- Sequential processing & parallel processing with benefits:-

- Sequential processing is the type of ~~Computing~~ Processing where one instruction is given at a particular

Sequential processing:-

- It is a straight-forward approach where instructions ~~tasks~~ are executed one after the other.

- It is also known as traditional processing method as all the instructions are executed in a sequence.

- It contains a single processor with low performance and high work-load of processor.

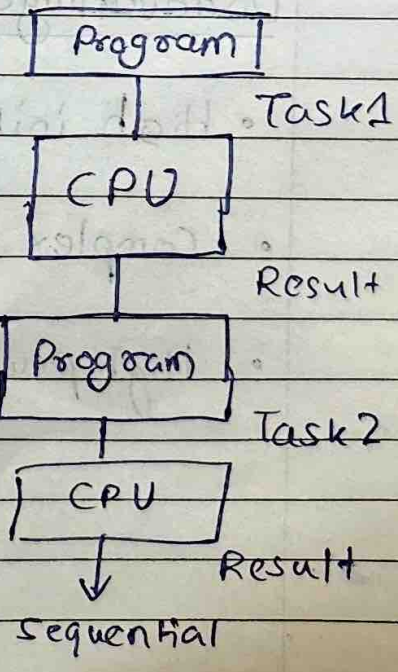
Advantages:-

- Simple
- Easy to ~~use~~ understand
- Easy to maintain

Disadvantages:-

- Time Consuming
- Limited Performance
- Higher cost
- Less efficient

Diagram:-



Parallel Processing :-

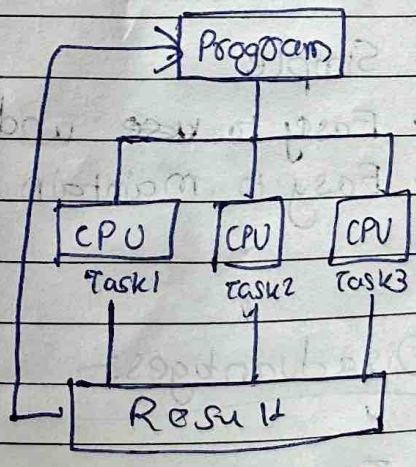
- Parallel Processing involves executing multiple instructions or processes simultaneously.
- The primary propose of parallel processing is to enchance the computer processing capability and increase its throughput
- It can be achieved by multiplicity of functional units that perform identical or different operations simultaneously.

Advantages :-

- Faster Execution
- ~~Solving~~ Suitable for Large problem solving
- Increased computation power.

Disadvantages :-

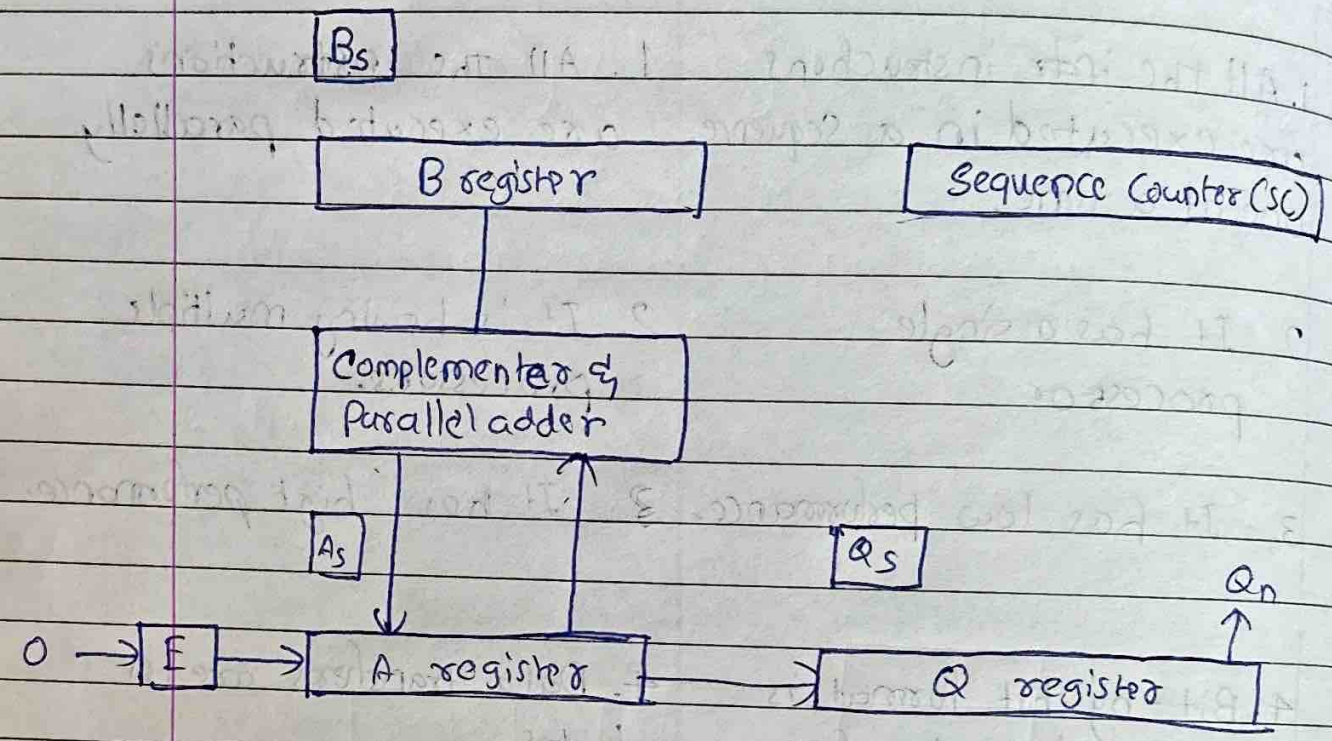
- High initial cost
- Complex
- High power consumption



Sequential Processing	Parallel Processing
1. All the ~~instr~~ instructions are executed in a sequence, one at a time.	1. All the instructions are executed parallelly
2. It has a single processor	2. It is having multiple processors.
3. It has low performance.	3. It has high performance.
4. Bit-by-bit format is used for data transfer.	4. Data transfers are in bytes.
5. Slow execution	5. Faster execution
6. Cost is low	6. Cost is high

2.

- Hardware Implementation for multiplication



- B Register:- Holds the multiplier value (B_s) & interacts with the Sequence Counter (SC)

- A Register:- Contains the multiplicand value (A_s) and is connected to the complementer & parallel adder for processing.

- Sequence Counter (SC): Likely controls the sequence of operations.

- Complementer & parallel adder:- Performs arithmetic operations on the value from the A Register.

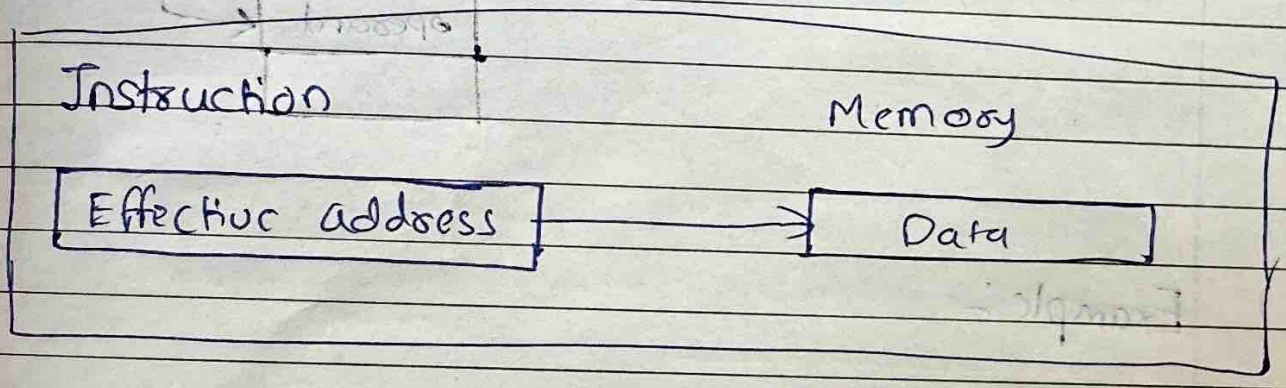
- **Q Register:-**

Stores the result (Q_s) & is linked to the rightmost bit (Q_n) indicating the least significant bit of the result.

3. **Direct Memory addressing:-**

- It is an addressing ~~mode~~ mode where the memory address of an operand is directly specified in the instruction.

- The address field in the instruction contains effective address of the operand and no intermediate memory access is required.



Example:-

LOAD R1, 1000

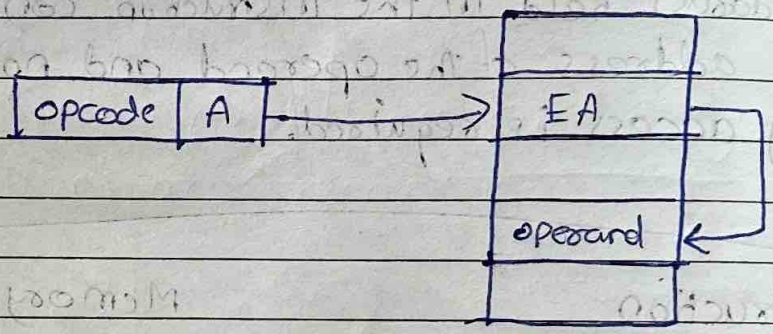
- LOAD is the operation to load data
- R1 is the register where we want to load data
- 1000 is the memory address where the data is located.

CPU directly fetches data from memory address 1000

and loads it into register R1 without any additional steps.

- Indirect Memory Addressing Mode:-

- It is an addressing mode where the memory address of an operand is not directly specified in the instruction.
- The address field contains a reference to the effective address.



Example :-

LOAD R1, @500

It loads the content from the memory location stored at address 500 @ into register R1.

4. Interrupt :-

Interrupt is a signal generated by either hardware devices or software components to interrupt the normal execution of a program and request

Interrupt is a signal ~~general~~ generated by the peripheral when it is ready to communicate with the CPU.

Priority Interrupt :-

It is a system which decides the priority at which various devices, which generates the interrupt signal at the same time, will be serviced by CPU.

The system has authority to decide which conditions are allowed to interrupt the CPU, while some other interrupt is being serviced.

Types :-

1. Hardware Interrupt :-

When the signal for the processor is from an external device ~~then this~~ or hardware then this interrupts is known as hardware interrupt.

Types :-

- Maskable Interrupt :- The hardware interrupts which can be delayed when a much high priority interrupt has occurred at the same time.

- Non- Maskable interrupt :-

The hardware interrupts which cannot be delayed & should be processed by the processor immediately.

2. Software interrupts :-

The interrupt that is caused by any internal system of the computer system is known as a software Interrupt.

Types:-

- Normal Interrupt:-

The interrupts that are caused by software instructions are called normal interrupt.

- Exception:-

Unplanned interrupts which are produced during the execution of some program are called exceptions.

5.

In RISC (Reduced instruction set Computer), the instruction set of the computer is simplified to reduce execution time.

- It has small set of instructions which include register - to - register operations.
- RISC avoids complex & specialized instructions.

Features :-

- Few instructions
- Few addressing modes
- It is used for memory access limited to load & store instructions.
- All operations are done within the registers of CPU.
- It has fixed length
- It is used for single-cycle instruction execution.

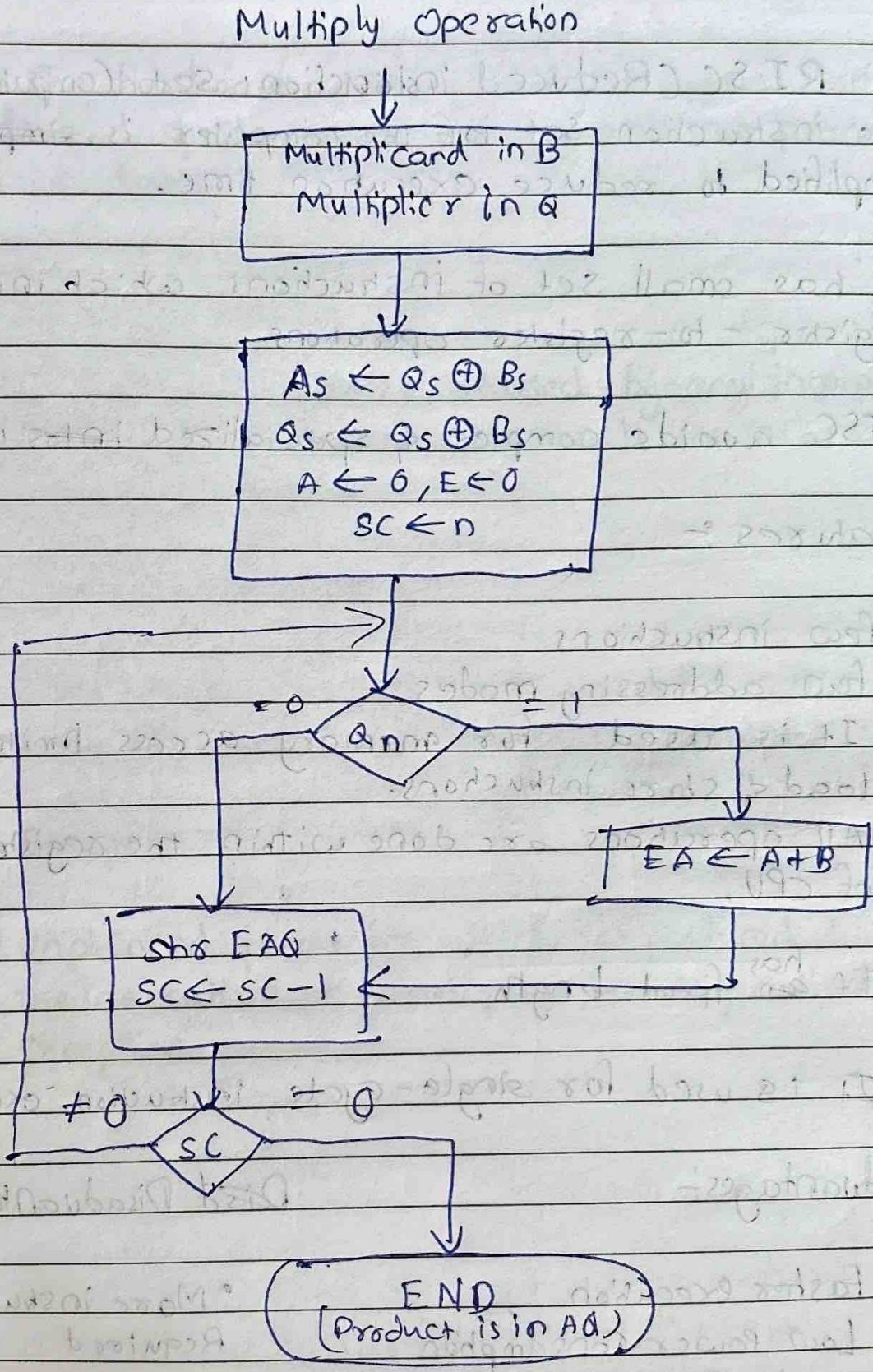
Advantages :-

- Faster execution
- Low Power Consumption
- simple Design.

Disadvantages :-

- More instructions Required
- Increased Memory usage
- High Development cost.

6. Mul


Multiply Operation
Multiplicand in B
Multiplier in Q
As ← Qs ⊕ Bs
Qs ← Qs ⊕ Bs
A ← 0, E ← 0
SC ← n
= 0
Qn
= 1
EA ← A+B
shr EAQ
SC ← SC−1
≠ 0
SC
= 0
END
(Product is in AQ)

7.

Advantages of Assembly language:-

1. Control & efficiency:-

It provides direct control over hardware to optimize code for specific tasks.

2. Low-level Access:-

It allows direct direct access to hardware components.

3. Compact code

Disadvantages:-

1. Complexity:-

Requires deep understanding of hardware.

2. Machine Dependence:-

Tied to a specific CPU architecture.

3. Lack of abstraction

8

- Peripheral devices are external hardware components connected to a computer system.
- They enhance the user's experience by providing input/output functions.
- They extend Capabilities of computer.

Classification of Peripheral :-

a. Input Devices :-

These devices allow users to input data from the outside world into the computer.

E.g :-

Keyboard, Mouse, Scanner, Microphone.

b. Output Devices :-

These devices provide output information from the Computer to outside world.

E.g :-

Printer, Monitor, Speaker.

a.

Main Memory :-

- Main memory also known as primary Memory or RAM (Random access memory) is a crucial component of a computer system.
- It provides temporary storage for data & instructions that the CPU actively uses during program execution.

Characteristics :-

- Volatile :- Data is lost when computer is powered off.
- Fast access :- The CPU can quickly read from writh to main memory.
- Directly Addressable :- Each memory location has a unique address

Virtual Memory :-

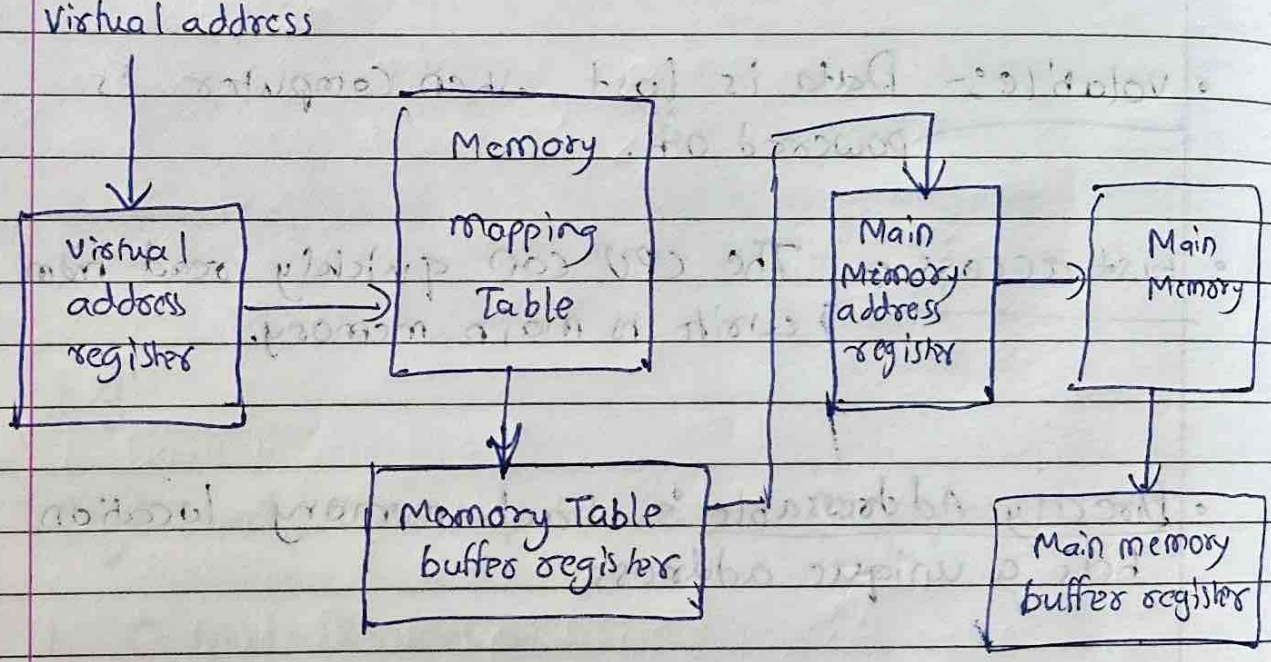
- Virtual Memory is a memory management technique.
- It allows a Computer to temporarily increase the Capacity of its main memory by using Secondary storage.

- **Address space:-**

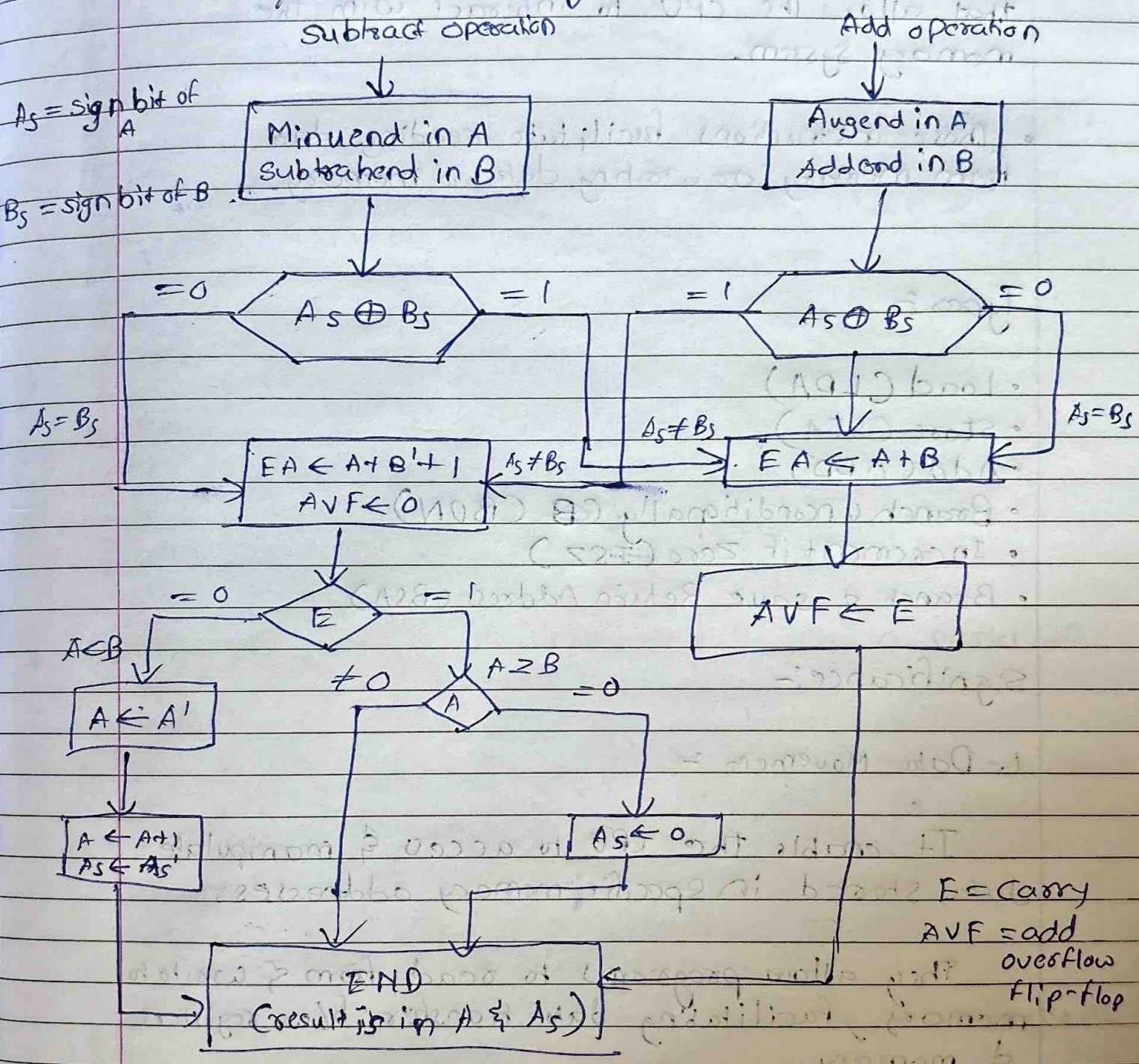
An address used by a programmer is called virtual address and set of such address is known as address space.

- **Memory space:-**

An address in main memory is called a location or physical address. The set of such locations is called memory space.



10. Addition / subtraction Algorithm:



11. • Memory - reference instructions is a command that allow the CPU to interact with the memory system.

• These instructions facilitate reading data from memory or writing data to memory.

Types :-

- Load (LDA)
- Store (STA)
- Add (ADD)
- Branch Unconditionally (BUN)
- Increment if Zero (ISZ)
- Branch & save Return Address (BSA)

Significance :-

1. Data Movement :-

It enable the CPU to access & manipulate data stored in specific memory addresses.

They allow programs to read from & write to memory, facilitating data transfer b/w registers & memory.

2. Resource Management :-

- STA (store) instruction ~~ensures data~~ prevents data loss when the program terminates or switches context.

3. Efficient Execution:-

- It is optimized for direct memory access.
- They minimize overhead, allowing efficient data manipulation & communication with the memory system.

12.

Pipelining:-

- Pipelining is much like an assembly line. Because the processor works on different steps of the instruction at the same time, more instructions can be executed in a shorter period of time.
- It splits processor instructions into a series of small independent stages.
- Each stage performs a specific part of the instruction.

Advantages of pipelining:-

- Increased Throughput:-
  - It allows multiple instructions to be processed simultaneously.
- Reduced Cycle Time
  - Breaking down instruction execution into smaller stages reduces cycle time.

- Faster ALU Design:-
  - It enables design of faster ALUs
- Higher clock frequencies:-
  - Pipelined CPUs can work at higher clock frequencies than RAM.

1:• Assembly Language Program to add two numbers.

```
START
MVI A 15H
MVI B 20H
ADD B
OUT 05H
STA D
HLT
```

2.•

Program control instructions are the machine code that are used ~~by~~ in assembly language by user to command the processor act accordingly.

They determine the sequence in which instructions are executed, enabling a program to make decisions.

Program control instructions controls the flow of execution through these features:-

1. Sequential Execution:-

It ensures that instructions are executed in the order they appear in the program's code.

2. ~~Contro~~ Conditional Branching:-

It enables the program to make decisions based on certain conditions.

For e.g:- An 'if' statement in a high-level language translates to conditional branch instructions in machine code.

3. Unconditional Branching :-

Sometimes, it's necessary to redirect the flow of execution without any condition being met.

Unconditional branch instructions allow the program to jump to a different part of the code unconditionally.

4. Subroutine Calls:-

When a program needs to execute a set of instructions repeatedly or separately, it can call subroutines.

5. Exception Handling:-

When an unexpected event occurs such as an hardware interrupt or software interrupt then program control instructions helps by jumping to specific error-handling routines.

Extra point on Program Control instruction:-

They form the backbone of program control flow enabling the execution of instructions in a logical and structured manner.

3.

- Stack is a data structure that follows the Last In, First Out (LIFO) principle.

- In Computer Organization, Stack is used for CPU organization. It plays a major role in managing program flow and data.

- It's commonly used for function calls & local variables.

~~There are two types of stack based CPU organization.~~

- It has two operations:-

PUSH:- Inserts an operand at the top of the stack & increments the stack pointer

POP:- Removes an operand from top of stack & decrements the stackpointer.

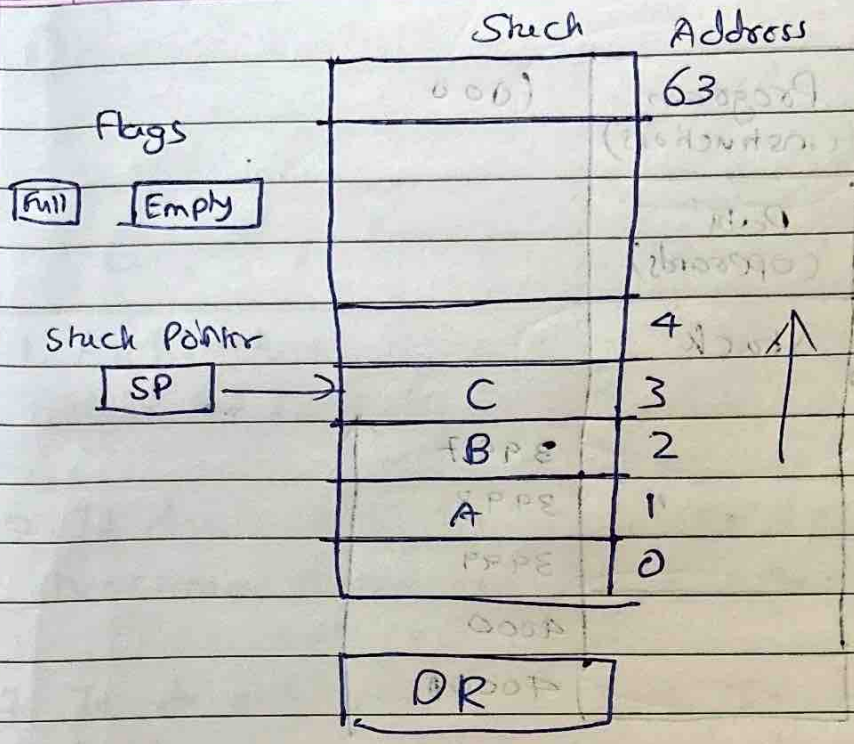
- It has two types of stack based CPU organization:

1. Register stack CPU Organization:-

The stack can be arranged as a set of memory words or register.

Stack point register:-

It includes a binary number, which is the address of the element present at the top of stack.



2. Memory based CPU organization:-

A portion of memory is assigned to a stack operation to implement the stack in CPU.

It consists of:-

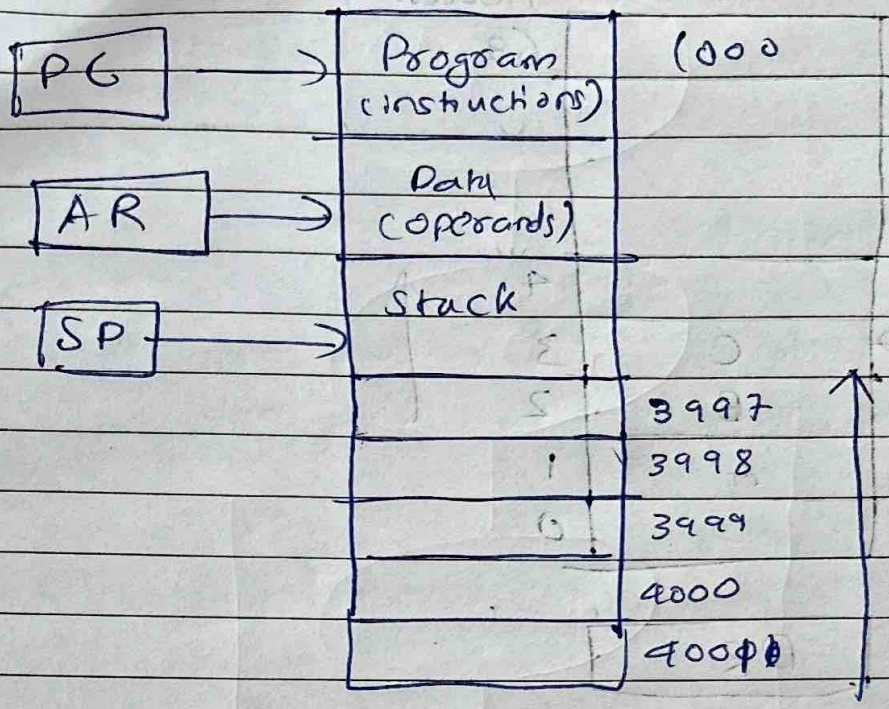
Program Counter(PC):- It is register that points to the address of the next instruction.

Address Register(AR):-

This register points at the collection of data & is used during the execute phase to read an operand.

Stack pointer:-

It points at the top of the stack and is used to perform operation on stack.

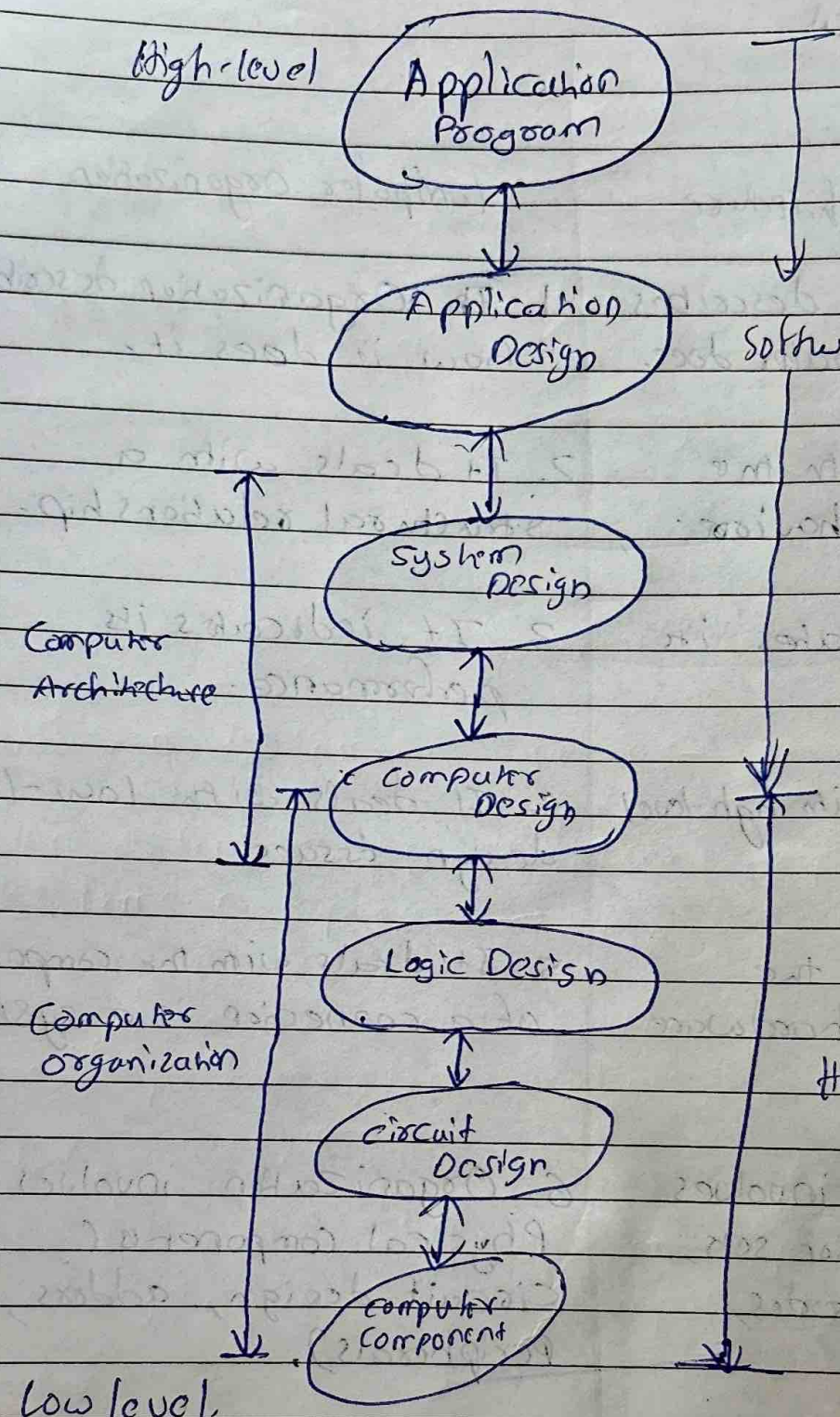


E.g. of stack-based Computers:-

PDP-11, Intel's 8085, HP3000

4. Difference b/w

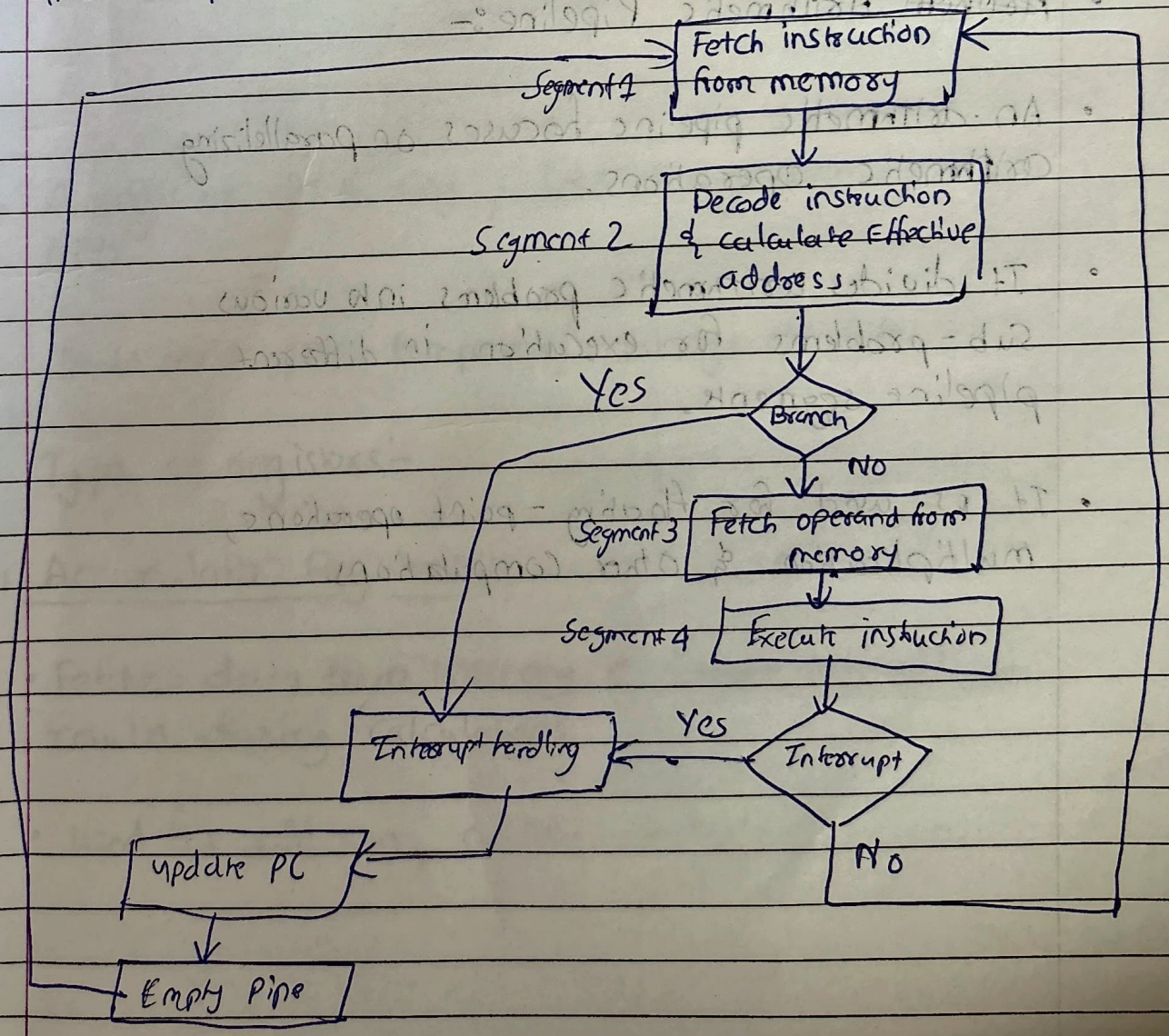
Computer Architecture	Computer Organization
1. Architecture describes what the computer does.	1. The Organization describes how it does it.
2. It deals with the functional behaviour.	2. It deals with a structural relationship.
3. It indicates its hardware.	3. It indicates its performance.
4. It deals with high-level design issues.	4. It deals with low-level design issues.
5. It acts as the interface b/w hardware & software.	5. It deals with the components of a connection in a system.
6. Architecture involves Logic (instruction sets, addressing modes, Data types)	6. Organization involves Physical components (Circuit design, adders, Peripherals).



Computer System.

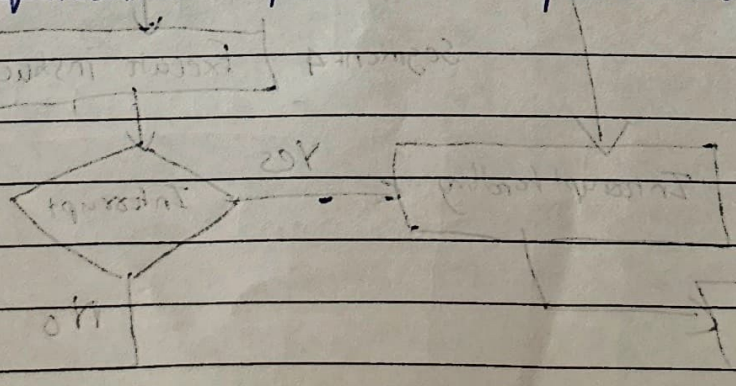
5.

- An instruction pipeline is a technique used to enchance the performance and efficiency of processors.
- The primary goal of an instruction pipeline is to maximize the utilization of hardware resources and improve the throughput of instructions.
- It achieves this by breaking down the instruction execution process into smaller stages and processing them in parallel.



1. Fetch the instruction from memory
2. Decode the instruction
3. Calculate the effective address
4. Fetch the operands from memory
5. Execute the instruction
6. Store the result in the proper place

- ~~Addition~~ Arithmetic Pipeline :-
- An arithmetic pipeline focuses on parallelizing arithmetic operations.
- It divides arithmetic problems into various sub-problems for execution in different pipeline segments.
- It is used for floating-point operations, multiplication & other computations.



Instruction pipeline	Arithmetic Pipeline
1. Optimizes Instruction execution.	1. Enchances arithmetic operation execution.
2. Deals with instruction fetch, decode and execution.	2. Focuses on arithmetic calculations.
3. Overlaps instruction phases	3. Overlaps ~~arithm~~ arithmetic sub-operations.

6. A register is a tiny, fast storage memory within CPU.

It is used for temporary storage of data.

Types of registers:-

1. Accumulator Register:-

- Fetches data from memory & stores intermediate results during calculations.
- Used for addition, subtract, multiply & division.

2. Program Counter Register:-

- Keeps track of the memory address of the next instruction.

3. Address Register:-

Holds address for memory.

4. Instruction Register:-

Holds instruction code.

5. Temporary Register:-

Holds temporary data

6. Data Register:-

Hold memory operand.

7. Output Register:-

Holds Output character

8. Input Register:-

Hold input Character.

6. Machine language:

7. • Machine language is a low-level language.

• The instructions are given in binary format, in the form of 0s & 1s.

Assembly language:-

• It is also known as an intermediate language.

• It has mnemonics & is understandable by humans only.

High-level language:-

• It is a language that is designed to make it easier for humans to understand & write.

• It is a language which is used for writing programs which could be understood by the humans & computer.

Machine Language	Assembly language	High-level language
1. Only understood by computers	1. Only understand by human.	1. Understood by both machine & humans
2. Data represent in binary format (0 & 1)	2. Data represented with the help of mnemonics.	2. ~~Data~~ It is almost similar to English language.
3. Difficult to understand by humans	3. Easy to understand by humans.	3. Easy to understand by ~~both~~ humans & ~~machine~~
4. Debugging cannot be done.	4. Debugging can be done.	4. Debugging can be done.
5. Execution is fast	5. Execution is slow compared to machine language	5. Execution is slow compared to assembly & machine language
6. No need of translator.	6. Assembler is used as translator.	6. Compiler is used as translator.

8.

Memory-Mapped I/O :-

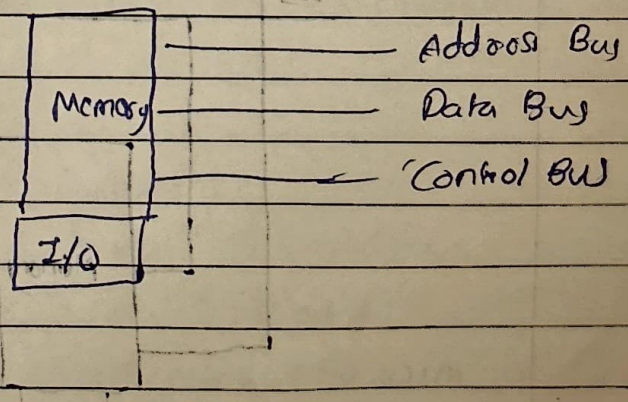
- In this, both memory and I/O devices share the same address space.
- The CPU communicates with both memory & I/O devices using the same bus.
- I/O devices are assigned specific memory addresses.

How it works :-

- CPU treats I/O devices like regular memory locations.
- Same set of instructions can be used to access both memory & I/O devices
- Data transfer occurs seamlessly b/w CPU & I/O devices.

Advantages :-

- Faster I/O operations.
- simple
- Efficient.

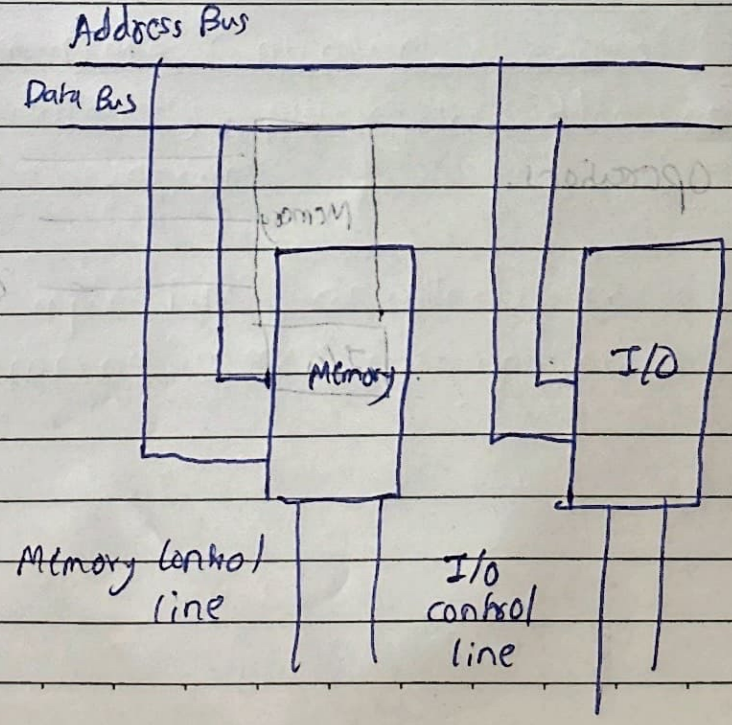


Isolated I/O:-

- In isolated I/O, memory & I/O devices also have separate address space, but they have separate read & write control lines for I/O.
- Separate memory.
  So when the CPU decodes an instruction, it places the address on the address line.
- The appropriate I/O read or write control line is activated.
- Data transfer occurs b/w the CPU & I/O devices.

Advantages:-

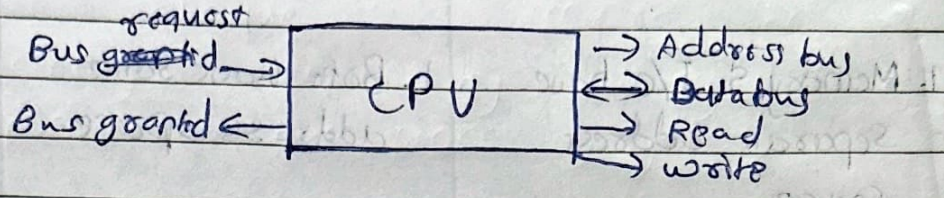
- Large I/O address space.
- Improved Reliability.
- Greater Flexibility.



Isolated I/O	Memory Mapped I/O
1. Memory & I/O have separate address space	1. Both have same address space.
2. Separate instructions for read & write operation in I/O & Memory.	2. Same instructions can control both I/O & memory.
3. More efficient.	3. less efficient
4. Larger in size	4. Smaller in size.
5. It is comptex. logic.	5. It has simpler logic

- DMA :-
  - Direct memory Access is a special feature within a Computer system that enables data transfer b/w memory & peripheral devices without intervention of CPU.
  - It used for Large data transfers.
  - DMA shifts the work to a dedicated DMA controller instead of burdening CPU with these.

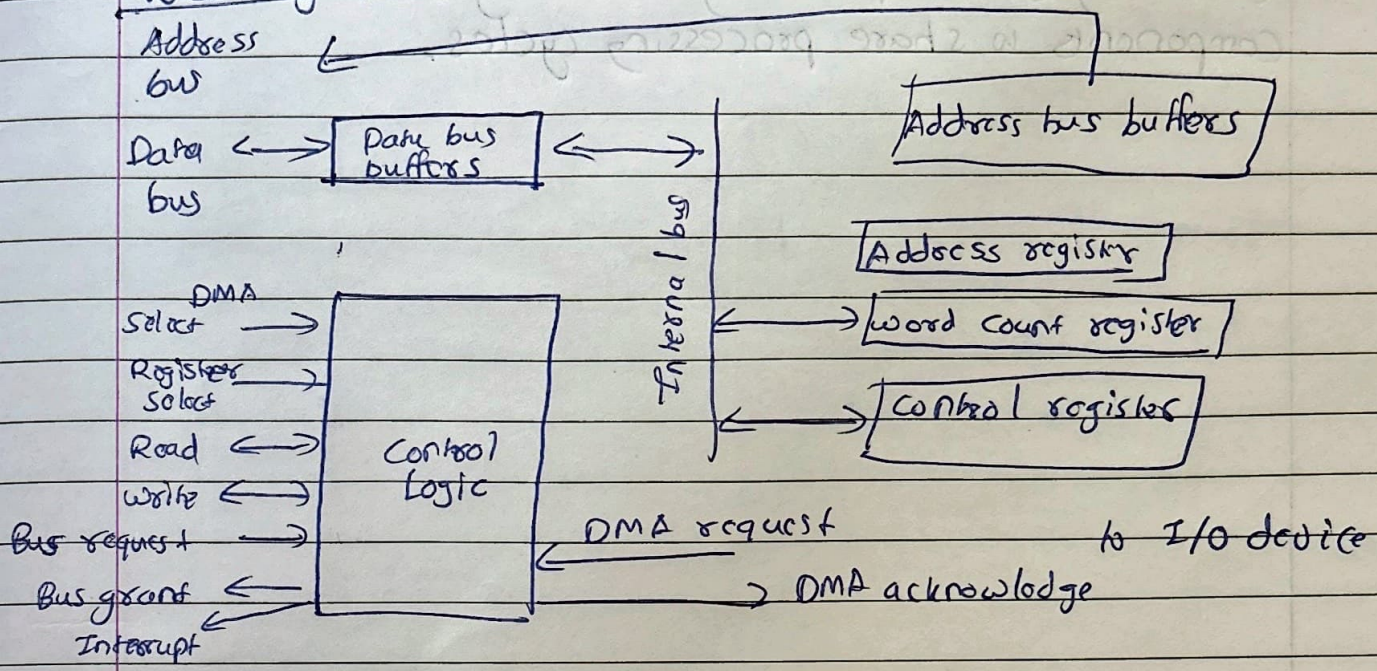
- **DMA Transfer:-**



- The host writes a DMA command block into memory.
- This block contains:
  - Pointer to the source of the transfer
  - Pointer to the destination of the transfer.
  - Count of the no. of bytes to be transferred.
- CPU writes the address of this command block & continues with other tasks.
- The DMA Controller operates the memory bus directly, placing the address on it without CPU intervention.
- A simple DMA controller is now a standard component in modern computers.

Working:

- Working:-



- DMA request :- Device controller signals when data is available for transfer.

- DMA acknowledge :- DMA controller seizes the memory bus & places the desired address on this wire.

- Upon Successful data transfer, the device controller receives the DMA acknowledge and removes DMA request.

- When entire transfer is finished, DMA controller interrupts the CPU.

- This process improves system performance.

Cycle Stealing:-

It is a technique used to improve the efficiency of computer by allowing different components to share processing cycles.