ECT282

MICROCONTROLLERS

Minor Course

Module 1 - Syllabus

- Functional units of a computer, Von Neuman and Harvard computer architecture. Processor Architecture –General internal architecture, Address bus, Data bus, control bus. Register set – status register, accumulator, program counter, stack pointer, general purpose registers. Processor operation – instruction cycle, Instruction fetch, instruction decode, instruction execute.

Course Outcome

- After completion of the module the student will be able to
  - Explain the building blocks of a typical microcomputer/microcontroller system

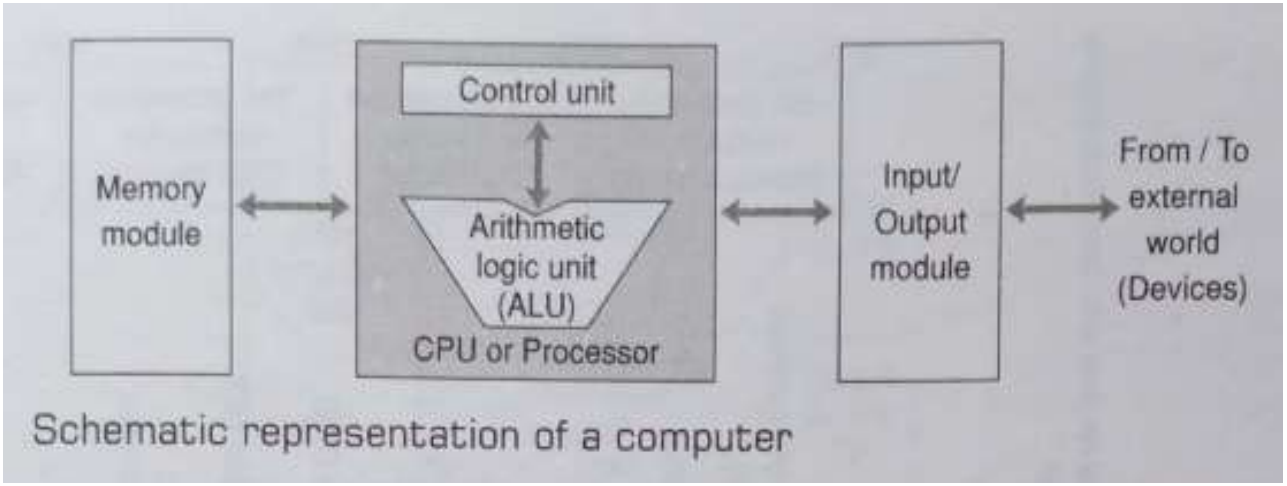
Computer, Computer Architecture, Computer Organization

- Computer : A computer may be taken as an electronic instrument capable of very high speed numerical calculations and carry out some specific control operations, completely depending on the software, which it would be executing.
- Computer Architecture : How to integrate the components to build a computer system to achieve a desired level of performance. It deals with the overall design of the computer.

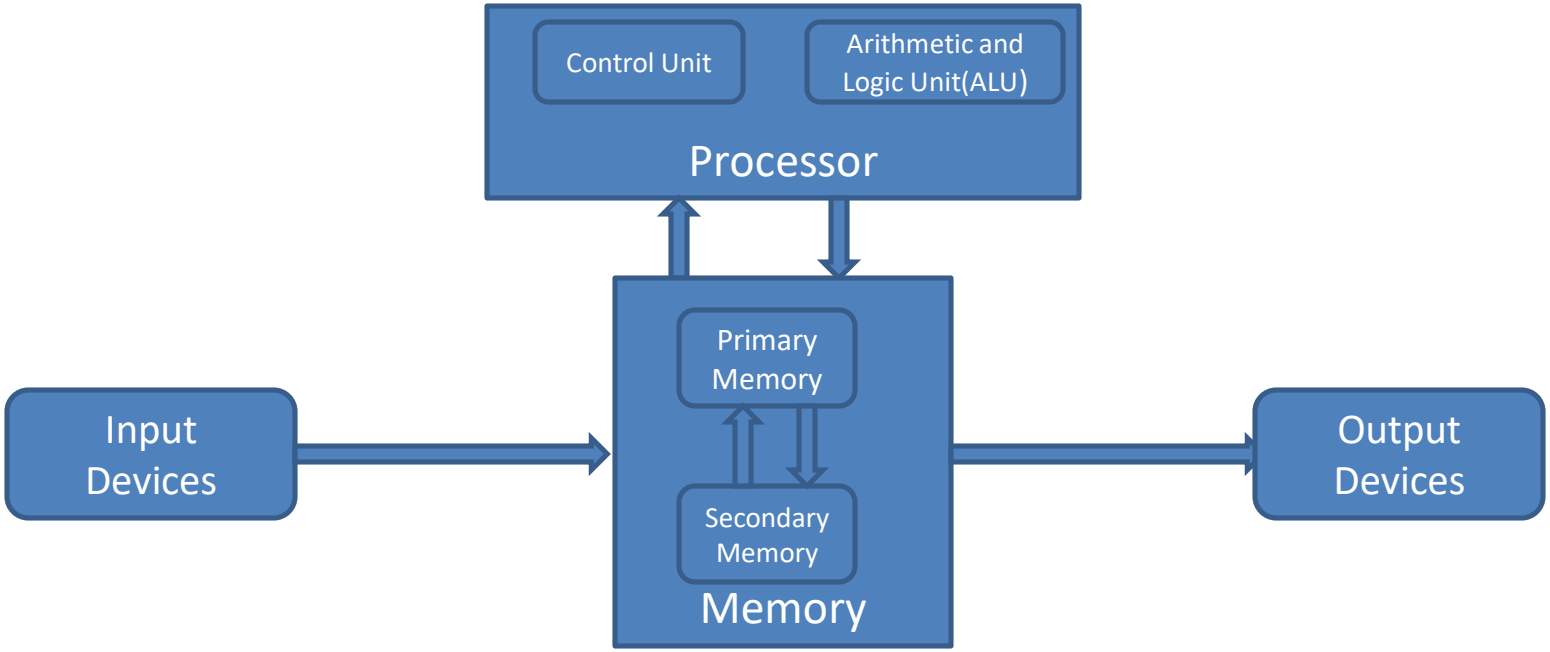
  - Analogy: Architect's task during the planning of a building (overall layout, floor plan etc.)
- Computer Organization : Design of the components and functional blocks using which computer systems are built. It deals with the electronic implementation of the architectural modules.

  - Analogy: Civil engineers task during building construction (How much percentage of cement, bricks, iron rods etc.)

Functional Units of a Computer



Schematic representation of a computer



Basic Computer Organization

- Main parts of a computer system:
  - Main Memory
  - I/O devices
  - Processor (Also called as CPU)
- These are the essential architectural modules of any computer in general.
  - All instructions and data are stored in memory
  - An instruction and the required data are brought into the processor for execution.
  - Input and output devices interface with the outside world.

1. Processor…..

- The processor fetches an instruction from memory for execution
  - An instruction specifies the exact operation to be carried out
  - It also specifies the data that are to be operated on.
  - A program refers to a set of instructions that are required to carry out some specific task(eg: sorting a set of numbers)

Inside the Processor…..

- Consists of a Control Unit and an Arithmetic Logic Unit(ALU)
  - All calculations happen inside the ALU.
  - The control unit generates sequence of control signals to carry out all operations

- Control Unit : To manage the instruction execution
- ALU : Arithmetic and Logical Unit (hardware to do arithmetic, logical operations)
- Registers: small units of memory to hold data/instructions temporarily during execution

What is the role of ALU?

- It contains circuitry to carry out logic operations like AND,OR,NOT, shift, compare, etc.
- It contains circuitry to carry out arithmetic operations like addition, subtraction, multiplication, division, etc.
- During instruction execution, the data(operands) are brought in and stored in some registers, the desired operation carried out, and the result stored back in some register or memory.

What is the role of Control unit

- Control unit is used to coordinate the operations of the input, output, memory, ALU in some way. It coordinates the operation of all the units using control signals like timing signals.
- Timing signals are the signals determining the time when a given action must take place. A large set of control lines carriers the signals used for timing and synchronization of events in all units.

2. Memory

- **What is a memory?**
  - Something that can remember things
- There are different kinds of memory in a computer system
  - Some remember by the state an electrical circuit is in. eg: SRAM
  - Some remember by the amount of electrical charge stored in a capacitor. eg: DRAM
  - Yet others remember by magnetic or optical properties. eg: hard disk / magnetic Tape, CD/DVD
- They can vary substantially in their speed and capacity

- **Two main types of memory**
  - Primary or main memory, which stores the active instructions and data for the program being executed on the processor.
    - Eg: ROM, RAM (SRAM, DRAM)
  - Secondary memory, which is used as a back up and stores all active and inactive programs and data, typically as files.
    - Eg: hard disk , magnetic Tape, CD/DVD, Flash Memory
- The processor only has direct access to the primary memory.
- In reality, the memory system is implemented as a hierarchy of several levels
  - Cache, Primary memory, Secondary memory
  - Objective is to provide faster memory access at affordable levels

- **Primary or main memory:** stores the active instructions and data for the program being executed on the processor.

- Read Only Memory(ROM), which is used as part of the primary memory to store some fixed data that cannot be changed.
- Random Access Memory(RAM), which is used for the cache and primary memory. Read and write access times are independent of the location being processed.

- **Secondary storage**: Secondary memory are used when large amount of data and programs have to be stored.
- Magnetic Disk, which uses direction of magnetization of tiny magnetic particles on a metallic surface to store data. Access times vary depending on the location being accessed, and is used as a secondary memory.
- Flash Memory, which is replacing magnetic disks as secondary memory devices. They are faster, but smaller in size as compared to disk.

Example:

- Magnetic disks,
- Magnetic tapes, and
- Optical disks.

Fig: Magnetic disk

Fig: Magnetic tapes

Fig: Optical disk

3. Input Unit

- Used to feed data to the computer system from the external environment
  - Data are transferred to the processor/memory after appropriate encoding
- Common input devices:
  - Keyboard
  - Mouse
  - Joystick
  - Camera

Some of the Input Units

Fig: joysticks

Fig: Trackballs

Fig: Mouse

Fig: Microphones

4. Output Unit

- Used to send the result of some computation to the outside world. They send the processed results to the user. Output devices are pieces of equipment that are used to get information or any other response out from computer.
- The most common output device is a monitor. These devices display information that has been held or generated within a computer.

- Common output devices:
  - LCD/LED screen
  - Printer
  - Speaker
  - Projection system

Some of the output units are

Fig: Printer

Fig: Graphics Displays

Fig: CRT Displays

Fig: Microfilm

Computer Architecture

(Von-Neumann *vs Harvard)*

- Computer Architecture refers to the internal design of a computer with its CPU, which includes:

  •Arithmetic and logic unit

  •Control unit

  •Registers

  •Memory for data and instructions

  •Input/output interface and

  •External storage functions

Computer Architecture

(Von-Neumann *vs Harvard)*

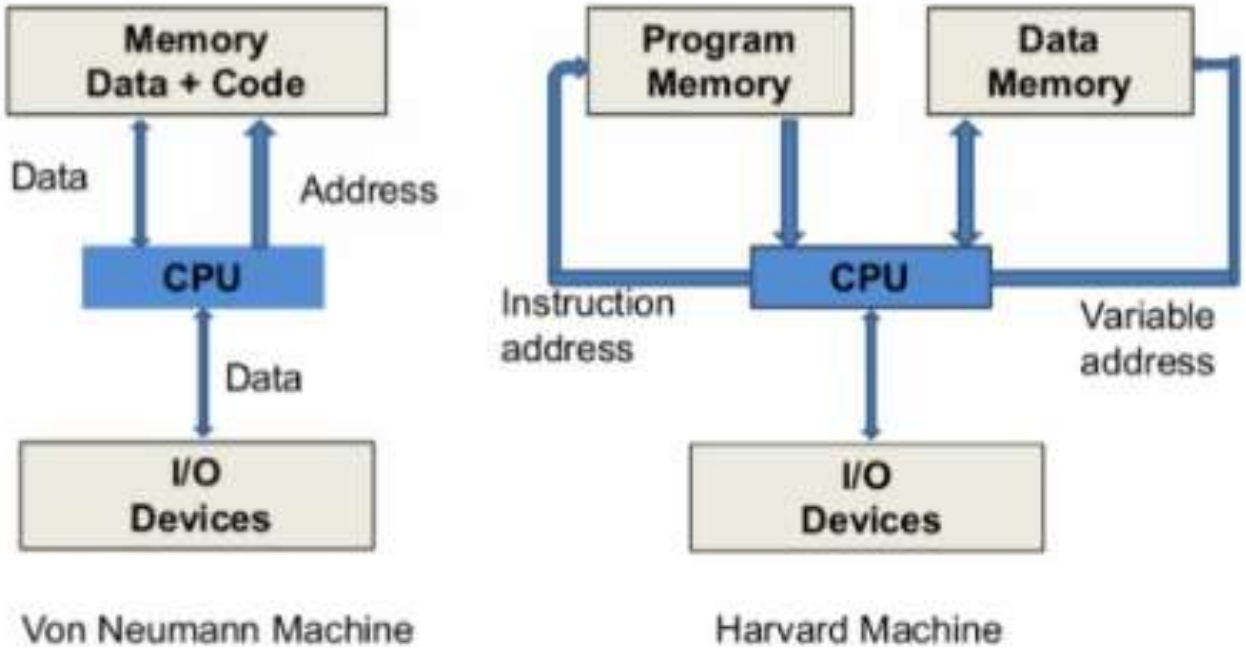
There are two computer architectures, which are *different in the way of accessing memories:* ***Von Neumann Architecture*** *(also names " Princeton Architecture") and* ***Harvard Architecture.***

• In a **Von Neumann architecture** , programs and data are stored in the same memory and managed by the same information-handling subsystem.

•In the **Harvard architecture** , programs and data are stored and handled by different subsystems. This is the essential difference between these two architectures.

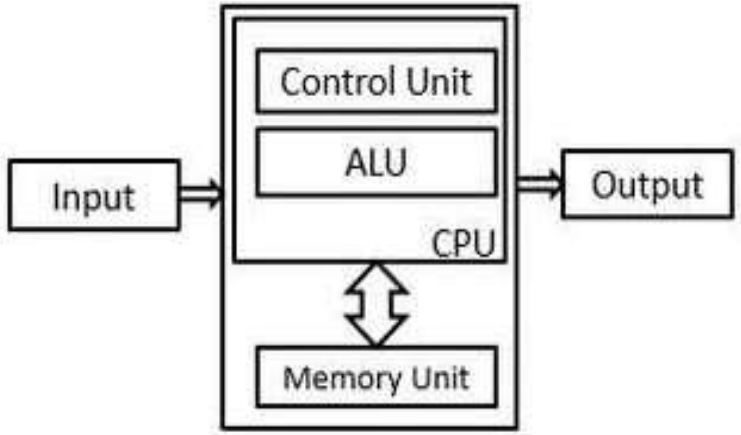
Von Neumaan and Harvard Computer Architectures

There are basically two types of architectures : **Von Neumann architecture** and **Harvard architecture**.



- **Von Neumann Architecture:**

In a Von-Neumann architecture, the **same memory** and **bus** are used to **store both data and instructions** that run the program. Since you cannot access program memory and data memory simultaneously, the Von Neumann architecture is susceptible to bottlenecks and system performance is affected.



Von Neumann Model

- **The Von Neumann Architecture has following specialties :**

1. Instructions and Data are stored in the same memory.
2. Instructions and Data share one memory system.
3. It has only one bus which is used for both data transfers and instruction fetches, and therefore data transfers and instruction fetches must be scheduled – they cannot be performed at the same time.

Von Neumann is better for Desktops, laptops,workstations, High end processing machines

- **The Harvard Architecture has following specialties:**

1. Physically separates storage and signal pathway for instructions and data.
2. It has separate data and instruction busses, allowing transfers simultaneously on both busses.
3. Generally, the bit of Instructions is wider than Data.
4. For some computers, the Instruction memory is read-only.
5. In cases without caches, the Harvard Architecture is more efficient than Von Neumann.

Harvard architecture is used primarily for embedded systems and signal processing(DSP)

Comparison between Von neumann and Harvard Architecture:

COMPARISON B/W HARVARD AND VON-NEUMANN ARCHITECTURE	
HARVARD ARCHITECTURE	**VON-NEUMANN/PRINCETON ARCHITECTURE**
This architecture consists of separate program and code memory.	It consists of common program and code memory.
Execution of instructions are faster, because fetching of code and data are separate.	Execution of instructions is relatively slow ,because not possible to fetch code and data simultaneously.
Execution of instruction takes less instruction (machine) cycle.	Execution of instruction takes more instruction (machine) cycle.
Use RISC processor	Use CISC processor
This is greater amount of parallelism	This is no need to have parallelism
Chip design is complex because of separate memory.	The largest advantages is that it simplifies the chip design because only one memory is accessed.

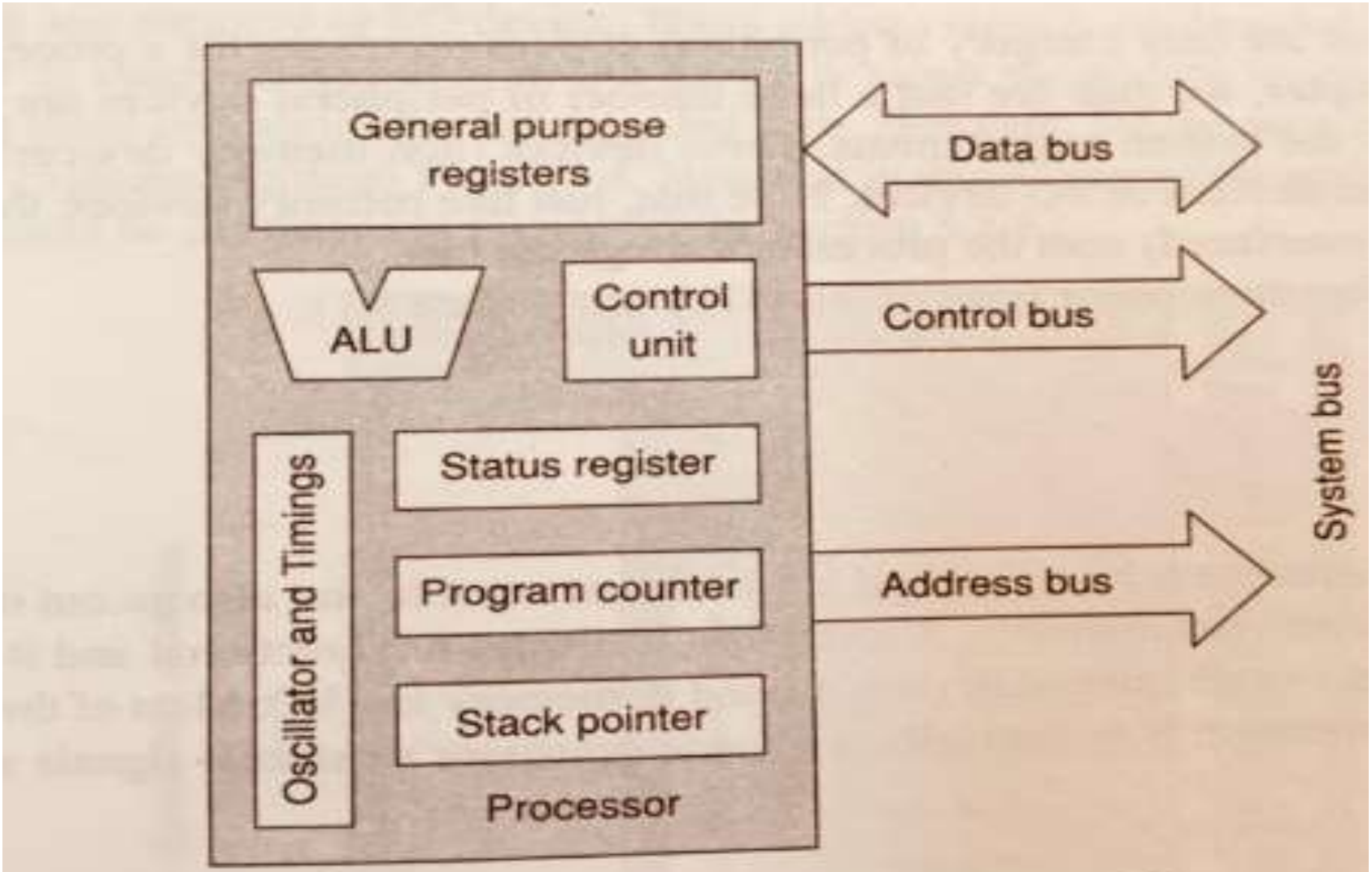
- **Storage**
  - The smallest form of storage in a computer is a bit, this is a Binary Digit, either 1 or 0. This table shows you how bytes are organized:

Bit	= binary digit: 0 or 1
1 byte	= 8 bits
1 KiloByte	= 1024 bytes
1 MegaByte	= 1024 KB
1 GigaByte	= 1024 MB
1 TeraByte	= 1024 GB
1 Petabyte	= 1024 TB

  - The above table can be also written as follows

Unit of Storage	Composed of	
1 bit	Can be 1 or 0	
1 byte	8 bits	
1 Kilobyte (KB)	2^{10} = 1024 bytes	1024 bytes
1 Megabyte (MB)	2^{20} = 1,048,576 bytes	1024 KB
1 Gigabyte (GB)	2^{30} = 1,073,741,824 bytes	1024 MB
1 Terabyte (TB)	2^{40} = 1,099,511,627,776 bytes	1024 GB
1 Petabyte (PB)	2^{50}= 1,125,899,906,842,624 bytes	1024 TB

Processor Architecture



General internal architecture of a processor

Processor Internal Architecture

The general internal architecture of any processor is shown

- System Bus:

  Address Bus, Data Bus, Control Bus

- Oscillator and timing module :

  generates the clock signal

- Control Unit :

  generates various control signals

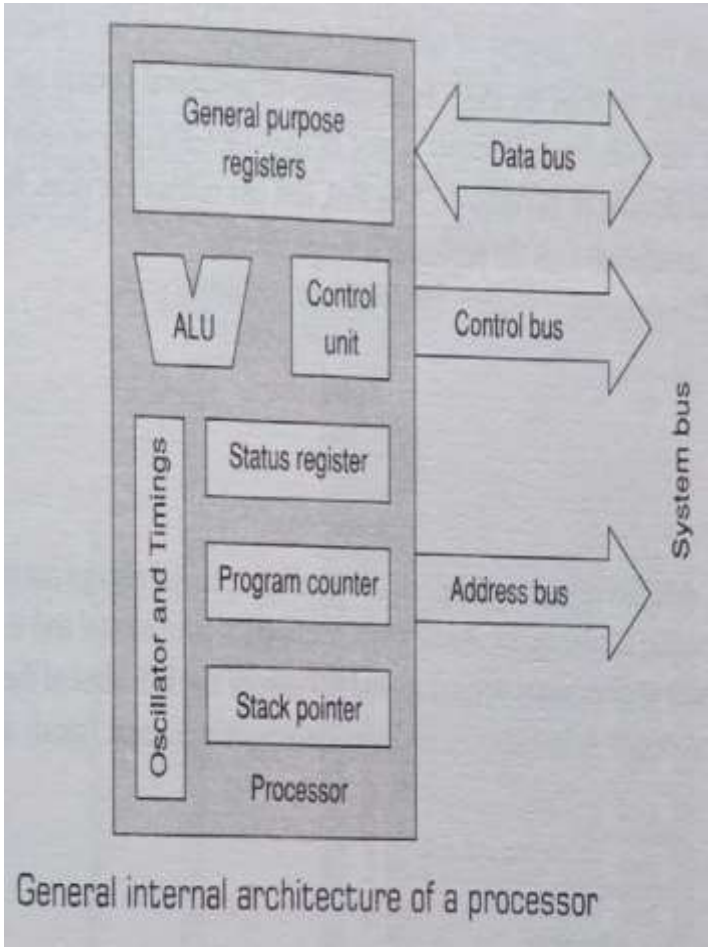
- ALU : performs arithmetic & logical operations

- General purpose registers :

  used for various storage and operations

- Special purpose registers :

  eg: PC, Status register, Stack Pointer etc.



General internal architecture of a processor

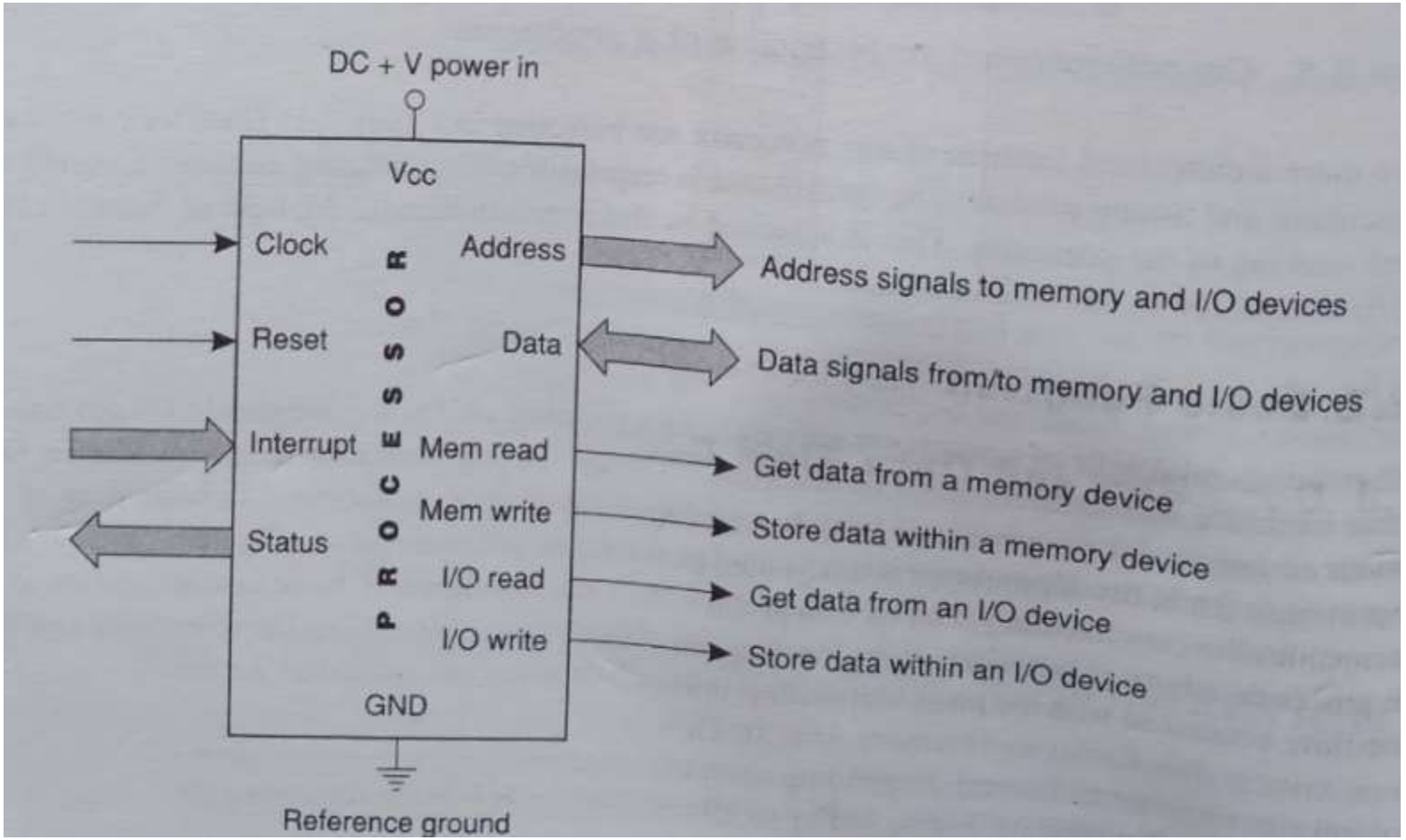
- **System Bus:**
  - Wires connecting memory & I/O peripherals to microprocessor
    - Address bus - carries memory addresses from the processor to other components such as primary storage and input/output devices. The address bus is unidirectional.
    - Data bus - carries the data between the processor and other components. The data bus is bidirectional.
    - Control bus - carries control signals from processor to other components. The control bus also carries the clock's pulses.(most of signals moves out of the processor)
- **Oscillator and Timing Module:** It generates signals like clock, reset etc needed for the working of processor.
- **Control Unit:** Responsible for generating all control signals and general working of the processor. This is achieved by the instructions with the help of internal clock, which is maintained by *Oscillator and Timing Module*

- **Arithmetic & Logic Unit (ALU):** Performs all the arithmetic and logical operations.
- **Set of general purpose register:** for various storage and operations while it is being manipulated.
- **Status Register :** Accommodates the **status** of the different arithmetic and logical operation, Normally used for conditional branching (flags)
- **Program Counter (Sequencing):**

  This is a register that is used to control the sequencing of the execution of instructions. This register always **holds the address of the instruction to be executed next.**
- **Stack Pointer:**

  This register indicates the **address of stack-top.**

External Signals of a Generic Processor



1. System Bus

Peripheral Devices and External Communication

- BUS is a group of wires connecting memory & I/O peripherals to microprocessor
- Memory and Input/Output devices(I/O devices) are the peripheral devices necessary for a processor to be functional
- They are connected to the processor through bus
- Every processor offers three major types of bus:
  - Address Bus : it is unidirectional and it carries address signals from the processor to all the external devices(memory and I/O)
  - Data Bus : it is bidirectional, as data must come in and go out of the processor
  - Control Bus : Most of the control signals go out of the processor. ex: Memory Read, Memory Write, I/O read, I/O write etc.

BUS in computer architecture

Inside computers, there are many internal components.

In order for these components to communicate with each other, they make use of wires that are known as a '**bus**'.

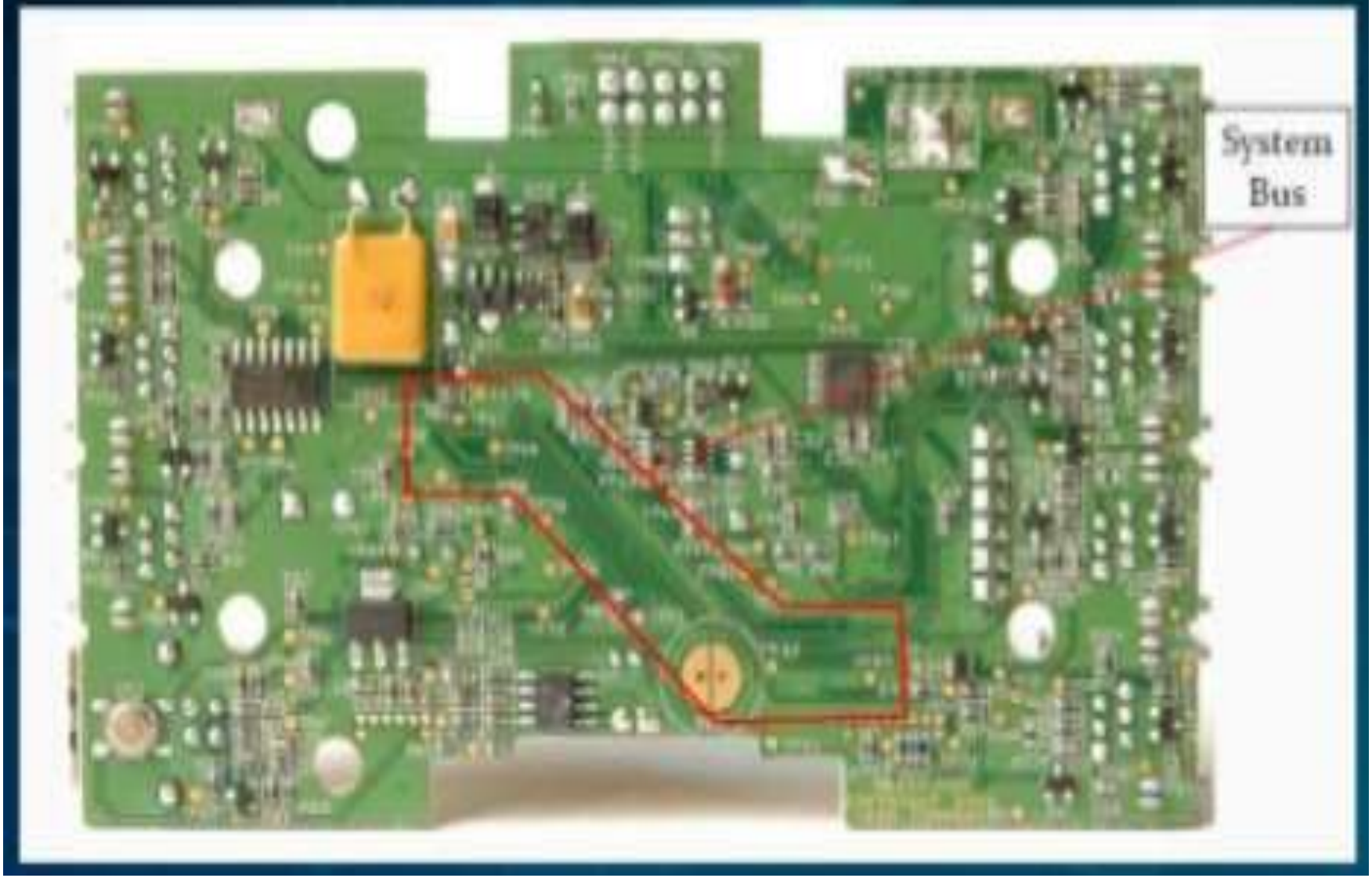
A **bus** is a **common pathway** through which information flows from one computer component to another.

This pathway is used for communication purposes and it is established between two or more computer components.

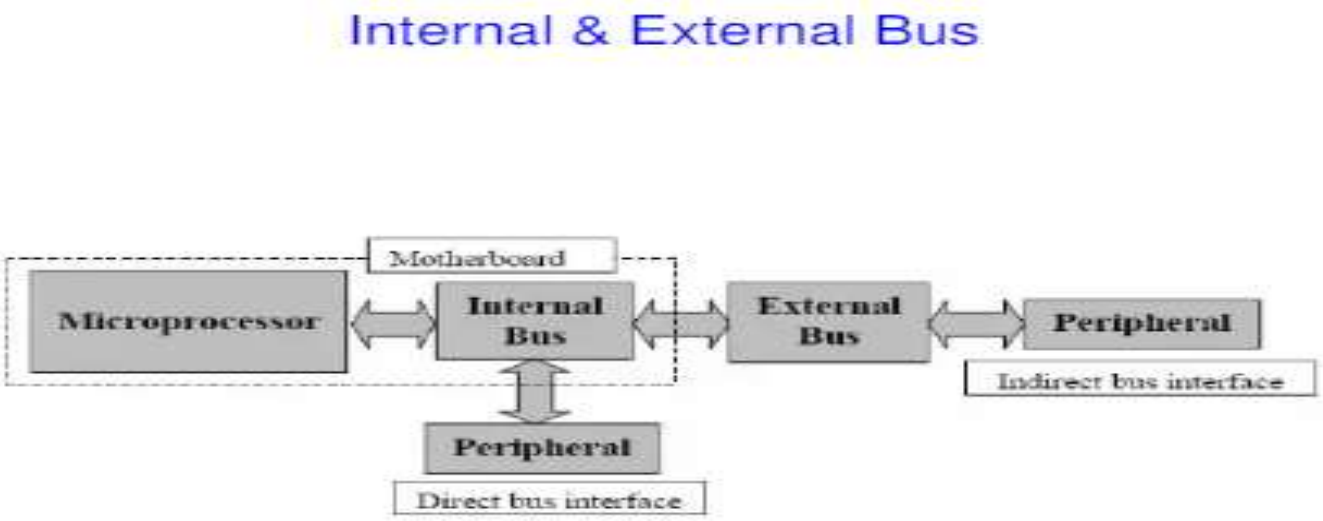
This network of wires or electronic pathways is called "BUS".

Bus is a bunch of signal lines intended to carry digital signals in computer system.

- Bottom of motherboard



Bus Terminologies



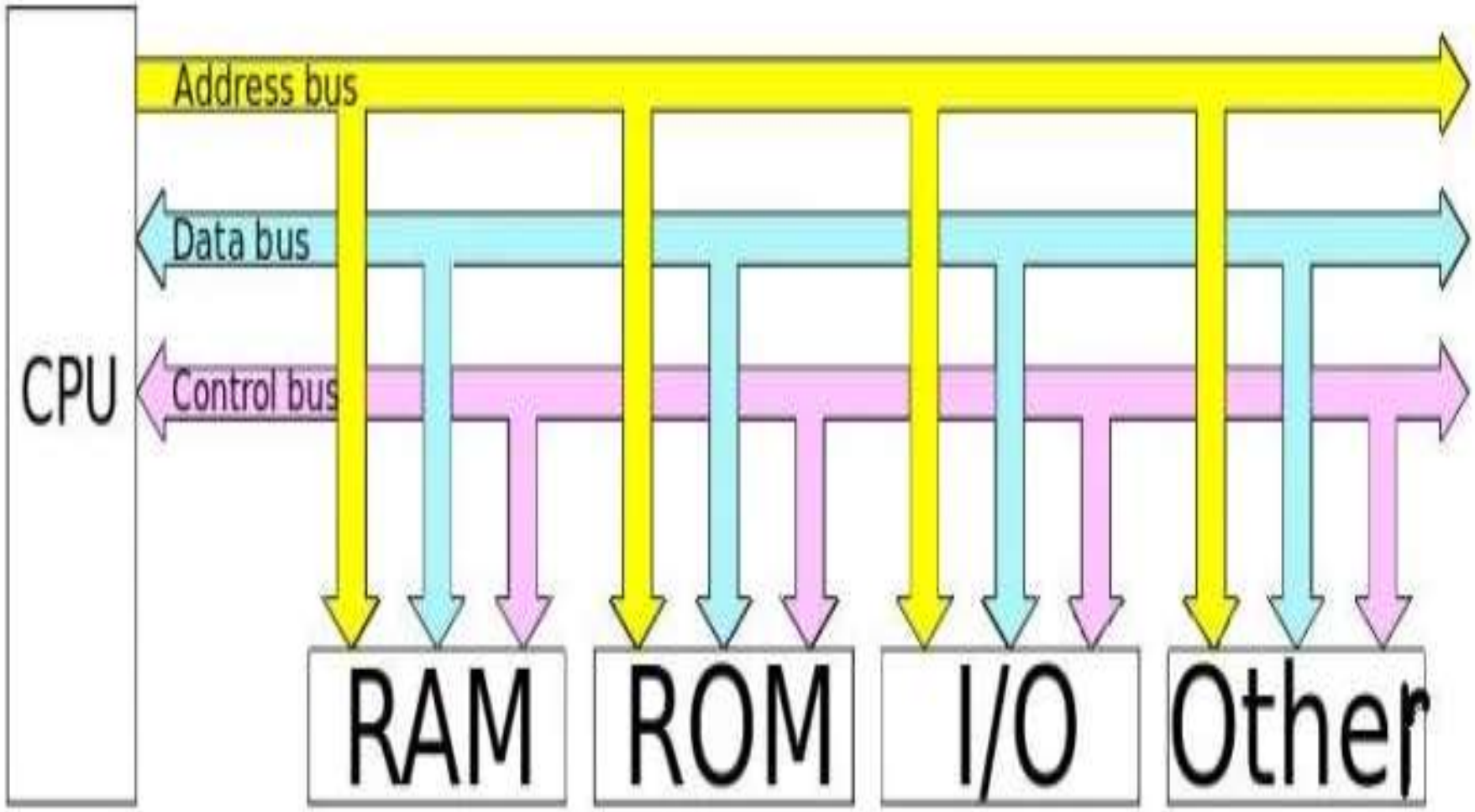
In a Computer system there are two types of buses:

Internal bus:- This is the bus that connects the CPU to the main memory on the motherboard. The system bus is also called the front-side bus, memory bus, local bus, or host bus.

External Bus:- connecting various peripheral devices to the CPU. Other name for the I/O bus include “expansion bus“

Internal BUS / System BUS

- A bus that connects major computer components (processsor, memory and I/O) is called a system bus.
- System bus is usually separated into three functional groups.
  - Data Bus (bidirectional)
  - Address Bus (unidirectional)
  - Control Bus(Most of the signals go out of CPU)
- In addition there may be power distribution lines that supply power to the attached modules.


Address bus
Data bus
Control bus
CPU
RAM
ROM
I/O
Other

❑ Address Bus

- Collection of wires used to identify particular location in main memory is called **ADDRESS BUS**.
- Or in other words the information used to describe the memory locations travels along the address bus.
- Clearly, width of the **address bus** or **number of address lines** available from any processor indicates maximum possible memory capacity of the system
- **n address lines directly address 2^n memory locations**
- It is a unidirectional bus
- CPU sends the address to a particular memory location or I/O device through the address bus.
- Address bus consists of 16, 20, 24 or more parallel signal lines.
- If the processor offers 16 address lines then it can address 2^{16} or 65536 locations. ie memory size of 64 K

2^{10}= 1024 memory locations ie 1K

- Each memory byte is identified by a *unique* memory address !!!

So:

- A address bus that consists of 8 wires (= bits), can convey 2^8 (= 256) different addresses
- A address bus that consists of 16 wires (= bits), can convey 2^{16} (= 64K) different addresses
- A address bus that consists of 24 wires (= bits), can convey 2^{24} (= 16M) different addresses
- A address bus that consists of 32 wires (= bits), can convey 2^{32} (= 4G) different addresses

  - If your PC has 8 G byte memory, then your PC has an address bus that contain at least 33 wires (bits) because 2^{33} = 8,589,934,592 (~= 8×10^9 = 8 G byte)

- All PCs has at least 33 bits address buses and can use 8 G byte memory

---

- Some (high end) PCs has more than 34 or 35 bits address bus and can use maximum 16 or 32 GBytes memory

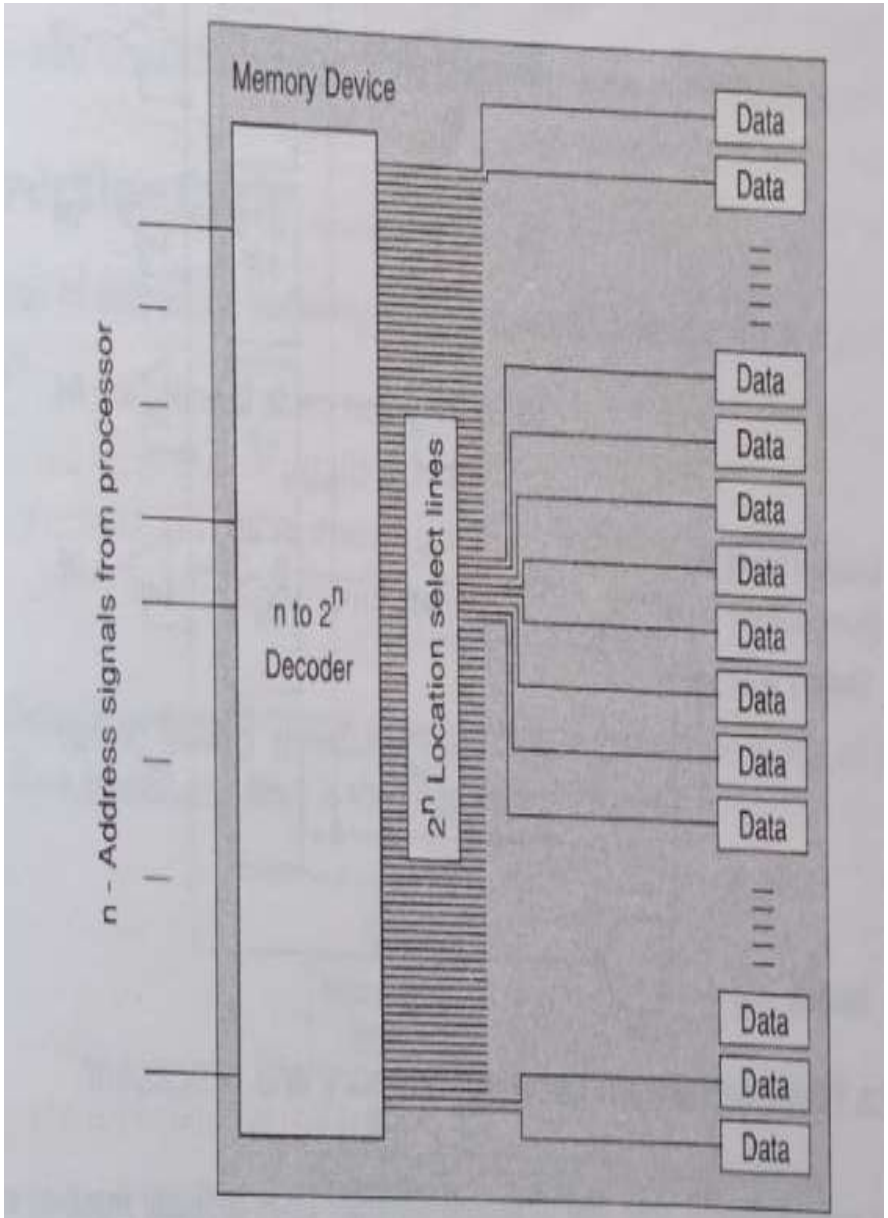
- Conclusion:

  - The width of the address bus determines the *size* of the memory that the computer can use

    - The wider the address bus, the more memory a computer can use.

      (More memory allows the computer to store more data and solve *larger* size problems, e.g., sort more data)

Address Bus and Addressing



Addressing of memory location by the processor

Address signals are decoded by a decoder inside the memory or I/O device to target the desired location.

❑ Data Bus

- It is 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 path on which data travels within a computer.
- This bus connects all the computer components to the CPU and main memory.
- The data bus may consists 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 one bit at a time, number of lines determines how many bits can be carried at a time.
- The width of data bus of any processor indicates its simultaneous handling capability of the maximum number of bits.

- It is a bidirectional bus.
- The size(width) of the bus determines how much data can be transmitted at one time.

  Eg:

  - An 8-bit bus can transmit 8 bits (1 byte) of data at a time.
  - A 16- bit bus can transmit 16 bits (2 bytes) of data at a time.
  - 32- bit bus can transmit 32 bits (4 bytes) of data at a time.
- Generally, a processor is designated by its data bus width.

  For example,

  - an 8-bit processor is capable of communicating 8-bit data at the same time or having an 8-bit data bus
  - Similarly, a 16-bit processor has 16 parallel data lines for data communications
- The size of a bus is a critical parameter in determining system performance.
- The wider the data bus, the better , but they are expensive.

- Width of the databus = # wires used in the data bus

  (The more wires you use, the more costly the computer system will become - it's like the number of lanes in a high way system

- Effect of the width (= number of wires) of the databus:

  - A databus that consists of 8 bits, can transfer 1 byte of data per read/write operation
  - A databus that consists of 16 bits, can transfer 2 bytes of data per read/write operation
  - A databus that consists of 32 bits, can transfer 4 bytes of data per read/write operation
  - And so on

- Conclusion:

  - The width of the data bus determines the *amount* of data transfered per memory transfer operation

    - The wider (= more wires) the the daate bus, the more data you can transfer per time unit (second)

      That will result is a faster running computer

  (The width of the databus is analogous to the # lanes on the high way: the more lanes (= more wires), the more cars (more bits) you can transfer per time unit)

- Current trend:

  - All computers made in 2019 has 64 bits databuses (64 bit machine)

  - Back in 2012, computers used to have 32 bits data buses

Data Bus and Data Flow Control

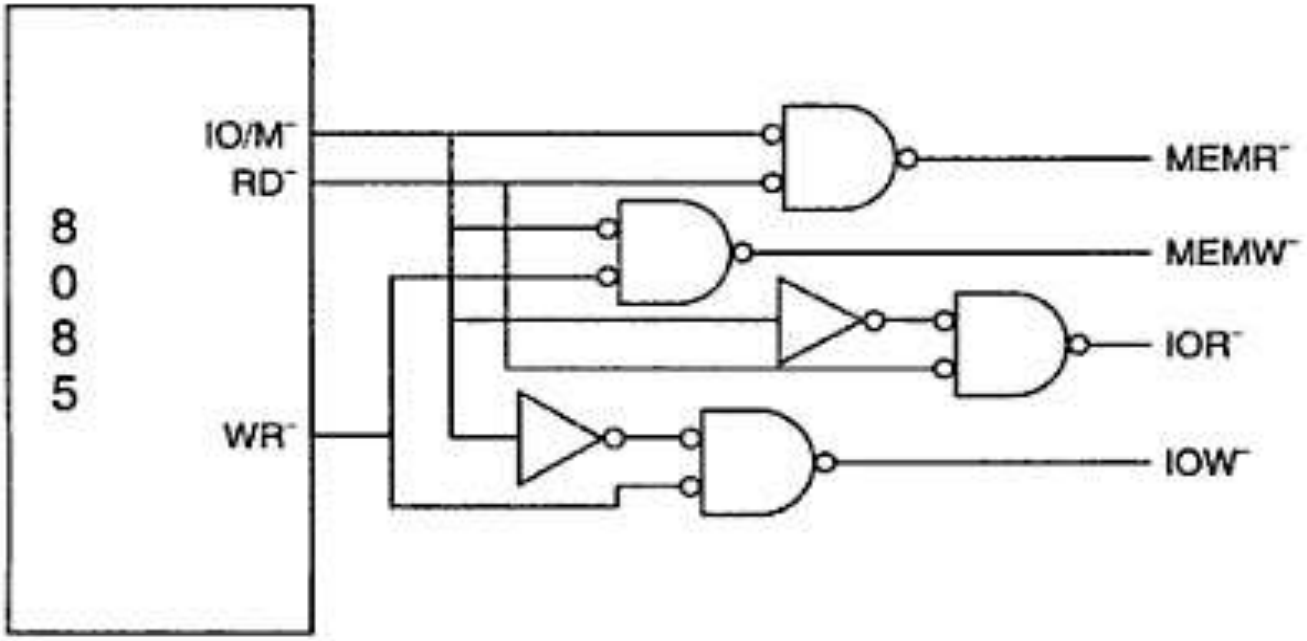
- The flow of data is bi-directional, depending on whether the processor is interested in reading from or writing into the device(memory or I/O)
- This intension of the processor is expressed through its control signals(read and write)

❑ Control Bus

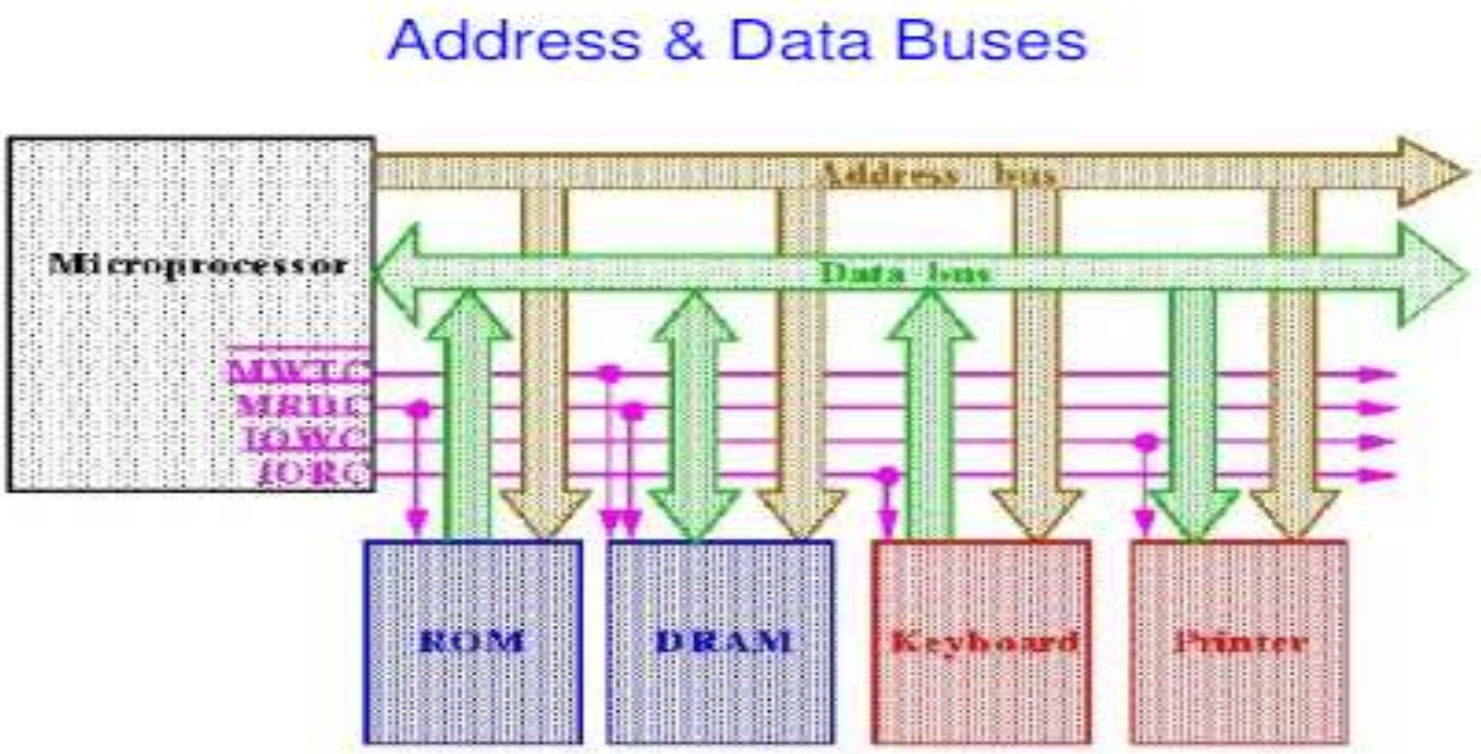
- Because the data and address lines are shared by all components, there must be a means of controlling their use.
- The control lines regulate the activity on the bus.
- Control signals transmit both command and timing information among the system modules.
- Control bus carries signals that reports the status of various devices.
- Number and functions of control signals, constituting the control bus varies with the processor

- Two important control signals are READ and WRITE
- As the processor is to interact with two types of devices ie memory and I/O, there are four READ/WRITE signals .
- Typical control bus signals are:
  - Memory Read: causes data from 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 Read: causes data from addressed I/O port to be placed on the data bus.
  - I/O Write: causes data on the bus to be output to the addressed I/O port.

Generation of Control Signals

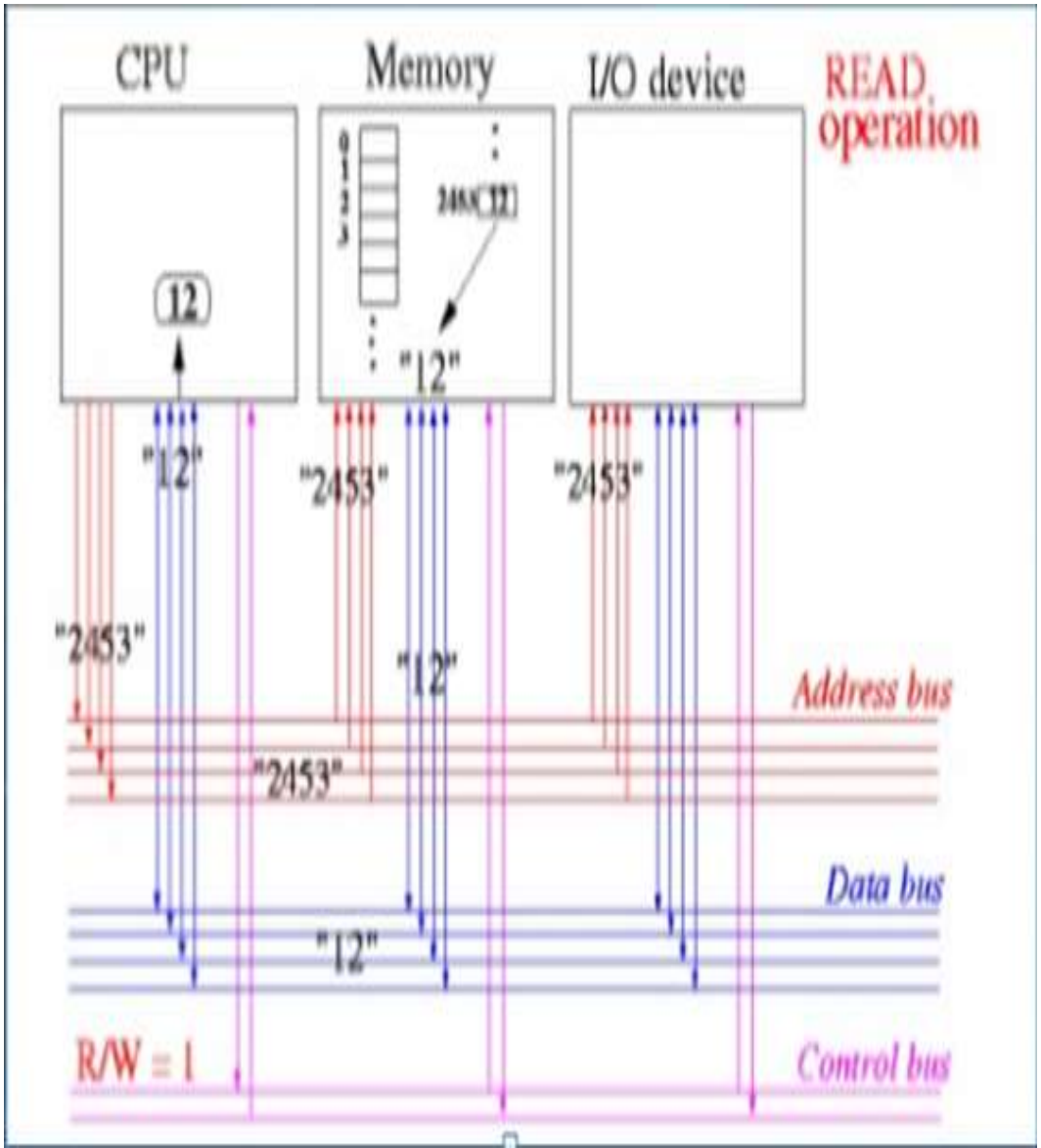


Address Bus, Data Bus and Control Bus



Example: Memory Read

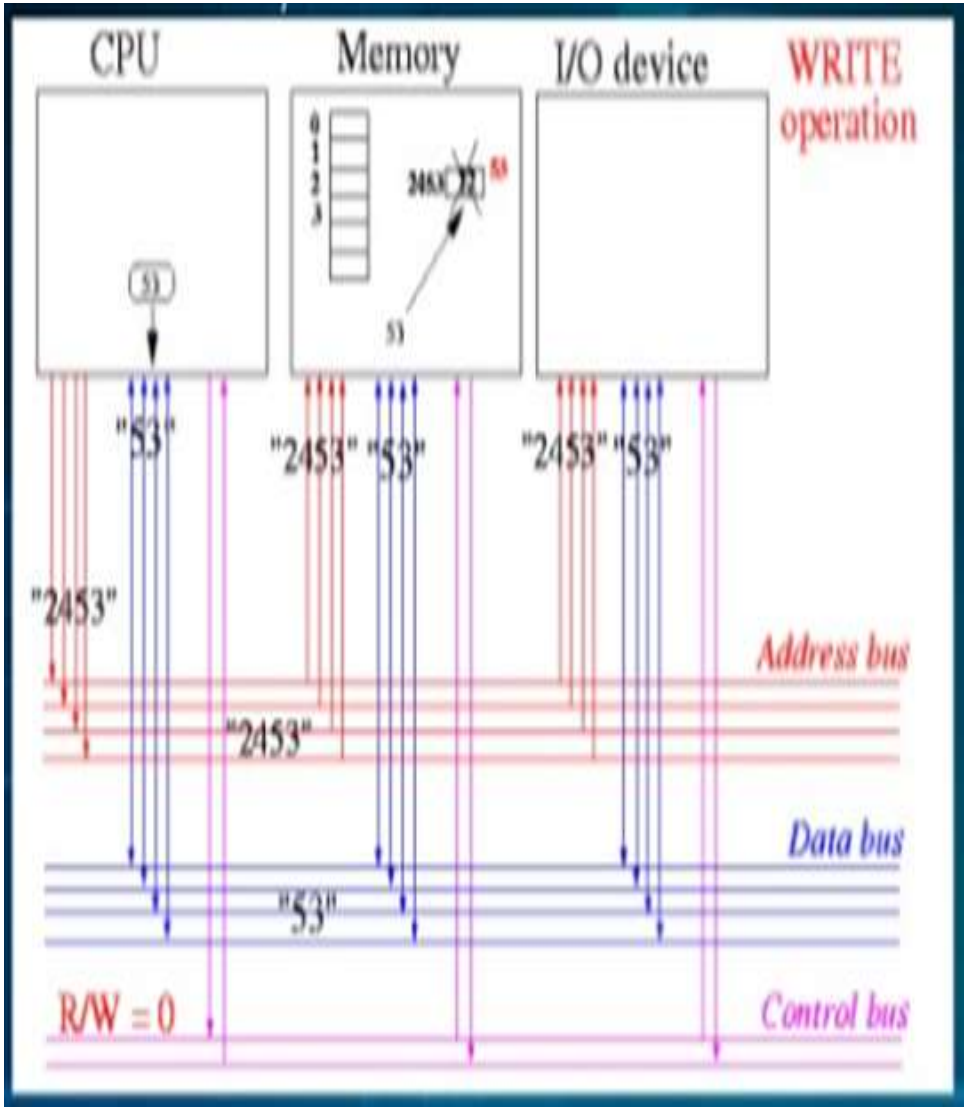
The following figure shows how CPU reads the value 12 from memory location 2453h



- CPU sends out the address value 2453 on the ***address bus***.
- Simultaneously CPU sends out the signal ***R/W=1*** on the ***control bus,*** which indicates ***a read operation.***
- CPU then waits for the data from memory on the ***data bus***.
- The R/W=1 signal and the address bus value = 2453 will cause the memory to retrieve the value at memory location 2453 to be send out on data bus.

Example: Memory Write

The following figure shows how CPU writes the value 53 to memory location 2453h



- CPU sends out the address value 2453 on the ***address bus***.
- Simultaneously CPU also sends out the value 53 on the data bus.
- and the signal ***R/W=0*** on the ***control bus,*** which indicates ***a write operation.***
- CPU then waits for the data from memory on the ***data bus***.
- The R/W=0 signal along with the address bus value = 2453 and data bus value 53will cause the memory to store the value 53 at memory location 2453.

BUS Types:

- Dedicated buses
  - Separate buses dedicated to carry data and address information
  - Good performance
    - But increases cost
- Multiplexed buses
  - Data and address information is time multiplexed on a shared bus.
  - Reduced performance
    - But reduces cost

Dedicated Time multiplexed bus:

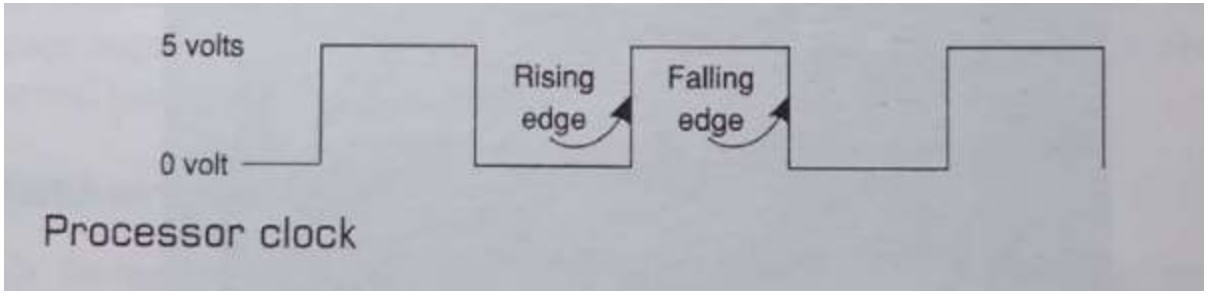
- ***Address and data information may be transmitted over the same set of lines using an address valid control line.***
- ***During the beginning of data transfer, address is placed on the bus and address valid line is activated.***
- ***At this point each module has a specified period of time to copy the address and determine if it is their address in the address bus.***
- ***After the specified period address is removed from the bus and the same bus can be used to carry data.***
- ***This method of using the same lines for multipurpose is known as time- multiplexing.***

2. Oscillator and Timing Module

Processor Clock

- The CPU contains a clock which, along with the CU, is used to coordinate all of the computer's components.
- The clock sends out a regular electrical pulse which synchronises (keeps in time) all the components.
- The heart of the processor is its clock, which starts the moment the computer is turned ON.
- 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.

- It is a digital signal producing on and off states alternately, at equal time intervals.



- The oscillating phenomenon generates two different edges of the clock signal , rising edge(positive edge) and falling edge(negative edge)
- Processor starts a basic activity at these edges

3. Control Unit

- Control Unit is responsible for generating all control signals and general working of the processor. This is achieved by the instructions with the help of internal clock which is maintained by the internal clock.
- A control unit (CU) handles all processor control signals. It directs all input and output flow, fetches the code for instructions and controlling how data moves around the system
- The CU provides several functions:
  - it **fetches**, **decodes** and **executes instructions**
  - it issues control signals that control **hardware** components within the CPU
  - it transfers **data** and instructions around the system

4.The Arithmetic/Logic Unit (ALU)

The arithmetic logic unit is that part of the CPU that handles all the calculations the CPU may need, e.g. Addition, Subtraction, Comparisons. It performs Logical Operations, Bit Shifting Operations, and Arithmetic Operation.

The **ALU** has two main functions:

- it performs arithmetic and logical operations (decisions).
- it acts as a gateway between **primary storage** and **secondary storage** - data transferred between them passes through the ALU.

5. Register Set

- Registers are small amounts of high-speed memory contained within the CPU. They are used by the processor to store small amounts of data that are needed during processing, such as:
  - the address of the next instruction to be executed
  - the current instruction being decoded
  - the results of calculations
- To perform internal operations, all processors offer some internal registers, which can store temporary information or some operands (data).
- It can be divided into two types
  - General Purpose registers
  - Specific Purpose registers

5.a. General Purpose Registers

- Available for use by the programmer
- Useful for remembering frequently used data
- Why is it a good idea to do this?
  - There is a large speed disparity between the processor (CPU) and the memory where instructions and data are stored
  - Consider a 1GHz processor, this frequency corresponds to 1 nanosecond time scale
  - Memory: is having approx, 100 nanosecond time scale
  - So, since memory is almost 100 times slower than processor, keeping frequently used data inside the processor (in Registers) will make the processing faster.

5.b. Specific purpose registers:

- They are used for some specific purposes.
  - Examples: Accumulator, Status register, Stack Pointer, Program Counter etc. – these are accessible by the programmer
  - There are certain other specific purpose registers which can be accessed only by the processor. Eg: temporary registers

Accumulator

- Stores the results of calculations made by ALU.
- In a computer's CPU, Accumulator register is used to store the result of an arithmetic operation.
- Also, in some arithmetic and logical operations, Accumulator register is used to hold one of the operand(for example in an addition operation).

Status Register:

- It is a register in the processor used to remember status information about current state of processor.
- It accommodates the status of the different arithmetic and logical operation.
- The status register is used to reflect the result status of the last performed arithmetic or logical operation through its pre-assigned bits.
- Generally ,each bits of this status register is assigned for one particular indication, eg, carry, parity, zero and so on.
- Normally used for conditional branching (flags).
- After performing an arithmetic or logical operation, the result of the operation can be zero or negative or produced a carry or odd/even parity etc.
- Eg: an arithmetic overflow has occurred.

Stack pointer (SP)

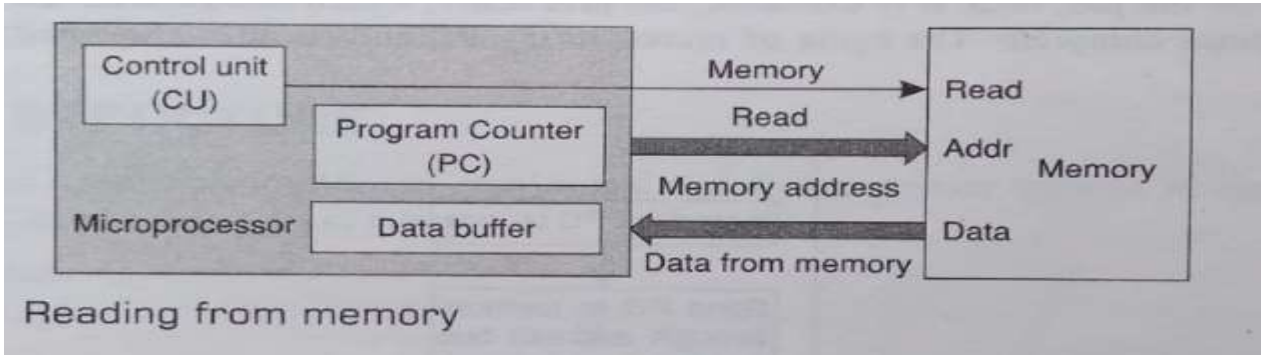
- Stack Pointer is a register used to point to a special type of memory area called stack.
- Stack is a memory area, which is used by the processor to keep data temporarily.
- The stack is an area of memory used to hold data that will be retrieved soon.
- The stack is usually accessed in a Last in First out (LIFO) fashion.
- Stack pointer always points to the top of the stack area.

Program Counter(PC):

- A program counter is a register in a computer processor that contains the address (location) of the instruction being executed at the current time.
- This is a register that is used to control the sequencing of the execution of instructions.
- This register always **holds the address of the instruction to be executed next.**
- As each instruction gets fetched, the program counter increases its stored value by 1.

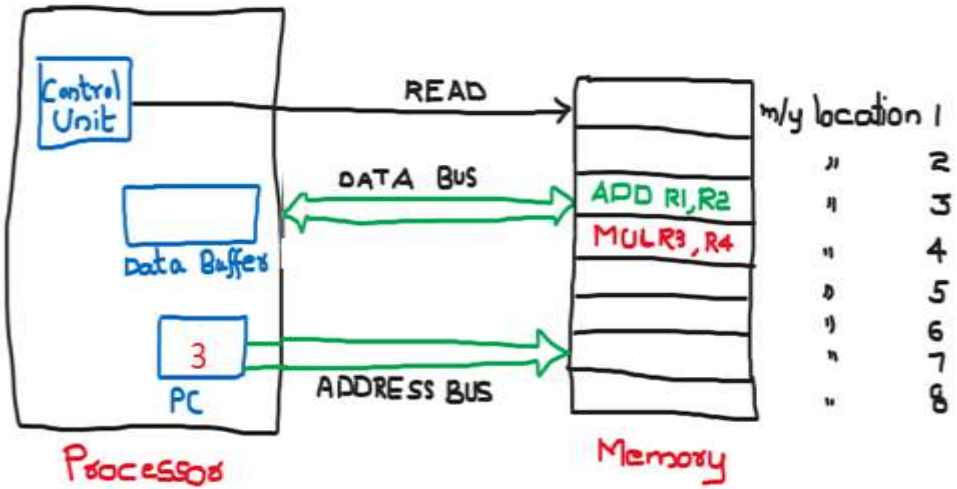
More about the role of PC – Reading from Memory

- Step 1: The content of PC is sent out to the memory through a bus(This bus is known as Address Bus since address of a memory location is transferred through this bus)
- Step 2: The control unit send a Memory Read control signal to the Memory
- Step 3: Memory then transfers the content of that memory location to the Processor through a bus known as Data Bus

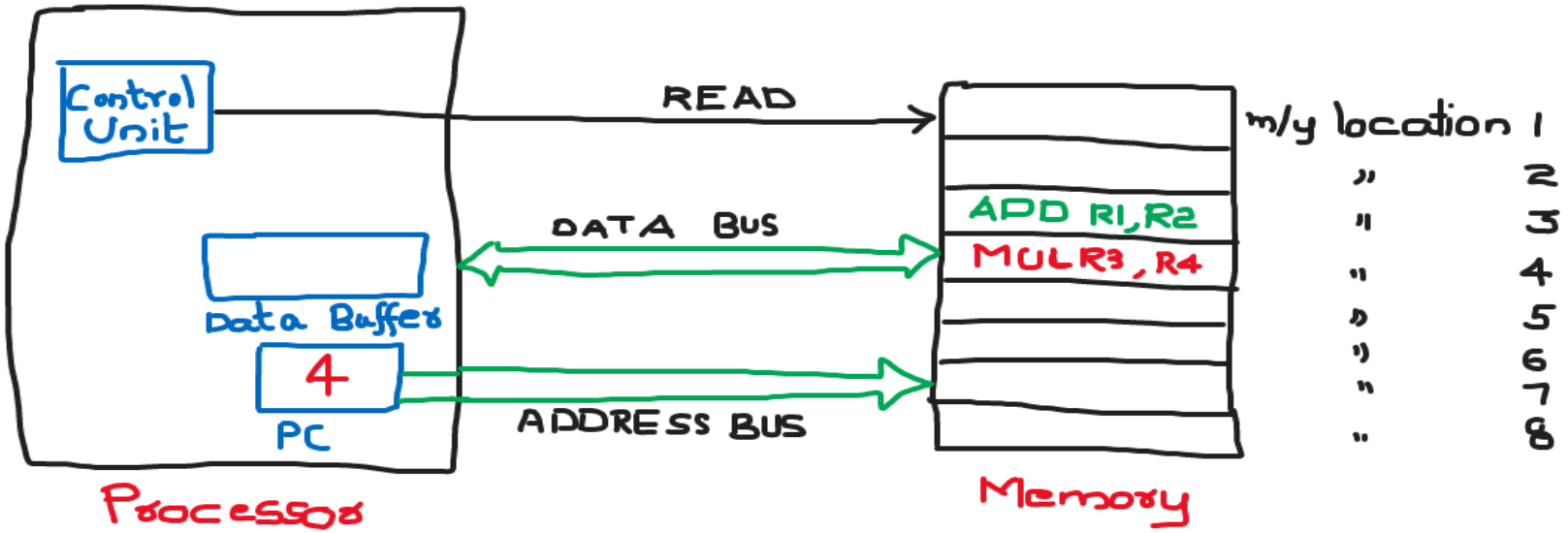


More about the role of PC –Reading from Memory

- Consider a scenario as shown below where the processor wants to execute the instruction residing in memory location having address 3
- First let us assume that the content of PC is 3 where the instruction ADD R1,R2 resides.
- So to fetch the content of memory location 3, PC content is send to memory through a bus known as address bus. The content of address bus will uniquely identify the memory location 3 in memory.
- The control unit sends a READ control signal to the memory.
- When the memory unit receives the READ signal, it will transfer the content of the third Memory Location to the processor (to Data Buffer) through Data Bus



- Once the processor understands what to do, then it will start executing that.
- The execution may vary depending on the type of instruction. Here in this example, execution is the Addition of two registers R1 and R2.
- Once execution is over, the PC is incremented by one to start fetching and executing the next instruction (MUL R3, R4) from the memory location 4 as shown below

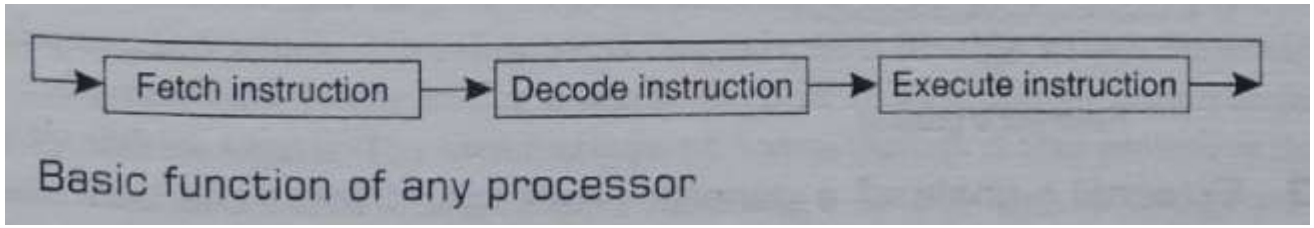


Basic Operational Concepts

- The two duties of an computer when it is operational are
  - Executing the software by fetching instructions from memory
  - Look for any external signal and react accordingly
- Both the duties are carried out by Processor
- The second duty deals with Interrupt handling
- So, here, we will discuss about the first duty, ie, execution of a program(or software).

Basic Function of a Processor

- Processor is meant for executing a software
- The basic duty of any processor is to fetch, decode and execute instructions as long as it is powered on



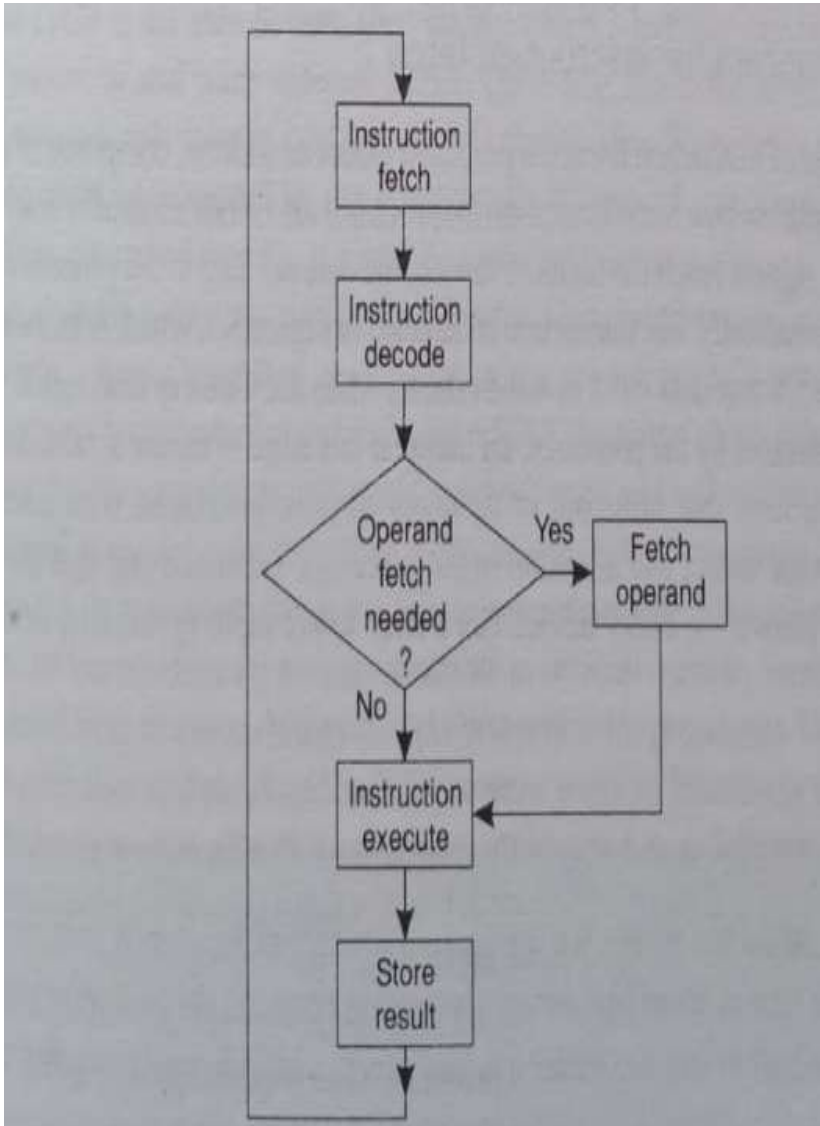
Basic function of any processor

Processor Operation

- The job of the processor is to execute programs which are composed of multiple instructions
- To execute an instruction, the processor should perform the following steps
  - Fetch
  - Decode
  - Execute
- The combination of these three steps are known as an Instruction Cycle

Instruction Cycle

- After fetching the instruction and decoding, processor checks for any operand fetch, which might be necessary for some instructions.
- If needed, the operand is fetched from memory and then the instruction is executed
- Finally, the result of the instruction is stored and the whole cycle is repeated.

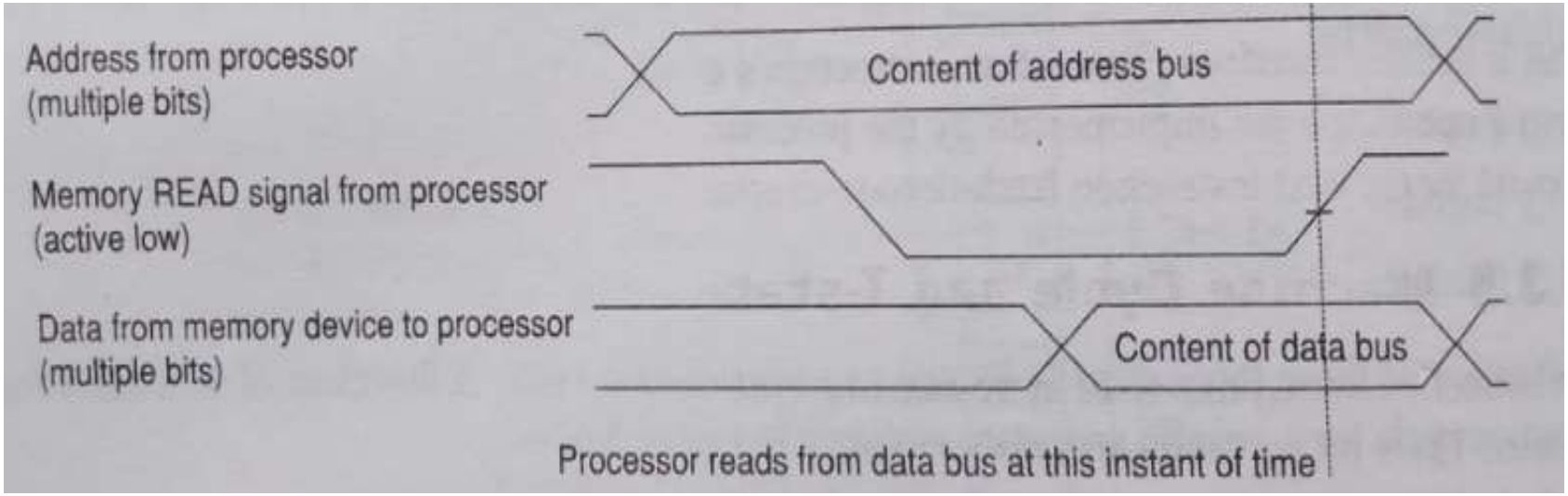


Flowchart for simplified instruction cycle

Instruction Fetch

- The first step is to fetch the instruction byte(s) from memory
- The memory contains many memory locations
- So, the processor needs to pin-point the correct location in the memory to fetch the instruction.
- Every memory location has a unique binary address
- After receiving the address, the duty of the memory device is to decode the address to locate the memory location and place the instruction on the data bus.
  - The instruction will then be available for the processor

Timing Diagram for Instruction Fetch



- Processor places the address of the memory location on the address bus
- Simultaneously, the processor also sends a memory read signal through its control bus.
- When these signals reach the memory device, the data are sent to the processor
- Schematically, this transaction is shown in the figure, which is known as Timing Diagram

Instruction Decode

- After receiving the instruction, the processor becomes busy in understanding it(What to do?)
- This part is known as instruction decode, carried out within the processor itself.
- After the completion of instruction decoding, the processor knows whether to fetch operands from external memory or to increment a register by one or to store a register content in memory location etc.
- Decoding can be implemented in hardware or software

Instruction Execute

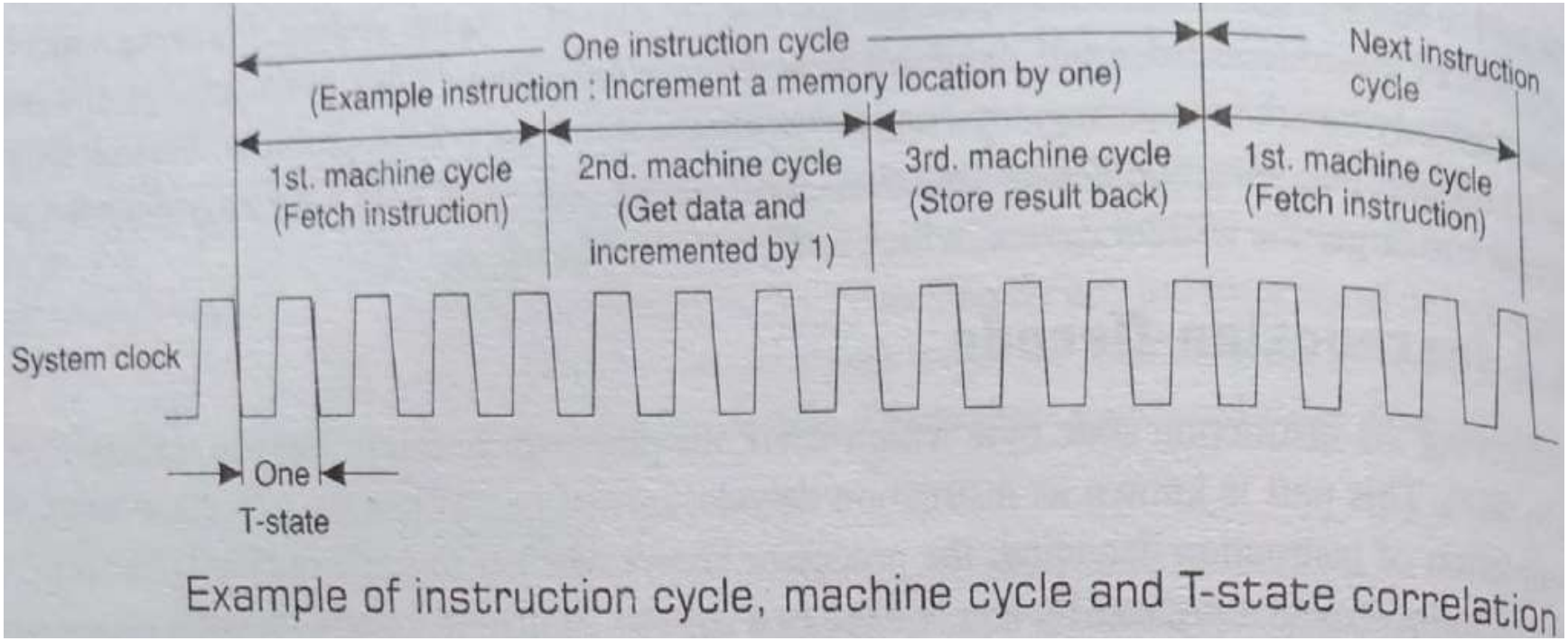
- This is the last and final phase of an instruction's execution
- Depending upon the instruction, one or several operations are implemented by the processor.
- Once this part is complete, the processor looks forward for the next instruction fetch-decode-execute, and the process continues.

Machine Cycles & T- States

- An ***instruction cycle*** has 1 or more ***machine cycles***
  - ***A machine cycle*** is the ***time – slice*** during which ***1- byte*** of data is transferred between processor and memory (or I/O device).
- And every ***machine cycle*** is composed of ***several T- states.***
  - ***One complete oscillation*** of processor clock is known as ***T-state.***
  - **Number of T- states** required to complete **one machine cycle** will be different for different processors based on the clock frequency they are using.

We can see an example showing the correlation between instruction cycle, machine cycle and T- state.

Execution of an instruction "***increment a memory location by one"*** is illustrated in the figure, which shows that 3 machine cycles are needed to complete one instruction cycle. It also shows the number of T- states needed in each machine cycle.



Example of instruction cycle, machine cycle and T-state correlation

University Questions

- Which is more important for the functioning of a basic processor, Program Counter or Stack Pointer. Justify your answer [3 marks]
- Explain Instruction Cycle with a sample timing diagram [10 marks]
- Differentiate between Harvard and Von-neumann architecture [3 marks]
- Write down the control signal for a register transfer. [3 marks]
- Explain the different stages of microprocessor operations [6 marks]
- Explain the role of different buses in a processor architecture. [8 marks]
- Explain the data path for branch execution showing all control signals and sequences. [6 marks]
- Explain the function of following registers: status register, accumulator, program counter, stack pointer, general purpose registers. [8marks]