COMPUTER ORGANIZATION AND ARCHITECTURE

Course Code : CSE 2151

Credits : 04

COURSE OBJECTIVES

This course will enable students to

- Summarize the fundamental concepts of the organization and architecture of a computer.
- Analyze taxonomy of Execution, Processor, Memory and I/O Units.
- Explain the pipelining principles, Data dependencies and hazards, SIMD and Multiprocessor concepts.

COURSE MODULES

1. Basic structure of computers
2. Instruction set architecture
3. Arithmetic and logic unit
4. Control unit
5. Memory systems
6. Input/output organization
7. Introduction to parallel architecture

COURSE OUTCOMES

At the end of this course, the student should be able to:		No. of Hours	Marks
CO1	Describe the functionalities of the various units of computers and the instruction set architecture.	10	20
CO2	Appreciate the hardware implementation of addition, subtraction, multiplication, and division and perform arithmetic operations.	7	16
CO3	Design the control unit for simple algorithms.	10	20
CO4	Explain basics of memory system such as cache memories, mapping functions, replacement algorithms and virtual memory concept and design simple memory systems.	10	20
CO5	Outline the I/O handling techniques and realize the improvement in performance using the concepts of pipelining and parallel processing.	11	24
Total hours/ Marks		48	100

TEXTBOOKS AND REFERENCE BOOKS

Textbooks:

1. Carl Hamacher, ZvonkoVranesic and SafwatZaky, *Computer Organization and Embedded Systems,* (6e), McGraw Hill Publication, 2012.
2. William Stallings, *Computer Organization and Architecture – Designing for Performance,* (9e), PHI, 2015.
3. Mohammed Rafiquzzaman and Rajan Chandra, *Modern Computer Architecture,* Galgotia Publications Pvt. Ltd., 2010.

Reference Books:

1. D.A. Patterson and J. L. Hennessy, Computer Organization and Design-The Hardware/Software Interface, (5e), Morgan Kaufmann, 2014.
2. J. P. Hayes, Computer Architecture and Organization, McGraw Hill Publication, 1998.

INTRODUCTION

- What is Computer Organization and Architecture?

INTRODUCTION

- Computer
- Addition of 2 bits

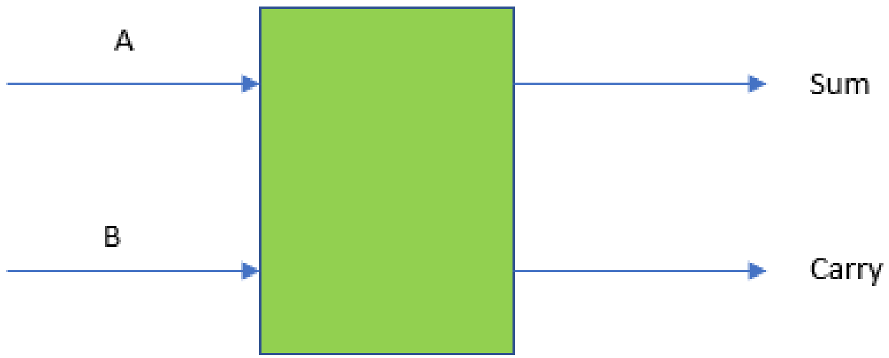


Figure 2: Structure to add two bits

A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

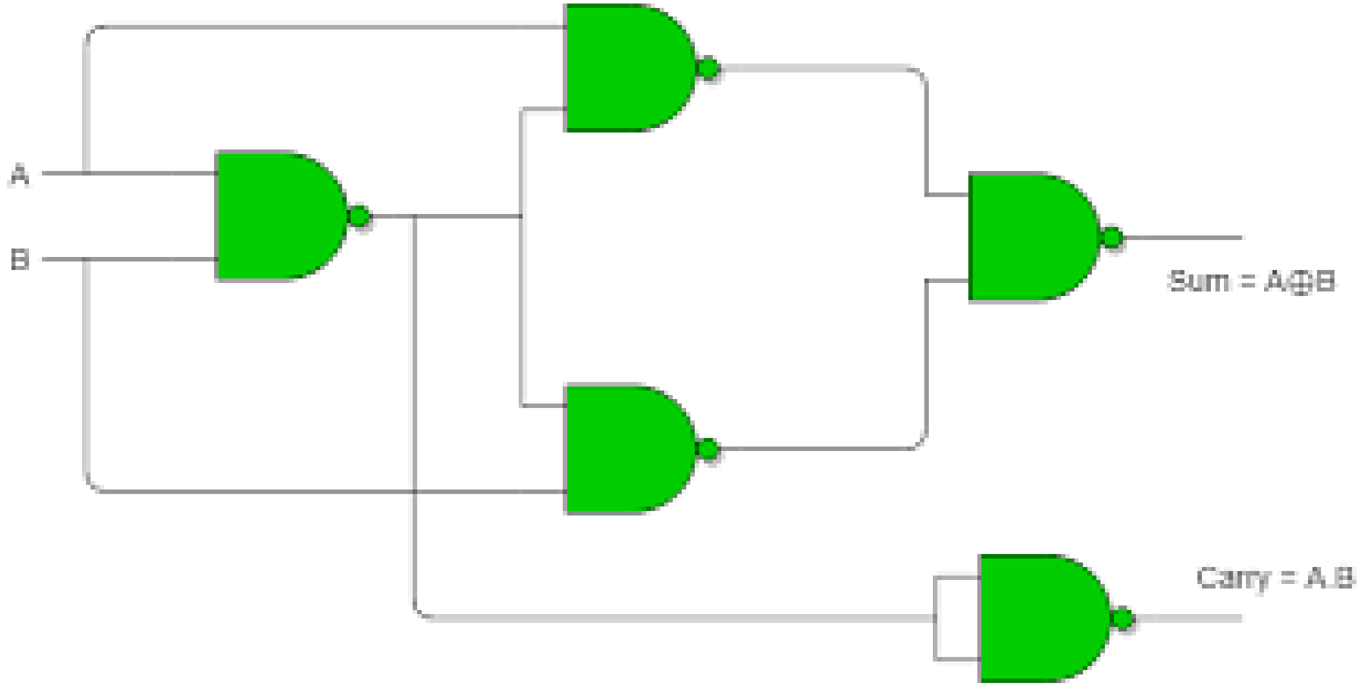


Figure 3: Half Adder using NAND gates.
Source: https://www.geeksforgeeks.org/

INTRODUCTION

- Computer
- Addition of 2 bits

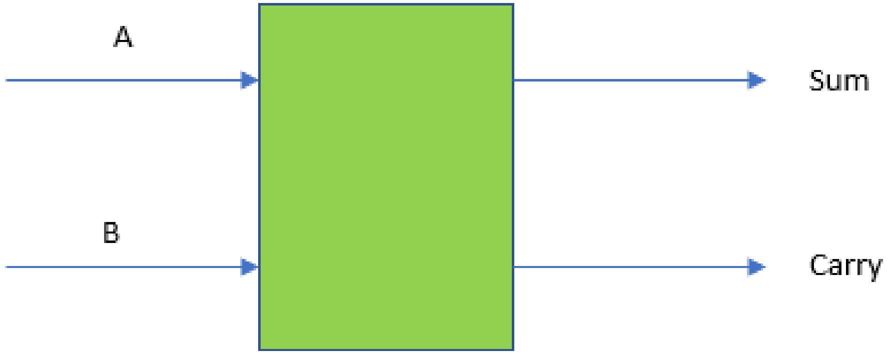


Figure 2: Structure to add two bits

A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

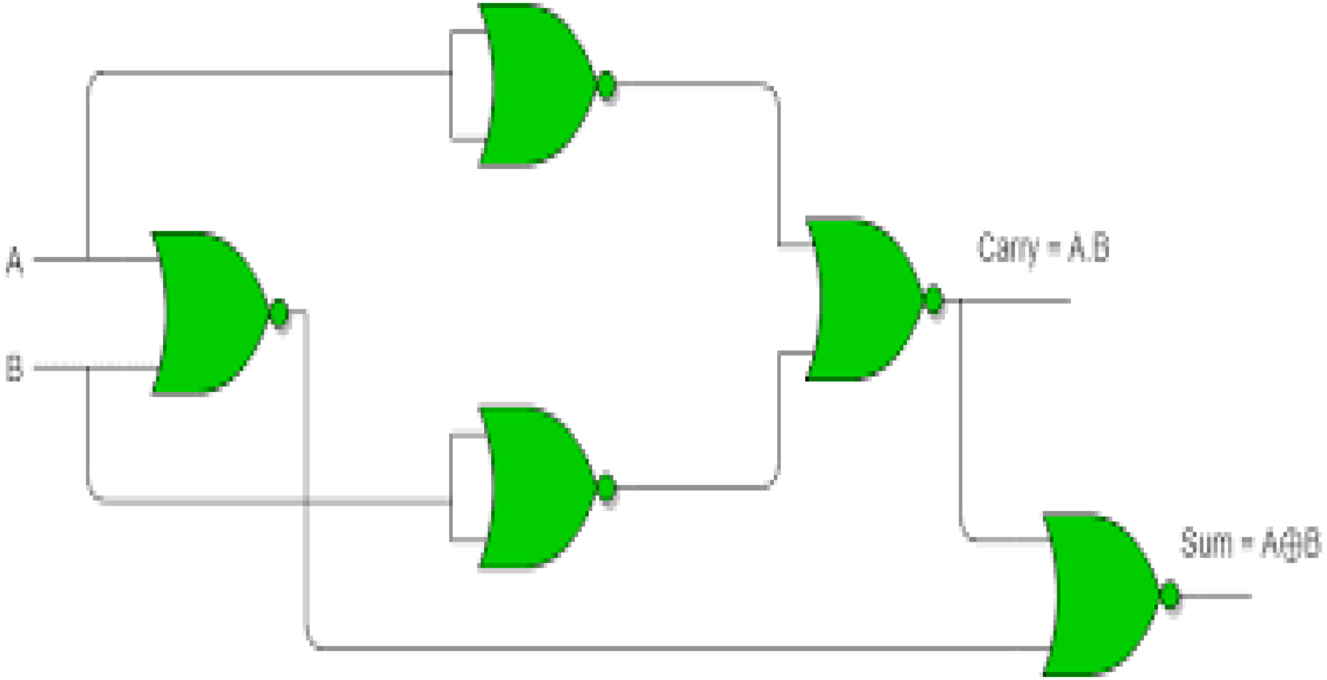


Figure 4: Half Adder using NOR gates.
Source: https://www.geeksforgeeks.org/

INTRODUCTION

- Computer Architecture: refers to those attributes of a system visible to a programmer or those attributes that have a direct impact on the logical execution of a program.
  - Example: instruction sets, the number of bits used to represent various datatypes, I/O mechanisms and techniques for addressing memory.

- Computer Organization: refers to the operational units and their interconnections that realize the architectural specifications.
  - Example: hardware details transparent to the programmer, such as control signals, interfaces between the computer and the peripherals; and the memory technology used.

BASIC STRUCTURE OF COMPUTERS

- The different types of computers
- The basic structure of a computer and its operation
- Machine instructions and their execution
- Number and character representations
- Addition and subtraction of binary numbers

MODULE 1

TYPES OF COMPUTERS

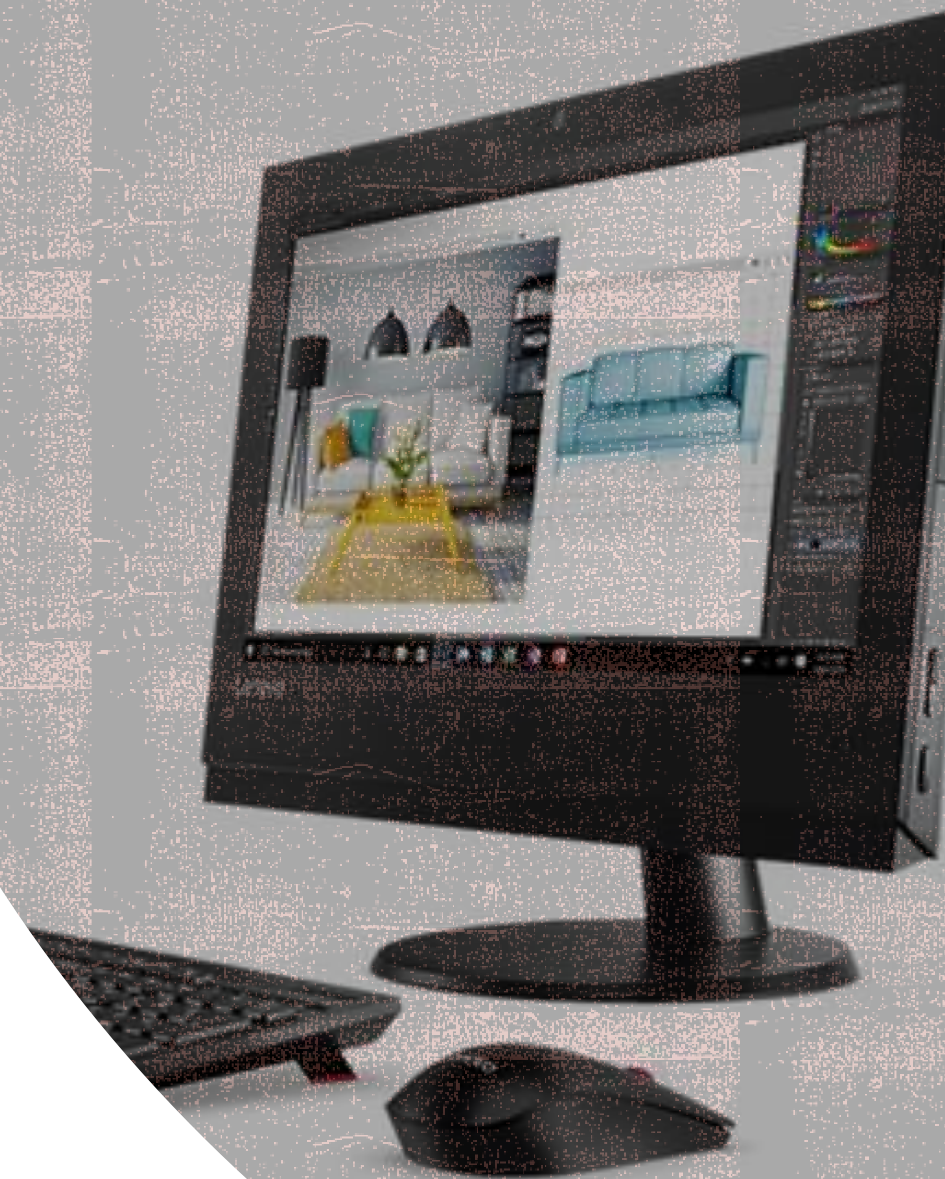
1. Embedded computers
2. Personal computers
3. Servers and Enterprise systems
4. Supercomputers and Grid computers

TYPES OF COMPUTERS

- **Embedded computers**
  - Integrated into a larger device or system in order to automatically monitor and control a physical process or environment.
  - Used for a specific purpose.
  - Applications include industrial and home automation, appliances, telecommunication products, and vehicles

TYPES OF COMPUTERS

- **Personal computers**
  - Primarily for dedicated individual use.
  - Types of personal computers
    1. Desktop computers
    2. Workstation computers
    3. Portable and Notebook computers
  - Applications include general computation, document preparation, computer-aided design, audiovisual entertainment, interpersonal communication, and Internet browsing



TYPES OF COMPUTERS

- **Servers and Enterprise systems**
  - Large computers shared by many users through personal computer over a public or private network.
  - May host large databases and provide information processing for a government agency or a commercial organization.

TYPES OF COMPUTERS

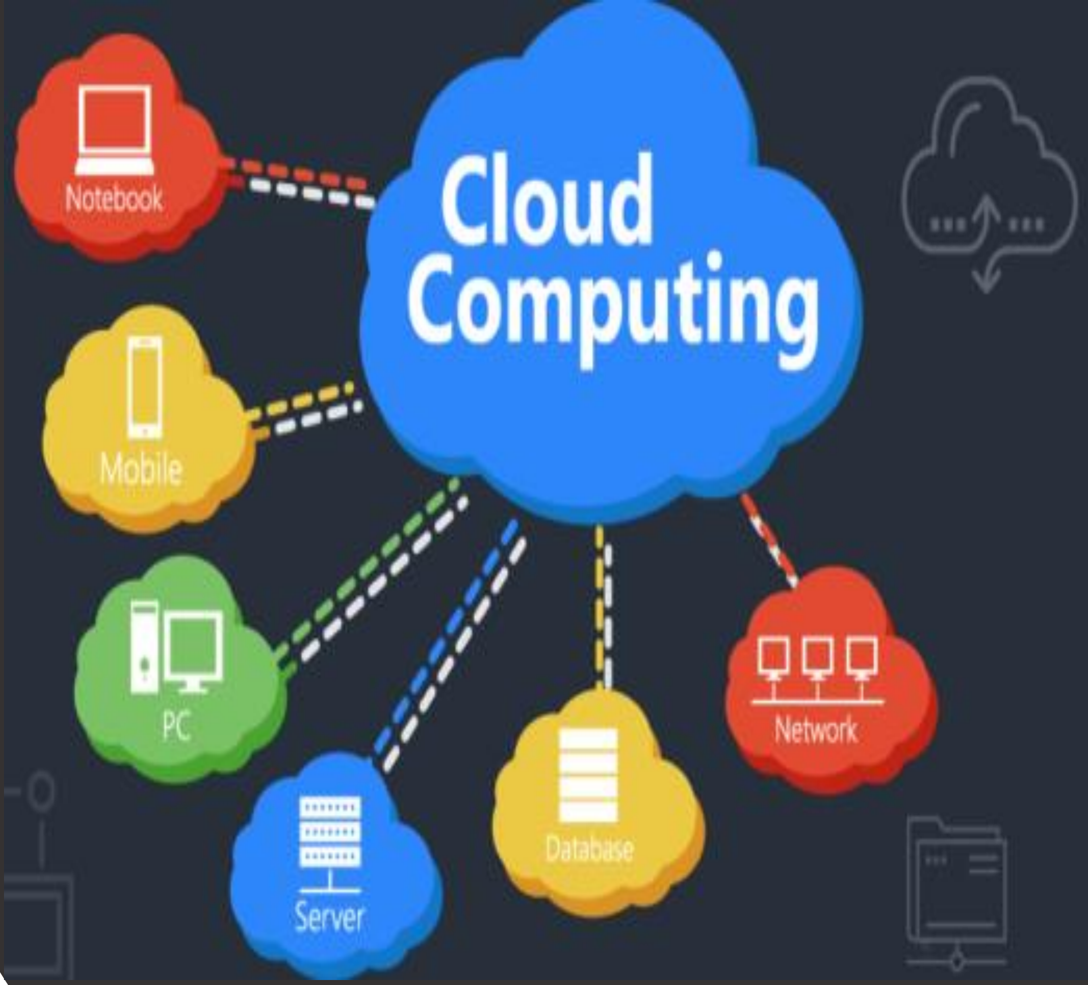
- **Supercomputers and Grid computers**
  - Offer the highest performance
  - Most expensive and physically the largest category of computers.
  - **Supercomputer**
    - High-cost systems.
    - Performance of a supercomputer is measured in floating-point operations per second (FLOPS) instead of million instructions per second (MIPS)
    - Applications include high demanding computation systems such as weather forecasting, engineering design and simulation, and scientific work.

TYPES OF COMPUTERS

- **Supercomputers and Grid computers**
  - **Grid computers**
    - More cost-effective alternative
    - A grid of large number of personal computers and disk storage units are created using a physically distributed high-speed network, which is managed as a coordinated computing resource.
    - The computational workload is distributed across the grid to achieve high performance on large applications ranging from numerical computation to information searching.

TYPES OF COMPUTERS

- **Cloud Computing**
  - Distributed computing and storage server resources for individual, independent, computing need accessed via personal computers.
  - Communication facility via internet.
  - Cloud hardware and software service providers operate as a utility and charge on a pay-as-you-use basis.



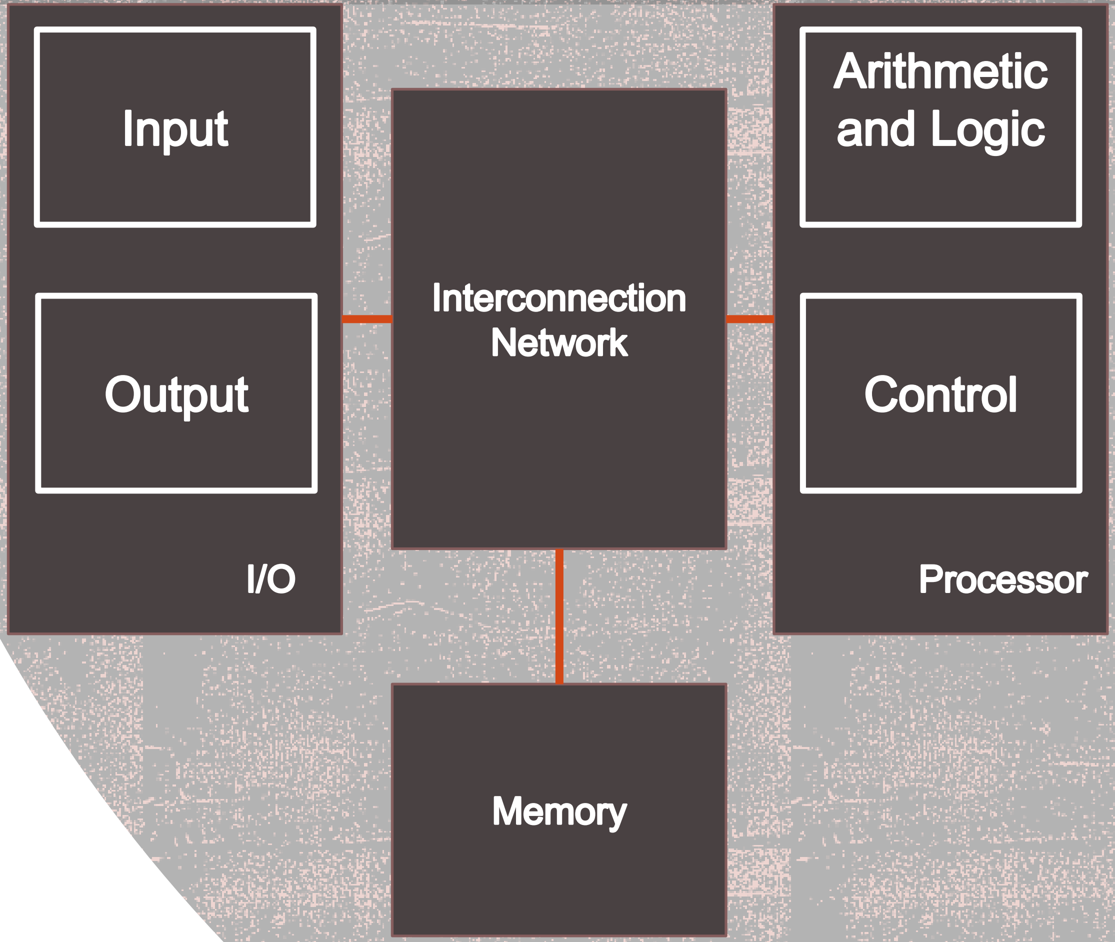
TYPES OF COMPUTERS

1. Embedded computers
2. Personal computers
3. Servers and Enterprise systems
4. Supercomputers and Grid computers

- Cloud Computing

FUNCTIONAL UNITS

- Input
- Output
- Memory
- Arithmetic and Logic Unit
- Control Unit

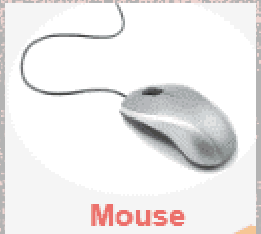


HOW DOES THE COMPUTER HANDLE INFORMATION?

- Instructions are explicit commands that
  - govern the transfer of information within a computer as well as between the computer and its I/O devices
  - specify the arithmetic and logic operations to be performed
- A program is
  - a list of instructions which performs a task.
  - stored in the memory.
- Data are
  - numbers and characters that are used as operands by the instructions.
  - stored in the memory.
- Instructions and data are encoded in binary format (0 and 1)

INPUT UNIT

- **Keyboard**
  - Whenever a key is pressed, the corresponding letter or digit is automatically translated into its corresponding binary code and transmitted to the processor
- **Graphic input**
  - touchpad, mouse, joystick, and trackball.
- **Audio and Video input**
  - The audio input captured by microphones and video input captured by camera are sampled and converted into digital codes for storage and processing.
- **Digital communication facility: Internet**
  - provides input to a computer from other computers and database servers.



OUTPUT UNIT

- send processed results to the outside world.
- Graphic display
- Printers
- Audio output: Speakers, headphones

MEMORY UNIT

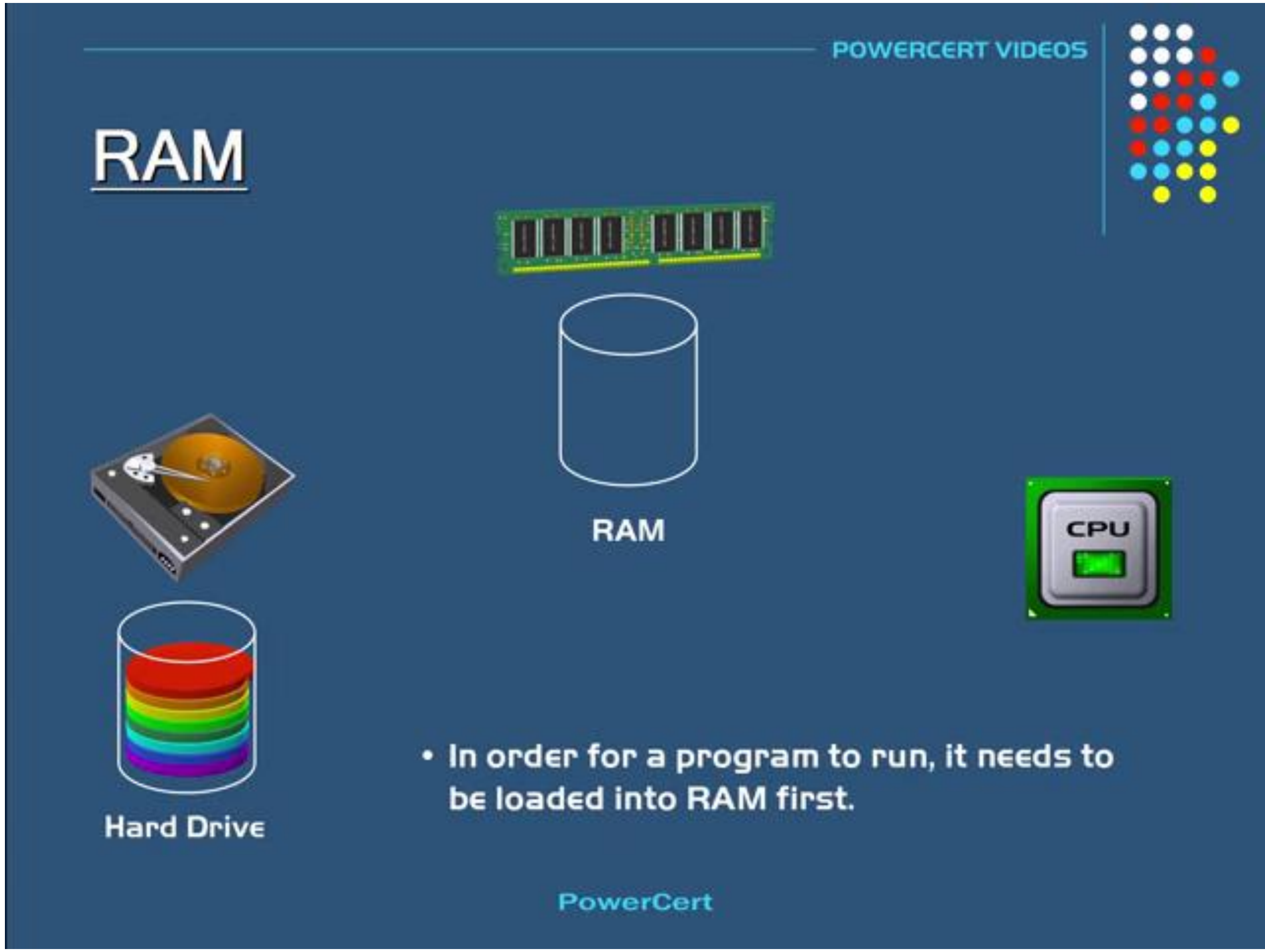
- **Store programs and data**
- **Two classes of storage**
  - Primary memory or main memory
  - Secondary Storage

MEMORY UNIT

- **Primary Memory**
  - Fast memory
  - Programs are stored in this memory while being executed
  - Large number of semiconductor storage cells capable of storing one bit of information
  - **Words:** groups of cells, fixed size.
  - Word length of the computer: number of bits in each word -16, 32, or 64 bits.
  - **Address:** associated with each word location. Consecutive numbers, starting from 0, that identify successive locations.
  - **Random-access memory (RAM):** A memory in which any location can be accessed in a short and fixed amount of time after specifying its address
  - **Memory access time:** The time required to access one word. This time is independent of the location of the word being accessed. Typically ranges from a few nanoseconds (ns) to about 100 ns.

MEMORY UNIT

- **Primary Memory**
  - All programs stored in the secondary memory must be loaded from secondary memory to primary memory (RAM) before its execution.
  - The data is then transferred to the cache, a smaller, faster RAM unit, for fast retrieval of instructions to the CPU



- In order for a program to run, it needs to be loaded into RAM first.

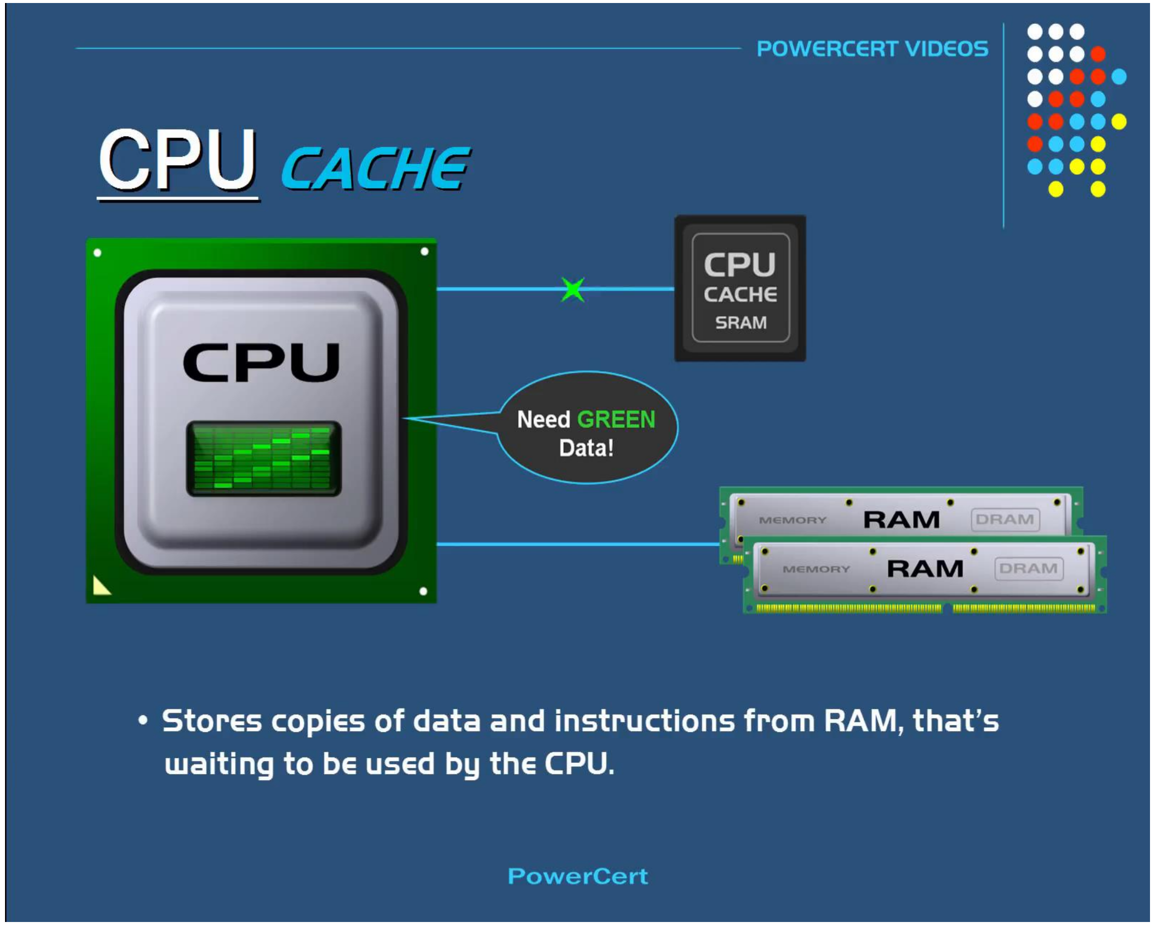
Source: https://gfycat.com/contentmeanacornbarnacle

MEMORY UNIT

- **Cache Memory**
  - Used to hold sections of a program that are currently being executed, along with any associated data
  - is tightly coupled with the processor and is usually contained on the same integrated-circuit chip
  - facilitates high instruction execution rates

MEMORY UNIT

- **Cache Memory**
  - At the start of program execution, the cache is empty.
  - As execution proceeds, instructions are fetched into the processor chip, and a copy of each is placed in the cache.
  - Suppose several instructions are executed repeatedly as happens in a program loop and these instructions are available in the cache, they can be fetched quickly during the period of repeated use.
  - Similarly, if the same data locations are accessed repeatedly while copies of their contents are available in the cache, they can be fetched quickly.



- Stores copies of data and instructions from RAM, that's waiting to be used by the CPU.

Source: https://gfycat.com/fluidcheapkitty

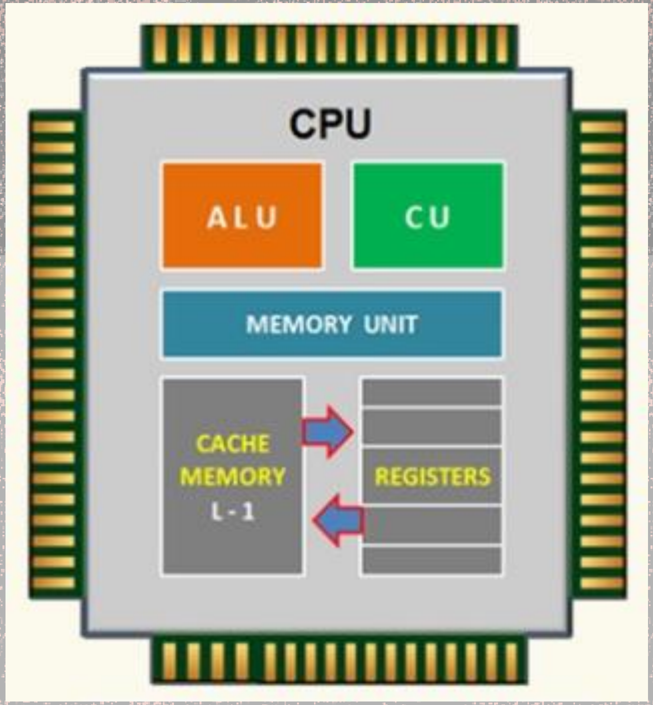
MEMORY UNIT

- **Secondary Memory**
  - used when large amounts of data and many programs must be stored, particularly for information that is accessed infrequently.
  - Access times for secondary storage are longer than for primary memory.
  - magnetic disks, optical disks (DVD and CD), and flash memory devices.

- **Memory Hierarchy**
  - Secondary > Primary > Cache (storage)
  - Secondary < Primary < Cache (speed and cost)

ARITHMETIC & LOGIC UNIT

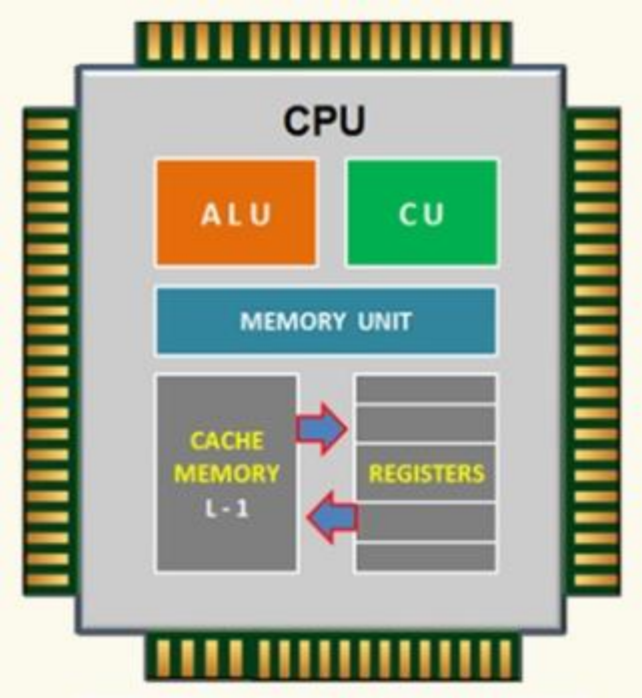
- Most computer operations are executed in ALU of the processor.
- Load the operands into memory – bring them to the processor – perform operation in ALU – store the result back to memory or retain in the processor.
- Registers-high-speed storage elements-store one word of data
- Fast control of ALU



CONTROL UNIT

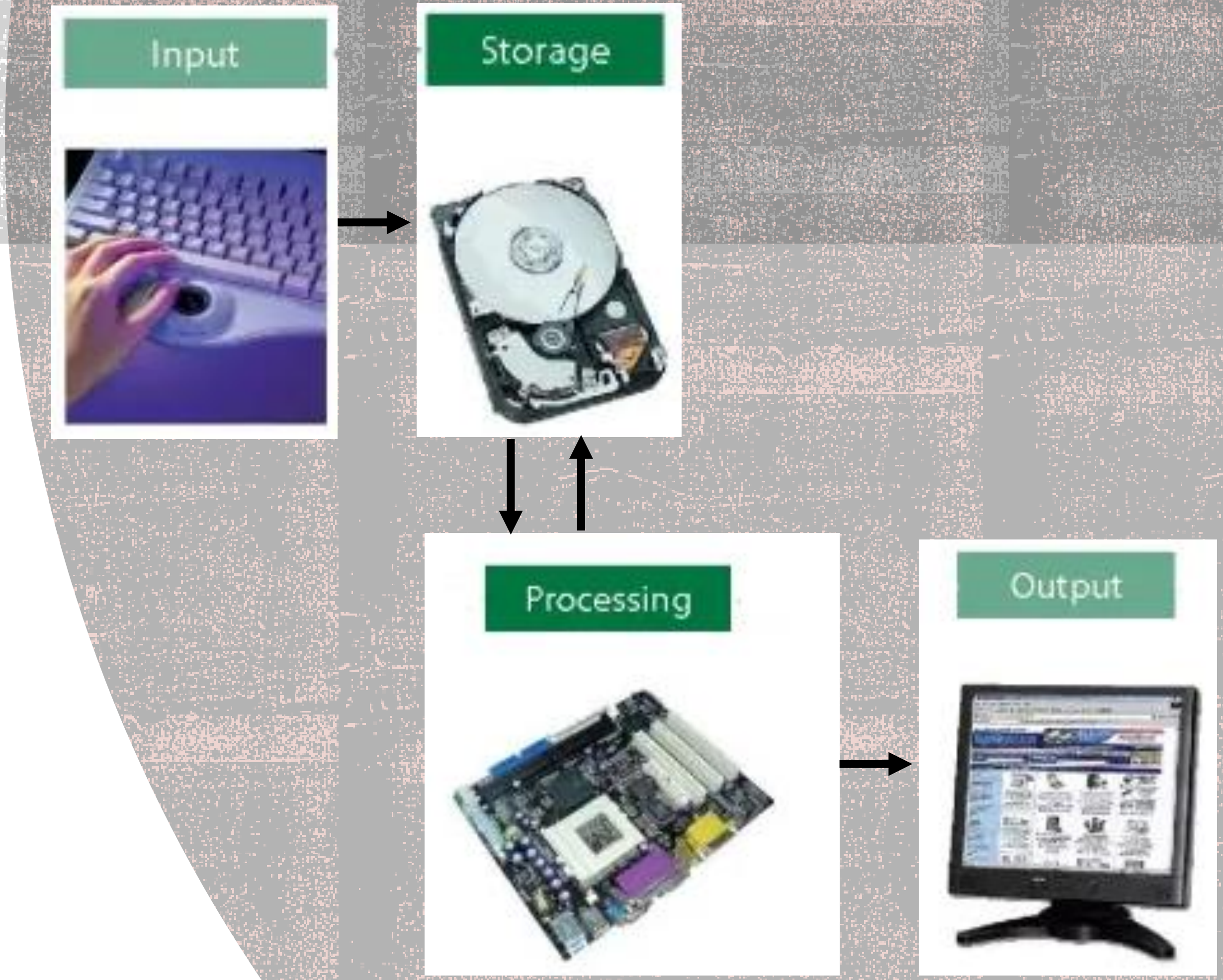
- All computer operations are controlled by the control unit.
- The timing signals that govern the I/O transfers are also generated by the control unit.
- Control unit is usually distributed throughout the machine instead of standing alone.

- What is inside CPU?
  - https://youtu.be/NKYgZH7SBjk



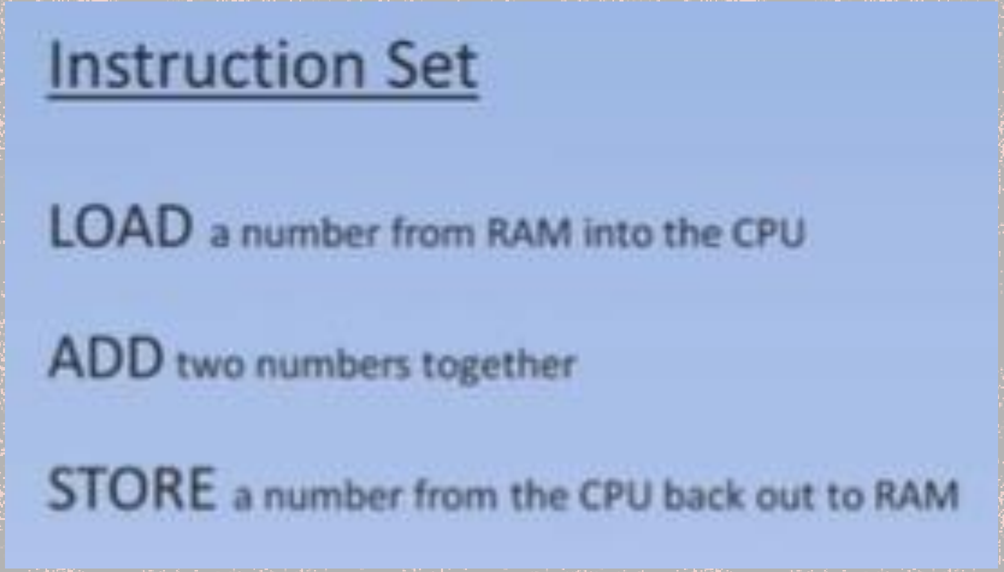
OPERATIONS OF A COMPUTER

- Accept information in the form of programs and data through an input unit and store it in the memory
- Fetch the information stored in the memory, under program control, into an ALU, where it is then processed
- Output the processed information through an output unit
- All activities inside the machine are controlled through a control unit



BASIC OPERATIONAL CONCEPTS

- A typical instruction set:
  - Load R2, LOC
  - Add R4, R2, R3
  - Store R4, LOC
- Load R2, LOC
  - reads the contents of a memory location with address represented as LOC and loads them into processor register R2
- Add R4, R2, R3
  - adds the contents of registers R2 and R3 and stores the result in R4
- Store R4, LOC
  - copies the operand in register R4 to memory location LOC



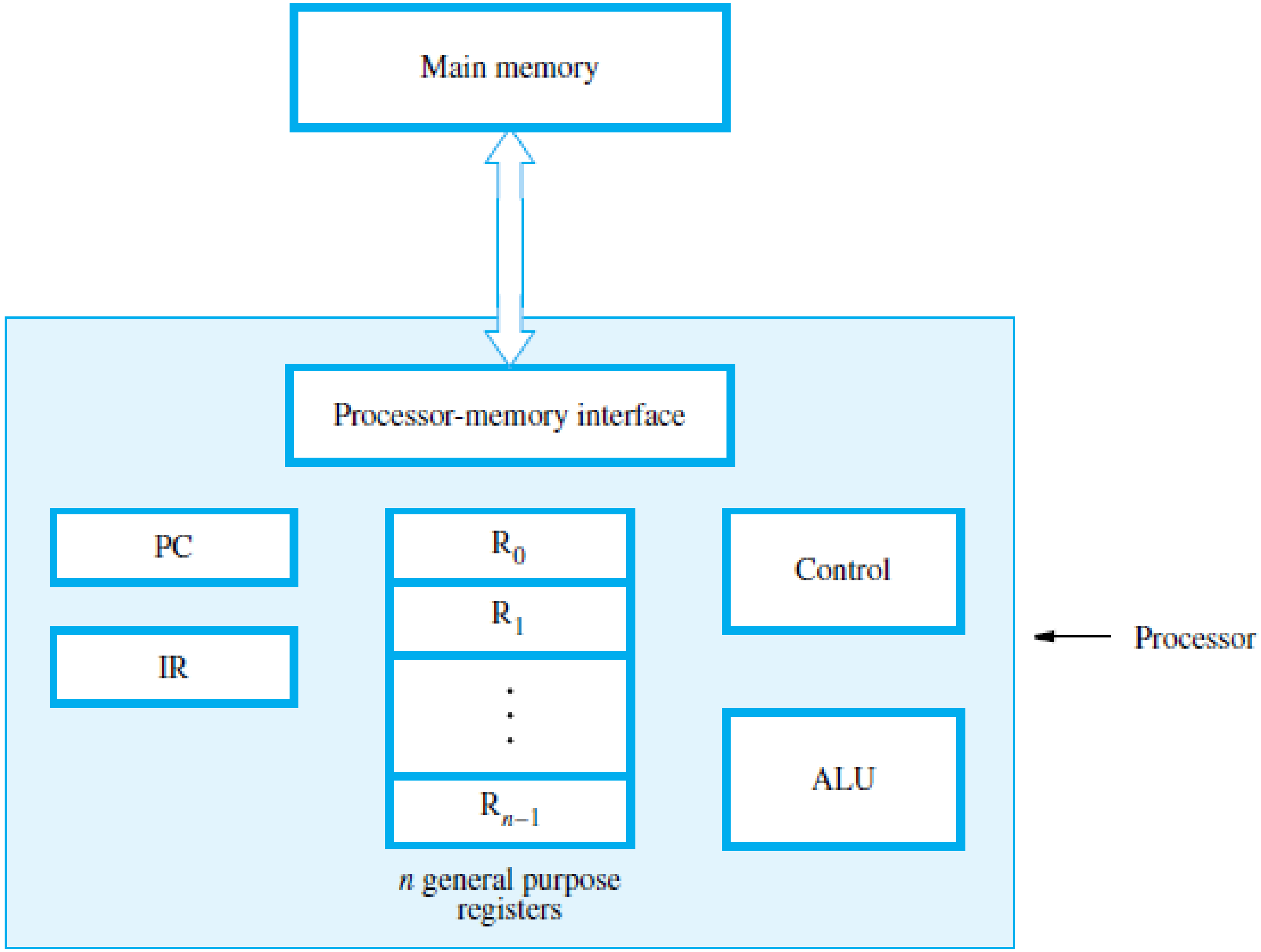
Instruction Set

LOAD a number from RAM into the CPU

ADD two numbers together

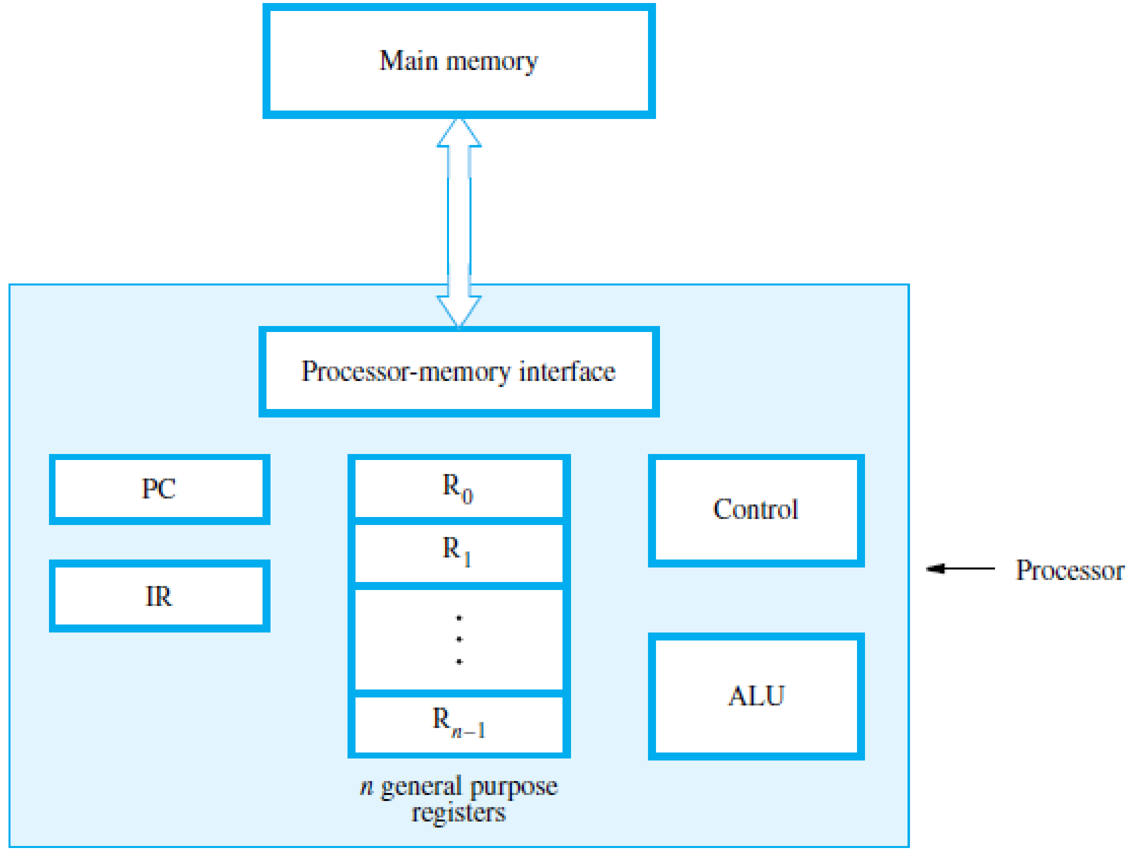
STORE a number from the CPU back out to RAM

CONNECTION BETWEEN THE PROCESSOR AND THE MEMORY



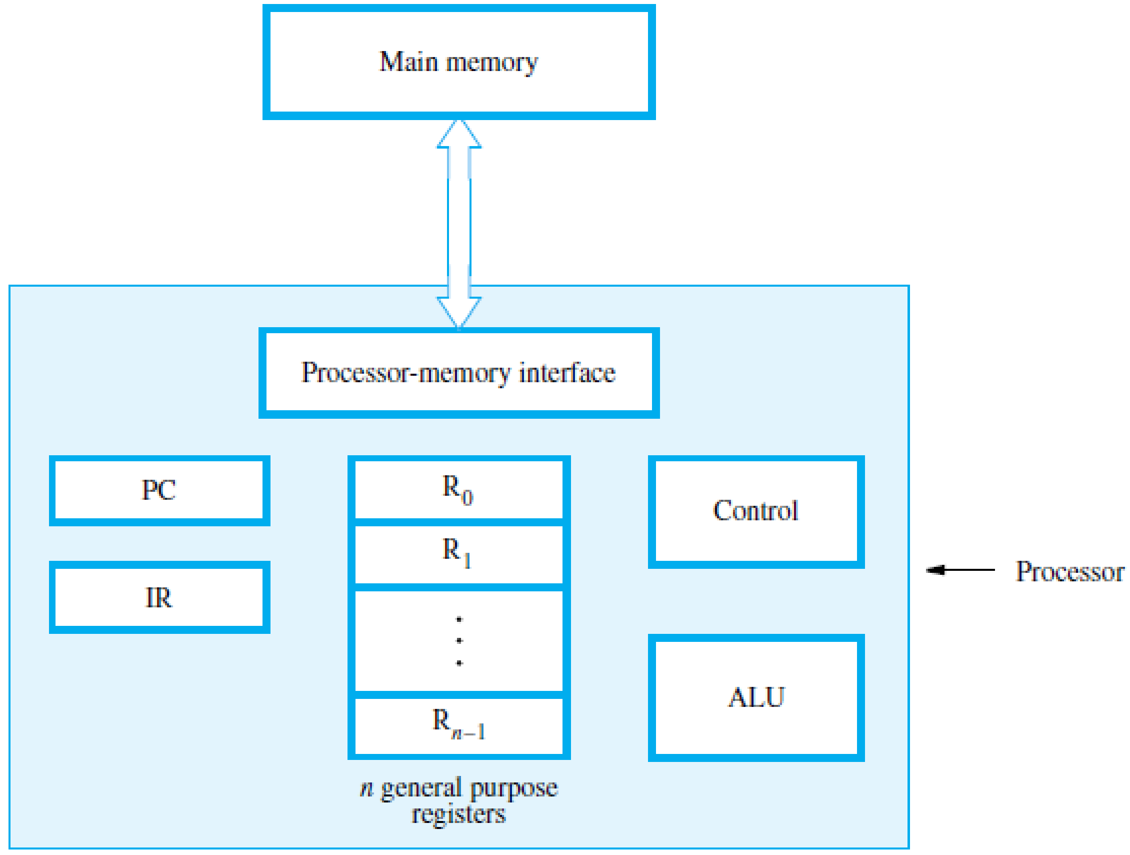
REGISTERS

- Instruction register (IR)
  - holds the instruction that is currently being executed. Its output is available to the control circuits, which generate the timing signals that control the various processing elements involved in executing the instruction
- Program counter (PC)
  - points to the next instruction that is to be fetched from the memory
- General-purpose register
  - $R_0 – R_{n-1}$
  - Holds operands that have been loaded from the memory for processing



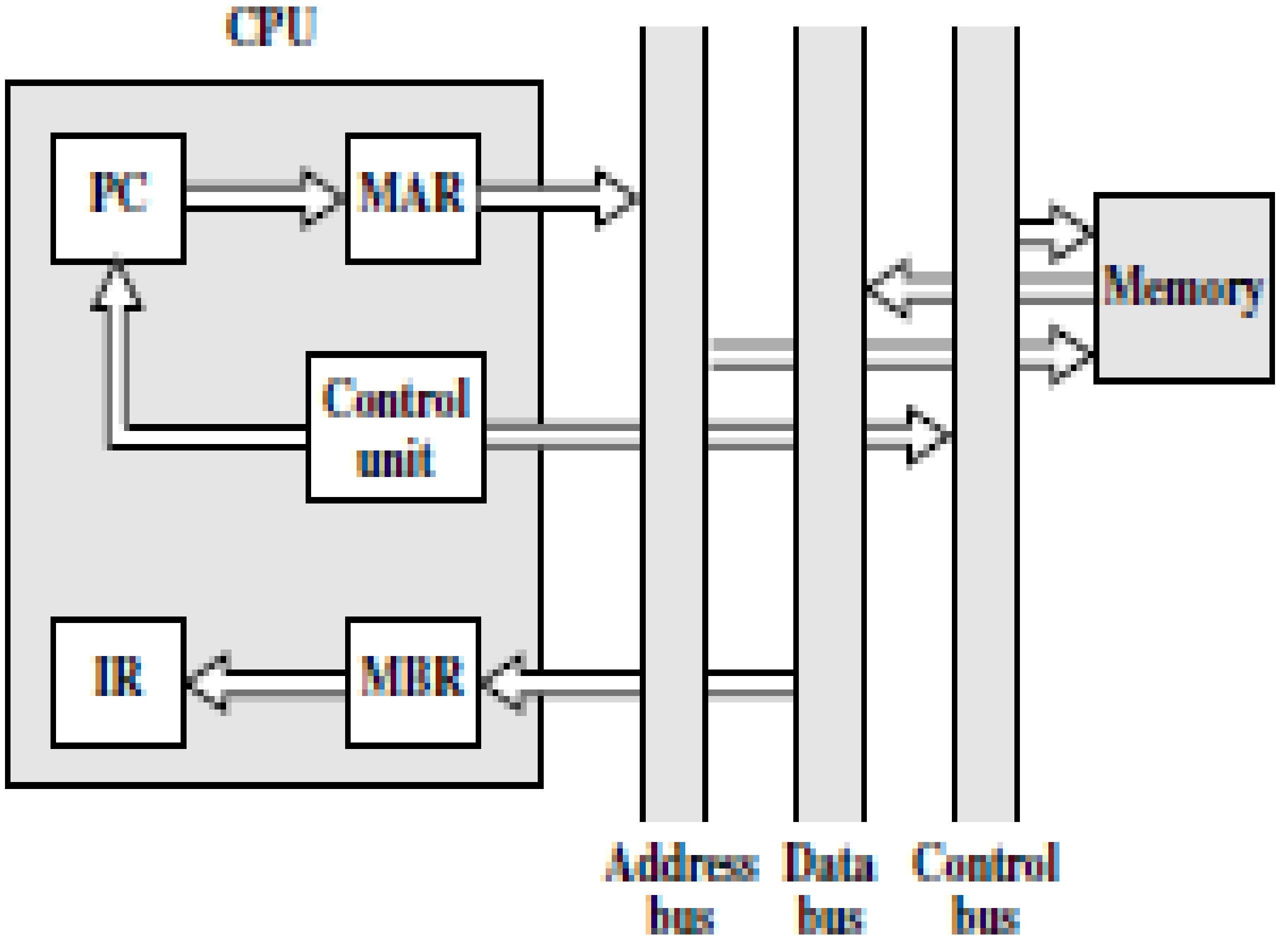
PROCESSOR-MEMORY INTERFACE

- It is a circuit which manages the transfer of data between the main memory and the processor.
- If a word is to be read from the memory, the interface sends the address of that word to the memory along with a Read control signal.
- The interface waits for the word to be retrieved, then transfers it to the appropriate processor register.
- If a word is to be written into memory, the interface transfers both the address and the word to the memory along with a Write control signal.
- How CPU works?
  - https://youtu.be/cNN_tTXABUA



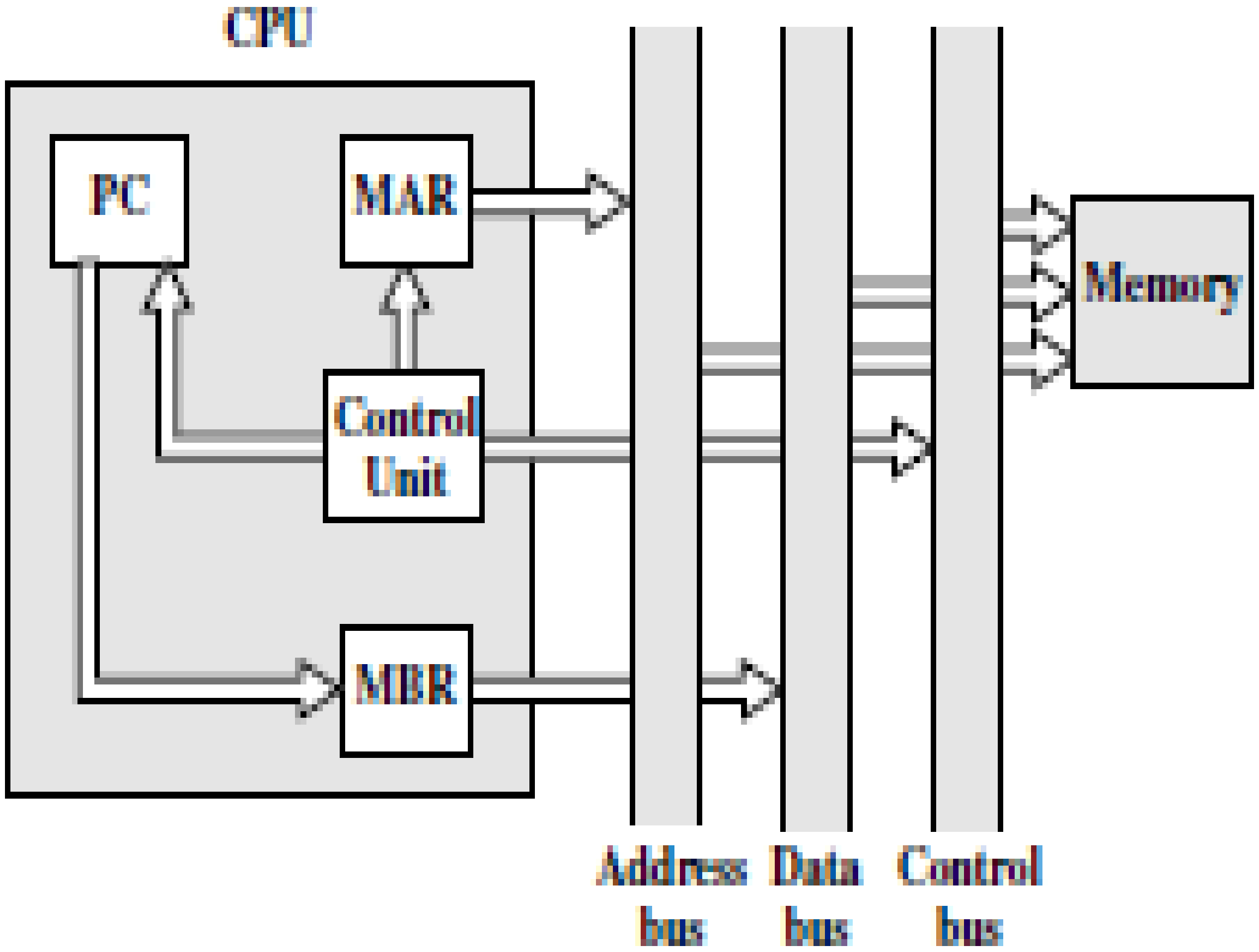
TYPICAL OPERATING STEPS

- Programs to be executed must be in main memory
- PC is set to point to the first instruction
- The contents of PC are transferred to memory along with a Read control signal
- The first instruction read out is transferred to IR
- The instruction is ready to be decoded and executed
- Get operands for ALU
  - Memory
  - General-purpose register
- Perform operation in ALU
- Store the result back
  - To general-purpose register
  - To memory
- During the execution, PC is incremented to the next instruction



INTERRUPTS

- Normal execution of programs may be preempted if some device requires urgent servicing.
- The normal execution of the current program must be interrupted – the device raises an interrupt signal.
- Interrupt-service routine
- Current system information backup and restore (PC, general-purpose registers, control information, specific information)



EXERCISE

- List the steps needed to execute the machine instruction

  Load R2, LOC

- **Solution:**
  - Send the address of the instruction word from register PC to the memory and issue a Read control command.
  - Wait until the requested word has been retrieved from the memory
  - load it into register IR, where it is interpreted (decoded) by the control circuitry to determine the operation to be performed.
  - Increment the contents of register PC to point to the next instruction in memory.
  - Send the address value LOC from the instruction in register IR to the memory and issue a Read control command.
  - Wait until the requested word has been retrieved from the memory, then load it into register R2

EXERCISE

- Repeat for
    1. Add R4, R2, R3
    2. Store R4, LOC

EXERCISE

- Give a short sequence of machine instructions for the task
  - Add the contents of memory location A to those of location B, and place the answer in location C
  - Following instructions are the only instructions available to transfer data between the memory and the general-purpose registers.
    - Load Ri, LOC
    - Store Ri, LOC
  - Do not change the contents of either location A or B.

TOPICS COVERED FROM

- Textbook 1:
  - Chapter 1: 1.1, 1.2, 1.3