COMPUTER ARCHITECTURE

Introductions and Basics

By
Protik Chakroborty
Lecturer, CSE
Varendra University, Rajshahi

OVERVIEW

Course Title: Computer Architecture

Course Code: CSE 3105

Course Type: Theory

Credits: 3

Prerequisite Knowledge: Discrete Logic Design

Year and Semester: 3rd Year, Summer Semester, 2025

LEARNING OUTCOMES?

COs	Description	Taxonomy domain/level	POs	K	P	A
CO1	Illustrate hardware-software interaction to run the computer system.	Cognitive/ Understand	**PO1**	**K3, K4**		
CO2	Analyze various data communication between different components and processing unit.	Cognitive/ Understand	**PO2**	**K4**		
CO3	Discuss detail functionality and connection architecture of different types of memories leading to high performance computing.	Cognitive/ Analyze	**PO1**	**K3, K4**		

Knowledge Profile (K1–K8)

Code	Description
K1	Mathematics, science, engineering fundamentals
K2	Engineering specialization fundamentals
K3	Advanced engineering knowledge
K4	Research literature and methods
K5	Engineering design
K6	Engineering practices, tools, and resources
K7	Effects of engineering on society and environment
K8	Principles of project management and finance

WHAT IS A COMPUTER?

- ❑ A computer is a sophisticated electronic calculating machine that:
  - Accepts input information,
  - Processes the information according to a list of internally stored instructions and
  - Produces the resulting output information.

- ❑ Functions performed by a computer are:
  - Accepting information to be processed as input.
  - Storing a list of instructions to process the information.
  - Processing the information according to the list of instructions.
  - Providing the results of the processing as output.

INTRODUCTION

Consider three terms – Computer organization, computer architecture and computer design

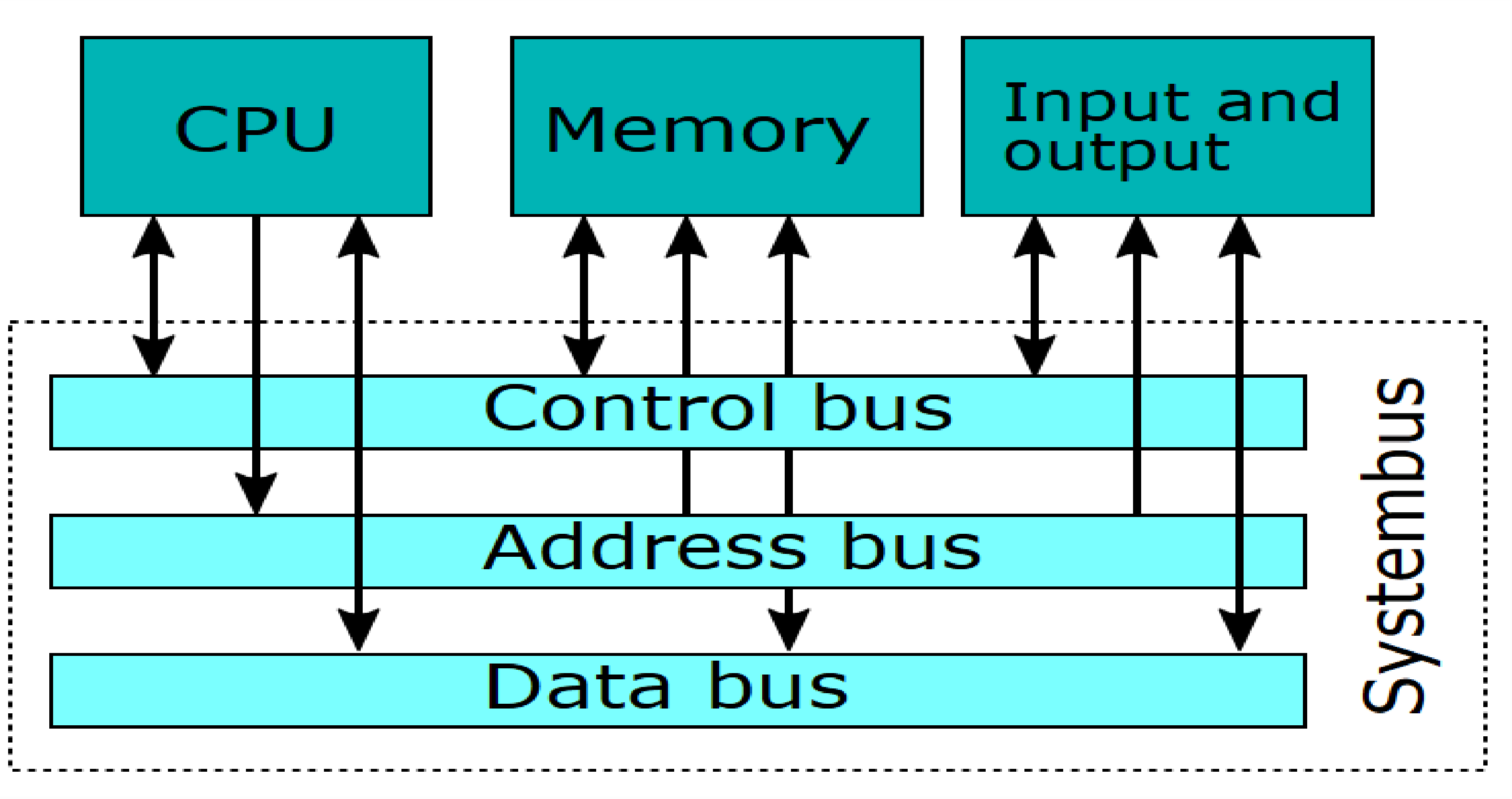
Computer Organization refers to the level of concept between digital logic level and OS.
The major components are functional units or subsystems that correspond to specific pieces of hardware built from the lower level building blocks.

Introduction

- ➢ Computer organization is concerned with the way the hardware components
- ➢ Operation and connection together to form the computer system.
- ➢ It is concerned with all physical aspects of computer systems e.g. circuit design, control signals, memory types.
- ➢ The various components are assumed to be placed for proper functioning.

- ○ **Scenario: Designing the interaction between the CPU and memory**.
- ○ Example:
- ❑ The CPU fetches instructions from memory.
- ❑ The memory controller ensures the data is transferred correctly.
- ❑ Components like buses, control signals, and registers are organized to ensure smooth operation.
- ○ **Key Focus:**
- ○ How the components (CPU, memory, I/O devices) are connected and how they operate together.
- ○ Example hardware: Data bus width, control unit operation, ALU (Arithmetic Logic Unit).

Introduction



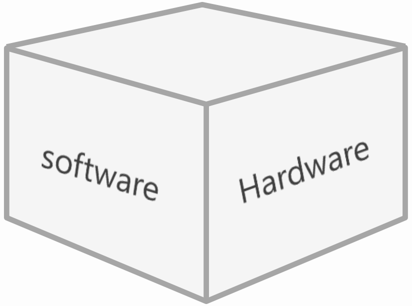
Introduction

- ➢ Computer design is concerned with the hardware design of the computer.
- ➢ Once the **computer specifications are formulated**, designer develops the hardware.
- ➢ Determine what hardware should be used and how the parts should be connected.
- ➢ Sometimes referred to as computer implementation.

- **Scenario: Building a specific CPU for a smartphone.**
- **Example:**
  - Deciding to use an ARM processor because it is energy-efficient.
  - Designing the hardware to include specific features like **low-power cores, integrated GPUs, and hardware for neural processing**.
  - Creating the circuits, determining which transistors and materials to use, and deciding how components will be connected.
- **Key Focus:**
  - Implementation of hardware components to meet specific design goals.
  - Example design: A processor with low power consumption for mobile devices.

What is Computer Architecture?

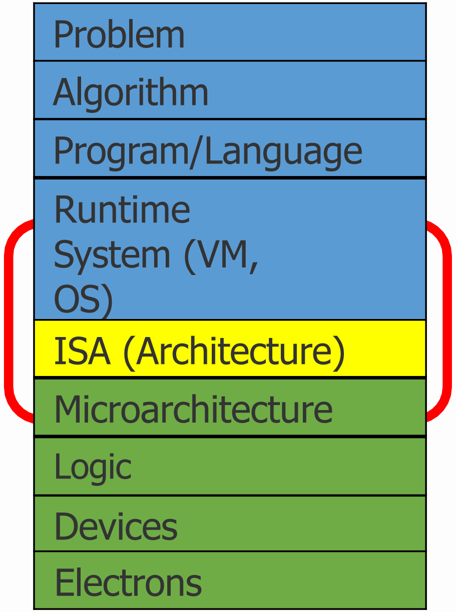
Computer Architecture is a higher-level concept that **defines the overall functionality and capabilities of the computer system**. It includes the **instruction set, addressing modes, and system performance**. It is concerned with how a computer is structured and behaves from the perspective of the user and software.



Performance Size Cost Energy

In other words, a system concept integrating software, hardware, and firmware to specify the design of computing systems.

COMPUTER ARCHITECTURE IN LEVELS OF TRANSFORMATION



Problem

Algorithm

Program/Language

Runtime System (VM, OS)

ISA (Architecture)

Microarchitecture

Logic

Devices

Electrons

ISA (Instruction Set Architecture) Interface/contract between SW and HW.

COMPUTER : ARCHITECTURE VS ORGANIZATION

Computer Architecture
is concerned with the structure and behavior of a computer system.

It helps us to understand the functionalities.

Architecture involves -
Processor and memory specification, information format, instruction sets, addressing techniques etc.

Computer Organization is concerned with the way hardware components are connected together to form a computer system.

It helps us to understand how exactly all the units in the system are arranged and interconnected.

Organization involves –
Physical Components (circuit design, control signals, memory circuit types)

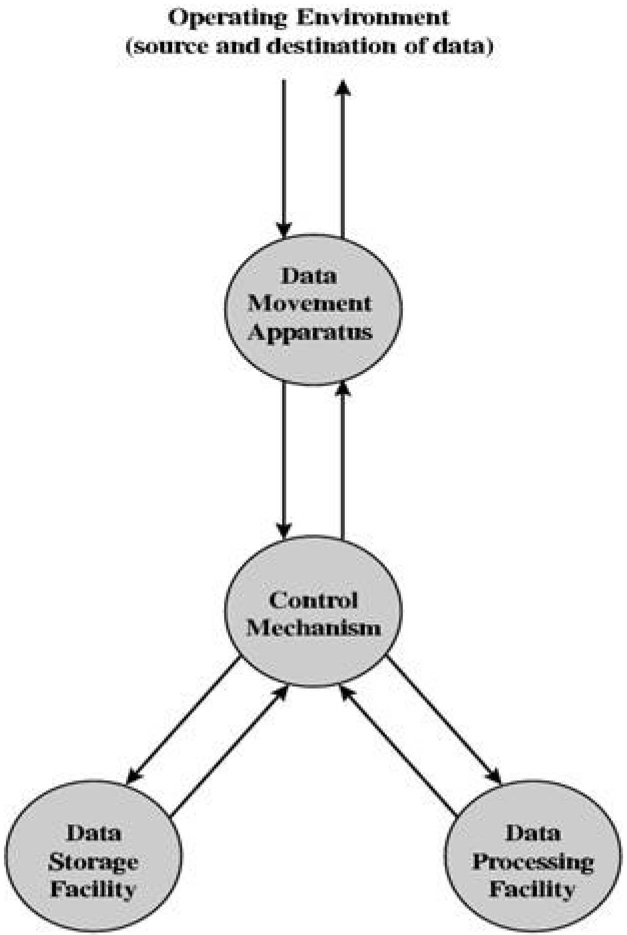
WHY STUDY COMPUTER ARCHITECTURE?

- Design better programs, including system software such as compilers, operating systems, and device drivers.

- Optimize program behavior.

- Evaluate (benchmark) computer system performance.

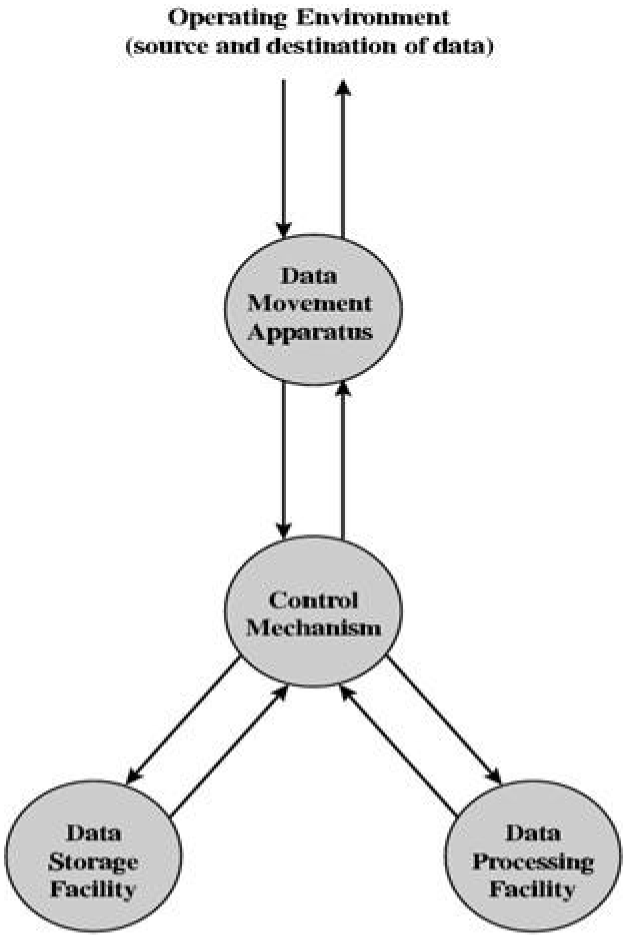
- Understand time, space, and price tradeoffs.



A Function View of The Computer

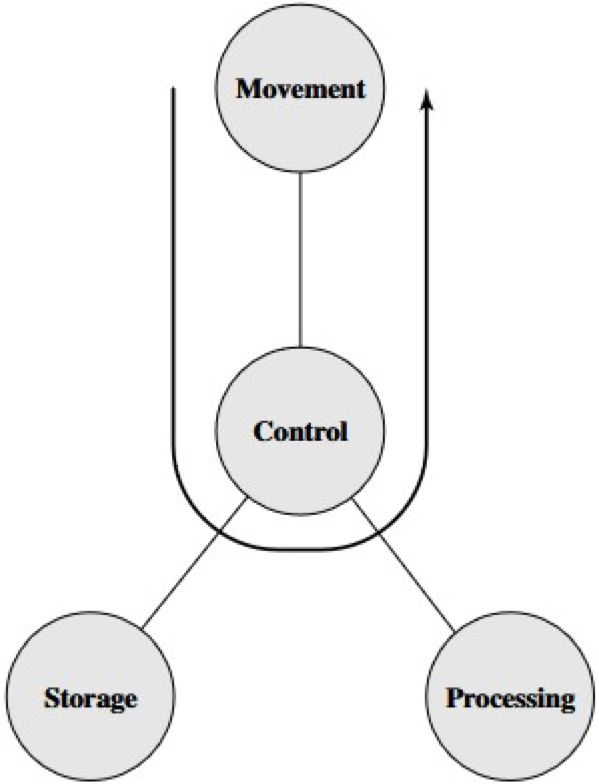
The computer, of course, must be able to process data. The data may take a wide variety of forms, and the range of processing requirements is broad.

It is also essential that a computer store data. Event if the computer is processing data on the fly (i.e., data come in and get processed, and the results go right out), the computer must temporarily store at least those pieces of data that are being worked on at any given moment. Thus, there is at least a short-term data storage function. Files of data are stored on the computer for subsequent retrieval and update.



The computer must be able to move data between itself and the outside world. The computer's operating environment consists of devices that serve as either sources or destinations of data. When data are received from or delivered to a device that is directly connected to the computer, the process id known as input-output (I/O), and the device is referred to as a peripheral. When data are moved over longer distances, to or from a remote device, the process is known as data communications.

Finally, there must be control of there three functions. Ultimately, this control is exercised by the individual who provides the computer with instructions. Within the computer system, a control unit manages the computer's resources and orchestrates the performance of its functional parts in response to those instructions.



(a)

Data Movement Device

What it does:

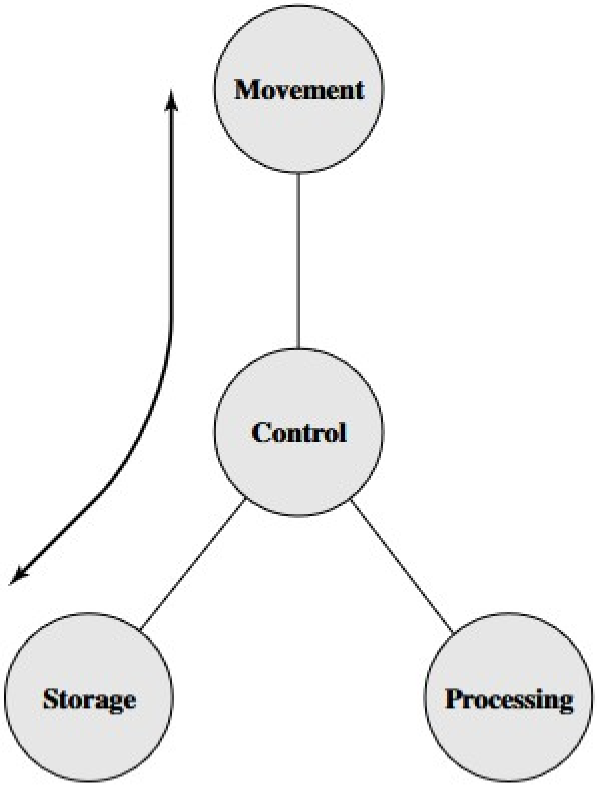
- ✓ The computer acts as a data movement device, simply transferring data from one location to another.
- ✓ The data moves between peripherals (e.g., keyboard, printer) or between communication lines (e.g., network connections).

Key Features:

- ✓ No data is stored or processed.
- ✓ The operation is purely about moving data from source to destination.

Example:

- ✓ Transferring a file from a USB drive to another device.
- ✓ Forwarding data packets across a network.



(b)

Data Storage Device

What it does:

The computer functions as a data storage device, enabling data to be:

Read: Data is transferred from storage to the external environment (e.g., reading a file from a disk to display on a screen).

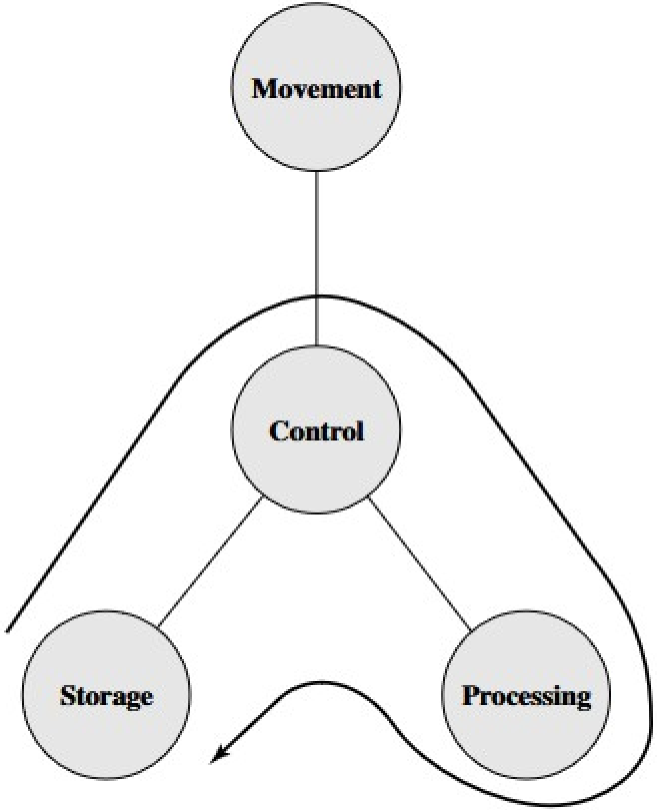
Written: Data is transferred from the external environment to storage (e.g., saving a file from an application to a hard drive).

Key Features:

- ✓ Involves data movement between the computer's storage and the external environment.
- ✓ No processing is performed; the focus is on reading and writing.

Example:

- ✓ Saving a scanned document to a hard drive.
- ✓ Reading a video file from a disk and sending it to a display.



(c)

Data Processing on Stored Data

What it does:
The computer processes data that is already stored in its memory or storage.

The operation involves:

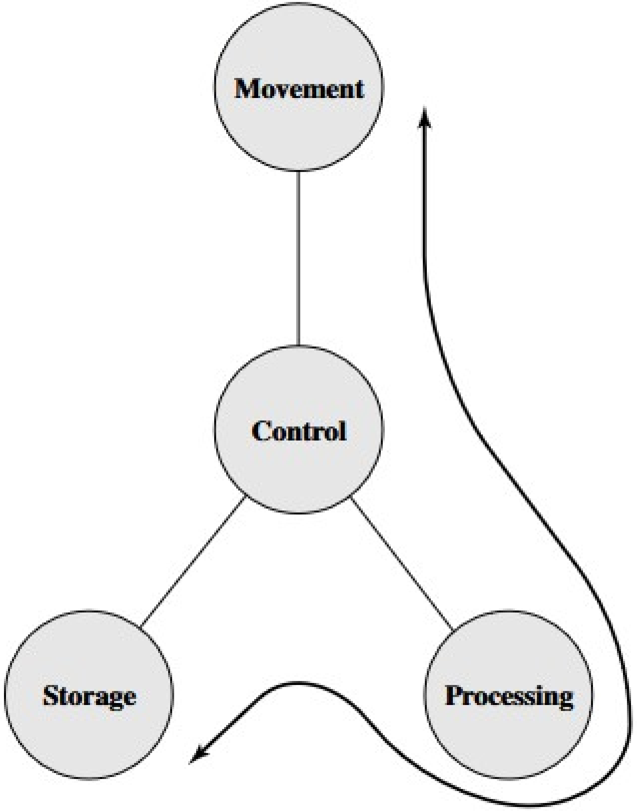
- ✓ Retrieving data from storage.
- ✓ Processing the data (e.g., performing calculations, transformations).
- ✓ Returning the processed data to storage or output.

Key Features:

- ✓ The processing unit (e.g., CPU) works on data fetched from storage.
- ✓ The results of processing may be saved back to storage.

Example:

- ✓ Running a program stored on disk to calculate payroll.
- ✓ Opening a spreadsheet, performing calculations, and saving the updated file.



(d)

Data Processing on En Route Data

What it does:
The computer processes data while it is being transferred between storage and the external environment.

The operation involves:

- ✓ Receiving data from an external source or storage.
- ✓ Processing the data during transfer.
- ✓ Sending the processed data to storage or output.

Key Features:

- ✓ Combines data movement and data processing.
- ✓ The processing occurs "on the fly," without storing the raw data first.

Example:

- ✓ Reading sensor data in real-time, processing it (e.g., filtering noise), and storing the processed data in a database.
- ✓ Receiving compressed data from a network, decompressing it, and saving it to disk.

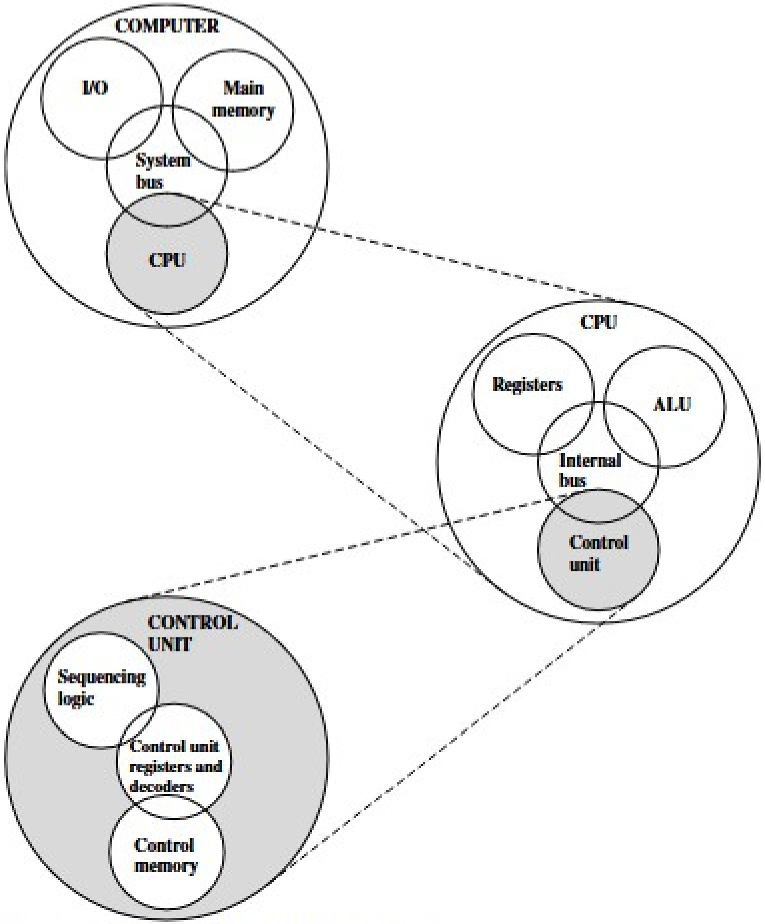


Figure 1.4 The Computer: Top-Level Structure

BASIC COMPONENTS

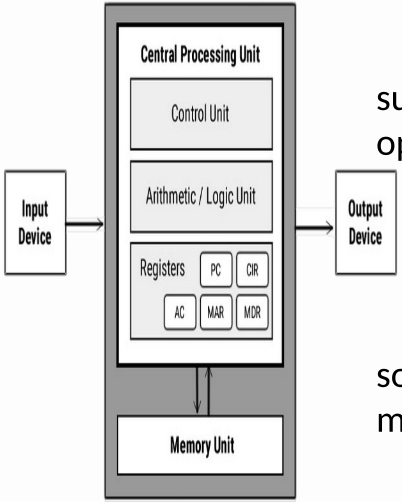
Central Processing Unit (CPU) Responsible for executing the instructions of a computer program. contains the ALU, CU and a variety of registers.

Control Unit (CU)

Controls the operation of the computer’s ALU, memory and input/output devices, telling them how to respond to the program instructions by providing the timing and control signals.

Bus

A bus is a physical group of signal lines which transfer electrical signals (thus data) between different parts of the computer system.



Arithmetic and Logic Unit (ALU)

The ALU allows arithmetic (add, subtract etc) and logic (AND, OR, NOT etc) operations to be carried out.

Memory Unit

The memory unit consists of RAM, sometimes referred to as primary or main memory. And directly accessible by the CPU.

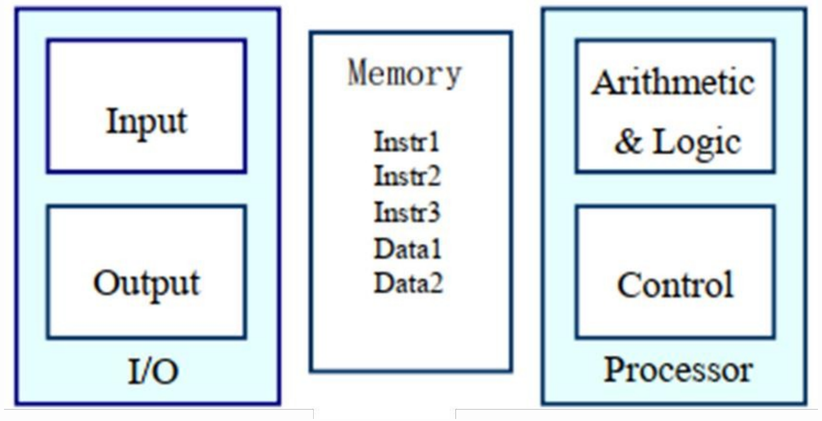
Registers

Registers are high speed storage areas in the CPU. All data must be stored in a register before it can be processed.

FUNCTIONAL UNITS OF A COMPUTER

Input unit accepts information:
Human operators.
Electromechanical devices
(keyboard) Other computers

Arithmetic and logic unit(ALU):
Performs the desired operations on the input information as determined by instructions in the memory



Output unit sends results of processing:
To a monitor display.
To a printer

Stores Information
Instructions
Data

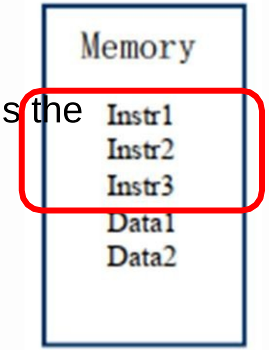
Control unit coordinates various actions:
Input
Output
Processing

Information in a Computer - Instructions

- Instructions specify commands to:
  - Transfer information within a computer (e.g., from memory to ALU)
  - Transfer of information between the computer and I/O devices(e.g., from keyboard to computer, or computer to printer)
  - Perform arithmetic and logic operations (e.g., Add two numbers, Perform a logical AND).

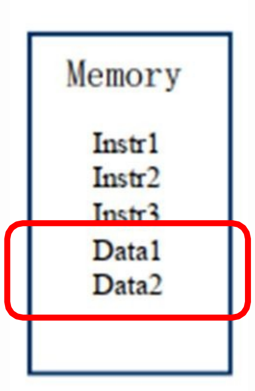
- A sequence of instructions to perform a task is called a program, which is stored in the memory.

- Processor fetches instructions that make up a program from the memory and performs the operations stated in those instructions.



INFORMATION IN A COMPUTER - DATA

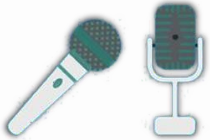
- ❑ Data are the "operands" upon which instructions operate. Data could be:
- ▪ Numbers.
- ▪ Encoded characters.
- ❑ Data, in a broad sense means any digital information.
- ❑ Computers use data that is encoded as a string of binary digits called bits



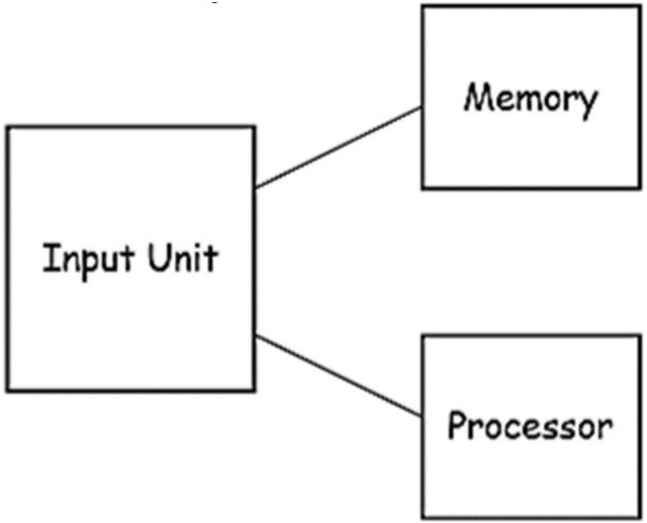
INPUT UNIT

- ❑ Binary information must be presented to a computer in a specific format. This task is performed by the input unit:
- ❑ Interfaces with input devices.
- ▪ Accepts binary information from the input devices.
- ▪ Presents this binary information in a format expected by the computer in specific binary form.
- ▪ Transfers this information to the memory or processor.

KEYBOARD



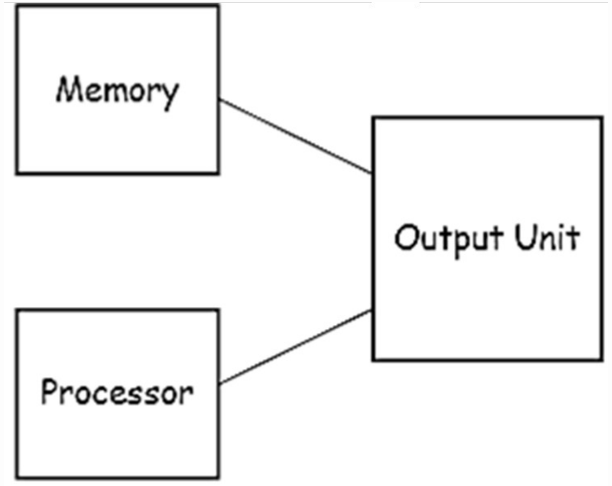
MICROPHONE



OUTPUT UNIT

❑ Computers represent information in a specific binary form via Output units:

- Interface with output devices.
- Accept processed results provided by the computer in specific binary form.
- Convert the information in binary form to a form understood by an output device.



Memory unit

- Memory unit stores instructions and data.
- Recall, data is represented as a series of bits.
- To store data, memory unit thus stores bits.
- Processor reads instructions and reads/writes data from/to the memory during the execution of a program.
- In theory, instructions and data could be fetched one bit at a time.
- In practice, a group of bits is fetched at a time.
- Group of bits stored or retrieved at a time is termed as "word"
- Number of bits in a word is termed as the "word length" of a computer.

- In order to read/write to and from memory, a processor should know where to look:
- "Address" is associated with each word location.

MEMORY UNIT

- ❑ Processor reads/writes to/from memory based on the memory address:
- ❑ Access any word location in a short and fixed amount of time based on the address.
- ❑ Random Access Memory (RAM) provides fixed access time independent of the location of the word.
- ❑ Access time is known as "Memory Access Time".
- ❑ Memory and processor have to communicate with each other in order to read/write information.
- ❑ In order to reduce communication time", a small amount of RAM (known as Cache) is tightly coupled with the processor.
- ❑ Modern computers have three to four levels of RAM units with different speeds and sizes:
- ❑ Fastest, smallest known as Cache
- ❑ Slowest, largest known as Main memory

Arithmetic and Logic Unit (ALU)

- ❑ Operations are executed in the Arithmetic and Logic Unit (ALU).
- ▪ Arithmetic operations such as addition, subtraction.
- ▪ Logic operations such as comparison of numbers.

- ❑ In order to execute an instruction, operands need to be brought into the ALU from the memory.
- ▪ Operands are stored in general purpose registers available in the ALU.
- ▪ Access times of general purpose registers are faster than the cache.
- ▪ Results of the operations are stored back in the memory or retained in the processor for immediate use.

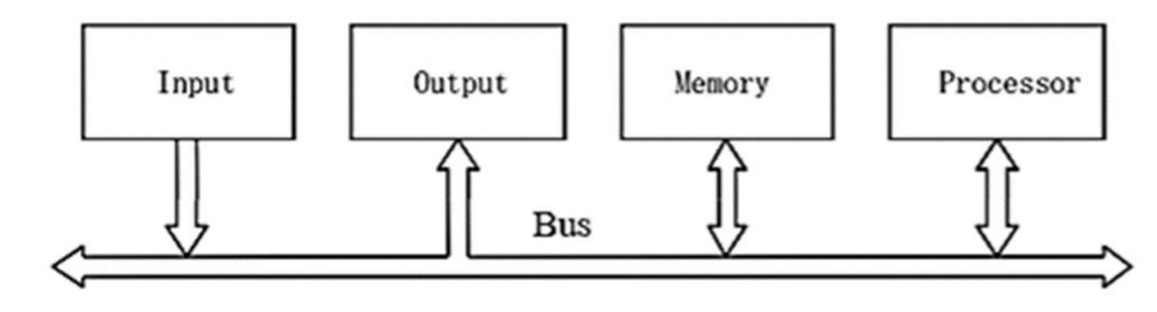
CONTROL UNIT

- ❑ Operation of a computer can be summarized as:
- ▪ Accepts information from the input units (Input unit).
- ▪ Stores the information (Memory).
- ▪ Processes the information (ALU).
- ▪ Provides processed results through the output units (Output unit)

- ❑ Operations of Input unit, Memory, ALU and Output unit are coordinated by Control unit.
- ▪ Instructions control "what" operations take place (e.g. data transfer, processing).
- ▪ Control unit generates timing signals which determines "when" a particular operation takes place.

HOW ARE THE FUNCTIONAL UNITS CONNECTED?

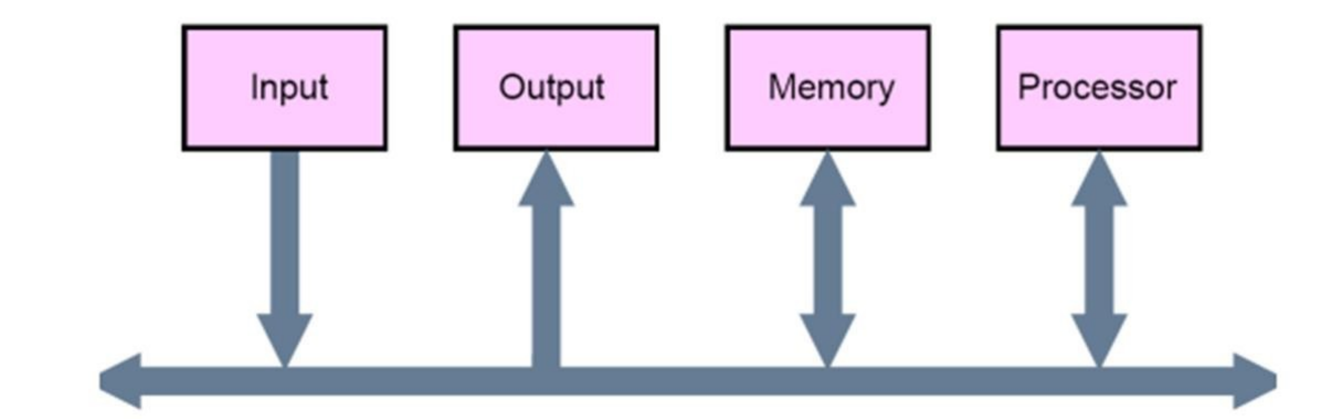
- For a computer to achieve its operation, the functional units need to communicate with each other.
- In order to communicate, they need to be connected.



- Functional units may be connected by a group of parallel wires.
- The group of parallel wires is called a bus.
- Each wire in a bus can transfer one bit of information
- The number of parallel wires in a bus is equal to the word length of a computer

BUS STRUCTURES

- ❑ A group of lines that serves a connecting path for several devices is called a bus
- ❑ In addition to the lines that carry the data, the bus must have lines for address and control purposes.
- ❑ The simplest way to interconnect functional units is to use a single bus, as shown below:



Performance Metrics

Latency: time to completely execute a certain task.

Throughput: amount of work that can be done over a period of time.

Power: instantaneous power during execution of a program.

Energy: Total energy consumption during the execution of the whole program.

Reliability: Failure rate.

CPI = Clock Cycle Per Instruction

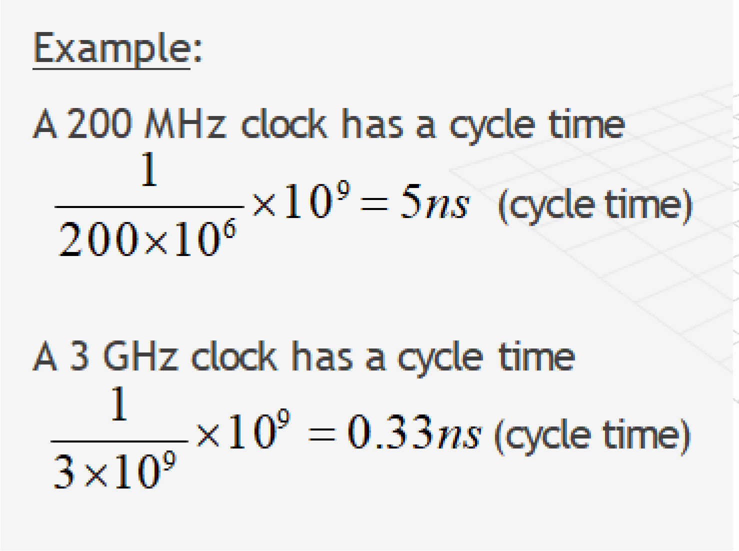
MIPS = Million Instructions Per Second

Transactions/minute, Transactions/hour, MIPS/watt etc

MEASURE OF PERFORMANCE:

cycle time=time between ticks= seconds/cycle

clock rate (frequency)=cycles per second (1 Hz. = 1 cycle/sec)



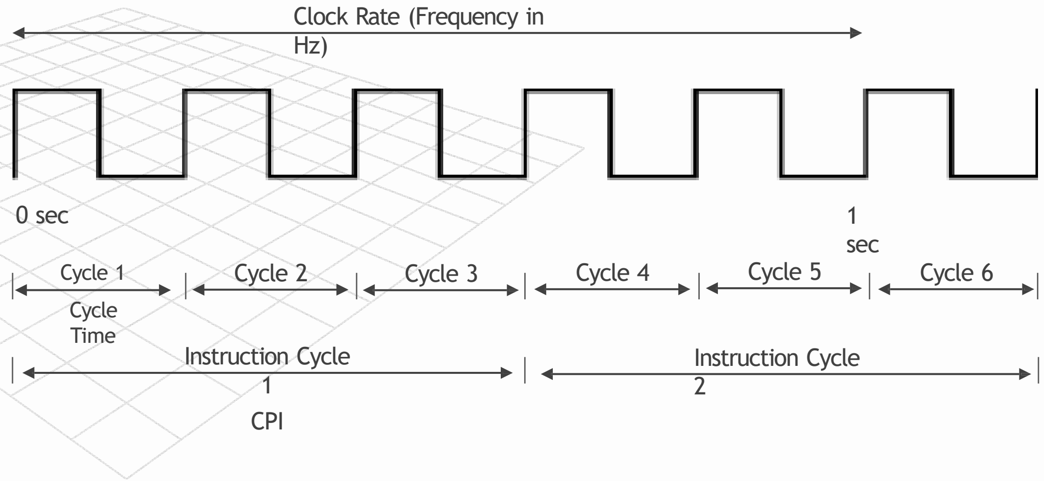
Example:

A 200 MHz clock has a cycle time

$$\frac{1}{200\times10^{6}}\times10^{9}=5ns \quad \text{(cycle time)}$$

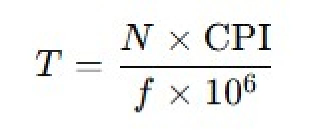
A 3 GHz clock has a cycle time

$$\frac{1}{3\times10^{9}}\times10^{9}=0.33ns \text{ (cycle time)}$$



EXECUTION TIME FORMULA

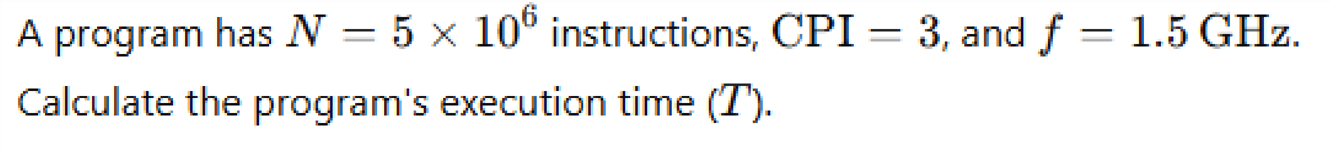
Program Execution Time Formula:



$$T = \frac{N \times \text{CPI}}{f \times 10^6}$$

T: Execution time (in seconds).
N: Total number of instructions.
CPI: Average clock cycles per instruction.
f: Clock frequency (in MHz).

Problem 01:



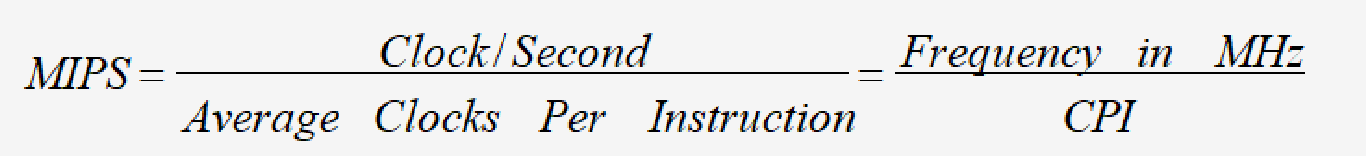
A program has $N = 5 \times 10^6$ instructions, $\text{CPI} = 3$, and $f = 1.5\,\text{GHz}$.

Calculate the program's execution time (T).

MIPS (Million Instructions Per Second):

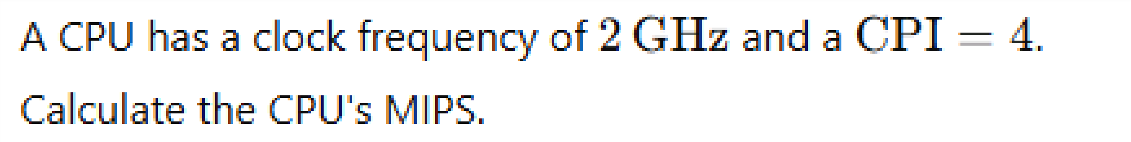
Measures how many instructions a CPU executes per second.

Formula:



$$MIPS = \frac{Clock/Second}{Average \quad Clocks \quad Per \quad Instruction} = \frac{Frequency \quad in \quad MHz}{CPI}$$

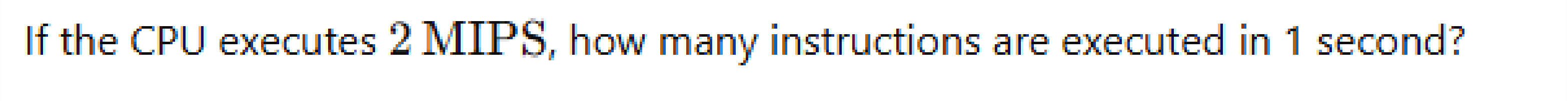
Problem 01:



A CPU has a clock frequency of $2\ \mathrm{GHz}$ and a $\mathrm{CPI} = 4$.

Calculate the CPU's MIPS.

Problem 02:



If the CPU executes $2\ \mathrm{MIPS}$, how many instructions are executed in 1 second?

Example 03:

Machine A has a clock cycle time of 10 ns. and a CPI of 2.0

Machine B has a clock cycle time of 20 ns. and a CPI of 1.2 Which machine is faster for this program, and by how much?

FACTORS AFFECTING PERFORMANCE

I. **Software:** The efficiency with which the programs are written and compiled into object code influences N, the number of instructions executed. Other factors being equal, reducing N tends to reduce the overall execution time T.

II. **Architecture:** The efficiency with which individual instructions are processed directly affects CPI, the number of cycles per instruction executed. Reducing CPI also tends to reduce T.

III. **Hardware:** The raw speed of the processor circuits determines f, the clock frequency. Increasing f tends to reduce T.

In general, the complex instruction sets of CISC processors aim to reduce N at the expense of CPI, whereas RISC processors aim to reduce CPI at the expense of N. Advances in VLSI technology affecting all types of computers tend to increase f.

SPEEDUP TECHNIQUES

❑**Cache Memory**:

- **Objective**: Provide faster access to instructions and data.
- **Description**: Cache is a smaller, faster memory closer to the CPU than main memory. It reduces the time required to fetch frequently used instructions or data.

❑**Pipelining**:

- **Objective**: Increase performance by allowing multiple stages of instruction processing to overlap.
- **Description**: The CPU divides instruction execution into stages (e.g., fetch, decode, execute). While one instruction is being executed, the next can be fetched or decoded simultaneously.

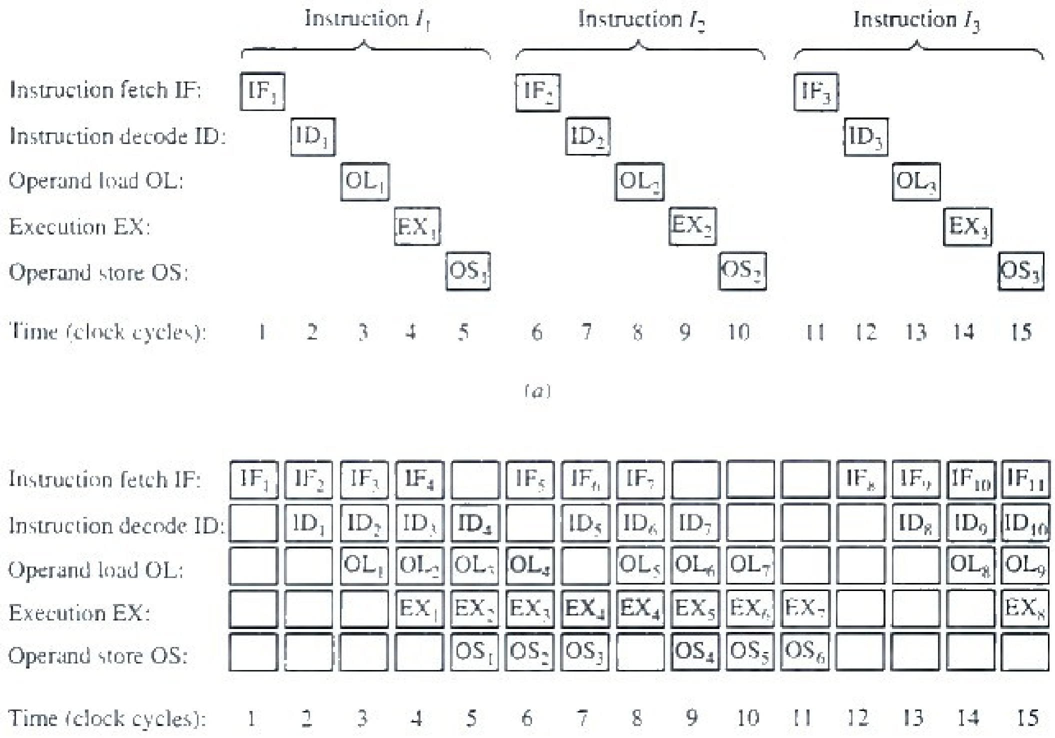
❑**Superscalar Processing**:

- **Objective**: Increase performance by processing multiple instructions simultaneously.
- **Description**: Multiple pipelines are used to execute several instructions in parallel, enabling better utilization of CPU resources.

PIPELINING IN DETAIL

Stages of Instruction Execution in a Pipeline:

I. **Instruction Fetch (IF):** Fetch the instruction from memory.
II. **Instruction Decode (ID):** Decode the fetched instruction.
III. **Operand Load (OL):** Load operands required for the instruction.
IV. **Execution (EX):** Perform the operation.
V. **Operand Store (OS):** Store the result back into memory.



	Instruction I_1					Instruction I_2					Instruction I_3				
Instruction fetch IF:	IF_1					IF_2					IF_3				
Instruction decode ID:		ID_1					ID_2					ID_3			
Operand load OL:			OL_1					OL_2					OL_3		
Execution EX:				EX_1					EX_2					EX_3	
Operand store OS:					OS_1					OS_2					OS_3
Time (clock cycles):	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

(a)

Instruction fetch IF:	IF_1	IF_2	IF_3	IF_4		IF_5	IF_6	IF_7				IF_8	IF_9	IF_{10}	IF_{11}
Instruction decode ID:		ID_1	ID_2	ID_3	ID_4		ID_5	ID_6	ID_7				ID_8	ID_9	ID_{10}
Operand load OL:			OL_1	OL_2	OL_3	OL_4		OL_5	OL_6	OL_7				OL_8	OL_9
Execution EX:				EX_1	EX_2	EX_3	EX_4	EX_4	EX_5	EX_6	EX_7				EX_8
Operand store OS:					OS_1	OS_2	OS_3		OS_4	OS_5	OS_6				
Time (clock cycles):	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Figure 1.24

Instruction processing: (*a*) sequential or nonpipelined and (*b*) pipelined.

In a **non-pipelined processor**, each instruction completes all stages before the next one begins.

In a pipelined processor, multiple instructions are in different stages simultaneously:

At clock cycle 1, instruction I1 is in the fetch stage.

At clock cycle 2, I1 moves to the decode stage while I2 starts fetching.

This overlap continues, improving throughput.

SUPERSCALAR ARCHITECTURE

Superscalar architecture is a type of CPU design that allows a processor to execute **multiple instructions simultaneously** during a single clock cycle. It achieves this by having multiple execution units, such as arithmetic units, floating-point units, and memory access units, operating in parallel.

In other words, the instructions can be completely overlapped. CPUs with this capability are said to be superscalar. (Note that two instructions in the same pipeline must be issued sequentially rather than in parallel.) For example, if the logic needed for the IF, ID, OL, EX, and OS steps is duplicated (with or without pipelining), then two instructions can be issued simultaneously. However, if the instructions are not independent, for example, if they share the same operands or one takes as input a result computed by the other, then delays not unlike those illustrated in Figure 1.24b can occur. Pipelining and superscalar design are both instances of instruction-level parallelism. The logic circuits needed to deal with parallelism of this kind add considerable complexity to the CPU's program control and execution units.