Computer Organization

Lec One: Introduction

Outlines

- What is a computer
- Why Computer
- Computer Components
- Types of Computers

What is a computer?

- A computer is an electronic device that accepts input, processes data, stores data, and produces output, all according to a series of stored instructions.

- You probably already know that you can use a computer to type documents, send an email, play games, and browse the Web.

Why Computer?

- A computer is a powerful and useful tool because it gives you a number of benefits, like including the ability to produce high quality work quickly. It also enables you to learn new skills that are an important part of today’s technological world.

1. **Speed**
2. **Reliability**
3. **Communications**
4. **Storage**
5. **Accuracy**

Basic Terminology

- ✓ Hardware

  Is the physical equipment. It includes the case, keyboard, monitor, cables, storage drives, speakers, and printers

- ✓ Software

  A computer program that tells the computer how to perform particular tasks

- ✓ Input

  Whatever is put into a computer system

- ✓ Data

  Refers to the symbols that represent facts, objects, or ideas

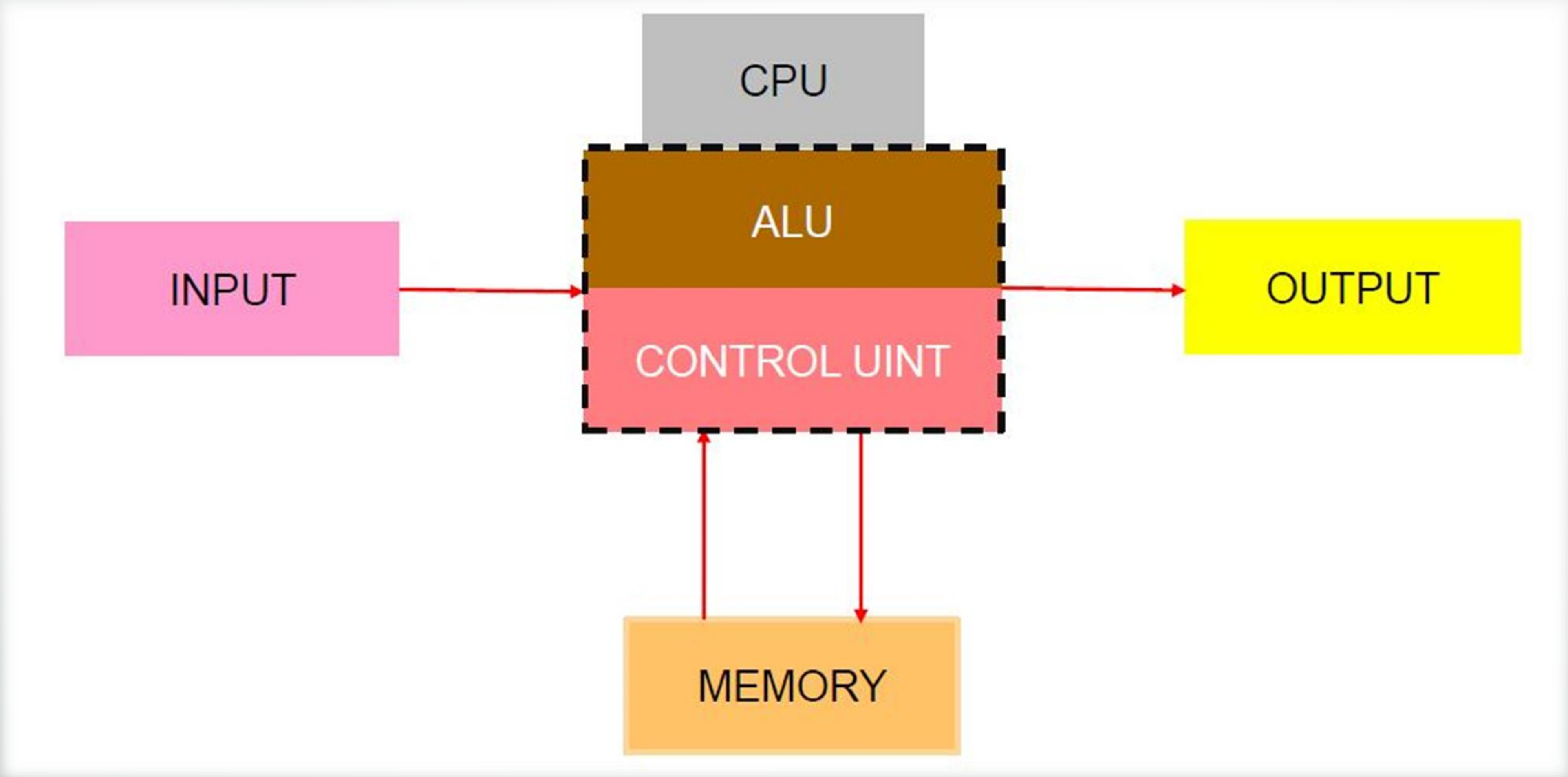
- ✓ Information

  The results of the computer storing data as bits and bytes the words, numbers, sounds, and graphics

Basic Terminology (Cont.)

- ✓ Output
  Consists of the processing results produced by a computer
- ✓ Processing
  Manipulation of the data in many ways
- ✓ Memory
  Area of the computer that temporarily holds data waiting to be processed, stored, or Output
- ✓ Storage
  Area of the computer that holds data on a permanent basis when it is not immediately needed for processing

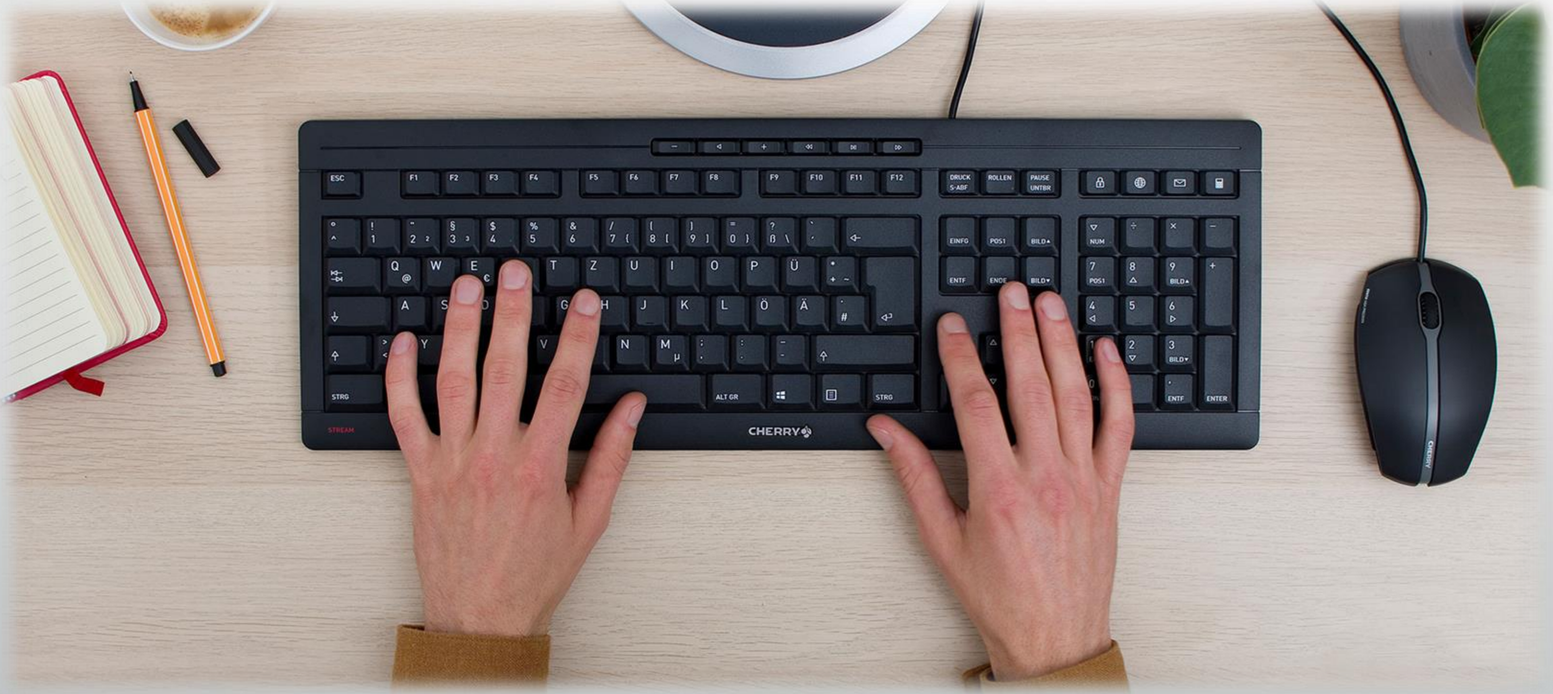
Computer Components



Computer Components (Cont.)

Here is the Definition of the most important components of computer system:

Input unit: this unit consists of all the circuits needed to get programs and data into the computer. The input section includes a keyboard, mouse, track ball, voice recognition system.



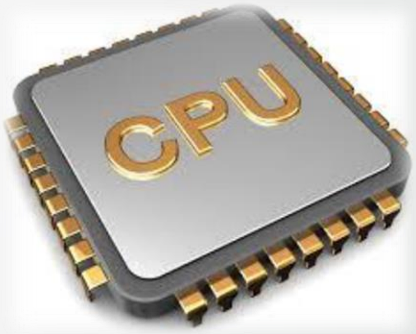
Computer Components (Cont.)

Output units: this unit process answer and other processed data to the outside world. The output section usually includes a monitor, printer, scanner.

Memory unit: this unit stores the program and data. CPU can work with the information stored in the memory unit.

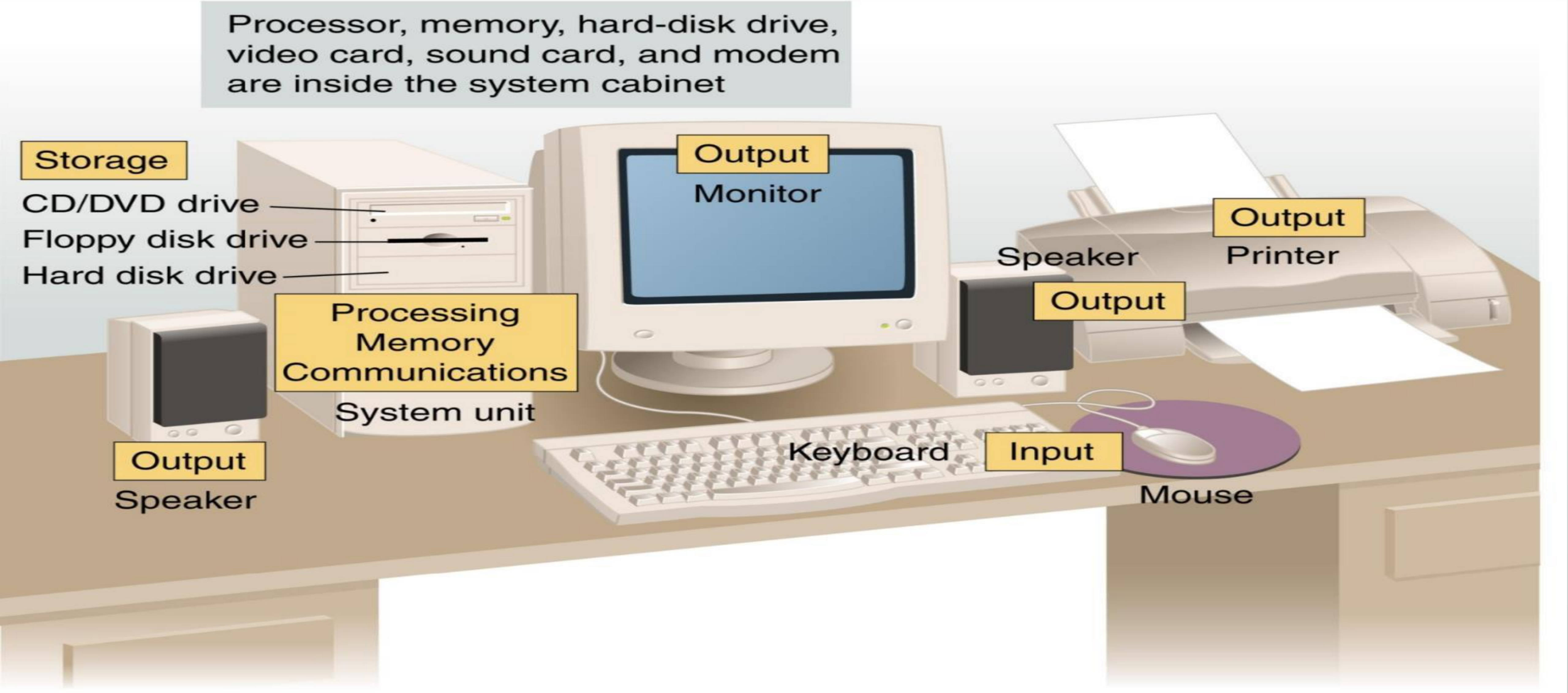
Computer Components (Cont.)

Central Processing Unit (CPU): The brain of the computer. It contains two components:



- Control Unit: the control unit coordinates and controls the other parts of the computer system
- Arithmetic/logic unit (ALU): the ALU performs the computer's principal logical and arithmetic operations. It adds, subtracts, multiples, and divides, determining whether a number is positive, negative, or zero

Computer Components (Cont.)



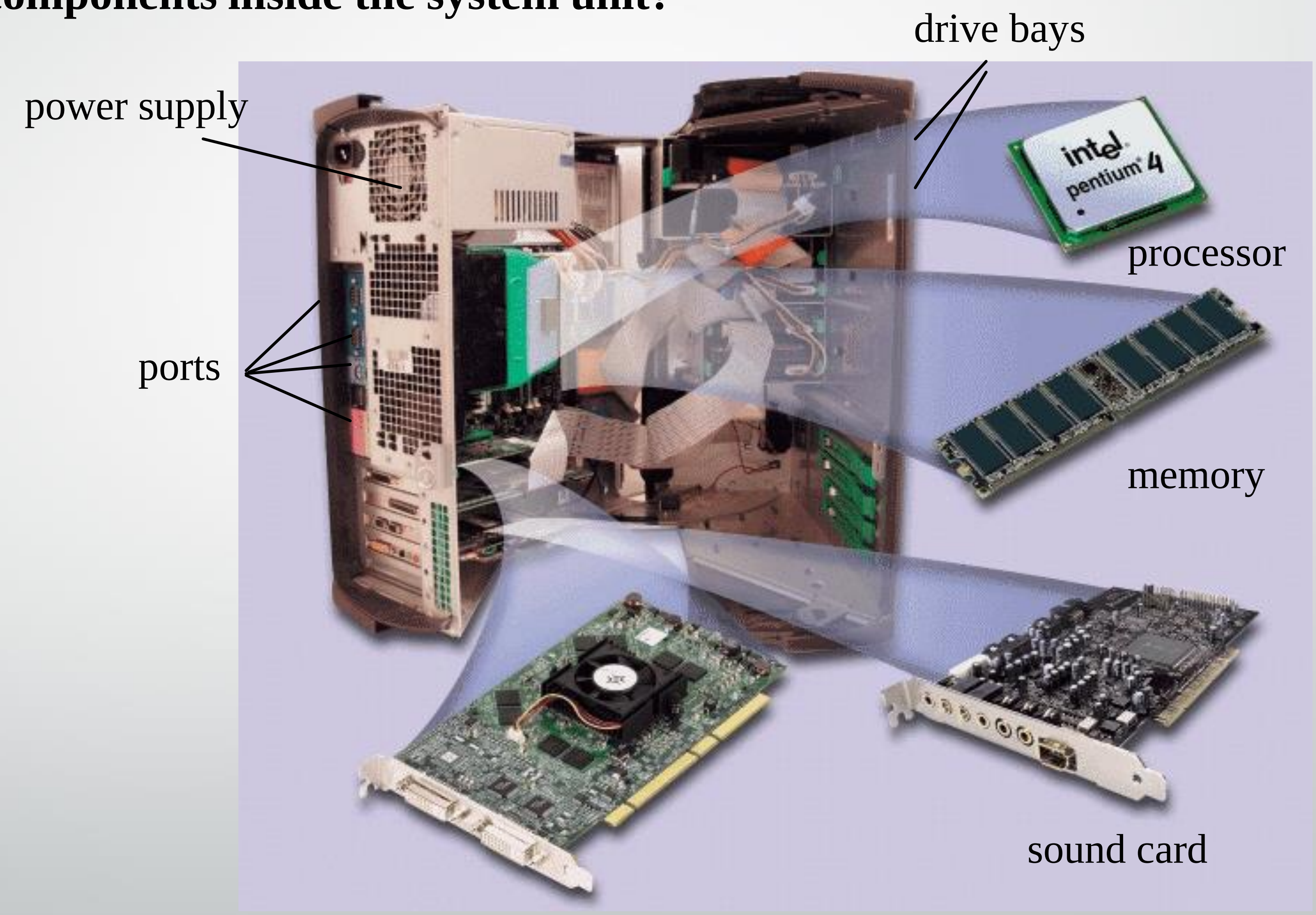
System Unit

- **What are common components inside the system unit?**

- Processor
- Memory
- Adapter cards

  - Sound card

  - Video card
- Ports
- Drive bays
- Power supply



Types of Computers

- **Supercomputers:**
- **Mainframes:**
- **Mini-Computers:**
- **Personal Computers**
  - **Desktop**
  - **Portable (Notebook/Laptop/tablet)**

Computer Organization

Lec Two: Von Neumann Architecture

Brief History

- Computer architecture has undergone unbelievable changes in the past 30 years, from the number of circuits that can be integrated onto silicon wafers to the degree of sophistication with which different algorithms can be mapped directly to a computer's hardware. One element has remained constant throughout the years, however, and that is the von Neumann concept of computer design.

What is Von Neumann Architecture?

- The basic concept behind the Von Neumann architecture is the ability to store program instructions in memory along with the data on which those instructions operate i.e. the program and data are stored in the same memory system in the same way that will enable the computer to get its instruction by reading them from memory.

What is Von Neumann Architecture? (Cont.)

- The Von Neumann architecture is a design model for a stored-program digital computer.

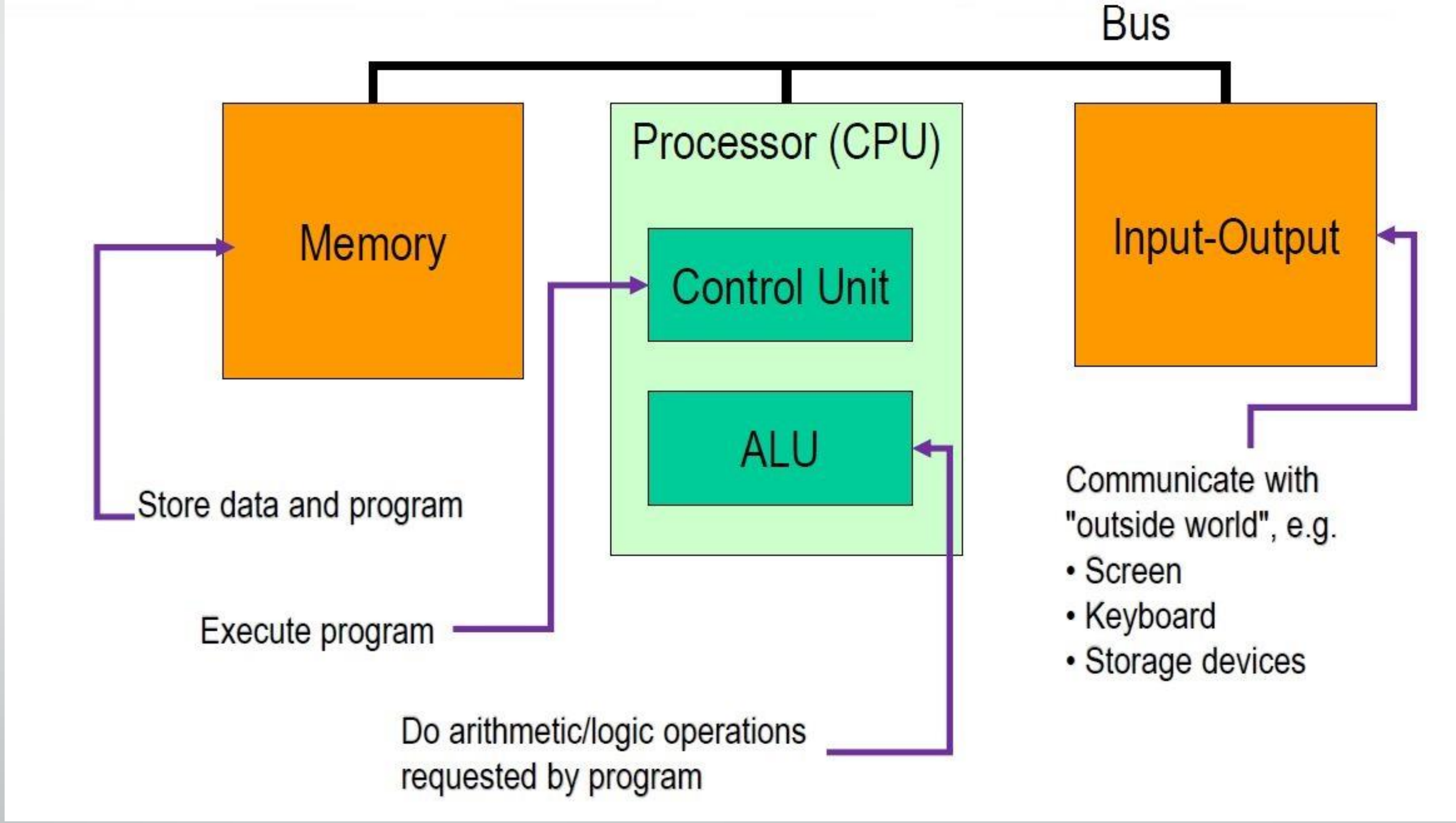
- This model (architecture) describes a general framework, or structure, that a computer's hardware, programming, and data should follow.

- All computers more or less based on the same basic design, the von Neumann Architecture! what ever it be a multi millions dollar.

Von Neumann Architecture Characteristics

- This Model is based on the following three characteristics:
  - The computer consists of four main sub-systems:
    - ✓ Input / Output System (I/O)
    - ✓ ALU (Arithmetic/Logic Unit)
    - ✓ Control Unit
    - ✓ Memory
  - Program is stored in memory during execution.
  - Program instructions are executed sequentially.

Von Neumann Components



Von Neumann Components

- **Memory**

  The most important feature is the Memory that can holds both Data and the program processing that data, this memory is called RAM (Random Access Memory).

- **Input – Output:**

  This architecture allows the users to interact with the Computer.

- **Arithmetic Logic Unit (ALU):**

  This unit performs:

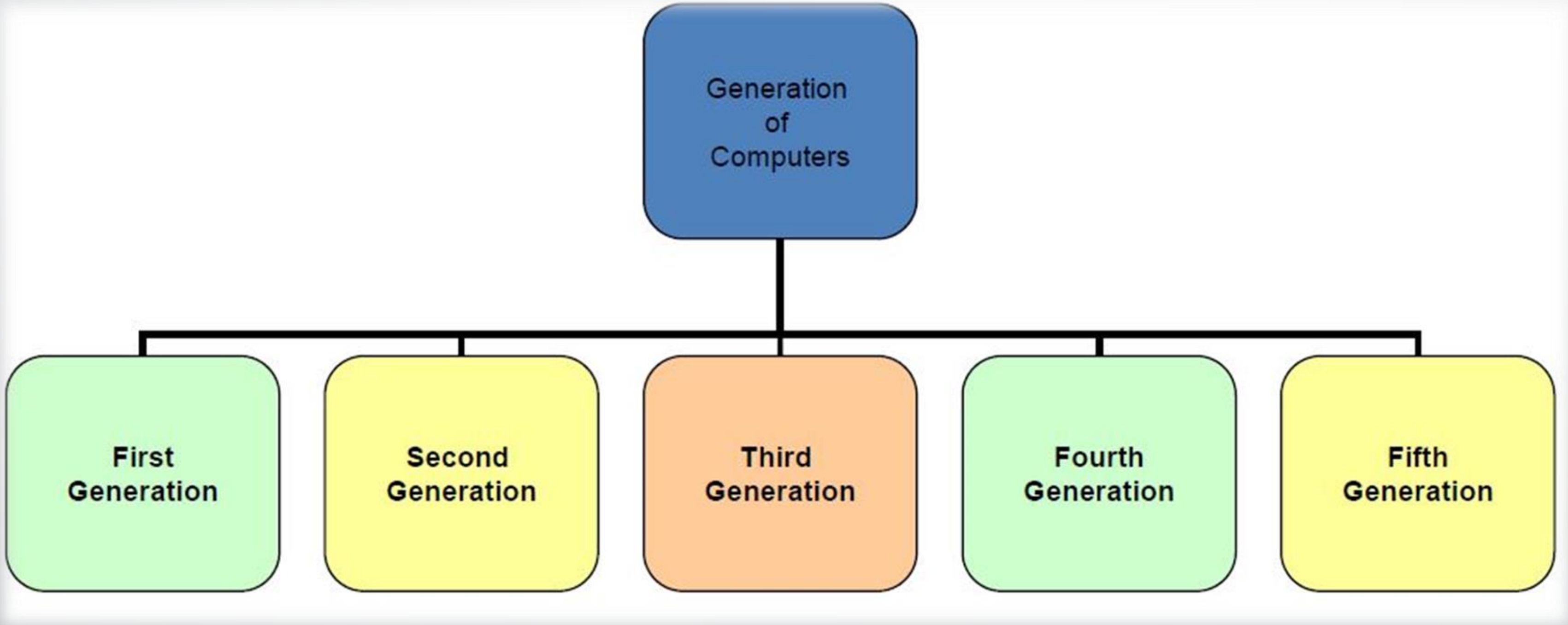
  - mathematical operations (+, -, x, /, …)
  - logic operations (=, <, >, and, or, not, ...)

- **Control Unit (CU):**

  The CU manages the process of moving data and program into and out of memory and also deal with execution of program instructions - one at a time.

Generation of Computers

Based on the characteristics of various computers developed from time to time, they are categorized as generations of computers.



First Generation Computers

Characterized By:-

Magnetic Drums
• Magnetic Tapes
• Difficult to program
• Used machine language & assembly language

Time Period : 1951 to 1959
Technology : Vacuum Tubes
Processing : Very Slow
Size : Very Large System

Second Generation Computers

Characterized By:-

- Magnetic Cores
- Magnetic Disk
- Used high level language
- Easier to program

Time Period : 1959 to 1963
Technology : Transistors
Processing : Faster
Size : Smaller

Third Generation Computers

Characterized by:-

- Minicomputers are accessible by multiple users from remote terminals.

Time Period : 1963 to 1975
Technology : ICs (Integrated Circuits)
Incorporated many transistors & electronic circuits on a single chip
Size : Small as compared to 2nd generation computers
Processing : Faster then 2nd generation computers

Fourth Generation Computers

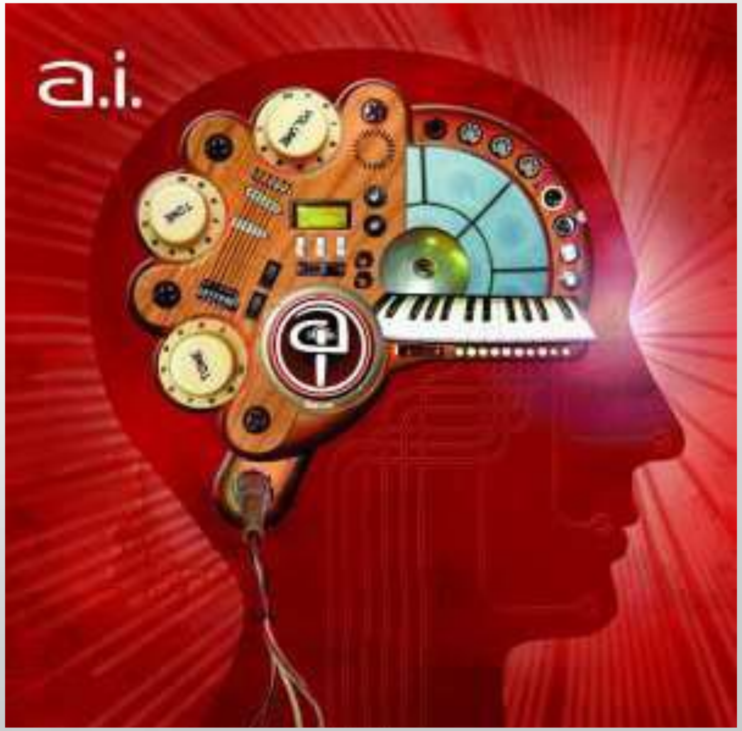
Characterized by:

- The personal computer and user friendly micro-programs, very fast processor chip high level language, OOP (Object Oriented Programming)

Time Period : 1975 to Today
Technology : VLSI (Very Large Scale Integration)
Incorporated many millions of transistors & electronic circuits on a single chip
Size : Small as compared to first generation computer
Processing : Faster then first generation computer

Fifth Generation Computers

Time Period : Future Technology
Technology : AI (Artificial Intelligence)



Computer Organization

Lec Three: Introduction to main digital component

Introduction to Digital Components

- Computers are built using logic circuits that operate on information represented by two valued electrical signals, thus represented mathematically by either 1 or 0 (The binary numbering system). The amount of information represented by such a signal is called a bit.

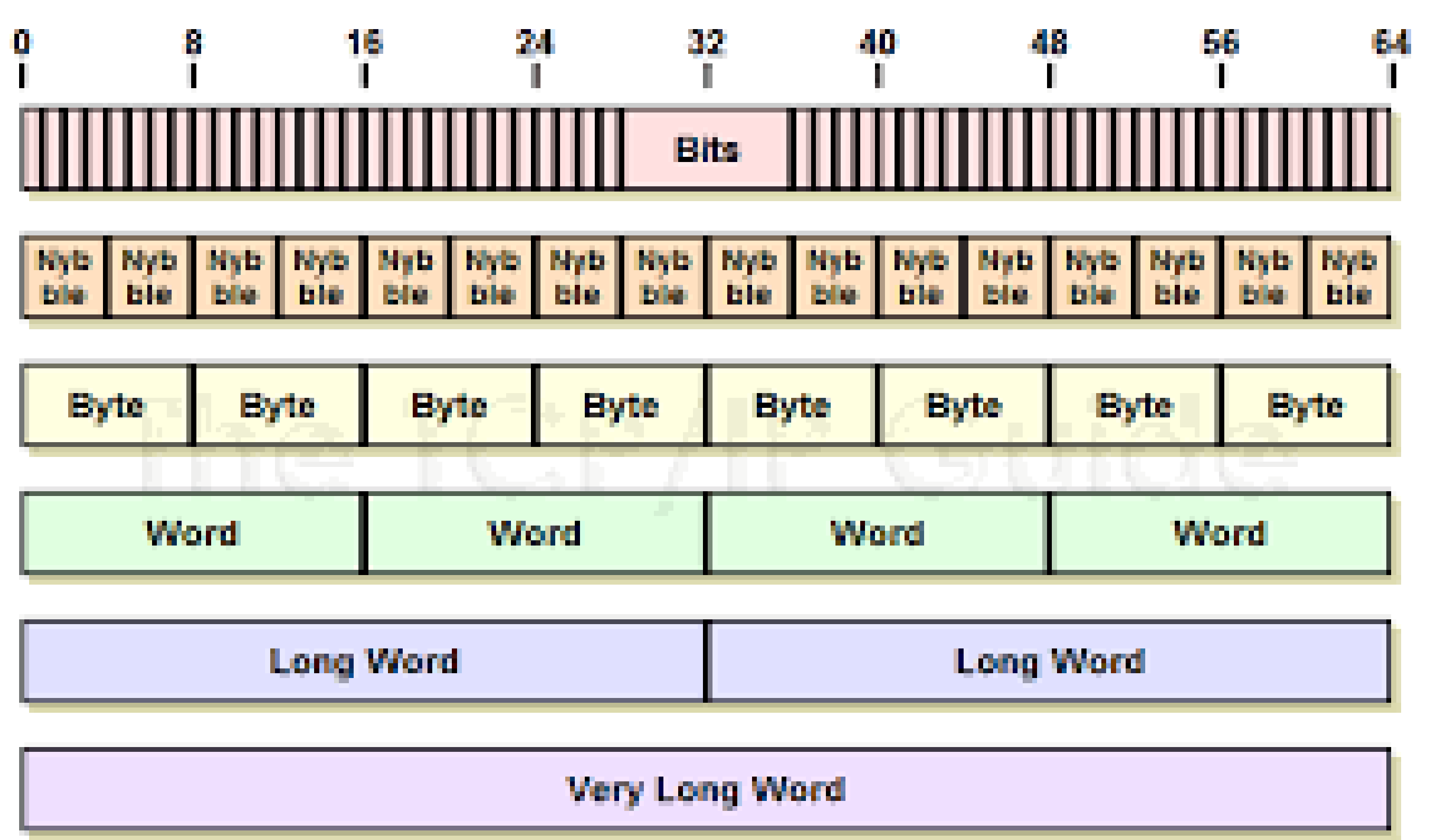
- **Bit:** Short for Binary digit, the smallest unit of information on a machine. A single bit can hold only one of two values: 0 or 1.

What Does Binary Digit (Bit) Mean?

- A binary digit, or bit, is the smallest unit of information in a computer. It is used for storing information and has a value of true/false, or on/off. An individual bit has a value of either 0 or 1, which is generally used to store data and implement instructions in groups of bytes.

- A computer is often classified by the number of bits it can process at one time or by the number of bits in a memory address. Many systems use four eight-bit bytes to form a 32-bit word.

- The value of a bit is typically stored above or below an allocated level of an electrical charge within a capacitor inside a memory module. For devices that use positive logic, value 1 (true value or high) is positive voltage relative to the electrical ground and value 0 (false value or low) is 0 voltage.


0
8
16
24
32
40
48
56
64
Bits
Nybble
Byte
Word
Long Word
Very Long Word

Binary Digit (Bit) (Cont.)

- For example, counting from zero to 10 in binary looks like this: 0, 1, 10, 11, 100, 101, 110, 111, 1000, 1001, 1010

- There is also binary code for uppercase and lowercase letters:

- A: 01000001 a: 01100001
- B: 01000010 b: 01100010
- C: 01000011 c: 01100011

- Storing a single character requires eight bits.

➢ There are many units of information that contain multiples of bits. These include:

Byte = 8 bits
Kilobit = 1,000 bits
Megabit = 1 million bits
Gigabit = 1 billion bits

Computer Digital Components

- Each bit stored within a memory cell.
- Each cell presents a flip-flop.
- **Flip-flop** is a 1 bit memory cell which can be used for storing the digital data.
- To increase the storage capacity in terms of number of bits, a group of flip-flops are used. Such a group of flip-flop is known as a **Register**. The **n-bit register** will consist of n number of flip-flop and it is capable of storing an n-bit word.

Computer Digital Components (Cont.)

- The binary data in a register can be moved within the register from one flip-flop to another. The registers that allow such data transfers are called as **shift registers**.

- Gates are combined into circuits by connecting the outputs of some gates to the inputs of others in which the logic circuits are built.

Introduction to Digital Logic Basics

- A **gate** is an electronic device with one or more inputs, each of which can assume either the value 0 or the value 1. The logical values 0 and 1 are generally represented electronically by two different voltage levels. A gate usually has one output, which is a function of its inputs, and which is also either 0 or 1.

Introduction to Digital Logic Basics (Cont.)

- The Hardware consists of a few simple building blocks
  - These are called logic gates:
    - AND, OR, NOT, …
    - NAND, NOR, XOR, …
  - Logic gates are built using transistors
  - Transistors are the fundamental devices

Binary Number

- To convert from decimal to binary, start with the binary number and keep dividing by 2, writing the remainder (of any) after each division. Keep doing this until reach one. The result, then, is the remainders, starting from the bottom.

Here's an example:

132 --------- 0
66 ---------- 0
33 ---------- 1
16 ---------- 0
8 ----------- 0
4 ----------- 0
2 ----------- 0
1

- Starting from the 1 at the bottom, the binary equivalent of 132 is 10000100.

Binary Number

- The binary number system works like the decimal number system, but it is a base 2 system. To convert binary to decimal, use the same method used above but use base 2.

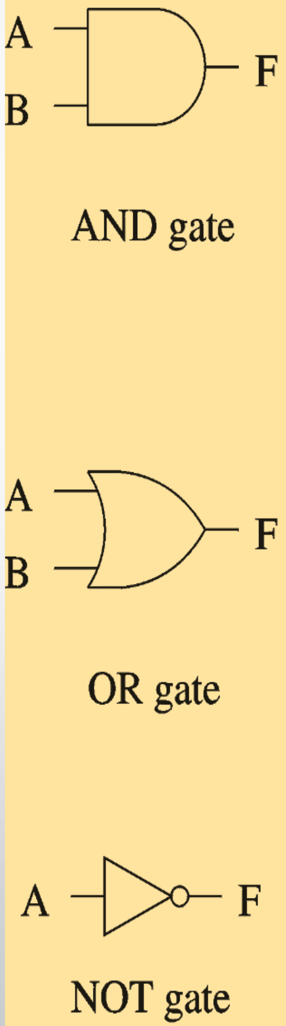
$$11010101 = 1x2^7 + 1x2^6 + 0x2^5 + 1x2^4 + 0x2^3 + 1x2^2 + 0x2^1 + 1x2^0$$
$$= 1x128 + 1x64 + 0x32 + 1x16 + 0x8 + 1x4 + 0x2 + 1x1$$
$$= 128 + 64 + 0 + 16 + 0 + 4 + 0 + 1$$
$$= 213$$

Basic Concepts

Logical operations (Logic Gates)

- Simple gates
  - AND
  - OR
  - NOT
- Functionality can be expressed by a truth table

A truth table lists output for each possible input combination



Logic symbol

A	B	F
0	0	0
0	1	0
1	0	0
1	1	1

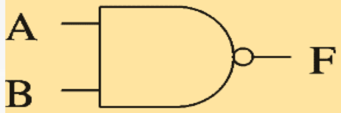
A	B	F
0	0	0
0	1	1
1	0	1
1	1	1

A	F
0	1
1	0

Truth table

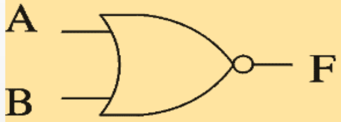
Basic Concepts (Cont.)

- Additional useful gates
  - NAND
  - NOR
  - XOR
- ➢ NAND = AND + NOT
- ➢ NOR = OR + NOT



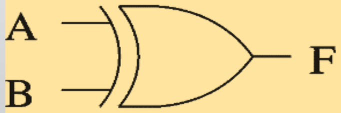
NAND gate

A	B	F
0	0	1
0	1	1
1	0	1
1	1	0



NOR gate

A	B	F
0	0	1
0	1	0
1	0	0
1	1	0



XOR gate

A	B	F
0	0	0
0	1	1
1	0	1
1	1	0

Logic symbol — Truth table

AND gate

- AND function finds the minimum between two binary digits
- AND gate outputs a 1 only if all the inputs are 1.

A	B	F
0	0	0
0	1	0
1	0	0
1	1	1

Truth table

AND gate

Graphical representation

F = A . B or F = AB

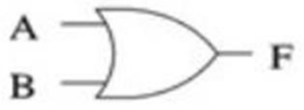
Logical representation

OR gate

- OR finds the maximum between two binary digits
- OR gate outputs a 1 if any one of the inputs is1.

A	B	F
0	0	0
0	1	1
1	0	1
1	1	1

Truth table



OR gate

Graphical representation

F = A + B

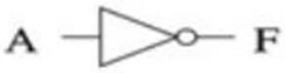
Logical representation

NOT (Inventer) gate

- Has only <u>one</u> input
- Outputs the <u>inverse</u> of the value inputted

A	F
0	1
1	0

Truth table



NOT gate

Graphical representation

$F = \overline{A}$ or A'

Logical representation

NAND gate

- NAND = AND + NOT
- The output is "false" if <u>both</u> inputs are "true." Otherwise, the output is "true."

A	B	F
0	0	1
0	1	1
1	0	1
1	1	0

Truth table

NAND gate

Graphical representation

$$F = \overline{A \, . \, B}$$

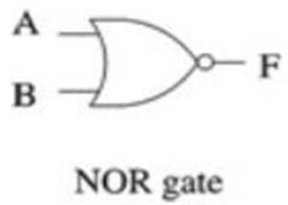
Logical representation

NOR gate

- NOR = OR + NOT
- The output is "true" if <u>both</u> inputs are "false." Otherwise, the output is "false."

A	B	F
0	0	1
0	1	0
1	0	0
1	1	0

Truth table



Graphical representation

$$F = \overline{A + B}$$

Logical representation

XOR gate

- XOR implements exclusive-OR function
- The output is "true" if either, but not both, of the inputs are "true." The output is "false" if both inputs are "false" or if both inputs are "true."

A	B	F
0	0	0
0	1	1
1	0	1
1	1	0

Truth table

XOR gate

Graphical representation

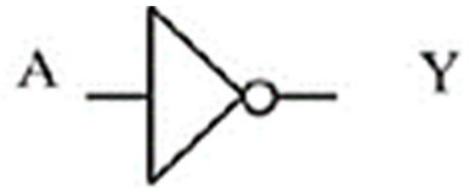
$$A \oplus B = A \cdot \overline{B} + \overline{A} \cdot B$$

Logical representation

Basic Logic Gates

NOT

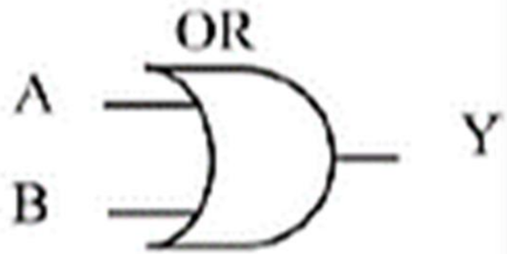
A	Y
0	1
1	0



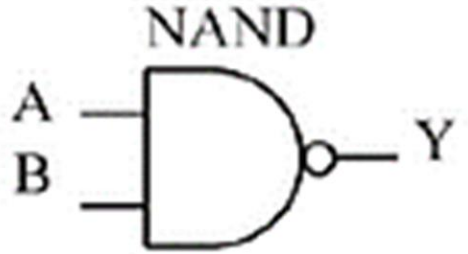
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

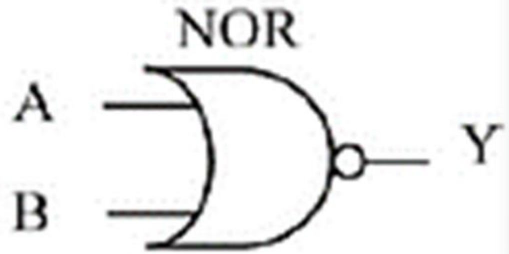
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1



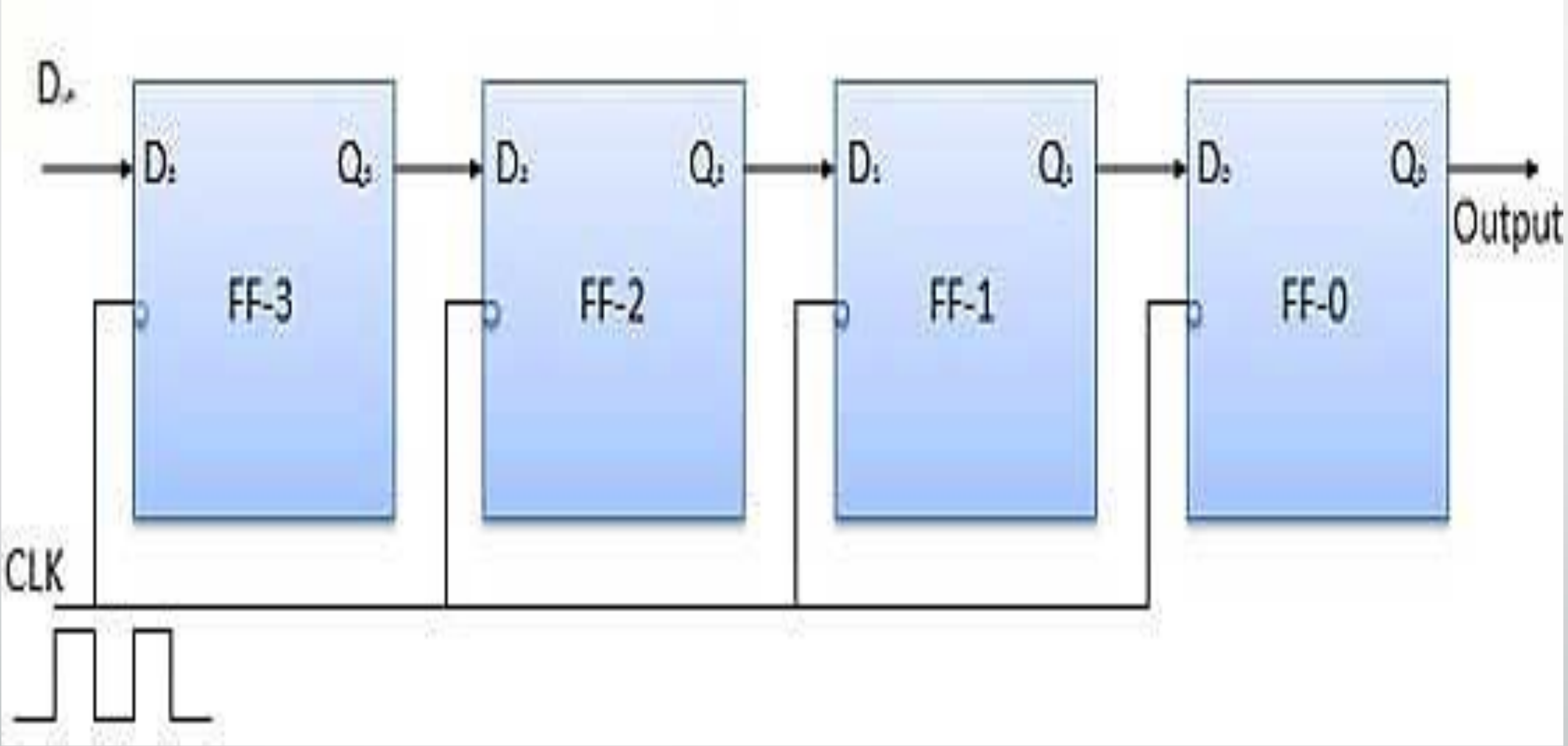
A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0



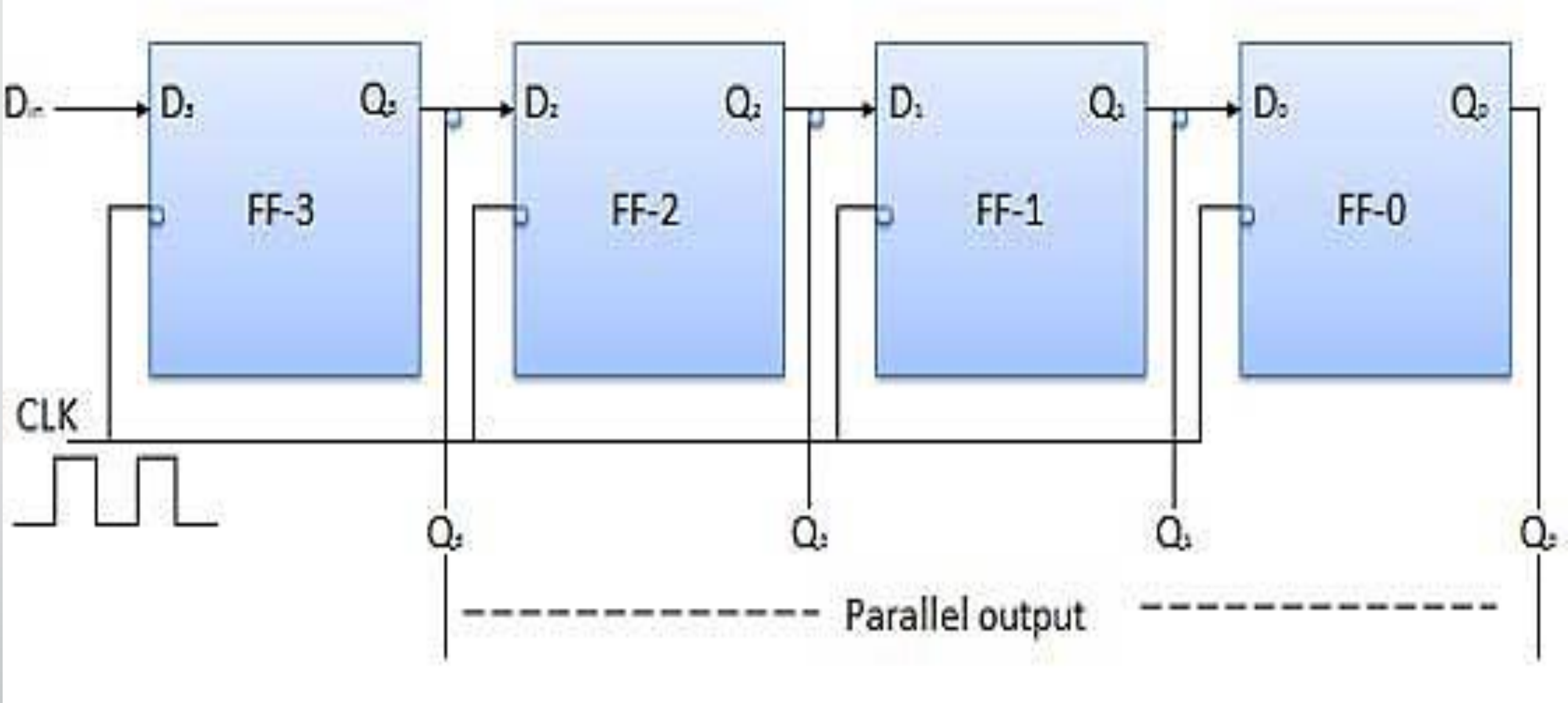
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0



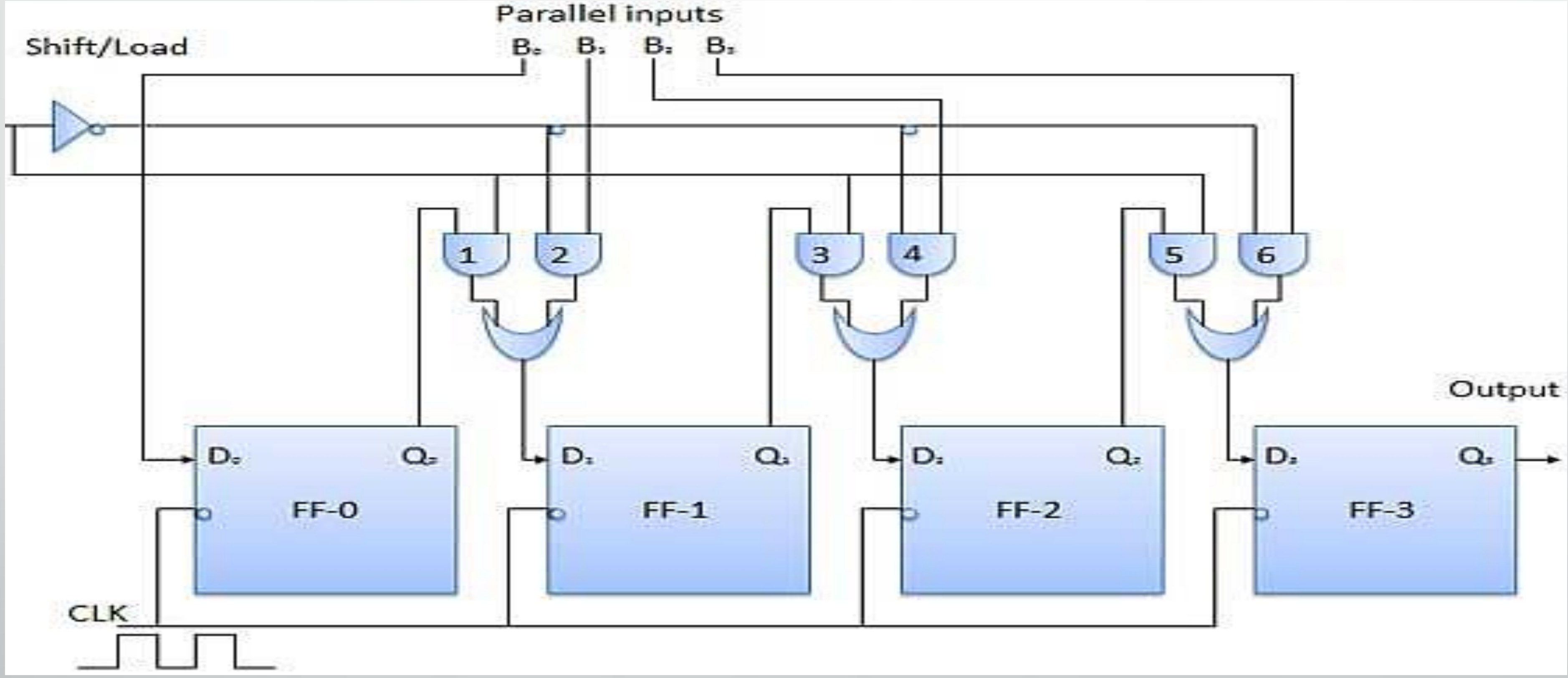
Digital Registers: Serial Input Serial Output



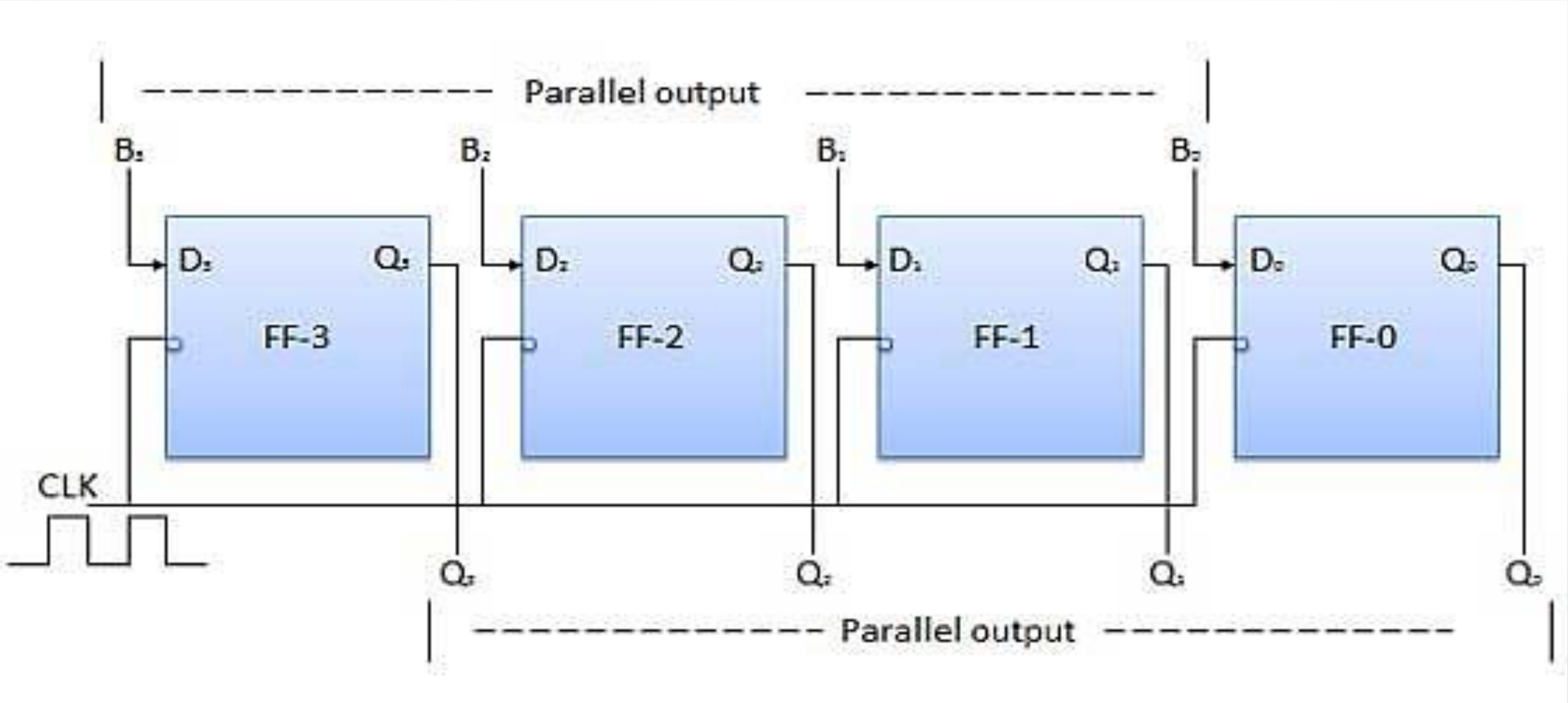
Digital Registers: Serial Input Parallel Output



Digital Registers: Parallel Input Serial Output



Digital Registers: Parallel Input Parallel Output



Computer Organization

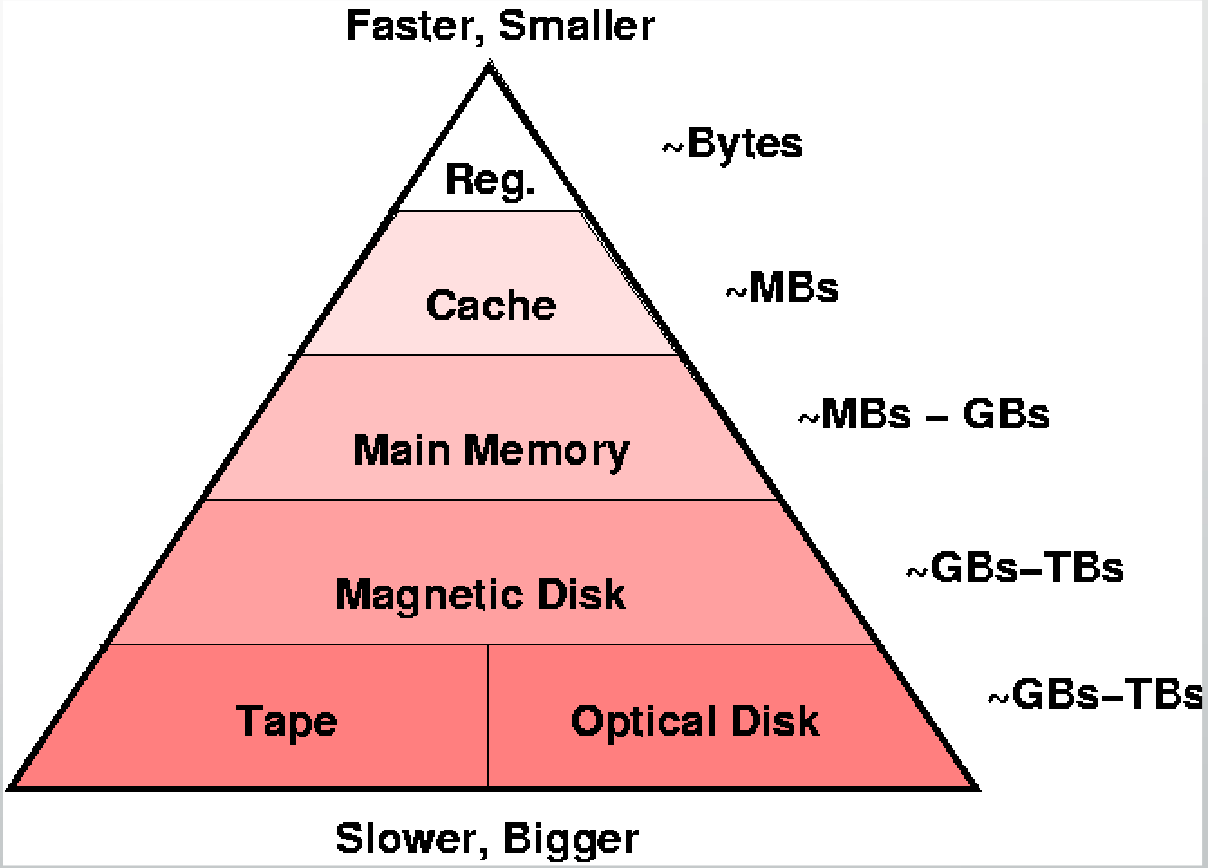
Lec Four: Memory Hierarchy

Memory Hierarchy

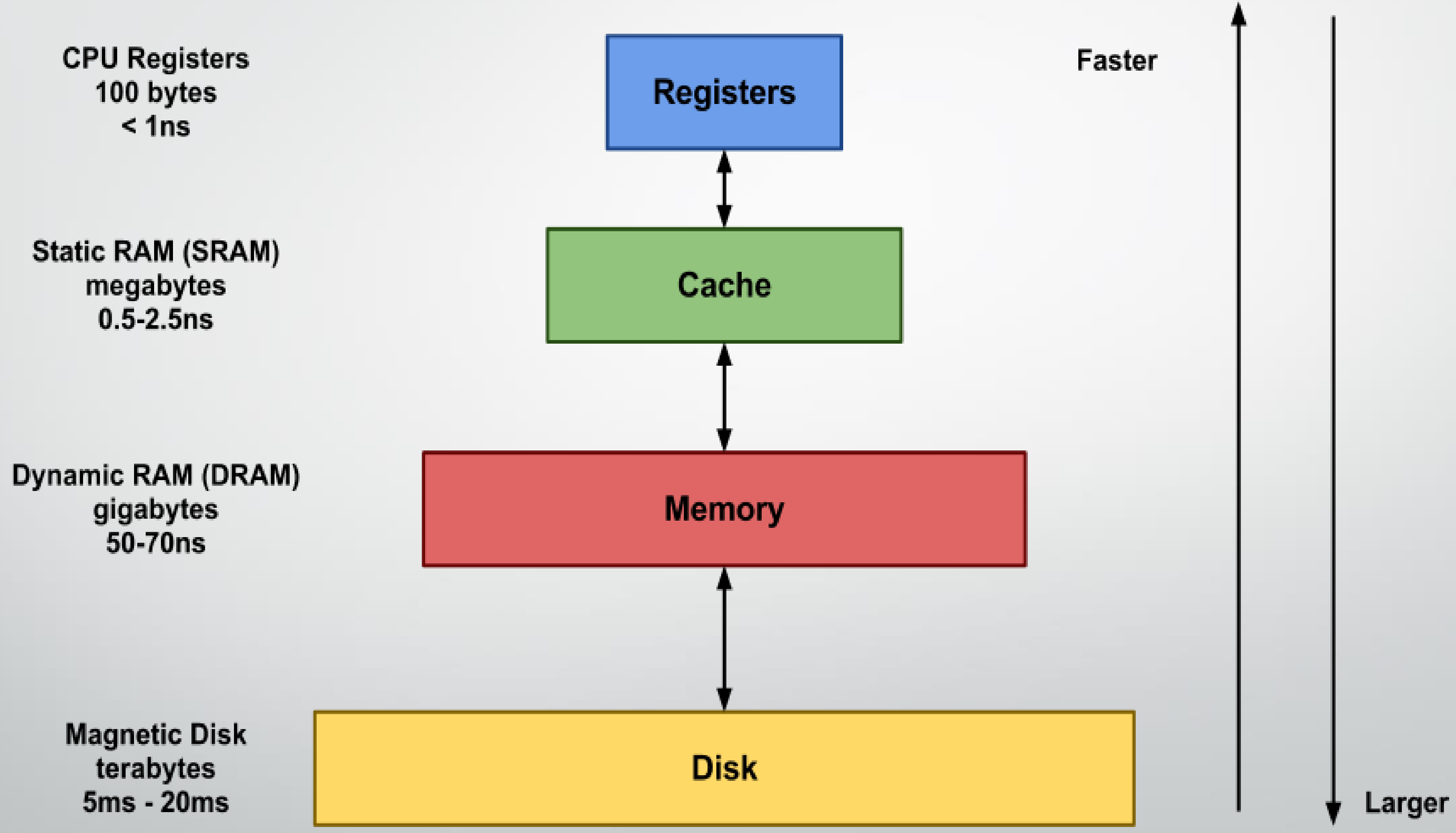
- The memory is the part of the computer that holds in formation (data and instruction) for processing. The memory hierarchy is given by:

  - **Register**
  - **Cache Memory**
  - **Main Memory**
  - **Magnetic Disk**
  - **Removable media (Magnetic tape)**

- Main memory also known as primary or internal memory or primary storage.



Memory Hierarchy (Cont.)



Memory Hierarchy (Cont.)

❖ Register:

This is a part of Central Processor Unit, so they reside inside the CPU. The information from main memory is brought to CPU and keep the information in register. Due to space and cost constraints, there is a limited number of registers in a CPU. These are basically faster devices.

❖ Cache Memory:

Cache memory is a storage device placed in between CPU and main memory. It is a semiconductor memory. It is basically fast memory device, faster than main memory. Generally, the most recently used information is kept in the cache memory. It is brought from the main memory and placed in the cache memory. Nowadays, we get CPU with internal cache.

Following is the list of some of the most common registers used in a basic computer

Reg. Symbol	No. of Bits	Register	Register Function
DR	16	Data Register	Hold operand
AR	12	Address Register	Hold address for Mem
AC	16	Accumulator	Processor register
IR	16	Instruction Register	Hold instruction code
PC	12	Program Counter	Hold address of instruct
TR	16	Temp Register	Hold temp data
INPR	8	Input Register	Holds input char
OUTR	8	Output Register	Holds output char

Memory Hierarchy (Cont.)

❖ Main Memory:

Like cache memory, main memory is also semiconductor memory. But the main memory is relatively slower memory.

The information had to first bring (whether it is data or program), to main memory. CPU can work with the information available in main memory only.

❖ Magnetic Disk:

This is bulk storage device. We have to deal with huge amount of data in many application. But we don't have so much semiconductor memory to keep these information in our computer. On the other hand, semiconductor memories are volatile in nature. It loses its content once we switch off the computer. For permanent storage, magnetic disk are used. The storage capacity of magnetic disk is very high.

Memory Hierarchy (Cont.)

- **Removable media:**

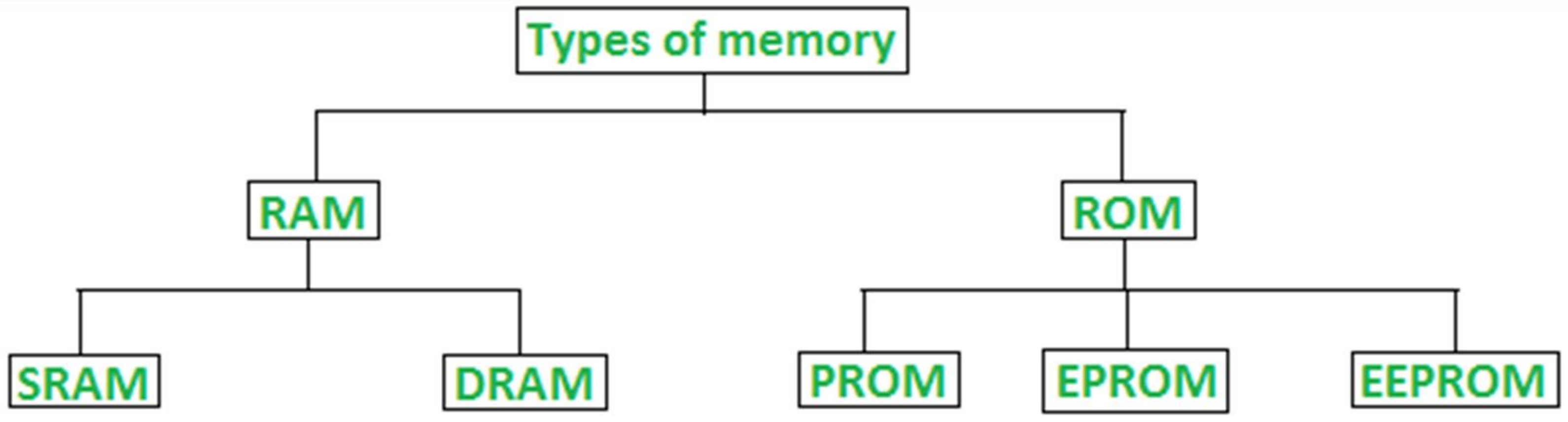
For different application, we use different data. It may not be possible to keep all the information in magnetic disk. So, which ever data we are not using currently, can be kept in removable media. Magnetic tape is one kind of removable medium CD is also a removable media, which is an optical device.

Register, cache memory and main memory are internal memory Magnetic Disk, removable media are external memory.

Internal memories are semiconductor memory: Semiconductor memories are categorized as volatile memory and non volatile memory.

Types of main memory

Memory is the most essential element of a computing system because without it computer can't perform simple tasks. Computer memory is of two basic types: Primary memory (RAM and ROM) and Secondary memory (hard drive, etc). Random Access Memory (RAM) is primary-volatile memory and Read-Only Memory (ROM) is primary-non-volatile memory.

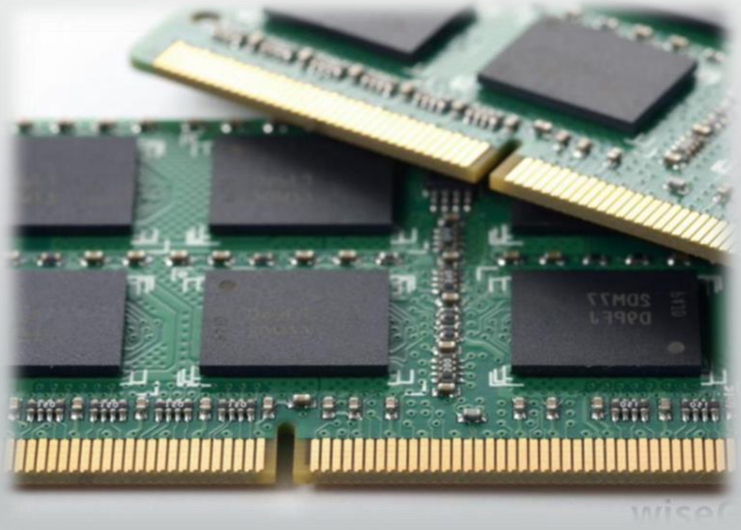


Classification of computer memory

Random Access Memory (RAM)

The RAM memory is mounted on the motherboard and it is a very important part of the computer. Computers use RAM to keep temporary data in order to complete its task.

- It is also called read-write memory or the main memory or the primary memory.
- The programs and data that the CPU requires during the execution of a program are stored in this memory.
- It is a volatile memory as the data is lost when the power is turned off.
- RAM is further classified into two types:
  SRAM (Static Random Access Memory) and DRAM (Dynamic Random Access Memory).



DRAM	SRAM
1. Constructed of tiny capacitors that leak electricity.	1.Constructed of circuits similar to D flip-flops.
2.Requires a recharge every few milliseconds to maintain its data.	2.Holds its contents as long as power is available.
3.Inexpensive.	3.Expensive.
4. Slower than SRAM.	4. Faster than DRAM.
5. Can store many bits per chip.	5. Can not store many bits per chip.
6. Uses less power.	6.Uses more power.
7.Generates less heat.	7.Generates more heat.
8. Used for main memory.	8. Used for cache.

Difference between SRAM and DRAM

Read-Only Memory (ROM)

ROM is used to store items that the computer needs to execute when it is turn on. These items are called BIOS (Basic Input Output System). BIOS used to start up the operating system. It is stored on a computer chip in a way that causes the data to remain even when power is turned off.

- Stores crucial information essential to operate the system, like the program essential to boot the computer.
- It is not volatile.
- Always retains its data.
- Used in embedded systems or where the programming needs no change.
- Used in calculators and peripheral devices.
- ROM is further classified into four types:
  MROM, PROM, EPROM, and EEPROM.

Types of Read-Only Memory (ROM)

- **MROM (Mask read-only memory):** Mask ROM cannot enable the user to change the data stored in it. If it can, the process would be difficult or slow.

- **PROM (Programmable read-only memory):** It can be programmed by the user. Once programmed, the data and instructions in it cannot be changed.

- **EPROM (Erasable Programmable read-only memory):** It can be reprogrammed. To erase data from it, expose it to ultraviolet light. To reprogram it, erase all the previous data.

- **EEPROM (Electrically erasable programmable read-only memory):** The data can be erased by applying an electric field, with no need for ultraviolet light. We can erase only portions of the chip.

Comparison between ROM and RAM

RAM (Random Access Memory)	ROM (Read only Memory)
It is a volatile memory.	*It is a non-volatile memory.*
It is a read-write memory.	*It is a read-only memory.*
It loses the data stored in it when the power is turned off	*The data inside it retains even if the power of the CPU is switched off.*
It is a temporary storage.	*It is a permanent storage.*
It is costlier than ROM.	*It is cheaper.*
It can hold a large amount of data as compared to ROM.	*It can only store small amount of data.*
It is faster.	*It is slower.*
The data in RAM can be Modified easily.	*ROM can be hardly or never be modified.*
It is used in the normal operations of a computer.	*It is used primarily in the startup process of a computer*

Auxiliary Memory

- Also called as Secondary Memory, used to store large chunks of data at a lesser cost per byte than a primary memory for backup.
- It does not lose the data when the device is powered down it is non volatile.
- It is not directly accessible by the CPU, they are accessed via the input/output channels.
- The most common form of auxiliary memory devices used in consumer systems is flash memory, optical discs, and magnetic disks, magnetic tapes.

Types of Auxiliary Memory

- ✓ **Flash memory:** (It is a special type of EEPROM) An electronic non volatile computer storage device that can be electrically erased and reprogrammed and works without any moving parts. Examples of this are USB flash drives and memory cards.

- ✓ **Optical disc:** Its a storage medium from which data is read and to which it is written by lasers. There are **three** basic types of optical disks:

  CD ROM (read only), WORM (write once read many) and EO (erasable optical disks).

External Memory or Secondary Memory:

Comprising of Magnetic Disk, Optical Disk, Magnetic Tape i.e. peripheral storage devices which are accessible by the processor via I/O Module.

Internal Memory or Primary Memory:

Comprising of Main Memory, Cache Memory & CPU registers. This is directly accessible by the processor.

Different between the primary memory and the secondary memory

Primary memory:

- The memory devices used for primary memory are semiconductor memories.
- The primary memory is categorized as volatile and non-volatile memories, RAM is the volatile memory and ROM is the non-volatile memory.
- The primary memory is composed of programs and data that are presently being used by the microprocessor.
- The primary memories are more effective and fast to interact with the microprocessor.
- Primary memory is known as main memory.
- These memories are also called as internal memory.
- Primary memory is temporary.
- Commonly used primary memory (main memory) available in the range of 512 MB to 8 GB RAMs.
- The primary memory devices are connected to the computer through “slots”.

Different between the primary memory and the secondary memory (Cont.)

Secondary memory:

- The secondary memory devices are magnetic and optical memories.
- The secondary memory is always non-volatile.
- The secondary memory is enough capable to store huge amount of information.
- The secondary memories are somewhat slow in interacting with the microprocessor, when compared with the primary memory.
- Secondary memory is known as additional memory or back memory.
- These memories are also called as external memory.
- The secondary memory is permanent.
- Generally secondary memories range between 80 GB to 4 TB Hard Disc Drives.
- The secondary memory devices are connected to the computer through Cables.

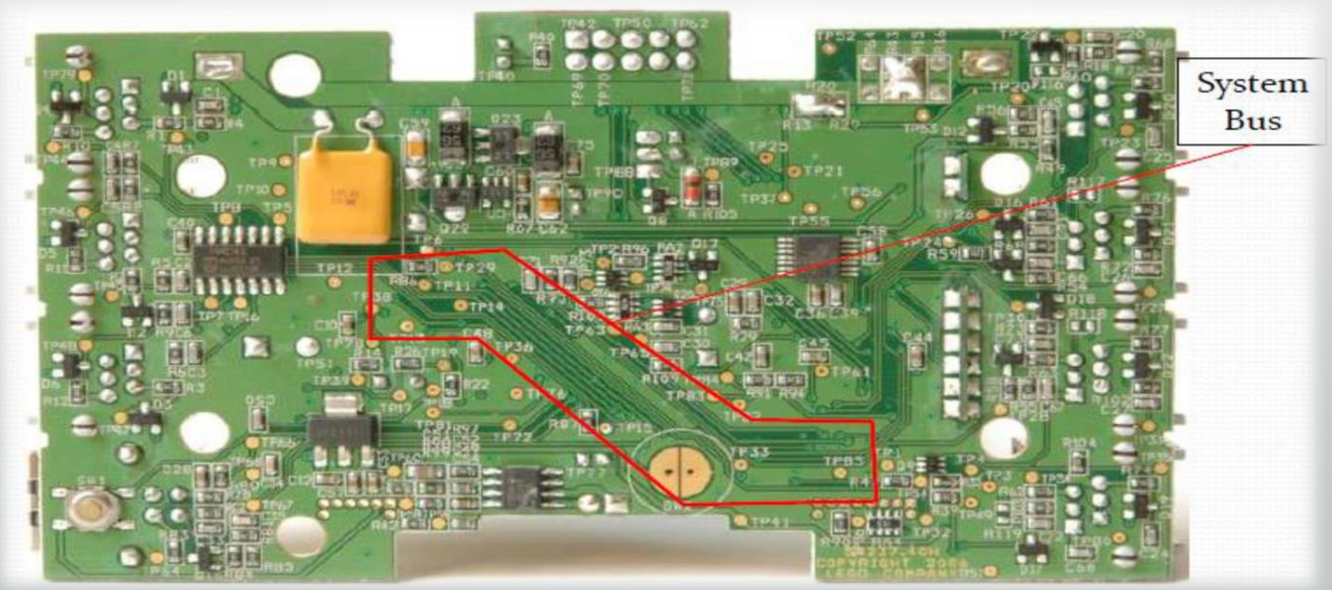
Computer Organization

Lec Five: System Buses

INTRODUCTION

The CPU sends various data values, instructions, and information to all the devices and components inside the computer. If you look at the bottom of a motherboard you'll see a whole network of lines or electronic pathways that join the different components together. This network of wires or electronic pathways is called the '**Bus**'.

A **system bus** is a single computer bus that connects the major components of a computer system, combining the functions of a data bus to carry information, an address bus to determine where it should be sent, and a control bus to determine its operation. More modern computers use a variety of separate buses adapted to more specific needs.



System Bus

A bus is a high-speed internal connection. Buses are used to send control signals and data between the processor and other components.

A bus is a communication pathway connecting two or more devices. The main property of a bus is that it is a shared transmission medium.

Multiple devices connect to the bus, and a signal transmitted by anyone device is available for reception by all other devices attached to the bus. If two devices transmit during the same time period, their signals will overlap and become garbled. Thus, only one device at a time can successfully transmit.

System Bus (Cont.)

Typically, a bus consists of multiple communication pathways or lines. Each line is capable of transmitting signals representing binary 1 and binary 0.

Several lines of a bus can be used to transmit binary digits in parallel.

For example, an 8 bit unit of data can be transmitted over eight bus lines.

System bus usually is separated into **three functional groups:**

1. **Data Bus**
2. **Address Bus**
3. **Control Bus**

The components of the bus are:

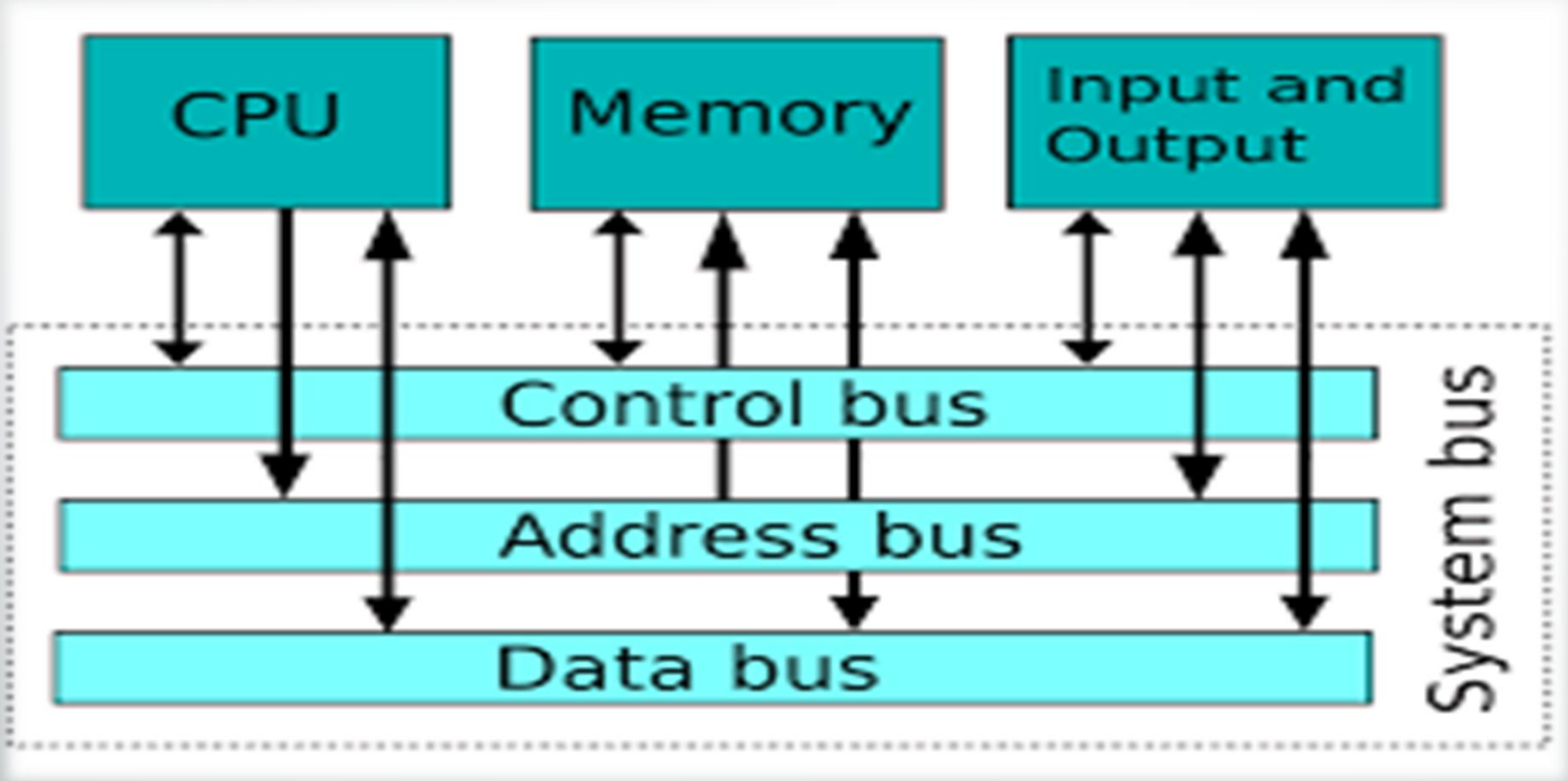
- **Cables**: Which are used to transmit electricity.
- **License plate:** That has a circuit of roads, tracks, and other elements printed with materials that are conductive or not of electricity. Its mission is to drive and deliver data between components.
- **Resistors:** They are electronic components that are designed to work as a resistance, that is, it analyzes the transmission that exists between two points of the electrical frequency, generating a control of the maximum current that passes through it.
- **Capacitors:** They work to be able to store energy and at the same time transmit it to different elements that need electricity immediately.

Types of Transfer

- **Memory to Processor:** The processor reads an instruction or a unit of data from memory.
- **Processor to Memory:** The processor writes a unit of data to memory.
- **I/O to Processor:** The processor reads data from an I/O device via an I/O module.
- **Processor to I/O:** The processor sends data to the I/O device.
- **I/O to or from Memory:** For these two cases, an I/O module is allowed to exchange data directly with memory, without going through the processor, using direct memory access (DMA).

Direct memory access (DMA): is a method that allows an input/output (I/O) device to send or receive data directly to or from the main memory, by passing the CPU to speed up memory operations.

SYSTEM BUS MODEL



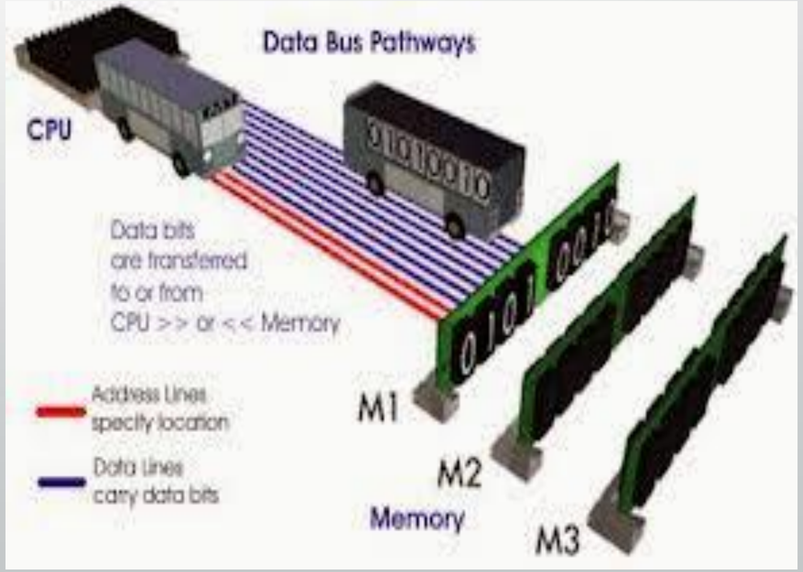
DATA BUS

A collection of wires through which data is transmitted from one part of a computer to another. Data Bus can be thought of as a highway on which data travels within a computer.

This bus connects all the computer components to the CPU and main memory. The data bus may consist of 32, 64, 128, or even more separate lines.

The number of lines being referred to as the width of the data bus. Because each line can carry only 1 bit at a time, the number of lines determines how many bits can be transferred at a time.

It is a **bidirectional** bus.

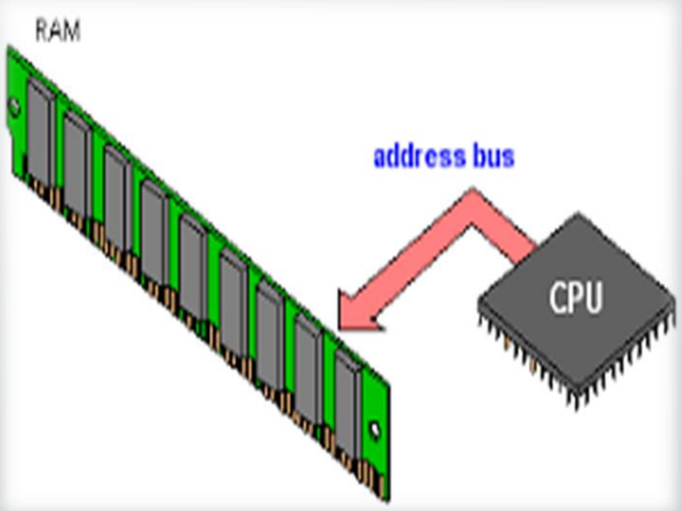


ADDRESS BUS

Carries memory addresses from the processor to other components such as primary storage and input/output devices. In addition, a collection of wires used to identify the particular location in the main memory is called **Address Bus**.

Clearly, the width of the address bus determines the maximum possible memory capacity of the system. For example, a system with a **32-bit** address bus can address $\mathbf{2^{32}}$ (4,294,967,296) memory locations. If each memory location holds one byte, the addressable memory space is 4 GB.

The address bus consists of 16, 20, 24, or more parallel signal lines.



It is a **unidirectional** bus.

ADDRESSABILITY

- Data is stored in storage locations.
- Each piece data which can be stored in its own storage location in main memory is called a **Word.**
- A word is the number of bits that the processor can process in a single operation (clock pulse).
- Each storage location has its own unique address.
- The method used to identify each unique address is called addressability.

Control Bus

Carries control signals from the processor to other components. The control bus also carries the clock's pulses.

In computer architecture, A **control bus** is part of the system bus, used by CPUs for communicating with other devices within the computer.

While the address bus carries the information about the device with which the CPU is communicating and the data bus carries the actual data being processed, the control bus carries commands from the CPU and returns status signals from the devices.

It is an **unidirectional** bus.

Clock

The CPU contains a clock which, along with the control unit, is used to coordinate all of the computer's components. The clock sends out a regular electrical pulse which synchronizes (keeps in time) all the components.

The frequency of the pulses is known as **clock speed**. Clock speed is measured in **hertz (Hz)**. The greater the speed, the more instructions can be performed in any given moment of time.

In the 1980s, processors commonly ran at a rate of between 3 **megahertz (MHz)** and 5 MHz, which is 3 million to 5 million pulses or cycles per second. Today, processors commonly run at a rate of between 3 **gigahertz (GHz)** and 5 GHz, which is 3 billion to 5 billion pulses or cycles per second.

Control Bus (Cont.)

Typical control bus signals are :

Memory Read: causes data from the addressed location to be placed on the data bus.

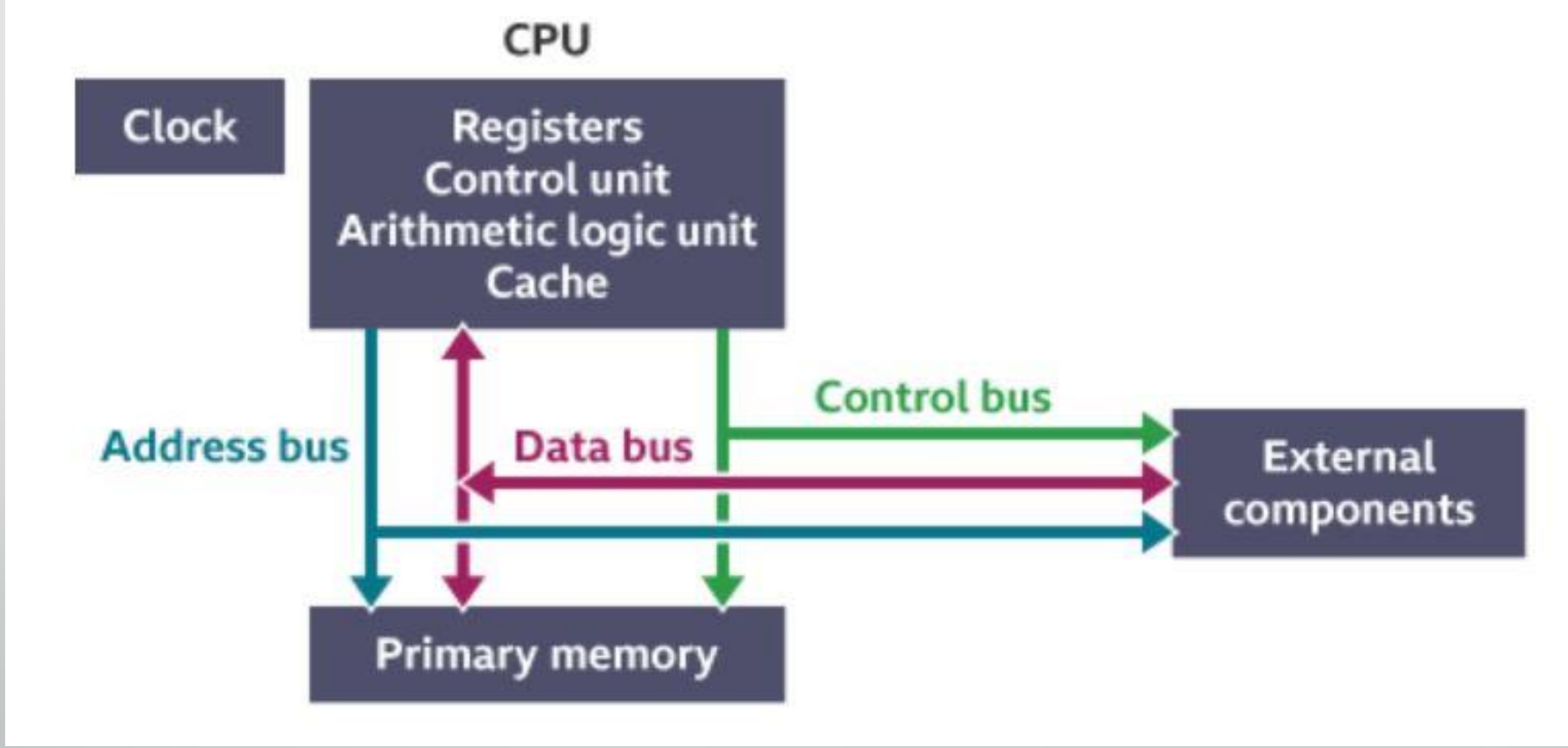
Memory Write: causes data on the bus to be written into the addressed location.

I/O write: causes data on the bus to be output to the addressed I/O port.

I/O read: causes data from the addressed I/O port to be placed on the bus.

Common CPU components

- All the components work together to allow processing and system control.



RECOMMENDED QUESTIONS

1. Define the system bus, and give the functional groups.
2. What are the types of transfers in the bus system?
3. Draw the block diagram of system the bus model.
4. Define the data bus, address bus and direct memory access.
5. List the functions of control bus and explain briefly.

Computer Organization

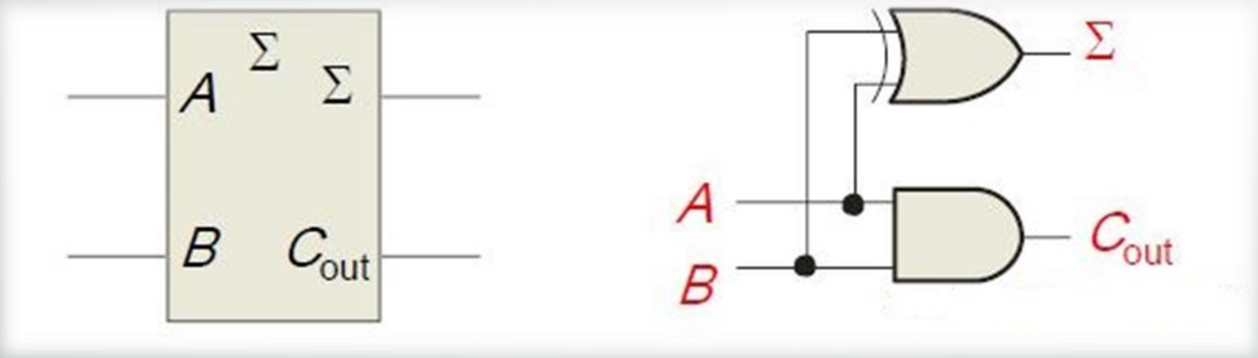
Lec Six: Introduction to Main Digital Component (Part 2)

Combinational logic Functions

Half Adder

- Arithmetic Operations: Binary Addition
- Basic rules of binary addition are performed by a **half adder**, which has two binary inputs **(A and B)** and two binary outputs **(Carry out and Sum).**
- The inputs and outputs can be summarized on a truth table.
- The logic symbol and equivalent circuit are:

Inputs		Outputs	
A	B	C_{out}	Σ
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

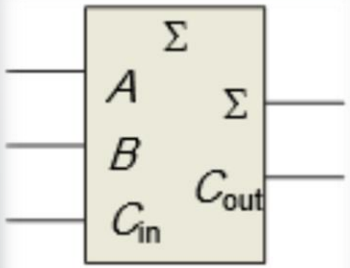


Full-Adder

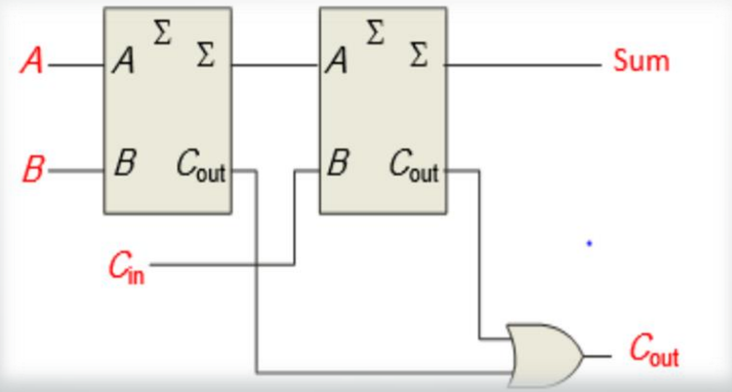
- By contrast, a **full adder** has three binary inputs **(A, B, and Carry in)** and two binary outputs **(Carry out and Sum).**

- The truth table summarizes the operation.

- A full–adder can be constructed from two half adders as shown:



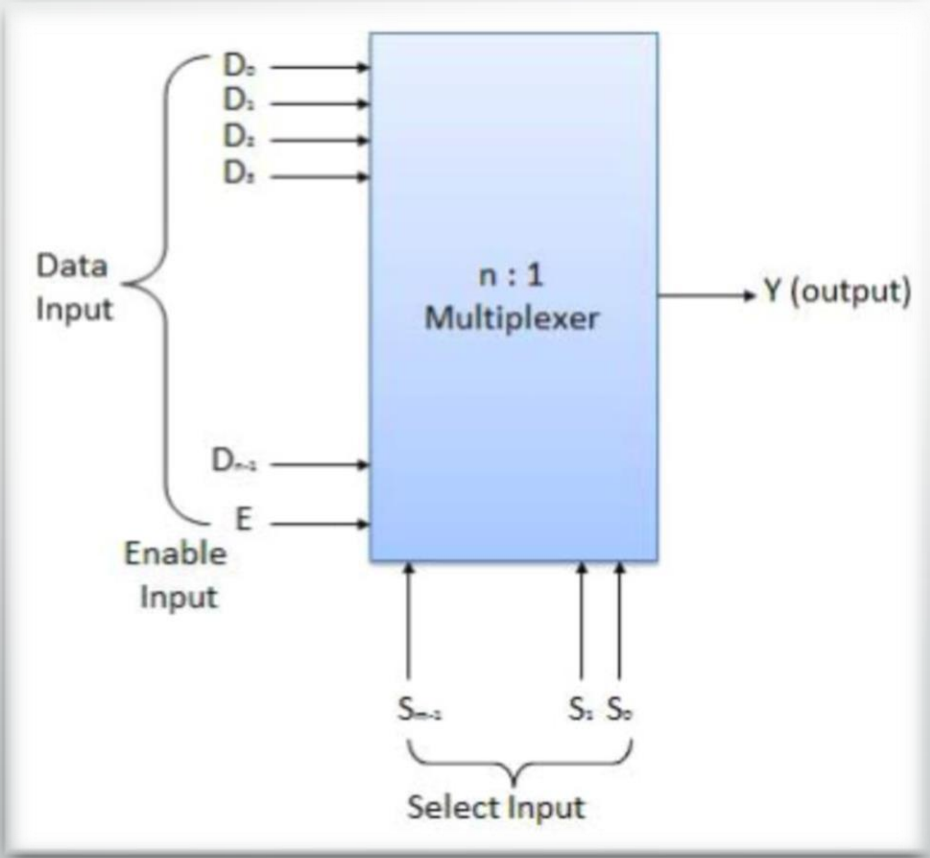
Symbol



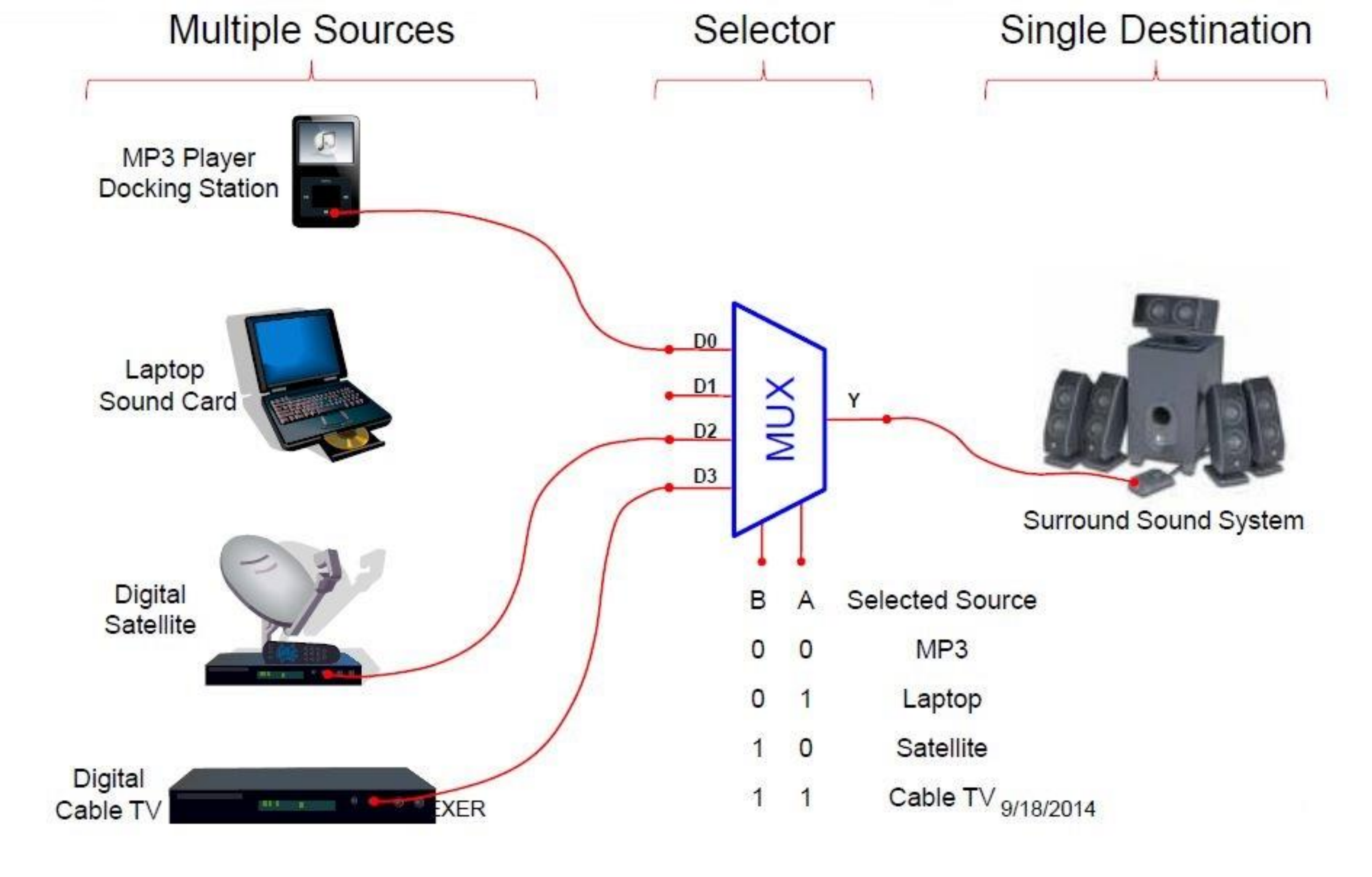
Inputs			Outputs	
A	B	C_{in}	C_{out}	Σ
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Multiplexer

- A **MULTIPLEXER** is a digital circuit that has multiple inputs and a single output. The selection of one of the n inputs is done by the select inputs.



Application of Multiplexer

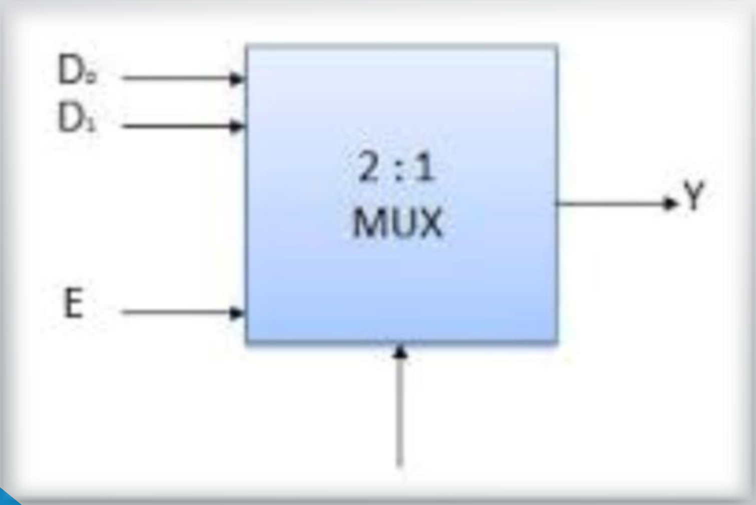


Multiplexer Types

- ✓ 2 to 1 (1 select line)
- ✓ 4 to 1 (2 select lines)
- ✓ 8 to 1 (3 select lines)
- ✓ 16 to 1 (4 select lines)

Example/ Here 2:1 means 2 inputs and 1 output

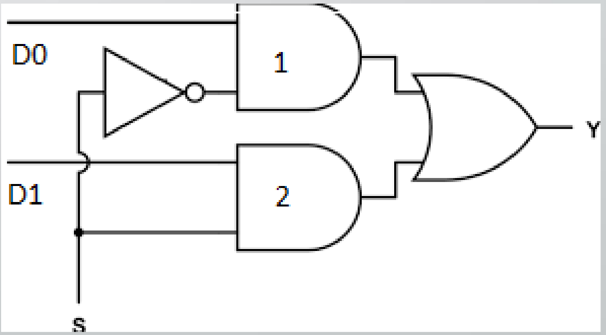
BLOCK DIAGRAM



TRUTH TABLE

S	OUTPUT Y
0	D0
1	D1

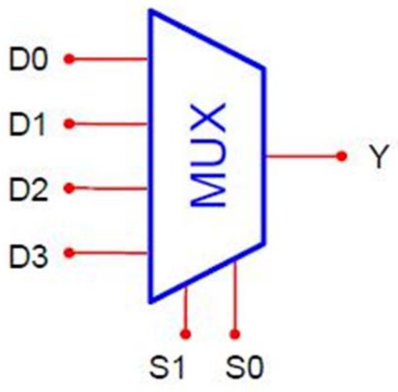
The output,
Y=D0S’+D1S



Multiplexer Types (Cont.)

- ✓ **4:1** MUX has 4 inputs (D0, D1, D2, D3) & 2 select lines (S0, S1)

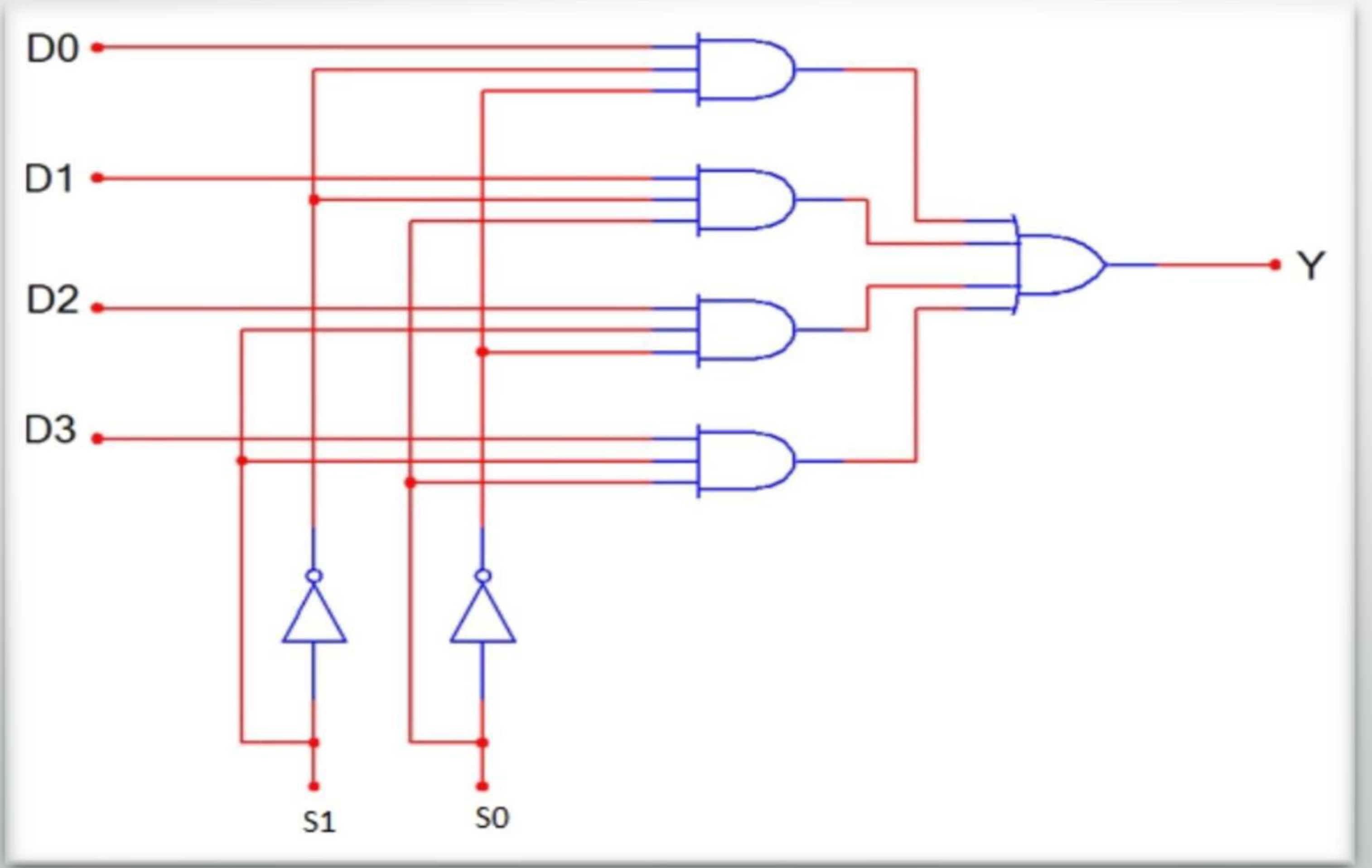
BLOCK DIAGRAM



TRUTH TABLE

S1	S0	Y
0	0	D0
0	1	D1
1	0	D2
1	1	D3

- ✓ The output, Y=S1’ S0’ D0 + S1’ S0D1 + S1S0’ D2 + S1S0D3



Demultiplexer

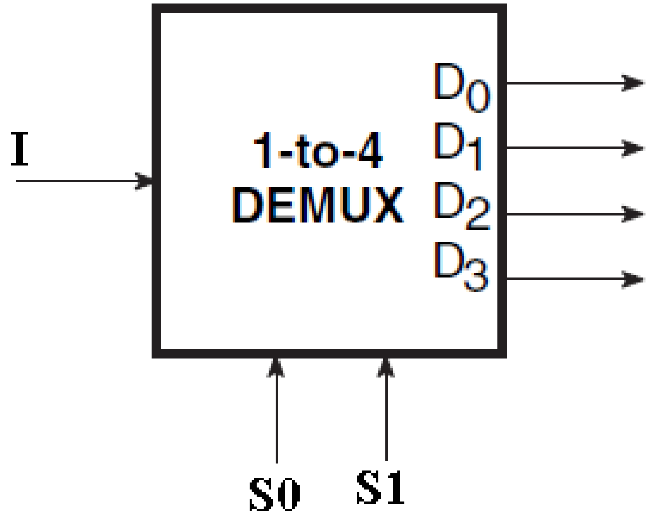
- A demultiplexer (DEMUX) performs the opposite function from a MUX. It switches data from one input line to two or more data lines depending on the select inputs.

- De-multiplexer is used to connect a single source to multiple destinations, like ALU (Arithmetic Logic Unit).

- Types of de-multiplexer are:

  - ✓ 1 to 2
  - ✓ 1 to 4
  - ✓ 1 to 8
  - ✓ 1 to 16

1-to-4 Demultiplexer



I	Select		O/P			
	S0	S1	D_0	D_1	D_2	D_3
1	0	0	1	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	1	0
1	1	1	0	0	0	1

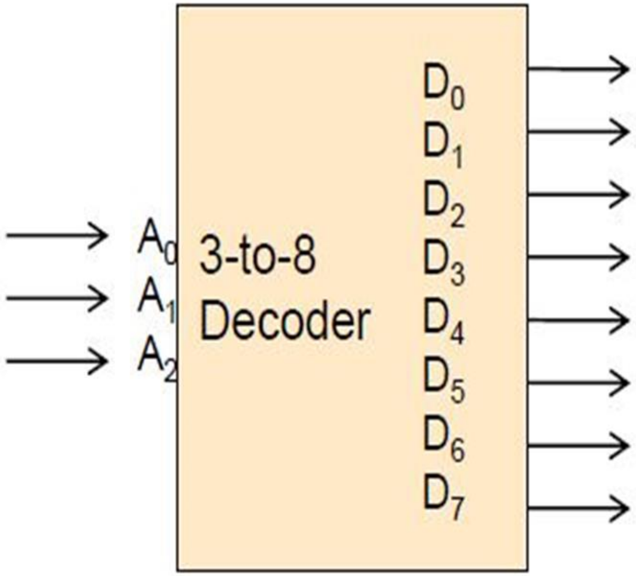
Decoder

- A **decoder** is a logic circuit that accepts a set of inputs that represents a binary number and activates only the output that corresponds to the input number, all other outputs remain inactive.

- A decoder has:
  - N inputs
  - 2^N outputs

- A decoder selects one of 2^N outputs by decoding the binary value on the N inputs.

3-to-8 Decoder



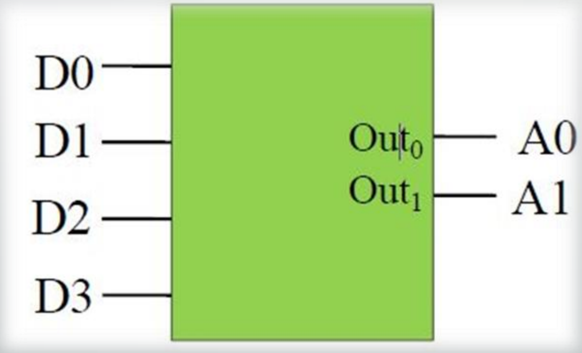
A_2	A_1	A_0	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

Encoder

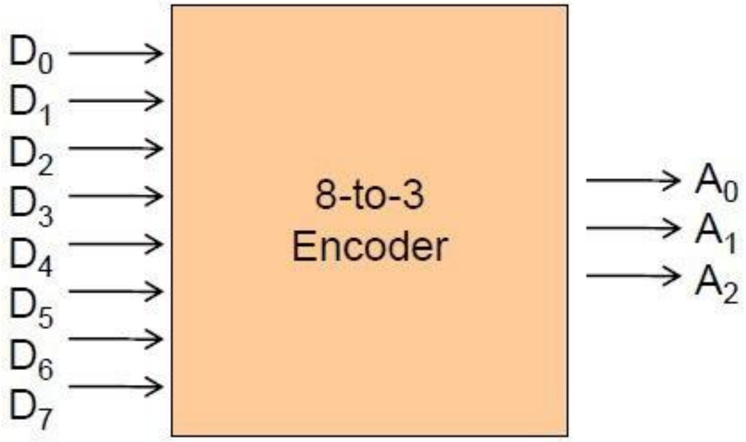
- An encoder has:
  - 2^N inputs
  - N outputs

- An encoder performs the inverse operation of a decoder, It outputs the binary value of the selected (or active) input.

- Encoders only work when **exactly one** binary input is equal to 1.
- 4 to 2 Encoder:

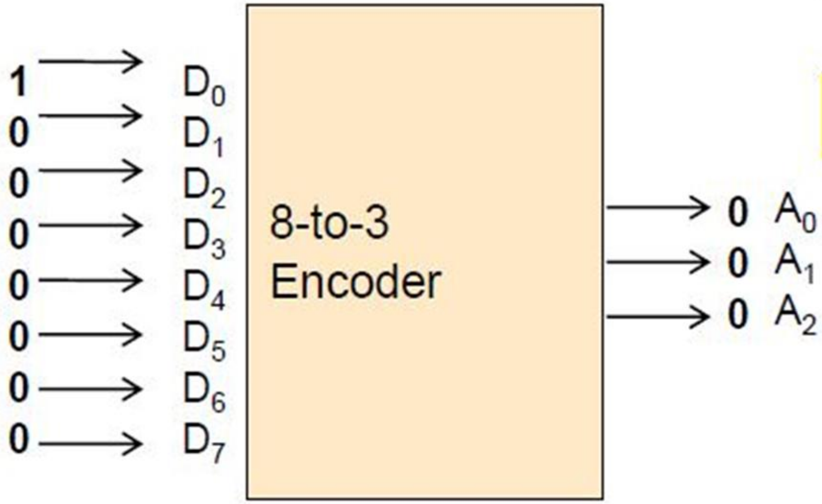


8-to-3 Encoder (truth table)

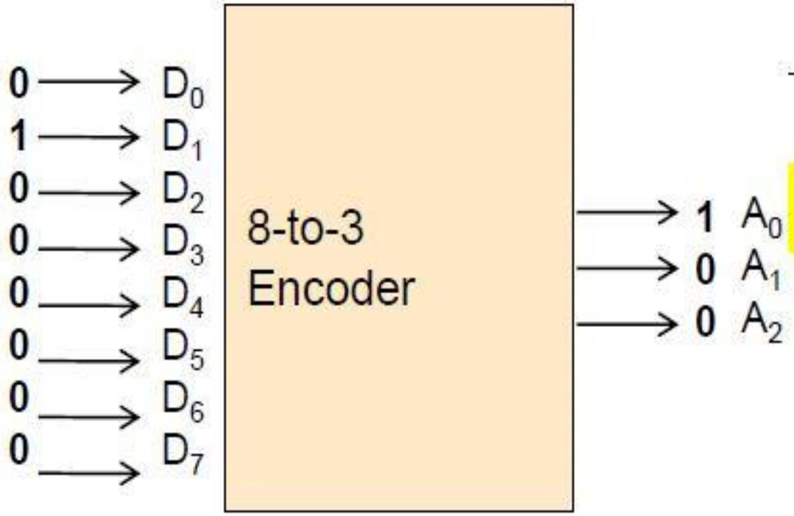


inputs								outputs		
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	A_2	A_1	A_0
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	1	0	1
0	1	0	0	0	0	0	0	1	1	0
1	0	0	0	0	0	0	0	1	1	1

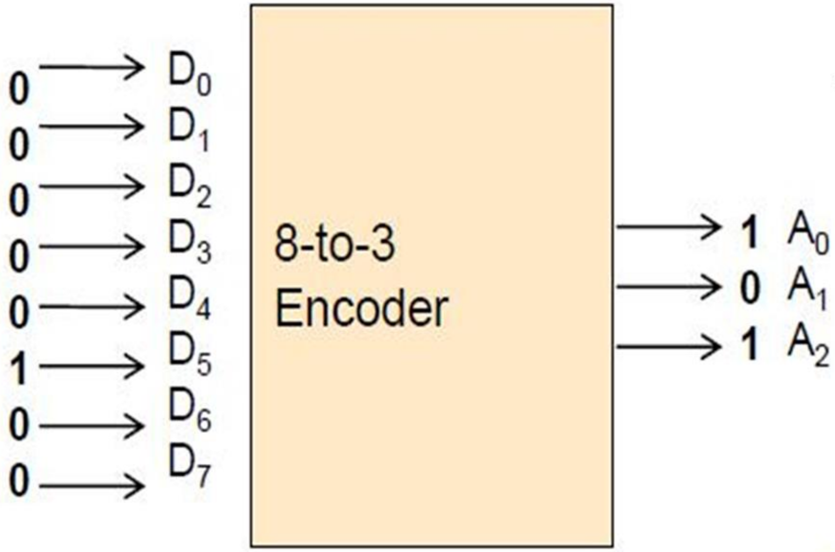
8-to-3 Encoder (truth table): Example



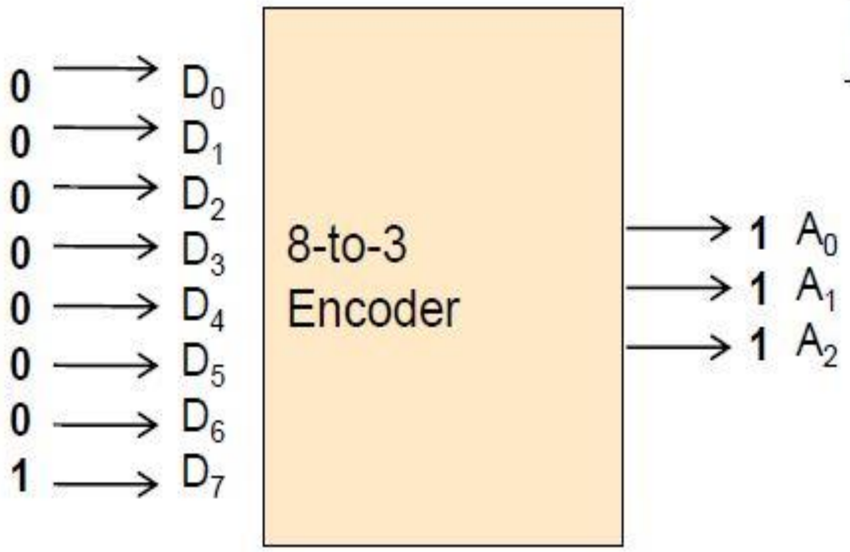
inputs								outputs		
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	A_2	A_1	A_0
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	1	0	1
0	1	0	0	0	0	0	0	1	1	0
1	0	0	0	0	0	0	0	1	1	1



inputs								outputs		
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	A_2	A_1	A_0
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	1	0	1
0	1	0	0	0	0	0	0	1	1	0
1	0	0	0	0	0	0	0	1	1	1



inputs								outputs		
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	A_2	A_1	A_0
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	1	0	1
0	1	0	0	0	0	0	0	1	1	0
1	0	0	0	0	0	0	0	1	1	1



inputs								outputs		
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	A_2	A_1	A_0
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	1	0	1
0	1	0	0	0	0	0	0	1	1	0
1	0	0	0	0	0	0	0	1	1	1

Computer Organization

Lec Seven: Memory addressing, Memory organization and Expansion

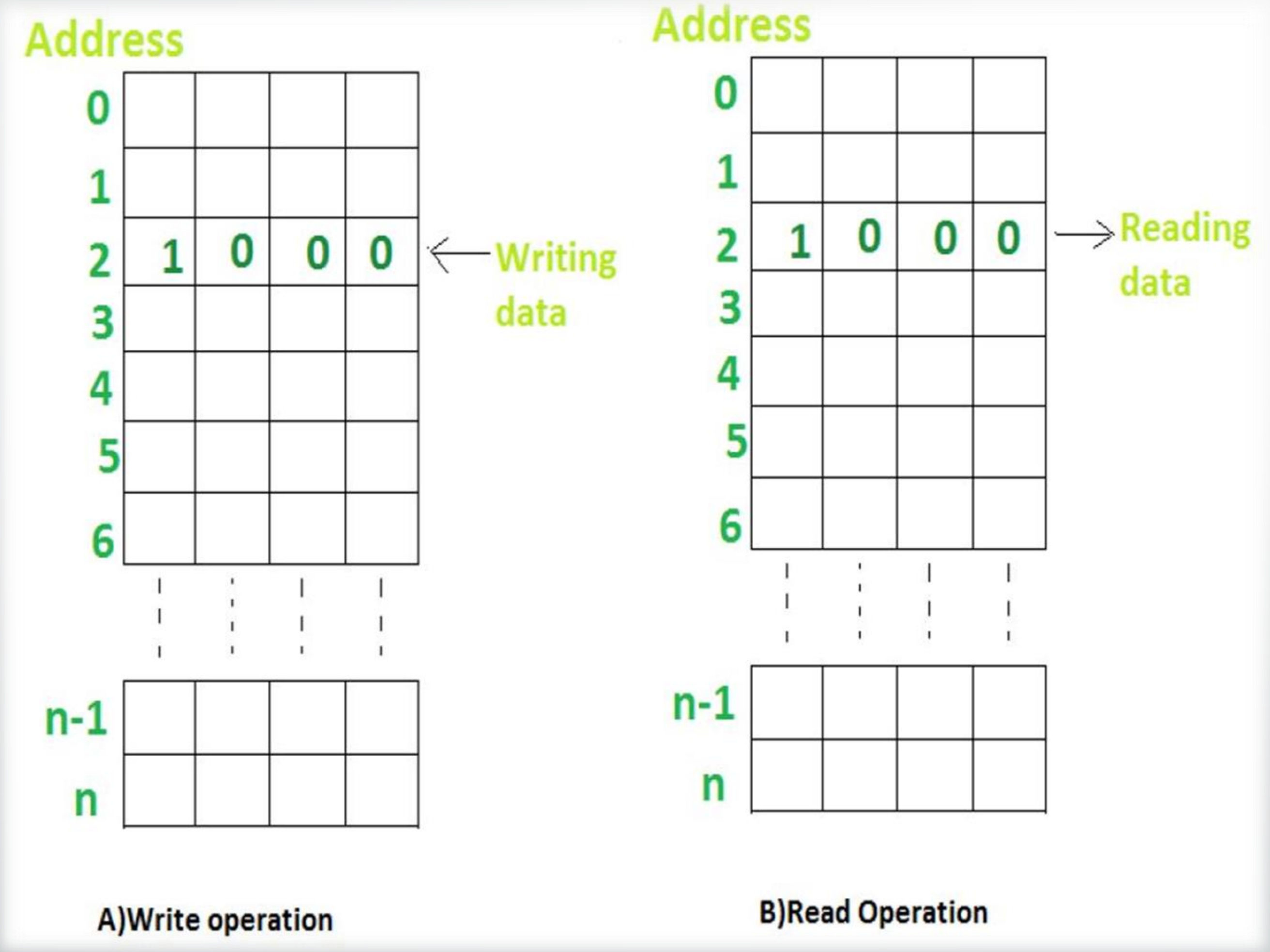
Memory addressing and Memory organization

A memory address is a unique identifier used by a device or CPU for data tracking. This binary address is defined by an ordered and finite sequence allowing the CPU to track the location of each memory byte.

Memories are made up of registers. Each register in the memory is one storage location. The storage location is also called a memory location. Memory locations are identified using **Address**. The total number of bit a memory can store is its **capacity**.

A storage element is called a **Cell**. Each register is made up of a storage element in which **one bit** of data is stored. The data in a memory are ***stored*** and ***retrieved*** by the process called ***writing*** and ***reading*** respectively.

Memory organization



A **word** is a group of **bits** where a memory unit stores binary information.

A word with group of 8 bits is called a **byte**.

Memory Locations and Addresses

A memory unit consists of **data lines, address selection lines, and control lines** that specify the direction of transfer.

Data lines provide the information to be stored in memory. The control inputs specify the direction transfer. The **k-address lines** specify the word chosen. When there are **k** address lines, $\mathbf{2^k}$ memory word can be accessed.

For example, a memory with **64k words and a word size of 1 byte**, then this memory unit has 64 * 1024 = **65536** memory locations. The **address** of these locations varies **from 0 to 65535**.

Unit	No. of Bytes
Kilobyte (KB)	2^{10} (1024) bytes
Megabyte (MB)	2^{20} (1048576) bytes
Gegabyte (GB)	2^{30} (1073741824)
Terabyte (TB)	2^{40} bytes

Memory Locations and Addresses (Cont.)

Example-1: A computer has **32 MB** of memory. How many bits are needed to address any single byte in memory?

Solution:

The memory address space is 32 MB, which means $2^5 \times 2^{20} = 2^{25}$. This means that we need **25 bits**, to address each byte.

$$2^5 * 2^{10} * 2^{10} = 2^{25}$$

32 * 1024 *1024 = 33.554.432 bytes

So you will need, at least 25 bits to address a single byte in that memory.

Example-2: A computer has 128 MB of memory. Each word in this computer is eight bytes. How many bits are needed to address any single word in memory?

Solution:

The memory address space is 128 MB, which means $\mathbf{2^7 \times 2^{20} = 2^{27}}$. However, each word is eight (2^3) bytes, which means that we have 2^{24} words. This means that we need $\mathbf{\log_2 2^{24}}$, or **24 bits**, to address each word.

So you will need, at least 24 bits to address a single byte in that memory.

Main Memory Organization

The number of bits stored in a register is called **a memory word**. Memory devices (chips) are available in variable word sizes. Each word in memory has a specific binary number called **memory address**.

- To communicate with the main memory, the CPU should be able to:
  - ✓ Select the chip.
  - ✓ Identify the memory location, and
  - ✓ Identify the memory operation (Read from / Write into) the memory location.

To do its job, any memory chip must have the following terminals:

a) Address lines: to identify number of memory locations in the chip according to the following relation:

n-address line ⇒ 2^n *memoey locations*

Ex: if n=9 then the chip contains $2^9 = 512$ locations.

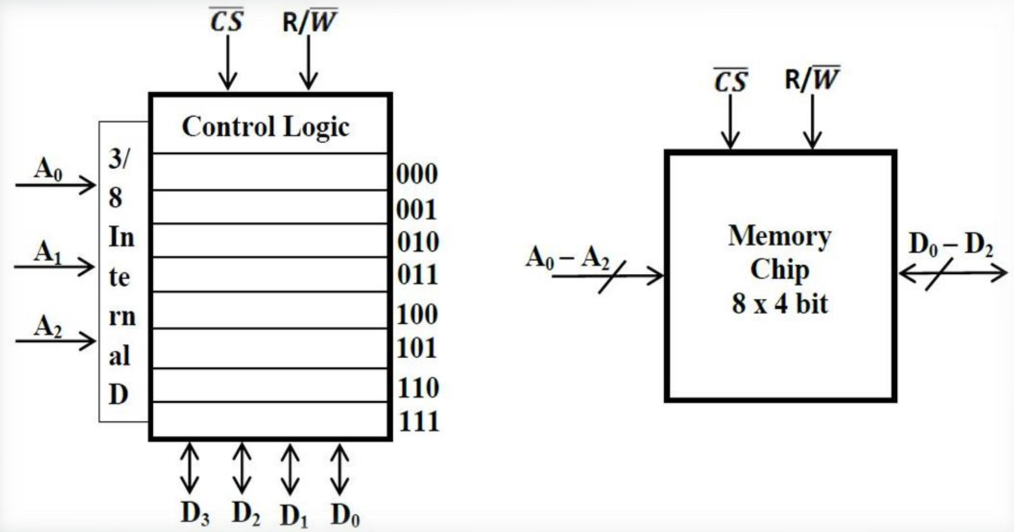
b) Data lines: represent data input / output lines in a memory chip.

Ex: 2M x 8 of memory has 8-data lines and 21 address lines since.

$$\mathbf{2m = 2x2^{20} = 2^{21}}$$

c) Control lines: memory chips are provided two control lines, R/$\overline{W}$ and $\overline{CS}$. The R/$\overline{W}$ line is used to specify the required operation about read or write. The $\overline{CS}$ (Chip Select) line is required to select a given chip in a multi- chip memory system.

The following figure shows hypothetical memory chip of eight registers (locations), 3 address lines, one chip select $\overline{CS}$, one $R/\overline{W}$ line and 4 data lines.

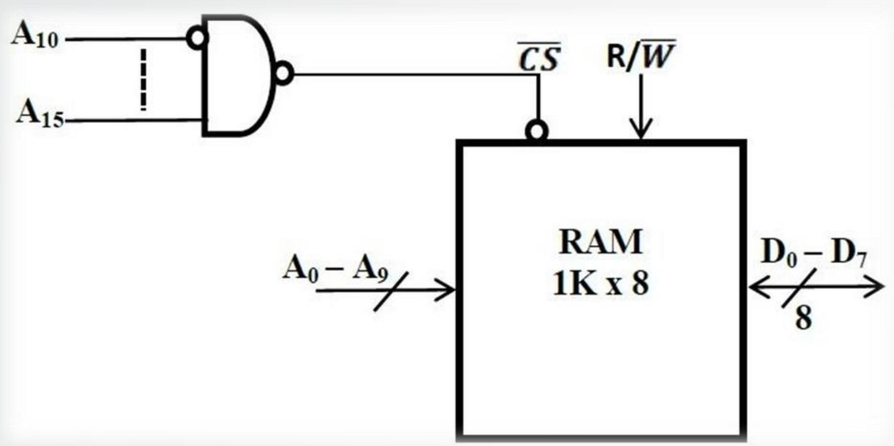


Note: The address bus lines are split into two sections: N **most significant bits** for CS and M **least significant bits** addressing memory locations.

Ex: How many memory locations can be addressed by a CPU with 16 address lines?

Solution: $2^{16} = 2^{6} \times 2^{10} = 64K$.

Ex: Give the effective address for the following memory chip?

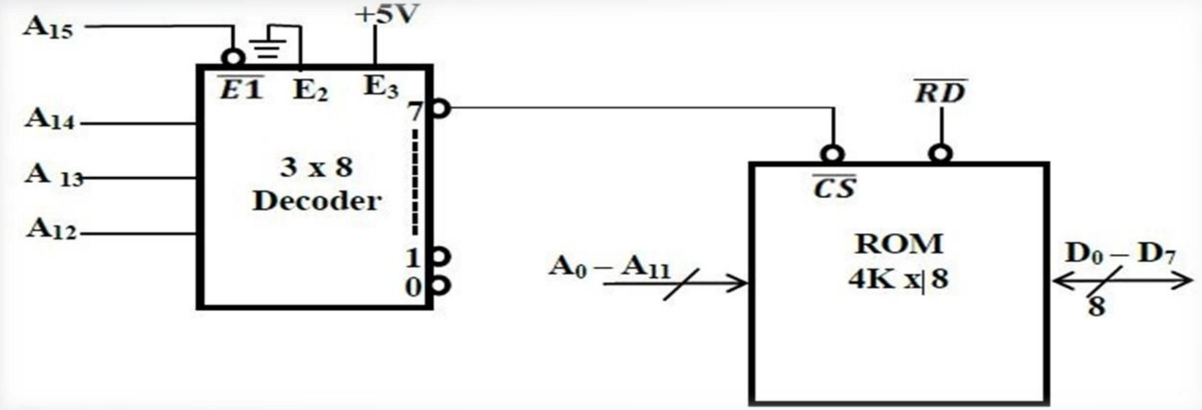


Solution:

A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	
1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	= $F800_h$
1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	= $FBFF_h$

Chip select (A_{15} – A_{10})   Location select (A_9 – A_0)

Ex: Give the effective address for the following memory chip?



Solution:

A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	
0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	= 7000_h Start
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	= $7FFF_h$ End

Chip select (A_{15}–A_{12})   Location select (A_{11}–A_0)

Memory Expansion

Main memory can be expanded either in word size (horizontal) or by increasing number of words (vertical).

a) Horizontal Expansion (increase number of data)

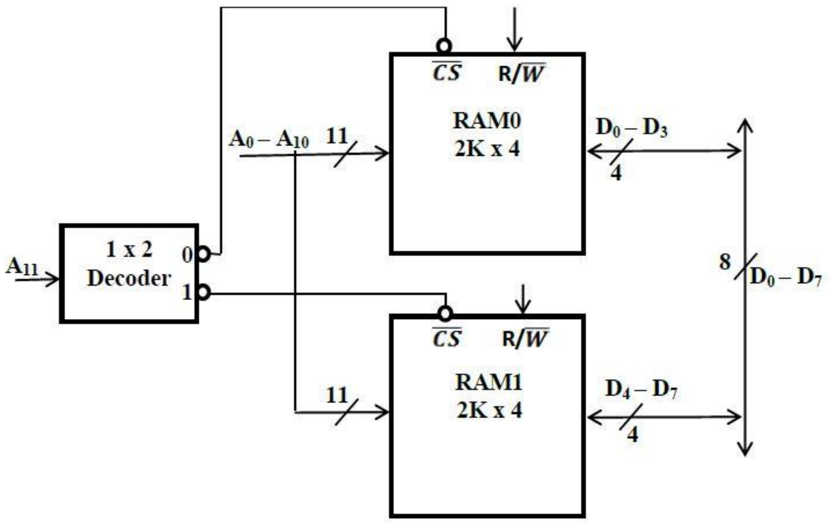
Ex: Implement (Design) a 2k bytes memory using 2k x 4 bit chip RAM and 8 bit data lines.

Solution:

$$No.of\ chips = \frac{Total\ size}{size\ of\ a\ chip} = \frac{2K \times 8}{2K \times 4} = 2\ chips$$

$no.of\ address\ lines\ for\ each\ chip: 2K = 2 \times 2^{10} = 2^{11}$

$\Rightarrow 11\ address\ lines\ common\ between\ the\ chips\ (A_0 - A_{10})$



A_{11}	A_{10}	A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0		
0	0	0	0	0	0	0	0	0	0	0	0	Start	RAM0
0	1	1	1	1	1	1	1	1	1	1	1	End	
1	0	0	0	0	0	0	0	0	0	0	0	Start	RAM1
1	1	1	1	1	1	1	1	1	1	1	1	End	

b)**Vertical Expansion** (increase number of address line)

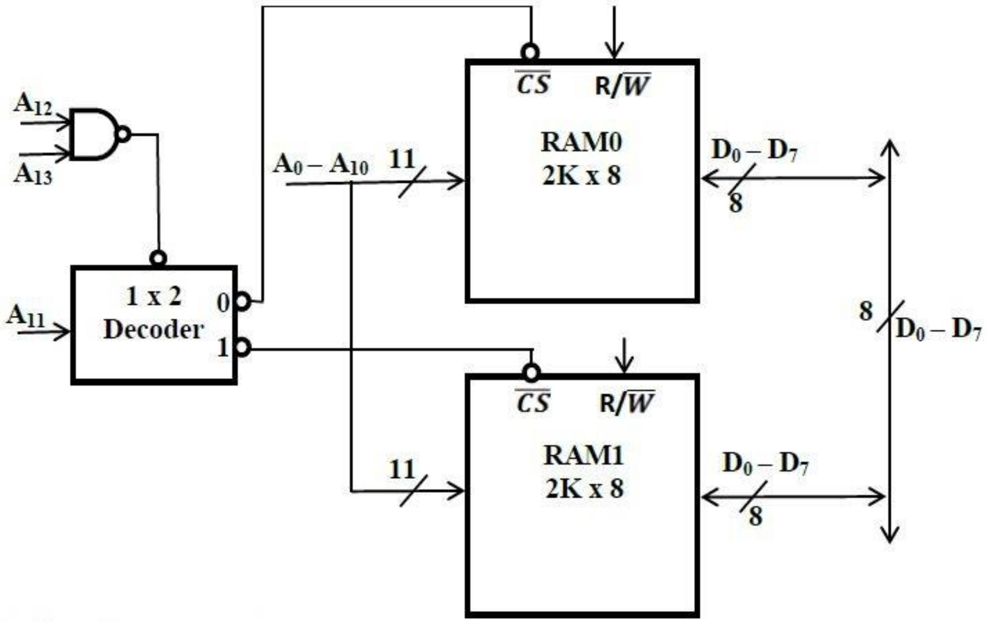
Ex: Implement a memory with 4KB x8 using 2KB x8 RAM chips, if the CPU has 14 address lines?

Solution:

$$No.of\ chips = \frac{Total\ size}{size\ of\ a\ chip} = \frac{4K \times 8}{2K \times 8} = 2\ chips$$

$$no.of\ address\ lines\ for\ each\ chip: 2K = 2 \times 2^{10} = 2^{11}$$

$$\Rightarrow 11\ address\ lines\ common\ between\ the\ chips\ (A_0 - A_{10})$$



Computer Organization

Lec Eight: CPU Organization: Control Unit (CU)

Control unit

- The function of control unit is to generate timing and control signals to all operations in the computer.

- Control Unit is “the brain within the brain“.

- It controls the flow of data between the processor and memory and peripherals.

- The control unit must communicate with both the arithmetic logic unit (ALU) and main memory.

- The control unit instructs the arithmetic logic unit that which logical or arithmetic operation is to be performed.

- The control unit coordinates the activities of the arithmetic/logic units as well as all peripherals and auxiliary storage devices linked to the computer.

Control unit Organization

- The basic task of the control unit for each instruction the control unit causes the CPU to go through a sequence of control steps in each control step the control unit issues a set of signals.

- The control unit is driven by the processor clock

- The signals to be generated at a certain moment depend on:

  - The actual step to be executed
  - The condition and status flags of the processor
  - The actual instruction executed
  - External signals received on the system bus (e g interrupt signal)

Control unit Organization

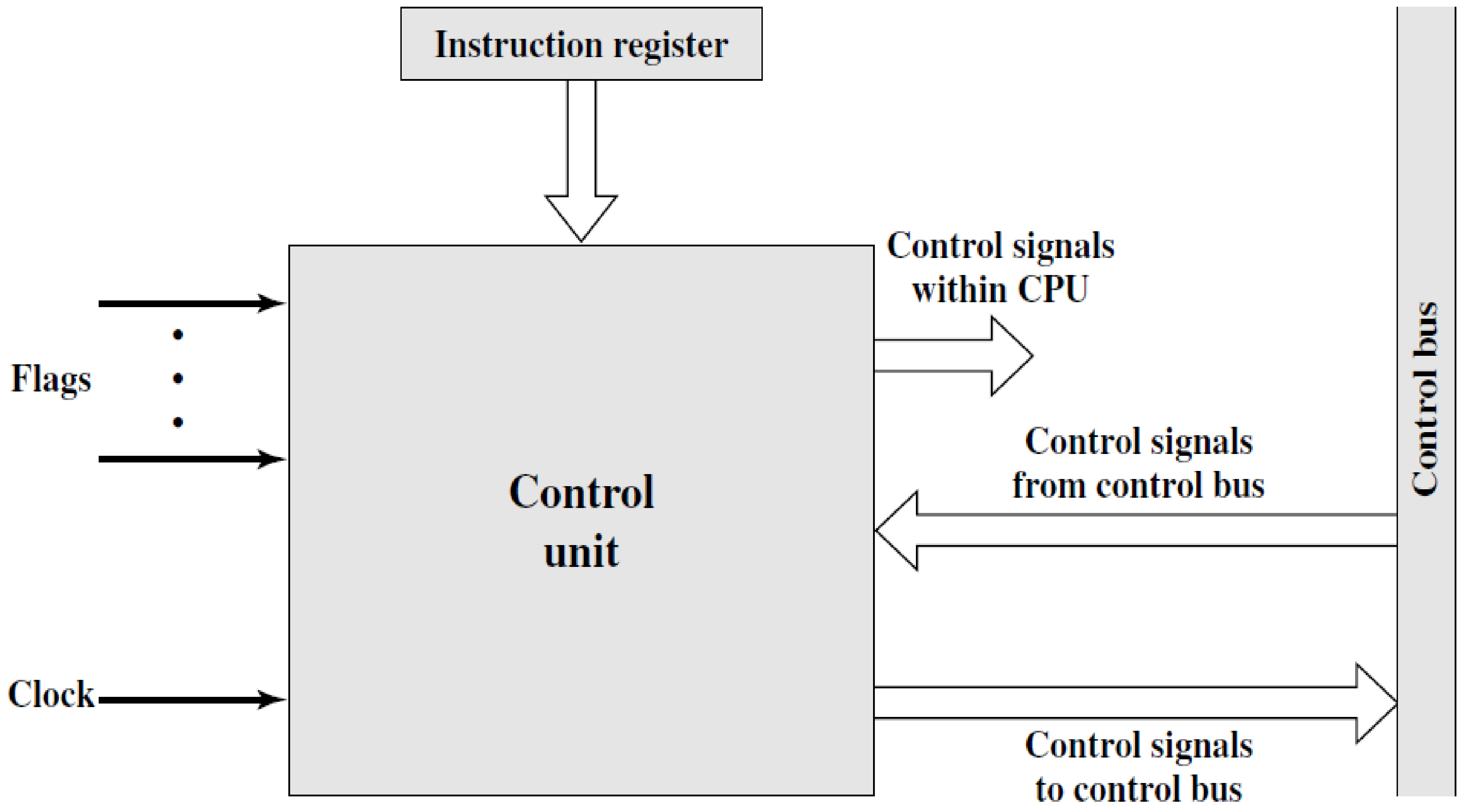


Figure Block Diagram of the Control Unit

Control Signal Sources

- Clock
  It helps to synchronize the operation. It causes one micro-operation to be performed for each clock pulse
- Instruction Register
  - Op-code for the current instruction
  - Determines which micro-instructions are performed
- Flags
  - State of CPU
  - Results of previous operations
- From Control Bus
  - Interrupts / Bus Requests
  - Acknowledgements

Control Signal Outputs

- Within Processor
  - Cause data movement
  - Activate specific functions

- Via Main Bus
  - To memory
  - To I/O modules

Types

- There are two design approach for CU:
  - Hardwired approach
  - Micro –programming approach

Hardwired Approach

- The control signals are generated with the help of the hardware.
- Hardwired control units are implemented through the use of sequential logic units or circuits like gates flip-flops decoders multiplexers and other logic buildings blocks.
- Hardwired control units are generally faster than micro programmed designs.
- This architecture is preferred in reduced instruction set computers (as they use a simpler instruction set.

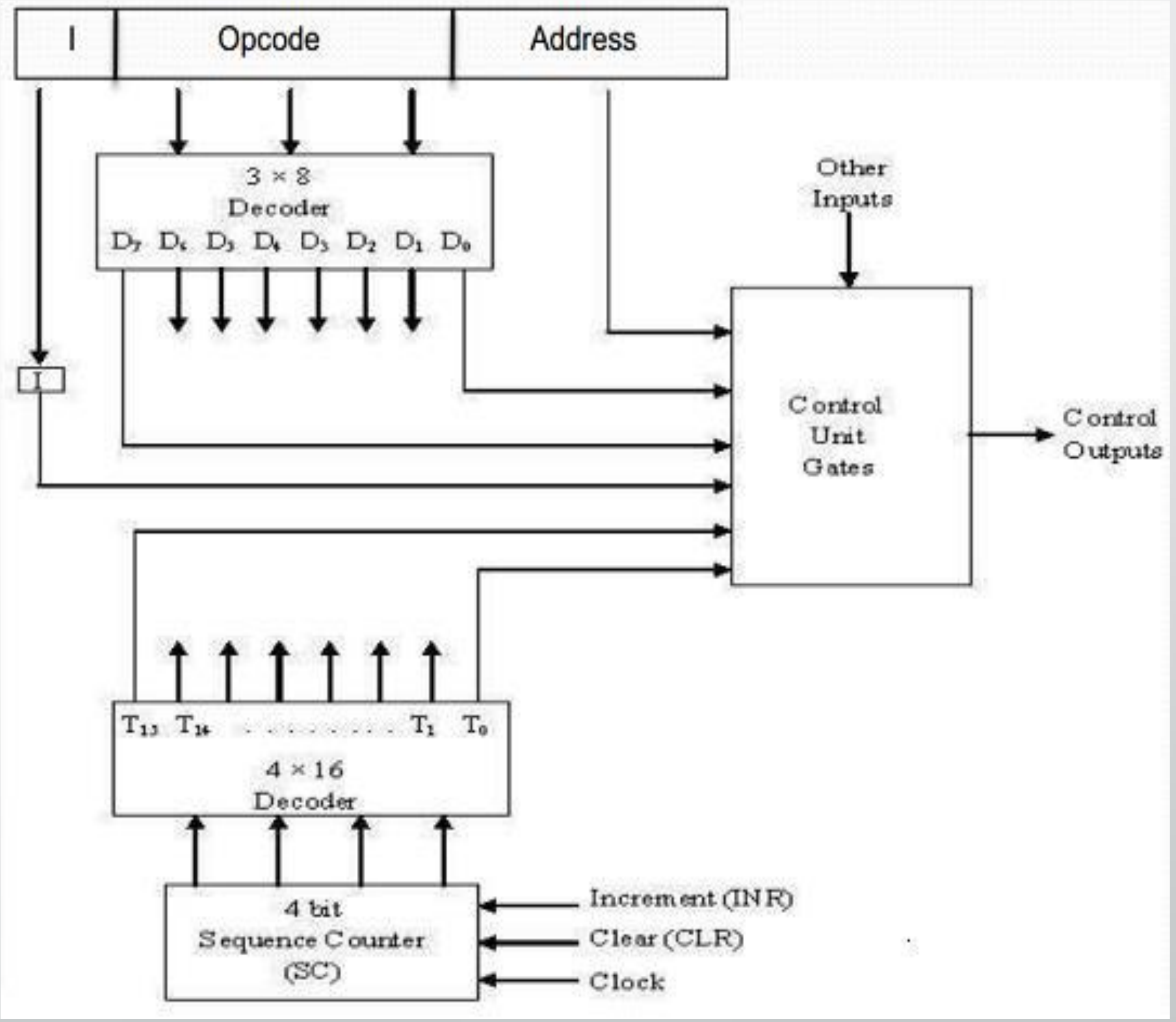
Advantages and Disadvantage Hardwired Approach

Advantages	Disadvantage
1. Hardwired Control Unit is fast because control signals are generated by combinational circuits.	1. The control signals required by the CPU will be more complex.
2. The delay in the generation of control signals depends upon the number of gates.	2. Modifications in control signal are very difficult. That means it requires rearranging of wires in the hardware circuit.
3. The performances is high as compared to micro –programmed control unit.	3. It is difficult to correct mistake in original design or adding new features in existing design of control unit.

Hardwired Architecture

Control unit consist of :

- Instruction Register
- Number of Control Logic Gates,
- Two Decoders
- 4 –bit Sequence Counter



Micro programmed Approach

- Micro programs were organized as a sequence of microinstructions and stored in the special control memory.
- A microprogrammed control unit is implemented using a programming approach. A sequence of micro-operations is carried out by executing a program consisting of micro instructions.
- The microprogram, consisting of microinstructions is stored in the control memory of the control unit.
- Execution of micro instruction is responsible for the generation of a set of control signals.

Definitions

- ✓ **Micro–Programs:** Microprogramming is the concept for generating control signals using programs. These programs are called micro–programs.

- ✓ **Micro–Instructions:** The instructions that make micro –program are called micro-instructions.

- ✓ **Micro –Code:** Micro –program is a group of microinstructions. The micro –program can also be termed as micro –code.

- ✓ **Control Memory:** Micro–programs are stored in the read only memory (ROM). That memory is called control memory.

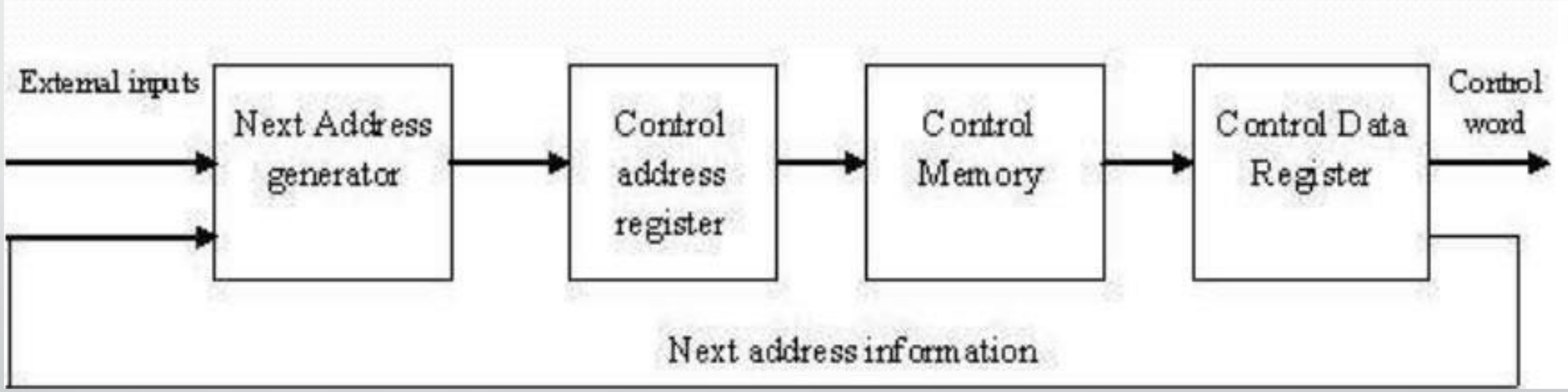
Advantages and Disadvantage Micro programmed Approach

Advantages	Disadvantage
1. The design of micro program control unit is less complex because micro programs are implemented using software routines	1. The micro program control unit is slower than hardwired control unit That means to execute an instruction in micro program control unit requires more time
2. The micro programmed control unit is more flexible because design modifications, correction and enhancement is easily possible	2. The design duration of micro program control unit is more than hardwired control unit for smaller CPU
3. The fault can be easily diagnosed in the micro program control unit using diagnostics tools by maintaining the contents of flags, registers and counters	

Micro programmed Architecture

Control unit consist of:

- ❖ Next address generator
- ❖ Control address register
- ❖ Control memory
- ❖ Control data register



Comparison

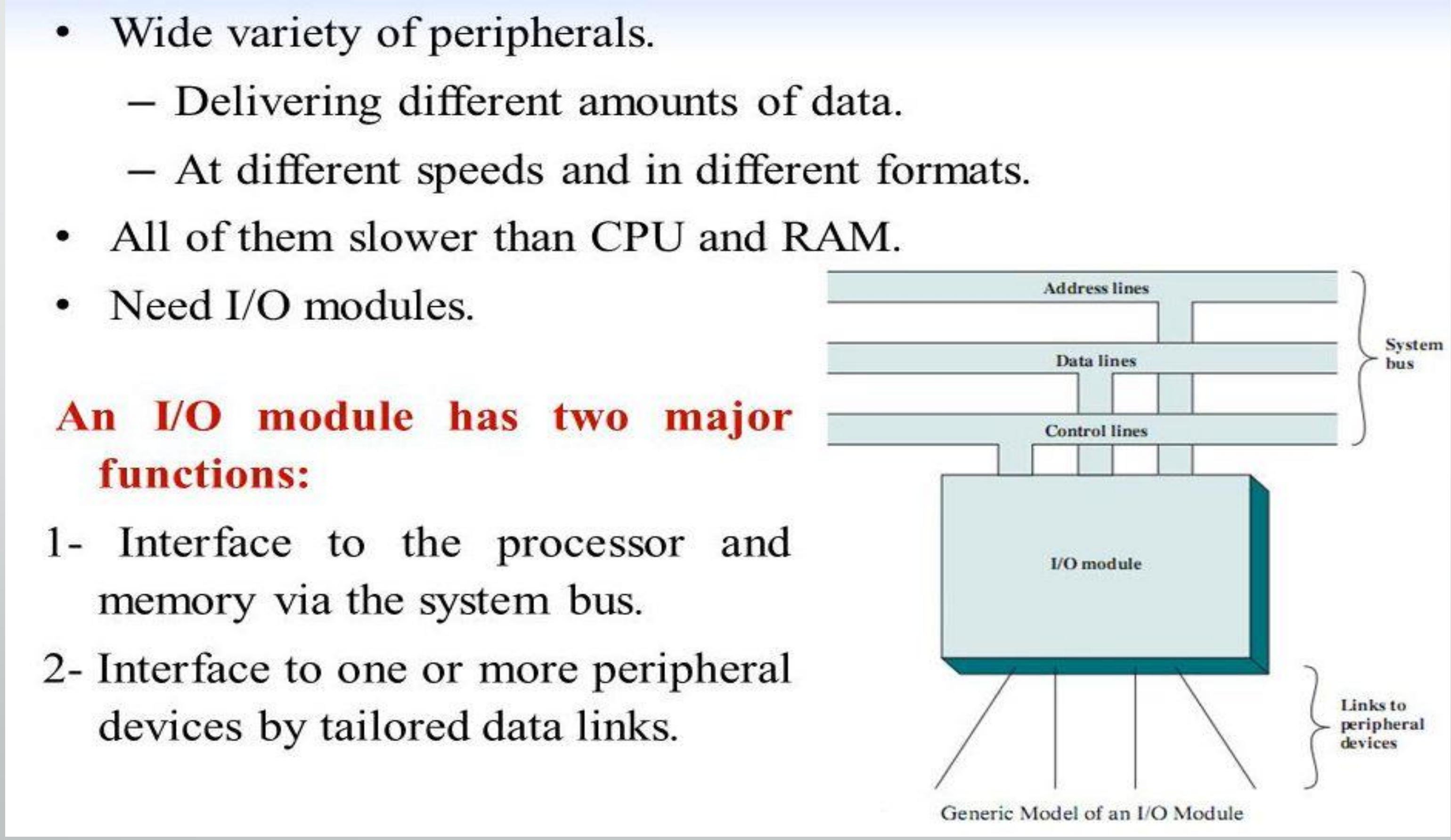
Attributes	Hardwired Control	Microprogramming Control
Speed	Fast	Slow
Cost of Implementation	More	Cheaper
Flexibility	Difficult to modify	Flexible
Ability to handle complex instruction	Difficult	Easier
Decoding	Complex	Easy
Instruction Set Size	Small	Large
Control Memory	Absent	Present

Computer Organization

Lec Nine: Input/Output Organization

Input/ Output Module

- The I/O organization of computer depends upon the size of computer and the peripherals connected to it.
- I/O module is an interface for the external devices (peripherals) to CPU and Memory.



Peripheral Devices

External devices are classified into three categories:

- Human readable: Suitable for communicating with the computer user.
  - Screen, printer, keyboard.
- Machine readable: Suitable for communicating with equipment.
  - Magnetic disk and tape systems, sensors and actuators.
- Communication: Suitable for communicating with remote devices.
  - Modem.
  - Network Interface Card (NIC).

I/O Module Function

- Control & Timing.
- Device Communication .
- Error Detection.
- Processor Communication.
- Data Buffering.

Example: Control and timing of the transfer of data from an external device to the processor steps:

- The processor checks I/O module device status.
- I/O module returns status.
- If ready, CPU requests data transfer using command to module.
- I/O module gets data from external device.
- I/O module transfers data to the processor.

I/O Module Function (Cont.)

Processor communication involves the following:

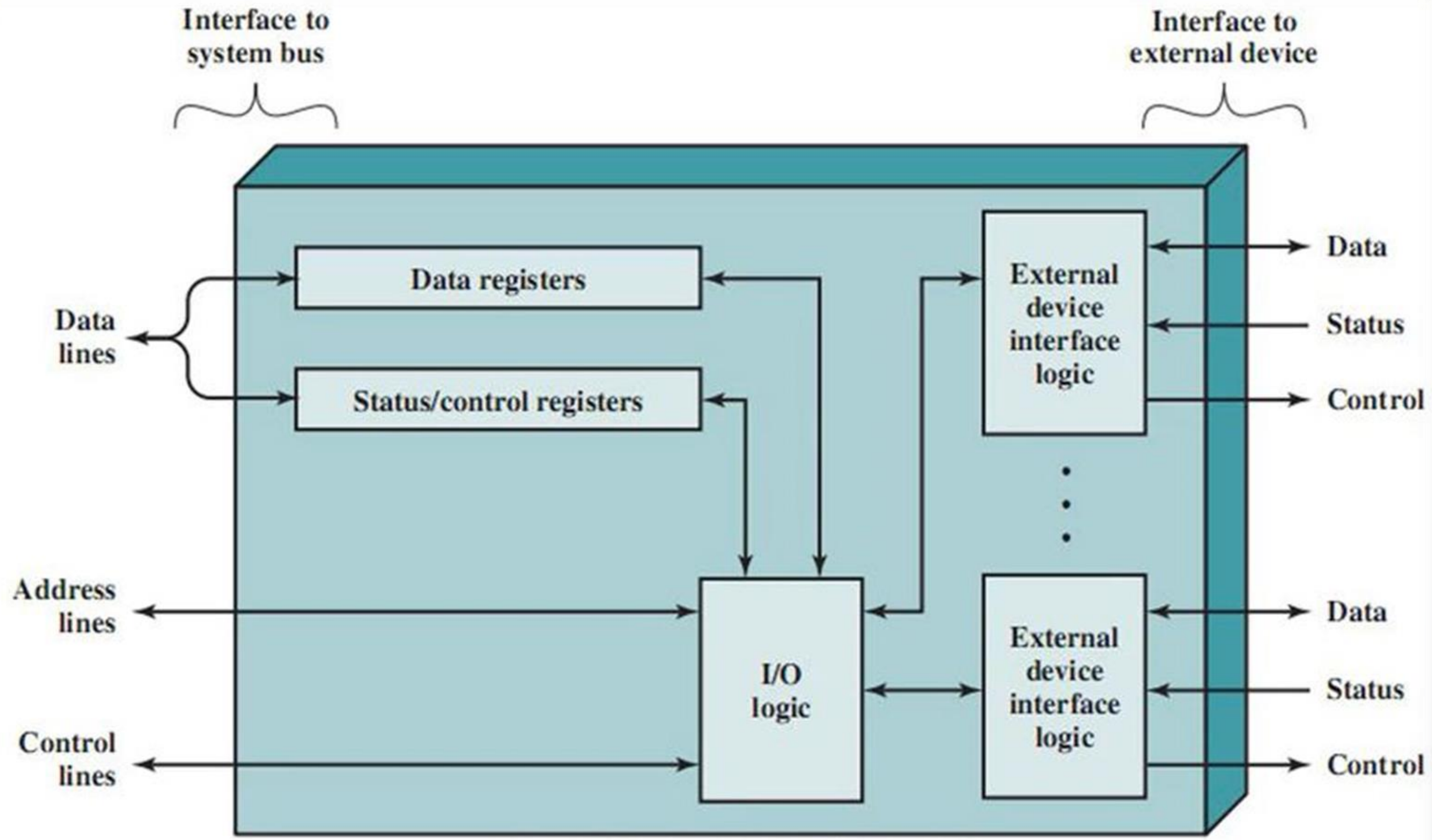
- Command decoding (READ SECTOR, SEEK track number).
- Data: exchange data between CPU & I/O module over data bus.
- Status reporting.
- Address recognition: I/O module must recognize unique address.

Device communication (commands, status information, data).

Data buffering: The transfer rate into and out of main memory or the processor is quite high.

Error detection (paper jam, bad disk track).

I/O Module Structure



Block Diagram of an I/O Module

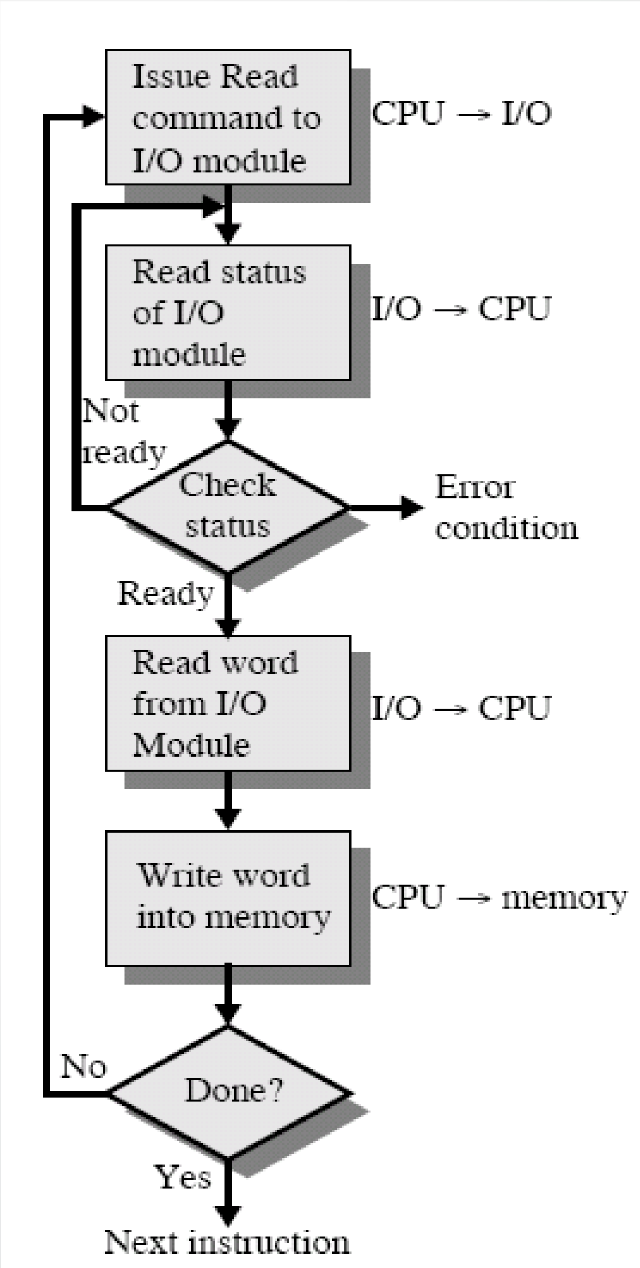
Input Output Techniques

- Data transfer between computer and I/O device can be handled in the following ways:

  - **Programmed I/O**
  - **Interrupt driven I/O**
  - **Direct Memory Access (DMA)**

1- Programmed I/O

- CPU has direct control over I/O
  - Sensing status
  - Read/write commands
  - Transferring data
- CPU waits for I/O module to complete operation
- Programmed I/O **flowchart**:



I/O Commands

- **Control:** Used to activate a peripheral and tell it what to do.
- **Test:** Used to test various status conditions associated with an I/O module and its peripherals.
- **Read:** Causes the I/O module to obtain an item of data from the peripheral and place it in an internal buffer.
- **Write:** Causes the I/O module to take an item of data (byte or word) from the data bus and transmit it to the peripheral.

I/O Addressing and Mapping

- Each device given unique identifier (Address).
- CPU commands contain identifier.

- **Memory mapped I/O.**
  - Devices and memory share an address space.
  - I/O looks just like memory read/write.
  - No special commands for I/O.

    Large selection of memory access commands available.
- **Isolated I/O**
  - Separate address spaces.
  - Need I/O or memory select lines.
  - Special commands for I/O.

    Limited set.

2- Interrupt Driven I/O

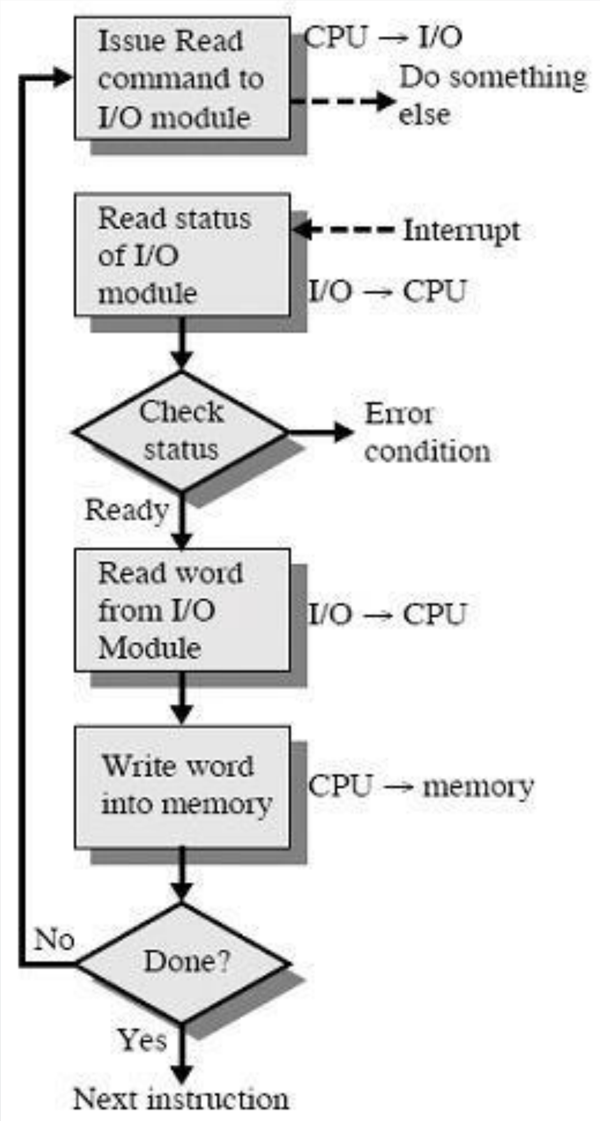
- Overcomes CPU waiting.
- No repeated CPU checking of device.
- I/O module interrupts when ready.

Interrupt Processing

1) The device issues an interrupt signal to the processor.
2) CPU finishes execution of the current instruction before the interrupt.
3) Acknowledgment signal sent by CPU which allows the device to remove its interrupt signal.
4) The processor prepare to transfer control to the interrupt routine. it save needed information (Status of the processor and the location of the next instruction to be executed).
5) When interrupt processing is complete, the saved register values are retrieved from the stack and restored to the registers.
6) Restore the next instruction address to be executed.

Interrupt Driven Basic Operation

- CPU issues read command
- I/O module gets data from peripheral while CPU does other work
- I/O module interrupts CPU
- CPU requests data
- I/O module transfers data
- Interrupt Driven I/O **Flowchart**:



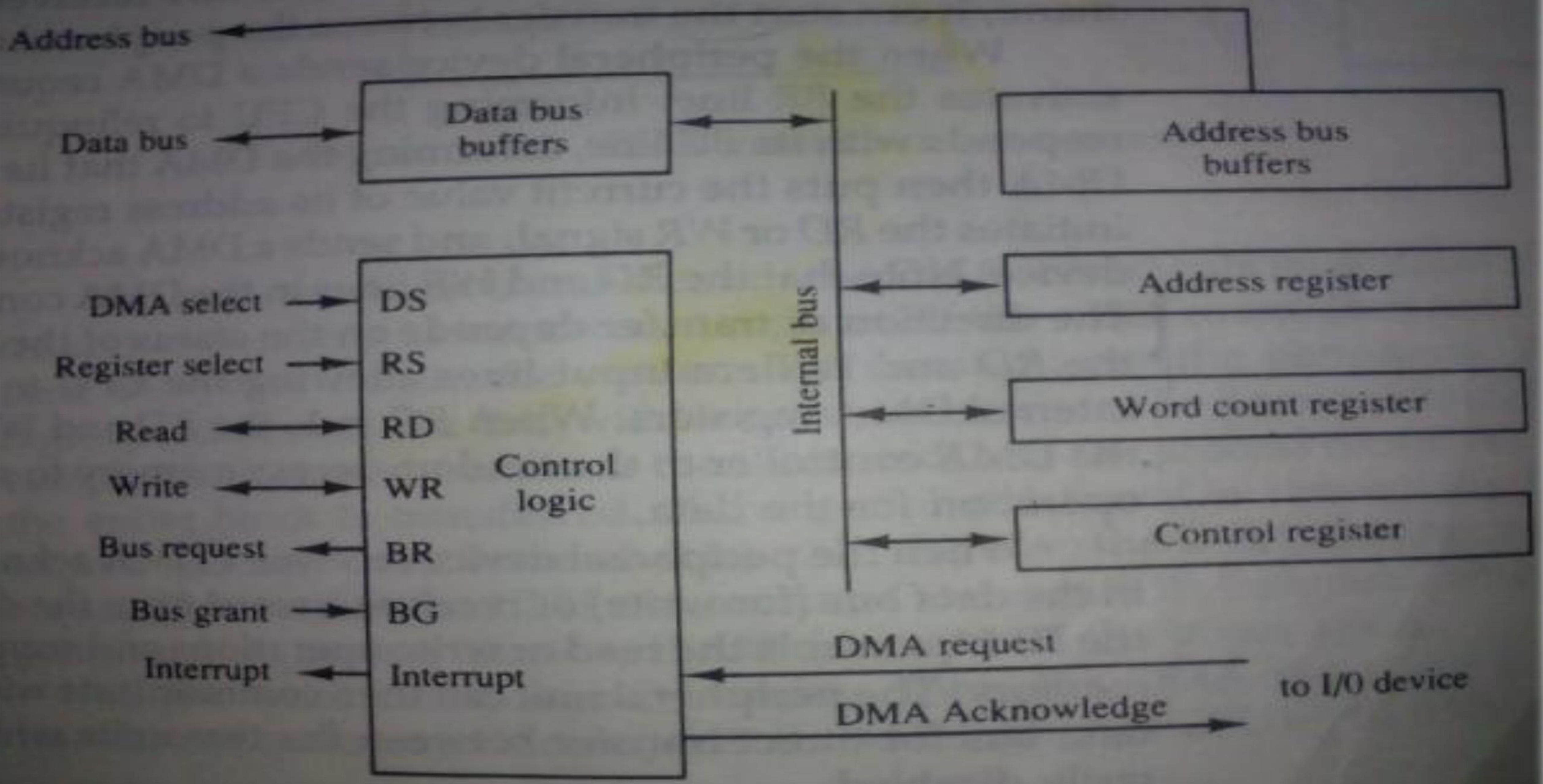
3– Direct Memory Access (DMA)

The DMA controller is a piece of hardware that controls one or more peripheral devices. It allows devices to transfer data to or from the system's memory without the help of the processor.

- **A DMA controller has :**

  - ✓ An address register: an address that specifies the memory location of the data to be transferred.
  - ✓ A word count register: is holds the number of words to be transferred and it is decremented by one after each word transfer.
  - ✓ A control register: is specifies the transfer mode.

Figure 11-17 Block diagram of DMA controller.



Computer Organization

Lec Ten: Computer Software

Computer Software

- Computer hardware is useless without software.
- Software is the set of instructions and associated data that direct the computer to do a task.
- Software can be divided into two categories: system software and application software.
- System software: helps the computer to carry out its basic operating tasks.
- Application software: helps the user carry out a variety of tasks.

Types of Software

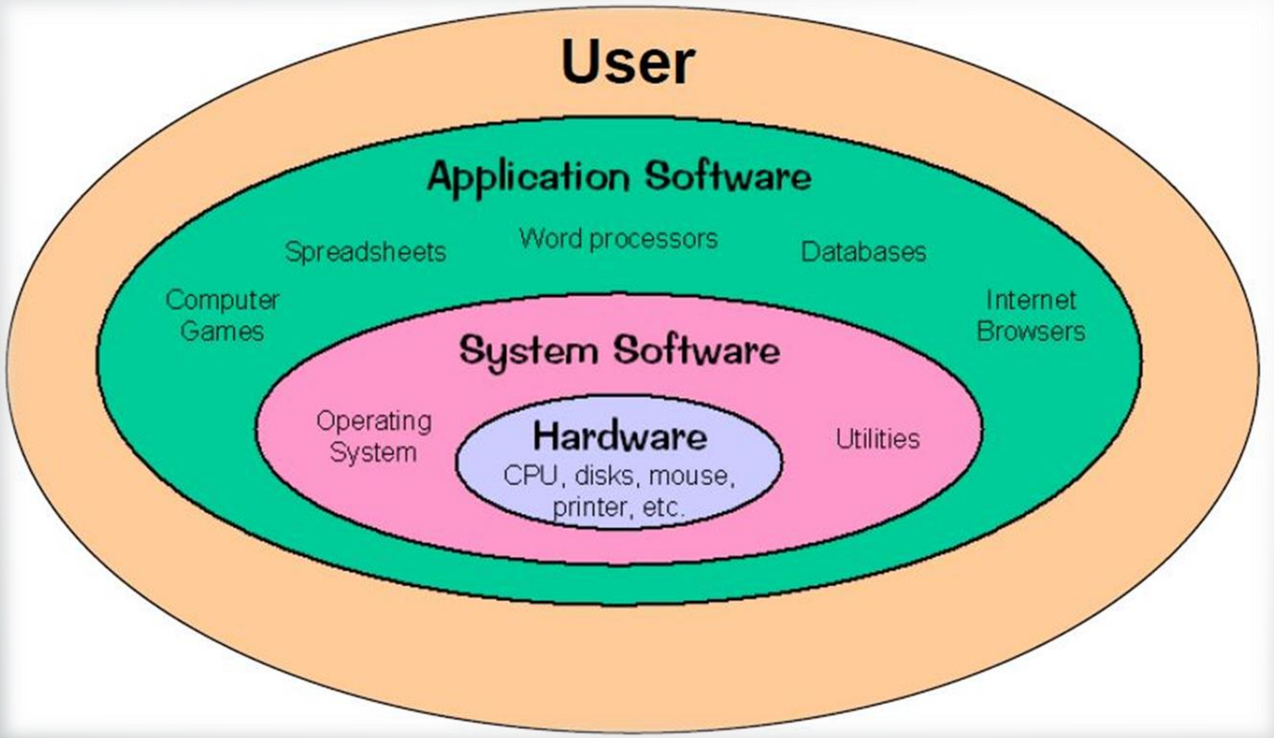
- System Software

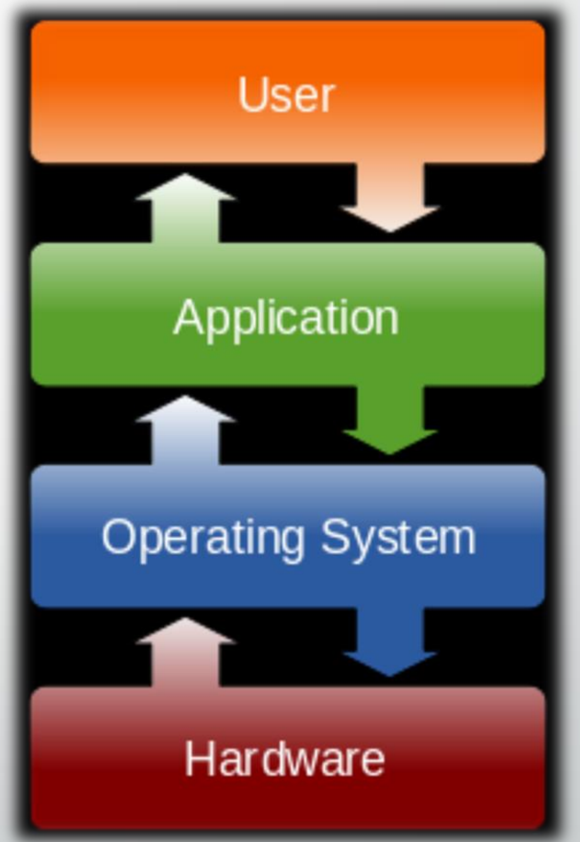
  The programs are directly related to the computer hardware and perform tasks associated with controlling and utilizing computer hardware.

- Application Software

  An application is a job or task a user wants to accomplish through a computer, application software are programs that help a user perform a specific job.

Types of Software





System Software

- **Manages the fundamental operations of the computer, such as:**
  - Loading programs and data into memory, executing programs, saving data to disks, displaying information on the monitor, and transmitting data through a port to a peripheral device.
  - System software is responsible for managing a variety of independent hardware components so that they can work together in an arranger way.
  - System software: operating systems, utilities, device drivers.

Operating System

- Collection of computer programs that control the interaction of the user and the computer hardware.

- A part of the operating system code is stored in a ROM and the rest of it resides on a disk.

Responsibilities of an Operating System

- Communicate with a user, receive and execute commands, and show error messages.
- Manage allocation of memory, processor time, and other resources.
- Collect input from keyboard, mouse, and provide data to running programs.
- Convey program output to screen, printer, or other output devices.
- Access data from secondary storage.
- Write data to secondary storage.
- Maintains security (checks user name, password, virus infection).

Application software

- Opera (Web Browser)
- Microsoft Office:

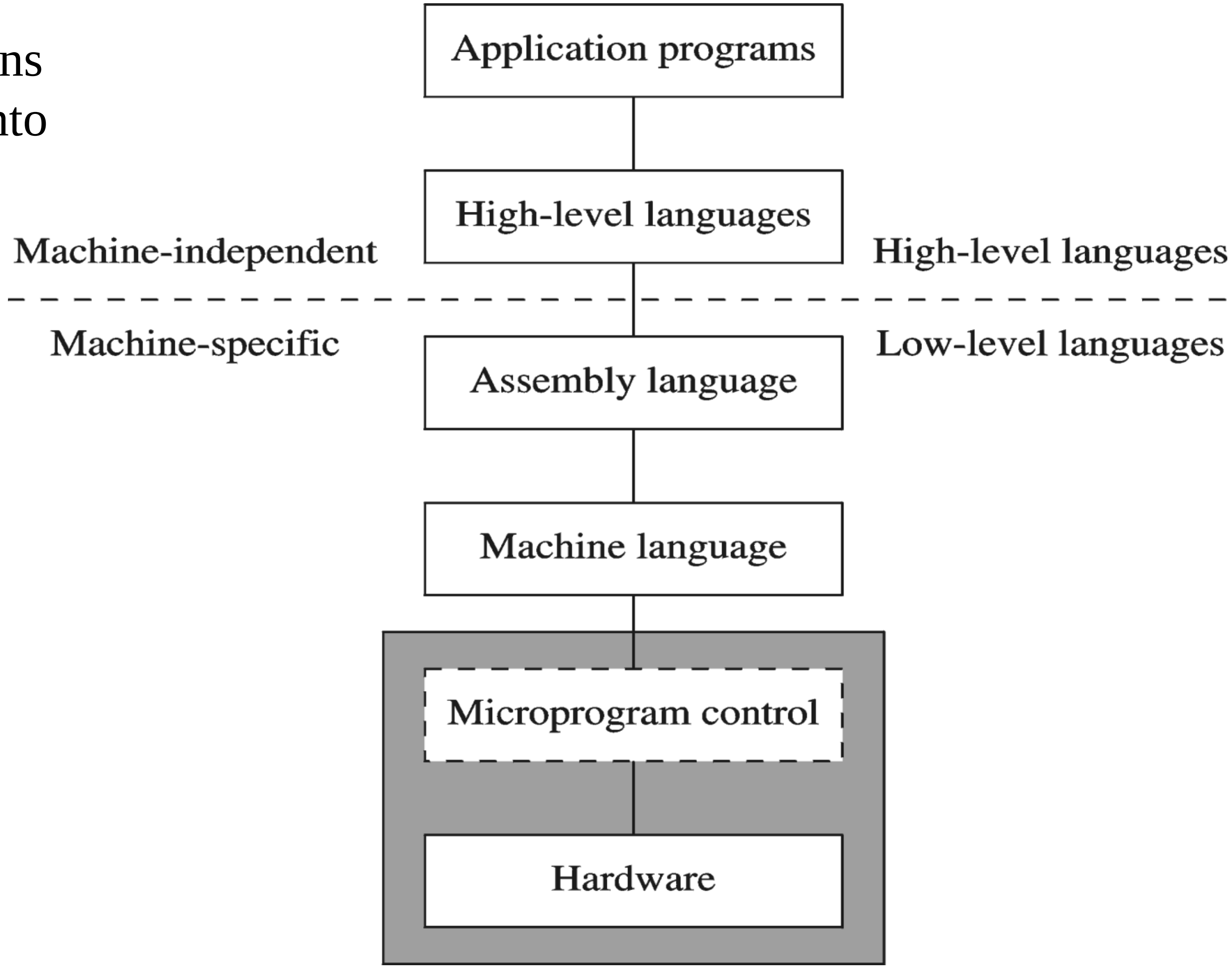
  I. Microsoft Word (Word Processing).
  II. Microsoft Excel (Spreadsheet software).
  III. Microsoft PowerPoint (Presentation Software).
- iTunes (Music / Sound Software).
- VLC Media Player (Audio / Video Software).
- Game Software.
- Adobe Photoshop (Graphics Software).

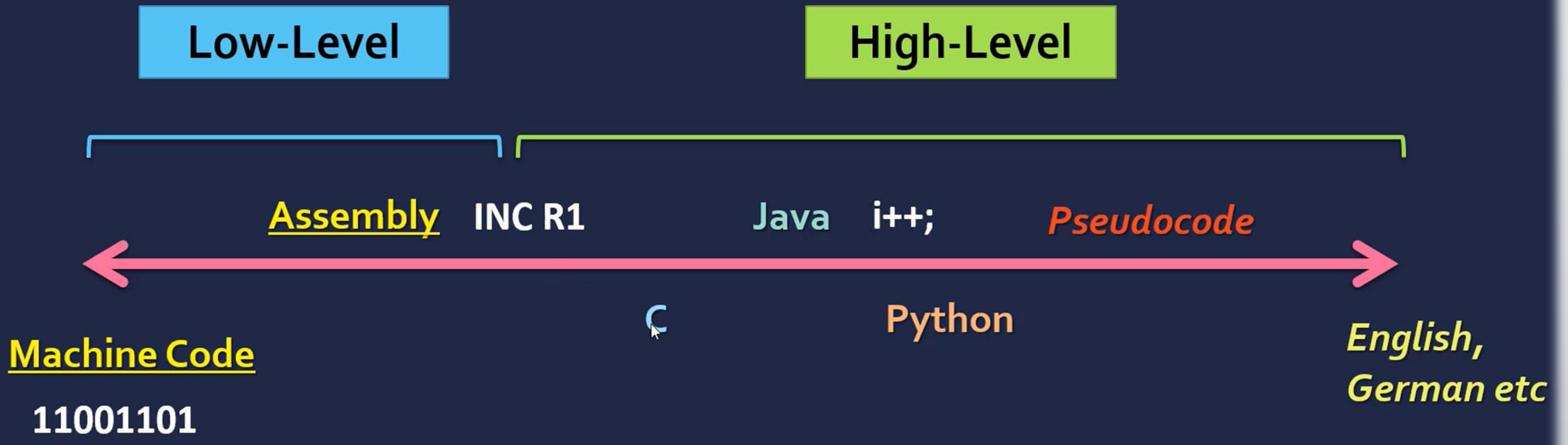
A Hierarchy of Languages

Programming language:

Allow the programmer to write instructions that direct the computer to process data into information.

1. High-Level Language
2. Assembly Language
3. Machine Language





Low-Level
High-Level
Assembly
INC R1
Java
i++;
Pseudocode
C
Python
Machine Code
11001101
English,
German etc

Machine Language

- Machine language
  - Machine Language is the only language understood by computers (i.e. its native to processors).
  - Executed directly by hardware.
  - There is nothing “below” machine language-only hardware.
  - Impossible for humans to read. Consists of only 0’s and 1’s: 0001001111110000.

Assembly language

- ✓ A programming language that uses symbolic names to represent operations, registers, and memory locations.
- ✓ Slightly higher-level language.
- ✓ The readability of instructions is better than machine language.
- ✓ One-to-one correspondence with machine language instructions.

High Level Language (HLL)

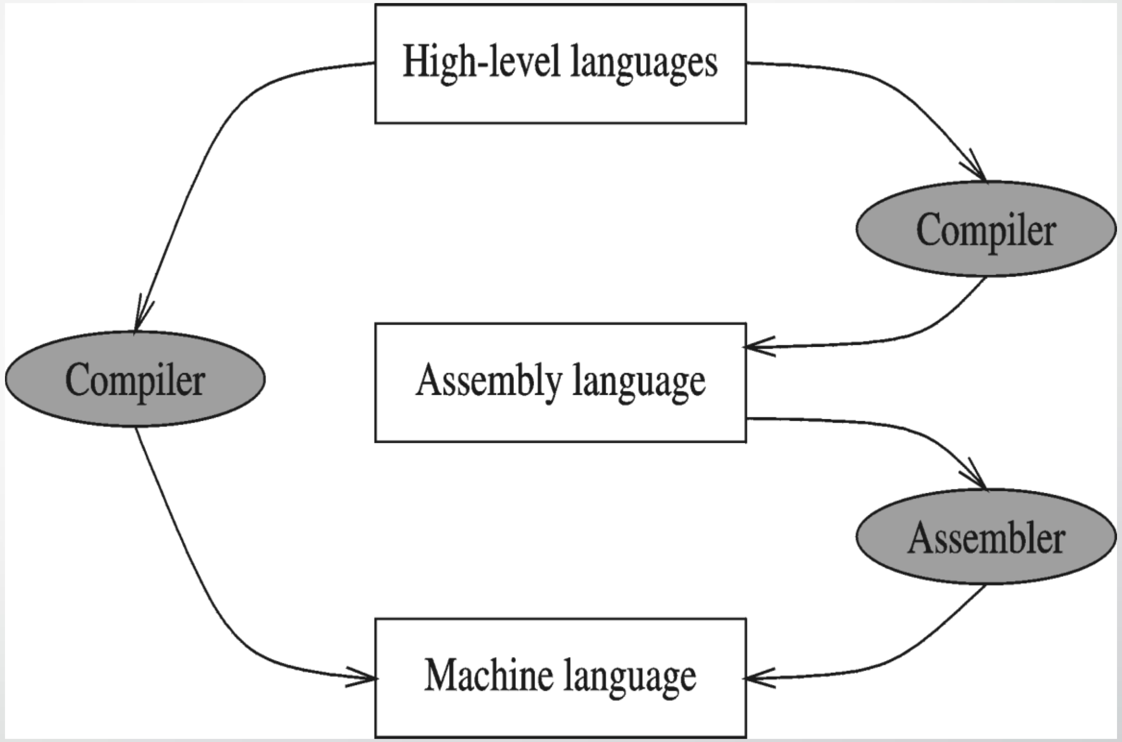
- It is a computer programming language that isn't limited by the computer, designed for a specific job, and it is easier to understand, However, for a computer to understand and run a program is created with a high-level language, it must be compiled into machine language.

- Such as BASIC, COBOL, PASCAL, C, C++, JAVA, SCHEME, Lisp, ADA, etc.

Language Translators

- **Assemblers:** translate assembly to machine code.

- **Compilers:** translate high-level programs to machine code.



Instructions and Machine Language

- Each command of a program is called an **instruction** (it instructs the computer what to do).

- Computers only deal with binary data, hence the instructions must be in binary format (0's and 1's).

- The set of all instructions (in binary form) makes up the computer's **machine language**. This is also referred to as the **instruction set**.

Instructions Fields

- Machine language instructions usually are made up of several fields. Each field specifies different information for the computer. **The major two fields are:**

- **Opcode** field stands for operation code and it specifies the particular operation that is to be performed, each operation has its unique opcode.

- **Operands** fields specify where to get the source and destination operands for the operation specified by the opcode.

  The source / destination of operands can be a constant.

Homework

What is the difference between Hardware and Software?