Introduction to Computer Organization and Architecture

Classes of Computer

- A computer is a device that can be instructed to carry out arbitrary sequences of arithmetic or logical operations automatically.
- Computers share a core set of technologies; the implementation and use of these technologies varies with the chosen application.
- In general, there are three classes of applications to consider: desktop computers, servers, and embedded computers.

Desktop Computers (or Personal Computers)

- Emphasize good performance for a single user at relatively low cost.
- Mostly execute third-party software.

Servers

- Emphasize great performance for a few complex applications.
- Or emphasize reliable performance for many users at once.
- Greater computing, storage, or network capacity than personal computers.

Embedded Computers

- Largest class and most diverse.
- Usually specifically manufactured to run a single application reliably.
- limitations on cost and power.

Architectural differences

- Hardware advances have allowed programmers to create wonderfully useful software, which explains why computers are omnipresent.
- Hardware differences – difference in architecture and design
- How are programs written in a high-level language, such as C or Java, translated into the language of the hardware, and how does the hardware execute the resulting program?
- What techniques can be used by hardware designers to improve performance?

- The performance of a program depends on a combination of the effectiveness of the algorithms used in the program, the software systems used to create and translate the program into machine instructions, and the effectiveness of the computer in executing those instructions, which may include input/output (I/O) operations.

Hardware or software component	How this component affects performance
Algorithm	Determines both the number of source-level statements and the number of I/O operations executed
Programming language, compiler, and architecture	Determines the number of computer instructions for each source-level statement
Processor and memory system	Determines how fast instructions can be executed
I/O system (hardware and operating system)	Determines how fast I/O operations may be executed

Computer Architecture and Computer Organization

- Computer Organization and Architecture provides in-depth knowledge of internal working, structuring, and implementation of a computer system.
- Organization defines the way the system is structured so that all those catalogued tools can be used properly.
- Architecture of a computer system can be considered as a catalogue of tools or attributes that are visible to the user such as instruction sets, number of bits used for data, addressing techniques, etc.

- **Computer Architecture** is a functional description of requirements and design implementation for the various parts of a computer. It deals with the functional behavior of computer systems. It comes before the computer organization while designing a computer.

  Examples:

  - the instruction set
  - the number of bits used to represent various data types
  - I/O mechanisms
  - memory addressing techniques

- *Architecture describes what the computer does.*

- Information representation

Decimal term	Abbreviation	Value	Binary term	Abbreviation	Value	% Larger
kilobyte	KB	10^3	kibibyte	KiB	2^{10}	2%
megabyte	MB	10^6	mebibyte	MiB	2^{20}	5%
gigabyte	GB	10^9	gibibyte	GiB	2^{30}	7%
terabyte	TB	10^{12}	tebibyte	TiB	2^{40}	10%
petabyte	PB	10^{15}	pebibyte	PiB	2^{50}	13%
exabyte	EB	10^{18}	exbibyte	EiB	2^{60}	15%
zettabyte	ZB	10^{21}	zebibyte	ZiB	2^{70}	18%
yottabyte	YB	10^{24}	yobibyte	YiB	2^{80}	21%

- Computer Organization refers to the operational units and their interconnections that realize the architectural specifications.
- **Computer Organization** comes after the decision of Computer Architecture . Computer Organization is how operational attributes are linked together and contribute to realizing the architectural specification. Computer Organization deals with a structural relationship.
- Examples are things that are transparent to the programmer:

  - control signals

  - interfaces between computer and peripherals

  - the memory technology being used

Computer Architecture	Computer Organization
Architecture describes what the computer does.	The Organization describes how it does it.
Computer Architecture deals with the functional behavior of computer systems.	Computer Organization deals with a structural relationship.
Architecture indicates its hardware.	Where Organization indicates its performance.
For designing a computer, its architecture is fixed first.	For designing a computer, an organization is decided after its architecture.
Computer Architecture is also called instruction set architecture.	Computer Organization is frequently called microarchitecture.
Computer Architecture comprises logical functions such as instruction sets, registers, data types, and addressing modes.	Computer Organization consists of physical units like circuit designs, peripherals, and adders.
Architecture coordinates between the hardware and software of the system.	Computer Organization handles the segments of the network in a system.

Differences between Computer Organization and Computer Architecture

Good Architecture ideas

- Improve performance via **pipelining**.

  - Break tasks into stages so that multiple tasks can be simultaneously performed in different stages.

  - Commonly used to improve instruction throughput.

- Improve performance via **prediction**.

  - Sometime faster to assume a particular result than waiting until the result is known.

  - Known as speculation and is used to guess results of branches.

- Use a **hierarchy of memories**.

  -Make the fastest, smallest, and most expensive per bit memory the first level accessed and the slowest, largest, and cheapest per bit memory the last level accessed.

  -Allows most of the accesses to be caught at the first level and be able to retain most of the information at the last level.

- Improve dependability via **redundancy**.

  - Include redundant components that can both detect and often correct failures.

  - Used at many different levels.

Why Computer Organization

- Improving a program's performance is not as simple as reducing its memory usage. Modern programmers need to have an understanding of the issues "below the program":
  - The **parallel nature of processors**.

    -How might you speed up your application by introducing parallelism via threading or multiprocessing?

    -How will the compiler translate and rearrange your own instruction-level code to perform instructions in parallel?
  - The **hierarchical nature of memory**.

    -How can you rearrange your memory access patterns to more efficiently read data?
  - The translation of high-level languages into hardware instructions
  - What decisions are made by the compiler in the process of generating instruction-level statements?

A simple Computer …

- A simple computer understand binary language whose vocabulary contains only two letters or states or symbols i.e. 0 and 1, True and False, On and off.
- To maintain the states **transistors** are used.

    - Transistors are tiny device that are used to store 2 values 1 and 0 or on and off.

    - If the transistor is on we say that it has a value 1, and if it is off the value is 0.

But how a transistor get its value ?
When a very little amount of electric current passes through transistor it maintains the state of 1 and when there is no electric current then the transistor has the state of 0.

Then how it's all connected to computer ?
This 0's and 1's forms the building block of computer. With the combinations of 0 and 1 we create a whole new language – ASCII codes

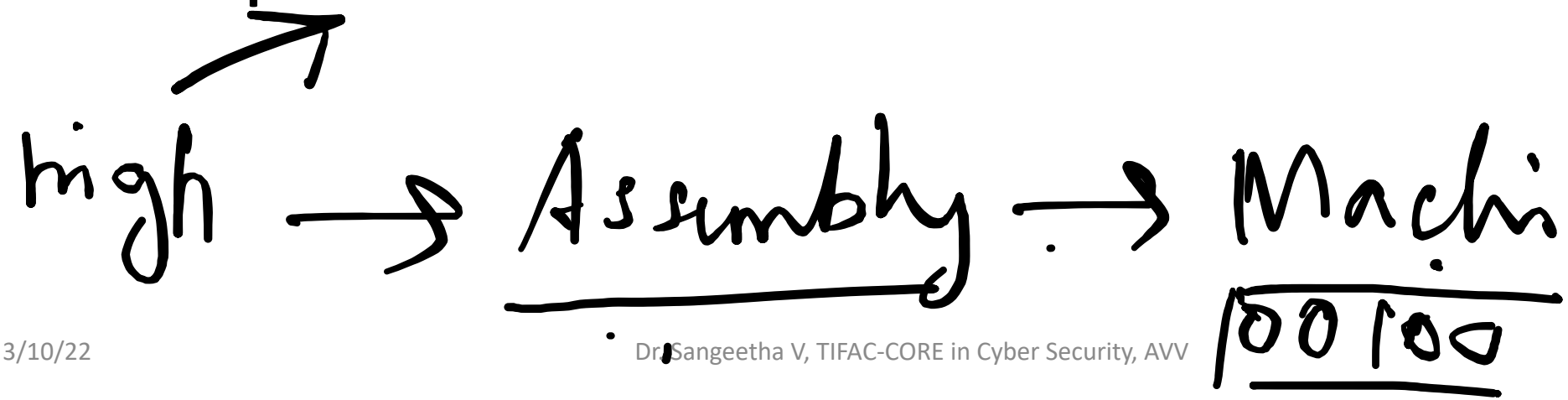
- **So now the question arises how can a human remember this code?**
  – Software
- Suppose we want to add 2 number and want to know what 2 + 2 is 4. Then we must give the computer instructions,

  Step-1: take 2 value.

  Step-2: store that 2 value

  Step-3: add 2 value by using + operator

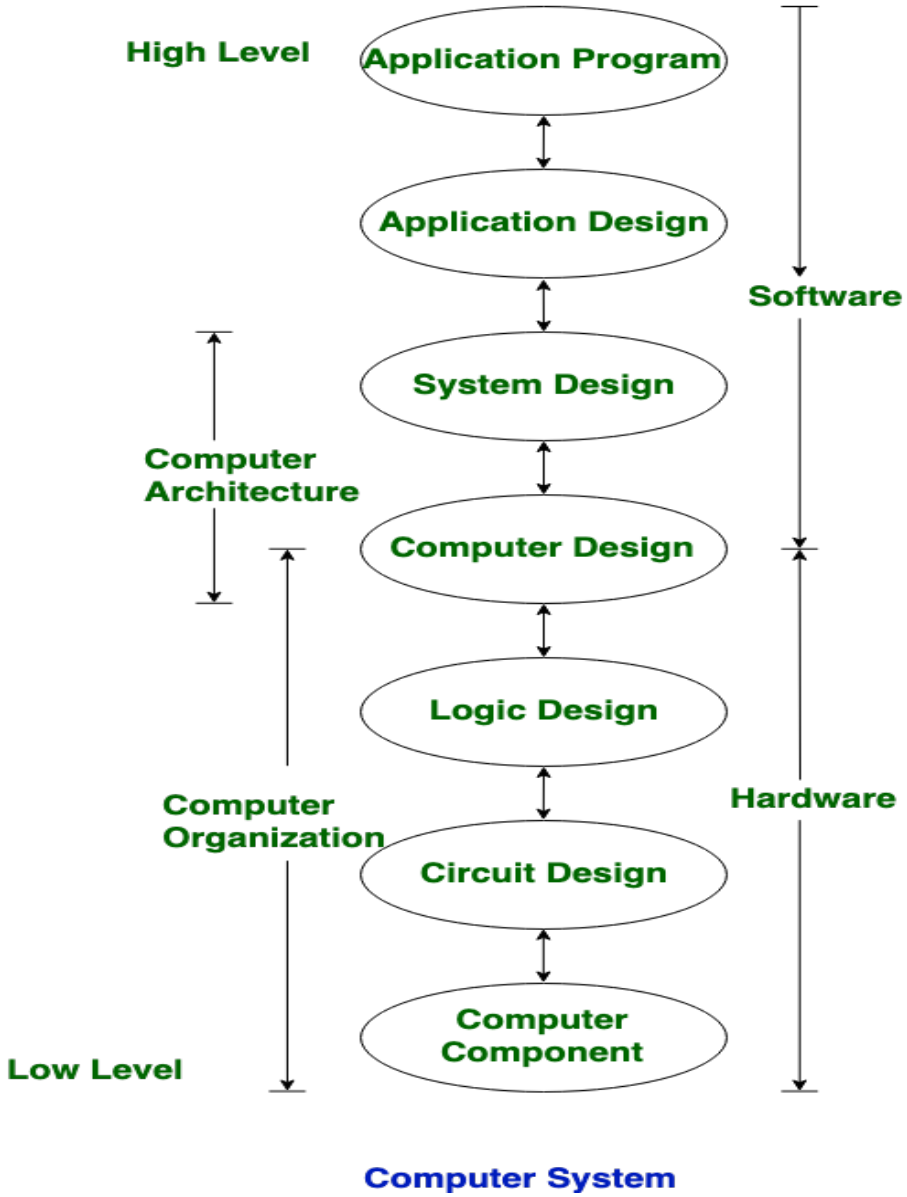
  Step-4: save the answer



Computer System Level Hierarchy

- **Computer Design** is the structure in which components relate to each other. The designer deals with a particular level of system at a time. At each level, the designer is concerned with the structure and function. The structure is the skeleton of the various components related to each other for communication. The function is the activities involved in the system.
- **Computer System Level Hierarchy** is the combination of different levels that connects the computer with the user and that makes the use of the computer. It also describes how the computational activities are performed on the computer and it shows all the elements used in different levels of system.

Computer System



Level 6	User	Executable Programs
Level 5	High Level Language	C++ , Java
Level 4	Assembly Language	Assembly Code
Level 3	System Software	Operating System
Level 2	Machine	Instruction Set Architecture
Level 1	Control	Microcode
Level 0	Digital Logic	Circuits , Gates

- **Level-0:** It is related to digital logic. Digital logic is the basis for digital computing and provides a fundamental understanding of how circuits and hardware communicate within a computer. It consists of various circuits and gates etc.
- **Level-1:** This level is related to control. Control is the level where microcode is used in the system. Control units are included in this level of the computer system.
- **Level-2:** This level consists of machines. Different types of hardware are used in the computer system to perform different types of activities. It contains instruction set architecture.
- **Level-3:** System software is a part of this level. System software is of various types. System software mainly helps in operating the process and it establishes the connection between hardware and user interface. It may consist operating system, library code, etc.

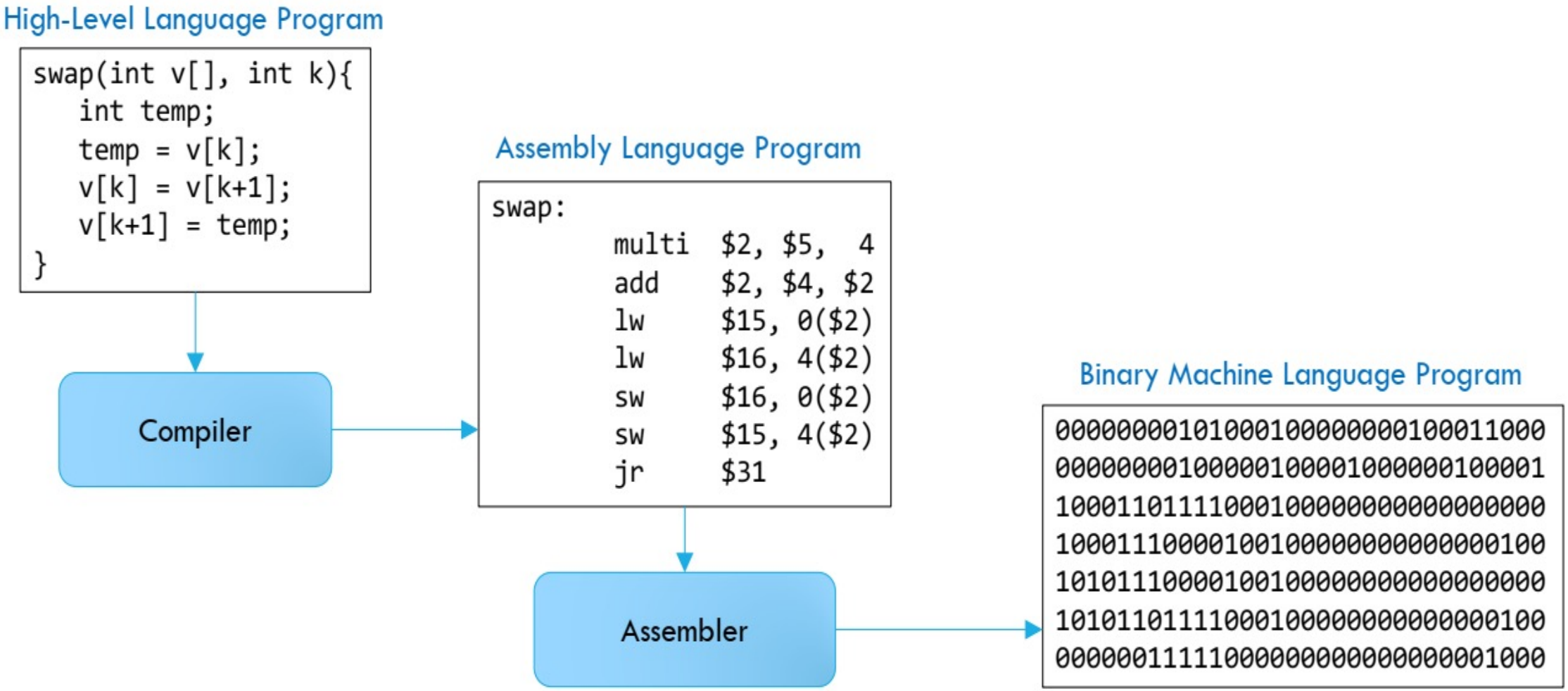
- **Level-4:** Assembly language is the next level of the computer system. The machine understands only the assembly language and hence in order, all the high-level languages are changed in the assembly language. Assembly code is written for it.
- **Level-5:** This level of the system contains high-level language. High-level language consists of C++, Java, FORTRAN, and many other languages. This is the language in which the user gives the command.
- **Level-6:** This the last level of the computer system hierarchy. This consists of users and executable programs.

- The stages of translation between these program levels are implemented by the following:
    - **Compiler**: high-level language to assembly language.
    - **Assembler**: assembly language to machine language.
    - **Linker**: combines multiple machine language files into a single executable that can be loaded into memory and executed.
- Benefits of this stages of transition (layers of abstraction)

    - Allows programmers to think in more natural terms

    - The most important advantage is portability. Programs are independent of the machine because compilers and assemblers can take a universal program and translate it for a particular machine

Example of translating a C program

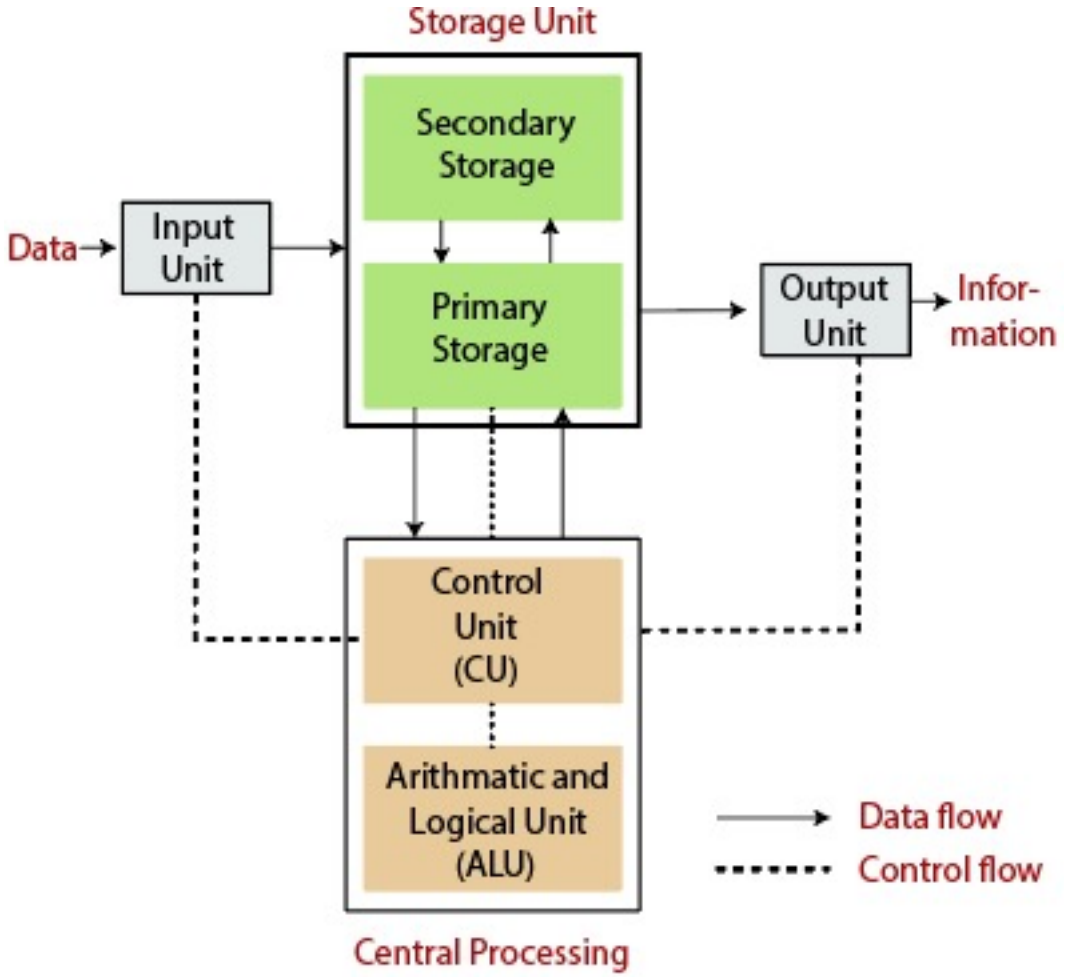


Difference between Assembly language and high level language

ASSEMBLY LEVEL LANGUAGE	HIGH-LEVEL LANGUAGE
•It needs an assembler for conversion	•It needs a compiler/interpreter for conversion
•In this, we convert an Assembly level language to machine level language	•In this, we convert a high-level language to Assembly level language to machine level language
•It is machine dependent	•It is machine-independent
•In this mnemonics, codes are used	•In this English statement is used
•It supports low-level operation	•It does not support low-level language
•In this, it is easy to access hardware component	•In this, it is difficult to access hardware component
•In this more compact code	•No compactness

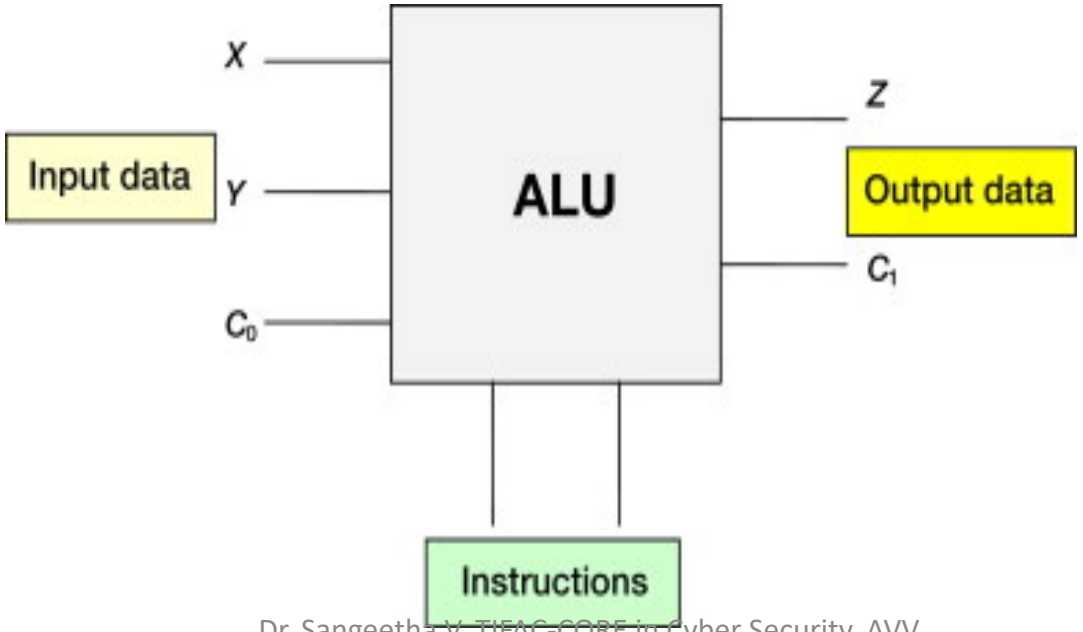
Functional Components of a computer

- There are basic components that aids the working-cycle of a computer i.e. the Input- Process- Output Cycle and these are called as the functional components of a computer.
- It needs certain input, processes that input and produces the desired output. The input unit takes the input, the central processing unit does the processing of data, and the output unit produces the output. The memory unit holds the data and instructions during the processing.

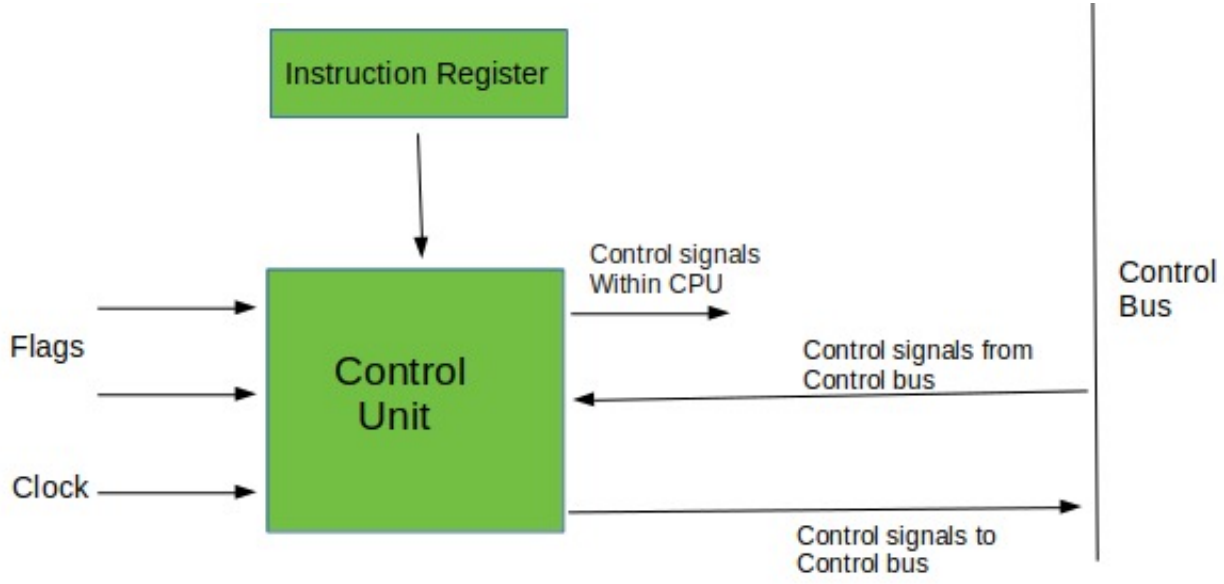


- **Input Unit :**The input unit consists of input devices that are attached to the computer. These devices take input and convert it into binary language that the computer understands. Some of the common input devices are keyboard, mouse, joystick, scanner etc.
- **Central Processing Unit (CPU) :** The CPU is called the brain of the computer because it is the control center of the computer. It first fetches instructions from memory and then interprets them so as to know what is to be done. If required, data is fetched from memory or input device. Thereafter CPU executes or performs the required computation and then either stores the output or displays on the output device. The CPU has three main components which are responsible for different functions – Arithmetic Logic Unit (ALU), Control Unit (CU) and Memory registers
- The speed of the CPU is measured in Hertz(MHz, GHz, etc). The various types of CPU chips are intel Celeron, intel core i3, i5, i7, etc.

- **Arithmetic and Logic Unit (ALU) :** The ALU, as its name suggests performs mathematical calculations and takes logical decisions. Arithmetic calculations include addition, subtraction, multiplication and division. Logical decisions involve comparison of two data items to see which one is larger or smaller or equal.



- **Control Unit :** The Control unit coordinates and controls the data flow in and out of CPU and also controls all the operations of ALU, memory registers and also input/output units. It is also responsible for carrying out all the instructions stored in the program. It decodes the fetched instruction, interprets it and sends control signals to input/output devices until the required operation is done properly by ALU and memory.



Block Diagram of the Control Unit

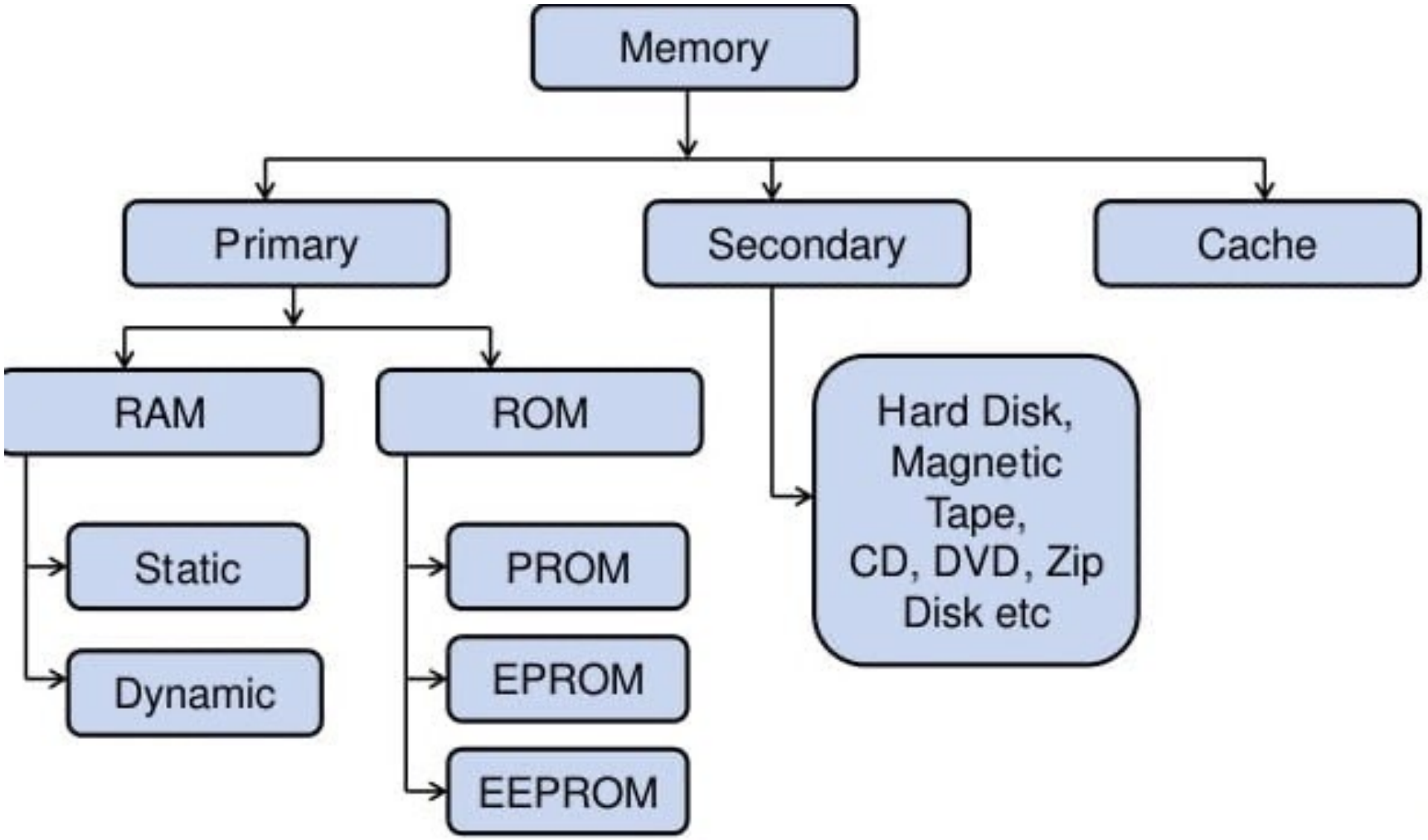
Memory Unit : The Memory unit can be referred to as the storage area in which programs are kept which are running, and that contains data needed by the running programs.

- The Memory unit can be categorized in two ways namely, primary memory and secondary memory.
- It enables a processor to access running execution applications and services that are temporarily stored in a specific memory location.
- Primary storage is the fastest memory that operates at electronic speeds. Primary memory contains a large number of semiconductor storage cells, capable of storing a bit of information. The word length of a computer is between 16-64 bits.
- It is also known as the volatile form of memory, means when the computer is shut down, anything contained in RAM is lost.

Word - natural unit of data used by a particular processor design. Word is nothing but a unit of data (bunch of bits (signal charges of zeros and ones)) that moves around from a computer component to another.
Word length refers to the number of bits processed by a computer's CPU in one go
Intel's x86 uses word size of 2 bytes = 16 bits.
16 bits = WORD, DWORD=32 bits, QWORD=64 bits

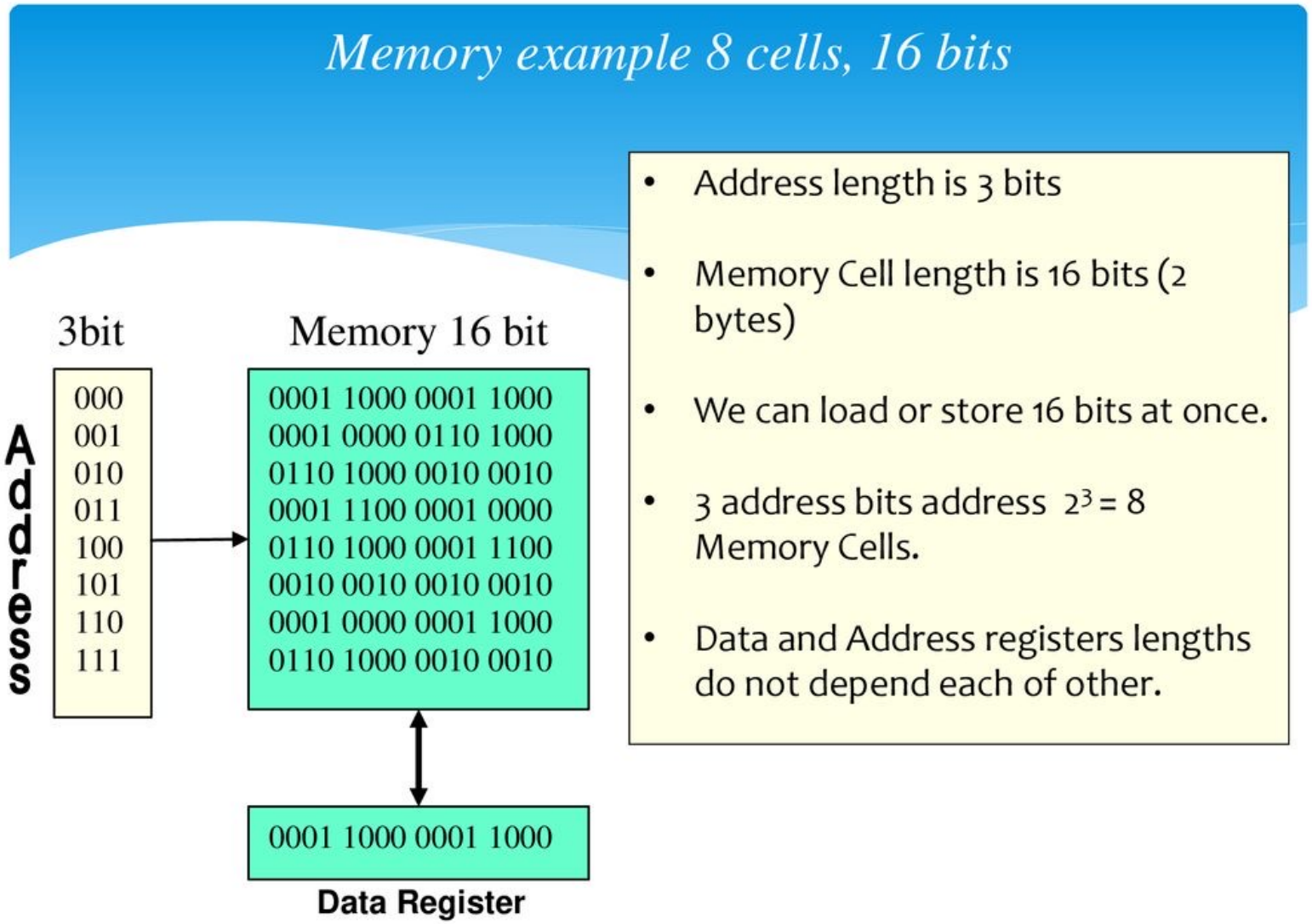
- Cache memory is also a kind of memory which is used to fetch the data very soon. They are highly coupled with the processor.
- The most common examples of primary memory are RAM and ROM.
- Secondary memory is used when a large amount of data and programs have to be stored for a long-term basis.
- It is also known as the Non-volatile memory form of memory, means the data is stored permanently irrespective of shut down.
- The most common examples of secondary memory are magnetic disks, magnetic tapes, optical disks, HDD, and other internal/external storage devices.

Types of memory

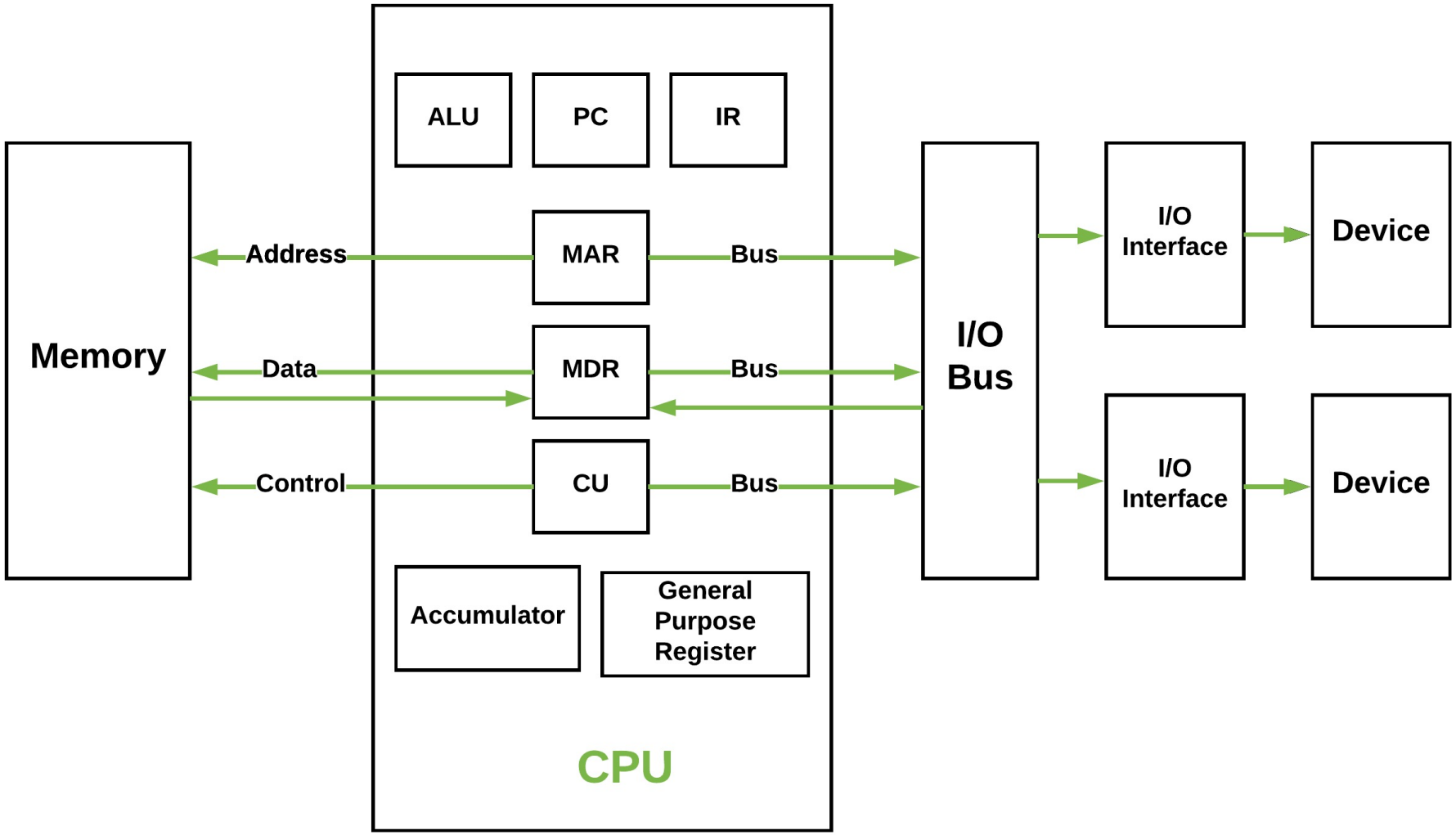


- **Memory Registers :** Memories are made up of registers. Each register in the memory is one storage location. The storage location is also called as memory location. Memory locations are identified using **Address**. The total number of bit a memory can store is its **capacity**.
- A storage element is called a **Cell**. Each register is made up of storage element in which one bit of data is stored. The data in a memory are stored and retrieved by the process called **writing** and **reading** respectively.
- Registers can be of different sizes(16 bit, 32 bit, 64 bit and so on) and each register inside the CPU has a specific function like storing data, storing an instruction, storing address of a location in memory etc. The user registers can be used by an assembly language programmer for storing operands, intermediate results etc. Accumulator (ACC) is the main register in the ALU and contains one of the operands of an operation to be performed in the ALU.

Example of memory unit



Basic CPU illustration



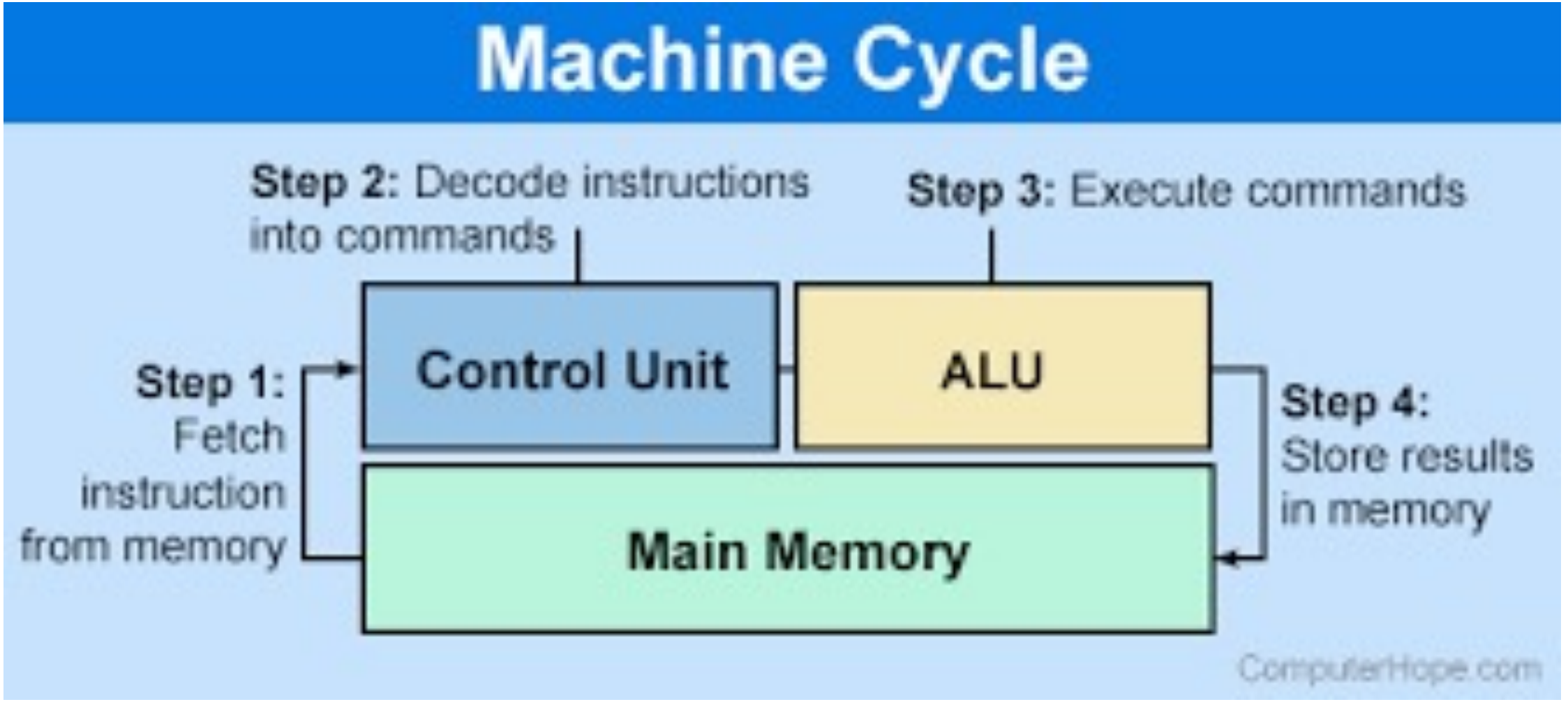
Registers –

- **Accumulator:** Stores the results of calculations made by ALU.
- **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).
- **Memory Address Register (MAR):** It stores the memory locations of instructions that need to be fetched from memory or stored into memory.
- **Memory Data Register (MDR):** It stores instructions fetched from memory or any data that is to be transferred to, and stored in, memory.
- **Instruction register (IR) :** holding the instruction currently being executed.

- 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.

- **General Purpose Register :** The general-purpose registers are used to store temporary data/address in the time of different operations in microprocessor.


Machine Cycle
Step 2: Decode instructions into commands
Step 3: Execute commands
Control Unit
ALU
Step 1: Fetch instruction from memory
Step 4: Store results in memory
Main Memory
ComputerHope.com

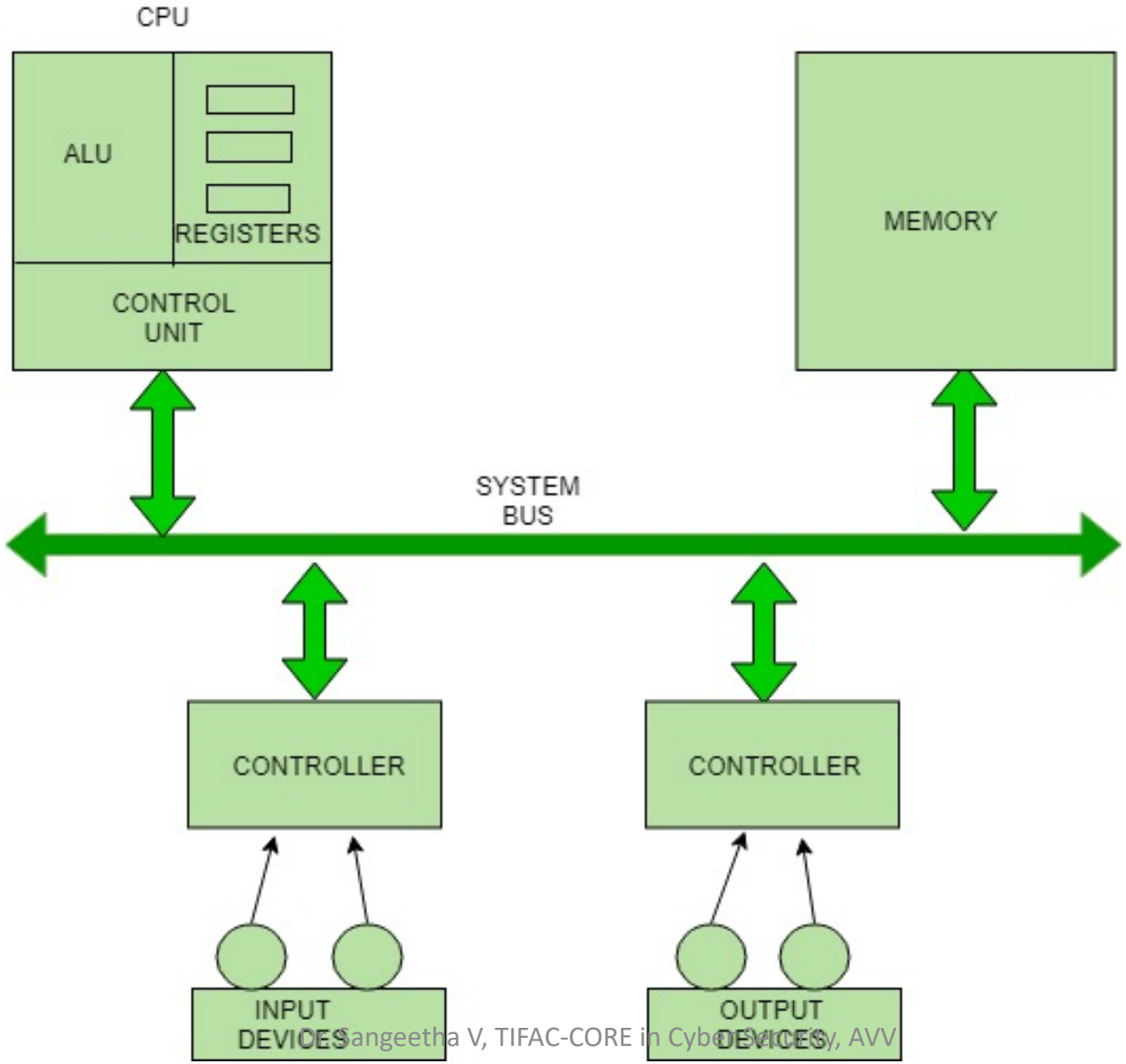
Output Unit :

- The primary function of the output unit is to send the processed results to the user. Output devices display information in a way that the user can understand.
- Output devices are pieces of equipment that are used to generate information, or any other response processed by the computer. These devices display information that has been held or generated within a computer.
- The most common example of an output device is a monitor.

Interconnection (Communication) between the units

- A computer consists of input unit that takes input, a CPU that processes the input and an output unit that produces output.
- All these devices communicate with each other through a common bus.
- A bus is a transmission path, made of a set of conducting wires over which data or information in the form of electric signals, is passed from one component to another in a computer.
- The bus can be of three types – Address bus, Data bus and Control Bus.

Connection of various functional components



- The address bus carries the address location of the data or instruction.
- The data bus carries data from one component to another and the control bus carries the control signals.
- The system bus is the common communication path that carries signals to/from CPU, main memory and input/output devices.
- The input/output devices communicate with the system bus through the controller circuit which helps in managing various input/output devices attached to the computer.

General System Architecture

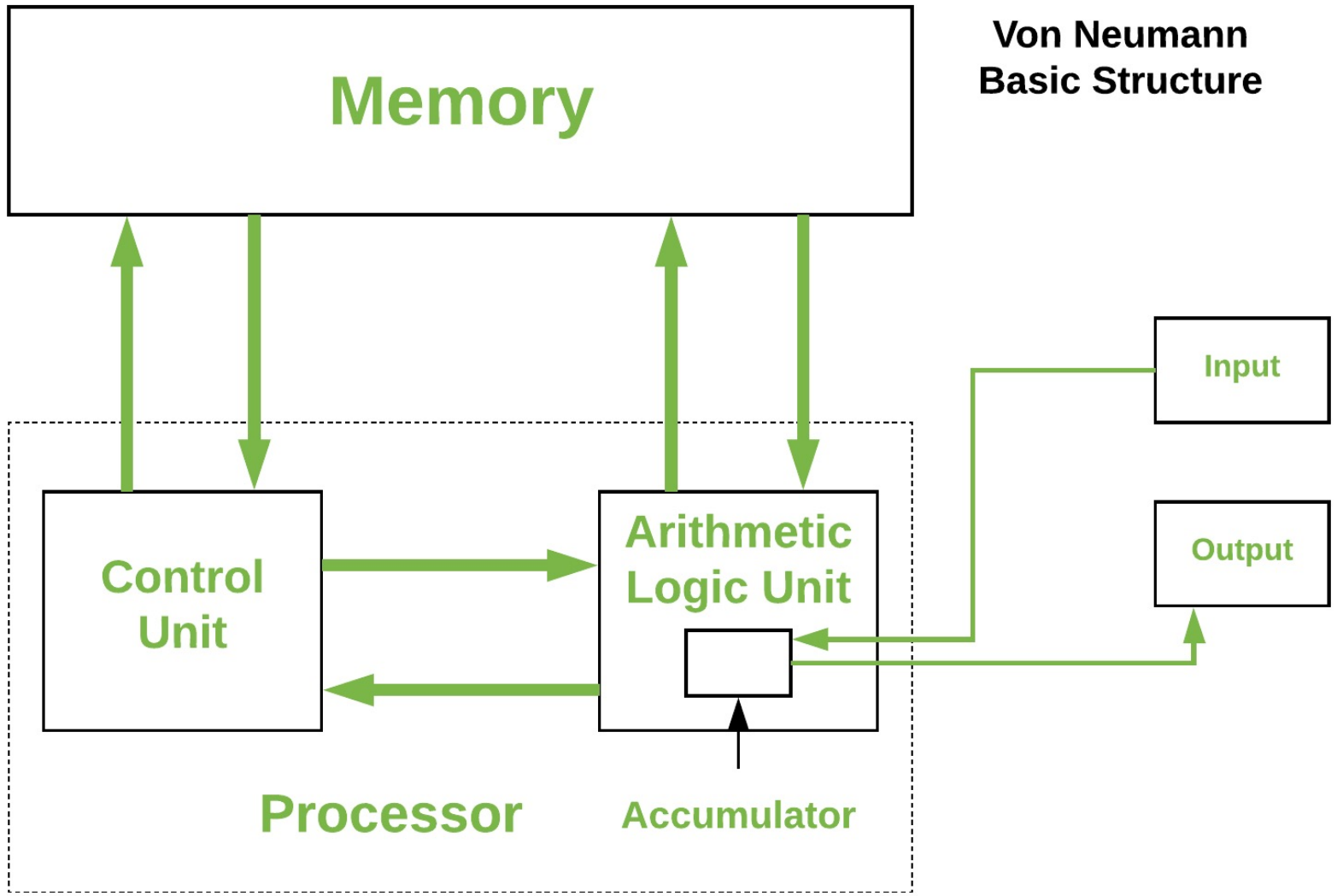
Based on COA, 2 classification of digital systems(computer)

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

The 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 a separate storage unit called memories and are treated the same. This novel idea meant that a computer built with this architecture would be much easier to reprogram.

Van Neumann Basic Structure



- **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'.

This architecture is very important and is used in our PCs and even in Super Computers.