1 Basic structure of Computer

- **Computer Architectures :-**

Computer architecture refers to the design and organization of a computer's components and systems. it includes the hardware components, their interconnections and how actually they are work together.

- **Computer organizations :-**

Computer organization is realisation of what is specified by the computer architecture It deals with how operational attributes are linked together.

Computer Architecture	Computer Organization
- Focuses on design and functionality	- Focuses on physical hardware
- It includes instructions set and data formats	- includes how components are connected
- Tells what the computer can do	- Tells how the computer is built
Note :- It works on what principle	Note :- It works on how principle

Date
Page

Computer types

- Based on Operating principles.

1) Digital computers :- process discrete (discontinous) data in binary form (0s and 1s). ex :- pcs and smartphones

2) Analog computers :- process continous data. They use physical quantites like voltage. ex :- old-school speedometer

3) hybrid computers :- combine both analog and digital features. They can process both continous and discrete data. ~~ex :-~~
ex :- some medical devices

- Based on Sized & capability.

1) Super computers :- very large, extremely powerful, used for complex tasks like scientific simulations

2) mainframe computers :- large systems, powerful enough for bulk data processing in big organizations

3) Minicomputers :- smaller than mainframes, used for specific tasks in buisiness.

4) personal computers :- desktops and laptops for individual use, suitable for everyday tasks.

5) workstations :- High performance pcs designed for technical tasks like graphic design or engineering.

6) micro computers :- small computers, often called personal computers.
designed for individual use

~~* functional units~~

~~- Functional units are the building blocks of a computer.~~

- Based on number of microprocessors

1) sequential computers :- process one task at a single time. instructions are executed in order.

2) parallel computers :- process multiple tasks at the same time. divide a problem into a smaller parts that runs simultaneously.

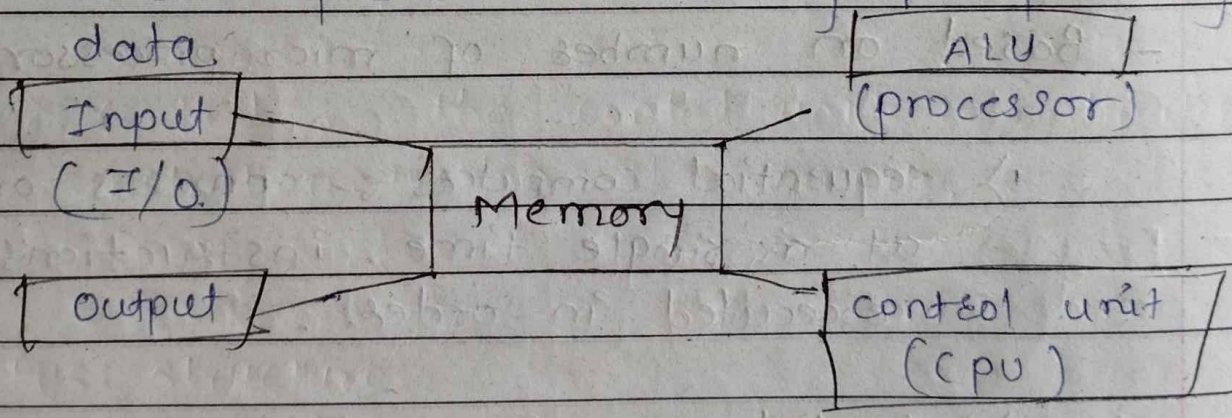
[Note :- List of instructions are called programs and internal storage is called computer memory]

Different types of computer

1) personal computer :- It is most common type of desktop that is use for individual use. ex :- desktops and laptops

Functional units

Functional units are the building blocks of a computer's architecture. They perform various operations necessary for processing data.



* Functional units of computer *

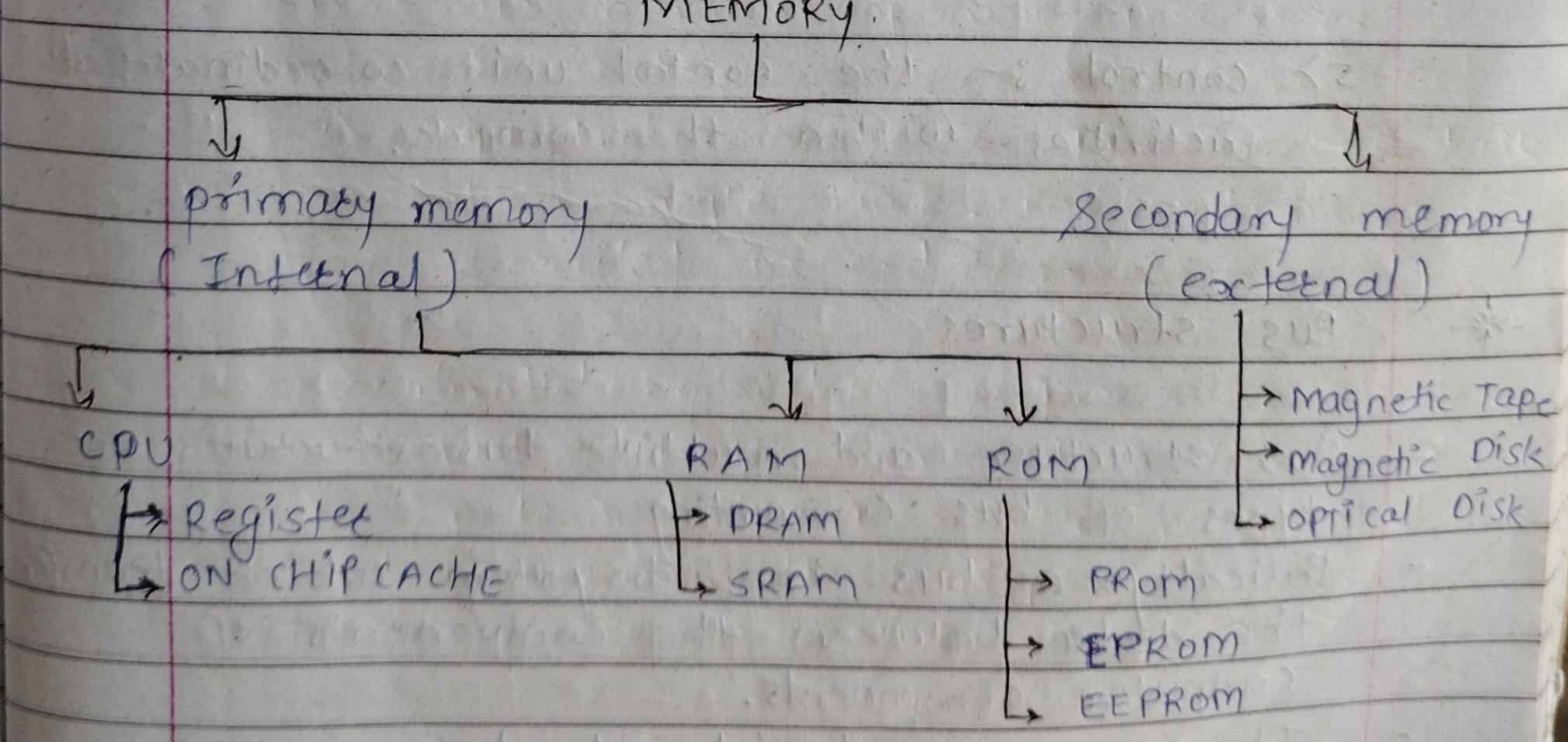
- Input / output ('i/o) units :- These allows the computer to interact or comunicate ~~to~~ ~~or~~ with the outside world.
  ex :- mouse or keyboard

- central processing unit (CPU) :- this is brain of the computer. It executes a instruction or process data. it consists of :-

- (ALU) Arithmetic logic unit :- performs arithmetic ~~oper~~ and logical operations (like addition, subtraction and comparison)

- (CU) control unit :- directs the operations of the processor. It tells other parts of the computer how to respond to the instruction

* Types of memory *



* Basic operational concepts of computer

1) input :- Computers receive data through input devices like keyboards, mouse, etc

2) Output :- processed data is sent to output screens like, monitors.

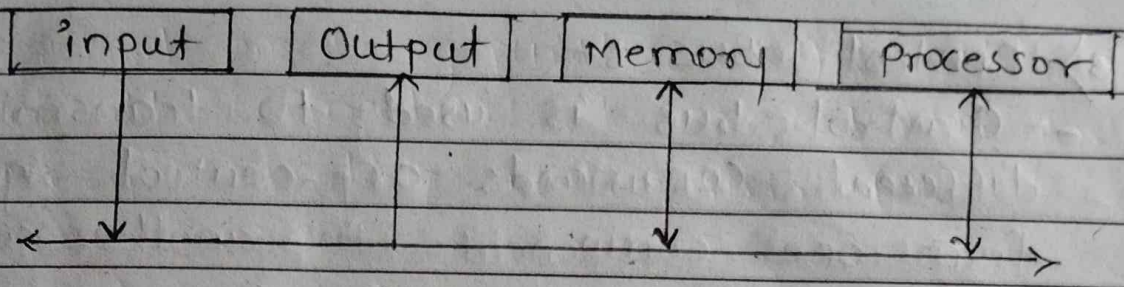
3) Storage :- Data is stored in memory for quick access and in secondary storage for long time access.

4) processing :- The CPU processes the input data by performing calculations and executing instructions.

5) control :- The control unit co-ordinates all activities within the computer.

* Bus structures

- Bus structure and multiple bus structures are types of bus computing
- Basically a bus is subsystem which transfer the data between the components of computer, components.
- It connects pheripheral devices at the same time



System Bus.

Types of Buses

1) Data Bus :- Data bus is the most common type of bus.
- It is used to transfer data between different components of computer.
- The data bus consists of 8, 16, 32, or 64 lines.
- A 64-line data bus can transfer 64 bits of data at one time

2) Address Bus :-
- Many components are connected to one another through buses. Each component is assigned a unique id. This id is called the address of that component.
- The address bus is a unidirectional bus
- It can carry information only in one direction.
- It carries address of memory location from microprocessor to the main memory

3) Control Bus :-

- Control bus is used to transmit different commands or control signals from one component to another component.
- Control bus is like the communication line in a computer that helps different parts talk to each other.
- Control signals carries signals that tell parts what to do.
- It works alongside the data bus and address bus.

Performance

- The performance of the computer bus is really important ~~because~~ for how well a computer system runs.
- First, speed is important because the faster the control signals can be sent
- then there's bandwidth, which refers how many signals can be sent at the same time. if the bandwidth is high then the system can handle more commands simultaneously.
- latency is less delay leads to faster overall operations.
- error rate is fewer errors ensure reliable communication.

Metrics.

- Metrics are basically measurements that helps us understand the performance of a system or process.
- Response time is important too because it tells us how long it takes for a request to get a response.
  - A lower response time means users get feedback quickly.
- Error rate is a crucial metric as well because it shows how often errors occurs

Multiprocessing and multicomputers.

- Multiprocessing.
  - Executes a number of different application tasks in parallel
  - All processors have access to all of the memory - shared memory multiprocessors
  - Execute subtasks of a single large tasks in parallel
  - multiprocessing is commonly used in applications like video processing, etc
  - Tasks complete faster due to simultaneous execution.

- Multicomputers.
  - Each computer only have access to its own memory
  - It refers to a setup where multiple independent computers work together.
  - But, they are not directly connected to each other. Each computer operates with its own recources and memory
  - If one computers fails, then the others continue to operate.
  - New hardware can be added easily
  - Different computers can share their resources.

★ Difference between multiprocessing or multicompute

Multiprocessing	Multicomputers
- speed depends on the all processors speed	- It can run faster
- A computer that has more than one CPU on its motherboard	- A computer made up of several computers.
- process can share the memory	- processor can not share the memory
- called as shared memory multi processor	- called as message passing multi computers.
- Cost is low	- Cost is more

- Memory Locations.

- Memory in computers is organized in a structure called memory location
- Each memory location can store a certain amount of data, typically measured in bytes.

- Byte Addressibility.

- Byte Addressibility means that each byte in memory has its own unique address.
- This allows the CPU to access individual bytes directly.
- For example, if a memory starts at address 0, the first byte is at address 0, the second byte ~~address~~ at addres 1 and so on

- Big Endian.

- In big-endian format, the most significant byte (the "big end") is stored at the lowest memory address.
- For example if we have the 4-bytes integer 0x12345678, it would be stored in memory like this:
  - Address 0 : 0x12
  - Address 1 : 0x34
  - Address 2 : 0x56
  - Address 3 : 0x78

- Little Endian

- In little-endian format, the least significant byte (the "little end") is stored at the lowest memory address.
- Using the 4-byte integer 0x12345678, it would be stored in memory like this,
  - Address 0 : 0x78
  - Address 1 : 0x56
  - Address 2 : 0x34
  - Address 3 : 0x12

- Addresses

- Address in memory refers to a specific location where data is stored.
- Each address is unique and allows the CPU to access data quickly.
- A memory address is a unique identifier for a specific location in a computer's RAM.
- It allows the CPU to read from or write to that location in memory.

11

Basic processing unit.

Date
Page

- Memory.
  - Memory in computer architecture is where data and instructions are stored for the CPU to access.
  - It is crucial for the performance of a computer system.
  - There are two main types of memory primary memory and secondary memory.

  - primary memory (volatile) :-
    - This includes RAM, which is used for temporarily storing data that the CPU is currently working on.
    - It is Fast but loses all its data when the power is tured off

  - Secondary memory (Non-volatile) :-
    - This includes hard drives and SSDs which store data permanently, even when the power is off!
    - They are slower than RAM but have much larger storage capacities.

- Cache Memory.
  - This is a smaller, faster type of volatile memory that provides high-speed access to frequently used data and instructions.
  - It is located closer to the CPU to speed up processing.

- Cache Memory.
  - Cache memory is a high-speed storage area located between the CPU and the RAM.
  - Cache memory holds data and instructions that the CPU uses repeatedly.
  - Cache memory reduces the time needed to retrive data from the RAM, which is slower
  - When the CPU needs data, it first checks the cache. If the data is found there, it can be accessed quickly.
    If each data is not found then the CPU has to retrive the data from the RAM, which takes more time.

- Cache memory principles.

- Cache memory principles involves several key concepts that help improve the efficiency and speed of data access in computer systems. Here are the main principles:

1) Locality Reference

- Locality Reference is both temporal and spatial locality.

- Temporal Locality :- This principle state that if a particular piece of data or instruction is accessed, it is likely to be accessed again in the near future

- Spatial Locality :- This principle suggests that if a particular data location is accessed, nearby data locations are likely to be accessed soon.

2) Cache Organization

- Direct-Mapped Cache :- Each block of main memory maps to exactly one cache line

- Set-Associative Cache : This is a compromise between direct-mapped and fully associative caches

- Fully Associative Cache : Any block can go into any line of the cache.

3) Cache - Replacement policies

- LRU (Least recently used)
- FIFO (First - In, First - Out)
- Random replacement.

4) Write policies

- Write Through :- Data is written to both the cache and the main memory simultane
- Write Back :- Data is written only to the cache initially, and the main memory is updated later when the cache line is ericted.

* Memory operations.

- Two basic operations are :

i) Load (Read) / Fetch
ii) Store (Write)

★ Write (Store)

★ Read (Load)

- A write operation refers to the process of storing data in memory within a computer system.
- The write operation is essential because it allows data to be stored either temporarily or permanently.
- Write operation helps organize data effectively
- Processor sends the address of the desired location to the memory, together with the data to be written into the location

★ Read (Load)

- The read operation is important because it helps us get data from files or memory.
- Memory reads the data stored at the address and sends them to the processor.
- Memory content remains unchanged.
- processors sends the address of the desired location to the memory and request that its content be read.

- Memory locations.
  - A memory location is a specified place in the computer's memory where data is stored.
  - Each memory location can hold a certain amount of data, typically measured in bytes.
  - For example, if you have an array of numbers, each number is stored in a different memory location.

- Memory Addresses.
  - A memory address is a unique identifier for a memory location.
  - It's like an address for a house; it tells the computer where to find a specific piece of data in memory.
  - Each memory address points to a specific location where data is stored.
  - It allows the CPU to access that data quickly.

- Instruction and Instruction Sequencing.

- Instruction.
  - An instruction is a binary-coded operations that tells the CPU to perform a specific task.
  - Instructions can include operations like arithmetic calculations (addition, subtraction), data movement (loading data from memory) or control operations
  - Each instruction is represented in machine code, which is the language that the CPU understand directly.

- Instruction sequencing.
  - This refers to the order in which instructions are executed by the CPU.
  - The sequence is crucial because the outcome of a program often depends on the order of operations.
  - proper instruction sequencing is essential for the logic of the program to function correctly.
  - If instructions are executed out of order, it can lead to unexpected results or errors.

- Register Transfer Notation (RTN) :-

Register Transfer Notation is a way to describe the operations of a computer's registers and the data transfer between them.

[some key concept of RTN]

1) Registers :- Registers are small, fast storage locations within the CPU that hold data temporarily during processing.

2) Transfer operations :- RTN focuses on the transfer of data between registers, memory and input/output devices.

examples :- R1 ← R2

It means transfer the data of Register R2 to register R1

3) Arithmetic and Logic operations :-
RTN can also represent arithmetic and logic operations.

examples :-

① R1 ← R2 + R3

It means add the register R1 and Register 3 and store the result in register 1

② R1 ← R2 AND R3

It means perform a logical AND operation between register R2 and R3 and store result in register R1

4) Control signals :- RTN can include control signals that dedicate how operations are execute.

Assembly language Notations.

- Assembly language is a low-level programming language that is closely related to machine code, which is native language of a computer's CPU.
- It provides a way to write programs using symbolic instructions and is specific to a particular computer architecture

Some assembly language notations :

1) Mnemonics :- These are symbolic names for machine instructions. Each mnemonic corresponds to a specific operations, for ex:
   - Mov :- more data from one location to another
   - ADD :- Add two values
   - SUB :- Subtract one value to another

2) Operands :- These are the values or addresses that the instructions operate on.
   Operands can be :
   - Registers (e.g. 'Ax', 'Bx')
   - Memory Addresses (e.g. '[1000h]')
   - Immediate values (e.g. '5', '0x1A')

3) Labels :- Lables are used to mark a location in the code, often for use in branching instructions. for example;
- 'start:' - It indicates a point in the program that can be jumped to.

4) Comments :- Comments are used to explain the code and are ignored by the assembler. They often start with a semicolon (;) for example:
- '; This is a comment'

Basic Instruction Types.

- Each instruction of the CPU contain specific information fields, which are required to execute it.

Basic instruction types :-

1) Data movement instructions :-
These instruction more data from one location to another

- MOV :- Transfer data from one memory location to another
- PUSH :- Places data onto the stack
- POP :- Removes the data from the stack

2) Arithmetic Instructions :-
It performs mathematical operations on data.

- ADD :- Add two values.
- SUB :- Subtracts one value to another
- MUL :- Multiple two values
- DIV :- Divides one value by another

3) Logical Instructions :-
It performs logical operations on binary data.
- AND :- performs a bitwise AND operation
- OR :- performs a bitwise OR operation
- XOR :- performs a bitwise exclusive OR oper-
- Not :- ~~performs~~ Inverts the bits of a value

Instruction Execution

- There are 2 phases for Instruction execution

1) Fetch Phase :- The instruction is fetched from the memory - location and placed in the IR
2) Execute phase :- The contest of IR is determine which operation is to be performed. The specified - operation is then performed by the processor.

- Addressing Modes.

  - The different ways in which the location of an operand is specified in an instruction is called as Addressing Mode

    - Generic Addressing Modes:
      - Immediate mode
      - Register mode
      - Absolute mode
      - Indirect mode
      - Index mode
      - Base with index
      - Base with index and offset
      - Relative mode
      - Auto-increament mode
      - Auto-decreament mode

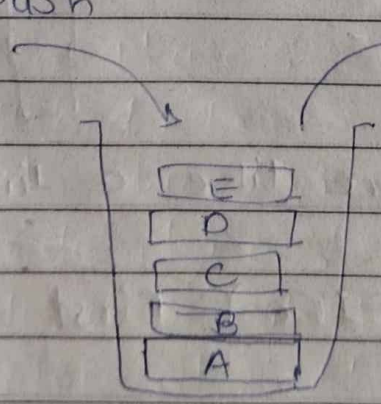
- Assembler directives.

  - Assembler directives are special instructions in assembly language that tell the assembler how to process the program, but they are not actual machine instructions that get executed by the CPU.
  - They help in organizing the code and managing data.
  - Some common assembler directives

Basic Input / Output Operations:

- Input and output operations are essential for a how computer interacts with the outside world.
- When we talk about input operations, then we are see how computer receives data. for ex :- keyboard, mouse & scanner etc.
- When we talk about output operation it involves sending processed data from the computer to output devices. for ex :-
- In conclusion input operations focuses on how a computer collets and processes data from users and output operations deals with how it presents that data back to users through various devices.

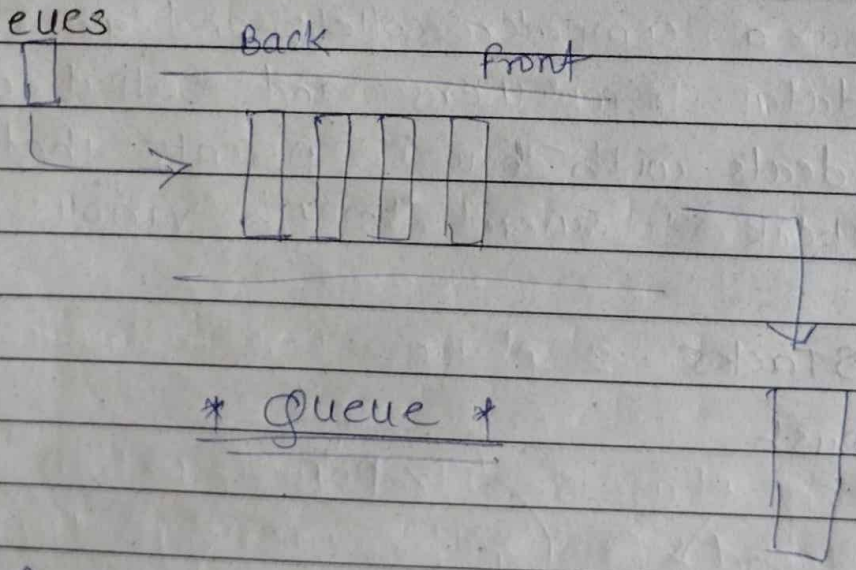
- STacks



* Stack *

- A stack is like a stack of plates.
- You can only add or remove the ~~top~~ top plate.
- Stack use (LIFO) Last in, first out principle.
- Two basic operations that can be performed on stack are PUSH and Pop, which add and remove elements from top of the stack respectively.
- It easy to implement using arrays ~~an~~ or linked lists.
- Useful for storing temporary data.

- Queues



- A queue is more like a line at a coffee shop.
- Queue used (FIFO) first in first out principle
- The first item added to the queue is the first one to be removed.

- queues are commonly used in scenarios like managing tasks in a printer or handling requests in a web server.

III

I/O Organization

Date
Page

- Accessing IO devices.

- An I/O interface is crucial for enabling communication between a computer and various input devices such as keyboard, mouse and scanners.
- It's main role is to convert the signals from these format that the computer can Undeestand and process.
- When you use an input device, it sends signals to the I/O interface.
- The I/O interface takes this signal and translates it into binary data.

- I/O interface for an input device.

- An interface for an input device is basically the way that the device communicates with the computer.
- When you use something like a keyboard or a mouse, it sends signals to the computer to tell it what you want to do.
- In short, the interface is ceucial for allowing us to interact with computee effectively.

- ~~Subroutines~~

- Memory - Mapped I/O

  - When I/O devices and the memory share the same address space. The arrangement is called memory - mapped I/O
  - Any machine instruction that can access memory can be used to transfer data to an I/O device
  - Some processors have special In and Out instructions to perform I/O transfer

- Program - Controlled I/O

  - I/O devices operate at speeds that are very much different from that of the processor.
  - keyboard, for example, is very slow.
  - It needs to make sure that only after a character is available in the input buffer of the keyboard interface; also, this character must be read only once.

* Three major mechanisms:

- program-controlled I/O - processor polls the device
- Interrupt
- Direct Memory Access (DMA)

* Interrupts *

- Interrupt hardware.

- Interrupt hardware is a crucial component in computer systems that allows the CPU to respond to events or signals from input devices or other hardware.
- This signals tells the CPU that it needs to pause its current tasks and handle the new event.
- This process allows the computer to be responsive and efficient, as it can handle multiple tasks and respond to events in real-time.

• Enabling and disabling interrupts.

- Enabling interrupts.
  - When interrupts are enabled, the processor can respond to interrupt signals.
  - This means that if an interrupt occurs, the current program execution will be paused and control will be transfered to a special routine known as interrupt service Routine (ISR)
  - Typically done by setting a specific bit in a control register of the processor.
  - The interrupt be disabled / enabled in the interrupt-service routine

- Disabling interrupts.
  - Disabling interrupts prevents the processor from responding to any interrupt signals.
  - Similar to enabling, this is done by clearing a specific bit in the control register.
  - An instruction like Disable interrupts is often used for this purpose

Interrupt priorities schemes.

- Interrupt priority schemes are important for managing multiple interrupts in a system.
- ensuring that the most critical tasks are handled first.
- choosing the right interrupt priority scheme depends on the specific requirements of the system.

different type of interrupt priorities scheme:

1) Fixed priority scheme :-
   - In a fixed priority scheme, each interrupt source is assigned a specific priority level.
   - Simple to implement and understand.

2) Dynamic priority scheme :-
   - In a dynamic priority scheme, the priority of interrupts can change based on certain conditions.
   - More flexible and can adapt to changing conditions
   - More Complex to implement and manage

3) Round robin scheme
   - In this scheme, interrupts are serviced in a circular order based on their priority levels.
   - fairness is handling interrupts, preventing starvation.

4) priority inheritance
- In system where lower-priority tasks may hold resources needed by higher-priority tasks.
- Reduces the risk of priority inversion
- It can lead to increased complexity in the scheduling of tasks.

Direct memory Access (DMA)

- Direct memory Access is a features that allows certain hardware subsystems to access the main system memory independantly of the CPU.
- The primary goal of DMA is to improve the overall performance of the system.
- DMA controller manages the data transfer between memory and pheripherals
- DMA is a crucial features in modern Computing that enhances system performance

- Bus Arbitration
  - Bus arbitration is a process used in computer systems to manage access to a shared communication pathway.
  - Bus arbitration is an essential component of computer architecture.
  - Need to establish a priority system.
  - ~~Two approaches~~ : ~~Centralized & distributed~~
  - Bus arbitration is the method used to control access to a shared bus in a computer system, ensuring that only one device can use bus at a time. ~~When~~
  - When multiple devices need the bus simultaneously, arbitration decides which device gets control.
  - In simple meaning, Bus arbitration is the process used to manage which device gets control of a shared bus in a computer system, ensuring that only one device can use the bus at a time
  - there are two approaches of bus arbitration:-
    1) Centralized :- In centralized ~~abr~~ arbitration a single bus arbiter performs the required arbitration

    2) Distributed :- In distributed arbitration, all devices participate in the selection of the next bus master.

- Interface circuits :- parallel port, serial port

- Interface circuits are used to connect the computer's internal system with external devices.
- I/O interface connect an I/O devices to a computer buses, like address bus, Data bus, control bus.
- There are two types of interface circuits:
  - → i) parallel port
  - → ii) serial port

i) parallel port.

- A parallel port transmits multiple bits of data at the same time using multiple data lines
- A parallel port transfers data in the form of a number of bits, typically 8 or 16 b
- parallel port transfers fast data for short distances.
- commonly used in printers, scanners, etc.
- keyboard is connected to a processor using a parallel port.

ii) Serial port:

- A serial port transmits data one bit at a time over a single data line.
- Serial port is used to connect the processor to I/O devices.
- Serial port transfer the data slowly compared to parallel port.
- Used for long distance communication.
- Commonly used in connecting modems, mice, and older pheripherals.

Standard I/O interface :

- I/O devices is connected to a computer using an interface circuit.
- A standard approach is the to develop standard interfaces and protocols.

i) processor bus — It links CPU to memory components

- The processor bus connects the CPU to the systems memory and other components.
- It's main purpose is makes communication between the CPU, memory and other hardware components within the computer system
- processor bus is defined by the signals on the processor chip.
- Which devices required high speed ~~data~~ connection to the processor that devices are directly connected to processor bus.

- Because of electrical reasons only a few devices can be connected directed to the processor bus.
- Components :
  - Address Bus :- Carries memory address.
  - Data Bus :- Transfer data between CPU and memory.
  - Control Bus :- Carries control signals to manage operations.

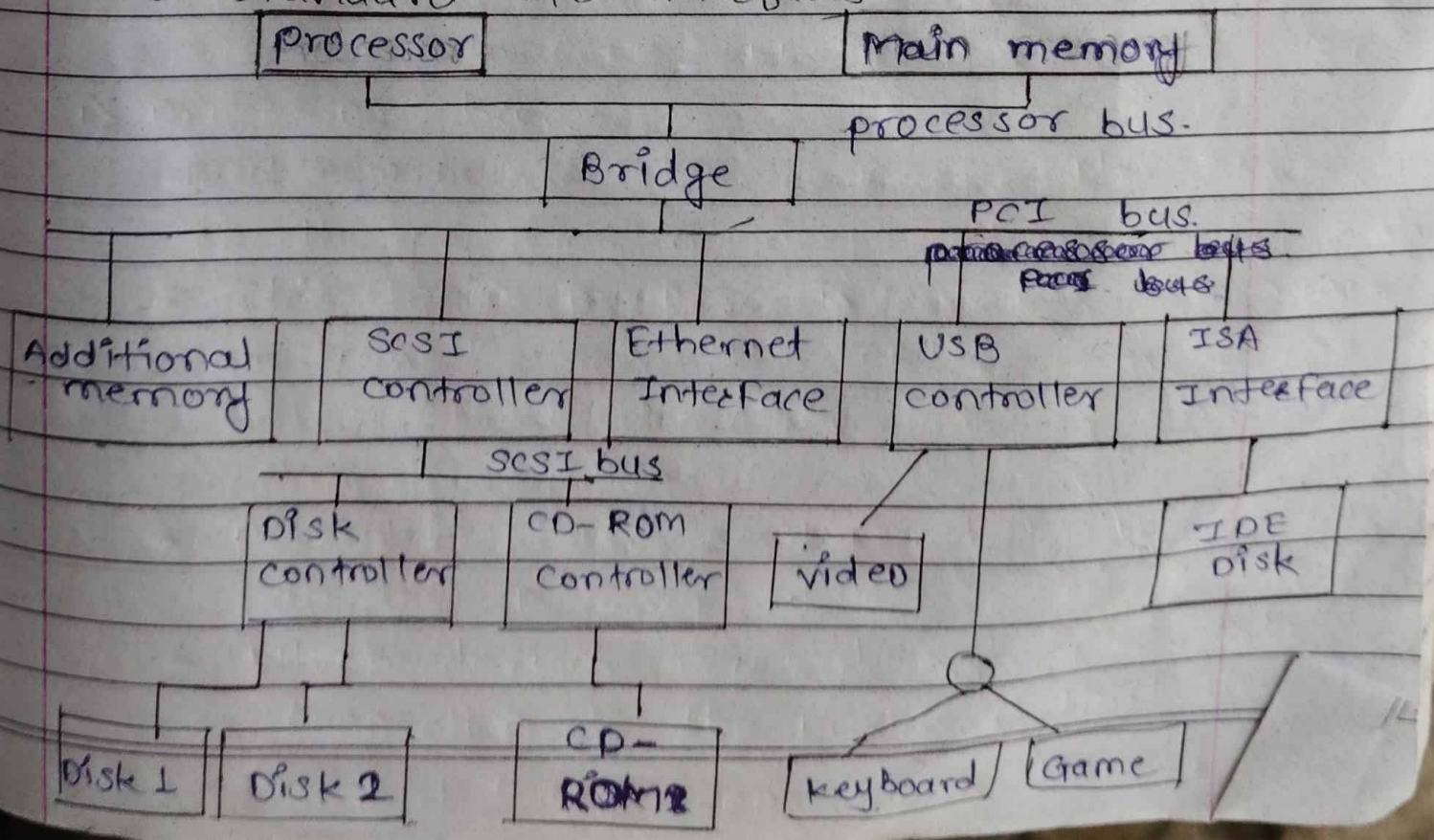
A high speed interface for adding expansion cards

2) PCI Bus (pheripheral Component Interconnect)
- PCI bus is a high-speed parallel bus standard for connecting pheripherals like sound cards, network cards and storage devices.
- PCI bus provides provides a fast, flexible connection for expansion cards.
- It can transfer data at high speed upto 533 MB/s.
- Devices a can be added or removed without shutting down the system.
- It is a low cost bus.
- PCI bus is totally depent on processor
- PCI bus supports three independant address spaces : Memory, I/O, and configuration

It connects multiple devices like hard-drives and printers to the system

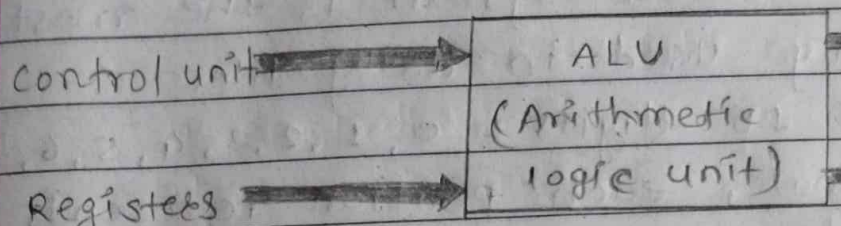
3) SCSI Bus (small computer system Interface)
- SCSI Bus is a standard for connecting multiple devices (like printers, scanners) to a computer.
- The main purpose of SCSI Bus is "to connect and transfer data between multiple peripherals and the computer".
- It can connect multiple devices upto 7 or 15 in a daisy chain.
- SCSI offers faster data transfer compared to parallel ports.
- Mainly used in servers, high performance storage devices. and older systems.
- Data transfer on the SCSI bus are always controlled by the ~~ini~~ target controller

- Standard I/O interfaces



4. Arithmetic and Logical Unit computation

- ALU (Arithmetic logic unit) Inputs and outputs.



- Control unit :- It sends control signals to the ALU, telling it what operation to perform (addition, subtraction, etc).

- Registers :- Register store operands (input values) for the ALU and also hold the result After ALU operations. They can also hold Addresses, and instructions.

- Flags :- Flags are used to indicates things like whether the result is zero, negative or whether an overflow/ underflow occured.

- Number systems.

System	Base	Digits
Binary	2	0 1
Octal	8	0 1 2 3 4 5 6 7
Decimal	10	0 1 2 3 4 5 6 7 8 9
Hexadecimal	16	0 1 2 3 4 5 6 7 8 9 A B C D E F

1) Decimal Number system (Base 10)

- Decimal number system is the most commonly used in daily life.
- It uses 10 digits: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
- In decimal, each position represents a power of 10 (starting from right to left)
- Example:
  - Number = 245 then,

$$2 \times 10^2 + 4 \times 10^1 + 5 \times 10^0$$
$$= 2 \times 100 + 4 \times 10 + 5 \times 1 = 245$$

2) Binary Number system (Base 2)

- Binary Number system is used by computers and digital systems.
- It uses 2 digits. 0 and 1. Each binary digit is called bit.
- In binary, each position represents a power of 2
- Example:
  - Number: 1101 (binary)

$$1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$$
$$= 1 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times 1 = 13 \text{ (decimal)}$$

3) Hexadecimal Number system

- Hexadecimal number system is used in programming and computer science
- It uses 16 symbols : 0-9 for values 0-9 and A-F for values 10-15.
- Example :

  • Number : 2F (Hexadecimal)

  $2 \times 16^1 + 15 \times 16^0 = 47$ (decimal)

  • the digit 2 represents $2 \times 16^1 = 32$ and F represents $15 \times 16^0 = 15$.
  • Total : 32 + 15 = 47 (decimal)

4) Octal Number system

- Octal number system is used in computing to represent binary values.
- It uses 8 digits : 0 1 2 3 4 5 6 7
- In octal, each position represents a power of 8
- example :

  • Number : 47 (octal)

  $4 \times 8^1 + 7 \times 8^0 =$

  $= 4 \times 8 + 7 \times 1 = 39$ (decimal)

• Hardwired control unit •

- TO execute instructions, processor needs a way to create control signals in proper sequence.
- A Hardwired control unit is a type of control unit used in a computer's central processing unit (CPU) to manage the execution of instructions.
- It is a method of generating control signals with the help of finite state machines (FSM)
- Hardwired control unit can operate at high speed but less flexible.
- Faster execution because fixed logic circuit
- " Only essential control signals are generated by the hardware in the hardware control unit " makes it faster.

• Microprogrammed control unit •

- A microprogrammed contro unit is a type of control unit used in computer's CPU to manage the execution of instructions but, it uses software (i.e. microprograms) to generate control signals.
- It uses microprograms to generate control signals instead of fixed hardware circuits, make's it more flexible.

- The micro routine for all instructions in the instructions set of a computer are stored in a special memory called the program stored.
- Microprogrammed control unit is easier to design and manage compare to hardwired units.
- It is slower because it requires memory access to fetch the microprograms

* Integer Representation *

- Sign and Magnitude

- In sign magnitude representation, the first bit represents the sign of the number.
  - 0 - means positive
  - 1 - means Negative

- example :-

B				Values represented	
b_3	b_2	b_1	b_0	1's complem-	2's complement
0	1	0	0	+4	+4
0	0	1	1	+3	+3
0	0	1	0	+2	+2
0	0	0	1	+1	+1
0	0	0	0	+0	+0
1	0	1	1	-4	-5
1	1	0	0	-3	-4
1	1	0	1	-2	-3
1	1	1	0	-1	-2
1	1	1	1	-0	-1

- One's complement representation

+0 = 0000	-0 = 1111
+1 = 0001	-1 = 1110
+2 = 0010	-2 = 1101
+3 = 0011	-3 = 1100
+4 = 0100	-4 = 1011
+5 = 0101	-5 = 1010
+6 = 0110	-6 = 1001
+7 = 0111	-7 = 1000

- 2's complement representation

+0 = 0000	
+1 = 0001	-1 = 1111
+2 = 0010	-2 = 1110
+3 = 0011	-3 = 1101
+4 = 0100	-4 = 1100
+5 = 0101	-5 = 1011
+6 = 0110	-6 = 1010
+7 = 0111	-7 = 1001
	-8 = 1000

- Addition using 2's complement

- Algorithm :-

step 1 :- Converts the numbers into binary
step 2 : Add the binary numbers bit by bit
step 3 : If there is carry then ignore it.
step 4 : If the most significant bit is 0 then result is positive otherwise negative

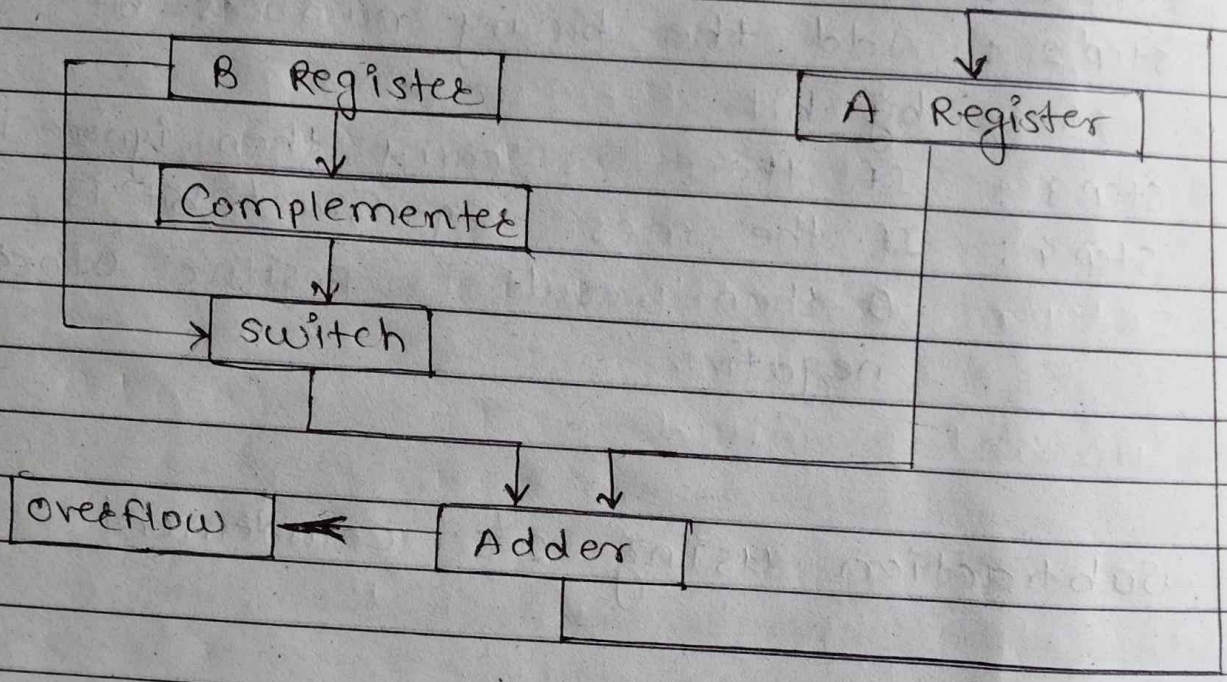
- Subtraction using 2's complement

- Algorithm :-

step 1 :- Convert the numbers into binary
step 2 :- find 2's complement of number to be subtracted
step 3 :- perform addition
step 4 :- If the most significant bit is 0 then result is positive otherwise negative

- Block diagram of Hardware for addition and subtraction.

Flowchart :-



* Explain the operations using flowchart

	Operations	Description
1>	A + B	Both A and B are added to adder for addition
2>	A - B	B is passed through complementer and B is pass through switch
3>	A + (-B)	A and (-B) are added in adder B is initialized as (-B)
4>	-A + (B)	A and B is initialize as -A and -B and added to adder
5>	-A + B(-B)	A and B is initializes as -A and -B and added to adder
6>	(-A) - (-B)	-B is pass through complementer and becomes +B then add to adder

Note :- If '+' is operation between two operands then we do not use complementers.
- If '-' is operation between two operands then we have to use complementer for all negative operators.

- examples :-

1) $(1001)_2 - (0100)_2$

→

A = 1001

B = 0100

∴ A + (-B)

(-B) = 1011 → 1's complement

+ 1

1100 → 2's complement

A + (-B) = 1001

1100

[1]0101

Carry ignore

2) $(0110)_2 - (1011)_2$

→

A = 0110
B = 1011

∴ A + (-B)

(-B) = 0100 → 1's Complement
+ 1
0101 → 2's Complement

A + (-B) = 0110
0101
1011 → carry is not negative hence get 2's complement

0100 → 1's complement of answer
+ 1
Ans :- 0101 → 2's complement

3) (1101) - (1010)

→

A = 1101
B = 1010

A + (-B)

-B = 0101 - 1's complement
+ 1
0110 → 2's complement

$A + (-B) = 1101 + 0110$

$= 1\,1\,0\,1 + 0\,1\,1\,0$

$= \boxed{1}\,0\,0\,1\,1$

carry ignored.

4) Subtract 6 from −4 using 2's complement

→ ∴ A + (−B)

$(4)_{10} = (0100)_2$

A = −4 = 1011 → 1's complement

+ 1

1100 → 2's complement

B = 6 = 0110

(−B) = 1001 → 1's complement

+ 1

1010 → 2's complement

$A + (-B) = 1100 + 1010$

$= \boxed{1}\,0\,1\,1\,0$

Ignore carry.

Ex 5) 55 - 12

→

$(55)_{10} = (110111)_2$

$(12)_{10} = (001100)_2$

A + (-B)

(-B) = 110011 → 1's complement

+ 1

110100 → 2's complement

A + (-B) = 110111

+ 110100

1 101011

ignore

6) 2 + 3

→

A = 2 $(0010)_2$

B = 3 $(0011)_2$

A+B = 0010 (2)

0011 + (3)

0101 (5)

7) 0100 + 1011

→

A = 0100

B = 1011

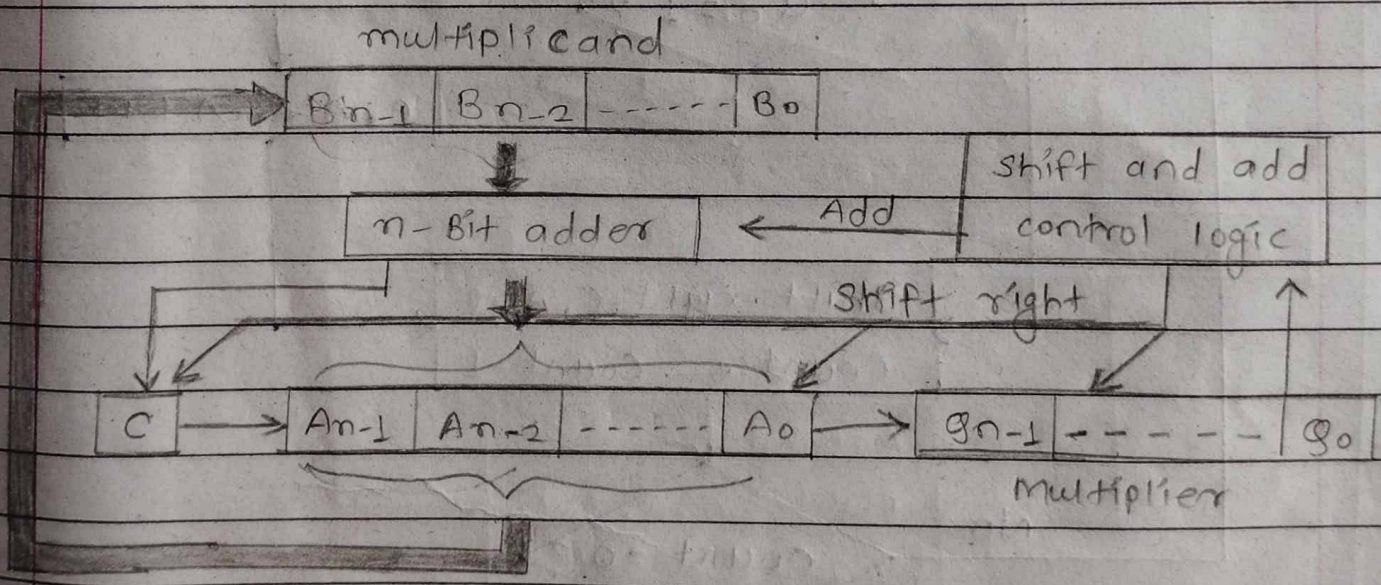
A+B = 0100

1011

1111

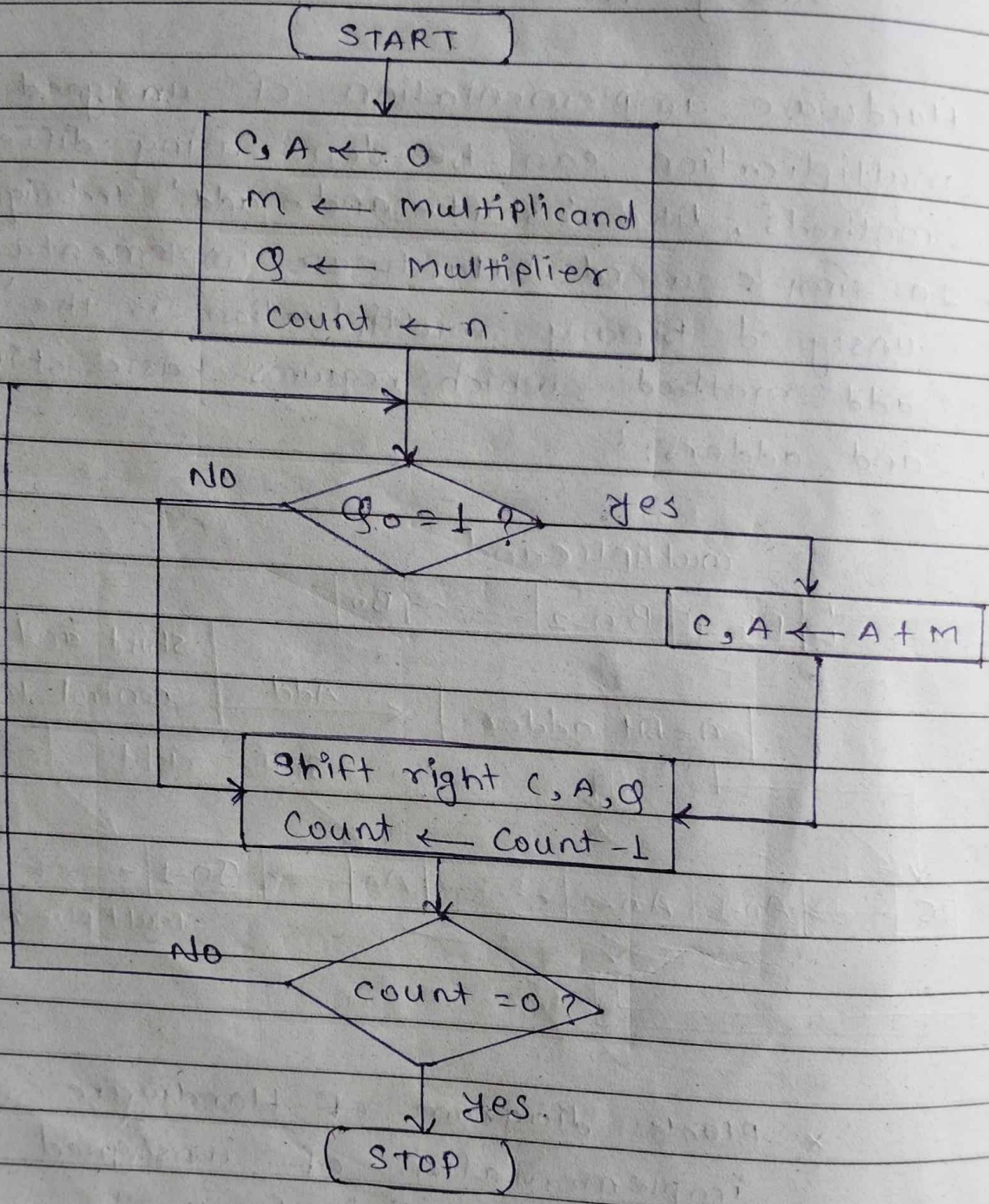
- Hardware implementation of unsigned bindry multiplication.

- Hardware implementation of unsigned binary multiplication can be done using different methods, like 'shift-and-add' technique
- In simple words Hardware implementation of unsigned binary multiplication is the 'shift and add' method, which requires basic shift registers and adders.



* Block diagram of Hardware implementation of unsigned binary multiplication *

- Flowchart :-


START
C, A ← 0
M ← Multiplicand
Q ← Multiplier
Count ← n
Q0 = 1 ?
No
yes
C, A ← A + M
Shift right C, A, Q
Count ← Count - 1
Count = 0 ?
No
yes
STOP

Steps:

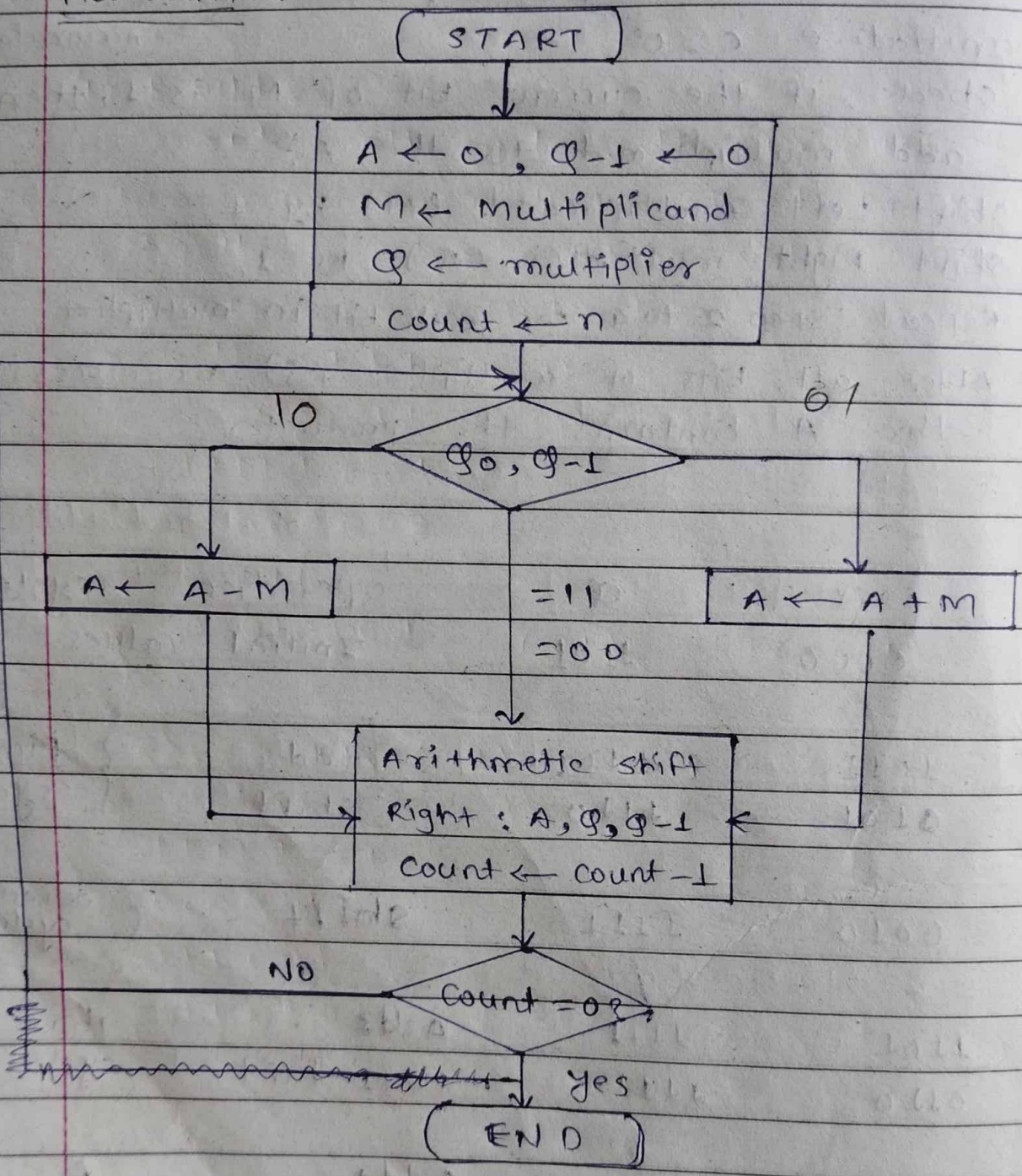
1) ~~Set~~ the multiplier (Q) and Multiplicand (M) is loaded in two registers, Register A = 0. (Accumulator)
2) Initialize C = 0.
3) Check, if the current bit of M is 1, then, add multiplicand to the A.
4) Shift Left multiplicand (M) by 1.
5) Shift Right multiplier (Q) by 1.
6) Repeat step 2 to 4 for each bit in multiplier.
7) After all bits of multiplier (Q) are process, the A contains the productor

C	A	Q	operation	cycle
0	0000	1101	Initial values	
0	1011	1101	Add	first cycle
0	0101	1110	shift	first cycle
0	0010	1111	shift	second cycle
0	1101	1111	Add	third cycle
0	0110	1111	shift	third cycle
1	0001	1111	Add	fourth cycle
0	1000	1111	shift	fourth cycle

IMP

- Booth's Algorithm for two's complement multiplication

Flowchart :-



Booth's algorithm for two's complementary multiplication

- **Steps :-**

1] Initialize Registers :
- Let the multiplicand be stored in a register A
- Let the multiplier be stored in a register Q
- A third register, Q_{-1} is used to store an extra bit (usually initialized to 0)
- Another register M holds the multiplicand.
- Start with the initial values and set the number of cycles based on the size of numbers (e.g. for 4-bit multiplier, 4-cycles).

2] check pairs of Bits :
- Look at current Q_0 (the least significant bit of multiplier) and Q_{-1} bit (extra bit)
- Depending on values of Q_0 and Q_{-1}, take one action :
  - If Q_0 and Q_{-1} = 01 : Add the multiplicand to the value in register A.
  - If Q_0 and Q_{-1} = 10 : Subtract multiplicand from the value in register A
  - If Q_0 and Q_{-1} = 00 or 11 : Do nothing.

// eat 5 stars

3] Arithmetic shift :
- After performing the addition or subtraction, shift all the bits in A, Q and Q_{-1} to right by 1 bit. This is called Arithmetic shift.

- Continue this process for each bit of multiplier, repeating the steps for number of cycles required (e.g. 4 cycles for a 4-bit numbers)

4) Result:

After all cycles, the product of the two numbers will be stored in A and Q together. This combine values gives final result of multiplication

ex:-1) 0111 (7)
× 0011 × (3)
11111001
00000000
000111
00010101 (21) (7×3=21)

2) 0111 (7)
× 1101 × (-3)
11111001
0000111
111001
11101011 (-21)

A	Q	Q-1	M	Initial values
0000	0011	0	0111	Initial values
1001	0011	0	0111	A ← A-M } first
1100	1001	1	0111	shift } cycle
1110	0100	1	0111	shift } second cycle
0101	0100	1	0111	A ← A+M } Third
0010	1010	0	0111	shift } cycle
0001	0101	0	0111	shift } fourth cycle

5 Memory systems.

- Basic Concepts of Memory
  - Memory in computers is used to store data and instructions that the CPU needs to execute
  - It can be temporary or permanent
  - Execution speed of programs is highly dependent on the speed with which instructions and data can be transferred between the processor and the memory.
  - Ideally, the memory would be fast, large and inexpensive
  - Unfortunately, it is not possible to meet all three of these requirement simultaneously.
  - If speed and size are increased then always cost will be increased.
  - Types of memory :
    - primary memory (RAM) :-
      - It is a temporary memory of computer
      - It is a faster memory.
      - It loses data when the power is turned off
    - Secondary memory (ROM) :-
      - It is a permonent memory of computer
      - It is slower than RAM
      - It stores data for long-time.
  - One way to reduce the memory access time is to used a cache memory.

- **Computer memory system overview:**

- Characteristics of memory system:

1) Location
   - Where the memory is physically located in syste
   - Primary memory: Located close to the CPU
   - Secondary memory: Located further away.

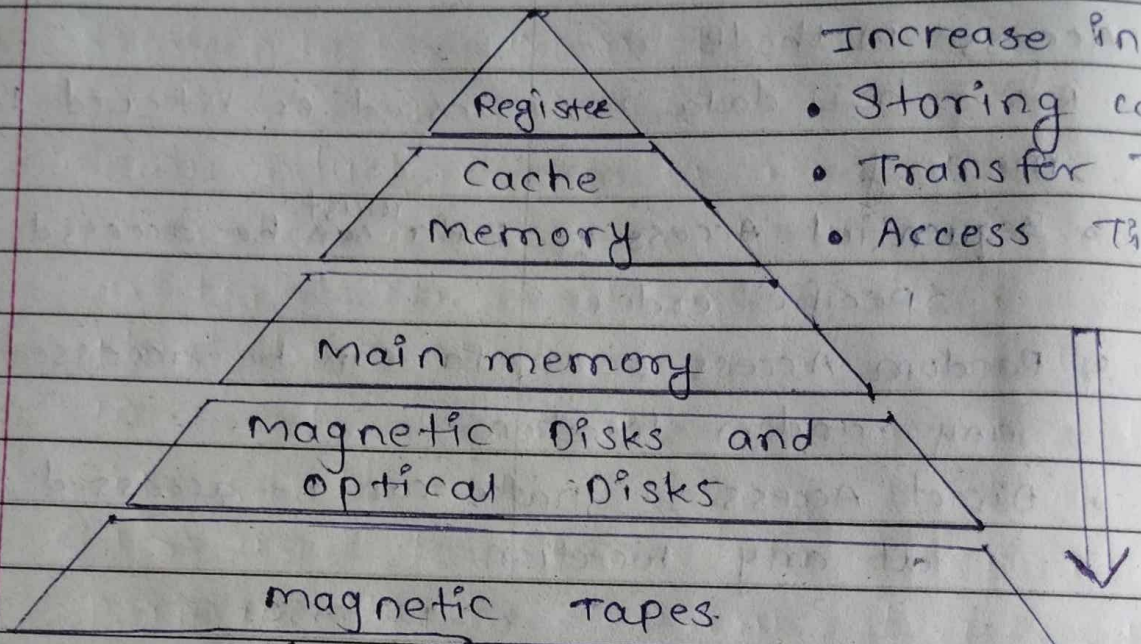
2) Capacity
   - Refers the amount of data that can be stored
   - Primary memory (RAM) :- Typically in the range of GB's
   - Secondary memory (ROM) :- Can be in the range of TB's
   - Cache memory :- Smallest in size, usually in the range of kB's or MB's

3) Unit of Transfer
   - The amount of data transferred at one time
   - cache: Transfers data in small blocks (e.g., 64 bytes)
   - RAM: Typically transfers data in larger blocks (e.g. 128 / 256 bytes)
   - Storage devices :- Transfers larger chunks of data (e.g. MBs or GBs)

4) Access Method:
- Refers how data is accessed or retrieved from memory
  - Sequential Access :- Data must be accessd in specific order
  - Random Access :- Data can be accessed at any random location.
  - Direct Access :- Data can be accessed directly at any location

5) Performance :
- Refers how quickly and efficiently the memory system can perform operations.
  - Access Time :- The time it takes to access data from memory
  - Memory cycle Time :- The time redquired to read and write in a memory location
  - Transfer rate :- The rate at which data can be transferred between memory and other system components.

- Hierarchy of Memory :
As one goes down the hierarchy, the following occur :
a) Decreasing cost per bit ;
b) Increasing capacity ;
c) Increasing access time
d) Decreasing frequency of access of the memory by the processor.



* diagram of memory hierarchy *

- Basic Concepts of pipelining

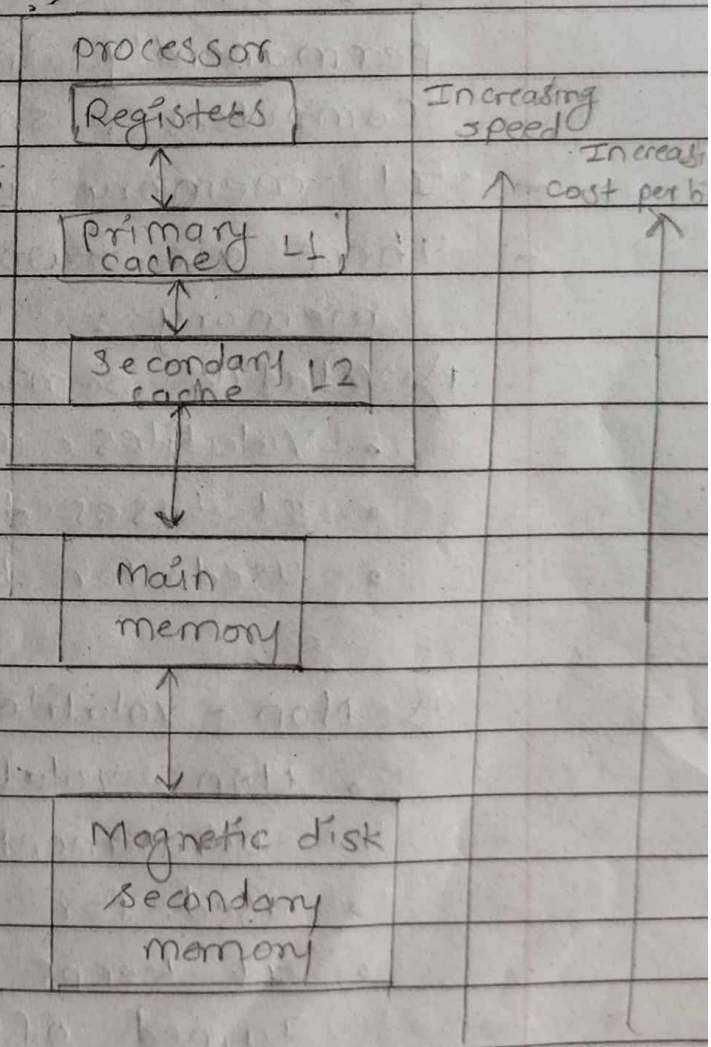
- pipelining is a taking used in computers to make processing faster by breaking down tasks into smaller stages and processing multiple tasks at same time
- It is also known as pipeline processing
- pipelining is an speed up technique where multiple instructions are overlapped in execution on a processor.
- A pipelining is a ~~stages~~ series of stages, where some work is done at each stage in parallel.
- In our real life example it's like an assembly line where different workers do different jobs at simultaneously, making the process faster.

- Stages of pipelining

1. fetch :- Get the instructions from memory
2. Decode :- Figure out what the instruction means
3. Execute :- perform the operation (addition, subtracti
4. memory :- Read and write in memory.
5. Write-back :- write the result back to memory

- Memory hierarchy Theorey :-

- Memory hierarchy refers to the organization of different types of memory in a computer system, arranged from the fastest and smallest to the slowest and largest
- The fastest access is to data held in processor registers
- The fastest memory is very expensive and small.
- The system can access data quickly.
- In simple, This organization improves overall system performance, reducing the time spent waiting for data



* memory hierarchy

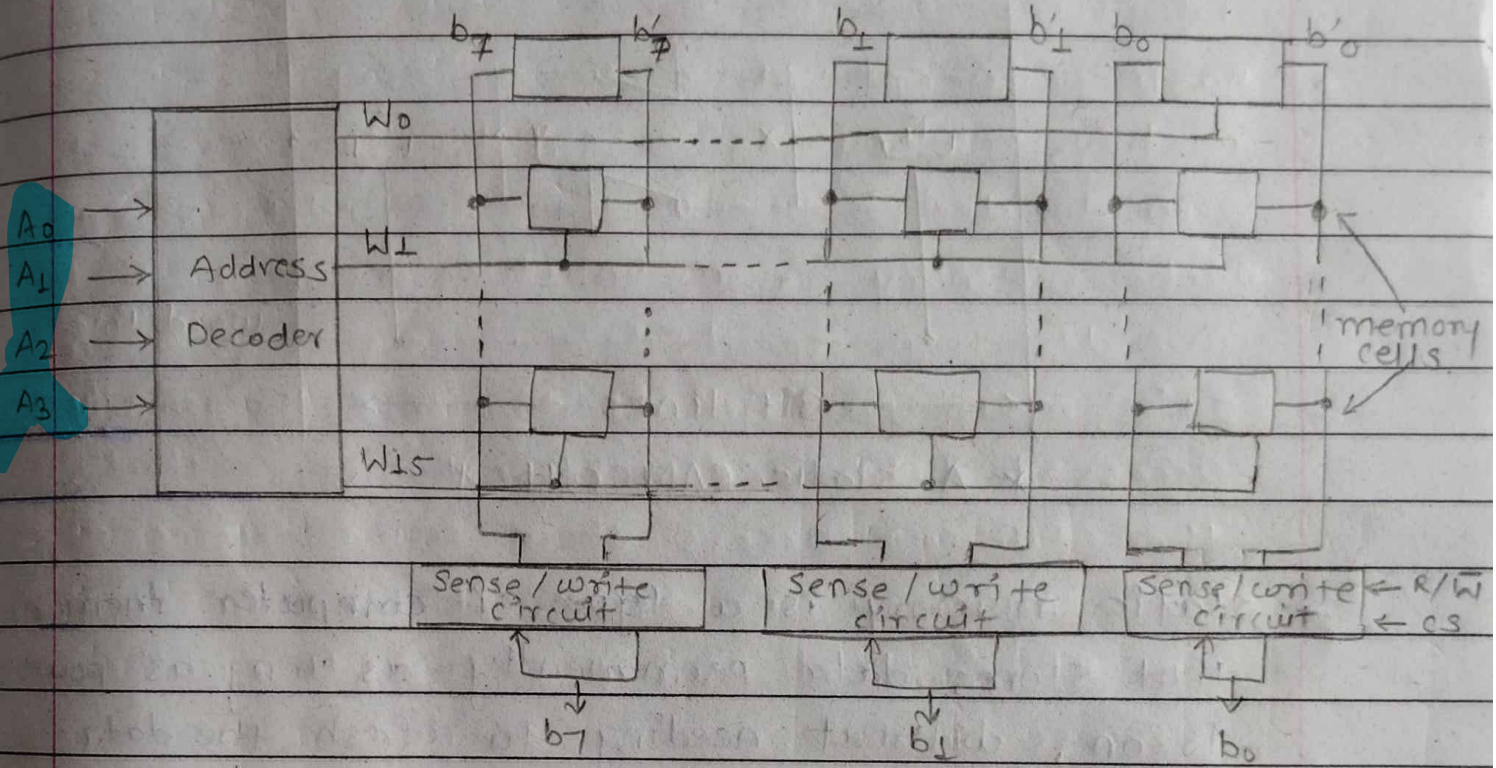
Semiconductor Memory

- Semiconductor memory is a type of computer memory made from semiconductor materials like silicons.
- This type of memory is commonly used in modern computer system due to its speed and efficiency.
- It's used to store data temporarily or permanently in electronic devices., like Computers, smartphones and other gadgets
- It memory is faster and more reliable.
- There are two types of semiconductor memory :

1) Volatile semiconductor memory
   - Volatiles means stores data temporarily.
   - It loses data when power is turned off
   - Used for temporary storage

2) Non - volatile semiconductor memory.
   - Non - volatile means stores data permanently
   - Used for long-term storage of data
   - It keeps data when power is turned off

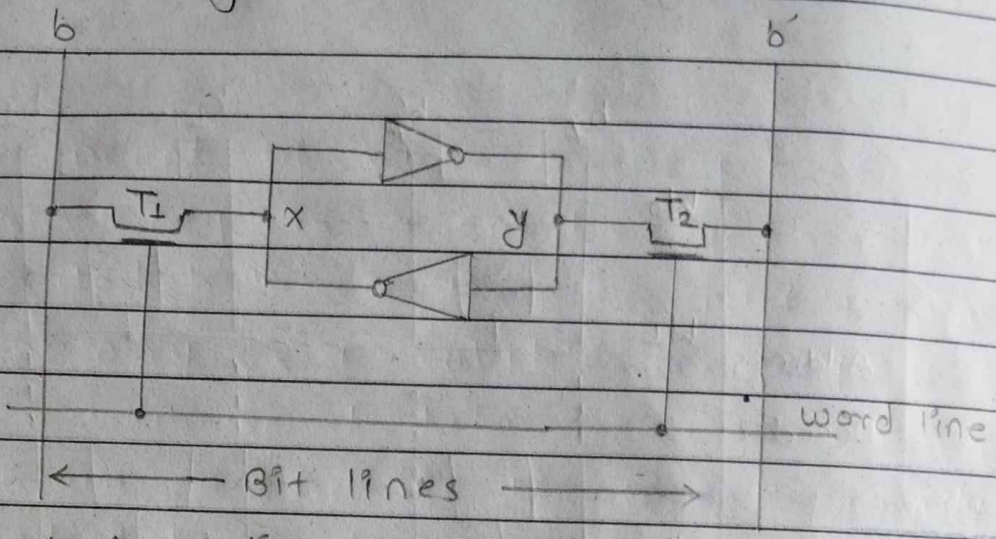
- Internal organization of memory chip.



* Internal organization of memory chip *

- Memory cells are usually organized in the form of an array.
- Each cell is capable of storing one bit of information
- Each row of cells constitutes a memory word and all cells of a row are connected to a common line is word line driven by address decoder on the chip.
- The cells in each column are connected to sense/write circuit by two bit lines.
- Sense/write circuit are connected to the data I/O lines of the chip. During a Read operation, these circuit read the information stored in the cells selected by word line and transmit this information to the output data lines
- During a write operation, sense/write circuit receive input information and store it in the cells of the selected word.

- Static Memory



* A static RAM cell *

- static memory is a type of computer memory that stores data permanently as long as power is on, without needing to refresh the data
- Statics memory is fast, but expensive and power consuming
- Continuous power is needed for the cell to retain its state. If power is interrupted, then the cell's contents are lost.
- The most common example of static memory is SRAM (static Random Access memory)
- static RAM's can be accessed very quickly.
- Static RAMs are fast, but their cells require several transistors.
- static RAM uses more power
- static memory allows for faster random access
- SRAM is used in applications where speed is critical.

- Speed, size and cost

- Speed :- Speed refers to how fast data can be accessed or processed by the system
- Size :- Size refers to how much data the memory or storage can hold.
- Cost :- Cost refers to the price to manufacture or purchase memory and storage

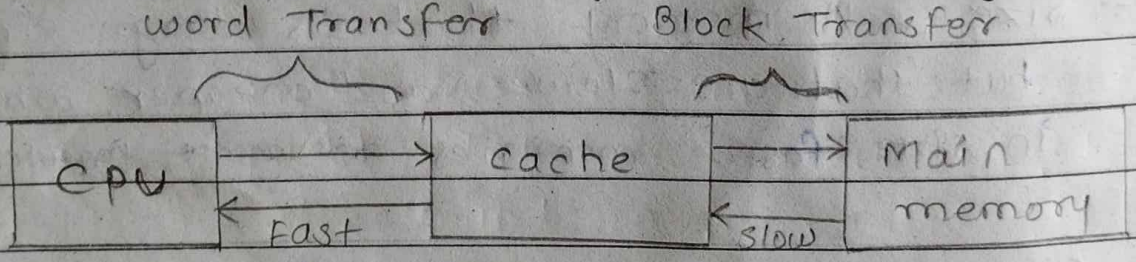
* Trade-off *

- Speed :- Faster memory technologies but expensive and more power consuming
- Size :- A memory that is larger in size but that are slower and cheaper, while smaller in size faster memories ~~are more limited~~
- Cost :- More expensive technologies tends to be faster or more reliable, but you can't always afford them in larger quantities.

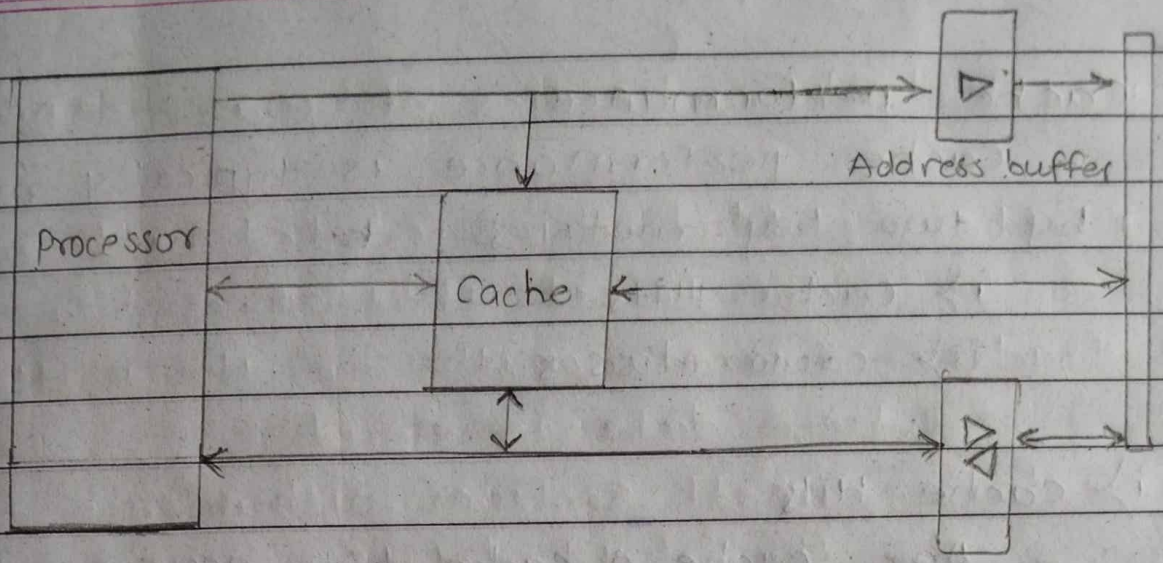
Cache Memory

- Basic concept of cache memory
  - cache memory is a small, high-speed storage area located between the CPU and Main memory (RAM)
  - Cache memory is a volatile memory that provides high speed data access to processor
  - It's main purpose is to store frequently accessed data and instructions so that the CPU can access them much faster
  - Cache memory is essential for speeding up CPU operations by storing frequently accessed data.

- Role of cache memory in processing instructions



  - Faster instruction fetching /access
  - cache boost overall system performance
  - cache enhances speed
  - cache Reduces delay
  - cache helps to maintain smooth and efficient pipeline processing
  - Cache stores Recently accessed instructions for future
  - It reduces the time to access data



* Diagram of cache organization *

- Elements of cache Design

- Cache Addresses
  - Logical
  - physical
- Cache size
- Mapping function
  - Direct
  - Associative
  - Set Associative
- Replacement Algorithm
  - Least Recently used (LRU)
  - First In First out (FIFO)
  - Least frequently used (LFU)
  - Random
- Write policy
  - Write through
  - Write back
  - once
- Line size
- Number of caches
  - Single or two level
  - unified or split

- Cache performance
  - Cache performance is typically measured by two key metrics :
    i) Cache Hits
    ii) cache Misses

i) Cache Hits
- A cache hits occurs when the CPU requests the data or instruction and it is found in the cache
- Because of cache hits CPU can access the data quickly
- Formula :

$$\text{Hit Ratio} = \frac{\text{Number of caches hits}}{\text{Total memory access}}$$

ii) Cache misses
- A cache miss occurs when the CPU request data or instruction, but it not found in cache
- because of cache misses it takes more time to fetch instructions
- Formula :

$$\text{Miss Ratio} = 1 - \text{Hit Ratio}$$

- Secondary storage :
  - Secondary storage refers to non-volatile storage devices used to store data long-term.
  - Primary storage temporarily stores data while computer is on, and secondary storage retains data even when the power is off.
  - Magnetic discs (i.e. hard drive and HDD's) are one of the most common types of secondary storage devices.

- Magnetic Disc
  - Magnetic discs stores data using magnetism
  - They consist of circular disks coated with a magnetic material.
  - Data is read and write from the surface of discs.
  - It have high storage capacity
  - It is reliable for Long-Term storage
  - It is slower than SSD's.
  - The disk system consist of three key parts :
    1) The assembly of disk platters; which is usually referred to as the disk
    2) Disk Drives
    3) Disk controllers

- How digital information is stored in magnetic hard disc
- & physical characteristics of magnetic Discs

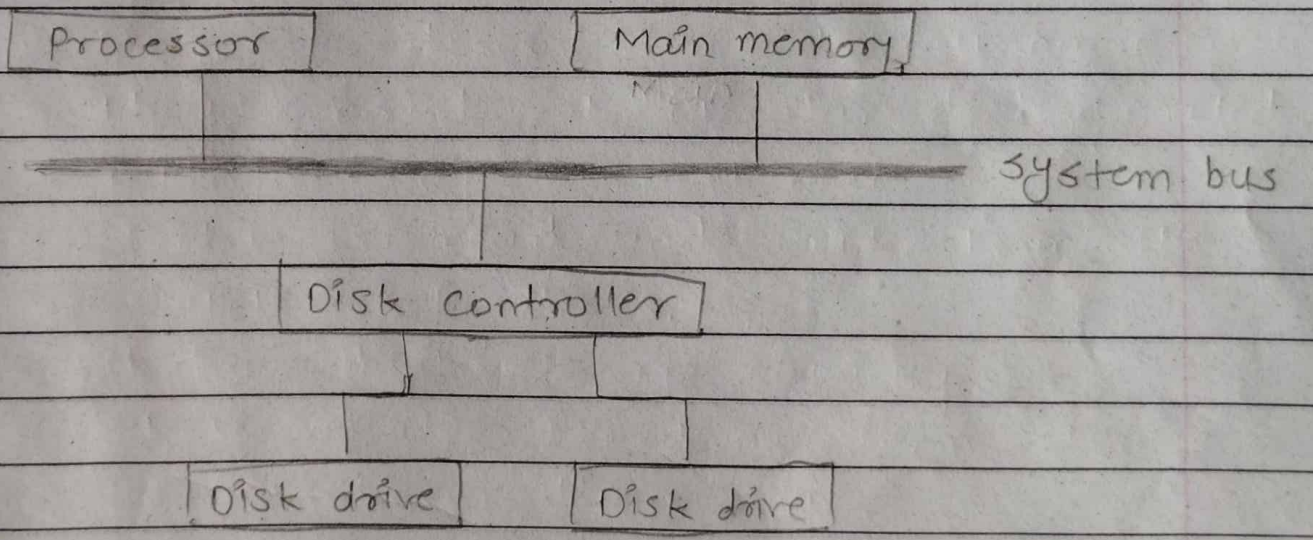
1. platters : flat discs coated with magnetic material that stores data
2. Tracks : Circular paths on the platter where data is stored
3. Sectors : Small sections within each track that hold data
4. cylinders : Stacked tracks for align vertically
5. Read / write heads : Small devices that read or write data by moving across the platters
6. Spindle : The motor that spins the platters at high speed
7. Arm Actuator : The mechanism that moves the read / write heads across the platters
8. Disc Cache : Small memory on the disk to speed up access to frequently used data

★ Magnetic Disc functions

- Store data :- Magnetically stores information on rotating platters.
- Retrieve data / Read data :- Read refers to the process of retrieving or accessing data from the magnetic dis
- Seek : Seek refers to the process of moving the read / write head to the correct track on a magnetic disc

- write data : write's data by changing the magnetization of the disk's surface
- organize data : Arrange data into tracks, sectors, and cylinders.
- Manage data : The disc controller manages read / write operations and data organization.
- Cache data : Temporarily stores frequently accessed data for faster retrieval

• Disc Controller •



- One disk controller may be used to control more than one drive.
- A disk controller that communicates directly with the processor
- The disk controller uses DMA scheme to transfer data between the disk and the main memory.

6. Computer Pheripherals

★ Working of input and output devices.

Input devices

1) keyboard

- keyboard are used for direct interaction between human or computer.
- keyboard allows users to type text, execute commands and control software by pressing keys.
-

• Working •

- keyboard uses a matrix of rows and columns to detect key pressed ~~and computer's operating system~~
- Each key of keyboard is connected to the electrical circuit.

When key pressed, circuit gets complete and send signal to the ~~computer~~ keyboard's controller.

The controller converts the signal into a digital format and send's it to the computer.

Then computer processes the signals and executes corresponding action

2) Mouse

- Mouse is a pointing device used to interact with the graphical user interface of the computer.
- It allows users to move a cursor on the screen.

• Working •

- The mouse uses sensors to detect movement and button presses.
- Mouse sends signals to the computer to move the cursor on the screen.

• optical mouse :- It uses a LED light and senser to detect movement on a surface by capturing changes in the light patterns.

• Mechanical mouse :- It uses a ball that moves in response to surface movement, which is detected by sensors.

- Mouse can be connect via USB, wireless or bluetooth.

✳ Output devices ✳

1) Video Display.

- Video display refers to the process of showing images, graphics or text on a screen.
- Video display converts electrical signals into visible images on the screen

• Working •

1) Signal input :- The computer or device sends a video signals to the screen via a connection like HDMI or VGA

2) Display Technology :-

- CRT (cathode Ray Tube) :- It uses an electron gun that shoots tiny particles (electrons) onto a screen to light up different colors and form an image.

old style monitors

- LCD (liquid crystal Display) :- It uses liquid crystals to control light passing through a ~~screen~~ each pixel. Each picture can show different colors. (red, green and blue)

modern monitors

3) Color and Image :- The screen mixes red, green and blue light for each pixel to form a full image. The image refreshed many times per second to create a smooth and continous display.

2) Flat-panel Display.

- Flat_panel displays are mordern screens.
- Flat_panel display provides a slim, lightweight and space saving screen to display images and videos.

• Working •

- Flat-panel displays works on the pixels. Each pixel contains sub-pixels (red, green and blue) which can adjust their brightness to form various colors. Depending on the technology used (LCD, LED, OLED)

  • LCD :- It uses liquid crystals with backlight for visibility
  • LED :- It improves energy efficiency with LED backlighting
  • OLED :- It provides better contrast and color using organic light-emitting materials.

- The graphics card in the computer sends digital data to the monitor to display images, text and video.

3) printer

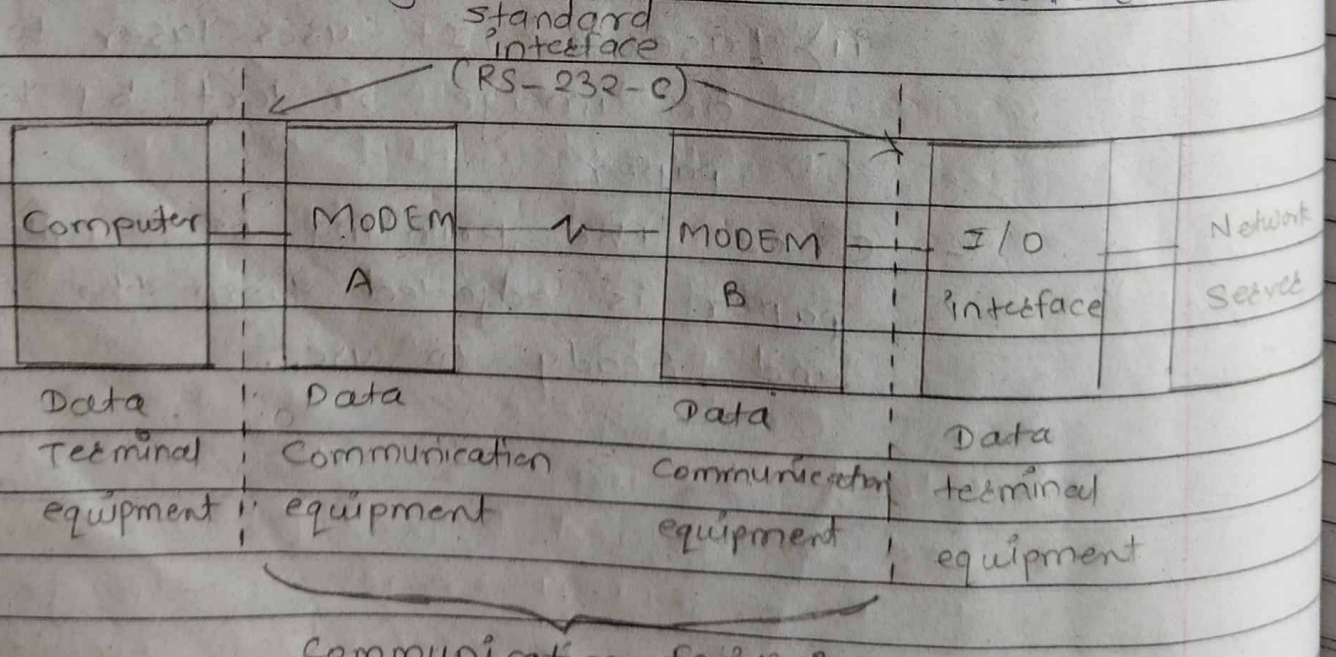
- printer converts digital data into physical copies.
- It produces a physical copy of digital documents and images.

• Working •

- Firstly, printer gets the documents or image data from a computer
- Then printer converts it into printable format.
- The printer pulls paper from the tray using rollers.
- printing images and text :-
  i) Inkjet :- It spreys ink onto paper
  ii) Laser :- It uses laser to create an image or text by heating paper
- Once the printing is completed paper is edjected from the printer and ready to use.

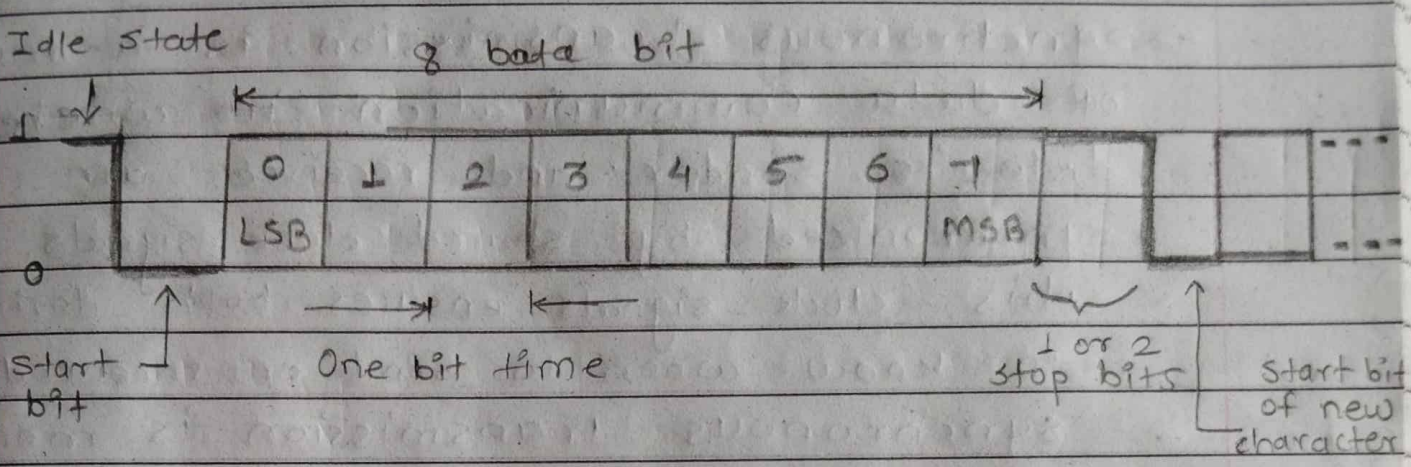
Serial Communication Links.

- Serial communication is a method of sending data one bit at a time.
- It is commonly used for communication between computers and phesipheral devices.
- keyboard and mouse are connected directly to the computer through sesial communication link.
- It transfers data sequentially.
- It uses start and stop bits and its typically slower than parallel communication but simpler and more reliable over long distances.
- There are two types of serial communication :-
  1) Asynchronous Transmission
  2) synchronous Transmission



* Remote connection to network *

- Asynchronous Transmission



* Asynchronous serial Transmission *

- Asynchronous transmission is a method of data communication where data is sent one byte at a time.
- There is no clock signals to synchronize the sender and receiver, so each data packet is sent independantly.
- Asynchronous transmission is suitable for continous data transfer and also high-speed data transfer.
- In asynchronous transmission, each character or data byte is sent with its own start and stop signals. which help the receiver know when the transmission begins and ends.
- It is suitable for application where small amounts of data is transmitted

- Synchronous Transmission.

- Synchronous Transmission is a method of data communication where the sender and receiver are synchronized by shared clock signals.
- This clock signals ensures both devices to know when to send or receive data
- Synchronous Transmission is more efficient for transferring large block of data.
- In synchronous transmission data are transmitted in blocks consisting of several hundreds and thousands of bits each.
- There are no start and stop bits in synchronous transmission. Data flows continously, synchronized with the clock signals.
- It requires more complex hardware and synchronization mechanisms compared to asynchronous transmission.

- Difference between synchronous & Asynchronous.

Synchronous Transmiss:	Asynchronous Transmission
i) It uses clock signals	i) It don't uses clock signals
ii) Data sent in continous blocks	ii) Data sent one bit at a time
iii) more efficient	iii) Less efficient
iv) Higher data rate	iv) Lower data rate
v) suitable for large amount of data transfer	v) suitable for small amount of data transfer
vi) suitable for irregular data transfer	vi) suitable for continous data transfer
vii) Low-speed data Transfer	viii) High-speed data Transfer
viii) Applications:- keyboards serial ports.	Viii) Applications:- Ethernet, SPI. etc

- sTart bit :- A bit (usually 0) that indicates the beginning of data bytes.

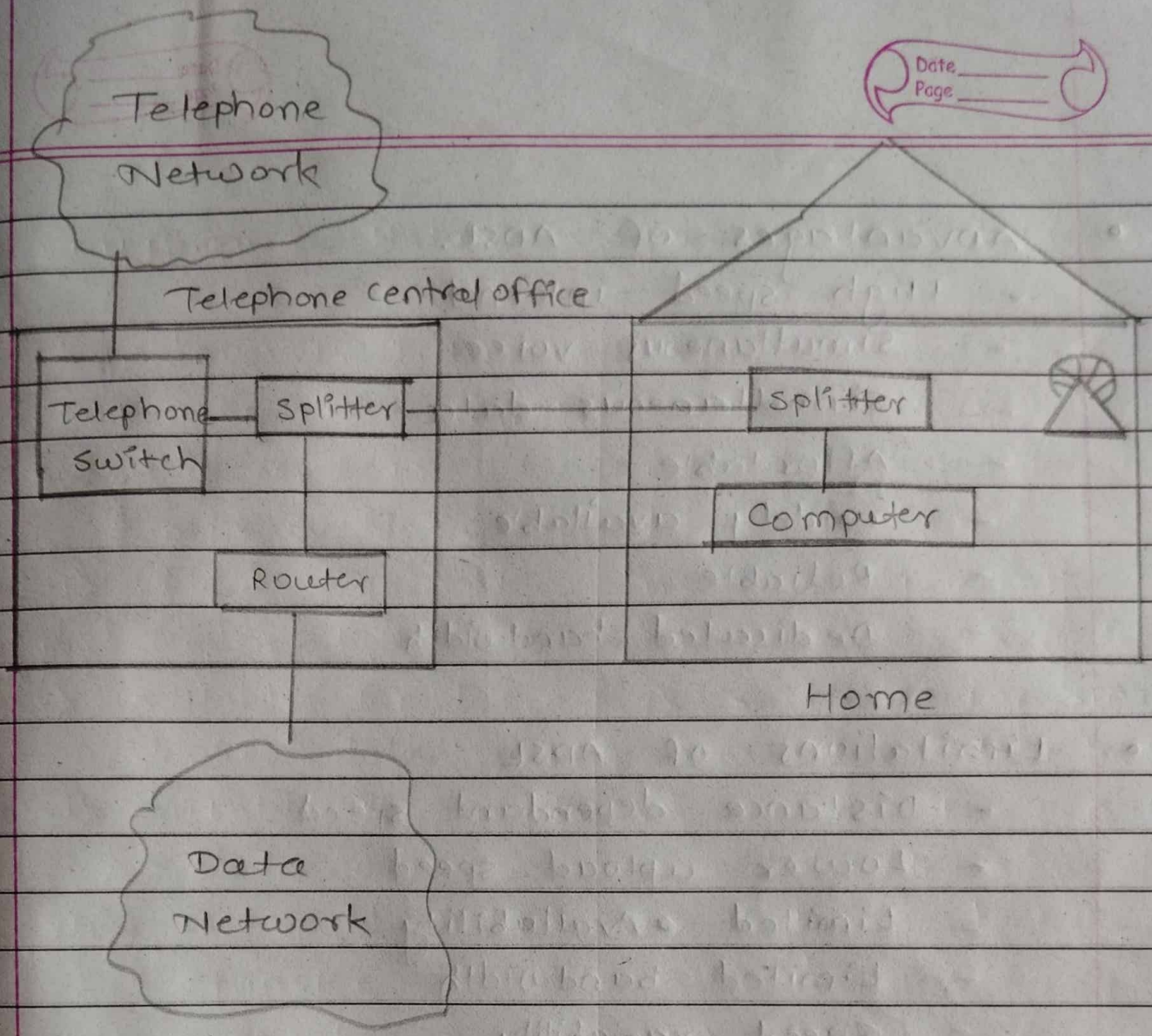
- sTop bit :- A bit (usually 1) that marks the end of data bytes.

- clock signals.

- A clock signals is a timing signal used to synchronize the sender and receiver.
- A clock signal ensure that the both the sender and receiver transmits the data at the same speed or not
- A clock signal ensure when to send or receive data

- ADSL (Asymmetric Digital subscriber line).

- ADSL is a type of high-speed internet connection that uses existing telephones lines to deliver brodband data.
- It offers faster download speed than upload speeds, making it ideal for typical internet usage. like web browsing.
- ADSL allows simultaneous internet and phone use without interference, but its speed is limited by the distance from the telephone exchange
- ADSL is still available in many areas but generally offers lower speeds compared to modern options



* ADSL connection *

- Features of ADSL

→
- Faster download speed
- Distance dependant speed
- Suitable for real time activities like gaming
- Uses existing telephone lines
- Always on connection (no need to dial in)
- widely available in areas in telephone infrastructure
- Affordable
- Dedicated connection

- Advantages of ADSL.
  - High speed internet
  - Simultaneous voice
  - Simultaneous data
  - Affordable
  - widely available
  - Reliable
  - Dedicated bandwidth

- Limitations of ADSL
  - Distance dependant speed
  - Lower upload speed
  - Limited Availability
  - Limited bandwidth
  - Speed variability
  - Not ideal for high volume downloaders
  - Limited quality of services

- Standard communication interfaces.

- Standard communication interfaces define how devices communicate and interact with each other across various enviornments.
- These interfaces ensure that devices from different manufacturers can work together, enabling interoperability across networks and systems.

1) Serial communication interfaces :-
- These interfaces allow data to be transmitted one bit at a time over single channel.

2) parallel communication Interfaces :-
- These interfaces transmit multiple bits simultaneously.