

How to do?
How it does it?

COMPUTER ORGANIZATION AND ARCHITECTURE

Unit 1: Introduction

↳ what to do?
what the computer does?

Functional units of digital system and their interconnections, buses, bus architecture, types of buses and bus arbitration.

Register, bus and memory transfer.

Processor organization, general registers organization, stack organization and addressing model.

Computer Organization refers to the operational units and their interconnections that realize the architectural specification

Example: that are transparent to the programmer.

- Control signal
- The memory technology used

(To Hardware Karta Hai,
High level)

To programme

Karla Mai

Date / /

Page No.

7 Low level logical things

Shrivastav

Computer Architecture refers those attributes of a system that have a direct impact on the logical execution of a program examples.

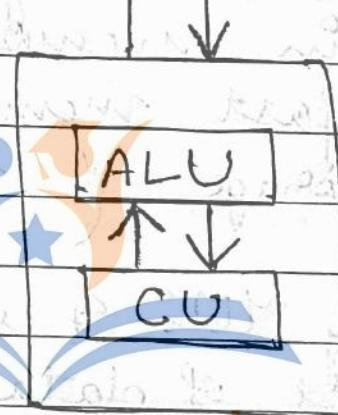
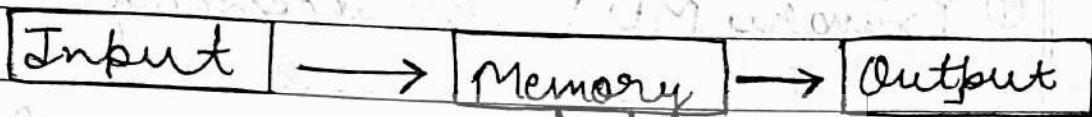
- The instruction set
- The no. of bits used to represent various data types.

Difference b/w CO and CA

S-No.	Key	CO	CA
1.	Purpose	CO explains how a computer works	CA explain what a computer should do.
2.	Target	CO provides structural relationship b/w parts of computer system.	CA provides functional behaviour of computer system
3.	Order	CO is started after finalizing the	

Functional Units: are a part of a CPU that performs the operations and calculations called for by the computer program.

Functional units and their interconnection
The collection of functional unit connecting the various module is called interconnection.



Input: Computer accepts coded information through input units which read the data. The most common input device is keyboards.

Output: Output units sends process result to outside world. The familiar example of such device is monitor, printer etc.

Memory Unit: MU is used to store programs and data.

Two types of memory device are used to form a MU.

- ① Primary MU
- ② Secondary MU

Primary MU: Primary MU commonly called a main memory and it is very fast memory. It's used for the storage of programs and data.

Secondary MU: It stores large amount of data. Ex: Hard disk

Collegesmate

ALU (Arithmetic Logic Unit)

→ Arithmetic or logical operation like addition, subtraction, multiplication and division, AND gate, OR gate are performed by ALU.

Control Unit

In simplified form: ① when to do
② what to do

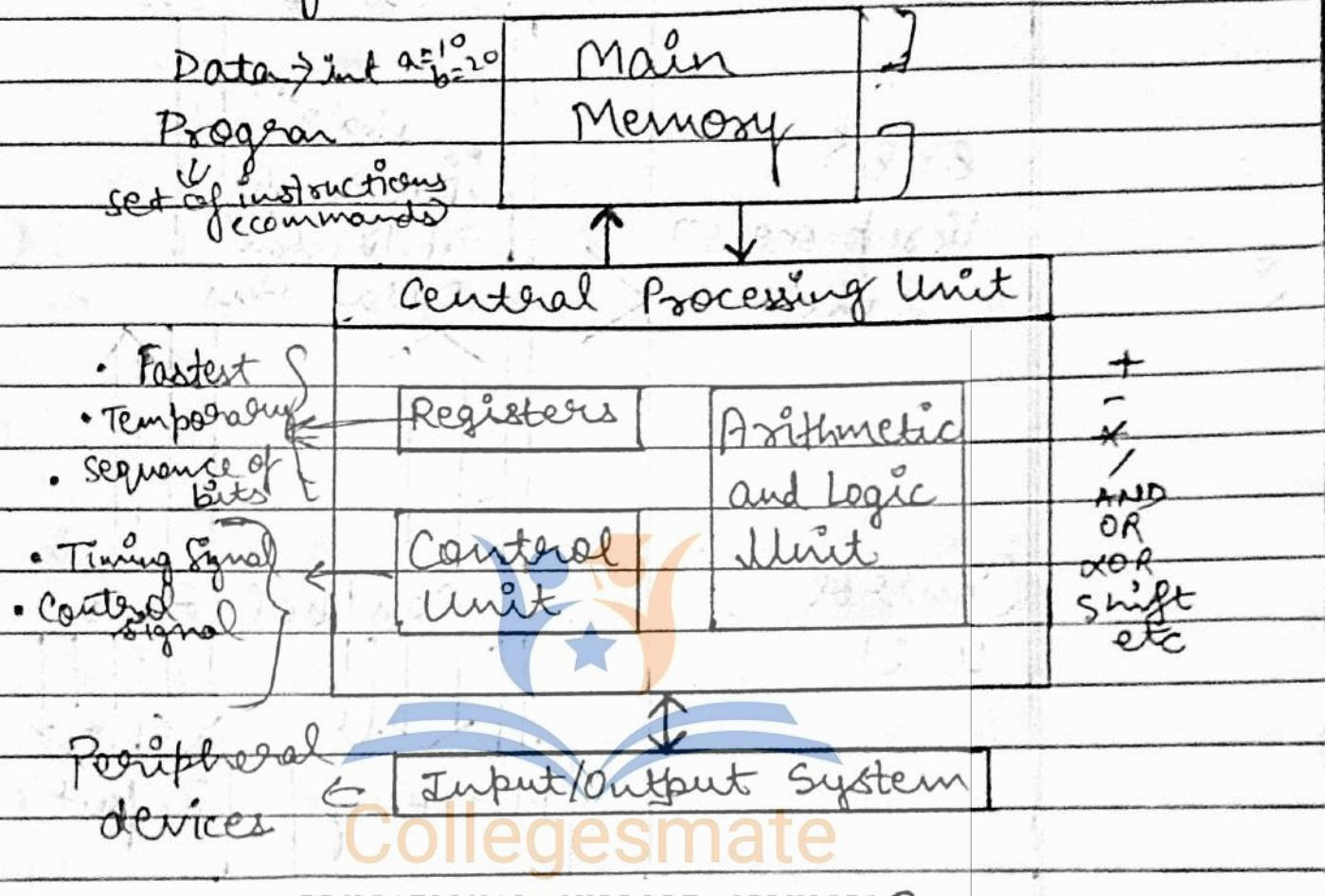
In other words: To transfer of the data b/w processor and memory and controlled by control unit? Control unit provides control signals.

Ex: Input read, output read, input/output write, Bus grant, Bus acknowledge.



Igurk
 Page No. _____
 Date _____
 (From _____)

von Neumann Architecture (Stored Memory Program)

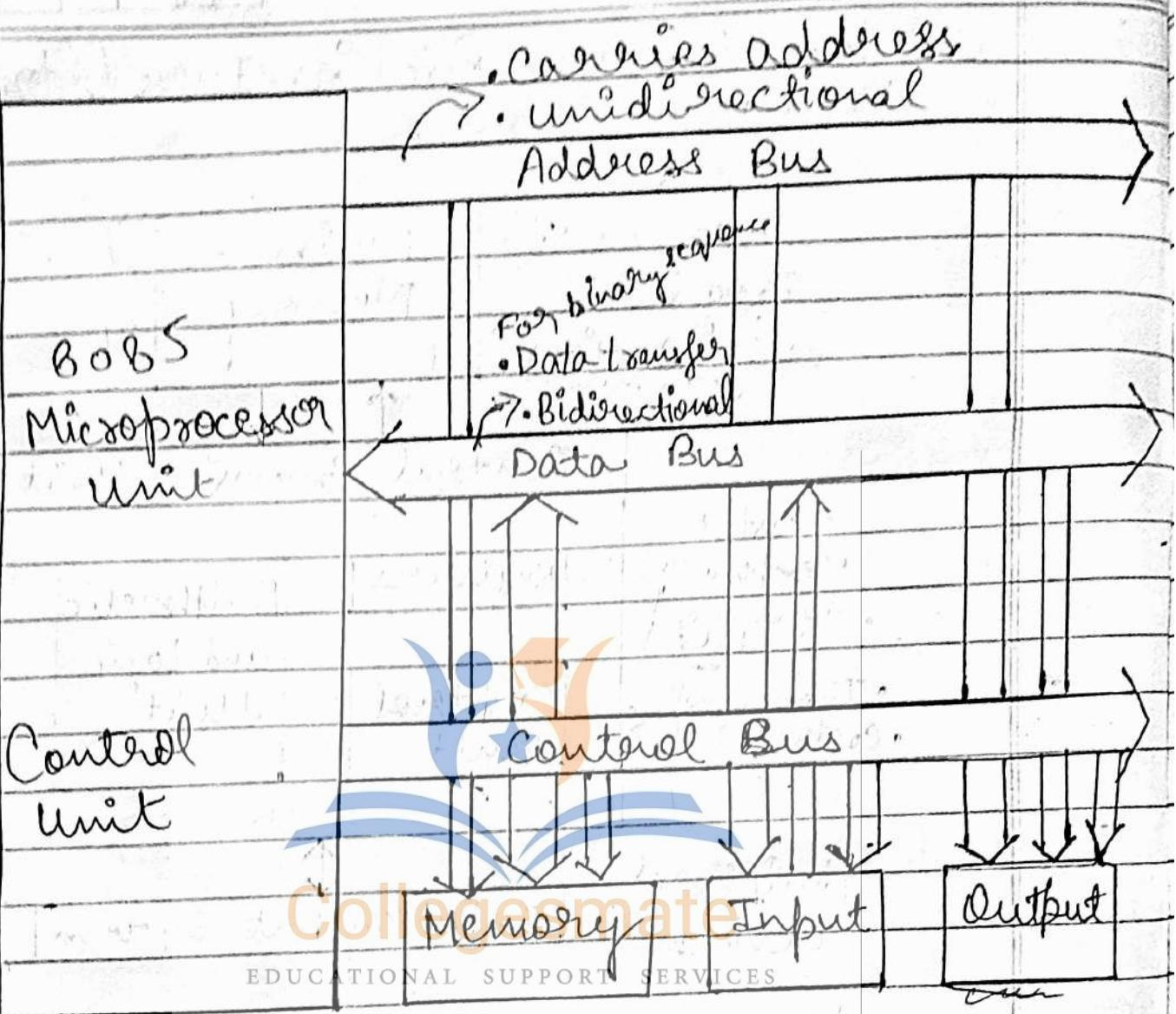


Now, these networks b/w memory, CPU, input/output devices is made using topologies (network topologies).

So, these are connected using Bus (using multiplexer)

for 8085 microprocessor unit, length of address bus is 16 bits (2^{16} slots)

Date: _____



Like if ~~size~~ length of address bus is 3 bits
means: 000

001	0
010	1
011	2
100	3
101	4
110	5
111	6

3 bits $\Rightarrow 2^3 = 8$ slots

word: memory

representable unit
(Size is not fixed it may change)

Byte: Byte size (8 bits)

Buses and Bus Architecture

"A group of wires is called Bus". It is used to provide a necessary signals for to communicate b/w modules.

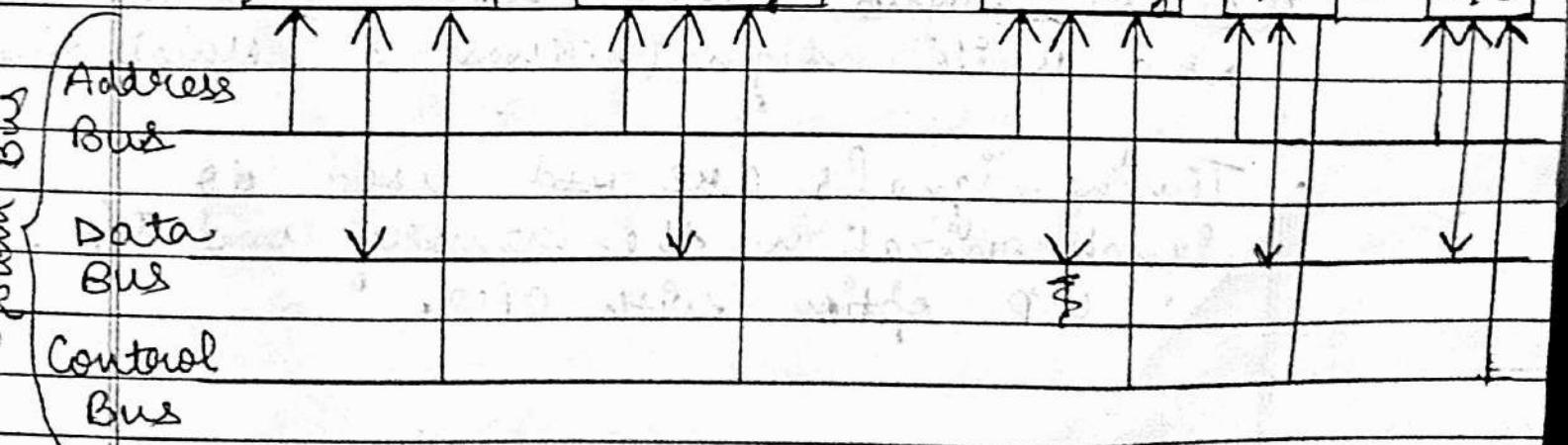
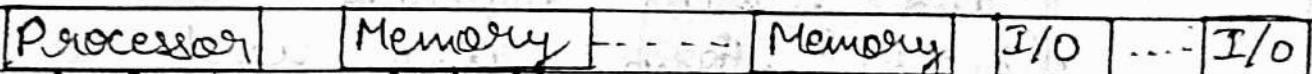
A Bus is a Shared Transmission medium from one place to another place.

Need of Bus

1. A purpose of Bus is to reduce a number of paths ways needed for communication between the components.
2. High speed transfer b/w CPU and Memory.
3. Must only be used by one device at a time.

Collegesmate

Bus Interconnection Schemes



System Bus: It is a communication channel provided for the communication between major components of system.

Example: Memory, CPU, I/P and O/P

Address (Bus or lines): It is a unidirectional Bus and these lines are used to carry the address towards Memory and I/P, O/P.

Data (Bus or lines): These lines or Bus are used to carry the Binary sequence (data) between CPU, Memory (I/P - O/P). It is a bilateral Bus. It can read and write. These are 32 or 64 parallel lines of data bus.

Collegesmate

Control (Bus or lines): These lines or Bus are used to carry the control signals and timing signals.

- Control signals are used to read and write signal. These are unidirectional.

- Timing signals are used to synchronize the memory and I/P, O/P option with CPU.

Options of Control Signals: G/P read, S/P read, I/O write, Bus grant, Bus acknowledge.

System Bus Categories on Basis of Timing

- ① Synchronous Bus
 - ② Asynchronous Bus

Synchronous Bus : In the Synchronous Bus the address line are sent to peripheral device then it activate the clock signal which are connected in different peripheral devices. When the address line are match with the peripheral device firstly the clock pulse is in working when clock

The device is an activate condition
 $CP = 0$, then device is deactivate condition.

Asynchronous Bus: In the asynchronous bus, the address line sent to the peripheral devices than for activating the peripheral devices or we don't need any timing signal.

These are activated on the basis of address line.

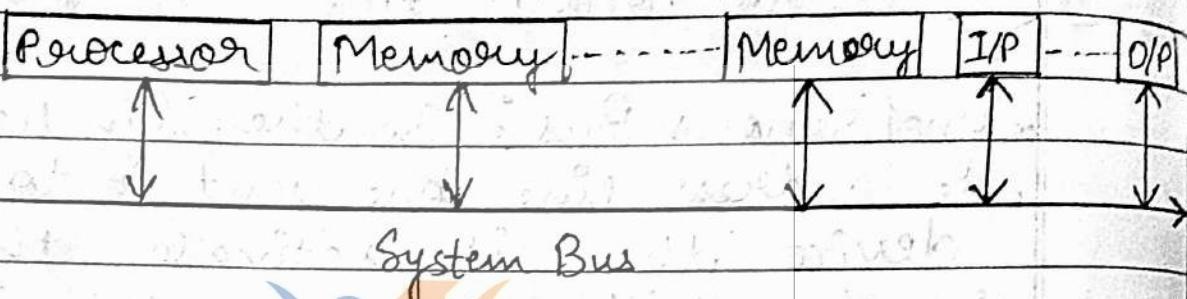
Bus Architecture.

There are 2 types of Bus architecture

(a) Single Bus Structure

(b) Multiple Bus hierarchy structure

Single Bus structure



In this two units communicate at a time, all units are connected to common bus called System Bus. Here address Bus, data Bus, control Bus are shown by single bus called System Bus. Hence such interconnection Bus structure is called single Bus structure.

Advantage

① It's low cost

② It's flexible for attaching peripheral devices.

Multiple Bus Hierarchy Structure

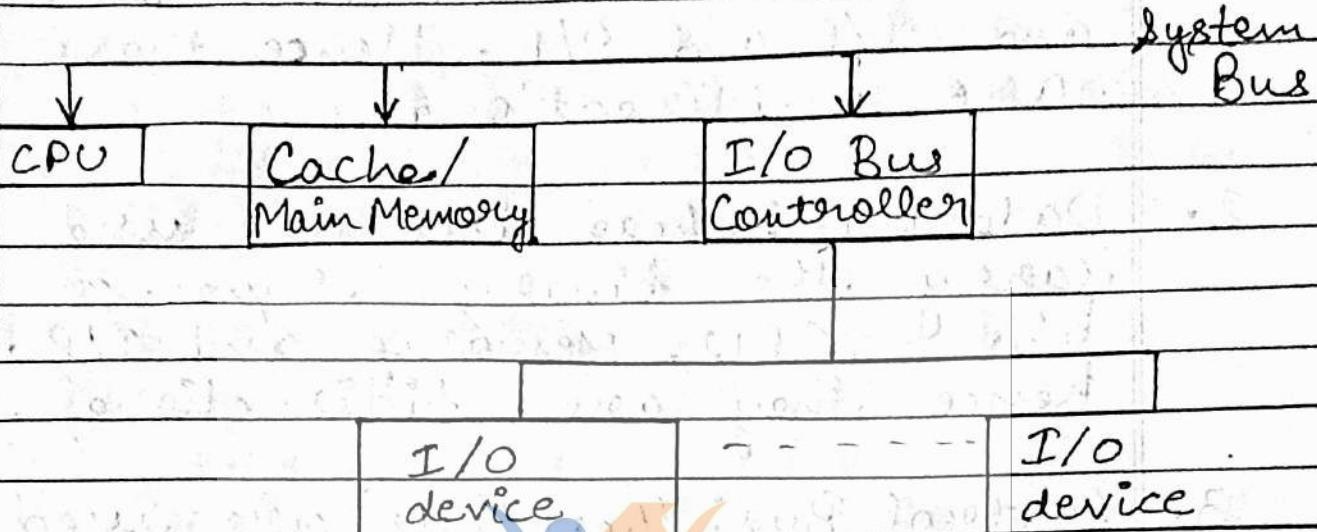


Fig. Multiple Bus Structure

Multiple Bus are connected to devices that are called Multiple Bus hierarchy. System Bus connected to CPU, Main Memory and another I/O Bus connected to I/O device.

Types of Buses

Buses basically are of 4 types :

1. System Bus: It is a communication channel provides the communication between major components of system.
Eg: Memory, CPU, I/P , O/P

System Bus consist of three categories of it :

- (1) Control Bus
- (2) Data Bus
- (3) Address Bus

1. Address Bus: These Bus are used to carry the address towards Memory and I/P and O/P. Hence these lines are unidirectional.
2. Data Bus: These Bus are used to carry the binary sequence data b/w CPU, Memory and I/P, O/P hence they are bidirectional.
3. Control Bus: These Bus are used to carry the control signals and timing signal.

Bus Type

Bus line can be separated into two generic types "dedicated" and "multiplexed".

Dedicated bus line is permanently assigned either to a function or to a physical subset of computer components.

In multiplexed bus type address and data information may be transmitted over the same set of line using an Address Valid control lines. At the beginning of a date transfer, the

address is placed on the bus and Address valid line is activated. After that the address is then removed from the bus, the same bus connection are used for the subsequent read or write data transfer.

Bustype



Bus Arbitration

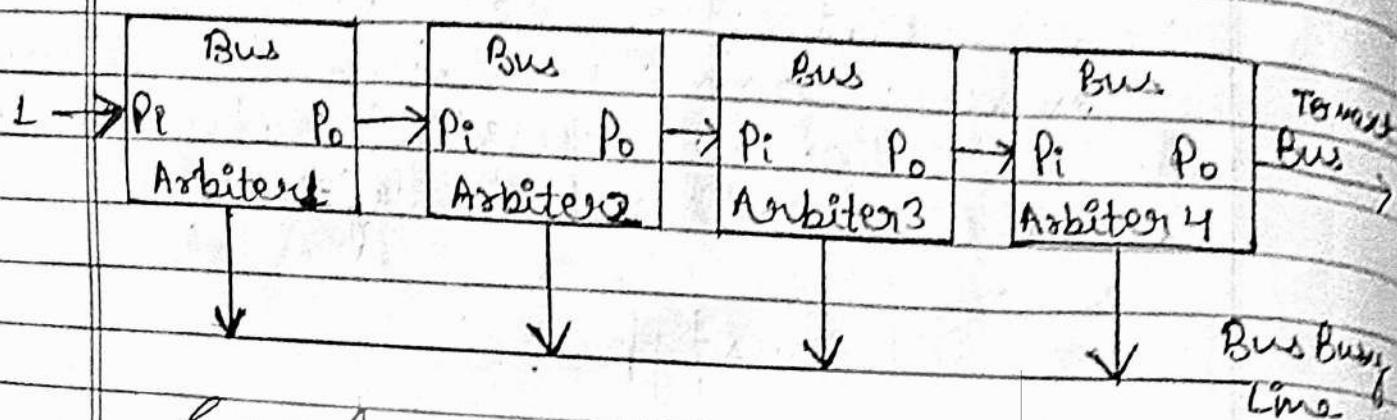
The arbitration process service all processor request on the basis of priority. The Hardware bus priority (HWP) resolves the technique which can establish by requesting control signal.

The Bus arbitration is process for acknowledgement of priority b/w system bus and processor or peripheral devices.

There are 2 types of Bus

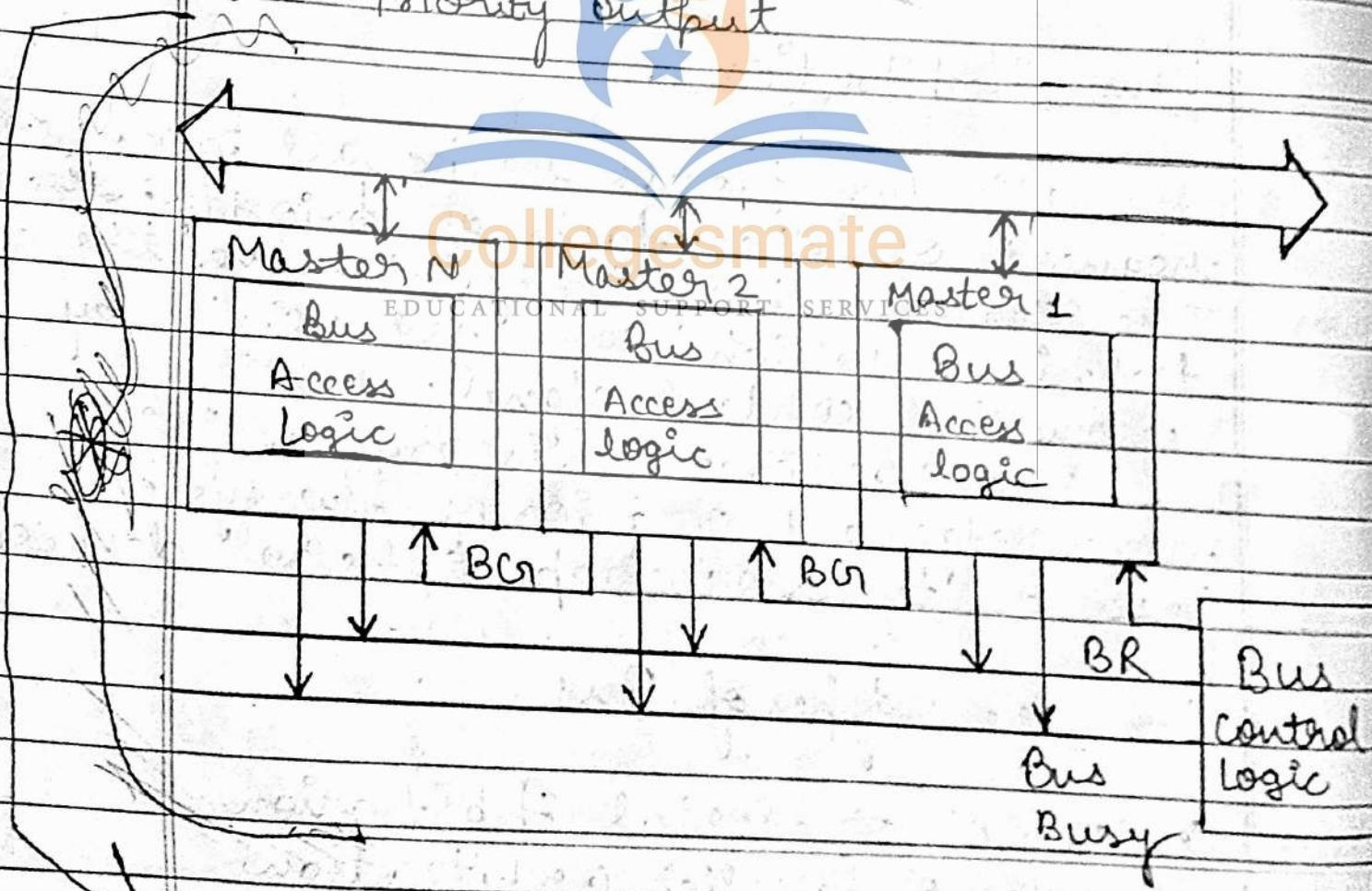
- (a) Centralised or serial Arbitration
- (b) Distributed or parallel arbitration

(a) Centralized or Serial Arbitration



Serial (Daisy chain) Arbitration

P_i : Priority input
P_o : Priority output



The above diagram shows the serial arbitration which is also called daisy chain

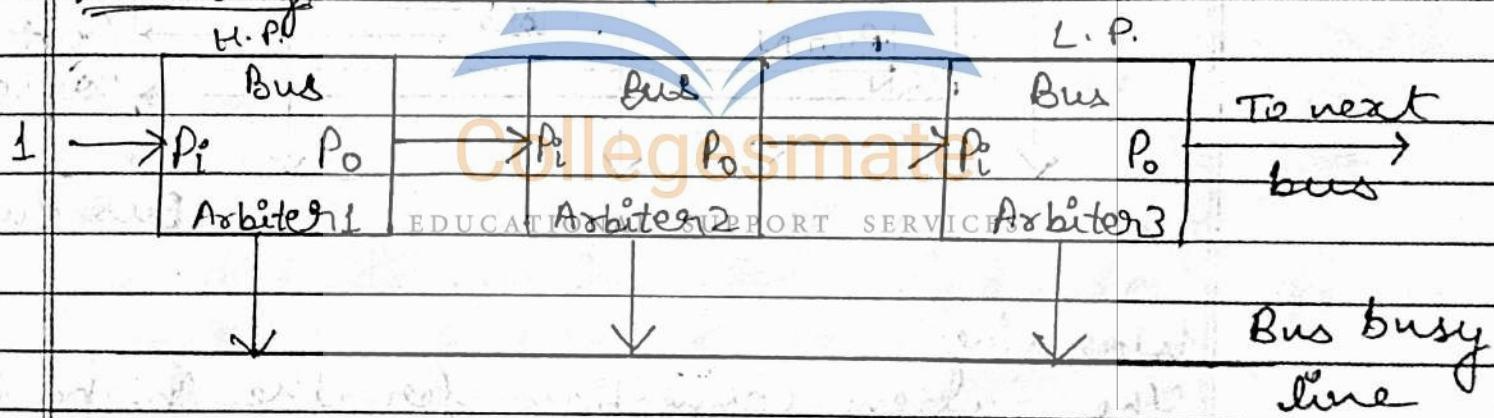
method. In this diagram, four processor P_1, P_2, P_3, P_4 in series which are also connected to system bus. The priority is fixed for every processor. In this diagram, P_1 processor has highest priority and P_4 has lowest priority.

There are 2 types of signals put on every processor.

(i) P_i (ii) P_o

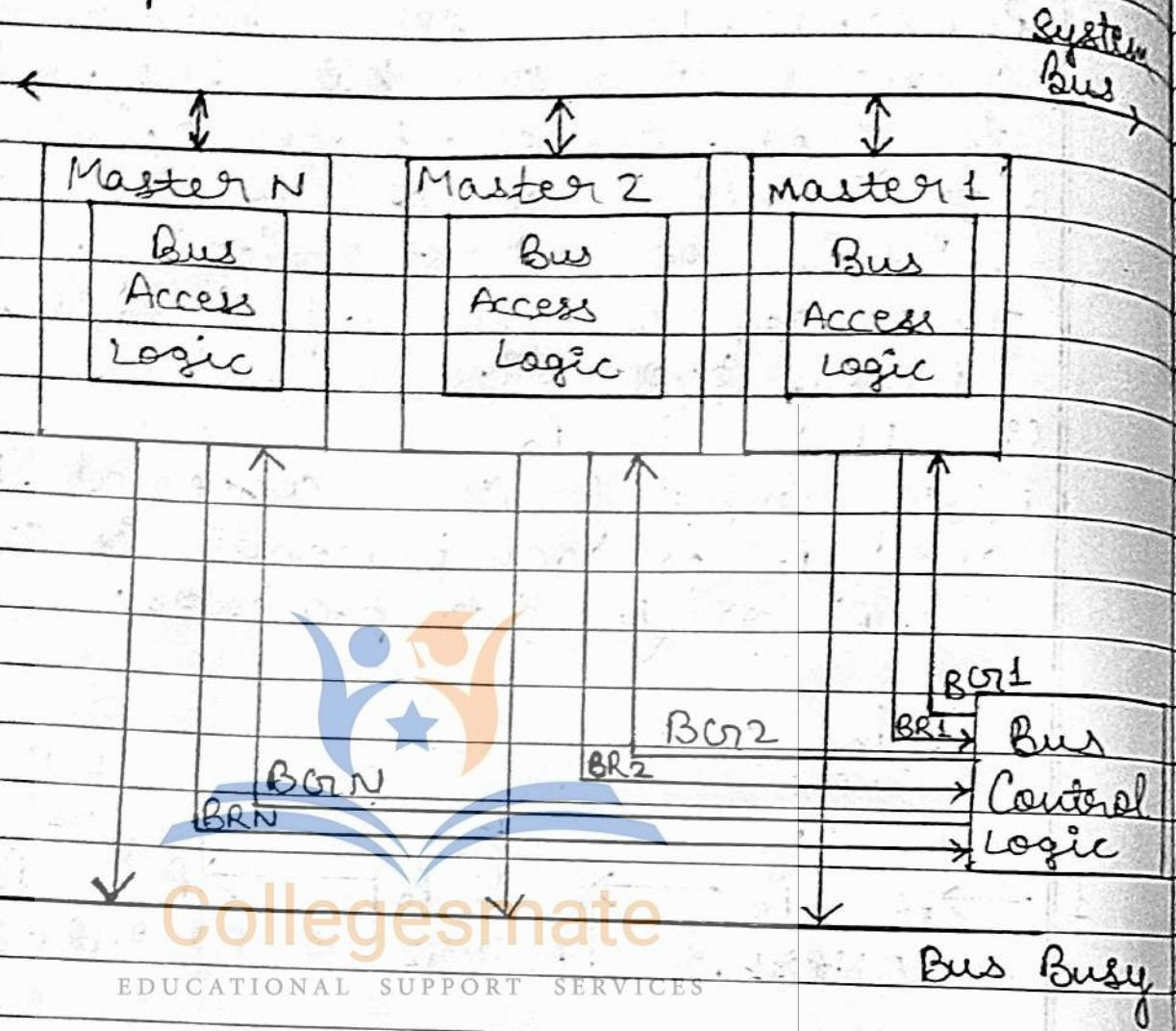
Priority input for P_i is connected to logic 1 and P_o for each processor is connected to the P_i for further processor.

Working
H.P.



When $P_i = 1$ at the first arbitrator, it means the arbitrator's request of the bus for transmission of the data, the acknowledge is provided to the arbitrator. For the bus uses the arbitrator's transfer the data using bus and during this time priority out ($P_o = 0$). So that further arbitration deactivate. continuing this time, other arbitrator request for the bus. The bus is not granted. It sends Bus busy signal.

Independent Request Method



Working

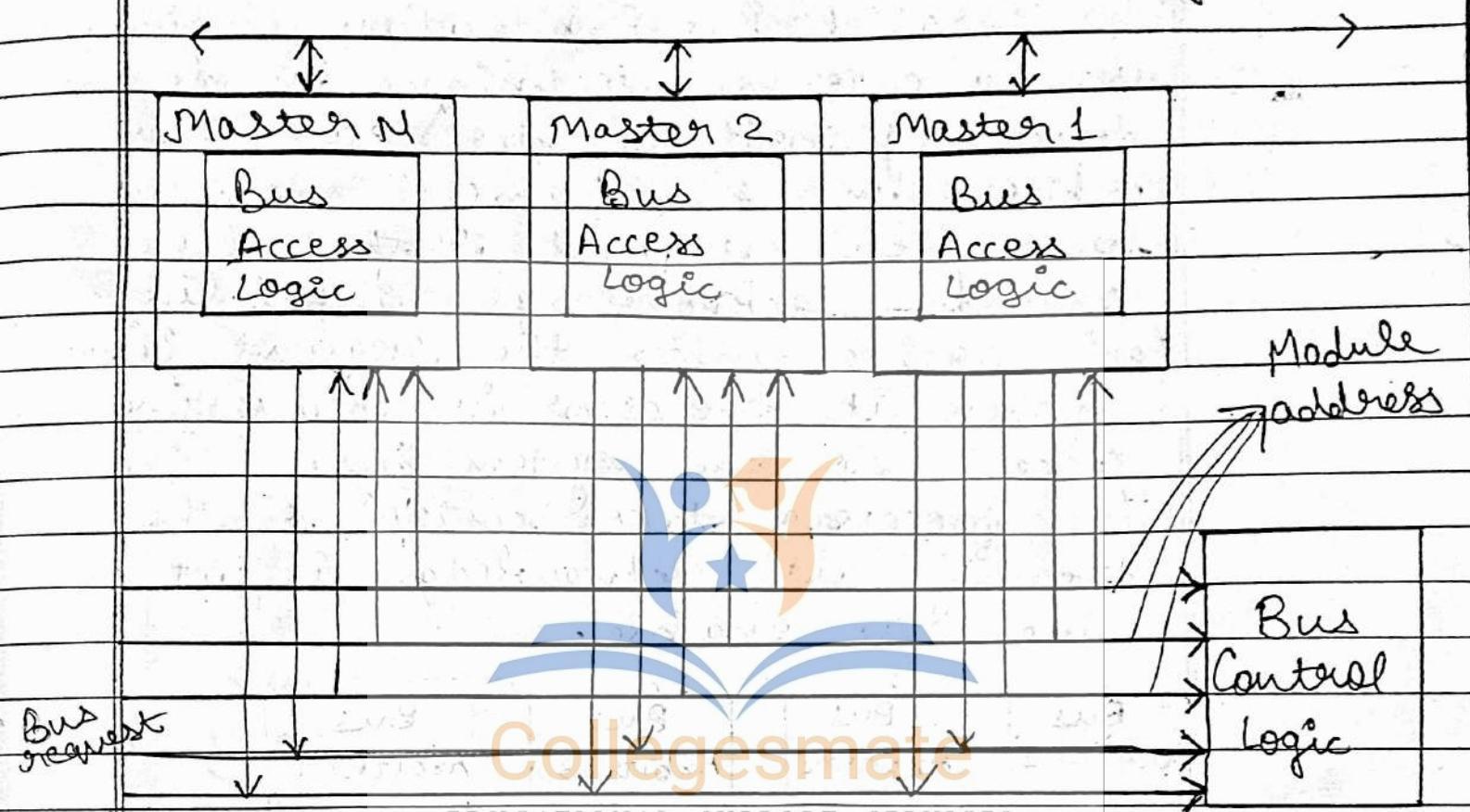
The System Connection for the Independent scheme. In this scheme each master / processor / Arbiter has a separate pair of bus request and bus grant lines and each pair has a priority to assign it.

Advantage

- If any failed then no any other is failed.

Pooling Method

System Bus



No. of address line required depends on the no. of master connected in system.

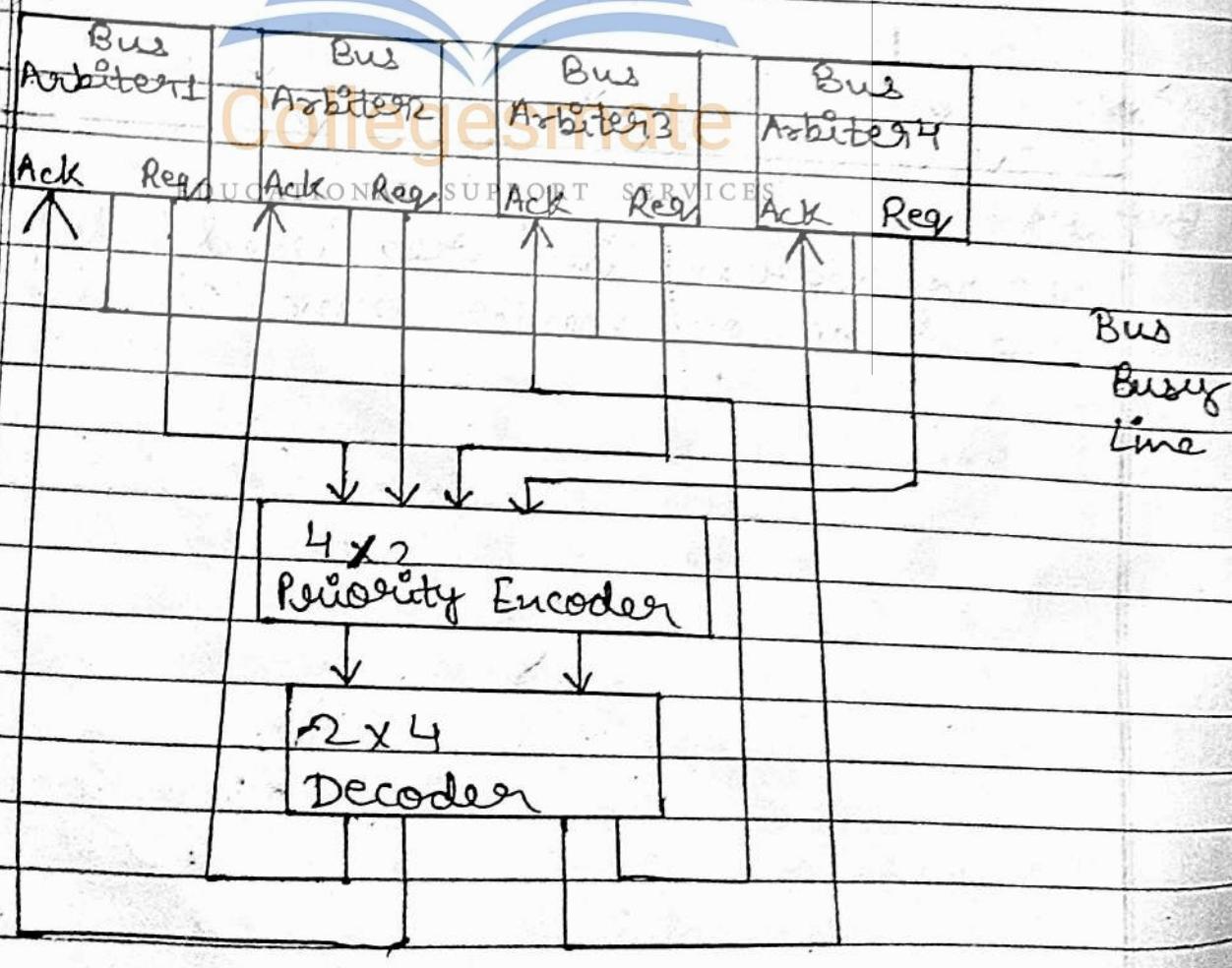
flip flop is a device which stores a single bit binary

Page No.	
Date:	

Distributed or Parallel Arbitration

The Parallel Bus Arbitration technique uses an external technique encoder and decoder (priority decoder) each bus arbiter in the parallel scheme has a bus request output line and bus acknowledge input line. Each arbiter enables the request line when it processor is requesting access to the system bus.

The processor take control of the bus if it acknowledge input line is enable.



Registers are sequence of bits i.e. sequence of flip flops

Page No.	
Date:	

Register, Bus and Memory Transfer

A typical digital computer has many Registers and Path must be provided to transfer information from one register to another.

A more efficient scheme for transferring information between registers in a multiple-register configuration is a common bus system.

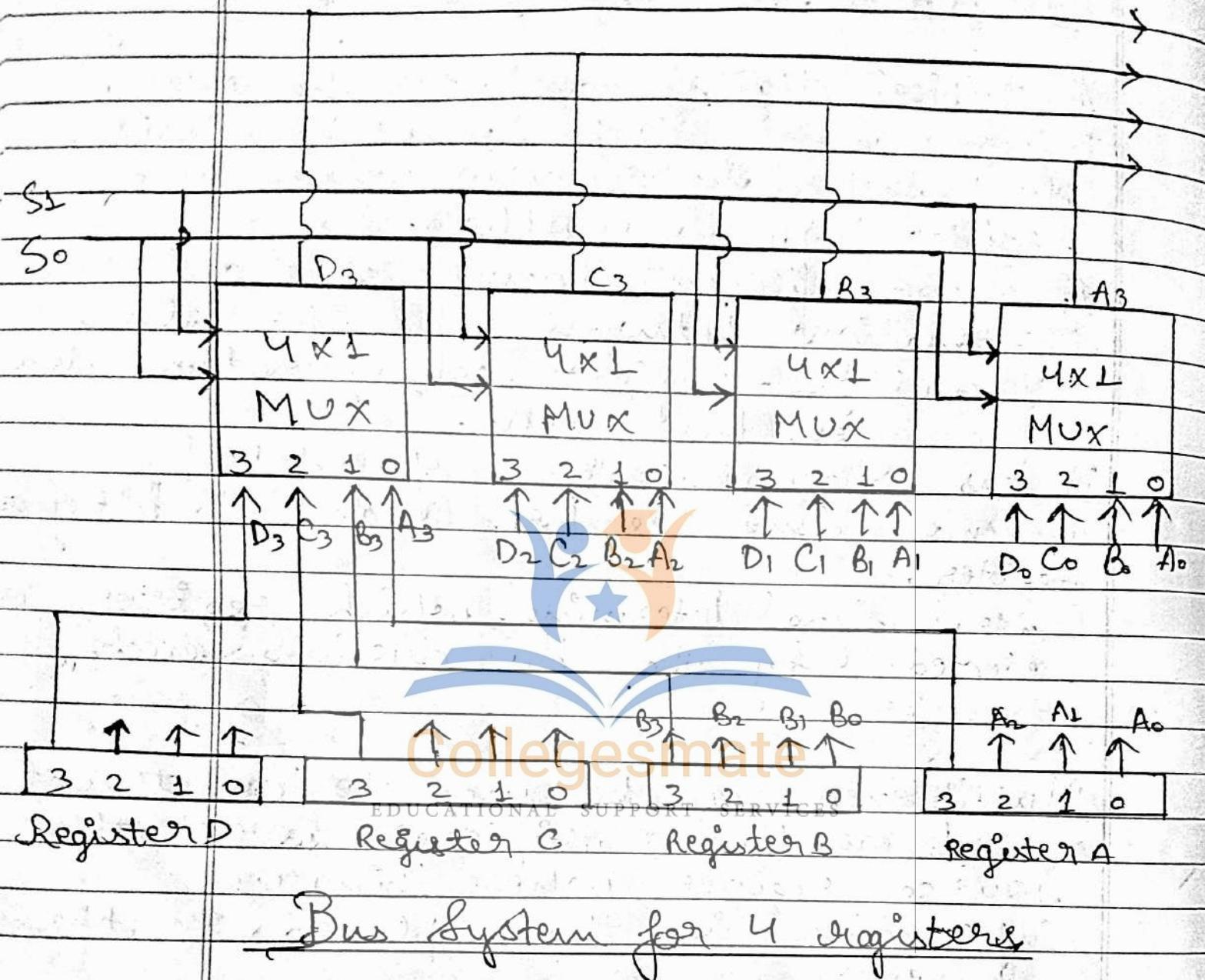
A bus structure consists of a set of common line one for each bit of a register.

Control signal determine which register is selected by the bus during each transfer.

Collegesmate

Multiplexers can be used to construct a common bus. Multiplexers select the source register whose binary information is then placed on the bus.

The select line are connected to the selection inputs of the multiplexers and choose the bit of one register.



Bus System for 4 Registers

Page No.	
Date:	

Memory Transfer

Consider a Memory unit that receives the address from register called the Address Register symbolized as AR. The data are transferred to another register called data register symbolized by DR. The read option can be started as follows:

Read : $DR \leftarrow M[AR]$

This cause a transfer of the content of a data register to a memory word M selected by the address.

Assume that the input data are in Register R, and the address is in AR. The write option is as follows:

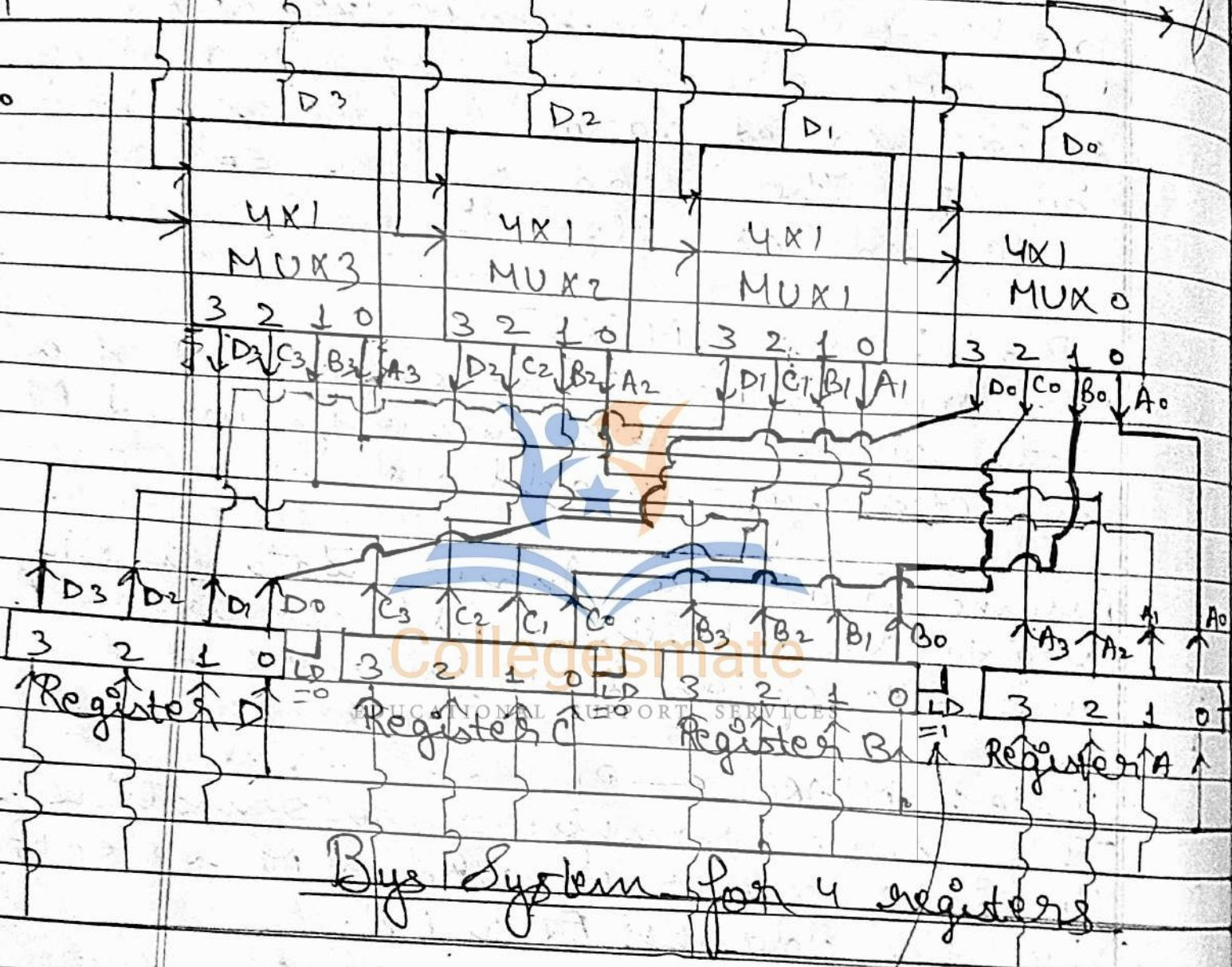
Write : $M[AR] \leftarrow R$

This cause transfer of information from R into memory word M selected by the Address in AR.

Common
bus

S1

S0



(Load) $LD = 1$ This will load the data on common bus that is register B & D here on register B.

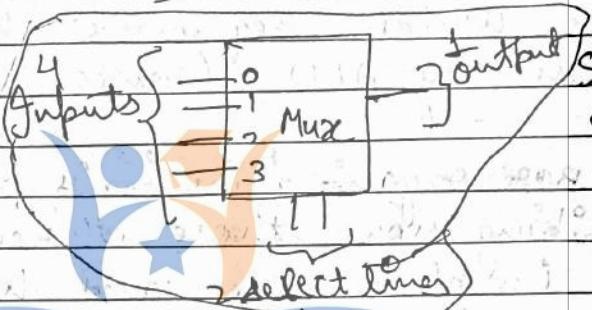
2^n : No. of select lines
 2^n : max no. of inputs
 Date: _____

General no. of registers used
in computer: 16 bit registers

Here in 4×1

4 means no. of inputs

4 inputs = 2^2 inputs
→ 2 select lines



$0, 0 \rightarrow 0 \rightarrow \text{Reg A}$
 $0, 1 \rightarrow 1 \rightarrow \text{Reg B}$
 $1, 0 \rightarrow 2 \rightarrow \text{Reg C}$
 $1, 1 \rightarrow 3 \rightarrow \text{Reg D}$

Select lines ki jo value hogi
multiplexer ko usi input keo
uthaega.

like jis eg diagram H, let's say
select lines ki value 3 Hai
To multiplexer D_0, D_1, D_2, D_3 Ko
uthaega aur bus per load
karega.

Ques ① No. of Mux required = No. of bits
in registers

for
GATE

② No. of Inputs in Mux
= No. of Registers

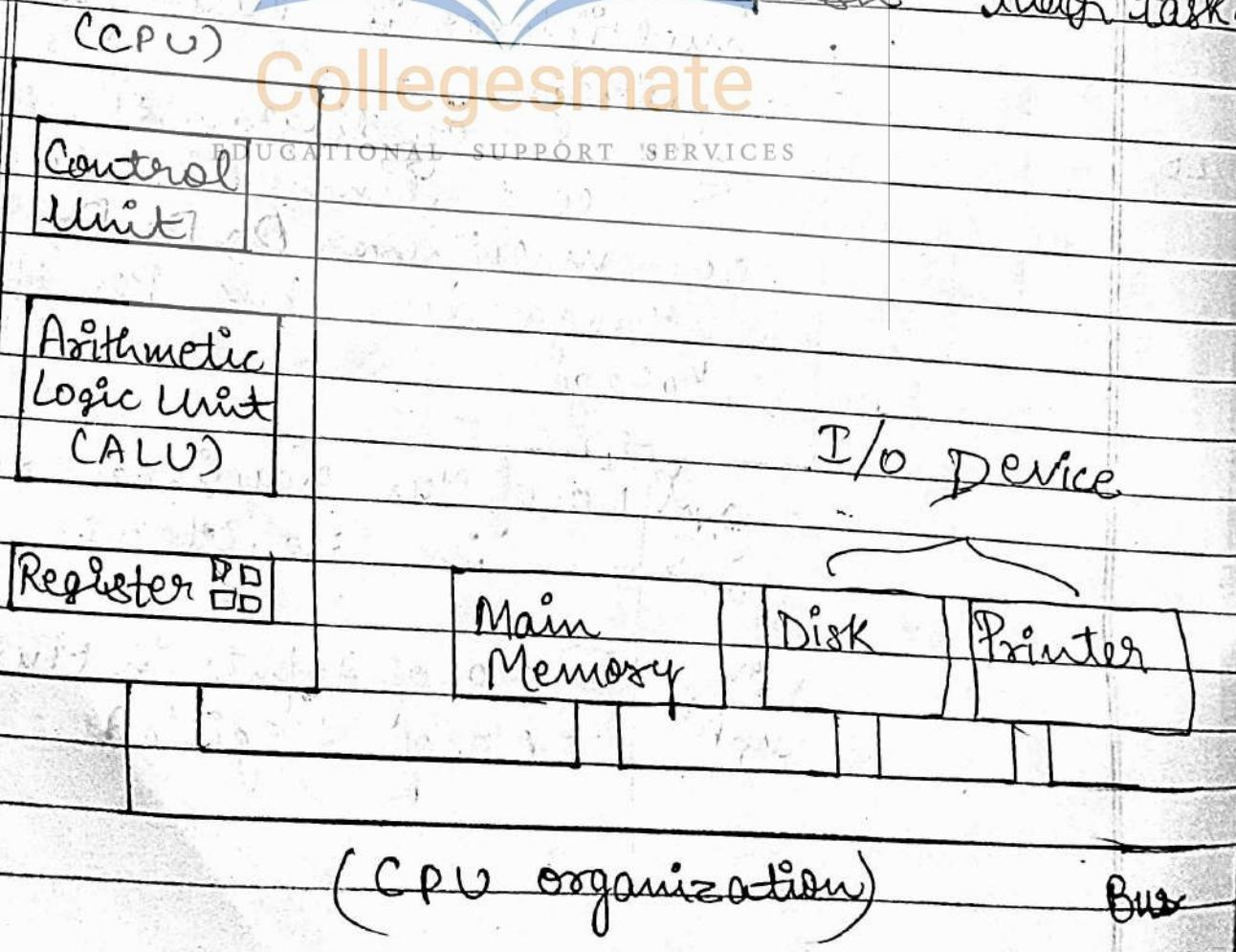
the data
is
register B...

Processor Organization or CPU organization

A processor must ~~to~~ have three functional unit to be what we call a computer. A unit that perform ALU operation on data (ALU) A unit that which remember the data (Memory).

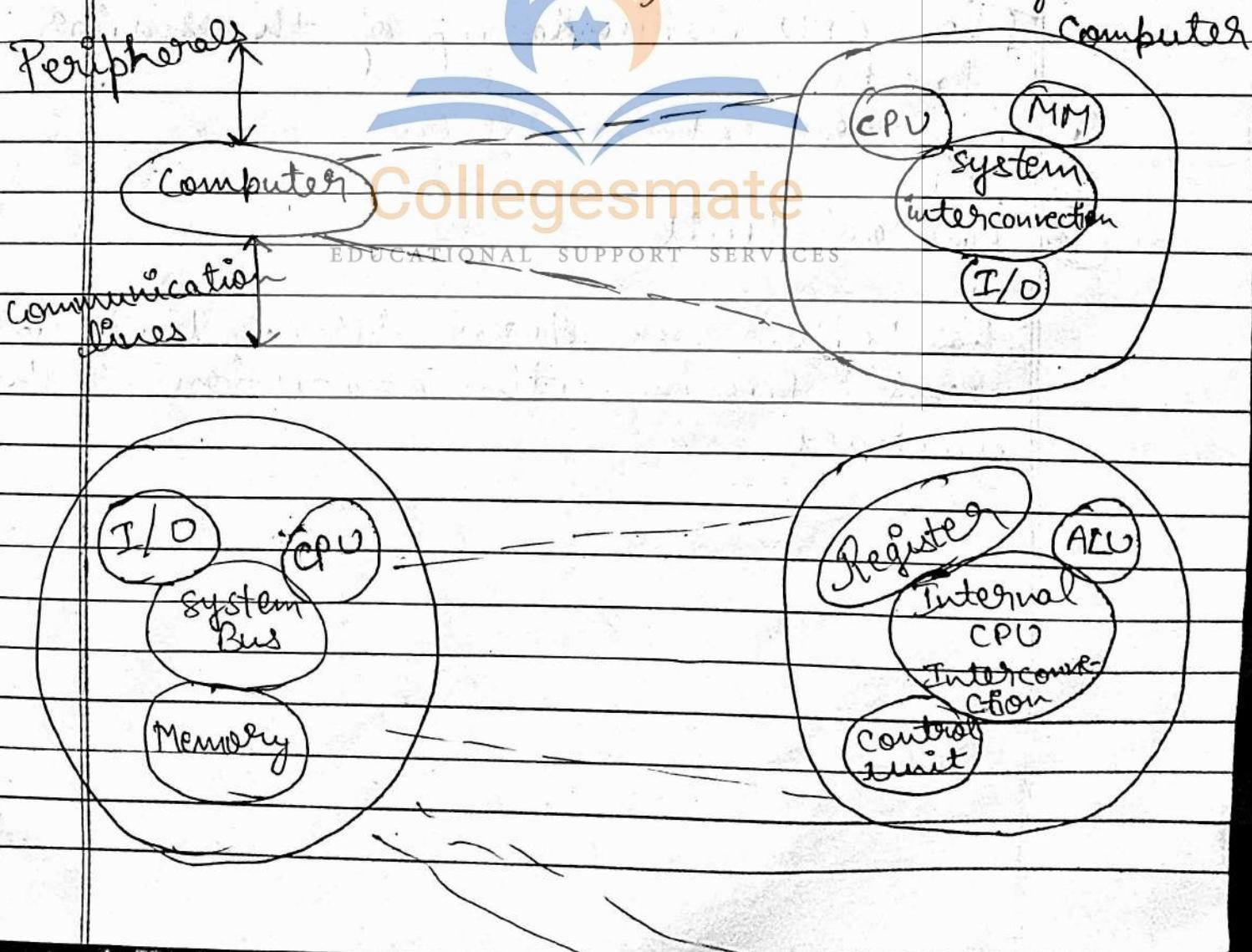
A unit which sequence the option the option by ALU (control unit).

"Processor organization" is a term describing how those three elements are implemented and how they interconnect to accomplish their task.



The major structural component of CPU are:

- ① Control Unit: Control the operation of the CPU and hence the computer.
- ② ALU: Performs computer's data Processing functions.
- ③ Registers: Provide storage internal to the CPU.
- ④ CPU interconnection communication among the control unit, ALU and registers.



Sequential logic

Register
Decoder of
CU

Control
Memory

Control Unit Structure

General Register Organization

The CPU is made up of three major parts

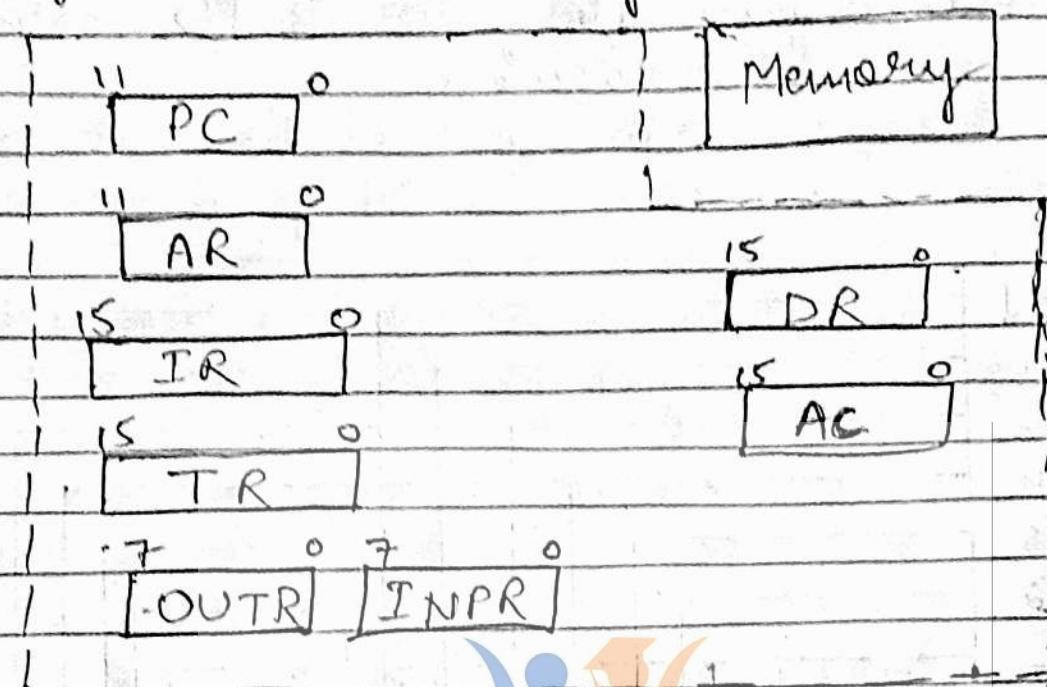
- ① Register set
- ② ALU
- ③ Control Unit

Collegesmate

EDUCATIONAL SUPPORT SERVICES

The Register set stores intermediate data used during the execution of the instructions.

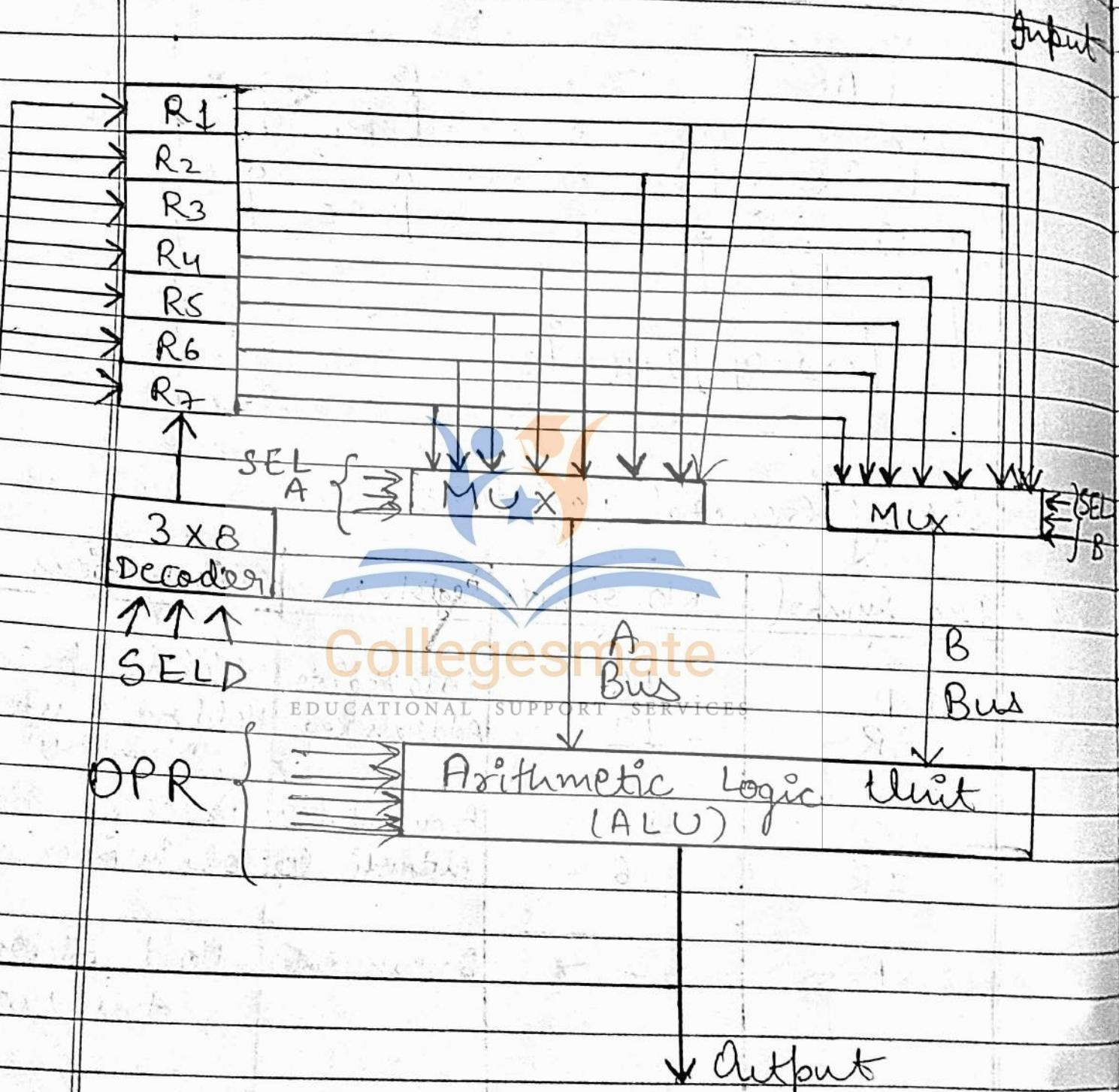
Registers in Basic Computer



List of General Registers

Reg Symbol	No. of bits	Register	Function
DR	16	Data Register	Hold operant
AR	12	Address Reg	Hold Address for Memory
AC	16	Accumulator	Processor Register
IR	16	Instruction Reg	Hold Instruction code
PC	12	Program Counter	Hold address of instruction
TR	16	Temp Reg	Hold Temp data
INPR	8	Input Reg	Hold Input char
OUTR	8	Output Reg	Hold O/P char

If Bus organization for 7 CPU registers are shown:



A bus organization for seven CPU registers is shown in fig. The OP of each register is connected to two multiplexers to form the two buses A and B. The selection lines in each multiplexer select one register or the input data for the particular bus.

The A and B buses from the input to a common ALU. The operation selected in the ALU determines the arithmetic or logic micro operation that is to be performed.

The result of the microoperation is available for OP data and also goes into the IP of all the registers.

The reg that receive the information from the OP bus is selected by a decoder. The decoder activates one of the reg load inputs providing a transfer path between inputs providing a the data in the OP bus and the input of the selected destination reg.

$R_1 \leftarrow R_2 + R_3$

- 1) Multiplexer A (MUX A) \rightarrow MUX A Selector (SEL A)
- 2) MUX B selector (SEL B)
- 3) ALU op^m selector (OPR)
- 4) Decoder destination selector (SEL D)

- ① To place the content R_2 into Bus A.
- ② To place the content R_3 into Bus B.
- ③ To provide the arithmetic addition
- ④ To transfer the content of the OP bus into R_1 .

Registers: Sequence of bits or sequence of flip flops

Generally size of registers = 16 bits

Memory

4096 \times 16 word size
No. of words

$(2^{12}) \times 16 \rightarrow$ word size
 \times no. of words

11

Address Register (AR) (12 bits)

12

000000
address

16 bits

12

15

Data Register (DR) (16 bits)

15

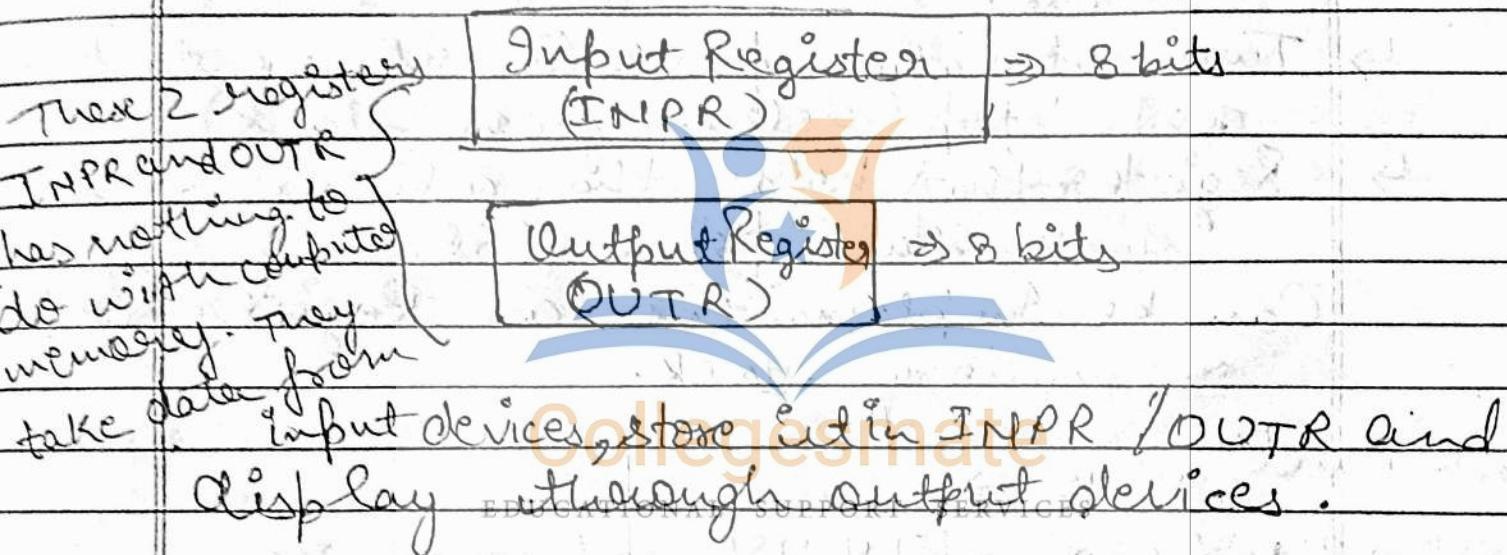
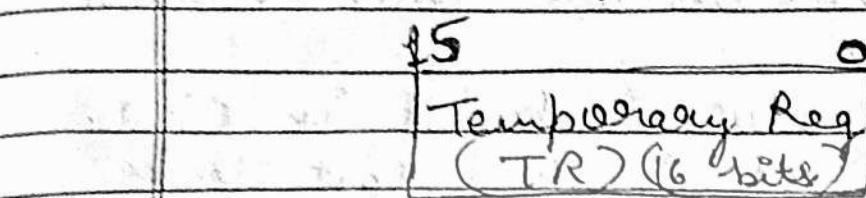
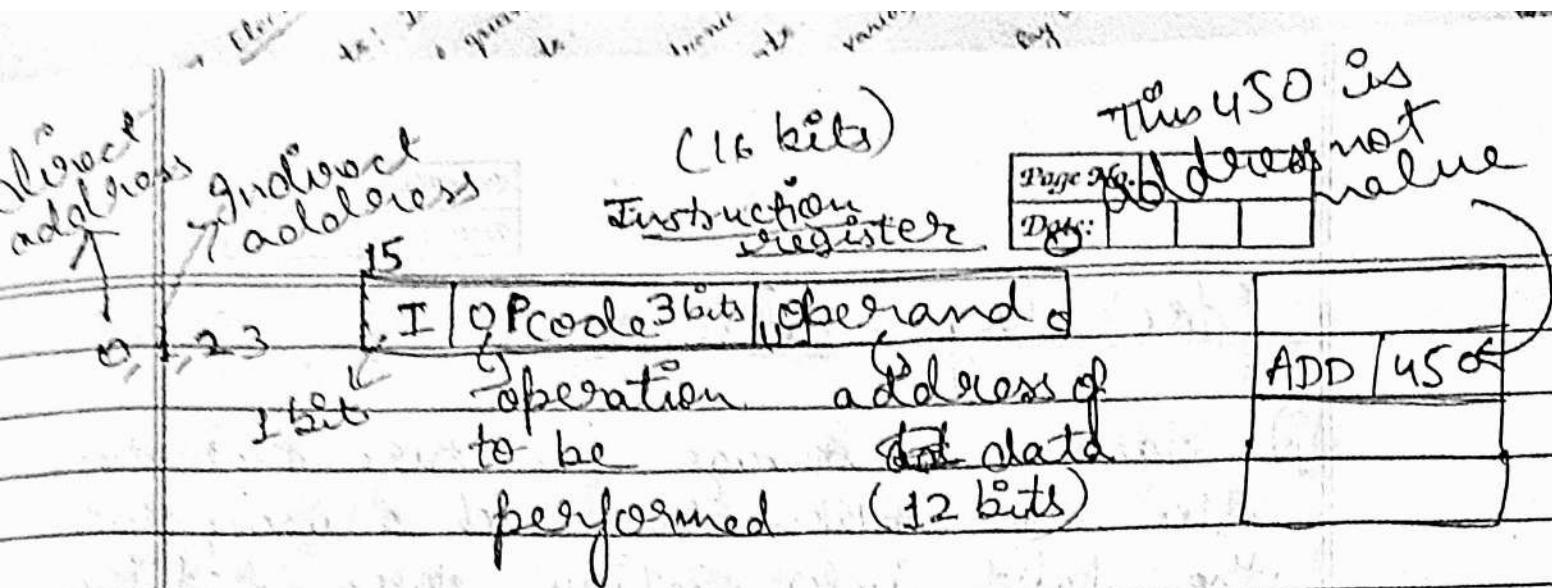
Accumulator (16 bits)

0

Stores address of next instruction

Program Counter (PC) (12 bits)

0



Stack Organization

- A. Stack is a storage structure that store the information in such a way that the last information stored retrieve first (LIFO).
- ↳ It is useful feature included in CPU.
 - ↳ Item stored first is the last to be retrieve.
 - ↳ Two operation push (insert to stack) and pop (delete from stack)
 - ↳ Register that holds the address of the stack - stack pointer (SP).

Can be implemented as Register Stack or memory stack.

Stack limit value →

Collegesmate

Stack Pointer (SP) →

EDUCATIONAL SUPPORT SERVICES

Stack Base value →

FREE

} Stack Area

(Stack - Organization)

Full = 1

when stack is full

EMPTY

FULL

EMPTY = 1
when stack
is Empty

Stack pointer
SP

PQRS
XYZ
ABCD

Type No.	
Date:	

Hold the data to be pushed onto stack or that is popped off

Data Register (DR)

Block Diagram of 64 word stack

PUSH Operation

$SP \leftarrow SP + 1$ increment stack pointer
 $M(SP) \leftarrow DR$ whole item top of the stack

If ($SP = 0$) then ($FULL \leftarrow 1$) Check if stack is full

$Empty \leftarrow 0$ mark the stack not empty

POP Operation

$DR \leftarrow M(SP)$ Read item from Top of stack

$SP \leftarrow SP - 1$ decrement SP

If ($SP = 0$) then ($EMPTY \leftarrow 1$)

Check if stack is empty

$FULL \leftarrow 0$

mark the stack not full

Stack Organization types

- ① Register Stack
- ② Memory Stack

Register Stack

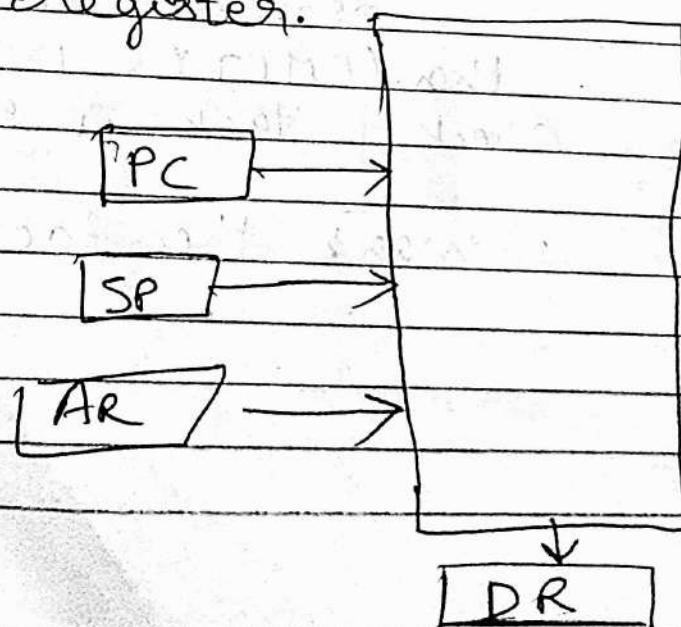
In Register stack we insert the digit 10, 20, 30 and the initial index of stack

is empty. In Reg. stack it store the register accept initial box of stack but it is always empty.

:	4
30	3
20	2
10	1
empty	0

Memory Stack

In Register stack register / Data is stored in the limitation for the large number of data stored we make memory stack means we say that in memory stack we store large no. of data as compare to register.





Addressing mode

The different kind of ways the programmer can refer to data stored in memory is known as addressing mode.

Most common addressing modes are:

- ① Immediate
- ② Direct
- ③ Indirect
- ④ Register
- ⑤ Register indirect
- ⑥ Displacement
- ⑦ Stack

 Collegemate
EDUCATIONAL SUPPORT SERVICES

Instruction

(Direct)

Instruction

Memory

Instruction

operand

(Indirect)

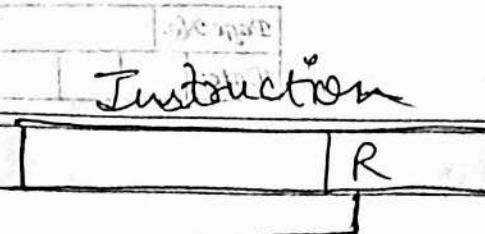
Instruction

R

Operand

(Register)

(3)



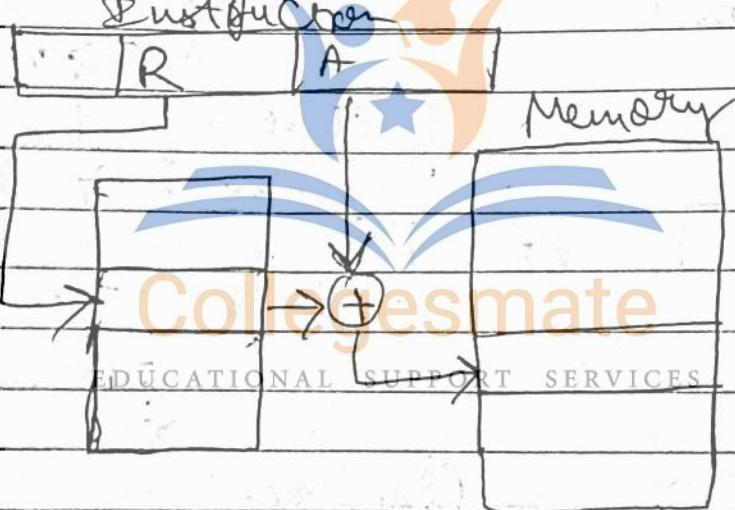
Memory

Operand

Register

(Register Indirect)

(4)



Memory

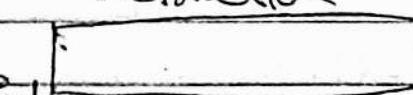
EDUCATIONAL SUPPORT SERVICES

(Displacement)

Instruction

(5)

Implicit



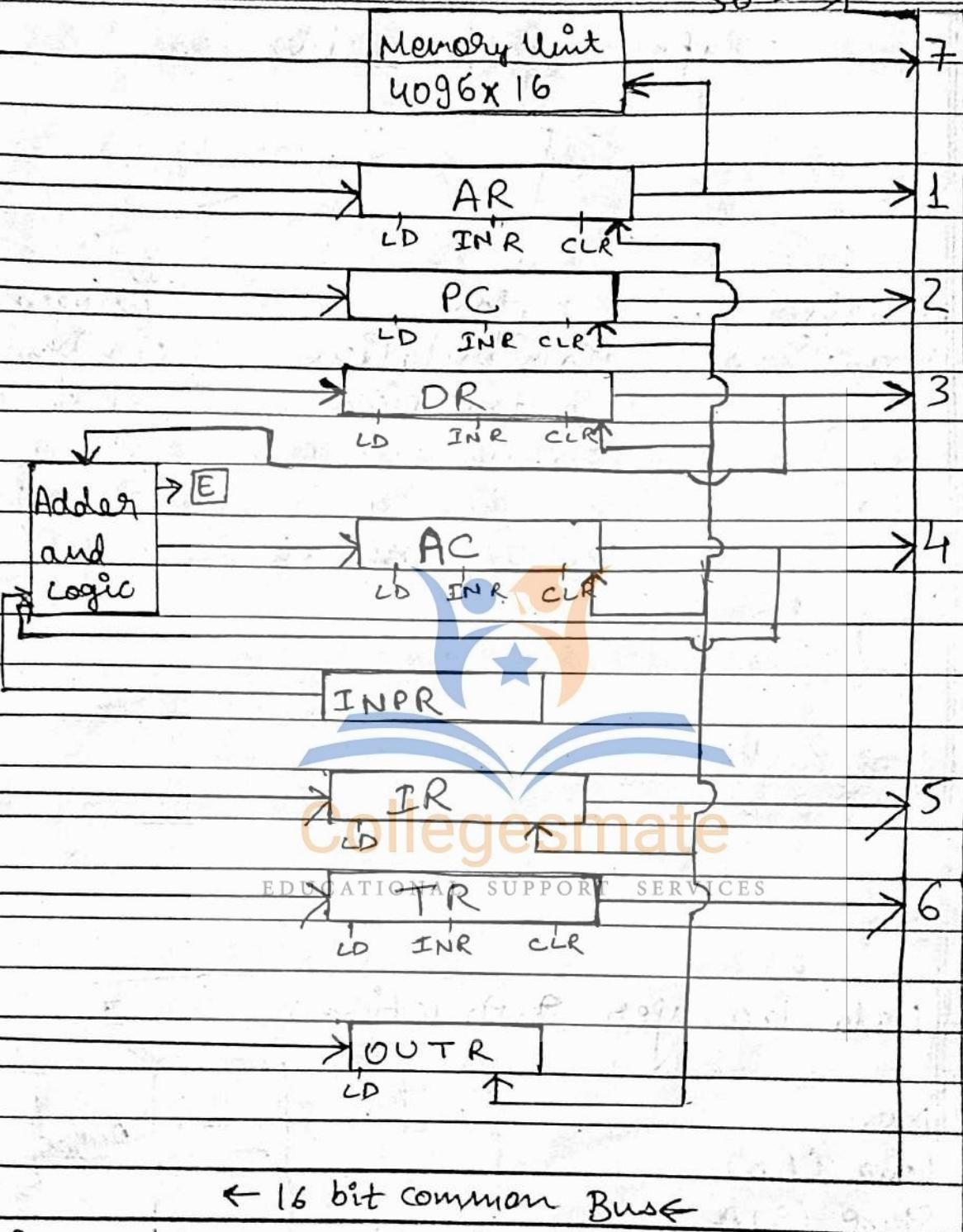
(Stack)

Top of stack Register

- * A → content of an address field in the instruction.
- * R → content of an address field in the instruction that refers to a register

*May not
play over*

Common Bus System | How basic computer works



One Address Instruction

$$X = (A + B) * (C + D)$$

Load A $AC \leftarrow M[A]$

Add B $AC \leftarrow AC + M[B]$

Store T

Load C $AC \leftarrow M[C]$

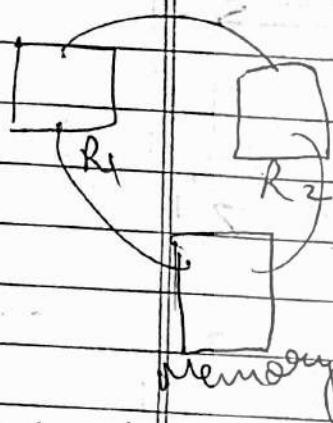
Add D $AC \leftarrow AC + M[D]$

MUL T

Store X

Types of Instructions

Data Transfer Instructions



- Load
- Store
- Move
- Exchange
- Push
- Pop

Data Manipulation Instructions

- Arithmetic
- Logical
- Shift/Conversion

Program Control Instructions

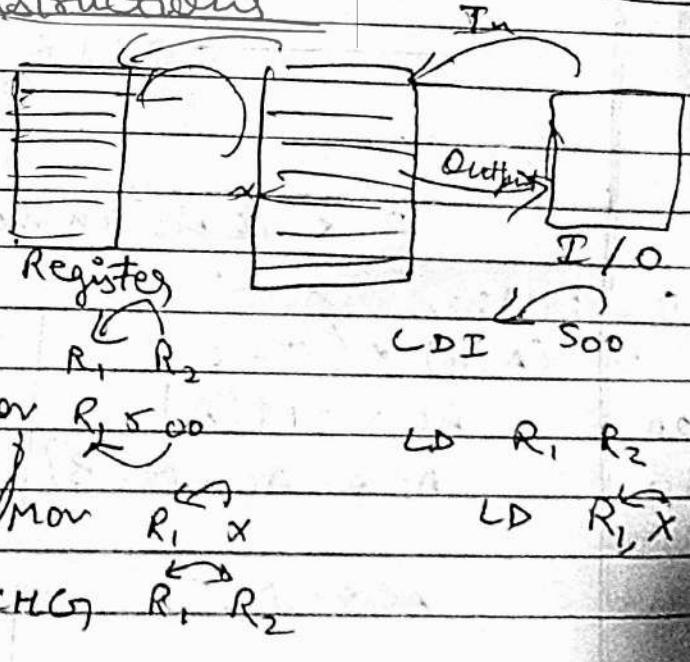
- if
- loop
- for
- while

Collegesmate

EDUCATIONAL SUPPORT SERVICES

For General Purpose computers Data Transfer Instructions

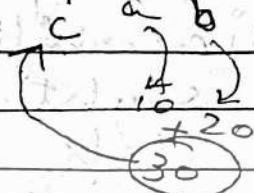
- Mov
- Load (LD)
- Store (STA)
- Exchange (XCHG)
- Input (IN)
- Output
- Push when memory is used as stack etc



Arithmetic Instructions (Data Manipulation)

- ↳ Add
- ↳ Sub
- ↳ MUL
- ↳ DIV
- ↳ INC (Increment)
- ↳ DEC (Decrement)
- ↳ Add with Carry
- ↳ Sub with borrow
- ↳ Negate

opcode	operand
+ IP, EIS	c a b



Logical Instructions (Data Manipulation)

- ↳ Complement (COM or NOT)
- ↳ Clear (CLR)
- ↳ logical-And (AND)
- ↳ logical - OR (OR)
- ↳ Ex-OR (XOR)
- ↳ Clear Carry (CLRC)
- ↳ Set Carry (STC)
- ↳ Complement Carry (CMC)
- ↳ Enable Interrupt (EI) 1 Flag
- ↳ Disable Interrupt (DI)

Collegesmate
EDUCATIONAL SUPPORT SERVICES

Page No.			
Date			

Shift Instructions (Data Manipulation)

- ↳ Logical shift left
 - ↳ Logical shift right
 - ↳ Arithmetic shift right
 - ↳ Arithmetic shift left
 - ↳ Rotate right
 - ↳ Rotate left
 - ↳ Rotate right through carry
 - ↳ Rotate left through carry
(most significant bit)

Both are same

~~left shift~~

Multiplication of 2

cognac

Right shift

(Divide by 2)

Collegesmate

EDUCATIONAL SUPPORT SERVICES

logical

left shift

(Multiplication of 2)

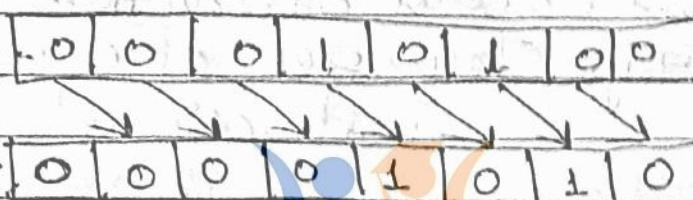
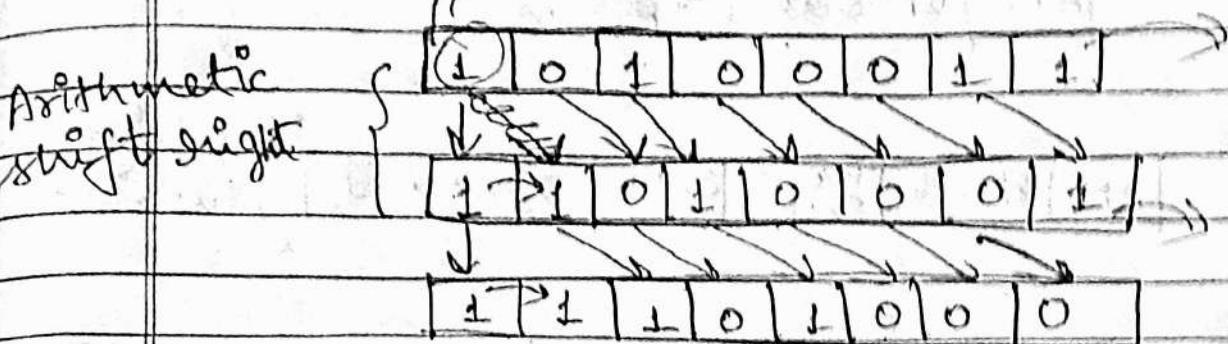
0000110110

A horizontal strip of material, possibly a scale or a ruler, featuring diagonal hatching and small arrows pointing towards the center.

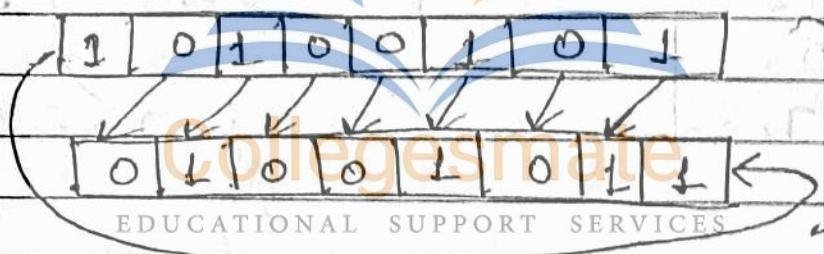
Arithmetic shift left and logical shift left are same

(MSB) and

(Most Significant bit & sign bit)

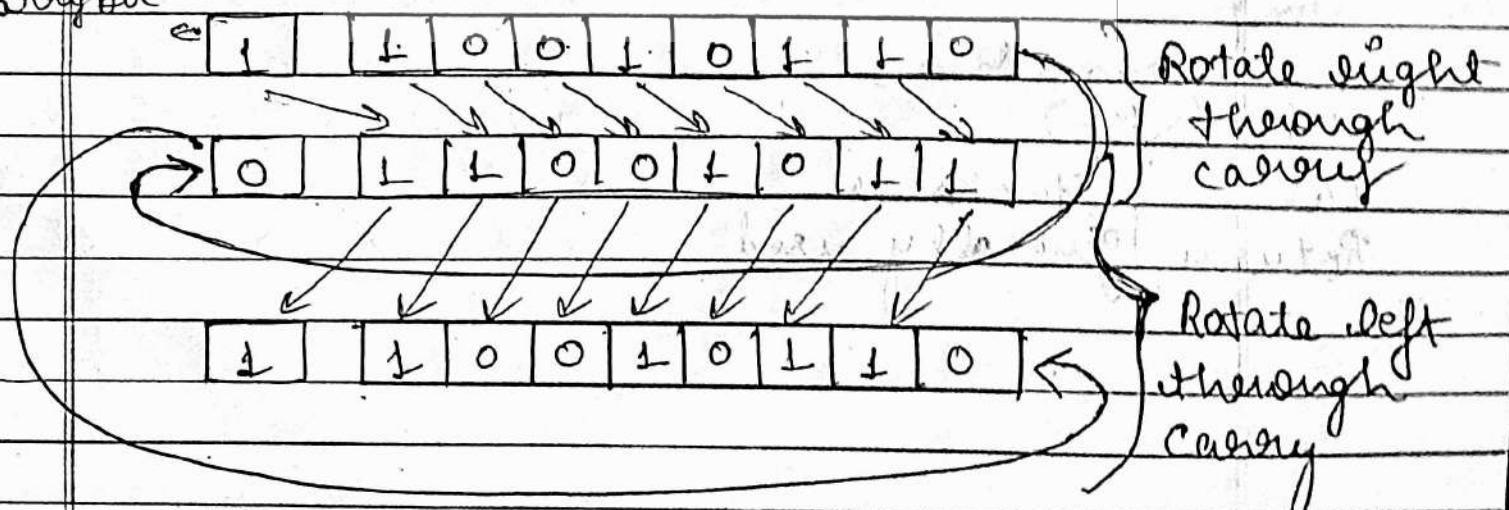


Rotate right



Rotate left

carry bit



Addressing mode

Instruction

[AM | opcode | operand]

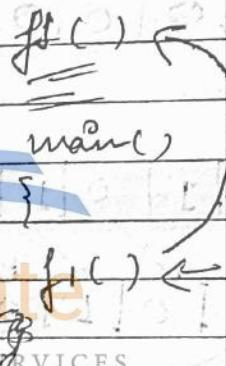
[AM | opcode | type of shift | count | operand] } Shift Instruction

Jmp
Jeq
Jne
Jgt
Jge
Jlt
Jle
Jset
Jseteq
Jsetne
Jsetgt
Jsetge
Jsetlt
Jsetle

Program Control Instructions /

Transfer of control
instructions

100	I ₁	PC
101	I ₂	101
102	I ₃	
103	I ₄	

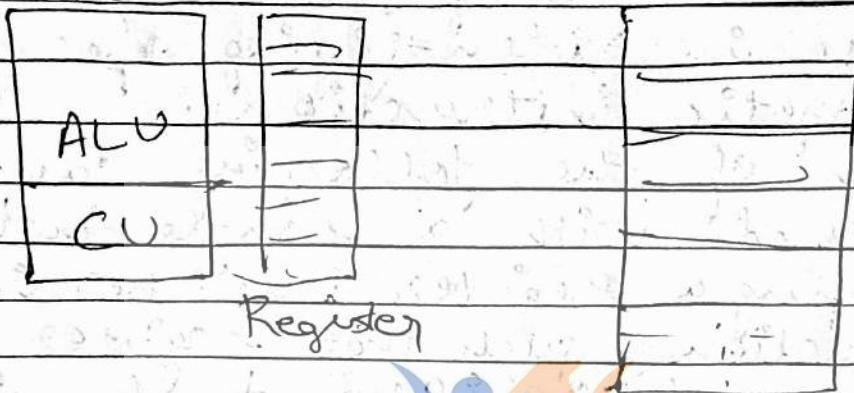


Page No.	
Date:	

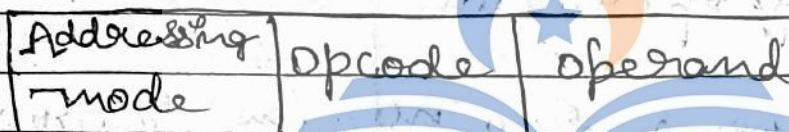
Instruction Format

CPU

main()



$\text{int } a = 0;$ Data
 $\text{int } b = 0;$
 $\text{int } c;$
 $c = a + b;$ Instruction
 $\text{print}(c);$ Instruction



Collegesmate

EDUCATIONAL SUPPORT SERVICES

A certain architecture supports indirect, direct and register addressing mode for use in identifying operands of arithmetic instructions.

Which of the following cannot be achieved with a single instruction.

(a)

Register
direct

Specifying a register number in instruction, such that register contains a value of operand that will be used by operation.

(b)

register
indirect

Specifying a reg no. in instruction such that register will serve as destination of opⁿ output.

(c)

immediate

Specifying an operand value in instruction such that value will be used by opⁿ.

(d)

Direct

Specifying a memory location such that it contains value of operand that will be used by operation.

Address

Memory

200	Load for AC Mode
201	Add res = 500
202	Next instruction
399	450
400	700
500	800
600	900
702	325
800	300

$$R_1 = 400$$

EDUCATIONAL SUPPORT SERVICES

PCT 200

$$XR = 100$$

[AC]

XR - Index ; R₁ - Register

Memory having first instruction to Load AC Mode will specify the addressing mode to get operand

Address field of instruction is 500
 Find out the effective address of operand and operand value by considering different addressing mode.

Addressing mode: EA

Content of operand

Content of accumulator

Direct 500 800

Indirect 800 300

~~Immediate~~ 500 800

Register direct 400 700

Register-indirect 800 300

1) Immediate 201 500

2) Direct 500 800

3) Indirect 800 300

4) Auto decrement 400 700

5) Auto increment 399 450

Collegesmate

SUPPORT SERVICES