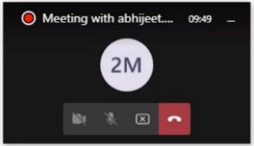
Computer Organization and Architecture

Introduction to Computer 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.

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



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

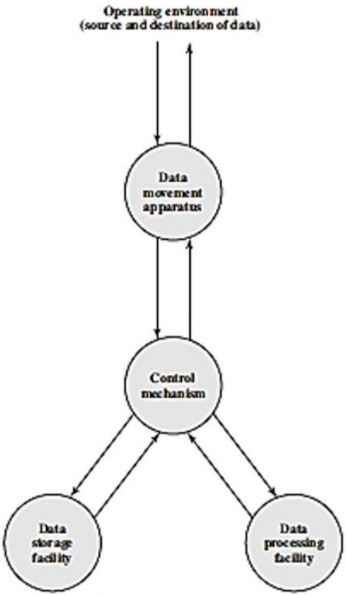
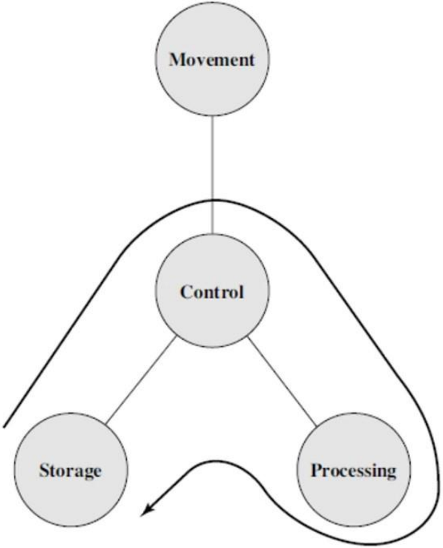


Figure 1.1 A Functional View of the Computer

Function Both the structure and functioning of a computer are, in essence, simple. Figure 1.1 depicts the basic functions that a computer can perform.

In general terms, there are only four:

- Data processing
- Data storage
- Data movement : peripheral devices, data communication, I/O
- Control: computer's resources and orchestrates the perfor
its functional parts


Movement
Control
Storage
Processing

structure

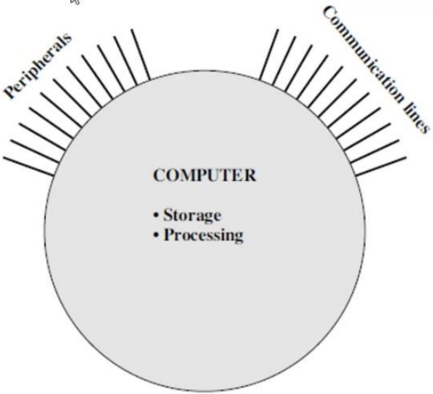


Figure 1.3 The Computer

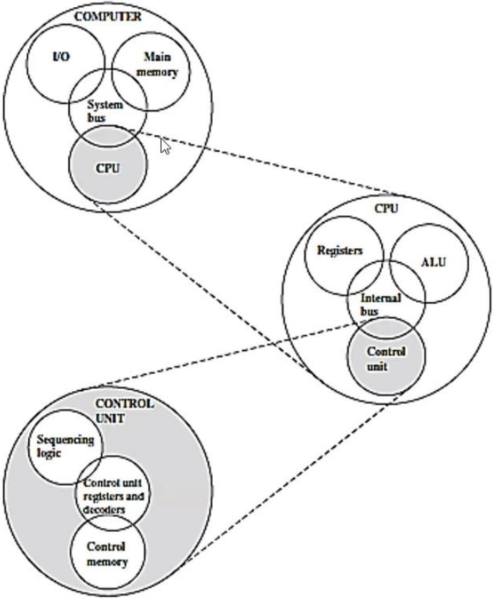


Figure 1.4 The Computer: Top-Level Structure

- 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.

- **system bus**, consisting of a number of connecting wires to which all the other components attach.

the most complex component is the CPU. Its major structural components are as follows:

- **Control unit:** Controls the operation of the CPU and hence the computer
- **Arithmetic and logic unit (ALU):** Performs the computer's data processing

functions

- **Registers:** Provides storage internal to the CPU
- **CPU interconnection:** Some mechanism that provides for communication
- mong the control unit, ALU, and registers

A Brief History of Computers

The evolution of computers has been characterized by

- increasing processor speed,
- decreasing component size,
- increasing memory size,
- and increasing I/O capacity and speed.

- One factor responsible for the great increase in processor speed is the shrinking size of microprocessor components; this reduces the distance between components and hence increases speed.

- **A critical issue** in computer system design is balancing the performance of the various elements so that gains in performance in one area are not handicapped by a lag in other areas.

- **The First Generation: Vacuum Tubes**

- ***ENIAC*** The ENIAC (Electronic Numerical Integrator And Computer), designed and constructed at the University of Pennsylvania, was the world's first general-purpose electronic digital computer.

- The major drawback of the ENIAC was that it had to be programmed manually by setting
- switches and plugging and unplugging cables.

- von Neumann and his colleagues began the design of a new stored program computer, referred to as the IAS computer, at the Princeton Institute for Advanced Studies.

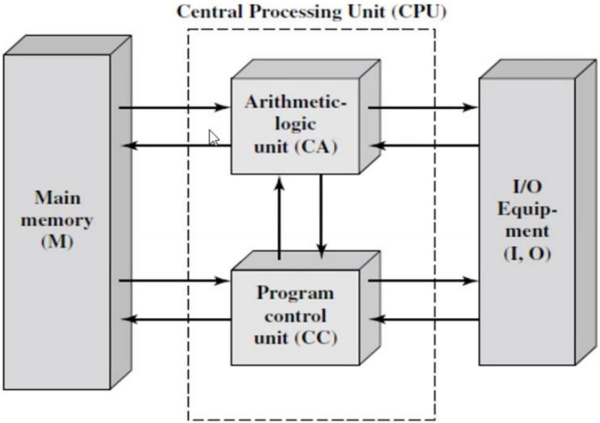


Figure 2.1 Structure of the IAS Computer

The Third Generation: Integrated Circuits

- A single, self-contained transistor is called a *discrete component.*

- Fourth and fifth generation

PERFORMANCE ASSESSMENT

- **Clock Speed and Instructions per Second:**

clock signals are generated by a quartz crystal, which generates a constant signal wave while power is applied. This wave is converted into a digital voltage pulse stream that is provided in a constant flow to the processor. For example, a 1-GHz processor receives 1 billion pulses per second. The rate of pulses is known as the **clock rate**, or **clock speed.**

- ***INSTRUCTION EXECUTION RATE***
- ***MIPS and MFLOPS***

Improvements in Chip Organization and Architecture

- There are three approaches to achieving increased processor speed:
- Increase the hardware speed of the processor:
- With gates closer together, the propagation time for signals is significantly reduced, enabling a speeding up of the processor.
- An increase in clock rate means that individual operations are executed more rapidly.

Improvements in Chip Organization and Architecture

- Increase the size and speed of caches that are interposed between the processor and main memory:
- In particular, by dedicating a portion of the processor chip itself to the cache, cache access times drop significantly.
- Make changes to the processor organization and architecture that increase the effective speed of instruction execution: Typically, this involves using parallelism in one form or another.

Improvements in Chip Organization and Architecture

- as **clock speed and logic density increase**, a number of **obstacles** become more significant
- **Power:** As the density of logic and the clock speed on a chip increase, so does the power density (Watts/cm2). The difficulty of **dissipating the heat generated on high-density,** high-speed chips is becoming a serious design issue.
- **RC delay:** The **speed at which electrons can flow on a chip between transistors is limited by the resistance and capacitance** of the metal wires connecting them; specifically, delay increases as the RC product increases. As components on the chip decrease in size, the wire interconnects become thinner, increasing resistance. Also, the wires are closer together, increasing capacitance.

Improvements in Chip Organization and Architecture

- as clock speed and logic density increase, a number of obstacles become more significant
- **Memory latency and throughput:** Memory access speed (latency) and transfer speed (throughput) lag processor speeds.

Multicore, Mics, and GPGPUs

- **Multicore:** placing multiple processors on the same chip, with a large shared cache.
- Studies indicate that, within a processor, **the increase in performance is roughly proportional to the square root of the increase in complexity.**
- But if the software can support the effective use of multiple processors, then doubling the number of processors almost doubles performance.
- Thus, the strategy is to **use two simpler processors on the chip rather than one more complex processor**.
- the **power consumption of memory logic on a chip is much less than that of processing logic**.

Multicore, Mics, and GPGPUs

- As the caches became larger, it made performance sense to create two and then three levels of cache on a chip, with initially, **the first-level cache dedicated to an individual processor and levels two and three being shared by all the processors.**
- It is now common for the second-level cache to also be private to each core.
- **many integrated core (MIC)**: more than 50 cores per chip.
- The multicore and MIC strategy involves a homogeneous collection of general purpose processors on a single chip.

multicore computer structure

- **Central processing unit (CPU):** That portion of a computer that **fetches and executes instructions.** It consists of an **ALU, a control unit, and registers.** In a system with a single processing unit, it is often simply referred to as a *processor*.
- **Core:** An individual processing unit on a processor chip. A core may be equivalent in functionality to a CPU on a single-CPU system. Other specialized processing units, such as one optimized for vector and matrix operations, are also referred to as cores.
- **Processor: A physical piece of silicon containing one or more cores.** The processor is the computer component that interprets and executes instructions. If a processor contains multiple cores, it is referred to as a **multicore processor**.

multicore computer structure

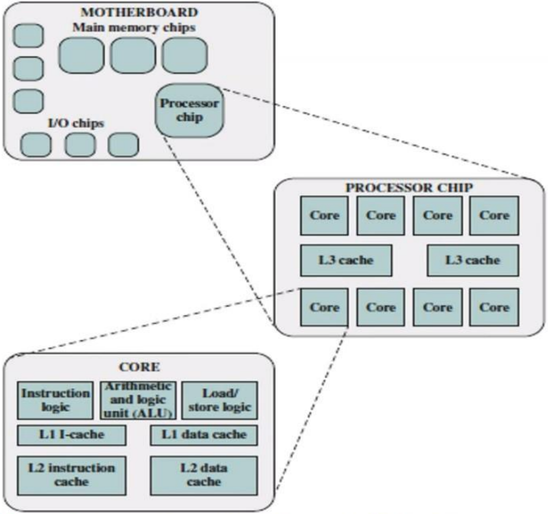


Figure 1.2 Simplified View of Major Elements of a Multicore Computer

Little's Law

- we have a steady state system to which **items arrive at an average rate of λ items per unit time. The items stay in the system an average of *W* units of time.** Finally, there is an **average of *L* units** in the system at any one time. Little's Law relates these three variables as
    - $L = \lambda W$.
- The server in this model can represent anything that performs some function or service for a collection of items.

- Since items arrive at a rate of λ, we can reason that in the time *w*, a total of λW items must have arrived. Thus $w = \lambda W$.

- To summarize, under steady state conditions, **the average number of items in a queuing system equals the average rate at which items arrive multiplied by the average time that an item spends in the system.**