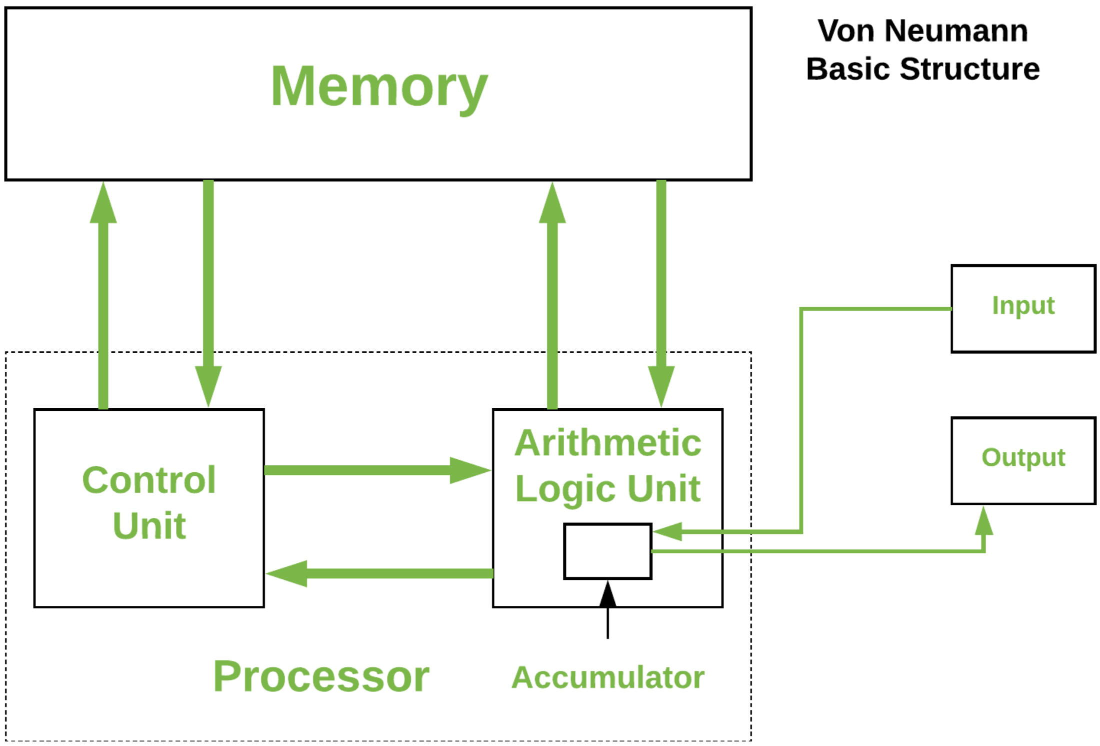
Von Neumann architecture

Computer Organization is like understanding the "blueprint" of how a computer works internally. One of the most important models in this field is the **Von Neumann architecture**, which is the foundation of most modern computers. Named after John von Neumann, this architecture introduced the concept of storing both data and instructions in the same memory.

Historically there have been 2 types of Computers:

1. **Fixed Program Computers -** Their function is very specific and they couldn't be reprogrammed, e.g. Calculators.
2. **Stored Program Computers -** These can be programmed to carry out many different tasks, applications are stored on them, hence the name.

The Von Neumann architecture popularized the stored-program concept, making computers more flexible and easier to reprogram. This design stores both data and instructions in the same memory, simplifying hardware design and enabling general-purpose computing.



The structure described in the figure outlines the basic components of a computer system, particularly focusing on the memory and processor. Here's a breakdown of the components:

- **Memory:** This is where data and instructions are stored. It is a crucial part of the computer system that allows for the storage and retrieval of information.
- **Control Unit:** This component manages the operations of the computer. It directs the flow of data between the CPU and other components.
- **Arithmetic Logic Unit (ALU):** The ALU performs arithmetic and logical operations. It is responsible for calculations and decision-making processes.
- **Input:** This refers to the devices or methods through which data is entered into the computer system.
- **Output:** This refers to the devices or methods through which data is presented to the user or other systems.
- **Processor:** The processor, or CPU, is the central component that carries out the instructions of a computer program. It includes the ALU and Control Unit.
- **Accumulator:** This is a register in the CPU that stores intermediate results of arithmetic and logic operations.

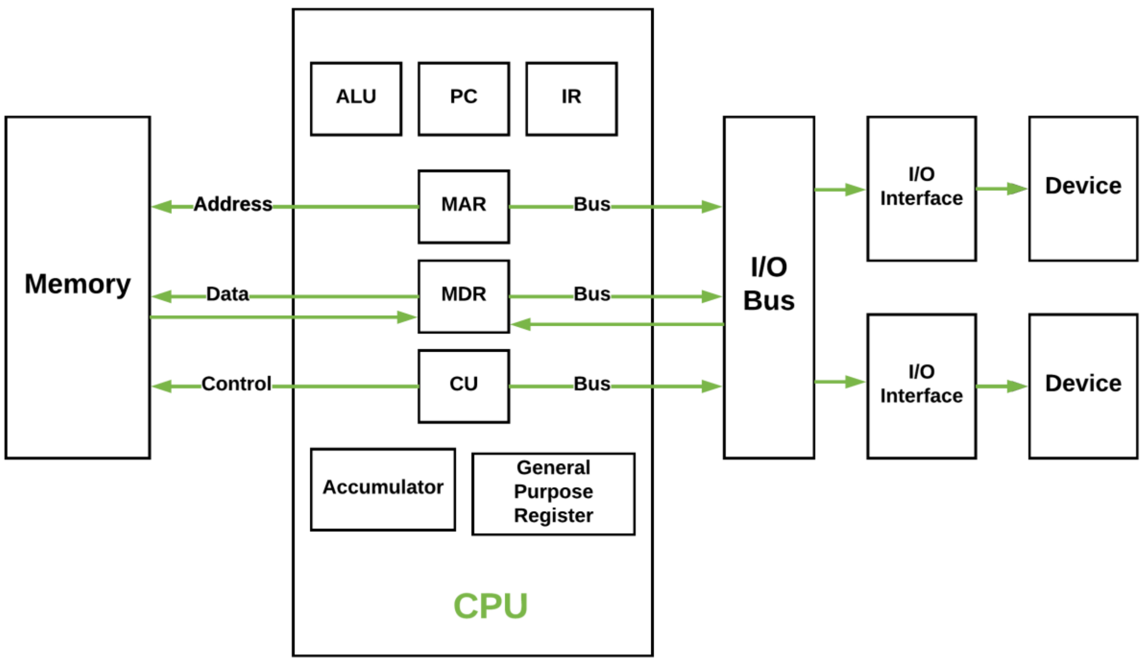


Figure - Basic CPU structure, illustrating ALU

The structure describes **Von Neumann Architecture**, which is a foundational design for modern computers. In this architecture, both data and instructions are stored in the same memory and share a common bus for communication. Here's an explanation of the components of Von Neumann architecture:

Memory

- **Address**: Specifies the location in memory where data or instructions are stored or retrieved.
- **Data**: The actual information (either data or instructions) stored in memory.
- **Control**: Manages the flow of data and instructions between memory and the CPU.

CPU (Central Processing Unit)

The CPU is the core processing unit that executes instructions. It consists of:

- **ALU (Arithmetic Logic Unit)**: Performs arithmetic and logical operations (e.g., addition, subtraction, comparisons).
- **PC (Program Counter)**: Keeps track of the address of the next instruction to be executed.
- **IR (Instruction Register)**: Holds the current instruction being executed.
- **MAR (Memory Address Register)**: Stores the address of the memory location being accessed.
- **MDR (Memory Data Register)**: Temporarily holds data being transferred to or from memory.
- **CU (Control Unit)**: Coordinates the activities of the CPU, managing the flow of data and instructions.
- **Accumulator**: A register that stores intermediate results of arithmetic and logic operations.
- **General Purpose Registers**: Used for temporary storage of data during processing.

Bus

The bus is a communication system that transfers data, addresses, and control signals between the CPU, memory, and I/O devices. In Von Neumann architecture, a single bus is shared for both data and instructions, which can create a bottleneck (known as the Von Neumann bottleneck).

I/O Bus

- **I/O Interface**: Connects the CPU and memory to input/output devices.
- **Device**: Refers to external hardware like keyboards, monitors, or storage devices.

Key Characteristics of Von Neumann Architecture

1. **Single Memory for Data and Instructions**: Both data and program instructions are stored in the same memory.
2. **Shared Bus**: A single bus is used for transferring data, addresses, and control signals, which can limit performance.
3. **Sequential Execution**: Instructions are executed one at a time in a sequential manner.

Von Neumann bottleneck

Whatever we do to enhance performance, we cannot get away from the fact that instructions can only be done one at a time and can only be carried out sequentially. Both of these factors hold back the competence of the CPU. This is commonly referred to as the 'Von Neumann bottleneck'. We can provide a Von Neumann processor with more cache, more RAM, or faster components but if original gains are to be made in CPU performance then an influential inspection needs to take place of CPU configuration.

Advantages of Von Neumann Architecture

- **Simplified Design**: Uses a single memory for data and instructions, reducing hardware complexity.
- **Cost-Effective**: Lower production costs due to fewer components.
- **Flexibility**: Can run various programs and makes it suitable for general-purpose computing.
- **Ease of Programming**: Unified memory structure simplifies software development.
- **Widely Adopted**: Forms the foundation of most modern computers hence, ensures widespread compatibility.

Limitations of Von Neumann Architecture

- **Memory Bottleneck**: Shared memory slows down data and instruction transfer.

- **Sequential Processing**: Cannot process data and instructions simultaneously.
- **Scalability Issues**: Struggles with high-performance tasks requiring rapid memory access.
- **Energy Inefficiency**: Frequent memory access increases power consumption.
- **Latency**: Data and instruction fetch delays reduce overall system efficiency.

Applications of Von Neumann Architecture

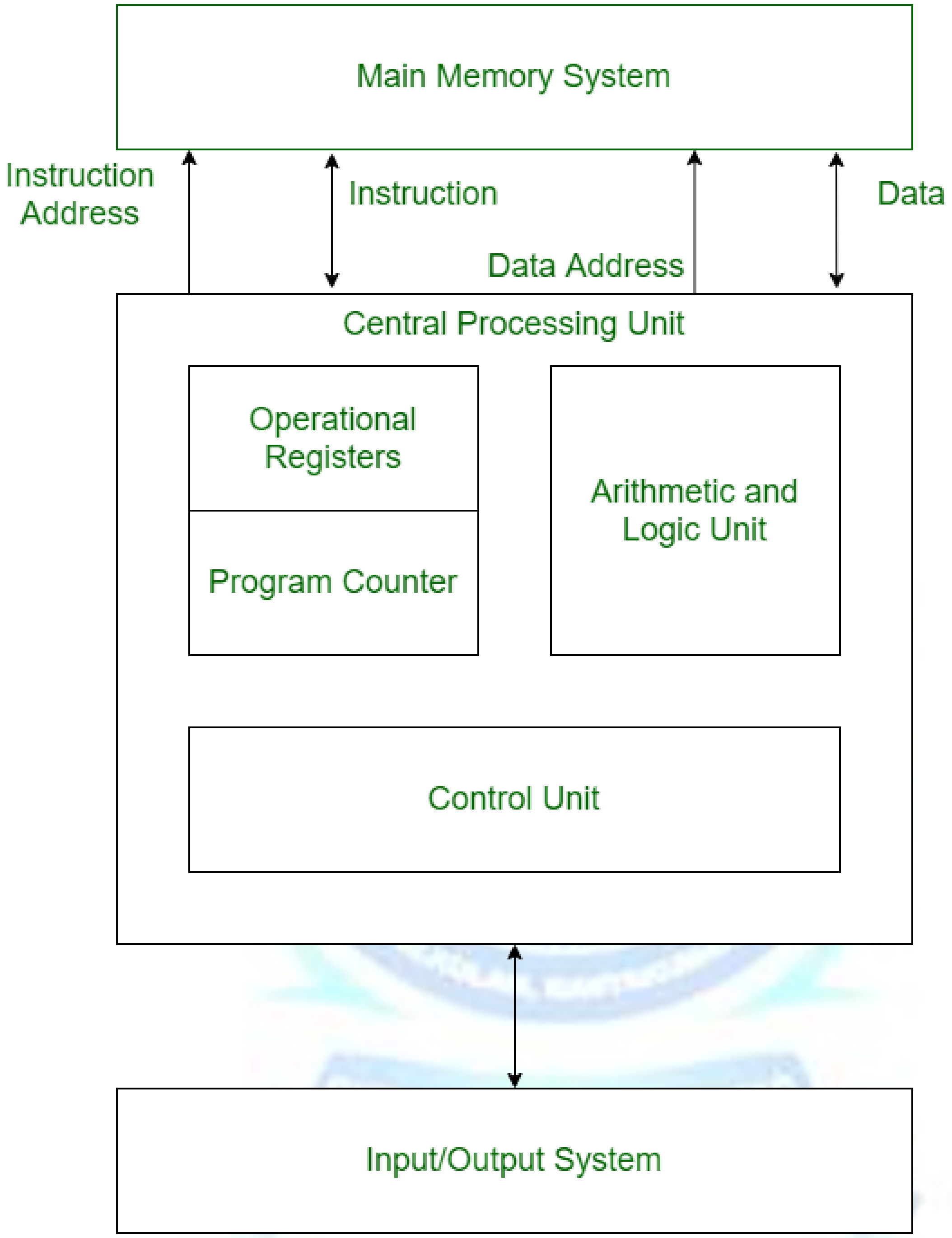
- **General-Purpose Computing**: Powers desktops, laptops, and smartphones.
- **Embedded Systems**: Used in simple devices where cost and simplicity are priorities.
- **Software Development**: Shapes programming tools and languages due to its unified structure.
- **Education**: A foundational concept in computer science courses.
- **Gaming and Multimedia**: Supports complex applications like video games and editing software.

Harvard Architecture

In a normal computer that follows von Neumann architecture, instructions, and data both are stored in the same memory. So same buses are used to fetch instructions and data. This means the CPU cannot do both things together (read the instruction and read/write data). So, to overcome this problem, Harvard architecture was introduced.

The Harvard architecture is a computer architecture that separates memory storage and buses for instructions and data, unlike the von Neumann architecture, which uses a single memory and bus for both. This separation allows the CPU to access instructions and read/write data simultaneously, overcoming the bottleneck of von Neumann's architecture and enhancing processing speed. By enabling parallel access to instructions and data, the Harvard architecture improves performance, making it particularly suitable for applications requiring high-speed processing, such as embedded systems and digital signal processing.

Structure of Harvard Architecture



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

There are different types of registers involved in it which are used for storing addresses of different types of instructions. *For example*, the Memory Address Register and Memory Data Register are operational registers.

- **Program Counter:** It has the location of the next instruction to be executed. The program counter then passes this next address to the memory address register.
- Arithmetic and Logic Unit: The arithmetic logic unit is part of the CPU that operates all the calculations needed. It performs addition, subtraction, comparison, logical Operations, bit Shifting Operations, and various arithmetic operations.
- **Control Unit:** The Control Unit is the part of the CPU that operates all processor control signals. It controls the input and output devices and also controls the movement of instructions and data within the system.
- **Input/Output System:** Input devices are used to read data into main memory with the help of CPU input instruction. The information from a computer as output is given through Output devices. The computer gives the results of computation with the help of output devices.

Features of Harvard Architecture

- **Separate memory spaces:** Harvard architecture uses distinct memory spaces for instructions and data, enabling simultaneous access for faster processing.
- **Fixed instruction length:** In Harvard architecture, instructions are of fixed length, which simplifies the instruction fetch process and allows for faster instruction processing.
- **Parallel instruction and data access:** The separation of instruction and data memories allows parallel access, improving overall efficiency.

- **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.
- **Suitable for embedded systems:** Harvard architecture is commonly used in embedded systems because it provides fast and efficient access to both instructions and data, which is critical in real-time applications.
- **Limited flexibility:** The separate memory spaces restrict tasks like modifying instructions at runtime, reducing flexibility.

Advantage of Harvard Architecture

- **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.
- **Better performance:** The use of fixed instruction length, parallel processing, and optimized memory usage in Harvard architecture can lead to improved performance and faster execution of instructions.
- **Suitable for real-time applications:** Harvard architecture is commonly used in embedded systems and other real-time applications where speed and efficiency are critical.
- **Security:** The sepa5y 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.

Difference between Von Neumann and Harvard Architecture

VON NEUMANN ARCHITECTURE	HARVARD ARCHITECTURE
It is ancient computer architecture based on stored program computer concept.	It is modern computer architecture based on Harvard Mark I relay based model.
Same physical memory address is used for instructions and data.	Separate physical memory address is used for instructions and data.
There is common bus for data and instruction transfer.	Separate buses are used for transferring data and instruction.
Two clock cycles are required to execute single instruction.	An instruction is executed in a single cycle.
It is cheaper in cost.	It is costly than Von Neumann Architecture.
CPU can not access instructions and read/write at the same time.	CPU can access instructions and read/write at the same time.
It is used in personal computers and small computers.	It is used in micro controllers and signal processing.