

ohm's law.

Date:
Page:

$V = IR$

SI = $\Omega - \text{Ohm}$

- If element follows ohm's law called linear elements.

disadvantages.

- Ohm's law not applicable to unilateral networks.
- Ohm's law not applicable to non-linear elements.
- Ohm's law not applicable for vacuum tubes.

voltage applied across $R_1 \Rightarrow$ dynamic.

Modified ohm's law for electric current is given by :- $J_n = \frac{\sigma e}{e} \frac{dU}{dx}$.

$$R = \frac{\rho l}{A}$$

$$\sigma = \frac{1}{\rho}$$

$$J_n = \frac{I}{A}$$

Ohm's Law is applicable for metal conductors.

Ohm's law only for metal conductors and linear elements. (R, I, V).

Relation between voltage and current depends on the sign of voltage.

Ohm's law applicable when temperature of the conductor is constant.

Resistivity changes with temperature.

$$3V_{ph} I_{ph \text{ lost}} + V = fIR$$

$$\sqrt{3}V_L \bar{I}_{L \text{ lost}} + V = fIR$$

“我就是想让你知道，你不是唯一一个被我爱着的人。”

$$V_{ph} = \frac{V_D f_t}{\sqrt{3} I_{ph} \cdot R} = \left(25 + \frac{I}{2} \right)$$

$$R = \left(\frac{50 + I}{2} \right)$$

$$300 = I \times \left(\frac{50+I}{2} \right)$$

$$300 = \frac{50J + J^2}{2}$$

$$600 = 50I + J^2$$

I = 10.

$$R = 5 \Omega \rightarrow 10 \Omega$$

$$I = 5 A \rightarrow I = ?$$

$$\textcircled{1} \quad \underline{V = 25 \text{ V}} \quad I = \frac{Q}{t}$$

$$28 = 101 - 6 \times 360 = 0$$

$\text{F} = 2.5 \quad \text{O} = \text{Cl}$

$$V = 1400 \quad V = 12$$

$$\frac{50}{5} = \frac{10}{1}$$

10Ω 10Ω $\frac{2 \times 300}{600}$

20 5 V

$$20 = \cancel{J} \times 20$$

$\delta - \epsilon A$

$$\begin{array}{r} \cancel{+} 8 \\ \times 12 \\ \hline \end{array}$$

$$= \frac{1}{20} \cdot 5$$

en
6/3
N: 5R

$$Q = 4 \times 10^{-3} \text{ J/mole}$$

$$\begin{array}{r} .500 = \\ \hline -20 \\ \hline 20 \\ \hline 20 \\ \hline 2 \end{array}$$

$$P = \frac{150}{20} \text{ rix}$$

$$3J_{ph} = \sqrt{3J} L \cdot \frac{JL}{\sqrt{13}}$$

SC \Rightarrow Series R, L, C and so on

Date: _____
Page: _____

parallel connection

- voltage is same
- current gets divided. $\left[\frac{1}{R} = \frac{1}{R_1} + \frac{1}{R_2} + \dots \right]$

series connection

- current is same $[R = R_1 + R_2 + R_3 + \dots]$
- voltage divide

$I = \frac{q}{t} = \frac{\text{Coulomb}}{\text{sec}}$

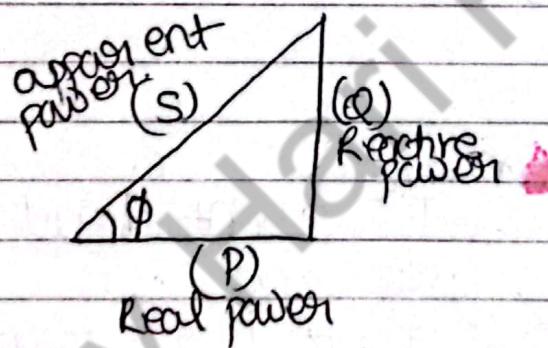
apparent power $S = V \times I$

average or active power $= P = S \cos \phi$

$$\rightarrow P = V \times I \times \cos \phi$$

Reactive power is given by $= Q = S \sin \phi$

$$\rightarrow V \times I \times \sin \phi.$$

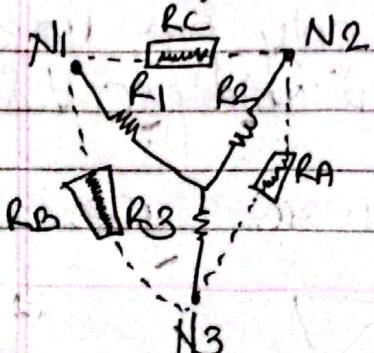


Heat = 1 calorie $\Rightarrow 4.2 J$

$J = \omega x t \cdot [E = P \times T]$.

KCL \Rightarrow Incoming current = outgoing current

KVL \Rightarrow Total sum of Voltage drop in circuit = 0.



$$R_1 = \frac{R_A R_C}{R_1 + R_2 + R_3}$$

$$R_2 = \frac{R_A R_B}{R_1 + R_2 + R_3}$$

$$R_3 = \frac{R_C R_B}{R_1 + R_2 + R_3}$$

$$R_A = \frac{R_1 R_2 + R_2 R_3 + R_3 R_1}{R_3}$$

Linear circuit :- the circuit's parameter which do not change with respect to voltage & current
eg:- parameters R,L,C

Non Linear circuit :- parameter changes with respect to voltage & current
eg:- waveforms, R,L,C, [diode, transistor]

Unilateral :- current flows in 1 direction] circuit
bilateral :- current flows in both direction] allow

Active :- circuit with atleast 1 active element
eg:- transistor, opamp

Passive :- circuit with only 1 passive element
eg:- R,L,C.

⇒ # Networks theory :-

superposition theorem

voltage source or short circuit

current source or open

$$\text{so: } [V=0 ; I=\infty]$$

superposition theorem or complex circuit or
Norton's theorem or Thevenin equivalent circuit \rightarrow it is not JTS

Superposition theorem \rightarrow only for linear circuit
 \rightarrow applicable for V & I not P

application requires two or more sources in the circuit

Independent Voltage Source - S.C
Independent Current Source - O.P.C

Date: _____
Page: _____

Thevenin's theorem \rightarrow can be applied for both AC & DC circuits.

but AC circuits of consisting only linear elements (R, L, C).

- The power dissipation of the Thevenin's equivalent is not identical to the power dissipation of the real system.

Norton's theorem:-

$$I = I_N \frac{R_N}{R + R_N}$$

$$V = I R$$

R_L & X_L variable : $\begin{cases} R_L = R_s \\ X_L = -X_s \\ Z_L = Z_s^* \end{cases}$

$$\frac{10 - 10}{1} \frac{1}{2}$$

R_L varied, X_L constant

$$\Rightarrow R_L = \sqrt{R_s^2 + (X_L + X_s)^2}$$

$$R_L \text{ varied, } X_L = 0 \Rightarrow R_L = \sqrt{R_s^2 + X_s^2}$$

Power transfer is 50%.

$R_L = R_{th}$ (for Thevenin's)

Maximum power transferred = $\frac{V_s^2}{4R_L}$.

If source impedance is complex then load impedance has to be complex conjugate for maximum power transfer to occur.

Maximum efficiency is not related to maximum power transfer.

$I_{rms} = \frac{I_m}{\sqrt{2}}$

Resonant frequency in series-parallel circuit
of $\text{Res} \parallel \text{Cap}$ is $\frac{2\pi f}{\sqrt{LC}}$.

Active power - The power which is actually consumed or utilized in an AC circuit is called active power or true power or real power.

Reactive power : - the power which flows back & forth in both direction, in the circuit or reacts upon itself is called reactive power. ($Q = V I \sin \phi$)

Peak value = RMS value \times peak factor.

Average value = $\frac{\text{RMS value}}{\text{form factor}}$

for star connection (λ)

$$I_L = I_{ph}$$

$$V_{ph} = \frac{V_L}{\sqrt{3}}$$

$$P_\lambda = 3(I_{ph})^2 R$$

for delta connection

$$V_L = V_{ph}$$

$$I_{ph} = \frac{I_L}{\sqrt{3}}$$

$$P_\Delta = 3(I_{ph})^2 R$$

$$P = 3V_{ph}I_{ph} \cos \phi$$

$$P = \sqrt{3}V_L I_L \cos \phi$$

The average e.m.f during the positive half cycle of an AC supply of peak value E_0 is $\frac{2E_0}{\pi}$

$$\text{eg:- } E = E_0 \sin \omega t$$

$$E = E_0 \cos \omega t$$

$$\omega = \frac{2\pi}{T}$$

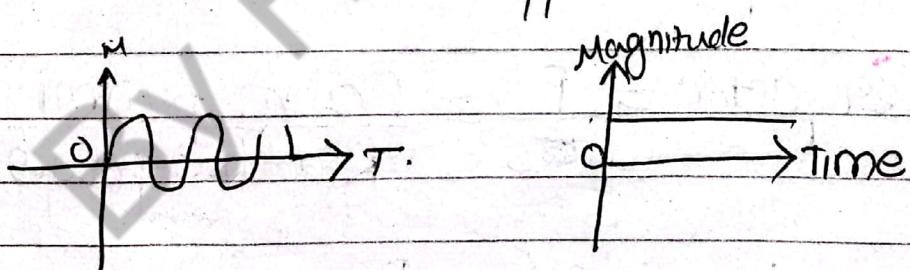
for positive half cycle $t = \frac{T}{2}$.

$$E_{\text{av}}(\text{half}) = \frac{E}{T}$$

$$E_{\text{av}} = \frac{2E_0}{T} \int_0^{T/2} \sin \omega t dt$$

$$E_{\text{av}} = \frac{2E_0}{\pi}$$

AC :- current that can change the direction & magnitude and polarity at regular intervals
- also repeatedly changes direction or reverses its direction opposite to DC.



AC तरीके DC या विपरीत हुन्हा तरीके torque 'O' गणित।

$I_{\text{rms}} = \frac{I_0}{\sqrt{2}}$ peak current.

$$\# P = V_{\text{rms}} \times I_{\text{rms}} \Rightarrow \frac{V_m}{\sqrt{2}} \times \frac{I_0}{\sqrt{2}} = \frac{I_0^2 R}{2} \quad (V_m = I_0 R)$$

$$\# P = VI = \frac{V^2}{R} = I^2 R$$

$$\# i = 10 \sin 314t$$

$$i = I_m \sin \omega t$$

$$\omega = \frac{2\pi}{T}$$

$$\frac{314}{2\pi} = \frac{1}{T} \quad \therefore T = 0.020$$

$$f = \frac{1}{T} = \frac{1}{0.020} = 50 \text{ Hz}$$

Mean value of current is minimum for the alternating current.

Peak value is maximum gained by AC.

we cannot measure AC current by DC ammeter because DC ammeter measures the mean value of current and mean value of AC is zero.

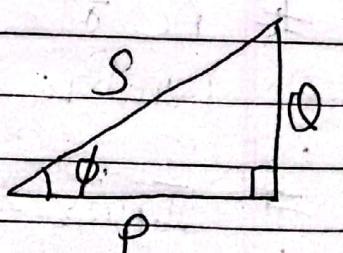
Reactive power using

$$X \# \text{power factor} = \cos \phi = \frac{\text{Active power}}{\text{Apparent power}} = \frac{P}{S}$$

$$\text{Power factor} = \cos \phi = \frac{P}{S}$$

P = Active power

Q = Reactive power



for sinusoidal alternating wave form :

- amp
- # peak to peak value of voltage (V_{P-P}) = $2 \times V_P$
 - # peak value of voltage (V_P) = $\sqrt{2} V_{rms}$
 - # RMS value = $\frac{V_P}{\sqrt{2}}$
 - # $V_{avg} = \frac{2V_P}{\pi}$

power factor for purely resistive circuit is 0°
 $PF = \cos 0^\circ$
 $PF = 1$ (Unity)

power factor for purely inductive circuit,
lags by 90°
so, $PF = 0$ lagging.

power factor of purely capacitive circuit leads by
 90° so, $PF = 0$ leading.

SO; L , R , C
'I' lagg 'V' 90° ; 'I' lead 'V' 90° .
 $I = current$; $V = voltage$.

Neutral is available in star not in delta.

Peak factor = $\frac{\text{Maximum value}}{\text{RMS value}}$ ← Peak value.

form factor = $\frac{\text{RMS value}}{\text{Average value}}$

oscillator \Rightarrow दृष्टि = 1.
transistor = $B_d = \frac{\beta}{\beta+1}$.

Increasing order of turn-off times is:-

MOSFET > BJT > IGBT > Thyristor (SCR)

MOSFET are of 2 types \rightarrow D-MOSFET
 \hookrightarrow e-MOSFET

Depletion Mosfet & enhancement Mosfet.

stability factor for common emitter (S) = $1 + \beta$.

Hartley oscillator is a LC oscillator
used for generation of radio freqn.

Crystal oscillator \rightarrow fixed frequency.
stable \rightarrow Tidcker feed back \rightarrow Nearly fixed frequency.
oscillator

(AF) \rightarrow Wien Bridge oscillator \rightarrow 1Hz to 1MHz
LC \rightarrow Phase-shift oscillator \rightarrow 1Hz to 10MHz
Unstable \rightarrow Hartley's oscillator \rightarrow 10kHz to 100MHz
 \rightarrow Colpitts oscillator \rightarrow 10Hz to 100MHz.

oscillator employs positive feedback

- oscillator doesn't have any input.
- A = amplifier gain
- B is feedback gain.

- $AB = 1$ (should satisfy Barkheuser criterion)

The loop gain must be unity or greater

$$|AB| \geq 1.$$

$$\text{Crystal Oscil} \quad QF = \frac{1}{R} \sqrt{\frac{L}{C}}$$

Date _____
Page _____

Damped oscillation

whose amplitude goes on decreasing or increasing continuously with time.

Damped

Amplitude goes on decreasing with time is under damped.

Amplitude goes on increasing with time is over damped.

Undamped : - whose amplitude remains constant with time.

A crystal diode is an rectifier

RC oscillator :- $f = \frac{1}{2\pi RC \cancel{\sqrt{A}}$

LC oscillator : - $f = \frac{1}{2\pi \sqrt{LC}}$

for LC :

$$T = \frac{2\pi}{\omega}$$

$Q = Q_0 \cos(\omega t)$ → The charge on capacitor varies sinusoidally.

$I = I_0 \sin(\omega t)$ → The current in circuit
↳ max. current at any time t .

The relation between the maximum charge and maximum current is :-

$$I_0 = C_0 Q_0$$

Crystal oscillator:-

- Commonly used for rectification (A.C. to pulsating D.C.)
- Used as clipper to clip portion of A.C.
- Used as clumper to change the ref. voltage
- Used as switches in many electronic circuit
- Used in voltage multiplier to increase the output voltage
- Used in power supplies
- Used as detector

Crystal oscillator generates electrical oscillation of constant frequency based on piezo electric.

Amplifier:

Overall gain in amplifier is given by :-

$$A_v = 20 \log(A)$$

	Class A	Class B	Class C
Conduction angle	360°	180°	less than 180°
Max. efficiency	50%	78.5%	50% 90%
Linearity	Highest	lower than A	Poor
Distortion	Low	Small to moderate	Highest

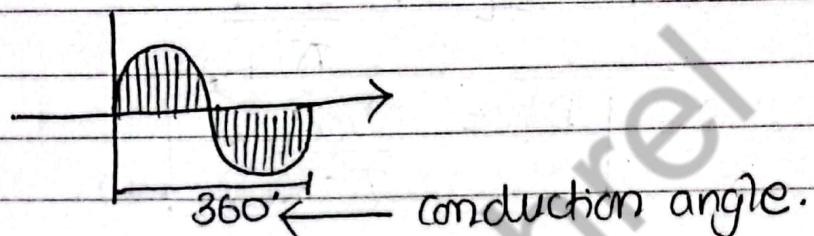
Class AB : \rightarrow More than 180° and less than 360°]

$$\eta = 50\% + 0.78.5\%$$

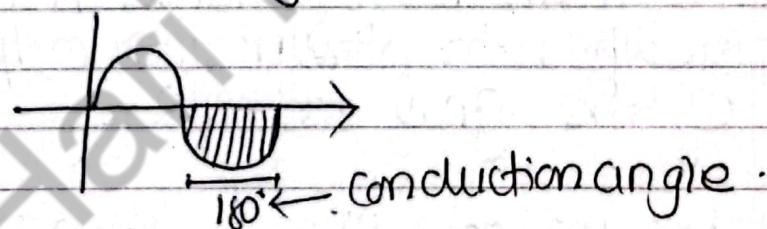
Minimum Present less.

- # Push pull amplifier is used as power amplifier
- consist of two transistors → NPN
 - capable of generating high gains. → PNP.

Class A :- No cross over distortion occurs as they are biased in the center of load line.



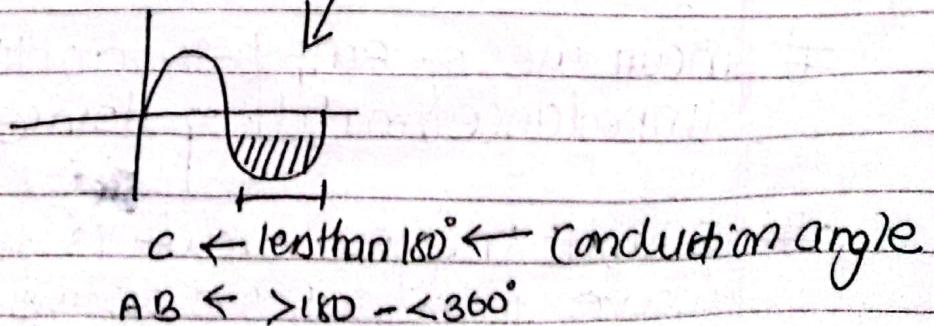
Class B:- Large amount of crossover distortion due to biasing at the cut-off point



Class AB:- crossover distortion is avoided as the biasing level is above the cut-off.

AB also same.

Class C :-



Class B amplifier, the power dissipation by each transistor

$$P_d = V_L I_L$$

Then, maximum power dissipated by the transistor

$$V_L = \frac{V_{cc}}{\pi}$$

and $I_L = \frac{V_{cc}}{\pi \cdot R_L}$

so; $P_d = \frac{V_{cc}^2}{\pi^2 R_L}$

Combining 2 push pull class B amplifiers we can make push pull amplifier.

maximum no. of transistors used in class B amplifier to obtain the output for a full cycle of the signal is:- 2.

class AB operation is often used in power amplifiers to overcome cross over distortion.

tuned amplifier is used in:- RF frequency applications.

main use of amplifier emitter follower is as impedance matching device.

Schmitt trigger circuit is used to convert sine wave into square wave.

- It is comparator circuit with positive feedback.
- Output changes when I/P exceed ; 2 states V_{sat} & V_{off}

The large signal bandwidth of an opamp is limited by its slew rate.

The ideal input resistance of an opamp is ∞ .

- input impedance = infinity.
- open loop impedance (open loop) $\ll 0$
- voltage gain = ∞ .
- common-mode voltage gain = 0.
- $V_{out} = 0$ (when both inputs are same)
- output can change instantaneously (infinite slew rate).

Single Bus structure is used to connect the I/O devices.

Multi bus is an open system bus architecture for designing general purpose 8, 16 or 32-bit micro computer system.

In an RC coupled amplifier, frequency response is improved with Higher C.c. (coupling capacitor).

$$AV = \frac{V_{out}}{V_{input}}$$

Chapter - 2

The binary code of $(21 \cdot 125)_{10}$.

$$21 \rightarrow 10101.$$

$$0.125 \times 2 = 0.25$$

$$0.25 \times 2 = 0.5$$

$$0.5 \times 2 = 1$$

$$\underline{001}.$$

$$\therefore (10101.001)_2$$

BCD convert of $(81)_{10}$.

$$8 = 1000$$

$$1 = 0001$$

$$\therefore (81)_{10} = (10000001)_{BCD}$$

BCD \rightarrow each digit represented by 4-bit binary.

$(110)_x = (132)_4$.

$$\Rightarrow 1x\alpha^2 + 1x\alpha^1 + 0x\alpha^0 = 1x4^2 + 3x4^1 + 0x4^0$$

$$= \alpha^2 + \alpha + 0 = 16 + 12 + 0$$

$$\alpha^2 + \alpha - 30 = 0$$

$$\boxed{\alpha = 5}$$

2's complement of $-g$.

$$\begin{array}{r} -g = 11110111 \\ 00001000 \\ +1 \\ \hline 00001001 \end{array}$$

wrong.

~~6~~

4bit

$$(+g) = 00001001 = (1001)_2$$

$$1's \text{ comp} = 0110$$

$$2's \text{ comp} = +1$$

$$(0111)_2$$

(-g)

$$\begin{array}{r} 11110111 \\ \hline 4bit \quad 1bit \end{array}$$

adding this don't affect the MSB.

Decimal number $(85)_{10} \Rightarrow 11000101$ in:-

a) 8421

b) 2421

c) 4421

d) 2221

Rough :- $(85)_{10} = (01010101)_2$

for 85 \Rightarrow 4421 4421
 $1100 \quad 0101$

$$\begin{array}{r} 8421 \\ 2421 \\ 4421 \\ 2221 \\ \hline 11000101 \end{array}$$

but after 10.

$$\begin{array}{r} 8421 \quad 8421 \\ 1000 \quad 0101 \\ \hline 10000101 \end{array}$$

Logical low = 0
logical high = 1.

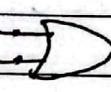
Most commonly used logic levels in digital electronics are TTL and CMOS. (complementary metal oxide-semicond.)

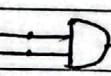
↳ Transistor-Transistor logic
high around (2-5 V)
low around 0V.

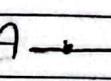
In CMOS → high close to 5V
low close to 0V.

Logic gates:-

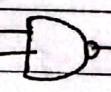
Basic gates

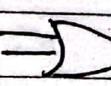
OR Gate \Rightarrow  $y = (A + B)$

AND Gate \Rightarrow  $y = A \times B$.

NOT Gate \Rightarrow  $y = \bar{A}$

Universal gates:

NAND \Rightarrow  $y = (A \cdot B)$

NOR \Rightarrow  $y = (A + B)$.

Derived or arithmetic gates :-

- XOR gate

- EX NOR gate.

$$\text{XOR gate} \Rightarrow Y = A\bar{B} + \bar{A}B. [0, 1, 1, 0]$$

$$\text{EXNOR} \Rightarrow Y = AB + \bar{A}\bar{B} [1, 0, 0, 1]$$

Name AND form OR form.

$$\textcircled{1} \text{ Identity law } 1 \cdot A = A \quad 0 + A = A$$

$$\textcircled{2} \text{ Null law } 0 \cdot A = 0 \quad 1 + A = 1$$

$$\textcircled{3} \text{ Idempotent law. } A \cdot A = A \quad A + A = A.$$

$$\textcircled{4} \text{ Inverse law } A \cdot A' = 0 \quad A + A' = 1$$

$$\textcircled{5} \text{ Commutative law } AB = BA \quad A + B = B + A$$

$$\textcircled{6} \text{ Associative law } (AB)C \quad (A+B)+C = A + (B+C)$$

Absorption

$$\textcircled{7} \text{ Distributive law } A(A+B) = A \quad A + AB = A.$$

$$\textcircled{8} \text{ Distributive law } A+BC = (A+B)(A+C) \quad A(B+C) = AB+AC$$

$$\textcircled{9} \text{ De-Morgan's law } (AB)' = A' + B' \quad (A+B)' = A'B'$$

NOT and OR gate, xgate, xprobe AND Norway.

Logic gates

Min no. of
NOR gate

Min no. of
NAND gate

NOT

1

1

AND

3

2

OR

2

3

Ex-OR

5

4

Ex-NOR

4

5

NAND

4

1

NOR

1

4

Half adder

5

5

Half subtractor

5

5

full adder

9

9

full subtractor

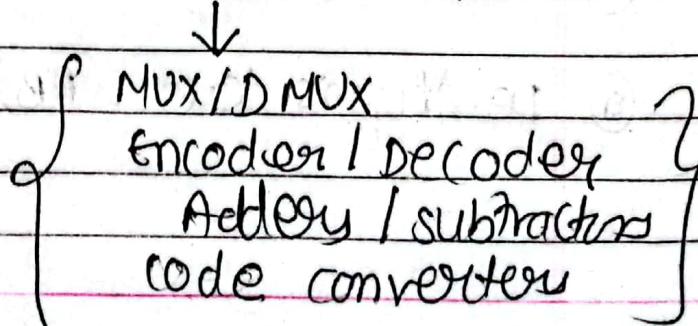
9

9

flipflops

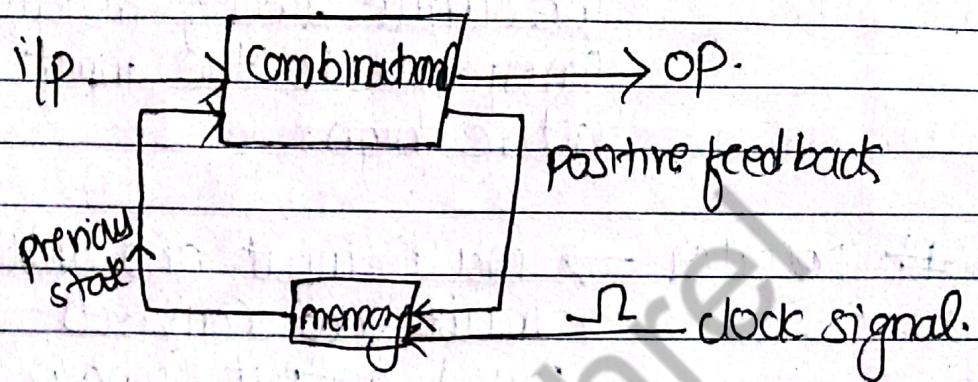
		JK	SR	T	D
	0	nochange	nochange	nochange	0 0
	0	Reset	Reset	Toggle	1 1
	1	Set	Set		
	1	Toggle	Invalid		

A multiplexer is a combinational circuit



multiplexers are many to one data selector.

Sequential circuit, the output depends on both the present and the past values.



Example of sequential logic circuit are :-

- Shift register
- flip flops
- counters..

decimal input or binary to encoder or convert J/E

Demultiplexer changes serial data to parallel.

for (16:1) how many select line are required :-

$$d^4 = 4 \text{ select lines}$$

Multiplexer Shuts. (AND-OR function) \Rightarrow sum of products.

Mux changes parallel to serial data.

Mux is also called universal logic circuit

k-map max variable 6 - In general 5.

digital line data to analog data = Modem.

Encoder \rightarrow 2^n input lines \rightarrow $n \rightarrow$ output lines
The output lines generates binary code corresponds to the input value (which is active high).

Decoder \rightarrow multi-input and multi-output logic circuit that converts coded inputs into coded output (where inputs and outputs are different)

- If code has fewer bits than output code.
- There is one to one mapping from input to output.

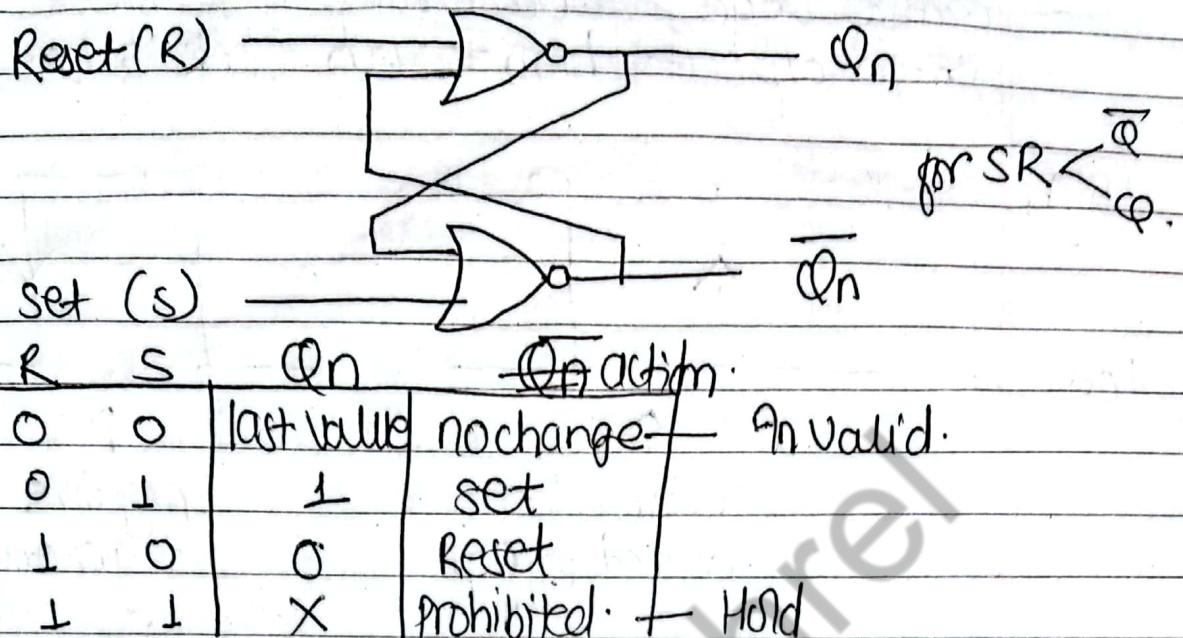
Multiplexer \rightarrow It is a digital switch with multiple input and single output line.

- The selection of a particular input line is controlled by selection lines.
- many to one mapping.

Demultiplexer \rightarrow Single input line to multiple output line.

R-S flip flop.

Date:
Page:



#. flip flop output should be Q and \bar{Q} compulsory

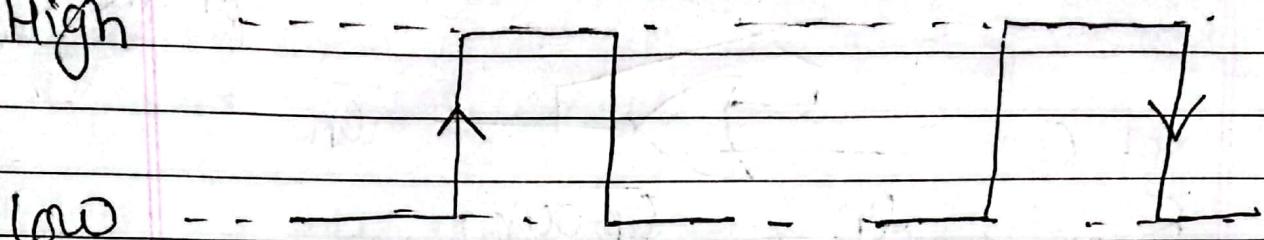
when $R=S=1$ then it's called race condition.

- latches are the building blocks of flip flop.
- flip flop is activated when applied clock.
- flip flop is a single bit storage device
- latches and flip flops are the basic elements to store 1-bit data.
- latches change output immediately when change in input
- flip flop is combination of latches and clock, it changes output that is adjusted by the clock
- main difference in latches and flip flop are
 - latches — no clock
 - flip flop — have clock.

- latches are asynchronous, which means that the output of the latch depends on its input.
- 4 types of latches — SR, JK, D & T.

- negative edge triggered flip flop transfers data from Jp to Qp at high to low clock pulse.

High



Low

positive edge
triggered flip flop

negative edge
triggered flip flop

flip flops:

$Q(t)$	$Q(t+1)$	S R	J K	D	T
0	0	0 X	0 X	0	0
0	1	1 0	1 X	1	1
1	0	0 1	X 1	0	1
1	1	X 0	X 0	1	0

- Triangle on the clock input of JK means edge triggered
- circle means "negative"
- direct line means level triggered
- half circle means level triggered
- difference between NAND based S-R latch & a NOR based S-R latch is - The input of NOR latch are 0 but 1 for NAND.

Due to inverter input of NAND based S-R input,

- Master slave flip flop also referred to pulse triggered flip flop.
- for designing binary counter preferred type of flip flop is JK.
- JK is universal flip flop.
- 3 types of triggering
 - level triggering
 - edge triggering
 - pulse triggering
- Synchronous \rightarrow o/p changes when clock input triggers
asynchronous \rightarrow o/p depends on i/p.
- SR, JK and D inputs are synchronous i/p.
- generally D-flipflops is used to construct shift register.
- 4 types of shift registers are.

SISO	serial in - serial out shift Register
SIPD	serial in - parallel out shift Register
PISO	parallel in - serial out shift Register
PIPO	parallel in - parallel out shift Register

for serial out :- $(n-1)$ clock pulse is required

for parallel out :- no clk pulse is required to output data.

for serial in :- n data bits \Rightarrow n clk required.

for parallel in :- 1 pulse clk required.

SISO - To enter n-bit of data n-clk pulse is required
To output data serially (n-1) clk pulses is required

SIPO:- n serial input data, n-clk pulses are required
To output, no clk pulse is required

PISO - To store n-bit data - 1 clk pulse required
To output data (n-1) clk pulses are required

PIPO - 1 pulse is required no input data
No clk pulse is required to dp data.

4 bit in a Shift register for eg :- 1100

1st phase :- 0000 }
2nd phase :- 0000 }
3rd phase :- 1000 }
4th phase :- 1100 . }

left shift to make other 0.

eg. - $(0100)_2 \Rightarrow (4)_{10}$. — original

Shift :- $(1000)_2 \Rightarrow (8)_{10}$. — after rev

i.e., it becomes multiply by 2.

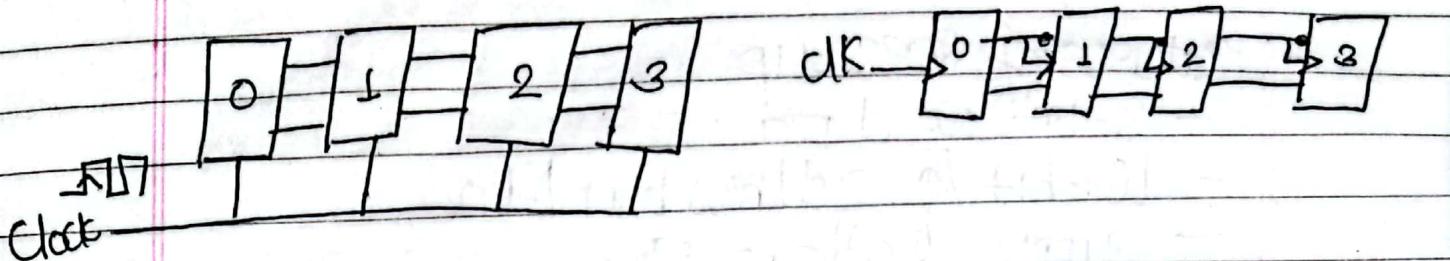
Clock :

0	0	0	0	0
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0

so, clk pulse after 2 is 0 100.

synchronous counter

asynchronous counter.



- all are triggered at once
 - $0 - 3 - 2 - 1$
 - faster
 - extra gates because of randomly can work
 - So complex design and high cost
 - e.g.: Ring Counter, Twisted ring, Johnson ring
 - delay time low
- ~~0 or 1 will not clk~~
 - ~~1 or 2 or 0 will not be clk~~
similarly ~~2 or 3 or 1~~
and ~~3 or 0 or 2~~.
 - up counter $3 - 4 - 5 - 6$
 - down counter $6 - 5 - 4 - 3$
sequence is imp.
 - slower as compared
 - easily circuit, low cost
 - e.g. Ripple counter
 - delay time high

microprocessor:-

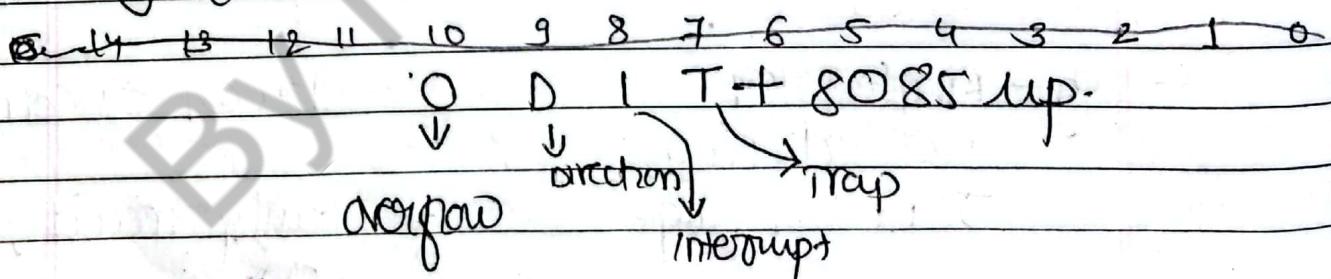
Internal architecture :-

- features of 8085 up:
 - 8-bit of data
 - 16-bit of address bus line
 - supply voltage 5V.
 - maximum frequency - 3MHz (practically 1.5MHz)
 - 5 hardwired interrupts :- RST 7.5, RST 6.5, RST 5.5, TRAP, INTR.
 - 8 bit I/O signal, so it can access 2^8 devices.
 - Two serial I/O signals; SID and SOD.

- flags of 8085

7	6	5	4	3	2	1	0
S	Z	X	AC	X	PE	X	cyl

- flag of 8086



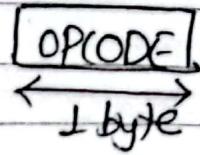
- 8085 has 6 general purpose registers B, C, D, E, H, L ; if data greater than 8bit then store in groups as BC, DE, HL \rightarrow 16bit
- 8086 has 8 general purpose registers AX, BX, CX, DX, BP, SI, DI, SP.

Instruction format:

- one-byte instruction

eg:- MOV C,A.

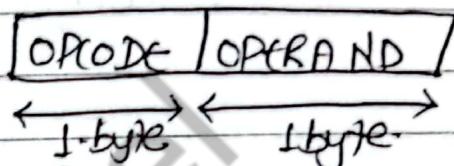
ADD B.



- Two-byte instruction.

eg:- MVI A, 32H.

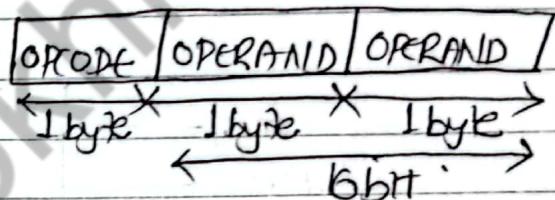
↓ ↓
1 byte 8 bit data.



- Three-byte instruction.

eg:- LDA 2050H.

↓ ↓
1 byte 2byte = 16 bit data.



6 addressing mode

1. Immediate A.N

2. Direct A.N

3. Indirect A.N — X

4. Register A.M

5. Register Indirect A.N.

6. Inherent or Implied A.N

Stack A.M.

Machine cycle in 8085 UP:

opcode fetch (4T to 6T)

memory Read (3T)

memory write (3T)

IO Read IO write — (3T) (3T)

Interrupt acknowledgement — (6T to 12T)

Bus idle cycle — (2T to 3T).

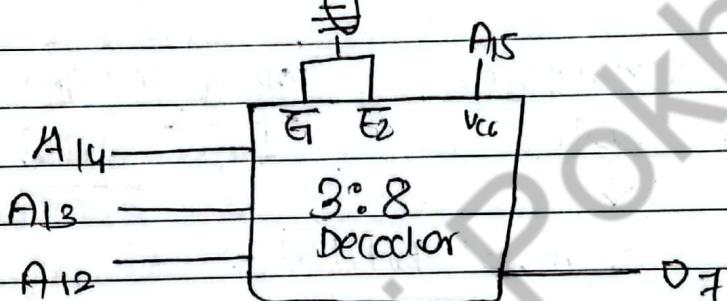
Read = 10
Write = 01

$$\# \text{ Memory address line (SRAM)} = \log_2(2048) \\ = \log_{10}(2048) \\ = \frac{\log 2048}{\log 10} \\ = 11$$

Address line 11 #.

Two ways of address decoding are -

- Using NAND gate
- Using decoder.



$$\underline{3:8} \Rightarrow 2^3 = 8 \#$$

(Q.) I/O mapped I/O

memory mapped I/O.

- | | |
|-----------------------------|---|
| - address is 8-bit | - address is 16-bit |
| - less hardware required | - more hardware required |
| to decode 8-bit | to decode 16-bit |
| - different signals used | - diff. signals used are |
| are <u>IOR</u> & <u>IOW</u> | <u>MEMR</u> & <u>MEMW</u> |
| - The control signal is | - The control signal is |
| <u>Io/m</u> = 1 | <u>Io/m</u> = 0 |
| - The no. of T-states are | - No. of T-state depends upon instructions used |
| 10. | |

3 types of interrupts.

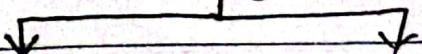
① External ② Internal ③ Software

Interrupts

Interrupts

Interrupts

Types:



maskable

e.g.: EI & DI

I/O priority



- non-maskable

- e.g.: TRAP

- high priority

- caused by

some exceptional

conditions

caused by

programs itself.

RST 7.5

RST 6.5

RST 5.5

if both activated
at same time non-
maskable interrupt first.

- caused by external I/O devices.

DMA :- transfer of data from microcomputer to external device without use of microprocessor.

- memory to I/O devices
- I/O device to memory
- memory to memory
- I/O device to I/O device.

Programmable peripheral Interface:-

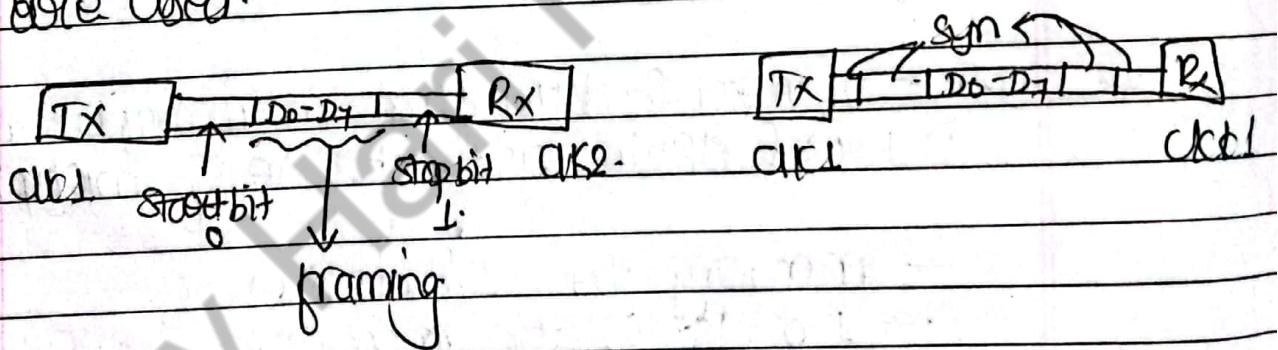
- used for parallel data transfer
- transfer data in various condition from simple I/O to Interrupt I/O.
- The general purpose I/O can be used with all up
- It has 3-8 bit ports; P0F/A, P0F/B & P0F/C
- Two modes - set, reset mode & I/O mode

- each port can be used as individual bits or grouped into two 4-bit ports represented as C-upper and C-lower.
- 8255 PPI.

Asynchronous

Synchronous

- used in system whose speed is below 20Kbps.
- whose speed is above 20Kbps.
- Transmitter & receiver are operated in different clock.
- frequent start & stop bit are used.
- operated at same clock.
- no start & stop bit.



- used for hardware and software implementation.
- used for hardware implementation only.

8085 UP

8086 UP

- 8-bit UP & data bus
- 16-bit address line = 2^{16}
- contains 5 flags \rightarrow S Z AC PCY.
- doesn't support pipelining
- doesn't support mem. segmentation
- operates on 5MHz frequency (3MHz practically).
- 16-bit UP & data bus
- 20-bit address line = 2^{20}
- contains 9 flags ODIT + SZACPCY
- supports pipelining
- supports mem. segmentation
- both same: 5MHz

- Three different parts of microprocessors are:-
 - ① Arithmetic & logic unit (ALU).
 - ② Control unit (CU).
 - ③ Register Unit (RU).

program counter stores what is next to be executed and stack pointer stores address of top of stack memory.

Accumulator

- 8-bit register which contains 8 flipflop
- It stores a maximum of 8-bit of data.

8085 has 6 general purpose registers

B,C,D,E,H,L → each one of 8-bit

If data greater than 8 bit then stored in pair BC,DE & HL which has capacity of 16-bit (2 byte)

Temporary registers in 8085 :-

- W,X and Z
- stores 8 bit of data
- cannot be used or accessed by programmer.
- for storing 16-bit W-Z pair can be used by microprocessor.

- Trap flag is used for on-chip debugging.

Priority :- TRAP > RST 7.5 > RST 6.5 > RST 5.5 > INTR.

↑
Highest

↑
Lowest

also (RST 4.5).

microprocessor core of 3 types:

- Reduced Instruction Set Computer (RISC)
- Complex Instruction set Computer (CISC)
- Explicitly parallel Instruction Computing (EPIC)

- PMOS technology was used by Intel to design 8-bit microprocessor because it was slow but simple.

- 8-bit microprocessor contains $2^8 = 256$ opcodes core possible but it only contains 246.

- 8085 chip is manufactured using NMOS

- PMOS technology is used in 4004 chip.

- NMOS is faster than PMOS.

Interrupts Priority Trigger Mask Vector Address

TRAP (RST4.5)	1 (Higher)	Edge & (level)	Non- maskable	vectorized	0024H
------------------	---------------	-------------------	------------------	------------	-------

RST7.5	2	Edge	maskable	vectorized	003CH
--------	---	------	----------	------------	-------

RST6.5	3	Level	maskable	vectorized	0034H
--------	---	-------	----------	------------	-------

RST5.5	4	Level	maskable	vectorized	002CH
--------	---	-------	----------	------------	-------

INTR. (TRAP)	5 (lowest)	Level	maskable	non- vectorized	0000
-----------------	---------------	-------	----------	--------------------	------

\rightarrow RST4.5 - RST5.5 - RST6.5 - RST7.5
24H - 2CH - 34H - 3CH. } address book.

RST0	—	00H	software interrupt.
RST1	—	08H	
RST2	—	10H	
RST3	—	18H	
RST4	—	20H	
RST5	—	28H	
RST6	—	30H	
RST7	—	38H	

- flags are only affected if they are performed by ALL

MOV A,B	Machinecycle Tstate
all register MOV	
all register ADDR	
SUB R, AND R,	
OF(ODE)	1
	4T

MVI A,8 bit	Machinecycle Tstate
all immediate	
operators ADI, SVI	
ANI, ORI	
AND M, CMP M.	- OF, MR
	2
	4+3

LDA address	→ OF, MR, MR, MR	4	4+3+3+3
STA odd res	→ OF, MR, MR, NW	4	4+3+3+3
IN Odd res	→ OF, MR, I/O R	3	4+3+3
OUT address	→ OF, MR, I/O W	3	4+3+3
MOV Y, M	→ OF, MR	2	4+3
MOV M, R	→ OF, NW	2	4+3

- 8155 is programmable I/O device
- 8283/84 is programmable counter
- 8279 is keyboard display controller
- 825137 is DMA controller.
- Assembler converts the program written in assembly language into machine instructions.
- MOV, ADD are opcodes.
- Instructions which don't appear in object program are called Assembly directives.
- EQU is basically used to replace the variable with a constant value.
- Return directive is used to terminate the program execution.
- last statement of source program should be END.
- assembler stores value and names in symbol Table.
- assembler stores object code in Magnetic disk.

2.5 - Memory device & hierarchy

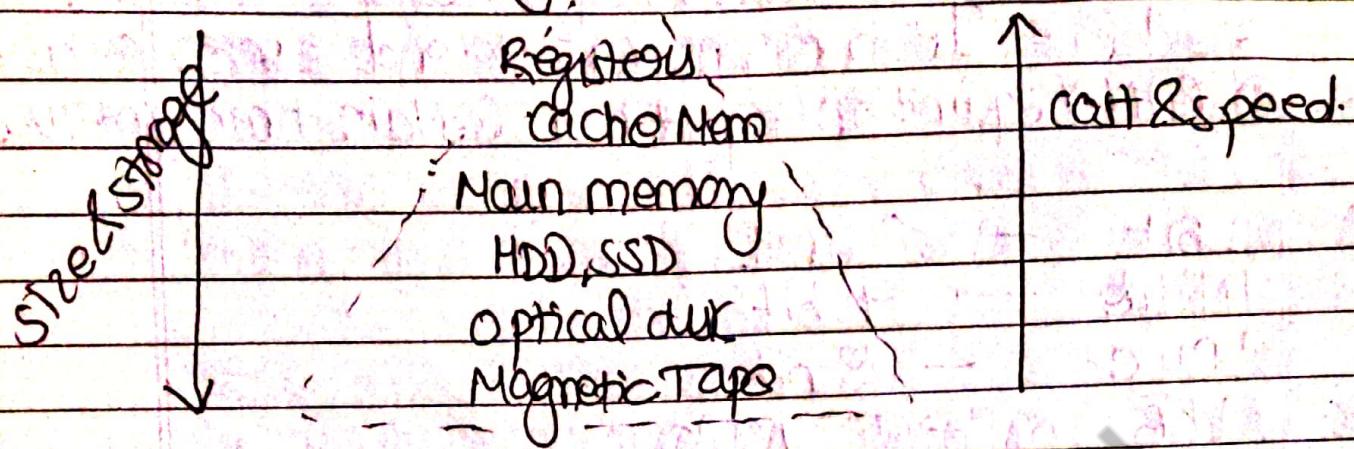
- Cache memory is principle of locality
- Speed of execution of program depends upon cache memory
- DRAM required to refresh in certain time interval

1 Bit	= 0 or 1
1 Nibble	= 4 bit
1 Byte	= 8 bit
1 KB	= 1024 Byte
1 MB	= 1024 KB
1 GB	= 1024 MB
1 TB	= 1024 GB
1 PB	= 1024 TB
1 EB	= 1024 PB
1 ZB	= 1024 EB
1 YB	= 1024 ZB.

PE2A \Rightarrow A22A

- The memory size of n address line and m data line is given by:- $2^n \times m$.
- Auxiliary memory:- also known as external or secondary memory
 - It is non-volatile
 - not directly accessed to CPU.
- Main memory \leftarrow primary memory
 - directly accessed to CPU via memory bus
 - Eg: RAM, ROM.
- micro-program:- The process of code generation for the main memory is called micro-programming

- Main memory - stores instruction & data.
- Control memory - stores micro programs.



- store copy of data or instruction stored in larger memory inside the CPU is Level 1 Cache.
- memory placed between primary cache and memory (Level 2 Cache)
- syntax error are error in source code of program
- internal error due to faulty error
- semantic error due to violation of rules of the meaning of a natural processing of a programming language
- logical error is mistake in program's source code.
→ it is runtime error.
- baud rate :- no of bits transmitted or received per second.
- DMA controller has 3 registers:
 - store starting address
 - word count
 - status of the operation.

- DMA transfers are performed by DMA controller.
- In DMA transfer required signals are given by DMA controller.
- After transfer is completed interrupts are used to notify.
- $R/W = 1$ in DMA controlled then read operation is done.

- If an interrupt occurs while executing a program and the processor is executing the interrupt, if one or more interrupt occurs again, then it is called nested interrupt and interrupt within interrupt.

- If the processor handles more devices as interrupts then it has multiple processing ability.
- NMI - non-maskable interrupt.
- Non-maskable means which interrupt request which input cannot be masked or disabled by any means.
- Programmable interrupt controller is required to handle one or more interrupt requests at a time. (external chip).
- INTR interrupts may be masked using interrupt flag.

- highest priority among all external devices - NMI
- highest priority among all internal devices - TRAP.
- The NMI clock should be high for atleast 2 clock cycle.
- In INTR signal for execution of next cycle it must go high.
- Status of pending interrupt are checked at the end of each instruction cycle.
- After responding the INTR, the IF flag is reset.

Interface 2 8KB RAM starting from the address C000H.

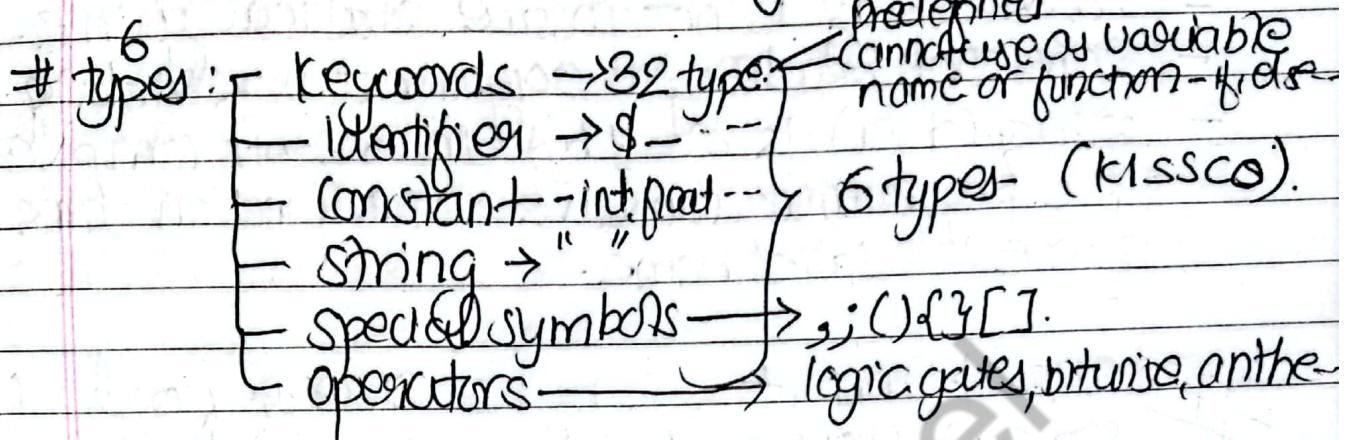
$$\begin{array}{rcl}
 1KB = 1024 & \Rightarrow & 16 \boxed{1024} \quad 0 \\
 1KB = 400 & & 16 \boxed{64} \quad 0 \\
 2KB = 800 & & \qquad \qquad \qquad 4 \\
 8KB = 2080 & & = 400
 \end{array}$$

$$\begin{aligned}
 - \text{Starting add. of RAM1} &= .(000) \\
 \text{Ending Address.} &= SA + (\text{capacity of RAM1} - 1) \\
 &= (000 + (2080 - 1)) \\
 &= DFFF
 \end{aligned}$$

$$\begin{aligned}
 - \text{Starting of RAM2} &= E_H + RAM1 + 1 \\
 &= DFFF + 1 \\
 &= E000
 \end{aligned}$$

$$\begin{aligned}
 - \text{Ending of RAM2} &= SA + (\text{capacity} - 1) \\
 &= (E000 + (2080 - 1)) \\
 &= FFFF
 \end{aligned}$$

C-Tokens :- also known as smallest individual unit in C-programming



C-programming is a general purpose high level programming language that was first developed in early's 1970's.

- C is designed to be a low level system programming language that provides a lot of control over the computer hardware.
- C is a compiled language so, it converts the source code into machine code before it is executed.
- C is a structured programming uses a top-down approach, modular approach for solving problems.
- Father of C is Dennis R. Ritchie.
- Developed in 1972 at American Telephone & Telegraph Bell Laboratories of USA.
- only underscore is used and valid in C.
- all keywords are in lower case.
- Variable name cannot start with a digit.
- volatile is a C-keyword.
- short is C is a qualifier and int is a basic datatype.
- functions in C are always External.

- When former is used, standard directives is searched and when latter is used current directive is searched.
- Jagged array is not possible statically in C-program.
- Standard header for argument is < stdarg.h>.
- sizeof(char) is of 1 byte in 32-bit compiler.
- A C-programming can have same function name and same variable name.

size of short int = 2 } size of float = 4 bytes
 size of char = 1 } bytes. size of double = 8 bytes
 size of int = 4.

- Formatted data :- which can be easily readable by human beings
- Unformatted data :- which are not read by user's eg:- FILE *fp = fopen ("output.bin", "wb") .

operators:

c = 5 d = 2 logical AND.

if c == 5 && d > 5 = 0. (all operands should be true)

c = 5 d = 2 logical OR.

c == 5 || d > 5 = 1 (true because 1 is true).

| c = 5 then

!c == 5 = 0. (True only if the operand is 0).

Date _____
Page _____

single character read - `getchar()`
formatted output - `printf()`
unformatted output - `puts()`

`scanf()` - reads input until it encounters whitespace, newline or end of file.

`gets()` - reads until it encounters newline or end of file.
- It takes whitespace as string.

user defined function:-

```
return-type function_name (parameter_list) {  
    return value;  
}.
```

- pass by value is in default.
 - original value are not modified.
- pass by reference
 - original value are modified.
 - They are used as passing pointer of variable to modify the main variable.
- function calling itself is called recursive function.
 - ↳ 2 types
 - a base case ①
 - a recursive case ②
- ① is the simplest version of the problem that can be solved without recursion. further
- ② the function calls itself to solve a smaller version of the problem.

- Recursion requires more system memory than iteration due to the maintenance of stack.

Array

- delete the entire array pointed to by q;
 - `delete[] q;`
- type array-name [size];
`int number[3] = {1, 2, 3};`
`int num[10] = {1, 2};`
↳ Then remaining 8 are 0.
- no of elements in 2D arrays are determined by math:
$$A[5][5] = \text{Total} = 25$$

[Rows & Column]
- Type array-name [rowsize][columnszie].
- array data is stored in a contiguous block of memory.
- first index is i and last index is j then length of array is $(j-i+1)$.

```
# int arr1[] = {2, 3, 4, 5};  
printf("%d", arr1[2]);  
printf("%d", arr1);  
printf("%d", *(2 + arr1));  
printf("%d", *(arr1 + 2));
```

Output = 4 4 4 4

Integer requires two byte space then size of
int arr[3][4] = (0);

= int ≥ 2 .

$$3 \times 4 = 12 \times 2 = 24 \text{ byte.}$$

arr[1][9] \Leftrightarrow *(arr[1] + 9).

- Simple variable is called a scalar variable.
- array in data can be distinguished using subscript no.
- initializing in array is also called populating an array.
- GetUpperBound returns an integer that represent the highest subscript in the specified dimension of array.
-
- array stores data of same type
- array can be part of structure
- array of structure is allowed.
- structures are used to grouping together different piece of information (of different data type) under the same name.
- length of array is found by :- arr[] = {1, 2, 3, 4, 5}.
$$\text{length} = \frac{\text{size of (array)}}{\text{size of (array[0])}}$$
- syntax to send an array as parameter to function
= func(&array);

- An array is a set of values stored in a contiguous block of memory, while pointer is a variable that stores the address of another variable.

string literals :- "Hello world".

- character array :- char str[6] = "hello".
 ↑ ↑
 size string

- strlen() :- returning length of string excluding null character.

- strcpy() :- copying one string to another

- strcmp() :- comparing strings

- strchr() :- finds first occurrence of a character in string

- strstr() :- To find a substring in a string.

- size of a pointer in 32-bit architecture is 2-byte.

- size of union is determined by size of biggest member
in the union.
 ↓
 longest

- members of unions are accessed as :-

both union-name.member & union-pointer → member

struct structure_name {

 data type member 1;

};

N

Structure

- unique storage area of memory.

- attching value don't affect other members

- individual member can be assigned at a time.

- several members of structure can initialize at once.

- shared by the individual of the location.

- affects other members

- one member can assign at a time.

- only the first member of Union can initialize.

`fprintf` — write formatted data in file
- `scanf` — Read formatted data from file
~~getchar~~ — Read character from file stream
`putc` — write character in file
`gets` — Read a line from the file into stream
`puts` — Write a string from buffer to file
`rewind` — sets the file position indicator in beginning.
`feof` — check if last indicator reached or not
`tell` — Returns the current position
`seek` — set the file position and finds or moves position
`close` — close the file.
`clearerr` — clear the error & eof of file.
`fread` → large amount of data read & write.
`fwrite`

namespace namespace_name {
 // code
}

name::code; ← calling.

- namespace is a mechanism for managing the scope of identifier in large programs.
- can define namespace within a function.
- scope resolution operator ::
- namespace keyword is used to bring all members of a namespace into current scope.
- scope resolution is used to access members of class
- default namespace in C++ is std.
- scope operator is used to signify namespace .(::)
- namespace is used to group class, object and functions.
- main aim of namespace is to structure a program into logical units.

Two function having same name is function overloading
→ either type of argument differs
or
→ no of arguments. 1 parameter.

- compiler automatically selects the appropriate version.
- access level doesn't affect the function overloading.
- both member & non-member functions can be overloaded
- we can't overload return function.
- function overloading is same as constructor overloading.
- when func. doesn't require any parameter, leave blank.

ASCII value
↳ a → 97
↳ A → 65

Date:
Page:

inline function :- The function that are not actually called but rather are expanded in line.

adv. - They can execute much faster than normal function

dis-adv. - If they are long and often called, program grows longer. For this reason only short functions are declared as inline function.

default argument :- If value is passed it used the value else uses the default value.

- arguments which are not mandatory to pass.

- normal argument should pass first.

- default argument should pass at last; to ensure ambiguity doesn't arises.

✓ void test(int x=0, int y=0, int z=0);

↳ default argument with value '0' so, zero argument.

✗ int test(int x=0, y=0, z=0);

If independent then it doesn't return any value so; wrong.

✓ void test(int x~~=0~~, int y, int z=0);

↓
default argument passed at last.

void test(int x, int y=0, int z=0);

✓ - test(5, 6, 7)

↳ test(5)

✓ - test(5, 6)

✗ - test(x) or test()

because 2 default arg.
and 1 normal so, at least 1 value is needed.

- Default arguments are allowed in parameter list of function declaration.
- The using declaration restores over all the base class default arguments.
- The virtual function overrides do not acquire base class declaration of default arguments.
- void is default return type of a function to handle both empty and non-empty values.

pointers and function arguments:-

- Array can never be sent by call-by-value.
- both static and functional arguments can have same name in a different function.
- Command line arguments are those which takes input by user before running a program.
- 256 maximum number of arguments can passed in a single function.

pass by value → actual value is passed / ^{only func.} 1 change.
pass by reference → value passed & variable changed
pass by address or pointer → same as reference but we pass address or pointer

- Object is returned by value, temporary object is created.
- temporary objects are created within a function.
- Syntax of returning object by value:-

classname functionName() { }

- Object by reference :-
classname & functionName() { }
ref. ←

- If the object is declared inside the function then it can't be returned by reference outside of function.
- Only 1 object can be returned at a time.
 - Single value at a time.
 - Through array of objects can be returned from a function.
- Object passed by reference then temporary object is not created.
- Individual data members can be returned.
- The object array can be returned from a function. This is done by putting a className* as a return type of a function.
- It is not necessary if we pass object by reference then it must return by reference itself.
- Two objects can point same memory location.
 - Whenever object is created and instead of calling a constructor, another object is assigned to it and both object point to the same memory location. This can be illustrated by help of return by reference.
- If object is being returned by value then its member values are copied using copy constructor.

- temporary objects are not created in return by reference because it just make the object points to values memory location.
- Reference cannot refer to constant value.
- A variable can have more than 1-references.
- reference provide the alternative name of the variable because we are pointing memory address using the temp variable.
- we cannot create the array of reference.
eg:- `int &arr[] = {&a, &b, &c};` → X

class & object

- a class is a blue print of a object
 - private access specifier only can access inside same class
 - class holds data & functions.
 - :: is used to define the member of class externally.
 - :: operator is used to define the body of any class function outside the class.
 - both struct & union takes same definition of class but differs in the access techniques.
 - class declaration starts with keyword called class.
- class of and ends with ;
- ```

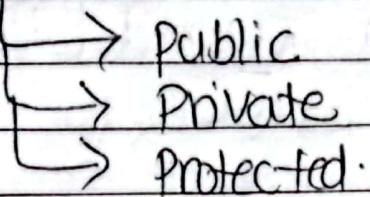
int x;
y;

```
- The data members and function are default private.

- Constructors are used to initialize objects.
- By default all data members & function are public by default. So if no keyword class, user struct then its public.
- Syntax of accessing a static member of a class  
 $A::value$ .  
    ↳ scope resolution operator.
- In classes objects are self-referenced using this pointer inside a member function.  
    eg:- this → value.
- mutable members are those which can be updated even if it a member of a constant object.
  - we can change their value even from constant member function of that class.
- inline function is avoided when
  - function contains static variables
  - functions have recursive calls
  - functions have loop.
- Macros looks like function calls but they are actually not.
  - inline function are defined inside the function or with the keyword inline.

## # Access Specification

3 types



- Private members class can't be inherited.
  - They are accessible from its own class only.
- Protected access is used to make the members private.  
But those members can be inherited
  - Used for code reusability and security.
- Default members are similar to private members
  - They can't be inherited.
- Implicit constructor will always be public; otherwise the class wouldn't be able to have instances.

| #         | within same class | Indenred class | outside of class |
|-----------|-------------------|----------------|------------------|
| Private   | Yes               | No             | No               |
| Protected | Yes               | Yes            | No               |
| Public    | Yes               | Yes            | Yes              |

- object in a class can be accessed using direct member access operator which is (.) operator.
- Objects are the instance of the class.

- Date:  
Page:
- member function are function that are belonging to class.
    - They are inherited as well.
  - 5 types of member function
    - Simple
    - static
    - const
    - inline
    - friend
  - Static member function can be called in the main function using dot, arrow or using scope resolution operator with class name.
  - inline member function are expanded in place of its call.
  - If a non-static function uses static member function then compile time error.
  - static function → This function belong to the class rather than any instance of the class. They can be called without any instance of class.
  - const function :- They can't be modified on which they are called.
  - inline function :- They are expanded in place at the point of call.
  - friend function :- They are not members of class, but they can access private & protected member using friend keyword

- for overloading '()' '[]' or '→' operator a class should not be static member function and should not be friend's function.  
    (☞ must be non-static & not friend function.)
- virtual function defined using virtual keyword.
  - They should not contain any body and defined by subclasses.
- Member function of generic class are auto-generic.
- member function can access all the members of class.
- void A :: disp() {}.
- constructor and destructor function doesn't require any return type.
  - They never have a data type.
- A member function cannot access public member of subclass
- constructor is automatically called when an object is created.
- It is not necessary for constructor to contain a definition body.
- eg:- class student {  
        public : student() {}  
    };
- 3 types of constructor:
  - default ← no need to define always
  - parameterized-
  - copy constructor.

- class C inherits class B, B inherits A; Then while creating the object of class C. the sequence will be.  
→ class A then class B and class C.
- If object is passed by value to copy constructor it will give out of memory error.
- Explicit call to a constructor helps to create temporary instance.
- For constructor overloading each constructor must differ in number of arguments and type of arguments.
- Copy constructor allows the user to initialize an object with the value of another object.
- Default constructor doesn't require any parameters.
- Destructor used to destroy the object when lifetime of class object ends.  
eg:-  $\sim A() \{ \}$ .  
→ no semicolon in cons & destr
- Destructor are called
  - at end of program to destroy objects declared in the main() or global scope.
  - at end of function to destroy object declared at that function scope
  - when user tries to delete using delete operator
  - at end of block to destroy object declared in that block scope.

- Constructor aliases function parameter whereas as destructor does not.
- only 1 destructor is used / allowed.

## # Dmalloc.

- local variable are stored in area called stack.
- Global variable, static variable and program execution are stored in permanent storage area.
- The memory space between these two region is known as heap.
- size of stack remains same during time.
- size of heap doesn't remain same.
- Execution of program is slower than that using static memory allocation.
  - because memory has to be allocated during run time.
- header <stdlib.h> to use dynamic mem. allocation.
  - malloc → allocate for specified size in byte
  - calloc → copy of element, 2 arguments
  - realloc → change allocation of previously allocated.
  - free → use to deallocate from previously allocated.
- singly linked list does not contain any pointer or reference to the previous node.
- Tree is non-linear type / data type.
- Queue → FIFO
- Stack → LIFO.
- array is static memory allocation.

- Linked list, stack, queue → dynamic mem allocation
- Array is preferred over linked list for the implementation of Binary search.
- insertion & deletion can be done from any position in linked list.

## # this pointer:-

- this pointer is passed implicitly when member function are called.
- this pointer is accessible only with non-static fun.
- an object's this pointer isn't the part of object itself.
- eg:- function (&object, parameter).
- The address of the object is available inside the member function using this pointer.
- This pointer used to guard against self-refernce.
- The this pointers are non-modifiable.
- This pointer can be used directly to manipulate self-referential data structures.
- const or volatile are types of pointer.  
eg:- [cv-qualifier-list] classType \*const this;
- friend function can be invoked as a normal function.
- friend function can be declared either in private or public part of class.
- A friend function cannot directly access the member of class.
- friend function use the dot membership operator with @ member name.

eg:- friend class1 class2;

the class2 is the friend of class1 and we can access private & protected data members of class1

- A friend function may or may not be a member of another class.
- friend keyword is placed only in the function declaration.
- A destructor is automatically called when object of class is deleted.
- A constructor is automatically called when object of class is created.
- A destructor can be declared as virtual function but not static, volatile or const.
- polymorphism
  - compile time (function & operator overloading)
  - run time 2 (overriding)

# operator:

- action on 1 operator — Unitary operator (--) .
- action on 2 operator — binary operator (&, ==, /)
  - ternary operator (? :).
- binary operator
  - (+) → adding two operands.
  - (++) → increase by 1
  - (\*) → dereferencing used for accessing values of pointers. (1 operand).

- precedence & associativity of operators don't change by operator overloading.
- Both arithmetic & non-arithmetic operators can be overloaded.

### — Arity

- 3 approaches are used :-
    - overloading unary operator
    - overloading binary operator
    - overloading binary operator using friend function
  - cannot be overloaded  $\rightarrow ?:, ::, . . .$
  - can be overloaded  $\rightarrow +, -, \%, == . . .$
- unary  
o [ ] binary  $\rightarrow 1$

non-static member function unary  $\rightarrow 0$

non-static member function binary  $\rightarrow 1$

friend function unary operator  $\rightarrow 1$

friend function binary operator  $\rightarrow 2$

\* Diamond problem includes Hierarchical and Multiple hybrid inheritance.

- Deriving new classes from existing class is inheritance
- multiple inheritance means deriving a derived class from more than one base class.
- (comma ,) symbol is used to create multiple inheritance
- Adapter and Observer patterns benefit from the multiple inheritances.
- The diamond problem arises when multiple inheritance is used.
- for implementation of multiple inheritance, there must be at least 3 classes in a program.
  - Atleast 2 base class and 1 class to inherit those 2 classes
  - If lenses then it becomes single level inheritance.

Q. - Class A inherits class B and class C.

→ Class B will be first called then class C then class A.

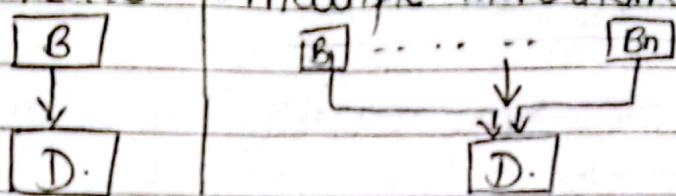
→ Diamond problem arises because the method with same name creates ambiguity and conflict.

- diamond problem can be solved using virtual keyword with same name function.
- if no abstract class is involved, it is compulsory to have constructor for all the classes involved in multiple inheritance.
- private members can't be accessed in derived class in multiple inheritance.
- derived class have access to protected & public members only.
- The derived class must not be abstract.  
because abstract class doesn't have constructor.

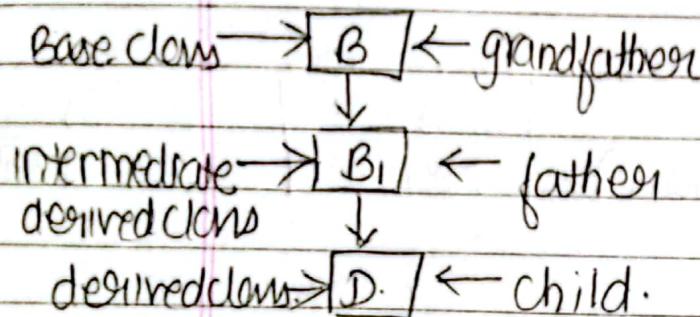
B = Base class  
D = Derived class

Diaper  
Pages

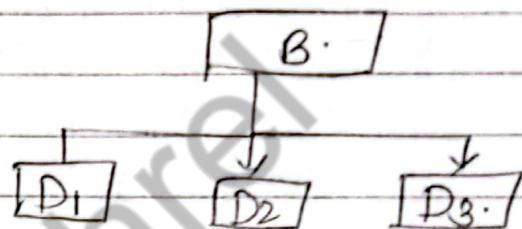
## # single inheritance | # multiple inheritance



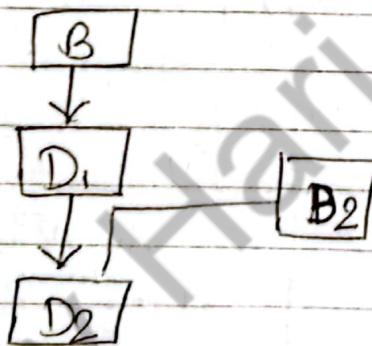
## # multilevel inheritance



## # hierarchical inheritance



## # Hybrid inheritance



Accessibility of:- Public inheritance.

Accessibility      Private      Protected      Public.

## Accessibility :

### Public inheritance:

| Accessibility | Public | Private | Protected |
|---------------|--------|---------|-----------|
| Base class    | Yes    | Yes     | Yes       |
| Derived class | Yes    | No      | Yes       |

### Private inheritance

|               |               |     |               |
|---------------|---------------|-----|---------------|
| Base class    | Yes           | Yes | Yes           |
| Derived class | Yes (private) | No  | Yes (private) |

### Protected inheritance:

|               |                 |     |     |
|---------------|-----------------|-----|-----|
| Base class    | Yes             | Yes | Yes |
| derived class | Yes (protected) | No  | Yes |

| Base class visibility | Derived class |               |           |
|-----------------------|---------------|---------------|-----------|
|                       | Public        | Private       | Protected |
| Public                | Public        | Private       | Protected |
| Private               | NO            | NO            | NO        |
| Protected             | Protected     | Private<br>NO | Protected |

- The class with highest degree of abstraction will be the class at 1<sup>st</sup> level.
- Such as :- A CAR is more abstract than a SPORTS CAR class.
- The level of abstraction decreases as the details comes out.
- How many abstract classes can be used in multilevel inheritance?
  - At least one less than number of levels.
- It is not mandatory to define constructor explicitly.

### # hybrid inheritance:-

- Combination of two or more inheritance type.
- at least 2 types are needed.
- no specific keyword is used for defining hybrid inheritance.
- The destructor are always the reverse of constructors being called.
  - The type of inheritance doesn't matter.
  - The only important concept is sequence of class being inherited.
- destructor is called just before end of object's life.
- abstract class doesn't have destructor.
- There can be only one destructor in a class.
- When object passed by reference, then no constructor is called; so, no destructor is also called at end of program/function.

- An abstract class is defined as a class which is specifically used as a base class and it should have at least one pure virtual function.
- A function declared in base class is called pure virtual function.
  - ↓
  - eg:- virtual return-type function(param) = 0.
  - ↓
  - { } is pure specifier.
- derived class may or may not have an implementation of pure virtual function.
- Pure virtual function has no implementation in the base class, whereas virtual function may have an implementation in the base class.
- The base class has at least one pure virtual function.

## # file handling:-

<fstream> → header used for I/O operations.

- <ofstream> → create output stream in C++.
- <ifstream> → used to create both input & output operations.
- It is possible to combine two or more file opening mode in open() mode.
- ~~if~~ is los::in and los::out are input and output file opening mode respectively.
- los::trunc → to truncate existing file to zero.

A file can be opened 2 ways  
↳ using mem. func. open().

- default mode of opening using ifstream  $\Rightarrow$  ios::out.
- return type of open() method is bool.
- default mode of opening using ifstream  $\Rightarrow$  ios::in
- default mode of fstream  $\Rightarrow$  ios::out | ios::in.
- tell() is used to get current position of pointer.
- seekg() is used to reposition the file pointer.
- move file pointer to start of file  $\Rightarrow$  ios::beg.
- check if file is open :- if(file.is\_open()).
- <fstream> read data
- <iostream> write data.
- read data from file :- file >> data.
- fail() used to check if error occurred in opening & closing of file.
- default mode for opening a file is output
- open file in output mode : fstream file("filename.txt", ios::out);
- good() is used to check if file opened successfully
- getline  $\rightarrow$  read line from file
- putline  $\rightarrow$  write line from file.
- seekg(0)  $\rightarrow$  beginning of file pointer.
- assert  $\rightarrow$  to test a condition and terminate the program if it is false.
- cerr  $\rightarrow$  to write error messages to standard error stream.
- ignore()  $\rightarrow$  ignores the character from the stream.
- try  $\rightarrow$  handle exceptions
- throw  $\rightarrow$  throw exception
- catch  $\rightarrow$  catch exception.

- what() → used to get error messages associated with an exception.
- basic\_ios → base class for all stream.
- istream — input stream
- ostream — output stream.
- read formatted data → >>
- output " " → <<
- put() — member function to write single character
- iostream — bidirectional stream
- ifstream — input file stream
- ofstream — output file stream.
- iostream → handles the formatted & non-formatted data
- ifstream → unformatted i/o
- ofstream → formatted i/o
- eg:- read(), write(), getline(), puts(),  
cin, cout, fprintf,  
scanf, setw(),  
setprecision().
- function template :-  
That can work with multiple data type.
- defining function template keyword :- template.
- template parameter :- <> ← symbol.
- std::max → maximum of two values.
- template is declared by :- Keyword :-  
template, typename & class.

- syntax for defining function template:-

template <typename T> void functionName (T arg)

- function template specialization :- allows for the template to handle specific data type differently.

Eg:- template <> void funName <int> (int arg).

template <> void funName <double> (double arg)

- default template argument :- helps template to be more flexible.

Eg:- template <typename T = int> void funName (T arg)

↑  
main.

# function template overloading :- It is the process of defining multiple function template with same name, but different parameters types or number.

- priority order for overloading function :-

Exact match, promotion, standard conversion, user-defined conversion.

- Template is used for generic programming.

- Templates are compile time polymorphism

- Template allows the programmer to write code for all data types.

- 2 template concepts → function templates.

→ class templates.

# # Computer Architecture

Date:  
Page:

- CPU is composed of
  - Control Unit
  - ALU
  - Registers
- Control Unit :- It is responsible in fetching and decoding instructions.
- ALU :- It is arithmetic logic unit, which is used to implement arithmetic and logical information.
- Registers :- Registers are the fastest, smallest and storage device.
  - They hold the immediate results of operations.
- memory hierarchy - The organization of different types of memory by speed and capacity.
- ROM - permanently storing BIOS
- Cache - temporarily storing
- Paging unit - responsible for managing virtual mem.
- memory controller - To regulate the flow of data between the CPU and memory.
- DDR4 has higher clock speed than DDR3.
- MMU (memory management unit) is used to map virtual memory address to physical memory addresses.
- Store program that is currently being executed by the CPU - main memory.
- OS is responsible for managing virtual memory.

# AAU $\Rightarrow$ Address Arithmetic Unit

Date:  
Page:

- data is the raw material used as input and instruction is the processed data obtained as output of data processing.
- ALU + CU = CPU
- MAR - memory address register.
- The main task of control unit is to generate control signals.  
2 types of control units :-
  - Hardwired controlled unit
    - ↳ generated using combinational logic circuit
  - Microprogrammed control unit
    - ↳ generates control signal by using some softwares.
- State-table method is the simplest method of implementing hardwired control unit.
- A set of micro instructions for a single machine instruction is called micro program.
- Micro-program consists of set of micro instructions which are strings 0's and 1's.
- A decoder is required in case of a vertical microinstruction.
- MBR - memory Buffer Register.
- IR  $\rightarrow$  holds the last instruction fetched.
- D-in D-flipflop means delay.

## # Address decoding:-

- In case of 2-to-10-address instruction method the operands are stored in push down stack.
- Relative addressing mode uses PC instead of a general purpose register.
- Immediate
- Direct addressing mode where direct operand value is specified.
- Relative addressing mode is suitable to change the normal sequence of execution of instructions.
  - since it directly updates the program counter.

- A set of principles and methods that specify the functioning, organization, and implementation of computer system is known as computer architecture.
- Structure and behaviour of a computer system as observed by the user is computer organization

- Types of computer architecture are:-

- ① Von-Neumann Architecture
- ② Harvard Architecture
- ③ Instruction set Architecture
- ④ Microarchitecture
- ⑤ System Design.

- Von-Neumann is the architecture used in todays computers.

- RISC is power efficient and followed in the design of mobile devices.
- speed.  
- CSA - computer addition
- IA-32 follows → CISC design.
  - ↳ suitable for wide range of data type.
- VLIW follows → multiple instruction & multiple data
  - ↳ MIMD.
  - ↳ very long instruction word.
- In CISC architecture most of complex instructions are stored in Transistors.
- Both RISC & CISC are developed to reduce semantic gap.
  - ↳ The gap between high level language and low level language.
- The small extremely fast RAM is called cache.
- superscalar architecture has faster execution as they group sets and decoded and then executed.
- ISA - International Standard → American.
- latch were used to read multiple byte of a row at the same time.
- The SRAM's uses buffered storage of address & data.

- The chip gets enabled if the CS is set otherwise the chip gets disabled.
- EEPROM or flash memory is much similar in functionality by ~~but~~ EEPROM is much cheaper.
- during write operation if required info. is not present in cache then its cache miss. / contemiss
- The dirty bit is used to show that the block was recently modified and for a replacement algorithm.

### # Direct mapping :-

The simplest way to determine cache locations in where to store memory blocks is direct mapping.

It is a procedure used to assign each memory block in the main memory to a particular line in the cache.

### # Associative mapping:-

It is used to store memory word information and addresses. Any block can be placed in any line of the cache.

### # Set associative :-

It allows each word in the cache to have two or more word in the main memory for the same index location.

It combines the benefit of both, direct & associative mapping.

- # Direct addressing mode :- STA FFFFH.
- Indirect addressing mode :- MOV A, M.
  - Register addressing mode :- MOV A, B
  - Immediate addressing mode :- MVI A, FFH

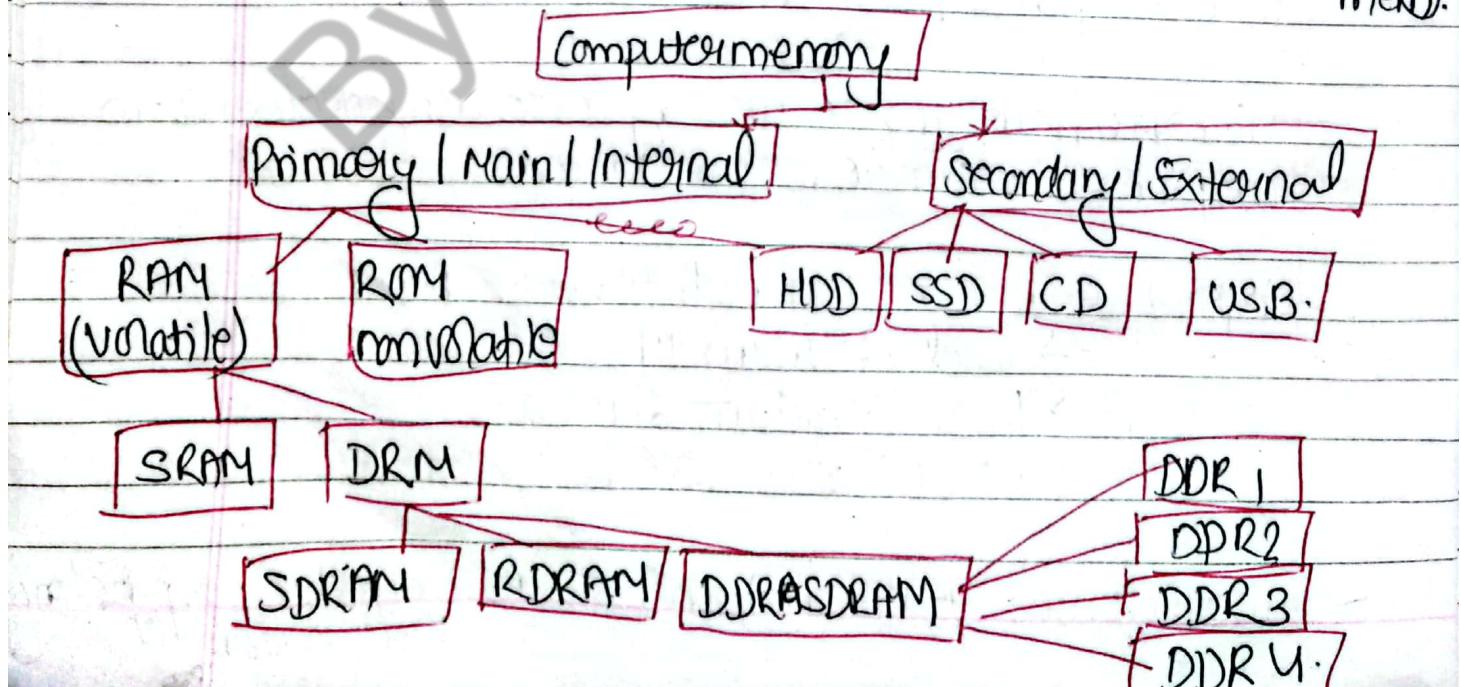
- Implicit addressing mode :- The instruction itself specifies the operands implicitly.  
e.g:- CNP, RRC, RLC, etc.
- The sun micro system processors usually follows RISC architecture.
- RISC machines iconic feature is having a branch delay slot.
- RISC and CISC are developed to reduce semantic gaps.
- Pipelining is the unique feature of RISC machine.
- RISC is power efficient.
- microprogramming is used by CISC architecture to implement complex instructions.
- hybrid of RISC & CISC architecture are :-
  - Intel platinum III
  - Intel Itanium 64
  - AMD's X86 & X64.
- The feature of RISC not in CISC is branching & pipeline prediction

- 5 stages of pipelining are:-

- ① fetch instruction from memory
- ② Read register and decode the instructions
- ③ Execute the instruction or calculate an address
- ④ Access an operand in data memory
- ⑤ Write result into a register.

- L1 Cache :- is small, high speed memory unit located within or very close to the CPU chip used to store frequently accessed data and instructions for rapid access
- L2 cache :- larger, secondary cache memory situated between L1 and main memory, serving as an intermediary storage for frequently used data to improve overall system performance.

C1 cache — L2 cache — Register ----- ; Secondary mem.

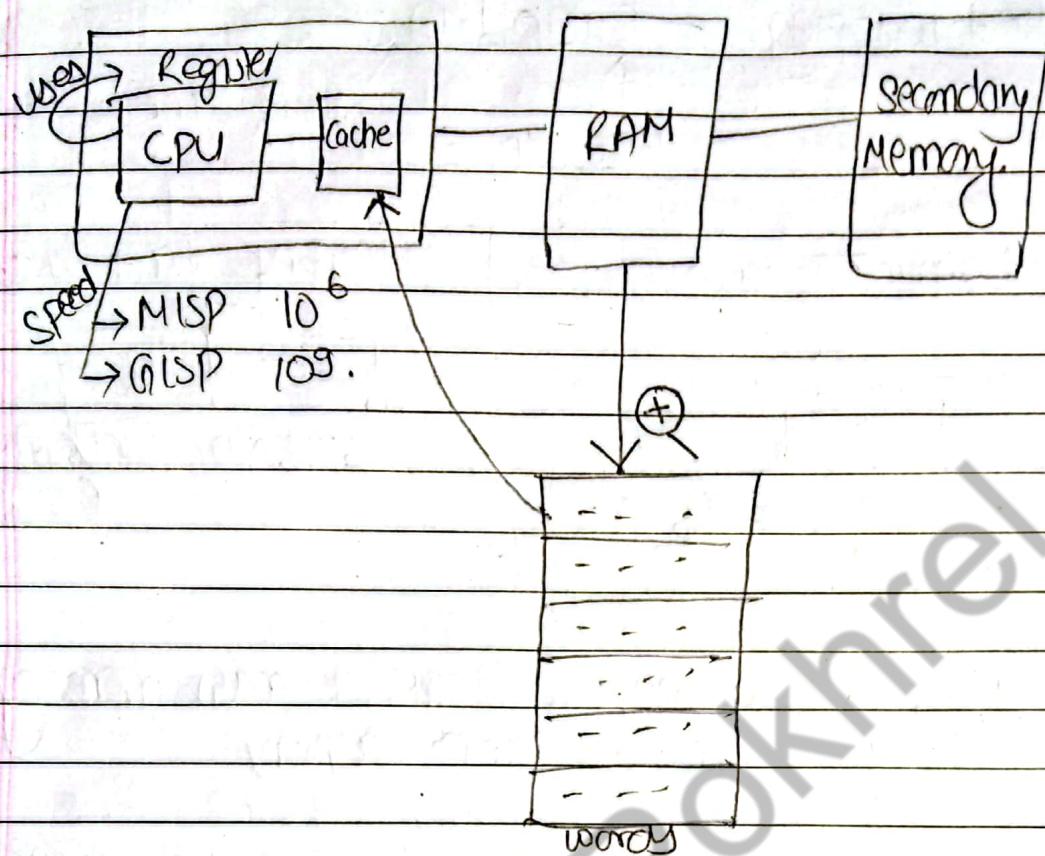


- avoid cache coherence techniques are:-
  - + write through
  - + write back
  - + buffered write
- snoopy write is efficient method of cache updating.
- memory mapped I/O
  - + same address space share
  - + simple
  - + same set of command.
  - + may reason ambiguity and ability conflicts with memory.
- I/O mapped I/O
  - + different address space
  - + complex circuit because of
  - + different set of commands.
  - + separate interrupt vector for I/Os
- The system is notified of read or write operation by sending special signals along the Bus.
- The method of accessing the I/O devices by repeatedly checking the status flag is program-controlled I/O
- higher speed of I/O transfer is DMA.
- polling is the process of constantly checking the status flag of processor.
- The DMA controller has 3 registers
  - Starting address
  - word count
  - status of operation

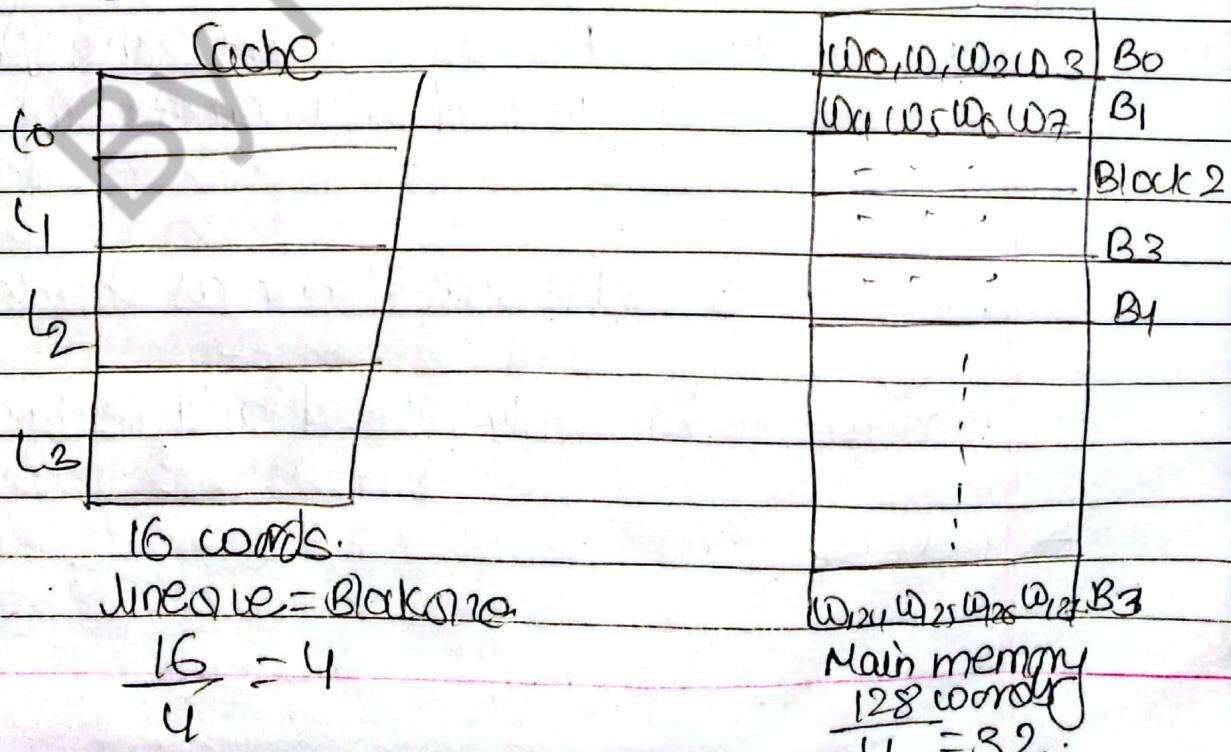
- When R/W bit of status register of DMA controller is set to 1 → Read operation is performed.
- The DMA controller is connected to System Bus to provide faster transfer.
- DMA transfer is initiated by I/O devices.
- Distributed systems are designed to run parallel process.
- Multiprocessor OS must take care of unauthorized data access and data protection.
- Memory to Register → LDA (Load).
- Register to Memory → STR (Store).
- Cache memory → found between the CPU and main memory
  - act as a buffer.
  - data found → cache hit
  - data not found → cache miss and copy from main memory to store in a cache.
- Cache Replacement policy → LRU & FIFO.
- Cache write back → To write data to the main memory everytime it is modified in the cache.

Cache Mapping:

Direct fully associative  
Random set associative



- cache divided into line  $\rightarrow$
- Ram to Cache ~~add~~ add data with reference of CPU.
- memory represented in byte, KB or word.

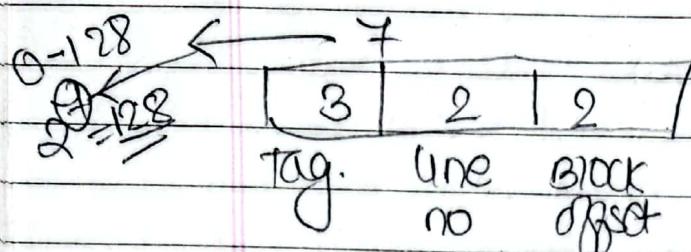


direct mapping : -  $k \bmod n$

↑  
Block No.

↑  
No of lines

Cache map bit formula:-



Memory formula:-

|          |              |          |
|----------|--------------|----------|
| 1        | 5            | 2        |
| Block No | Block offset | (contd.) |
| 7        | 4            | 3        |

- mapping function maps a block of main memory to single cache line — direct mapping
  - mapping function maps a block of main memory to multiple cache line — set associative mapping
  - mapping function allows any block of main memory to be placed in any cache line — Associative mapping
- Replacement policy:-
- LRU (Least Recently Used) : - the cache block that has not been accessed for a longest time is replaced.
  - FIFO - The cache block that was loaded first
  - Random : - Randomly selected and replaced.
  - LRU uses stack.

\* Cache write policy :-

→ write through policy -

updated both cache & memory simultaneously

→ write back policy -

updated in cache as a page bit called dirty bit. and updated data is written in main memory.

# Number of cache :-

- L1 cache
  - ↓
    - level 1. small size, low latency. 18KB-16KB.
    - most expensive to implement.
- L2 cache
  - ↓
    - level 2
    - or on a separate chip. 14-8 MB
    - larger size & high latency than L1 cache
- L3 cache
  - ↓
    - level 3
    - typically located on separate chip or mother board.
    - larger size & high latency than L2 cache

# DMA transfer data in 3 mode :-

- burst mode
- cycle stealing mode
- transparent mode:

# Interprocessor communication :-

→ shared memory - fast → may occur dead lock race condition

→ message passing - slower → due to overhead of sending & receiving message.

- synchronization is essential to avoid race conditions, etc.
  - can be achieved by locks, semaphores, barriers & atomic operations

# An embedded system is a microcontroller or microprocessor based system which is designed to perform a specific task.

for eg:- a fire alarm is a embedded system; it will only sense smoke.

- Embedded system has 3 components:-

Hardware

Application software  
RTOS.

designed using

- small scale ES - 8 bit or 16 bit microcontroller.
- medium scale ES - 16 or 32 bit "
- sophisticated or - 32 bit or 64 bit  
complex ES.

- Task - smallest unit of work that can be assigned to a processor

- Process - A process is a collection of task that share a common address space & resources.

- Thread - The lightweight unit of execution within a process.

- Multitasking & Multiprocessing :-

- Multiprocessing :- allows multiple CPUs to execute different task simultaneously

- Multitasking :- allows a single CPU to execute multiple task simultaneously.

- semaphores allows multiple thread or processes to access a shared timer resources with limited capacity
- mutex only allows one thread or process to access a shared resource at a time.

both are mechanism help to prevent race condition and ensure that shared resources are accessed in controlled and synchronized manner.

- open loop control  $\rightarrow$  non feedback control system :-  
output not compared to desired d/p
- closed loop control - feedback control system :-  
output compared to desired o/p.  
and error is used to adjust error  
or input control actions ..
- less sensitive to disturbance
- They can be unstable .

# VHDL (VHSIC Hardware description language).

(Very High speed integrated circuit.)

- Entity declaration interface to outside world that defines input and output signals.
- VHDL uses reserved keywords which can't be used as signal names or identifiers.
- keywords and user defined identifiers are case insensitive.
- The comments start with the (--) .
- VHDL ignores line break and extra spaces.

- VHDL is strongly typed language which implies that one has always to declare type of every object that can have a value, such as signals, constants and variables.

- Identifier
- Identifier rule choose :- (A-Z, a-z, 0-9) & (-).
  - first must be a letter and last one can't be underscore
  - It can't include two consecutive underscores
  - It is case insensitive
  - Can be of any length.

Signal list of signal-name : type[: initial value];

eg:- signal SUM, CARRY : std\_logic;

- operators in VHDL - 6

- logical (AND, OR, NOT ... etc)

- Relational (=, !=, <, >, ~~BOSS man~~)

- Shift - (SLL, SRL, SLA, SRA, RL, RR)

- Adding

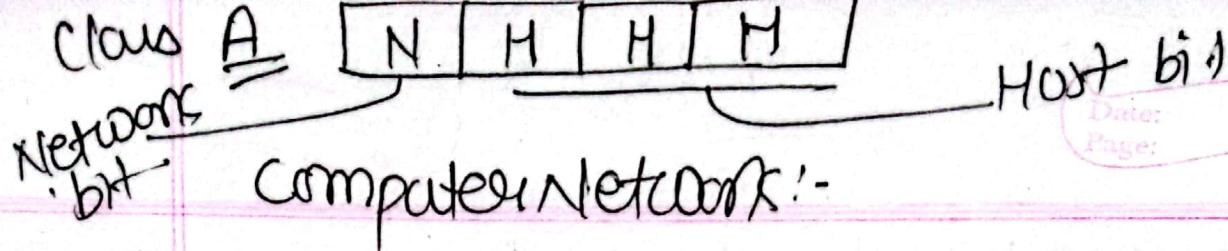
- multiplying

- Miscellaneous - abs, \*\*

(absolute) (exponent)  
value (tail).

[ SLL - shift left logical  
SLA - shift left arithmetic  
RL - rotate left ]

- std\_logic\_1164 - package defines standard data type.
- std\_logic\_arith package : provide arithmetic (on various And comparison functions for signed, unsigned, std\_logic, std\_logic and std\_logic\_vector types).
- std\_logic\_miscpackage : - types, subtypes, constant  
 → library ieee;  
 → use ieee.std\_logic\_1164.all;



Class A  $\rightarrow$  1 to 126

Class B  $\rightarrow$  128 to 191

Class C  $\rightarrow$  192 to 223

Class D  $\rightarrow$  224 to 239

Class E  $\rightarrow$  240 to 255.

| Group  | 128 | 64  | 32  | 16  | 8   | 4   | 2   | 1   |
|--------|-----|-----|-----|-----|-----|-----|-----|-----|
| Bit    | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   |
| Subnet | 128 | 192 | 224 | 240 | 248 | 252 | 254 | 255 |
|        | 125 | 127 | 128 | 129 | 130 | 131 | 132 |     |

*↑ 256 - group = subnet*

private ip range:-

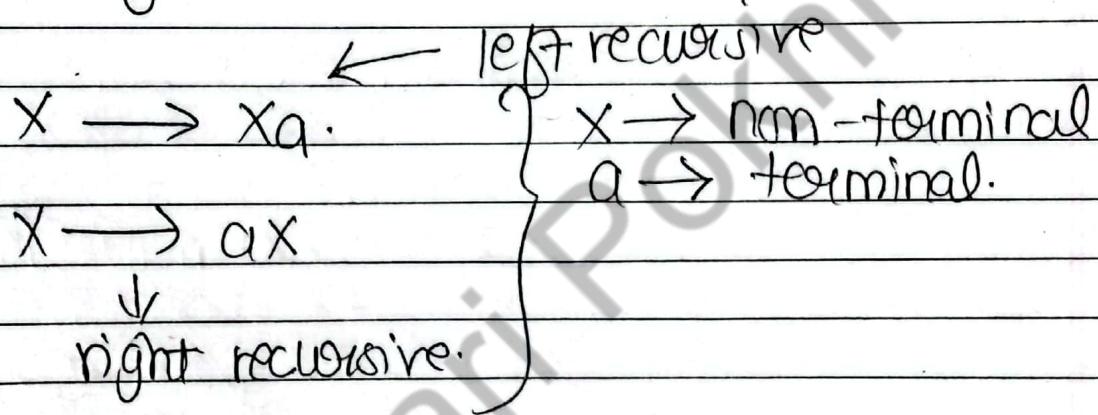
Class A :- 10.0.0.0 to 10.255.255.255

Class B :- 172.16.0.0 to 172.31.255.255

Class C :- 192.168.0.0 to 192.168.255.255.

- Ethernet (802.3)
- Token Bus (802.4)
- Token Ring (802.5).

- Regular expression to finite automata:
- Convert RE into equivalent NFA (N-DFA)
- Convert NFA into equivalent DFA.
- finite automata to Regular expression:
- Convert DFA to an equivalent NFA
- Convert NFA to an equivalent Regular E. using state elimination method.



- a grammar is ambiguous; if one that can generate more than one parse tree for a given sentence.
- CNF  $\Rightarrow$   $A \rightarrow Bc$  {Non-terminal symbol}  
 $A \rightarrow a.$  {Terminal symbol}.
- GNF  $\Rightarrow$   $Ad.$   $\{x \rightarrow \text{string of non-terminal symbols}$   
 $\text{until at least one non-terminal symbol}\}$ .

[ Pumping lemma provides the certain way to prove the given language is not a context free.  
 It checks whether a given grammar is context free or not.

- Intractability : - Refers to the property of a computational problem that cannot be solved by any algorithm. (Travelling salesman problem).
- Reducibility : - Refers to the ability of transforming one problem into another.

## Computer Networks :-

- Application layer (7) :- User interaction and application services such as email, FTP and web browsing.
- Presentation layer (6) :- Ensures data is presented in a compatible format for communication, including data encryption and compression.
- Session layer (5) :- Manages communication sessions between applications, including establishing, maintaining and terminating connections.
- Transport layer (4) :- Provides reliable and unreliable data delivery and flow control between hosts, using protocols like TCP and UDP.
- Network layer (3) :- Routes data packets between networks, forwarding packets based on logical addresses (IP addresses).
- Data Link layer (2) :- Manages access to the physical network medium, handles framing, error detection and flow control.
- Physical layer (1) :- Transmits raw bit data over the physical medium, including encoding, signaling and voltage level.

socket = port + ipaddress

Date:  
Page:

## # Networking devices:-

- Hubs :- central connection.
- Bridges - Unring segments (L2, reduce traffic & collision)
- Switches - Intelligent forwarding (L2, MAC, reduce collision)
- Router - Routing traffic (L3, IP address)

## # Persistent:-

- I-Persistent  $\rightarrow$  Always try, don't wait
- Non-Persistent  $\rightarrow$  wait and retry
- P-persistent  $\rightarrow$  probability Retry
- O-persistent  $\rightarrow$  orderly Retry.

## Economics:

- Project cash flow :- The difference between a project's total inflow and outflow.
- discount rate :- The rate of return an investor could earn by investing in an alternative project.
- Interest :- The cost of borrowing money or the return earned on invested fund.
- Time value of money :- the principle that money today is worth more than the same amount of money in future.

$$F = P(1+i)^n$$

$$P = F(1+i)^{-n}$$

- Discounted pay back period :-

$$\left[ \frac{\text{cash inflow}}{\text{discount rate}} - \text{initial cost} \right]$$

- Net present value :-

$$\text{Present value of cash inflow} - \text{Present value of cash outflow}$$

- Internal Rate of Return :-

$$\left[ \text{Discounted Rate at which NPV} = 0 \right]$$

- Minimum Acceptance Rate of Return :-

The minimum rate of return required by an investor to justify the investment in a project.

- Resource leveling :-

Adjusting the schedule to balance the workload of resources over time

- Resource smoothing :-

Adjusting the schedule without changing the overall project duration.

- PMIS - Project Management Information System :-

To provide project managers with tools for tracking project progress

- Project financing :- The process of raising funds to finance a specific project.

Drawing paper :-

$$\rightarrow A0 = 841 \times 1189 \text{ mm}$$

$$\rightarrow A1 = 594 \times 841 \text{ mm}$$

$$\rightarrow A2 = 420 \times 594 \text{ mm}$$

$$\rightarrow A3 = 297 \times 420 \text{ mm}$$

$$\rightarrow A4 = 210 \times 297 \text{ mm}$$

1: $\sqrt{2}$  ratio

ISO 2016

~~3D TO 2D~~

Orthographic projection :- Unfolded box

~~120°~~

Isometric projection :- All 3 axes are equally inclined looking at object from a corner.

Pictorial view :- Show object in 3D

Sectional view :- cut cake to see its layer

- object line
  - hidden line
  - center line
  - cutting plane
- \_\_\_\_\_  
----  
-.-.  
↓      ↓

### # CPM

- critical path method
- Activity oriented
- Predictable activities are managed
- focus of (ost) optim Path on
- 1957
- single time estimate
- deterministic model

### PERT

- Project Evaluation & Review Technique.
- Event oriented.
- Unpredictable activities are managed.
- focus on time control
- 1958
- Three time estimate
- probabilistic model.

*optimistic*

$$\text{Expected time : } t_e = \frac{(t_p + 4t_m + t_o)}{6}$$

$$\text{variance} = \left( \frac{t_p - t_o}{6} \right)^2$$

$$SD = \sqrt{V}$$

2055/11/27 → 1999 march 11

Date:  
Page:

registered Engineers till 2078/13) → 61

Total engineer till → 64,620

Total 50 eng. colleges have been approved  
3700 every year locally graduated.

2079 Bhadra 5 → licence exam date.

appoint prime minister

4 year tenure

Meeting 4 times in year by chairman  
more than 50% member attendance in  
meeting.

21 member in NEC

21 member in NEA

NEA 2 year tenure.

NEC effective from : - 2057/318.

first president of NEC : - Er. Ram Babu Sharma

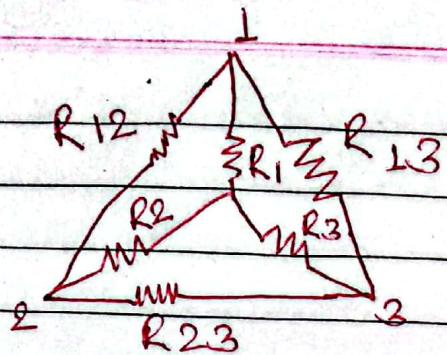
present : - padam bahadur Shahi.

its samana :- 13. P Eng.

Tadhar

Bmokshay

5/5/1



Δ to Y.

$$\rightarrow R_1 = \frac{R_{12} \times R_{13}}{R_{12} + R_{13} + R_{23}}$$

$$R_{\text{delta}} = 3 R_{\text{star}}$$

Y to Δ

$$R_{12} = R_1 + R_2 + \frac{R_1 \cdot R_2}{R_3}$$

$$R_{\text{star}} = \frac{R_{\text{delta}}}{3}$$

$\text{IRR} > \text{NARR} \Rightarrow$  project accept.  
less than ① project is rejected.

## TOC :-

- 5 elements or tuples:-

$Q$  :- finite set of states

$\Sigma$  :- set of <sup>input</sup> empty symbols

$q_0$  :- initial state

$F$  :- final states

$\delta$  :- transition function.  $\Rightarrow \delta = Q \times \Sigma$ .

$\{Q, \Sigma, q_0, F, \delta\}$ .

- Finite automata is characterized into two types.

① Deterministic finite automata (DFA)

- 5 tuples.  $\{Q, \Sigma, q_0, F, \delta\}$ .

- A DFA with minimum number is generally preferred.

② NDFA:-

- one difference is ( $\Lambda$  null or  $\epsilon$ ) more is allowed.  
It can move forward without reading symbols  
- all are same except transition function to DFA.

$\delta : Q \times (\Sigma \cup \{\epsilon\}) \rightarrow 2^F$ .

# DFA :-

- simplest model of computation

- It has very limited memory

- for a single input there is only one transition state.

- Given the current state we know what next state will be.

## finite automata

FA with output

Noise  
machine

Monly  
machine

FA without output

DFA

NFA

FNFA  
( $\epsilon$ psilon)

- Every DFA is NFA but not all NFA are DFA.

NFA :-  $\{\emptyset, \Sigma, S, q_0, F\}$

$\downarrow$   
finite no of  
internal states

$\rightarrow$  finite no of  
sum. no. called  
alphabet

~~$S \times \Sigma \rightarrow 2^S$~~

~~DNF =  $m^2$~~

DFA

- Transition function =  
 $\delta = Q \times \Sigma \rightarrow Q$

- DFA accepts if final state  
of automata is a acceptor

- There is always a new  
state for a single input

- less efficient than NFA

- It can't be multiple initial state

- Every DFA is converted to NFA

NFA

- Transition function =  
 $\delta = Q \times \Sigma \rightarrow 2^Q$

- NFA accepts only if any  
of final states is turned  
to an accept state.

- There may be a new,  
multiple state for one

- more efficient than DFA

- It can be multiple ini-

- There must exist equiv-  
alent DFA for NFA.

## # Pumping lemma for Regular expression / language

- @  $xy^iz \in A$  for every  $i \geq 0$
- (b)  $|y| > 0$  (length of  $y$ ).
- (c)  $|xy| \leq p$  (length of  $x \& y$ ).

## # properties of regular language :-

- Union
- Intersection
- Complementation
- Kleen Star
- Complementation

## # Context free grammar:-

A quadruple defined as follows:-

$$(n = (V, S, R, S))$$

$V =$  is an alphabet (set of terminals and non-terminals)

$\Sigma$  (the set of terminals) is a subset of  $V$ .

$R$  (the set of rules) is a finite subset

$$(V - \Sigma)^* V^*$$

$S$  (the start symbol) is an element.

## # Left and right most derivative:-

- If we replace the left most variable by one of its production bodies such derivation is called left most derivation
- If we replace the right most variable by one of its production bodies such derivation is called right most derivation.

## # Ambiguous grammar:-

A grammar  $G$  is said to be ambiguous if string  $SE(L(G))$  has two or more than two leftmost, rightmost derivation or parse tree for its derivation.

- inherently ambiguous language: - It is a language for which no unambiguous grammar exists.

## # Chomsky Normal form:-

- is Cf if the grammar is in the form of :-  
$$A \rightarrow BC \quad ] \rightarrow \text{Cf form.}$$
 or;  $A \rightarrow a$

any CFn that doesn't contain ' $\epsilon$ ' can be put into CNF.

procedure to find CNF:-

- a. Eliminate the useless grammar. (non-generating or non-reachable symbols).
- b. Eliminate the unit production and  $\epsilon$ -production.
- c. Eliminate the terminals on the right hand side of length two or more.
- d. Restrict the number of production to two variable.

### CNF

- production takes the form  
$$A \rightarrow BC$$
  
$$A \rightarrow a.$$

$A, B, C$  are non-terminals  
and  $a$  is terminal symbol.

### GNF

- production takes the form  
$$A \rightarrow \alpha$$
.  
 $\alpha$  is a terminal symbol  
and  $\alpha$  denotes no or many variable.

- It is not useful in proving the equivalence of NPDA and CFG.
- It has longer derivation than CNF.
- It has 2 forms
- It is useful for proving the equivalence of NPDA and CFG.
- Shorter than CNF
- It has single form.

### # PDA (push down automata):

$L(M) = \{ \dots \}$  It consists of 7 tuples.

$Q$  : a finite set of states

$\Sigma$  : a finite set of inputs

$\Gamma$  : a finite stack alphabet

$\delta$  : transition function

$q_0$  : the start state

$z_0$  : the start stack symbol (optional)

$F$  : final state / accepting state

### # PDA = CFG:

# Turing Machine also has 7 tuples.  $\{ Q, \Sigma, \Gamma, b, \delta, q_0, F \}$

$\delta : Q \times F \rightarrow Q \times \Gamma' \times \{ L, R, N \}$ .

$b$  : blank symbol.

- Non-deterministic Turing machine has 6 tuples.  
 $b$  = blank symbol is not there.

$\{ Q, \Sigma, \Gamma, \delta, q_0, F \}$

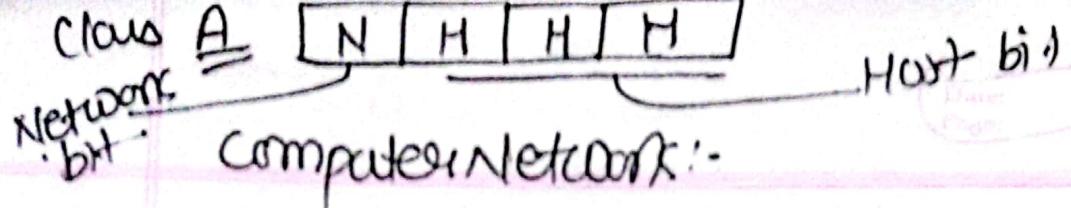
- NFA = DFA; if they accept the same language, even though they may be use different method to do so.
  - Turing machine : - Tape - cell.  
 head - read & write.  
 $\hookrightarrow \{a^n b^n \mid n \geq 0\}$ .
  - Turing machine  $\rightarrow$  regular language is recognized by deterministic Turing machine
- $\hookrightarrow$  CFN is recognized by Non-Deterministic Turing Machine
- Turing machine  $\pi$  DFS used. for searching.
  - TM with multiple tracks  $\rightarrow$  There are multiple tapes or tracks, each of which can move independently.
  - TM without multiple tapes  $\Rightarrow$  It is a theoretical model that has multiple tapes, where each tape is capable of reading, writing and moving in both directions independently.
- (Non-Deterministic TM)  $\rightarrow$  theoretical, that allows multiple paths of computation and powerful than (DTM).

UTM  $\rightarrow$  can compute with any string inputs:  
 Computational complexity  $\rightarrow$  P, NP & NP-complete.

P — The class problem that can be solved in polynomial time.

NP — The class of problem which a solution can be verified in polynomial time.

NP-complete — hardest problem in NP, any NP problem can be reduced to NP-complete problem in polynomial time.



Class A  $\rightarrow$  1 to 126

Class B  $\rightarrow$  128 to 191

Class C  $\rightarrow$  192 to 223

Class D  $\rightarrow$  224 to 239

Class E  $\rightarrow$  240 to 255

|        |                             |     |     |     |     |     |     |     |
|--------|-----------------------------|-----|-----|-----|-----|-----|-----|-----|
| Group  | 128                         | 64  | 32  | 16  | 8   | 4   | 2   | 1   |
| Bit    | 1                           | 1   | 1   | 1   | 1   | 1   | 1   | 1   |
| Subnet | 128                         | 192 | 224 | 240 | 248 | 252 | 254 | 255 |
|        | ↑ /25<br>256-group = subnet | 127 | 128 | 129 | 130 | 131 | 132 |     |

private ip range :-

Class A :- 10.0.0.0 to 10.255.255.255

Class B :- 172.16.0.0 to 172.31.255.255

Class C :- 192.168.0.0 to 192.168.255.255

- Ethernet (802.3)
- Token Bus (802.4)
- Token Ring (802.5).

- Regular expression to finite automata:
- convert RE into equivalent NFA (N-DFA)
- Convert NFA into equivalent DFA.
- finite automata to Regular expression:
- convert DFA to an equivalent NFA
- convert NFA to an equivalent Regular E. using state elimination method.

$$\begin{array}{l}
 X \rightarrow Xa. \quad \left. \begin{array}{l} X \rightarrow \text{non-terminal} \\ a \rightarrow \text{terminal} \end{array} \right\} \text{left recursive} \\
 X \rightarrow aX. \quad \left. \begin{array}{l} X \rightarrow \text{non-terminal} \\ a \rightarrow \text{terminal} \end{array} \right\} \text{right recursive}
 \end{array}$$

- a grammar is ambiguous; if one that can generate more than one parse tree for a given sentence.
- CNF  $\Rightarrow$   $A \rightarrow BC$  (Non-terminal symbol)  
 $A \rightarrow a$ . (Terminal symbol).
- GNF  $\Rightarrow$   $Ad.$   $\left. \begin{array}{l} \alpha \rightarrow \text{string of non-terminal symbols} \\ \text{with at least one non-terminal symbol.} \end{array} \right\}$

L Pumping lemma provides the certain way to prove the given language is not a context free.  
 It checks whether a given grammar is context free or not.

- Intractability: - Refers to the property of a computational problem that cannot be solved by any algorithm. (Travelling salesman problem).
- Reducibility: - Refers to the ability of transforming one problem into another.

## computer graphics:-

① Translation :-  $P' = P + T$ .

st. line

$$x' = x + tx, y' = y + ty.$$

② Rotation :-  $R' = R_0 \cdot P$ .

circle

$$x' = x(\cos\theta - y\sin\theta), y' = x\sin\theta + y\cos\theta.$$

③ Scaling :-  $Sx = Sy$  (uniform)  $Sx \neq Sy$  (ununiform).

changing size

increased size

$$Sx, Sy > 1$$

$Sx, Sy$  must be +ve.

decreased size

$$Sx, Sy < 1 \text{ by } (> 0)$$

$$x' = Sx \cdot x + 0 \cdot y$$

$$y' = Sy \cdot y + 0 \cdot x.$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} Sx & 0 \\ 0 & Sy \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}.$$

④ Reflection :-

about x-axis

mirroring

$$x' = x$$

$$y' = -y.$$

about y-axis

$$x' = -x$$

$$y' = y.$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

$$P' = M_{ref/x} \cdot P$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

$$P' = M_{ref/y} \cdot P$$

about xy axis:-

$$x' = -x$$

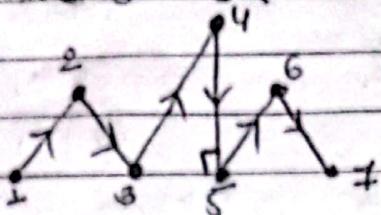
$$y' = -y.$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

$$P' = M_{ref/(x,y)} \cdot P$$

## # Tree traversal :-

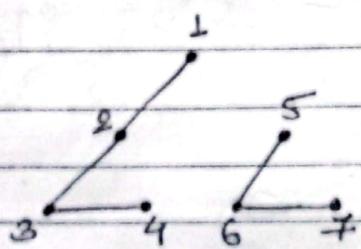
- Inorder traversal



→ left leaf node \*

- middle BT root node.

- Pre-order traversal.

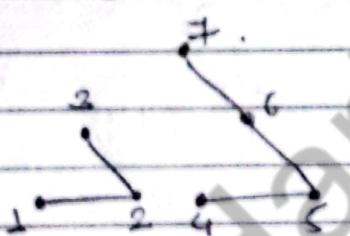


- first BT root node

- left child BT root /

- Root node - left subtree  
right subtree.

- Post order traversal:



- last BT root node

← pattern follow.

- left subtree - right subtree-  
root node.

## # Insertion and deletion in BinaryTree.

- Insertion:-

insert from given left to right

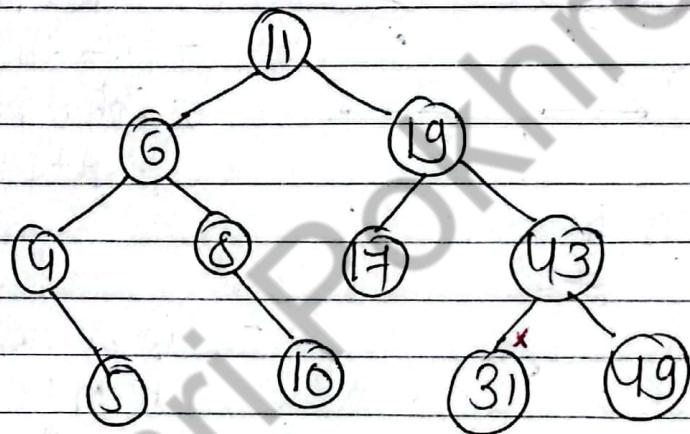
compare each value with root node and

less go to left , more go to right similar for  
other level as well

- deletion:- That has 0 child - simply delete/unlink  
That has 1 child - delete and link to  
child.

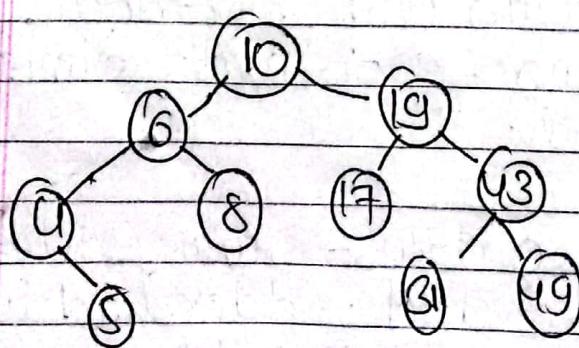
- That has 2 child - predecessor and successor.
    - replace with the most greater value in the left subtree.
    - replace the node with smallest value from right subtree
  - data will always be in ascending order.

Eg: 11, 6, 8, 19, 4, 10, 5, 17, 43, 49, 31.

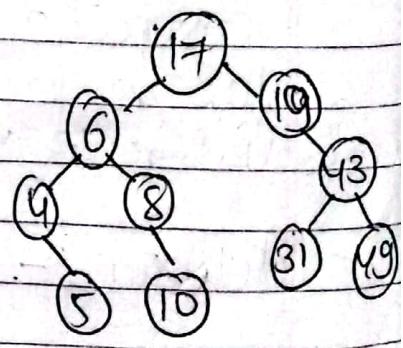


- delete 31 → that has 0 child delete
  - delete 04 → that has 1 child, delete and replace 4 with 5.
  - delete 11 → that has 2 child.

Preorder predecessor



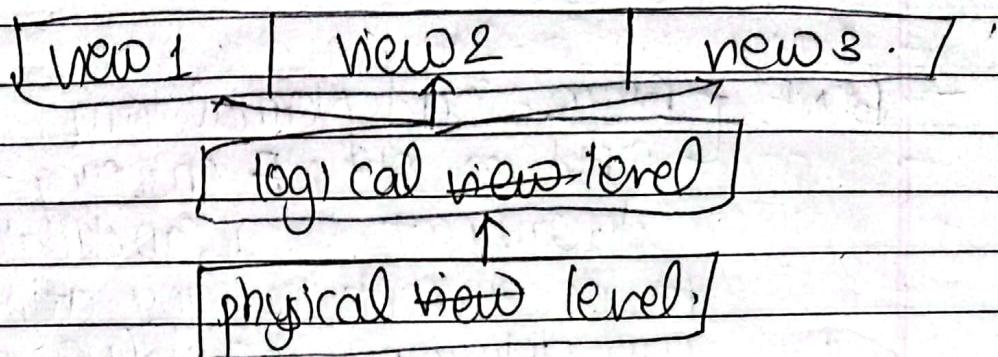
successor.



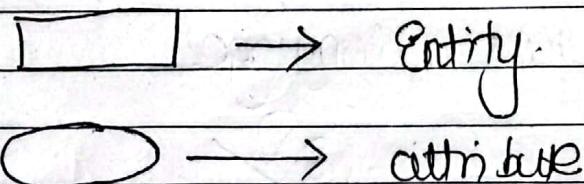
Trick: 4, 5, 6, 8, 10, 11, 14, 19, 31, 43, 49. → sorted.  
decreasing successor

## DBMS:-

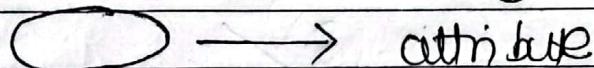
Data abstraction and data independence.



ER diagram



Relation among entity.



Link attribute and entity set relationship.

- strong entity → not dependent to other
- weak entity → dependent to other.
- foreign key :- attribute or set of attributes that must have values from primary key of the strong entity on which the weak entity depends.

# Normalization:

- 1NF : [ no repeating group or values arrays  
[ contains only atomic value  
[ indivisible value.

- 2NF : [ each non-key is fully dependent on primary key  
[ if it's 1NF and all non key are fully dependent on primary key.

- 3NF - If it is 2<sup>nd</sup> NF and has no transitive dependencies
- BCNF [Boyce-Codd Normal form]
  - Build on 3<sup>rd</sup> NF ensuring that every determinant is a candidate key
  - A determinant is any attribute that can determine the value of another attribute in same table.

DDL → Data definition language

- CREATE
- ALTER
- DROP

DML - Data Manipulation Language :-

- INSERT
- UPDATE
- DELETE
- SELECT

- View - are used to simplify complex queries
- Assertions - are used to validate data used for different purpose against predefined rules
- Triggers are used to automate database tasks and enforce data integrity.
- A - Atomicity - एक सूची के लिए पूरा होना चाहिए
- C - Consistency - consistent before & after
- I - Isolation → multiple transaction occurs independently
- D - Durability - success even in failure without interference in system.

- serializability

↳ Locking :- lock the db objects such as tablets or rows to prevent other transaction accessing

TimeStamping :- assigning unique timestamping to each transaction as it is executed.

## # Operating System

### Type:-

- Batch operating OS
- Time sharing OS
- Distributed OS
- Network OS.
- Real-time OS.

## # OOSE :-

Software process is collection of:

- Activities (budget, complexity, )
- Action (model, design)
- Task (coding, testing).

### ① Agile model:-

- flexible
- Advanced
- parallel processing dividing into sprints.

### ② Iterative model:-

- Basic
- problem is well understood

### ③ Prototype model -

- user requirement not clear
- Not costly
- NO early lock on Requirements
- High user involvement
- Reusability



- Agile model :- (Adaptability)  
↳ flexibility and the ability to adapt to changing requirements through out the development process
- V-model :- (V) - Verification & Validation.  
{ verification in phases }  
{ validation in testing }  
Sequential approach where each phase corresponding to the testing phase.
- Generative model (Repetition)  
- development occurs in cycle or iteration with each iteration building upon the previous one, allowing for continual refinement and improvement.
- Prototype model :- (Preview):  
- creating an early simplified version of software to gather feedback and refine requirement before proceeding with full scale development.
- Big Bang model :- (Bang)  
- starting development abruptly without planning or documentation often characterized by a sudden burst of activity.

- function and non function.

- defined by user / client that should be there in program. (use case).  
(end to end, API testing).
- defined by the technical people.
  - define quality attribute of project
  - (stress, performance, usability, security testing)

# Design process :-

- Interface design :- "User focus"
- Architectural design :- visualize blueprints.
- Detail design :- fine tuning.  
(include algorithms, data structure and modules).

# Design heuristic - golden ratio :-  $1:1.618$

# Pareto principle :- 80% effect comes from 20% cause

# software testing → unit testing



Integration Testing

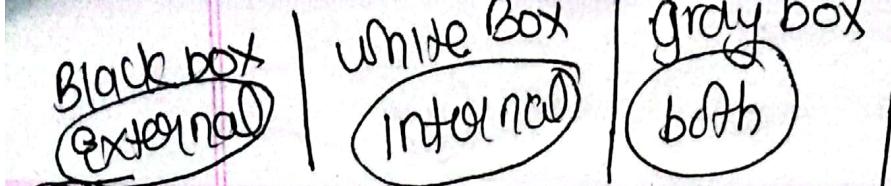


System Testing



Acceptance Testing.

- Unit Testing :- Manual and Automated - (Type)  
Technique
  - Black Box
  - White Box
  - Gray Box.



Date:  
Page:

- Q18** - Black Box :- Unit test for inputs, UT and I/O pair  
 Closed box testing.
- White Box :- testing functional behaviour of the system by giving I/O & checking functional O/P including the internal design structure and code module
  - Gray Box :- Test case, test module and analyzing code performance.

## # Integration testing:-

- Big Bang IT :- sudden, whole module testing.
- Bottom Up - down to up.
- Top-Bottom - higher module first then lower.
- Mixed Integration - both Bottom up & Top-down

## # System Testing :- tests all the system module

### # Acceptance testing:-

- User acceptance testing (UAT) - End user, ATC & E
- Business AT
- Contract AT
- Regulation AT
- Operational AT

## constructive cost Model.

- coromo :- used for estimating software development effort and cost based on project size and complexity.
- functional point analysis:- Measures the size of software deliverables based on user requirements.
- Expert Judgment :- Relies on the expertise of experienced professionals for estimation.
- ISO 9001 → Quality Management.
- ISO 29119 → Software Testing.
- ISO 19761 → cost estimation.
- ISO 12207 → Configuration Management.

## #(CMNI) :- Capability Maturity Model Integration.

| CMNI Maturity level          | CMNI capability level    |
|------------------------------|--------------------------|
| Maturity 1 :- initial        | C0 :- Incomplete         |
| M2 :- Managed                | C1 :- Performed          |
| M3 :- Defined                | C2 :- Managed            |
| M4 :- Quantitatively managed | C3 :- Defined            |
| M5 :- Optimizing             | C4 :- Quantitatively Man |
| [IM DQO]                     | C5 :- Optimizing.        |

# CASE :- Computer Aided Software Engineering

Date:

Page:

Planning

Analysis

Design

Implementation

Testing

Maintenance

Upper Case

Integrated  
CASE

lower case

# diagram

→ Structural diagram (static aspect)

→ Behavior diagram (dynamic aspect)

A1:-

Properties:

- Autonomy : - operate independently without human
- Adaptability - able to adapt in new situation and learn from the experience.
- Reactivity : - able to respond in real-time to change environment
- Proactivity : - the ability to take initiative and act before being instructed
- Communicate : - able to communicate with other agent or human.
- Rationality : - make decisions that maximize the chances of achieving the agent objective.

# PEAS

- Performance Measures
- Environment
  - { fully & partially observable environment
  - { static & dynamic
  - { deterministic & stochastic
  - { episodic & sequential
  - { discrete & continuous

- Actuators.
- Sensors.

## # Types of agents :-

### ① Simple Reflex agent :- (Reacting quickly)

- Reacts to its environment quickly based solely on the current percept without maintaining any internal state or model of the world.
- It focus on quickly selecting actions based on simple condition-action rules.

### ② Model-Based Agent - (Building a Model)

- It maintains an internal model of the world or environment, allowing it to anticipate the consequences of its action.
- It uses this model to make decisions by considering possible future state and outcomes.

### ③ Goal Based Agent - (Aiming for goals) :-

A goal Based agent operates by setting and pursuing specific goals or objectives. It evaluates action based on their potential to bring the system closer to achieve its goal. Considering factor such as feasibility, desirability and optimality.

### ④ Utility-Based Agent - (Maximum Utility)

- It evaluates action based on their expected utility or value, considering not only the achievement of goals but also other factors such as resources, risk, etc. aiming to maximize overall utility or satisfaction while selecting actions.

Time, space, complete, optimal.

- uninformed algorithms:-

- Depth first search  $O(nm)$   $O(bm)$  ✓ ✗
- Breadth first search  $O(bd)$   $O(bd)$  ✓ ✓
- Depth limited search  $O(bl)$   $O(bxl)$  ✓ ✗
- Iterative deepening search  $O(bd)$   $O(bd)$ , ✓ ✓
- Bidirectional search.  $O(bd)$   $O(bd)$  ✓ ✓

- Informed search

- greedy BFS - short term gain

(path cost & heuristic value).

- A\* - optimal path  
uploaded progress.

- Hill Climbing  $\rightarrow$  total cost = (actual cost + estimated cost)

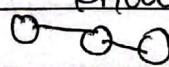
- Simulated annealing - similar to hill climbing but get stuck in local minima.

- minimax-gam  $\rightarrow$  it is DFS & DLS.

-  $\alpha$ - $\beta$  pruning algorithm  $\rightarrow$   $\alpha$  - minimum  
 $\rightarrow$   $\beta$  - maximum.

- mini-max search  $\rightarrow$  backtracking.

# logic-based approach  $\rightarrow$  first order logic to represent knowledge in formal way

- Semantic Network  $\rightarrow$  

- frames  $\rightarrow$  Semantic Networks + attributes & value associated with each node.

- Rule Based system - work with set of rules

- Neural Network - pattern learning & used where rules are not defined.

- Proposition logic :- deals with True & False  
AND  $\wedge$

OR  $\vee$

NOT  $\neg$

Implies  $\rightarrow$

If and only if  $\leftrightarrow$

- Tautology :- always true

- Universal quantification ( $\forall$ )

- Existential quantification ( $\exists$ )

# Bayes Theorem:-

$$P(A/B) = \frac{P(B/A) \cdot P(A)}{P(B)} \text{ where } P(B) \neq 0.$$

- Bayesian Network :- a probabilistic graphical model

- Knowledge acquisition - gathering, structuring & integrating knowledge from various source in order to create a KB.

- procedural knowledge

- Interpretive knowledge

- generally not used

- process oriented

- Declarative knowledge  $\rightarrow$  descriptive knowledge

- basic knowledge abt. something

- data oriented

# ARTIFICIAL INTELIGENT

All robots in europe understands Robotic technique.

Knowledge Acquisition — Representation — Inference engine

Knowledge refinement — UI — Explanation facility

→ Testing & validation.

# Steps in NLP:-

① Lexical analysis :- dissecting a sentence into basic building blocks, such as words and punctuation marks.

② Syntactic Analysis :- diagramming a sentence to understand its grammatical structure, such as identifying nouns, verbs and modifiers.

③ Semantic Analysis :- focus on understanding the meaning of sentences by analyzing the relationships between words and interpreting the context in which they are used.

④ Discourse Integration :- connecting the dots between sentences to understand overall message or theme of a text.

⑤ Pragmatic Analysis :- Reading between the lines to understand the speakers intentions and the implicit meaning behind the words.

# Artificial Neural Network :- Type of neural network that is based on feed-forward strategy.

- Biological Neural Network :- Structure that consists of synapse, dendrites, cell body and axon.

| ANN               | BNN                       |
|-------------------|---------------------------|
| - Stable          | - Unstable                |
| - Input driven    | - self organizing         |
| - Task oriented   | - goal oriented           |
| - Nested circuits | - micro-macro interactive |
| - Computational   | - Dynamic                 |
| - bits            | - flow                    |
| - symbols         | - patterns                |
| - information     | - meaning                 |

- Mathematical model of ANN.

- input layer
- hidden layers
- output layers
- Activation function -  $\Theta$
- weights
- Bias
- loss function

- Sigmoid function  $\Rightarrow (0, 1)$

- ReLU (Rectified Linear Unit) function  $\Rightarrow 0$  for negative i/p and i/p value for +ve i/p

- tanh (hyperbolic) function  $\Rightarrow (-1, 1)$

- softmax - probability of each class

#

## feed forward Neural Network.

- only in 1 direction.
- no feedback loop or cycle.
- suitable for pattern recognition, classifying and regression.
- Recurrent NN :-
  - both direction
  - cyclic manner
  - loop, feed back
  - NLP and speech recognition.
- CNN — particularly suited for processing image data.

#

## delta rule

- Input vector
- weight vector
- o/p vector.

- Hebb's law :- two neurons fire simultaneously
- Adaline :- single linear unit is called Adaline

#

## Hopfield ANN :-

$$w_{ij} = x_i * x_j$$

one o/p to other i/p.

If i/p to a neuron is +ve, the neuron o/p is +1  
If i/p to a neuron is -ve; the " " " -1

#

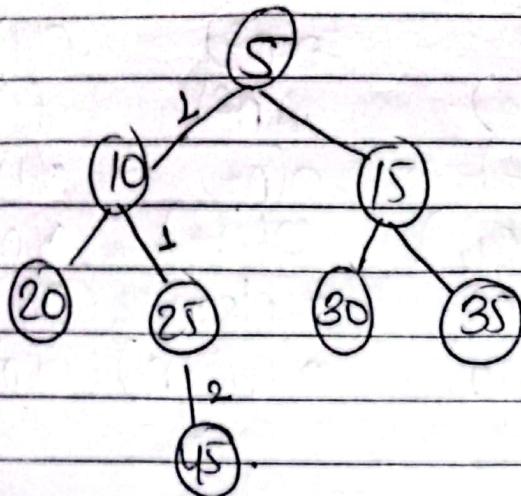
## An offspring of horse is a horse.

$$\forall x, y \text{ Horse}(x) \wedge \text{Offspring}(x, y) \Rightarrow \text{Horse}(y).$$

$$H = D + L - 1$$

any longest

- height  $\rightarrow$  distance from root to node.
- level  $\rightarrow$  root node to that node ?
- depth  $\rightarrow$  min node to that node } distance.



Binary Tree



Depth of node 10 = 1.  
height of node 10 = 2.

If depth and level given height can be calculated

$$\text{Height} = \text{depth} + \text{level} - 1$$

OR.

- Binary tree at most 2 children = {0, 1, 2}.

AVL:

Balance factor :- height of left ST - height of RST.

If Result = { -1, 0, +1 }. then its  
Okay else unbalanced

RR

L Rotation  
anticlockwise

LL

LR Rotation  
clockwise

RL

RL  $\rightarrow$  RR  
2 rotation  
RL  $\rightarrow$  RR  $\rightarrow$  L  
and 1.

LR

LR  $\rightarrow$  LL  
2 rotation.

Time complexity :: search

|           | Best case     | Average       | Worst         | Space  | Stable |
|-----------|---------------|---------------|---------------|--------|--------|
| Linear    | $O(1)$        | $O(n)$        | $O(n)$        |        |        |
| Binary    | $O(1)$        | $O(\log n)$   | $O(\log n)$   |        |        |
| Inception | $O(n)$        | $O(n^2)$      | $O(n^2)$      | $O(1)$ | Yes    |
| selection | $O(n^2)$      | $O(n^2)$      | $O(n^2)$      | $O(1)$ | No     |
| Bubble    | $O(n^2)$      | $O(n^2)$      | $O(n^2)$      | $O(1)$ | Yes    |
| Quick     | $O(n \log n)$ | $O(n \log n)$ | $O(n^2)$      | $O(n)$ | No     |
| Heap      | $O(n \log n)$ | $O(n \log n)$ | $O(n \log n)$ | $O(1)$ | No     |
| Merge     | $O(n \log n)$ | $O(n \log n)$ | $O(n \log n)$ | $O(n)$ | Yes    |

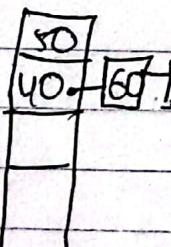
- hash mode function =  $[h(k) = k \bmod m]$

- Collision Technique in Hashing-

↓ separate hashing (open hashing)

↓ open addressing (closed hashing)

Time & space  $O(n)$  ← worst case.

| Linear probing                                                                                        | Quadratic probing                 | Double hashing                                                               |
|-------------------------------------------------------------------------------------------------------|-----------------------------------|------------------------------------------------------------------------------|
| <br>Next available | $h(k) + i^2 \bmod m$<br>↓ (index) | $h_1(k) = k \bmod m$<br>$h_2(k) = 8 - (k \bmod 8)$<br>$h_1 + h_2(k) \bmod m$ |
| $(h(k) + i) \bmod m$                                                                                  |                                   | ↓<br>If 1 fails<br>then with<br>works.                                       |