

Lecture 1

* Computer Organization & Architecture

1. Memory interfacing

- Memory hierarchy
- Main Memory
- Cache Memory → 2M.
- Secondary Storage
 - * Mag. disks
 - * Mag. Tapes

2. I/O interfacing

- I/O transfer
- Data transfer modes

1. Machine instructions

2. Control unit design

3. Addressing modes

4. Data representation → 2M.

X 5. ALU Data Path

6. Pipelining

→ 1/2M

≈ Avg 5-7 M

* 1 bit represents 0/1.

8 bits rep → 1 byte.

1024 bits → 1 KB (Kilo) 2^{10}

1024 MB → 1 MB (Mega) 2^{20}

1024 GB → 1 GB (Giga) 2^{30}

1024 TB → 1 TB (Tera) 2^{40}

4 bits → 1 nibble

S.R.
No
- AM from (cache).

- Data REP (AM)

- Data REP (AM)
 1. Morris Mano
 2. Hamacher
 3. Paul Graybill

3. Paul Graybill

• 1 bit → 0 or 1 → 2 possibilities.
 2 bits → 00, 01, 10, 11 → 4 poss.
 3 bits → 000 - 111 → 8 poss.

K bits → 2^K possibilities

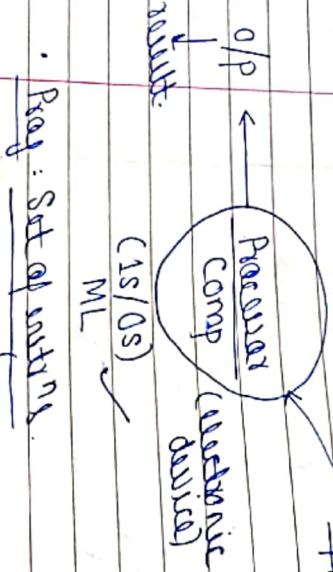
VE

A digital system represents 986 possible
the no. of bits to specify that:

$$986 \approx 2^{10}$$

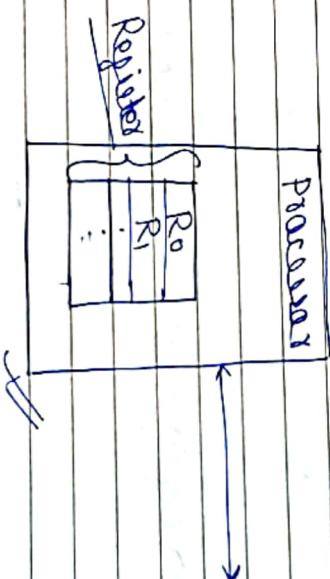
10 bits required.

$$\lceil \log_2 986 \rceil = 10$$



* Memory Interfacing

[PTO]



Passive

Date _____
Page No. _____
Page _____

Very IMP.

- prog. data reside in secondary storage.
- into MM: Process.

Processor Cache: Prog used by processor.

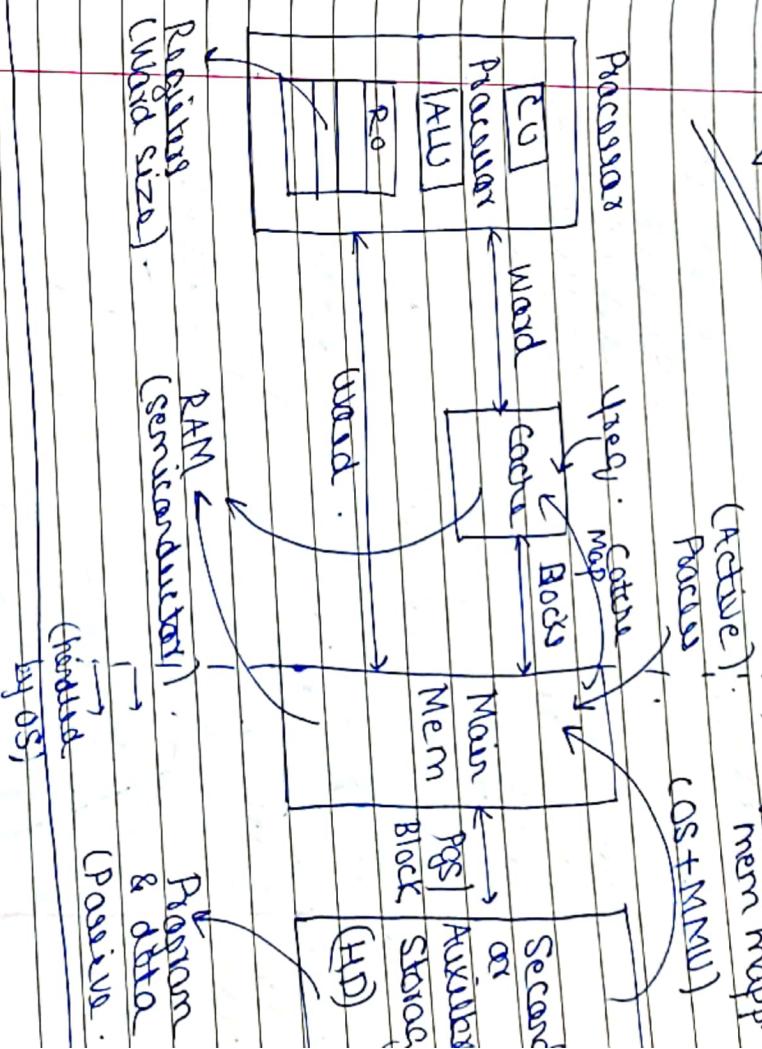
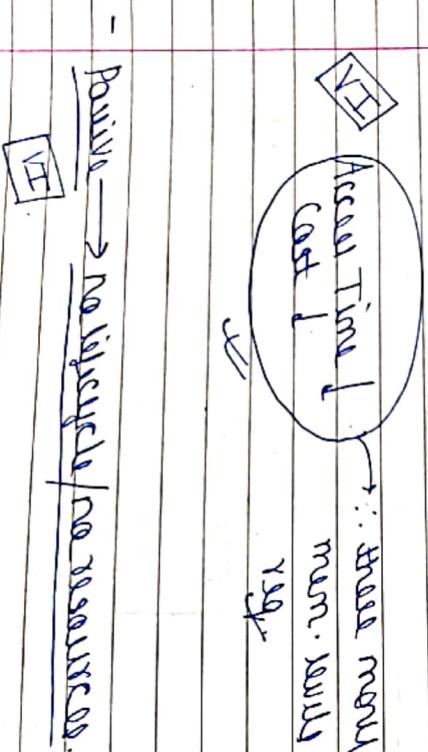
Processing takes place using ALU / CU.

\Rightarrow Processor ref. just 1 word at a time.

\hookrightarrow very fast
"4 MIPS".

1. Access time \downarrow (fast)

\rightarrow Accessing from semiconductor memory is fast. (RAM).
 \rightarrow Cost of semic. device \uparrow . (RAM) \uparrow .



- Possible \rightarrow no physical free resources.

Logical Aspects

Page No.: 7
 Date: 10/10/2023

Physical Aspects:

- Access time (T_i)
- Size (S_i)
- Cost (C_i)
- Frequency (f_i)

- Flaps

- wide processor

- limited set.

Registers

- made up of SRAMs

- Internal mem banks

- no processor.

(KB - MBS)

Main memory

- made up of DRAMs

- External

- mass to processor

(MB - GBS)

Magnetic Disks

- made up of DRAMs

- Internal

- mass to processor

(MB - GBS)

Magnetic Tapes

- made up of DRAMs

- External

- mass to processor

(GBS - TBS)

Access time increases (T_i)

* Size increases (S_i)

* Cost decreases (C_i): Cost of MD < cost of RAM.

* Frequency decreases (f_i)

* Processor operates to take data from main mem. hierarchy.

* $T_{th} = C(T+1)T_m$

* In the Mem. Hierarchy:

• T_m level is a subset of $(T+1)T_m$ level.

• when processor refers to T_m level memory if it is found → hit otherwise miss.

Yield:

• can be 40-60% also

• can be directly accessed by processor.

Main Memory (RAM)

Address

Main Memory

- column of rows
- column → 1 word
- Each row is accessed by spec address.
- Forward is occurred →
- Each word is required.
- In row → n bits required
- Group of consec. words → blocks
- Group of consec. words → blocks

[Block Size = 2^k]

→ bits on ground

- Each word → 8 cells (0/1)
- (1 word → 8 bits → 1 byte).

bit organised memory

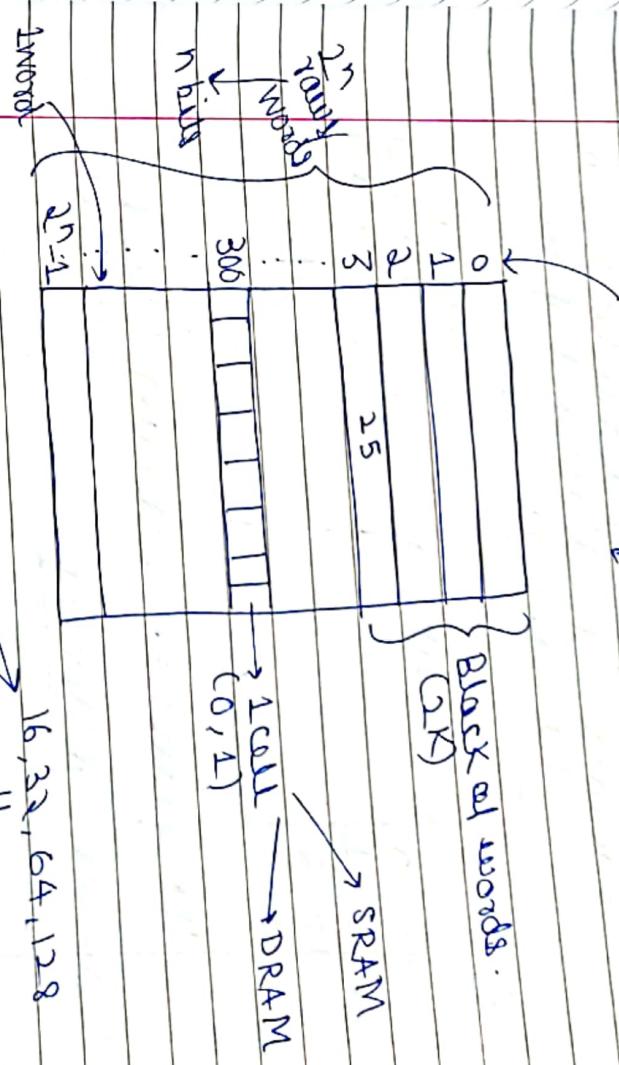
\times 1 word → 16
→ 32
→ 64
[VI]

word organised memory

\times Row → 0/1 (Row?) [Based on structure
of 1 cell]
Static RAM
Dynamic RAM.

* if add = 300 → 8 bits loc.
(Any word is accessed in fixed amount
of time)

- Organisation is same; differ in how 1 cell
- is reconstructed.



RAM (i.e. Random Access Memory)

(Each cell stores 0/1)

Page No.: 111 | Date:

- Once address is specified, any location can be accessed in fixed amount of time.

- One address is specified, any location can be accessed in fixed Time].

[NA Search Time]

DRAM

- Static Random Access Memory

- Dynamic Random Access Memory

~~SRAM~~

addressable memory



$$\text{No. of rows} = 4 \therefore 2 \text{ bits} \quad [1 \text{ word} = 16 \text{ bits}]$$

SRAM

= 16 bits

- Sustaining power supply required.

. No additional power required.

. Refreshing circuit is reqd.

(new value)

[Charging done row wise]

visit one cell is

less.

- Number of cell more than SRAM.

(due to G transistor)

memory

→ 8 words

(By bank)

· small dimensions

Big memory depth

(slow).

· 3 bits

for it just lost.

· Conventional delay

Conventional delay is

in less (G transistor is more)

high). [fast]

· Relatively fast.

· Relatively slow.

· Cache Memories

· Main memory

· cost is high.

· cost is low.

SIZE OF RAM

- No. of words \times No. of bits/word.

$$\text{Eq: } 32K * 16$$

$$32K \text{ rows} = 2^{15} \text{ rows} \Rightarrow 15 \text{ bit address}$$

$$16 \text{ bits/row} \Rightarrow 16 \text{ data bits}$$

+ major comp. in program comm with each other via System bus.

$\left\{ \begin{array}{l} 16 \rightarrow \text{Data bus} \\ 15 \rightarrow \text{Address lines} \\ 7 \rightarrow \text{Control lines} \end{array} \right\}$ System bus.

depends on CPU.

depends on control signals

$$\text{if } 100CS \rightarrow 2^7 = 7 \text{ bits}$$

$$\cdot \text{Capacity} = \frac{32K * 16}{8} = 64 \text{ KB}$$

in terms of bytes \nwarrow

~~How many bits required to address a 4 GB byte organised Main memory.~~

$$2^2 \times 2^{30} = 2^{32}$$

~~byte organised memory.~~

$$\text{no. of rows} = \text{no. of bytes} =$$

$$\boxed{4K} \quad 4 \text{ GB}$$

$$\Rightarrow 2^2 \times 2^{30} = 2^{32} \text{ rows}$$

$$\Rightarrow 32 \text{ bits reqd. for address}$$

~~How many bits required to address a 4 GB memory with a word size of 64 bits.~~

\nwarrow Word addressable.

$$1 \text{ row} = 1 \text{ word} = 64 \text{ bits} = 8 \text{ bytes.}$$

$$\Rightarrow \boxed{\text{no. of rows} = \text{no. of words}}$$

\nwarrow word Address.

$$4 \times 2^{30} \Rightarrow 2^{29} \text{ rows.}$$

\nwarrow 29 bit address.

The possible number of address and data lines to access a RAM size of $256M \times 32$.

How many "RAM chips" of size of $32K \times 1$ required to construct 512 KB main memory.

$$2^{28} \times 32$$

\rightarrow Byte.

SOL:

$$2^9 \times 2^{10}$$

$$2^9 \times 2^{20} \Rightarrow 2^{28} \text{ rows}$$



28 bit address

• 32 data lines
[No. of words = No. of rows]

The possible no. of address and data lines to access 512 MB memory with a word size of 64 bits.

SOL:

$$64 \text{ bits} = 8 \text{ bytes}$$

word addressable:

$$2^9 \times 2^{20} = 2^{29} = 2^{26} \text{ rows}$$

$$\text{No. of rows} = 2^{26} \text{ words}$$

word add.

26 address lines

64 data lines

10ns/row.

$$\Rightarrow 2^{26} \text{ rows}$$

10ns/row.

$$\Rightarrow (2^{28} * 10) \text{ ns}$$

bit.

→ [We need only no. of such as].

Date: _____
Page No.: _____
Name: _____
Date: _____

$$2^{28} = 10^{12}$$

$$K_2 = 28 \log_{10}^2$$

Consider a DRAM with 2^{14} rows. The time for each row to be accessed is 50 ns. What is the percentage of time left used for performing read/write operations if the refresh period is set to 2 ms.

Soln:

$$\text{Refresh time / chip} = 2^{14} * 50 \text{ ns}$$

$$\Rightarrow \frac{2^{14} \times 50}{10^6} = 0.8192 \text{ ms}$$

$$\Rightarrow 0.82 \text{ ms}$$

Time ref. to refresh
the chip.

$$\Rightarrow 2 - 0.82 = 1.18 \text{ ms}$$

But for read/write

$$\frac{1.18 \times 100}{2} \Rightarrow 59\%$$

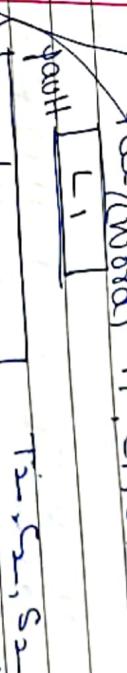
(word)

[refresh in DRAM done row wise]

* Memory Hierarchy

- In a level memory system:

Case 1: (Cyc allowed) → Access Time of L1



If percent hits is α in L1, β from L2.

$$L_1 = \text{catches}$$

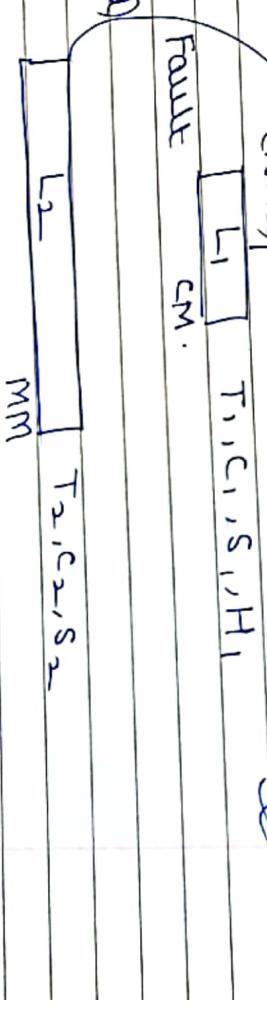
(Analogies).

$$L_2 = \text{misses}$$

Case 2:



→ percent of hits in L1



We don't include search time :: RAM.

→ Upward 100 words
80 is in L.

Date:

Page No.:

Page No. : 19

Date:

Page No. : 19

Page No. : 19

visit of 100 words.

$$80\% \rightarrow L_1 = 80 \text{ words}$$
$$20\% \rightarrow L_2 = 20 \text{ words}$$

H₁ (Hit ratio)

$$T_{avg} = H_1 * T_1 + (1 - H_1) * T_2$$

hit
ratio

Miss
ratio

Miss
penalty

fault

HD.

MM.

block (word)

T₂, C₂, S₂

T₁, C₁, S₁

hit (word)

miss (word)

start hierarchy

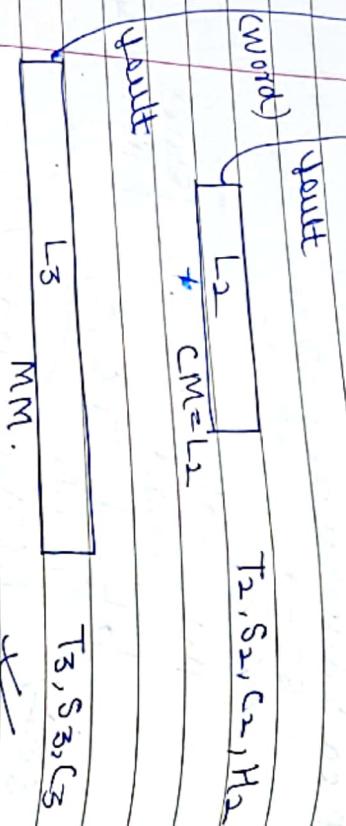
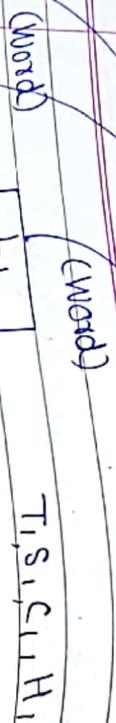
- P contact directed occurs SD
SD → MM → P

$$\text{avg/bit} = \frac{C_1 S_1 + C_2 S_2}{S_1 + S_2}$$

$$T_{avg} = H_1 * T_1 + (1 - H_1) (T_2 + T_1)$$

* with a 3 level memory system:

case 1:
(default case)

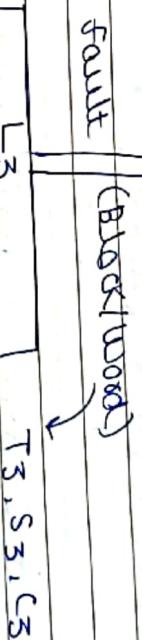


$\Rightarrow 100 \text{ words}$

$H_1 \leftarrow 80\% \rightarrow L_1 \Rightarrow 80 \text{ words}$

$H_2 \leftarrow 90\% \rightarrow L_2 = 18 \text{ words}$

$H_3 \leftarrow 2 \text{ words}$



→ needed to RQ.

$$\therefore T_{avg} = H_1 * T_1 + (1-H_1)H_2(T_1 + T_2) + (1-H_1)(1-H_2)(T_1 + T_2 + T_3)$$

$$(1-H_1)(1-H_2)T_3$$

- Each word fixed access time:

fixed in time of manufacturing.

$$\therefore C_{avg/bit} = C_1 S_1 + C_2 S_2 + C_3 S_3$$

~~Q.~~ Consider a 2 level Memory System. If the access time of level 1 and level 2 is 150 ns. memory is 10ns and hit ratio is 90%. What is L1 hit ratio? Avg. Access time?

Soln.

- Case 1: word transfer (Cache)

$$\Rightarrow T_{avg} = 0.9 * 10 + 0.1 * 150$$

$$\Rightarrow 9 + 15 = 24\text{ ns}$$

~~Q.~~ consider a system in which cache read miss $\rightarrow 50\text{ ns}$. and hit will take 5 ns. when a program is referenced 90% of read request result in hit. what is avg. access time?

Soln.

$$T_{avg} = 0.9 * 5 + 0.1 * 50$$

$$\Rightarrow 4 + 10$$

$$\Rightarrow \underline{\underline{14\text{ ns}}}$$

In a 2 level Memory System, average access time without L1 is 150 ns. and with L1 is 30 ns.

$$L1(\text{Access time}) = 20\text{ ns}$$

- (ii) What is the hit ratio?

$$T_2 = 150\text{ ns} \quad (\text{Without } L1)$$

$$T_{avg} = 30\text{ ns} \quad (\text{with } L1)$$

$$T_1 = 20\text{ ns}$$

$$\Rightarrow 30 = x * 20 + (1-x) * 150$$

$$30 = 20x + 150 - 150x$$

$$\Rightarrow -120 = +130x$$

$$x = \frac{12}{13} = 0.92$$

92.3% Hit ratio.

$$H \cdot T_1 + (1-H) T_2 = 30$$

$$H \cdot 20 + (1-H) (150) = 30$$

$$20H + 150 - 150H = 30$$

$$+130H = +120$$

~~✓~~

If the hit ratio is made to 100%. what is the access time of $L_1 \oplus L_2$

(iii)

$$T_{avg} = H_1 T_1 * (1 - H_1) T_2$$

$$T_{avg} = \frac{H_1}{1 - H_1} * 100\%$$

$$T_{avg} = \frac{100}{1 - 0.1} = 111.1$$

→ Main memory



(a) 3% decrease in Hit Ratio.

$$\Rightarrow T_1 = 20, T_2 = 150$$

$$\Rightarrow 33 = 20H_1 + (1 - H_1)150$$

$$\Rightarrow H_1 = \frac{111.1}{130} \Rightarrow 0.85$$

T_{avg} = 20 ns

T₁ = 20 ns

T₂ = 150 ns

VI

NOTE: Hit ratio doesn't influence individual memory access times.

At 0.8 Hit via level 1 memory, avg access time is 160 ns, level 1 memory is 5 times faster than level 2. If avg. Access time is increased by 20%, what is the percentage change in hit ratio?

→ If avg. access time is increased by 10%, what is the change in hit ratio?

- (i) 10% increase
(ii) 20% increase
(iii) 20% decrease
(iv) 10% decrease

VI

$$T_{avg} = 20 + 0.1 \times 2$$

$$\Rightarrow 20.2$$

$$= 33 \text{ ns}$$

Soln: $H_1 = 0.8, T_{avg} = 60, T_2 = 5T_1$

$$\frac{T_2 = 5T_1}{H_1 = 0.8}$$

$$T_{avg} = H_1 T_1 + (1 - H_1) T_2$$

$$60 = 0.8 T_1 + 0.2 \cdot 5 T_1$$

$$0.8 T_1 + T_1$$

$$60 = 1.8 T_1$$

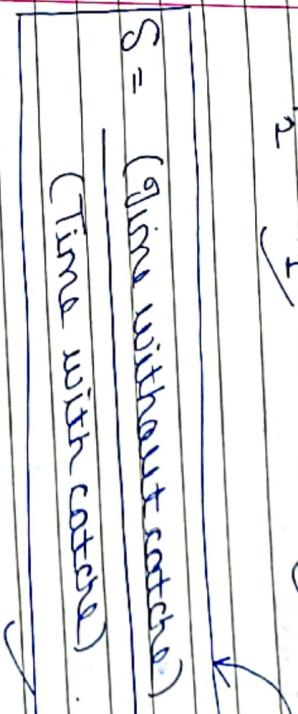
$$T_1 = 33.33 \text{ ns}, \quad T_2 = 166.65$$

$$H_1 = 80\%, \quad T_{avg} = 60 \text{ ns}, \quad T_2 = 5T_1.$$

Q) Consider a system in which the access time of main memory is 5 times to that of cache. If cache hit ratio is 0.8. What is the speed up?

Soln:

$$T_2 = 5T_1, \quad H_1 = 0.8$$



$$S = \frac{T_2}{T_1}$$

$$\Rightarrow T_{avg} = H_1 T_1 + (1 - H_1) T_2$$

$$\Rightarrow T_{avg} = H_1 T_1 + (1 - H_1) T_2$$

\therefore (a) 20% decrease.

~~∴~~

$$\Rightarrow \frac{1}{0.8 + 1 - 0.8}$$

$$\Rightarrow \frac{1}{0.8 + 0.2} \Rightarrow 2.75.$$

$$\text{Speed up} = 2.75 \times$$

~~∴~~

$$\frac{120 \times 10^9}{100}$$

$$T_{avg} = H_1 T_1 + (1 - H_1) T_2$$

$$60 = 0.8 T_1 + 0.2 \cdot 5 T_1$$

$$0.8 T_1 + T_1$$

$$60 = 1.8 T_1$$

$$T_1 = 33.33 \text{ ns} ; T_2 = 166.65$$

$$H_1 = 80\% , T_{avg} = 60 \text{ ns} , T_2 = 5 T_1$$

$$\therefore T_2 = \kappa(33.33) + (1-\kappa)(166.65)$$

$$\Rightarrow H_1 = 0.7 \Rightarrow 70\%$$

\therefore (d) 10% decrease.

$\frac{100}{100} \times 10$

$\frac{100}{100} \times 10$

$$T_2 = 5 T_1$$

$$H_1 = 0.8$$

Page No.: 27
Date:

Q. Consider a system in which the access time of main memory is 5 times to that of cache.
If cache hit rate is 0.8.
What is the speed up?

Soln.

$$T_2 = 5 T_1 \quad H_1 = 0.8$$

VI

$$S = \frac{\text{(Time without cache)}}{\text{(Time with cache)}}$$

$$\Rightarrow S = \frac{T_2}{T_{avg}}$$

$$\Rightarrow \frac{T_2}{T_1 H_1 + (1 - H_1) T_2}$$

$$\Rightarrow \frac{T_2}{T_2 \left(\frac{H_1}{5} \right) + (1 - H_1) T_2}$$

$$\Rightarrow \frac{1}{0.8 + 1 - 0.8}$$

$$\Rightarrow \frac{1}{0.8 + 0.2} \Rightarrow 2.78$$

$$\therefore \text{Speed up} = 2.78$$

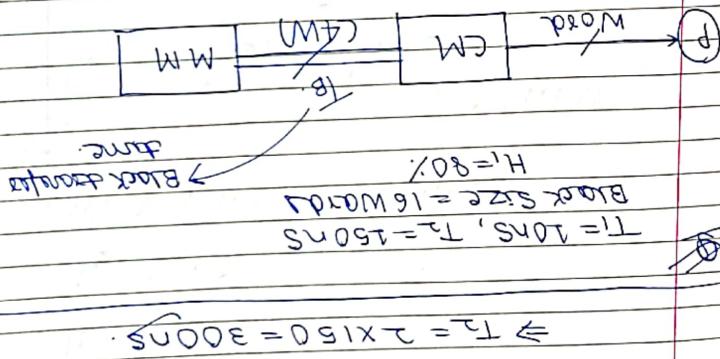
$$T_{avg} = H_1 T_1 + (1-H_1) (T_1 + T_2)$$

$$= 0.8 \times 10 + 0.2 \times (6.10)$$

$$= 8 + 2 \times 6.1$$

$$= 8 + 12.2$$

$$\Rightarrow 130 \text{ ns}$$

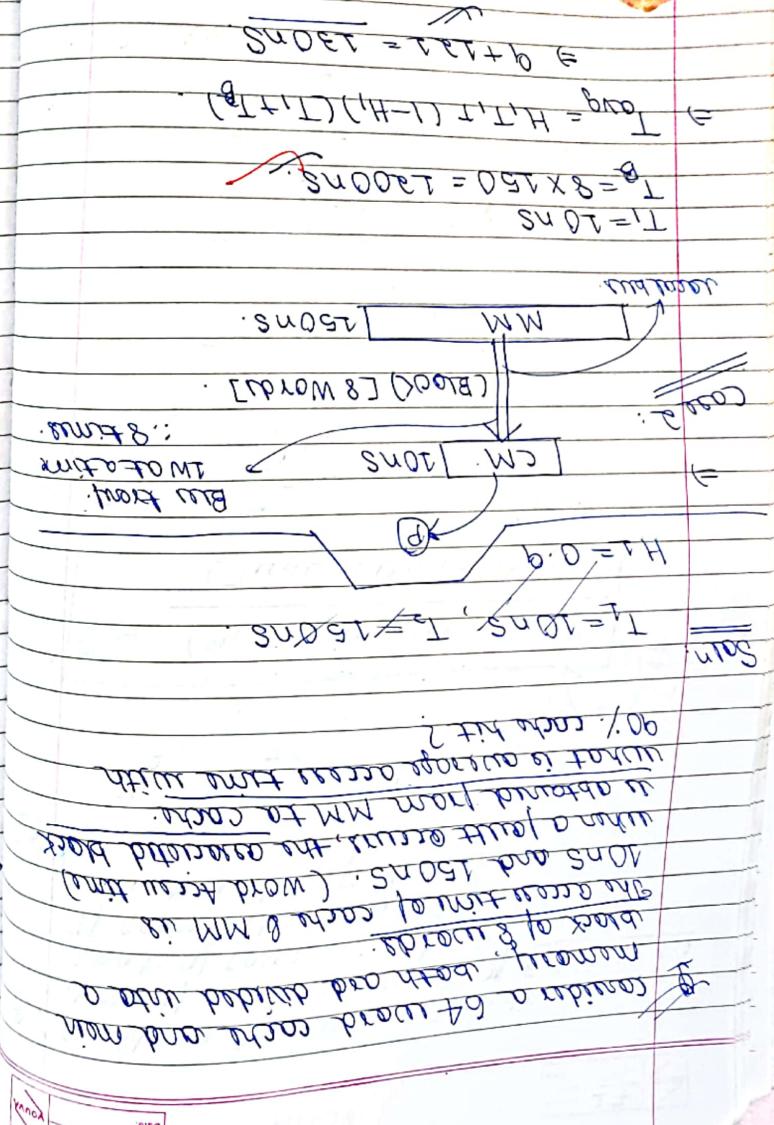


$$T_2 = 4 \times 150 = 600 \text{ ns}$$

\therefore Bus contention (a.w.) \rightarrow 4 times

$$T_2 = 4 \times 150 = 600 \text{ ns}$$

Page No.: 29
Date: _____



$$T_{avg} = \frac{14}{20} \times 18 + \frac{6}{20} \times 14$$

$$= 12.6 + 4.2$$

$$= 16.8 \text{ ns}$$

~~$T_{avg} = 16.8 \text{ ns}$~~

Date: _____
Page No.: 31

$$T_{avgw} = 0.8 \times 10 + 0.2 \times 50$$

$$= 8 + 10 = 18$$

$$T_{avgw} = 0.9 \times 10 + 0.1 \times 50$$

$$= 9 + 5 = 14$$

$$T_{avg} = f_L T_{avgw} + f_W T_{avgw}$$

bottom
base
op.
outlier

$$T_{avg} = \frac{140}{200} T_{avgw} + \frac{60}{200} T_{avgw}$$

$$T_{avg} = f_L T_{avgw} + f_W T_{avgw}$$

T_{avgw}

what is avg. acc. in form?

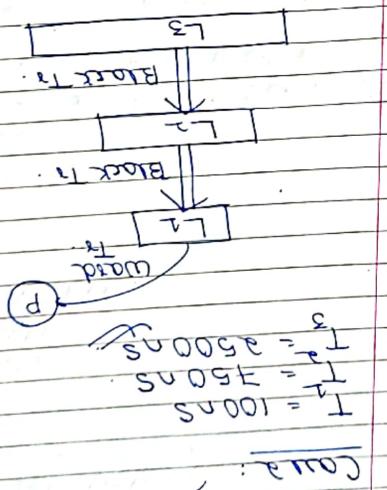
$$T_1 = 10 \text{ ns}, T_2 = 50 \text{ ns}$$

$$H^R = 80\%, H^W = 90\%$$

found a program:
for $i = 1$ to n
do $f[i] = 0$
do $j = 1$ to n
do $k = 1$ to n
if $A[i][j] > A[i][k]$ then
 $f[j] = f[j] + 1$
if $A[i][j] < A[i][k]$ then
 $f[k] = f[k] + 1$
done
done
done

for $i = 1$ to n
do $j = 1$ to n
do $k = 1$ to n
if $A[i][j] > A[i][k]$ then
 $f[j] = f[j] + 1$
if $A[i][j] < A[i][k]$ then
 $f[k] = f[k] + 1$
done
done
done

$$\begin{aligned}
 T_{avg} &\rightarrow 3000ns \\
 0.2 \times 0.1 \times 3350 & \\
 + 0.8 \times 1000 + 0.2 \times 0.9 \times 1450 + & \\
 (1-H_1)(1-H_2)(T_1+T_2+T_3) & \\
 + (1-H_1)H_2(T_1+T_2) + &
 \end{aligned}$$



3-level memory

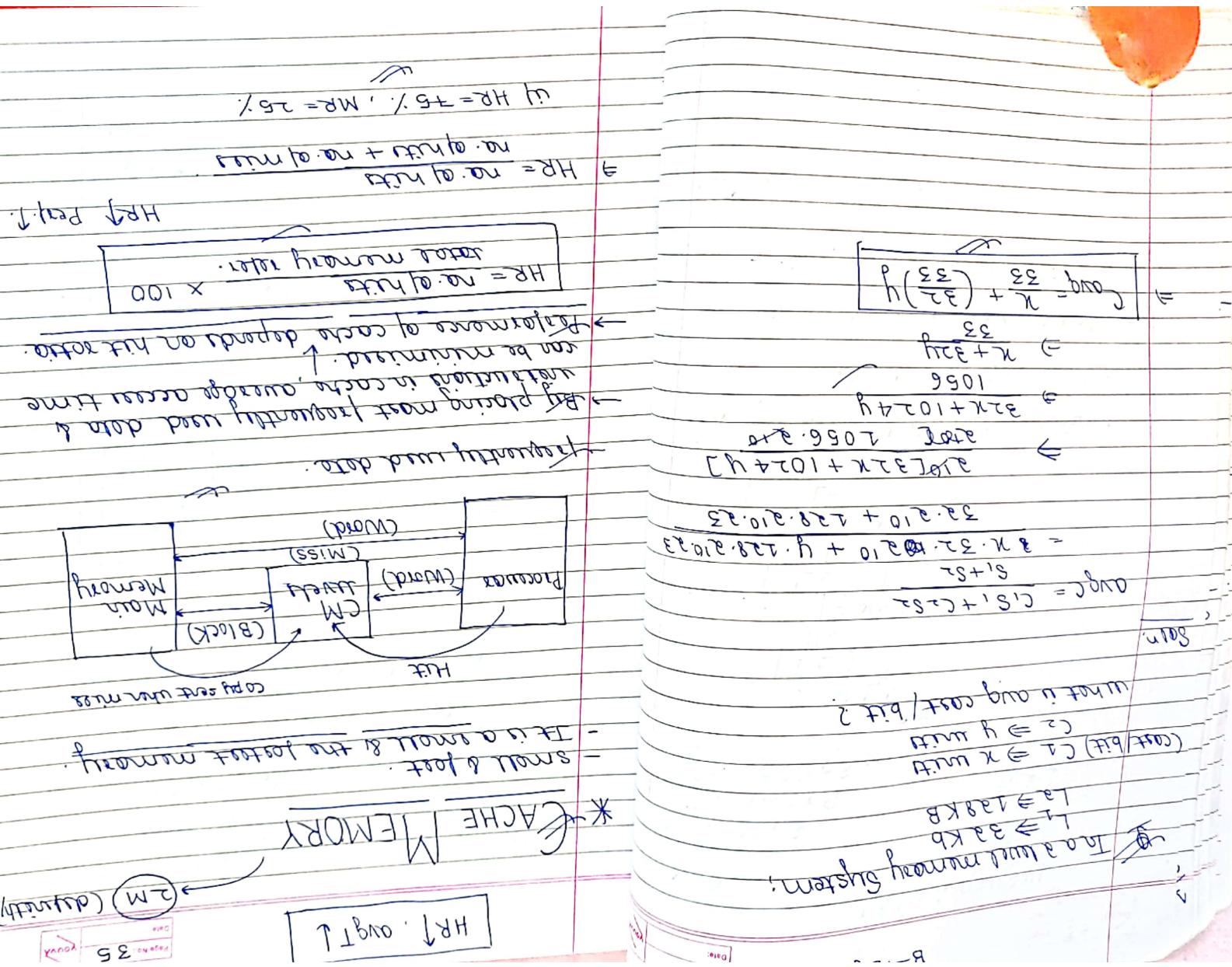
avg. access time? \therefore
1% and 1% is 80% and 90%. What is
hit rate of memory? The hit rate of
main memory is 500 ns. The access time of
main memory is 100 ns, 150 ns and
500 ns. The L2 is 13 ns memory and
access time of main memory is 1, will be and
consider a 3-level memory system. The
hit rate of cache is 80% and 90%.

$$\begin{aligned}
 T_{avg} &= 12.6ns \\
 = 12.6 & \\
 0.8 + 0.2 \times 4.8 + 10 & \\
 \times 0.1 \times 500 & \\
 T_{avg} = 0.8 \times 1 + 0.2 \times 0.9 \times 10 + 0.2 & \\
 (1-H_2)T_3 & \\
 T_{avg} = H_1T_1 + (1-H_1)H_2T_2 + (1-H_1) &
 \end{aligned}$$

avg. access time? \therefore
The hit rate of cache is 0.8 & 0.9. What
is memory is 1 ns, 10 ns & 500 ns.
of main memory is cache, \therefore cache of Main
memory is cache. The access time
(ignoring access time in cache)

of main memory is cache, \therefore cache of Main
memory is cache. The access time
(ignoring access time in cache)

defective



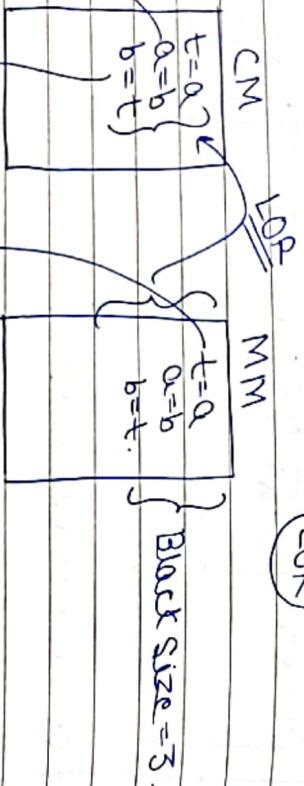
* Cache memory works on the principle:

Variability of reference (LOR)

LOR states that the references to memory at a given time tends to be confined to within a localized area.

(heat-unit: access nearby location only)

so:



LOR

CM

LOR

MM

LOR

④ Miss (Compulsory Miss)

$$HR = \frac{2}{3} = 67\%$$

Block size $\rightarrow 3$

10

$$HR = \frac{2}{3} = 67\%$$

$BS \uparrow, HR \uparrow$

2

→

→ Hit ratio improves with large size block.
→ The performance of cache (Hit ratio) depends on:

- 1) Cache Size: (Small) [KBS - MBS]
- 2) Cache Block Size (Large)
- 3) No. of ways of cache (Min - 2 ways)

One is inside P.C.H.

- 4) Cache Mapping Techniques.
- 5) Cache Replacement Policies.

Logical {
of Cache Update Policies

VI

of

- 1) no. of ways of cache.
- 2) keeping consistency with data diff.

P
H

Answer proc. will be onward for first time its always miss. (in MM block)
"compulsory miss"

- access to LOR, block is sent to CM.

- CMT: when to place MM block into CM.

all address: MM must be same

Date: _____
Page No.: _____
Name: _____

Mapping: where to place MM block
in cache.

Page No. 39
Date: _____

- Cache is full mapped LRU
- Cache is full mapped LRU, which block is to be replaced.

↳ Cache replacement policies.

* CACHE MAPPING TECHNIQUES

- The main memory of the system is having 1024 blocks of 16 words each.
- The cache consists of 64 blocks.
- $\text{No. of words/rows} \rightarrow 1024 \times 16 \text{ words}$
- $\text{MM address} \Rightarrow 2^{10+4} = 2^{14} \text{ words}$

14 bit address

NOTE:		
CM	b - 0	Every MM block has fixed location in cache.
	b - 1	size: itn block of page must be mapped to itn block
	b - 2	→ Cache location of a memory block:
	b - 63	$\boxed{\text{CL}(M) = M : n}$
	n = 64	M → Memory Block ref no. n → No. of cache blocks

* Mapping Techniques:

1) Direct Mapping (DM):

1 page = 64 blocks

$\therefore \text{PAG} = \frac{1024}{64} = 16 \text{ PAG}$.

$$\text{Compuress: } \left\{ \begin{array}{l} \text{CL}(0) = 0 : 64 = 0 \\ \text{CL}(120) = 120 : 64 = 1 \\ \text{CL}(1023) = 1023 : 64 = 63. \end{array} \right.$$

$$\text{Cache: } \left\{ \begin{array}{l} \text{CL}(65) = 65 : 64 = 1 \\ [\text{Replace Blk 120 placed in B-1}] \end{array} \right.$$

↳ : also occurred for last time: comp. min.

→ Page 0000
Page 0001

NE

Page 1111

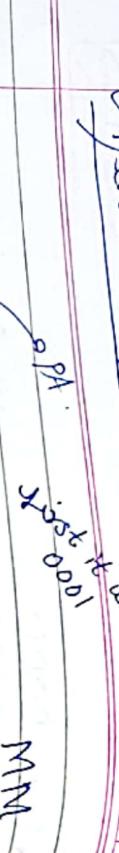
i. itn block of pg → itn block of CM.
ii. (fixed position) ✕

Direct Mapping

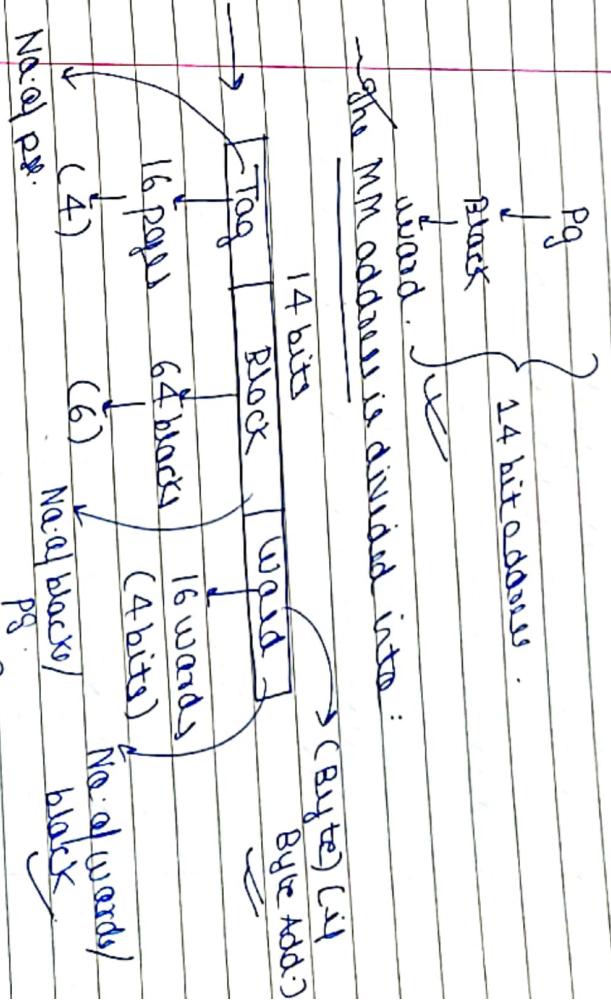


case : no rep. address

Page No:	41
Date:	Today



- The replacement is done when a conflict miss occurs. [no replacement strategy in DML]
- The MM address is divided into :



Higher address bits denote page no.

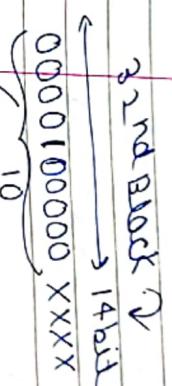


* B-65 conflicted with B-129

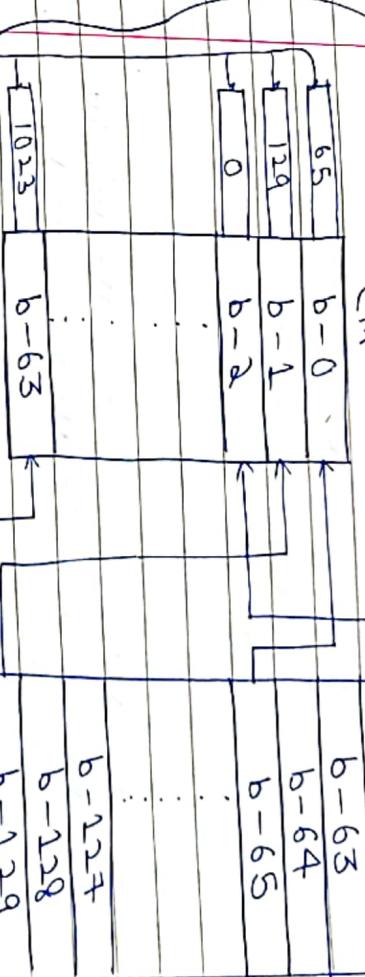
⇒ Max Tag comparison = 1. (if Tag pres = HIT. Not = miss).

5 //

- The higher order bits in the address compound with block offset tag, if a match is found, then hit otherwise, miss. Fault.
- Maximum no of tag comparisons = 1.



C.M

MM
b-0
b-1

1. MAPPING PROCESS IS SIMPLE.
- Assuming same block from different pages simultaneously is always conflict miss.
- Hit ratio is very low. ↘
- [Even though empty blocks, replacement HR less, efficiency ↓]

2. ASSOCIATIVE MAPPING (AM)

- fully associative.

(Cache index → 64 blocks)

(→ 6 bits.)

- a MM block is mapped to any location of cache.
- the replacement is done when cache is full (capacity miss)

→ When a new block is written, at that time cache is full.

(empty block)

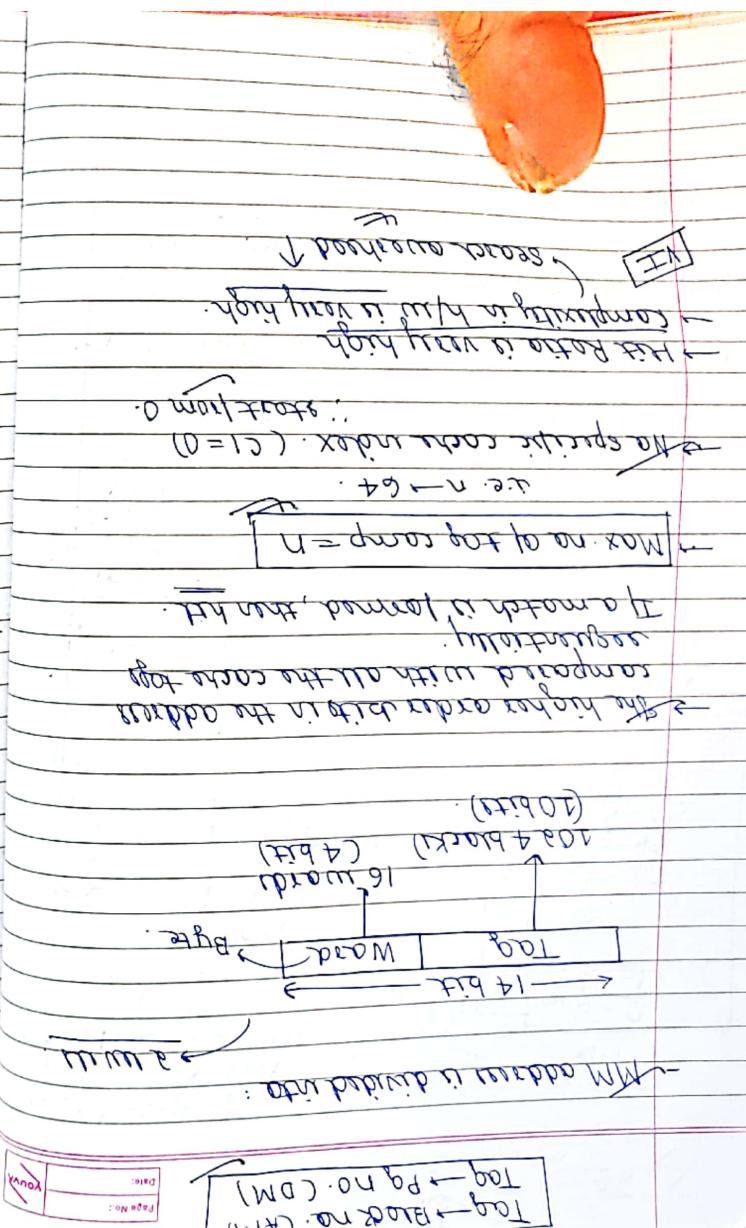
→ DM.

→ AM = 0

(when soft limit)
 → superseded
 (can. with 0, 64)
 $SL(32) = 32 / 32 = 0$ (current min)
 $SL(64) = 64 / 32 = 0$ { Min
 $SL(0) = 0 / 32 = 0$ } current max
 $P = n \cdot M$
 $SL(M) = M / P$
 * a way-SAM. [a block in set]
 Here, K is no. of blocks in a set. (Set size)
 e.g. a way SAM
 + way SAM
 8 way SAM
 ← SAM can be K-way SAM

SET ASSOCIATIVE MAPPING (SAM)

45 bytes



$$\text{Max no. of Tag comp} = k \quad (2)$$

If a match is found, then hit
with all the blocks of set address compared
with high address bits in the address compared

(3) Sample Mapping

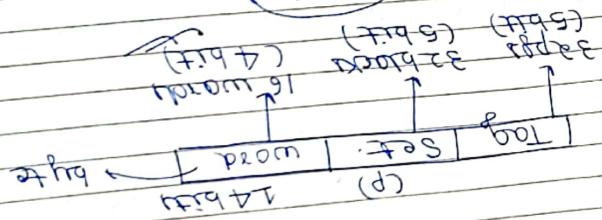
(4) Cache Hit (than DM)

~~Collision~~

$$\text{Set index} = 17 \quad (5 \text{ bits})$$

$$\text{Max Tag comp} = k$$

$$f = n/k$$

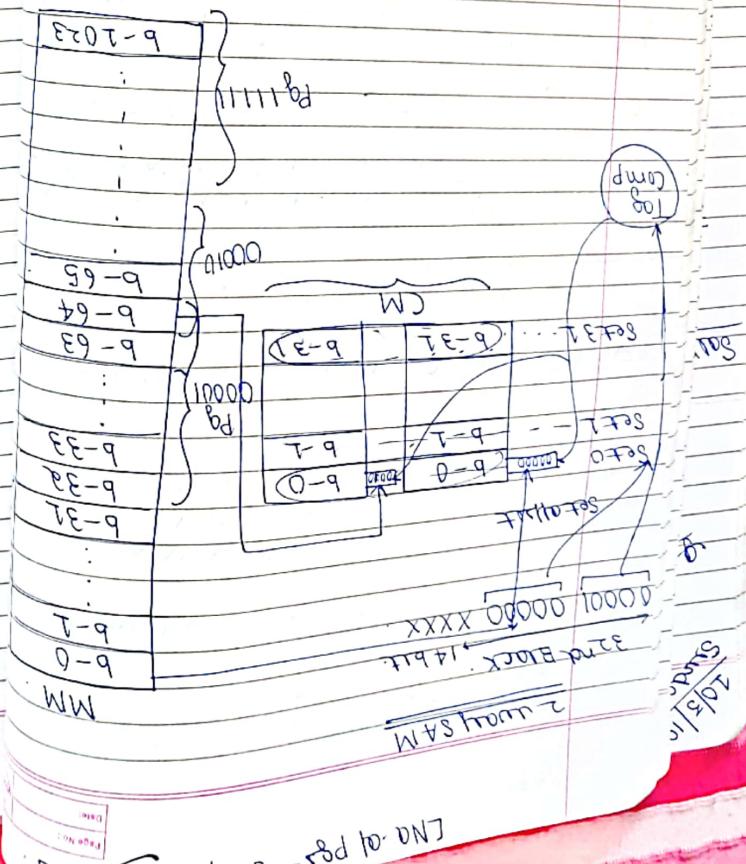


All MM address is divided into

Memory MM Block has k association.
If all blocks in a set is full.
Then it will

$$= 64/2 = 32$$

$$\text{No. of sets (P)} = n/k$$



- Hit Ratio and complexity in hardware
- Hit Ratio and complexity in hardware
- is optimal.
- ~~ASIM is of both DM and AM.~~

~~ASIM is of both DM and AM.~~

~~1 way SAM = DM~~

- SAM is implemented with set of DMs.

~~How SAM can be :~~

b-0	{	s-0
b-1		
b-2		s-1

b-3	{	s-1
-----	---	-----

~~Consider a 64 block cache with 16 words / block. The main system is accessed using 16 bit address. How the address is divided when the tag size is given?~~

4096 blocks \rightarrow MN

~~* DM~~



$$\text{Catch index : } i^{\text{th}} \text{ place} \rightarrow 64 \text{ blocks}$$

6 bits

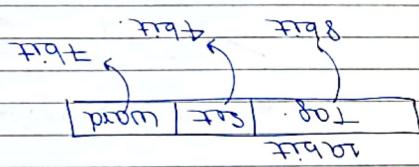
$\frac{\text{6 bits}}{\text{6 bits}}$

~~* AM~~



Catch Index : 0

$$P = \frac{64}{4} = 16 \text{ words}$$



$$\Leftrightarrow 2^{12} \times 2^7 = 2^{19} \therefore 19 \text{ bits.}$$

MM address

$$4096 \times 128$$

Solution

Tag, Set, word
System consists of 4096 blocks. The MM of the computer has 128 words each. Cache with 8 blocks

4-way SAM
h/w
(bits)

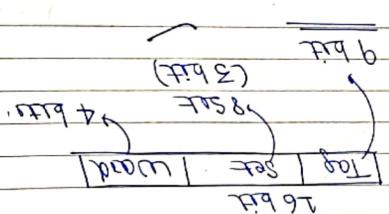


with tag size K
(Tag size) increases

Note

As, $K \downarrow$ Tag size \downarrow complexity \downarrow

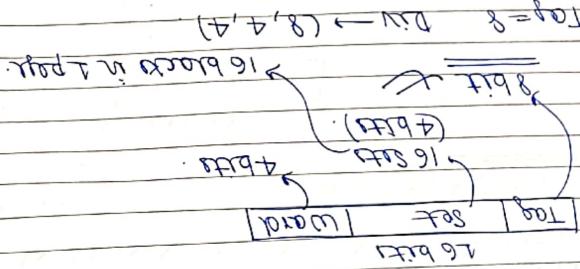
\Rightarrow Set index: 3 bits
 \Rightarrow Tag = 9 bits. DIV $\rightarrow (9, 3, 4)$



$$P = \frac{64}{8} = 8$$

4-way-SAM

\Rightarrow Set index: 4 bits: 16: 4 bits



$$P = \frac{64}{16} = 4$$

4-way-SAM

Scanned with CamScanner

Diagram of a word:

Tag	Black	Word
-----	-------	------

8 * 8 bit = 64 bit

$8 \times 8 \times 23 = 16 \times 23 = 368$

$368 = 2^9$

$n = \log_2 10^6 = 20$

$2^{20} = 10^6$

1 Million word $\leftarrow 10^6$ words

8 million word $\leftarrow 2^{23}$ words

Word addition $\leftarrow 23$ bit address

Word address $\leftarrow 23$ words

Word \leftarrow word

Diagram of a word divided into 8 bytes:

8 bit							
-------	-------	-------	-------	-------	-------	-------	-------

SL(k) = $k/c + a$ \rightarrow $(k/c) + (a - 1)$

$P = \frac{n}{k} = \frac{VC}{c} = C$

Row SAM with VC blocks

Row $\rightarrow k/c + a$

SL(k) = $(k/c) + a$ \rightarrow $(k/c) + 1$

Row for K+1 Black

Column SAM (2 blocks - 1 fill)

Column $\rightarrow k/c$

SL(k) = k/c

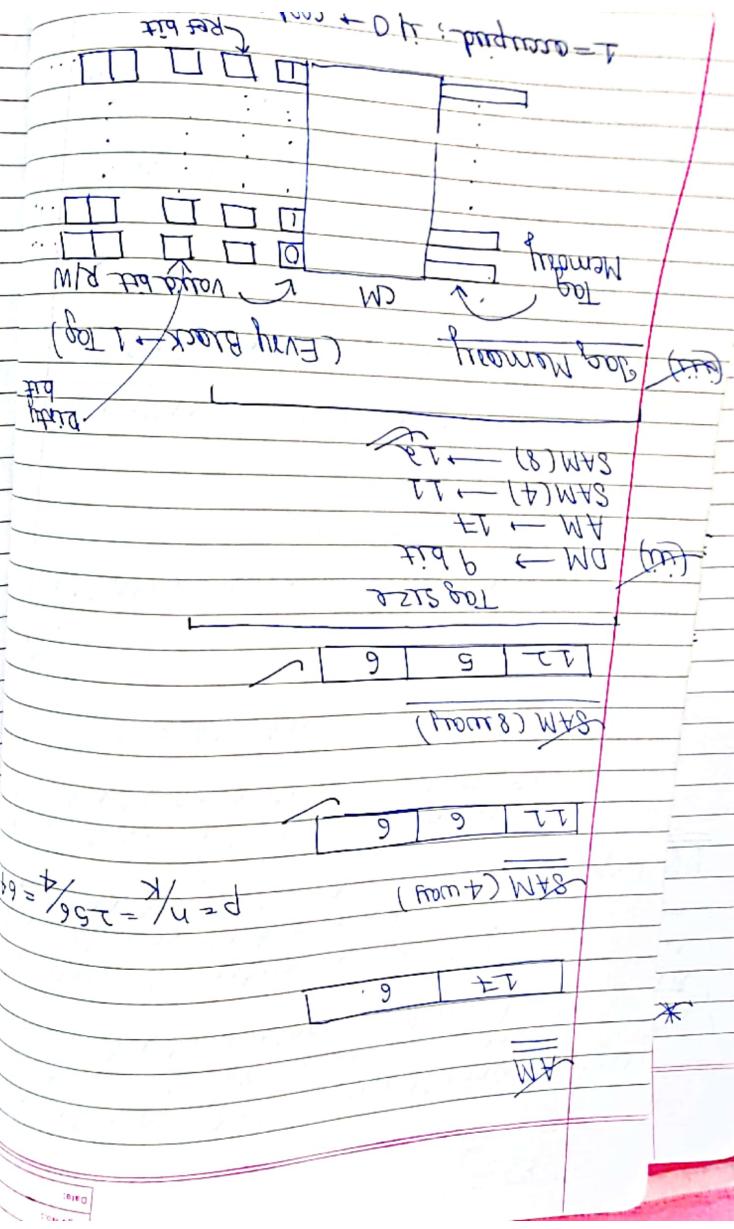
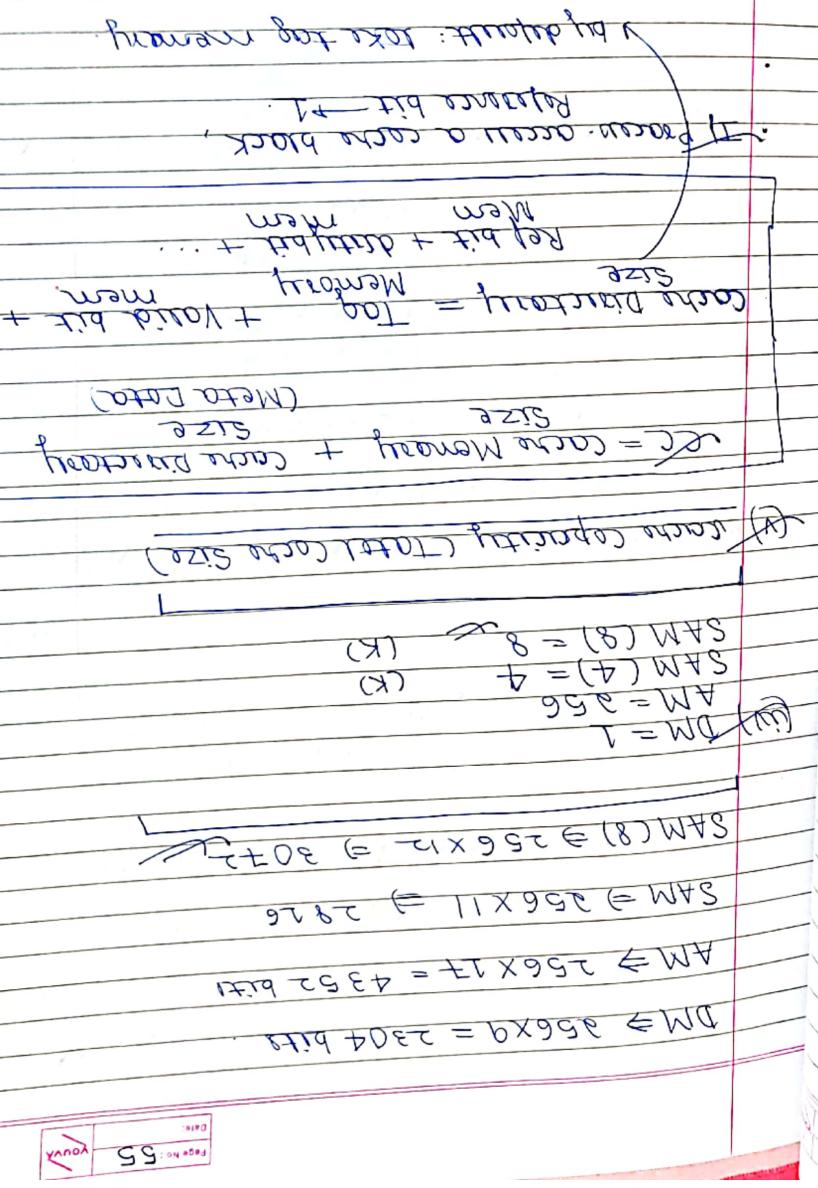
$P = n/c = 2C = C$

SL(k) = $k/c + a$

Row SAM

A way SAM suitable for cache

For a way set associativity cache that has c blocks and M cache blocks of c blocks each and the M count of c blocks do k/c location. The k/c block of memory is mapped.



Sol 1:

Q.

(By default 1 word = 1 byte).

$$CC = \left[256 * 64 * 1 \text{ Byte} + \frac{256 * 9}{8} \right]$$

$$\Rightarrow 16384 +$$

Sol 2:
(166+2) Bytes.Sol 2:

Lat size of word is 32 bit. (Word Size)

$$CC = \left[256 * 64 * 4 + \frac{256 * 9}{8} \right]$$

$$\Rightarrow 2^8 * 2^6 * 2^2 + 2^8 * 9$$

2²³

$$\Rightarrow 64 \cdot 5 \text{ KB}$$

X

$$\Rightarrow 2^{16} + 2^5 * 9$$

$$\Rightarrow 64K + 25 \times 9$$

$$\Rightarrow (64K + 225) \text{ Bytes}$$

DM → Cache Index : no. of blocks
AM → Cache Index : 0
SAC(4) → Cache Index : no. of sets

$$\Rightarrow (n/4)$$

Conclusion: Similarly, for all other mappings.

DM, AM, SAC(4) → Cache Index : no. of blocks/sets
 n/K , $K = n \cdot \alpha$ blocks/sets

Consider a 4-way SAM with 32 bit word

- 1 - dirty bit, 1 valid bit, 1 tag bit,
- 2 - 16 bits of data given.

Cache capacity

16 * 16 * 6 * 6

tag size : 11 bit.

dirty memory : 2816 bit.

Cache directory size = 2821 bits

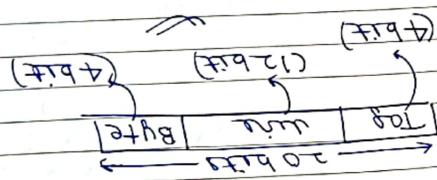
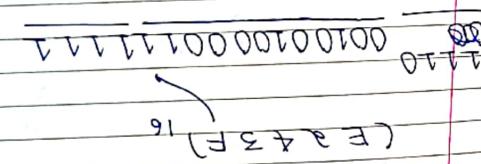
Cache capacity =

$$256 * 64 * 4 + 2821$$

VI

$$\Rightarrow 64K + 3885512$$

$$[\begin{array}{l} \text{Byte} = F \\ \text{unit} = 001001000011 = (\text{Address}) \\ \text{Tag} = 1110 \Rightarrow (E) \\ \text{Data} = 43 \end{array}]$$



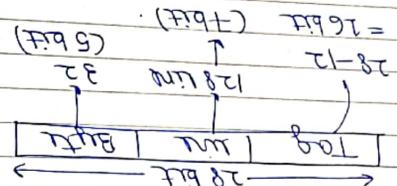
$\rightarrow 20 \text{ bits} \rightarrow 20 \text{ bits}$

Solution:

(E 43 F)₁₆

for the memory address:
what is tag, unit, byte:
A direct mapped cache is of 12-bits.
(a 20 Bit) with 16 bits/bits/bits.

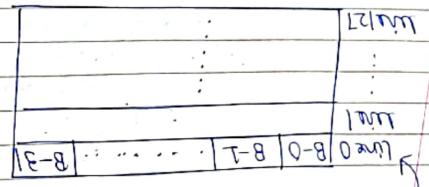
59
Date: _____
Page No. _____



$28 \times 2 = 28 \text{ bits}$

$$28 \times 2 = 28 \text{ bits}$$

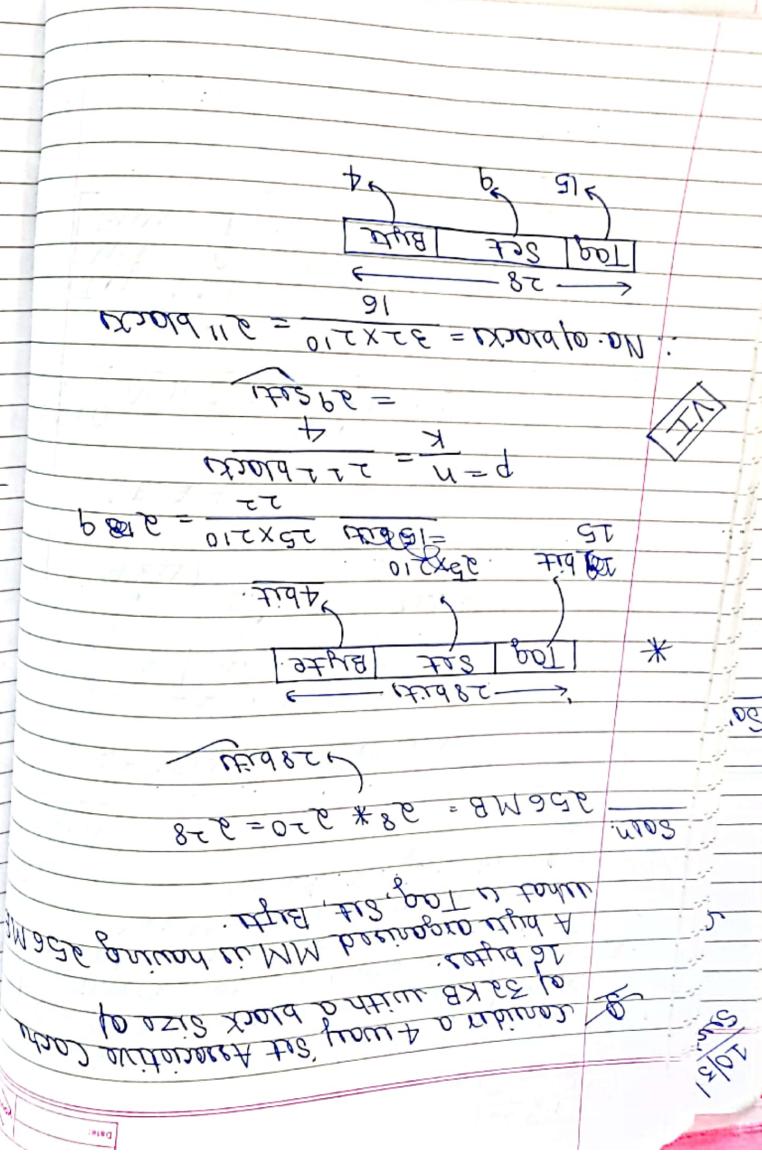
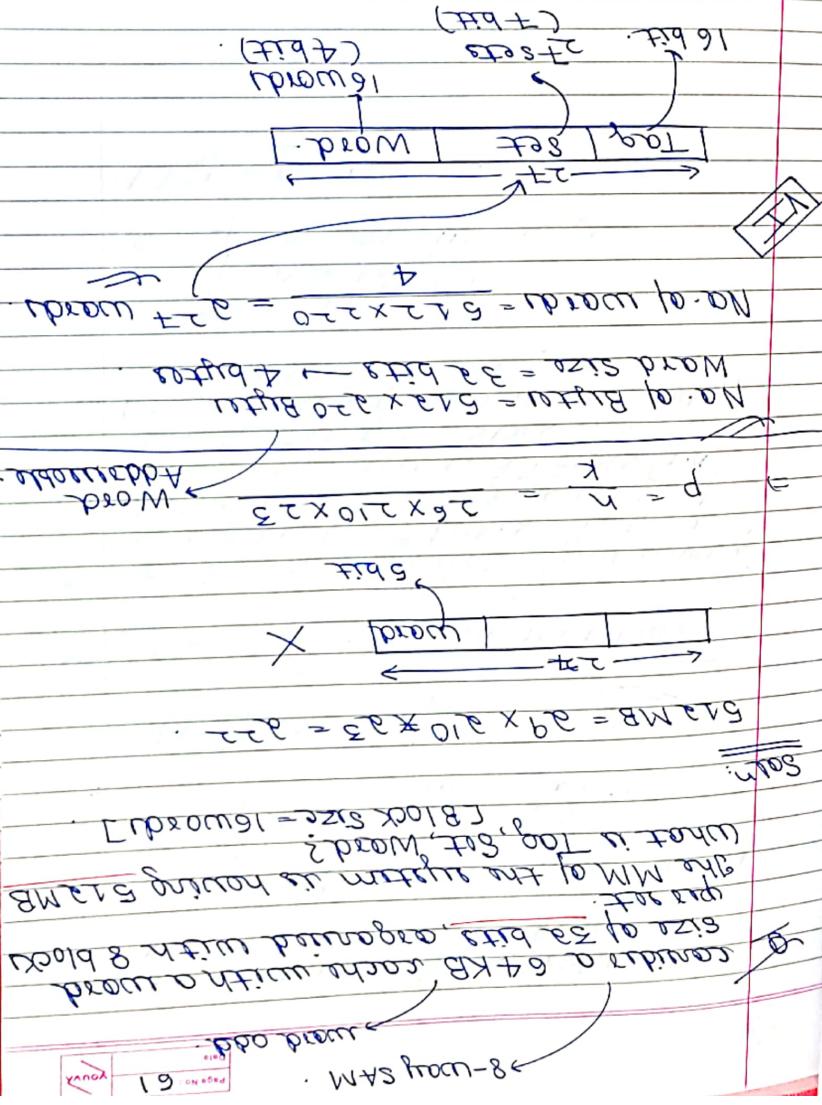
Solution



What is Tag, unit, Byte:
The MMU of the system consists of 256 units of 32 bits each.
Consider a direct mapped cache with 128 units → 128 bytes.
(Byte Addressable)

Date: _____

Page No. _____



$$\begin{aligned}
 & \Rightarrow P - N + \log_2 K \\
 & P - N + \log_2 K + \log_2 K \\
 & \therefore P - N - M - K + \log_2 K \\
 & \text{No. of bits} = 2^{N-M-K-\log_2 K} \\
 & \frac{\text{No. of words}}{2^M} = \frac{2^N}{2^{N-M}} \\
 & \text{No. of words} = 2^{N-M} \\
 & \text{No. of bytes} = 2^N \\
 & \text{No. of words} = 2^M \\
 & \text{No. of bytes} = 2^{M \text{ bytes}} \\
 & \text{No. of words} = 2^{(P-M) \text{ bits}} \\
 & \text{No. of bytes} = 2^{(P-M) \text{ bits}} \\
 & \text{word address} = \frac{P-M}{2^M} \\
 & \text{word address} = \frac{P-M}{2^M} = 2^{(P-M) \text{ bits}}
 \end{aligned}$$

(a) $(P - n - m - w + \log_2 K)$
 (b) $(P - n + \log_2 K)$
 (c) $(P - n - m - w - \log_2 K)$
 (d) $(P - n - \log_2 K)$

What is the tag size?
 For a k -way set associativity, the size of each cache block is m words.
 Cache memory is in n bytes, capacity of which is p bytes. The word associativity is a power of 2 .
 The size of a physical address is the sum of word address, word addressee, and tag.

GATE 2018 [2M]

$$\text{No. of bits} = 2^{10} = 8 + 4 + 2 + 1$$

16

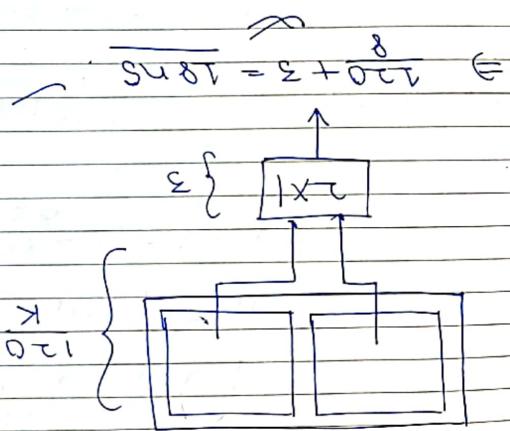
$$\text{No. of blocks} = 2^{14} = 16$$

$$\text{No. of words} = 64 \times 2^{10} = 2^{14} \text{ words}$$

$$\text{word size} = 32 \Rightarrow 4 \text{ bytes}$$

$$\text{No. of bytes} = 64 \times 2^{10} \text{ bytes}$$

55
105



~~Link~~

~~a-SAM~~

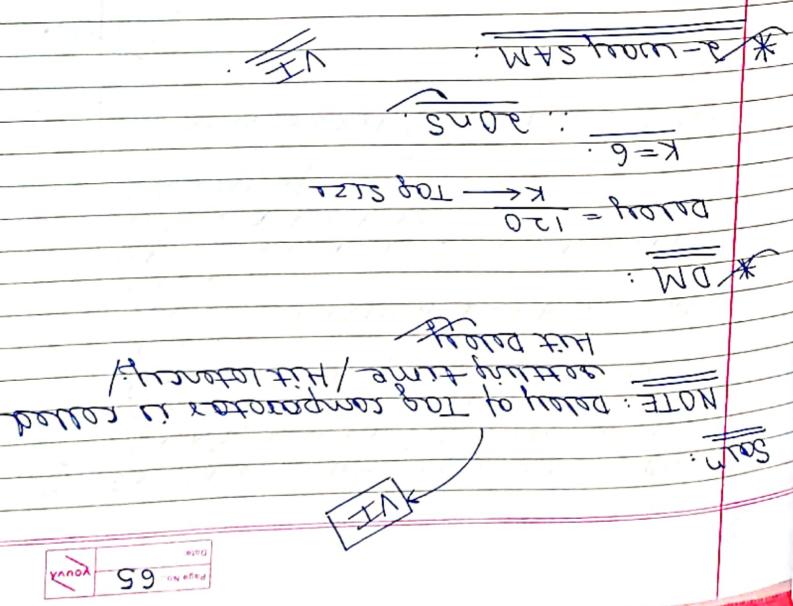
~~20ns~~

~~DM~~

~~Hit~~

~~Note: Ratio of Tag comparators is 1:16~~

~~65~~
~~Yours~~



~~K = tag size.~~

~~What is hit ratio from both~~

~~DM and SAM?~~

~~Max of 120 ns.~~

~~Cache access time is 6 bits. The delay of 8 bit tag, which the direct mapped A 3-way set associated cache has~~

~~tag comparators is 120 ns, while~~

~~cache access is 6 bits. The direct mapped~~

~~Cache → 28 [Index 0 ←]~~

~~28~~

~~Cache → 32 [Index 4 ←]~~

~~32~~

~~SAM~~

~~How many bits required for tag.~~

~~This addressable and uses 32 bit address.~~

~~is 16 bits. Allowing that, the MMU~~

~~Access of size 16KB, the cache block size~~

~~A certain processor will fully associate~~

~~with addresseable and uses 32 bit address.~~

~~HW~~

~~GATE-2019~~

~~25~~

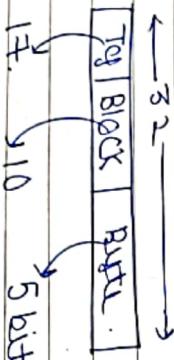
Note
 Hi
 consider a cache organization with
 32 KB; a way size
 32 byte block size, but DM
 Another one is of same size, but DM
 cache.
 The size of address is 32 bits in both
 cases.

Latency is 0.2 ns
 while a k bit tag comparator has
 latency of $k/10$ ns ✓

What is the hit latency in DM, SAM.

SAM

* DM



2-way SAM
 $\Rightarrow P = \frac{1}{K} = \frac{1}{2}$

opt. LRU (Least Recently Used)

1-way SAM
 $\Rightarrow P = \frac{1}{K} = \frac{1}{1}$
 a block with no address for the longest time. (inactive block).

18 bit → 5 bit → 5 bit ✓

$$\Rightarrow 17/10 = 1.7 \text{ ns} \rightarrow \text{DM}$$

$$18/10 = 1.8 + 0.6 = 2.4 \text{ ns. } \cancel{\text{DM}}$$

* CACHE REPLACEMENT POLICIES

- which one to replace
- replace cache block
- update possibility ↓



* Random

- No specific criteria

* FIFO (First in First out)

- oldest block replaced
- A block which enters last, leaves first

Best

2017	
NA of capacity $M_{NA} = 4$	
NA of concurrent $M_{NA} = 5$	
NA of computation $M_{NA} = 8$	
$HR = 3/20, MR = 17/20$	
$NA \text{ of } M_{NA} = 17$	
$NA \text{ of } M_{NA} = 3$	
(b) 18 not in the cache.	
$(24, 2, 82) \oplus$	
3. capacity miss.	
a. concurrent miss ($0, 16, 17, 35, 18, 30$)	
4. computation miss ($3, 5, 2, 63, 30, 9, 26$)	
63	
6	30
5	
4	20
3	182
2	$182 \oplus 82$
1	$182 \oplus 2517$
0	861624
99	YODA

Solutions	
63	$CL(M) = M/8 =$
6	
5	
4	
3	
2	218282
1	182517
0	861624
3	
2	
1	
0	
18	
3	
20	
25	
17	
35	
16	
1	
24	
2	
82	
3. 5, 2, 63, 30, 9, 26, 17, 35, 18, 30,	
all 8 bits set in the address:	
consider a direct mapped cache with	
8 blocks ($0 \rightarrow 7$) giving many blocks	
which of the following memory blocks	
will not be in the cache at the end	
of the simulation?	
NOTE: No interleaved strategy for DM;	
A block with four transfers.	
4. LRU (Least Frequently Used)	
99	YODA

Consider a fully associative cache with 8 blocks and the following memory block requests:

4, 3, 25, 8, 19, 6, 15, 8, 26, 35, 45, 2, 3, 10, 16.

NO

* If LRU is used:

The memory block 7 is in 5 cache block?

0	4, 45	0	4, 45.
1	3, 22	1	3, 22.
2	25	2	25
3	8	3	8
4	19, 3	4	19, 3.
5	7	5	7
6	16	6	16.
7	35	7	35.

0, 25, 1, 4, 3, 8, 13, 3, 15, 9, 2, 16, 12, 9, 6, 3, 8, 4, 2.

Consider a 4-way set associative cache with 16 blocks & the following memory block requests:

$$HR = 5/17, \quad MR = 12/17.$$

LRU is used:

Which of the following memory blocks will not be in the cache at the end of the sequence?

(a) 3

(b) 8

(c) 12, 9

16, 216.

$$\text{Ans} \quad p = n/k = 16/4 = 4 \quad \text{sets}$$

→ Cacheability: 4, 3, 25, 8, 19, 6, 16, 35

Conflict Miss: NO

[No conflict miss in associativity mapping]

S1
20

S-1

S2

$$SL(M) = M/4 =$$

Compulsory - {0, 255, 1, 4, 3, 8,
133, 159, 216, 12, 63, 73}

Conflict Miss = {48, 32, 92, 155}

No. of hits = 1

No. of miss = 16.

HR = 1/17, MR = 16/17.

S-0 0, 48
432

8
216 92

133

129

73

155

3

159

63

Capacity Miss

Page No:
Date:

Page No: 73
Date: 10/07/2023

- All rules are applicable ✓

Q consider a 2 way set associative cache with 8 blocks & following memory block req

0, 3, 5, 9, 7, 0, 16, 55

If LRU used, what memory blocks will be present in the cache at the end of the reference.

Set 0 0
16

Set 1 3
5
9
7

No. of sets = $\frac{8}{2} = 4$

0/4 = 0
3/4 = 3

K = n/p

away → Set 0 0
16

Set 1 5
9

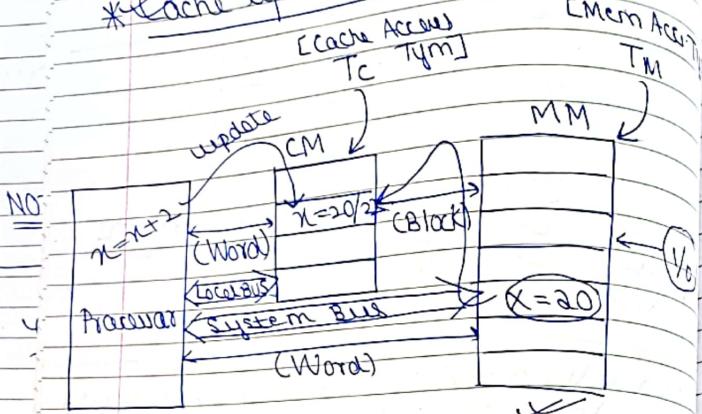
Set 2 3
5
7

Set 3 5
7

= 0, 5, 7, 9, 16, 55.

∴ 3 misses Ans.

* Cache Updation Policies



- When processor modify something in cache, it also has to update in MM.

(\hookrightarrow 2 techniques)

\rightarrow write-through policy

\rightarrow write-back policy.

• T_g : Block Transfer time.

\rightarrow The updation in MM is done using:

(i) write-through policy:

- Simult.

- Simultaneous update of a word in both cache and MM.

\rightarrow Consistency (Cache Coherence) is resolved.

• P takes Active page from MM via System Bus.

Page No.: 77 Date: youva

overhead at bus control

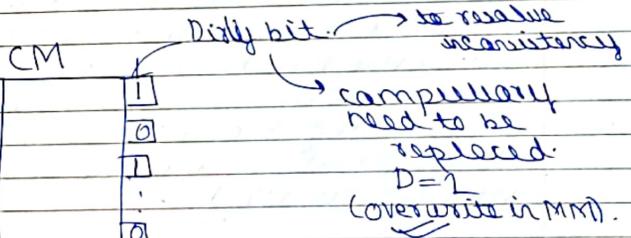
\rightarrow Always i/o has correct data.
 \rightarrow The updation of a word takes T_m time. ($T_m > T_c$)
 \rightarrow updation is done with increased bus traffic and overhead. Effective for less updations.

(ii) Write Back Policy \rightarrow processor involves after replacement.

\rightarrow Processor will modify the word only in cache.
 \rightarrow updation in MM is postponed until the modified block is replaced.
 \rightarrow consistency can't be resolved.
 \rightarrow $CM \rightarrow 22, MM \rightarrow 20$.

\rightarrow not a big issue, \because replacement occurs quickly \leftarrow

\rightarrow Effective for more updations. (less overhead)



- When proc does some updation on cache.

\rightarrow unwanted transfer.

~~C~~
 A block which is modified in the cache is called dirty block. Its dirty bit is set.

- 2 levels : start.

→ A block is replaced only if its dirty bit is 1.

NOTE :

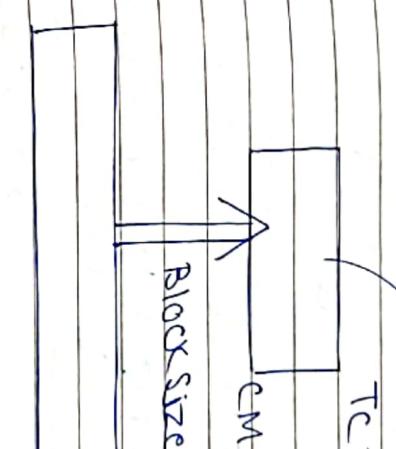
- (i) Update in cache means writing back and vice-versa.
 (ii) If no. of dirty blocks is 0, both would through and write back will have same performance.

Consider a 64 word cache and MM both are divided into a block of 16 words. The access time of cache and MM is 10ns and 50ns. When a miss occurs, the associated block is obtained from MM to cache. For both read & write operations. The hit ratio for read operations is 80% and write operations is 90%. Let then $H_R = 80\%$, $H_W = 90\%$.

$$T_{avg} = F_R \cdot T_{avg,R} + f_W \cdot T_{avg,W}$$

$$T_{avg} = 80\% \cdot 10 + 20\% \cdot 50$$

$$T_{avg} = 80\% \cdot 10 + 20\% \cdot 50$$



MM.

$$T_{avg} = F_R \cdot T_{avg,R} + f_W \cdot T_{avg,W}$$

$$\Rightarrow T_{avg} = 80\% \cdot 10 + 20\% \cdot 50$$

$$H_R * T_c + (1 - H_R)(T_B + T_c)$$

$$\Rightarrow T_{avg} = 0.8 \times 10 + 0.2 \times (200 + 10)$$

$$\Rightarrow 8 + 160 = 170 \text{ ns}$$

T_{avgW}

Score 1:

$$\boxed{T_{avgW} = H_R T_C + (1-H_R) T_M}$$

No.

$$T_{avgW} = H_R T_C + (1-H_R) \cdot (T_B + T_C)$$

$$= 0.9 * 10 + 0.1 * (820)$$

$$\Rightarrow 9 + 82 = 90 \text{ ns}$$

~~white through policy:~~

→ (upstation)

$$T_{avg} = 0.6 * 170 + 0.4 * 130$$

$$\boxed{T_{avg} = 154 \text{ ns}}$$

* throughput:

No. Amount of work done/unit time.

VI

$$T_{avgW} = H_W T_M + (1-H_W) (T_B + T_M)$$

Hit while waiting
in both CW & MM.

Block Transfer from MM to CW

$$\Rightarrow T_{avgW} = 0.9 * 50 + 0.1 * (800 + 50)$$

$$\Rightarrow 45 + 85 = 130 \text{ ns}$$

Score 2:

$$T_{avgW} = H_W T_M + (1-H_W) \cdot T_M$$

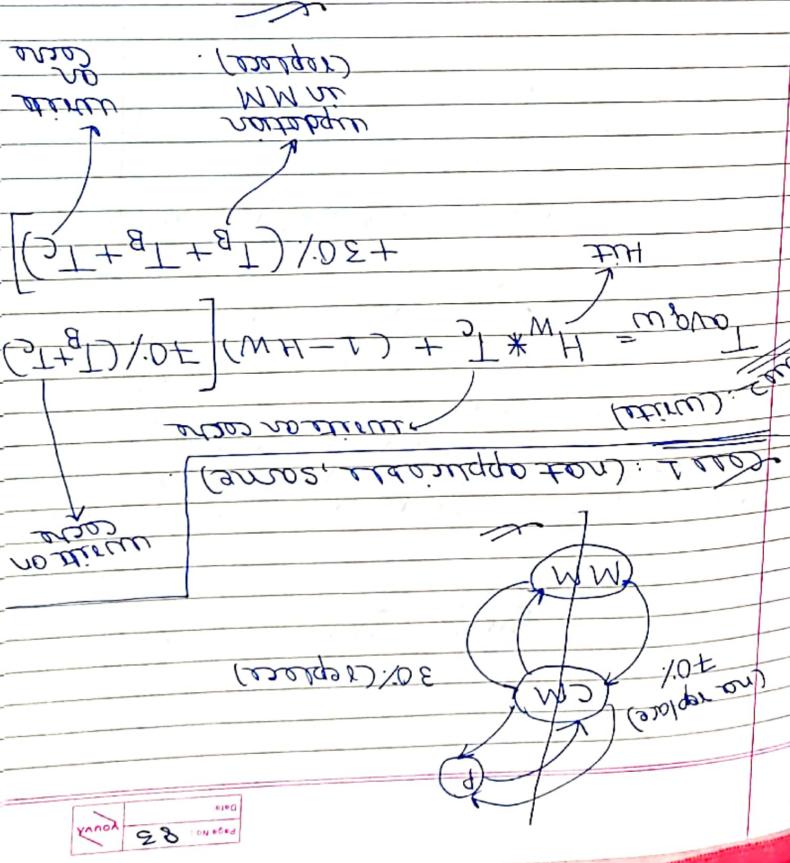
$$\Rightarrow H_W T_M + T_M - T_M H_W = T_M$$

constant highway update in TM.

VI:

$$f_C = \frac{1}{t_C}$$

→ one clock cycle time



$$\text{avg} = f_R * T_{avg} + f_M * T_{avg}$$

~~$\text{avg} = H^R * T_c + (1-H^R) * [f_B * (T_B + T_C) + f_M * (T_B + T_C)]$~~

$\text{avg} = H^R * T_c + (1-H^R) * [f_B * T_B + f_M * T_C]$

(apply to each block)

$T_{avg} = H^M * T_c + (1-H^M) * [f_B * T_B + f_M * T_C]$

(apply to each block)

At any point of time, 30% of cache blocks

and throughputput?

In the above problem, if we take back

policy is used, what is our access time

and throughputput?

$\text{Throughput} = \frac{1}{\text{Latency}}$

$\text{Latency} = \frac{154}{109} \text{ microseconds}$

$\text{Latency} = \frac{154}{109} \times 10^{-9} \text{ seconds}$

No.

lecture 5

* Cache Memory & Arrays

N

- LOR

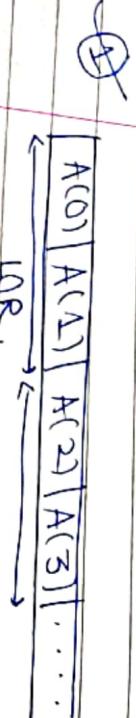
Consider an array $A[100]$, each element occupies 4 words. A 32 word cache is used and divided into a 8 word block. What is the hit ratio for the following statements?

F) $\text{for}(i=0; i < 100; i++)$

$A[i] = A[i] + 10;$

R & W

→ ~~extra memory structural (Array)~~



LOR.

A(0) is compul. miss
A(1), A(2) is brought to CM by LOR.

$$\Rightarrow \boxed{\frac{HR = 3}{4} = 75\%}$$

e.g. w, BS = 3 elements
HR = $\frac{5}{6} > 75\%$



R

W

A(0) → M H

H

A(1) → H H

H

A(2) → H H

H

word

2 word

4 blocks

4 blocks

4 blocks

4 blocks

4 blocks

4 blocks

for($i=0; i < 100; i++$)

$\{$

$\}$

Read.

① $\underbrace{[A(0) | A(1) | A(2) | A(3)] \dots}$

② Return in each block.

③ (array read)

NO R

$A(0) \rightarrow M$ → spot pattern

$A(1) \rightarrow H$

$A(2) \rightarrow M$

$A(3) \rightarrow H$

② MM

same

④ Array Structure (AS)

→ Matrix

e.g.:
$$\begin{bmatrix} 0 & 1 & 2 \\ 3 & 4 & 5 \\ 6 & 7 & 8 \end{bmatrix}$$
 $\begin{matrix} \\ \\ \end{matrix}$ 3×3

→ Row Major Order (RMO)

$$HR = \frac{1}{2} = 50\%$$

R/W pattern affects HR

$$\begin{bmatrix} 3 & 4 & 9 & 8 & 2 & 16 & 5 & 1 & 7 \\ 0 & 0 & 0 & 1 & 0 & 11 & 12 & 20 & 21 & 22 \end{bmatrix}$$

$$[00 \ 01 \ 02 \ \dots \ 0n-1 \ 10 \ 11 \ 12 \ \dots \ 1n-1]$$

Consider an Array $A[10][5][10]$. Each element occupies 4 words. A 32 word cache is used and divided into 8 word-blocks. What is the hit ratio for the following statement?

for ($i=0; i < 10; i++$)

{
 for ($j=0; j < 10; j++$)

$A[i][i][j] = A[i][i][j] + 10;$

}

}

}

→ Column Major Order (CMO)

(iii) By CMO:

3	8	5	4	12	1	9	6	11
00	10.	20	01	11	21	02	12	22

Not ~ $\underline{\underline{00 \ 10 \ 20 \ 30 \ \dots \ 01 \ 11 \ 21 \ 31 \ \dots \ n-1 \ n-1}}$

④ Memory Structure

② MM
} 8 words / 2 elements

A(0)(0)	A(0)(1)
A(0)(2)	A(0)(3)
A(1)(0)	A(1)(1)

$$HR = \frac{1}{2} = 50\%$$

Memory structure changes HR

for ($i=0; i < 10; i++$)
 for ($j=0; j < 10; j++$)
 $AC[i][j] = AC[i][j] + 10;$

A(0)(0)	A(0)(1)
A(1)(0)	A(1)(1)
A(2)(0)	A(2)(1)
A(3)(0)	A(3)(1)

CM

MM

③

R W
(Both read / writing) ~~By RMO~~

By RMO

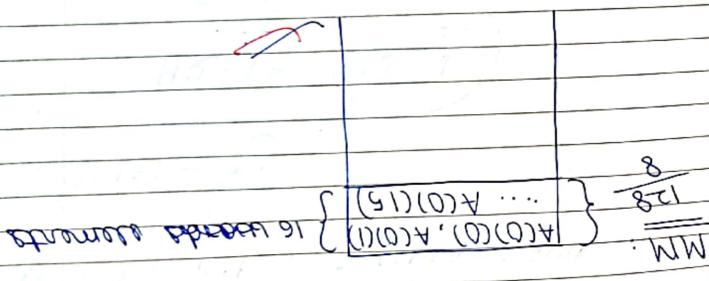
A(0)(0) → M H
 A(0)(1) → H H
 A(0)(2) → M H
 A(0)(3) → H H
 rep. pattern

$$HR = \frac{3}{4} = 75\% \quad \checkmark$$

$$HR = \frac{1}{2} = 50\% \quad \checkmark$$

(c) 16384
 (b) 2048
 (a) 0
 (d) 262144

PA, PA is calculated independently from same
 address of cache miss of PLU M1.
 i, j, k are in registers.
 initial state.



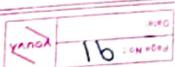
23
 24 = 24

R1 RMO : (default)

128 Array :

Soln:

100
 110
 110
 100
 $x+ = A[j][i]$
 $y[i] (i=0 : i < 512, i++)$
 $x[i] (i=0 : i < 512, i++)$



[Answers go with RMO]

PA:

g(i) M_i = ?

and PA is M1.

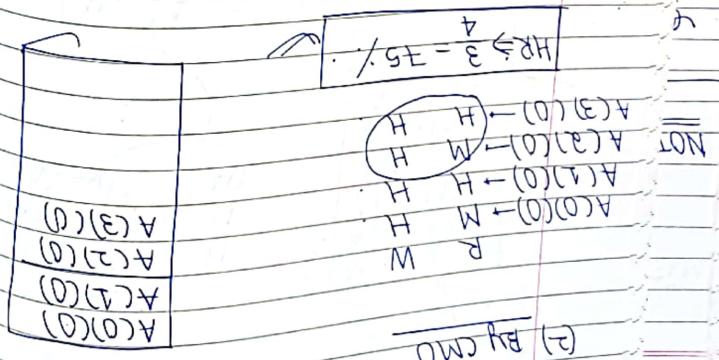
$j = 0 : j < 512, j++$
 $j = 0 : j < 512, j++$

(a) PA

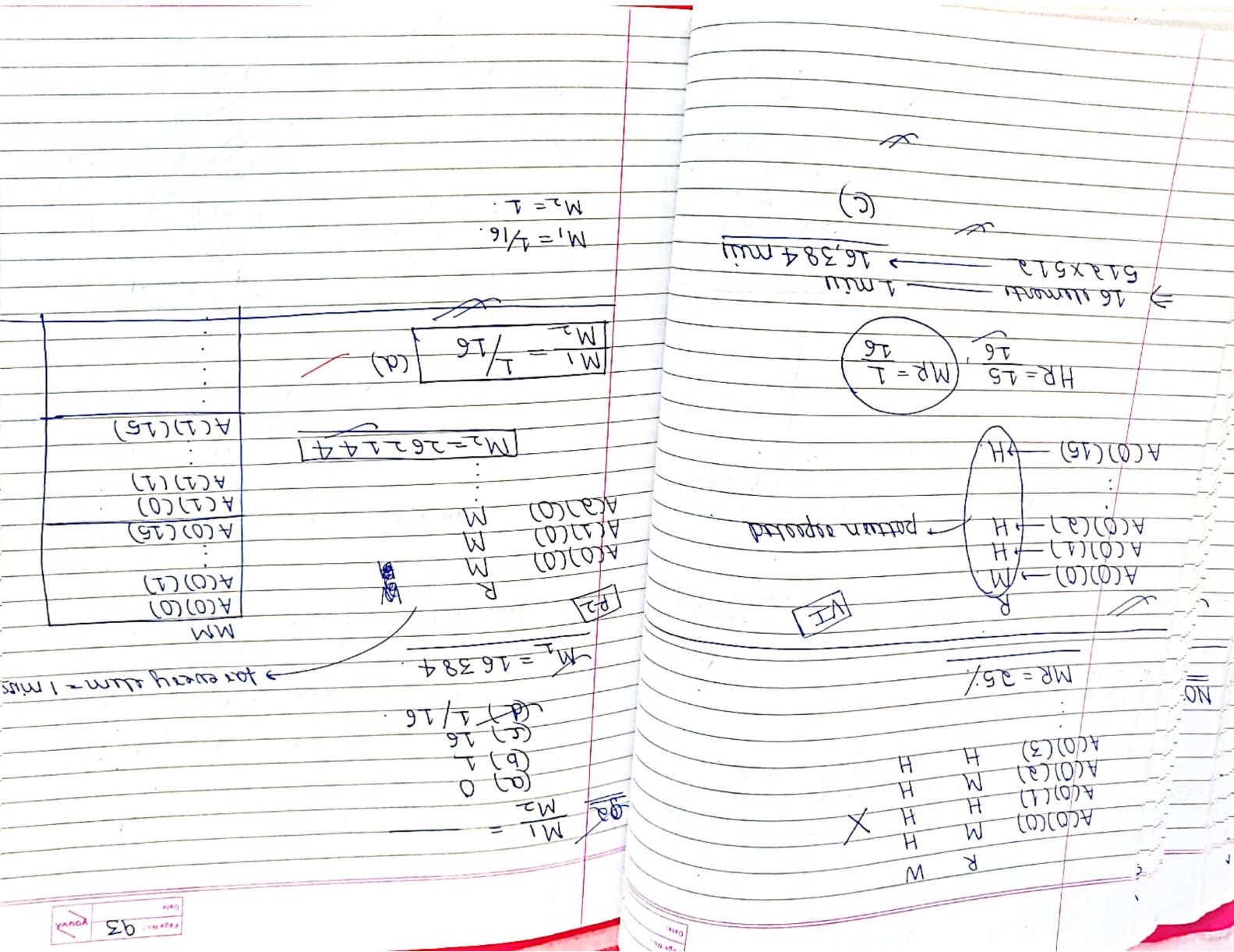
Cache Size: Suppose A is a 2D Array of size 512×512 . Each element occupies 8 Bytes. For the following program

contains a 32 KB cache with 128 Bytes

NOTE: How array is accessed also affects HR



(a) R1 RMO



SECONDARY STORAGE

NO

Magnetic Disk

- disks placed one above other HR
- A magnetic disk is a very thin circular metal plate. Data on the disk surface is organized as:

\downarrow increases from outer to inner track \uparrow

- is max at innermost track.
- Data transfer rate of the disk.

$$D = \text{No. of Bytes / sec}$$

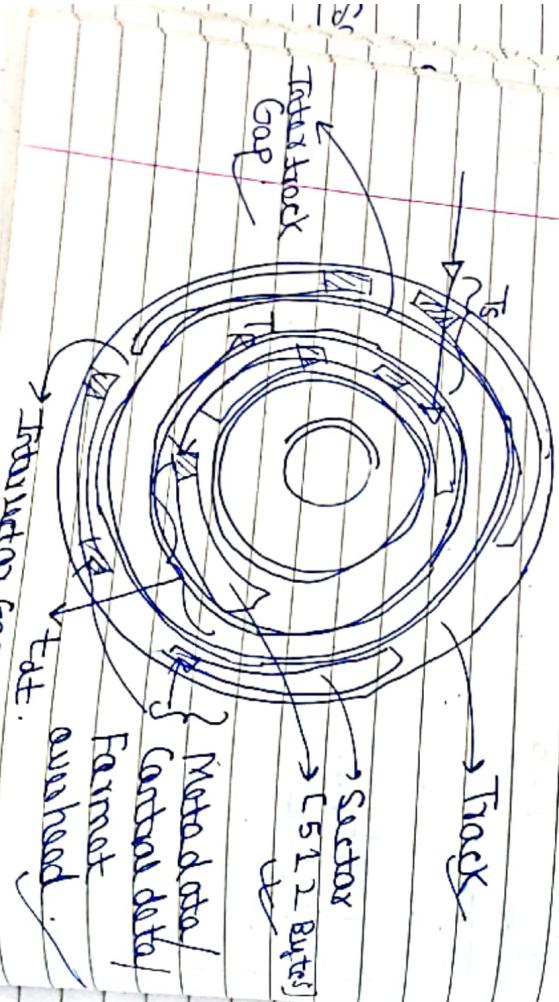
"RPM"

- D depends on RPM (rotations per minute).

- (Each rotation covers 1 Track)

Seek Time (Ts)

- Time to move read/write head to the desired track number.



- A set of concentric circles called Tracks.
- Each track holds same no. of manageable units, called Sectors.
- Basic unit of Access is a Sector.

- Disk Space without Format overhead is Free/Unallocated disk Space.
- Storage density on the disk is (Active)

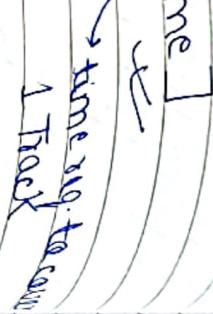
$$\boxed{D = \text{No. of Bytes / cm}}$$

i. Rotational latency (t_R) [Avg. latency
the time to move R/W Head to disk
sector beginning]

$$[t_R = \frac{1}{2} \text{ rotation time}]$$

= NO.

Access Time of Disk

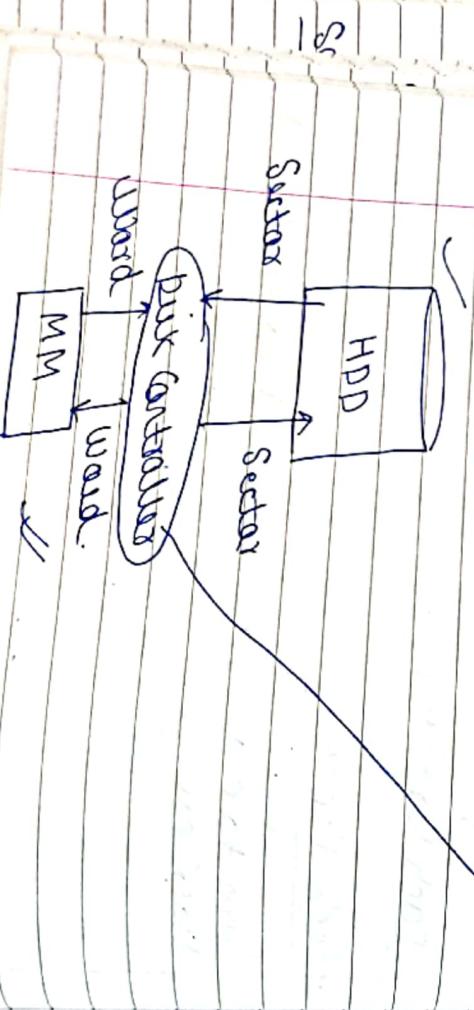


Per Sector per Disk Access

Average Access time of the Disk

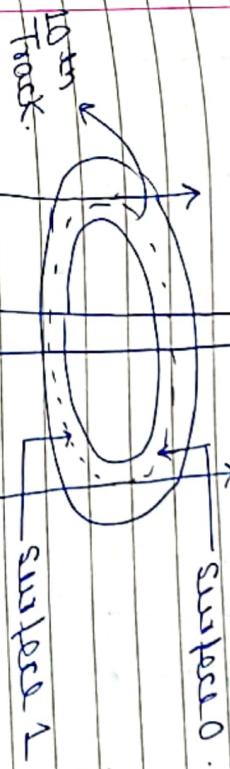
$$t_{avg} = t_S + t_R + t_d + t_{disk movement} \\ (\text{sector})$$

per sector.



* Multiple disks can be organised as:

Spindle



(i) Single Sided / double sided Disks

By default.

50 → 10 surfaces (by default)

3 characteristics:

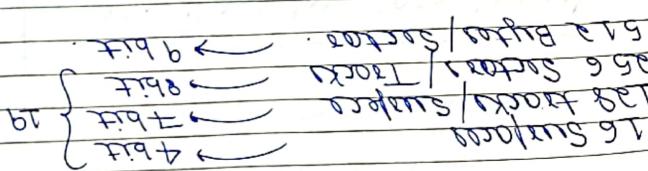
$$N_{\text{no. of sectors}} = 16 \times 138 \times 256$$

(g) How many bits required to address a sector?

$$\Rightarrow 256 \text{ MB} \rightarrow$$

$$C = 24 \times 24 \times 28 \times 29$$

(h) What is the storage capacity of the disk?



(i) A disk system has the following specifications.

VI

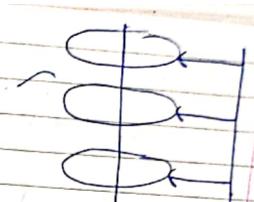
[No. of cylinders = No. of Tracks.]

A virtual set of all the tracks of some

blocks access time & cost.

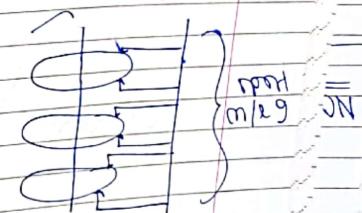
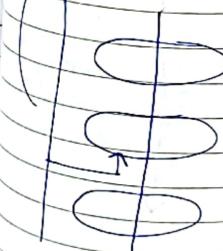
Magnetic Raftte Disk mostly used

Page No.: 99
Date:



(a) Single Platter / Multiplte Platter Disk

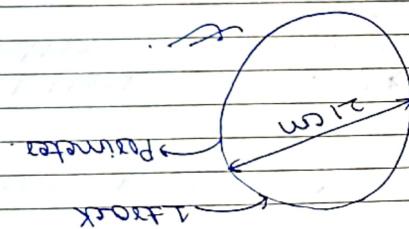
- * Cyl → Surface
- * Track → Surface
- * Disk → Surface
- * Access Time ↓
- * Hard disk capacity ↓
- * [N = No. of Surfaces]
- * Contains n Tracks
- * Each rotation takes one track
- * All surfaces
- * One cylinder holds for
- * (magnetic)
- * R/W Head



(b) Fixed R/W Head / Moving R/W Head Disk

- * Head
- * NC
- *

(ii) The diameter of the innermost track is 28 cm with a maximum storage density of 128 KB/cm². What is the track storage capacity?



$$d_{max} = \frac{66}{\pi} \times 256 \times 512 = 29 \times 28$$

$$66 \text{ cm} \rightarrow 1 \text{ track}$$

$$\text{Radius} = \pi D = \frac{\pi}{2} \times 21 = 66 \text{ cm}$$

$$D = \text{No. of bits/cm}$$

ANSWER: (i) The diameter of the innermost track is 21 cm. What is the maximum storage?

$$= 256 \text{ MB} - 32 \text{ MB} = 224 \text{ MB}$$

$$= 16 \times 138 \times 256 \times (512 - 64) \text{ Bytes}$$

$$219 \times 2^6 = 2^{15}$$

(ii) In the above disk pack, format word size is 64 bytes/sector. What is the formatted disk size (in sectors)?

$$= 224 \text{ Bytes} \rightarrow 16 \text{ MB}$$

$$= (24 + 8) \times 32 \text{ Bytes}$$

$$= 16 \times 138 \times 256 \text{ Sectors (219)}$$

(iii) In the above disk pack, format word size is 32 bytes/sector. How much storage is lost due to fragmentation?

19 bits required

$$\Rightarrow 16.879 \text{ KB} \quad (\text{Track Capacity})$$

$$\Rightarrow 138,380 \text{ bits} \rightarrow \text{Inner track}$$

$$x \quad 69.14 \text{ cm}$$

$$2000 \text{ bits} \quad 1 \text{ cm}$$

$$\frac{1}{2\pi} \times 22 = 69.14 \text{ cm}$$

So?

(Q) capacity of the disk pack?
 disk is 0.25 m. What is the storage
 bits/cm and spacing among the
 The maximum storage density is 2000
 diameter of 22 cm and outer of 33 cm.
 area available should have an inner
 A disk system has 10 surfaces, the average

$$\Rightarrow D = 16 \times 3.5 \text{ MBPS}$$

$$\text{A rotation} = 16 \text{ tracks}$$



in fixed I/U Head

$$\Rightarrow 3.5 \text{ MBPS}$$

$$16.66$$

$$\Rightarrow 1000 \times 128 \text{ Kbps}$$

$$16.66$$

$$D = 1000 \times 256 \times 512$$

Page No. 103
Date _____

$$16.66 \text{ msec} \rightarrow \text{ATack}$$

$$(256 \times 512)$$

$$1000 \text{ msec}$$

$$\Rightarrow 16.66 \text{ msec}$$

$$3600$$

$$60 \times 10^3$$

$$3600 \text{ rotation}$$

$$60 \times 10^3 \text{ msec}$$

$$3600 \text{ rotation} = 1 \text{ track}$$

Head

Moving

I/U

$$\Rightarrow 22 \times 3600 \text{ bytes} = 60 \text{ s}$$

(Ans) Why disk system has rotational spread
 of 3600 rpm what is the data
 transfer rate?

$$\text{Track capacity} = 176 \text{ KB}$$

$$1 \text{ track} = 176 \text{ KB}$$

$$1 \text{ cm} = 2 \text{ KB}$$

$$\text{Periphery} = \frac{1}{2\pi} \times 28 = 8.9 \text{ cm}$$

$C = \text{no. of surfaces} * \text{no. of tracks} *$
 $\text{no. of sectors per track capacity}$

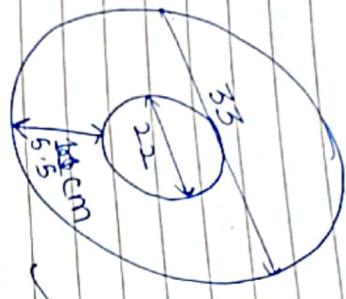
✓

$$C = 19 * 16.879 * 220$$

$$\Rightarrow 7068.9 \text{ MB}$$

$$C = 68.90 \text{ MB}$$

NO



width of storage = w cm.

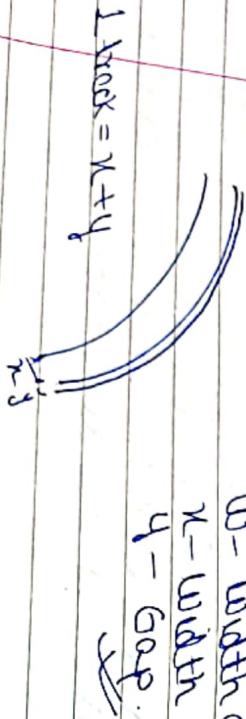
$$\Rightarrow \frac{33}{2} - \frac{22}{2} = 5.5 \text{ cm}$$

$$4200 \text{ rotation} = 60 \text{ s}$$

$$1 \text{ rotation} = 0.014 \text{ s.}$$

→ movable

$$1 \text{ rotation: } 1 \text{ track} \rightarrow 16.879 \text{ KB} \rightarrow 14.28 \text{ ms}$$



$$16.879 \text{ KB} = 0.014$$

$$\therefore D = \frac{1000 * 16.879}{14.28} = 1.15 \text{ MBPS}$$

$$\begin{aligned} \text{No. of tracks} &= \frac{w}{w} \\ (\text{No. of Gap}) &= \frac{w}{k} \\ (\text{No. of Gap}) &= \frac{w}{k} \cdot (\text{Gap} \leftarrow \text{Track}) \end{aligned}$$

$$\text{Fixed: } 19 \times 1.15 = 21.91 \text{ MBPS}$$

(Q) The disk system has rotational speed
of 4200 RPM. What is the data transfer
rate?

X → not given

Page No. 105
Date _____

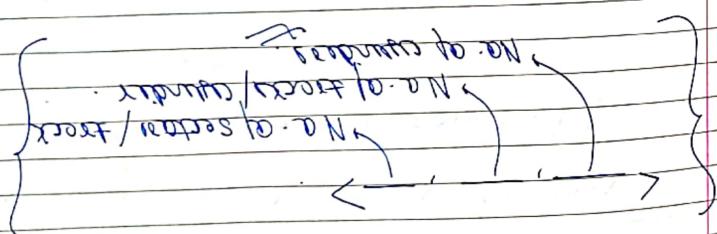
(e) 505037

$$400 \times 20 \times 63 = 504,000$$

<400, 16, 29>

$$16 \times 63 = 1008$$

504,000 +



<300, 17, 17>

<300, 17, 17>

<300, 16, 25>

<300, 17, 17>

<300, 16, 25>

1 cylinder \rightarrow 1008 sectors
1 cylinder \rightarrow 20 track surfaces
1 track \rightarrow 63 sectors

1008 sectors

1000 cylinders \rightarrow 1000 tracks

(d) 505038

(e) 505036

(f) 505035

Q8. Ques: $(400, 16, 29)$ corresponds to

Ans: (a, b, c)

$(a, b, c) \leftrightarrow (0, 0, 4)$

$(0, 0, 0) \leftrightarrow$ cylinder, 0 cylinders

$(0, 0, 1) \leftrightarrow$ cylinder number.

where: $c =$ cylinder number

$a =$ cylinder number (track surface)

$b =$ cylinder number (sector)

Q9. A Hard disk has 63 sectors/Track/Surface. Total number of cylinders is given as:

A Hard disk has 63 sectors and 1000 cylinders.

Given address of a sector is given as:

Q10. A Hard disk has 63 sectors/Track/Surface. Total number of cylinders is given as:

Given address of a sector is given as:

Q11. A Hard disk has 63 sectors/Track/Surface. Total number of cylinders is given as:

Given address of a sector is given as:

Q12. A Hard disk has 63 sectors/Track/Surface. Total number of cylinders is given as:

Given address of a sector is given as:

Q13. A Hard disk has 63 sectors/Track/Surface. Total number of cylinders is given as:

Given address of a sector is given as:

Q14. A Hard disk has 63 sectors/Track/Surface. Total number of cylinders is given as:

Given address of a sector is given as:

Q15. A Hard disk has 63 sectors/Track/Surface. Total number of cylinders is given as:

Given address of a sector is given as:

$$TS = 10 \text{ msec}$$

$$TR = 5 \text{ msec}$$

$$X = 60 \times 10^3 = 10 \text{ msec}$$

$$6 \times 10^3$$

$$6000 \text{ rot} \xrightarrow[60 \times 10^3 \text{ msec}]{1 \text{ rotation}} 100 \text{ msec}$$

S

R

What is the total time taken to load all the bits? It takes 100 msec to read all the bits. What is the total time taken at 6000 rpm. What is the time per bit? If we take 10 msec for a standard sector. Then seek time to a standard track is 10 msec. So effectively within each library frame there will be a 10 msec seek time. An application loads 100 bits during

$$q = 2.5 \times 10^6$$

$$\Rightarrow 2.5 \times 10^6 \text{ bits/s}$$

25

$$X = 62500 \times 1000$$

$$X = 2500 \text{ msec}$$

$$X = 2500 \text{ bits}$$

25

$$S = n \cdot 0.1 \text{ bits} \quad q =$$

$$P = 12.5 \text{ msec}$$

$$P = \frac{1}{2} \times 10 \text{ msec} \rightarrow 12.5 \text{ msec}$$

40

$$2400$$

$$X = 80 \times 10^3 \xrightarrow[25]{1 \text{ rotation}} 100 \text{ msec}$$

$$100 \text{ rotation} \xrightarrow[60 \times 10^3 \text{ msec}]{2} 100 \text{ msec}$$



What is P/q?

$$\text{Data Transfer Rate} = q \text{ bits/s}$$

$$\text{Avg Latency} = p \text{ msec}$$

$$\text{Disc Speed} = 2400 \text{ rpm}$$

$$\text{Track capacity} = 63500 \text{ bits}$$

$$\text{No. of tracks} = 300$$

Considering a rotation moving arm disk storage with 1/3 sec head for the reading writing operation. (marked)

$$Pavg = 11.5 + 10 = 21.5 \text{ msec}$$

$$\frac{1}{2} \times 20 = 10 \text{ ms}$$

$$\frac{1}{2} \times 10 \text{ msec}$$

$$\frac{1}{2} \times 60 \times 10^3 = 30 \text{ msec}$$

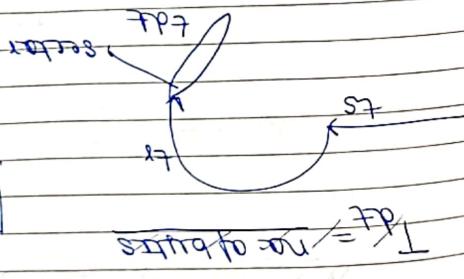
$$T_{avg} \rightarrow 20.5 \text{ msec}$$

$$T_{avg} = 20 + 0.4 + 0.1$$

$$X = \frac{1000}{1 \times 10^3} = 0.1 \text{ msec}$$

$$X = \frac{10 \times 20}{2 - 10} = \frac{20}{2 - 10}$$

$$10MB \rightarrow 1s (10^3 \text{ msec})$$



$$1K = 1000$$

time

$$size$$

$$T_{avg} = T_s + T_l + T_d + T_f$$

$$T_s = 10 \text{ msec}$$

$$T_d = 10 \text{ msec}$$

$$= 20 + 0.4 +$$

113
1000

$$\frac{3000}{20 \times 10^3} = 0.015 \text{ sec}$$

$$3000 \times 10^3 = 60 \times 10^3$$

What is avg access time?

Find out:

$$10 \text{ MBPS}$$

$$ts = 10 \text{ msec}$$

$$td = 0.4 \text{ msec}$$

$$1 \text{ RW Head}$$

$$3000 \text{ rpm and}$$

$$1 \text{ KB/sector}$$

$$512 \text{ sectors/track}$$

$$128 \text{ tracks/ platter}$$

$$16 \text{ platters}$$

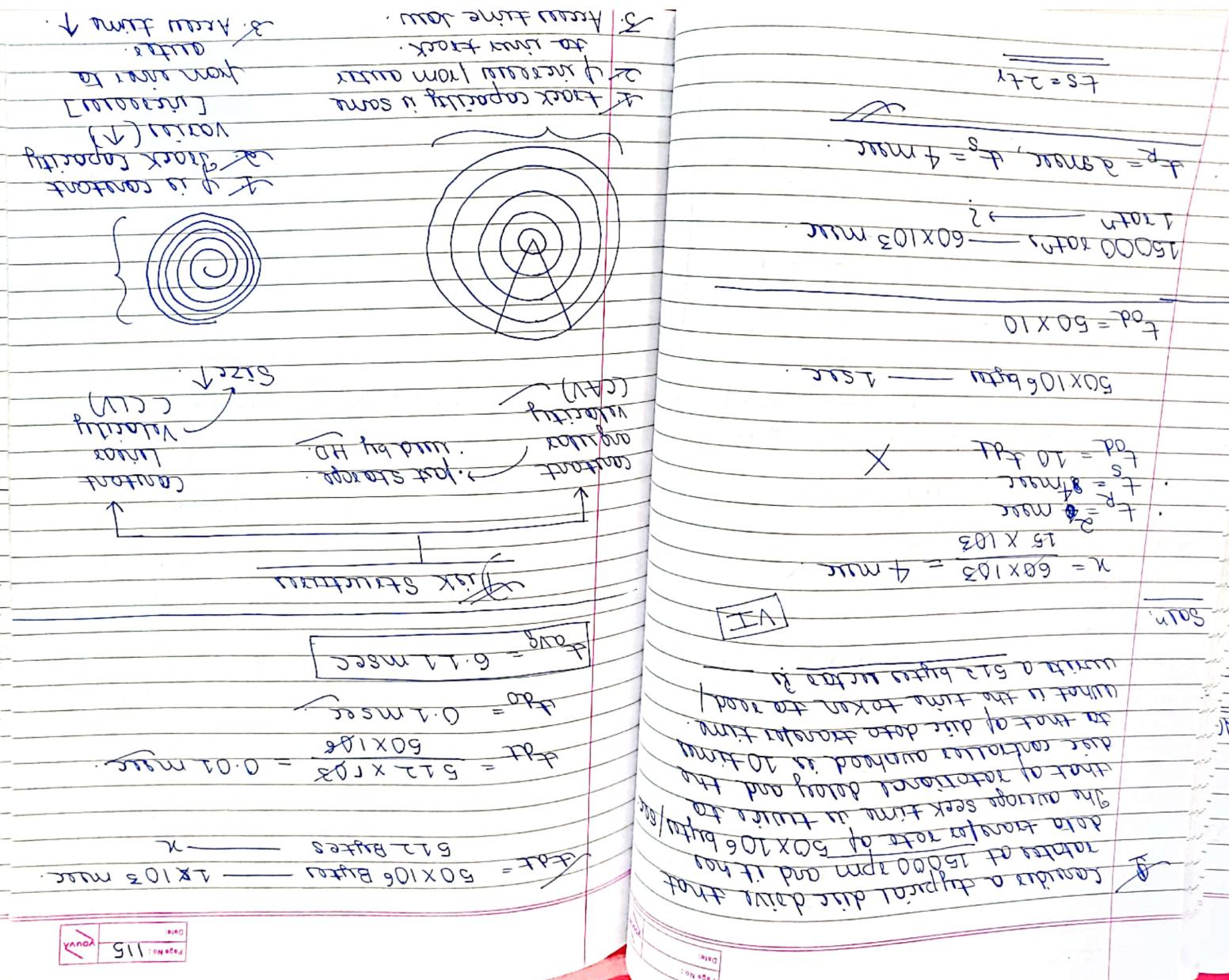
Q A disk system has the following specifications:

$$1500 \text{ msec} = 1.5 \text{ sec}$$

For 1000 transfers:

$$10 + 5 = 15 \text{ msec}$$

$$T_s = ts + LR$$



(1) $f = \text{No. of bits/track}$

\rightarrow Sector density on the tops (f)

$\overline{\text{f}}$

R/W Heads
many with a uniform density (unit/track)

Data tracks for platters when the tops are

$$L_{\text{Block}} = BL + GL$$

fixed R/W Head

GL

BL

GL

BL

S

N

T

L

B

A

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

Data on the tape organized as:

If a sequential media stored as:

APES

AGNETIC

8 bits \rightarrow 16 track

1 byte \rightarrow 2 track

1 sector \rightarrow 8 bits

1 track \rightarrow 16 sectors

1 byte \rightarrow 16 sectors

1 track \rightarrow 16 sectors

1 byte \rightarrow 16 sectors

1 track \rightarrow 16 sectors

1 byte \rightarrow 16 sectors

\rightarrow $\text{Sect}(CIV) > \text{Sect}(CAV)$

$$= 10 + 30 + \dots + 100$$

$$\text{no. of tracks} = 10$$

CIV

$$= 100 \text{ KB} \rightarrow$$

$$\text{No. of tracks} = 10$$

$$\text{Sector capacity} = 10 \text{ KB}$$

CAV

sequential

\rightarrow CAV

sequential data / CDs.

$$\phi = \text{No. of bytes/inch-track}$$

* Effective data transfer rate (D_{eff})

$$D_{eff} = U * D_{c}$$

$$C_{eff} = U * C$$

* Capacity of the tape
depends on the tape
constant storage density

Max. data transfer rate: (D_c)

$$D_c = V * N * \phi$$

where V is the tape speed
in inches per second
and N is the number of tracks
per inch.

* Capacity of the Tape:

$$C = L * N * \phi$$

\Rightarrow where L is the length of the tape

* Utilization factor of the Tape:

$$U = \frac{BL}{BL+GL}$$

Soln:

$$U = \frac{BL}{BL+GL}$$

1 inch \rightarrow 3000 bytes
 \therefore

$$N = \frac{6000}{3000} = 2 \text{ inches}$$

Calculate the utilization factor of the tape
where Gap length is 0.5 inch, storage
density is of 3000 bytes/inch and has
block storage capacity of 6 KB.

$$B_L = \frac{BC}{\phi}$$

$$\therefore U = \frac{2}{0.5 + 2} = 0.8$$

$\therefore U = 0.8$ \rightarrow 80% tape utilized

$$C = 88 \text{ MB}$$

$$\Rightarrow 88 \text{ MB} \\ \Leftarrow 800 \text{ cm} * 8 * 110 \text{ kb} \\ \text{cm} \\ = 8 * 8 * 110 \\ \text{sqm} \quad C = 1 * N * f$$

(i) what is the capacity of the tape.

with a velocity of 50 cm/s and it has a track density of 110 kb/track - cm writing 8 parallel tracks.

$$\Rightarrow 5.5 \text{ MBps} \\ \Leftarrow 5500 \text{ Kbps}$$

$$\Rightarrow 50 * 110 \text{ kb} * 8 \\ T_f = V \cdot N \cdot f$$

so
of 110 kb/inch - track density is 50 tracks/inch. So if it has a velocity of 50 cm/s, max data transfer rate is

121	volu
-----	------

$$C \Leftarrow 30.87 \text{ MB}$$

$$\text{Capacity} = 5243 \times 32 \text{ KBPS}$$

$$B_L = BC = \frac{32 \times 10^3}{6250} = 5.12 \text{ MB}$$

$$\frac{5.12 + 0.4}{2400 \times 12} = 53.187$$

$$BL + GL$$

$$- \text{No. of Blocks} = L$$

$$\text{Capacity } (C) = \text{No. of Blocks} * B$$

$$6350 \text{ bits} \quad 1 \text{ bit} \quad \alpha (B_f)$$

EN

so
How many bits can stored on a tape of 3400 feet?
density is 6350 bits/inch. The maximum storage from each other. The data is stored in a group of 0.4 inch supports the blocks and sectors. Each result of 32 KBPS and supports data on the tape is organized in blocks of 32 KBPS.

(iii) Let the length of a record is 2.4 cm and a gap of 0.6 cm separates each other. What is effective data transfer rate?

Soln:

$$D_{\text{eff}} = N * D$$

$$D = N * \frac{L}{T}$$

$$= 50 * 2 * 1.10 \text{ Kbps}$$

$$\Rightarrow 5500 \text{ Kbps.}$$

$$N = \frac{2.4}{2.4 + 0.6} = 0.8$$

$$D_{\text{eff}} = 0.8 * 5500 \\ = 4.4 \text{ MBps}$$

$$\boxed{D_{\text{eff}} = 4.4 \text{ MBps}}$$

(iv) what is block storage capacity?

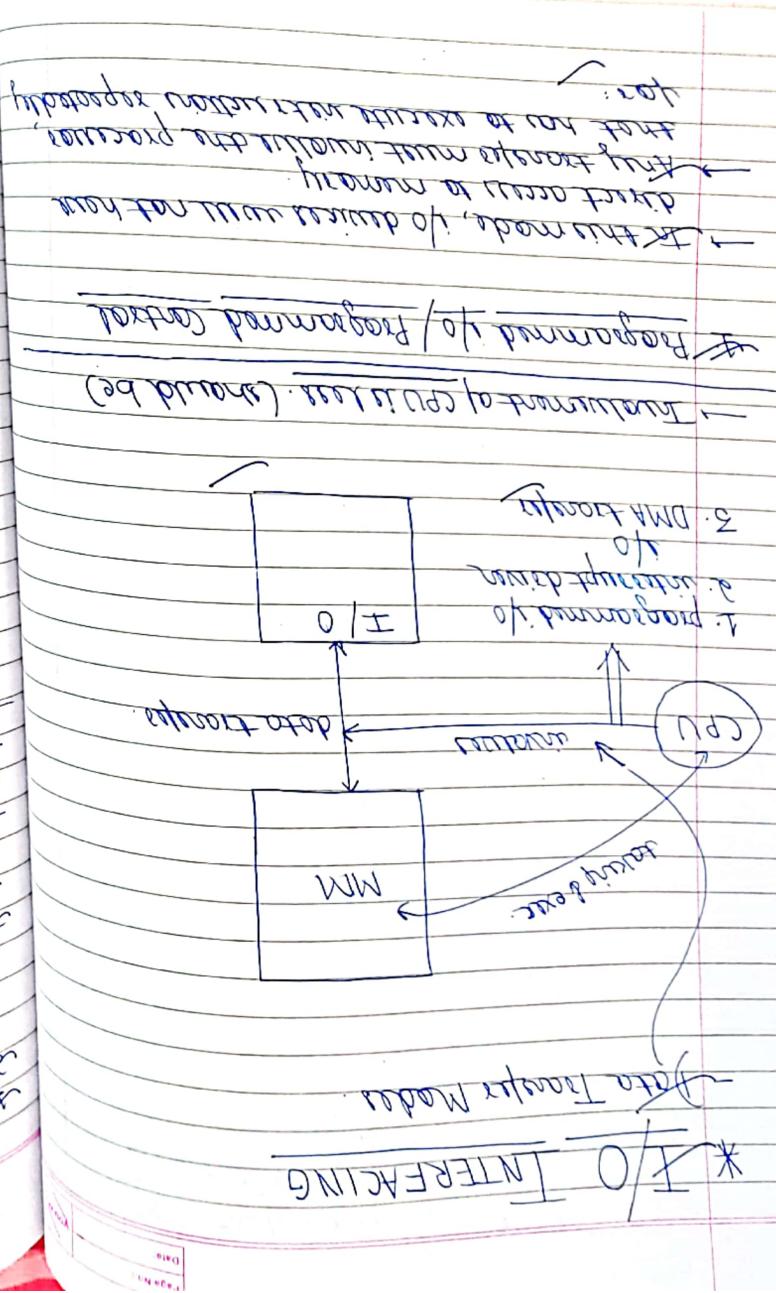
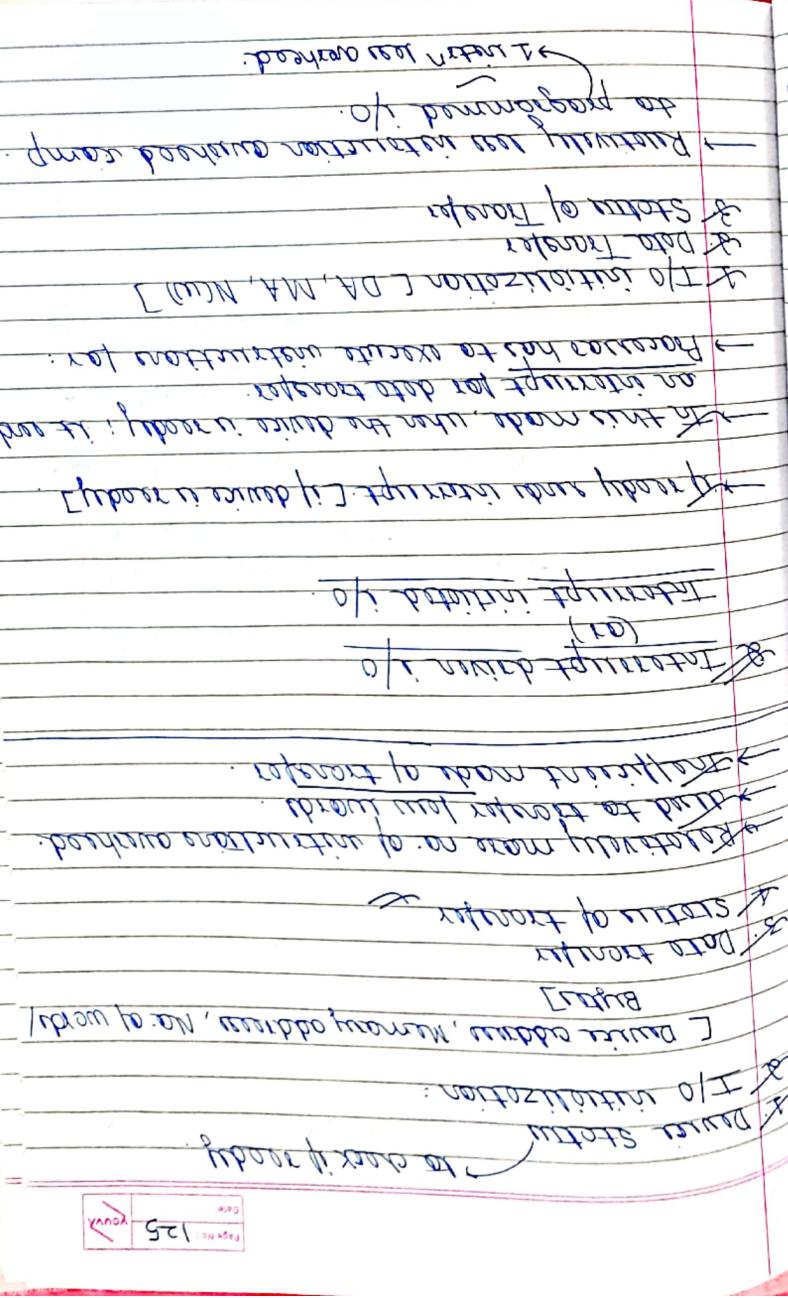
$$B_C = B_L * \frac{L}{T} =$$

$$B_C = B_L * N * \frac{L}{T}$$

$$\Rightarrow 2.4 * 2 * 1.10 \text{ KB} \\ \boxed{B_C = 264 \text{ KB}}$$

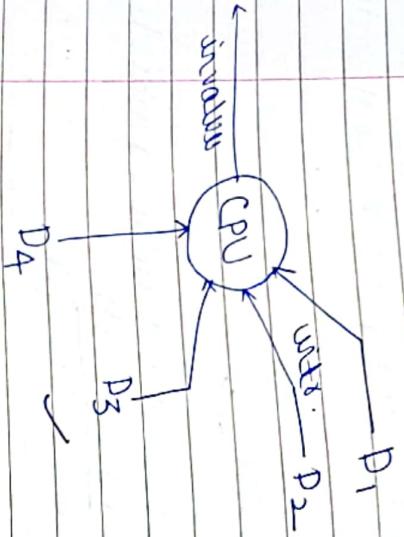
$$B_C = B_L * N * \frac{L}{T}$$

④



I/O INTERFACING

- Insufficient for large volumes of data transfer.



→ If many devices simultaneously request processor implements priority division I/O.
(Not suitable for large volumes of data transfer) ↴

* DMA TRANSFER [Direct Memory Access]

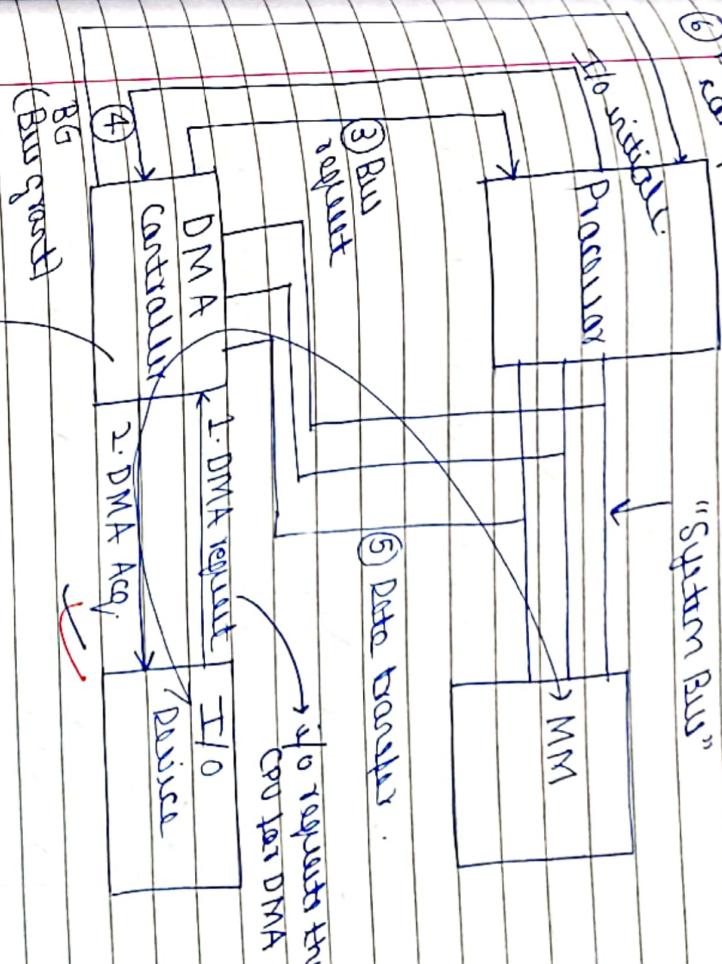
→ For large volumes of data transfer b/w disks and memory, processor is relieved comparatively using DMA controllers.

* only the I/O initialization is done by CPU.

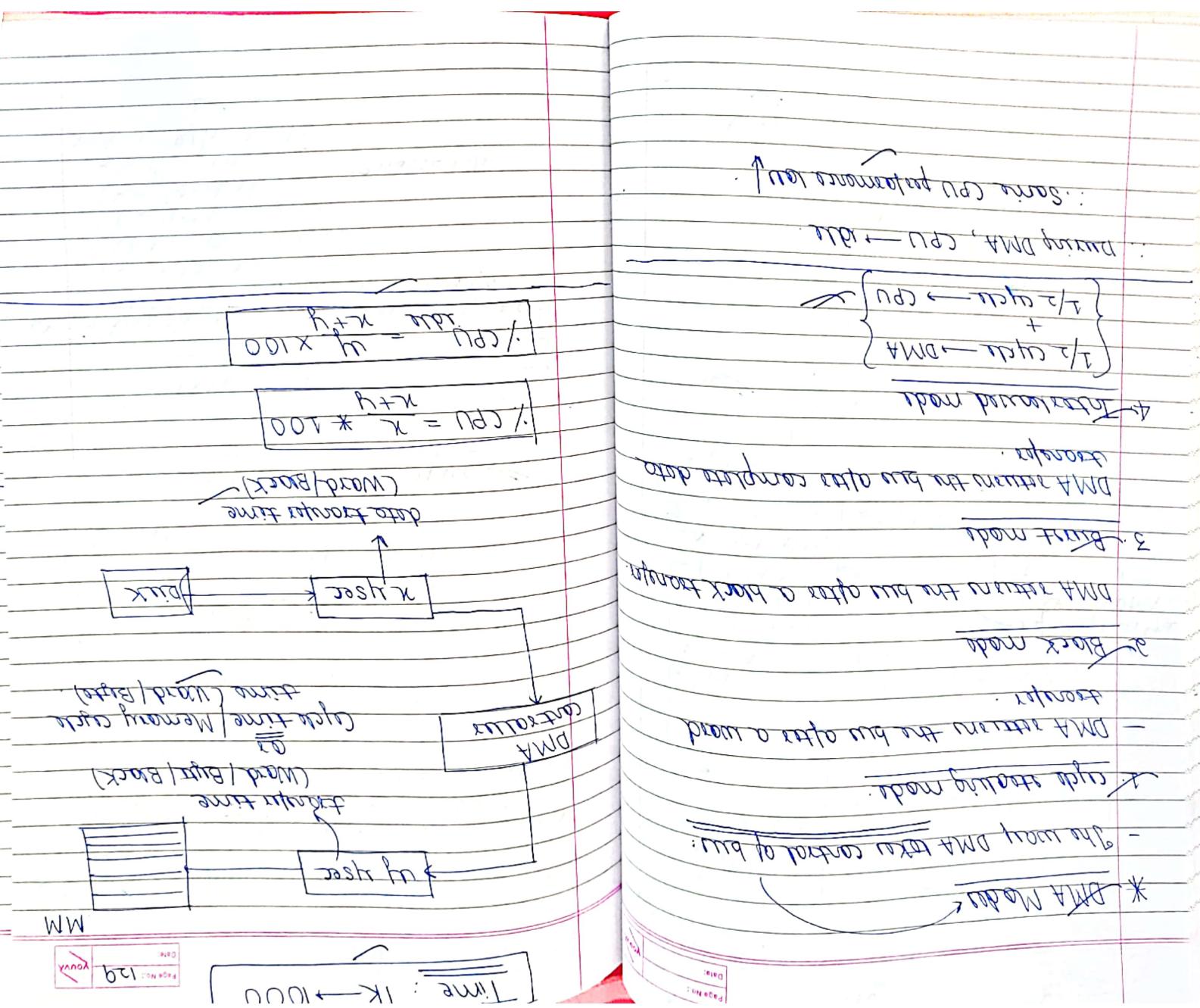
Page No.:
Date:

Page No.: 127
Date: 10/10/2023

Page No.: 127
Date: 10/10/2023



- No instruction by CPU, rest to DMA.



$$\begin{aligned}
 & \text{X} = 156.25 \text{ ns/c} \\
 & = \frac{612 \times 10^3}{4 \times 20 \times 10^3} \text{ ns/c} \\
 & = 20 \times 4 \text{ ns/c} \\
 & \text{X} \leftarrow 1 \text{ word} \quad (4 \text{ bytes}) \\
 & \text{X} \leftarrow 1 \text{ track} \quad (512 * 1K) \\
 & \text{X} \leftarrow 1 \text{ rotm} \rightarrow 1 \text{ track} \quad \leftarrow 20 \text{ msec} \\
 & \text{X} = 60 \times 10^3 \text{ msec} \\
 & \text{X} \leftarrow 3000 \text{ rotations} \quad \leftarrow 60 \times 10^3 \text{ msec} \\
 & \text{X} = ? \\
 & Y = 40 \text{ ns}
 \end{aligned}$$

What memory cycle time is 40ns. What per second?

What is the program to calculate it back side?

Cache memory model

between memory and I/O: using DMA when a 4 byte word is available, transferred

132	Y000
130	Y001

A disk system has the following properties:

- 3000 rpm
- 1KB/sector
- 512 sectors/track
- 128 tracks/ platter
- 16 platters
- and a R/W Head
- motor time

$$\begin{aligned}
 & \% \text{ CPU idle} = \frac{45\%}{100} = 45\% \\
 & \text{X} = 4 \times 10^6 = 400 \text{ msec} \\
 & \text{X} \leftarrow 1 \text{ word (4B)} \quad \leftarrow \text{X} \leftarrow 1 \text{ sec (10}^6 \text{ msec)} \\
 & \text{X} = 50 \text{ msec.} \\
 & \% \text{ CPU idle} = \frac{y + h}{400} \times 100
 \end{aligned}$$

What memory is 50 msec. What is the program to calculate it back side?

4 byte word is available, transfered

cache memory model of DMA when a word is available, then a

counts a cache of 10 Kbytes is used in

it's

$$\% \text{ CPU idle} = \frac{y}{x+y} = \frac{40}{40+156} = \frac{40}{256} \approx 0.203$$

$$\approx 20\% \quad \checkmark$$

$$x = \frac{64 \times 14 \cdot 28}{44 \times 1000} \approx 0.020 \text{ msec}$$

$$\Rightarrow 20.74 \mu\text{sec} \quad \checkmark$$

The storage area of a disk having inner diameter of 10cm & outer of 20 cm. The maximum storage density is 1400 bits/cm² and the disk rotates at 4200 rpm. The MM of the system is having 64 bit word and 1 usec cycle time.

$$\approx 4.6 \cdot \approx 5\% \quad \checkmark$$

- (i) What is the percentage of memory cycle utilization for transferring 1 word?

- (ii) What is the processor performance?

Soln.

$$P = \pi d = \frac{\pi}{7} \times 10 = 31.42 \text{ cm.}$$

$$\Rightarrow 43988 \leftarrow \text{Track capacity}$$

$$\Rightarrow 44 \text{ K bits}$$

$$4000 \text{ rotations} \quad \frac{1 \text{ rot}}{x} \quad 60 \times 10^6 \text{ usec}$$

$$x = \frac{60 \times 10^6}{4200} \Rightarrow 14.28 \times 10^3 \text{ usec}$$

4000

70

$$\Rightarrow 14.28 \text{ msec}$$

$$\% \text{ CPU busy} = 95\%$$

$$\Rightarrow \frac{x}{x+y} \times 100 = \frac{20.77}{21.77} \Rightarrow 95.40\%$$

$$\approx 95\% \quad \checkmark$$

$$44 \text{ K bits} \quad \frac{1}{64 \text{ bit}} \quad 14.28 \text{ msec}$$

$$y = 14.5.$$

word

~~8 bits/tick/disk~~ ~~outward to forward a~~

(ii) what is peak data transfer rate?

Total transfer per sec

conducts a transfer of 4 MBps (MHz)

$\text{No. of ticks} = 15480 \times 2^{10}$

$\text{No. of ticks} = 15480 \times 2^{10}$

$\text{Kbit} \in 0 \dots k-1 \quad (k \text{ bits})$

[why when count = 0]

a tick size of 15480 Kbit?

all max DMA request transfer to forward

using 16 bit count register. Hold many

data after buffer between memory and I/O

conduct a shift in which DMA is used to

$\% \text{ CPU busy} = 99.9\%$

$\% \text{ CPU busy} = \frac{200}{2} \times 100$

frame

~~600x10⁶~~ 600

$f = \frac{1}{T} = \frac{600}{156} = 3.904 \text{ MHz}$

$f = 600 \text{ MHz}$

Total seek cycle = 1200 msec

$T = 20 \times 10^6 \times 10^6$

$f = 20 \text{ KB} \quad 10^6 \text{ msec}$

$f = 20 \text{ KB} \quad 10^6 \text{ msec}$

A hard disk with a transfer rate of

20MBps is currently transferring data

which many using DMA. Given

processors runs at 600 MHz and it

takes 300 & 900 clock cycles to

initiate an interrupt DMA transfer.

If size of buffer is 20 KB, what is the

percentage of processor programs

idle?

if seek time is 0.1 sec

$f \leftarrow 6, f \leftarrow 10, f \leftarrow 15$

$$D = \text{no. of words/sec}$$

$$4 \times 10^6 \text{ instructions} \rightarrow 1 \text{ sec}$$

8 instructions

$$n = \frac{8}{4 \times 10^6} = 2 \mu\text{sec} \rightarrow \text{To read head}$$

2 μsec → 1 word

$$10^6 \mu\text{sec} \rightarrow n$$

$$n = 10^6 \rightarrow 0.5 \times 10^6 \text{ words}$$

$$\Rightarrow D_p = 0.5 \text{ million words/sec}$$

Q What is pack data transfer rate using interrupt mode which requires 4 instructions ahead to transfer a word?

P & 1 To read head (Byte/Word)

$$4 \times 10^6 \text{ instr.} \rightarrow 1 \text{ sec}$$

$$4 \text{ instr.} \rightarrow n$$

$$n = \frac{4}{4 \mu\text{sec}} = \frac{1}{\mu\text{sec}}$$

$$1 \mu\text{sec} \rightarrow 1 \text{ word}$$

$$10^6 \mu\text{sec} \rightarrow n$$

$$D_p = 1 \text{ million words/sec}$$

$$D_i = 1 \text{ million words/sec}$$

~~P & D~~ → Data Transfer Rate

$$P_I = \frac{D_I}{D_p}$$

$$P_I = \frac{1}{0.5} * P_p$$

$$P_I = 2 * P_p$$

~~P & 1~~ → Perform. of interrupt driven

Q Consider a CPU which has instruction execution completed in 7 clock cycles with 700 MHz. What is the data transfer rate using programmed control which requires 8 instructions per word to transfer a byte (word).
Soln.

1 word \rightarrow 8 instruction time

1 instruction \rightarrow 7 clock cycles

$$\begin{aligned} t_c &= 700 \text{ nsec} \\ t_c &= 7 \text{ nsec} \\ t_c &= 700 \end{aligned}$$

\downarrow
1 clock cycle time

\Rightarrow 1 instruction \rightarrow 7 clock cycles

$$(7 \times \frac{7}{700}) \text{ nsec}$$

1 byte \rightarrow 0.014 nsec.

8 bytes \rightarrow n .

Total $n = 0.084 \text{ sec}$ \rightarrow 1 word.

\Rightarrow

1 byte $0.084 \text{ sec} \rightarrow 1 \text{ byte}$

106 bytes (1 s) \rightarrow x

$$x = 106$$

$$0.084$$

\rightarrow 12.5 million bytes / sec.

Q Consider a device of 10 Kbps is connected to a CPU, the data is transferred in byte wise. Let the interrupt overhead is 4 nsec. What is the performance gain of operating the device in interrupt mode over programmed control?

Soln.

$$P_I = \frac{T_P}{T_I} = \frac{T_P}{4}$$

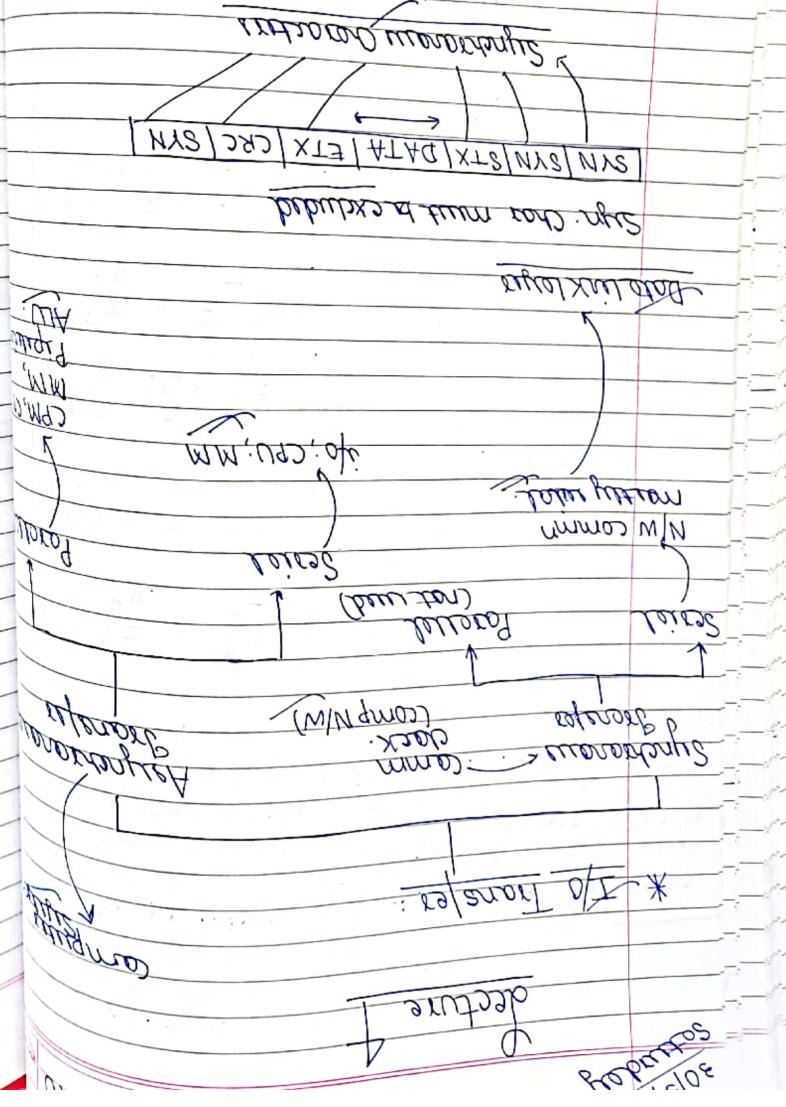
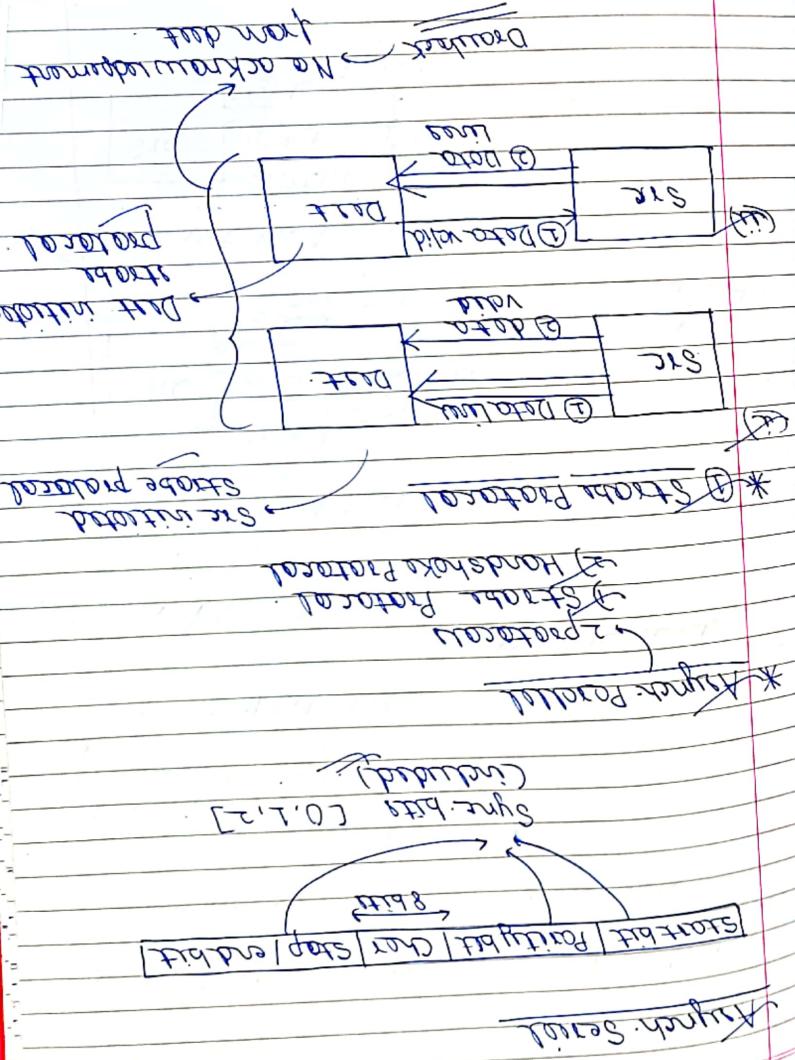
① $10 \text{ KB} \rightarrow 1 \text{ sec} (10^6 \text{ nsec})$
 $1 \text{ B} \rightarrow n$

$$n = \frac{10^6}{10 \times 10^3} = \frac{100}{10} \text{ nsec}$$

$$\frac{P_I}{P_P} = \frac{100}{4} = 25$$

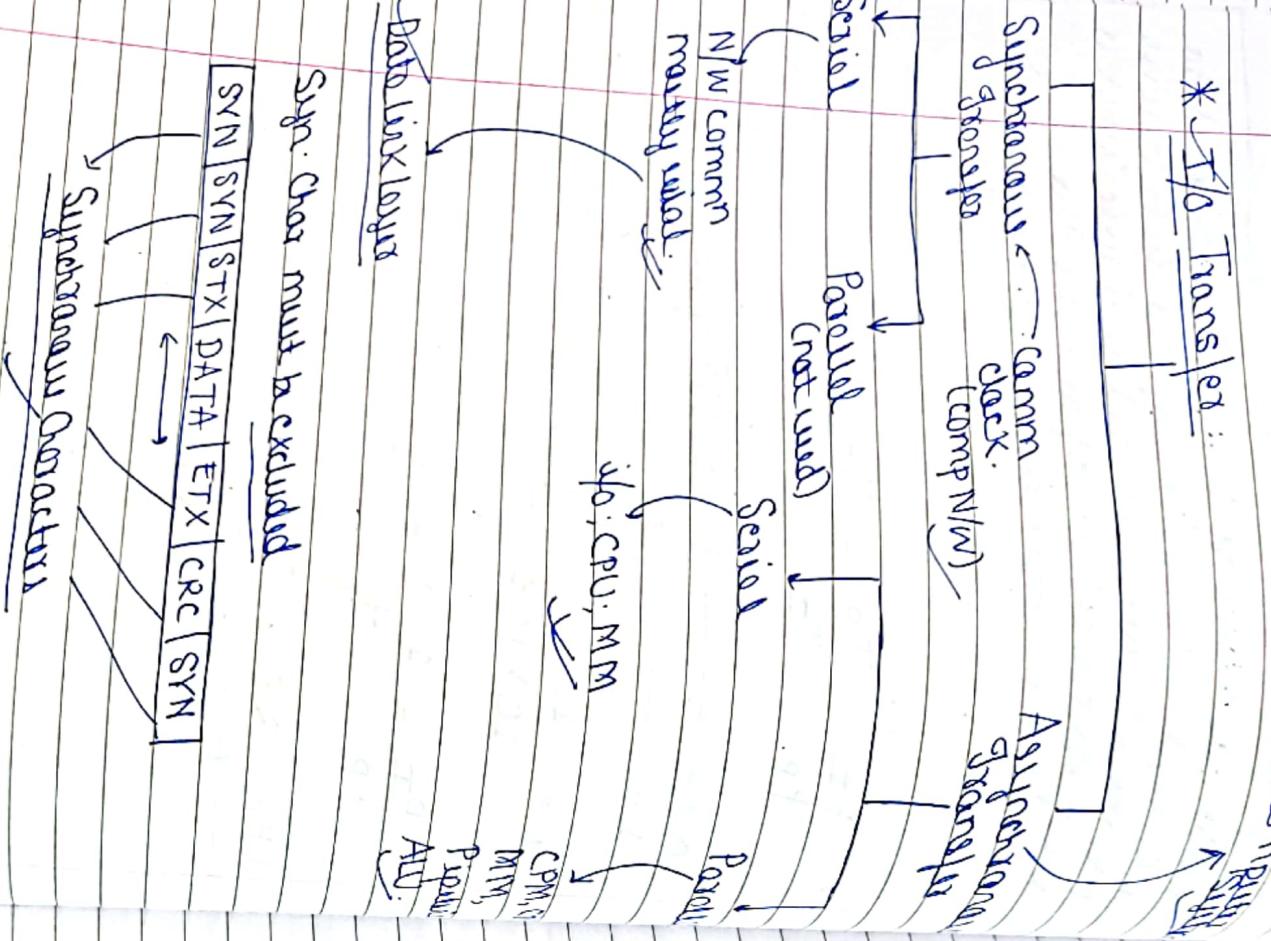
$$P_I = 25 * P_P$$

\downarrow
as times better performance.

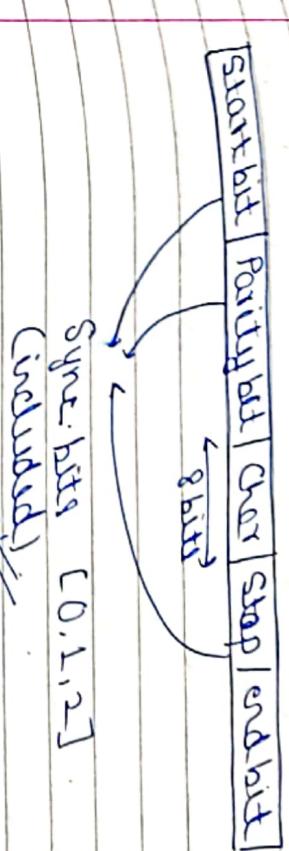


lecture 4

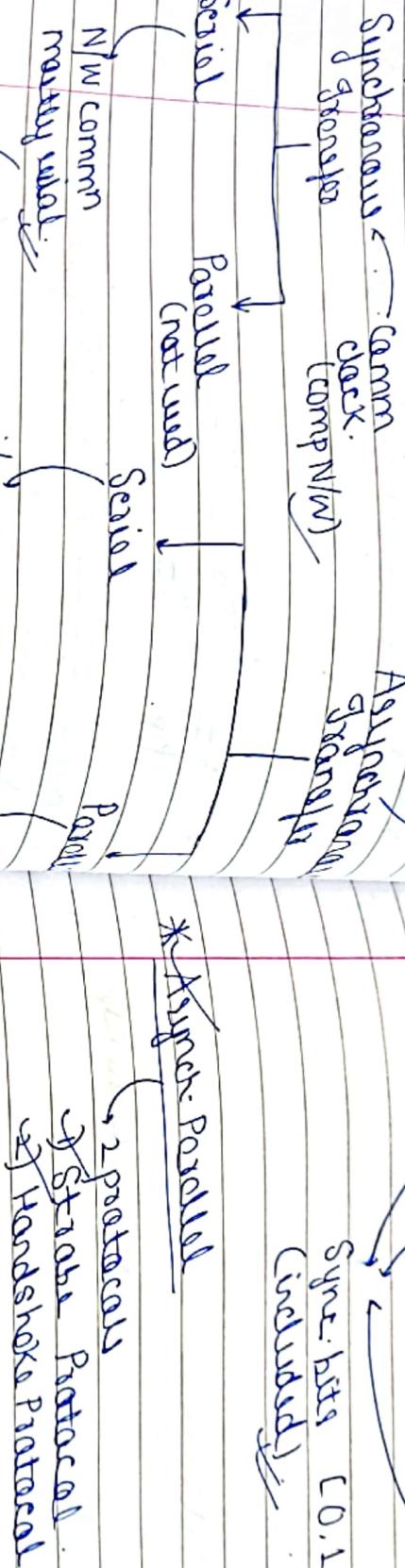
* I/O Transfer:



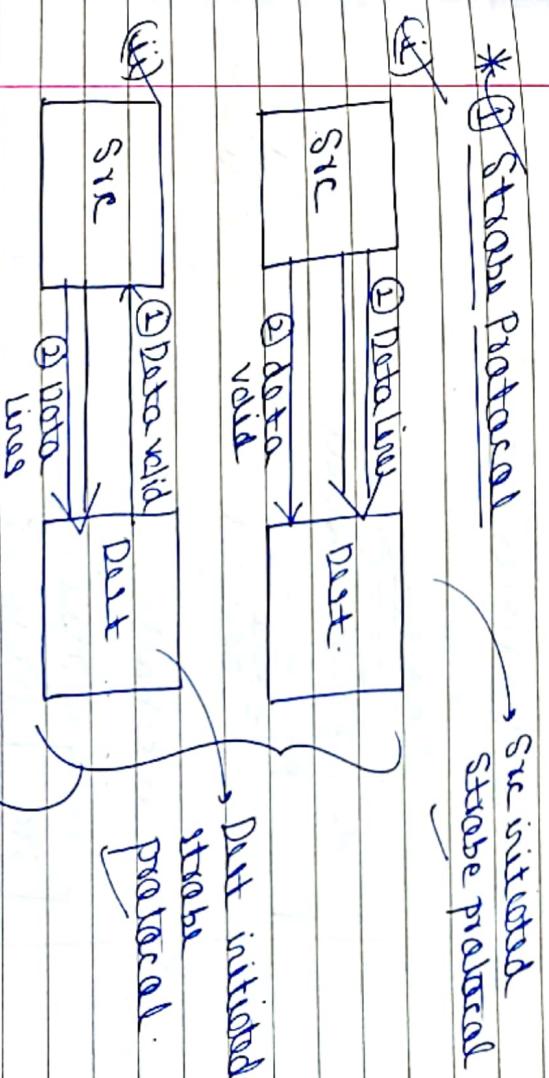
Kynch Serial



* ① Strobe Parallel



* ② Strobe Protocol



Drawback → No acknowledgement from data.

Transfer Rate

$$T = \frac{240 \text{ char/sec}}{(8 + 2)}$$

$\rightarrow 2400 \text{ bits}$

$\rightarrow 10 \text{ bits}$

$\rightarrow 3300 \text{ bits/sec}$

Q: Assume each character code consists of 8 bits.

a. What rate of 2400 & a stop bit.

Q: No. of characters transmitted per sec which consist of 8 bits.

~~Serial transmission is implemented by serial port.~~

Q: Serial transmission is implemented by serial port.

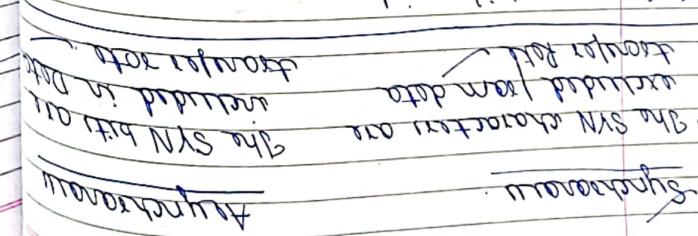
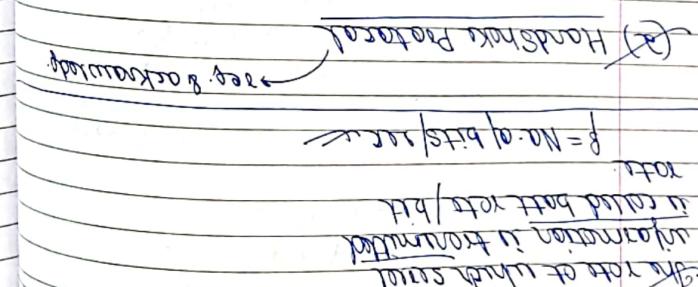
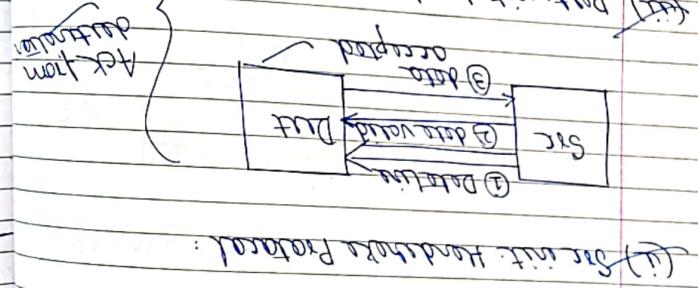
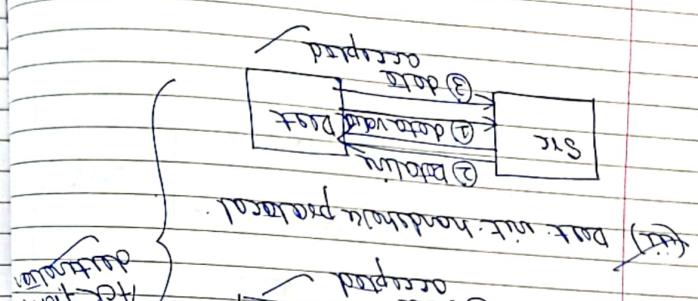
Ans: A parallel port has 8 data bits, a parity bit & 2 stop bits for each character. The maximum baud rate is 11 bits/sec. The formula for calculating baud rate is:

$$\text{Baud rate} \leftarrow (8 + 1 + 2) * 300 \text{ bits/sec}$$

$\rightarrow 300 \text{ bits/sec}$

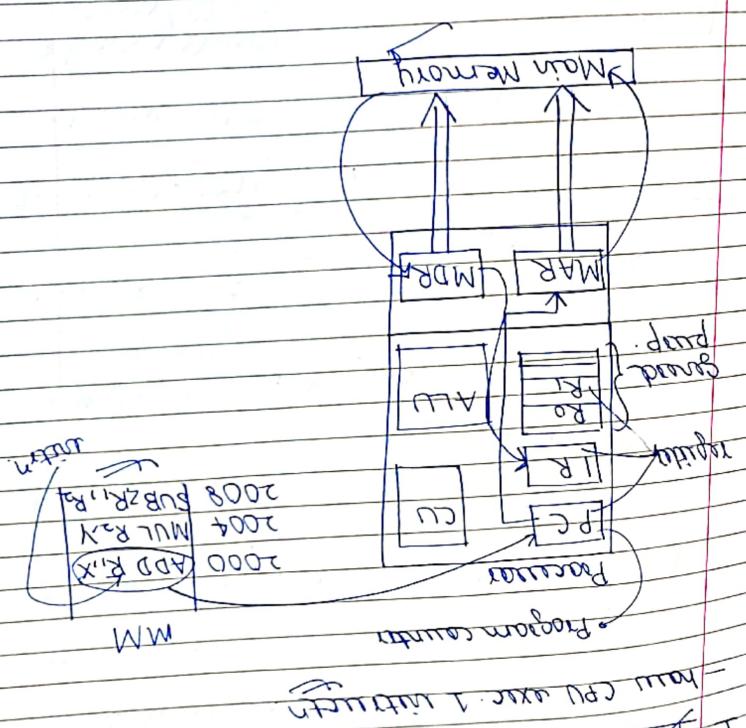
$\rightarrow 300 \text{ bits/sec} \times 11 \text{ bits} \rightarrow 3300 \text{ bits/sec}$

$\rightarrow 300 \text{ bits/sec} \times 8 \text{ bits} \rightarrow 2400 \text{ bits/sec}$



MACHINE INSTRUCTIONS

1. BASIC OPERATIONAL CONCEPT



$$\begin{aligned}
 & \text{Total no. of instructions per second} = 1200 \times 24 = 28800 \text{ instructions/s} \\
 & \text{Instruction size} = 8 \text{ bytes} \\
 & \text{Data size} = 135 \text{ bytes} \\
 & \text{Time taken by one instruction} = \frac{135}{8} = 17.125 \text{ microseconds} \\
 & \text{Time taken by all instructions} = 17.125 \times 28800 = 487440 \text{ microseconds} \\
 & \text{Time taken by all instructions} = 487440 \text{ microseconds} = 0.48744 \text{ seconds} \\
 & \text{Time taken by all instructions} = 0.48744 \text{ seconds} = 487.44 \text{ milliseconds} \\
 & \text{Time taken by all instructions} = 487.44 \text{ milliseconds} = 0.48744 \text{ seconds}
 \end{aligned}$$

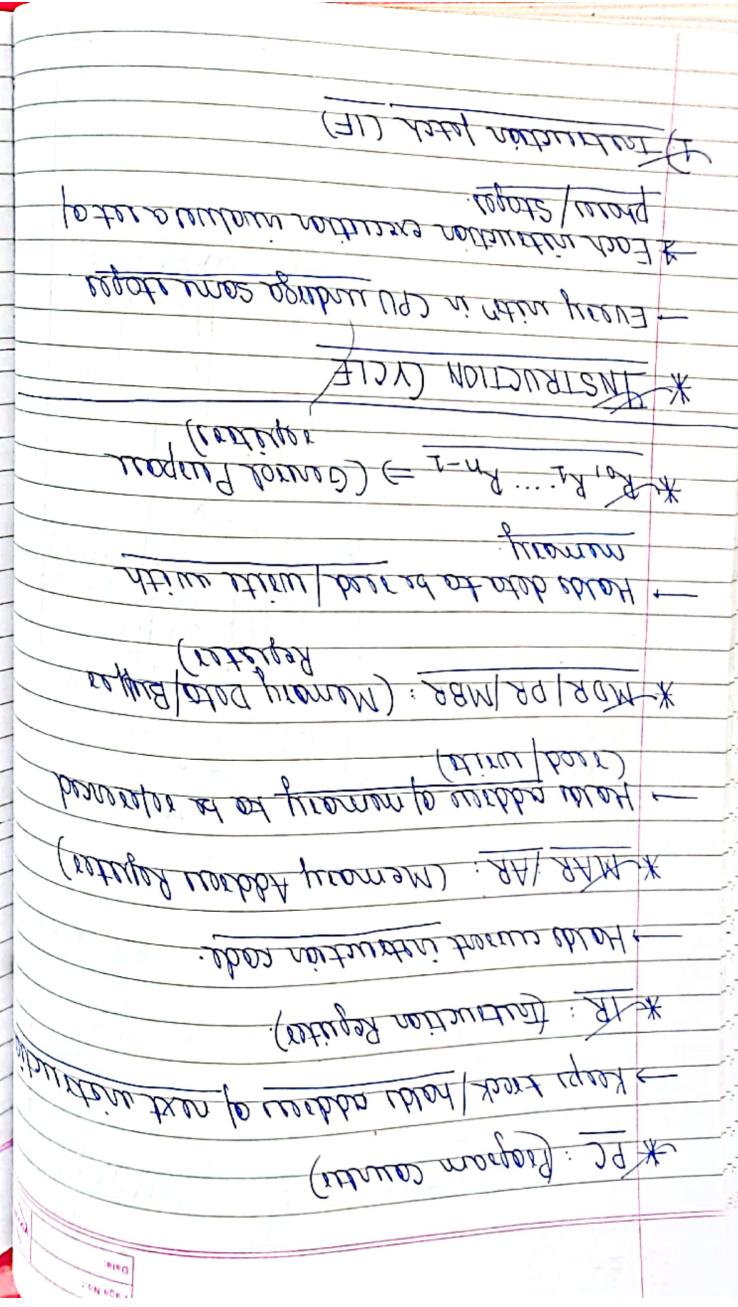
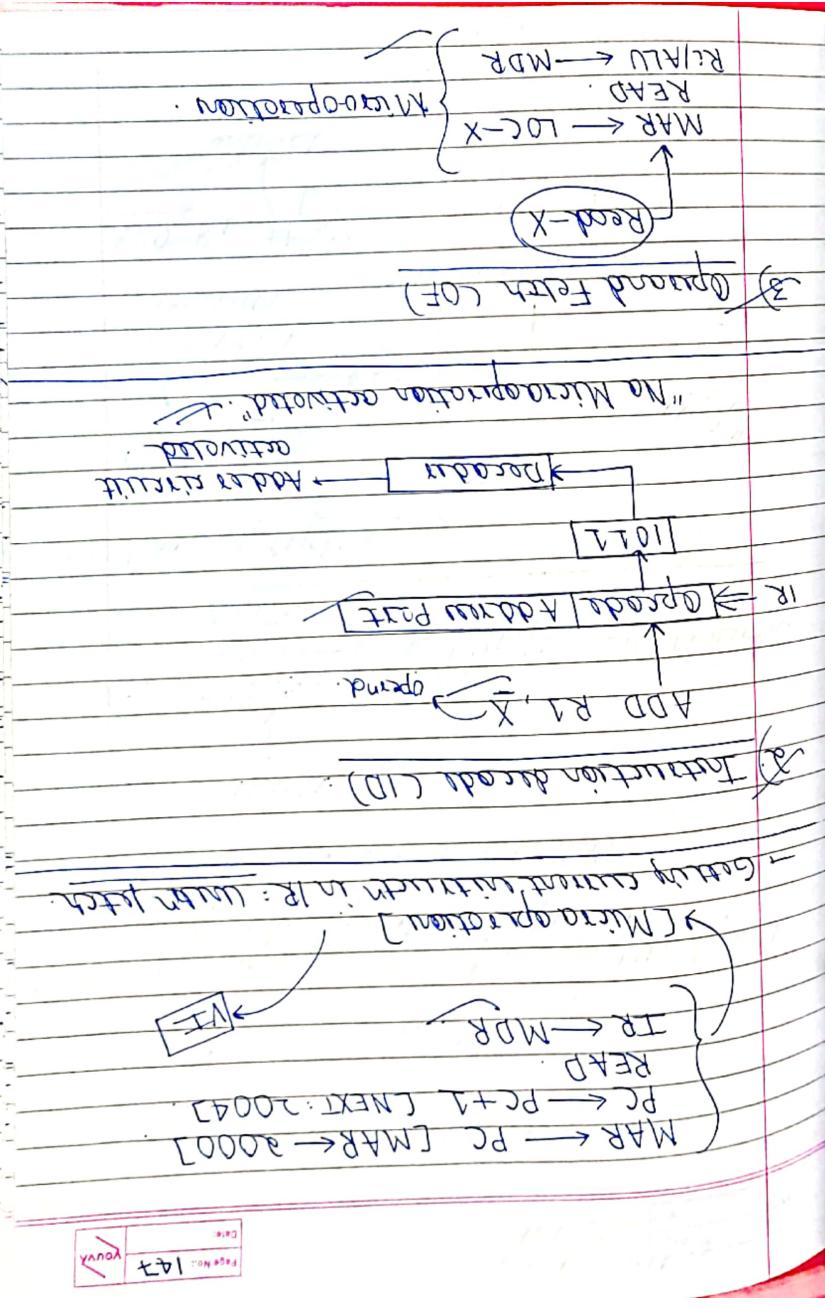
Q. A serial formular T is with 8 bits and a parity bit for each character. A microcontroller T has 32 bits, 1 step bit and a stop bit. The total no. of bits in both cases will be

A. 32 bits + 1 start bit + 1 step bit + 1 parity bit + 1 stop bit = 36 bits

B. 32 bits + 1 start bit + 1 step bit + 1 parity bit + 1 stop bit = 36 bits

C. 32 bits + 1 start bit + 1 step bit + 1 parity bit + 1 stop bit = 36 bits

D. 32 bits + 1 start bit + 1 step bit + 1 parity bit + 1 stop bit = 36 bits



4) Execute and Store (EX)

→ Perform required ALL operation.

NOTE: The microoperations for execution depend on operand (operation performed).

Store → write back (WB)

will (Z)



MAR \leftarrow LOC(2)
MDR \leftarrow R₁ R₂ | ALU(R) } Microop
WRITE }

5) Indirect cycle (IN)

MM

ADD R₁, 300

300 | 20
address of
operand

direction

absolute

(absolute address).

ADD R₁, #300

300 | 20
address of
operand

absolute

850 | 20
EA

* Points to memory
2 mem cycles required.

(II) indirect address (points to mem op)

→ Indirect cycle ✓

6) Interrupt cycle phase (IP)

- If interrupt occurs, CPU has to send the
interrupt.

1. Init. Fetch (IF)

2. Inst. Decode (ID) } T₁

3. Operand Fetch (OF) } T₂

4. Execute & Store (EX) } T₃

5. Indirect cycle (IN) } T₄

6. Interrupt cycle (IP) } T₅

Total width
cycle time.

* Each instruction execution involves a set of
micro-operation to be implemented by
the processor.

→ (Proc. per op or
in register only)

Micro-operations

Arithmetical microoperations

[VI]

(ADD R₁, 300) X [Not working]

Instruction

Micro-operation

Pg No.: 151
 Date:

8 micro	15.1
operations.	hours

→ Operations that are performed on register
only

2. Logic micro-operation

$$R_2 \leftarrow R_2 \wedge R_3$$

3. Disk Micro opns

$$R_2 \leftarrow R_2 \ll 2$$

4. Register transfer Micro operation:

$$R_3 \leftarrow R_4.$$

[Two or 1 clock cycle]

Next micro-
operation.

NOTE: The micro operations of instruction cycle
are mostly register transfers. So in
computer in 1 clock cycle].

e.g: 1 instruction takes 7 clock cycles

mean,
7 micro-operations

↓
VIE

consider a CPU in which execution of each
instruction consists in 8 clock cycles,
operated with 800MHz clock. What is
the time taken to execute a program
consisting of 16 instructions.

Soln:
 $\Rightarrow \text{Total CC} = 16 \times 8 = 128 \text{ clock cycles}$

$$\frac{1}{800} \times 128 =$$

Given:

$$f_C = 800 \text{ MHz}$$

$$T_C = \frac{1}{800} \text{ sec} \Rightarrow \frac{1}{800} \times 10^3 \text{ nsec}$$

$$\Rightarrow 1.25 \text{ nsec}$$

1 clock cycle
time

1 instruction → $8 \times 1.25 \text{ ns}$

$$16 \text{ inst} \rightarrow \xrightarrow{\quad} \\ \Rightarrow N = 16 \times 8 \times 1.25 = 160 \text{ nsec}$$

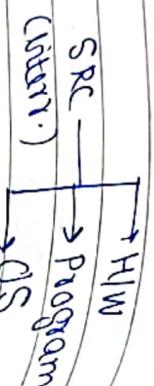
$$\boxed{T = 160 \text{ ns}}$$

* INTERRUPT PROCESSING:

- current prog is suspended

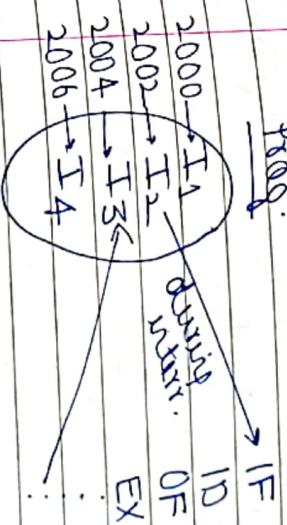
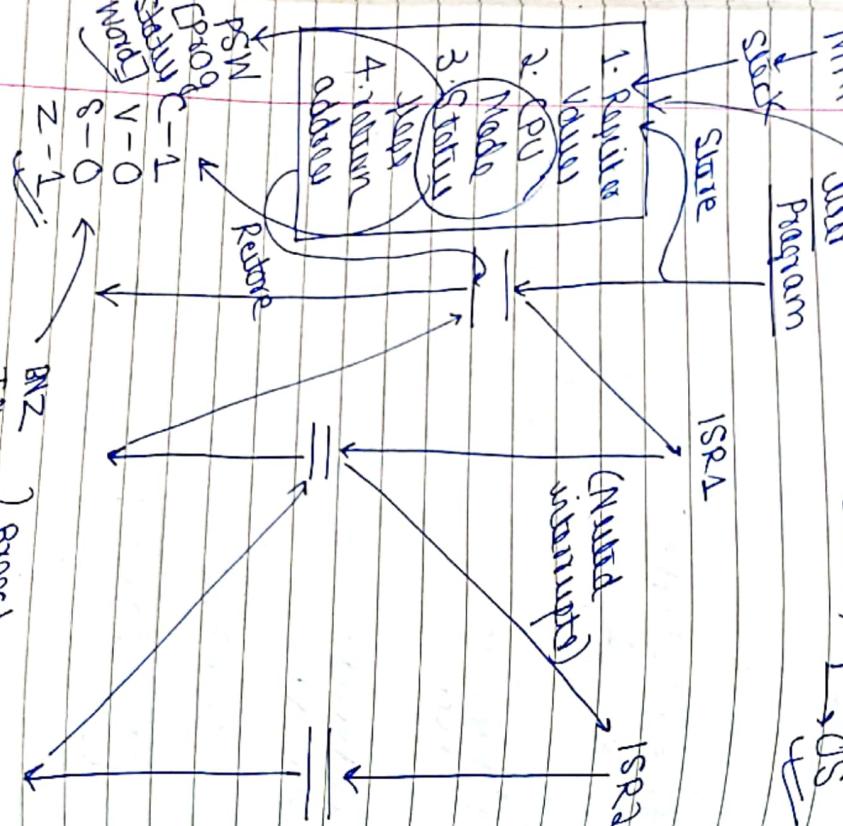
- status has updated from last ALU op.
(Branch with depend on status
logic)

Overhead:



From when CPU has to resume executing the program after ISR:
the return address.

\rightarrow IP is on switched to processor but it is its responsibility.



Even though CPU gets interrupt while performing I_2 , it completes I_2 , it stores the return address (I_3)

BNZ

JNN

} Branch

PSW
Status
Word

C-1

V-O

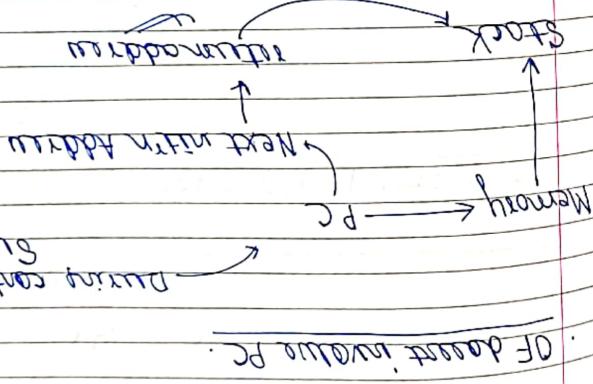
Z-O

Z-1

PC

X

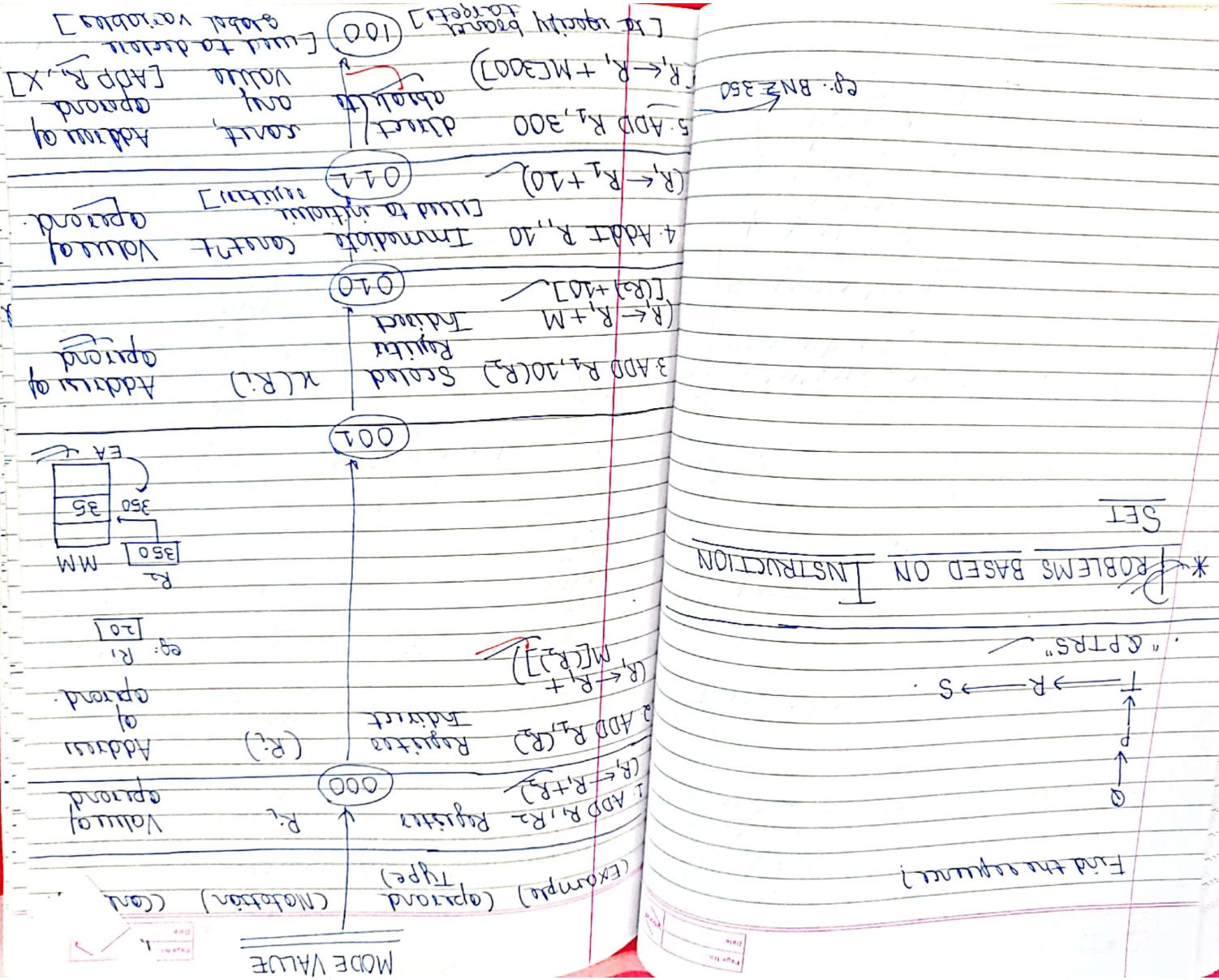
- q, p, f, r, s
 return address
 the instruction
 t: loads the new PC value based on
 S: pops the stack from the stack
 R: returns the ISR
 u: interrupt resolution
 q: yields to the execution of current
 P: pushes the process state onto the
 stack
 accu which is the same of the result
 the following are the same of the result



- (d) Initiation of Interrupt Service
 (c) Conditional Branch
 (b) Periodic Fetch
 (a) Interrupt Handler Function

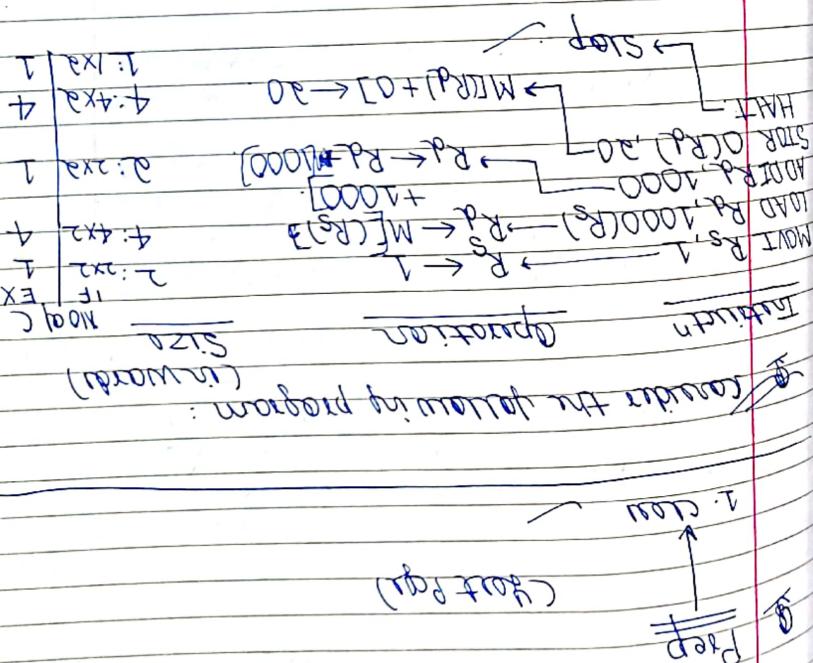
157 Page No.

- A processor needs software interrupt
 to handle — interrupt is handled
 non-privileged to change from an
 user mode — interrupt is handled
 by software to initiate system
 call or to invoke a supervisor program
 (a) As soon as it arrives
 (b) Provides time intervals
 (c) After completion of a program
 (d) After completion of interrupt cycle
- What is the possible application of
 micro-operation
 at the following situations of
 a) the parallel operation of
 memory → MBR
 PC → Y
 MAR → X
 MBR → PC
- A processor handles an interrupt
 (d) To implement re-callable
 (e) To activate the System Software
 (f) To test the interrupt System
- A processor needs software interrupt
 (a) To implement re-callable
 (b) To invoke the Supervisor Program
 (c) To invoke the System Call
 (d) After completion of a program

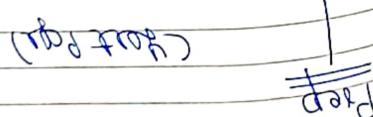


program is executed.

Initially the correct format of the
values 18, 2, and 16.
1000, 1001, 1010. are having the
format of memory location



Consider the following program:



161

What is an instruction during ADDI?

(V) An about program is stored in the adder module from location 1000

$$13 * 64 = 104 \text{ Bytes}$$

What is the size of a word by 64 bit.

$$\therefore 46.25 \text{ ns}$$

$$TP = \text{no. of clock cycles} * T_C$$

$$800 \text{ ns} = ?$$

What is the time taken to complete the about program.

Given CPU operates at 800 MHz

3+CC

I ₁	8	4	1	E	X
I ₂	8	4	1		
I ₃	8	4	1		
I ₄	8	4	1		
I ₅	2	4	1		

How many clock cycles required to complete the about program?

After execution of memory located in data register.

(iii) For all instructions:

$$\begin{aligned} & RS=1 \\ & RD = MC[1000] + 1 = MC[1001] \\ & RD = 1 + 1000 = 1001 \rightarrow 20 \\ & MC[1001 + 0] \rightarrow 20 \end{aligned}$$

$$\begin{aligned} & (a) MC[1021] \rightarrow 20 \\ & (b) MC[1001] \rightarrow 20 \\ & (c) MC[1020] \rightarrow 20 \\ & (d) MC[1000] \rightarrow 20 \end{aligned}$$

Return Add.



- (Q) What return address is pushed onto stack?
A) There is no interrupt during HALT?

Conceptually, no interrupt can occur during HALT.

(During HALT PC is not updated).
PC ← PC + 1

∴ Return Address = 1096 ✓

- RA will be 1096 till OS updates it with next program.

A word add:

During ADD : 1008 (RA)
During HALT : 1012 (RA)

(Q) consider the following program.

I₁ → 1000 - 1015
I₂ → 1016 - 1047
I₃ → 1048 - 1063
I₄ → 1064 - 1095
I₅ → 1096 - 1103

Return Add = 1064. ✓

$$(\text{Word Size} = 32 \text{ bits})$$

How many dark bytes required?

Instructions Fetch & Decode: Cache / Word
Add with both operands: 1CC
Results from memory: 3CC
in Registers

(ii) The dark bytes needed for various operations is:

Hault	
Stop	1
M0V 6000, R2	M[6000] → R2
M0D R2, R3	R2 → R2 + R3
M0R R2, (R1)	R2 → M[R1]
M0W R1, 5000	R1 → M[5000]

Size
(Words)

OpCode

Memory

(iii) Bits addition Memory

$$\begin{aligned} I_5 &= 1000 - 1063 & (33 \times 2 = 64) \\ I_4 &= 1064 - 1095 \\ I_3 &= 1096 - 1127 \\ I_2 &= 1128 - 1194 \\ I_1 &= 1192 - 1223 \end{aligned}$$

$$\therefore \text{Divide Hault : RA} = 1192 \cdot \text{Ans}$$

X

16	16
Even	Odd

Total
memory

$$\therefore CC = 7 + 11 + 5 + 11 + 2$$

$$X \quad [CC = 36]$$

F8 D RM ADD

-	-	2	I ₅
-	8	4	I ₄
-	4	2	I ₃
-	8	2	I ₂
-	4	3	I ₁

$$+ 5 + 3 + 7 + 2$$

\sum

\approx

Lecture 5

Consider the following program Segment:

$I_1 : 1000 - 1007 \checkmark$
 $I_2 : 1008 - 1011$
 $I_3 : 1012 - 15$
 $I_4 : 1016 - 23$
 $I_5 : 1024 - 27$

$I024 = RA$

Instruction

Operation

(word)

MOV R1, 3000	$R1 \leftarrow M[3000] \curvearrowright ①$	2
LOOP: R2, (R3)	$R2 \leftarrow M[R3] \curvearrowright ②$	1
MOV R2, R1	$R2 \leftarrow R1 + R2$	1
ADD (R3), R2	$M[R3] \leftarrow R2 \curvearrowright ③$	1
MOV R3, R3	$R3 \leftarrow R3 + 1$	1
INC R1	$R1 \leftarrow R1 - 1$	1
DEC R1		
BNZ LOOP	Branch if nonzero Loop;	2
HALT.	Stop	1

- (x) Assume the contents of memory location 3000 is 10. $M[3000] \leftarrow 10$ is word addressable memory. The no. of memory references for accessing data in executing the program completely:

SAR
 $R1 \leftarrow 10$

$$1 + (2 \times 10) = \underline{21 \text{ memory address}}$$

- ① happens one time; ② & ③ occur till loop ends

Q) Let the size of a word is 32 bits. What return address is saved onto stack if there is an interrupt during INC R3.
(SA = 1000)

Soln:

I1: 1000 -> 1001

T2:-

I3: 2 -3

I4: 3 -4.

I5: 1004

I6: 1005

I7: 1006

I8: 1007 -08

I9: 1009 ✓

Return Address

Word Addressing

[Assume]

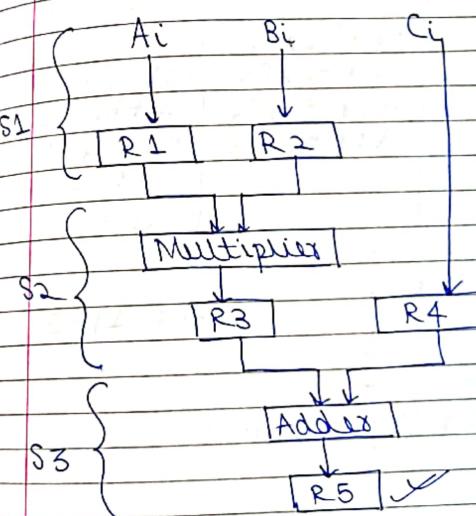
Page No.: 173
Date: 10/10/2023

PIPELINING

* 1. ARITHMETIC PIPELINE

- Eq:

$$D_i \leftarrow A_i * B_i + C_i \quad \text{where } i = 1 \text{ to } 4$$



∴ 1 stage : 1 CC : 3 stages : 3 CC
∴ Total = 12 CC (Without pipeline)

Assume each stage completes in 1 clock cycle.

Lat: $n \rightarrow$ no. of tasks/instructions ($n=4$)
 $K \rightarrow$ no. of stages/segments ($K=3$)

using sequential/non-pipelined execution:

$$\text{No. of CC} = n * K \quad : \quad 4 * 3 = 12 \text{CC}$$

* Using Pipelined execution:

→ Parallel execution.

→ A task is the total operation performed by passing through all the stages of the pipeline.

clock	S1	S2	S3
1	A1, B1	-	-
2	A2, B2	A1 * B1 + C1	-
3	A3, B3	A2 * B2 + C2	A1 * B1 + C1
4	A4, B4	A3 * B3 + C3	A2 * B2 + C2
5	-	A4 * B4 + C4	A3 * B3 + C3
6	-	-	A4 * B4 + C4

→ Total stop delay (td):

Passing time in a stage;
 It varies from 1 stage to another.

→ Total Stop delay (td):

Total Stop transfer of data, it is round between the stages.

VE

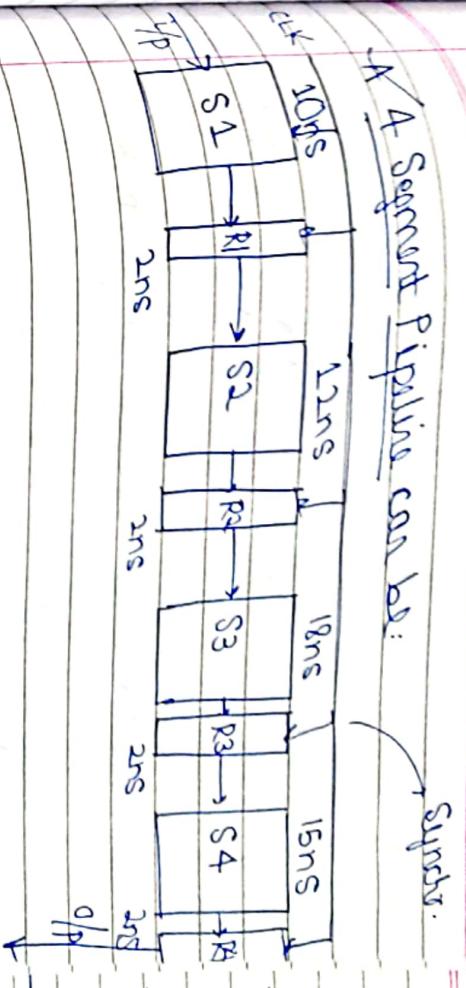
→ The time period for clock cycle in Pipelined execution

$$tp = \max(t_i) + td$$

\rightarrow 1 clock time:
 max(t_i) >>> td
 time

VI

∴ 1st stop takes K clock cycles.



Time
overhead computation

$$(T) = \frac{1}{n} * k * (n-1) * T_p = \frac{n-1}{n} * k * T_p$$

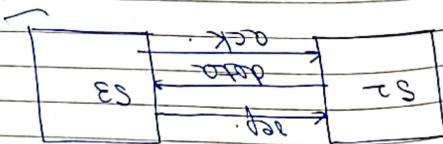
Time \leq T_p : $T_p \leq$ time to compute a task / solution in non-pipeline.

$S = \text{Time without pipeline}$

Speed up (S)

- After ACC, $S_1 \rightarrow S_2$
- Add to the next stages simultaneously
- On a common clock all the stages simultaneously

(ii) Synchronous Pipeline (by delay):



Counted using Head-to-tail Solution
Data flow along the pipeline. Stop in Asynchronous Pipeline.

Throughput = $N_A \cdot \text{Time/Task}$

System: time required to process a task in a non-pipeline parallel processing.

Parallel processing in a pipeline system:

- Building a task into stages;
- All tasks must operate simultaneously;
- Pipeline;

$T_p = \text{no. of clock cycles} * T_p$

$T_p = (k + (n-1)) T_p$

Time taken in pipeline with (n, k) :

$T_p = \frac{1}{ACC \cdot Time}$

$f_p = \frac{T_p}{Time}$

Frequency of the Pipeline:

$$t_n \approx K \cdot t_p \quad \checkmark$$

time for 1 task.

Substitute in (1).

$$\Rightarrow S = \frac{n * K \cdot t_p}{(K + (n - 1)) * t_p}$$

$$S = \frac{n K}{K + (n - 1)} \quad \checkmark \quad (2)$$

* For less no. of tasks instructions:

$$K + (n - 1) \longrightarrow n$$

Substitute in (1)

$$\Rightarrow S = \frac{n t_n}{n + t_p} \quad \checkmark \quad (3)$$

$$E_K = \frac{n K}{K + (n - 1)}$$

$$E_K \Rightarrow \frac{n}{K + (n - 1)} \quad \checkmark$$

$$i), S = K : E_K = 100\% \quad \therefore S = S_{max}$$

* Throughput of the Pipeline:

no. of tasks pipelined time

Substitute in (2):

$$H_K = \frac{n}{(K + (n - 1)) t_p}$$

$$\Rightarrow S = \frac{n K}{n} = K$$

$$\boxed{S = K} \quad \longrightarrow \quad (4)$$

$\checkmark \rightarrow$: KX faster than non-pipelined.

NOTE: Max Speedup using a Pipeline is equal to no. of stages.

$$\boxed{S_{max} = S_{ideal} = K}$$

* Efficiency of the Pipeline:

$$E_K = \frac{S}{K}$$

The number of clock cycles needed to perform 100 tasks in a 6 segment pipeline

$$\text{No. of CC} = K + (n-1) \\ \Rightarrow 6 + (100-1) \\ \Rightarrow 105$$

$$\boxed{\text{No. of CC} = 105}$$

For a given 6 Segment Pipeline, what is the speed up for 100 tasks.

$$S = \frac{nK}{K+(n-1)} \\ \Rightarrow \frac{100 \times 6}{105}$$

$$\boxed{S = 5.71}$$

For a given 6 Segment Pipeline, what is the efficiency for 100 tasks.

$$E_K = \frac{n}{K+(n-1)} \\ \Rightarrow \frac{6}{105} = 0.95$$

∴ 95% efficient

During floating point arithmetic, the time delay in 4 stages of a pipeline is 60ns, 70ns, 90ns & 80ns. If the inter stage delay of 10ns.

(e) What is the time period for clock cycle?

$$t_p = \max\{t_i\} + t_d \\ \Rightarrow 90 + 10 = 100$$

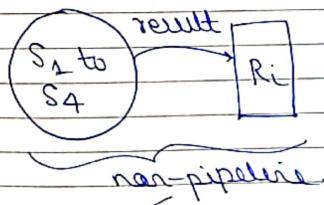
$$\therefore \boxed{t_p = 100 \text{ ns}}$$

(f) What is the speed up?

$$S = \frac{t_n}{t_p} \quad (\text{Don't Know})$$

$$\Rightarrow \frac{60+70+90+80+10}{90+10} \xrightarrow{\text{overhead}}$$

$$\Rightarrow \frac{310}{100} = \underline{\underline{3.1}}$$



(iii) What is the max Speed up?

$$\begin{aligned} S_{\max} &= K \\ \therefore K = 4 & \quad [S_{\max} = 4] \end{aligned}$$

Consider a synchronous pipeline with 4 stages, where each stage takes a uniform delay of 20ns and has buffer overhead of 5ns.

(iv) What is the clock frequency of the pipeline?

Soln:

$$f_p = \frac{1}{t_p}$$

$$t_p = t_h + t_d$$

$$\Rightarrow 20 + 5 = 25 \text{ ns.}$$

$$f_p = \frac{1}{25 \times 10^{-9}} = \frac{10^9}{25} \text{ Hz}$$

$$\Rightarrow 40 \text{ MHz.}$$

$$\boxed{f_p = 40 \text{ MHz}}$$

(v) What is the speed up?

$$\begin{aligned} \frac{S_c}{t_p} &= \frac{85}{25} \\ \therefore t_h &= (20+20+20+20)+5 = 3.4 \end{aligned}$$

Consider a synchronous pipeline with 5 stages. The delay of S_3 is twice to that of S_2 and S_4 is half that of S_5 . S_2 & S_4 are having same delay as that of S_1 . The delay of $S_1 = 10 \text{ ns}$. What is the speed up?

$$S_1 = S_3 = S_4 = 10 \text{ ns.}$$

$$S_2 = 5 \text{ ns}$$

$$S_5 = 20 \text{ ns.}$$

$$S_c = \frac{t_h}{t_p} = \frac{55}{20} = 2.75 \cdot 10^9 \text{ ns}$$

non-pipeline

Consider a CPU in which each instruction execution requires 4 clock cycles, operated with 2.5 GHz. The same processor is implemented into pipelined processor with 5 stages. What is the Speed up?

Scanned with CamScanner

Comparing 2 diff pipelines:

$$t_R = \frac{1}{f_c} = \frac{1}{2.5 \times 10^9} = 0.4 \text{ ns}$$

$$S = t_R / t_p$$

$$\Rightarrow S = f_c = 2.5 \text{ GHz}, t_c = \frac{1}{2.5 \times 10^9} = 0.4 \text{ ns}$$

$$\Rightarrow \frac{1}{2.5} = 0.4 \text{ ns}$$

$$\therefore 1 \text{ width} \rightarrow 4 \text{ Cycles} \quad (4 \text{ ns}) \rightarrow (t_p)$$

$$\Rightarrow t_p = \frac{1}{f_c} = \frac{1}{2.5 \times 10^9} = 0.4 \text{ ns}$$

$$t_i = 4 \text{ ns}, 5 \text{ ns}, 4 \text{ ns}, 3 \text{ ns}, 5 \text{ ns.}$$

and

with buffer overhead of 1 ns.
→ Design A is having 5 stages with a
total pipeline delay of 2 ns.

(i) How much time is saved using design B.
over A for 100 instruction?

Soln.

$$T_A = ((K + (n-1)) t_p$$

$$\Rightarrow (5 + (100-1)) \cdot 0.4 = 624 \text{ ns}$$

$$T_B = (9 + (100-1)) * 2 = 216 \text{ ns.}$$

~~Consider a pipelined processor with 4 stages operating at 800 MHz. What is the time taken to perform 20 tasks?~~

Soln.

~~408 ns saved.~~

~~What is the speed up?~~

~~✓~~

$$t_p = \frac{1}{f_c}$$

$$\Rightarrow \frac{1}{800 \times 10^9} = 1.25 \text{ ns}$$

$$\Rightarrow 1.25 \text{ ns}$$

$$T_p = (K + (n-1)) * t_p$$

$$\Rightarrow (4 + 19) * 1.25 = 28.75 \text{ ns}$$

$$S = 2.89$$

same pipeline

Date:

Page No.: 185
Year:

$$\begin{aligned} K &= 5, n = 100 \\ t_i &= 4 \text{ ns}, 5 \text{ ns}, 13 \text{ ns}, 6 \text{ ns}, 5 \text{ ns} \\ t_d &= 1 \text{ ns} \end{aligned}$$

$$7 \text{ ns} \quad 6 \text{ ns} \quad 4 \text{ ns}$$

what is the speed up?

VI

$$T_2 = \frac{(5+99)*14}{(6+99)*8}$$

$$\Rightarrow 1456 \rightarrow 1.73$$

$$840$$

For a given task:

$T_1 \rightarrow$ Time taken in pipeline.
 $T_2 \rightarrow$ Time taken in non-pipeline.

$$T_1 = T_2$$

VE

$$b) T_1 < T_2$$

$$c) T_1 > T_2$$

$$(d) None$$

Soln

$$T_2 = \frac{(K + (n-1)) t_p}{t_d}$$

INSTRUCTION PIPELINE

- overlapping various stages of instructions
cycle for several instructions is instruction
pipeline.

- the objective is $[CPI_{avg} = 1]$

(clock per instruction)

- A 4 stage instruction pipeline can be:

clock cycle	1	2	3	4	5	6	7
I ₁	IF	ID	OF	EX			
I ₂		IF	ID	OF	EX		
I ₃			IF	ID	OF	EX	
I ₄				IF	ID	OF	EX

Each stage takes 1cc.

completely overlapped pipeline

4 stage - 4 mix. [Each stage → 1cc].

If for T_2 $(4+3) = \frac{T}{4}$

$$\begin{aligned} n &= 1 \\ T_1 &= (K + (n-1)) t_p \quad T_2 = t_n \\ \Rightarrow K t_p &= t_n \\ t_n &= 10 + 12 + 18 + 15 + 2 \\ \Rightarrow 57 \text{ ns} & \end{aligned}$$

(i) Consider an instruction pipeline with 4 stages. The clock cycles needed by the stages for each instruction is:

Solved: What is the Speedup?

$$S = \frac{30}{14} (n_p) = \underline{\underline{2.14}}$$

	F	D	E	S
I ₁	2	1	2	2
I ₂	1	3	3	2
I ₃	2	2	2	2
I ₄	1	3	1	1

∴ Computationally overlapping pipeline for
the no. of clock cycles needed in sequential non-pipeline execution.

Solved:
 $T = 9 + 8 + 6 = 30 \text{ clock cycles}$
 $\{I_1, I_2, I_3, I_4\}$

Solved: How many clock cycles needed in pipeline execution?
 $\Rightarrow 14$.

$$n_p = P + K + (n-1)X \quad [\text{Each stage} \rightarrow 1CC]$$

$$\Rightarrow \frac{1}{800 \times 10^6} = \underline{\underline{1.25 \text{ ns}}} (t_p)$$

(Time for 1CC).

$$T_p = 1.25 * 14 = \underline{\underline{17.5 \text{ ns}}}$$

	I ₁	I ₂	I ₃	I ₄
L	3 4 5 6 7 8 9 10 11 12 13			
F	F D E E S S			
-	F D D D E E S S			
I ₃	- F F - D D - E E S S			
I ₄	- - F - D D D E - S			

→ stall cycle.

* STALL CYCLE

→ During it no meaningful operation performed
you can instruction you can instruction will stall due to:
→ A stall cycle will occur due to:

- (i) different clock cycles needed by the stages for each instruction.
- (ii) maximum buffer overhead.
[next stage waits for]
- (iii) Instruction pipeline dependency

Instruction pipeline dependency

$$S_{eff} = \frac{S_{ideal}}{1 + stall_frac * stall_cycle}$$

↳ effective speed up

$$S_{ideal} = K$$

VH

$$S_{eff} = \frac{K}{(1 + stall_frac * stall_cycle)}$$

↳ the clock cycles needed by the stages for each instruction

Consider an instruction pipeline with 5 stages, it has a stall cycle depending on many memory dependency.
Let there are 20% memory references.
What is the speed up?

	S ₁	S ₂	S ₃	S ₄
I ₁	1	2	1	2
I ₂	2	1	2	1
I ₃	1	2	1	
I ₄	2	1	2	1

2 2 3 4 5 6 7 8 9 10 11 12 13 14

$$S_{eff} = \frac{5}{1 + 0.4} = \frac{5}{1.4} = 3.57$$

$$\Rightarrow \frac{5}{1 + 0.4} = \frac{5}{1.4} = 3.57$$

[Pipeline with no stall cycle]

↳ completely unblocked pipeline.

↳ consider a 4 stage instruction pipeline to execute a loop.

$$y[i] = 1; i < 100; i++)$$

I₁:
I₂:
I₃:
I₄:

(ii) The output of I_1 for $i=2$ is available after 5 clock cycles

1	2	3	4	5	6	7	8	9	10	11	12	13
S_1	S_2	S_3	S_4	S_4								
I_1	S_1	S_1	S_2	S_3	S_3	S_4						
I_2							S_1	S_2	$-$	S_3	S_4	
I_3								S_1	S_2	$-$	S_3	S_4
I_4									S_1	S_2	$-$	S_3
i_2										S_1	S_2	$-$

5th : 13 clock cycles ✓

instruction:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
S_1	S_2	S_3	S_4	S_4												
I_1							S_1	S_2	$-$	S_3	S_4					
I_2								S_1	S_2	$-$	S_3	S_4				
I_3									S_1	S_2	$-$	S_3	S_4			
I_4										S_1	S_2	$-$	S_3	S_4		

S_4

continuation of prev.

18 clock cycle

& consider an instruction pipeline with 4 stages.

IF, OF, & WB → 1 clock cycle each
EX → 45 instruction → 3 clocks

35 bits → 2 clocks
20 bits → 1 clock

How many clock cycles required?

Total 100
bits.

Ques:

clock cycle

Page No.:	103
Date:	10/10/2023

$$\textcircled{1} \quad K = 4, \quad n = 100$$

$$\text{No. of clock cycles} = K + (n - 1)$$

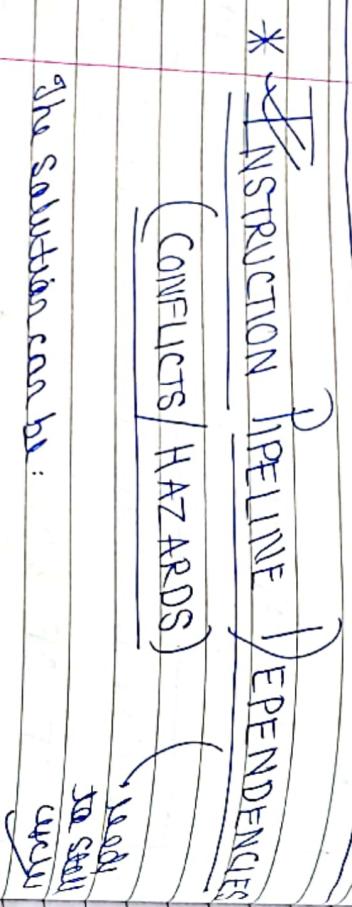
$$\Rightarrow 4 + (100 - 1) = 103$$

\checkmark

Excuse for 45 = $45 \times 2 = 90$ caused by 100m
 Excuse for 35 = $35 \times 1 = 35$.
 Excuse for 20 = $20 \times 0 = 0$.

: 229 clock cycles

The solution can be:



	1	2	3	4	5	6	7
I ₁	IF	ID	OF	EX			
I ₂	IF	ID	OF	EX			
I ₃	*	IF	ID	OF	EX		
I ₄							

→ Shell cycle

(data dependency)

- Pipelining is taken care.

~~Software (S/W)~~

- The compiler in which we are writing, takes care.

⇒ 3 types of dependences :

(A) Data dependency → I₃ depends on result of I₂

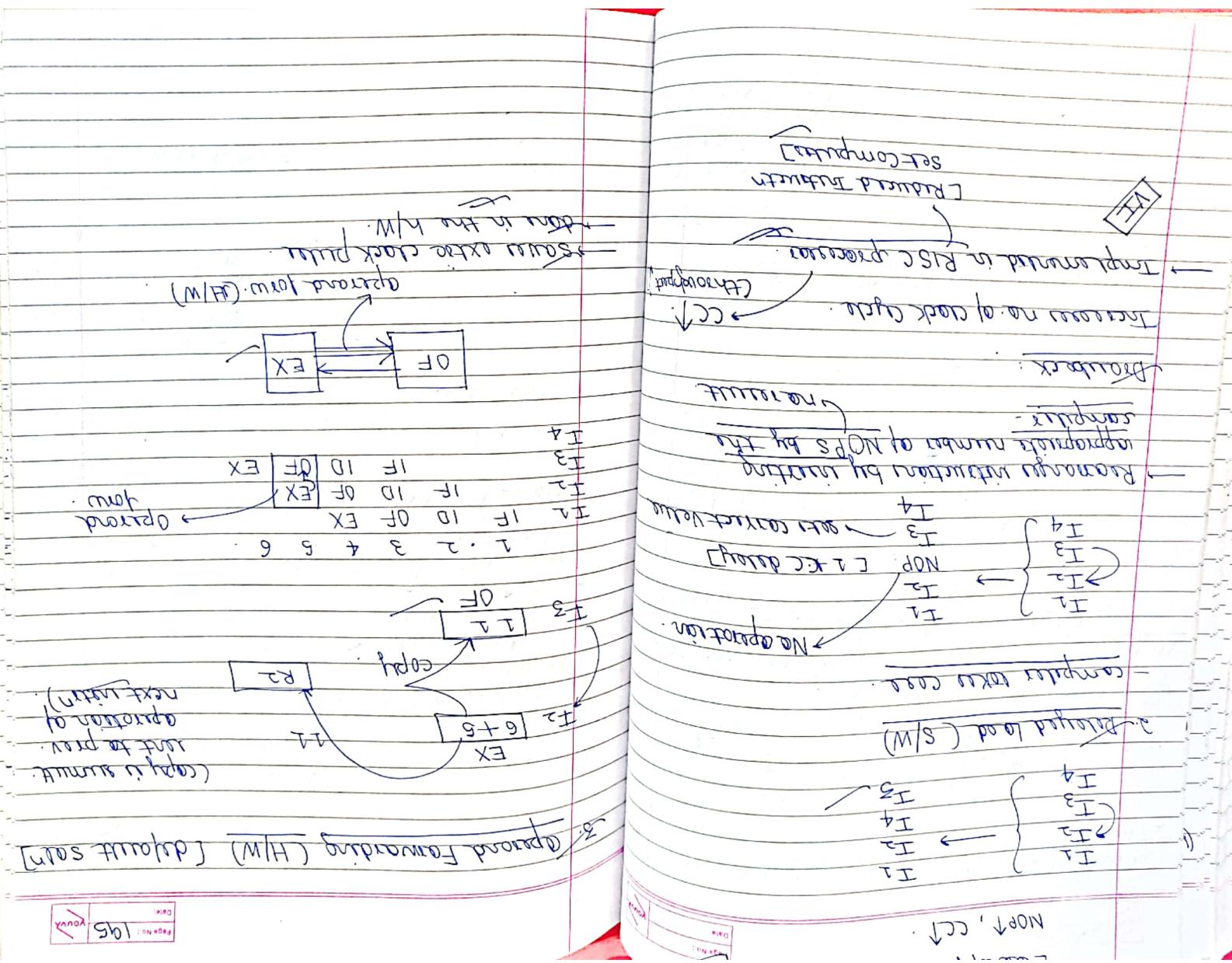
Eq:
 I₁: R₀ ← R₁ + R₂
 I₂: R₂ ← R₃ + R₄

→ If one instruction depends on the result of its previous instruction, it leads to data dependency. We

→ The solution can be:

1. Dynamic scheduling (H/W & S/W)

→ Rearrange the instructions such that the conflict is removed.
 → Instruction after dependency need parallel statements.



(*) consider an instruction pipeline with 4 stages:

[Fetch (F), Decode (D), Execute (E), and Write Back (W)]

The stages F, D & W $\xrightarrow{\text{takes}}$ 1 clock cycle for every instruction.

E \rightarrow 3 clocks $\leftarrow *$

and

1 clock $\rightarrow + 8$

If operand forwarding is used, how many clock cycles required to complete the following program?

Marks



\rightarrow 8 clock cycles

Stages: Instruction Fetch: IF \rightarrow 1 clock
& Decode
Op-code Fetch: OF \rightarrow 1 clock cycle
Register operation: PO \rightarrow 1 clock cycle
(ex)

$$3 \text{ CC: } *$$

$$5 \text{ CC: } =$$

ADD $R_2, R_1, R_0 \leftarrow$
MUL $R_4, R_3, R_5 \leftarrow$
SUB $R_6, R_5, R_4 \leftarrow$ 2 dependences.

I₁ ADD R_2, R_1, R_0
I₂ MUL R_4, R_3, R_5
I₃ SUB R_6, R_5, R_4

I₁ F D E
I₂ F D [E]
I₃ F D --

I₁ F D E E
I₂ F D [E] E E
I₃ F D -- E

I₁ F D E E E
I₂ F D [E] E E E
I₃ F D -- E

I₁ F D E E E E
I₂ F D [E] E E E E
I₃ F D -- E

If operand forwarding is used, how many clock cycles required to complete the program?

Given

R₂ after 4th CC.

needed

8 clock cycles

	1	2	3	4	5	6	7	8	9	10	11	12	13
I1	L	IF	OF	PO	PO	WB							
I2	-	IF	OF	-	-	PO	PO	PO	PO	WB			
I3	-	IF	-	-	-	-	-	-	-	WB			
I4	-	IF	-	-	-	-	-	-	-	WB	WB	WB	

13 clock cycles required

* Data dependency can be:

(A) RAW Hazard or True Dependency

ADD $\boxed{R_7, R_5, R_6}$
SUB $\boxed{R_8, R_4, R_7}$

$$\boxed{R_1 \cap P_2 \neq \emptyset}$$

(B) Anti-dependency (WAR Hazard)

ADD $\boxed{R_5, R_6, R_7}$ ← Will open
SUB $\boxed{R_7, R_4, R_3}$ ← Read

$$\boxed{D_1 \cap R_2 \neq \emptyset}$$

(C) Output dependency (WAW Hazard)

(A) No dependency (RAR Hazard)

ADD $\boxed{R_7, R_6, R_5}$
SUB $\boxed{R_8, R_6, R_4}$

$$\boxed{D_1 \cap P_2 \neq \emptyset}$$

What are the possible Hazards for the following instructions :

ADD $\boxed{R_5, R_6, R_7}$ → Domain (D1)
SUB $\boxed{R_5, R_5, R_4}$

$$\begin{array}{c} \xrightarrow{R_2} \\ \xleftarrow{D_2} \end{array} \quad \boxed{VI}$$

RAW: $R_1 \cap P_2 = \{R_5\} \cap \{R_5, R_4\} = \{R_5\} \neq \emptyset$
WAR: $D_1 \cap R_2 = R_6 \cap R_4 \cap R_5 = \emptyset$
WAW: $R_1 \cap R_2 = R_5 \cap R_5 = R_5 \neq \emptyset$

RAW & WAW hazards will be possible.

~~CONTROL DEPENDENCY~~

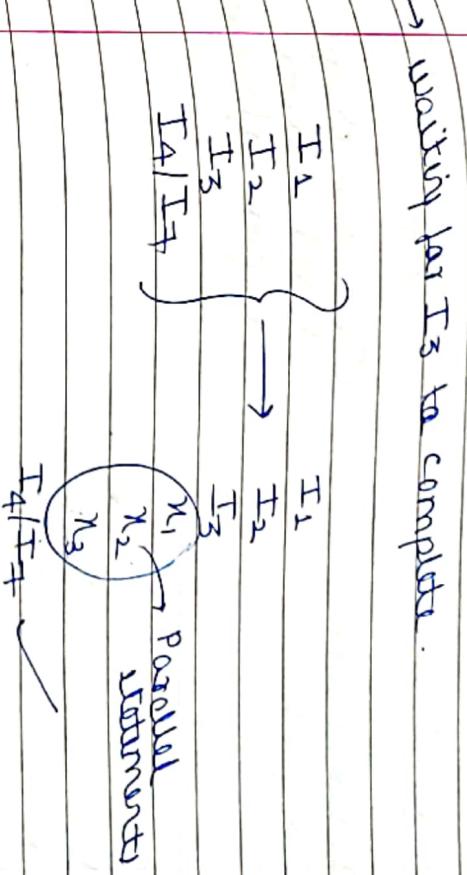
- Branch instruction leads to saturation
- dependency of Branch is not allowed
- the target of Branch is not evaluated until the branch instruction is executed completely.

Eq:

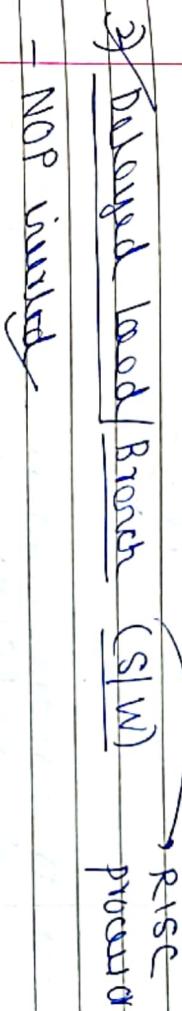
$$\left. \begin{array}{l} I_1 \\ I_2 \\ I_3: \text{BNZ } I_7 : \\ \downarrow I_4 \end{array} \right\}$$

\Rightarrow

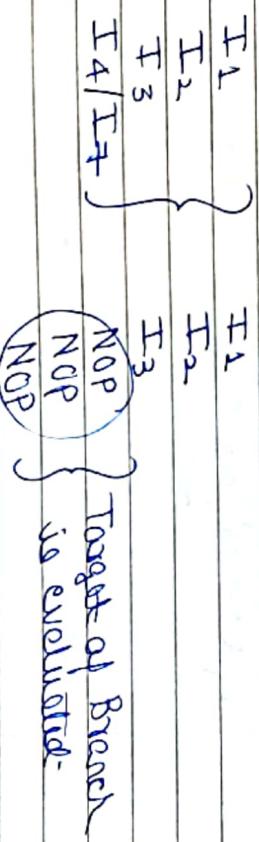
	1	2	3	4	5	6	7
I_1	IF	ID	OF	EX			
I_2	IF	ID	OF	(EX)			
I_3	IF	ID	OF	EX			
I_4/I_7	-	-	-	-	IF		



- waiting for Branch inst to complete its execution.



- NOP inserted



I_4/I_7

= Branch ↑, NOP↑, CC↑.

* The Solution can be:

1) Dynamic Scheduling (H/W or S/W)

→ waiting for I3 to complete.

3) Branch Production

* Control dependency

→ implemented using 2 techniques



[Production never changes]

[Production based on outcome of previous branch]

↳ without production

↳ with production

↳ without production

↳ Branch

↳ Never

↳ Prediction correct

↳ Prediction wrong

↳ Personal stat.

↳ STOP statements

↳ VI

[Target = I₂]

[Target = I₄]

→ If Production goes wrong, it leads to Branch

possibility (Stall cycle).
 If correct, no branch possibility.

(S₁ - S₅)

K = 5

(IF, ID, OF, EX, WB)

I₃: S₁ S₂ S₃ S₄ S₅

I₃: IF ID OF EX WB

I₄/I₂: — — — IF

I₄/I₂: — — — IF

I₄/I₂: — — — S₁

4 stall

cycle

open branch

always

3 stall

cycle

Consider an instruction pipeline with 5 stages. It allows overlapping of all instructions except branch type. Let there be 20% Branch instructions and the pipeline is operated with 800 MHz.

(Q) What is the speed up?

$$\text{Soln: } S = \frac{K}{(1 + \text{STALL}) * \text{STALL CYCLES}} \rightarrow K - 1.$$

$$\Rightarrow \frac{5}{1 + 0.2 * 4}$$

$$\Rightarrow \frac{5}{1.8} = \frac{2.7}{\text{avg. stall}} \quad \text{avg. stall} = 4$$

(Q) What is avg instruction time? throughput

$$\text{Avg.} = (1 + \text{STALL CYCLES} * \text{STALL CYCLES}) * t_p$$

$$f_p = 800 \text{ MHz}$$

$$t_p = 1.25 \text{ ns}$$

$$\text{Avg.} = (1 + 0.2 * 4) * 1.25$$

$$T_{avg} \rightarrow 2.25 \text{ ns}$$

$$10^9 \rightarrow 2.25 \text{ ns} \rightarrow 10^9$$

K = 5 (stall)
(with no prediction)

Defn of stall given [IF, ID, OF, EX, WB]

$$t_i \rightarrow 6 \text{ ns}, 7 \text{ ns}, 9 \text{ ns}, 8 \text{ ns}, 7 \text{ ns}.$$

$$t_d \rightarrow 1 \text{ ns}$$

Program:

I1

I2 : BNZ I9;

I4

Branch is taken

I9

I10

What is the time taken for 10 million instructions?

$$10^9 \rightarrow 2.25 \text{ ns} \rightarrow 10^9$$

L1: 32-bit word

Page No.:
Date:

4cc

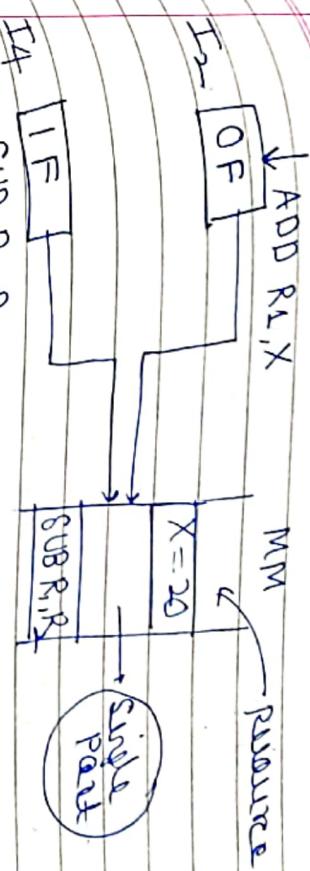
$R_1 \leftarrow R_1 + M[X]$
(direct)

Page No. 207
Date: 10/10/2023

$$T_p = \text{no. of clock cycles} * t_p.$$

$$t_p = \max(t_i) + t_d = 10 \text{ ns.}$$

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
I ₁	IF	ID	OF	EX	WB									
I ₂	IF	ID	OF	EX	WB									
I ₃	IF	ID	OF	EX	WB									
I ₄	IF	ID	OF	EX	WB									
I ₅	IF	ID	OF	EX	WB									
I ₆	IF	ID	OF	EX	WB									
I ₇	IF	ID	OF	EX	WB									
I ₈	IF	ID	OF	EX	WB									
I ₉	IF	ID	OF	EX	WB									
I ₁₀	IF	ID	OF	EX	WB									
I ₁₁	IF	ID	OF	EX	WB									
I ₁₂	IF	ID	OF	EX	WB									
I ₁₃	IF	ID	OF	EX	WB									
I ₁₄	IF	SUB	R ₁ , R ₂											



∴ more stores simultaneously request same source leads to resource conflict.
→ the solution can be:

(i) use of dual multi-parted memories
[H/W implementation]

(ii) use of separate instructions & data memories.

[1 module → OF; another → IF]

(3) STRUCTURAL DEPENDENCY

(Resource Conflict)

H/W provides
same

(iii) Resource duplication (Default)

Providing multiple copies of same resource

→ With these H/W Solutions, no stall cycles.

Lesson Part: Only 1 unit can access mem.



both will add.

: Resource duplication

* INSTRUCTION FORMATS AND ADDRESSING MODES

RISC instruction

manipulates data in registers only.

[Fast execution]

$$\Rightarrow \text{ADD } R_1, R_2, R_3 \\ R_1 \leftarrow R_2 + R_3$$

(imp)

+ Basic Instruction Format:

Opcode	Address part
4	16 bit

i → 16 bit → 12

→ 12 location

16 operations

→ direct address

instruction

⇒ ADD R₁, X, Y
 $R_1 \leftarrow M[X] + M[Y]$

Opcode	Mode	Address part
4	3	9

(23 modes).

ADD L0(R₁) #300

Text

Pg 159

Opcode	M ₁	A ₁	M ₂	A ₂
1011 0'10	101	Binary	101	Binary

one address

ADD X

AC ← AC + M[X]

→ zero address

unification

(CPU)

translator

Stack
(Instruction)

ADD
both operands
from top of stack

* The various Addressing Modes are :

1.) (000)
2. } Register indirect (001)
3. } Scaled Register indirect (010)
4. } given (011)
5. } Direct Absolute (100)
6. } Indirect (101)
7. } Zero addressing (110)
8. } Displacement Addr. (111)

8. Displacement Addressing Modes :

$EA = Address\ Post + Register$

Address offset (displacement)
 Address

→ memory address plus Δ Address operation.

→ provides port execution
 (from Indirect)

(i) Relative Addressing Mode:

$EA = Offset + PC$

Branch control
 Statements.

$PC \rightarrow 400:$
 $399: BNZ 50.$
 $= 450$

(ii) Indexed Addressing Mode:

$$EA = Address + IR \leftarrow$$

Base Address Index
 @ Array. Register

(iii) Base Register Addressing Mode:

$$EA = offset + BR \leftarrow$$

(Base Add of Memory).

(allocatable code | Pastn code |
 word in structures/records/ subprogram
 declaration)

[3 Type of AS]

Q) Watch the following:

Start student S1;

~~use $x = 1;$~~ \rightarrow auto inc dec
 ~~$A[i] = B[j];$~~ \rightarrow indirect
~~while ($A^{*}++$)~~ \rightarrow base register
~~int $*P = &A[0];$~~ \rightarrow immediate
~~displacement~~

~~(i)~~ consider a word instruction:

ADD R1, @X, #Y

indirect.

Register

displacement

- (ii) No. of memory references for accessing data

R1 \leftarrow @X + #Y

①
②

: 3 memory references

- (iii) No. of memory references for execution
- compilation

$$2 + \textcircled{1} + \textcircled{2} = 5$$

2 word Instruction Fetch

(iv) $2 + (2 + 1 + 2) = 6$.

ADD 10(R1), @X .

10(R1) \leftarrow 10(R1) + @X

①

②

③

3 memory references

$$3 + 2 = 5$$

32 bit instruction

32 registers

20 operations

8 addressing modes

512 mem. locations

3 address

instruction.

ADDI R1 300 10

Reg. Add. Value

32 reg. 5 bits

3

[Opcode | M1 | A1 | M2 | A2 | M3 | A3]

20 op modes

3

= 5 bits

3

= 5 bits

3

= 5 bits

3

= 5 bits

3

~~N = 4 bits~~

2 word @ X, #Y

① X \leftarrow @X + #Y

②

③

4 Mem. references.

Known: 0 0 0 1 5

Signed: -8 to +7 (less comp.)

⇒ If explicitly ad. mode not given



Unsigned: → 0 to 2¹³ - 1

Cyclic → -2¹² to 2¹² - 1

* DATA REPRESENTATION

→ Fixed point representation.

→ floating

→ generally used for small range of values in the computer system

- integer data.

1. Signed magnitude form }

2. One's complement form }

3. Two's complement form



- Represents
 - Range with K bits
 - overflow condition.
- 100% IMOS.

2. Floating point Representation

- large range of values

- most useful data.

(±m × f_c) → exponent
mantissa → radix

Eq: 32.625 × 10⁴.

Eq: 1.101101 × 2⁶

mantissa

→ Radix/Base is assumed.

④ Normalized Floating point number

→ 0; $\underbrace{101101}_{\text{mantissa}} \times 2^{10}$

Mantissa

+ 0.1bbb...b * 2^e

position of radix point is assumed.

MSB of mantissa → 1.

4

$$\Rightarrow (-1) * (1 \cdot 01111111) * 2^3$$

$$\Rightarrow -(10111.111)_2$$



$$-(\underline{3 \cdot 8+5})_{10} \cdot \frac{\text{Ans}}{10}$$

What is max value in single precision?

$$S=0$$

$$E=254$$

$$M=1111 \dots 1$$

$$\xleftarrow{23} \quad \xrightarrow{23}$$



Control Signals

managed by

CU

Everything designed
at time of design

Instruction Microop's Control Signal

	IF	PC	PC
I1	IF	PC \leftarrow PC + 1	PCOUT
I2	ID		
I3	OF		
I4	EX	READ	MARIN
I5		IRE-MDR	

* CONTROL UNIT DESIGN

- The purpose of Control Unit is to provide appropriate timing & control signals.

bus datapath.

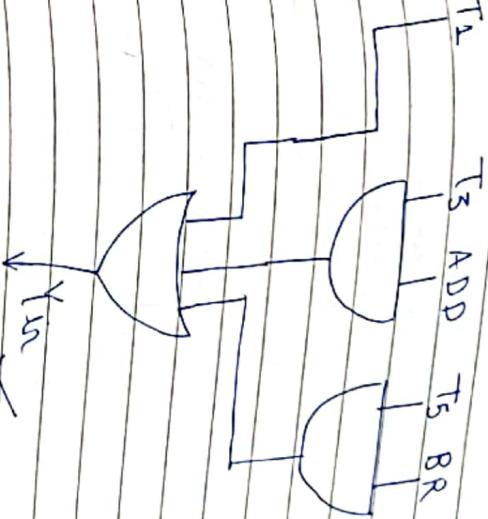
NE

- ① Hardwired CU
- ② Microprogrammed CU

* HARDWIRED CONTROL UNIT DESIGN

- Control Functions are implemented in HW.
- Every Control Signal is represented in SOP expansion and realized using digital hardware.

e.g: PCin, MARout \rightarrow separate SOP expression.



* Micro-Programmed Control Unit Design

- Control functions are implemented in SW.
- Every control signal is generated after decoding micro-instruction.
- Very flexible to accommodate changes.
- Effective for large no. of instructions & control signals.
- Relatively control signals are generated slow.
- Implemented in CISC Processors.
- Control Memory is ROM.

Storage \rightarrow Fetching \rightarrow decoding

- e.g.: $Y_{in} = T_1 + T_3 \cdot ADD + T_5 \cdot BR + \dots$
- Here, Y_{in} gets encoded; during T_2 for all instructions, during T_3 for ADD and so on.
- \rightarrow It can be realized as:

