Logic 'High' is represented with '1' and the range of voltages corresponding to logic 'Low' is represented with '0'.

Postulates and Basic Laws of Boolean Algebra

In this section, let us discuss about the Boolean postulates and basic laws that are used in Boolean algebra. These are useful in minimizing Boolean functions.

Boolean Postulates

Consider the binary numbers 0 and 1, Boolean variable (x) and its complement (x'). Either the Boolean variable or complement of it is known as **literal**. The four possible **logical OR** operations among these literals and binary numbers are shown below.

$$x + 0 = x$$

$$x + 1 = 1$$

$$x + x = x$$

$$x + x' = 1$$

Similarly, the four possible **logical AND** operations among those literals and binary numbers are shown below.

$$x.1 = x$$

$$x.0 = 0$$

$$x.x = x$$

$$x.x' = 0$$

These are the simple Boolean postulates. We can verify these postulates easily, by substituting the Boolean variable with '0' or '1'.

Note− The complement of complement of any Boolean variable is equal to the variable itself. i.e., $(x')'=x$.

Basic Laws of Boolean Algebra

Following are the three basic laws of Boolean Algebra.

- Commutative law
- Associative law
- Distributive law

Commutative Law

If any logical operation of two Boolean variables give the same result irrespective of the order of those two variables, then that logical operation is said to be **Commutative**. The logical OR & logical AND operations of two Boolean variables x & y are shown below

$$x + y = y + x$$

$$x.y = y.x$$

The symbol '+' indicates logical OR operation. Similarly, the symbol '.' indicates logical AND operation and it is optional to represent. Commutative law obeys for logical OR & logical AND operations.

Associative Law

If a logical operation of any two Boolean variables is performed first and then the same operation is performed with the remaining variable gives the same result, then that logical operation is said to be **Associative**. The logical OR & logical AND operations of three Boolean variables x, y & z are shown below.

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

$$x.(y.z) = (x.y).z$$

Associative law obeys for logical OR & logical AND operations.

Distributive Law

If any logical operation can be distributed to all the terms present in the Boolean function, then that logical operation is said to be **Distributive**. The distribution of logical OR & logical AND operations of three Boolean variables x, y & z are shown below.

$$x.(y + z) = x.y + x.z$$

$$x + (y.z) = (x + y).(x + z)$$

Distributive law obeys for logical OR and logical AND operations.

These are the Basic laws of Boolean algebra. We can verify these laws easily, by substituting the Boolean variables with '0' or '1'.

Theorems of Boolean Algebra

The following two theorems are used in Boolean algebra.

- Duality theorem
- DeMorgan's theorem

Duality Theorem

This theorem states that the **dual** of the Boolean function is obtained by interchanging the logical AND operator with logical OR operator and zeros with ones. For every Boolean function, there will be a corresponding Dual function.

Let us make the Boolean equations (relations) that we discussed in the section of Boolean postulates and basic laws into two groups. The following table shows these two groups.

Group1	Group2
x + 0 = x	x.1 = x
x + 1 = 1	x.0 = 0
x + x = x	x.x = x
x + x' = 1	x.x' = 0
x + y = y + x	x.y = y.x
x + (y + z) = (x + y) + z	x.(y.z) = (x.y).z
x.(y + z) = x.y + x.z	x + (y.z) = (x + y).(x + z)

In each row, there are two Boolean equations and they are dual to each other. We can verify all these Boolean equations of Group1 and Group2 by using duality theorem.

DeMorgan's Theorem

This theorem is useful in finding the **complement of Boolean function**. It states that the complement of logical OR of at least two Boolean variables is equal to the logical AND of each complemented variable.

DeMorgan's theorem with 2 Boolean variables x and y can be represented as

$$(x + y)' = x'.y'$$

The dual of the above Boolean function is

$$(x.y)' = x' + y'$$

Therefore, the complement of logical AND of two Boolean variables is equal to the logical OR of each complemented variable. Similarly, we can apply DeMorgan's theorem for more than 2 Boolean variables also.

Simplification of Boolean Functions

Till now, we discussed the postulates, basic laws and theorems of Boolean algebra. Now, let us simplify some Boolean functions.

Example 1

Let us **simplify** the Boolean function, f = p'qr + pq'r + pqr' + pqr

We can simplify this function in two methods.

Method 1

Given Boolean function, f = p'qr + pq'r + pqr' +pqr.

Step 1 − In first and second terms r is common and in third and fourth terms pq is common. So, take the common terms by using **Distributive law**.

$$\Rightarrow f = (p'q + pq')r + pq(r' + r)$$

Step 2 − The terms present in first parenthesis can be simplified to Ex-OR operation. The terms present in second parenthesis can be simplified to '1' using **Boolean postulate**

$$\Rightarrow f = (p \oplus q)r + pq(1)$$

Step 3 − The first term can't be simplified further. But, the second term can be simplified to pq using **Boolean postulate**.

$$\Rightarrow f = (p \oplus q)r + pq$$

Therefore, the simplified Boolean function is **$f = (p \oplus q)r + pq$**

Method 2

Given Boolean function, f = p'qr + pq'r + pqr' + pqr.

Step 1 − Use the **Boolean postulate**, x + x = x. That means, the Logical OR operation with any Boolean variable 'n' times will be equal to the same variable. So, we can write the last term pqr two more times.

$$\Rightarrow f = p'qr + pq'r + pqr' + pqr + pqr + pqr$$

Step 2 − Use **Distributive law** for 1st and 4th terms, 2nd and 5th terms, 3rd and 6th terms.

$$\Rightarrow f = qr(p' + p) + pr(q' + q) + pq(r' + r)$$

Step 3 − Use **Boolean postulate**, x + x' = 1 for simplifying the terms present in each parenthesis.

$$\Rightarrow f = qr(1) + pr(1) + pq(1)$$

Step 4 − Use **Boolean postulate**, x.1 = x for simplifying the above three terms.

$$\Rightarrow f = qr + pr + pq$$

$$\Rightarrow f = pq + qr + pr$$

Therefore, the simplified Boolean function is **f = pq + qr + pr**.

So, we got two different Boolean functions after simplifying the given Boolean function in each method. Functionally, those two Boolean functions are same. So, based on the requirement, we can choose one of those two Boolean functions.

Example 2

Let us find the **complement** of the Boolean function, f = p'q + pq'.

The complement of Boolean function is f' = (p'q + pq')'.

Step 1 − Use DeMorgan's theorem, (x + y)' = x'.y'.

$$\Rightarrow f' = (p'q)'.(pq')'$$

Step 2 − Use DeMorgan's theorem, (x.y)' = x' + y'

$$\Rightarrow f' = \{(p')' + q'\}.\{p' + (q')'\}$$

Step3 − Use the Boolean postulate, (x')'=x.

$$\Rightarrow f' = \{p + q'\}.\{p' + q\}$$

$$\Rightarrow f' = pp' + pq + p'q' + qq'$$

Step 4 − Use the Boolean postulate, xx'=0.

$$\Rightarrow f = 0 + pq + p'q' + 0$$

$$\Rightarrow f = pq + p'q'$$

Therefore, the **complement** of Boolean function, p'q + pq' is **pq + p'q'**.

We will get four Boolean product terms by combining two variables x and y with logical AND operation. These Boolean product terms are called as **min terms** or **standard product terms**. The min terms are x'y', x'y, xy' and xy.

Similarly, we will get four Boolean sum terms by combining two variables x and y with logical OR operation. These Boolean sum terms are called as **Max terms** or **standard sum terms**. The Max terms are x + y, x + y', x' + y and x' + y'.

The following table shows the representation of min terms and MAX terms for 2 variables.

x	y	Min terms	Max terms
0	0	$m_0=x'y'$	$M_0=x + y$
0	1	$m_1=x'y$	$M_1=x + y'$
1	0	$m_2=xy'$	$M_2=x' + y$
1	1	$m_3=xy$	$M_3=x' + y'$

If the binary variable is '0', then it is represented as complement of variable in min term and as the variable itself in Max term. Similarly, if the binary variable is '1', then it is represented as complement of variable in Max term and as the variable itself in min term.

From the above table, we can easily notice that min terms and Max terms are complement of each other. If there are 'n' Boolean variables, then there will be 2^n min terms and 2^n Max terms.

Canonical SoP and PoS forms

A truth table consists of a set of inputs and output(s). If there are 'n' input variables, then there will be 2^n possible combinations with zeros and ones. So the value of each output variable depends on the combination of input variables. So, each output variable will have '1' for some combination of input variables and '0' for some other combination of input variables.

Therefore, we can express each output variable in following two ways.

- Canonical SoP form
- Canonical PoS form

Canonical SoP form

Canonical SoP form means Canonical Sum of Products form. In this form, each product term contains all literals. So, these product terms are nothing but the min terms. Hence, canonical SoP form is also called as **sum of min terms** form.

First, identify the min terms for which, the output variable is one and then do the logical OR of those min terms in order to get the Boolean expression (function) corresponding to that output variable. This Boolean function will be in the form of sum of min terms.

Follow the same procedure for other output variables also, if there is more than one output variable.

Example

Consider the following **truth table**.

Inputs		**Output**	
p	**q**	**r**	**f**
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Here, the output (f) is '1' for four combinations of inputs. The corresponding min terms are p'qr, pq'r, pqr', pqr. By doing logical OR of these four min terms, we will get the Boolean function of output (f).

Therefore, the Boolean function of output is, f = p'qr + pq'r + pqr' + pqr. This is the **canonical SoP form** of output, f. We can also represent this function in following two notations.

$$f=m3+m5+m6+m7f=m3+m5+m6+m7$$

$$f=\sum m(3,5,6,7)f=\sum m(3,5,6,7)$$

In one equation, we represented the function as sum of respective min terms. In other equation, we used the symbol for summation of those min terms.

Canonical PoS form

Canonical PoS form means Canonical Product of Sums form. In this form, each sum term contains all literals. So, these sum terms are nothing but the Max terms. Hence, canonical PoS form is also called as **product of Max terms** form.

First, identify the Max terms for which, the output variable is zero and then do the logical AND of those Max terms in order to get the Boolean expression (function) corresponding to that output variable. This Boolean function will be in the form of product of Max terms.

Follow the same procedure for other output variables also, if there is more than one output variable.

Example

Consider the same truth table of previous example. Here, the output (f) is '0' for four combinations of inputs. The corresponding Max terms are p + q + r, p + q + r', p + q' + r, p' + q + r. By doing logical AND of these four Max terms, we will get the Boolean function of output (f).

Therefore, the Boolean function of output is, f = (p + q + r).(p + q + r').(p + q' + r).(p' + q + r). This is the **canonical PoS form**of output, f. We can also represent this function in following two notations.

$$f=M0.M1.M2.M4f=M0.M1.M2.M4$$

$$f=\prod M(0,1,2,4)f=\prod M(0,1,2,4)$$

In one equation, we represented the function as product of respective Max terms. In other equation, we used the symbol for multiplication of those Max terms.

The Boolean function, f = (p + q + r).(p + q + r').(p + q' + r).(p' + q + r) is the dual of the Boolean function, f = p'qr + pq'r + pqr' + pqr.

Therefore, both canonical SoP and canonical PoS forms are **Dual**to each other. Functionally, these two forms are same. Based on the requirement, we can use one of these two forms.

Standard SoP and PoS forms

We discussed two canonical forms of representing the Boolean output(s). Similarly, there are two standard forms of representing the Boolean output(s). These are the simplified version of canonical forms.

- Standard SoP form
- Standard PoS form

We will discuss about Logic gates in later chapters. The main **advantage** of standard forms is that the number of inputs applied to logic gates can be minimized. Sometimes, there will be reduction in the total number of logic gates required.

Standard SoP form

Standard SoP form means **Standard Sum of Products** form. In this form, each product term need not contain all literals. So, the product terms may or may not be the min terms. Therefore, the Standard SoP form is the simplified form of canonical SoP form.

We will get Standard SoP form of output variable in two steps.

- Get the canonical SoP form of output variable
- Simplify the above Boolean function, which is in canonical SoP form.

Follow the same procedure for other output variables also, if there is more than one output variable. Sometimes, it may not possible to simplify the canonical SoP form. In that case, both canonical and standard SoP forms are same.

Example

Convert the following Boolean function into Standard SoP form.

$$f = p'qr + pq'r + pqr' + pqr$$

The given Boolean function is in canonical SoP form. Now, we have to simplify this Boolean function in order to get standard SoP form.

Step 1 − Use the **Boolean postulate**, x + x = x. That means, the Logical OR operation with any Boolean variable 'n' times will be equal to the same variable. So, we can write the last term pqr two more times.

$$\Rightarrow f = p'qr + pq'r + pqr' + pqr + pqr + pqr$$

Step 2 − Use **Distributive law** for 1^{st} and 4^{th} terms, 2^{nd} and 5^{th} terms, 3^{rd} and 6^{th} terms.

$$\Rightarrow f = qr(p' + p) + pr(q' + q) + pq(r' + r)$$

Step 3 − Use **Boolean postulate**, x + x' = 1 for simplifying the terms present in each parenthesis.

$$\Rightarrow f = qr(1) + pr(1) + pq(1)$$

Step 4 − Use **Boolean postulate**, x.1 = x for simplifying above three terms.

$$\Rightarrow f = qr + pr + pq$$

$$\Rightarrow f = pq + qr + pr$$

This is the simplified Boolean function. Therefore, the **standard SoP form** corresponding to given canonical SoP form is **f = pq + qr + pr**

Standard PoS form

Standard PoS form means **Standard Product of Sums** form. In this form, each sum term need not contain all literals. So, the sum terms may or may not be the Max terms. Therefore, the Standard PoS form is the simplified form of canonical PoS form.

We will get Standard PoS form of output variable in two steps.

- Get the canonical PoS form of output variable
- Simplify the above Boolean function, which is in canonical PoS form.

Follow the same procedure for other output variables also, if there is more than one output variable. Sometimes, it may not possible to simplify the canonical PoS form. In that case, both canonical and standard PoS forms are same.

Example

Convert the following Boolean function into Standard PoS form.

$$f = (p + q + r).(p + q + r').(p + q' + r).(p' + q + r)$$

The given Boolean function is in canonical PoS form. Now, we have to simplify this Boolean function in order to get standard PoS form.

Step 1 − Use the **Boolean postulate**, x.x = x. That means, the Logical AND operation with any Boolean variable 'n' times will be equal to the same variable. So, we can write the first term p+q+r two more times.

$$\Rightarrow f = (p + q + r).(p + q + r).(p + q + r).(p + q + r').(p +q' + r).(p' + q + r)$$

Step 2 − Use **Distributive law,** x + (y.z) = (x + y).(x + z) for 1^{st} and 4^{th} parenthesis, 2^{nd} and 5^{th} parenthesis, 3^{rd} and 6^{th}parenthesis.

$$\Rightarrow f = (p + q + rr').(p + r + qq').(q + r + pp')$$

Step 3 − Use **Boolean postulate**, x.x'=0 for simplifying the terms present in each parenthesis.

$$\Rightarrow f = (p + q + 0).(p + r + 0).(q + r + 0)$$

Step 4 − Use **Boolean postulate**, x + 0 = x for simplifying the terms present in each parenthesis

$$\Rightarrow f = (p + q).(p + r).(q + r)$$

$$\Rightarrow f = (p + q).(q + r).(p + r)$$

This is the simplified Boolean function. Therefore, the **standard PoS form** corresponding to given canonical PoS form is **f = (p + q).(q + r).(p + r)**. This is the **dual** of the Boolean function, f = pq + qr + pr.

Therefore, both Standard SoP and Standard PoS forms are Dual to each other.

Digital electronic circuits operate with voltages of **two logic levels**namely Logic Low and Logic High. The range of voltages corresponding to Logic Low is represented with '0'. Similarly, the range of voltages corresponding to Logic High is represented with '1'.

The basic digital electronic circuit that has one or more inputs and single output is known as **Logic gate**. Hence, the Logic gates are the building blocks of any digital system. We can classify these Logic gates into the following three categories.

- Basic gates
- Universal gates
- Special gates

Now, let us discuss about the Logic gates come under each category one by one.

Basic Gates

In earlier chapters, we learnt that the Boolean functions can be represented either in sum of products form or in product of sums form based on the requirement. So, we can implement these Boolean functions by using basic gates. The basic gates are AND, OR & NOT gates.

AND gate

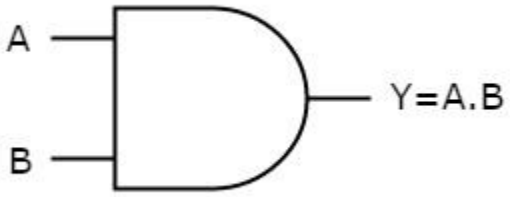
An AND gate is a digital circuit that has two or more inputs and produces an output, which is the **logical AND** of all those inputs. It is optional to represent the **Logical AND** with the symbol '.'.

The following table shows the **truth table** of 2-input AND gate.

A	B	Y = A.B
0	0	0
0	1	0
1	0	0
1	1	1

Here A, B are the inputs and Y is the output of two input AND gate. If both inputs are '1', then only the output, Y is '1'. For remaining combinations of inputs, the output, Y is '0'.

The following figure shows the **symbol** of an AND gate, which is having two inputs A, B and one output, Y.



This AND gate produces an output (Y), which is the **logical AND**of two inputs A, B. Similarly, if there are 'n' inputs, then the AND gate produces an output, which is the logical AND of all those inputs. That means, the output of AND gate will be '1', when all the inputs are '1'.

OR gate

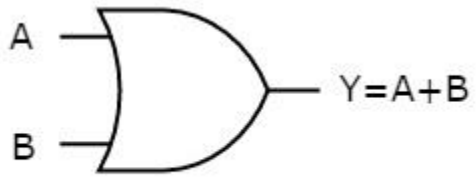
An OR gate is a digital circuit that has two or more inputs and produces an output, which is the logical OR of all those inputs. This **logical OR** is represented with the symbol '+'.

The following table shows the **truth table** of 2-input OR gate.

A	B	Y = A + B
0	0	0
0	1	1
1	0	1
1	1	1

Here A, B are the inputs and Y is the output of two input OR gate. If both inputs are '0', then only the output, Y is '0'. For remaining combinations of inputs, the output, Y is '1'.

The following figure shows the **symbol** of an OR gate, which is having two inputs A, B and one output, Y.



This OR gate produces an output (Y), which is the **logical OR** of two inputs A, B. Similarly, if there are 'n' inputs, then the OR gate produces an output, which is the logical OR of all those inputs. That means, the output of an OR gate will be '1', when at least one of those inputs is '1'.

NOT gate

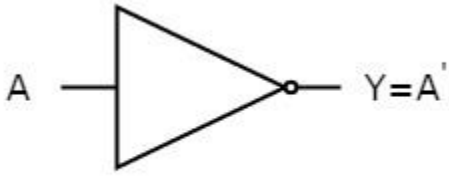
A NOT gate is a digital circuit that has single input and single output. The output of NOT gate is the **logical inversion** of input. Hence, the NOT gate is also called as inverter.

The following table shows the **truth table** of NOT gate.

A	Y = A'
0	1
1	0

Here A and Y are the input and output of NOT gate respectively. If the input, A is '0', then the output, Y is '1'. Similarly, if the input, A is '1', then the output, Y is '0'.

The following figure shows the **symbol** of NOT gate, which is having one input, A and one output, Y.



This NOT gate produces an output (Y), which is the **complement** of input, A.

Universal gates

NAND & NOR gates are called as **universal gates**. Because we can implement any Boolean function, which is in sum of products form by using NAND gates alone. Similarly, we can implement any Boolean function, which is in product of sums form by using NOR gates alone.

NAND gate

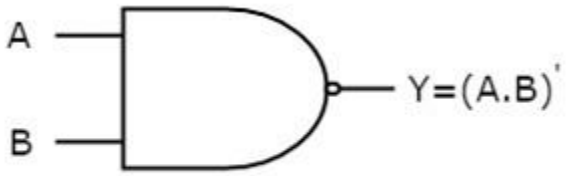
NAND gate is a digital circuit that has two or more inputs and produces an output, which is the **inversion of logical AND** of all those inputs.

The following table shows the **truth table** of 2-input NAND gate.

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

Here A, B are the inputs and Y is the output of two input NAND gate. When both inputs are '1', the output, Y is '0'. If at least one of the input is zero, then the output, Y is '1'. This is just opposite to that of two input AND gate operation.

The following image shows the **symbol** of NAND gate, which is having two inputs A, B and one output, Y.



NAND gate operation is same as that of AND gate followed by an inverter. That's why the NAND gate symbol is represented like that.

NOR gate

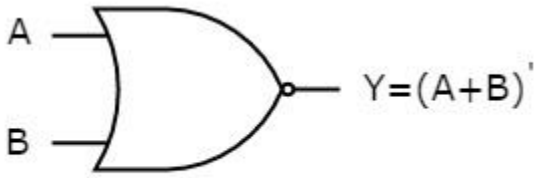
NOR gate is a digital circuit that has two or more inputs and produces an output, which is the **inversion of logical OR** of all those inputs.

The following table shows the **truth table** of 2-input NOR gate

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

Here A, B are the inputs and Y is the output. If both inputs are '0', then the output, Y is '1'. If at least one of the input is '1', then the output, Y is '0'. This is just opposite to that of two input OR gate operation.

The following figure shows the **symbol** of NOR gate, which is having two inputs A, B and one output, Y.



NOR gate operation is same as that of OR gate followed by an inverter. That's why the NOR gate symbol is represented like that.

Special Gates

Ex-OR & Ex-NOR gates are called as special gates. Because, these two gates are special cases of OR & NOR gates.

Ex-OR gate

The full form of Ex-OR gate is **Exclusive-OR** gate. Its function is same as that of OR gate except for some cases, when the inputs having even number of ones.

The following table shows the **truth table** of 2-input Ex-OR gate.

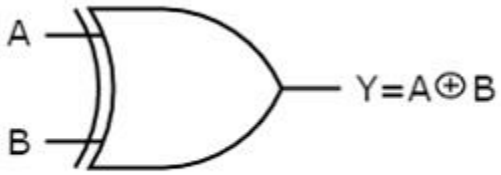
A	B	Y = A⊕B
0	0	0

0	1	1
1	0	1
1	1	0

Here A, B are the inputs and Y is the output of two input Ex-OR gate. The truth table of Ex-OR gate is same as that of OR gate for first three rows. The only modification is in the fourth row. That means, the output (Y) is zero instead of one, when both the inputs are one, since the inputs having even number of ones.

Therefore, the output of Ex-OR gate is '1', when only one of the two inputs is '1'. And it is zero, when both inputs are same.

Below figure shows the **symbol** of Ex-OR gate, which is having two inputs A, B and one output, Y.



Ex-OR gate operation is similar to that of OR gate, except for few combination(s) of inputs. That's why the Ex-OR gate symbol is represented like that. The output of Ex-OR gate is '1', when odd number of ones present at the inputs. Hence, the output of Ex-OR gate is also called as an **odd function**.

Ex-NOR gate

The full form of Ex-NOR gate is **Exclusive-NOR** gate. Its function is same as that of NOR gate except for some cases, when the inputs having even number of ones.

The following table shows the **truth table** of 2-input Ex-NOR gate.

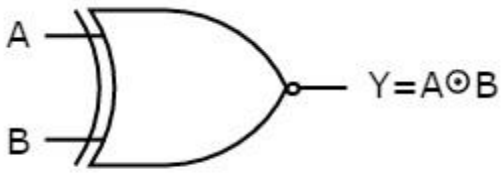
A	B	Y = A⊙B
0	0	1
0	1	0
1	0	0

1	1	1

Here A, B are the inputs and Y is the output. The truth table of Ex-NOR gate is same as that of NOR gate for first three rows. The only modification is in the fourth row. That means, the output is one instead of zero, when both the inputs are one.

Therefore, the output of Ex-NOR gate is '1', when both inputs are same. And it is zero, when both the inputs are different.

The following figure shows the **symbol** of Ex-NOR gate, which is having two inputs A, B and one output, Y.



Ex-NOR gate operation is similar to that of NOR gate, except for few combination(s) of inputs. That's why the Ex-NOR gate symbol is represented like that. The output of Ex-NOR gate is '1', when even number of ones present at the inputs. Hence, the output of Ex-NOR gate is also called as an **even function**.

From the above truth tables of Ex-OR & Ex-NOR logic gates, we can easily notice that the Ex-NOR operation is just the logical inversion of Ex-OR operation.

The maximum number of levels that are present between inputs and output is two in **two level logic**. That means, irrespective of total number of logic gates, the maximum number of Logic gates that are present (cascaded) between any input and output is two in two level logic. Here, the outputs of first level Logic gates are connected as inputs of second level Logic gate(s).

Consider the four Logic gates AND, OR, NAND & NOR. Since, there are 4 Logic gates, we will get 16 possible ways of realizing two level logic. Those are AND-AND, AND-OR, ANDNAND, AND-NOR, OR-AND, OR-OR, OR-NAND, OR-NOR, NAND-AND, NAND-OR, NANDNAND, NAND-NOR, NOR-AND, NOR-OR, NOR-NAND, NOR-NOR.

These two level logic realizations can be classified into the following two categories.

- Degenerative form
- Non-degenerative form

Degenerative Form

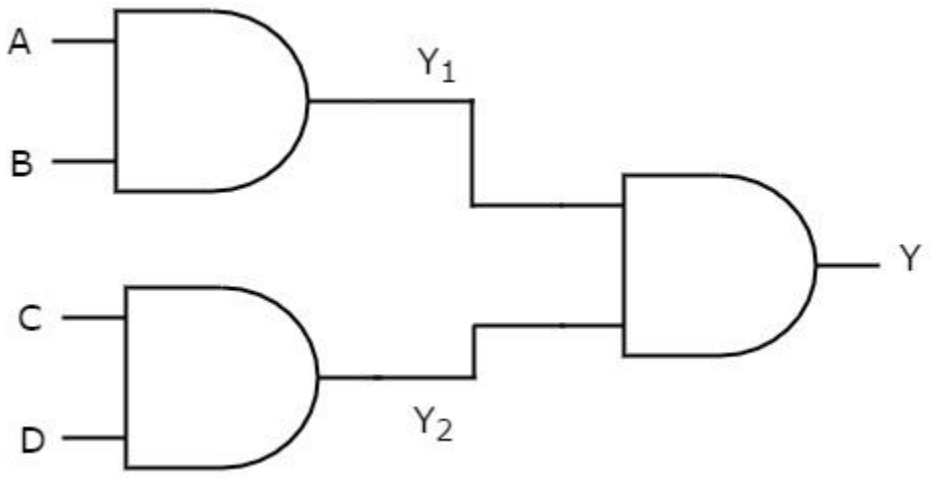
If the output of two level logic realization can be obtained by using single Logic gate, then it is called as **degenerative form**. Obviously, the number of inputs of single Logic gate increases. Due to this, the fan-in of Logic gate increases. This is an advantage of degenerative form.

Only **6 combinations** of two level logic realizations out of 16 combinations come under degenerative form. Those are AND-AND, AND-NAND, OR-OR, OR-NOR, NAND-NOR, NORNAND.

In this section, let us discuss some realizations. Assume, A, B, C & D are the inputs and Y is the output in each logic realization.

AND-AND Logic

In this logic realization, AND gates are present in both levels. Below figure shows an example for **AND-AND logic** realization.



We will get the outputs of first level logic gates as Y1=ABY1=AB and Y2=CDY2=CD

These outputs, Y1Y1 and Y2Y2 are applied as inputs of AND gate that is present in second level. So, the output of this AND gate is

$$Y=Y1Y2Y=Y1Y2$$

Substitute Y1Y1 and Y2Y2 values in the above equation.

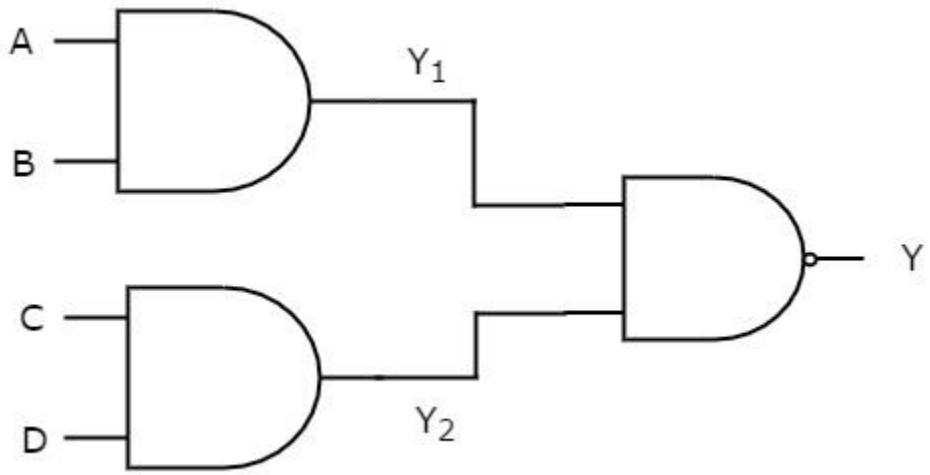
$$Y=(AB)(CD)Y=(AB)(CD)$$

$\Rightarrow Y=ABCD \Rightarrow Y=ABCD$

Therefore, the output of this AND-AND logic realization is **ABCD**. This Boolean function can be implemented by using a 4 input AND gate. Hence, it is **degenerative form**.

AND-NAND Logic

In this logic realization, AND gates are present in first level and NAND gate(s) are present in second level. The following figure shows an example for **AND-NAND logic** realization.



Previously, we got the outputs of first level logic gates as Y1=ABY1=AB and Y2=CDY2=CD

These outputs,Y1Y1 and Y2Y2 are applied as inputs of NAND gate that is present in second level. So, the output of this NAND gate is

$$Y=(Y1Y2)'Y=(Y1Y2)'$$

Substitute Y1Y1 and Y2Y2 values in the above equation.

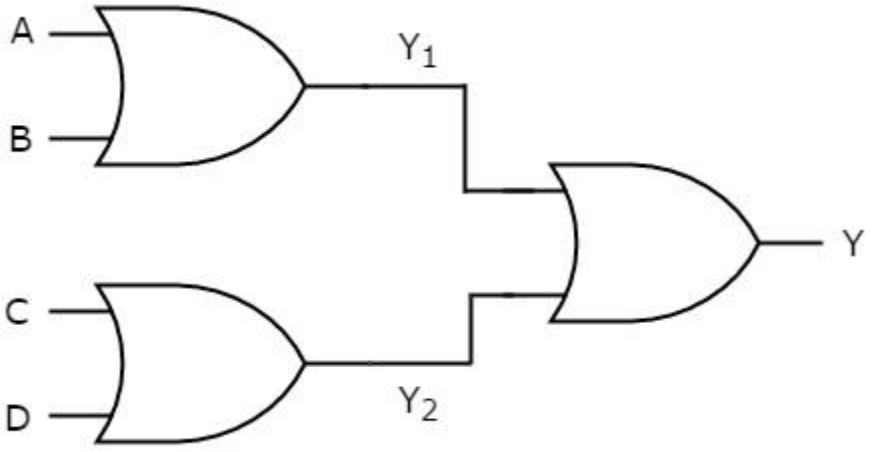
$$Y=((AB)(CD))'Y=((AB)(CD))'$$

$\Rightarrow Y=(ABCD)'\Rightarrow Y=(ABCD)'$

Therefore, the output of this AND-NAND logic realization is (ABCD)'(ABCD)'. This Boolean function can be implemented by using a 4 input NAND gate. Hence, it is **degenerative form**.

OR-OR Logic

In this logic realization, OR gates are present in both levels. The following figure shows an example for **OR-OR logic** realization.



We will get the outputs of first level logic gates as Y1=A+BY1=A+Band Y2=C+DY2=C+D.

These outputs, Y1Y1 and Y2Y2 are applied as inputs of OR gate that is present in second level. So, the output of this OR gate is

$$Y=Y1+Y2Y=Y1+Y2$$

Substitute Y1Y1 and Y2Y2 values in the above equation.

$$Y=(A+B)+(C+D)Y=(A+B)+(C+D)$$

$$\Rightarrow Y=A+B+C+D\Rightarrow Y=A+B+C+D$$

Therefore, the output of this OR-OR logic realization is **A+B+C+D**. This Boolean function can be implemented by using a 4 input OR gate. Hence, it is **degenerative form**.

Similarly, you can verify whether the remaining realizations belong to this category or not.

Non-degenerative Form

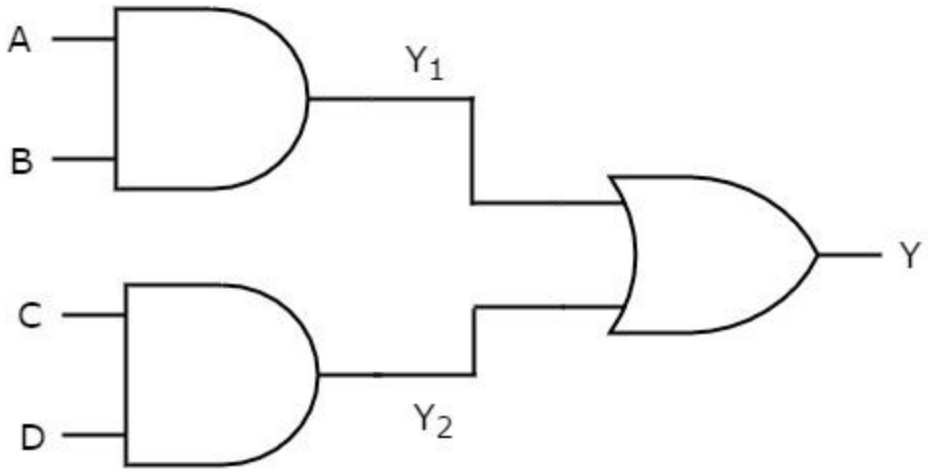
If the output of two level logic realization can't be obtained by using single logic gate, then it is called as **non-degenerative form**.

The remaining **10 combinations** of two level logic realizations come under nondegenerative form. Those are AND-OR, AND-NOR, OR-AND, OR-NAND, NAND-AND, NANDOR, NAND-NAND, NOR-AND, NOR-OR, NOR-NOR.

Now, let us discuss some realizations. Assume, A, B, C & D are the inputs and Y is the output in each logic realization.

AND-OR Logic

In this logic realization, AND gates are present in first level and OR gate(s) are present in second level. Below figure shows an example for **AND-OR logic** realization.



Previously, we got the outputs of first level logic gates as Y1=ABY1=AB and Y2=CDY2=CD.

These outputs, Y1 and Y2 are applied as inputs of OR gate that is present in second level. So, the output of this OR gate is

Y=Y1+Y2Y=Y1+Y2

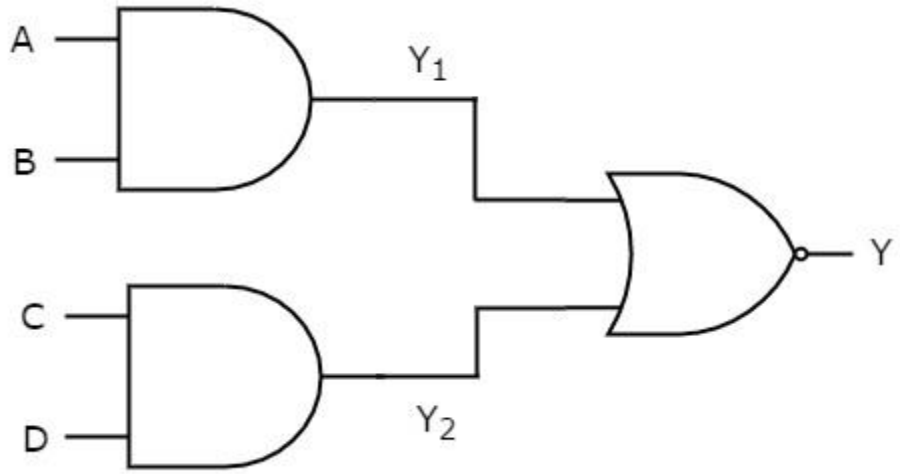
Substitute Y1Y1 and Y2Y2 values in the above equation

Y=AB+CDY=AB+CD

Therefore, the output of this AND-OR logic realization is **AB+CD**. This Boolean function is in **Sum of Products** form. Since, we can't implement it by using single logic gate, this AND-OR logic realization is a **non-degenerative form**.

AND-NOR Logic

In this logic realization, AND gates are present in first level and NOR gate(s) are present in second level. The following figure shows an example for **AND-NOR logic** realization.



We know the outputs of first level logic gates as Y1=ABY1=AB and Y2=CDY2=CD

These outputs, Y1 and Y2 are applied as inputs of NOR gate that is present in second level. So, the output of this NOR gate is

Y=(Y1+Y2)'Y=(Y1+Y2)'

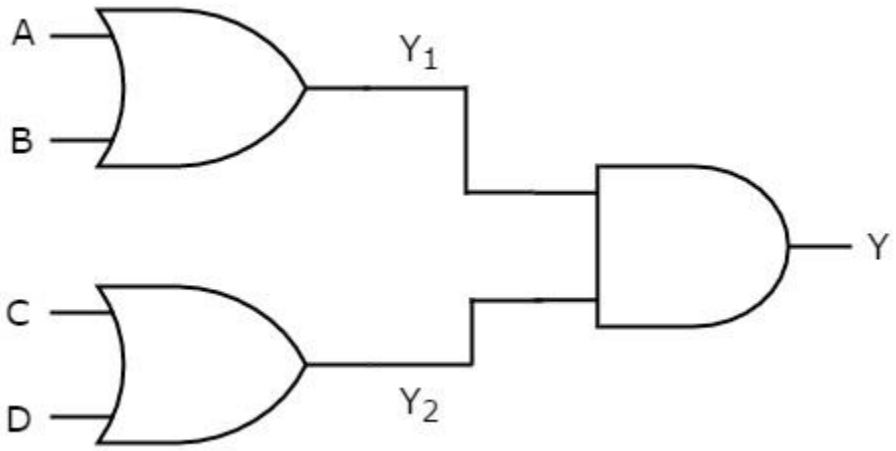
Substitute Y1Y1 and Y2Y2 values in the above equation.

Y=(AB+CD)'Y=(AB+CD)'

Therefore, the output of this AND-NOR logic realization is (AB+CD)'(AB+CD)'. This Boolean function is in **AND-OR-Invert** form. Since, we can't implement it by using single logic gate, this AND-NOR logic realization is a **non-degenerative form**

OR-AND Logic

In this logic realization, OR gates are present in first level & AND gate(s) are present in second level. The following figure shows an example for **OR-AND logic** realization.



Previously, we got the outputs of first level logic gates as Y1=A+BY1=A+B and Y2=C+DY2=C+D.
These outputs, Y1Y1 and Y2Y2 are applied as inputs of AND gate that is present in second level. So, the output of this AND gate is

$$Y=Y1Y2Y=Y1Y2$$

Substitute Y1Y1 and Y2Y2 values in the above equation.

$$Y=(A+B)(C+D)Y=(A+B)(C+D)$$

Therefore, the output of this OR-AND logic realization is **(A + B) (C + D)**. This Boolean function is in **Product of Sums** form. Since, we can't implement it by using single logic gate, this OR-AND logic realization is a **non-degenerative form**.

Similarly, you can verify whether the remaining realizations belong to this category or not.

UNIT-IV

Minimization Techniques

In previous chapters, we have simplified the Boolean functions using Boolean postulates and theorems. It is a time consuming process and we have to re-write the simplified expressions after each step.

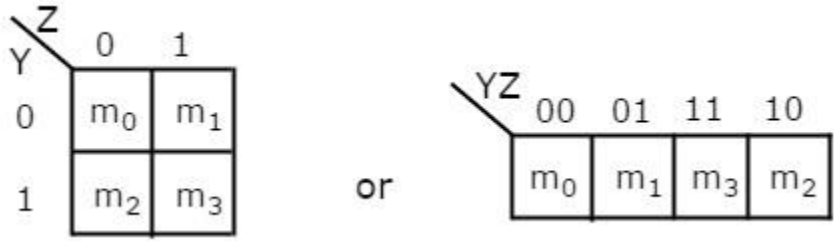
To overcome this difficulty, **Karnaugh** introduced a method for simplification of Boolean functions in an easy way. This method is known as Karnaugh map method or K-map method. It is a graphical method, which consists of 2^n cells for 'n' variables. The adjacent cells are differed only in single bit position.

K-Maps for 2 to 5 Variables

K-Map method is most suitable for minimizing Boolean functions of 2 variables to 5 variables. Now, let us discuss about the K-Maps for 2 to 5 variables one by one.

2 Variable K-Map

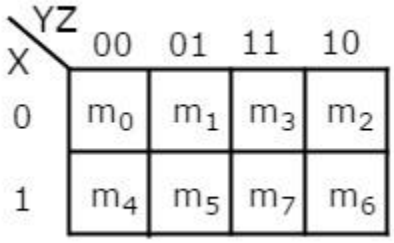
The number of cells in 2 variable K-map is four, since the number of variables is two. The following figure shows **2 variable K-Map**.



- There is only one possibility of grouping 4 adjacent min terms.
- The possible combinations of grouping 2 adjacent min terms are {(m_0, m_1), (m_2, m_3), (m_0, m_2) and (m_1, m_3)}.

3 Variable K-Map

The number of cells in 3 variable K-map is eight, since the number of variables is three. The following figure shows **3 variable K-Map**.



- There is only one possibility of grouping 8 adjacent min terms.
- The possible combinations of grouping 4 adjacent min terms are {(m_0, m_1, m_3, m_2), (m_4, m_5, m_7, m_6), (m_0, m_1, m_4, m_5), (m_1, m_3, m_5, m_7), (m_3, m_2, m_7, m_6) and (m_2, m_0, m_6, m_4)}.
- The possible combinations of grouping 2 adjacent min terms are {(m_0, m_1), (m_1, m_3), (m_3, m_2), (m_2, m_0), (m_4, m_5), (m_5, m_7), (m_7, m_6), (m_6, m_4), (m_0, m_4), (m_1, m_5), (m_3, m_7) and (m_2, m_6)}.
- If x=0, then 3 variable K-map becomes 2 variable K-map.

4 Variable K-Map

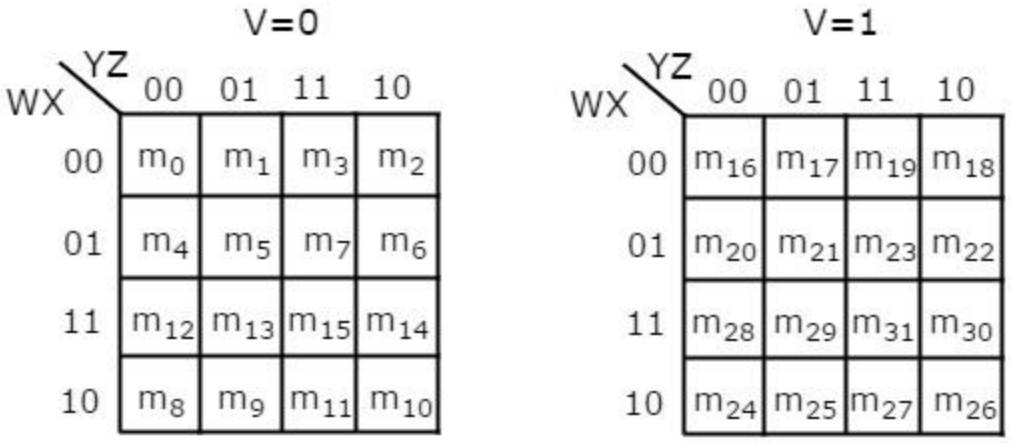
The number of cells in 4 variable K-map is sixteen, since the number of variables is four. The following figure shows **4 variable K-Map**.

WX \ YZ	00	01	11	10
00	m_0	m_1	m_3	m_2
01	m_4	m_5	m_7	m_6
11	m_{12}	m_{13}	m_{15}	m_{14}
10	m_8	m_9	m_{11}	m_{10}

- There is only one possibility of grouping 16 adjacent min terms.
- Let R_1, R_2, R_3 and R_4 represents the min terms of first row, second row, third row and fourth row respectively. Similarly, C_1, C_2, C_3 and C_4 represents the min terms of first column, second column, third column and fourth column respectively. The possible combinations of grouping 8 adjacent min terms are {(R_1, R_2), (R_2, R_3), (R_3, R_4), (R_4, R_1), (C_1, C_2), (C_2, C_3), (C_3, C_4), (C_4, C_1)}.
- If w=0, then 4 variable K-map becomes 3 variable K-map.

5 Variable K-Map

The number of cells in 5 variable K-map is thirty-two, since the number of variables is 5. The following figure shows **5 variable K-Map**.



- There is only one possibility of grouping 32 adjacent min terms.
- There are two possibilities of grouping 16 adjacent min terms. i.e., grouping of min terms from m_0 to m_{15} and m_{16} to m_{31}.
- If v=0, then 5 variable K-map becomes 4 variable K-map.

In the above all K-maps, we used exclusively the min terms notation. Similarly, you can use exclusively the Max terms notation.

Minimization of Boolean Functions using K-Maps

If we consider the combination of inputs for which the Boolean function is '1', then we will get the Boolean function, which is in **standard sum of products** form after simplifying the K-map.

Similarly, if we consider the combination of inputs for which the Boolean function is '0', then we will get the Boolean function, which is in **standard product of sums** form after simplifying the K-map.

Follow these **rules for simplifying K-maps** in order to get standard sum of products form.

- Select the respective K-map based on the number of variables present in the Boolean function.
- If the Boolean function is given as sum of min terms form, then place the ones at respective min term cells in the K-map. If the Boolean function is given as sum of products form, then place the ones in all possible cells of K-map for which the given product terms are valid.
- Check for the possibilities of grouping maximum number of adjacent ones. It should be powers of two. Start from highest power of two and upto least power of two. Highest power is equal to the number of variables considered in K-map and least power is zero.
- Each grouping will give either a literal or one product term. It is known as **prime implicant**. The prime implicant is said to be **essential prime implicant**, if atleast single '1' is not covered with any other groupings but only that grouping covers.

- Note down all the prime implicants and essential prime implicants. The simplified Boolean function contains all essential prime implicants and only the required prime implicants.

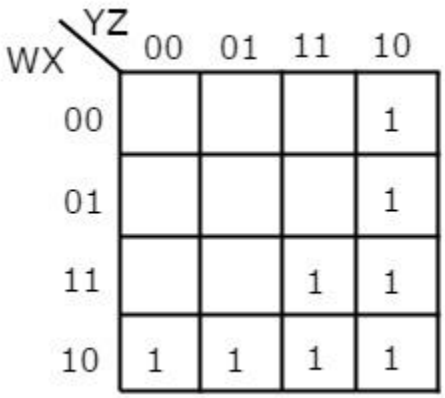
Note 1 − If outputs are not defined for some combination of inputs, then those output values will be represented with **don't care symbol 'x'**. That means, we can consider them as either '0' or '1'.

Note 2 − If don't care terms also present, then place don't cares 'x' in the respective cells of K-map. Consider only the don't cares 'x' that are helpful for grouping maximum number of adjacent ones. In those cases, treat the don't care value as '1'.

Example

Let us **simplify** the following Boolean function, **f(W, X, Y, Z)= WX'Y' + WY + W'YZ'** using K-map.

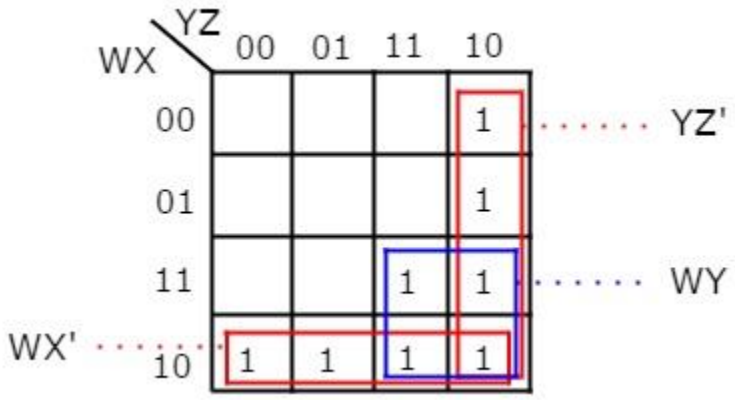
The given Boolean function is in sum of products form. It is having 4 variables W, X, Y & Z. So, we require **4 variable K-map**. The **4 variable K-map** with ones corresponding to the given product terms is shown in the following figure.



Here, 1s are placed in the following cells of K-map.

- The cells, which are common to the intersection of Row 4 and columns 1 & 2 are corresponding to the product term, **WX'Y'**.
- The cells, which are common to the intersection of Rows 3 & 4 and columns 3 & 4 are corresponding to the product term, **WY**.
- The cells, which are common to the intersection of Rows 1 & 2 and column 4 are corresponding to the product term, **W'YZ'**.

There are no possibilities of grouping either 16 adjacent ones or 8 adjacent ones. There are three possibilities of grouping 4 adjacent ones. After these three groupings, there is no single one left as ungrouped. So, we no need to check for grouping of 2 adjacent ones. The **4 variable K-map** with these three **groupings** is shown in the following figure.



Here, we got three prime implicants WX', WY & YZ'. All these prime implicants are **essential** because of following reasons.

- Two ones **(m_8 & m_9)** of fourth row grouping are not covered by any other groupings. Only fourth row grouping covers those two ones.
- Single one **(m_{15})** of square shape grouping is not covered by any other groupings. Only the square shape grouping covers that one.
- Two ones **(m_2 & m_6)** of fourth column grouping are not covered by any other groupings. Only fourth column grouping covers those two ones.

Therefore, the **simplified Boolean function** is

f = WX' + WY + YZ'

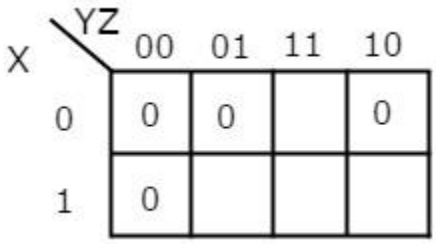
Follow these **rules for simplifying K-maps** in order to get standard product of sums form.

- Select the respective K-map based on the number of variables present in the Boolean function.
- If the Boolean function is given as product of Max terms form, then place the zeroes at respective Max term cells in the K-map. If the Boolean function is given as product of sums form, then place the zeroes in all possible cells of K-map for which the given sum terms are valid.
- Check for the possibilities of grouping maximum number of adjacent zeroes. It should be powers of two. Start from highest power of two and upto least power of two. Highest power is equal to the number of variables considered in K-map and least power is zero.
- Each grouping will give either a literal or one sum term. It is known as **prime implicant**. The prime implicant is said to be **essential prime implicant**, if atleast single '0' is not covered with any other groupings but only that grouping covers.
- Note down all the prime implicants and essential prime implicants. The simplified Boolean function contains all essential prime implicants and only the required prime implicants.

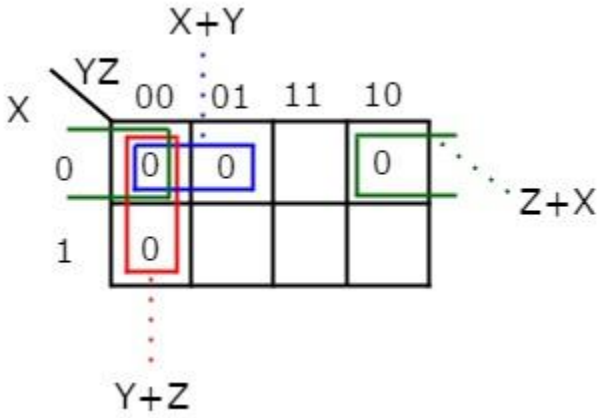
Note − If don't care terms also present, then place don't cares 'x' in the respective cells of K-map. Consider only the don't cares 'x' that are helpful for grouping maximum number of adjacent zeroes. In those cases, treat the don't care value as '0'.

Example

Let us **simplify** the following Boolean function, f(X,Y,Z)=∏M(0,1,2,4)f(X,Y,Z)=∏M(0,1,2,4) using K-map. The given Boolean function is in product of Max terms form. It is having 3 variables X, Y & Z. So, we require 3 variable K-map. The given Max terms are M_0, M_1, M_2 & M_4. The 3 **variable K-map** with zeroes corresponding to the given Max terms is shown in the following figure.



There are no possibilities of grouping either 8 adjacent zeroes or 4 adjacent zeroes. There are three possibilities of grouping 2 adjacent zeroes. After these three groupings, there is no single zero left as ungrouped. The **3 variable K-map** with these three **groupings** is shown in the following figure.



Here, we got three prime implicants X + Y, Y + Z & Z + X. All these prime implicants are **essential** because one zero in each grouping is not covered by any other groupings except with their individual groupings.

Therefore, the **simplified Boolean function** is

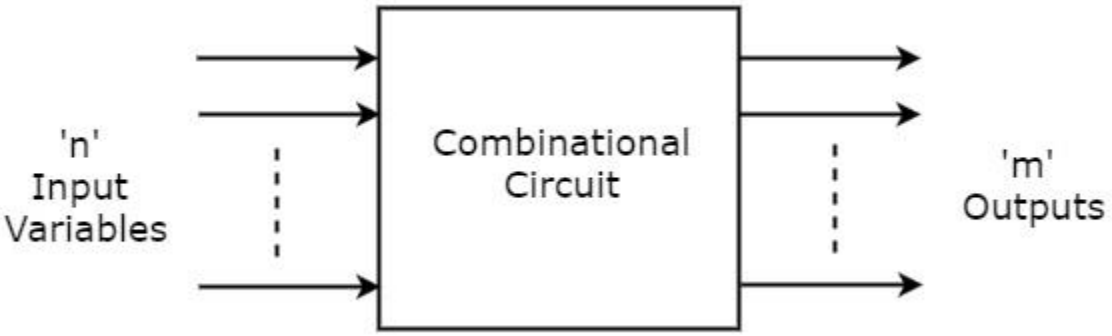
$$f = (X + Y).(Y + Z).(Z + X)$$

In this way, we can easily simplify the Boolean functions up to 5 variables using K-map method. For more than 5 variables, it is difficult to simplify the functions using K-Maps. Because, the number of **cells** in K-map gets **doubled** by including a new variable.

UNIT-V

Combinational Circuits & Sequential circuits

Combinational circuits consist of Logic gates. These circuits operate with binary values. The output(s) of combinational circuit depends on the combination of present inputs. The following figure shows the **block diagram** of combinational circuit.



This combinational circuit has 'n' input variables and 'm' outputs. Each combination of input variables will affect the output(s).

Design procedure of Combinational circuits

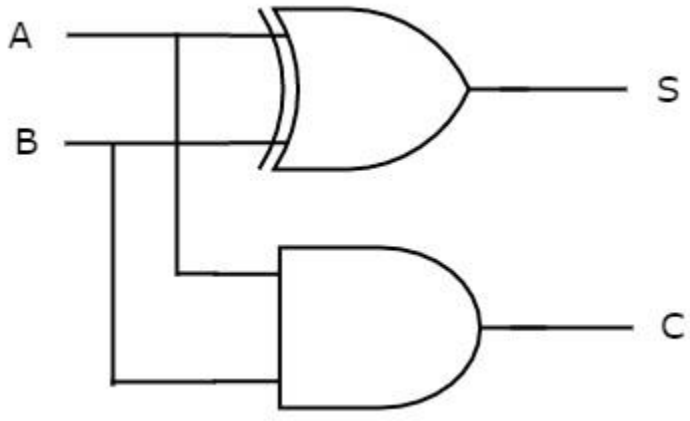
- Find the required number of input variables and outputs from given specifications.
- Formulate the **Truth table**. If there are 'n' input variables, then there will be 2n possible combinations. For each combination of input, find the output values.
- Find the **Boolean expressions** for each output. If necessary, simplify those expressions.
- Implement the above Boolean expressions corresponding to each output by using **Logic gates**.
- In this chapter, let us discuss about the basic arithmetic circuits like Binary adder and Binary subtractor. These circuits can be operated with binary values 0 and 1.
- Binary Adder
- The most basic arithmetic operation is addition. The circuit, which performs the addition of two binary numbers is known as **Binary adder**. First, let us implement an adder, which performs the addition of two bits.

Half Adder

- Half adder is a combinational circuit, which performs the addition of two binary numbers A and B are of **single bit**. It produces two outputs sum, S & carry, C.
- The **Truth table** of Half adder is shown below.

Inputs		Outputs	
A	B	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

- When we do the addition of two bits, the resultant sum can have the values ranging from 0 to 2 in decimal. We can represent the decimal digits 0 and 1 with single bit in binary. But, we can't represent decimal digit 2 with single bit in binary. So, we require two bits for representing it in binary.
- Let, sum, S is the Least significant bit and carry, C is the Most significant bit of the resultant sum. For first three combinations of inputs, carry, C is zero and the value of S will be either zero or one based on the **number of ones** present at the inputs. But, for last combination of inputs, carry, C is one and sum, S is zero, since the resultant sum is two.
- From Truth table, we can directly write the **Boolean functions**for each output as
    - S=A⨁BS=A⨁B
- C=ABC=AB
- We can implement the above functions with 2-input Ex-OR gate & 2-input AND gate. The **circuit diagram** of Half adder is shown in the following figure.



- In the above circuit, a two input Ex-OR gate & two input AND gate produces sum, S & carry, C respectively. Therefore, Half-adder performs the addition of two bits.

Full Adder

- Full adder is a combinational circuit, which performs the **addition of three bits** A, B and C_{in}. Where, A & B are the two parallel significant bits and C_{in} is the carry bit, which is generated from previous stage. This Full adder also produces two outputs sum, S & carry, C_{out}, which are similar to Half adder.
- The **Truth table** of Full adder is shown below.

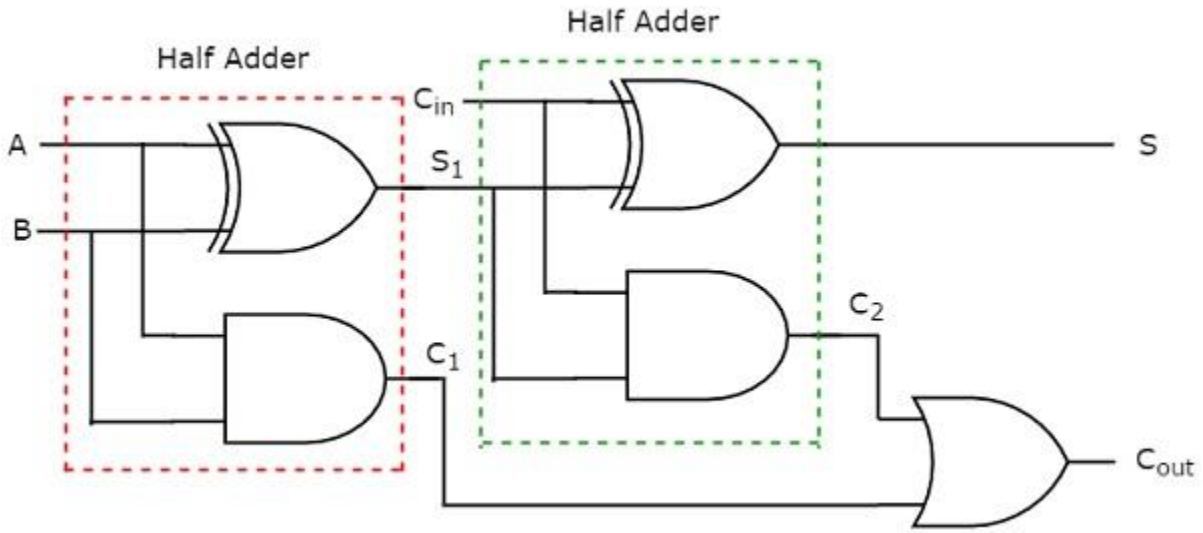
Inputs			**Outputs**	
A	**B**	**C_{in}**	**C_{out}**	**S**
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

- When we do the addition of three bits, the resultant sum can have the values ranging from 0 to 3 in decimal. We can represent the decimal digits 0 and 1 with single bit in binary. But, we can't represent the decimal digits 2 and 3 with single bit in binary. So, we require two bits for representing those two decimal digits in binary.
- Let, sum, S is the Least significant bit and carry, C_{out} is the Most significant bit of resultant sum. It is easy to fill the values of outputs for all combinations of inputs in the truth table. Just count

the **number of ones** present at the inputs and write the equivalent binary number at outputs. If C_{in} is equal to zero, then Full adder truth table is same as that of Half adder truth table.

- We will get the following **Boolean functions** for each output after simplification.
    - S=A⊕B⊕CinS=A⊕B⊕Cin

- cout=AB+(A⊕B)cincout=AB+(A⊕B)cin
- The sum, S is equal to one, when odd number of ones present at the inputs. We know that Ex-OR gate produces an output, which is an odd function. So, we can use either two 2input Ex-OR gates or one 3-input Ex-OR gate in order to produce sum, S. We can implement carry, C_{out} using two 2-input AND gates & one OR gate. The **circuit diagram** of Full adder is shown in the following figure.

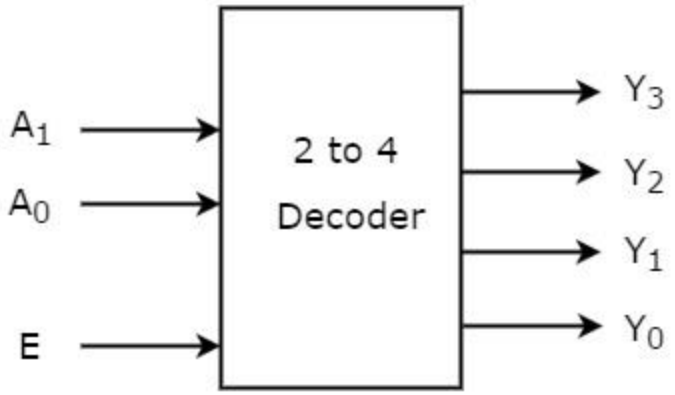


- This adder is called as **Full adder** because for implementing one Full adder, we require two Half adders and one OR gate. If C_{in} is zero, then Full adder becomes Half adder. We can verify it easily from the above circuit diagram or from the Boolean functions of outputs of Full adder.

Decoder is a combinational circuit that has 'n' input lines and maximum of 2^n output lines. One of these outputs will be active High based on the combination of inputs present, when the decoder is enabled. That means decoder detects a particular code. The outputs of the decoder are nothing but the **min terms**of 'n' input variables (lines), when it is enabled.

2 to 4 Decoder

Let 2 to 4 Decoder has two inputs A_1 & A_0 and four outputs Y_3, Y_2, Y_1 & Y_0. The **block diagram** of 2 to 4 decoder is shown in the following figure.



One of these four outputs will be '1' for each combination of inputs when enable, E is '1'. The **Truth table** of 2 to 4 decoder is shown below.

Enable	**Inputs**		**Outputs**			
E	**A_1**	**A_0**	**Y_3**	**Y_2**	**Y_1**	**Y_0**
0	x	x	0	0	0	0
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0

From Truth table, we can write the **Boolean functions** for each output as

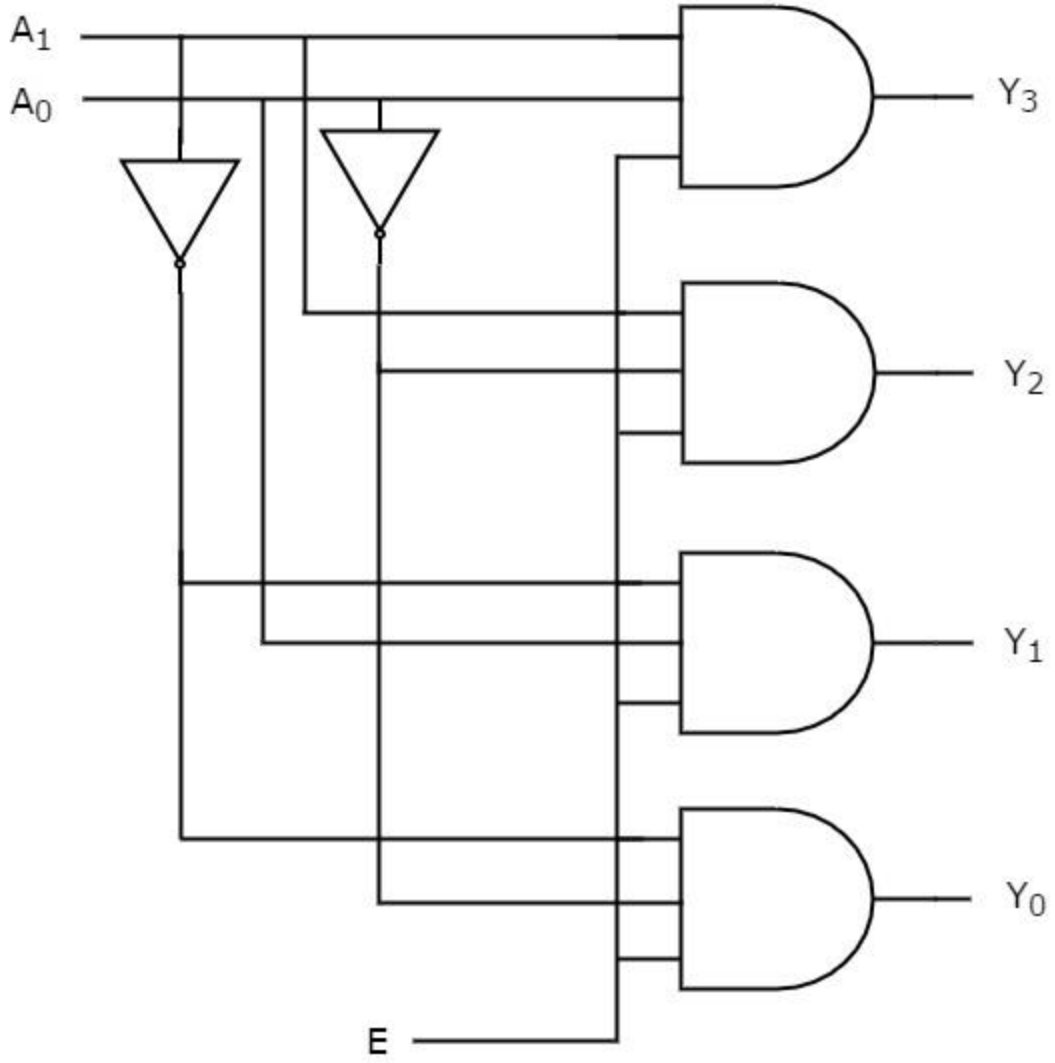
Y3=E.A1.A0Y3=E.A1.A0

Y2=E.A1.A0'Y2=E.A1.A0'

Y1=E.A1'.A0Y1=E.A1'.A0

Y0=E.A1'.A0'Y0=E.A1'.A0'

Each output is having one product term. So, there are four product terms in total. We can implement these four product terms by using four AND gates having three inputs each & two inverters. The **circuit diagram** of 2 to 4 decoder is shown in the following figure.



Therefore, the outputs of 2 to 4 decoder are nothing but the **min terms** of two input variables A_1 & A_0, when enable, E is equal to one. If enable, E is zero, then all the outputs of decoder will be equal to zero.

Similarly, 3 to 8 decoder produces eight min terms of three input variables A_2, A_1 & A_0 and 4 to 16 decoder produces sixteen min terms of four input variables A_3, A_2, A_1 & A_0.

Implementation of Higher-order Decoders

Now, let us implement the following two higher-order decoders using lower-order decoders.

- 3 to 8 decoder
- 4 to 16 decoder

3 to 8 Decoder

In this section, let us implement **3 to 8 decoder using 2 to 4 decoders**. We know that 2 to 4 Decoder has two inputs, A_1 & $A_0$and four outputs, Y_3 to Y_0. Whereas, 3 to 8 Decoder has three inputs A_2, A_1 & A_0 and eight outputs, Y_7 to Y_0.

We can find the number of lower order decoders required for implementing higher order decoder using the following formula.

Requirednumberoflowerorderdecoders=m2m1Requirednumberoflowerorderdecoders=m2m1

Where,

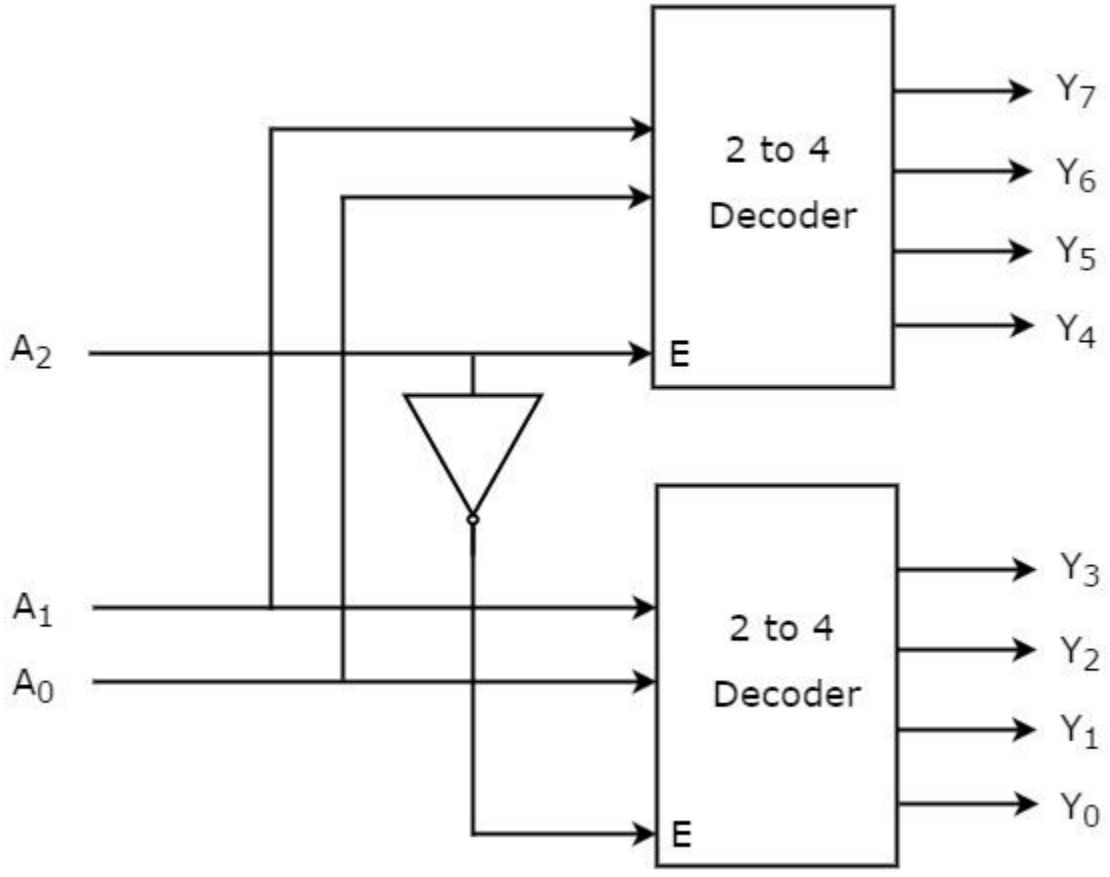
m1m1 is the number of outputs of lower order decoder.

m2m2 is the number of outputs of higher order decoder.

Here, m1m1 = 4 and m2m2 = 8. Substitute, these two values in the above formula.

Requirednumberof2to4decoders=84=2Requirednumberof2to4decoders=84=2

Therefore, we require two 2 to 4 decoders for implementing one 3 to 8 decoder. The **block diagram** of 3 to 8 decoder using 2 to 4 decoders is shown in the following figure.



The parallel inputs A_1 & A_0 are applied to each 2 to 4 decoder. The complement of input A_2 is connected to Enable, E of lower 2 to 4 decoder in order to get the outputs, Y_3 to Y_0. These are

the **lower four min terms**. The input, A_2 is directly connected to Enable, E of upper 2 to 4 decoder in order to get the outputs, $Y_7$to Y_4. These are the **higher four min terms**.

4 to 16 Decoder

In this section, let us implement **4 to 16 decoder using 3 to 8 decoders**. We know that 3 to 8 Decoder has three inputs A_2, A_1& A_0 and eight outputs, Y_7 to Y_0. Whereas, 4 to 16 Decoder has four inputs A_3, A_2, A_1 & A_0 and sixteen outputs, Y_{15} to Y_0

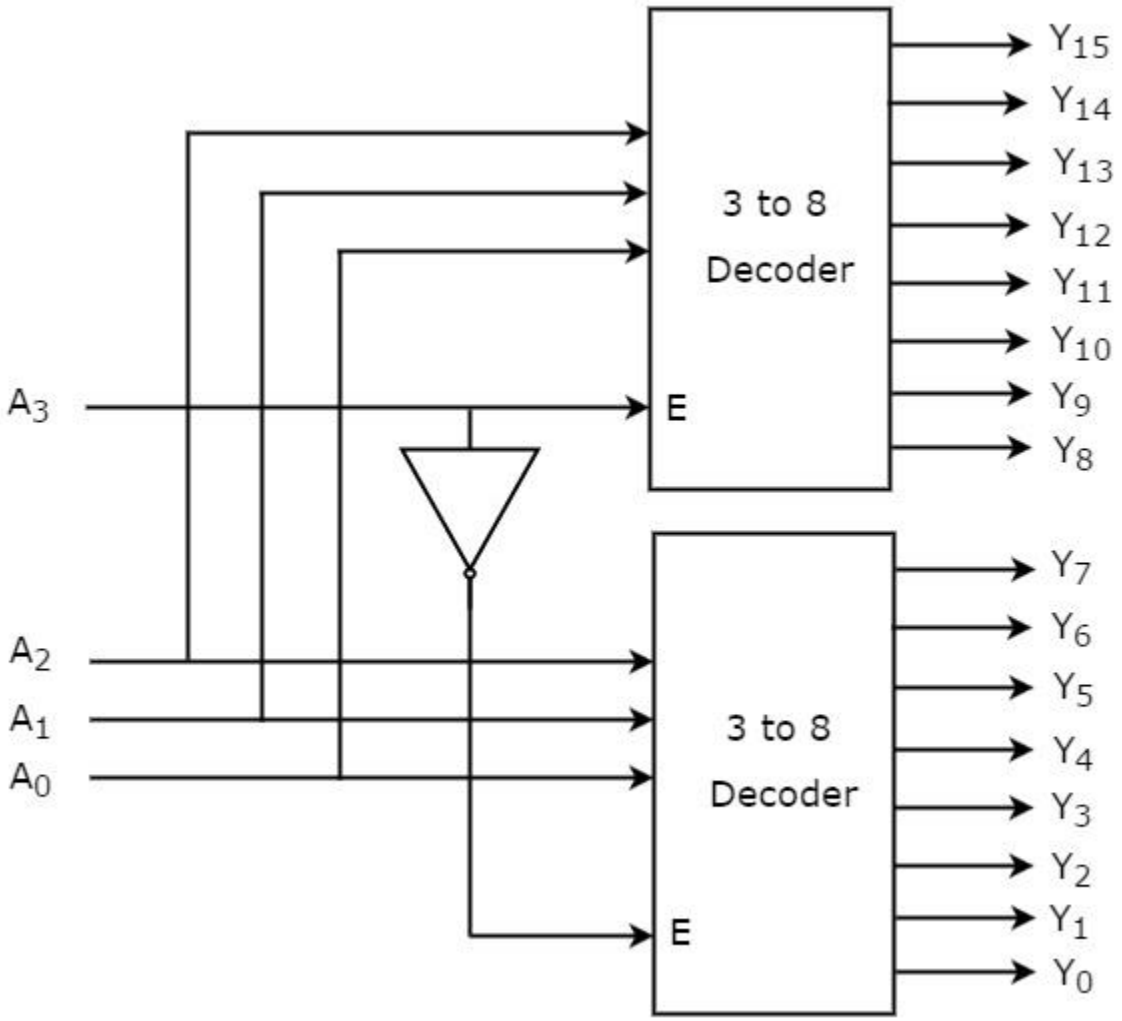
We know the following formula for finding the number of lower order decoders required.

Requirednumberoflowerorderdecoders=m2m1Requirednumberoflowerorderdecoders=m2m1

Substitute, m1m1 = 8 and m2m2 = 16 in the above formula.

Requirednumberof3to8decoders=168=2Requirednumberof3to8decoders=168=2

Therefore, we require two 3 to 8 decoders for implementing one 4 to 16 decoder. The **block diagram** of 4 to 16 decoder using 3 to 8 decoders is shown in the following figure.

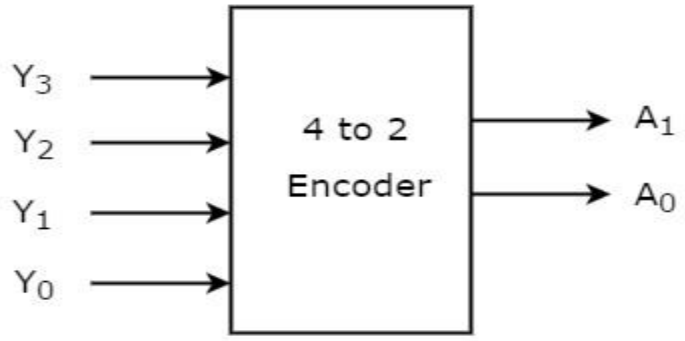


The parallel inputs A_2, A_1 & A_0 are applied to each 3 to 8 decoder. The complement of input, A3 is connected to Enable, E of lower 3 to 8 decoder in order to get the outputs, Y_7 to Y_0. These are the **lower eight min terms**. The input, A_3 is directly connected to Enable, E of upper 3 to 8 decoder in order to get the outputs, Y_{15} to Y_8. These are the **higher eight min terms**.

An **Encoder** is a combinational circuit that performs the reverse operation of Decoder. It has maximum of 2^n input lines and 'n' output lines. It will produce a binary code equivalent to the input, which is active High. Therefore, the encoder encodes 2^ninput lines with 'n' bits. It is optional to represent the enable signal in encoders.

4 to 2 Encoder

Let 4 to 2 Encoder has four inputs Y_3, Y_2, Y_1 & Y_0 and two outputs A_1 & A_0. The **block diagram** of 4 to 2 Encoder is shown in the following figure.



At any time, only one of these 4 inputs can be '1' in order to get the respective binary code at the output. The **Truth table** of 4 to 2 encoder is shown below.

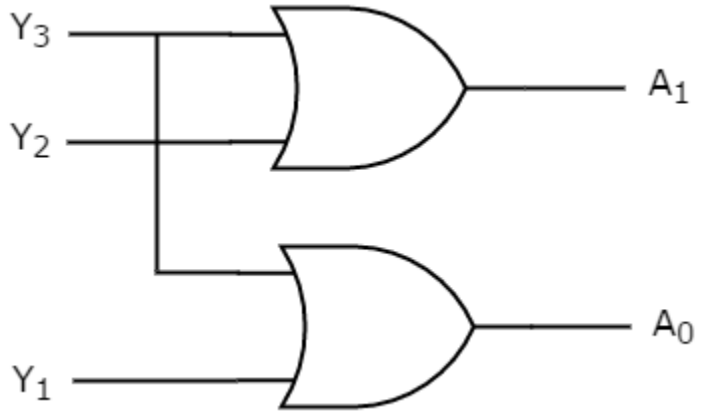
Inputs				**Outputs**	
Y_3	**Y_2**	**Y_1**	**Y_0**	**A_1**	**A_0**
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1

From Truth table, we can write the **Boolean functions** for each output as

$$A1=Y3+Y2A1=Y3+Y2$$

$$A0=Y3+Y1A0=Y3+Y1$$

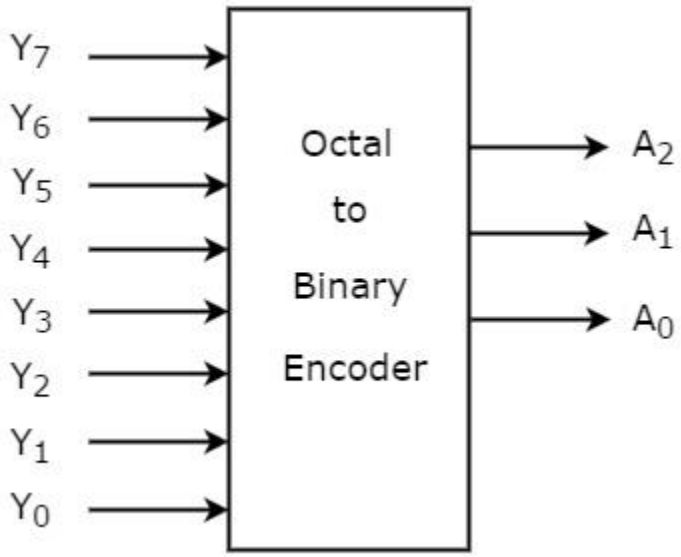
We can implement the above two Boolean functions by using two input OR gates. The **circuit diagram** of 4 to 2 encoder is shown in the following figure.



The above circuit diagram contains two OR gates. These OR gates encode the four inputs with two bits

Octal to Binary Encoder

Octal to binary Encoder has eight inputs, Y_7 to Y_0 and three outputs A_2, A_1 & A_0. Octal to binary encoder is nothing but 8 to 3 encoder. The **block diagram** of octal to binary Encoder is shown in the following figure.



At any time, only one of these eight inputs can be '1' in order to get the respective binary code. The **Truth table** of octal to binary encoder is shown below.

Inputs								Outputs		
Y_7	Y_6	Y_5	Y_4	Y_3	Y_2	Y_1	Y_0	A_2	A_1	A_0
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	1	0	1
0	1	0	0	0	0	0	0	1	1	0
1	0	0	0	0	0	0	0	1	1	1

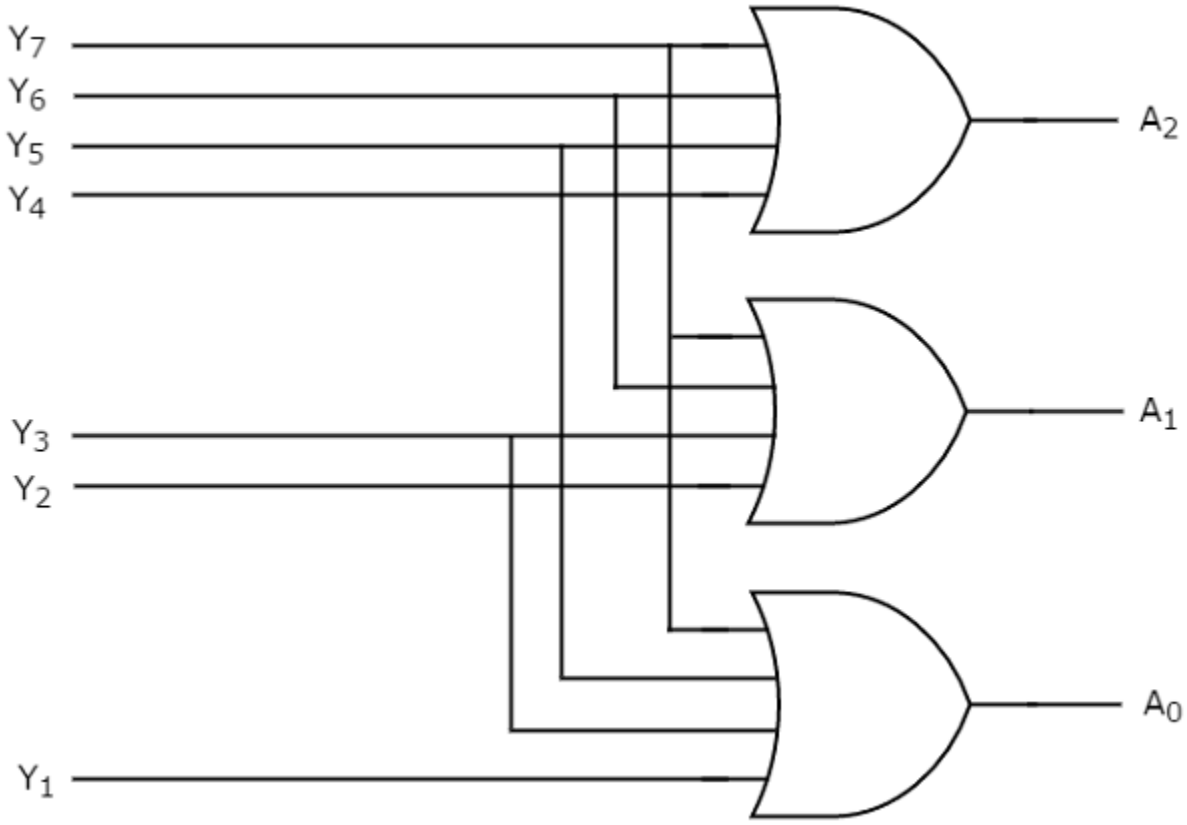
From Truth table, we can write the **Boolean functions** for each output as

$$A2=Y7+Y6+Y5+Y4A2=Y7+Y6+Y5+Y4$$

$$A1=Y7+Y6+Y3+Y2A1=Y7+Y6+Y3+Y2$$

$$A0=Y7+Y5+Y3+Y1A0=Y7+Y5+Y3+Y1$$

We can implement the above Boolean functions by using four input OR gates. The **circuit diagram** of octal to binary encoder is shown in the following figure.



The above circuit diagram contains three 4-input OR gates. These OR gates encode the eight inputs with three bits.

Drawbacks of Encoder

Following are the drawbacks of normal encoder.

- There is an ambiguity, when all outputs of encoder are equal to zero. Because, it could be the code corresponding to the inputs, when only least significant input is one or when all inputs are zero.
- If more than one input is active High, then the encoder produces an output, which may not be the correct code. For **example**, if both Y_3 and Y_6 are '1', then the encoder produces 111 at the output. This is neither equivalent code corresponding to Y_3, when it is '1' nor the equivalent code corresponding to Y_6, when it is '1'.

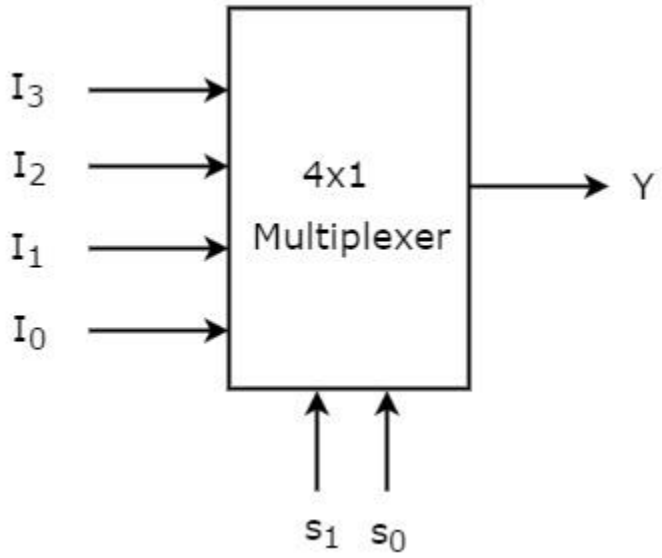
Multiplexer

is a combinational circuit that has maximum of 2^ndata inputs, 'n' selection lines and single output line. One of these data inputs will be connected to the output based on the values of selection lines.

Since there are 'n' selection lines, there will be 2^n possible combinations of zeros and ones. So, each combination will select only one data input. Multiplexer is also called as **Mux**.

4x1 Multiplexer

4x1 Multiplexer has four data inputs I_3, I_2, I_1 & I_0, two selection lines s_1 & s_0 and one output Y. The **block diagram** of 4x1 Multiplexer is shown in the following figure.



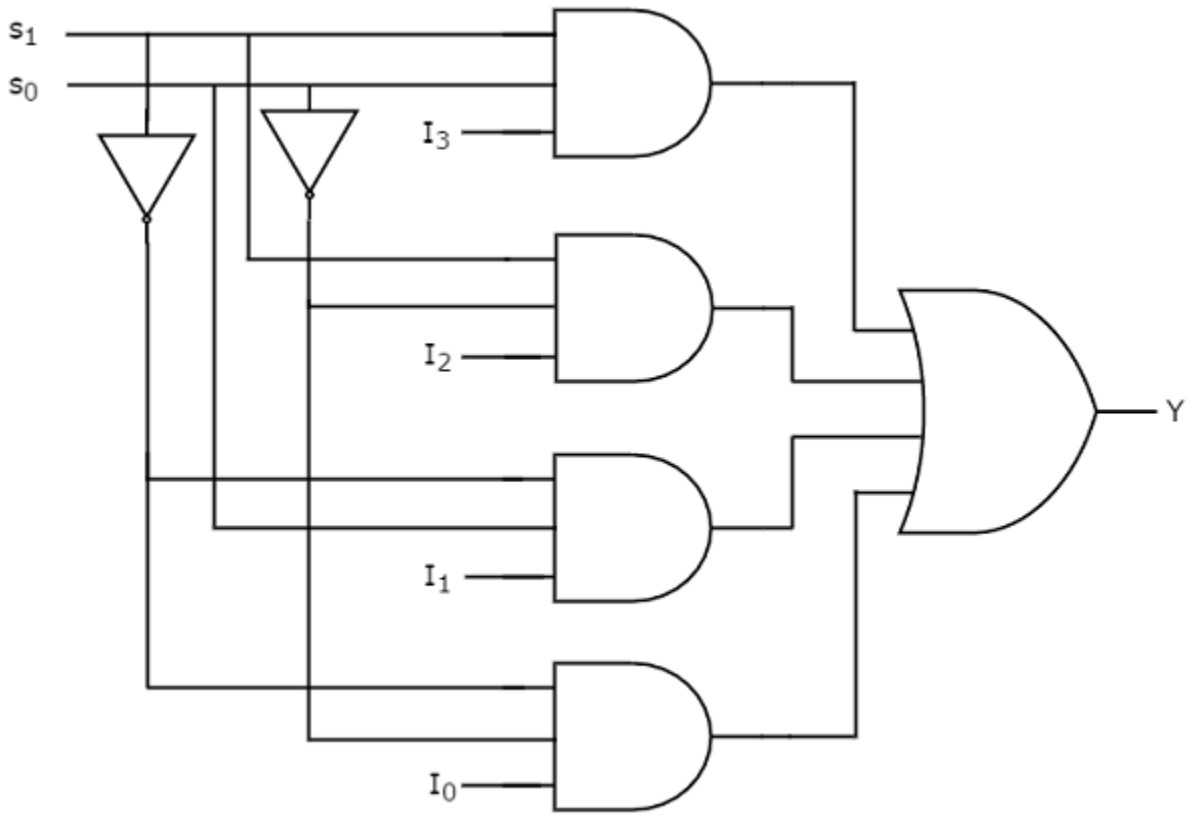
One of these 4 inputs will be connected to the output based on the combination of inputs present at these two selection lines. **Truth table** of 4x1 Multiplexer is shown below.

Selection Lines		Output
S_1	S_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

From Truth table, we can directly write the **Boolean function**for output, Y as

$$Y=S1'S0'I0+S1'S0I1+S1S0'I2+S1S0I3Y=S1'S0'I0+S1'S0I1+S1S0'I2+S1S0I3$$

We can implement this Boolean function using Inverters, AND gates & OR gate. The **circuit diagram** of 4x1 multiplexer is shown in the following figure.



We can easily understand the operation of the above circuit. Similarly, you can implement 8x1 Multiplexer and 16x1 multiplexer by following the same procedure.

Implementation of Higher-order Multiplexers.

Now, let us implement the following two higher-order Multiplexers using lower-order Multiplexers.

- 8x1 Multiplexer
- 16x1 Multiplexer

8x1 Multiplexer

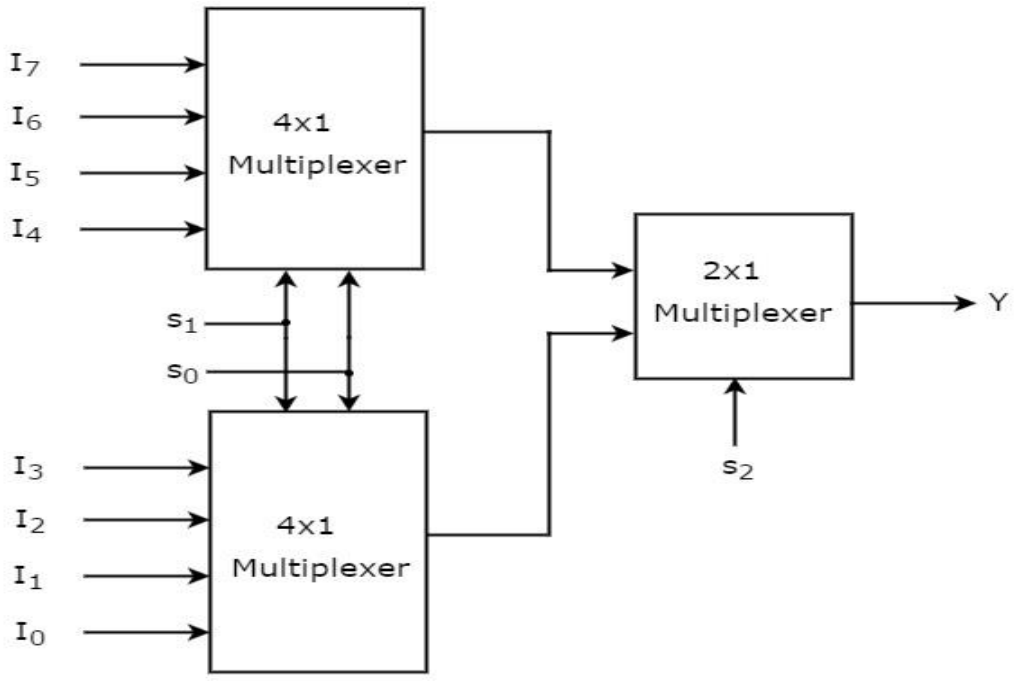
In this section, let us implement 8x1 Multiplexer using 4x1 Multiplexers and 2x1 Multiplexer. We know that 4x1 Multiplexer has 4 data inputs, 2 selection lines and one output. Whereas, 8x1 Multiplexer has 8 data inputs, 3 selection lines and one output.

So, we require two **4x1 Multiplexers** in first stage in order to get the 8 data inputs. Since, each 4x1 Multiplexer produces one output, we require a **2x1 Multiplexer** in second stage by considering the outputs of first stage as inputs and to produce the final output.

Let the 8x1 Multiplexer has eight data inputs I_7 to I_0, three selection lines s_2, s_1 & s0 and one output Y. The **Truth table** of 8x1 Multiplexer is shown below.

Selection Inputs			Output
S_2	S_1	S_0	Y
0	0	0	I_0
0	0	1	I_1
0	1	0	I_2
0	1	1	I_3
1	0	0	I_4
1	0	1	I_5
1	1	0	I_6
1	1	1	I_7

We can implement 8x1 Multiplexer using lower order Multiplexers easily by considering the above Truth table. The **block diagram**of 8x1 Multiplexer is shown in the following figure.



The same **selection lines, s_1 & s_0** are applied to both 4x1 Multiplexers. The data inputs of upper 4x1 Multiplexer are I_7 to I_4 and the data inputs of lower 4x1 Multiplexer are I_3 to I_0. Therefore, each 4x1 Multiplexer produces an output based on the values of selection lines, s_1 & s_0.

The outputs of first stage 4x1 Multiplexers are applied as inputs of 2x1 Multiplexer that is present in second stage. The other **selection line, s_2** is applied to 2x1 Multiplexer.

- If s_2 is zero, then the output of 2x1 Multiplexer will be one of the 4 inputs I_3 to I_0 based on the values of selection lines s_1 & s_0.
- If s_2 is one, then the output of 2x1 Multiplexer will be one of the 4 inputs I_7 to I_4 based on the values of selection lines s_1 & s_0.

Therefore, the overall combination of two 4x1 Multiplexers and one 2x1 Multiplexer performs as one 8x1 Multiplexer.

16x1 Multiplexer

In this section, let us implement 16x1 Multiplexer using 8x1 Multiplexers and 2x1 Multiplexer. We know that 8x1 Multiplexer has 8 data inputs, 3 selection lines and one output. Whereas, 16x1 Multiplexer has 16 data inputs, 4 selection lines and one output.

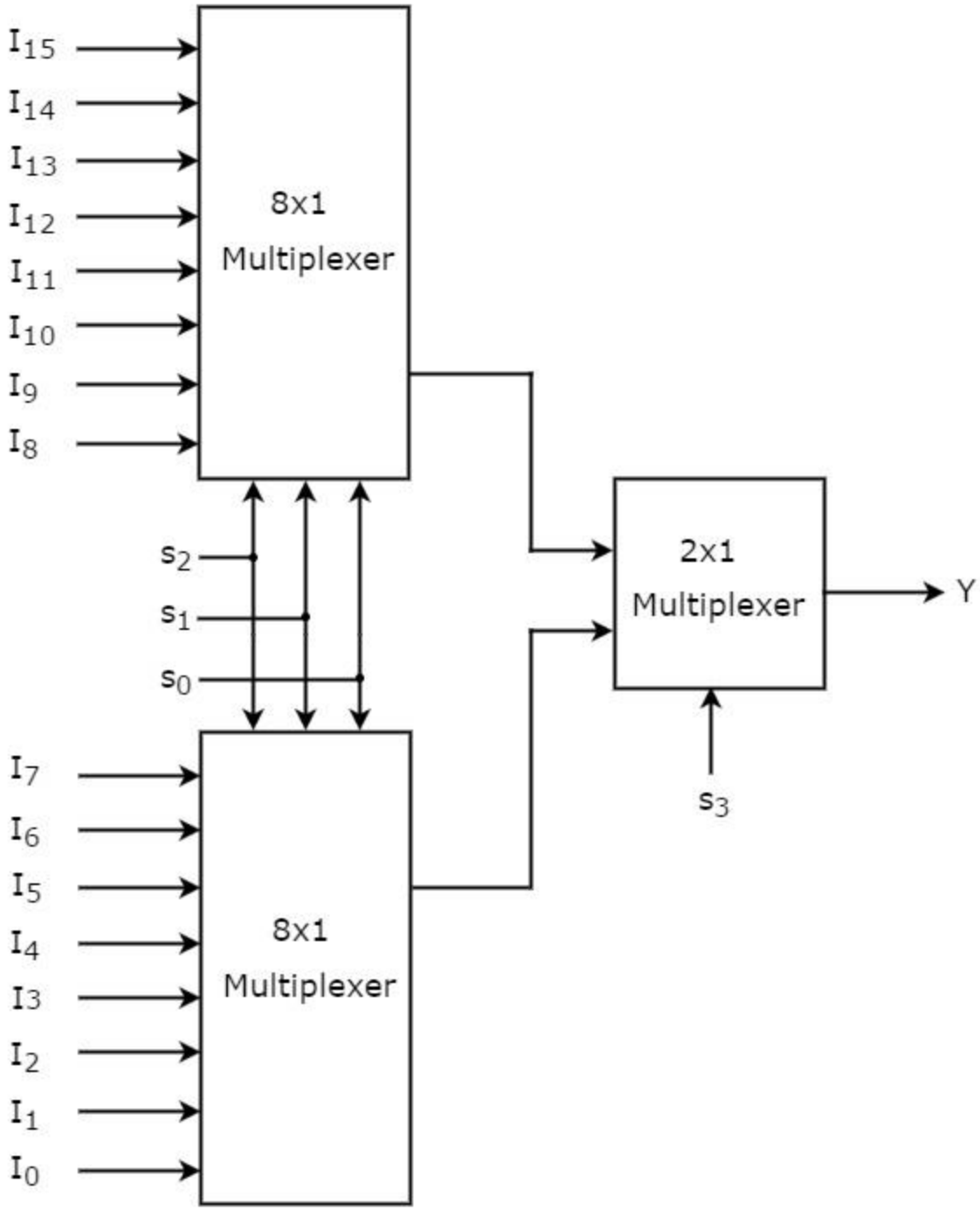
So, we require two **8x1 Multiplexers** in first stage in order to get the 16 data inputs. Since, each 8x1 Multiplexer produces one output, we require a 2x1 Multiplexer in second stage by considering the outputs of first stage as inputs and to produce the final output.

Let the 16x1 Multiplexer has sixteen data inputs I_{15} to I_0, four selection lines s_3 to s_0 and one output Y. The **Truth table** of 16x1 Multiplexer is shown below.

Selection Inputs				**Output**
S_3	**S_2**	**S_1**	**S_0**	**Y**
0	0	0	0	I_0
0	0	0	1	I_1
0	0	1	0	I_2
0	0	1	1	I_3

0	1	0	0	I_4
0	1	0	1	I_5
0	1	1	0	I_6
0	1	1	1	I_7
1	0	0	0	I_8
1	0	0	1	I_9
1	0	1	0	I_{10}
1	0	1	1	I_{11}
1	1	0	0	I_{12}
1	1	0	1	I_{13}
1	1	1	0	I_{14}
1	1	1	1	I_{15}

We can implement 16x1 Multiplexer using lower order Multiplexers easily by considering the above Truth table. The **block diagram** of 16x1 Multiplexer is shown in the following figure.



The **same selection lines, s_2, s_1 & s_0** are applied to both 8x1 Multiplexers. The data inputs of upper 8x1 Multiplexer are I_{15} to I_8 and the data inputs of lower 8x1 Multiplexer are I_7 to I_0. Therefore, each 8x1 Multiplexer produces an output based on the values of selection lines, s_2, s_1 & s_0.

The outputs of first stage 8x1 Multiplexers are applied as inputs of 2x1 Multiplexer that is present in second stage. The other **selection line, s_3** is applied to 2x1 Multiplexer.

- If s_3 is zero, then the output of 2x1 Multiplexer will be one of the 8 inputs Is_7 to I_0 based on the values of selection lines s_2, s_1 & s_0.

- If s_3 is one, then the output of 2x1 Multiplexer will be one of the 8 inputs I_{15} to I_8 based on the values of selection lines s_2, s_1 & s_0.

Therefore, the overall combination of two 8x1 Multiplexers and one 2x1 Multiplexer performs as one 16x1 Multiplexer.

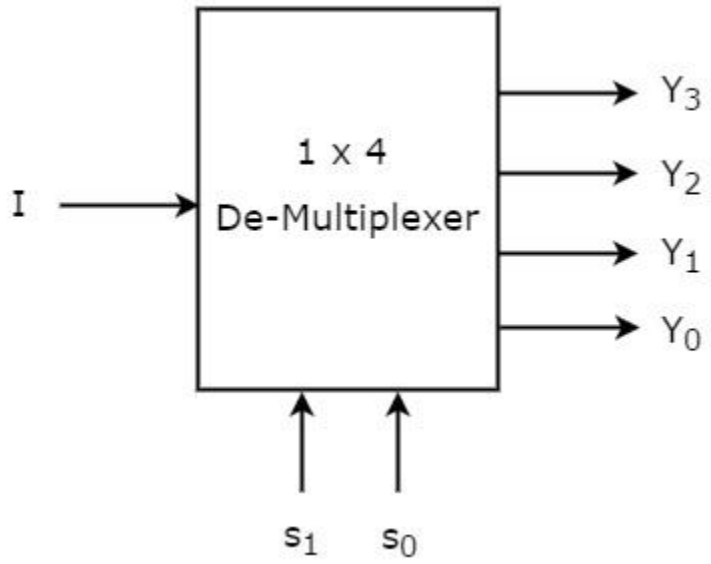
De-Multiplexer

is a combinational circuit that performs the reverse operation of Multiplexer. It has single input, 'n' selection lines and maximum of 2^n outputs. The input will be connected to one of these outputs based on the values of selection lines.

Since there are 'n' selection lines, there will be 2^n possible combinations of zeros and ones. So, each combination can select only one output. De-Multiplexer is also called as **De-Mux**.

1x4 De-Multiplexer

1x4 De-Multiplexer has one input I, two selection lines, s_1 & $s_0$and four outputs Y_3, Y_2, Y_1 &Y_0. The **block diagram** of 1x4 De-Multiplexer is shown in the following figure.



The single input 'I' will be connected to one of the four outputs, Y_3 to Y_0 based on the values of selection lines s_1 & s0. The **Truth table** of 1x4 De-Multiplexer is shown below.

Selection Inputs		Outputs			
S_1	S_0	Y_3	Y_2	Y_1	Y_0

0	0	0	0	0	I
0	1	0	0	I	0
1	0	0	I	0	0
1	1	I	0	0	0

From the above Truth table, we can directly write the **Boolean functions** for each output as

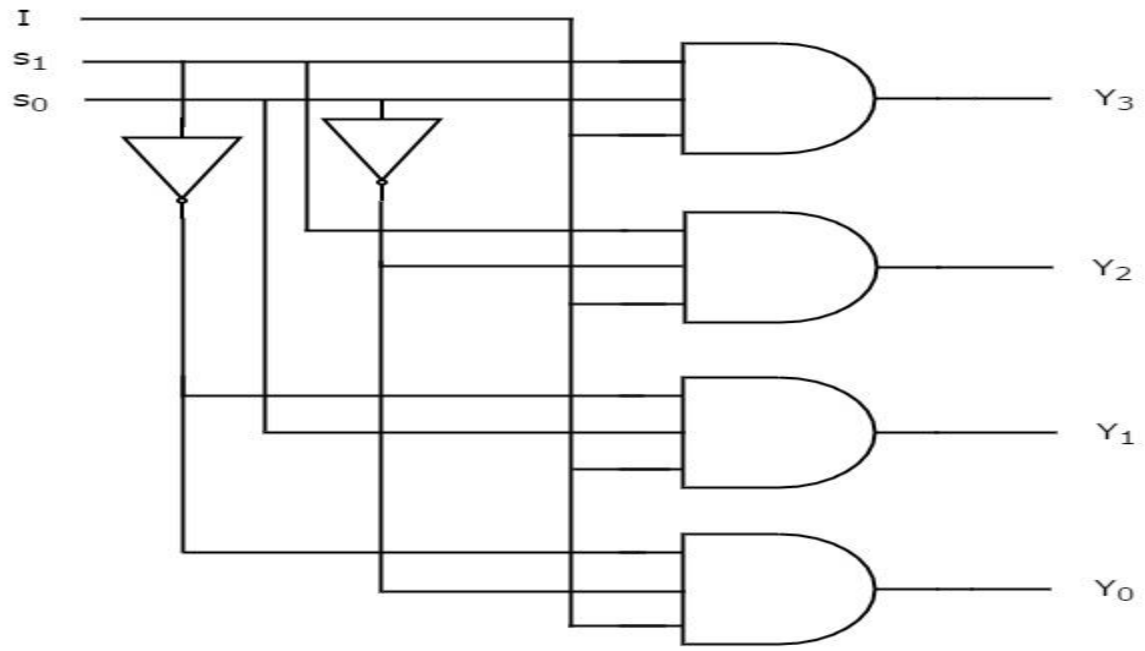
Y3=s1s0IY3=s1s0I

Y2=s1s0'IY2=s1s0'I

Y1=s1's0IY1=s1's0I

Y0=s1's0'IY0=s1's0'I

We can implement these Boolean functions using Inverters & 3-input AND gates. The **circuit diagram** of 1x4 De-Multiplexer is shown in the following figure.



We can easily understand the operation of the above circuit. Similarly, you can implement 1x8 De-Multiplexer and 1x16 De-Multiplexer by following the same procedure.

Implementation of Higher-order De-Multiplexers

Now, let us implement the following two higher-order De-Multiplexers using lower-order De-Multiplexers.

- 1x8 De-Multiplexer
- 1x16 De-Multiplexer

1x8 De-Multiplexer

In this section, let us implement 1x8 De-Multiplexer using 1x4 De-Multiplexers and 1x2 De-Multiplexer. We know that 1x4 De-Multiplexer has single input, two selection lines and four outputs. Whereas, 1x8 De-Multiplexer has single input, three selection lines and eight outputs.

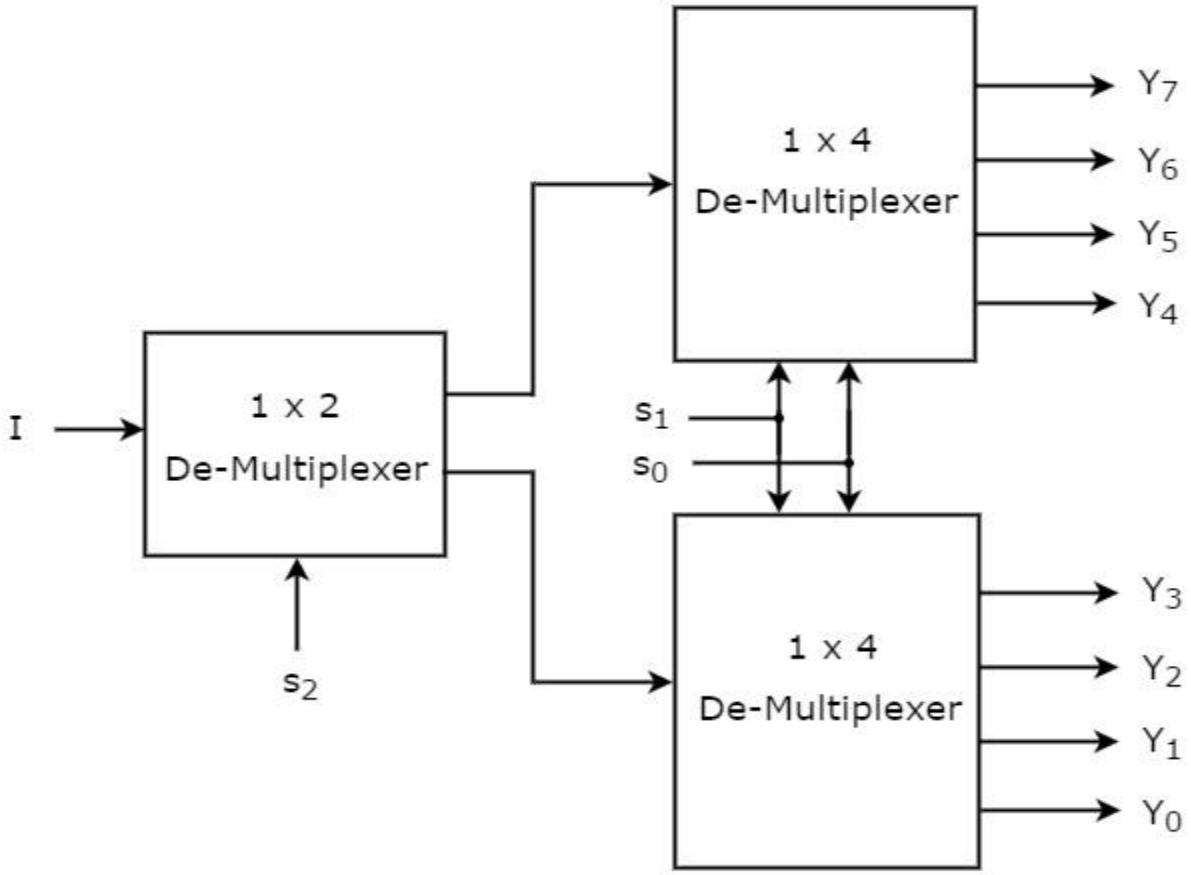
So, we require two **1x4 De-Multiplexers** in second stage in order to get the final eight outputs. Since, the number of inputs in second stage is two, we require **1x2 DeMultiplexer** in first stage so that the outputs of first stage will be the inputs of second stage. Input of this 1x2 De-Multiplexer will be the overall input of 1x8 De-Multiplexer.

Let the 1x8 De-Multiplexer has one input I, three selection lines s_2, s_1 & s_0 and outputs Y_7 to Y_0. The **Truth table** of 1x8 De-Multiplexer is shown below.

Selection Inputs			**Outputs**							
s_2	s_1	s_0	Y_7	Y_6	Y_5	Y_4	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	0	0	0	0	0	**I**
0	0	1	0	0	0	0	0	0	**I**	0
0	1	0	0	0	0	0	0	**I**	0	0
0	1	1	0	0	0	0	**I**	0	0	0
1	0	0	0	0	0	**I**	0	0	0	0
1	0	1	0	0	**I**	0	0	0	0	0
1	1	0	0	**I**	0	0	0	0	0	0

1	1	1	I	0	0	0	0	0	0	0

We can implement 1x8 De-Multiplexer using lower order Multiplexers easily by considering the above Truth table. The **block diagram** of 1x8 De-Multiplexer is shown in the following figure.



The common **selection lines, s_1 & s_0** are applied to both 1x4 De-Multiplexers. The outputs of upper 1x4 De-Multiplexer are $Y_7$to Y_4 and the outputs of lower 1x4 De-Multiplexer are Y_3 to Y_0.

The other **selection line, s_2** is applied to 1x2 De-Multiplexer. If s_2 is zero, then one of the four outputs of lower 1x4 De-Multiplexer will be equal to input, I based on the values of selection lines s_1 & s_0. Similarly, if s_2 is one, then one of the four outputs of upper 1x4 DeMultiplexer will be equal to input, I based on the values of selection lines s_1 & s_0.

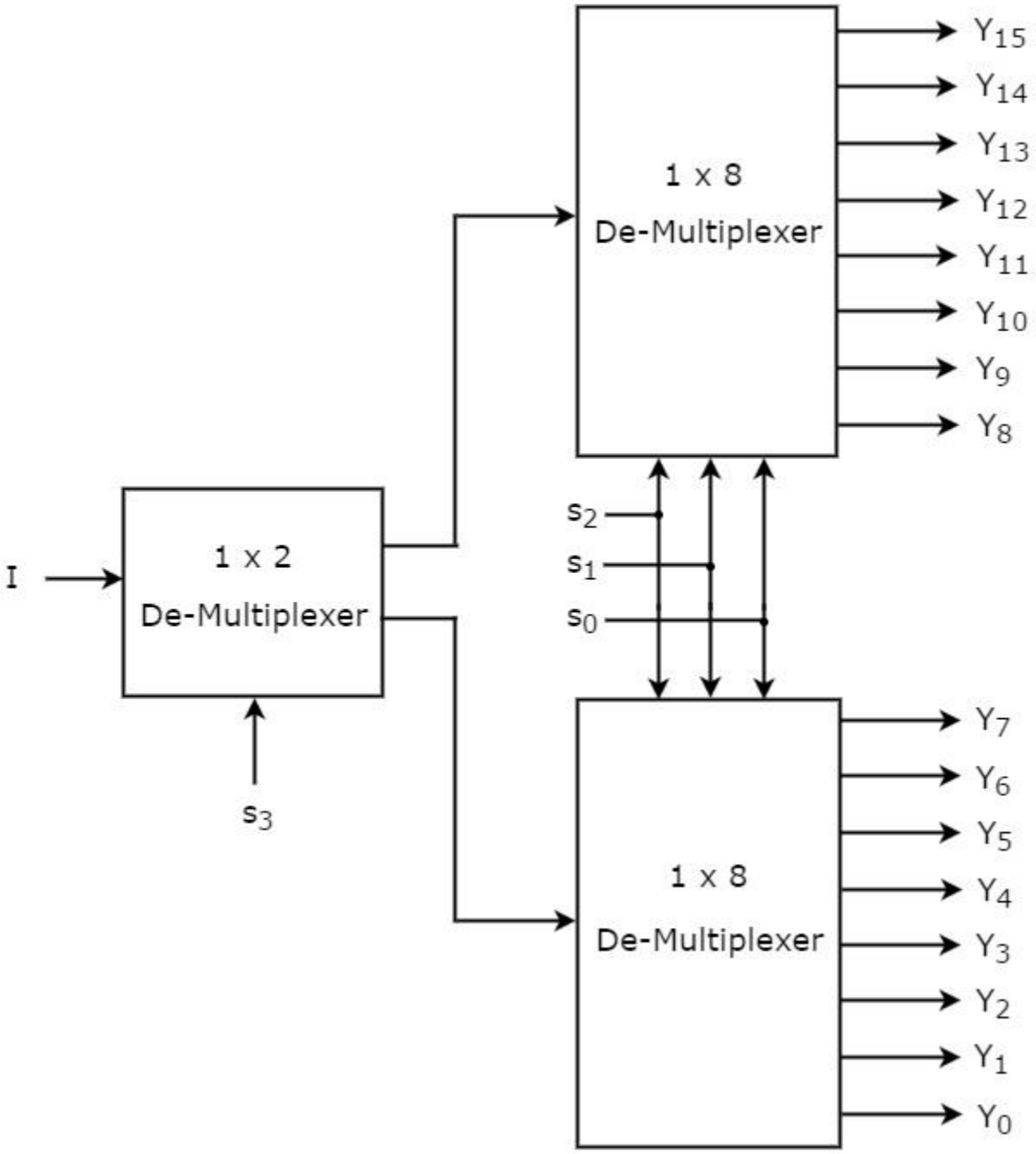
1x16 De-Multiplexer

In this section, let us implement 1x16 De-Multiplexer using 1x8 De-Multiplexers and 1x2 De-Multiplexer. We know that 1x8 De-Multiplexer has single input, three selection lines and eight outputs. Whereas, 1x16 De-Multiplexer has single input, four selection lines and sixteen outputs.

So, we require two **1x8 De-Multiplexers** in second stage in order to get the final sixteen outputs. Since, the number of inputs in second stage is two, we require **1x2 DeMultiplexer** in first stage so that

the outputs of first stage will be the inputs of second stage. Input of this 1x2 De-Multiplexer will be the overall input of 1x16 De-Multiplexer.

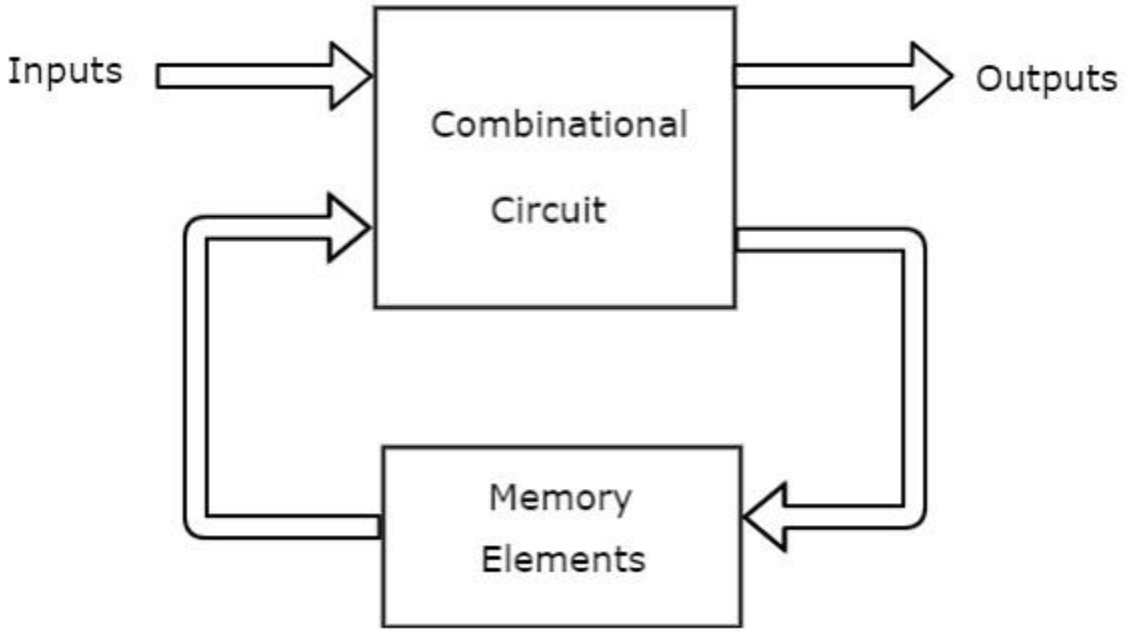
Let the 1x16 De-Multiplexer has one input I, four selection lines s_3, s_2, s_1 & s_0 and outputs Y_{15} to Y_0. The **block diagram** of 1x16 De-Multiplexer using lower order Multiplexers is shown in the following figure.



The common **selection lines s_2, s_1 & s_0** are applied to both 1x8 De-Multiplexers. The outputs of upper 1x8 De-Multiplexer are Y_{15} to Y_8 and the outputs of lower 1x8 DeMultiplexer are $Y_7$to Y_0.

The other **selection line, s_3** is applied to 1x2 De-Multiplexer. If s_3 is zero, then one of the eight outputs of lower 1x8 De-Multiplexer will be equal to input, I based on the values of selection lines s_2, s_1 & s_0. Similarly, if s3 is one, then one of the 8 outputs of upper 1x8 De-Multiplexer will be equal to input, I based on the values of selection lines s_2, s_1 & s_0.

We discussed various combinational circuits in earlier chapters. All these circuits have a set of output(s), which depends only on the combination of present inputs. The following figure shows the **block diagram** of sequential circuit.



This sequential circuit contains a set of inputs and output(s). The output(s) of sequential circuit depends not only on the combination of present inputs but also on the previous output(s). Previous output is nothing but the **present state**. Therefore, sequential circuits contain combinational circuits along with memory (storage) elements. Some sequential circuits may not contain combinational circuits, but only memory elements.

Following table shows the **differences** between combinational circuits and sequential circuits.

Combinational Circuits	Sequential Circuits
Outputs depend only on present inputs.	Outputs depend on both present inputs and present state.
Feedback path is not present.	Feedback path is present.

Memory elements are not required.	Memory elements are required.
Clock signal is not required.	Clock signal is required.
Easy to design.	Difficult to design.

Types of Sequential Circuits

Following are the two types of sequential circuits −

- Asynchronous sequential circuits
- Synchronous sequential circuits

Asynchronous sequential circuits

If some or all the outputs of a sequential circuit do not change (affect) with respect to active transition of clock signal, then that sequential circuit is called as **Asynchronous sequential circuit**. That means, all the outputs of asynchronous sequential circuits do not change (affect) at the same time. Therefore, most of the outputs of asynchronous sequential circuits are **not in synchronous** with either only positive edges or only negative edges of clock signal.

Synchronous sequential circuits

If all the outputs of a sequential circuit change (affect) with respect to active transition of clock signal, then that sequential circuit is called as **Synchronous sequential circuit**. That means, all the outputs of synchronous sequential circuits change (affect) at the same time. Therefore, the outputs of synchronous sequential circuits are in synchronous with either only positive edges or only negative edges of clock signal.

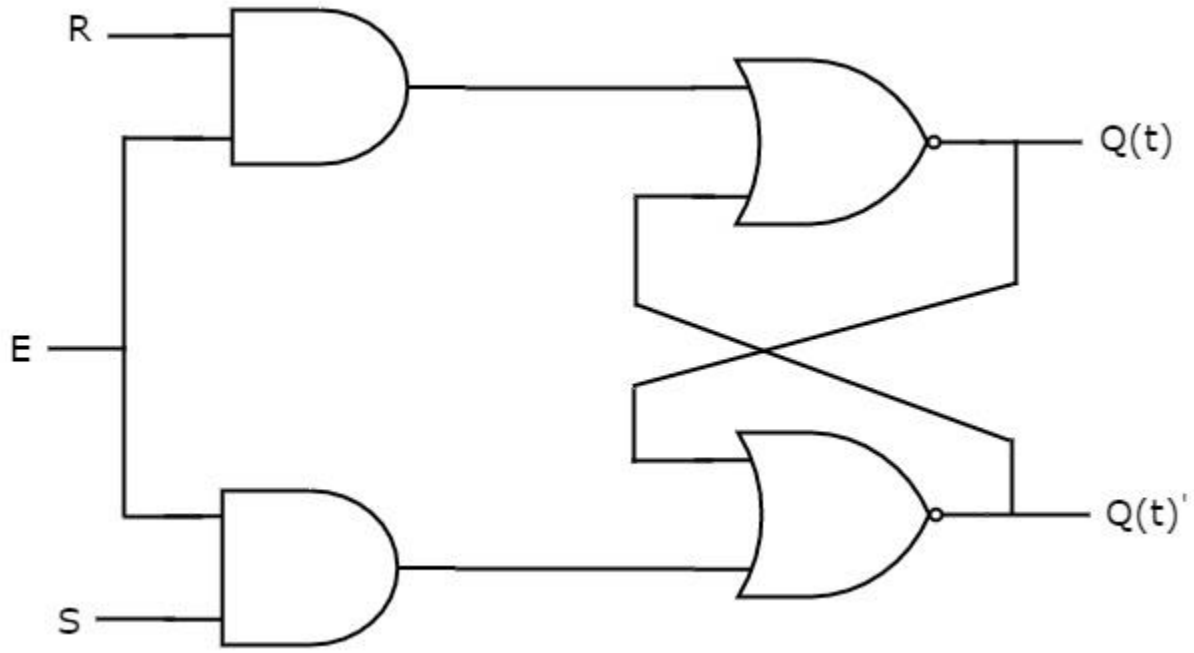
There are two types of memory elements based on the type of triggering that is suitable to operate it.

- Latches
- Flip-flops

Latches operate with enable signal, which is **level sensitive**. Whereas, flip-flops are edge sensitive. We will discuss about flip-flops in next chapter. Now, let us discuss about SR Latch & D Latch one by one.

SR Latch

SR Latch is also called as **Set Reset Latch**. This latch affects the outputs as long as the enable, E is maintained at '1'. The **circuit diagram** of SR Latch is shown in the following figure.



This circuit has two inputs S & R and two outputs Q(t) & Q(t)'. The **upper NOR gate** has two inputs R & complement of present state, Q(t)' and produces next state, Q(t+1) when enable, E is '1'.

Similarly, the **lower NOR gate** has two inputs S & present state, Q(t) and produces complement of next state, Q(t+1)' when enable, E is '1'.

We know that a **2-input NOR gate** produces an output, which is the complement of another input when one of the input is '0'. Similarly, it produces '0' output, when one of the input is '1'.

- If S = 1, then next state Q(t + 1) will be equal to '1' irrespective of present state, Q(t) values.
- If R = 1, then next state Q(t + 1) will be equal to '0' irrespective of present state, Q(t) values.

At any time, only of those two inputs should be '1'. If both inputs are '1', then the next state Q(t + 1) value is undefined.

The following table shows the **state table** of SR latch.

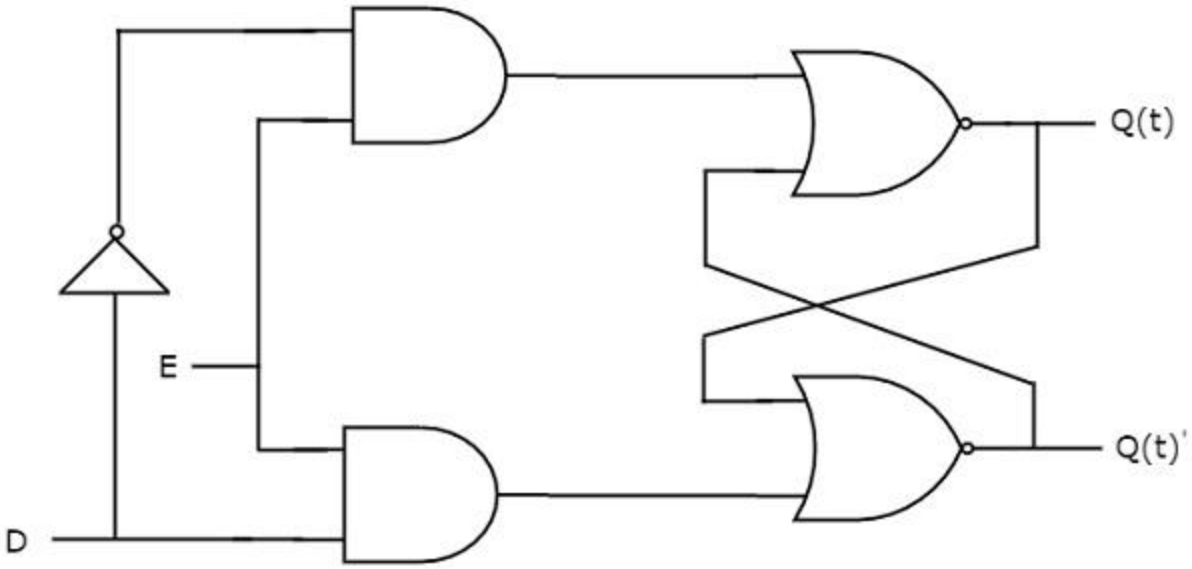
S	R	Q(t + 1)
0	0	Q(t)
0	1	0
1	0	1

1	1	-

Therefore, SR Latch performs three types of functions such as Hold, Set & Reset based on the input conditions.

D Latch

There is one drawback of SR Latch. That is the next state value can't be predicted when both the inputs S & R are one. So, we can overcome this difficulty by D Latch. It is also called as Data Latch. The **circuit diagram** of D Latch is shown in the following figure.



This circuit has single input D and two outputs Q(t) & Q(t)'. D Latch is obtained from SR Latch by placing an inverter between S amp;& R inputs and connect D input to S. That means we eliminated the combinations of S & R are of same value.

- If D = 0 → S = 0 & R = 1, then next state Q(t + 1) will be equal to '0' irrespective of present state, Q(t) values. This is corresponding to the second row of SR Latch state table.
- If D = 1 → S = 1 & R = 0, then next state Q(t + 1) will be equal to '1' irrespective of present state, Q(t) values. This is corresponding to the third row of SR Latch state table.

The following table shows the **state table** of D latch.

D	Q(t + 1)
0	0
1	1

Therefore, D Latch Hold the information that is available on data input, D. That means the output of D Latch is sensitive to the changes in the input, D as long as the enable is High.

In this chapter, we implemented various Latches by providing the cross coupling between NOR gates. Similarly, you can implement these Latches using NAND gates.

In previous chapter, we discussed about Latches. Those are the basic building blocks of flip-flops. We can implement flip-flops in two methods.

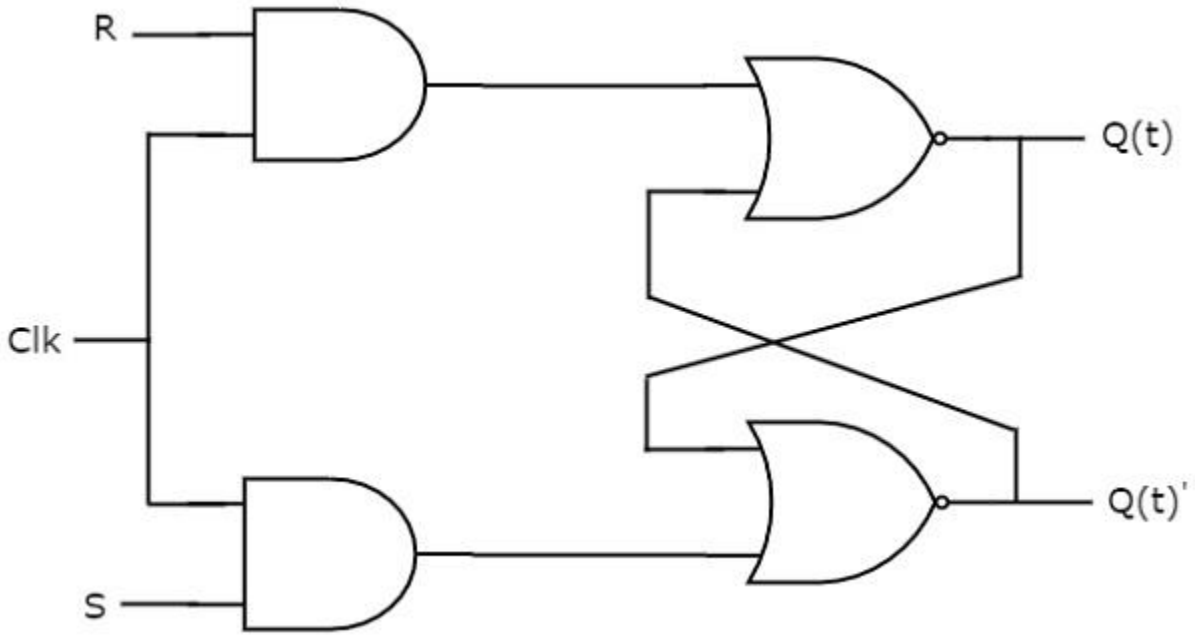
In first method, **cascade two latches** in such a way that the first latch is enabled for every positive clock pulse and second latch is enabled for every negative clock pulse. So that the combination of these two latches become a flip-flop.

In second method, we can directly implement the flip-flop, which is edge sensitive. In this chapter, let us discuss the following **flip-flops** using second method.

- SR Flip-Flop
- D Flip-Flop
- JK Flip-Flop
- T Flip-Flop

SR Flip-Flop

SR flip-flop operates with only positive clock transitions or negative clock transitions. Whereas, SR latch operates with enable signal. The **circuit diagram** of SR flip-flop is shown in the following figure.



This circuit has two inputs S & R and two outputs Q(t) & Q(t)'. The operation of SR flipflop is similar to SR Latch. But, this flip-flop affects the outputs only when positive transition of the clock signal is applied instead of active enable.

The following table shows the **state table** of SR flip-flop.

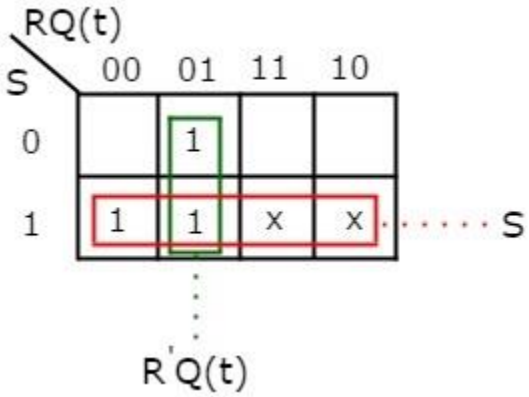
S	R	Q(t + 1)
0	0	Q(t)
0	1	0
1	0	1
1	1	-

Here, Q(t) & Q(t + 1) are present state & next state respectively. So, SR flip-flop can be used for one of these three functions such as Hold, Reset & Set based on the input conditions, when positive transition of clock signal is applied. The following table shows the **characteristic table** of SR flip-flop.

Present Inputs		Present State	Next State
S	R	Q(t)	Q(t + 1)
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1

1	1	0	x
1	1	1	x

By using three variable K-Map, we can get the simplified expression for next state, Q(t + 1). The **three variable K-Map**for next state, Q(t + 1) is shown in the following figure.

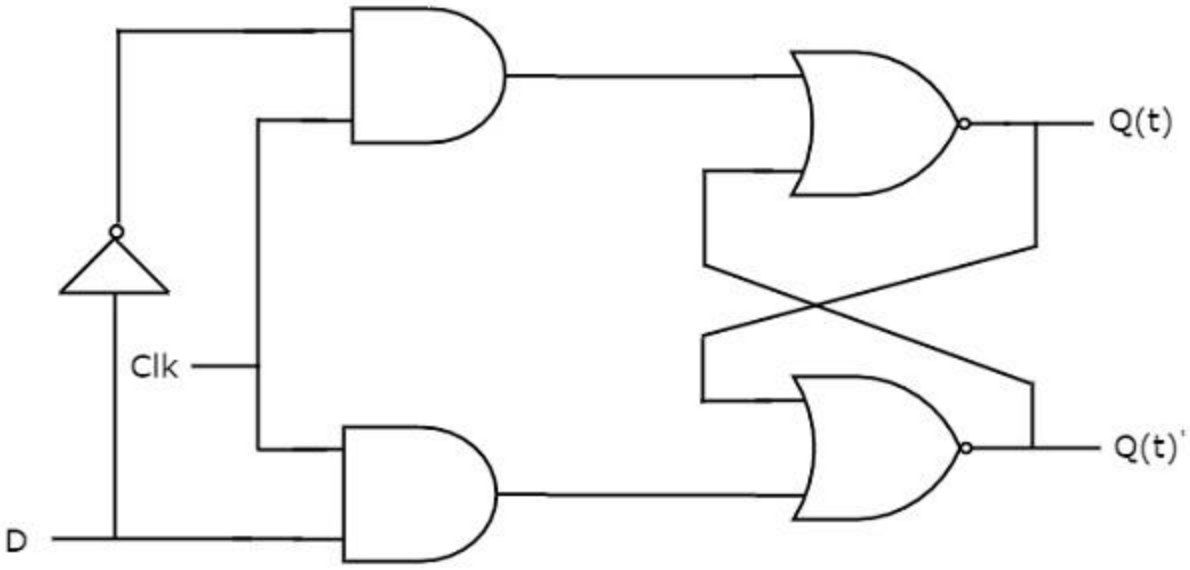


The maximum possible groupings of adjacent ones are already shown in the figure. Therefore, the **simplified expression** for next state Q(t + 1) is

$$Q(t+1)=S+R'Q(t)Q(t+1)=S+R'Q(t)$$

D Flip-Flop

D flip-flop operates with only positive clock transitions or negative clock transitions. Whereas, D latch operates with enable signal. That means, the output of D flip-flop is insensitive to the changes in the input, D except for active transition of the clock signal. The **circuit diagram** of D flip-flop is shown in the following figure.



This circuit has single input D and two outputs Q(t) & Q(t)'. The operation of D flip-flop is similar to D Latch. But, this flip-flop affects the outputs only when positive transition of the clock signal is applied instead of active enable.

The following table shows the **state table** of D flip-flop.

D	Q(t + 1)
0	0
0	1

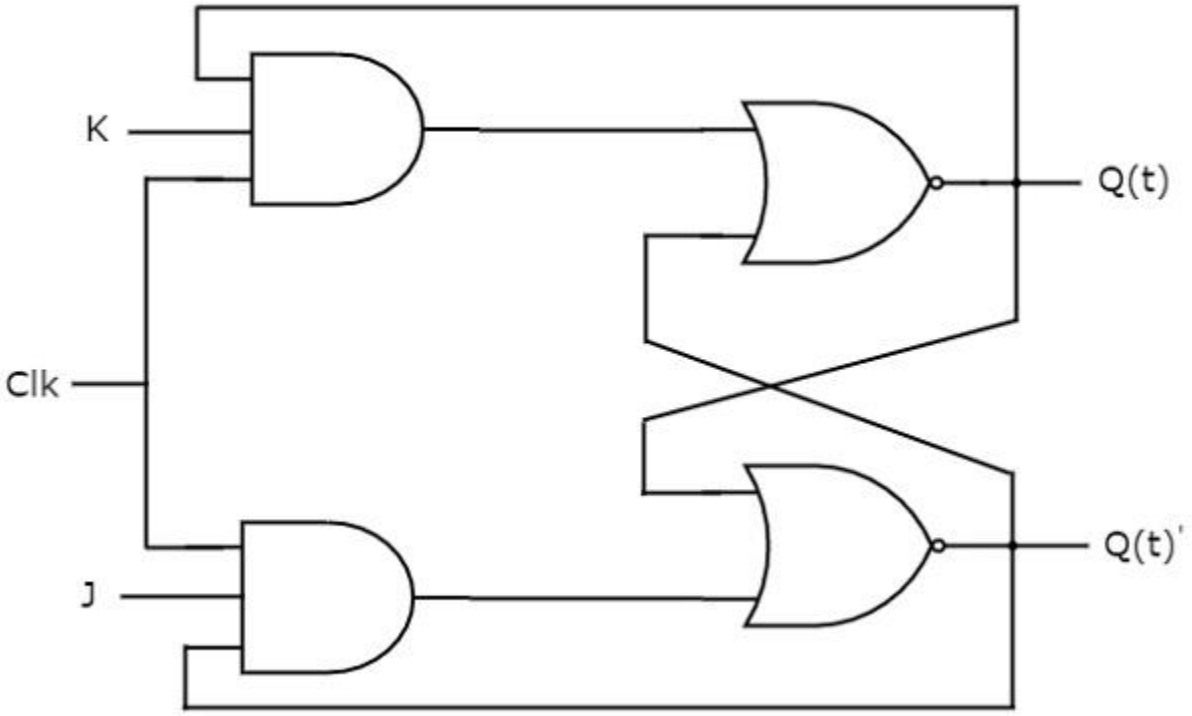
Therefore, D flip-flop always Hold the information, which is available on data input, D of earlier positive transition of clock signal. From the above state table, we can directly write the next state equation as

$$Q(t + 1) = D$$

Next state of D flip-flop is always equal to data input, D for every positive transition of the clock signal. Hence, D flip-flops can be used in registers, **shift registers** and some of the counters.

JK Flip-Flop

JK flip-flop is the modified version of SR flip-flop. It operates with only positive clock transitions or negative clock transitions. The **circuit diagram** of JK flip-flop is shown in the following figure.



This circuit has two inputs J & K and two outputs Q(t) & Q(t)'. The operation of JK flip-flop is similar to SR flip-flop. Here, we considered the inputs of SR flip-flop as **S = J Q(t)'** and **R = KQ(t)** in order to utilize the modified SR flip-flop for 4 combinations of inputs.

The following table shows the **state table** of JK flip-flop.

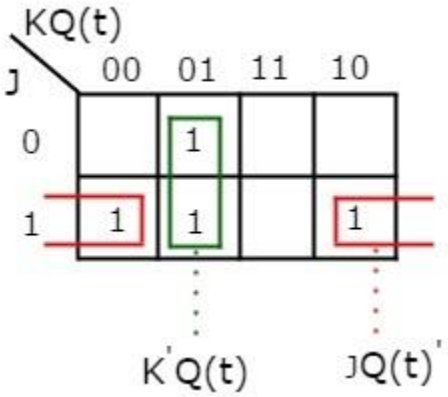
J	K	Q(t + 1)
0	0	Q(t)
0	1	0
1	0	1
1	1	Q(t)'

Here, Q(t) & Q(t + 1) are present state & next state respectively. So, JK flip-flop can be used for one of these four functions such as Hold, Reset, Set & Complement of present state based on the input conditions, when positive transition of clock signal is applied. The following table shows the **characteristic table** of JK flip-flop.

Present Inputs		Present State	Next State
J	K	Q(t)	Q(t+1)
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1

1	1	0	1
1	1	1	0

By using three variable K-Map, we can get the simplified expression for next state, Q(t + 1). **Three variable K-Map** for next state, Q(t + 1) is shown in the following figure.

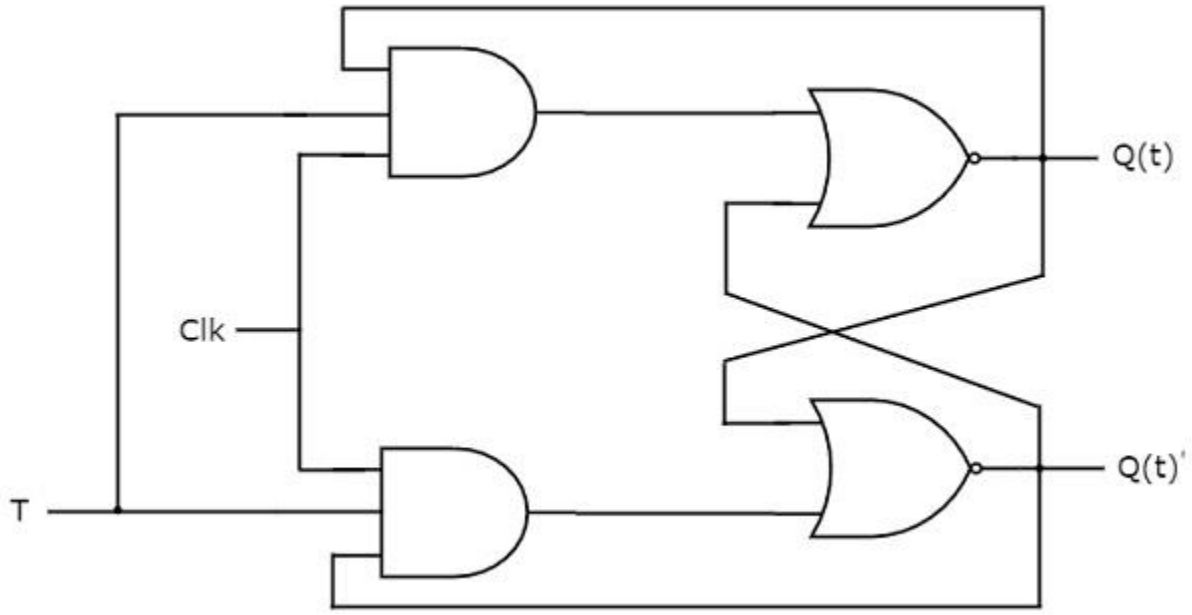


The maximum possible groupings of adjacent ones are already shown in the figure. Therefore, the **simplified expression** for next state Q(t+1) is

$$Q(t+1)=JQ(t)'+K'Q(t)Q(t+1)=JQ(t)'+K'Q(t)$$

T Flip-Flop

T flip-flop is the simplified version of JK flip-flop. It is obtained by connecting the same input 'T' to both inputs of JK flip-flop. It operates with only positive clock transitions or negative clock transitions. The **circuit diagram** of T flip-flop is shown in the following figure.



This circuit has single input T and two outputs Q(t) & Q(t)'. The operation of T flip-flop is same as that of JK flip-flop. Here, we considered the inputs of JK flip-flop as **J = T** and **K = T** in order to utilize the modified JK flip-flop for 2 combinations of inputs. So, we eliminated the other two combinations of J & K, for which those two values are complement to each other in T flip-flop.

The following table shows the **state table** of T flip-flop.

D	Q(t + 1)
0	Q(t)
1	Q(t)'

Here, Q(t) & Q(t + 1) are present state & next state respectively. So, T flip-flop can be used for one of these two functions such as Hold, & Complement of present state based on the input conditions, when positive transition of clock signal is applied. The following table shows the **characteristic table** of T flip-flop.

Inputs	Present State	Next State
T	**Q(t)**	**Q(t + 1)**
0	0	0
0	1	1
1	0	1
1	1	0

From the above characteristic table, we can directly write the **next state equation** as

$$Q(t+1)=T'Q(t)+TQ(t)'Q(t+1)=T'Q(t)+TQ(t)'$$

$$\Rightarrow Q(t+1)=T\oplus Q(t)\Rightarrow Q(t+1)=T\oplus Q(t)$$

The output of T flip-flop always toggles for every positive transition of the clock signal, when input T remains at logic High (1). Hence, T flip-flop can be used in **counters**.

In this chapter, we implemented various flip-flops by providing the cross coupling between NOR gates. Similarly, you can implement these flip-flops by using NAND gates.

In previous chapter, we discussed the four flip-flops, namely SR flip-flop, D flip-flop, JK flip-flop & T flip-flop. We can convert one flip-flop into the remaining three flip-flops by including some additional logic. So, there will be total of twelve **flip-flop conversions**.

Follow these **steps** for converting one flip-flop to the other.

- Consider the **characteristic table** of desired flip-flop.
- Fill the excitation values (inputs) of given flip-flop for each combination of present state and next state. The **excitation table** for all flip-flops is shown below.

Present State	Next State	SR flip-flop inputs		D flip-flop input	JK flip-flop inputs		T flip-flop input
Q(t)	Q(t+1)	S	R	D	J	K	T
0	0	0	x	0	0	x	0
0	1	1	0	1	1	x	1
1	0	0	1	0	x	1	1
1	1	x	0	1	x	0	0

- Get the **simplified expressions** for each excitation input. If necessary, use Kmaps for simplifying.
- Draw the **circuit diagram** of desired flip-flop according to the simplified expressions using given flip-flop and necessary logic gates.

Now, let us convert few flip-flops into other. Follow the same process for remaining flipflop conversions.

SR Flip-Flop to other Flip-Flop Conversions

Following are the three possible conversions of SR flip-flop to other flip-flops.

- SR flip-flop to D flip-flop

- SR flip-flop to JK flip-flop
- SR flip-flop to T flip-flop

SR flip-flop to D flip-flop conversion

Here, the given flip-flop is SR flip-flop and the desired flip-flop is D flip-flop. Therefore, consider the following **characteristic table** of D flip-flop.

D flip-flop input	Present State	Next State
D	**Q(t)**	**Q(t + 1)**
0	0	0
0	1	0
1	0	1
1	1	1

We know that SR flip-flop has two inputs S & R. So, write down the excitation values of SR flip-flop for each combination of present state and next state values. The following table shows the characteristic table of D flip-flop along with the **excitation inputs** of SR flip-flop.

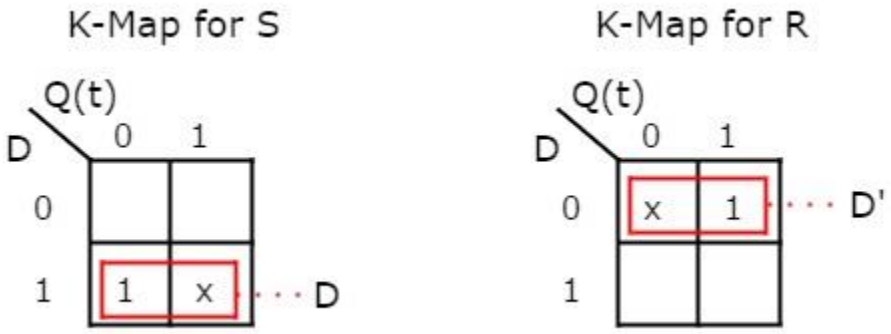
D flip-flop input	Present State	Next State	SR flip-flop inputs	
D	**Q(t)**	**Q(t + 1)**	**S**	**R**
0	0	0	0	x
0	1	0	0	1
1	0	1	1	0
1	1	1	x	0

From the above table, we can write the **Boolean functions** for each input as below.

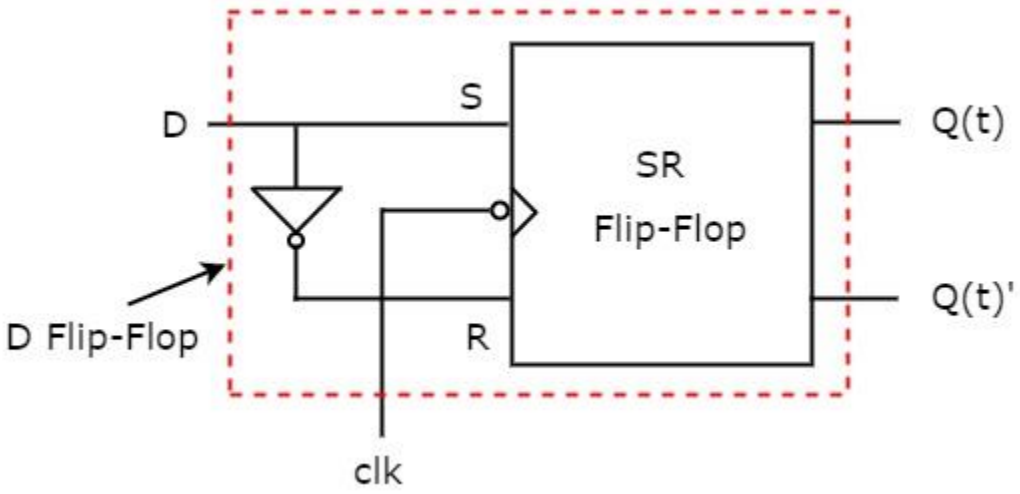
S=m2+d3S=m2+d3

R=m1+d0R=m1+d0

We can use 2 variable K-Maps for getting simplified expressions for these inputs. The **k-Maps** for S & R are shown below.



So, we got S = D & R = D' after simplifying. The **circuit diagram** of D flip-flop is shown in the following figure.



This circuit consists of SR flip-flop and an inverter. This inverter produces an output, which is complement of input, D. So, the overall circuit has single input, D and two outputs Q(t) & Q(t)'. Hence, it is a **D flip-flop**. Similarly, you can do other two conversions.

D Flip-Flop to other Flip-Flop Conversions

Following are the three possible conversions of D flip-flop to other flip-flops.

- D flip-flop to T flip-flop
- D flip-flop to SR flip-flop
- D flip-flop to JK flip-flop

D flip-flop to T flip-flop conversion

Here, the given flip-flop is D flip-flop and the desired flip-flop is T flip-flop. Therefore, consider the following **characteristic table**of T flip-flop.

T flip-flop input	Present State	Next State
T	**Q(t)**	**Q(t + 1)**
0	0	0
0	1	1
1	0	1
1	1	0

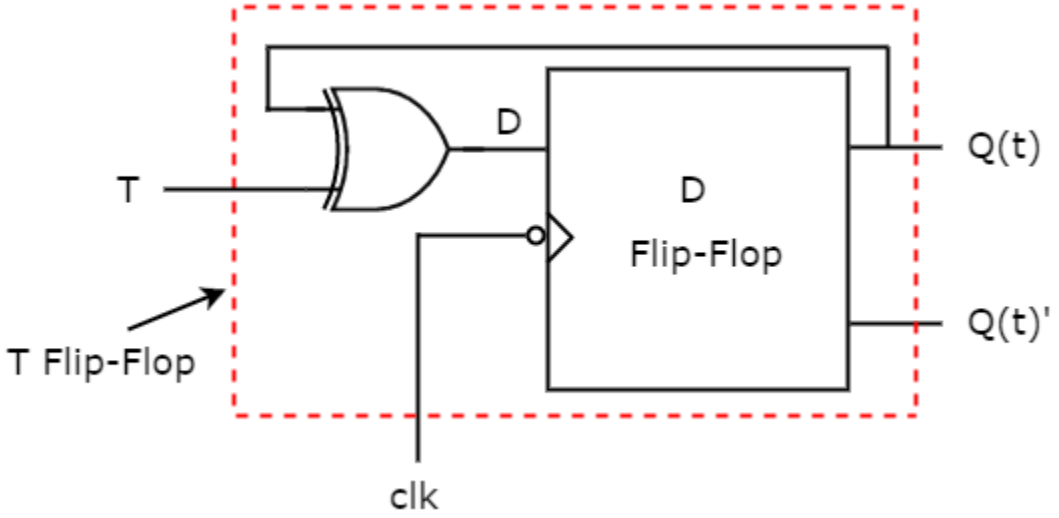
We know that D flip-flop has single input D. So, write down the excitation values of D flip-flop for each combination of present state and next state values. The following table shows the characteristic table of T flip-flop along with the **excitation input**of D flip-flop.

T flip-flop input	Present State	Next State	D flip-flop input
T	**Q(t)**	**Q(t + 1)**	**D**
0	0	0	0
0	1	1	1
1	0	1	1
1	1	0	0

From the above table, we can directly write the **Boolean function** of D as below.

$$D=T\oplus Q(t)D=T\oplus Q(t)$$

So, we require a two input Exclusive-OR gate along with D flip-flop. The **circuit diagram** of T flip-flop is shown in the following figure.



This circuit consists of D flip-flop and an Exclusive-OR gate. This Exclusive-OR gate produces an output, which is Ex-OR of T and Q(t). So, the overall circuit has single input, T and two outputs Q(t) & Q(t)'. Hence, it is a **T flip-flop**. Similarly, you can do other two conversions.

JK Flip-Flop to other Flip-Flop Conversions

Following are the three possible conversions of JK flip-flop to other flip-flops.

- JK flip-flop to T flip-flop
- JK flip-flop to D flip-flop
- JK flip-flop to SR flip-flop

JK flip-flop to T flip-flop conversion

Here, the given flip-flop is JK flip-flop and the desired flip-flop is T flip-flop. Therefore, consider the following **characteristic table** of T flip-flop.

T flip-flop input	Present State	Next State
T	**Q(t)**	**Q(t + 1)**
0	0	0
0	1	1

1	0	1
1	1	0

We know that JK flip-flop has two inputs J & K. So, write down the excitation values of JK flip-flop for each combination of present state and next state values. The following table shows the characteristic table of T flip-flop along with the **excitation inputs** of JK flipflop.

T flip-flop input	**Present State**	**Next State**	**JK flip-flop inputs**	
T	**Q(t)**	**Q(t + 1)**	**J**	**K**
0	0	0	0	x
0	1	1	x	0
1	0	1	1	x
1	1	0	x	1

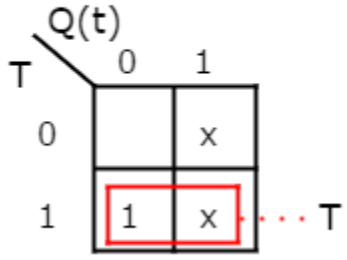
From the above table, we can write the **Boolean functions** for each input as below.

$$J=m2+d1+d3J=m2+d1+d3$$

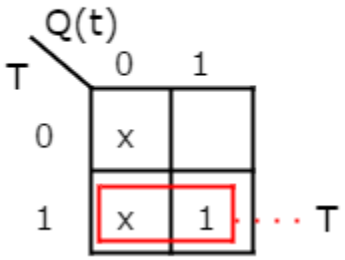
$$K=m3+d0+d2K=m3+d0+d2$$

We can use 2 variable K-Maps for getting simplified expressions for these two inputs. The **k-Maps** for J & K are shown below.

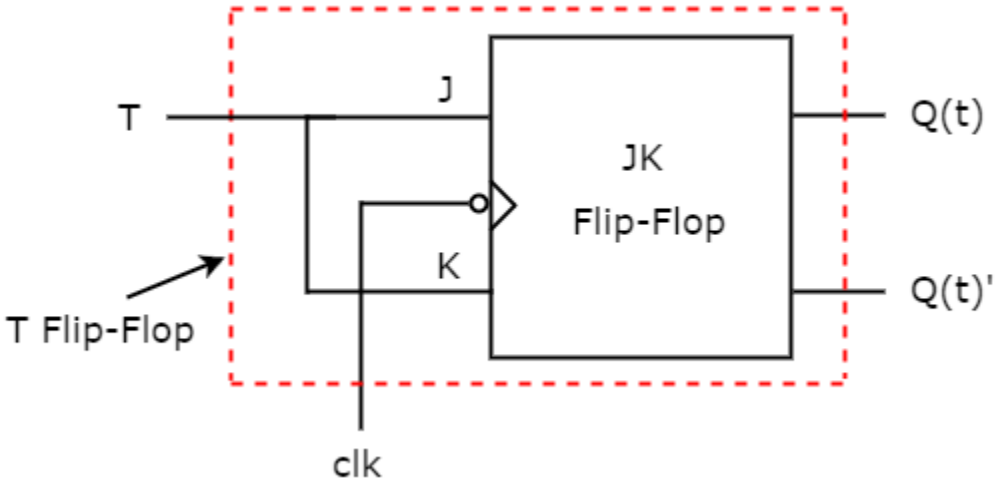
K-Map for J



K-Map for K



So, we got, J = T & K = T after simplifying. The **circuit diagram**of T flip-flop is shown in the following figure.



This circuit consists of JK flip-flop only. It doesn't require any other gates. Just connect the same input T to both J & K. So, the overall circuit has single input, T and two outputs Q(t) & Q(t)'. Hence, it is a **T flip-flop**. Similarly, you can do other two conversions.

T Flip-Flop to other Flip-Flop Conversions

Following are the three possible conversions of T flip-flop to other flip-flops.

- T flip-flop to D flip-flop
- T flip-flop to SR flip-flop
- T flip-flop to JK flip-flop

T flip-flop to D flip-flop conversion

Here, the given flip-flop is T flip-flop and the desired flip-flop is D flip-flop. Therefore, consider the characteristic table of D flip-flop and write down the excitation values of T flip-flop for each combination of present state and next state values. The following table shows the **characteristic table** of D flip-flop along with the **excitation input** of T flip-flop.

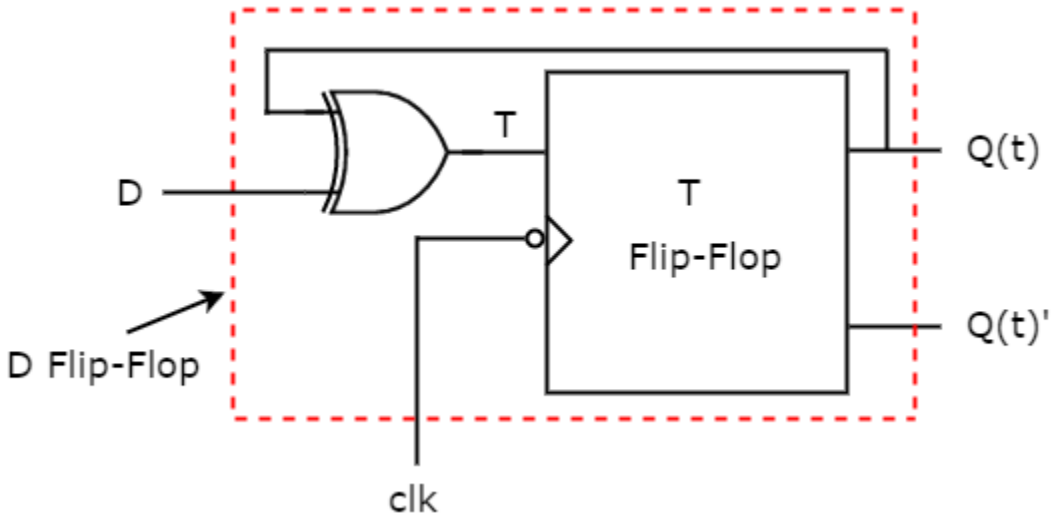
D flip-flop input	Present State	Next State	
D	**Q(t)**	**Q(t + 1)**	**T**
0	0	0	0

0	1	0	1
1	0	1	1
1	1	1	0

From the above table, we can directly write the Boolean function of T as below.

$$T=D \oplus Q(t) T=D \oplus Q(t)$$

So, we require a two input Exclusive-OR gate along with T flip-flop. The **circuit diagram** of D flip-flop is shown in the following figure.



This circuit consists of T flip-flop and an Exclusive-OR gate. This Exclusive-OR gate produces an output, which is Ex-OR of D and Q(t). So, the overall circuit has single input, D and two outputs Q(t) & Q(t)'. Hence, it is a **D flip-flop**. Similarly, you can do other two conversions.

MALLA REDDY COLLEGE OF ENGINEERING & TECHNOLOGY

(Autonomous Institution – UGC, Govt. of India)

II B. Tech I Semester Regular Examinations, Nov 2019

ANALOG & DIGITAL ELECTRONICS

(Common to CSE & IT)

Roll No										

Time: 3 hours **Max. Marks: 70**

Note: This question paper Consists of 5 Sections. Answer **FIVE** Questions, Choosing ONE Question from each SECTION and each Question carries 14 marks.

SECTION-I

1a. Explain the effect of temperature on V-I characteristics of a diode. [7]

b. Distinguish between drift and diffusion current in a semiconductor. [7]

OR

2.a Draw the equivalent circuits of diode [7]

b)How a PN junction diode works? Draw and explain V-I characteristics of PN diode with neat diagram [7]

SECTION-II

3.(a) Explain different current components in a transistor. [7]

(b) Calculate the values of I_E, α and β for a transistor with I_B=13μA, I_C=200mA, I_{CBO}=6μA [7]

OR

4.(a) Draw the circuit diagram of a transistor in CB configuration and explain the output Characteristics with the help of different regions. [14]

SECTION-III

5. (a)Convert the given expression in standard SOP form f(A,B,C)=AC+BA+BC [7]

(b)Convert the given expression in standard POS form Y=A.(A+B+C) [7]

OR

6. Find the complement of the following Boolean functions and reduce them to minimum number of literals.

a) (b c' +a' d) (ab' +cd') [7]

b) (b' d+ a' b c' +a c d+ a' b c) [7]

SECTION-IV

7. Simplify the following Boolean functions, using Karnaugh maps:

i. $F(w,x,y,z) = \sum m(11,12,13,14,15)$ [7]

ii. $F(A,B,C,D) = \sum m(0,2,4,5,6,7,8,10,13,15)$ [7]

OR

8. (a) Draw the multiple-level NAND circuit for the following expression: $w(x+y+z)+xyz$

(b) Draw a logic diagram using only two-input NOR gates to implement the following function: $F(A,B,C,D) = (A \oplus B)'(C \oplus D)$

SECTION-V

9.(a)Design full-adder using half adders.

(b)Realize full adder using two half adders and logic gates.

OR

10. (a)Draw the logic diagram of a JK flip- flop and using excitation table explain its operation

(b)Convert the JK flip into T flip-flop

MALLA REDDY COLLEGE OF ENGINEERING & TECHNOLOGY

(Autonomous Institution – UGC, Govt. of India)

II B. Tech I Semester Regular Examinations, Nov 2019

ANALOG & DIGITAL ELECTRONICS

(Common to CSE & IT)

Roll No										

Time: 3 hours **Max. Marks: 70**

Note: This question paper Consists of 5 Sections. Answer **FIVE** Questions, Choosing ONE Question from each SECTION and each Question carries 14 marks.

SECTION-I

1(a). Find the value of D.C. resistance and A.C resistance of a Ge junction diode at 25°C with reverse saturation current, I_o = 25μA and at an applied voltage of 0.2V across the diode. [7]

(b) Explain about Zener Diode characteristics [7]

OR

2. Explain the operation of PN Junction Diode with neat diagrams? [14]

SECTION-II

3.(a) Based on the currents flowing through a BJT illustrate the amplification process. [7]

(b) In a germanium transistor collector current is 51mA, when base current is 0.04mA. If $h_{fe} = \beta_{dc} = 51$, Calculate cut off current, I_{CEO} . [7]

OR

4. (a) Draw the circuit diagram of a transistor in CB configuration and explain the output characteristics with the help of different regions. [7]

(b) Compare CB, CC, and CE configurations. [7]

SECTION-III

5. Implement the following Boolean function using minimum number of basic gates. [14]

(a) (AB + AB') (AB)' (b) [(ABD(C + D + E)) + (A +DBC)'] (ABC + (CAD)')

OR

6. (a) Express the following numbers in decimal: [7]

(i) (26.24)8 (ii) (16.5)16

(b) Convert the following number to Hexadecimal: [7]

i) (735.5)8 ii) (1011011)2

SECTION-IV

7.Simplify the following Boolean functions, using Karnaugh maps:

i. $F(x, y, z) = \sum(2, 3, 6, 7)$ $ii. F(A, B, C, D) = \sum(2, 3, 6, 7, 12, 13, 14)$ [14]

OR

8.(a) Draw the multiple-level NOR circuit for the following expressions: [7]

$CD(B + C)A + (BC' + DE')$

(b) Simplify and implement the following function with two-level NAND gate circuit: [7]

$F(A, B, C, D) = A'B'C'D + CD + AC'D$

SECTION-V

9.(a)Define decoder. Construct 3x8 decoder using logic gates and truth table. [7]

(b)Define an encoder. Design octal to binary encoder. [7]

OR

10. (a)Convert JK flip-flop to T flip-flop b) Convert RS flip-flop to D flip-flop [7]

(b)What is the drawback of JK flip-flop? How is it eliminated in Master Slave flip-flop? Explain. [7]

MALLA REDDY COLLEGE OF ENGINEERING & TECHNOLOGY

(Autonomous Institution – UGC, Govt. of India)

II B. Tech I Semester Regular Examinations, Nov 2019

ANALOG & DIGITAL ELECTRONICS

(Common to CSE & IT)

Roll No										

Time: 3 hours **Max. Marks: 70**

Note: This question paper Consists of 5 Sections. Answer **FIVE** Questions, Choosing ONE Question from each SECTION and each Question carries 14 marks.

SECTION-I

1.Draw the basic structure of PN Junction Diode and explain its operation and V-I Characteristics [14]

OR

2.(a) Explain the V-I characteristics of Zener diode ? [7]

(b)Distinguish between Avalanche and Zener Break downs. [7]

SECTION-II

3. (a)Explain the input and output characteristics of a transistor in CC configuration [7]

(b)Calculate the collector current and emitter current for a transistor with $\alpha_{D.C.} = 0.99$ and $I_{CBO} = 50\ \mu A$ when the base current is 20µA [7]

OR

4. (a) Summarize the salient features of the characteristics of BJT operatives in CE, CB and CC configurations. [7]

(b) Calculate the collector current and emitter current for a transistor with $\alpha_{D.C.} = 0.99$ and $I_{CBO} = 20\ \mu A$ when the base current is 50µA. [7]

SECTION-III

5.(a)Find the complement and dual of the given function: XY+X(WZ+WZ') [7]

(b)Convert the following numbers to Binary [7]

(i) $(27.315)_{10}$ (ii) $(68BE)_{16}$

OR

6. (a) Reduce the following Boolean function to four literals and draw the logic diagram:

(A'+C)(A'+C')(A+B+C'D) [7]

(b) Convert the following numbers to Octal:

(i) $(1010.1010)_2$ (ii) $(FAFA)_{16}$ [7]

SECTION-IV

7(a) Reduce the following function using k-map technique $F(A,B,C,D)=\pi(0,2,3,8,9,12,13,15)$ [7+7]

(b) Minimize the expression using k-map y=(A+B+C') (A+ B+ C) (A' + B' + C') (A' + B +C) (A+B+C)

OR

8.Simplify the following Boolean expressions using K-map and implement it by using NOR gates. [14]

a) F(A,B,C,D)=AB'C' +AC+A'CD' b)F(W,X,Y,Z)=w' x'y'z' + wxy'z' + w'x'yz + wxyz

SECTION-V

9.(a)Convert D flip-flop into T and JK flip-flops. [7]

(b) Implement a full adder using 8X1 multiplexer. [7]

OR

10.(a)Design a 1:8 demultiplexer using two 1:4 demultiplexer. [3]

(b)Convert JK flip-flop into D flip-flop [4]

(c) Construct a 5-to-32-line decoder with four 3-to-8-line decoders with enable and a 2-to- 4 line decoder. Use block diagrams for the components. [7]

MALLA REDDY COLLEGE OF ENGINEERING & TECHNOLOGY

(Autonomous Institution – UGC, Govt. of India)

II B. Tech I Semester Regular Examinations, Nov 2019

ANALOG & DIGITAL ELECTRONICS

(Common to CSE & IT)

Roll No										

Time: 3 hours **Max. Marks: 70**

Note: This question paper Consists of 5 Sections. Answer **FIVE** Questions, Choosing ONE Question from each SECTION and each Question carries 14 marks.

SECTION-I

1(a)Explain semi-conductors, insulators and metals classification using energy band diagrams. [7]

(b) Explain in detail the break down mechanisms in a diode. [7]

OR

2 (a) Explain the working of pn diode in forward and reverse bias conditions [7]

(b) Draw and explain VI characteristics of Si & Ge diode. [7]

SECTION-II

3. (a) Draw the circuit diagram of a transistor in CB configuration and explain the output characteristics with the help of different regions. [7]

(b) Explain the working of a PNP transistor with a neat diagram [7]

OR

4 (a) Explain the working of a NPN transistor. [7]

(b)Derive an expression between transistor parameters (α, β, γ)? [7]

SECTION-III

5.(a)Implement all logic gates using NAND gates. [4]

(b)Write the following Boolean expression in product of sums form: $a'b + a'c' + abc$ [4]

(c)Find the dual and complement of the following function: A'BD'+B'(C'+D')+A'C [6]

OR

6.(a) Obtain the 1's and 2's complements of the following binary numbers: [7]

(i) 00010000 (ii) 00000000 (iii) 11011010 (iv) 10101010

(v) 10000101 (vi) 11111111

(b)Write the following Boolean expressions in sum of products form: $(b + d)(a' + b' + c)$ [7]

SECTION-IV

7.(a) Simplify the following using K- map and implement the same using NAND gates. [7+7]

Y (A, B, C) =∑(0,2,4,5,6,7)

(b) Implement the following Boolean function with only two input NAND gates: F=(AB'+D')E+C(A'+B')

OR

8.(a)Simplify the following Boolean function with the don't conditions d using Kmap method:

F(A, B, C, D)=Σ(1,3,8,10,15); d(A, B, C, D)=Σ(0, 2, 9) [7+7]

(b)Implement the following Boolean function with only two input NOR gates: F=(AB'+CD')E+BC(A+B)

SECTION-V

9.(a)Define decoder. Construct 3x8 decoder using logic gates and truth table. [7]

(b)Define an encoder. Design octal to binary encoder. [7]

OR

10. (a)Convert JK flip-flop to T flip-flop b) Convert RS flip-flop to D flip-flop [7]

(b)What is the drawback of JK flip-flop? How is it eliminated in Master Slave flip-flop? Explain. [7]