

## Introduction to Microprocessor

### Microprocessor Structure

Microprocessor consists of Bus, Registers, ALU, Control Unit, Memory.

### Unit 1

Introduction : Functional units of digital System & their interconnections , Buses, Bus architecture , Types of Bus & Bus Arbitration, Register , Bus and memory transfer. Processor Organization , general registers organization , stack organization & Addressing Modes.

### Functional Units

ALU, Register, Bus

Register, Bus

Register, Bus

Register, Bus

### Bus Organization

CPU, ALU, Register

CPU, Register, Register

CPU, Register, Register

Memory, Bus, Register & ALU

CPU, Register, Memory

## Computer Organization Vs Computer Architecture

### Computer Architecture

① Functional description of requirements.

② Design implementation for the various parts of computer.

③ Deals with functional behaviour of computer system.

④ comes before the computer organisation

### Computer Organization

comes after the design of computer architecture

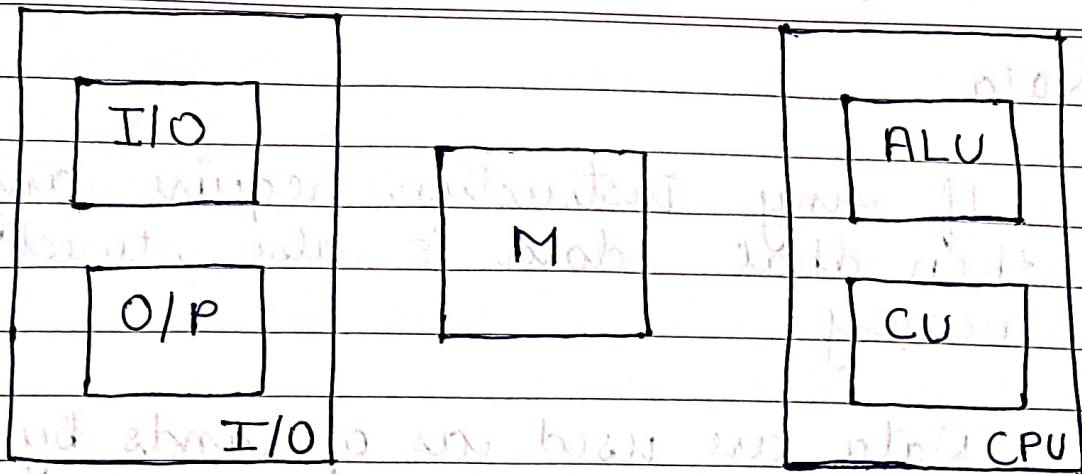
How operational attribute are linked together & contribute to realize the architectural specification.

Deals with structural relationship.

How the functional units collectively work together to execute a computer program/instruction.

## Functional Units

- ① Input Device (I/O)
- ② Output Device (O/P)
- ③ Memory Unit (M)
- ④ Arithmetic & Logic Unit (ALU)
- ⑤ Control Unit (CU)



## Basic functional units of a computer

### Information Handled by a Computer

#### ► Instructions / machine instructions

Govern the transfer of information within a computer as well as between the computer and its I/O devices.

Specify the ALU operations to be performed

Program - set of instructions { I<sub>0</sub>, I<sub>1</sub>, ... } stored in memory and fetched from memory and processed by the processor (CPU).

After fetching the instruction, that instr. will be decoded and various control signals will be issued & based on control signals different components of the system will perform their task.

## ➤ Data

If any instruction require any data then that data is also stored in the memory.

➤ Data are used as operands by the instructions.

These instructions are written in any particular language like C, Java so machine can't understand directly.

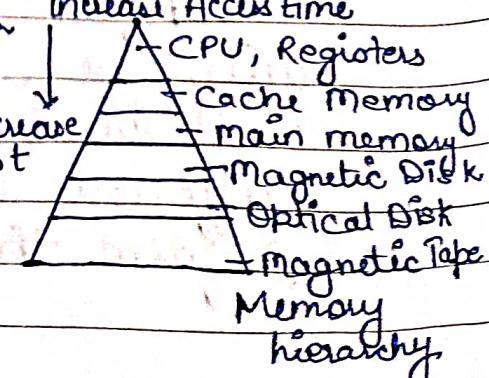
As machine can only understand its own machine language that is why after fetching the instruction it is decoded in a sequence of 0 or 1 & then these sequence is passed to the CPU for execution and accordingly control signals are issued and then various microoperations are performed.

## ➤ Source program

Encoded in binary code — 0 and 1.

### Memory Unit

- Stores programs & data.
- Two classes of storage.



## (1) Primary storage

- Fast
- Programs must be stored in memory while they are being executed.
- Processed in words.
- Ram & memory access time

Memory hierarchy:

Cache, Main memory

## (2) Secondary storage

- Larger & cheaper

## Arithmetic & Logic Unit (ALU)

- Most computer operations are executed in ALU of the processor.
- Load the operands into memory - bring them to the processor - perform operation in ALU - store the result back to memory or retain it in the processor.
- Registers
- Fast control of ALU

## Control Unit

- All computer operations are controlled by the control unit.
- The timing signals that govern the I/O transfers

are also generated by the control unit.

### Operations of a computer

- ① Accept information in the form of programs & data through an input unit & store it in the memory.
- ② Fetch the information stored in the memory under program control into an ALU, where the information is processed.
- ③ Output the processed information.

### A Typical Instruction

① Add LOCA, R0       $\leftarrow$  inst<sup>n</sup> given in assembly language.

↑  
meaning of instruction

② Add the operand at memory location LOCA to the operand in a register R0 in the processor & then place the sum into the register R0.

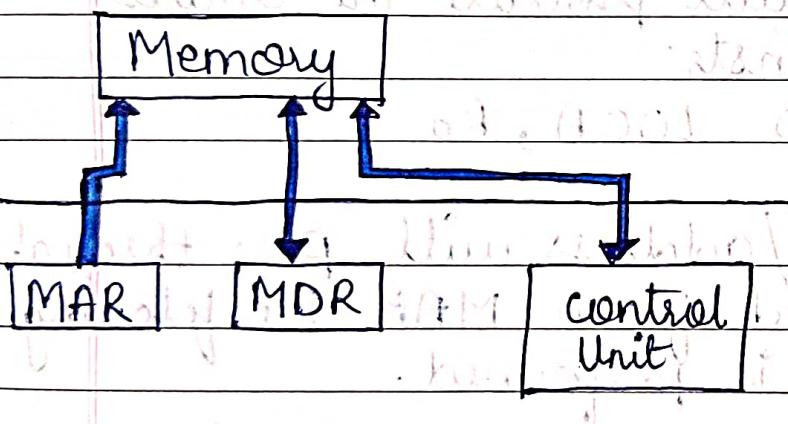
$$\text{i.e. } R_0 = M[A] + R_0$$

③ Original contents of LOCA are preserved.

④ Original contents of R0 is overwritten.

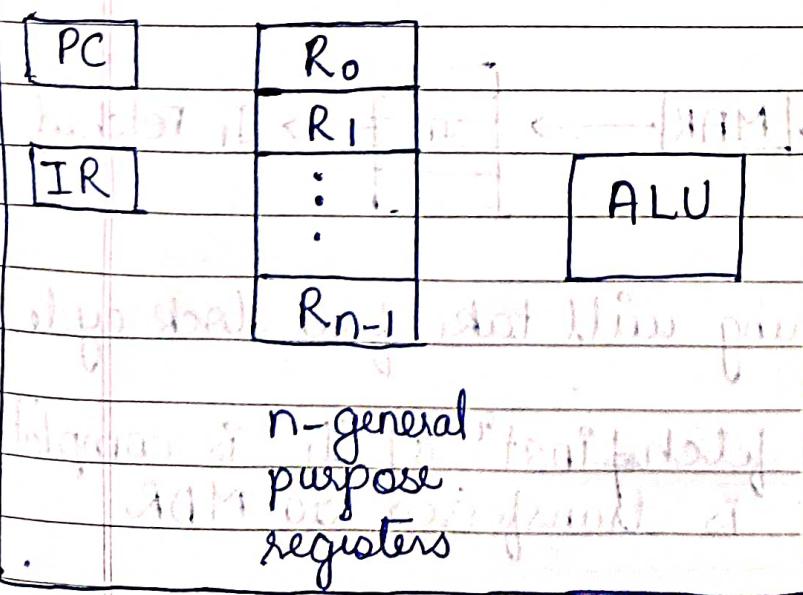
- (5) Inst' is fetched from the memory into the processor.
- (6) The operand at LOCA is fetched & added to the contents of R0.
- (7) Resulting sum is stored in Register R0.

### Connection between Processor & Memory



MAR - Memory address register

MDR - Memory data register

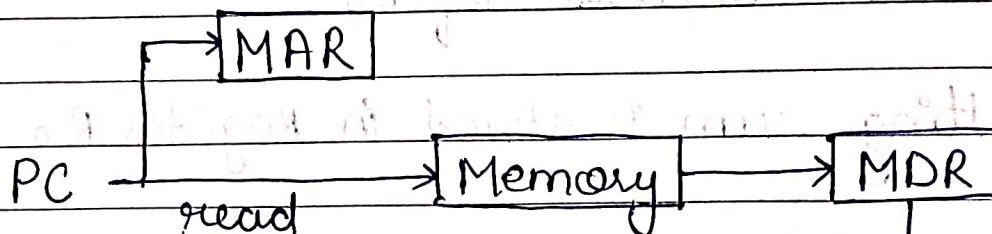


PC - Program counter, used to store address of next instr. that is to be processed.

IR - Inst' register

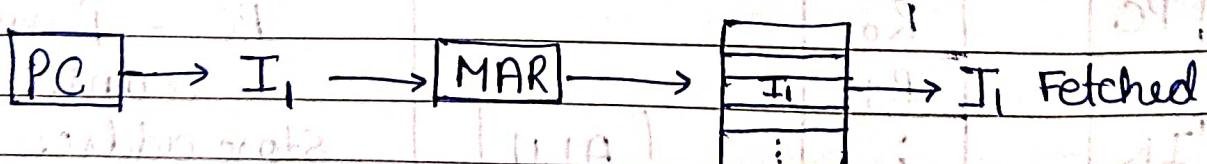
Example

I: ADD LOCA, R0.



- First PC will provide the address of this inst.

This will/address will pass through MAR and from MAR the fetching operation is performed.



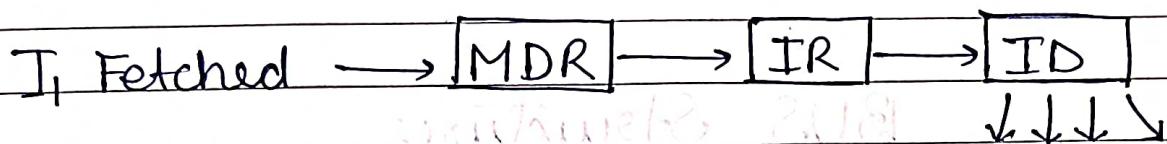
This fetching will take few clock cycle.

Once the fetch of inst<sup>n</sup> of I<sub>1</sub> is completed that inst<sup>n</sup> is transferred to MDR.

Now after the inst<sup>n</sup> is transferred to MDR and since this is type of instr. it will be stored in register IR.

Inst<sup>r</sup>: is transferred to IR for decoding purpose.

As this Inst<sup>r</sup>: I<sub>i</sub> that is stored in form of data in MDR is actually a type of inst<sup>r</sup>: that is why transferred to IR.



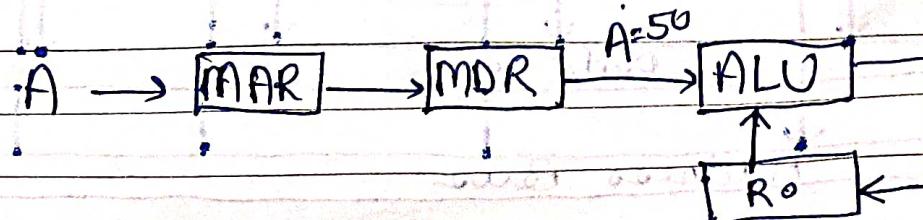
After decoding it (inst<sup>r</sup>) is transferred to accordingly various control signals are issued in a particular sequence.

Now one operand A is in location of memory so, same step will repeat.

Address of A will be transferred to MAR. Now A is fetched from MAR and data of A be 50 that is read from MAR will now be transferred to MDR.

From MDR this data will be transferred to ALU.

After every fetching operation PC is updated.  
p.e.  $PC = PC + 1$

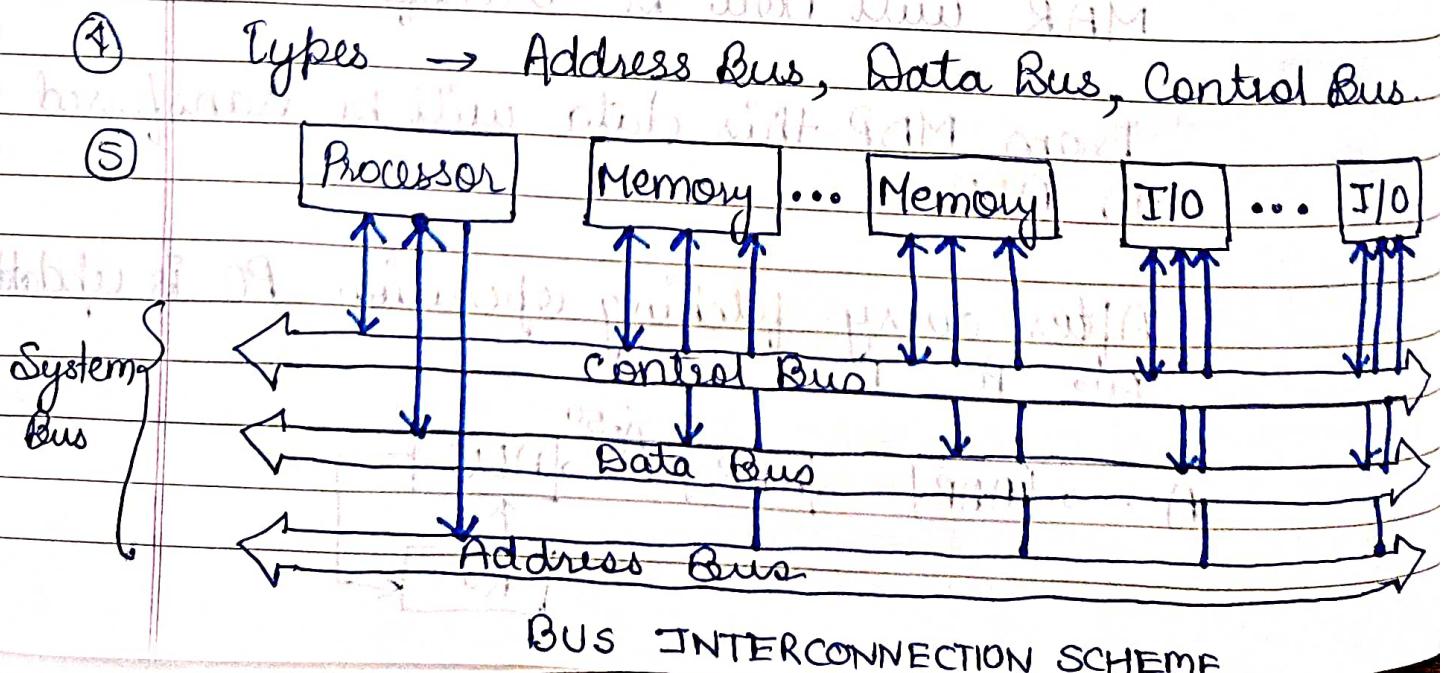


## Instruction Cycle

1. Inst. fetch (IF)
2. Inst. Decode (ID)
3. Operand fetch (optional) (OF)
4. Inst. Execution. (IE / EX)

## BUS Structures

- ① Various components of a digital system like CPU, I/O devices, Memory are connected with a common group of lines/wires known as BUS.
- ② There are many ways to connect different parts inside a computer together.
- ③ A group of lines that serves as a connecting path for several devices is called a Bus.



- ⑥ Task of Control Bus - used to transfer control signals. Since most of the signals are issued by the Control Unit in the processor so they are transferred through a control Bus.
- ⑦ Task of Data Bus - used to transfer the data or instruction or operand or any result or processed data.
- ⑧ Task of Address Bus - used to transfer the address from one unit to another unit. These addresses are always generated by the Processor. Bcoz control unit is basically responsible to generate all the addresses.
- ⑨ Direction & Interconnection of the three system bus. Control and Data bus are Bi-directional whereas Address Bus is Uni-directional i.e. address bus is outward through Processor and inward to other components like memory unit & Input/Output Unit.
- ⑩ Why address bus is outward through Processor? Because all the address are generated by Processor only. And then the address is checked in each component or matching component.
- ⑪ Let's take an example of READ operation.

Suppose READ operation has to be performed.

Step 1 : Address is generated by memory where the read has to be performed.

Suppose an item at any memory location has to be read. Then processor will generate the address first of that memory location.

Step 2 : Now the address generated by the processor is available on ADDRESS BUS.

Step 3 : Now whose ever component's address matches that component will be activated & in parallel to that central bus will send the signal READ. So whose ever address will match with them operation will perform.

Step 4 : The data will be read by memory and will be transferred with help of data Bus to processor.

## Single Bus Structure

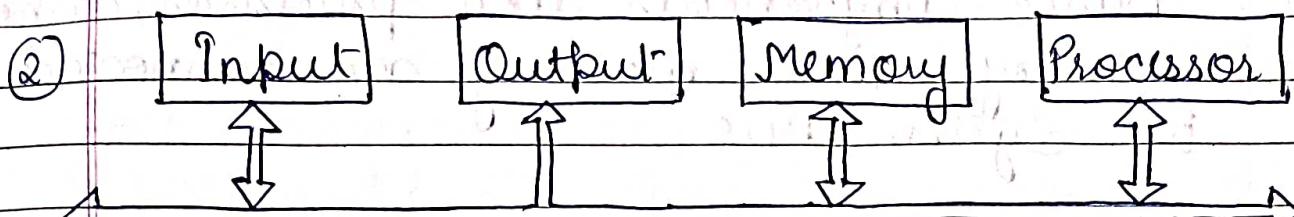
①

Drawback :

① Propagation Delay

② Limited Capacity - can't connect more no. of components.

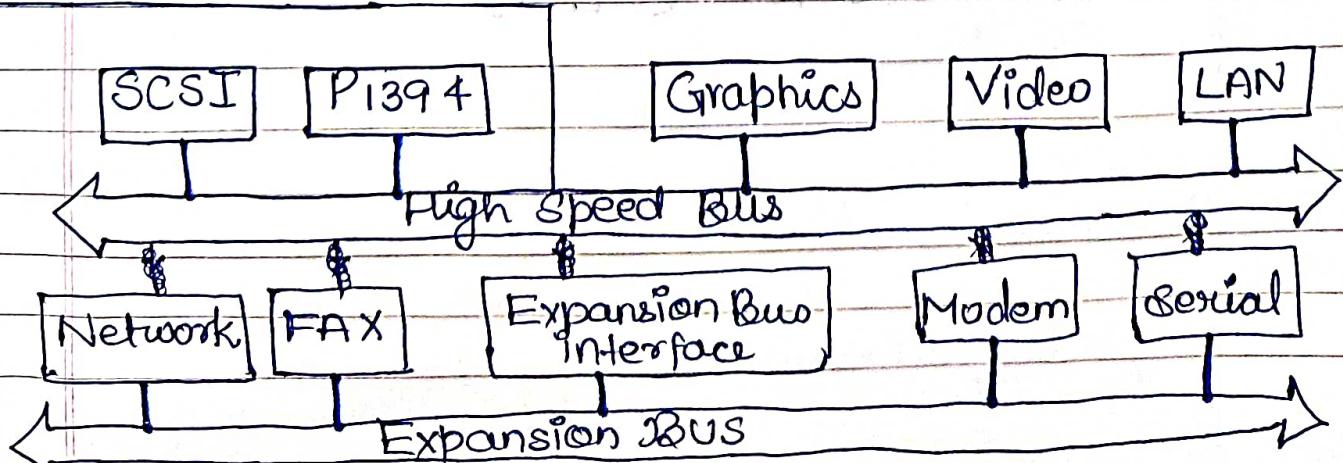
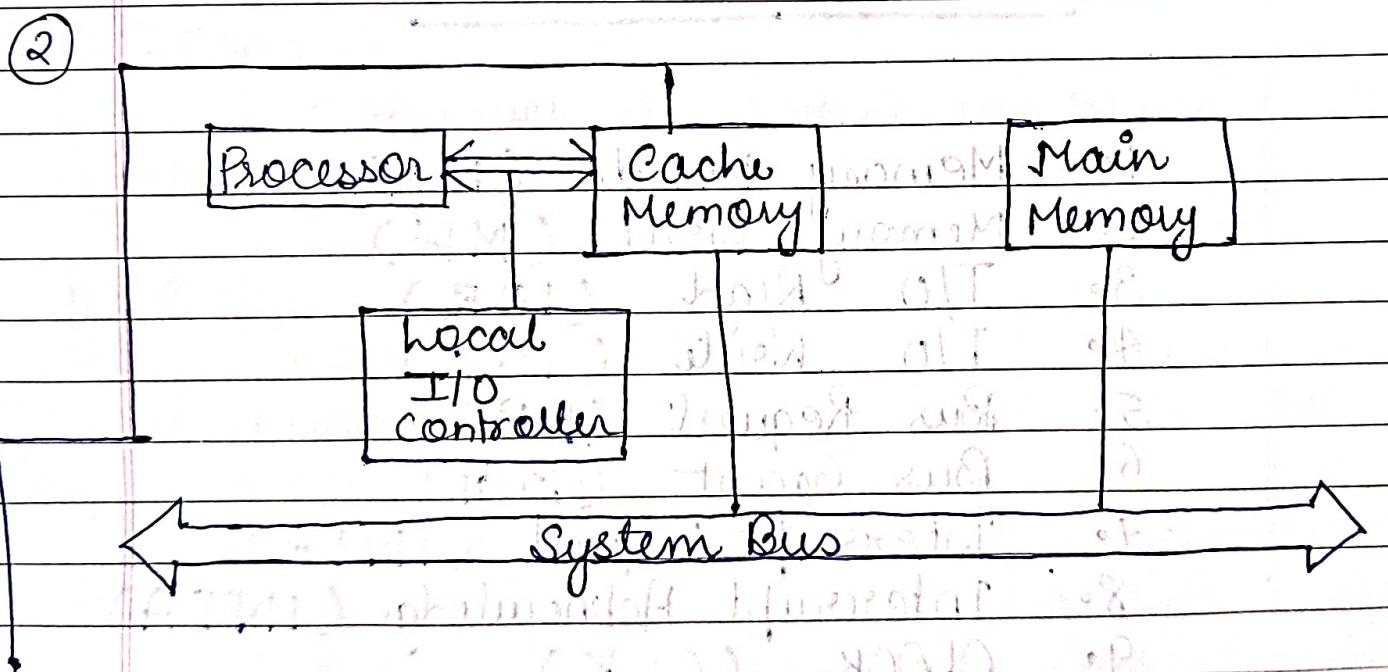
③ Bottleneck Problem



Single bus architecture  
Set of all three bus

### Multiple Bus Structure

- ① To overcome the drawback of single bus structure.



③ Basic components like processor, Cache memory, main memory are connected to System Bus.

④ High capacity components are connected to High Speed Bus.

⑤ Advantages are :

- Fast accessing (High rate of data transfer)
- High performance
- Scalable

### Control Commands

1. Memory Read (MR)
2. Memory Write (MW)
3. I/O Read (IOR)
4. I/O Write (IOW)
5. Bus Request (BR)
6. Bus Grant (BG)
7. Interrupt Request (INTR)
8. Interrupt Acknowledge (INTA)
9. CLOCK (CLK)

1. MR

ATM Appendix

→ command is basically used to instruct the memory that read operation is to be performed & this signal is issued by the Control Unit.

2. MW

→ whenever processed data has to be stored in memory the MW command is issued.

3. IOR

→ whenever input is taken from the IO device then IOR command is used.

4. IOW

→ whenever the output has to be written then IOW command is used.

5. BR

→ whenever a particular device/unit/component is requesting for BUS then BR command is issued.

Device requesting for Bus to transfer the data and if the Bus is available control unit will generate the signal Bus Grant i.e. BG.

6. INTR

→ whenever any unit is requesting for an interrupt that request is handled by INTR. and that request is acknowledged by the

processor through INTA.

Q7. CLK  $\rightarrow$  In our System / Digital System, all the operations or components are synchronized through a clock (CLK).

## BUS Arbitration

① In a multiple bus architecture, several components are connected through a bus. So they are requesting to transfer the data. Thus, they require bus.

Because only on the availability of bus the transfer operation will take place.

But the BG signal is issued by the control unit only to a particular component among all the components who are requesting for bus i.e. BR.

So, switching of bus from one component to another component is called as Arbitration.

③ Bus Arbitration refers to the process by which the current bus master accesses & then leaves the control of the bus & passes it to the another bus requesting processor unit.

(A)

Two types:

i) Centralized bus arbitration ↳ A single

bus arbiter performs the required arbitration  
that means one component is deciding to whom the BUS has to GRANT.

ii) Distributed bus arbitration ↳ All devices

participate in the selection of the next bus master.

The component that is using the bus is called as master.

Various methodologies to handle bus arbitration

### Daisy Chaining

#### Advantages

- Cheaper & Simple
- Least number of lines

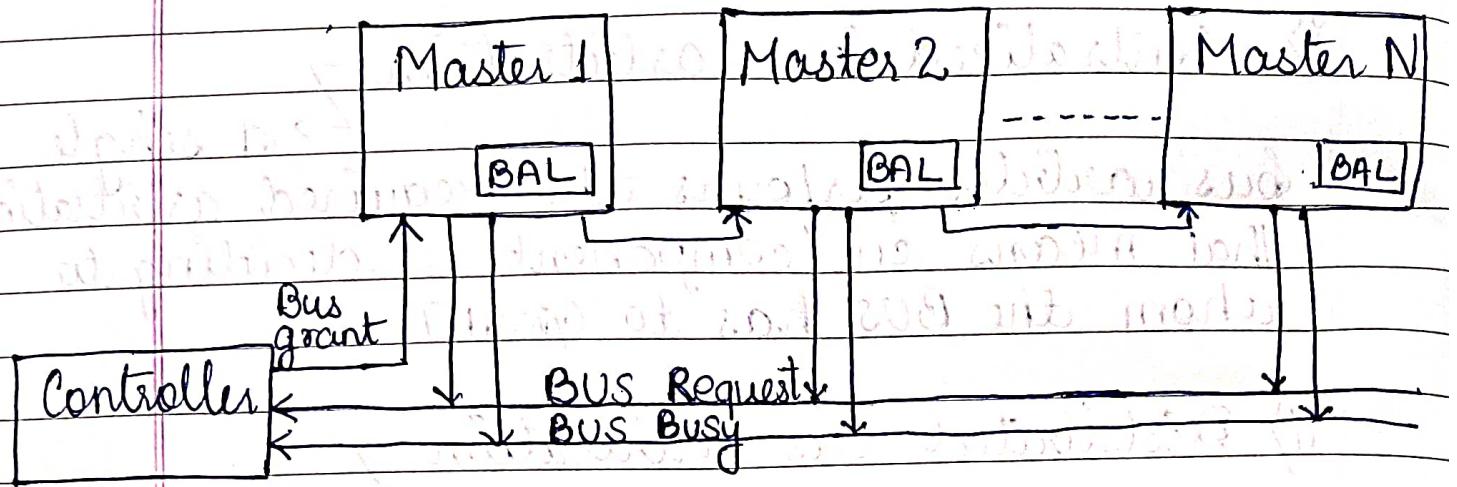
#### Disadvantages

- Slow arbitration time
- Priority depends upon physical location
- Chances of failure is more.

BB: Bus Busy

BAL: Bus access logic

BR: Bus Request



Several masters are connected through a common line i.e. BR & BB.

Master can request for Bus through BR line. If the BG has been placed by the controller for any master then rest of the masters will be informed that Bus is busy through BB line.

That is why simple & cheaper & least no. of lines are required.

But in this method, the signal for BG is propagated or broadcasted through a first master. So, if the Bus grant is issued to the last or mid master then the BG signal will take time to reach there as it is propagated from 1<sup>st</sup> master to 2<sup>nd</sup> then ... to last. Thus, slow arbitration time will be observed.

Secondly, if there are more number of masters are connected. So master at the last location will get the BUS in the last. Thus, priority depends upon physical location.

And if any particular masters fail, so, then the propagation of BG signals is not possible. So in that case remaining masters can't receive the BG signal. Thus chances of failure is more.

So, failing of one component will reflect the whole system to fail.

### Polling or Rotating Priority Method

#### Advantages

- Fast Arbitration
- Simple
- less chance of failure

#### Disadvantages

- Increasing the size will require more number of address lines.