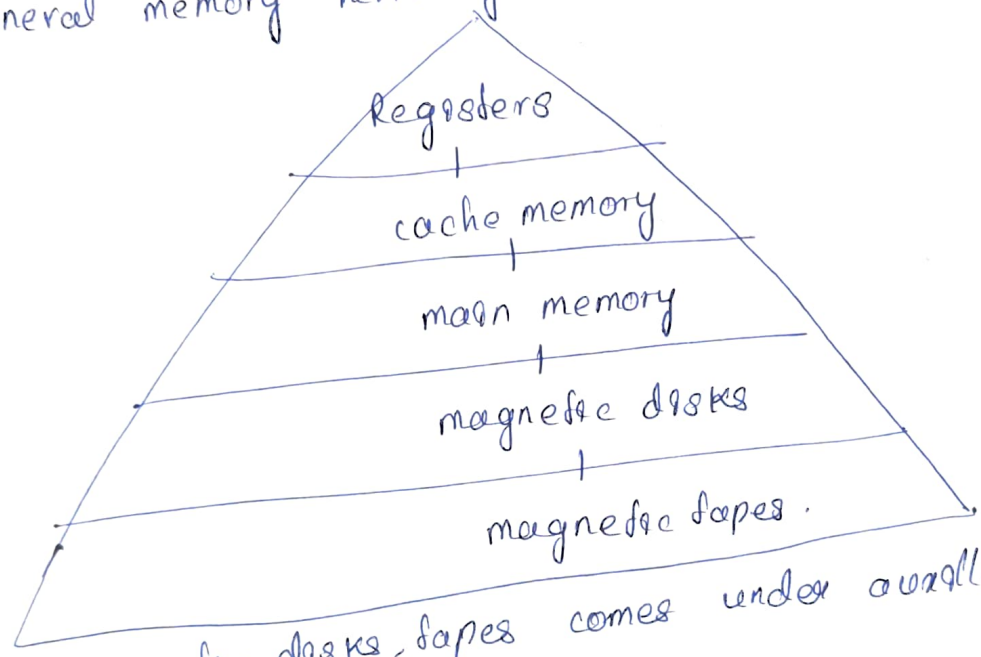
UNIT-V

(I) **memory hierarchy** :- memory devices are used to store the data, instructions, information.

- memory devices can perform only read and write operations.
- different types of memory devices are :- main (or) primary memory, secondary (or) storage devices.
- general memory hierarchy is :-



- where magnetic disks, tapes comes under auxillary (or) secondary storage devices.
- In the above hierarchy, speed is decreased when we move from top to bottom.
- storage capacity is increased from top to bottom.
- cost is decreased from top to bottom.

memory access methods :- following are methods to access information from memory locations —

(1) Random access :- In this each memory location has unique address, using this address any memory location can be reached in same amount of time in any order.

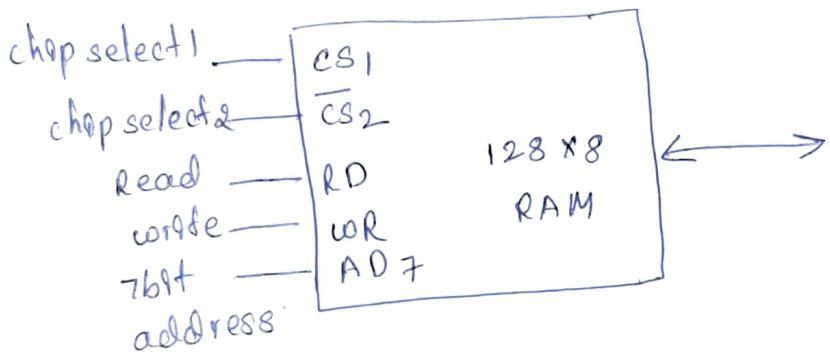
b) sequential access :- memory is accessed sequentially.

c) direct access :- in this method, information is stored in tracks, with each track having seperate read/write header.

(ii) main memory :- it communicates directly with CPU.

- It is central storage unit of computer.
- It is large and fast memory used to store data during computer operations.
- main memory is made up of RAM and ROM.

RAM (Random access memory) :-



- It is random access memory i.e. directly searches for element.
- It is voladile memory, used for both read & write operations.

- types are — static, dynamic.

static RAM → Data is stored in transistors and requires a constant power flow, because of continues power, SRAM doesn't need to be refreshed to remember data being stored.

- binary data presend here will be stored until power is off.

- The word static indicates that memory contains data

as long as power is supplied.
- data is lost when power gets down due to volatile nature
- SRAM chip uses matrix of 6 transistors and no capacitors
- transistors do not require power to prevent leakage, so SRAM not be refreshed on regular basis.
- SRAM uses more chips than DRAM for some amount of storage space, making manufacturing cost high.

- characteristics of static RAM →
  - long life
  - no need to refresh
  - faster
  - used as cache
  - large size
  - expensive.

advantages → low power consumption, faster access.
dis advantages → less memory capacity, high cost.

DRAM (dynamic Random access memory) → it must be continuously refreshed in order to maintain the data.
- it is used in many systems as it is cheap & small.
- characteristics of DRAM →
  - short data lifetime
  - needs to be refreshed continuously.
  - slower compared to SRAM
  - used as RAM, smaller size, less expensive.

advantage → low cost, more memory capacity.
dis advantage → slow access, high power consumption.

SRAM
- Transistors r used to store information in SRAM
- capacitors r not used hence no refreshing is required.
- faster compared to DRAM
- expensive
- low density devices
- used in cache

DRAM
- capacitors r used to store data in DRAM.
- To store information for a longer time, contents of capacitor needs to be refreshed periodically.
- slow
- cheap
- high density.
- used in main memory.

ROM (read only memory) :- used only to read not to write.

- It is non volatile memory.
- It generally stores such instructions that r required to start a computer.
- ROM chips r also used in electronic items like washing machines, ovens etc.
- data is inserted at the time of manufacturing.

types of ROM :- PROM, EPROM, EEPROM.

PROM (programmable read only memory) :- It can be modified only once by user.

- user buys a blank PROM and enters required data using PROM program.
- It can be programmed only once and is not erasable.

EPROM (Erasable and programmable Read only memory) :- It can be erased by using UV (ultra violet) light for duration upto 40 minutes.

- we can erase and write new content by multiple times.
- written instructions can be erased and changed with help of uv light.

EEPROM (electrically erasable and programmable Read only memory) :-

- data can be erased electrically.
- It can be erased and reprogrammed about ten thousand times.

Advantages of ROM :- Non volatile in nature, cheaper than RAM, easy to test, reliable than RAM, static & no need of refreshment.

Auxiliary memory :- most common auxiliary memory devices used in computer system are magnetic disks and tapes.

(a) magnetic disks :- explain about hard disks.

(b) magnetic tape :- data is read/write sequentially.

- only one side of ribbon is used for storing data.
- It is a sequential memory which contains thin plastic ribbon to store data and coated by magnetic oxide.
- data read/write slower because of sequential access.
- it is highly reliable which requires magnetic tape drive writing and reading data.

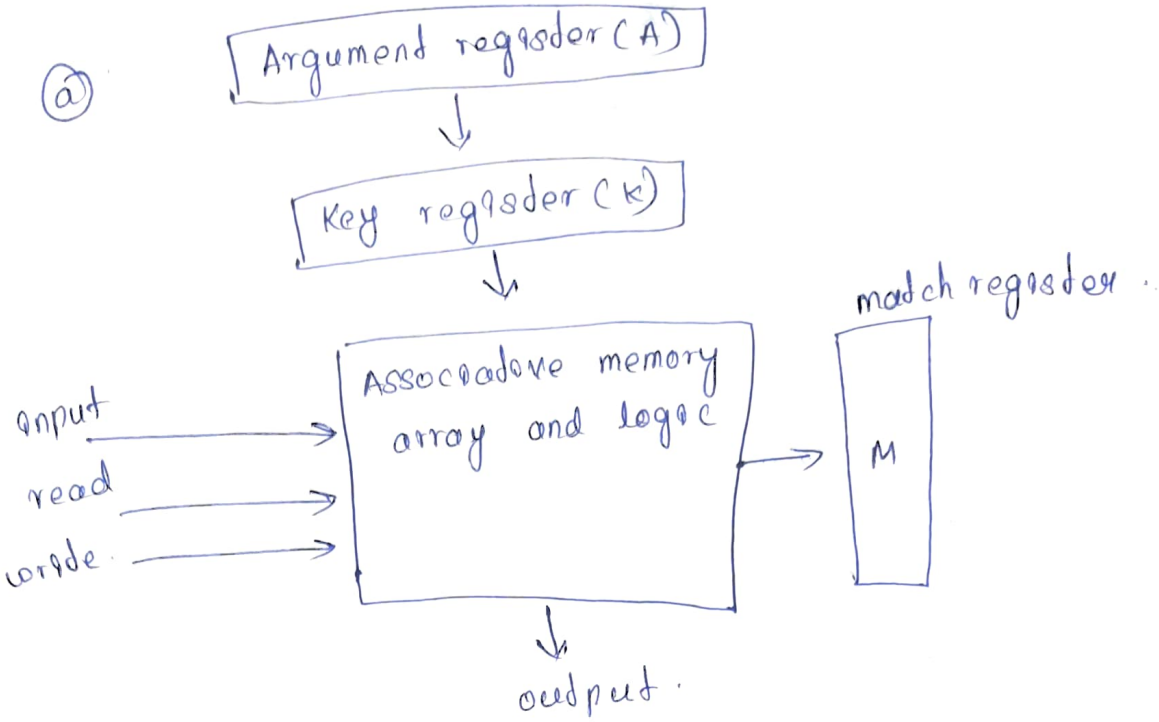
advantages :- low cost, provides backup or archival storage, reusable memory used for large files,

disadvantages :- sequential access, data stored cannot be easily updated (or) modified. i.e. difficult to make updates on data.

Associative memory :- it can be considered as memory unit whose data stored can be identified for access by content of data rather than address.

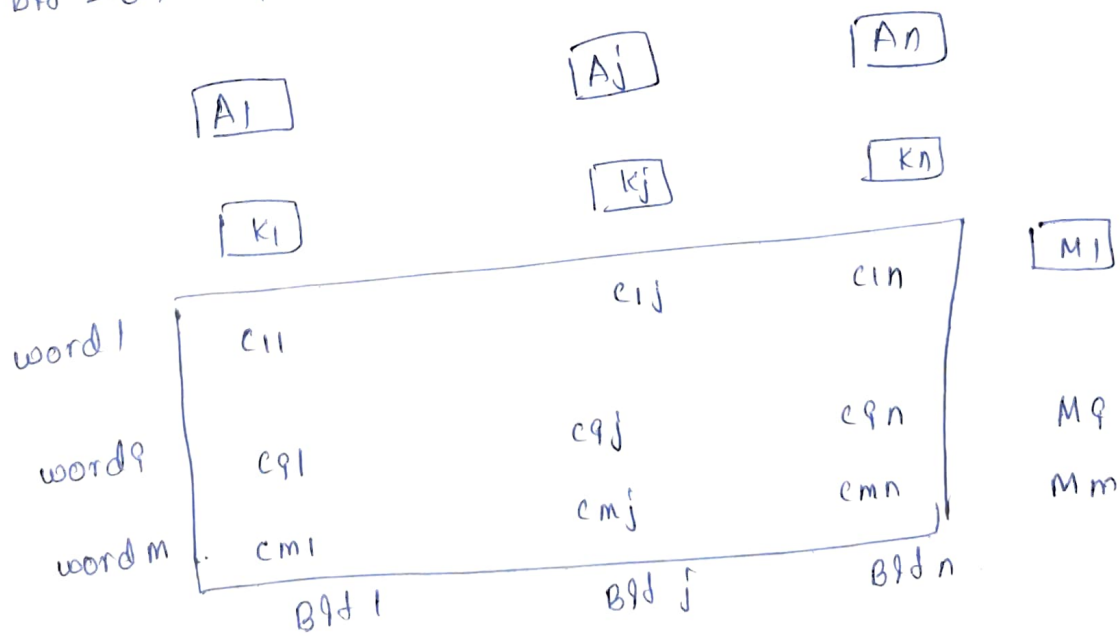
- it is referred as content Addressable memory (CAM).
- when write operation is performed, no address or memory location is given to word.
- the word itself is capable of finding empty location to store word.
- In read operation, content of word is specified, the word matching with specified content are selected for reading.

(a)

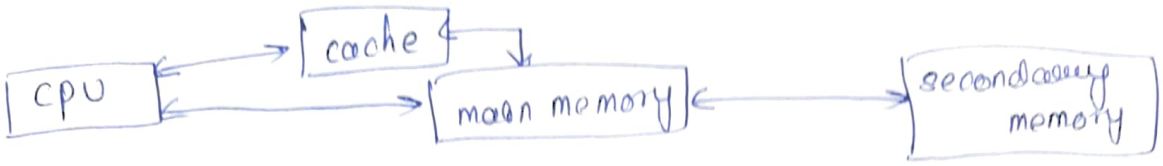


① Argument register (A) contains part of word.

② key register contains information, if bit = 1, will be compare Argument value and word value in memory, if bit = 0, comparision neglected.



cache memory :-



- cache is the fastest memory device.
- whenever processor requires any data, first it will search for data in cache, if not found goes to main memory, if found, a copy will be sended to cpu and a copy to cache,
- if data is not at main, then system searches for data in secondary, if found copy will be sended to main memory.
- it is very costly so we take less storage capacity.
- the data (or) contents of main memory that r used again and again by cpu are stored in cache, so that it can easily accessed in short time.

Hit ratio :- The performance of cache memory is calculated in terms of quantity called Hit ratio.

Hit ratio ⇒ whenever cpu refers to memory and if it finds that data in cache it is said to be hit.

- if data not found, it is known as miss.
- The ratio of number of hits to total cpu references to memory is called hit ratio.

$$\text{Hit ratio} = \frac{\text{Hit}}{\text{Hit} + \text{Miss}}.$$

- cache performance depends on hit ratio, we can improve cache performance using higher cache blocks size, higher associativity, reduce miss rate, and reduce time to hit in cache.

cache mapping :- Three different types of mappings used.

1. Direct mapping 2. Associative mapping 3. Set associative mapping.

(a) Direct mapping :- It is simple, less expensive to implement.

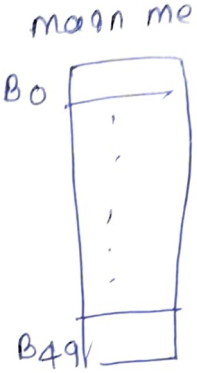
① certain block of main memory could be able to map cache only up to a certain line of cache.

cache line no = (Address of main memory Block) mod (Total no. of lines in cache)

Eg :- Let

cache = 0, 1, 2, 3

has 4 lines of blocks.



① If B0 has to be placed in cache, then (= k mod n).

formula =

block no = B0 mod 4

= 0 mod 4 = 0

∴ B0 has to be placed at 0.

If B3 has to be placed, then

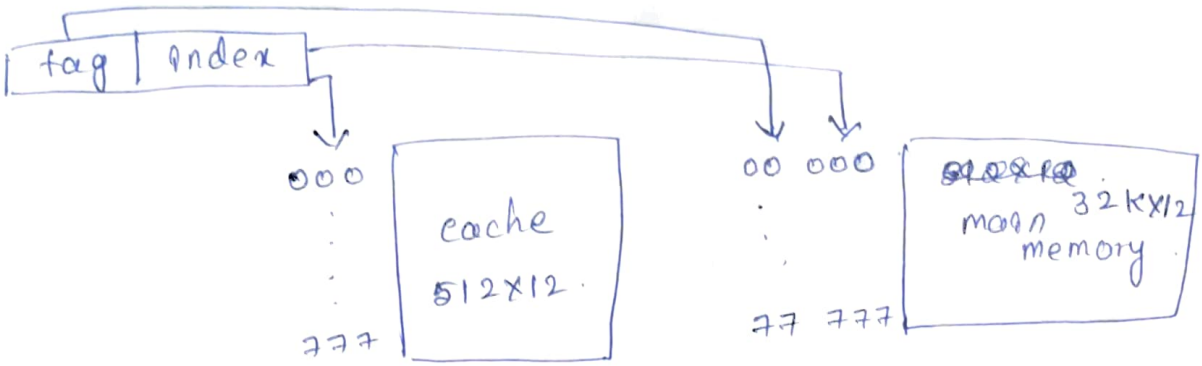
block no = B3 mod 4

= 3 mod 4

= 3

B3 has to be placed only at 3rd block of cache.

- address is divided into 2 parts i.e. tag and index.



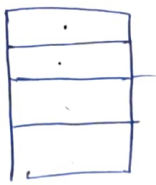
Associative mapping :- The main memory block is capable of mapping to any given line of cache, which is free at that particular moment.

- It helps us make a fully associative mapping comparatively more flexible than direct mapping.
- searching for content done with content only. like associative memory.

Set associative mapping :- It is the mixture of both direct and associative.

- cache is divided into number of sets.

Eg:- 2 way set associative



- if any memory block has to be mapped with cache, then sets r selected as direct mapping i.e.

= (Block of memory) mod (set numbers)

If B_4 has to be placed then

set no = $4 \% 2 = 0$

- block B_4 has to be placed at S0, but on S0, it can be placed any where (on any block), like associative.

Types of cache memory :-

L1 :- It is first level of cache, called as Level 1 cache (or) L1 cache, here small amount of memory is present inside the CPU itself.

- As this memory is present on CPU, it can work at same speed as CPU.
- The size of memory ranges from 2KB to 64KB.

L2 :- This level 2 cache may be inside (or) outside CPU.

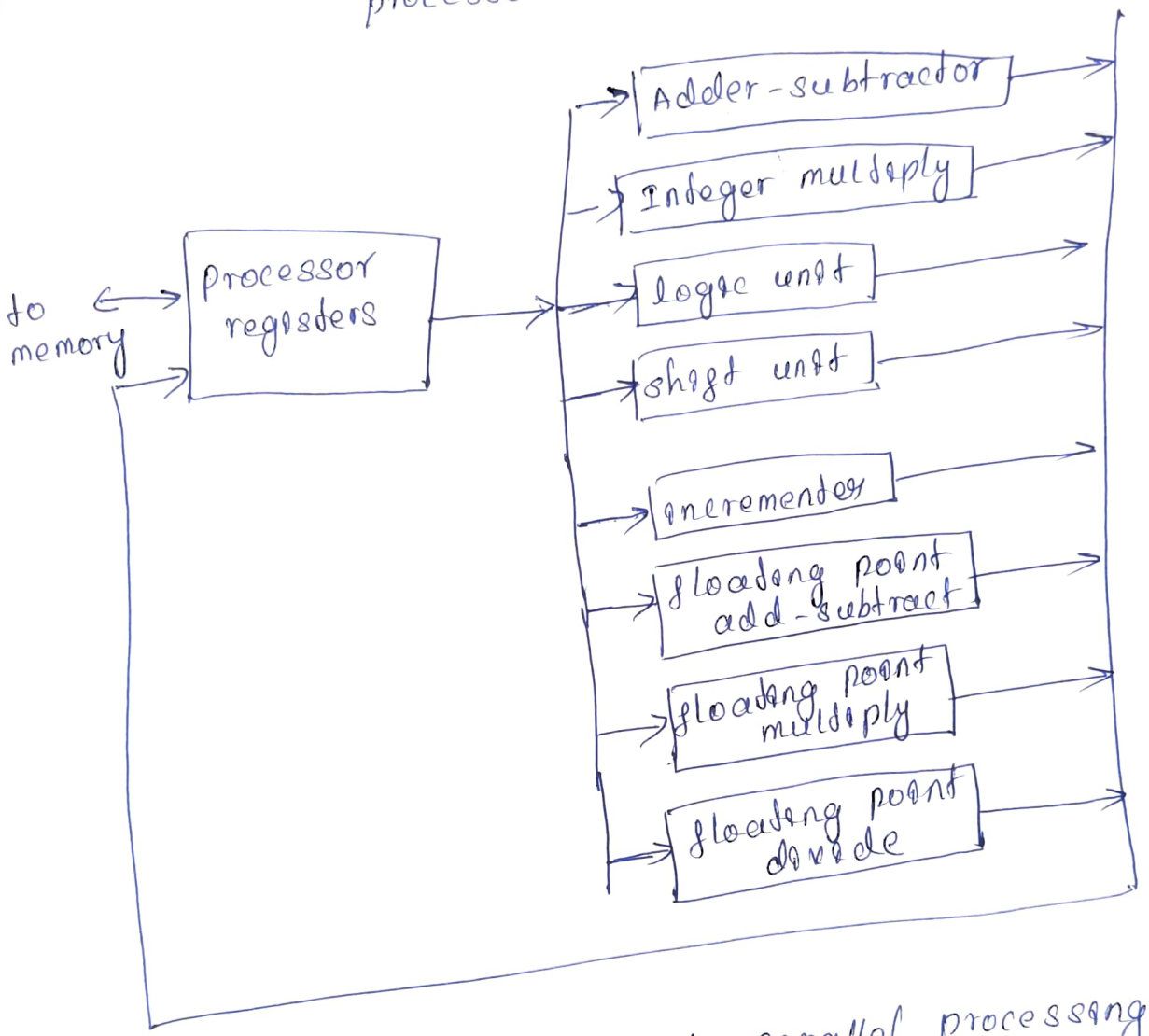
- In case if is outside of CPU, it is connected with CPU with very high speed bus.
- All the cores of CPU can have their own seperate level 2 cache (or) they can share one L2 cache among themselves.
- The memory size of this cache is in range of 256 KB to 512 KB.
- these are slower than L1 cache.

L3 :- This is not present in all processors.

- this is used to enhance performance of Level and level 2 caches.
- It is located outside of CPU and is shared by all cores of CPU.
- memory size ranges from 1MB to 8MB.
- slower than L1 and L2 cache, but faster than RAM.

Parallel processing is a term used to denote large class of technique used to provide simultaneous data processing tasks to increase speed of computer.

- It can be achieved by using multiple functional units.
- processor with multiple functional units



- There are different ways that parallel processing can be classified.
- one classification introduced by M. J. Flynn.
- Flynn's classification divides computers into 4 major groups as follows:

Single instruction stream, single data stream (SISD)
single instruction stream, multiple data stream (SIMD)
multiple " " , single " " (MISD)
multiple " " , multiple " " (MIMD)

- Instruction stream → sequence of instructions read from memory
- data stream → operation performed on data in processor.

- SISD → represents a system with control unit, processor and memory. instructions are executed sequentially.
- here parallel processing can be achieved with multiple functional units (or) by pipeline processing.

- SIMD → contains many processing units under common control unit; at a time only one instruction fetched but different operations on different data.

- MISD → only theoretical interest no practical system has constructed with this concept.

- MIMD → ~~capable~~ this systems capable of processing many programs at a time. Most multiprocessor and multi computer systems comes under this concept.

Pipelining :- parallel processing can be done with help of pipelining also.

- It as a technique of dividing sequential process into suboperations.
- each subprocess executed in a special dedicated segment that operates concurrently with all other segments.
- It as a collection of segment, where each segment performs fixed tasks.

Eg :- $A_i * B_i + C_i$ for $i = 1, 2, 3, \ldots 7$.

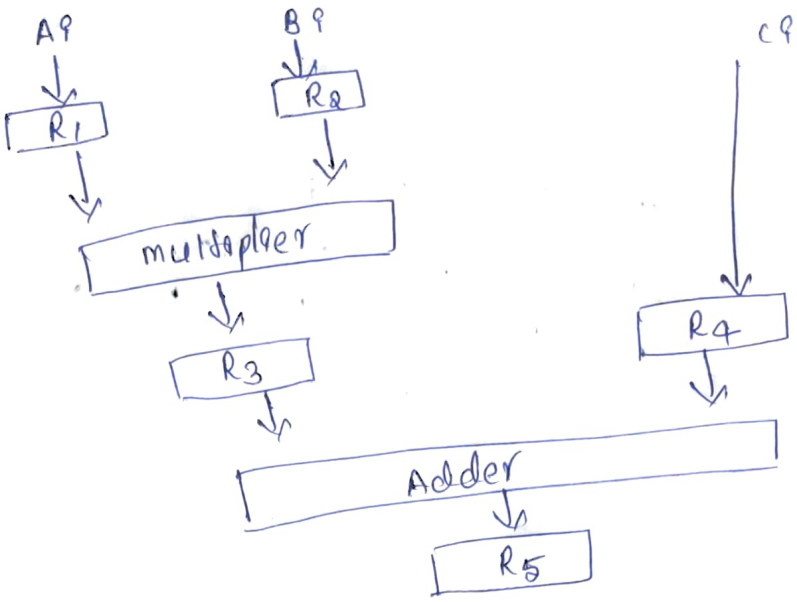
is performed with operations :-

$R_1 \leftarrow A_i$, $R_2 \leftarrow B_i$

$R_3 \leftarrow R_1 * R_2$, $R_4 \leftarrow C_i$

$R_5 \leftarrow R_3 + R_4$.

(a)

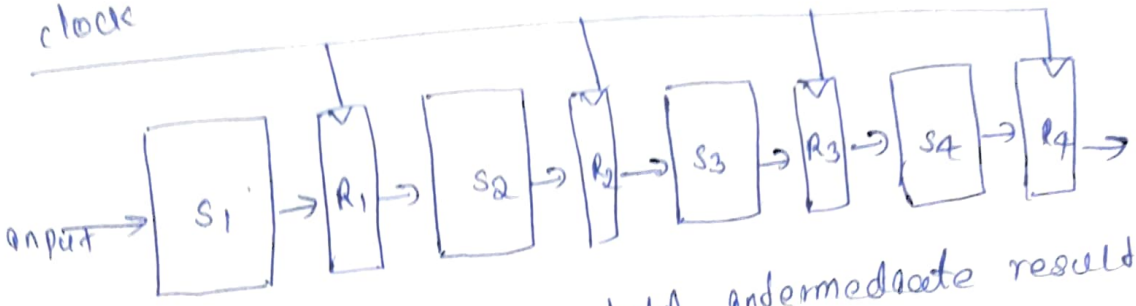


(b) content of Registers in pipeline eg.

clock Pulse number	segment 1 R1	R2	segment 2 R3	R4	segment 3 R5
1	A_1	B_1	$A_1 * B_1$	C_1	$A_1 * B_1 + C_1$
2	A_2	B_2	$A_2 * B_2$	C_2	$A_2 * B_2 + C_2$
3	A_3	B_3	$A_3 * B_3$	C_3	$A_3 * B_3 + C_3$
4	A_4	B_4	$A_4 * B_4$	C_4	$A_4 * B_4 + C_4$
5	A_5	B_5	$A_5 * B_5$	C_5	$A_5 * B_5 + C_5$
6	A_6	B_6	$A_6 * B_6$	C_6	$A_6 * B_6 + C_6$
7	A_7	B_7	$A_7 * B_7$	C_7	$A_7 * B_7 + C_7$
8					
9					

9 clock cycles.

(c) 4 segment pipeline



S = segments, R = registers to hold intermediate result.

(i) The behaviour of pipeline can be seen with space-time

(a) diagram

(i) state space diagram of a 4-segment pipeline is.
(horizontal axis displays time in clock cycle,
vertical axis gives segment number.)

T = task.

(d) space time diagram for pipeline

segment:	1	2	3	4	5	6	7	8	9	→ clock cycles
S1	T1	T2	T3	T4	T5	T6				
S2		T1	T2	T3	T4	T5	T6			
S3			T1	T2	T3	T4	T5	T6		
S4				T1	T2	T3	T4	T5	T6	

(i) 6 different tasks T1 to T6.

(i) segments ⇒ S1, S2, S3, S4.

(i) clock cycles ⇒ 9.

○ there are 2 areas of computer design where pipeline is applicable.

Arithmetic pipeline,
Instruction pipeline.

ⓐ Arithmetic pipeline is usually found in very high speed computers.

ⓘ They r used to implement floating point operations, multiplication of fixed point numbers.

ⓘ

eg:

$X = A \times 2^a$

$Y = B \times 2^b$

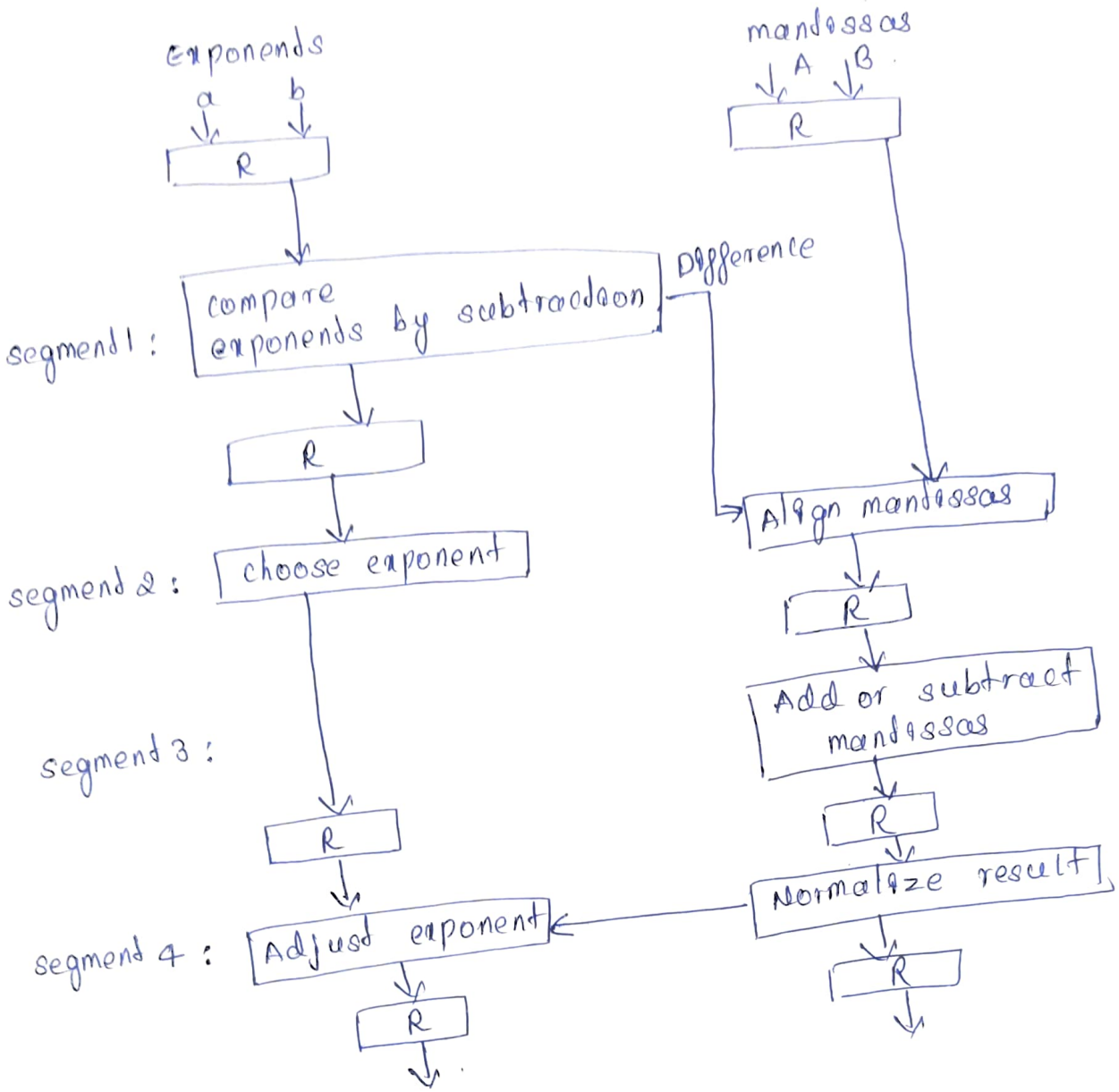
A, B = mantissas.
a, b = exponents.

ⓘ floating point addition and subtraction can be performed in 4 segments.

1. compare exponents.
2. Align mantissas
3. Add or subtract mantissas
4. Normalize result.

ⓐ

(a) pipeline for floating point addition and subtraction.

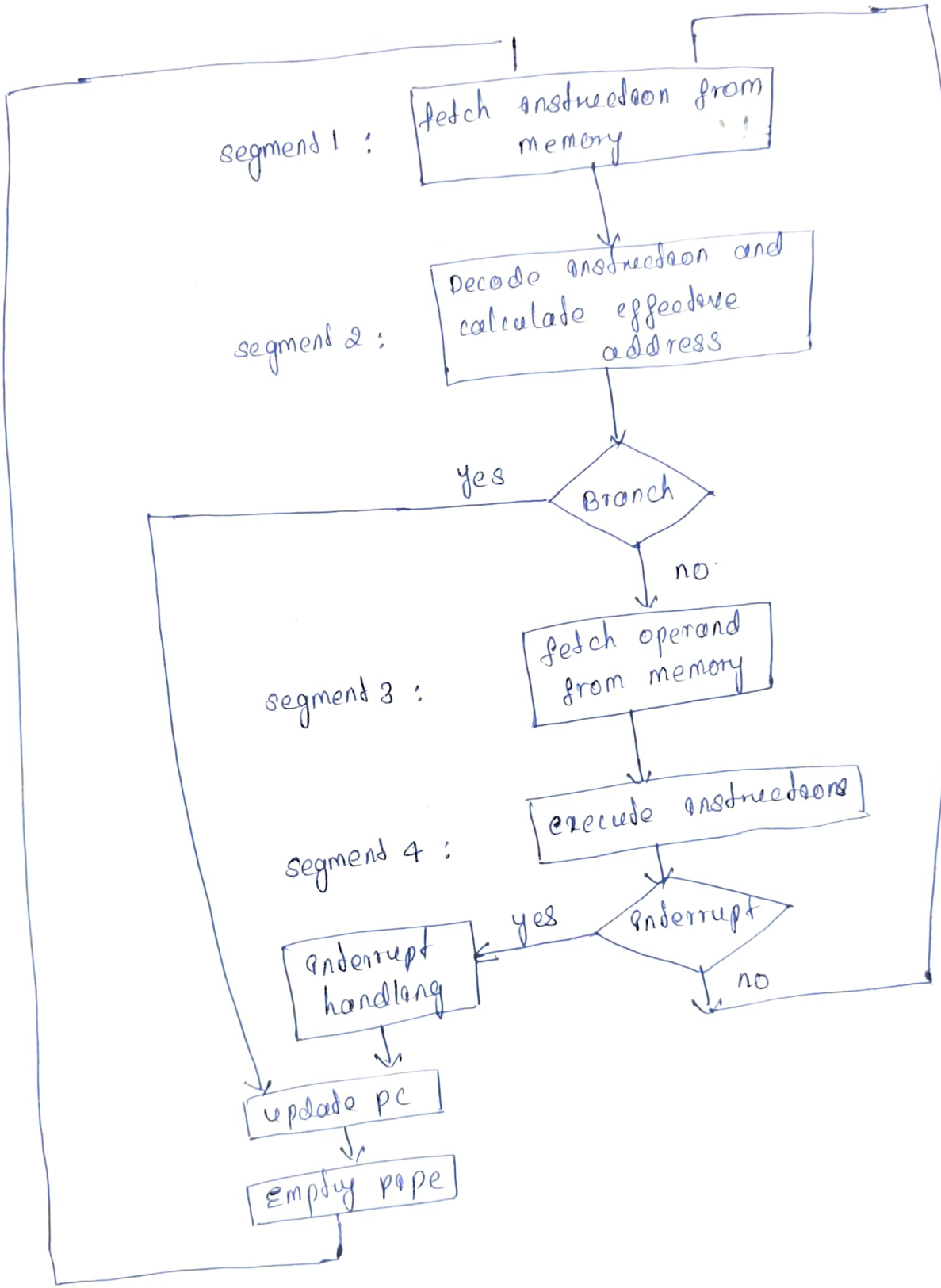


Instruction pipeline is pipeline processing can occur not only on data stream but on the instruction stream as well.

① 4 segment instruction pipeline is

1. fetch an instruction (FI)
2. decodes the instruction and calculating Effective address (DA).
3. fetch operand (FO).
4. executing instruction (EX).

(a) 4 segment cpu pipelane



(b) timing of instruction pipeline.

step: / Instruction: (Branch)	1	2	3	4	5	6	7	8	9	10	11	12	13
1	FI	DA	FO	EX									
2		FI	DA	FO	EX								
3			FI	DA	FO	EX							
4				FI	-	-	FI	DA	FO	EX			
5					-	-	-	FI	DA	FO	EX		
6									FI	DA	FO	EX	
7										FI	DA	FO	EX

① 3 is a branch instruction, As soon as this instruction is decoded in segment DA in step 4, transfer from FI to DA of other instruction is halted until branch instruction is executed in step ⑥.

② If branch is taken, new instruction is fetched in step ⑦.

③ If branch not taken, instruction fetched previously in step ④ can be used.

④ In general there are 3 major difficulties cause instruction pipeline to deviate from normal operation.

① Resource conflicts → caused by access to memory by 2 segments at same time., can be solved by using seperate instruction and data memories.

② data dependency → arises when instruction depends on result of previous instruction, but result is not available.

③ branch difficulties → arise from branch and other instructions that change value of PC.