Unit-1

• Digital Computer

→ digital system that performs various computational tasks.

→ info in comp. is represented by variables.

→ use binary no. system (0 or 1)

→ BIT: binary digit.

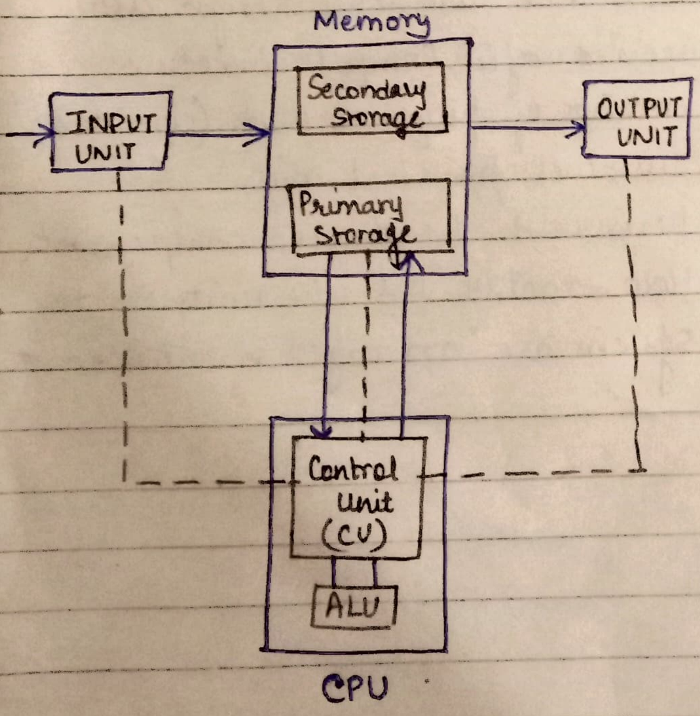
→ INFO: grp of bits.

→ Two functional entities

(1) Hardware → electronic components and electromechanical device. Eg. Keyboard, mouse, monitor etc

(2) Software → instruct^{ns} & data that a computer manipulates to perform various data processing tasks.

• Functional component of Digital Computer



1) Input Unit: External device connected to CPU.

→ used to feed data & instructions for solving problem at hand.

→ Control unit sends signal to it, to receive data & instruct^{ns} from user.

2) Output Unit: Used to display processed data after execution of a program. Eg. Headphns, Speakers, Printers etc

3) CPU: Central Processing Unit.

→ It processes the data and instruct^{n} received from input unit.

ⓐ Control Unit

→ fetch data & instructions from main memory

→ Interprets instruct^{ns}

→ Controls I/O devices

ⓑ ALU

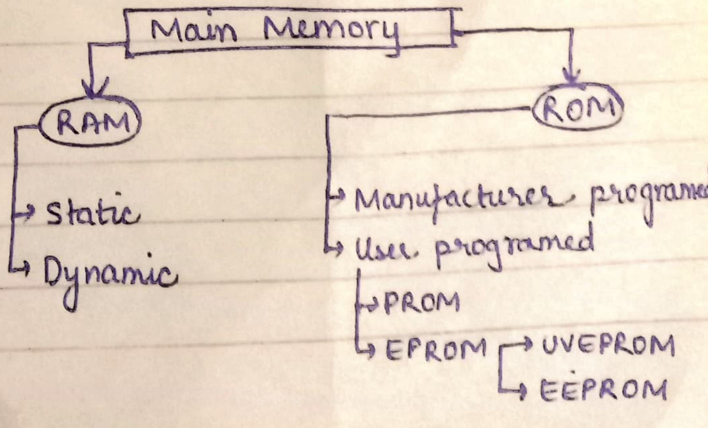
→ All math. calculat^{n} & logical calculat^{n}

→ consists of electronic circuitary.

4) Memory → storage device.

→ Instruct^{n}/data are stored here in the form of 0s & 1s.

→ Types
- Main Memory
- Secondary Memory
- Cache Memory



Parametess	RAM	ROM
	Random Acess Memory	Read Only Memory
1) Data & Program	stores during & after Processing	Stored by Manufacturer
2) Content	Temporary stored	Permanently Stored
3) Processing Time	very fast, uses lot of power	fast, less power is used
4) Volatility	Volatile	Non-Volatile
5) Data Writing	faster	slower
6) Costing	Expensive	Cheap.
7) Hardware Structure	In form of Chips	In form of drives (mag. tapes)
8) Data Modification	Can be modified	Can't be modified
9) User Control	Managed by OS.	Preloaded with Software
10) Read / Write Operat^n	Both can be done	Only reading.
11) Eg.	Static & Dynamic RAM	PROM, EPROM etc.

Computer Architecture	Computer Organisation
→ deals with functional behavior	→ deals with structural behavior (How h/w is connected)
→ Tells What to Do?	→ Tells how to do?
→ Deals with high level design issue	→ Deals with low level design issue
→ Architecture is fixed first	→ Decided after Comp Architecture
→ Comprises logical functions	→ consists of physical units (circuits etc)
→ Involves logic	→ Involves physical units
→ Acts as interface b/w H/w & S/w	→ Deals with hardware components
→ Helps us to understand the functionality of a system	→ How exactly are the units in the system are arranged & inter connected

⊛ BUS

→ communicat^n system that helps data transfer b/w different modules of comp.

→ communicat^n pathway connecting devices

→ grp of wires/electric lines that carry computer signals.

→ consist of multiple lines. Each line is capa~~ba~~ble of transmitting signals in 0s or 1s.

- System bus: bus that connects major computer components (processor, memory, I/O).

- Bus lines (Types)

1) Data Bus → moves data b/w components

→ Bidirectional

→ width : no of lines in databus

→ Each line can carry 1 bit at a time

2) Address Bus → Desiginates the source or destination of the data on data bus

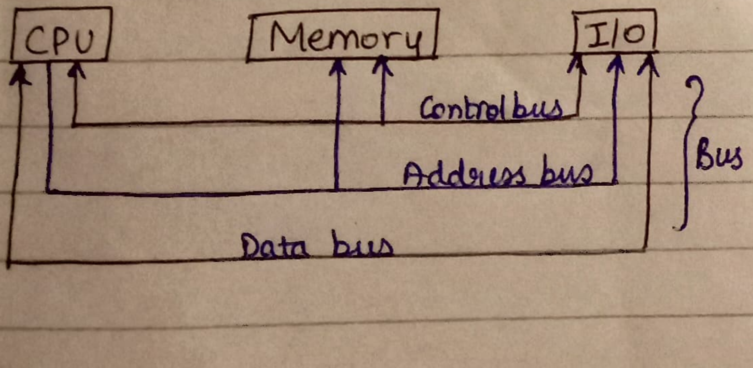
→ unidirectional

→ width : max possible memory capacity of system

3) Control Bus → controls the access to and the use of the data & address lines

→ transmits both command & timing info.

→ Bidirectional



- Types of Bus

1) Dedicated :

⊛ Bus Arbitration

- Since, more that one module may need control of bus (eg {CPU & DMA controller) Thus, the process by which multiple requests are recognised & priority given to one of them is called Bus Arbitration

- Approaches

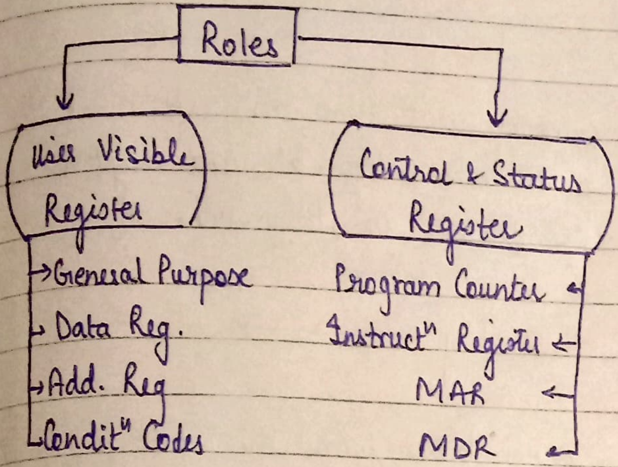
↳ Centralised single bus arbiter performs required arbitration

↳ Distributed : there is no central controller thus device participate in the selection of the next master.

⊛ REGISTERS

→ stores data temporarily

→ grp of flip flops with each flip flop capable of storing 1-bit info.



Users Visible Register

1) General Purpose Register

→ can be used by programmers

→ contains operands for any opcode

→ can be used for addressing function (eg. Register indirect, displacement)

2) Data Register

→ may be only used to hold data

3) Address Register

→ devoted to a particular addressing modes

→ Eg → ① Segment Pointer

↳ holds add. of base of segment

② Index Registers

↳ used for indexed address & may be autoindexed

③ Stack Pointers

↳ pts the top of the stack

↳ allows implicit addressing i.e push, pop & other stack operat^{n}s.

4) Condition Codes (Flags)

→ bits set by the processor hardware as a result of operat^n

Eg → +ve or -ve result

→ 0 result

→ Overflow

→ Underflow etc.

Control & Status Register

→ controls the operat^n of processor

→ not visible to the users.

1) Program Counter

↳ address of instruction to be fetched.

2) Instruction Register

↳ contains instruct^n most recently fetched

3) MAR (Memory Address Register)

↳ address of locat^n of data in memory

4) MDR/MBR (Memory Buffer Register)

↳ contains word of data to be written in memory or the word most recently read.

Processor Organization

→ Processor contains

1) ALU (Arithmetic and logical Unit)
↳ does actual computation or processing of the data.

2) CU (Control Unit) Supervisor
↳ controls movement of data & instruct^n's into & out of the processor & controls the operation of ALU

3) Registers Minimal Internal Memory.

→ Processor Organization means how the components of processor are connected together and how they accomplish their tasks.

→ A processor does the following things

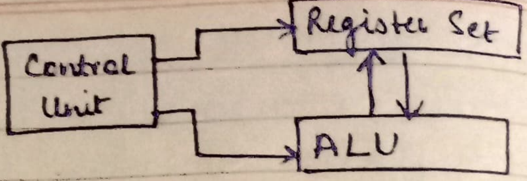
1) Fetches Instructions Reads instruct^n from memory
2) Interprets Instruction Decodes Instruct^n's
3) Fetches Data Data from memory is fetched for execution
4) Process Data Data is processed
5) Write Data. Data is stored in memory

① General Register Organization

→ When large no. of registers is included in the CPU, the most efficient way to connect them is through a common bus system.

→ The registers communicate with each other not only for direct data transfer, but also while performing various microoperations

→ Hence, it is necessary to provide a common unit that can perform all the arithmetic, logical and shift micro-operations in the processor.



Major Components of CPU

Fig ⟶

→ A & B buses forms the input for ALU

→ The OPR (operat^n selection) in ALU determines the Arithmetic & logical microoperations that is to be performed.

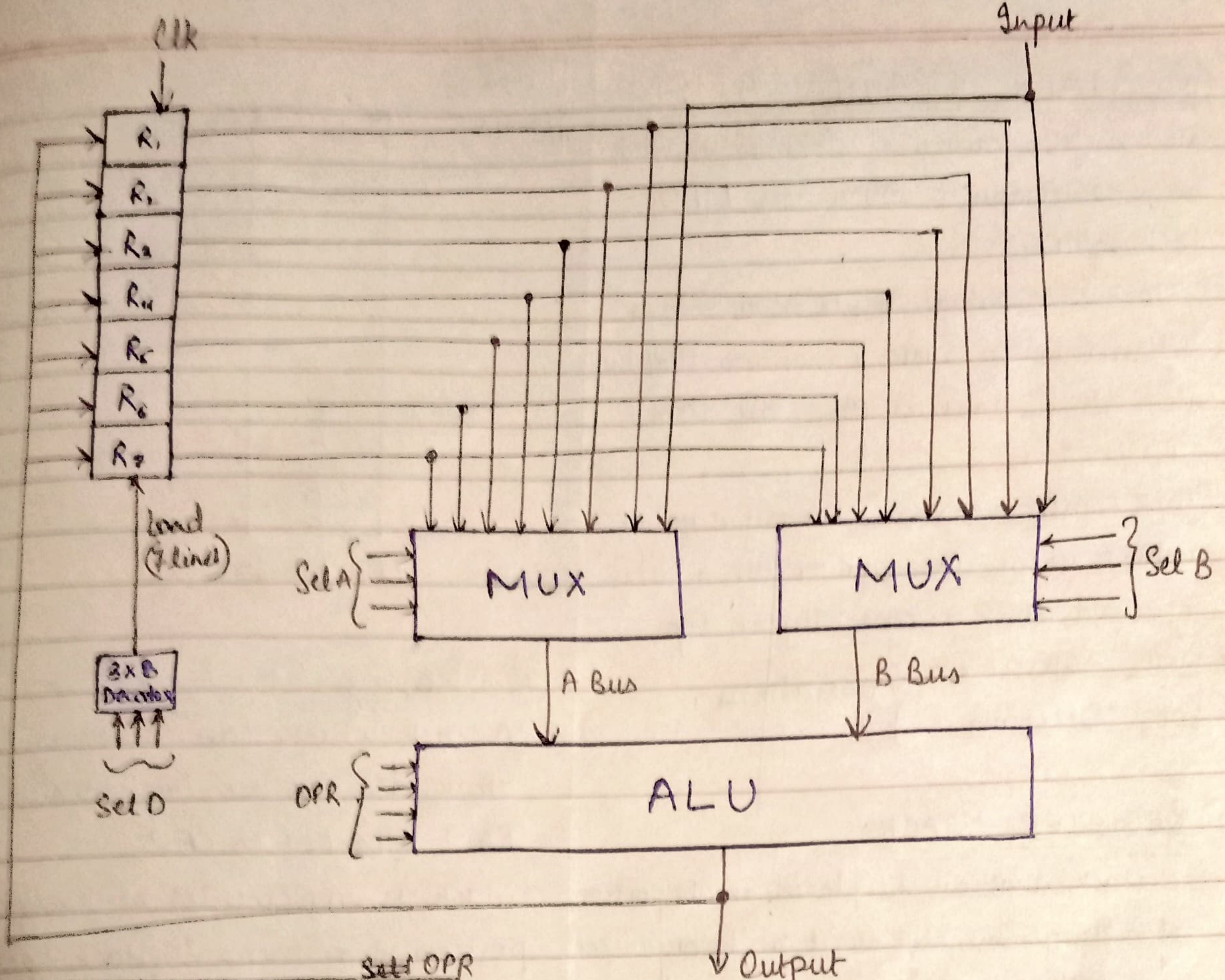
→ The result of microoperation is available for output data and it is also ~~availab~~ ~~f~~ goes into the input of all the registers

→ The register that receives the info from the output bus is selected by the decoder (SELD)

→ The decoder activates one of the 7 load lines, thus providing a transfer path b/w data in the output bus and the input of selected destinat^n register

→ The Control Unit (CU) directs the info flow through registers & ALU by selecting the various components in the system.

→ Give an Eg. $R_1 \leftarrow (R_2 + R_3)$



~~Sel~~ OPR			
SelA	SelB	Sel D	Operation

↖ control word

General Register Organization

- Control word

→ There are 14 binary selection inputs in the unit & their combined value specifed by CONTROL WORD.

- 3 fields contains 3 bit each and 1 field has 5 bits.

1) SELA (3 bit) → select source register for input A

2) SEL B (3 bit) → select source register for input B.

3) SELD (3 bit) → select destination register using decoder.

4) OPR (5 bit) → Select one of the operations in ALU

□□□	□□□	□□□	□□□□□
SELA	SELB	SELD	OPR

← 14 bit →

⊛ STACK ORGANISATION

→ most useful feature of stack is included in most computer which uses LIFO techniques.

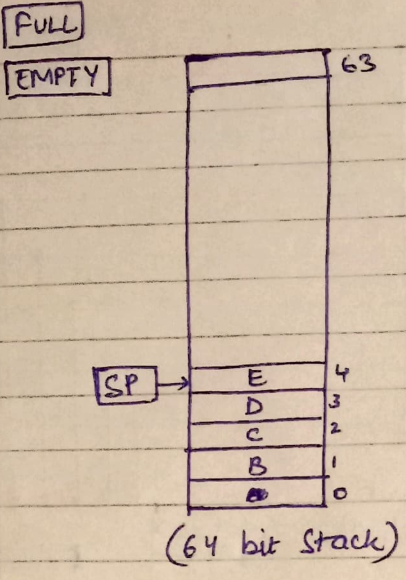
→ A Stack is a storage device that stores information in such a manner that the item stored last is the first item retrieved.

- Stack Pointer → contains binary number whose value is equal to the address of word that is currently on top.
- Push : Insertion } operations.
- Pop : Deletion

① REGISTER STACK

→ A stack that can be placed in the portion of a large memory or it can be organized as a collection of finite number of words (memory wordes) or registers.

- SP (stack Pointer) → - - - - - -
- Push : Insertion of new element (top)
  ↳ SP will get incremented & will point to the next address which is the new pushed element
- Pop : Deletion of new element (from top)
  ↳ SP will get decremented & will pt to the previous address having the second last element before delet^n.
- Empty : Denotes that stack is ~~full~~ empty
- Full : Denotes that stack is full.
  ↳ One bit reg. is set to 1.



(64 bit Stack)

② Memory Stack

→ A stack which can exist as register stack or can be implemented in RAM Attached to CPU

→ Stack is implemented by assigning a portion of memory to stack operat^n. & using a processor ~~&~~ register as SP.

→ Memory is partioned in 3 segment

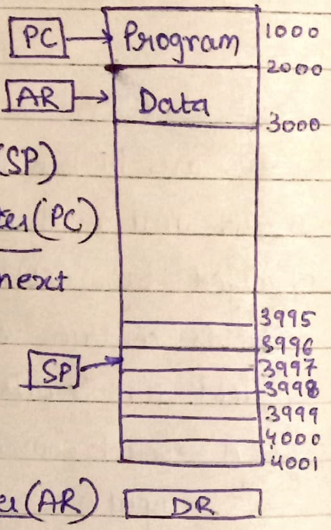
1) Program
2) Data
3) Stack

- Stack Pointer (SP)
- Program Counter (PC)

→ points address of next instructions

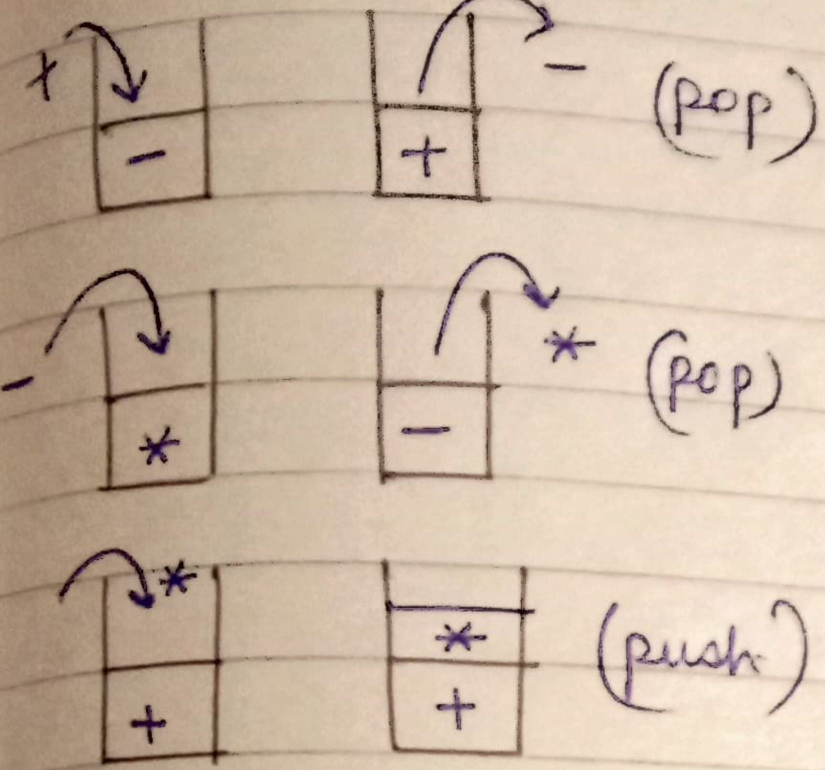
→ Used during fetch phase to read instructions

- Address Register (AR)
  ↳ pts at the array of data
  ↳ used during Execut^n phase.



- Advantage : CPU can refer to memory stack without having to specify an address, ∵ the address is always available & is automat updated by SP.

• INFIX TO POSTFIX



- ( ) (L to R)
- ↑ (R to L)
- / * % (L to R)
- + - (L to R)

- Same priority → Pop
- Low on high → ~~pop~~ Push
- High on Low → Pop