1.) Programmed - IO :-

: In this mode IO devices are directly connected to a CPU without using a IO-interface chip so CPU takes the responsibility of an IO operation.

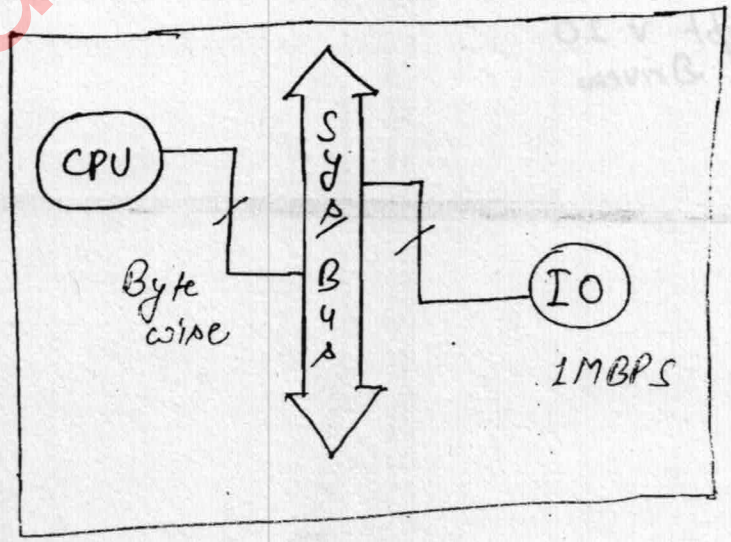
< In this mode CPU undergoes waiting state until the IO-operation is completed so processor utilization is inefficient.

★ In this mode CPU-time is directly depends on the I/O speed and the Data-size.

★ This mode is suitable in the System-centric Applications to satisfy the System-goals.

Eg: Banking System
ATC (Air-traffic controller

eg: Consider 1MBPS IO-device interfaced to CPU in a Programmed-IO mode :



ET. Prog-IO :

1MB — 1sec

1B — ?

$\Rightarrow \frac{1B}{1MB}$ sec

$\Rightarrow$ 1 μsec.

2.) "Interrupt - driven IO" :-

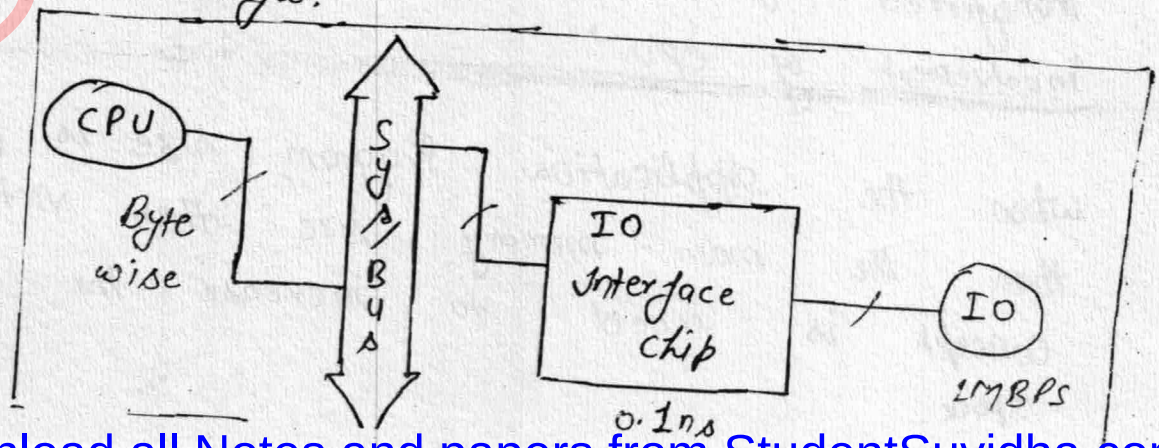
★ In this mode IO operations are controlled based on the interrupt signals.

★ In this mode IO - Devices are connected to a CPU via IO - interface chip

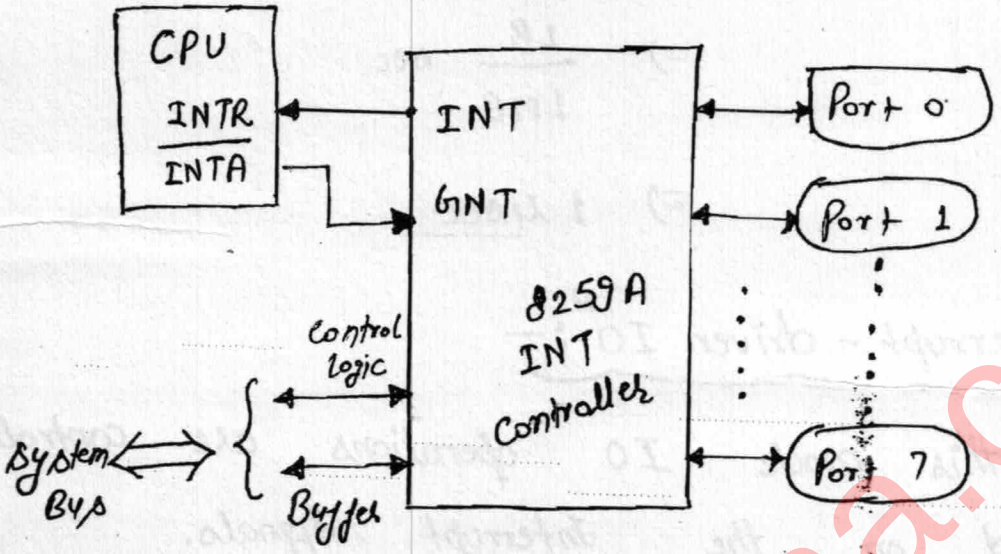
★ In this mode Processor utilization is Efficient because IO - interface takes the responsbility of IO - operations

★ In this mode CPU time is depends on the latency of a interface rather than the speed of a Device.

★ This mode is suitable with the personal computer design.



$ET_{INT-IO} = 0.1np$



* It can be used in "Single Buffer" or "Cascading Buffer" mode
* In Single Buffer It can manage 8 IO devices and in cascading Buffer It can manage upto 8 slave (8259A) i.e. 64 IO devices.
* It is a programmable chip.

"DMA (Direct M/y Access)" :-

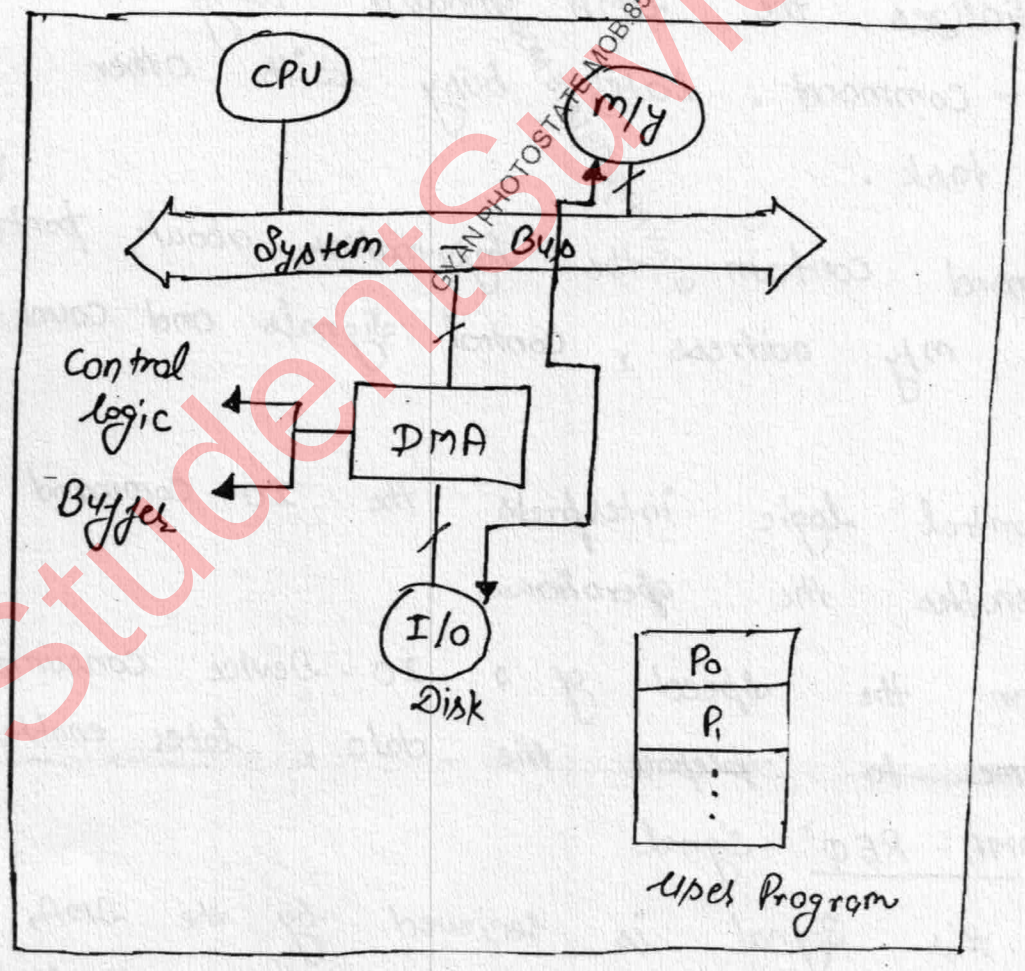
* In this mode Bulk Amount of Data will be transferred from I/O to main memory "without involvement of CPU."

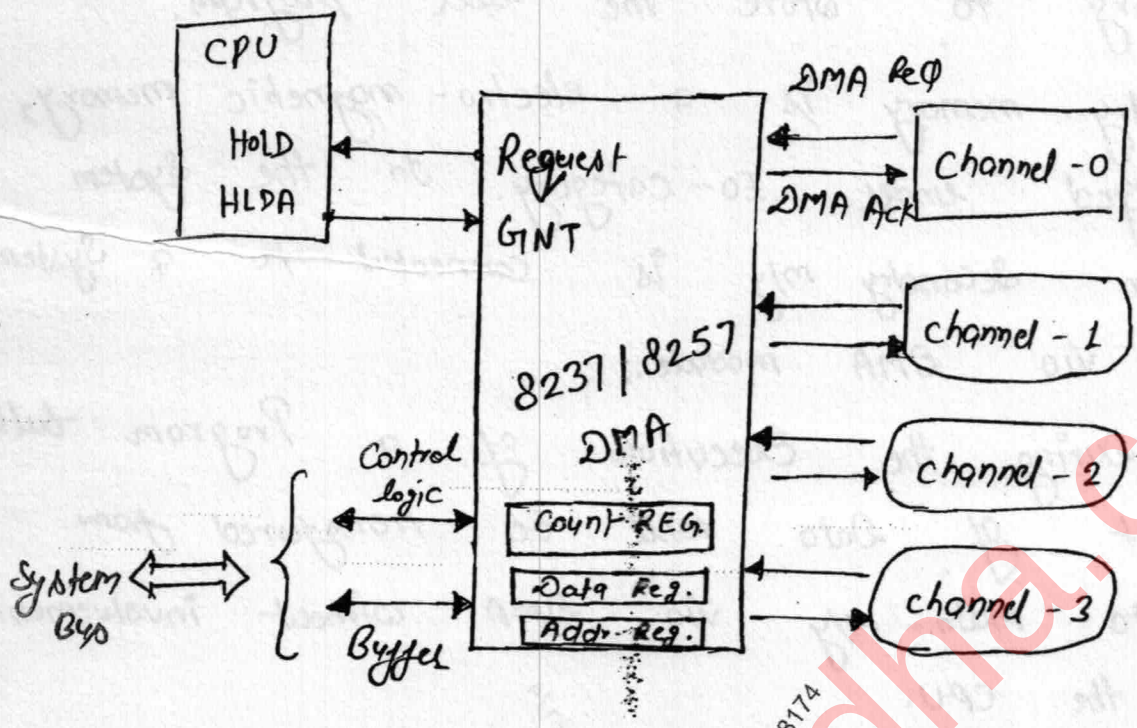
When the Application Program size is greater than the main-memory size then virtual m/y concept is used to increase the Address space.

★ This concept states that use the secondry memory to store the user program.

★ Secondry memory is a electro-magnetic memory, employed under Io-category. In the system design secondry m/y is connected to a System Bus via DMA module.

:- So during the Execution of a Program bulk amount of Data will be transferred from Io to main-m/y via DMA without involvement of the CPU.





★ CPU intializes the DMA module along with a IO - Commond . Later busy with other useful task .

★ IO - command contain the information about port address, m/y address , control signals and count - value.

★ DMA control logic interprets the IO - Command and enables the operation.

★ Based on the speed of a IO - Device consume the time to prepare the data , later enables the 'DMA - REQ' signal.

★ When this signal is recieved by the DMA, It enables the Hold signal to gain the control of a System Bus and waiting for HLDA signal.

* After Recieving the HLDA Signal, DMA module send the 'DMA - Ack' Signal to IO.

* After Recieving this Signal, IO transfers the Data to memory via DMA until count becomes '0'.

* After the DMA operation, Bus connection will be re-established to CPU.

* Note :-

:- In the DMA operation, CPU present in 2- states :

1) Busy state
2.) Blocked (Hold) State.

* CPU is in Busy state until IO- Device prepares the Data.

* CPU is in Blocked state until Transfer the data to main - memory.

eg :- Let x is a Preparation time
y is a transfer time

$$\%\ \text{time CPU Busy} = \left(\frac{x}{x+y}\right) \times 100$$

$$\%\ \text{time CPU Blocked} = \left(\frac{y}{x+y}\right) \times 100$$

:- DMA operates in 3 - modes :-

1) Burst mode
2) Cycle stealing mode
3) Block mode.

1.) "Burst mode" :-

In a Burst mode of a DMA Bulk amount of data will be transferred to main m/y baped on the Hold and HLDA signal.

2.) "cycle Stealing mode" :-

In a cycle stealing mode of DMA, small amount of Data will be transferred to main m/y by stealing the control of the System Bus. when the CPU enters into a ideal state.

3.) "Blocked mode" :-

In Block mode of DMA, Data will be transferred to main - m/y Block-wise based on the Hold & HLDA signal.

Que:- Consider 10MBPS IO Device interfaced to a CPU in a Programmed - IO mode. Data Transmission b/w the CPU and I/O is in word - wise. word length of the CPU is 64 bit. Interrupt Interface overhead is 2 ns. How much performance will be gain by using [illegible] over Programmed IO

Soln :- word length = 64 bit = 8 Bytes.

:- Programmed I/O.

10MBPS :- $\frac{1}{10 \times 10^{+6}}$ = 0.1 μsec

1 B = 0.1 μsec

8 Bytes = 0.1 × 8 = 0.8 μsec.

= 800 ns

:- Interrupt I/O :

CPU time depends on interface Latency.

$E_{interface}$ = 2ns

$S = \frac{E_T \text{ Programmed}}{E_T \text{ Int}}$

$\frac{800}{2}$ = 400

S = 400

* Que :- Consider A 32 bit Hypothetical Processor which is interfaced to a 10MBPS IO device in a cycle stealing mode of DMA. whenever 32 words are present in the Interface Buffer then it is transferred to main m/y. machine cycle takes 1 μs. How much % of CPU time is consumed in the DMA operation.

word length = 32 bit = 4B

10MBPS data so $\Rightarrow \frac{10MB}{4B}$ =

Soln :- * Preparation time : (x)

Depends on the IO speed & Data size.

- Data size = 32 words
  = 32 X 32bit
  = 32 x 4B
  = 128B

10 MB — 1sec

128 B — ?.

$\Rightarrow \frac{128\ B}{10MB}$ sec

$\Rightarrow$ 12.8 usec

X = 12.8 usec

* Transfer time (y) : Depends on the m/y speed.

1 machine cycle — To read/write 1 word Data

- Total time required to transfer 32 words :
  - Data = 32 words
    = 32 * 1 usec
    = 32 usec

% time CPU Blocked $= \frac{y}{x+y} \times 100 = \frac{32}{32+12.8} \times 100 \Rightarrow 71.42$

Que :- A 64 bit Hypothetical Processor interfaced with a 10 kBPs I/O device in a DMA ~~Cycle stealing~~ mode of DMA. DMA module contain 8 bit count Register and 64 bit Data Register. How many Bytes of Data will be transferred to main m/y in a DMA operation..

Soln :- ~~Cycle stealin~~ DMA mode :-

:- DMA operation controlled by 'count' Register

count Register size = 8 bit

As it manages 2^8 counts.

:- In every count 1 word data will be transferred.

therefore total data size : = 256 words

= 256 x 64 bit

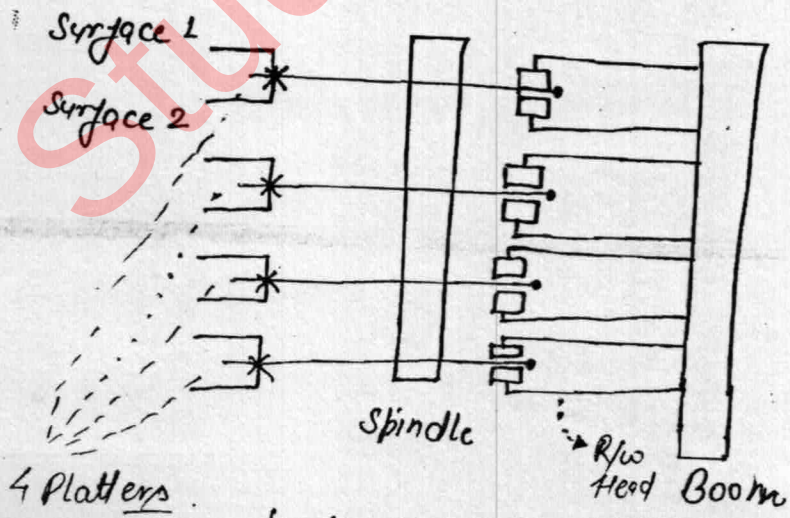
= 256 x 8B

= 2^{11} B

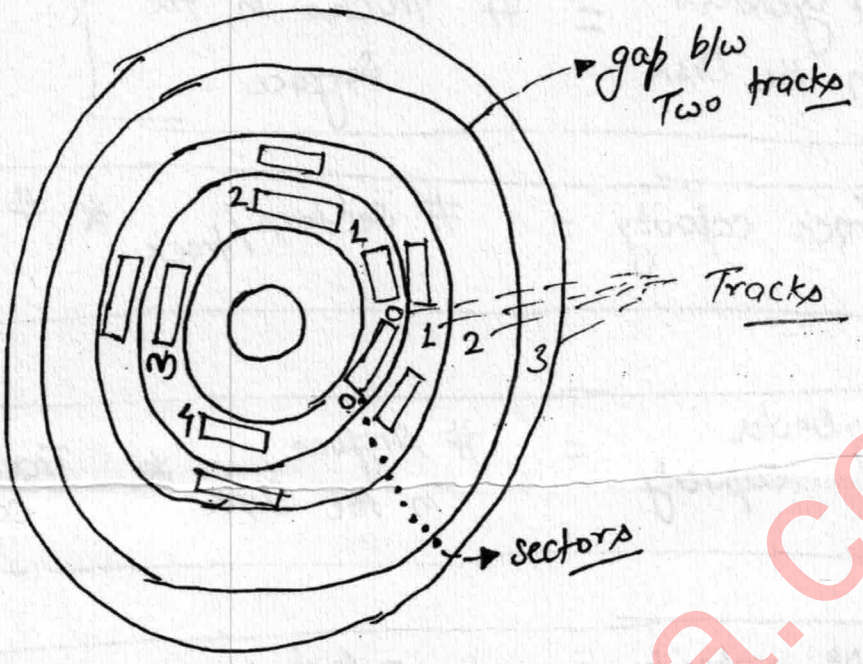
"Hard- Disk Structure" :-

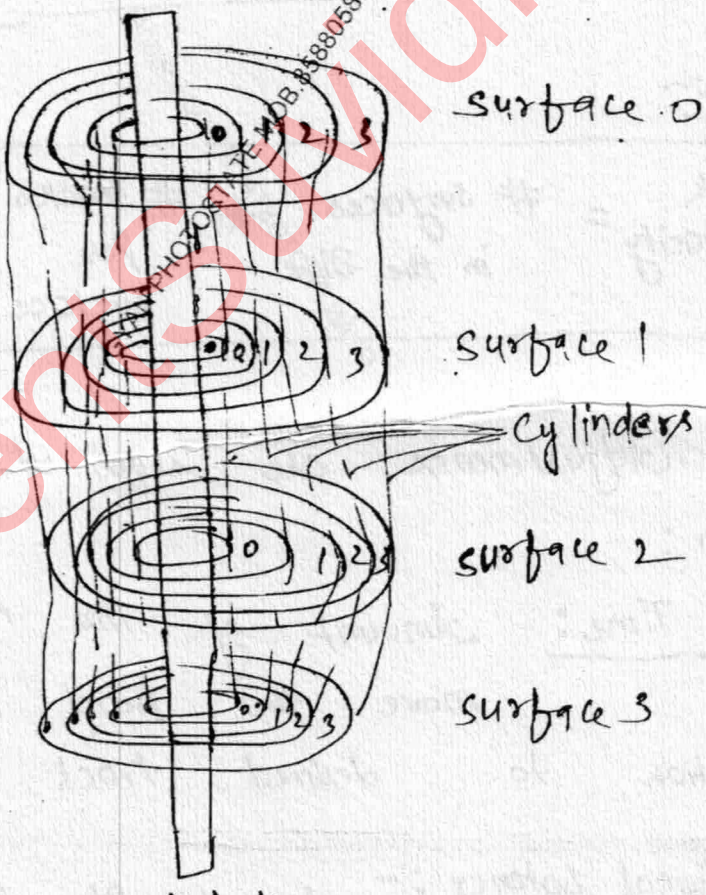
- Hard- Disk is a Direct- Access electro- magnetic storage component. It contain a bunch of magnetic coated platters used to store the data.

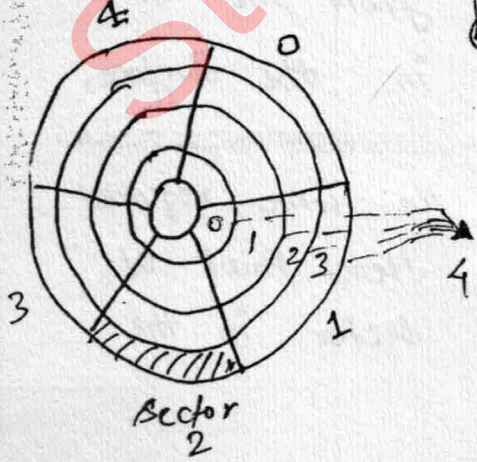
* Each Platter contain 2 -surfaces. Surface contain set of concentric circles, called as Tracks.
* Track contain set of sectors. Sectors holds the Data.
* In the Hard- disk each surface has its own Read/write Head to access the data directly from the Hard- Disk.
* In the Hard- Disk Surfaces, Tracks, and Sectors are the Addressible units.
* In the Hard- Disk Cylinder is formed by Connecting the same track number in all the surfaces.



{ Hard Disk with 4 Platters }







① # cylinders in the Disk = # tracks in the Surface

② Track capacity = # sectors/track * # Bytes ~~in the~~ Per sector

③ Cylinder capacity = # Surface in the Disk * Track capacity

④ Disk capacity = # cylinders in the Disk * Cylinder capacity

Alternate :-

Disk capacity = # surfaces in the Disk * # tracks Per surface * # sectors Per track * # Bytes Per sector

Different Adjustments are required in the Disk-operation :

1) Seek Time : Amount of the time required to move the head from the initial position to desired track in the Surface.

2) Rotational Latency :- Amount of the Time required to adjust the Head-Point at the begining of a desired sector in the track.

In Best case, Rotational latency is '0' (zero) because Head is adjusted during the seek time itself.

In worst case, ~~Rot~~ one complete revolution is required to adjust the Head point.

In Average case Half of the Revolution is required to adjust the Head point. ⇒ $\frac{1}{2} * \frac{\text{Revolution}}{\text{speed}}$

Transfer Time :-

Amount of time required, to access the Data from the Disk

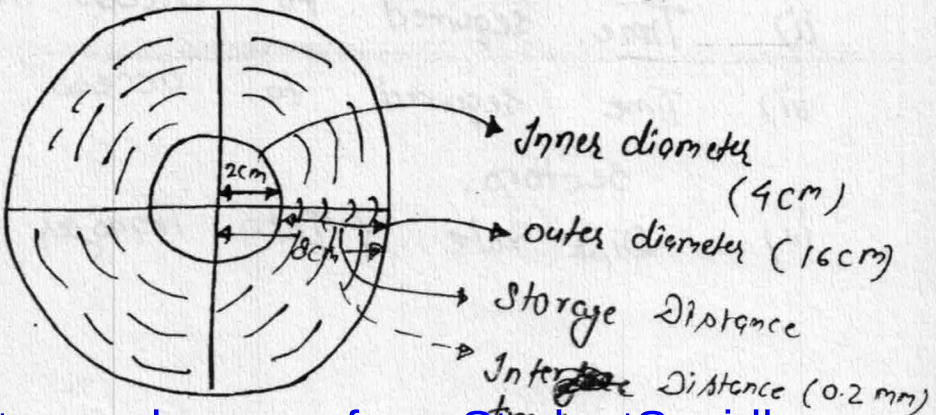
overhead :-

Additional Delays in the Disk operation.

∴ So Total Time Required to Access the Data from the Hard disk :

$$T_{avg} = \text{Seek time} + \text{Avg Rotational latency} + \text{Transfer time} + \text{overhead}$$

∴ In the Hard Disk Design, Storage distance existed ~~adjusted~~ between inner & outer diameter of a surface i.e.



$$\text{Storage Distance} = \begin{bmatrix} \text{outer} \\ \text{Radius} \end{bmatrix} - \begin{bmatrix} \text{Inner} \\ \text{Radius} \end{bmatrix}$$

$$= [\,8\text{cm} - 2\text{cm}\,]$$

$$= 6\text{cm}$$

$$\#\text{ tracks in the Surface} = \frac{\text{Storage Distance}}{\text{Inter Track Distance}}$$

$$= \frac{6\text{cm}}{0.2\text{mm}} \Rightarrow \frac{60\text{mm}}{0.2\text{mm}} = 300$$

***

Que:- Consider A Hard Disk with a rotational rate of 9800 RPM. Average seek time of a Disk is 8.4 msec. Disk contain 32 platters. Surface contain 128 tracks. track contain 256 sectors. sectors holds 512 Bytes Data. File size in the Application is 8kB. calculate the following:

i) Disk capacity.
ii) Time required to access the file
iii) Time required to access the 100 random sectors.
iv) Disk rate { Data Transfer Rate }

Sol^{n} :-

(i) Capacity of Disk = # surfaces in disk * # tracks/surface * # sectors/track * # Bytes/sector

= 64 * 128 * 256 * 512 B

= $2^6 * 2^7 * 2^8 * 2^9$ B

= 2^{30} B = 1GB

ii) Time required to access file :-

a) Seek time = 8.4 msec

b) Rotational latency = Depends on the rotational speed

9800 revolution = 1 min (60 sec)

1 revolution = $\frac{60}{9800}$ = 6.12 ms

Avg Rotational latency = $\frac{1}{2}$ revolution time

= $\frac{1}{2}$ x 6.12 ms

= 3.06 ms

c) Transfer time = Depends on Rotational speed.

1 revolution time = 1 track (256 sectors)

? time = (1 file) = 16 sectors

$$\left\{\begin{array}{l} 1 \text{ sector} - 512\,B \\ \#\text{ sector} - 1 \text{ file } (8kB) \\ \Rightarrow \dfrac{8kB}{512} = \underline{16} \end{array}\right\}$$

$$\Rightarrow \frac{6.12\ ms * 16}{256}$$

$$\Rightarrow 0.381\ \underline{ms}$$

$$\therefore T_{avg} = 8.4 + 3.06 + 0.381$$

$$= 11.841\ \underline{ms}$$

(ii) Random sectors accessing requires the adjustment for every sector.

1 revolution time — 1 track (256 sectors)

? time — 1 sector

$$\Rightarrow \frac{6.12\ ms * 1}{256}$$

$$= 0.023\ \underline{ms}$$

$$T_{avg} = \underbrace{(8.4 + 3.06 + 0.023)\ ms * 100}_{1 \text{ sector access time}}$$

$$= 1146.0023\ ms$$

iv) Disk Rate {Bandwidth} :-

:- Surface Rate :-

1 revolution time — 1 track Data {256 * 512 B}

1 sec — ?

$$\Rightarrow \frac{256 * 512\ B}{6.12\ ms}$$

Surface Rate $\Rightarrow \frac{256 * 512}{6.12} * 10^3$ Bytes/sec.

∴ Disk Rate :-

$\Rightarrow$ # Surface in Disk * Surface Rate

$$\Rightarrow 64 * \frac{256 * 512}{6.12} * 10^3\ B/sec.$$

$$= 1370687.58 * 10^3\ B/sec.$$

$$= 1370\ MBPS$$

— X —

module-2