

## Contents

<b>1 Computer Architecture and Organization</b>	<b>1</b>
1.1 Basics: . . . . .	1
1.1.1 Computer Architecture . . . . .	1
1.1.2 Computer Organization . . . . .	1
1.1.3 Functional Units . . . . .	1
1.2 Basics Of Computer Architecture . . . . .	3

## 1 Computer Architecture and Organization

### 1.1 Basics:

#### 1.1.1 Computer Architecture

Deals with the **functional behaviour** of Computer Systems.

**Design Implementation** for the various parts of the computer  
(*Designing*)

#### 1.1.2 Computer Organization

Deals with **structural relationship**

Operational attributes are linked together and contribute to realize the architectural specification  
(*Utilization*)

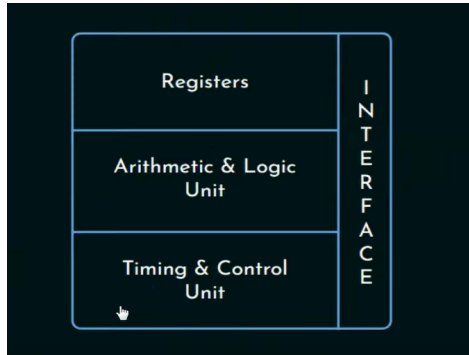
**Computer organization** refers to the operational units and their interconnections that realize the architectural specifications.

. Examples of architectural attributes include the instruction set, **the number of bits** used to represent various data types (e.g., **numbers, characters**), **I/O mechanisms**, and **techniques for addressing memory**. Organiza-

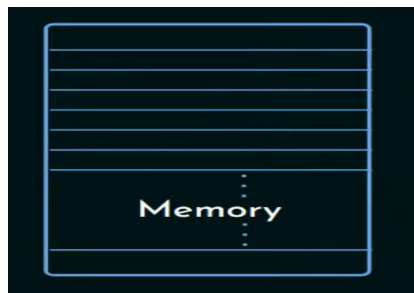
tional attributes include those hardware details transparent to the programmer, such as **control signals**; **interfaces between the computer and peripherals**; and the **memory technology** used.

#### 1.1.3 Functional Units

Processor: We can think of this as the brain of the system.

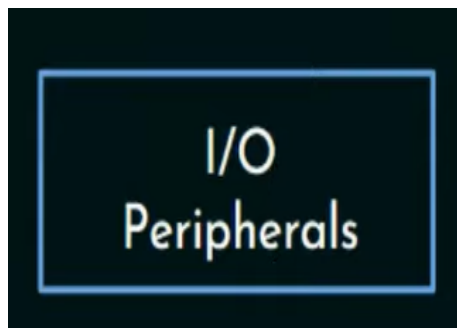


Memory: Stores all the instructions for the processor to work accordingly.  
It can also store data

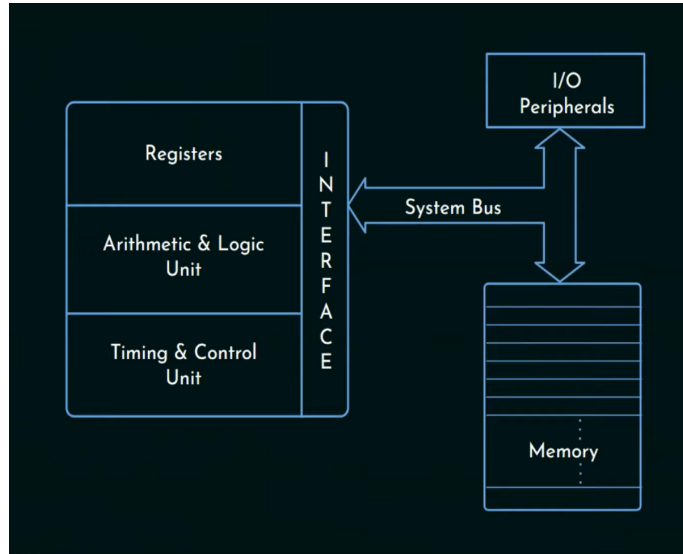


I/O Peripherals:

Input → Memory → Processor → (Memory) / Output



The intercommunication of all these functional units is carried out with the help of the **System Bus**  
Full picture:



## 1.2 Basics Of Computer Architecture

It is the **design of computers**, including their **instruction sets**, **hardware components** and **system organization**.

**Computer architecture** refers to those attributes of a system visible to a programmer or, put another way, those attributes that have a direct impact on the logical execution of a program.

It has two parts:

### 1. **Instruction Set Architecture**(ISA)

Specification to determine how machine language programs will interact with the computer

The ISA defines instruction formats, instruction opcodes, registers, instruction and data memory; the effect of executed instructions on the registers and memory; and an algorithm for controlling instruction execution.

### 2. **Hardware Set Architecture**(HSA)

Deals with the major computer subsystems like CPU, Storage, I/O etc. It includes both the logical design and data flow organization of the subsystems and hence determines their efficiency.

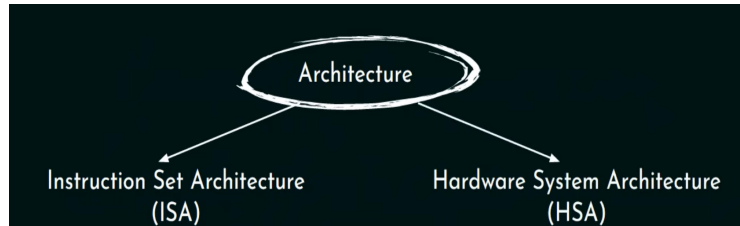


Illustration:

Suppose we want to add two values: (Example: 2 and 3) and store them in a variable ( $z$ )

$$z = 2 + 3$$