Unit I: Basic Structure of Computers

Basic functional blocks of a computer: CPU, memory, input-output subsystems, control unit. Instruction set architecture of a CPU –registers, instruction execution cycle, RTL interpretation of instructions, addressing modes, instruction set, Instruction set architecture CISC, RISC, Case study –instruction sets of common CPUs

The Evolution of Computers

First Generation Computers(1940-1956)

- 1941-huge,slow,expensive and often unreliable
- ENIAC (Electronic Integrator and Computer)
- It used vacuum tube
- **Vacuum tube**
- electronic tube about the size of light bulbs,
- used as the internal computer components.
- **Problems** : vacuum tubes generated a great deal of heat causing many problems in regulation and climate control
- The tubes also burnt out frequently.

Second generation computers (1956-1963)

- The famous computer scientist - John Bardeer , Walter Houser Brattain •William Shockley
- **Transistor** were smaller than vacuum tubes
- They needed no warm up time
- Consumes less energy
- Generated much less heat
- Faster and more reliable.

Third generations computer(1964-1971)

- The **IBM 370 series** were introduced in 1964.
- It was used for **business and scientific programs**.
- Other computer models introduced were CDC 7600 and B2500
- Silicone chips were manufactured in 1961 at the Silicone Valley
- **integrated circuit technology**
- reduced the size and cost of computers
- the Magnetic Core Memory was replace by a device called the microchip

Fourth generation computers (1971-Present)

- It took only **55 years for the 4 generations** to evolve.
- There are many types of **computer models** such as:

  1 **Apple Macintosh** 2 **Dell** 3 **IBM** 4 **Acer**
- **1971**, Intel created the **first microprocessor**.
- **1976**, **Steve Jobs** built the **first Apple computer**.
- **1981**, **IBM** introduces its **first personal computer**.
- During the fourth generations **hardware technology such as silicone chips, microprocessor and storage devices** were invented.
- The microprocessor is a **large scale integrated circuit** which contained **thousands of transistor**
- The **transistor on this chip are capable of performing all of the function** of a computer's central processing unit.

Advantages

- Computers became **100 times smaller than ENIAC**
- **Gain in speed, reliability and storage capacity**
- **Personal and software industry boomed**

Fifth generation computers (Present & Beyond)

- Are technologies more advance
- The fifth generation computers are such as :

1 Silicone chips

2 Processor Robotics

3 Virtual reality

4 Intelligent systems Programs which translate languages

New Era Computers

- Technology of computers are more advance, sophisticated and modern
- The latest invention of the new era are :

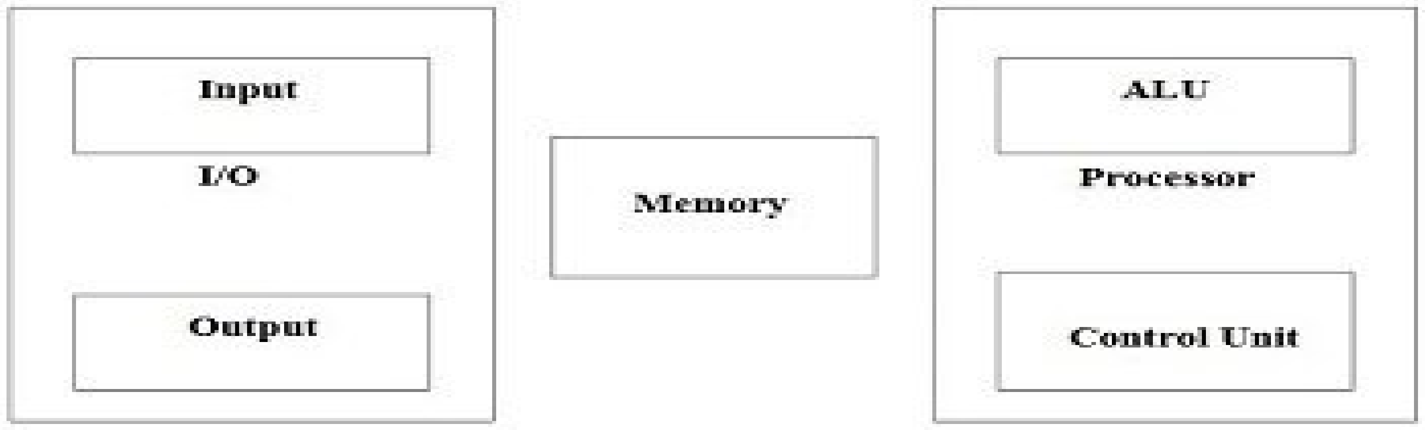
1. Supercomputers
2. Mainframe computers
3. Mini computers
4. Personal computer
5. Mobile computer

Functional Unit

functional units :

1. Input unit
2. Output unit
3. Memory unit
4. Arithmetic & logic unit
5. Control unit.

Figure the functional units of a computer system.



1. Input Unit:

- Computer accepts encoded information through input unit.
- The standard input device is a keyboard. Whenever a key is pressed, keyboard controller sends the code to CPU/Memory.

Examples include Mouse, Joystick, Tracker ball, Light pen, Digitizer, Scanner etc.

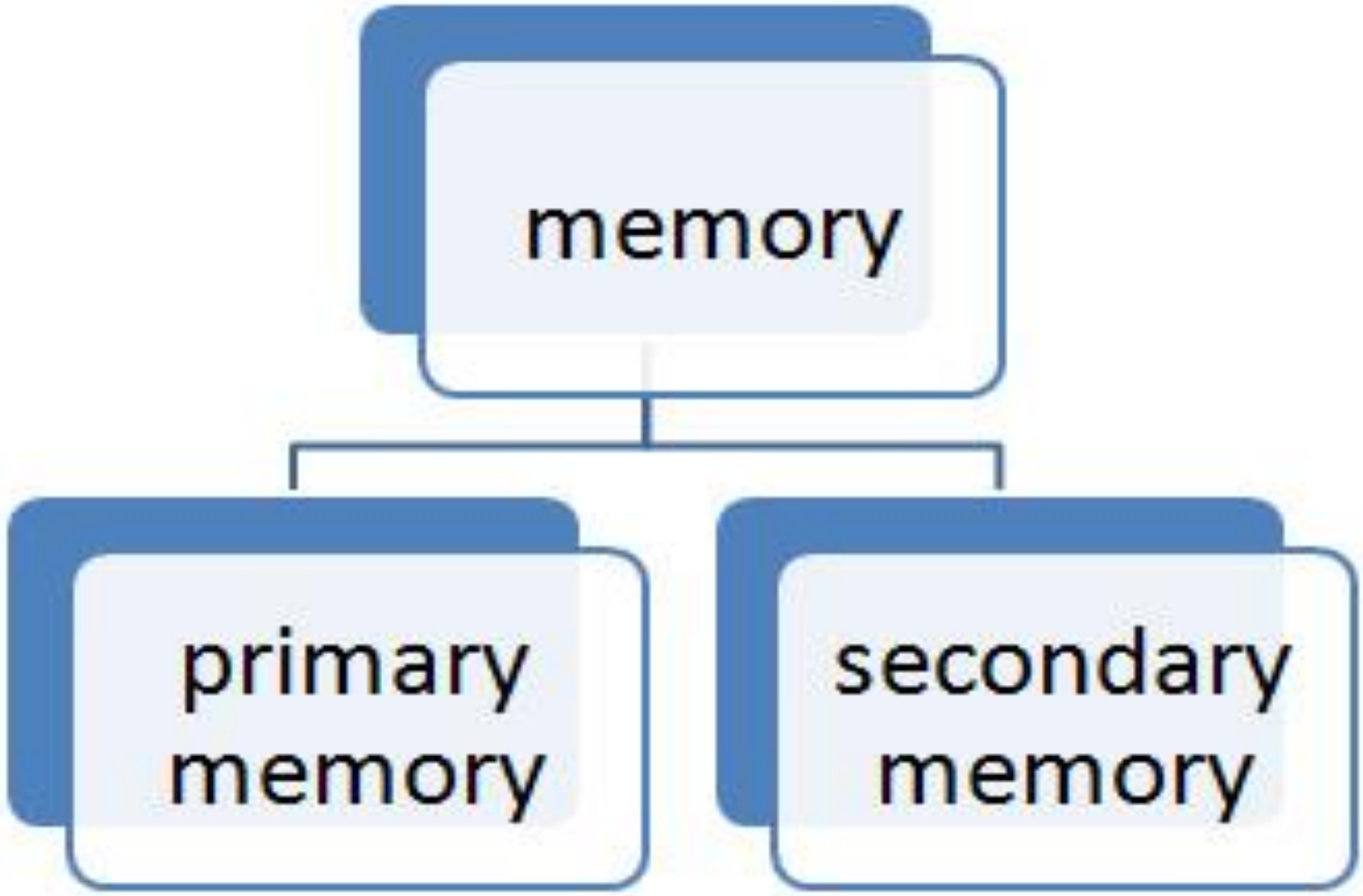
2. **Memory Unit**: Memory unit stores the program instructions (Code), data and results of computations etc.

Memory unit is classified as:

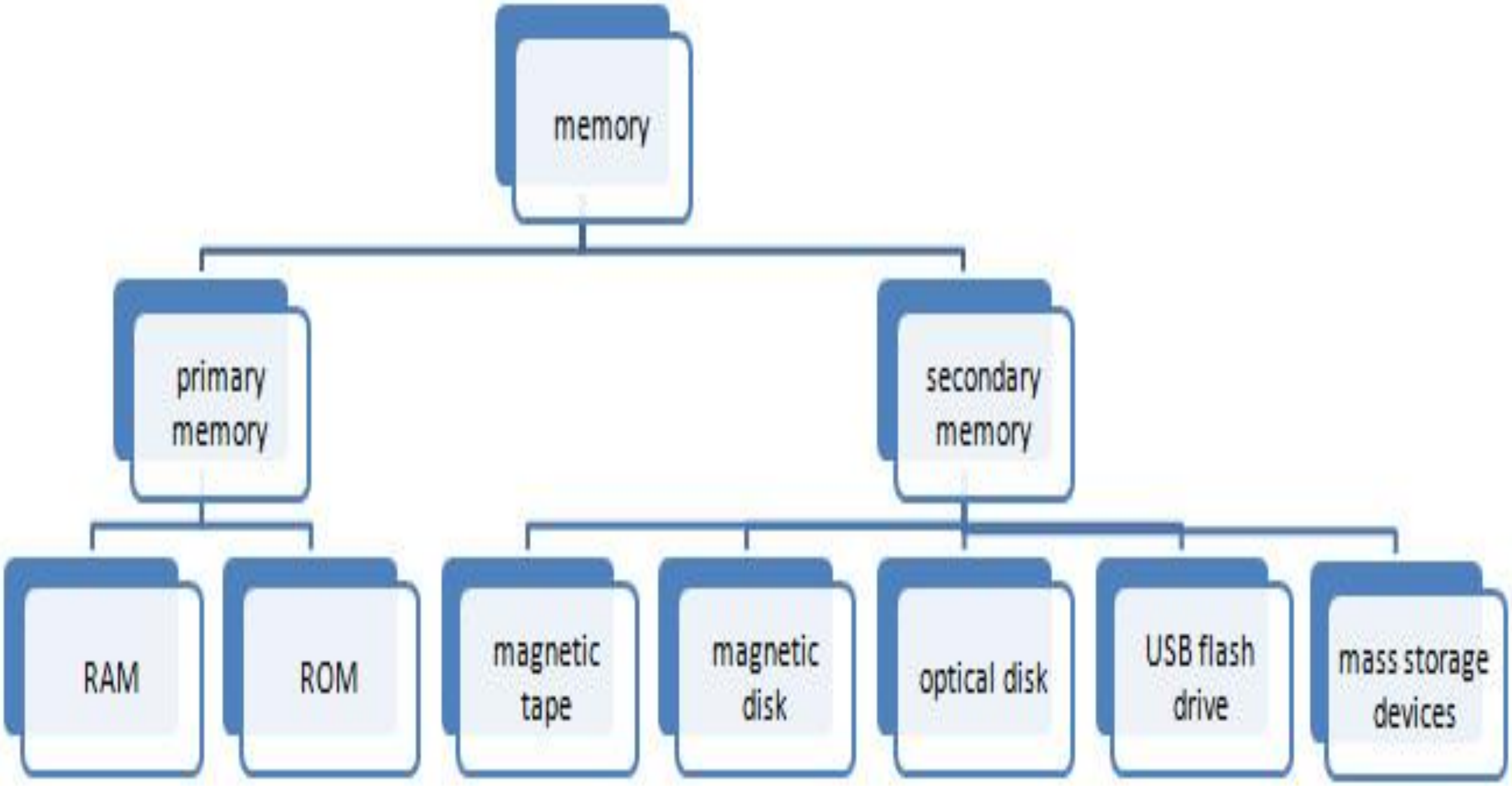
2.1 Primary /Main Memory

2.2 Secondary /Auxiliary Memory

classification of computer memory



Classification of Primary Memory and Secondary Memory



1. Primary memory:

- Primary memory is directly accessed by the CPU.
- The CPU continuously reads instructions stored in the primary memory and executes them.
- CPU is also stored.
- The information is transferred to various locations through the BUS.

Primary memories are of **two types.: 1.** RAM   2. ROM

1. RAM:

- It stands for **Random Access Memory**.
- data can be **stored temporarily**, so this type of memory is called as **temporary memory or volatile memory** because when **power fails the data from RAM will be erased**.
- RAM is of distinct **types l**ike **SRAM, DRAM, and VRAM**.

2 ROM:

- It stands for **Read Only Memory**.
- information can simply be read by the user but cannot add new data or it cannot be modified.
- ROMs are of distinct **types**:

1. **PROM** – Programmable Read Only Memory
2. **EPROM** – Erasable Programmable Read Only Memory
3. **EEPROM** – Electrically Erasable Programmable Read Only Memory

Secondary memory:

- Secondary memory or auxiliary memory consists of **slower and less expensive** device
- **communicates indirectly with CPU** via main memory.
- The secondary memory **stores the data and keeps it even when the power fails.**
- It is used to **store or save large data or programs** or other information.

The **secondary storage devices** are explained below:

- **Magnetic disks** :Floppy disks, Hard disks
- **Magnetic tape**
- **Optical disk:** CD – ROM,DVD – ROM,CD – RECORDABLE,CD – REWRITABLE,PHOTO – CD
- USB flash drive
- Mass storage devices

3. Arithmetic and logic unit(ALU):

- ALU consist of necessary logic circuits like adder, comparator etc., to perform operations of addition, multiplication, comparison of two numbers etc.

4. Output Unit:

- Computer after computation returns the computed results, error messages, etc. via output unit.
- E.g. video monitor, LCD/TFT monitor, printers, plotters etc.

5. **Control Unit**:

- Control unit co-ordinates activities of all units by issuing control signals.
- Control signals issued by control unit govern the data transfers and then appropriate operations take place.
- Control unit interprets or decides the operation/action to be performed.

The operations of a computer can be summarized as follows:

- 1. A **set of instructions** called a program reside in the main memory of computer.
- 2. The **CPU fetches those instructions** sequentially one-by-one from the main memory, **decodes** them and performs the **specified operation** on associated data operands in ALU.
- 3. Processed data and **results will be displayed** on an output unit.
- 4. All activities pertaining to processing and data movement inside the computer machine

lecture topic Question

Q1. Difference between primary memory and secondary memory?

Q2. Difference between RAM and ROM ?

Or

Q3. explain volatile and non-volatile memory?

Q4. difference between DRAM and SRAM?

Q5. Explain Different parts of Central processing units?

Q6 Explain basic components of computer?

What are the differences between Primary and Secondary Memory?

Primary memory	Secondary memory
The memory devices used for primary memory are semiconductor memories	The secondary memory devices are magnetic and optical 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 secondary memory is always non volatile
Primary memory is known as main memory	Secondary memory is known as additional memory or back memory
These memories are also called as internal memory	These memories are also called as external memory
The primary memory devices are	The secondary memory devices are

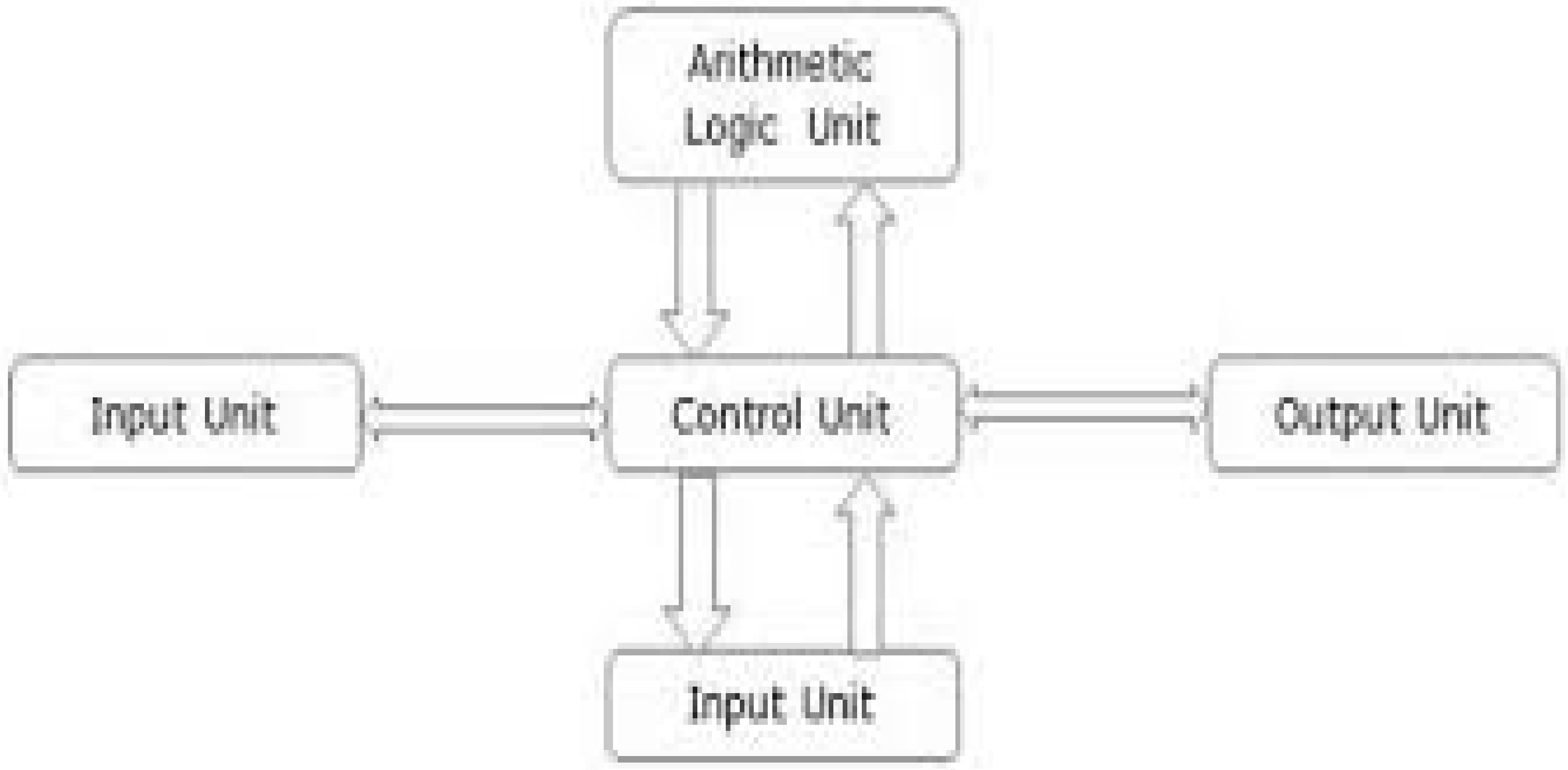
RAM	ROM
1. Temporary Storage.	1. Permanent storage.
2. Store data in MBs.	2. Store data in GBs.
3. Volatile.	3. Non-volatile.
4.Used in normal operations.	4. Used for startup process of computer.
5. Writing data is faster.	5. Writing data is slower.

Difference between RAM and ROM

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

Basic Components of Computer



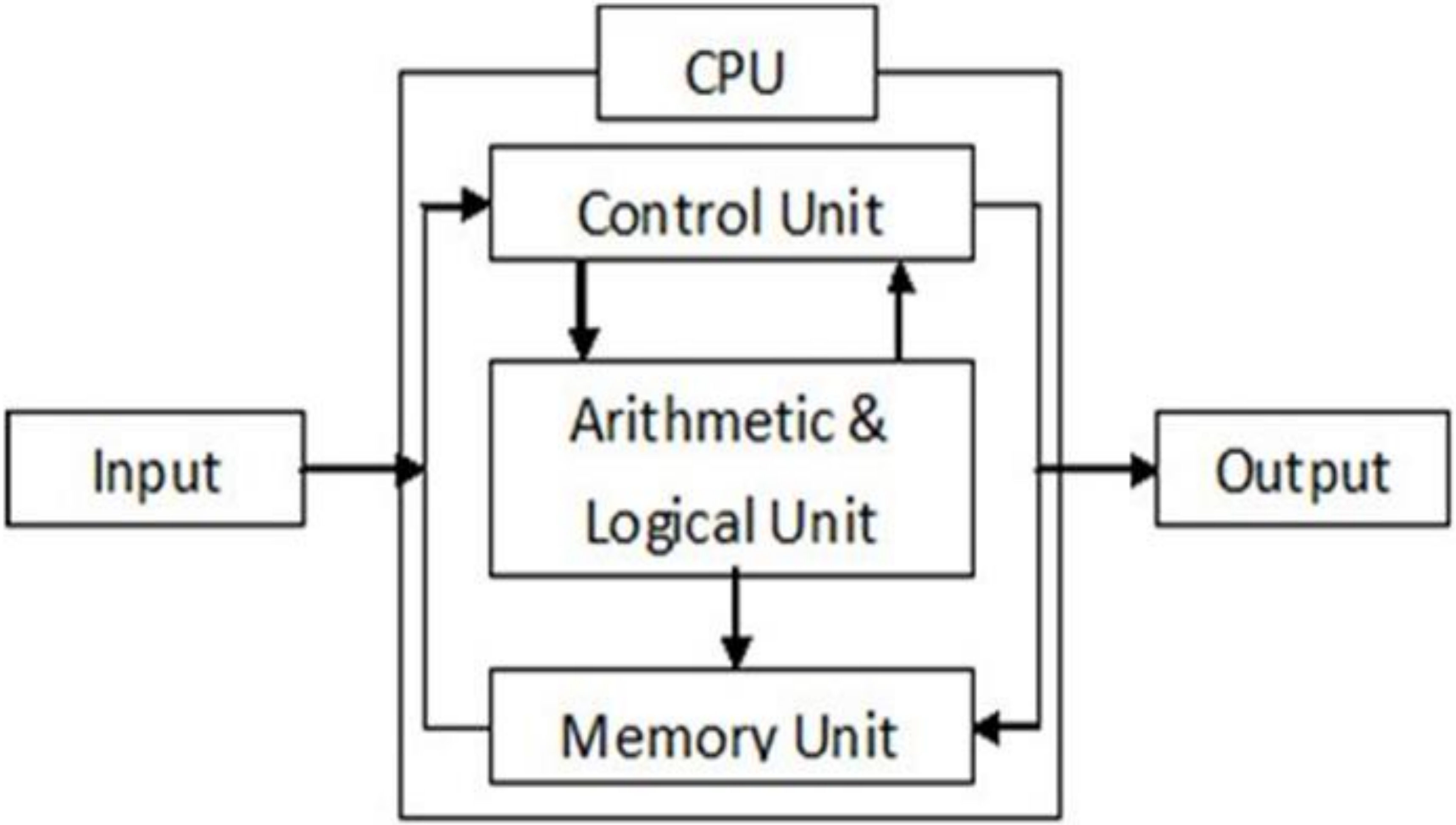


Fig. Block Diagram of Computer

Basic Components of Computer

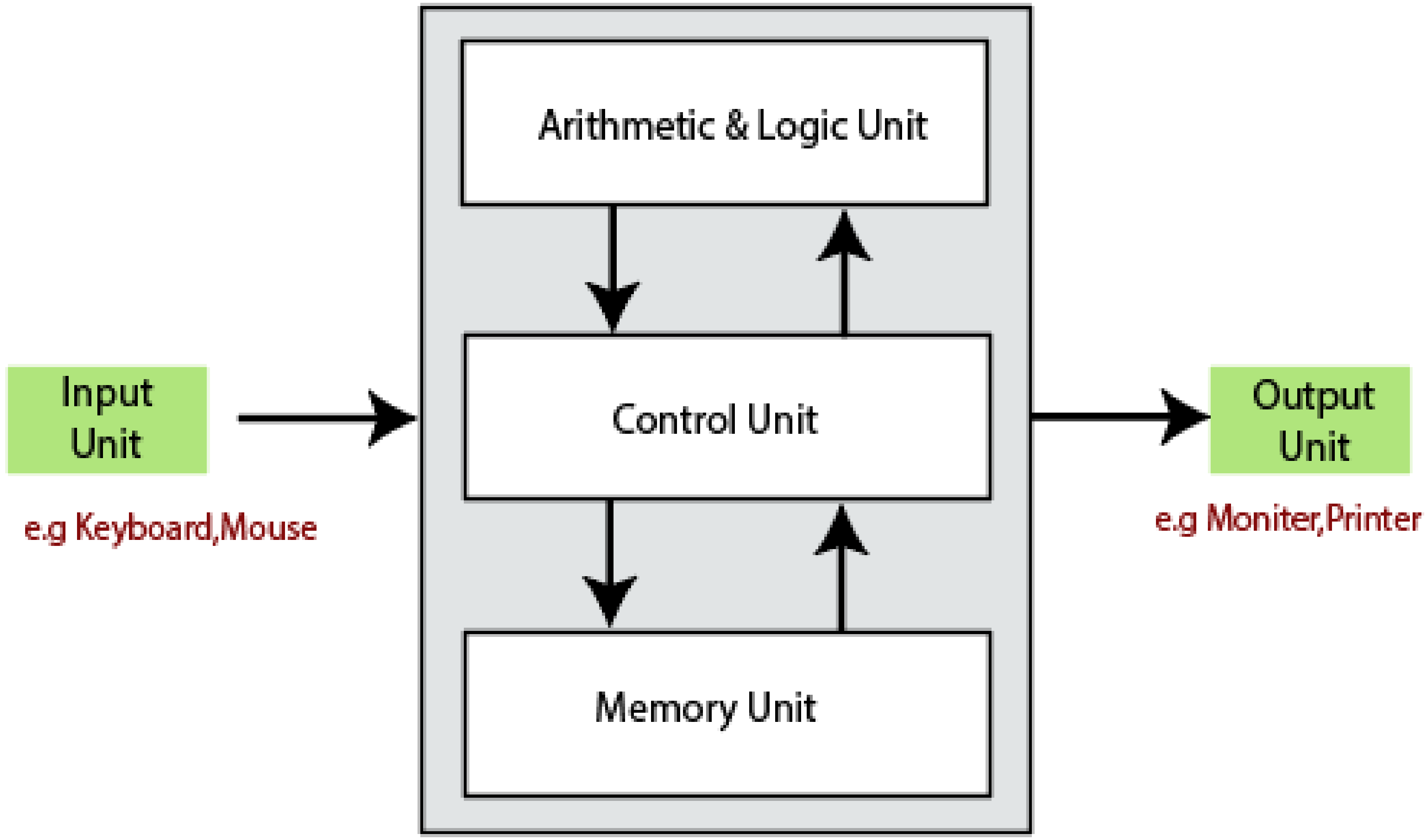
- CPU (Central Processing Unit)
- Storage Unit
- ALU(Arithmetic Logic Unit)
- Control Unit

Basic Components of Computer

1 Central Processing Unit (CPU)

- The computer system is nothing without the Central processing Unit.
- it is also known as the brain or heat of computer. The CPU is an electronic hardware device which can perform different types of operations such as arithmetic and logical operation.
- The CPU contains two parts:
- the arithmetic logic unit and control unit.

Central Processing Unit (CPU)



Basic Components of Computer

2 Control Unit

- The control unit (CU) controls all the activities or operations which are performed inside the computer system.
- It receives instructions or information directly from the main memory of the computer.
- When the control unit receives an instruction set or information, it converts the instruction set to control signals .
- these signals are sent to the central processor for further processing.
- The control unit understands which operation to execute, accurately, and in which order.

Basic Components of Computer

3 Arithmetic and Logical Unit

- The arithmetic and logical unit is the combinational digital electronic circuit that can perform arithmetic operations on integer binary numbers.
- It presents the arithmetic and logical operation.
- The outputs of ALU will change asynchronously in response to the input.
- The basic arithmetic and bitwise logic functions are supported by ALU.

Lecture topic Questions

Q1. List out and explain components of computer system?

Q2 Draw and explain basic operation of Computer System?

Q3. List all operation steps in detail?

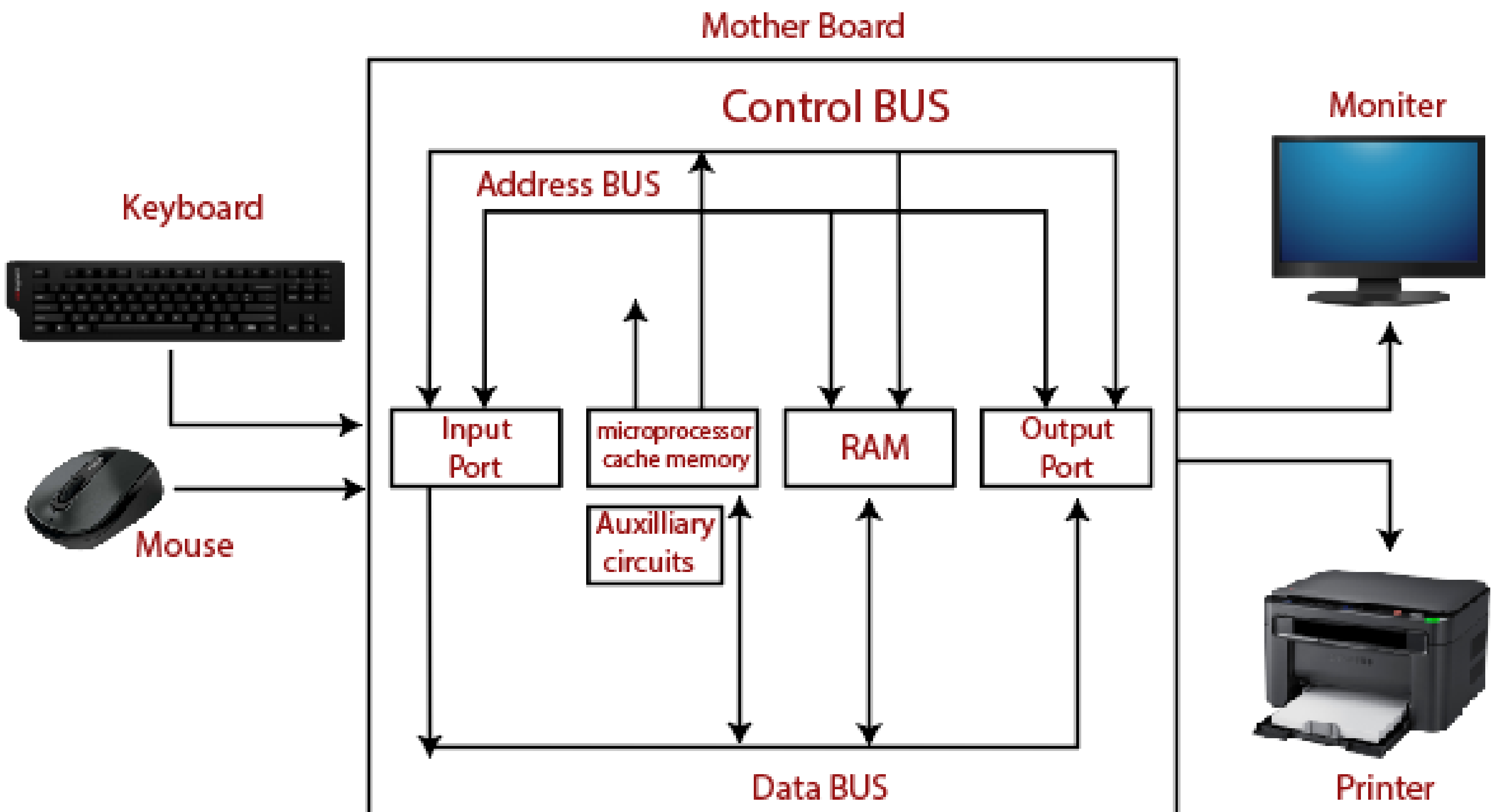
Q4. **Scenario based Question** –

e.g. If you want purchase a laptop. What are different components consider, write in detail.

Q5. **Scenario based Question-**

e.g. in Computer dept. we want to design high end configuration lab of computers. List of in detail different components are required with configuration.

Components of Computer System



Components of a Computer System

Components of Computer System

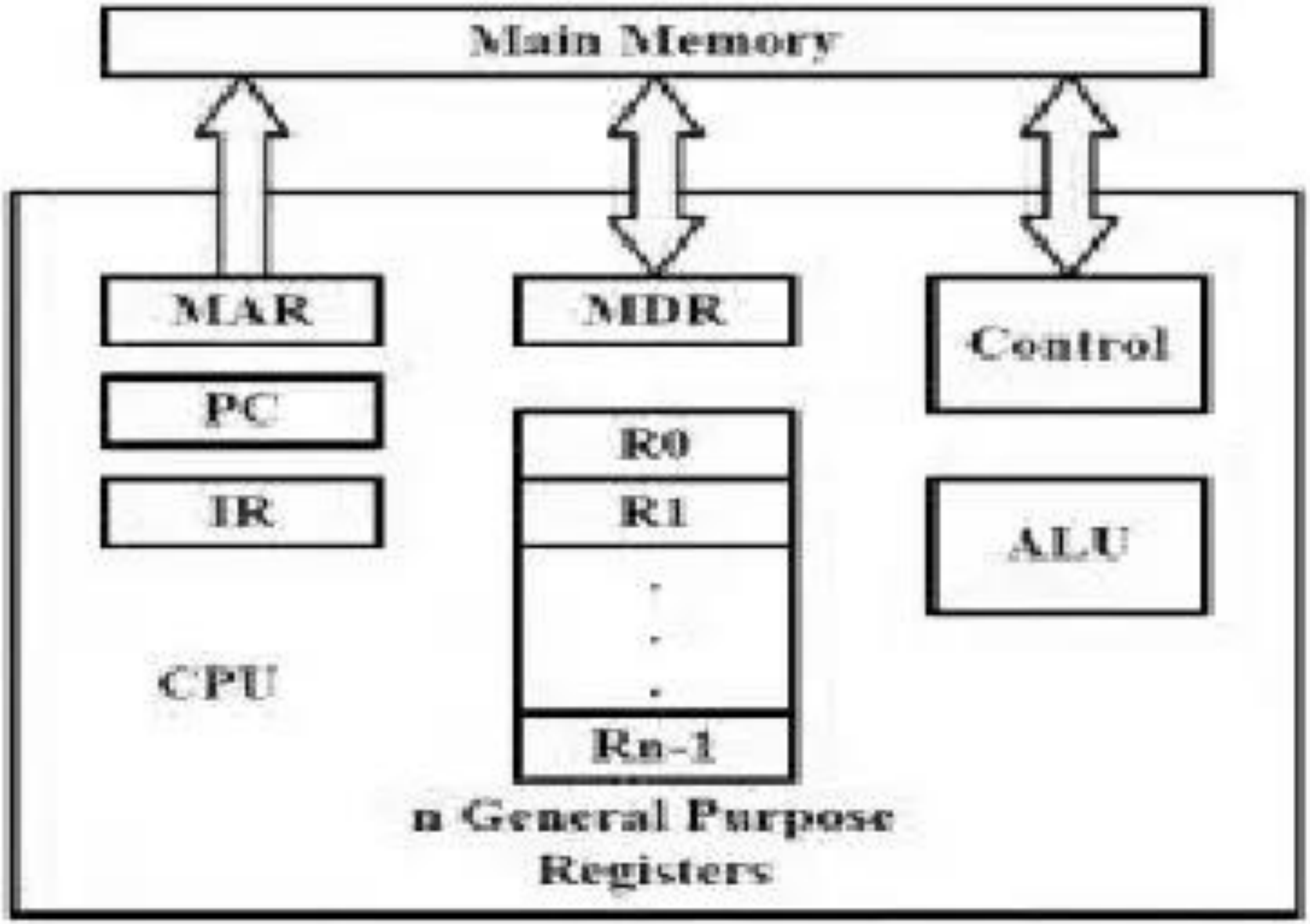
Design high end computer system using latest configuration.(laptop or desktop)

- **Processor.**

Eg.i5,i7,i8 ,no. processor, manufacture ,processor speed

- **Main Memory.-** RAM size
- **Secondary Memory**.-HDD
- **Input Devices**.-K/B, mouse
- **Output Devices**.-Monitor size, printer

Basic Operational of Computer



Basic operational concepts

Program Counter (PC)-

- This is **specialized register** that keeps track of execution of a program.
- It contains the **memory address of the next instruction** to be fetched and executed.

MAR–(Memory Address Register):-

- It holds the address of the location to be accessed.

MDR–(Memory Data Register):-

- It contains the data to be written into or read out of the address location.

Registers

MAR	Memory Address Register	Holds the memory location of data that needs to be accessed
MDR	Memory Data Register	Holds data that is being transferred to or from memory
AC	Accumulator	Where intermediate arithmetic and logic results are stored
PC	Program Counter	Contains the address of the next instruction to be executed
IR	Instruction Register	Contains the current instruction during processing

Operating steps

1. **Programs reside in the memory** & usually get these through the I/P unit.
2. **Execution of the program starts** when the **PC is set to point** at the first instruction of the program.
3. Contents of **PC are transferred to MAR** and a Read Control Signal is sent to the memory.
4. After the time required to access the memory elapses, the address word is read out of the **memory and loaded into the MDR**.
5. Now contents of **MDR are transferred to the IR** & now the instruction is ready to be decoded and executed.
6. If the **instruction involves an operation by the ALU**, it is necessary to obtain the required operands.
7. An operand in the **memory is fetched by sending its address to MAR** & Initiating a read cycle.
8. When the **operand has been read from the memory to the MDR**, it is transferred from MDR to the ALU.
9. After **one or two such repeated cycles**, the ALU can perform the desired operation.
10. If the **result of this operation is to be stored in the memory**, the result is sent to MDR.
11. Address of location where the **result is stored is sent to MAR** & a **write cycle** is initiated.
12. The contents **of PC are incremented** so that PC points to the next instruction that is to be executed

Questions

Q1. Explain bus structure with diagram?

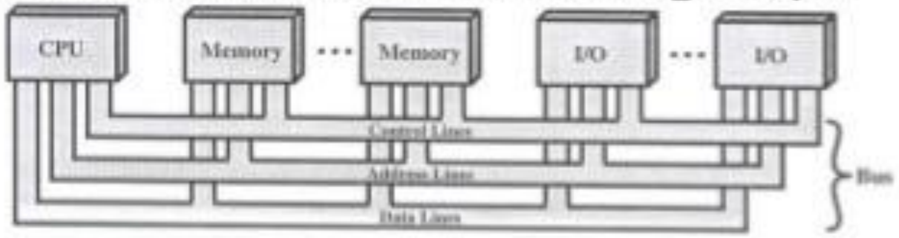
Q2. Differential between single bus and multiple bus structure?

Q3.Expalin how to measure performance of computer?

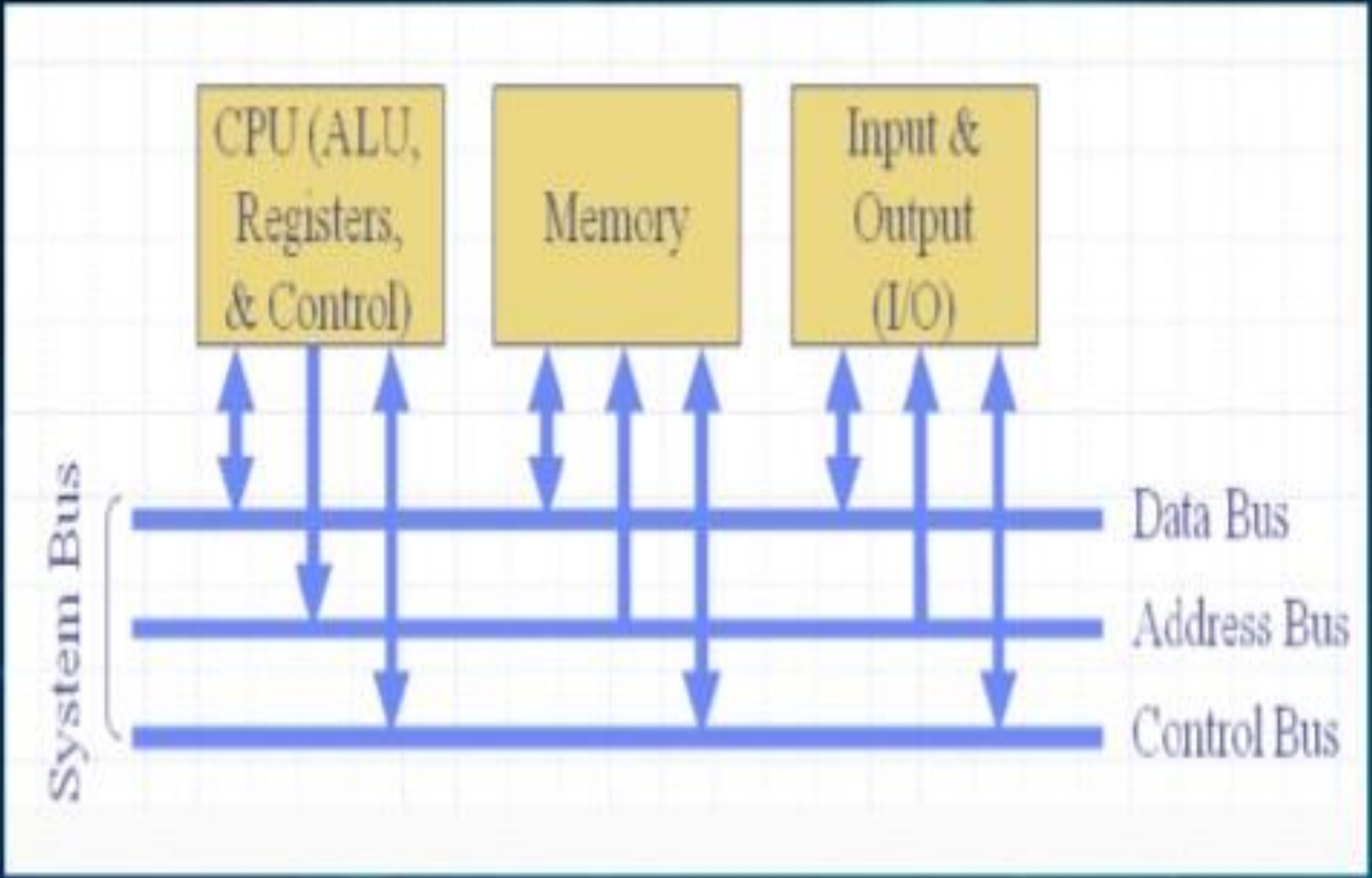
Q4. write a equation and explain various parameter used to measuring performance of computer?

Bus Structure

- A system bus consists of 50-100 lines.
- Each line is assigned a particular meaning or function.
- On any bus the lines can be classified into 3 groups
  - Data lines
  - Address lines
  - Control lines



SYSTEM BUS MODEL



Computer Buses

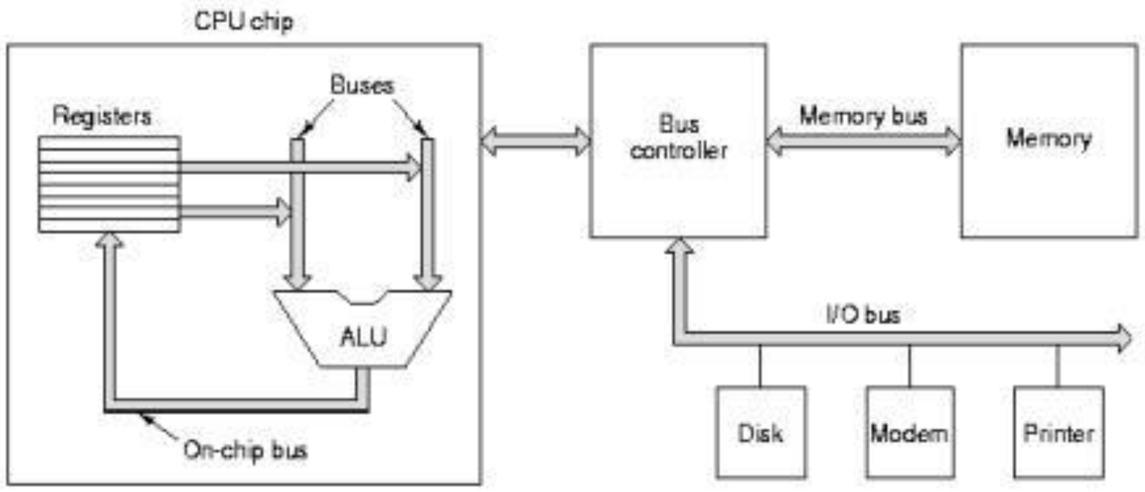


Figure 3-34. A computer system with multiple buses.

Computer Buses

Question: Define computer bus and discuss different types of computer buses.

Computer Bus:

Computer buses are fine conducting wires used to carry electrical signals between two units in a computer system.

Computer buses are three types such as:

(1) Data Bus
(2) Address Bus
(3) Control Bus

Data Bus:

Data buses are 8, 16, 32, 64, or 128 parallel lines used to carry data signal between two units in a computer system. Data buses are bidirectional lines.

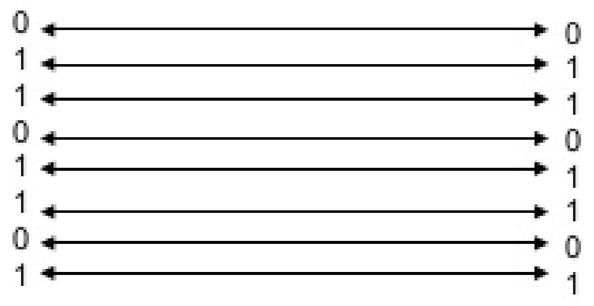


Figure 1: An 8-bit data bus.

Address Bus:

Address buses are 16, 32, 64, 128, or 256 parallel lines used to carry address signal between processor and other units in a computer system. Address buses are unidirectional lines.

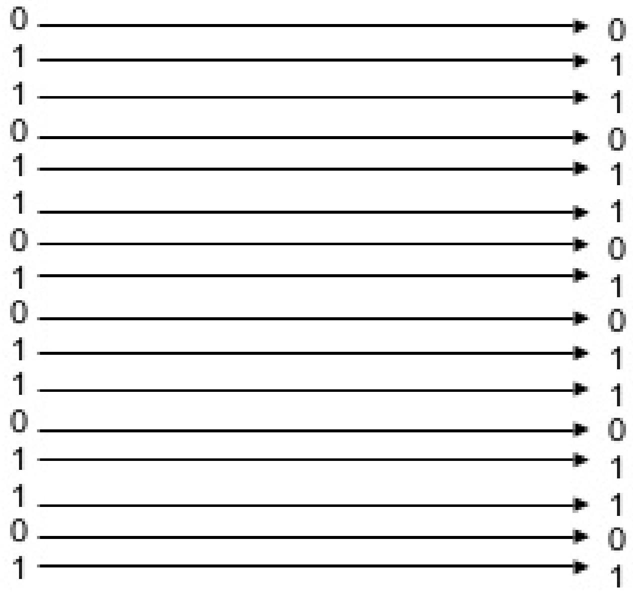


Figure 2: A 16-bit address bus.

Control Bus:

Control bus is a single line used to carry control signal between processor and other units in a computer system. control bus is unidirectional lines. The possible control signals are $\overline{\text{MEMR}}$, $\overline{\text{MEMW}}$, $\overline{\text{IOR}}$, and $\overline{\text{IOW}}$.

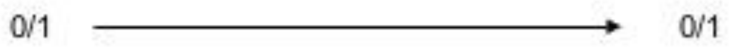
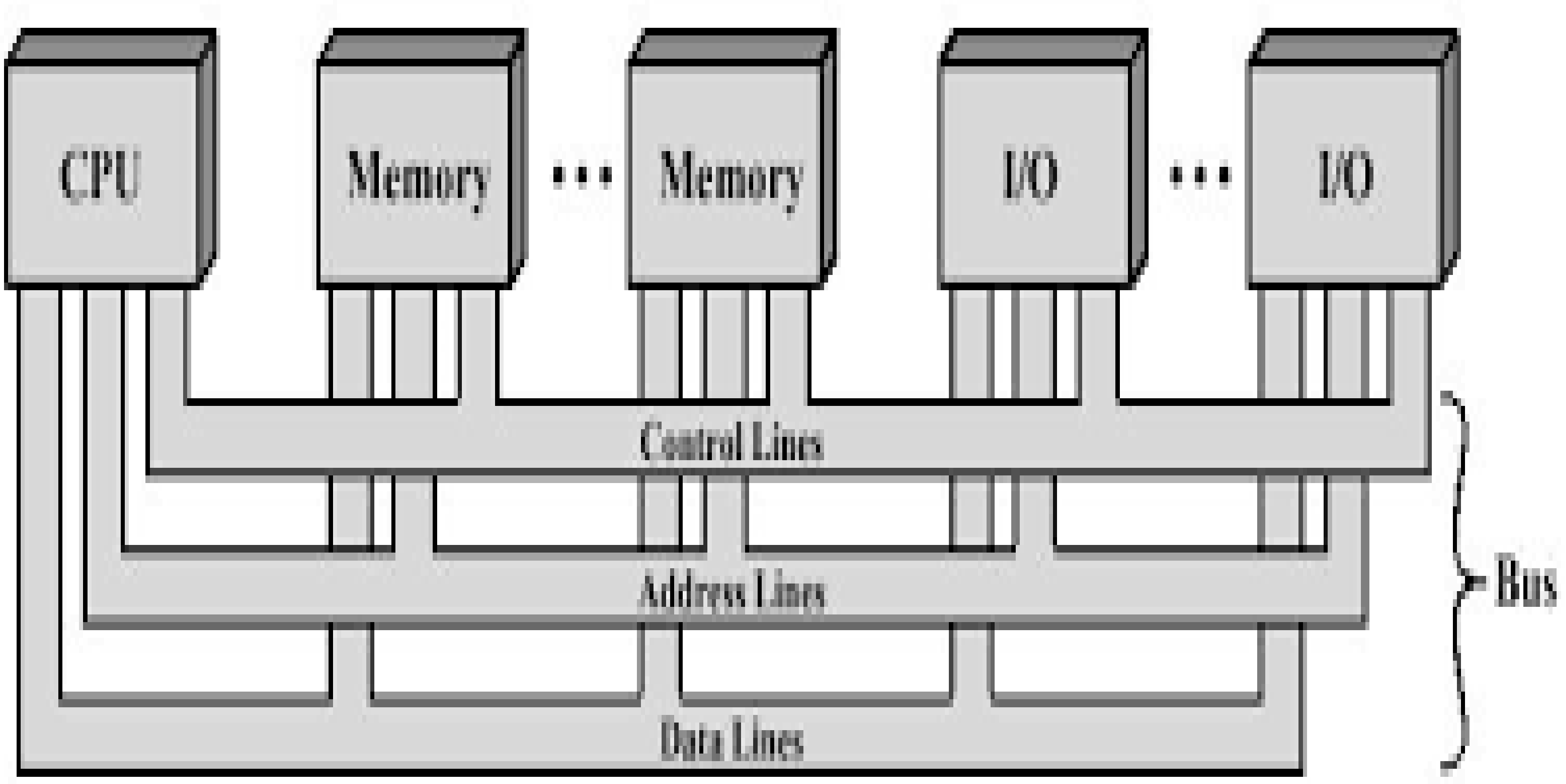


Figure 3: A control bus.

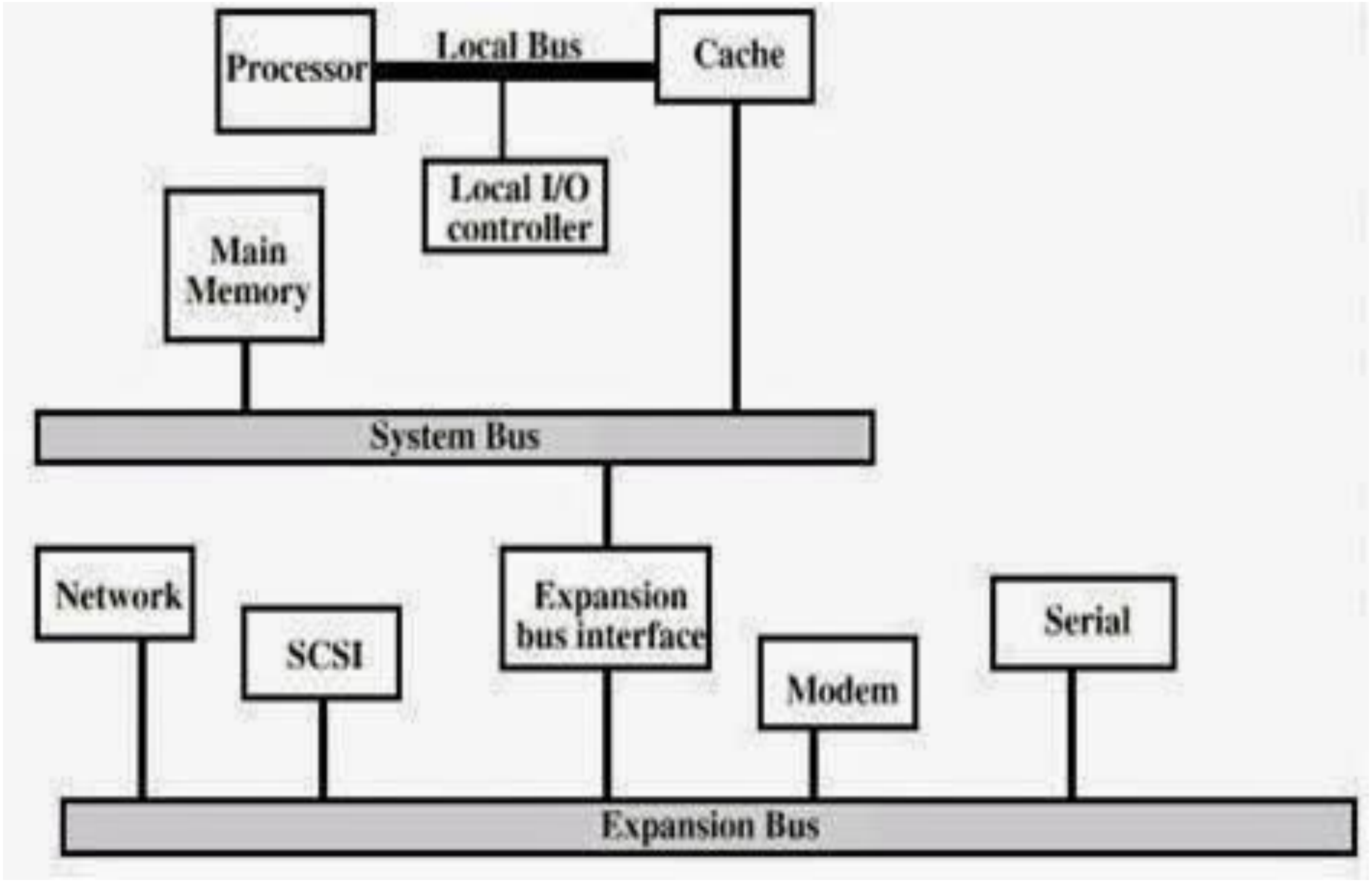
Computer Buses

- A number of buses are in widespread use in the computer world.
  - Multibus (8086)
  - IBM PC (PC/XT)
  - ISA bus (PC/AT)
  - EISA bus (80386)
  - Microchannel (PS/2)
  - PCI bus (Many PCs)
  - Nubus (macintosh)
  - Universal Serial Bus (modern PCs)
  - FireWire (consumer electronics)

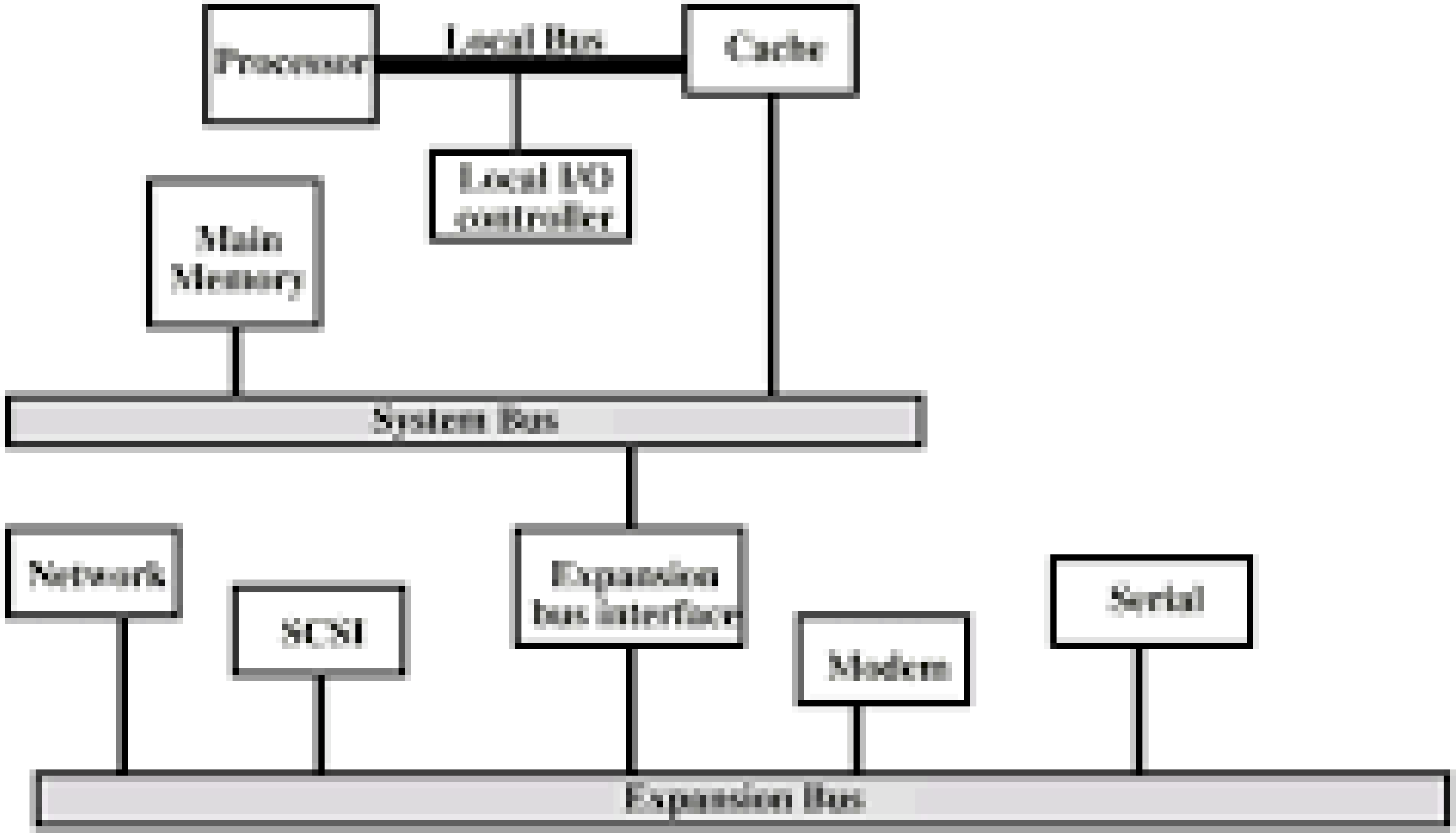
Single BUS


CPU
Memory
Memory
I/O
I/O
Control Lines
Address Lines
Data Lines
Bus

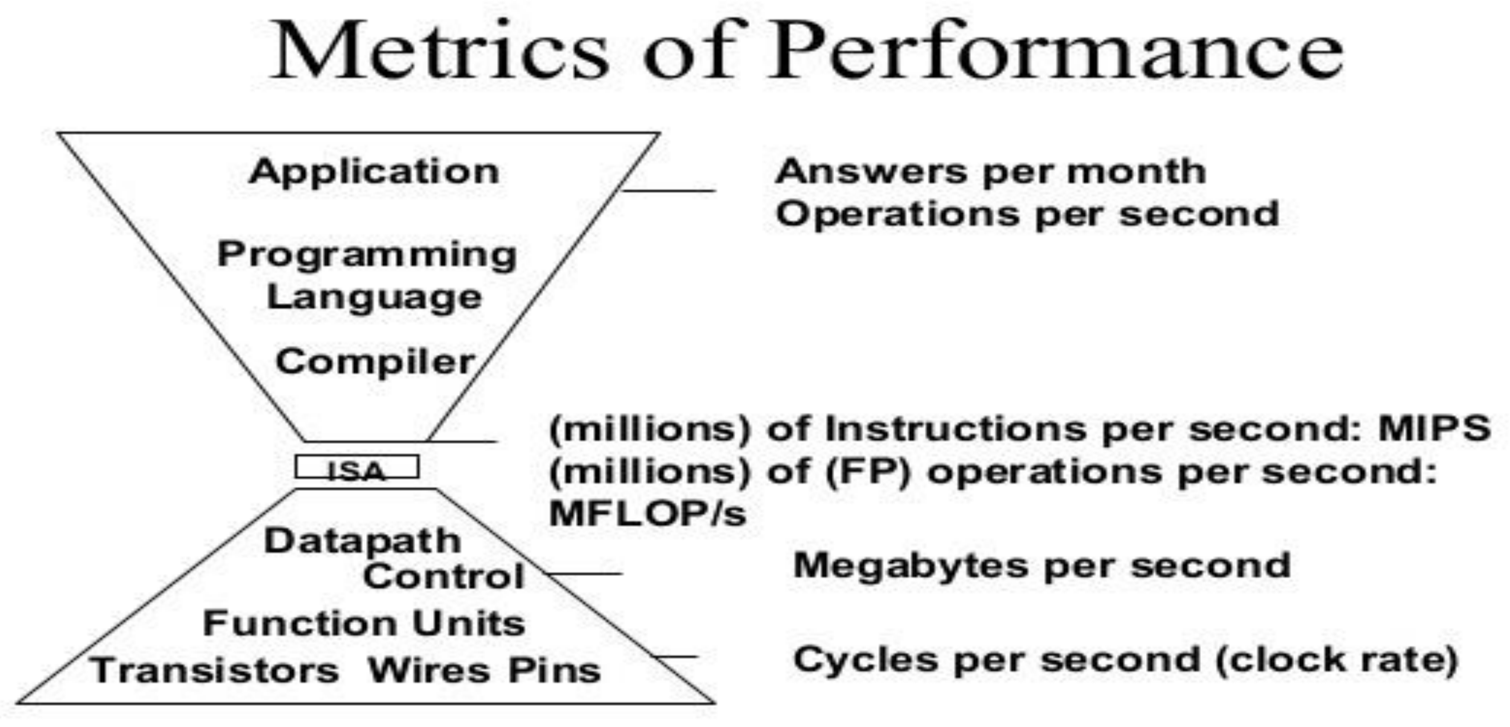
Multiple Hierarchical BUS



Traditional Bus Architecture



Performance Measures



Performance Metrics

- Possible measures:
  - response time – time elapsed between start and end of a program
  - throughput – amount of work done in a fixed time
- The two measures are usually linked
  - A faster processor will improve both
  - More processors will likely only improve throughput
  - Some policies will improve throughput and worsen response time
- What influences performance?

Performance Equation

The total amount of time (t) required to execute a particular benchmark program is

$$t = N * C/f, \qquad \text{or equivalently}$$

$$P = I * f/N$$

where

- P = 1/t is "the performance" in terms of time-to-execute
- N is the number of instructions actually executed (the instruction path length).
- f is the clock frequency in cycles per second.
- C= is the average cycles per instruction (CPI) for this benchmark.
- I= is the average instructions per cycle (IPC) for this benchmark.

Performance Equation

An another performance equation- The equation, which is fundamental to measuring computer performance is :

$$\frac{\text{time}}{\text{program}} = \frac{\text{time}}{\text{cycle}} \times \frac{\text{cycles}}{\text{instruction}} \times \frac{\text{instructions}}{\text{program}}$$

where the time per program is the required CPU time.

Commonly Used Metrics

- Nominal capacity: maximum achievable under ideal conditions
  - networks: nominal capacity = bandwidth
- Throughput: requests / unit time (must be high)
- Usable capacity: max throughput for given response time limit (response time must be low)
- Efficiency: usable capacity / nominal capacity
- Utilization: fraction of time resource busy servicing requests (normal is best)
- Idle time
- Reliability: probability of error, MTBE
- Availability: fraction of time system servicing requests
- Mean uptime: MTBF

Question

Q1. Draw and explain Computer System architecture in detail?

Q2. Draw Central processing unit. explain each component in detail?

Q3. Explain all registers of Central processing unit?

Q4. Explain VLSI era in term of computer evaluation?

Q5 List out commonly used performance metrics of computer?

Performance Equation

The total amount of time (t) required to execute a particular benchmark program is

$t = N * C/f$, or equivalently

$P = I * f/N$

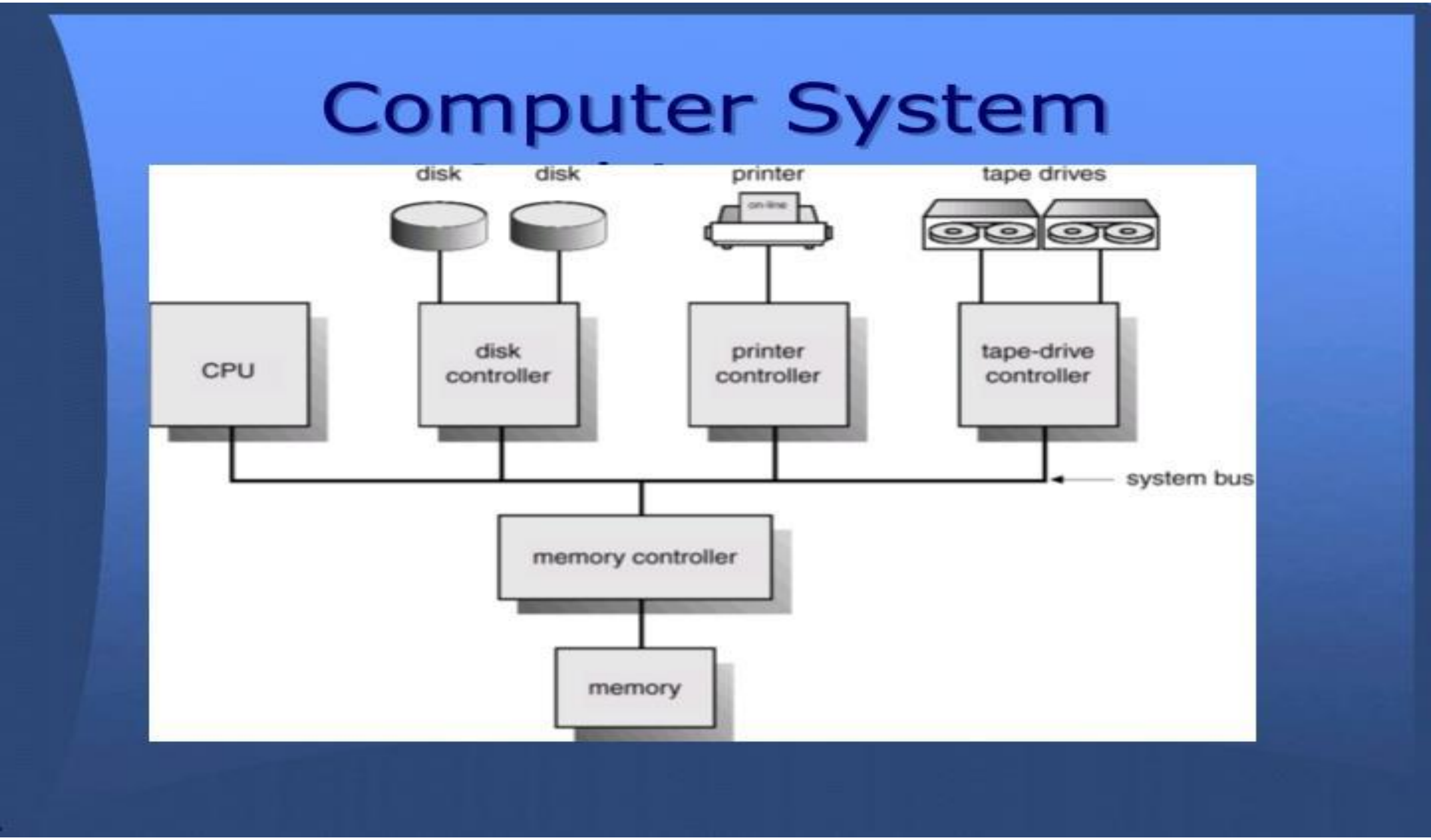
where

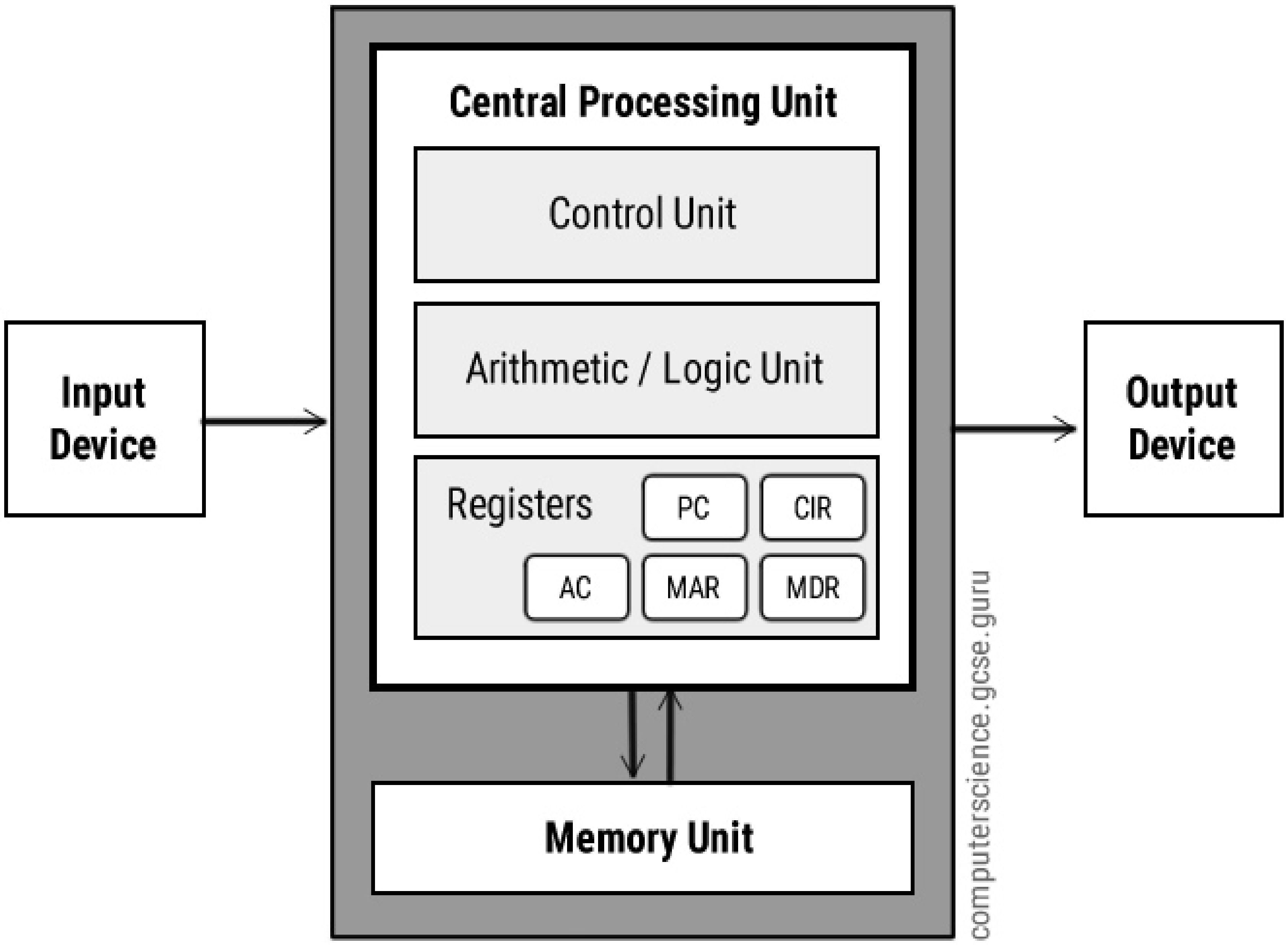
- $P = 1/t$ is "the performance" in terms of time-to-execute
- N is the number of instructions actually executed (the instruction path length).
- f is the clock frequency in cycles per second.
- C= is the average cycles per instruction (CPI) for this benchmark.
- I= is the average instructions per cycle (IPC) for this benchmark.

Commonly Used Metrics

- Nominal capacity: maximum achievable under ideal conditions
  - networks: nominal capacity = bandwidth
- Throughput: requests / unit time (must be high)
- Usable capacity: max throughput for given response time limit (response time must be low)
- Efficiency: usable capacity / nominal capacity
- Utilization: fraction of time resource busy servicing requests (normal is best)
- Idle time
- Reliability: probability of error, MTBE
- Availability: fraction of time system servicing requests
- Mean uptime: MTBF

System Architecture

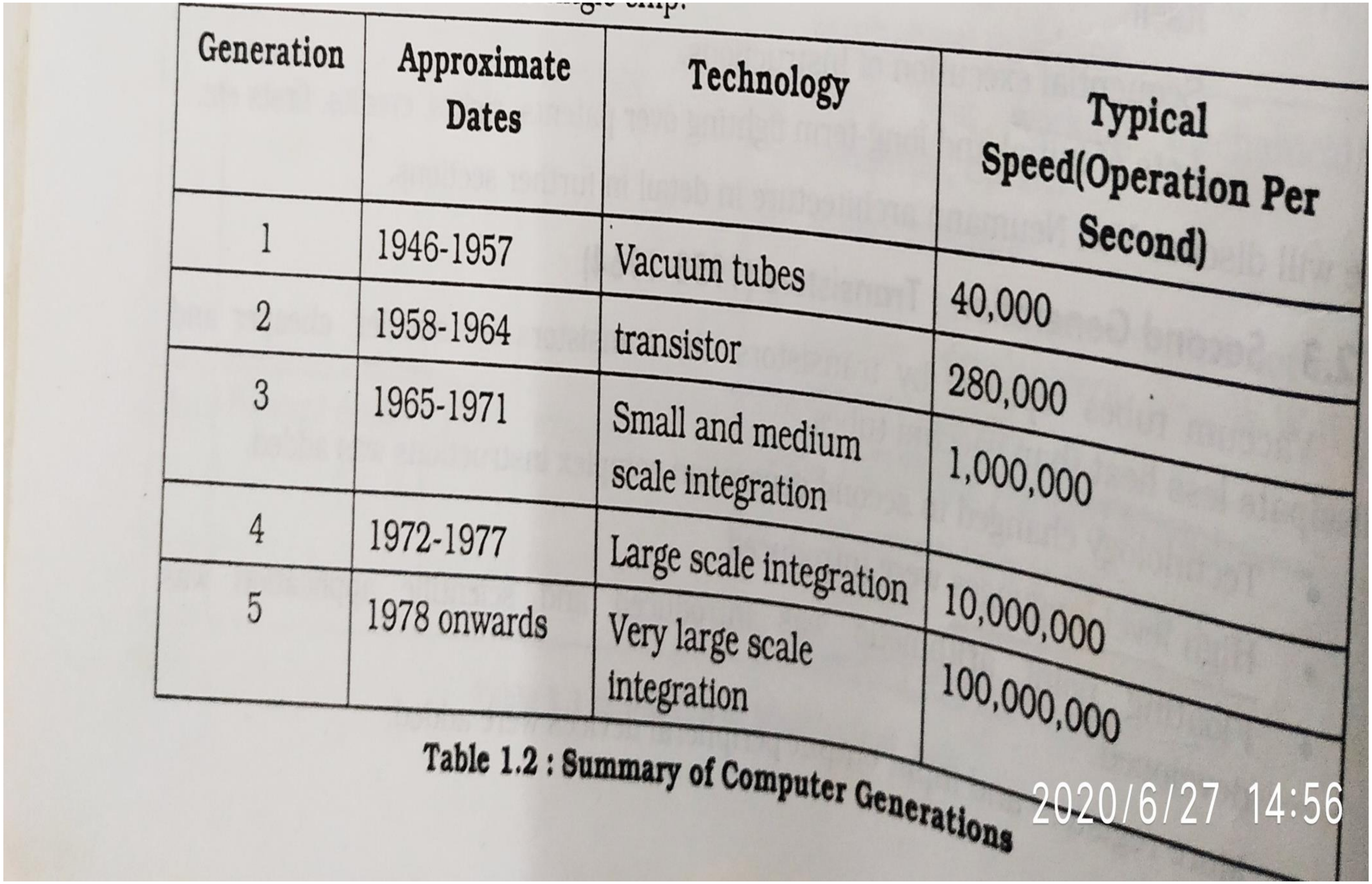



Central Processing Unit
Control Unit
Arithmetic / Logic Unit
Registers
PC
CIR
AC
MAR
MDR
Input Device
Output Device
Memory Unit

Registers

MAR	Memory Address Register	Holds the memory location of data that needs to be accessed
MDR	Memory Data Register	Holds data that is being transferred to or from memory
AC	Accumulator	Where intermediate arithmetic and logic results are stored
PC	Program Counter	Contains the address of the next instruction to be executed
CIR	Current Instruction Register	Contains the current instruction during processing

VLSI era



Generation	Approximate Dates	Technology	Typical Speed(Operation Per Second)
1	1946-1957	Vacuum tubes	40,000
2	1958-1964	transistor	280,000
3	1965-1971	Small and medium scale integration	1,000,000
4	1972-1977	Large scale integration	10,000,000
5	1978 onwards	Very large scale integration	100,000,000

Table 1.2 : Summary of Computer Generations

2020/6/27 14:56

Buses

Address Bus	Carries the addresses of data (but not the data) between the processor and memory
Data Bus	Carries data between the processor, the memory unit and the input/output devices
Control Bus	Carries control signals/commands from the CPU (and status signals from other devices) in order to control and coordinate all the activities within the computer

Von Neumann Architecture

- Von Neumann Architecture-EDVAC(Electronics discrete variable computer)
- 1 st generation computer
- Further developed –Institute for advanced studies(IAS)
- Data format- 40 bit(first bit sign bit, remaining data bit
- Sign bit(0=positive 1 for negative bit)

put (I/O) eq

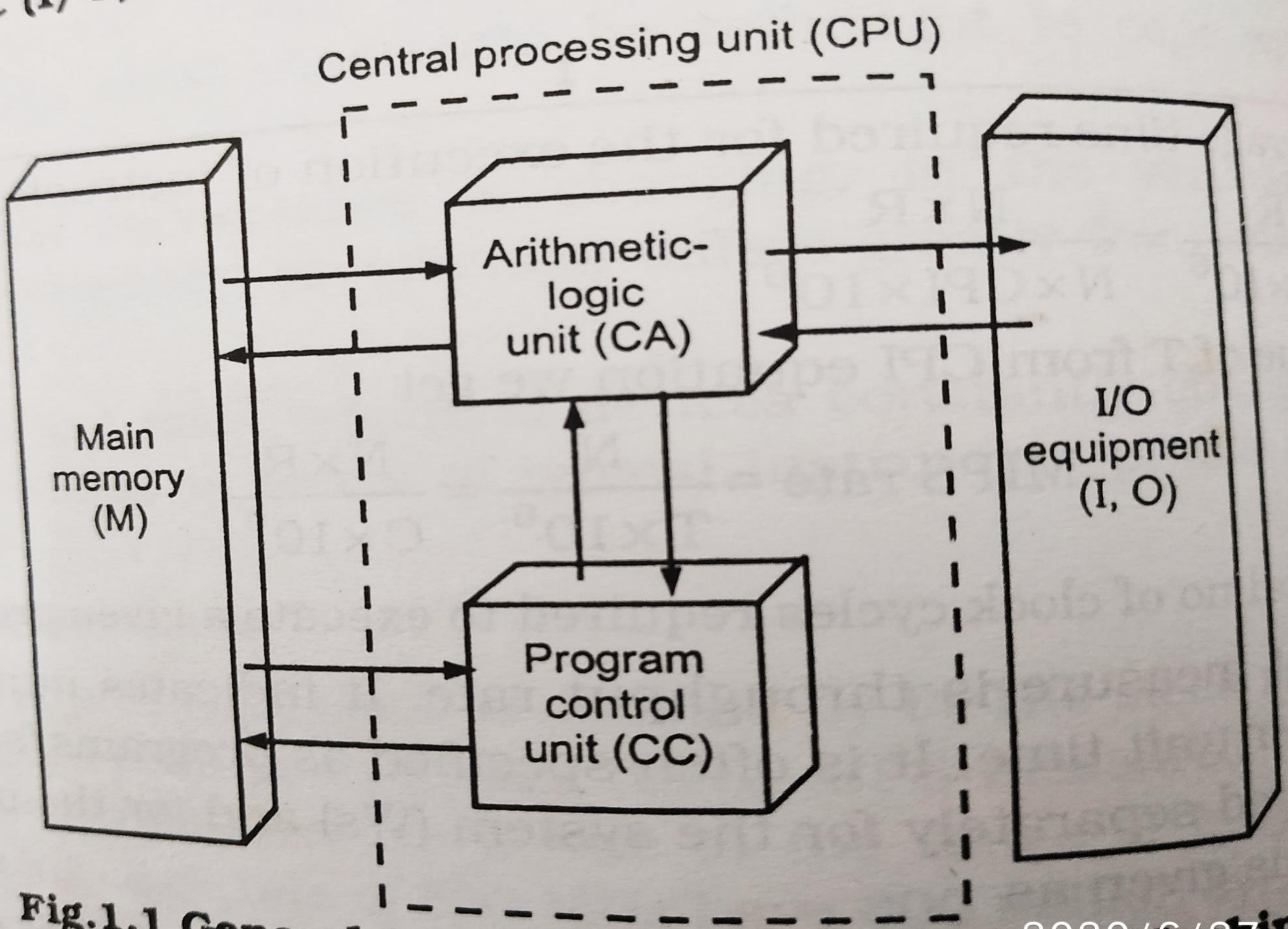
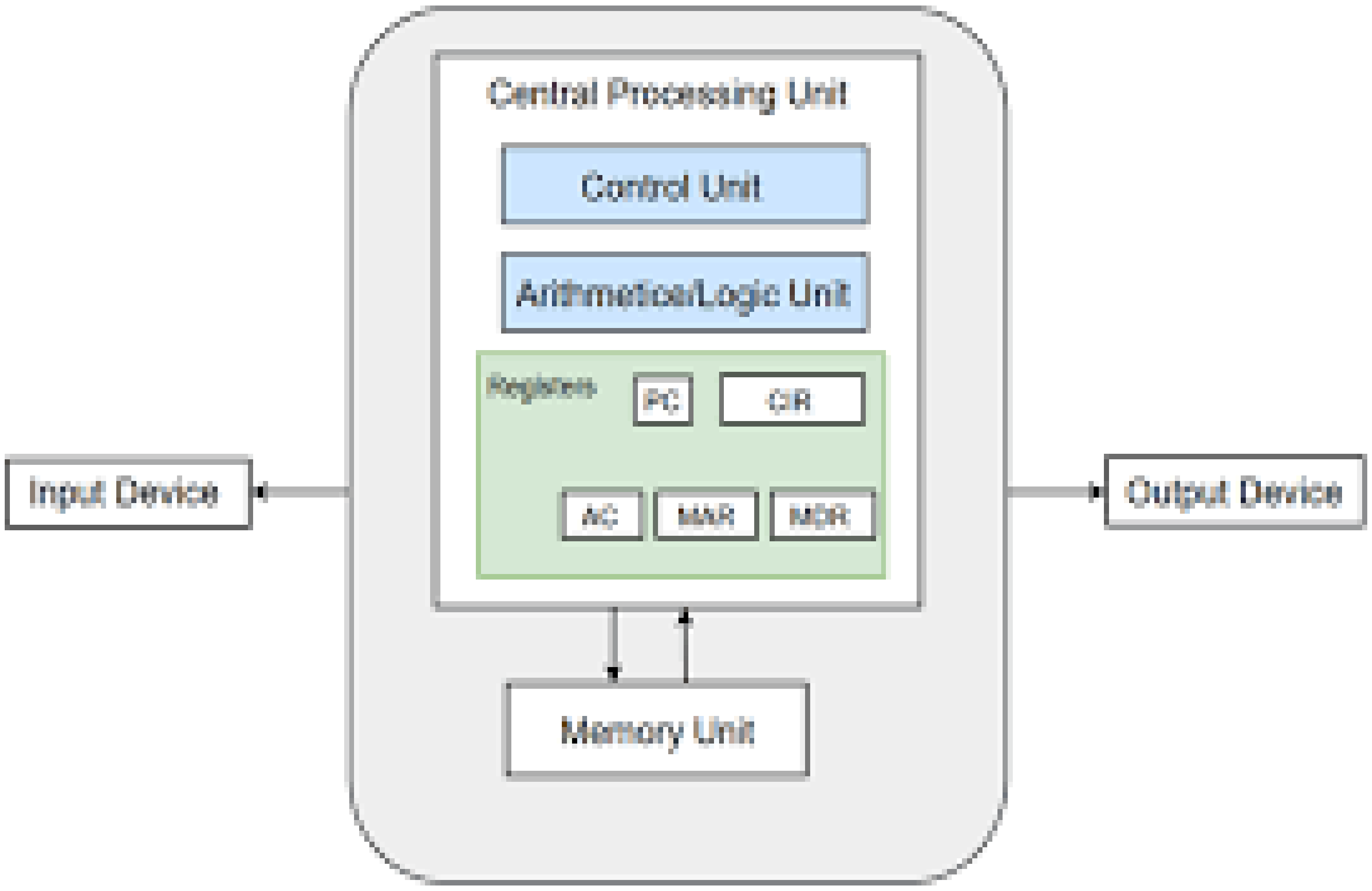

Central processing unit (CPU)
Main memory (M)
Arithmetic-logic unit (CA)
Program control unit (CC)
I/O equipment (I, O)

Fig.1.1 General Structure of Von-Neuman Machine

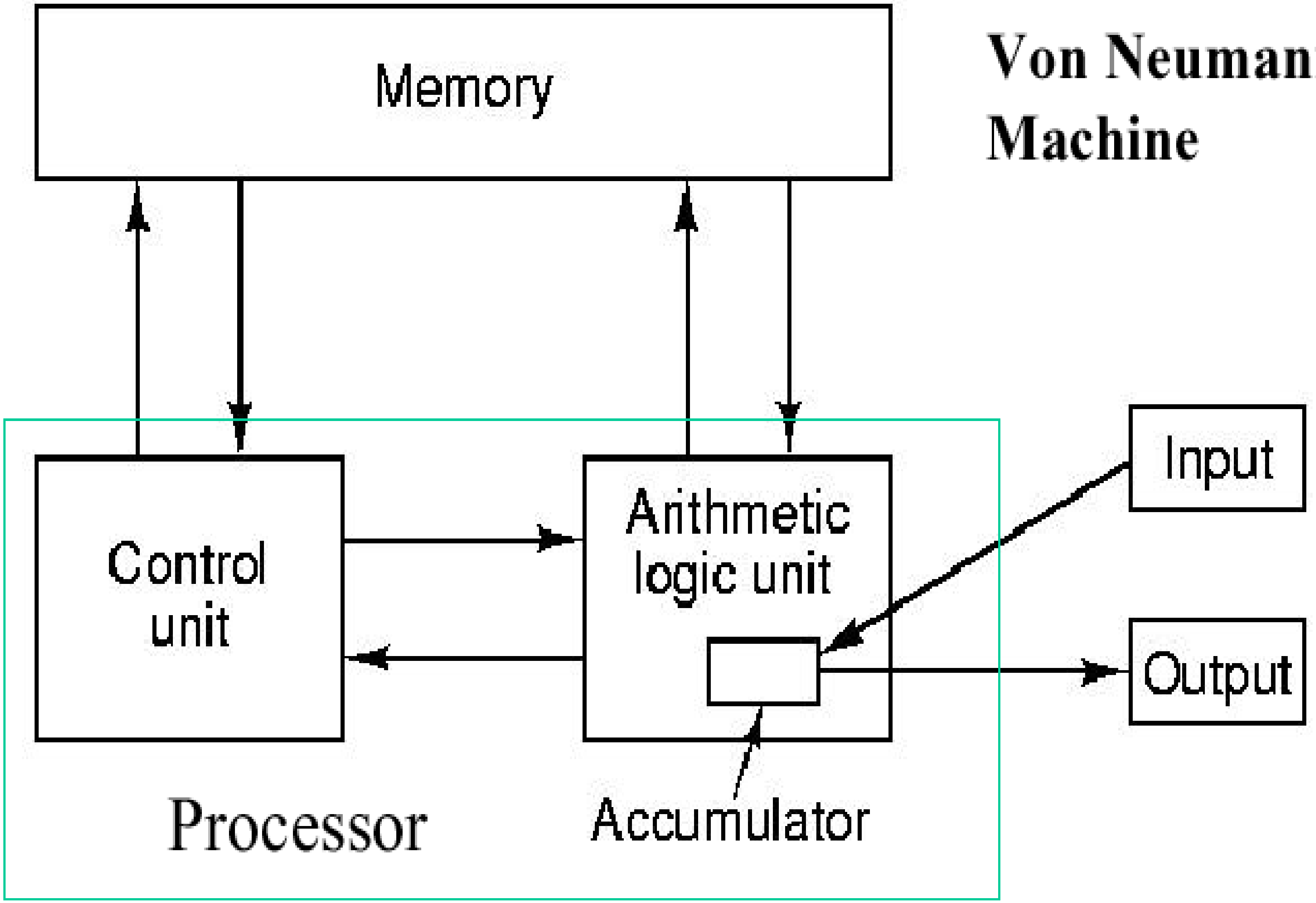
nn memory

2020/6/27 14

Von-Neumann Basic Structure:



Von Neumann Machine



operand of an instruction.

.4.1 Detail structure of IAS/Von-Neumann Structure

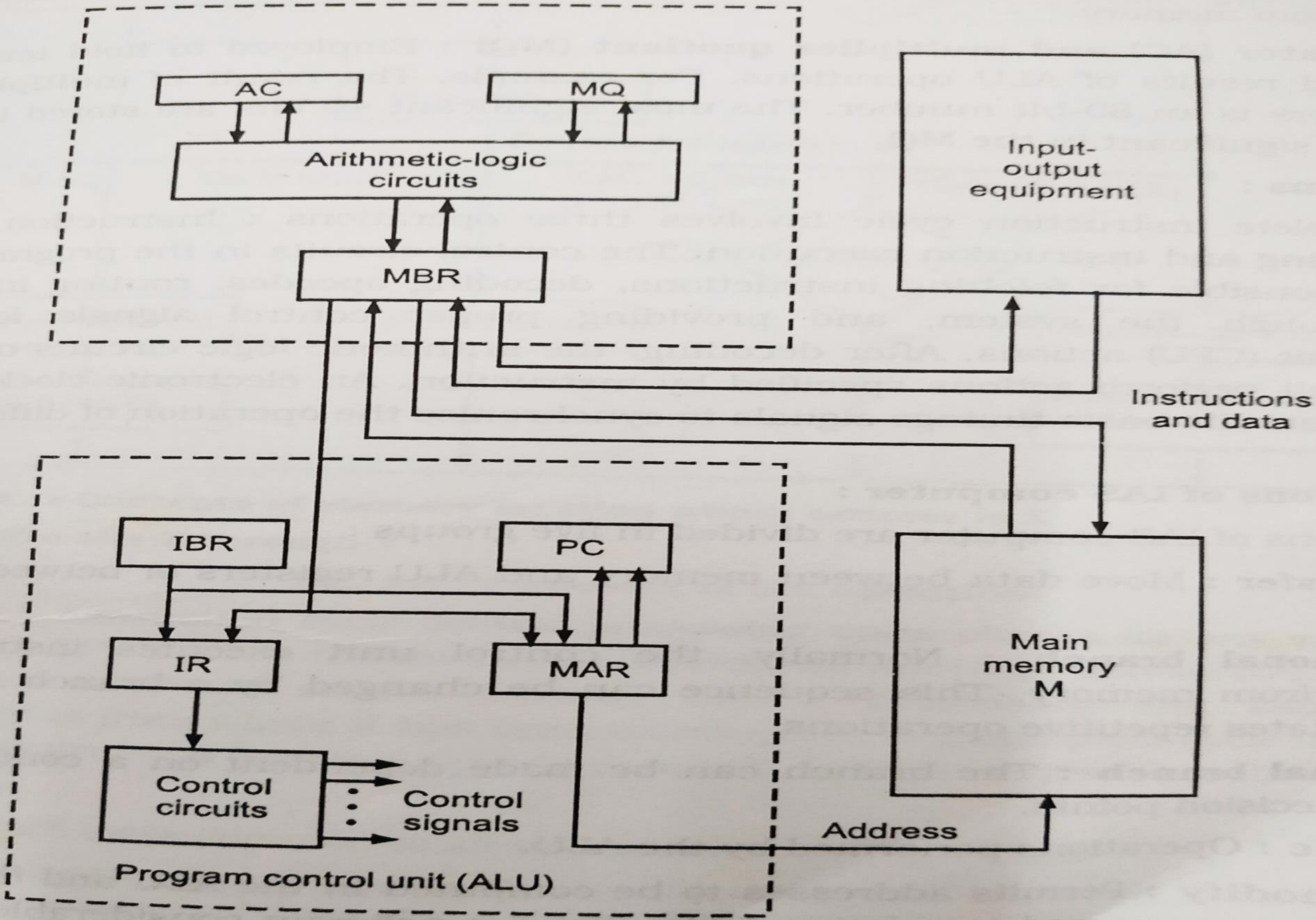


Fig. 1.4 : Detail Structure of IAS computer

2020/6/27 14:56

Central Processing Unit:

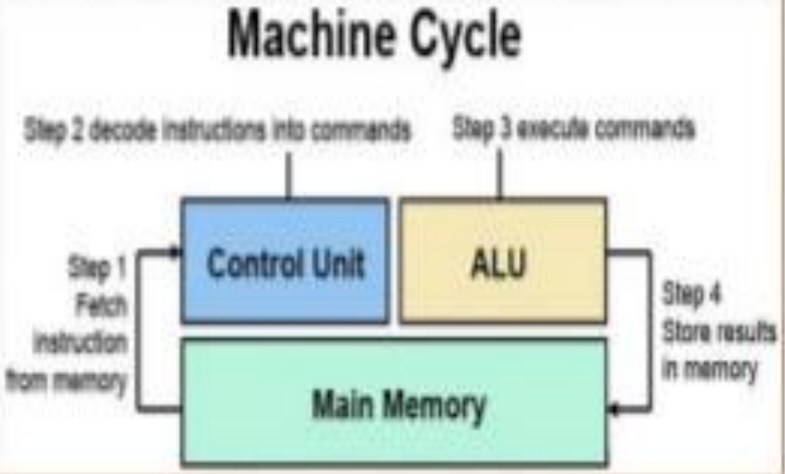
- CPU is the abbreviation for *central processing unit*. Sometimes referred to simply as the *central processor*, but more commonly called *processor*.
- CPU is the brains of the computer where all computations take place. In terms of computing power, the CPU is the most important element of a computer system.
- It is a set of electronic circuitry that executes the stored program instructions.
- The four primary functions of a processor are fetch, decode, execute and write back.
- Components of CPU are ALU (Airthematic and Logic Unit) and CU (Control Unit).

Airthematic and Logic Unit:

- ALU is a digital circuit used to perform arithmetic and logic operations.
- Most of the operations of a CPU are performed by one or more ALUs, which load data from input registers.
- ALU perform basic arithmetic and logic operations. Addition, subtraction, multiplication and division come under Airthematic operations while Logical AND, OR and XOR come under logical operations. Logic operations can be accomplished by connecting multiple transistors.

Control Unit

- This section is the boss of the CPU and coordinates all activity within the CPU.
- It act as a traffic signal directing the flow of data through the CPU as well as to and from other devices.
- The control unit controls the computer by repeating four operations called machine cycle. These four operations are;
  - Fetching program instructions from memory.
  - Decoding the instructions into commands.
  - Executing the commands.
  - Storing the results in the memory.

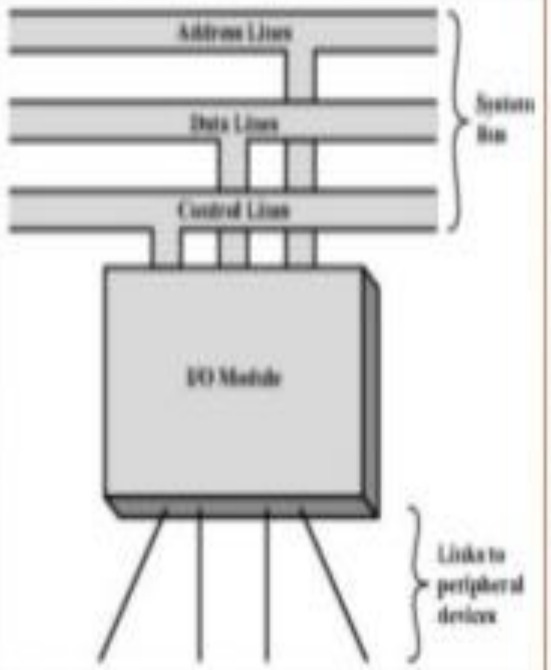

Machine Cycle
Step 2 decode instructions into commands
Step 3 execute commands
Step 1 Fetch instruction from memory
Control Unit
ALU
Step 4 Store results in memory
Main Memory

Main Memory

- In Computing, Memory refers to a physical memory used to store programs or data on the temporary or permanent basis for use in a computer.
- Main memory is divided into two parts RAM and ROM
- The computer is able to change data that is in random access memory but it is a volatile memory and don't retains its contents when the power is lost.
- ROM contains pre-recorded data that can be read but not modified or deleted but it is a non-volatile memory and so retains its contents when the power is removed.

Input / Output Subsystem:

- This Architecture handles devices that allow the computer system to communicate and interact with the outside world.
- Inputs are the signals or data received by the system and outputs are the signals or data sent from it.
- I/O system includes two basic components.
  - I/O module is normally connected to the computer system.
  - I/O device is connected to I/O module of the computer called peripheral device.



Advantage and Disadvantage:

- Easy memory organization for the user
- Data from memory and from devices are accessed in the same way.
- It is better for desktop computers, laptops, workstations and high performance computers.
- The programs can be optimized in smaller size.
- Only handles one task at a time.
- Bottlenecking is an issue because it take more time to execute.

VON NEUMANN ARCHITECTURE
VERSUS
HARVARD ARCHITECTURE

It is a theoretical design based on the stored-program computer concept.	It is a modern computer architecture based on the Harvard Mark I relay-based computer model.
It uses same physical memory address for instructions and data.	It uses separate memory addresses for instructions and data.
Processor needs two clock cycles to execute an instruction.	Processor needs one cycle to complete an instruction.
Simpler control unit design and development of one is cheaper and faster.	Control unit for two buses is more complicated which adds to the development cost.
Data transfers and instruction fetches cannot be performed simultaneously.	Data transfers and instruction fetches can be performed at the same time.
Used in personal computers, laptops, and workstations.	Used in microcontrollers and signal processing.

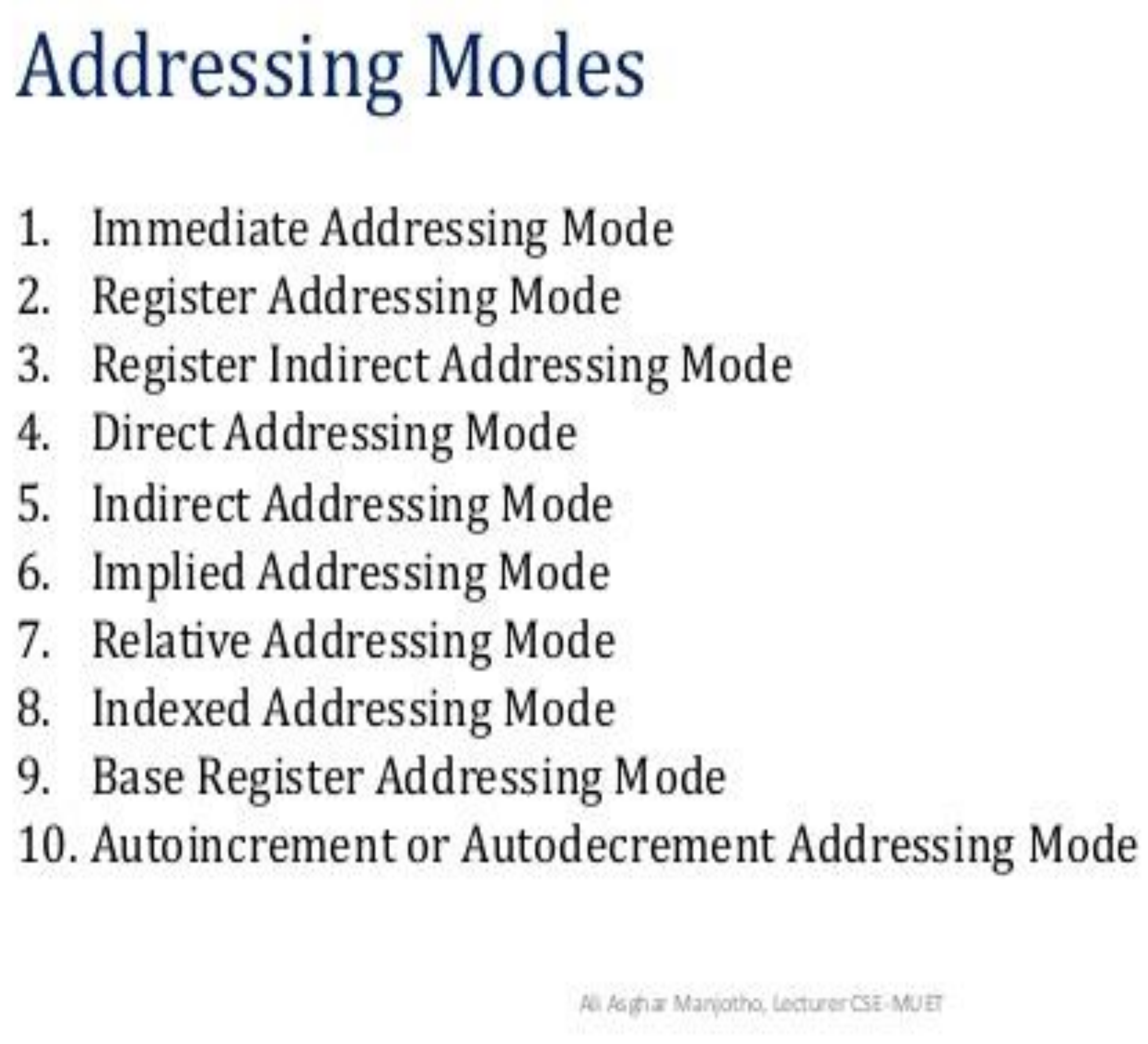
Question

Q1. List out and explain addressing modes in detail?

Q2 difference between

- Direct and indirect
- Direct and immediate
- Indirect and immediate
- Direct and implicit
- Indirect and implicit
- Memory and relative register

Addressing Modes



Addressing Modes

1. Immediate Addressing Mode
2. Register Addressing Mode
3. Register Indirect Addressing Mode
4. Direct Addressing Mode
5. Indirect Addressing Mode
6. Implied Addressing Mode
7. Relative Addressing Mode
8. Indexed Addressing Mode
9. Base Register Addressing Mode
10. Autoincrement or Autodecrement Addressing Mode

1. Immediate Addressing Mode

- The operand is specified with in the instruction.
- Operand itself is provided in the instruction rather than its address.

Move Immediate

MVI A , 15h   A ← 15h   Here 15h is the immediate operand

Add Immediate

ADI 3Eh   A ← A + 3Eh   Here 3Eh is the immediate operand

2. Register Addressing Mode

- The operand is specified with in one of the processor register.
- Instruction specifies the register in which the operand is stored.

	Move	
MOV C , A	$C \leftarrow A$	Here A is the operand specified in register

	Add	
ADD B	$A \leftarrow A + B$	Here B is the operand specified in register

3. Register Indirect Addressing Mode

- The instruction specifies the register in which the memory address of operand is placed.
- It do not specify the operand itself but its location with in the memory where operand is placed.

Move

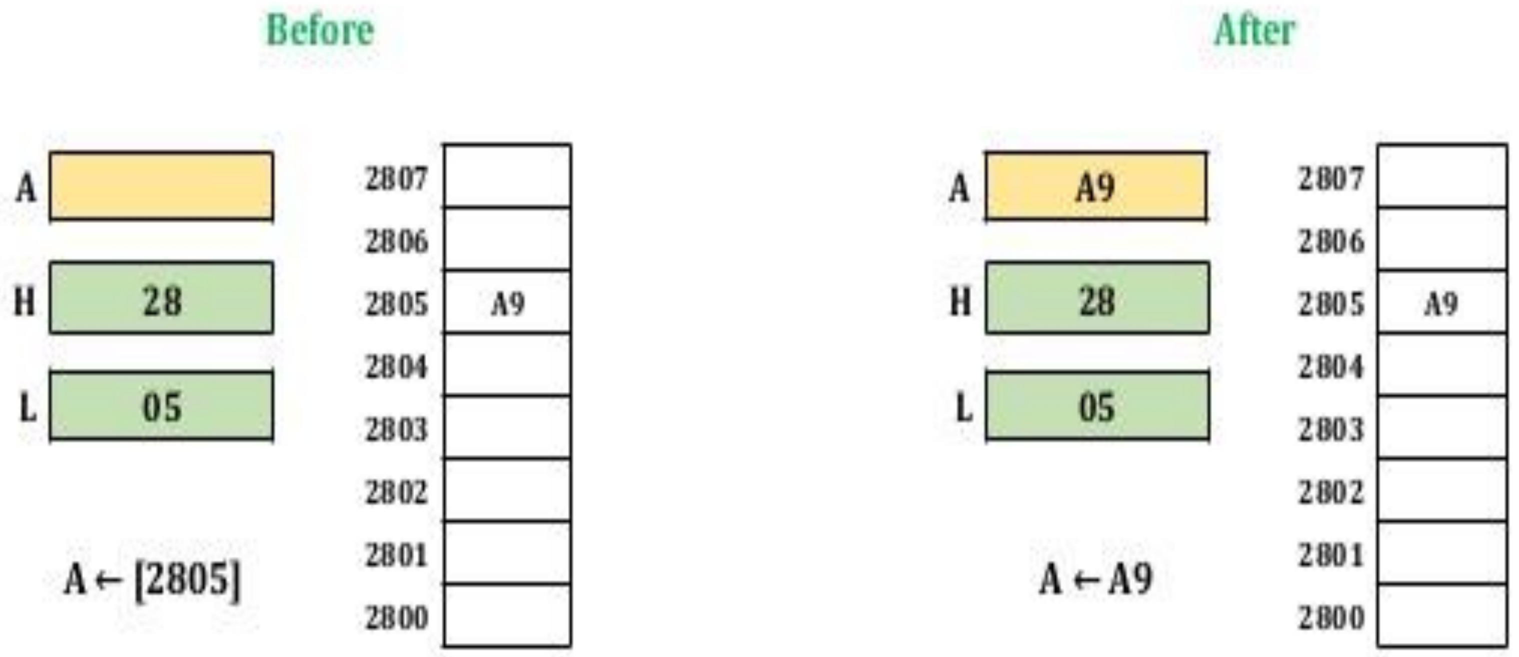
MOV A , M $A \leftarrow [[H][L]]$

It moves the data from memory location specified by HL register pair to A.

3. Register Indirect Addressing Mode

MOV A , M $A \leftarrow [[H][L]]$

It moves the data from memory location specified by HL register pair to A.



4. Direct Addressing Mode

- The instruction specifies the direct address of the operand.
- The memory address is specified where the actual operand is.

Load Accumulator

LDA 2805h   A ← [2805]

It loads the data from memory location 2805 to A.

Store Accumulator

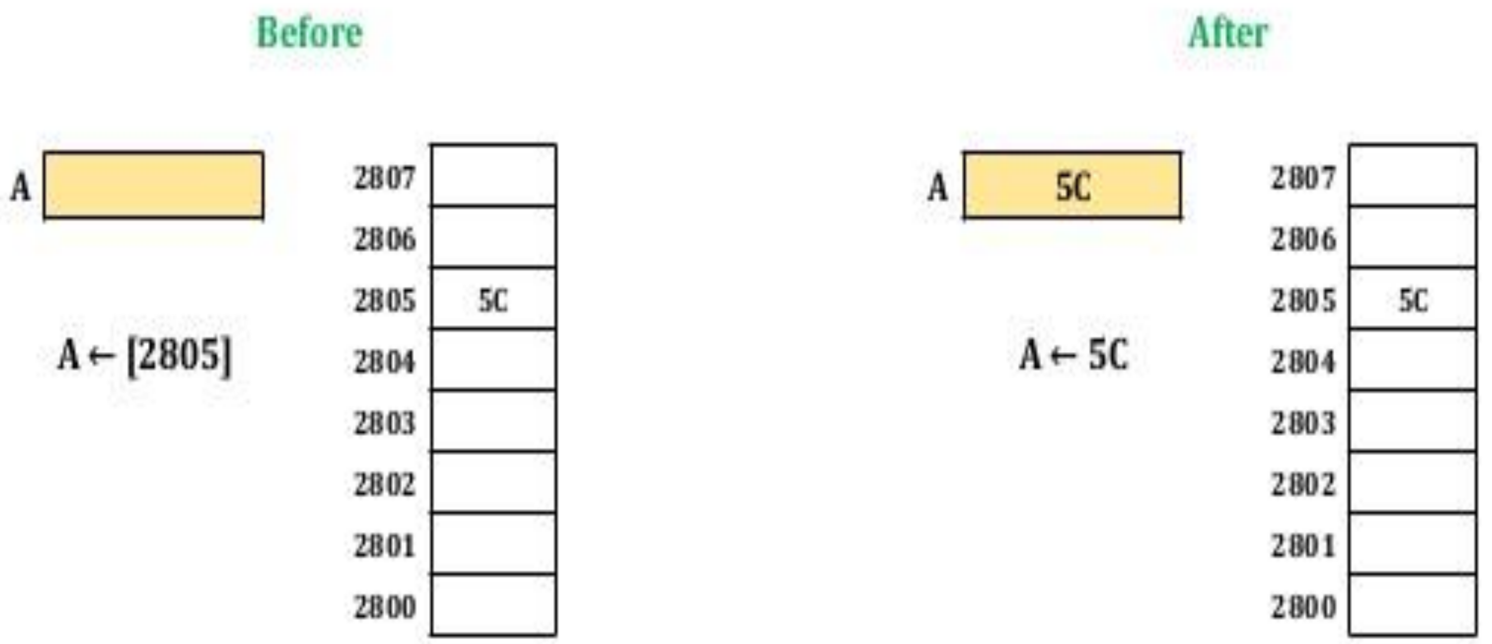
STA 2803h   [2803] ← A

It stores the data from A to memory location 2803.

4. Direct Addressing Mode

LDA 2805h A ← [2805]

It loads the data from memory location 2805 to A.

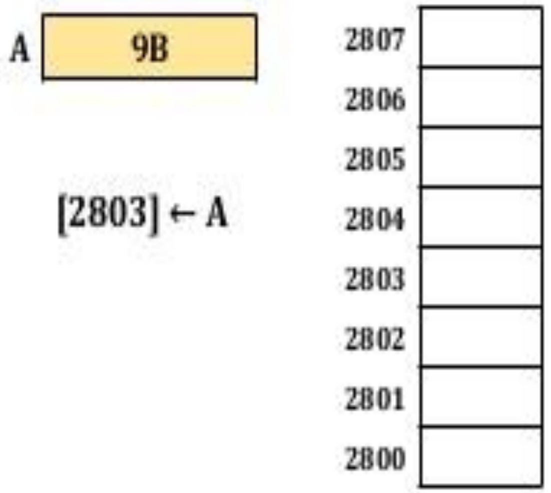


4. Direct Addressing Mode

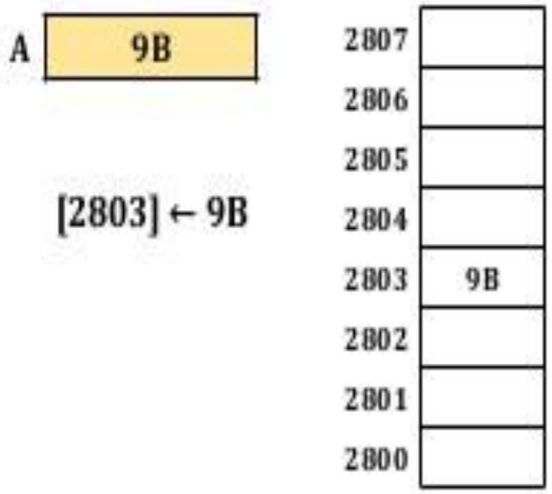
STA 2803h   [2803] ← A

It stores the data from A to memory location 2803.

Before



After



5. Indirect Addressing Mode

- The instruction specifies the indirect address where the effective address of the operand is placed.
- The memory address is specified where the actual address of operand is placed.

Move

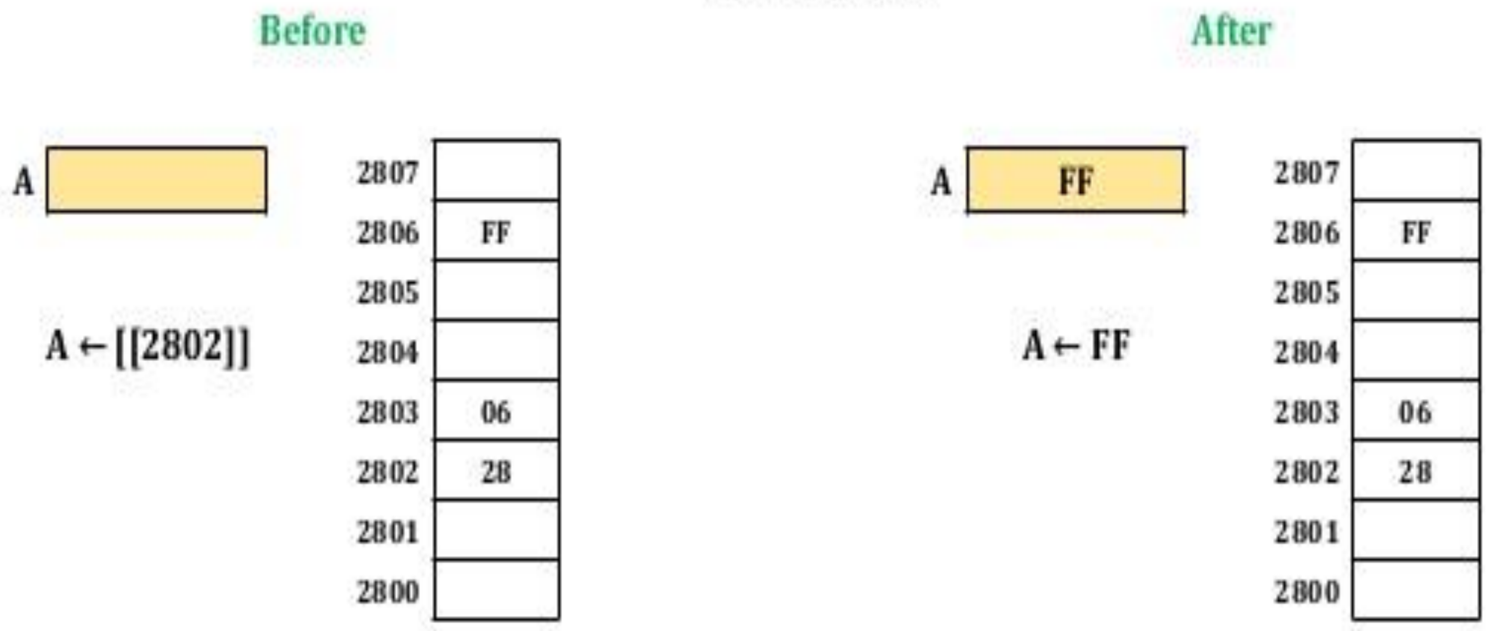
MOV A, 2802h $A \leftarrow [[2802]]$

It moves the data from memory location specified by the location 2802 to A.

5. Indirect Addressing Mode

MOV A, 2802h A ← [[2802]]

It moves the data from memory location specified by the location 2802 to A.



6. Implied Addressing Mode

- It is also called inherent addressing mode.
- The operand is implied by the instruction.
- The operand is hidden/fixed inside the instruction.

Complement Accumulator CMA
(Here accumulator A is implied by the instruction)

Complement Carry Flag CMC
(Here Flags register is implied by the instruction)

Set Carry Flag STC
(Here Flags register is implied by the instruction)

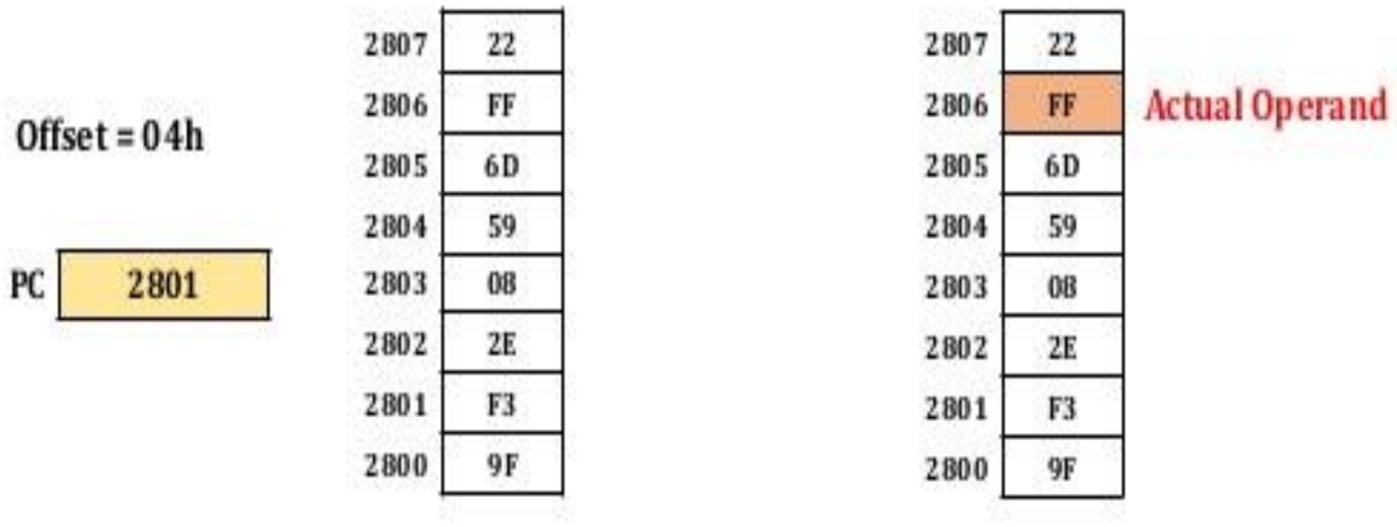
7. Relative Addressing Mode

- In relative addressing mode, contents of Program Counter PC is added to address part of instruction to obtain effective address.

- The address part of the instruction is called as offset and it can +ve or –ve.

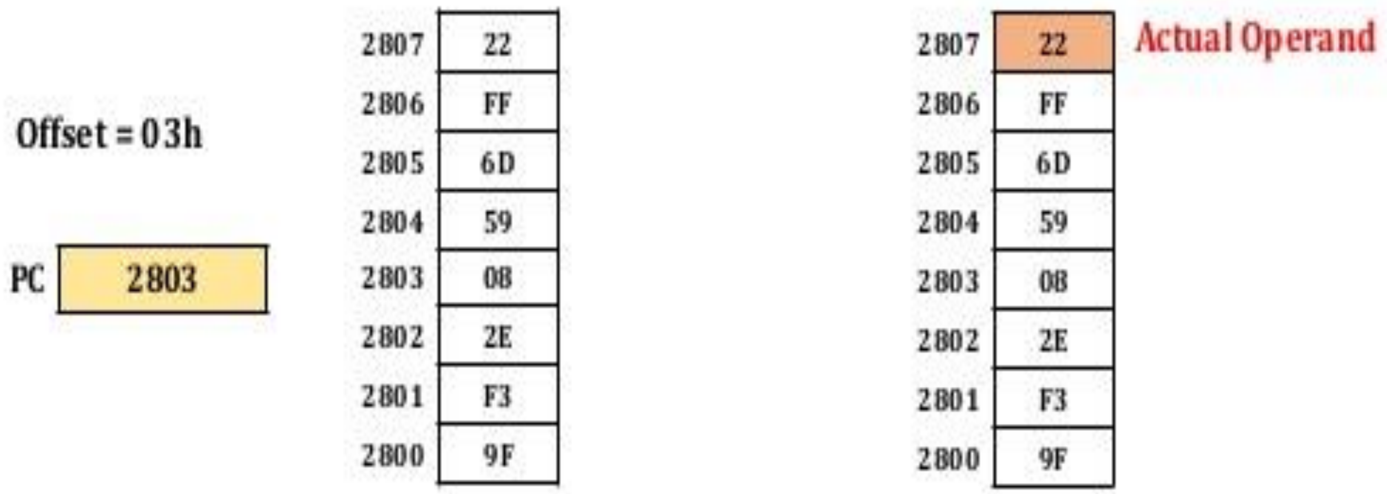
- When the offset is added to the PC the resultant number is the memory location where the operand will be placed.

7. Relative Addressing Mode



Effective address of operand = **PC + 01 + offset**
Effective address of operand = **2801 + 01 + 04**
Effective address of operand = **2806h**

7. Relative Addressing Mode



Effective address of operand = **PC + 01 + offset**
Effective address of operand = **2803 + 01 + 03**
Effective address of operand = **2807h**

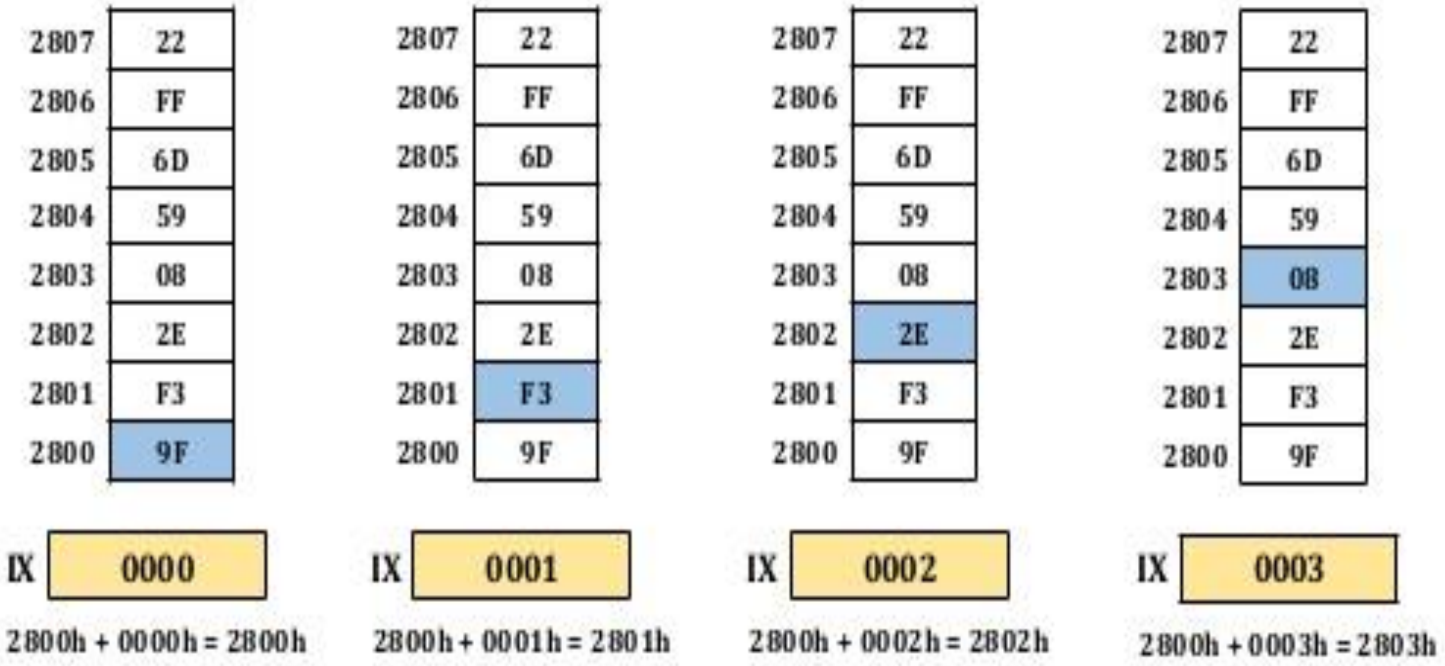
8. Indexed Addressing Mode

- In index addressing mode, contents of Index register is added to address part of instruction to obtain effective address.
- The address part of instruction holds the beginning/base address and is called as base.
- The index register hold the index value, which is +ve.
- Base remains same, the index changes.
- When the base is added to the index register the resultant number is the memory location where the operand will be placed.

8. Indexed Addressing Mode

Base = 2800h

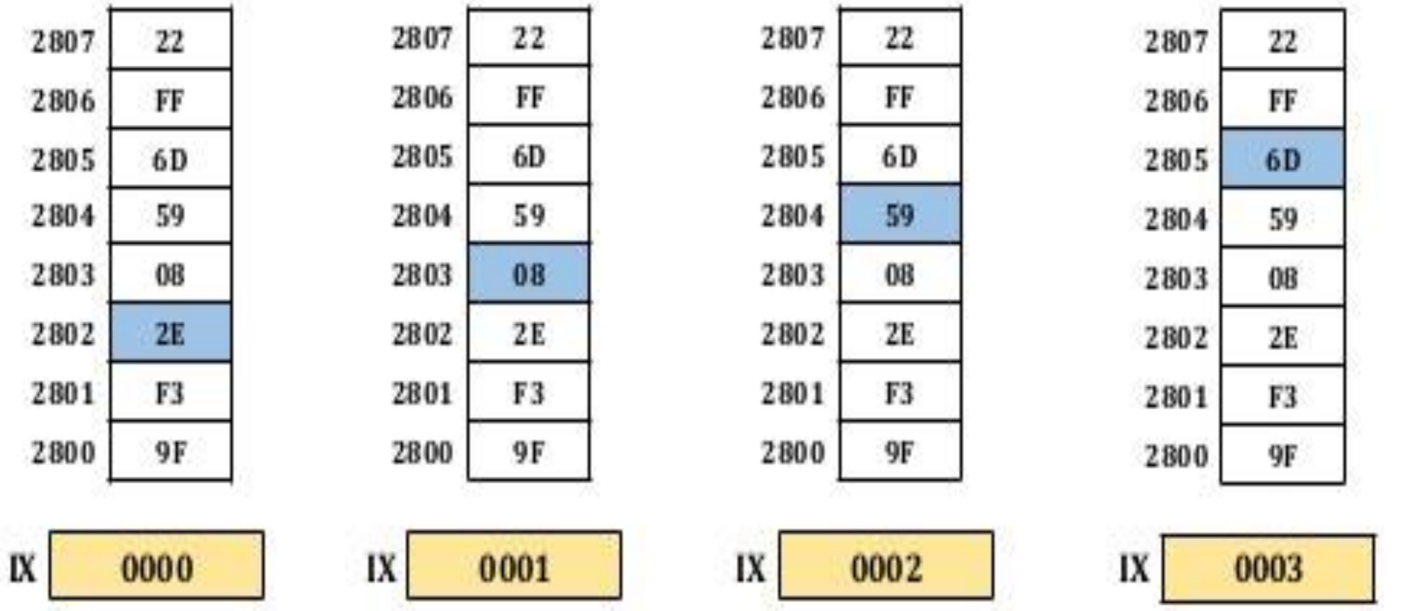
Effective address of operand = **Base + IX**



8. Indexed Addressing Mode

Base = 2802h

Effective address of operand = **Base + IX**



2802h + 0000h = 2802h

2802h + 0001h = 2803h

2802h + 0002h = 2804h

2802h + 0003h = 2805h

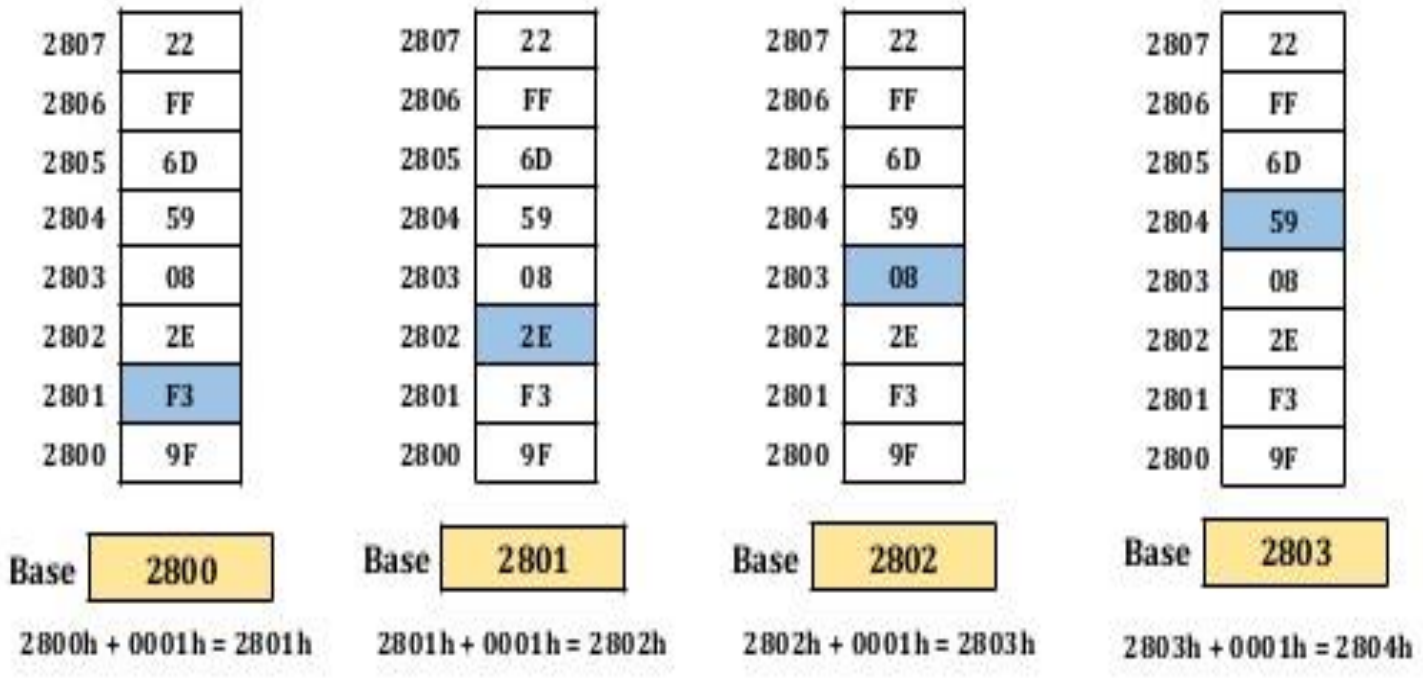
9. Base Register Addressing Mode

- In base register addressing mode, contents of base register is added to address part of instruction to obtain effective address.
- It is similar to the indexed addressing mode except the register now is called as base instead of index.
- The base register hold the beginning/base address.
- The address part of instruction holds the offset.
- Offset remains same, the base changes.
- When the offset is added to the base register the resultant number is the memory location where the operand will be placed.

9. Base Register Addressing Mode

Offset= 0001h

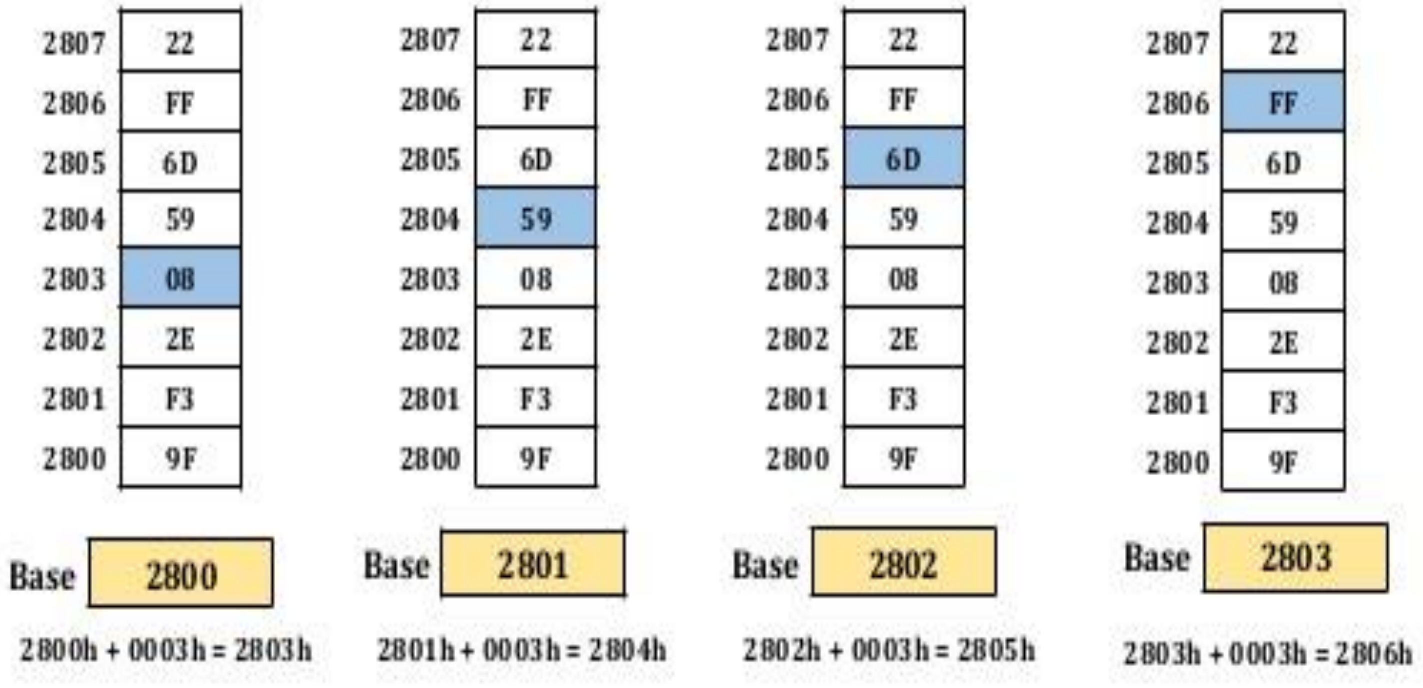
Effective address of operand = **Base Register + offset**



9. Base Register Addressing Mode

Offset= 0003h

Effective address of operand = **Base Register + offset**

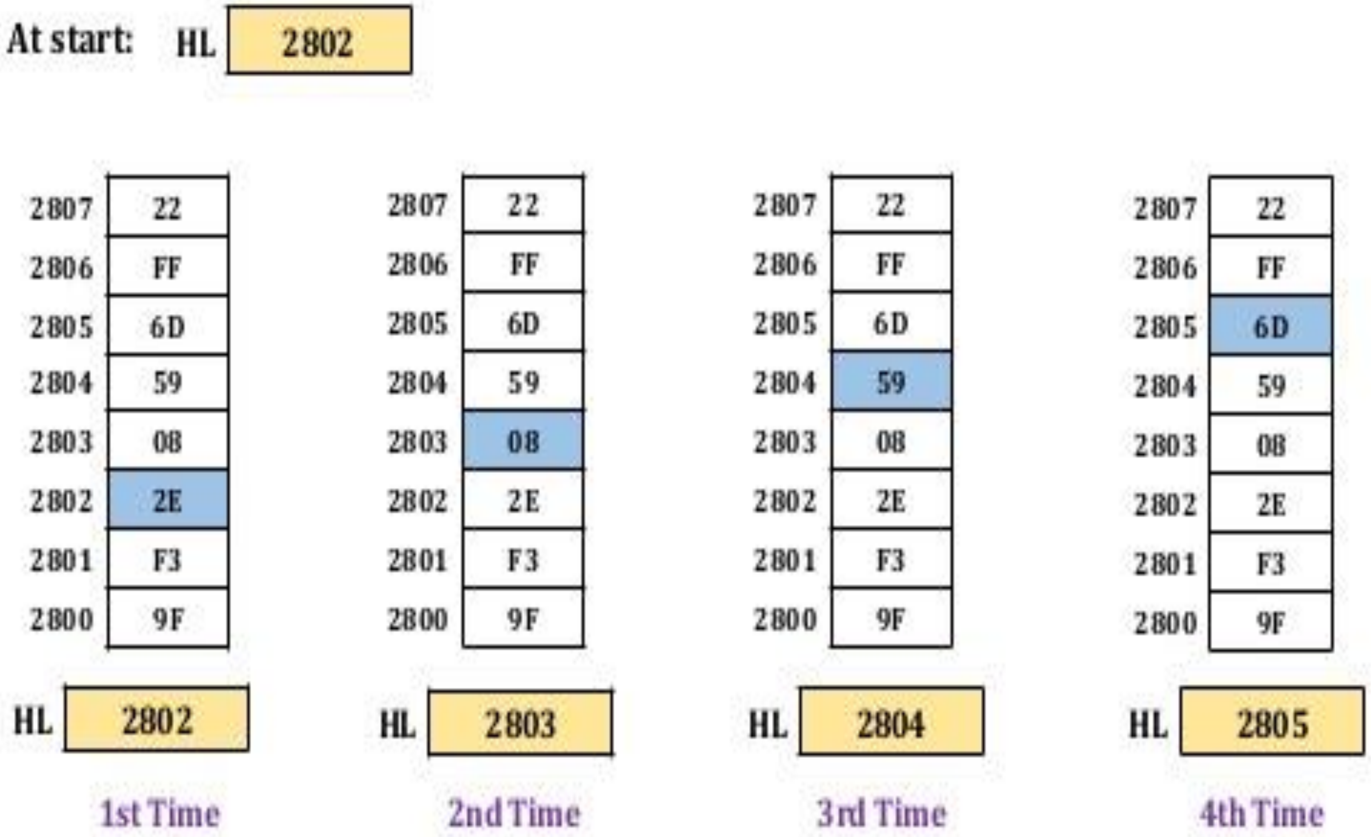


10. Autoincrement or Autodecrement Addressing Mode

- It is similar to register indirect addressing mode.
- Here the register is incremented or decremented before or after its value is used.

10. Autoincrement or Autodecrement Addressing Mode

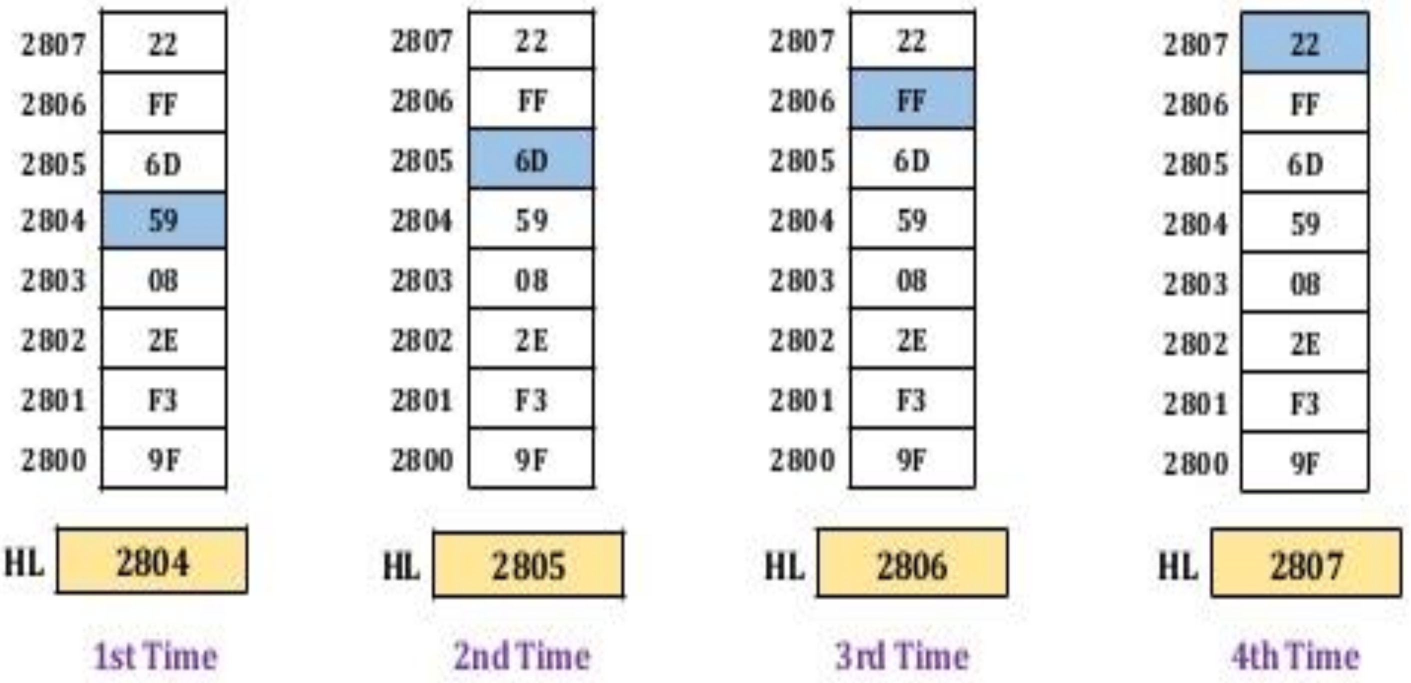
HL pair incremented after its value is used



10. Autoincrement or Autodecrement Addressing Mode

HL pair incremented before its value is used

At start: HL 2803



Difference between Direct and Indirect Addressing Modes

Direct Addressing Mode	Indirect Addressing Mode
Address field contains the effective address of operand	Address field contains reference of effective address
Requires only one memory reference	Requires two memory references
Fast addressing	Slower than direct addressing mode
No further classification	Further classified into two categories
No further calculation is required to perform the operation	Require further calculation to find the effective address

compare Direct and Immediate Addressing Modes.

Indirect Addressing Mode	**Immediate Addressing Mode**
The address field of the instruction holds the address of the operand.	**There is no address field as the operand is a part of the instruction.**
It requires two reference to memory.	**It does not require any reference to memory.**
It is slower compared to immediate mode.	**It is a faster process.**
It has more range than in immediate mode.	**It has a limited range.**
It is further classified into two categories.	**No further classification.**
Example: ADD [B]	**Example: ADD 5**

Difference between Indirect and Implied Addressing Modes

Indirect Addressing Mode	Implied Addressing Mode
Multiple memory spaces are used.	No memory Intervension
Operands are explicit	Operands are implicit
Mostly used in 2 address instructions and more	Mostly used in zero address and single address instructions
3 memory references are required	No memory references are required
Address space is large	Address space is small
Additional calculations are the only way to perform operation	No additional calculations are required
Execution speed is less	Execution speed is more

Difference between Direct and Implied Addressing Modes

DIRECT ADDRESSING MODE	IMPLIED ADDRESSING MODE
Address fields contains the effective address of operand.	Effective address of operand is specified implicitly.
Instruction size is larger since operand has to be explicitly specified.	Instruction size is smaller since operand is specified implicitly.
It requires one reference to memory.	No memory references are required.
Mostly used in 2 address instructions and more.	Mostly used in zero address and single address instructions.
It is slower compared to implied mode.	It is a faster method.
It has more range than implied mode.	It has less range than direct mode.

Difference between Indirect Addressing Mode and Immediate Addressing

Indirect Addressing Mode	**Immediate Addressing Mode**
The address field of the instruction holds the address of the operand.	**There is no address field as the operand is a part of the instruction.**
It requires two reference to memory.	**It does not require any reference to memory.**
It is slower compared to immediate mode.	**It is a faster process.**
It has more range than in immediate mode.	**It has a limited range.**
It is further classified into two categories.	**No further classification.**
Example: ADD [B]	**Example: ADD 5**

Difference between Implied Addressing Mode and Immediate Addressing Mode

Implied Addressing Mode	Immediate Addressing Mode
In Implied addressing mode, no operand is specified in the instruction .	In Immediate addressing mode, operand is specified in the instruction itself .
Basically, the operands are specified implicitly in the definition of instruction .	Here, the operands are contained in an operand field rather than address field .
This type of mode can be used in all register reference instructions .	This type of mode is quite useful for initializing the registers to a constant value .
It requires 8 bits or 16 bits long data and is the part of instruction.	It requires more bits than the address.
There is no need to acquire a operand .	It is fast in acquiring an operand .
Zero-address instructions in a stack-organized computer are implied-mode instructions .	The address field of an instruction may specify either a memory word or a processor register.
Example: CMA (Complement Accumulator)	Example: MVI A 45

compare Direct and Immediate addressing modes.

Direct Addressing Mode	**Immediate Addressing Mode**
Address fields contains the effective address of operand	**There is no address field as the operand is a part of the instruction.**
It requires one reference to memory.	**It does not require any reference to memory.**
It is slower compared to immediate mode.	**It is a faster process.**
It has more range than in immediate mode.	**It has a limited range.**
Example: Add (1001)	**Example: ADD 5**

Difference between PC Relative And Base Register Addressing modes:

PC Relative Addressing Mode	Base Register Addressing Mode
The content of program counter is added to the addressing field of the instruction i to obtain the effective address.	The base register content is added to the addressing field of the instruction to obtain the effective address.
The addressing field of the instruction is mostly a signed number which can be either positive or negative.	A base register holds a base address and the addressing field of the instruction gives displacement according to the base address.
A program counter always keeps track of the instructions of the program stored in its memory.	A particular register has to be selected from the register set, according to the instruction.
Uses more bits as it has to specify a memory address directly.	Uses less bits as it has to select a register from a register set.
A program counter always contains the address of the immediately next instruction to be executed. After fetching the address mentioned in the instruction, the program counter value immediately increases.	In Base Register addressing mode the displacement value can be the same as the value required to reference the desired address as it does not immediately go to the next instruction.
Effective address of the operand is obtained by adding the program counter content to the addressing field of instruction.	Effective address of the operand is obtained by adding the base register content to the addressing field of instruction.

EA = PC + Address field value
PC = PC + Relative value
	EA = Base register + Address field value
	PC = Base register + Relative value

memory Based Addressing Modes and Register Based Addressing Modes

memory Based Addressing Modes	Register Based Addressing Modes
The operand is present in memory and its address is given in the instruction itself. This addressing mode is taking proper advantage of memory address, e.g., Direct addressing mode	An operand will be given in one of the register and register number will be provided in the instruction.With the register number present in instruction, operand is fetched, e.g., Register mode
The memory address specified in instruction may give the address where the effective address is stored in the memory. In this case effective memory address is present in the memory address which is specified in the instruction, e.g., Indirect Addressing Mode	The register contains the address of the operand. The effective address can be derived from the content of the register specified in the instruction. The content of the register might not be the effective address. This mode takes full advantage of registers, e.g., Register indirect mode
The content of base register is added to the address part of the instruction to obtain the effective address. A base register is assumed to hold a base address and the address field of the instruction gives displacement relative to the base address, e.g., Base Register Addressing Mode	If we are having a table of data and our program needs to access all the values one by one we need something which decrements the program counter/or any register which has base address. Though in this case register is basically decreased, it is register based addressing mode, e.g., In Auto decrements mode
The content of the index register is added to the address part that is given in the instruction to obtain the effective address. Index Mode is used to access an array whose elements are in successive memory locations, e.g., Indexed Addressing Mode	If we are having a table of data and our program needs to access all the values one by one we need something which increment the program counter/or any register which has base address, e.g., Auto increment mode
The content of program counter is added to the address part of the instruction in order to obtain the effective address. The address part of the instruction in this case is usually a signed number which can be either positive or negative, e.g., Relative addressing mode	Instructions generally used for initializing registers to a constant value is register based addressing mode,and this technique is very useful approach, e.g., Immediate mode.