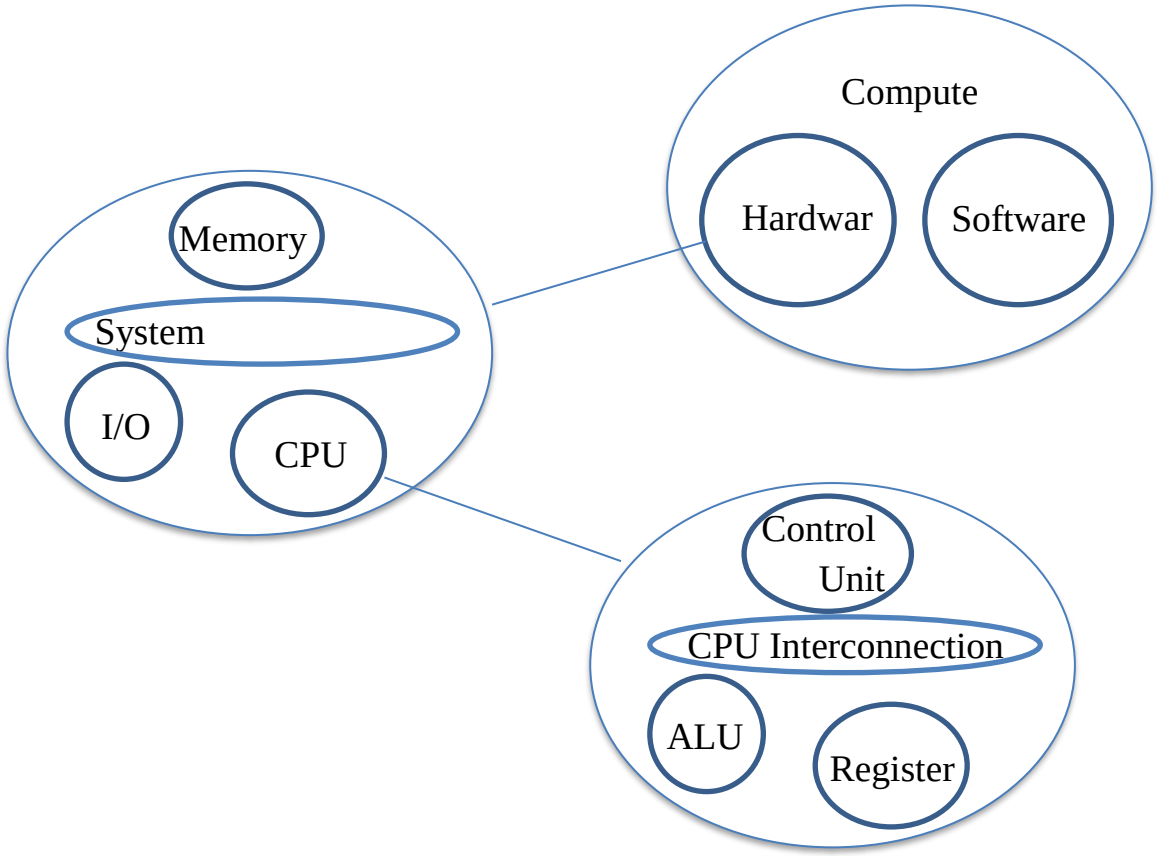
Computer Science Dept. / 2nd year 2nd Course (2024-2025)
Computer Architecture
Dr.Haleema Essa

Computer Architecture

Computer architecture is a specification describing how hardware and software technologies interact to create a computer system. Computer Architecture deals with two essential roles:

- **Instruction-Set Architecture (ISA):** includes the specifications that determine how machine language programmers will interact with the computer.
- **Hardware-System Architecture (HSA):** deals with the computer's major hardware subsystems, including its Central Processing Unit (CPU); its storage system and its Input-Output (I/O) system.



Computer Architecture System

A classification of computer architectures:

➢ Von Neumann Machines:

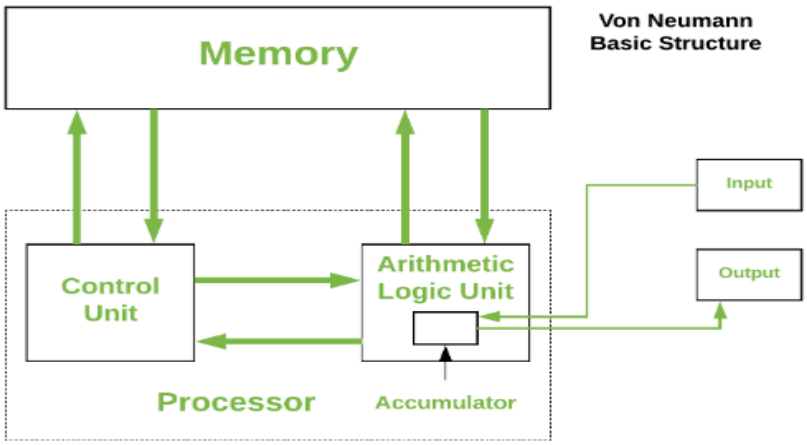
The von Neumann architecture is a computer architecture design that describes a system where both program instructions and data reside in the same memory space.

Historically there have been 2 types of Computers:

1. **Fixed Program Computers** – Their function is very specific and they cannot be reprogrammed, e.g. Calculators.
2. **Stored Program Computers** – These can be programmed to carry out many different tasks, applications are stored on them.

Modern computers are based on a ***stored-program*** concept introduced by John Von Neumann. In this stored-program concept, programs and data are stored in the same memory. This novel idea meant that a computer built with this architecture would be much easier to reprogram.

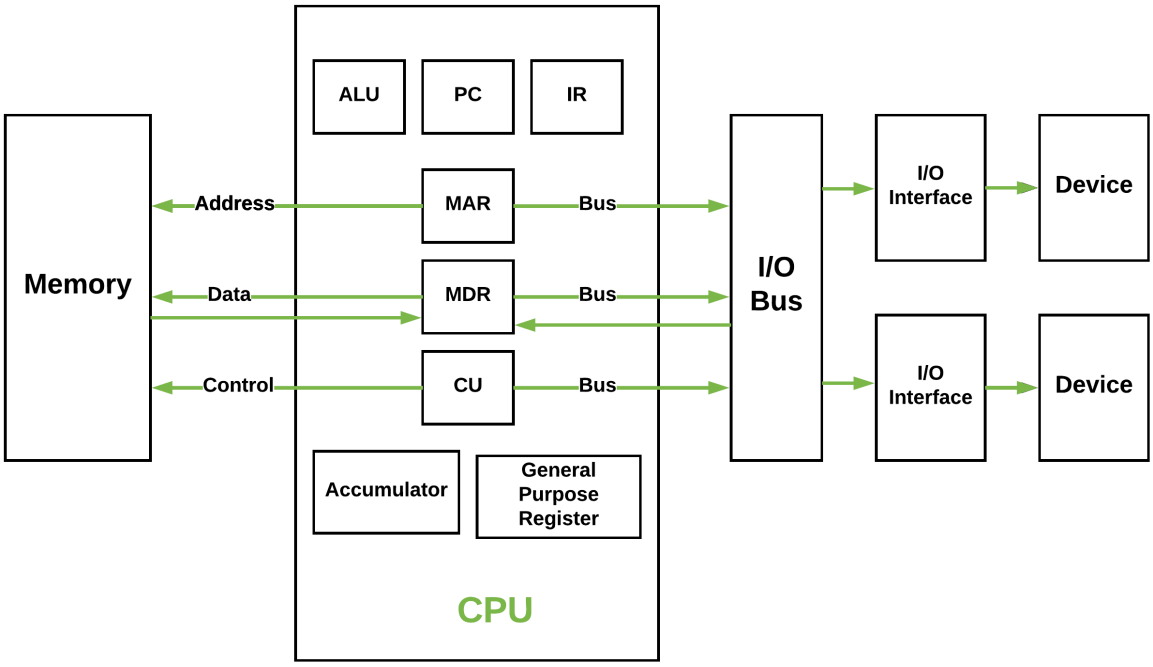
The basic structure of Neumann architecture is shown in the following figure:



It is also known as ISA (Instruction set architecture) computer and is having three basic units:

1. The Central Processing Unit (CPU)
2. The Main Memory Unit
3. The Input/Output Devices.

1. Central Processing Unit: The central processing unit is defined as an electric circuit used for the executing the instruction of computer program. It has following major components: **Control Unit(CU), Arithmetic and Logic Unit(ALU), and Variety of Registers.**



Basic CPU structure

- **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, Comparison. It performs Logical Operations, Bit Shifting Operations, and Arithmetic operations.

- **Registers** – Registers refer to high-speed storage areas in the CPU. The data processed by the CPU are fetched from the registers. There are different types of registers used in architecture :-
  - ✓ **Accumulator**: Stores the results of calculations made by ALU. It holds the intermediate of arithmetic and logical operatoins. it acts as a temporary storage location or device.
  - ✓ **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 the Memory Address Register (MAR).
  - ✓ **Memory Address Register (MAR):** It stores the memory locations of instructions that need to be fetched from memory or stored in memory.
  - ✓ **Memory Data Register (MDR):** It stores instructions fetched from memory or any data that is to be transferred to, and stored in, memory.
  - ✓ **Current Instruction Register (CIR):** It stores the most recently fetched instructions while it is waiting to be coded and executed.
  - ✓ **Instruction Buffer Register (IBR):** The instruction that is not to be executed immediately is placed in the instruction buffer register IBR.

2- Buses

Data is transmitted from one part of a computer to another, connecting all major internal components to the CPU and memory through 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) to control and coordinate all the activities within the computer**.**

3- 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 information from a computer.

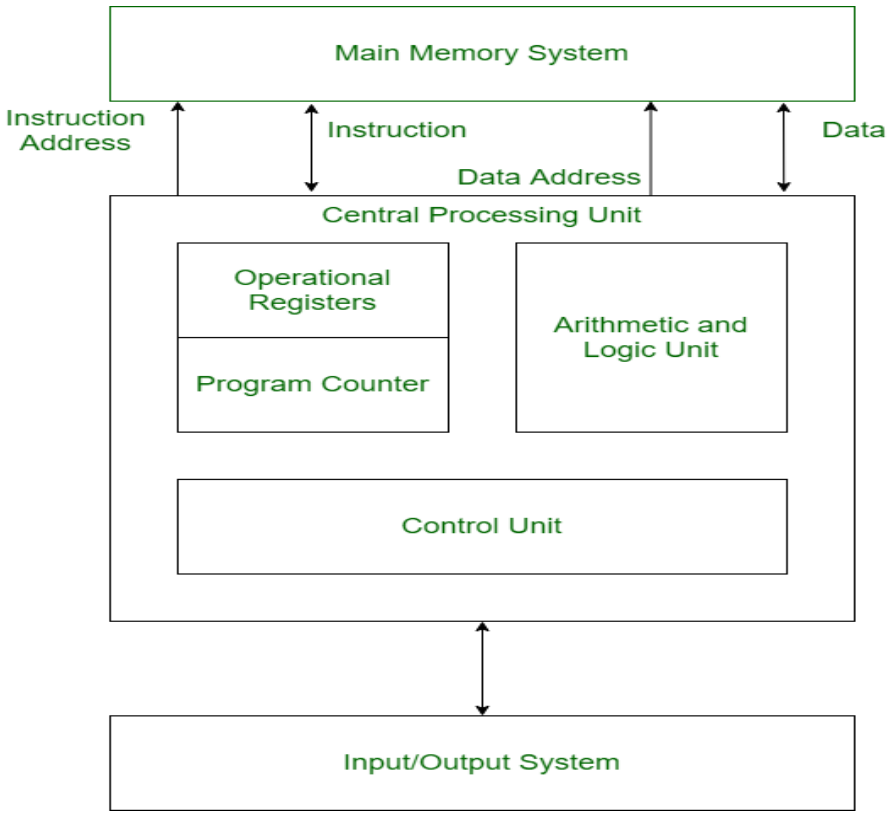
Von Neumann bottleneck: The single pathway between the CPU and memory can create a bottleneck, ***limiting the speed of data transfer.***

HW1: Discuss this issue. How we can overcome it???

➢ Harvard Architecture

Harvard Architecture is the computer architecture that contains separate storage and separate buses (signal path) for instruction and data. It was basically developed to overcome the bottleneck of Von Neumann's Architecture. The main advantage of having separate

buses for instruction and data is that the CPU can access instructions and read/write data at the same time.



Structure of Harvard Architecture

➢ *Buses*

Buses are used as signal pathways. In Harvard architecture, there are separate buses for both instruction and data. Types of Buses:

- **Data Bus:** It carries data among the main memory system, processor, and I/O devices.
- **Data Address Bus:** It carries the address of data from the processor to the main memory system.
- **Instruction Bus:** It carries instructions among the main memory system, processor, and I/O devices.
- **Instruction Address Bus:** It carries the address of instructions from the processor to the main memory system.

➢ *Operational Registers*

- **Program counter (PC):** It functions similarly to the PC in Von Neumann architecture.
- **Instruction address register (IAR):** which is also known as the ***instruction fetch register*** and stores the address of the instruction currently being fetched from the instruction memory.
- **Data address register (DAR):** which stores the address of data to be read from or written to data memory.

***HW2*:** Does Harvard architecture use other registers similar to Von Neumann architecture ??

❖ Advantage of Harvard Architecture

- **Parallel instruction and data access:** Since Harvard architecture separates the memory spaces for instructions and data, the processor can access both memory spaces simultaneously, allowing for parallel instruction and data processing.
- **More efficient memory usage:** Harvard architecture allows for more efficient use of memory as the data and instruction memories can be optimized independently, which can lead to better performance.
- **Fast and efficient data access:** Since Harvard architecture has separate memory spaces for instructions and data, it allows for parallel and simultaneous access to both memory spaces, which leads to faster and more efficient data access.

- **Suitable for embedded systems and real-time applications:** Harvard architecture is commonly used in embedded systems and other real-time applications where speed and efficiency are critical.
- **Security:** The separate spaces can also provide a degree of security against certain types of attacks, such as buffer overflow attacks.

❖ Disadvantages of Harvard Architecture

- **Complexity:** The use of separate memory spaces for instructions and data in Harvard architecture adds to the complexity of the processor design and can increase the cost of manufacturing.
- **Limited flexibility:** Harvard architecture has limited flexibility in terms of modifying instructions at runtime because instructions and data are stored in separate memory spaces. This can make certain types of programming more difficult or impossible to implement.
- **Higher memory requirements:** Harvard architecture requires more memory than Von Neumann architecture, which can lead to higher costs and power consumption.
- **Code size limitations:** Fixed instruction length in Harvard architecture can limit the size of code that can be executed, making it unsuitable for some applications with larger code bases.