



# POORNIMA

## COLLEGE OF ENGINEERING

### DETAILED LECTURE NOTES ①

#### Computer Architecture and Organization

##### 1. Computer data representation: →

1. Basic computer datatype

2. Complements

3. fixed point representation

4. Register Transfer & Micro operations

- floating point representation

- Register Transfer lang.

- Register Transfer

- Bus & Memory Transfers (Tree-state  
Bus-Buffers)  
Memory

- Logic Micro Operations

- Shift Micro-Operations

- Arithmetic logical shift unit

- Basic Computer Organization & Design, Instruction codes

- Computer register, Computer Instructions

- Timing & Control, Instruction cycle

- Memory-Reference Instructions, Input op and interrupt

- Complete Computer Description, Design of Basic Computer Design of Techniques

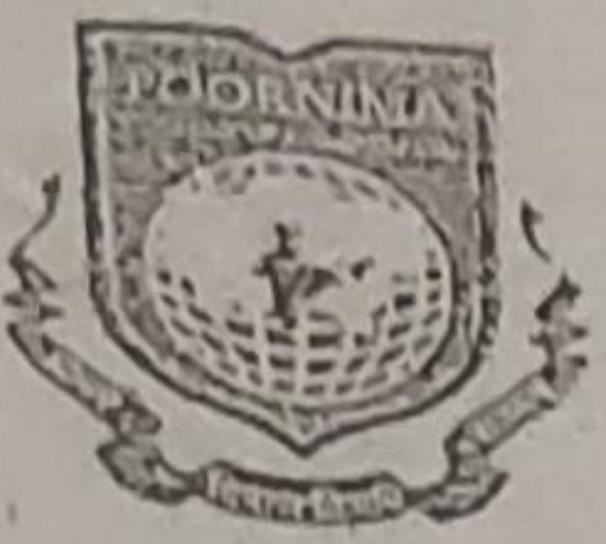
## Digital:-

- Digital computer is a digital system that performs various computational tasks. Digital computer use the binary number system, which has two digits of 1 (bit)
- A computer system is subdivided into two parts entities
  - Hardware
  - Software.
- { Electronic component and electromechanical Devices }
- { Instruction and data }
  - Set of instructions is called program.
- Hardware of computer is divided into three parts.
  - 1. CPU (Central Processing Unit)
  - 2. RAM (Random Access Memory)
  - 3. Input Output Processor (IOP)

containing  
① Arithmetic & logic unit for manipulating data  
② No of Registers for storing data  
③ Control unit for fetching & executing instructions

→ CPU can access RAM only.

IOP contains electronic circuits for control & control the transfer of info b/w the computer and outside world (keyboard, mouse, magnetic disk)



# Poornima

COLLEGE OF ENGINEERING

## DETAILED LECTURE NOTES

PAGE NO. ....

### Computer Architecture v/s Computer Organization

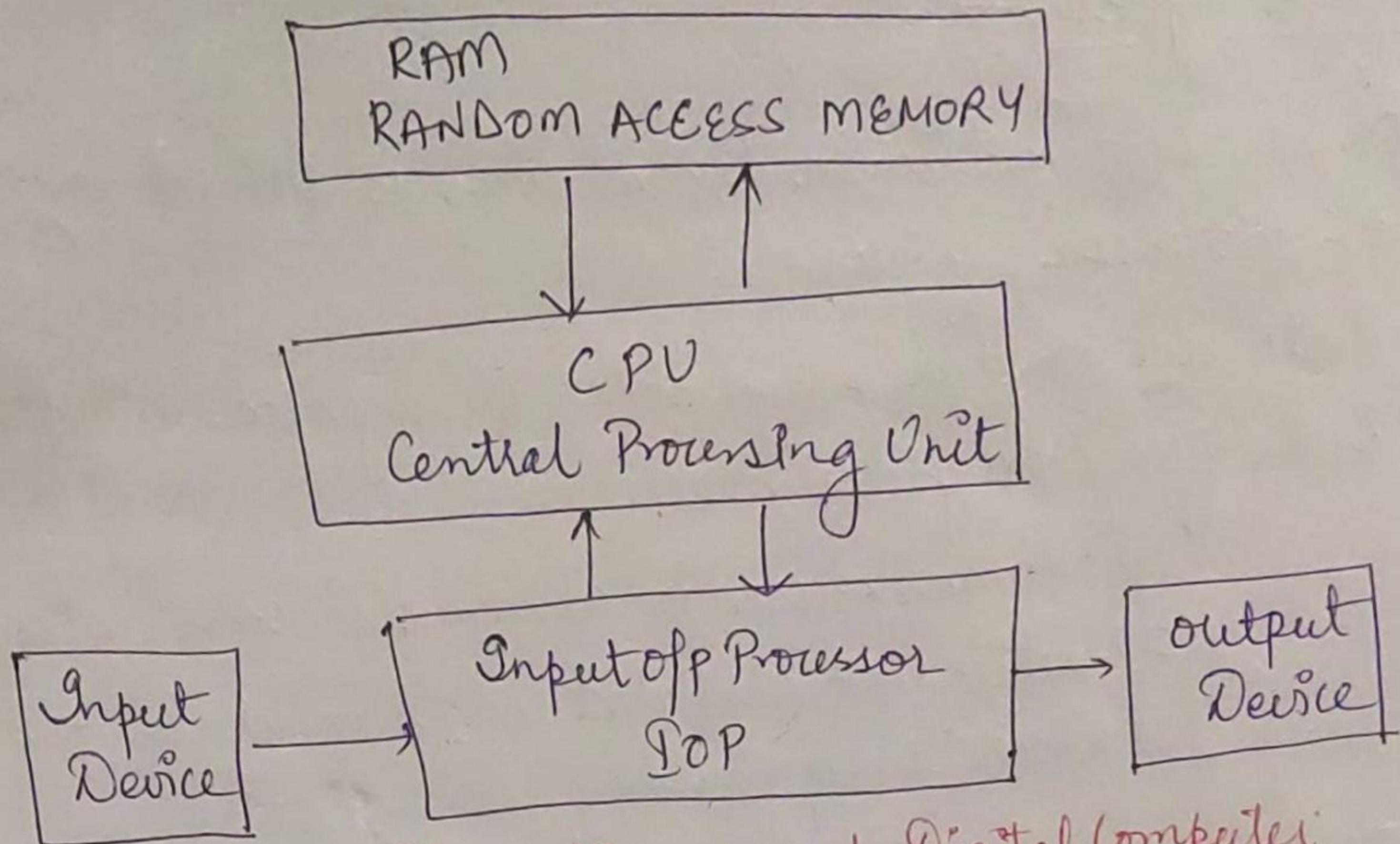
- ① CA is concerned with the way hardware components are connected together to form a computer system.
- ② CO is concerned with the structure and behaviour of the computer system.
- ② It acts as interface b/w H/W & S/W
- ② It deals with the components of a connection in a system.
- ③ CA help to understand the functionality of system.
- ③ CO tells us how exactly all the units in system are arranged and interconnected.
- ④ Programmer can view Architecture in terms of instructions, addressing modes & registers
- ④ CO basis of Architecture
- ⑤ Deal with high design issues
- ⑤ Deal with low-level design issues
- ⑥ Involves → logic (Instruction set)  
→ Addressing  
→ Datatype  
→ Cache
- ⑥ Physical components
  - (1) Circuit
  - (2) Adders
  - (3) Signals
  - (4) Peripherals



# POORNIMA

COLLEGE OF ENGINEERING

## DETAILED LECTURE NOTES ②



• Block diagram of Digital Computer

### Computer Organization :-

Hardware components operate and connected together to form the computer system.

### Computer Design :-

- Hardware design of the computer.

Computer design is concerned with the determination of what hardware should be used and how the parts should be connected.

## Computer Architecture :-

- Structure & Behaviour of the computer.
- It include info format, instruction set & technique for addressing memory.

## Data Types :-

Binary information in digital comp is stored in memory or processor registers.

- Data are numbers and other binary-coded information.
- Data types in register are
  - (1) Numbers used in arithmetic computation.
  - (2) Letters of alphabet used in data processing
  - (3) Discrete symbols.
- All types of data represented in computer registers in binary-coded forms (Because register are made up of flip-flop and flip-flop are two-state devices that can store only 1's or 0's)

## Number Systems :-

① Decimal:-  $724 = 7 \times 10^2 + 2 \times 10^1 + 4 \times 10^0$   
 $= 700 + 20 + 4 = 724$

② Binary  $101101 \rightarrow 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$   
 $= 32 + 0 + 8 + 4 + 0 + 1$



# POORNIMA

## COLLEGE OF ENGINEERING

### DETAILED LECTURE NOTES

③ Octal  $(736)_8 = 7 \times 8^2 + 3 \times 8^1 + 6 \times 8^0$   
 $= 7 \times 64 + 24 + 6$   
 $= 448 + 30 = (478)_{10}$

④ HexaDecimal  $(F3)_{16} = F \times 16^1 + 3 \times 16^0$   
 $= (243)_{10}$

Conversion: →

$(41.6875)_{10}$  = Binary

$\begin{array}{r} .6875 \\ \times 2 \\ \hline 1.3750 \\ \times 2 \\ \hline 0.7500 \\ \times 2 \\ \hline 1.5000 \end{array}$

$$(101001)_2 + (1011)_2 \quad \begin{array}{r} \times 2 \\ \hline 100000 \end{array} \quad 1$$

$$= (101001 \cdot 1011)$$

## Alphanumeric representation:-

26 letters, special character 32 to 64.

↓  
64, 128,

— in Binary coded form

for upper & lower case.

## Complements :-

- Complements are used in digital computer for simplifying subtraction operation & logical manipulation
- For  $r$ 's (base) system — (1)  $r$ 's complement  
(2)  $(r-1)$ 's complement

e.g. Binary ( $2=r$ ) — 1's complement  
2's complement

## $(r-1)$ 's complement:-

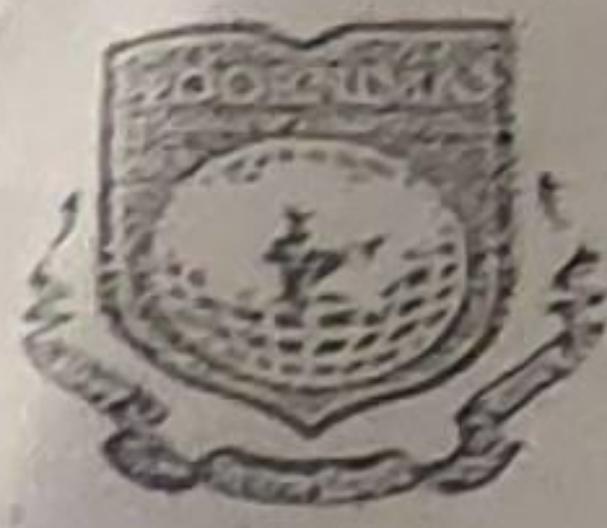
$N_r$  (n digits)

$$r-1 \text{'}s \text{ complement} = (r^n - 1) - N$$

~~$$r-1 \text{'}s \text{ complement}$$~~
$$10_{10} (n=2) = (10^2 - 1) - 10 = 99 - 10 = 89$$

$$546700_{10} (n=6) = (10^6 - 1) - 546700$$
$$= 999999 - 546700$$
$$= 453299$$

$$\begin{aligned} n=5 & \quad 12389 \text{ is } = (10^5 - 1) - 12389 \\ r=10 & \quad = \underline{87610} \end{aligned}$$



# POORNIMA

## COLLEGE OF ENGINEERING

### DETAILED LECTURE NOTES

PAGE NO. 4

1's complement -

$$\textcircled{1} \quad (1011001)_2 = 2^7 - 1 = 10000000 - 1 \\ = \begin{array}{r} 1111111 \\ - 1011001 \\ \hline 0100110 \end{array}$$

e.g.  $n=4$

$$2^n = 10000$$

four zeros followed by 1

$$\textcircled{2} \quad (0001111)_2 \quad n=7 = 2^7 - 1 \\ = \begin{array}{r} 1111111 \\ - 0001111 \\ \hline 1110000 \end{array}$$

2's complement :

$N_2$  (n-digit)

$$2^n - N \quad \text{for } N \neq 0$$

$$0 \text{ for } N=0$$

$$\text{e.g. } -2389_{10} = 10^4 - 2389 \quad n=4$$

$$\begin{aligned} 10\text{'s complement} &= 10000 - 2389 \\ &= \underline{\underline{7611}} \end{aligned}$$

$$\begin{aligned} 10^4 \text{ complement} &= (10^4 - 1) - 2389 \\ &= 9999 - 2389 = \underline{\underline{7610}} + 1 = \underline{\underline{7611}} \end{aligned}$$

Thus 10's complement of the decimal is obtained by adding 1 to the 9's complement.

$$246700 = 10^6 - 246700 \\ = 753300$$

2's complement:-

$$\begin{aligned} \therefore 1101100 &= 2^8 - 1101100 \\ &= 10000000 - 1101100 \\ &= 0010100 \end{aligned}$$

Subtraction of Unsigned Number: →

Subtraction of  $n$ -digit No. unsigned  
 $M-N$  ( $N \neq 0$ ) with base  $r$ .

1. Add minuend  $M$  to the  $r$ 's complement of the subtrahend  $N$ .

$$M + (r^n - N) = M - N + r^n$$

2. If  $M \geq N$ , sum will produce end carry  $r^n$  discard &  $(M-N)$  value is left, which is answer.

3. If  $M < N$ , sum does not produce ~~an~~ end carry and is equal to  $r^n - (N-M)$ , which is  $r$ 's complement of  $(N-M)$

Answer =  $r$ 's complement of sum and place negative sign in front.



# POORNIMA

COLLEGE OF ENGINEERING

## DETAILED LECTURE NOTES

eg:  $\rightarrow \cancel{735} \quad 72532 - 13250 = 59282$

(P)  
PAGE NO.

10's complement of 13250 =  $\begin{array}{r} 100000 \\ - 13250 \\ \hline 86750 \end{array}$  ( $10^5 - N$ )

Therefore

$$\begin{array}{rcl} M > N & = & M = 72532 \\ & & 10's \text{ of } N + 86750 \\ & & \hline & & 59282 \\ & & \text{discard carry} \end{array}$$

eg:  $M < N \quad 13250 - 72532$   
 $= -59282$

$$\begin{array}{rcl} M + (r^n - N) & \rightarrow & \\ M = 13250 & & \\ 10's \text{ of } N = 27468 & & \\ \hline \text{sum} = 40718 & & \text{no carry} \end{array}$$

10's complement of sum

$$\begin{array}{r} 100000 \\ - 40718 \\ \hline - 59282 \end{array}$$

put -ve sign in front.

eg.  $X = 1010100 \rightarrow X - Y$

$$Y = 1000011$$

$$X - Y = \begin{array}{r} X \\ \text{2's complement of } Y \\ \hline \end{array} = \begin{array}{r} 1010100 \\ + 0111101 \\ \hline 0010001 \end{array}$$

Carry discarded.

$$\text{Answer} = 0010001$$

if  $Y - X$  ( $Y < X$ )

$$Y = 1000011$$
$$\text{2's of } X = \begin{array}{r} 1000000 \\ + 0101100 \\ \hline 1101110 \end{array} = \text{sum}$$

no carry.

$$\text{so 2's of sum} = \boxed{-0010001} \quad \text{Answer.}$$



# POORNIMA

PAGE NO. 6

## DETAILED LECTURE N

In general positive & negative nos are represented by sign (+ve or -ve). But in computer system every no. is presented in the form of 0's or 1's

So:- Sign bit is there

## ① Integer Representation :-

Three possible ways

- ① Signed-magnitude representation
  - ② Signed-1's complement representation
  - ③ Signed-2's complement representation

- With the help of these possible way we represents the integer nos. like if no is 14 and stored in 8 bit

$+14 \rightarrow 0,00001110$  } only one way to register  
for positive no }

but to represent  $-14$  {to represent -ve no., 3 ways are

- ① Signed - 10001110
  - ② 1's - 1,11100001
  - ③ 2's - 11110010

- Signed-magnitude is used in ordinary arithmetic, or normally used
- It's used for arithmetic operations in some older computers
- It's for logical operation and it is also used.

## ② ~~Add~~ Arithmetic Addition

- ① Signed-magnitude system
- Ordinary arithmetic rule

$$\begin{array}{r}
 +25 \\
 +37 \\
 \hline
 +52
 \end{array}
 \quad
 \begin{array}{r}
 +25 \\
 -32 \\
 \hline
 -12
 \end{array}
 \quad
 \text{use comparison to find greater no.}$$

## ② 2's complement

- No need of comparison
- Add two no.s including sign bit and discard carry

eg:

$$\begin{array}{r}
 +6 \quad 00000110 \\
 +13 \quad 00000101 \\
 \hline
 +19 \quad 00010011
 \end{array}$$

$$\begin{array}{r}
 \cancel{2's of 6} \quad 11111010 \\
 -6 \quad \boxed{10000110} \\
 +13 + \cancel{00001101} \\
 \hline
 \end{array}$$

$$\begin{array}{r}
 +7 \quad \cancel{100000111} \\
 \cancel{\text{discard sign +ve}}
 \end{array}$$

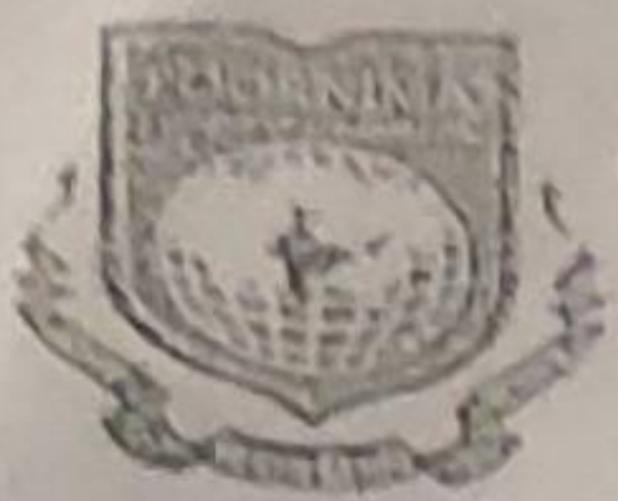
$$\begin{array}{r}
 +6 \quad 00000110 \\
 -13 \quad 11110011 \\
 \hline
 \leftarrow 11111011
 \end{array}$$

$\leftarrow$  2's

$$\begin{array}{r}
 2's - 6 \quad 11111010 \\
 2's - 13 \quad 11110011 \\
 \hline
 \cancel{11101101} \\
 \cancel{\text{discard sign}}
 \end{array}$$

$$\begin{array}{r}
 -7 \quad \leftarrow 10000110
 \end{array}$$

$$\begin{array}{r}
 \boxed{-19} \leftarrow 10010011
 \end{array}$$



# POORNIMA

COLLEGE OF ENGINEERING

## DETAILED LECTURE NOTES

PAGE NO. 7

### Arithmetic Subtraction:-

$$\pm A - (+B) = \pm A + (-B)$$

$$\pm A - (-B) = \pm A + (+B)$$

eg:  $-6 - (-13) = +7$

is  $11111010$  is  $10001101$  = ~~11111011~~  
now  $-13$  f. subtract  $= +13$

$$\begin{array}{r} +13 \\ -6 \\ \hline 00001101 \\ 11111010 \\ \hline 00000011 \end{array} \rightarrow +7$$

Discard  $\leftarrow$  If both the digit sign is plus and overflow generated than that value can't be discarded.

So, for that, first check the sign of both digits and accordingly discard or accept the carry value.

## Floating-Point Representation:-

①  $+6132.789 \rightarrow$  exponent  
 ↓  
 ① fixed point no.  
 mantissa  
 fraction Exponent  
 $+0.6132789 +04 \rightarrow m \times 2^e$   
 $+0.6132789 \times 10^{+4}$

② Binary  $+1001.11$   
 fraction Exponent  
 +ve  $\begin{array}{ll} 01001110 & 000100 \end{array}$  means 4 so  $2^4$ .  
 $(.1001110) \times 2^{+4}$

## Other-Binary Codis:-

① Gray Code: $\rightarrow$  Gray code changes by only one bit as it sequences from one number to next.

like 0 to 1 or 1 to 1

0	0000	}
1	0001	
2	0010	
3	0011	
4	0101	
5	0100	



# POORNIMA

COLLEGE OF ENGINEERING

## DETAILED LECTURE NOTES

PAGE NO. 8

### ② Other Decimal Codes:-

-- Require min of four bits.

BCD

EBCDIC

four diff Binary codes for the Decimal Digit

Decimal Digit	BCD	2421	Excess-3	Excess-3 Gray
0	0000	0000	0011	0010
1	0001	0001	0100	0110
2	0010	0010	0101	0111
3	0111	0011	0110	0101
4	0100	0100	0111	0100
5	0101	0101	1000	1100
6	0110	0110	1001	1101
7	0111	0111	1010	1110
8	1000	1110	1011	1110
9	1001	1111	1100	1010

Questions:-

① Convert the following decimal no. to the bases indicated

(1) 7562 to octal → 16612

(2) 1938 to hexadecimal → E9L / 792

(3) 175 to binary 1010111

② 9's complement of

12349876, 90009951

③ Obtain 1's & 2's complement

10101110

10000001

00000001

④ Perform Arithmetic Operation using 10's comple.

(1) (-638) + (+785)

(2) (-638) - (+185)



# POORVIMA

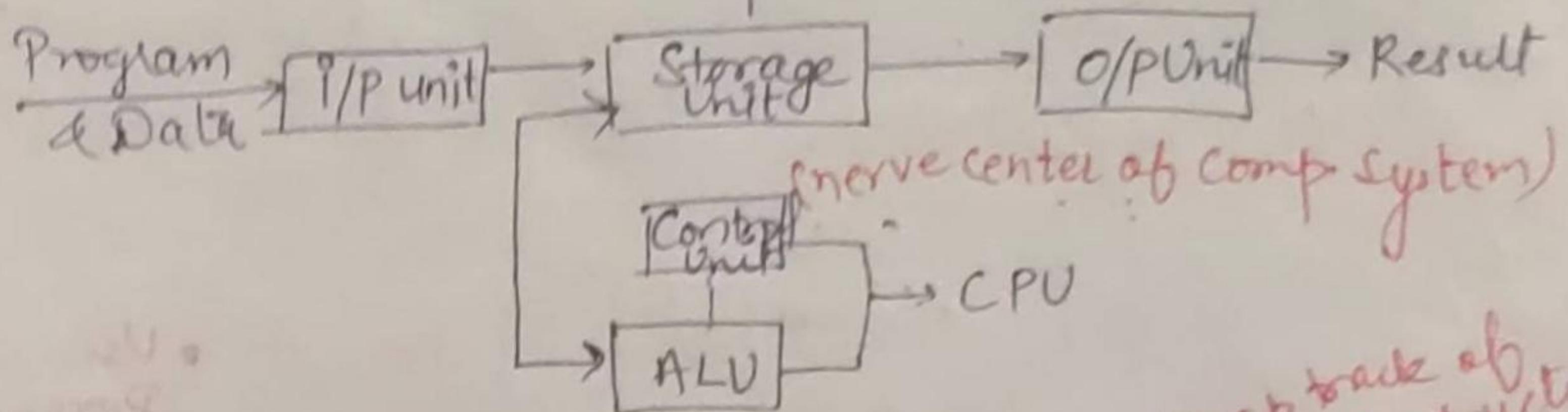
COLLEGE OF ENGINEERING

## DETAILED LECTURE NOTES

①

### Functional Units of Digital System:-

it's of two type  $\rightarrow$  Primary  
 $\uparrow$   $\rightarrow$  Secondary



Special Register are

Program Counter (PC)

Instruction Registers (IR)

Memory address registers (MAR)

Memory data register (MDR)

keep track of  
which instruction  
is being executed  
& next instruction

hold  
current instruction

add of  
memory

data to be  
written into or  
read from MM.

handle  
data transfer

b/w main memory  
& processor

Computer Architecture divided into two classification Unit

1. Storage Program Control Concept

2. Flynn's classification of computers.

1. Store Program Control Concept

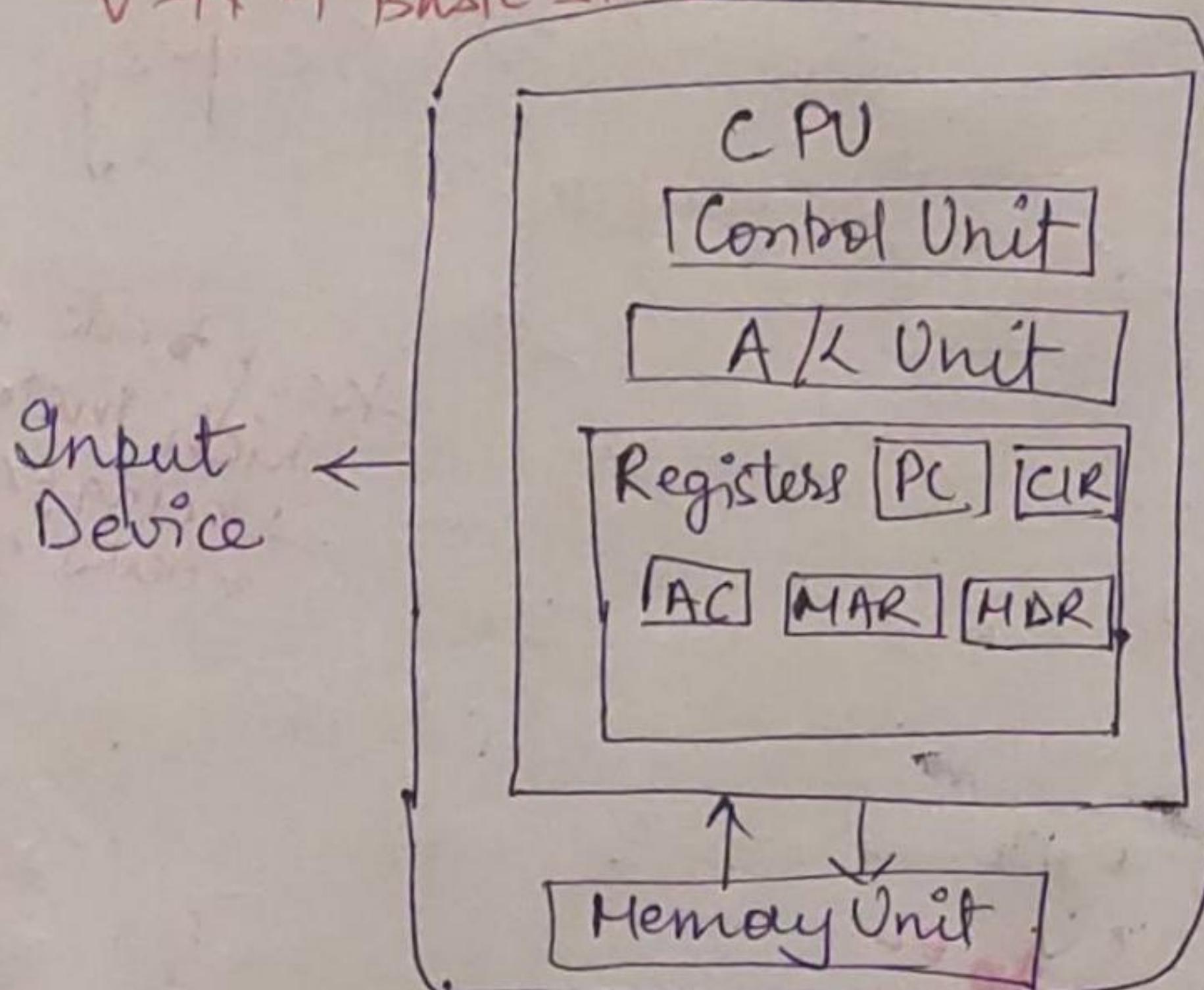
Von-Neumann Model

General Purpose System

It is storage of instruction  
in comp. memory to enable it to perform a variety of tasks  
Parallel Processing

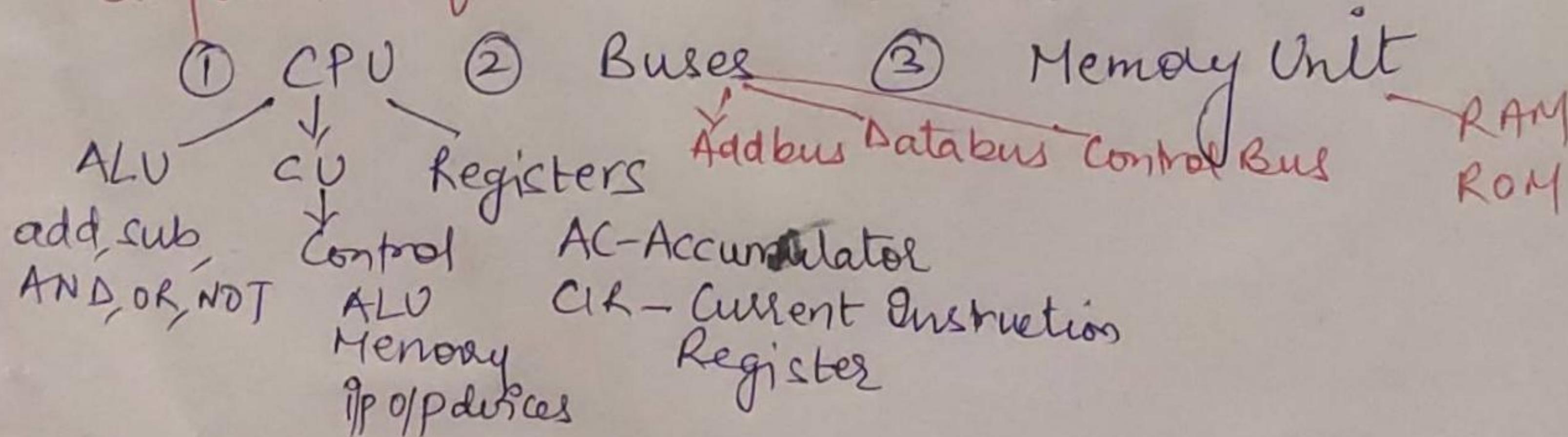
CA in 1945, consist of Control Unit, ALU, Registers & I/P, O/P

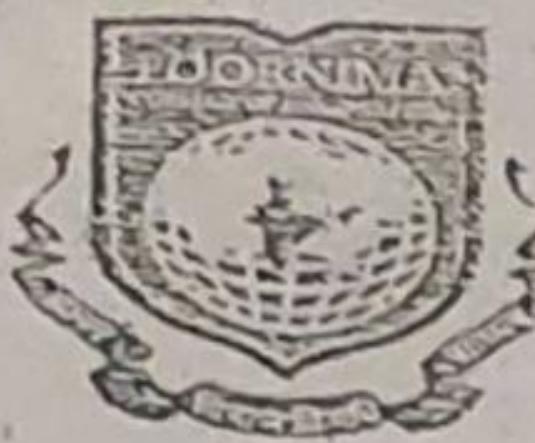
V-N M Basic Structure



- Uses single processor
- Uses one memory for both instruction & data
- Executes programs following the fetch-decode-execute cycle.

Components of Von-Neumann Model:





# Poornima

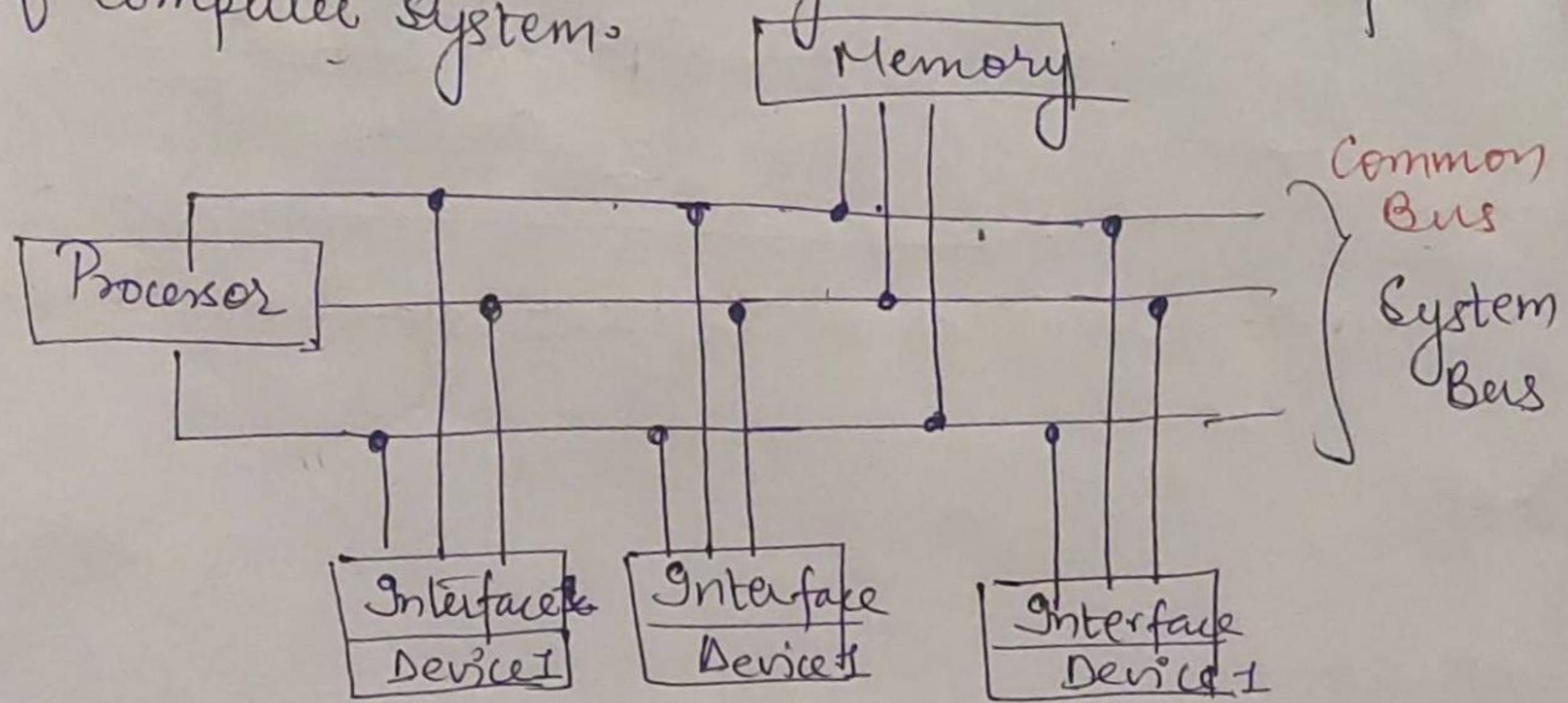
COLLEGE OF ENGINEERING

## DETAILED LECTURE NOTES

PAGE NO. 2

### General Purpose System:-

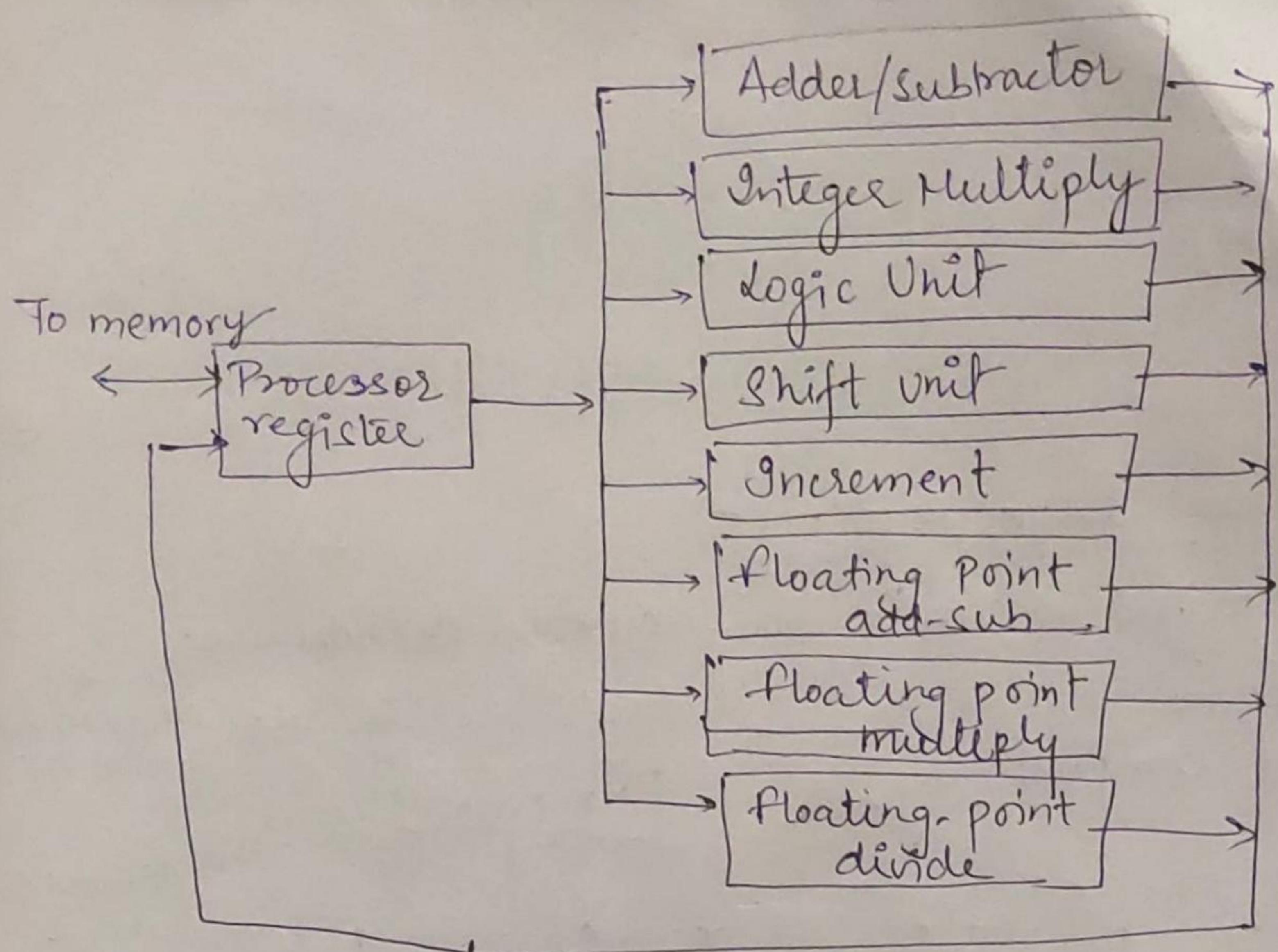
- Modified version of Von-Neumann Architecture.
- So we can say it is today's architecture representation of computer system.



### Parallel Processing:-

It enables the system to achieve simultaneous data-processing tasks to increase the computational speed of a computer system.

- Main goal is to enhance the processing capability and increase its throughput



— perform identical or different operations simultaneously



# POORNIMA

COLLEGE OF ENGINEERING

## DETAILED LECTURE NOTES

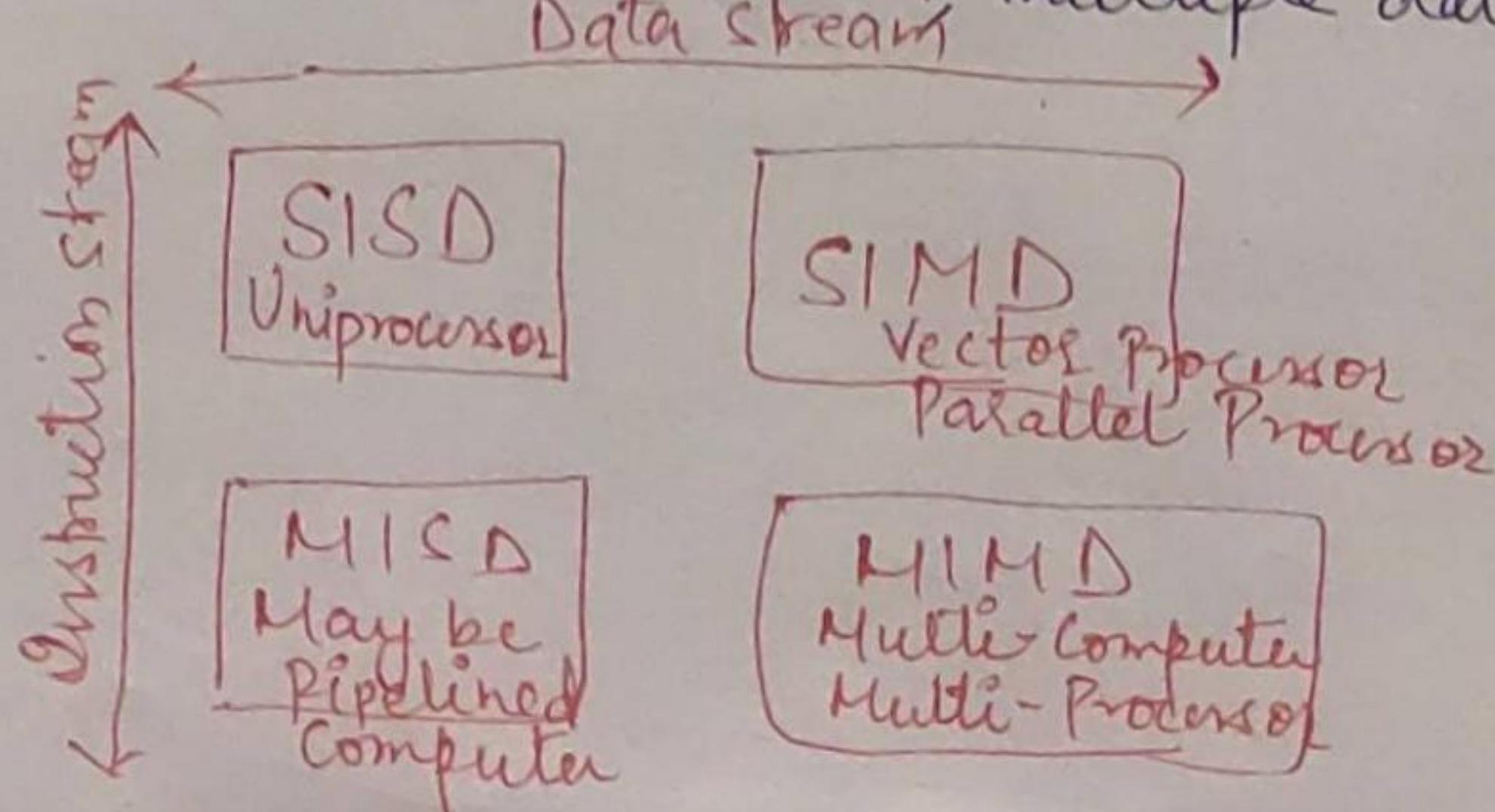
PAGE NO. 3

### Flynn's classification of Computers:

- M.J Flynn proposed a classification for the organizations of computer system by the no. of instructions and data items that are manipulated simultaneously.
- The sequence of instructions read from memory constitutes an instruction stream.
- The sequence of operations performed on the data in the processor constitute a data stream.

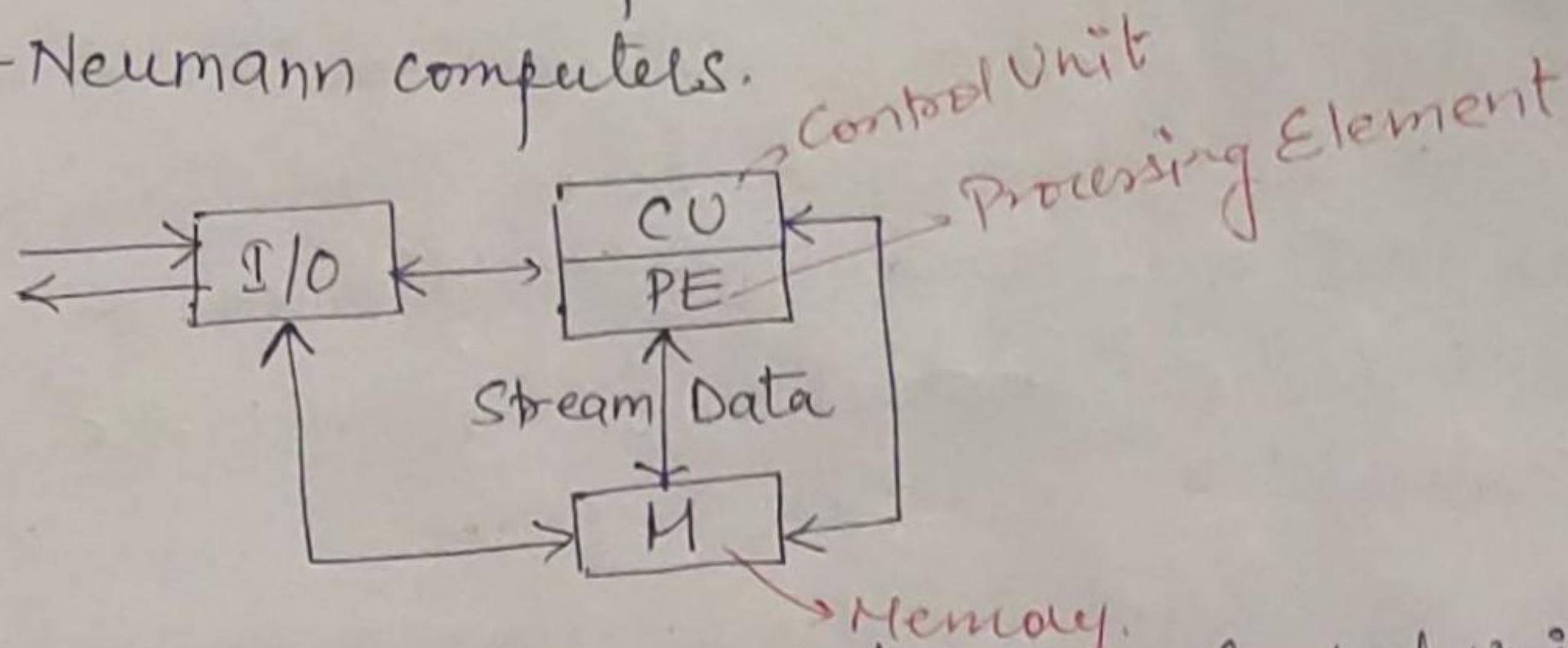
Flynn's classification divides computer into four major groups that are:

1. Single instruction stream, single data stream (SISD)
2. Single Instruction stream, multiple data stream (SIMD)
3. Multiple instruction stream, single data stream (MISD)
4. Multiple Instruction stream, multiple data stream (MIMD)



## ① SISD

- It represents the organization of a single computer containing a control unit, a processor unit & memory unit.
- Instructions are executed sequentially, system may not have internal parallel processing capabilities.
- Most conventional computer have SISD like traditional Von-Neumann computers.



- Instructions are decoded by the control unit & then control unit sends the instructions to the processing units for execution.
- Data stream flow b/w the processor & memory bi-direction

e.g.: - Older generation computers, minicomputers & Workstations.



# POORNIMA

COLLEGE OF ENGINEERING

## DETAILED LECTURE NOTES

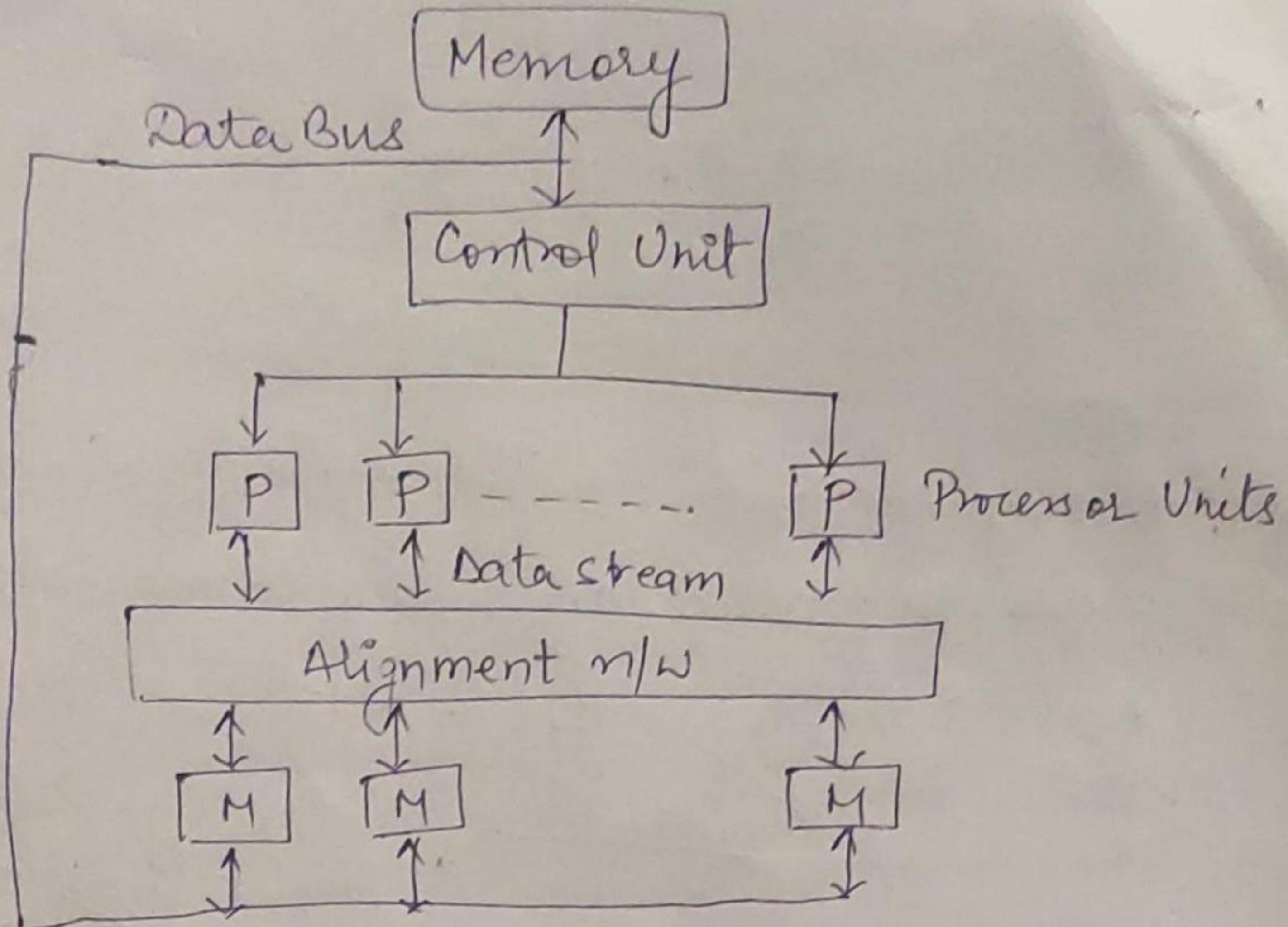
PAGE NO. 4

~~- Bus System with multiplexers, k register of n bits to produce an n-line common bus  
eg: 8 register with 16 bits~~

~~16 multiplexers → Each register have 8 input lines and three selection line.~~

### SIMD

- It represents an organization that includes many processing units under the supervision of a common control unit.
- All processor receive the same instruction from the control unit but operate on different items of data.
- The shared memory unit must contain multiple modules so that it can communicate with all the processors simultaneously.



SIMD is mainly dedicated to array processing m/c. However, vector processors can also be part of SIMD.



# POORNIMA

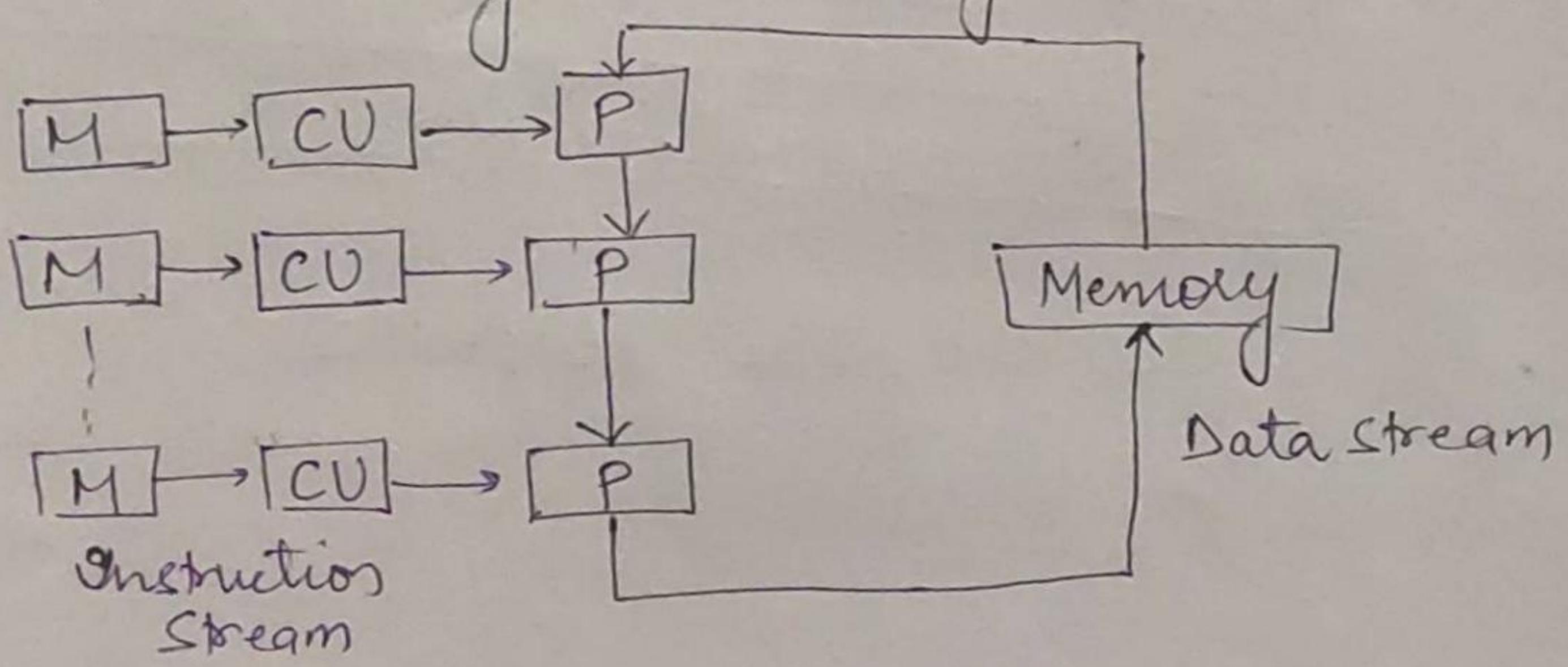
COLLEGE OF ENGINEERING

## DETAILED LECTURE NOTES

MISD

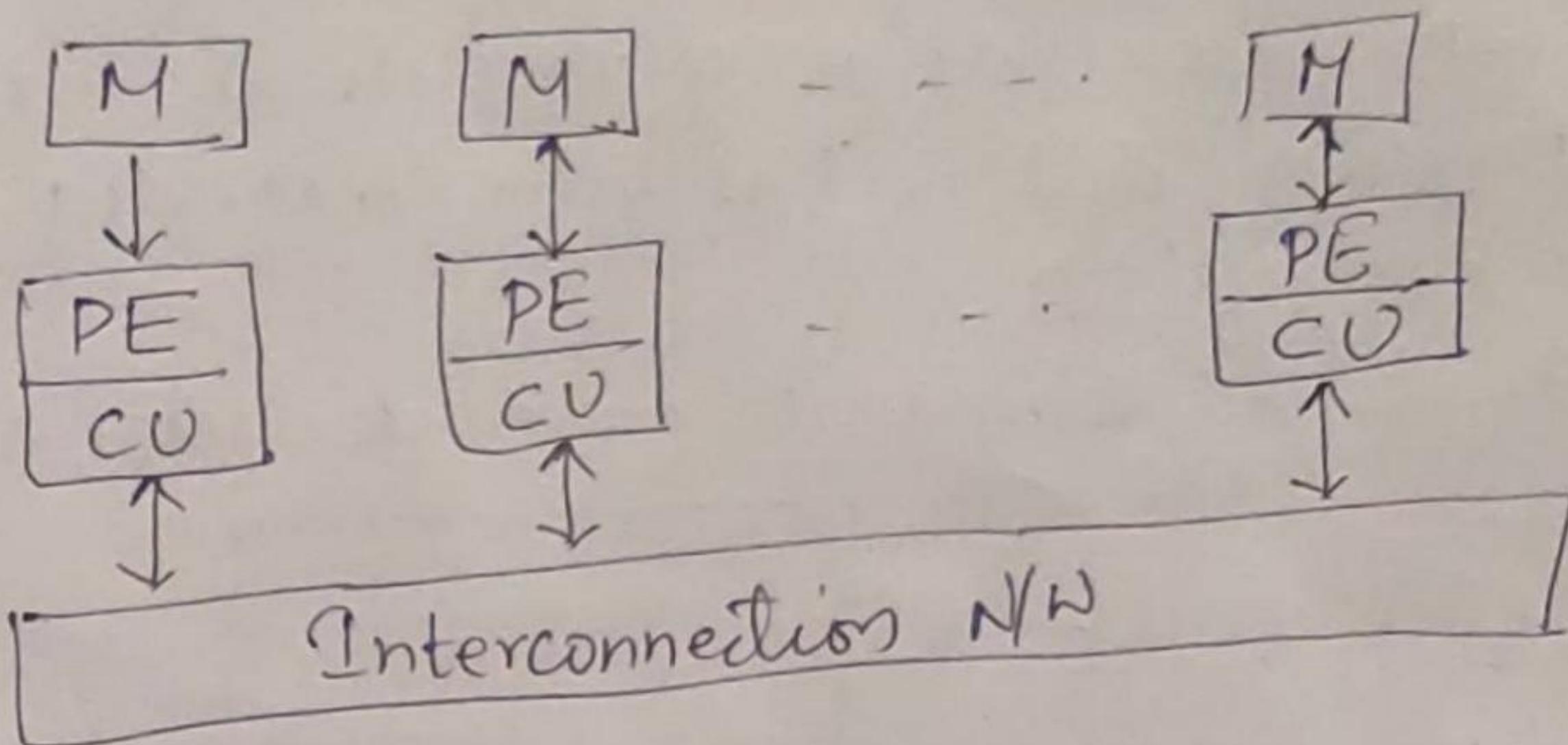
PAGE NO.

- MISD structure is only of theoretical interest since no practical system has been constructed using this organization.
- multiple processing unit on single-data stream.



## MIMD:

- All processors in a parallel computer can execute different instructions and operate on various data at the same time.
- Each <sup>processor</sup> program has a separate program and an instruction stream is generated for each program



eg: Cray T90, Cray T3E, IBM - SP2



# POORNIMA

COLLEGE OF ENGINEERING

## DETAILED LECTURE NOTES

PAGE NO 1

### Register Transfer

#### ① Register Transfer Language

- A digital computer system is an interconnection of digital HW modules such as registers, decoders, arithmetic elements and control logic.
- These digital modules are interconnected with some common data and control paths to form a complete digital system.
- Digital modules are best defined by registers. They contain and the operations that are performed on the data stored in them.
- Operations performed on the data stored in register are called Micro-operations.
- Micro-operations is an elementary operation performed on the info. stored in one or more registers.
- Result of operation may replace the previous binary info. of a register / transferred to another register.

- Example of micro-operation

shift, count, clear & load.

The internal h/w organization of a digital computer is best defined as:

(1) Set of registers and dataflow b/w them.

(2) Sequence of micro-operations performed on the data which are stored in the registers.

(3) Control path that initiates the sequence of microoperations.

- The symbolic notation used to describe the micro-operation transfers among register is called a register transfer language.

### Register Transfer :-

- Computer registers are designated by capital letters

eg: ① Registers that hold an add for the memory unit

↓  
Memory address register (MAR)

② PC (Program counter)

③ IR (Instruction register)

④ RI (Processor register)

- Individual flip-flop in an n-digit register are (0 to n-1)

Registers R.

R1
----

Individual bits

7	6	5	4	3	2	1	0
---	---	---	---	---	---	---	---

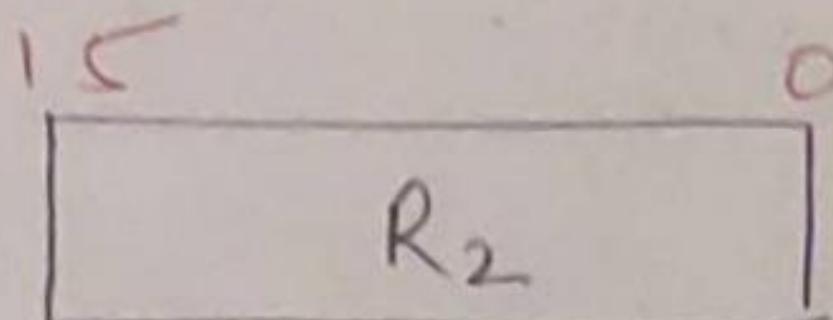


# POORNIMA

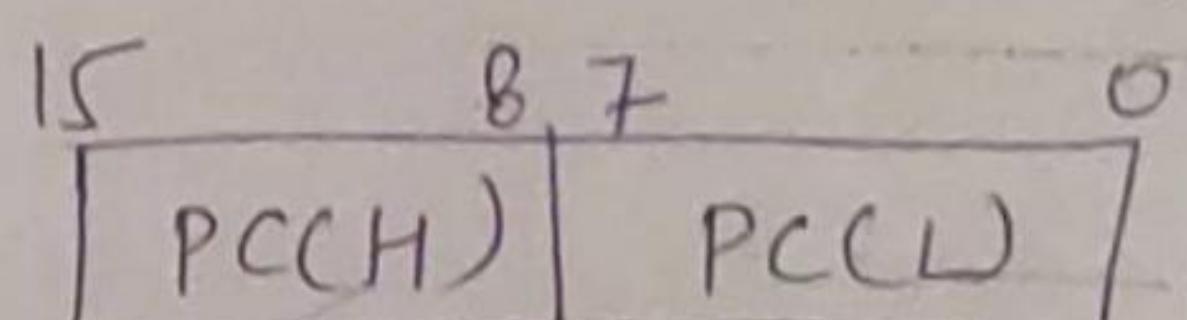
COLLEGE OF ENGINEERING

## DETAILED LECTURE NOTES

PAGE NO. 2



Numbering of bits  
(16-bit register)



High Register divided into  
byte two-parts.

Low byte

- Information transfer from one register to another is designated in system symbolic form by means of a replacement operator.

Like

$R_2 \leftarrow R_1$  denotes the transfer of the content of register  $R_1$  into register  $R_2$ .

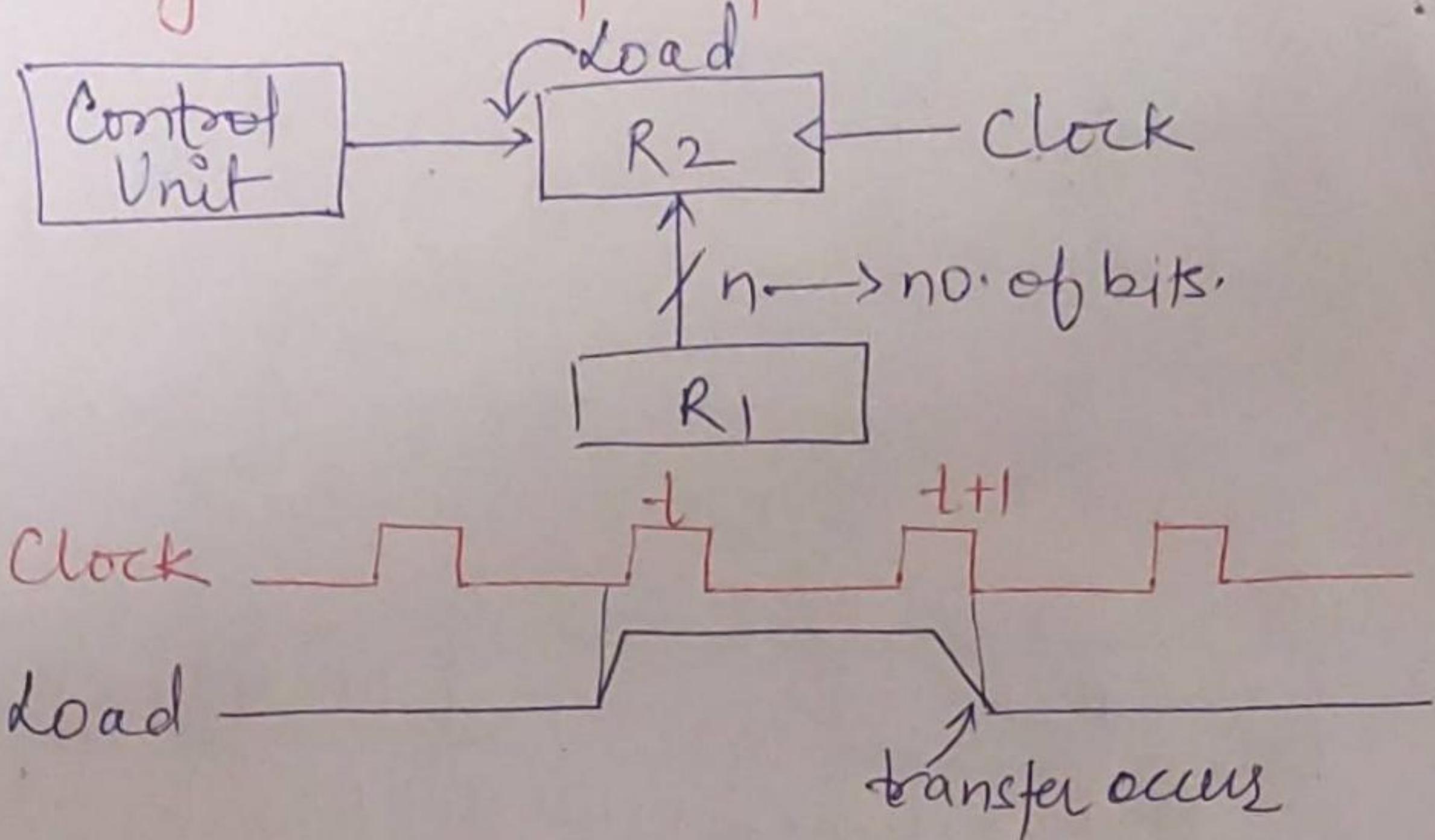
- Some time we want to transfer in a predetermined control condition. This is with the help of if-then statement.  
*Control Signal* If ( $P=1$ ) then  $(R_2 \leftarrow R_1)$

- It is better to separate the control signal from the register transfer operation by specifying a control func.

$\xrightarrow{P} R_2 \leftarrow R_1$

It means transfer operation will be performed if  $P=1$

Block Diagram: Transfers from R<sub>1</sub> to R<sub>2</sub> when P=1



② T: R<sub>2</sub> ← R<sub>1</sub>, R<sub>1</sub> ← R<sub>2</sub>

Exchange the contents of two registers during one common clock pulse that T=1

### Basic Symbols for Register Transfer

Symbol	Description	Examples
① Letter	Denotes a Register	MAR, R <sub>2</sub>
② Parentheses	Denotes a part of a register	R <sub>2(0-7)</sub> R <sub>2(L)</sub>
③ Arrow (←)	Denotes transfer of info	R <sub>2</sub> ← R <sub>1</sub>
④ Comma	Separate two micro-operations	R <sub>2</sub> ← R <sub>1</sub> , R <sub>1</sub> ← R <sub>2</sub>



# POORNIMA

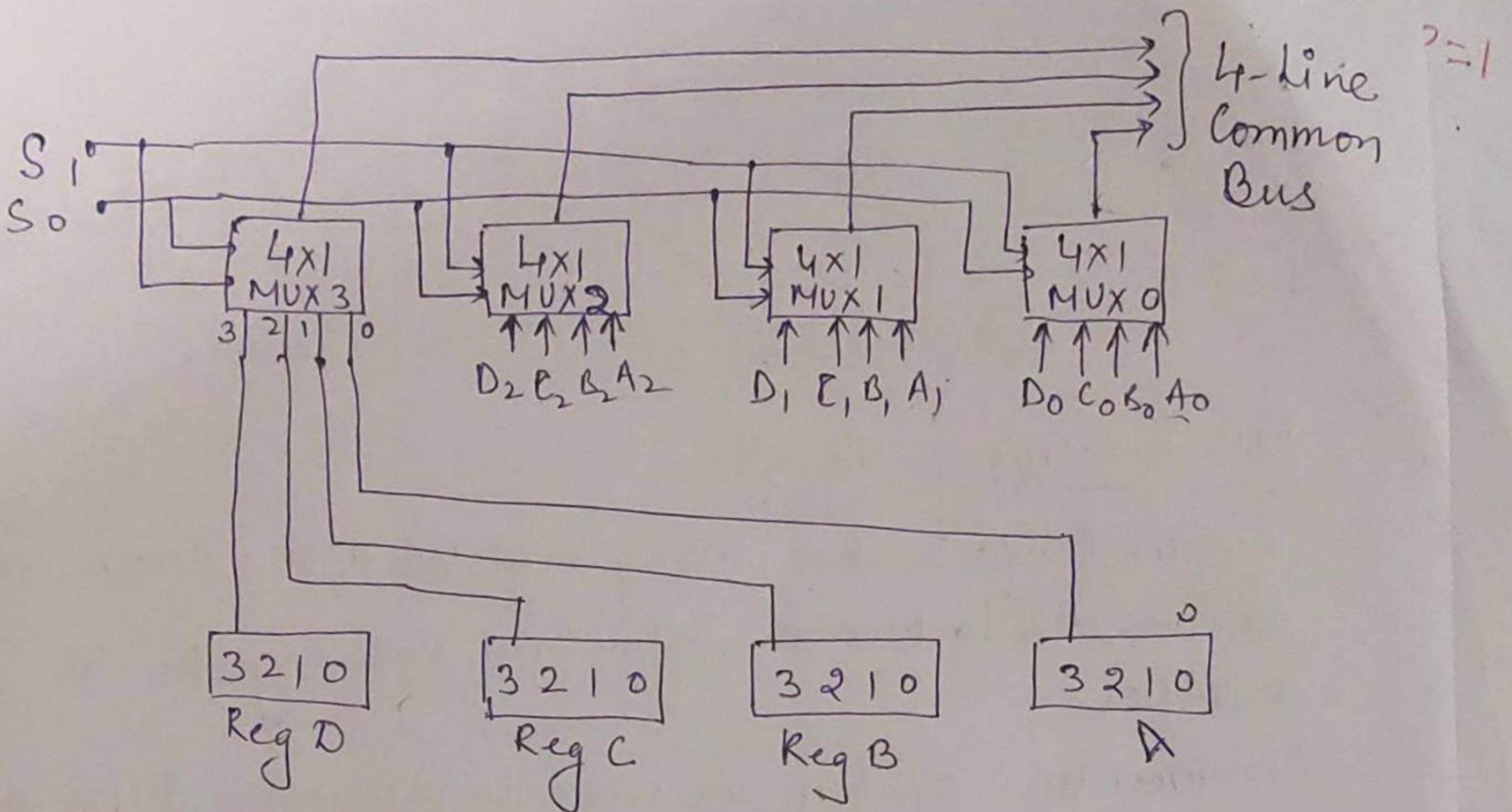
COLLEGE OF ENGINEERING

## DETAILED LECTURE NOTES

PAGE NO. ①

### Bus & Memory Transfer :-

- Digital Computer has many register and paths must be provided to transfer info. from one register to another.
- Common bus system is used to transfer info b/w registers. ( tid gwt )
- A bus structure consist of a set of common lines, one for each bit of a register, through which binary info. 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 multiplexer ( multiplexes choose the source register whose binary info is then placed on the bus ).



- Each register has four bits (3210)
- Bus consist of 4x1 multiplexers
- Two selection inputs S<sub>1</sub>, S<sub>0</sub>
- Selection line choose the four bits of one register and transfer them into the four-line common bus.

When S<sub>1</sub>, S<sub>0</sub> = 00, 0 data input for all four multiplexers are selected and applied to the op that form the bus.

So A register is selected because this register are connected to the 0 data input of the multiplexer.

S<sub>1</sub>, S<sub>0</sub> = 01 → B Register

S<sub>1</sub>, S<sub>0</sub> = 10 → C Register

S<sub>1</sub>, S<sub>0</sub> = 11 → D Register



# POORNIMA COLLEGE OF ENGINEERING DETAILED LECTURE NOTES

PAGE NO. ②

- A bus system multiplex  $k$  registers of  $n$  bits - each bit for  $n$ -line common bus

So, no. of multiplexor is  $n^{\text{with}} K \times 1$  input line

e.g.: 8 Register with 16 bits.

need 16 ~~register~~ multiplexor with 8 data input lines & 3 selection line.

- The symbolic statement for a bus transfer includes  
In the statement,

BUS  $\leftarrow C$ , RI  $\leftarrow$  BUS

Content of Reg C  
placed on the bus

Content of bus is  
loaded into Reg RI  
by activating its load  
control input

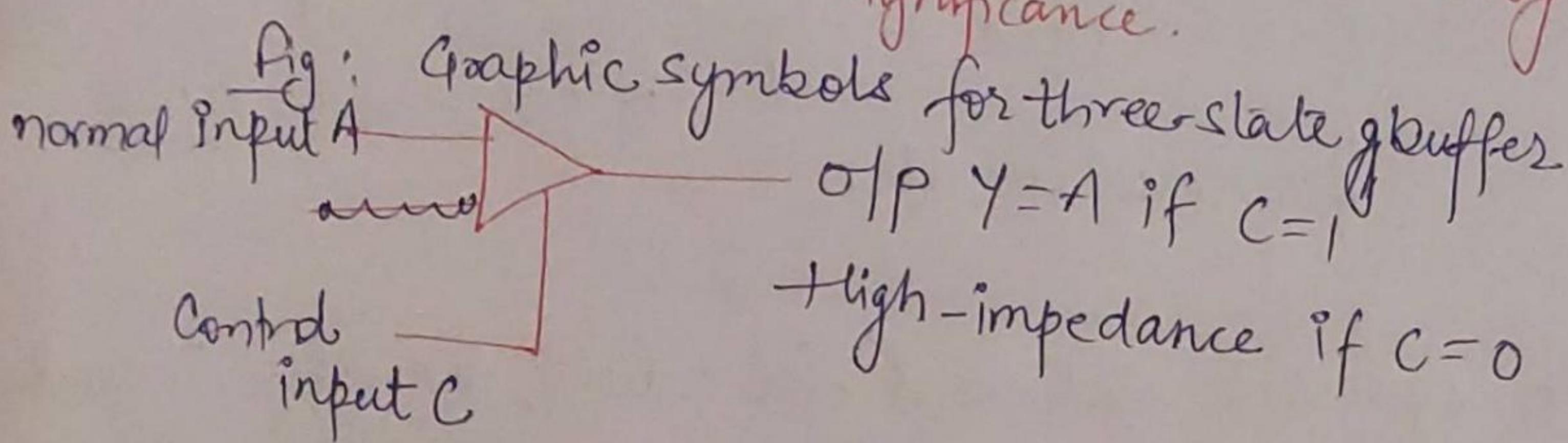
RI  $\leftarrow C$  ← direct statement

## Three-State Bus Buffers

Bus system can be constructed with three-state gates instead of multiplexers.

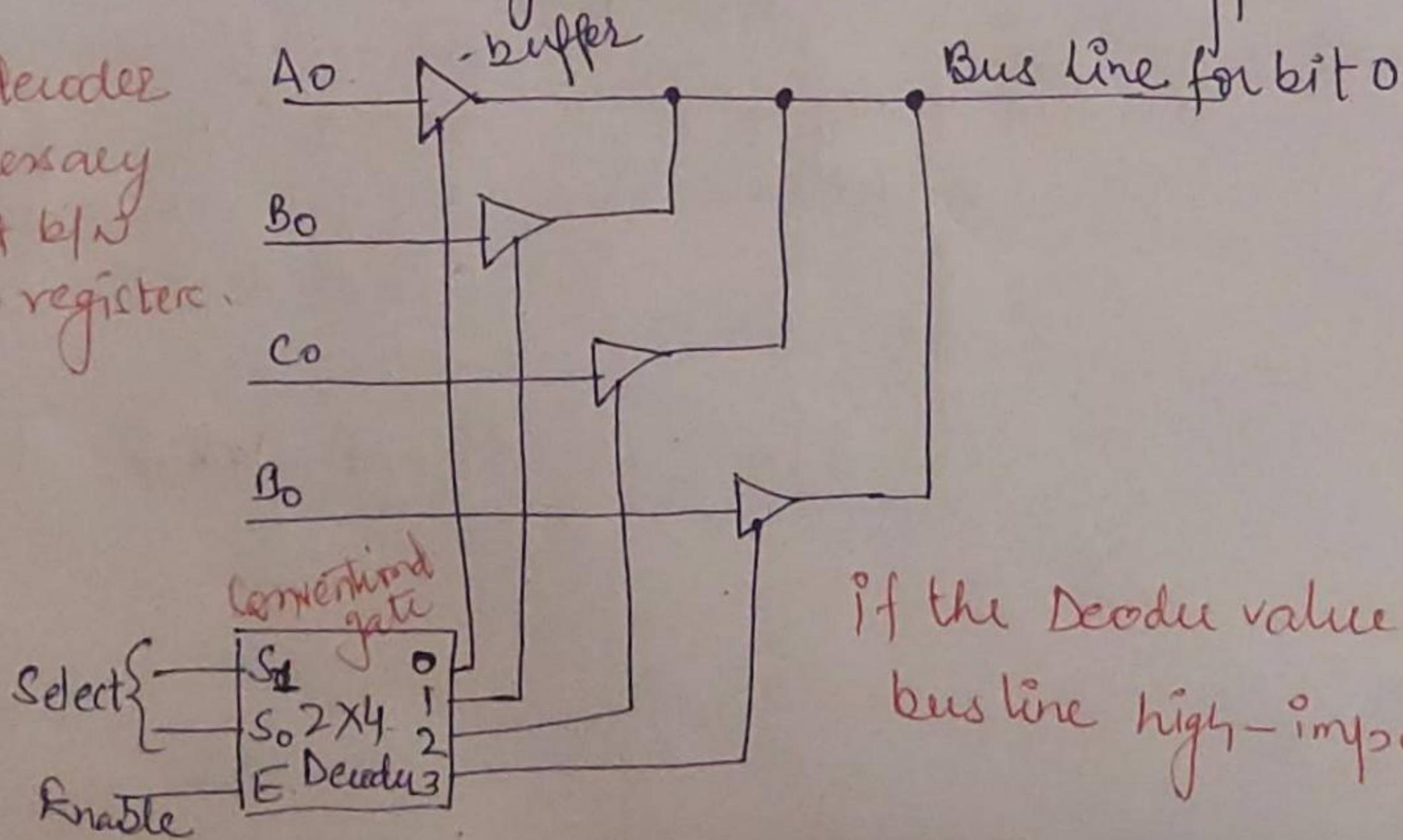
- Three-state gate is a digital circuit that exhibits three states.

↳ 1, 2 states are signals (1, 0) <sup>conventional buffer</sup>  
 ↳ 3 state is a high-impedance state  
     means open circuit  
     means o/p is disconnected  
     and does not have a logic significance.



- Construction of a bus system with three-state buffer

only 1 decoder is necessary to select b/w the two registers.





# POORNIMA

COLLEGE OF ENGINEERING

## DETAILED LECTURE NOTES

PAGE NO. ③

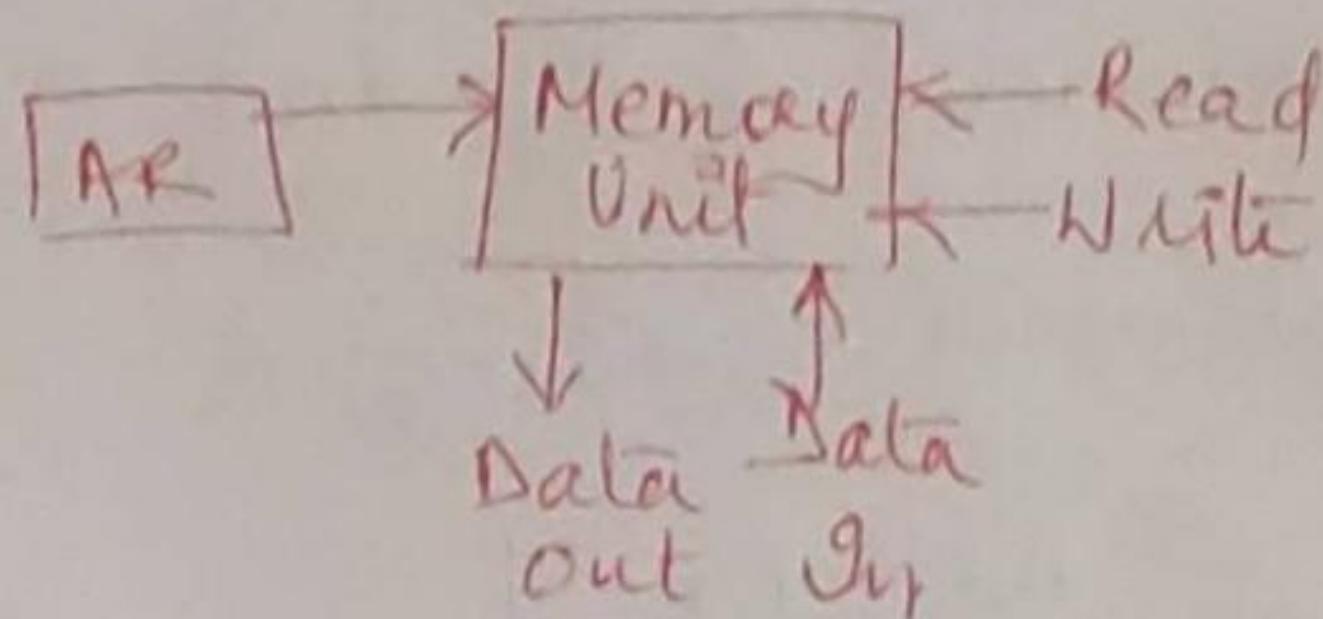
### Memory Transfer :-

- Read : Transfer of info. from a memory word to outside environment.
- Write : Transfer of new info. to be stored into the memory
- M → Memory (Symbolized)
- Memory unit that receive the address from a register is called → address register (AR)
- Data transfer to another register called the data register DR

Memory Read → Read :  $DR \leftarrow M[AR]$  Transfer info into DR from Memory word M selected by add reg AR

Memory Write → Write :  $M[AR] \leftarrow RI$

Input data are in Reg. RI and the add is in AR, Memory word (M) selected by add reg (AR)



## Arithmetic Micro-operations

Micro-operation is an elementary operation performed with the data stored in registers. It is classified into four categories:

- 1. Register Transfer microoperation
  - info not changed
  - transfer binary info from one register to another.
- 2. Arithmetic microoperation
  - change info
  - perform arithmetic operation on numeric data stored in register
- 3. Logic microoperation
  - info
  - perform bit manipulation operation on non-numeric data stored in register
- 4. Shift microoperation
  - register
  - perform shift operation on data stored in registers.

- Arithmetic microoperations are addition, subtraction, increment, decrement and shift.

$$\text{Addition} \quad R_3 \leftarrow R_1 + R_2$$

Content of  $R_1$  are added to the content of Register  $R_2$  and transferred to register  $R_3$ .

$$\text{Subtraction} \quad R_3 \leftarrow R_1 + \overline{R_2} + 1$$

$\overline{R_2}$  is 1's complement of  $R_2$  and  $+1$  produces the  $R_2$ 's complement of  $R_2$

$$R_1 - R_2 = R_1 + \overline{R_2} + 1$$



# POORNIMA

COLLEGE OF ENGINEERING

## DETAILED LECTURE NOTES

PAGE NO. 4

$$R_3 \leftarrow R_1 + R_2 \quad - \text{ Content of } R_1 + R_2 - \text{ transfer } R_3$$

$$R_3 \leftarrow R_1 - R_2 \quad \text{Subtraction}$$

$$R_2 \leftarrow \bar{R}_2 \quad \text{is Complement}$$

$$R_2 \leftarrow \bar{R}_2 + 1 \quad 2's \quad //$$

$$R_3 \leftarrow R_1 + \bar{R}_2 + 1 \quad \text{Subtraction}$$

$R_1 \leftarrow R_1 + 1$  } increment & decrement microoperations  
 $R_1 \leftarrow R_1 - 1$  } are implemented with a combinational circuit or binary up down counter

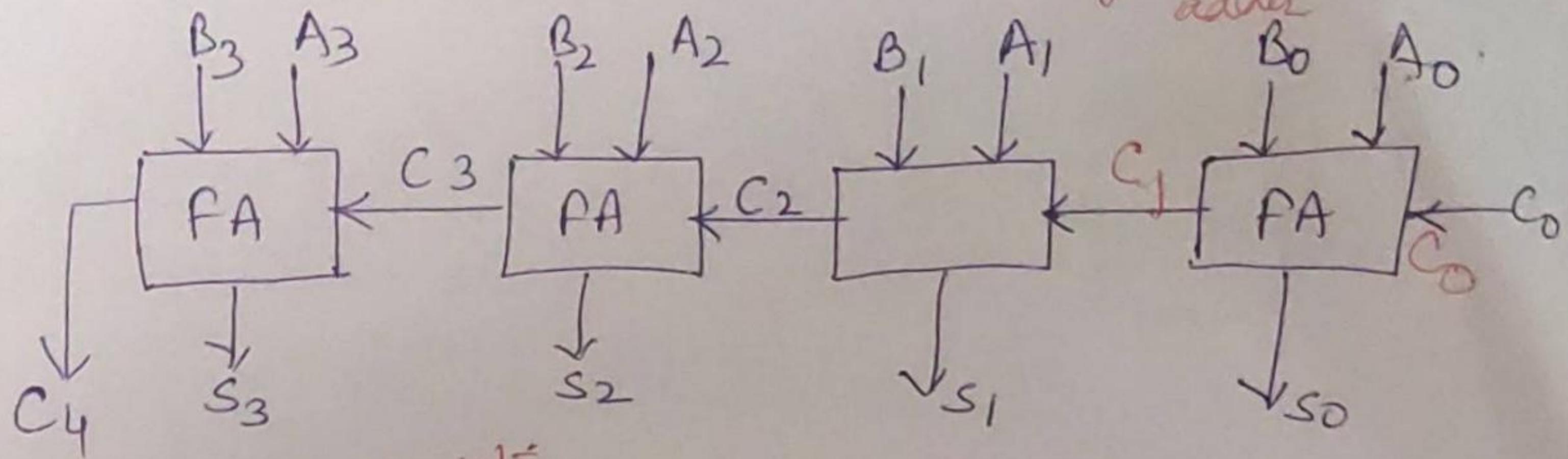
Multiplication & Division are implemented by means of a Binary Adder: <sup>use combinational circuit</sup>

For arithmetic addition digital computer need registers ~~&~~ and digital circuit  $\rightarrow$  full adder.  
(to hold data)

- Binary adder:  $\rightarrow$  digital circuit that generates the arithmetic sum of two binary no. of any length is called a binary adder. <sup>it consists of full adders connected in cascade.</sup>

full adder  $\rightarrow$  sum of two bits + previous carry

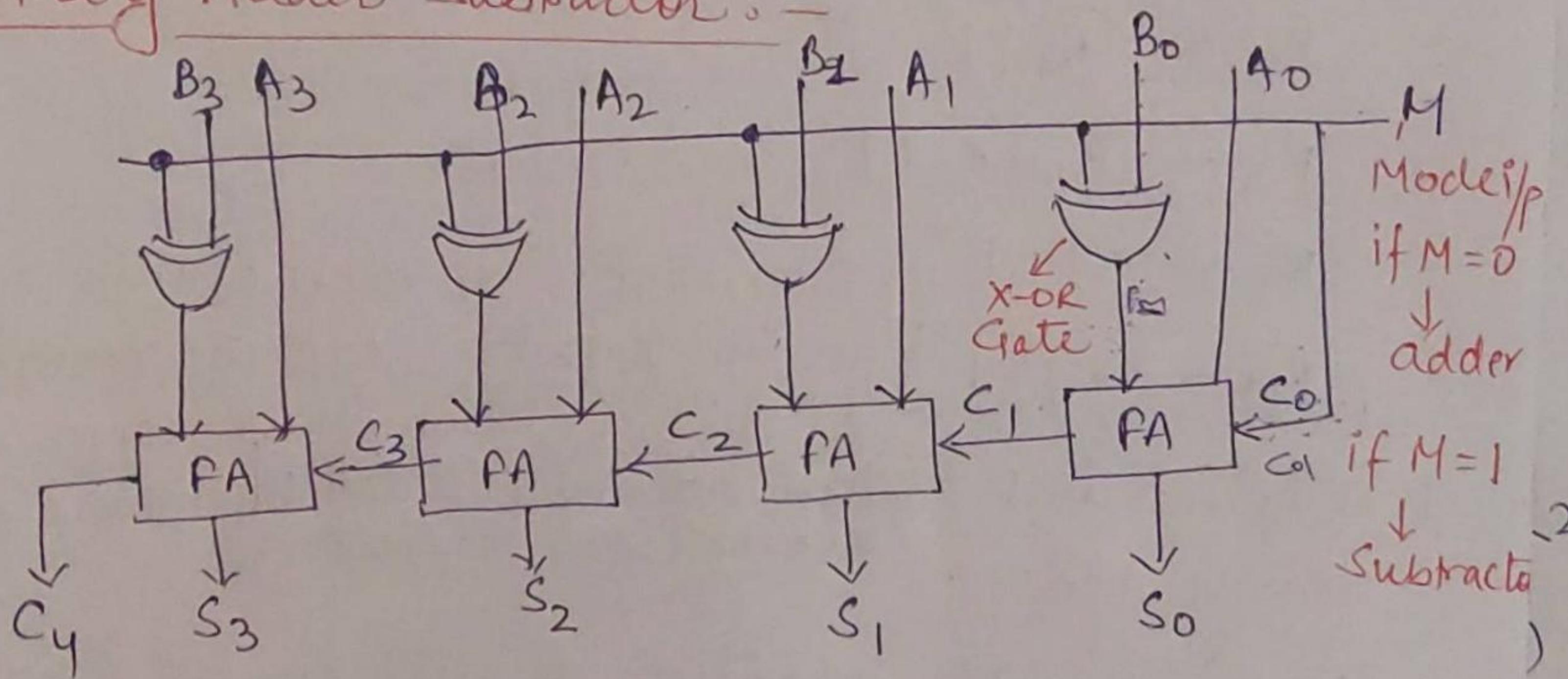
Carry from one full adder to next full adder



A stored in R<sub>1</sub> Register  
B stored in R<sub>2</sub> Register 4-bit binary adder

4-bit binary adder requires 4 full-adders.

Binary Adder-Subtractor :-



$$\text{When } M=0 \quad B \oplus 0 = B$$

$$A + B$$

$$\text{When } M=1 \quad B \oplus 1 = B' \quad \& \quad C_0 = 1$$

$$A - B$$



# POORNIMA

COLLEGE OF ENGINEERING

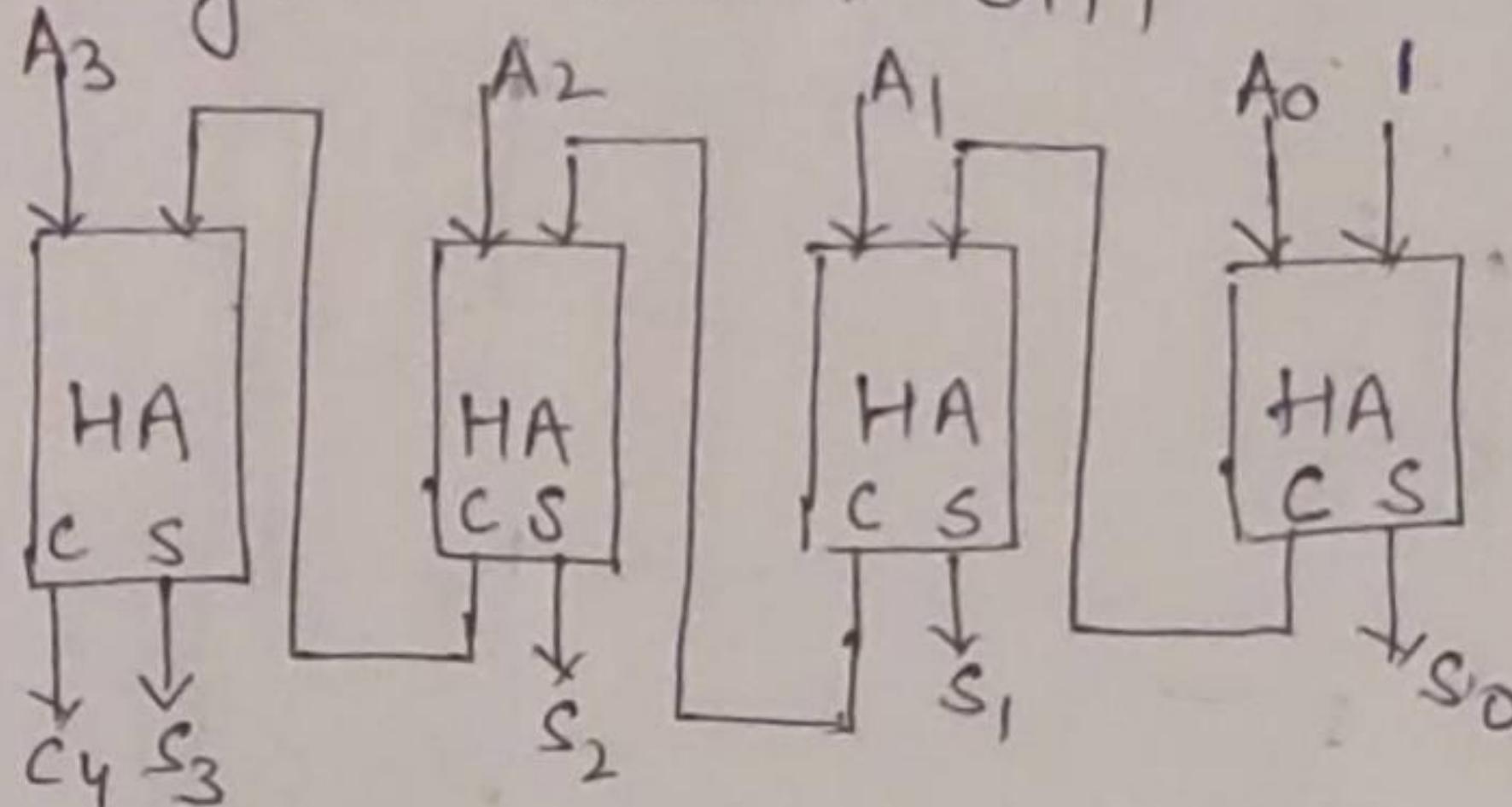
## DETAILED LECTURE NOTES

PAGE NO 5

### Binary Incrementer:-

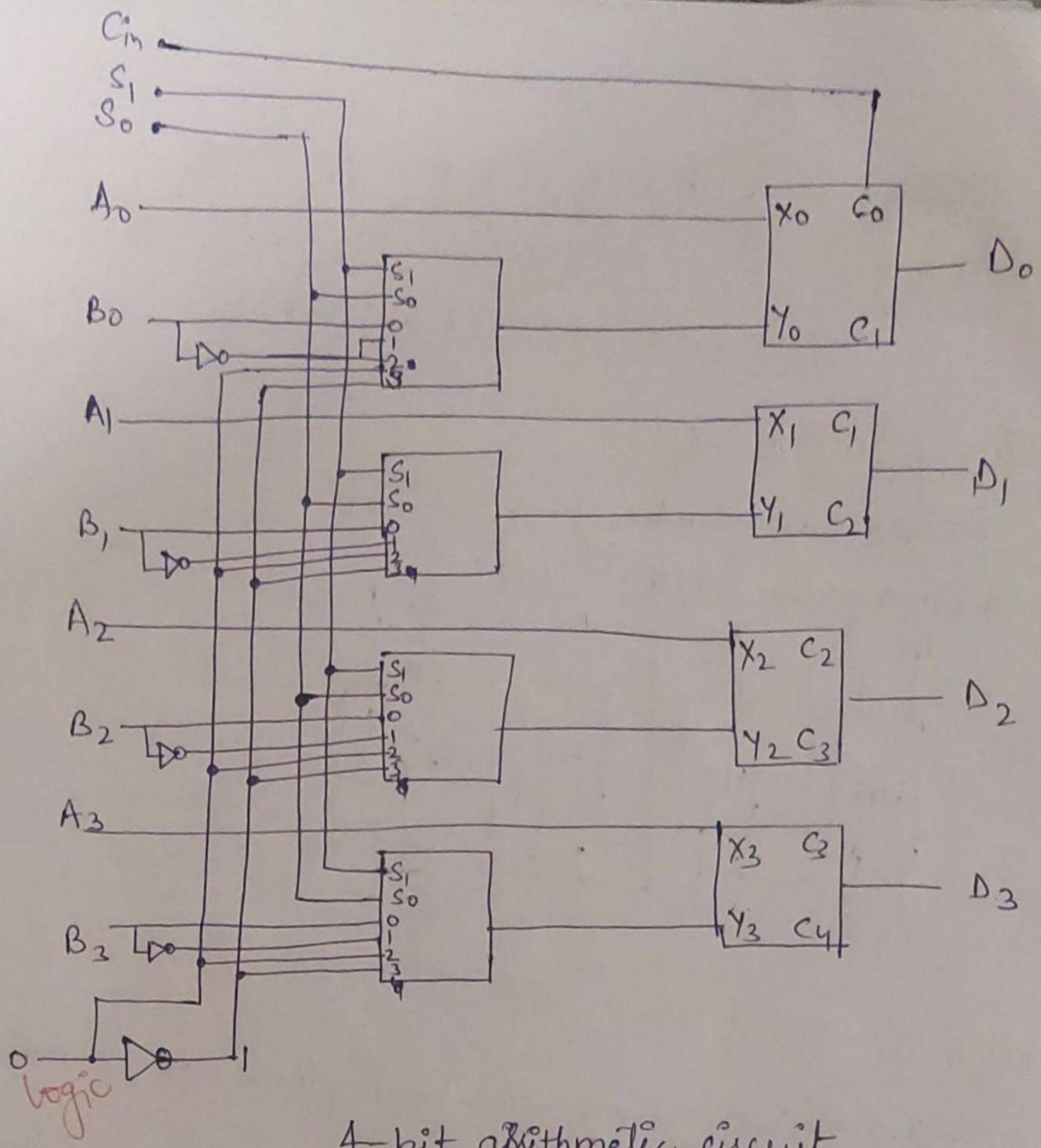
add one to a number in a register

e.g.: 4 bit register  $0110 \xrightarrow{+1} 0111$



### Arithmetic Circuit:-

- Arithmetic microoperations can be implemented in one composite arithmetic circuit.
- Basic component of an arithmetic circuit is the parallel adder shown in diagram (next page)
- Two 4-bit inputs A and B
- 4-bit output D.
- A go directly to the X inputs of the binary adder
- B are connected to the data points of the multiplexer.



4-bit arithmetic circuit

- four multiplexers are controlled by two selection inputs,  $S_1$  &  $S_0$

$$D = A + Y + C_{in}$$

2 There are 2 4-bit inputs  $A$  &  $B$  — 4 bit  $D$  off  
 $\rightarrow A, B, \text{logic}_0, \text{logic}_1$



# POORNIMA

COLLEGE OF ENGINEERING

## DETAILED LECTURE NOTES

PAGE NO. 6

Arithmetic circuit function table

Select			Input Y	O/P	Microoperation
S <sub>1</sub>	S <sub>0</sub>	C <sub>in</sub>		D = A + Y + C <sub>in</sub>	
0	0	0	B	D = A + B	→ Add
0	0	1	B	D = A + B + 1	→ Add with carry
0	1	0	$\bar{B}$	D = A + $\bar{B}$	→ Subtract with borrow
0	1	1	$\bar{B}$	D = A + $\bar{B}$ + 1	→ Subtract
1	0	0	0	D = A	→ Transfer A
1	0	1	0	D = A + 1	→ Increment A
1	1	0	1	D = A - 1	→ Decrement A
1	1	1	1	D = A	→ Transfer A

## logic Micro-operations

- logic microoperations specify binary operations for strings of bits stored in registers.

P:  $R_1 \leftarrow R_1 \oplus R_2$  { content of two registers  
 $R_1 + R_2$  is }

Let  $R_1 = 1010$

$R_2 = 1100$

if P=1       $\begin{array}{r} 0110 \\ + 1100 \\ \hline 0110 = R_1 \end{array}$

- logic microoperations OR — V  
AND — ^  
complement —  $\bar{A}$  { bar }

*plus + when X in microoperator  
OR when in control boolean*

P+Q:  $R_1 \leftarrow R_2 + R_3, R_4 \leftarrow R_5 \vee R_6$

OR operation / in control fun.

+ b/w R2 & R3 → specifies add micro operation

∨ → or microoperation b/w register

## List of logic Micro-operations

- 16 different logic operations that can be performed with two binary variable.



# POORNIMA

COLLEGE OF ENGINEERING

## DETAILED LECTURE NOTES

PAGE NO. (7)

Truth Table for 16 func of two Variable

x	y	$f_0$	$f_1$	$f_2$	$f_3$	$f_4$	$f_5$	$f_6$	$f_7$	$f_8$	$f_9$	$f_{10}$	$f_{11}$	$f_{12}$	$f_{13}$	$f_{14}$	$f_{15}$
0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	1	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	1	1	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	1	1

Sixteen logic Microoperation

Boolean func

Micro operation

Name

$$f_0 = 0$$

$$f \leftarrow 0$$

Clear

$$f_1 = xy$$

$$f \leftarrow A \wedge B$$

AND

$$f_2 = x'y'$$

$$f \leftarrow \bar{A} \wedge \bar{B}$$

$$f_3 = x$$

$$f \leftarrow A$$

Transfer A

$$f_4 = x'y$$

$$f \leftarrow \bar{A} \wedge B$$

$$f_5 = y$$

$$f \leftarrow B$$

Transfer B

$$f_6 = x \oplus y$$

$$f \leftarrow A \oplus B$$

Exclusive-OR

$$f_7 = x + y$$

$$f \leftarrow A \vee B$$

OR

$$f_8 = (x+y)'$$

$$f \leftarrow \overline{A \vee B}$$

NOR

$$f_9 = (x \oplus y)'$$

$$f \leftarrow \overline{A \oplus B}$$

Exclusive-NOR

$$f_{10} = y'$$

$$f \leftarrow \bar{B}$$

Complement B

$$f_{11} = x + y'$$

$$f \leftarrow A \vee \bar{B}$$

$$f_{12} = x'$$

$$f \leftarrow \bar{A}$$

Complement A

$$f_{13} = x' + y$$

$$f \leftarrow \bar{A} \vee B$$

$$f_{14} = (xy)'$$

$$f \leftarrow \overline{A \wedge B}$$

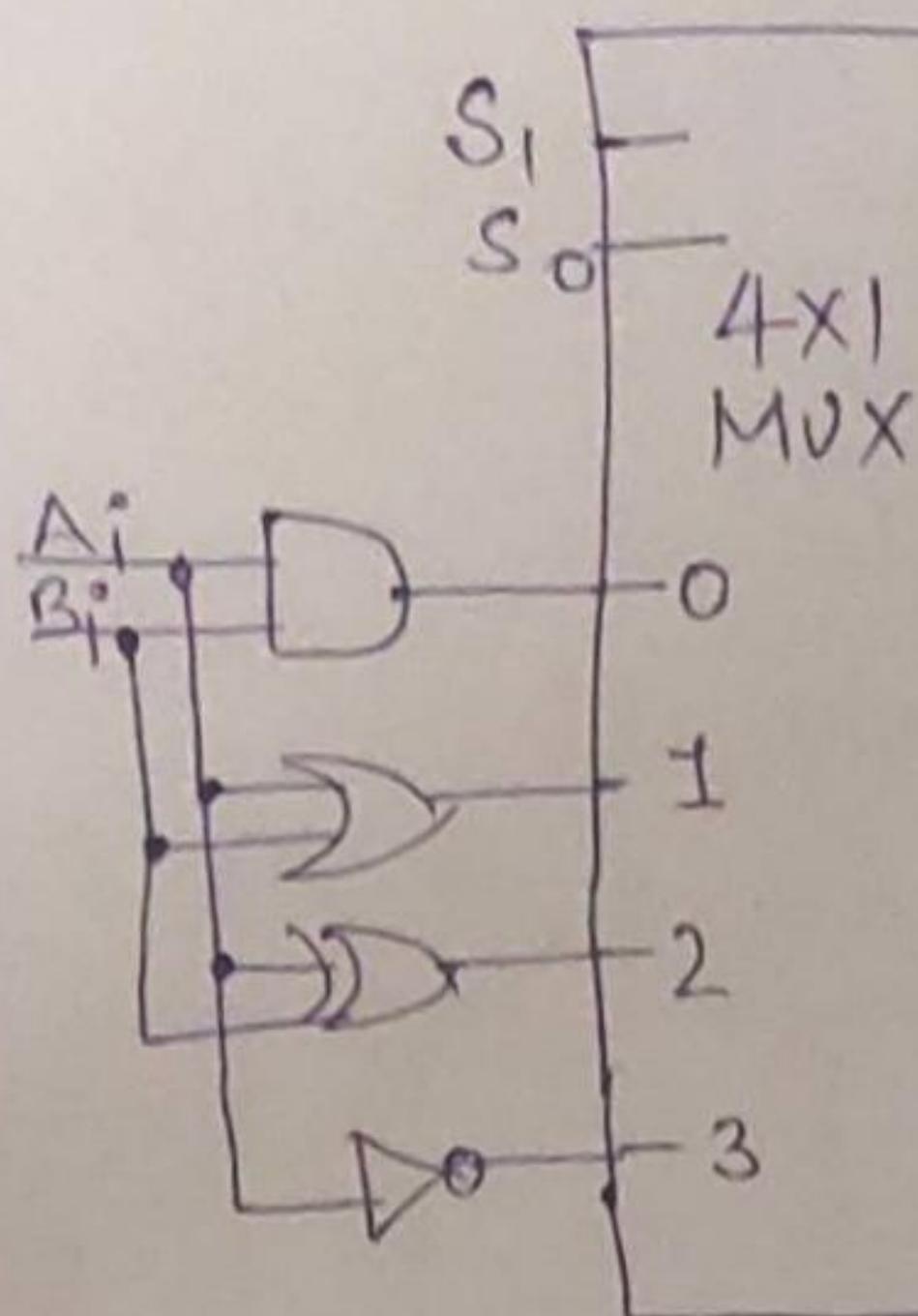
NAND

$$f_{15} = 1$$

$$f \leftarrow \text{all } 1's$$

Set all 1's

## Hardware Implementation:-



4 gated 1 Multiplexer

$S_1$	$S_0$	O/P	Operation
0	0	$E = A \wedge B$	AND
0	1	$E = A \vee B$	OR
1	0	$E = A \oplus B$	XOR
1	1	$E = \bar{A}$	Complement

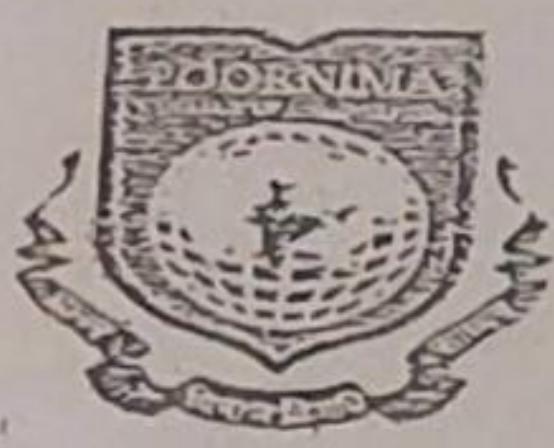
- Most computer use only four - AND, OR, XOR, complement.

## Applications:-

logic microoperations are very useful for manipulating individual bits or a portion of a word stored in a register. They can

- change bit values
  - delete a group of bits
  - insert new bit
- } Into a register.

- A is processor register
- B logic operand extracted from memory and placed in register B.



# POORNIMA

COLLEGE OF ENGINEERING

## DETAILED LECTURE NOTES

PAGE NO. ....

Selective-set operation



# POORNIMA

COLLEGE OF ENGINEERING

## DETAILED LECTURE NOTES

PAGE NO. 01

### Shift Micro-operations :-

- Shift micro-operations are used for serial transfers of data.
- They are also used in conjunction with arithmetic, logic, and other data processing operations.
  - Content of register can be shifted left or right.
  - Shift-left operation the serial i/p transfers a bit into the rightmost position.
  - Shift-right operation the serial i/p transfers a bit into the leftmost position.
- There are three types of shift:
  - (1) logical
  - (2) Circular
  - (3) Arithmetic

① logical :- It transfer 0 through the serial i/p.

Shl → shift-left

Shr → shift-right

R1 ← Shl K1

R2 ← Shr R2

## ④ Circular shift: - or rotate operation

It circulates the bits of the register around the two ends without loss of info.

csl → circular shift left

csr → circular shift right.

## ⑤ Arithmetic shift: -

It shifts a signed binary no. to the left or right.

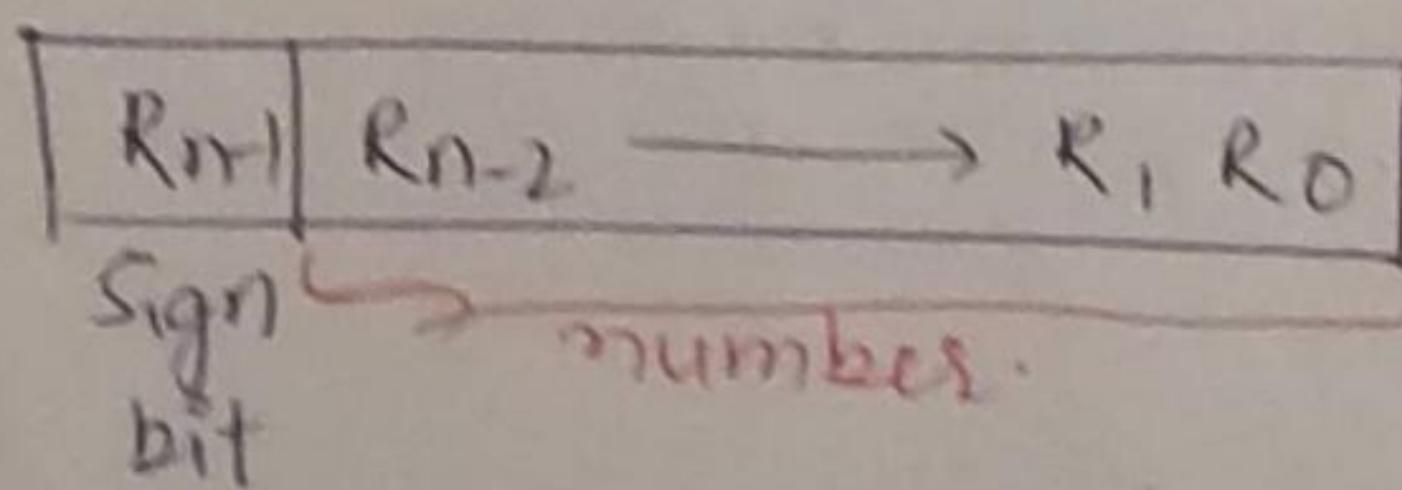
- It multiplies a signed binary no. by  $2^{\text{shift-left}}$ .

- It divides shift-right dividee the no. by  $2^{\text{shift-right}}$ .

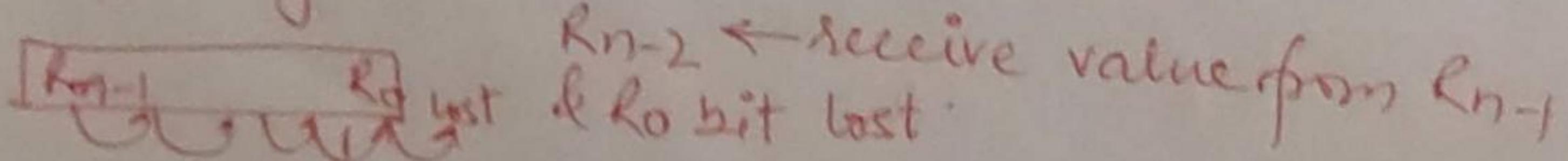
- Arithmetic shifts must leave the sign bit unchanged because the sign of the no. remains the same when it is multiplied or divided by 2.

$R \leftarrow \text{ashl } R$  (arithmetic shift left)

$R \leftarrow \text{ashr } R$  (arithmetic shift right)



Shift-right →  $R_{n-1}$  remain same



Shift-left: - insert 0 at  $R_0$   
shift bits to left

so,  $R_{n-1} - \text{lost} \leftarrow$  and get value from  $R_{n-2}$



# POORNIMA

COLLEGE OF ENGINEERING

## DETAILED LECTURE NOTES

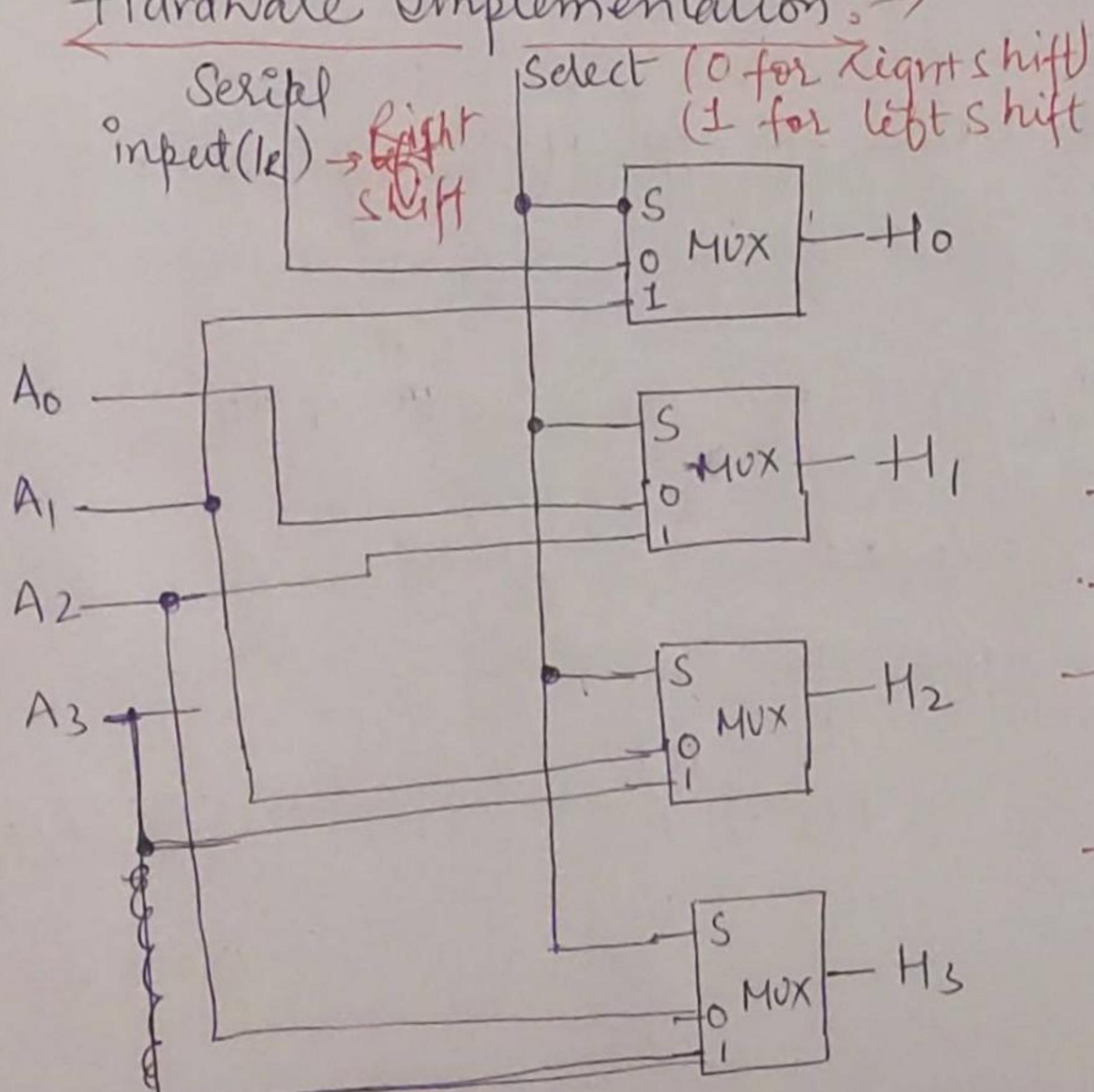
PAGE NO. 2

$$V_S = R_{n-1} \oplus R_{n-2}$$

If  $V_S = 0$  - no overflow

but if  $V_S = 1$  - overflow

Hardware Implementation:



function table

Select	O/P
S	$H_0 H_1 H_2 H_3$
0	$I_R A_0 A_1 A_2$
1	$A_1 A_2 A_3 I_L$

4-bit combinational circuit  
shifter

Serial left shift  
input ( $I_L$ )

- Information can be transferred to the register in parallel and then shifted to the right or left.
- Clocks <sup>Pulse</sup> are required for the loading of data into register and for the initiate of shift.
- When  $S=0$   $I_R$  selected  
 $S=1$   $I_L$  selected
- A shifter with  $n$  data inputs and  $o/p$  require  $n$  multiplexer.

Q.: What is the value of  $O/P + H$  in fig (prev page) if input  $A$  is 1001

$$S = 1$$

$$I_R = 1 \quad I_L = 0$$

$$\begin{matrix} 1 & 0 & 0 & 1 \\ A_3 & A_2 & A_1 & A_0 \end{matrix}$$

$$S = 1, \quad I_L \text{ selected}$$

$$S = 1. \quad \therefore O/P \quad H_0 = A_1 = 0$$

$$H_1 = A_2 = 0$$

$$H_2 = A_3 = 1$$

$$H_3 = S_L = 0$$

$\boxed{0110}$

$\boxed{0010}$



# POORNIMA

COLLEGE OF ENGINEERING

## DETAILED LECTURE NOTES

PAGE NO. ....

### Arithmetic logic Shift Unit:-

- Use individual register for micro-operations directly.
- In computer system, no. of storage registers connected to a common operational unit called arithmetic logic unit (ALU).
- ALU performs an operation and the result of the operation is then transferred to a destination register.
- ALU is combinational circuit so that the entire register transfer operation from the source register through the ALU and into the destination register can be performed during one clock pulse period.
- Arithmetic, logic and shift circuits are combined into ALU.
- Input  $A_i$  &  $B_j$  for both arithmetic and logic units.

Arithmetic op  $\rightarrow$   $E_i$  &

logical op  $\rightarrow$   $H_i$

## Operations select

$S_3$	$S_2$	$S_1$	$S_0$	$Cin$	Operation	Function
0	0	0	0	0	$F = A$	Transfer A
0	0	0	0	1	$F = A + 1$	Increment A
0	0	0	1	0	$F = A + B$	Add
0	0	0	1	1	$F = A + B + 1$	Add with carry
0	0	1	0	0	$F = A + \bar{B}$	Sub with borrow
0	0	1	0	1	$F = A + \bar{B} + 1$	Sub
0	0	1	1	0	$F = A - 1$	Decrement A
0	0	1	1	1	$F = A$	Transfer A
0	1	0	0	X	$F = A \wedge B$	AND
0	1	0	1	X	$F = A \vee B$	OR
0	1	1	0	X	$F = A \oplus B$	Logic XOR
0	1	1	1	X	$F = \bar{A}$	Complement A
1	0	X	X	X	$F = Shr\ A$	Shift sign bit A into F
1	1	X	X	X	$F = Shl\ A$	Shift left A into F



# POORNIMA

COLLEGE OF ENGINEERING

## DETAILED LECTURE NOTES

14

PAGE NO.

### Basic Computer Organization & Design

#### Instruction Codes:-

- The organization of computer is defined by its internal registers, timing & control structure and set of instructions.
- Internal organization of a digital system is defined by the sequence of microoperations it performs on data stored in the registers.
- Program is a set of instructions that specify the operations, operands, and sequence by which processing occurs.
- Computer instruction is a binary code that specifies a sequence of micro-operation for the computer.
- Instruction codes + data are stored in memory.
- Computer read each instruction from memory and places it in a control register.
- The control then interprets the binary code of the instruction and proceeds to execute it by issuing a sequence of micro-operations.

- An instruction code is a group of bits that instruct the computer to perform a specific operation.
- Instruction code is its operation part.
- Operation code of an instruction is a group of bits that define such operations as add, subtract, multiply, shift and complement.
- The no. of bits required for the operation code of an instruction depends on the total no. of operations available in the computer.
- The operation code must consist of at least  $n$  bits for a given  $2^n$  ( $less$ ) distinct operations.
- eg: ADD operation
  - ↳ Operation code → six bits
  - ↳ 110010
- When this Op. code → decoded in → control unit.  
& given signal  
to read operand  
from memory and  
add in register.

## Stored Program Organization

- Computer organize have
  - (1) one processor register
  - (2) instruction code format with two parts.
    - ① Operation
    - ② address.
- Memory add tells the control where to find an operand in memory

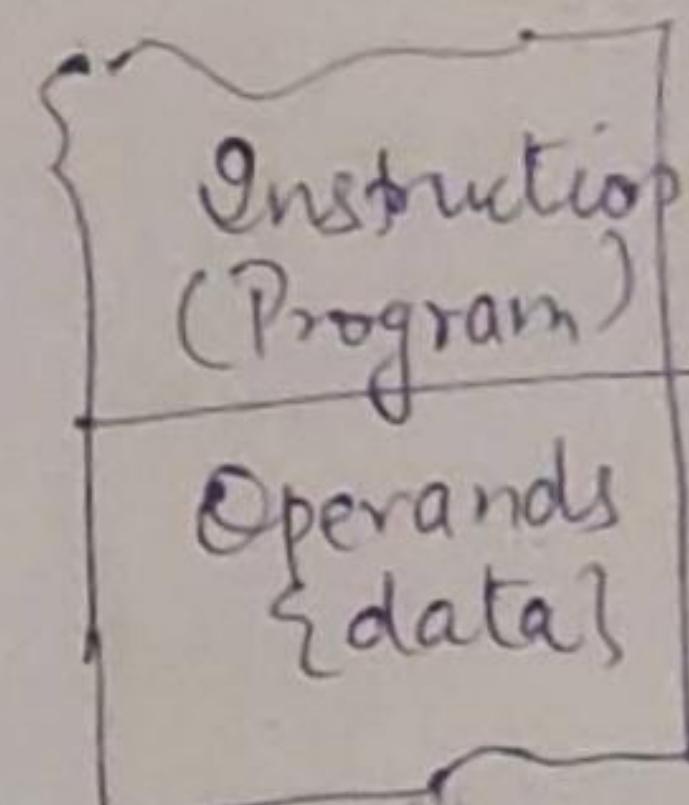
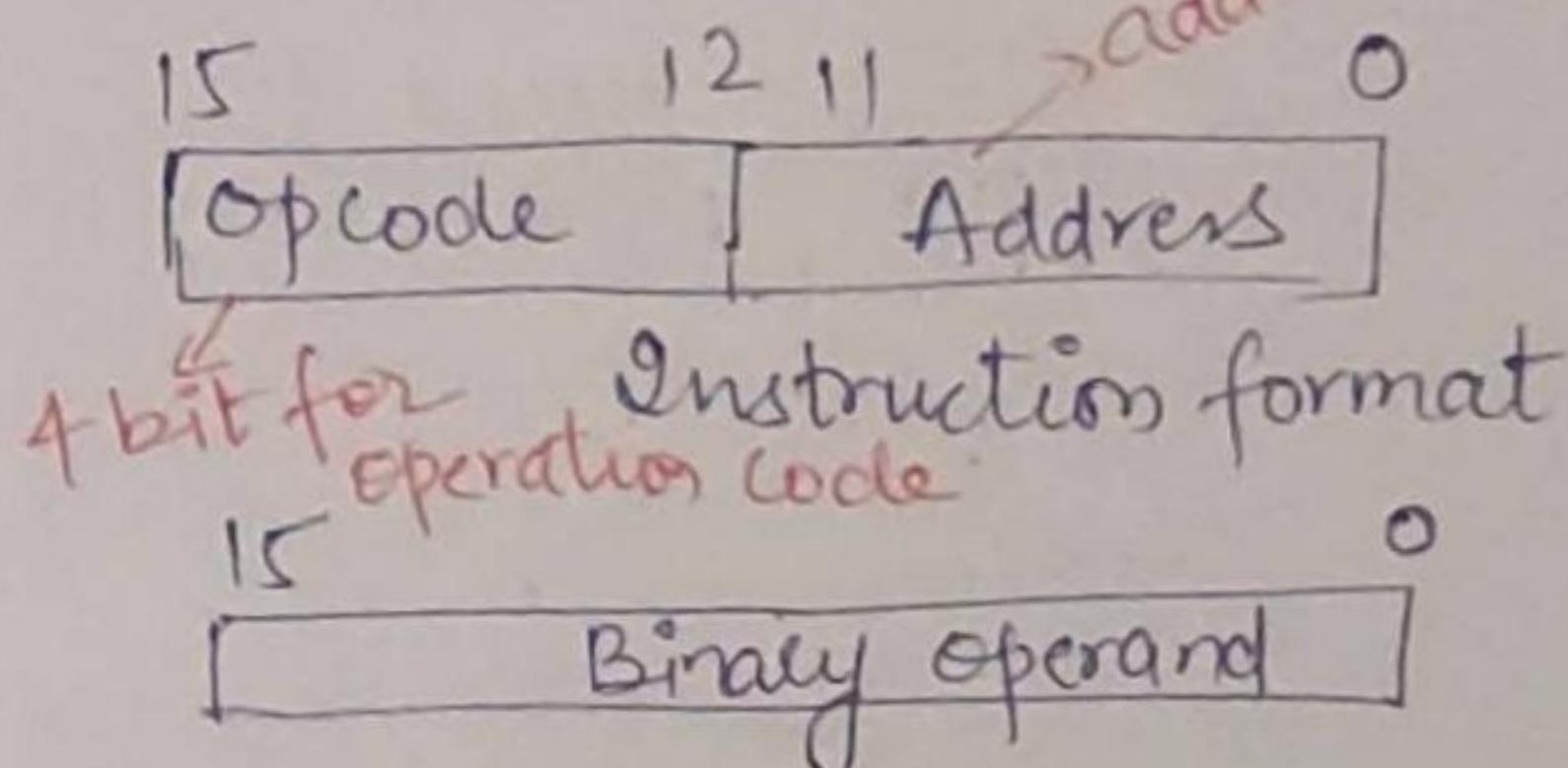


# POORNIMA

COLLEGE OF ENGINEERING

## DETAILED LECTURE NOTES

PAGE NO.



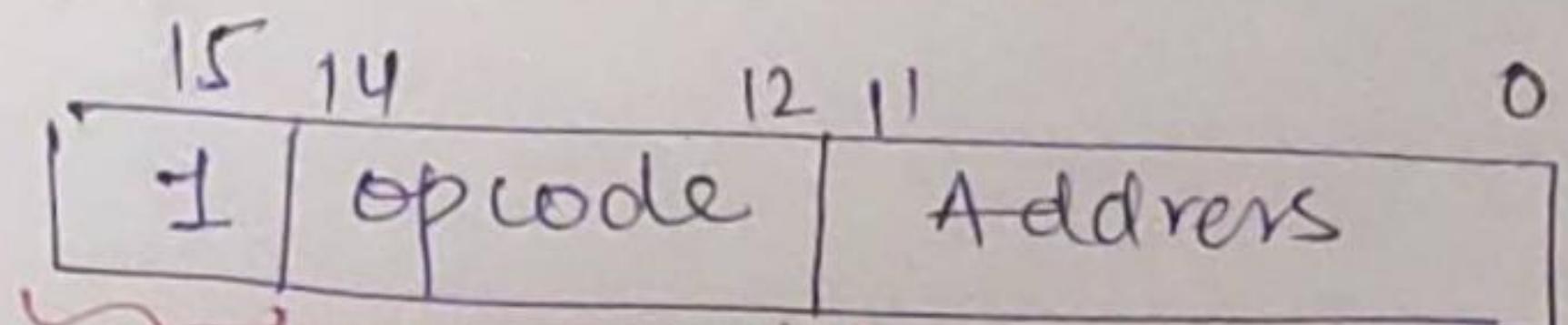
- Operation is performed with the memory operand and the content of AC.
- When do not need operand from memory  $\rightarrow$  memory add left empty.

Processor Register  
(AC - Accumulator)

### Indirect Address:

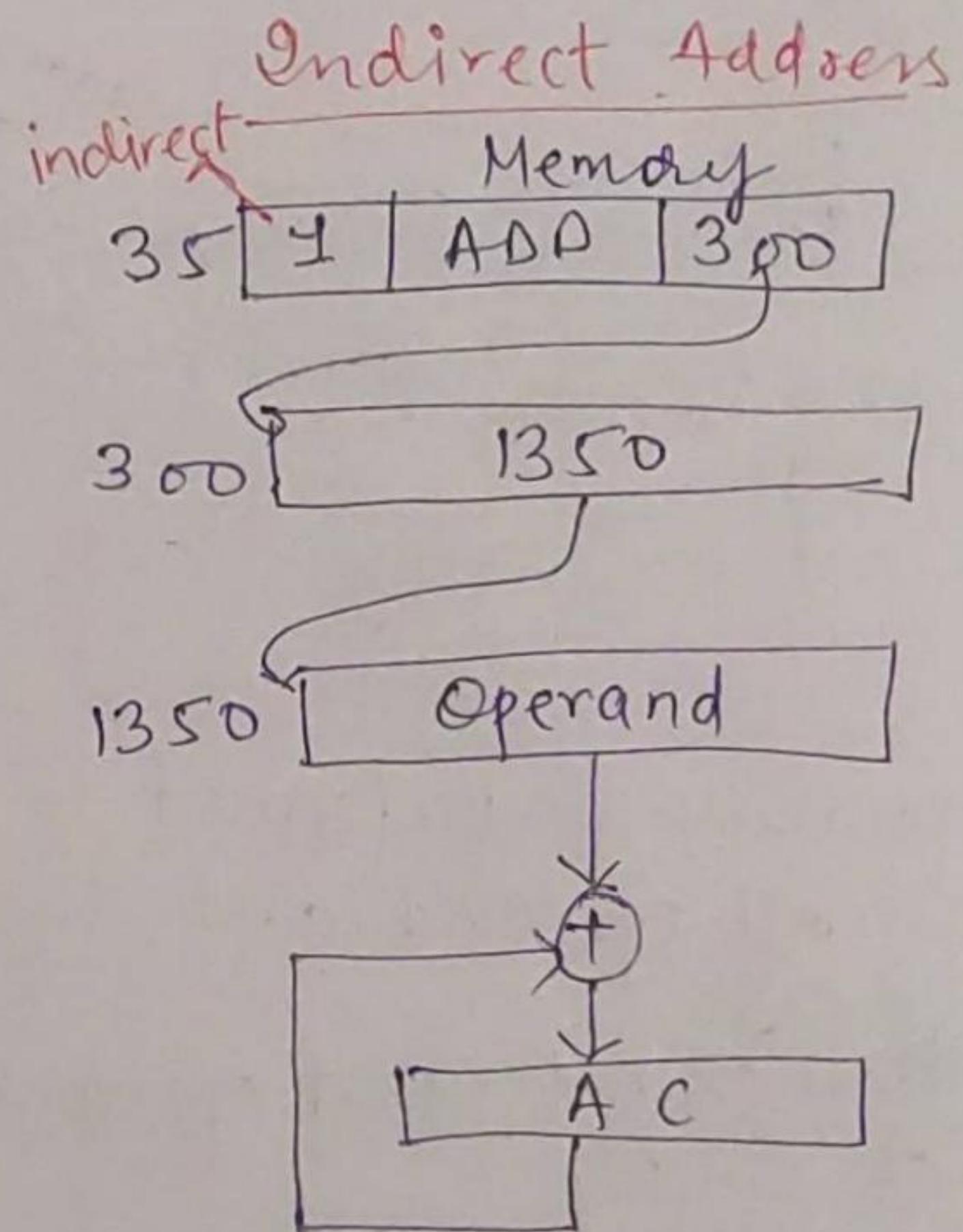
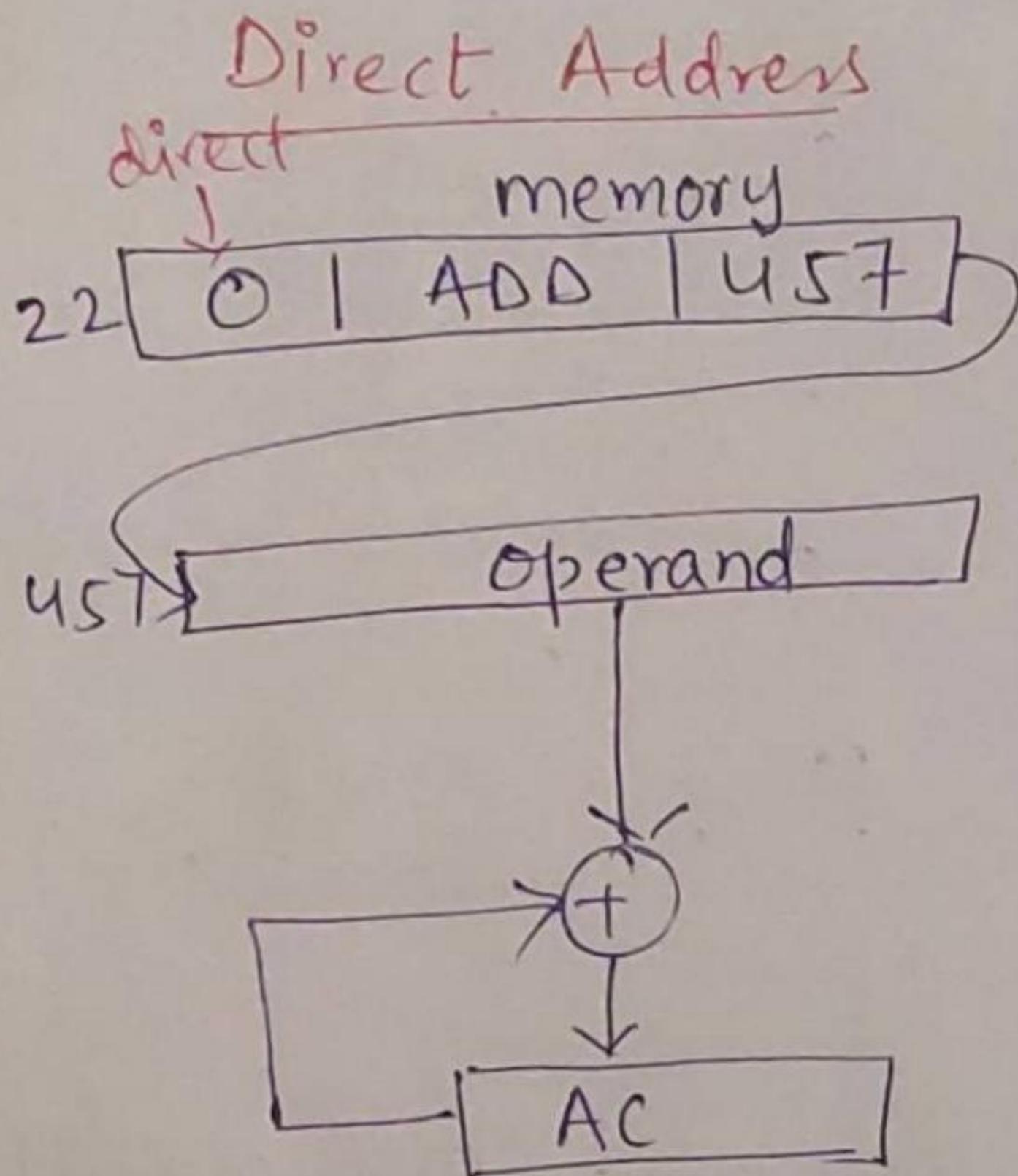
When second part of an instruction code

- ① specifies an operand  $\rightarrow$  immediate operand
- ② specifies an address of operand
  - $\downarrow$  direct add
- ③ designate an add of memory  $\rightarrow$  indirect add  
Word in which the address of operand found.



1 bit operation  
indirect 3-bit code  
add

① immediate operand





DETAILED LECTURE NOTES

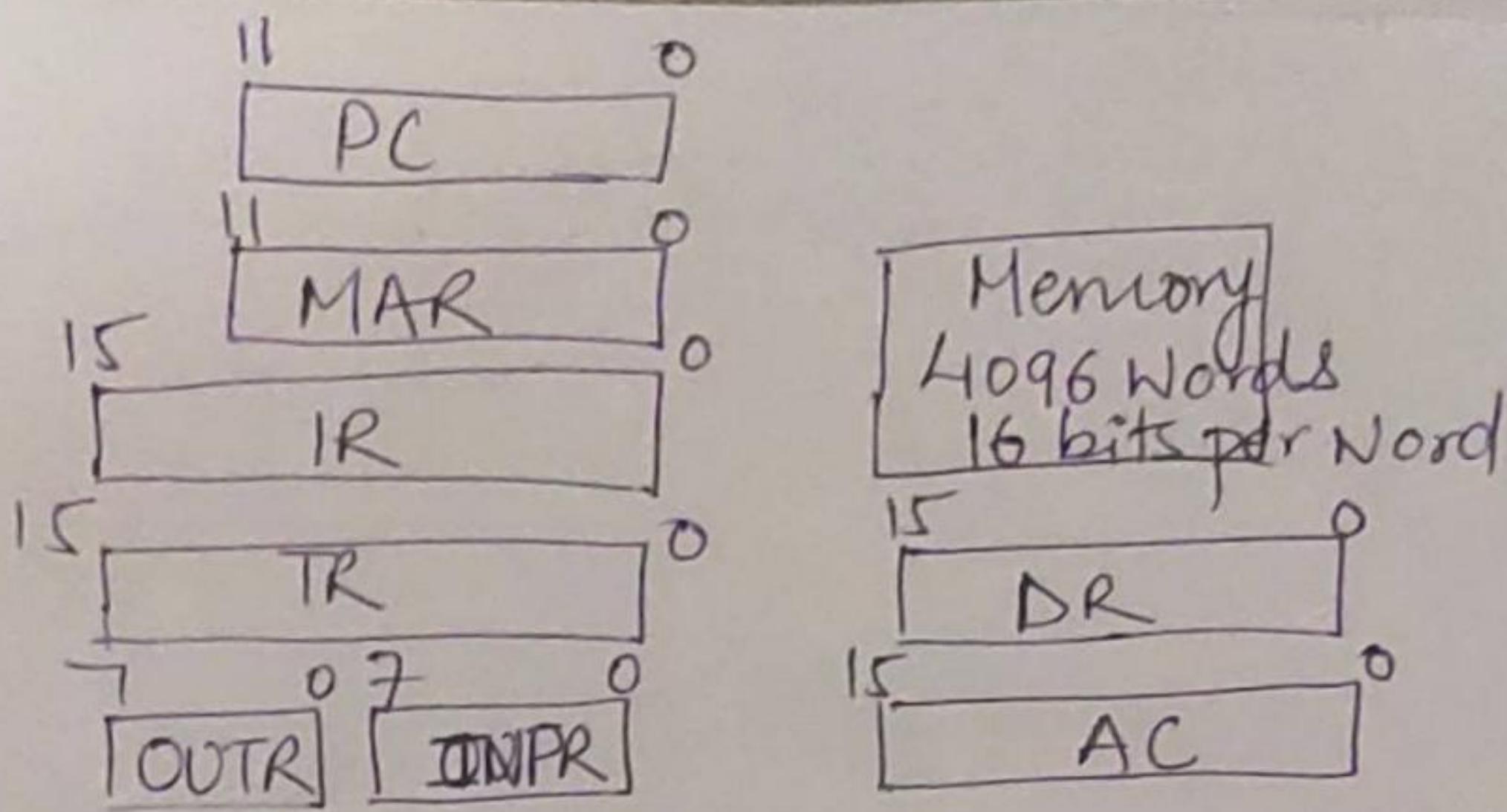
Computer Register:-

PAGE NO.

- Type of computer Memory used to quickly accept
- Register used by the CPU are often termed store
- Processor register can hold transfer data
- Instruction
- Storage add
- Data

Basic computer register

① Data register (DR)	16 bits	Hold memory operand
② Add Register (AR)	12 bits	Hold add of memory
③ Accumulator (AC)	16 "	Processor Register
④ Instruction register (IR)	16 "	Holds instruction code
⑤ Program counter (PC)	12	Holds add of instruction
⑥ Temporary Register (TR)	16	Hold temp. data
⑦ Input Register (INPR)	8	Carries I/P character
⑧ Output Register (OUTR)	8	Carries O/P character



## Working of a System

- Computer Instructions are normally stored in consecutive memory locations and are executed sequentially one at a time.
- So some counter is required to calculate the no. of instructions and address of the next instruction and memory address.
- Memory has capacity 4096 words, 16 bits / word.
- 12 bits of an instruction word are needed to specify the add. of operand.  
& 3 bits for operation part of instruction  
or direct & indirect add.



# POORNIMA

COLLEGE OF ENGINEERING

## DETAILED LECTURE NOTES

PAGE NO. ....

### Common Bus System : →

- Basic computer has 8 registers + Memory Unit + control unit.
- To transfer the info from one register to another or in memory, we use common bus (Multiplexor & 3 buffer gates)
- The o/p of 7 registers & Memory are connected to the common bus.
- Specific o/p is selected for the bus line at any given time, determined by the selection variable  $s_1, s_2, s_0$
- PC & AR are 12-bits so when AR, PC contents are applied to the 16-bit common bus, the 4 most significant bits are set to 0's.
- When AR or PC receive info from the bus only the 12 least significant bits are transferred into the register.

ift  
li  
ment

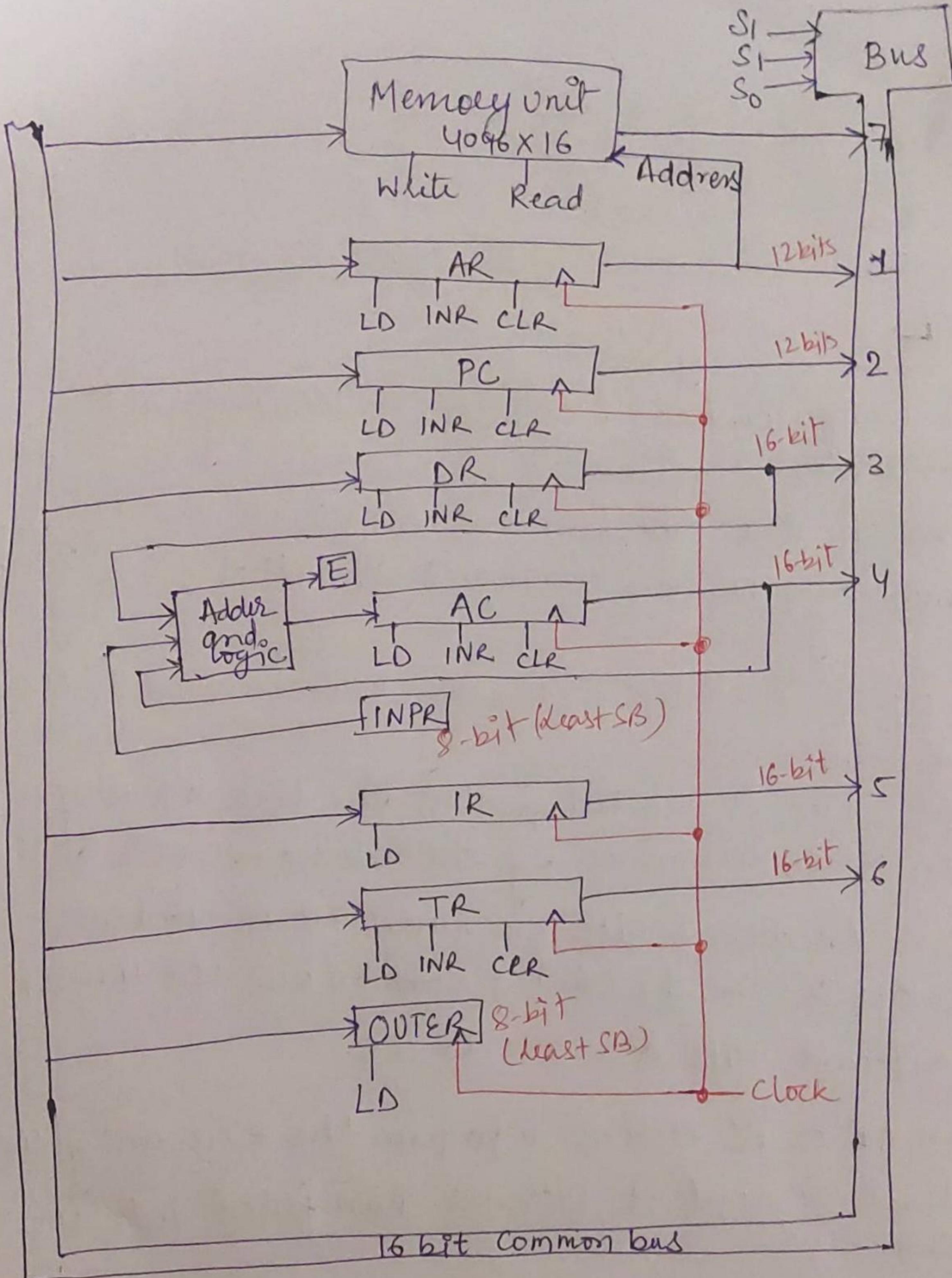


fig: Basic computer registers connected to a Common bus.



# POORNIMA

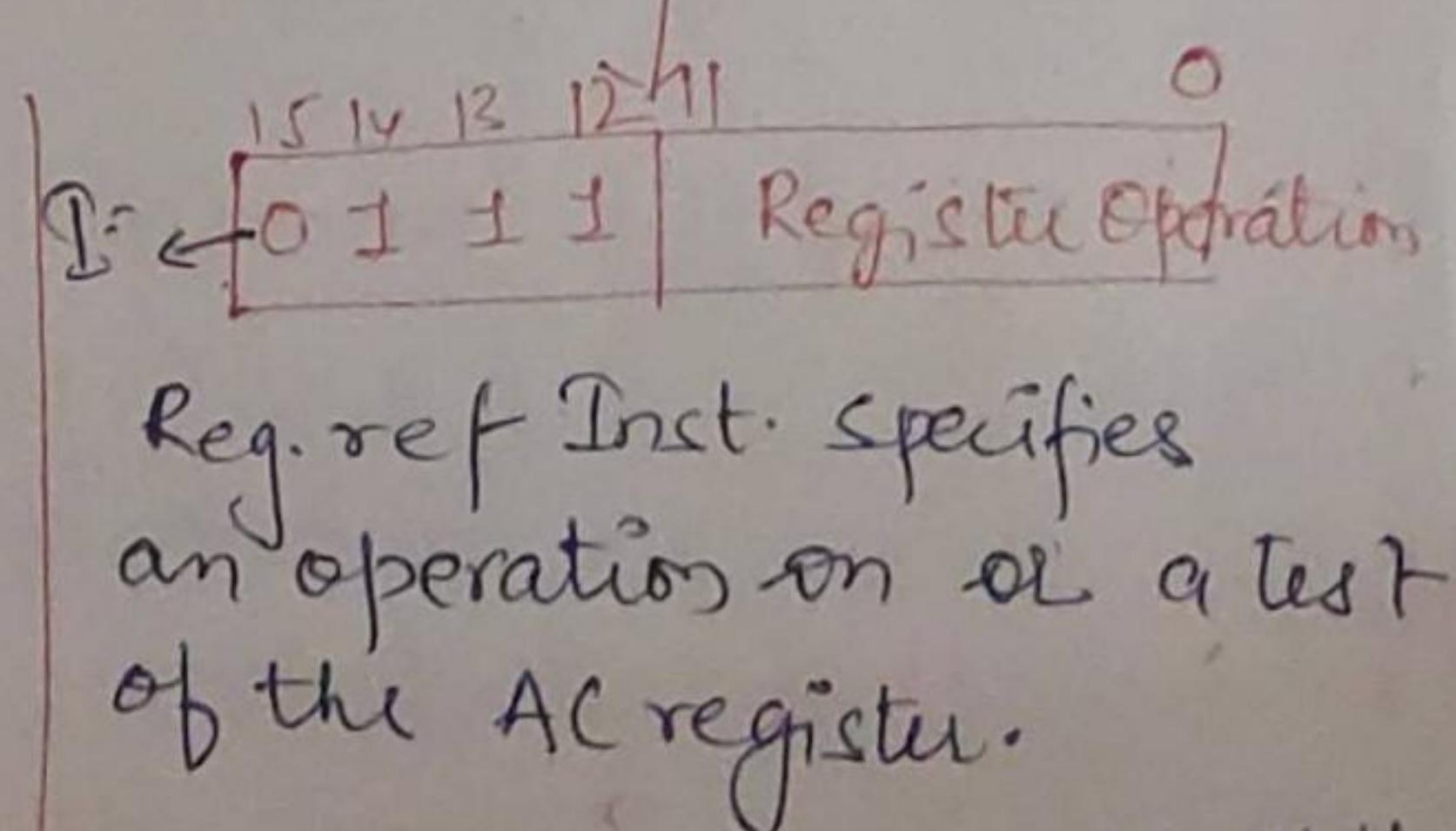
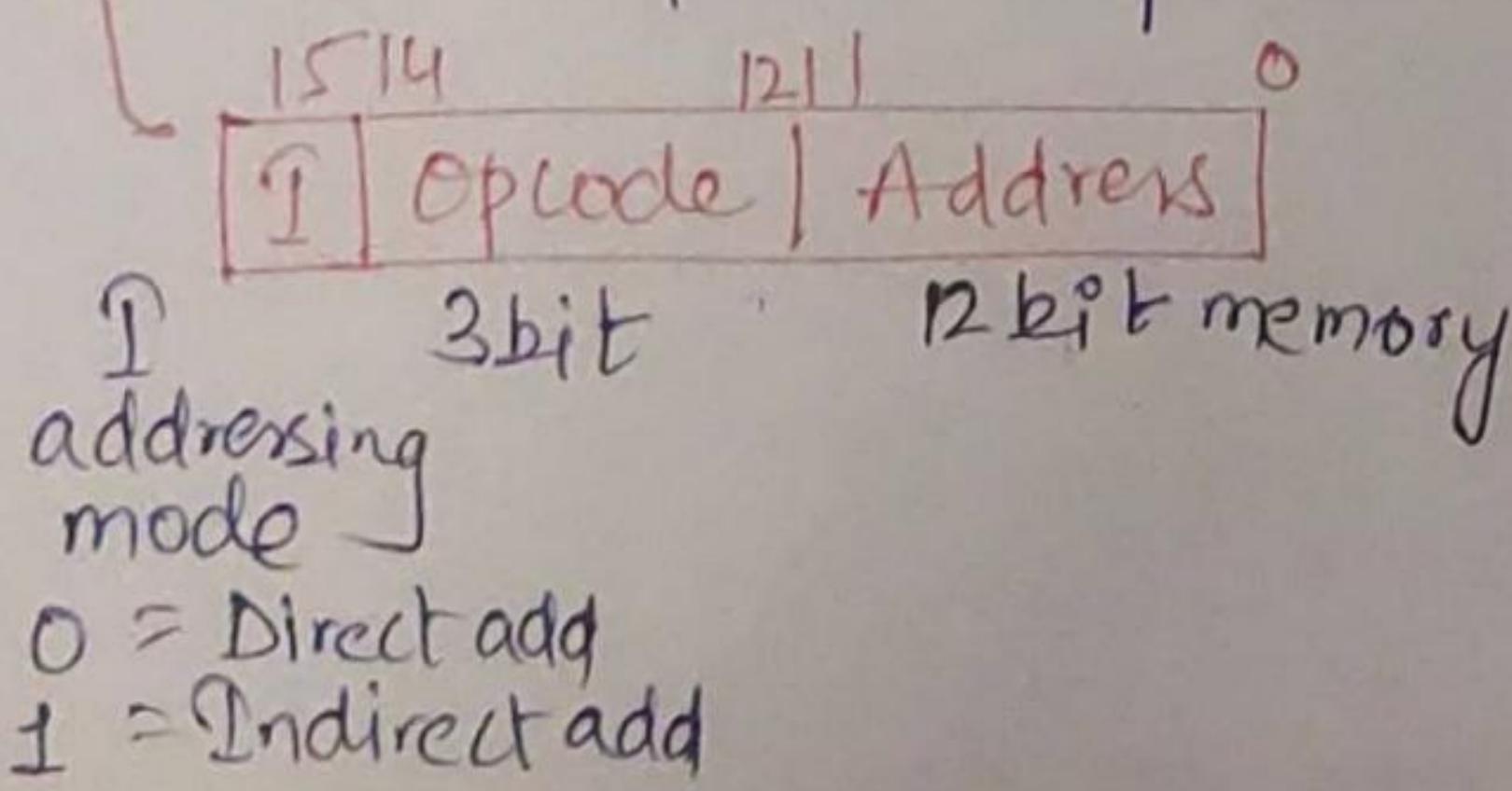
COLLEGE OF ENGINEERING

## DETAILED LECTURE NOTES

PAGE NO. 11

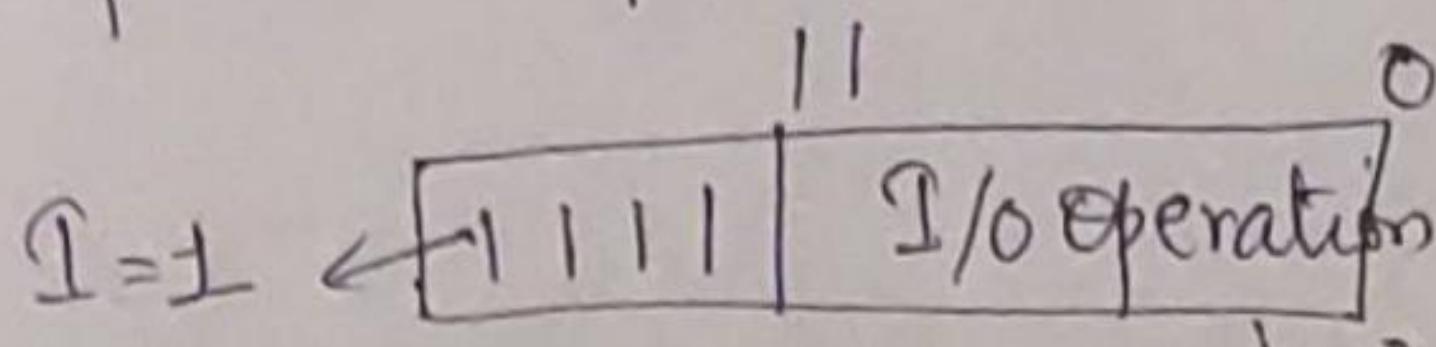
### Computer Instructions

- Computer Instructions are a set of m/c lang. instr. that a particular processor understand and execute.
- An inst. comprises of groups called field. These fields includes.
  - ① Operation Code (Opcode) filed
  - ② Add filed { location of operand Reg/M }
  - ③ Mode filed { operand will located }
- Basic comp has three instruction code formate
  1. Memory - reference instruction
  2. Register - reference instruction
  3. Input - output instruction



opcode 111      Add Shift  
Sub Compli  
MULT ment

### ③ Input - Output Instruction



12 bits are used to specify the type of input-op operation to

### Instruction - Set Completeness:

A set of instructions is said to be complete if the computer includes a sufficient no. of instructions in of the following categories;

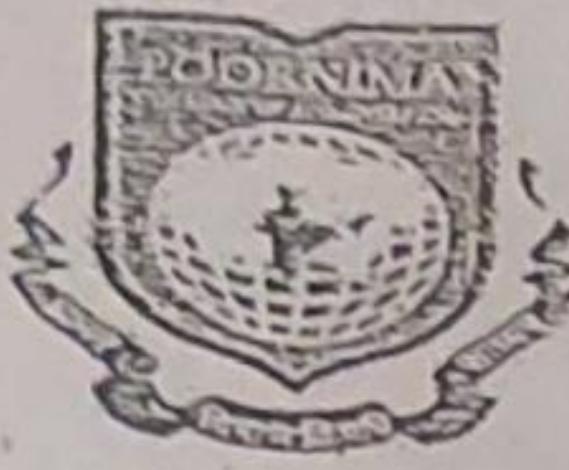
- ① Arithmetic, logical, and shift instructions
- ② Inst. for moving info. to and from memory and processor register.
- ③ Program control inst. together with instruction that check status conditions.
- ④ Input & Output instructions.

→ provide computational capabilities for processing the type of data the user may wish to employ.

→ huge data in memory so and computation in processor register, so capability of moving info b/w these two units.

→ branch instructions are used to change the sequence in which program is executed.

→ Act as interface b/w computer & user. Program & data must be translated into M and result back to user.



# Poornima

COLLEGE OF ENGINEERING

## DETAILED LECTURE NOTES

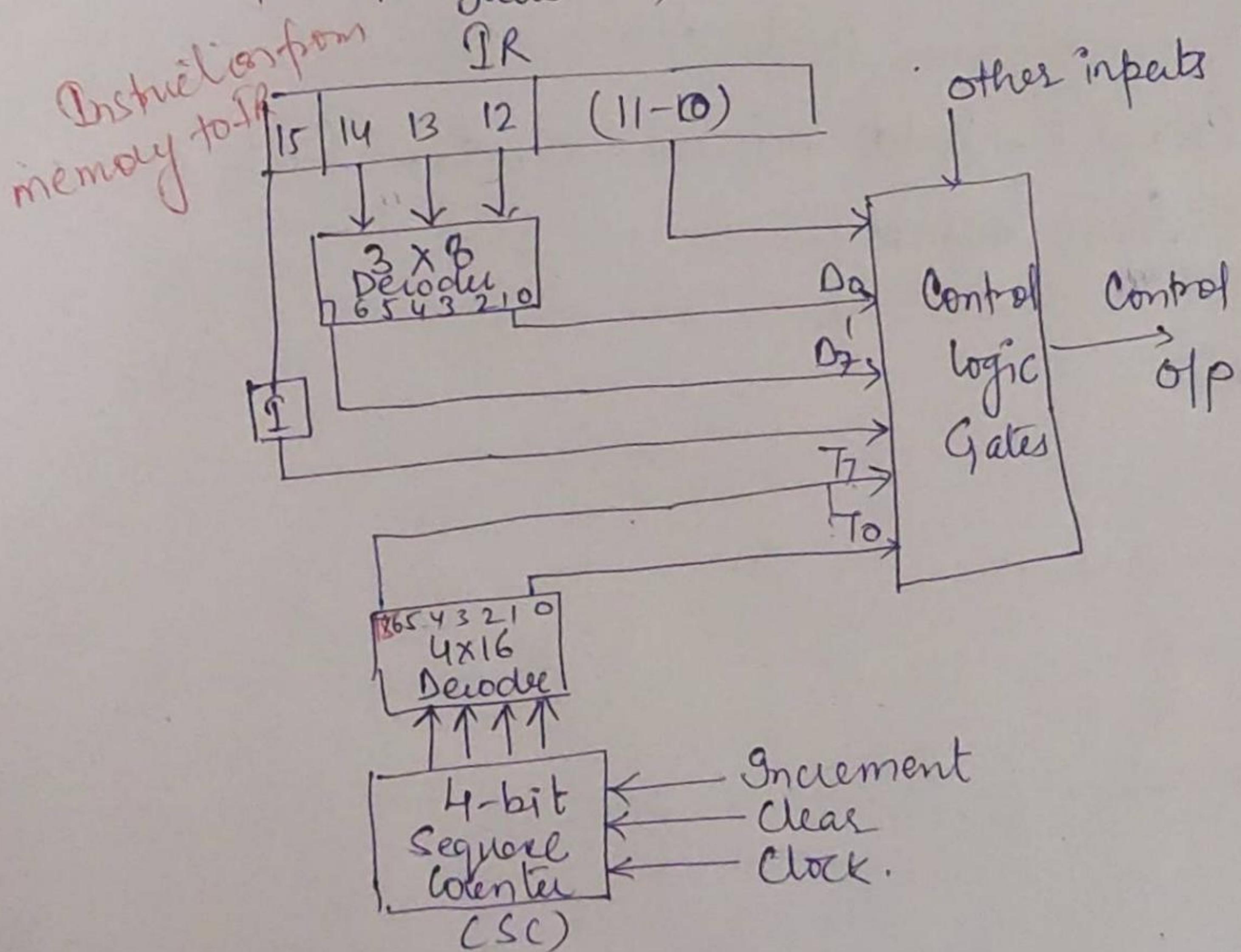
PAGE NO. 2

### Timing & Control :-

- Timing for all registers controlled by master clock.
- Two type of control organization

H/W control  
involves  
{ flip-flop, decoders }  
gates

Micro-Programmed Control.  
{ stored in control memory }



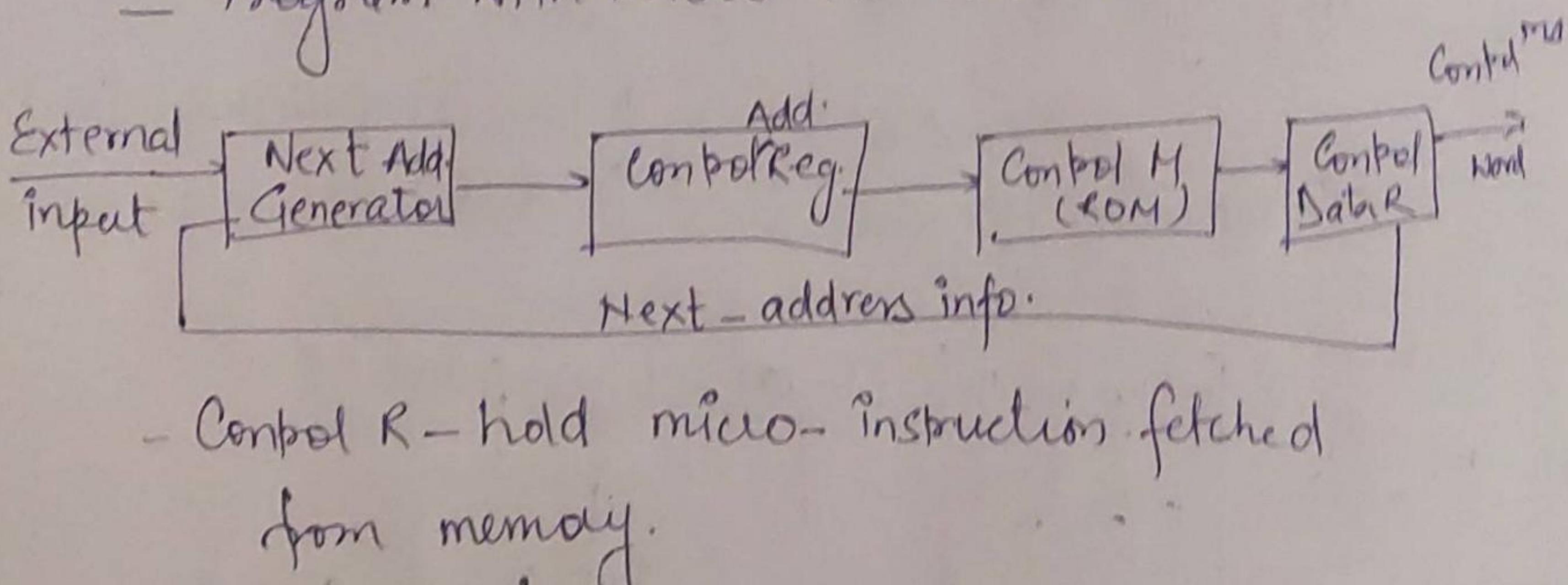
## Instruction cycles

### - Hardware control

- ① consist of two decoders, sequence counter, logic gates
- ② Instruction fetched from M to IR
- ③ IR include 1bit operation, 3bit Operand, 0 11 operand

## Microprogrammed Control

- Program with micro instructions.



- Control R - hold micro-instruction fetched from memory.



# POORNIMA

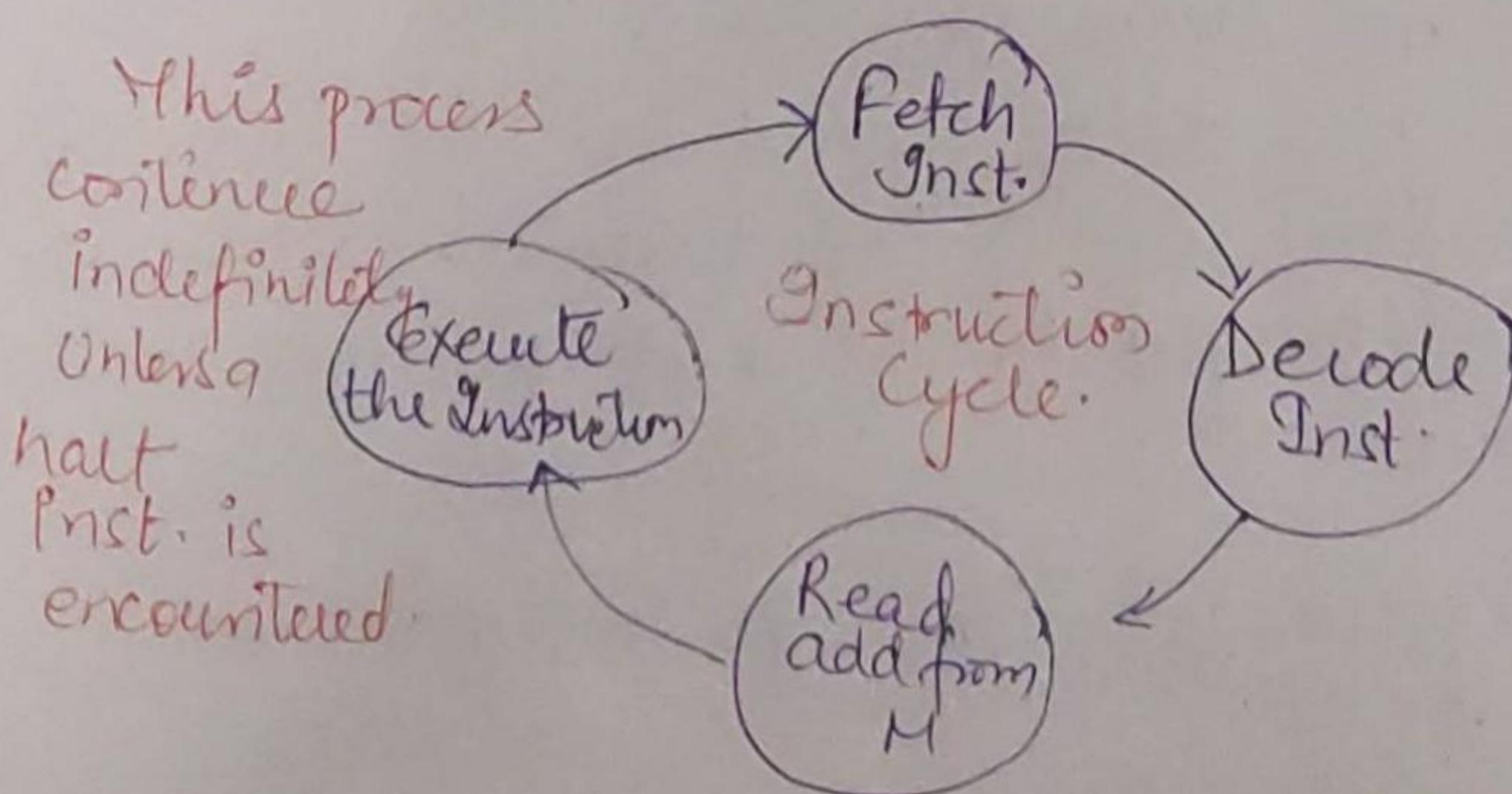
COLLEGE OF ENGINEERING

## DETAILED LECTURE NOTES

PAGE NO.

### Instruction Cycle :-

- A program residing in the memory unit of a computer consist of a seq; of instruction. These instruction are executed by the processor by going through a cycle for each instruction.
- Instruction cycle consists of following phases
  - ① Fetch an instruction from M
  - ② Decode the instruction
  - ③ Read the effective add from Memory
  - ④ Execute the instruction.



## Memory-Reference Instructions

- Micro-operations needed for the execution of each instruction.
- Memory-Reference Instructions are below

<u>Symbol</u>	<u>Operation Decoder</u>	<u>Symbolic Description</u>
AND	D <sub>0</sub>	$AC \leftarrow AC \wedge M[AR]$
ADD	D <sub>1</sub>	$AC \leftarrow AC + M[AR]$
Load to AC $\leftarrow LDA$	D <sub>2</sub>	$AC \leftarrow M[AR]$
Store AC $\leftarrow STA$	D <sub>3</sub>	$M[AR] \leftarrow AC$
Branch $\leftarrow BN$	D <sub>4</sub>	$PC \leftarrow AR$
Unconditionally $\leftarrow BSA$	D <sub>5</sub>	$M[AR] \leftarrow PC, PC \leftarrow AR+1$
Branch & $\leftarrow ISZ$	D <sub>6</sub>	$M[AR] \leftarrow M[AR]+1$ If $M[AR]+1=0$ then $PC \leftarrow PC+1$
Save Return Address	Increment & Skip if zero.	

## Input-Output & Interrupt:

- Computer used for comm' b/w User & HW.
- Instructions and data stored in the memory must come from some input device. The result are displayed to the user through some output device.
- block diagram for i/p - o/p configurations

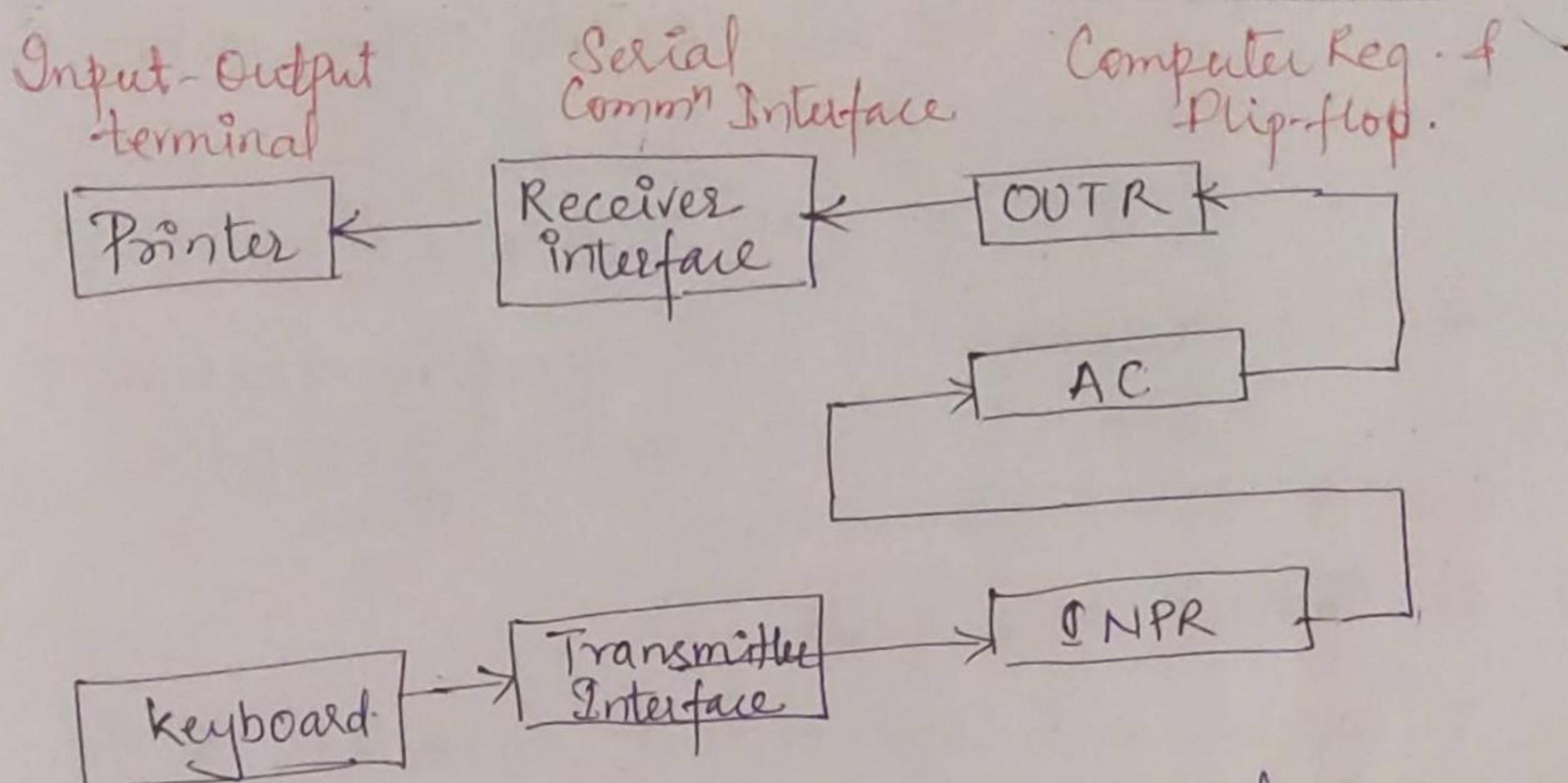


# POORNIMA

COLLEGE OF ENGINEERING

## DETAILED LECTURE NOTES

PAGE NO.



- GP-OP terminal send & receive info.
- Ant. of info. transferred will always have 8-bits of an alphanumeric code.
- The info generated through the keyboard is shifted into INPR register.
- If info for printer is stored in OUTR
- Register INPR and OUTR comm' with a comm' interface serially and with AC in parallel.
- The transmitter interface receives info from the keyboard and transmit it to INPR printer.
- Receiver I/f receives info. from OUTR and sent it to ^

## Design of Basic Computer :-

Basic computer consist of following h/w components

1. A memory unit with 4096 words of 16 bits each.
2. Nine Registers:  
AC, PC, DR, AC, IR, TR, OUTR, INPR, SC  
(Seq. Unit)
3. Seven flip-flop  
T, S, E, R, IEN, FG<sub>1</sub> and FG<sub>0</sub>
4. Two decoders: a 3x8 operation decoder & 4x16 timing decoder
5. A 16-bit common bus
6. Control logic gates
7. Adder and logic circuit connected to the input of AC.

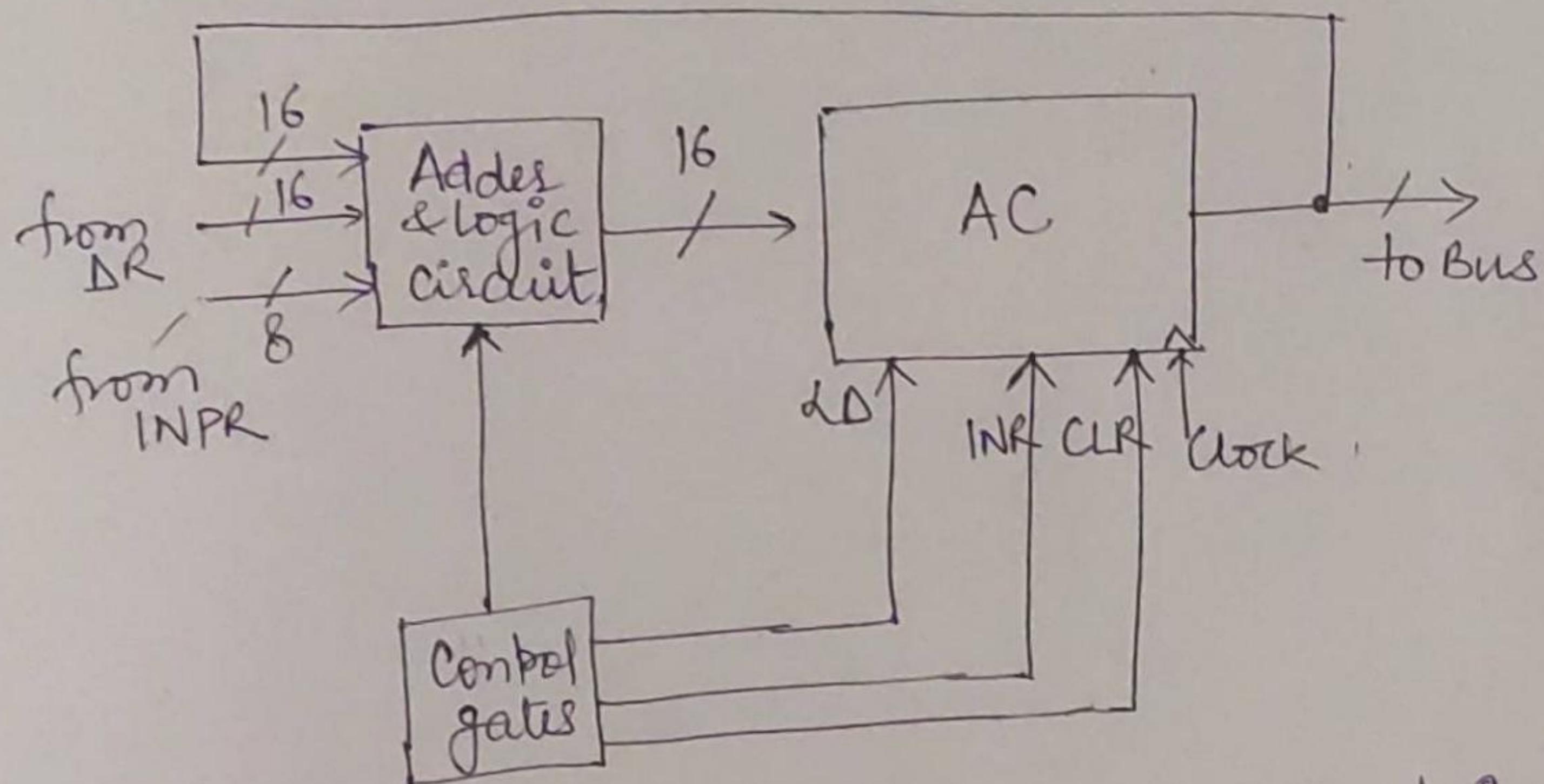


DETAILED LECTURE NOTES

PAGE NO. \_\_\_\_\_

Design of Accumulator logic: →

- The circuits associated with the AC register below.



- Adder & logic circuit has 3 sets of inputs.
- One set of 16 inputs comes from the op of AC, second 16 inputs from DR, third set 8 inputs from INPR.
- AC have control gates LD, INR, CLR and clock.