

MODULE – 1

FETs & OP-AMPS

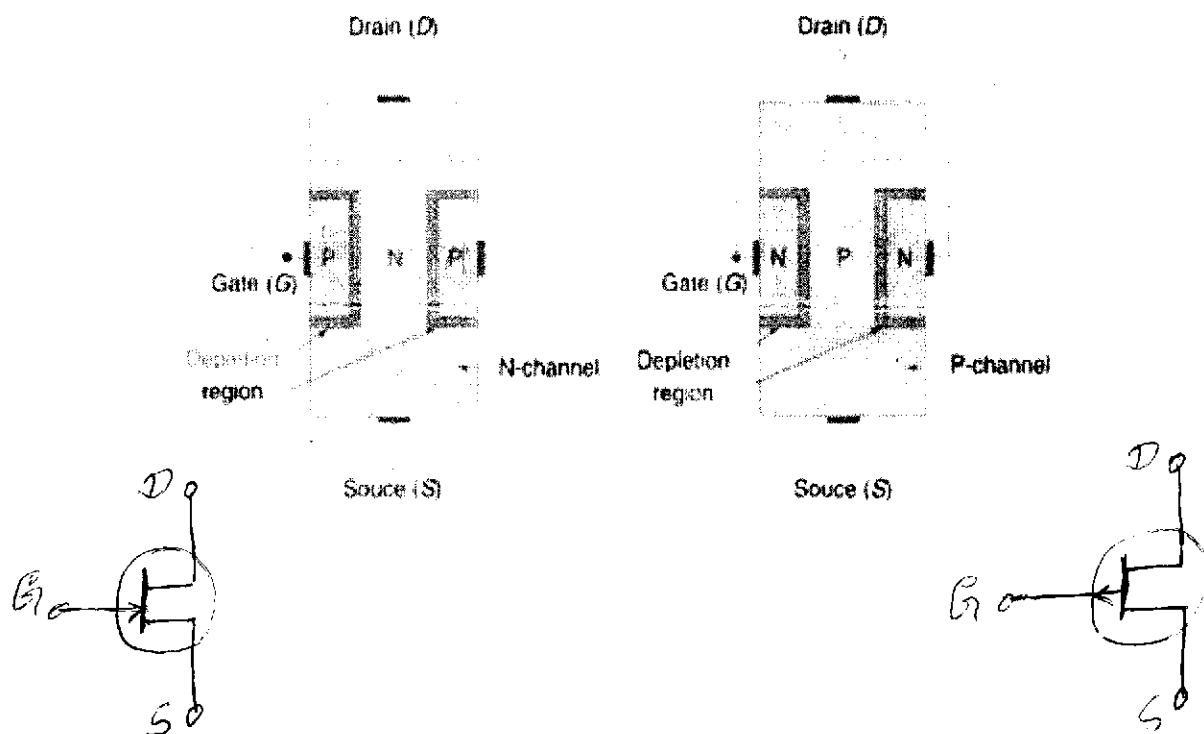
FIELD EFFECT TRANSISTORS

JUNCTION FIELD EFFECT TRANSISTORS (JFETs):

JFET is the simplest of the FETs. **JFET** is a three-terminal device where the voltage applied at one terminal controls the current through the other two terminals.

JFETs comprise a semiconductor channel embedded into semiconductor layers of opposite polarity. Depending upon the semiconductor channel, JFETs are classified as N-channel or P-channel JFETs.

Construction & Principle of Operation:



The Cross-sectional View of N-channel & P-channel JFETs

In N-channel JFET, an N-type semiconductor material forms a channel between embedded layers of P-type material. In P-channel JFET, a P-type semiconductor forms a channel between the embedded layers of N-type material.

Hence, two PN junctions are formed between the semiconductor channel and the embedded semiconductor layers.

Contacts are made at the top and bottom of the channel and are referred to as the Drain (D) and the Source (S). The channel behaves as a resistive element between its drain and source terminals.

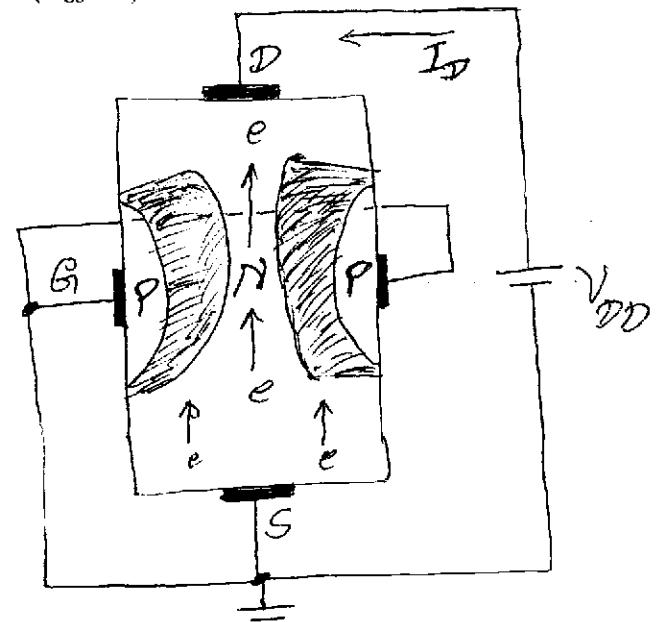
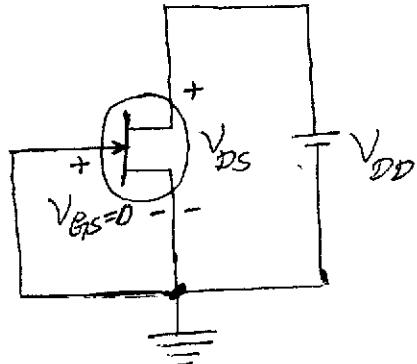
ANALOG AND DIGITAL ELECTRONICS

In an N-channel JFET, both the embedded P-type layers are connected together and form the Gate (G) terminal. Similarly, in a P-channel JFET, Gate (G) terminal is formed by connecting the two N-type embedded layers.

In the absence of any externally applied potential, both the PN junctions are ^{open} circuit and a small depletion region is formed at each of the junction (as shown in above Fig). The externally applied potential between the Gate and the Source terminals controls the flow of Drain current for a given potential between the Drain and the Source terminals.

Characteristic Curves:

Consider an N-channel JFET with a situation when a positive Drain-Source voltage (V_{DS}) is applied to the JFET with the Gate terminal shorted to the Source terminal ($V_{GS} = 0$).

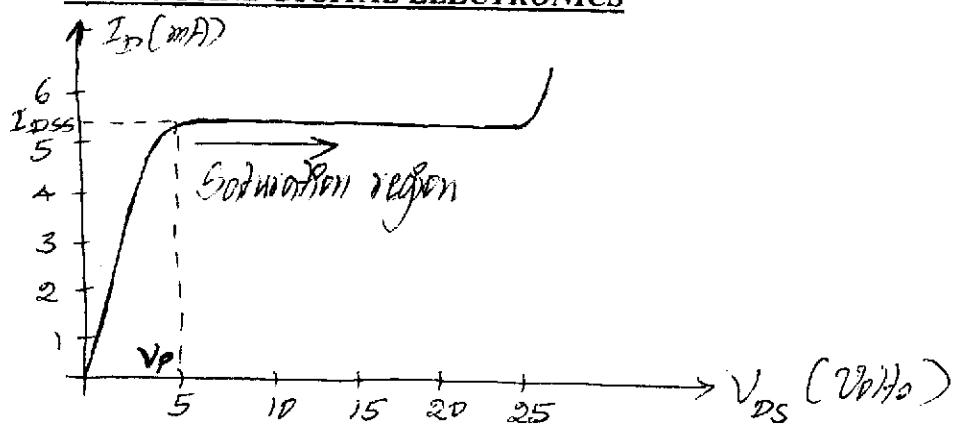


N-channel JFET with $V_{GS} = 0$ and Positive Value of V_{DS}

When the Drain-Source voltage is applied, the electrons in the N-channel are attracted to the Drain terminal, establishing the flow of Drain current (I_D) as shown in the above Fig. The value of I_D is determined by the value of applied V_{DS} and the resistance of the N-channel between the Drain and the Source terminals.

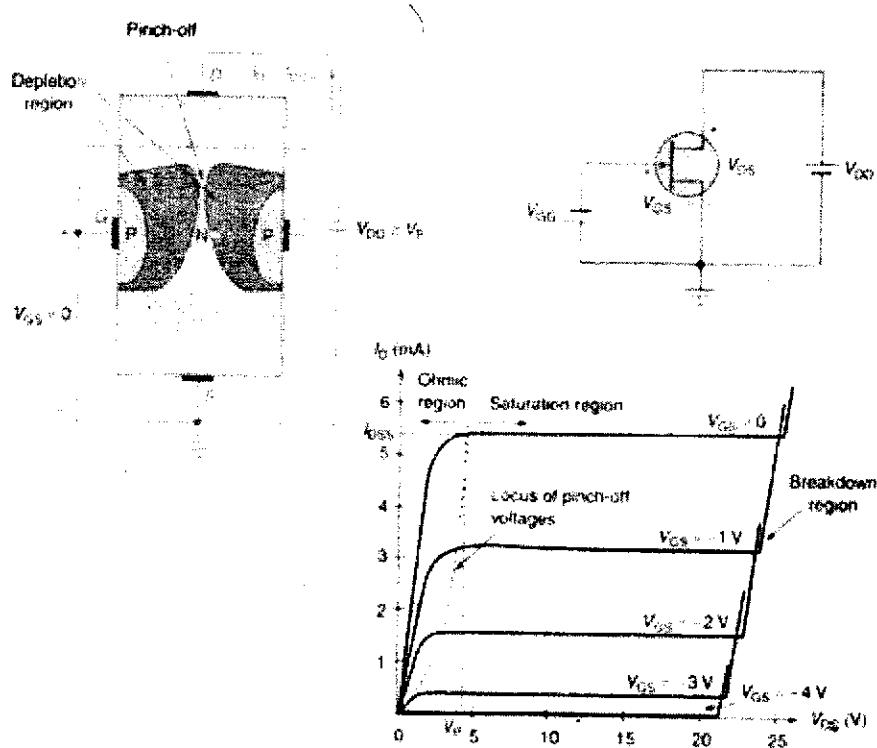
Due to the flow of I_D , there is a uniform voltage drop across the channel resistance, which reverse biases the two PN junctions. This results in increase in the width of the depletion region. The depletion region is wider near the Drain-region than the Source-region. This is because I_D and the channel resistance establish more reverse-bias voltage at the PN junction near the Drain-region than the Source-region.

ANALOG AND DIGITAL ELECTRONICS



The Relationship between I_D and V_{DS} for $V_{GS} = 0$

I_D increases linearly with increase in V_{DS} , till the V_{DS} reaches a value where the saturation effect sets in. The value of V_{DS} where saturation effect sets in is referred to as *pinch-off voltage* (V_p). When V_{DS} reaches V_p , the value I_D does not change with further increase in V_{DS} . This condition is referred to as *pinch-off condition*. This happens because the width of the depletion regions of the PN junctions has increased significantly near the drain region, resulting in reduction of channel width. Hence, I_D remain constant for $V_{DS} > V_p$.



N-channel JFET Biasing Circuit & Output Characteristic Curves

The Gate-Source voltage (V_{GS}) is the control voltage for JFETs. When a negative bias is applied to the Gate terminal, there is an increase in width of the depletion region. Hence, pinch-off phenomenon occurs at lower values of V_{DS} . Also, the value of Saturation Drain current decreases. As the value of V_{GS}

ANALOG AND DIGITAL ELECTRONICS

becomes more negative the value of saturation current decreases further. The Drain current becomes zero for $V_{GS} = -V_p$. This voltage is referred to as *Gate-Source cut-off voltage* or the *Gate-Source pinch-off voltage* ($V_{GS(\text{off})}$).

In the output characteristic curve (Fig shown above), the region to the left of the locus of pinch-off voltages is the *Ohmic region* or the *Voltage-Controlled Resistance region*. The region to the right of the locus of pinch-off voltages is the *Saturation region* or the *Constant Current region*. In Ohmic region, JFET acts as a variable resistor whose resistance is controlled by the applied Gate-Source voltage.

The Drain resistance in the saturation region is given by;

where, r_0 – is the resistance at $V_{GS} = 0$

r_d – is the resistance at a particular value of V_{GS} .

$$r_d = \frac{r_0}{(1 - V_{GS}/V_p)^2}$$

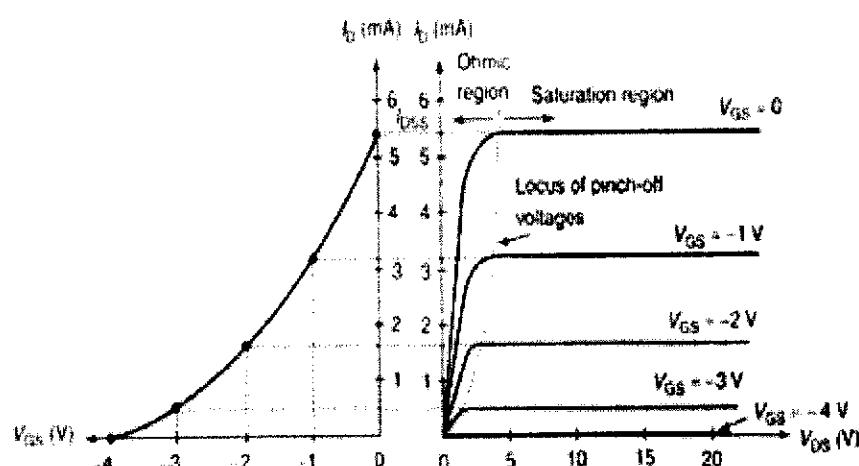
The relationship between the output current I_D in the saturation region for the given value of input V_{GS} is given by (*Shockley's equation*);

$$I_D = I_{DSS} \left[1 - \frac{V_{GS}}{V_p} \right]^2$$

where, I_{DSS} – is the Drain current for short circuit connection between Gate and the Source.

From this Shockley's equation, it is clear that, there is a non-linear square law relationship between the output Drain current (I_D) and the input Gate-Source voltage (V_{GS}). Because of this square law characteristic, JFETs are very useful devices in radio tuners and TV receivers.

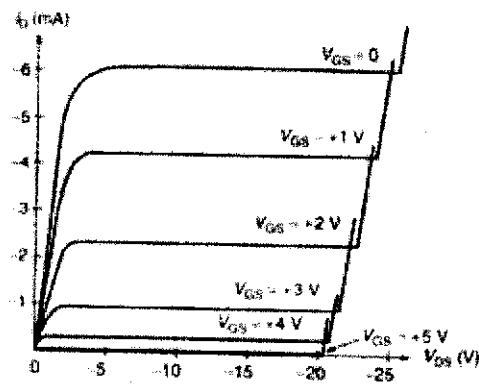
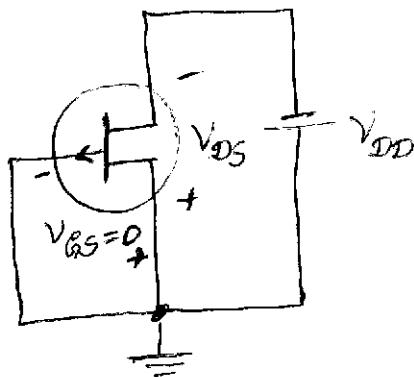
The transfer characteristic of a JFET device is a plot between I_D and V_{GS} . This can be plotted using Shockley's equation or using the output characteristics curves. The following Fig shows how to obtain the transfer characteristics curves using the output characteristic curves.



Transfer Characteristic Curves of N-channel JFET

The P-channel JFETs behave in the same manner as the N-channel JFETs with the direction of currents and polarities of voltages reversed.

ANALOG AND DIGITAL ELECTRONICS



P-channel JFET Biasing Circuit & Output Characteristic Curves

Effect of Temperature on JFET Parameters:

JFETs offer better thermal stability as compared to BJTs. Increase in JFET temperature results in *decrease in the depletion region width* and the *decrease in the carrier mobility*.

- *Decrease in the width of depletion region* results in increase in channel width; which in turn increases in I_D . This results in positive temperature coefficient for I_D . Increase in I_D with temperature results in increase in $V_{GS(OFF)}$. $V_{GS(OFF)}$ has a positive temperature coefficient of the order of $2.2 \text{ mV}^{\circ}\text{C}$.
- *Decrease in carrier mobility* gives a negative temperature coefficient.

Since both mechanisms occur simultaneously, the effect of one mechanism compensates for the other. Hence, JFETs offer better temperature stability.

METALOXIDE FIELD EFFECT TRANSISTORS (MOSFETS):

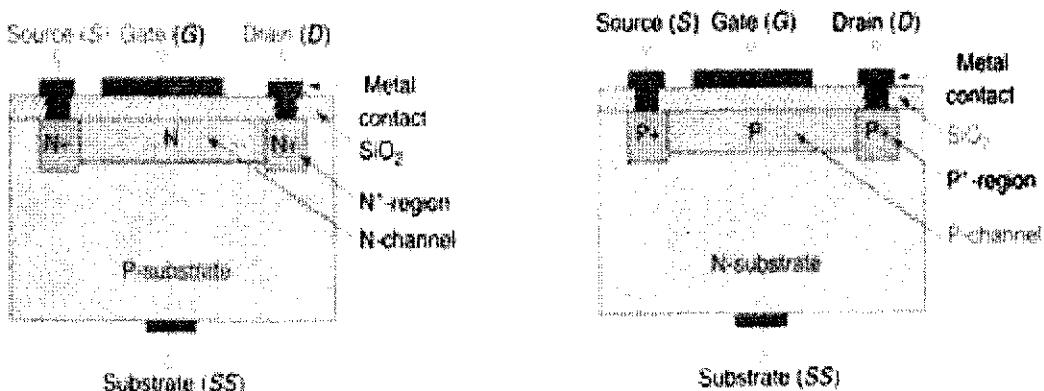
A MOSFET is also a three-terminal device where the Drain current is controlled by the applied Gate voltage (like a JFET). MOSFETs are also referred to as Insulated Gate FETs (IGFETs). MOSFET is insulated from the semiconductor channel by a very thin oxide layer.

MOSFETs are classified into two types depending upon their construction and mode of operation – the *depletion MOSFET (DE-MOSFET)* the *enhancement MOSFET (E-MOSFET)*.

Depletion MOSFETs:

In a DE-MOSFET, a channel is physically constructed between the Drain and the Source terminals. Depending on the channel material, DE-MOSFETs are classified as *N-Channel DE-MOSFETs* and *P-channel DE-MOSFETs*.

ANALOG AND DIGITAL ELECTRONICS

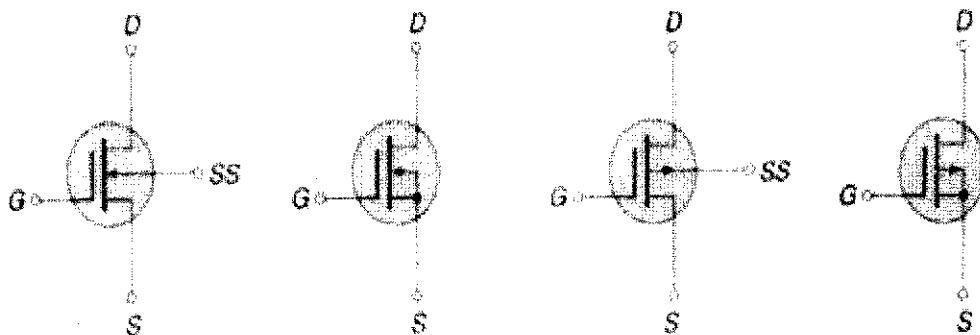


Cross-section of an N-channel & P-channel DE-MOSFET

The cross-sectional view of an N-channel DE-MOSFET comprises a substrate made of a P-type semiconductor material. Two N+ type regions linked by an N-channel are formed on the substrate. The Source and the Drain terminals are formed by connecting metal contacts to the two N+ regions (as shown in the above Fig). The Gate terminal is connected to the insulating silicon dioxide (SiO₂) layer on the top of the N-channel. Hence, there is no direct electrical connection between the Gate terminal and the channel of DE-MOSFET.

There is a capacitance that exists between the Gate terminal and the channel as the metal Gate contact and the channel act as walls of a parallel plate capacitor and the SiO₂ layer form the dielectric. Hence, the input impedance of a DE-MOSFET is very high (in the order of $10^{10} - 10^{15} \Omega$).

The construction of P-channel DE-MOSFET is similar to that of an N-channel DE-MOSFET, with the difference being that the substrate is an N-type semiconductor and the channel is P-type material.



Circuit Symbol of an N-channel DE-MOSFET & P-channel DE-MOSFET

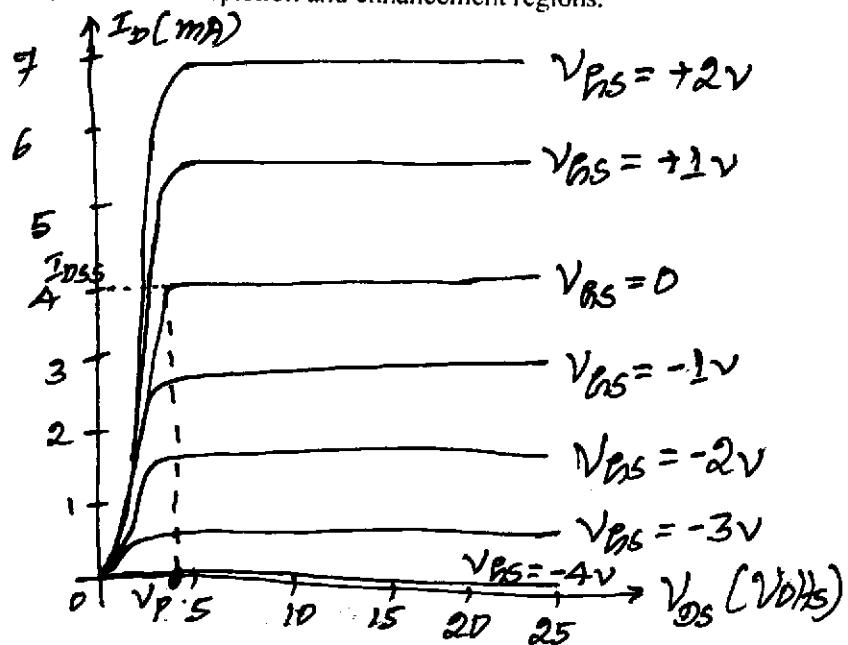
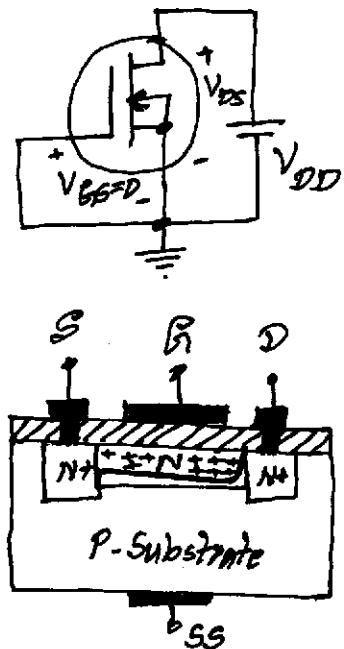
Operation of N-channel DE-MOSFET: When the Gate and the Source terminals are shorted, ($V_{GS} = 0$) and a positive voltage is applied between the Drain the Source terminals; there is a flow of current in the N-channel, as the electrons are attracted by positive potential at the Drain terminal. The current increases with increase in V_{DS} ; and after certain value of V_{DS} , it becomes constant.

ANALOG AND DIGITAL ELECTRONICS

When the Gate terminal is at a negative potential as compared to the Source terminal, electrons in the N-channel are repelled by this negative potential towards the P-type substrate. Also, holes in the P-type substrate are attracted towards the Gate. This results in recombination of holes and electrons and there will be reduction of number of free electrons in the N-channel. Higher the negative potential more is the rate of recombination and less the number of free electrons in the N-channel. Hence, the drain current decreases with increase in the value of the negative Gate-Source potential.

For positive values of Gate-Source voltage, electrons in the P-type substrate are attracted into the channel and establish new carriers through the collisions between accelerating particles. Thus the Drain current increases rapidly with increase in the positive value of Gate-Source voltage.

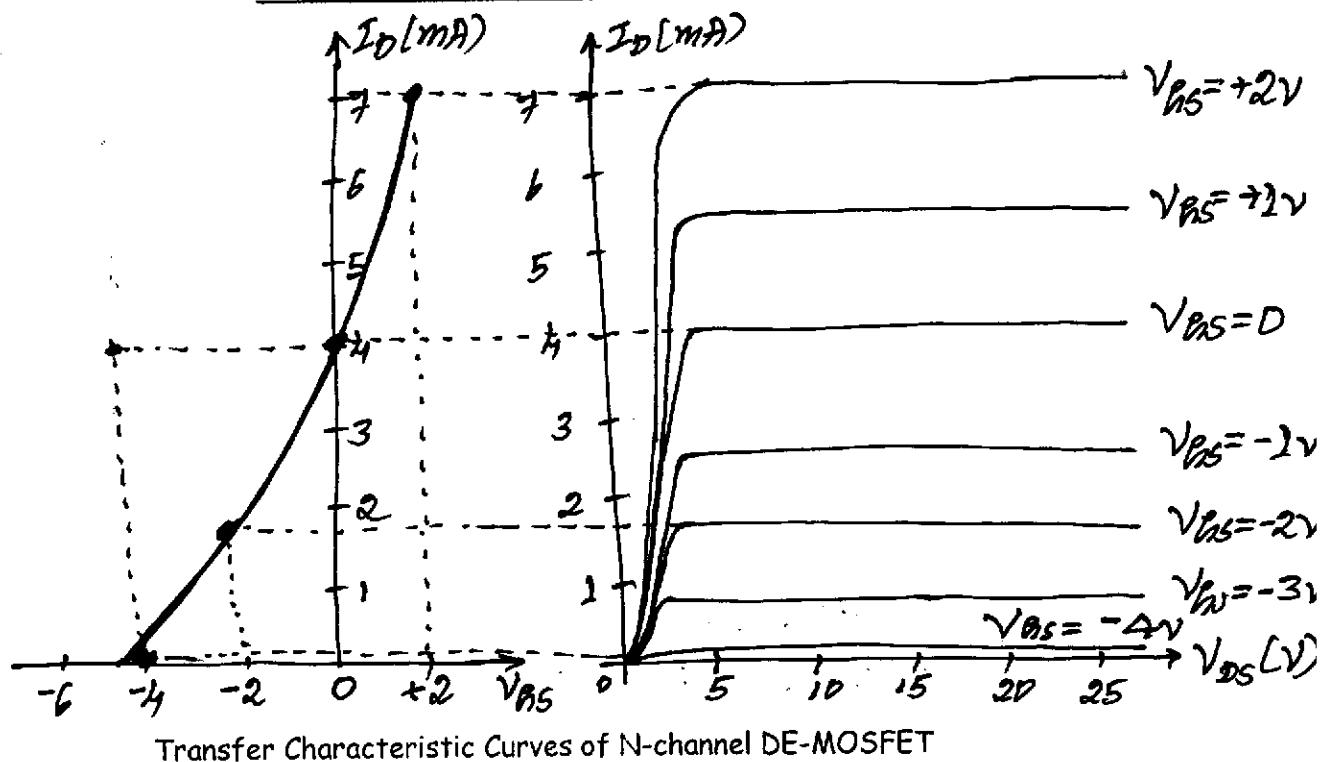
As the application of positive Gate-Source voltage increases the value of Drain current, the region of positive Gate-Source voltage is referred to as the *enhancement region*. The region for zero and negative values of Gate-Source voltage is referred to as *depletion region*. The Shockley's equation defined for JFETs is applicable for DE-MOSFETs, in both the depletion and enhancement regions.



Output Characteristic Curves of N-channel DE-MOSFET

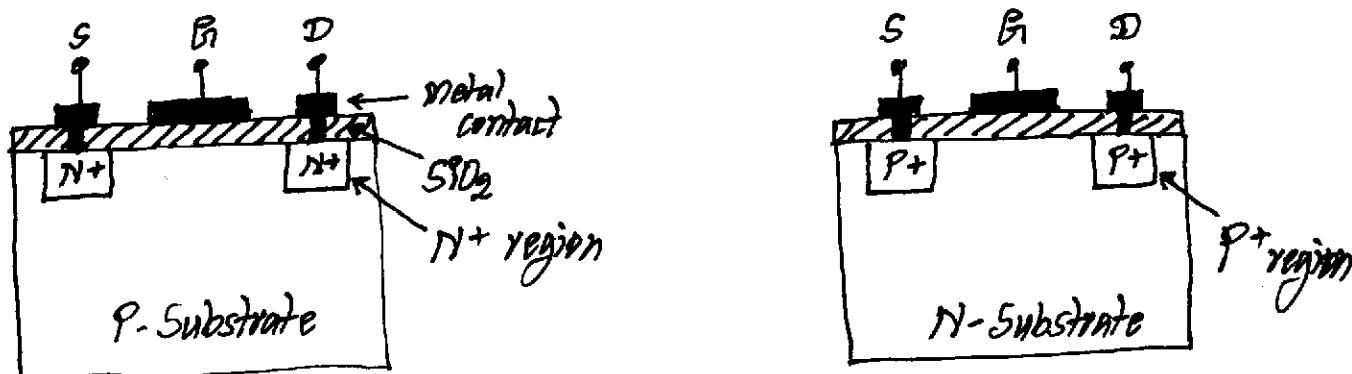
The transfer characteristics for DE-MOSFET can be plotted in a similar fashion for that of a JFET (see the following Fig.).

ANALOG AND DIGITAL ELECTRONICS



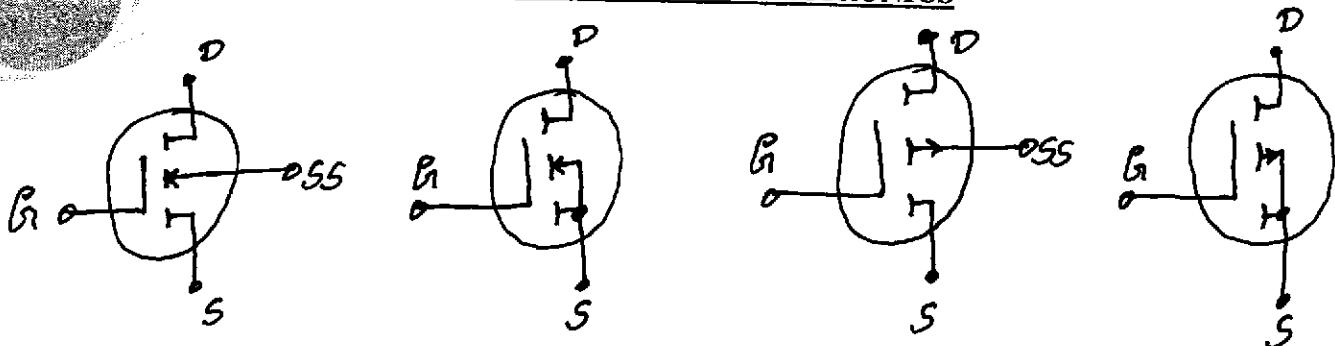
Enhancement MOSFETs:

The construction of an E-MOSFET is similar to that of a DE-MOSFET with the difference that there is no physical channel between the Source and Drain terminals in the E-MOSFET.



Cross-section of an N-channel & P-channel E-MOSFET

ANALOG AND DIGITAL ELECTRONICS

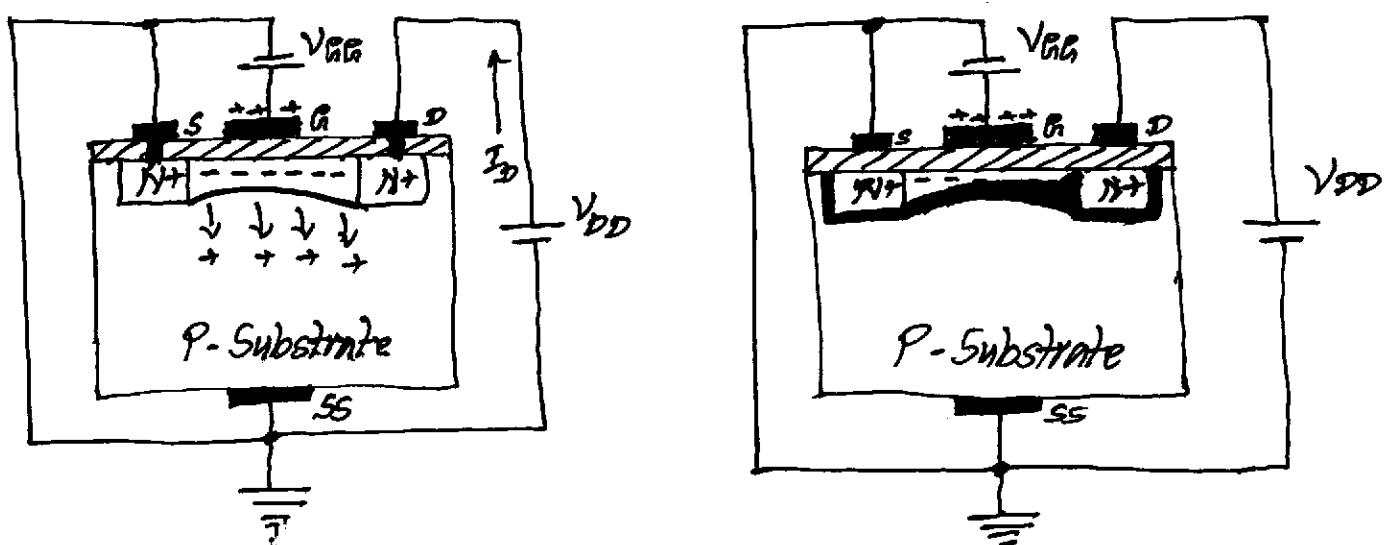


Circuit Symbol of an N-channel DE-MOSFET & P-channel DE-MOSFET

Operation of N-channel DE-MOSFET: When the Gate-Source voltage is zero ($V_{GS} = 0$) and some positive Drain-Source voltage (V_{DS}) is applied, there is no Drain current; as there is no channel available for flow of Drain current. Hence, E-MOSFETs are also referred to as OFF-MOSFETs, as they do not conduct when $V_{GS} = 0$.

When a positive Gate-Source voltage (V_{GS}) is applied, electrons (minority carriers) in the P-type substrate will accumulate near the surface of the SiO_2 layer. Also, holes in the P-substrate are forced to move away from the edge of SiO_2 layer. This forms a channel (as shown in following Fig). As SiO_2 layer is insulating, it prevents the electrons from being absorbed at the Gate terminal. Hence, the Drain current flows.

As the value of Gate-Source voltage is increased, more and more electrons accumulate leading to an enhanced flow of Drain current. The level of Gate-Source voltage that leads to significant flow of Drain current is referred to as *threshold voltage* (V_t).

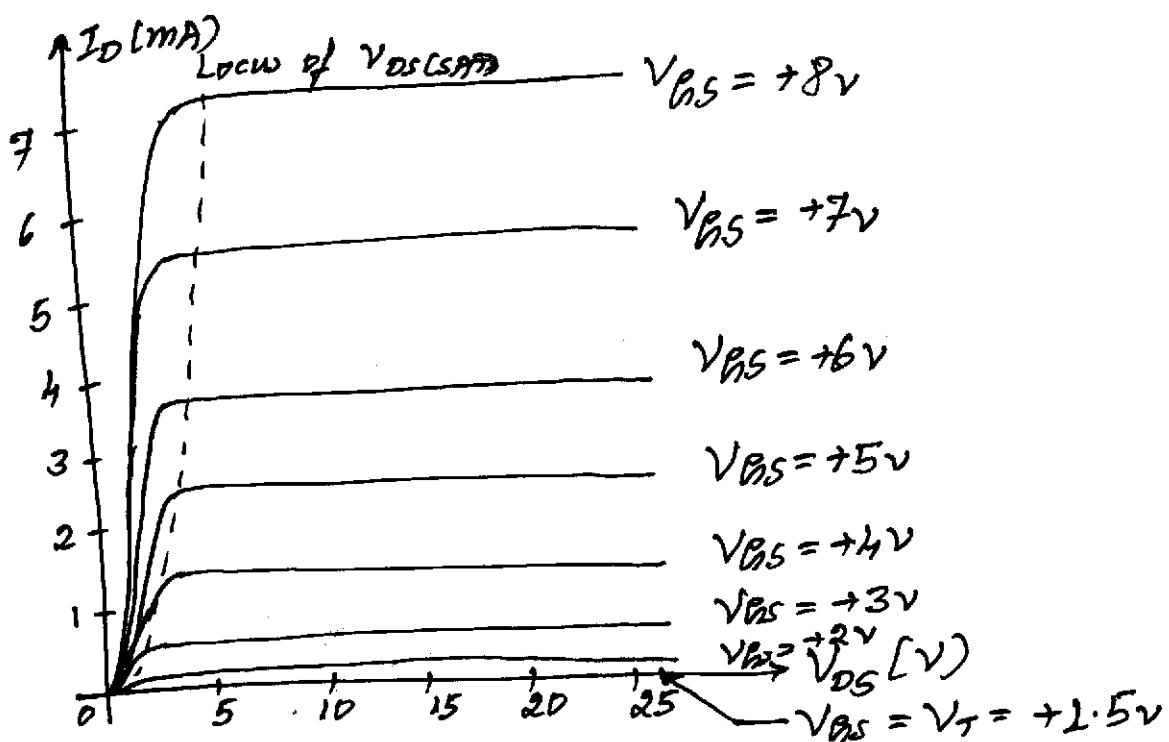


Working of N-channel E-MOSFET & Pinching Phenomenon in E-MOSFET

MAHESH PRASANNA K., VCET, PUTTUR

ANALOG AND DIGITAL ELECTRONICS

For a fixed Gate-Source voltage and increasing the level of Drain-Source voltage (V_{DS}), leads to an initial increase in the Drain current, which eventually saturates due to the reduction of Gate-Drain voltage (V_{GD}). Reduction in the Gate-Drain voltage reduces the attractive forces for the free carrier in the induced channel near the Drain region. This results in the reduction of effective channel width near the Drain region. This effect is referred to as *pinching effect*. Pinching effect refers to the reduction in the width of the induced channel near the Drain region with increase in the Drain-Source voltage (as shown in the above Fig). The value of Drain-Source voltage at which the Drain current saturates is given by $V_{DS(SAT)}$.



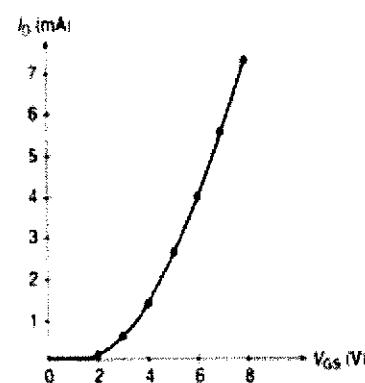
Output Characteristic Curves for an N-channel E-MOSFET

The relationship between $V_{DS(SAT)}$ and V_{GS} is given by; $V_{DS(SAT)} = V_{GS} - V_T$

where, V_T is the threshold Gate-Source voltage.

Also, the Drain current is zero for Gate-Source voltage less than the threshold voltage V_T .
 For voltages greater than the threshold voltage, the Drain current is given by; $I_D = K (V_{GS} - V_T)^2$
 where, K is a constant.

The relationship between the Drain current and the Gate-Source voltage is non-linear and the current is proportional to the square of the voltage (shown in the Fig).



Transfer Characteristics of an N-channel E-MOSFET



ANALOG AND DIGITAL ELECTRONICS

Problem: Draw the output characteristics of an N-channel E-MOSFET, when the Gate-Source voltage (V_{GS}) is twice the threshold voltage (V_T). An external supply V_{GG} is applied between the Gate and the Drain terminals. Draw the curve between the Drain current and voltage V_D , for $V_{GG} = V_T/2$.

Solution:

- E-MOSFETs conduct for $V_{GS} > V_T$.

We have; $I_D = K(V_{GS} - V_T)^2$

Let I_{DS} is the current for $V_{GS} = 2V_T$.

$$\Rightarrow I_{DS} = K(2V_T - V_T)^2 \Rightarrow K = I_{DS}/V_T^2$$

- The output characteristics can be plotted by using the equations;

$$I_D = \frac{I_{DS}}{V_T^2} (V_{GS} - V_T)^2 \quad V_{DS(SAT)} = V_{GS} - V_T$$

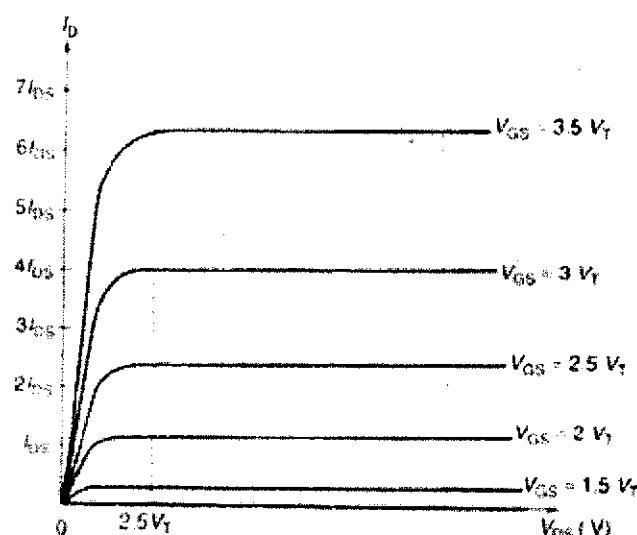
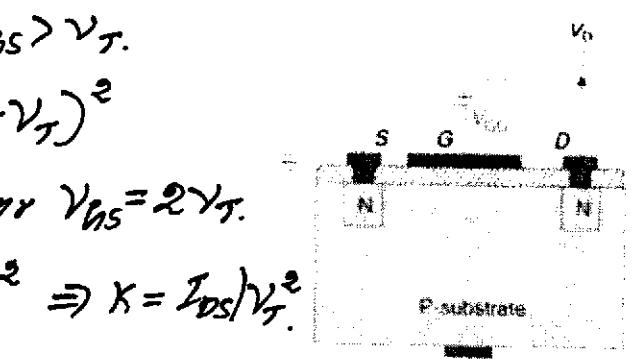
- Since voltage V_{GG} is applied b/w G & D-terminals;

$$V_{GS} = V_{GG} + V_{DS} = V_{GG} + V_D \quad [\because V_D = V_{DS}]$$

- Given, $V_{GG} = V_T/2 \Rightarrow V_D/V_{DS} = V_{GS} - V_T/2$

V_{GS}	I_D	V_{DS}/V_D
$0.5V_T$	0	0
V_T	0	$0.5V_T$
$1.5V_T$	$0.25I_{DS}$	V_T
$2V_T$	I_{DS}	$1.5V_T$
$2.5V_T$	$2.25I_{DS}$	$2V_T$
$3V_T$	$4I_{DS}$	$2.5V_T$
$3.5V_T$	$6.25I_{DS}$	$3V_T$

MAHESH PRASANNA K., VCET, PUTTUR



ANALOG AND DIGITAL ELECTRONICS

DIFFERENCES BETWEEN JFETs AND MOSFETs:

JFETs	MOSFETs
1. JFETs are operated in depletion mode only.	1. DE-MOSFETs can be operated in both depletion and enhancement modes and E-MOSFETs are operated in enhancement mode only.
2. Input resistance for JFETs is greater than $10^9 \Omega$.	2. Input resistance for MOSFETs is much higher than JFETs (around $10^{13} \Omega$).
3. JFETs have higher Drain resistance (in the range of $100 \text{ k}\Omega$ to $1 \text{ M}\Omega$) than MOSFETs; and hence their characteristic curve is more flat than that of MOSFETs	3. The Drain resistance of MOSFETs is in the range of 1 to $50 \text{ k}\Omega$.
4. The Gate current for JFETs is in the range of $100 \mu\text{A}$ to 10 nA .	4. The leakage current in MOSFET is much smaller than that in JFETs. The Gate current for MOSFETs is in the range of 100 nA to 10 pA .
5. MOSFETs are easier to construct and are used more widely than JFETs.	

Biasing MOSFETs:

Biasing is done to produce the required Gate-to-Source voltage (V_{GS}) to get the desired value of Drain current (I_D). The biasing circuits should maintain the Drain current and Drain-Source voltage within reasonable limits.

Depletion MOSFETs:

Problem: The following Fig shows a biasing configuration using DE-MOSFET. Given that the saturation Drain current is 8 mA and the pinch-off voltage is -2 V ; determine the value of Gate-Source voltage, Drain current and the Drain-Source voltage.

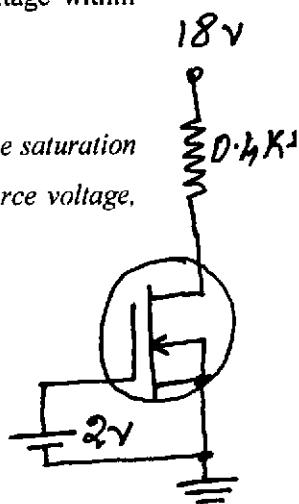
Solution:

o In a DE-MOSFET, $I_D = I_{DSS} \left[1 - \frac{V_{GS}}{V_P} \right]^2$
 $\Rightarrow I_D = 8 \times 10^{-3} \left[1 - \frac{2}{(-2)} \right]^2 = 32 \times 10^{-3} = \underline{\underline{32 \text{ mA}}}$

o Applying Kirchhoff's voltage law to the output section, we get; $-V_{DD} + 0.4 \times 10^3 \times I_D + V_{DS} = 0$
 $\text{or } -18 + 0.4 \times 10^3 \times 32 \times 10^{-3} + V_{DS} = 0$

$$\Rightarrow V_{DS} = 18 - 12.8 = \underline{\underline{5.2 \text{ V}}}$$

MAHESH PRASANNA K., VCET, PUTTUR



- $V_{GS} = 2 \text{ V}$
- MOSFET operating in enhancement region.
- $I_{DSS} = 8 \text{ mA}$
- $V_P = -2 \text{ V}$
- $V_{DD} = 18 \text{ V}$

ANALOG AND DIGITAL ELECTRONICS

Hence; Gate-Source voltage, $V_{GS} = 2V$

Drain current, $I_D = 32mA$

Drain-Source voltage, $V_{DS} = 5.2V$.

Problem: Design a voltage-divider-bias network using DE-MOSFET with the supply voltage $V_{DD} = 16V$, $I_{DSS} = 10mA$ and $V_P = -5V$ to have a quiescent Drain current of $5mA$ and Gate voltage of $4V$ (Assume the Drain resistor R_D to be four times the source resistor R_S).

Solution: Given; $V_{DD} = 16V$, $I_{DSS} = 10mA$, $V_P = -5V$, $I_D = 5mA$, & $V_G = 4V$.

Since, $I_D < I_{DSS}$; the MOSFET is operated in depletion mode.

We have; $I_D = I_{DSS} \left[1 - \frac{V_{GS}}{V_P} \right]^2$

i.e., $5 \times 10^{-3} = 10 \times 10^{-3} \left[1 - \frac{V_{GS}}{(-5)} \right]^2$ or $0.5 = \left(1 + \frac{V_{GS}}{5} \right)^2$

$$\Rightarrow V_{GS}/5 = 0.7 - 1 = -1.5V$$

Also; $V_{GS} = V_G - V_S = V_G - I_D R_S$

i.e., $-1.5 = 4 - V_S$

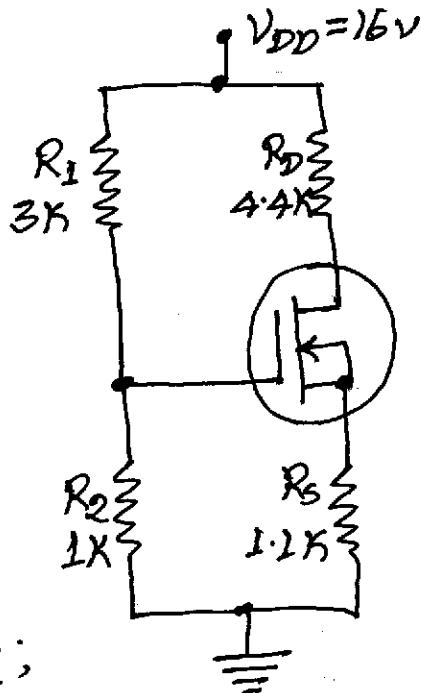
or $V_S = I_D R_S = 4 \times 1.5 = 5.5V$

$$\therefore R_S = \frac{5.5}{5 \times 10^{-3}} = 1.1k\Omega$$

We have; $R_D = 4R_S \Rightarrow R_D = 4.4k\Omega$.

$V_B = \left[\frac{R_2}{R_1 + R_2} \right] V_{DD}$ Assume, $R_2 = 1k\Omega$;

$$\Rightarrow A = \left[\frac{1 \times 10^3}{R_1 + 1 \times 10^3} \right] 16 \quad \text{or} \quad R_1 = 3k\Omega$$

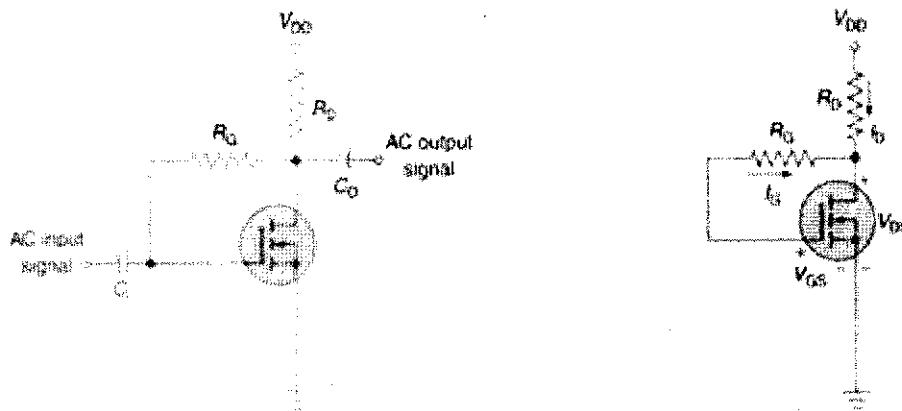


Enhancement MOSFETs: Two most popular biasing configurations for E-MOSFETs are the feedback biasing configuration and the voltage-divider configuration.

Feedback Biasing Configuration:

MAHESH PRASANNA K., VCET, PUTTUR

ANALOG AND DIGITAL ELECTRONICS



Feedback Biasing Configuration for N-channel E-MOSFET & DC Equivalent Circuit

The feedback connection to the Gate terminal is taken from the Drain terminal through resistor R_G . The resistor R_G brings bias voltage to the Gate terminal to turn the MOSFET on.

Applying Kirchhoff's voltage law to the input section, we get; $V_{DD} - I_D R_D + I_G R_G - V_{GS} = 0$

But, Gate current is approximately equal to zero; and hence voltage drop across R_G will be approximately equal to zero; i.e., $I_G R_G = 0$ Hence, we get; $V_{DD} - I_D R_D - V_{GS} = 0$

$$\text{Or; } V_{GS} = V_{DD} - I_D R_D \quad \dots \dots \dots (1)$$

Applying Kirchhoff's voltage law to the output section, we get; $V_{DD} - I_D R_D - V_{DS} = 0$

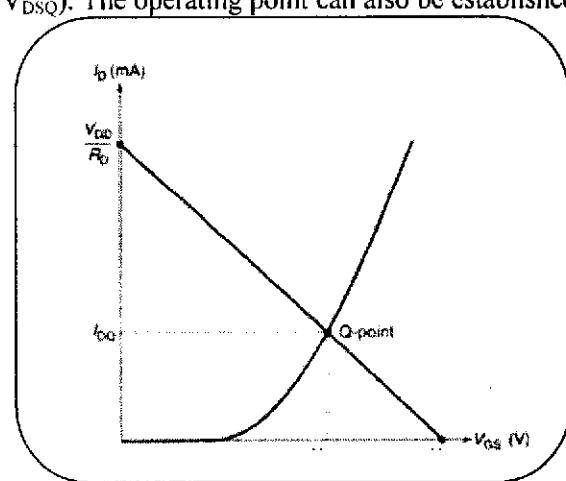
$$\text{Or; } V_{DS} = V_{DD} - I_D R_D \quad \dots \dots \dots (2)$$

It is clear from equations (1) and (2) that, the Gate-Source voltage (V_{GS}) and the Drain-Source voltage (V_{DS}) are equal; i.e., $V_{GS} = V_{DS}$

The operating point (Q-point) is given by (I_{DQ}, V_{DSQ}) . The operating point can also be established by using graph method (shown in following Fig).

The operating point can be obtained by –

- ✓ Superimposing the equation (1) on the transfer characteristics of the MOSFET, or
- ✓ Superimposing the DC load line defined by equation (2) on the output characteristic curves of the MOSFET.



Graphical Method for Determining the Q-point for N-channel E-MOSFET

Problem: The following Fig shows a circuit using E-MOSFET. Given that, the threshold voltage for the MOSFET is 2 V and $I_{D(on)} = 6 \text{ mA}$ for $V_{G(on)} = 5 \text{ V}$; determine the value of the operating point.

MAHESH PRASANNA K., VCET, PUTTUR

ANALOG AND DIGITAL ELECTRONICS

Solution:

• We have; $I_D = K(V_{BS} - V_T)^2$

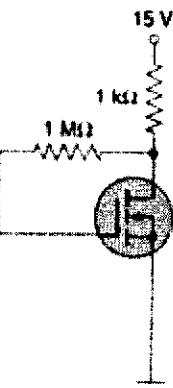
$$\therefore K = 6 \times 10^{-3} / (5-2)^2 = 0.67 \text{ mA/V}^2$$

• The gate-source voltage in the feedback configuration is;

$$V_{BS} = V_{DD} - I_D R_D$$

$$= 15 - I_D * 1 \times 10^3$$

$$= 15 - 1000 I_D.$$



$$R_D = 1 \times 10^3$$

$$R_B = 1 \text{ M}\Omega$$

$$V_{BS} = 5 \text{ V}$$

$$V_T = 2 \text{ V}$$

$$I_D = 6 \text{ mA}$$

• Substituting the value of I_D in the expression;

$$I_D = K(V_{BS} - V_T)^2, \text{ we get;}$$

$$I_D = 0.67 \times 10^{-3} (15 - 1000 I_D - 2)^2$$

$$= 0.67 \times 10^{-3} (13 - 1000 I_D)^2$$

$$\text{or } 3000 I_D = 2 (169 - 26000 I_D + 10^6 I_D^2)$$

$$\text{or } 2 \times 10^6 I_D^2 - 55000 I_D + 338 = 0 \Rightarrow I_D = 9.3 \text{ mA}$$

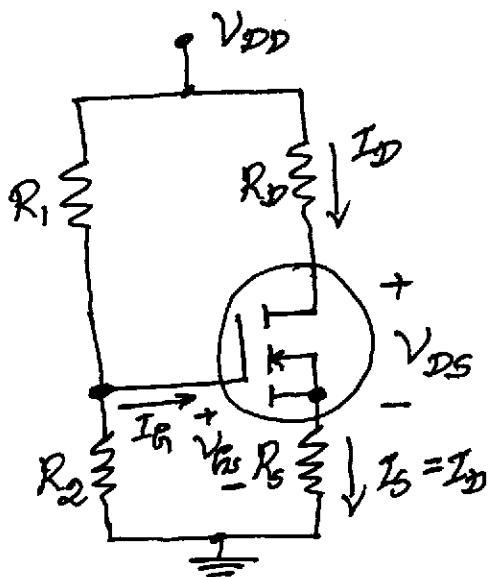
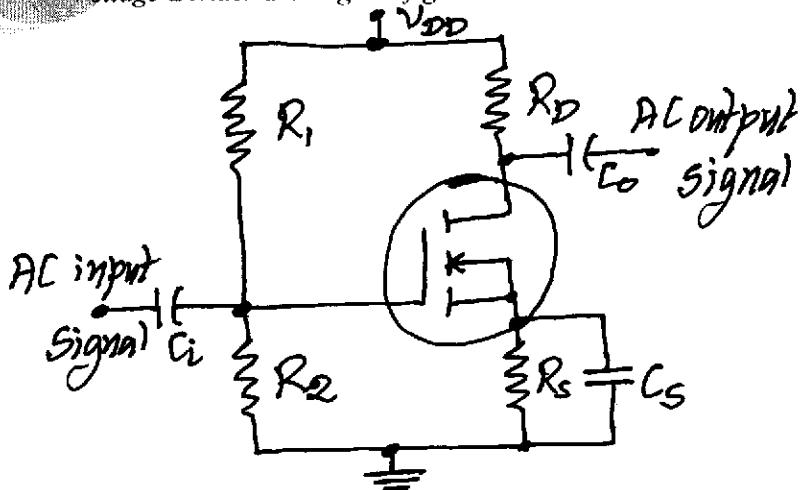
• The drain-source voltage is given by; $V_{DS} = V_{DD} - I_D R_S$

$$\text{or } V_{DS} = 15 - 9.3 \times 10^{-3} \times 1 \times 10^3 = 15 - 9.3 = 5.7 \text{ V.}$$

• Hence, the operating point is (9.3mA, 5.7V).

ANALOG AND DIGITAL ELECTRONICS

Voltage-Divider-Biasing Configuration:



Voltage-Divider-Biasing Configuration for N-channel E-MOSFET & DC Equivalent Circuit

Problem: The following Fig shows a voltage-divider configuration for the E-MOSFET. Given that the threshold voltage for the MOSFET is 4 V and value of $I_{D(on)} = 6 \text{ mA}$ for $V_{GS(on)} = 8 \text{ V}$; use graphical method to determine the value of the Drain current, Gate-Source voltage and Drain-Source voltage.

Solution:

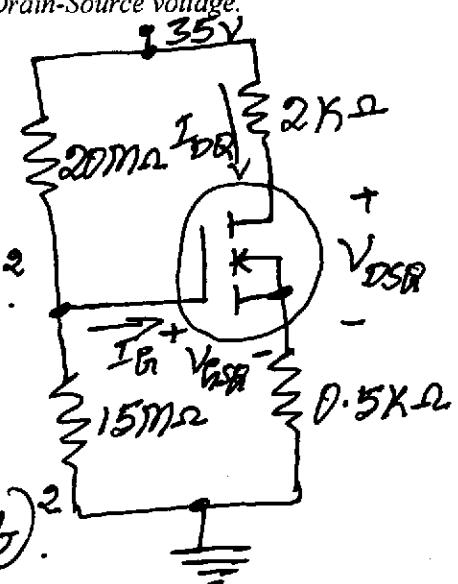
• We have; $I_{D(on)} = K(V_{GS(on)} - V_T)^2$

$$\Rightarrow K = \frac{6 \times 10^{-3}}{(8 - 4)^2} = 0.375 \text{ mA/V}^2$$

• Therefore, the value of the drain

current is given by; $I_D = K(V_{GS} - V_T)^2$.

$$\text{or } I_D = 0.375 \times 10^{-3} (V_{GS} - V_T)^2$$



$$R_1 = 20\text{m}\Omega \quad R_2 = 15\text{m}\Omega \quad R_D = 2\text{k}\Omega \quad R_S = 0.5\text{k}\Omega$$

V_{GS}	I_D
5V	0.375 mA
7.5V	4.59 mA
10V	13.5 mA
12.5V	27.09 mA
15V	45.375 mA

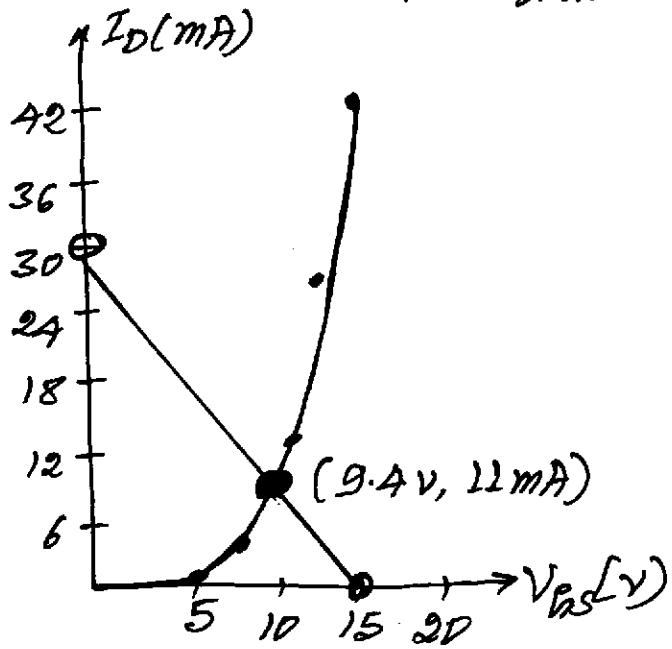
$$V_T = 4V$$

$$I_{D(on)} = 6 \text{ mA}$$

$$V_{GS(on)} = 8V$$

ANALOG AND DIGITAL ELECTRONICS

- Using the values of V_{BS} & I_D , the transfer characteristics of the MOSFET can be drawn.



- The gate voltage is;

$$V_B = V_{DD} \frac{R_2}{R_1 + R_2}$$

$$= 35 \times \frac{15 \times 10^6}{(20 + 15) \times 10^6}$$

$$= 15 \text{ V.}$$

- The gate-source voltage is;

$$V_{BS} = V_B - V_S = V_B - I_D R_S$$

- For $I_D = 0$; $V_{BS} = 15 \text{ V.}$

- For $V_{BS} = 0$; $I_D = 15 / 500 = 30 \text{ mA.}$

- Hence, the load-line coordinates are $(0, 15 \text{ V})$ & $(30 \text{ mA}, 0)$.

From the above fig.; the quiescent values of gate-source voltage and the drain current are 9.4 V & 11 mA . respectively.

- The drain-source voltage is given by;

$$V_{DS} = V_{DD} - I_D (R_D + R_S)$$

$$= 35 - 11 \times 10^{-3} (2 \times 10^3 + 0.5 \times 10^3)$$

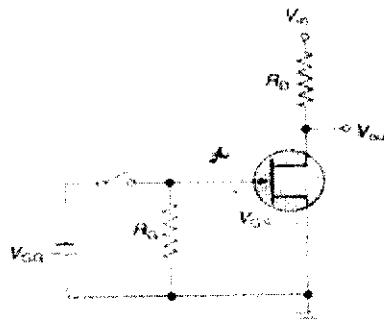
$$= \underline{\underline{7.5 \text{ V.}}}$$

ANALOG AND DIGITAL ELECTRONICS

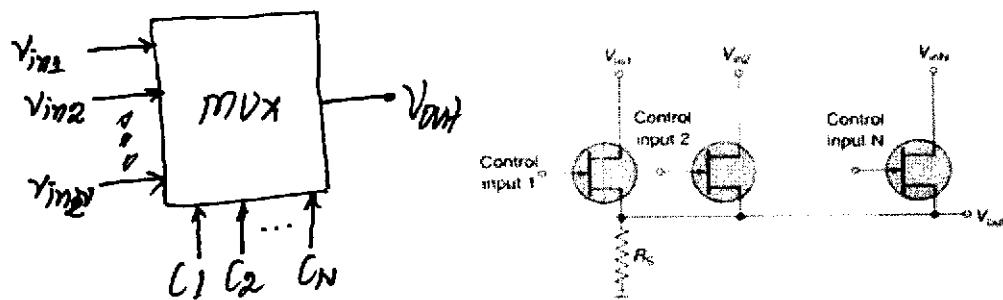
FET APPLICATIONS:

FETs offer very high value of input impedance as compared to BJTs. Some of the common applications of FETs are discussed below:

- 1. Amplifiers:** FET devices are commonly used as low-noise amplifiers and buffer amplifiers. FETs are low-noise devices and hence they are used in the front-end of receivers and other electronic systems. JFETs in common-drain configuration offer high input impedance and low output impedance; and hence, they are used as buffer amplifiers to isolate the preceding stage from the following stage.
- 2. Analog Switch:** FETs are used as analog switches (as shown in the following Fig). When no Gate voltage (V_{GG}) is applied, the FET operates in the saturation region and acts as a closed switch. When a negative Gate voltage is applied, the FET operates in cut-off region (offers very high resistance and acts as open switch).

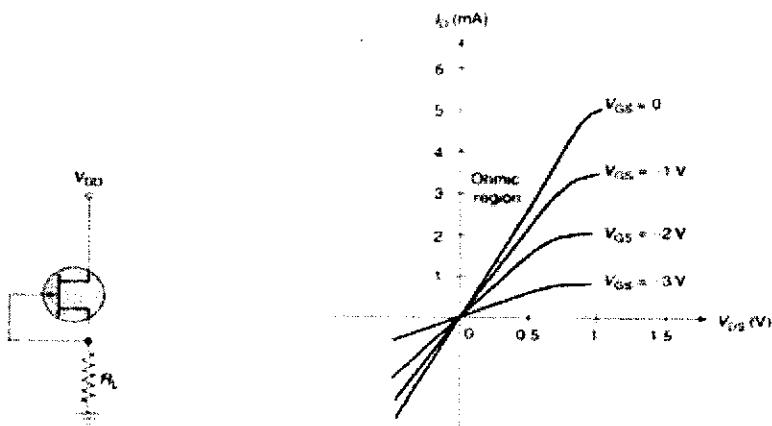


- 3. Multiplexers:** FETs are used in multiplexer circuits where each FET device acts as single-pole single-through switch (as shown in the following Fig). The input signals are applied to the Drain terminals of the JFETs, while the corresponding control inputs are applied to the Gate terminals. When a control input is zero, the input is transferred to the output. All other control inputs will be made more negative than the $V_{GS(off)}$ voltage; hence, all the other input signals are blocked.

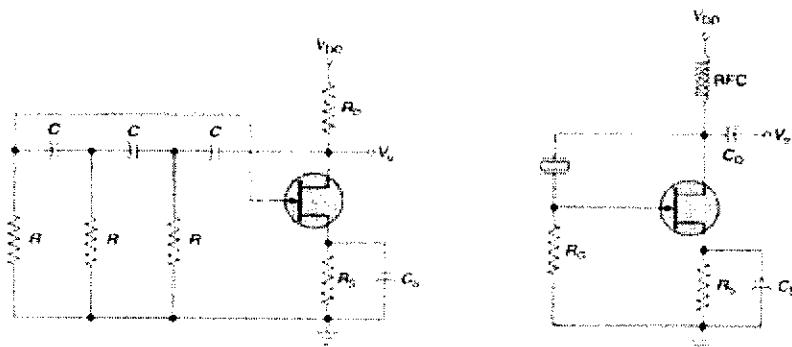


- 4. Current Limiters:** FETs can also be used in current limiting applications (as shown in the following Fig). During the normal operation of the circuit, the JFET acts in the Ohmic region. When the load current increases (due to short-circuit or other reasons), the JFET operates in the saturation region. Hence, it acts as a constant current source and prevents excessive current through the load.

ANALOG AND DIGITAL ELECTRONICS



5. **Voltage-Variable Resistors (VVRs):** FETs when operated in the Ohmic region (for small positive values of V_{DS}), acts as voltage-variable resistors. In this region the Drain resistance can be controlled by the V_{GS} . For an N-channel FET, the value of R_D increases with increase in the negative value of V_{GS} (as shown in the above Fig). If an AC voltage (small peak-to-peak) is applied between the Drain the Source terminals, then FET acts as a linear resistor, for a given Gate-Source voltage. FET based VVRs are used in automatic gain control circuits.
6. **Oscillators:** FETs are also used in phase-shift oscillators & crystal controlled oscillators (as shown in the following Fig).



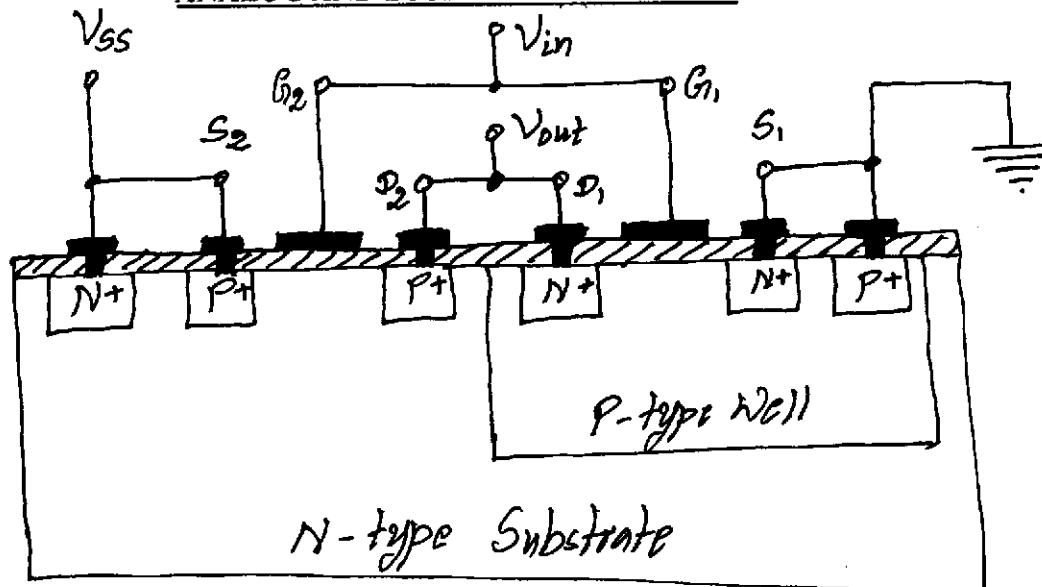
CMOS DEVICES:

Complementary Metal Oxide Semiconductor (CMOS) are those semiconductor devices, in which both P-type and N-type E-MOSFETs are diffused onto the same chip. CMOS devices offer high input impedance, low power consumption, and require far less space, as compared to BJT-based circuits. Hence, CMOS devices are extensively used in computer logic design.

The following Fig shows the basic Inverter circuit using CMOS configuration. Inverter is a logic circuit that inverts the applied input signal (logic LOW to logic HIGH and/or vice-versa). The complementary N-type and P-type E-MOSFETs are connected in series, with their Gate terminals tied together to form the input terminal. Also the Drain terminals are connected together to form the output terminal. Source terminal of P-channel MOSFET is connected to voltage V_{SS} and the Source terminal of N-channel MOSFET is connected to the ground (as shown in the following Fig).

MAHESH PRASANNA K., VCET, PUTTUR

ANALOG AND DIGITAL ELECTRONICS



CMOS Inverter

The simplified circuit diagram of CMOS inverter is shown in the following Fig. When the input voltage V_{in} is logic LOW, the Gate-Source voltage (V_{G2S2}) of the P-channel E-MOSFET is equal to $-V_{ss}$ and the MOSFET is in the ON-state. This provides a low resistance path between the V_{ss} and the output terminal. The Gate-Source voltage (V_{G1S1}) of N-channel E-MOSFET is 0 V. Hence, it is OFF, resulting in very high impedance between the output terminals and the ground. Therefore, the output voltage V_{out} is equal to the supply voltage V_{ss} (or on other words, the output voltage is HIGH).

$$V_{in} = 0$$

P-channel: ON

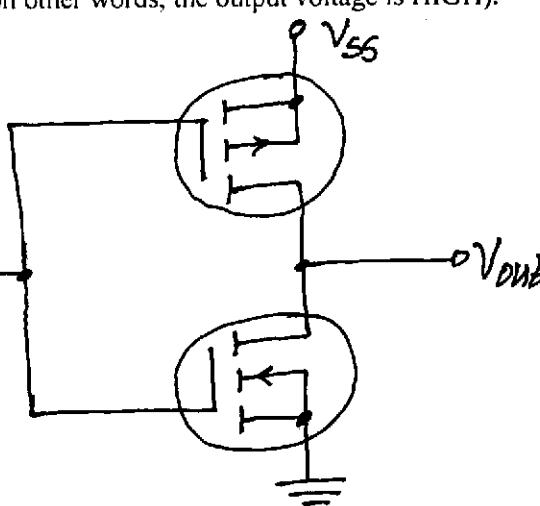
N-channel: OFF

$$V_{in}$$

$$V_{in} = 1 \text{ (High)}$$

P-channel: OFF

N-channel: ON



~~N-channel E-MOSFET:~~

~~+ V_{GS} : OFF~~

~~- V_{GS} : ON~~

~~P-channel E-MOSFET~~

~~+ V_{GS} : ON~~

~~- V_{GS} : OFF~~

Simplified Circuit Diagram of CMOS Inverter

When the input voltage V_{in} is at logic HIGH, the Gate-Source voltage (V_{G2S2}) of the P-channel E-MOSFET is 0 V. Hence, this MOSFET is in the OFF-state. The Gate-Source voltage (V_{G1S1}) of N-channel E-MOSFET is equal to the supply voltage (V_{ss}). Hence, it is switched ON, and offers a low resistance path. Therefore, the output voltage (V_{out}) is approximately 0 V; a logic LOW.

ANALOG AND DIGITAL ELECTRONICS

WAVE-SHAPING CIRCUITS

INTEGRATED CIRCUIT (IC) MULTIVIBRATORS:

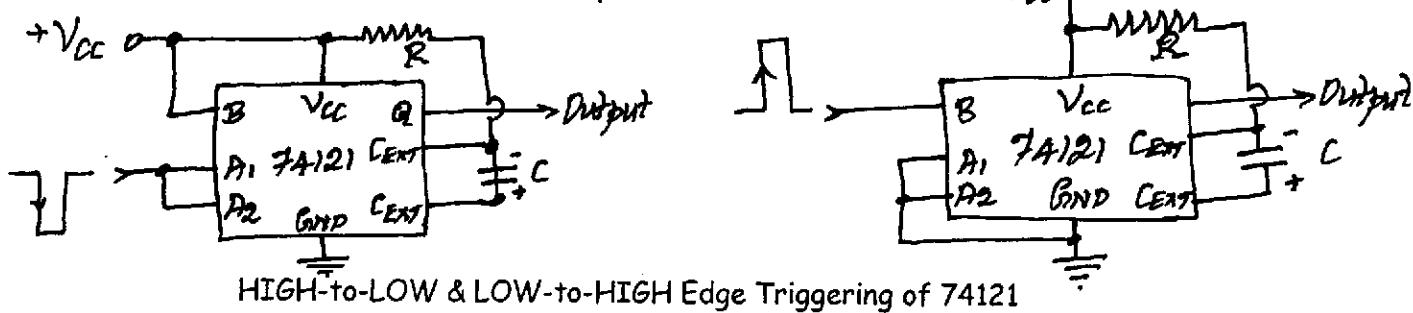
A multivibrator (like an oscillator) is a circuit with regenerative feedback, which produces a pulsed output. There are three basic types of multivibrator circuits:

- Astable – has no stable states, but switches continuously between two states. This action produces a train of square wave pulses at a fixed frequency.
- Monostable – one of the states is stable, but the other state is unstable (transient).
- Bistable – the circuit is stable in either state.

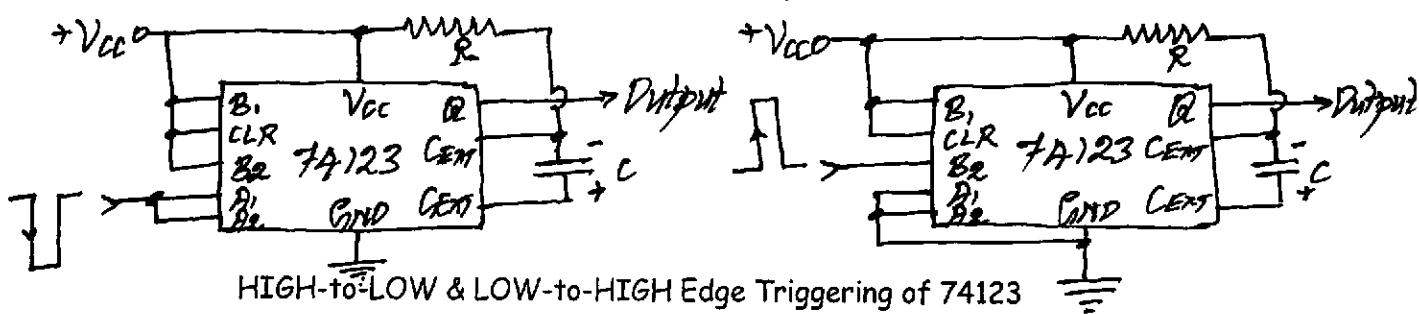
Digital IC-Based Monostable Multivibrators:

The ICs that can be used as monostable multivibrators include –

- TTL Family
 - 74121 – single monostable multivibrator: The IC provides features for triggering on either LOW-to-HIGH or HIGH-to-LOW edge trigger pulses. Output pulse width depends on external R and C, and can be computed from $T = 0.7RC$.



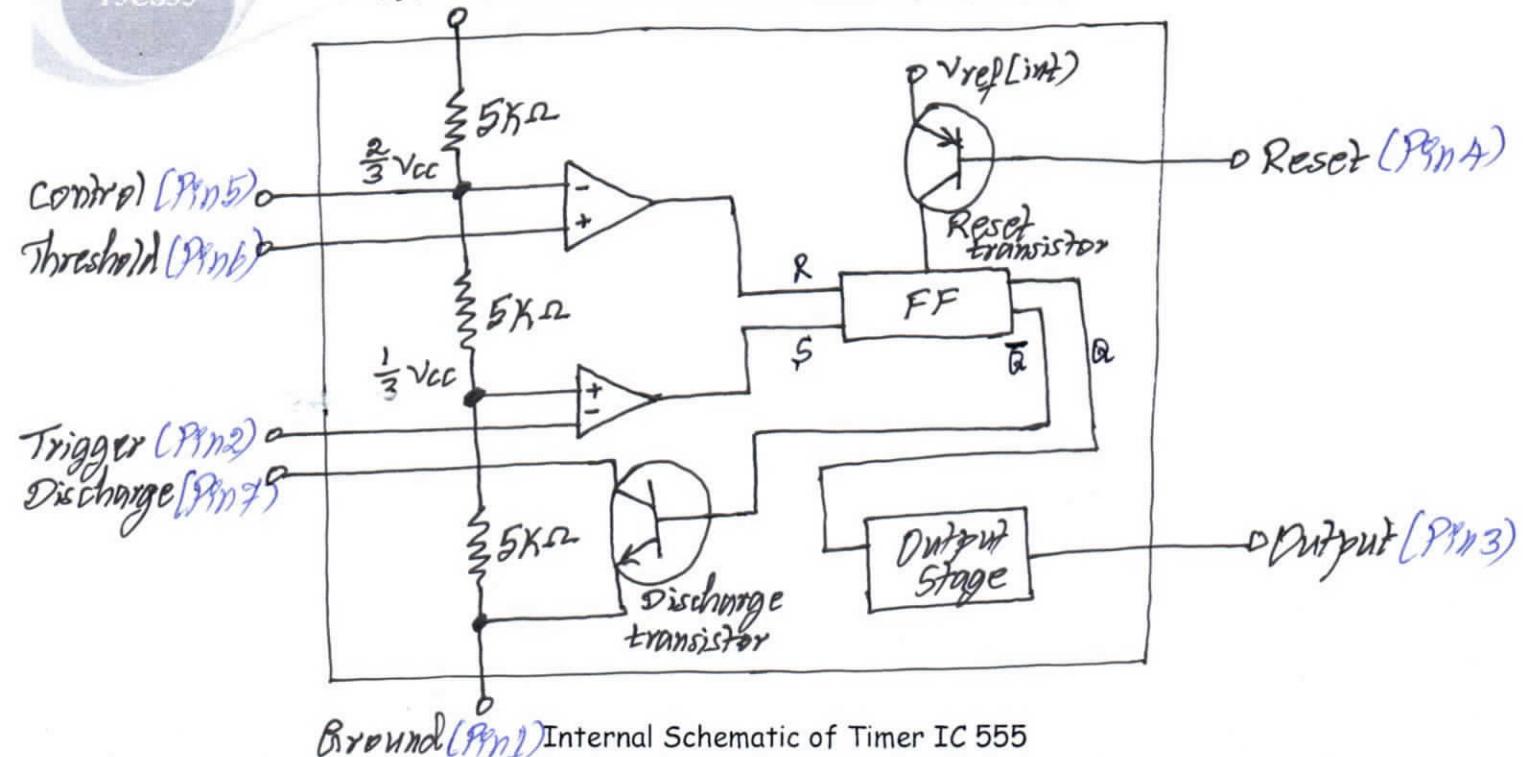
- 74122 – single retriggerable monostable multivibrator
- 74123 – dual retriggerable monostable multivibrator: The IC provides features for triggering on either LOW-to-HIGH or HIGH-to-LOW edge trigger pulses. Output pulse width depends on external R and C, and can be computed from $T = 0.28RC * [1 + (0.7/RC)]$, where R and C are, respectively in kilo-ohms and pico-farads; and T is in nano-seconds.



- CMOS Family
 - 4098B – dual retriggerable monostable multivibrator

Timer IC-Based Multivibrators:

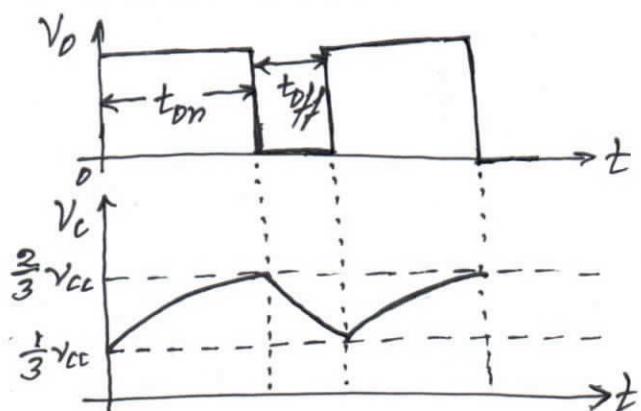
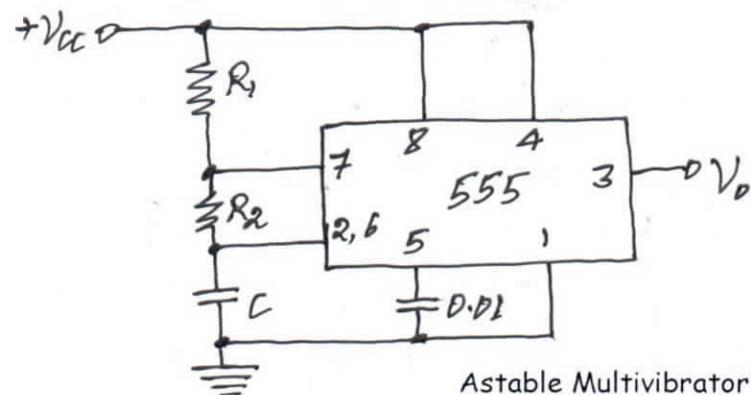
Timer IC is the one of the most commonly used general-purpose linear integrated circuits.



Ground (Pin 1) Internal Schematic of Timer IC 555

The Timer IC 555 comprises two Op-Amp comparators, a flip-flop, a discharge transistor, a reset transistor, three identical resistors and an output stage. The resistors set the reference voltage levels at the non-inverting input of the lower comparator and inverting input of the upper comparator at $\frac{1}{3}V_{CC}$ and $\frac{2}{3}V_{CC}$, respectively. The output of two comparators feed SET and RESET inputs of the Flip-Flop. This decided the logic state of its output and subsequently the final output. The Flip-Flops complementary outputs feed the output stage and the base of the discharge transistor. Hence, when the output is HIGH, the discharge transistor is OFF and when the output is LOW, the discharge transistor is ON.

Astable Multivibrator Using Timer IC 555:



Astable Multivibrator & Its Relevant Waveforms

The working is as follows:

- Initially, capacitor C is fully discharged, which forces output to go to HIGH-state.
- Now, the discharge transistor allows the capacitor C to charge from +V_{CC} through R₁ and R₂.

ANALOG AND DIGITAL ELECTRONICS

- When the voltage across C exceeds $+2V_{CC}/3$, the output goes to LOW-state; and the discharge transistor is switched ON.
- Hence, the capacitor C begins to discharge through R_2 and the discharge transistor.
- When the voltage across C falls below $+V_{CC}/3$, the output goes back to the HIGH-state.
- The charge and the discharge cycles repeat and the circuit behave like a multivibrator.

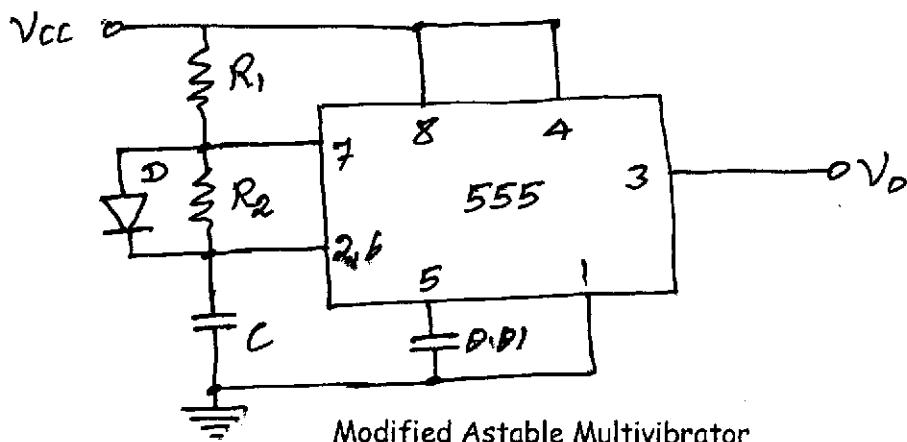
HIGH-state time period, $T_{HIGH} = 0.69 * (R_1 + R_2) * C$

LOW-state time period $T_{LOW} = 0.69 * R_2 * C$

Time period $T = 0.69 * (R_1 + 2R_2) * C$

Frequency $F = \frac{1}{0.69 * (R_1 + 2R_2) * C}$

In the above Fig, the HIGH-state time period is always greater than the LOW-state time period. The following Fig shows a modified circuit where HIGH-state and LOW-state time periods can be chosen independently.



HIGH-state time period, $T_{HIGH} = 0.69 * R_1 * C$

LOW-state time period $T_{LOW} = 0.69 * R_2 * C$

For $R_1 = R_2 = R$;

Time period $T = 1.38 * R * C$

Frequency $F = \frac{1}{1.38 * R * C}$

Monostable Multivibrator Using Timer IC 555:

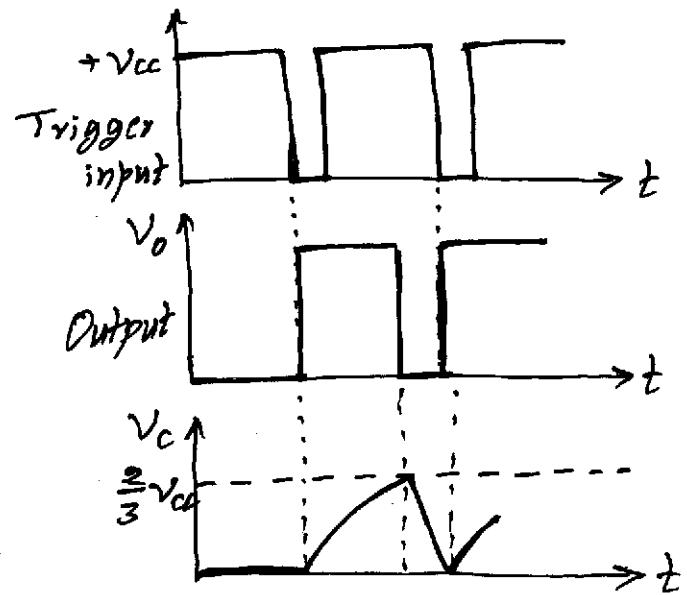
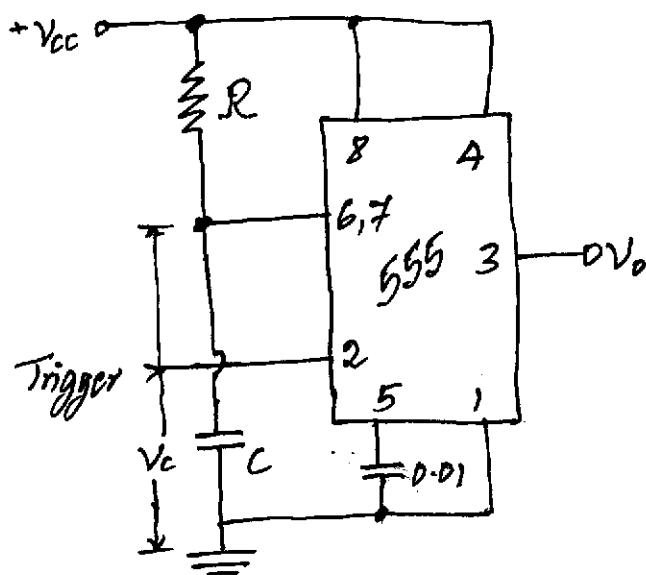
The monostable multivibrators are also referred to as monoshots. The working of the following monoshot shown in the Fig is as follows:

- Trigger pulse is applied to Terminal-2 of the IC, which should initially be kept at $+V_{CC}$.
- A HIGH at terminal-2 forces the output to LOW-state.
- A HIGH-to-LOW trigger pulse at terminal-2 holds the output in the HIGH-state and simultaneously allows the capacitor to charge from $+V_{CC}$ through R.



ANALOG AND DIGITAL ELECTRONICS

- When the capacitor voltage exceeds $+2V_{CC}/3$, the output goes back to the LOW-state.
- Once again, another trigger pulse will have to be applied to terminal-2 to make the output to go to HIGH-state again.



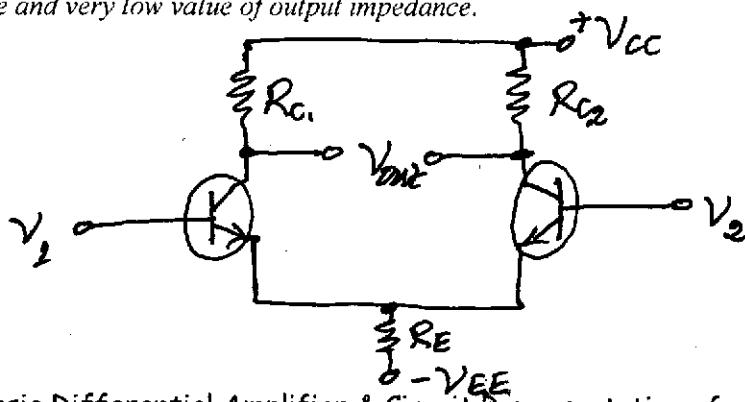
Monostable Multivibrator & Its Relevant Waveforms

Every time the timer is appropriately triggered, the output goes to HIGH-state, and stays there for a time period taken by capacitor to charge from 0 to $+2V_{CC}/3$. This time period, which equal the monoshot output pulse width, is given by; $T = 1.1 * R * C$

INTRODUCTION TO OPERATIONAL AMPLIFIERS

IDEAL OP-AMP versus PRACTICAL OP-AMP:

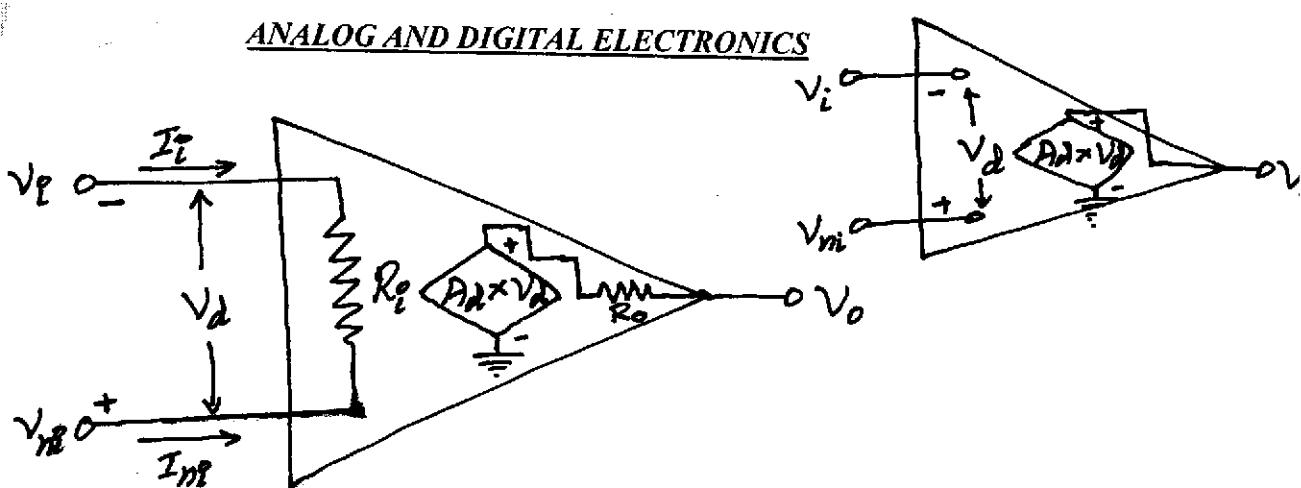
An *Op-Amp* is a direct-coupled high gain, high bandwidth differential amplifier with very high value of input impedance and very low value of output impedance.



Basic Differential Amplifier & Circuit Representation of an OP-Amp

The following Fig shows the Thevenin's Equivalent Model of an Ideal Op-Amp and a Practical Op-Amp.

ANALOG AND DIGITAL ELECTRONICS



Thevenin's Equivalent Model of an Ideal Op-Amp and a Practical Op-Amp

V_i and V_{ni} are, respectively, inverting and non-inverting inputs; and A_d is the open loop differential voltage gain. In a practical Op-Amp, there are loading effects at the input and output ports, due to finite values of input and output resistances.

The ideal Op-Amp model was derived to simplify circuit calculations. The ideal Op-Amp model makes three assumptions:

1. Input resistance (impedance), $R_i = \infty$
2. Output resistance (impedance), $R_o = 0$
3. Open-loop (differential voltage) gain, $A_d = \infty$

Based on these three assumptions, other assumptions can be derived:

1. Since $R_i = \infty$, $I_i = I_{ni} = 0$
2. Since $R_o = 0$, $V_o = A_d \cdot V_d$
3. Zero DC input and output offset voltages
4. Bandwidth and slew rate are also infinite, as no frequency dependencies are assumed.
5. Drift is also zero, as there is no changes in performance over time, temperature, power supply variations, and so on
6. Since output voltage depends only on differential input voltage, it rejects any voltage common to both inputs. Hence, common mode gain = 0

Open-loop gain is the differential voltage gain in the absence of any positive or negative feedback.

Practical Op-Amps have –

1. Input impedance can vary from hundred of kilo-ohms (for some low-grade Op-Amps) to tera-ohms (for high grade Op-Amps).
2. Output impedance may be in the range of 10 to 100 Ω
3. Open-loop gain in the range of 10,000 to 1, 00,000
4. Bandwidth is limited and is specified by gain-bandwidth product

ANALOG AND DIGITAL ELECTRONICS

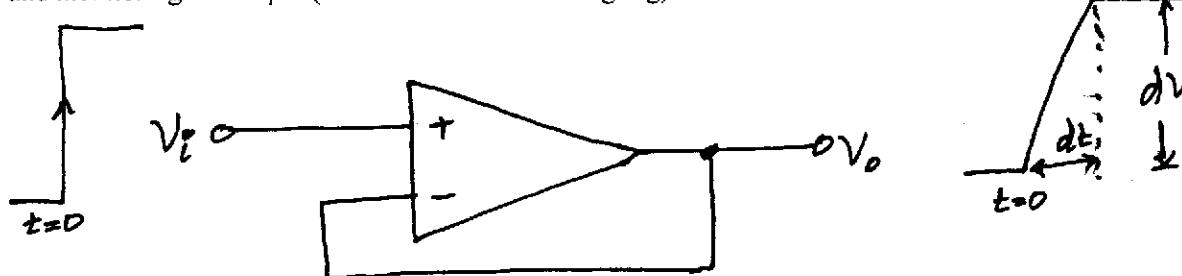
5. There may be some finite DC output (referred to as output offset voltage), even when both the inputs are grounded.

PERFORMANCE PARAMETERS:

The key parameters of an Op-Amp decide its suitability for a particular application: Key parameters of Op-Amp include the following:

Bandwidth: of an Op-Amp is the range of frequencies that it can amplify for a given amplifier gain. The bandwidth is usually expressed in terms of the unity gain crossover frequency (also called gain-bandwidth product). It is 1 MHz in the case of Op-Amp 741. It could be as high as 1500 MHz in the case of high bandwidth Op-Amps.

Slew Rate: is the rate of change of output voltage with time. It gives us an idea how well the Op-Amp output voltage follows a rapidly changing waveform at the input. It is determined by applying a step input and monitoring the output (as shown in the following Fig).

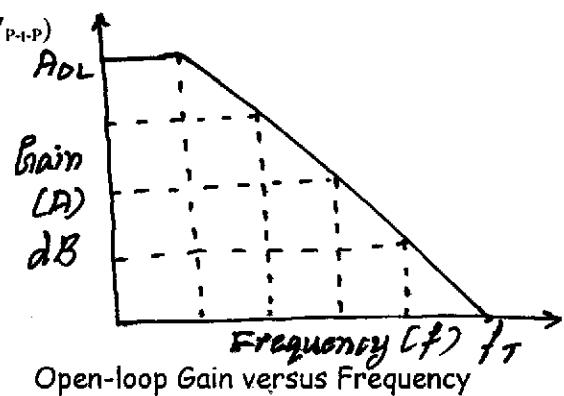


Response of Op-Amp to Step Input

The step input simulates the large signal conditions. The incapability of the Op-Amp to follow rapidly rising and falling input is due to the minimum charge and discharge times required by an internally connected capacitor across the output; i.e., slew rate limits the large signal bandwidth. Hence, the rate of change of output should be less than the slew rate of the Op-Amp. Peak-to-peak output voltage swing for a sinusoidal signal (V_{P-P}), slew rate and bandwidth are interrelated by the following equation:

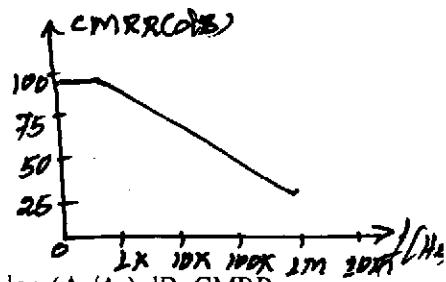
$$\text{Bandwidth (highest frequency, } f_{\text{MAX}} \text{)} = \text{Slew rate} / (\pi * V_{P-P})$$

Open-Loop Gain: is the ratio of single-ended output to the differential input. The (closed) loop gain depends upon the (application) circuit. The following Fig shows the open-loop gain versus frequency curve of an Op-Amp. The gain error at any given frequency is given by the ratio of the closed-loop gain to the open-loop gain.



Common Mode Rejection Ratio (CMRR): the ratio of the desired differential gain (A_d) to the undesired common mode gain (A_c). It is a measure of the ability of the Op-Amp to suppress common mode signals.

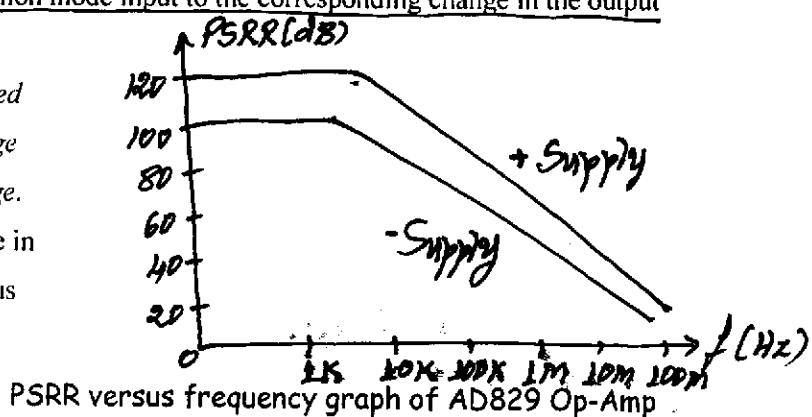
ANALOG AND DIGITAL ELECTRONICS



The ratio of CMRR (a DC parameter) is usually expressed as CMR, given by $20 \log (A_d/A_c)$ dB. CMRR is also defined as the ratio of change in the common mode input to the corresponding change in the output offset voltage.

Power Supply Rejection Ratio (PSRR): is defined as the ratio of change in the power supply voltage to the corresponding change in the output voltage.

PSRR (a DC parameter) value falls with increase in frequency. The following Fig shows PSRR versus frequency graph of AD829 Op-Amp.



PSRR versus frequency graph of AD829 Op-Amp

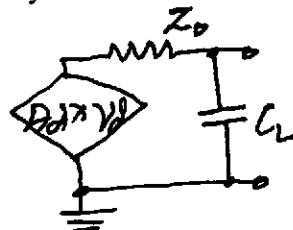
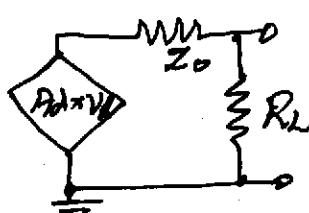
Input Impedance: is the impedance looking into the input terminals of the Op-Amp and is expressed in terms of resistance. The effective input impedance will be different from that specified in the data sheets, when the Op-Amp is used in the closed-loop mode. In an inverting amplifier, the effective input impedance equals the input resistance connected externally from the source of input signal to the inverting input terminal of the Op-Amp. In the non-inverting amplifier, it equals the product of the loop gain and the specified Op-Amp input impedance.

Output Impedance: is defined as the impedance between the output terminal of the Op-Amp and ground. Output impedance becomes a critical parameter, when using output of Op-Amps to drive heavy loads. The expression for the output in the case of –

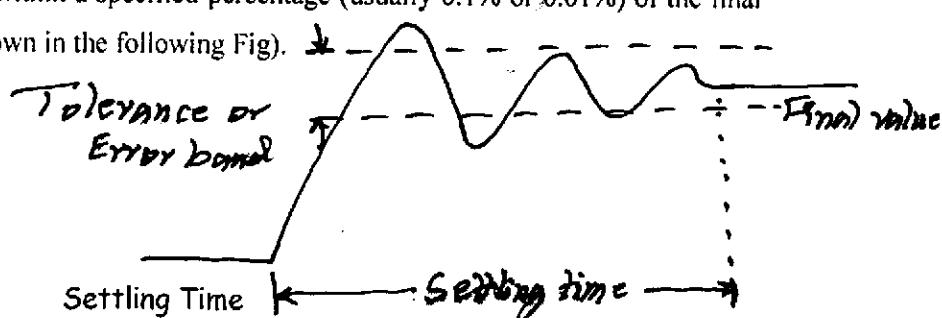
$$\text{Resistive load: } V_o = A_d * V_d \left[\frac{R_L}{R_L + Z_o} \right]$$

$$\text{Capacitive load: } V_o = A_d * V_d \left[\frac{1}{j(f/f_o) + 1} \right]$$

where, $f_o = 1/2\pi Z_o C_L$ and Z_o is the output impedance of the Op-Amp.



Settling Time: gives the response of the Op-Amp to large step signals. It is a parameter specified in the case of high-speed Op-Amps or Op-Amps with a high value of gain-bandwidth product. It is expressed as time taken by the Op-Amp output to settle within a specified percentage (usually 0.1% or 0.01%) of the final value in response to a step input (as shown in the following Fig).



Offsets and Offset Drifts: An ideal Op-Amp should produce a zero output for a zero differential input. But, it is not so in the case of real Op-Amps. It is observed that, we need to apply DC differential voltage externally to get a zero output. This externally applied input is referred to as *input offset voltage*. *Output*

ANALOG AND DIGITAL ELECTRONICS

offset voltage is the voltage at the output with both the input terminals grounded. The difference between the two bias currents flowing towards the inputs of the Op-Amp is referred to as *input offset current*.

Problem: Op-Amp LM 741 is specified to have slew rate of $0.5 \text{ V}/\mu\text{s}$. If the Op-Amp were used as amplifier and the expected peak output voltage were $10V$, determine the highest sinusoidal frequency that would get satisfactorily amplified.

Solution: Given, Slew rate = $0.5 \text{ V}/\mu\text{s}$ & $V_p = 10V$.

$$\begin{aligned}\text{The highest sinusoidal freq. } f_{\max} &= \text{Slew rate} / (\pi * V_p) \\ &= (0.5 * 10^6) / (\pi * 2 * 10) \\ &= \underline{\underline{7.96 \text{ kHz}}}.\end{aligned}$$

Problem: The differential voltage gain and CMRR of an Op-Amp when expressed in decibels are 110 dB and 100 dB , respectively. Determine the common mode gain expressed as a ratio.

$$\text{CMRR} = 20 \log (A_d / A_{cm}) = 20 \log A_d - 20 \log A_{cm}$$

$$\text{i.e., } 20 \log A_{cm} = 20 \log A_d - \text{CMRR} = 110 - 100 = 10 \text{ dB.}$$

$$\therefore \log A_{cm} = \frac{10}{20} = 0.5$$

$$\Rightarrow A_{cm} = \text{Antilog } 0.5 = \underline{\underline{3.16}}$$

Problem: In the case of certain Op-Amp, 0.5 V change in the common mode input causes a DC output offset change of $5 \mu\text{V}$. Determine CMRR in dB.

Solution:

$$\text{CMRR} = \Delta V_{cm} / \Delta V_{os} = 0.5 / (5 * 10^{-6}) = 10^5.$$

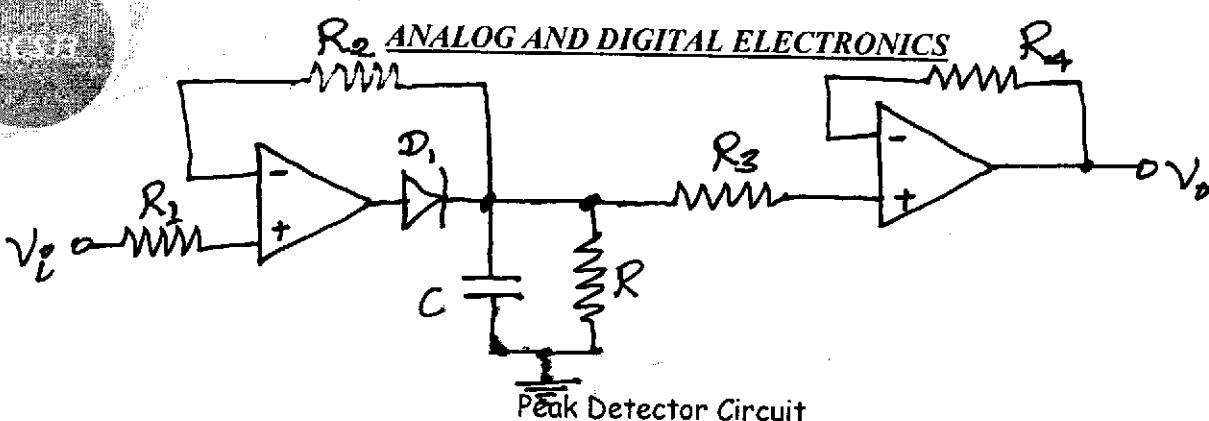
$$\text{CMRR in dB} = 20 \log 10^5 = \underline{\underline{100 \text{ dB}}}.$$

OPERATIONAL AMPLIFIER APPLICATION CIRCUITS

PEAK DETECTOR CIRCUIT:

Peak detector circuit produces a voltage at the output equal to the peak amplitude (positive or negative) of the input signal.

MAHESH PRASANNA K., VCET, PUTTUR

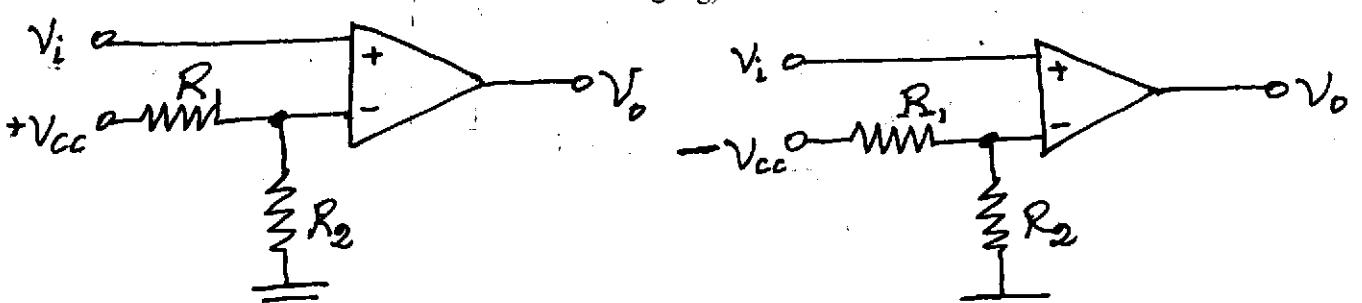


During the positive half cycles of input, the diode D_1 will be forward biased. The capacitor C rapidly charges to the peak, from the output of the Op-Amp. As the input starts decreasing beyond the peak, the diode gets reverse-biased; and hence, capacitor will be isolated from the output of the Op-Amp. Now, the capacitor starts to discharge through resistance R connected across it. The value of this resistor will be much larger to allow a discharge path. The buffer circuit connected ahead of the capacitor prevents any discharge of the capacitor due to loading effects. This circuit can be made to respond to the negative peaks by reversing the polarity of the diode.

COMPARATOR:

A comparator circuit is a two-input, one-output building block that produces a high or low output depending upon the relative magnitudes of the two inputs. An Op-Amp produces (without feedback) either positively saturated or negatively saturated output voltage depending upon whether the amplitude of the voltage applied at the non-inverting input terminal is more or less positive than the voltage applied at the inverting input terminal.

One of the inputs of the comparator is applied with a reference voltage and the other input is fed with the input voltage that needs to be compared with the reference voltage. The reference voltage may be a positive or negative voltage (as shown in the following Fig.).



Non-inverting Comparator with Positive Reference & Negative Reference

V_{REF} for non-inverting comparator with positive reference is given by $+V_{cc} * [R_2 / (R_1 + R_2)]$. $[V_{in} > V_{REF}]$

V_{REF} for non-inverting comparator with negative reference is given by $-V_{cc} * [R_2 / (R_1 + R_2)]$. $[V_{in} < V_{REF}]$

Inverting-type voltage comparators can similarly be built for positive and negative reference voltages.

Summary: In comparator –

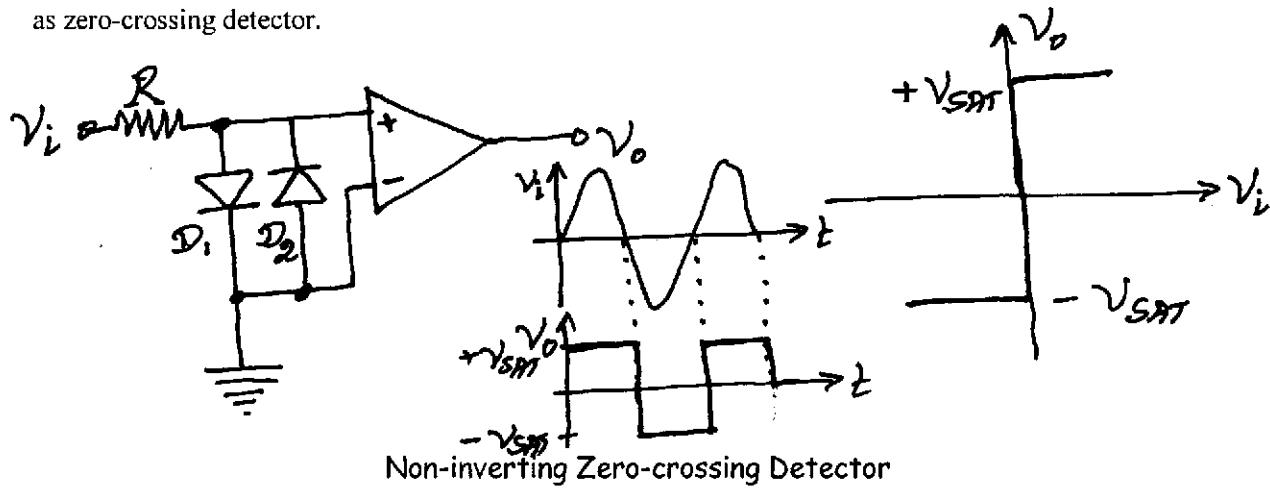
$$V_{out} = \begin{cases} +V_{cc} & V_{in} > V_{REF} \\ -V_{cc} & V_{in} < V_{REF} \end{cases}$$

$$V_{out} = -V_{in}$$

ANALOG AND DIGITAL ELECTRONICS

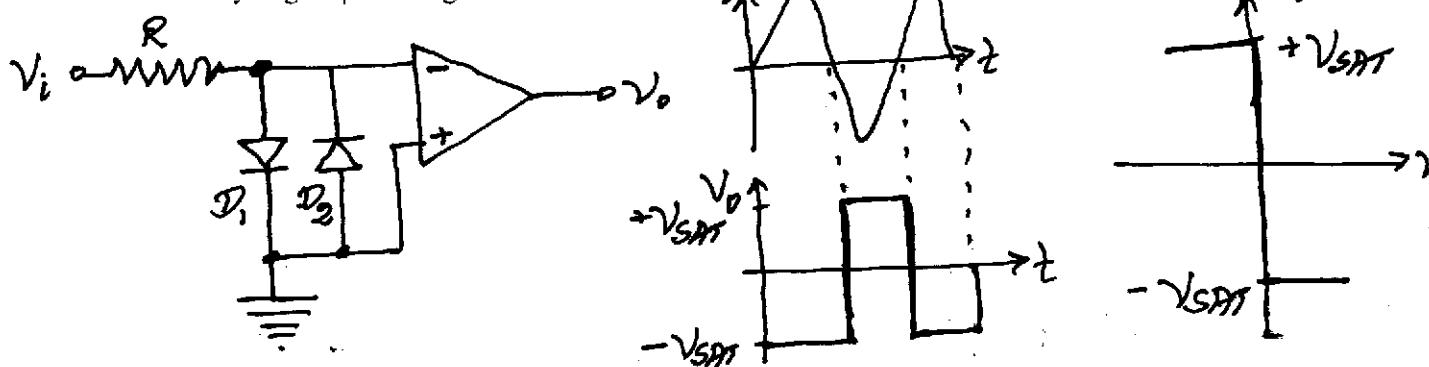
- ✓ If the input voltage is less than the reference voltage, then the output voltage will be negatively saturated.
- ✓ If the input voltage is equal to the reference voltage, then the output voltage will be zero.
- ✓ If the input voltage is greater than the reference voltage, then the output voltage will be ~~positively~~
~~negatively~~ saturated.

Zero-Crossing Detector: A special case of comparator, where the reference voltage is zero, is referred to as zero-crossing detector.



Non-inverting Zero-crossing Detector

Here, an input more positive than zero leads to a positively saturated output (as shown in the waveform above). The diodes connected at the input are to protect the sensitive input circuits inside the Op-Amp from excessively large input voltages.



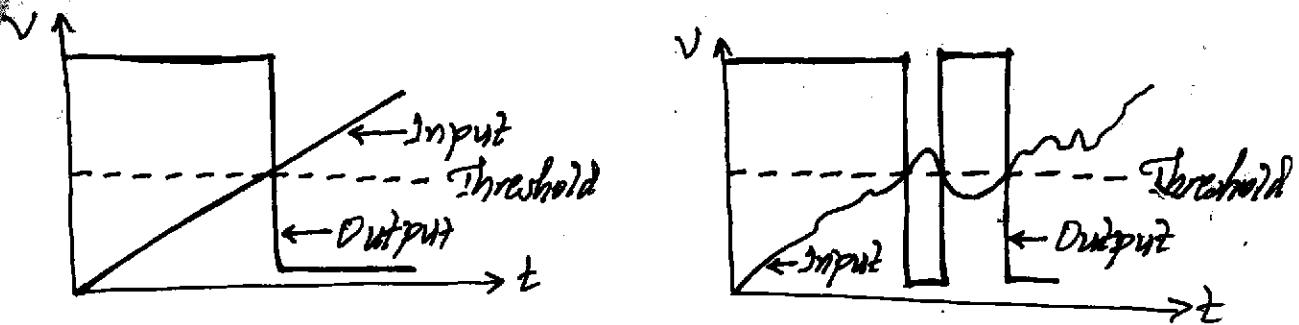
Inverting Zero-crossing Detector

Here, an input more positive than zero leads to a negatively saturated output (as shown in the waveform above).

Application of zero-crossing detector is to convert sine wave signal to a square wave signal.

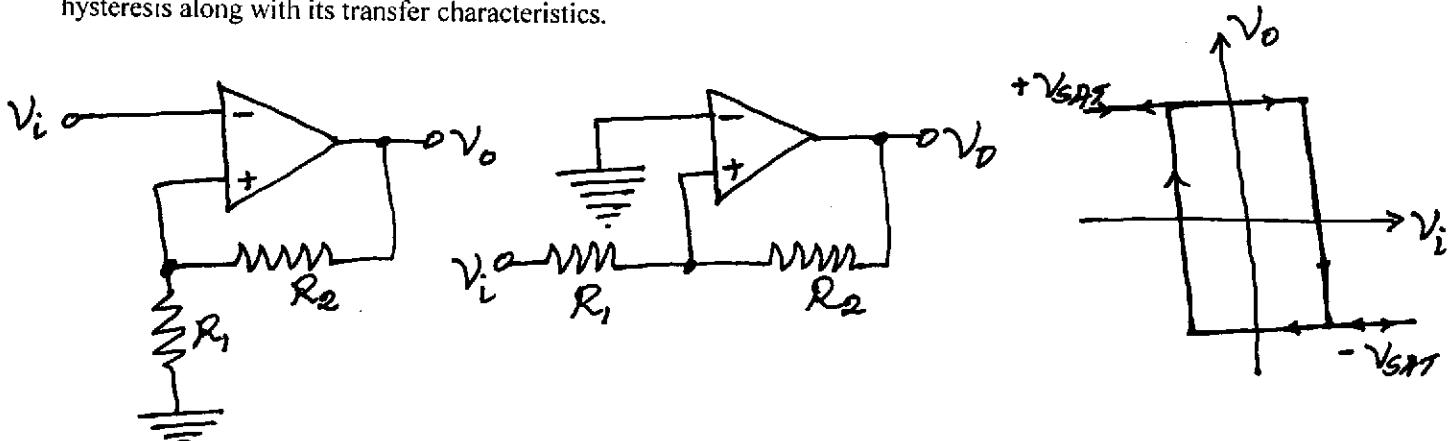
Comparator with Hysteresis: When the input signal applied to the comparator contains noise, transitions at the output around the trip point tend to become highly erratic (as shown below).

ANALOG AND DIGITAL ELECTRONICS



Transitions Caused by Ideal Input Signal & Noisy Input Signal

The following Fig shows the circuit schematic of a non-inverting and inverting comparators with hysteresis along with its transfer characteristics.



Inverting Comparator & Non-inverting Comparator with Hysteresis

Assume that, the output is in positive saturation ($+V_{SAT}$). Voltage at non-inverting input is $+V_{SAT} * R_1 / (R_1 + R_2)$. Due to this small positive voltage at the non-inverting input, the output is reinforced to stay in positive saturation.

Now, the input signal needs to be more positive than this voltage for the output to go to negative saturation. Once the output goes to the negative saturation ($-V_{SAT}$), voltage feedback to the non-inverting input becomes $-V_{SAT} * R_1 / (R_1 + R_2)$. Due to this small negative voltage at the non-inverting input, the output is reinforced to stay in negative saturation.

In this manner, the circuit offers a hysteresis of $2V_{SAT} * R_1 / (R_1 + R_2)$.

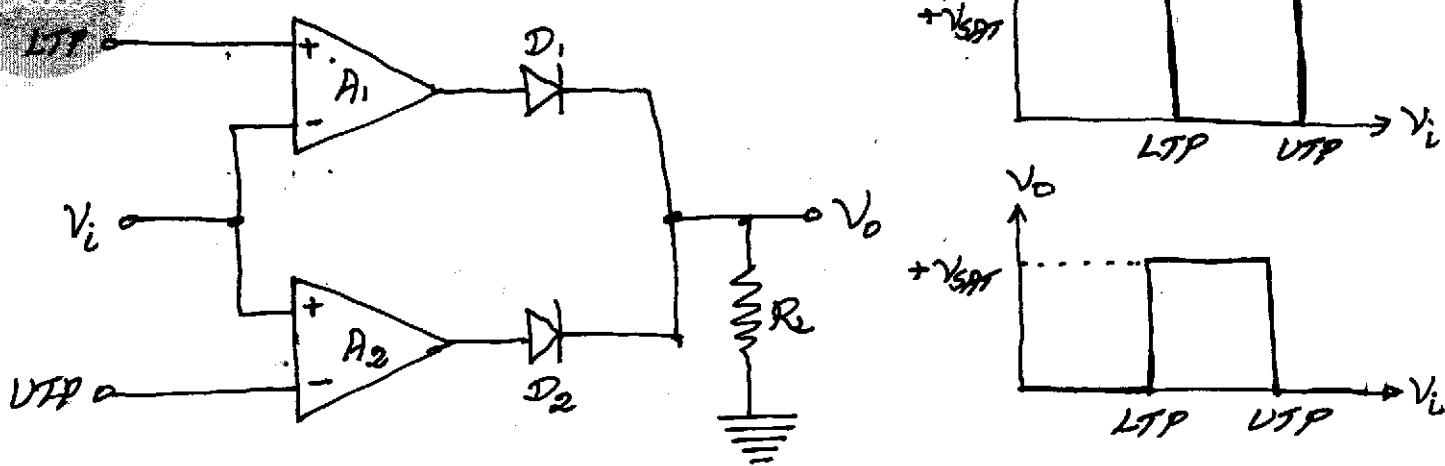
Non-inverting comparator with hysteresis can explained in the similar manner. In this case, the upper and the lower trip points (UTP & LTP) are $+V_{SAT} * R_1 / R_2$ and $-V_{SAT} * R_1 / R_2$. Hysteresis is equal to $2V_{SAT} * R_1 / R_2$.

Window Comparator: In window comparator, the output changes state when the input voltage goes above or below the reset reference voltage. In a window comparator, there are two reference voltages, called lower and upper trip points. Output is in one state, when it is inside the window created by the lower and the upper trip points and in the other state when it is outside the window.

Non-Inverting: $V_{in} \geq UTP$; $V_{out} = -V_{SAT}$; $D_1 \rightarrow FB$; $V_{out2} = +V_{SAT}$; $V_o = +V_{SAT}$

Inverting: $V_{in} < LTP$; $V_{out1} = +V_{SAT}$; $D_2 \rightarrow RB$; $V_o = -V_{SAT}$

ANALOG AND DIGITAL ELECTRONICS



Window Comparator & Its Transfer Characteristics

When the input voltage is less than the voltage reference corresponding to the LTP, output of the Op-Amp A₁ is at $+V_{SAT}$ and that of Op-Amp A₂ is $-V_{SAT}$. Diodes D₁ and D₂ are respectively, forward- and reverse-biased. Hence, output across R_L is $+V_{SAT}$.

When the input voltage is greater than the voltage reference corresponding to the UTP, output of the Op-Amp A₁ is at $-V_{SAT}$ and that of Op-Amp A₂ is $+V_{SAT}$. Diodes D₁ and D₂ are respectively, reverse- and forward-biased. Hence, output across R_L is again $+V_{SAT}$.

When the input voltage is greater than LTP and less than UTP, the output of both Op-Amps is at $-V_{SAT}$. Hence, both the diodes are reverse-biased and output across R_L is zero.

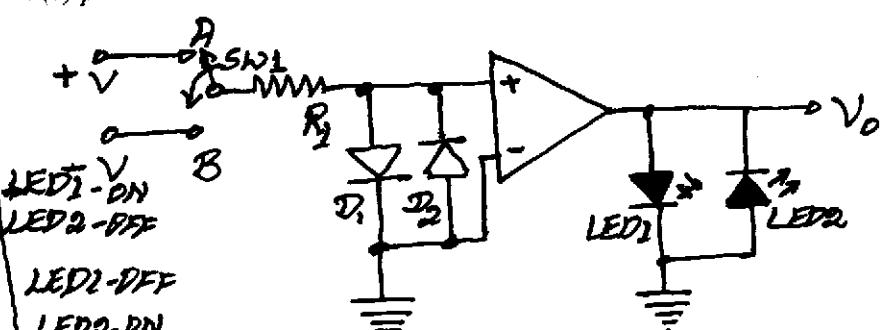
The first waveform shown in the above Fig shows the transfer characteristics of window comparator shown. The second waveform shown in the above Fig shows the transfer characteristics of window comparator, if we interchange the positions of LTPs and UTPs.

Problem: Refer to the comparator circuit given below. Determine the state of LED1 and LED2, when the switch SW1 is in (a) position A and (b) position B.

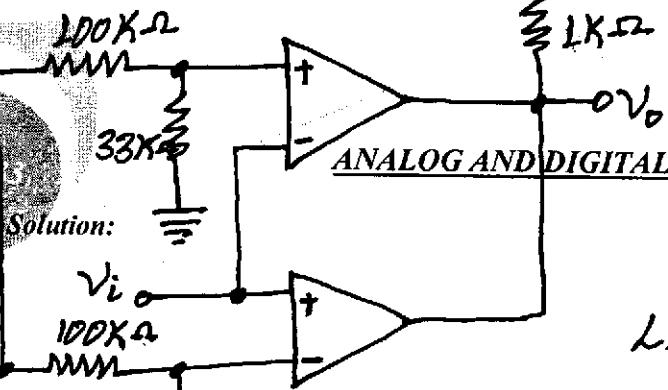
Solution:

SW1	V_{in}	V_o
A	$+0.7V$	$+V_{SAT}$
B	$-0.7V$	$-V_{SAT}$

LED1 - ON
LED2 - OFF
LED1 - OFF
LED2 - ON

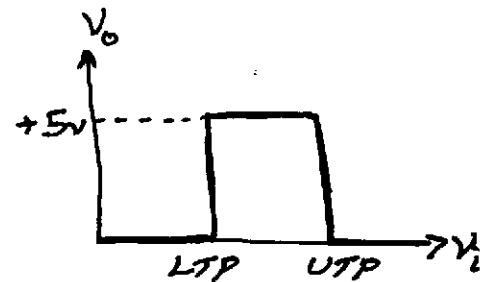


Problem: The following Fig shows a non-inverting type of window comparator. Determine the lower and upper trip points of the comparator and also draw the output voltage V_o versus input voltage V , transfer characteristics.



Solution:

ANALOG AND DIGITAL ELECTRONICS

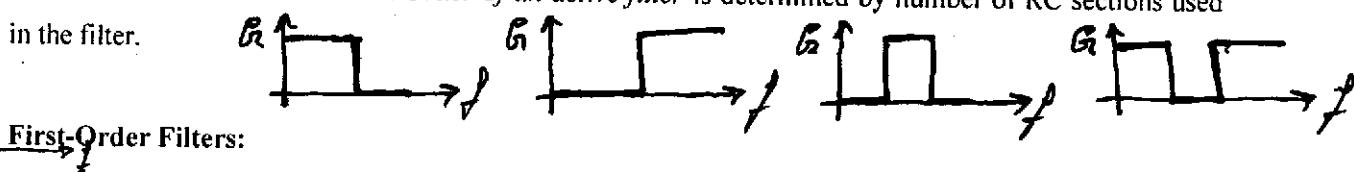


$$LTP = \frac{12 \times 15 \times 10^3}{(100+15) \times 10^3} = 1.565V.$$

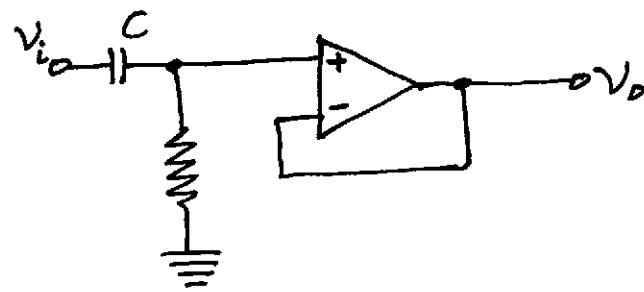
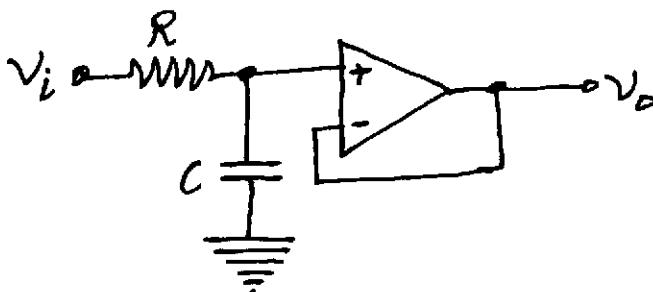
$$UTP = \frac{12 \times 33 \times 10^3}{(100+33) \times 10^3} = 2.977V.$$

ACTIVE FILTERS:

Here four types of active filters, viz., low-pass, high-pass, band-pass, and band-reject filters of first-order and second-order will be studied. Order of an active filter is determined by number of RC sections used in the filter.



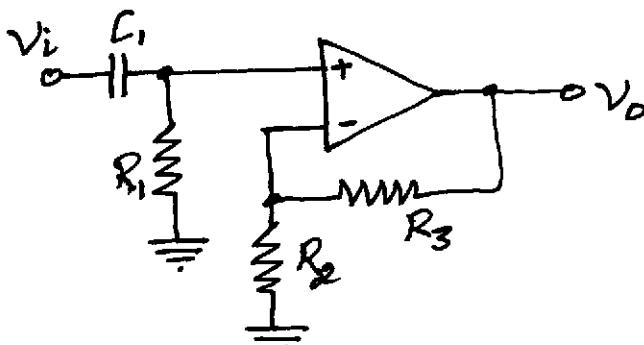
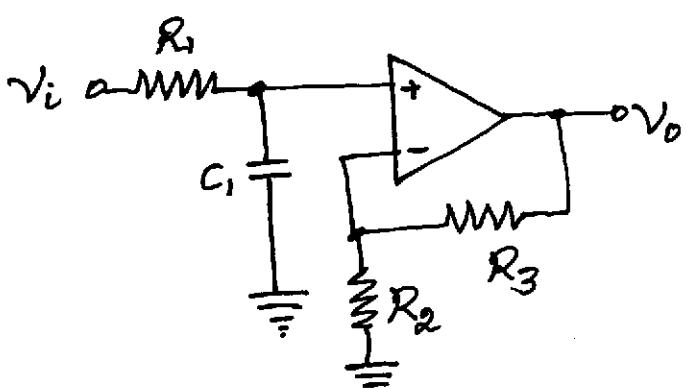
First-Order Filters:



First-order Active Filters (Non-inverting): Low-pass & High-pass

In the case of low-pass circuit; at low frequencies, reactance offered by the capacitor is much larger than the resistance value. Hence, the applied input signal appears at the output un-attenuated. At high frequencies, the capacitive reactance becomes much smaller than the resistance value. Hence, the output will be zero. When the signal frequency is such that, the capacitive reactance is equal to resistance value, the output is 0.707 times the input. This is called upper cut-off frequency.

$$f_C = \frac{1}{2\pi RC}$$



First-order Filters with Gain (Non-inverting): Low-pass & High-pass

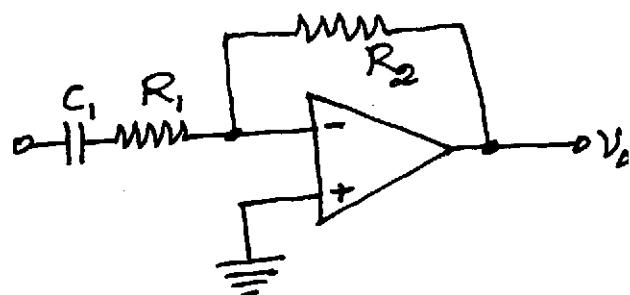
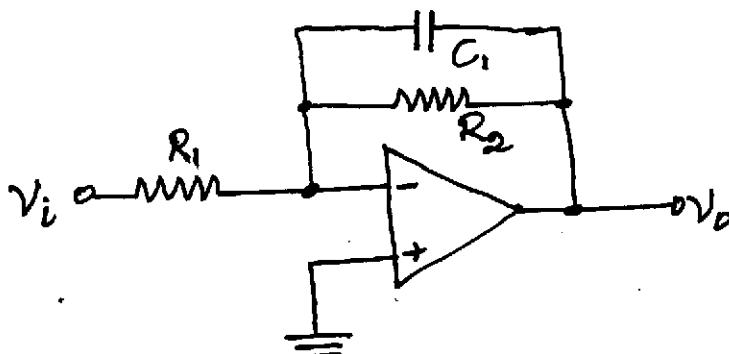


ANALOG AND DIGITAL ELECTRONICS

If an active filter with desired amplification is required; the above circuits may be modified as given in the ~~above~~ Fig. The voltage gain is given by;

$$A_v = 1 + \frac{R_3}{R_2}$$

These filters could also be implemented using inverting amplifier configuration.



First-order Active Filters (Inverting): Low-pass & High-pass

Cut-off frequency and mid-band gain values in the case of low-pass filter are; $f_c = \frac{1}{2\pi R_2 C_1}$

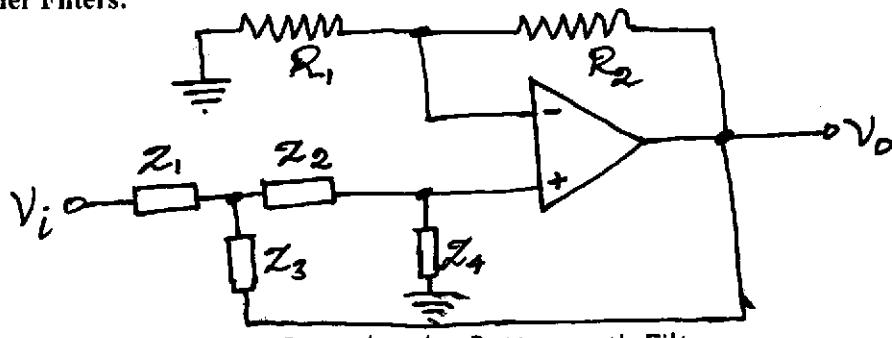
$$A_v = -\frac{R_2}{R_1}$$

Cut-off frequency and mid-band gain values in the case of high-pass filter are; $f_c = \frac{1}{2\pi R_1 C_1}$

$$A_v = -\frac{R_2}{R_1}$$

Second

First-Order Filters:



Second-order Butterworth Filter

Butterworth filters, also called maximally flat filters, offers a relatively flat pass and stop band response.

This has the disadvantage of relatively sluggish roll-off.

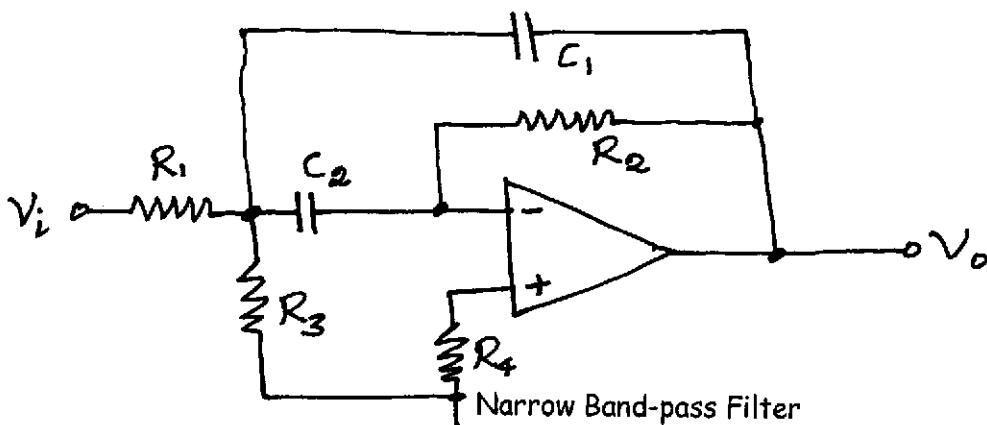
In the above Fig –

- ✓ If $Z_1 = Z_2 = R$ and $Z_3 = Z_4 = C$, we get a second-order low-pass filter
- ✓ If $Z_1 = Z_2 = C$ and $Z_3 = Z_4 = R$, we get a second-order high-pass filter

The cut-off frequency and pass-band gain values are given by; $f_c = \frac{1}{2\pi RC}$ $A_v = 1 + \frac{R_2}{R_1}$

ANALOG AND DIGITAL ELECTRONICS

A narrow band-pass filter required multiple feedbacks, as shown in the following Fig.



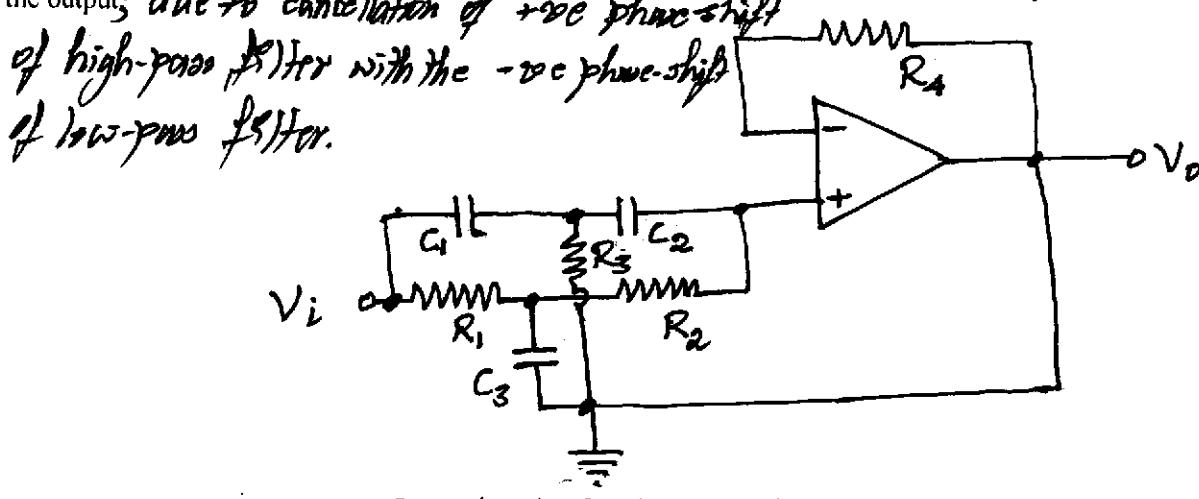
At very low frequencies, C_1 and C_2 offer very high reactance. As a result, the input signal is prevented from reaching the output. At very high frequencies, output is shorted to the inverting input. Hence, again there is no output. At some intermediate frequencies, the gain provided by the circuit offsets the loss due to the potential divider $R_1 - R_3$.

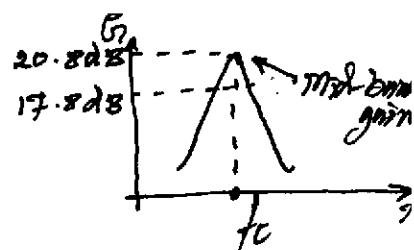
The resonant frequency is given by; $f_R = \frac{2Q}{2\pi R_2 C}$ where, Q is the quality factor.

For $C_1 = C_2 = C$, the quality factor is given by; $Q = [R_1 R_2 / 2R_3]^{1/2}$

The voltage gain is given by; $A_v = \frac{Q}{2\pi R_1 f_R C}$

A second-order narrow band-reject filter uses a twin-T network as shown in the following Fig. A twin-T network offers very high reactance at the resonance frequency and very low reactance at frequency off-resonance. In the circuit diagram, very low frequency signals find their way to the output via low-pass filter (formed by $R_1 - R_2 - C_3$); and very high frequency signals find their way to the output via high-pass filter (formed by $C_1 - C_2 - R_3$). Hence, in an intermediate band of frequencies, both filters pass signals to the output; due to cancellation of +ve phase shift of high-pass filter with the -ve phase-shift of low-pass filter.





ANALOG AND DIGITAL ELECTRONICS

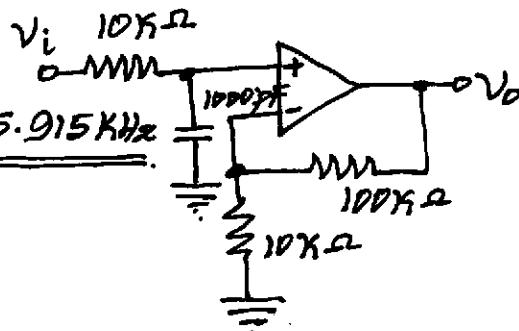
If $R_1 = R_2 = R$, $R_3 = R/2$ and $C_1 = C_2 = C$, $C_3 = 2C$, with $0 \leq R_4 \leq (R_1 + R_2)$ then; $f_c = \frac{1}{2\pi RC}$

Problem: Refer to the first-order low-pass filter of following Fig. Determine the cut-off frequency and the gain value at four times the cut-off frequency.

Solution:

$$\textcircled{1} f_c = \frac{1}{2\pi \times 10 \times 10^3 \times 1000 \times 10^{-12}} = \frac{10^5}{2\pi} \text{ Hz} = 15.915 \text{ kHz}$$

$$\textcircled{2} A_g = 1 + \frac{100 \times 10^3}{10 \times 10^3} = 11 = 20.827 \text{ dB.}$$



- Gain at cut-off frequency; $= 20.827 - 3 = 17.827 \text{ dB.}$
- Gain at frequency, four times the cut-off frequency $10^5 \times 3 \times 4 = 12 \text{ dB}$ below the value of mid-band gain. Therefore, gain at four times the cut-off frequency $= 20.827 - 12 = 8.827 \text{ dB.}$

Problem: The following Fig shows a second-order low-pass filter. Calculate the values of R_1 , R_2 , C_1 , C_2 and R_3 , if the cut-off frequency of the filter is 10 KHz, Q-factor is 0.707 and input impedance not less than $10 \text{ k}\Omega$.

Solution:

$$\textcircled{1} \text{ For } R_1 = R_2 = R; f_c = \frac{1}{2\pi R \sqrt{C_1 C_2}}$$

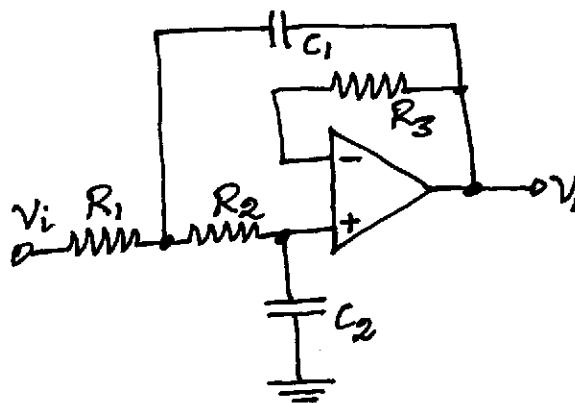
$$\textcircled{2} Q = \frac{1}{2} \sqrt{\frac{C_1}{C_2}}$$

$$\textcircled{3} \text{ For } Q = 0.707; C_1 = 2C_2$$

$$\textcircled{4} \text{ For input impedance of } 10 \text{ k}\Omega; R_1 = 10 \text{ k}\Omega = R_2$$

$$\textcircled{5} f_c = \frac{1}{2\pi \times 10 \times 10^3 \times C_2 \sqrt{2}} \Rightarrow C_2 = 0.0011 \mu\text{F} \text{ & } C_1 = 0.0022 \mu\text{F}$$

$$\textcircled{6} \text{ To have equal resistance, between Op-Amp inputs & ground; } R_3 = R_1 + R_2 = 20 \text{ k}\Omega.$$



ANALOG AND DIGITAL ELECTRONICS

Problem: Design an Op-Amp based twin-T band-reject filter having a notch frequency of 100 KHz. Specify the small-signal bandwidth of the Op-Amp if the highest expected frequency were 1 MHz.

Solution:

Notch frequency $f_R = \frac{1}{2\pi RC}$

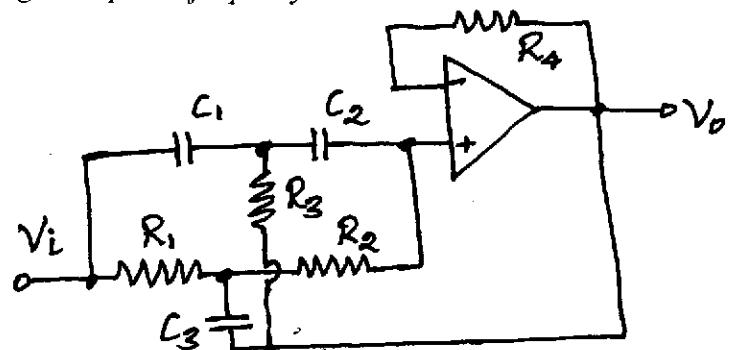
where; $R_1 = R_2 = R$; $C_1 = C_2 = C$;
 $R_3 = R/2$; $C_3 = 2C$.

Let $C = 0.0001 \mu F$. This gives; $R = \frac{1}{2\pi \times 0.0001 \times 10^{-6} \times 10^5 \times 10^3} = 15.92 \times$

Hence; $R_1 = R_2 = 15.92 \times \Omega$ $R_3 = 7.96 \times \Omega$ $C_1 = C_2 = 0.0001 \mu F$

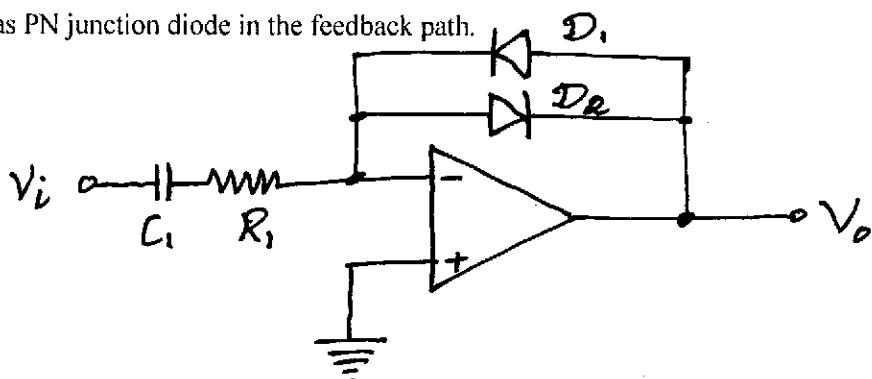
$C_3 = 0.0002 \mu F$

$\therefore R_4 = R_1 + R_2 = 31.84 \times \Omega$.



NON-LINEAR AMPLIFIER:

In a non-linear amplifier, the gain value is a non-linear function of the amplitude of the signal applied at the input. For example, the gain may be very large for weak input signals and may be very small for large input signals. A simple method to achieve non-linear amplification is by connecting a non-linear device such as PN junction diode in the feedback path.



Non-linear Amplifier

high

For small values of input signal, diodes act as open circuit and the gain is due to minimum feedback. When the amplitude of the input signal is large, diodes offer very small resistance and thus gain is low.

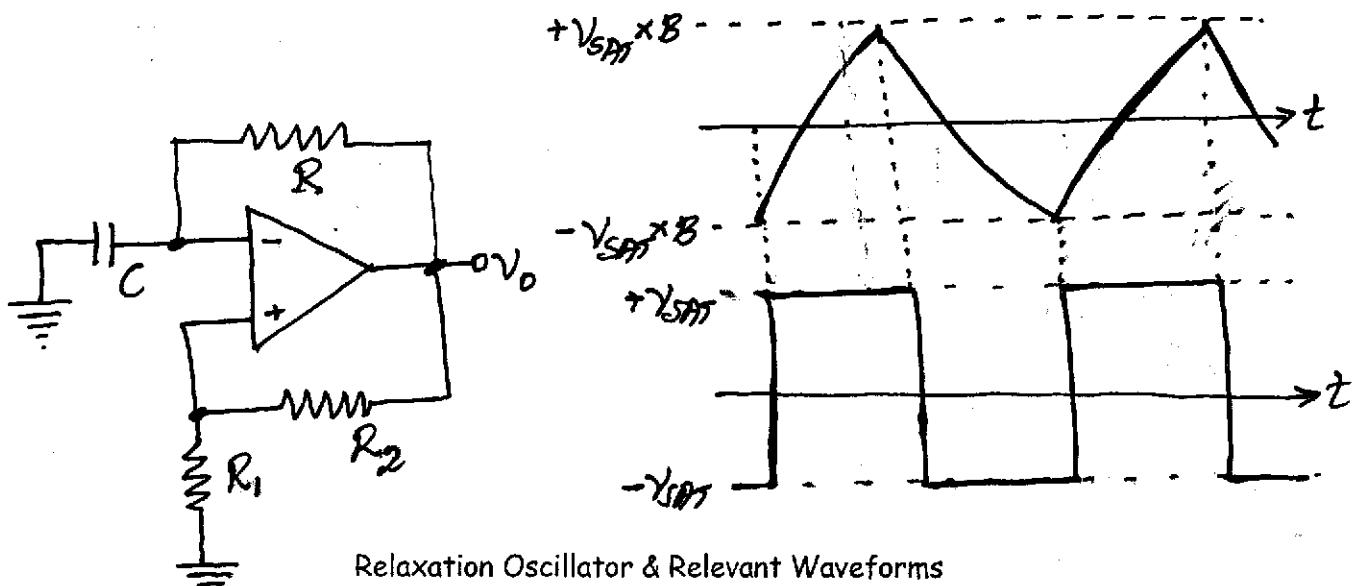


ANALOG AND DIGITAL ELECTRONICS

Application of non-linear amplifier is in AC bridge balance detectors.

RELAXATION OSCILLATOR:

Relaxation oscillator is an oscillator circuit that produces a non-sinusoidal output whose time period is dependent on the charging time of a capacitor connected as a part of the oscillator circuit.



Relaxation Oscillator & Relevant Waveforms

Assume that, the output is initially in positive saturation. As a result, voltage at non-inverting input of Op-Amp is $+V_{SAT} * R_1 / (R_1 + R_2)$. This forces the output to stay in positive saturation as the capacitor C is initially in fully discharged state. Capacitor C starts charging towards $+V_{SAT}$ through R. The moment the capacitor voltage exceeds the voltage appearing at the non-inverting input, the output switches to $-V_{SAT}$.

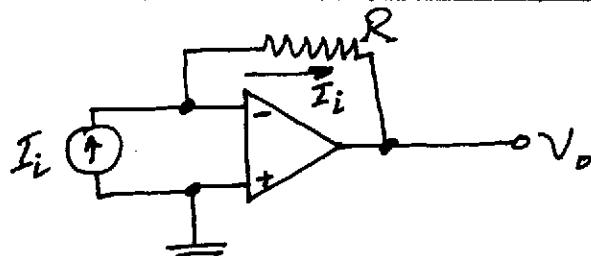
Now, the voltage appearing at the non-inverting input changes to $-V_{SAT} * R_1 / (R_1 + R_2)$. The capacitor starts discharging and after reaching zero, it begins to discharge towards $-V_{SAT}$. Again, as soon as it becomes more negative than the voltage appearing at the non-inverting input of the Op-Amp, the output switches back to $+V_{SAT}$.

The expression for the time period of the output rectangular waveform is given by; $T = 2 RC \ln \left(\frac{1+B}{1-B} \right)$ where, $B = R_1 / (R_1 + R_2)$. By varying the value of resistor R, the time period of the output waveform can be varied.

CURRENT-TO-VOLTAGE CONVERTER:

Current-to-voltage converter is nothing but a transimpedance amplifier. An ideal transimpedance amplifier makes a perfect current-to-voltage converter as it has zero input impedance and zero output impedance.

ANALOG AND DIGITAL ELECTRONICS



Current-to-Voltage Converter

The circuit is characterized by voltage shunt feedback with a feedback factor of unity. The expression for output voltage is given by;

$$V_o = I_i * R * \left(\frac{A_{OL}}{1 + A_{OL}} \right)$$

$$\text{For } A_{OL} \gg 1; \quad V_o = I_i * R \quad \text{Also, } Z_{in} = \left(\frac{R}{1 + A_{OL}} \right) \quad \text{and} \quad Z_o = \left(\frac{R_o}{1 + A_{OL}} \right)$$

where, R_o is the output impedance of the Op-Amp.

VOLTAGE-TO-CURRENT CONVERTER:

Voltage-to-current converter is a case of a transconductance amplifier. An ideal transconductance amplifier makes a perfect voltage-controlled current source or a voltage-to-current converter.

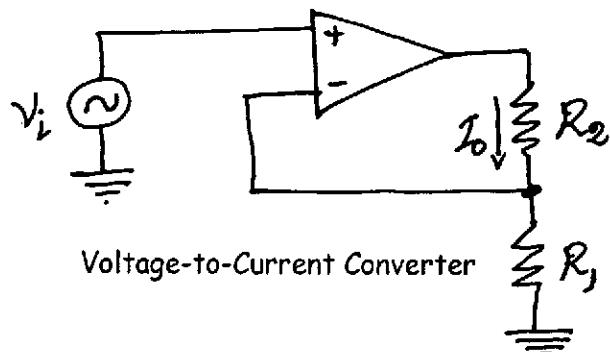
The circuit is characterized by current series feedback. The expression for output current is given by;

$$I_o = \frac{V_i}{R_f + [(R_1 + R_2)/A_{OL}]} \quad \text{For } A_{OL} \gg 1, \quad I_o = \frac{V_i}{R_1}$$

$$\text{Closed-loop input impedance; } Z_{in} = R_i * \left(1 + A_{OL} * \frac{R_1}{R_1 + R_2} \right)$$

where, R_i is the input impedance of the Op-Amp.

$$\text{Closed-loop output impedance; } Z_o = R_1 * \left(1 + A_{OL} * \frac{R_1}{R_1 + R_2} \right)$$

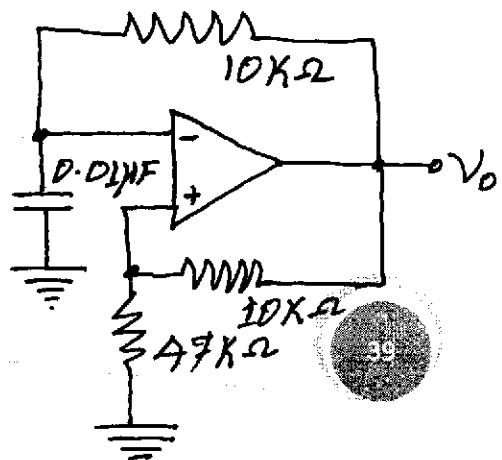


Problem: Refer to the following relaxation oscillator circuit. Determine the peak-to-peak amplitude and frequency of the square wave output; given that, the saturation output voltage of the Op-Amp is ± 12.5 V at power supply voltages of ± 15 V.

Solution:

• The feedback factor,
 $B = \frac{47 \times 10^3}{47 \times 10^3 + 10 \times 10^3} = 0.825$

MAHESH PRASANNA K, VCET, PUTTUR



ANALOG AND DIGITAL ELECTRONICS

• The Time period T is given by; $T = 2RC \ln \left[\frac{1+B}{1-B} \right]$

$$= 2 \times 10 \times 10^3 \times 0.01 \times 10^{-6} \ln \left[\frac{1+0.825}{1-0.825} \right]$$

$$= \underline{\underline{0.469 \text{ ms.}}}$$

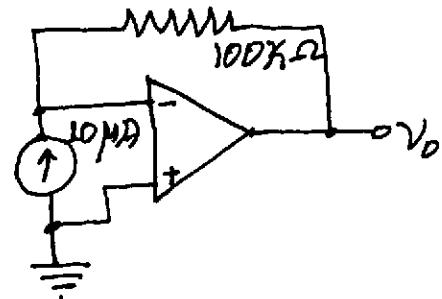
$$\therefore f = \frac{1}{0.469 \times 10^{-3}} = \underline{\underline{2.13 \text{ kHz}}}$$

Peak-to-Peak amplitude of output = $2V_{SAT} = \underline{\underline{25 \text{ V.}}}$

Problem: For current-to-voltage converter shown in the following Fig, determine output voltage, closed loop input and output impedances; given that the Op-Amp has open-loop transimpedance gain of 100,000, input impedance of $1 \text{ M}\Omega$, and output impedance of 100Ω .

Solution:

• Output voltage, $V_o = 10 \times 10^{-6} \times 100 \times 10^3$
 $= \underline{\underline{1 \text{ V.}}}$



• Closed-loop input impedance, $Z_{in} = \frac{R}{1 + A_{DL}} = \frac{200 \times 10^3}{1 + 100,000} = \underline{\underline{2 \Omega}}$

• Closed-loop output impedance, $Z_o = \frac{R_o}{1 + A_{DL}} = \frac{100}{1 + 100,000} = \underline{\underline{0.001 \Omega}}$

By: MAHESH PRASANNA K.,
 DEPT. OF CSE, VCET.

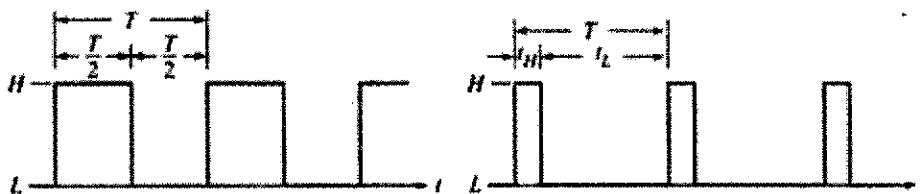
MODULE – 2**THE BASIC GATES & THE COMBINATIONAL LOGIC CIRCUITS**THE BASIC GATES**PREREQUISITES:**

Electronic circuits and systems can be divided into two broad categories – *analog* and *digital*. Analog circuits are designed for use with small signals and are used in a linear fashion. Digital circuits are generally used with large signals and are considered nonlinear. Any quantity that changes with time can be represented as an analog signal or it can be treated as digital signal.

Digital electronics involves circuits that have exactly two possible states. A system having only two states is said to be *binary*. The binary number system is widely used in digital electronics.

Hexa-Decimal	Decimal	Binary	Hexa-Decimal	Decimal	Binary
0	0	0 0 0 0	8	8	1 0 0 0
1	1	0 0 0 1	9	9	1 0 0 1
2	2	0 0 1 0	A	10	1 0 1 0
3	3	0 0 1 1	B	11	1 0 1 1
4	4	0 1 0 0	C	12	1 1 0 0
5	5	0 1 0 1	D	13	1 1 0 1
6	6	0 1 1 0	E	14	1 1 1 0
7	7	0 1 1 1	F	15	1 1 1 1

The operation of electronic circuits can be described in terms of its voltage levels – *high* (H) level and *low* (L) level. This could be related to the binary number system by assigning L = 0 = F (false) and H = 1 = T (true).



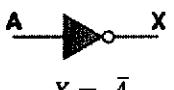
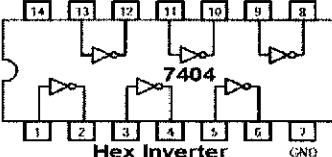
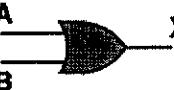
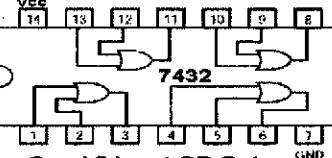
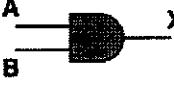
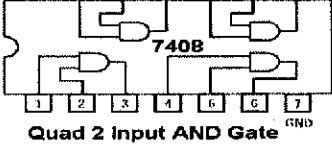
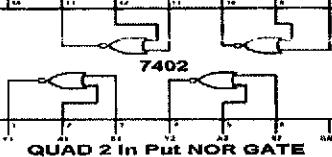
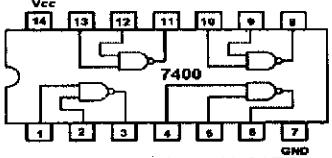
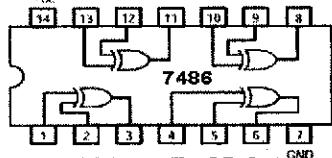
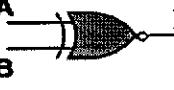
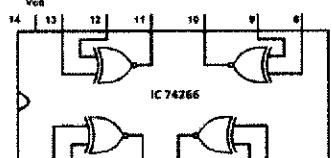
Symmetrical Signal & Asymmetrical Signal

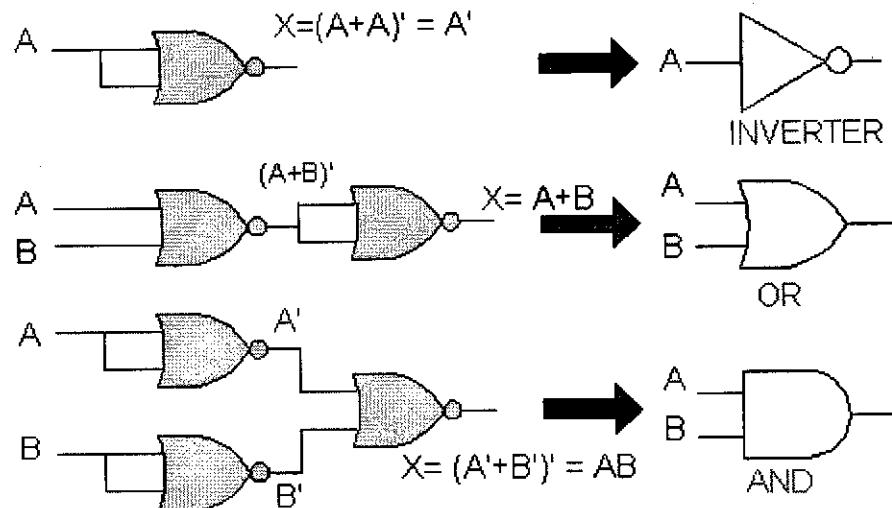
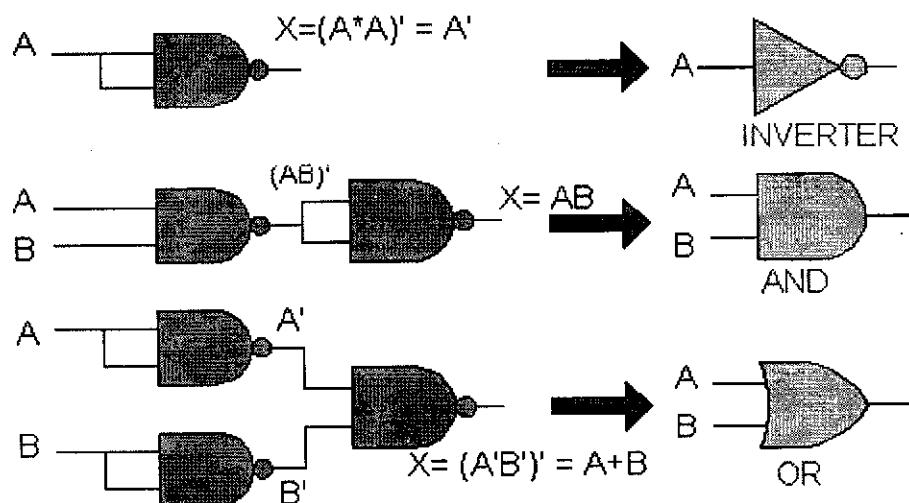
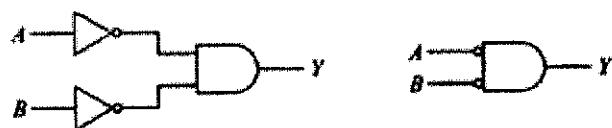
The frequency is defined as, $f = 1 / T$ where, T is the period of the signal.

Duty Cycle is a convenient measure of how symmetrical or how unsymmetrical a waveform is.

$$\text{Duty Cycle} = \frac{T_{on}}{T_{on} + T_{off}} \quad \text{Duty Cycle, } H = \frac{T_{on}}{T_{on} + T_{off}} \quad \text{Duty Cycle, } L = \frac{T_{off}}{T_{on} + T_{off}}$$

REVIEW OF LOGIC GATES:

Circuit Symbol	Truth Table			Verilog	IC Details
	A	B	X		
NOT Gate:  $X = \bar{A}$	0	-	1	$X = \sim A$ $\text{not}(X, A)$	
	1	-	0		
OR Gate:  $X = A + B$	0 0 1 1	0	0	$X = A \mid B$ $\text{or}(X, A, B)$	
		1	0		
		1	1		
AND Gate:  $X = A \cdot B$		0	0	$X = A \& B$ $\text{and}(X, A, B)$	
		0	1		
		1	0		
		1	1		
NOR Gate:  $X = \overline{A+B}$	0 0 1 1	0	1	$X = \sim(A \mid B)$ $\text{nor}(X, A, B)$	
		1	0		
		1	1		
NAND Gate:  $X = \overline{A \cdot B}$		0	0	$X = \sim(A \& B)$ $\text{nand}(X, A, B)$	
		0	1		
		1	0		
		1	1		
XOR Gate:  $X = A \oplus B$ $= \bar{A}B + A\bar{B}$	0 0 1 1	0	0	$X = A \wedge B$ $\text{xor}(X, A, B)$	
		0	1		
		1	0		
		1	1		
XNOR Gate:  $X = A \odot B$ $= \bar{A}\bar{B} + A\bar{B}$	0 0 1 1	0	1	$X = \sim(A \wedge B)$ $\text{xnor}(X, A, B)$	
		0	1		
		1	0		
		1	1		

Universality of NOR Gate:**Universality of NAND Gate:****Bubbled AND Gate:**

Bubbled AND gate and NOR gate are equivalent

De Morgan's First Theorem:

The complement of a sum equals the product of the complements.

$$\overline{A + B} = \bar{A} \cdot \bar{B}$$

Proof:

A	B	$A+B$	$\bar{A}+\bar{B}$	\bar{A}	\bar{B}	$\bar{A}\cdot\bar{B}$
0	0	0	1	1	1	1
0	1	1	0	1	0	0
1	0	1	0	0	1	0
1	1	1	0	0	0	0

NOR Gate

Bubbled AND Gate



Bubbled OR Gate:



Bubbled OR gate and NAND gate are equivalent

De Morgan's Second Theorem:

The complement of a sum equals the product of the complements. $\bar{A}\bar{B} = \bar{A} + \bar{B}$

Proof:

A	B	AB	\bar{AB}	\bar{A}	\bar{B}	$\bar{A}+\bar{B}$
0	0	0	1	1	1	1
0	1	0	1	1	0	1
1	0	0	1	0	1	1
1	1	1	0	0	0	0

NAND Gate

Bubbled OR Gate



Duality Theorem: Starting with a Boolean relation, you can derive another Boolean relation by –

1. Changing each OR sign to an AND sign
2. Changing each AND sign to an OR sign
3. Complementing any 0 or 1 appearing in the expression.

Example: 1. We say that, $A+0 = A$; the dual is, $A \cdot 1 = A$ 2. Consider, $A(B+C) = AB + AC$

By changing the OR and AND operation, we get the dual relation:

$$A + BC = (A+B)(A+C)$$

Laws of Boolean Algebra:

- ✓ The following laws are of immense use in the simplification of Boolean expressions.

- ✓ Note that, if A is a variable, then either $A = 0$ or $A = 1$. Also, when $A = 0$, $A \neq 1$; and when $A = 1$, $A \neq 0$.

De Morgan's First Theorem:-

The complement of sum is equal to the product of the complements.

$$(A + B)' = A' \cdot B' \quad \text{i.e., a bubbled AND gate \& a NOR gate are equivalent.}$$

De Morgan's Second Theorem:-

The complement of a product is equal to the sum of the compliments.

$$(A \cdot B)' = A' + B' \quad \text{i.e., a bubbled OR gate \& a NAND gate are equivalent.}$$

1) Commutative Law:-

$$A + B = B + A \quad \text{and} \quad A \cdot B = B \cdot A$$

2) Associative Law:-

$$A + (B + C) = (A + B) + C \quad \text{and} \quad A \cdot (BC) = (AB) \cdot C$$

3) Distributive Law:-

$$A(B + C) = AB + AC$$

4) In relation to OR operation, the following laws hold good:-

$$A + 0 = A$$

$$A + A = A$$

$$A + 1 = 1 \text{ and}$$

$$A + A' = 1$$

5) In relation to AND operation, the following laws hold good:-

$$A \cdot 1 = A$$

$$A \cdot A = A$$

$$A \cdot 0 = 0$$

$$A \cdot A' = 0$$

$$A'' = A$$

6) Some more useful Boolean relations:-

$$A + AB = A$$

$$A + A'B = A + B$$

$$A(A + B) = A$$

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

$$A + (B \cdot C) = (A + B)(A + C)$$

Simplification of Boolean Expressions:-

- ❖ The following hints are found to be of use, in reducing complex Boolean expressions –

1. If there are parentheses present in the given expression, they are removed first; since, multiplication should precede addition.

E.g.: - $AB + C(A + B) = AB + AC + BC$

2. If there are several identical terms, all except one can be removed.

E.g.: - $A + B + C + A \cdot 1 = A + B + C + A = A + B + C$

3. If a variable repeats in a term, only one variable may be retained.

E.g.: - $A \cdot A = A$

$B \cdot B \cdot C = BC$

4. If in any term, both a variable & its complement are present, that term may be removed; since, $AA' = 0$.

E.g.: - $XX'Y = 0 \cdot Y = 0$

5. Identify pairs of terms which contains same variables. If in a pair, a variable is absent in one term, it can be removed.

E.g.: - $ABCD + ABC = ABC(D + 1)$

$= ABC \cdot 1 \quad \text{since, } 1 + D = 1$

$= ABC$

6. If, in a pair of terms, several variables are common, and another variable is present in one term & its complement is present in another term, this variable & its complement can be removed.

E.g.: - $ABC + A'BC = BC(A' + A)$

$= BC \cdot 1 \quad \text{since, } A' + A = 1$

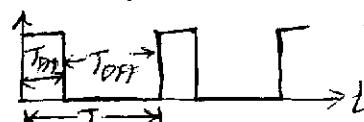
$= BC$

Problem: A signal waveform has a frequency of 5 MHz, and the width of the positive pulse is 0.05 μs.

What is the high duty cycle?

Solution: Given: $f = 5 \text{ MHz}$ & $T_{ON} = 0.05 \mu\text{s}$.

The period of the waveform is; $T = \frac{1}{f} = \frac{1}{5 \times 10^6} = 0.2 \mu\text{s}$.

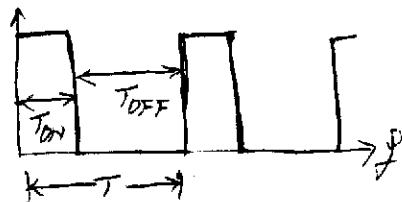


∴ Duty cycle, $H = \frac{T_{ON}}{T} = \frac{0.05 \mu\text{s}}{0.2 \mu\text{s}} = \underline{\underline{0.25 \text{ or } 25\%}}$.

Problem: An asymmetrical signal waveform is high for 2 ms and low for 5 ms. Find the frequency and duty cycle L of the waveform.

MAHESH PRASANNA K., VCET, PUTTUR





Solution:

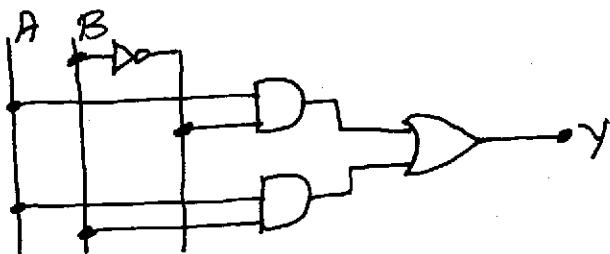
Given: $T_{on} = 2\text{ms}$ & $T_{off} = 5\text{ms}$.

$$\text{Frequency, } f = \frac{1}{T} = \frac{1}{T_{on} + T_{off}} = \frac{1}{(2+5) \times 10^{-3}} = \frac{1}{7 \times 10^{-3}} = 142.86 \text{ Hz}$$

$$\text{Duty cycle, } D = \frac{T_{off}}{T} = \frac{5 \times 10^{-3}}{7 \times 10^{-3}} = D \cdot 7/4 = \underline{\underline{7/4}}$$

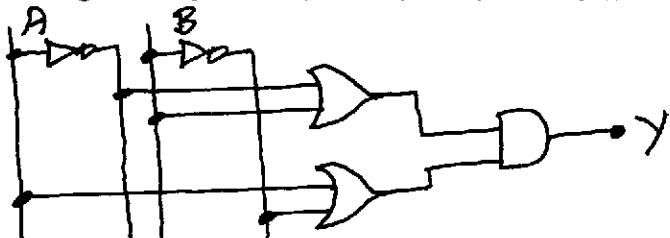
Problem: Show the logic circuit for; $Y = A\bar{B} + AB$

Solution:



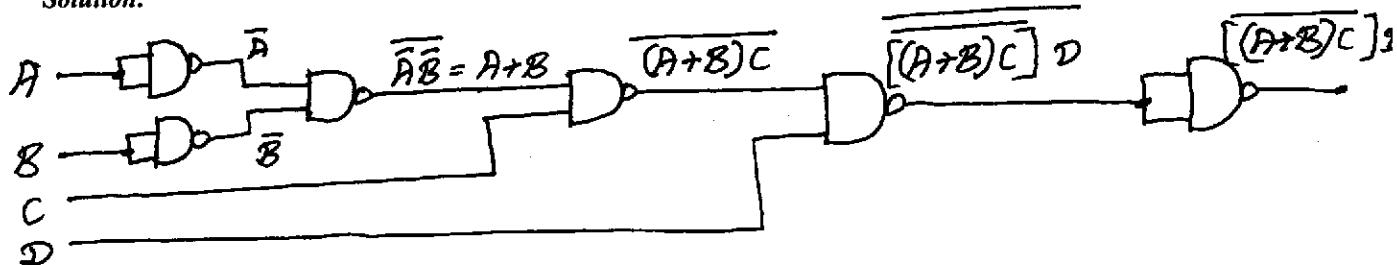
Problem: Show the logic circuit for; $Y = (\bar{A} + B)(A + \bar{B})$ and simplify.

Solution:



Problem: Implement the following function using only NAND gates: $(\overline{(A+B)}C)D$.

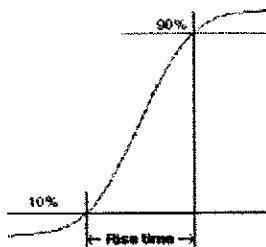
Solution:



NOTE: The signal changes between the logic levels are not instantaneous but take an amount of time.

The time taken for the signal voltage to rise from low-level to a high-level is called *rise time*, t_r .

The time for the signal voltage to fall from a high-level to a low-level is called *fall time*, t_f .



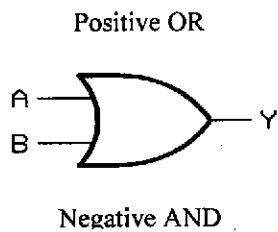
POSITIVE AND NEGATIVE LOGIC:

If we use a binary 0 for low voltage and binary 1 for high voltage, then it is called *positive logic*. Choosing H = 1 = T and L = 0 = F is called positive logic. If we use a binary 0 for high voltage and binary 1 for low voltage, then it is called *negative logic*. Choosing H = 0 = F and L = 1 = T is called negative logic.

Positive and Negative Gates:

In a positive logic system, binary 0 stands for low and binary 1 for high. Consider the following Table. Note that, Y is 1 if either A or B is 1. This is OR gate; and it is because, we are using positive logic.

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



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

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

In a negative logic system, binary 1 stands for low and binary 0 for high. With this code we can convert the 1st Table to 3rd Table. Here, the output Y is 1, only when both A and B are 1. This is AND gate; and it is because, we are using negative logic. Hence, gates are defined by the way they process the binary 0s and 1s.

In the similar way, we can find the following equivalences between the positive and negative logic:

Positive OR	\leftrightarrow	Negative AND
Positive AND	\leftrightarrow	Negative OR
Positive NOR	\leftrightarrow	Negative NAND
Positive NAND	\leftrightarrow	Negative NOR

Assertion-Level Logic:

Many designers draw logic circuits with bubbles on all pins with active-low signals or omit bubbles on all pins with active-high signals. This use of bubbles with active-low signals is called *assertion-level logic*. It means that you draw chips with the kind of input that causes something to happen, or with the kind of output that indicates something has happened. If a low input turns on a chip, you show a bubble on that input. If a low output is a sign of chip action, you draw a bubble on that output. You can equate the word assert with activate.

E.g.: The 74150 Multiplexer has an active low input STROBE; this input turns on the chip only when it is low. This is an active-low signal; which causes something to happen when it is low, rather than high.

INTRODUCTION TO HDL:

Hardware Description Language (HDL) – a textual description of a digital circuit – a language which is more crisp, and machine readable.

Advantages:

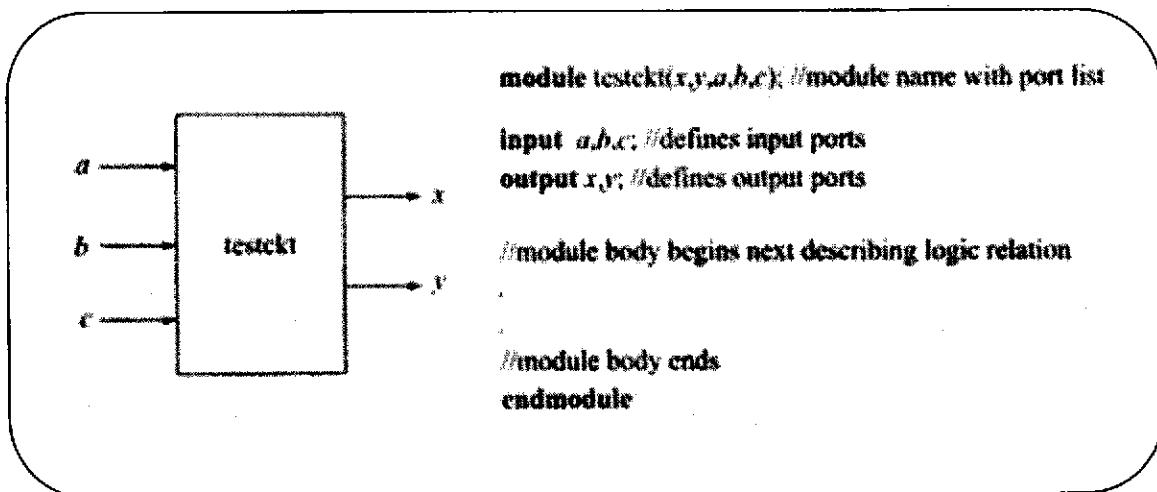
1. To describe a large complex design requiring hundreds of logic gates in a convenient manner, in a smaller space
2. To use software test bench to detect functional error, if any and correct it (called *simulation*)
3. To get hardware implementation details (called *synthesis*)

There are two widely used HDLs – Verilog and Very high speed integrated circuit Hardware Description Language (VHDL). Verilog is considered simple of the two and is more popular.

Verilog HDL:

Verilog, introduced in 1980, as a simulation and verification tool by *Gateway Design Associations*, later acquired by *Cadence Data Systems*. Put to public domain in 1990, and is now controlled by a group of companies and universities, called *Open Verilog International*.

Describing Input Output: In any digital circuit, there will be a set inputs and a set of outputs, often termed as *ports*. The relationship between these inputs and outputs are explained within the digital circuit. To design any circuit, that has (say) three inputs a, b, c and two outputs x, y as shown in the following Fig; the corresponding Verilog code can be written as follows:



Note that, *module* and *endmodule* are keywords for Verilog. A *module* describes a design entity with a name or identifier selected by user (here it is *testckt*) followed by input output port list. The symbol “//” is used to put comments and improve readability for a human. The module body describes the logic within the black box which acts on the inputs a, b, c and generates output x, y. Observe, where semicolon “;” is used and where not to end the statement.

Writing Module Body: There are three different models of writing module body in Verilog HDL – Structural, Data flow, and Behavioral.

Structural Modeling:

```
module or_gate(A,B,Y);
  input A,B; // defines two input port
  output Y; // defines one output port
  or g1(Y,A,B); /*Gate declaration with predefined keyword or representing
                   logic OR. g1 is optional user defined gate identifier */
endmodule
```



Verilog supports predefined gate level primitives such as and, or, not, nor, nand, xor, xnor, etc. The syntax followed above can be extended to other gates.

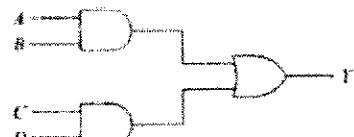
For NOT Gate: not (output, input)

For 2 input OR Gate or (output, input1, input2)

For 4 input OR Gate or (output, input1, input2, input3, input4)

Note that, Verilog can take up to 12 inputs for logic gates. Comments when extended to the next line is written within /* . . . */. Identifiers in Verilog are case sensitive, begin with a letter or underscore and can be of any length. Observe the following:

```
module fig2_24a(A,B,C,D,Y);
  input A,B,C,D;
  output Y;
  wire and_op1, and_op2; // internal connections
  and g1(and_op1,A,B); // g1 represents upper AND gate
  and g2(and_op2,C,D); // g2 represents lower AND gate
  or g3(Y, and_op1, and_op2); // g3 represents the OR gate
endmodule
```



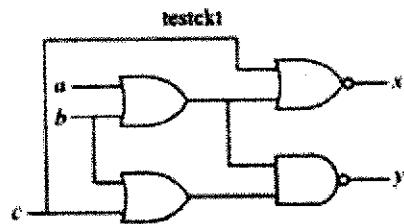
Note that, we define two intermediate variables; and_op1 and and_op2, representing two AND gate outputs through keyword *wire*. Wire represents a physical wire in a circuit.

Now, write the Verilog code for the testckt (shown below):

```

module testckt(a,b,c,x,y);
  input a,b,c;
  output x,y;
  wire or_op1, or_op2; /* internal
    connections, outputs of upper and
    lower OR gates respectively */
  or g1(or_op1,a,b); // g1 represents upper OR gate
  or g2(or_op2,b,c); // g2 represents lower OR gate
  nor g3(x,c,or_op1); // g3 represents the NOR gate
  nand g4(y,or_op1,or_op2); // g4 represents the NAND gate
endmodule

```



Preparation of Test Bench: A test bench in Verilog is used to simulate a digital circuit. Consider the example of simulating a simple OR gate, for which Verilog code is described. The test bench creates the input in the form of a timing waveform and passes this to OR gate module through a function or procedural call. To generate timing waveform, we use the time delay available in Verilog in the form of `#n` where n denotes a number in decimal that gives delay in nanosecond; E.g.: `#20`. (NOTE: All practical logic circuit comes with finite gate delay, i.e., output changes according to input after certain time).

```

module testor;                      // Simulation module given a name testor
  reg A, B;                         // Storage of data for passing it to module OR_Gate
  wire X;
  OR_Gate org (A, B, Y);           // Circuit is instantiated with name OR_Gate
  initial                           // Start simulation
    begin                            /* Input is generated to test the circuit through following
      statements, simulation begins */
      A = 1'b0; B = 1'b0;          // 1'b0 signifies on binary digit with value 0, AB = 00
      #20
      A = 1'b0; B = 1'b1;          // After 20 ns, AB = 01
      #20
      A = 1'b1; B = 1'b0;          // After 20 ns, AB = 10
      #20
      A = 1'b0; B = 1'b1;          // After 20 ns, AB = 11
      #20
    end
  endmodule

```

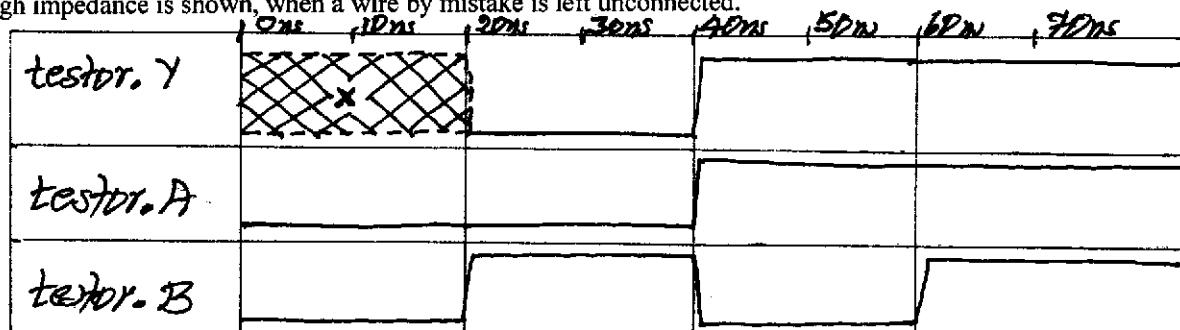
ANALOG AND DIGITAL ELECTRONICS

```

module OR_Gate (A, B, Y); // OR gate used as procedure in simulation
    input A, B; // Define two input ports (for two input OR gate)
    output Y; // Define one output port (OR gate output)
    or #(20) g1(Y, A, B); /* Gate declaration with a gate delay of 20 ns; output will be
                           effected after 20 ns */
endmodule

```

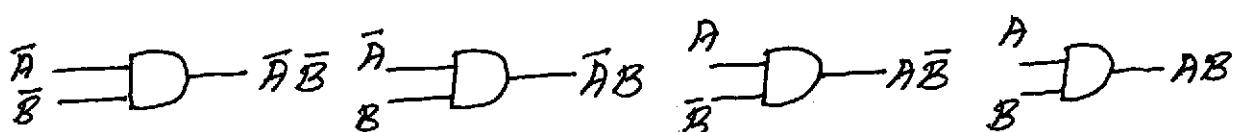
The input AB given by testor is taking values 00, 01, 10, 11 and retain them for 20 ns. Output of OR gate changes according to input but after a delay of 20 ns. For first 20 ns, OR gate output is unknown, as it needs 20 ns (gate delay) to respond. Note that, Verilog, in general offers four logic values in simulation: 0, 1, x (unknown), and z (high impedance). Unknown value is exhibited, when the input is ambiguous and high impedance is shown, when a wire by mistake is left unconnected.



Verilog Simulation of 2 input OR gate with 20 ns given Delay

COMBINATIONAL LOGIC CIRCUITSSUM-OF-PRODUCTS (SOP) METHOD:

The following Fig shows four possible ways to AND two input signals that are in complemented and un-complemented form. These outputs are called *fundamental products*.



The following Table lists each fundamental product next to the input conditions producing a high output.

A	B	Fundamental Products		A	B	C	Fundamental Products
0	0	$\bar{A}\bar{B}$		0	0	0	$\bar{A}\bar{B}\bar{C}$
0	1	$\bar{A}B$		0	0	1	$\bar{A}\bar{B}C$
1	0	$A\bar{B}$		0	1	0	$\bar{A}BC$
1	1	AB		0	1	1	$\bar{A}BC$



		1	0	0	$A\bar{B}\bar{C}$
		1	0	1	$A\bar{B}C$
		1	1	0	$A\bar{B}\bar{C}$
		1	1	1	ABC

The fundamental products are also called *minterms*. Products are represented by m_0 , m_1 , m_2 , and m_3 respectively.

Sum-of-Products Equation: Given a truth table, to get the sum-of-products solution –

1. Locate each output 1 in the truth table and write down the fundamental products
2. Identify all the fundamental products
3. OR the fundamental product.

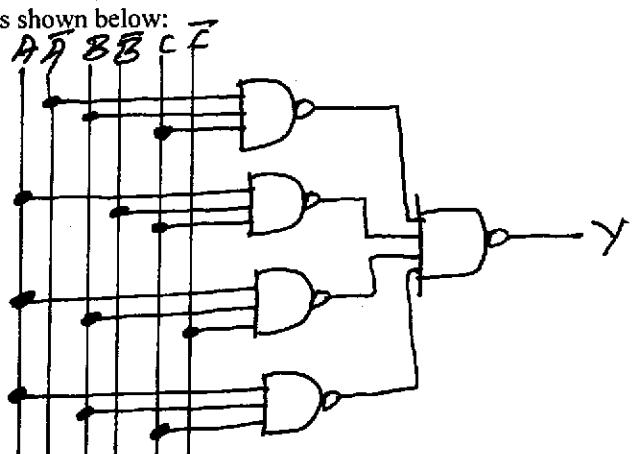
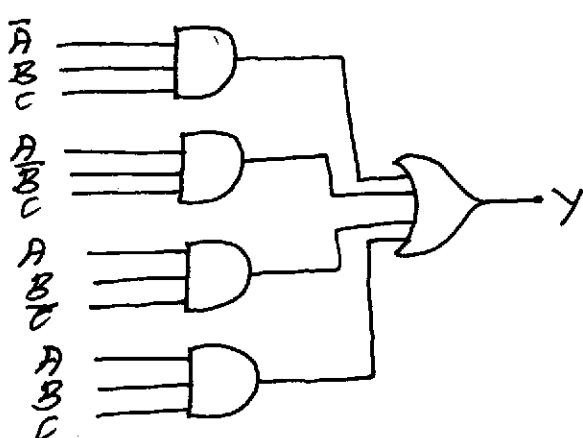
Problem: For the following truth table, get the sum-of-products solution.

A	B	C	Y
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

Solution:

A	B	C	Y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1 $\rightarrow \bar{A}BC$
1	0	0	0
1	0	1	1 $\rightarrow A\bar{B}C$
1	1	0	1 $\rightarrow A\bar{B}\bar{C}$
1	1	1	1 $\rightarrow ABC$

Logic Circuit: After getting a sum-of-products equation, derive the corresponding logic circuit by drawing an AND-OR network or NAND-NAND network, as shown below:



AND-OR Solution & NAND-NAND Solution

$$\begin{aligned}
 Y &= \bar{A}BC \bar{A}\bar{B}C A\bar{B}\bar{C} \bar{A}\bar{B}\bar{C} \\
 &= \bar{A}BC + \bar{A}\bar{B}C + A\bar{B}\bar{C} + \bar{A}\bar{B}\bar{C}.
 \end{aligned}$$

ANALOG AND DIGITAL ELECTRONICS

Problem: Simplify the Boolean equation: $Y = \bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + A\bar{B}\bar{C} + AB\bar{C}$

Solution:

$$\begin{aligned}
 Y &= \bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + A\bar{B}\bar{C} + AB\bar{C} \\
 &= \bar{C} [\bar{A}\bar{B} + \bar{A}B + A\bar{B} + AB] \\
 &= \bar{C} [\bar{A}(\bar{B} + B) + A(\bar{B} + B)] \\
 &= \bar{C} [\bar{A} \cdot 1 + A \cdot 1] = \bar{C} \cdot 1 = \bar{C}.
 \end{aligned}$$

	\bar{C}	C
$\bar{A}\bar{B}$	1	0
$\bar{A}B$	1	0
$A\bar{B}$	1	0
AB	1	0

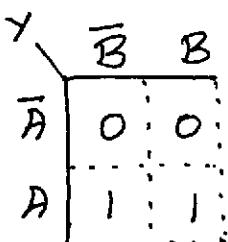
$\underline{\underline{Y = \bar{C}}}$

TRUTH TABLE TO KARNAUGH MAP (K-MAP):

A Karnaugh Map is a visual display of the fundamental products needed for a sum-of-products solution.

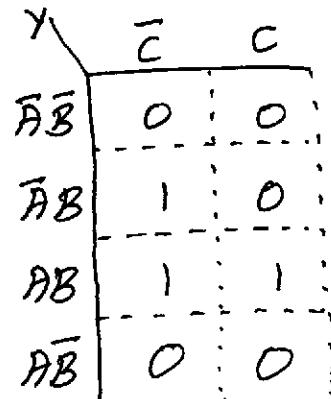
Two-Variable Map:

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



Three-Variable Map:

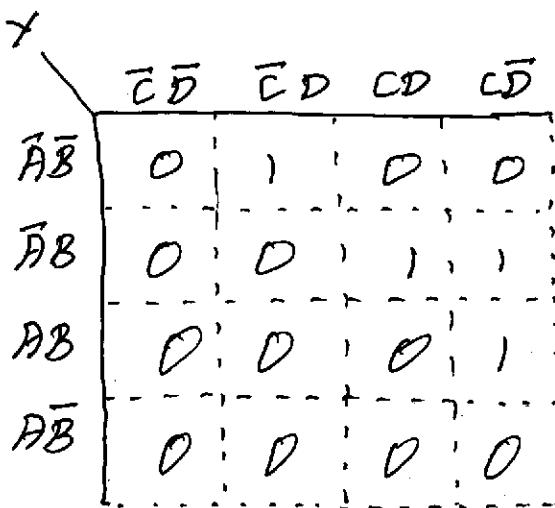
A	B	C	Y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1



Four-Variable Map:

A	B	C	D	Y
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	1

A	B	C	D	Y
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	1
1	1	1	1	1



PIRS, QUADS, AND OCTETS:

Pairs: The following K-map contains a pair of 1s that are horizontally adjacent. Two adjacent 1s, such as these are called a *pair*. A pair eliminates one variable and its complement.

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0	0	0	0
$\bar{A}B$	0	0	0	0
$A\bar{B}$	0	0	(1)	(1)
AB	0	0	0	0

The sum of products eqn is;

$$\begin{aligned} Y &= ABCD + ABC\bar{D} \\ &= ABC(D + \bar{D}) \\ &= ABC. \end{aligned}$$

Quad: A *quad* is a group of four 1s that are horizontally or vertically adjacent. A quad eliminates two variables and their complements.

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0	0	0	0
$\bar{A}B$	0	0	0	0
$A\bar{B}$	(1)	(1)	(1)	(1)
AB	0	0	0	0

$$\begin{aligned} Y &= ABC\bar{C} + ABC\bar{C} \\ &= AB(\bar{C} + C) \\ &= AB. \end{aligned}$$

The Octet: The *octet* is a group of eight 1s, as shown in the following Fig. An octet eliminates three variables and their complements.

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0	0	0	0
$\bar{A}B$	0	0	0	0
$A\bar{B}$	(1)	(1)	(1)	(1)
AB	(1)	(1)	(1)	(1)

$$\begin{aligned} Y &= AB + A\bar{B} \\ &= A(B + \bar{B}) \\ &= A. \end{aligned}$$

KARNAUGH SIMPLIFICATIONS:

A pair eliminates one variable and its complement. A quad eliminates two variables and their complements. An octet eliminates three variables and their complements. Because of this, after drawing the K-map, first encircle the octets, then the quads, and finally the pairs. Hence, the greatest simplification results.

Problem: Using K-map, simplify; $Y = \sum m(1, 2, 3, 6, 8, 9, 10, 12, 13, 14)$.

Solution:

	$\bar{C}\bar{D}$	$\bar{C}D$	$C\bar{D}$	CD	Y
$\bar{A}\bar{B}$	0	1	1	1	(889)
$\bar{A}B$	0	0	0	1	
$A\bar{B}$	1	1	0	1	(89)
AB	1	1	0	1	(12)

Overlapping Groups: Always overlap groups.

	$\bar{C}\bar{D}$	$\bar{C}D$	$C\bar{D}$	CD	Y_1
$\bar{A}\bar{B}$	0	0	0	0	
$\bar{A}B$	0	1	0	0	
$A\bar{B}$	1	1	1	1	
AB	1	1	1	1	

	$\bar{C}\bar{D}$	$\bar{C}D$	$C\bar{D}$	CD	Y_2
$\bar{A}\bar{B}$	0	0	0	0	
$\bar{A}B$	0	1	0	0	
$A\bar{B}$	1	1	1	1	
AB	1	1	1	1	

Rolling the Map:

	$\bar{C}\bar{D}$	$\bar{C}D$	$C\bar{D}$	CD	Y_3
$\bar{A}\bar{B}$	0	0	0	0	
$\bar{A}B$	1	0	0	0	
$A\bar{B}$	1	0	0	0	
AB	0	0	0	0	

$$Y_3 = B\bar{C}\bar{D} + B\bar{C}D$$

	$\bar{C}\bar{D}$	$\bar{C}D$	$C\bar{D}$	CD	Y_4
$\bar{A}\bar{B}$	0	0	0	0	
$\bar{A}B$	1	0	0	0	
$A\bar{B}$	1	0	0	0	
AB	0	0	0	0	

$$Y_4 = BD$$

Budding and Overlapping:

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	1	1	0	0
$\bar{A}B$	1	1	0	1
$A\bar{B}$	1	1	0	1
AB	1	1	0	0

$$Y_1 = \bar{C} + B\bar{C}\bar{D}$$

$$Y_2 = \bar{C} + B\bar{D}$$

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	1	1	0	0
$\bar{A}B$	1	1	0	1
$A\bar{B}$	1	1	0	1
AB	1	1	0	0

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	1	1	0	1
$\bar{A}B$	1	1	0	1
$A\bar{B}$	1	1	0	0
AB	1	1	0	0

$$Y_3 = \bar{C} + \bar{A}C\bar{D} + A\bar{B}C\bar{D}$$

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	1	1	0	1
$\bar{A}B$	1	1	0	1
$A\bar{B}$	1	1	0	0
AB	1	1	0	0

$$Y_4 = \bar{C} + \bar{A}\bar{D} + A\bar{B}\bar{D}$$

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	1	1	0	0
$\bar{A}B$	1	1	0	1
$A\bar{B}$	1	1	0	0
AB	1	1	0	1

$$Y_5 = \bar{C} + \bar{A}\bar{D} + \bar{B}\bar{D}$$

Eliminating Redundant Groups: After encircling groups, eliminate any *redundant groups*. This is a group whose 1s are already used by other groups.

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0	0	1	0
$\bar{A}B$	1	1	1	0
$A\bar{B}$	0	1	1	1
AB	0	1	0	0

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0	0	1	0
$\bar{A}B$	1	1	1	0
$A\bar{B}$	0	1	1	1
AB	0	1	1	0

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0	0	1	0
$\bar{A}B$	1	1	1	0
$A\bar{B}$	0	1	1	1
AB	0	1	0	0

Conclusion: The summary of the K-map method for simplifying Boolean equation:

1. Enter a 1 on the K-map for each fundamental product that produces a 1 output in the truth table. Enter 0s elsewhere.
2. Encircle the octets, quads, and pairs. Remember to roll and overlap to get the largest groups possible.
3. If any isolated 1s remain, encircle each.
4. Eliminate any redundant group.
5. Write the Boolean equation by ORing the products corresponding to the encircled groups.

Problem: What is the simplified Boolean equation for the following equation expressed by minterms?

$$Y = F(A, B, C, D) = \sum m(7, 9, 10, 11, 12, 13, 14, 15). \quad Y = AB + AC + AD + BCD.$$

Solution:

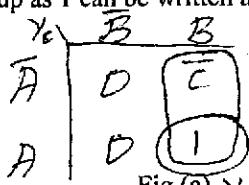
	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0	0	0	0
$\bar{A}B$	0	0	1	0
$A\bar{B}$	1	1	1	1
AB	0	1	1	1

Let $Y = \sum m(2, 6, 7)$.

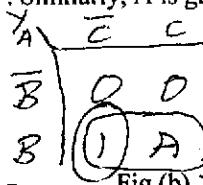
ABC	Y	$A \bar{B}$	Y_c	BC	Y_A
000	0	0.0	0	00	0
001	0	0.1	C	01	0
010	1	1.0	0	10	1
011	0	1.1	1	11	A
100	0	B B	B B	B C	C C
101	0	A 0	C	B 0	D 0
110	1	A 0	C	B 1	D A
111	1	A 0	C	B 1	D A

Entered Variable Map (EVM): In EVM, one of the input variables is placed inside K-map. This reduces the K-map size by one degree; i.e. a three variable problem that requires $2^3 = 8$ locations in K-map will require $2^{(3-1)} = 4$ locations in EVM. This technique is particularly useful for mapping problems with more than four variables.

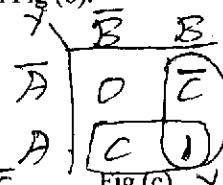
Simplification of EVM: This is similar to K-map method. In Fig (a), C' is grouped with 1 to get a larger group as 1 can be written as $1 = 1 + C'$. Similarly, A is grouped with 1 in Fig (b).



$$\text{Fig (a)} \quad Y_c = BC' + AB$$



$$\text{Fig (b)} \quad Y_A = AB + BC$$



$$\text{Fig (c)} \quad Y = AB + BC$$

Now, the product term representing each group is obtained by including map entered variable (MEV) in the group as an additional ANDed term.

Hence, for Fig (a): $Y = B\bar{C} + AB$. For Fig (b): $Y = B\bar{C} + AB$.

Consider the EBM shown in Fig (c). This has only two product terms; and doesn't need a separate coverage for 1. This is because, one can write $1 = C + C'$, and C is included in one group and C' is included in other group.

MAHESH PRASANNA K., VCET, PUTTUR

Problem: Simplify $Y(A, B, C) = \sum m(2, 6, 7)$ by using entered variable map method by –

- a) "A" as map entered variable
- b) "C" as map entered variable.

Solution:

DON'T CARE CONDITIONS:

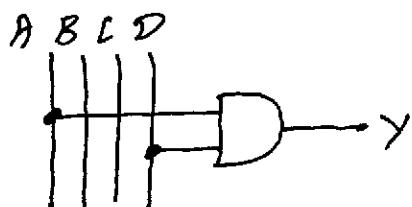
In some digital systems, certain input conditions never occur during normal operation; therefore, the corresponding output never appears. Since the output never appears, it is indicated by an X in the truth table. The X is called a *don't-care condition*.

Example: Consider the following truth table with don't care conditions for all the inputs from 1010 to 1111.

A B C D	Y
0 0 0 0	0
0 0 0 1	0
0 0 1 0	0
0 0 1 1	0
0 1 0 0	0
0 1 0 1	0
0 1 1 0	0
0 1 1 1	0
1 0 0 0	0
1 0 0 1	1
1 0 1 0	X
⋮	⋮
1 1 1 1	X

A	B	C	D	Y	\bar{CD}	$\bar{C}D$	CD	$C\bar{D}$
				\bar{AB}	0	0	0	0
\bar{A}	B	\bar{C}	D	0	0	0	0	0
\bar{A}	B	\bar{C}	\bar{D}	X	X	X	X	X
\bar{A}	B	C	\bar{D}	0	1	X	X	X

$$Y = AD$$



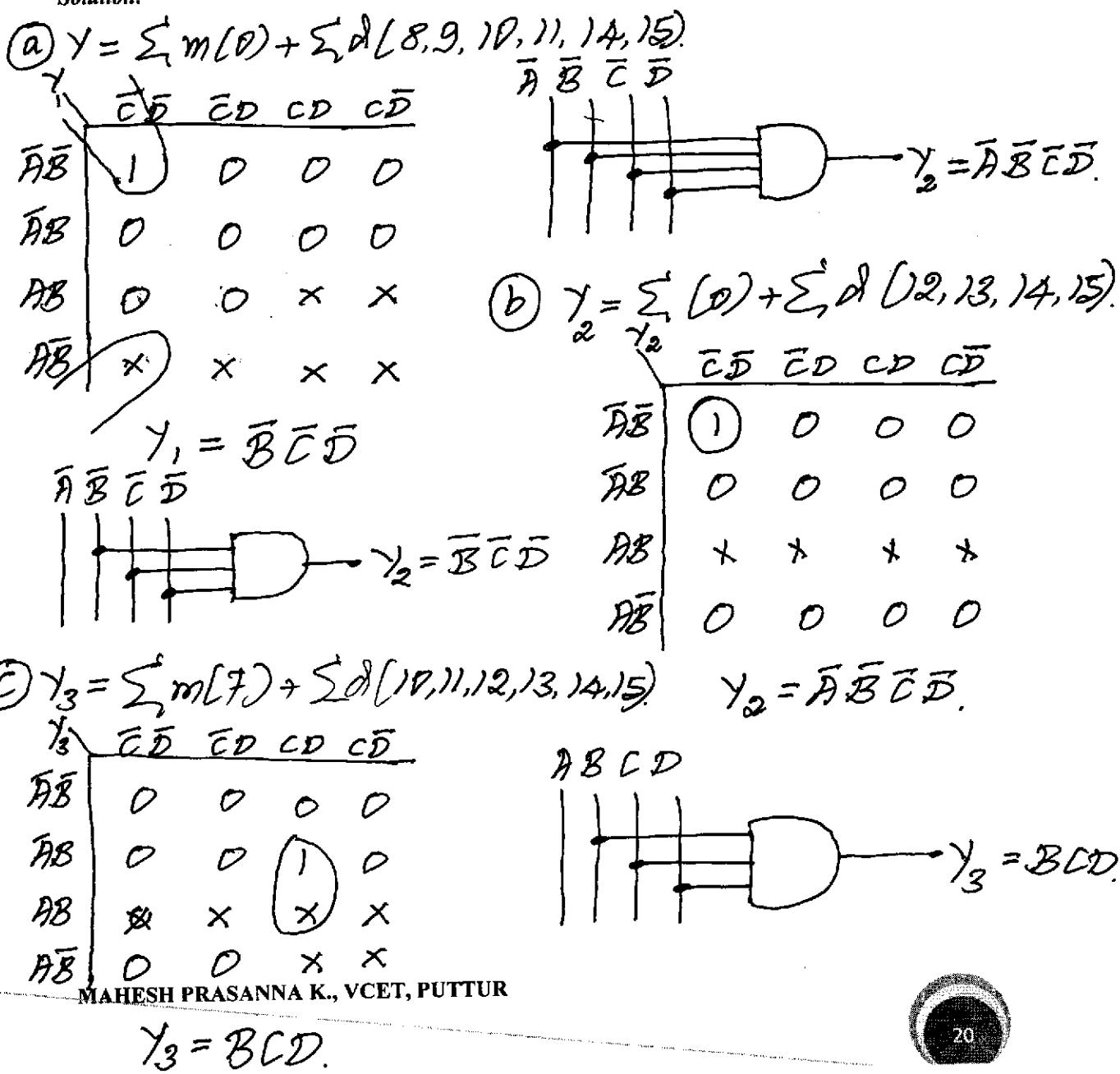
remember these points about don't-care conditions:

- Given the truth table, draw the K-map and transfer 0s, 1s, and don't-care terms.
- Encircle the actual 1s on the K-map in the largest groups you can find treating don't cares as 1s.
- After the actual 1s have been included in the groups, disregard the remaining don't cares by visualizing them as 0s.

Problem: What is the simplest logic circuit for -

- $Y = F(A, B, C, D) = \sum m(0) + \sum d(8, 9, 10, 11, 14, 15)$
- $Y = F(A, B, C, D) = \sum m(0) + \sum d(12, 13, 14, 15)$
- $Y = F(A, B, C, D) = \sum m(7) + \sum d(10, 11, 12, 13, 14, 15)$.

Solution:



PRODUCT-OF-SUMS (POS) METHOD:

With SOP method –

- A fundamental product produces an output 1 for the corresponding input condition.

With POS method –

- A fundamental sum produces an output 0 for the corresponding input condition.

Product-of-Sums Equation: In the following Table, the first output 0 appears for $A = 0, B = 0$, and $C = 0$. The fundamental sum for these inputs is $A + B + C$; because, this produces an output zero for the corresponding input condition: $Y = A + B + C = 0 + 0 + 0 = 0$.

A	B	C	Y – Fundamental Sum	Max-term
0	0	0	$0 - A + B + C$	M0
0	0	1	1	M1
0	1	0	1	M2
0	1	1	$0 - A + \bar{B} + \bar{C}$	M3
1	0	0	1	M4
1	0	1	1	M5
1	1	0	$0 - \bar{A} + \bar{B} + C$	M6
1	1	1	1	M7

The second output 0 appears for the input condition $A = 0, B = 1$, and $C = 1$. The fundamental sum for this is $A + B' + C'$. Notice that, B and C are complemented because; this is the only way to get a logical sum of 0 for the given input condition: $Y = A + \bar{B} + \bar{C} = 0 + \bar{1} + \bar{1} = 0 + 0 + 0 = 0$.

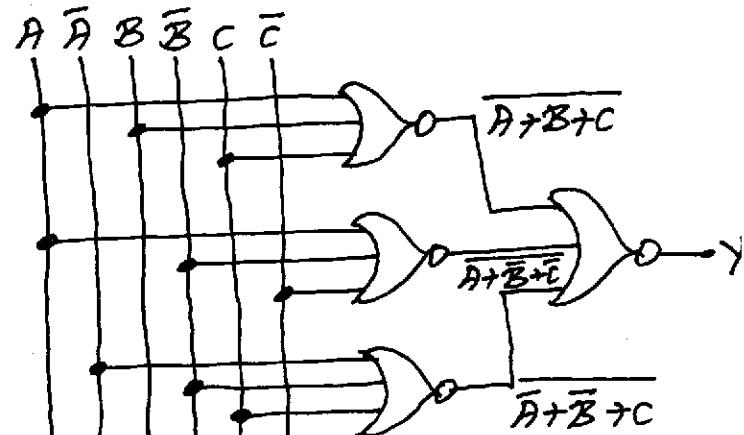
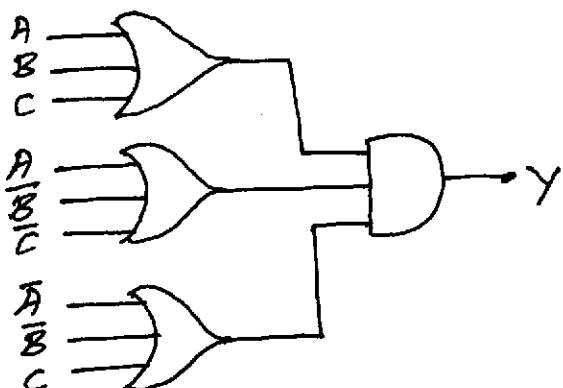
Similarly, the third output 0 occurs for $A = 1, B = 1$, and $C = 0$; hence, its fundamental sum is $A' + B' + C$: $Y = \bar{A} + \bar{B} + C = \bar{1} + \bar{1} + 0 = 0 + 0 + 0 = 0$.

To get the POS equation, AND the fundamental sums:

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

$$\text{OR } Y = F(A, B, C) = \Pi M(0, 3, 6)$$

Logic Circuit:



OR-AND Network & NOR-NOR Network

MAHESH PRASANNA K., VCET, PUTTUR

$$\begin{aligned}
 Y &= \overline{A+B+C} + \overline{A+\bar{B}+\bar{C}} + \overline{\bar{A}+\bar{B}+C} \\
 &= (A+B+C)(A+\bar{B}+\bar{C})(\bar{A}+\bar{B}+C)
 \end{aligned}$$

ANALOG AND DIGITAL ELECTRONICS

In SOP method –

6. Given truth table
7. Identify 1s
8. Write the fundamental products
9. OR the fundamental products
10. AND-OR network or
NAND-NAND network.

In POS method –

1. Given truth table
2. Identify 0s
3. Write the fundamental sums
4. AND the fundamental sums
5. OR-AND network or
NOR-NOR network.

Problem: Write the POS and SOP representations for the following truth tables:

A	B	C	Y
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

Solution:

$$\begin{aligned} Y_{POS} &= \prod M(0, 1, 2, 4) \\ &= (\bar{A} + B + C)(A + \bar{B} + \bar{C}) \\ &\quad (A + \bar{B} + C)(\bar{A} + B + C) \\ Y_{SOP} &= \sum m(3, 5, 6, 7) \\ &= \bar{A}BC + A\bar{B}C + \\ &\quad ABC + A\bar{B}C. \end{aligned}$$

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

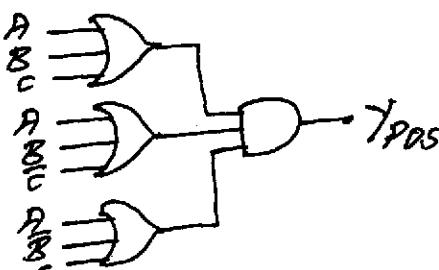
Solution:

$$\begin{aligned} Y_{POS} &= \prod M(0, 3, 6) \\ &= (\bar{A} + B + C)(A + \bar{B} + \bar{C}) \\ &\quad (\bar{A} + \bar{B} + C). \\ Y_{SOP} &= \sum m(1, 2, 4, 5, 7) \\ &= \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + \\ &\quad A\bar{B}C + ABC. \end{aligned}$$

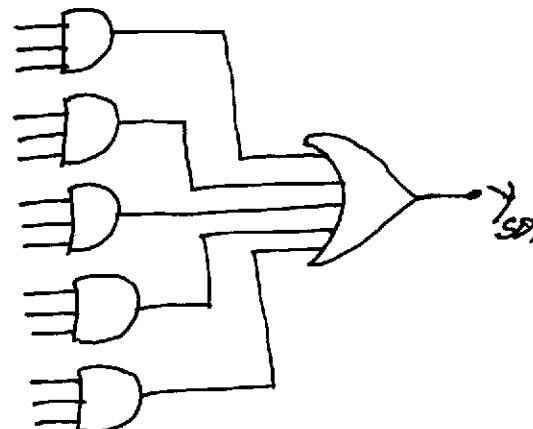
Problem: Suppose a truth table has a low output for the first three input conditions: 000, 001, and 010. If all other outputs are high, write POS and SOP circuits?**Solution:**

ABC	Y
000	0
001	0
010	0
011	1
100	1
101	1
110	1
111	1

$$\begin{aligned} Y_{POS} &= \prod M(1, 2, 4) \\ &= (\bar{A} + B + C)(A + \bar{B} + \bar{C})(A + \bar{B} + C). \end{aligned}$$



$$\begin{aligned} Y_{SOP} &= \sum m(3, 4, 5, 6, 7) \\ &= \bar{A}BC + \bar{A}\bar{B}C + A\bar{B}\bar{C} + \\ &\quad A\bar{B}C + ABC. \end{aligned}$$



PRODUCT-OF-SUMS SIMPLIFICATION:

Method 1: After writing the POS equation, one can simplify by using Boolean algebra.

Method 2: After writing the POS equation, simplification can be done on the K-map.

- Forming largest group of zeros
- Replace each group by a sum term
- The variable going in the formation of sum term is inverted if it remains constant with a value of 1 in the group and it is not inverted if that value is 0
- Finally, all the sum terms are ANDed to get simplest POS form.

Problem: By grouping zeros, give the POS form of –

a) $Y_1 = F(A, B, C, D) = \prod M(0, 1, 2, 3, 4, 5, 7)$

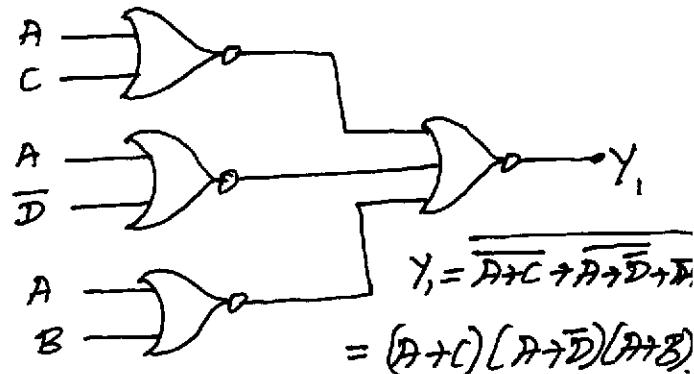
b) $Y_2 = F(A, B, C, D) = \prod M(0, 1, 2, 4, 5, 10) + d(8, 9, 11, 12, 13, 15)$

Solution:

$$Y_1 = \prod M(0, 1, 2, 3, 4, 5, 7)$$

	$\bar{C}\bar{D}$	$\bar{C}D$	$C\bar{D}$	CD	
$\bar{A}\bar{B}$	0	0	0	0	... iii
$\bar{A}B$	0	0	0	1	... ii
$A\bar{B}$	1	1	1	1	
AB	1	1	1	1	

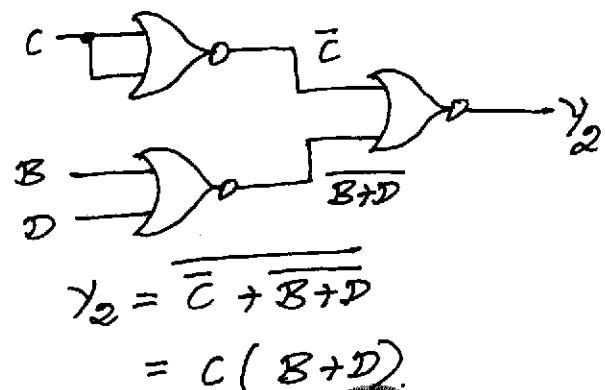
$$\therefore Y_1 = (A+C)(A+\bar{D})(A+B)$$



$$Y_2 = \prod M(0, 1, 2, 4, 5, 10) + d(8, 9, 11, 12, 13, 15)$$

	$\bar{C}\bar{D}$	$\bar{C}D$	$C\bar{D}$	CD	
$\bar{A}\bar{B}$	0	0	1	0	... i
$\bar{A}B$	0	0	1	1	
$A\bar{B}$	x	x	x	1	
AB	x	x	x	0	... ii

$$\therefore Y_2 = (C)(B+D)$$



Method 3: Self Study.

MAHESH PRASANNA K., VCET, PUTTUR

Problem: Give the SOP and POS circuits for -

$$Y = F(A, B, C, D) = \sum m(6, 8, 9, 10, 11, 12, 13, 14, 15).$$

Solution:

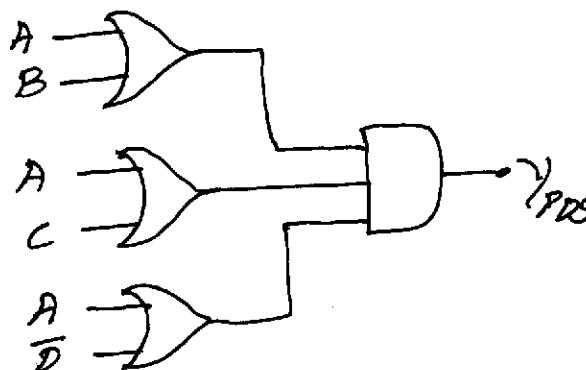
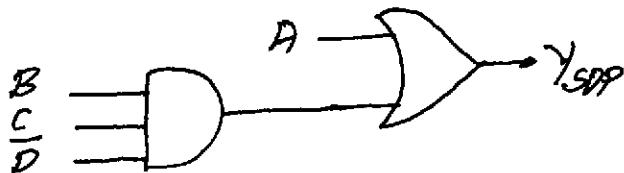
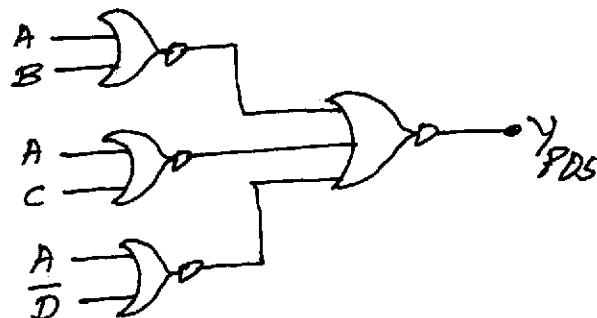
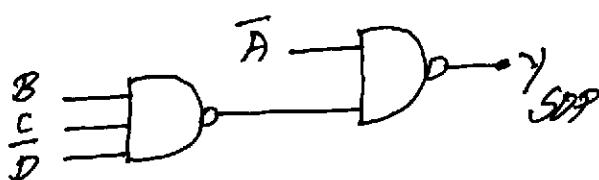
$$Y = \sum m(6, 8, 9, 10, 11, 12, 13, 14, 15)$$

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$	Y
$\bar{A}\bar{B}$	0	0	0	0	
$\bar{A}B$	0	0	0	1	
$A\bar{B}$	1	1	1	1	
AB	1	1	1	1	

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$	Y
$\bar{A}\bar{B}$	0	0	0	0	i
$\bar{A}B$	0	0	0	0	ii
$A\bar{B}$	1	1	1	1	iii
AB	1	1	1	1	

$$Y_{SOP} = A + BCD.$$

$$Y_{POS} = (A+B)(A+C)(A+\bar{D}).$$



SIMPLIFICATION BY QUINE-McCLUSKY (QM) METHOD:

Reduction of logic equation by K-map method is very simple, but has some limitations:

1. It depends on the user's ability to identify patterns that gives largest size
2. The method becomes difficult to adapt for simplification of 5 or more variables.

Quine-McClusky method is a systematic approach for logic simplification that does not have these limitations and also can easily be implemented in a digital computer.

Determination of Prime Implicants: QM method involves preparation of two tables – one determines prime implicants and the other selects essential prime implicants to get minimal expression. Prime implicants are expressions with least number of literals that represents all the terms given in a truth table. Prime implicants are examined to get essential prime implicants for a particular expression that avoids any type of duplication.

Consider the min-term expression $Y = \sum m(0, 1, 2, 3, 10, 11, 12, 13, 14, 15)$. The following truth table can be written based on the given min-term expression:

A	B	C	D	Y	Stage 1	Stage 2	Stage 3
					ABCD	ABCD	ABCD
0	0	0	0	1	0000 (0)	000- (0,1)	00-- (0,1,2,3)
0	0	0	1	1	-	00-0 (0,2)	-00- (0,2,1,3)
0	0	1	0	1	0011 (1)	-	-
0	0	1	1	1	-	00-1 (1,3)	-
0	1	0	0	0	0100 (2)	00-1 (1,3)	-
0	1	0	1	0	-	-01- (2,3,10,11)	-
0	1	1	0	0	0110 (3)	-01- (2,10,3,11)	-
0	1	1	1	0	-	-	-
1	0	0	0	0	1000 (4)	-	-
1	0	0	1	0	-	-011 (3,11)	-01- (10,14,11,15)
1	0	1	0	1	1011 (11)	101- (10,11)	11-- (12,13,14,15)
1	0	1	1	1	-	-110 (10,14)	-11- (12,14,13,15)
1	1	0	0	1	1100 (4)	110- (12,13)	-
1	1	0	1	1	-	-110 (12,14)	-
1	1	1	0	1	1110 (5)	-	-
1	1	1	1	1	-	-111 (11,15)	-

In Stage 1 of the process, we find out all the terms that gives output 1 from truth table. Put them in different groups depending on how many 1 input variable combinations (ABCD) have. For example, first group has no 1 in input combination, second group has only one 1, third two 1s, and fourth four 1s. We also write decimal equivalent of each combination to their right for convenience.

In Stage 2, we first try to combine first and second group of Stage 1, on a member to member basis. The rule is to see if only one binary digit is differing between two members and we mark that position by “—“. This means corresponding variable is not required to represent those members. Thus (0) of first group combine with (1) of second group to form (0, 1) in Stage 2 and can be represented by $A'B'C'$ (000—). The logic of this representation comes from the fact that, min-term $A'B'C'D'$ (0000) and $A'B'C'D$ (0001) can be combined as $A'B'C'(D'+D) = A'B'C'$. We proceed in the same manner to find rest of the combinations in successive groups of Stage 1 and table them. All the members of particular stage, which finds itself in at least one combination of next stage, are tick marked.

In Stage 3, we combine members of different groups of Stage 2 in a similar way. Now, it will have two “—“ elements in each combination. This means each combination requires two literals to represent it. For example, (0, 1, 2, 3) is represented by $A'B'(0\ 0\ -\ -)$. There are three other groups in Stage 3: (2, 10, 3, 11) represented by $B'C$, (10, 14, 11, 15) by AC and (12, 13, 14, 15) by AB . Note that, (0, 2, 1, 3), (10, 11, 14, 15) and (12, 14, 13, 15) get represented by $A'B$, AC and AB respectively and do not give any new term.

There is no Stage 4 for this problem, as no two members of Stage 3 has only one digit changing among them. This completes the process of determination of prime implicants. The rule is all the terms that are not ticked at any stage is treated as prime implicants for that problem.

Selection of Prime Implicants: Now, we try to select essential prime implicants and remove redundancy or duplication among them. For this, we prepare a Table as shown below:

-	0	1	2	3	10	11	12	13	14	15
$\bar{A}\bar{B} (0,1,2,3)$	✓	✓	✓	✓						
$\bar{B}C (2,3,10,11)$			✓	✓	✓	✓				
$AC (10,11,14,15)$					✓	✓			✓	✓
$AB (12,13,14,15)$							✓	✓	✓	✓

Here, row lists all the prime implicants and columns lists all min-terms. The cross point of a row and column is ticked if the term is covered by corresponding prime implicants.

Now, find minimum number of prime implicants that covers all the min-terms. We find $A'B'$ and AB cover terms that are not covered by others and they are essential prime implicants. $B'C$ and AC among themselves cover 10, 11 which are not covered by others. So, one of them has to be included in the list of essential prime implicants.

Hence, we get;
$$Y = \bar{A}\bar{B} + \bar{B}C + AB \quad \text{or} \quad Y = \bar{A}\bar{B} + AC + AB$$

Homework: Solve the above problem by using K-map method.

Solution:

Homework: Give simplified logic equation of $Y = \sum m(2, 6, 7)$ by Quine-McClusky method.

Solution:

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

Stage 1			Stage 2		
ABC			ABC		
010 (2) ✓			-10 (2,6)		
0	1	0	1	0	1
1	0	0	1	1	— (6,7)
1	1	0	1	1	(7)

—	2	6	7
$B\bar{C}(2,6)$	✓	✓	
$A\bar{B}(6,7)$		✓	✓

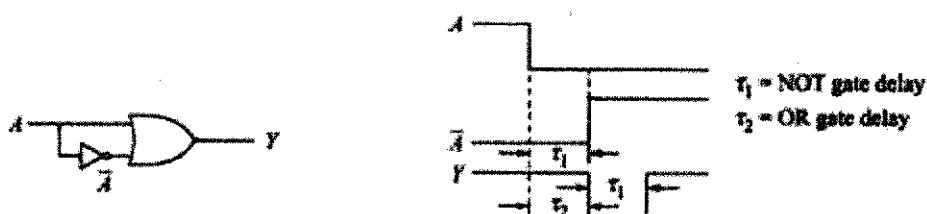
$$\therefore Y = AB + B\bar{C}.$$

HAZARDS AND HAZARD COVERS:

We have discussed various simplification techniques that give minimal expression for a logic equation, which in turn requires minimum hardware for realization. But, due to some practical problems, in certain cases, we may prefer to include more terms than given by simplification techniques. The discussion so far considered gates generating outputs instantaneously. But, practical circuits always offer finite propagation delay, though very small, in nanoseconds order.

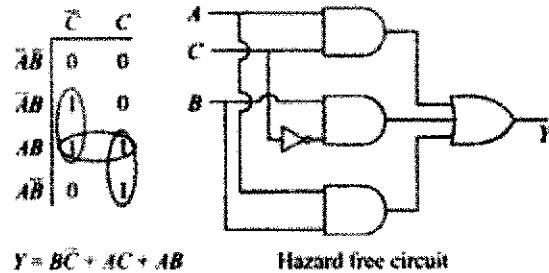
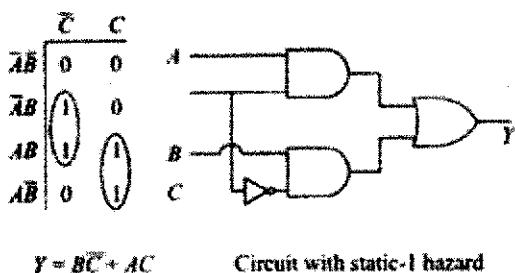
Sattic-1 Hazard:

This type of hazard occurs when $Y = A + A'$ type of situation appear for a logic circuit and when A makes a transition $1 \rightarrow 0$. An $A + A'$ condition should always generate 1 at the output (static-1). But, the NOT gate output (as shown in the following Fig) takes finite time to become 1 following $1 \rightarrow 0$ transition of A. Thus for the OR gate there are two zeros appearing at its input for a small duration, resulting a 0 at its output. The width of this zero is in nanoseconds and is called a *glitch*.



Static-1 Hazard

Cover Static-1 Hazard: Refer to the K-map shown in the following Fig; represented by $Y = BC' + AC$. The corresponding circuit is also shown in the following Fig. For this circuit, if input $B = 1$, $A = 1$ and C makes a transition $1 \rightarrow 0$; the output shows glitch.



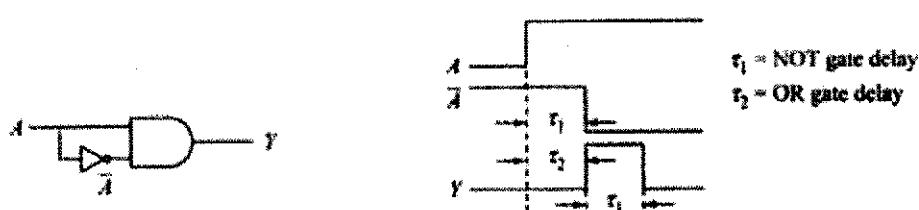
Static-1 Hazard & Its Cover

Consider another grouping for the same K-map (as shown in the above Fig). This includes one additional AND term, and now, output $Y = BC' + AC + AB$. The corresponding circuit diagram is also given. This circuit requires more hardware, but it is hazard free. The additional term AB ensures $Y = 1$ for $B = 1$, $A = 1$ and C makes a transition $1 \rightarrow 0$, does not affect output.

NOTE: A NAND gate with A and A' connected at its input for certain input combination will give static-1 hazard when A makes a transition $0 \rightarrow 1$ and requires hazard cover.

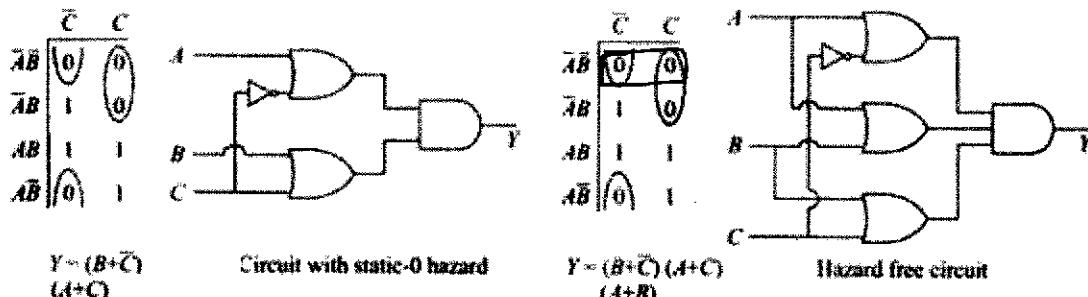
Static-0 Hazard:

This type of hazard occurs when $Y = A \cdot A'$ type of situation appear for a logic circuit and when A makes a transition $0 \rightarrow 1$. An $A \cdot A'$ condition should always generate 0 at the output (static-0). But, the NOT gate output (as shown in the following Fig) takes finite time to become 0 following $0 \rightarrow 1$ transition of A . Thus for the AND gate there are two ones appearing at its input for a small duration, resulting a 1 at its output.



Static-0 Hazard

Cover Static-0 Hazard: Refer to the K-map shown in the following Fig; POS is given by $Y = (B + C)(A + C')$. The corresponding circuit is also shown in the following Fig. For this circuit, if input B = 0, A = 0 and C makes a transition 0 → 1; there will be static-0 hazard occurring at output.



Static-0 Hazard & Its Cover

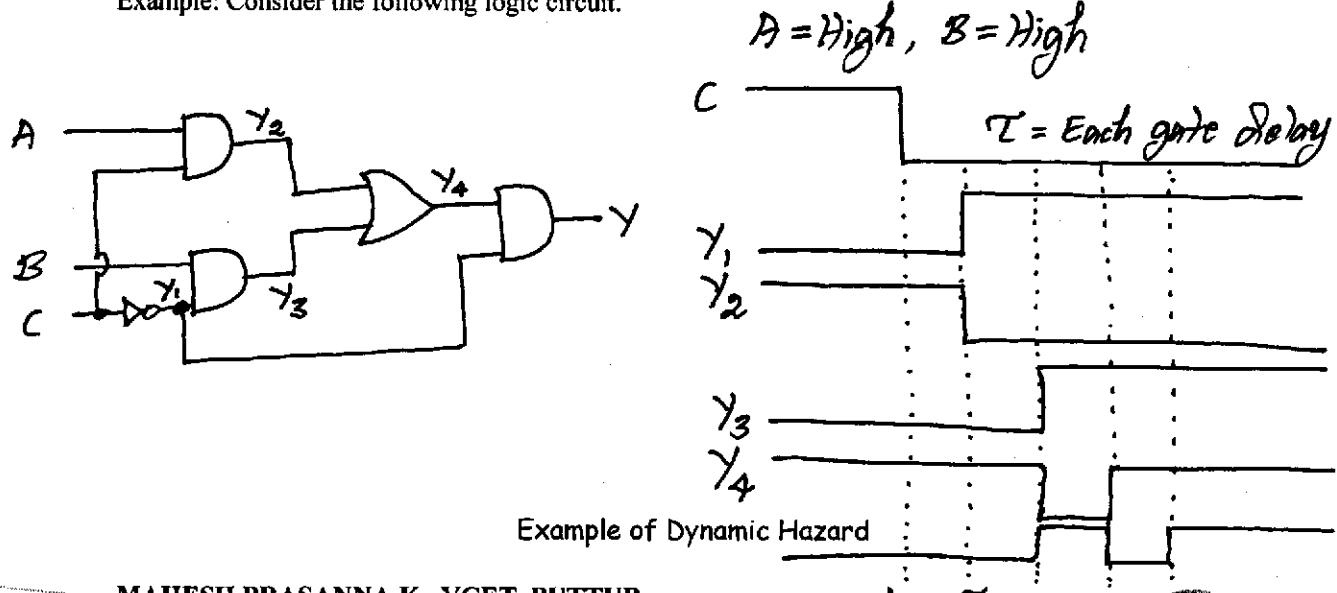
Consider another grouping for the same K-map (as shown in the above Fig). This includes one additional OR term, and now, output $Y = (B + C)(A + C')(A + B)$. The corresponding circuit diagram is also given. This circuit requires more hardware, but it is hazard free. The additional term $A + B$ ensures $Y = 0$ for $B = 0, A = 0$ and C makes a transition $0 \rightarrow 1$, does not affect output.

NOTE: A NOR gate with A and A' connected at its input for certain input combination will give static-0 hazard when A makes a transition 1 → 0 and requires hazard cover.

Dynamic Hazard:

Dynamic hazard occurs when circuit output makes multiple transitions before it settles to a final value, while the logic equation asks for only one transition. For example, an output transition designed as $1 \rightarrow 0$, may give $1 \rightarrow 0 \rightarrow 1 \rightarrow 0$ when such hazard occurs and a $0 \rightarrow 1$, can behave like $0 \rightarrow 1 \rightarrow 0 \rightarrow 1$.

Example: Consider the following logic circuit.



ANALOG AND DIGITAL ELECTRONICS

This logic circuit can be written in the equation form as $Y = (AC + BC')C$. This shows dynamic hazard; for $AB = 11$, and C makes a transition $1 \rightarrow 0$ (as shown in the waveform). The hazard can be prevented by using an additional two input AND gate fed by input A and B and replacing the two input OR gate by a three input OR gate.

HDL IMPLEMENTATION MODELS:

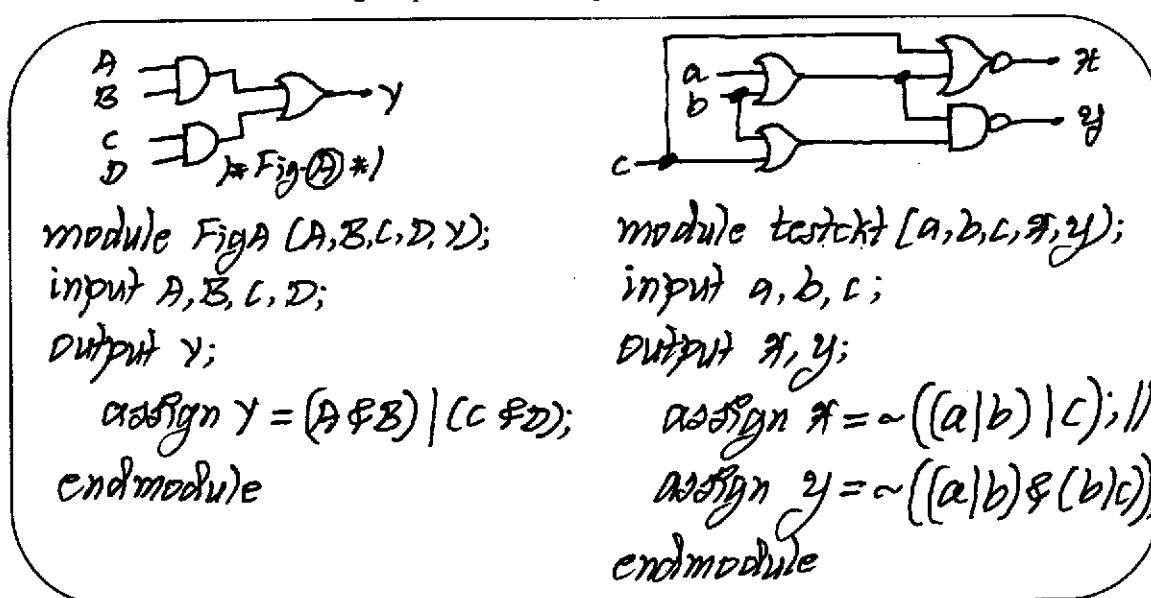
Structural modeling, though convenient, consumes more space in describing a circuit, and is unsuitable for large, complex design.

Dataflow Modeling:

Verilog provides a keyword *assign* and a set of operators (given in the following Table) to describe a circuit through its function. All *assign* statements are concurrent and continuous.

Relational Operation	Symbol	Bit-wise Operation	Symbol
Less than	<	Bit-wise NOT	\sim
Less than or equal to	\leq	Bit-wise OR	$ $
Greater than	$>$	Bit-wise AND	$\&$
Greater than or equal to	\geq	Bit-wise Ex-OR	\wedge
Not equal to	\neq	Arithmetic Operation	Symbol!
Logical Operation	Symbol	Binary addition	+
Logical NOT	!	Binary subtraction	-
Logical OR	\parallel	Binary multiplication	*
Logical AND	$\&\&$	Binary division	/

Data flow model resembles a logic equation and thus gives a more crisp representation.



Behavioral Modeling:

In a behavioral model, statements are executed sequentially following algorithmic description. It always uses *always* keyword followed by a sensitivity list. The procedural statements following *always* are executed only if any variable within sensitivity list changes its value. Procedure assignment or output variables within *always* must be register type, defined by *reg*.

```
module FigA (A,B,C,D,Y);
input A,B,C,D; output Y;
reg Y;
always @ (A or B or C or D) // Sensitivity list
if (A==1) && (B==1) // If A=1 & B=1; Y=1.
    Y=1;
else if (C==1) && (D==1) // If C=1 & D=1; Y=1.
    Y=1;
else Y=D; // For all other combination of A,B,C,D.
endmodule
```

Problem: Realize $Y = \sum m(0, 1, 2, 4, 5, 6, 8, 9, 10, 12, 13)$ using dataflow model.

Solution:

\bar{AB}	\bar{CD}	$\bar{C}\bar{D}$	$C\bar{D}$	$C\bar{D}$
1	1	1	0	1
1	1	0	0	1
1	1	0	0	0
1	1	0	1	1

$$Y = \bar{C} + \bar{A}\bar{D} + \bar{B}\bar{D}.$$
NOTE:

- Structural \leftrightarrow Gates / Circuits
- Dataflow \leftrightarrow assign / Expression
- Behavioral \leftrightarrow always / Truth table

```
module X-Map (A,B,C,D,Y);
input A,B,C,D; output Y;
assign Y = ~C | (~A & ~D) | (~B & ~D);
endmodule
```

Problem: Find the minimal sums for -

a) $f_1(a, b, c) = \sum(1, 3, 4, 5, 6, 7)$

b) $f_2(a, b, c) = \prod(2, 4, 7)$

Solution:

f_1	a	b	c	f_1
$\bar{a}b$	0	0	0	1
00	0	0	1	1
01	0	1	0	1
11	1	1	1	1
10	1	0	1	1

$$\therefore f_1 = c + a.$$

$$\begin{aligned} \therefore f_2 &= \bar{a}\bar{b} + \bar{a}c \\ &\quad + \bar{b}c + ab\bar{c}. \end{aligned}$$

f_2	a	b	c	f_2
$\bar{a}b$	0	0	0	1
00	1	0	1	1
01	0	1	0	1
11	1	1	0	1
10	0	1	1	1

Problem: Find the minimal products for -

a) $f_1(a, b, c) = \sum(0, 1, 2, 3, 4, 6, 7)$

b) $f_2(a, b, c) = \prod(1, 4, 5)$

Solution:

f_1	a	b	c	f_1
$\bar{a}b$	0	0	0	1
00	1	1	1	1
01	1	1	0	1
11	1	1	1	1
10	1	0	0	0

$$\therefore f_1 = \bar{a} + b + \bar{c}.$$

$$\therefore f_2 = (\bar{a} + b)(a + \bar{c}).$$

f_2	a	b	c	f_2
$\bar{a}b$	1	0	0	1
00	1	1	1	0
01	1	1	0	1
11	1	1	1	1
10	0	0	0	0

Problem: Solve for the simplified Boolean expression using K-map:

a) $f_1(a, b, c, d) = \bar{a}\bar{c}d + \bar{a}cd + \bar{b}\bar{c}\bar{d} + ab\bar{c} + \bar{a}\bar{b}cd$

b) $f_2(a, b, c, d) = (a + b + \bar{d})(\bar{a} + b + \bar{d})(a + \bar{b} + \bar{c} + d)(\bar{a} + \bar{b} + \bar{c} + \bar{d})(\bar{a} + \bar{b} + \bar{c} + d)$

Solution:

f_1	$\bar{c}\bar{d}$	$\bar{c}d$	cd	cd	f_1
$\bar{a}\bar{b}$	1	1	1	1	i x
$\bar{a}b$	0	1	1	0	ii
ab	0	0	0	0	
$a\bar{b}$	1	0	1	1	iii
					iv

f_2	$\bar{c}\bar{d}$	$\bar{c}d$	cd	cd	f_2
$\bar{a}\bar{b}$	1	0	0	1	i
$\bar{a}b$	1	1	1	0	ii
ab	1	1	0	0	iii
$a\bar{b}$	0	0	1	1	iv x

MAHESH PRASANNA K., VCET, PUTTUR

$$\therefore f_1 = \bar{a}\bar{d} + \bar{b}c + \bar{b}\bar{d}.$$

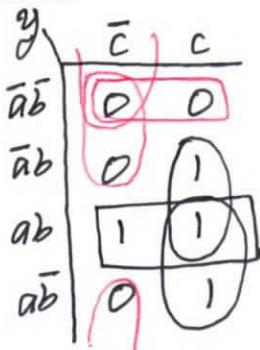
$$\therefore f_2 = (b + \bar{d})(\bar{b} + \bar{c} + d)(\bar{a} + \bar{b} + \bar{c}).$$

Problem: Design a 3-input, 1-output, minimal two-level gate combinational circuit; which has an output equal to 1 when majority of its inputs are at logic 1, and has output 0 when majority of inputs are at logic 0.

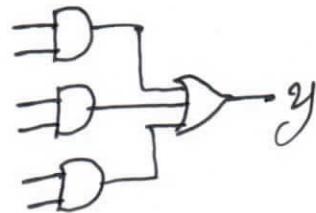
Solution:

Construct the truth table:

a	b	c	y
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



Grouping for logic 1s; $y = ab + bc + ac$



If no. of gate inputs is taken as cost; the cost of implementation is 9.

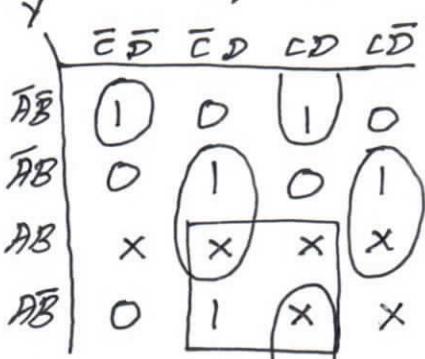
Grouping for 0s; $y = (a+b)(b+c)(a+c)$. Here also, the cost is 9.

Problem: Design a minimal sum and minimal product combinational gate circuit to generate the odd parity bit for an 8421 BCD code.

Solution: Construct the truth table, as described in the question.

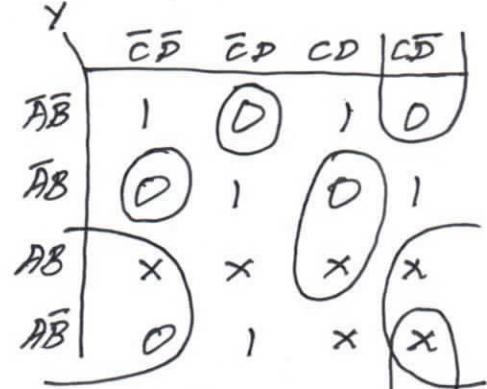
A	B	C	D	Y
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	x
1	0	1	1	x
1	1	0	0	x
1	1	0	1	x
1	1	1	0	x
1	1	1	1	x

$$\therefore Y = \sum m[0, 3, 5, 6, 9] + \sum d[10, 11, 12, 13, 14, 15]$$



$$Y_{SOP} = AD + B\bar{C}\bar{D} + B\bar{C}\bar{D} + \bar{B}CD + \bar{A}\bar{B}\bar{C}\bar{D}$$

$$Cost = 2D.$$



$$Y_{POS} = (\bar{A}+D)(B+\bar{C}+D) \\ (\bar{B}+\bar{C}+\bar{D})(A+\bar{B}+\bar{C}+\bar{D}) \\ (A+B+C+\bar{D})$$

$$Cost = 2L.$$

ANALOG AND DIGITAL ELECTRONICS

problem: Using Quine-McClusky method, simplify; $f(a, b, c, d) = \sum (3, 4, 5, 7, 10, 12, 14, 15) + \sum d(2)$.

Solution: Determination of prime implicants, considering don't care terms as 1:

State 1		Stage 2						
a	b	c	d					
0010	0	1	0	(2) ✓	001-	0	1, 3	
0100	0	1	0	(4) ✓	-010	0	2, 10	
0011	0	1	1	(3) ✓	010-	0	4, 5	
0101	0	1	1	(5) ✓	-100	0	4, 12	
1010	1	0	0	(10) ✓				
1100	1	1	0	(2) ✓	0-11	0	3, 7	
0111	0	1	1	(7) ✓	01-	1	5, 8	
1110	1	1	0	(14) ✓	-1-0	0	10, 14	
1111	1	1	1	(15) ✓	11-0	0	12, 14	
					-111	0	7, 15	
					111-	0	14, 15	

There is no stage 3 here., as no two members of stage 2 has only one digit changing among them.

Selection of essential prime implicants: (Please note, don't care is not considered here)

-	3	4	5	7	10	12	14	15	Row
$\bar{A}\bar{B}C (2,3)$	✓								Q
$\bar{B}CD (2,10)$						✓			R
$\bar{A}\bar{B}\bar{C} (4,5)$		✓	✓						S
$B\bar{C}D (4,12)$		✓				✓			T
$\bar{A}CD (3,7)$	✓			✓					U
$\bar{A}BD (5,7)$			✓	✓					V
$A\bar{C}\bar{D} (0,14)$					✓		✓		W
$A\bar{B}\bar{D} (12,14)$						✓	✓		X
$BCD (7,15)$				✓				✓	Y
$ABC (14,15)$							✓	✓	Z

o Row U dominates row Q

o Row W dominates row R

o Columns with min-terms 3 & 10 and 7 & 14 have one tick (mark) each. The corresponding rows U & W are essential rows. Hence, minimal sum = U + W +

- Delete row U, row W, columns 3, 10, 7, & 14. Now, the table reduces to:

-	4	5	12	15	Row
$\bar{A}B\bar{C}$ (4, 5)	✓	✓			S
$B\bar{C}\bar{D}$ (4, 12)	✓		✓		T
$\bar{A}BD$ (5, 7)		✓			V
$AB\bar{D}$ (12, 14)			✓		X
BCD (7, 15)				✓	Y
$A\bar{B}C$ (14, 15)				✓	Z

- Row S dominates row V
- Row T dominates row X
- Rows Y and Z are equal. Hence, only one row out of Y and Z need to be considered for minimal sum
- Rows S, T, and Y are considered for minimal terms. Delete row V, row X, and row Z. Therefore, minimal sum = U + W + S + T + Y.

Hence, $f = \bar{A}CD + A\bar{C}\bar{D} + \bar{A}B\bar{C} + B\bar{C}\bar{D} + BCD$.

Problem: Simplify $f(a, b, c, d) = \sum(2, 3, 4, 5, 13, 15) + \sum d(8, 9, 10, 11)$ using EVM techniques, taking –

- The variable in the last significant position as MEV
- The variables c and d as MEVs.

Solution:

a	b	c	d	f	MEV: d	MEVs: c & d
0	0	0	0	0		
0	0	0	1	0	0	
0	0	1	0	1		c
0	0	1	1	1	1	
0	1	0	0	1		
0	1	0	1	1	1	
0	1	1	0	0		\bar{c}
0	1	1	1	0	0	
1	0	0	0	x		
1	0	0	1	x		
1	0	1	0	x		
1	0	1	1	x		
1	1	0	0	0	d	d
1	1	0	1	1	d	d

$$\begin{aligned}
 f_a &= ad + \bar{b}c + \bar{a}\bar{b}\bar{c} \\
 f_b &= bc + \bar{b}\bar{c} + a\bar{b}\bar{c}
 \end{aligned}$$

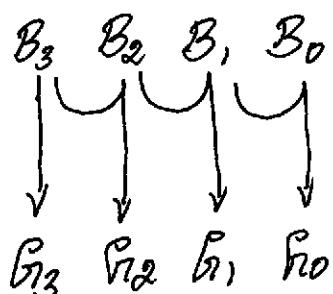
1	1	0	0	d	
1	1	1	1	1	

Try these: Design a minimal sum combinational circuit to –

- Find the 9s complement of BCD numbers
- Convert BCD to Excess-3
- Multiply two 2-bit numbers
- Output a 1 when an illegal BCD code occurs
- Output the 2s complement of a 4-bit binary number.

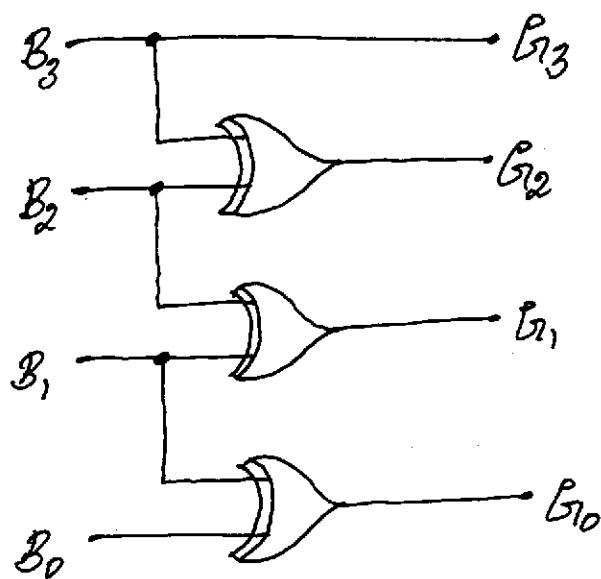
Problem: Design Binary-to-Gray Code Converter.

Solution:



$$\begin{aligned}
 G_3 &= B_3 \\
 G_2 &= B_3 \oplus B_2 \\
 G_1 &= B_2 \oplus B_1 \\
 G_0 &= B_1 \oplus B_0
 \end{aligned}$$

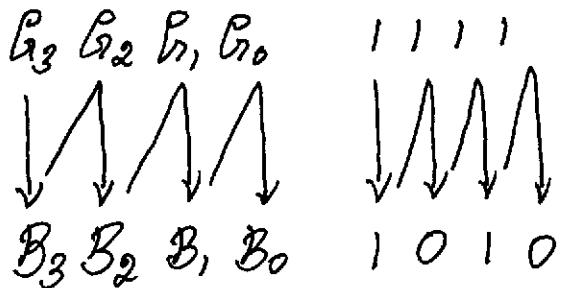
Binary Code				Gray Code			
B3	B2	B1	B0	G3	G2	G1	G0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	0	1	0
0	1	0	1	0	0	1	1
0	1	1	0	0	0	1	0
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	0
1	1	0	0	1	0	1	0



1	0	1	1	0	1	1
1	1	1	0	1	0	0
1	1	1	1	1	0	0

Problem: Design Gray-to-Binary Code Converter.

Solution:

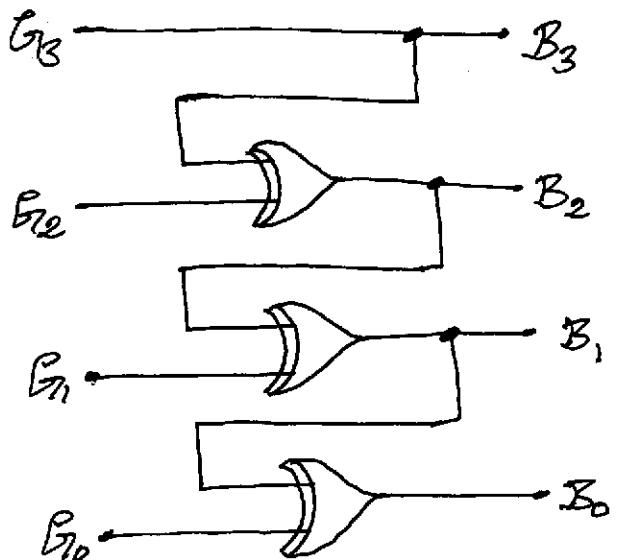


$$B_3 = G_3$$

$$B_1 = B_2 \oplus G_1$$

$$B_2 = B_3 \oplus G_2$$

$$B_0 = B_1 \oplus G_0$$



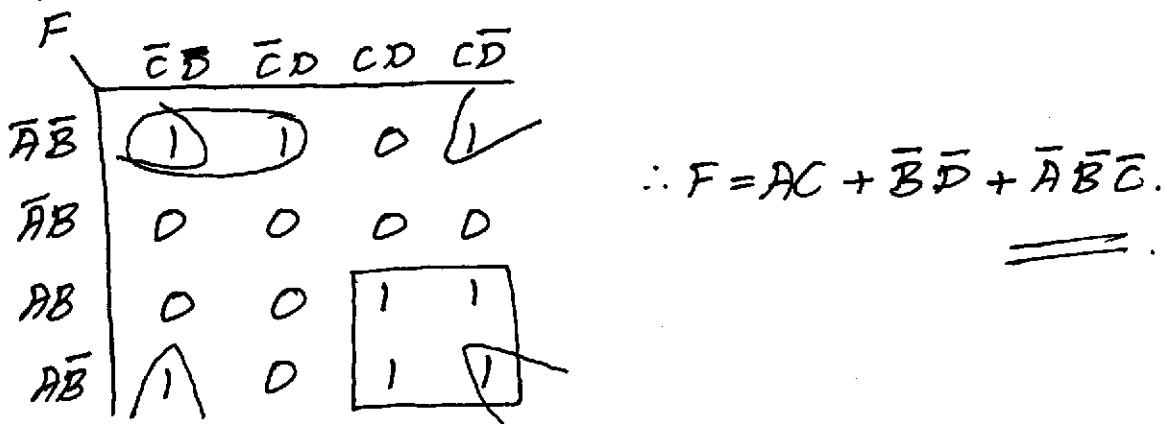
Gray Code				Binary Code			
G3	G2	G1	G0	B3	B2	B1	B0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	1	0	0	1	0
0	0	1	0	0	0	1	1
0	1	1	0	0	1	0	0
0	1	1	1	0	1	0	1
0	1	0	1	0	1	1	0
0	1	0	0	0	1	1	1
1	1	0	0	1	0	0	0
1	1	0	1	1	0	0	1
1	1	1	1	1	0	1	0
1	1	1	0	1	0	1	1
1	0	1	0	1	1	0	0
1	0	1	1	1	1	0	1
1	0	0	1	1	1	1	0
1	0	0	0	1	1	1	1

Gray Code				Binary Code			
G3	G2	G1	G0	B3	B2	B1	B0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	0	1
0	1	0	0	0	1	1	1
0	1	0	1	0	1	1	0
0	1	1	0	0	1	0	0
0	1	1	1	0	1	0	1
1	0	0	0	1	1	1	1
1	0	0	1	1	1	1	0
1	0	1	0	1	1	0	0
1	0	1	1	1	1	0	1
1	1	0	1	1	0	0	1
1	1	1	0	1	0	1	1
1	1	1	1	1	0	1	0

problem: Simplify, using K-map method Quine Mc-Clusky method: $F = \sum m(0, 1, 2, 8, 10, 11, 14, 15)$.

Solution:

K-map method:



Quine Mc-Clusky method:

A	B	C	D	F	Stage 1	Stage 2	Stage 3
					ABCD	ABCD	ABCD
0	0	0	0	1	<u>0000 (0)</u>	<u>000-</u> <u>(0,1)</u>	<u>-0-0 (0,2,8,10)</u>
0	0	0	1	1	<u>0001 (1)</u>	<u>00-D</u> <u>(0,2)</u>	<u>-0-D (0,8,2,10)</u>
0	0	1	0		<u>0010 (2)</u>	<u>-000 (0,8)</u>	
0	0	1	1	0	<u>1000 (8)</u>		
0	1	0	0	0	<u>1010 (10)</u>	<u>-0)0 (2,10)</u>	
0	1	0	1	0	<u>1011 (11)</u>	<u>10-D (8,10)</u>	<u>1-) - (10,11,14,15)</u>
1	0	0	0	1	<u>1110 (14)</u>	<u>10)- (10,11)</u>	<u>1-) - (10,14,11,15)</u>
1	0	0	1	0	<u>1111 (15)</u>	<u>1-)D (10,14)</u>	
1	0	1	1	1		<u>1-)1 (11,15)</u>	
1	1	0	0	0		<u>11)- (14,15)</u>	
1	1	0	1	0			
1	1	1	0	1			
1	1	1	1	1			

ANALOG AND DIGITAL ELECTRONICS

-	0	1	2	8	10	11	14	15
$\bar{A}\bar{B}E(0,1)$	✓	✓						
$\bar{B}\bar{D}(0,2,8,10)$	✓		✓	✓	✓			
$AC(10,11,14,15)$					✓	✓	✓	✓

Therefore, $F = \underline{\underline{AC + \bar{B}\bar{D} + \bar{A}\bar{B}C}}$.

By: MAHESH PRASANNA K.,

DEPT. OF CSE, VCET.

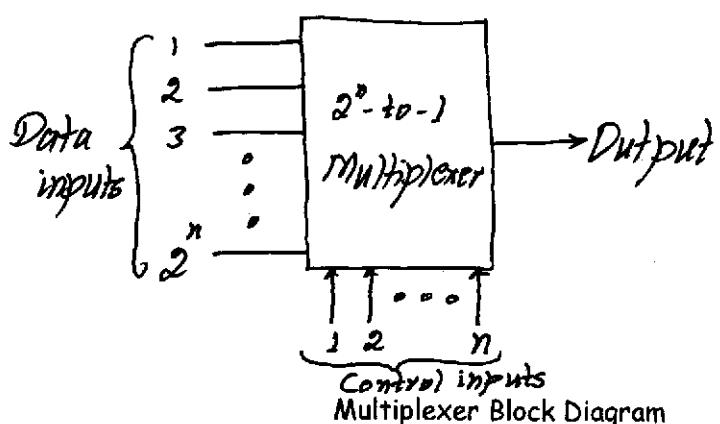
MODULE - 3

DATA PROCESSING CIRCUITS & FLIP-FLOPS

DATA PROCESSING CIRCUITS

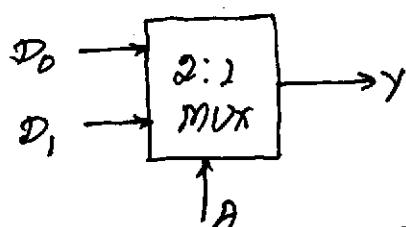
MUXES (MUX):

Multiplex means *many into one*. A multiplexer is a circuit with many inputs but only one output. By applying control signals, we can steer any input to the output. Thus, it is also called a *data selector* and the control inputs are termed *select inputs*. The following Fig illustrates the general idea.

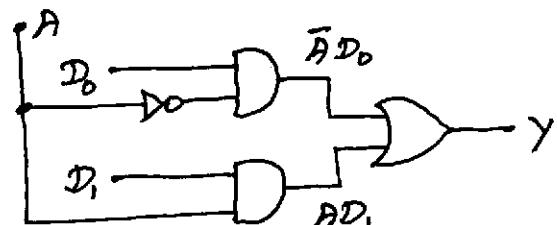


The block diagram has 2^n input signals, n control signals, and 1 output signal.

2-to-1 MUX:

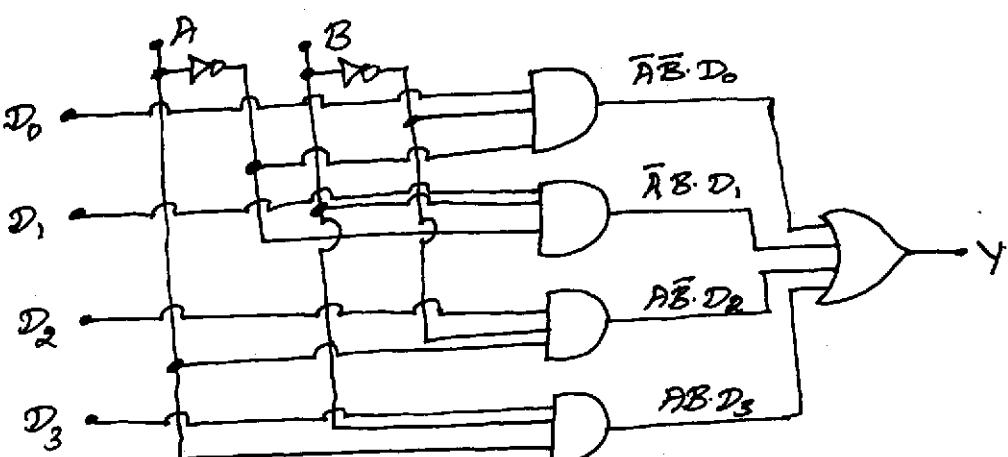
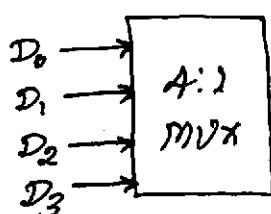


A	y
0	D_0
1	D_1



$$y = \bar{A}D_0 + AD_1.$$

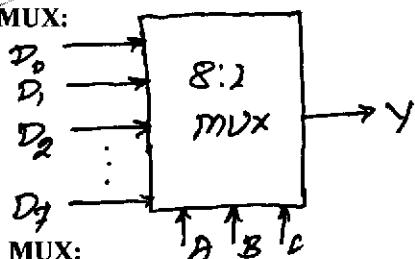
4-to-1 MUX:



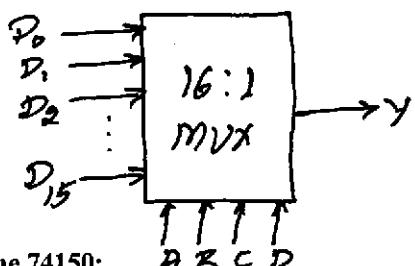
$$y = \bar{A}\bar{B}D_0 + \bar{A}BD_1 + A\bar{B}D_2 + AB\bar{D}_3.$$



16-to-1 MUX:

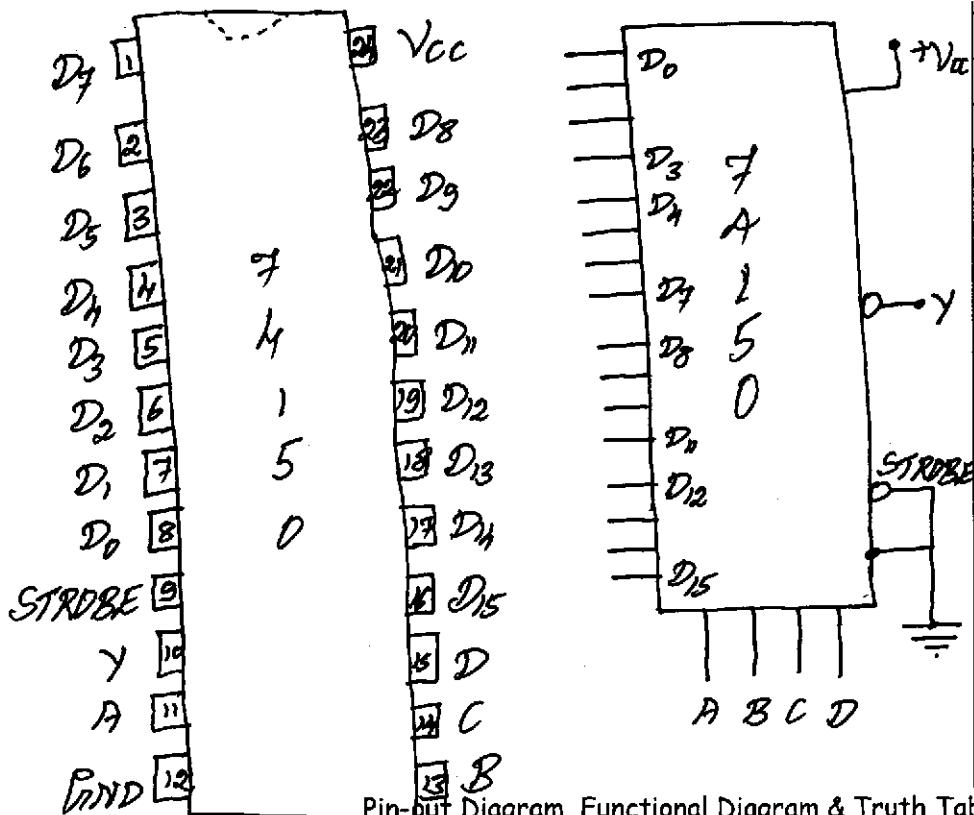


16-to-1 MUX:



The 74150:

The 74150 is a 16-to-1 TTL multiplexer with the pin diagram as shown below. Pins 1 to 8 and 16 to 23 are input data bits D_0 to D_{15} . Pins 11, 13, 14, and 15 are control bits ABCD. Pin 10 is the output; and it equals the complement of the selected data bit. Pin 9 is for STROBE, an input signal that enables or disables the multiplexer. A low STROBE enables the multiplexer, so that, the output Y equals the complement of input data bit; $Y = \bar{D}_n$, where n is the decimal equivalent of ABCD.



STROBE	A	B	C	D	Y
L	L	L	L	L	\bar{D}_0
L	L	L	L	H	\bar{D}_1
L	L	L	H	L	\bar{D}_2
L	L	L	H	H	\bar{D}_3
L	L	H	L	L	\bar{D}_4
L	L	H	L	H	\bar{D}_5
L	L	H	H	L	\bar{D}_6
L	L	H	H	H	\bar{D}_7
L	H	L	L	L	\bar{D}_8
L	H	L	L	H	\bar{D}_9
L	H	L	H	L	\bar{D}_{10}
L	H	L	H	H	\bar{D}_{11}
L	H	H	L	L	\bar{D}_{12}
L	H	H	L	H	\bar{D}_{13}
L	H	H	H	L	\bar{D}_{14}
L	H	H	H	H	\bar{D}_{15}
H	X	X	X	X	H

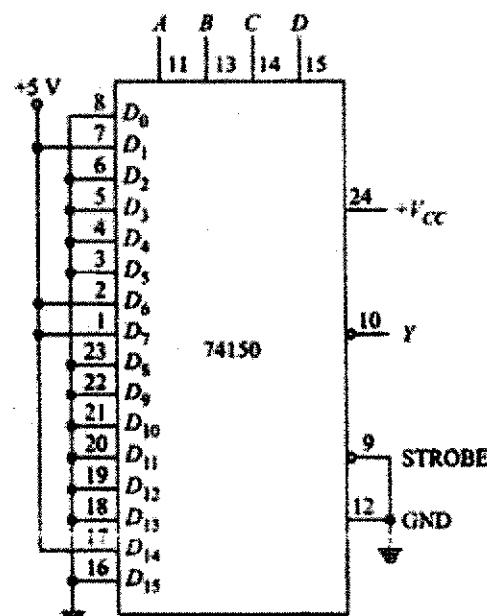
Pin-out Diagram, Functional Diagram & Truth Table of 74150

Multiplexer Logic: A digital design usually begins with a truth table. The problem is to come up with a logic circuit that has the same truth table. We have two standard methods for implementing a truth table – the SOP and the POS solution. The third method is the *multiplexer solution*.

Problem: Implement $Y(A, B, C, D) = \sum m(0, 2, 3, 4, 5, 8, 9, 10, 11, 12, 13, 15)$ using 74150 multiplexer.

Solution:

STROBE	A	B	C	D	Y
0	0	0	0	0	1
0	0	0	0	1	0
0	0	0	1	0	1
0	0	0	1	1	1
0	0	1	0	0	1
0	0	1	0	1	1
0	0	1	1	0	0
0	0	1	1	1	0
0	1	0	0	0	1
0	1	0	0	1	1
0	1	0	1	0	1
0	1	1	0	0	1
0	1	1	0	1	1
0	1	1	1	0	0
0	1	1	1	1	1
1	X	X	X	X	1



Bubbles on Signal Lines:

Data sheets often show inversion bubbles on some of the signal lines. (Notice the bubble on Pin 10 of 74150. This bubble is reminder that the output is the complement of the selected data bit). Notice the bubble on the Pin 9, STROBE input. The multiplexer is active (enabled) when the STROBE is low, and is inactive (disabled) when it is high. Because of this, the STROBE is called an *active-low signal*; it causes something to happen when it is low rather than when it is high.

Universal Logic Circuit:

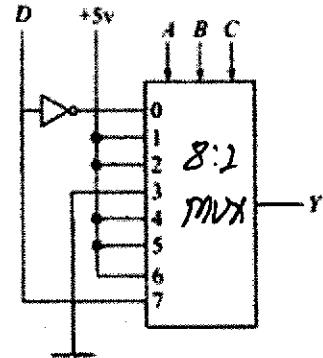
Multiplexer, sometimes, is called *universal logic circuit*; because a 2^n -to-1 multiplexer can be used as a design solution for any n variable truth table. We have seen the realization of 4 variable truth table by 16-to-1 multiplexer; now we will see the realization of same expression using 8-to-1 multiplexer.

Problem: Implement $Y(A, B, C, D) = \sum m(0, 2, 3, 4, 5, 8, 9, 10, 11, 12, 13, 15)$ using 8-to-1 multiplexer.

Solution: We follow a procedure that is similar to the one that we adopted in Entered Variable Map method.

A	B	C	D	Y	8-to-1 MUX Data Inputs
0	0	0	0	1	
0	0	0	1	0	\bar{D}
0	0	1	0	1	
0	0	1	1	1	
0	1	0	0	1	
0	1	0	1	1	
0	1	1	0	0	0
0	1	1	1	0	
1	0	0	0	1	
1	0	0	1	1	
1	0	1	0	1	
1	0	1	1	1	
1	1	0	0	1	
1	1	0	1	1	
1	1	1	0	0	D
1	1	1	1	1	

A	B	C	8-to-1 MUX Data Inputs
0	0	0	\bar{D}
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	D



$$D_0 = \bar{D}$$

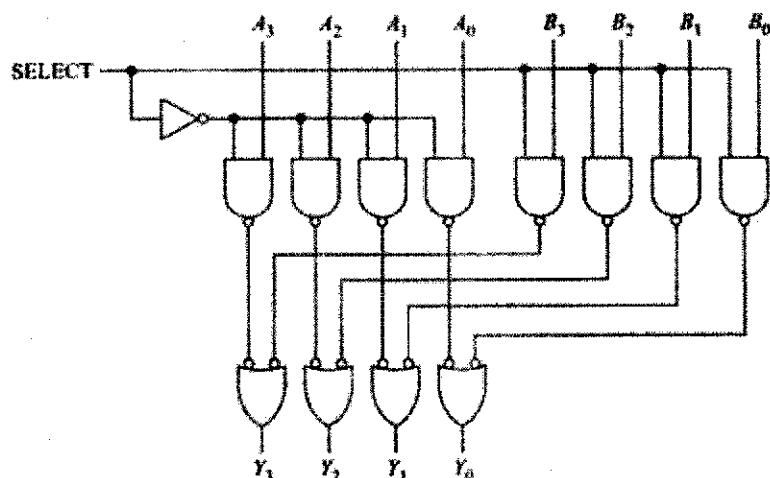
$$D_1 = D_2 = D_3 = D_5 = D_6 = 1$$

$$D_4 = 0$$

$$D_7 = D.$$

Nibble Multiplexer:

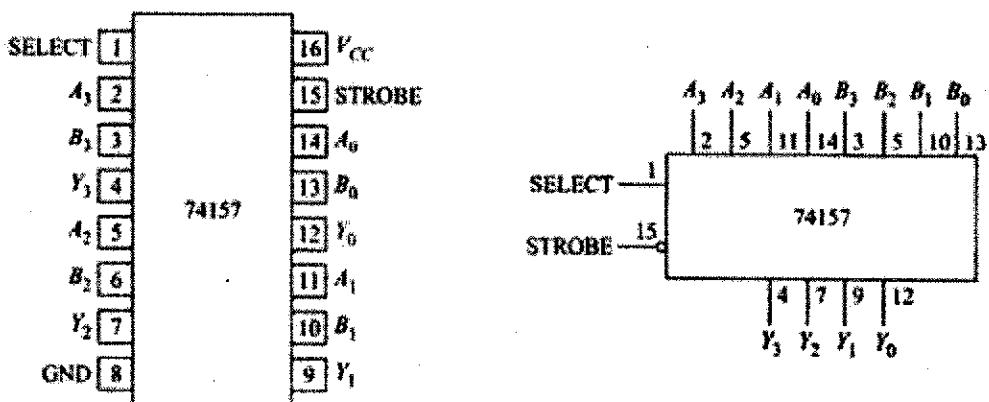
Sometimes, we may want to select one of two input nibbles. In this case, we can use a nibble multiplexer, as shown below:



Nibble Multiplexer

The input nibble on the left is $A_3A_2A_1A_0$, and the one on the right is $B_3B_2B_1B_0$. The control signal labeled SELECT determines which input nibble is transmitted to the output. When SELECT is low, the four NAND gates on the left are activated; therefore, $Y_3Y_2Y_1Y_0 = A_3A_2A_1A_0$. When SELECT is high, the four NAND gates on the right are activated; therefore, $Y_3Y_2Y_1Y_0 = B_3B_2B_1B_0$.

The 74157: The following Fig shows the pin-out diagram and functional diagram of a 74157, a nibble multiplexer. The STROBE input must be low for the multiplexer to properly operate. When the STROBE is high, the multiplexer is inoperative.



Pin-out Diagram, Functional Diagram of 74157

Problem: Show how 4-to-1 multiplexer can be obtained using only 2-to-1 multiplexers.

Solution:

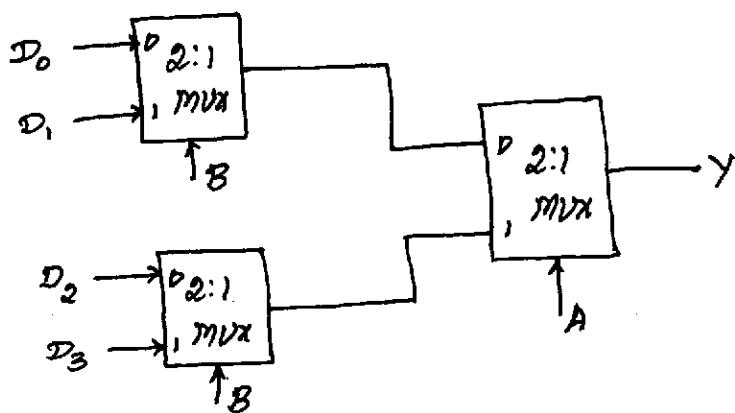
$$\text{Logic equation for 2-to-1 multiplexer: } Y = \bar{A}D_0 + AD_1 \rightarrow ①$$

$$\text{Logic equation for 4-to-1 multiplexer: } Y = \bar{A}\bar{B}D_0 + \bar{A}BD_1 + A\bar{B}D_2 + AB\bar{D}_3$$

$$Y = \bar{A}[\bar{B}D_0 + BD_1] + A[\bar{B}D_2 + BD_3] \rightarrow ②$$

Comparing above equations, we need two 2-to-1 multiplexers to realize two bracketed terms, where B serves as select input. The output of these two multiplexers can be sent to a third multiplexer as data inputs, where A serves as select input and we get the 4-to-1 multiplexer.

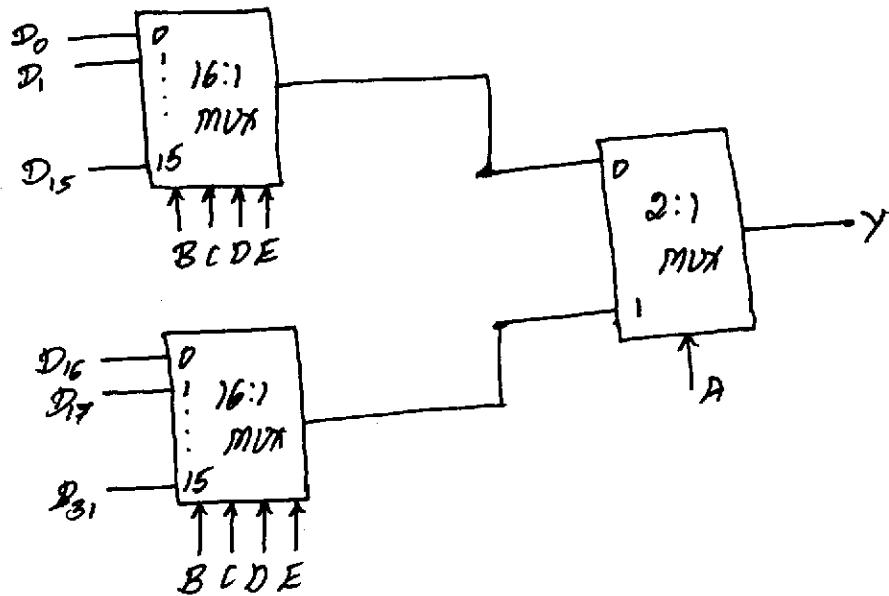
The following Fig shows the circuit diagram:



Problem: Design a 32-to-1 multiplexer using two 16-to-1 multiplexer and one 2-to-1 multiplexer.

Solution:

The circuit diagram is shown in the following Fig. A 32-to-1 multiplexer required 5 ($\log_2 32$) select lines (say, ABCDE). The lower four select lines (BCDE) chose 16-to-1 multiplexer outputs. The 2-to-1 multiplexer chooses one of the output of two 16-to-1 multiplexers, depending on the 5th select line (A).



Problem: Realize $Y = \bar{A}B + \bar{B}\bar{C} + ABC$ using an 8-to-1 multiplexer. Also, realize the same with a 4-to-1 multiplexer.

Solution: Given,

$$Y = \bar{A}B + \bar{B}\bar{C} + ABC$$

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

$$Y = \bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + \bar{A}BC + A\bar{B}\bar{C} + ABC = \sum m(0, 2, 3, 4, 7)$$

Hence, to generate the given logic function, using 8-to-1 multiplexer,

we find $D_0 = D_2 = D_3 = D_4 = D_7 = 1$ and $D_1 = D_5 = D_6 = 0$.

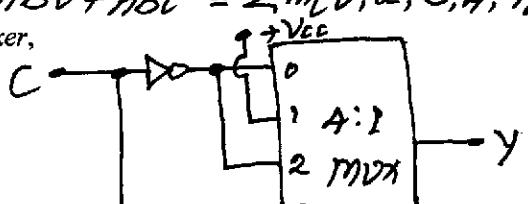
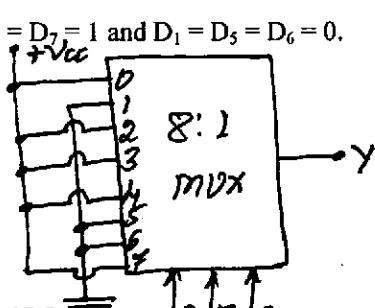
$$\text{Now, } Y = \bar{A}B + \bar{B}\bar{C} + ABC$$

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

$$Y = \bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + A\bar{B}\bar{C} + A\bar{B}C \quad \Rightarrow ABC$$

Hence, for a 4-to-1 multiplexer,

we find $D_0 = C'$, $D_1 = 1$, $D_2 = C'$, and $D_3 = C$ generates the given function.

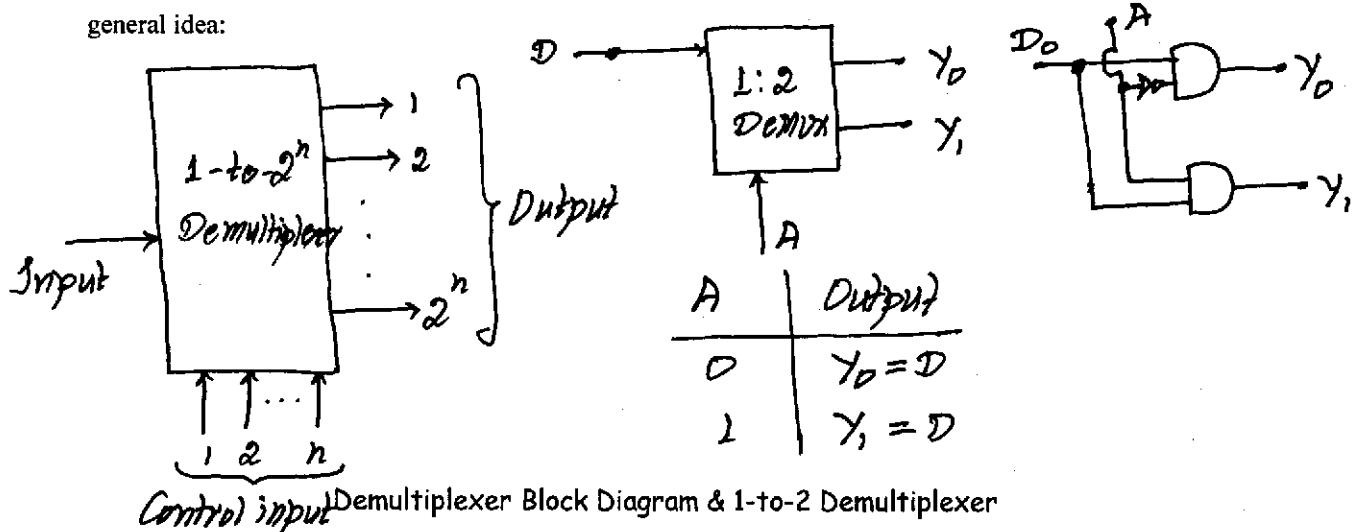


			8-to-1 MUX Data Inputs	4-to-1 MUX Data Inputs
A	B	C	$1 = D_0$	$C = D_0$
0	0	0	$0 = D_1$	
0	0	1		$1 = D_1$
0	1	0	$1 = D_2$	
0	1	1	$1 = D_3$	
1	0	0	$1 = D_4$	$\bar{C} = D_2$
1	0	1	$0 = D_5$	
1	1	0	$0 = D_6$	$C = D_3$
1	1	1	$1 = D_7$	

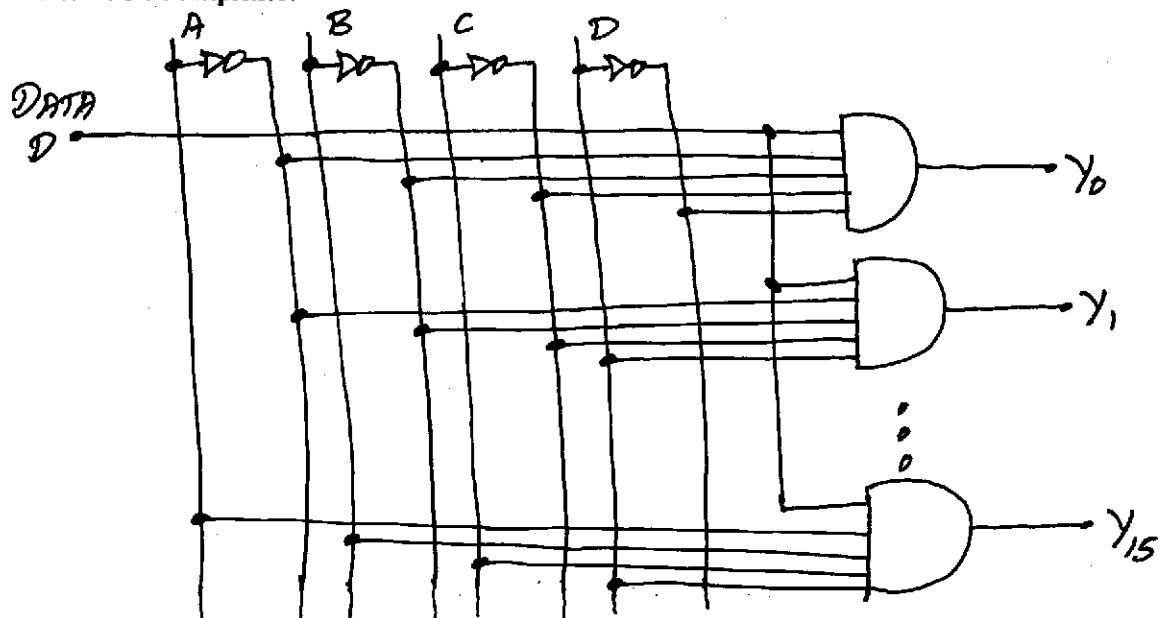
DEMULTIPLEXERS (DE-MUX):

Demultiplexer means one into many. A Demultiplexer is a logic circuit with one input and many outputs.

By applying control signals, we can steer the input signal to one of the output lines. A Demultiplexer circuit has I input signal, n control or select signals, and 2^n output signals. The following Fig shows the general idea:



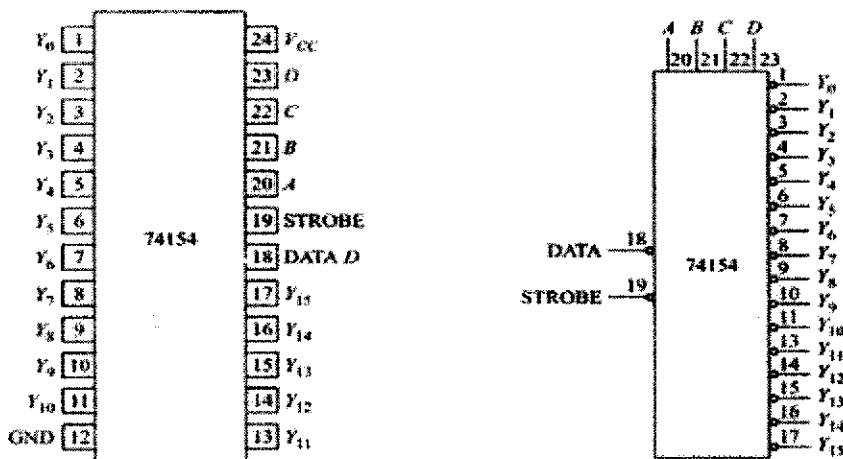
1-to-16 Demultiplexer:



1-to-16 Demultiplexer

The input bit is labeled D. This input data bit is transmitted to the data bit of the output line. When ABCD = 0000, the upper AND gate is enabled, while all other AND gates are disabled. Hence, data bit D is transmitted only to the y_0 output, giving $y_0 = D$. If D is low, y_0 is low; if D is high, y_0 is high. All other outputs are in the low state. If the control nibble is changed to ABCD = 1111, all gates are disabled, except the bottom AND gate. Then, D is transmitted only to the y_{15} output, hence, $y_{15} = D$.

The 74154: The 74154 is a 1-to-16 Demultiplexer with the pin diagram and functional diagram as shown below. STROBE is active-low signal and must be low to activate the 74154. When STROBE is high, all the output lines are high. When STROBE is low, the control input ABCD determines the output line; the DATA bit is steered to the output line, whose subscript is the decimal equivalent of ABCD. Notice that, DATA is inverted at the input (the bubble on pin 18) and again on any output (the bubble on each output line). With this double inversion, DATA passes through the 74154 unchanged.



Pin-out Diagram & Functional Diagram of 74154

ST	DA	A	B	C	D	Y ₀	Y ₁	Y ₂	Y ₃	Y ₄	Y ₅	Y ₆	Y ₇	Y ₈	Y ₉	Y ₁₀	Y ₁₁	Y ₁₂	Y ₁₃	Y ₁₄	Y ₁₅
L	L	L	L	L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	L	L	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	L	H	L	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	L	H	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	L	L	H	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	H	L	H	H	H	H	H	L	H	H	H	H	H	H	H	H	H	H
L	L	L	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H	H	H	H	H
L	L	L	H	L	L	H	H	H	H	H	H	H	L	H	H	H	H	H	H	H	H
L	L	H	L	L	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H	H	H
L	L	H	L	L	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H	H
L	L	H	L	H	L	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H
L	L	H	H	L	H	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H
L	L	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H
L	L	H	H	H	H	X	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H
H	L	X	X	X	X	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
H	H	X	X	X	X	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H

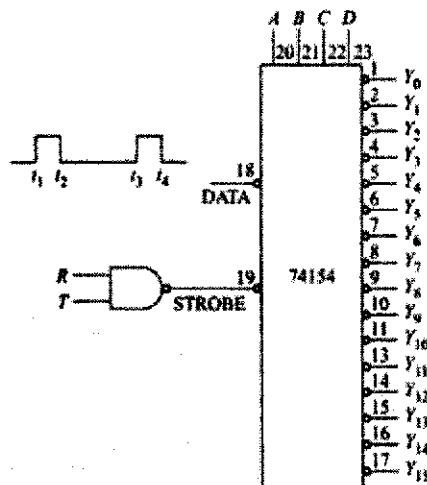
74154 Truth Table

Problem: In the following Fig, what does Y_{12} output equal for the following conditions:

- R is high, T is high, $ABCD = 0110$
- R is low, T is high, $ABCD = 1100$
- R is high, T is high, $ABCD = 1100$.

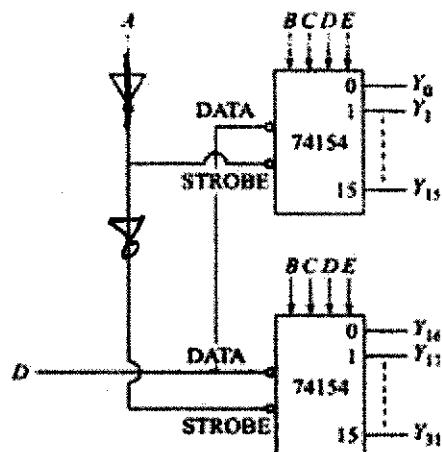
Solution:

- Since R and T both are high, the STROBE is low and the 74154 is active. Since, $ABCD = 0110$, the input data is steered to the Y_6 output line. The Y_{12} output remains in the high state.
- Since, the STROBE will be high, the 74154 will be inactive. The Y_{12} output will be high.
- Since, the STROBE is ~~high~~^{low}, the 74154 is active. Since, $ABCD = 1100$, the two pulses (from DATA input) are steered to the Y_{12} output.



Problem: Show how two 1-to-16 Demultiplexers can be connected to get a 1-to-32 Demultiplexer.

Solution: The following Fig shows the circuit diagram. A 1-to-32 Demultiplexer has 5 select lines, ABCDE. Four of them (BCDE) are fed to two 1-to-16 Demultiplexer; and the fifth (A) is used to select one of these two Demultiplexer through STROBE input. If $A = 0$, the top 74154 is chosen and BCDE directs DATA to one of the 15 outputs of that IC. If $A = 1$, the bottom 74154 is chosen and BCDE directs DATA to one of the 15 outputs of this IC.



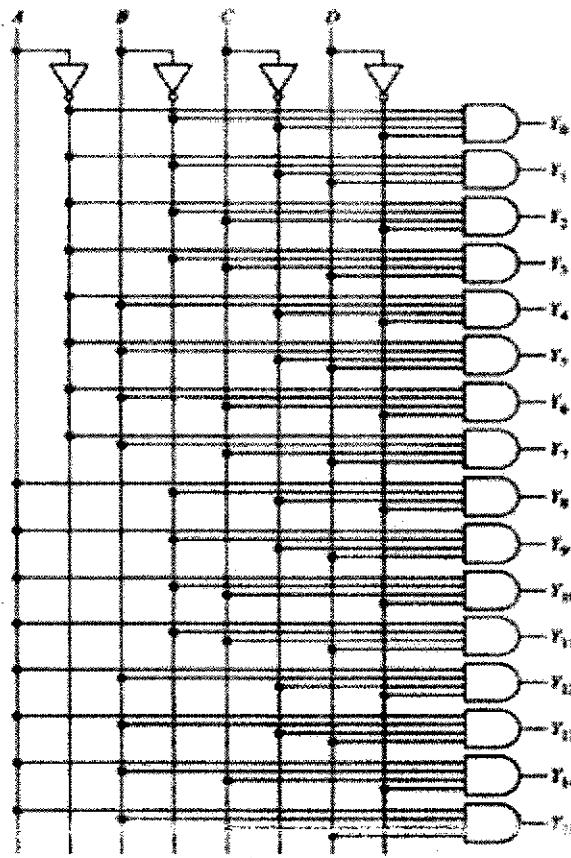
1-OF-16 DECODER:

A *decoder* is similar to demultiplexer, with one exception – there is no data input. The only inputs are the control bits ABCD, which are shown in the following Fig. This logic circuit is called a *1-of-16 decoder*, because only 1 of the 16 output lines is high. For example, when ABCD is 0001, only the Y_1 AND gate has all inputs high; therefore, only the Y_1 output is high. If ABCD changes to 0100, only the Y_4 AND gate has all the inputs high; hence, only Y_4 output goes high.

The subscript of the high output always equals the decimal equivalent of ABCD. For this reason, the circuit is also called as *binary-to-decimal decoder*. Since, it has 4 input lines and 16 output lines, the circuit is also known as a *4-line to 16-line decoder*.

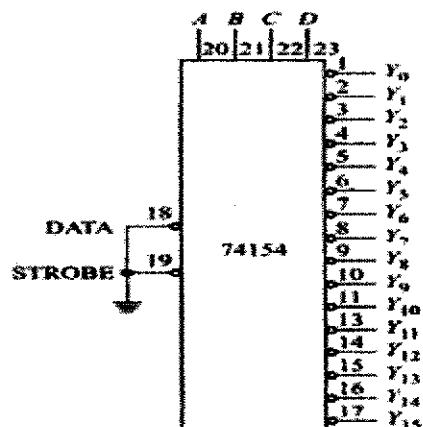
MAHESH PRASANNA K., VCET, PUTTUR





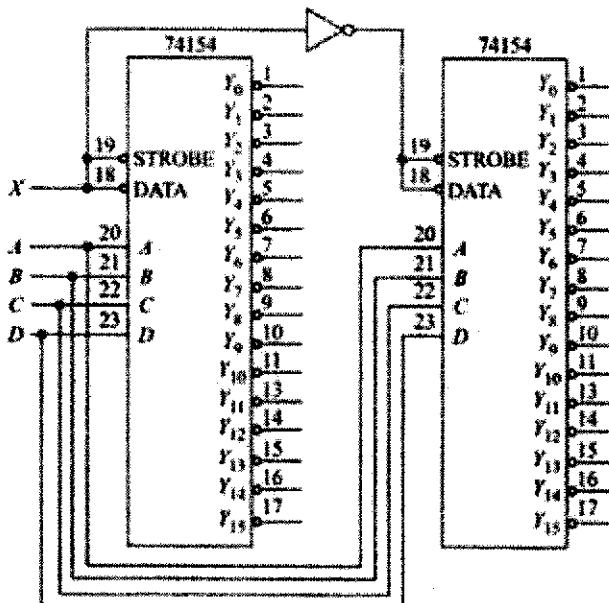
1-of-16 Decoder

The 74154 is called a decoder-demultiplexer, because it can be used either as a decoder or as a demultiplexer. To use the same IC as a decoder, all you have do is ground the DATA and STROBE inputs as shown in the following Fig. Then the selected output line is in the low state. For example, if the binary input is ABCD = 0111, then the Y_7 output is low, while all other outputs are high.



Using 74154 as Decoder

Chip Expansion: The following Fig illustrates *chip expansion*. Here, we have expanded two 74154s to get a 1-of-32 decoder. Bit X drives the first 74154, and the complement of X drives the second 74154. When X is low, the first 74154 is active and the second is inactive. The ABCD input drives both decoders.

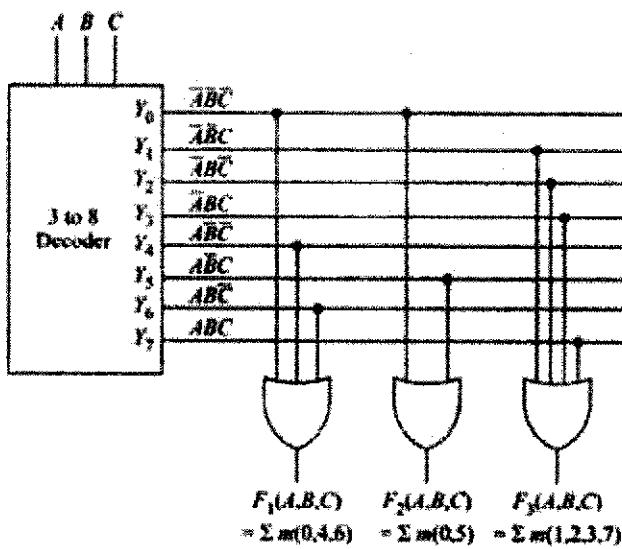


Chip Expansion

Problem: Show how using a 3-to8 decoder and multi-input OR gates following Boolean expression can be realized simultaneously.

$$F_1(A, B, C) = \sum m(0, 4, 6) \quad F_2(A, B, C) = \sum m(0, 5) \quad F_3(A, B, C) = \sum m(1, 2, 3, 7).$$

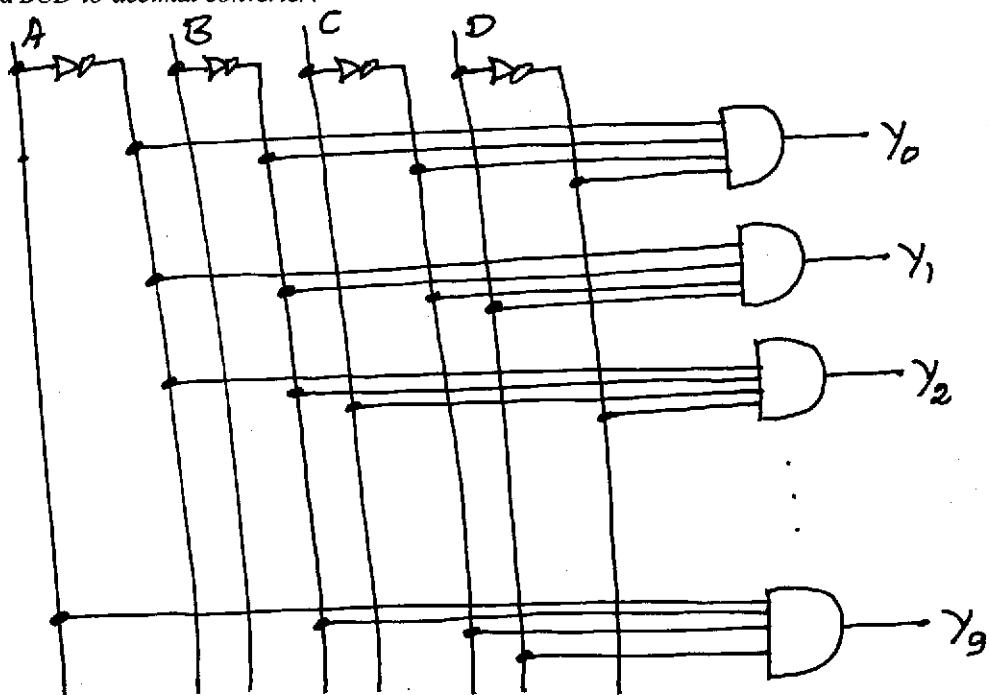
Solution: Since, at the decoder output, we get all the min-terms, we use them as shown in the following Fig, to get the required Boolean expression:



BCD-TO-DECIMAL DECODER:

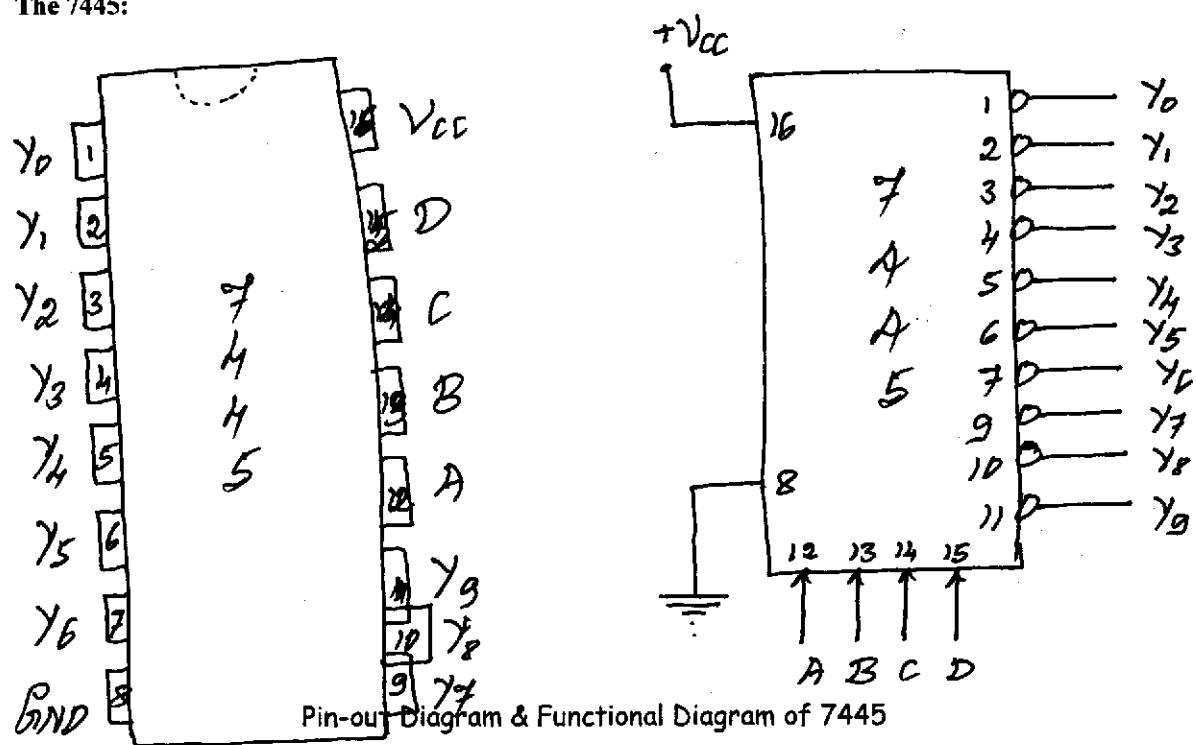
The circuit of the following Fig is called a *1-of-10 decoder*, because, only 1 of the 10 output lines is high. For example, when ABCD is 0011, only the Y_3 AND gate has all high inputs; therefore, only the Y_3 output is high.

If you check the other ABCD possibilities (0000 to 1001), you will find that, the subscript of the high output always equals the decimal equivalent of the input BCD digit. For this reason, the circuit is also called a *BCD-to-decimal converter*.



1-of-10 decoder

The 7445:

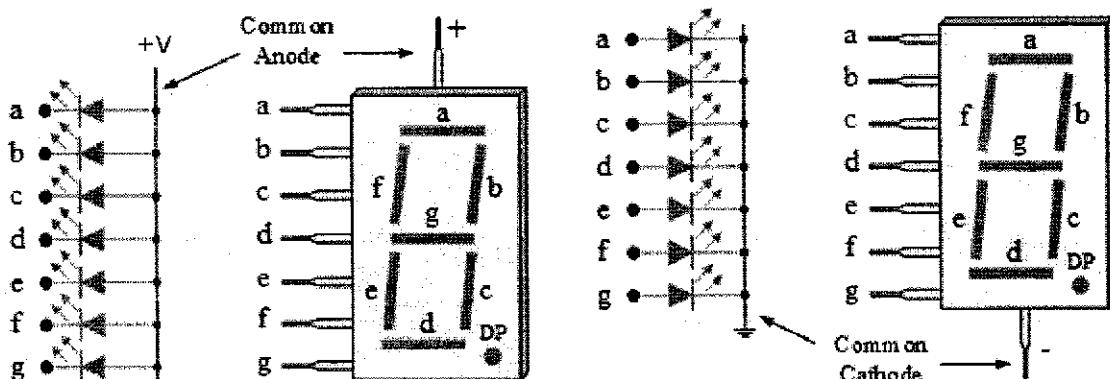


No.	Inputs				Outputs									
	A	B	C	D	Y_0	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7	Y_8	Y_9
0	L	L	L	L	L	H	H	H	H	H	H	H	H	H
1	L	L	L	H	H	L	H	H	H	H	H	H	H	H
2	L	L	H	L	H	H	L	H	H	H	H	H	H	H
3	L	L	H	H	H	H	H	L	H	H	H	H	H	H
4	L	H	L	L	H	H	H	H	L	H	H	H	H	H
5	L	H	L	H	H	H	H	H	L	H	H	H	H	H
6	L	H	H	L	H	H	H	H	H	L	H	H	H	H
7	L	H	H	H	H	H	H	H	H	H	L	H	H	H
8	H	L	L	L	H	H	H	H	H	H	H	L	H	H
9	H	L	L	H	H	H	H	H	H	H	H	H	L	H
	H	L	H	L	H	H	H	H	H	H	H	H	H	H
	H	L	H	H	H	H	H	H	H	H	H	H	H	H
	H	H	L	L	H	H	H	H	H	H	H	H	H	H
	H	H	L	H	H	H	H	H	H	H	H	H	H	H
	H	H	H	L	H	H	H	H	H	H	H	H	H	H
	H	H	H	H	H	H	H	H	H	H	H	H	H	H

7445 Truth Table

SEVEN-SEGMENT DECODERS:

The following Fig shows a *seven-segment indicator*, i.e. seven LEDs labeled *a* through *g* (actually, eight LEDs labeled through *a* through *h*). By forward biasing the LEDs, we can display the digits 0 through 9. For example, to display the digit 0, we need to light-up the segments *a*, *b*, *c*, *d*, *e*, and *f*. Similarly, to light-up the digit 5, we need segments *a*, *c*, *d*, *f*, and *g*.



Seven-Segment Indicator: Common Anode Type & Common Cathode Type

Seven-segment indicators may be *common-anode* type; where all anodes are connected together (as shown above) or *common-cathode* type; where all cathodes are connected together (as shown above).

CA: Low input for bright LED.

MAHESH PRASANNA K., VCET, PUTTUR

0: 1 1 0 0 0 0 0 = C0H

3: 0 0 1 1 0 0 0 = B0H

8: 1 0 0 0 0 0 0 = 80H

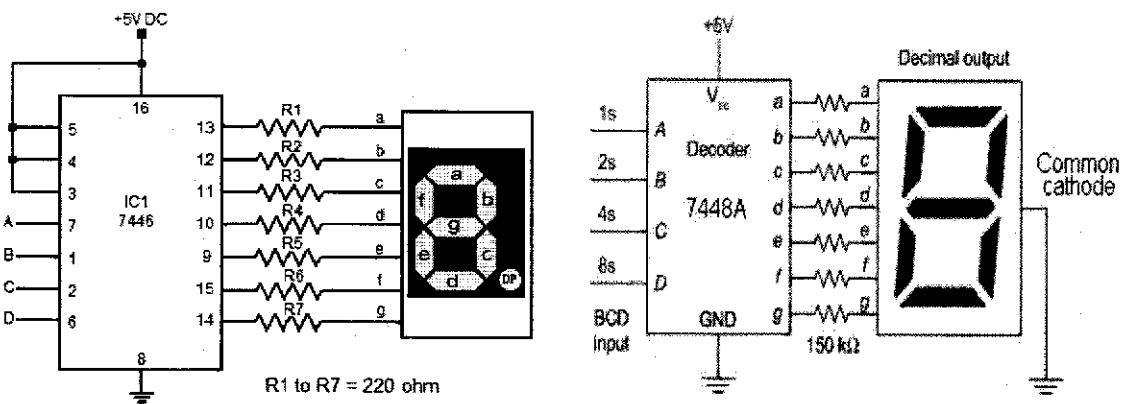
CC: High input for bright LED.
dp g f e & e b a

0: 0 0 1 1 1 1 1 = 3FH

3: 0 1 0 0 1 1 1 1 = 4FH

8: 0 1 1 1 1 1 1 1 = 7FH

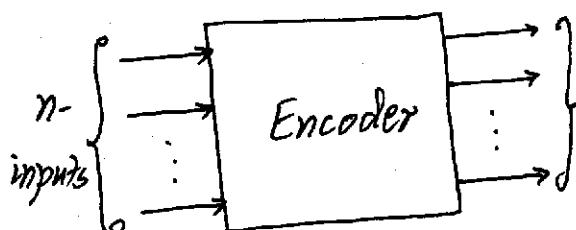
The 7446 & The 7448: A seven-segment decoder-driver is an IC decoder that can be used to drive a seven-segment indicator. There are two types of decoder-drivers, corresponding to common-anode (IC 7446) and common cathode (IC 7448) indicators. Each decoder driver has 4 input pins (the BCD input) and 7 output pins (*a* through *h* segments), as shown in the following Fig.



7446 Decoder-driver & 7448 Decoder-driver

The logic circuits inside 7446 / 7448 convert the BCD input to the required output. For Example, if the BCD input is 0111, the internal logic of the 7446 / 7448 will force segments *a*, *b*, and *c* to conduct. As a result, digit 7 will appear on the seven-segment indicator.

ENCODERS:

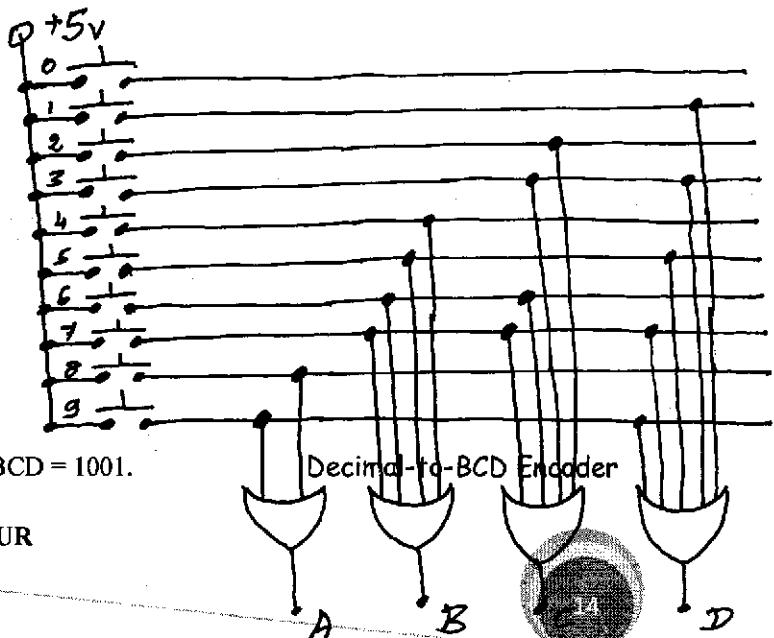


An encoder converts an active input signal to a coded

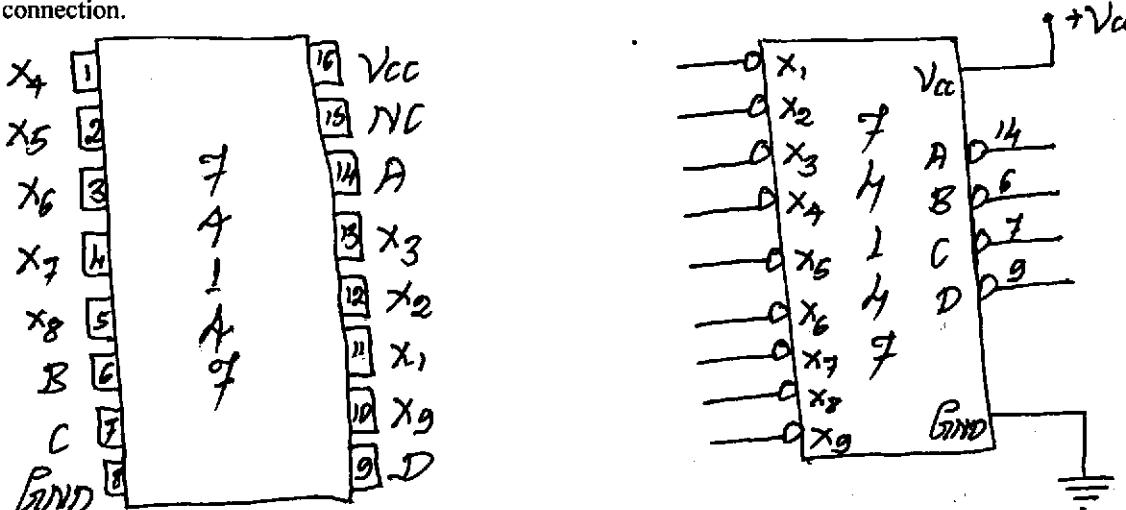
m-output signal. The Fig illustrates the general idea. There are *n* input lines, only one of which is active. The internal logic within the encoder converts this active input to a coded binary output with *m* bits.

Decimal-to-BCD Encoder:

The following Fig shows a common type of encoder – the decimal-to-BCD encoder. The switches are push-button switches. When button 3 is pressed, the C and D OR gates have high inputs; therefore, the output is ABCD = 0011. If button 5 is pressed, the output becomes ABCD = 0101. When switch 9 is pressed, ABCD = 1001.



Qe 74147: The following Fig shows the pin-out and the logic diagram of a 74147- a decimal-to-BCD encoder. The decimal inputs, X_1 to X_9 , connect to pins 1 to 5, and 10 to 13. The BCD output comes from pins 14, 6, 7, and 9. Pin 16 is for supply voltage, and pin 8 is grounded. The label NC on pin 15 means no connection.



Pin-out Diagram & Logic Diagram of 74147

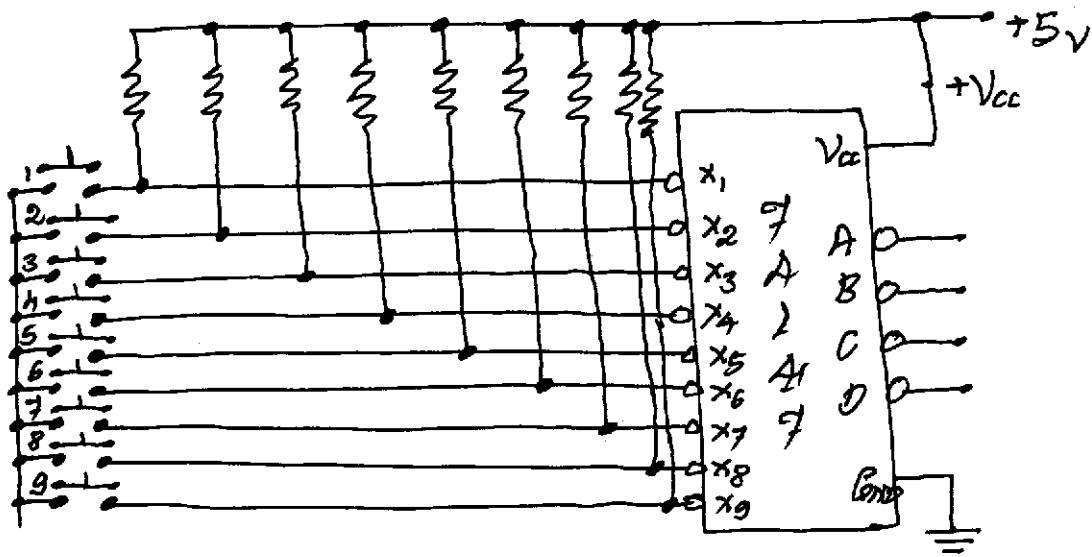
The bubbles indicate active-low inputs and outputs. The following Table is the truth table of 74147. When all X inputs are high, all outputs are high. When X_9 is low, the ABCD output is LHHL (equivalent to 9 if you complement the bits). When X_8 is the only low input, ABCD is LHHH (equivalent to 8 if the bits are complemented).

Inputs										Outputs			
X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	A	B	C	D	
H	H	H	H	H	H	H	H	H	H	H	H	H	H
H	H	H	H	H	H	H	H	L	L	H	H	L	
H	H	H	H	H	H	H	L	H	L	H	H	H	H
H	H	H	H	H	H	L	H	H	H	L	L	L	L
H	H	H	H	H	L	H	H	H	H	L	L	H	H
H	H	H	H	L	H	H	H	H	H	L	H	L	H
H	H	L	H	H	H	H	H	H	H	H	H	L	L
H	L	H	H	H	H	H	H	H	H	H	H	L	H
L	H	H	H	H	H	H	H	H	H	H	H	H	L

74147 Truth Table

The 74147 is called a *priority encoder*, because it gives priority to the highest-order input. If all inputs X_1 through X_9 are low, the highest of these, X_9 , will be encoded to get an output LHHL. In other words, X_9 has priority over all others. When X_9 is high, X_8 is the next inline of priority and gets encoded if it is low.

Problem: What is the ABCD output of the following Fig. when button 6 is pressed?



Solution:

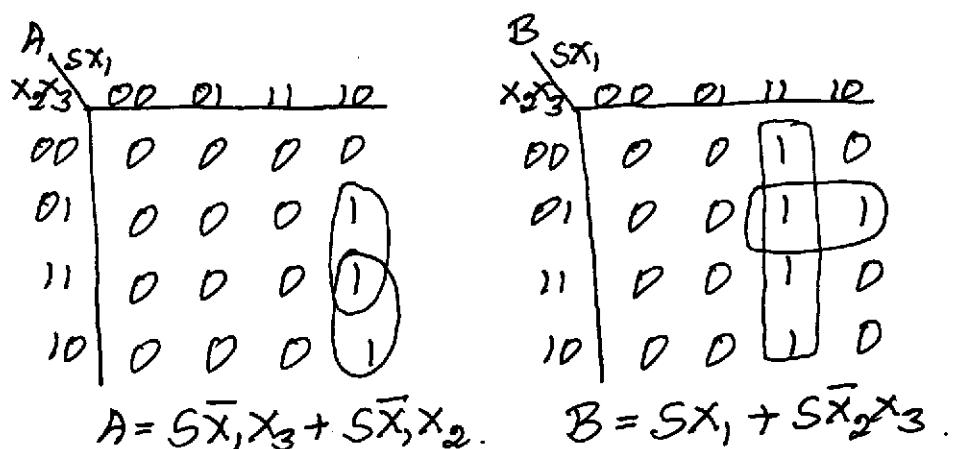
When all switches are open, the X_1 to X_9 inputs are pulled up to the high state (+5 V). Hence, ABCD output is HHHH at this time.

When switch 6 is pressed, the X_6 input is grounded. Therefore, all X inputs are high, except for X_6 . Hence, the ABCD output is HLLH, which is equivalent to 6 when the output bits are complemented.

Problem: Design a priority encoder, the truth table of which is shown in the following Fig. The order of priority for these inputs is $X_1 > X_2 > X_3$. However, if the encoder is not enabled by S or all the inputs are inactive, the output AB = 00.

Solution:

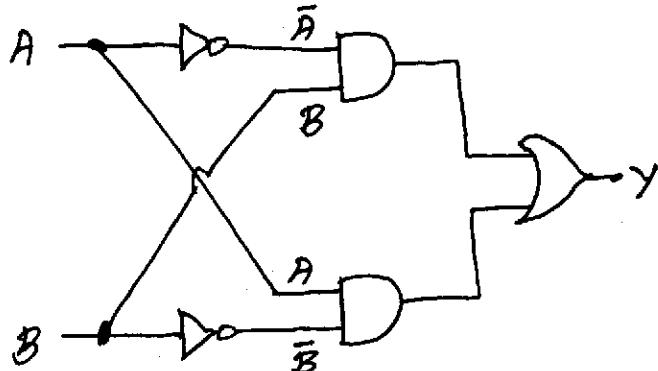
Inputs				Outputs	
S	X_1	X_2	X_3	A	B
0	x	x	x	0	0
1	1	x	x	0	1
1	0	1	x	1	0
1	0	0	1	1	1
1	0	0	0	0	0



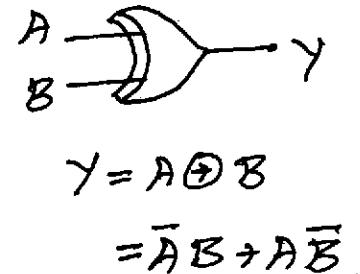
Construct Karnaugh map for output A and B. Then, taking groups of 1s, we get the design equations, as shown above. The logic circuits for input A and B can be directly drawn from the equations.

EXCLUSIVE-OR (Ex-OR) GATES:

The *exclusive-OR gate* has a high output, when an odd number of inputs are high. The following Fig shows how to build an exclusive-OR gate. The upper AND gate forms the product $A'B$, while the lower one produces AB' . Therefore, the output of the OR gate is $Y = \bar{A}B + A\bar{B}$.



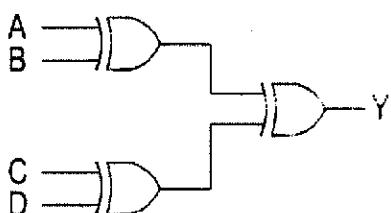
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0



Exclusive-OR Gate, Its Logic Symbol & Truth Table

Problem: Construct the truth table for a 4 input XOR gate.

Solution:



The gate produces an output 1, when the ABCD input has an odd number of 1s.

NOTE: In general, you can build an exclusive-OR gate with any number of inputs. Such a gate always produces an output 1 only when an n -bit input has an odd number of 1s.

PARITY GENERATORS AND CHECKERS:

Parity is the number of 1s in an n -bit input. *Even parity* means an n -bit input has an even number of 1s. *Odd parity* means an n -bit input has an odd number of 1s.

1111 0000 1111 0011 even parity

1111 0000 1111 0111 odd parity

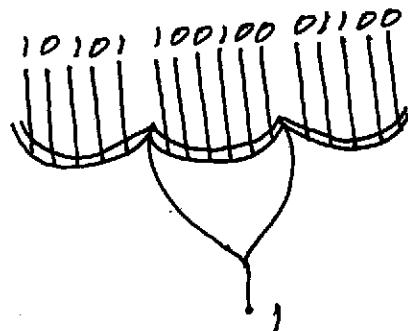
A	B	C	D	Y	Comment
0	0	0	0	0	Even
0	0	0	1	1	Odd
0	0	1	0	1	Odd
0	0	1	1	0	Even
0	1	0	0	1	Odd
0	1	0	1	0	Even
0	1	1	0	0	Even
0	1	1	1	1	Odd
1	0	0	0	1	Odd
1	0	0	1	0	Even
1	0	1	0	0	Even
1	0	1	1	1	Odd
1	1	0	0	0	Even
1	1	0	1	1	Odd
1	1	1	0	1	Odd
1	1	1	1	0	Even



Parity Checker:

Exclusive-OR gates are ideal for checking the parity of a binary number, because they produce an output 1 when the input has an odd number of 1s. Therefore, an even-parity input to an exclusive-OR gate produces a low output, while an odd-parity input produces a high output. The following Fig shows a 16-input exclusive-OR gate, which produces an output 1, because the input has odd parity.

$1111\ 0000\ 1111\ 0011$: even parity
 $1111\ 0000\ 1111\ 0111$: odd parity



Exclusive-OR gate with 16 Inputs

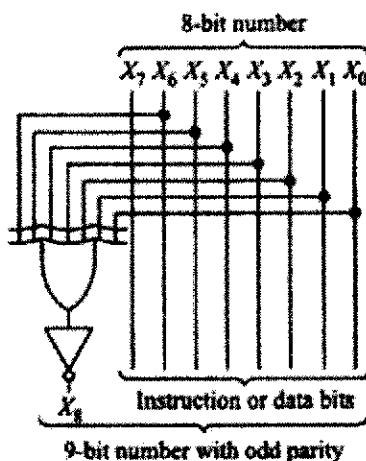
Parity Generation

In a computer, a binary number may represent –

1. An instruction that tells the computer to add, subtract, and so on or
2. A data to be processed like number, letter, etc.

In both the cases, sometimes, an extra bit will be added to the original binary number to produce a new binary number with even or odd parity.

Consider the following Fig:

**Odd-Parity Generation**

The Fig shows an 8-bit binary number: $X_7\ X_6\ X_5\ X_4\ X_3\ X_2\ X_1\ X_0$.

Suppose this number equals: 0100 0001. Then, the number has even-parity, which means the exclusive-OR gate produces an output of 0. Because of the inverter, $X_8 = 1$. Hence, the final 9-bit output is 1 0100 0001. Notice that, this has odd-parity.

Suppose, we change the 8-bit input to 0110 0001. Now, it has odd-parity. In this case, the exclusive-OR gate produces an output 1. But the inverter produces a 0, so that the final 9-bit output is 0 0110 0001. Again, the final output has odd-parity.

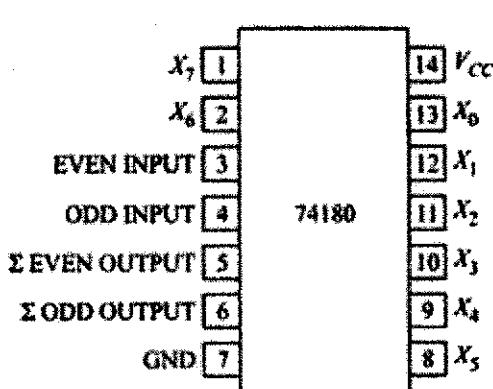
Hence, the circuit given above is called an *odd-parity generator*, because it always produces a 9-bit output number with odd-parity. If the 8-bit has even-parity, a 1 comes out of the inverter to produce the final output with odd-parity. On the other hand, if the 8-bit input has odd-parity, a 0 comes out of the inverter, and the final 9-bit output again has odd-parity.

NOTE: To get an even-parity generator, delete the inverter.

Applications of parity generation and checking: When binary data is transmitted over telephone lines, sometimes, 1-bit error occurs due to noise and other disturbances. One way to check for errors is to use an odd-parity generator at the transmitting end and an odd-parity checker at the receiving end. If no 1-bit error occurs in the transmission, the received data will have odd parity. But, if one of the transmitted bits is changed by noise or other disturbance, the received data will have even-parity.

For example; suppose, we want to send 0100 0011. With an odd-parity generator, the data to be transmitted will be 0 0100 0011. This data can be sent over telephone lines. If no error occurs in transmission, the odd-parity checker at the receiving end will produce a high output, meaning the received number has odd-parity. On the other hand, if a 1-bit error does creep into the transmitted data, the odd-parity checker will have a low output, indicating the received data is invalid.

The 74180: The following Fig shows the pin-out diagram and truth table for a 74180, a TTL parity generator-checker. The input data bits are X_7 to X_0 ; these bits may have even or odd parity. The even input (pin 3) and the odd input (pin 4) control the operation of the chip. The symbol Σ stands for summation. In the left input column of the table, Σ of H's (highs) refers to the parity of the input data X_7 to X_0 . Depending on how you set up the values of the even and odd inputs, the Σ even and Σ odd outputs may be low or high.



Inputs		Outputs		
Sum of H's of X_7 to X_0	Even	Odd	Σ even	Σ odd
Even	H	L	H	L
Odd	H	L	L	H
Even	L	H	L	H
Odd	L	H	H	L
X	H	H	L	L
X	L	L	H	H

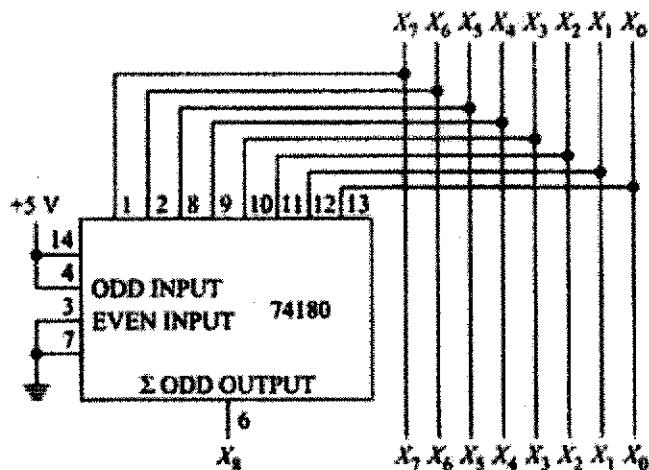
Pin-out Diagram & Truth Table of 74180

For example, suppose even input is high and odd input is low. When the input data has even-parity, the \sum even output is high and \sum odd output is low. When the input data has odd-parity, the \sum even output is low and the \sum odd output is high.

The 74180 can be used to detect even or odd parity. It can also be set up to generate even or odd parity.

Problem: Show how to connect a 74180 to generate a 9-bit output with odd parity.

Solution:



Using a 74180 to Generate Odd-Parity

The ODD INPUT (pin 4) is connected to +5 V, and the EVEN INPUT (pin 3) is grounded. Suppose the input data $X_7 \dots X_0$ has even-parity. Then, the 3rd entry of the above Table, tells us, the \sum ODD OUTPUT (pin 6) is high. Therefore, the 9-bit number $X_8 \dots X_0$ coming out of the circuit has odd parity.

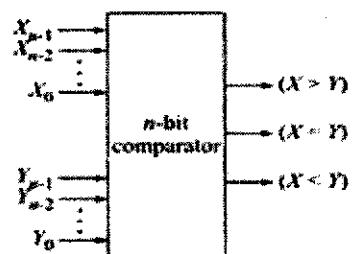
On the other hand, suppose the input data $X_7 \dots X_0$ has odd-parity. Then, the 4th entry of the above Table, tells us, the \sum ODD OUTPUT (pin 6) is low. Again, the 9-bit number $X_8 \dots X_0$ coming out of the circuit has odd parity.

Hence, the circuit shown in the above Fig, always generates a 9-bit output with odd parity.

MAGNITUDE COMPARATOR:

Magnitude comparator compares magnitude of two n-bit numbers, say X and Y, and activates one of these three outputs $X < Y$, $X = Y$, and $X > Y$.

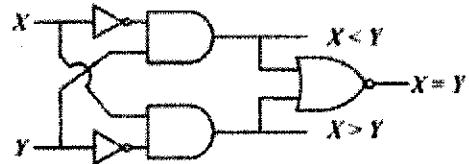
The Fig presents block diagram of such a comparator.



Block Diagram of Magnitude Comparator

1-Bit Comparator:

Inputs		Outputs		
X	Y	(X > Y): G	(X < Y): L	(X = Y): E
0	0	0	0	1
0	1	0	1	0
1	0	1	0	0
1	1	0	0	1
Logic Equations:		$G = XY'$	$L = X'Y$	$E = X'Y' + XY$ $= (XY' + X'Y)'$ $= (G - L)'$



Truth Table, Logic Equations & Circuit Diagram for a 1-Bit Comparator

2-Bit Comparator: We can form a 4-variable (X: X₁X₀ and Y: Y₁Y₀) truth table and get logic equations through and simplification technique. But this procedure will become very complex. Hence, we shall use the truth table of 1-bit comparator that generates less than, equal to, and greater than terms. Hence;

Bit-wise greater than (G):

$$G_1 = X_1 \bar{Y}_1$$

$$G_0 = X_0 \bar{Y}_0$$

Bit-wise less than (L):

$$L_1 = \bar{X}_1 Y_1$$

$$L_0 = \bar{X}_0 Y_0$$

Bit-wise equality term (E):

$$E_1 = \overline{(G_1 \quad L_1)}$$

$$E_0 = \overline{(G_0 \quad L_0)}$$

From these definitions; we can easily write, 2-bit comparator outputs as follows:

$$(X = Y) \text{ if } X_1 = Y_1 \& X_0 = Y_0$$

$$(X = Y) = E_1 \cdot E_0$$

$$(X > Y) \text{ if } X_1 > Y_1 \text{ or } X_1 = Y_1 \& X_0 > Y_0$$

$$(X > Y) = G_1 + E_1 \cdot G_0$$

$$(X < Y) \text{ if } X_1 < Y_1 \text{ or } X_1 = Y_1 \& X_0 < Y_0$$

$$(X < Y) = L_1 + E_1 \cdot L_0$$

The logic followed is this: X = Y, when both the bits are equal. X > Y if MSB of X is higher (G₁ = 1) than that of Y. If MSB is equal (given by E₁ = 1), then LSB of X and Y is checked and if found higher (G₀ = 1), the condition X > Y is fulfilled. Similar logic gives us the X < Y term.

Thus for any two n-bit numbers X: X_{n-1}X_{n-2}...X₀ and Y: Y_{n-1}Y_{n-2}...Y₀ We can write –

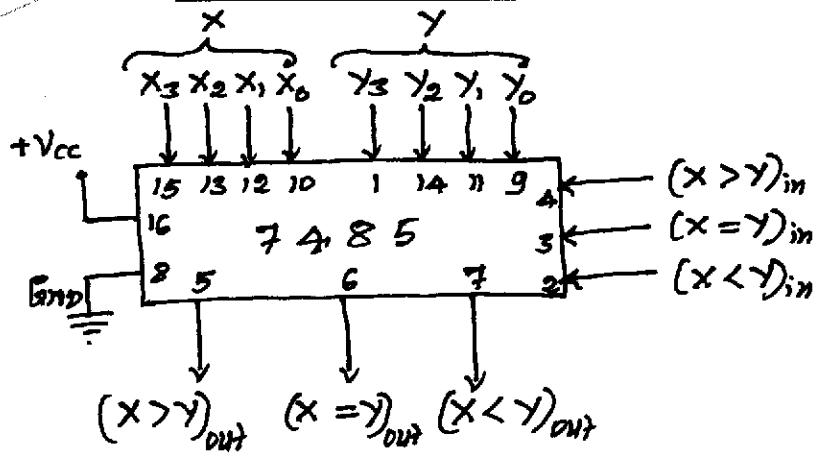
$$(X = Y) = E_{n-1} \cdot E_{n-2} \dots E_0$$

$$(X > Y) = G_{n-1} + E_{n-1} \cdot G_{n-2} + \dots + E_{n-1} \cdot E_{n-2} \dots E_1 \cdot G_0$$

$$(X < Y) = L_{n-1} + E_{n-1} \cdot L_{n-2} + \dots + E_{n-1} \cdot E_{n-2} \dots E_1 \cdot L_0$$

The 7485: The block diagram of IC 7485, which compares two 4-bit numbers, is shown in the following Fig. This is a 16-pin IC. Note that the IC has three additional inputs in the form of (X = Y)_{in}, X > Y)_{in}, and (X < Y)_{in}. These inputs are used when we need to cascade more than one IC 7485 to compare numbers having more than 4-bits. When IC 7485 is not used in cascade, we keep (X = Y)_{in} = 1, X > Y)_{in} = 0, and (X < Y)_{in} = 0.

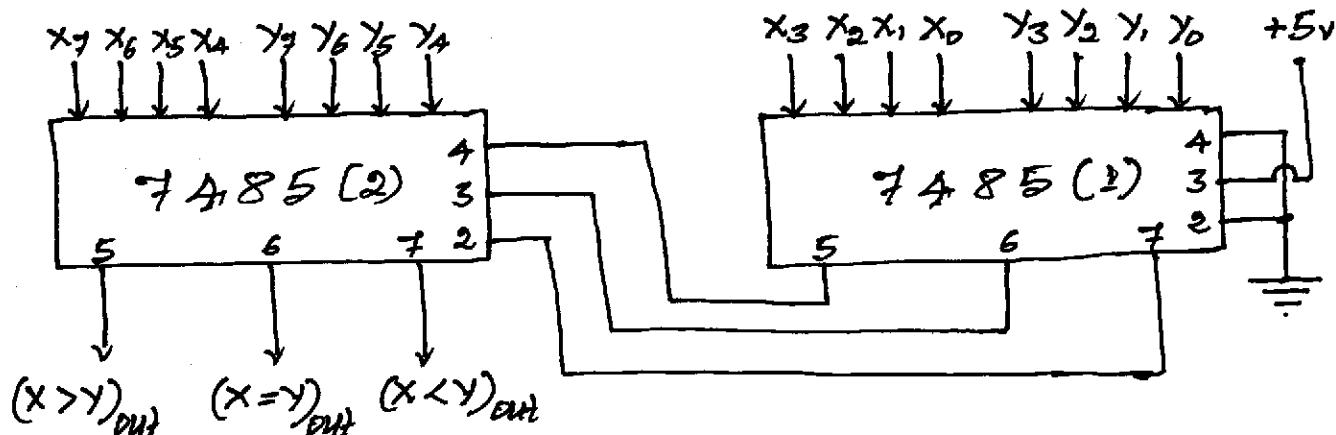
ANALOG AND DIGITAL ELECTRONICS



Functional Diagram of IC 7485

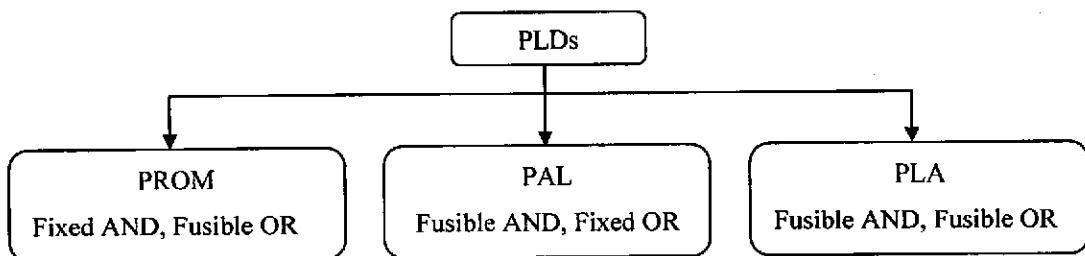
Problem: Show how two IC 7485 can be used to compare magnitude of two 8-bit numbers.

Solution:

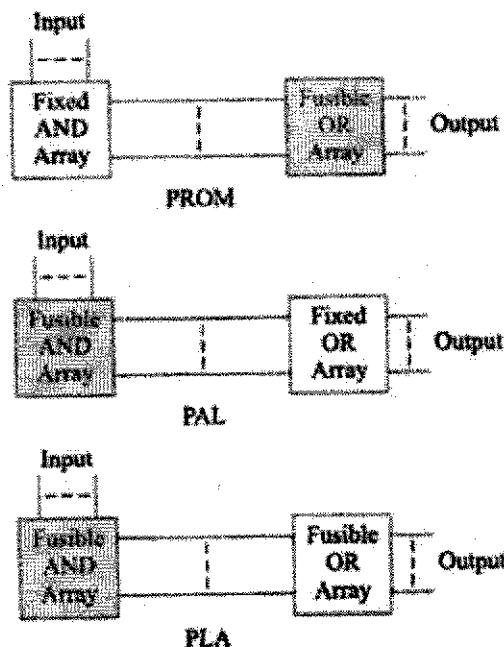


8-Bit Comparator from Two 4-Bit Comparators

NOTE: *Programmable Logic Device (PLD)* is an electronic component, used to build reconfigurable digital circuits. Programmable Read Only Memory (PROM), Programmable Array Logic (PAL), and Programmable Logic Array (PLA) are included in the general classification.



General Classification of PLDs



Basic Operation of Three PLDs

In PLDs, the input signals are presented to an array of AND gates, while the outputs are taken from an array of OR gates.

- The input AND gate array used in a PROM is fixed and cannot be altered, while the output OR gate array is fusible-linked, and can be programmed.
- The output OR gate array used in a PAL is fixed and cannot be altered, while the input AND gate array is fusible-linked, and can be programmed.
- In PLA, both its input AND gate array and output OR gate array are fusible-linked, and can be programmed.

PROGRAMMABLE ARRAY LOGIC (PAL):

Programmable Array Logic (PAL) is a programmable array of logic gates on a single chip. PALs are another design solution, similar to SOPs, POSs, and multiplexer logic.

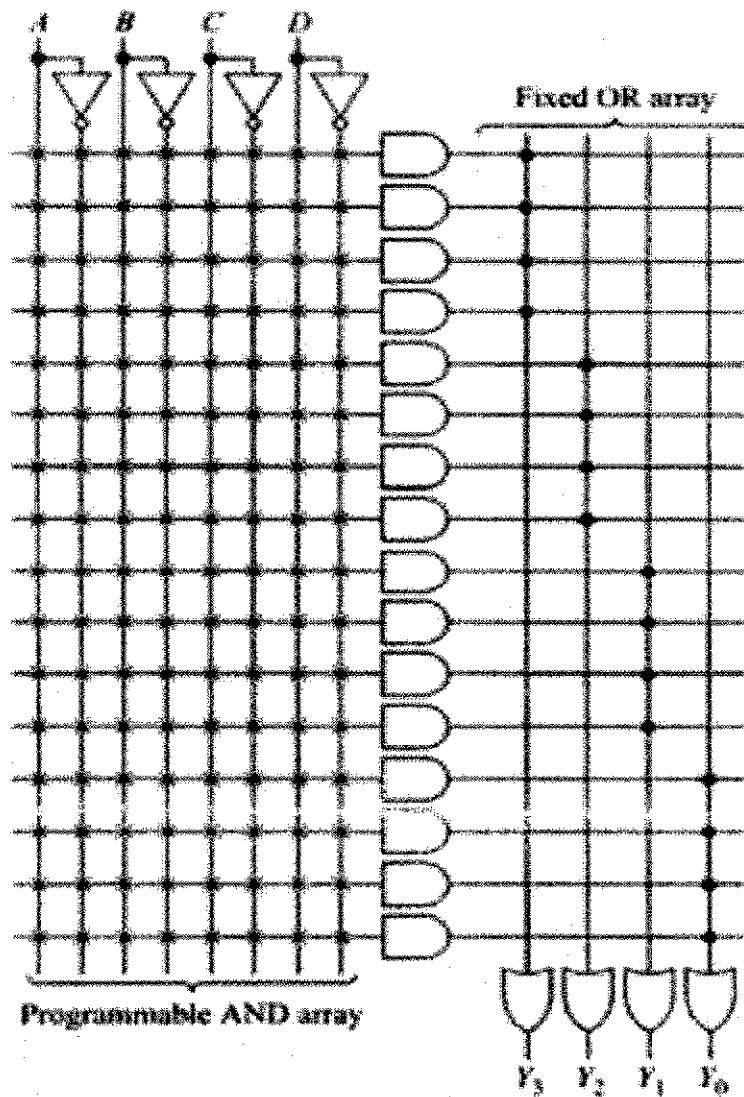
A PAL has a programmable AND array and a fixed OR array. The following Fig shows a PAL with 4 inputs and 4 outputs. The x's on the input side are fusible links, while the solid bullets on the output side are fixed connections.

Commercially available PALs: 10H8 – 10 input and 8 output AND-OR

16H2 – 16 input and 2 output AND-OR

14L4 – 14 input and 4 output AND-OR-INVERT



NOTE:

1. PALs are not universal logic solution; because, only some of the fundamental products can be generated and ORed at the final outputs.
2. PALs have enough flexibility to produce all kinds of complicated logic functions.
3. PALs have the advantage of 16 inputs compared to typical limit of 8 inputs for PROMs.

Problem: Generate the following Boolean function using PAL.

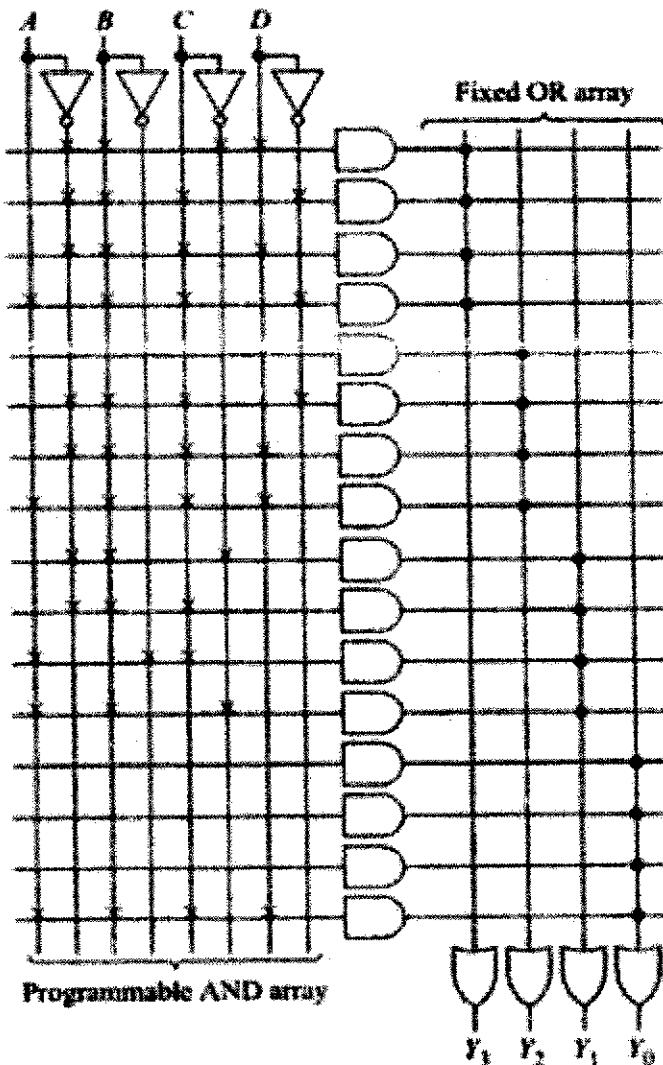
$$Y_3 = \bar{A}\bar{B}\bar{C}D + \bar{A}BC\bar{D} + \bar{A}B\bar{C}D + ABC\bar{D}$$

$$Y_2 = \bar{A}BC\bar{D} + \bar{A}B\bar{C}D + ABCD$$

$$Y_1 = \bar{A}B\bar{C} + \bar{A}BC + A\bar{B}C + AB\bar{C}$$

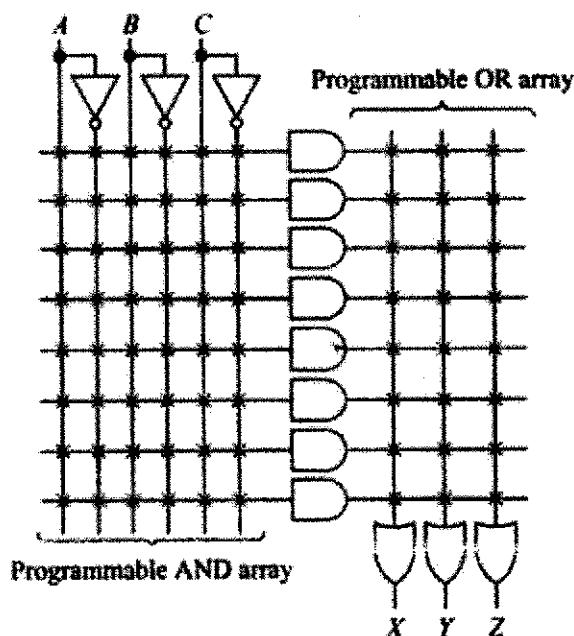
$$Y_0 = ABCD$$

Solution: Start with first equation. The first desired product is $A'BC'D$, which is marked as shown in the following Fig. The fixed OR connections on the output side imply that the first OR gate produces an output of first equation.

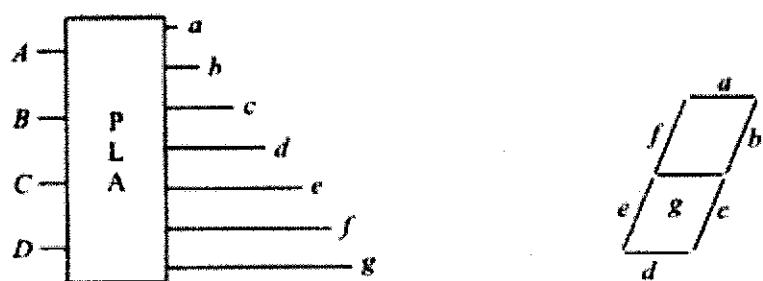


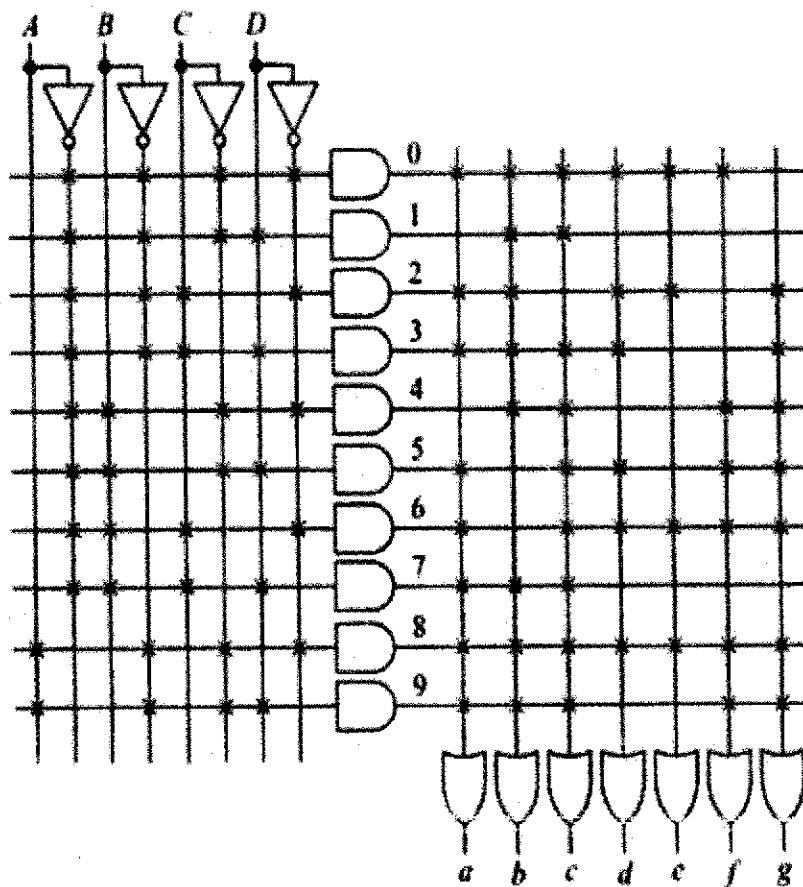
PROGRAMMABLE LOGIC ARRAY (PLA):

A PLA having 3 input variables (ABC) and 3 output variables (XYZ) is illustrated in the following Fig. Eight AND gates are required to decode the 8 possible input states. There are three OR gates that can be used to generate logic functions at the output. Note that, there could be additional OR gates at the output, if desired.



Example: Suppose it is desired to use a PLA to recognize each of the 10 decimal digits represented in binary form and to correctly drive a 7-segment display. Then, the PLA must have 4 inputs, as shown in the following Fig. Four bits are required to represent the 10 decimal numbers. There must be 7 outputs (a, b, c, d, e, f, g).





7-Segment Decoder using PLA

The circuit given in the above Fig shows the links after programming. The input AND-gate array is programmed such that, each AND gate decodes one of the decimal numbers. Then, links are removed from the output OR-gate array, such that proper segments of the indicator are illuminated. For example, when ABCD = LHLH, segments *a**f**g**c**d* are illuminated to display the decimal number 5.

NOTE: Many PLDs are programmable only at the factory. However, PLDs can be programmed by the user. These are said to be *field-programmable*, and the letter F is often used to indicate this fact. For example, the Texas Instruments TIFPLA840 is field-programmable PLA with 14 inputs, 32 AND gates, and 6 OR gates; describes as 14 x 32 x 6 FPLA.

HDL IMPLEMENTATION OF DATA PROCESSING CIRCUITS:

The data flow model provides a different use of keyword *assign* in the form of –

MAHESH PRASANNA K., VCET, PUTTUR

Recall:

Structural \leftrightarrow Gates / Circuits

Dataflow \leftrightarrow *assign* / Expression

Behavioral \leftrightarrow *always* / Truth table

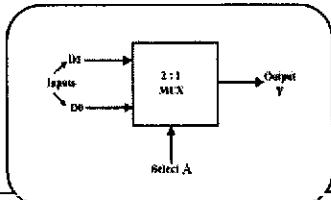
```
assign X = S ? A : B;
```

If S = 1, X = A, and if S = 0, X = B.

2-to-1 Multiplexer: The data flow model and behavioral model for a 2-to-1 multiplexer is given below:

```
module mux2to1(A,D0,D1,Y);
  input A,D0,D1; /* Circuit shown */
  output Y;
  assign Y=(~A&D0)+(A&D1);
endmodule
```

```
module mux2to1(A,D0,D1,Y);
  input A,D0,D1; /* Circuit shown */
  output Y;
  reg Y;
  always @ (A or D0 or D1)
    if (~A&1) Y=D1;
    else Y=D0;
endmodule
```



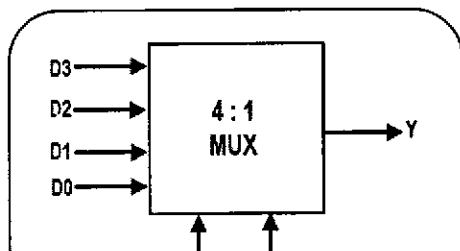
```
module mux2to1(A,D0,D1,Y);
  input A,D0,D1; /* Circuit shown */
  output Y;
  assign Y= A ? D1 : D0; /*Conditional
  assignment*/
endmodule
```

```
module mux2to1(A,D0,D1,Y);
  input A,D0,D1; /* Circuit shown */
  output Y;
  reg Y;
  always @ (A or D0 or D1)
    case (A)
      0 : Y=D0;
      1 : Y=D1;
    endcase
endmodule
```

Problem: Design a 4-to-1 multiplexer (for the Fig given below), using conditional *assign* and *case* statements.

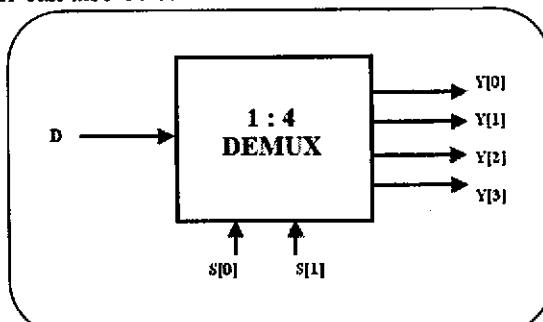
Solution:

```
module mux4to1(A,B,D0,D1,D2,D3,Y);
  input A,B,D0,D1,D2,D3;
  output Y; /* Circuit shown */
  assign Y = A ? (B ? D3 : D2):(B ?
  D1 : D0);
endmodule
```



```
module mux4to1(A,B,D0,D1,D2,D3,Y);
  input A,B,D0,D1,D2,D3;
  output Y;
  reg Y;
  always @ (A or B or D0 or D1 or D2
  or D3)
    case ((A,B)) /*Concatenation of A
    and B, A is MSB*/
      0:Y=D0; /*Two binary digit can
      generate*/
      1:Y=D1; /*four different values
      0,1,2,3 for*/
      2:Y=D2; /*binary combination
      00, 01,10*/
      3:Y=D3; /*and 11 respectively
      endcase
endmodule
```

Ans Representation in HDL: A 1-to-4 demultiplexer can also be served as 2-to-4 decoder. The Verilog code for this demultiplexer / decoder is given below:



```

module demux1to4(S,D,Y);
  input [1:0] S;
  input D;
  output [3:0] Y;
  reg [3:0] Y;
  always @ (S or D)
    case ({(D,S)}) //Concatenation of D and S to give 3 bits, D is MSB
      3'b000 : Y= 4'b0001; /*Binary representation, refer to Section 2-5.
      If D=1, S=00, Y=0001*/
      3'b101 : Y= 4'b0010; // if D=1, S=01, Y=0010
      3'b110 : Y= 4'b0100; // if D=1, S=10, Y=0100
      3'b111 : Y= 4'b1000; // if D=1, S=11, Y=1000
      default : Y= 4'b0000; //For other combinations D=0, then Y=0000
    endcase
endmodule
  
```

Problem: A Verilog HDL code for a digital circuit is given as follows. Can you describe the function it performs? Can it be related to any logic circuit?

```

module unknown (A, B, C, Y);
  input [3:0] A, B;
  input [2:0] C
  output [2:0] Y
  reg [2:0] Y;
  always @ (A or B or C)
    if (A<B) Y = 3'b001;
    else if (A>B) Y = 3'010;
    else Y = C;
  endmodule
  
```

Solution: The circuit described by the HDL compares two numbers A and B and generates a 3-bit output Y. It also has a 3-bit input C. If A is less than B, output Y = 001 and does not depend on C. Similarly, if A is greater than B, output Y = 010, irrespective of C. But, if these two conditions are not met, i.e. if A = B, then Y = C.

If we consider three bits of Y represent (starting from MSB) $A = B$, $A > B$, and $A < B$ respectively then, this circuit represents a 4-bit magnitude comparator, where C represents comparator output of previous stage. Hence, this is similar to IC 7458, 7485.

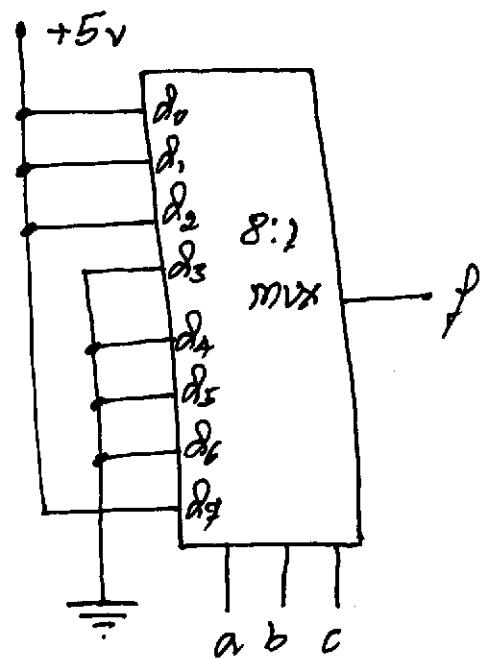
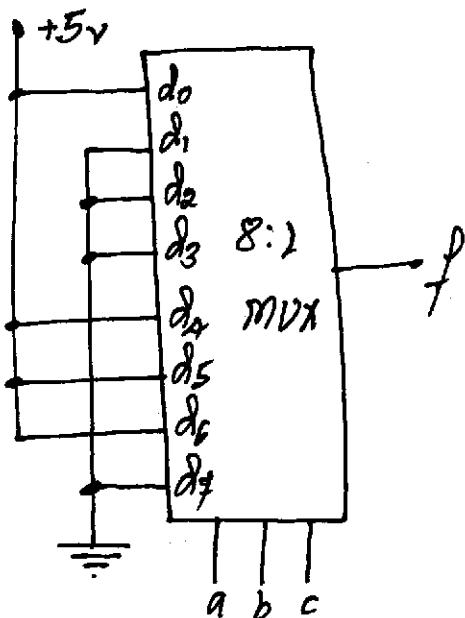
Problem: Implement the following functions using 8:1 Mux

a) $f(a, b, c) = \sum(0, 4, 5, 6)$

b) $f(a, b, c) = \sum(0, 1, 2, 7)$

c) $f(a, b, c) = \sum(0, 4, 5, 7)$

Solution:



Problem: Implement the following functions using 4:1 Mux

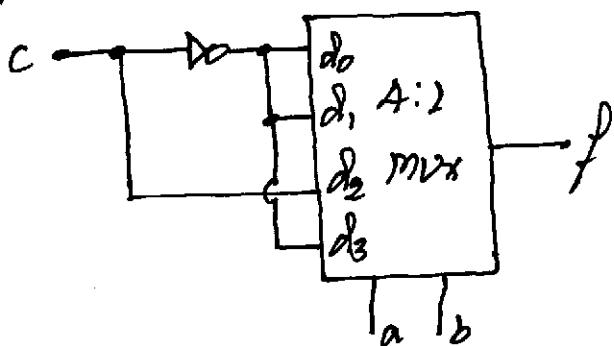
a) $f(a, b, c) = \sum(0, 4, 5, 6)$

b) $f(a, b, c) = \sum(0, 1, 2, 7)$

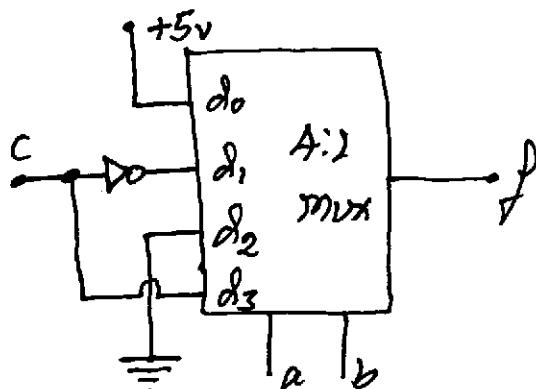
c) $f(a, b, c) = \sum(0, 4, 5, 7)$

Solution:

a) $f = \bar{a}\bar{b}\bar{c} + a\bar{b}\bar{c} + \bar{a}b\bar{c} + ab\bar{c}$



$$\begin{aligned} b) f &= \bar{a}\bar{b}\bar{c} + \bar{a}\bar{b}c + \bar{a}b\bar{c} + abc \\ &= \bar{a}b(\bar{c}+c) + \bar{a}b(\bar{c}) + ab(c) \\ &= \bar{a}\bar{b} \cdot 1 + \bar{a}b(\bar{c}) + \bar{a}b(0) + ab(c). \end{aligned}$$

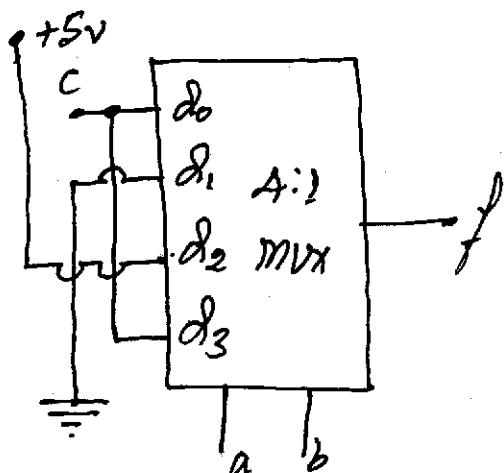


Problem: Implement $f(a, b, c) = \sum(1, 4, 5, 7)$ using 4:1 Mux, taking –

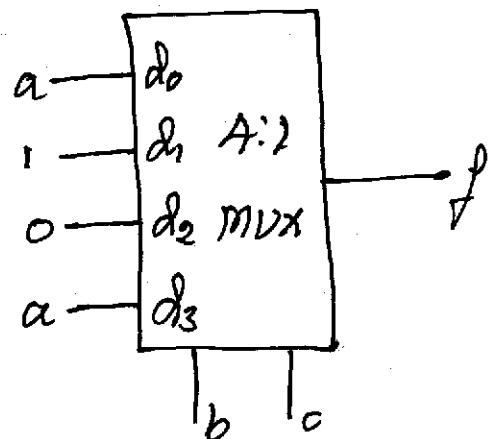
- a and b as address lines
- b and c as address lines.

Solution:

$$\begin{aligned}
 \textcircled{a} \quad f &= \bar{a}\bar{b}c + a\bar{b}\bar{c} + a\bar{b}c + abc \\
 &= \bar{a}\bar{b}(c) + a\bar{b}(\bar{c}+c) + ab(c) \\
 &= \bar{a}\bar{b}(c) + \bar{a}b(D) + a\bar{b}(1) + ab(C).
 \end{aligned}$$



$$\begin{aligned}
 \textcircled{b} \quad f &= \bar{a}\bar{b}\bar{c} + a\bar{b}\bar{c} + a\bar{b}c + abc \\
 &= \bar{b}\bar{c}(a) + \bar{b}c(\bar{a}+a) + bc(a) \\
 &= \bar{b}\bar{c}(a) + \bar{b}c(1) + b\bar{c}(0) \\
 &\quad + bc(a)
 \end{aligned}$$



Home Work: Implement $f(a, b, c) = \sum(0, 1, 2, 4)$ using b and c as address lines.

Solution:

Problem: Implement the Boolean function $f(a, b, c, d) = \sum(4, 5, 7, 8, 10, 12, 15)$ using 4:1 Mix and external gates if –

- a) a and b are connected to select lines
- b) c and d are connected to select lines.

Solution:

$$\begin{aligned}
 \textcircled{a} \quad f &= \bar{a}b\bar{c}\bar{d} + \bar{a}b\bar{c}d + \bar{a}bc\bar{d} + ab\bar{c}\bar{d} + ab\bar{c}d + abc\bar{d} \\
 &= \bar{a}\bar{b}[0] + \bar{a}b[\bar{c}\bar{d} + \bar{c}d + cd] + ab[\bar{c}\bar{d} + cd] \\
 &= \bar{a}\bar{b}[0] + \bar{a}b[\bar{c}+d] + ab[\bar{c}\oplus d] \quad [:\bar{c}+cd = \bar{c}+d].
 \end{aligned}$$

$$\begin{aligned}
 \textcircled{b} \quad f &= \bar{a}b\bar{c}\bar{d} + \bar{a}b\bar{c}d + \bar{a}bc\bar{d} + ab\bar{c}\bar{d} + ab\bar{c}d + abc\bar{d} \\
 &= \bar{c}\bar{d}[\bar{a}b + a\bar{b} + ab] + \bar{c}d[\bar{a}b] + c\bar{d}[ab] + cd[\bar{a}b + ab] \\
 &= \bar{c}\bar{d}[a+b] + \bar{c}d[\bar{a}b] + c\bar{d}[ab] + cd[b] \quad [:\bar{a}+\bar{a}b=a+b]
 \end{aligned}$$

Home Work: Implement $u = ad + b\bar{c} + bd$ using -

- (i) 16:1 Mux (ii) 8:1 Mux (iii) 4:1 Mux.

Solution:

Try these: (i) Design a full adder using a 4-to-1 multiplexer, a 2-to-1 multiplexer, and a decoder
 (ii) Design a full subtractor using a 4-to-1 multiplexer, a 2-to-1 multiplexer, and a decoder.

MAHESH PRASANNA K., VCET, PUTTUR

ARITHMETIC CIRCUITS**ARITHMETIC BUILDING BLOCKS:**

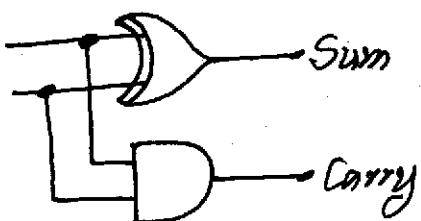
Half-adder, Full-adder and Controller Inverter are three basic circuits that will be used as building blocks.

Half Adder: When we add two binary numbers, we start with the least significant column. This means that, we have to add two bits with the possibility of a carry. The circuit used for this is called a *half-adder*. The half-adder performs binary addition.



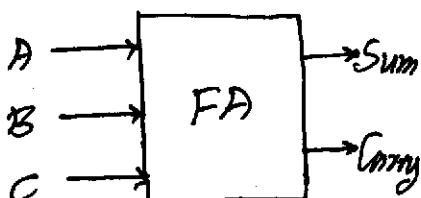
A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$$\begin{aligned} \text{Sum} &= \bar{A}B + A\bar{B} \\ &= A \oplus B \\ \text{Carry} &= AB \end{aligned}$$



Half-Adder Truth Table & Circuit Diagram

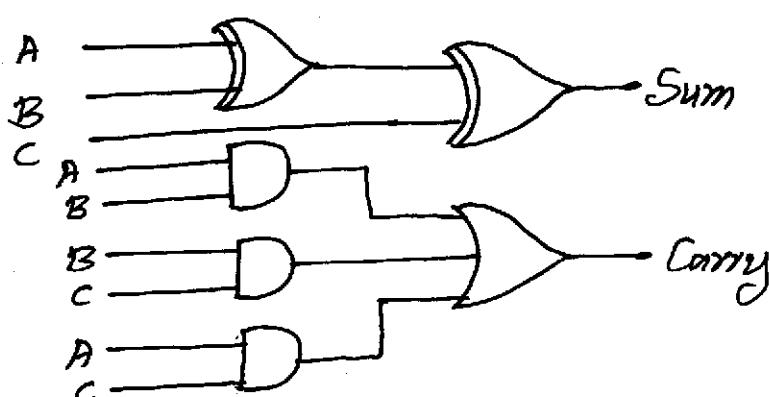
Full Adder: A logic circuit that can add three bits at a time is called a *full-adder*. The third bit is the carry from a lower column. Full-adder performs binary addition on three bits.



A	B	C	Sum	Carry
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

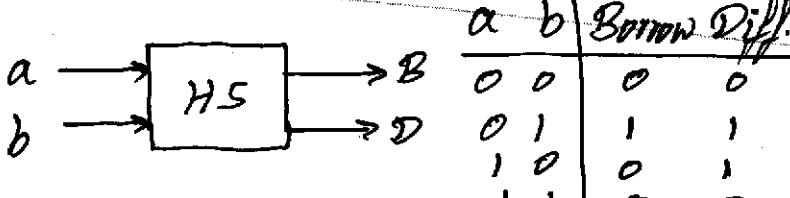
$$\begin{aligned} \text{Sum} &= \sum m(1, 2, 4, 7) \\ &= \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC \\ &= C[\bar{A}\bar{B} + A\bar{B}] + \bar{C}[\bar{A}B + A\bar{B}] \\ &= C[\bar{A}B + A\bar{B}] + \bar{C}[AB + A\bar{B}] \\ &= A \oplus B \oplus C. \end{aligned}$$

$$\begin{aligned} \text{Carry} &= \sum m(3, 5, 6, 7) \\ &= A\bar{B}\bar{C} + A\bar{B}C + AB\bar{C} + ABC \\ &= AB + BC + AC. \end{aligned}$$

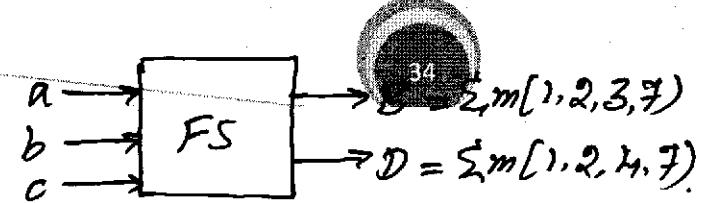


		C
\bar{A}	\bar{B}	0 0
\bar{A}	B	0 1
AB	\bar{B}	1 1
AB	B	0 1

MAHESH PRASANNA K., VCET, PUTTUR



a	b	Borrow	Diff.
0	0	0	0
0	1	1	1
1	0	0	1



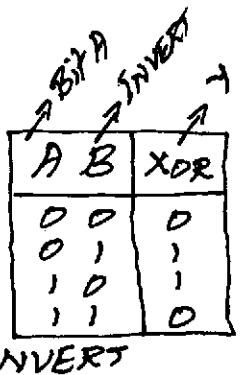
a	b	c	B	D
0	0	0	0	$\sum m(1, 2, 3, 7)$
0	0	1	1	$\sum m(1, 2, 4, 7)$
1	0	0	1	$\sum m(1, 2, 4, 7)$

The general representation of full-adder which adds i -th bit A_i and B_i of two numbers A and B along with the carry $(i-1)$ th bit could be

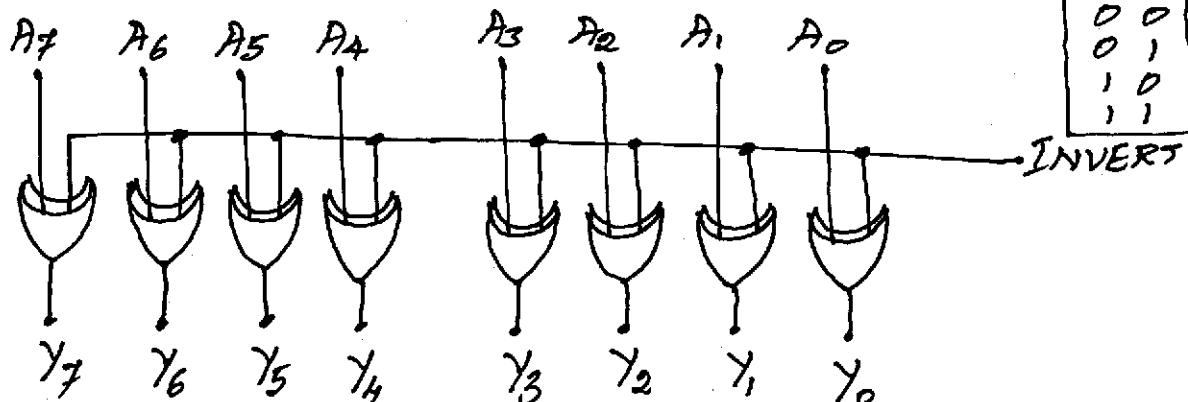
$$S_i = A_i \oplus B_i \oplus C_{i-1}$$

$$C_i = A_i B_i + B_i C_{i-1} + A_i C_{i-1} \quad \text{or} \quad C_i = A_i B_i + (A_i + B_i) C_{i-1}$$

where, S_i and C_i are the sum and carry bits generated from the full-adder.



Controlled Inverter: The following Fig shows a *controlled inverter*.



Controlled Inverter

When INVERT is low, it transmits the 8-bit input to the output; when INVERT is high, it transmits the 1's complement. For example –

If the input number is $A_7 \dots A_0 = 0110\ 1110$, a low INVERT produces $Y_7 \dots Y_0 = 0110\ 1110$.

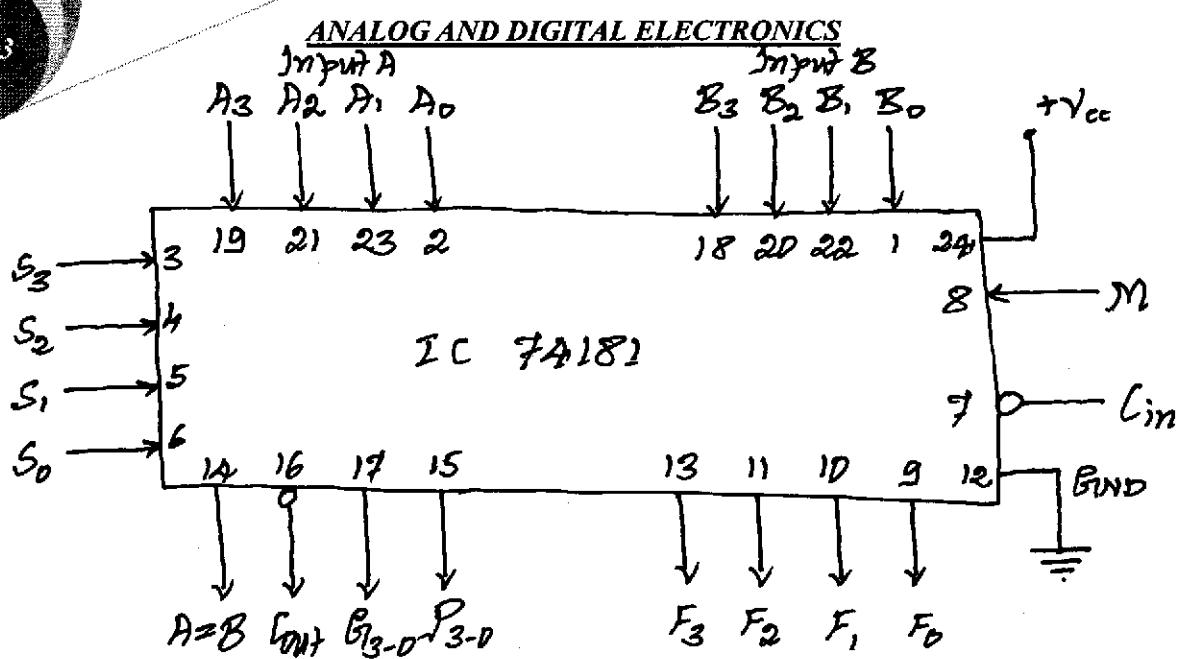
But a high INVERT results in $Y_7 \dots Y_0 = 1001\ 0001$.

The controlled inverter finds application in subtraction. During subtraction, we first need to take 2's complement of the subtrahend. Then, we can add the complemented subtrahend to obtain the answer. With a controlled inverter, we can produce 1's complement and proceed further accordingly.

ARITHMETIC LOGIC UNIT (ALU):

ALU is multifunctional device that can perform both arithmetic and logic function. A mode selector input (M) decides whether ALU performs a logic operation or an arithmetic operation. As an arithmetic unit, along with normal addition, subtraction, etc., it can also perform increment, decrement operations. As logic unit, it can perform usual OR, AND, NOT, Ex-OR, and many more complex logic functions. It also comes with PRESET and CLEAR options, invoking which all the function outputs are made 1 and 0 respectively. ALU is an integral part of Central Processing Unit (CPU) of a computer.

IC 74181 is a 4-bit ALU that can generate 16 different kinds of outputs, in each mode, selected by four selection inputs S_3, S_2, S_1 , and S_0 .



Functional Diagram of ALU IC 74181

S3	S2	S1	S0	M = 1 Logic Function	M = 0 Arithmetic Function $C_{in} = 1$ (For $C_{in} = 0$, add 1 to F)
				F = A'	F = A
0	0	0	0	F = A'	F = A
0	0	0	1	F = (A + B)'	F = A + B
0	0	1	0	F = A'B	F = A + B'
0	0	1	1	F = 0	F = minus 1
0	1	0	0	F = (AB)'	F = A plus (AB')
0	1	0	1	F = B'	F = (A + B) plus (AB')
0	1	1	0	F = A \oplus B	F = A minus B minus 1
0	1	1	1	F = AB'	F = AB' minus 1
1	0	0	0	F = A' + B	F = A plus (AB)
1	0	0	1	F = (A \oplus B)'	F = A plus B
1	0	1	0	F = B	F = (A + B') plus (AB)
1	0	1	1	F = AB	F = AB minus 1
1	1	0	0	F = 1	F = A plus A
1	1	0	1	F = A + B'	F = (A + B) plus A
1	1	1	0	F = A + B	F = (A + B') plus A
1	1	1	1	F = A	F = A minus 1

ALU IC 74181 Truth Table

Problem: Show how $A > B$ output can be generated in IC 74181 ALU. Also show how $A \geq B$ condition can be checked. Show how bits of input A shifted to left by one will appear at output F in IC 74181.

15CS33

Note that, the data inputs A and B are active high signals and C_{in} and C_{out} are active low signals. The outputs C_{out} , G_{3-0} , and P_{3-0} are useful in addition and subtraction of more than 4-bits.

Logic operations are done bit-wise, by making $M = 1$ and choosing appropriate select inputs. Carry is inhibited for $M = 1$. For example, to perform AND operation between two 4-bit numbers (1011) A and (0111) B, make $S = 1011$ (refer truth table) and $M = 1$ to choose the logic function. The output will be $F = 0011$.

For arithmetic operations, $M = 0$ to be chosen. For example, to add decimal numbers 6 with 4, we have to place 0110 for 6 (at A) and 0100 for 4 (at B). Then with $S = 1001$ (refer truth table), $C_{in} = 1$ (active low), the output generated is, $F = 1010$ – decimal equivalent of 10.

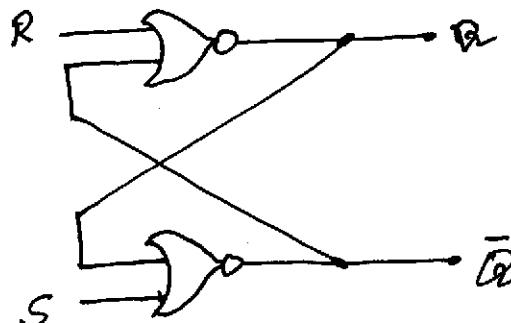
FLIP-FLOPS

- o Any device or circuit that has two stable states is said to be *bistable*. For example, a toggle switch – a switch is also said to have *memory*, since it will remain as set until someone changes its position.
- o A *flip-flop* is a bistable electronic circuit that has two stable states – i.e. its output is either 0 or +5 V.
- o The flip-flop also has memory, since its output will remain as set until something is done to change it.
- o The flip-flop is often called a *latch*, since it will hold (or latch) in either stable state.

SR FLIP-FLOPS:

NOR-Gate Latch:

Two 2-input NOR gates are connected as shown in the following Fig to form a flip-flop. The flip-flop has two outputs, Q and \bar{Q} . There are two inputs to the flip-flop defined as S and R. The input/output possibility for this SR flip-flop is also given below.

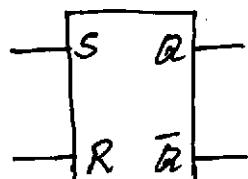


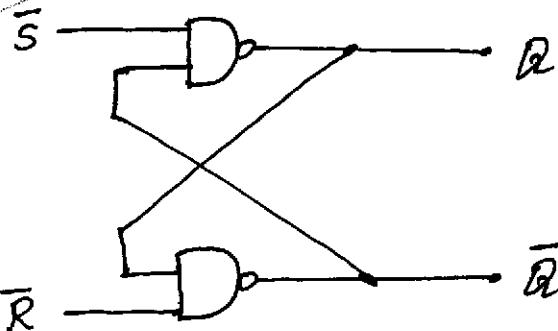
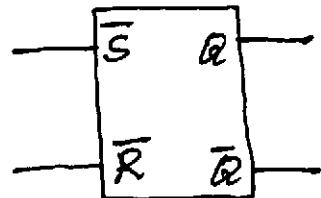
S	R	Q	Action
0	0	Last State	No Change
0	1	0	RESET
1	0	1	SET
1	1	?	Forbidden

NOR-Gate SR Flip-Flop, Symbol & Its Truth Table

NAND-Gate Latch:

A slightly different latch can be constructed by using NAND gates, as shown in the following Fig.



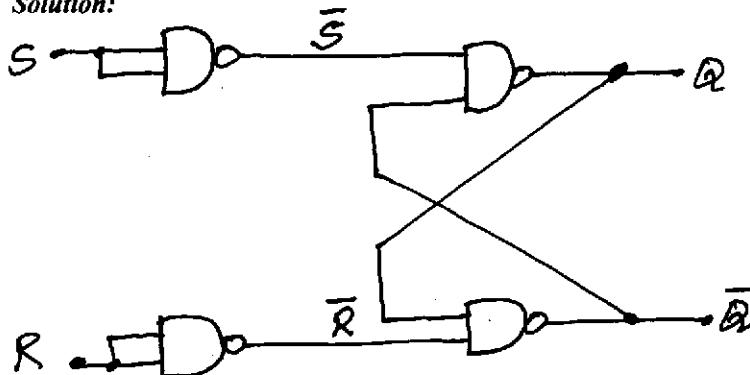


\bar{S}	\bar{R}	Q	Action
1	1	Last State	No Change
1	0	0	RESET
0	1	1	SET
0	0	?	Forbidden

NAND-Gate $\bar{S}\bar{R}$ Flip-Flop, Symbol & Its Truth Table

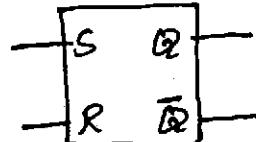
Problem: Show how to convert the $\bar{S}\bar{R}$ flip-flop into RS flip-flop.

Solution:



S	R	Q	Action
0	0	Last State	No Change
0	1	0	RESET
1	0	1	SET
1	1	?	Forbidden

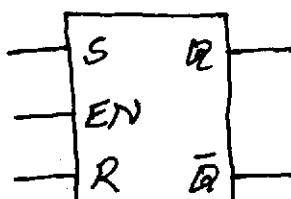
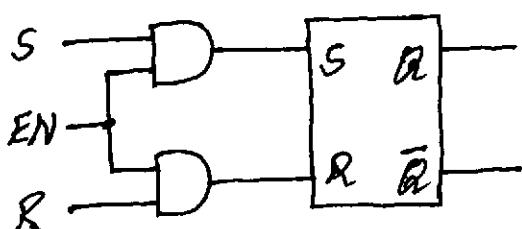
NAND-Gate SR Flip-Flop, Symbol & Its Truth Table



GATED FLIP-FLOPS:

Clocked RS Flip-Flop: The addition of two AND gates at the S and R inputs (as shown in the following Fig) will result in a flip-flop that can be enabled or disabled.

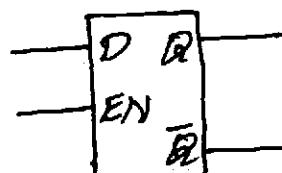
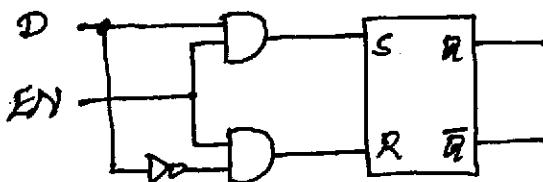
- ✓ When the ENABLE input is low, the AND gate (both) outputs must be low. Hence, neither R nor S will have any effect on the flip-flop output Q.
- ✓ When the ENABLE (EN) input is high, information at the S and R inputs will be transmitted directly to the outputs. Hence, the output will change in response to the inputs as long as the ENABLE is high.



EN	S	R	Q_{n+1}	Action
0	X	X	Q_n	No Change
1	0	0	Q_n	No Change
1	0	1	0	RESET
1	1	0	1	SET
1	1	1	?	Forbidden

Clocked SR Flip-Flop, Symbol & Its Truth Table

Clocked D Flip-Flop: A D flip-flop is a bistable circuit whose D input is transferred to the output when EN is high.

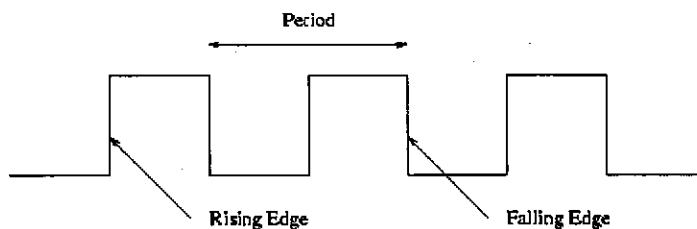


EN	D	Q_{n+1}
0	X	Q_n (No Change)
1	0	0
1	1	1

Clocked RS Flip-Flop, Symbol & Its Truth Table

NOTE:

1. Clock cycle:

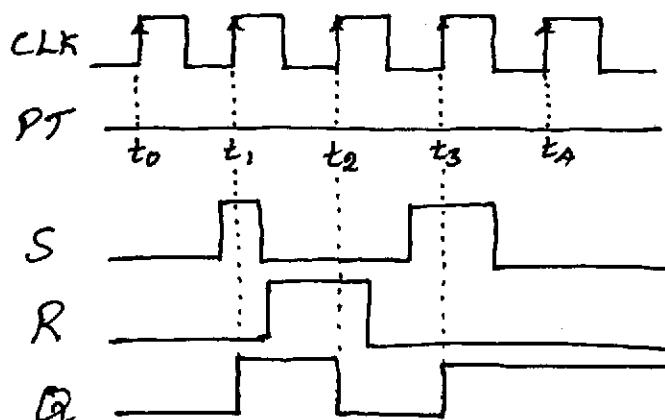
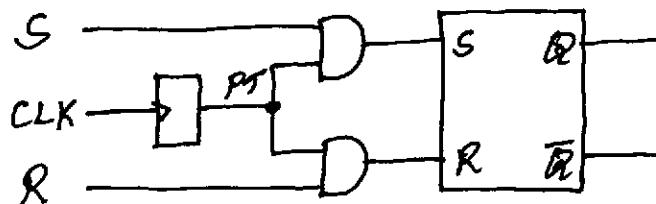
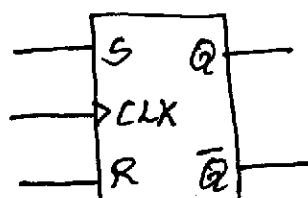


2. A low-to-high transition is *positive transition (PT)*. A circuit that changes state at this time is known as *positive-edge triggered*.
3. A high-to-low transition is *negative transition (NT)*. A circuit that changes state at this time is known as *negative-edge triggered*.

EDGE-TRIGGERED SR FLIP-FLOP:

Positive-Edge-Triggered SR Flip-Flops:

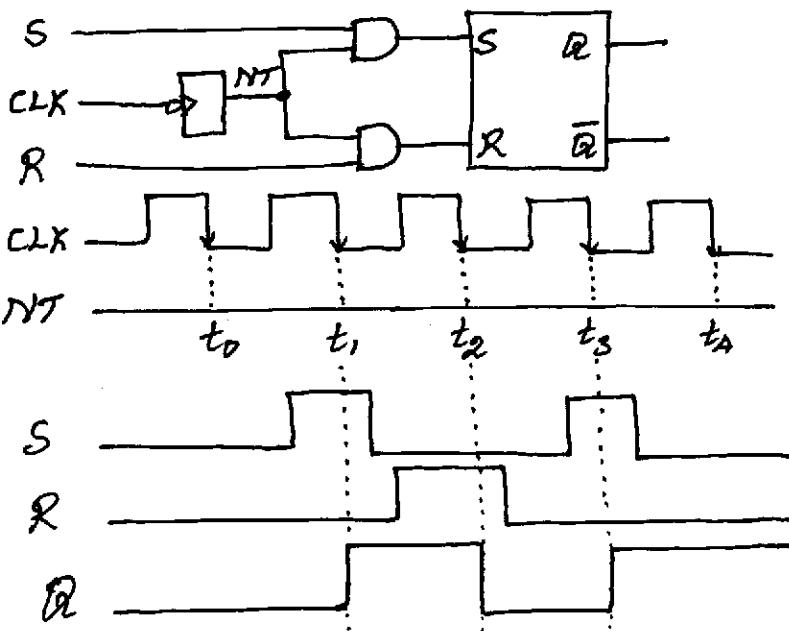
C	S	R	Q_{n+1}	Action
↑	0	0	Q_n	No Change
↑	0	1	0	RESET
↑	1	0	1	SET
↑	1	1	?	Illegal



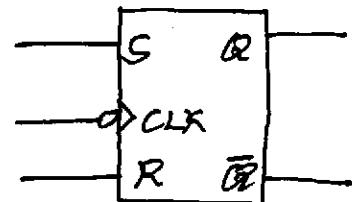
Positive-Edge-Triggered SR Flip-Flop, Symbol, Truth Table, & Waveform

MAHESH PRASANNA K., VCET, PUTTUR

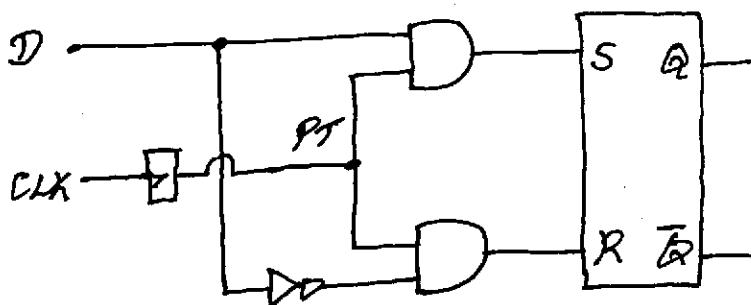


Negative-Edge-Triggered SR Flip-Flops:

EN	S	R	Q_{n+1}	Action
↓	0	0	Q_n	No Change
↓	0	1	0	RESET
↓	1	0	1	SET
↓	1	1	?	Illegal

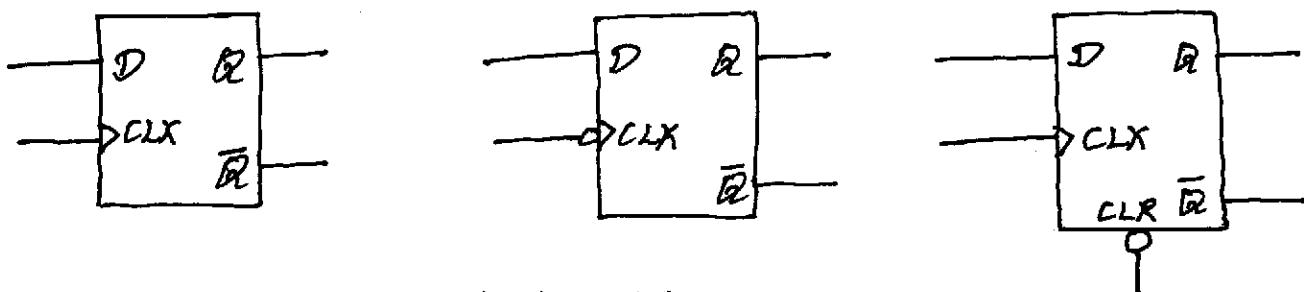


Negative-Edge-Triggered SR Flip-Flop, Symbol, Truth Table, & Waveform

EDGE-TRIGGERED D FLIP-FLOPS:

C	D	Q_{n+1}
0	X	Q_n (No Change)
↑	0	0
↑	1	1

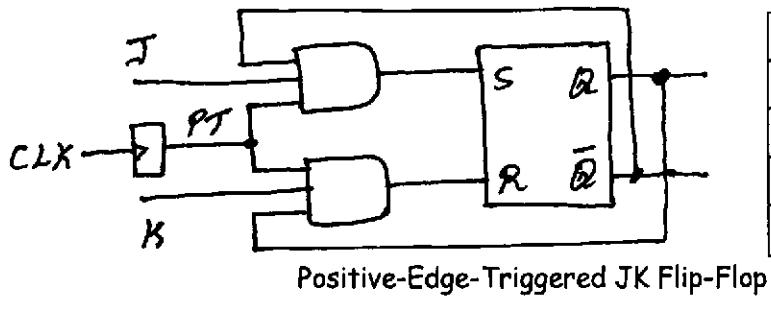
Positive-Edge-Triggered D Flip-Flop & Its Truth Table



D Flip-Flop Symbols

EDGE-TRIGGERED JK FLIP-FLOPS:

Setting $S = R = 1$ with an edge-triggered SR flip-flop forces both Q and \bar{Q} to the same logic level. This is an illegal condition; hence, it is not possible to find the final state of Q . The JK flip-flop accounts for this illegal input.

Positive-Edge-Triggered JK Flip-Flops:

C	J	K	Q_{n+1}	Action
↑	0	0	Q_n	No Change
↑	0	1	0	RESET
↑	1	0	1	SET
↑	1	1	\bar{Q}_n	Toggle

Working:

- When J and K both are low, both AND gates are disabled. Therefore, clock pulses have no effect. Hence, Q retains its previous value.
- When J is low and K is high, the upper AND gate is disabled. So, there is no way to set the flip-flop, and the only possibility is reset.

If Q is low in the previous state, same low value will be maintained as the next positive clock edge arrives. If the Q is high in the previous state, the lower AND gate passes a RESET pulse as soon as the next positive clock edge arrives. This forces Q to become low.

- When J is high and K is low, the lower gate is disabled. So, it's impossible to reset the flip-flop.

If Q is low in the previous state, \bar{Q} is high; therefore the upper AND gate passes SET pulse on the next positive clock edge. This drives Q into high state. If Q is high in the previous state, \bar{Q} is low; therefore, the SET signal through the upper AND gate will be 0. This maintains Q in the high state.

- When J and K are both high, it's possible to set or reset the flip-flop. If Q is high, the lower AND gate passes a RESET pulse on the next PT. If Q is low, the upper AND gate passes a SET pulse on the next PT. Either the way, the Q changes to the complement of the previous state. Hence, flip-flop will toggle.

JK Flip-Flop Symbols

By: MAHESH PRASANNA K.,
DEPT. OF CSE, VCET.
