M T W T F S S

Page No.: YOUVA

Date:

# Unit-V

Topics to cover:

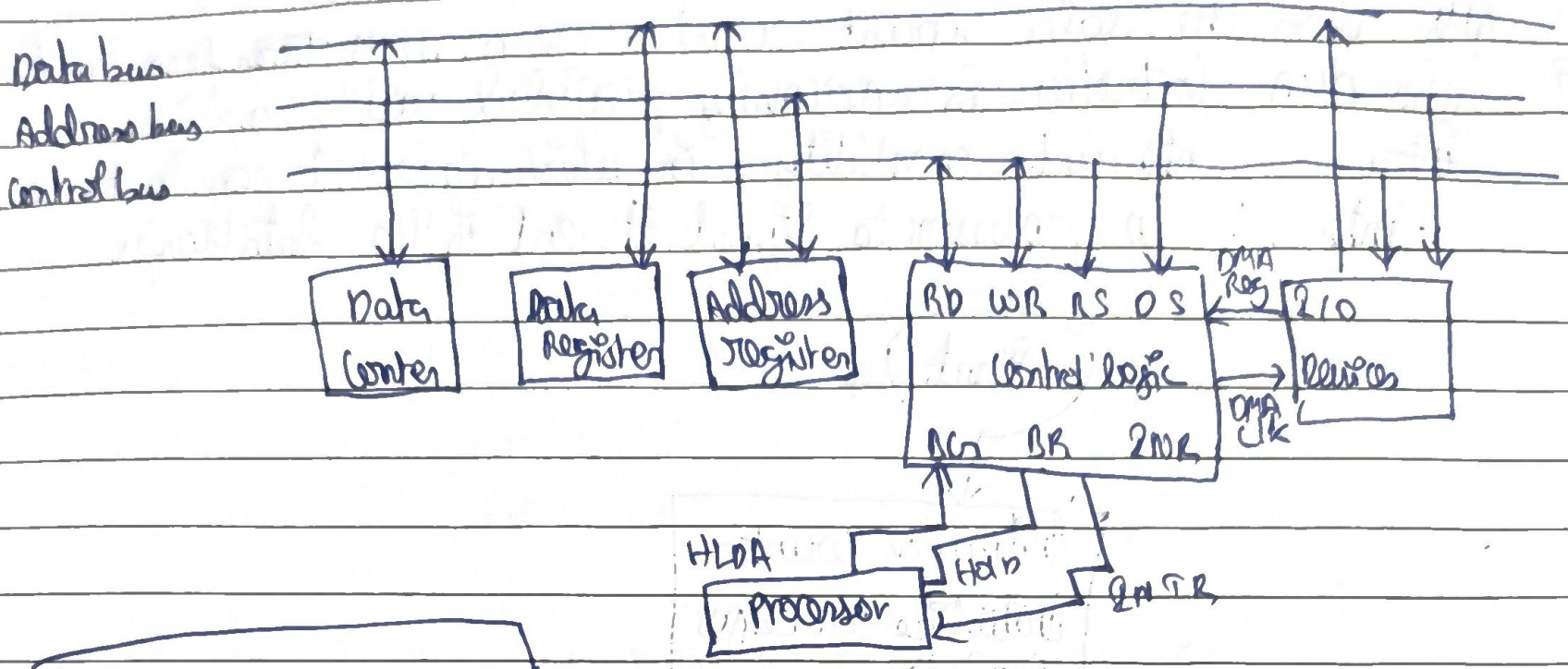
* Memory concept and Hierarchy
* Memory Management
* Cache Memory: Mapping and Replacement Techniques
* Virtual Memory ✓
* DMA ✓
* I/O
* Accessing I/O parallel and Serial interface ✓
* Interrupt I/O
* Inter Connection Standards: USB, SATA

**Direct Memory access** It allows certain hardware subsystem to access the Main System memory independently

DMA:

* DMA uses hardware for accessing the memory, that hardware is called DMA Controller
* It has the work of transferring the data between Input/output devices & Main memory with very less interaction with the processor
* The direct memory access controls is a control unit, which has the work of transferring data.

Data bus
Address bu
Control bu



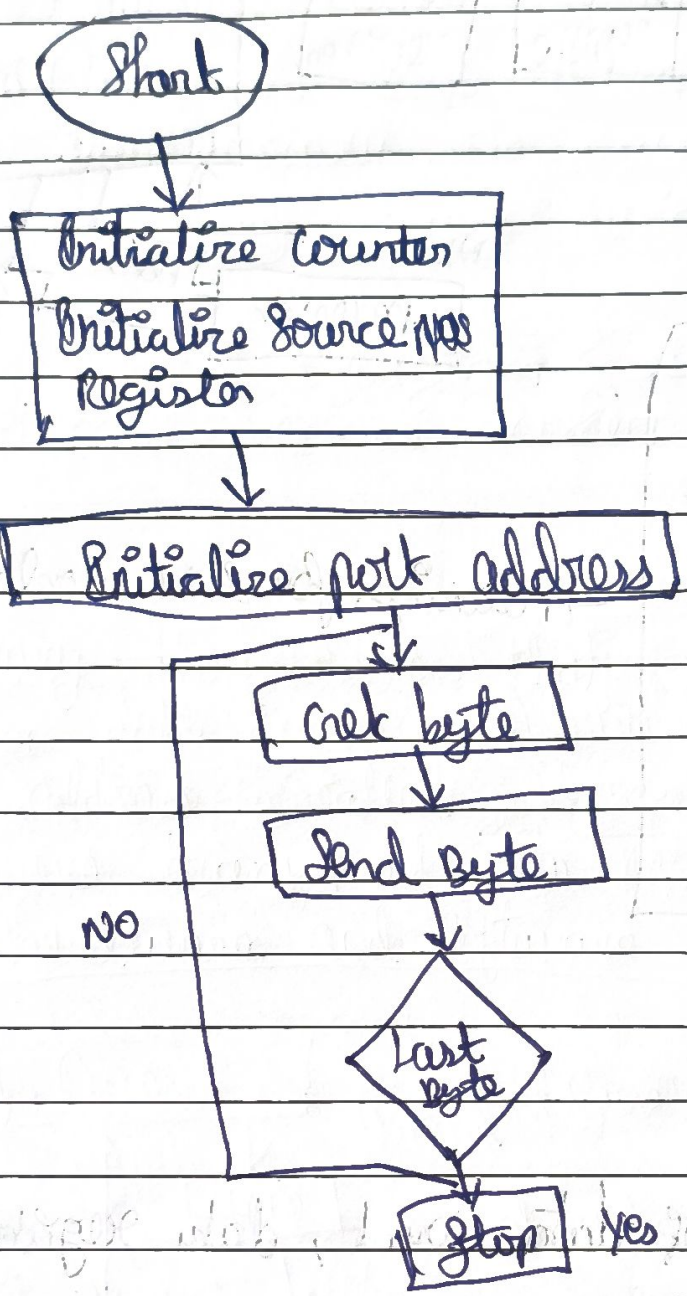

DS → Data Select
RS → Register Select
RD → Read
WR → Write
BR → Bus request
BG → Bus Grant
INTR → Interrupt

→ meanings for short words

Explanation:

① DMA consists of data count, data register, address register and Control logic
② Data ~~transfer~~ Counter Register Stores the number which gives the number data transfers to be done in one DMA cycle
③ It is automatically decremented after each word transfer
④ Data register acts as a buffer where as address register initially holds the starting address of the device
⑤ Actually, it stores the address of the next word to be transferred. It is automatically incremented and decremented after each word transfer.

(6) After each transfer, data counter is tested for zero
(7) when the data count reaches zero, DMA transfers halts
(8) The DMA controller is normally provided with an interrupts capability, in which case it sends an interrupt to processor to signal the end of I/O data transfer.




Types of DMA
=> Single Ended DMA
=> Dual Ended DMA
=> Arbitrated - Ended DMA
=> Interleaved DMA

i) It is operated by reading and writing from single memory address. They are the simplest DMA (Single Ended)
ii) It can read & write from two memory address Dual - Ended DMA is more advanced than Single - ended DMA
iii) It works by reading and writing to several memory address. It is more advanced the Dual - Ended DMA (Arbitrated - Ended)
iv) It is DMA that read from one memory address and write from another memory address

DMA Controller have 3 registers:

i) Address Register: It contains the address to specify the desired location in memory.
ii) Word count Register: It contains the number of words to be transferred
iii) Control Register: It specifies the transfer mode

## Virtual Memory

① The Virtual Memory: technique is used to expand the apperant size of the physical memory
② It use secondary storage such as disks, to extend the apparent size of the physical memory.
③ Virtual memory: Is a concept that allows user to construct programs as large as axilary memory.
④ When a program does not completely fit into main memory it is divided into segments

(5) The Segments which are currently being executed are kept in the Main memory & remaining segments are stored in the Secondary Storage devices such as magnetic disk.

(6) If an executing program needs a segment which is not currently in the Main memory & remaining segments are

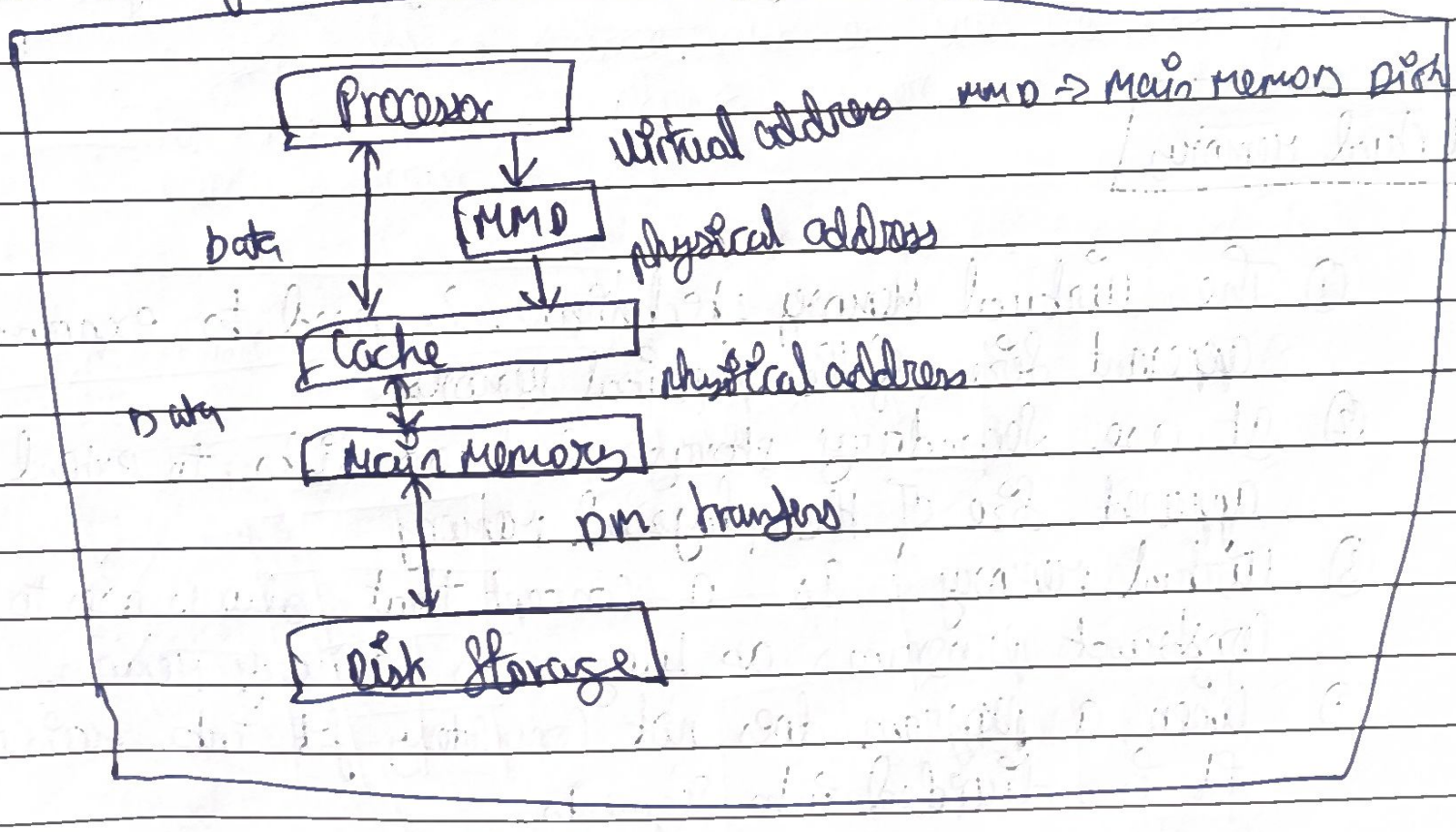
(6) If an executing program needs a segment which is not currently in the main memory, the required segment is copied from the secondary storage device

(7) When a new segment of a program is to be copied into a Main memory it must replace another segment in the memory.

(8) In virtual memory, the address that processor issues to access either instruction or data are called virtual address.

(9) The set of such address is called address space

(10) These address are transferred into physical address by a combination of hardware and software components



M T W T F S S

Page No.:

Date:

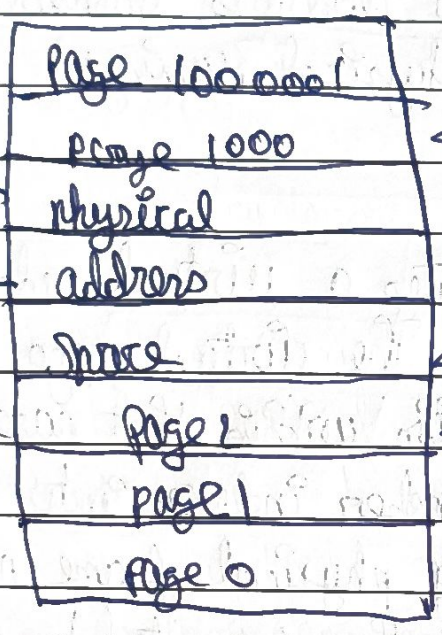
YOUVA

## Paging

*) It is a simple method for translating virtual address into physical address is to assume that all programs and data are composed of fixed length unit called pages.

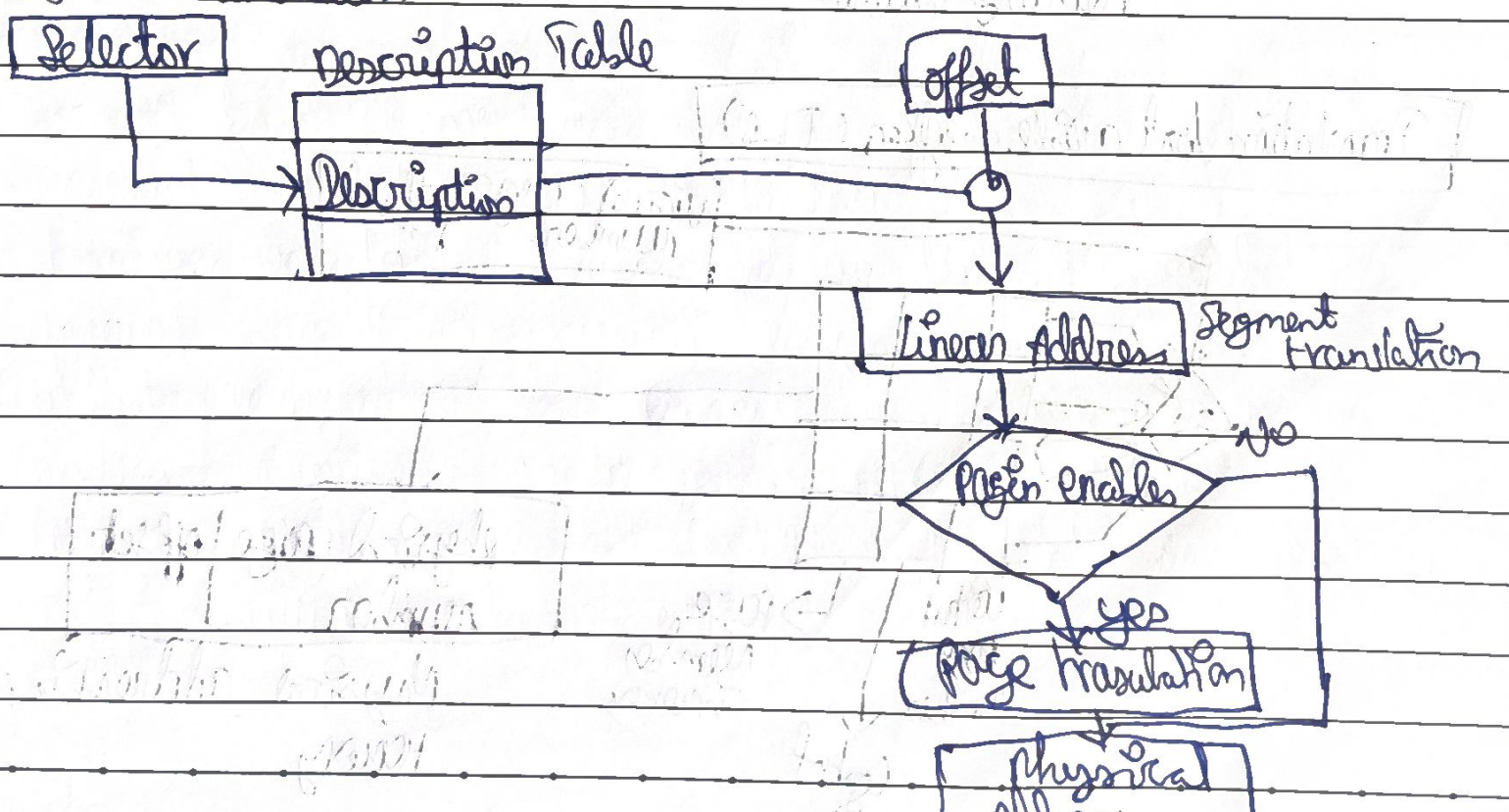
*) Pages constitute the basic unit of information that is moved between the main memory and the disk whenever the page translation mechanism determines that a swapping is required

Ex :




## Virtual to physical Addes Translation

① Segment translation



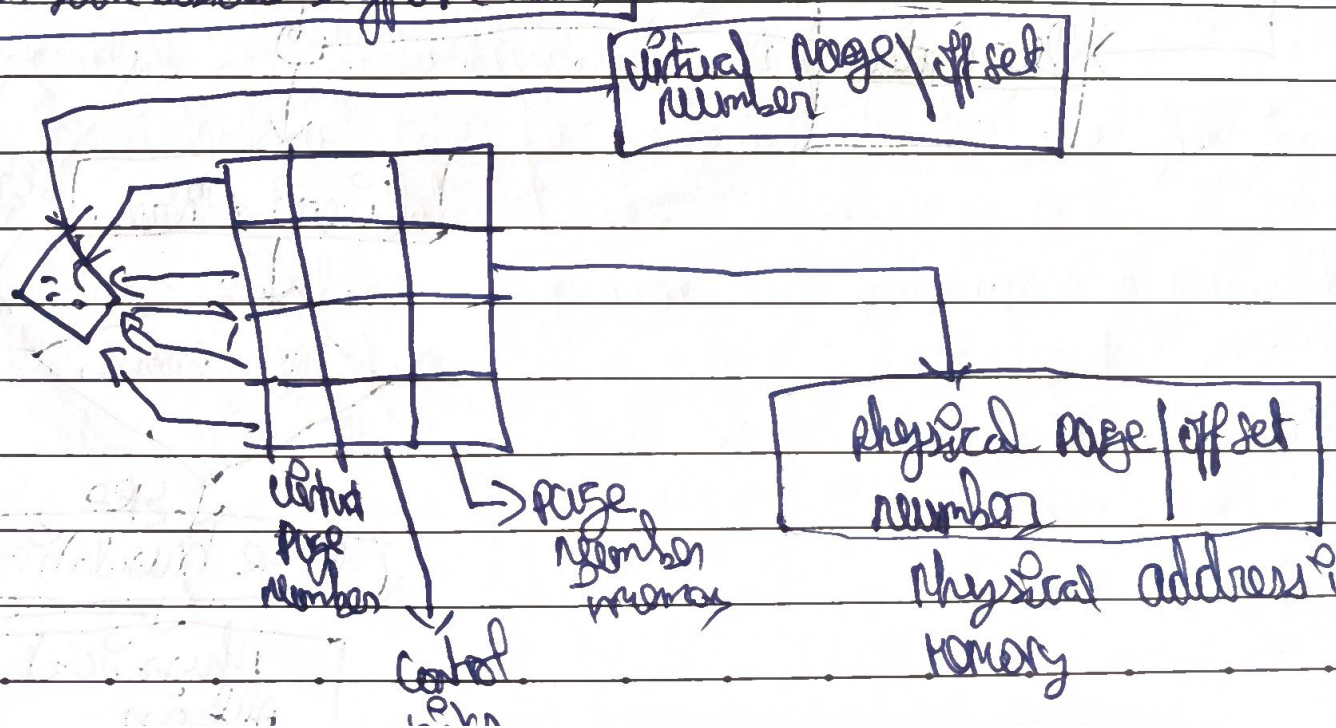
Key components :

* MMU : Memory Management Unit , A hardware device responsible for translating virtual address to physical address at runtime
* Page tables : Data structures maintained by the operating system that map virtual page numbers to physical frame numbers.
* Translation lookaside Buffer (TLB) : A cache used by the MMU to speed up address translation by storing recent virtual to physical mappings

Process overview :

1) The program generates a virtual address, which consists of two parts : The virtual page number and the page offset (location within that page)
2) The virtual page number indexes into the page table to find the corresponding physical frame number in RAM
3) The physical address is then computed by combining the physical frame number with the page offset
4) The MMU uses this physical address to access the actual memory location

Translation lookaside buffer (TLB)



i) To support demand paging and virtual memory processor has to access page table which is kept in main memory.
ii) To reduce the access time and degradation of performance, a small portion of the page table is accomodated in the memory management unit
iii) This portion is called translation lookaside buffer

## Serial inferance and parallel inferance

difference:

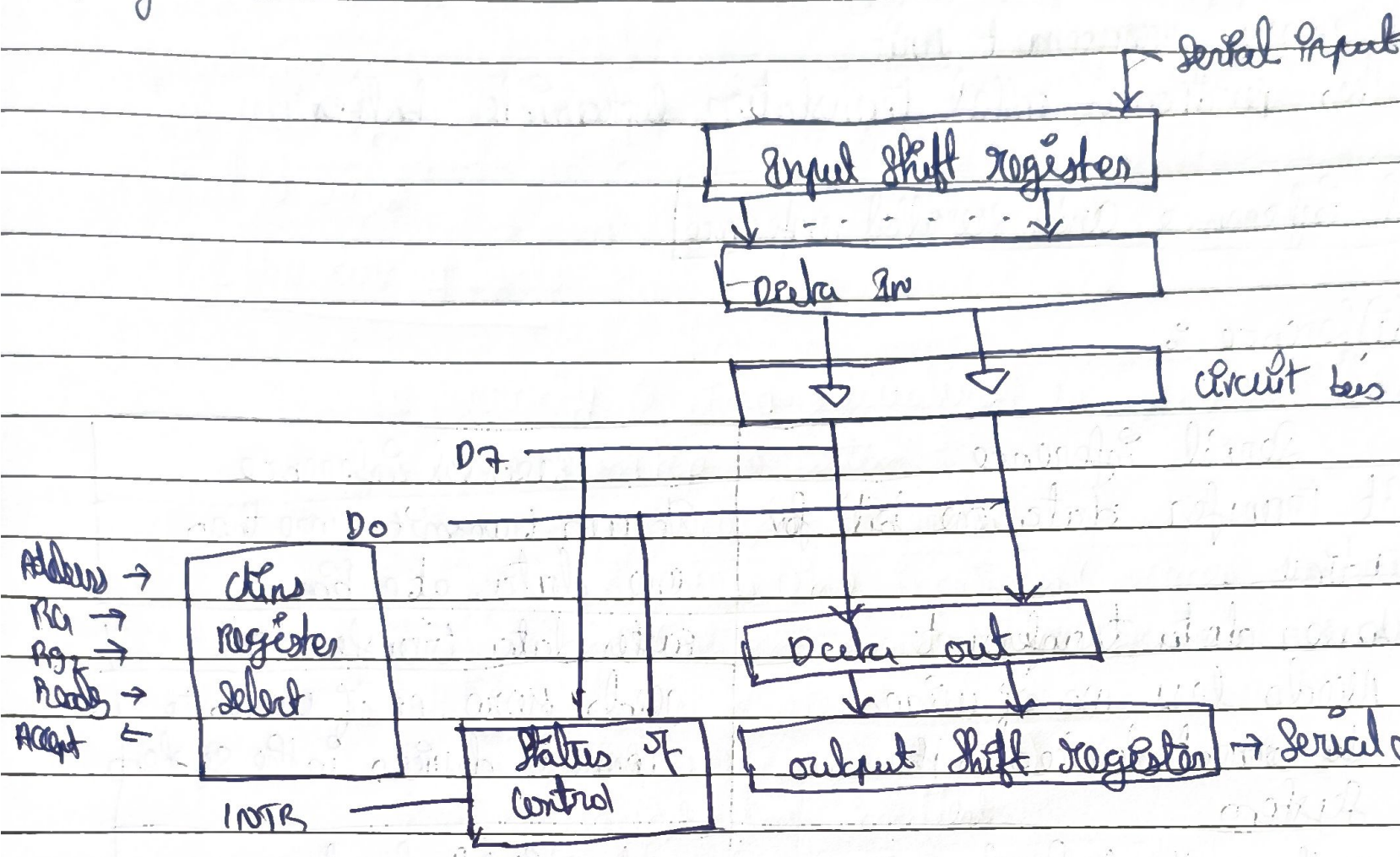
| Serial inference | Parallel inference |
|---|---|
| 1) It transfer data one bit for a time | It can transmit more than one data at a time |
| 2) Lower data transfer rate | Faster data transfer rate |
| 3) Needs less no of wires to connect devices in the system | Needs more no of wires to connect devices in the system |
| 4) Well suited for long distances because fewer wires are used | Not suitable for long distance interference |

## Serial Interface:

*) A serial interface is used to transmit/receive data serially (1 bit at a time)
*) A key feature of an interface circuit for a ~~serial~~ serial port is that it is capable for communicating in a bit serial fashion on processor side
*) A shift register is used to transform information between the parallel and serial formats.
*) The input shift register accepts serial data bit by bit and converts it into parallel data

*) The converted parallel data is located in the data register and it is then read by the processor using data bus.

Diagram:




Parallel Interface

*) The input and the output interfaces can be combined into a single interface and the direction of data flow can be controlled by data direction register
*) Single interface can be programmed to use for input the data or output the data
*) Such a interface is known as programmable parallel interface
*) Data and interface lines are bidirectional
*) Their direction is controlled by data direction register (DAR)

M T W T F S S

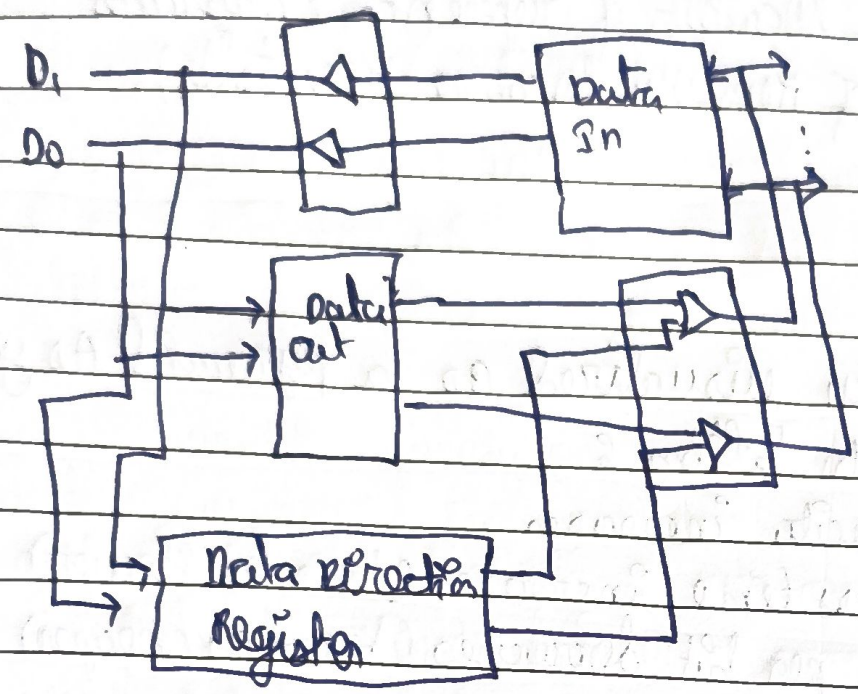
Page No.:

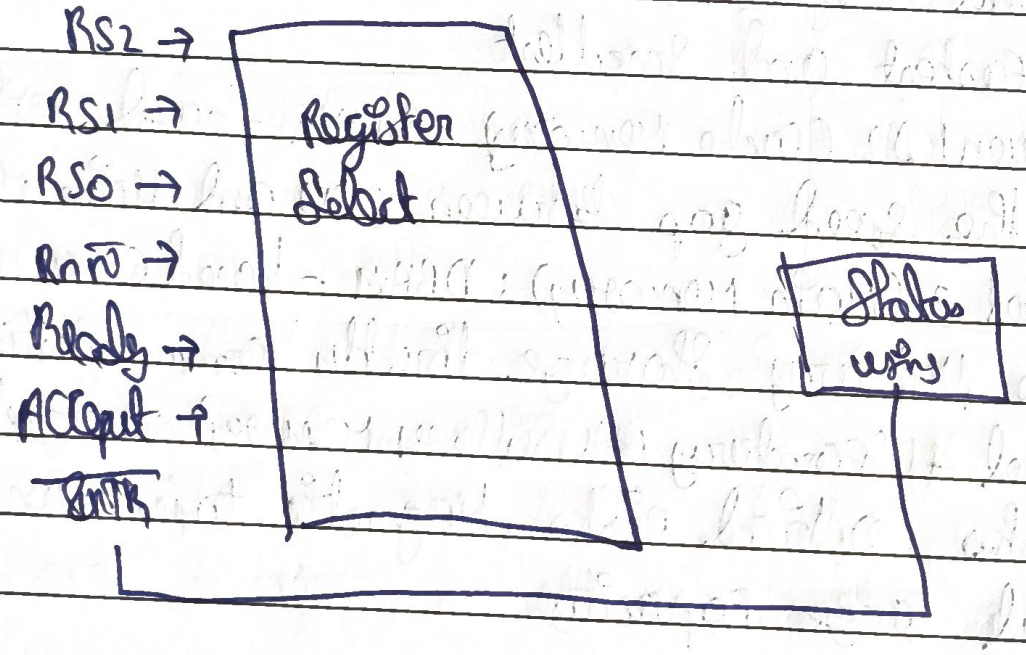
Date:

YOUVA

*) MO & M are connected to status control
*) Ready and Accept signals are provided as handshaking signals

Diagram:



M T W T F S S

Page No.:
Date:
YOUVA

## Memory concepts and Hierarchy

The Memory System is organized in a hierarchy based on Access time, capacity and cost. The goal is to optimize performance by utilizing the principle of locality of reference (programs tend to access the same set of memory location repeatedly).

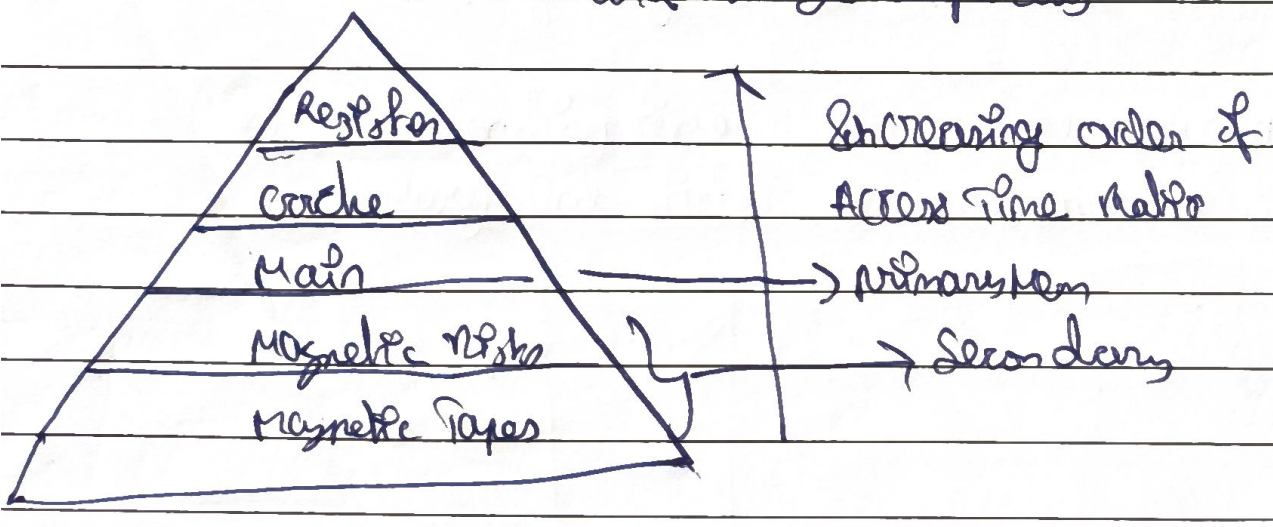
### The Hierarchy levels

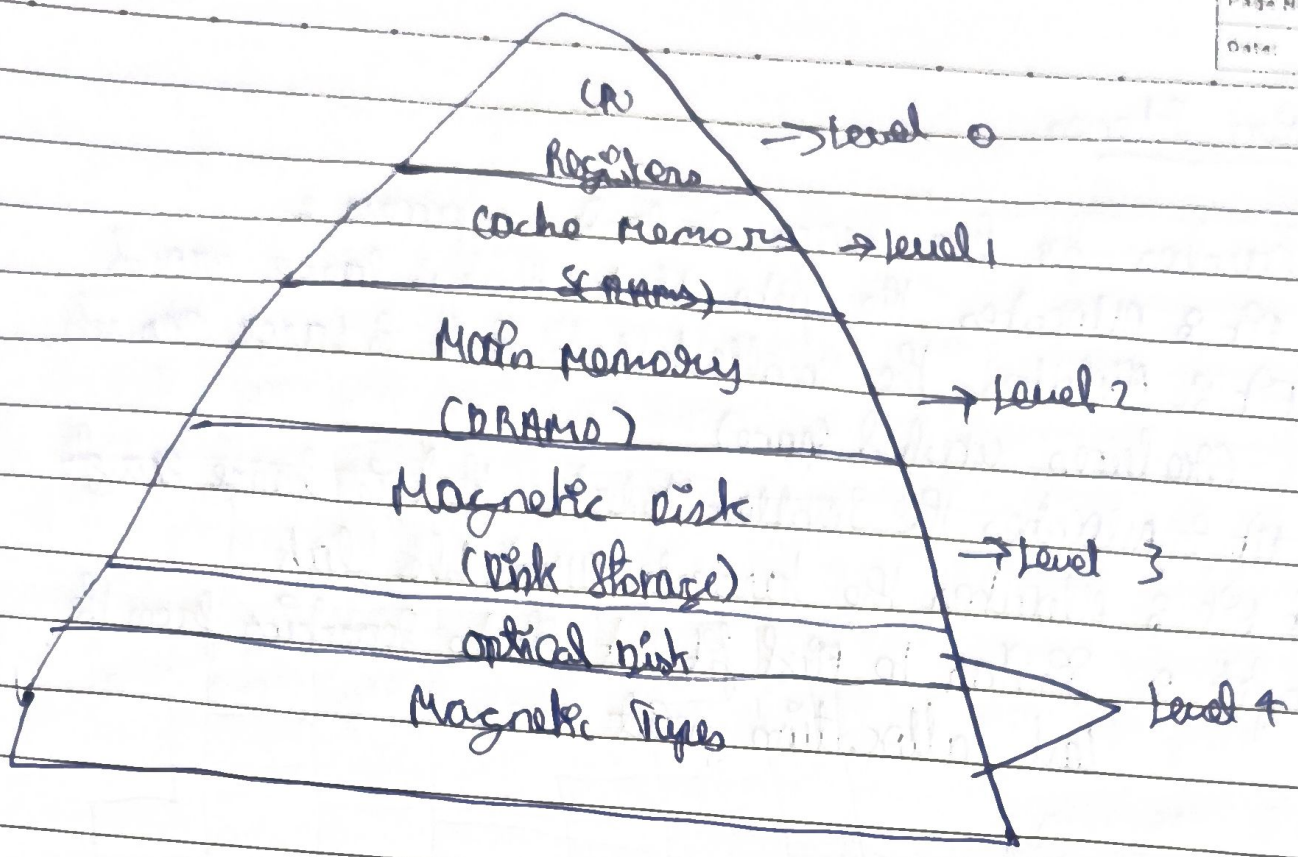
The hierarchy is often visualized as a pyramid As you you move from top to bottom:

*) Capacity increases
*) Access time increases (becomes slower)
*) Cost per bit decreases (becomes cheaper)

Level Breakdown:

o) Level 0 (Registers): Located inside the CPU. Fastest and Smallest
o) Level 1/2 (Cache Memory): SRAM - based Bridges the speed gap between CPU and Main Memory
o) Level 3 (Main Memory): DRAM - based (RAM/ROM). The Primary Storage directly communicating
o) Level 4 (Secondary / Auxillary Memory): Magnetic disks, Optical Disks, Magnetic tapes non-volatile and large capacity






## Memory Management

Memory management is the process of controlling and coordinating computer memory, assigning blocks to running programs to optimize system performance

Key Concepts:

- Swapping: A mechanism where a process is temporarily moved out of main memory to secondary storage (disk) to make room for other processes, and brought back later
- Fragmentation:
  - External fragmentation: Free memory is separated into small blocks of scattered across the system, making them unusable for large processes
  - Internal fragmentation: Memory is allocated in fixed blocks, but the process is smaller than the block, leaving unused space inside

## Partition Allocation Schemes

How the OS Searches for free space to load a process :

- First fit : Allocates the first block that is large enough
- Best Fit : Allocates the smallest block that is large enough (reduces wasted space)
- ~~Worst fit : Allocates the smallest block that is large enough~~
- Worst fit : Allocates the largest available block
- Next fit : Similar to first fit but starts searching from the last allocation point

## Paging Vs Segmentation

- Paging : Logical memory is divided into fixed-size blocks called pages. Physical memory is divided into frames. Eliminates external fragmentation
- Segmentation : memory is divided into variable-sized segment based on logical units

## Static loading :

*) Static loading is used when you want to load your program statically. Then at the time of compilation, the entire program will be linked and compiled without need of any external module or program dependency

*) At loading time, the entire program is loaded into memory and starts its execution

Dynamic loading

*) In a dynamically loaded program, reference will be provided and the loading will be done at the time of execution
*) routines of the library are loaded into memory only when they are required in the program

## Cache Memory

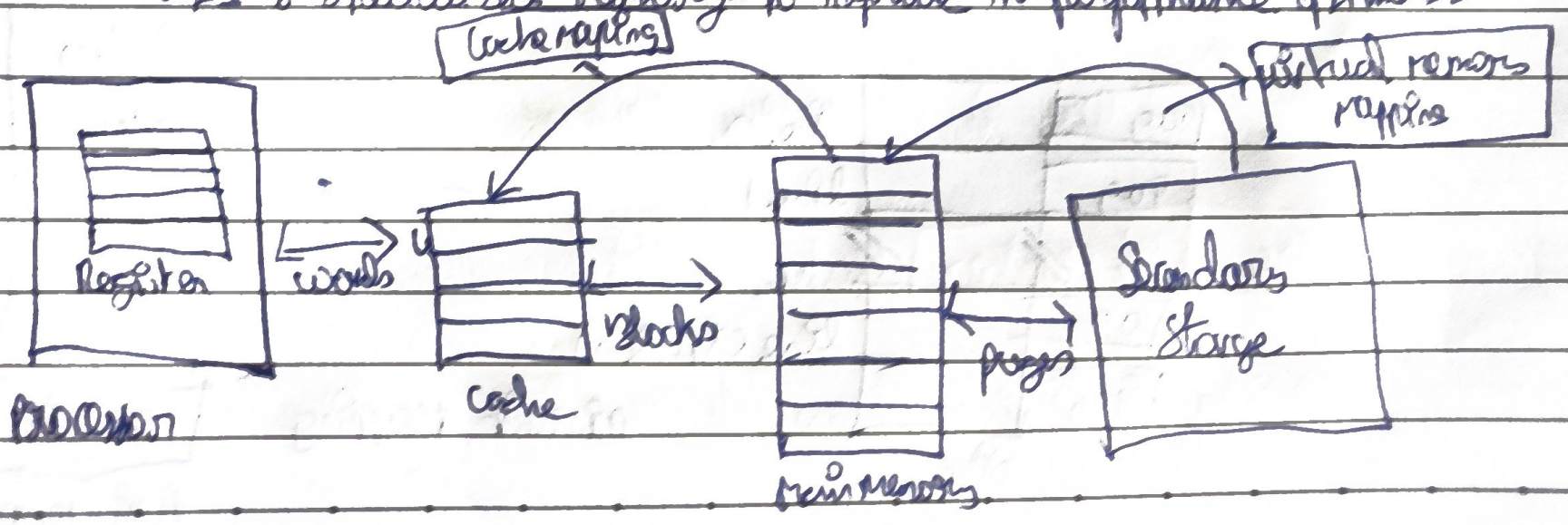
### overview

Cache memory is a small, high-speed memory unit located between the CPU and Main Memory (RAM). Its primary purpose is to bridge the significant speed mismatch between the fast processor and the slower main memory

- Function: It holds frequently requested data and instructions so the CPU can access them immediately.
- Locality of Reference: cache operates on the principle that programs tend to access the same set of memory locations locations repeatedly over a short period.

Levels:

- L1 (primary) Fastest, Smallest, usually embedded in the processor
- L2 (Secondary) Larger than L1, slightly slower
- L3: Specialized memory to improve the performance of L1 and L2



# Mapping Techniques

## Cache Mapping

Cache mapping defines the specific rules for placing a block of data from the main memory into the cache memory. Since the cache is smaller than main memory, we need to method to determine which main memory block occupies which cache line.
The main memory is divided into lines of the same size.

### Direct Mapping:

In this technique, a specific block from main memory maps to only one specific line in the cache.

*) Formula:

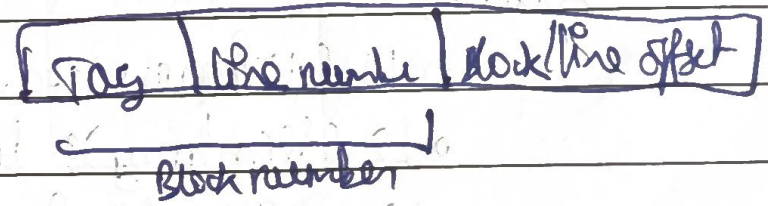
Cache line = (Main memory block Address) % (number of lines in cache)

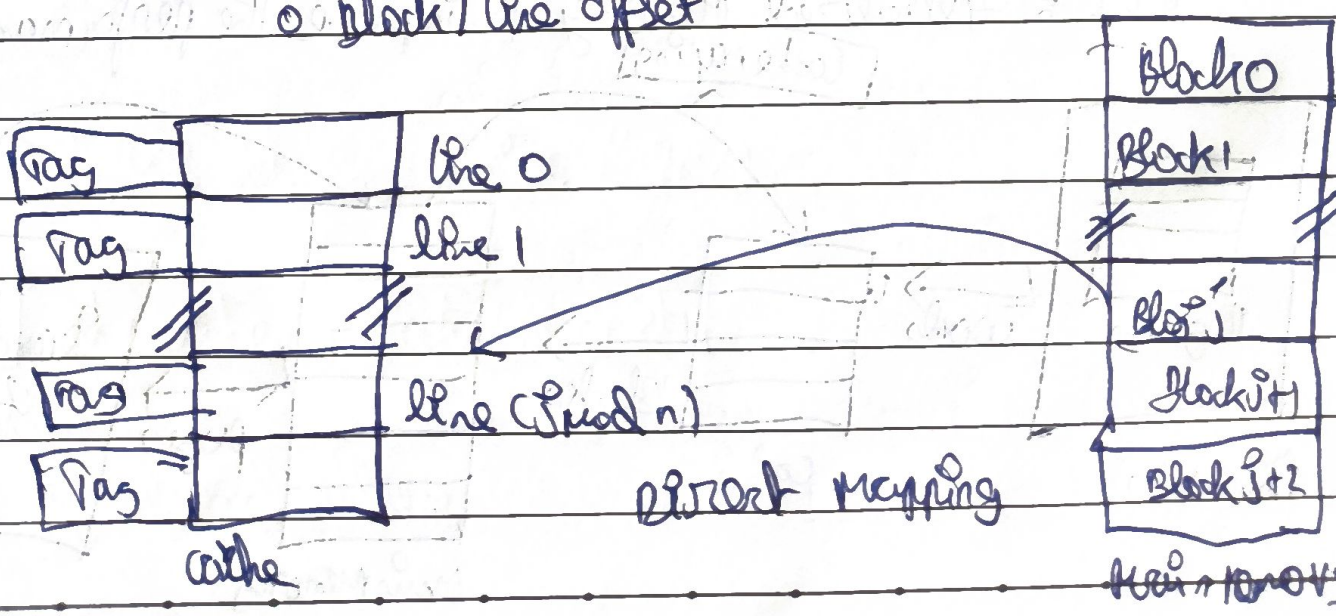
*) Mechanism: If the cache has n lines, block j of main memory maps to line (j mod n).

*) Replacement: No replacement algorithm needed. If a new block maps to a line already occupied, the old block is simply overwritten.

*) Address Structure: The physical address is split into three parts

- Tag
- Line number
- Block / line offset



M T W T F S S
Page No.:
Date:
YOUVA

## c) Fully Associative Mapping

This is the most flexible technique. A block from main memory can be placed in any freely available line in the cache.

- Mechanism: The cache control logic checks all lines to find the data.
- Replacement: A replacement algorithm is mandatory because if the cache is full, the system must decide which existing block to evict to make room for the new one.
- Address Structure: The physical address is split into two parts (no line number is needed):
  - Block number (Tag).
  - Block / Line offset.

## d) k-way Set Associative Mapping

This is a hybrid approach that combines the simplicity of Direct Mapping with the flexibility of fully Associative mapping.

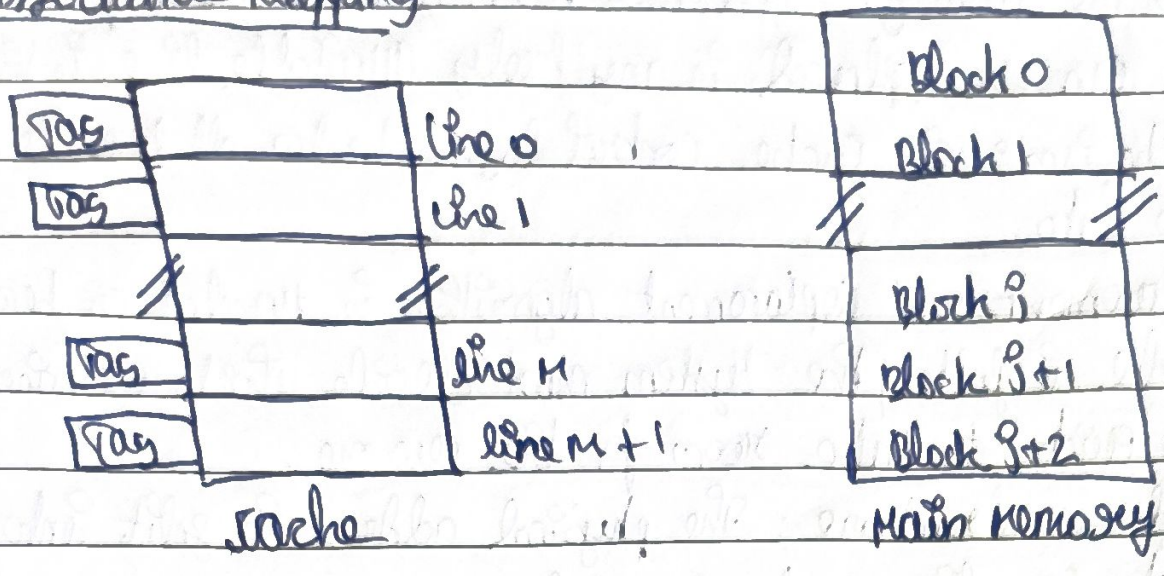
- Mechanism: Cache lines are grouped into sets, where each set contains k lines. A block from Main Memory maps to a specific set, but within that set, it can occupy any of the k lines.
- Formula:

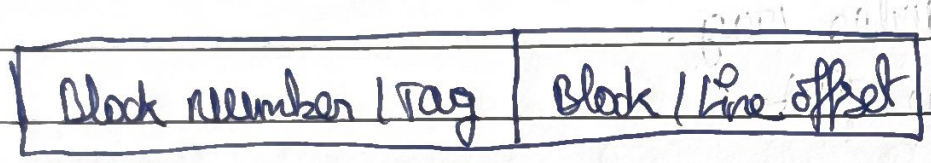
  Cache Set = (Block Address) (Mod Total Sets in Cache)

- Replacement: A Replacement algorithm is required only for the specific set if all lines in that set are full.
- Address Structure
  - Tag
  - Set number
  - Block/line offset
- Special Cases:
  - If k = 1, (it functions exactly like Direct Mapping)
  - If k = total lines, it functions exactly like fully Associative Mapping

## Block diagram

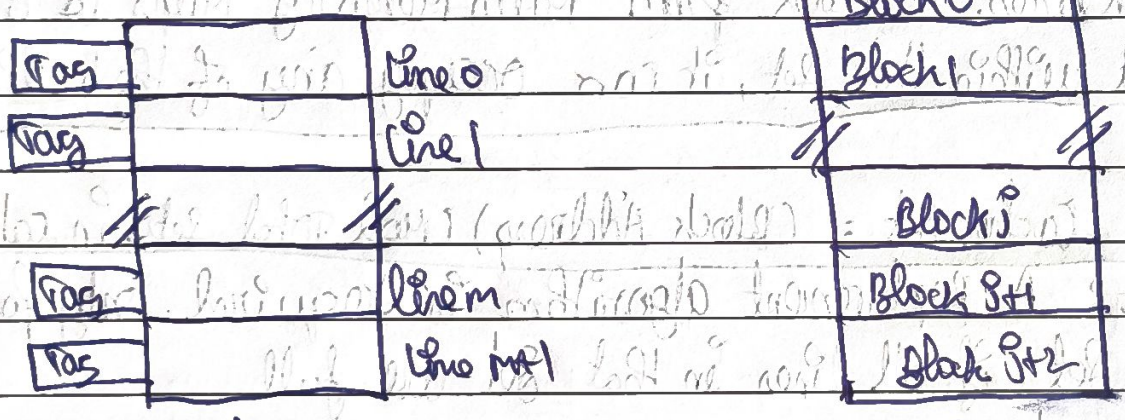
### Fully Associative Mapping



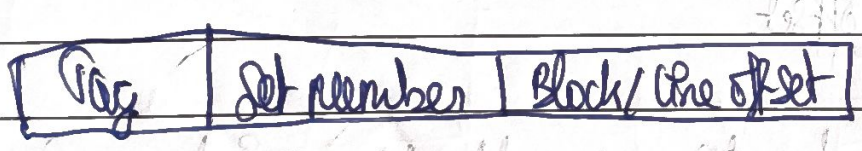




Division of Physical Address in fully associative mapping

### K-way set Associative Mapping




2-way set Associative Mapping




Division of physical address in k-way set associative mapping

M T W T F S S

Page No.: YOUVA

Date:

## Replacement Algorithm :

The performance of cache memory is measured by the hit ratio (h)
-The percentage of memory accesses satisfied by the cache

Average Memory Access Time ($T_{avg}$):

$$T_{avg} = h \times T_c + (1-h) \times M$$

- h = Hit rate
- 1-h = Miss rate
- $T_c$ = Time to access
- M = Miss penalty

## USB and SATA

USB Universal Serial Bus

* Plug and play → Starts automatically
* Hot pluggable → Can be attached or remove anytime
* Polling → device do not interrupt the host

Ex:

Type A, B, C

SATA (Serial Advanced Technology Attachment)

* Used to connect Standard Storage device, HDD, SSD, RAM
* So can not be removed easily, should not be removed while running