Name:- Md Farhan Ishmam

ID:~~Rohit~~ - 1800411 20

Sec: CSE-A

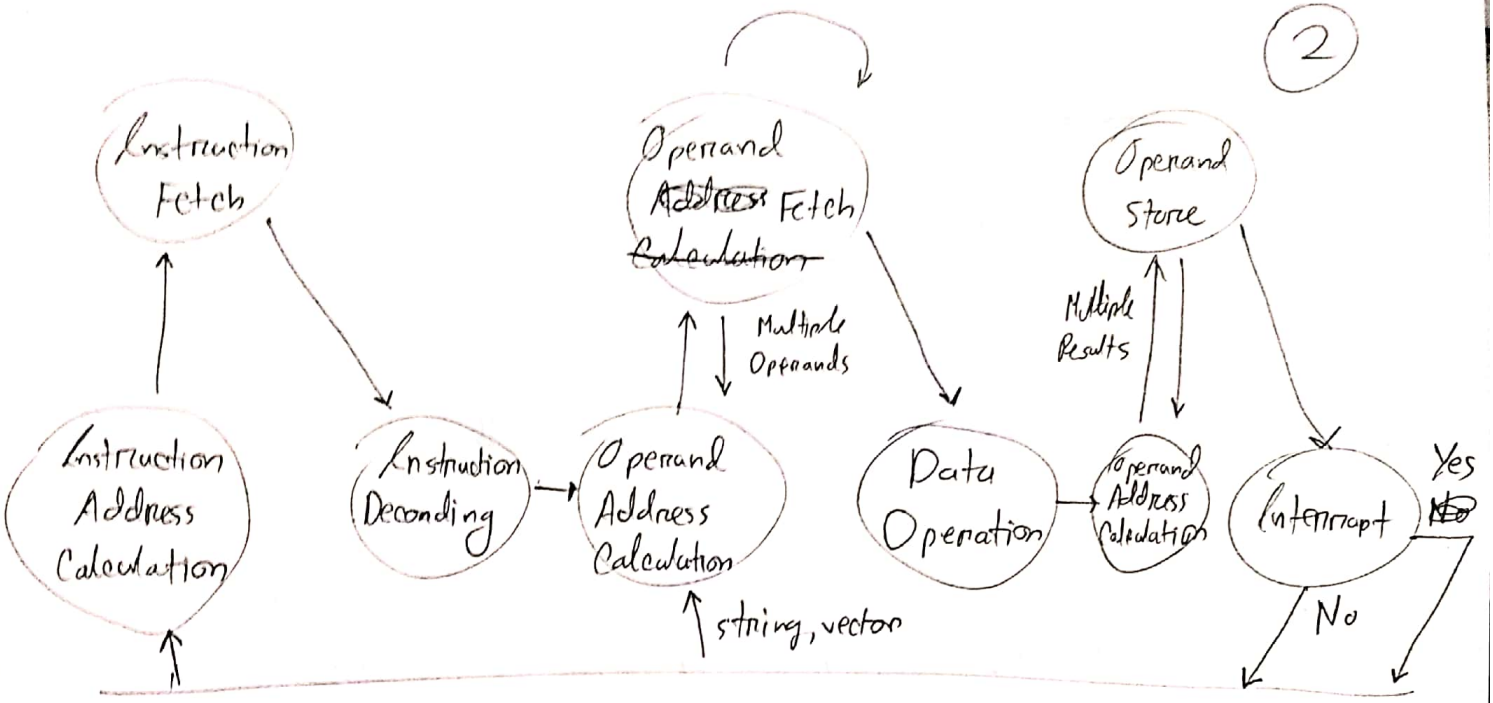
Course: CSE 4305

Student Signature: Ishmam

Ans. to Q.no. 1

Micro-operation: Micro-operation is the smallest unit of instruction for a processor. A complete operation is composed of a number of micro-operations. For example - calculating the address of operand, or storing a data. ~~is stored~~

Micro-instruction: A set of micro-operations together makes a micro-instruction. It is the smallest unit of an instruction given by the processor. For example the CPU ~~operation~~ instruction of ~~LOAD A will take 4 micro~~ ~~taking the~~ fetching data will take 4 micro instructions.



Here, ~~for~~ fetch cycle,

t_0 MAR ← (PC)

t_1 MBR ← Memory

t_2 PC ← (PC) + 1

t_3 IBR ← (MBR)

Ans. to Q.no, 6

$$Y = (A \times B) + (C \times D) + E$$

We need to use one address format.

```
LOAD  C
MUL   D
STORE Y   (Temporary register)
LOAD  A
MUL   B
ADD   E
ADD   Y
STORE Y
```

Ans. to Q.no. 8

In an interrupt driven I/O, we can not know from which I/O device the interrupt signal is given. Hence it is difficult to send the acknowledgement signal by the CPU. To tackle this design issue 4 solutions are given

(i) Multiple interrupt Lines: We take multiple lines from the I/O devices directly to processor. Then we can determine the source of interrupt. But lines directly to processor is limited and costly. This process is simple but not feasible.

(ii) Software Poll: A software can poll all the number of I/O devices and can trace back which device has send the I/O signal This can take longer time and will make processor wait.

(iii) Daisy Poll / Hardware Poll: Instead of software poll, a seperate harwired line is given which will be used by the interrupting I/O device. This is like software poll but done in hardware implementation and will store time.

(iv) Bus arbinarty: Here a bus is used as vector where the control of bus is taken by the I/O device generating interrupt signal.

Ans. to Q.no. 9

(i)

Disk capacity = No. of surface × Tracks_per_surface × sectors_per_track × sector size

$= 10 \times 600 \times 72 \times 512$ B

$= 221.184 \times 10^6$ Bytes

= 221.184 Mb (Ans.)

(ii)

Average access time = average seek time + average rotational_delay

$= 10\text{ ms} + \frac{1}{2r}$ [$r = 3600$ rpm $= 60$ rps

$= 10\text{ ms} + \frac{1}{2 \times 60}$

$= 10\text{ ms} + 8.33\text{ ms}$

$= 18.33$ ms. (approx.) (Ans.)

(iii)

The size of a cylinder is $512 \times 72 \times 10$

$= 368.64$ Kb

No. of bytes in a track is 512×72

$= 36.864$ Kb

To access a 5 MB file, we need,

$$13.56 \approx 14 \text{ cylinders}$$

The time required to access first cylinder is

$$T_1 = T_{seek} + T_{Rotation} + T_{transfer}$$

$$= 10ms \times \alpha + \frac{1}{2r} + \frac{b}{r N}$$

[b = no. of bytes in 6 to be transferred,
N = no. of bytes in a track
r = Rotational speed = 60 rps]

$$= 18.33 + \frac{368.64}{60 \times 36.86}$$

$$= 18.33 + 166.67 \text{ ms}$$

$$= 185 \text{ ms.}$$

For the 13 other cylinders we need

$$T_{rest} = 1.5 \times 13 + \frac{368.64}{60 \times 36.84} \times 13$$

$$= 2186.21 \text{ ms.}$$

$$T_{total} = (185 + 2186.21) \text{ ms}$$

$$= 2371.21 \text{ ms}$$

$$= 2.37 \text{ seconds (Ans.)}$$

(6)

Ans. to Q.no. 7

The TLB and ~~fetch~~ cache are used to fasten the fetch operation in our virtual memory system.

In the virtual memory, the process is divided into pages. The TLB takes this page number of previous operations. Now there is a probability of being the same page which will cause TLB hit. TLB hit will save some time and bypass the path.

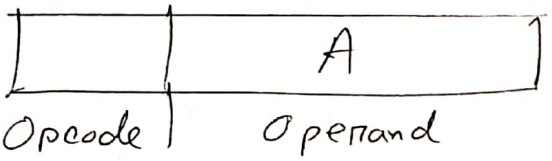
After TLB hit the ~~operations~~ data are added from page table and offset. The task of of TLB is to compare the page number of the virtual address.

In cache again, the real address is compared with address stored in cache. Cache is fasten than main memory. It stores limited data. If cache hit occurs then the data can be easily loaded from cache, very fast. If cache miss occurs then it has to be loaded from main memory. Thus using TLB and cache, we can cache and load data fasten than main memory. This is importance of caching. It saves time. ~~as~~

Ans. to Q.no. 5

(i) Direct.

In case of direct ~~operation~~ mode, operand is given directly in the instruction



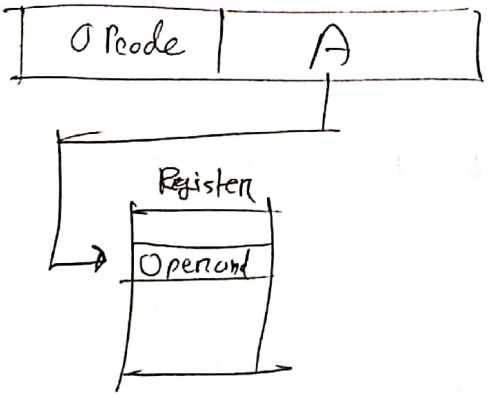
Here, Operand = A

Here, no effective address is required. It is the memory address 200 ~~operand~~ itself. (Ans.)

The loaded operand will be MM. (Ans)

After completion PC will take next instruction WXYZ.

(ii) Register.



∴ E.A = (R1)

Here E.A address is the content of register R1 which is 400,

∴ E.A = 400. (Ans.)

The location 400 contains 1000.

So, the operand is 1000 (Ans.)

After completion PC will load next instruction.

Ans. to Q.no. 2

(i) 11110000

(-) 00010100

~~11011100~~

Operation is done using 2s complement

```
                0 0 0 1 0 1 0 0
                ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓
1's complement  1 1 1 0 1 0 1 1
2's   "                     + 1
-------------------------------
                1 1 1 0 1 1 0 0
```

```
Carry = 1 →  (1) 1 1 1 1 0 0 0 0
             (+) 1 1 1 0 1 1 0 0
             -----------------------
                 1 | 0 | 1 0 1 1 0 0
                   ↑   ↑
              Overflow  Sign
```

Here, (iii) Overflow = 1 (Ans.)

(iv) Sign = 0 (Ans)

(i) Carry = 1 (Ans.)

Ans. to Q no. 4

Given, 2 stage pipeline is implemented

(i) $S = \frac{nk}{n+k-1}$

n = no. of instructions
k = pipeline

=

Bus fetches the instruction in the same time as

(ii) EU.

So, $S = 1$ (Ans.)

(ii) The five instructions will be instruction fetch (IF)
~~decode~~ instruction decoding

(10)

(iii) $S = \frac{nk}{n+k-1} = \frac{100 \times 2}{100 \times 2 - 1} = 2$ (Ans)

Ans. to Q.no. 3

Pipeline Hazards are resource hazards, data hazards and control hazards,