

Analog and Digital Electronics

Prepared by :

N Naresh Chary
Assistant Professor
Department of ECE
NRCM,Hyderabad

UNIT - I

Diodes and applications

Atomic Structure

An atom is composed of :

- Nucleus (which contains positively charged protons and neutral neutrons)
- Electrons (which are négative charged and that orbit the nucleus)

Valence Electrons

- Electrons are distributed in various shells at different distances from nucleus
- Electron energy increases as shell radius increases.
- Electrons in the outer most Shell are called as valence electrons
- Elements in the period table are grouped according to the number of valence electrons.

Valence Electrons

Table 1.1 A portion of the periodic table

III	IV	V
B	C	
Al	Si	P
Ga	Ge	As

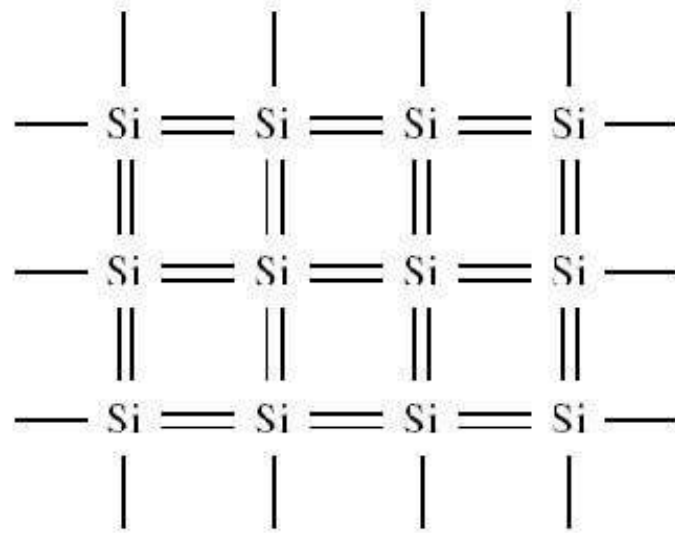
Elemental/Compound Semiconductor

- Silicon (Si) and Germanium (Ge) are in group IV, and are **elemental semiconductors**
- Gallium arsenide (GaAs) is a group III-V **compound semiconductor**

Silicon Crystal

- → At 0°K, each electron is in its lowest possible energy state, and each covalent bonding position is filled.
- → If a small electric field is applied, the electrons will not move → silicon is an **insulator**

Silicon Atom Diagram at 0°K

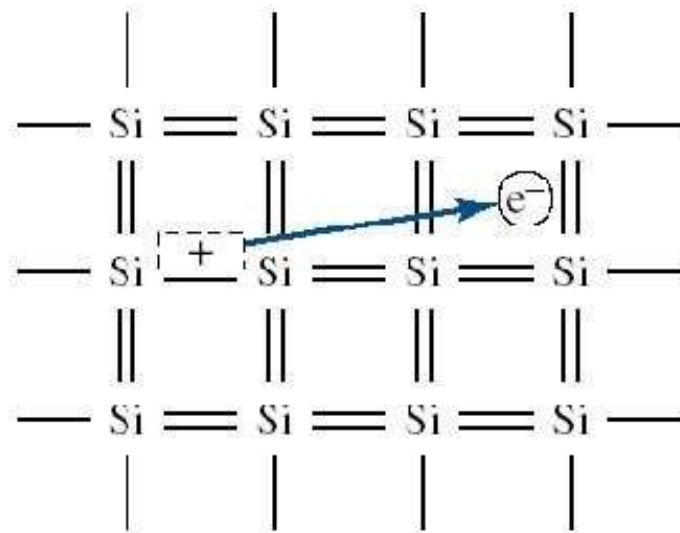


Two-dimensional
representation of the silicon
crystal at $T = 0^\circ\text{K}$

Intrinsic Silicon

- → If the temperature increases, the valence electrons will gain some thermal energy, and breaks free from the covalent bond → It leaves a positively charged hole
- → In order to break from the covalent bond, a valence electron must gain a minimum energy ***E_g***:
Bandgap energy

Silicon Atom Diagram at Ambient Temp



The breaking of
a covalent bond for $T > 0^\circ\text{K}$

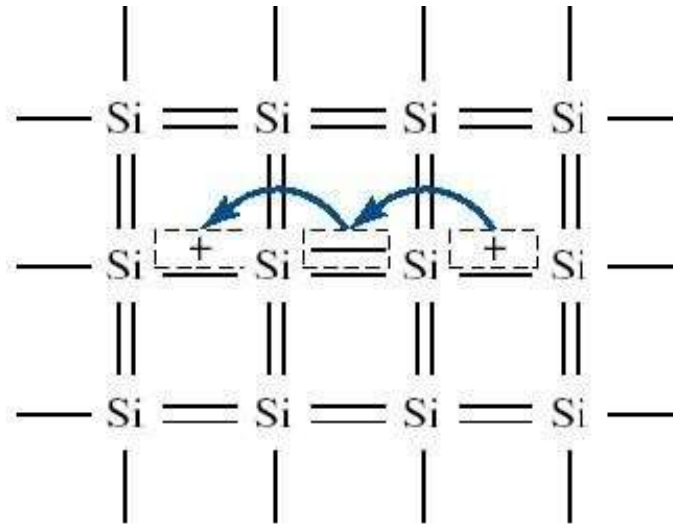
Insulators & Conductors

- Materials that have large bandgap energies (in the range of 3 to 6 electron-volts (eV)) are **insulators**, because at room temperature, essentially no free electron exists in the material
- Materials that contain very large number of free electrons at room temperature are **conductors**

Semiconductors

- In a semiconductor, the bandgap energy is in the order of 1 eV. The net flow of free electrons causes a current.
- In a semiconductor, two types of charged particles contribute to the current: the negatively charged electrons and the positively charged holes

Movement of Holes



A two- dimensional
representation of
the silicon crystal showing the
movement of the positively
charged hole

Semiconductor Constants

- The concentration of electrons and holes directly influence the magnitude of the current
- In an intrinsic semiconductor (a single crystal semiconductor) the densities of holes and electrons are equal.

n_i : intrinsic carrier concentration for free electrons (same for holes)

$$n_i = BT^{3/2} e^{\left(\frac{-E_g}{2kT}\right)}$$

B: constant related to specific semiconductor material

E_g: Bandgap energy (eV)

T: Temperature (°K)

K: Boltzman Constant (86 E-06 eV/°K)

Semiconductor Constants

Semiconductor constants

Material	E_g (eV)	B (cm⁻³ °K^{-3/2})
Silicon (Si)	1.1	5.23×10^{15}
Gallium arsenide (GaAs)	1.4	2.10×10^{14}
Germanium (Ge)	0.66	1.66×10^{15}

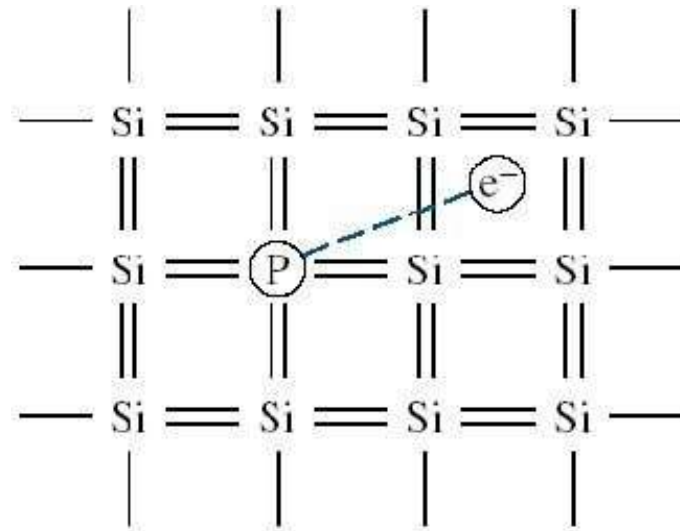
Extrinsic Semiconductor / Doping

- The electron or hole concentration can be greatly increased by adding controlled amounts of certain impurities
- For silicon, it is desirable to use impurities from the group III and V.
- An N-type semiconductor can be created by adding phosphorus or arsenic

Extrinsic Semiconductor / Doping

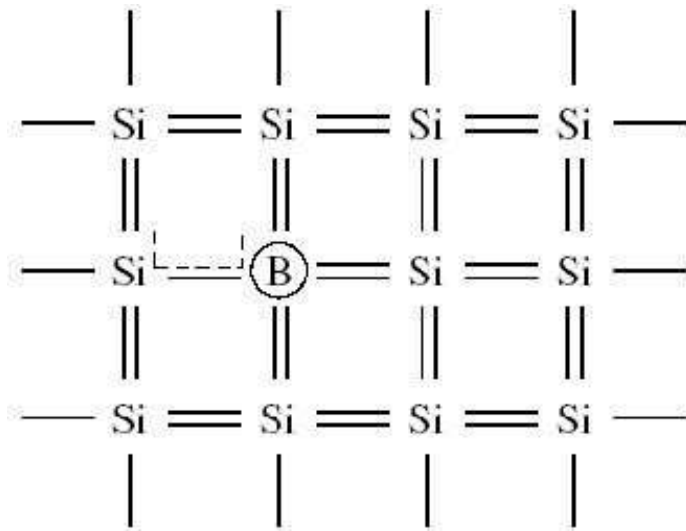
- The phosphorus (group **V**) atom is called **donor impurity** because it donates an electron that is free to move
- The boron (group **III**) has accepted a valence electron (or donated a hole), it is therefore called **acceptor impurity**

N-Type Semiconductor



Two-dimensional
representation of a silicon
lattice doped with a
phosphorus atom

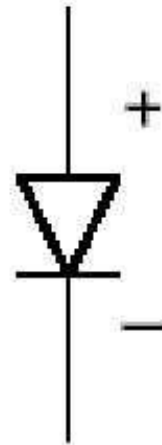
P-Type Semiconductor



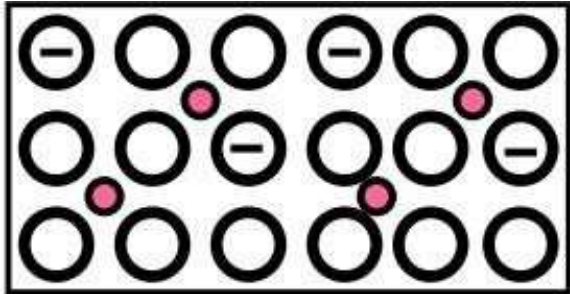
Two-dimensional
representation of a silicon
lattice doped with a boron
atom

Introduction to Semiconductor Devices

Semiconductor p-n junction diodes



p-n junction formation

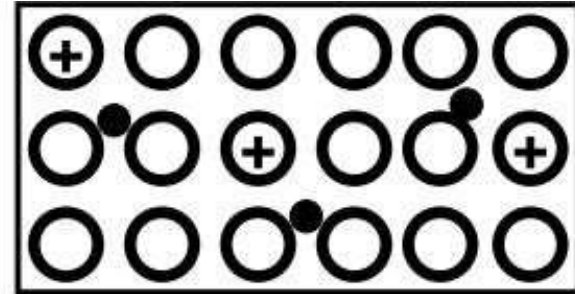


p-type material

Semiconductor material doped with **acceptors**.

Material has high hole concentration

Concentration of free electrons in p-type material is very low.



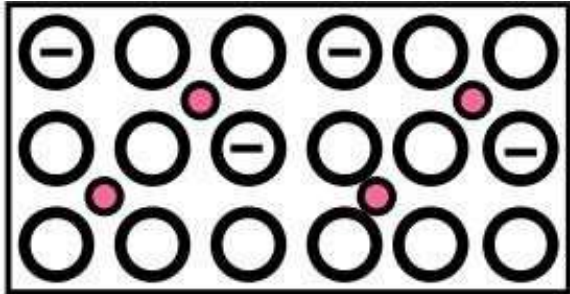
n-type material

Semiconductor material doped with **donors**.

Material has high concentration of free electrons.

Concentration of holes in n-type material is very low.

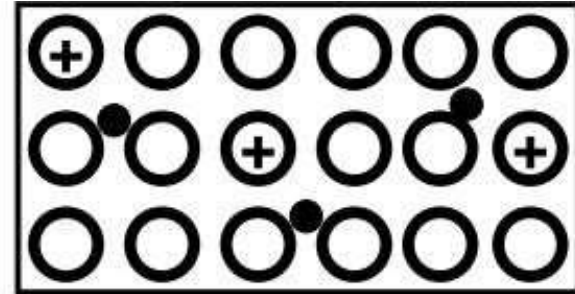
p-n junction formation



p-type material :

Contains
NEGATIVELY
charged acceptors
(immovable) and
POSITIVELY charged
holes (free).

Total charge = 0

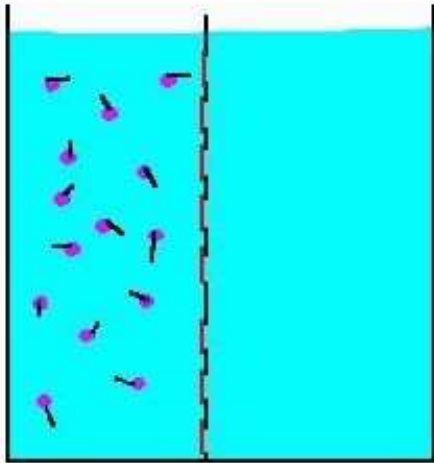


n-type material :

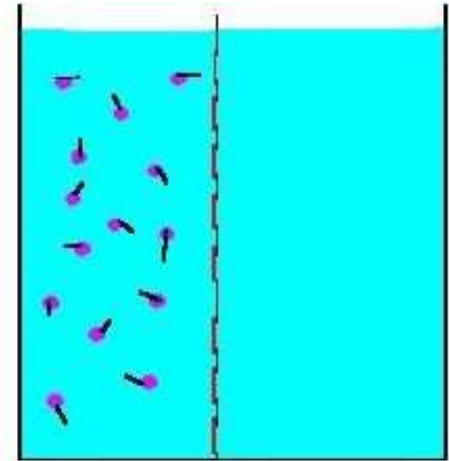
Contains POSITIVELY
charged donors
(immovable) and
NEGATIVELY
charged free electrons.

Total charge = 0

Diffusion

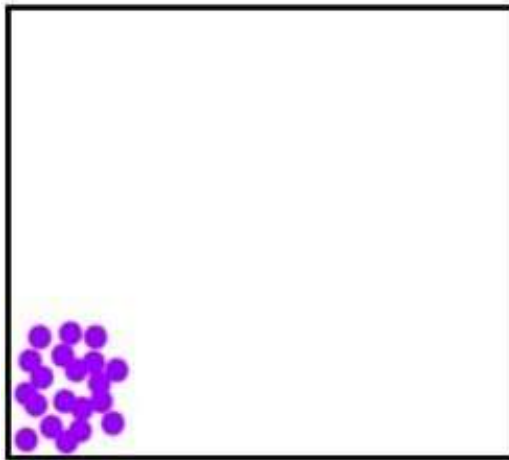


A substance, the purple dots, in solution. A membrane prevents movement of the water and the molecules from crossing from one side of the beaker to the other.

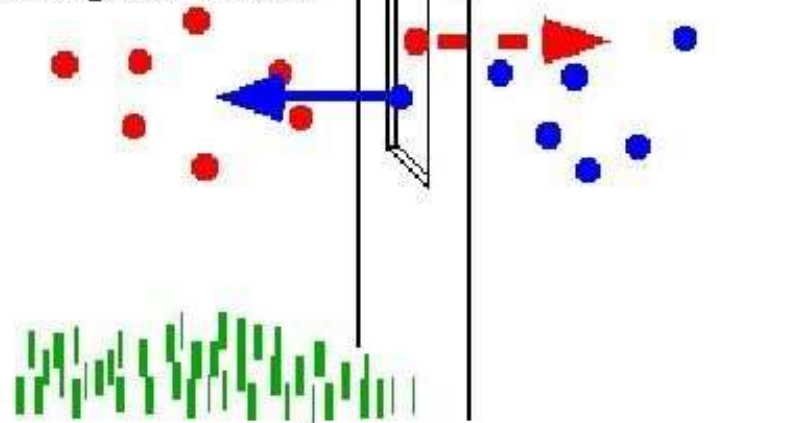


Now that the gates have been opened, the random movements of the molecules have caused, overtime, the number of molecules to be equal on the two sides of the barrier.

Diffusion



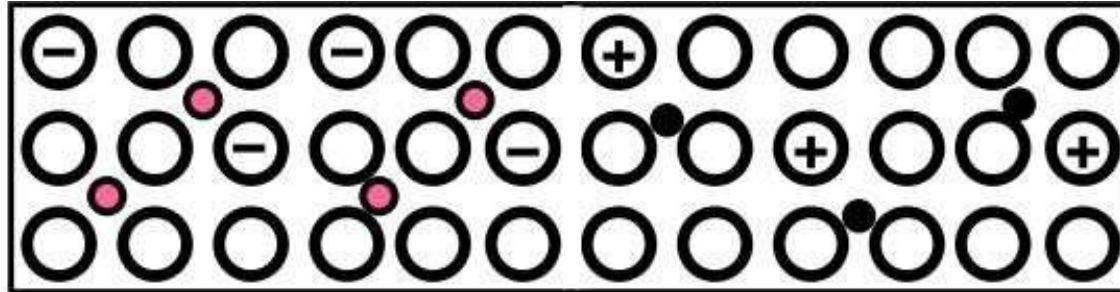
Diffusion of Air is:
like the exchange of
molecules from outside the
house to inside the house
through the window



As a result of diffusion, the molecules or other free particles distribute uniformly over the entire volume

p- n junction formation

What happens if n- and p-type materials are in close contact?



Being free particles, **electrons** start diffusing from n-type material into p-material

Being free particles, **holes**, too, start diffusing from p-type material into n-material

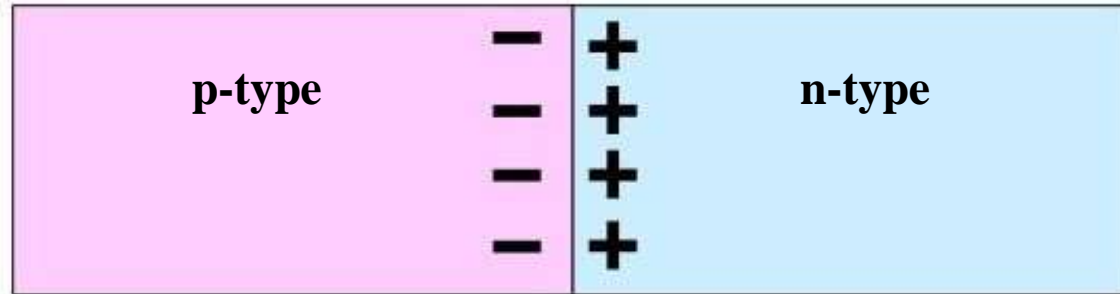
Have they been NEUTRAL particles, eventually all the free electrons and holes had uniformly distributed over the entire compound crystal.

However, every electrons transfers a negative charge ($-q$) onto the p-side and also leaves an uncompensated ($+q$) charge of the donor on the n-side.

Every hole creates one positive charge (q) on the n-side and ($-q$) on the p-side

p- n junction formation

What happens if n- and p-type materials are in close contact?



Electrons and holes remain staying close to the p-n junction because negative and positive charges attract each other.

Negative charge stops electrons from further diffusion

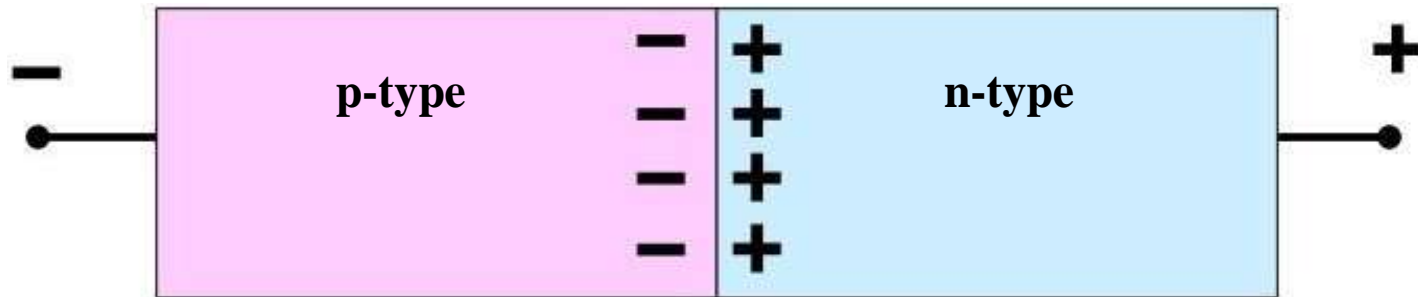
Positive charge stops holes from further diffusion

The diffusion forms a dipole charge layer at the p-n junction interface.

There is a “built-in” VOLTAGE at the p-n junction interface that prevents penetration of electrons into the p-side and holes into the n-side.

p- n junction current - voltage characteristics

What happens when the voltage is applied to a p-n junction?



The polarity shown, attracts holes to the left and electrons to the right.

According to the **current continuity law**, the current can **only** flow if all the charged particles move forming a closed loop

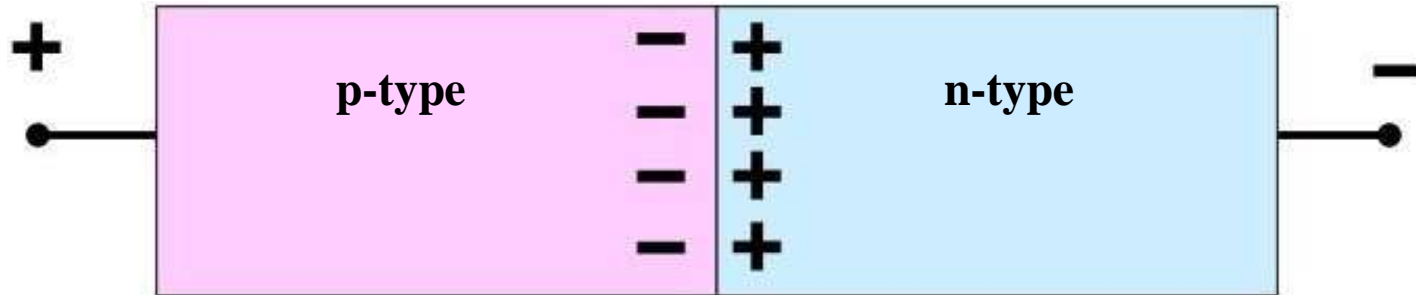
However, there are very few holes in n-type material and there are very few electrons in the p-type material.

There are very few carriers available to support the current through the junction plane

For the voltage polarity shown, the current is nearly zero

p- n junction current - voltage characteristics

What happens if voltage of opposite polarity is applied to a p-n junction?



The polarity shown, attracts electrons to the left and holes to the right.

There are plenty of electrons in the n-type material and plenty of holes in the p-type material.

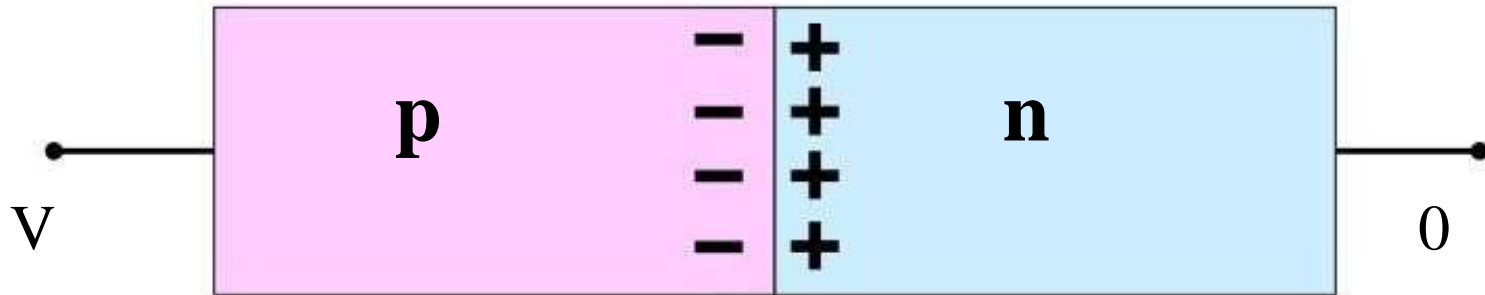
There are a lot of carriers available to cross the junction.

When the voltage applied is lower than the built-in voltage, the current is still nearly zero

When the voltage exceeds the built-in voltage, the current can flow through the p-n junction

Diode current - voltage (I-V) characteristics

Semiconductor diode consists of a p-n junction with two contacts attached to the p- and n- sides



$$I = I_S \exp\left(\frac{qV}{kT}\right) - 1$$

I_S is usually a very small current, $I_S \approx 10^{-17} \dots 10^{-13}$ A

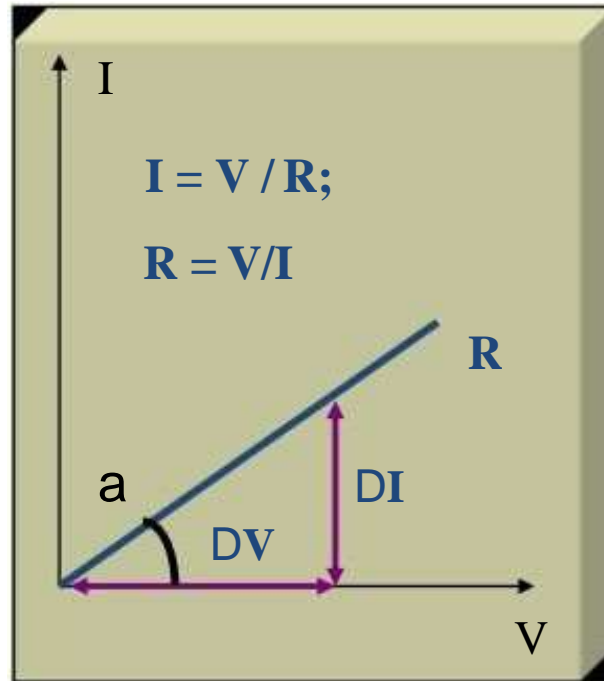
When the voltage V is negative (“reverse” polarity) the exponential term ≈ -1 ;
The diode current is $\approx I_S$ (very small).

When the voltage V is positive (“forward” polarity) the exponential term increases rapidly with V and the current is high.

Graphing the I-V characteristics of electronic component

The I-V plot represents is the dependence of the current I through the component on the voltage V across it.

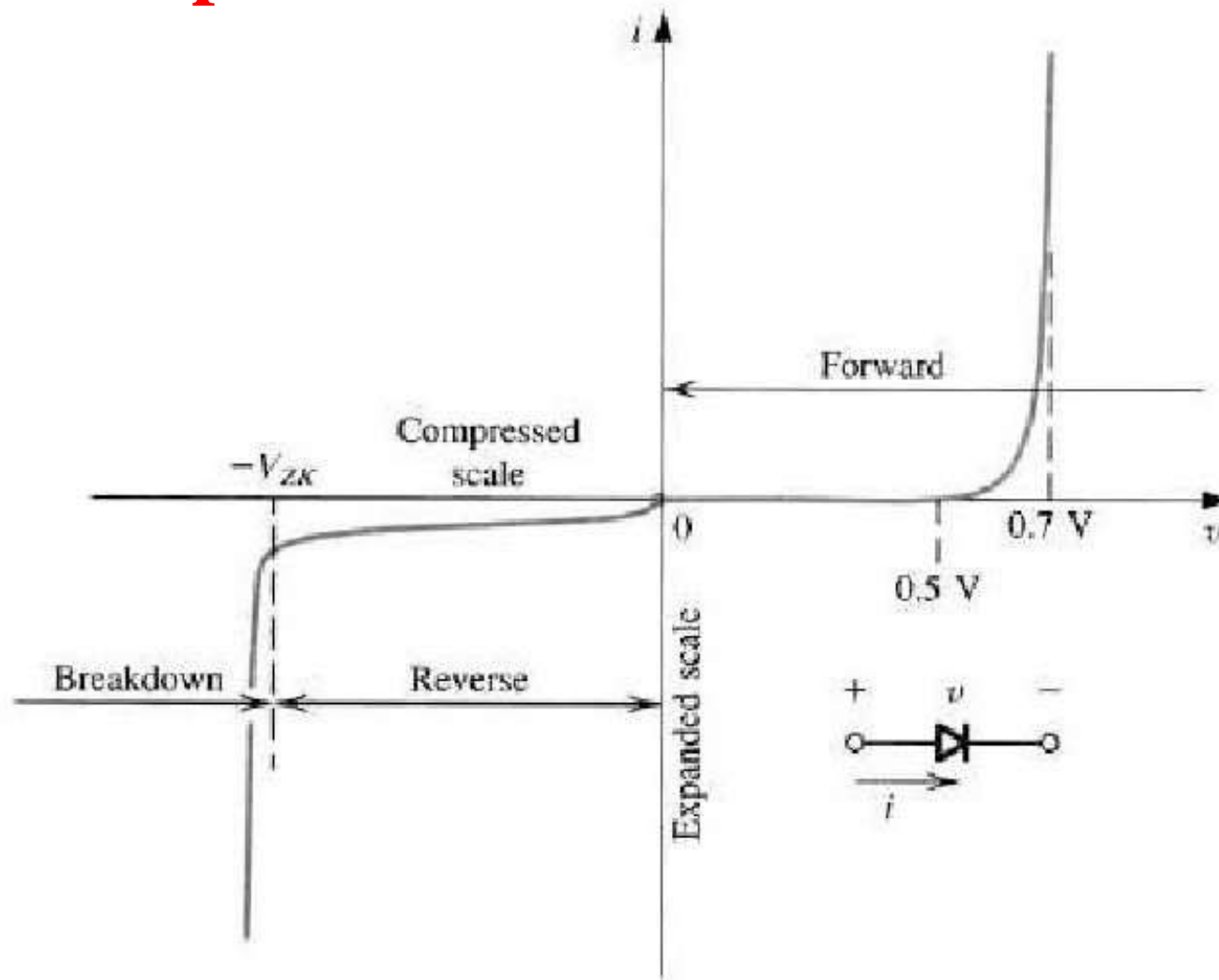
Resistor $V = I \cdot R; \quad I = \frac{V}{R}$



$$\text{tg}(a) = 1/R$$

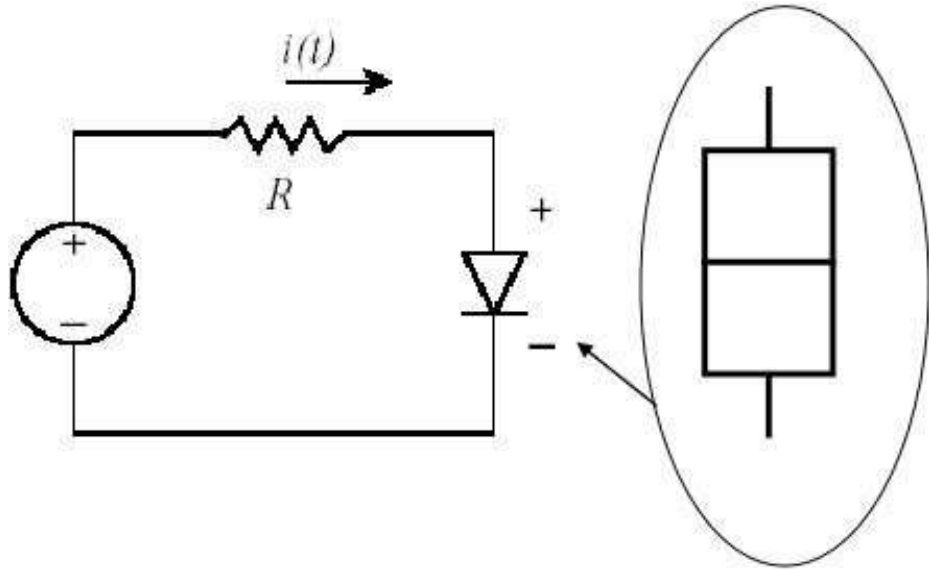
The I-V characteristic of the resistor

The experimental I-V characteristic of a Si



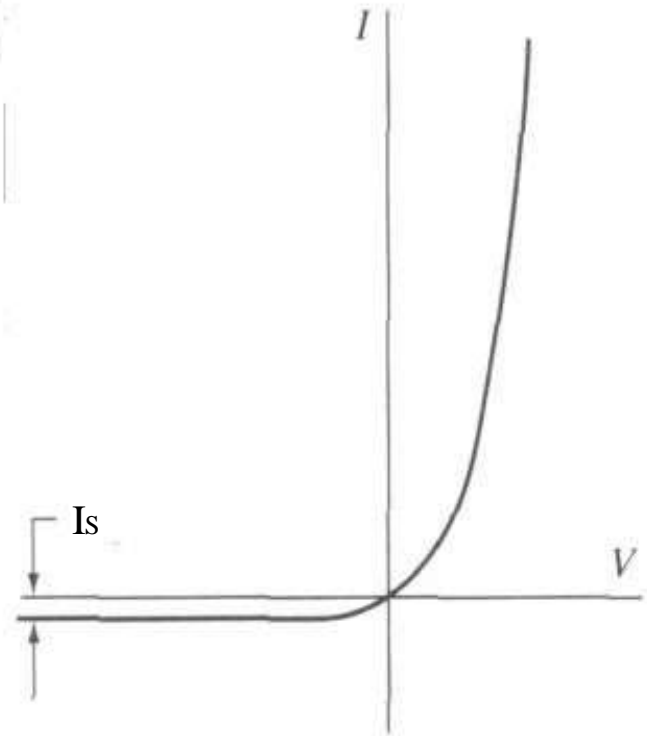
The diode $i-v$ relationship with some scales expanded and others compressed in order to reveal details.

p- n diode circuit notation

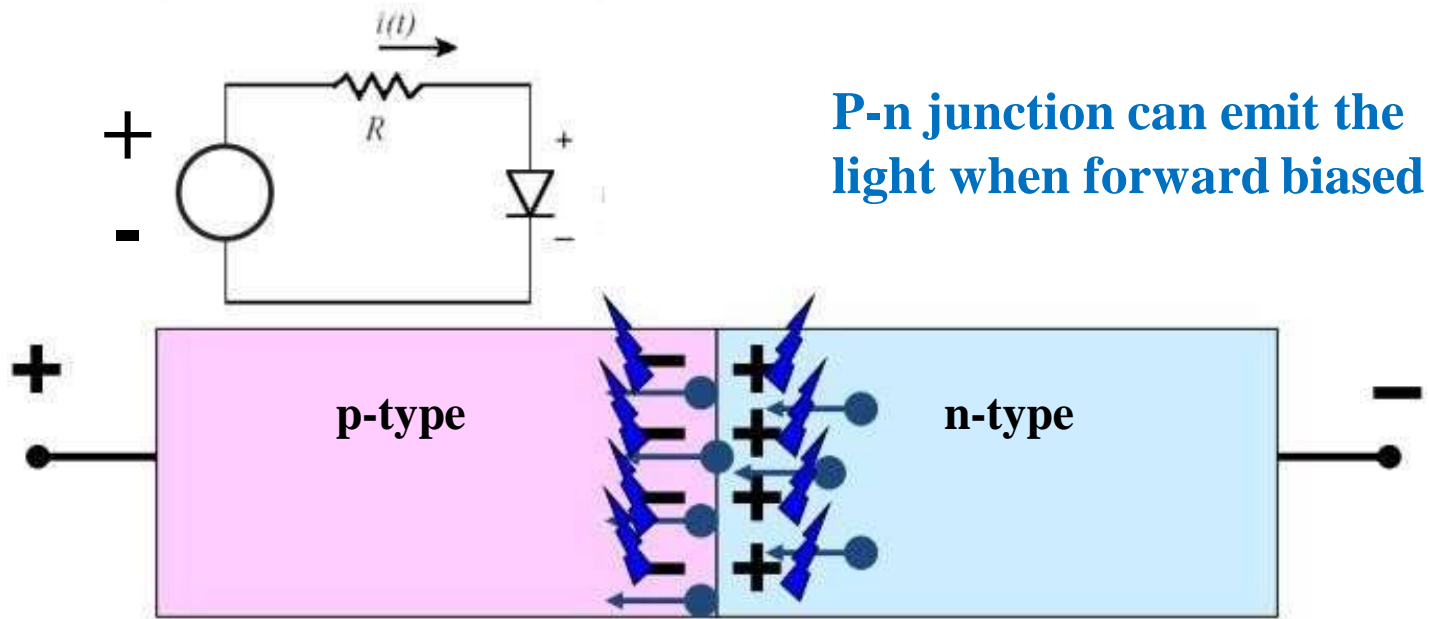


When “plus” is applied to the p-side, $I \cdot I_s e^{\dots}$ the current is high. This voltage polarity is called **FORWARD**.

When “plus” is applied to the n-side, the current is nearly zero. This voltage polarity is called **REVERSE**.



p- n diode applications Light emitters



P-n junction can emit the light when forward biased

Electrons drift into p-material and find plenty of holes there. They “RECOMBINE” by filling up the “empty” positions.

Holes drift into n-material and find plenty of electrons there. They also “RECOMBINE” by filling up the “empty” positions.

The energy released in the process of “annihilation” produces PHOTONS - the particles of light

Negative resistance devices

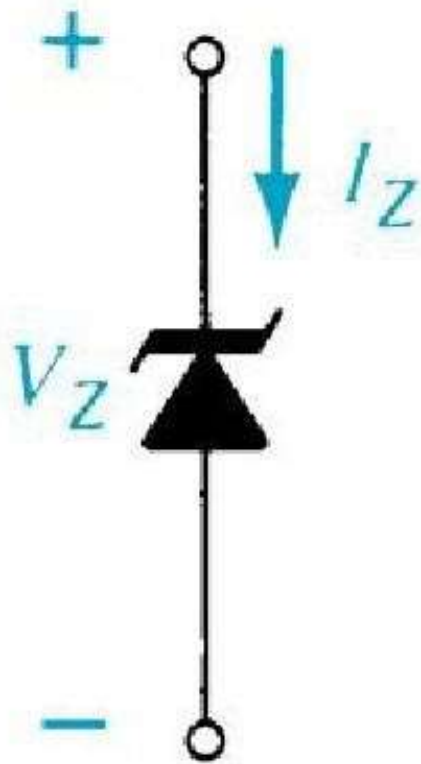
- It is a device which exhibits a negative incremental resistance over a limited range of V-I characteristic.
- It is of two types :-
 - **1. Current controllable type** : V-I curve is a multi valued function of voltage and single valued function of current .eg:- UJT, p-n-p-n diode
 - **2. Voltage controllable type** : V-I curve is a multi valued function of current and single valued function of voltage. eg:- SCS, Tunnel diode

TUNNEL DIODE (Esaki Diode)

- **It was introduced by Leo Esaki in 1958.**
- **Heavily-doped p-n junction**
 - **Impurity concentration is 1 part in 10^3 as compared to 1 part in 10^8 in p-n junction diode**
- **Width of the depletion layer is very small (about 100 Å).**
- **It is generally made up of Ge and GaAs.**
- **It shows tunneling phenomenon.**
- **Circuit symbol of tunnel diode is :**



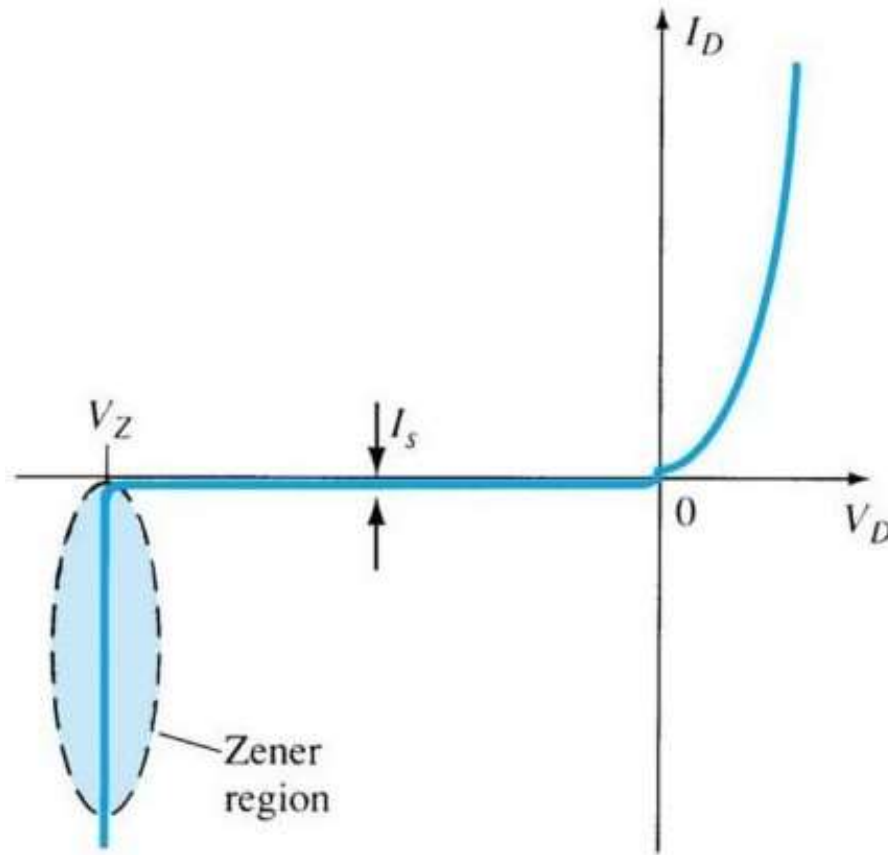
Zener Diode



(a)

- A Zener is a diode operated in reverse bias at the Peak Inverse Voltage (PIV) called the Zener Voltage (V_Z).
- Common Zener Voltages: 1.8V to 200V

Zener Region



The diode is in the reverse bias condition.

At some point the reverse bias voltage is so large the diode breaks down.

The reverse current increases dramatically.

This maximum voltage is called *avalanche breakdown voltage* and the current is called *avalanche current*.

Resistance Levels

Semiconductors act differently to DC and AC currents.
There are 3 types of resistances.

- DC or Static Resistance
- AC or Dynamic Resistance
- Average AC Resistance

- DC or Static Resistance

- The resistance of a diode at a particular operating point is called the dc or static resistance diode. It can be determined using equation (1.1):

$$R_D = V_D / I_D$$

(1.1)

Example : DC or Static Resistance - refer Figure 1.1

Ideal diode			Si diode		
$I_D(\text{A})$	$V_D(\text{V})$	$R_D(\cdot)$	$I_D(\text{A})$	$V_D(\text{V})$	$R_D(\cdot)$
20m	0	0	20m	0.8	40
2m	0	0	2m	0.5	250

dc resistance of forward-bias region decrease when higher currents and voltage.

Ideal diode			Si diode		
$I_D(A)$	$V_D(V)$	$R_D(\cdot)$	$I_D(A)$	$V_D(V)$	$R_D(\cdot)$
0	-10	\cdot	-2 \cdot	-10	5M

- dc resistance of reverse-bias region, its open-circuit equivalent.

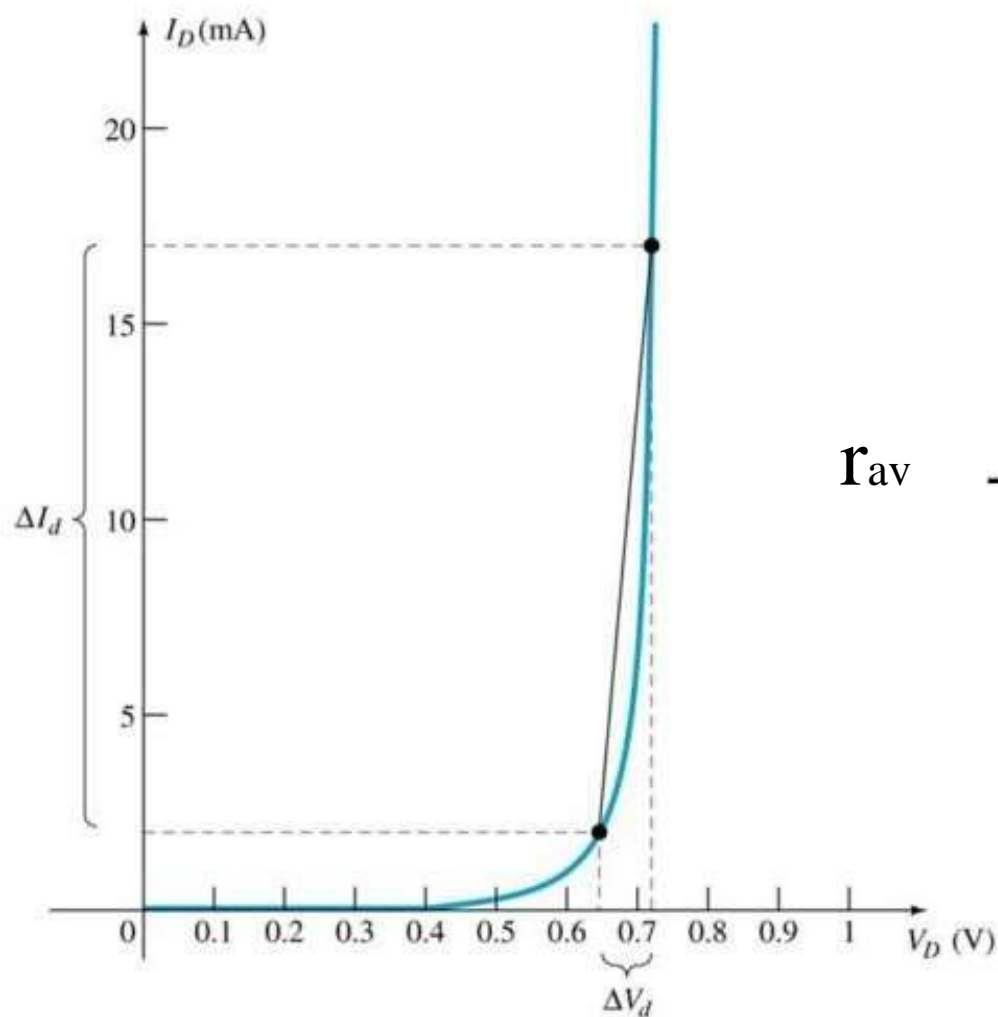
- **AC or Dynamic Resistance**

- Static resistance is using dc input. If the input is sinusoidal the scenario will be change.
- The varying input will move instantaneous operating point UP and DOWN of a region.
- Thus the specific changes in current and voltage is obtained. It can be determined using equation (1.2)

$$r_d = \Delta V_D / \Delta I_D$$

(1.2)

- Average AC Resistance



$$r_{av} = \frac{V_d}{I_d} \text{ (point to point)}$$

AC resistance can be determined by picking 2 points on the characteristic curve developed for a particular circuit.

Rectifier

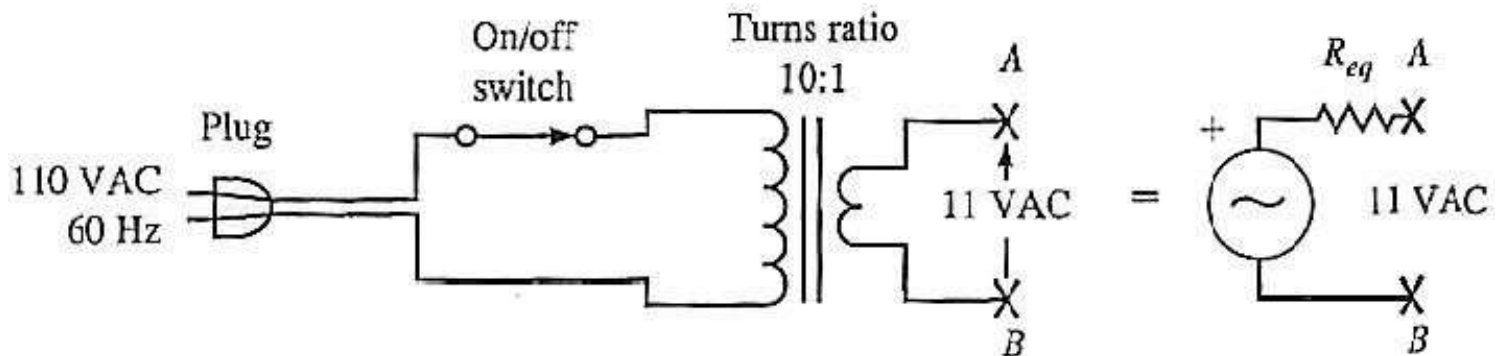
A rectifier is an electrical device that converts alternating current (AC), which periodically reverses direction, to direct current (DC), which is in only one direction, a process known as rectification.

Power Supply Circuits

- To achieve its purpose a power supply must:
 - Step down the voltage supplied;
 - Convert ac to dc by rectifying the ac.
- A transformer is used to step down the magnitude of the voltages from the wall receptacle.

Transformer

- A transformer consists of two coils of wire on a common iron core. The voltages on these two coils are related by the *turns ratio*, which is the ratio of the number of turns of wire in the secondary coil to that in the primary coil.



RMS Values

- Note that the 110-120 volts and 220-240 volts are RMS values.
- The actual amplitude of that sinusoidal signal is a factor of $\sqrt{2}$ larger.

Types of Rectifiers



Half wave
Rectifier



Full wave Rectifier



Bridge Rectifier

Half wave Rectifier

- In half wave rectification, either the positive or negative

half of the AC wave is passed, while the other

half is blocked.

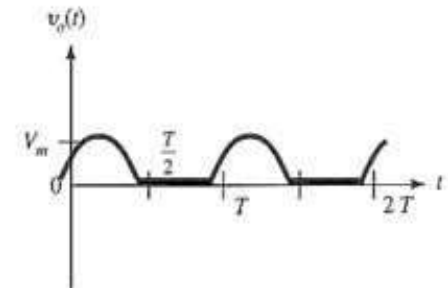
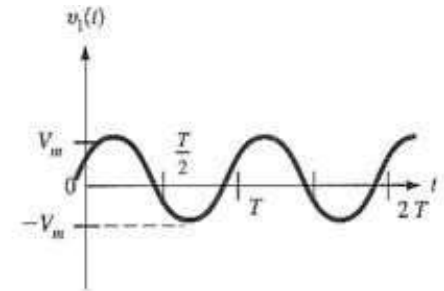
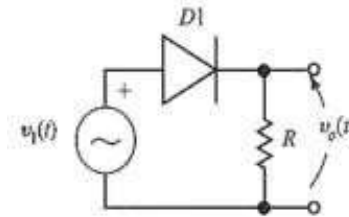
- Because only one half of the input waveform reaches

the output, it is very inefficient if used for power

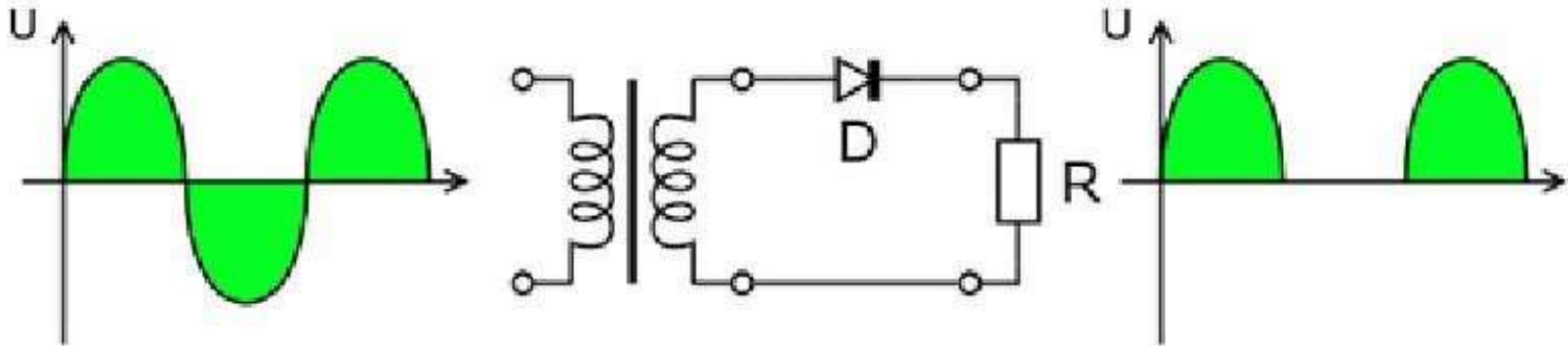
transfer.

Half-wave Rectification

- Simplest process used to convert ac to dc.
- A diode is used to clip the input signal excursions of one polarity to zero.



Half wave Rectification



Output dc voltage calculation

- The output DC voltage of a half wave rectifier can be

calculated with the following two ideal equations


$$V_{dc} = \frac{V_{peak}}{2}$$

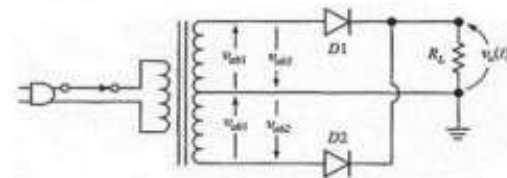
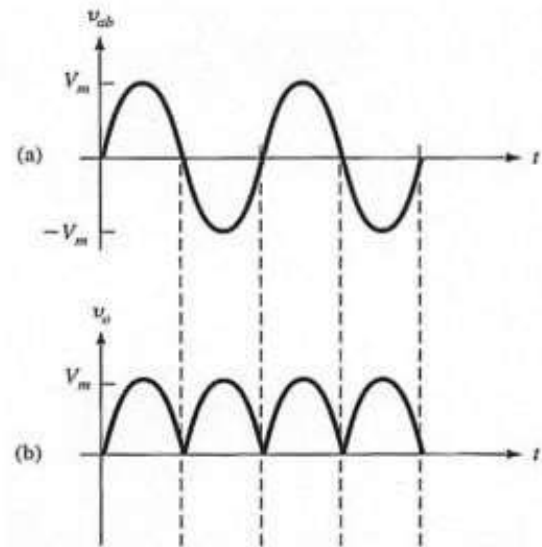
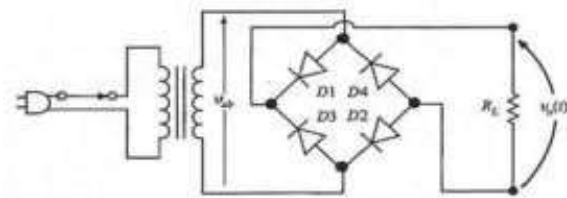
$$V_{dc} = \frac{V_{peak}}{\pi}$$

Full wave rectifier

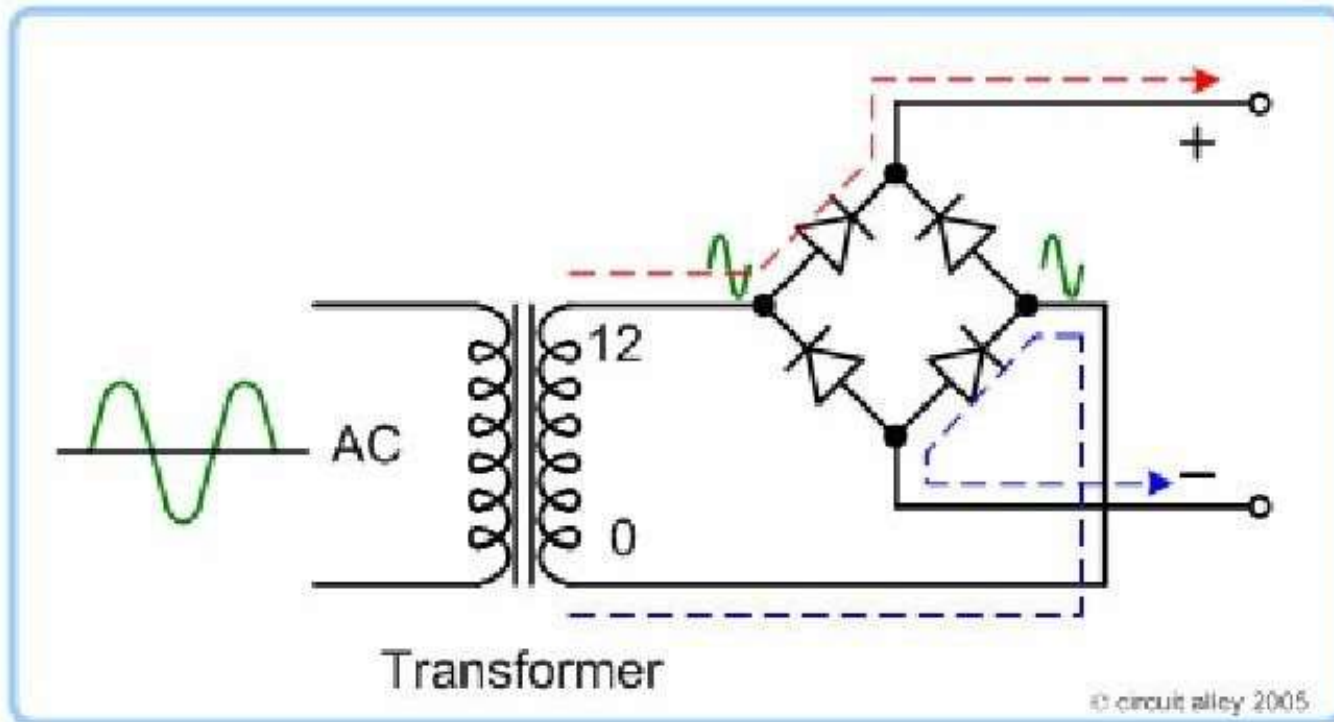
- A full-wave rectifier converts the whole of the input waveform to one of constant polarity (positive or negative) at its output. Full-wave rectification converts both polarities of the input waveform to DC (direct current), and is more efficient.0

Full-wave Rectification

- The output of a full-wave rectifier is driven by both the positive and negative cycles of the sinusoidal input, unlike the half-wave rectifier which uses only one cycle.



Full wave rectifier working animation



Full wave rectification

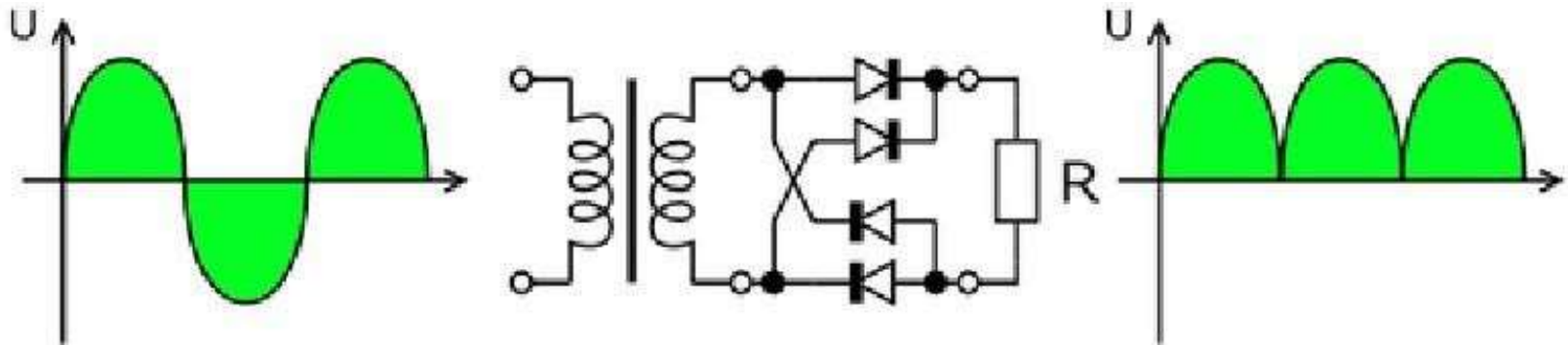
In a circuit with a non - center tapped transformer, four

- diodes are required instead of the one needed for half- wave rectification.

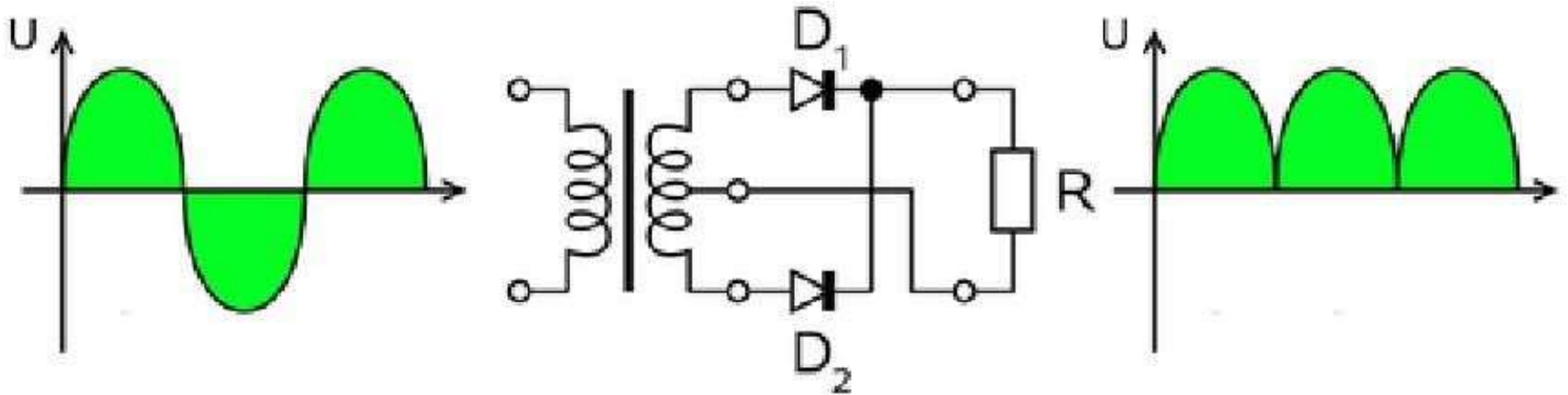
- For single-phase AC, if the transformer is center-tapped,

then two diodes back-to-back(i.e. anodes-to-anode or cathode-to-cathode) can form a full-wave rectifier.

Full wave rectifier using 4 diodes



Full wave rectifier using transformer and 2 diodes

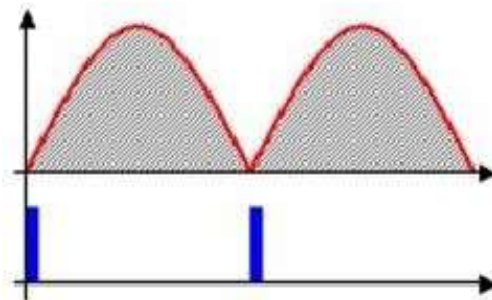


ideal single phase full wave rectifier can be calculated as:
The average and root-mean-square output voltages of an

$$V_{dc} = V_{av} = \frac{2V_p}{\pi}$$



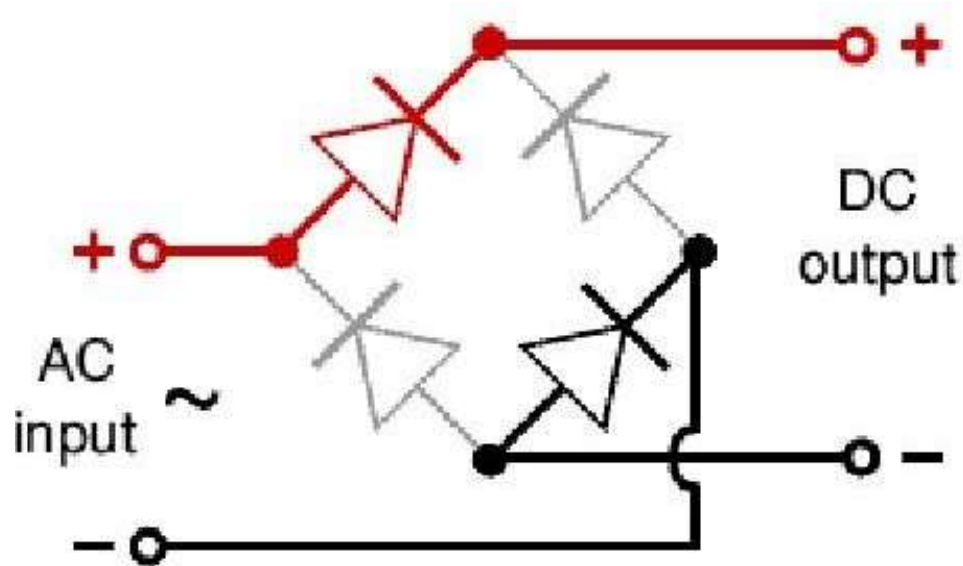
Output voltage of the full waverectifier Animation



Bridge rectifier

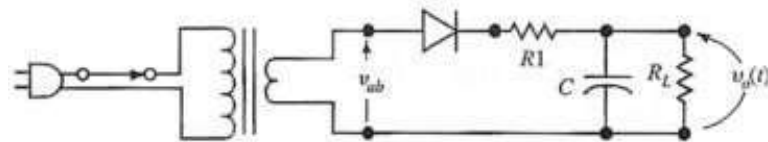
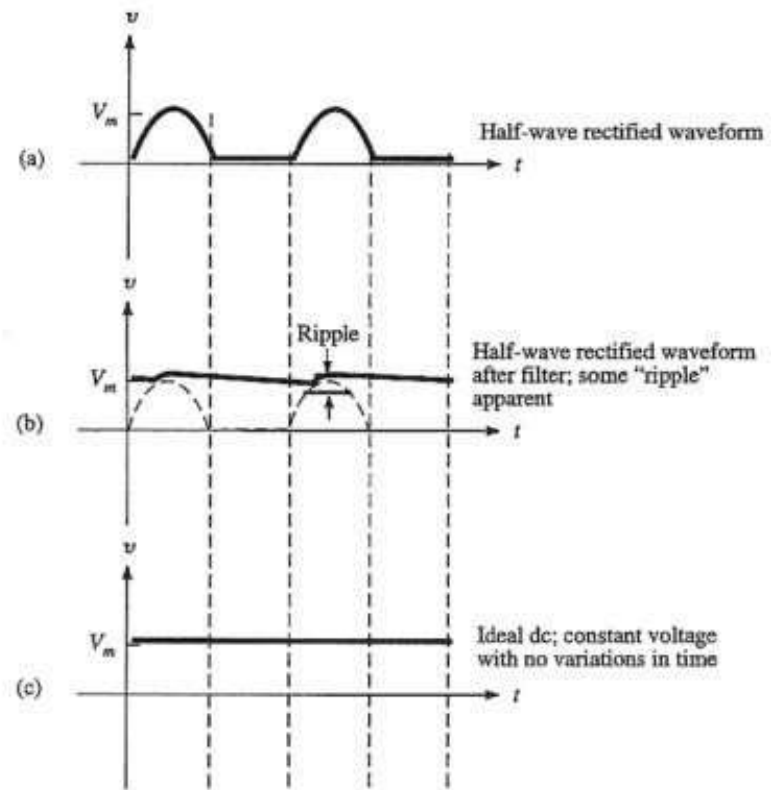
- A bridge rectifier makes use of four diodes in a bridge arrangement to achieve full-wave rectification.

Bridge rectifier working animation

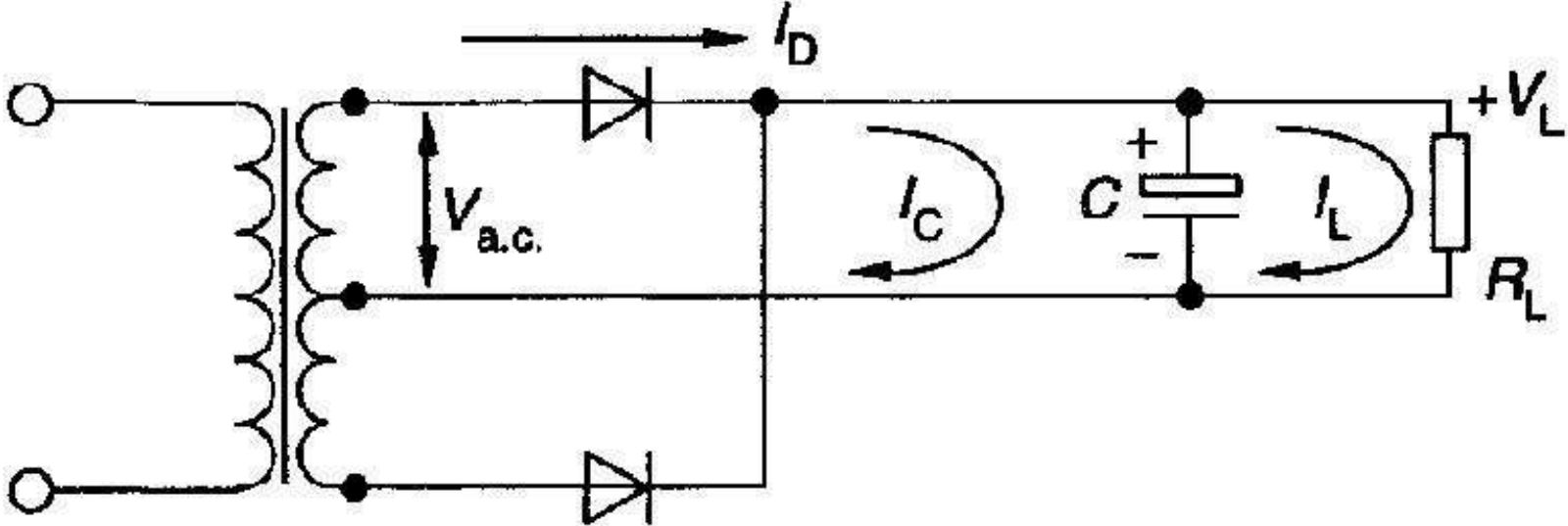


Filtering

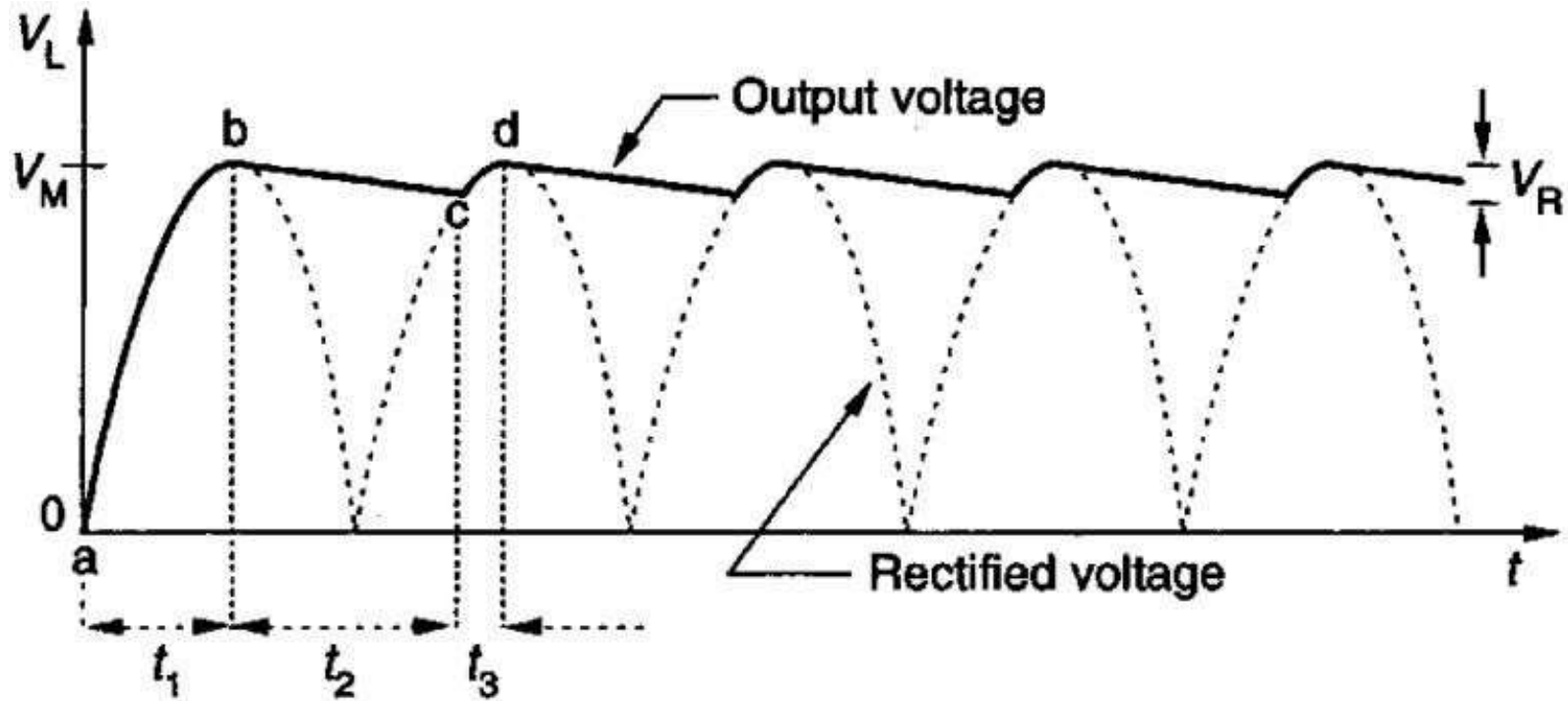
- Process used to smooth out the output of the rectifier circuit.
- One of the most common filter is the RC network.



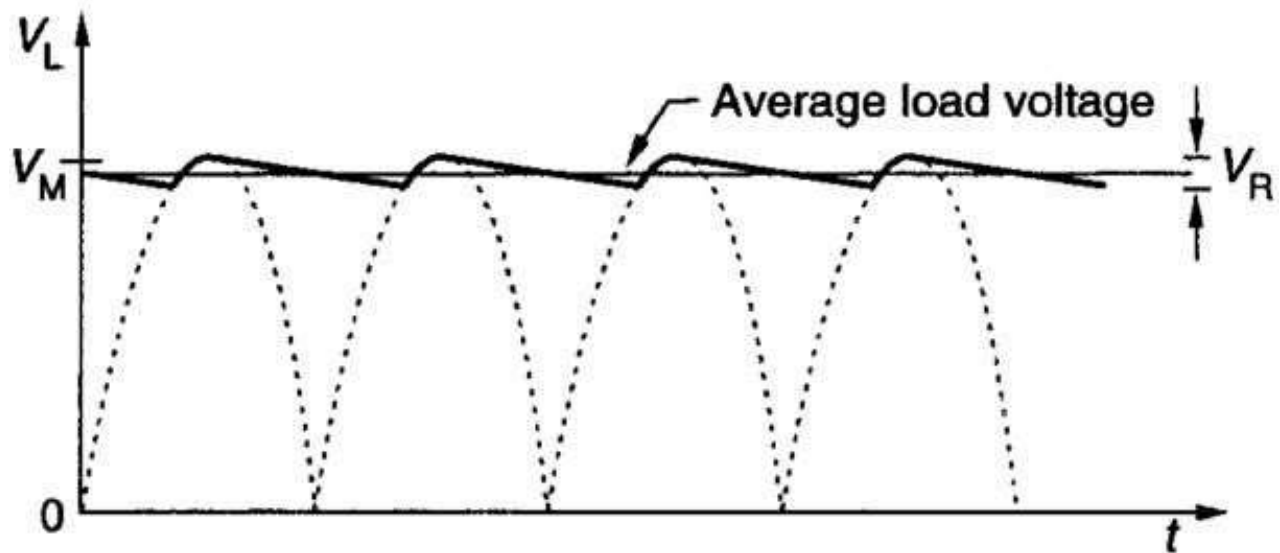
Full-wave centre-tap rectifier and capacitor filter with load connected



Capacitor filter output waveforms with load connected

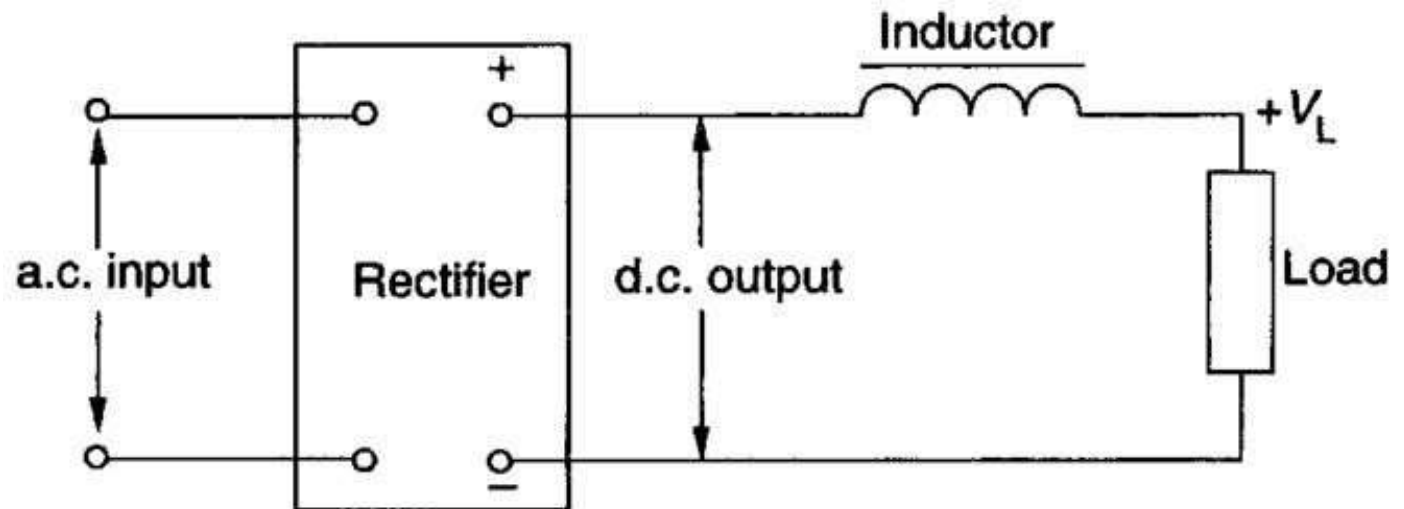


Effect of load on the output of a capacitor filter circuit

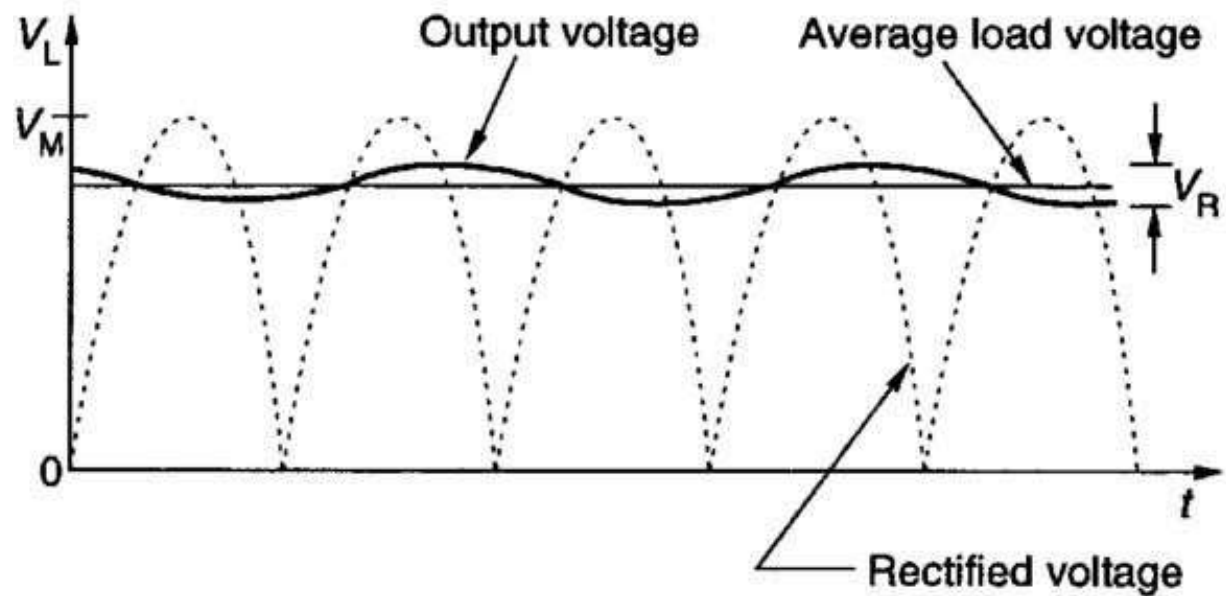


(a) Capacitor filter on light load

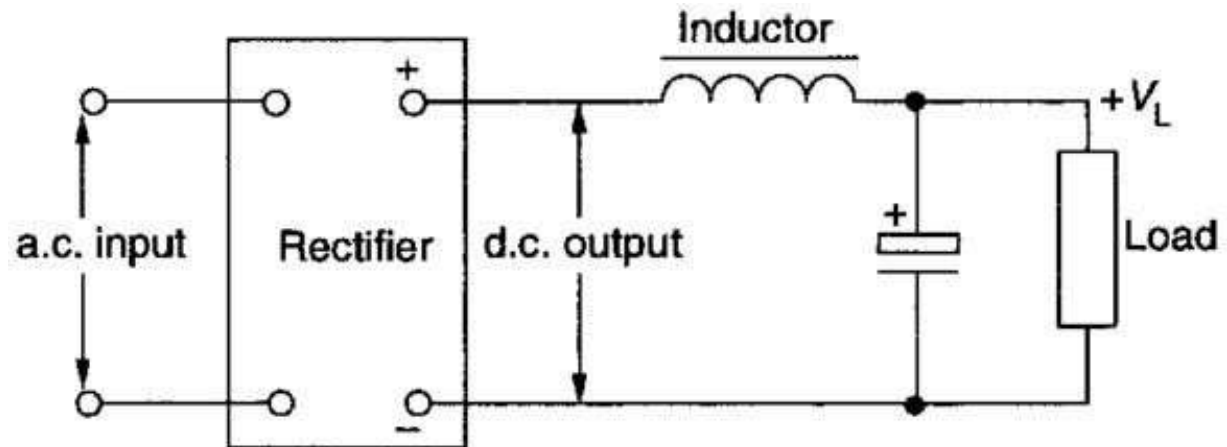
Inductor filter



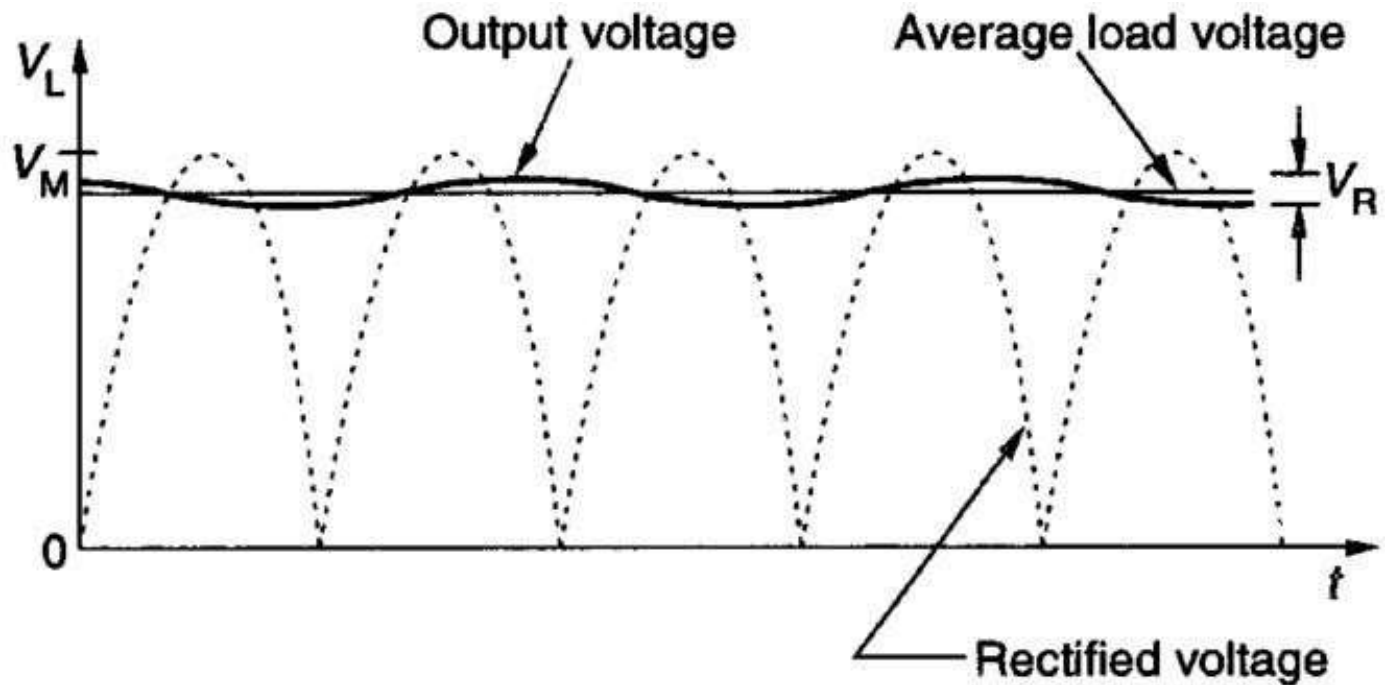
Inductor filter output voltage waveforms



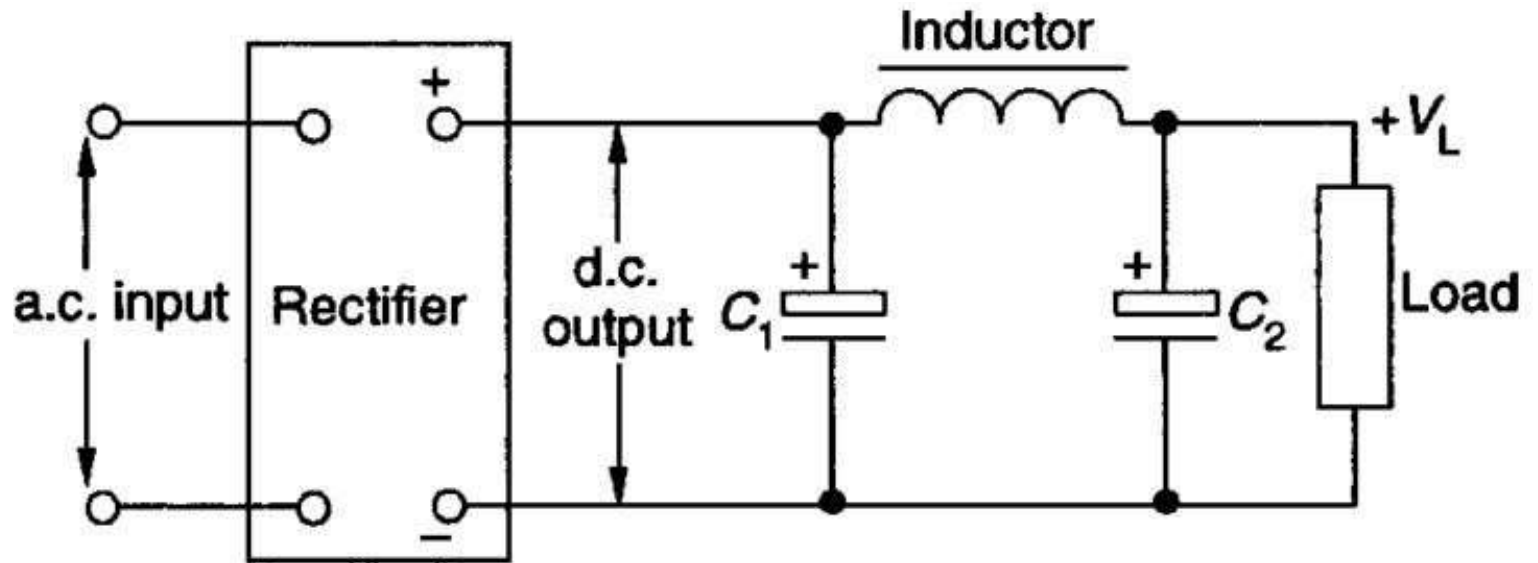
Choke input filter (L-section filter)



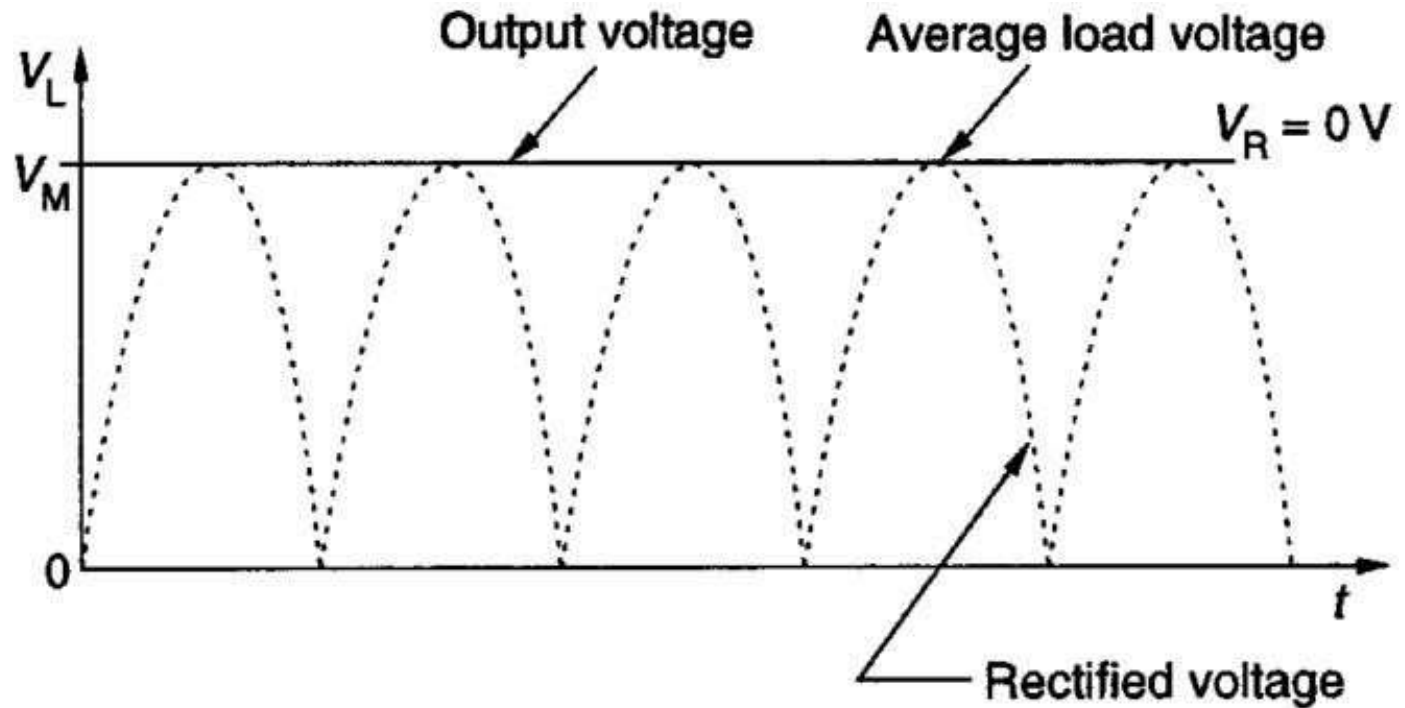
Choke input filter load voltage waveforms



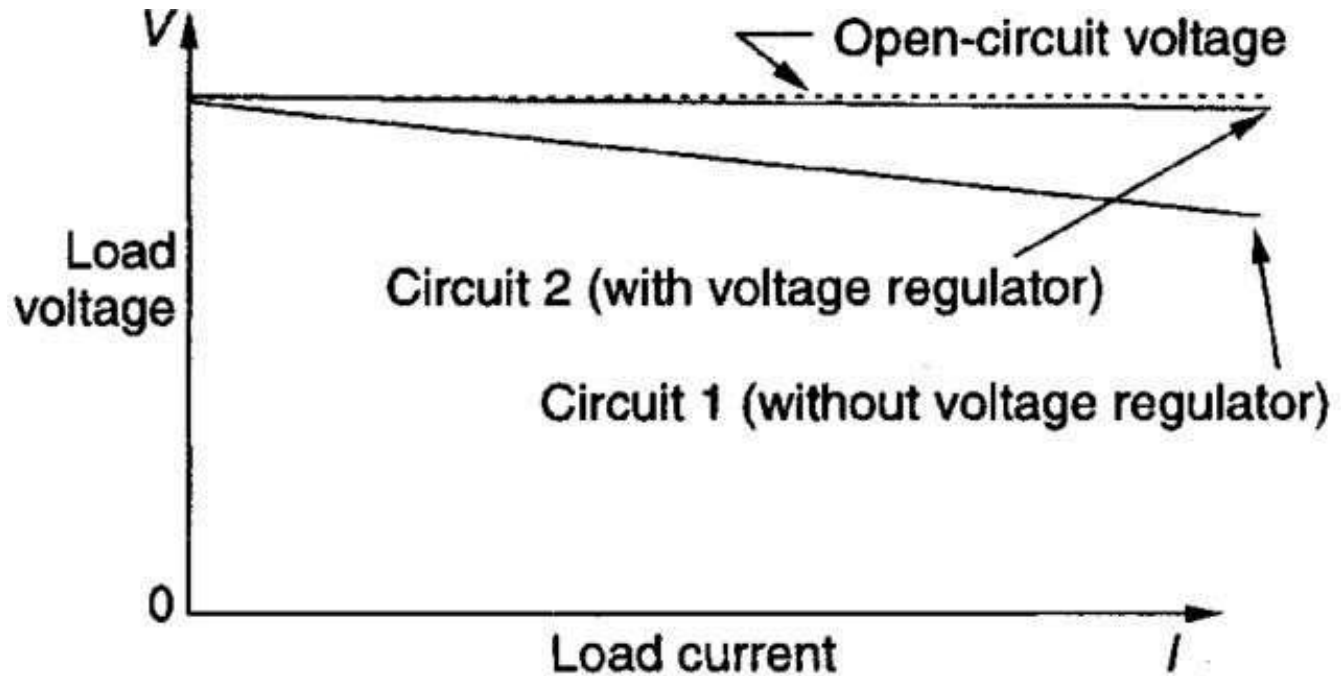
π -type filter



π -type filter output voltage waveforms

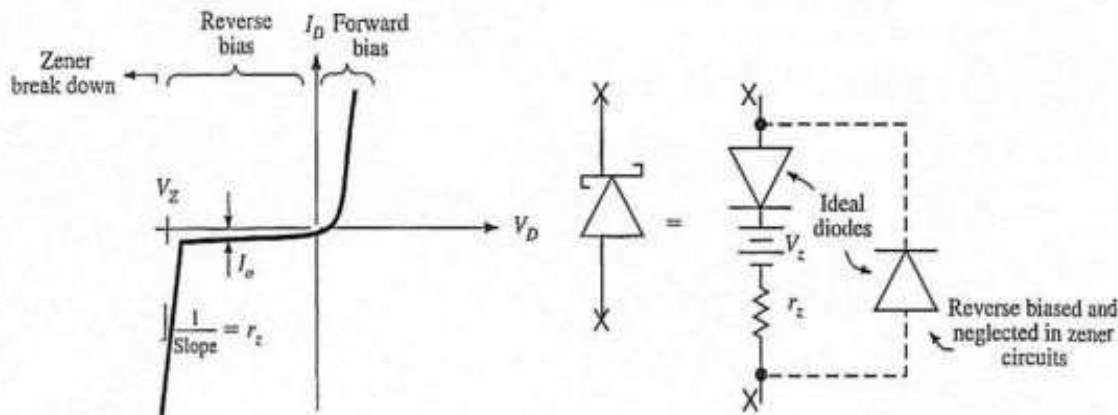


Load characteristics for circuits with and without voltage regulators

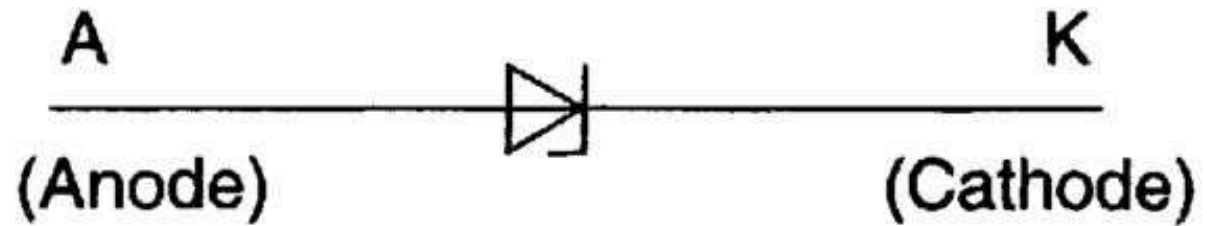


Zener Diode

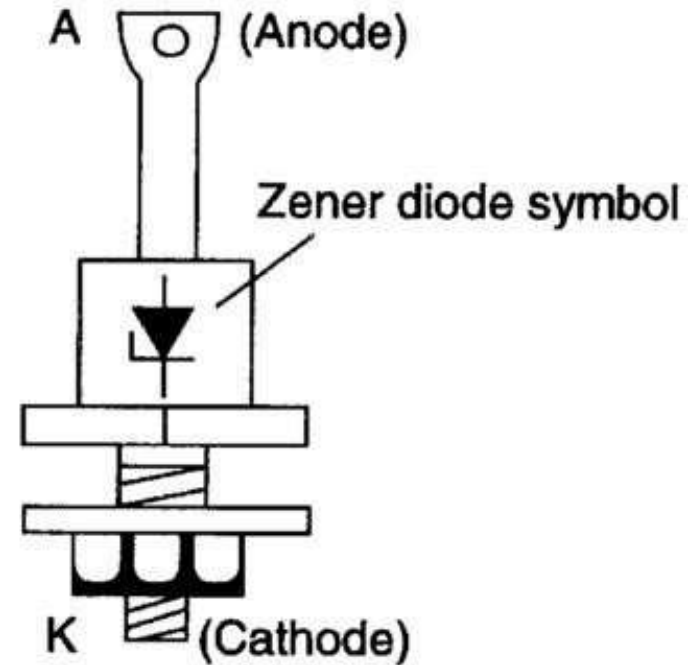
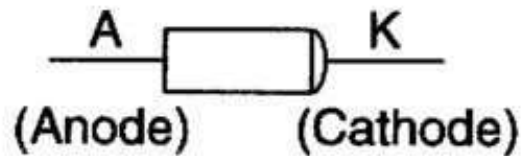
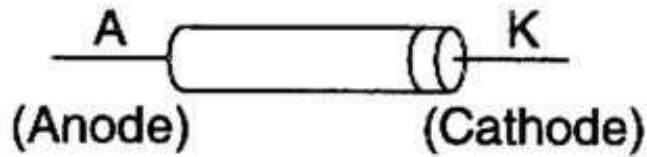
- Analyzing a diode operating in the reverse bias region will show that the current through it remains essentially constant until the *breakdown voltage*, also called the *avalanche* or *zener breakdown voltage*, is reached. At this point the current will increase very rapidly for a small voltage change.



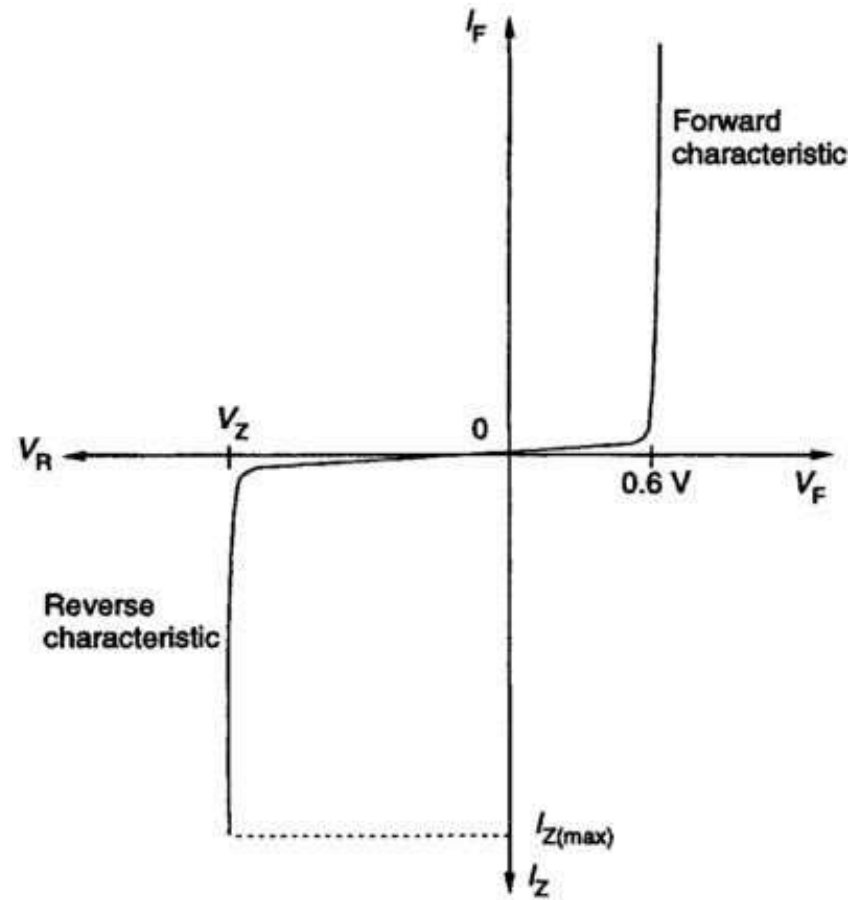
Zener diode symbol



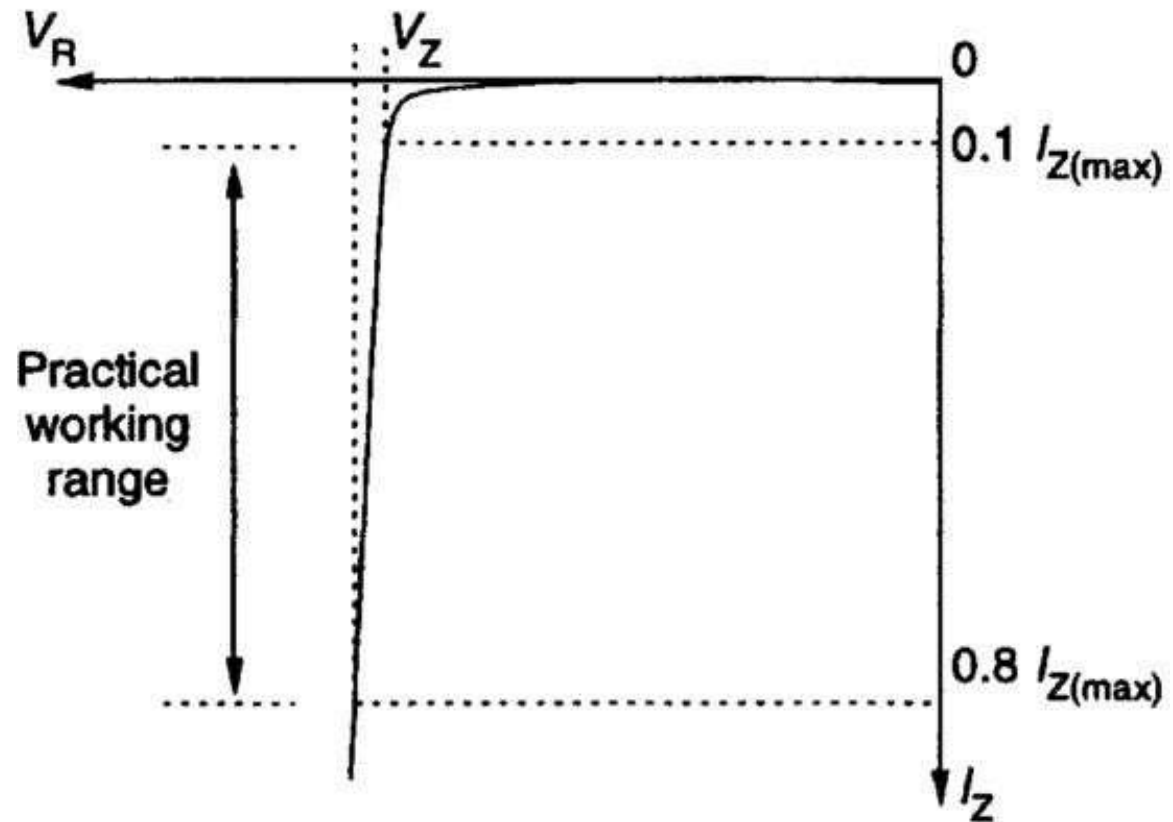
Zener diode terminal identification



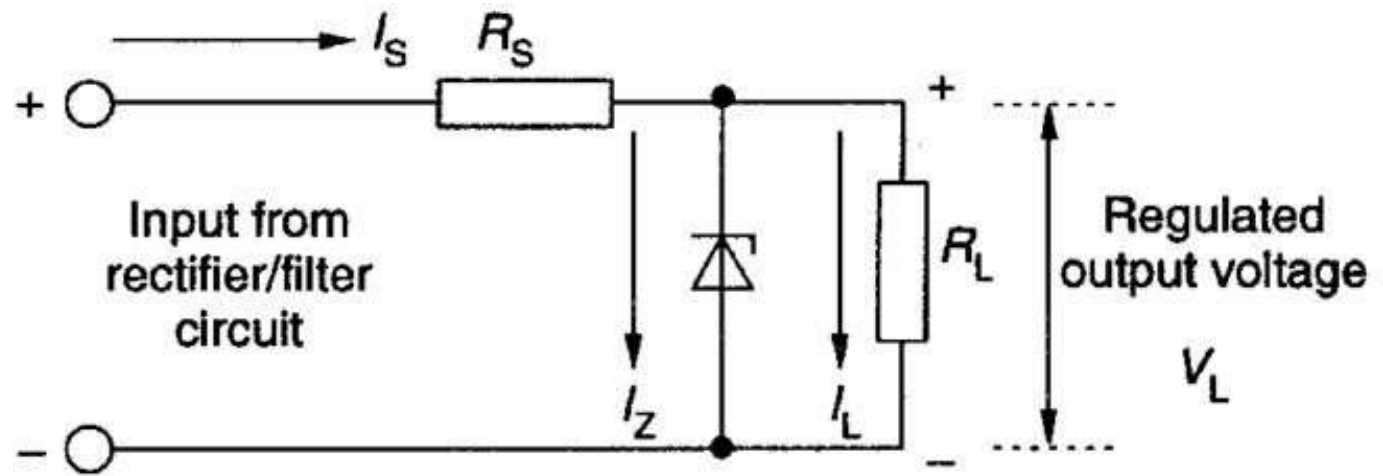
Zener diode V-I characteristics



Zener diode reverse-characteristic working range of currents



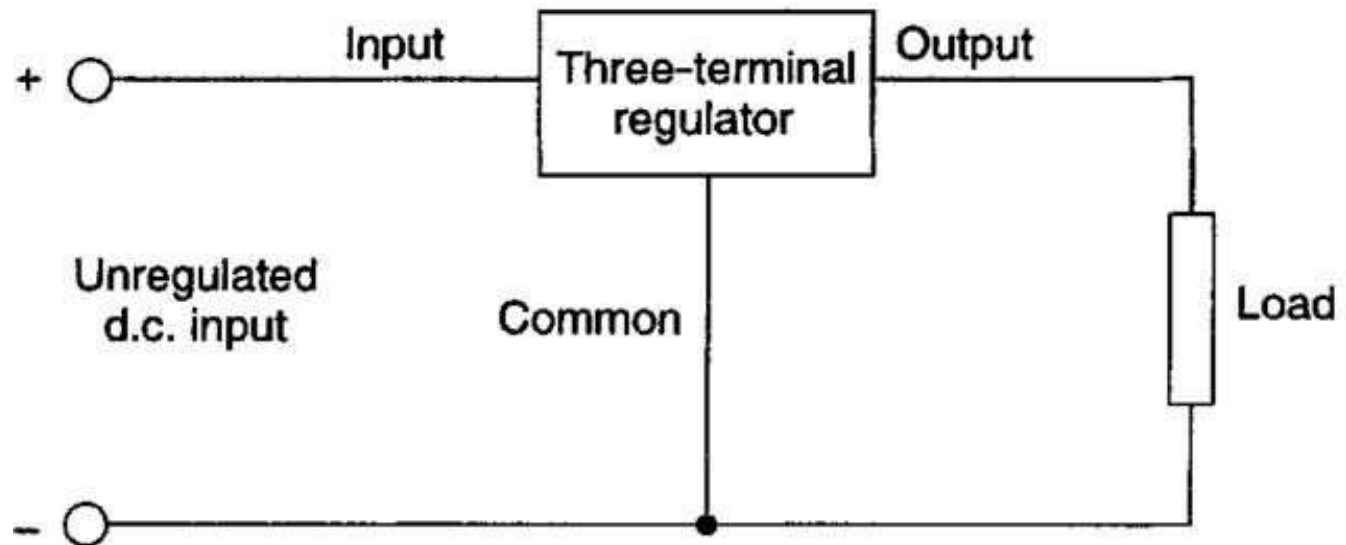
Zener diode shunt regulator



Voltage Regulation

- This characteristic of the zener diode is very useful for voltage regulation circuits. The zener diode provides an effective way to clamp or limit the voltage at a relatively constant value thus creating a voltage regulation capability.

Three-terminal integrated circuit regulator—basic circuit



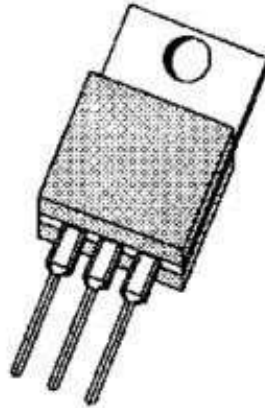
Three-terminal regulator case styles



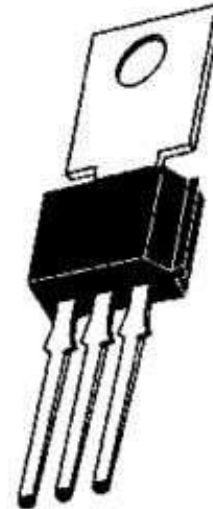
(a) TO-92



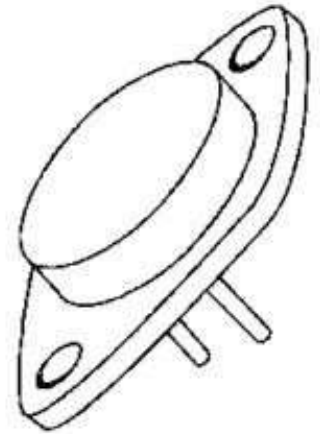
(b) TO-39



(c) TO-202



(d) TO-220



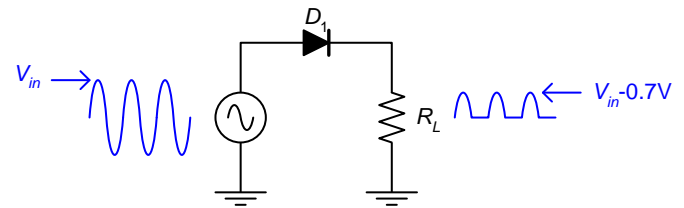
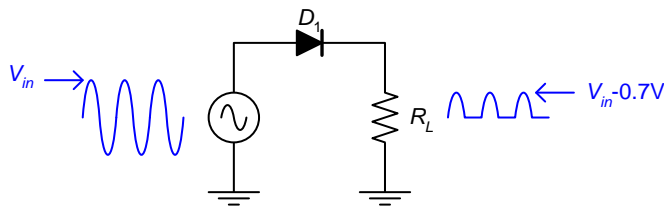
(e) TO-3

Three-terminal regulator connections— positive and negative regulators



Clipper (limiter) circuits. (1)

Clipper (limiter) circuits. (1)



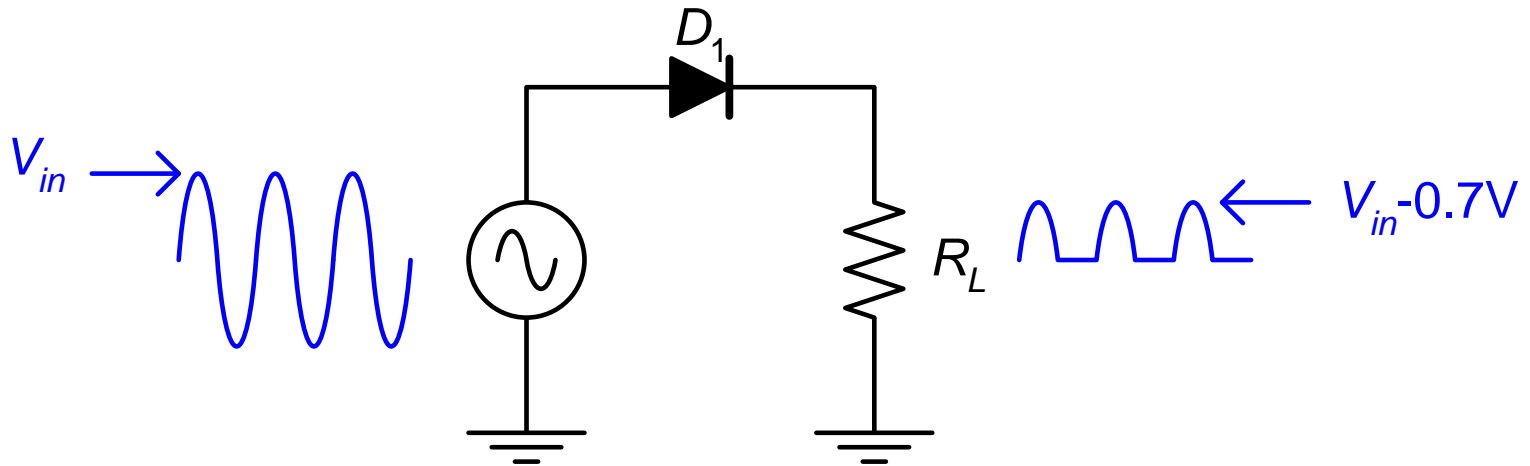
Negative series clipper.

Negative series clipper.



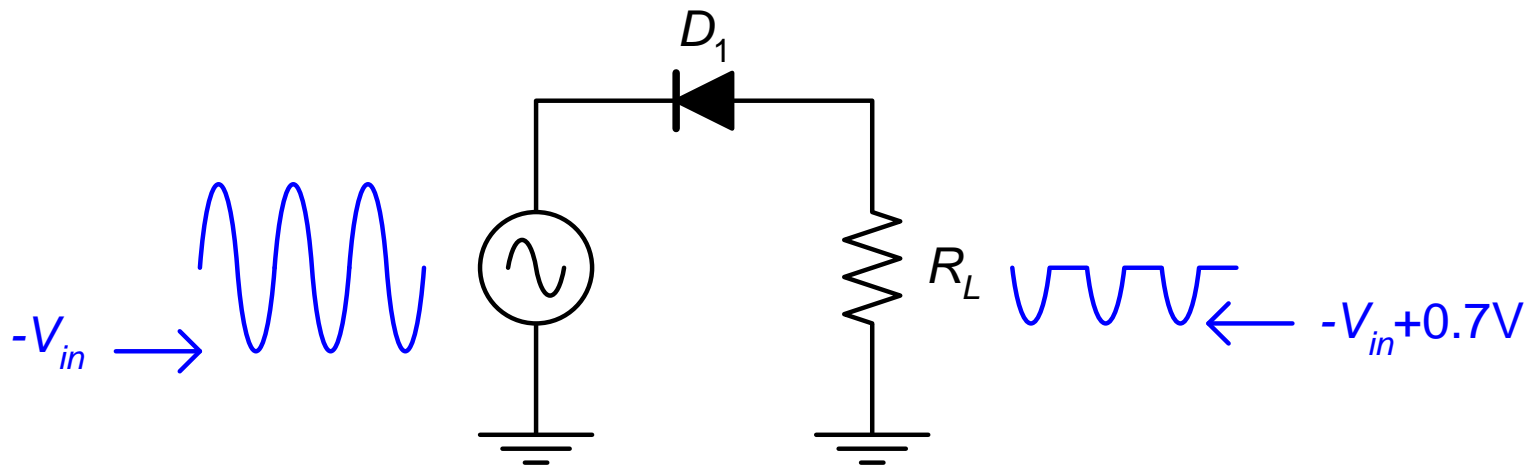
Negative voltage regulator

Clipper (limiter) circuits. (1)



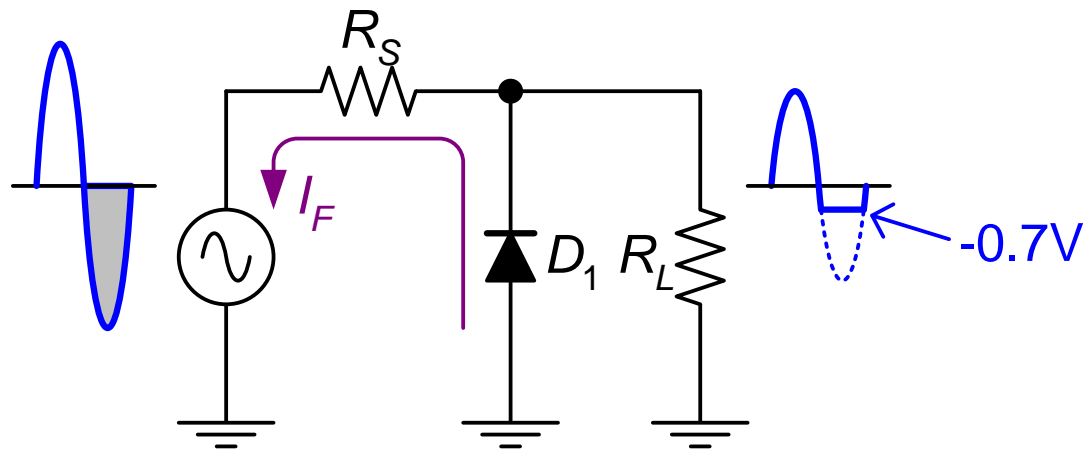
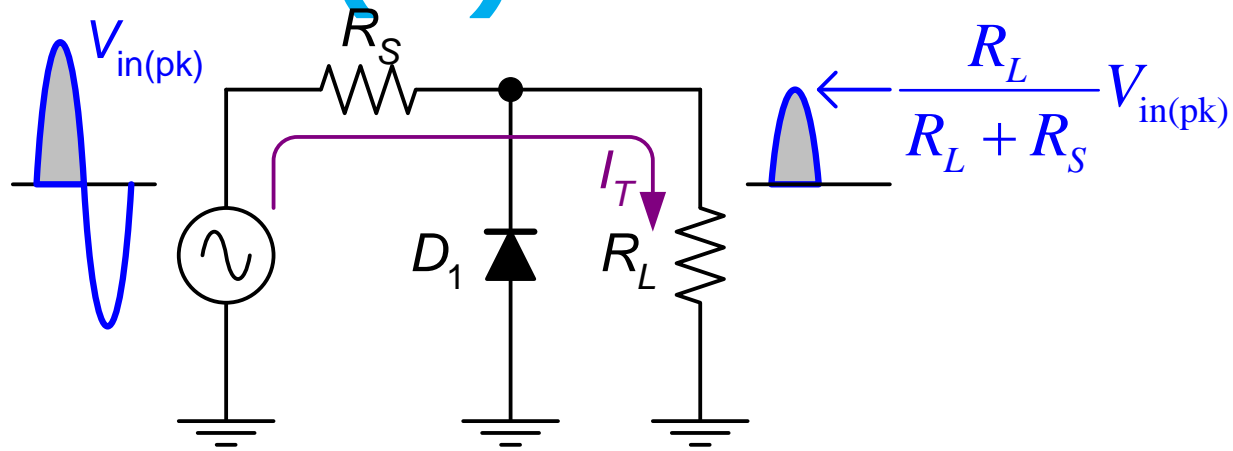
Negative series clipper.

Fig 4.1-11 Clipper circuits. (2)



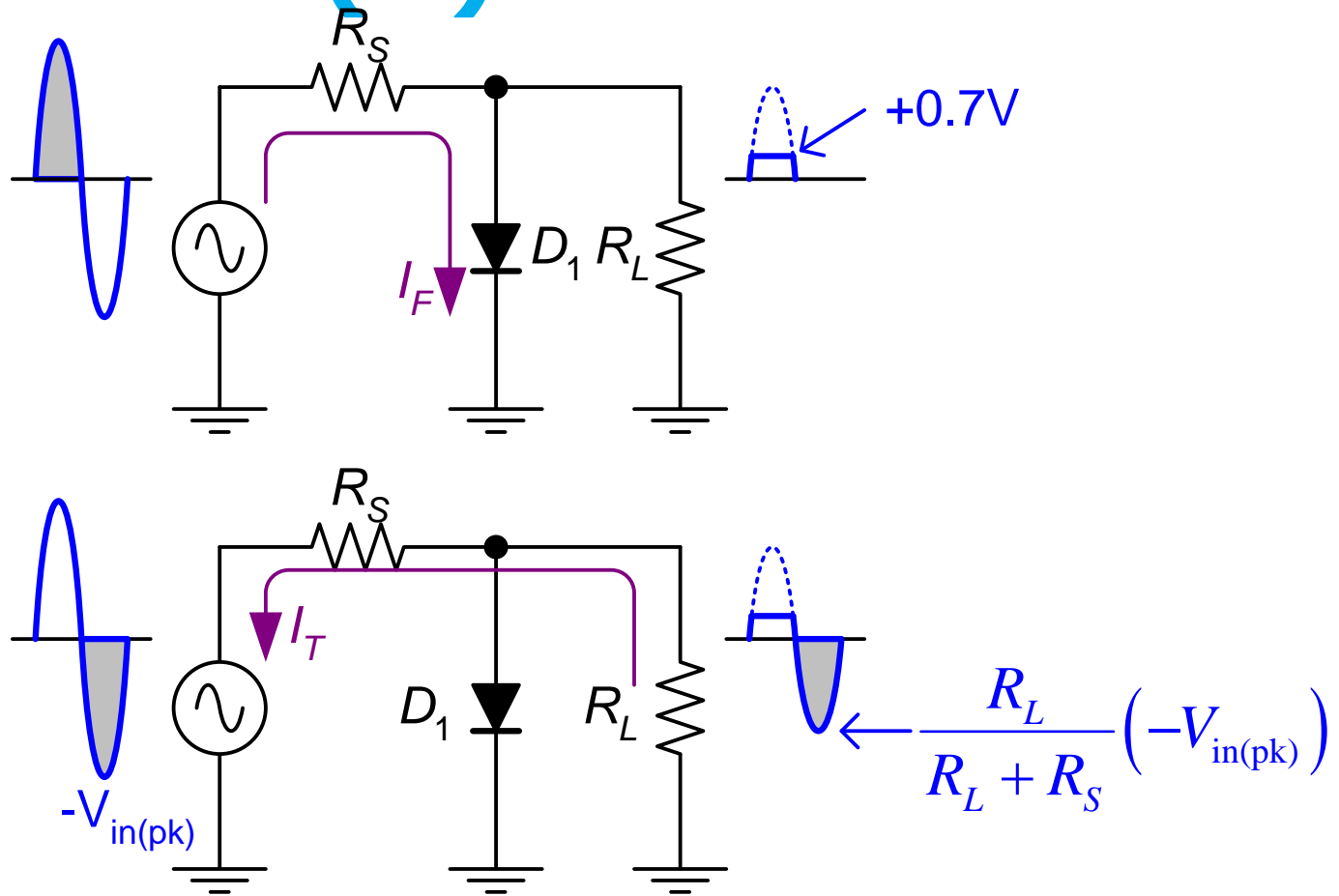
Positive series clipper.

Fig 4.1-11 Clipper circuits. (3)



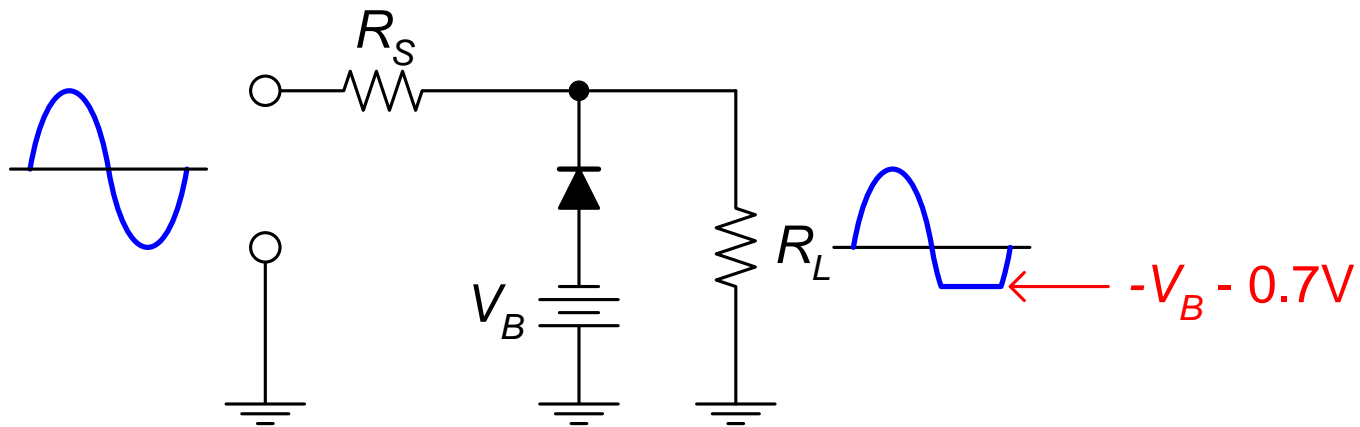
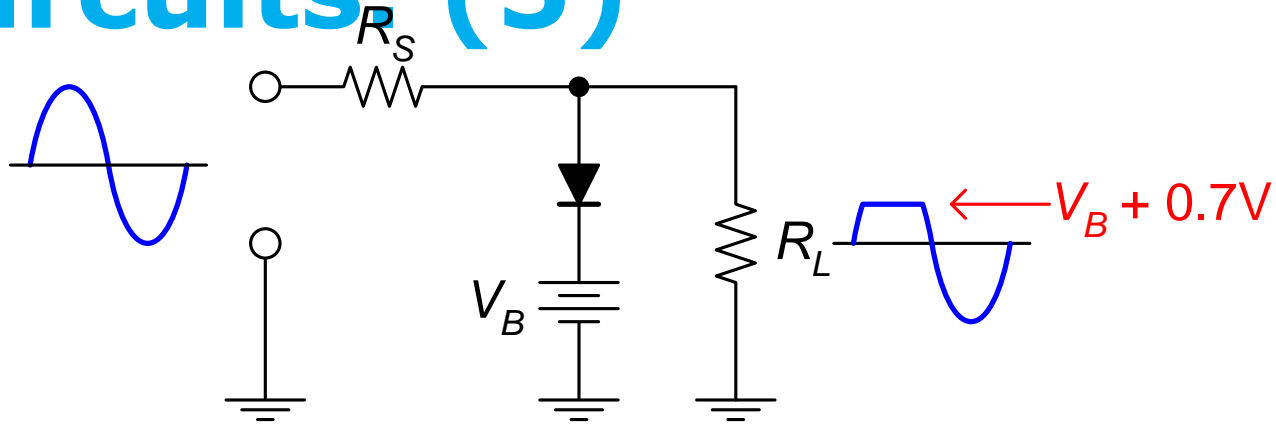
Negative shunt clipper.

Fig 4.1-11 Clipper circuits. (4)



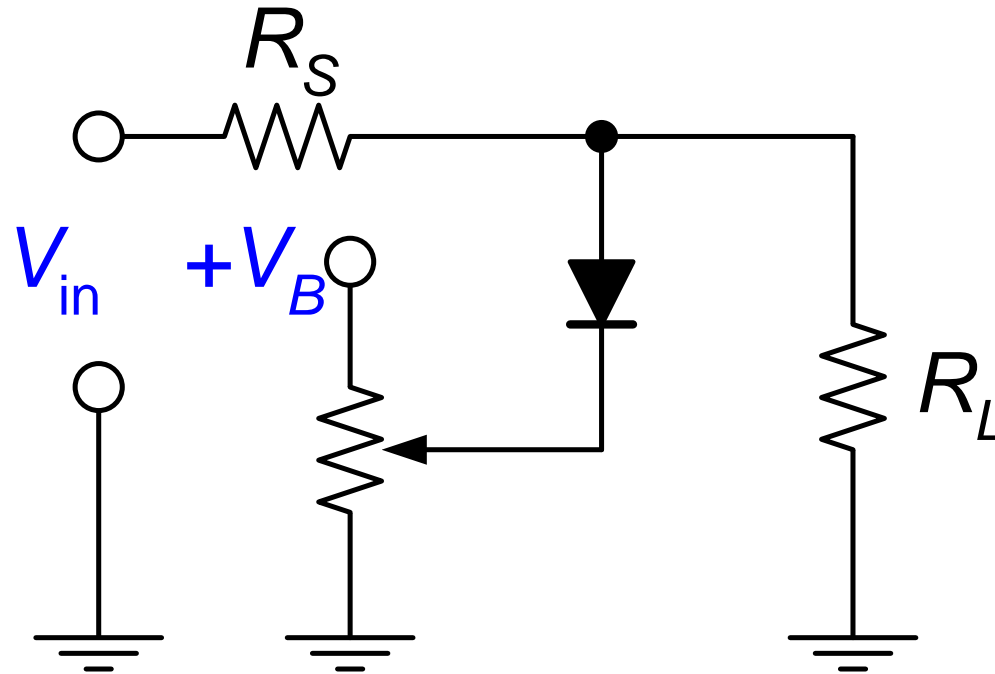
Positive shunt clipper.

Fig 4.1-11 Clipper circuits (5)



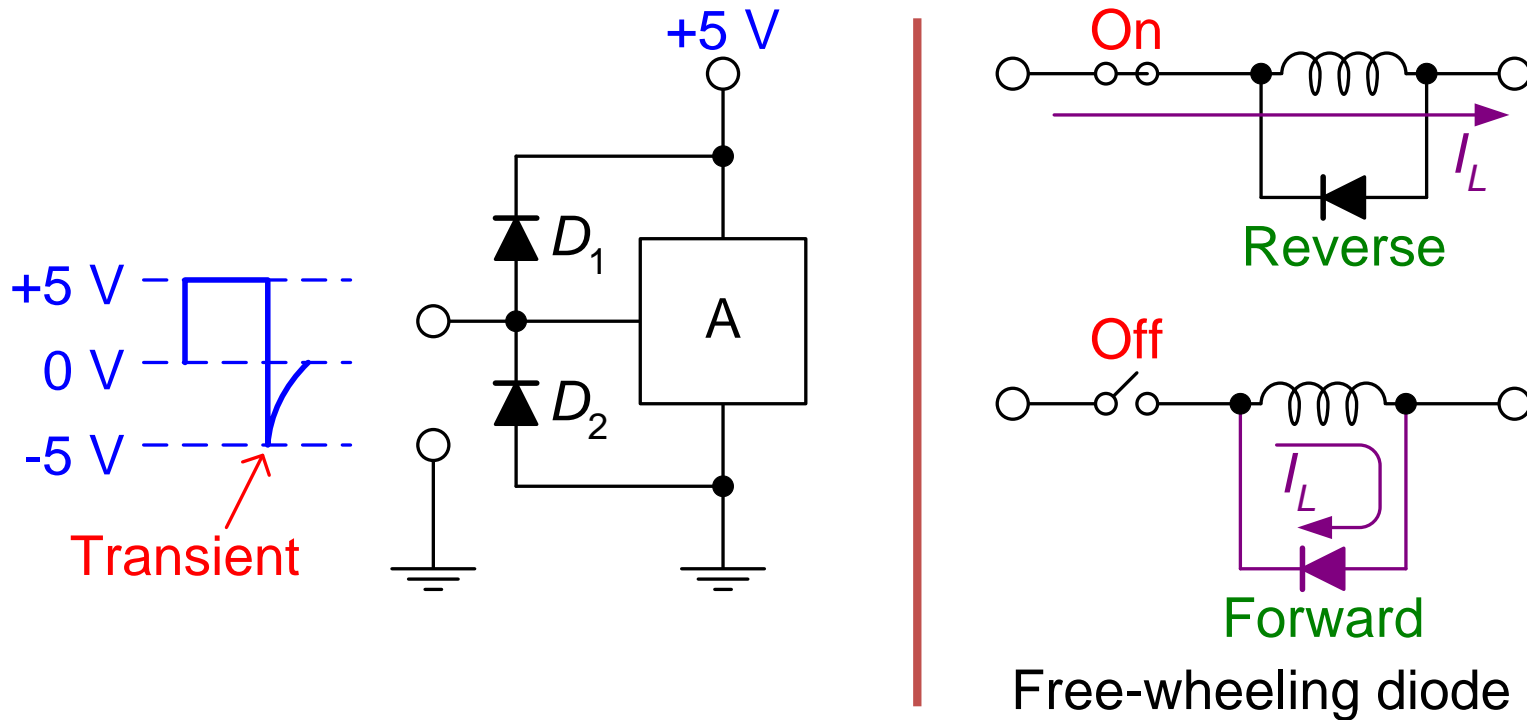
Biased shunt clipper.

Fig 4.1-11 Clipper circuits. (6)



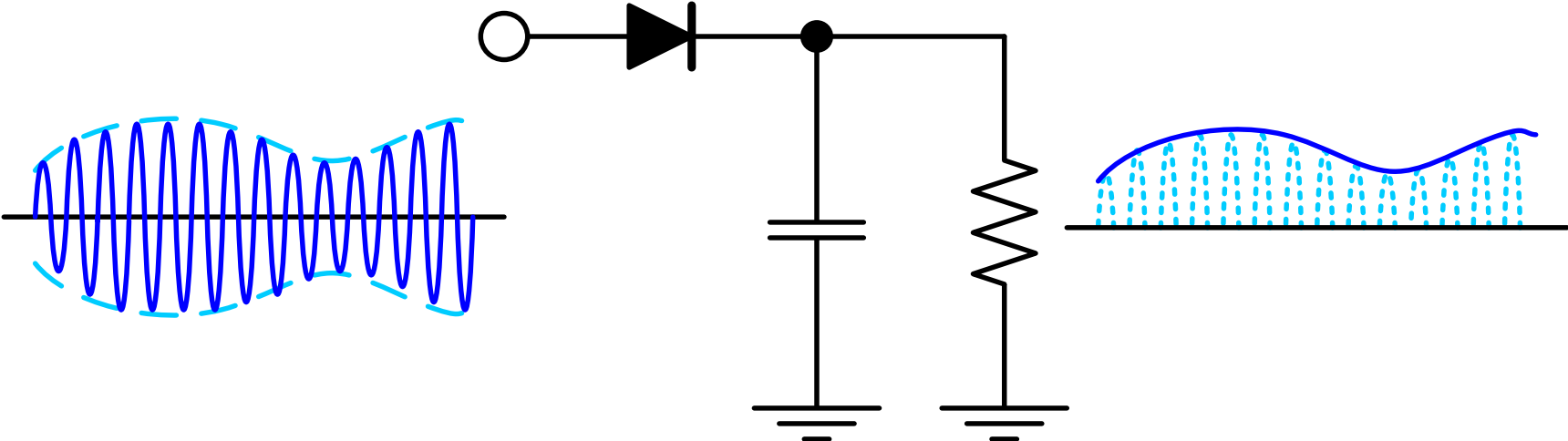
Adjustable shunt clipper.

Fig 4.12-13 Clipper applications. (1)



Transient-protection circuits

Fig 4.12-13 Clipper applications. (2)



AM detector

ANALOG AND DIGITAL ELECTRONICS.

UNIT – II BIPOLAR JUNCTION TRANSISTORS

Introduction:



Transistor: 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?

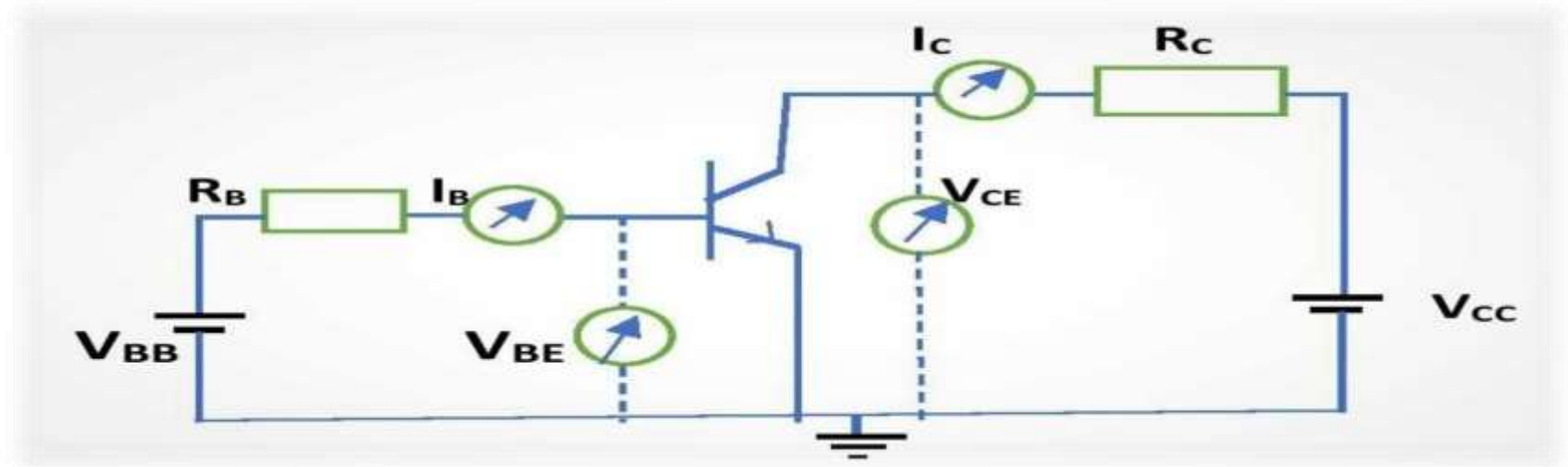
Suppose that you have a FM receiver which grabs the signal you want. The received signal will obviously be weak due to the disturbances it would face during its journey. Now if this signal is read as it is, you cannot get a fair output. Hence we need to amplify the signal. Amplification means increasing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

Transistor characteristics:

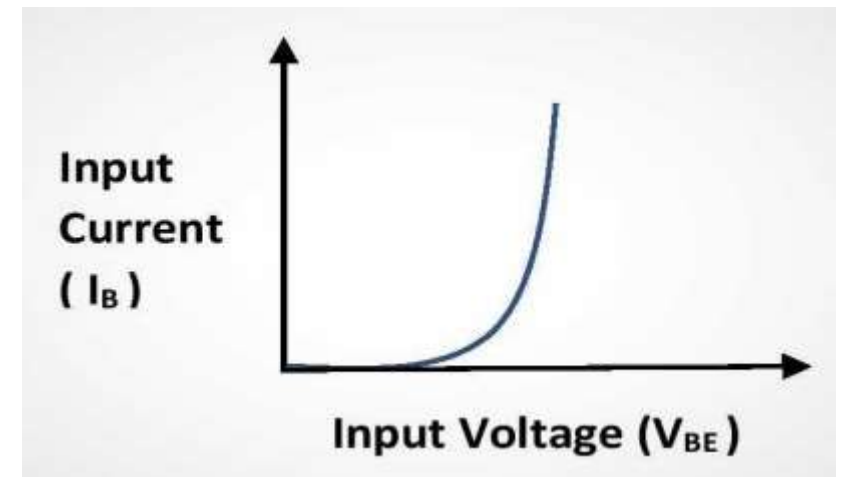
- One can draw the characteristics curves of a Transistor by using the following **Transistor Amplifier circuit**. Here, a NPN transistor is used in common emitter mode.
- V_{BE} is the input voltage, V_{CE} is the output voltage, I_B is input current and I_C is the output current.
- To measure input and output voltages and currents, two voltmeters and two ammeters are used in this circuit.

Transistor Amplifier Circuit



Input Characteristics of Transistor:

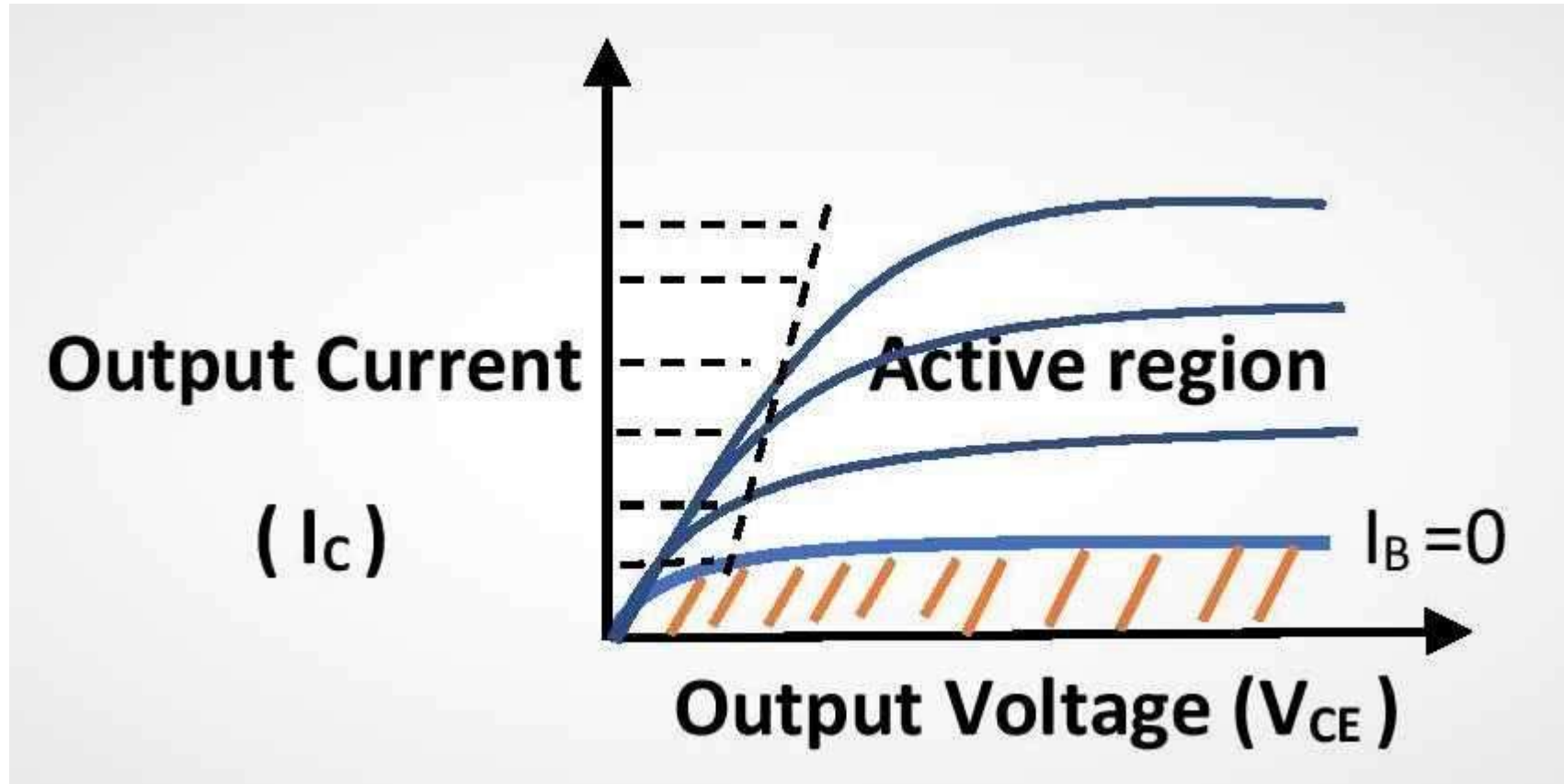
- Input characteristics curve indicates the variation of input current with the input voltage. So, here we will analyze the variation of base current (I_B) with the base-to-emitter voltage (V_{BE}).
- To draw I_B vs V_{BE} curve we need to ground the output terminal. That means we need to connect collector terminal to emitter terminal as the emitter is already grounded in CE mode. At this condition transistor behaves like a **p-n junction diode**.
- There is an exponential growth of base current with increase in V_{BE} .
- So, the I_B vs V_{BE} curve will be similar to diode characteristics curve. Fig.3 gives the Input characteristics curve of a Transistor in CE mode.



Output Characteristics of Transistor:

- Output Characteristics curve of a Transistor gives the variation of output current with the change in output voltage.
- In a common emitter transistor collector current is output current and collector-to emitter voltage is the output voltage.
- In the graph one can see that there is a rapid increase of collector current at the beginning and then the collector current becomes almost constant. If we increase V_{CE} further, the breakdown occurs and then the transistor may be damaged. One can divide the output curve into four regions –
 - Active Region (the region in which output current becomes almost constant)
 - Saturation Region (horizontal dotted lines (---) in output curve indicates transistor saturation region)
 - Cut off region (///// lines in output curve indicates the cut off region)
 - Inverted region (appears after breakdown which is not shown in the output curve)

Output Characteristics of Transistor:



fig(4): Output characteristics curve of Transistor

Active Region of Transistor



- The region on output curve of a transistor where the output current is almost constant and independent on output voltage is the Active region of Transistor.
- If the base resistance be greater than the maximum allowed value then the transistor operates in Active region.
- One can use **Transistor as an Amplifier** only if it operates in active region.
- Also for the operation in active region, emitter junction should be in forward bias and the collector junction should be in reverse bias.

Saturation region of Transistor

- Saturation region is the region on output curve of transistor where the collector current increases rapidly with the slight increase in output voltage.
- To operate the transistor in saturation region, the base resistance should be smaller than the maximum allowed value.
- Also, for the operation in saturation region, both of emitter junction and collector junction should be in forward bias.
- In saturation region, transistor acts like ON stage of a switch.

Cut Off region of transistor

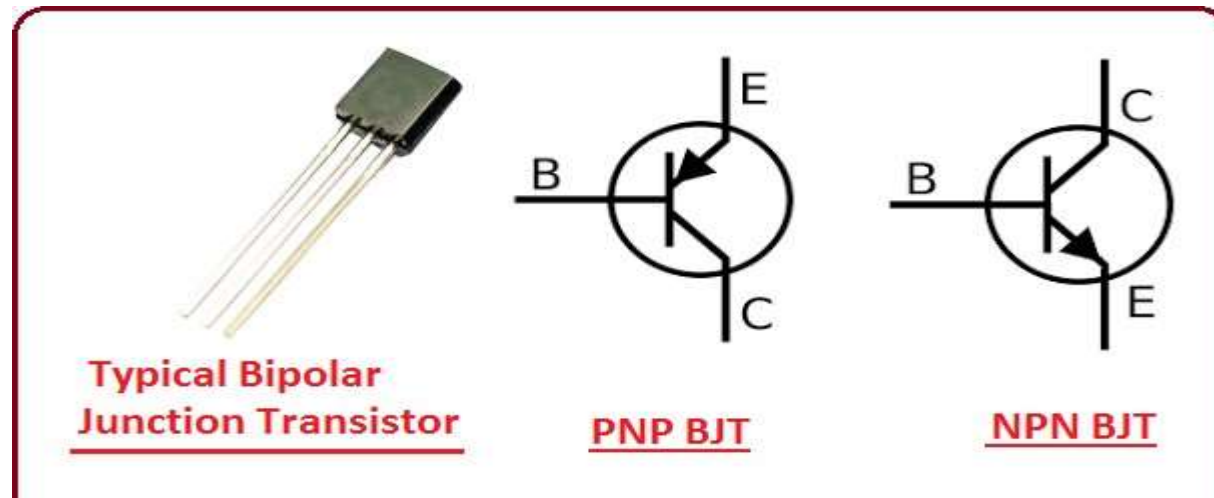
- In the Cut Off region the base current is almost zero. Therefore, collector current also becomes zero even at higher output voltage.
- To operate a transistor in the cut off region, both of emitter junction and collector junction should be in the reverse bias condition.
- In the cut off region a transistor acts like the OFF stage of a switch.

Inverted region of transistor

- This is the inverse of active region. A transistor will operate in inverted region if its emitter junction is in reverse bias and the collector junction is in forward bias.
- In this region, breakdown occurs and collector current increases rapidly. There is no major practical use of transistor in inverted region.
- Therefore, transistor operation in this region is rarely used. This region is not shown in the output diagram.

Bipolar Junction Transistor:

- A bipolar junction transistor 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 analogous 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.



Working of Junction Transistor

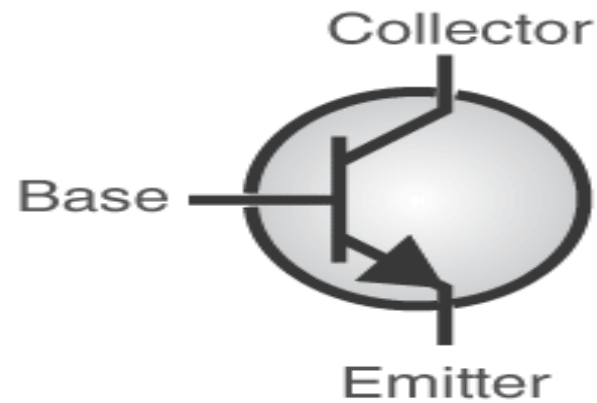
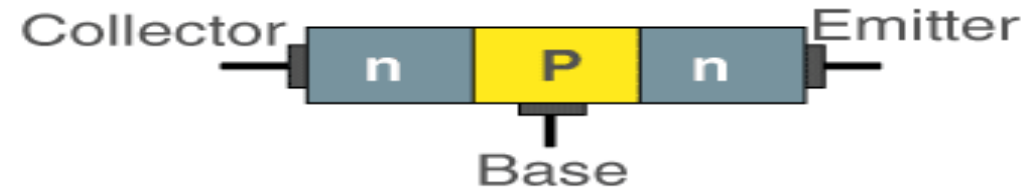
- In comparison to the emitter and collector, the base of a transistor is made thin and is lightly doped which means that the density of the majority carriers in the base is less than the density of majority carriers in the emitter and collector.
- The flow of current is supplied through the emitter and the collector collects them. For a proper connection between the emitter and the collector, the base provides the interaction for the connectivity.
- In a junction transistor, the arrow points towards the conventional current. In n-p-n transistor's emitter, the arrow points away from the base, and in a p-n-p transistor's emitter, the arrow points towards the base.
- The base-emitter junction is usually forward biased and the base-collector junction is reversed biased when the junction transistor is used in a circuit.

NPN transistor

- In an NPN transistor, a p-type semiconductor base is sandwiched between an n-doped emitter and n-doped collector. NPN transistors are the highest used bipolar transistors due to the ease of electron mobility over electron-hole mobility.
- The figure shows the construction and symbol of NPN transistors. The majority charge carriers in an n-p-n transistor are electrons and holes are the minority charge carriers
- . A small amount of current at the base terminal causes a large amount of current to flow from emitter to collector.
- Due to forward biasing of the transistor, the majority charge carriers in the emitter are repelled towards the base.
- The electron-hole recombination at the base is very small in the base region and most of the electrons cross into the collector region.

NPN transistor

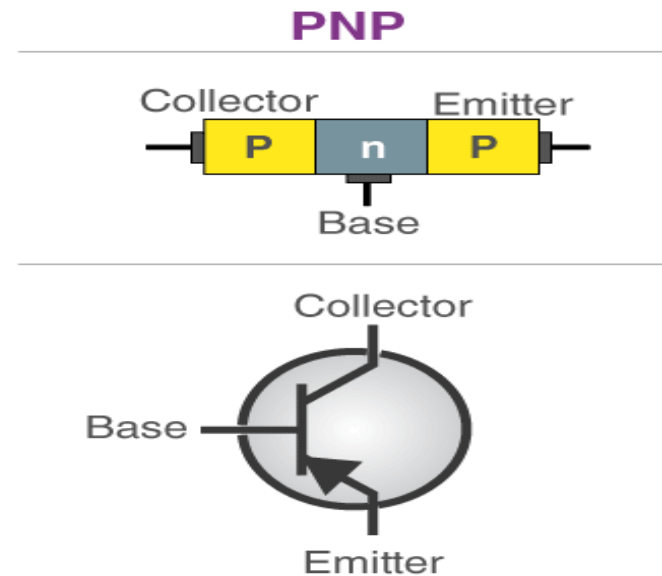
NPN



© Byjus.com

PNP transistor

- In a PNP transistor, an n-type semiconductor base is sandwiched between a p-doped emitter and p-doped collector.
- In this type of transistors, holes are the majority carriers and electrons are the minority carriers.
- In a PNP transistor, the emitter is forward biased and the collector is reverse biased.



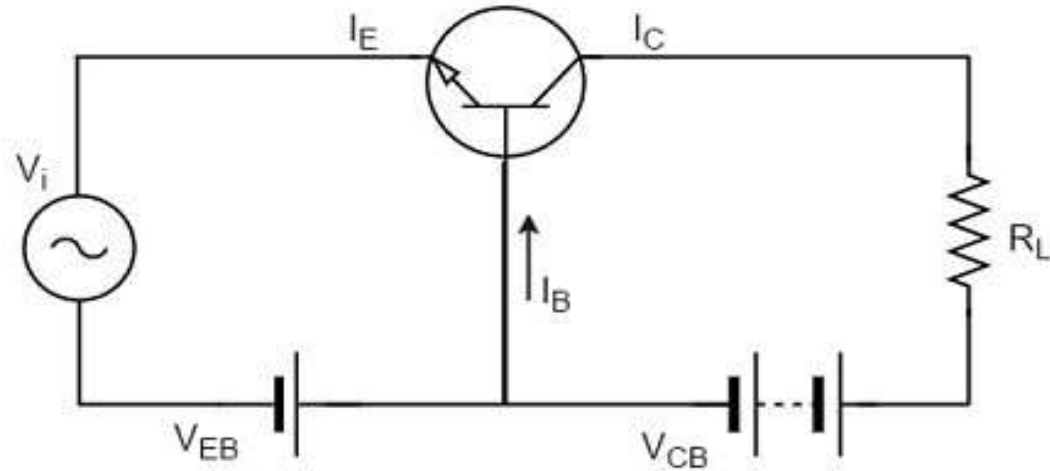
Transistor as an Amplifier:

- For a transistor to act as an amplifier, it should be properly biased. Here, let us focus how a transistor works as an amplifier.

Transistor Amplifier

- A transistor acts as an amplifier by raising the strength of a weak signal.
- The DC bias voltage applied to the emitter base junction, makes it remain in forward biased condition.
- This forward bias is maintained regardless of the polarity of the signal.
- The below figure shows how a transistor looks like when connected as an amplifier.

Transistor as an Amplifier:



- The low resistance in input circuit, lets any small change in input signal to result in an appreciable change in the output.
- The emitter current caused by the input signal contributes the collector current, which when flows through the load resistor R_L , results in a large voltage drop across it.
- Thus a small input voltage results in a large output voltage, which shows that the transistor working as an amplifier.

Transistor configurations:

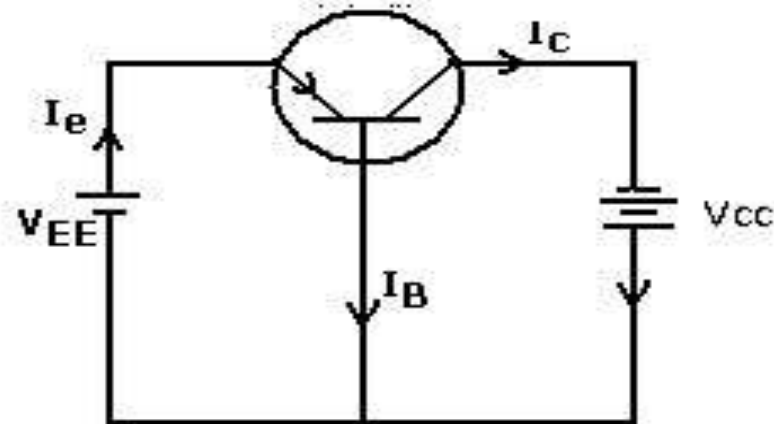
- We know that generally the transistor has three terminals – emitter (E), base (B) and collector.
- But in the circuit connections we need four terminals, two terminals for input and another two terminals for output.
- To overcome these problems we use one terminal as common for both input and output actions
- Using this property we construct the circuits and these structures are called transistor configurations.
- Generally there are three different configurations of transistors and they are common base (CB) configuration, common collector (CC) configuration and common emitter (CE) configuration.

Transistor configurations:

- The behavior of these three different configurations of transistors with respect to gain is given below.
 - **Common Base (CB) Configuration:** no current gain but voltage gain
 - **Common Collector (CC) Configuration:** current gain but no voltage gain
 - **Common Emitter (CE) Configuration:** current gain and voltage gain

Common Base Configuration:

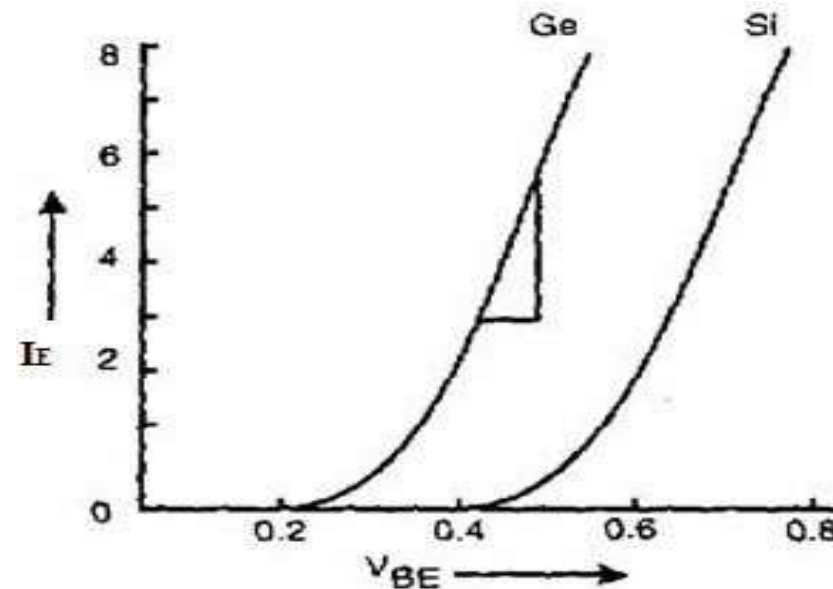
- The voltage is applied at the junction of the emitter and the base. Here the emitter and the base are referred to as the input side and the collector is known for the output side of the circuit connection.
- The characteristic of the input is based on the voltage applied at the terminals base and the emitter and the current at the emitter terminal.
- The output characteristic for this configuration is based on the parameters of the voltage applied at the terminals of the base and the collector and the current generated at the collector terminal.



Common Base Configuration:

Input characteristics of CB:

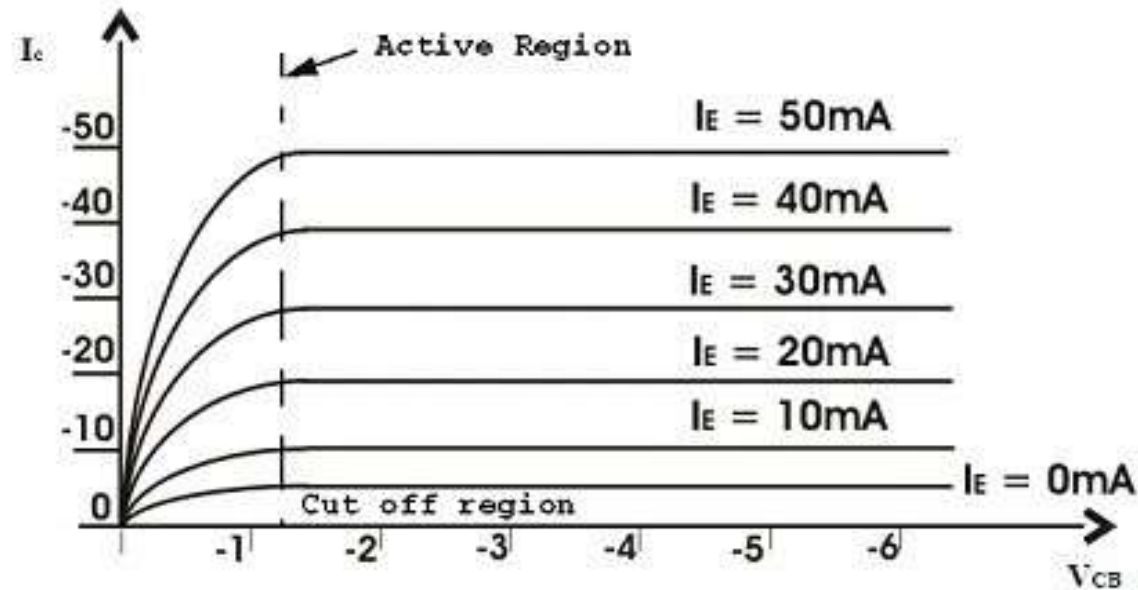
- The input characteristics for this type of configuration is measured by the variation at the voltage value at the terminals of the emitter and the base at different points by keeping the voltage value at the collector and the base as constant.
- From this the input value of the current that is at emitter is measured. Based on which the graph is plotted.



Common Base Input Characteristics

Output characteristics of CB:

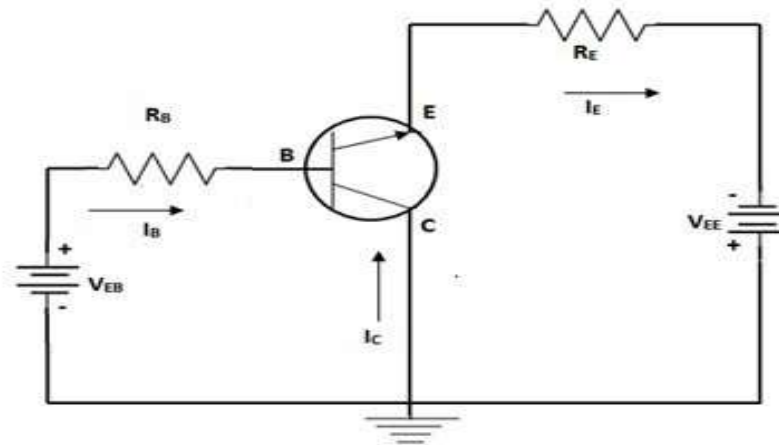
- The graph is plotted between the voltage at the output and the current by keeping the input value of the current at the constant gives the output characteristics for this configuration.



Common Base Output Characteristics:

Common Collector Configuration:

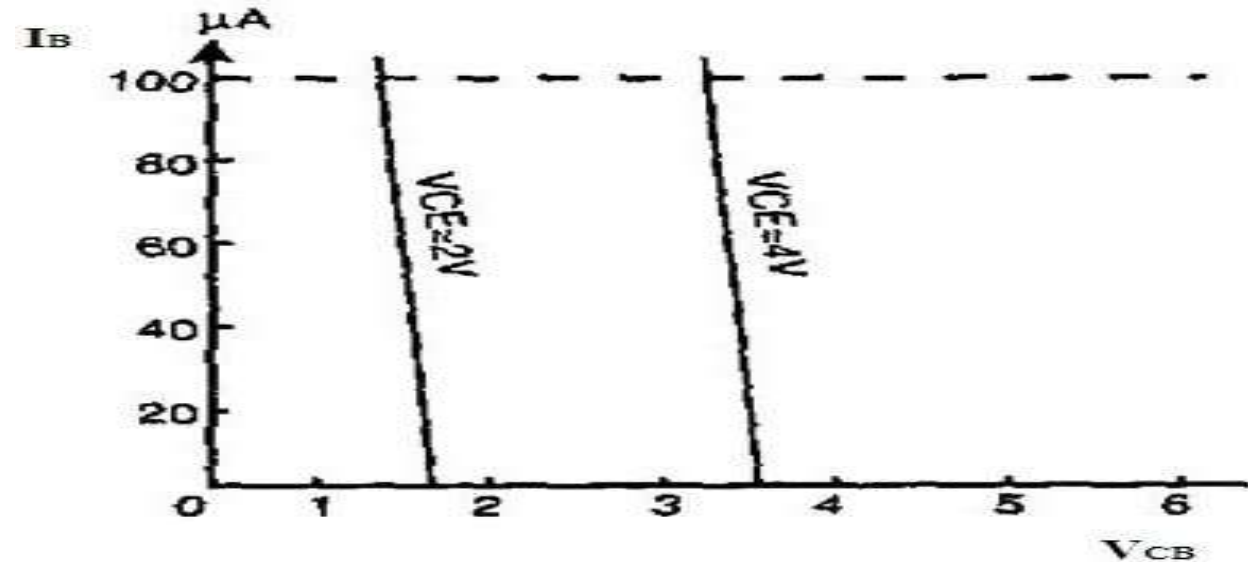
- This is the configuration in which the collector terminal is made common for both the input and the output connections of the circuit.
- In this the voltage at the terminal emitter follows the voltage of the base terminal. Hence this circuit is referred to as the emitter following circuit. The input value of the impedance is high.
- The considered input signals are applied in between the terminals of the collector and the base. The output is to be taken or considered in between the terminals of the collector and the emitter.



Common Collector Configuration:

Input characteristics of CC:

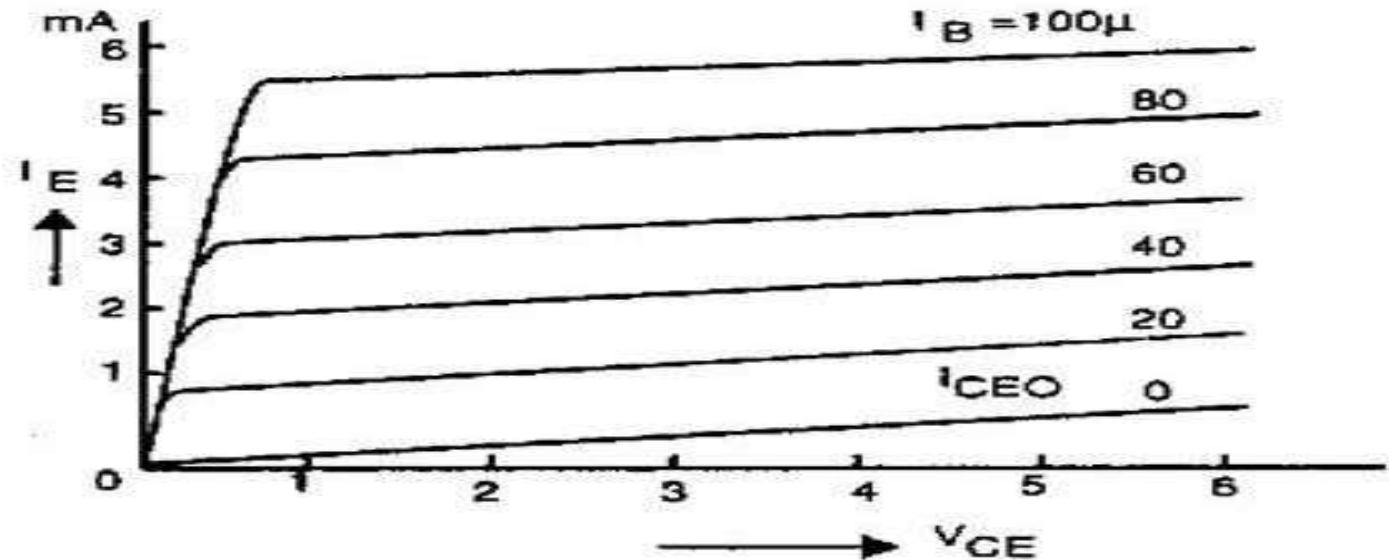
- The characteristics for this type of configuration are very different in comparison with other configurations.
- Here the voltage at the collector and the base terminal is determined by the level of the voltage at the emitter and the collector.
- By maintaining voltage at the collector and the emitter at the constant values the graph is plotted between the parameters of the base current and the voltage value at the collector and the base terminals.



Common Collector Input Characteristics:

Output characteristics of CC:

- As the collector configuration is known to follow the emitter configuration the operation of the output is similar to that of the emitter configuration.
- In this configuration if there is no voltage is applied at the base terminal there will be no amount of current flow is evident in the circuitry.
- The graph is plotted between the emitter current and the voltage at the terminals of the collector and the emitter by maintaining the value at the base current at constant.



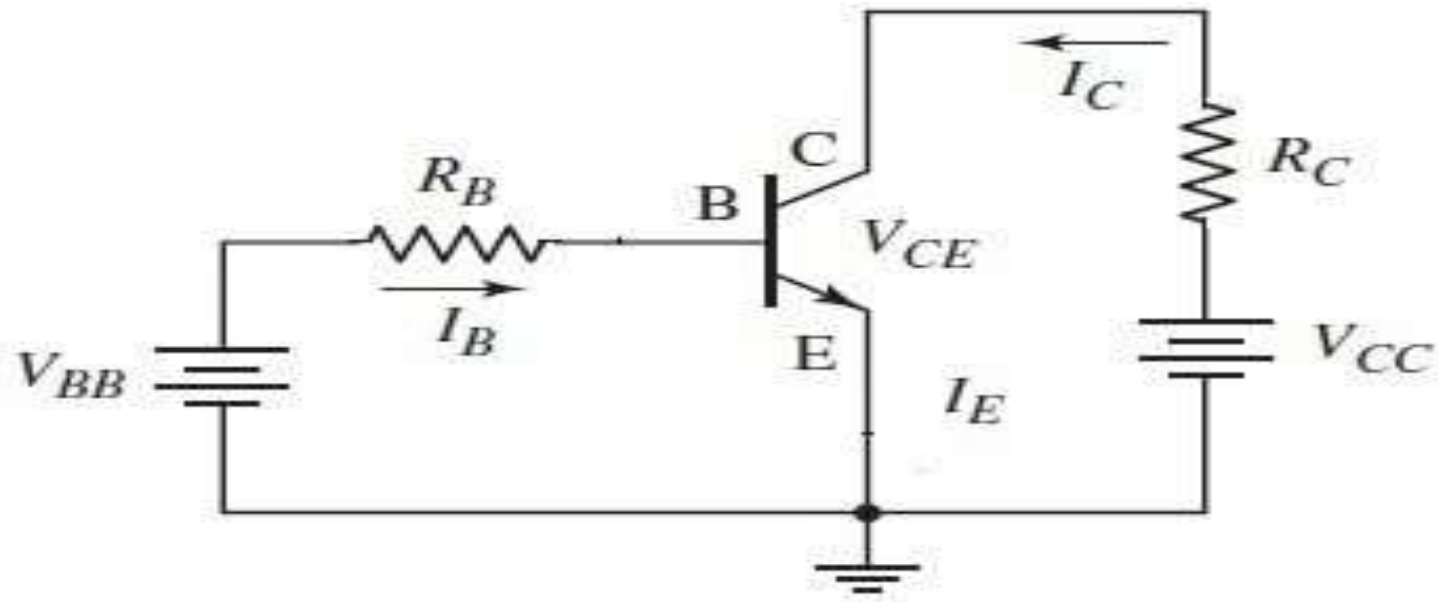
Common Collector Output Characteristics:

Common Emitter Configuration:



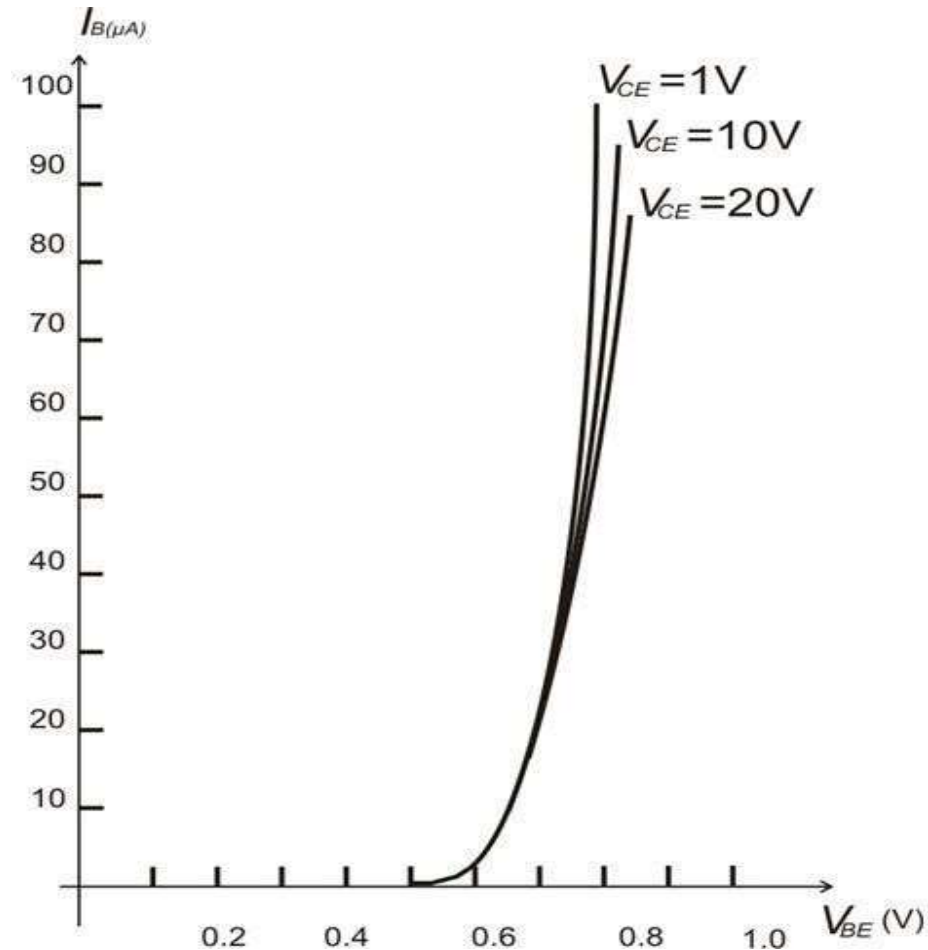
- This is the most widely used configuration because of the gain in both voltage and the currents increments the gain value of the power.
- In this the voltage of the input is applied in between the terminals of the emitter and the base. The output is taken across the terminals of emitter and the collector. Hence this circuit is inverting type.
- The gain between the ratio of current at the collector terminal and the emitter terminal is measured in terms of alpha. The gain for the ratio between the currents of the collector terminal and the base is measured in terms of beta.
- The generated output signal has the shifting in phase about 180 degrees that represents the input and the output signals are inversely related in terms of phases.

Common Emitter Configuration:



Input characteristics of CE:

- The graph is plotted between the current at the base and the voltage value at the terminals of base and the emitter.

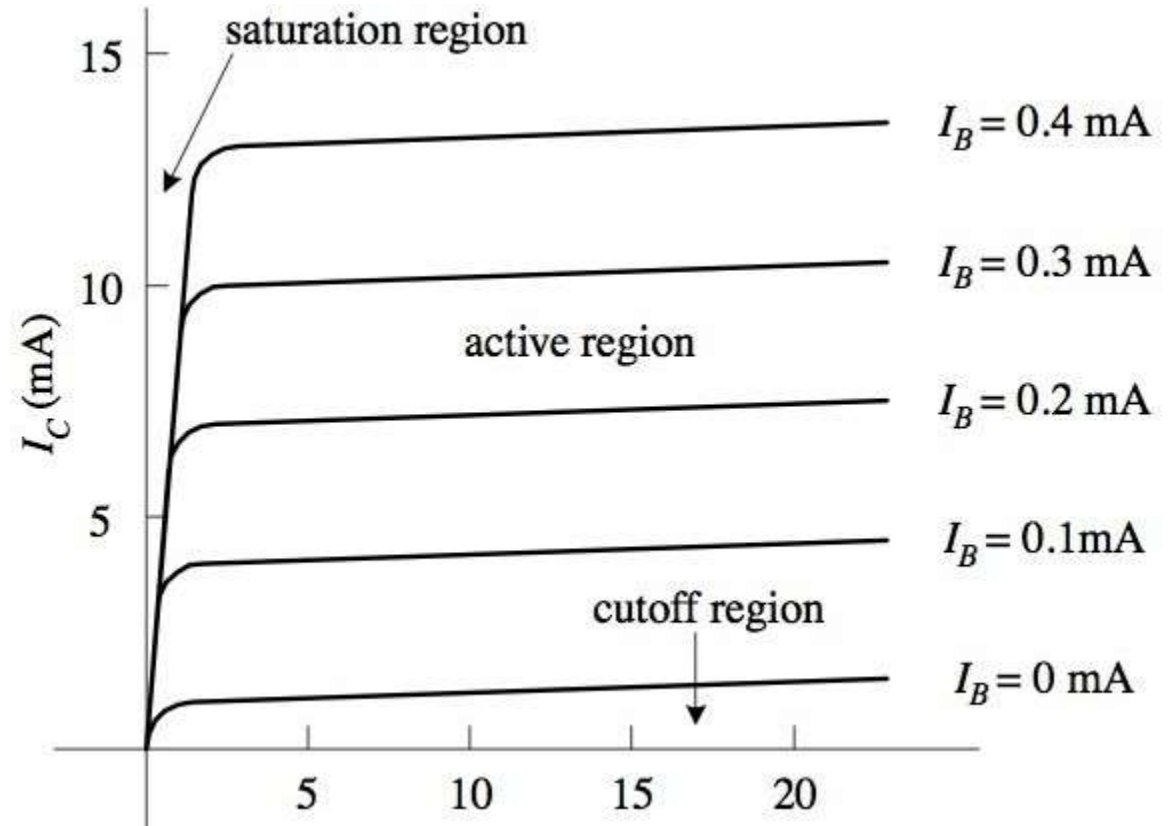


Common Emitter Input Characteristics:

Output characteristics of CE:

- The graph is plotted between the values of the collector current and the voltage value of the terminals of collector and the emitter.

Common Emitter Output Characteristics:



Comparison of CB, CE, CC transistor configurations:

S.No.	Characteristic	Common base	Common emitter	Common collector
1.	Input resistance	Very low (20 Ω)	Low (1 k Ω)	High (500 k Ω)
2.	Output resistance	Very high (1 M Ω)	High (40 k Ω)	Low (50 Ω)
3.	Input current	I_E	I_B	I_B
4.	Output current	I_C	I_C	I_E
5.	Input voltage applied between	Emitter and base	Base and emitter	Base and collector
6.	Output voltage taken from	Collector and base	Collector and emitter	Emitter and collector
7.	Current amplification factor	$\alpha_{dc} = I_C/I_E$	$\beta_{dc} = I_C/I_B$	$\gamma_{dc} = I_E/I_B$
8.	Current gain	Less than unity	High (20 to few hundreds)	High (20 to few hundreds)
9.	Voltage gain	Medium	Medium	Low
10.	Amplifications	At the input stage of multistage amplifier	For audio signal amplification	For impedance matching

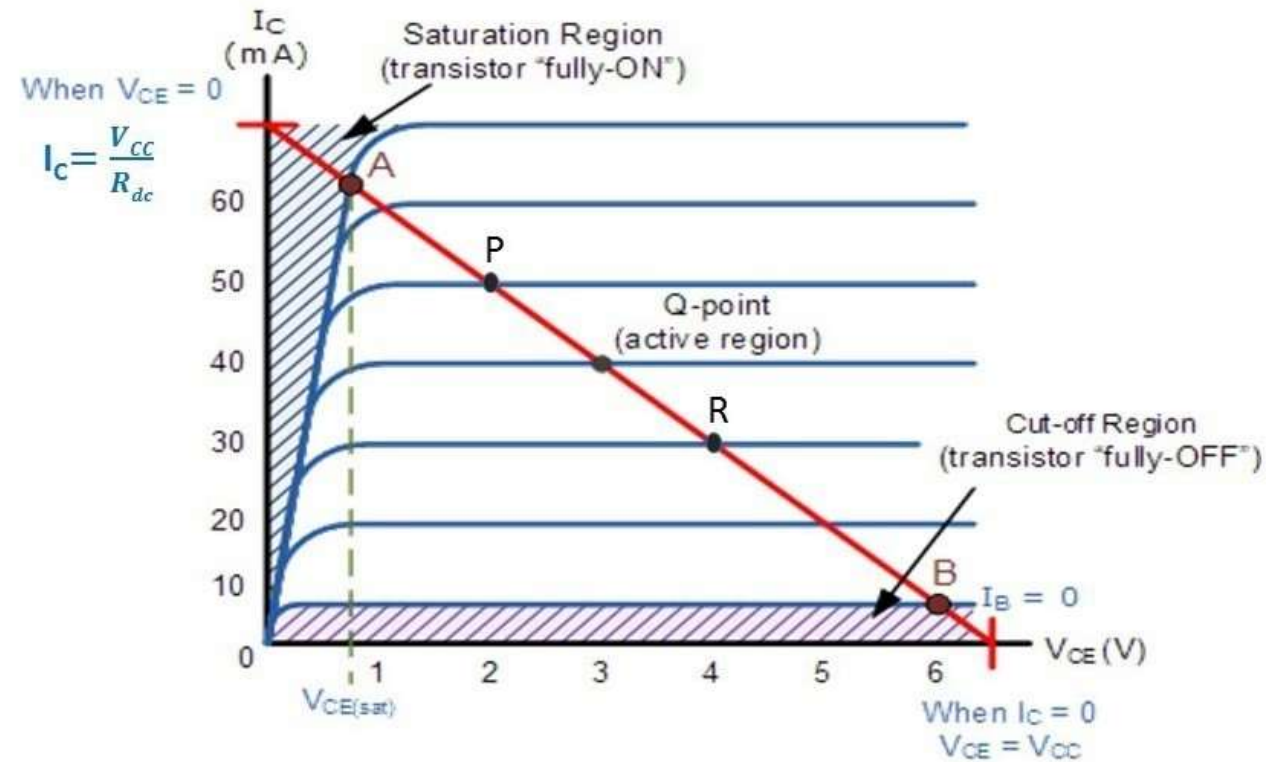
Operating point in transistor:

Definition: The point which is obtained from the values of the I_C (collector current) or V_{CE} (collector-emitter voltage) when no signal is given to the input is known as the 'operating point' or 'Q-point' in a transistor.

- It is called operating point because variations of I_C (collector current) and V_{CE} (collector-emitter voltage) takes place around this point when no signal is applied to the input.
- The operating point is also called quiescent (silent) point or simply Q-point because it is a point on $I_C - V_{CE}$ characteristic when the transistor is silent or no input signal is applied to the circuit.
- The operating point can be easily obtained by the DC load line method. The DC load line is shown in below graph.

Operating point in transistor:

DC LOAD LINE AND OPERATING POINT



Self bias of a Transistor:



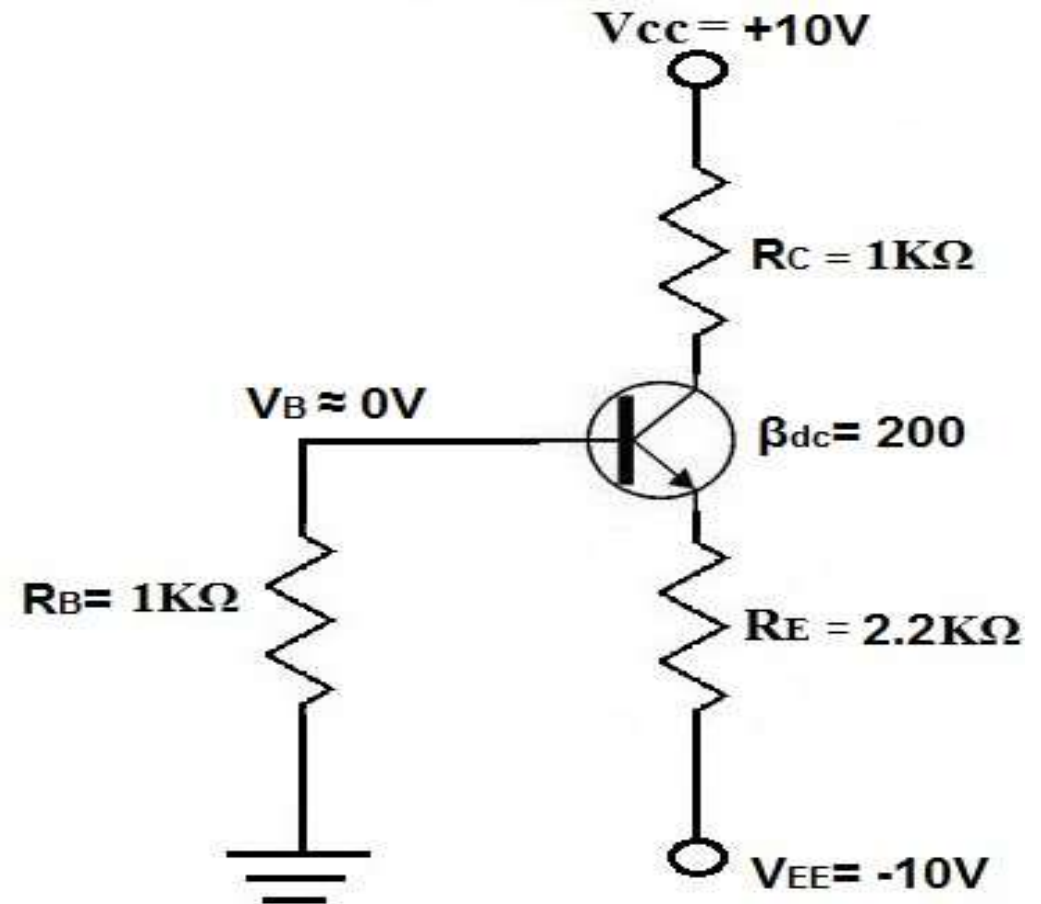
What is Transistor Biasing?

- Transistor Biasing is the process of setting a transistor's DC operating voltage or current conditions to the correct level so that any AC input signal can be amplified correctly by the transistor.
- One way to bias a BJT transistor is a method called emitter bias.
- **Emitter bias** is a very good and stable way to bias transistors if both positive and negative power supplies are available. Emitter bias fluctuates very little with temperature variation and transistor replacement.

Self bias of a Transistor:

Beside is a BJT transistor receiving emitter bias:

- You can see how that both positive and negative voltage supplies are necessary to bias a transistor in this way.
- Positive voltage is fed to the collector of the transistor and negative voltage is fed to the emitter.



Calculations:

- Assuming the above is a silicon transistor, the voltage drop across the base emitter is equal to 0.7v

- To calculate the emitter current(I_E):
$$I_{EQ} = \frac{V_{EE} - V_{BE}}{\frac{R_B}{(\beta_{dc})} + R_E}$$

- So in the circuit above, the emitter current calculation is:

$$I_E = (10V - 0.7V) / (2.2K\Omega + (1K\Omega / 200)) = 4.22mA$$

- To calculate the collector voltage, V_C , the formula is:

$$V_C = V_{CC} - I_{C}R_C = 10V - (4.23mA \times 1K\Omega) = 5.77V$$

- Again, emitter bias is an effective way to bias a BJT transistor.
- However, the voltage divider bias is still the most popular way to do so, while base bias is the least popular way because of the instability it provides if β changes.

Bias Compensation:

Bias compensation is of two techniques and they are;

1. Using thermistors
2. Using sensitors

Bias compensation using Thermistors:

- In this method, a temperature sensitive resistive element called "thermistor" is used.
- There are elements of negative temperature coefficient whose resistance decreases with increase in temperature.
- There are two for bias compensation using thermistor and they are;
 - i. One method is carried out by replacing the "R2" resistor with thermistor "Rt" in a self-bias circuit as shown in figure.

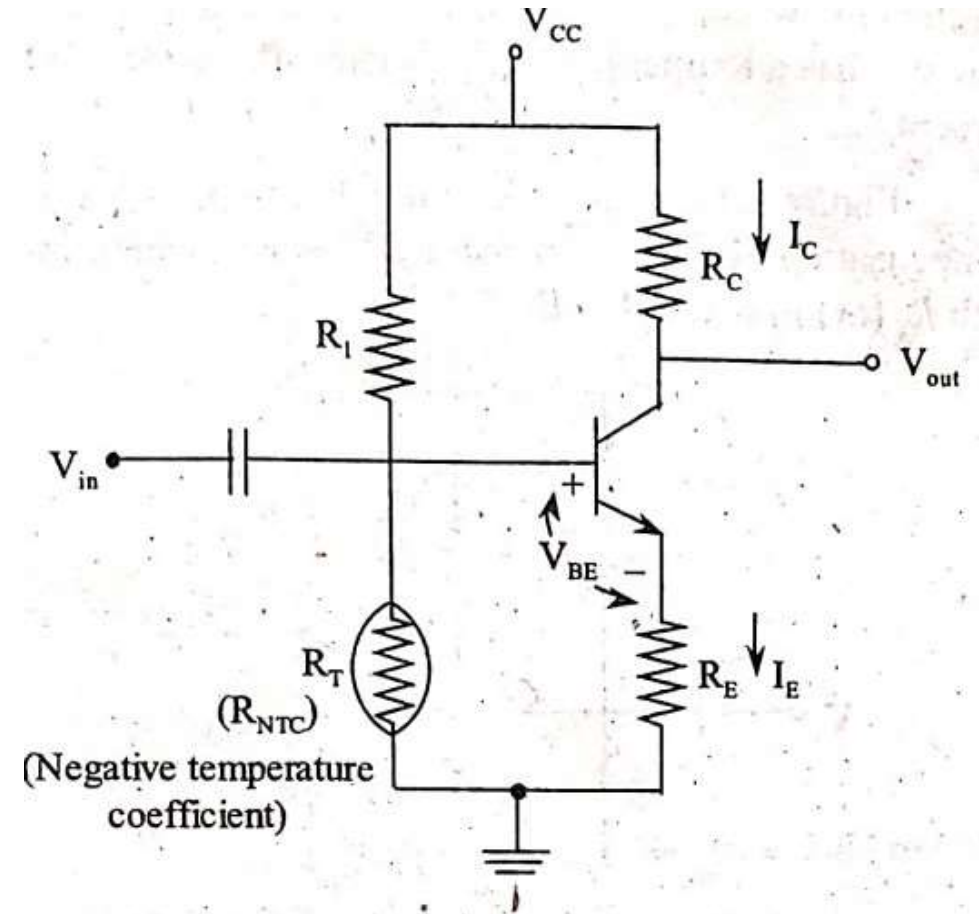


Figure (1) : Thermistor Compensation Technique

ii. In the second method, the thermistor is connected between the supply voltage V_{cc} and emitter as shown in figure.

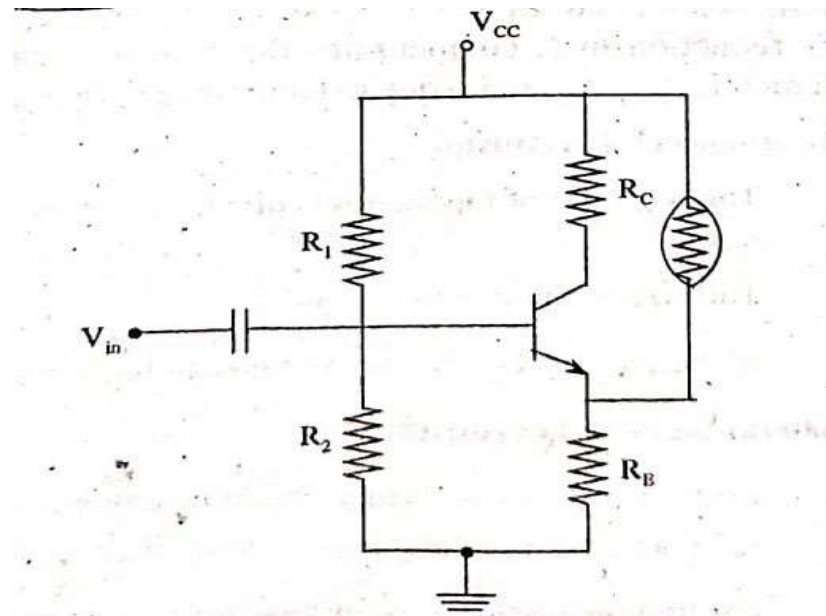


Figure (2) : Thermistor Compensation Technique

Bias compensation using sensistor:

- Sensistor is a temperature sensitive resistor with a positive temperature coefficient, which is used in transistor compensation.
- The sensistor has a temperature coefficient of resistance that is +0.7 percent/c.
- The below figure shows the circuit when sensistor is used as a compensating element.

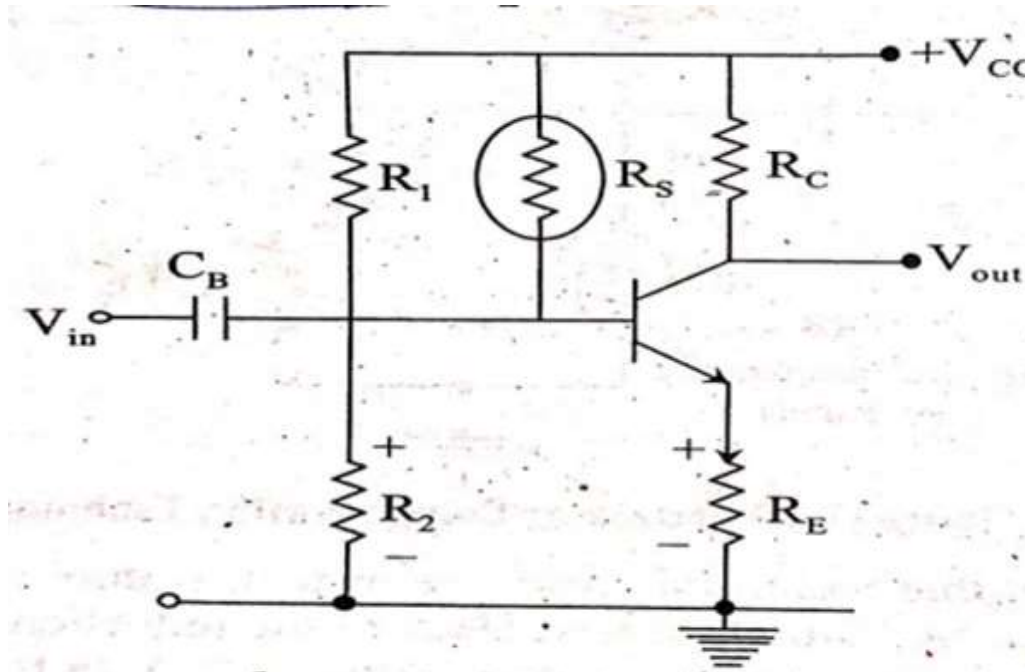


Figure (3): Sensistor Bias Compensation

Thermal runaway and stability:

Thermal runaway: The expression for the collector current of common emitter circuit is given as (equation 1).

- When the temperature raises, the parameters " β ", I_{cbo} and I_b in equation 1 also increase.
- The reverse saturation current I_{cbo} increases with rise in temperature, I_{cbo} gets doubles.
- Initially the collector base junction temperature is increased by collector current " I_c " which in turn increases I_{cbo} .
- This process becomes cumulative and leads to "Thermal runaway".

Condition for Thermal stability:

- To avoid thermal runaway the required condition is that the rate at which heat is released at collector junction must not exceed the rate at which the heat can be dissipated under steady state condition (equation 2).
- Since the steady state temperature rise at collector junction is proportional to the power dissipated at the junction (equation 3).

or
i.e., $\Delta T = T_J - T_A = \theta P_D$

where, $T_J =$ Junction temp ($^{\circ}\text{C}$)

$T_A =$ Ambient temp ($^{\circ}\text{C}$)

$P_D =$ Dissipate power (watts)

$\theta =$ Thermal resistance

$$T_J - T_A = \theta P_D \quad \text{--- (3)}$$

Differentiating eq (3) w.r.t T_J

$$\Rightarrow 1 = \theta \frac{dP_D}{dT_J}$$

$$\frac{dP_D}{dT_J} = \frac{1}{\theta} \quad \text{--- (4)}$$

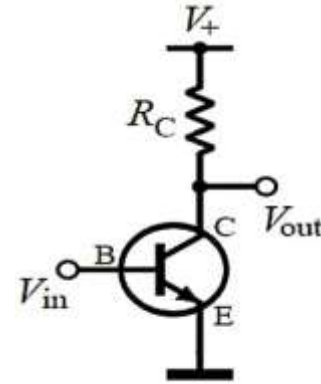
Then, from eq (4)

$$\frac{dP_D}{dT_J} < \frac{1}{\theta} \text{ is condition which must be}$$

- Thus the transistor cannot runaway below a specified ambient temperature under any biasing condition.

Common emitter amplifier frequency response:

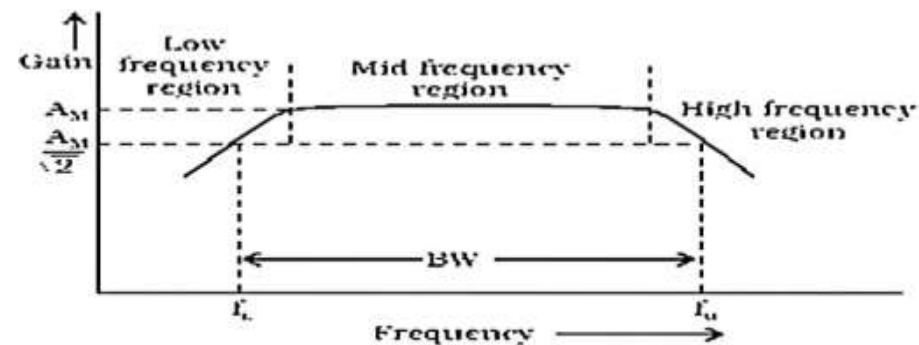
- Basic bi-polar junction transistor (BJT) amplifier topologies, typically used as a voltage amplifier.



Common Emitter Amplifier

CE amplifier frequency response:

- The voltage gain of CE amplifier varies with signal frequency.
- The curve drawn between voltage gain and the signal frequency of an amplifier is known as frequency response.

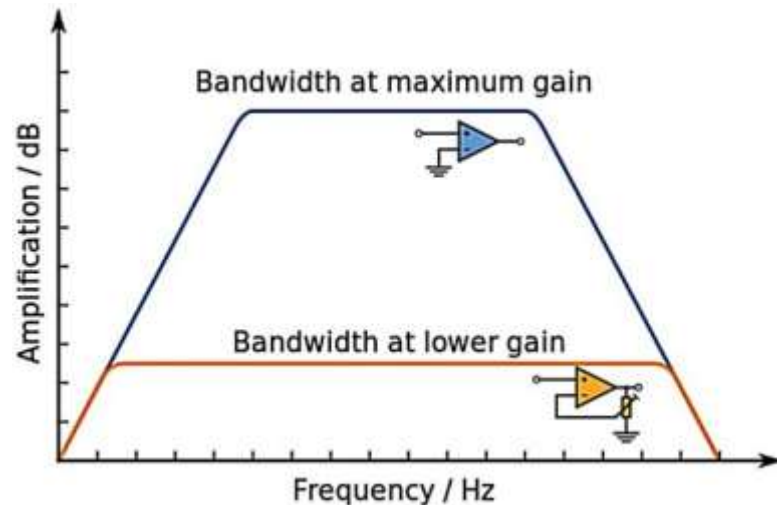


Frequency Response

- i. At low frequencies ($<f_l$): The reactance of coupling capacitor C_2 is relatively high and hence very small part of signal will pass from the amplifier stage to the load.
- ii. Due to this, it causes drop off of voltage gain at low frequencies.
- iii. The reactance of coupling capacitor C_2 is very small and it behaves as short circuit. This increases the loading effect of the amplifier stage.
- iv. Due to this, the voltage gain drops off at high frequency.

Gain bandwidth product:

- The gain bandwidth product, GBW is defined as the product of the open loop voltage gain and the frequency at which it is measured. GBW is expressed in units of Hertz.



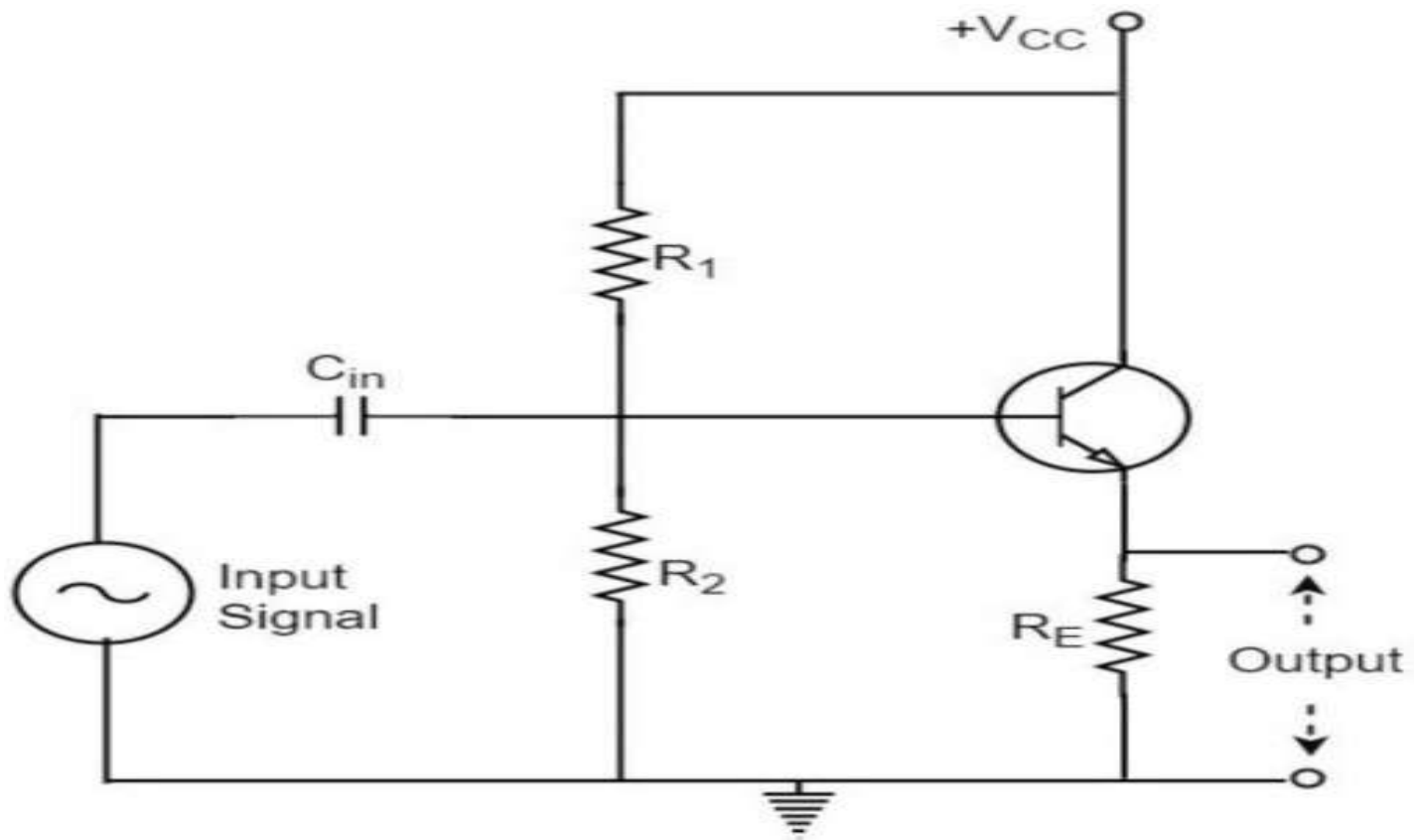
- Relevance to this quantity is commonly specified for operational amplifiers, and allows circuit designers to determine the maximum gain that can be extracted from the device for a given frequency and vice versa.
- Gain and bandwidth in an amplifier are inversely proportional to each other and their relationship is summarized as the unity gain bandwidth. Unity gain bandwidth defines the frequency at which the gain of an amplifier is equal to 1.

Emitter follower:

Emitter follower circuit has a prominent place in feedback amplifiers. Emitter follower is a case of negative current feedback circuit. This is mostly used as a last stage amplifier in signal generator circuits.

The important features of Emitter follower are:

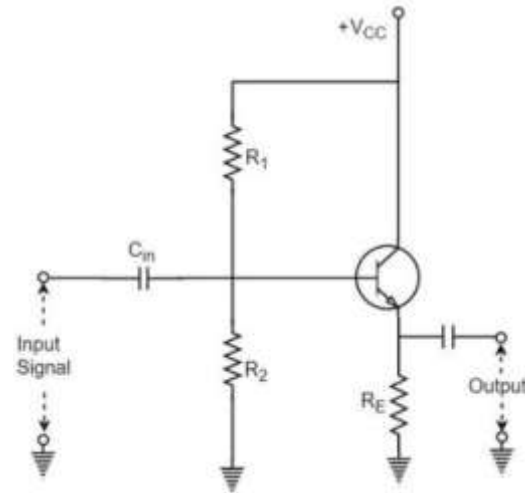
- It has high input impedance.
- It has low output impedance.
- It is ideal circuit for impedance matching.



Emitter follower

Voltage gain of emitter follower:

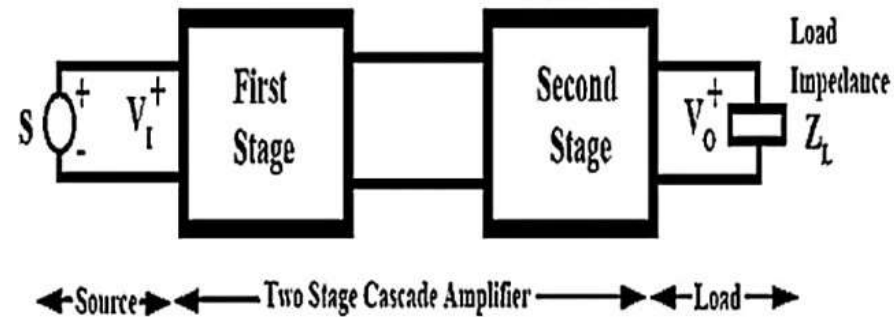
- As the emitter follower circuit is a prominent one, let us try to get the equation for the voltage gain of an emitter follower circuit. Our emitter follower circuit looks as follows-



Voltage gain of emitter follower

RC coupled amplifier:

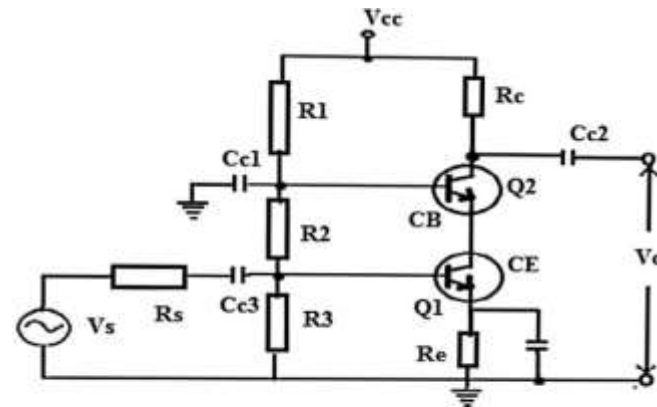
- RC coupling is the most widely used method of coupling in multistage amplifiers. In this case the resistance R is the resistor connected at the collector terminal and the capacitor C is connected in between the amplifiers. It is also called a blocking capacitor, since it will block DC voltage.
- The frequency response of RC amplifier provides constant gain over a wide frequency range, hence most suitable for audio applications.



cascade-amplifier

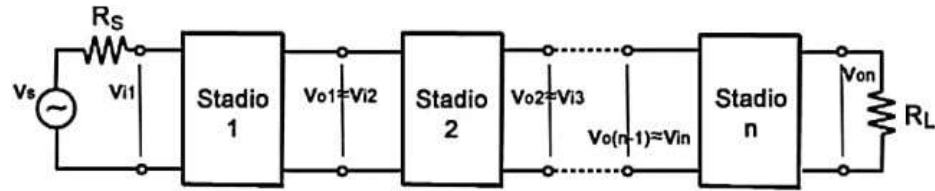
Two cascaded CE amplifier:

- The two stages of cascaded CE (common emitter) are shown in the following circuit. Here the voltage divider can be formed by using the input and output resistances of the first and next stage.



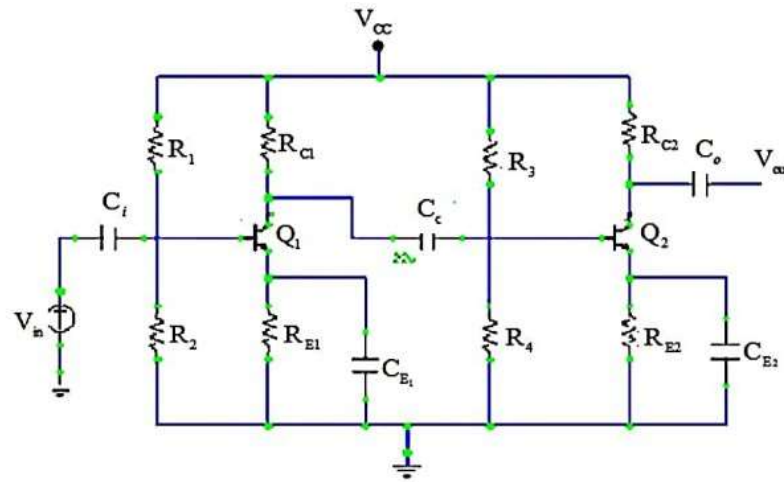
cascade-amplifier-circuit

- A multistage amplifier design using CE (common emitter) as the primary stage as well as CB (common base) as the second stage is named as a cascade amplifier. The connection between cascade & cascade can also be possible using FET amplifiers.
- A multistage amplifier using two or more single stage common emitter amplifier is called as cascade amplifier.



multistage-amplifier

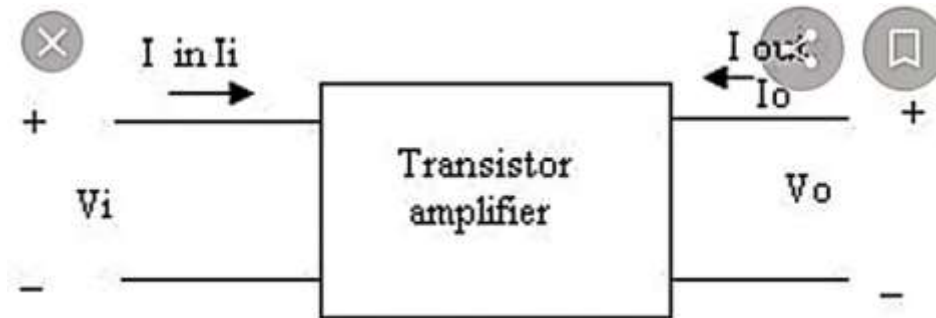
- The design of the multistage CE amplifier was done using the designed single stage CE amplifier as the basic configuration.



A multistage CE amplifier configuration design

Transistor at low frequencies:

- The transistor can be employed as an amplifying device, that is, the output ac power is greater than the input ac power. The factor that permits an ac power output greater than the input ac power is the applied DC power.
- Thus to simplify the analysis of BJT, its operation is restricted to the linear V-I characteristics around the Q-point that is in the active region.
- The factor that permits an ac power output greater than the input AC power is the applied DC power. The amplifier is initially biased for the required DC voltages and currents.



ANALOG AND DIGITAL ELECTRONICS.

UNIT – III

FET's and Digital Circuits

Introduction:



- The Field effect transistor is abbreviated as FET , it is an another semiconductor device like a BJT which can be used as an amplifier or switch.
- FET is also a three terminal device, labelled as source, drain and gate.
- The material that connects the source to drain is referred to as the channel.
- FET operation depends only on the flow of majority carriers ,therefore they are called uni polar devices.
- As FET has conduction through only majority carriers it is less noisy than BJT.
- FETs are much easier to fabricate and are particularly suitable for ICs because they occupy less space.
- FET amplifiers have low gain bandwidth product due to the junction capacitive effects and produce more signal distortion except for small signal operation.
- The performance of FET is relatively unaffected by ambient temperature changes. As it has a negative temperature coefficient at high current levels, it prevents the FET from thermal breakdown.

Classification of FET:

There are two major categories of field effect transistors:

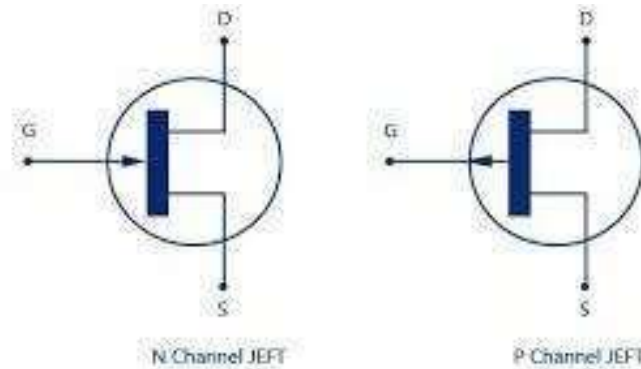
1. Junction Field Effect Transistors

2. MOSFETs

1. Junction Field Effect Transistors:

When the channel is of N-type the JFET is referred to as an N-channel JFET, when the channel is of P-type the JFET is referred to as P-channel JFET.

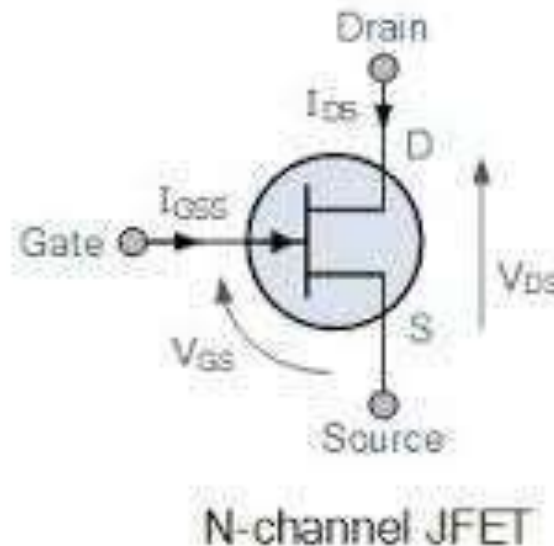
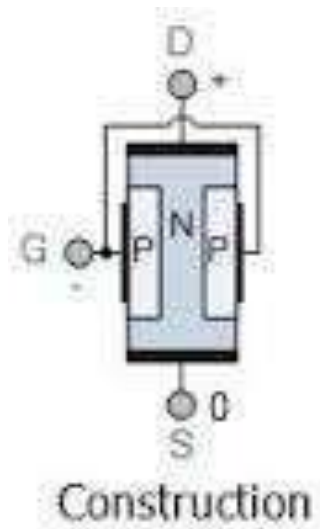
• The schematic symbols for the N-channel and P-channel JFETs are shown in the figure.



JFET-N-Channel and P-channel Schematic Symbol

Construction Of N-Channel JFET:

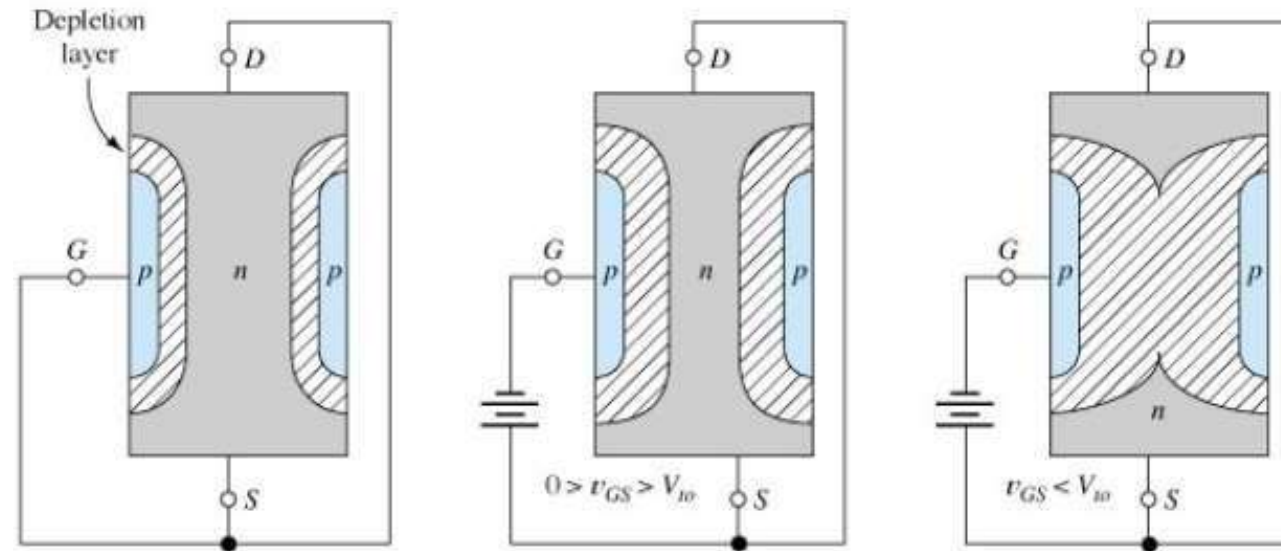
- A piece of N- type material, referred to as channel has two smaller pieces of P-type material attached to its sides, forming PN junctions. The channel ends are designated as the drain and source. And the two pieces of P-type material are connected together and their terminal is called the gate. Since this channel is in the N-type bar, the FET is known as N-channel JFET.



Operation Of N-Channel JFET:

The overall operation of the JFET is based on varying the width of the channel to control the drain current.

N-Channel JFET Operation



(a) Bias is zero and depletion layer is thin; low-resistance channel exists between the drain and the source

(b) Moderate gate-to-channel reverse bias results in narrower channel

(c) Bias greater than pinch-off voltage; no conductive path from drain to source

Characteristics Of N-Channel JFET:

The family of curves that shows the relation between current and voltage are known as characteristic curves.

There are two important characteristics of a JFET.

- 1) Drain or VI Characteristics
- 2) Transfer characteristics

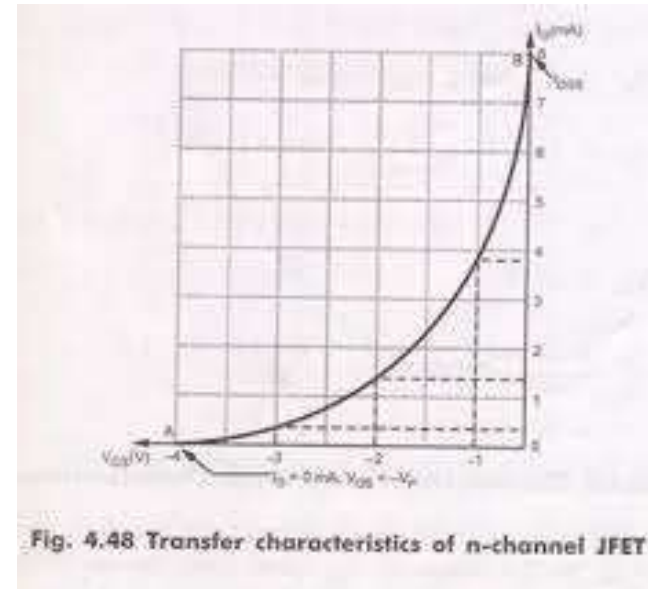
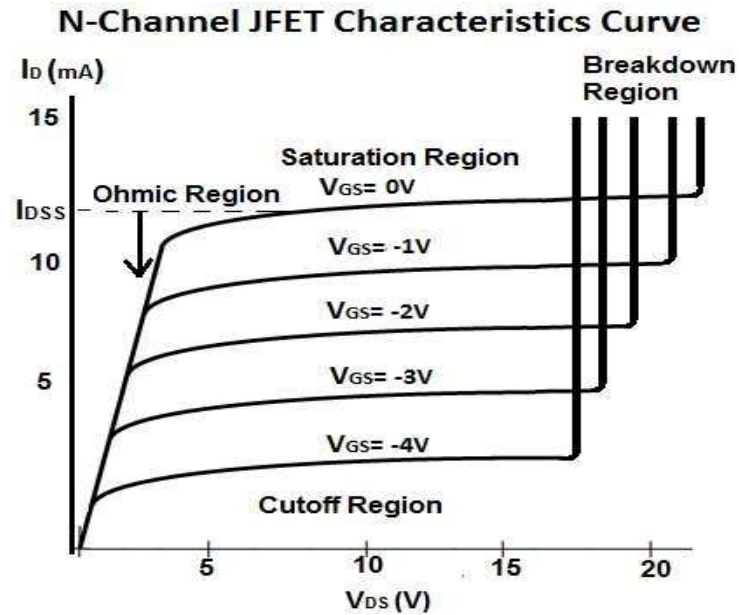
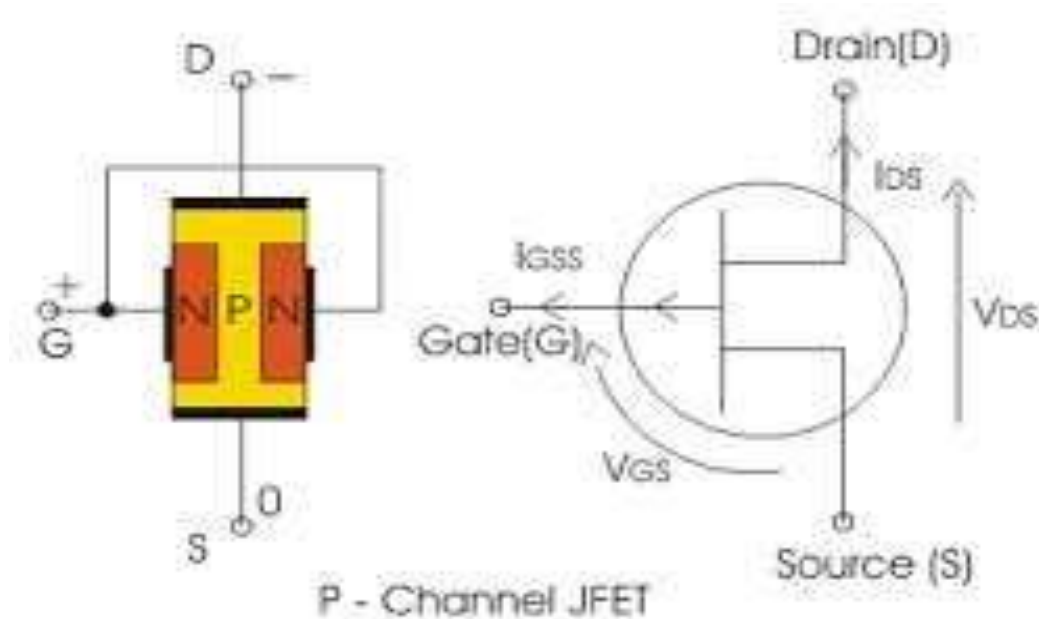


Fig. 4.48 Transfer characteristics of n-channel JFET

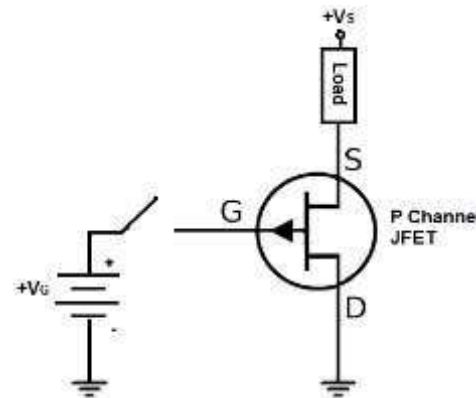
Construction Of P-Channel JFET:

- The construction diagram of P channel junction field effect transistor (JFET) is shown in above figure. Its construction is similar to the [N channel JFET](#) excepts that it consist of a P type silicon bar with two N type heavily doped regions diffused on opposites sides of its middle part. The JFET in which the current conduction takes place only due to holes as majority charge carriers is known as [P channel JFET](#).



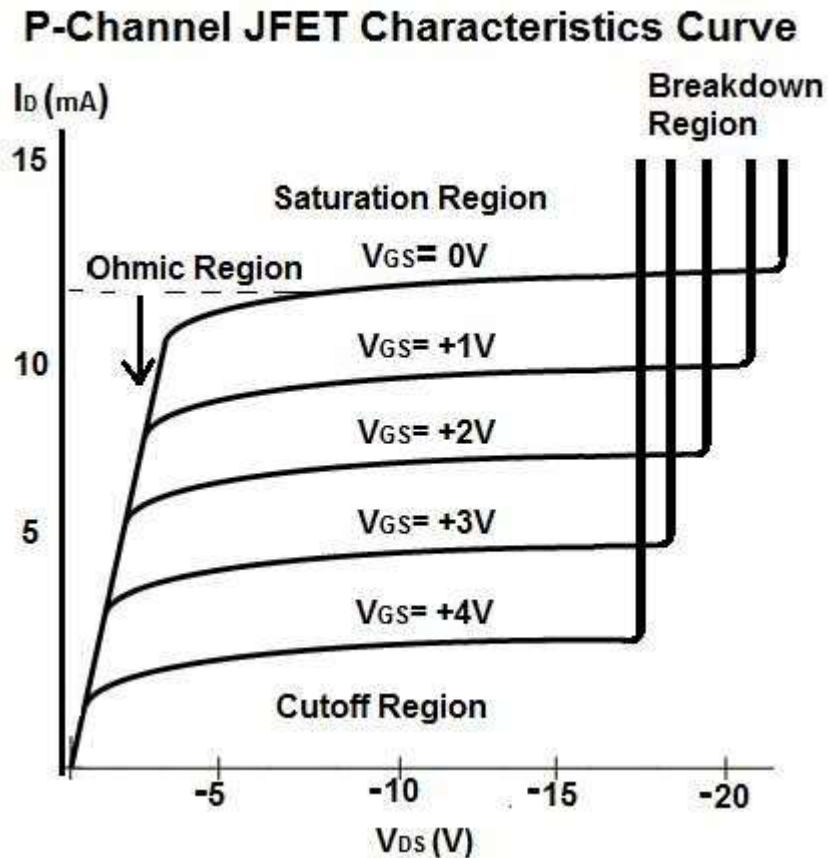
Operation Of P-Channel JFET:

- When no voltage is applied to the gate of a P-Channel JFET, current (holes) flows freely through the central P-channel. This is why JFETs are referred to as "normally on" devices. Even without any voltage, they conduct current across from source to drain.
- To turn on a P-channel JFET, apply a positive voltage V_S to the source terminal of the transistor with no voltage applied to the gate terminal of the transistor. This will allow a current to flow through the drain-source channel. If the gate voltage, V_G , is $0V$, the drain current is at its largest value for safe operation, and the JFET is in the ON active region.
- To turn off a P-channel JFET, there are 2 steps you can take. You can either cut off the bias positive voltage, V_S , that powers the source terminal. Or you can apply a positive voltage to the gate. When a positive voltage is applied to the gate, the source-drain current is reduced. As the gate voltage, V_G , becomes more positive, the current lessens and lessens until it completely reaches cutoff, which is when then JFET is in the Off condition and no current conducts across from source to drain. This stops all drain-source current flow.



Characteristics Of P-Channel JFET:

- The characteristics curve of a P Channel JFET transistor shown below is the the graph of the drain current, I_D versus the gate-source voltage, V_{GS} .



Characteristics Of P-Channel JFET:



The Regions that make this characteristic curve are the following:

Cutoff Region- This is the region where the JFET transistor is off, meaning no drain current, I_D flows through the source-drain region.

Ohmic Region- This is the region where the JFET transistor begins to show some resistance to the drain current I_D that is beginning to flow through the source-drain region. This is the only region in the curve where the response is linear.

Saturation Region- This is the region where the JFET transistor is fully operational and maximum current is flowing. During this region, the JFET is On and active.

Breakdown Region- This is the region where the voltage that is supplied to the source terminal of the transistor exceeds the necessary maximum. At this point, the JFET loses its ability to resist current because too much voltage is applied across its source-drain terminals. The transistor breaks down and current flows from source to drain.

Why the gate to source junction of a JFET be always reverse biased?

- The gate to source junction of a JFET is never allowed to become forward biased because the gate material is not designed to handle any significant amount of current. If the junction is allowed to become forward biased, current is generated through the gate material. This current may destroy the component.
- There is one more important characteristic of JFET reverse biasing i.e. J FET 's have extremely high characteristic gate input impedance. This impedance is typically in the high mega ohm range. With the advantage of extremely high input impedance it draws no current from the source. The high input impedance of the JFET has led to its extensive use in integrated circuits. The low current requirements of the component makes it perfect for use in ICs. Where thousands of transistors must be etched on to a single piece of silicon. The low current draw helps the IC to remain relatively cool, thus allowing more components to be placed in a smaller physical area.

JFET Parameters:

- The electrical behavior of JFET may be described in terms of certain parameters. Such parameters are obtained from the characteristic curves.

1. AC Drain resistance(r_d): It is also called dynamic drain resistance and is the a.c. resistance between the drain and source terminal, when the JFET is operating in the pinch off or saturation region. It is given by the ratio of small change in drain to source voltage ΔV_{ds} to the corresponding change in drain current ΔI_d for a constant gate to source voltage V_{gs} .

Mathematically it is expressed as $r_d = \Delta V_{ds} / \Delta I_d$ where V_{gs} is held constant.

2. TRANS CONDUCTANCE (g_m): It is also called forward transconductance . It is given by the ratio of small change in drain current (ΔI_d) to the corresponding change in gate to source voltage (ΔV_{gs}).

Mathematically the transconductance can be written as $g_m = \Delta I_d / \Delta V_{gs}$.

3. AMPLIFICATION FACTOR (μ): It is given by the ratio of small change in drain to source voltage (ΔV_{ds}) to the corresponding change in gate to source voltage (ΔV_{gs}) for a constant drain current (I_d).

Thus $\mu = \Delta V_{ds} / \Delta V_{gs}$ when I_d held constant.

The amplification factor μ may be expressed as a product of transconductance (g_m) and ac drain resistance (r_d)

$$\mu = \Delta V_{ds} / \Delta V_{gs} = g_m r_d$$

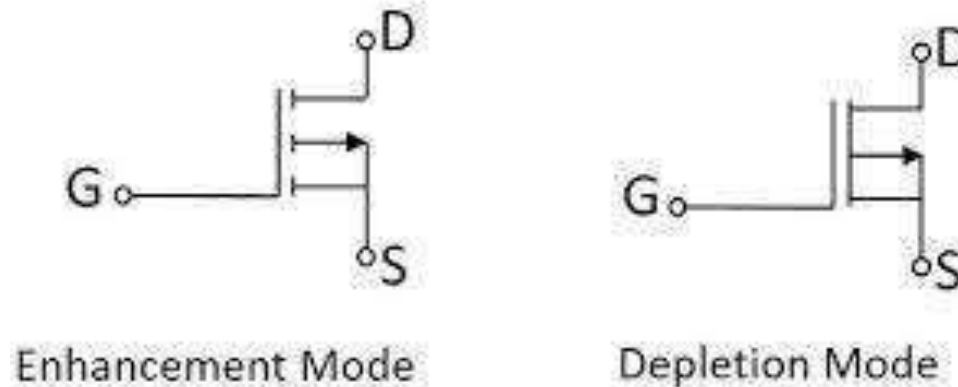
MOSFET:

These are further sub divided in to P- channel and N-channel devices.

MOSFETs are further classified in to two types

1. Enhancement MOSFETs and
2. Depletion MOSFETs

Symbols of P-Channel MOSFET

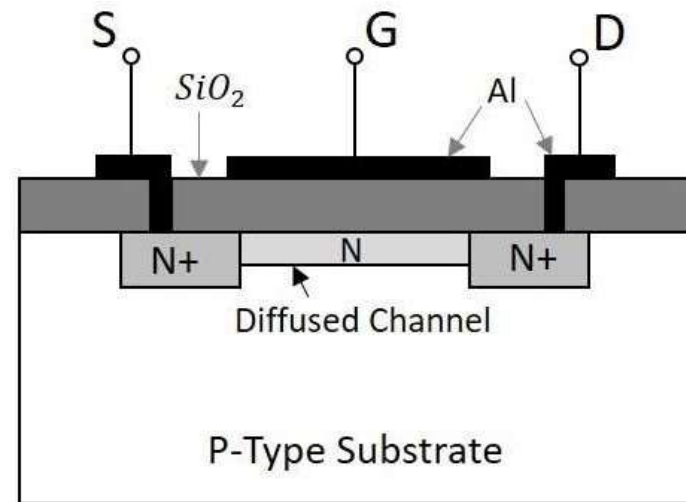


•D-MOSFETS can be operated in both the depletion mode and the enhancement mode. E MOSFETS are restricted to operate in enhancement mode. The primary difference between them is their physical construction.

Construction Of AN N-Channel MOSFET:

A thin layer of insulation silicon dioxide (SiO_2) is grown over the surface of the structure, and holes are cut into oxide layer, allowing contact with the source and drain. Then the gate metal area is overlaid on the oxide, covering the entire channel region. Metal contacts are made to drain and source and the contact to the metal over the channel area is the gate terminal.

The metal area of the gate, in conjunction with the insulating dielectric oxide layer and the semiconductor channel, forms a parallel plate capacitor. The insulating layer of SiO_2 is the reason why this device is called the insulated gate field effect transistor.

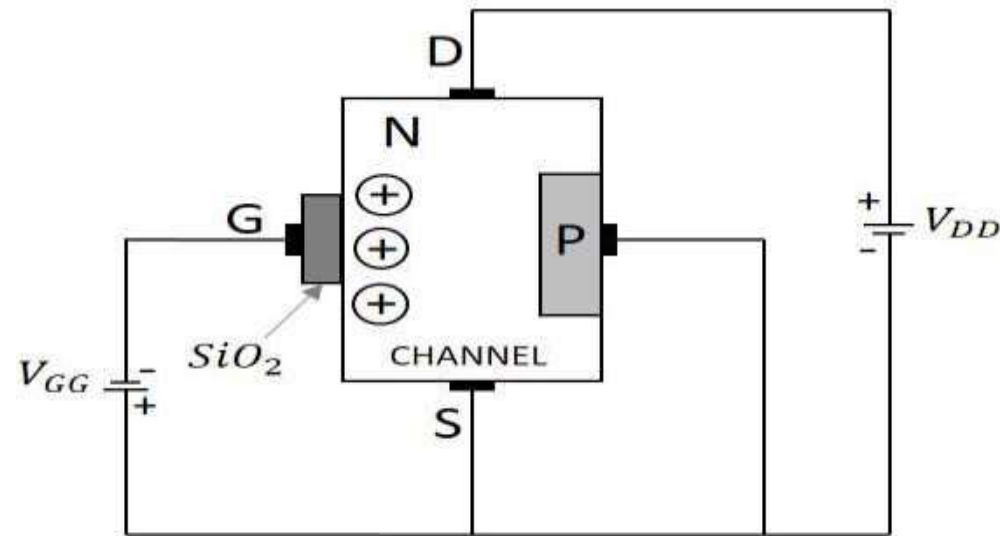


Structure of N-channel MOSFET

Working of N-Channel MOSFET-Depletion Mode

When no voltage is applied between gate and source, some current flows due to the voltage between drain and source. Let some negative voltage is applied at V_{GG} . Then the minority carriers i.e. holes, get attracted and settle near SiO_2 layer. But the majority carriers, i.e., electrons get repelled.

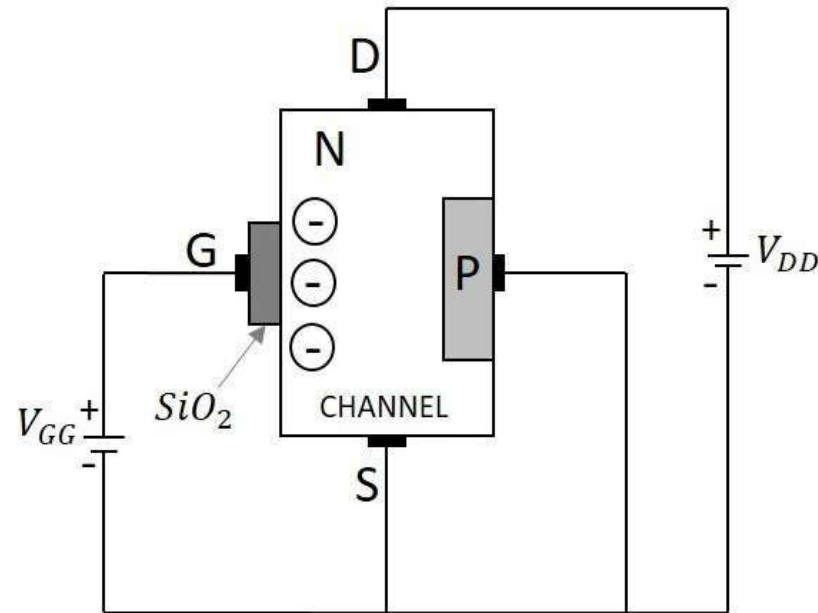
The channel nearer to drain gets more depleted than at source like in FET and the current flow decreases due to this effect. Hence it is called as depletion mode MOSFET.



Working of MOSFET in depletion mode

Working of N-Channel MOSFET-Enhancement Mode

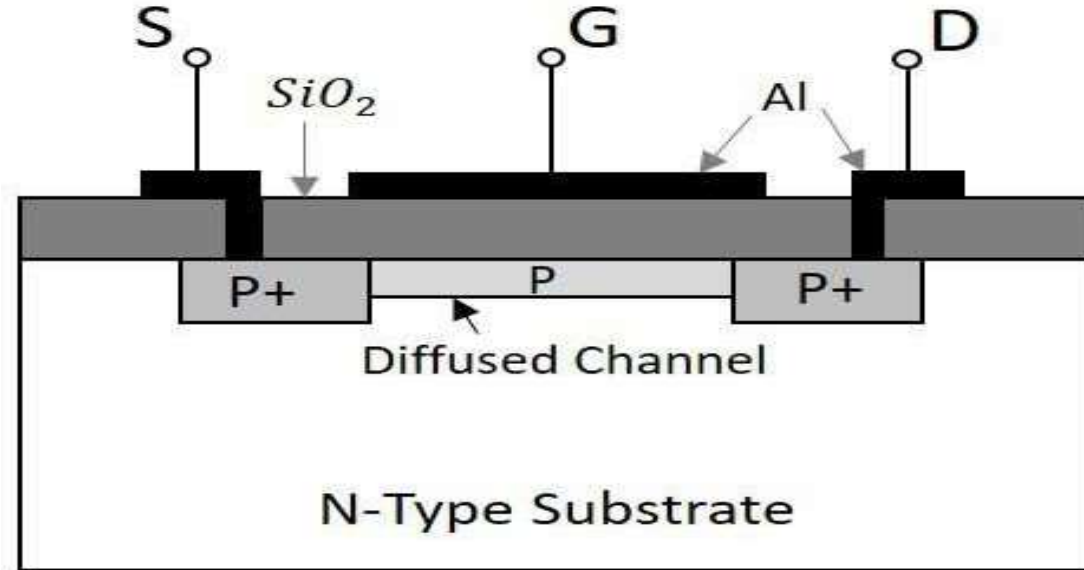
When this positive potential is further increased, the current I_D increases due to the flow of electrons from source and these are pushed further due to the voltage applied at V_{GG} . Hence the more positive the applied V_{GG} , the more the value of drain current I_D will be. The current flow gets enhanced due to the increase in electron flow better than in depletion mode. Hence this mode is termed as **Enhanced Mode MOSFET**.



Working of MOSFET in Enhancement mode

Construction of an P-Channel MOSFET

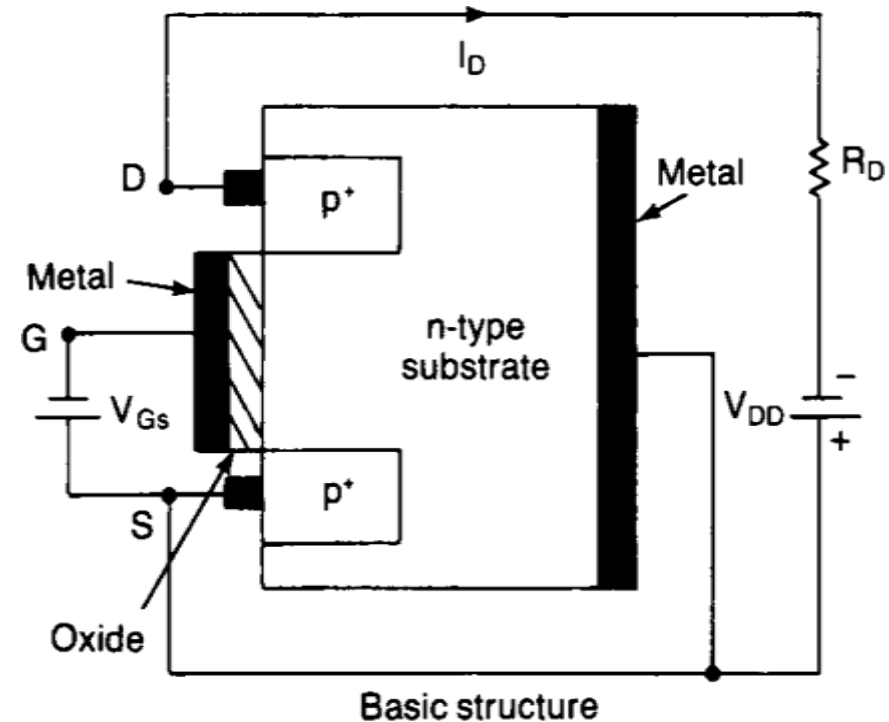
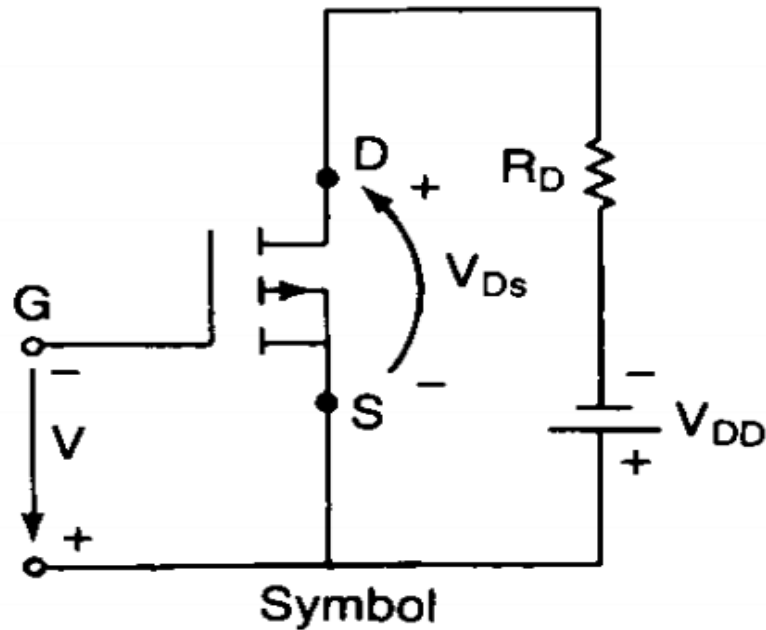
The construction and working of a PMOS is same as NMOS. A lightly doped **n-substrate** is taken into which two heavily doped **P+ regions** are diffused. These two P+ regions act as source and drain. A thin layer of **SiO₂** is grown over the surface. Holes are cut through this layer to make contacts with P+ regions, as shown in the following figure.



Structure of P-channel MOSFET

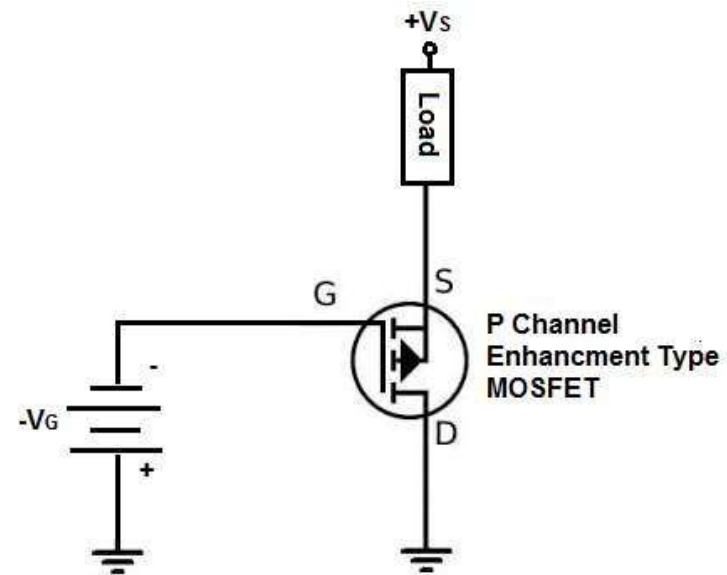
Working of P-Channel MOSFET-Depletion Mode

In p channel depletion MOSFET channel is already created between drain and source. As we know for MOSFET gate terminal has a thin [metal oxide layer](#). Generally, SiO_2 used as this metal oxide layer. Now we apply the negative voltage at the gate terminal. Due to the capacitive effects electrons are repelled each other and shifted to the p channel. In this way here depletion occurs so it named depletion MOSFET. As per figure you can see we applied negative gate voltage and positive drain voltage. This is the basic working of p channel depletion MOSFET.

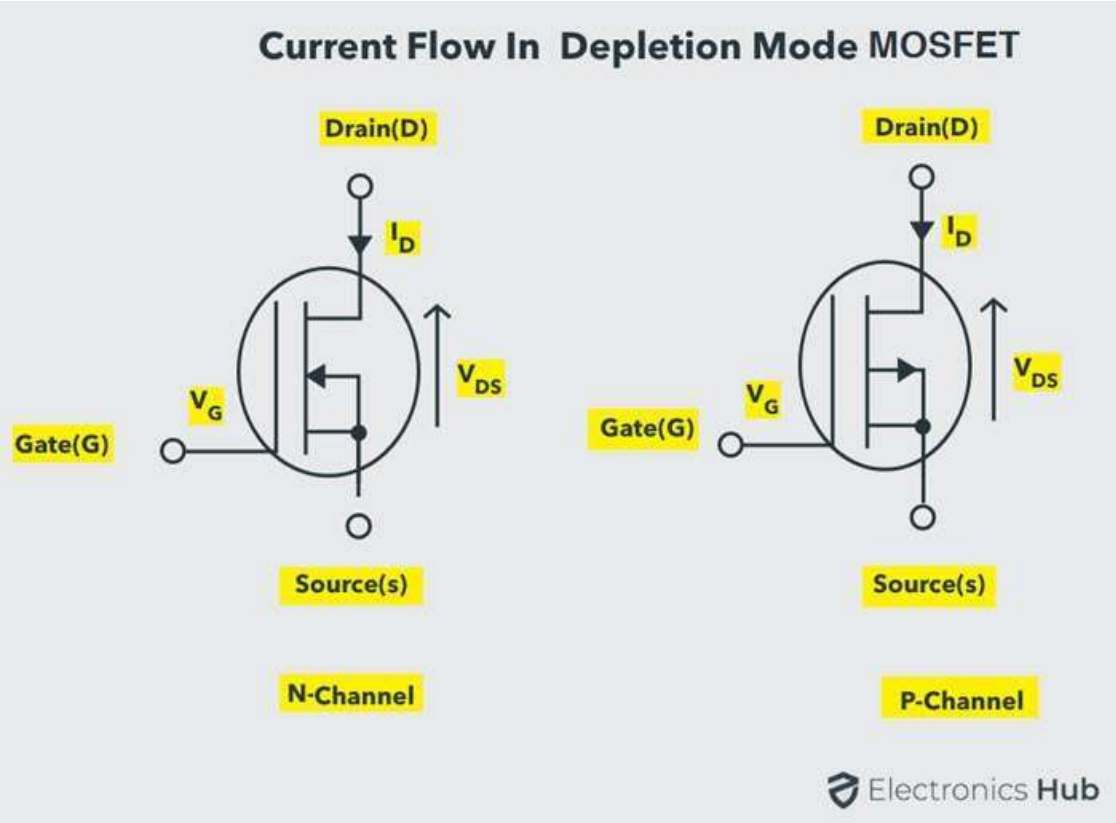
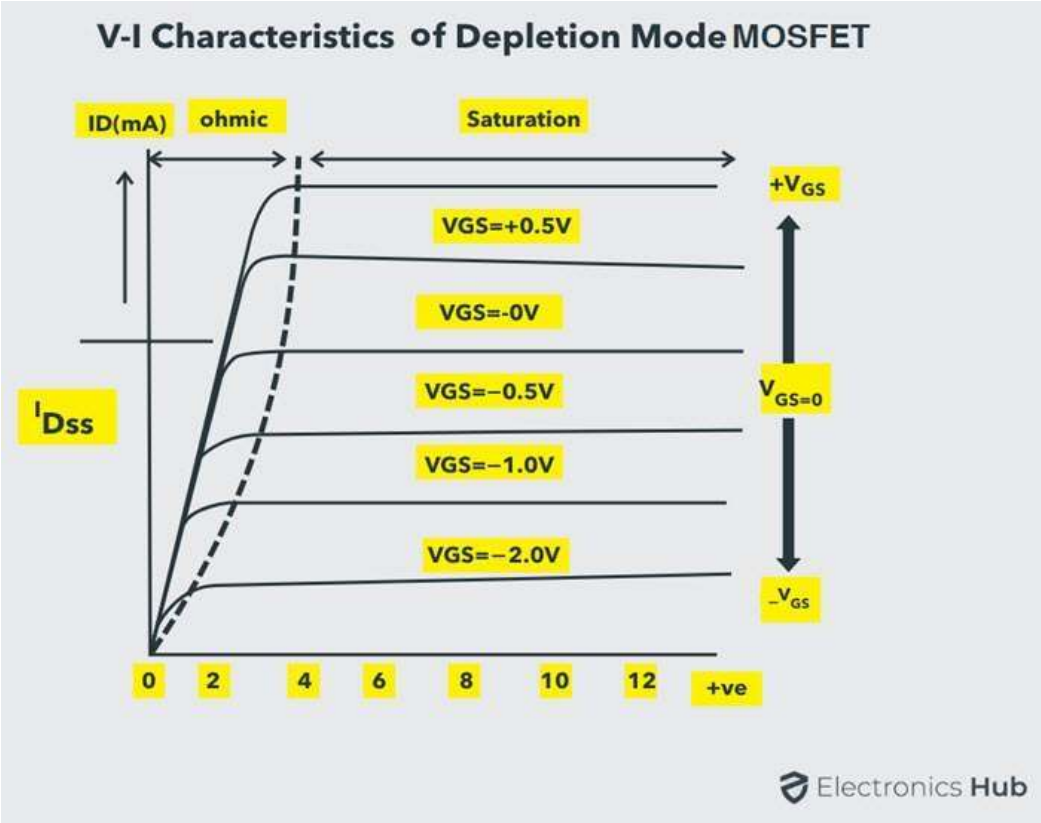


Working of P-Channel MOSFET-Enhancement Mode

- To turn on a P-Channel Enhancement-type MOSFET, apply a positive voltage V_S to the source of the MOSFET and apply a negative voltage to the gate terminal of the MOSFET (the gate must be sufficiently more negative than the threshold voltage across the drain-source region ($V_G < v_{DS}$)). This will allow a current to flow through the source-drain channel.
- To turn off a P-channel enhancement type MOSFET, there are 2 steps you can take. You can either cut off the bias positive voltage, V_S , that powers the source. Or you can turn off the negative voltage going to the gate of the transistor.

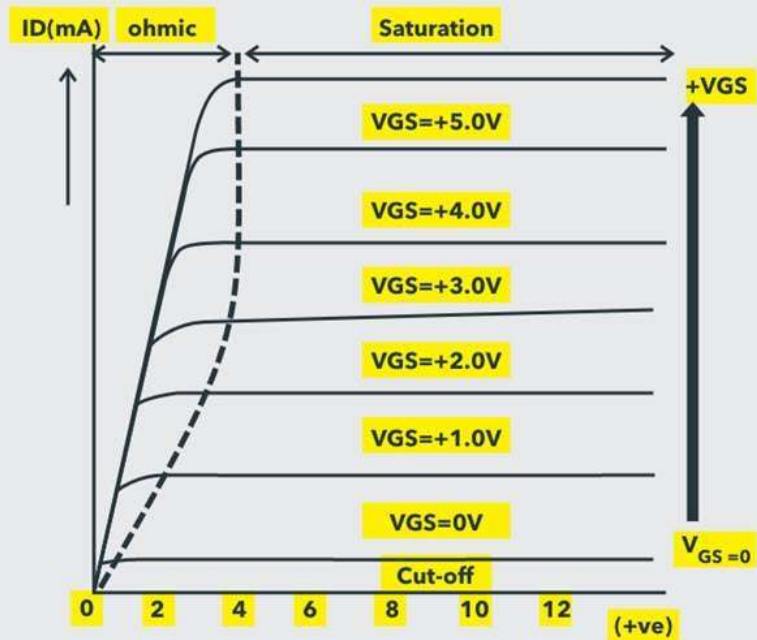


Depletion Mode MOSFET



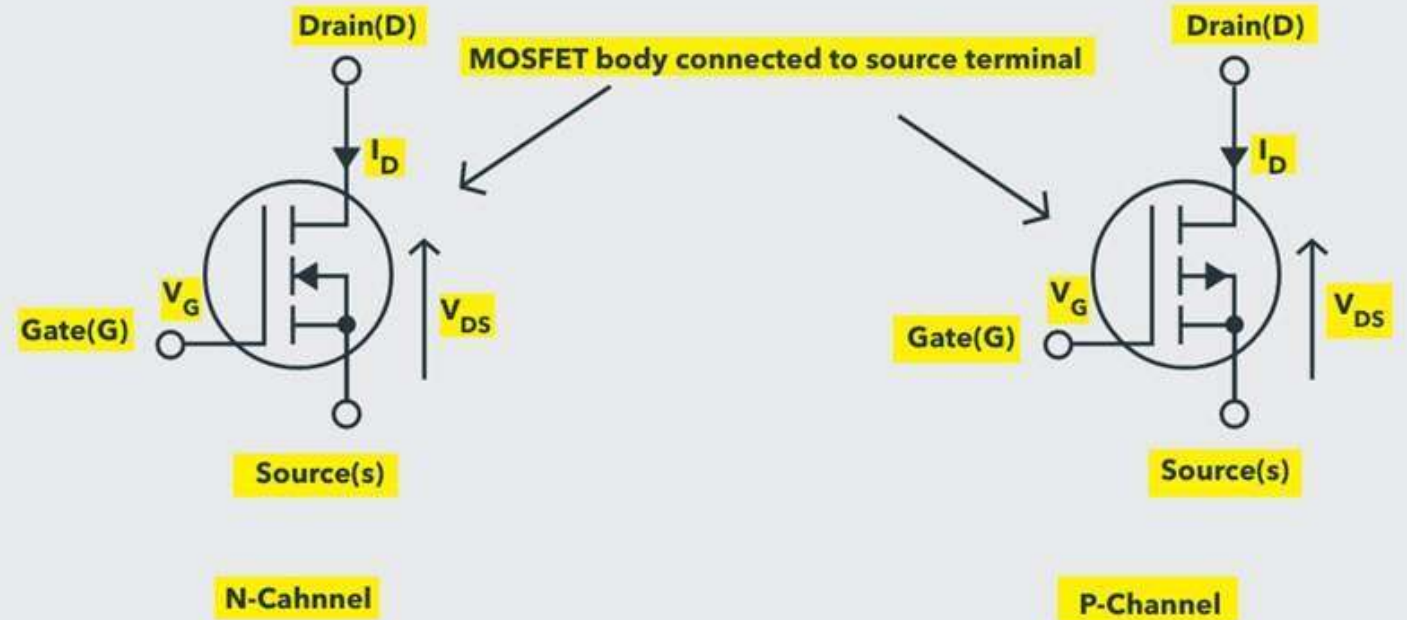
Enhancement Mode MOSFET

V-I Characteristics Of Enhancement Mode MOSFET



Electronics Hub

Current Flow In Enhancement Mode MOSFET

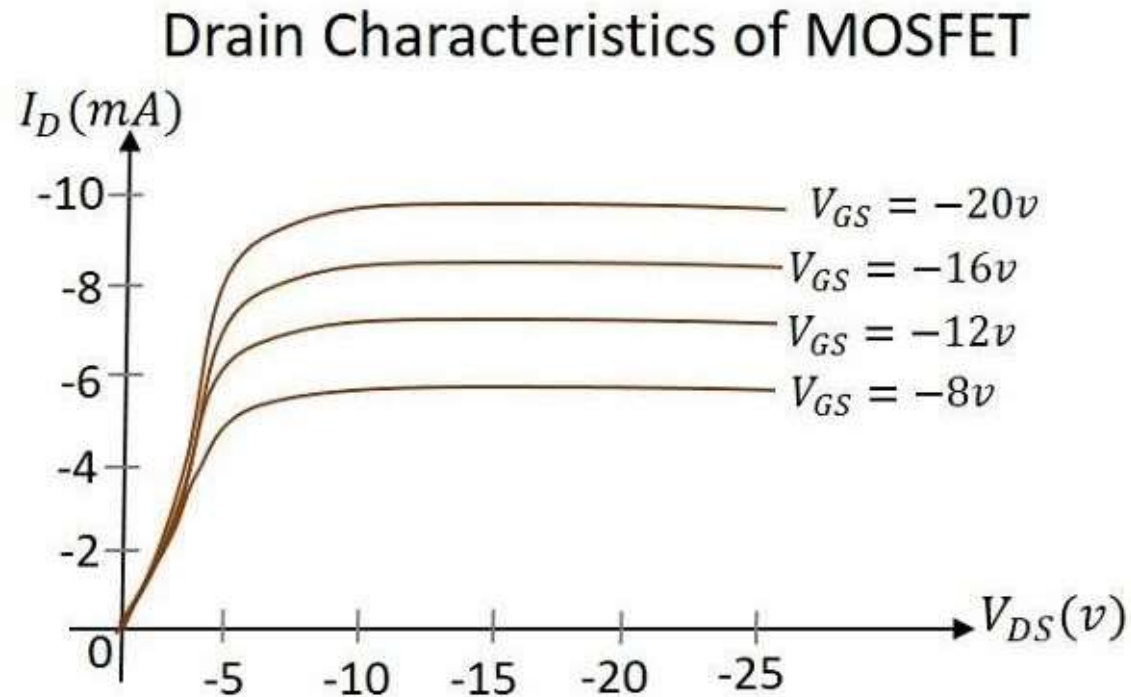


Electronics Hub

Drain Characteristics Of MOSFET

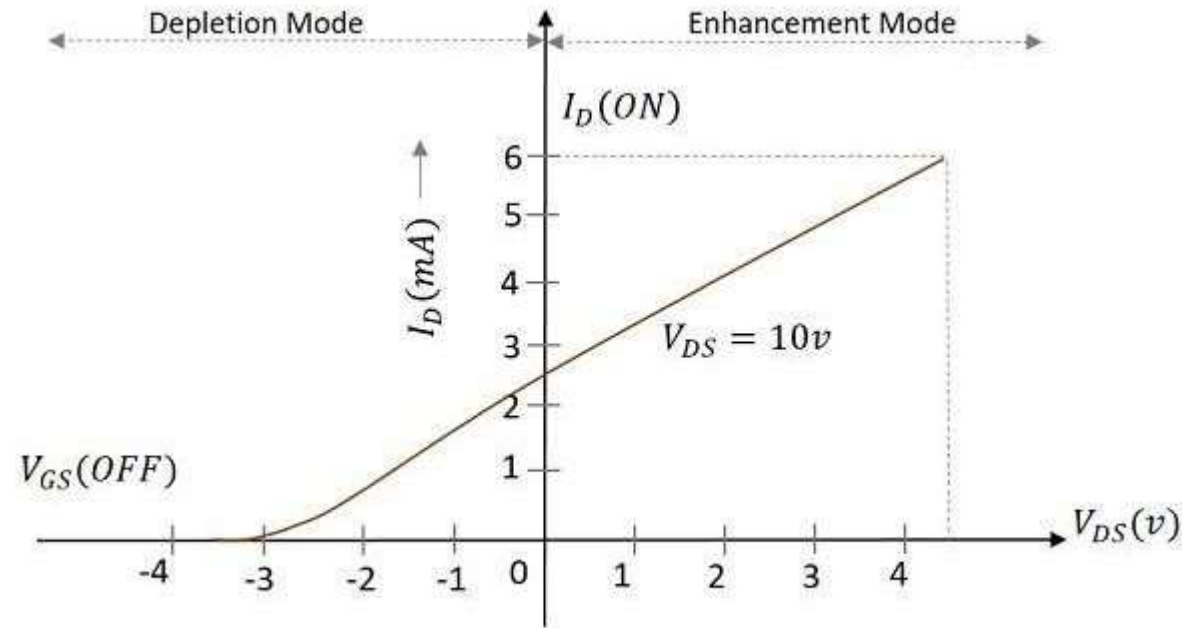
The drain characteristics of a MOSFET are drawn between the drain current I_D and the drain source voltage V_{DS} . The characteristic curve is as shown below for different values of inputs.

Actually when V_{DS} is increased, the drain current I_D should increase, but due to the applied V_{GS} , the drain current is controlled at certain level. Hence the gate current controls the output drain current.



Transfer Characteristics Of MOSFET

Transfer characteristics define the change in the value of V_{DS} with the change in I_D and V_{GS} in both depletion and enhancement modes. The below transfer characteristic curve is drawn for drain current versus gate to source voltage.



Transfer Characteristics of a MOSFET

Application Of MOSFET



- One of the primary contributions to electronics made by MOSFETs can be found in the area of digital (computer electronics). The signals in digital circuits are made up of rapidly switching dc levels. This signal is called as a rectangular wave ,made up of two dc levels (or logic levels). These logic levels are 0V and +5V.
- A group of circuits with similar circuitry and operating characteristics is referred to as a logic family. All the circuits in a given logic family respond to the same logic levels, have similar speed and power handling capabilities , and can be directly connected together. One such logic family is complementary MOS (or CMOS) logic. This logic family is made up entirely of MOSFETs.
- MOSFET amplifiers are extensively used in radio frequency applications.
- DC motors can be regulated by power MOSFETs.
- It acts as a passive element like resistor, capacitor and inductor.
- High switching speed of MOSFETs make it an ideal choice in designing chopper circuits.

Comparison Of Mosfet With JFET



- In enhancement and depletion types of MOSFET, the transverse electric field induced across an insulating layer deposited on the semiconductor material controls the conductivity of the channel.
- In the JFET the transverse electric field across the reverse biased PN junction controls the conductivity of the channel.
- The gate leakage current in a MOSFET is of the order of 10^{-12} A. Hence the input resistance of a MOSFET is very high in the order of 10^{10} to $10^{15} \Omega$. The gate leakage current of a JFET is of the order of 10^{-9} A., and its input resistance is of the order of $10^8 \Omega$.
- The output characteristics of the JFET are flatter than those of the MOSFET, and hence the drain resistance of a JFET (0.1 to $1 \text{M}\Omega$) is much higher than that of a MOSFET (1 to $50 \text{k}\Omega$).
- JFETs are operated only in the depletion mode. The depletion type MOSFET may be operated in both depletion and enhancement mode.
- Comparing to JFET, MOSFETs are easier to fabricate.
- Special digital CMOS circuits are available which involve near zero power dissipation and very low voltage and current requirements. This makes them suitable for portable systems.

FET Amplifiers

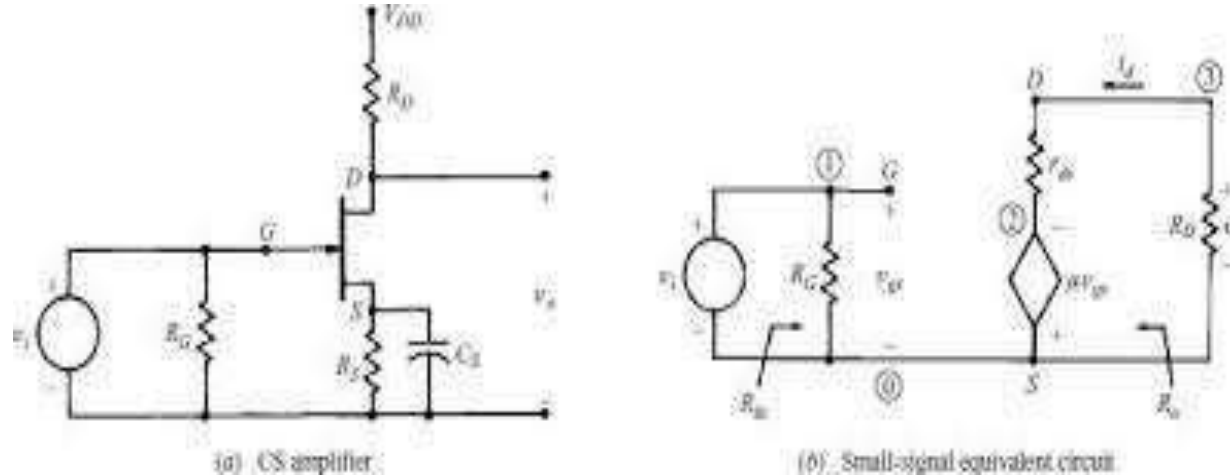


INTRODUCTION

- Field-effect transistors are three-terminal devices, but in contrast with the bipolar transistor, it is the voltage across two terminals that controls the current flowing in the third terminal. The three terminals in an FET are the *drain*, *source* and *gate*.
- Field Effect Transistor (FET) amplifiers provide an excellent voltage gain and high input impedance. Because of high input impedance and other characteristics of JFETs they are preferred over BJTs for certain types of applications.
- There are 3 basic FET circuit configurations:
 - i) Common Source
 - ii) Common Drain
 - iii) Common Gain

Common Source (CS) Amplifier

A simple Common Source amplifier is shown in Fig. and associated small signal equivalent circuit using voltage-source model of FET is shown in Fig.



•Voltage Gain

Source resistance (R_S) is used to set the Q-Point but is bypassed by CS for mid-frequency operation. From the small signal equivalent circuit, the output voltage

$$V_O = -R_D \mu V_{gs} (R_D + r_d)$$

Where $V_{gs} = V_i$, the input voltage,

Hence, the voltage gain, $AV = V_O / V_i = -R_D \mu / (R_D + r_d)$

•Input Impedance

From Input Impedance is $Z_i = R_G$

For voltage divider bias as in CE Amplifiers of BJT

$$R_G = R_1 \parallel R_2$$

•**Output Impedance:** Output impedance is the impedance measured at the output terminals with the input voltage $V_i = 0$

From the above when the input voltage $V_i = 0$, $V_{gs} = 0$ and hence $\mu V_{gs} = 0$

The equivalent circuit for calculating output impedance is given in Fig. Output impedance $Z_o = r_d \parallel R_D$

Normally r_d will be far greater than R_D . Hence $Z_o \approx R_D$

Common Drain (CD) Amplifier

- A simple common drain amplifier is shown in Fig and associated small signal equivalent circuit using the voltage source model of FET is shown in Fig . Since voltage V_{gd} is more easily determined than V_{gs} , the voltage source in the output circuit is expressed in terms of V_{gs} and Thevenin's theorem

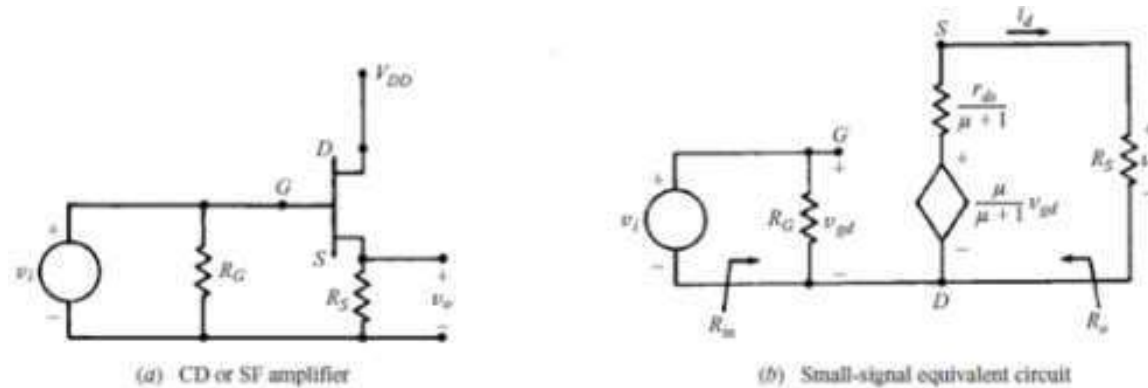


Fig. 7.2 (a)CD Amplifier (b)Small-signal equivalent circuit

- Voltage Gain**

The output voltage,

$$V_O = R_S \mu V_{gd} / (\mu + 1) R_S + r_d$$

Where $V_{gd} = V_i$ the input voltage.

Hence, the voltage gain, $A_v = V_O / V_i = R_S \mu / (\mu + 1) R_S + r_d$.

- **Input Impedance**

From Fig , Input Impedance $Z_i = R_G$

- **Output Impedance**

From Fig , Output impedance measured at the output terminals with input voltage $V_i = 0$ can be calculated from the following equivalent circuit.

As $V_i = 0$: $V_{gd} = 0$: $\mu v_{gd} / (\mu + 1) = 0$ Output Impedance

$$Z_O = r_d / (\mu + 1) \parallel R_S$$

When $\mu \gg 1$

$$Z_O = (r_d / \mu) \parallel R_S = (1/g_m) \parallel R_S.$$

INTRODUCTION

- DIGITAL OPERATION OF A SYSTEM
- Digital systems are designed to store, process and communication information in digital form.
- A computer manipulates information in digital, are more precisely, binary form
- A binary number has only two discrete values zero or one.
- Each of these discrete values is represented by the OFF and ON status of electronic switch called transistor.

Why are binary number important

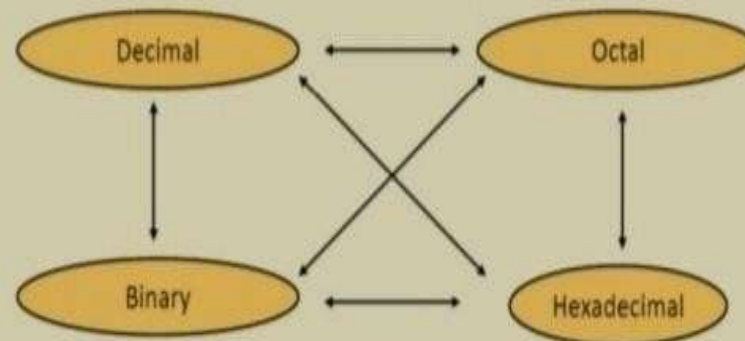
- We concentrate on binary numbers in this course
- Some convention followed:
 - **Bit**:single binary digit(0,1)
 - **Nibble**:collection of four bits
 - **Byte**:collection of eight bits
 - **Word**:collection of sixteen bits

Number system

• Decimal	Binary	Octal	Hexa decimal
• 0.	0	0.	0
• 1.	1.	1.	1
• 2.	10.	2.	2
• 3.	11.	3.	3
• 4.	100.	4.	4
• 5.	101.	5.	5
• 6	110.	6.	6
• 7.	111.	7	7
• 8	1000	10	8
• 9	1001	11.	9

Conversion Among Bases

- The possibilities:



Exercise – Convert ...

Decimal	Binary	Octal	Hexa- decimal
29.8			
	101.1101		
		3.07	
			C.82

Negative numbers

1. Sign and Magnitude Representation
2. 1's Complement Representation
3. 2's Complement Representation

Goal of negative number systems

- **Signed system:** Simple. Just flip the sign bit
 - 0 = positive
 - 1 = negative
- **One's complement:** Replace subtraction with addition
 - Easy to derive (Just flip every bit)
- **Two's complement:** Replace subtraction with addition
 - Addition of one's complement and one produces the two's complement.

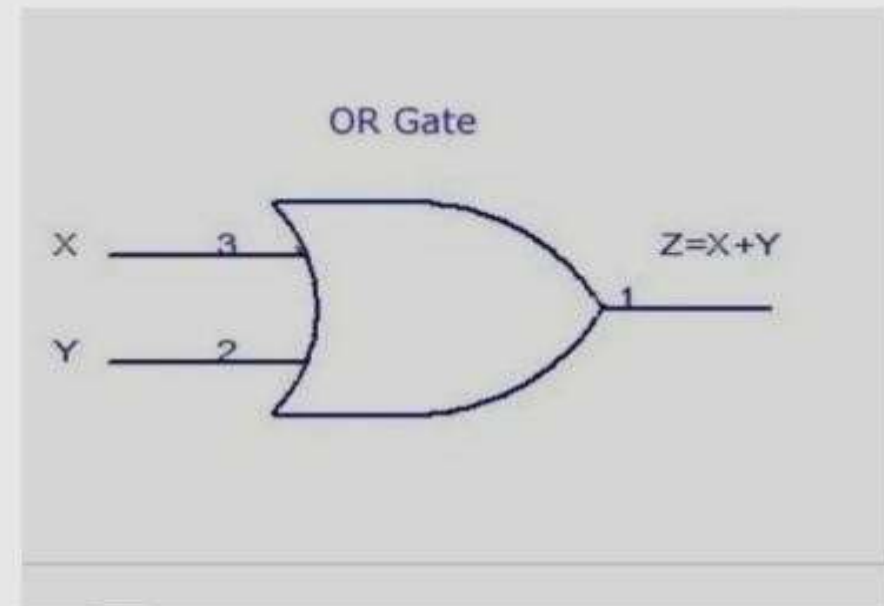
- 1's complement:
 - Formula: $2^n - 1 - x$
 - i.e. $n=4$, $2^4 - 1 - x = 15 - x$
 - In binary: $(1\ 1\ 1\ 1) - (b_3\ b_2\ b_1\ b_0)$
 - Just flip all the bits.
- 2's complement:
 - Formula: $2^n - x$
 - i.e. $n=4$, $2^4 - x = 16 - x$
 - In binary: $(1\ 0\ 0\ 0\ 0) - (0\ b_3\ b_2\ b_1\ b_0)$
 - Just flip all the bits and add 1.

OR Gate

- Or gate is a digital logic gate that gives an output of 1 when any of its inputs are 1, otherwise 0

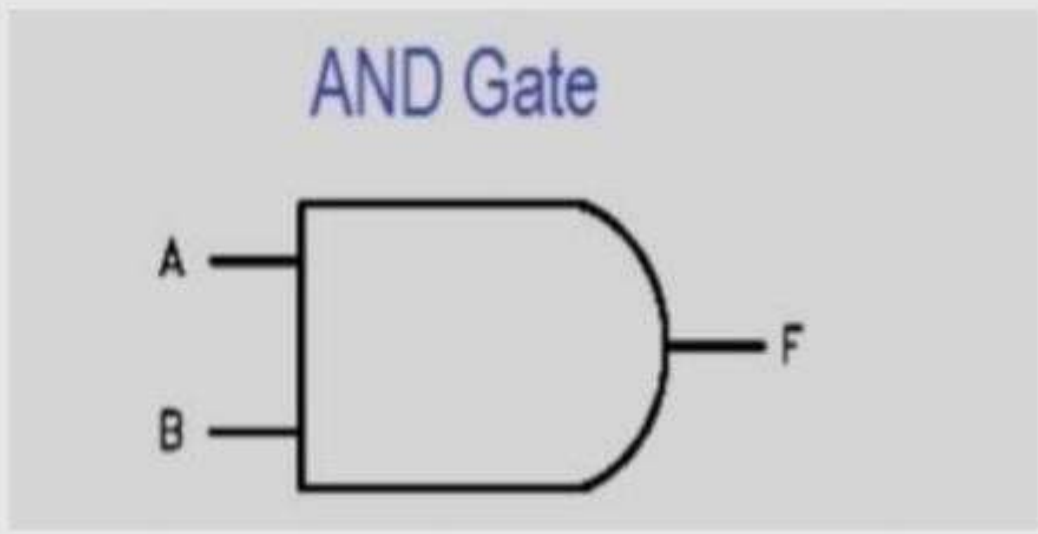
Truth table

Two Input OR gate		
A	B	$Y = A + B$
0	0	0
0	1	1
1	0	1
1	1	1



AND Gate

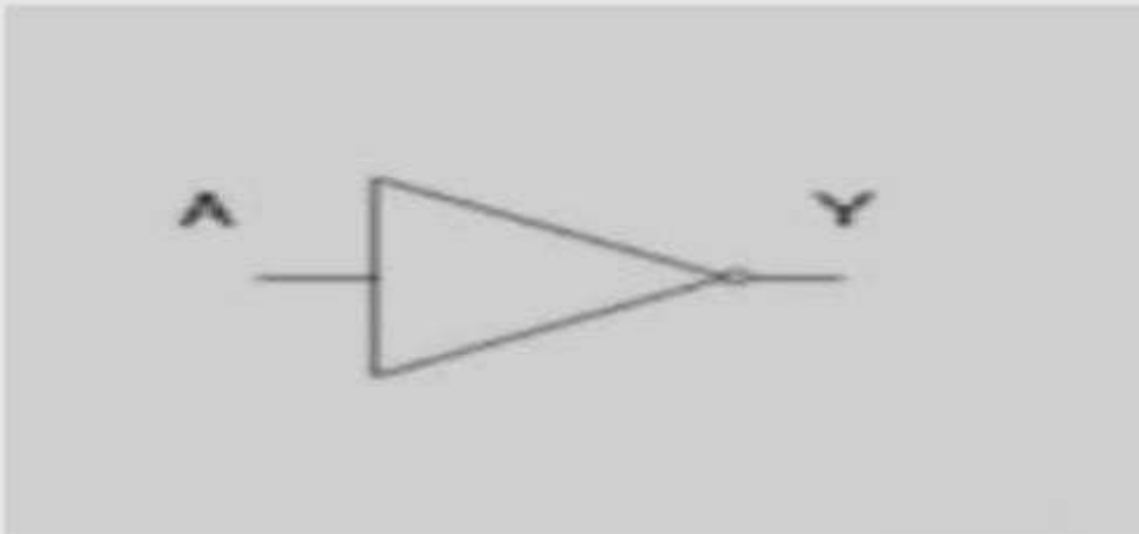
- The and gate is a basic digital logic gate that implement logical conjunction from mathematical logic.



INPUT		OUTPUT
A	B	F
0	0	0
0	1	0
1	0	0
1	1	1

NOT Gate

- In digital logic an inveter or NOT gate is a logic gate which implement logical negation.



Truth Table	
A	A'
1	0
0	1

EXCLUSIVE OR GATE

It is a digital logic gate that results in true either 1 or high. Output when the number of true inputs is an odd count.



(a) Circuit symbol

A	B	C
0	0	0
0	1	1
1	0	1
1	1	0

(b) Truth table

$$C = A \oplus B$$

(c) Boolean expression

- The output of a two input XOR gate assumes the 1 state if one and only input assumes the 1 state
- The boolean notation for the XOR is

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

$$Y = A\overline{B} + B\overline{A}$$

DEMORGANS LAW

- The statement “if and only if all inputs are true(1), then the output is true(1)” is logically equivalent to the statement “If at least one input is false(0), then the output is false(0)”
- In boolean notation this equivalence can be written as

$$ABC \dots = \overline{\bar{A} + \bar{B} + \bar{C} + \dots}$$

or

$$\overline{ABC \dots} = \bar{A} + \bar{B} + \bar{C} + \dots$$

and its dual

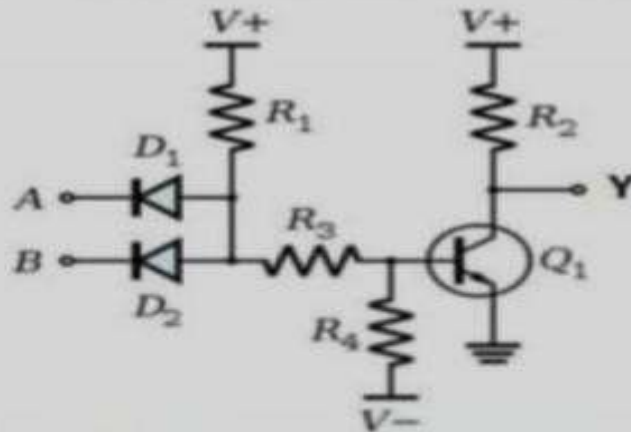
$$\overline{\bar{A} + \bar{B} + \bar{C} + \dots} = \bar{A}\bar{B}\bar{C} \dots$$

are known as De Morgan's laws

LOGIC FAMILIES

- Diode transistor

- DTL was first commercial available IC logic family in 53/73 series.
- The basic circuit in the DTL logic is the NAND gate.
- Each input associated with one diode.
- The diode and resistor form an AND gate.
- The transistor services as a NOT gate

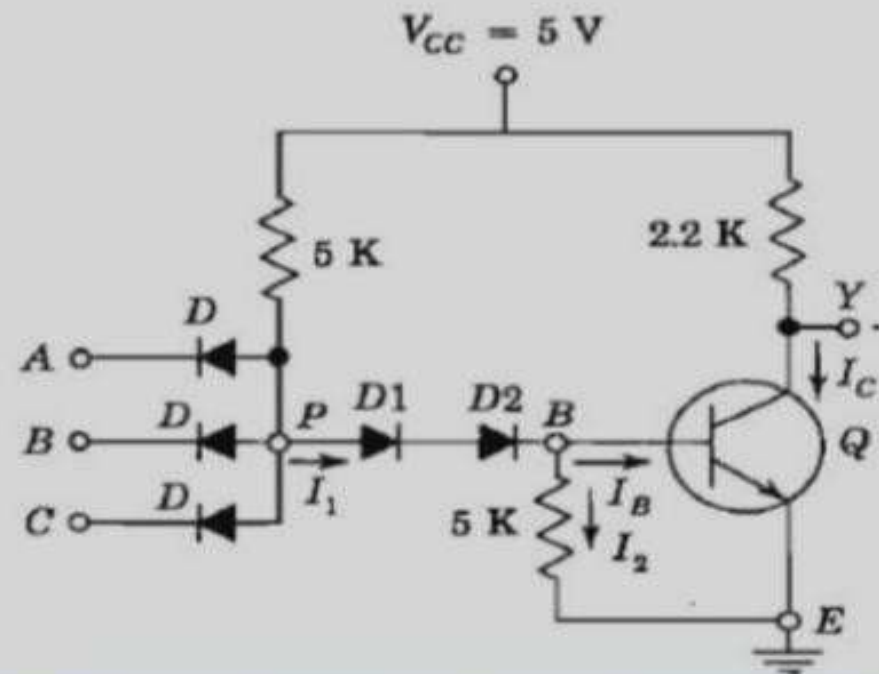


A	B	$Y = \overline{A \cdot B}$
0	0	1
0	1	1
1	0	1
1	1	0

MODIFIED DTL GATES

- Most logic gates are fabricated as an integrated circuit. In this all the transistors, diodes, resistors and capacitors in a fairly complicated circuit may be shaped within a tiny chip of single crystal silicon
- It turns out that large values of resistance and capacitance can not be fabricated economically
- In view of these facts the NAND gate is modified for integrated circuit implementation by eliminating the capacitor, reducing the resistance values drastically and using diodes or transistors to replace resistors wherever possible

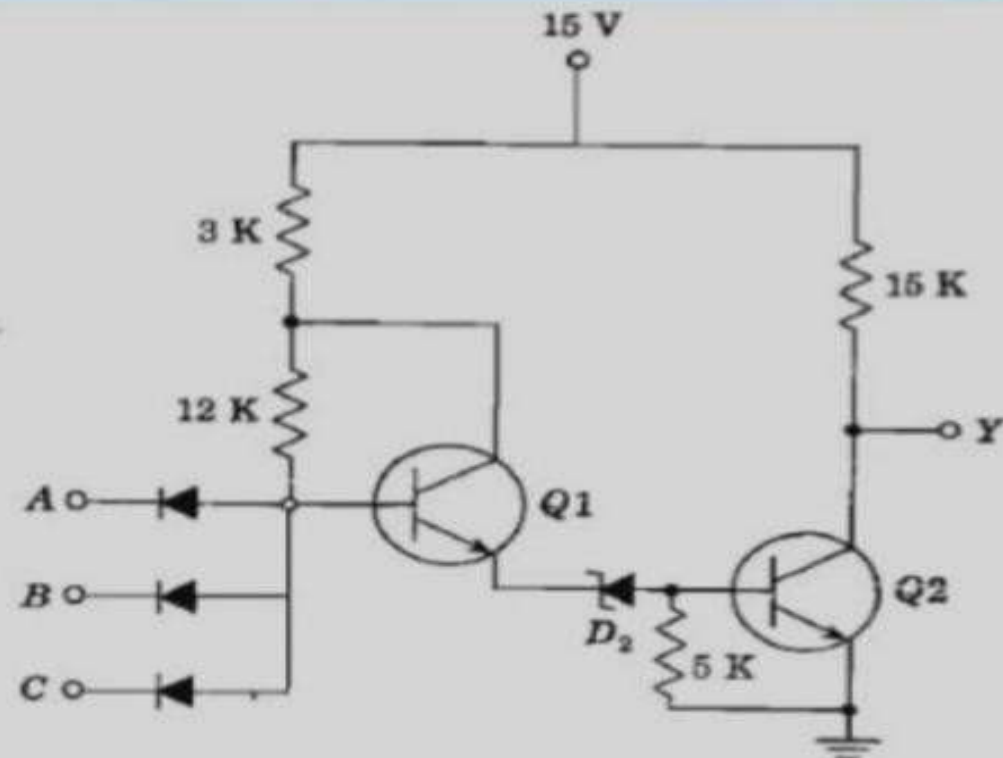
An integrated or Modified positive DTL NAND gate



HIGH THRESHOULD LOGIC

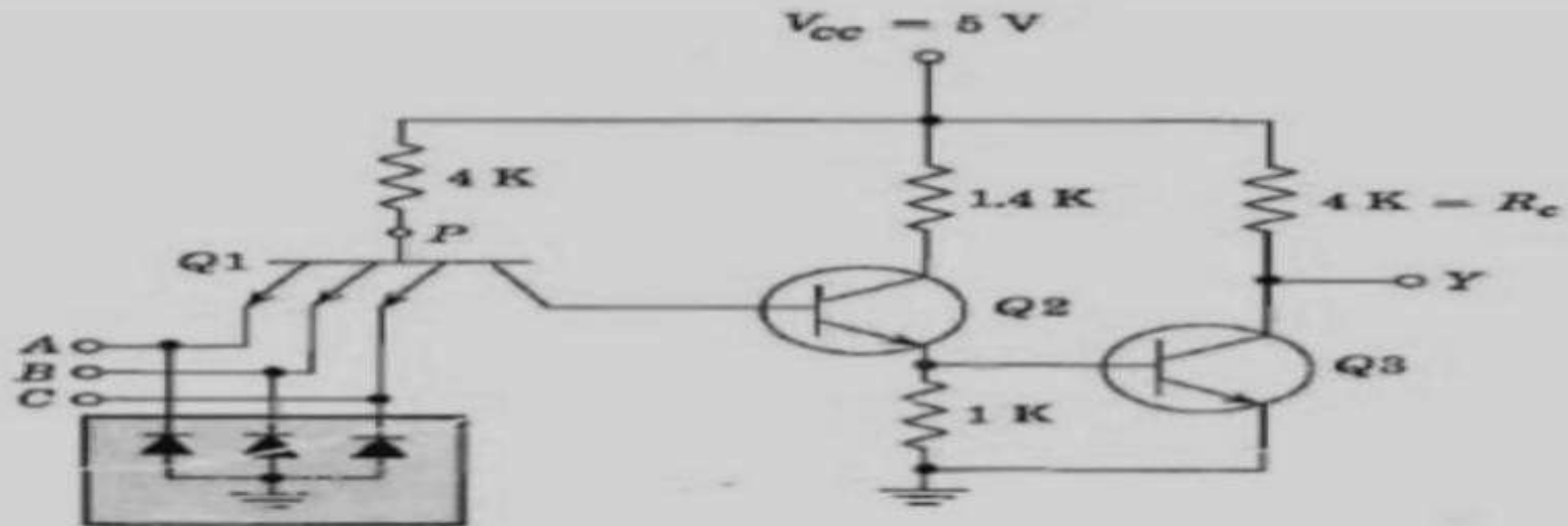
➤ In an Industrial environment the noise level is quite high because of the presence of motors, high voltage switches, on – off control circuitsetc

➤ By using a Zener diode in the place of diode D2 in the DTL gate this circuit is converted to the high noise immunity gate



TRANSISTOR TRANSISTOR LOGIC

- TTL family is modification to the DTL it has come to existence so has to overcome speed limitation of DTL family



TTL WORKING

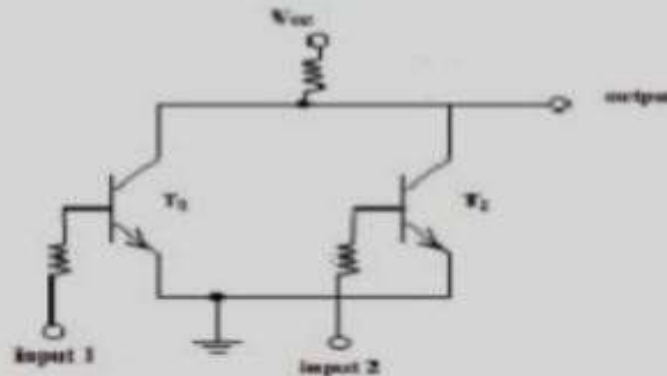
The working of this circuit is identical to that of DTL circuit.

Case1- When at least one input is logic LOW, transistor Q2 and Q3 are in cut-off and hence, output of Q3 is HIGH.

Case2- When all inputs are HIGH, Q1 operates in active inverse mode, driving Q2 & Q3 in saturation. Since Q3 is ON, the output is LOW.

Case3- While all inputs are HIGH, if any of the inputs suddenly goes LOW, then Q2 and Q3 will be turned off only when stored base charge is removed. The collector-base junction of Q1 is back-biased and Q1 operates in normal active region. A large collector current of Q1 is in such direction that it helps removing base charge of Q2 and Q3. In this way, the circuit speed is increased in TTL over speed of DTL.

RESISTOR TRANSISTOR LOGIC



Truth Table

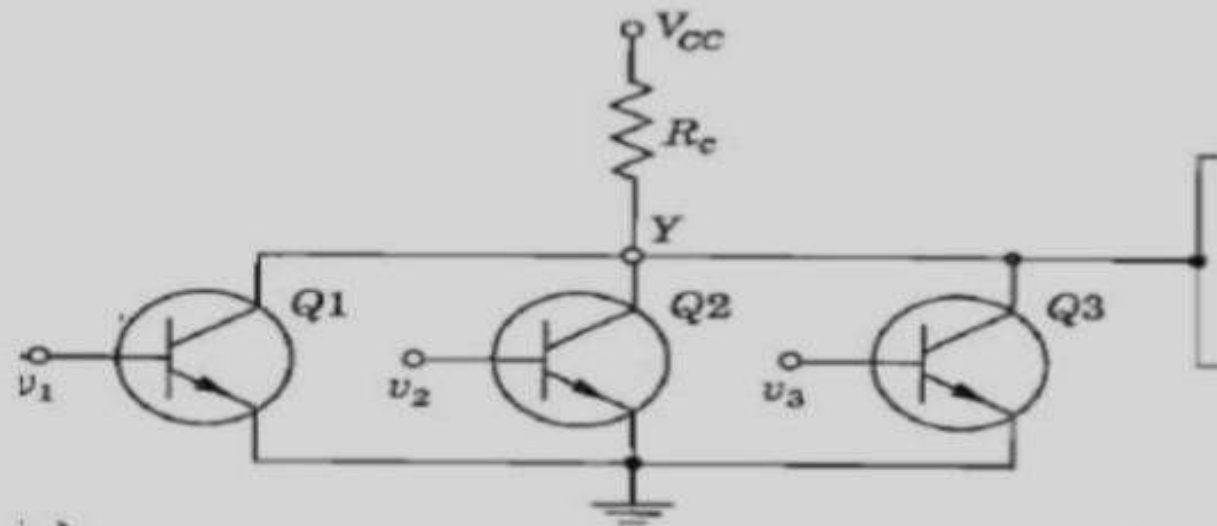
A	B	$Y = \overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

If any input is high the corresponding transistor is driven into saturation and the output goes low, regardless of the states of the other transistor.

If all inputs are low then all transistor are in cutoff state and the output of the circuit goes high.

DIRECT COUPLED TRANSISTOR LOGIC

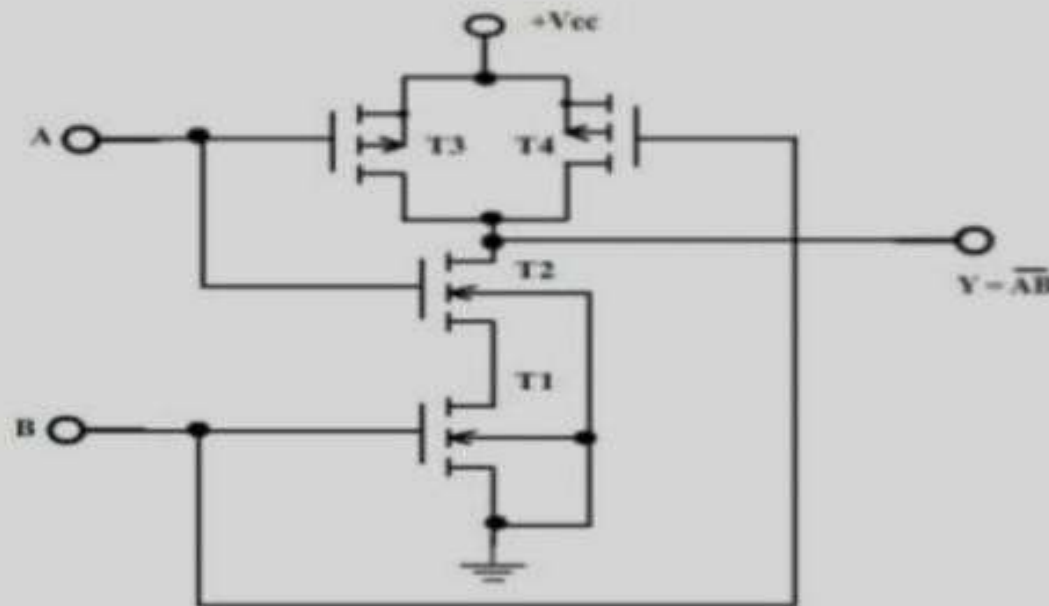
➤ DCTL configuration is the same as RTL except that the base resistors are omitted



A positive NOR DTCL gate with a Fan – in of 3 and Fan – out of 2

CMOS LOGIC FAMILY

CMOS NAND gate

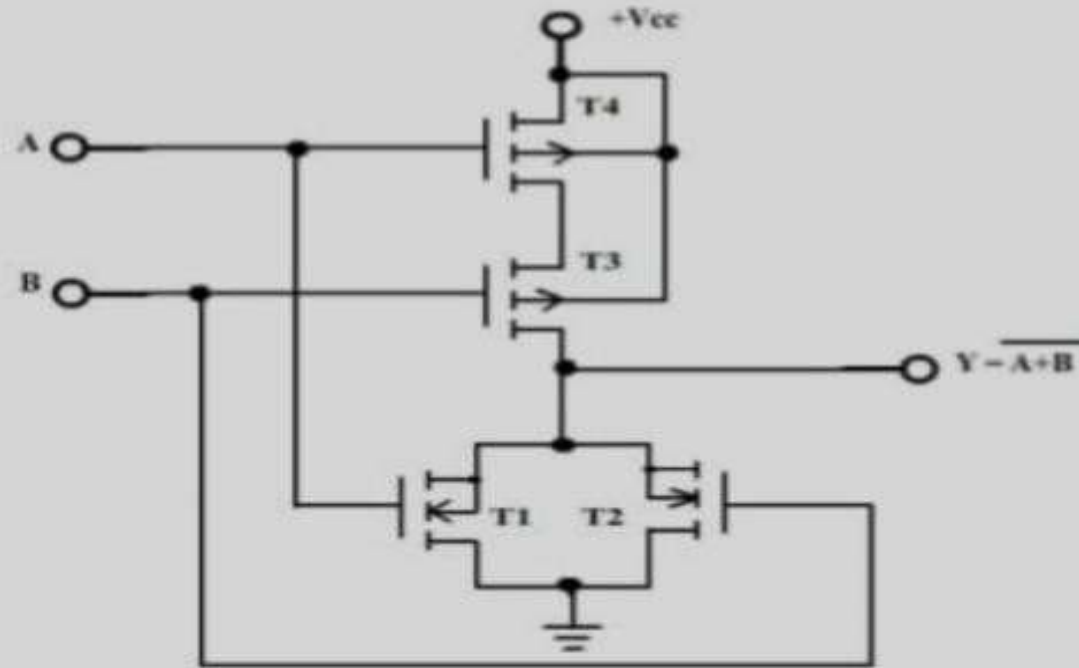


➤ T1 and T2 are n-channel MOSFETs while T3 and T4 are p-channel MOSFETs.

➤ When both inputs A & B are HIGH, then T1 & T2 are ON while T3 & T4 are OFF. Hence, output is connected to GND i.e. LOW.

➤ If either input is LOW, then either T3 or T4 is ON, connecting output is +Vcc i.e. HIGH.

CMOS NOR gate



Similar is working of CMOS NOR gate shown in figure aside. Here, p-channel devices are in series and n-channel devices are in parallel.

COMPARISATION OF LOGIC FAMILY

	Parameter	RTL	PL	DTL	HTL	TTL	ECL	MOS	CMOS
1.	Basic Gate	NOR	NOR	NAND	NAND	NAND	OR-NOR	NAND	NOR or NAND
2.	Fan-out	5	Depends on injector current	8	10	10 to 20	25	20	20 to 50
3.	Power dissipation in mW	12	6mm to 70 μ M	8-12	55	10	40-55	0.2-10	0.0025
4.	Noise immunity	Nominal	Poor	Good	Excellent	Very Good	Poor	Good	Very Good
5.	Propagation delay (in sec.)	12	25-250	30	90	10	0.75	300	70.0
6.	Clock rate (MHZ)	8	-	72	4	35	>60	2	10
7.	Available functions	High	LSI only	Fairly high	Nominal	Very high	High	low	High



UNIT-4



INTRODUCTION TO BOOLEAN ALGEBRA

BOOLEAN ALGEBRA

- Also known as Switching Algebra
- › Invented by mathematician George Boole in 1849
- › Used by Claude Shannon at Bell Labs in 1938
 - To describe digital circuits built from relays
- Digital circuit design is based on
 - › Boolean Algebra
 - Attributes
 - Postulates
 - Theorems
 - › These allow minimization and manipulation of logic gates for optimizing digital circuits

BOOLEAN ALGEBRA ATTRIBUTES

- Binary
 - › A1a: $X=0$ if $X=1$ /
 - › A1b: $X=1$ if $X=0$ /
- Complement
 - › aka *invert, NOT*
 - › A2a: if $X=0$, $X'=1$
 - › A2b: if $X=1$, $X'=0$
 - The tick mark ' means

X	X'
0	1
1	0

complement, invert, or NOT
 -Other symbol for
 complement: $X' = \bar{X}$

- AND operation
 - › A3a: $0 \bullet 0 = 0$
 - › A4a: $1 \bullet 1 = 1$
 - › A5a: $0 \bullet 1 = 1 \bullet 0 = 0$
 - The dot \bullet means AND
 - Other symbol for AND:
 $X \bullet Y = XY$ (*no symbol*)

X	Y	$X \bullet Y$
0	0	0
0	1	0
1	0	0
1	1	1

- OR Operation
 - › A3b: $1+1=1$
 - › A4b: $0+0=0$
 - › A5b: $1+0=0+1=1$
 - The plus $+$ means OR

X	Y	$X+Y$
0	0	0
0	1	1
1	0	1
1	1	1

BOOLEAN ALGEBRA ATTRIBUTES

- **Variable:** Variables are the different symbols in a Boolean expression
- **Literal:** Each occurrence of a variable or its complement is called a literal
- **Term:** A term is the expression formed by literals and operations at one level

$$\bar{A} + A.B + A.\bar{C} + \bar{A}.B.C$$

- A, B, C are three variables
- Eight Literals
- Expression has five terms including four AND terms and the OR term that combines the first-level AND terms.

BOOLEAN ALGEBRA POSTULATES

- Identity Elements
 - › P2a: $X+0=X$
 - › P2b: $X\cdot 1=X$
- Commutativity
 - › P3a: $X+Y=Y+X$
 - › P3b: $X\cdot Y=Y\cdot X$
- Complements
 - › P6a: $X+X'=1$
 - › P6b: $X\cdot X'=0$

OR operation

X	Y	$X+0$	$X+Y$	$Y+X$	X'	$X+X'$
0	0	0	0	0	1	1
0	1	0	1	1	1	1
1	0	1	1	1	0	1
1	1	1	1	1	0	1

AND operation

X	Y	$X\cdot 1$	$X\cdot Y$	$Y\cdot X$	X'	$X\cdot X'$
0	0	0	0	0	1	0
0	1	0	0	0	1	0
1	0	1	0	0	0	0
1	1	1	1	1	0	0

BOOLEAN ALGEBRA POSTULATES

- Associativity
 - › P4a: $(X+Y)+Z=X+(Y+Z)$
 - › P4b: $(X \cdot Y) \cdot Z=X \cdot (Y \cdot Z)$

X	Y	Z	X+Y	(X+Y)+Z	Y+Z	X+(Y+Z)	X•Y	(X•Y)•Z	Y•Z	X•(Y•Z)
0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	1	1	1	0	0	0	0
0	1	0	1	1	1	1	0	0	0	0
0	1	1	1	1	1	1	0	0	1	0
1	0	0	1	1	0	1	0	0	0	0
1	0	1	1	1	1	1	0	0	0	0
1	1	0	1	1	1	1	1	0	0	0
1	1	1	1	1	1	1	1	1	1	1

BOOLEAN ALGEBRA POSTULATES

- Distributivity
 - › P5a: $X+(Y \cdot Z) = (X+Y) \cdot (X+Z)$
 - › P5b: $X \cdot (Y+Z) = (X \cdot Y) + (X \cdot Z)$

X	Y	Z	X+Y	X+Z	$(X+Y) \cdot (X+Z)$	$Y \cdot Z$	$X+(Y \cdot Z)$	$X \cdot Y$	$X \cdot Z$	$X \cdot Y + X \cdot Z$	Y+Z	$X \cdot (Y+Z)$
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0	1	0
0	1	0	1	0	0	0	0	0	0	0	1	0
0	1	1	1	1	1	1	1	0	0	0	1	0
1	0	0	1	1	1	0	1	0	0	0	0	0
1	0	1	1	1	1	0	1	0	1	1	1	1
1	1	0	1	1	1	0	1	1	0	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1

BOOLEAN ALGEBRA THEOREMS

- Idempotency
 - › T1a: $X+X=X$
 - › T1b: $X\cdot X=X$
- Null elements
 - › T2a: $X+1=1$
 - › T2b: $X\cdot 0=0$
- Involution
 - › T3: $(X')'=X$

OR AND

X	Y	X+Y	X•Y	X+X	X•X	X+1	X•0	X'	X''
0	0	0	0	0	0	1	0	1	0
0	1	1	0	0	0	1	0	1	0
1	0	1	0	1	1	1	0	0	1
1	1	1	1	1	1	1	0	0	1

BOOLEAN ALGEBRA THEOREMS

- Absorption (aka *covering*)
 - › T4a: $X+(X\cdot Y)=X$
 - › T4b: $X\cdot(X+Y)=X$
 - › T5a: $X+(X'\cdot Y)=X+Y$
 - › T5b: $X\cdot(X'+Y)=X\cdot Y$

OR AND

X	Y	$X+Y$	$X\cdot Y$	$X+(X\cdot Y)$	$X\cdot(X+Y)$	X'	$X'\cdot Y$	$X+(X'\cdot Y)$	$X'\cdot Y$	$X\cdot(X'+Y)$
0	0	0	0	0	0	1	0	0	1	0
0	1	1	0	0	0	1	1	1	1	0
1	0	1	0	1	1	0	0	1	0	0
1	1	1	1	1	1	0	0	1	1	1

BOOLEAN ALGEBRA THEOREMS

- Absorption (aka *combining*)
 - › T6a: $(X \cdot Y) + (X \cdot Y') = X$
 - › T6b: $(X + Y) \cdot (X + Y') = X$

OR AND

X	Y	X+Y	X•Y	Y'	X•Y'	$(X \cdot Y) + (X \cdot Y')$	X+Y'	$(X+Y) \cdot (X+Y')$
0	0	0	0	1	0	0	1	0
0	1	1	0	0	0	0	0	0
1	0	1	0	1	1	1	1	1
1	1	1	1	0	0	1	1	1

BOOLEAN ALGEBRA THEOREMS

- Absorption (aka *combining*)
 - › T7a: $(X \cdot Y) + (X \cdot Y' \cdot Z) = (X \cdot Y) + (X \cdot Z)$
 - › T7b: $(X + Y) \cdot (X + Y' + Z) = (X + Y) \cdot (X + Z)$

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

BOOLEAN ALGEBRA THEOREMS

- DeMorgan's theorem (very important!)

› T8a: $(X+Y)' = X' \cdot Y'$

- $\overline{X+Y} = \overline{X} \cdot \overline{Y}$ break (or connect) the bar & change the sign

› T8b: $\underline{(X \cdot Y)'} = \underline{X' + Y'}$

break (or connect) the bar & change the

- $X \cdot Y = X+Y$

› Generalized DeMorgan's theorem:

- GT8a: $(X_1+X_2+\dots+X_{n-1}+X_n)' = X_1' \cdot X_2' \cdot \dots \cdot X_{n-1}' \cdot X_n'$
- GT8b: $(X_1 \cdot X_2 \cdot \dots \cdot X_{n-1} \cdot X_n)' = X_1' + X_2' + \dots + X_{n-1}' + X_n'$

OR AND

X	Y	X+Y	X•Y	X'	Y'	(X+Y)'	X'•Y'	(X•Y)'	X'+Y'
0	0	0	0	1	1	1	1	1	1
0	1	1	0	1	0	0	0	1	1
1	0	1	0	0	1	0	0	1	1
1	1	1	1	0	0	0	0	0	0

BOOLEAN ALGEBRA THEOREMS

- Consensus Theorem

- T9a: $(X \cdot Y) + (X' \cdot Z) + (Y \cdot Z) = (X \cdot Y) + (X' \cdot Z)$

- T9b: $(X+Y) \cdot (X'+Z) \cdot (Y+Z) = (X+Y) \cdot (X'+Z)$

X	Y	Z	X'	XY	X'Z	YZ	(XY)+ (X'Z)+ (YZ)	(XY)+ (X'Z)	X+Y	X'+Z	Y+Z	(X+Y)• (X'+Z)• (Y+Z)	(X+Y)• (X'+Z)
0	0	0	1	0	0	0	0	0	0	1	0	0	0
0	0	1	1	0	1	0	1	1	0	1	1	0	0
0	1	0	1	0	0	0	0	0	1	1	1	1	1
0	1	1	1	0	1	1	1	1	1	1	1	1	1
1	0	0	0	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	0	0	0	0	1	1	1	1	1
1	1	0	0	1	0	0	1	1	1	0	1	0	0
1	1	1	0	1	0	1	1	1	1	1	1	1	1

MORE THEOREMS?

- Shannon's expansion theorem (also very important!)
 - › T10a: $f(X_1, X_2, \dots, X_{n-1}, X_n) =$
 - $(X_1' \cdot f(0, X_2, \dots, X_{n-1}, X_n)) + (X_1 \cdot f(1, X_2, \dots, X_{n-1}, X_n))$
 - Can be taken further:
 - - $f(X_1, X_2, \dots, X_{n-1}, X_n) = (X_1' \cdot X_2' \cdot f(0, 0, \dots, X_{n-1}, X_n))$
 - + $(X_1 \cdot X_2' \cdot f(1, 0, \dots, X_{n-1}, X_n)) + (X_1' \cdot X_2 \cdot f(0, 1, \dots, X_{n-1}, X_n))$
 - + $(X_1 \cdot X_2 \cdot f(1, 1, \dots, X_{n-1}, X_n))$
 - Can be taken even further:
 - - $f(X_1, X_2, \dots, X_{n-1}, X_n) = (X_1' \cdot X_2' \cdot \dots \cdot X_{n-1}' \cdot X_n' \cdot f(0, 0, \dots, 0, 0))$
 - + $(X_1 \cdot X_2' \cdot \dots \cdot X_{n-1}' \cdot X_n' \cdot f(1, 0, \dots, 0, 0)) + \dots$
 - + $(X_1 \cdot X_2 \cdot \dots \cdot X_{n-1} \cdot X_n \cdot f(1, 1, \dots, 1, 1))$
 - › T10b: $f(X_1, X_2, \dots, X_{n-1}, X_n) =$
 - $(X_1 + f(0, X_2, \dots, X_{n-1}, X_n)) \cdot (X_1' + f(1, X_2, \dots, X_{n-1}, X_n))$
 - Can be taken further as in the case of T10a
 - We'll see significance of Shannon's expansion theorem later

BOOLEAN ALGEBRA THEOREMS

- Idempotency
 - › T1a: $X+X=X$
 - › T1b: $X\bullet X=X$
- Null elements
 - › T2a: $X+1=1$
 - › T2b: $X\bullet 0=0$
- Involution
 - › T3: $(X')'=X$

		OR		AND					
X	Y	X+Y	X•Y	X+X	X•X	X+1	X•0	X'	X''
0	0	0	0	0	0	1	0	1	0
0	1	1	0	0	0	1	0	1	0
1	0	1	0	1	1	1	0	0	1
1	1	1	1	1	1	1	0	0	1

BOOLEAN ALGEBRA THEOREMS

- Absorption (aka *covering*)
 - › T4a: $X+(X\cdot Y)=X$
 - › T4b: $X\cdot(X+Y)=X$
 - › T5a: $X+(X'\cdot Y)=X+Y$
 - › T5b: $X\cdot(X'+Y)=X\cdot Y$

OR AND

X	Y	$X+Y$	$X\cdot Y$	$X+(X\cdot Y)$	$X\cdot(X+Y)$	X'	$X'\cdot Y$	$X+(X'\cdot Y)$	$X'+Y$	$X\cdot(X'+Y)$
0	0	0	0	0	0	1	0	0	1	0
0	1	1	0	0	0	1	1	1	1	0
1	0	1	0	1	1	0	0	1	0	0
1	1	1	1	1	1	0	0	1	1	1

BOOLEAN ALGEBRA THEOREMS

18

- Absorption (aka *combining*)
 - › T6a: $(X \cdot Y) + (X \cdot Y') = X$
 - › T6b: $(X + Y) \cdot (X + Y') = X$

OR AND

X	Y	$X+Y$	$X \cdot Y$	Y'	$X \cdot Y'$	$(X \cdot Y) + (X \cdot Y')$	$X+Y'$	$(X+Y) \cdot (X+Y')$
0	0	0	0	1	0	0	1	0
0	1	1	0	0	0	0	0	0
1	0	1	0	1	1	1	1	1
1	1	1	1	0	0	1	1	1

BOOLEAN ALGEBRA THEOREMS

20

- DeMorgan's theorem (very important!)
 - › T8a: $(X+Y)' = X' \cdot Y'$
 - $\overline{X+Y} = \overline{X} \cdot \overline{Y}$ break (or connect) the bar & change the sign
 - › T8b: $\underline{(X \cdot Y)'} = \underline{X' + Y'}$ break (or connect) the bar & change the sign
 - $X \cdot Y = \overline{\overline{X+Y}}$
 - › Generalized DeMorgan's theorem:
 - GT8a: $(X_1+X_2+\dots+X_{n-1}+X_n)' = X_1' \cdot X_2' \cdot \dots \cdot X_{n-1}' \cdot X_n'$
 - GT8b: $(X_1 \cdot X_2 \cdot \dots \cdot X_{n-1} \cdot X_n)' = X_1' + X_2' + \dots + X_{n-1}' + X_n'$

OR AND

X	Y	X+Y	X•Y	X'	Y'	(X+Y)'	X'•Y'	(X•Y)'	X'+Y'
0	0	0	0	1	1	1	1	1	1
0	1	1	0	1	0	0	0	1	1
1	0	1	0	0	1	0	0	1	1
1	1	1	1	0	0	0	0	0	0

BOOLEAN ALGEBRA THEOREMS

- Consensus Theorem

- T9a: $(X \cdot Y) + (X' \cdot Z) + (Y \cdot Z) = (X \cdot Y) + (X' \cdot Z)$

- T9b: $(X+Y) \cdot (X'+Z) \cdot (Y+Z) = (X+Y) \cdot (X'+Z)$

X	Y	Z	X'	XY	X'Z	YZ	(XY)+(X'Z) (YZ)	(XY)+(X'Z)	X+Y	X'+Z	Y+Z	(X+Y)·(X'+Z) (Y+Z)	(X+Y)·(X'+Z)
0	0	0	1	0	0	0	0	0	0	1	0	0	0
0	0	1	1	0	1	0	1	1	0	1	1	0	0
0	1	0	1	0	0	0	0	0	1	1	1	1	1
0	1	1	1	0	1	1	1	1	1	1	1	1	1
1	0	0	0	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	0	0	0	0	1	1	1	1	1
1	1	0	0	1	0	0	1	1	1	0	1	0	0
1	1	1	0	1	0	1	1	1	1	1	1	1	1

MORE THEOREMS?

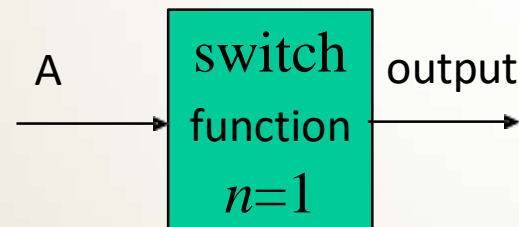
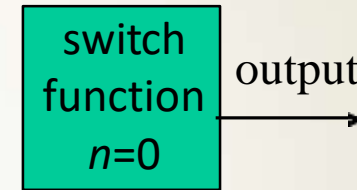
- Shannon's expansion theorem (also very important!)
 - › T10a: $f(X_1, X_2, \dots, X_{n-1}, X_n) =$
 - $(X_1' \cdot f(0, X_2, \dots, X_{n-1}, X_n)) + (X_1 \cdot f(1, X_2, \dots, X_{n-1}, X_n))$
 - Can be taken further:
 - - $f(X_1, X_2, \dots, X_{n-1}, X_n) = (X_1' \cdot X_2' \cdot f(0, 0, \dots, X_{n-1}, X_n))$
 - + $(X_1 \cdot X_2' \cdot f(1, 0, \dots, X_{n-1}, X_n)) + (X_1' \cdot X_2 \cdot f(0, 1, \dots, X_{n-1}, X_n))$
 - + $(X_1 \cdot X_2 \cdot f(1, 1, \dots, X_{n-1}, X_n))$
 - Can be taken even further:
 - - $f(X_1, X_2, \dots, X_{n-1}, X_n) = (X_1' \cdot X_2' \cdot \dots \cdot X_{n-1}' \cdot X_n' \cdot f(0, 0, \dots, 0, 0))$
 - + $(X_1 \cdot X_2' \cdot \dots \cdot X_{n-1}' \cdot X_n' \cdot f(1, 0, \dots, 0, 0)) + \dots$
 - + $(X_1 \cdot X_2 \cdot \dots \cdot X_{n-1} \cdot X_n \cdot f(1, 1, \dots, 1, 1))$
 - › T10b: $f(X_1, X_2, \dots, X_{n-1}, X_n) =$
 - $(X_1 + f(0, X_2, \dots, X_{n-1}, X_n)) \cdot (X_1' + f(1, X_2, \dots, X_{n-1}, X_n))$
 - Can be taken further as in the case of T10a
 - We'll see significance of Shannon's expansion theorem later

SWITCHING FUNCTIONS

- For n variables, there are 2^n possible combinations of values
 - › From all 0s to all 1s
- There are 2 possible values for the output of a function of a given combination of values of n variables
 - › 0 and 1
- There are 2^{2^n} different switching functions for n variables

SWITCHING FUNCTION EXAMPLES

- $n=0$ (no inputs) $\Rightarrow 2^{2^n} = 2^{2^0} = 2^1 = 2$
 - › Output can be either 0 or 1
- $n=1$ (1 input, A) $\Rightarrow 2^{2^n} = 2^{2^1} = 2^2 = 4$
 - › Output can be 0, 1, A, or A'



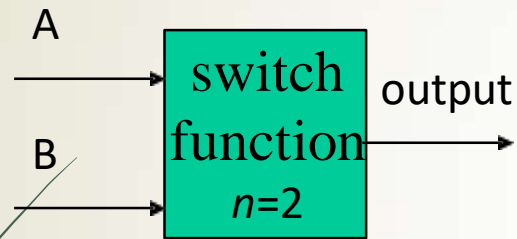
A	f_0	f_1	f_2	f_3
0	0	1	0	1
1	0	0	1	1

$f_0 = 0$
 $f_1 = A'$
 $f_2 = A$
 $f_3 = 1$

SWITCHING FUNCTION EXAMPLES

25

- $n=2$ (2 inputs, A and B) $\rightarrow 2^{2^n} = 2^{2^2} = 2^4 = 16$



AB	f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}	0	0
0 0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0 1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	1
1 0	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	1
1 1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	1

$$f_0 = 0$$

$$f_1 = A'B' = (A+B)'$$

$$f_2 = A'B$$

$$f_3 = A'B' + A'B = A'(B' + B) = A'$$

logic 0

NOT-OR or NOR

invert A

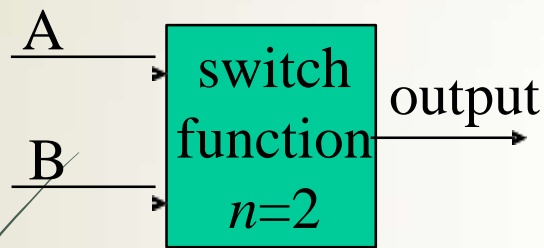
Most frequently used

Less frequently used

Least frequently used

SWITCHING FUNCTION EXAMPLES

- $n=2$ (2 inputs, A and B) $\rightarrow 2^{2^n} = 2^{2^2} = 2^4 = 16$



AB	f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}	0	0
00	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	0
01	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	1
10	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	1
11	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	1

$$f_4 = AB'$$

$$f_5 = A'B' + AB' = (A' + A)B' = B'$$

$$f_6 = A'B + AB'$$

$$f_7 = A'B' + A'B + AB' = A'(B' + B) + (A' + A)B' = A' + B' = (AB)'$$

Most frequently used

Less frequently used

invert B

exclusive-OR

NOT-AND or

NAND

Least frequently used

SWITCHING FUNCTION EXAMPLES

27

- $n=2$ (2 inputs, A and B) $\rightarrow 2^{2^n} = 2^{2^2} = 2^4 = 16$

A	B	switch function $n=2$	output	AB	f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}			
0	0			00	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1			
0	1			01	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	
1	0			10	0	0	0	0	1	1	1	1	0	0	0	0	0	1	1	1	1	0	0
1	1			11	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1

$$f_8 = AB$$

$$f_9 = A'B' + AB$$

$$f_{10} = A'B + AB = (A' + A)B = B$$

$$f_{11} = A'B' + A'B + AB = A'(B' + B) + (A' + A)B = A' + B$$

AND

exclusive-

NOR buffer B

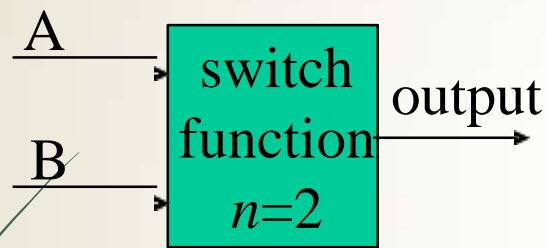
Most frequently used

Less frequently used

Least frequently used

SWITCHING FUNCTION EXAMPLES

- $n=2$ (2 inputs, A and B) $\rightarrow 2^{2^n} = 2^{2^2} = 2^4 = 16$



AB	f ₀	f ₁	f ₂	f ₃	f ₄	f ₅	f ₆	f ₇	f ₈	f ₉	f ₁₀	f ₁₁	f ₁₂	f ₁₃	f ₁₄	f ₁₅	0	0
00	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	0
01	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
10	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0
11	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0

$f_{12} = AB' + AB = A(B' + B) = A$ bufferA

$f_{13} = A'B' + AB' + AB = A(B' + B) + A'B' = A + A'B' = A + B'$

$f_{14} = A'B + AB' + AB = A(B' + B) + (A' + A)B = A + B$ OR

$f_{15} = A'B' + A'B + AB' + AB = A'(B' + B) + A(B' + B) = A' + A = 1$ logic 1

Most frequently used

Less frequently used

Least frequently used

CANONICAL AND STANDERED FORMS

29

- Logical functions are generally expressed in terms of different combinations of logical variables with their true forms as well as the complement forms. Binary logic values obtained by the logical functions and logic variables are in binary form. An arbitrary logic function can be expressed in the following forms.
 - Sum of the Products (SOP)
 - Product of the Sums (POS)

CANONICAL AND STANDERED FORMS

30

- **Product Term:** In Boolean algebra, the logical product of several variables on which a function depends is considered to be a product term. In other words, the AND function is referred to as a product term or standard product.
- **Sum Term:** An OR function is referred to as a sum term
- **Sum of Products (SOP):** The logical sum of two or more logical product terms is referred to as a sum of products expression

$$Y = AB + BC + AC$$

- **Product of Sums (POS):** Similarly, the logical product of two or more logical sum terms is called a product of sums expression

$$Y = (A + B + C)(\bar{A} + \bar{B} + \bar{C})$$

- **Standard form:** The standard form of the Boolean function is when it is expressed in sum of the products or product of the sums fashion

$$Y = AB + BC + AC$$

CANONICAL AND STANDERED FORMS

31

- **Nonstandard Form:** Boolean functions are also sometimes expressed in nonstandard forms like $F = (AB + CD)(\bar{A}\bar{B} + \bar{C}\bar{D})$, which is neither a sum of products form nor a product of sums form.
- **Minterm:** A product term containing all n variables of the function in either true or complemented form is called the minterm. Each minterm is obtained by an AND operation of the variables in their true form or complemented form.
- **Maxterm:** A sum term containing all n variables of the function in either true or complemented form is called the Maxterm. Each Maxterm is obtained by an OR operation of the variables in their true form or complemented form.

CANONICAL SUM OF PRODUCTS

32

- When a Boolean function is expressed as the logical sum of all the minterms from the rows of a truth table, for which the value of the function is 1, it is referred to as the canonical sum of product expression
- For example, if the canonical sum of product form of a three-variable logic function F has the minterms $\bar{A}BC$ and $A\bar{B}C$ this can be expressed as the sum of the decimal codes corresponding to these minterms as below..

$$\begin{aligned}F(A, B, C) &= \Sigma(3,5,6) \\ &= m_3 + m_5 + m_6 \\ &= \bar{A}BC + A\bar{B}C + ABC\end{aligned}$$

CANONICAL SUM OF PRODUCTS

33

- The canonical sum of products form of a logic function can be obtained by using the following procedure:
 - Check each term in the given logic function.
Retain if it is a minterm, continue to examine the next term in the same manner.
 - Examine for the variables that are missing in each product which is not a minterm. If the missing variable in the minterm is X , multiply that minterm with $(X+X')$.
 - Multiply all the products and discard the redundant terms.

CANONICAL SUM OF PRODUCTS

34

- **Example:** Obtain the canonical sum of product form of the following function $F(A, B, C) = A + BC$

- Solution:

$$\begin{aligned}F(A, B, C) &= A + BC \\&= A(B + \bar{B})(C + \bar{C}) + BC(A + \bar{A}) \\&= (AB + A\bar{B})(C + \bar{C}) + ABC + \bar{A}BC \\&= ABC + A\bar{B}C + AB\bar{C} + A\bar{B}\bar{C} + ABC + \bar{A}BC\end{aligned}$$

$$= ABC + A\bar{B}C + AB\bar{C} + A\bar{B}\bar{C} + \bar{A}BC \text{ (as } ABC + ABC = ABC\text{)}$$

- Hence the canonical sum of the product expression of the given function is

$$F(A, B, C) = ABC + A\bar{B}C + AB\bar{C} + A\bar{B}\bar{C} + \bar{A}BC$$

CANONICAL SUM OF PRODUCTS

35

The product of sums form is a method (or form) of simplifying the Boolean expressions of logic gates. In this POS form, all the variables are ORed, i.e. written as sums to form sum terms. All these sum terms are ANDed (multiplied) together to get the product-of-sum form. This form is exactly opposite to the SOP form. So this can also be said as —Dual of SOP form.

$$(A+B) * (A + B + C) * (C +D) (A+B)^{\neg} * (C + D + E^{\neg})$$

CANONICAL SUM OF PRODUCTS

36

POS form can be obtained by

- Writing an OR term for each input combination, which produces LOW output.
- Writing the input variables if the value is 0, and write the complement of the variable if its value is 1 AND the OR terms to obtain the output function.

CANONICAL SUM OF PRODUCTS

37

Example:

Boolean expression for majority function $F = (A + B + C) (A + B + C') (A + B' + C) (A' + B + C)$

A	B	C	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

Now write the input variables combination with high output. $F = AB + BC + AC$.

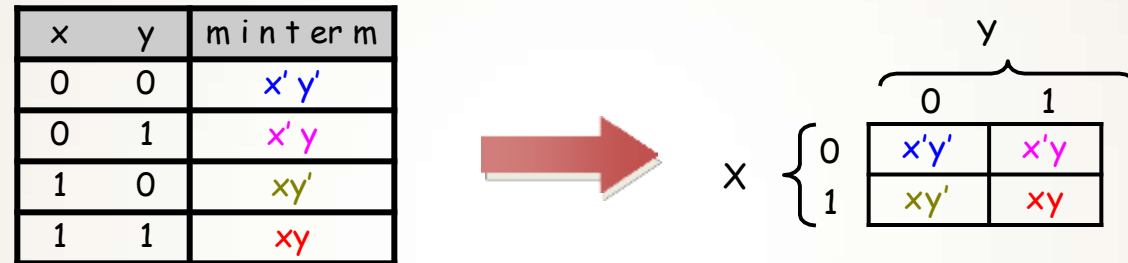
KARANAUGH MAP

38

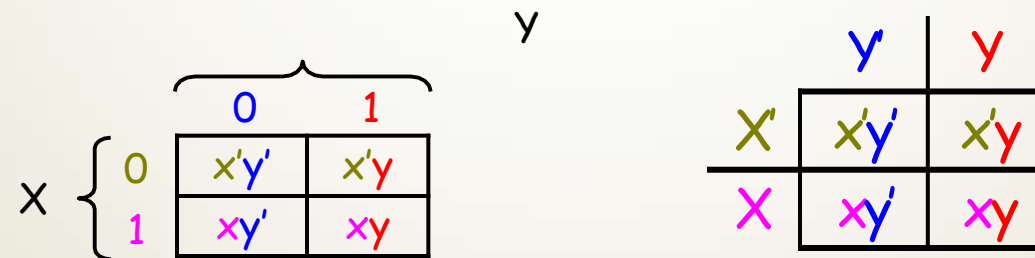
- Boolean algebra helps us simplify expressions and circuits
- Karnaugh Map: A graphical technique for simplifying a Boolean expression into either form:
 - minimal sum of products(MSP)
 - minimal product of sums (MPS)
- Goal of the simplification.
 - There are a minimal number of product/sumterms
 - Each term has a minimal number of literals

KARANAUGH MAP

- A two-variable function has four possible minterms. We can re-arrange these minterms into a Karnaugh map



- Now we can easily see which minterms contain common literals
 - Minterms on the left and right sides contain y' and y respectively
 - Minterms in the top and bottom rows contain x' and x respectively



KARANAUGH MAP

- Make as few rectangles as possible, to minimize the number of products in the final expression.
- Make each rectangle as large as possible, to minimize the number of literals in each term.
- Rectangles can be overlapped, if that makes them larger
- The most difficult step is grouping together all the 1s in the K-map
 - Make rectangles around groups of one, two, four or eight 1s
 - All of the 1s in the map should be included in at least one rectangle. Do not include any of the 0s
 - Each group corresponds to one product term

	0	1	0	0
x	0	1	1	1
		Z		

KARANAUGH MAP

- Maxterms are grouped to find minimal PoS expression

	00	01	11	10
	yz			
0	$x + y + z$	$x + y + z'$	$x + y' + z'$	$x + y' + z$
1	$x' + y + z$	$x' + y + z'$	$x' + y' + z'$	$x' + y' + z$

3 Variable k-Map

- Let's consider simplifying $f(x,y,z) = xy + y'z + xz$
- You should convert the expression into a sum of minterms form,
 - The easiest way to do this is to make a truth table for the function, and then read off the minterms
 - You can either write out the literals or use the minterm shorthand
 - Here is the truth table and sum of minterms for our example:

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

$$\begin{aligned} f(x,y,z) &= x'y'z + xy'z + xyz' \\ &\quad + xyz \\ &= m_1 + m_5 + m_6 + m_7 \end{aligned}$$

3 VARIABLE K-MAP

- For a three-variable expression with inputs x, y, z , the arrangement of minterms is more tricky:

		YZ			
		00	01	11	10
X	0	$x'y'z'$	$x'y'z$	$x'yz$	$x'yz'$
	1	$xy'z'$	$xy'z$	xyz	xyz'

		y			
X		$x'y'z'$	$x'y'z$	$x'yz$	$x'yz'$
		$xy'z'$	$xy'z$	xyz	xyz'

		YZ			
		00	01	11	10
X	0	m_0	m_1	m_3	m_2
	1	m_4	m_5	m_7	m_6

		y			
X		m_0	m_1	m_3	m_2
		m_4	m_5	m_7	m_6

3-VARIABLE K-MAP

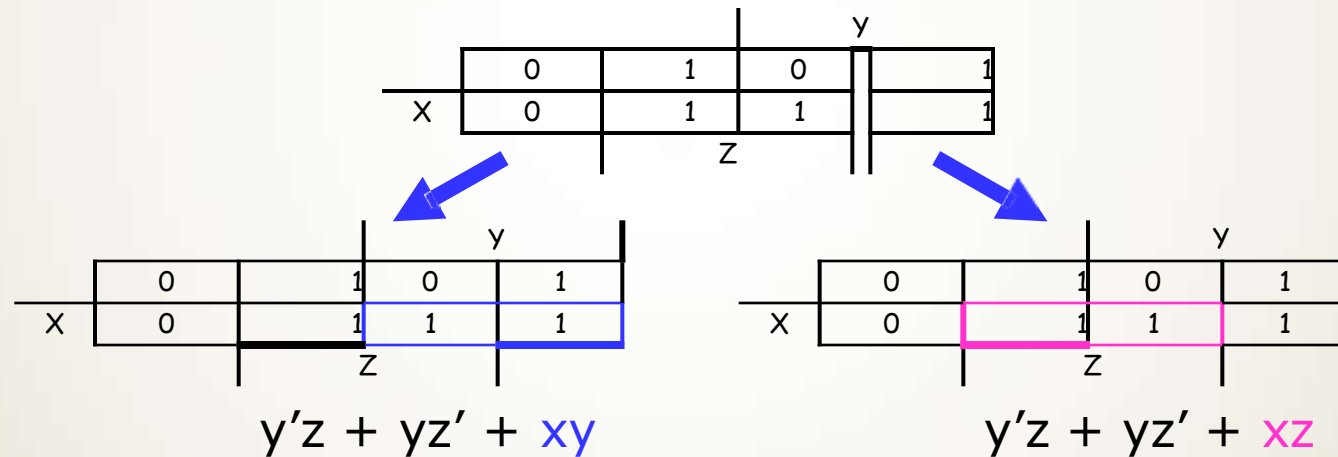
- Here is the filled in K-map, with all groups shown
 - The magenta and green groups overlap, which makes each of them as large as possible
 - Minterm m_6 is in a group all by its lonesome

				y
	0	1	1	0
x	0	1	0	1
		z		

- The final MSP here is $x'z + y'z + xyz'$

3-VARIABLE K-MAP

- There may not necessarily be a *unique* MSP. The K-map below yields two valid and equivalent MSPs, because there are two possible ways to include minterm m_7



- Remember that overlapping groups is possible, as shown above

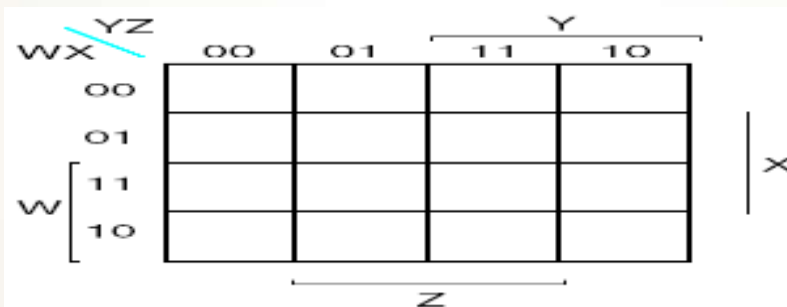
3-VARIABLE K-MAP

- Maxterms are grouped to find minimal PoS expression

	00	01	11	10
	yz			
x	0	1	1	0
0	$x+y+z$	$x+y+z'$	$x+y'+z'$	$x+y'+z$
1	$x'+y+z$	$x'+y+z'$	$x'+y'+z'$	$x'+y'+z$

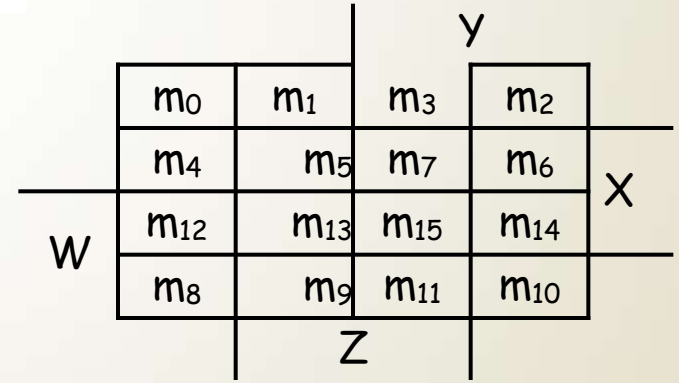
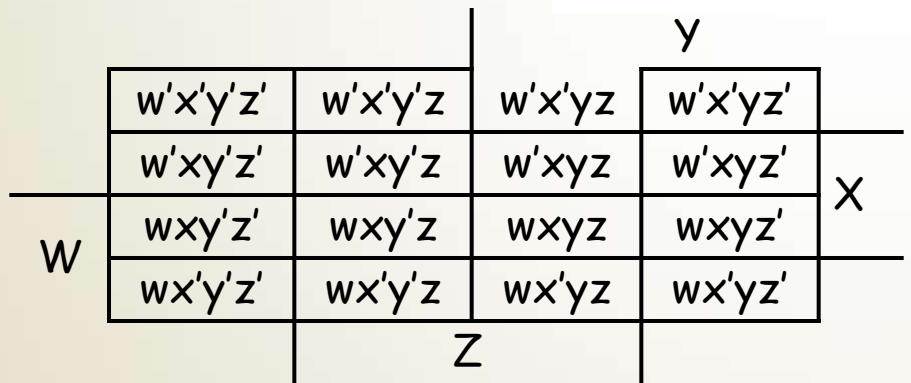
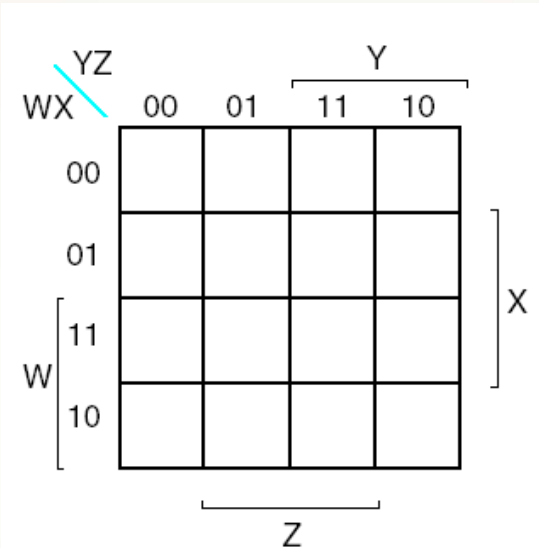
4-VARIABLE K-MAP

- We can do four-variable expressions too!
 - The minterms in the third and fourth columns, *and* in the third and fourth rows, are switched around.
 - Again, this ensures that adjacent squares have common literals



- Grouping minterms is similar to the three-variable case, but:
 - You can have rectangular groups of 1, 2, 4, 8 or 16 minterms
 - You can wrap around all four sides

4-VARIABLE K-MAP



4-VARIABLE K-MAP

- The expression is a sum of minterms, so here's the K map:

ion
y is a
read

	1	0	0	1	
	0	1	0	0	X
W	0	1	0	0	
	1	0	0	1	
	Z				

	m ₀	m ₁	m ₃	m ₂	
	m ₄	m ₅	m ₇	m ₆	X
W	m ₁₂	m ₁₃	m ₁₅	m ₁₄	
	m ₈	m ₉	m ₁₁	m ₁₀	
	Z				

- We can make the following groups, resulting in the MSP $x'z' + xy'z$

		y		
	1	0	0	
	0	1	0	X
W	0	1	0	
	1	0	0	1
	Z			

		y		
	w'x'y'z	w'x'y'z	w'x'yz	w'x'yz'
	w'xy'z'	w'xy'z	w'xyz	w'xyz'
W	wxy'z'	wxy'z	wxyz	wxyz'
	wx'y'z	wx'y'z	wx'yz	wx'yz'
	Z		I	

Example: Simplify $m_0 + m_2 + m_5 + m_8 + m_{10} + m_{13}$

4-VARIABLE K-MAP

- $F(W,X,Y,Z) = \prod M(0,1,2,4,5)$

		y z	
		0 0	0 1
x	0	$x + y + z$	$x + y + z'$
		1 1	
	1	$x' + 0 + y + z$	$x' + y + 0 + z' + 1 + z$
		1 0	
		$x + y' + z'$	$x + y' + z$

$$F(W,X,Y,Z) = Y \cdot (X + Z)$$

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

5-VARIABLE K-MAP

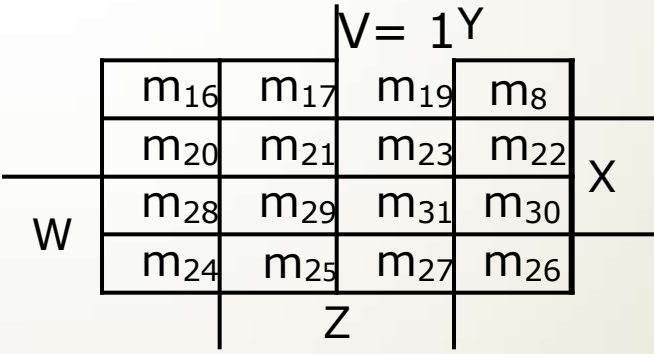
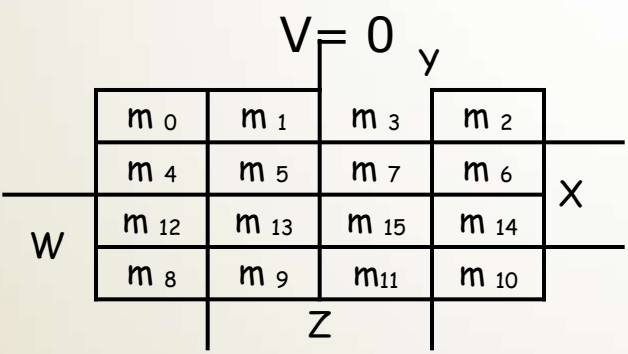
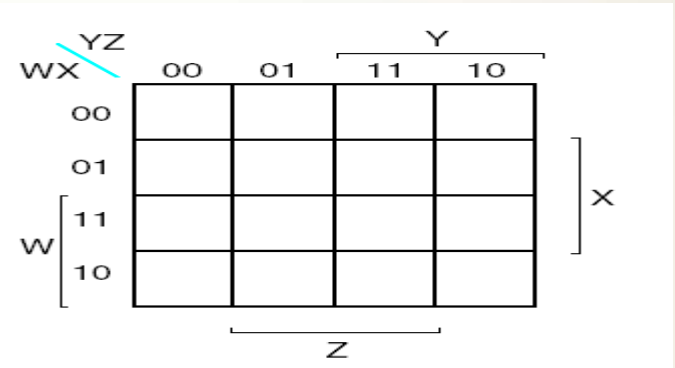
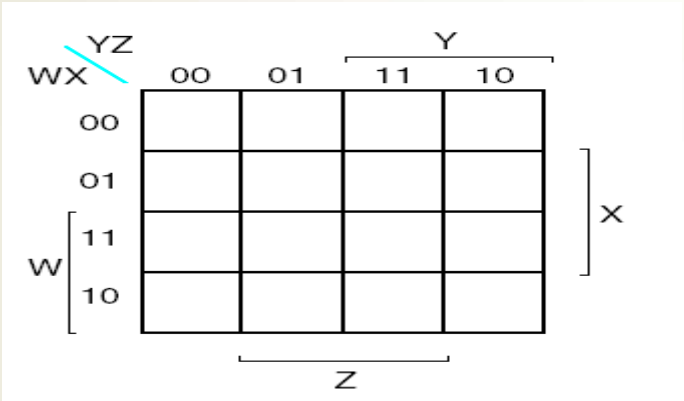
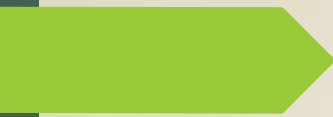
- **Objective:**

Understand the 5-Variable K-map

- **Course Outcomes(CAEC020.06):**

Evaluate the functions using various types of minimizing algorithms like Karanaugh map method.

5-variable K-map



5-VARIABLE K-MAP

- In our example, we can write $f(x,y,z)$ in two equivalent ways

$$f(x,y,z) = x'y'z + xy'z + xyz' + xyz$$

		y	
	$x'y'z'$	$x'y'z$	$x'yz$
X	$xy'z'$	$xy'z$	xyz'
		z	

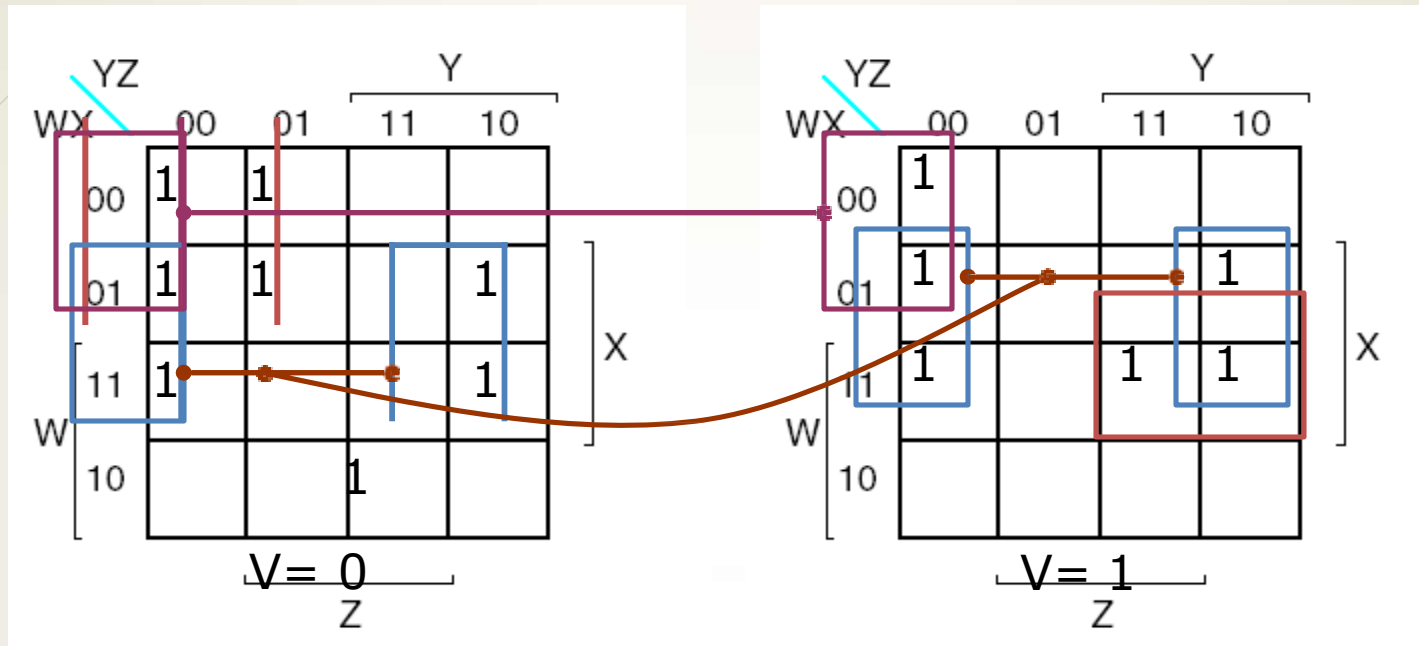
$$f(x,y,z) = m_1 + m_5 + m_6 + m_7$$

		y	
	m_0	m_1	m_3
X	m_4	m_5	m_6
		z	

- In either case, the resulting K-map is shown below

		y	
	0	1	0
X	0	1	1
		z	

5-VARIABLE K-MAP



$$f = XZ'$$

$$\Sigma m(4,6,12,14,20,22,28,30)$$

$$+ V'W'Y'$$

$$\Sigma m(0,1,4,5)$$

$$+ W'Y'Z'$$

$$\Sigma m(0,4,16,20)$$

$$+ VWXY$$

$$\Sigma m(30,31)$$

+

m11

$$V'WX'YZ$$

DON'T CARE CONDITION

- You don't always need all 2^n input combinations in an n -variable function
 - If you can guarantee that certain input combinations never occur
 - If some outputs aren't used in the rest of the circuit

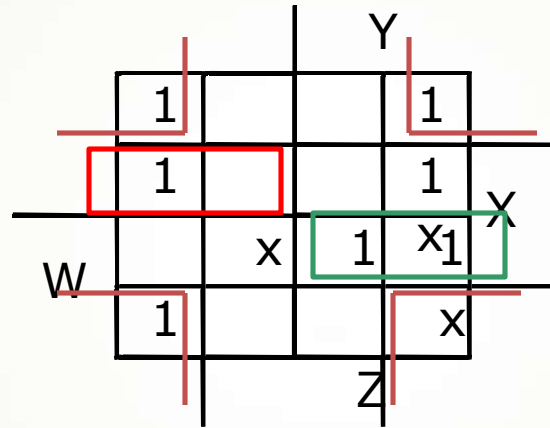
x	y	z	$f(x,y,z)$
0	0	0	0
0	0	1	1
0	1	0	X
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	X
1	1	1	1

- We mark don't-care outputs in truth tables and K-maps with Xs.

DON'T CARE CONDITIONS

- Find a MSP for:

$$f(w,x,y,z) = \sum m(0,2,4,5,8,14,15), d(w,x,y,z) = \sum m(7,10,13)$$



$$f(w,x,y,z) = x'z' + w'xy' + wxy$$

NAND NOR IMPLEMENTATION

58

- The objectives of this lesson are to learn about:
 1. Universal gates - NAND and NOR.
 2. How to implement NOT, AND, and OR gate using NAND gates only.
 3. How to implement NOT, AND, and OR gate using NOR gates only.
 4. Equivalent gates.

NAND NOR IMPLEMENTATION

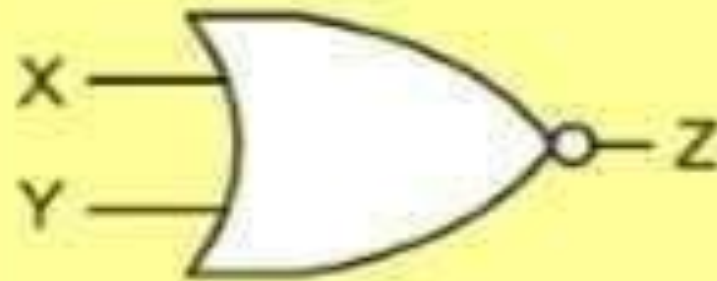
59

X	Y	NAND
0	0	1
0	1	1
1	0	1
1	1	0



$$Z = \overline{X \cdot Y}$$

X	Y	NOR
0	0	1
0	1	0
1	0	0
1	1	0



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

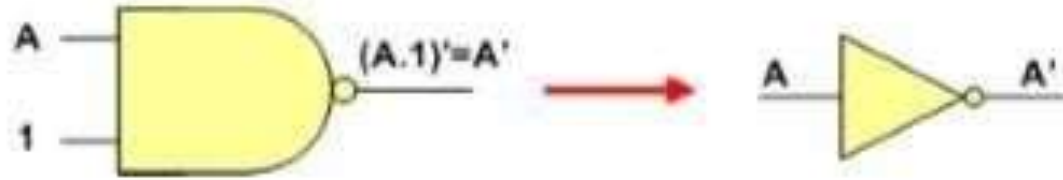
NAND NOR IMPLEMENTATION

61

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

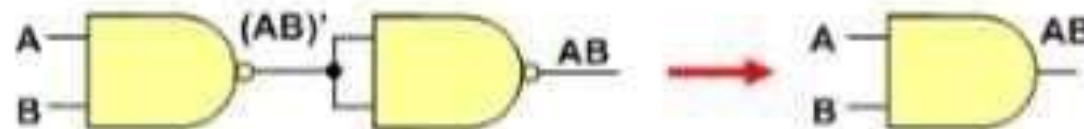


2. One NAND input pin is connected to the input signal **A** while all other input pins are connected to logic **1**. The output will be **A'**.



Implementing AND Using only NAND Gates

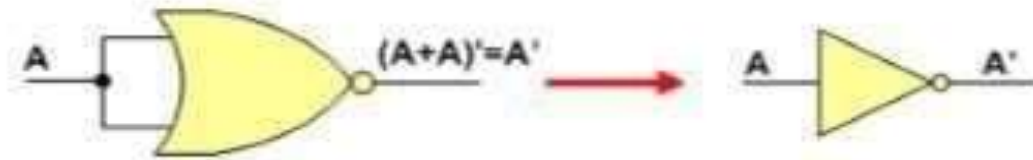
An AND gate can be replaced by NAND gates as shown in the figure (The AND is replaced by a NAND gate with its output complemented by a NAND gate inverter).



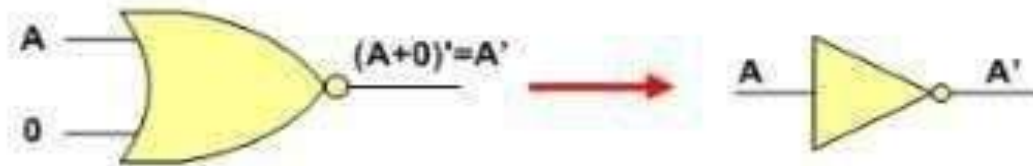
NAND NOR IMPLEMENTATION

62

1. All NOR input pins connect to the input signal **A** gives an output **A'**.

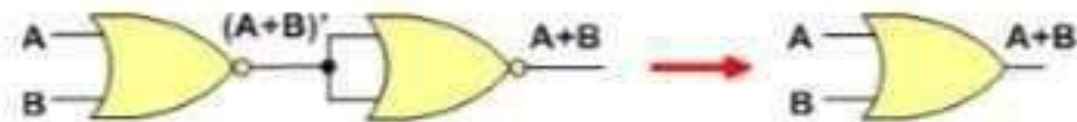


2. One NOR input pin is connected to the input signal **A** while all other input pins are connected to logic **0**. The output will be **A'**.



Implementing OR Using only NOR Gates

An **OR gate** can be replaced by NOR gates as shown in the figure (The OR is replaced by a NOR gate with its output complemented by a NOR gate inverter)



TWO LEVEL Implementation

63

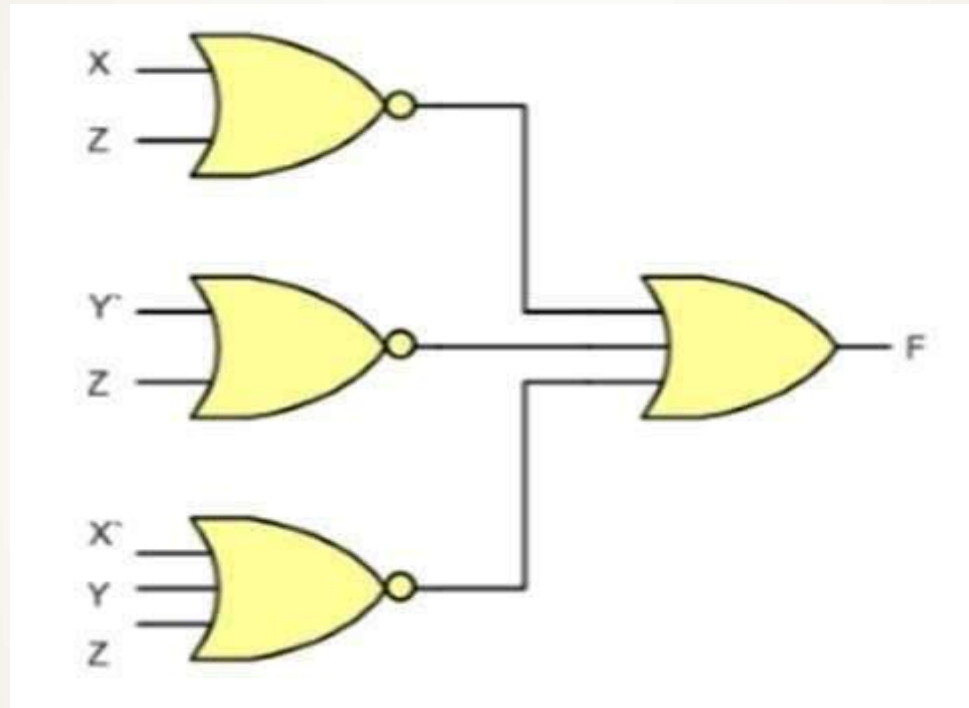
OR NAND Function:

$$F = \overline{(X+Z)(\bar{Y}+Z)(\bar{X}+Y+Z)} \text{ or}$$
$$\bar{F} = (X+Z)(\bar{Y}+Z)(\bar{X}+Y+Z)$$

Since 'F' is in POS form Z can be implemented by using NOR NOR circuit. Similarly complementing the output we can get F, or by using NOR –OR Circuit as shown in figure

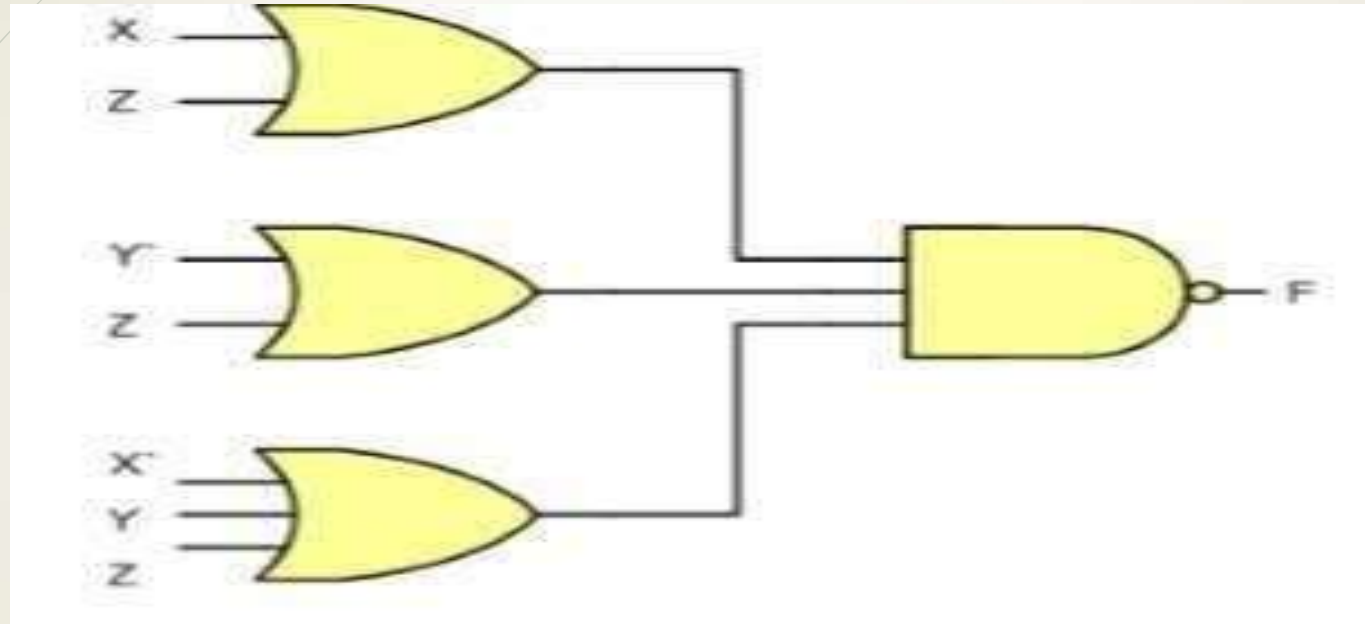
TWO LEVEL Implementation

64



TWO LEVEL Implementation

65

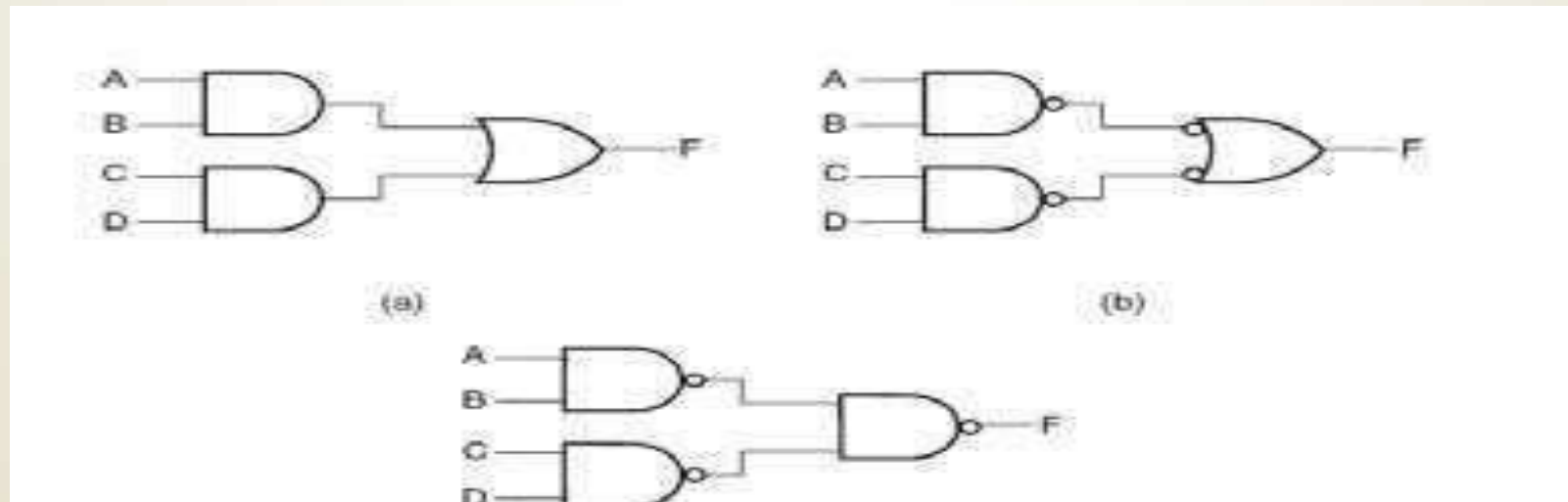


It can also be implemented using OR-NAND circuit as it is equivalent to NOR-OR circuit

Two Level Implementation

66

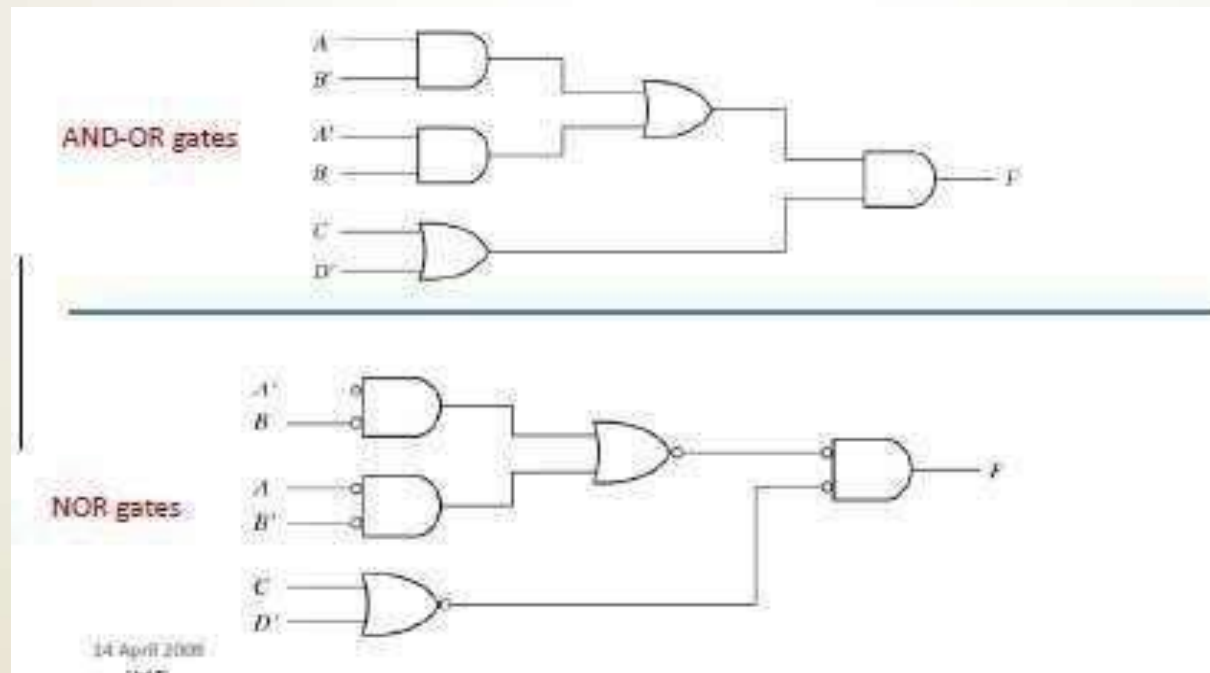
- **Example1: implement the following function $F = AB + CD$**
- The implementation of Boolean functions with NAND gates requires that the functions be in
- sum of products (SOP) form.
- The Rule
- This function can be implemented by three different ways as shown in the circuit diagram a, b, c



Two Level Implementation

67

Example 2: Consider the following Boolean function, implement the circuit diagram by using multilevel NOR gate. $F = (AB' + A'B)(C + D')$

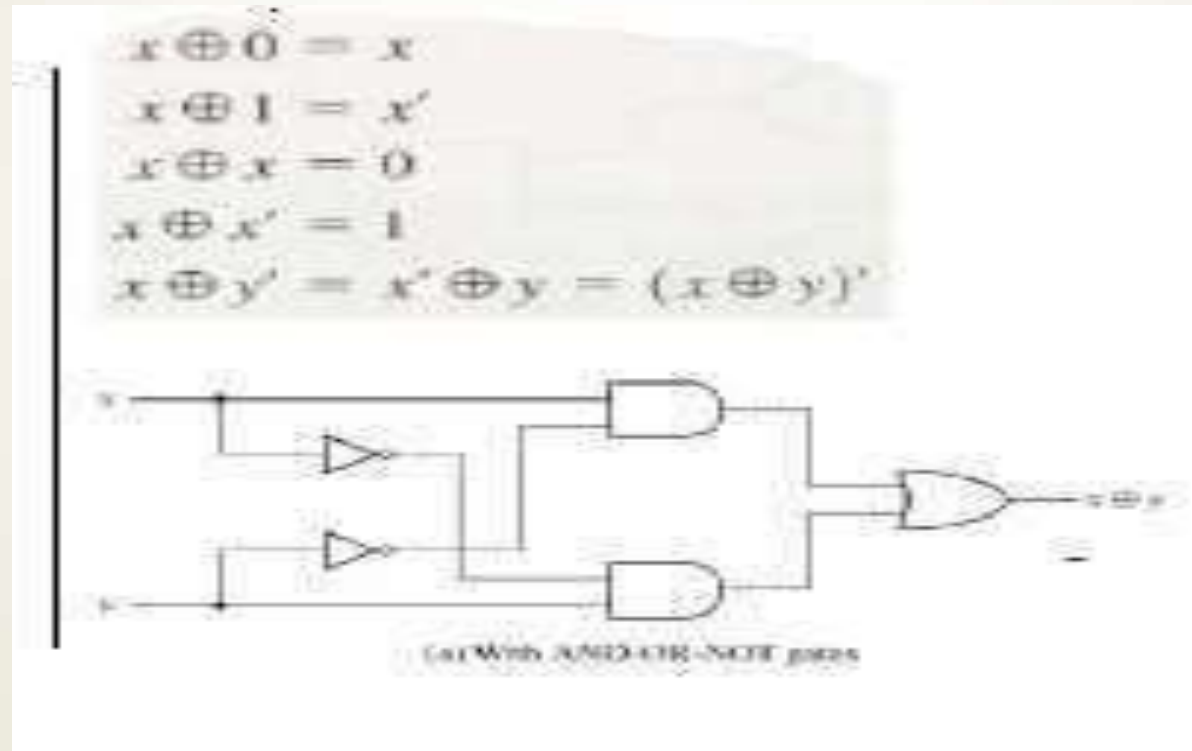


Two Level Implementation

68

Exclusive-OR (XOR) Function:

$$\text{XOR: } x \oplus y = xy' + x'y$$

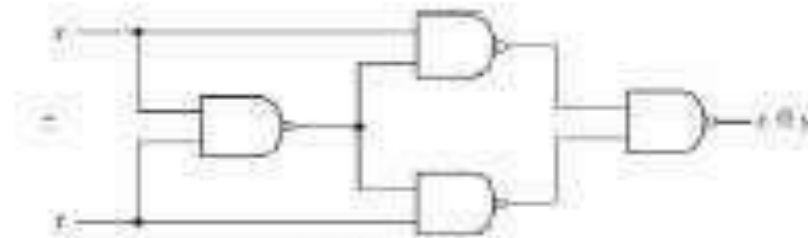


Two Level Implementation

69

Exclusive-NOR = equivalence

$$\begin{aligned}(x \oplus y)' &= (xy' + x'y)' \\ &= (x' + y)(x + y') = x'y' + xy\end{aligned}$$



(a) With NAND gates

TWO LEVEL Implementation

70

X	Y	NAND
0	0	1
0	1	1
1	0	1
1	1	0



$$Z = \overline{X \cdot Y}$$

TWO LEVEL Implementation

71

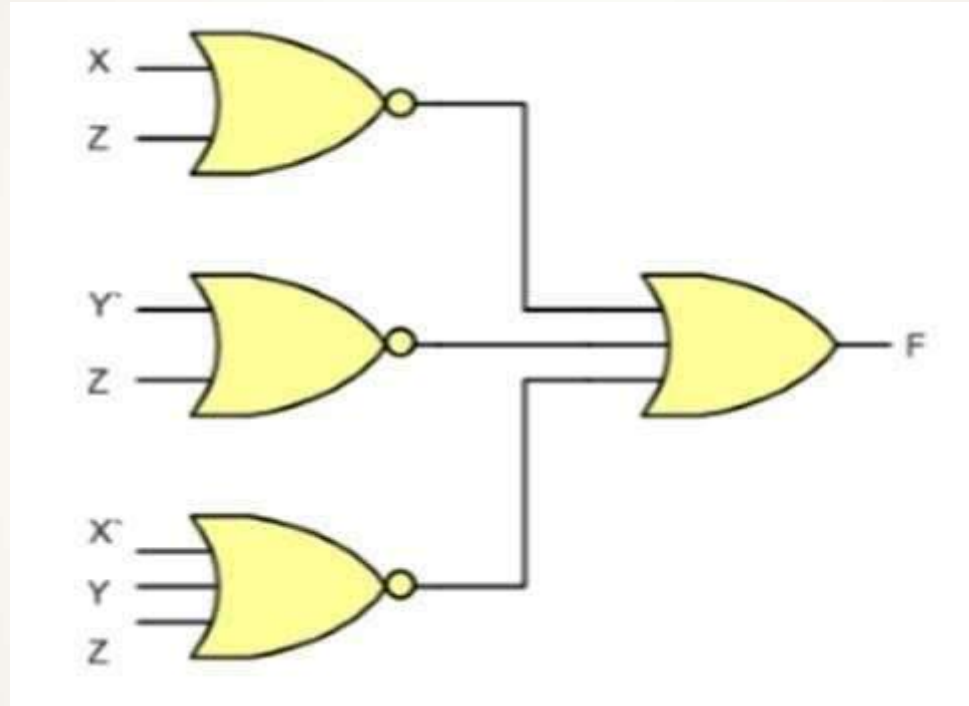
OR NAND Function:

$$F = \overline{(X+Z)(\bar{Y}+Z)(\bar{X}+Y+Z)} \text{ or}$$
$$\bar{F} = (X+Z)(\bar{Y}+Z)(\bar{X}+Y+Z)$$

Since 'F' is in POS form Z can be implemented by using NOR NOR circuit. Similarly complementing the output we can get F, or by using NOR –OR Circuit as shown in figure

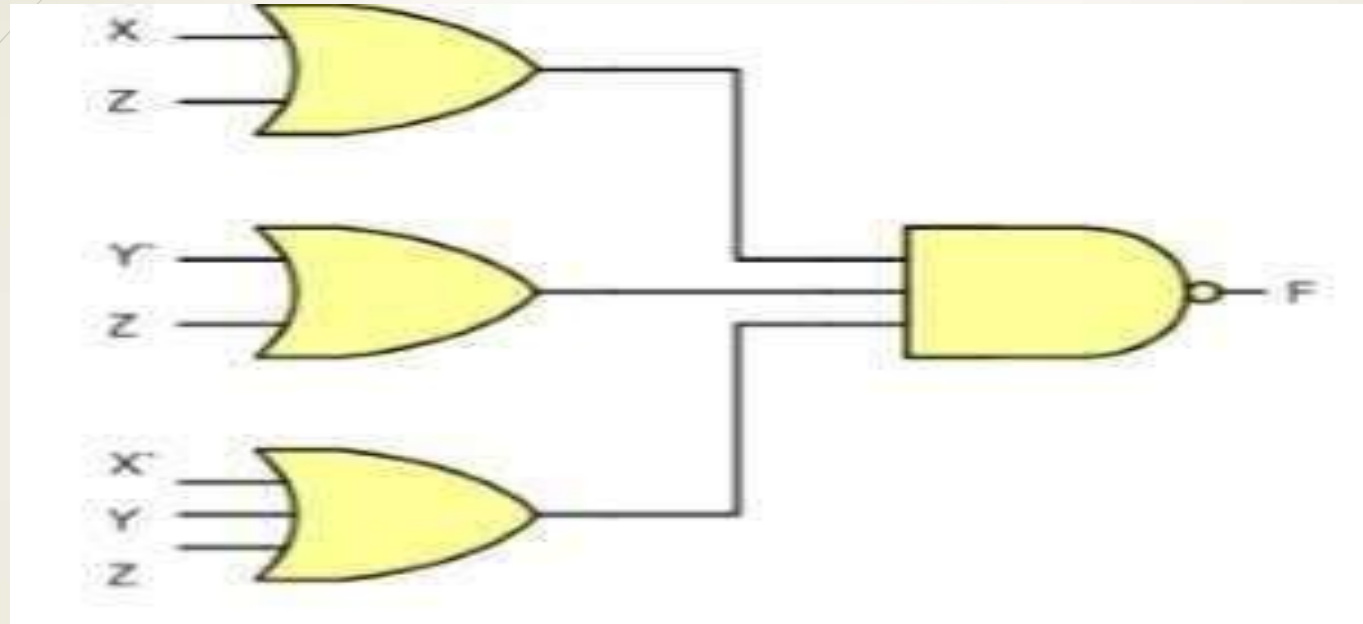
TWO LEVEL Implementation

72



TWO LEVEL Implementation

73

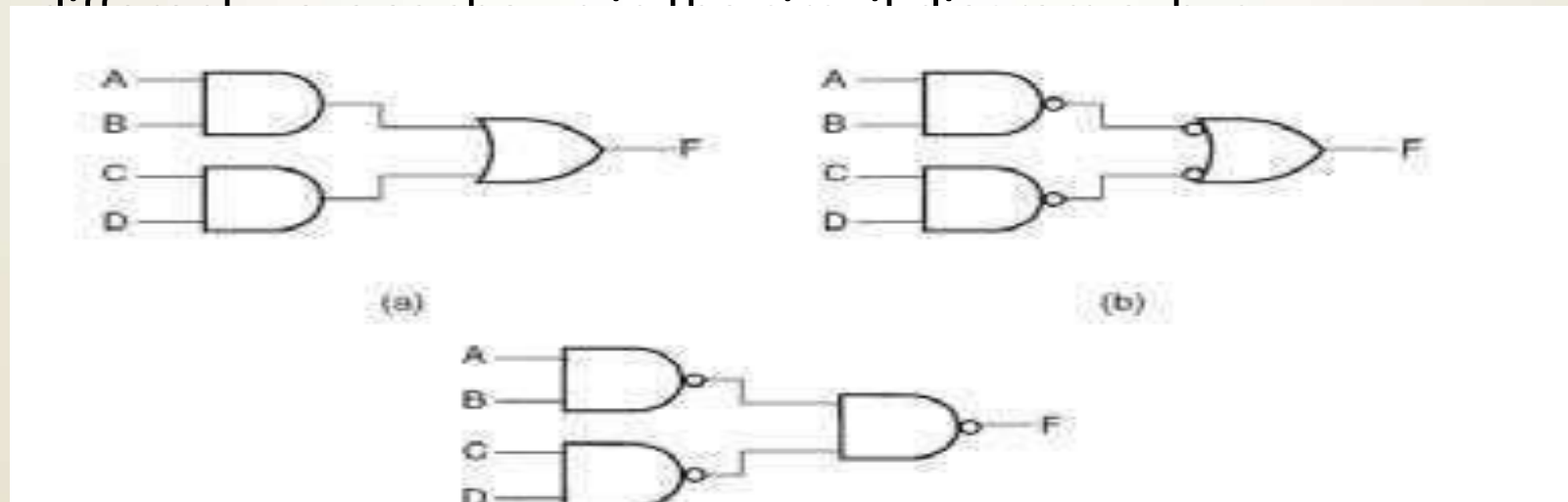


It can also be implemented using OR-NAND circuit as it is equivalent to NOR-OR circuit

Two Level Implementation

74

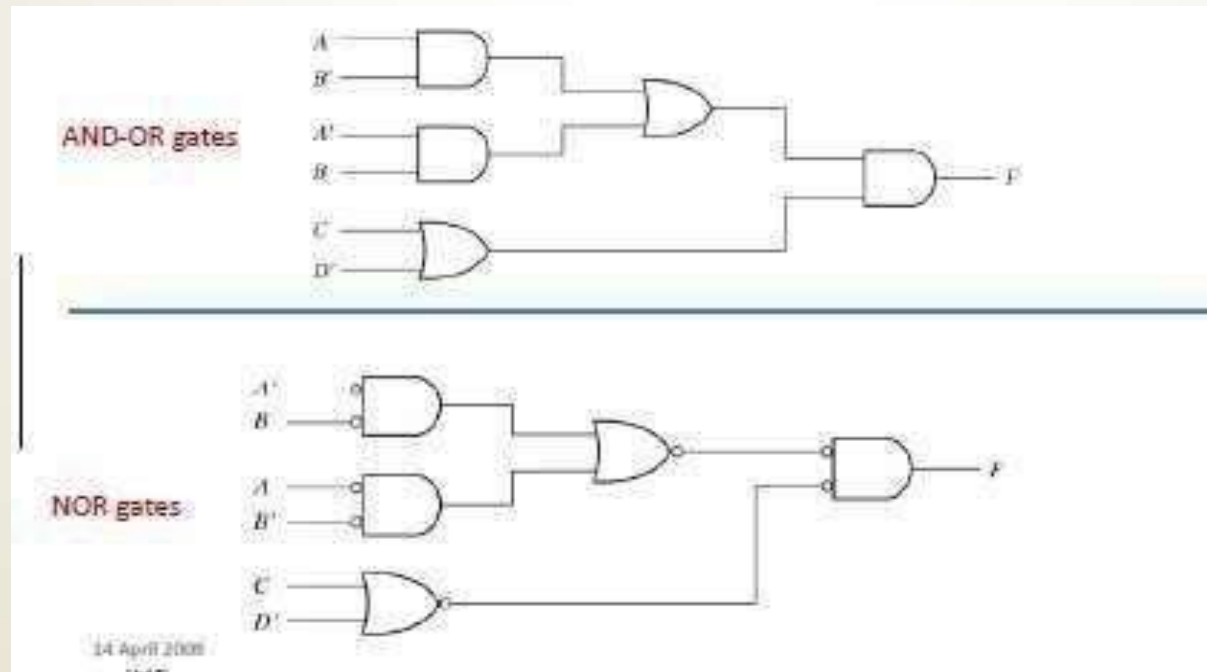
- **Example1: implement the following function $F = AB + CD$**
- The implementation of Boolean functions with NAND gates requires that the functions be in
 - sum of products (SOP) form.
 - The Rule
- This function can be implemented by three



Two Level Implementation

75

Example 2: Consider the following Boolean function, implement the circuit diagram by using multilevel NOR gate. $F = (AB' + A'B)(C + D')$

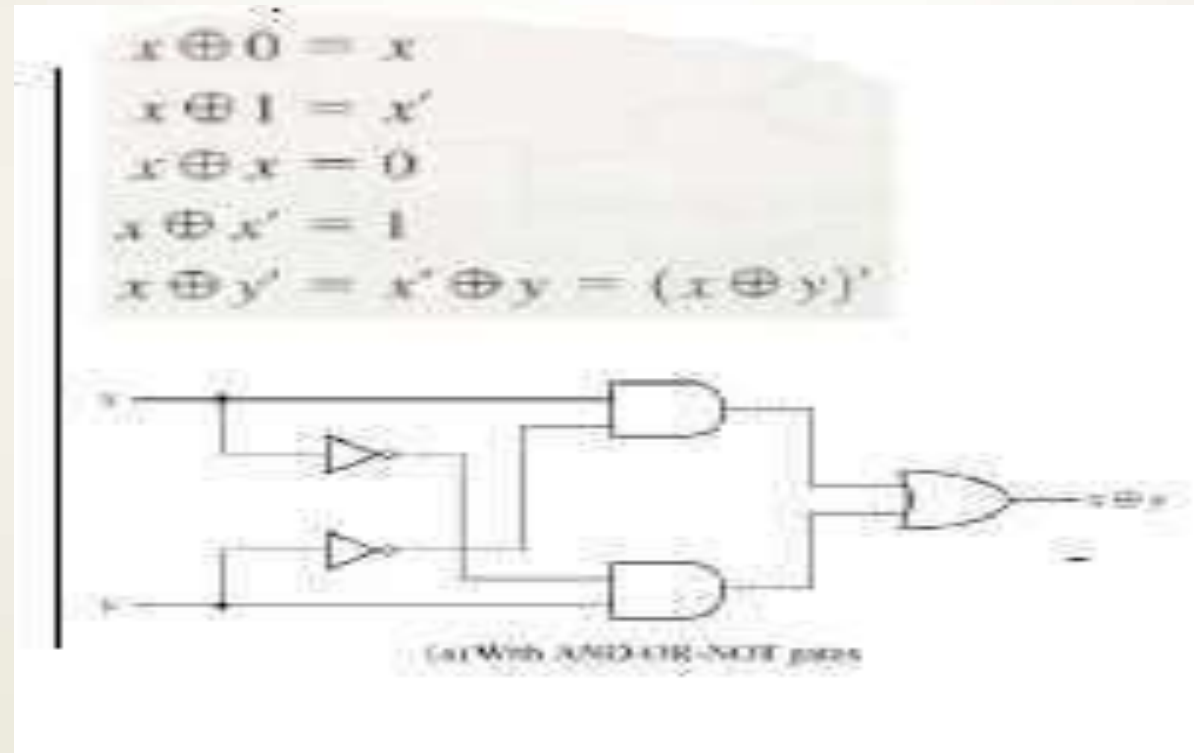


Two Level Implementation

76

Exclusive-OR (XOR) Function:

XOR: $xy' + x'y$

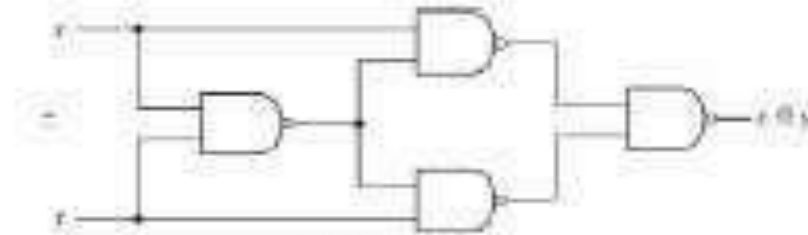


Two Level Implementation

77

Exclusive-NOR = equivalence

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



(a) With NAND gates

COMBINATIONAL CIRCUITS

Combinational Circuits

- Combinational circuit is a circuit in which we combine the different gates in the circuit, for example encoder, decoder, multiplexer and demultiplexer.

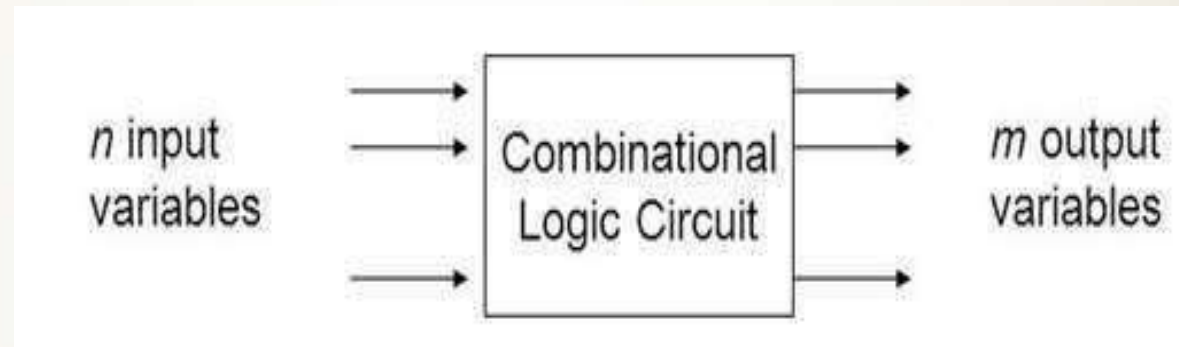
Some of the characteristics of combinational circuits are following:

- The output of combinational circuit at any instant of time, depends only on the levels present at input terminals.
- The combinational circuit do not use any memory. The previous state of input does not have any effect on the present state of the circuit.
- A combinational circuit can have an n number of inputs and m number of outputs.

Combinational Circuits

- **Block diagram:**

2^n possible combinations of input values.



- Specific functions :of combinationalcircuits

Adders,subtractors,multiplexers,comprators,encoder,Decoder.

MSI Circuits and standard cells

ANALYSIS PROCEDURE

81

Analysis procedure

To obtain the output Boolean functions from a logic diagram, proceed as follows:

1. Label all gate outputs that are a function of input variables with arbitrary symbols. Determine the Boolean functions for each gate output.
2. Label the gates that are a function of input variables and previously labeled gates with other arbitrary symbols. Find the Boolean functions for these gates.
3. Repeat the process outlined in step 2 until the outputs of the circuit are obtained.

DESIGN PROCEDURE

Design Procedure

1. The problem is stated
2. The number of available input variables and required output variables is determined.
3. The input and output variables are assigned letter symbols.
4. The truth table that defines the required relationship between inputs and outputs is derived.
5. The simplified Boolean function for each output is obtained.
6. The logic diagram is drawn.

BINARY ADDERS

ADDERS

Half Adder

A Half Adder is a combinational circuit with two binary inputs (augends and addend bits) and two binary outputs (sum and carry bits.) It adds the two inputs (A and B) and produces the sum (S) and the carry (C) bits.



Fig 1:Block diagram

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

Fig 2:Truth table

$$\text{Sum} = A'B + AB' = A \oplus B$$

$$\text{Carry} = AB$$

BINARY ADDERS

84

Full Adder

The full-adder adds the bits A and B and the carry from the previous column called the carry-in C_{in} and outputs the sum bit S and the carry bit called the carry-out C_{out} .

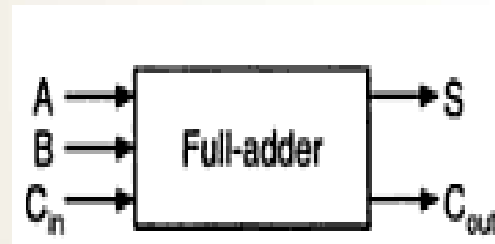


Fig 3: block diagram

$$S = \bar{A}\bar{B}C_{in} + \bar{A}B\bar{C}_{in} + A\bar{B}\bar{C}_{in} + ABC_{in}$$

$$C_{out} = \bar{A}BC_{in} + A\bar{B}C_{in} + AB\bar{C}_{in} + ABC_{in}$$

$$S = A \oplus B \oplus C_{in}$$

$$C_{out} = AC_{in} + BC_{in} + AB$$

Inputs			Sum	Carry
A	B	C_{in}	S	C_{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Fig 4: Truth table

BINARY SUBTRACTORS

Half Subtractor

A Half-subtractor is a combinational circuit with two inputs A and B and two outputs difference(d) and barrow(b).

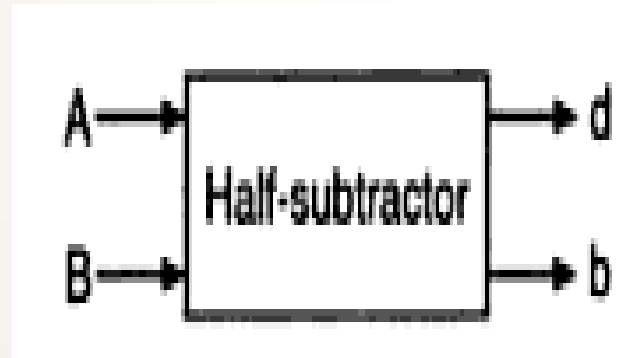


Fig 5:Block diagram

$$d = A'B + AB' = A \oplus B$$

$$b = A'B$$

Inputs		Outputs	
A	B	d	b
0	0	0	0
1	0	1	0
1	1	0	0
0	1	1	1

Fig 6: Truth table

BINARY SUBTRACTORS

86

Full subtractor

The full subtractor perform subtraction of three input bits: the minuend , subtrahend , and borrow in and generates two output bits difference and borrow out .

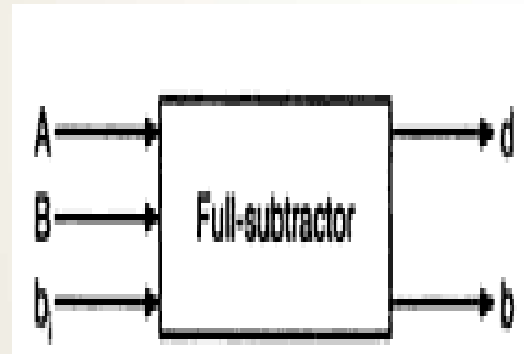


Fig 7:Block diagram

$$d = \bar{A}\bar{B}b_i + \bar{A}B\bar{b}_i + A\bar{B}\bar{b}_i + ABb_i = A \oplus B \oplus b_i$$

$$b = \bar{A}\bar{B}b_i + \bar{A}B\bar{b}_i + \bar{A}Bb_i + AB\bar{b}_i = \bar{A}B + (A \oplus B)b_i$$

Inputs			Difference	Borrow
A	B	b_i	d	b
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Fig 8: Truth table

PARALLEL ADDER AND SUBTRACTOR

A binary parallel adder is a digital circuit that adds two binary numbers in parallel form and produces the arithmetic sum of those numbers in parallel form.

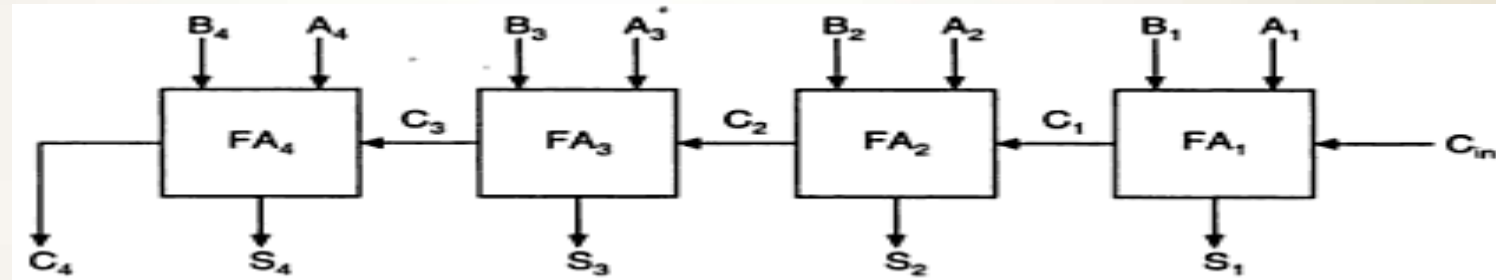


Fig 9:parallel adder

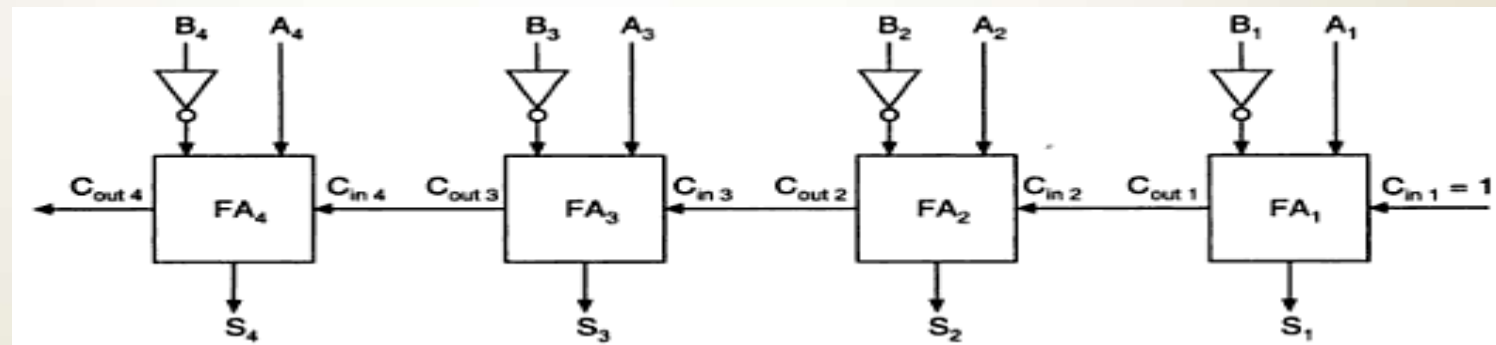


Fig 10:parallel subtractor

CARRY LOOK-A-HEAD ADDER

- In parallel-adder , the speed with which an addition can be performed is governed by the time required for the carries to propagate or ripple through all of the stages of the adder.
- The look-ahead carry adder speeds up the process by eliminating this ripple carry delay.

$$S_n = P_n \oplus C_n \text{ where } P_n = A_n \oplus B_n$$

$$C_{00} = C_{n+1} = G_n + P_n C_n \text{ where } G_n = A_n \cdot B_n$$

CARRY LOOK-A-HEAD ADDER

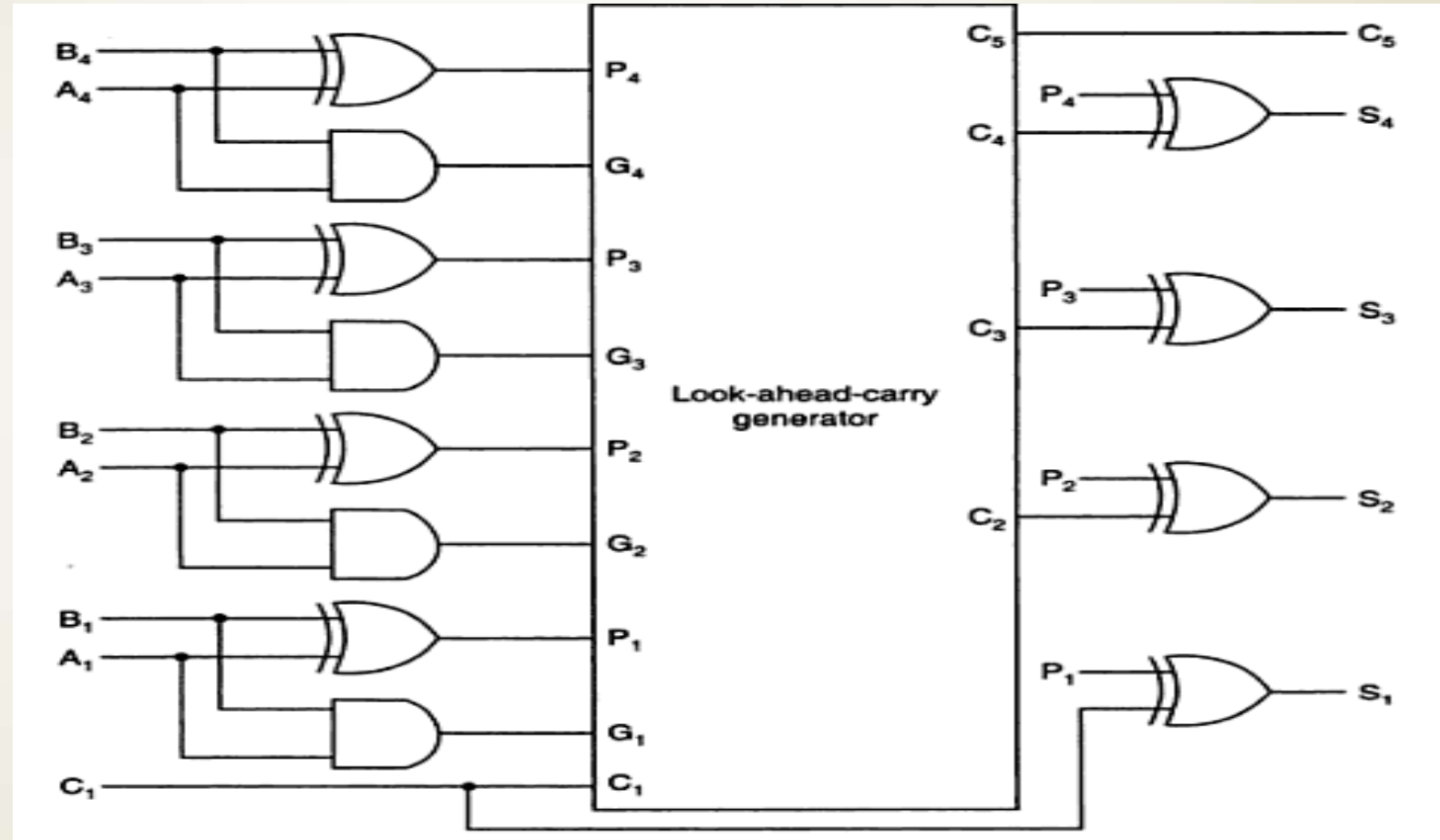


Fig:1 block diagram

BINARY MULTIPLIER

- We can also make an $n \times m$ “block” multiplier and use that to form partial products.
- Example: 2×2 – The logic equations for each partial-product binary digit are shown below
- We need to “add” the columns to get the product bits P0, P1, P2, and P3.

		b_1	b_0
		a_1	a_0
		$(a_0 \cdot b_1)$	$(a_0 \cdot b_0)$
+	$(a_1 \cdot b_1)$	$(a_1 \cdot b_0)$	
	P_3	P_2	P_1
	P_2	P_1	P_0

BINARY MULTIPLIER

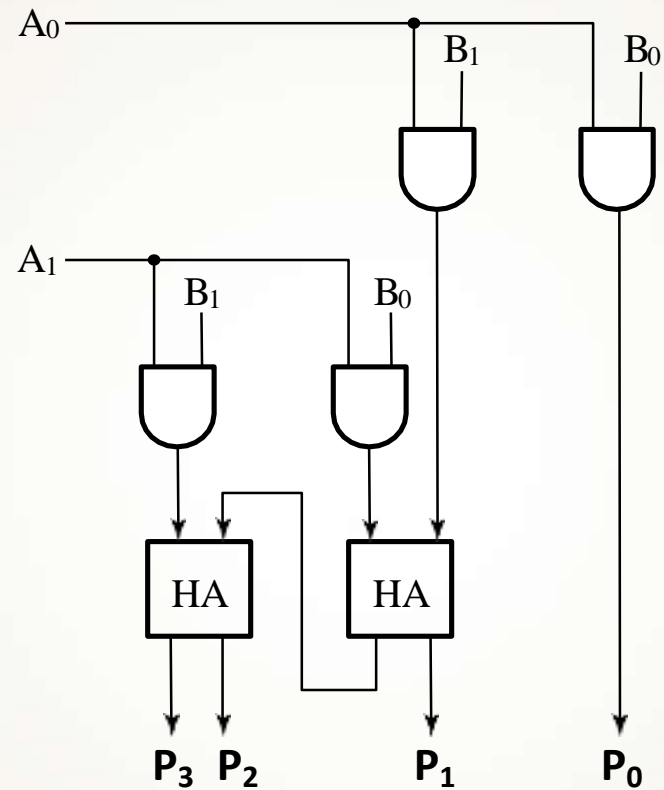


Fig 1: 2 x 2 multiplier array

MAGNITUDE COMPARATOR

93

Magnitude comparator takes two numbers as input in binary form and determines whether one number is greater than, less than or equal to the other number.

1-Bit Magnitude Comparator

A comparator used to compare two bits is called a single bit comparator.

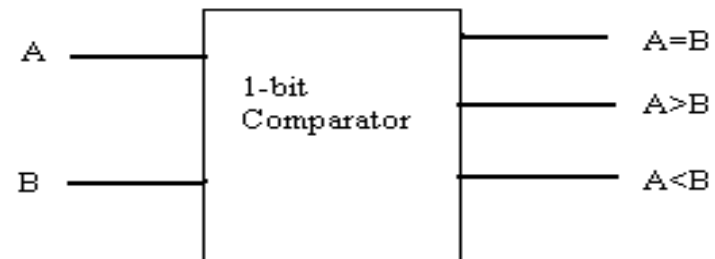


Fig :1 Block diagram

MAGNITUDE COMPARATOR

Inputs		Outputs		
<i>A</i>	<i>B</i>	<i>A > B</i>	<i>A = B</i>	<i>A < B</i>
0	0	0	1	0
0	1	0	0	1
1	0	1	0	0
1	1	0	1	0

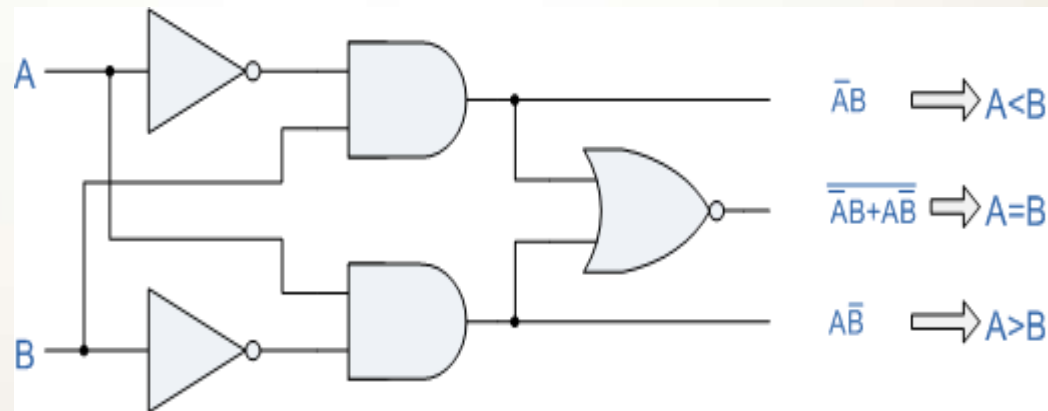


Fig 2: Logic diagram of 1-bit comparator

MAGNITUDE COMPARATOR

- 2 Bit magnitude comparator

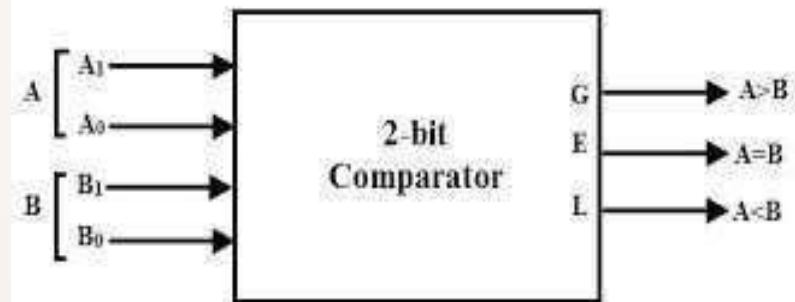


Fig :3 Block diagram

Inputs				Outputs		
A_1	A_0	B_1	B_0	A>B	A=B	A<B
0	0	0	0	0	1	0
0	0	0	1	0	0	1
0	0	1	0	0	0	1
0	0	1	1	0	0	1
0	1	0	0	1	0	0
0	1	0	1	0	1	0
0	1	1	0	0	0	1
0	1	1	1	0	0	1
1	0	0	0	1	0	0
1	0	0	1	1	0	0
1	0	1	0	0	1	0
1	0	1	1	0	0	1
1	1	0	0	1	0	0
1	1	0	1	1	0	0
1	1	1	0	1	0	0
1	1	1	1	0	1	0

Fig :4 Truth table

MAGNITUDE COMPARATOR

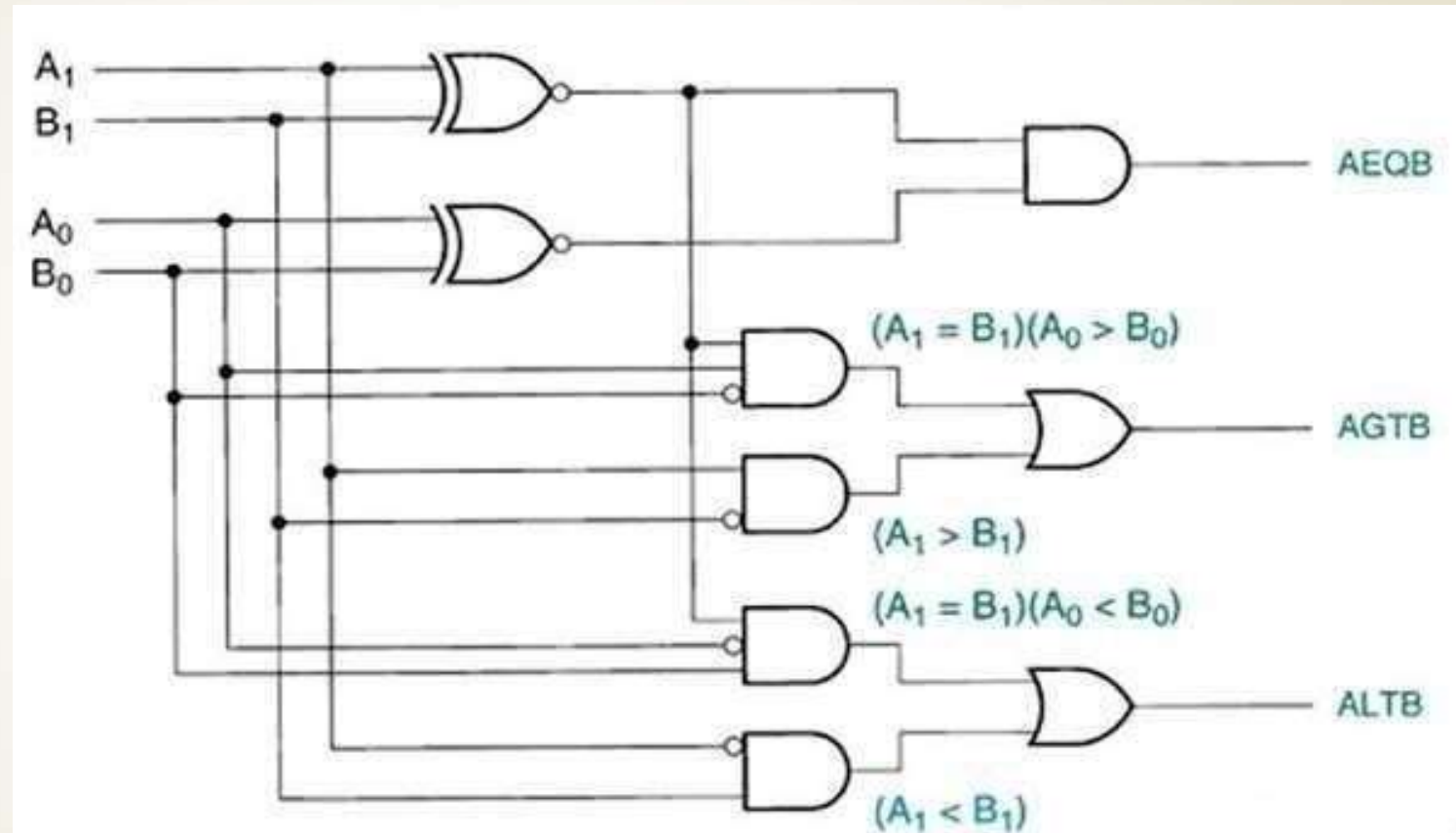


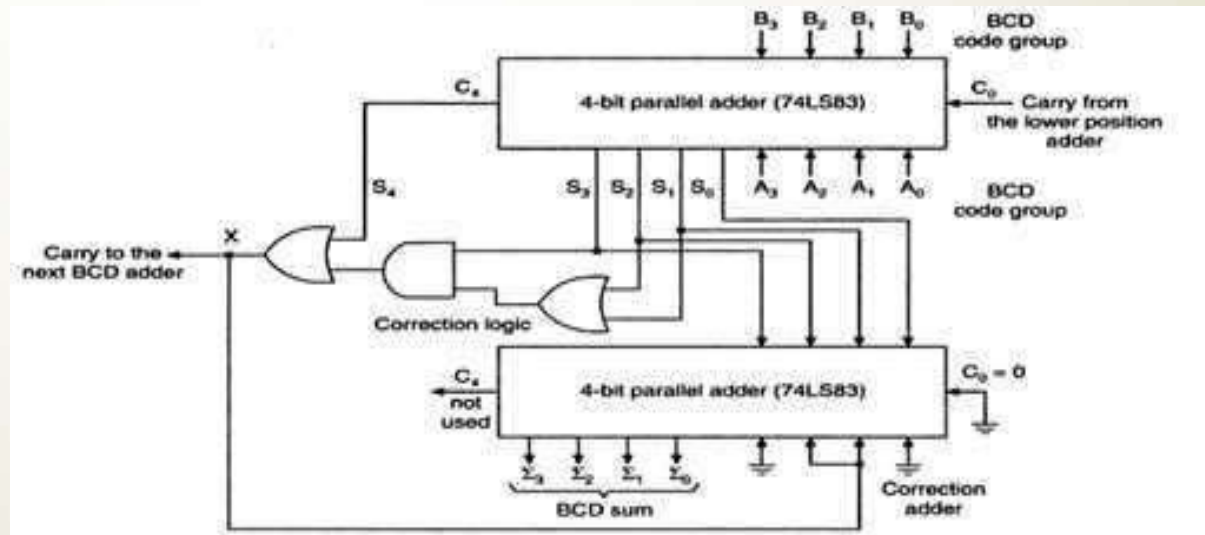
Fig 5: Logic diagram of 2-bit comparator

BCD ADDER

97

BCD Adder

- Perform the addition of two decimal digits in BCD, together with an input carry from a previous stage.
- When the sum is 9 or less, the sum is in proper BCD form and no correction is needed.
- When the sum of two digits is greater than 9, a correction of 0110 should be added to that sum, to produce the proper BCD result. This will produce a carry to be added to the next decimal position.



DECODER

- A binary decoder is a combinational logic circuit that converts binary information from the n coded inputs to a maximum of 2^n unique outputs.
- We have following types of decoders $2 \times 4, 3 \times 8, 4 \times 16, \dots$

2x4 decoder

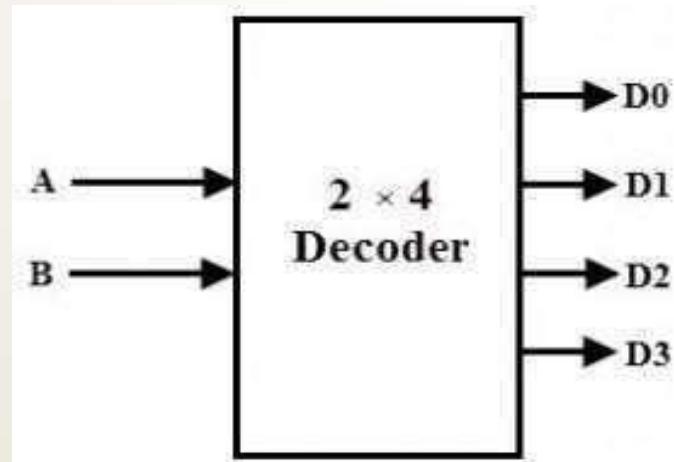


Fig 1: Block diagram

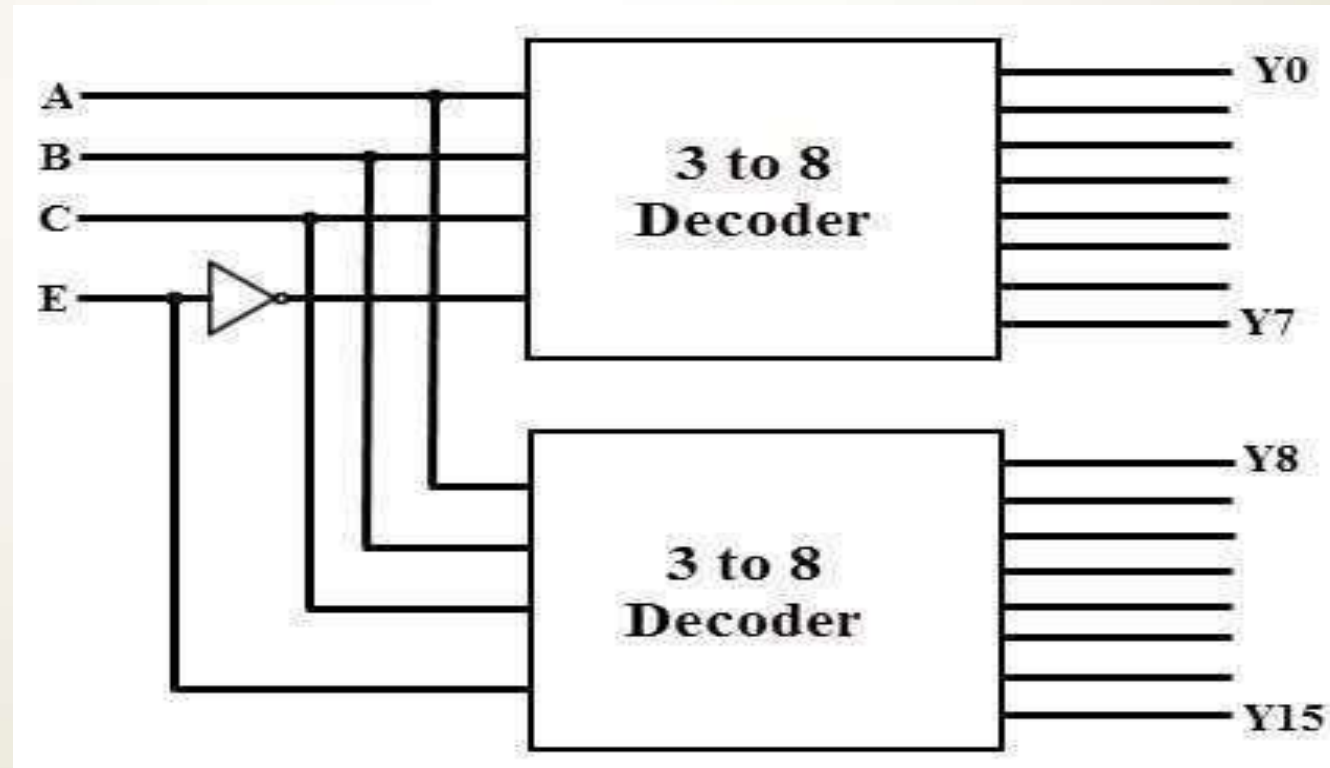
Inputs		Output			
A	B	D ₀	D ₁	D ₂	D ₃
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

Fig 2: Truth table

DECODERS

Higher order decoder implementation using lower order:

Ex: 4x16 decoder using 3x8 decoders



ENCODERS

100

- 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.

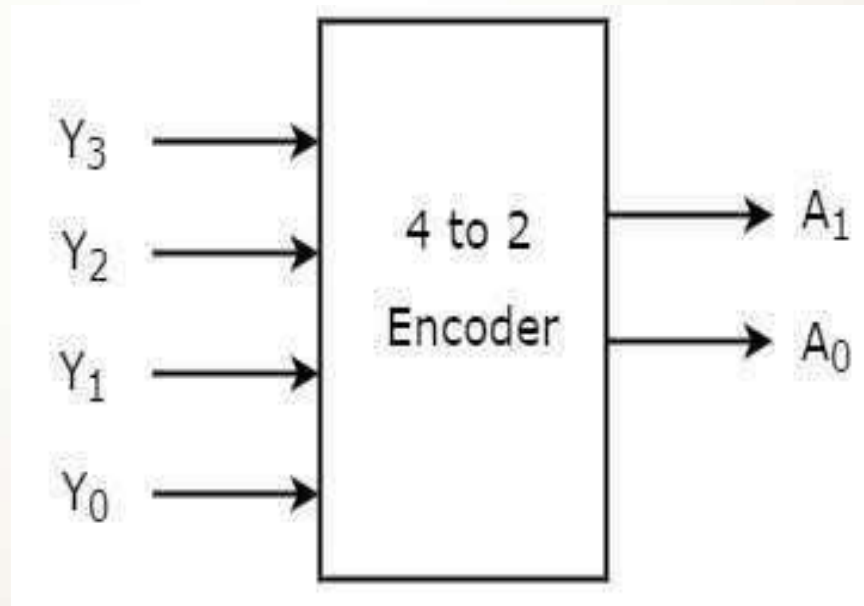


Fig 1: block diagram of 4x2 encoder

ENCODERS

Octal to binary encoder

Octal digits		Binary		
		A_2	A_1	A_0
D_0	0	0	0	0
D_1	1	0	0	1
D_2	2	0	1	0
D_3	3	0	1	1
D_4	4	1	0	0
D_5	5	1	0	1
D_6	6	1	1	0
D_7	7	1	1	1

Fig 2: Truth table

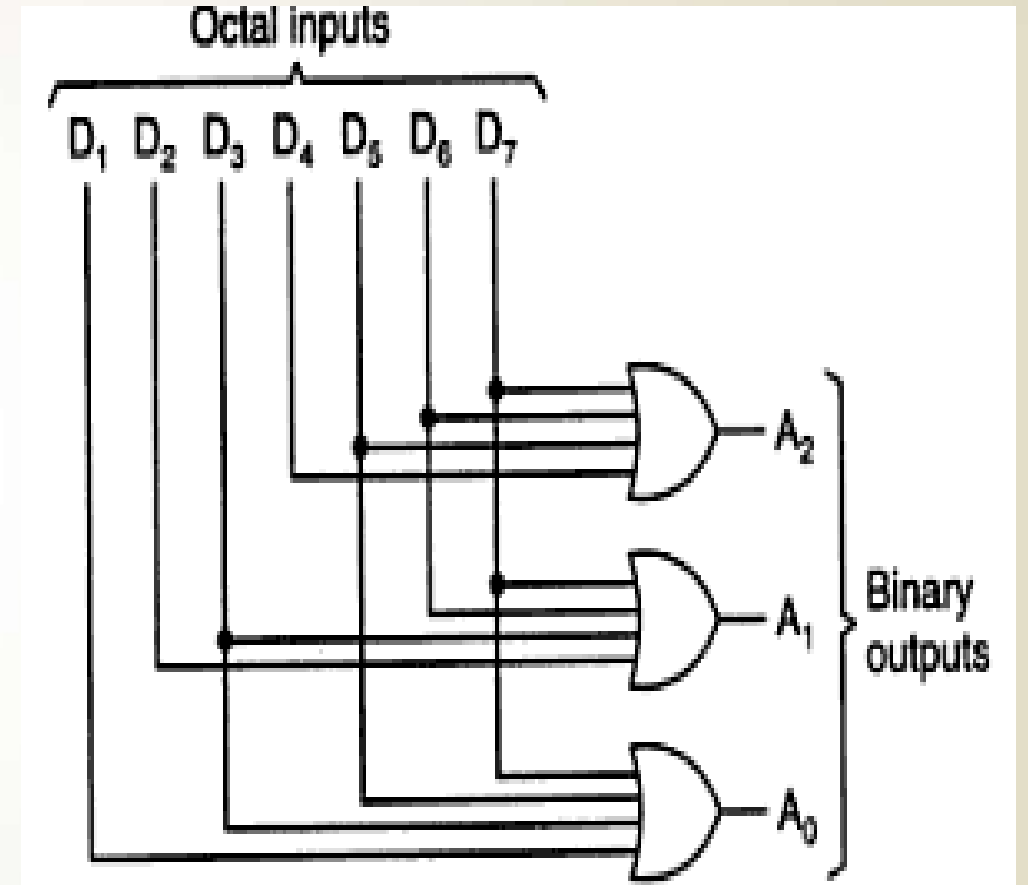


Fig 3: Logic diagram

ENCODER

102

Priority encoder

A 4 to 2 priority encoder has four inputs Y_3 , Y_2 , Y_1 & Y_0 and two outputs A_1 & A_0 . Here, the input, Y_3 has the highest priority, whereas the input, Y_0 has the lowest priority.

Inputs				Outputs		
Y_3	Y_2	Y_1	Y_0	A_1	A_0	V
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	x	0	1	1
0	1	x	x	1	0	1
1	x	x	x	1	1	1

Fig 4: Truth table

MULTIPLEXERS

- Multiplexer is a combinational circuit that has maximum of 2^n data 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.
- We have different types of multiplexers $2 \times 1, 4 \times 1, 8 \times 1, 16 \times 1, 32 \times 1, \dots$

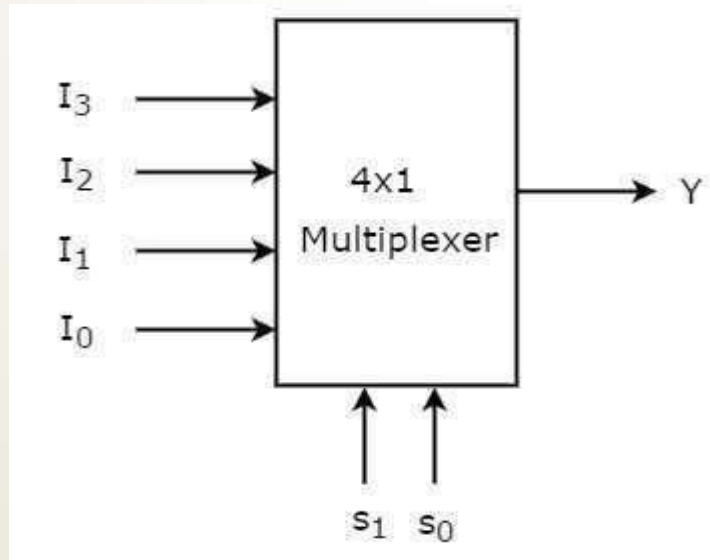


Fig 1: Block diagram

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

Fig 2: Truth table

MULTIPLEXERS

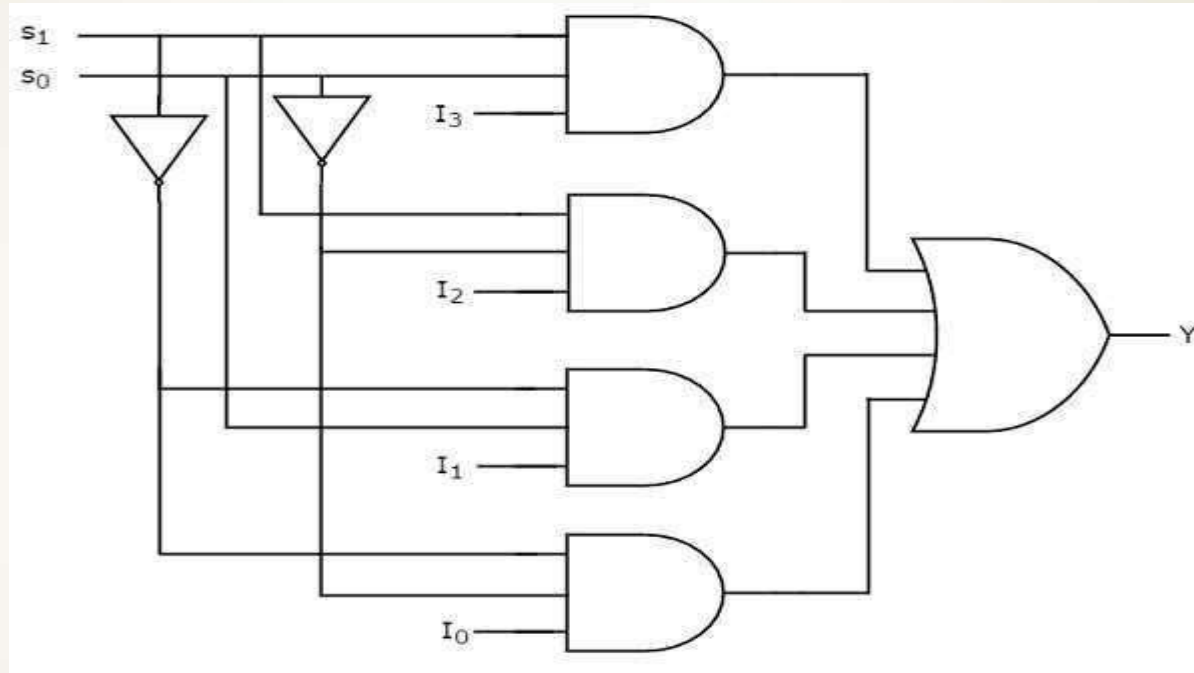


Fig 3: Logic diagram

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

MULTIPLEXERS

- **Ex: 8x1 Multiplexer**

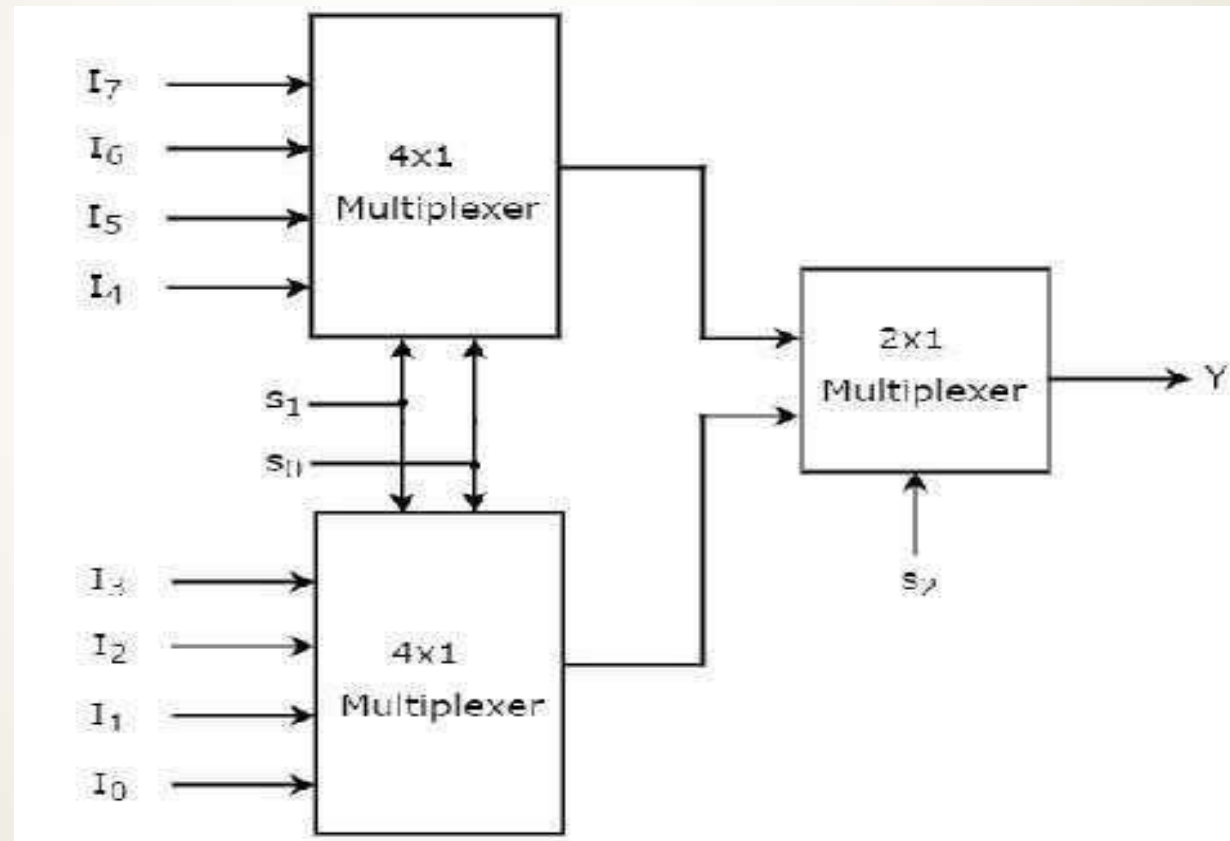
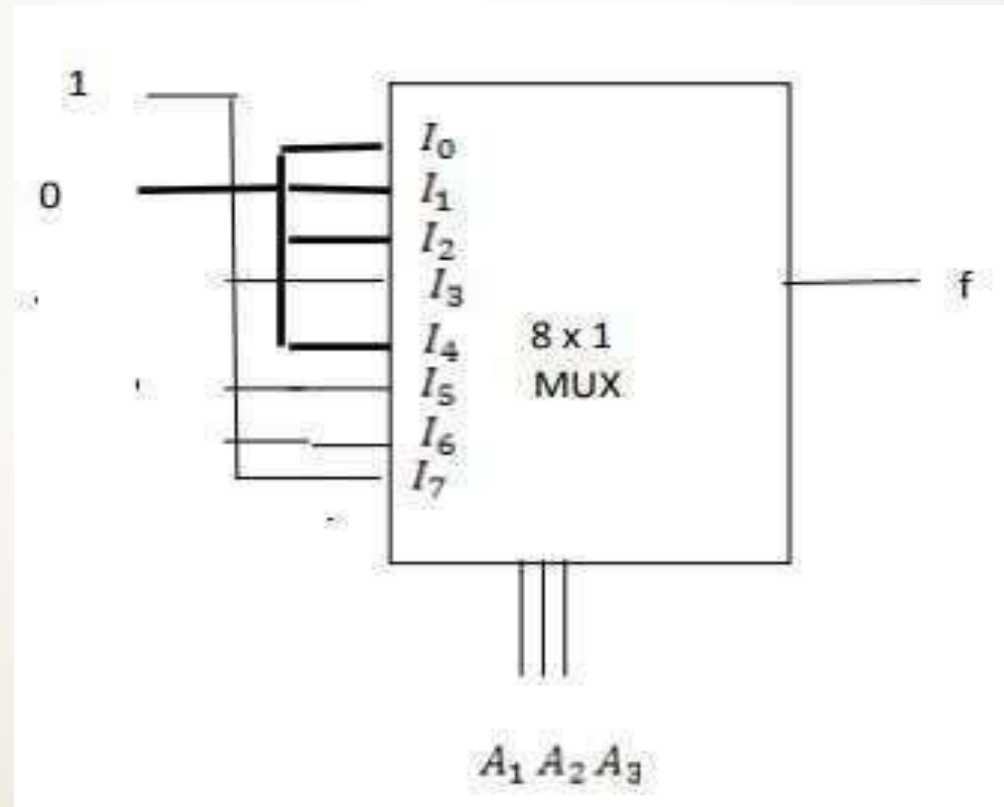


Fig 3: 8x1 Multiplexer diagram

MULTIPLEXERS

- Implementation of Boolean function using multiplexer
- $f(A_1, A_2, A_3) = \Sigma(3, 5, 6, 7)$ implementation using 8x1 mux



MULTIPLEXERS

107

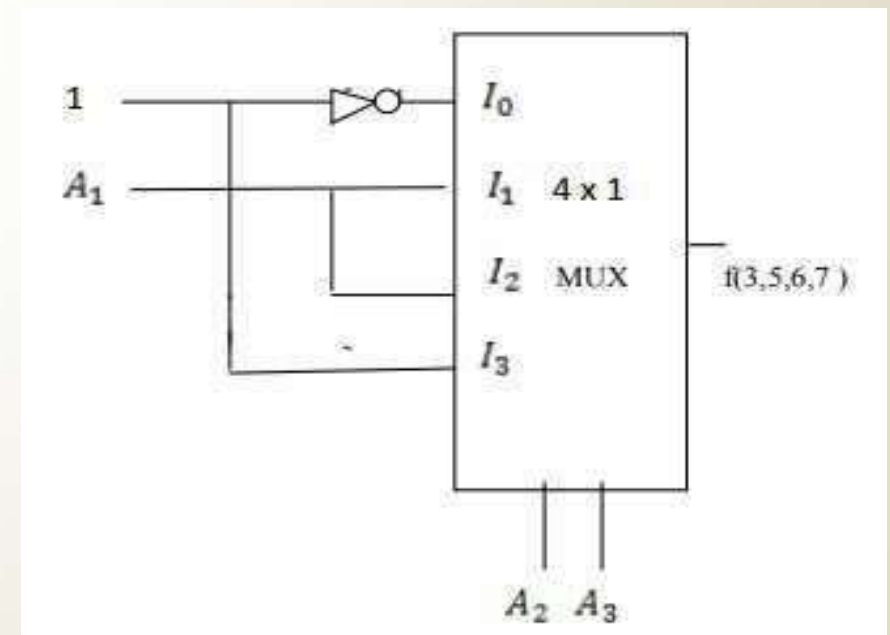
$f(A_1, A_2, A_3) = \Sigma(3,5,6,7)$ implementation using 4x1 mux

Method:1

Minterms	A_1	A_2	A_3	f
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

Fig 1: Truth table

	I_0	I_1	I_2	I_3
$\overline{A_1}$	0	1	2	3
A_1	4	5	6	7
	0	A_1	A_1	1



MULTIPLEXERS

Method:2

Minterm	A_1	A_2	A_3	f	
0	0	0	0	0	
1	0	0	1	0	$f=0$ I_0
2	0	1	0	0	
3	0	1	1	1	$f=A_3$ I_1
4	1	0	0	0	
5	1	0	1	1	$f=A_3$ I_2
6	1	1	0	1	
7	1	1	1	1	$f=1$ I_3

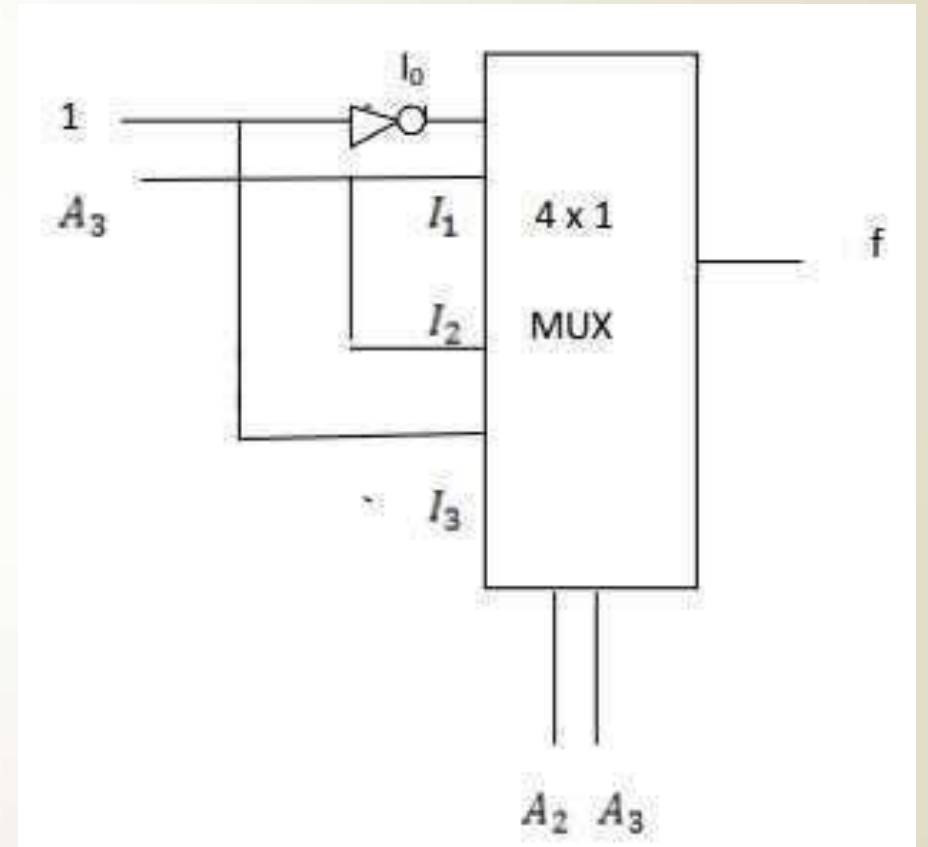


Fig 1: Truth table

DEMULTIPLEXER

- A demultiplexer is a device that takes a single input line and routes it to one of several digital output lines.
- A demultiplexer of 2^n outputs has n select lines, which are used to select which output line to send the input.
- We have $1 \times 2, 1 \times 4, 8 \times 1 \dots$ Demultiplexers.

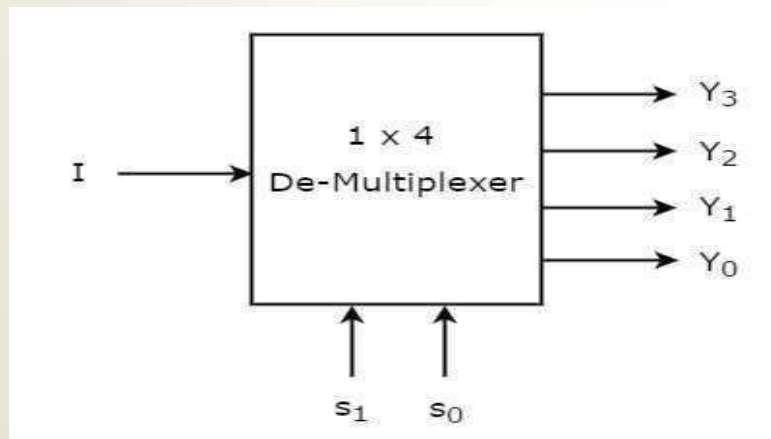


Fig:1 Block diagram

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

Fig :2 Truth table

DEMULTIPLEXER

Boolean functions for each output as

$$Y_3 = s_1 s_0 I$$

$$Y_2 = s_1 s_0' I$$

$$Y_1 = s_1' s_0 I$$

$$Y_0 = s_1' s_0' I$$

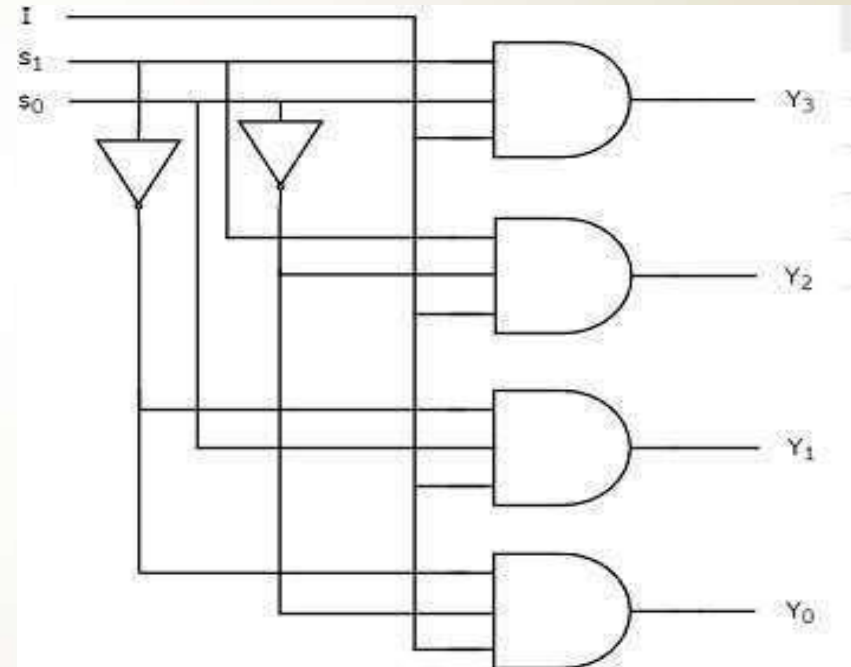


Fig:3 Logic diagram

CODE CONVERTERS

111

A code converter is a logic circuit whose inputs are bit patterns representing numbers (or character) in one code and whose outputs are the corresponding representation in a different code.

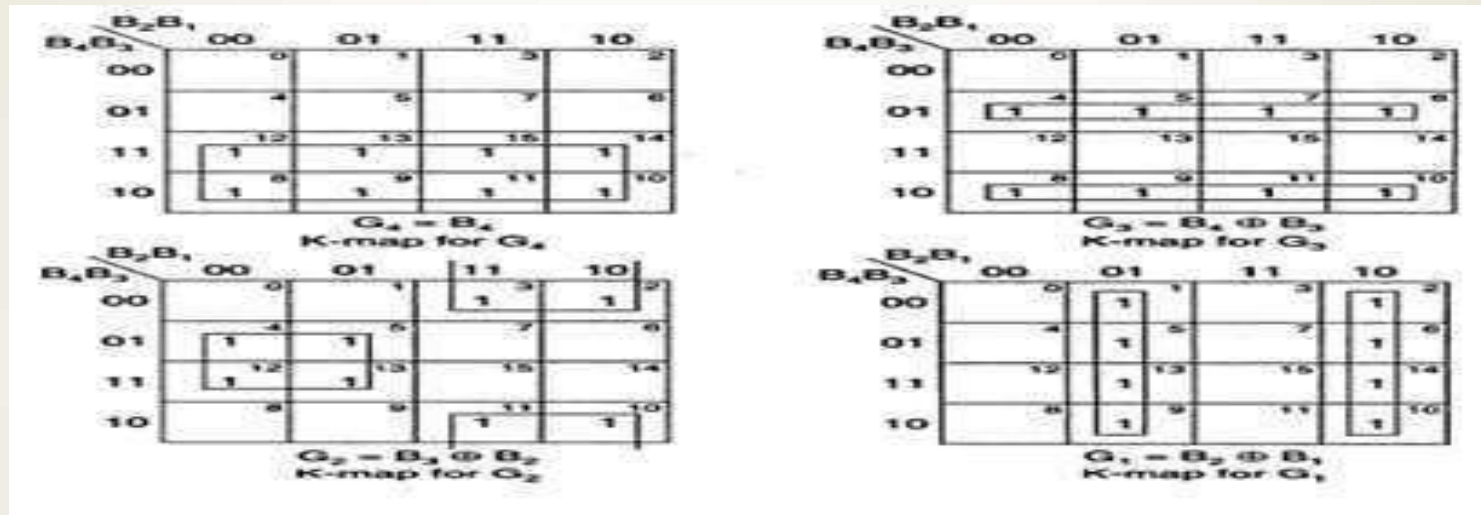
Design of a 4-bit binary to gray code converter

4-bit binary				4-bit Gray			
B ₄	B ₃	B ₂	B ₁	G ₄	G ₃	G ₂	G ₁
0	0	0	0	0	0	0	0
0	0	0	1	0	0	1	1
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	0	1	0	0
0	1	1	0	0	0	1	1
0	1	1	1	0	0	0	0
1	0	0	0	1	0	0	0
1	0	0	1	1	0	1	1
1	0	1	0	1	0	0	0
1	0	1	1	1	0	1	1
1	1	0	0	1	1	1	0
1	1	0	1	1	1	0	1
1	1	1	0	1	1	1	0
1	1	1	1	1	1	0	1

Fig :1 Truth table

CODE CONVERTERS

K-map simplification



$$G_4 = \Sigma m(8, 9, 10, 11, 12, 13, 14, 15)$$

$$G_3 = \Sigma m(4, 5, 6, 7, 8, 9, 10, 11)$$

$$G_2 = \Sigma m(2, 3, 4, 5, 10, 11, 12, 13)$$

$$G_1 = \Sigma m(1, 2, 5, 6, 9, 10, 13, 14)$$

$$G_4 = B_4$$

$$G_3 = \bar{B}_4 B_3 + B_4 \bar{B}_3 = B_4 \oplus B_3$$

$$G_2 = \bar{B}_3 B_2 + B_3 \bar{B}_2 = B_3 \oplus B_2$$

$$G_1 = \bar{B}_2 B_1 + B_2 \bar{B}_1 = B_2 \oplus B_1$$

CODE CONVERTERS

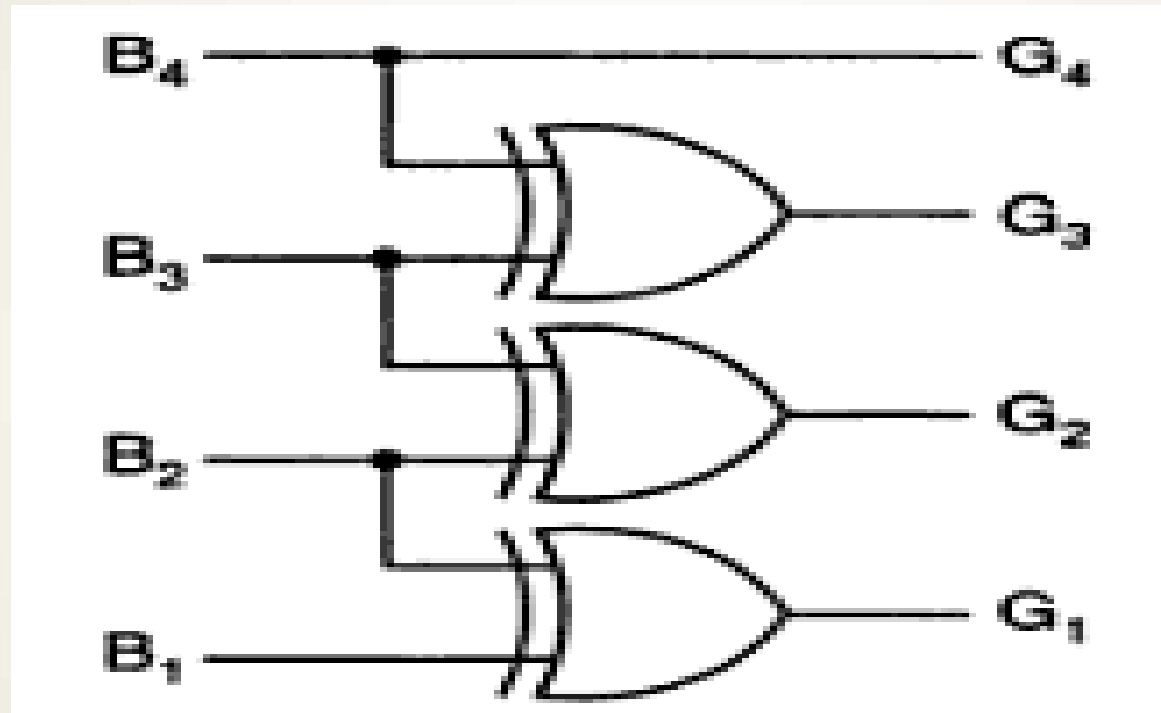


Fig: 2 Logic diagram

UNIT 5

INTRODUCTION TO SEQUENTIAL LOGIC CIRCUITS

SEQUENTIAL LOGIC CIRCUITS

Sequential logic circuit consists of a combinational circuit with storage elements connected as a feedback to combinational circuit

- output depends on the sequence of inputs (past and present)
- stores information (state) from past inputs

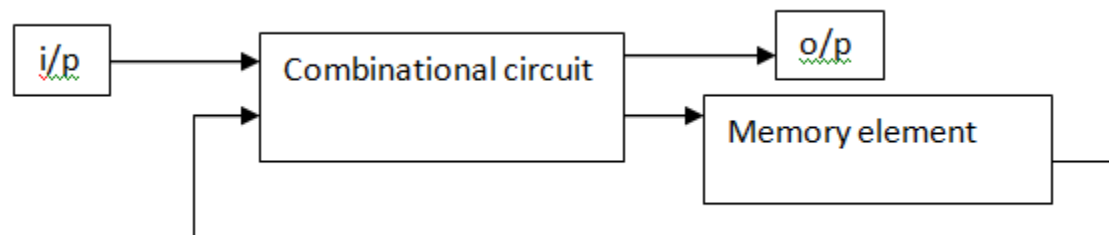
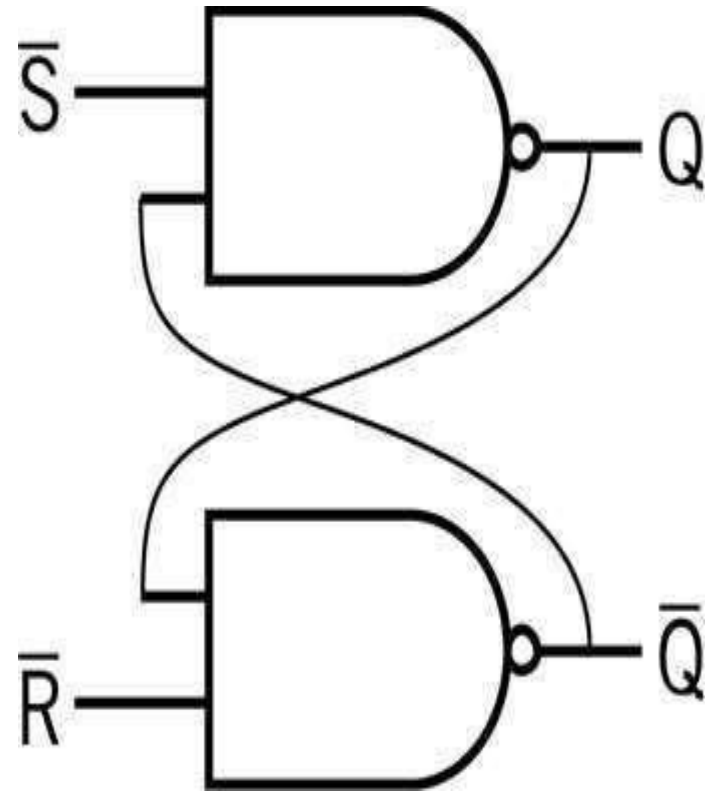


Figure 1: Sequential logic circuits

SEQUENTIAL LOGIC CIRCUITS

- Output depends on
 - Input
 - Previous **state** of the circuit
- **Flip-flop**: basic memory element
- **State table**: output for all combinations of input and previous states (Truth Table)



SEQUENTIAL LOGIC CIRCUITS

1. Sequential circuit receives the binary information from external inputs and with the present state of the storage elements together determine the binary value of the outputs.
2. The output in a sequential circuit are a function of not only the inputs, but also the present state of the storage elements.
3. The next state of the storage elements is also a function of external inputs and the present state.
4. There are two main types of sequential circuits
 1. synchronous sequential circuits
 2. asynchronous sequential circuits

SEQUENTIAL LOGIC CIRCUITS

Synchronous sequential circuits

It is a system whose behavior can be defined from the knowledge of its signals at discrete instants of time

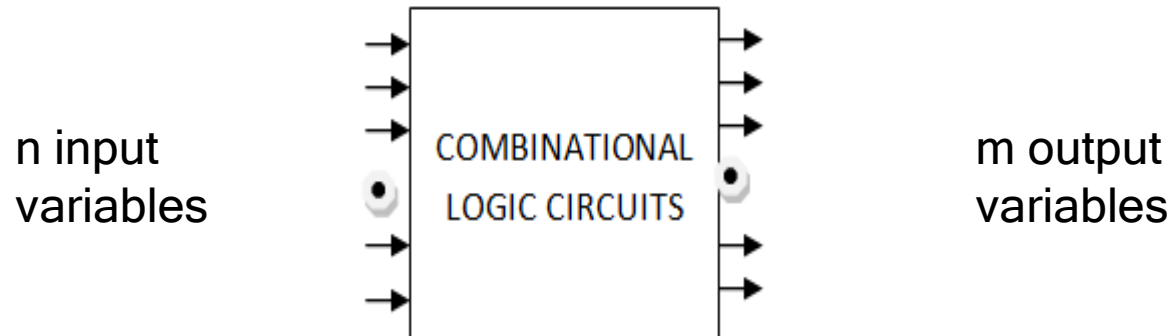
Asynchronous sequential circuits

It depends upon the input signals at any instant of time and the order in which the input changes

Combinational vs. Sequential

COMBINATIONAL LOGIC CIRCUIT

Combinational logic circuit consists of input variables, logic gates and output variables. The logic gate accepts signals from the inputs and generates signals to the outputs.



For n input variables there are 2^n possible combinations of binary input variables

Combinational vs. Sequential

SEQUENTIAL LOGIC CIRCUITS :

Sequential logic circuit consists of a combinational circuit with storage elements connected as a feedback to combinational circuit

- output depends on the sequence of inputs (past and present)
- stores information (state) from past inputs

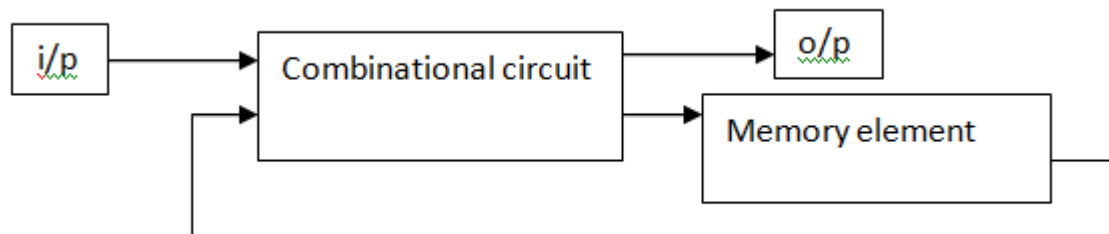


Figure 1: Sequential logic circuits

Combinational vs. Sequential

Combinational Circuit

always gives the same output for a given set of inputs ex: adder always generates sum and carry, regardless of previous inputs

Sequential Circuit

- stores information
- output depends on stored information (state) plus input so a given input might produce different outputs, depending on the stored information
- *example*: ticket counter
- advances when you push the button output depends on previous state
- useful for building —memory‖ elements and —state machines‖

Combinational Vs Sequential

<u>Combinational Logic Circuits</u>	<u>Sequential Logic Circuits</u>
Output is a function of the present inputs (Time Independent Logic).	Output is a function of clock, present inputs and the previous states of the system.
Do not have the ability to store data (state).	Have memory to store the present states that is sent as control input (enable) for the next operation.
It does not require any feedback. It simply outputs the input according to the logic designed.	It involves feedback from output to input that is stored in the memory for the next operation.
Used mainly for Arithmetic and Boolean operations.	Used for storing data (and hence used in RAM).
Logic gates are the elementary building blocks.	Flip flops (binary storage device) are the elementary building unit.
Independent of clock and hence does not require triggering to operate.	Clocked (Triggered for operation with electronic pulses).
Example: Adder [$1+0=1$; Dependency only on present inputs i.e., 1 and 0].	Example: Counter [Previous O/P +1=Current O/P; Dependency on present input as well as previous state].

LATCHES

STORAGE ELEMENTS :

Storage elements in a digital circuit can maintain a binary state indefinitely, until directed by an input signal to switch states. The major difference among various storage elements are the number of input they possess and the manner in which the inputs affect the binary state. There are two types of storage elements

1. Latches

2. Flipflops

Storage elements that operate with signal level are referred as latch and those controlled by a clock transition are referred as flipflops.

1. LATCHES:

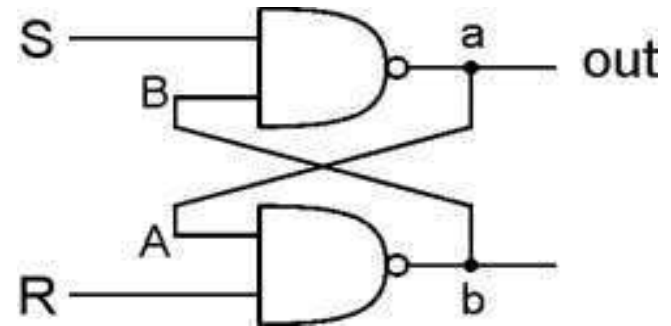
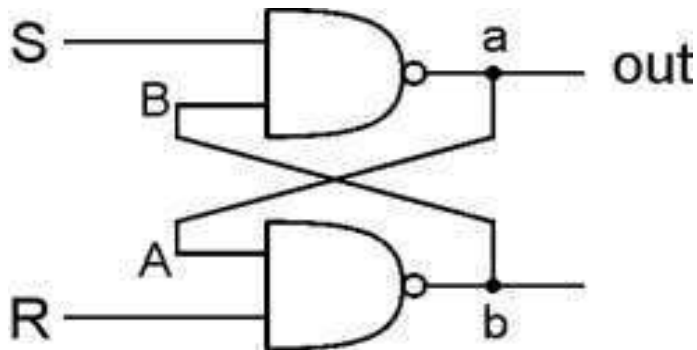
A latch has a feedback path, so information can be retained by the device. Therefore latches can be memory devices, and can store one bit of data for as long as the device is powered. As the name suggests, latches are used to "latch onto" information and hold in place. Latches are very similar to [flip-flops](#), but are not synchronous devices, and do not operate on clock edges as flip-flops do. Latch is a level sensitive device. Latch is a monostable multivibrator

2. FLIPFLOPS:

A **flip-flop** is a circuit that has two stable states and can be used to store state information. A flip-flop is a bistable multivibrator. The circuit can be made to change state by signals applied to one or more control inputs and will have one or two outputs. It is the basic storage element in sequential logic. Flip-flops and latches are fundamental building blocks of digital electronics systems used in computers, communications, and many other types of systems. Flipflop is a edge sensitive device

SR LATCH :

An **SR latch** (Set/Reset) is an asynchronous device: it works independently of control signals and relies only on the state of the S and R inputs. In the image we can see that an SR flip-flop can be created with two NOR gates that have a cross-feedback loop. SR latches can also be made from NAND gates, but the inputs are swapped and negated. In this case, it is sometimes called an **SR latch**.



R is used to **—reset** or **—clear** the element – set it to zero. S is used to **—set** the element – set it to one.

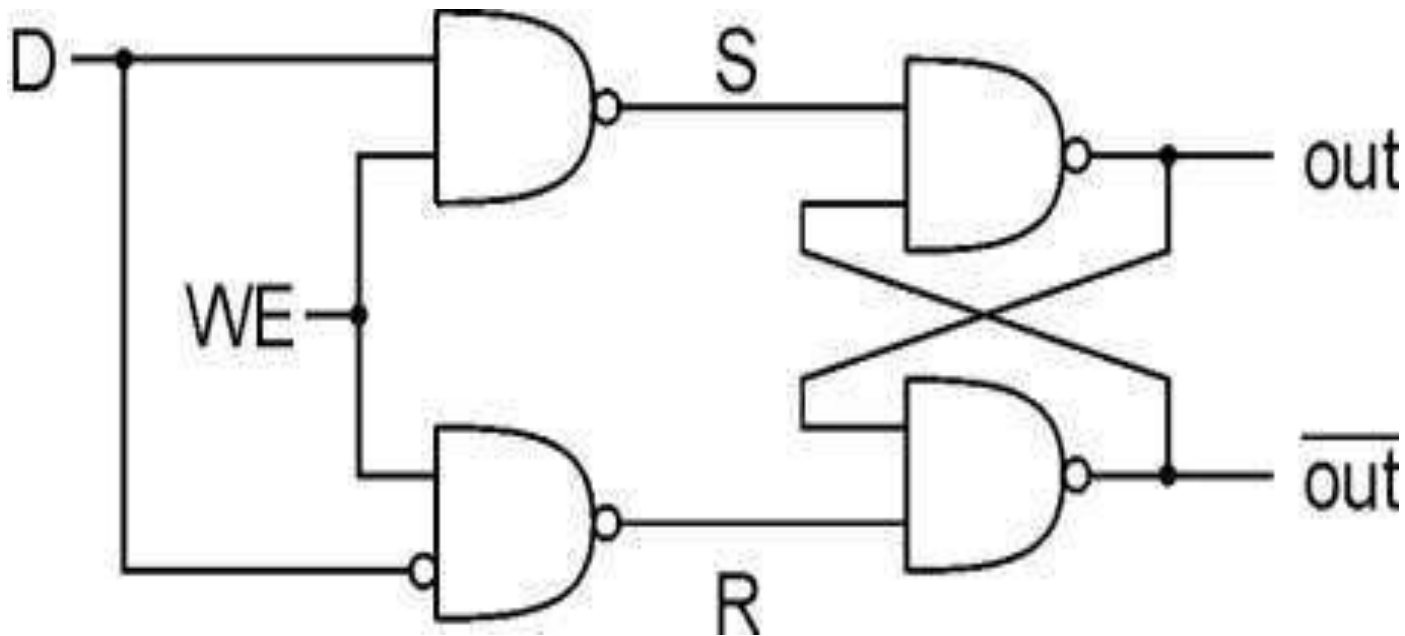
If both R and S are one, out could be either zero or one. **—quiescent** state -- holds its previous value. note: if a is 1, b is 0, and vice versa

GATED D-LATCH :

The **D latch** (D for "data") or **transparent latch** is a simple extension of the gated SR latch that removes the possibility of invalid input states. Two inputs: D (data) and WE (write enable)

when $WE = 1$, latch is set to value of D $S = \text{NOT}(D)$, $R = D$

when $WE = 0$, latch holds previous value $S = R = 1$



Flip Flops

A flip flop is an electronic circuit with two stable states that can be used to store binary data. The stored data can be changed by applying varying inputs. Flip-flops and latches are fundamental building blocks of digital electronics systems used in computers, communications, and many other types of systems. Flip-flops and latches are used as [data storage](#) elements. There are 4 types of flipflops

1.RS flip flop

2.Jk flip flop

3.D flip flop

4.T flip flop

Applications of Flip-Flops

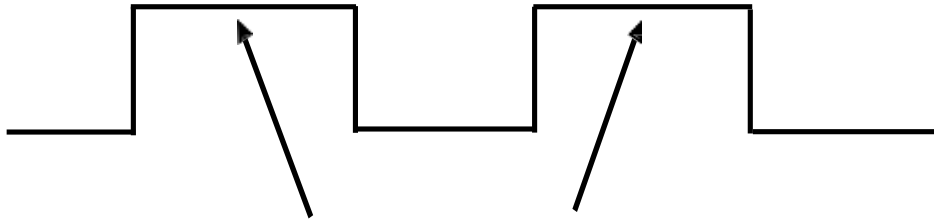
These are the various types of flip-flops being used in digital electronic circuits and the applications like Counters, Frequency Dividers, Shift Registers, Storage Registers

EDGE-TRIGGERED FLIP FLOPS

Characteristics

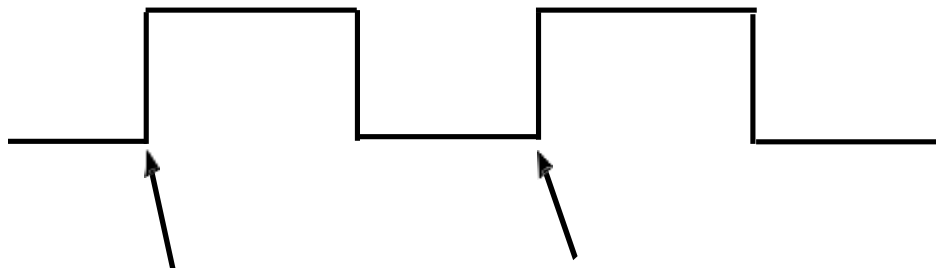
- State transition occurs at the rising edge or falling edge of the clock pulse

Latches



respond to the input only during these periods

Edge-triggered Flip Flops (positive)

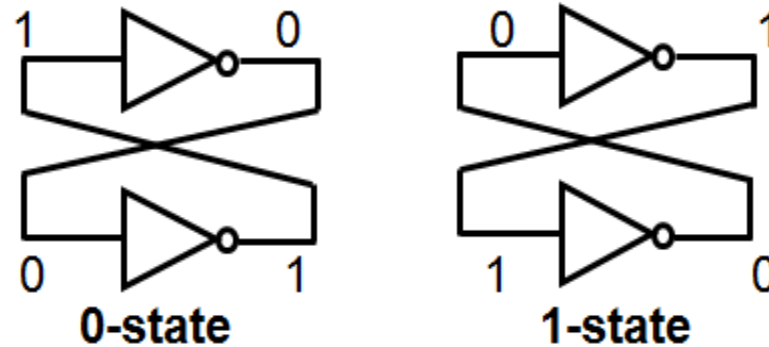


respond to the input only at this time

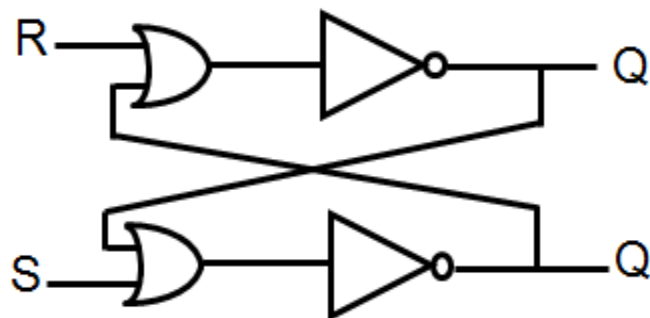
FLIP-FLOPS

Characteristics

- 2 stable states
- Memory capability
- Operation is specified by a Characteristic Table



In order to be used in the computer circuits, state of the flip flop should have input terminals and output terminals so that it can be set to a certain state, and its state can be read externally.



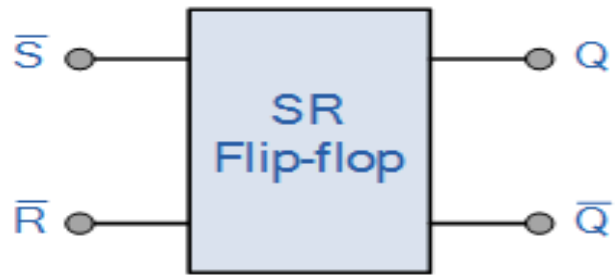
S	R	Q(t+1)
0	0	Q(t)
0	1	0
1	0	1
1	1	indeterminate (forbidden)

SR Flip-Flop

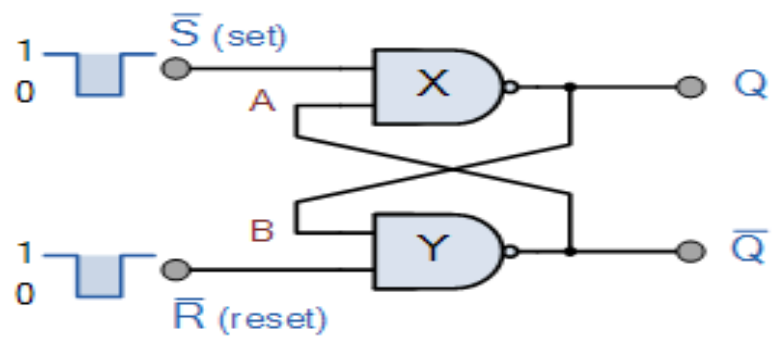
The **SR flip-flop**, also known as a *SR Latch*, can be considered as one of the most basic sequential logic circuit possible. This simple flip-flop is basically a one-bit memory bistable device that has two inputs, one which will —SET the device (meaning the output = —1), and is labelled **S** and one which will —RESET the device (meaning the output = —0), labelled **R**. The reset input resets the flip-flop back to its original state with an output Q that will be either at a logic level —1 or logic —0 depending upon this set/reset condition.

A basic NAND gate SR flip-flop circuit provides feedback from both of its outputs back to its opposing inputs and is commonly used in memory circuits to store a single data bit.

Then the SR flip-flop actually has three inputs, Set, Reset and its current output Q relating to its current state or history.



Symbol



Circuit

State	S	R	Q	\bar{Q}	Description
Set	1	0	0	1	Set $\bar{Q} \gg 1$
	1	1	0	1	no change
Reset	0	1	1	0	Reset $\bar{Q} \gg 0$
	1	1	1	0	no change
Invalid	0	0	1	1	Invalid Condition

Truth Table for this Set-Reset Function

JK Flip flop

The JK Flip-flop is similar to the SR Flip-flop but there is no change in state when the J and K inputs are both LOW. The basic S-R NAND flip-flop circuit has many advantages and uses in sequential logic circuits but it suffers from two basic switching problems.

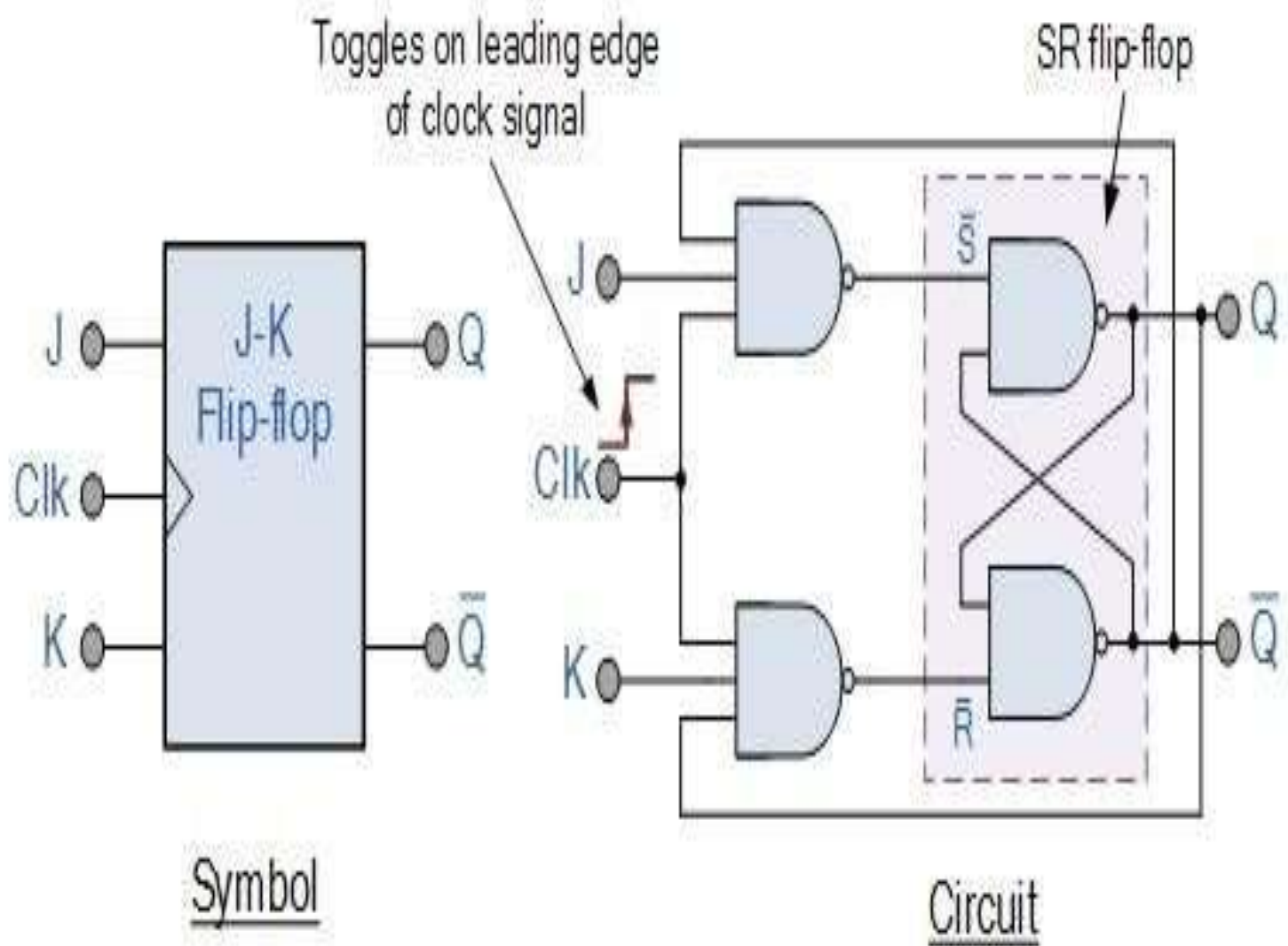
1.the Set = 0 and Reset = 0 condition ($S = R = 0$) must always be avoided

2.if Set or Reset change state while the enable (EN) input is high the correct latching action may not occur

Then to overcome these two fundamental design problems with the SR flip-flop design, the **JK flip Flop** was developed by the scientist name Jack Kirby.

The **JK flip flop** is basically a gated SR flip-flop with the addition of a clock input circuitry that prevents the illegal or invalid output condition that can occur when both inputs S and R are equal to logic level $\text{—}1\|$. Due to this additional clocked input, a JK flip-flop has four possible input combinations, $\text{—logic } 1\|$, $\text{—logic } 0\|$, $\text{—no change}\|$ and $\text{—toggle}\|$. The symbol for a JK flip flop is similar to that of an *SR Bistable Latch* as seen in the previous tutorial except for the addition of a clock input.

- ❖ Both the S and the R inputs of the previous SR bistable have now been replaced by two inputs called the J and K inputs, respectively after its inventor Jack Kilby. Then this equates to: $J = S$ and $K = R$.
- ❖ The two 2-input AND gates of the gated SR bistable have now been replaced by two 3-input NAND gates with the third input of each gate connected to the outputs at Q and \bar{Q} . This cross coupling of the SR flip-flop allows the previously invalid condition of $S = 1$ and $R = 1$ state to be used to produce a toggle action as the two inputs are now interlocked.
- ❖ If the circuit is now SET the J input is inhibited by the 0 status of Q through the lower NAND gate. If the circuit is RESET the K input is inhibited by the 0 status of \bar{Q} through the upper NAND gate. As Q and \bar{Q} are always different we can use them to control the input. When both inputs J and K are equal to logic 1, the JK flip flop toggles as shown in the following truth table.



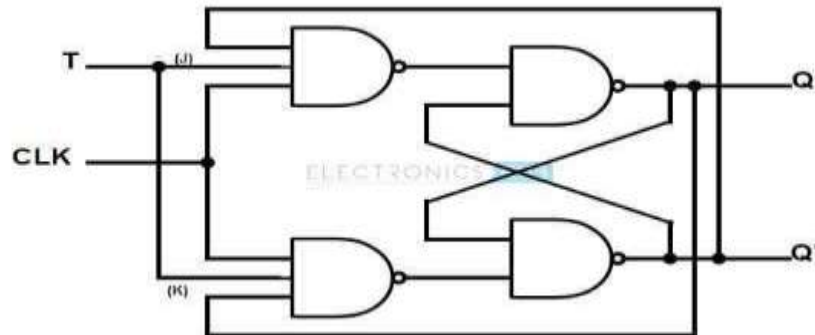
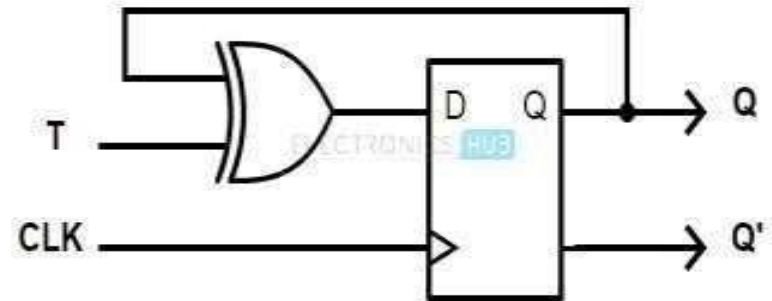
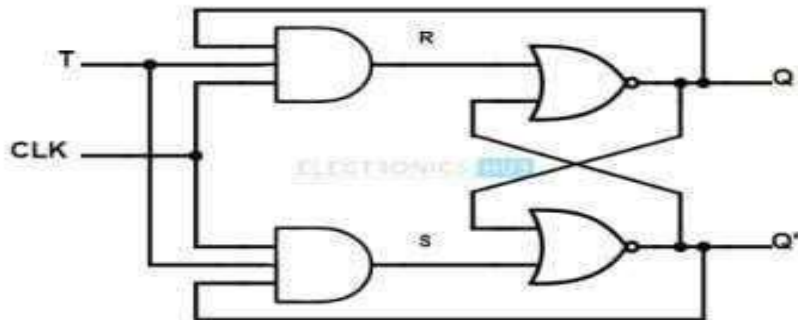
	Input		Output		Description	
	J	K	Q	\overline{Q}		
same as for the SR Latch	0	0	0	0	Memory no change	
	0	0	0	1		
	0	1	1	0	Reset Q » 0	
	0	1	0	1		
	1	0	0	1	Set Q » 1	
	1	0	1	0		
	toggle action	1	1	0	1	Toggle
		1	1	1	0	

The Truth Table for the JK Function

Then the JK flip-flop is basically an SR flip flop with feedback which enables only one of its two input terminals, either SET or RESET to be active at any one time thereby eliminating the invalid condition seen previously in the SR flip flop circuit. Also when both the J and the K inputs are at logic level 1 at the same time, and the clock input is pulsed $HIGH$, the circuit will $toggle$ from its SET state to a RESET state, or visa-versa. This results in the JK flip flop acting more like a T-type toggle flip-flop when both terminals are $HIGH$.

T FLIP FLOP

We can construct a T flip flop by any of the following methods. Connecting the output feedback to the input, in SR flip flop. Connecting the XOR of T input and Q PREVIOUS output to the Data input, in D flip flop. Hard – wiring the J and K inputs together and connecting it to T input, in JK flip – flop.



Working

T flip – flop is an edge triggered device i.e. the low to high or high to low transitions on a clock signal of narrow triggers that is provided as input will cause the change in output state of flip – flop. T flip – flop is an edge triggered device.

Truth Table of T flip – flop

	Previous		Next	
T	Q_{Prev}	Q'_{Prev}	Q_{Next}	Q'_{Next}
0	0	1	0	1
0	1	0	1	0
1	0	1	1	0
1	1	0	0	1

❖ If the output $Q = 0$, then the upper NAND is in enable state and lower NAND gate is in disable condition. This allows the trigger to pass the S inputs to make the flip – flop in SET state i.e. $Q = 1$.

❖ If the output $Q = 1$, then the upper NAND is in disable state and lower NAND gate is in enable condition. This allows the trigger to pass the R inputs to make the flip – flop in RESET state i.e. $Q = 0$.

In simple terms, the operation of the T flip – flop is

When the T input is low, then the next state of the T flip flop is same as the present state.

$T = 0$ and present state = 0 then the next state = 0

$T = 1$ and present state = 1 then the next state = 1

When the T input is high and during the positive transition of the clock signal, the next state of the T flip – flop is the inverse of present state.

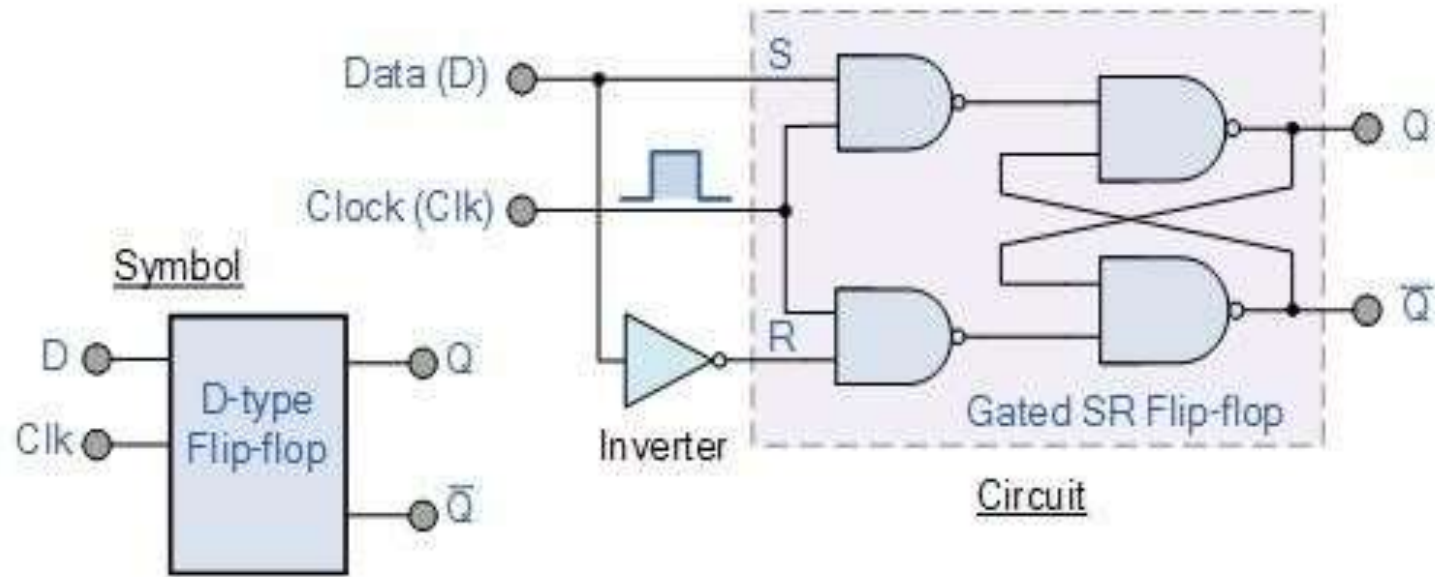
$T = 1$ and present state = 0 then the next state = 1 $T = 1$ and present state = 1 then the next state = 0

D FLIP FLOP

- The D-type flip-flop is a modified Set-Reset flip-flop with the addition of an inverter to prevent the S and R inputs from being at the same logic level
- One of the main disadvantages of the basic **SR NAND Gate Bistable** circuit is that the indeterminate input condition of $SET = \text{—}0\|$ and $RESET = \text{—}0\|$ is forbidden.
- This state will force both outputs to be at logic $\text{—}1\|$, over-riding the feedback latching action and whichever input goes to logic level $\text{—}1\|$ first will lose control, while the other input still at logic $\text{—}0\|$ controls the resulting state of the latch.

- But in order to prevent this from happening an inverter can be connected between the —SET and the —RESET inputs to produce another type of flip flop circuit known as a **Data Latch, Delay flip flop, D-type Bistable, D-type Flip Flop** or just simply a **D Flip Flop** as it is more generally called.
- The **D Flip Flop** is by far the most important of the clocked flip-flops as it ensures that inputs S and R are never equal to one at the same time.
- The D-type flip flop are constructed from a gated SR flip-flop with an inverter added between the S and the R inputs to allow for a single D (data) input.
- Then this single data input, labelled —D and is used in place of the —Set signal, and the inverter is used to generate the complementary —Reset input thereby making a level-sensitive D-type flip-flop from a level-sensitive SR-latch as now $S = D$ and $R = \text{not } D$ as shown.

D-type Flip-Flop Circuit



We remember that a simple SR flip-flop requires two inputs, one to —SET|| the output and one to —RESET|| the output. By connecting an inverter (NOT gate) to the SR flip-flop we can—SET|| and —RESET|| the flip-flop using just one input as now the two input signals are complements of each other.

- This complement avoids the ambiguity inherent in the SR latch when both inputs are LOW, since that state is no longer possible. Thus this single input is called the $\overline{\text{DATA}}$ input.
- If this data input is held HIGH the flip flop would be $\overline{\text{SET}}$ and when it is LOW the flip flop would change and become $\overline{\text{RESET}}$.
- However, this would be rather pointless since the output of the flip flop would always change on every pulse applied to this data input.

To avoid this an additional input called the —CLOCK or —ENABLE input is used to isolate the data input from the flip flop's latching circuitry after the desired data has been stored. The effect is that D input condition is only copied to the output Q when the clock input is active. This then forms the basis of another sequential device called a **D Flip Flop**.

The —D flip flop will store and output whatever logic level is applied to its data terminal so long as the clock input is HIGH. Once the clock input goes LOW the —set and —reset inputs of the flip-flop are both held at logic level —1 so it will not change state and store whatever data was present on its output before the clock transition occurred. In other words the output is —latched at either logic —0 or logic —1 .

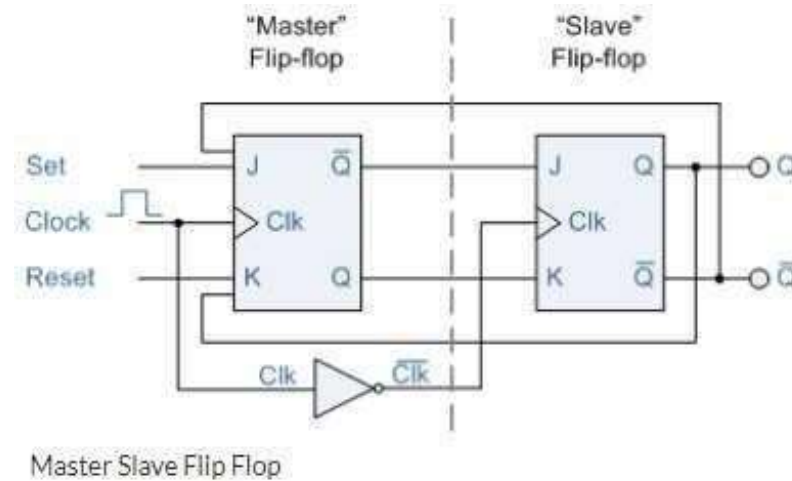
Truth Table for the D-type Flip Flop

Clk	D	Q		Description
$\downarrow \gg 0$	X	Q	\overline{Q}	Memory no change
$\uparrow \gg 1$	0	0	1	Reset Q \gg 0
$\uparrow \gg 1$	1	1	0	Set Q \gg 1

Note that: \downarrow and \uparrow indicates direction of clock pulse as it is assumed D-type flip flops are edge triggered

MASTER SLAVE FLIPFLOP :

Master-slave flip flop is designed using two separate flip flops. Out of these, one acts as the master and the other as a slave. The figure of a master-slave J-K flip flop is shown below.



From the above figure you can see that both the J-K flip flops are presented in a series connection. The output of the master J-K flip flop is fed to the input of the slave J-K flip flop. The output of the slave J-K flip flop is given as a feedback to the input of the master J-K flip flop. The clock pulse [Clk] is given to the master J-K flip flop and it is sent through a NOT Gate and thus inverted before passing it to the slave J-K flip flop.

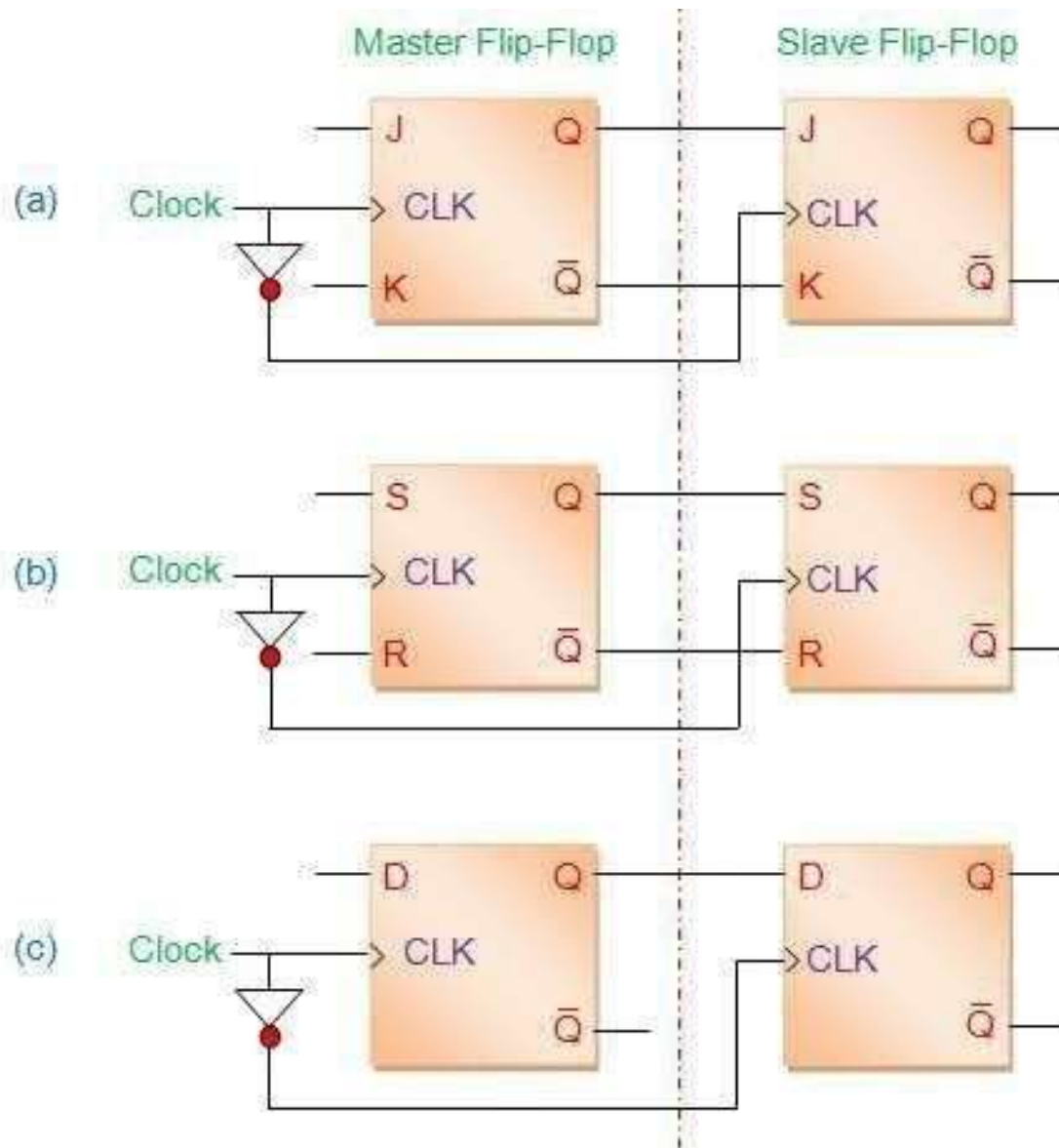


Figure 1 (a) Master-Slave JK flip-flop (b) Master-Slave SR flip-flop (c) Master-Slave D flip-flop

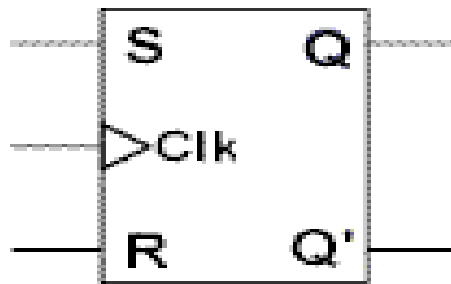
FLIPFLOPS:EXCITATION FUNCTIONS

- In [electronics design](#), an **excitation table** shows the minimum inputs that are necessary to generate a particular next state (in other words, to "**excite**" it to the next state) when the current state is known.
- They are similar to [truth tables](#) and [state tables](#), but rearrange the data so that the current state and next state are next to each other on the left-hand side of the table, and the inputs needed to make that state change happen.
- All flip-flops can be divided into four basic types: **SR**, **JK**, **D** and **T**. They differ in the number of inputs and in the response invoked by different value of input signals.

- The *characteristic table* in the third column of Table 1 defines the state of each flip-flop as a function of its inputs and previous state. Q refers to the present state and $Q(\text{next})$ refers to the next state after the occurrence of the clock pulse.
- The characteristic table for the RS flip-flop shows that the next state is equal to the present state when both inputs S and R are equal to 0. When R=1, the next clock pulse clears the flip-flop. When S=1, the flip-flop output Q is set to 1. The equation mark (?) for the next state when S and R are both equal to 1 designates an indeterminate next state.
- The characteristic table for the JK flip-flop is the same as that of the RS when J and K are replaced by S and R respectively, except for the indeterminate case. When both J and K are equal to 1, the next state is equal to the complement of the present state, that is, $Q(\text{next}) = Q'$.
- The next state of the D flip-flop is completely dependent on the input D and independent of the present state.
- The next state for the T flip-flop is the same as the present state Q if T=0 and complemented if T=1.

FLIPFLOPS:EXCITATION FUNCTIONS

SR Flip flop



FLIP-FLOPSYMBOL

S	R	Q _(next)
0	0	Q
0	1	0
1	0	1
1	1	?

CHARACTERISTIC TABLE

$$Q_{(next)} = S + R'Q$$

$$SR = 0$$

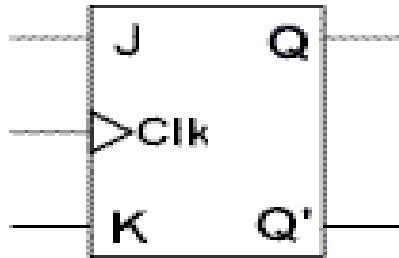
CHARACTERISTIC EQUATION

Q	Q _(next)	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

EXCITATION TABLE

FLIPFLOPS:EXCITATION FUNCTIONS

JK Flip flop



FLIP-FLOPSYMBOL

J	K	$Q_{(next)}$
0	0	Q
0	1	0
1	0	1
1	1	Q'

CHARACTERISTIC TABLE

$$Q_{(next)} = JQ' + K'Q$$

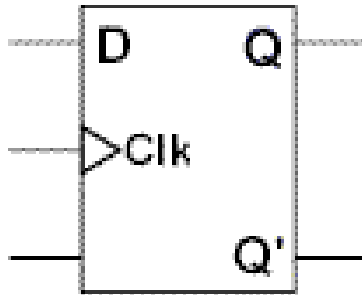
CHARACTERISTIC EQUATION

Q	$Q_{(next)}$	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

EXCITATION TABLE

FLIPFLOPS:EXCITATION FUNCTIONS

D Flip flop



FLIP-FLOPSYMBOL

D	$Q_{(next)}$
0	0
1	1

CHARACTERISTIC TABLE

$$Q_{(next)} = D$$

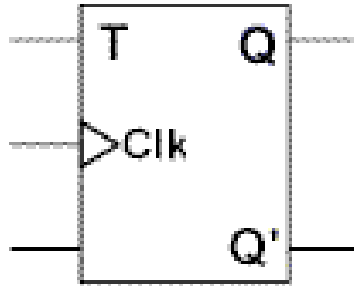
CHARACTERISTIC EQUATION

Q	$Q_{(next)}$	D
0	0	0
0	1	1
1	0	0
1	1	1

EXCITATION TABLE

FLIPFLOPS:EXCITATION FUNCTIONS

T Flip flop



FLIP-FLOPSYMBOL

T	Q _(next)
0	Q
1	Q'

CHARACTERISTIC TABLE

$$Q_{(next)} = TQ' + T'Q$$

CHARACTERISTIC EQUATION

Q	Q _(next)	T
0	0	0
0	1	1
1	0	1
1	1	0

EXCITATION TABLE

CONVERSION OF ONE FF TO ANOTHER FF

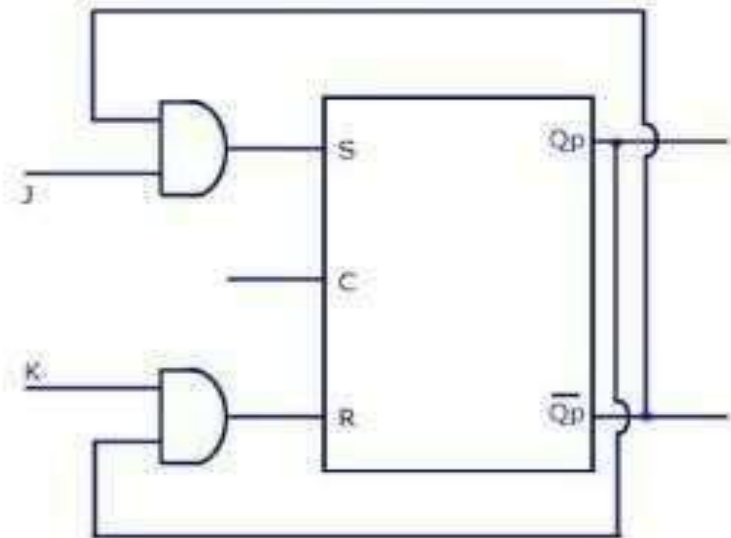
CONVERSION OF SR FLIP FLOP - JK FLIPFLOP

- J and K will be given as external inputs to S and R. As shown in the logic diagram below, S and R will be the outputs of the combinational circuit.
- The truth tables for the flip flop conversion are given below.
- The present state is represented by Q_p and Q_{p+1} is the next state to be obtained when the J and K inputs are applied.
- For two inputs J and K, there will be eight possible combinations. For each combination of J, K and Q_p , the corresponding Q_{p+1} states are found. Q_{p+1} simply suggests the future values to be obtained by the JK flip flop after the value of Q_p .
- The table is then completed by writing the values of S and R required to get each Q_{p+1} from the corresponding Q_p .
- That is, the values of S and R that are required to change the state of the flip flop from Q_p to Q_{p+1} are written.

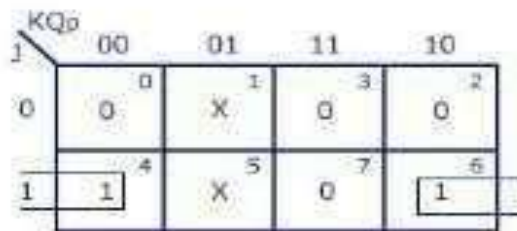
Conversion Table

J-K Inputs		Outputs		S-R Inputs	
J	K	Q _p	Q _{p+1}	S	R
0	0	0	0	0	X
0	0	1	1	X	0
0	1	0	0	0	X
0	1	1	0	0	1
1	0	0	1	1	0
1	0	1	1	X	0
1	1	0	1	1	0
1	1	1	0	0	1

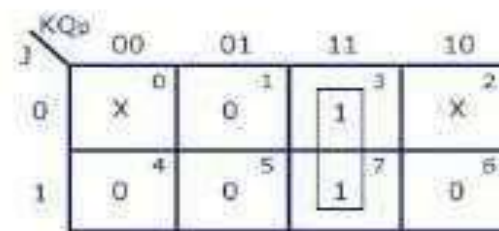
Logic Diagram



SR Flip Flop to JK Flip Flop



$$S = \bar{J}Q_p$$



$$R = KQ_p$$

K-Map

SR Flip Flop to JK Flip Flop

CONVERSION OF JK FLIP FLOP TO SR FLIPFLOP

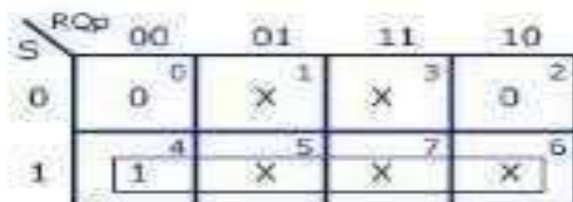
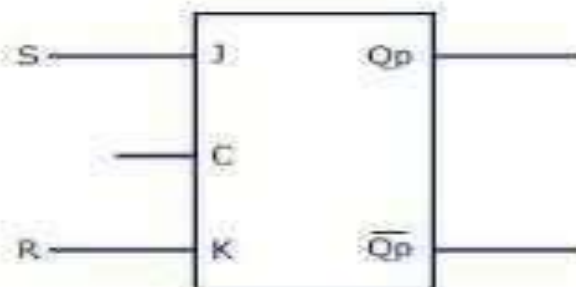
- This will be the reverse process of the above explained conversion. S and R will be the external inputs to J and K. As shown in the logic diagram below, J and K will be the outputs of the combinational circuit.
- Thus, the values of J and K have to be obtained in terms of S, R and Q_p . The logic diagram is shown below.
- A conversion table is to be written using S, R, Q_p , Q_{p+1} , J and K. For two inputs, S and R, eight combinations are made.
- For each combination, the corresponding Q_{p+1} outputs are found out. The outputs for the combinations of S=1 and R=1 are not permitted for an SR flip flop.
- Thus the outputs are considered invalid and the J and K values are taken as —don't cares—.

J-K Flip Flop to S-R Flip Flop

Conversion Table

S-R Inputs		Outputs		J-K Inputs	
S	R	Q _p	Q _{p+1}	J	K
0	0	0	0	0	X
0	0	1	1	X	0
0	1	0	0	0	X
0	1	1	0	X	1
1	0	0	1	1	X
1	0	1	1	X	0
1	1	Invalid		Dont care	
1	1	Invalid		Dont care	

Logic Diagram



$$J = S$$



$$K = R$$

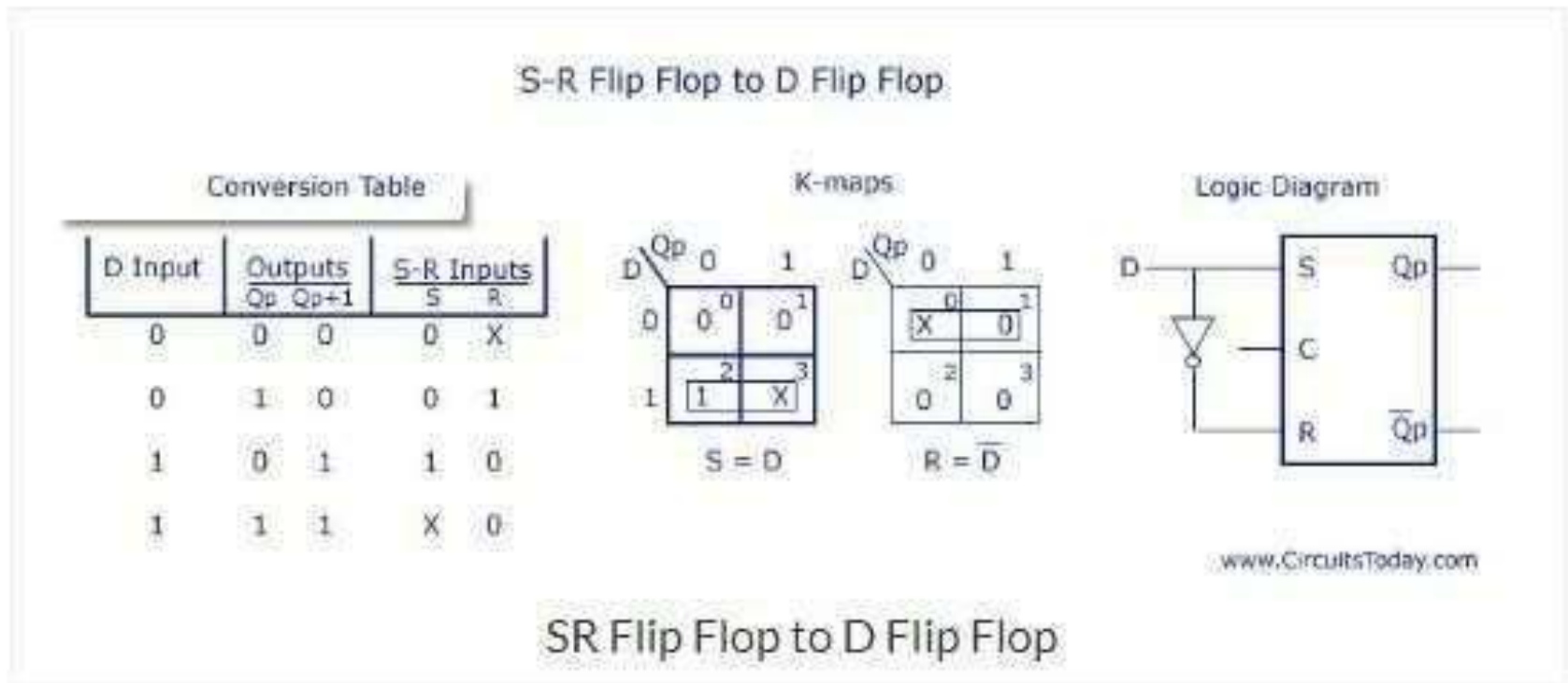
K-maps

www.CircuitsToday.com

JK Flip Flop to SR Flip Flop

CONVERSION OF SR FLIP FLOP TO D FLIPFLOP

As shown in the figure, S and R are the actual inputs of the flip flop and D is the external input of the flip flop. The four combinations, the logic diagram, conversion table, and the K-map for S and R in terms of D and Q_p are shown below.



CONVERSION OF D FLIP FLOP TO SR FLIPFLOP

D is the actual input of the flip flop and S and R are the external inputs.

Eight possible combinations are achieved from the external inputs S, R and Q_p .

But, since the combination of $S=1$ and $R=1$ are invalid, the values of Q_{p+1} and D are considered as —don't cares—. The logic diagram showing the conversion from

D to SR, and the K-map for D in terms of S, R and Q_p are shown below.

D Flip Flop to S-R Flip Flop

Conversion Table:

S-R Inputs		Outputs		D Input
S	R	Q_p	Q_{p+1}	
0	0	0	0	0
0	0	1	1	1
0	1	0	0	0
0	1	1	0	0
1	0	0	1	1
1	0	1	1	1
1	1	Invalid		Don't care
1	1	Invalid		Don't care

K-map

	RQ_p	00	01	11	10
0	S	0	1	0	0
1	S	1	1	X	X

$D = S + RQ_p$

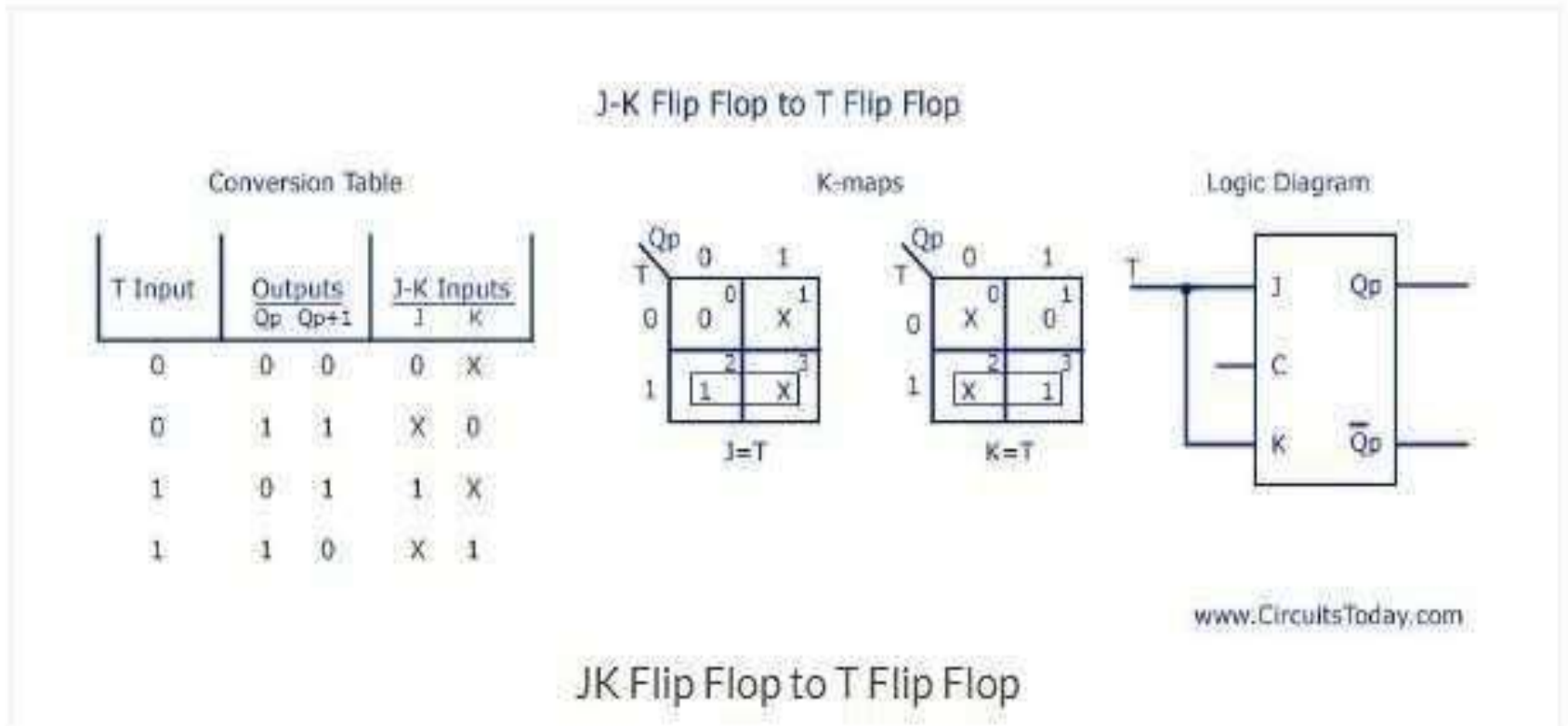
Logic Diagram

www.CircuitsToday.com

D Flip Flop to SR Flip Flop

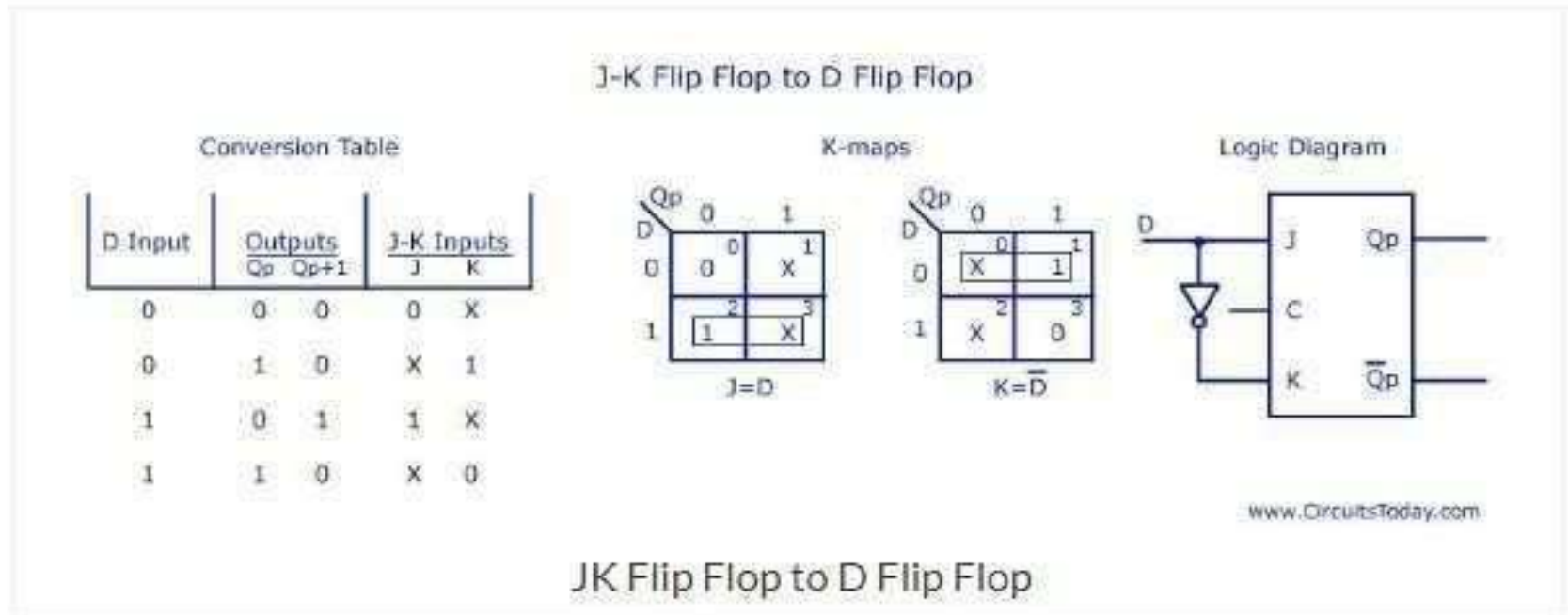
CONVERSION OF JK FLIP FLOP TO T FLIP FLOP

J and K are the actual inputs of the flip flop and T is taken as the external input for conversion. Four combinations are produced with T and Q_p . J and K are expressed in terms of T and Q_p . The conversion table, K-maps, and the logic diagram are given below.



CONVERSION OF JK FLIP FLOP TO D FLIPFLOP

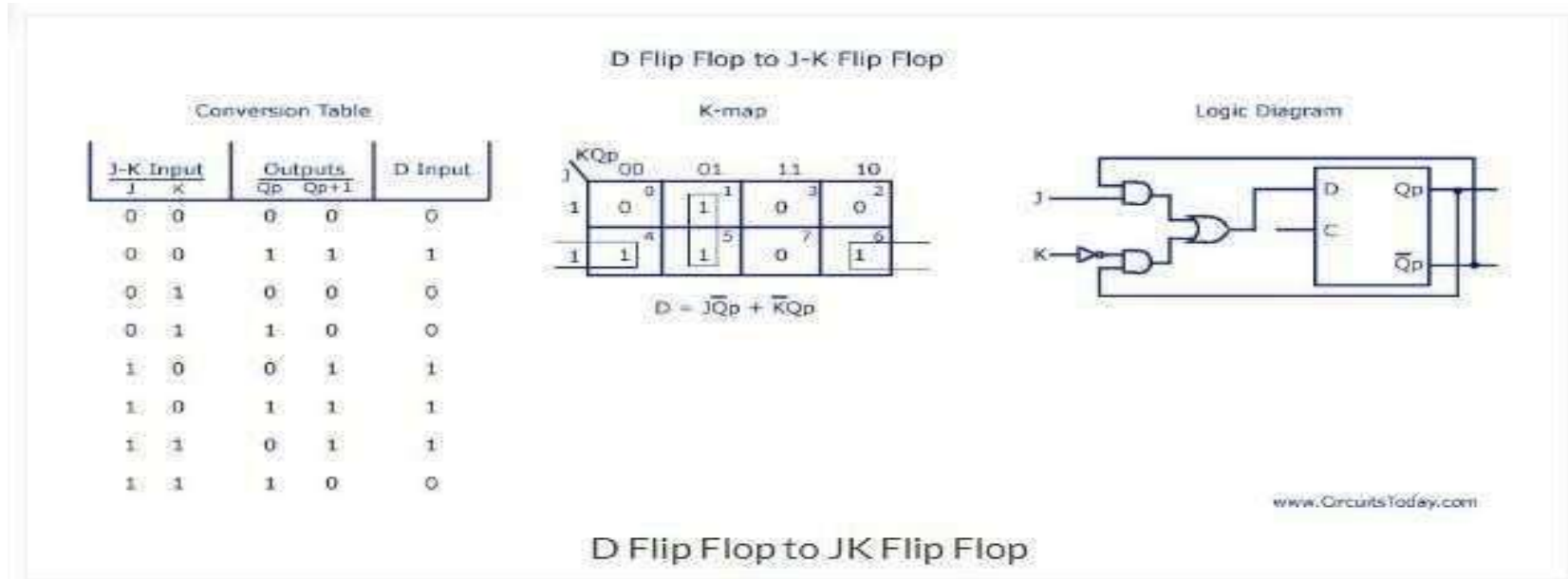
D is the external input and J and K are the actual inputs of the flip flop. D and Q_p make four combinations. J and K are expressed in terms of D and Q_p . The four combination conversion table, the K-maps for J and K in terms of D and Q_p , and the logic diagram showing the conversion from JK to D are given below.



CONVERSION OF D FLIP FLOP TO JK FLIPFLOP

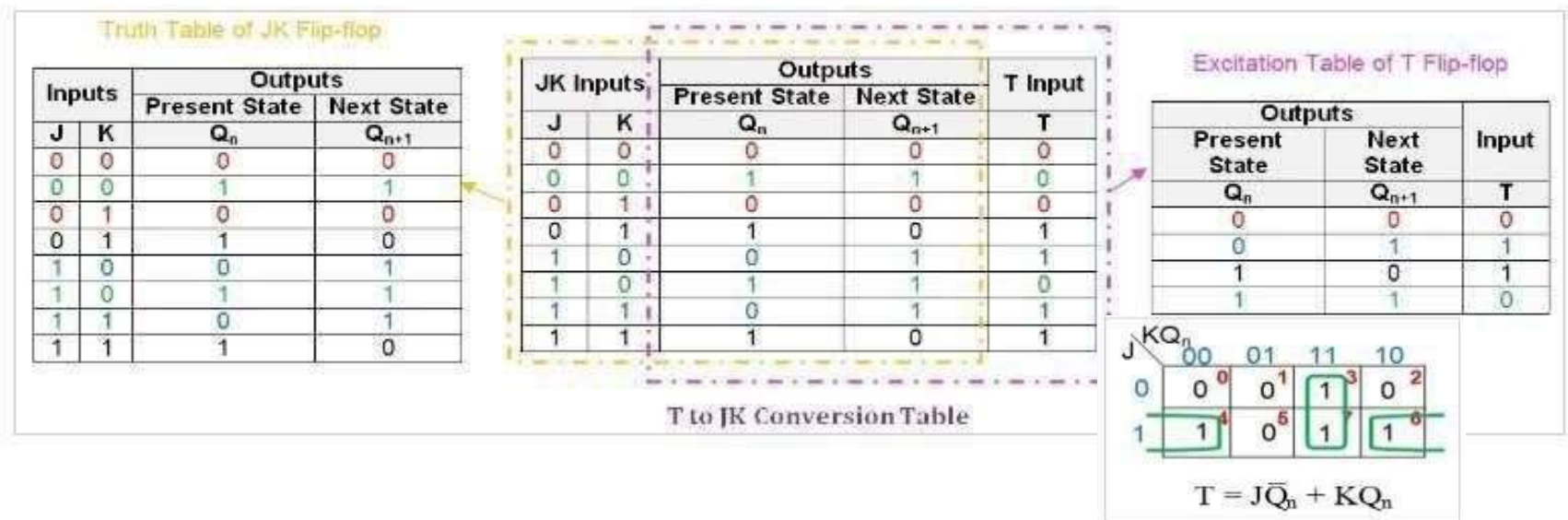
In this conversion, D is the actual input to the flip flop and J and K are the external inputs. J, K and Qp make eight possible combinations, as shown in the conversion table below. D is expressed in terms of J, K and Qp.

The conversion table, the K-map for D in terms of J, K and Qp and the logic diagram showing the conversion from D to JK are given in the figure below.



CONVERSION OF T FLIP FLOP TO JK FLIP FLOP

We begin with the T-to-JK conversion table (see Figure 5), which combines the information in the JK flip-flop's truth table and the T flip-flop's excitation table.



Next, we need to obtain the simplified Boolean expression for the T input in terms of J, K, and Q_n . The expression for the T input is $J\bar{Q}_n + KQ_n$. This means that to convert the T flip-flop into a JK flip-flop, the T input is driven by the output of a two-input OR gate which has as inputs J ANDed with the negation of the present-state Q_n , i.e., $\bar{Q}_n K$ ANDed with the present-state, Q_n

SHIFT REGISTERS

Introduction :

- *Shift registers* are a type of sequential logic circuit, mainly for storage of digital data.
- They are a group of flip-flops connected in a chain so that the output from one flip-flop becomes the input of the next flip-flop.
- Most of the registers possess no characteristic internal sequence of states.
- All the flip-flops are driven by a common clock, and all are set or reset simultaneously.
- Shift registers are divided into two types.

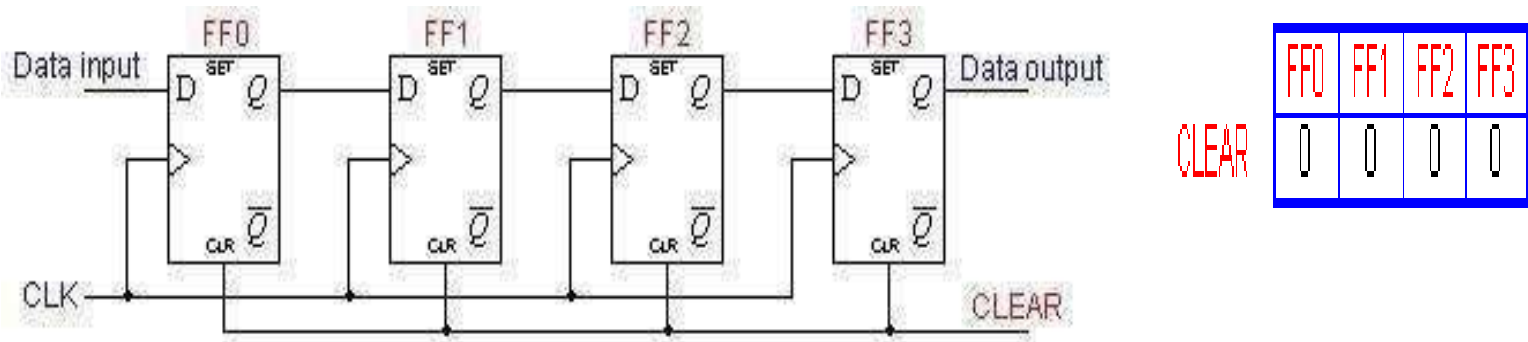
1.Uni directional shift registers

- 1.Serial in – serial out shift register
- 2.Serial in – parallel out shift register
- 3.Parallel in – serial out shift register
4. Parallel in – parallel out shift register

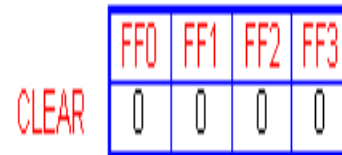
2.Bidirectional shift registers

1. Serial in – serial out shift register

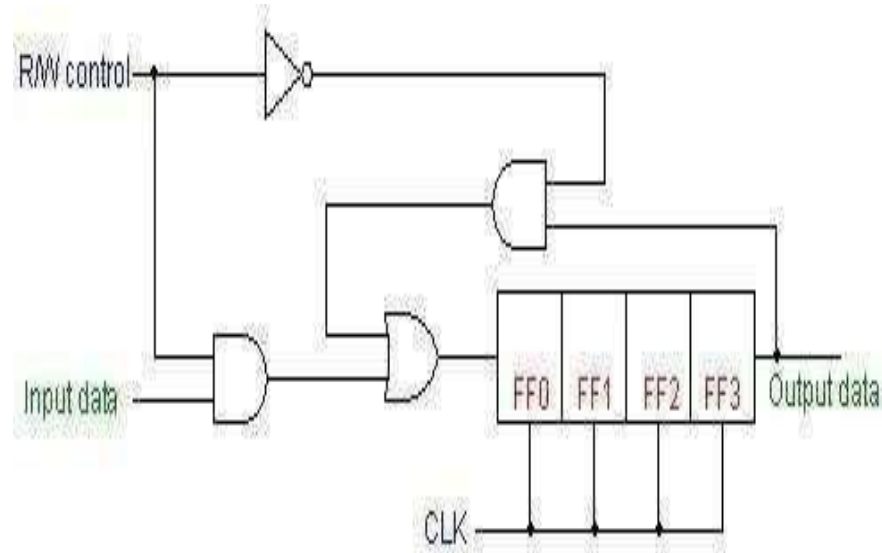
A basic four-bit shift register can be constructed using four D flip-flops, as shown below. The operation of the circuit is as follows. The register is first cleared, forcing all four outputs to zero. The input data is then applied sequentially to the D input of the first flip-flop on the left (FF0). During each clock pulse, one bit is transmitted from left to right. Assume a data word to be 1001. The least significant bit of the data has to be shifted through the register from FF0 to FF3.



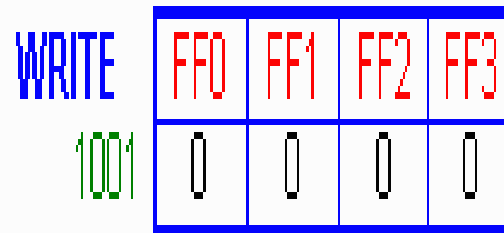
In order to get the data out of the register, they must be shifted out serially. This can be done destructively or non-destructively. For destructive readout, the original data is lost and at the end of the read cycle, all flip-flops are reset to zero.



To avoid the loss of data, an arrangement for a non-destructive reading can be done by adding two AND gates, an OR gate and an inverter to the system. The construction of this circuit is shown below.

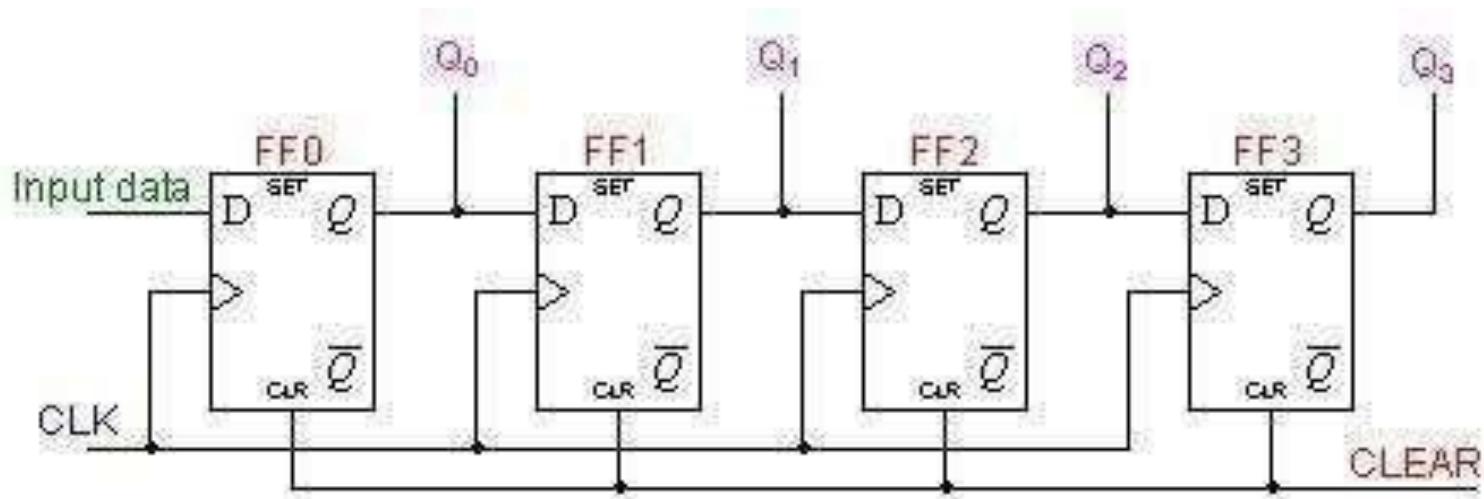


The data is loaded to the register when the control line is HIGH (ie WRITE). The data can be shifted out of the register when the control line is LOW (ie READ). This is shown in the animation below.



2. Serial in – parallel out shift register

The difference is the way in which the data bits are taken out of the register. Once the data are stored, each bit appears on its respective output line, and all bits are available simultaneously.

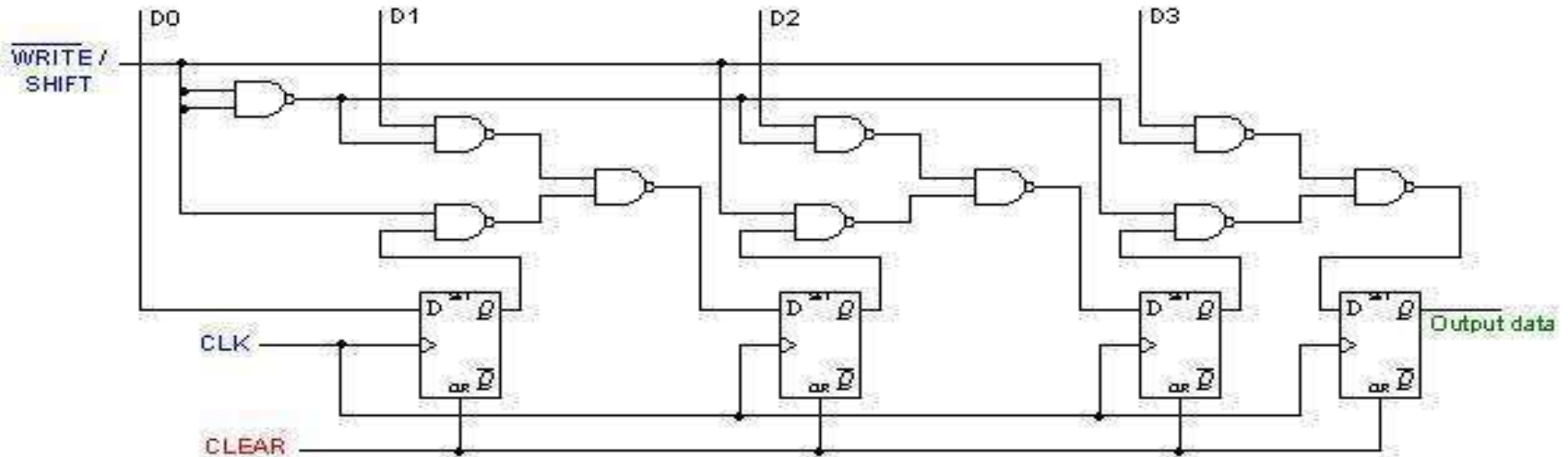


In the animation below, we can see how the four-bit binary number 1001 is shifted to the Q outputs of the register.

CLEAR	Q ₀	Q ₁	Q ₂	Q ₃
1001	0	0	0	0

3.Parallel in – serial out shift register

A four-bit parallel in - serial out shift register is shown below. The circuit uses D flip-flops and NAND gates for entering data (ie writing) to the register.

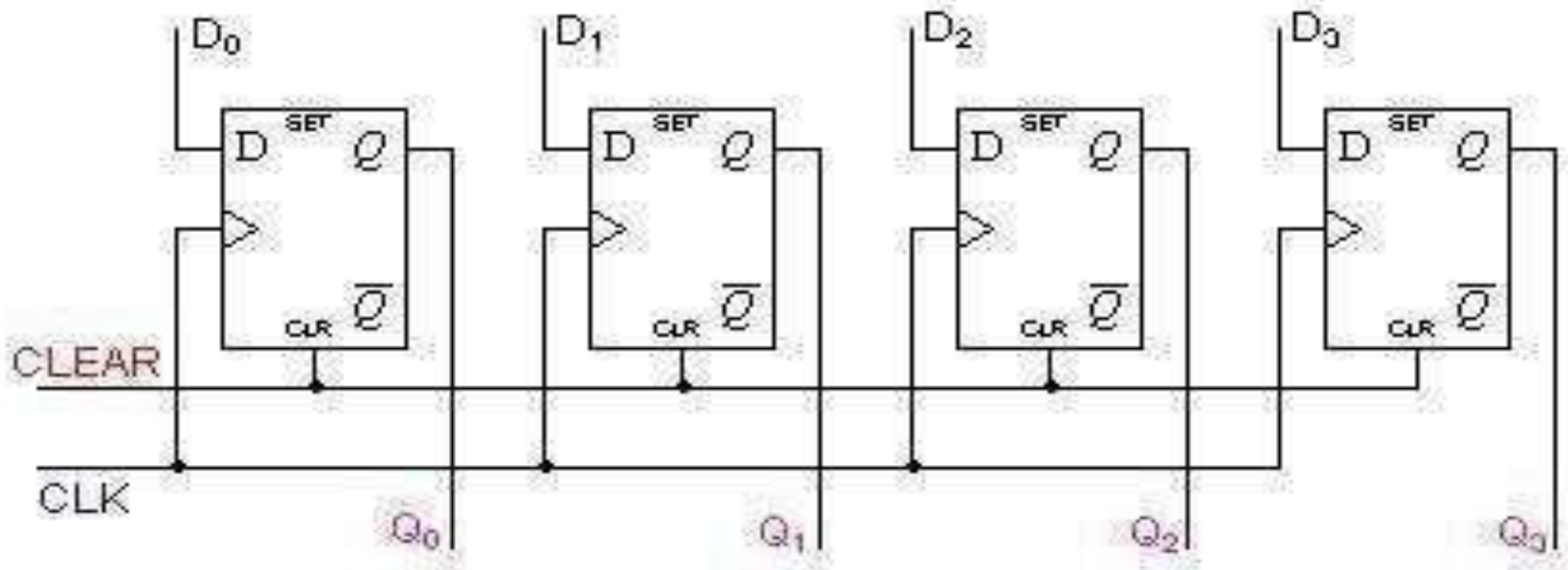


D0, D1, D2 and D3 are the parallel inputs, where D0 is the most significant bit and D3 is the least significant bit. To write data in, the mode control line is taken to LOW and the data is clocked in. The data can be shifted when the mode control line is HIGH as SHIFT is active high. The register performs right shift operation on the application of a clock pulse, as shown in the animation below.

	Q0	Q1	Q2	Q3
CLEAR	0	0	0	0

4.Parallel in –parallel out shift register

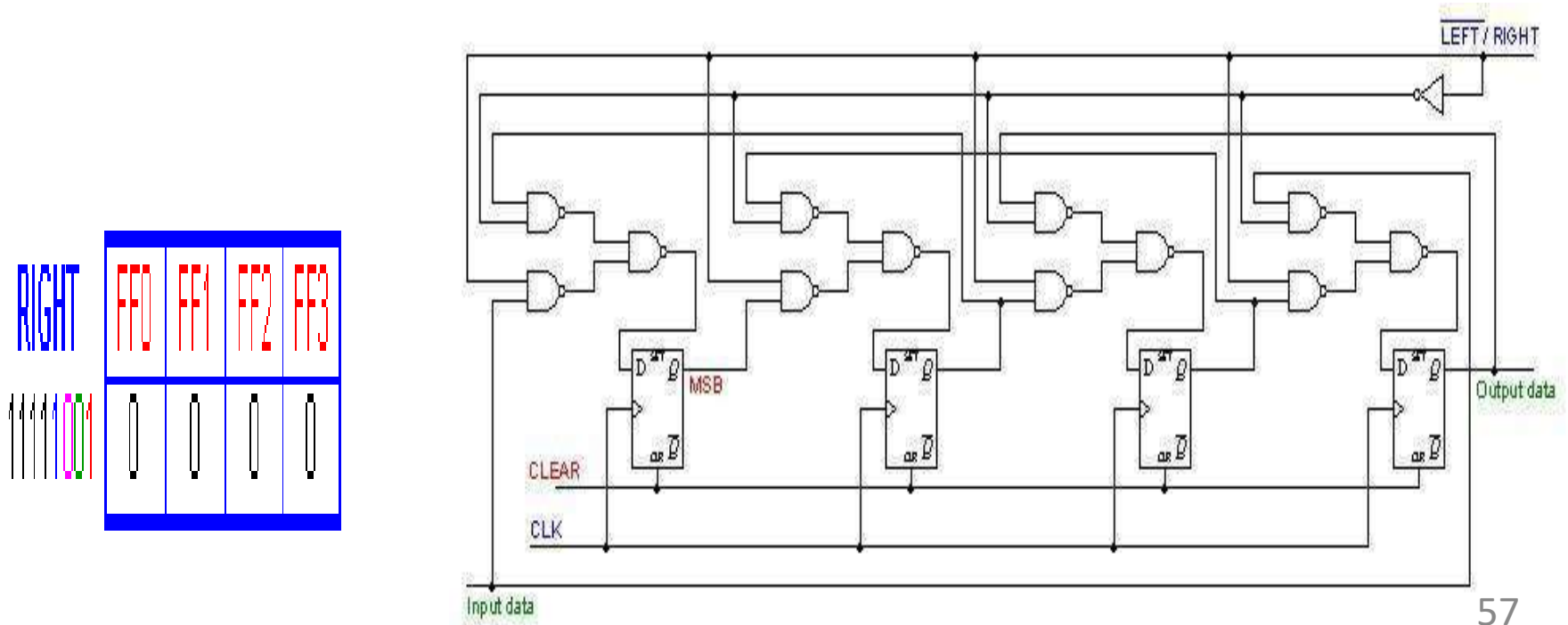
For parallel in - parallel out shift registers, all data bits appear on the parallel outputs immediately following the simultaneous entry of the data bits. The following circuit is a four-bit parallel in - parallel out shift register constructed by D flip-flops.



The D's are the parallel inputs and the Q's are the parallel outputs. Once the register is clocked, all the data at the D inputs appear at the corresponding Q outputs simultaneously.

Bidirectional Shift Registers :

The registers discussed so far involved only right shift operations. Each right shift operation has the effect of successively dividing the binary number by two. If the operation is reversed (left shift), this has the effect of multiplying the number by two. With suitable gating arrangement a serial shift register can perform both operations. A *bidirectional*, or *reversible*, shift register is one in which the data can be shift either left or right. A four-bit bidirectional shift register using D flip-flops is shown below.



Here a set of NAND gates are configured as OR gates to select data inputs from the right or left adjacent bitable, as selected by the LEFT/RIGHT control line.

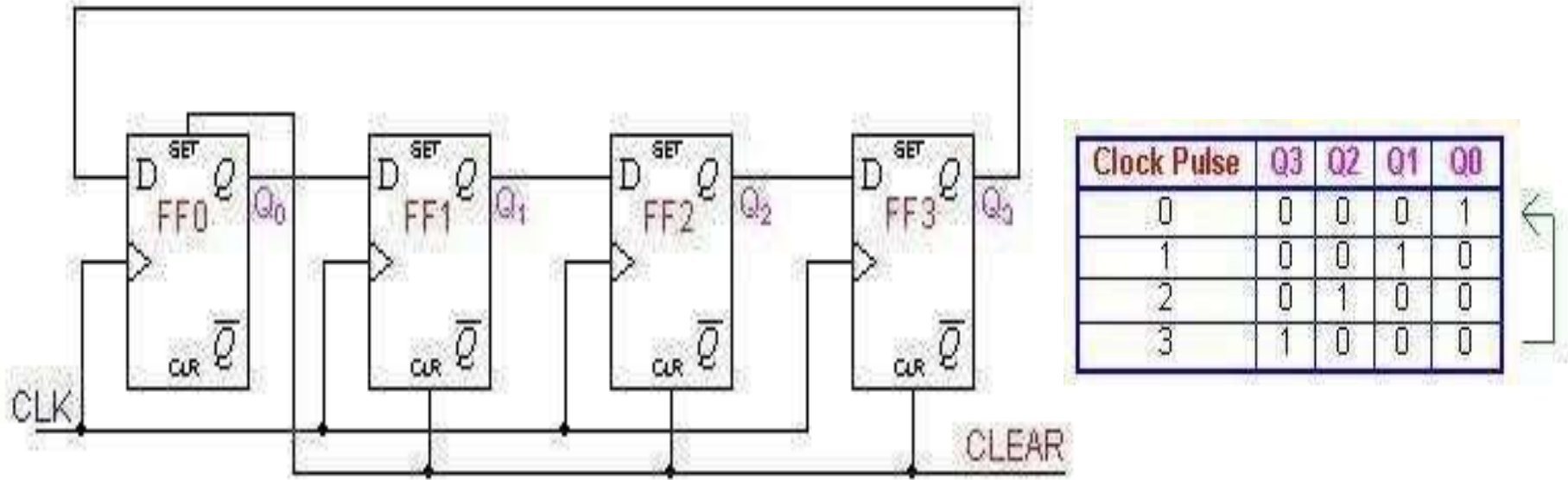
The animation below performs right shift four times, then left shift four times. Notice the order of the four output bits are not the same as the order of the original four input bits.

COUNTERS

- Two of the most common types of shift register counters are introduced here: the Ring counter and the Johnson counter.
- They are basically shift registers with the serial outputs connected back to the serial inputs in order to produce particular sequences.
- These registers are classified as counters because they exhibit a specified sequence of states.

Ring Counters

- A ring counter is basically a circulating shift register in which the output of the most significant stage is fed back to the input of the least significant stage.
- The following is a 4-bit ring counter constructed from D flip-flops.
- The output of each stage is shifted into the next stage on the positive edge of a clock pulse.
- If the CLEAR signal is high, all the flip-flops except the first one FF0 are reset to 0. FF0 is preset to 1 instead.

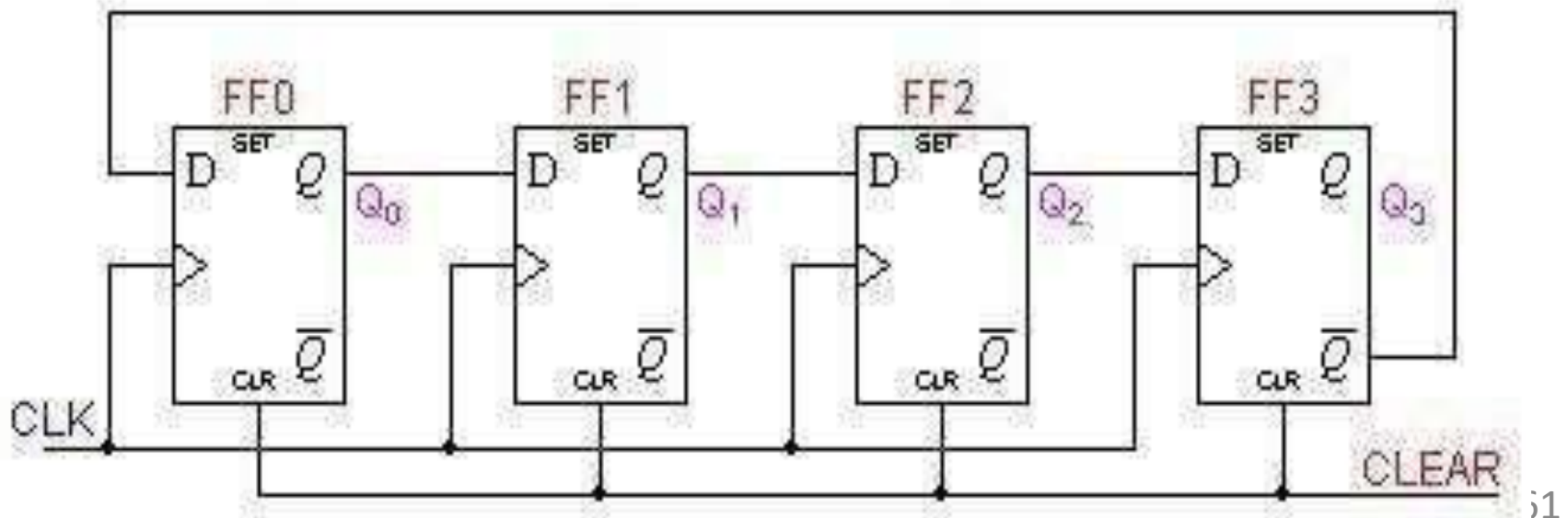


- Since the count sequence has 4 distinct states, the counter can be considered as a mod-4 counter. Only 4 of the maximum 16 states are used, making ring counters very inefficient in terms of state usage.
- But the major advantage of a ring counter over a binary counter is that it is self-decoding. No extra decoding circuit is needed to determine what state the counter is in.

CLEAR	FF0	FF1	FF2	FF3
	1	0	0	0

Johnson Counters

- Johnson counters are a variation of standard ring counters, with the inverted output of the last stage fed back to the input of the first stage.
- They are also known as **twisted ring counters**. An n - stage Johnson counter yields a count sequence of length $2n$, so it may be considered to be a **mod- $2n$ counter**.
- The circuit above shows a **4-bit Johnson counter**.
- The state sequence for the counter is given in the table as well as the animation on the left.



Clock Pulse	Q3	Q2	Q1	Q0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	1
3	0	1	1	1
4	1	1	1	1
5	1	1	1	0
6	1	1	0	0
7	1	0	0	0



CLEAR

FF0	FF1	FF2	FF3
0	0	0	0

- Again, the apparent disadvantage of this counter is that the maximum available states are not fully utilized. Only eight of the sixteen states are being used.
- Beware that for both the Ring and the Johnson counter must initially be forced into a valid state in the count sequence because they operate on a subset of the available number of states. Otherwise, the ideal sequence will not be followed.

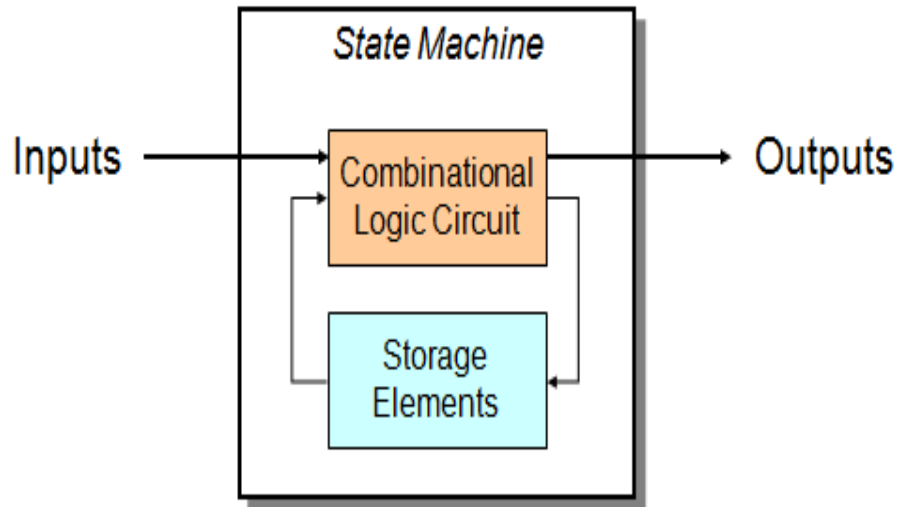
STATE MACHINES

State Machine

Another type of sequential circuit

Combines combinational logic with storage

—Remembers state, and changes output (and state) based on **inputs** and **current state**



State

- The state of a system is a snapshot of all the relevant elements of the system at the moment the snapshot is taken.

Examples:

- The state of a basketball game can be represented by the scoreboard.
Number of points, time remaining, possession, etc.
- The state of a tic-tac-toe game can be represented by the placement of X's and O's on the board.

STATE TABLES AND STATE DIAGRAMS

- In this model the effect of all previous inputs on the outputs is represented by a state of the circuit.
- Thus, the output of the circuit at any time depends upon its current state and the input.
These also determine the next state of the circuit.
- The relationship that exists among the inputs, outputs, present states and next states can be specified by either the **state table** or the **state diagram**.

State Table

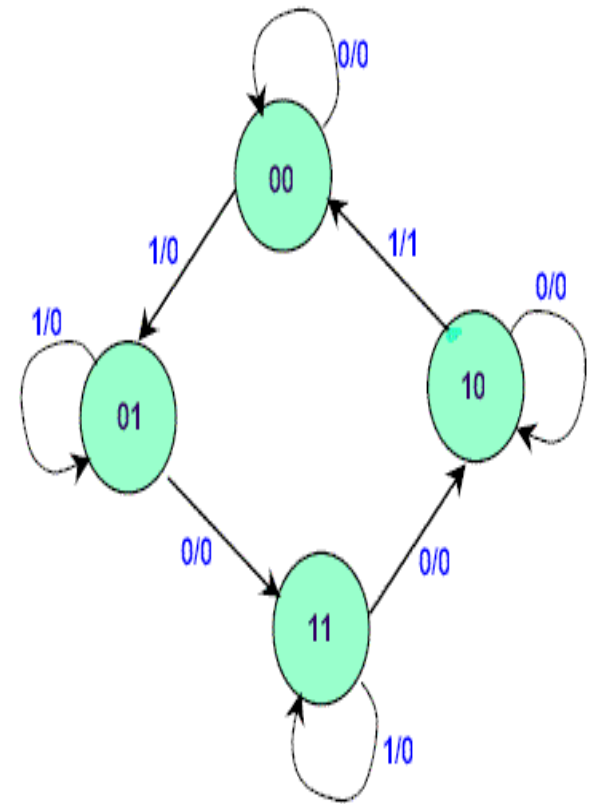
- The state table representation of a sequential circuit consists of three sections labeled *present state*, *next state* and *output*.
- The present state designates the state of flip-flops before the occurrence of a clock pulse.
- The next state shows the states of flip-flops after the clock pulse, and the output section lists the value of the output variables during the present state.

Present State	Next State		Present Output
	X=0	X=1	
a	d	c	0
b	d	c	0
c	d	a	0
d	d	c	1

State Diagram :

- In addition to graphical symbols, tables or equations, flip-flops can also be represented graphically by a state diagram.
- In this diagram, a state is represented by a circle, and the transition between states is indicated by directed lines (or arcs) connecting the circles.
- The binary number inside each circle identifies the state the circle represents.
- The directed lines are labelled with two binary numbers separated by a slash (/).
- The input value that causes the state transition is labelled first.
- The number after the slash symbol / gives the value of the output.

- For example, the directed line from state 00 to 01 is labelled 1/0, meaning that, if the sequential circuit is in a present state and the input is 1, then the next state is 01 and the output is 0.
- If it is in a present state 00 and the input is 0, it will remain in that state.
- A directed line connecting a circle with itself indicates that no change of state occurs.
- The state diagram provides exactly the same information as the state table and is obtained directly from the state table.



Example:

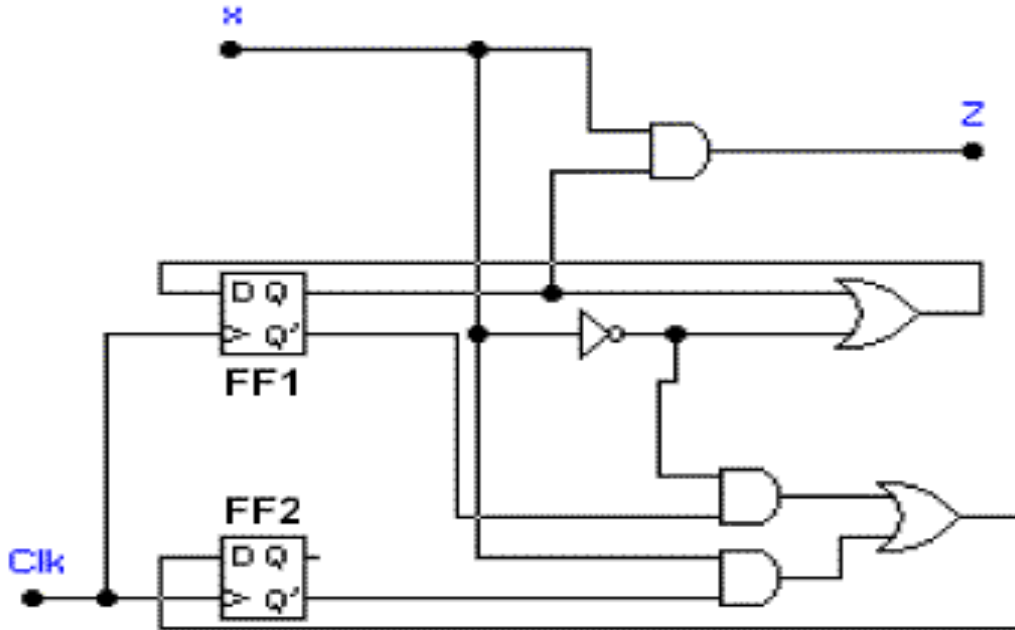
Consider a sequential circuit

The behavior of the circuit is determined by the following Boolean expressions:

$$Z = x * Q1$$

$$D1 = x' + Q1$$

$$D2 = x * Q2' + x' * Q1'$$



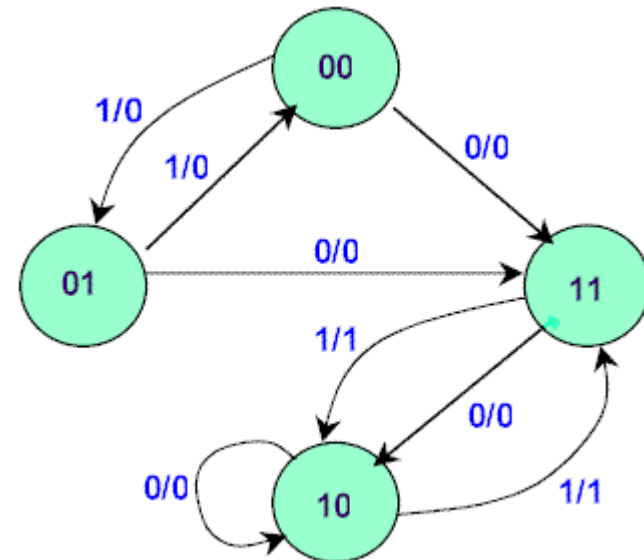
These equations can be used to form the state table.

- Suppose the present state (i.e. $Q1Q2$) = 00 and input $x = 0$. Under these conditions, we get $Z = 0$, $D1 = 1$, and $D2 = 1$.
- Thus the next state of the circuit $D1D2 = 11$, and this will be the present state after the clock pulse has been applied.
- The output of the circuit corresponding to the present state $Q1Q2 = 00$ and $x = 1$ is $Z = 0$.
- This data is entered into the state table as shown in Table 2.

State table for the sequential circuit

Present State $Q1Q2$	Next State		Output	
	$x=0$	$x=1$	$x=0$	$x=1$
00	11	01	0	0
01	11	00	0	0
10	10	11	0	1
11	10	10	0	1

The state diagram for the sequential circuit



State diagrams of the four types of flip-flops

NAME	STATE DIAGRAM
SR	<p>State diagram for SR flip-flop showing two states: $Q = 0$ and $Q = 1$. Transitions are labeled with S, R values:</p> <ul style="list-style-type: none"> Self-loops on $Q = 0$ and $Q = 1$ are labeled $S, R = 0, 0$. Transition from $Q = 0$ to $Q = 1$ is labeled $S, R = 1, 0$. Transition from $Q = 1$ to $Q = 0$ is labeled $S, R = 0, 1$.
JK	<p>State diagram for JK flip-flop showing two states: $Q = 0$ and $Q = 1$. Transitions are labeled with J, K values:</p> <ul style="list-style-type: none"> Self-loops on $Q = 0$ and $Q = 1$ are labeled $J, K = 0, 0$. Transition from $Q = 0$ to $Q = 1$ is labeled $J, K = 1, 0$ or $1, 1$. Transition from $Q = 1$ to $Q = 0$ is labeled $J, K = 0, 1$ or $1, 1$.
D	<p>State diagram for D flip-flop showing two states: $Q = 0$ and $Q = 1$. Transitions are labeled with D values:</p> <ul style="list-style-type: none"> Self-loops on $Q = 0$ and $Q = 1$ are labeled $D = 1$. Transition from $Q = 1$ to $Q = 0$ is labeled $D = 0$.
T	<p>State diagram for T flip-flop showing two states: $Q = 0$ and $Q = 1$. Transitions are labeled with T values:</p> <ul style="list-style-type: none"> Self-loops on $Q = 0$ and $Q = 1$ are labeled $T = 0$. Transitions between $Q = 0$ and $Q = 1$ are labeled $T = 1$.

STATE REDUCTION

State Reduction

Any design process must consider the problem of minimising the cost of the final circuit. The

two most obvious cost reductions are reductions in the number of flip-flops and the number of gates.

The number of states in a sequential circuit is closely related to the complexity of the resulting circuit. It is therefore desirable to know when two or more states are equivalent in all aspects. The process of eliminating the equivalent or redundant states from a state table/diagram is known as **state reduction**.

Example: Let us consider the state table of a sequential circuit

Present State	Next State		Output	
	x = 0	x = 1	x = 0	x = 1
A	B	C	1	0
B	F	D	0	0
C	D	E	1	1
D	F	E	0	1
E	A	D	0	0
F	B	C	1	0

It can be seen from the table that the present state A and F both have the same next states, B (when $x=0$) and C (when $x=1$). They also produce the same output 1 (when $x=0$) and 0 (when $x=1$). Therefore states A and F are equivalent. Thus one of the states, A or F can be removed from the state table. For example, if we remove row F from the table and replace all F's by A's in the columns, the state table is modified

Present State	Next State		Output	
	$x = 0$	$x = 1$	$x = 0$	$x = 1$
A	B	C	1	0
B	A	D	0	0
C	D	E	1	1
D	A	E	0	1
E	A	D	0	0

State F removed

It is apparent that states B and E are equivalent. Removing E and replacing E's by B's results in the reduce table

Present State	Next State		Output	
	x = 0	x = 1	x = 0	x = 1
A	B	C	1	0
B	A	D	0	0
C	D	B	1	1
D	A	B	0	1

Reduced state table

The removal of equivalent states has reduced the number of states in the circuit from six to four. Two states are considered to be **equivalent** if and only if for every input sequence the circuit produces the same output sequence irrespective of which one of the two states is the starting state.

STATE ASSIGNMENT

STATE ASSIGNMENT

Each circuit state given in a state table has to be assigned a unique value, which represents combinations of flip – flop output states.

- A circuit having 2 internal states requires one flip – flop in its implementation
- A circuit having 3 or 4 internal states requires two flip – flops in its implementation
- A circuit having 5 → 8 internal states requires three flip – flops in its implementation etc.

It should be noted that although assignments are arbitrary, one assignment might be more economical than another.

Consider the state table shown below for a circuit having two input pulses \mathbf{x}_1 , \mathbf{x}_2 and a level output \mathbf{Z} .

Since the circuit has four internal states then two flip-flops are required. Let the two flip-flop outputs be represented by variables y_1 and y_2 , which can have combinations of values $y_1y_2 = 00, 01, 11, 10$. The state table can then be translated into a state table with secondary assignments as shown. Note that this is just one of many possible assignments (in fact there are 24)

PS	NS		Z
	x_1	x_2	
①	2	1	0
②	3	1	0
③	4	1	1
④	4	1	0

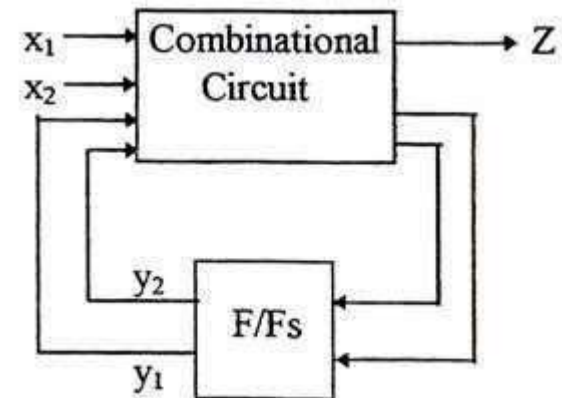


PS y_1y_2	Next State ($y_1'y_2'$)	
	x_1	x_2
00	01	00
01	11	00
11	10	00
10	10	00

Present F/F o/ps	Next state F/F o/ps
------------------	---------------------

Example of state assignment

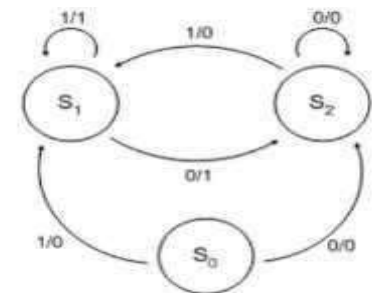
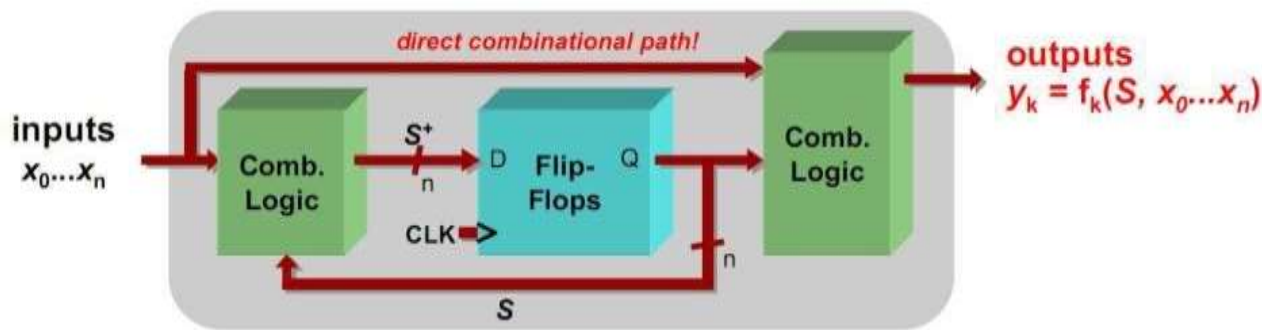
With $y_1y_2 = 0$ (i.e. in state 1), if x_1 is applied then y_1y_2 must change to 01 (i.e. state 2). That is, the flip/flop generating y_1 must not be disturbed, but the y_2 generating flip-flop requires an input such that the circuit settles in state 2, (for example a SET input if using SR flip-flops).



MEALY AND MOORE STATE MACHINES

Mealy state machine

In the theory of computation, a Mealy machine is a finite state transducer that generates an output based on its current state and input. This means that the state diagram will include both an input and output signal for each transition edge. In contrast, the output of a Moore finite state machine depends only on the machine's current state; transitions are not directly dependent upon input. The use of a Mealy FSM leads often to a reduction of the number of states. However, for each Mealy machine there is an equivalent Moore machine.

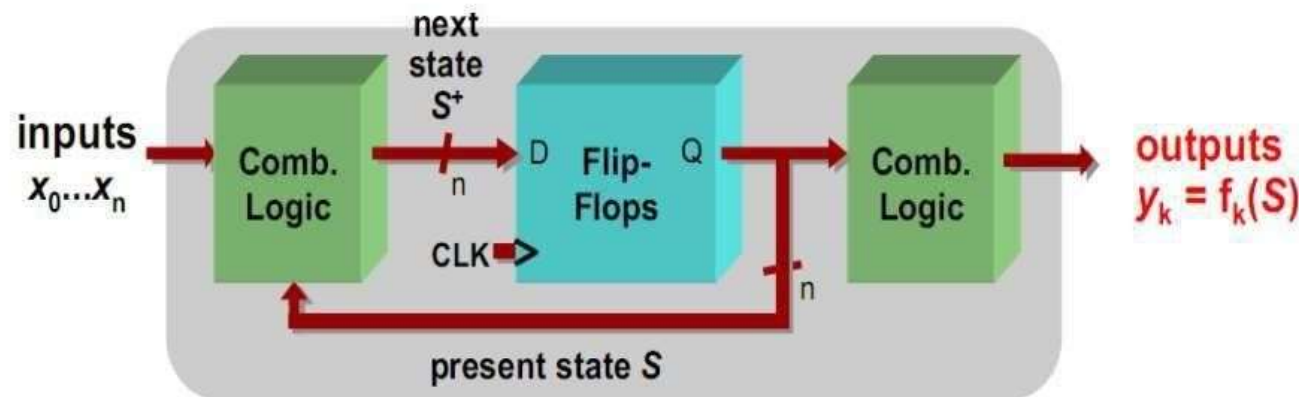


Next state = $F(\text{current state, input})$

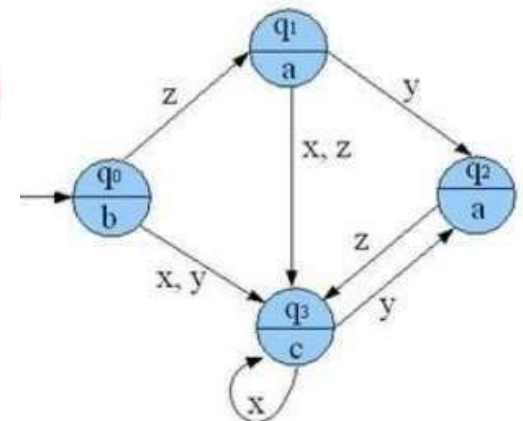
Output = $G(\text{current state, input})$

Moore state machine

In the theory of computation, a Moore machine is a finite state transducer where the outputs are determined by the current state alone (and do not depend directly on the input). The state diagram for a Moore machine will include an output signal for each state. Compare with a Mealy machine, which maps transitions in the machine to outputs. The advantage of the Moore model is a simplification of the behavior.



$$\text{Output} = G(\text{current state})$$



Examples for Mealy and Moore machines

Derive a minimal state table for a single-input and single-output Moore-type FSM that produces an output of 1 if in the input sequence it detects either 110 or 101 patterns. Overlapping sequences should be detected. (Show the detailed steps of your solution.)

Task description: ↵

clock	:	t ₀	t ₁	t ₂	t ₃	t ₄	t ₅	t ₆	t ₇	t ₈	t ₉	t ₁₀ ⁺
input w:		0	1	1	0	0	1	0	1	1	0	0
output z:		0	0	0	1	0	0	0	1	0	1	0

↵

State Assignment (Moore FSM) : ↵

state A: Got no 1 ↵

state B: Got "1" ↵

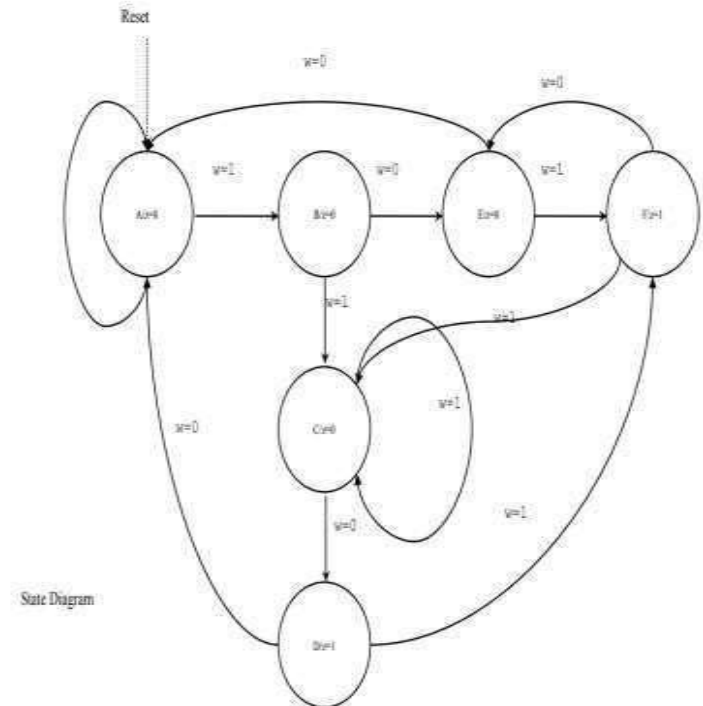
state C: Got "11" ↵

state D: Got "110" ↵

state E: Got "10" ↵

State F: Got "101" ↵

State Diagram



State Table (Moore FSM)

Present state	Next State		Output Z
	w=0	w=1	
A	A	B	0
B	E	C	0
C	D	C	0
D	A	F	1
E	A	F	0
F	E	C	1

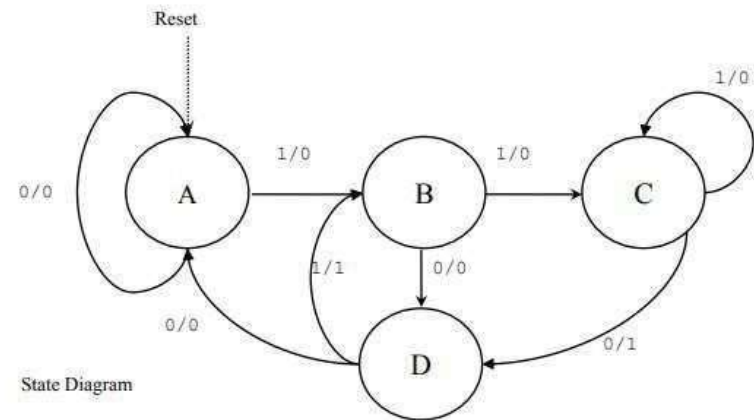
6 states need 3 flip-flops

State Assignment (Mealy FSM):

state A: Got no 1 state B: Got 11 state C:

Got 111 state D: Got 10”

State Table (Mealy FSM)



Present state	Next State		Output Z	
	w=0	w=1	w=0	w=1
A	A	B	0	0
B	D	C	0	0
C	D	C	1	0
D	A	B	0	1

4 states need 2 flip-flops

Sequential Logic Implementation

➤ Models for representing sequential circuits

Abstraction of sequential elements

Finite state machines and their state diagrams Inputs/outputs

Mealy, Moore, and synchronous Mealy machines

➤ Finite state machine design procedure

Verilog specification

Deriving state diagram

Deriving state transition table

Determining next state and output functions

Implementing combinational logic

Mealy vs. Moore Machines

Moore: outputs depend on current state only

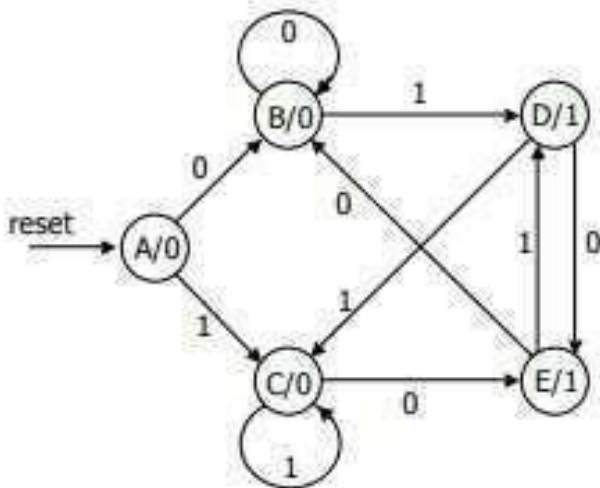
Mealy: outputs depend on current state and inputs

➤ Ant brain is a Moore Machine (Output does not react immediately to input change)

➤ We could have specified a Mealy FSM (Outputs have immediate reaction to inputs . As inputs change, so does next state, doesn't commit until clocking event)

Specifying Outputs for a Moore Machine

Output is only function of state. Specify in state bubble in state diagram. Example: sequence detector for 01 or 10

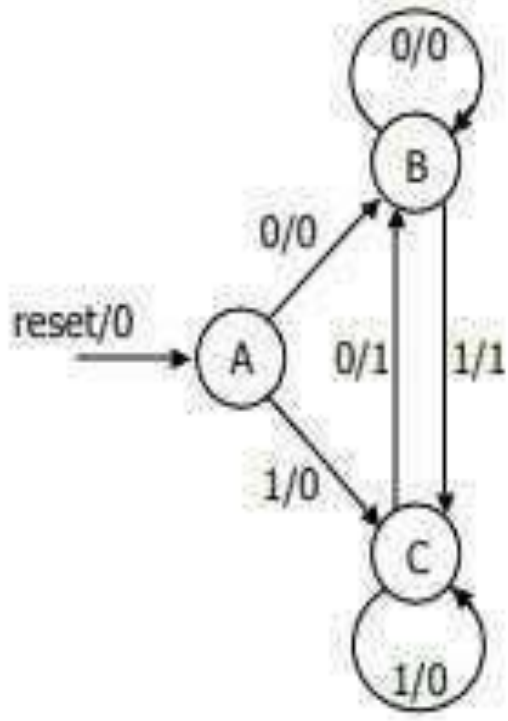


reset	input	current state	next state	output
1	—	—	A	
0	0	A	B	0
0	1	A	C	0
0	0	B	B	0
0	1	B	D	0
0	0	C	E	0
0	1	C	C	0
0	0	D	E	1
0	1	D	C	1
0	0	E	B	1
0	1	E	D	1

Specifying Outputs for a Mealy Machine :

Output is function of state and inputs .Specify output on transition arc between states.

Example: sequence detector for 01 or 10



reset	input	current state	next state	output
1	-	-	A	0
0	0	A	B	0
0	1	A	C	0
0	0	B	B	0
0	1	B	C	1
0	0	C	B	1
0	1	C	C	0

Comparison of Mealy and Moore Machines :

- Mealy Machines tend to have less states
- Different outputs on arcs (n^2) rather than states (n)
- Moore Machines are safer to use
- Outputs change at clock edge (always one cycle later)
- In Mealy machines, input change can cause output change as soon as logic is done – a big problem when two machines are interconnected – asynchronous feedback
- Mealy Machines react faster to inputs
- React in same cycle – don't need to wait for clock
- In Moore machines, more logic may be necessary to decode state into outputs – more gate delays after

