COMPUTER ORGANIZATION AND ARCHITECTURE

CHAPTER 1

Introduction

The aim of this Chapter is to :

1. Understand the meaning of Computer Architecture and Organization.
2. Examine the main functions of the computer.
3. Look at how the different components of the computer are structured.
4. Explore the functions of each of the different components that make up the computer structure.
5. Understand the key performance issues that relate to computer design.

Introduction

- The course is about the **structure** and **functions** of modern day computers.

- Challenging task because:
  - There are so many computer products.
  - Computer technology is changing fast.

- The course will describe **general principles** of computer architecture that apply to **computers of any category**.

COMPUTER ARCHITECTURE

- The ***structure and behaviour*** of the various functional modules of the computer and how they interact to provide the processing needs of the user.

- Architectural attributes include:
  - Instruction set of the computer
  - Number of bits used to represent various data types
  - I/O mechanisms
  - Addressing Techniques
  - etc.

Computer Organisation

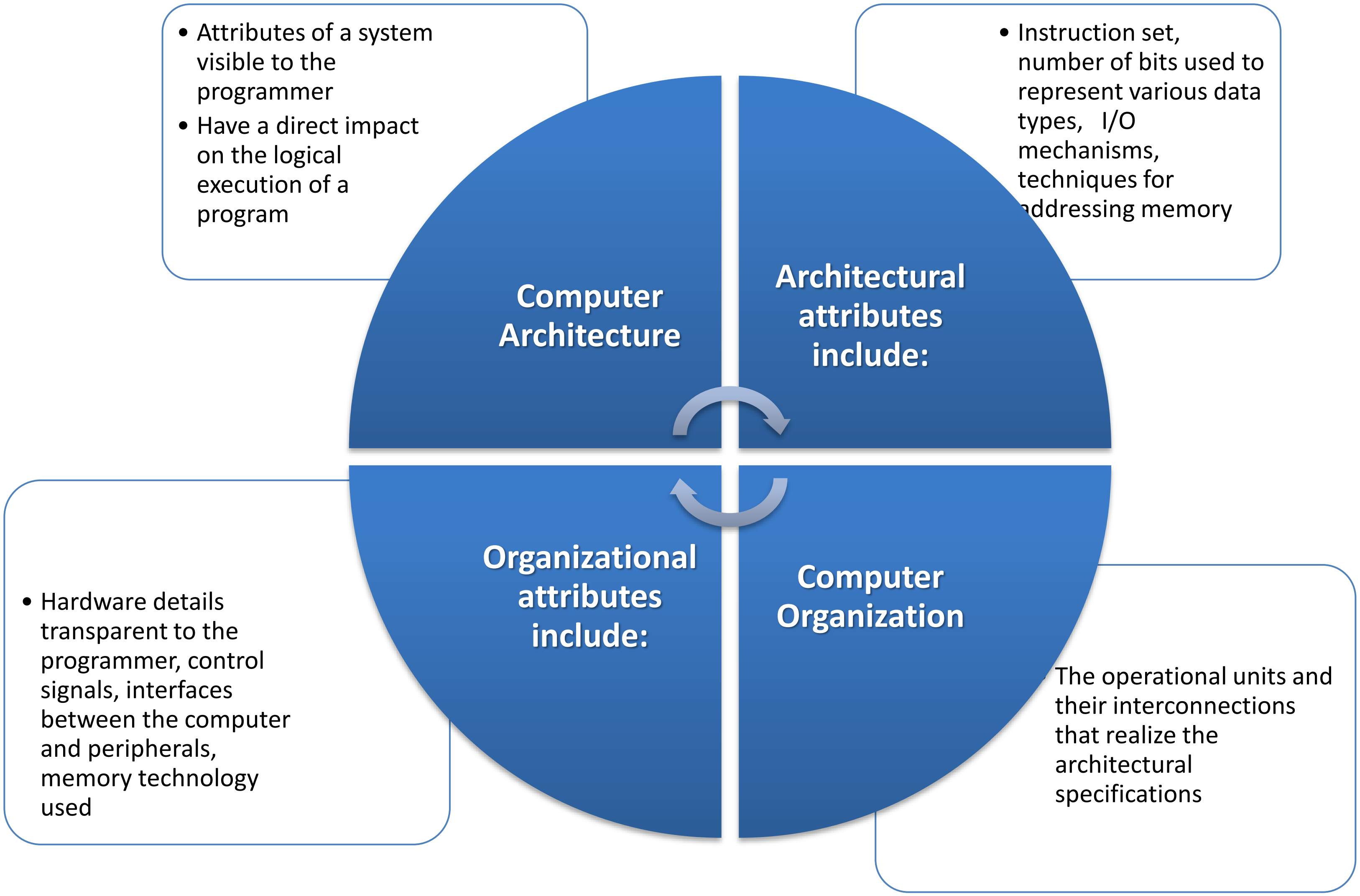
- The **way the hardware components are connected together** to form a computer system.

- Organisational attributes include hardware details visible to the user e.g. the interfaces, memory technology.

N.B. A number of manufacturers offer many different computer models (organizations) but all having the same architecture and thus differing in costs.

Computer Architecture

Computer Organization


Computer Architecture
Architectural attributes include:
Organizational attributes include:
Computer Organization

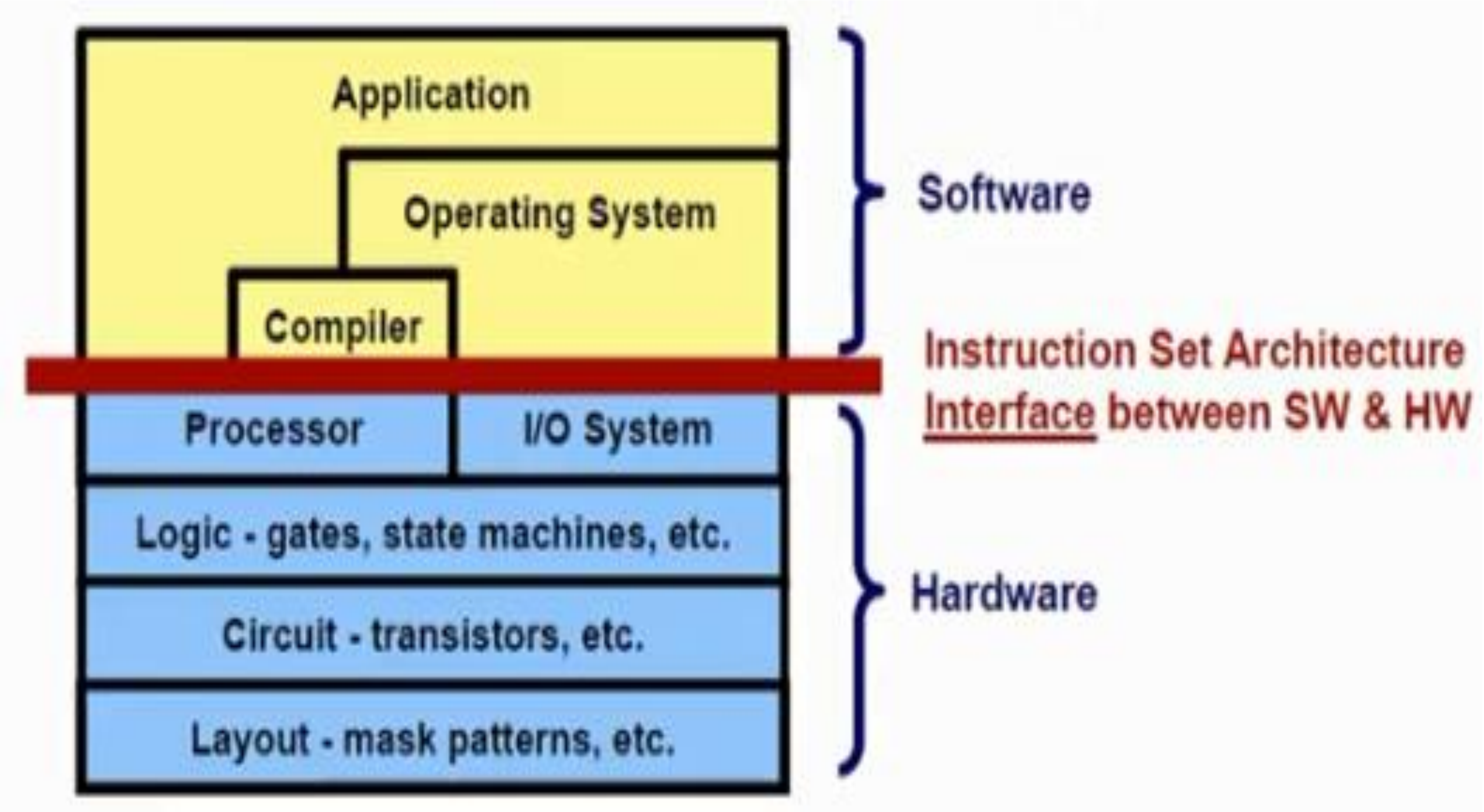
- Attributes of a system visible to the programmer
- Have a direct impact on the logical execution of a program

- Instruction set, number of bits used to represent various data types, I/O mechanisms, techniques for addressing memory

- Hardware details transparent to the programmer, control signals, interfaces between the computer and peripherals, memory technology used

- The operational units and their interconnections that realize the architectural specifications

COMPUTER ARCHITECTURE


Application
Operating System
Compiler
Software
Instruction Set Architecture
Interface between SW & HW
Processor
I/O System
Logic - gates, state machines, etc.
Circuit - transistors, etc.
Layout - mask patterns, etc.
Hardware

WHY STUDY COMPUTER ORGANIZATION AND ARCHITECTURE

- To understand the computer’s functional components, their characteristics, their performance and their interactions.

- Computer architecture helps to structure programs that can run more efficiently on a real machine (CPU speed, memory etc)

- To know the most cost effective computer for use in an organization.

- Computer architecture concepts are needed in other courses e.g. (programming, and operating Systems)

Structure and Function

- Hierarchical system
  - Set of interrelated subsystems

- Hierarchical nature of complex systems is essential to both their design and their description

- Designer need only deal with a particular level of the system at a time
  - Concerned with structure and function at each level

- Structure
  - The way in which components relate to each other
- Function
  - The operation of individual components as part of the structure

Function

- There are four basic functions that a computer can perform:
  - Data processing
    - Data may take a wide variety of forms and the range of processing requirements is broad
  - Data storage
    - Short-term
    - Long-term
  - Data movement
    - Input-output (I/O) - when data are received from or delivered to a device (peripheral) that is directly connected to the computer
    - Data communications – when data are moved over longer distances, to or from a remote device
  - Control
    - A control unit manages the computer’s resources and orchestrates the performance of its functional parts in response to instructions

Structure

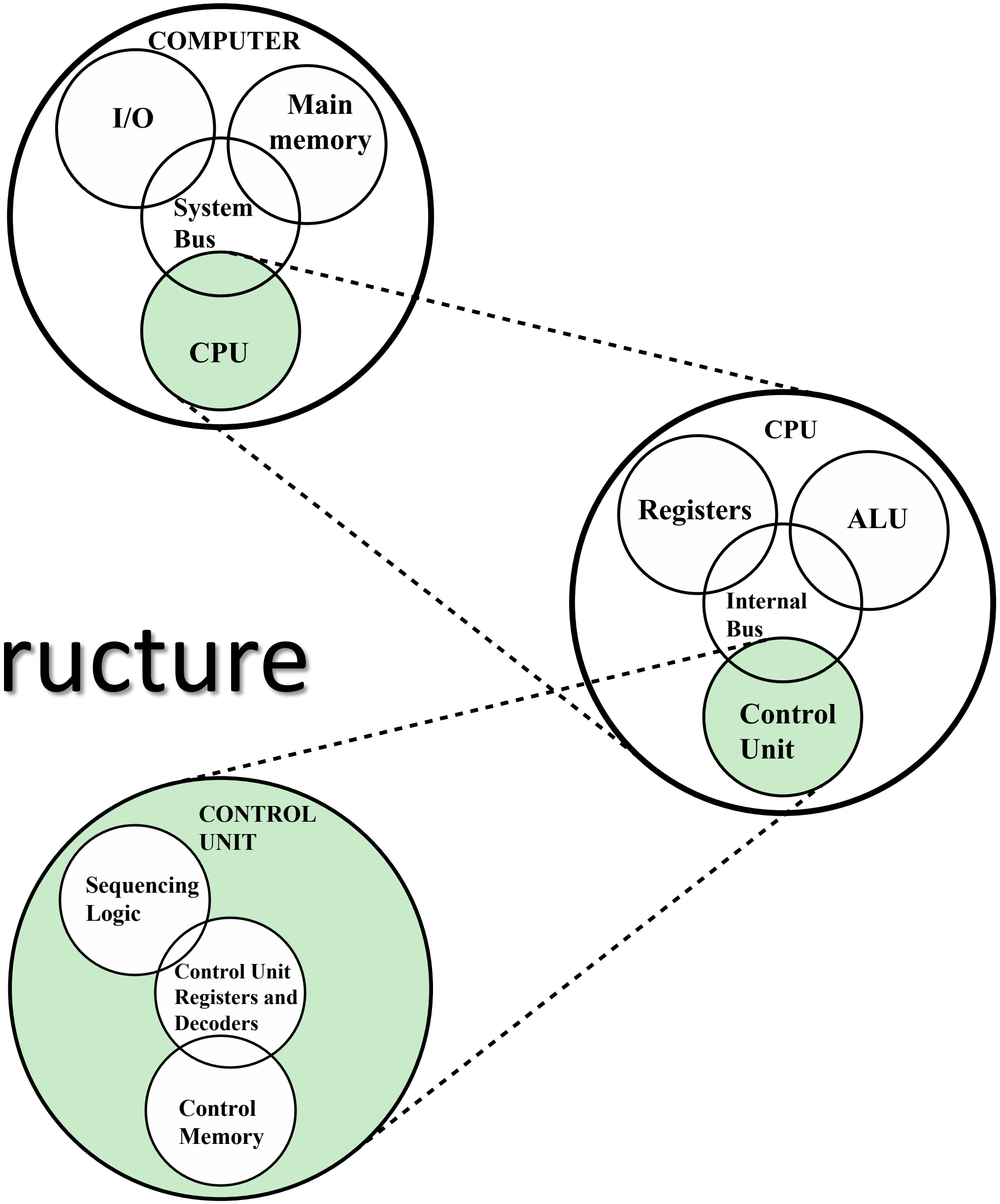
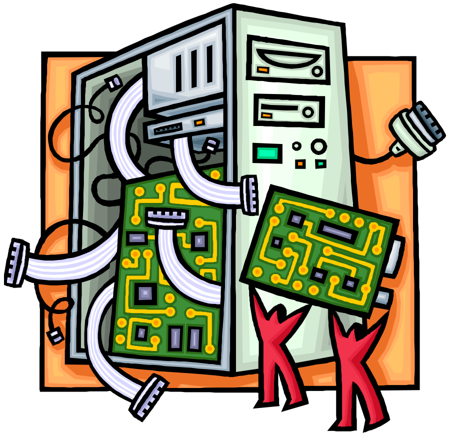


Figure 1.1 A Top-Down View of a Computer

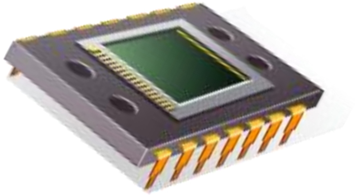
There are four main structural components of the computer:

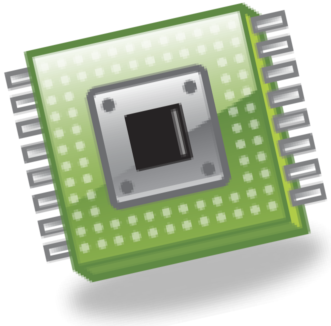


- CPU – controls the operation of the computer and performs its data processing functions
- Main Memory – stores both data and instructions that are currently being used.
- I/O – moves data between the computer and its external environment
- System Interconnection – some mechanism that provides for communication among CPU, main memory, and I/O.

CPU

Major structural components:





- Control Unit
  - Controls the operation of the CPU and hence the computer
- Arithmetic and Logic Unit (ALU)
  - Performs the computer’s data processing function
- Registers
  - Provide storage internal to the CPU
- CPU Interconnection
  - Some mechanism that provides for communication among the control unit, ALU, and registers

Multicore Computer Structure

- Central processing unit (CPU)
  - Portion of the computer that fetches and executes instructions
  - Consists of an ALU, a control unit, and registers
  - Referred to as a processor in a system with a single processing unit
- Core
  - An individual processing unit on a processor chip
  - May be equivalent in functionality to a CPU on a single-CPU system
  - Specialized processing units are also referred to as cores
- Processor
  - A physical piece of silicon containing one or more cores
  - Is the computer component that interprets and executes instructions
  - Referred to as a *multicore processor* if it contains multiple cores

Cache Memory

- Multiple layers of memory between the processor and main memory.
- Is smaller and faster than main memory.
- Used to speed up memory access by placing in the cache data from main memory that is likely to be used in the near future.
- A greater performance improvement may be obtained by using multiple levels of cache, with level 1 (L1) closest to the core and additional levels (L2, L3, etc.) progressively farther from the core.
- The L2 cache is slower and typically larger than the L1 cache, and the L3 cache is slower and typically larger than the L2 cache.
- When the processor attempts to read a word of memory, a check is made to determine if the word is in the cache. If so, the word is delivered to the processor. If not, a block of main memory, consisting of some fixed number of words, is read into the cache and then the word is delivered to the processor.

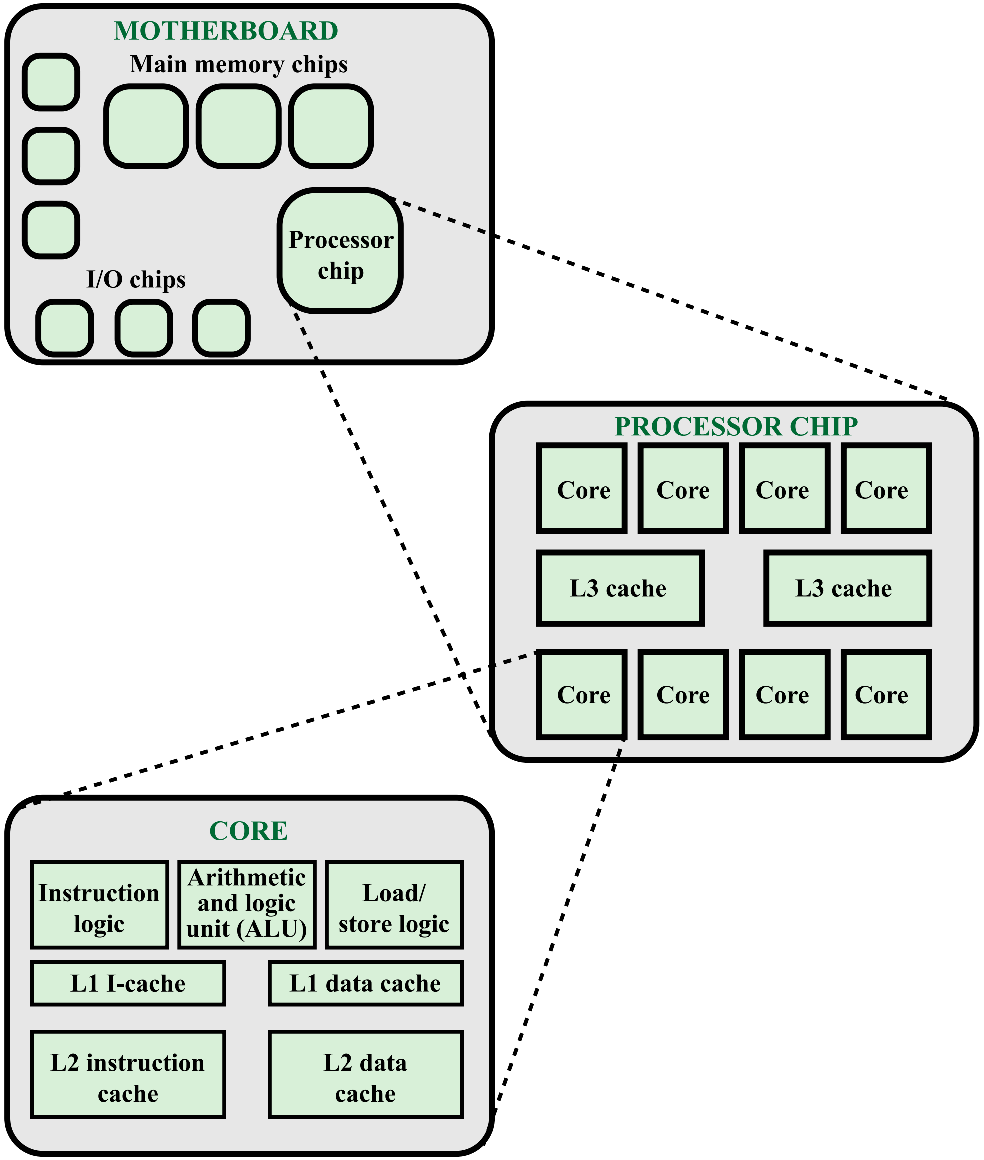


Figure 1.2 Simplified View of Major Elements of a Multicore Computer

The Microprocessor(CPU)

1. 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.
- Results of the operations are stored back in the memory or retained in the processor for immediate use.

The Microprocessor(CPU)

2. The Control Unit (CU)

- 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.

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.

I/O Function

- I/O module can exchange data directly with the processor
- Processor can read data from or write data to an I/O module
  - Processor identifies a specific device that is controlled by a particular I/O module
  - I/O instructions rather than memory referencing instructions
- In some cases it is desirable to allow I/O exchanges to occur directly with memory
  - The processor grants to an I/O module the authority to read from or write to memory so that the I/O memory transfer can occur without tying up the processor.
  - The I/O module issues read or write commands to memory relieving the processor of responsibility for the exchange.
  - This operation is known as direct memory access (DMA).

System Interconnection

Mechanism to provide communication between the CPU, memory and the I/O sub system. It consists of the **System Bus** and the **Interfaces**

System Bus.

A set of conductors that connect the CPU to its memory and I/O devices. The bus conductors are normally separated into 3 groups:

- ***The Data Lines****: for transmitting information*
- ***Address Lines****: Indicate where information is to come from or where it is to be placed.*
- ***Control Lines****: To regulate the activities on the bus.*

Interfaces

Circuitry needed to connect the bus to a device.

Memory interfaces

- *Decode the address of the memory location being accessed.*
- *Buffer data onto/off the bus.*
- *Contain circuitry to perform memory reads or write*.

I/O interfaces

- *Buffer data onto/off the system bus*
- *Receive commands from the CPU*
- *Transmit information from their devices to the CPU.*

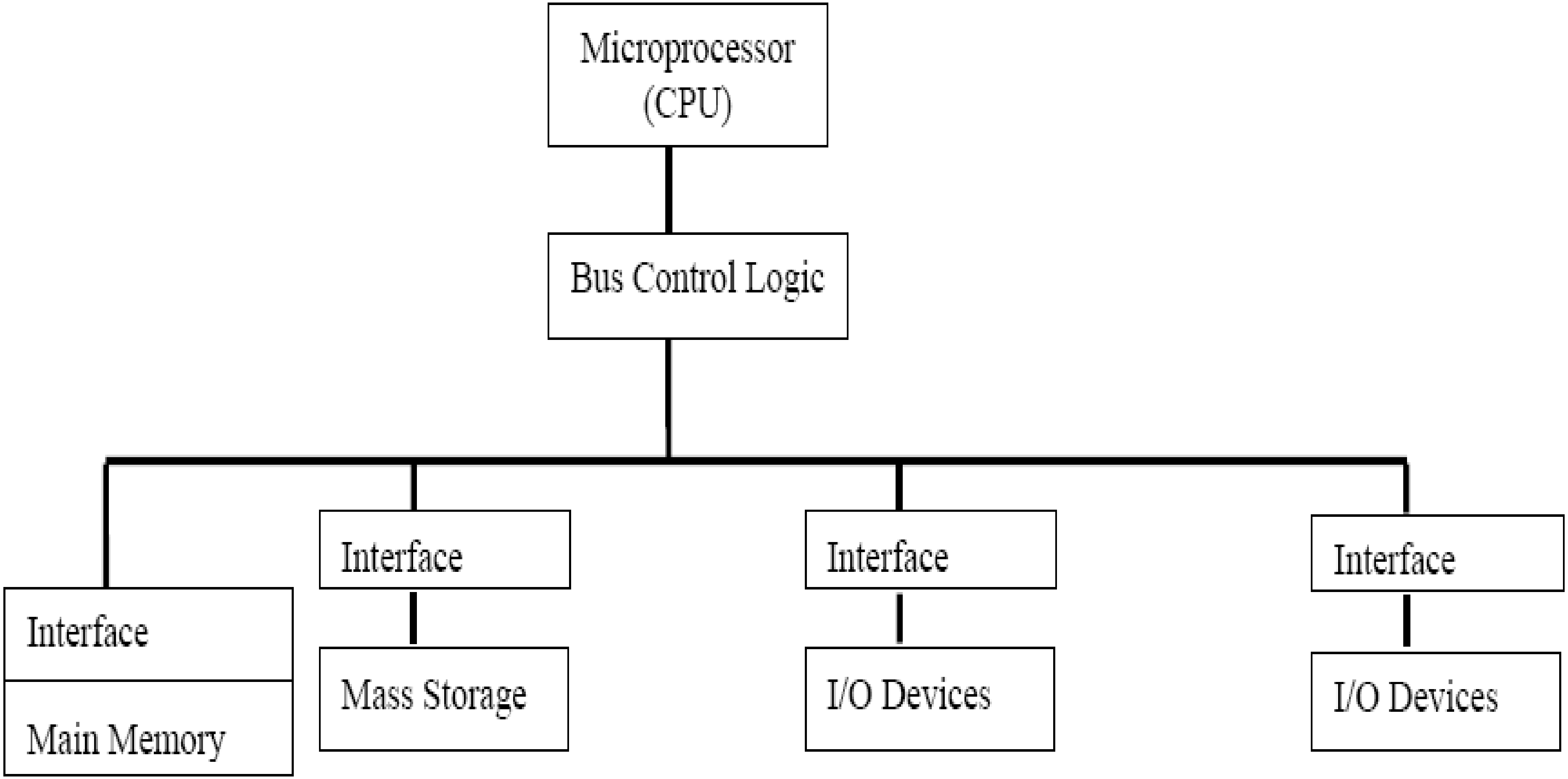
COMPUTER STRUCTURE

Two arrangements of these components can be described:

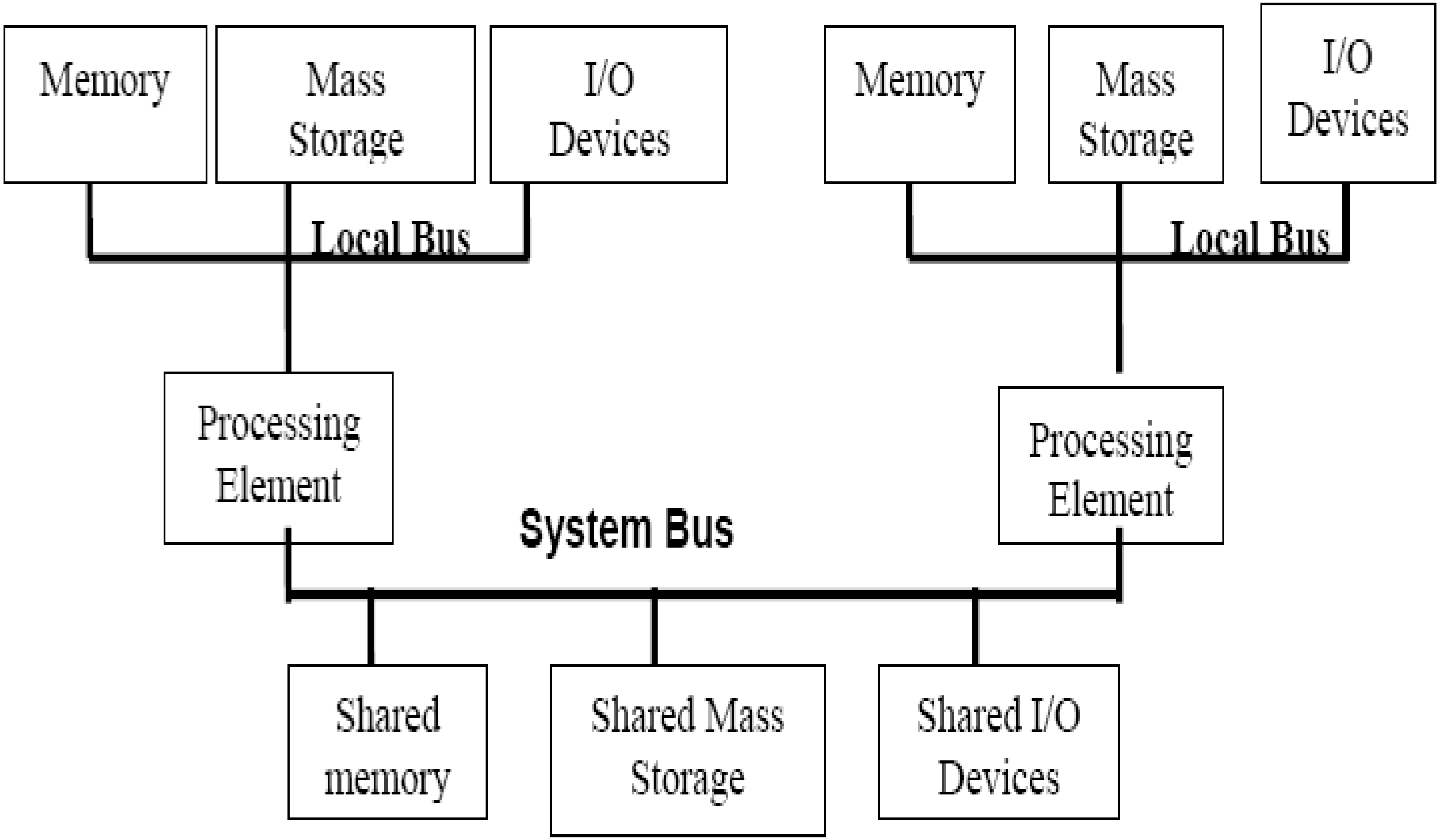
- The **single bus / Single processor architecture:** one processing element and all the other components are connected to a single link (the **System Bus)**

- The **Multiprocessing System:** has several processing elements surrounded by different subsystems and a central link (the **system bus**) connecting the different subsystems together.

Single Bus / Single Processor



Multi Processing System



Multi Processing System

- The links in the subsystems are called **local buses**.

- Each subsystem operates as an independent computer but can take advantage of the shared resources.

- The shared main memory can be used for passing information between subsystems

- The shared mass storage can be used to store large programs and large quantities of data that are needed by more than one subsystem.

- The competition for the shared resources by the different elements is called **contention**.

The Computer Structure in Detail

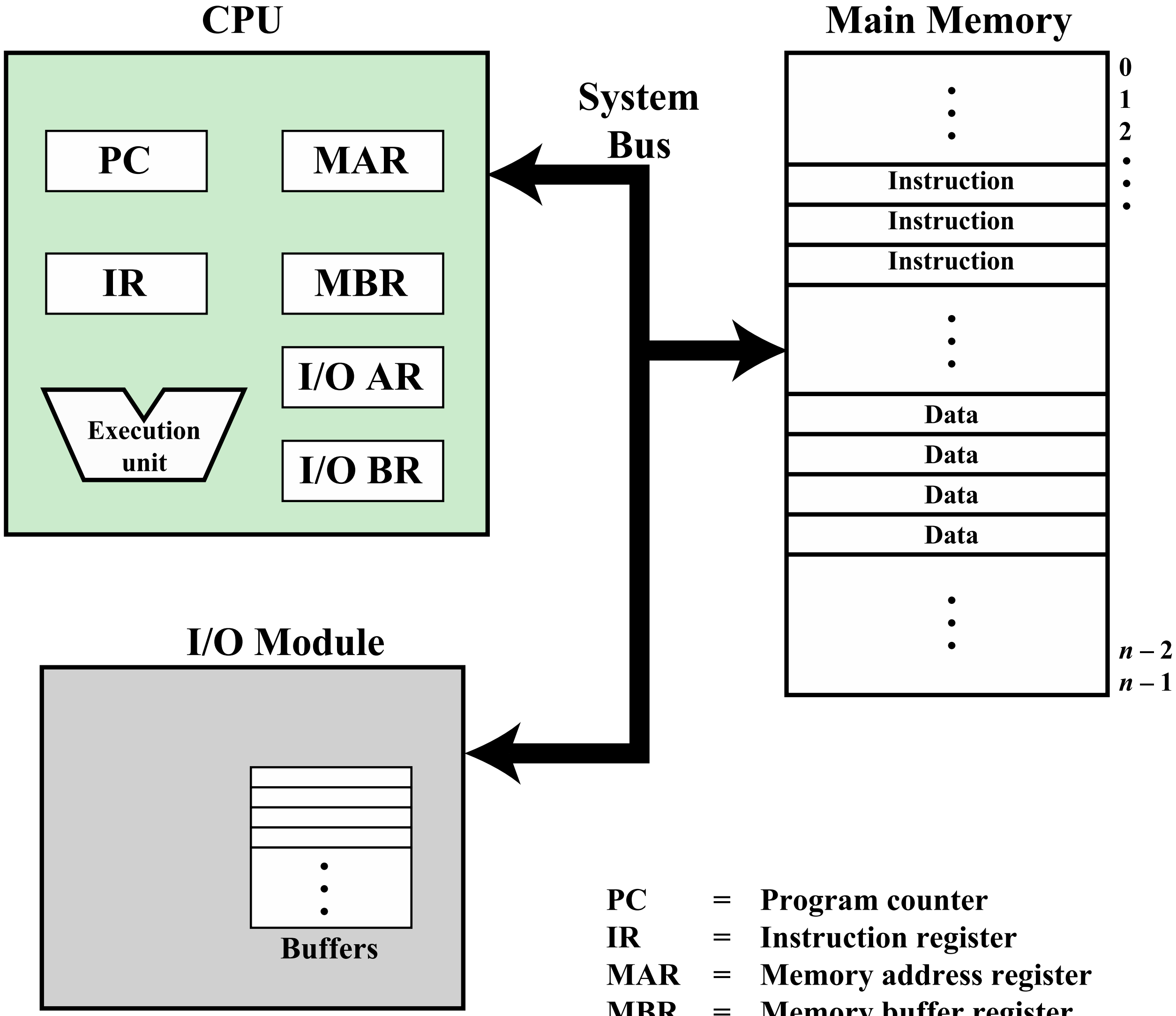
In this section we examine:

- The operations of the CPU, memory, I/O and System bus.

The CPU

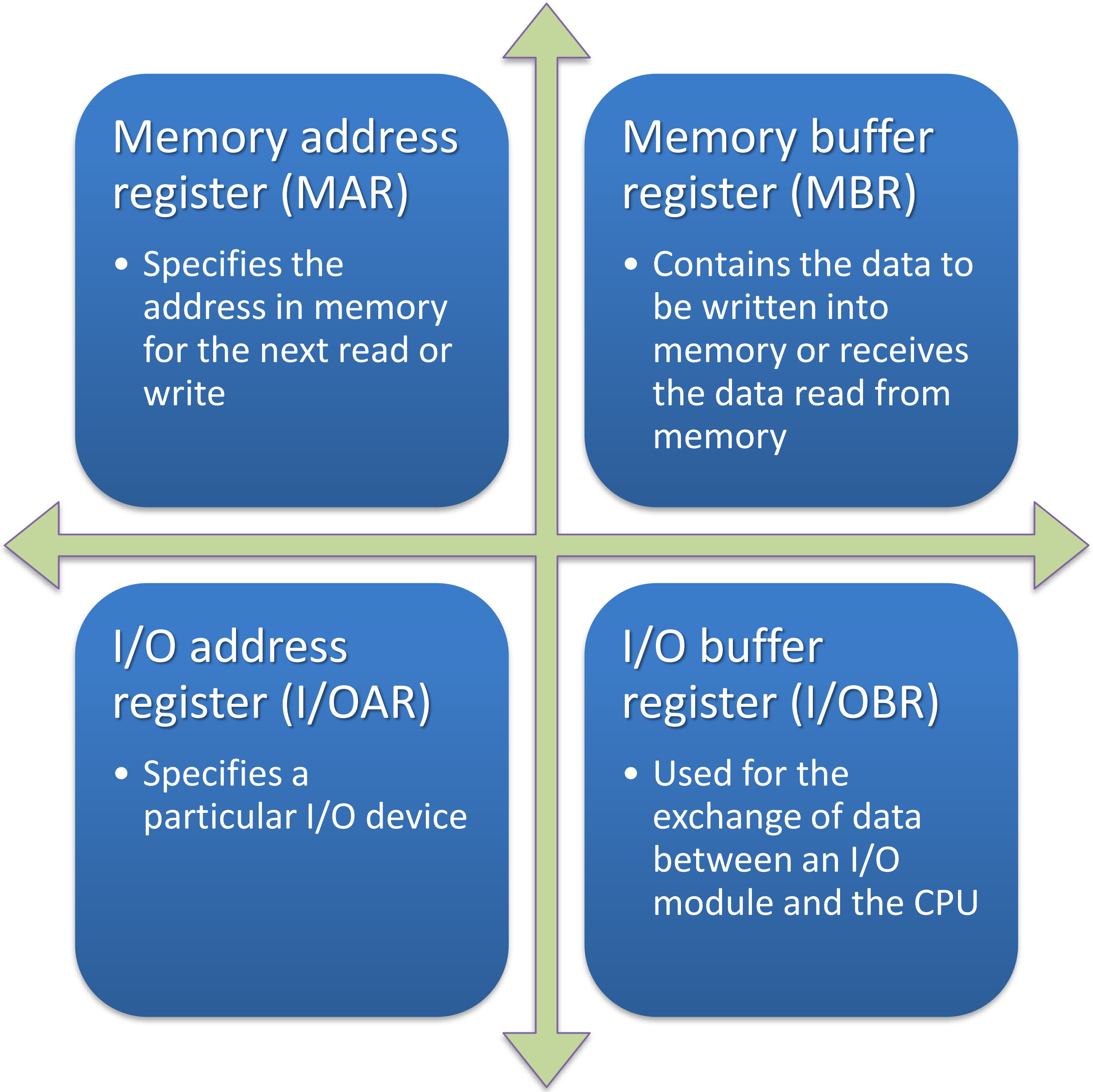
The main structural components of the CPU are:

- The Control Unit
- The working Registers
- The Arithmetic and Logic Unit.



PC = Program counter
IR = Instruction register
MAR = Memory address register
MBR = Memory buffer register
I/O AR = Input/output address register
I/O BR = Input/output buffer register

Figure 3.2 Computer Components: Top-Level View



Memory address register (MAR)

- Specifies the address in memory for the next read or write

Memory buffer register (MBR)

- Contains the data to be written into memory or receives the data read from memory

I/O address register (I/OAR)

- Specifies a particular I/O device

I/O buffer register (I/OBR)

- Used for the exchange of data between an I/O module and the CPU

MEMORY

MAR

MBR

The CPU

- ***The Program Counter (PC)***

  Holds the address of the main memory location from which the next instruction is to be fetched

- ***Instruction Register (IR)***

  Receives the instruction when it is brought from memory and holds it while it gets decoded and executed

The CPU

Arithmetic/Logic Unit

- ➢ It performs arithmetic and logical operations on the contents of the working registers, the PC, memory locations etc.

- ➢ It also sets and clears the appropriate flags.

Working Registers

They are Arithmetic registers (accumulators) and address registers.

Arithmetic Registers:
Temporarily hold the operands and the result of the arithmetic operations

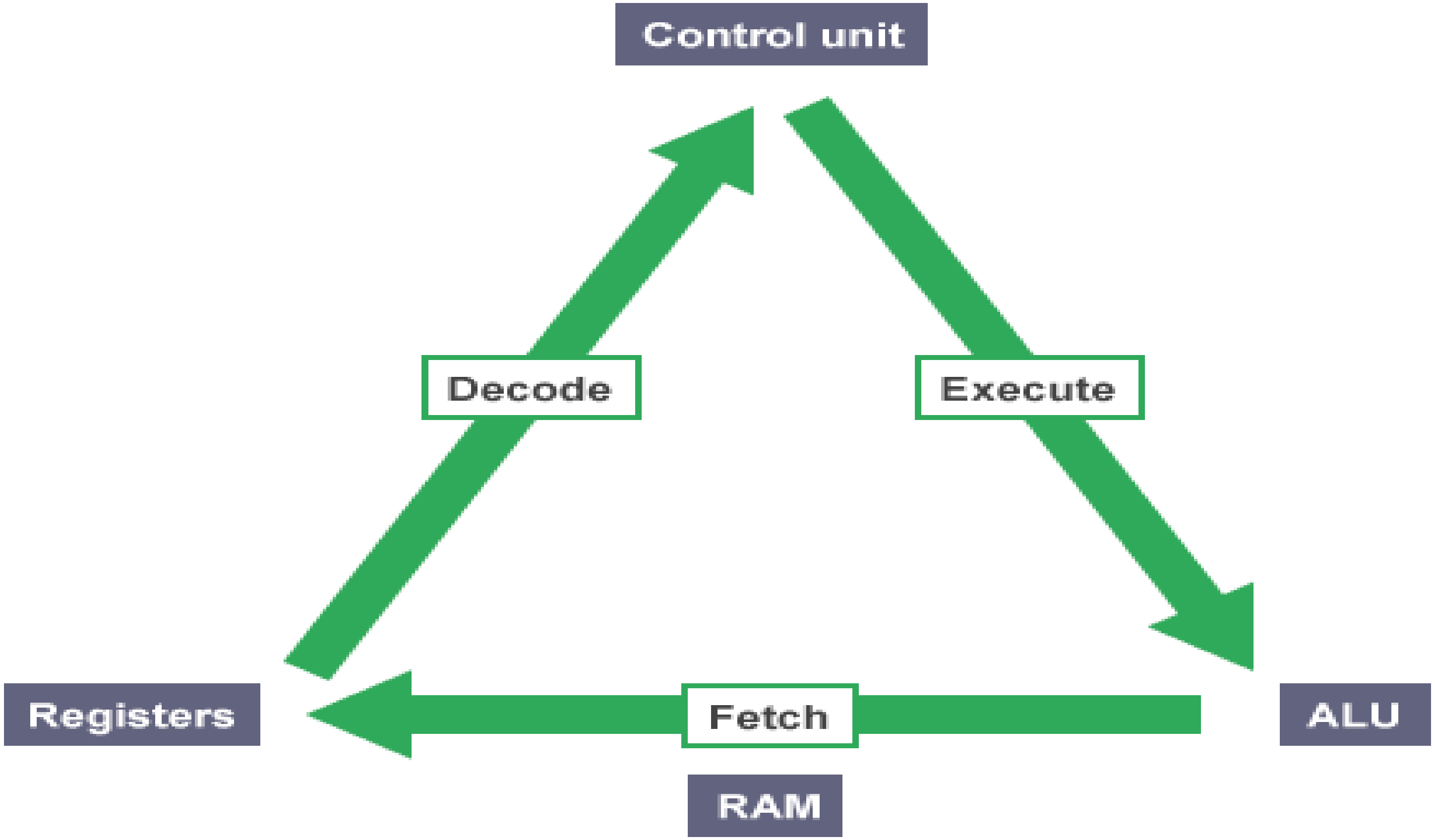
Address Registers:
for addressing data and instructions in main memory.

Accessing a register is faster than accessing memory.

If a register can be used for both arithmetic operations and addressing it is then called a general purpose register.

The CPU

- **Fetch-decode-execute cycle**
  - The fetch-decode-execute cycle is the sequence of steps that the CPU follows to process instructions.



The Fetch-Decode-Execute cycle Explained:

1. The processor checks the program counter to see which instruction to run next.
2. The program counter gives an address value in the memory of where the next instruction is.
3. The processor fetches the instruction value from this memory location.
4. Once the instruction has been fetched, it needs to be decoded and executed. For example, this could involve taking one value, putting it into the ALU, then taking a different value from a register and adding the two together.
5. Once this is complete, the processor goes back to the program counter to find the next instruction.
6. This cycle is repeated until the program ends.

Memory

- All memory locations and I/O registers are composed of bits.
- Because bits contain very little information, they are grouped together to form bytes and words.

- **A byte**: a group of 8 bits
- **A nibble**: a group of 4 bits
- **A word**: a group of 2,3, or 4 bytes depending on the computer and its system bus structure.

- Each byte has an identifying address associated with it.
- When a byte is to be accessed its address is transmitted to the appropriate interface via the address lines.

- Addresses are composed of bit combinations and the set of all bit combinations for a given situation is called an **address space**.

Memory

- The number of bits in an address determines the size of an address space.

  If an address is **n** bits wide then there are $2^{\mathbf{n}}$ possible addresses ($0 – 2^{\mathbf{n}}–1$).

- Some high order bits in a memory address are used to select the module and the remaining lower order bits identify the bytes or word within the module.

- Similarly an interface is identified by the high order bits of an I/O address and the register within the interface is selected by the 2 or 3 low order bits.

Memory

- The number of address lines in the system bus dictates the size of memory, or memory and the I/O space. A total of **n** address lines would imply a maximum memory (or overall memory and I/O) capacity of $2^{\mathbf{n}}$ bytes.

- 16 address lines imply $2^{\mathbf{16}} = 2^{\mathbf{6}} (2^{\mathbf{10}}) = 64K$

- Putting information into or taking information from a memory location is called ***memory access***.

Classifications of memory

Memory can be classified as to whether it can retain its contents when power is turned off.

- **Volatile**: Metal Oxide Semiconductor
  - Erased when computer is switched off. E.g RAM

- **Non Volatile**: Magnetic Core
  - Can retain its content while power is switched off. E.g ROM

Classification of Memory

It can also be classified according to its Read/Write capabilities.

❑ **ROM**: (Read Only Memory):

- Can only be read.

- Once its contents are set, they can only be read by special equipment.

- It is useful for storing software that is rarely changed during the life of the system, also known as firmware.

❑ **RAM:** (Random Access Memory):

- Memory that can be both read and written into(read/write).

Classification of ROM

Classified according to the way in which their contents are set (**programmed**)

- ❑ **MASKED ROM:**

  Programmed by a masking operation while the chip is being manufactured. They cannot be altered by the user.

- ❑ **PROM (Programmable ROM)**:

  contents can be set by the user using special equipment. Once programmed its contents can never be changed.

Classification of ROM

❑ **EPROM (Erasable Programmable ROM)**

- Can be reprogrammed by using a special equipment.
- Programmed by charge injection and once programmed the charge distribution is maintained until it is disturbed by some external energy source like Ultra Violet light.
- The external energy destroys the old memory contents and EPROM can be reprogrammed.

❑ **EAPROM (Electrically Alterable Programmable ROM)**

Programmed and erased electrically instead of ultra violet light.

RAM

Ram is of two types:

- **Static Ram**:

  keep its contents so long as power is on.

- **Dynamic Ram**:

  made of capacitors that can be charged or discharged. It must be refreshed often because of charge leakage.

I/O INTERFACES

Memory and peripherals are connected to buses through *interfaces and controllers*.

- **A controller:** initiates commands given to a device and it senses the status of the device.

- **An interface** connects the peripheral and its control circuitry to the bus.

I/O INTERFACE

Functions of the interface include:

- Make the status of the peripheral available to the computer.
- Provide buffer storage for input data.
- Provide buffer storage for output data.
- Relay commands from the computer to the peripheral.
- Signal to the CPU when the operation is complete.
- Signal to the computer when an error occurs.
- Pack bits into bytes or words for input and unpack them for output.

Data Transfer

It is categorized according to the amount of data transferred.

- **Byte/Word Transfer**

  one byte or word is moved by one command.

- **Block Transfer**

  A whole block of information is moved by a single command e.g. Direct memory Access transfers which are between memory and the peripheral.

Data Transfer

In block transfers a device’s interface must be used in conjunction with a **DMA controller** that can access memory directly without intervention by the CPU. e.g. a disk uses DMA.

Most devices that require high transfer rates are DMA devices.

When DMA capability is available it has higher priority over all other bus activity.

Many interfaces are designed to perform both types of transfers.

System Bus:

A system bus is classified into three functional groups; data, address, and control lines.

Data Bus:

❑ Data lines that provide a path for moving data among system modules

- May consist of 32, 64, 128, or more separate lines
- The number of lines is referred to as the *width* of the data bus
- The number of lines determines how many bits can be transferred at a time
- The width of the data bus is a key factor in determining overall system performance

Address Bus

- Used to designate the source or destination of the data on the data bus
  - If the processor wishes to read a word of data from memory it puts the address of the desired word on the address lines
- Width determines the maximum possible memory capacity of the system
- Also used to address I/O ports
  - The higher order bits are used to select a particular module on the bus and the lower order bits select a memory location or I/O port within the module

Control Bus

- Used to control the access and the use of the data and address lines
- Because the data and address lines are shared by all components there must be a means of controlling their use
- Control signals transmit both command and timing information among system modules
- Timing signals indicate the validity of data and address information
- Command signals specify operations to be performed

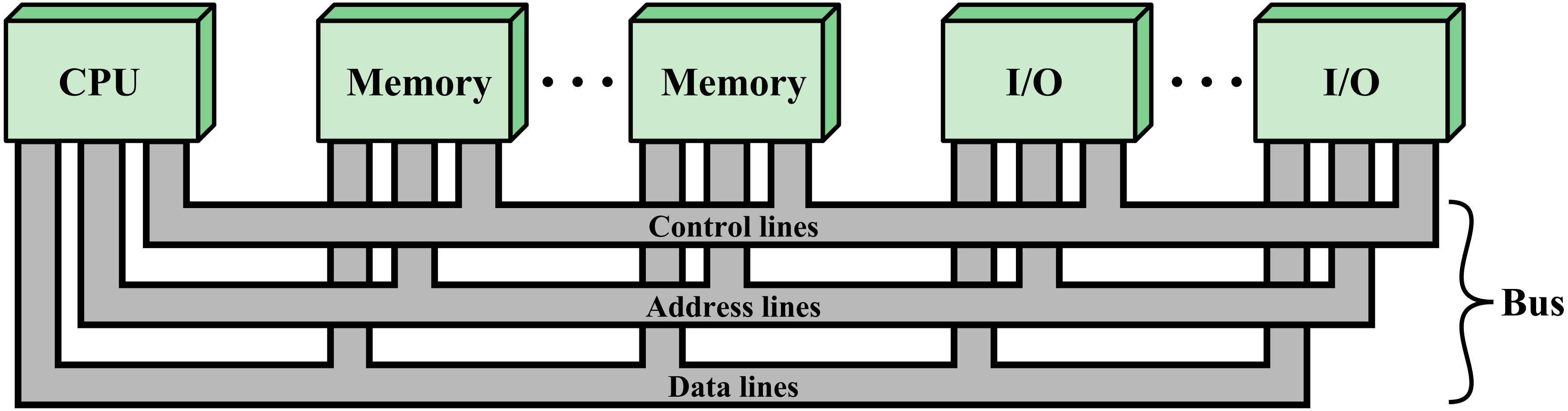


Figure 3.16 Bus Interconnection Scheme

Performance Issues:
Designing for Performance

- The cost of computer systems continues to drop dramatically, while the performance and capacity of those systems continue to rise equally dramatically.
- Today's laptops have the computing power of an IBM mainframe from 10 or 15 years ago.
- Processors are so inexpensive that we now have microprocessors we throw away.
- Desktop applications that require the great power of today's microprocessor-based systems include:
  - Image processing
  - Speech recognition
  - Videoconferencing
  - Multimedia authoring
  - Voice and video annotation of files
  - Simulation modeling

- Businesses are relying on increasingly powerful servers to handle transaction and database processing and to support massive client/server networks that have replaced the huge mainframe computer centers of yesteryear.

- Cloud service providers use massive high-performance banks of servers to satisfy high-volume, high-transaction-rate applications for a broad spectrum of clients.

Microprocessor Speed

Techniques built into contemporary processors include:

Pipelining

- Processor moves data or instructions into a conceptual pipe with all stages of the pipe processing simultaneously.

Branch prediction

- Processor looks ahead in the instruction code fetched from memory and predicts which branches, or groups of instructions, are likely to be processed next.

Superscalar execution

- This is the ability to issue more than one instruction in every processor clock cycle. (In effect, multiple parallel pipelines are used.)

Data flow analysis

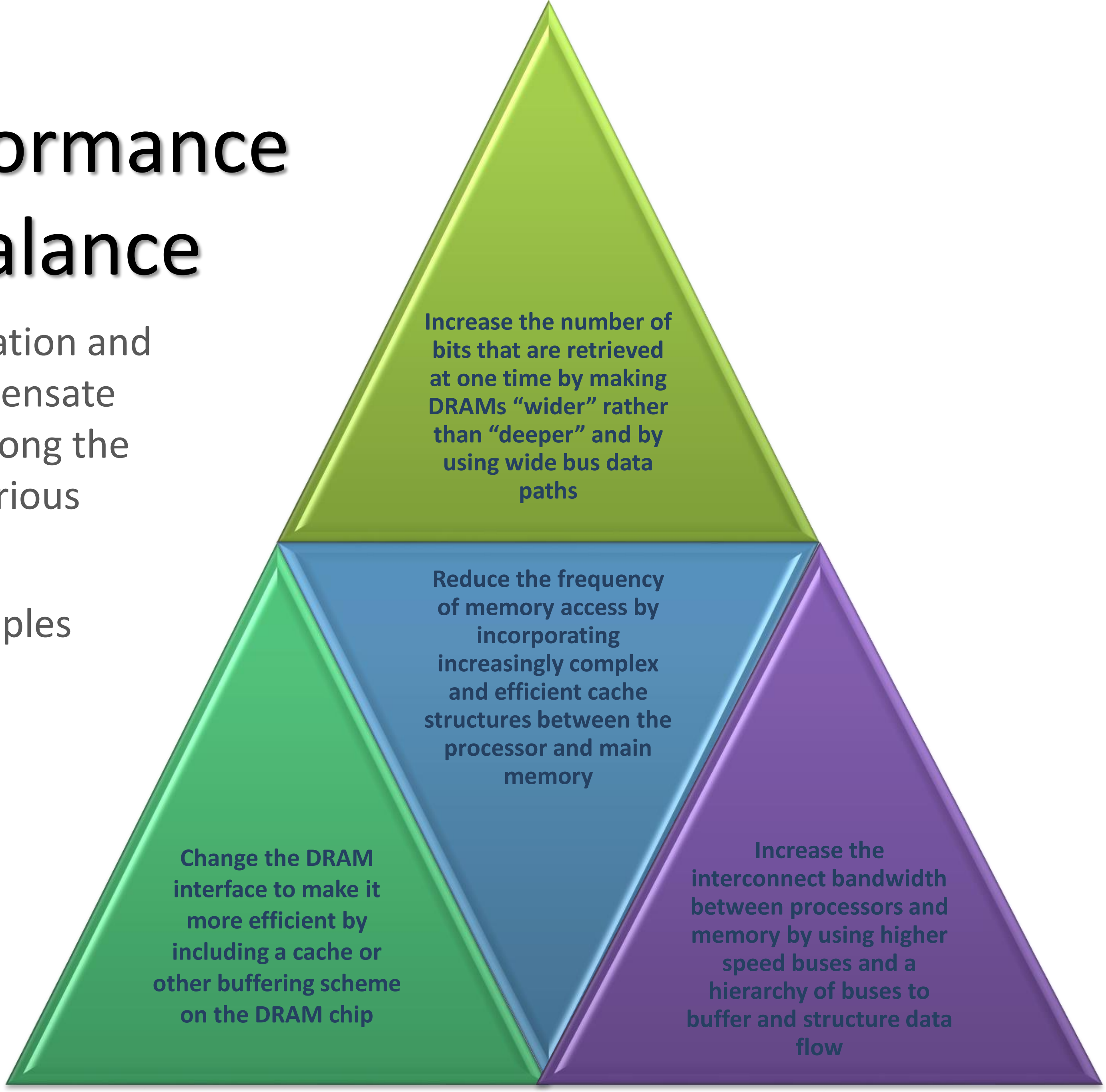
- Processor analyzes which instructions are dependent on each other's results, or data, to create an optimized schedule of instructions.

Speculative execution

- Using branch prediction and data flow analysis, some processors speculatively execute instructions ahead of their actual appearance in the program execution, holding the results in temporary locations, keeping execution engines as busy as possible.

Performance Balance

- Adjust the organization and architecture to compensate for the mismatch among the capabilities of the various components
- Architectural examples include:



Increase the number of bits that are retrieved at one time by making DRAMs “wider” rather than “deeper” and by using wide bus data paths

Change the DRAM interface to make it more efficient by including a cache or other buffering scheme on the DRAM chip

Reduce the frequency of memory access by incorporating increasingly complex and efficient cache structures between the processor and main memory

Increase the interconnect bandwidth between processors and memory by using higher speed buses and a hierarchy of buses to buffer and structure data flow

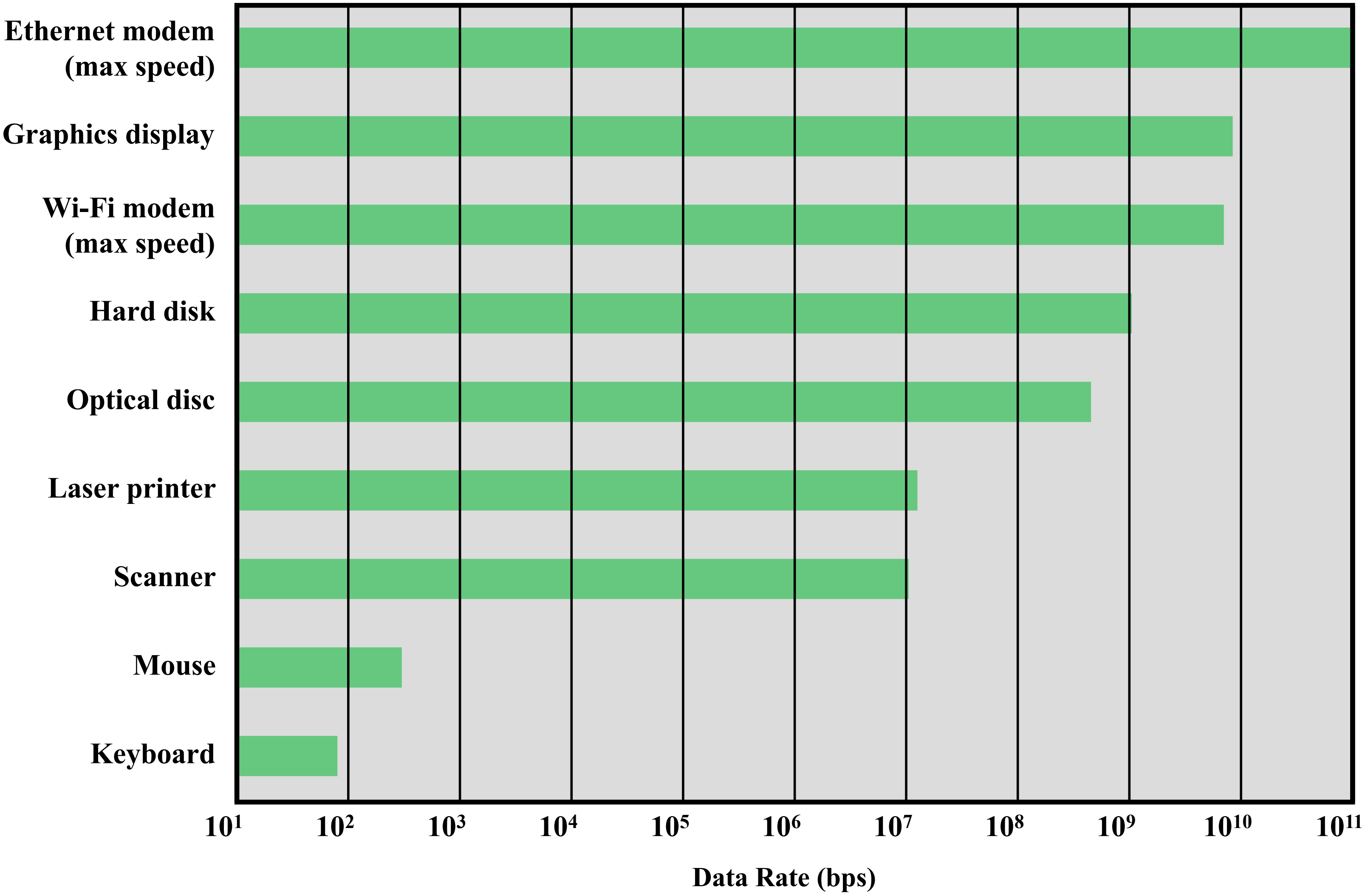


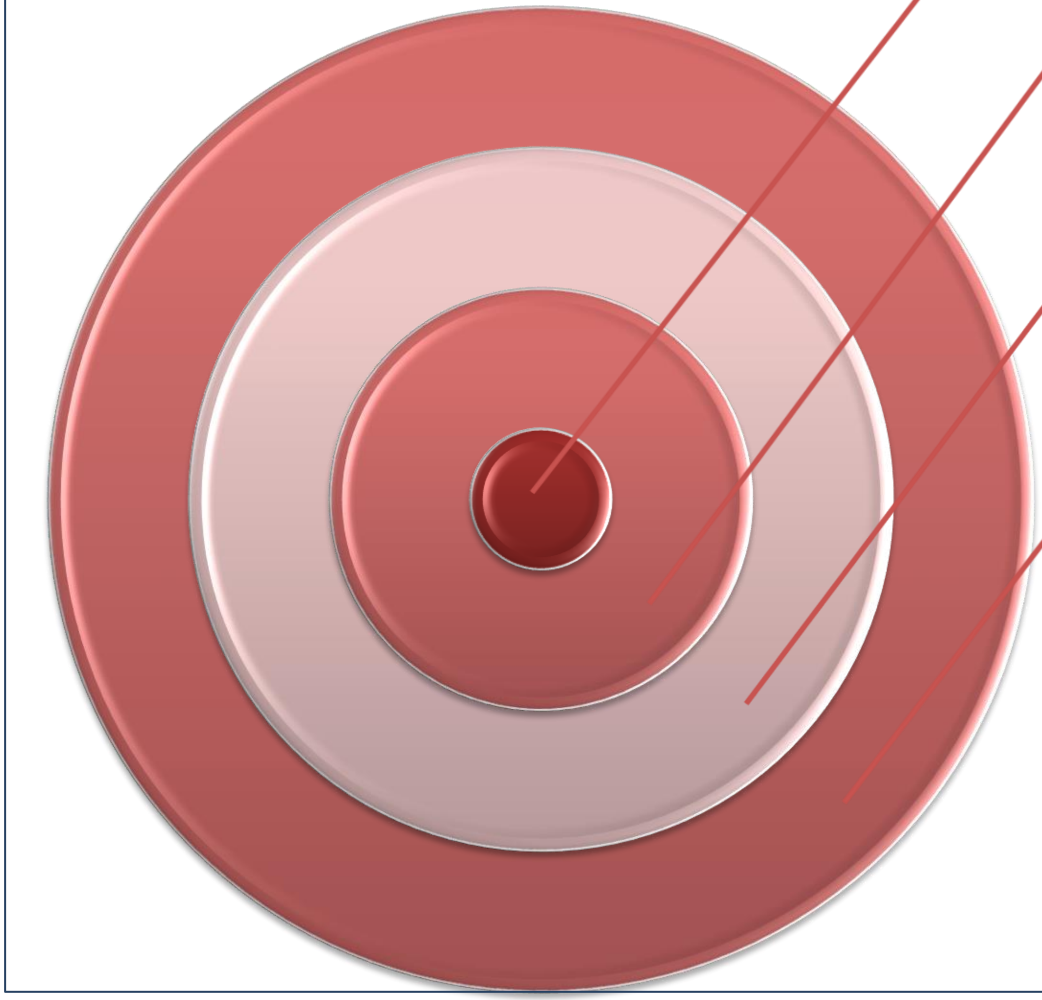
Figure 2.1 Typical I/O Device Data Rates

Improvements in Chip Organization and Architecture:

Approaches to achieving increased processor speed:

- Increase hardware speed of processor
  - Fundamentally due to shrinking logic gate size
    - More gates, packed more tightly, increasing clock rate
    - Propagation time for signals reduced
- Increase size and speed of caches that are interposed between the processor and main memory.
  - Dedicating part of processor chip to the cache.
    - Cache access times drop significantly
- Change processor organization and architecture
  - Increase effective speed of instruction execution
  - Parallelism

Multicore



The use of multiple processors on the same chip provides the potential to increase performance without increasing the clock rate

Strategy is to use two simpler processors on the chip rather than one more complex processor

With two processors larger caches are justified

As caches became larger it made performance sense to create two and then three levels of cache on a chip

Basic measures of Computer Performance.

1. Clock speed:

- Operations performed by a processor, such as fetching an instruction, decoding the instruction, performing an arithmetic operation, and so on, are governed by a system clock.
- Typically, all operations begin with the pulse of the clock.
- Thus, at the most fundamental level, the speed of a processor is dictated by the pulse frequency produced by the clock, measured in cycles per second, or Hertz (Hz).
- 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.**
- The time between pulses is the **cycle time.**
- The execution of an instruction involves a number of discrete steps, thus, most instructions on most processors require multiple clock cycles to complete.
- Some instructions may take only a few cycles, while others require dozens.
- In addition, when pipelining is used, multiple instructions are being executed simultaneously.
- Thus, a straight comparison of clock speeds on different processors does not tell the whole story about performance.

Basic measures of Computer Performance

2. Instruction Execution Rate

- A common measure of performance for a processor is the rate at which instructions are executed in one second, expressed as millions of instructions per second (MIPS), referred to as the MIPS rate.
- The machine cycle time is the time it takes to fetch and execute one instruction.

- Another common performance measure deals only with floating-point instructions.
- These are common in many scientific and game applications. Floating-point performance is expressed as millions of floating-point operations per second (MFLOPS). This is the measure of the arithmetical speed of a processor.

Next Chapter is on!!

Data Representation:

- In this chapter we shall cover the different formats used by the computer to represent the information it receives.

Reading Assignment

Note: Ensure to read about the following as this knowledge is going to be required in chapter 2.

Read and make notes on the following: Make sure you work out some examples.

1) Converting from decimal to binary and vice versa.
2) Converting from binary to Hexadecimal and vice versa.
3) Converting from Hexadecimal to decimal and vice versa.
4) Converting from Binary to octal and vice versa.
5) Converting from Octal to Hexadecimal and vice versa.