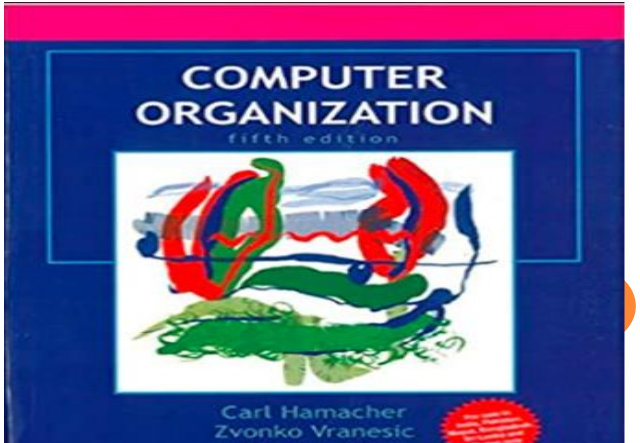
Introduction and Overview of Computer Architecture
Module 1 Part A

Dr. Ilavarasi A K,
SCOPE, VIT Chennai

TEXT BOOK(S)

- Carl Hamacher, Zvonko Vranesic, Safwat Zaky, Computer organization, Mc Graw Hill, Fifth edition ,Reprint 2011.
- W. Stallings, Computer organization and architecture, Prentice-Hall, 8th edition, 2009

EIGHTH EDITION
COMPUTER ORGANIZATION AND ARCHITECTURE
Designing for Performance
WILLIAM STALLINGS



Reference Book

- David A. Patterson and . John L. Hennessy ―Computer Organization and Design-The Hardware/Software Interface‖ 5th edition, Morgan Kaufmann, 2011.

COMPUTER ORGANIZATION AND DESIGN
THE HARDWARE/SOFTWARE INTERFACE
FIFTH EDITION
DAVID A. PATTERSON
& JOHN L. HENNESSY
MK
23 × 500

Grading Policy

- Digital Assignment 1= 10 marks (Written Assignment, Rubric based Evaluation)
- Digital Assignment 2 & 3 = 20 marks (Quiz)
- CAT 1- 2 to 2.5 Modules (15% Weightage)
- CAT 2- 3 Modules (15% Weightage)
- FAT- 100 marks (40 % Weightage)

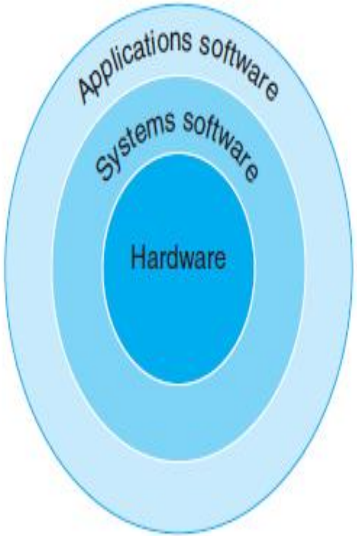
OUTLINE

- Introduction to computer systems
- Overview of Organization and Architecture
- Functional components of a computer
- Registers and register files
- Interconnection architecture

INTRODUCTION TO COMPUTER SYSTEMS

- A computer system is a *electronic device*
  - Accepts digitized input information
  - Process it according to a list of internally stored instructions
  - Produces the resulting output information
- The list of instructions are called computer program
- Internal storage is via computer Memory

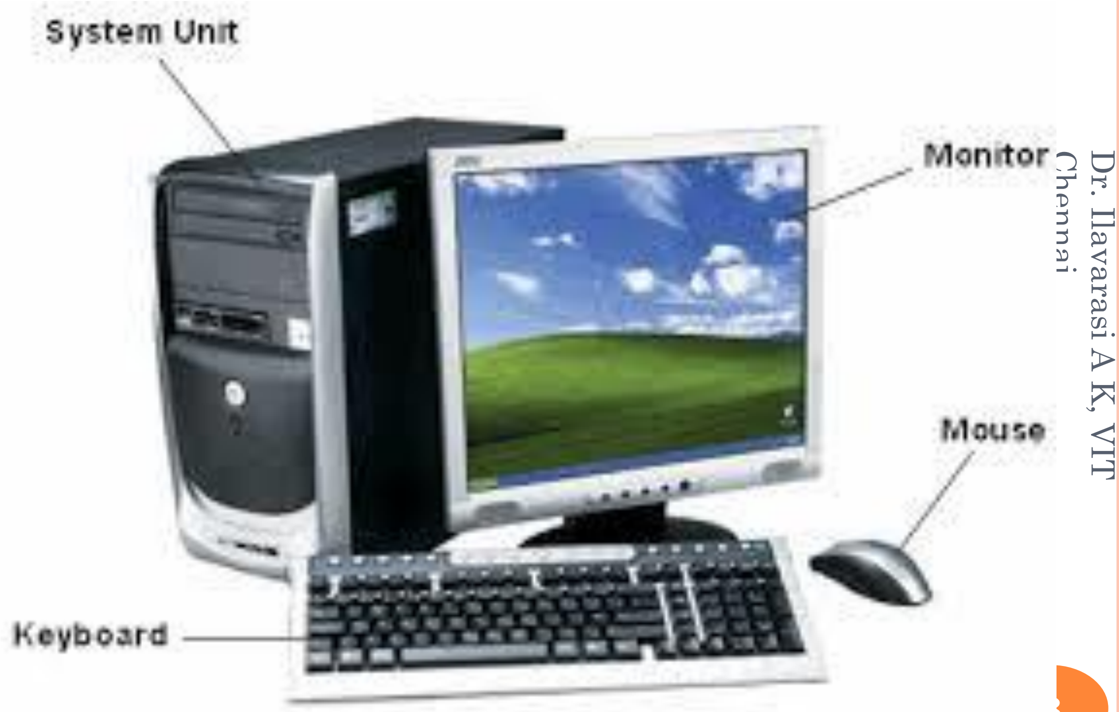
Introduction to computer systems



An user interacts with computer system with the help of application software

Application software communicates with the operating system and returns the results of required operation

Hardware components



SYSTEM SOFTWARE

- Operating System

  Interfaces between a user's program and the hardware and provides a variety of services and supervisory functions

- Compilers

  A program that translates high-level language statements into assembly language statements

- Assembler

  A program that translates assembly language statements in to 1's and 0's

HIGH LEVEL TO HARDWARE LANGUAGE

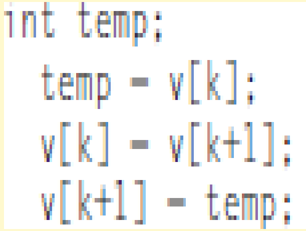
- If programmer writes, Add A,B

  The compiler translate to a symbolic language called as assembly language:

  ```
  Mov A, R1
   Add B,R1
  Mov R1, C
  ```

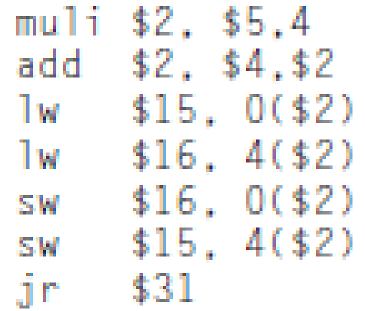
- Assembler translates to binary language that a machine understands and is called as machine language, Eg. 1000110010100000

Power of abstraction



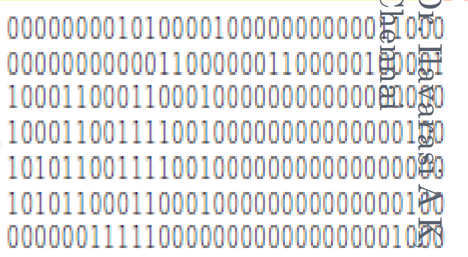
```
int temp;
  temp = v[k];
  v[k] = v[k+1];
  v[k+1] = temp;
```

High level language program in C



```
muli  $2, $5,4
add   $2, $4,$2
lw    $15, 0($2)
lw    $16, 4($2)
sw    $16, 0($2)
sw    $15, 4($2)
jr    $31
```

Assembly language program



```
00000000101000010000000000011000
00000000000110000001100000100001
10001100011000100000000000000000
10001100111100100000000000000100
10101100111100100000000000000000
10101100011000100000000000000100
00000011111000000000000000001000
```

Binary machine language program

Computer Organization?

- Encompasses all physical/hardware aspects of computer systems.

Eg., circuit design, control unit, memory chip.

- How does a computer work?

Computer Architecture?

- Logical aspects of system implementation as seen by the programmer.
- Eg., instruction sets, instruction formats, addressing modes.
- *How do I design a computer?*

ORGANIZATION VS. ARCHITECTURE

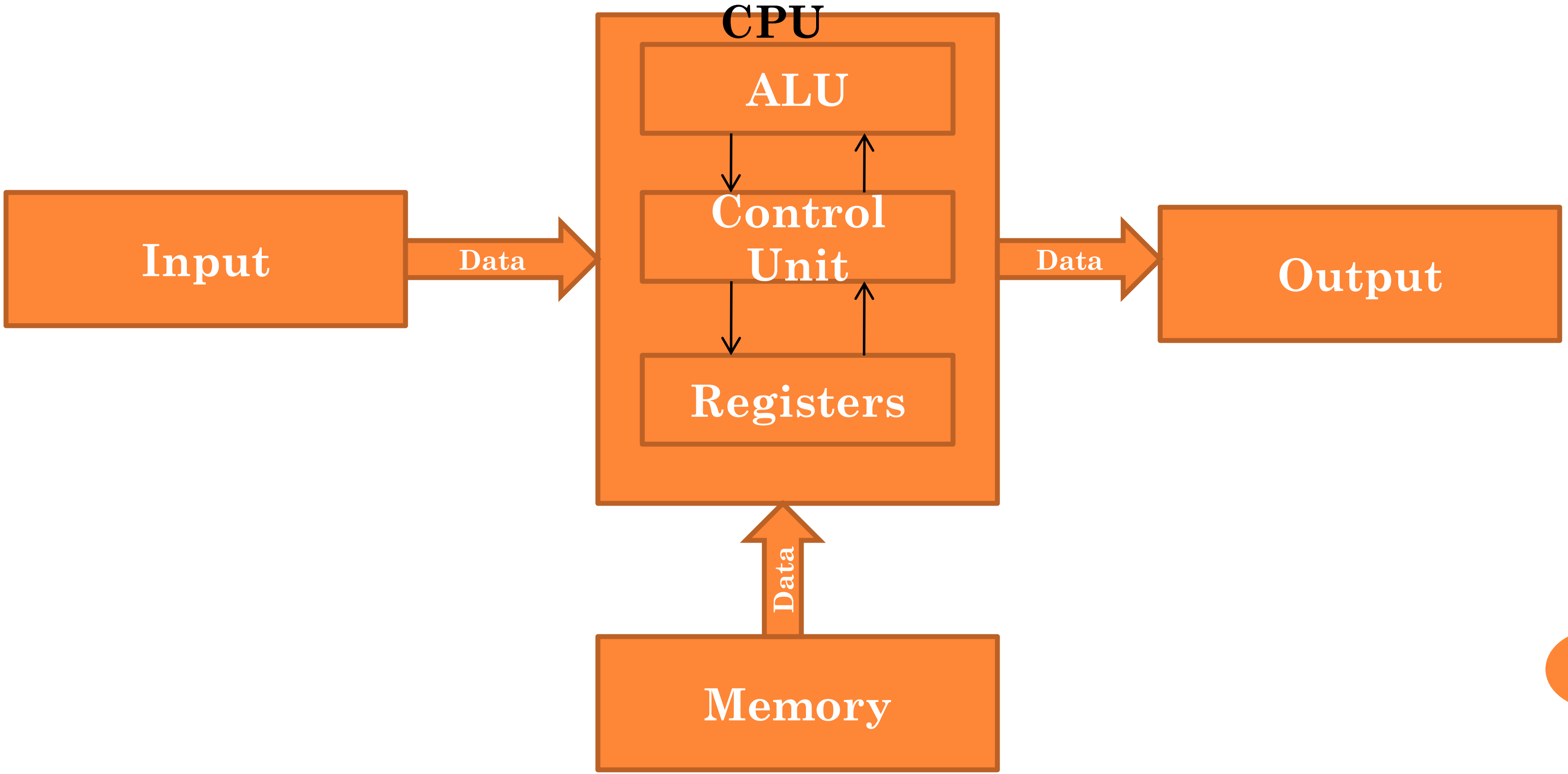
Computer Organization

- **Physical aspects** of a computer system
- Study on how various circuits and components fit together to create a working computer

Computer Architecture

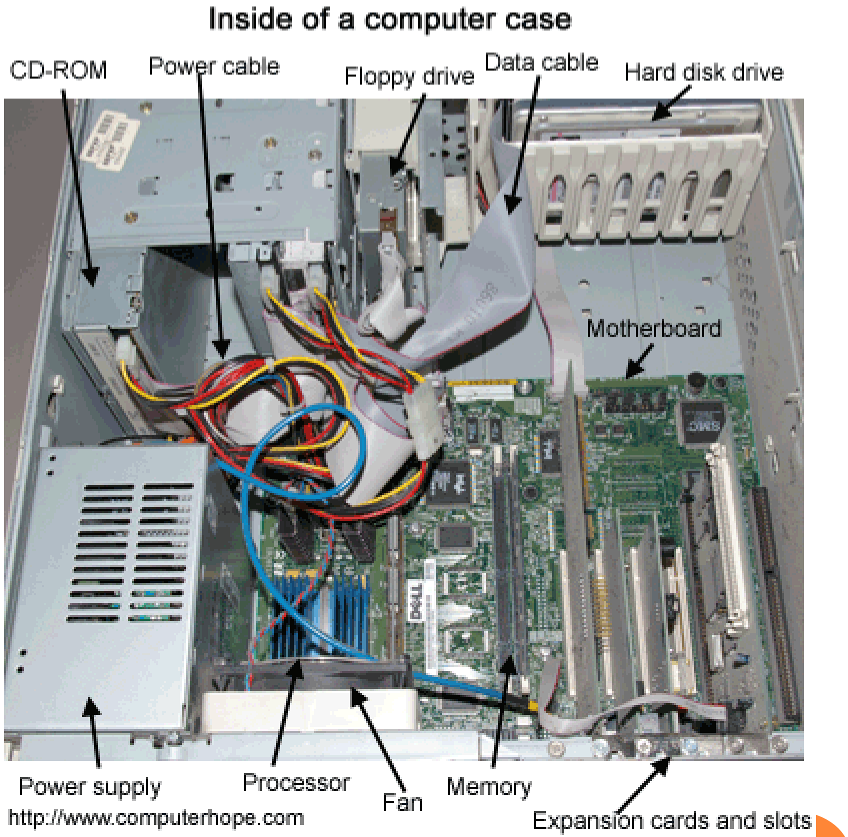
- **Logical Aspects** of a computer system
- Focuses on the structure and behavior of a computer system

FUNCTIONAL COMPONENTS OF A COMPUTER



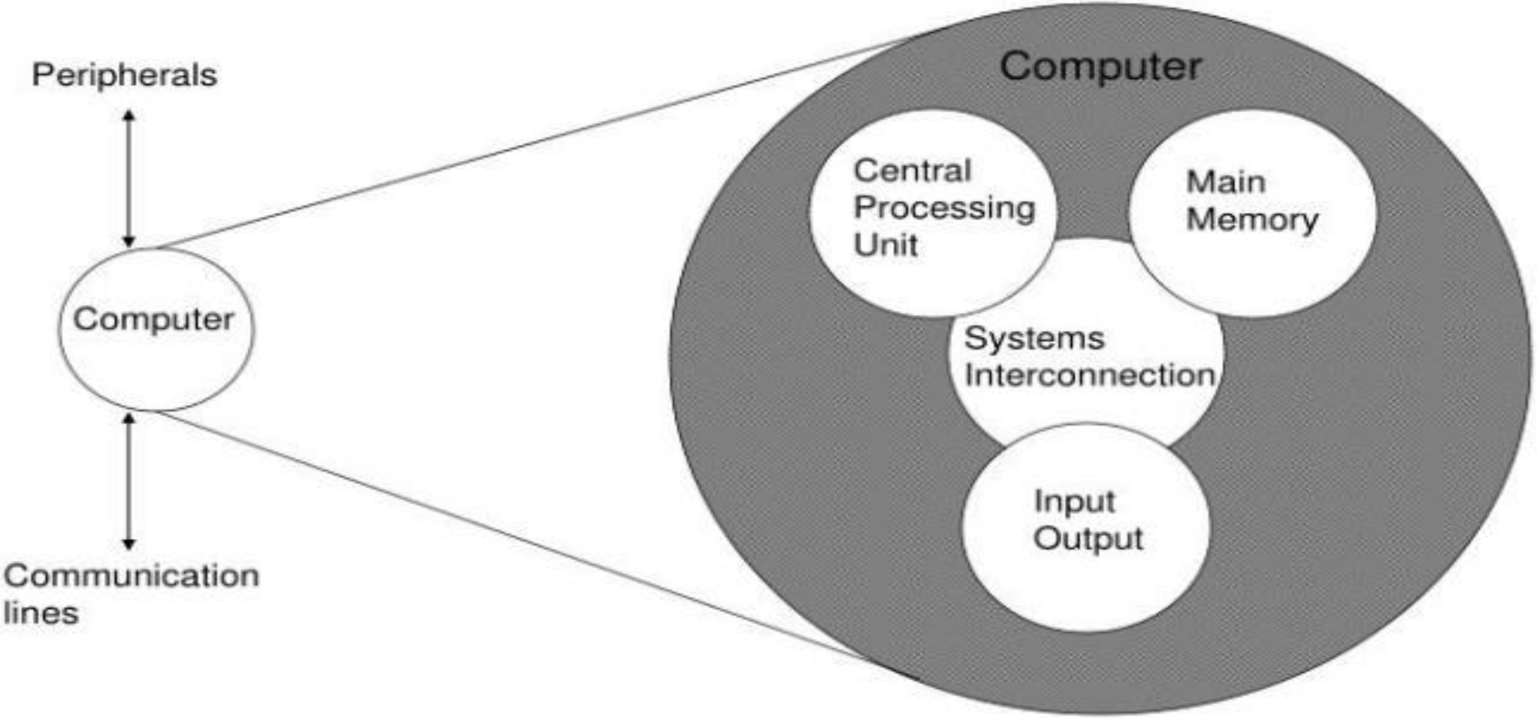
Dr. Ilavarasi A K, VIT Chennai

Functional Components of a Computer

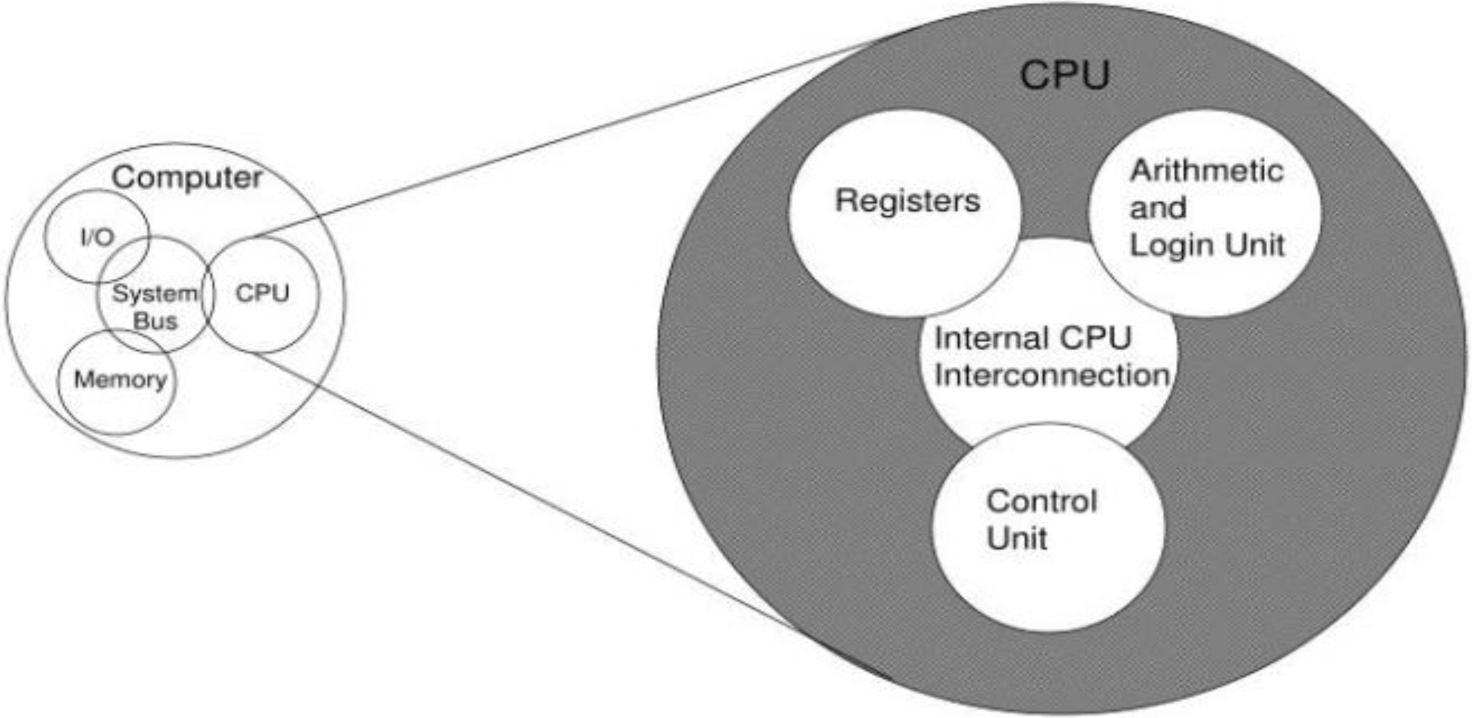


http://www.computerhope.com

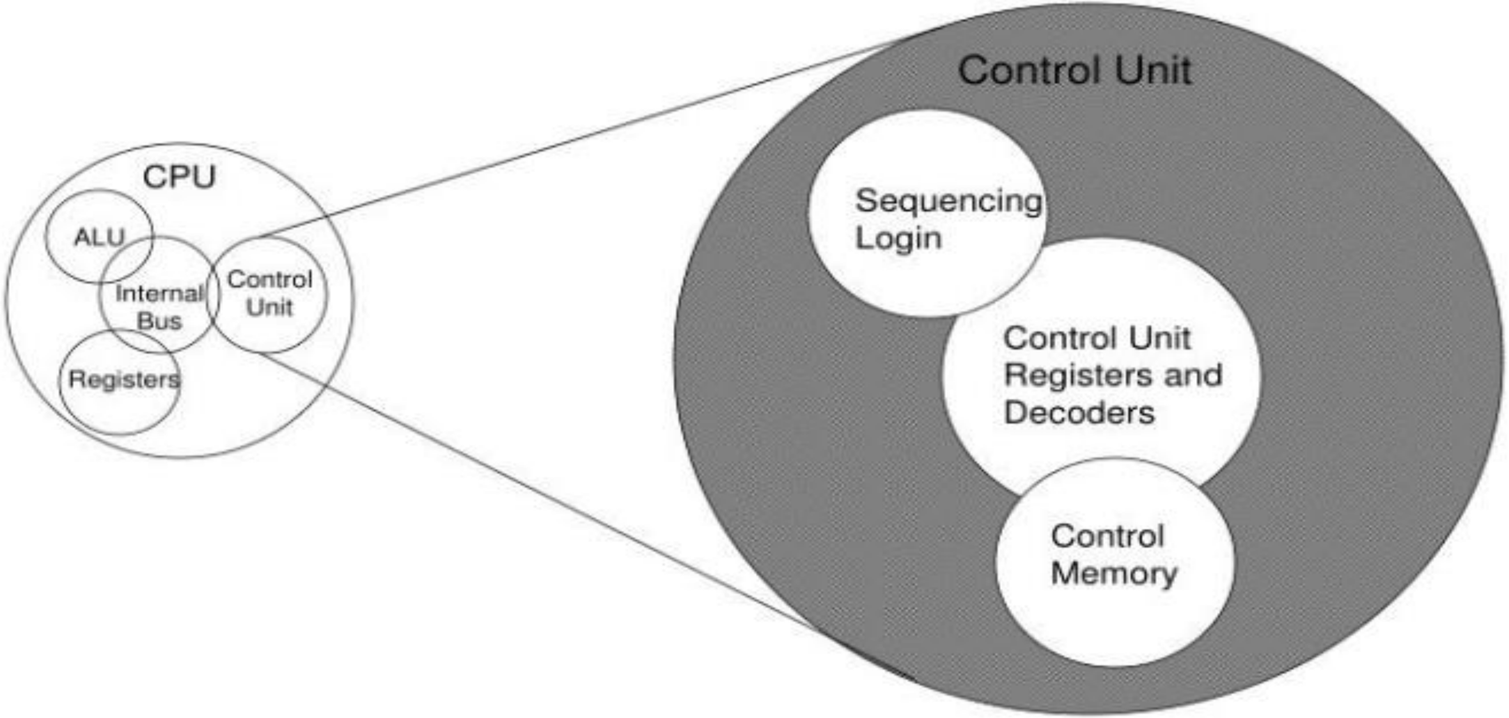
Structure - Top Level



Structure - The CPU



Structure - The Control Unit



CPU

ALU (Arithmetic & Logic Unit): This unit performs the data processing functions of a computer

Control unit:

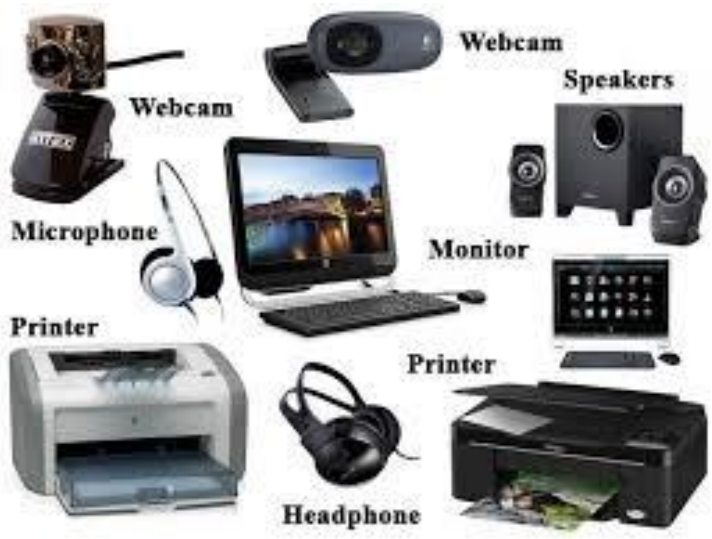
- ❑ Coordinates the tasks between the computer components.
- ❑ It takes the data from input devices
- ❑ Sends the input to processing units
- ❑ Transmits the final result/data to output devices.

Registers: This is a storage unit within the CPU. The data in this unit is collected from the higher memory components (i.e. cache).

INPUT UNIT

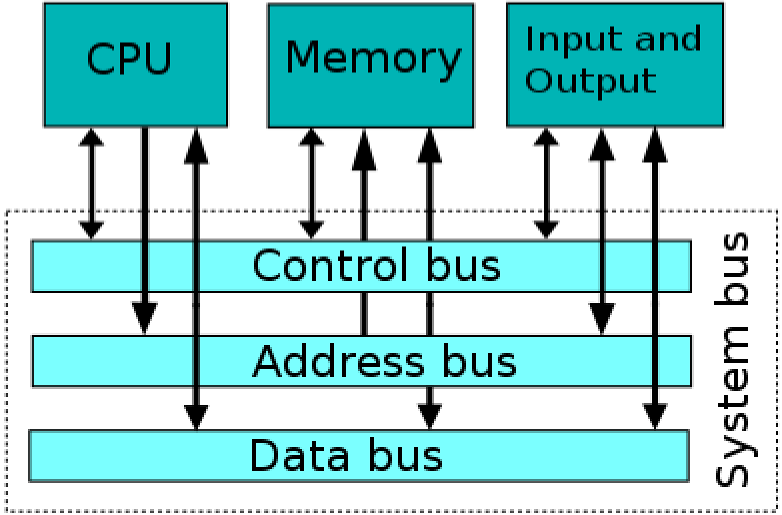
- Input units are used to provide the input data to the computer to solve the problem given to the computer.
- The commonly used input devices are shown in the diagram.

OUTPUT UNIT



- The output devices are used to produce the results which are processed by the computer system.
- The most commonly used output devices are shown in the diagram.

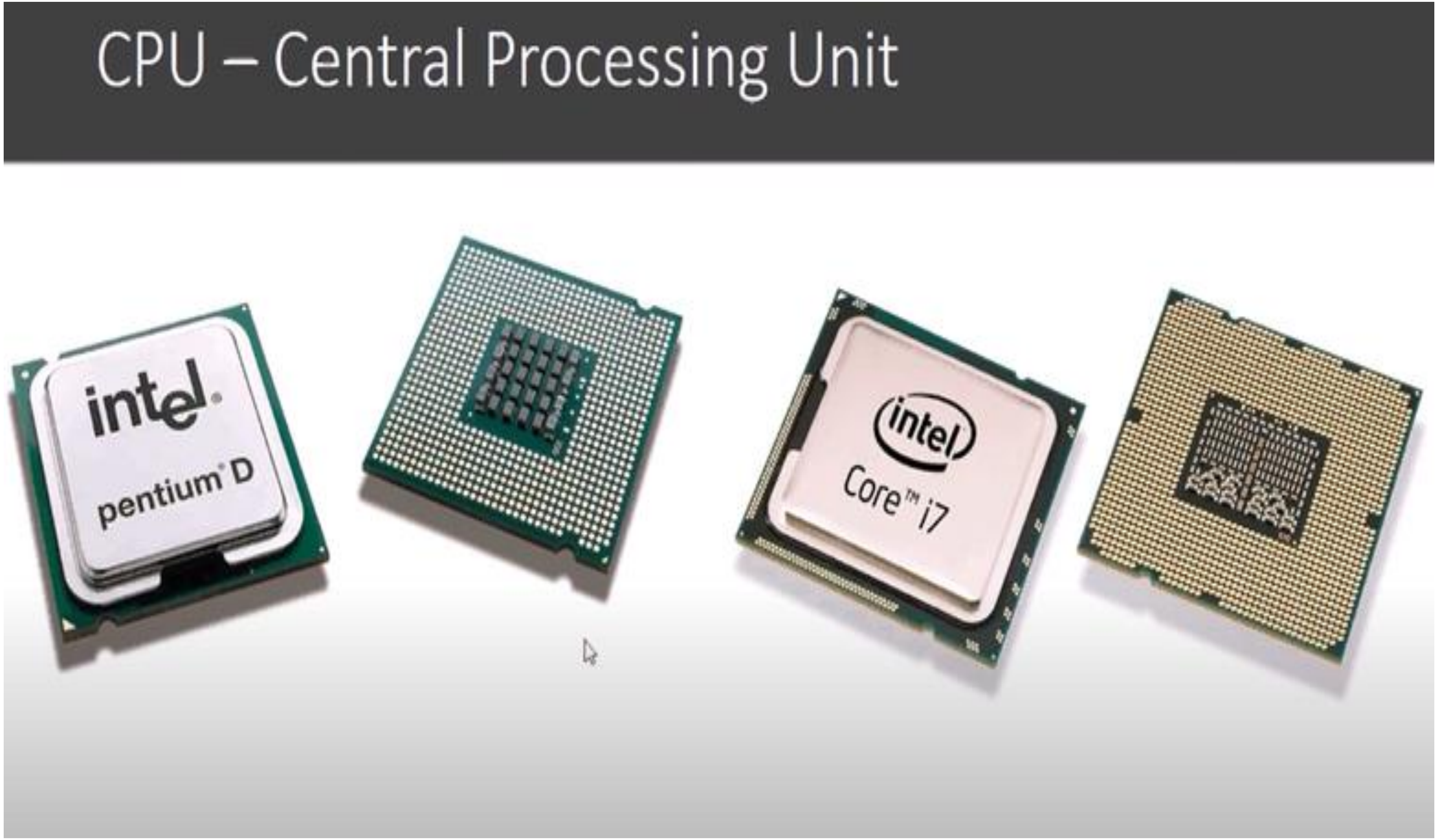
System interconnection



- A mechanism that provides for communication among various components of a computer like CPU, main memory, secondary memory etc.
- *System bus* is very common mechanism for connecting such components

SYSTEM INTERCONNECTION-SYSTEM BUS

- *System bus* is a very common mechanism for providing communication path for transfer of data across the functional units of a computer like CPU, main memory and I/O devices

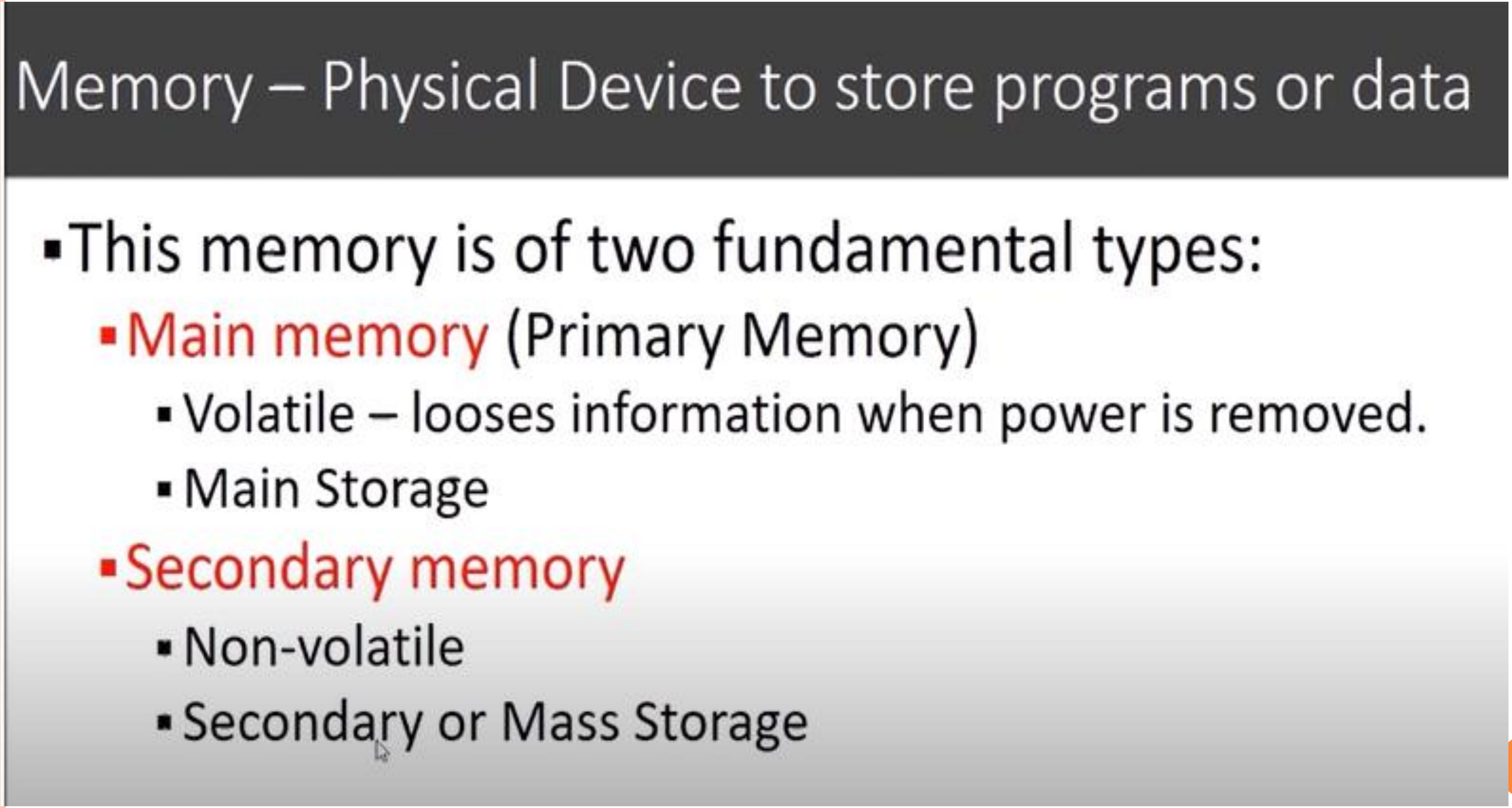


CPU – Central Processing Unit

Where operating system will be stored in Computer?

Where operating system will be stored in Computer?

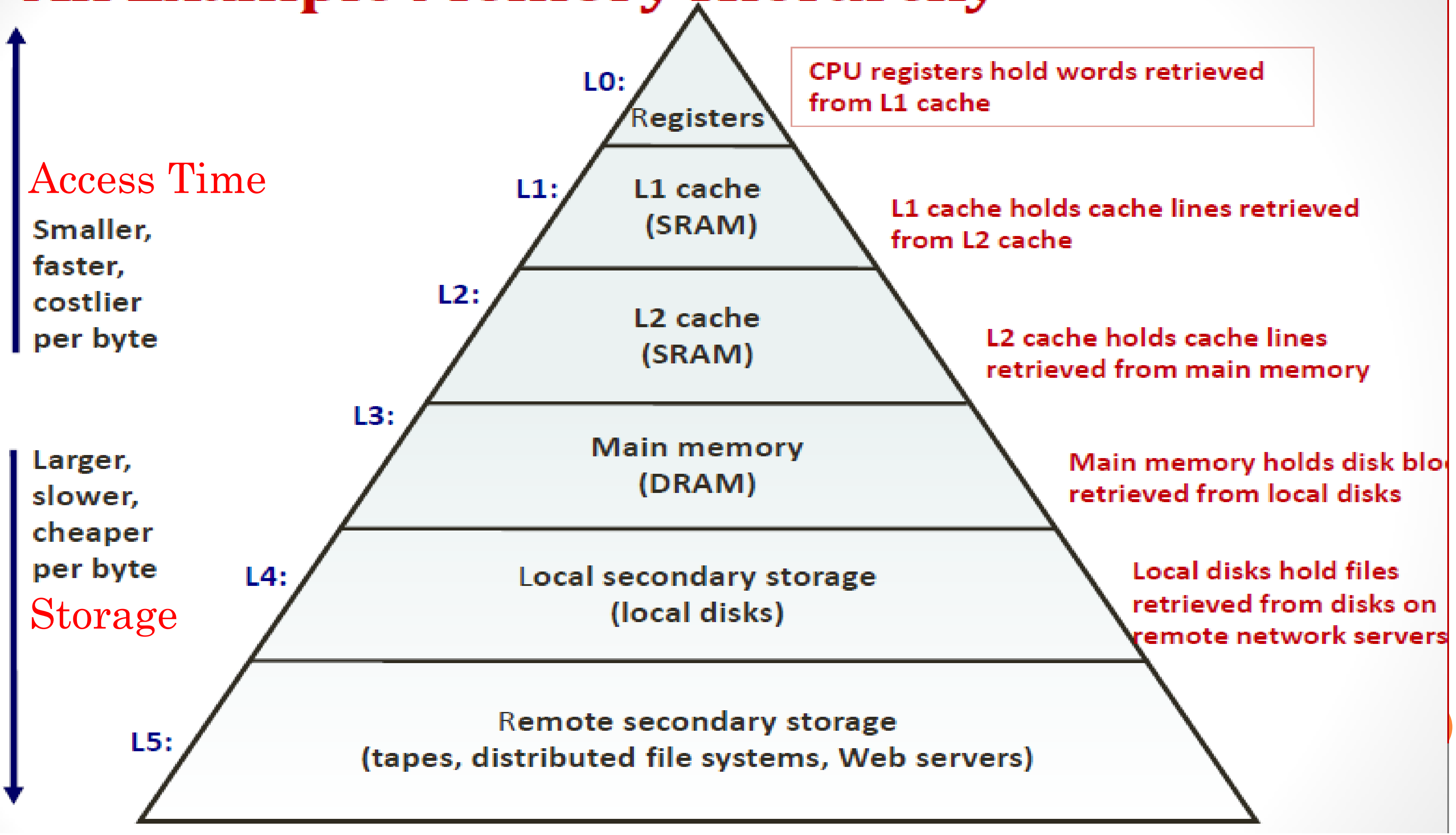
- It resides on your **Hard Drive**. If you're a Windows user, it's on your C drive by default.



Memory – Physical Device to store programs or data

- This memory is of two fundamental types:
  - Main memory (Primary Memory)
    - Volatile – looses information when power is removed.
    - Main Storage
  - Secondary memory
    - Non-volatile
    - Secondary or Mass Storage

An Example Memory Hierarchy



Access Time

Smaller,
faster,
costlier
per byte

Larger,
slower,
cheaper
per byte

Storage

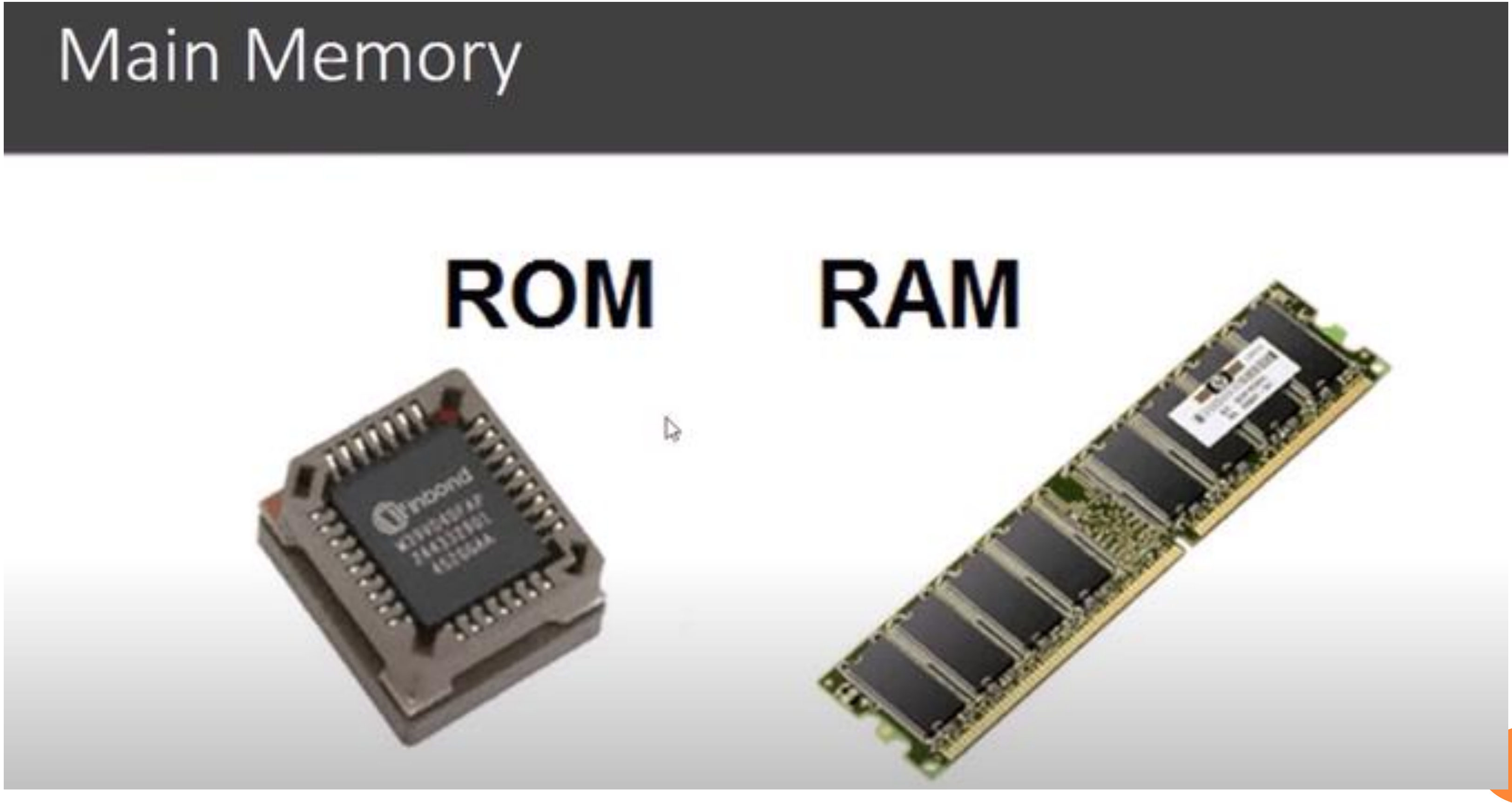
CPU registers hold words retrieved from L1 cache

L1 cache holds cache lines retrieved from L2 cache

L2 cache holds cache lines retrieved from main memory

Main memory holds disk blo
retrieved from local disks

Local disks hold files retrieved from disks on remote network servers



Main Memory

ROM

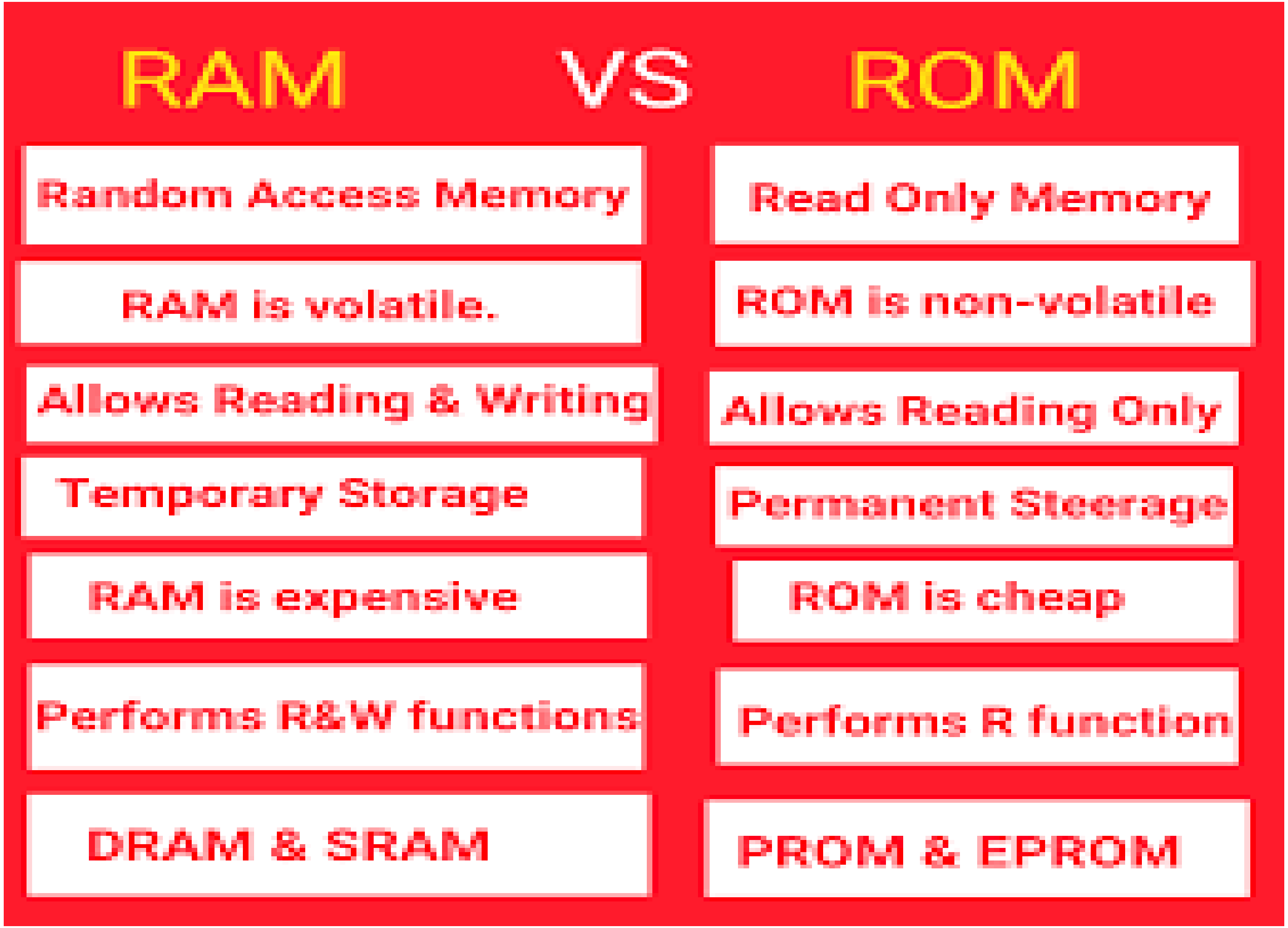
RAM

What is RAM?

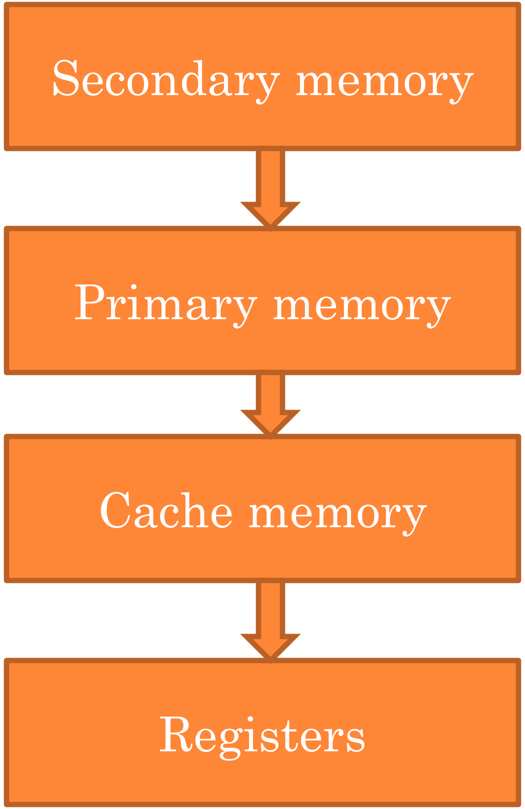
- The information stored in this type of memory is lost when the power supply to the PC or laptop is switched off.

SRAM

DRAM



RAM	VS	ROM
Random Access Memory		Read Only Memory
RAM is volatile.		ROM is non-volatile
Allows Reading & Writing		Allows Reading Only
Temporary Storage		Permanent Steerage
RAM is expensive		ROM is cheap
Performs R&W functions		Performs R function
DRAM & SRAM		PROM & EPROM



REGISTERS

- Register is memory unit inside the processor and capable of high speed processing.
- The memory hierarchy defined in the figure shows the memory levels of high storage capacity to low storage capacity as well as low speed to high speed
- In memory hierarchy secondary memory is less expensive one and register is high expensive memory unit

REGISTERS

Registers have 2 specific roles:

1. **User variable registers:** These registers minimizes main memory references
2. **Control and status registers:** These registers are used by control unit in CPU to control the processor's operation and execution of operating system programs

USER VARIABLE REGISTERS

These registers are 4 types:

1. General purpose
2. Data
3. Address
4. Condition codes

GENERAL PURPOSE REGISTERS

- The programmer assigns these registers for different functions
- These registers can hold the operands of arithmetic operations

Data registers

- These registers only holds the data
- The restriction of these registers is, these can not be used for operand address calculation

ADDRESS REGISTERS

There are different categories of address registers

1. Segment pointers:
   a. These holds segments base address
   b. There may be multiple registers
2. Index registers:
   a. These can be used for indexed addressing
   b. Autoindexed
3. Stack pointers: for user variable stack addressing, top of the stack is pointed by a dedicated register

CONTROL AND STATUS REGISTERS

- The operation of the processor is controlled
- These registers are not visible to the user on most of the machines
- Some of these registers are visible in operating system mode

Control and status registers

There are 4 registers, which are important for execution of an instruction

1. **Program counter:** This register contain an instruction address selected for fetching
2. **Instruction register:** Most recently fetched instruction is stored in this register
3. **Memory address register:** This register holds the address of the instruction in memory
4. **Memory data/ buffer register:** The data in this register is either read from memory or written to memory

CONTROL AND STATUS REGISTERS

Program status word (PSW): This register contains status information. This also contains condition codes as well as some status information. Common fields or flags of this register contain the following:

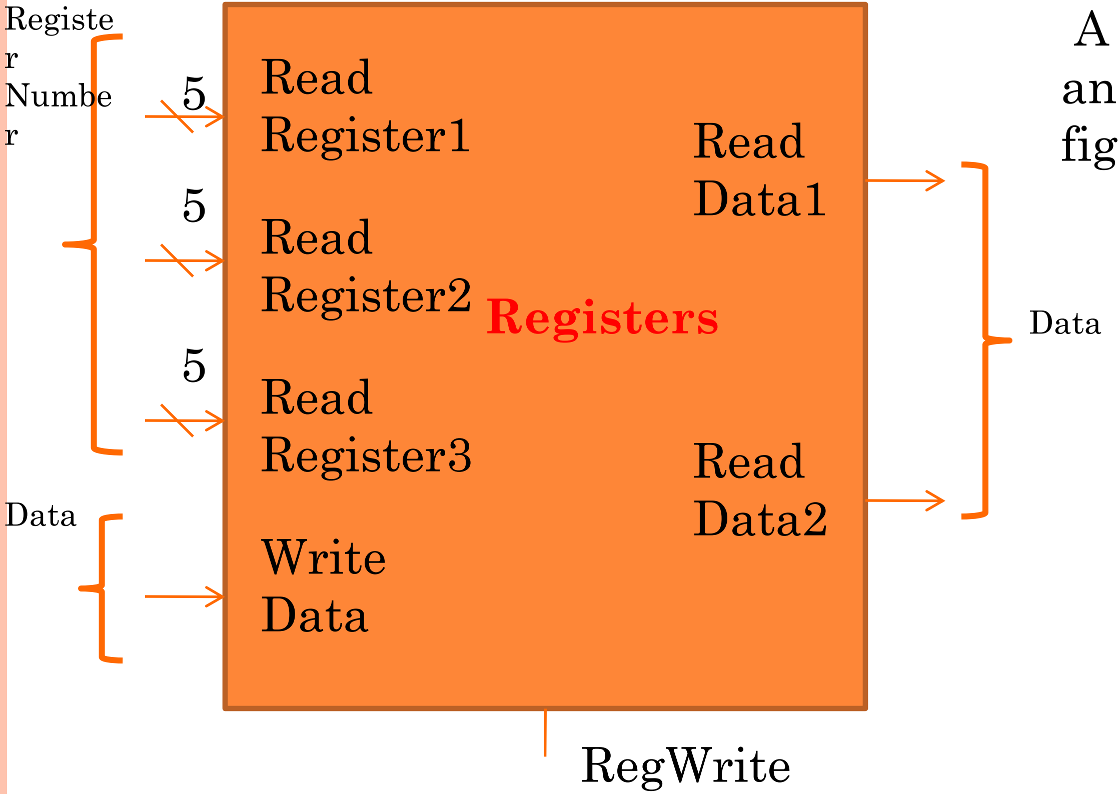
a. Sign
b. Zero
c. Carry
d. Equal
e. Overflow
f. Interrupt enable/disable
g. Supervisor

REGISTER FILE

- Register file is a structure that stores the processor's 32 general purpose registers

- It contains a register state of the computer.

Register file



A register file with four inputs and two outputs are shown in the figure.

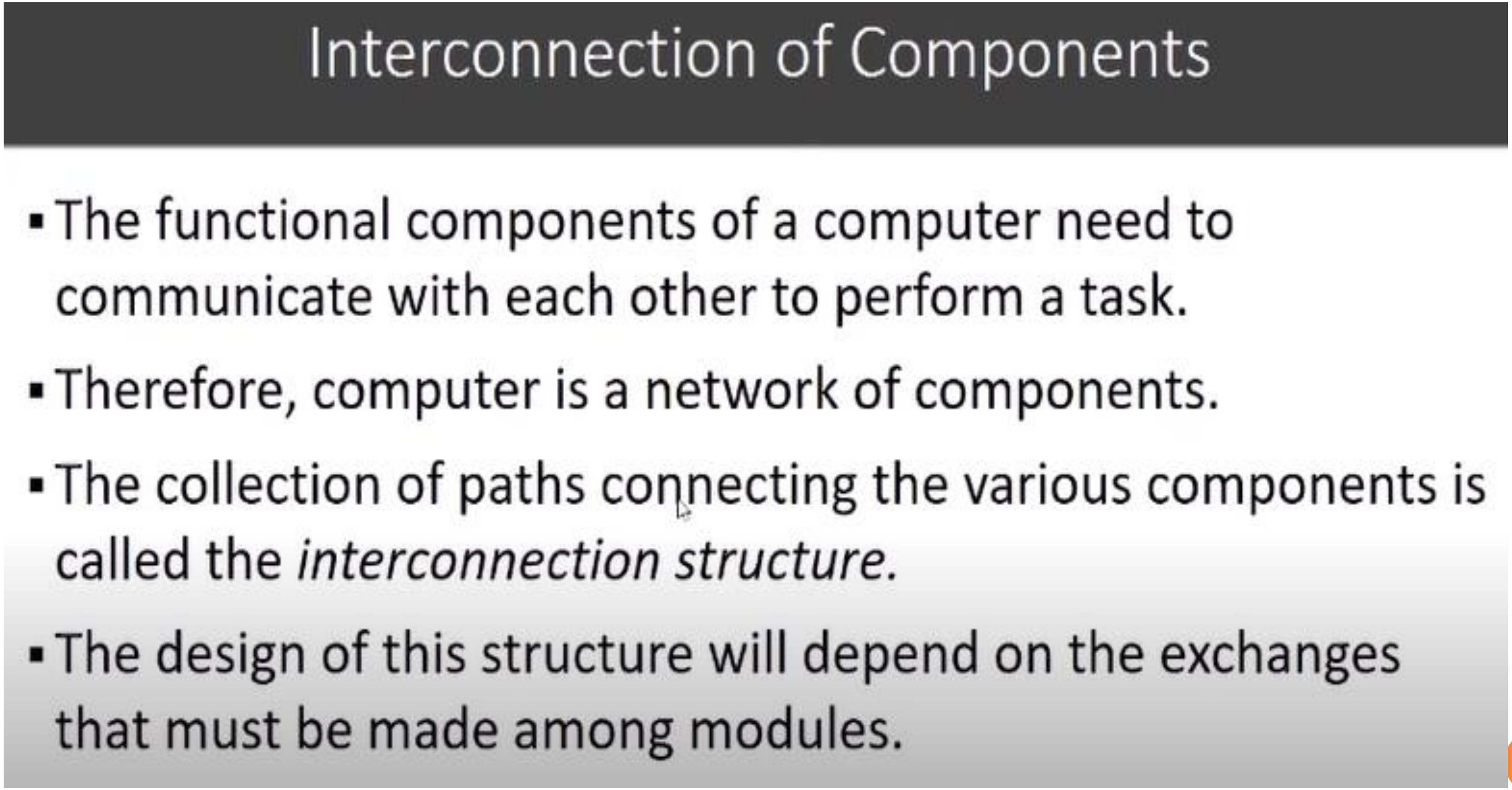
Dr. Ilavarasi A K, VIT Chennai

REGISTER FILE

- For reading of each data word from the registers, the following are needed
  1. Input – number of the register to be read
  2. Output – this will carry the value that has been read from the registers

REGISTER FILE

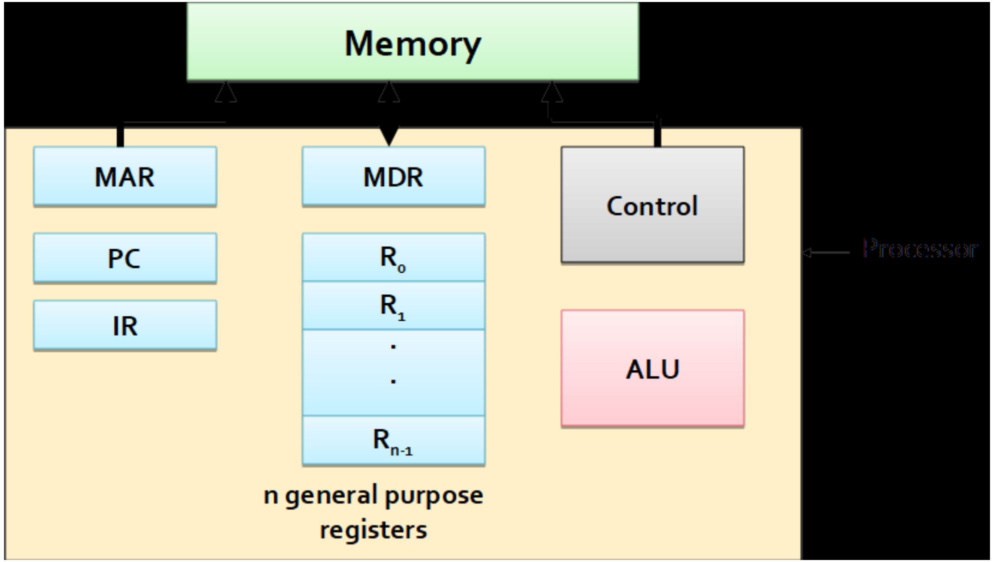
- Write of data word needs two inputs:
  1. Specifies the number of the register to be written
  2. Supplies the data to be written into the register.
- It outputs the contents of Read register inputs (i.e. register number).



Interconnection of Components

- The functional components of a computer need to communicate with each other to perform a task.
- Therefore, computer is a network of components.
- The collection of paths connecting the various components is called the *interconnection structure.*
- The design of this structure will depend on the exchanges that must be made among modules.

Interconnection Architecture



INTERCONNECTION OF COMPONENTS

- Program Counter (PC) is specialized register
  - Keep track of the execution of a program
  - It contains memory address of the next instruction to be fetched and executed
  - During the execution of an instruction, the contents of the PC are updated to correspond to the address of the next instruction to be executed
  - PC points to the next instruction that is to be fetched from memory.

Interconnection of components

- The Instruction Register (IR):
  - Holds the instruction that is currently being executed
  - Its output is available to the control circuits
  - Generates the timing signals that control the various processing elements involved in executing the instruction

PROCESSOR- MEMORY INTERACTION

- Two registers facilitates communication with the memory
- Memory Address Register (MAR)
  - holds the address of the location to be accessed
- Memory Data Register (MDR)
  - Contains the data to be written into or read out of the addressed location

TYPICAL OPERATIONAL STEPS

- Programs reside in the memory
- Execution of the program starts when PC is set to point to the first instruction of the program
- The content of the PC are transferred to MAR
- A read control signal is sent to the memory
- The addressed word is read out of the memory and loaded into the MDR
- Next, the contents of the MDR are transferred to the IR
- At this point the instruction is ready to be decoded and executed

TYPICAL OPERATIONAL STEPS

- It is necessary to obtain the required operands to complete the execution of instruction
- If an operand resides in the memory, it has to be fetched by sending its address to the MAR and initiate read cycle
- Operand is read from the memory to MDR, then it is transferred from MDR to ALU
- After one or more operands are fetched in this way, the ALU can perform the desired operation.
- If the results of this operation is to be stored in the memory, the results is sent to MDR
- The address of the location where the result is to be stored is sent to the MAR, and write cycle is initiated

REFERENCES

1. Carl Hamacher, Zvonko Vranesic, Safwat Zaky, Computer organization, Mc Graw Hill, Fifth edition ,Reprint 2011.
2. David A. Patterson and . John L. Hennessy “Computer Organization and Design-The Hardware/Software Interface” 5th edition, Morgan Kaufmann, 2011.