Monika Shaorma

UNIT - 1

COMPUTER ARCHITECTURE AND ORGANIZATION

COMPUTER ARCHITECTURE:- In general terms the architecture of a computer system can be considered as a catalogue system can be of tools or attributes that are visible to the user such as instruction sets. number of Bits used for data, addressing techniques etc.

COMPUTER ORGANIZATION:- Computer organization of a computer system defines the way system is structured. So that all these catalogued tools can be used. The significant components of computer organization are ALU, CPU memory and memory organization.

COMPUTER ARCHITECTURE VS COMPUTER ORGANIZATION:-

COMPUTER ARCHITECTURE	COMPUTER ORGANIZATION
① Computer Architecture is concerned with the way hardware components are connected together to form a computer system.	① computer organization is concerned with the structure and behaviour of a computer system as seen by the user.
② It acts as the interface between hardware and software.	② It deals with the components of a computer connection in a system.

3 computer architecture helps us to understand the functionalities of a system.	3 computer organization tells us How exactly all the units in the system are arranged and inter-connected.
4 A Programmer can view architecture in terms of instructions, addressing modes and ~~risk~~ registers.	4 whereas organization expresses the realization of architecture.
5 while designing a computer system architecture is considered first.	5 computer organization deals with Low - level design issues
6 Computer Architecture involves Logic (Instruction sets, Addressing modes, Data types, cache optimization).	6 organization involves Physical components (circuit, design, Address signals peripherals).

EVOLUTION OF COMPUTER (COMPUTING) Devices:-

ENIAC (Electronic Numerical integrator and computer) was the first computing system designed in the early 1940's. It consisted of 18000 buzzing electronic switches called vacum tubes. 42 panels each 9' x 2' x 1'. It was organized in U-Shaped around the Perimeter of a room with forced air-calling.

⇒ Atanasoff - Berry computer (ABC) design was known first digital electornic computer (though not Programmmable). It was designed and Built by John vincent Atanasoff and His assistand clifford E. Berry in 1937.

⇒ In 1941, Z3 was invented by german inventor Konrad zuse. It was the first working Programmable, fully automatic computing machine.

⇒ Transistors were invented in 1947 at Bell Laboratories which were a fraction the size of vacuum tubes and consumed less Power but Still the complex circuits were not easy to Handle.

⇒ Jack Kilby and Robert Noyce invented the integrated circuit at the same time. In July 1959 Noyce field a patent for this.

⇒ In 1983, Lisa was Launched as the first Personal computer with a graphical user interface (GUI) that was sold commercially. It ran on the motorola 68000, dual floppy disk drives, a 5MB hard drive and Had 1MB of RAM.

⇒ In 1990, Apple released the macintosh Portable. It was heavy weigning 7.3 kg (16Lb) and extremely. expensive. It was not met with great success and was discontinued only two years later.

⇒ In 1990 Inteal introduced the touch stone Delta super computer, which had 512 micro-processors. this technological advancement was very significant as it was used as a MODEL for SOME OF THE FASTEST multi-Processors System in the world.

FUNCTIONAL UNITS OF DIGITAL SYSTEM:-

A computer organization describes the functions and design of the various units of a digital system.

A general - purpose computer system is the best -known example of a digital system. other example include telephone switching exchanges, digital voldmeters, digital counters. electronic calculators and digital displays.

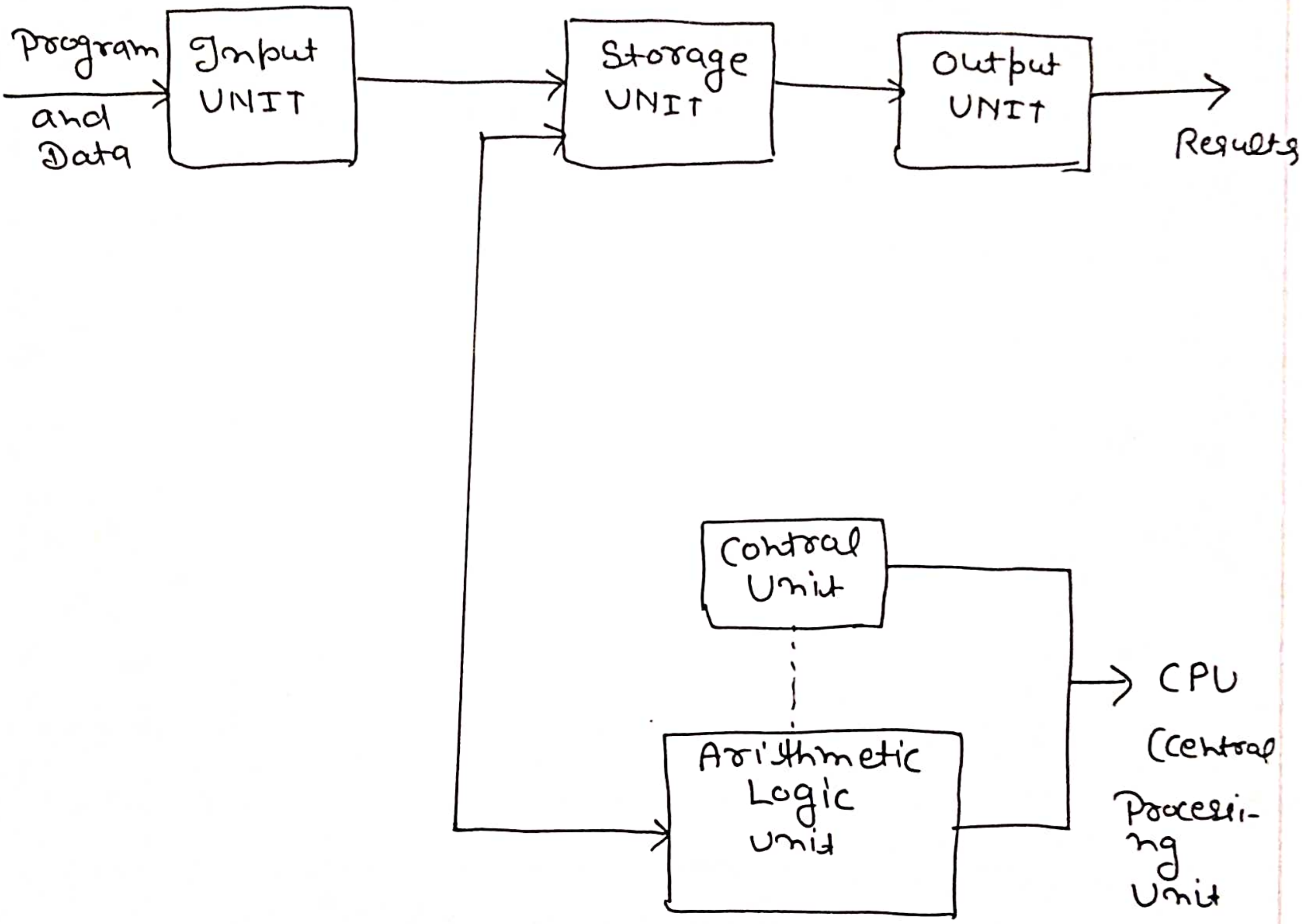
computer architecture deals with the specification of the instruction set and the hardware units that implement the instructions. Set. a

Computer Hardware consists of electronic circuits, displays, magnetic and optic storage media and also the communication facilities.

functional units are a Part of a CPU that Performs the operations and calculations called for by the computer Program.

Functional units of a computer system are parts of the CPU (central Processing unit) that Performs the operations and called for by the computer Program.

A computer consists of five main components namely, Input unit, central Processing unit, memory unit Arithmetic & logical Unit, Control unit and an output unit.



Input unit:- Input units are used by the computer to read the Data. The most commonly used input devices are keyboards, mouse, Joysticks trackballs, microphones etc.

However, the most well-known input device is a keyboard. whenever a key is Pressed, the corresponding letter or digit is automatically translated into its corresponding binary code and transmitted over a cable to either the memory or the Processor.

Central Processing Unit :- Central Processing Unit commonly known as CPU can be referred as an electronic circuitry within a computer that carries out the instructions given by a computer program by performing the basic arithmetic, logical, control and input/output (I/O) operations specified by the instructions.

Memory Unit :-

⇒ The memory unit can be referred to as the storage area in which programs are kept which are running and that contains data needed by the running programs.

⇒ The memory unit can be categorized in two ways namely. Primary memory and secondary memory.

=> It enables a processors to access running execution applications and services that are temporarily stored in a specific memory location.

⇒ Primary storage is the fastest memory that operates at electronic speeds. Primary memory contains a large number of semi-conductor storage cells, capable of storing a bit of information. The word length of a computer is between 16 - 64 bits.

⇒ It is also known as the volatile form of memory, means when the computer is shut down, anything contained in RAM is lost.

⇒ Cache memory is also a kind of memory which is used to fetch the data very soon. They are highly coupled with the Processor.

⇒ The most common examples of Primary memory are RAM and ROM.

⇒ Secondary memory is used when a large amount of data and Programs have to be stored for a long-term basis.

⇒ It is also known as the Non-volatile memory form of memory. means the data is stored Permanently irrespective of Shut-down.

⇒ The most common examples of secondary memory are magnetic disks, magnetic tapes and optical disks.

Arithmetic & Logical Unit :- most of all the arithmetic and Logical operations of a computer are executed in the ALU (Arithmetic and Logical Unit) of the Processor. It Performs arithmetic operations like addition, Subtraction, multiplication, division and also the logical operations like AND, OR, NOT operations.

Control Unit :- The control unit is a component of a computers central Processing unit that coordinates the operation of the Processor. It tells the computer's memory, arithmetic /Logic unit and Input and output devices How to respond to a Program's instructions.

The control unit is also known as the never center of a computer System.

Output unit!-

⇒ The Primary function of the output unit is to send the Processed results to the user. output devices display information in a way that the user can understand.

⇒ output devices are pieces of equipment that are used to generate information or any other responce Processed by the computer. These devices display information that has been held or generated within a computer.

⇒ The most common example of an output device is a monitor.

COMPUTER BUS!- The computer Bus is a communication link used in a computer system to send the data, addresses, control signals and the Power to various components in a computer system.

⇒ A computer system is a digital electronic device that can be Programmed to Perform various operations as per the user requirements.

⇒ The computer system consist of Both h/w and software components.

⇒ A computer system consist of number of components. Some components are internal to the system which are mounted inside the computer casing.

⇒ The system internal components include CPU, RAM, Harddisk, graphic card and motherboard.

⇒ Some of these hardware components are connected to the system externally such as peripheral devices.

⇒ The input peripheral devices typically include input devices keyboard, mouse and scanner.

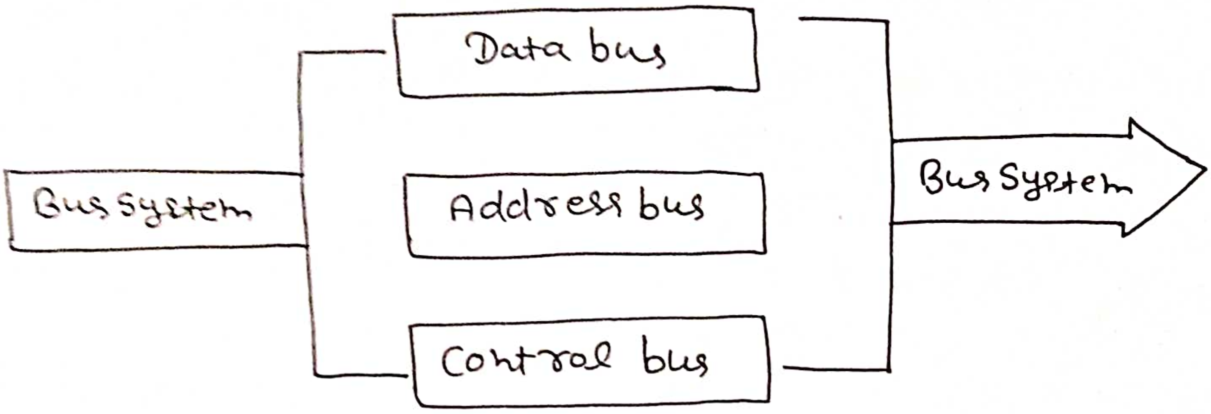
⇒ The output peripheral devices include display monitor, Printer or other external storage devices.

⇒ The computer system hardware components consist of Both electronic components and as well as some mechanical components.

⇒ All these hardware components are connected to the system with a network of wires that runs across the computer system.

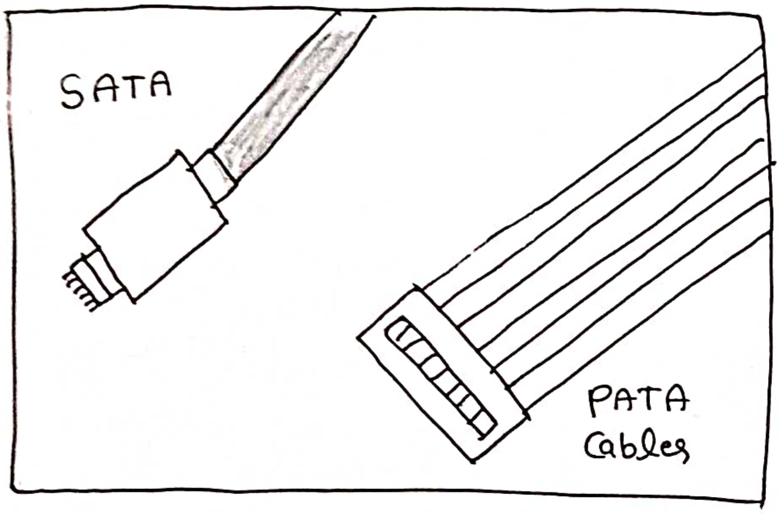
⇒ In computer architecture, the network of electrical wires inside the computer system which interconnects various Hardware components are known as computer buses.

⇒ In other words, the computer buses are wires in various forms that connects various forms that connects various electronic components.



* Each computer Bus, internally consist of group of wires ~~Present in a Bus case~~ tied together. The number of wires Present in a Bus could be 8-bit, 16-bit 32-bit or 64-bit.

* The computer system makes use of different sizes of Buses. This network of different types buses used in the computer is called a Bus System.

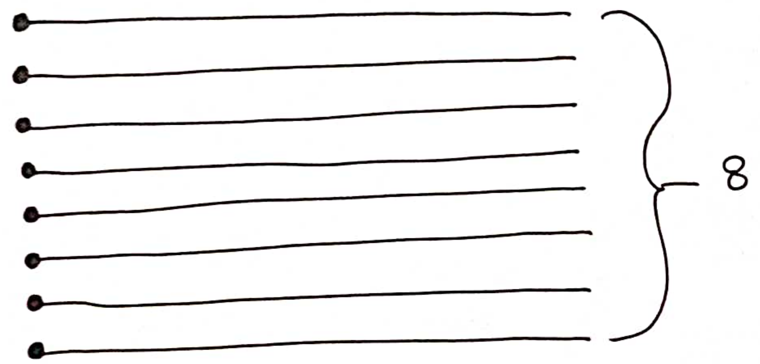


SATA – Serial Advanced technology Attachment

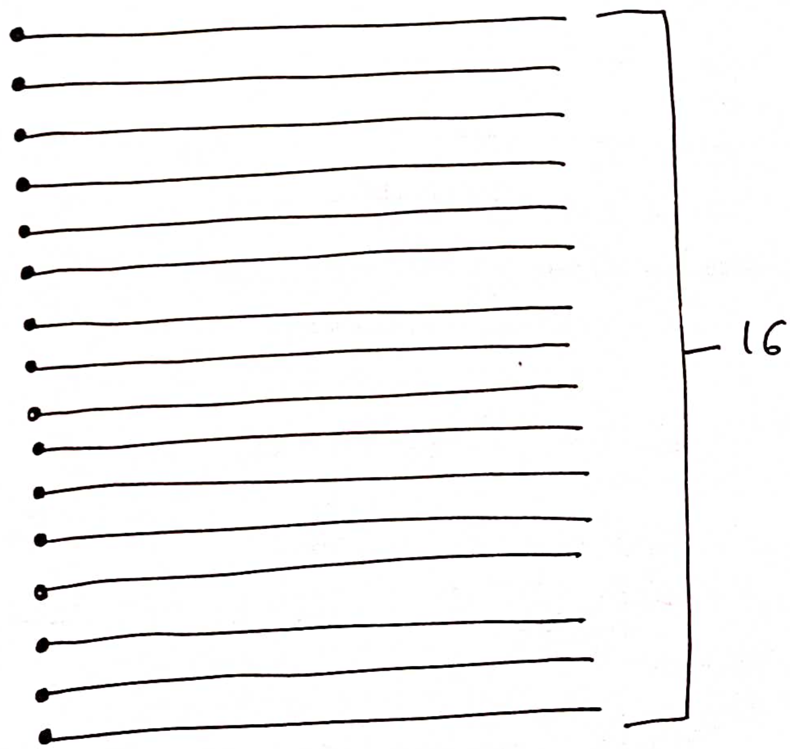
PATA – Paralled Advanced technology Attachment

⇒ Depending upon the system architecture the computer system makes use of different types of Buses.

⇒ The bus consist of group of cables and each of these cable can carry 1-bit (Binary 0 or 1) at a time. And therefore, a bus consist of a group of cables. So that a group of Bits can be sent through the bus at the same time.



8-bit Bus



16-bit Bus

⇒ The size of a Bus is known as its Bus width is important because It determines How much data (in bits) can be transmitted at one time the larger system bus width significantly improves the Processor Performance.

for example:

⇒ A 64-bit Bus has sixty four (64) Cables thus can transmit 64-bits of data at a time.

⇒ whereas a 32-bit bus has thirty two (32) wires and it can transmit 32-bit of data at a time.

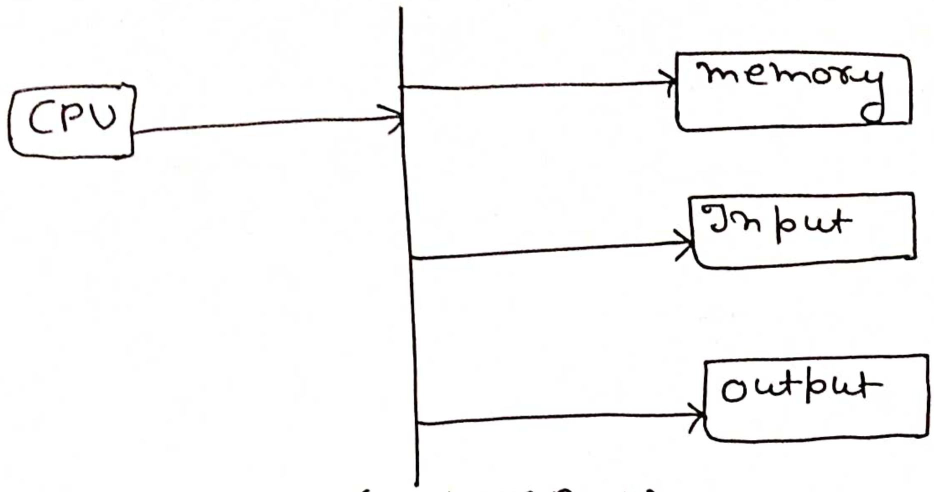
Types of BUS:- The computer system buses can be classified on the basis of type of components being connected as.

1. Data Bus
2. Address Bus
3. control Bus

Buses are used to connect Processor with other component of computer like memory and input output devices / inorder to transfer information Buses are set of wires.

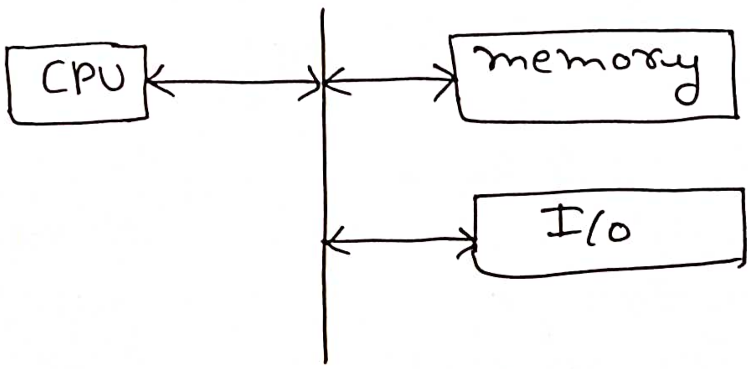
1. Control BUS:- Control Bus transfer the control signal Predefined instruction in specific interval of time from Processor to other components.

This is a unidirectional Bus.



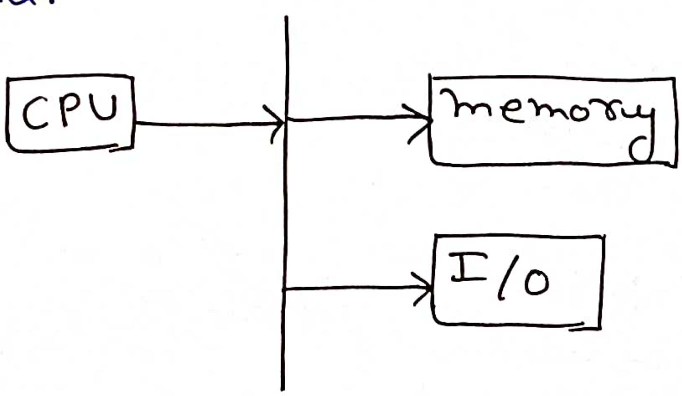
(Control BUS)

② Data Bus :- Data Bus is used to transfer the information or data from one component to another component of computer. Data bus is a Bidirectional Bus.



(Data BUS)

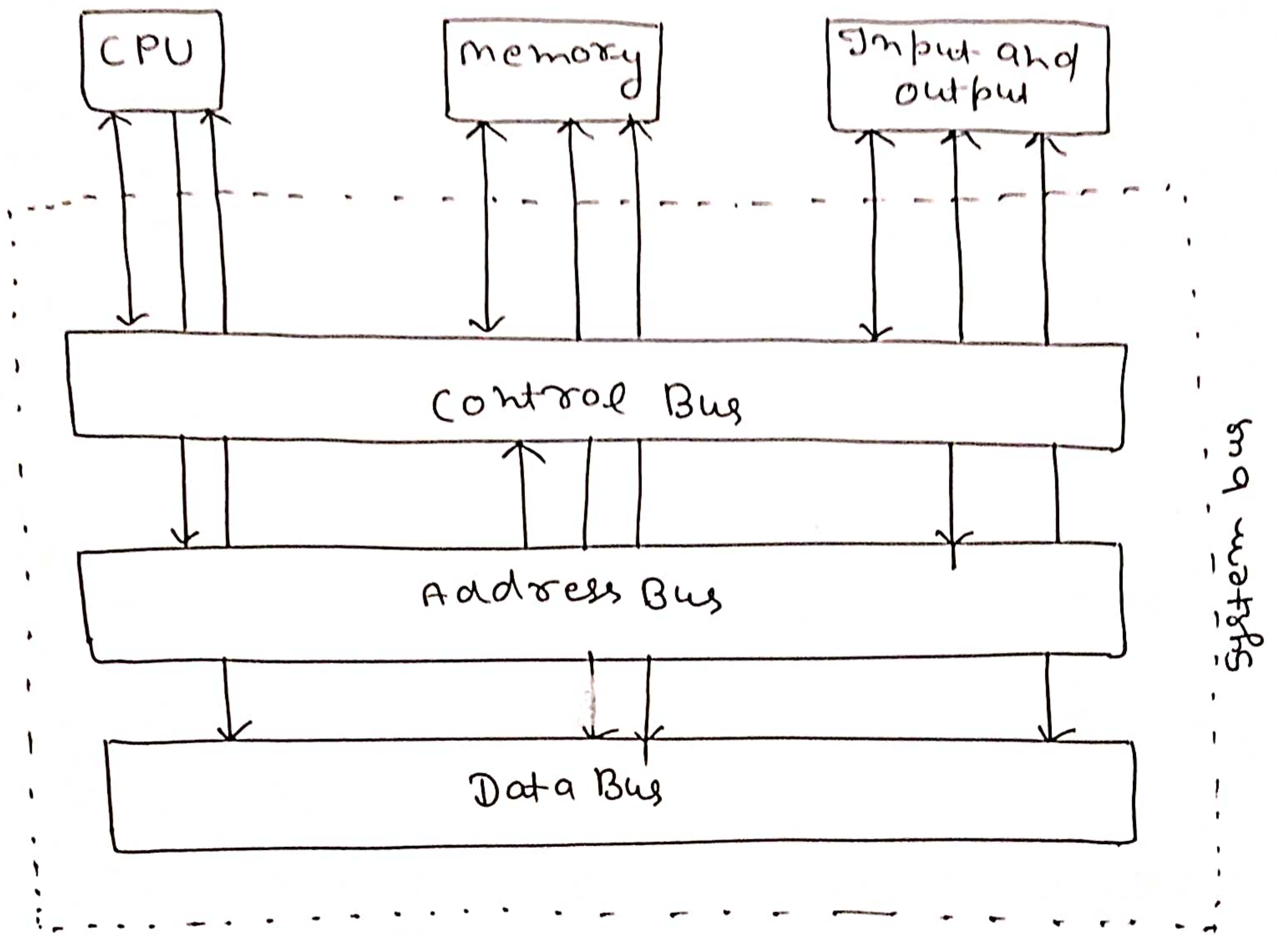
③ Address Bus :- This Bus is used to carry address of different memory location/register or device.



(Address Bus)

Address Bus is unidirectional Bus. becoz address are generated by CPU.

System Bus:- Combination of Address Bus, Control Bus and Data Bus is termed as system bus. generally when we use talk bus it mean to system bus.



BUS ARBITRATION:- Bus arbitration is the Process by which Bus allocation master assign or Release the Bus control for different components. many components may request to access the bus simultaneously that may aries conflict as system buses may be nonsharable.

The bus arbitration is of two types

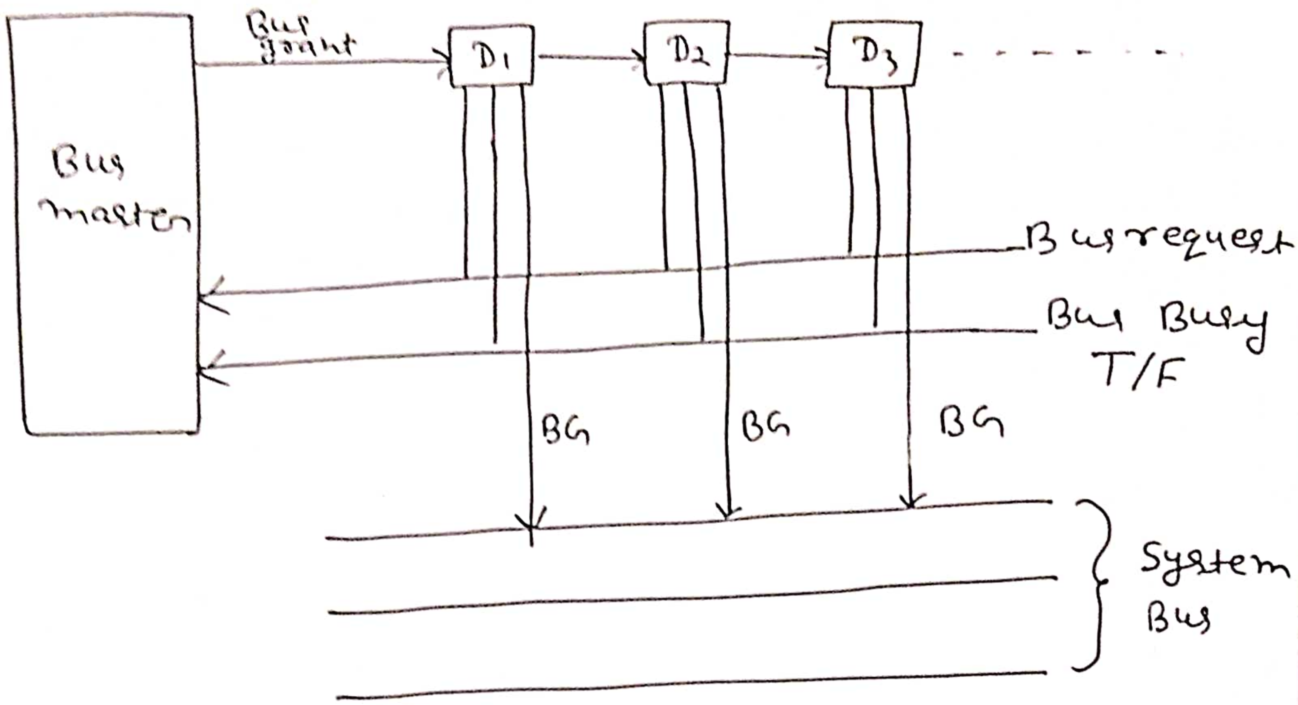
① Centeralized Bus Arbitration :- In this approach a single Bus master or single Bus arbiter controlled to bus arbitration.

There are three methods of centralized Bus Arbitration.

(a) Daisy Chaining method
(b) Palling or Rotationg Priority method.
(c) Fixed Priority or independent Request method.

② Distributed Bus Arbitration :- All Devices Participate in the selection Process of Buses and Bus master.

(a) Daisy Chaining Arbitration :- It is a simple and cheaper method where all the bus masters use the same Line for making bus requests. The bus grant signal serially Propagates through each master until it encounters the first one that is requesting access to the Bus. This master blocks the Propagation of the bus grant signal therefore any other requesting module will not receive the grant signal and hence cannot access the Bus.



Advantages:-

1. Simplicity and scalability.
2. The user can add more devices anywhere along the chain, up to a certain maximum value.
3. Less hardware costs.

Disadvantages:-

1. The value of Priority assigned to a device depends on the position of the master device bus.
2. Propagation delay arises in this method

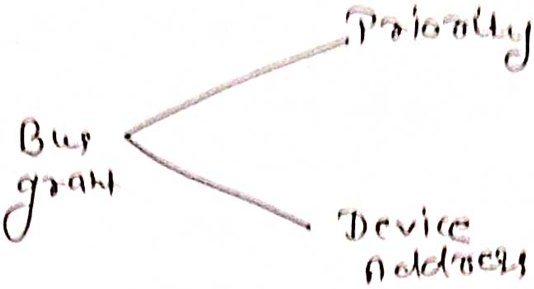
{ Not fair allocation
No fault tolerance }

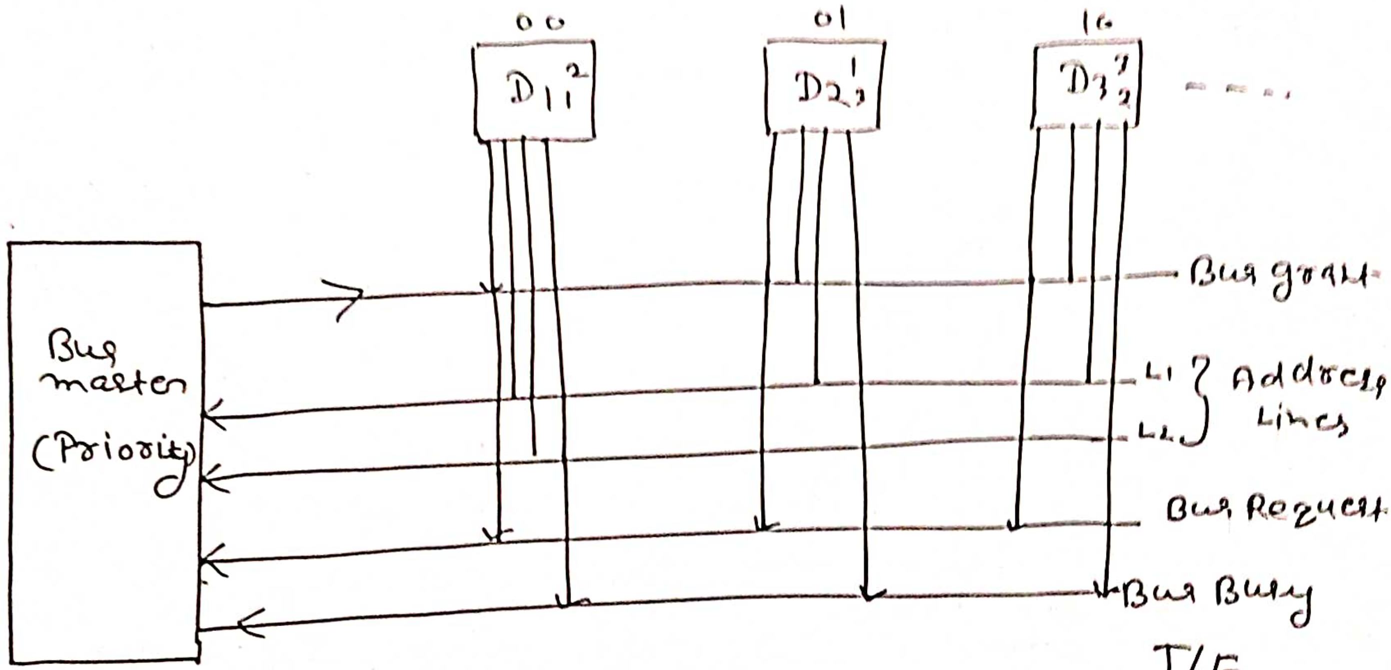
(2) Polling or Rotating Priority method:-

In this method the controller is used to generate the address for the master (unique Priority).

The number of address lines required depends on the number of masters connected in the system. The controller generates sequence of master recognizes its address. It activates the busy Line and begins to use the bus

Dynamic Priority
↓
Priority may change by bus master time to time





T/F

Advantages :-

① fair allocation

② Bus grant which common to all devices.

③ Dynamic Priority

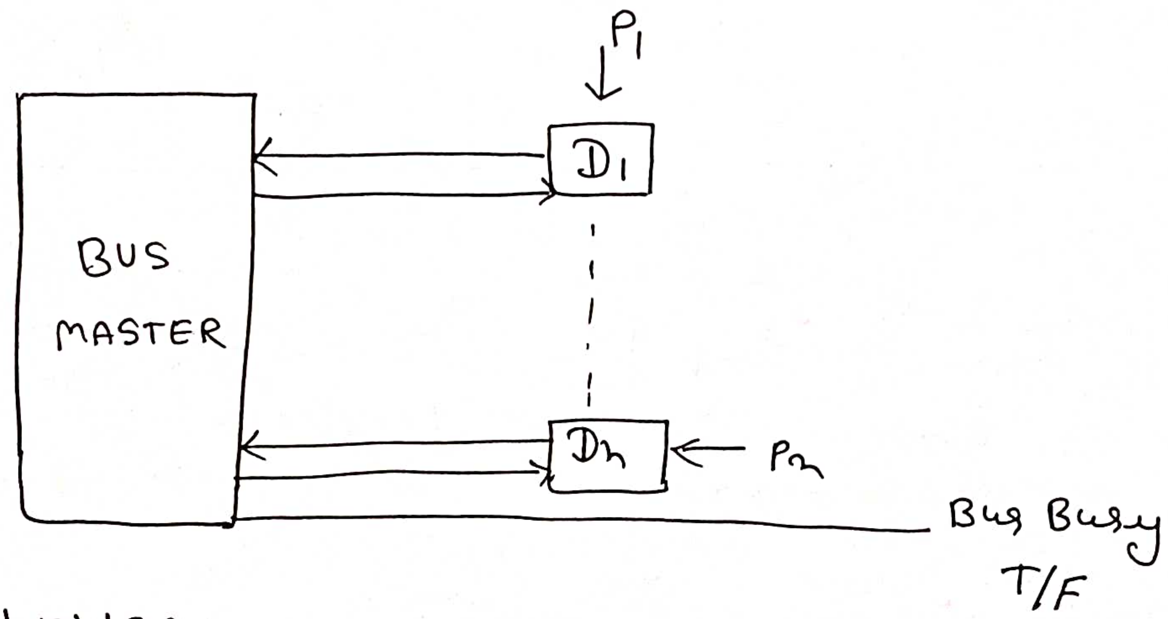
Disadvantages :-

① Hardware cost increase.

② Add Line will increase as the number of devices will increases.

③ Structure will be complex than Daisy Chaining methods.

③ Independent Request method:- In this method each device is directly connected to bus master or bus arbiter. and each device has its on Bus Request Line and bus grant Line. Also every device has Preallocated or fixed Priorities. A device send its Priority with its Bus request signal and whenever Bus is free. bus Arbiter serve the device with Highest Priorities.



Advantages:-

① This method generates a fast response.

② It has fixed Priority.

Disadvantages:- Hardware cost is high as a large number of control lines is required.

CONSTRUCTION OF COMMON BUS USING :-

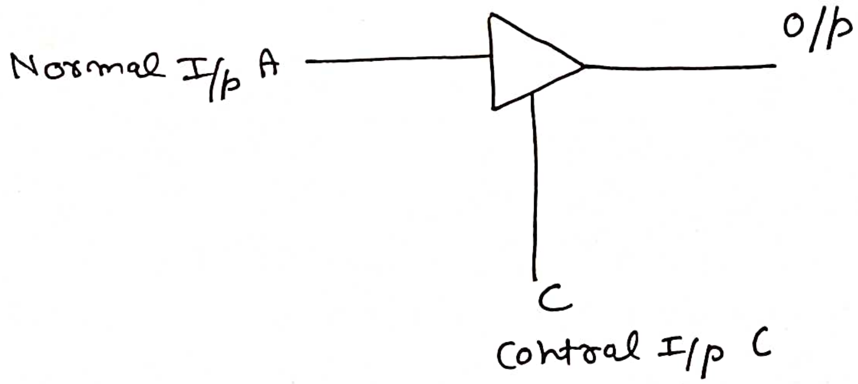
⇒ A bus system can be constructed with three - state gates instead of multiplexers.

⇒ A three - State gate is a digital circuit that exhibits three States.

⇒ two of the States are signals equivalent to Logic 1 and 0 as in a conventional gates. The third State is a High - impedance State.

⇒ The high - impedance State behaves like an open circuit, which means that the output is disconnected and does not have a Logic significance.

uses - Perform as NAND or AND design of a Bus system is the buffer gate.



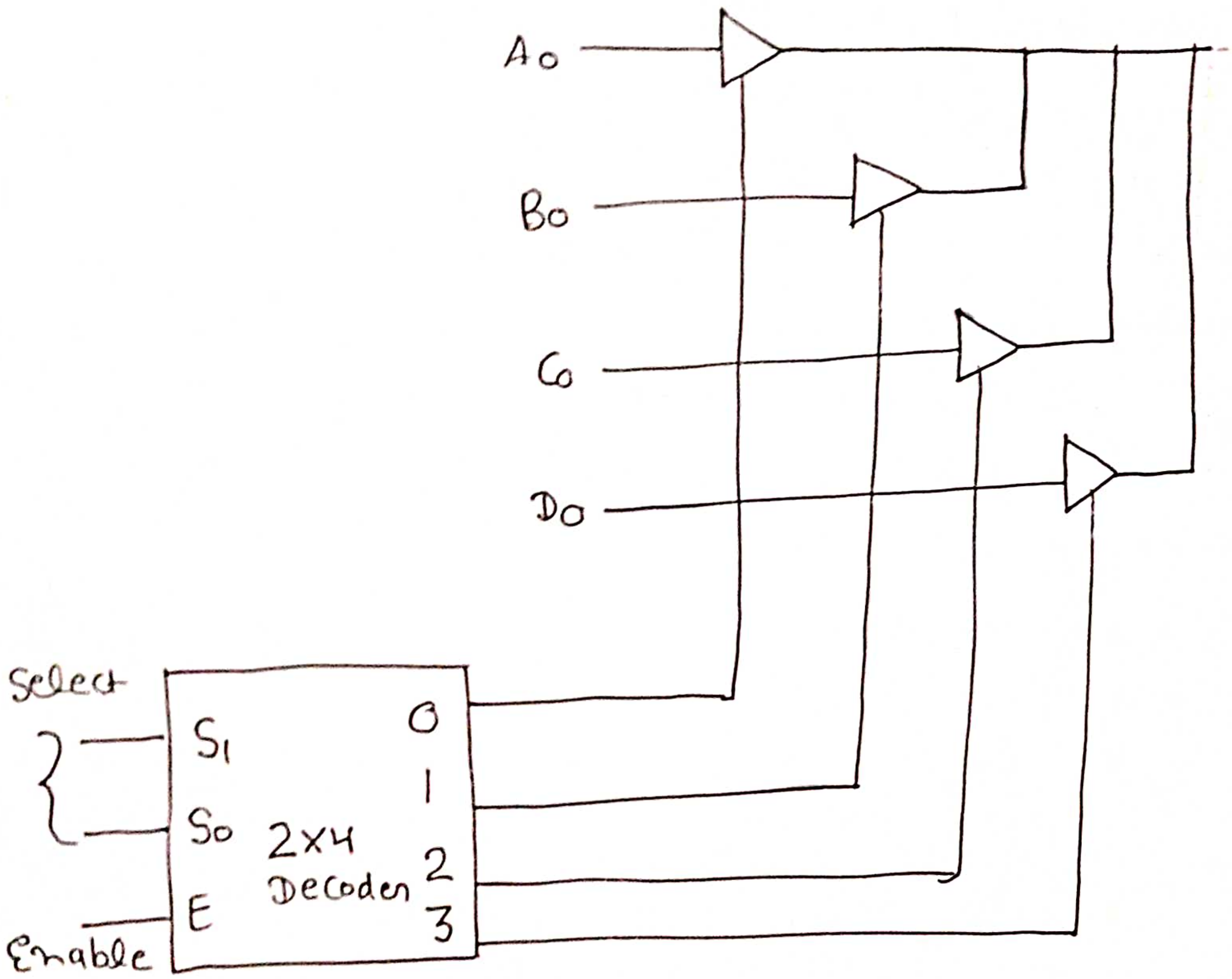
Y = A if C = 1

High impedance if C = 0

{graphical symbol of three state buffer}

The High impedance State of a three - state gate Provides a Special feature not available in other gates.

Because of this feature, a large number of three-state gate o/p can be connected with wires to form a common bus line without endangering loading effects.



{bus line with three state-buffers}

⇒ To construct a common bus for four registers of n bits three state buffers. we need n circuits with four buffers.

⇒ Each group of four buffers receives are significant bit from the four registers

⇒ Each common output produces one of the lines for the common bus for a total of n-lines.

REGISTERS:- Register is a very fast computer memory used to store data or instruction in execution.

A Register is a group of flip-flops and flip-flops are the device which can store 1-bit of information.

following are some commonly used Registers.

① **Output Register:- (OUTR)**

output - Register is used to holds output character. The size of output Register is 8-bit and the symbol for output Register is (OUTR).

② **Input Register:(INPR):-** Input Register is used to Holds Input character. The size of Input Register is 8-bit and the symbol of input Register is (INPR).

③ **Accumulator:-** (only Processor use any one no use this Register).

Accumulator is a Processor Register with size 16-bit. The symbol of accumulator is (AC). It is used by Processor to store Data for short duration.

④ **Program counter:-** This register is used to Hold the address of different instruction. The size of Program counter is 12-bit and the symbol of Program counter is (PC).

(5) Address Register:- Address Registe are used to Hold address of any instruction in memory. The size of Program counter is 12-bit and the symbal of Program counter is (PC).

(6) Data Register:- Data Register are used to hold the memory operands the values (used) in execution of any instruction) The size of Data Register is 16-bit. The symbal of Data Register is (DR) (MDR)

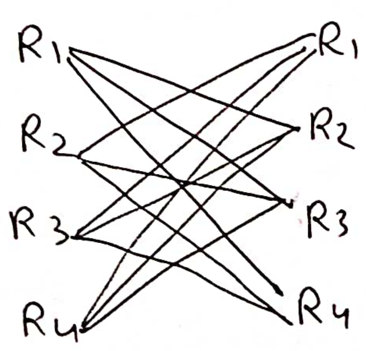
(7) IR (Instruction Register):- This Register is used to store the instruction code. This Register is of 16-bit and the symbol of this Register is (IR).

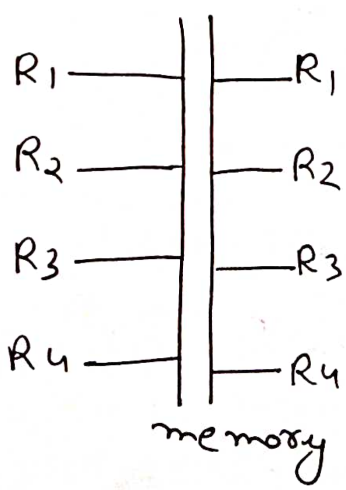
(8) Temporary Register:- Temporary Register are used to Store temporary values. The size of Store temporary Register is 16-bit. The symbol of Temporary Register is (TR).

REGISTER: BUS & MEMORY TRANSFER:-

How we transfer the data of a Register to memory or any other register using Bus.

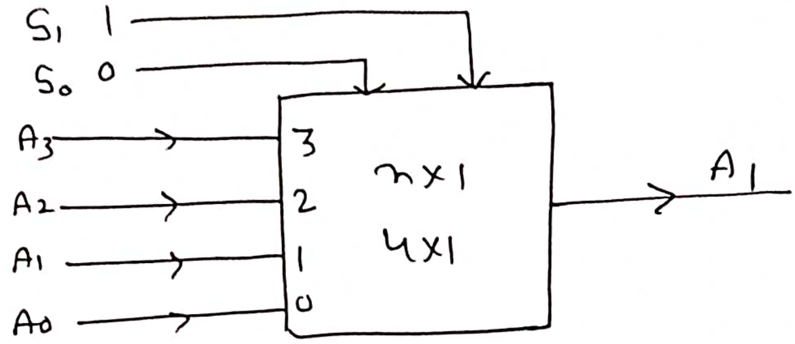
* Common Bus with the help of MUX (multi-blexer)





multiplexers :- The multiplexers select the source registers whose binary information is then Placced on the bus.

2^{power} (block diagram) circuit.



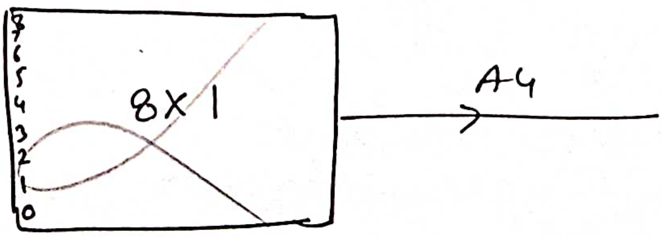
(n must in power of 2

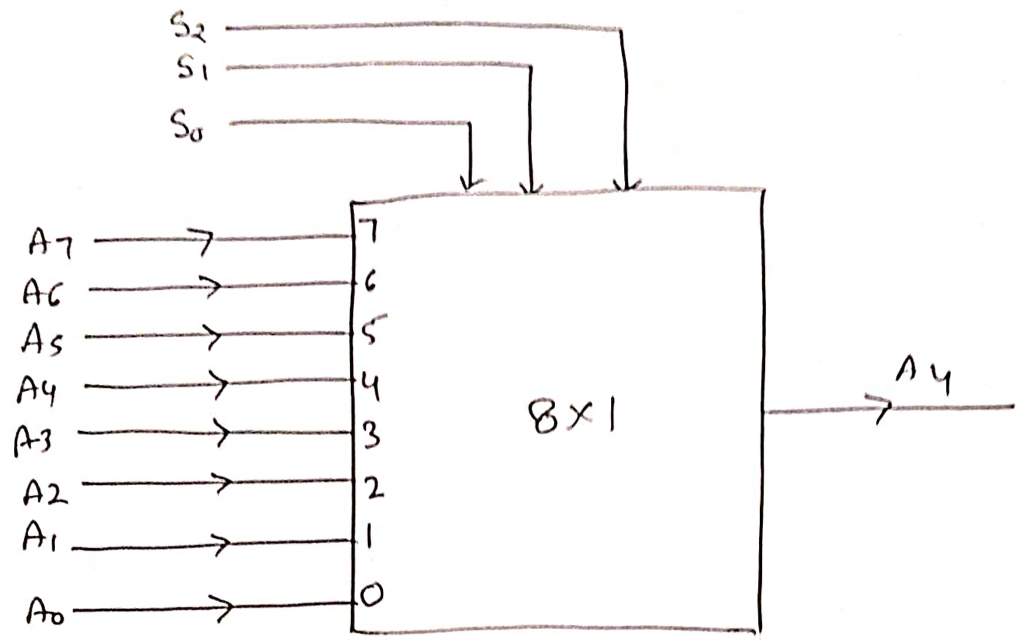
$n = 2^x \rightarrow$ No of selection Line

$4 = 2^{2}$ → selection Line

S_0	S_1	
0	0	0
0	1	1
1	0	2
1	1	3

Q_3 8×1 MUX Design How many selection line and output.





$n = 2^x \rightarrow$ no of selection Line

$8 = 2^{3} \rightarrow$ No of selection Line

S_0	S_1	S_2	
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

* No of multiplexer is equal to No of Bit in Register and
* No of input in multiplexer is equal to number of Register. if we have n register

then we have to take mux of n×1.

BUS AND MEMORY TRANSFER:

⇒ A bus structure consists of set of common lines, one for each bit of a register, through which binary information is transferred one at a time.

⇒ Control signals determine which register is selected by the bus during each particular register transfer.

⇒ One way of constructing a common bus system is with multiplexers. The multiplexers select the source register whose binary information is then placed on the bus.

e.g The construction of bus system for four registers.

S_1	S_0	Register Selected
0	0	A
0	1	B
1	0	C
1	1	D

⇒ A bus system will multiplex K-registers of n bits each to Produce an n-line common bus.

⇒ The number of multiplexers needed to construct the bus is equal to n, the number of bits in each register.

⇒ The size of each multiplexer must be $K \times 1$ since it multiplexers K data Line.

⇒ The symbolic statement for a bus transfer may mention the bus or its Presence may be implied in the statement.

⇒ when the bus is included in the statement, the register transfer is symbolized as follows:

$BUS \leftarrow C \qquad R_1 \leftarrow BUS$

The content of Register C is Placed on the Bus and the Content of the bus is loaded into register R_1 by activating its load control Iinput.

to show direct Transfer like $R_1 \leftarrow C$ is converted.

PROCESSOR ORGANIZATION :- A Processor must have 3 functional units to be what to be what we call a computer ! a unit that performs arithmetic and Logical operations on data i.e ALU, a unit that remembers data while it is not being worked i.e memory, and a unit which Sequences the operations Performed by the ALU i.e sequencer.

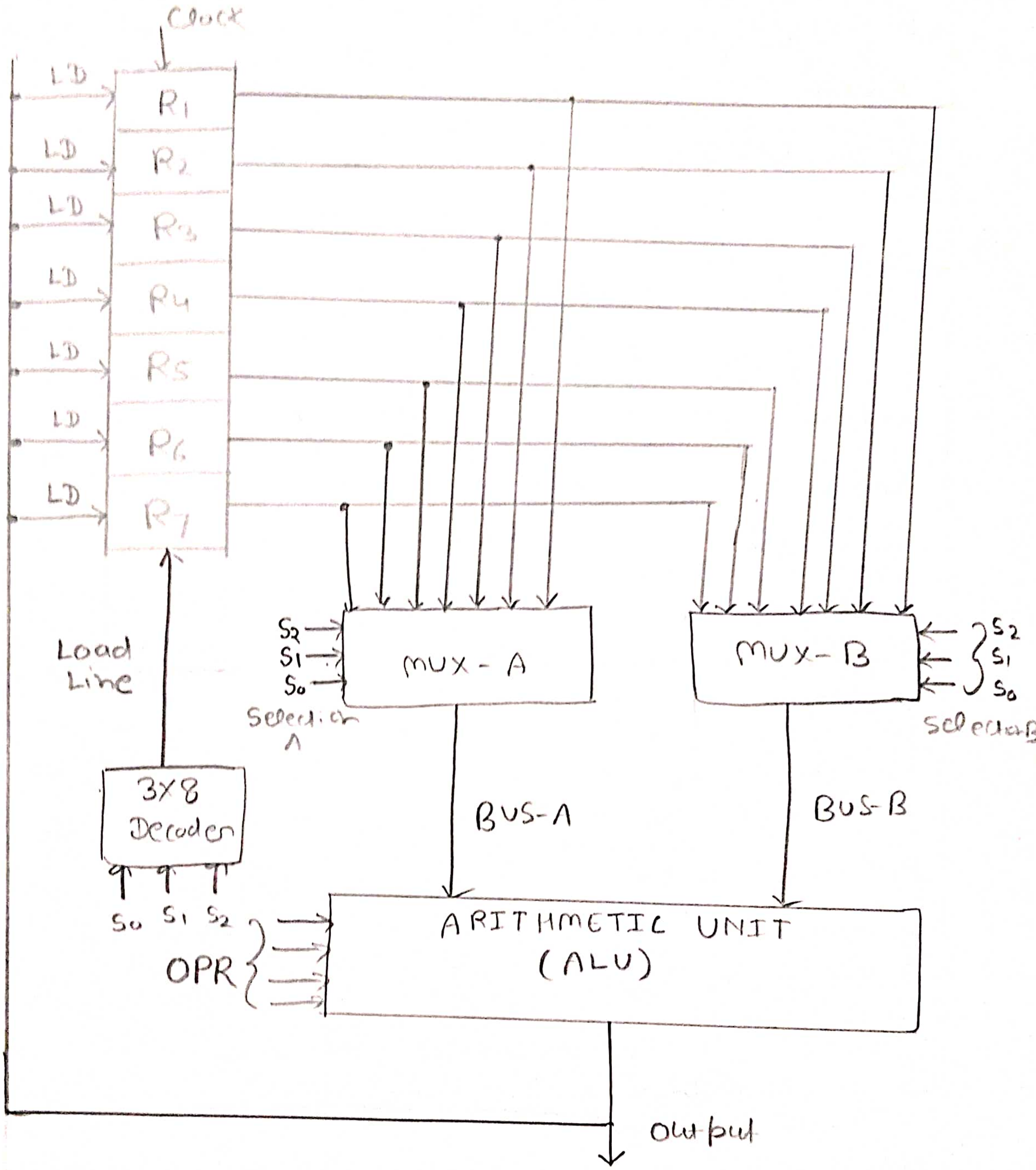
Processor organization is a term describing How those three elements are implemented and How they interconnect to accomplish their tasks. The basic Processor organization is of fallowing type

General Register organization :- Generally CPU has seven general Registers. Register organization Show how registers are selected and how data flow between register and ALU.

A decoder is used to select a Particular register. The output of each register is connected to two multiplexers to form the two buses A and buses B. The selection lines in each multiplexer select the input data for the Particular bus.

The A and B buses form the two input of an ALU. The operation Select lines decide the micro operation to be Performed by ALU.

The result of the micro operation is available at the output bus. The output bus connected to the inputs of all Registers. Thus by Selection a destination registe It is Possible to store the result in it.



Example: To perform the operation $R3 = R1 + R_2$ we have to provide following selection variable to selection inputs

SELECTION : 001

(1) SEL A : 001 - To place the contents of R_1 into BUS A.

(2) SEL B : 010 - To Place the contents of R_2 into BUS B.

(3) SEL OPR : 10010 - to Perform the arithmetic addition A + B

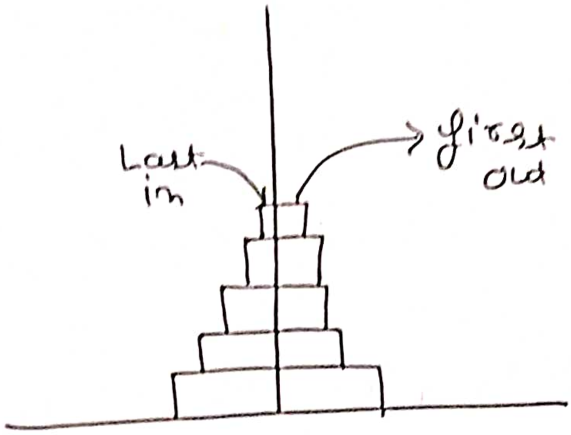
(4) SEL REG or SELD : 011 - To Place the Result available on output bus in R3.

Register and multiplexer input selection code:

Binary Code	SEL-A	SEL-B	SEL-D OR SEL-REG
000	Input	Input	- - - - -
001	R_1	R_1	R_1
010	R_2	R_2	R_2
011	R_3	R_3	R_3
100	R_4	R_4	R_4
101	R_5	R_5	R_5
110	R_6	R_6	R_6
111	R_7	R_7	R_7
	- - - -		

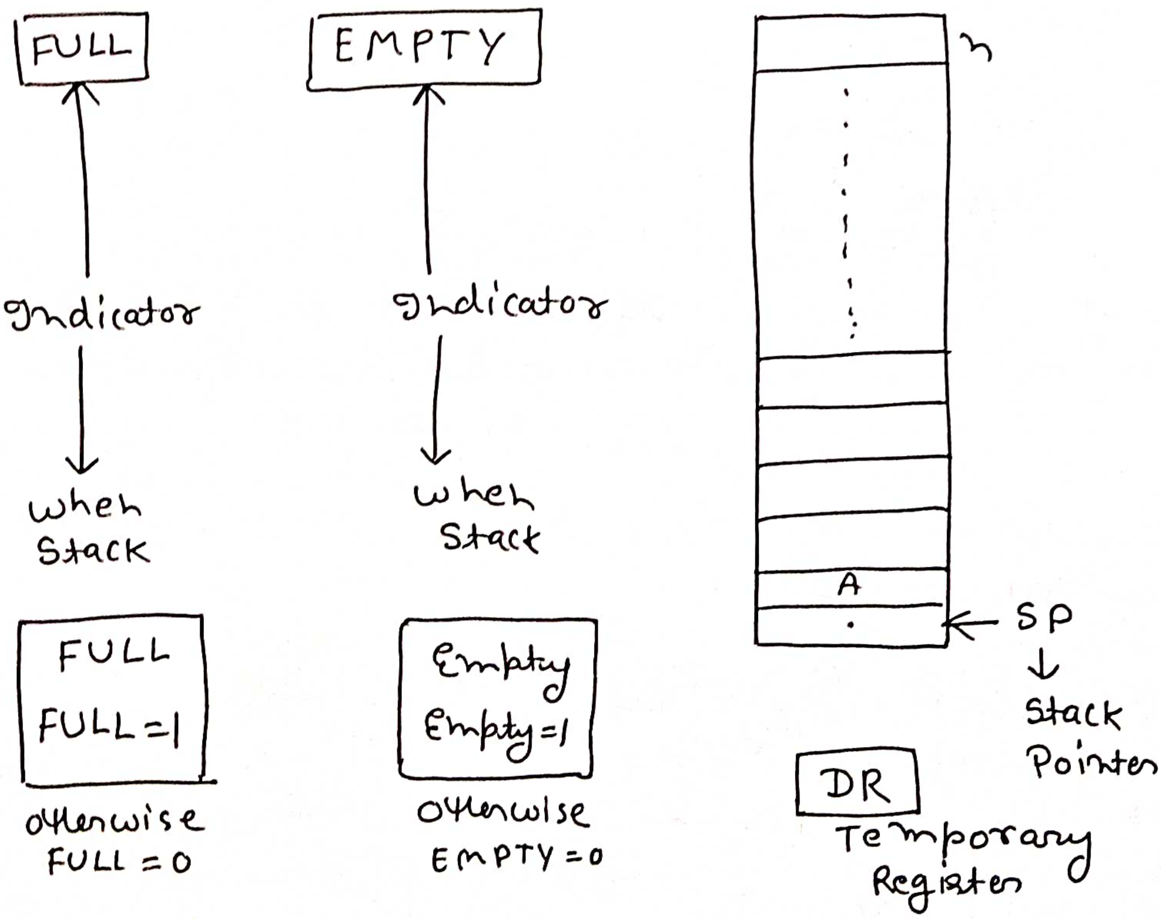
STACK ORGANIZATION OF REGISTERS :-

Stack :- The way do organasing data in which we follow Last in first out (LIFO)



Insertion is called PUSH operation and deletion is called POP operation.

The concept of Stack can be used to orgnaise the Registers in computer organization.



PUSH (INSERT)	POP (DELETE)
FULL = 0 EMPTY = 1	FULL = 1 EMPTY = 0
① if EMPTY = 1	① if FULL = 1
② SP → SP+1	(2) DR ← M(SP)
③ M(SP) ← DR (write item on the ~~to~~ top of the stack)	(3) SP ← SP−1
④ if (SP = N−1) FULL = 1 ← EMPTY = 0 → Check if stack is full	(4) if (SP = 0) EMPTY = 1 FULL = 0
⑤ stop	(5) Stop

ADDRESSING MODES :-

⇒ The way the operands are chosen during Program execution dependent on the addressing mode of the instruction.

⇒ The addressing mode specifies a rule for interpreting or modifying the address field of the instruction before the operand is actually referenced.

⇒ The availability of the addressing modes.

The operation field of an instruction specifies the operation to be Performed. This operation will be executed on some data which is stored in computer registers or the main memory. The way any operand is selected during the Program execution is dependent. The Purpose of using addressing mode of the instruction. The Purpose of using addressing modes is as follows.

① To give the Programming versatility to the user.

② To reduce the number of Bits in addressing field of instruction.

Types of Addressing modes:- In computer architecture, there are following types of addressing modes.

① Stack Addressing mode
② Immediate Addressing mode.
③ Direct Addressing mode.
④ Indirect Addressing mode.
⑤ Register Direct Addressing mode.
⑥ Register Indirect Addressing mode.
⑦ Register Relative Addressing mode
⑧ Indexed Addressing mode
⑨ Base Register Addressing mode.
⑩ Auto - Increment Addressing mode.
⑪ Auto - Decrement Addressing mode.
⑫ Auto - Decrement Addressing mode.

1 Stack Addressing mode:- In this addressing mode

⇒ The operand is contained at the Top of the stack.

example

ADD

⇒ This instruction simply pops out two symbols contained at the top of the stack.

⇒ The addition of those two operands is performed.

⇒ The result so obtained after addition is pushed again at the top of the stack.

2 Immediate Addressing mode:-

In this addressing mode,

* The operand is specified in the instruction explicitly.

* Instead of address field, an operand field is present that contains the operand.

opcode	operand

Immediate Addressing mode.

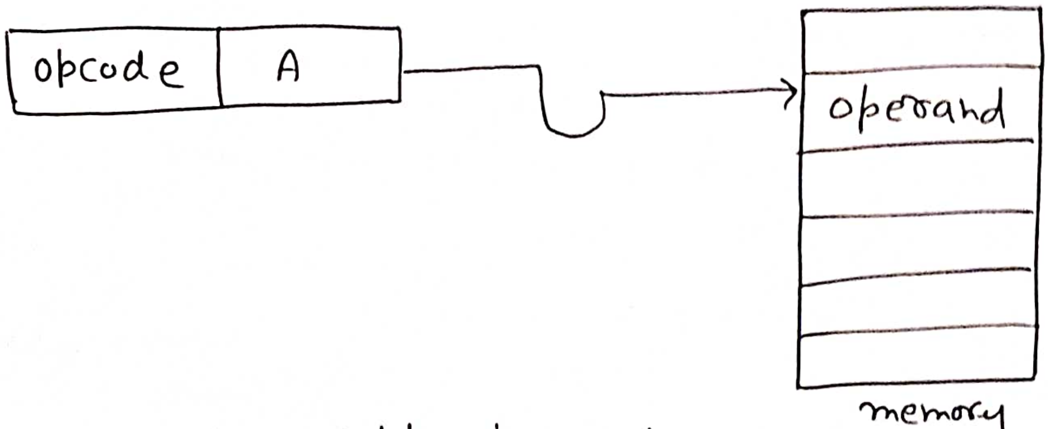
Example:

⇒ ADD 10 will increment the value stored in the accumulator by 10.

⇒ MOV R #20 initializes register R to a constant value 20.

③ Direct Addressing Mode:- In this addressing mode

* The Address field of the instruction contains the effective address of the operand.
* only one reference to memory is required to fetch the operand.
* It is also called as absolute addressing mode.



{Direct Addressing mode}

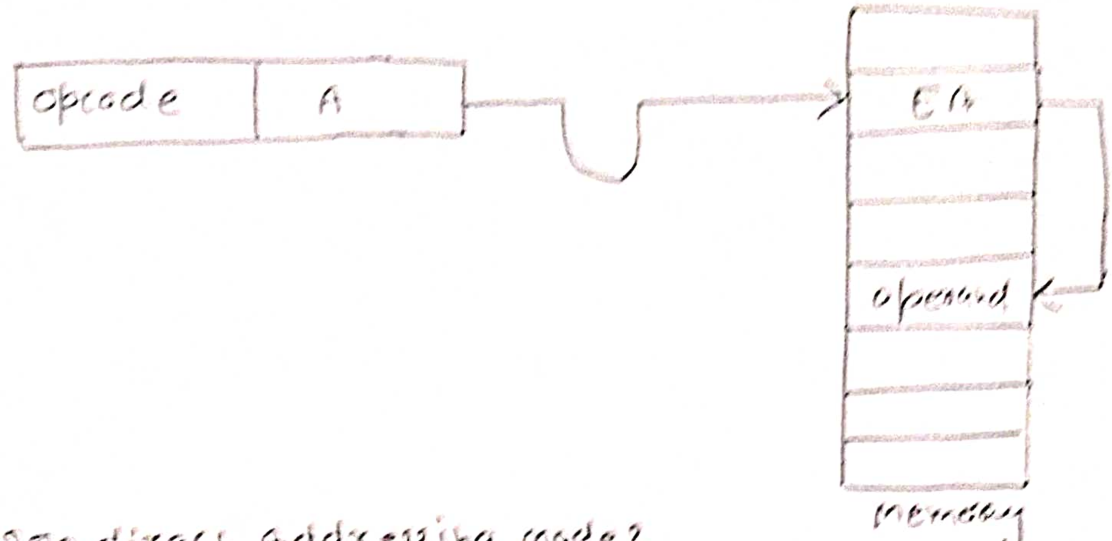
Example ADD X will increment the value stored in the accumulator by the value stored at memory location X.

AC ← AC + [X]

④ Indirect Addressing mode:- In this addressing mode.

⇒ The address field of the instruction specifies the address of memory location that contains the effective.
address of the operand

⇒ Two references to memory are required to fetch the operand.



{Indirect Addressing mode}

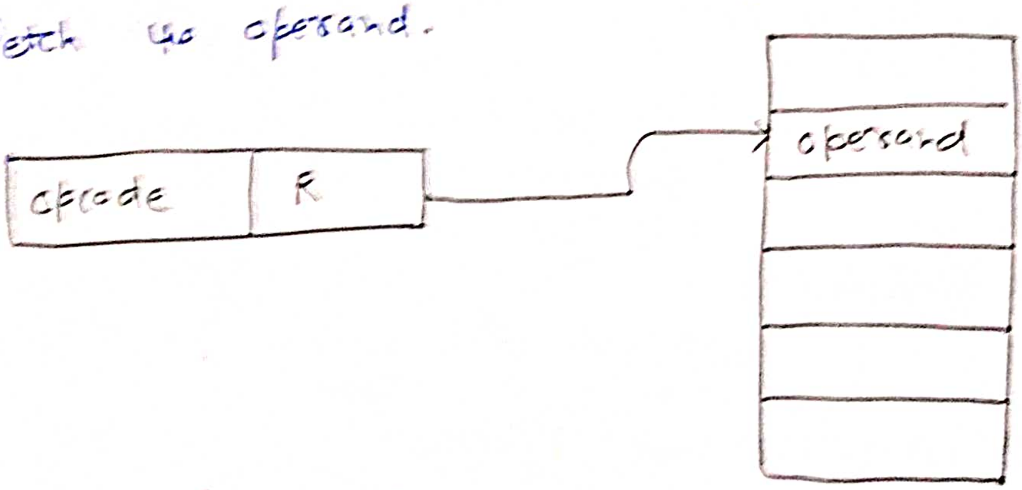
ADD X will increment the value stored in the accumulator by the value stored at memory location specified by X.

Register Direct Addressing mode :- In this, addressing mode.

⇒ The operand is contained in register set.

⇒ The address field of the instruction refers to a CPU register that contains the operand.

⇒ No reference to memory is required to fetch the operand.



{Register Direct Addressing mode}

- ADD R will increment the value stored in the accumulator by the content of Register R

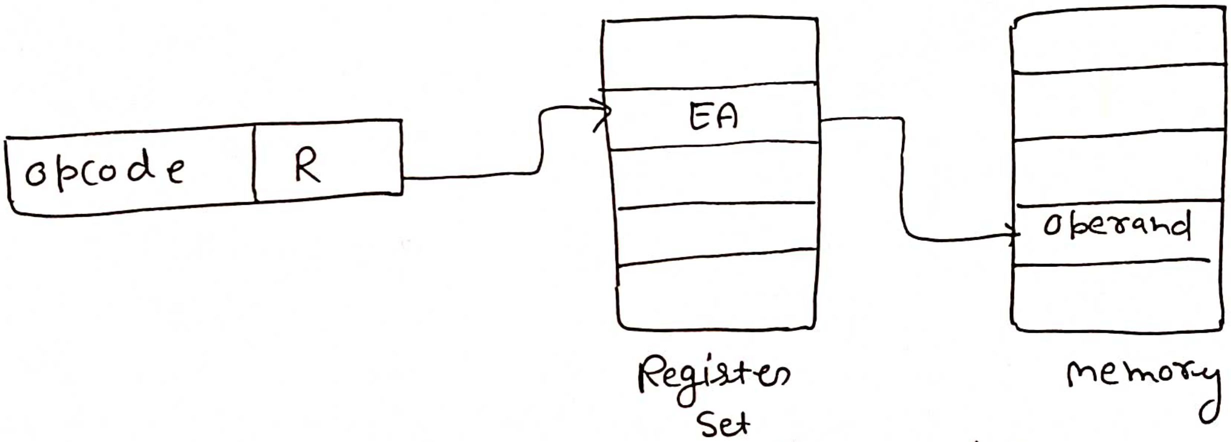
$AC \leftarrow AC + [R]$

Note: This addressing mode is similar to direct addressing mode.
The only difference is address field of the instruction refers to a CPU register instead of main memory.

Register Indirect Addressing mode:- In this addressing mode:

⇒ The address field of the instruction refers to a CPU register that contains the effective address of the operand.

⇒ only one reference to memory is required to fetch the operand.



{Register Indirect Addressing mode}

Example:- ADD R will increment the value stored in the accumulator by the content of memory location specified in register R.

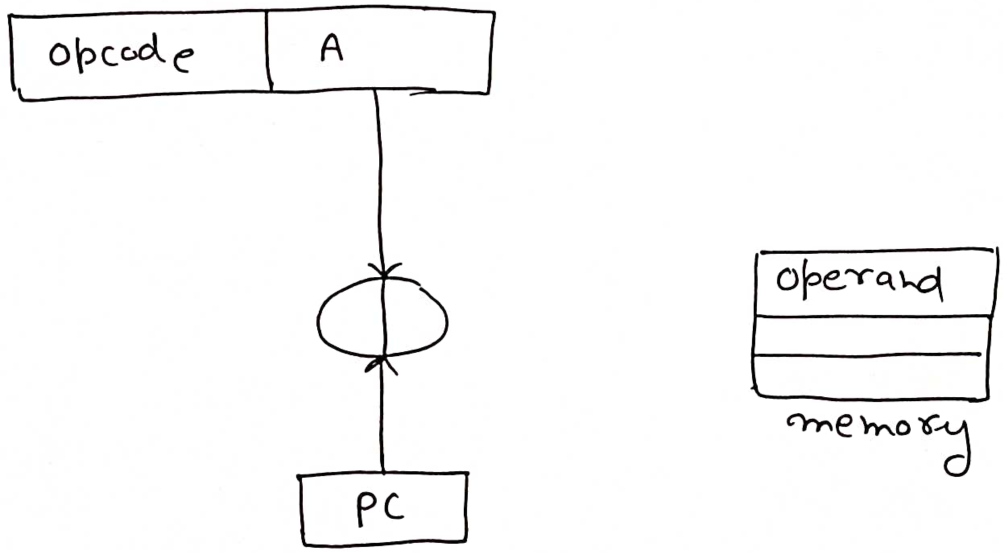
$$AC \leftarrow AC + [[R]]$$

Note! This addressing mode is similar to indirect addressing mode.

⇒ The only difference is address field of the instruction refers to a CPU register.

Relative Addressing mode:- In this addressing mode
⇒ effective address of the operand is obtained by adding the content of Program counter with the address part of the instruction.

effective Address = Content of Program counter + Address Part of the instruction



{ Relative Addressing mode }

⇒ Program counter (PC) always contains the address of the next instruction to be executed.

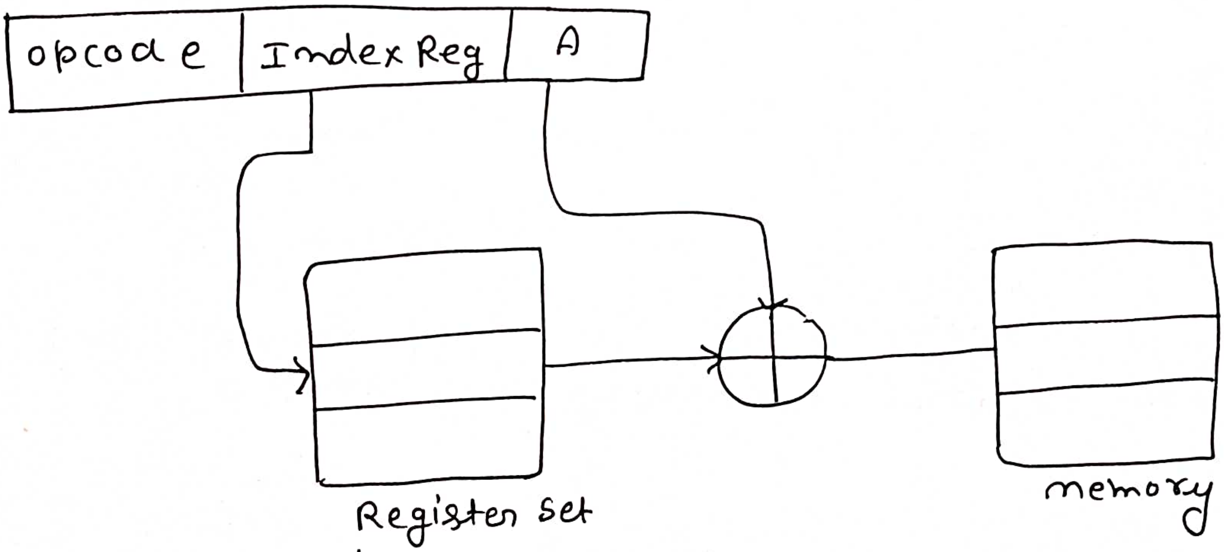
⇒ After fetching the address of the instruction, the value of Program counter immediately increases.

⇒ The value increases irrespective of whether the fetched instruction has completely executed or not.

Indexed Addressing mode:- In this Addressing mode.

⇒ effective addressing of the operand is obtained by adding the content of index register with the address part of the instruction.

effective Address = Content of Index Register + Address part of the Instruction

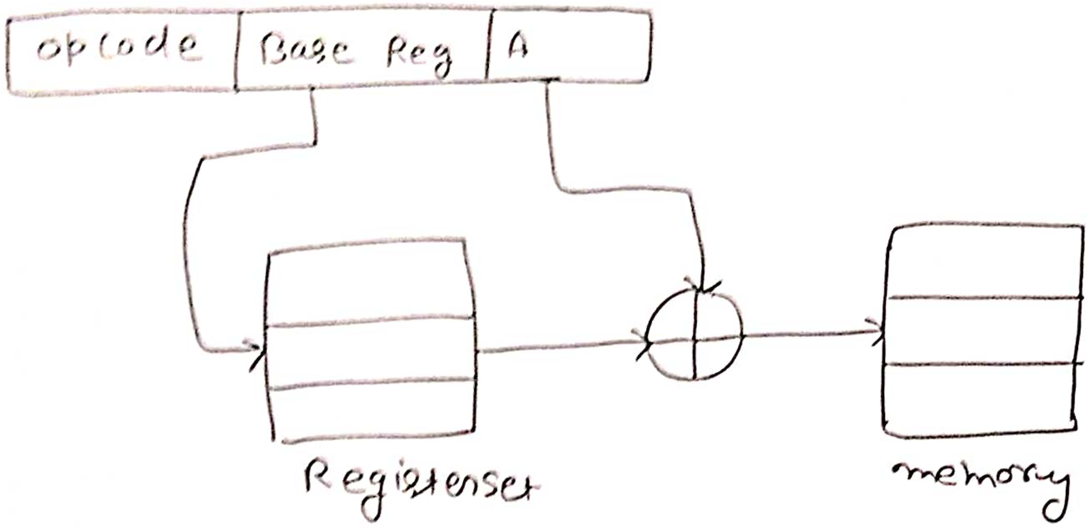


{indexed Addressing mode}

Base Register Addressing mode:- In this Addressing mode

⇒ Effective Address of the operand is obtained by adding the content of base register with the address part of the instruction.

effective Address = content of Base Register + Address Part of the instruction



{Base Register Addressing mode}

Auto - Increment Addressing mode:-

This Addressing mode is a special case of Register indirect Addressing mode where

$\boxed{\text{effective Address of the operand = Content of Register}}$

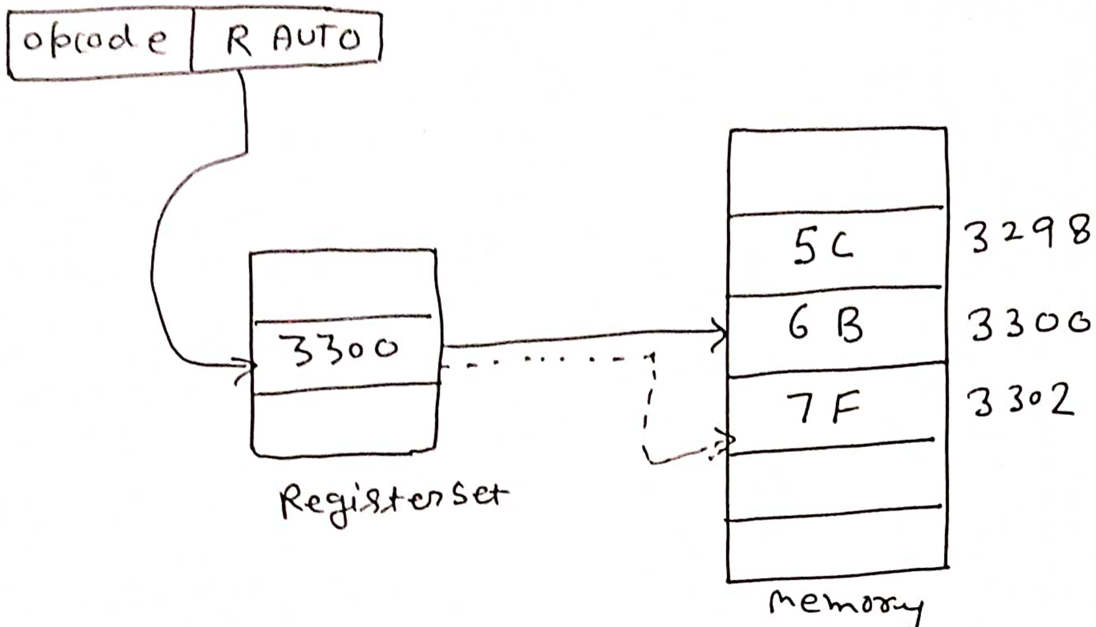
In this Addressing mode

⇒ After accessing the operand the content of the register is automatically incremented by step size d.

⇒ Step size 'd' up depends on the size of operand accessed.

⇒ only one reference to memory is required to fetch the operand.

Example:



{ Auto-increment Addressing modes

Assume operand size = 2 bytes

Here
⇒ After fetching the operand 6B, The instruction register RAUTO WILL Be automatically INCREMENTED by 2.

* Then, update value of RAUTO will be automatically 3300 + 2 = 3302

* At memory address 3302 the next operand will be found.

NOTE:- first the operand value is fetched. then the instruction register RAUTO value is incremented by step size 'd'.

AUTO - DECREMENT ADDRESSING MODE:- This Addressing mode is again a special case of Register Indirect Addressing mode where-

$$\boxed{\text{effective Address of the operand} = \text{Content of Register} - \text{Stepsize}}$$

In this addressing mode,

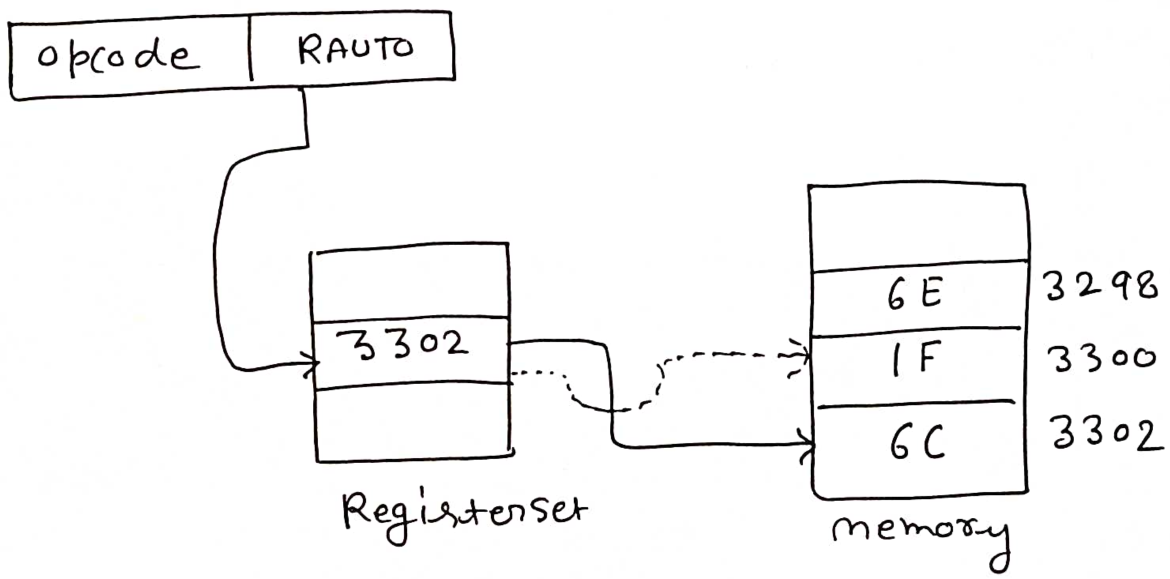
⇒ first, the content of the register is decremented by step size 'd'.

⇒ Step-size 'd' depends on the size of operand accessed.

⇒ After decrementing the operand is read.

⇒ only one reference to memory is required to fetch the operand.

Example:



{Auto decrement Addressing mode}

Assume operand size = 2 bytes

Here
* first, the instruction register RAUTO will be decrement by 2.
* Then updated value of RAUTO will be
3302 - 2 = 3300

* At memory address 3300, the operand will be found.

NOTE: first the instruction register RAUTO value is decremented by step size 'd'.

* Then the operand value is fetched.

Application of Addressing modes:-

Immediate Addressing mode:- To initialize registers to a constant value.

Direct Addressing mode:- To access static Data.

& Register Direct Addressing mode: To implement variables.

Indirect Addressing Mode:- To implement pointers because pointers are memory locations that store the address of another variables.

Register Indirect Addressing mode:- To Pass array as a Parameter because array name is the base address and pointer is needed to point the address.

Relative Addressing mode:-

* for Program relocation at run-time ie for Position independent code.

* To change the normal sequence of execution of instructions.

* for branch type instructions, since it directly updates the Program counter.

Index Addressing mode:-

⇒ for array implementation or array addressing
⇒ for records implementation.

Base register Addressing mode:-

⇒ for writting relocatable code i.e for relocation of Program in memory even at run time.
* for handling recursive Procedures.

Auto- increment Addressing mode:-

* for implementing loops.
* for stepping through arrays in a loop.

AUTO - decrement Addressing mode:-

* for implementing a stack as PUSH and POP.