IFT 307

Mr. Ibrahim Lawal

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.

Examples of Architectural attributes include:

- instruction set,
- number of bits used to represent various data types (e.g., numbers, characters),
- I/O mechanisms,
- techniques for addressing memory.

Computer organization

Refers to the operational units and their interconnections that realize the architectural specifications.

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

- A computer is a complex system; contemporary computers contain millions of elementary electronic components.
- How, then, can one clearly describe them?
- The key is to recognize the hierarchical nature of most complex systems.

A hierarchical system is a set of interrelated subsystems, each of the latter, in turn, hierarchical in structure until we reach some lowest level of elementary subsystem.

- The hierarchical nature of complex systems is essential to both their design and their description.
- The designer need only deal with a particular level of the system at a time.
- At each level, the system consists of a set of components and their interrelationships.

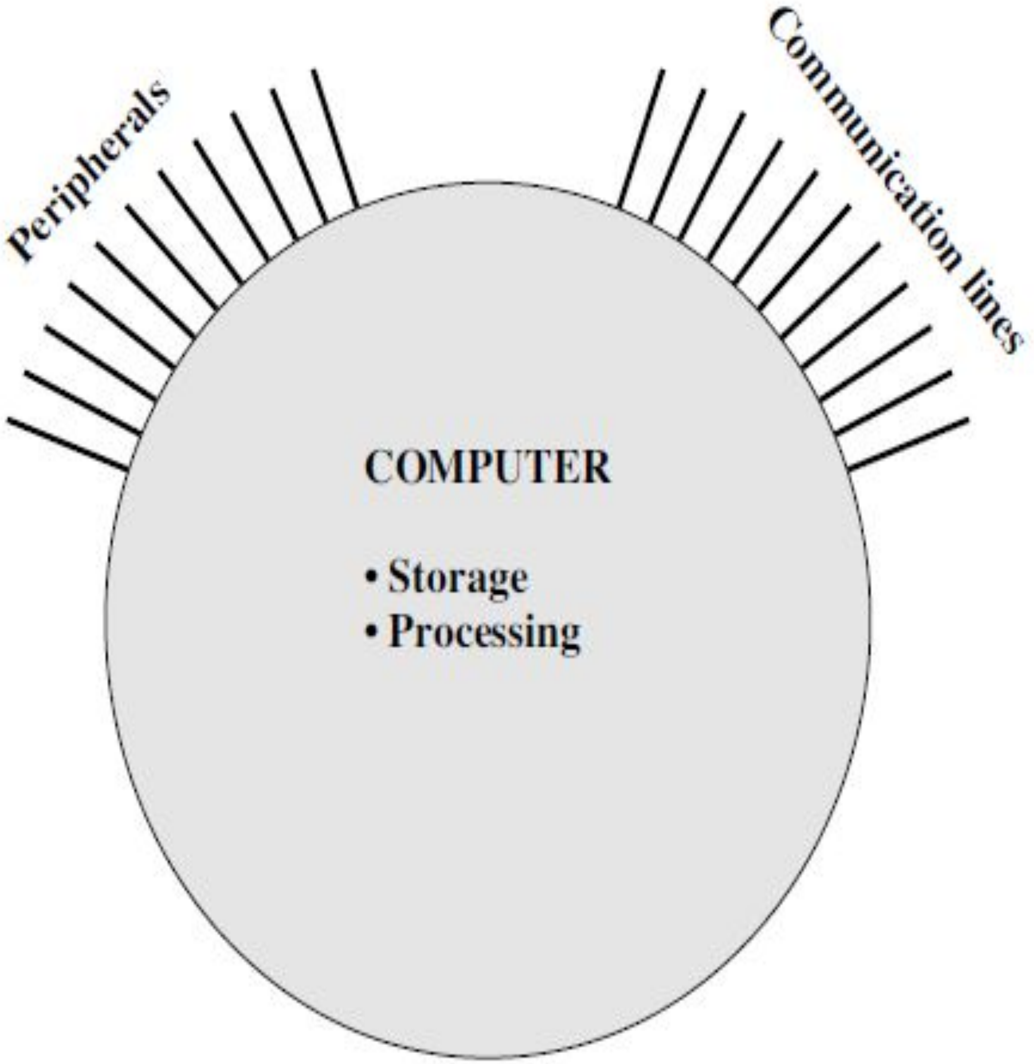
- The behavior at each level depends only on a simplified, abstracted characterization of the system at the next lower level.

- At each level, the designer is concerned with structure and function:

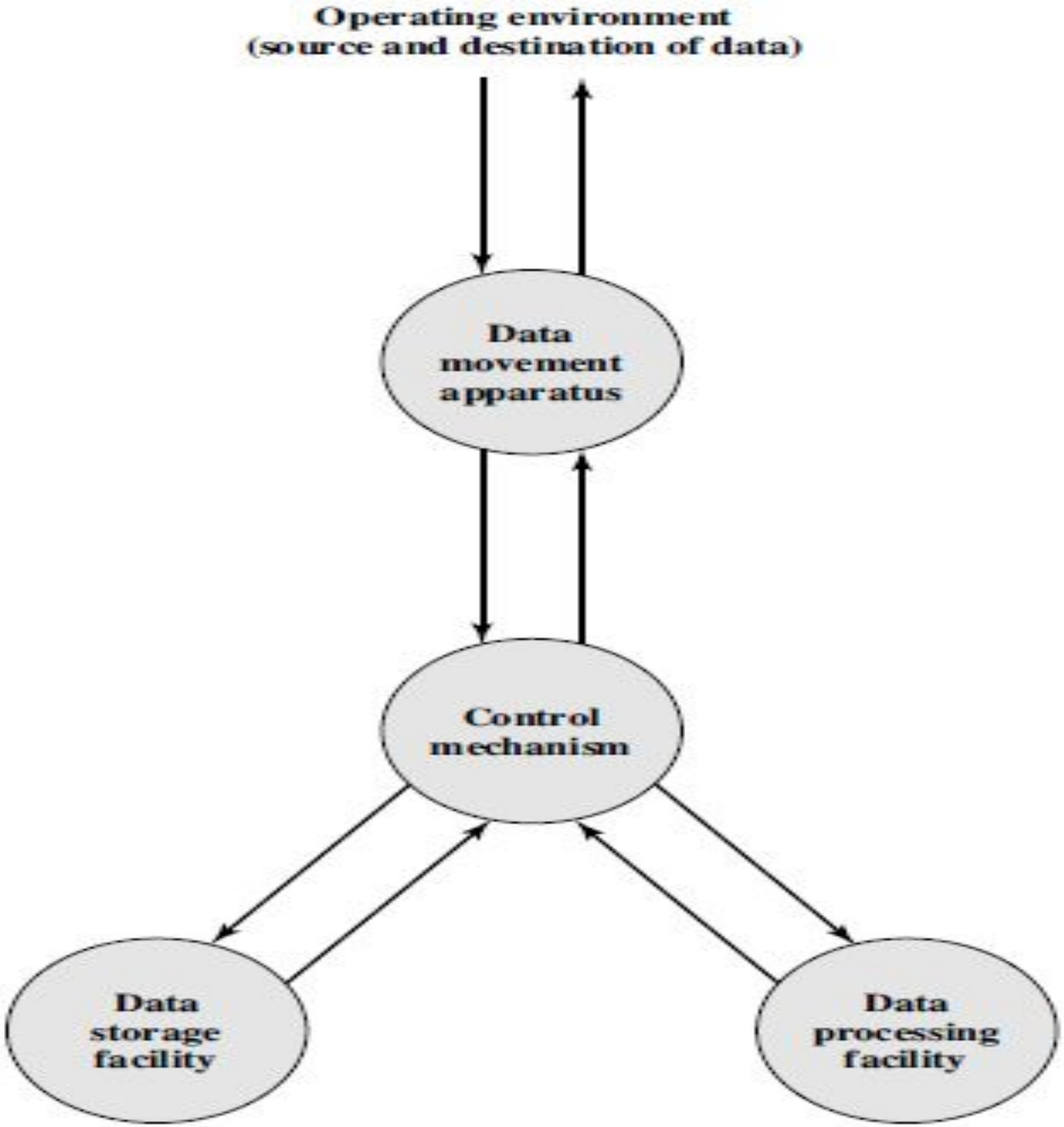
- **Structure:** The way in which the components are interrelated
- **Function:** The operation of each individual component as part of the structure.

The computer can function as:
A data movement device, simply transferring data from one peripheral or communications line to another.

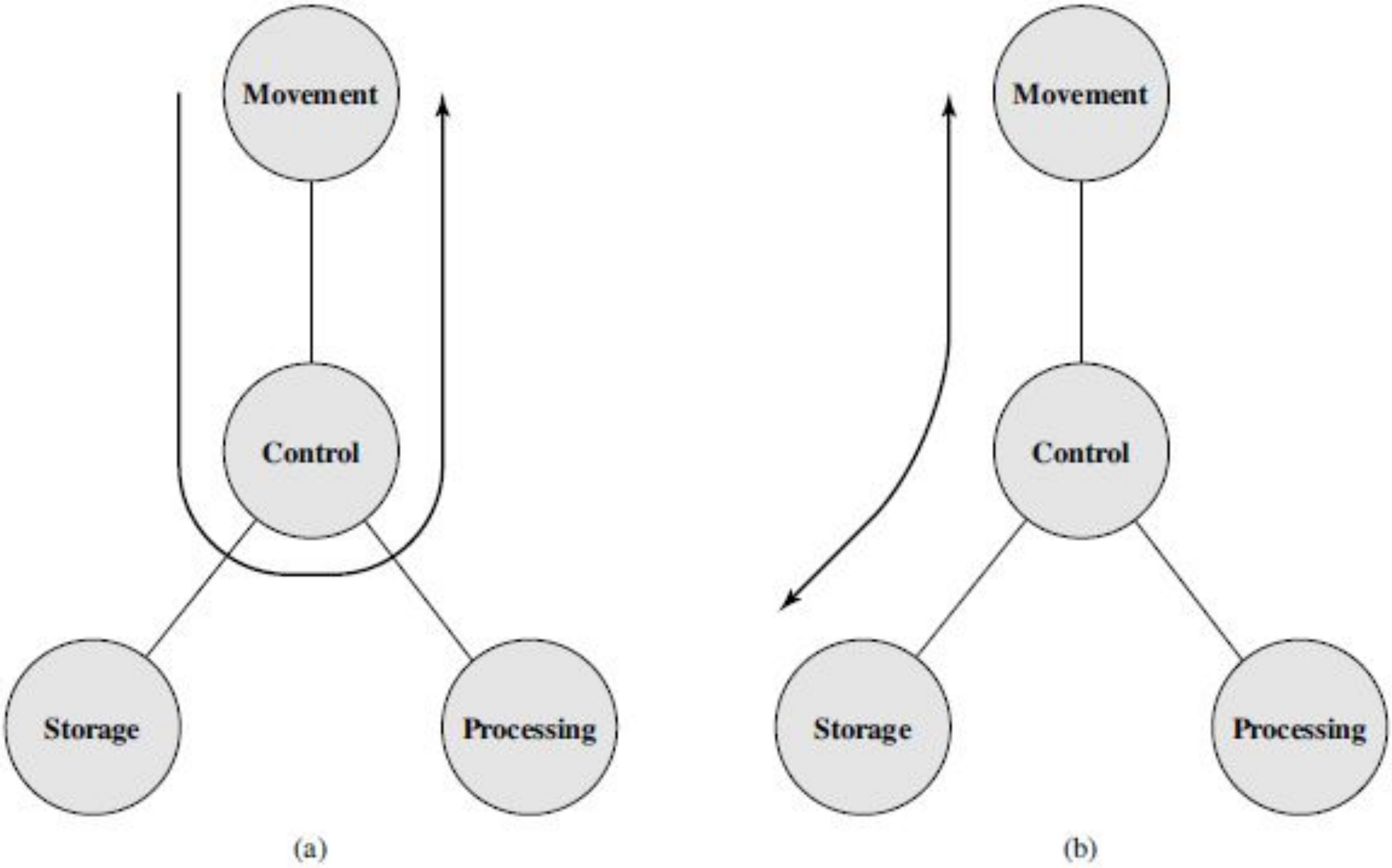
A data storage device, with data transferred from the external environment to computer storage (read) and vice versa (write).

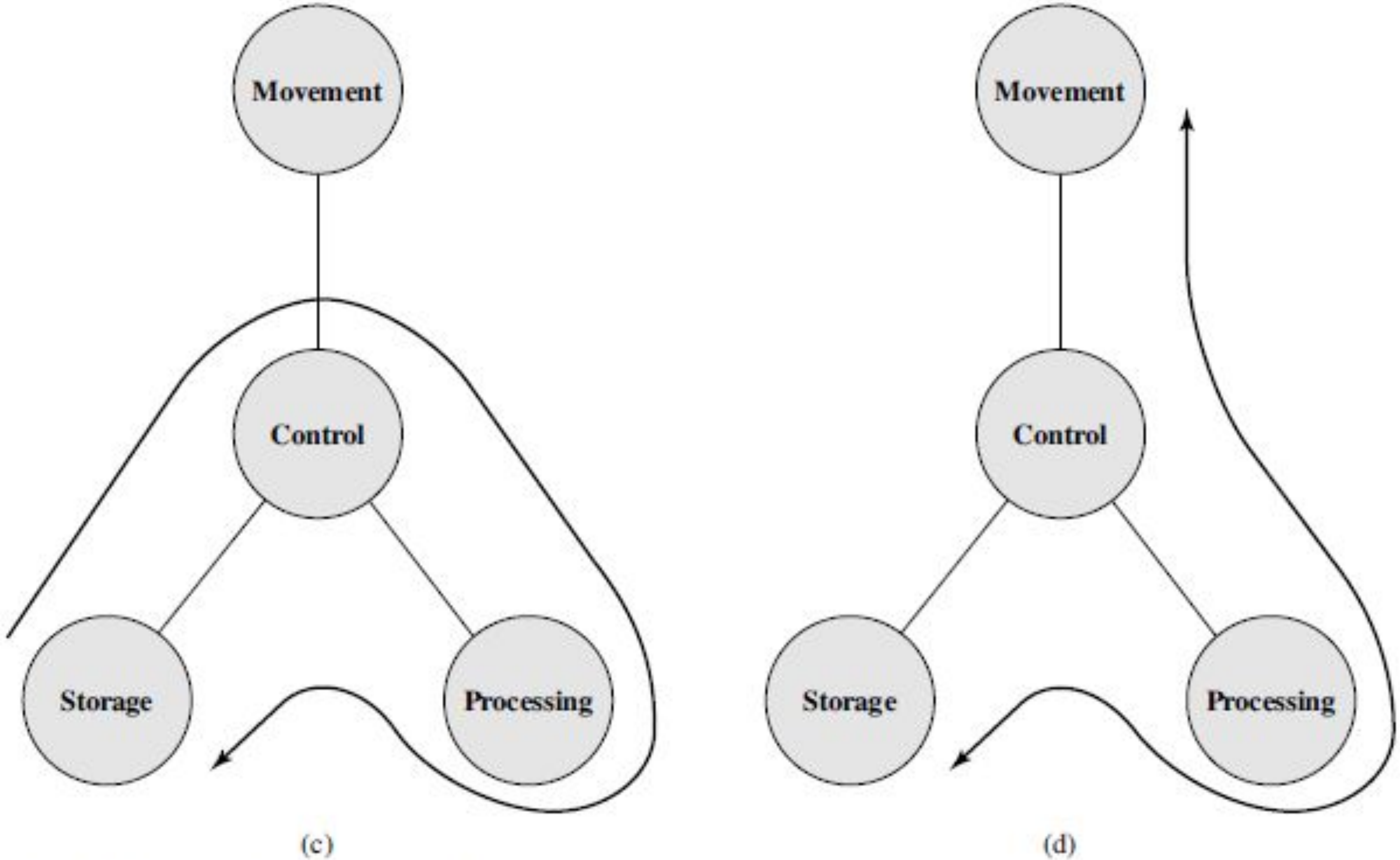


The Computer



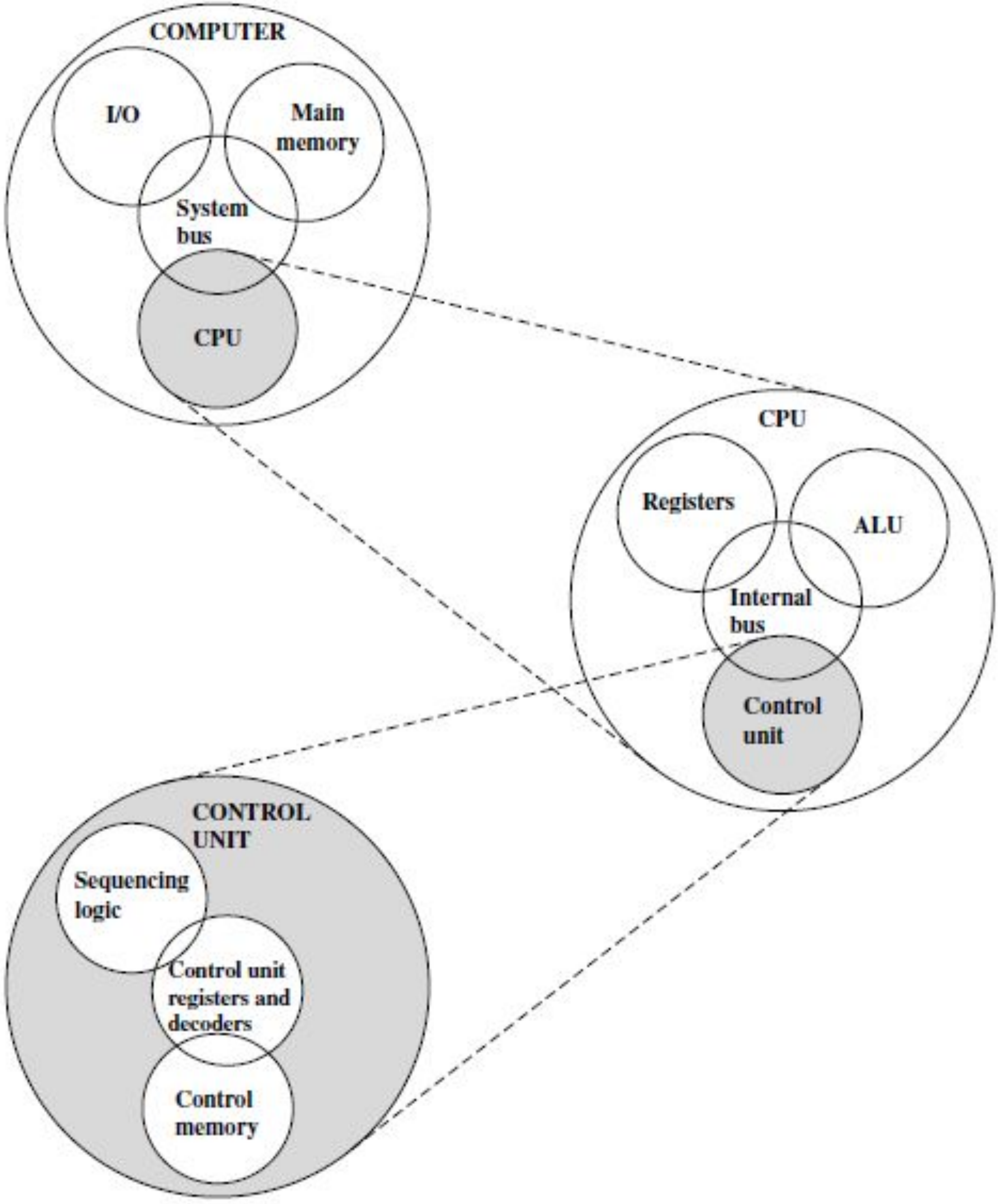
A Functional View of the Computer


Movement
Control
Storage
Processing
(a)
Movement
Control
Storage
Processing
(b)



Possible Computer Operations

Structure



The Computer: Top-Level Structure

There are four main structural components:

- **Central processing unit (CPU):** Controls the operation of the computer and performs its data processing functions; often simply referred to as **processor**.
- **Main memory:** Stores data.
- **I/O:** Moves data between the computer and its external environment.
- **System interconnection:** Some mechanism that provides for communication among CPU, main memory, and I/O. A common example of system interconnection is by means of a **system bus**, consisting of a number of conducting wires to which all the other components attach.

PERFORMANCE MEASURES

How do you measure the performance of your system?

PERFORMANCE MEASURES

there are various facets to the performance of a computer. For example.

- A user of a computer measures its performance based on the time taken to execute a given job (program).
- A laboratory engineer measures the performance base on the total amount of work done in a given time.

PERFORMANCE MEASURES

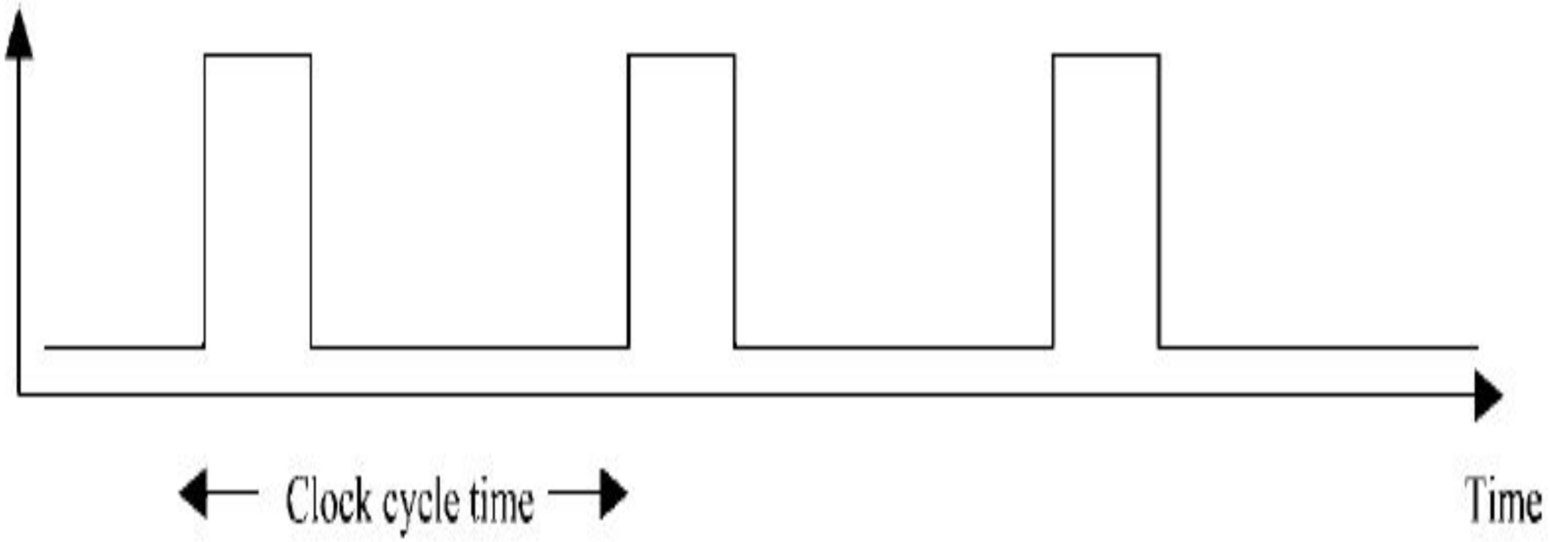
While the user considers the program execution time a measure for performance, the laboratory engineer considers the throughput a more important measure for performance.

Performance analysis should help answering questions such as how fast can a program be executed using a given computer?

In order to answer such a question, we need to determine the time taken by a computer to execute a given job.

- The time required to execute a job by a computer is often expressed in terms of clock cycles.
- We define the clock cycle time as the time between two consecutive rising (trailing) edges of a periodic clock signal

Clock cycles allow counting unit computations, because the storage of computation results is synchronized with rising (trailing) clock edges.



Clock signal

It may be easier to count the number of instructions executed in a given program as compared to counting the number of CPU clock cycles needed for executing that program.

- We denote the number of CPU clock cycles for executing a job to be the cycle count (CC),
- the cycle time by CT, and
- the clock frequency by $f = 1/CT$.

The time taken by the CPU to execute a job can be expressed as

$$\textbf{CPU time} = \mathbf{CC * CT = CC/f}$$

The average number of clock cycles per instruction (CPI) has been used as an alternate performance measure. The following equation shows how to compute the CPI.

$$CPI = \frac{CPU\ clock\ cycles\ for\ the\ program}{Instruction\ count}$$

$$CPU\ time = Instruction\ count \times CPI \times Clock\ cycle\ time$$

$$= \frac{Instruction\ count \times CPI}{Clock\ rate}$$

- It is known that the instruction set of a given machine consists of a number of instruction categories: ALU (simple assignment and arithmetic and logic instructions), load, store, branch, and so on.
  In the case that the CPI for each instruction category is known, the overall CPI can be computed as

where Ii is the number of times an instruction of type i is executed in the program and CPIi is the average number of clock cycles needed to execute such instruction.

$$CPI = \frac{\sum_{i=1}^{n} CPI_i \times I_i}{Instruction\ count}$$

Example Consider computing the overall CPI for a machine A for which the following performance measures were recorded when executing a set of benchmark programs. Assume that the clock rate of the CPU is 200 MHz.

Instruction category	Percentage of occurrence	No. of cycles per instruction
ALU	38	1
Load & store	15	3
Branch	42	4
Others	5	5

$$CPI = \frac{\sum_{i=1}^{n} CPI_i \times I_i}{Instruction\ count}$$

Assuming the execution of 100 instructions, the overall CPI can be computed as

$$CPI_a = \frac{\sum_{i=1}^{n} CPI_i \times I_i}{Instruction\ count} = \frac{38 \times 1 + 15 \times 3 + 42 \times 4 + 5 \times 5}{100} = 2.76$$

- It should be noted that the CPI reflects the organization and the instruction set architecture of the processor while the instruction count reflects the instruction set architecture and compiler technology used.
- This shows the degree of interdependence between the two performance parameters.

Therefore, it is imperative that both the CPI and the instruction count are considered in assessing the merits of a given computer or equivalently in comparing the performance of two machines.

A different performance measure that has been given a lot of attention in recent years is MIPS (million instructions-per-second (the rate of instruction execution per unit

$$MIPS = \frac{Instruction\ count}{Execution\ time \times 10^6} = \frac{Clock\ rate}{CPI \times 10^6}$$

Execution time = CPI * IC /CR

$$MIPS = \frac{Instruction\ count}{Execution\ time \times 10^6} = \frac{Clock\ rate}{CPI \times 10^6}$$

Thank You