Computer Fundamentals

Dr.Akshita.S.Chanchlani

Ph.D (Computer Science and Engineering)

akshita.chanchlani@sunbeaminfo.com

Trainer Introduction

- **Name**: Dr.Akshita S.Chanchlani

- **Designation** : Associate Head- Technical

- **Education** :

  - PhD : Computer Science and Engineering

  - Masters of Engineering (ME) in Information Technology

  - B.Tech in Computer Engineering, From VJTI Mumbai

- **Training Experience**

  - PreCAT : C, C++, Operating System, Computer Fundamentals and Computer Networks, AI

  - PG Course : Core Java, Python, Data Structure using Java

  - Modular Batches : Core Java, Python, Machine Learning

  - Internship with Live Project : Machine Learning

- **Professional Experience**

  - **13+ years**

- **Email** : akshita.chanchlani@sunbeaminfo.com

A Computer System
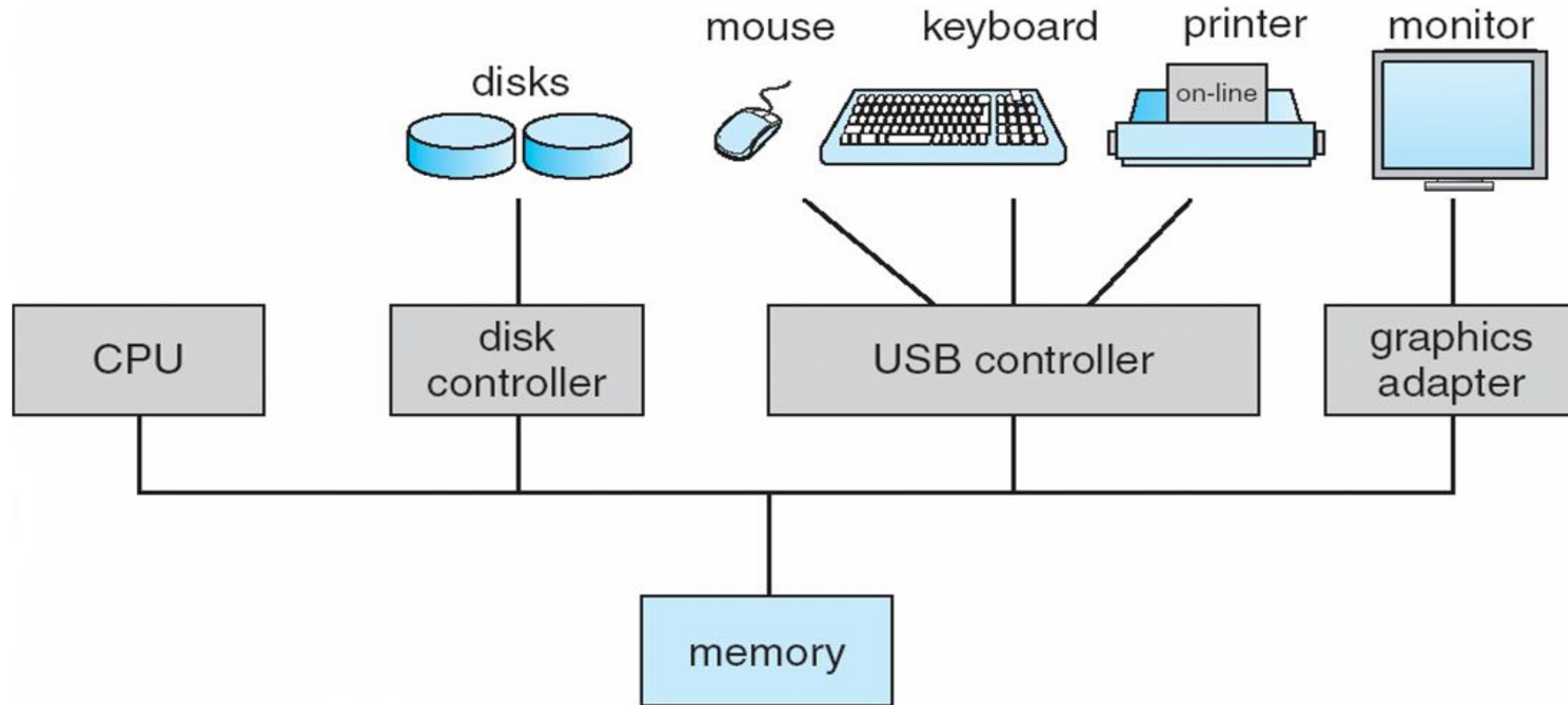
Computer System Components

1. **Hardware** – provides basic computing resources (CPU, Memory, I/O devices, Communication).

2. **Operating System** – controls and coordinates use of the hardware among various application programs for various users.

3. **System & Application Programs** – ways in which the system resources are used to solve computing problems of the users (Word processors, Compilers, Web browsers, Database systems, Video games).

4. **Users** – (People, Machines, other computers).

What happens when we start a computer?? (Booting Process)

- Hardware doesn't know where the operating system resides and how to load it.

- **Bootstrap Program :**
  - Initial program to run a system
  - Locating and Loading OS Kernel in main memory

- Where it is stored ??? ROM

> •If any storage device/partition contains one special program called as **"bootstrap program"** in its first sector i.e. in a boot sector then such a device/partition is referred as **bootable device/partition.**
> •e.g. hard disk drive, pen drive, CD/DVD

Steps of Booting

1. Machine Boot

- When we switch on the power current gets passed to the motherboard and one program gets invoked named as **"BIOS"** which is exists in the ROM memory on motherboard.

- BIOS -- Basic Input Output System -- which is **a micro-program.**

- A micro-program is a program which is smaller in size and can be stored into the memory with its all possible set of input values.

- first step of BIOS is **"POST" - Power On Self Test**, under POST BIOS checks whether all peripherals are connected properly or not and their working status.

- **"peripherals or peripheral devices" -**- devices which are connected to the motherboard externally are called as peripherals.

- after POST BIOS executes **"bootstrap loader",** bootstrap loader searches for available bootable devices and selects any one out of it as per the defined priorities.
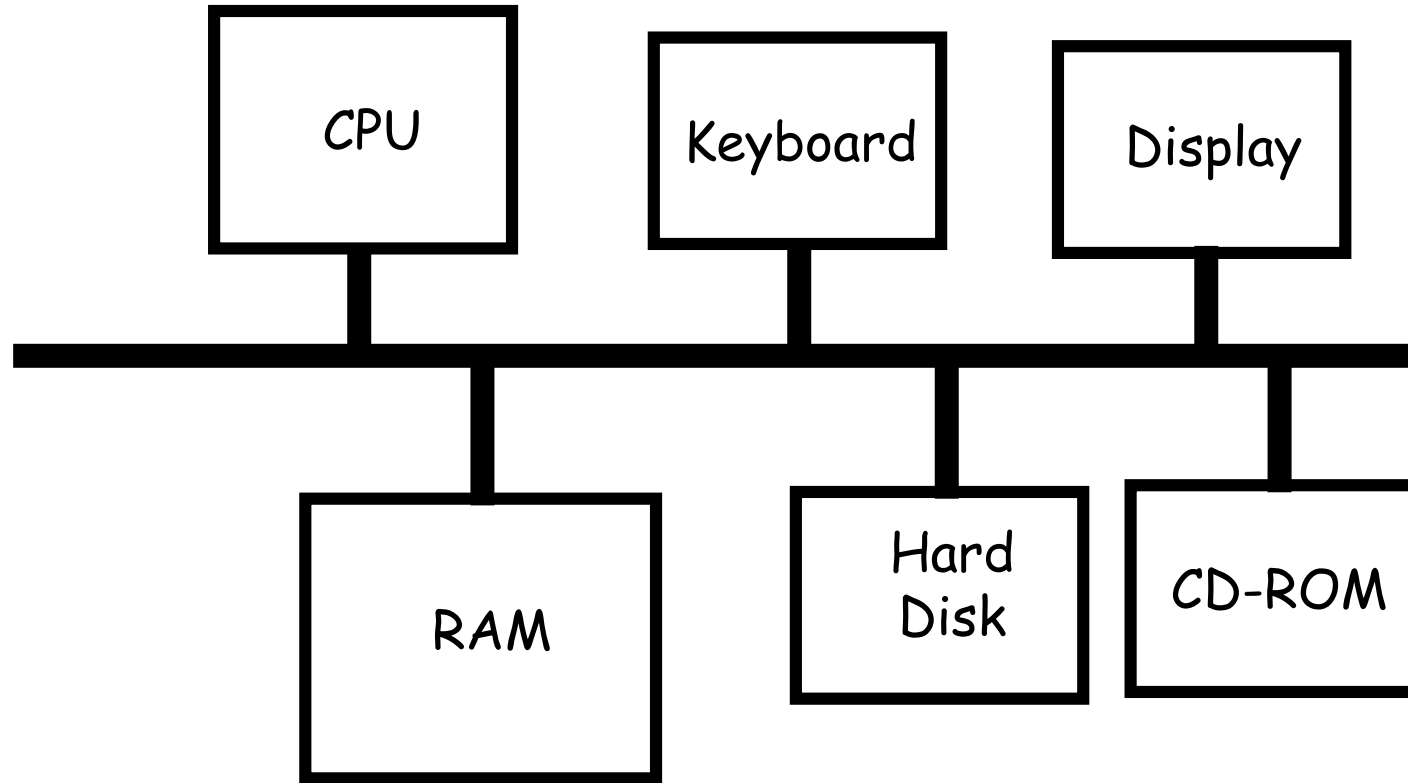
2.System boot:

- if hard disk drive got selected as a bootable device and if it contains multiple OS's have installed on it, then "**bootloader**" program gets executes.

- **Boot loader program** displays list of operating system installed onto the machine, so that user can select any one at a time from and it invokes bootstrap program of selected operating system.

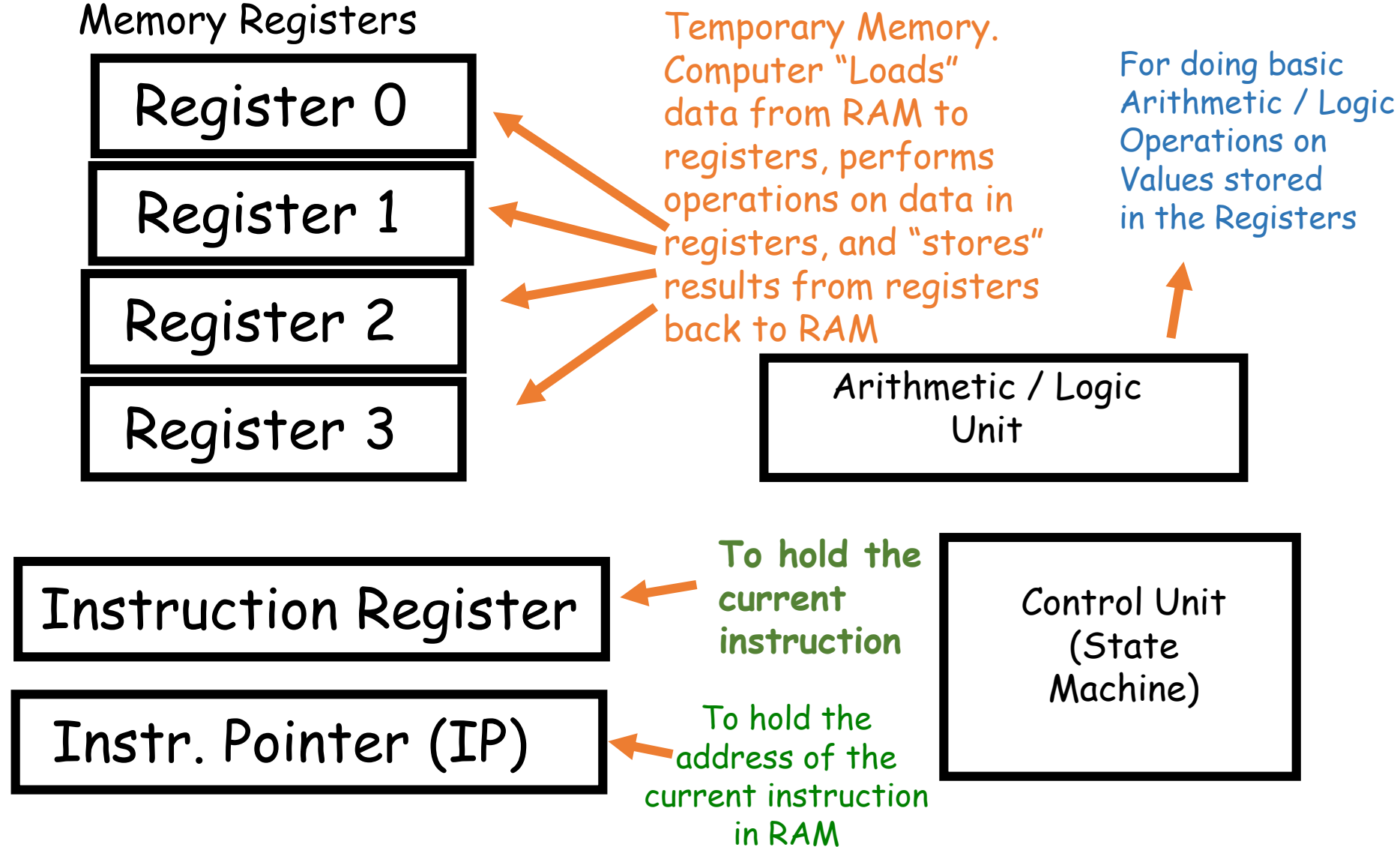- Bootstrap program locates the kernel and load it into the main memory.

Computer Fundamentals

```
┌─────────┐      ┌──────────┐      ┌─────────┐
│   CPU   │      │ Keyboard │      │ Display │
│         │      │          │      │         │
└────┬────┘      └────┬─────┘      └────┬────┘
     │                │                 │
═════╪════════════════╪═════════════════╪══════
     │                │                 │
┌────┴────┐      ┌────┴─────┐      ┌────┴────┐
│         │      │  Hard    │      │         │
│         │      │  Disk    │      │ CD-ROM  │
│   RAM   │      │          │      │         │
└─────────┘      └──────────┘      └─────────┘
```

It is a system concept integrating software and hardware to specify the design of computing systems

CPU

Memory Registers

| Register 0 |
| Register 1 |
| Register 2 |
| Register 3 |

Temporary Memory. Computer "Loads" data from RAM to registers, performs operations on data in registers, and "stores" results from registers back to RAM

For doing basic Arithmetic / Logic Operations on Values stored in the Registers

Arithmetic / Logic Unit

Instruction Register

To hold the current instruction

Control Unit (State Machine)

Instr. Pointer (IP)

To hold the address of the current instruction in RAM

Bus,CU,ALU

Bus

- It is a simplified way for many devices to communicate to each other.
- It is internal arrangement of computer system which includes design of the processor , memory and input/output units.

Control Unit

- Control is responsible for determining what action is to be performed on what data.
- controls all operations and it controls devices which are connected to the computer system by coordinating with device controllers.
- Fetch-Decode-Execute

ALU (Arithmetic Logic Unit)

- ALU is mainly comprised of **logic gates**, circuits made from transistors that take inputs.
- ALU performs all arithmetic and logical operations.

Input/Output

- The Input unit allows programs and data to be entered into the computer.
- The Output unit allows the results of processing to be exported to the outside world or other devices or saved to be used later.
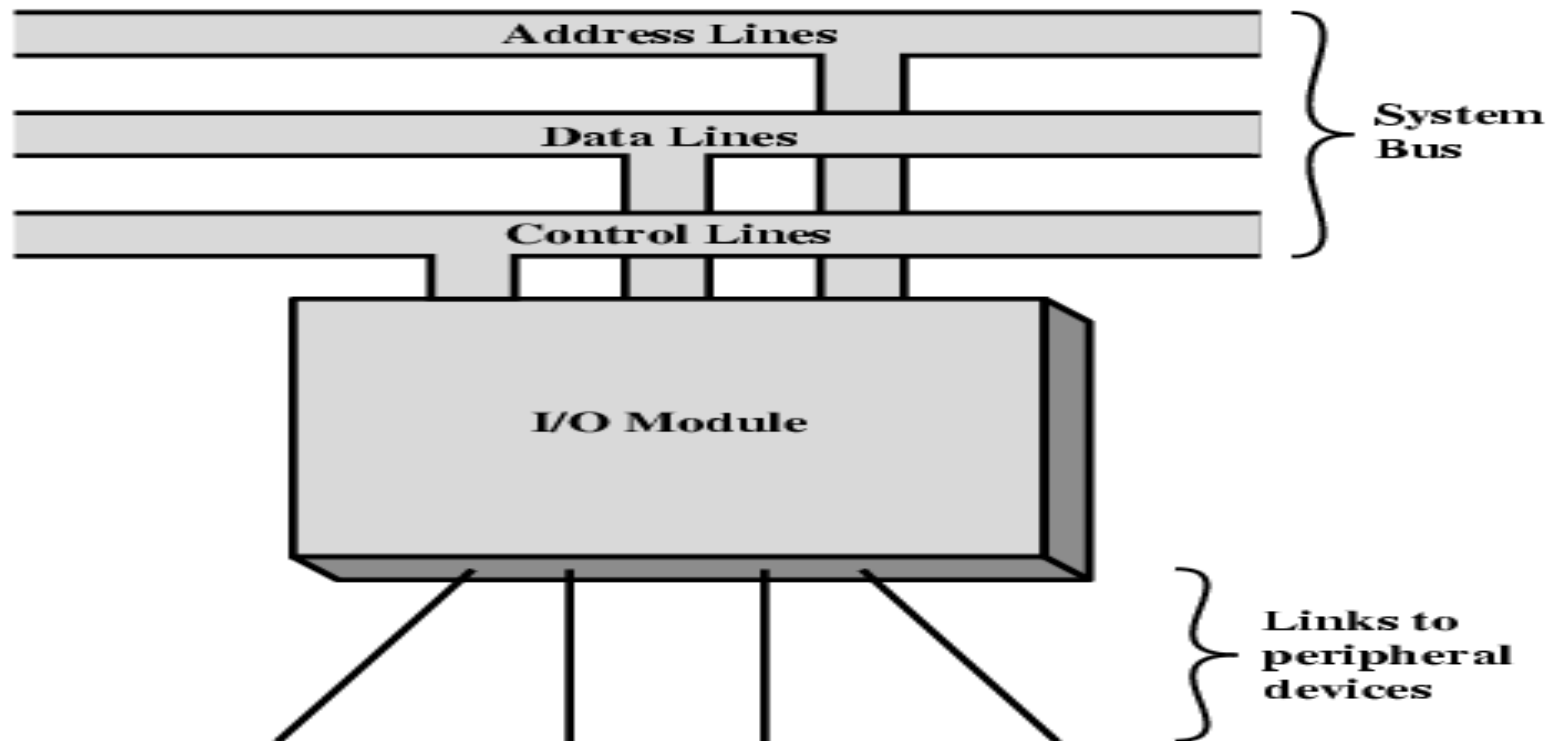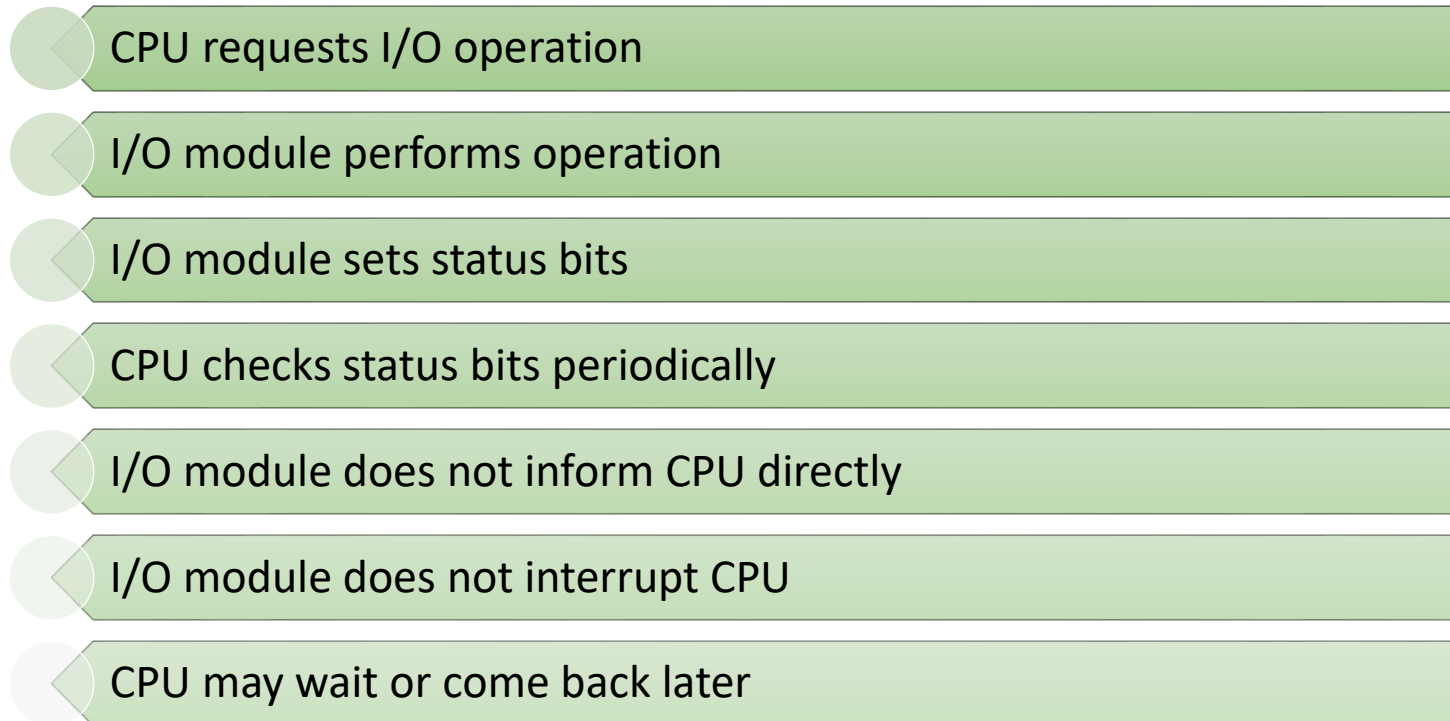
Input Output Techniques

- Communication between memory and IO devices.

- IO Techniques:
  - Programmed IO:
    - CPU waits for IO operations to be completed
    - As CPU is faster so time is wasted
  - Interrupt driven IO
    - CPU issues a command, and proceed for its work until interrupt by IO device
  - Direct Memory Access (DMA)
    - transferring data within main memory and external device without passing it through the CPU.
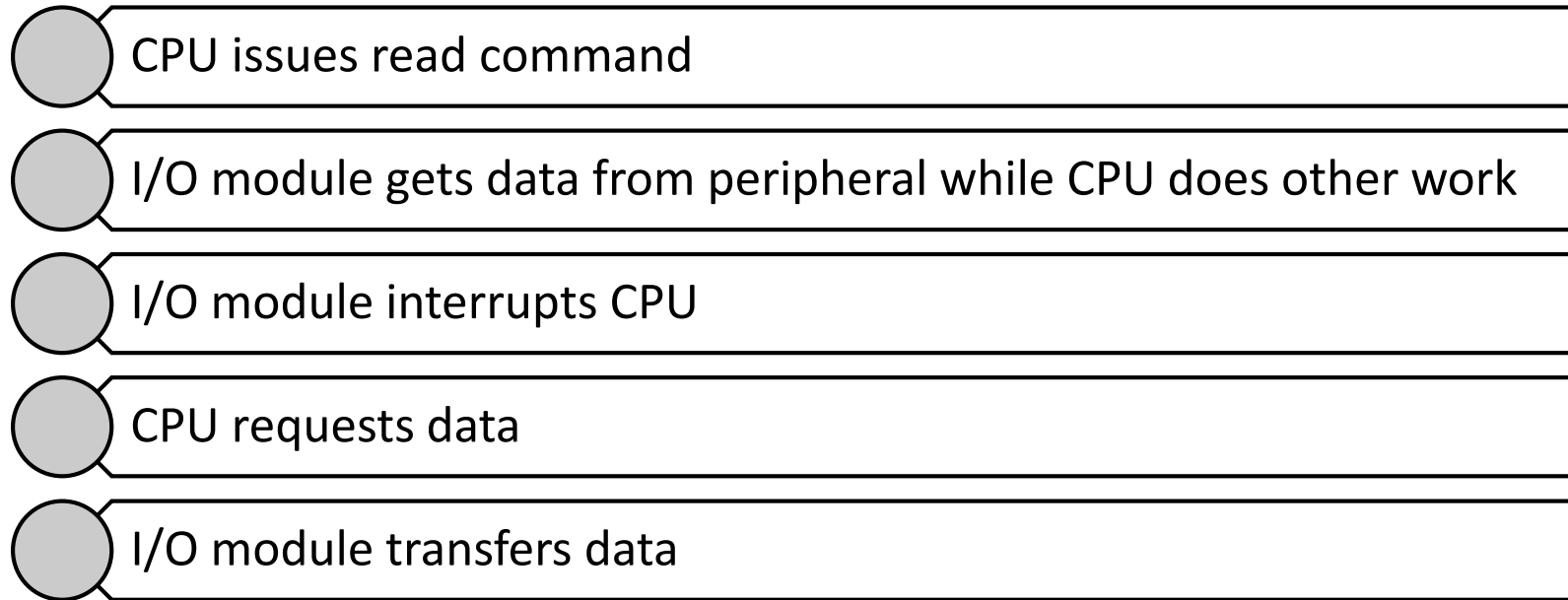
Programmed I/O

- CPU has direct control over I/O
  - Sensing status
  - Read/write commands
  - Transferring data

- CPU waits for I/O module to complete operation

- Four IO Commands : Control, Test, Read, Write

CPU requests I/O operation

I/O module performs operation

I/O module sets status bits

CPU checks status bits periodically

I/O module does not inform CPU directly

I/O module does not interrupt CPU

CPU may wait or come back later
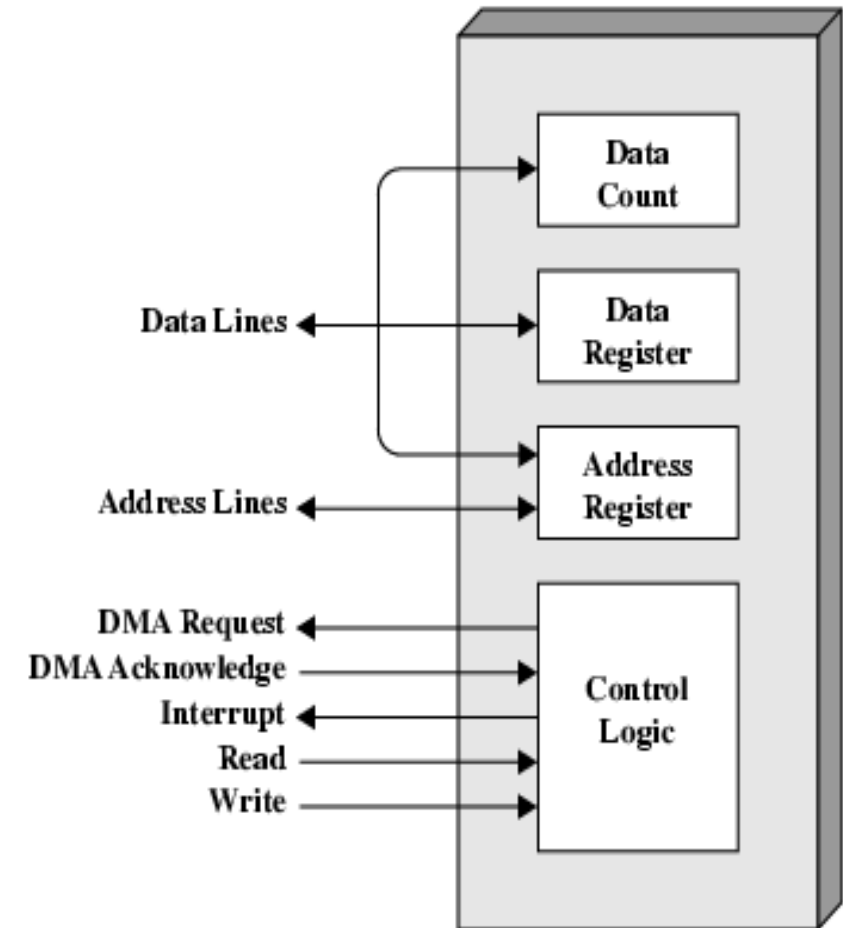
Interrupt Driven I/O

- Overcomes CPU waiting
- No repeated CPU checking of device
- I/O module interrupts when ready
- ISR (Interrupt Service Routine)
  - the processors enter an ISR
- **IVT and ISR**

CPU issues read command

I/O module gets data from peripheral while CPU does other work

I/O module interrupts CPU

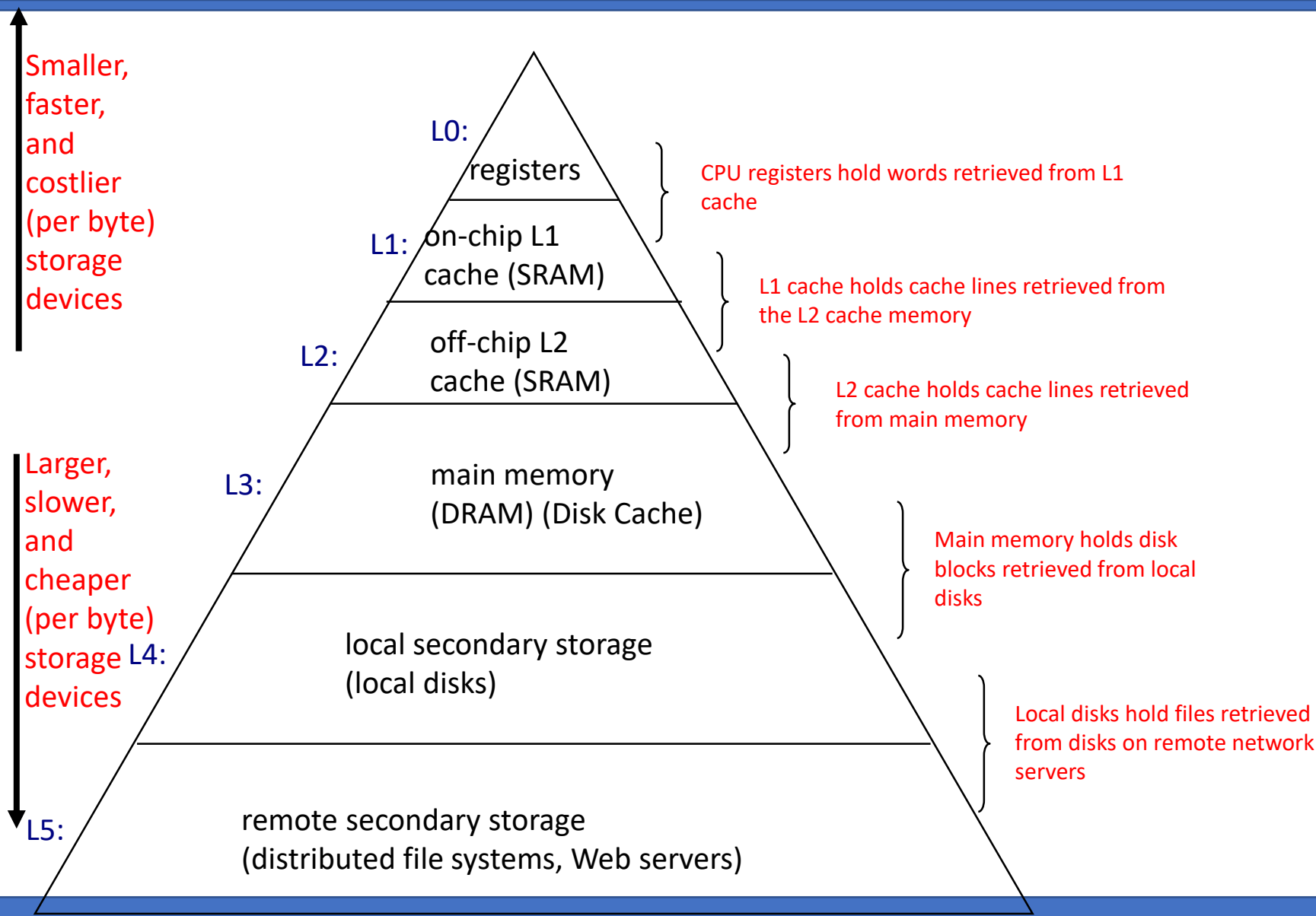CPU requests data

I/O module transfers data

Direct Memory Access

- Interrupt driven and programmed I/O require active CPU intervention
    - Transfer rate is limited
    - CPU is tied up
- DMA Operations:
    - When the processor wishes read or send a block of data, it issues a command to the DMA module by sending some information to DMA module.
    - The information includes:
        - read or write command, sending **through read and write control lines**.
        - number of words to be read or written, communicated on the **data lines** and stored in the **data count register.**
        - starting location in memory to read from or write to, communicated on data lines and stored in the **address register**.
        - **address of the I/O device** involved, communicated on the **data lines**.

        When the transfer is complete, the DMA module sends an interrupt signal to the processor to inform that it has finish using the system bus

Memory Hierarchy

Smaller,
faster,
and
costlier
(per byte)
storage
devices

Larger,
slower,
and
cheaper
(per byte)
storage
devices

L0: registers

L1: on-chip L1 cache (SRAM)

L2: off-chip L2 cache (SRAM)

L3: main memory (DRAM) (Disk Cache)

L4: local secondary storage (local disks)

L5: remote secondary storage (distributed file systems, Web servers)

CPU registers hold words retrieved from L1 cache

L1 cache holds cache lines retrieved from the L2 cache memory

L2 cache holds cache lines retrieved from main memory

Main memory holds disk blocks retrieved from local disks

Local disks hold files retrieved from disks on remote network servers

Cache & Main Memory

- Small amount of fast memory
- Sits between normal main memory and CPU
- May be located on CPU chip or module
  - An entire blocks of data is copied from memory to the cache because the principle of locality tells us that once a byte is accessed, it is likely that a nearby data element will be needed soon.
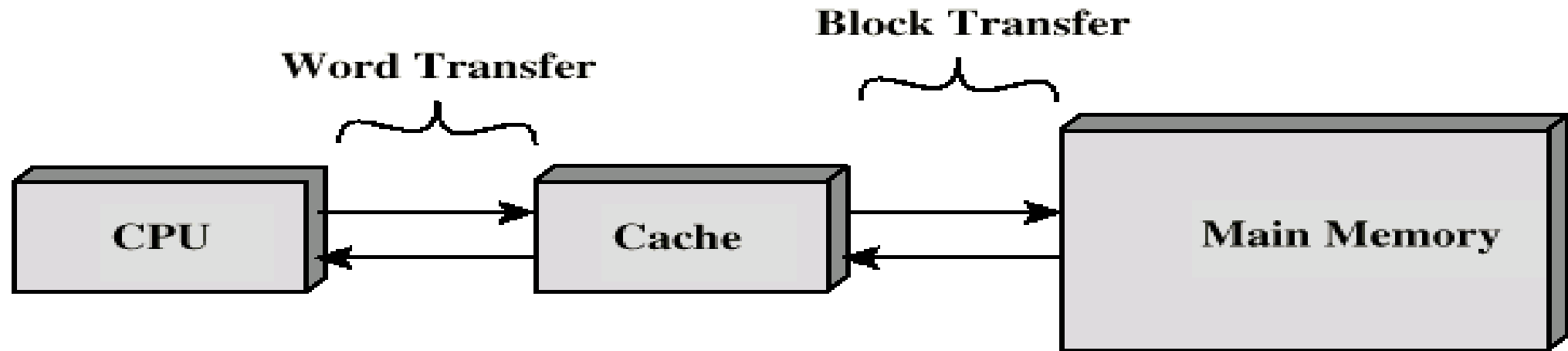
Random-Access Memory (RAM)

- Key features
  - RAM is packaged as a chip, Basic storage unit is a cell (one bit per cell)
  - Its internal memory of the CPU for storing data, program, and program result
  - Used for Read/ Write
  - Volatile (Temporary Storage)

Static RAM (SRAM)	Dynamic RAM (DRAM)
• memory retains its contents as long as power is being supplied. • Made up of transistor • Static because it doesn't need to be refreshed • SRAM is more often used for system cache. • SRAM is faster than DRAM	• memory must be constantly refreshed or it will lose its contents. • This is done by placing the memory on a refresh circuit that rewrites the data several hundred times per second • Made up of memory cells composed of capacitors and one transistor. • DRAM is typically used for the main memory in computing devices

ROM(Read Only Memory)

- The memory from which we can only read but cannot write on it.

- This type of memory is non-volatile.

- The information is stored permanently in such memories during manufacture.

- A ROM stores such instructions that are required to start a computer. This operation is referred to as **bootstrap.**

Memory Access Method

Sequential	Direct	Random	Associative
• Start at the beginning and read through in order • Access time depends on location of data and previous location • e.g. tape	• Individual blocks have unique address • Access is by jumping to vicinity plus sequential search • Access time depends on location and previous location • e.g. disk	• Individual addresses identify locations exactly • Access time is independent of location or previous access • e.g. RAM	• Data is located by a comparison with contents of a portion of the store • Access time is independent of location or previous access • e.g. cache

Thank You