

# Computer Organization and Architecture

---

By  
Smita Mande



# Computer Architecture

---

- **Refers to those attributes of a system ,visible to a programmer**
  - Example
    - Instruction set
    - No of bits used to represent diff data types(numbers, characters)
    - I/O mechanism
    - Techniques used for addressing memory

# Computer Organization

---

- **Refers to operational units and their interconnections**
  - **Example**
    - Hardware details transparent to programmer
    - Interface between computer and peripherals(devices connected to computer)
    - Memory technology used

# Difference between Computer Architecture and Computer Organization

SR. No.	Computer Architecture	Computer Organization
1.	Architecture describes <b>what the computer does.</b>	The Organization describes <b>how it does it.</b>
2.	Computer Architecture deals with the functional behavior of computer systems.	Computer Organization deals with a structural relationship.
3.	As a programmer, you can view architecture as a series of instructions, addressing modes, and registers.	The implementation of the architecture is called organization.
4.	For designing a computer, its architecture is fixed first.	For designing a computer, an organization is decided after its architecture.

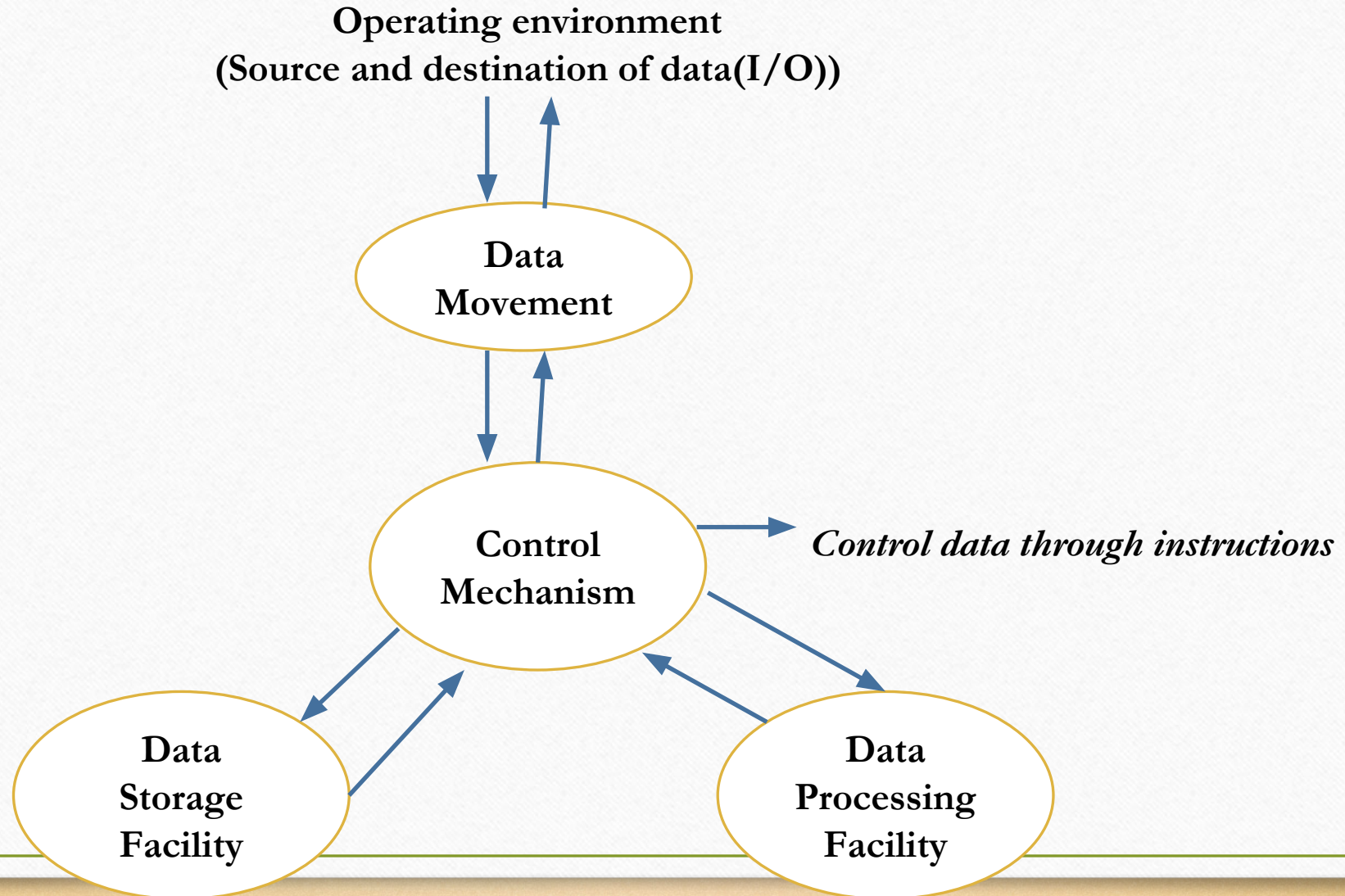


# Structure and Function

---

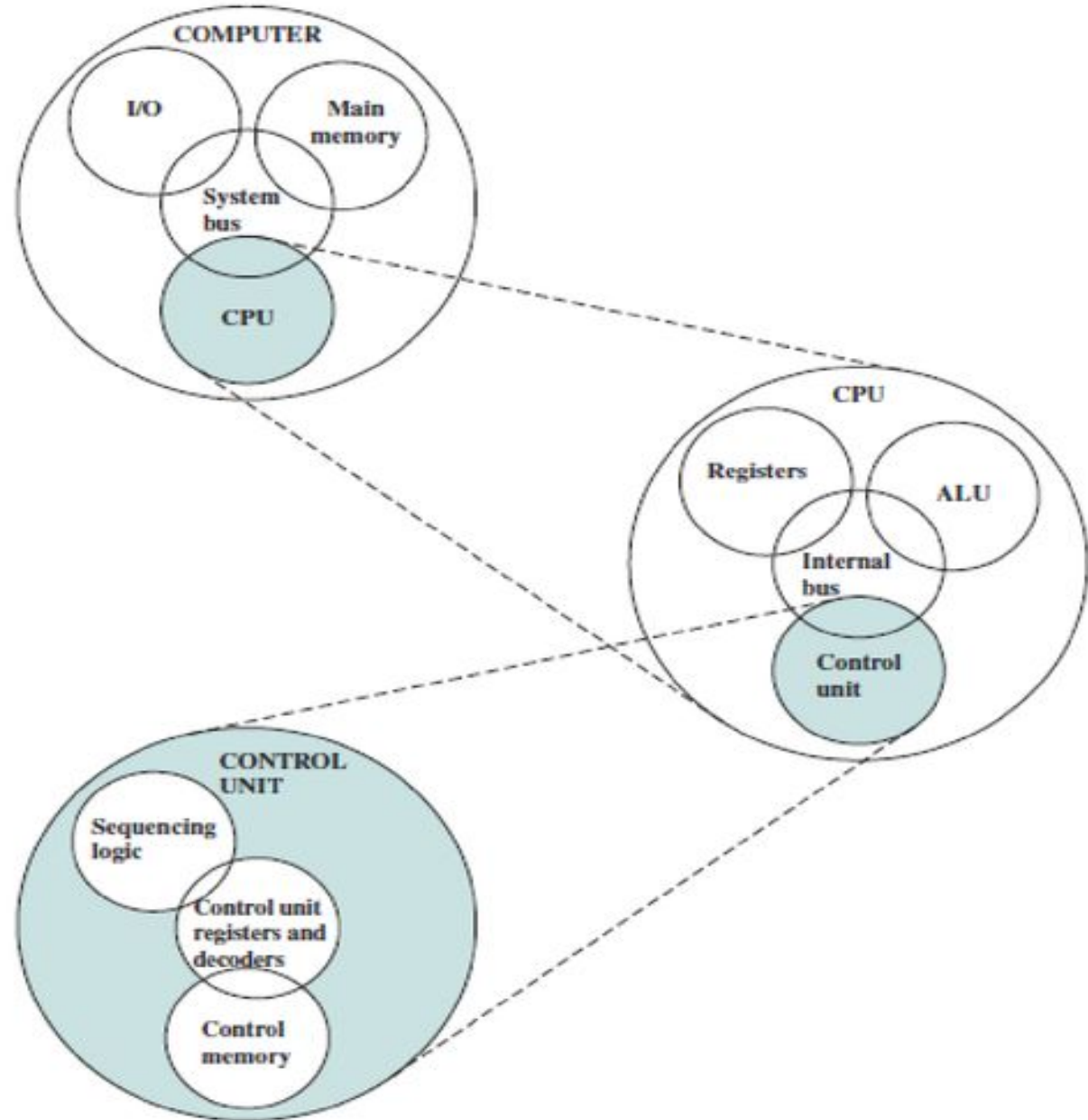
- **Structure:** The way in which the components are interrelated
- **Functions:** The operation of each individual components as part of the structure
  - Top down approach
  - Bottom up approach

# Basic functions that computer can perform





# The Computer: Top-Level Structure



# Structural Components

---

- ❖ **CPU(Processor):** Controls the operation of computer and performs its data processing functions
- ❖ **Main Memory:** Stores data
- ❖ **I/O:** Moves data between computer and its external environment
- ❖ **System interconnections:** mechanism provided for communication among CPU, main memory and I/O by means of **system bus**, which consist of number of connecting wires to which all other components attach

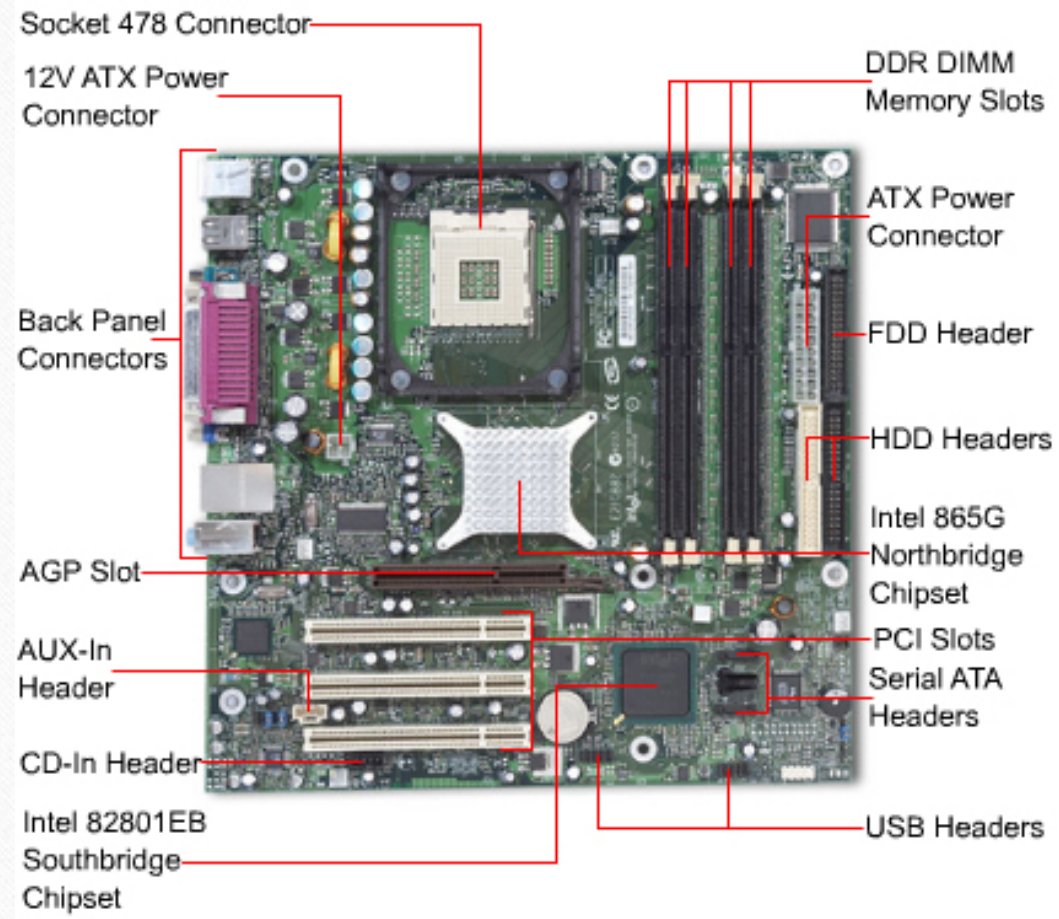


# Components of CPU

---

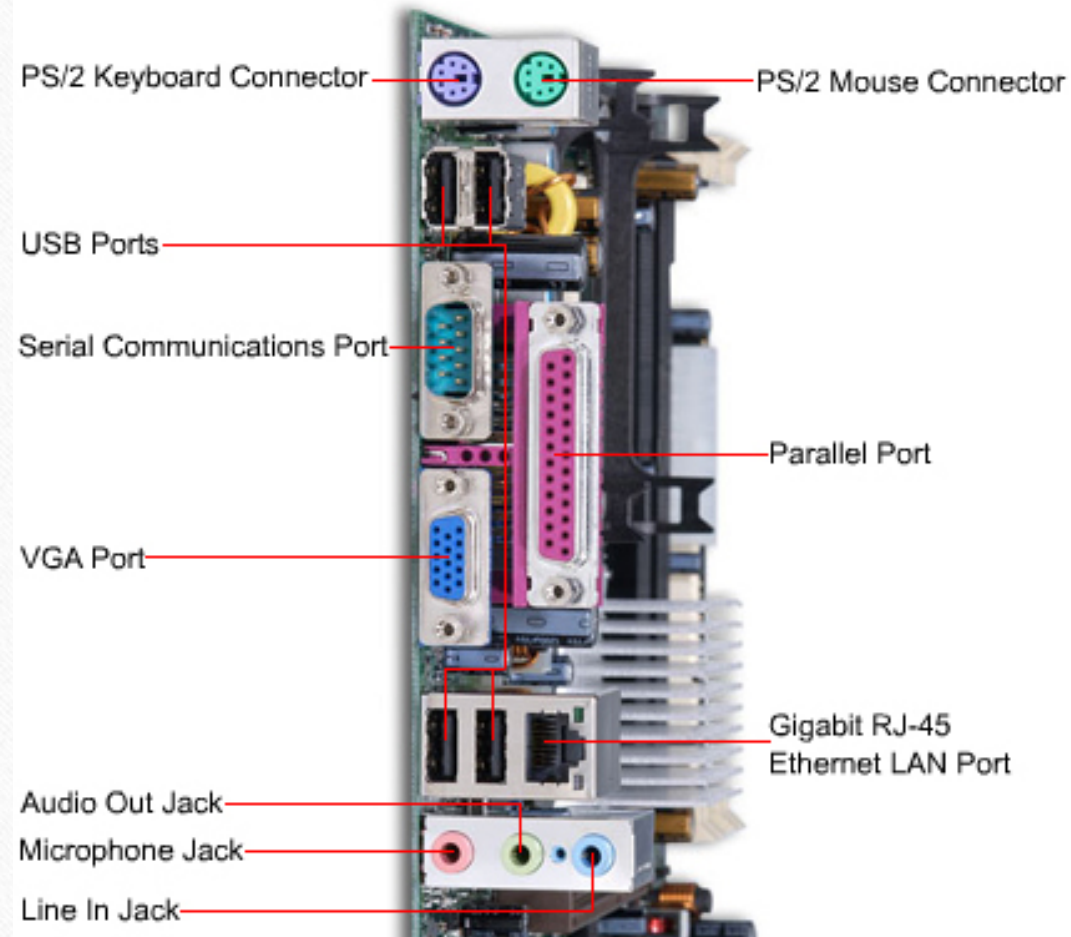
- ❖ **Control unit:** Controls the operation of the CPU
- ❖ **Arithmetic and logic unit(ALU):** performs computer's data processing functions
- ❖ **Registers:** Provides storage internals to the CPU
- ❖ **CPU interconnection:** mechanism provided for communication among ALU, CU and registers

# Hardware

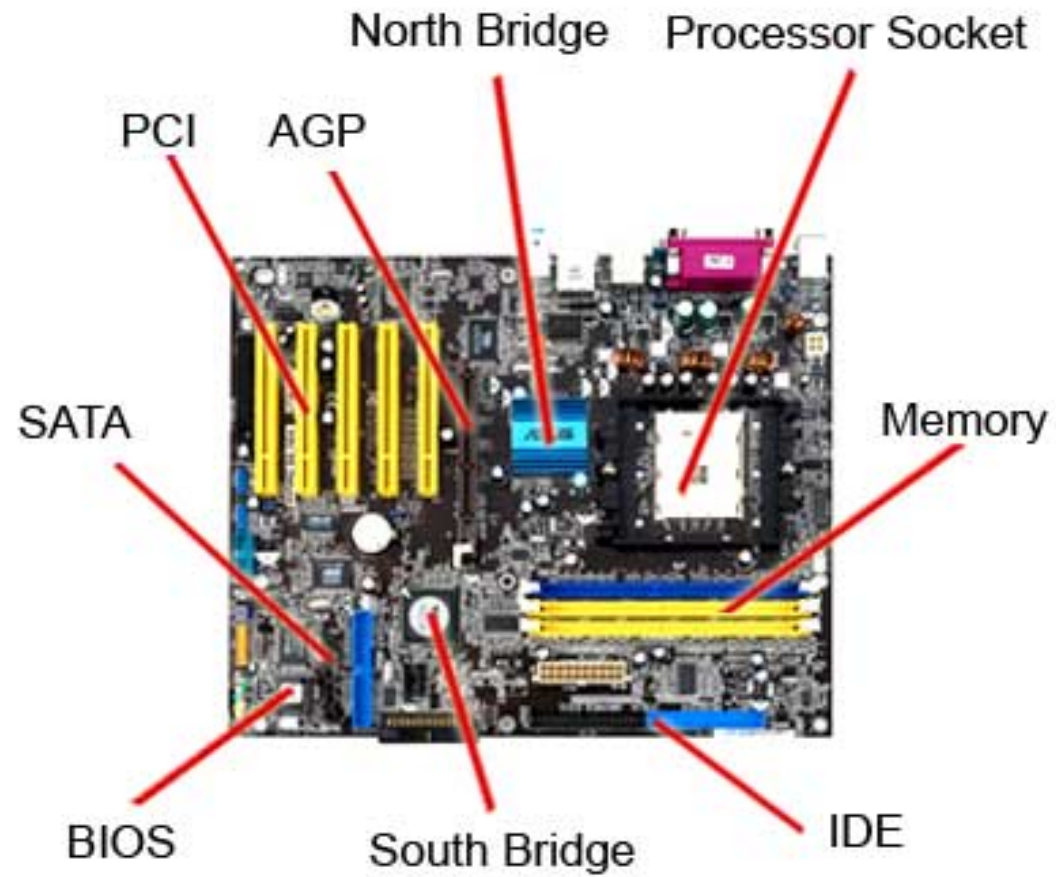




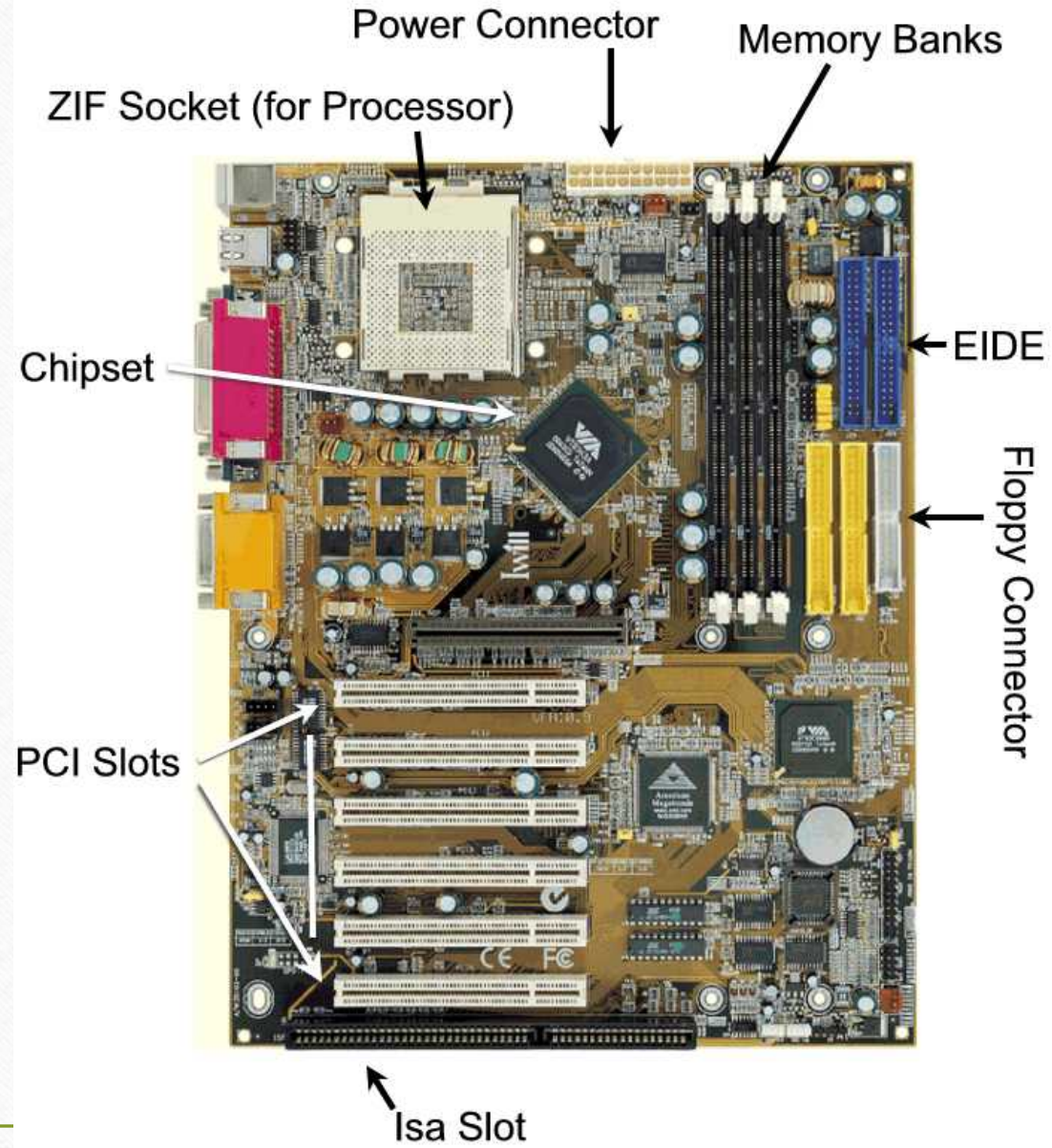
# Motherboard



# Motherboard

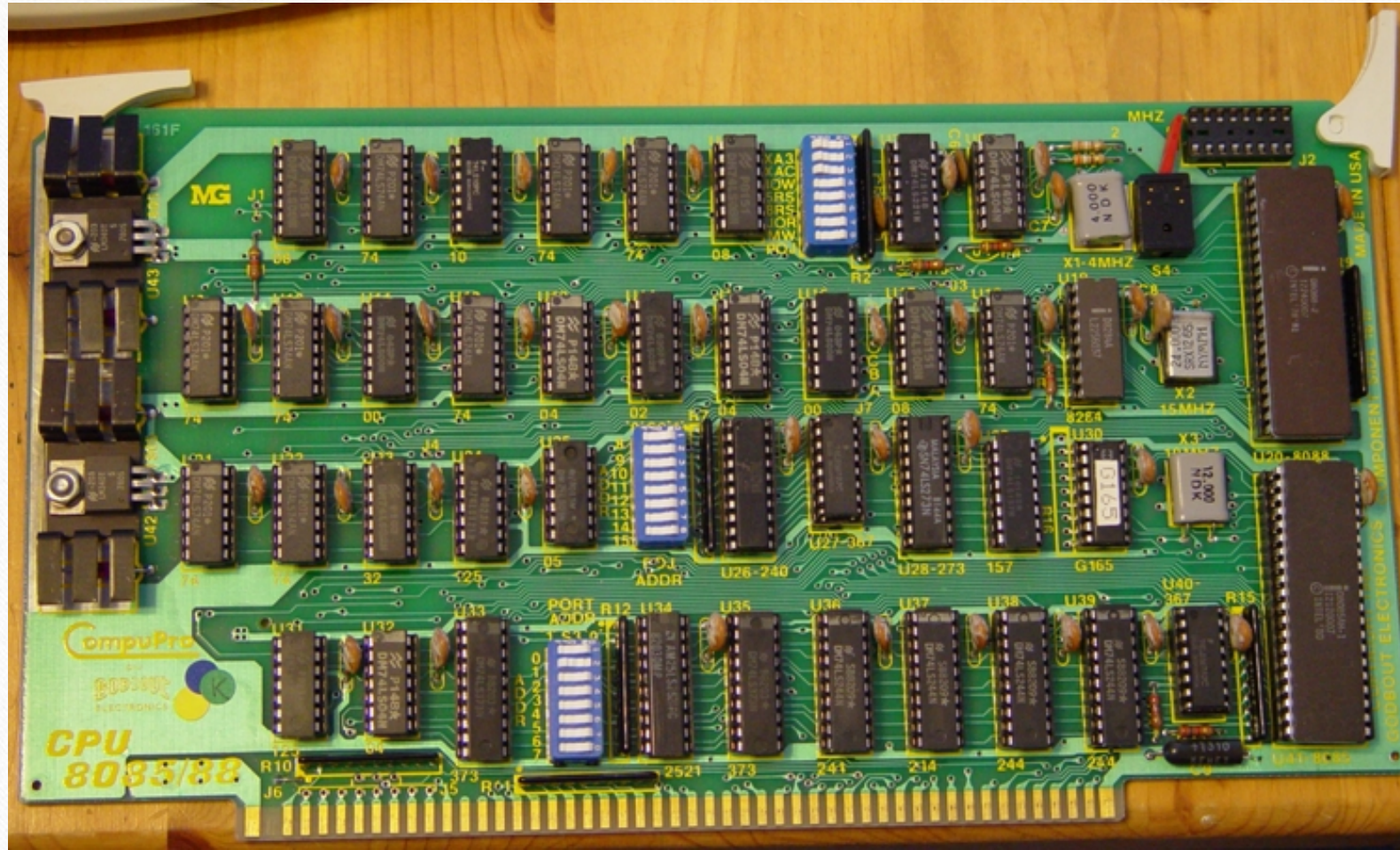








# Previous Motherboard





# Input Devices



# Input Devices





# Memory Devices



# Output Devices





# Evolution of computers

---

- **First Generation Computers (1940-1956)**
- **Second Generation Computers (1956-1963)**
- **Third Generation Computers (1964-1971)**
- **Fourth Generation Computers (1971-Present)**
- **Fifth Generation Computers (Present and Beyond)**

# First Generation Computers (1940-1956)

---

<b>Main electronic component</b>	Vacuum tube.
<b>Programming language</b>	Machine language.
<b>Main memory</b>	Magnetic tapes and magnetic drums.
<b>Input/output devices</b>	Paper tape and punched cards.
<b>Speed and size</b>	Very slow and very large in size (often taking up entire room).
<b>Examples of the first generation</b>	IBM 650, IBM 701, ENIAC, UNIVAC1, etc.



# Second Generation Computers (1956-1963)

<b>Main electronic component</b>	Transistor.
<b>Programming language</b>	Machine language and assembly language.
<b>Memory</b>	Magnetic core and magnetic tape/disk.
<b>Input/output devices</b>	Magnetic tape and punched cards.
<b>Power and size</b>	Smaller in size, low power consumption, and generated less heat (in comparison with the first generation computers).
<b>Examples of second generation</b>	PDP-8, IBM1400 series, IBM 7090 and 7094, UNIVAC 1107, CDC 3600 etc.

# Third Generation Computers (1964-1971)

<b>Main electronic component</b>	Integrated circuits (ICs)
<b>Programming language</b>	High-level language
<b>Memory</b>	Large magnetic core, magnetic tape/disk
<b>Input / output devices</b>	Magnetic tape, monitor, keyboard, printer, etc.
<b>Examples of third generation</b>	IBM 360, IBM 370, PDP-11, NCR 395, B6500, UNIVAC 1108, etc.



# Fourth Generation Computers (1971-Present)

---

<b>Main electronic component</b>	Very large-scale integration (VLSI) and the microprocessor (VLSI has thousands of transistors on a single microchip).
<b>Memory</b>	semiconductor memory (such as RAM, ROM, etc.)
<b>Input/output devices</b>	pointing devices, optical scanning, keyboard, monitor, printer, etc.
<b>Examples of fourth generation</b>	IBM PC, STAR 1000, APPLE II, Apple Macintosh, Alter 8800, etc.

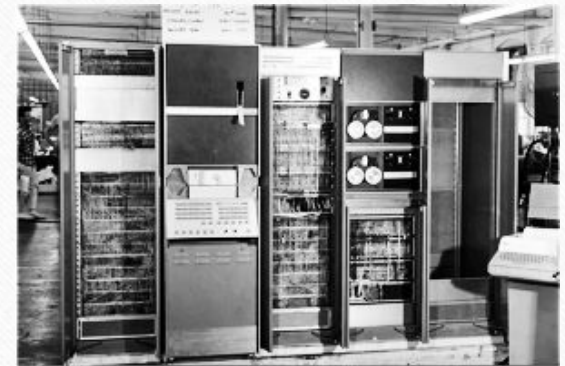
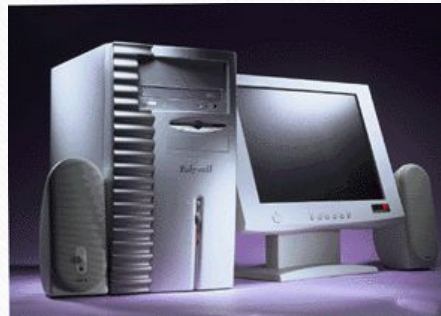
# Fifth Generation Computers (Present and Beyond)

<b>Main electronic component</b>	Based on artificial intelligence, uses the Ultra Large-Scale Integration (ULSI) technology and parallel processing method (ULSI has millions of transistors on a single microchip and Parallel processing method use two or more microprocessors to run tasks simultaneously).
<b>Language</b>	Understand natural language (human language).
<b>Size</b>	Portable and small in size.
<b>Input / output device</b>	Trackpad (or touchpad), touchscreen, pen, speech input (recognize voice/speech), light scanner, printer, keyboard, monitor, mouse, etc.
<b>Example of fifth generation</b>	Desktops, laptops, tablets, smartphones, etc.



# Designing Computers

- All computers more or less based on the same basic design, the Von Neumann Architecture!



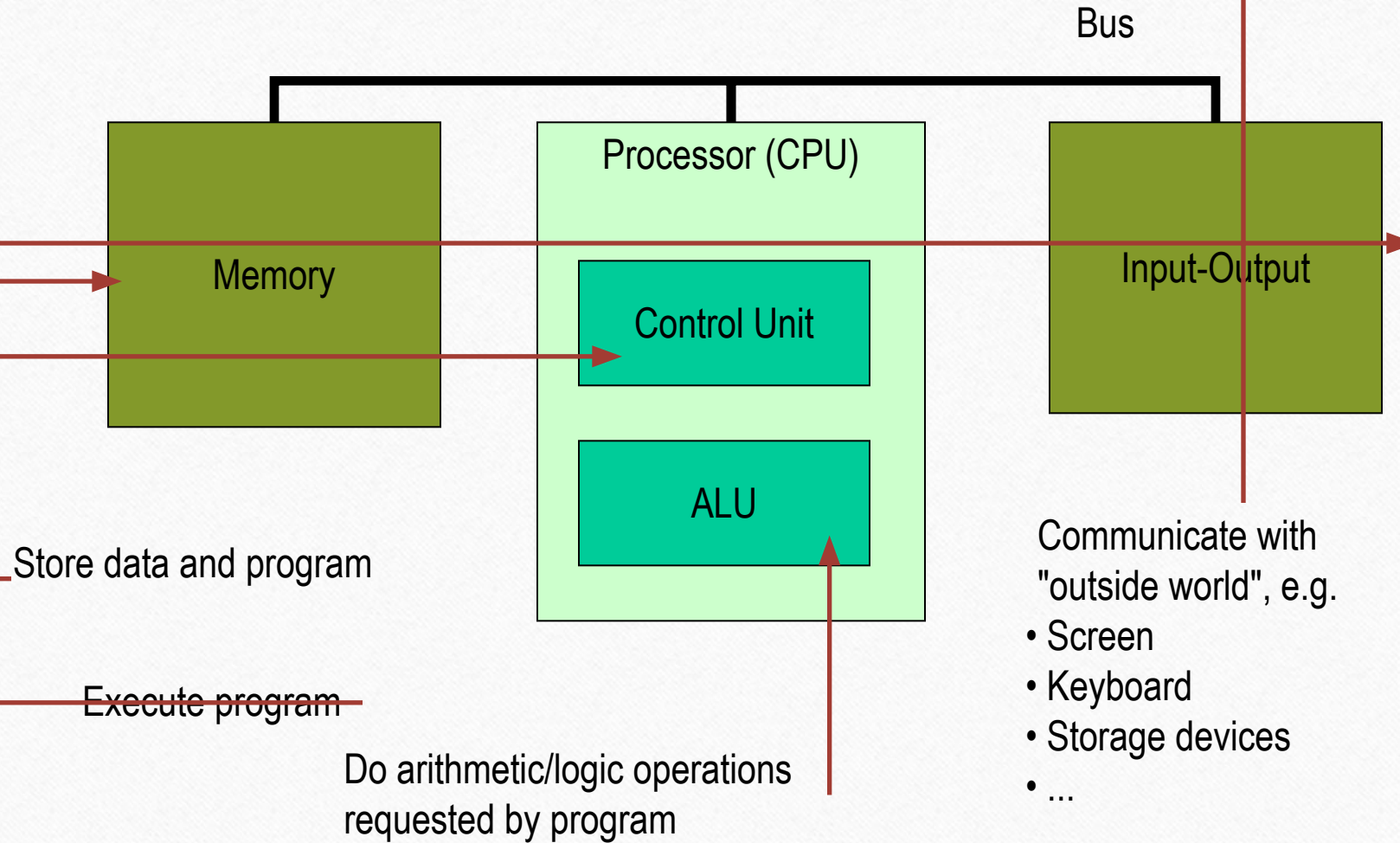
# Von Neumann Architecture

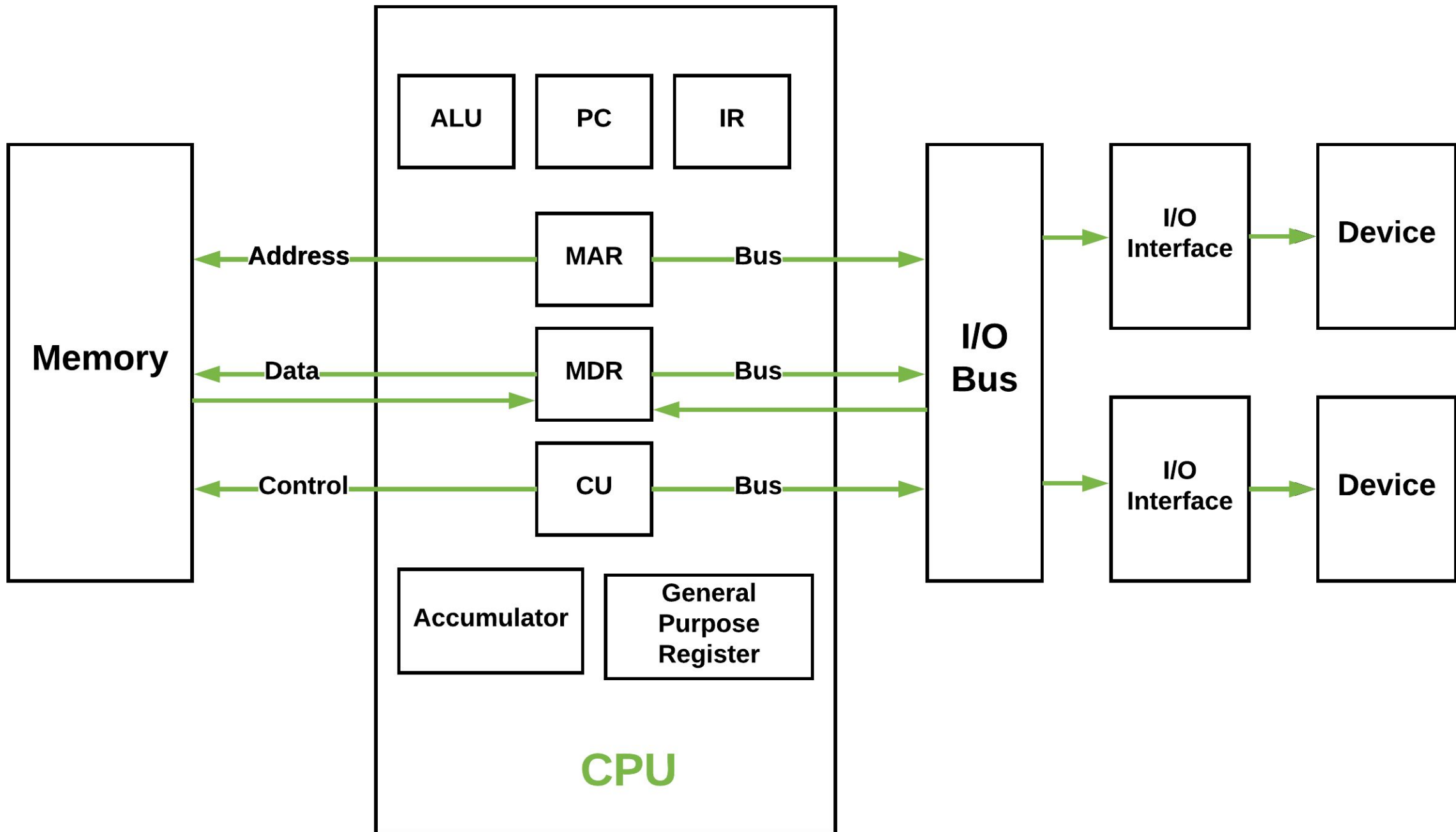
Model for designing and building computers, based on the following three characteristics:

- The computer consists of four main sub-systems:
  1. Memory
  2. ALU (Arithmetic/Logic Unit)
  3. Control Unit
  4. Input/Output System (I/O)
- Von Neumann architecture is based on the stored-program computer concept, where instruction data and program data are stored in the same memory

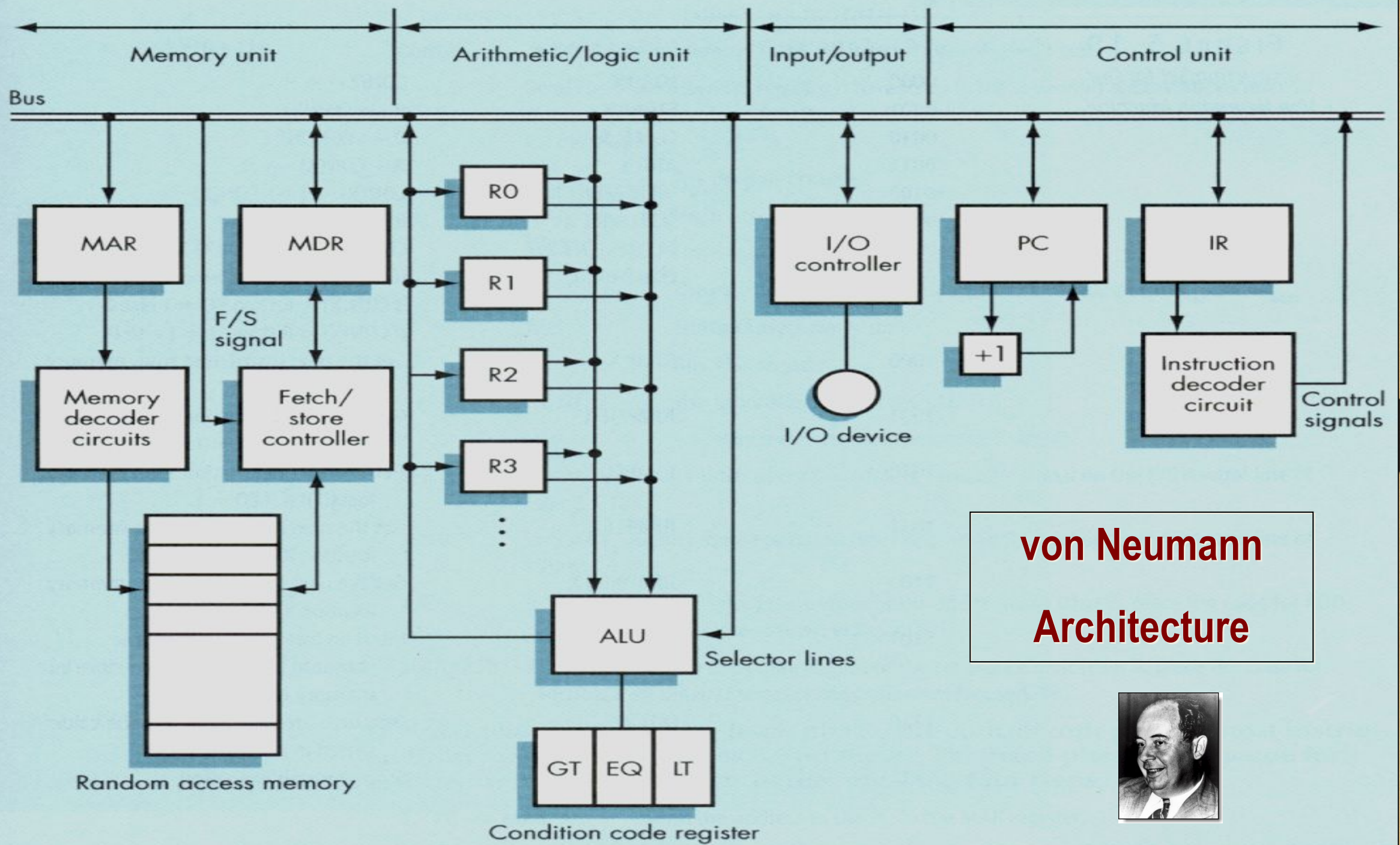


# The Von Neumann Architecture









## von Neumann Architecture



# The Von Neumann Architecture

## ✓ Control Unit –

A control unit (CU) handles all processor control signals.

It directs all input and output flow, fetches code for instructions, and controls how data moves around the system.

## ✓ Arithmetic and Logic Unit (ALU) –

The arithmetic logic unit is that part of the CPU that handles all the calculations the CPU may need, e.g. Addition, Subtraction, Comparisons.

It performs Logical Operations, Bit Shifting Operations, and Arithmetic operations.



# The Von Neumann Architecture

## ✓ Main Memory Unit (Registers) –

1. **Accumulator:** Stores the results of calculations made by ALU.

2. **Program Counter (PC):** Keeps track of the memory location of the next instructions to be dealt with. The PC then passes this next address to Memory Address Register (MAR).

3. **Memory Address Register (MAR):** It stores the memory locations of instructions that need to be fetched from memory or stored into memory.

4. **Memory Data Register (MDR):** It stores instructions fetched from memory or any data that is to be transferred to, and stored in, memory.

5. **Current Instruction Register (CIR):** It stores the most recently fetched instructions while it is waiting to be coded and executed.

6. **Instruction Buffer Register (IBR):** The instruction that is not to be executed immediately is placed in the instruction buffer register IBR.

# The Von Neumann Architecture

## ✓ Input/Output Devices –

Program or data is read into main memory from the input device or secondary storage under the control of CPU input instruction.

Output devices are used to output the information from a computer.

## ✓ Buses –

Data is transmitted from one part of a computer to another, connecting all major internal components to the CPU and memory, by the means of Buses.

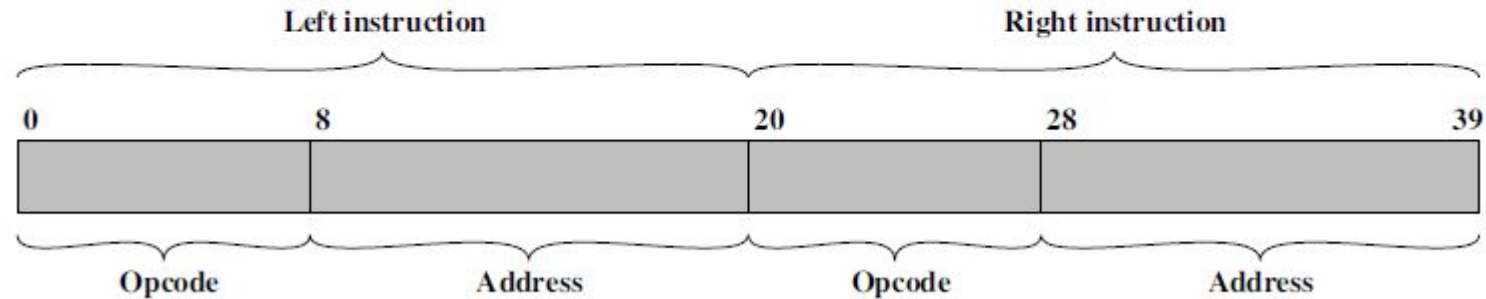
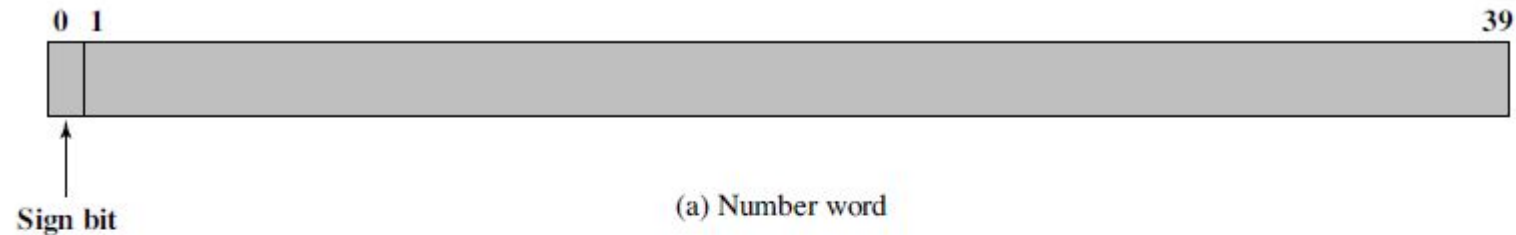
### Types:

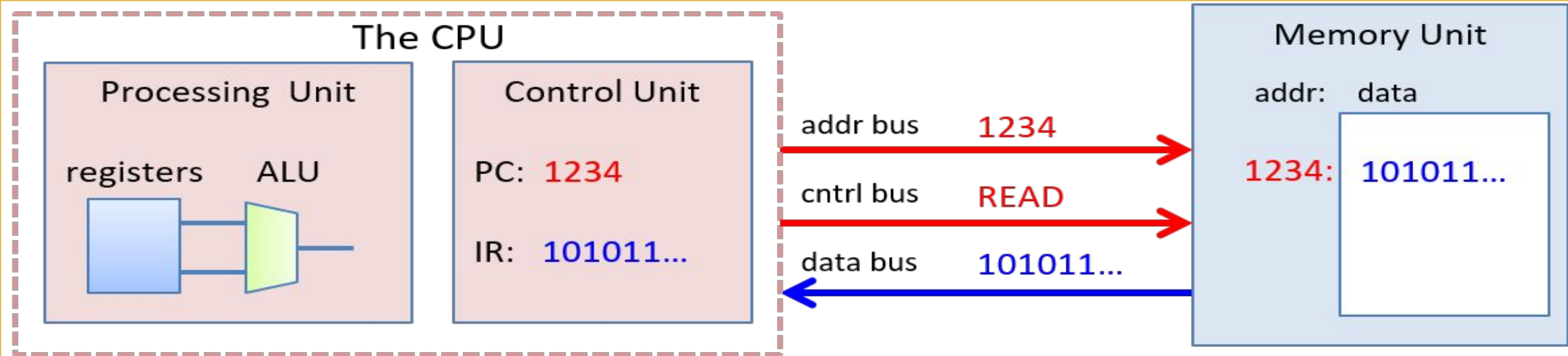
1. **Data Bus:** It carries data among the memory unit, the I/O devices, and the processor.
2. **Address Bus:** It carries the address of data (not the actual data) between memory and processor.
3. **Control Bus:** It carries control commands from the CPU (and status signals from other devices) in order to control and coordinate all the activities within the computer.



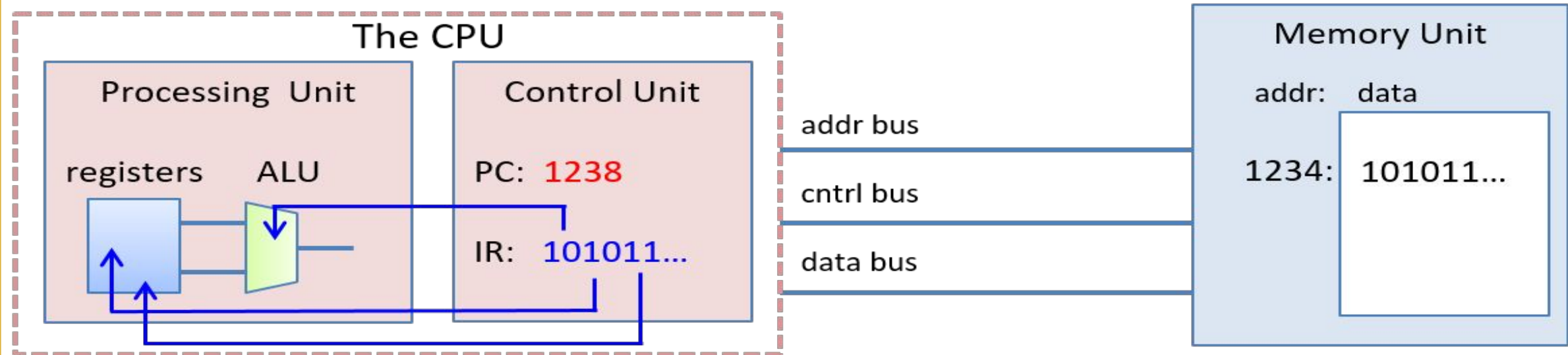
# IAS Memory Formats

The IAS machine was a binary computer with a 40-bit word, storing two 20-bit instructions in each word



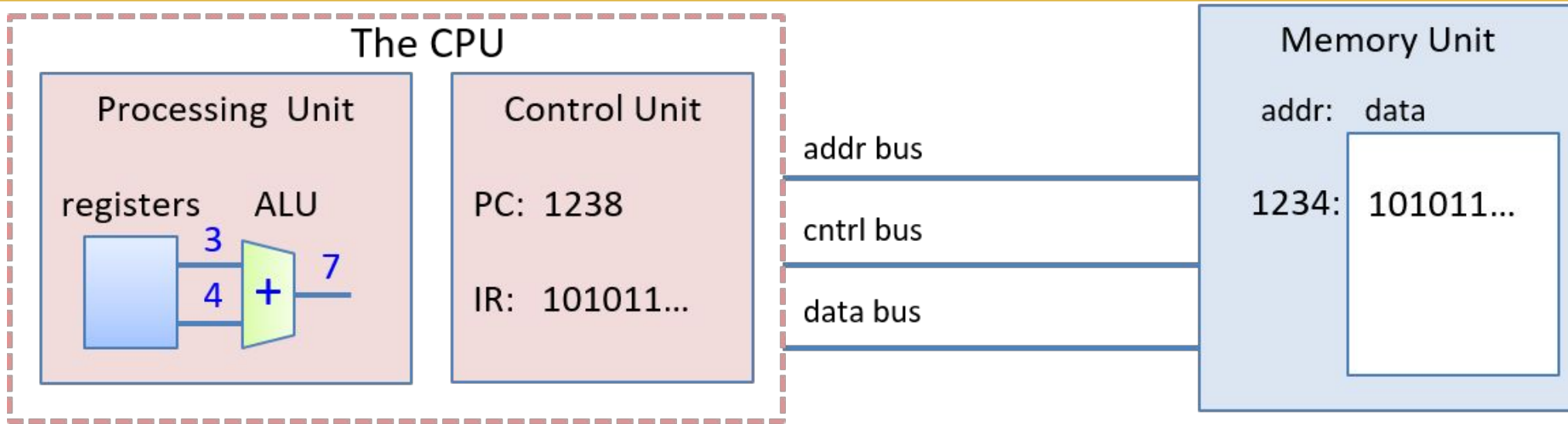


**1. Fetch:** Read instruction bits from memory at address in PC (1234), and store in IR

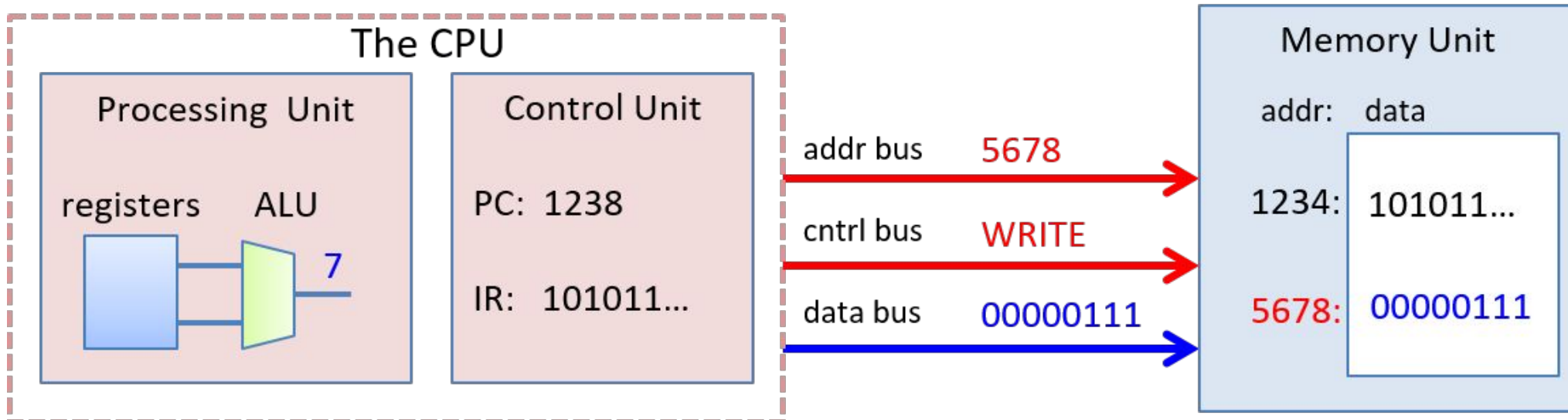


**2. Decode:** instruction bits in IR encode which registers store operands & the ALU operation



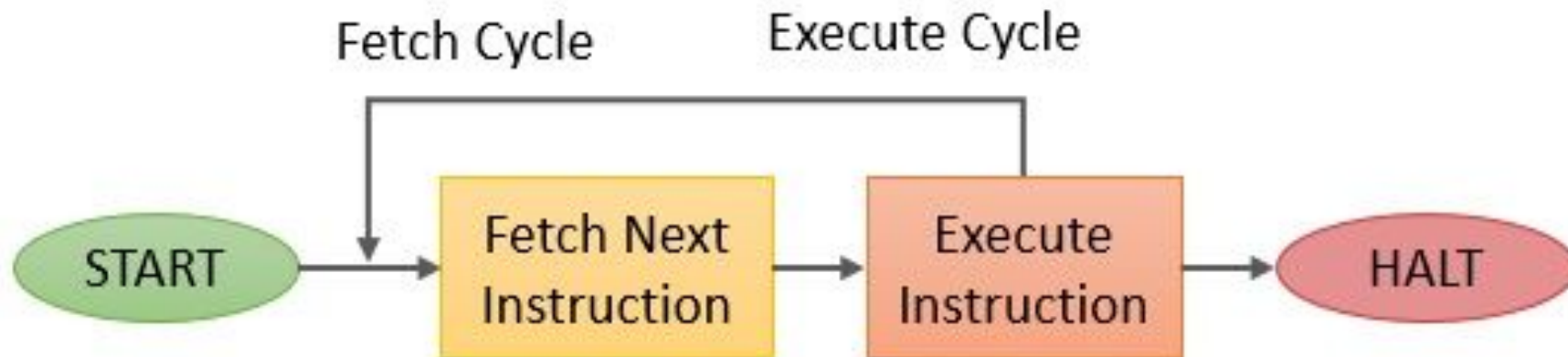


**3. Execute:** ALU performs instruction operation (+) on operands (3,4) to compute result (7)



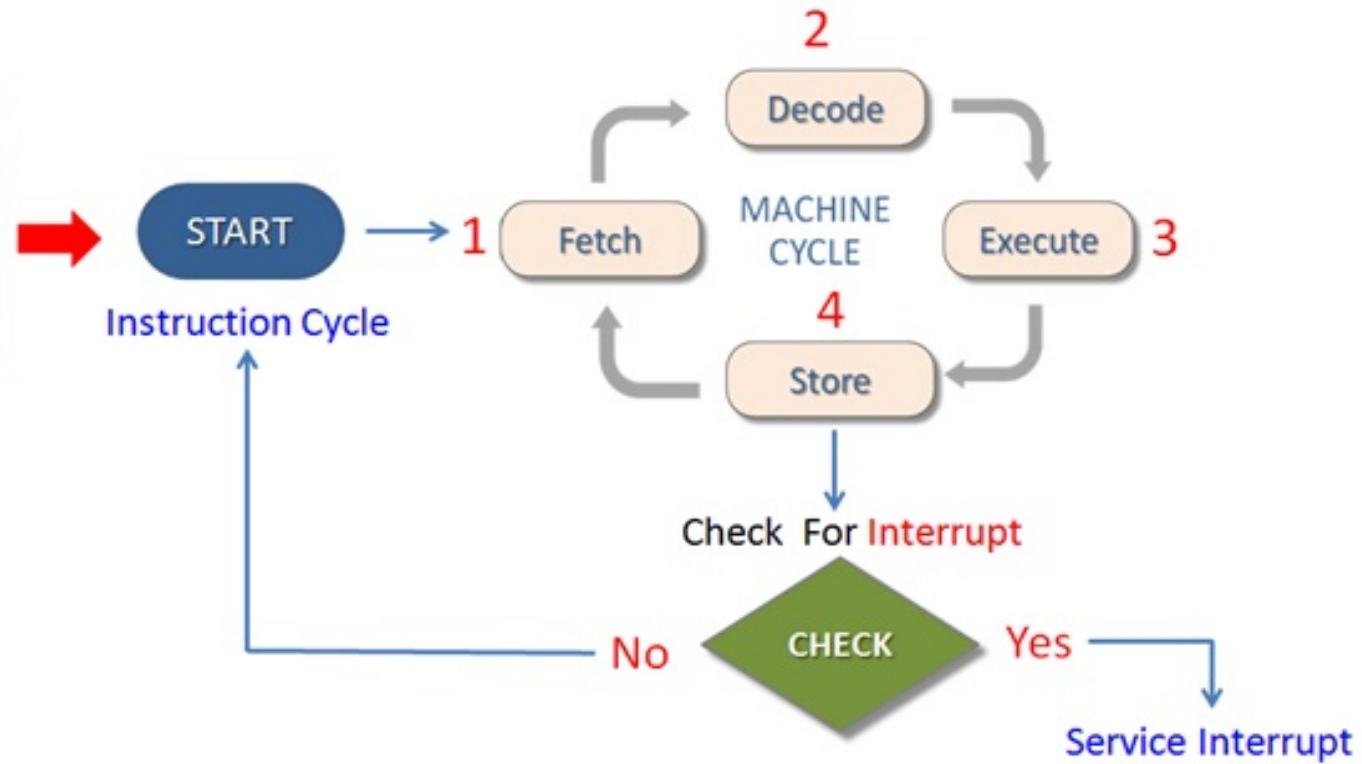
**4. Store:** the control unit stores the ALU result (7, binary 00000111) to memory

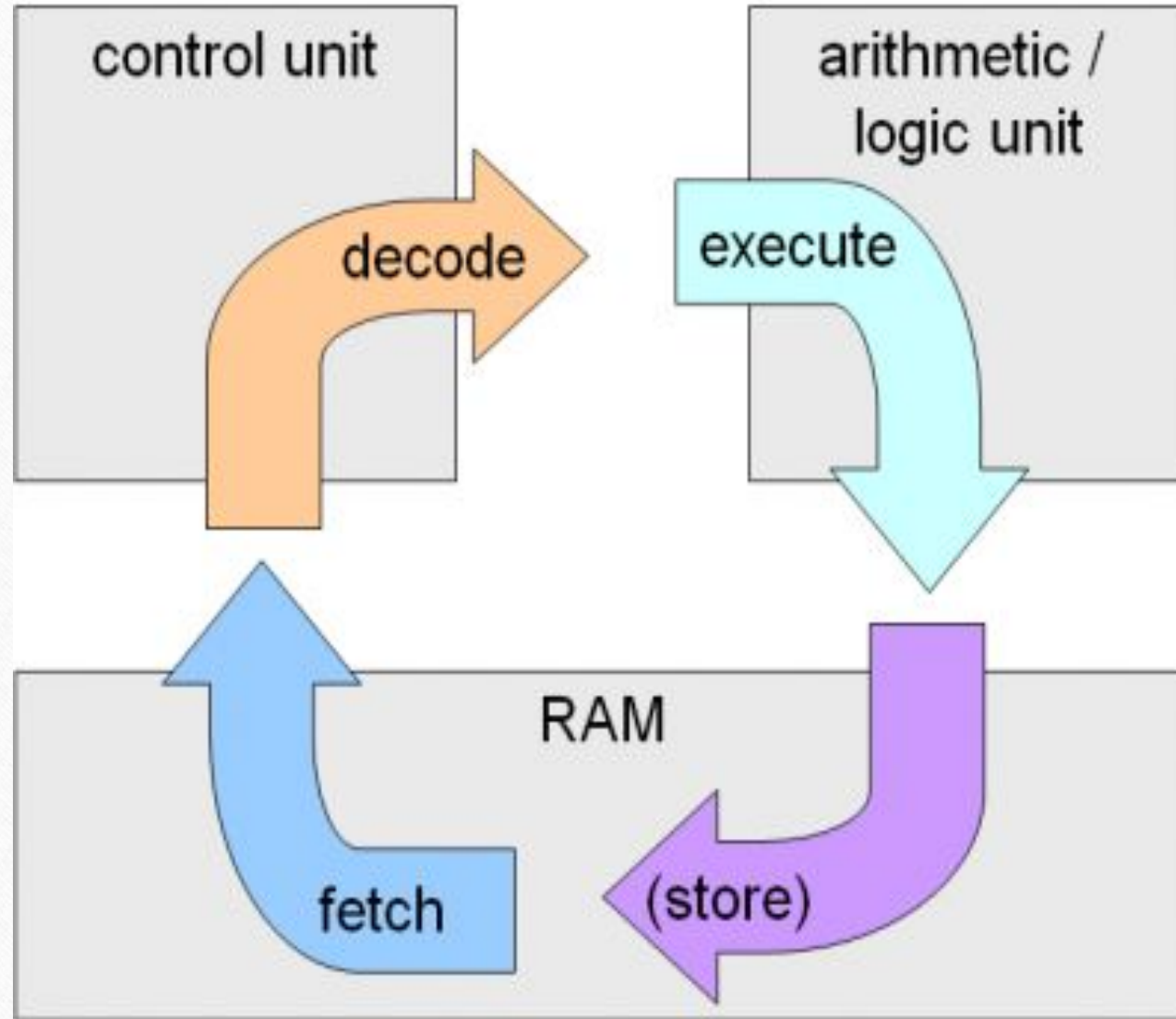
# Instruction execution cycle



**Basic Instruction Cycle**







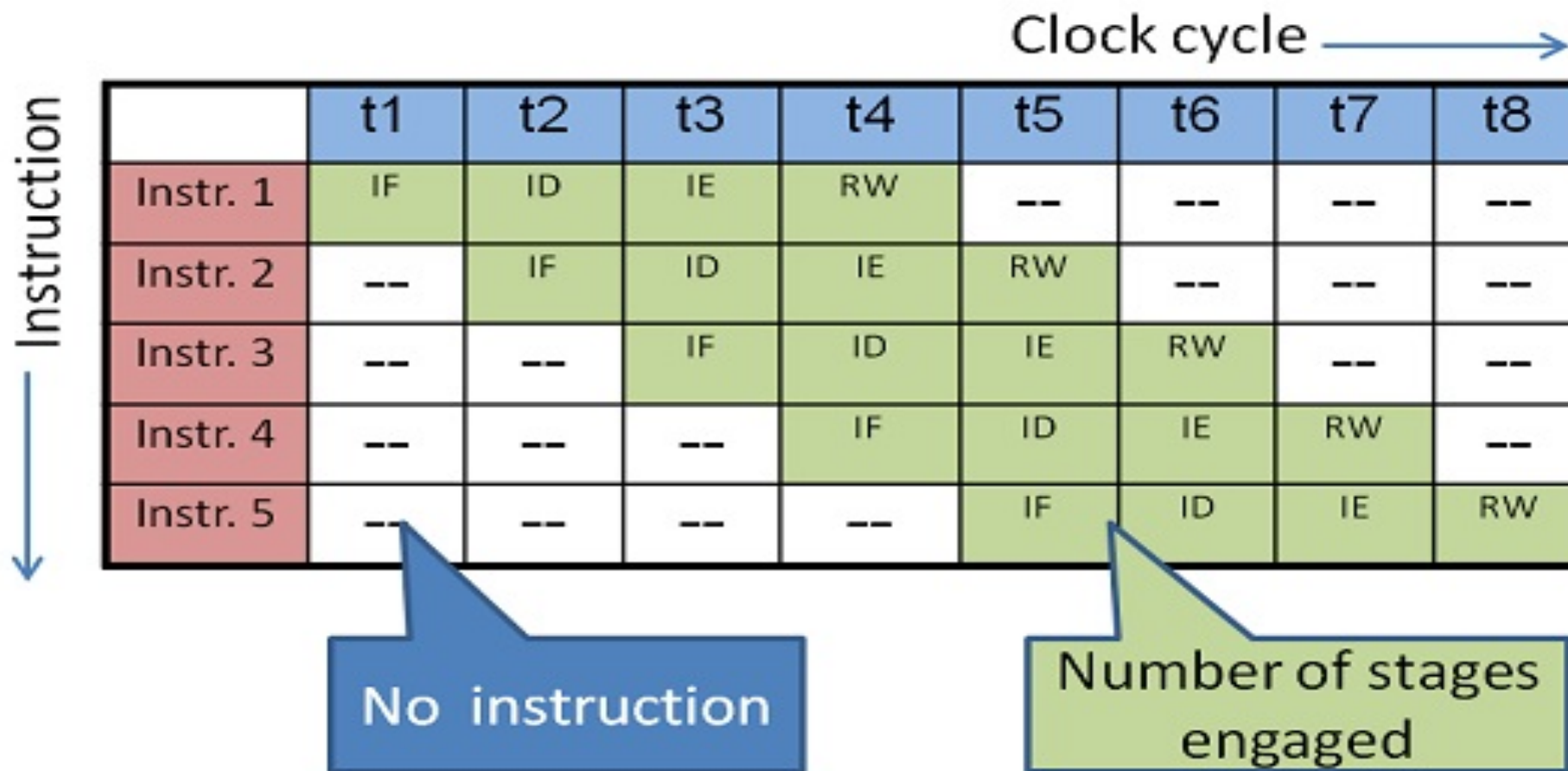


Each instruction cycle includes the following procedures –

1. It can fetch instruction from memory.
2. It is used to decode the instruction.
3. It can read the effective address from memory if the instruction has an indirect address.
4. It can execute the instruction.

- ✓ After the above four procedures are done, the control switches back to the first step and repeats the similar process for the next instruction.
- ✓ Therefore, the cycle continues until a **Halt** condition is met.
- ✓ The halt condition appears when the device receive turned off, on the circumstance of unrecoverable errors, etc.

# Instruction Pipeline





# Evolution of Intel processor architecture- 4 bit to 64 bit

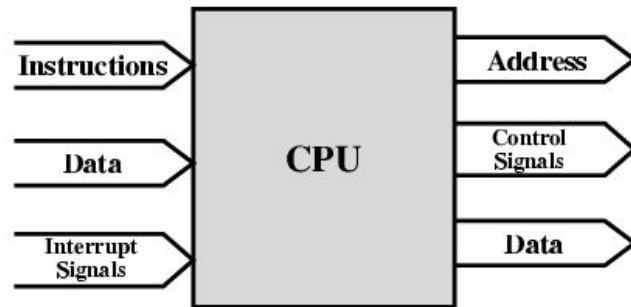
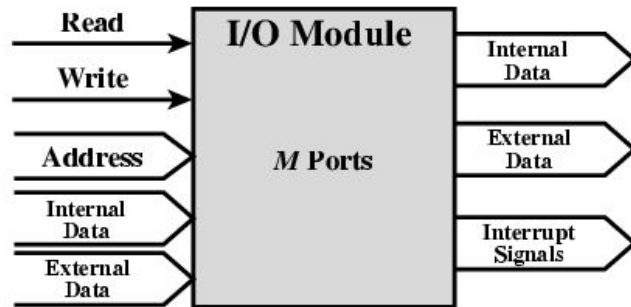
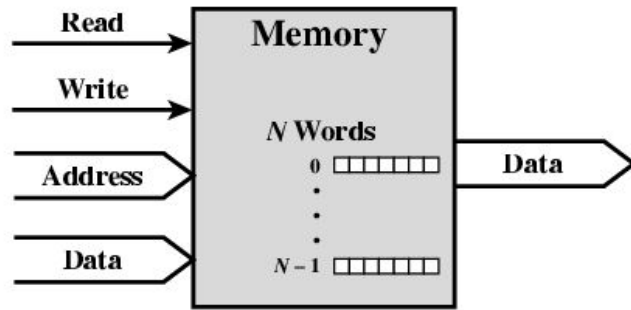
---

Presentation by Students

# Interconnection Structures

- A computer consists of a set of components (CPU, memory, I/O) that communicate with each other.
- The collection of paths connecting the various modules is called as *interconnection structure*.
- The design of this structure will depend on the exchange that must be made between modules.





# Computer Modules

# Type of transfers

- Memory to CPU
- CPU to Memory
- I/O to CPU
- CPU to I/O
- I/O to or from Memory (DMA)



# Bus Interconnection

- A bus is a communication pathway connecting two or more device.
- A key characteristic of a bus is that it is a shared transmission medium.
- A bus consists of multiple pathways or lines.
- Each line is capable of transmitting signal representing binary digit (1 or 0)

# Bus Interconnection

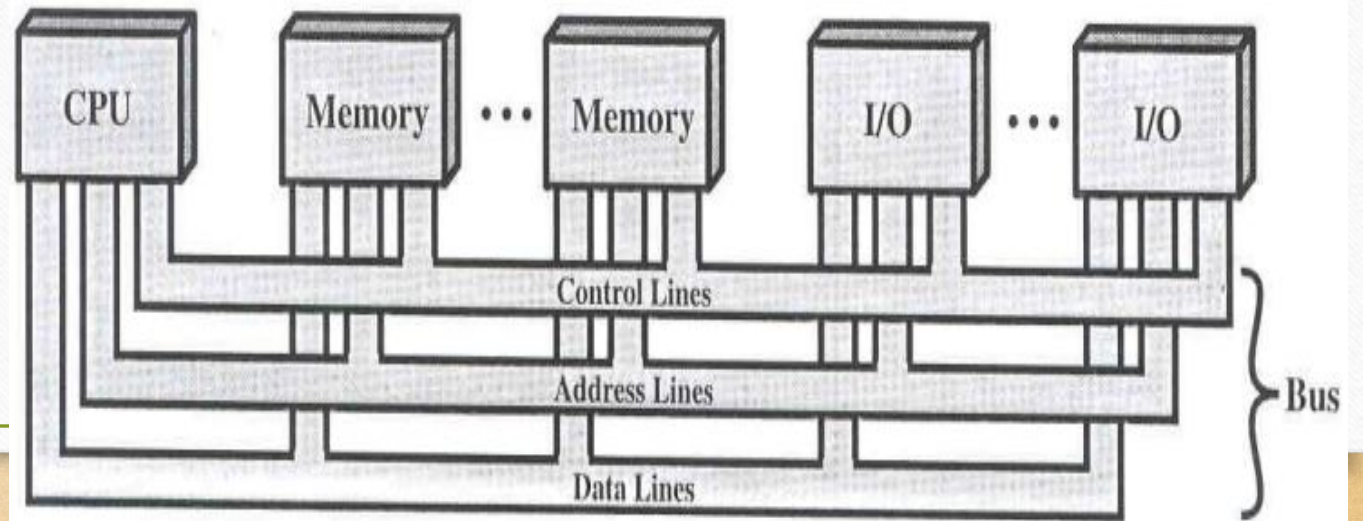
---

- A sequence of bits can be transmit across a single line.
- Several lines can be used to transmit bits simultaneously (in parallel).
- A bus that connects major components (CPU,Memory,I/O) is called *System Bus*.
- The most common computer interconnection structures are based on the use of one or more system buses.



# Bus Structure

- A system bus consists of 50-100 lines.
- Each line is assigned a particular meaning or function.
- On any bus the lines can be classified into 3 groups
  - Data lines
  - Address lines
  - Control lines



# Data Lines

---

- Provide a path for moving data between system modules.
- These lines, collectively, are called the data bus
- The data bus typically consists of 8,16 or 32 separate lines, the numbers of lines being transferred to as the width of the data bus.
- Each line carry only 1 bit at a time, the number of lines determines how many bits can transferred at a time - overall system performance.



# The Address Lines

- Used to designate the source or destination of the data on the data bus
- The width of the address bus determines the maximum possible memory capacity of the system.

# The Control Lines

- Used to control the access to and the use of the data and address lines.
- Typical control lines include
  - Memory write
  - Memory read
  - I/O write
  - I/O read
  - Clock
  - Reset
  - Bus request
  - Bus grant
  - Interrupt request
  - Interrupt ACK
  - Transfer ACK



# The operation of the bus

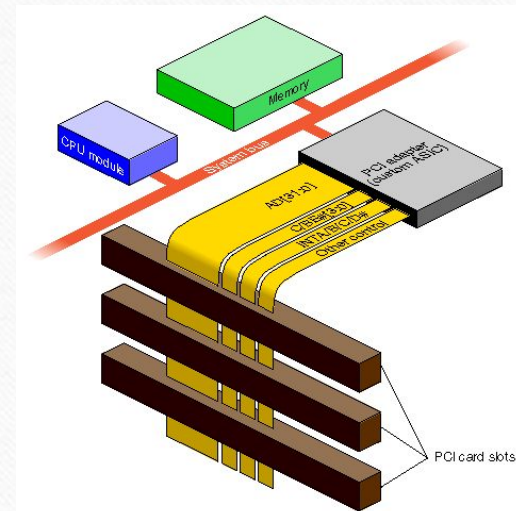
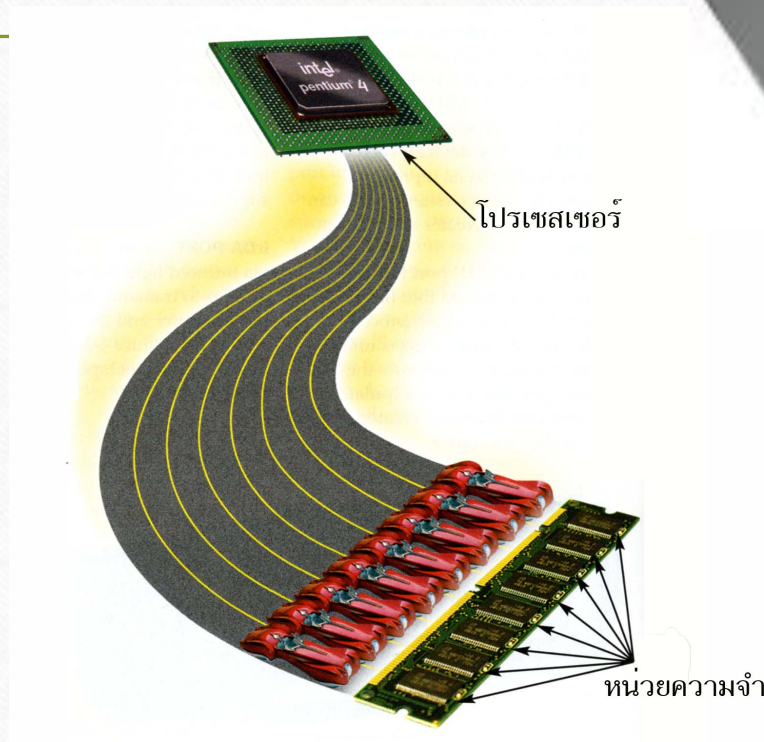
If one module wishes to send data

- obtain the use of the bus
- transfer data via the bus

If one module wishes to request data

- obtain the use of the bus
- transfer request to the other module over the control and address lines, then wait for that second module to send the data.

# What do buses look like?





# Bus Arbitration

---

- In computer system one or more modules may need control of the data bus or address bus. E.g I/O module may need to read or write data directly in memory without sending data to a processor.
- Because only one unit at a time can successfully transmit over the bus , some arbitration method is required.
- Two approaches are followed for the bus Arbitration:
  1. **Centralized Bus Arbitration**
  2. **Distributed Bus Arbitration**

# Centralized Bus Arbitration

- In a centralized scheme, a single hardware device, referred to as bus controller/bus arbiter/bus master, is responsible for allocating time to requesting module.
- Bus master can be a separate module or a part of processor.

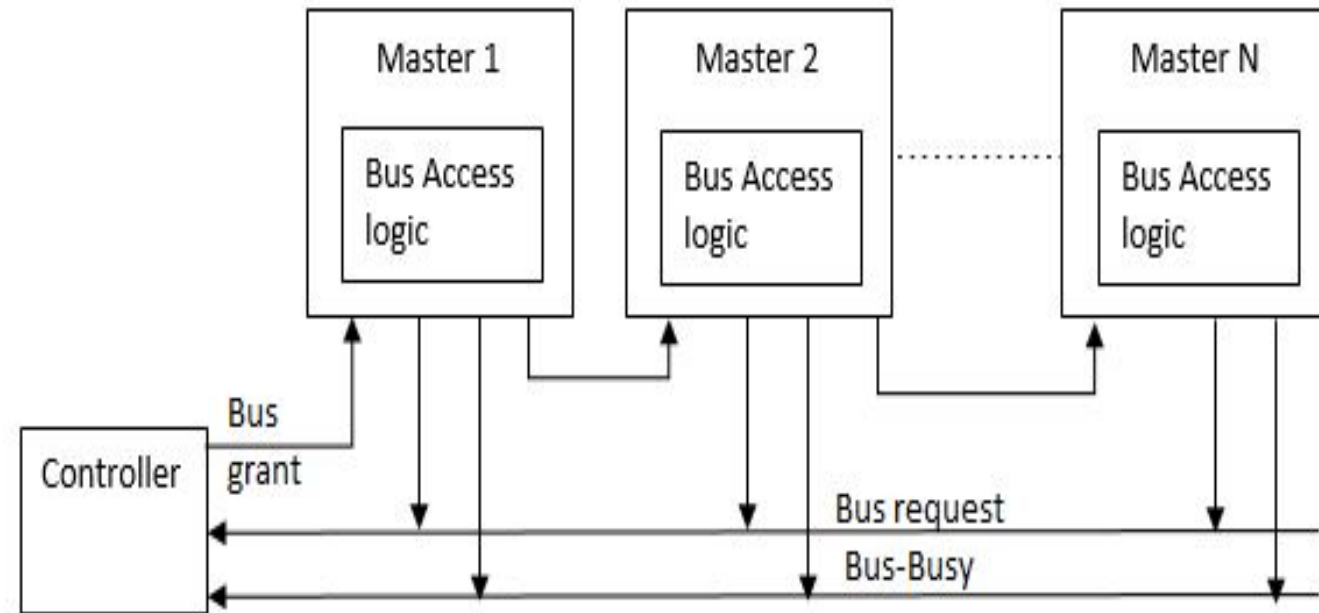
## Methods of Centralized BUS Arbitration:

1. Daisy Chaining Method
2. Polling or Rotating Priority method
3. Fixed priority or Independent Request method



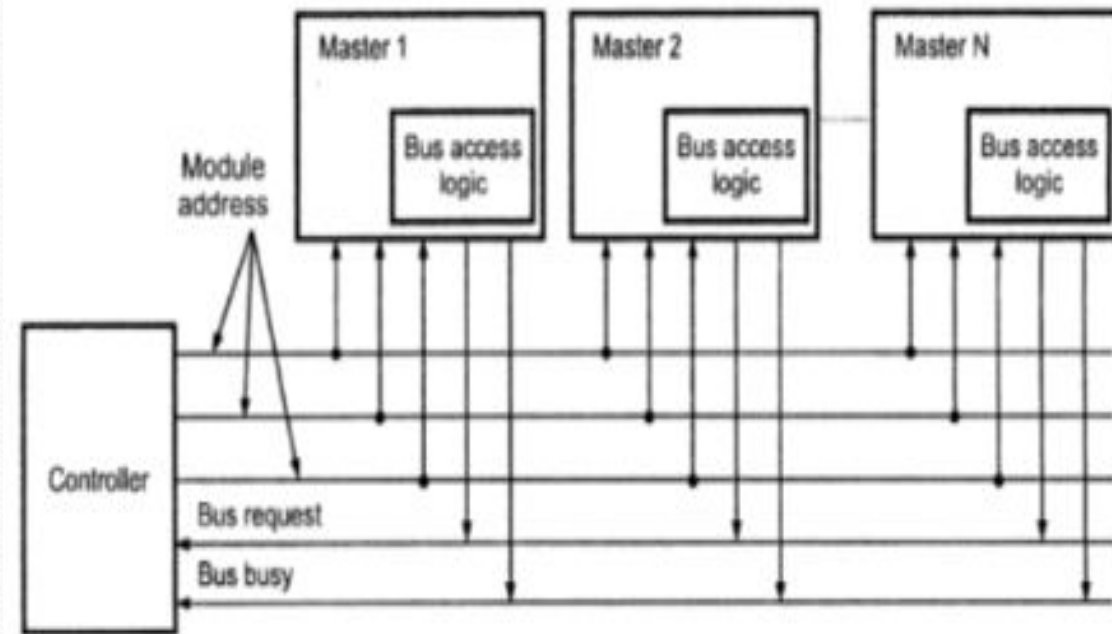
# Daisy Chaining Method

- It is a simple and cheaper method where all the bus masters use the same line for making bus requests.
- The bus grant signal serially propagates through each master until it encounters the first one that is requesting access to the bus.
- This master blocks the propagation of the bus grant signal by sending bus busy status.
- Therefore any other requesting module will not receive the grant signal and hence cannot access the bus.



# Polling or Rotating Priority method

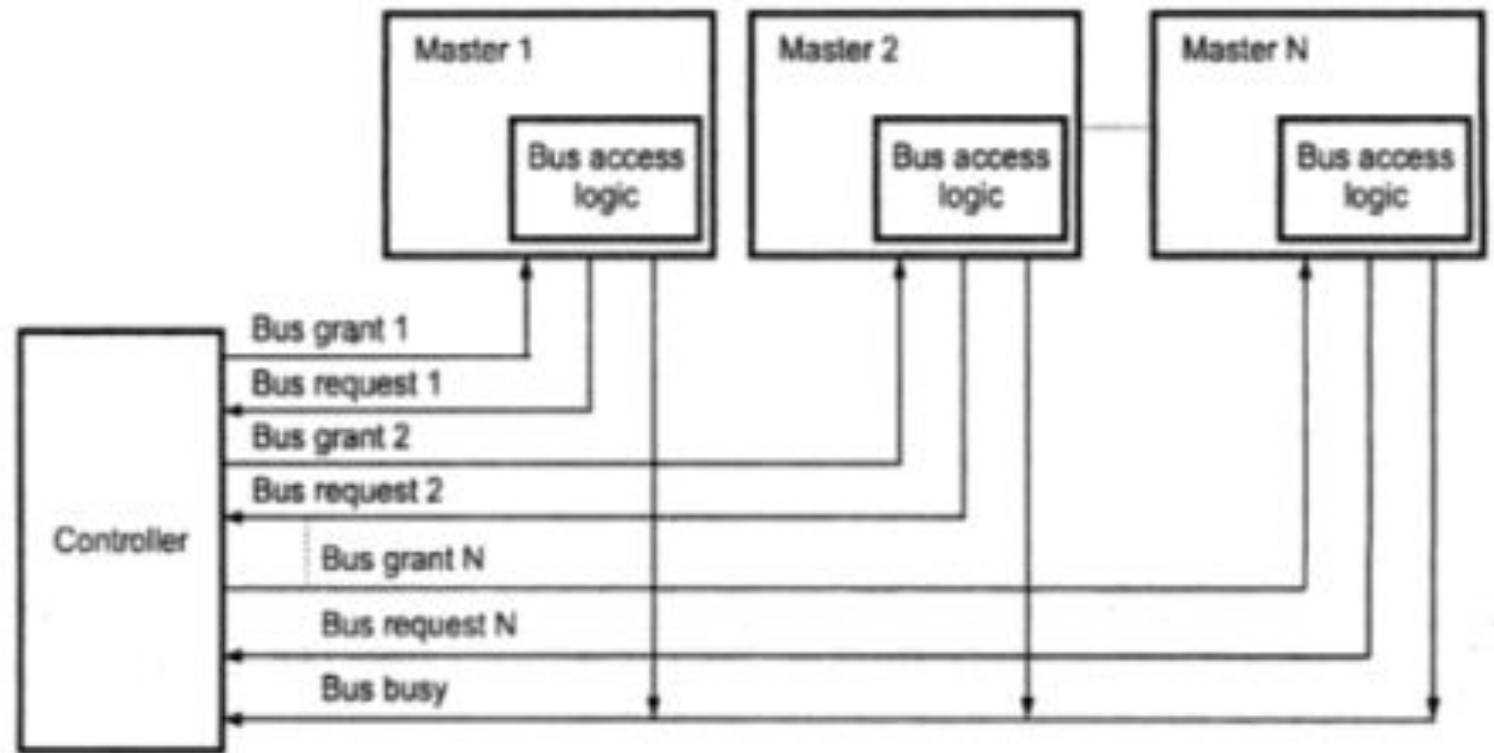
- In this, the controller is used to generate the address for the master(unique priority), the number of address lines required depends on the number of masters connected in the system.
- The controller generates a sequence of master addresses. When the requesting master recognizes its address, it activates the busy line and begins to use the bus.





# Fixed priority or Independent Request method

- In this, each master has a separate pair of bus request and bus grant lines and each pair has a priority assigned to it.
- The built-in priority decoder within the controller selects the highest priority request and asserts the corresponding bus grant signal.



# Distributed Bus Arbitration

- In distributed arbitration, all devices participate in the selection of the next bus master.
- In this scheme each device on the bus is assigned a 4-bit identification number.
- The number of devices connected on the bus when one or more devices request for the control of bus, they assert the start-arbitration signal and place their 4-bit ID numbers on arbitration lines, ARB0 through ARB3.
- These four arbitration lines are all open-collector.
- Therefore, more than one device can place their 4-bit ID number to indicate that they need to control of bus



# Asynchronous and synchronous bus

## ❑ Asynchronous bus

- A bus that interconnects devices of a computer system where information transfers between devices are self-timed rather than controlled by a synchronizing clock signal.
- A connected device indicates its readiness for a transfer by activating a request signal. For example, communication via fax or voice mail are all forms of asynchronous communication.
- low speed buses are asynchronous.

## ❑ Synchronous bus

- A bus used to interconnect devices that comprise a computer system where the timing of transactions between devices is under the control of a synchronizing clock signal.
- For example, a telephone conversation, a video conference and a chat room discussion are all forms of synchronous communication.
- High speed buses are generally synchronous