

## # Computer Instructions

- The basic computer has three instruction code formats and each format has 16 bits.

a) Memory - Reference Instruction

15	14	12 11		0
I	Opcode	Address		

(Opcode = 000 - 110)

b) Register - Reference Instruction

15	14	12 11		0
0	1 1 1	Register Operation		

(Opcode = 111, I = 0)

c) Input - Output Instruction

15	14	12 11		0
1	1 1 1	I/O Operation		

(Opcode = 111, I = 1)

## # Instruction Cycle

- A program residing in the memory unit of the computer consist of sequence of instructions.
- The program is executed in the computer by going through a cycle for each instruction.
- In the basic computer each instruction cycle consist of following phases:

1. Fetch an instruction from memory.
  2. Decode the instruction.
  3. Read the effective address from memory if the instruction has indirect address.
  4. Execute the instruction.
- Operations for Fetch and Decode
- T<sub>0</sub> : AR  $\leftarrow$  PC
- T<sub>1</sub> : IR  $\leftarrow$  M[AR], PC  $\leftarrow$  PC + 1
- T<sub>2</sub> : D<sub>0</sub> --- D<sub>7</sub>  $\leftarrow$  Decode IR (12-14)
- AR  $\leftarrow$  IR (0-11)
- I  $\leftarrow$  IR (15)
- Initially, program counter PC is loaded with the address of the first instruction in the program.
  - Place the content of PC into AR.
  - The instruction is extracted in IR (Instruction register) from AR (Address Register.)
  - Last, decode the instruction register.
  - The flow chart of instruction cycle is as shown

# Flow Chart of Instruction Cycle.

Memory.

$\downarrow$  IR

Fetch

$\downarrow$

Decode.

$14 \quad 12$

Address

decode the opcode

Start  $SC \leftarrow 0$

$\downarrow T_0$

$AR \leftarrow PC$

$\downarrow T_1$

$IR \leftarrow M[AR], PC \leftarrow PC + 1$

$\downarrow T_2$

Decode Opcode in  $IR(12-14)$   
 $AR \leftarrow IR(0-11), I \leftarrow IR(15)$

Fetch

Decode

Register or I/O

D $\oplus$

(000 - 110)

0 (Memory Reference)

I/O

I (Register)

(1) Indirect

I

(0) Direct

Execute  
I/P - O/P  
Instruction  
 $SC \leftarrow 0$

Execute  
Register-  
Reference  
Instn  
 $SC \leftarrow 0$

$AR \leftarrow M[AR]$

Execute  
Memory - Reference  
Instruction  
 $SC \leftarrow 0$

T4

T3

T2

111

Decision

Execute

## # Execution of Complete Instruction

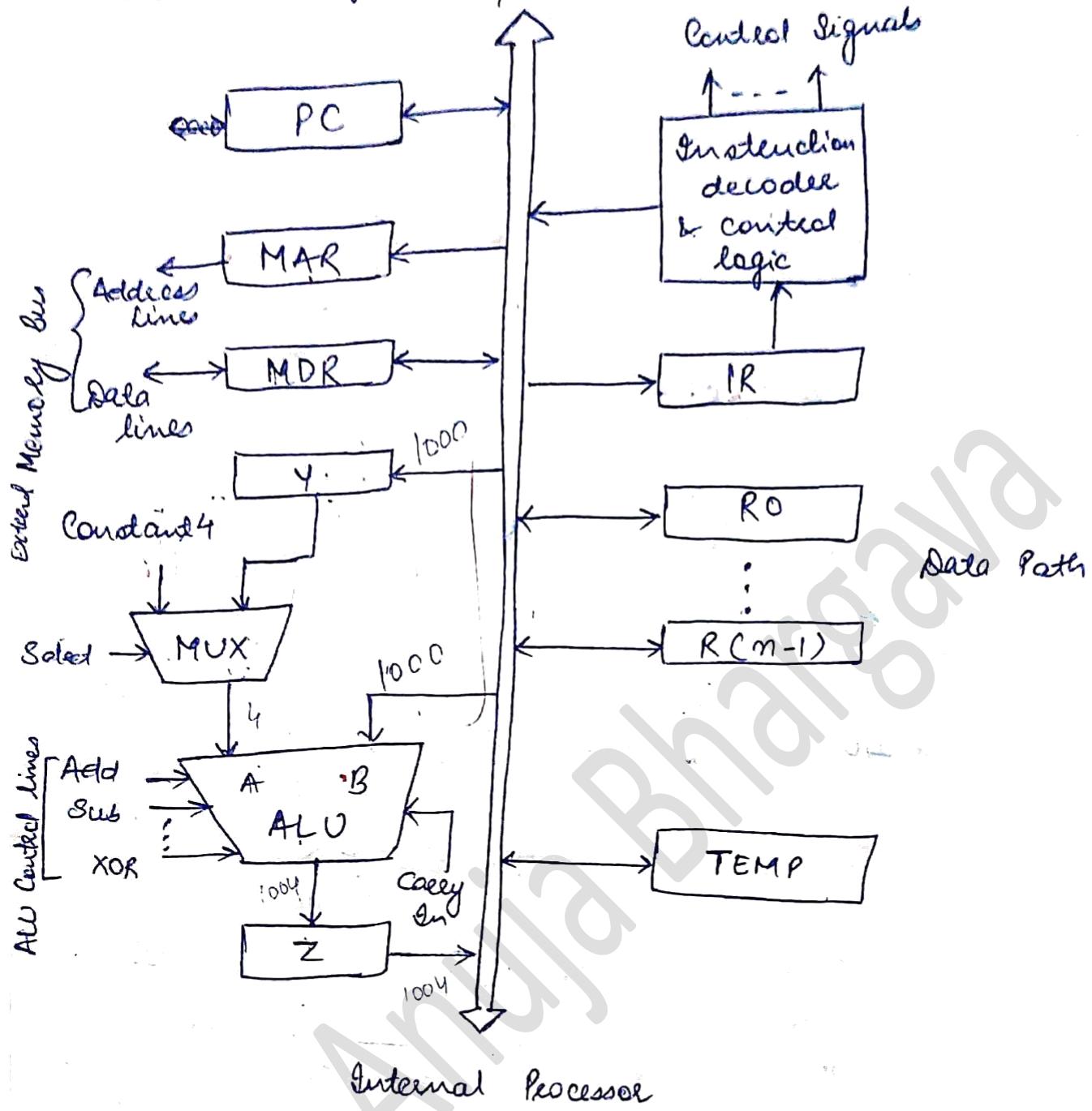


Fig:- Architecture of <sup>bus</sup> Single Bus Processor  
To execute the instruction, through a single bus following steps are to be followed.

- ① Fetch the data
- ② decode
- ③ Extract the address
- ④ Execute

In figure

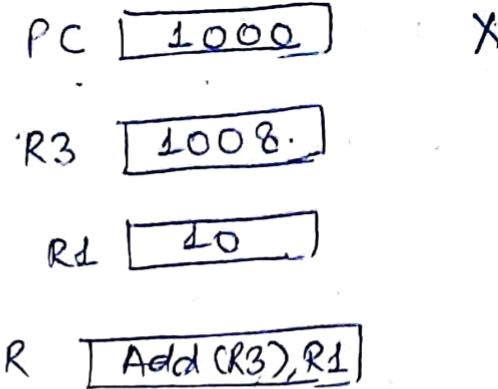
- ① R<sub>0</sub> to R<sub>n-1</sub> are general purpose register
- ② IR - Instruction / Indexed Register  
It holds the instruction currently being executed
- ③ Instruction decoder - tells about the instruction to be performed

for feeding :- (i) AR ← PC

- ④ Y, Z & TEMP are temporary registers used by processor during the execution of some instruction
- ⑤ PC : contains the address of next instruction to be executed.
- ⑥ MAR - Memory Address Register
- ⑦ MDR - All the data stored in Memory Data Register

Memory

Address	Content
1000	Add(R3), R1
1008	20



- 1) Fetch the instruction from memory
- 2) Fetch the first operand from memory.
- 3) Perform addition operation
- 4) Load the result into R1.

Control Sequence

- 1) PC<sub>out</sub>, MAR<sub>in</sub>, Read, Select 4, Add, Z<sub>in</sub>
- 2) Z<sub>out</sub>, PC<sub>in</sub>, W MFC  
(Wait for memory function to complete)
- 3) MDR<sub>out</sub>, IR<sub>in</sub>
- 4) ~~R3<sub>out</sub>, MAR<sub>in</sub>, Read~~  
5) ~~R1<sub>out</sub>, A<sub>in</sub>, W MFC~~
- 6) MDR<sub>out</sub>, Select 4, Add, Z<sub>in</sub>       $Z = 1004$
- 7) Z<sub>out</sub>, R<sub>in</sub>, End.       $PC \rightarrow 1004$   $MDR_{out} \rightarrow ADD(R3), R1$

Q Write control sequence for following instruction.

(a) ADD (R3), R1

1. PC<sub>out</sub>, MAR<sub>in</sub>, Read, Select 4, Add, Z<sub>in</sub>
  2. Z<sub>out</sub>, PC<sub>in</sub>, Y<sub>in</sub>, WMFC
  3. MDR<sub>out</sub>, IR<sub>in</sub>
  4. R<sub>3</sub> out, MAR<sub>in</sub>, Read
  5. R<sub>1</sub> out, Y<sub>in</sub>, WMFC
  6. MDR<sub>out</sub>, Select 4, Add, Z<sub>in</sub>
  7. Z<sub>out</sub>, R<sub>in</sub>, End.
- \* WMFC - Wait for memory function to complete.

b) Add the (immediate) number NUM to register R1.

1. PC<sub>out</sub>, MAR<sub>in</sub>, Read, Select 4, Add, Z<sub>in</sub>
2. Z<sub>out</sub>, PC<sub>in</sub>, Y<sub>in</sub>, WMFC
3. MDR<sub>out</sub>, IR<sub>in</sub>
4. IR NUM out, Y<sub>in</sub>
5. R<sub>1</sub> out, Select 4, Add, Z<sub>in</sub>
6. Z<sub>out</sub>, R<sub>1</sub> in, End

- c) Add the contents of memory location NUM to register R1.
1. PC<sub>out</sub>, MAR<sub>in</sub>, Read, Select 4, Add, Z<sub>in</sub>
  2. Z<sub>out</sub>, PC<sub>in</sub>, Y<sub>in</sub>, WMFC
  3. MDR<sub>out</sub>, IR<sub>in</sub>
  4. IR NUM<sub>out</sub>, MAR<sub>in</sub>, Read
  5. R1<sub>out</sub>, Y<sub>in</sub>, WMFC
  6. MDR<sub>out</sub>, Select 4, Add, Z<sub>in</sub>
  7. Z<sub>out</sub>, R1<sub>in</sub>, End.

d) Add the content of memory location whose address is at memory location NUM to register R1.

1. PC<sub>out</sub>, MAR<sub>in</sub>, Read, Select 4, Add, Z<sub>in</sub>
2. Z<sub>out</sub>, PC<sub>in</sub>, Y<sub>in</sub>, WMFC
3. MDR<sub>out</sub>, IR<sub>in</sub>
4. IR NUM<sub>out</sub>, MAR<sub>in</sub>, Read
5. WMFC
6. MDR<sub>out</sub>, MAR<sub>in</sub>, Read
7. R1<sub>out</sub>, Y<sub>in</sub>, WMFC
8. MDR<sub>out</sub>, Select 4, Add, Z<sub>in</sub>
9. Z<sub>out</sub>, R1<sub>in</sub>, End.

Q What is the sequence of operations to add the content of register R1 to those of R2 and store the result in R3?

1. R1 out, Y<sub>in</sub>
2. R2 out, Select Y, Add, Z<sub>in</sub>
3. Z out, R3<sub>in</sub>

Q Fetching a word from memory

MOV (R1), R2

1. R1 out, MAR<sub>in</sub>, Read
2. MDR<sub>in</sub>, WMFC
3. MDR out, R2<sub>in</sub>

Q Storing a word from memory.

MOV R2, (R1)

1. R1 out, MAR<sub>in</sub>
2. R2 out, MDR<sub>in</sub>, Write
3. MDR out E, WMFC

Dr Anuja Bhargava