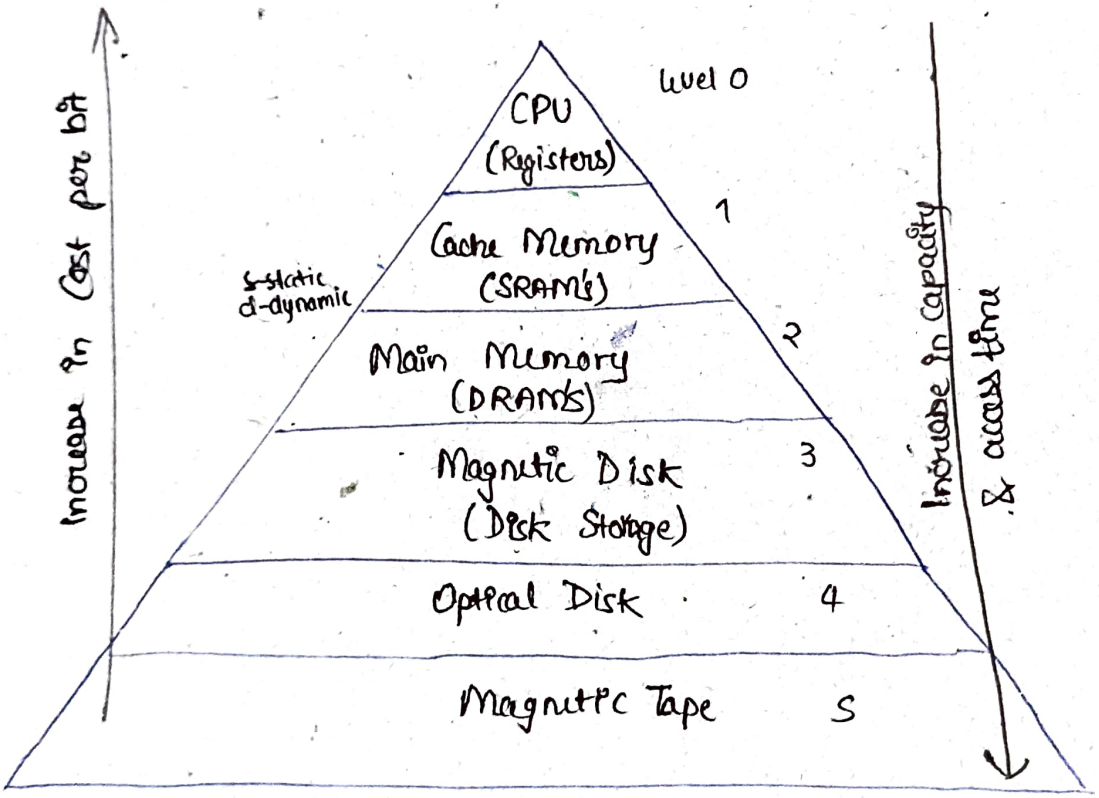
Memory Hierarchy:

* Organization of memory for saving access time.
* helps in optimizing available memory.
* arrangement of different kinds of storage devices in a computer, based on size, cost, & access speed.
* helps to speed up operations.



Characteristics:

(i) Cost per bit – total cost of memory by total no. of accessed bits.

(ii) Capacity – Volume of info. that memory device can store.

(iii) Access time – time interval when a read/write request is made to the time when data actually becomes available.

Memory Interleaving:

* It is a kind of abstraction technique designed to compensate for relatively low speed of DRAMs or core memory.

* It is a technique that spreads data or memory addresses across multiple memory banks to improve performance.

* This allows for simultaneous accessing of different banks, reducing the waiting time for memory banks.

Types:

1. High order Interleaving - MSB of memory address decides memory banks (module).
LSB provides address of data in the Module.

0000	10
0001	20
0010	30
0011	40
0100	50
0101	60
0110	70
0111	80
1000	90
1001	100
1010	110
1011	120
1100	130
1101	140
1110	150
1111	160

Module 00

00	10
01	20
10	30
11	40

Module 01

00	50
01	60
10	70
11	80

Module 10

00	90
01	100
10	110
11	120

Module 11

00	130
01	140
10	150
11	160

2. Low level Interleaving -

The LSB decides the module address & MSB determines the address within module.

Module 00

10
50
90
130

module 01

20
60
100
140

Module 10

30
70
110
150

Module 11

40
80
120
160

Advantages :

* It allows simultaneous access to different modules.
* It makes system fast & more responsive.
* CPU access time decreases & hence performance enhances.

Associative memory :

* A memory unit whose stored data can be identified for access by content of data itself or special metadata or additional tags (that describes it contents), rather than by address or memory location.

* also called CAM - Content addressable memory

* When a write operation is performed on associative memory, no address is given to word. The CAM is capable of finding an empty unused location to store the word.

* When a data is to be read from associative memory, the content of data or part of data is specified. The data which match with specified content are located by memory & are marked for reading.

* It is a special memory type optimized for performing searches through data.

Advantages :-

* Suitable for parallel searches
* used to speed up database.

Disadvantages :-

* more expensive than RAM.
*

Applications -

Used in DBMS's, AI application, Image processing applications, in networking.

Cache Memory –

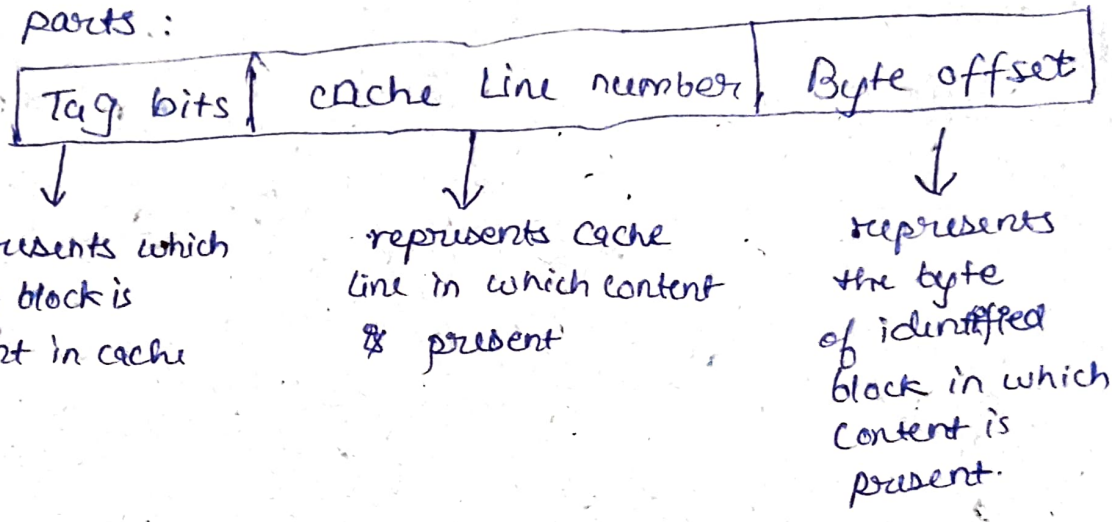
* Small, high-speed storage area in a computer.
* reduces average time to access data from main memory.
* It holds frequently requested data & instruction so that they are immediately availble to CPU when needed.

Cache mapping techniques –

* a technique that is used to bring the main content to cache or to identify cache block in which the required content is present.
* These are of following types:
  - → Direct
  - → Associative
  - → Set-Associative
* Cache mapping is the procedure to decide in which cache line the main memory block will be mapped.
* Cache mapping is the pattern to copy required main memory content to specific location in cache memory.

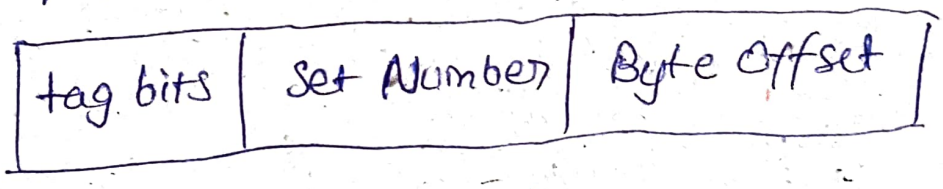
(i) Direct :

* physical address is divided into three parts :



(ii) Set Associative :

* cache blocks are divided in sets.
* physical address is divided into three parts,



(iii) Associative :

* physical address is divided into two parts :

I/O organization:

I/O Subsystem:

* a system responsible for communication b/w the computer's central system & outside environment.
* This system handles all the input-output operations of the computer system.

Peripheral devices –

* Input-output devices connected to the computer.
* The devices are designed to read or write info into or out of the computer memory.
* Eg: Keyboards, display units & printers

1. Input peripherals –

* Allow user input from out world to be read to computer memory.

eg – keyboards, mouse.

2. Output peripherals –

* Allows information output to be presented to user / outside enviroment from the computer

eg – Printer, Monitor

3. Input/Output peripherals –

* Allows to take input as well as print / present output.

* Eg: Touch Screen.

Interfaces –

* It is a shared boundary b/w two separate components of a computer system which is used to attach components to computer system for communication purpose.

* There are two types of interfaces –

1. CPU Interface 2. I/O Interface.

2. I/O Interface –

* The interface using which information is transferred b/w internal storage & external I/O devices is known as I/O interface.

* The special hardware components b/w CPU & peripherals to control or manage the input-output transfers are called I/O interface units.

Advantges:

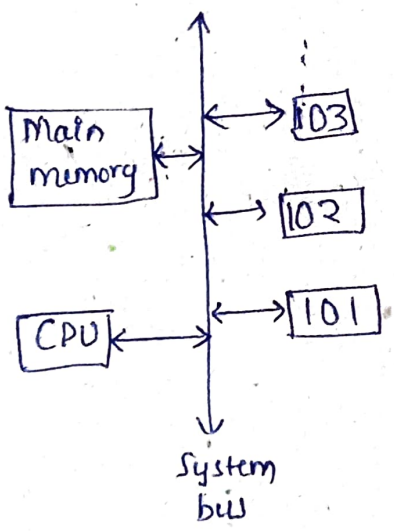
* provides standard way of communicating with external devices

* Increase efficiency & speed of data transfer between computer & external devices.

Mode of Transfer:

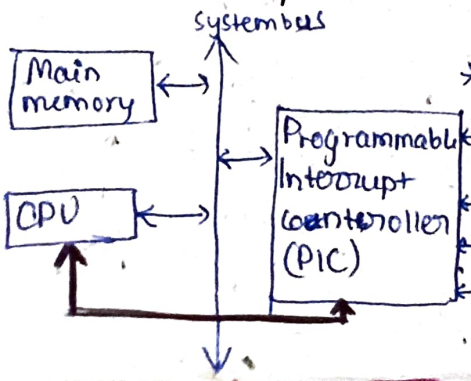
* Data transfer b/w central unit & I/O devices can be done in different modes.
* Data transfer to & from peripherals may be done in any of the 3 ways:
  1. Programmed I/O
  2. Interrupt - initiated I/O
  3. Direct memory access (DMA)

1. Programmed I/O:



* It is due to result of I/O instructions that are written in computer program.
* Each data item transfer is initiated by an instruction in the program.
* tranferring data under programmed I/O requires constant monitoring of peripherals by CPU.
* CPU takes more wait time.

2. Interrupt - initiated I/O:



* In Programmed I/O, the CPU stays in program loop until the I/O unit indicates that it is ready for data transfer. This is time -

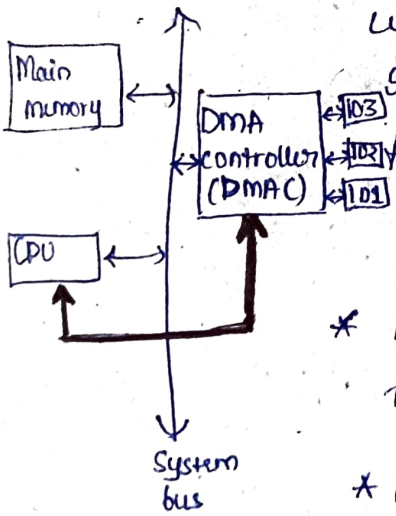
consuming & it keeps the processor busy unnecessarily.

* This problem is overcome by Interrupt-initiated I/O.

* In this, the interface issues an interrupt signal (using interrupt facility & special commands) when data is available from any device.

* Upon detection of an external interrupt signal, the CPU stops momentarily the task that it was already performing & services newly requested I/O transfer & then returns back to previous task.

3. Direct Memory access (DMA):



* When peripherals directly communicate with each other using memory buses, removing intervention of CPU, then this type of data transfer technique is known as DMA.

* In this transfer, the CPU is idle & has no control over the memory buses.

* It enhances parallelism in CPU & I/O operations to avoid wastage of valuable CPU time.

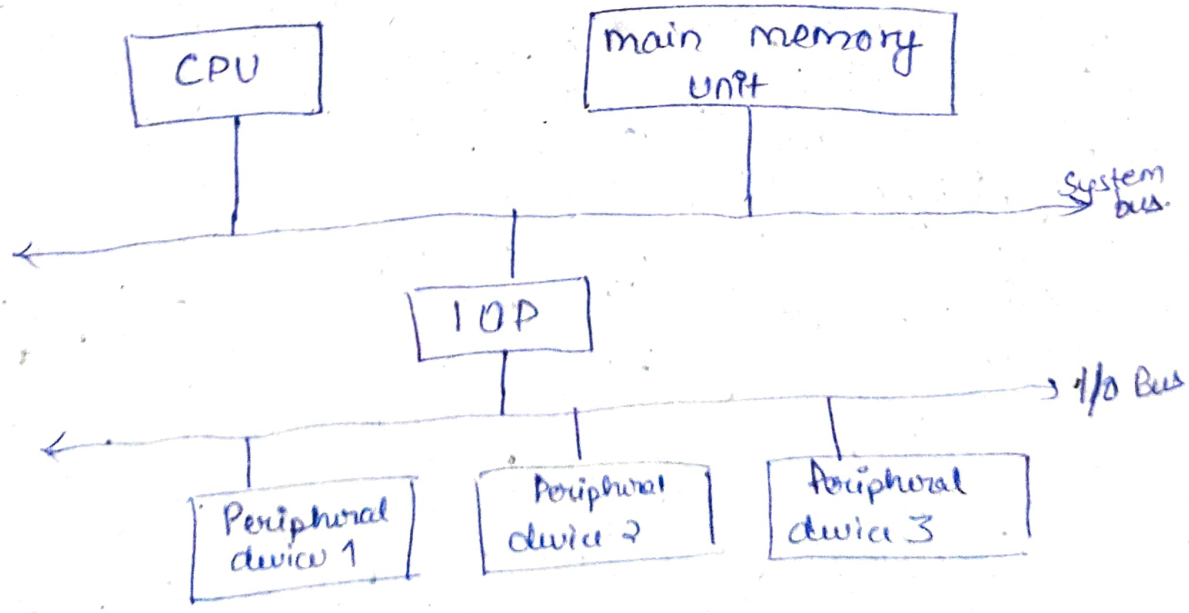
I/O Processor –

* special purpose processors that handles the I/O operations.

* Their main purpose is to relieve the CPU from involvement in I/O operations.

* specialized processors that loads & stores data in main memory along with execution of I/O operations.
* The IOP can fetch & execute its own instruction, making it independent of CPU.

Block diagram :



Features :

1. Specialized hardware for handling I/O operations.
2. DMA capability that allows data to be directly transfered from peripheral devices to memory & vica-versa.
3. Can handle Interrupts & manage them.
4. Can buffer data b/w CPU & peripheral devices.
5. Can process commands independent of CPU.
6. Can perform I/O operations parallel with CPU.

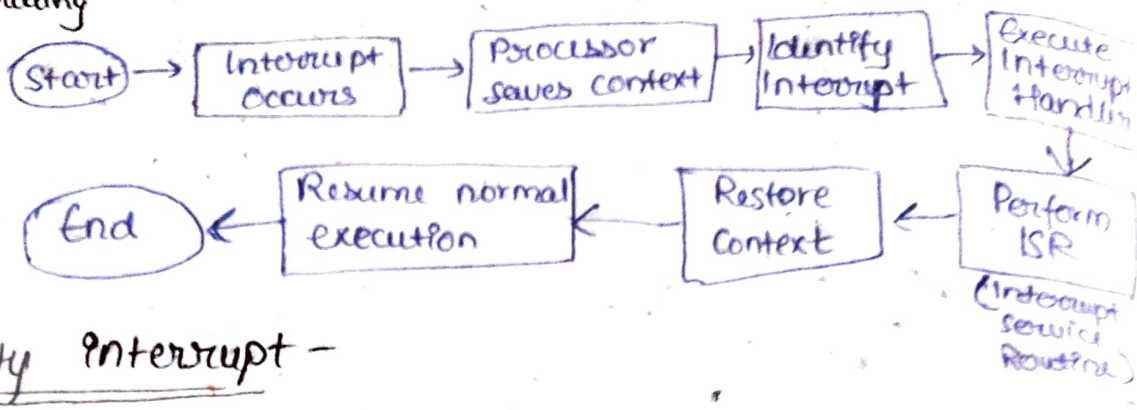
Interrupts :

* a signal from a hardware or software when a process needs immediate attention.

⇒ * Data transfer b/w peripheral & CPU is initiated by CPU.
* But CPU can't start transfer until peripheral is ready to communicate.
* When a device is ready to communicate it generates the interrupt signal.

* The main job of interrupt system is to identify the source of interrupt.
* In case of simultaneous interrupt requests, the interrupt system is to decide which device to be serviced first.

Interrupt handling mechanism :



Priority Interrupt -

* In case of multiple interrupt signals generated simultaneously, the interrupt system has to decide which device is to serviced first. We have to set priority among devices for a systematic interrupt servicing.

* The concept of defining priority among devices so as to to which one is to be srviced first in case of simultaneous request is called priority interrupt system.

* Generally, devices with higher speed such as magnetic disks are given high priority & slow devices like keyboards are given low priority.

*

Priority interrupt can be implemented by :
1. Software method (polling)
2. Hardware method (daisy chaining)

(I). * all interrupts are serviced by branching to same ~~prog~~ servica program.

* This program checks each device if it is the one generating interrupt.

* The device with highest priority is check first & then devices are checked in ascending order of priority.

(II). * Involves connecting all devices that can request an interrupt in a serial manner.

* The device with highest priority is placed first & followed by second highest & so on.

Types of Interrupts –

(I) Hardware interrupt –

* When interrupt signal is generated by an external device or hardware device.

(a) Maskable –

* can interrupt that can be delayed when a much higher priority interrupt has occured simultaneously.

(b) Non-Maskable –

* an interrupt that cannot be delayed & should be processed immediately.

(11) Software Interrupt -

* the interrupts caused by Internal system.

(a) Normal

* that are caused by software instruction

(b) Exception

* Unplanned interrupts produced during execution of some programs.

Pipeline -

* a technique used in modern processors to improve performance & speed by executing multiple instruction simultaneously.

* It breaks down instruction into several stages where each stage completes a part of instruction.

* Stages are allowed to be implemented or functioned in overlap, allowing processor to complete different stages simultaneously.

* So, stages are accomplished in parallel

design of pipeline-

* each pipeline has input & output ends.
* there are multiple stages between two ends & output of one stage is connected to input of next except for last stage which produces final output.

Non pipelined execution.

Stage/Cycle	1	2	3	4	5	6	7	8
S1	I_1				I_2			
S2		I_1				I_2		
S3			I_1				I_2	
S4				I_1				I_2

total time = 8 cycles

Pipelined Execution.

Stage/Cycle	1	2	3	4	5
S1	I_1	I_2			
S2		I_1	I_2		
S3			I_1	I_2	
S4				I_1	I_2

total time = 5 cycles

Performace of pipelined processor —

Let clock cycle time be T_p
total segments be k
total tasks be n

$ET_{pipeline} = k + n - 1$ cycles
$= (k + n - 1) * T_p$

$ET_{non\ pipeline} = k * n$ cycles
$= k * n * T_p$

Speedup (S) = $\frac{\text{performance of nonpipelined processor}}{\text{performance of pipelined processor}}$

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

$$= \frac{n * k}{n+k-1}$$

S_{max} is when $n >>> k$,

So, $S_{max} = \frac{n * k}{n}$

$$S_{max} = k$$

Efficiency = $\frac{\text{Given speed}}{\text{max speed}} = \frac{S}{S_{max}}$

$$= \frac{S}{k}$$

Throughput = $\frac{\text{no. of instruction}}{\text{total time to complete instruction}}$

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

The CPI (cycles per instruction) of ideal pipelined processor = 1.

Throughput –

* number of instructions completed per unit time.
* represents overall processing time for pipeline.
*

Latency –

* time taken by single instruction to complete its execution.
* latency = $\frac{1}{\text{throughput}}$ = $\frac{\text{time of execution}}{\text{no. of instruction}}$
* Lower latency represents better performance

Advantages –

* Increased throughput
* Improved CPU utilization.
* Better performance for repeated tasks.

Disadvantages –

* Increased complexity
* hardware overheat
* Increased cost