# Electronics (II)

Page No.
Date: / /

## Unit (I)

Binary numbers, Octal numbers, Hexadecimal numbers, Interconversions of numbers, Binary addition, subtraction, multiplication division, Hexadecimal addition, subtraction, Octal addition, subtraction signed numbers, 1's complement arithmetic, 2's complement arithmetic, 9's complement arithmetic, BCD code & arithmetic, Gray code, excess-3 code. Positive & negative logic designations, OR gate, AND, NOT, NAND, NOR, XOR gates, circuits & Boolean identities associated with gates, Boolean algebra De Morgan's Law, Sum of products & product of sums expressions, Minterm, Maxterm, K-Maps, don't care condition, deriving SOP & POS expressions from truth tables.

## Unit (II)

Combinational digital circuits: Binary adder. half & full adders, Decoders, Multiplexer Demultiplexer, Encoders, ROM & it's application (binary, BCD, Excess-3 code, Gray code & BCD to seven segment), Digital comparator, Parity checker & generator.

Sequential digital circuits: 1-bit memory, Flip-flops - RS, JK, master slave JK, T-type & D-type FF, shift register & app., Asynchronous counters & Synchronous counters.

## Unit (III)

Metal oxide semiconductor FET, enhancement mode transistor, depletion mode transistor, p-channel & n-channel device, MOS invertors - static inverter, dynamic inverter, 2 phase inverter, MOS NAND gate, NOR gates, complementary MOS FET Technology, CMOS inverter, CMOS NOR & NAND gates, MOS shift register & RAM

Unit (IV)

Fundamentals of Modulation, Frequency spectra in AM modulation, power in AM modulated class C amplifier, efficiency modulation, freq. conversion, SSB system, Balanced modulation, filtering the signal for SSB, phase shift method, product detector, Pulse modulation, Microwave devices: Resonant cavity, Klystrons & Magnetron

## Unit (4)

### Electronic Communication System

The communication process may be defined as the process in which the information is transferred from one place to another. The point from where the information is transferred is called transmitter, whereas the point where the information is received is called receiver.

(i) **Information & Message :-**

The information concept is main plank of the communication system. Information is specified as a measurable quantity in the technical description and the ~~anaytical~~ analytical treatment of the communication system. Apart from the information, the major concern is message.

(ii) **Signal :-** The message produced from the source is not electrical. Therefore, for an electrical system the message must be converted into electrical form. The transducer is used to convert the message into signal. The signal is a time varying quantity. It may be voltage or current corresponding to the message. After the processing and the transmission through communication system, the signal is received at the destination. The signal is converted into the original form with the help of transducer.

(iii) **Channel :-** The channel is basically a medium which is used for transmission and reception. The medium may be wire, coaxial cable, a wave guide, an optical fibre or a radio link.

(iv) Noise :- The signal that is being sent or transmitted is mixed by undesirable signals along the path. This undesired signal can be termed as noise. The noise may be either external or internal.

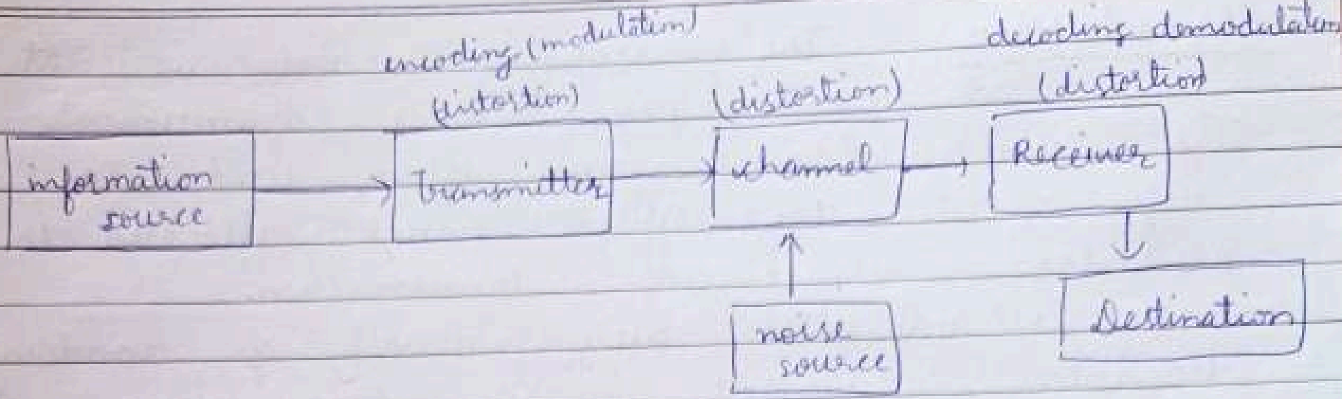
(a) External noise :- It includes interference from signals transmitted on nearby channels, man-made noise generated by wrong switch connections, sound from automobiles etc. With proper care this noise can be minimised.

(b) Internal noise :- It is caused by the thermal motion of e⁻ in conductors, random emission & due to diffusion or recombination of charged carriers in device. It can not be completely eliminated.

⇒ Classification of communication system

(1) According to the nature of source of information.
(2) According to the mode of transmission
(3) According to the type of modulation.
(4) According to transmission channel.

⇒ Block Diagram of an electronic communication system




## Terminology

Information source :- The communication system exists to convey a message. This message comes from the information source, which originates it, in the sense of selecting one message from a group of messages. So, information itself is that which is conveyed.

TRANSMITTER :- Unless the message arriving from the information source is electrical in nature, it will be unsuitable for immediate transmission. A transmitter is required to process and possibly encode the incoming information so as to make it suitable for transmission & subsequent reception.

OR In a communication system, the trans-mitter transmits the message. The original message is not suitable for transmission & so it is converted into the varying electrical signal, called the message signal.
e.g:- a microphone converts sound signals into electrical signal.

Date: / /

Channel Noise :- The acoustic channel is not used for long distance communication. The term channel is often used to refer to the freq. range allocated to a particular service or transmission.
Noise is unwanted energy, usually of random character, present in a transmission system due to a variety of causes. Noise may interfere with the signal at any point in a communication system, but it will have it's greatest effect when the signal is weakest.

Receiver :- Its most important function is demodulation. The purpose of a receiver and the form of its O/P influence its construction.

OR The purpose of receiver is to reconstruct the original message after its propagation through the communication channel. The process used for separating the message from modulated wave is called demodulation. The exact design of receiver depends upon whether the transmission was analog or digital.

⇒ // Modulation

Modulation is a process in which certain characteristics such as amplitude, frequency or phase of a high freq. sine wave varied with the instantaneous value of message (called modulating signal).

A sinusoidal carrier wave is a sine wave or cosine wave & is expressed as

$$V_c(t) = E_c \sin(w_c + \phi)$$

where $V_c$ = instantaneous value of the sine wave or carrier wave.
$E_c$ = max amplitude
$w_c$ = angular freq.
$\phi$ = phase angle w.r.t. some reference.

Any of these three parameters of carrier wave may be varied by the modulating signal. depending upon these three parameters the modulation may be classified into three categories :-

(I) Amplitude Modulation
(II) Frequency Modulation
(III) Phase Modulation.

→ Need of Modulation

Communication system is used to transmit the message or information signals. These signals are called pass band signals. No signal is a single freq. signal but spreads over a range of frequencies. Range of frequency over which a signal is spread is called band width of the signal. Pass band signals represent the band of freq. coming out of the source of information. If the frequency of signal to be transmitted is ($< 20$KHz) then it is not possible to transmit the signal over long distances because of following reasons :-

(I) Size of Antenna or areal

For efficient transmission & reception, the transmitting & receiving antenna should have height at least equal to quarter wavelength of freq. used.

For eg:-

as we know $c = f\lambda$

as $f = 1 MHz$ & $c = 3 \times 10^8$ m/s

$$\therefore \lambda = \frac{c}{f} = \frac{3 \times 10^8}{1 \times 10^6} = 300 m$$

Hence, effective ht. of antenna $= \lambda/4$

$$= \frac{300}{4} = 75m$$

Now, if a signal of freq. 15 kHz is to be transmitted, the ht. of antenna would be 5000 ~~times~~ metres which is quite impractical. Hence, there is a need of converting the information contained in original low freq. base band into high freq. before it is transmitted.

(II) Power radiated by the antenna :- It has been found that power radiated by an antenna is inversely proportional to the square of the wavelength of the waves being transmitted i.e.

$$\boxed{\text{Power radiated} \propto (1/\lambda)^2}$$

Hence, for shorter wavelength, power radiated will be large & for longer wavelength power radiated will be small. For good transmission it is required that power radiated by the antenna should be high is possible only if there is high freq. transmission.

(iii) An unwanted carrier has a const amplitude, a const. freq. & a const. phase relation w.r.t. some reference but a message consists of ever-varying quantities.

(iv) All sound/ audio signals are concentrated within the range from 20 Hz to 20 KHz so that all signals from different sources would be inseperably mixed up.

(v) Base Band Signals :- The basic signal is sometimes called base band signal & the spectral range occupied by the the basic signal is called the base band freq. range or simply base band.

## Amplitude Modulation

In amplitude modulation, the amplitude of the carrier wave is varied with the instantaneous amplitude of the modulating signal. In amplitude modulation, only the amplitude of the carrier wave is changed in accordance with the intensity of the signal, however the freq. of modulated wave remain the same.

Date : / /

Amplitude modulation is done by an electronic circuit called modulator. Amplitude modulation is widely used for commerical broadcast of voice signals. Carrier wave freq. lies in the range of 0.5 MHz to 2 MHz.

Imp. points of AM :-

(i) The amplitude of carrier wave changes acc to the amplitude of the signal

(ii) The amplitude variations of the carrier wave is at the signal freq.

(iii) The freq of the amplitude modulated wave remains the same i.e. carrier freq.

Let the carrier vol & modulating vol. $v_c$ & $v_m$ be given as

$\boxed{v_c(t) = A_c \sin w_c t}$ & $\boxed{v_m(t) = A_m \sin w_m t}$

Frequency Spectra of AM

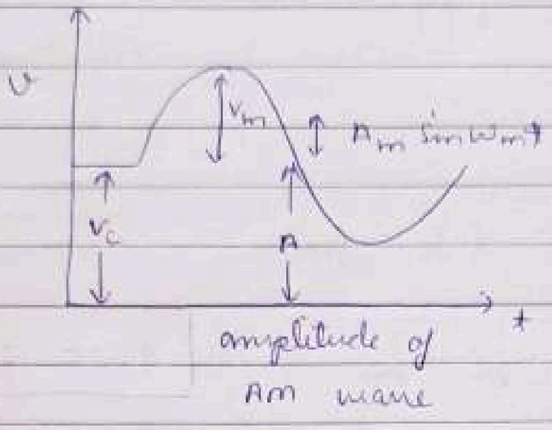
From the definition of AM, we know that $v_c$ has to be made proportional to $v_m \sin w_m t$ when the carrier is amplitude modulated.

Freq. present in the AM wave are the carrier freq. & the first pair of sideband frequencies where sideband freq. is defined as

$$f_{sb} = f_c \pm n f_m$$

where $n = 1$ for 1st pair of side band

when an carrier is amplitude modulated, the proportionality const is made equal to unity.




$$v_{Am}(t) = (A_c + v_m) \sin w_c t$$
$$= (A_c + A_m \sin w_m t) \sin w_c t$$

$$= A_c \sin w_c t + A_m \sin w_m t \sin w_c t$$

$$= A_c \sin w_c t + A_m \left[ \frac{\cos(w_c - w_m) t - \cos(w_c + w_m) t}{2} \right]$$

$$v_{Am}(t) = A_c \left[ \sin w_c t + \frac{A_m}{2A_c} (\cos(w_c - w_m) t - \cos(w_c + w_m) t) \right]$$

$$v_{Am}(t) = A_c \left[ \sin w_c t + \frac{m}{2} \cos(w_c - w_m) t - \frac{m}{2} \cos(w_c + w_m) t \right]$$

where $m$ = modulation index
$m = A_m / A_c$

From above eq^n it can thus be noted that AM contains 3 freq. (i) unmodulated carrier freq. $f_c$ & 2 side bands with $f_c + f_m$ & $f_c - f_m$. It must also be noted that the process of AM is just adding to the carrier not changing it.

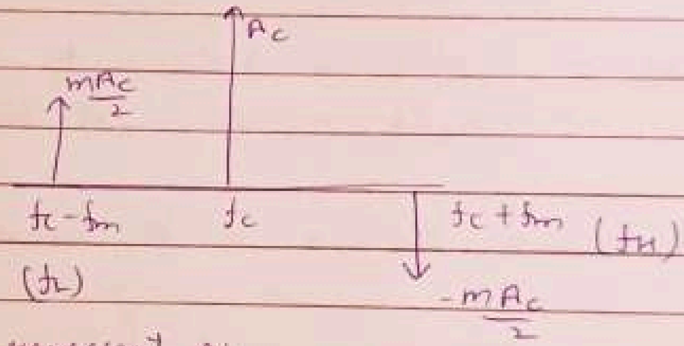
The Bandwidth required for AM is twice the modulating freq.

$BW = f_H - f_L$

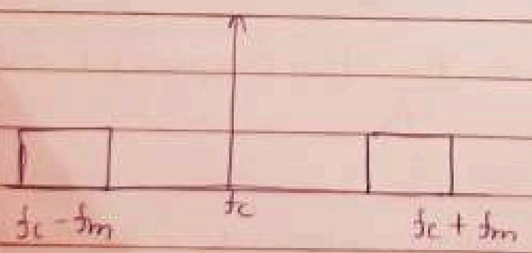
$BW = f_c + f_m - (f_c - f_m)$

$= \cancel{f_c} + f_m - \cancel{f_c} + f_m \Rightarrow \boxed{BW = 2f_m}$

Freq. Spectra




it can also represent as




∵ $f_m$ contains a band of freq. instead of single freq.

$f_c$ = freq. of carrier signal

$f_m$ = freq. of message signal

$$\boxed{\begin{aligned} V_{max} &= A_c + A_m \\ V_{min} &= A_c - A_m \end{aligned}}$$

∵ $-1 < \sin x < 1$

$m < 1$ ⇒ under modulation
$m = 1$ ⇒ critical modulation
$m > 1$ ⇒ over modulation

Representation of AM :-

Modulation Factor :-
An important consideration in amplitude modulation is to describe the extent to which amplitude of carrier wave is changed by the signal. This is described by a factor called Modulation Factor.

" The ratio of change of amplitude of carrier wave to the amplitude of normal carrier wave is called modulation factor (m)".

$$m = \frac{\text{change in amplitude of carrier wave}}{\text{Normal (unmodulated) carrier wave amplitude}}$$

$$V_m = \frac{V_{max} - V_{min}}{2}$$

Date : / /

$$V_c = V_{max} - V_m = V_{max} - \frac{V_{max} - V_{min}}{2}$$

$$= \frac{-V_{max} + V_{min} + 2V_{max}}{2}$$

$$V_c = \frac{V_{max} + V_{min}}{2}$$

$$\therefore m = \frac{V_m}{V_c} = \frac{A_m}{A_c} = \frac{V_{max} - V_{min}}{V_{max} + V_{min}}$$

$$\boxed{m = \frac{V_{max} - V_{min}}{V_{max} + V_{min}}}$$

**Importance of Modulation Factor :-**

Modulation factor is very important since it determines the strength & quality of transmitted signal. In an AM wave, the signal is contained in the variations of the carrier amplitude. When the carrier is modulated to a small degree (i.e. small m), the amount of carrier amplitude variation is small. Consequently, the audio signal being transmitted will not be very strong. The greater the degree of modulation (m) the stronger & clearer will be the audio signal.

If the carrier is overmodulated (i.e; m>1) distortion will occur during reception.

$\therefore$ the degree of modulation should never exceed 100% (i.e. m=1)

## ⇒ Power Relations :-

Double Side Band Full Carrier (DSBFC)

The power of AM wave can be calculated as the sum of contributed by the unmodulated carrier wave and the two side bands. Obviously, the modulated wave contains more power than the unmodulated carrier wave.

As the amplitude of sidebands depends upon the modulation index m, therefore, the total power of AM also depends upon the modulation index.

Let R be the resistance of antenna through which power is dissipated. The total power of AM wave is thus given by

$$\boxed{P_t = P_c + P_{LSB} + P_{USB}}$$

$$\because V_{AM}(t) = \underbrace{A_c \sin w_c t}_{\text{carrier } (f_c)} + \underbrace{\frac{mA_c}{2}\cos(w_c - w_m)t}_{\substack{LSB \\ (f_c - f_m)}} - \underbrace{\frac{mA_c}{2}\cos(w_c + w_m)t}_{\substack{USB \\ (f_c + f_m)}}$$

$\because$ Power $P = \dfrac{V_{rms}^2}{R}$

Let $A_c$ be the peak vol.

$\therefore$ rms value $V_{rms} = \dfrac{A_c}{\sqrt{2}}$

$$\therefore P_c = \left(\frac{A_c}{\sqrt{2}}\right)^2 \cdot \frac{1}{R} \Rightarrow \boxed{P_c = \frac{A_c^2}{2R}}$$

$$P_{LSB} = P_{USB} = \left(\frac{mA_c/2}{\sqrt{2}}\right)^2 \Big/ R$$

$$\boxed{P_{LSB} = P_{USB} = \frac{m^2 A_c^2}{8R}}$$

$$P_{LSB} = P_{USB} = \frac{m^2 A_c^2}{8R} = \frac{m^2}{4}\left(\frac{A_c^2}{2R}\right)$$

$$\boxed{P_{LSB} = P_{USB} = \frac{m^2}{4} P_c}$$

∴ Total transmitted Power

$$P_t = P_c + P_{LSB} + P_{USB}$$

$$P_t = \frac{A_c^2}{2R} + \frac{m^2 A_c^2}{8R} + \frac{m^2 A_c^2}{8R}$$

$$P_t = \frac{A_c^2}{2R} + \frac{m^2 A_c^2}{4R} \quad \text{or} \quad P_t = \frac{A_c^2}{2R}\left(1 + \frac{m^2}{2}\right)$$

$$\boxed{P_t = P_c\left(1 + \frac{m^2}{2}\right)}$$

This eq$^n$ can also be used to calculate modulation index (m) in terms of total power of AM wave transmitted & the power of unmodulated carrier wave.

The max. power in AM wave will be when the value of m will be unity i.e. $m = 1$

$$\therefore P_t(\text{max}) = \left(1 + \frac{1}{2}\right) P_c = 1.5\, P_c$$

Fraction of power carried by carrier wave

$$\frac{P_c}{P_t} \times 100\% = \frac{P_c}{P_c\left(1 + \frac{m^2}{2}\right)} \times 100\% = \frac{100\%}{1 + \frac{m^2}{2}}$$

for $m = 1$ ⇒ $\frac{2}{3} \times 100\% = \frac{200}{3} = 66.6\%$

## Double side band suppress carrier (DSBSC)

By using filter (BRF) the carrier wave can be blocked, only side bands will allow to pass. As the carrier wave is unmodulated so the power carried by it is useless, so we have to reject carrier wave freq.

In DSBSC

$$P_t = P_{USB} + P_{LSB}$$

$$P_t = \frac{m^2 A_c^2}{8R} + \frac{m^2 A_c^2}{8R}$$

$$\boxed{P_t = \frac{m^2 A_c^2}{4R} = \frac{m^2}{2} P_c}$$

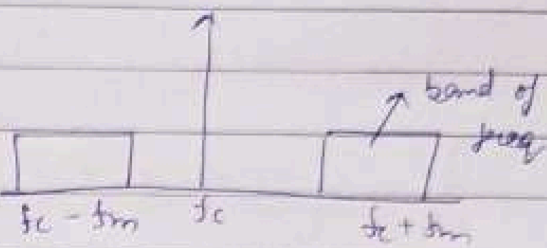
## Single side band suppress carrier (SSB)

LSB & USB have same message signal & both have same magnitude of amplitude. As the power carried by side bands is equal ∴ The total power carried by both side bands is double that of single side band viz useless. So, we have to use filter (LPF for blocking $f_c$ & $f_c + f_m$ / HPF for $f_c, f_c - f_m$) to block one side band freq.

$$\therefore \boxed{P_t = \frac{m^2 A_c^2}{8R}} \quad \& \quad \boxed{P_t = \frac{m^2 P_c}{4}}$$

⇒ If message signal has more than single freq.
i.e. $f_{m_1}$, $f_{m_2}$, $f_{m_3}$

| LSB | USB |
|---|---|
| $f_c - f_{m_1}$ | $f_c + f_{m_1}$ |
| $f_c - f_{m_2}$ | $f_c + f_{m_2}$ |
| $f_c - f_{m_3}$ | $f_c + f_{m_3}$ |




If there are n numbers of modulating signals then the resulting modulation index is given by

$$m_t = \sqrt{m_1^2 + m_2^2 + m_3^2 + \cdots}$$

⇒ Generation of AM

Balance Modulator (DSBSC) (Non-linear device transistor & diode ($i_1 = av_1 + bv_1^2$))

A balanced modulator circuit in the form of fig. is employed to generate an AM signal with-out carrier and provide the first step in SSB generation. Operation occurs over a device region having a transfer characteristics expressible by

$$i_c = aV_1 + bV_1^2$$

$$\& \quad i_c' = aV_2 + bV_2^2$$

$$V_1 = V_c(t) + V_m(t) \quad \& \quad V_2 = V_c(t) - V_m(t)$$

$\therefore i_c = av_1 + bv_1^2 = a(V_c(t) + V_m(t)) + b(V_c(t) + V_m(t))^2$

& $i_c' = av_2 + bv_2^2 = a(V_c(t) - V_m(t)) + b(V_c(t) - V_m(t))^2$

$\because$ $i_c$ & $i_c'$ are in opp. dirⁿ

$\therefore$ o/p will be

$V_0 = k(i_c - i_c')$

$V_0 = k\{a\cancel{V_c(t)} + aV_m(t) + bV_c^2(t) + bV_m^2(t) + 2bV_c(t)V_m(t)$
$- a\cancel{V_c(t)} + aV_m(t) - bV_c^2(t) - bV_m^2(t) + 2bV_c(t)V_m(t)\}$

$V_0 = k\{2aV_m(t) + 4bV_c(t)V_m(t)\}$

$V_0 = 2k\{aV_m(t) + 2bV_c(t)V_m(t)\}$

$\boxed{V_0 = 4kbV_c(t)V_m(t)}$

$\because$ $f_m$ will be very small than $f_c$ so neglected & it will be blocked by centre tap ckt (last LC)

If we change the polarity of I/P, then

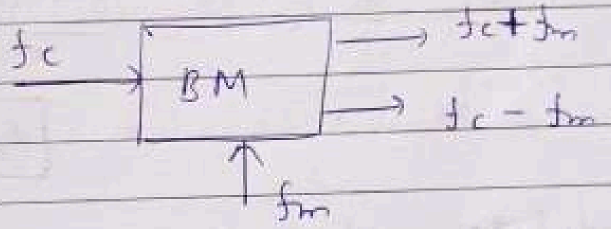
$V_1 = V_c(t) + V_m(t)$ & $V_2 = -V_c(t) + V_m(t)$

(we get full band double side)

DSBFC

$$\boxed{V_0 = k\left[2aV_c + 4bV_cV_m\right]}$$

Representation of B.M.




ques ① A 400 W carrier is amplitude modulation to a depth of 100%. Calculate total power in case of SSB technique.

→ $P_c = 400\ W$ , $m = 1$ (modulation depth)

$P_t = ?$ in SSB.

$$\because \boxed{P_c = 400\ W = \frac{A_c^2}{2R}}$$

In SSB $\quad P_t = \frac{m^2}{4} P_c = \frac{1}{4} \times \cancel{400}^{100} = 100\ W$ Ans

How much power saving is achieve for SSB compare to AM?

DSBFC $\because \quad P_t = P_c\left(1 + \frac{m^2}{2}\right) = 400\left(1 + \frac{1}{2}\right)$

$$P_t = \cancel{400}^{200} \times \frac{3}{\cancel{2}} = \boxed{600\ W = P_t}$$

Date : / /

∴ Power saving $600 - 100 = 500$ W Ans

DSBSC $P_t = \frac{m^2}{2} P_c = \frac{1}{2} \times 400 = 200W$

Power saving $600 - 200 = 400$ W

If depth of modulation is changed to 75%, then how much power in watt is required for transmitting SSB waves.

$m = 0.75$

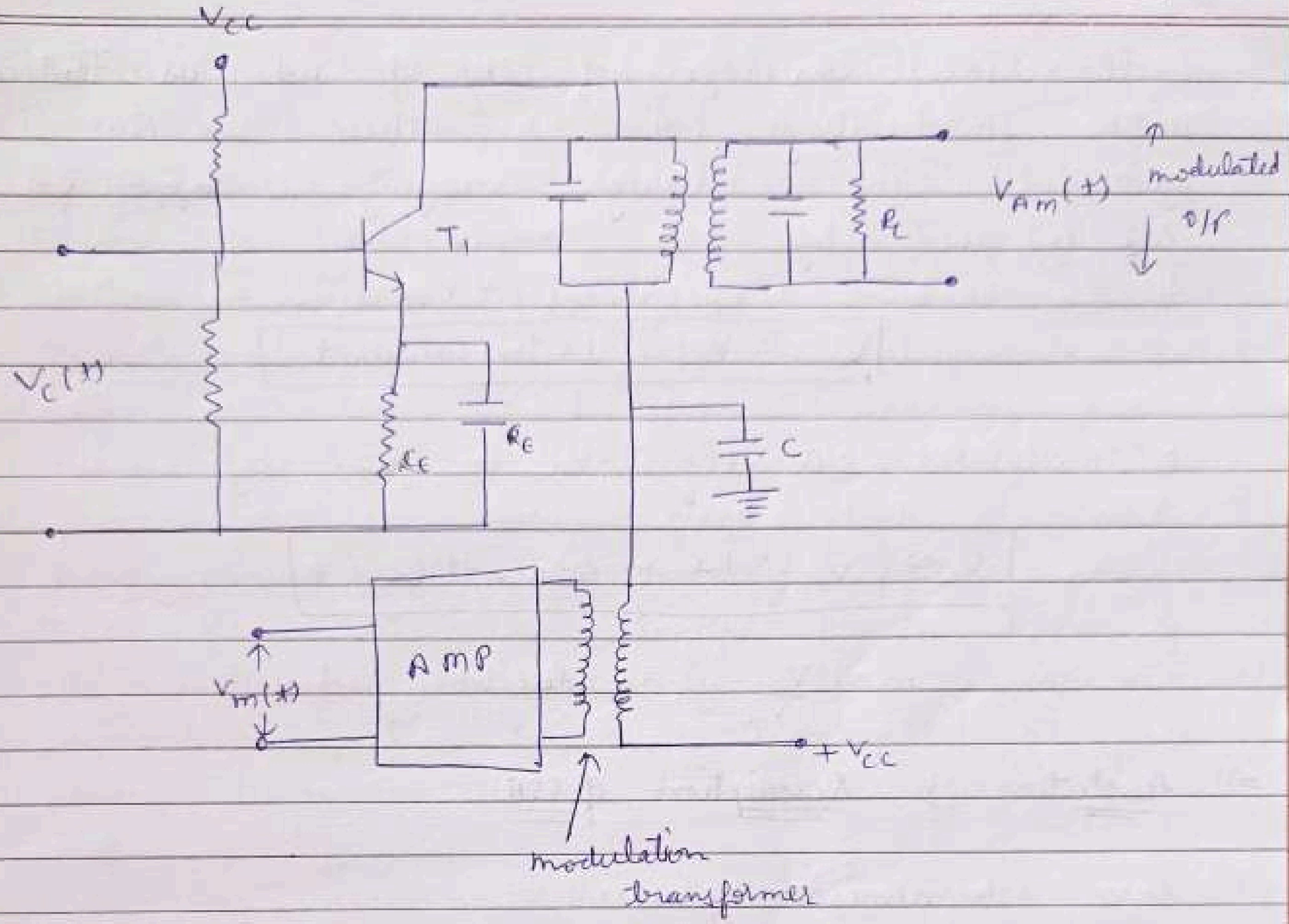
SSB $P_t = \frac{m^2}{4} P_c = \frac{(0.75)^2}{4} \times \cancel{400}^{100}$

$P_t = 0.5625 \times 100 \Rightarrow \boxed{P_t = 56.25 \text{ W}}$ Ans

(1) Collector Modulator class C

Modulator :- Amplitude of the AM carrier waves changes in the same way as that of the modulating or message signal. Modulators can be designed by using diodes or transistors. In transistorised modulators, modulating signal can be fed to any of three elements in RF power amplifiers resulting in base modulator, collector modulator & emitter modulator.

The efficiency is more in collector modulator as comparing with base & emitter modulator i.e we will study class C collector modulator




$T_1$ forms a RF amplifier in class C mode.

$\boxed{\text{AMP}}$ is a class B amplifier used to amplify base band signal.

The unmodulated carrier is amplified by the class C modulated amplifier & it's amplitude remains constant equal to $V_{CC}$ as no vol appears across the modulation transformer in the absence of modulated signal/vol.

When a modulating vol. $A_m \cos \omega_m t$ appears across the modulation transformer it is added with supply vol. $V_{CC}$. The net effect is the slow variation ($\because f_c > f_m$) in supply vol. $V_{CC}$. Slow because as compared to carrier freq., the modulating freq. is small. The envelope of the o/p vol. is identical with modulating vol. & thus an AM is generated. The slowly varying, supply vol. $V_c$ is given by

$$V_c = V_{cc} + V_m \cos w_m t$$

$$\boxed{V_c = V_{cc}\left[1 + m \cos w_m t\right]}$$

& modulated o/p vol. $V_o$ is given by

$$\boxed{V_o = V_{cc}(1 + m \cos w_m t) \cos w_c t}$$

where $m = \frac{V_m}{V_{cc}}$ = modulation index

⇒ Evolution & Description of SSB

Basic transmission processes:-

(I) The physical length of antenna must equal the wavelength of transmitted signal, usually in RF range.

(II) The audio signal is much too long to be transmitted directly by a conventional antenna.

(III) The intelligence (audio) must be processed by the electronic circuitry to meet transmission requirement of the system. This process is known as mixing.

The SSB system incorporates the mixing process plus a signal multiplying, filtering enhancement, to ensure that the signal meets the requirements of the antenna system.

If the audio (1 KHz) is mixed with basic carrier (1 MHz) the resultant freq. will be

1.001 MHz ; if the freq. is multiplied 1000 times, the resultant freq. will be 1001 MHz. If we filter out the carrier (100 MHz), a 1 MHz freq. (resultant audio) transmittable by the antenna system is left. To maintain audio quality in spite of inherent interference due to external sources, a reference carrier freq. is reinserted at the receiver & a new filtering process divides the freq. down to audio range once again. These mixing, filtering processes take place at the lower power levels of transmitter, allowing only the newly processed information signal to be transmitted at high power levels, thus ensuring efficiency and high fidelity.

We know that when a carrier is Amplitude modulated by a single sine wave, the resulting signal consists of three freq. The original carrier freq. ($f_c$), USB freq. ($f_c + f_m$) & LSB freq ($f_c - f_m$). Steps can be taken either during or after the modulation process, to remove or attenuate any combination of the components of AM wave.

The carrier of "standard" or DSB FC AM (also known as A3E modulation) conveys no information. This is bcz the carrier component remains const. in amplitude & freq., no matter what the modulating vol. does. Just as the fact that the two sidebands are images to each other, since each is affected by changes in modulating vol. amplitude via the exponent $mV_c/2$. All the information can be conveyed by use of one sideband. The carrier is superflous & the other sideband is redundant.

The main reason for the widespread use of A3E is relative simplicity of modulating & demodulating equipment. A3E is acceptable form used for broadcasting.

The AM power eqn states that the ratio of total power to carrier power is given by $(1+m^2/2):1$. If the carrier is suppressed, only sideband power remains. As this is only $P_c(m^2/2)$, a two-thirds (2/3) saving is effected at 100% modulation & even more is saved as the depth of modulation is reduced. If one of sideband is now also suppressed, the remaining power is $P_c(m^2/4)$, a further saving of 50% over carrier suppressed AM.

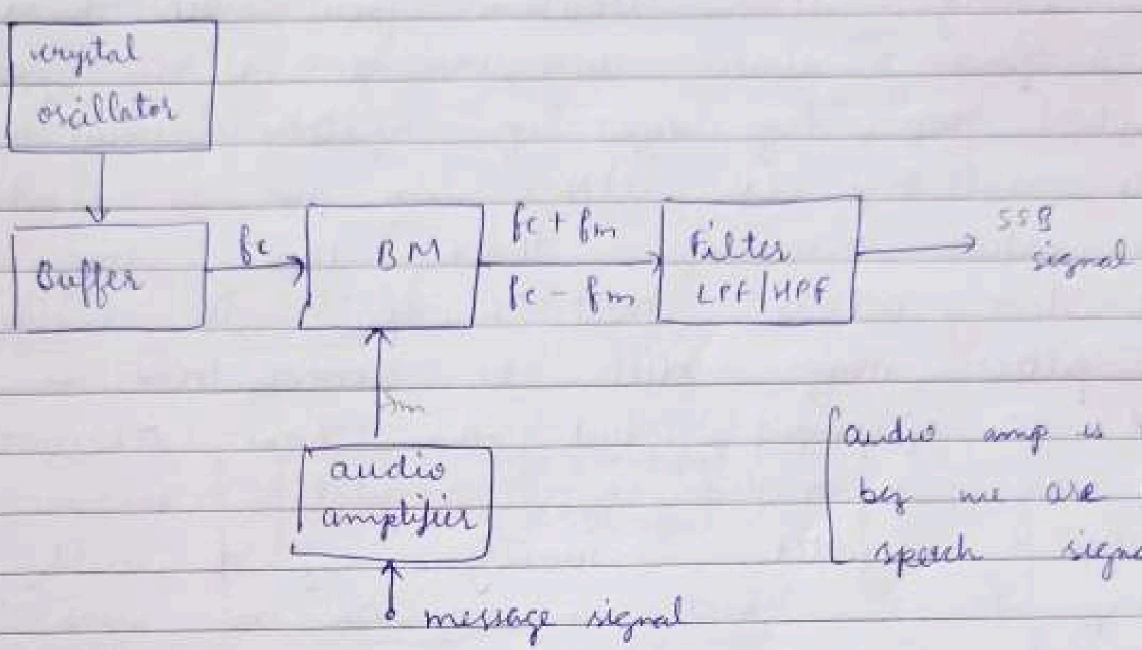
Also use of SSB immediately halves the bandwidth required for transmission as compared with A3E.

In practice, SSB is used to save power in app. where such a power saving is warranted i.e. in mobile systems.

→ Generation of SSB

Three main systems are employed for generation of SSB :- Filter method, Phase cancellation method & Third method. They differ from one another in the way of suppressing the unwanted sideband, but all use some form of AM to suppress carrier. The AM is the key ckt in SSB generation.

## (1) Filter Method




[audio amp is used bcz we are using speech signals]

The filter system is the simplest system of the three after BM the unwanted sideband is removed (actually heavily attenuated) by a filter. The filter may be LC, crystal, ceramic or mechanical, depending upon the carrier freq. The key circuits in this transmitter (employing) are BM & the sideband suppression filter. Such a filter must have a flat bandpass & extremely high attenuation outside the bandpass. In radio communication systems, the freq. range used for voice is 300 to about 2800 Hz. If it is required to suppress the LSB & if transmitting freq. is f, then lowest freq. that this filter must pass without attenuation is f + 300 Hz whereas highest freq. that must be fully attenuated is f − 300 Hz.

In other words, filter's response must change from zero attenuation to full attenuation over a range of only 600 Hz. If transmitting freq. is much above 10 MHz, this is virtually impossible.

Looking at situation from other side/end, we find that there must be an upper freq. limit for any type of filter circuit used. The multistage LC filters can not be used for RF values > 100 kHz. Above this freq. the attenuation outside the bandpass is insufficient. LC filters may still be encountered in currently used HF equipment, but they have otherwise tended to be superseded by crystal, ceramic or mechanical filter, mainly bcz of bulky size of components & great improvement in mechanical filters; Mechanical filters have been used at freq. upto 500 kHz & crystal or ceramic filters upto 20 MHz.

The mechanical filters seems to be the one with best all-around prop. :- small size, good bandpass, very good attenuation char. & an adequate upper freq. limit are its advantage. Crystal & ceramic may be cheaper but are preferable only at freq. above 1 MHz.

All these filters have same disadvantage that their max. operating freq. is below the usual transmitting frequencies.

Disadvantage of Filter Method

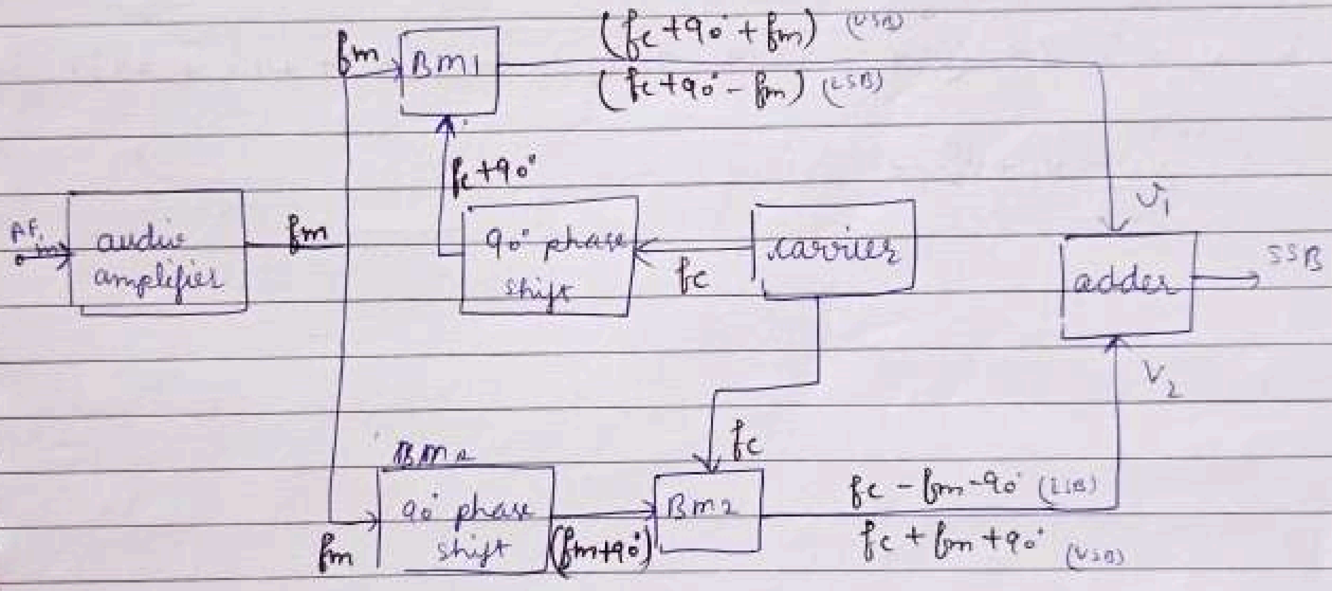
i) The cost of SSB receiver is higher than DSB counterpart by a ratio of about 3:1

ii) The average radio user wants only to flip a power switch & dial a ~~station~~ station. SSB receivers require several precise freq.

control settings to minimize distortion & may require continual readjustment during the use of system.

As the freq response curve of filters is not sharp ∴ it is not good for small freq.

(II) Phase Cancellation Method




The method avoids filters & used two Bm & two phase-shifting networks. BM1 receives the carrier vol. (shifted by 90°) & modulating vol. whereas Bm2 is fed the modulating vol. (shifted through 90°) & carrier vol. Sometimes the modulating vol. phase shift is arranged slightly differently. It is made +45° for one Bm & −45° for other, but the result is the same.

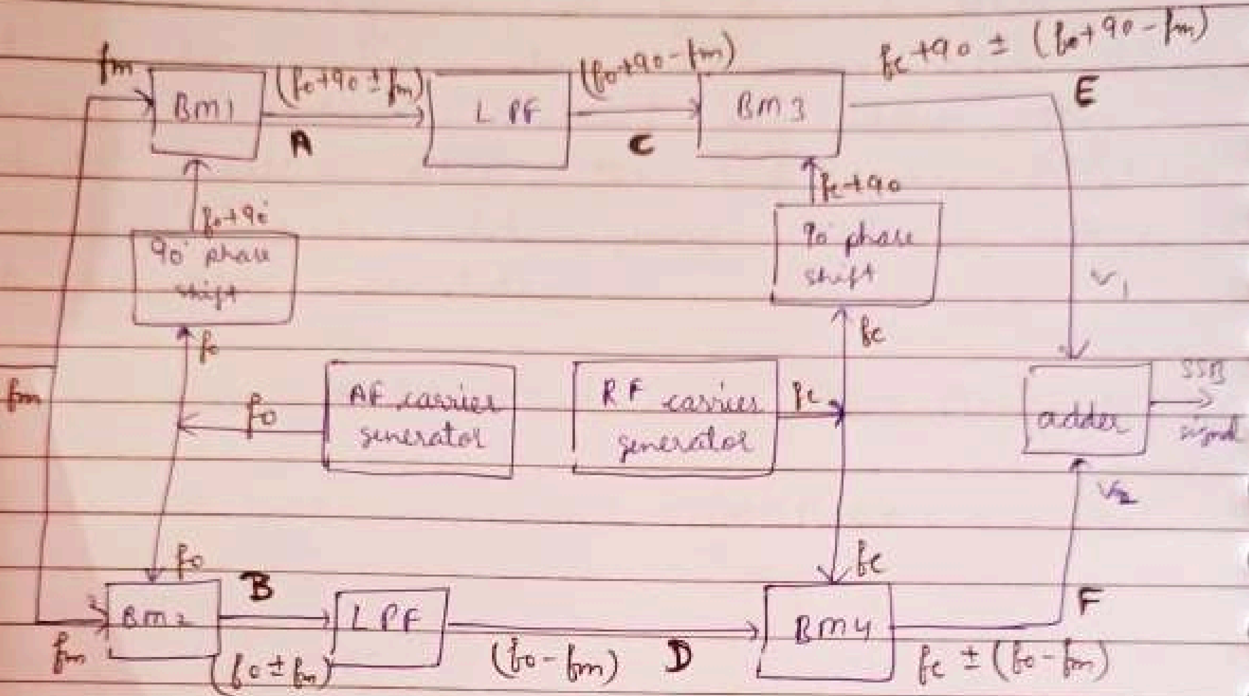
Both modulators produce an o/p consisting only of sidebands. It will be shown that both USB lead the input carrier vol. by 90°. One of LSB leads the reference vol. by 90° & other lags it by 90°. The two LSB are thus out of phase & when combined in adder, they cancel each other. The USB are in phase at adder & ∴ add, giving SSB in which LSB has been canceled.

$$V_1 = \underbrace{\cos(\omega_c t + 90 - \omega_m t)}_{LSB} - \underbrace{\cos(\omega_c t + 90 + \omega_m t)}_{USB}$$

& $$V_2 = \cos(\omega_c t - \omega_m t - 90) - \cos(\omega_c t + \omega_m t + 90)$$

$$\therefore \; V_0 = V_1 + V_2 =$$

## (III) The "Third" Method (used to rse dis. b/w 2 side bands)




The third method of generating SSB was developed by Weaver as a means of retaining the advantages of phase-cancellation method, such as its ability to generate SSB at any freq. & use low audio freq., without the associated disadvantage of an AF phase-shift network required to operate over a large range of audio freq. The third method is in direct competition with filter method, but is very complex & not often used commercially. From the dia., we see that the latter part of ckt is identical to that of phase cancellation method, but the way in which appropriate vol. are fed to the last two Bm (Bm3 & Bm4) has been changed. Instead of trying to phase-shift the whole range of audio freq., this method combines them with an AF carrier $f_o$ viz a fixed freq. in middle of the audio band 1650Hz. A phase shift is then applied to this freq. only & after the resulting vol. have been applied to first pair of Bm (Bm1 & Bm2), the low pass filters whose cut-off freq is $f_o$ ensure that I/p to last pair of Bm results in proper eventual sideband suppression.

at pt. F frequencies :- $f_c + 90 + f_o + 90 - f_m$

$$= \boxed{f_c + f_o - f_m + 180^\circ} \quad USB$$

& $f_c + 90 - f_o - 90 + f_m$

$$= \boxed{f_c - f_o + f_m} \quad LSB$$

at F $f_c \pm (f_o - f_m)$

$(f_c + f_o - f_m)$ USB & $(f_c - f_o + f_m)$ LSB

$$\therefore V_1 = \cos(\overset{LSB}{\omega_c t - \omega_o t + \omega_m t}) - \cos(\overset{USB}{\omega_c t + \omega_o t - \omega_m t + 180})$$

$$V_2 = \cos(\overset{LSB}{\omega_c t - \omega_o t + \omega_m t}) - \cos(\overset{USB}{\omega_c t + \omega_o t - \omega_m t})$$

$$V_o = V_1 + V_2 = \cos(\omega_c t - \omega_o t + \omega_m t) - \cos(180 + (\omega_c t + \omega_o t - \omega_m t))$$

$$+ \cos(\omega_c t - \omega_o t + \omega_m t) - \cos(\omega_c t + \omega_o t - \omega_m t)$$

$$V_o = 2\cos(\omega_c - \omega_o + \omega_m)t + \cos(\omega_c + \omega_o - \omega_m)t$$
$$- \cos(\omega_c + \omega_o - \omega_m)t$$

$$\boxed{V_o = 2\cos(\omega_c - \omega_o + \omega_m)t} \quad \underline{LSB}$$

$$V_0 = V_1 - V_2 = \cos(\omega_c - \cancel{\omega_0} + \omega_m)t + \cos(\omega_c + \omega_0 - \omega_m)t$$

$$- \cos(\omega_c \cancel{-} \omega_0 + \omega_m)t + \cos(\omega_c + \omega_0 - \omega_m)t$$

$$\boxed{V_0 = 2\cos(\omega_c + \omega_0 - \omega_m)t}$$ USB

**Number system :-** A number system is a code that uses symbols to count the number of items. The most common & familiar no. system is the decimal number system. The decimal no. system uses the symbols 0, 1, 2, 3, 4, 5, 6, 7, 8 and 9. Thus, the decimal system uses 10 digits for counting the items.

⇒ **Binary number system**

The binary number system is a system that uses only the digits 0 & 1 as codes. The binary no. system is a positional weighted system. The base or radix of this number system is 2. Hence, it has two independent symbols. The base itself can not be a symbol.

A binary digit is called a bit. A binary no. consists of sequence of bits, each of which is either a 0 or a 1. The binary point separates the integer and fraction part.

(*) 0, 1 → bits ; $r$ = radix or base = 2
i.e. value of $r$ = no. of symbols

Counting in binary no. system is performed much the same way as in the decimal no. system.

| Decimal | Binary |
| --- | --- |
| 0 | 0 |
| 1 | 1 |

| Decimal | Binary | Decimal | Binary |
|---|---|---|---|
| 2 | 10 | 11 | 1011 |
| 3 | 11 | 12 | 1100 |
| 4 | 100 | 13 | 1101 |
| 5 | 101 | 14 | 1110 |
| 6 | 110 | 15 | 1111 |
| 7 | 111 | 16 | 10000 |
| 8 | 1000 | 17 | 10001 |
| 9 | 1001 | 18 | 10010 |
| 10 | 1010 | 19 | 10011 |

Note that 0 and 1 count in the binary system is same as in the decimal counting. To represent 2, we use the second binary digit (i.e. 1) followed by the first (i.e. 0). This gives binary no. (10) as an equivalent of 2 in the decimal system. Likewise 3 in the decimal system can be represented by the binary no. (11). After this two binary digits have exhausted. We shall use the three digits to represent the next binary no. Thus to represent 4 (four) we use the second binary digit followed by two first binary digits. This gives the binary (100) as equivalent of 4 in the decimal system. This is the simple way to find binary equivalent.

* A string of <u>four bits</u> is called as a nibble and <u>eight bits</u> makes a byte.

Thus 1001 is a nibble & 10010110 is binary byte.

Highest decimal no. with n-Bit is

$\boxed{2^n - 1}$



\# The decimal value of the binary no. is the sum of the product of all its bits multiplied by the weights of their respective positions. In general, a binary no. with an integer part of (n+1) bits and a fraction part of K bits can be written as

$$\boxed{d_n\, d_{n-1}\, d_{n-2} \ldots d_1 d_0 \,.\, d_{-1}\, d_{-2}\, d_{-3} \ldots d_{-K}}$$

it's decimal equivalent is

$$(d_n \times 2^n) + (d_{n-1} \times 2^{n-1}) + \cdots + (d_1 \times 2^1) + (d_0 \times 2^0) +$$
$$(d_{-1} \times 2^{-1}) + (d_{-2} \times 2^{-2}) + \ldots (d_{-K} \times 2^{-K})$$

e.g:- we know that binary no. 1001 is equal to the decimal no. 9. This can be shown as

$$1001 = 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 9$$

For binary no., the digits at the extreme right is referred ~~to~~ as <u>least significant bit (LSB)</u> e.g:- In the binary no. 100<u>1</u>, the 1 at the right is the LSB.

The <u>left</u> most digit is called the <u>Most significant bit (MSB)</u> e.g:- In the binary no. <u>1</u>001, the 1 at left is MSB with value of 8 in decimal terms.

i.e. MSB $\leftarrow d_3\, d_2\, d_1\, d_0 \rightarrow$ LSB

in general <u>weightage</u>

$$\begin{array}{cccc} d_3 & d_2 & d_1 & d_0 \\ \downarrow & \downarrow & \downarrow & \downarrow \\ r^3 & r^2 & r^1 & r^0 \end{array}$$

here $r = 2$

$r$ = radix

The binary no. system used in digital computers because the switching circuits used in these computers use two state device such as transistor, diode etc. A transistor can be OFF or ON, a switch can be OPEN or CLOSED, a diode can be OFF or ON etc. These devices have to exist in one of the two possible states. So, these two states can be represented by the symbol 0 & 1 respectively.

→ **Binary to Decimal Conversions**

Binary no. can be converted to equivalent decimal no. quite easily. Suppose you are given the binary no. 110011. It's conversion to equivalent decimal no. involves the following two steps:-

(i) Place the decimal values of each position of the binary numbers 1 1 0 0 1 1
(ii) Add all the decimal values to get the decimal no. × (multiplied by) $2^5$ $2^4$ $2^3$ $2^2$ $2^1$ $2^0$ resp.

Thus $(110011)_2 = 1\times2^5 + 1\times2^4 + 0\times2^3 + 0\times2^2 + 1\times2^1 + 1\times2^0$

$= 32 + 16 + 0 + 0 + 2 + 1 = 51$

$\therefore (110011)_2 = (51)_{10}$ Ans

* In binary to decimal conversions all positions containing 0 can be ignored. Only add the decimal values of the positions where

1 appears. Thus in the case of binary no. $(110011)_2$

i.e:- $(110011)_2 = 1\times2^5 + 1\times2^4 + 1\times2^1 + 1\times2^0$
$= 32 + 16 + 2+1 = 51$

examples

① Convert the Binary no. (110001) to its equivalent decimal no.

$(110001)_2 = (?)_{10}$
$= 1\times2^5 + 1\times2^4 + 0\times2^3 + 0\times2^2 + 0\times2^1 + 1\times2^0$
$= 32 + 16 + 1 = 49$

$\boxed{(110001)_2 = (49)_{10}}$ Ans

② $(10101)_2 = (?)_{10}$

$(10101)_2 = 1\times2^4 + 0\times2^3 + 1\times2^2 + 0\times2^1 + 1\times2^0$
$= 16 + 0 + 4 + 0 + 1 = 21$

$\therefore \boxed{(10101)_2 = (21)_{10}}$ Ans

③ $(1001011)_2 = (?)_{10}$

$(1001011)_2 = 1\times2^6 + 0\times2^5 + 0\times2^4 + 1\times2^3 + 0\times2^2 + 1\times2^1 + 1\times2^0$
$= 64 + 0 + 0 + 8 + 0 + 2 + 1$
$= 75$

$\therefore \boxed{(1001011)_2 = (75)_{10}}$ Ans

(4) $(11011100)_2 = (?)_{10}$

$(11011100)_2 = 1\times2^7 + 1\times2^6 + 1\times2^4 + 1\times2^3 + 1\times2^2$
$= 128 + 64 + 16 + 8 + 4$
$= 220$

$\therefore$ $\boxed{(11011100)_2 = (220)_{10}}$ Ans

(5) $(101011)_2 = (?)_{10}$

$(101011)_2 = 1\times2^5 + 1\times2^3 + 1\times2^1 + 1\times2^0$
$= 32 + 8 + 2 + 1 = 43$

$\therefore$ $\boxed{(101011)_2 = (43)_{10}}$ Ans

⇒ **Binary Fraction to Decimal equivalent**

Binary fraction can also be converted to decimal equivalents. In this case, the weights of digit positions to the right of the point are given by negative power of 2. $\therefore$ The weights are $2^{-1}, 2^{-2}, 2^{-3}, \ldots$ etc

e.g.:- $(0.101) = 1\times2^{-1} + 0\times2^{-2} + 1\times2^{-3}$
$= 0.5 + 0 + 0.125 = 0.625$

$\therefore$ $(0.101)_2 = (0.625)_{10}$ Ans

Examples :-

(1) $(0.111)_2 = (?)_{10}$

$(0.111)_2 = 1\times2^{-1} + 1\times2^{-2} + 1\times2^{-3}$
$= 0.5 + 0.25 + 0.125 = 0.875$

$\therefore \boxed{(0.111)_2 = (0.875)_{10}}$ Ans

② $(0.10101)_2 = (?)_{10}$

$(0.10101)_2 = 1\times2^{-1} + 0\times2^{-2} + 1\times2^{-3} + 0\times2^{-4} + 1\times2^{-5}$
$= 0.5 + 0 + 0.125 + 0 + 0.03125$
$= 0.65625$

$\therefore \boxed{(0.10101)_2 = (0.65625)_{10}}$ Ans

③ $(0.1011)_2 = (?)_{10}$

$(0.1011)_2 = 1\times2^{-1} + 0\times2^{-2} + 1\times2^{-3} + 1\times2^{-4}$
$= 0.5 + 0 + 0.125 + 0.0625 = 0.6875$

$\therefore \boxed{(0.1011)_2 = (0.6875)_{10}}$ Ans

→ **Mixed Numbers :-**

For mixed numbers (no. with an integer & a fractional part), handle each part acc. to rule. The weights of mixed no. :-

$$\ldots \underbrace{2^3 \quad 2^2 \quad 2^1 \quad 2^0}_{\text{+ve power of 2}} \underset{\text{binary pt.}}{\bullet} \underbrace{2^{-1} \quad 2^{-2} \quad 2^{-3}}_{\text{-ve power of 2}} \ldots$$

Examples:-

① $(110.001)_2 = (?)_{10}$

$(110.001)_2 = 1\times2^2 + 1\times2^1 + 0\times2^0 + 0\times2^{-1} + 0\times2^{-2} + 1\times2^{-3}$
$= 4 + 2 + 0 + 0 + 0 + 0.125 = 6.125$

$\therefore \boxed{(110.001)_2 = (6.125)_{10}}$ Ans

② $(1110.1001)_2 = (?)_{10}$

$(1110.1001)_2 = 1\times2^3 + 1\times2^2 + 1\times2^1 + 0\times2^0 + 1\times2^{-1} + 0\times2^{-2} + 0\times2^{-3} + 1\times2^{-4}$
$= 8 + 4 + 2 + 0 + 0.5 + 0 + 0 + 0.0625$
$= 14.5625$

$\boxed{(1110.1001)_2 = (14.5625)_{10}}$ Ans

③ $(11001.1101)_2 = (?)_{10}$

$(11001.1101)_2 = 1\times2^4 + 1\times2^3 + 1\times2^0 + 1\times2^{-1} + 1\times2^{-2} + 1\times2^{-4}$
$= 16 + 8 + 1 + 0.5 + 0.25 + 0.625$
$= 25.8125$

$\therefore \boxed{(11001.1101)_2 = (25.8125)_{10}}$ Ans

④ $(110010.1011)_2 = (?)_{10}$

$(110010.1011)_2 = 1\times2^5 + 1\times2^4 + 1\times2^1 + 1\times2^{-1} + 1\times2^{-3} + 1\times2^{-4}$
$= 32 + 16 + 2 + 0.5 + 0.125 + 0.0625$
$= 50.6875$

$\therefore \boxed{(110010.1011)_2 = (50.6875)_{10}}$ Ans

## => Decimal to Binary Conversion

There are many methods to perform this conversion. The main method is called double dabble because it requires successive division by 2. This method can be summarised as:

Divide progressively the decimal no. by 2 & write down the remainder after each division continue this process till you get a quotient of 0 and remainder of 1, the conversion is now complete. The remainder taken in reverse order form the binary number

e.g:- Note that 13 is first divide by 2, giving a quotient of 6 with a remainder of 1. This remainder becomes the $2^0$ position in the binary no. The 6 is then divided by 2, giving a quotient of 3 with a remainder of 0. The remainder becomes $2^1$ position in binary no.

| | | | |
|---|---|---|---|
| i.e. - $13 \div 2 = 6$ | with a remainder | 1 | $2^0$ LSB |
| $6 \div 2 = 3$ | " " " | 0 | $2^1$ |
| $3 \div 2 = 1$ | " " " | 1 | $2^2$ |
| $1 \div 2 = 0$ | " " " | 1 | $2^3$ MSB |

continuing this procedure the equivalent binary no is 1101

OR

| | | |
|---|---|---|
| 2 | 13 | |
| 2 | 6 | 1 |
| 2 | 3 | 0 |
| | 1 | 1 |

$(13)_{10} = (1101)_2$ Ans

examples

① $(23)_{10} = (?)_2$

| | | |
|---|---|---|
| 2 | 23 | |
| 2 | 11 | 1 |
| 2 | 5 | 1 |
| 2 | 2 | 1 |
| | 1 | 0 |

$\boxed{(23)_{10} = (10111)_2}$ Ans

② $(52)_{10} = (?)_2$

| | | |
|---|---|---|
| 2 | 52 | |
| 2 | 26 | 0 |
| 2 | 13 | 0 |
| 2 | 6 | 1 |
| 2 | 3 | 0 |
| | 1 | 1 |

$\boxed{(52)_{10} = (110100)_2}$ Ans

③ $(283)_{10} = (?)_2$

| | | |
|---|---|---|
| 2 | 283 | |
| 2 | 141 | 1 |
| 2 | 70 | 1 |
| 2 | 35 | 0 |
| 2 | 17 | 1 |
| 2 | 8 | 1 |
| 2 | 4 | 0 |
| 2 | 2 | 0 |
| | 1 | 0 |

$\boxed{(283)_{10} = (100011011)_2}$ Ans

④ $(5280)_{10} = (?)_2$

| | | |
|---|---|---|
| 2 | 5280 | |
| 2 | 2640 | 0 |
| 2 | 1320 | 0 |
| 2 | 660 | 0 |
| 2 | 330 | 0 |
| 2 | 165 | 0 |
| 2 | 82 | 1 |
| 2 | 41 | 0 |
| 2 | 20 | 1 |
| | 10 | 0 |

| | | |
|---|---|---|
| 2 | 10 | 0 |
| 2 | 5 | 0 |
| 2 | 2 | 1 |
| | 1 | 0 |

$\boxed{(5280)_{10} = (1010010100000)_2}$ Ans

## ⇒ Decimal Fraction to Binary

As far as fractions are concerned, you multiply the no. by 2 & record a carry in the integer position. The carries read downward are the binary fractions.

e.g.:- $(0.85)_{10} = (?)_2$

| | | | | |
|---|---|---|---|---|
| $0.85 \times 2$ | = 1.7 | = 0.7 | with carry of | 1 |
| $0.7 \times 2$ | = 1.4 | = 0.4 | " " " | 1 |
| $0.4 \times 2$ | = 0.8 | = 0.8 | " " " | 0 |
| $0.8 \times 2$ | = 1.6 | = 0.6 | " " " | 1 |
| $0.6 \times 2$ | = 1.2 | = 0.2 | " " " | 1 |
| $0.2 \times 2$ | = 0.4 | = 0.4 | " " " | 0 |

Read downward

i.e. $(0.85)_{10} = (0.110110)_2$ Ans

Reading the carries downward gives binary fractions 0.110110. In this case, we stopped the conversion process after getting six binary digits. Because of this, the answer is an approximation. If more accuracy is needed continue multiplying by 2 until we get as many as digits necessary for our application.

**Example** ① $(0.8125)_{10} = (?)_2$

$$
\begin{array}{cr}
 & 0.8125 \\
 & \times 2 \\
\hline
1 & 1.6250 \\
 & \times 2 \\
\hline
1 & 1.2500 \\
 & \times 2 \\
\hline
0 & 0.5000 \\
 & \times 2 \\
\hline
1 & 1.0000
\end{array}
$$

$\therefore \boxed{(0.8125)_{10} = (0.1101)_2}$ Ans

## Mixed Numbers

① $(38.625)_{10} = (?)_2$

| | | |
|---|---|---|
| 2 | 38 | |
| 2 | 19 | 0 |
| 2 | 9 | 1 |
| 2 | 4 | 1 |
| 2 | 2 | 0 |
| | 1 | 0 |

$38 = (100110)_2$

$$
\begin{array}{ll}
 & 0.625 \\
 & \times 2 \\
1 & 1.250 \\
 & \times 2 \\
0 & 0.500 \\
 & \times 2 \\
1 & 1.000
\end{array}
$$

$(0.625)_{10} = (0.101)_2$

$\therefore \boxed{(38.625)_{10} = (100110.101)_2}$ Ans

② $(18.71875)_{10} = (?)_2$

| | | |
|---|---|---|
| 2 | 18 | |
| 2 | 9 | 0 |
| 2 | 4 | 1 |
| 2 | 2 | 0 |
| | 1 | 0 |

$(18)_{10} = (10010)_2$

$$
\begin{array}{ll}
 & 0.71875 \\
 & \times 2 \\
1 & 1.43750 \\
 & \times 2 \\
0 & 0.87500 \\
 & \times 2 \\
1 & 1.75000 \\
 & \times 2 \\
1 & 1.50000 \\
 & \times 2 \\
1 & 1.00000
\end{array}
$$

$(0.71875)_{10} = (0.10111)_2$

$\therefore \boxed{(18.71875)_{10} = (10010.10111)_2}$ Ans

⇒ **Binary Addition :-** Computer circuits don't process decimal numbers; they process binary no. Before we understand how a computer performs arithmetic we have to learn how to add binary no. Binary addition is the key to binary subtraction, multiplication & division. So, let's begin with the four most basic cases of binary addition

$0 + 0 = 0$ ⓐ $1 + 0 = 1$ ⓒ
$0 + 1 = 1$ ⓑ $1 + 1 = 10$ ⓓ

eqⁿ ⓐ, ⓑ, ⓒ are obvious because they are identical to decimal addition. The 4th case, however, is different. As we know 1+1 gives 2 in decimal & as 2 is equivalent to 10 in binary so (1+1) gives 10

so $1 + 1 = 10$ i.e. 0 with carry of 1

so, in general we could write

| | | Sum | Carry |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |

∴ addition in decimal :-

1 = one radix = 1 r = 10

0, 1, 2, 3, 4, 5, 6, 7, 8, 9

$\boxed{r = 10}$

$$\begin{array}{r} 25 \\ +\ 47 \\ \hline 72 \\ \hline \end{array}$$

Mul^ly in Binary :- (I)

$$\begin{array}{r} 1\ 0 \\ +\ 1\ 0 \\ \hline 1\ 0\ 0 \end{array}$$

1 = r

0, 1 → symbol

r = 2

0 + 0 = 0

1 + 1 = 2 - radix

2 - 2 = 0

(II) decimal

$$\begin{array}{r} 0\ 5 \\ 0\ 5 \\ 0\ 5 \\ +\ 0\ 5 \\ \hline 2\ 0 \end{array}$$

$\boxed{r = 10}$

20 - r = 20 - 10 = 10

10 - r = 10 - 10 = 0

Mul^ly Binary

$$\begin{array}{r} 0\ 0\ 1 \\ 0\ 0\ 1 \\ 0\ 0\ 1 \\ +\ 0\ 0\ 1 \\ \hline 1\ 0\ 0 \end{array}$$

r = 2

1+1+1+1 = 4

4 - r = 4 - 2 = 2 (1st r)

2 - r = 2 - 2 = 0 (2nd r)

Now, 1 + 1 = 2

2 - r = 2 - 2 = 0 (1)

examples:-

(1)

$$\begin{array}{r} 0\ 0\ 1\ 1\ 1 \\ +\ 0\ 1\ 1\ 0\ 1 \\ \hline 1\ 0\ 1\ 0\ 0 \end{array}$$

Ans

1+1 = 2 - r = 2 - 2 = 0 (1)

1+1+1 = 3 - r = 3 - 2 = 1 (1)

(2)

$$\begin{array}{r} 1\ 1\ 1\ 1\ 1 \\ +\ 1\ 1\ 0\ 1 \\ \hline 1\ 0\ 1\ 1\ 0\ 0 \end{array}$$

Ans

3 - 2 = 1 (1)

(3)

$$\begin{array}{r} 1\ 1\ 0\ 1\ .\ 1\ 0\ 1\ 1 \\ +\ 1\ 1\ 0\ 0\ 1\ .\ 0\ 1\ 0\ 0 \\ \hline 1\ 0\ 0\ 1\ 1\ 0\ .\ 1\ 1\ 1\ 1 \end{array}$$

Ans

⇒ Binary Subtraction:-

The Binary subtraction is performed in a manner similar to that in decimal subtraction. The rules for binary subtraction are:

| | | Diff. | Borrow |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |

in decimal

$$\begin{array}{r} 3\ 4\ 7 \\ -\ \ 2\ 9 \\ \hline 1\ 8 \end{array}$$

(1) → 10+7

∵ r = 10

borrow = one r = 10

i.e. in subtraction radix will borrow

in Binary

$$\begin{array}{r} 1\ 0 \\ -\ 0\ 1 \\ \hline 0\ 1 \end{array}$$

(1) → 2+0

borrow = one r = 2 ∵ r = 2

Ans

examples:-

(1)

$$\begin{array}{r} 1\ 1\ 1\ 0\ 0 \\ -\ 1\ 1 \\ \hline 1\ 1\ 0\ 0\ 1 \end{array}$$

1r = 2, 2nd r 2+0

Ans

1 0 → 2+0
2 0
1 1

(2)

$$\begin{array}{r} 1\ 0\ 0\ 0 \\ -\ 1 \\ \hline 0\ 1\ 1\ 1 \end{array}$$

1r = 2

Ans

(4)

$$\begin{array}{r} 1\ 0\ 1\ 0\ .\ 0\ 1\ 0 \\ -\ 1\ 1\ 1\ .\ 1\ 1\ 1 \\ \hline 0\ 0\ 1\ 0\ .\ 0\ 1\ 1 \end{array}$$

Ans

(3)

$$\begin{array}{r} 1\ 0\ 0\ 0 \\ -\ 1\ 0 \\ \hline 0\ 1\ 1\ 0 \end{array}$$

Ans

## Binary Multiplication

The method used for binary is the paper method.

Muliplication rules :- $0 \times 0 = 0$ ; $1 \times 1 = 1$
$1 \times 0 = 0$ ; $0 \times 1 = 0$

The paper method is similar to the multiplicat-ion of decimal no. on paper.
Multiply the multiplicand with each bit of the multiplier & add the partial products. The partial product is the same as the multiplicand of the multiplier bit is 1 & is zero if the multiplier bit is 0.

In decimal

```
   1
   2 3
 x 2 4
 -----
 1 9 2
 4 6 x
 -----
 5 5 2
```

in Binary

```
      1 1
    x 1 1
    -----
0 0   1 1
    1 1 x
  -------
  1 0 0 1  Ans
```

example:-

① multiply 1101 by 110

```
          1 1 0 1
          x 1 1 0
      -----------
          0 0 0 0
0 0   1 1 0 1 x
    1 1 0 1 x x
  ---------------
  1 0 0 1 1 1 0  Ans
```

The LSB of multiplier is 0. So, the first partial product is 0. The next two bits of multiplier are 1s. So, the next two partial products are equals to the multiplicand itself.

The sum of partial products gives the answer.

(2)

```
      1 1 1
    x 1 0 1
    -------
      1 1 1
 0 0 1 0 0 x
   1 1 1 x x
  -----------
 1 0 0 0 1 1   Ans
```

(3)

```
          1 0 0 1
        x 1 1 1 1
        ---------
          1 0 0 1
      0 1 0 0 1 x
    0 1 0 0 1 x x
  0 1 0 0 1 x x x
  ---------------
  1 0 0 0 0 1 1 1   Ans
```

(4)

```
          1 1 . 0 0 1
        x 1 0 . 0 1 0
        -------------
        1 0 0 1 0 0 0
      1 1 1 0 0 1 x
      0 0 0 0 0 x x
    0 0 0 0 0 x x x
  1 1 0 0 1 x x x x
  -------------------
  1 1 1 . 0 0 0 0 0 1 0   Ans
```

(5)

```
              1 1 0 0 . 0 0 1
            x 1 1 0 . 1 0 0
        -----------------------
              0 0 0 0 0 0 0
            0 0 0 0 0 0 0 x
          1 1 0 0 0 0 1 x x
        0 0 0 0 0 0 0 x x x
      0 1 1 0 0 0 0 1 x x x x
      1 1 0 0 0 0 1 x x x x x
  ---------------------------------
  1 0 0 1 1 1 0 . 1 1 0 1 0 0   Ans
```

⇒ Binary Division :-

Like multiplication the paper method is used for division as well. In the paper method long division procedure similar to those in decimal are used.

0 ÷ 0 = meaningless

0 ÷ 1 = 0

1 ÷ 0 = meaningless

1 ÷ 1 = 1

(1)

```
100 ) 1 0 1 1 0 0 ( 1011 Ans
      1 0 0
      -----
      0 0 1 1 0
          1 0 0
          -----
          0 1 0 0
            1 0 0
            -----
              x
```

② 1100 / 10

10 ) 1100 ( 110 Ans
10
010
10
000

③ 11000 / 110

110 ) 11000 ( 100 Ans
110
00000

④ 1110011 / 101

101 ) 1110011 ( 10111 Ans
101
1000
101
00111
101
101
101
×

⑤ 1100010 / 111

111 ) 1100010 ( 1110 Ans
111
01010
111
0111
111
00

⑥ 110101.11 / 101

101 ) 110101.11 ( 1010.11 Ans
101
00110
101
00111
101
101
101
×

## Octal Number System [r = 8]

The octal number system was extensively used by early minicomputers. It is also a positional weighted system. It's base or radix is 8. It has 8 independent symbols 0, 1, 2, 3, 4, 5, 6, 7. Since, it's base is $8 = 2^3$, every 3-bit group of binary can be represented by an octal digit. An octal no. is thus 1/3 rd the length of the corresponding binary no.

## Usefulness of the octal system :-

The ease with which conversion can be made b/w octal & binary makes the octal system more attractive as a shorthand means of expressing large binary numbers. In computers work, binary no. with upto 64 bits are not uncommon. These binary no. do not always represent a numerical quantity, they often represent some types of code which convey non-numerical information. In computers, binary numbers might represent (i) the actual numerical data

(ii) the no. corresponding to a location (address) in memory

(iii) An instruction code

(iv) A code expressing alphabetic and other non-numerical characters

(v) A group of bits representing the status of device internal or external to the computer.

When dealing with large binary no. of many digits, it is convenient & more efficient

⊛ decimal से किसी भी no. system में convert करना है तो उस decimal no. को जिस system में जाना है उसके radix से divide करो & Note remainder.

for us to write the numbers in octal rather than binary. However, the digital circuits & system work strictly in binary; we use octal only for the convenience of the operators of the system.

## Decimal to Octal conversion

To convert a decimal no. to octal, we employ the same repeated division method that we used in decimal to binary conversion. To convert given decimal integer no. ~~to~~ octal successively divide the given no. 8 (radix of octal no. system) till the quotient is 0. The last remainder is MSD. The remainders read upwards gives the equivalent octal integer no.

eg. To convert $(91)_{10}$ to octal no.

procedure is:

| Division | | Remainder | |
|---|---|---|---|
| $91 \div 8$ | $= 11$ | 3 | (LSD) |
| $11 \div 8$ | $= 1$ | 3 | ↑ |
| $1 \div 8$ | $= 0$ | 1 | (MSD) |

$$(91)_{10} = (133)_8$$

OR

$$\begin{array}{r|l l} 8 & 91 & \\ \hline 8 & 11 & 3 \\ \hline & 1 & 3 \end{array}$$

$(91)_{10} = (133)_8$ Ans

examples :-

① $(759)_{10} = (?)_8$

| | | |
|---|---|---|
| 8 | 759 | |
| 8 | 94 | 7 |
| 8 | 11 | 6 |
| | 1 | 3 |

$\boxed{(759)_{10} = (1367)_8}$ Ans

② $(1598)_{10} = (?)_8$

| | | |
|---|---|---|
| 8 | 1598 | |
| 8 | 199 | 6 |
| 8 | 24 | 7 |
| | 3 | 0 |

$\boxed{(1598)_{10} = (3076)_8}$ Ans

③ $(266)_{10} = (?)_8$

| | | |
|---|---|---|
| 8 | 266 | |
| 8 | 33 | 2 |
| | 4 | 1 |

$\boxed{(266)_{10} = (412)_8}$ Ans

→ **Decimal Fraction to Octal Fraction :-**

To convert the given decimal fraction to octal successively multiply the decimal fraction by 8 till the product is 0 or till the required accuracy is obtained. The first integer from top is MSD. The integer to the left of the octal pt read downward gives the octal fraction

examples :-

① $(0.125)_{10} = (?)_8$

$$\begin{array}{r} 0.125 \\ \times 8 \\ \hline 1.000 \\ \times 8 \\ \hline 000 \end{array}$$

$\boxed{(0.125)_{10} = (0.1)_8}$ Ans

② $(0.175)_{10} = (?)_8$

$$\begin{array}{r|l} & 0.175 \\ & \times 8 \\ \hline 1 & \underline{1}.400 \\ & \times 8 \\ \hline 3 & \underline{3}.200 \\ & \times 8 \\ \hline 1 & \underline{1}.600 \\ & \times 8 \\ \hline 4 & \underline{4}.800 \\ & \times 8 \\ \hline 6 & \underline{6}.400 \\ & \times 8 \\ \hline 3 & \underline{3}.200 \end{array}$$

$\boxed{(0.175)_{10} = (0.131463\ldots)_8}$ Ans

③ $(0.23)_{10} = (?)_8$

$$\begin{array}{r|l} & 0.23 \\ & \times 8 \\ \hline 1 & \underline{1}.84 \\ & \times 8 \\ \hline 6 & \underline{6}.72 \\ & \times 8 \\ \hline 5 & \underline{5}.76 \\ & \times 8 \\ \hline 6 & \underline{6}.08 \\ & \times 8 \\ \hline 0 & \underline{0}.64 \end{array}$$

$\boxed{(0.23)_{10} = (0.16560)_8}$ Ans

* Mixed Numbers :-

① $(22.21875)_{10} = (?)_8$

$$\begin{array}{r|l} 8 & 22 \\ \hline & 2 \quad 6 \end{array}$$

$(22)_{10} = (26)_8$

$$\begin{array}{r|l} & 0.21875 \\ & \times 8 \\ \hline 1 & \underline{1}.75000 \\ & \times 8 \\ \hline 6 & \underline{6}.00000 \end{array}$$

$(0.21875)_{10} = (0.16)_8$

$\therefore \boxed{(22.21875)_{10} = (26.16)_8}$ Ans

② $(79.4375)_{10} = (?)_8$

$$\begin{array}{r|l} 8 & 79 \\ \hline 8 & 9 \quad 7 \\ \hline & 1 \quad 1 \end{array}$$

$(79)_{10} = (117)_8$

$$\begin{array}{r|l} & 0.4375 \\ & \times 8 \\ \hline 3 & \underline{3}.5000 \\ & \times 8 \\ \hline 4 & \underline{4}.0 \end{array}$$

$(0.4375)_{10} = (0.34)_8$

$\therefore \boxed{(79.4375)_{10} = (117.34)_8}$ Ans

⇒ **Octal to Decimal conversion :-**

To convert an octal no. to decimal no. multiply each digit in the octal no. by the weight of its position & add all the product terms.

i.e. decimal value of octal no. $\boxed{d_n d_{n-1} d_{n-2} \cdots d_1 d_0 . d_{-1} d_{-2}}$

is : $(d_n \times 8^n) + (d_{n-1} \times 8^{n-1}) + \cdots + (d_1 \times 8^1) + (d_0 \times 8^0) + (d_{-1} \times 8^{-1}) + (d_{-2} \times 8^{-2}) + \cdots$

eg:- ① $(47)_8 = (?)_{10}$

$47 = 4 \times 8^1 + 7 \times 8^0 = 32 + 7 = 39$

$\therefore \boxed{(47)_8 = (39)_{10}}$ Ans

② $(564)_8 = (?)_{10}$

$(564)_8 = 5 \times 8^2 + 6 \times 8^1 + 4 \times 8^0 = 320 + 48 + 4 = 372$

$\boxed{(564)_8 = (372)_{10}}$ Ans

③ $(372)_8 = (?)_{10}$

$(372)_8 = 3 \times 8^2 + 7 \times 8^1 + 2 \times 8^0 = 192 + 56 + 2$

$= 250$

$\boxed{(372)_8 = (250)_{10}}$ Ans

④ $(0.34)_8 = (?)_{10}$

$(0.34)_8 = 3 \times 8^{-1} + 4 \times 8^{-2} = 3 \times 0.125 + 4 \times 0.015625$

$= 3.125 + 0.0625 = 3.1875$

$\boxed{(0.34)_8 = (3.1875)_{10}}$ Ans

Date : / /

⑤ $(0.542)_8 = (?)_{10}$

$(0.542)_8 = 5\times8^{-1} + 4\times8^{-2} + 2\times8^{-3}$

$= 5\times0.125 + 4\times0.015625 + 2\times0.001953$

$= \cancel{5.125} + \cancel{4 \times 0.015625} \ 0.625 + 0.0625 + 0.003906$

$= 0.691406$

$\therefore \boxed{(0.542)_8 = (0.691406)_{10}}$ Ans

⑥ $(4057.06)_8 = (?)_{10}$

$(4057.06)_8 = 4\times8^3 + 5\times8^1 + 7\times8^0 + 0\times8^{-1} + 6\times8^{-2}$

$= 4\times512 + 40 + 7 + 6\times0.015625$

$= 2048 + 47 + 0.09375 = 2095.09375$

$\therefore \boxed{(4057.06)_8 = (2095.09375)_{10}}$ Ans

→ Octal to Binary conversion :-

The advantage of octal no. system is the ease with which an octal no. can be converted to a binary no. & vice versa. It is because eight is the third power of Two, providing a direct correlation b/w three bit groups in a binary no. & the octal digit i.e.- each three bit group of binary bits can be represented by one octal digits. Therefore, conversion from octal to binary is performed by converting each octal digit to it's 3-bit binary equivalent

① $(472)_8 = (?)_2$

$4 \rightarrow 100$, $7 \rightarrow 111$, $2 \rightarrow 010$

$\therefore \boxed{(472)_8 = (100111010)_2}$ Ans

② $(5431)_8 = (?)_2$

5 4 3 1
101 100 011 001

$\therefore \boxed{(5431)_8 = (101100011001)_2}$ Ans

③ $(3574)_8 = (?)_2$

3 5 7 4
011 101 111 100

$\therefore \boxed{(3574)_8 = (011101111100)_2}$ Ans

④ $(35.216)_8 = (?)_2$

3 5 2 1 6
011 101 010 001 110

$\therefore \boxed{(35.216)_8 = (011101.010001110)_2}$ Ans

✱ we can also do that

octal no. → decimal no. → binary no.
$(\ )_8 \rightarrow (\ )_{10} \rightarrow (\ )_2$

e.g.:- ① $(24)_8$

(i) $(24)_8 = (?)_{10}$

$= 2 \times 8^1 + 4 \times 8^0 = 16 + 4 = 20$

$\therefore (24)_8 = (20)_{10}$

Now, (ii) $(20)_{10} = (?)_2$

| | | |
|---|---|---|
| 2 | 20 | |
| 2 | 10 | 0 |
| 2 | 5 | 0 |
| 2 | 2 | 1 |
| | 1 | 0 |

$(20)_{10} = (10100)_2$

i.e. $\boxed{(24)_8 = (10100)_2}$ Ans

## Binary To octal conversion :-

The conversion of binary no. to the octal no. is simply the reverse of above process. The bits of the binary no. are grouped into groups of three bits starting at the LSB. Then each group is converted into to its octal equivalent.

e.g.:- ① $(100111010)_2 = (?)_8$

$\underbrace{100}_{4}\ \underbrace{111}_{7}\ \underbrace{010}_{2}$ $\therefore \boxed{(100111010)_2 = (472)_8}$ Ans.

**Note** :- Sometimes the binary no. will not have even groups of 3 bits. In that case, we can add one or two zero's to the left of MSB of the binary no. to fill the last group.

eg:- ① $(11010110)_2 = (?)_8$

$\underbrace{011}_{3}\ \underbrace{010}_{2}\ \underbrace{110}_{6}$ $\therefore \boxed{(11010110)_2 = (326)_8}$ Ans

Note that a 0 is placed to the left of the MSB to produce even gps of 3 bits.

OR Binary → Decimal → Octal
$(\ )_2 \rightarrow (\ )_{10} \rightarrow (\ )_8$

eg: ① $(110110001010)_2 = (?)_8$

(i) $(110110001010)_2 = (?)_{10}$

$(110110001010)_2 =$

$= 1\times2^{11} + 1\times2^{10} + 0\times2^{9} + 1\times2^{8} + 1\times2^{7} + 0\times2^{6} + 0\times2^{5} + 0\times2^{4} + 1\times2^{3} + 0\times2^{2} +$
$1\times2^{1} + 0\times2^{0}$

$= 2^{11} + 2^{10} + 2^{8} + 2^{7} + 2^{3} + 2^{1}$

$= 2048 + 1024 + 256 + 128 + 8 + 2 = (3466)_{10}$

i.e. $(110110001010)_2 = (3466)_{10}$

Now (ii) $(3466)_{10} = (?)_8$

| | | |
|---|---|---|
| 8 | 3466 | |
| 8 | 433 | 2 |
| 8 | 54 | 1 |
| | 6 | 6 |

$\therefore (3466)_{10} = (6612)_8$

$\therefore \boxed{(110110001010)_2 = (6612)_8}$ Ans

⇒ **Octal Addition :-**

The octal addition is performed by the decimal method. Add the digits in each column in decimal & convert this sum into octal. Record the octal sum term in the column & carry the carry term to the next column.

Another easy way of addition is done by using the given table.

| + | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 1 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 10 |
| 2 | 2 | 3 | 4 | 5 | 6 | 7 | 10 | 11 |
| 3 | 3 | 4 | 5 | 6 | 7 | 10 | 11 | 12 |
| 4 | 4 | 5 | 6 | 7 | 10 | 11 | 12 | 13 |
| 5 | 5 | 6 | 7 | 10 | 11 | 12 | 13 | 14 |
| 6 | 6 | 7 | 10 | 11 | 12 | 13 | 14 | 15 |
| 7 | 7 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |

Date : / /

[circle the upper no. of given column e.g. $\binom{4}{5}$] i.e. 4 in upper most row & then circle the lower no. i.e. 5 in first column of the table & then comes down in the column of circled no. (4) & move forward in the row of circled no. (5), the corresponding no. of this operation is 11. So 11 will be the sum of 4 & 5.]

OR e.g.

$$\begin{array}{r} 5\ 6\ 7 \\ +\ 2\ 3\ 4 \\ \hline 1\ 0\ 2\ 3 \end{array}$$ Ans

here r = 8

0, 1, 2, 3, 4, 5, 6, 7

$7+4 = 11 - r = 11 - 8 = 3$

$1+6+3 = 10 - r$
$= 10 - 8 = 2$

$1+5+2 = 8 - r$
$= 8 - 8 = 0$

①
$$\begin{array}{r} 1\ 6\ 2 \\ +\ 5\ 3\ 7 \\ \hline 7\ 2\ 1 \end{array}$$ Ans

$2+7 = 9 - r = 9 - 8 = 1$

$10 - r = 10 - 8 = 2$

②
$$\begin{array}{r} 1\ 3\ 6 \\ +\ 6\ 3\ 6 \\ \hline 7\ 7\ 4 \end{array}$$ Ans

$6+6 = 12 - r = 12 - 8 = 4$

③
$$\begin{array}{r} 25.27 \\ +\ 13.20 \\ \hline 40.47 \end{array}$$ Ans

$5+3 = 8 - r = 8 - 8 = 0$

④
$$\begin{array}{r} 67.5 \\ +\ 45.6 \\ \hline 135.3 \end{array}$$ Ans

$5+6 = 11 - r = 11 - 8 = 3$

$7+5+1 = 13 - r = 13 - 8 = 5$

$11 - r = 11 - 8 = 3$

(5)
$$\begin{array}{r} 327.54 \\ +\ 665.37 \\ \hline 1215.13 \end{array}$$ Ans

$4+7 = 11-8 = 3$

$9-8=1$ ; $13-8=5$ ; $9-8=1$

$10-8=2$

## Octal Subtraction :-

It is similar to decimal subtraction. In this case we take the borrow 8 ($\because r = 8$)

i.e. $(55)_8 - (27)_8$

$$\begin{array}{r} 55 \\ -\ 27 \\ \hline 26 \end{array}$$ Ans

$1 = 8+5 = 13$ & $13-7=6$

i.e. 1 radix borrow = 8

(1)
$$\begin{array}{r} 321 \\ -\ 047 \\ \hline 252 \end{array}$$ Ans

$9\ (8+1)$

(2)
$$\begin{array}{r} 20.14 \\ -\ 16.47 \\ \hline 01.45 \end{array}$$ Ans

$8+4=12$

⇒ **Hexadecimal Number System** ($r=16$)

Binary numbers are long. These numbers are fine for machines but are too lengthy to be handled by human beings. So, there is a need to represent the binary no. concisely. One no. system developed with this objective is the hexadecimal no. system (or Hex). Although it is somewhat more difficult to interpret than the octal no. system, it has become the most popular means of direct data entry & retrieval in digital systems.

The hexadecimal no. system is a positional weighted system. The base or radix of this no. system is 16. That means it has 16 independent symbols. The symbols are 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E & F. Since it's base is $16 = 2^4$ every 4 binary digit combination can be represented by one hexadecimal digit. So, a hexadecimal no. is 1/4th the length of the corresponding binary no., yet it provides the same information as the binary no. A 4-bit group is called a nibble. Since, comp. words come in 8-bits, 16 bits, 32 bits & so on. that is multiple of 4 bit, they can be easily represented by in hexadecimal. The hexadecimal system is particularly useful for human communications with computer. By far, this is the most commonly used no. system in computer literature. It is used in both large & small computers.

| Hexadecimal | Decimal | Binary |
|---|---|---|
| 0 | 0 | 0000 |
| 1 | 1 | 0001 |
| 2 | 2 | 0010 |
| 3 | 3 | 0011 |
| 4 | 4 | 0100 |
| 5 | 5 | 0101 |
| 6 | 6 | 0110 |
| 7 | 7 | 0111 |
| 8 | 8 | 1000 |
| 9 | 9 | 1001 |
| A | 10 | 1010 |
| B | 11 | 1011 |

| Hexadecimal | Decimal | Binary |
|---|---|---|
| C | 12 | 1100 |
| D | 13 | 1101 |
| E | 14 | 1110 |
| F | 15 | 1111 |

## Decimal to Hexadecimal conversion:-

To convert a decimal integer no. to hexadecimal successively divide the given decimal no. by 16 till the quotient is zero. The last remainder is the MSD. The remainders read from bottom to top gives the equivalent hexadecimal integer.

To convert a decimal fraction to hexadecimal successively multiply the given decimal fraction by 16, till the product is zero or till the required accuracy is obtained & collect all the integers to the left of decimal pt. The first integer is MSD & the integers read from top to bottom give the hexadecimal fraction. This is known as the hex dabble method.

eg:- ① $(28)_{10} = (?)_{16}$

$16 \,|\, 28$
$\quad 1 \rightarrow 12 = C$

$\boxed{(28)_{10} = (1C)_{16}}$ Ans

② $(47)_{10} = (?)_{16}$

$16 \,|\, 47$
$\quad 2 \rightarrow 15 = F$

$\boxed{(47)_{10} = (2F)_{16}}$ Ans

③ $(0.875)_{10} = (?)_{16}$

$$\begin{array}{r} 0.875 \\ \times 16 \\ \hline 14.000 \end{array}$$

$14 = E$

$\therefore \boxed{(0.875)_{10} = (0.E)_{16}}$ Ans

④ $(0.28125)_{10} = (?)_{16}$

$$\begin{array}{r} 0.28125 \\ \times 16 \\ \hline 4.50000 \\ \times 16 \\ \hline 8.0 \end{array}$$

4 ↓ 8

$\boxed{(0.28125)_{10} = (0.48)_{16}}$ Ans

⑤ $(47.375)_{10} = (?)_{16}$

$$\begin{array}{r|l} 16 & 47 \\ \hline & 2 \rightarrow F \end{array}$$

$(47)_{10} = (2F)_{16}$

$$\begin{array}{r} 0.375 \\ \times 16 \\ \hline 6.000 \end{array}$$

↓ 6

$(0.375)_{10} = (0.6)_{16}$

$\therefore \boxed{(47.375)_{10} = (2F.6)_{16}}$ Ans

## Hexadecimal to Decimal conversion

In order to convert a hex no. to its decimal equivalent, simply add up the position weight of each digit in the hex no.

If hex no is $\boxed{d_n d_{n-1} \cdots d_1 d_0 d_{-1} d_{-2} \cdots d_{-k}}$

then its decimal equivalent is

$$(d_n \times 16^n) + (d_{n-1} \times 16^{n-1}) + \cdots + (d_1 \times 16^1) + (d_0 \times 16^0) + (d_{-1} \times 16^{-1}) \cdots (d_{-k} \times 16^{-k})$$

eg:- ① $(9D)_{16} = (?)_{10}$

$9D = 9\times16^1 + D\times16^0 = 144 + 13 = 157$

$\therefore \boxed{(9D)_{16} = (157)_{10}}$ Ans

② $(356)_{16} = (?)_{10}$

$356 = 3\times16^2 + 5\times16^1 + 6\times16^0 = 3\times256 + 80 + 6$
$= 768 + 86 = 854$

$\boxed{(356)_{16} = (854)_{10}}$ Ans

③ $(F8E6)_{16} = (?)_{10}$

$(F8E6)_{16} = F\times16^3 + 8\times16^2 + E\times16^1 + 6\times16^0$
$= (15\times4096) + (8\times256) + (14\times16) + (6\times1)$
$= 61440 + 2048 + 224 + 6 = 63718$

$\therefore \boxed{(F8E6)_{16} = (63718)_{10}}$ Ans

④ $(0.48)_{16} = (?)_{10}$

$0.48 = 4\times16^{-1} + 8\times16^{-2} = 4\times0.0625 + 8\times0.00390625$
$= 0.25 + 0.03125 = 0.28125$

$\therefore \boxed{(0.48)_{16} = (0.28125)_{10}}$ Ans

**Hex to Binary conversion**

The conversion from hex to binary is performed by converting each hex digit to it's 4-bit binary equivalent.

eg: ① $(9F2)_{16} = (?)_2$

| 9 | F | 2 |
|---|---|---|
| 1001 | 1111 | 0010 |

$\therefore \boxed{(9F2)_{16} = (100111110010)_2}$ Ans

② $(4BAC)_{16} = (?)_2$

| 4 | B | A | C |
|---|---|---|---|
| 0100 | 1011 | 1010 | 1100 |

$\therefore \boxed{(4BAC)_{16} = (0100101110101100)_2}$ Ans

③ $(F3A \cdot CB)_{16} = (?)_2$

| F | 3 | A | . | C | B |
|---|---|---|---|---|---|
| 1111 | 0011 | 1010 | . | 1100 | 1011 |

$\therefore \boxed{(F3A \cdot CB)_{16} = (111100111010 \cdot 11001011)_2}$ Ans

OR Hex → Decimal → Binary

$(\ )_{16} \rightarrow (\ )_{10} \rightarrow (\ )_2$

eg:- ① $(4AB)_{16} = (?)_2$

(i) $(4AB)_{16} = (?)_{10}$

$(4AB)_{16} = 4 \times 16^2 + A \times 16^1 + B \times 16^0$

$= 1024 + 160 + 11 = 1195$

$(4AB)_{16} = (1195)_{10}$

Now, (ii) $(1195)_{10} = (?)_2$

| | | |
|---|---|---|
| 2 | 1195 | |
| 2 | 597 | 1 |
| 2 | 298 | 1 |
| 2 | 149 | 0 |
| 2 | 74 | 1 |

| | | |
|---|---|---|
| 2 | 74 | 1 |
| 2 | 37 | 0 |
| 2 | 18 | 1 |
| 2 | 9 | 0 |
| 2 | 4 | 1 |
| 2 | 2 | 0 |
| | 1 | |

$(10010101011)_2$ Ans

## ⇒ Binary to Hex conversion :-

The conversion from binary to hex is just the reverse of the hex to binary conversion. The binary no. is grouped in groups of four bits & each group is converted to its equivalent hex digit.

e.g:- ① $(1110100110)_2 = (?)_{16}$

$\underbrace{0011}_{3}\ \underbrace{1010}_{A}\ \underbrace{0110}_{6}$

$\therefore \boxed{(1110100110)_2 = (3A6)_{16}}$ Ans

② $(1011011011)_2 = (?)_{16}$

$\underbrace{0010}_{2}\ \underbrace{1101}_{D}\ \underbrace{1011}_{B}$

$\boxed{(1011011011)_2 = (2DB)_{16}}$ Ans

③ $(01011111011.011111)_2 = (?)_{16}$

$\underbrace{0010}_{2}\ \underbrace{1111}_{F}\ \underbrace{1011}_{B}\ .\ \underbrace{0111}_{7}\ \underbrace{1100}_{C}$

$\therefore \boxed{(01011111011.011111)_2 = (2FB.7C)_{16}}$ Ans

## ⇒ Hexadecimal Addition

Addition in hexadecimal is performed in a manner similar to that in decimal. To add in hex all the digits in each column along with the carry if any are added in decimal & it's hex equivalent obtained. The sum term in hex is recorded in that column & carry term is carried to next column.

| + | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| 1 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F | 10 |
| 2 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F | 10 | 11 |
| 3 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F | 10 | 11 | 12 |
| 4 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F | 10 | 11 | 12 | 13 |
| 5 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F | 10 | 11 | 12 | 13 | 14 |
| 6 | 6 | 7 | 8 | 9 | A | B | C | D | E | F | 10 | 11 | 12 | 13 | 14 | 15 |
| 7 | 7 | 8 | 9 | A | B | C | D | E | F | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| 8 | 8 | 9 | A | B | C | D | E | F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 |
| 9 | 9 | A | B | C | D | E | F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
| A | A | B | C | D | E | F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |
| B | B | C | D | E | F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A |
| C | C | D | E | F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B |
| D | D | E | F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C |
| E | E | F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C | 1D |
| F | F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C | 1D | 1E |

$\boxed{x = 16}$

i.e.

$$\begin{array}{rcc} & 1 & 1 \\ & A & E \\ + & C & 5 \\ \hline & 17 & 3 \end{array}$$ Ans

$E = 14$, $A = 10$, $C = 12$

$E + 5 = 14 + 5 = 19$

$19 - x = 19 - 16 = 3$ (1)

$A + C = 10 + 12 = 22 + 1 = 23$

$23 - x = 23 - 16 = 7$ (1)

(1)
$$\begin{array}{rccc} & 1 & & \\ & 4 & A & 6 \\ + & 1 & B & 3 \\ \hline & 6 & 5 & 9 \end{array}$$ Ans

$A + B = 21 - x = 21 - 16 = 5$ (1)

(2)
$$\begin{array}{rccc} & 1 & & \\ & 3 & B & 6 \\ + & 1 & D & 2 \\ \hline & 5 & 8 & 8 \end{array}$$ Ans

$B + D = 11 + 13 = 24 - x$

$= 24 - 16 = 8$ (1)

(3)
$$\begin{array}{r} {}^{1}2\ {}^{1}A\ {}^{1}7\ {}^{1}C\ .\ {}^{1}3\ {}^{1}0\ D \\ +\ \ 8\ D\ 9\ .\ E\ 8\ B \\ \hline 3\ 3\ 5\ 6\ .\ 1\ 9\ 8 \end{array}$$ Ans

$B + D = 24 - r = 24 - 16 = 8$ ①

$E + 3 = 14 + 3 = 17 - r = 17 - 16 = 1$ ①

$C + 9 = 21 + 1 = 22 - 16 = 6$ ①

$D + 7 + 1 = 21 - 16 = 5$ ①

$A + 8 + 1 = 19 - 16 = 3$ ①

## Hexadecimal subtraction :-

Subtraction in hexadecimal is similar to that in decimal. In this case we take borrow as 16 (as r = 16 here)

eg:

$$\begin{array}{r} {}^{B}C\ \ {}^{16+4}4 \\ -\ 7\ \ B \\ \hline 4\ \ 9 \end{array}$$ Ans

$B = 11$, $C = 12$

(1)
$$\begin{array}{r} {}^{3}4\ A\ 6 \\ -\ 1\ B\ 3 \\ \hline 2\ F\ 3 \end{array}$$ Ans

$16 + A = 16 + 10 = 26 - B = 15 = F$

(2)
$$\begin{array}{r} {}^{8}9\ {}^{7}8\ {}^{5}6\ A\ 7\ F \\ -\ 8\ 9\ F\ F\ 6\ 3 \\ \hline 0\ E\ 6\ B\ 1\ C \end{array}$$ Ans

$26 - 15 = 11 = B$

$16 + 5 = 21 - F = 21 - 15 = 6$

$16 + 7 = 23 - 9 = 14 = E$

(3)
$$\begin{array}{r} {}^{A}B\ {}^{F}{}^{16}0\ 8\ E\ .\ {}^{9}A\ {}^{17}1 \\ -\ 7\ 8\ D\ 6\ .\ 3\ B \\ \hline 3\ 7\ B\ 8\ .\ 6\ 6 \end{array}$$ Ans

$17 - 11 = 6$

$16 + 8 = 24 - D = 24 - 13$
$= 11$
$= B$

## → Representation of signed numbers & binary arithmetic in computers

So far, we have considered only positive no. The representation of -ve no. is also equally important. There are two ways of representing signed numbers - sign magnitude & complement form

There are two complement forms : 1's complement form & 2's complement form

Most digital computers do subtraction by 2's complement method but some do it by 1's complement method. The advantage of performing subtraction by the complement method is reduction in the hardware. Instead of having separate digital circuits for addition & subtraction only adding circuits are needed i.e. subtraction is also performed by adders only. Instead of subtracting one no. from the other, the complement of subtrahend is added to the minuend.

(I) In Sign - Magnitude Form an additional bit called the sign bit is placed in front of the number. If the sign bit is 0, the no. is +ve & if it is 1 then the no. is -ve.

e.g. $\underbrace{\boxed{0}}_{\text{Sign bit}}\underbrace{\boxed{1}\boxed{0}\boxed{1}\boxed{0}\boxed{0}\boxed{1}}_{\text{magnitude}} = +41$

$$\underset{\text{sign bit}}{\boxed{1}}\ \underbrace{\boxed{1}\boxed{0}\boxed{1}\boxed{0}\boxed{0}\boxed{1}}_{\text{magnitude}} = -41$$

Under, the signed - magnitude system, a great deal of manipulation is necessary to add a +ve no. to a -ve no. Thus, though the signed - magnitude number system is possible, it is impractical.

## Representation of Signed numbers using the 2's (or 1's) complement method

The 2's (or 1's) complement system for representing signed numbers works like this :

1) If the no. is +ve, the magnitude is represented in it's true binary form & a sign bit 0 is placed in front of the MSB.
2) If the no is -ve, the magnitude is represented in it's 2's (or 1's) complement form & a sign bit 1 is placed in front of MSB.

To represent the no. in sign 2's (or 1's) complement form : determine the 2's (or 1's) complement of the magnitude of the no. & then attach the sign bit.

The 2's (or 1's) complement operation on a signed no will change a +ve no. to a -ve no & vice versa. The conversion of complement to true binary is the same as the process used to convert true binary to complement. The representation of +51 & -51 in both 2's, 1's complement form is shown below

$\underset{\text{sign bit}}{\boxed{0}}\,\underbrace{\boxed{1}\boxed{1}\boxed{0}\boxed{0}\boxed{1}\boxed{1}}_{\text{magnitude}}$ +51 in sign-magnitude form

$\underset{\text{sign bit}}{\boxed{1}}\,\underbrace{\boxed{1}\boxed{1}\boxed{0}\boxed{0}\boxed{1}\boxed{1}}_{\text{mag.}}$ -51 (in sign mag form)

$\underset{\text{sign bit}}{\boxed{1}}\,\underbrace{\boxed{0}\boxed{0}\boxed{1}\boxed{1}\boxed{0}\boxed{0}}_{\text{mag.}}$ -51 (in 1's complement form)

$\underset{\text{sign bit}}{\boxed{1}}\,\underbrace{\boxed{0}\boxed{0}\boxed{1}\boxed{1}\boxed{0}\boxed{1}}_{\text{mag.}}$ -51 (in 2's complement form)

## Special case in 2's complement representation

whenever a signed no. has a 1 in sign bit and all 0s for the magnitude bits, the decimal equivalent is $\rightarrow 2^n$ where $n$ is the no. of bits in the magnitude.

eg:- $\underbrace{1\,0\,0\,0}_{3\,\text{bits}} = -8$ & $\underbrace{1\,0\,0\,0\,0}_{4} = -16$

$2^3 = 8$ $\quad 2^4 = 16$

## characteristics of 2's complement numbers

i) There is one unique zero.
ii) The 2's complement of 0 is 0.
iii) The left most bit can not be used to express a quantity. It is a sign bit. If it is a 1 then the no. is -ve & if it's 0 then the no. is +ve.
iv) For an $n$ bit word which includes the sign bit there are $\boxed{2^{n-1} - 1}$ +ve integers, $\boxed{2^{n-1}}$ -ve integers & one 0, for a total of $2^n$ unique status.
v) Significant information is contained in the 1s of the positive no. & 0s of the -ve numbers.
vi) A negative no. may be converted into a +ve no. by finding it's 2's complement.

## * Methods of obtaining 2's complement of a no.

1) By obtaining the 1's complement of the given no. (by changing all 0s to 1s & 1s to 0s) and then adding 1.

2) By subtracting the given n bit no. N from $2^n$.

Digital System
↓
Binary system
↙ ↘
+ve -ve

Representation
↙ ↓ ↘
Sign magnitude | 1's complement | 2's complement

<u>Sign magnitude :-</u> $\boxed{0}\ \underbrace{1010}_{\text{mag.}}$

↓ sign digit

+ve no. → sign digit = 0
-ve no. → " " = 1

eg:- $\underbrace{1}_{-ve}\ \underbrace{101101}_{mag.}$ = −45

<u>1's comp. or 2's comp. :-</u>

<u>Rules</u> ① $\boxed{+ve}$ → true binary form

eg:- +51

| | | |
|---|---|---|
| 2 | 51 | |
| 2 | 25 | 1 |
| 2 | 12 | 1 |
| 2 | 6 | 0 |
| 2 | 3 | 0 |
| | 1 | 1 |

110011

0 110011

② For $\boxed{-ve}$ no. ⇒ $\underbrace{1}_{\text{show -ve}}\ \underbrace{1's}_{\text{mag.}}$

1 $\underline{2's}$

e.g.- (51) = 110011

∴ -51 in sign mag = 1110011

-51 in 1's complement = 1001100
show -ve no.

-51 in 2's complement = 1001101
(1's comp +1)

How to find out 2's complement?

e.g.- 1100 1's complement of 1100 = 0011

2's complement = 1's complement +1 =

$$\begin{array}{r} 0011 \\ +1 \\ \hline 0100 \end{array}$$

OR 1100
0100 (start write from right side

Date : / /

## 2's complement Arithmetic

The 2's complement system is used to represent -ve no using modulus arithmetic. The word length of a computer is fixed. That means if a 4-bit no is added to another 4-bit no the result will be only of 4-bits carry if any from the 4th bit will overflow. This is called the modulus arithmetic.

e.g:-
```
  1100
+ 1111
------
  1011
```

In the 2's complement subtraction add the 2's complement of the subtrahend to the minuend. If there is a carry out, ignore it look at the sign bit i.e. MSB of the sum term. If the MSB is a 0 the result is +ve & is in true binary form. If the MSB is 1 (whether there is a carry or no carry at all) the result is -ve and is in it's 2's complement form. Take it's 2's complement to find it's magnitude in binary.

**Examples:-**

① Add -75 to +26 using 8-bit 2's comp. arithmetic

⇒ 75 in 8-bit binary = 01001011
-75 in 2's complement = 10110101
+26 in 8-bit = 00011010

```
∴  +26         0 0 0 1 1 0 1 0
   -75    ⇒  + 1 0 1 1 0 1 0 1
   ---         ---------------
   -49         1 1 0 0 1 1 1 1
```

There is no carry, MSB is 1. So, the result

is -ve & is in 2's complement form ∴ magnitude is
Taking 2's complement of above result
i.e.- 11001111 = 00110001 = 49
Thus result is -49 Ans

② Subtract 14 from 46 using 8-bit 2's comp.

46
-14
+32

+14 in binary form = 00001110
-14 in 2's comp. = 11110010
46 in binary form = 00101110

∴ 46 - 14 ⇒

```
   ① ① ① ① ① ①
   0 0 1 0 1 1 1 0
 + 1 1 1 1 0 0 1 0
 ① 0 0 1 0 0 0 0 0
```

ignore the carry

There is a carry ignore it. The MSB is 0
∴ result is +ve & is in normal binary form
Thus result of 00100000 = +32 Ans

## One's (1's) complement arithmetic

The 1's complement of a no. is obtained by simply complementing each bit of the no. i.e.- by changing all 0s to 1s and all 1s to 0s. We can also say that the 1's complement of a no. is obtained by subtracting each bit of the no. from 1. This complement value represents the -ve of the original no. This system is very easy to implement in hardware by simply feeding all bits through inverters. One of the difficulties of 1's complement is its representation of zero. Both 00000000 & its 1's complement 11111111 represent zero. The 00000000 is called the zero & the 11111111 is called -ve zero.

e.g.:- ① add -25 to +14 using 8-bit 1's comp.

25 = 00011001
-25 in 1's comp. form = 11100110
14 = 00001110

+14
-25
-11

$$\begin{array}{r} 00001110 \\ +\ 11100110 \\ \hline 11110100 \end{array}$$ (no carry)

There is no carry & MSB is 1 ∴ result is -ve & is in its complemented form. The 1's comp. of 11110100 is 00001011
∴ result = $(-11)_{10}$ Ans

② Subtract 27.50 from 68.75 using 12 bit 1's complement arithmetic.

+68.75 = 01000100.1100
-27.50 = + 11100100.0111 (in 1's comp. form)

$\boxed{1}$ 00101001.0011
+1 (add the end around carry)
00101001.0100

The MSB is 0 ∴ result is +ve & is in it's normal binary form ∴ result is mag. of 00101001.0100
$= 2^5 + 2^3 + 1 + 2^{-2} = 32 + 8 + 1 + \frac{1}{4}$
$= 41 + \frac{1}{4} = +41.25$ Ans

⇒ <u>Decimal</u> $r = 10$ $(0-9)^{\text{symbols}}$

∴ R's complement ⇒ 10's complement
(R-1)'s " ⇒ 9's complement

$$\begin{array}{r} 99999 \\ -\ 12345 \\ \hline 87654 \end{array}$$ ⇒ 9's complement

$$\begin{array}{r} +1 \\ \hline 87655 \end{array}$$ ⇒ 10's complement

<u>For Binary</u> $r = 2$ $(0, 1)$

(R-1)'s comp. ⇒ 1's comp.
R's comp ⇒ 2's comp.

$$\begin{array}{r} 1111 \\ -\ 1011 \\ \hline 0100 \end{array}$$ (max no.)

⇒ 1's comp. of (1011)

$$\begin{array}{r} +1 \\ \hline 0101 \end{array}$$ ⇒ 2's comp.

⇒ Nine's complement (9's comp.)

The 9's comp. of a decimal no. is obtained by subtracting each digit of that decimal no. from 9.

To perform decimal subtraction using the 9's complement method obtain the 9's complement of the subtrachend & add it to the minuend. Call this no. the intermediate result. If there is a carry, it indicates that the answer is +ve. Add the carry to LSD of this result to get the answer. The carry is called the end around carry. If there is no carry it indicates the answer is -ve & the intermediate result is it's 9's complement. Take the 9's comp of this result & place a -ve sign in front to get the answer.

e.g.:-

① Find 9's comp. of following

(i) 782.54 (ii) 4526.075

$$\begin{array}{r} 999.99 \\ -\ 782.54 \\ \hline 217.45 \end{array}$$ Ans

9's comp. of 782.54

$$\begin{array}{r} 9999.999 \\ -\ 4526.075 \\ \hline 5473.924 \end{array}$$ Ans

② Subtract the following no. using 9's comp. method

(i) 745.81 - 436.62

First, find 9's complement of 436.62

$$\begin{array}{r} 999.99 \\ \text{i.e.} \quad -\ 436.62 \\ \hline 563.37 \\ \hline \end{array}$$

then add 9's comp. of 436.62 to given no.

$$\begin{array}{r} 745.81 \\ -\ 436.62 \\ \hline 309.19 \\ \hline \end{array} \Rightarrow \begin{array}{rl} 745.81 & \\ +\ 563.37 & \text{(9's comp. of 436.62)} \\ \hline \boxed{1}\ 309.18 & \text{(intermediate result)} \\ +1 & \text{(end around carry)} \\ \hline 309.19 & \text{Ans} \\ \hline \end{array}$$

(ii) 436.62 − 745.81

9's comp. of 745.81 ⇒

$$\begin{array}{r} 999.99 \\ -\ 745.81 \\ \hline 254.18 \\ \hline \end{array}$$

$$\begin{array}{r} 436.62 \\ -\ 745.81 \\ \hline -\ 309.19 \\ \hline \end{array} \Rightarrow \begin{array}{rl} 436.62 & \\ +\ 254.18 & \text{(9's comp. of 745.81)} \\ \hline 690.80 & \text{(intermediate result with no carry)} \end{array}$$

no carry ⇒ ans is −ve & intermediate result is in it's 9's comp. form

∴ result ⇒ 9's comp. of 690.80

$$\begin{array}{r} 999.99 \\ \text{i.e.} \quad -\ 690.80 \\ \hline 309.19 \\ \hline \end{array}$$

∴ Answer is − 309.19 Ans

## ⇒ Binary-Coded Decimal code (BCD code)

Circuits & machines can deal readily with binary numbers, but people are used to working with decimal numbers. Moreover, there are considerably fewer decimal digits required to represent a no. than binary. It is much easier to remember just a few digits than it is to remember many. Thus, whenever there is an interface b/w digital ckt. & people, the interface data usually take the decimal form. As a result the digital circuits must utilise some binary code to conveniently represent the decimal no. The code used for this purpose is called BCD code. In a BCD code, each decimal no. is represented by a 4-bit binary number.

e.g.:- To convert $(489)_{10}$ to BCD, the procedure is as:

4 8 9
0100 1000 1001

Note that highest BCD value that a 4-bit binary no. could represent is 9 which would be $(1001)_2$ in binary.

Clearly, only the 4-bit binary no. from 0000 to 1001 are used.

Note that each decimal digit is assigned a 4-bit binary no. even though the binary equivalent may require fewer than four binary places. This way, circuits which use BCD always handle the string of binary bits in 4-place groups when using BCD code.

| BCD code | Decimal digit |
|---|---|
| 0000 | 0 |
| 0001 | 1 |
| 0010 | 2 |
| 0011 | 3 |
| 0100 | 4 |
| 0101 | 5 |
| 0110 | 6 |
| 0111 | 7 |
| 1000 | 8 |
| 1001 | 9 |

remember that all the zeros must be retained, unlike a binary no where leading zeros can be dropped. The BCD code is used when it is necessary to transfer decimal information into and out of a digital machines. example of digital machines includes the digital clock, calculators, digital voltmeters & frequency counters.

# BCD Arithmetic :-

The BCD addition is, therefore, performed by individually adding the corresponding digits of the decimal no. expressed in 4-bit binary groups starting from the LSD. If there is a carry out of one group to the next group or if the result is an illegal code, Then $6_{10}$ (0110) is added to the sum term of that group and the resulting carry is added to the next group. (This is done to skip the six illegal states). The BCD subtraction is performed by subtracting the digits of each 4-bit group of the ~~subtract~~ subtrachend from the corresponding 4-bit group of minuend in binary starting from the LSD. If there is a borrow from the next group $6_{10}$ (0110) to subtracted from the difference term of this group (This is done to skip the six illegal states) In practice, subtraction is performed by the complement method. Since, we are subtracting decimal digits, we must form the 9's complement or 10's complement or the decimal subtrachend & encode that no. in the 8421 code. The resulting BCD no. are Then added.

eg: Perform the following decimal addition in 8421 code.

(i) 25 + 13

| | | | |
|---|---|---|---|
| 2 | 25 | | |
| 2 | 12 | 1 | |
| 2 | 6 | 0 | |
| 2 | 3 | 0 | |
| | 1 | 1 | |

11001

```
 25          0010 0101   (25 in BCD)
+13   =>   + 0001 0011   (13 in BCD)
 38          0011 1000   (no carry, no
              ↓    ↓      illegal code, so this is
              3    8      correct sum)  Ans
```

(ii)

```
  679.6     0110 0111 1001 . 0110     (679.6 in BCD)
+ 536.8   + 0101 0011 0110 . 1000     (536.8 in BCD)
 1216.4     1011 1010 1111 . 1110     (all are illegal codes)

          +0110 +0110 +0110 +.0110    (∴ add 0110 to each)
          10001 10000 10101 1.0100    (propagate carry)
            +1    +1    +1
  0001    0010  0001  0110   .0100    (corrected sum)
   1       2     1     6    .  4
```

Perform following decimal subtraction in the 8421 BCD code

(i) 38 - 15

```
 38          0011 1000   (38 in BCD)
-15   =>   - 0001 0101   (15 in BCD)
 23          0010 0011   (no borrow, so this is
              ↓    ↓      correct ans)
              2    3   Ans
```

(ii) 206.7 - 147.8

```
  206.7     0010 0000 0110 . 0111   (206.7 in BCD)
- 147.8   - 0001 0100 0111 . 1000   (147.8 in BCD)
   58.9     0000 1011 1110 . 1111   (borrow present
                                      ∴ subtract 0110
          -0110 -0110 -.0110
                0101 1000 .1001      corrected diff.)
```

⇒ Excess Three code (XS-3)

The excess-3 code also called XS-3 is a non-weighted BCD code. This code derives it's name from the fact that each binary code word is the corresponding 8421 code word plus 0011 (3). It is sequential code, therefore can be used for arithmetic operations. It is a self complementing code. Therefore, subtraction by this method of complement addition is more direct in XS-3 code than that in 8421 code.

The XS-3 code has six invalid states

0000, 0001, 0010, 1101, 1110, 1111

XS-3 Arithmetic :-

To add in XS-3, add the XS-3 no by adding the 4-bit groups in each column starting from the LSD. If there is no carry out from the addition of any of the 4-bit groups, subtract 0011 from the sum term of those groups (Because when two decimals digits are added in XS-3 & there is no carry, the result is in XS-6). If there is a carry out, add 0011 to the sum term of those groups (because when there is a carry, the invalid states are skipped & the result in normal binary)

To subtract in XS-3 subtract the XS-3 no. by subtracting each 4-bit group of the subtrahend from the corresponding 4-bit group of the minuend starting from the LSD. If there is no borrow from the next 4-bit group, add 0011 to the difference term of such groups (because when decimal digits are subtracted in XS-3 & there is no borrow, the result is in normal binary.) If there is a borrow, subtract 0011 from the

Binary = BCD + 0011

Page No. 87
Date: / /

difference term (because taking a borrow is equivalent to adding six invalid states, so the result is in XS-6). In practice, subtraction is performed by the 9's complement or the 10's complement method.

① Perform addition in XS-3 code.

(i) 5 + 3

∵ Binary = BCD + 0011

| | | | |
|---|---|---|---|
| 5 | ⇒ | 1000 | (5 in XS-3) |
| +3 | | +0110 | (3 in XS-3) |
| 8 | | 1110 | (no carry) |
| | | -0011 | (∴ subtract 0011) |
| | 8 = | 1011 | (correct sum in XS-3) |

Ans

5 → 0101
+0011
1000

3 → 0011
+0011
0110

1000
0011
1011 = 8

(ii) 37 + 28

| | | | |
|---|---|---|---|
| 37 | 0110 | 1010 | (37 in XS-3) |
| +28 | 0101 | 1011 | (28 in XS-3) |
| 65 | 1011 | ①0101 | (carry generated) |
| | +1 | | (propagate carry) |
| | 1100 | 0101 | (add 0011 to correct 0101) |
| | -0011 | +0011 | (subtract 0011 to correct 1100) |
| | 1001 | 1000 | (corrected sum in XS-3) |

7 = 0111
0011
1010

2 = 0010
0011
0101

② Perform the following subtraction in XS-3 code.

(i) 267 − 175

| | | | | |
|---|---|---|---|---|
| 267 | | 0101 | 1001 | 1010 |
| -175 | - | 0100 | 1010 | 1000 |
| 092 | | 0000 | 1111 | 0010 |
| | | +0011 | -0011 | +0011 |
| | | 0011 | 1100 | 0101 |

(correct 0010 & 0000 by adding 0011; correct 1111 by subtracting 0011)

(corrected diff in XS-3)

Ans

(ii) 57.6 − 27.8

| | | |
|---|---|---|
| 57.6 | 1000 1010 . 1001 | [57.6 in XS-3 |
| − 27.8 | − 0101 1010 . 1011 | 27.8 in XS-3] |
| 29.8 | 0010 1111 . 1110 | [correct 0010 by adding 0011 / correct 1110 & 1111 by sub. 0011] |
| | + 0011 − 0011 − 0011 | |
| | 0101 . 1100 . 1011 | correct diff. in XS-3 Ans |

⇒ Positive & Negative logic designation

Inputs & outputs of logic gates can occur only in two levels. These two levels are termed as HIGH & LOW, TRUE & FALSE, ON & OFF and simply 0 & 1

A table which lists all the possible combinations of I/p variable & the corresponding o/p is called a $\boxed{\text{Truth Table}}$. It shows how the logic circuits output responds to various combinations of logic levels at the inputs.

Level logic may be positive logic or negative logic.

A positive logic system is the one in which the higher of the two vol. levels represents the logic 1 & the lower of the two vol. levels represent the logic 0.

A negative logic system is the one in which the lower of the two vol. levels represents the logic 1 & the higher of two vol. levels represents the logic 0.

In transistor - transistor logic (TTL, most

widely used logic family) the vol. levels are +5V and 0V.
logic 1 corresponds to +5V & logic 0 to 0V.

→ **Logic Gates :-** It is fundamental building block of digital ckt. Logic gate is a device that takes decision depends on I/P combinations & then gives output.

A logic gate is a circuit that has one or more input signals but only one O/P signal. All logic gates can be analysed by constructing a truth table. A truth table lists all I/P possibilities & the corresponding O/P for each input.

The three basic logic gates that make up all digital circuits are (i) OR gate
(ii) AND gate
(iii) NOT gate

1) **OR gate :-** An OR gate is a logic gate that has two or more inputs but only one output. However, the O/P Y of an OR gate is low when all I/P are low & Y is high if any one or all I/P is high.
i.e. O/P of an OR gate assumes logic 1 state if even any one of I/P is logic 1 state.


$Y = A + B$ → logical addition, not binary addition

Boolean expression $\boxed{Y = A + B}$ for 2 I/P

For 3 I/P $Y = A + B + C$

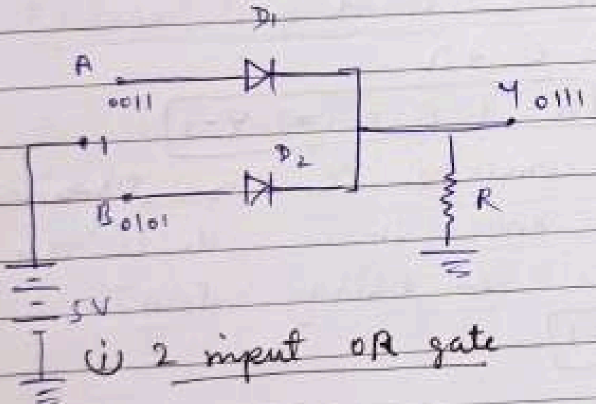
## Truth Table :-

| I/P | | O/P |
|---|---|---|
| A | B | Y |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

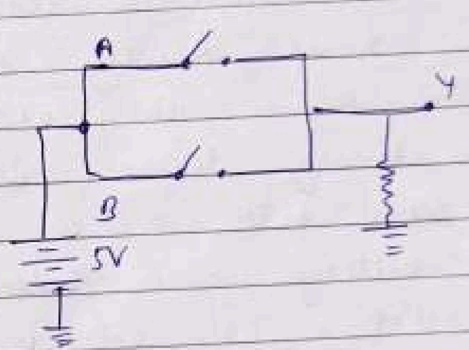
It is called OR gate bcz the o/p is high if any or all inputs are high. For the same reason, an OR gate is sometimes called as "any or all gate".

## * OR gate operation

or Realization of OR gate using diodes :-




(i) 2 input OR gate




(ii) simplified schematic dia.

The I/P vol. are labelled as A & B while the O/P vol. is Y. Note that -ve terminal of battery is grounded & corresponds to 0 state (low level). The +ve terminal of battery (+5V) corresponds to 1 state (high level). There are only four I/P combination possibilities.

(i) when both A & B are connected to ground, both diodes are non conducting. Hence, O/P vol. is ideally zero. In terms of binary when A=0, B=0
⇒ Y = 0

∴ when A=0, B=0 ⇒ $D_1$ & $D_2$ are open ckt & hence, Y = 0

(ii) when A is connected to ground & B connected to +ve terminal of battery, diode $D_2$ is F.B & $D_1$ is non conducting. diode $D_2$ conducts
i.e. $D_1$ = open ckt & $D_2$ = close ckt ⇒ Y is directly connected to B

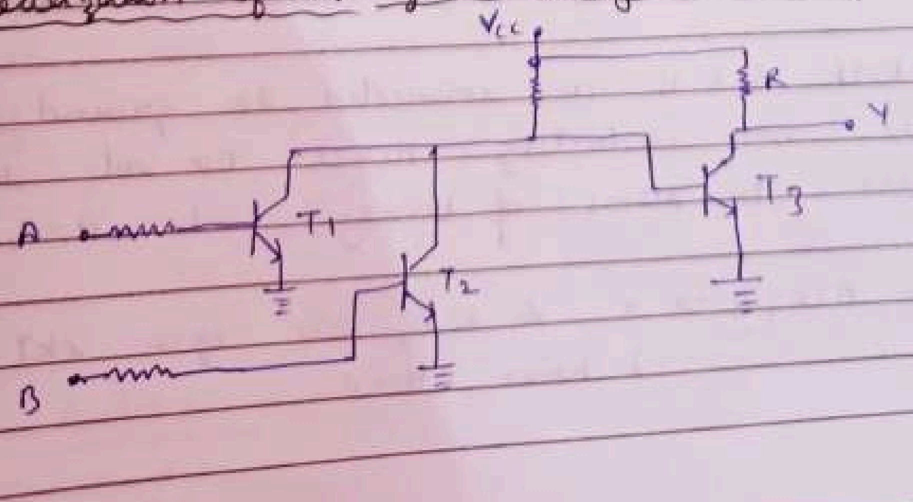
∴ The o/p vol. is ideally +5V. In terms of binary when $\boxed{A=0, \ B=1 \Rightarrow Y=1}$

(iii) when A is connected to +ve terminal of battery & B to the ground, $D_1$ is ON & $D_2$ is off.
i.e. $D_1$ ⇒ F.B. ⇒ close ckt & $D_2$ ⇒ open ckt
& Y is directly connected to A
i.e. again the o/p vol. is +5V
In binary terms when $\boxed{A=1, B=0 \Rightarrow Y=1}$

(iv) when both A & B are connected to +ve terminal of battery both diodes are ON. Since, diodes are in ||'el, o/p vol. is +5V. In binary terms when $\boxed{A=1, \ B=1 \Rightarrow Y=1}$

i.e.- For OR gate, the o/p is high if any or all the I/P are high. The only way to get a low o/p is by having all inputs low.

<u>Realization of OR gate using Transistor</u> :-



(i) when A = 0 & B = 0 ⇒ $T_1$ & $T_2$ both are OFF & $T_3$ is ON ⇒ Y is connected to ground

(ii) A = 0 & B = 1 ⇒ $T_1$ OFF & $T_2$ ON ⇒ base of $T_3$ connected to ground ⇒ $T_3$ is OFF & Y = $V_{cc}$ = 1

(iii) A = 1 & B = 0 ⇒ $T_1$ ON & $T_2$ OFF ⇒ $T_3$ OFF ⇒ Y = 1

(iv) A = 1 & B = 1 ⇒ $T_1$ ON & $T_2$ ON ⇒ $T_3$ OFF ⇒ Y = 1

**Boolean expression :-**

The algebra used to symbolically describe logic fun^n is called Boolean algebra. The "+" sign in boolean algebra refers to the logical OR fun^n. The Boolean expression for OR fun^n is

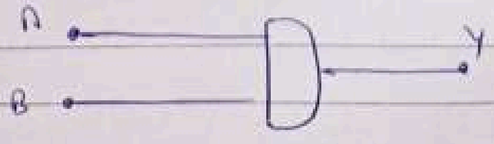
$\boxed{A + B = Y}$

OR symbol

| A + B | = Y |
|---|---|
| 0 + 0 | = 0 |
| 0 + 1 | = 1 |
| 1 + 0 | = 1 |
| 1 + 1 | = 1 |

## (II) AND Gate :-

The AND gate is a logic gate that has two or more I/P but only one O/P. The O/P Y of AND gate is HIGH when all I/P are HIGH. However, the O/P is LOW if any or all I/P are LOW i.e- It assumes logic 1 O/P only when all I/P are at logic 1.




$\boxed{Y = A \cdot B}$ for 2 I/P

$\boxed{Y = A \cdot B \cdot C}$ for 3 I/P

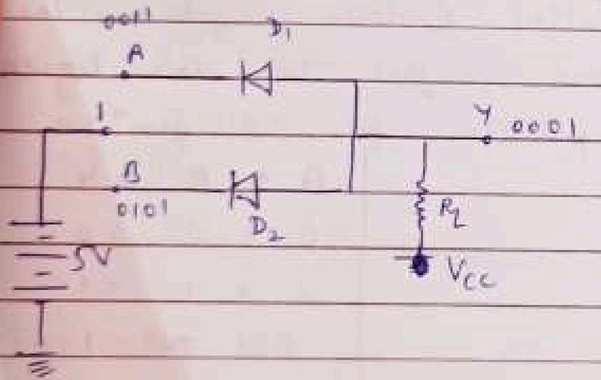
Truth Table :-

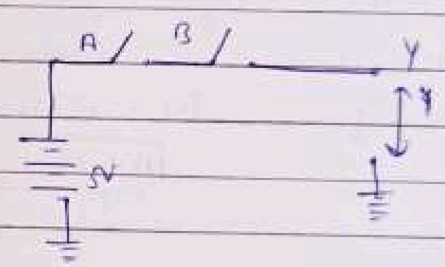
| A | B | Y |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

It is called AND gate bcz o/p is HIGH only when all I/p are HIGH. For this reason, the AND gate is sometime called as "all or nothing gate".

AND gate is also known as Binary multiplier

AND gate operation / Realization of AND gate using diodes :-




(i) when both A & B are connected to ground, both diodes $D_1$ & $D_2$ are F.B. & hence they conduct current. consequently, the two diodes are grounded & o/p vol. is zero. In terms of binary when $\boxed{A=0, B=0 \Rightarrow Y=0}$

i.e. $A=0, B=0 \Rightarrow D_1$ & $D_2$ are ON (F.B)

$\Rightarrow Y=0$

(diode = F.B = short ckt (ON))

(R.B = open ckt (OFF))

(ii) when A is connected to ground & B connected to +ve terminal of battery, $D_1$ is F.B while $D_2$ will not conduct. ∴ $D_1$ conducts & is grounded again o/p vol. will be zero.

i.e. when $\boxed{A = 0 \Rightarrow D_1 \text{ ON} \quad \& \; B = 1 \Rightarrow D_2 \text{ OFF} \quad \Rightarrow Y = 0}$

(iii) when B is connected to ground & A is connected to +ve terminal of battery, the roles of diode interchanged. Now $D_2$ will conduct & $D_1$ doesn't conduct. As a result, $D_2$ is grounded & again o/p vol. is zero.

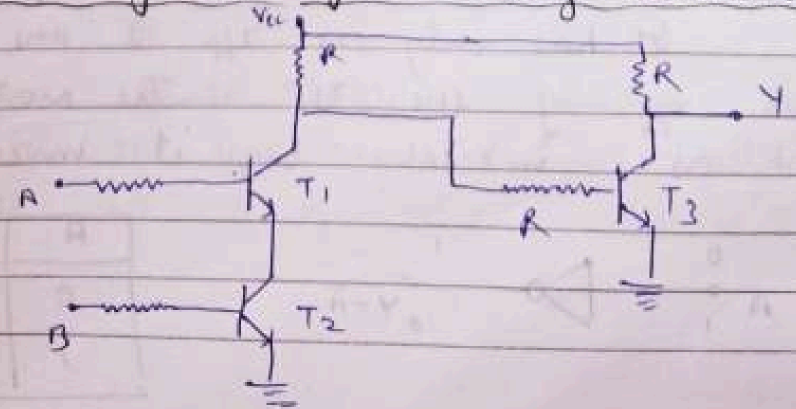
i.e. when $\boxed{A = 1 \Rightarrow D_1 \text{ OFF} \quad \& \; B = 0 \Rightarrow D_2 \text{ ON} \quad \Rightarrow Y = 0}$

(iv) when both A & B are connected to the +ve terminal of battery, both the diodes do not conduct. Now, the o/p vol is equals to $V_{cc}$.

i.e. when $\boxed{A = 1 \quad \& \; B = 1 \Rightarrow D_1 \; \& \; D_2 \text{ are R.B.} \Rightarrow Y = 1}$

i.e. For AND gate, o/p is high if all I/P are high. However, the o/p is low if any or all I/P are low.

<u>Realization of AND gate using transistor :-</u>



(i) A = 0 , B = 0 ⇒ $T_1$ & $T_2$ are off
$V_{cc}$ will ON $T_3$ ⇒ Y connected to ground $\boxed{Y = 0}$

(ii) A = 0, B = 1 ⇒ $T_1$ off & $T_2$ ON
⇒ $T_3$ ON ⇒ $\boxed{Y = 0}$

(iii) A = 1, B = 0 ⇒ $T_1$ ON ; $T_2$ = off
⇒ $T_3$ ON ⇒ $\boxed{Y = 0}$

(iv) A = 1, B = 1 ⇒ $T_1$ & $T_2$ are ON
$V_{cc}$ connected to ground
i.e. $T_3$ is OFF ⇒ $\boxed{Y = V_{cc} = 1}$

Boolean Expression :-

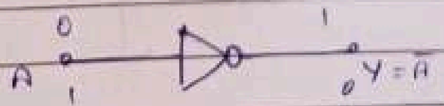
$\boxed{Y = A \cdot B}$
↓
AND symbol

where the multiplication dot stands for the AND operation

| A · B | = Y |
|---|---|
| 0 · 0 | = 0 |
| 0 · 1 | = 0 |
| 1 · 0 | = 0 |
| 1 · 1 | = 1 |

(III) NOT Gate / Inverter :-
I/P → invert → O/P

The NOT gate or inverter is the simplest of all logic gates. It has only one I/P & one O/P where the O/P is opp. of the I/P. The NOT gate is often called as inverter bcz it invents the I/P

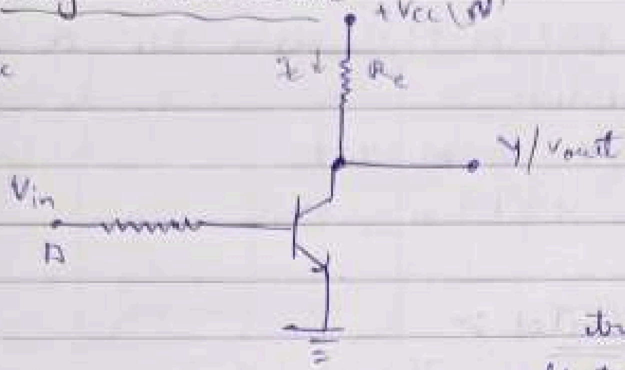



| A | Y |
|---|---|
| 0 | 1 |
| 1 | 0 |

when $V_{in} = 0 \Rightarrow V_{out} = V_{cc} - I_c R_c$
$\downarrow$
$I_c = 0 \Rightarrow V_{out} = V_{cc}$
i.e. $V_{in} = 0$, $V_{out} = 5V$

## Realization of NOT gate :- using Transistor

RTL Logic

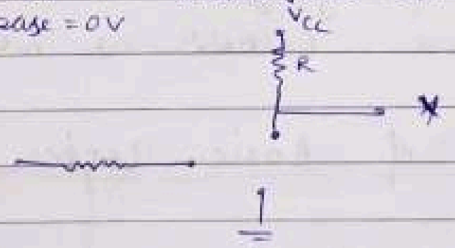



when A is connected to ground, the base of transistor will become -ve. This -ve ptl. cause the transistor to cut-off & collector current is zero & o/p is +Vcc volts. i.e. $\boxed{A = 0 \Rightarrow Y = 1}$

If sufficiently large +ve vol. is applied to A, the base of transistor will become +ve, causing the transistor to conduct heavily. ∴ o/p vol. is zero. i.e. $\boxed{\text{when } A = 1 \Rightarrow Y = 0}$   $V_{in} = 5V \Rightarrow V_{out} = 0$
$\because V_{cc} = I R_c$

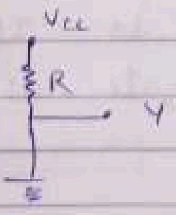
i.e.:- when $A = 0 = 0V \Rightarrow$ Transistor OFF $\Rightarrow$ work as open ckt
∵ base = 0V

∴ n/w




$\Rightarrow$ $\boxed{Y = 1}$

when $A = 1 \Rightarrow$ Transistor ON $\Rightarrow$ n/w is closed ckt




$\Rightarrow$ $\boxed{Y = 0}$

Also, from truth table, whatever the I/p to the inverter the o/p assumes opp. polarity. If I/P is 0 ⇒ o/p is 1 & if I/P = 1, o/p is 0.

Boolean expression :- $\boxed{Y = \bar{A} = A'}$

Bar (‾) or (') above I/P A represents inversion

If A = 0 then $Y = \bar{0}$ or Y = 1
If A = 1 then $Y = \bar{1}$ ⇒ Y = 0

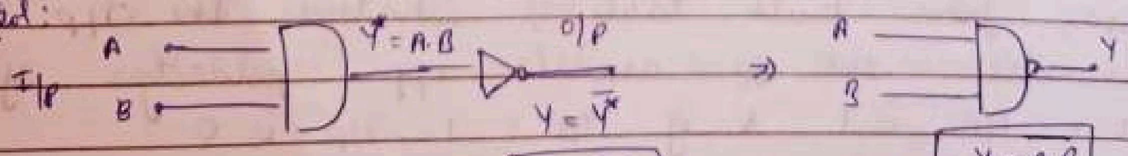
→ **The Universal Gates :-**

Though logic circuits of any complexity can be realized by using only the three basic gates (AND, OR & NOT), there are 2 universal gates (NAND & NOR) each of which can realize logic ckt. single handely. The NAND & NOR gates are also ∴ called universal building blocks. Both NAND & NOR gates can perform all the three basic logic fun^n (AND, NOT & OR). i.e. any gate can be implemented using these (NAND & NOR) gates.

**Combination of Basic Logic Gates :-**

(1) **NAND Gate :-** It is a combination of AND & NOT gate. In other words the o/p of AND gate is connected to I/p of NOT gate. Clearly, the o/p of NAND gate is opposite to the o/p of AND gate.

Symbol:




$\boxed{Y = \overline{AB}}$ $\boxed{Y = \overline{A \cdot B}}$

Truth table for NAND gate is developed by inverting the o/p of AND gate. Let $X = AB$ & $Y = \bar{X}$

| A | B | X | Y |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 |

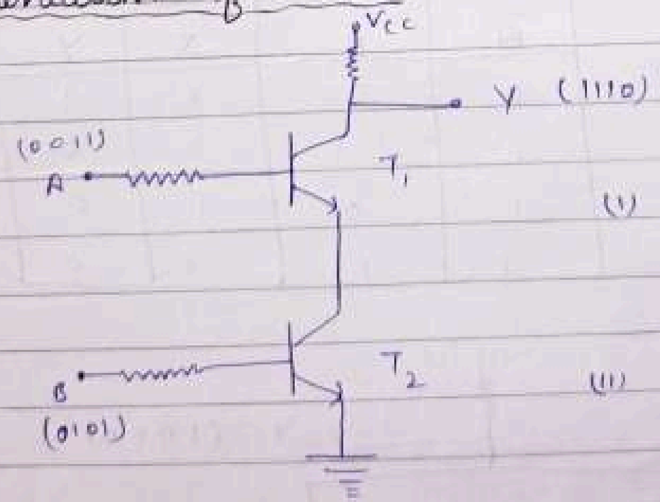
Boolean expression for NAND funⁿ is

$\boxed{Y = \overline{A \cdot B}}$ for 2 I/P

$\boxed{Y = \overline{A \cdot B \cdot C}}$ for 3 I/P

This boolean expression can be read as Y = not A.B. To perform the boolean algebra operation, first the inputs must be AND ed & then the inversion is performed. Note that o/p from a NAND gate is always 1 except when all the inputs are 1.

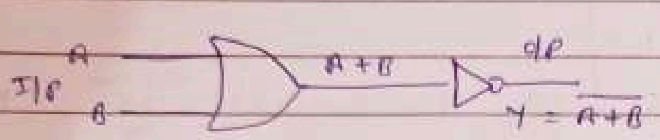
## Implementation of NAND



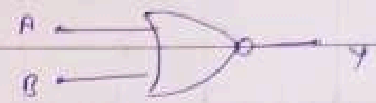

(i) when A = 0, B = 0
→ $T_1$ & $T_2$ are off
∴ Y = Vcc = 1

(ii) when A = 1, B = 0
→ $T_1$ = ON, $T_2$ = OFF

* **NOR Gate :-** It is a combination of OR & NOT gate. In other words output of OR gate is connected to the I/P of NOT gate. Note that output of OR gate is inverted to form NOR gate.




or




Boolean expression for NOR funⁿ is

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

The boolean expression can be read as Y = not A OR B. To perform boolean algebra operation, first I/P must be ORed & then inversion is performed. NOTE That o/p from a NOR gate is high (1) when all the I/P are low (0). If any of the I/P is high (1) → o/p is low (0).

**Truth Table:-**

let $X = A+B$

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

| I/P | | O/P | |
|---|---|---|---|
| A | B | X | Y |
| 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 |

**Realization:-**