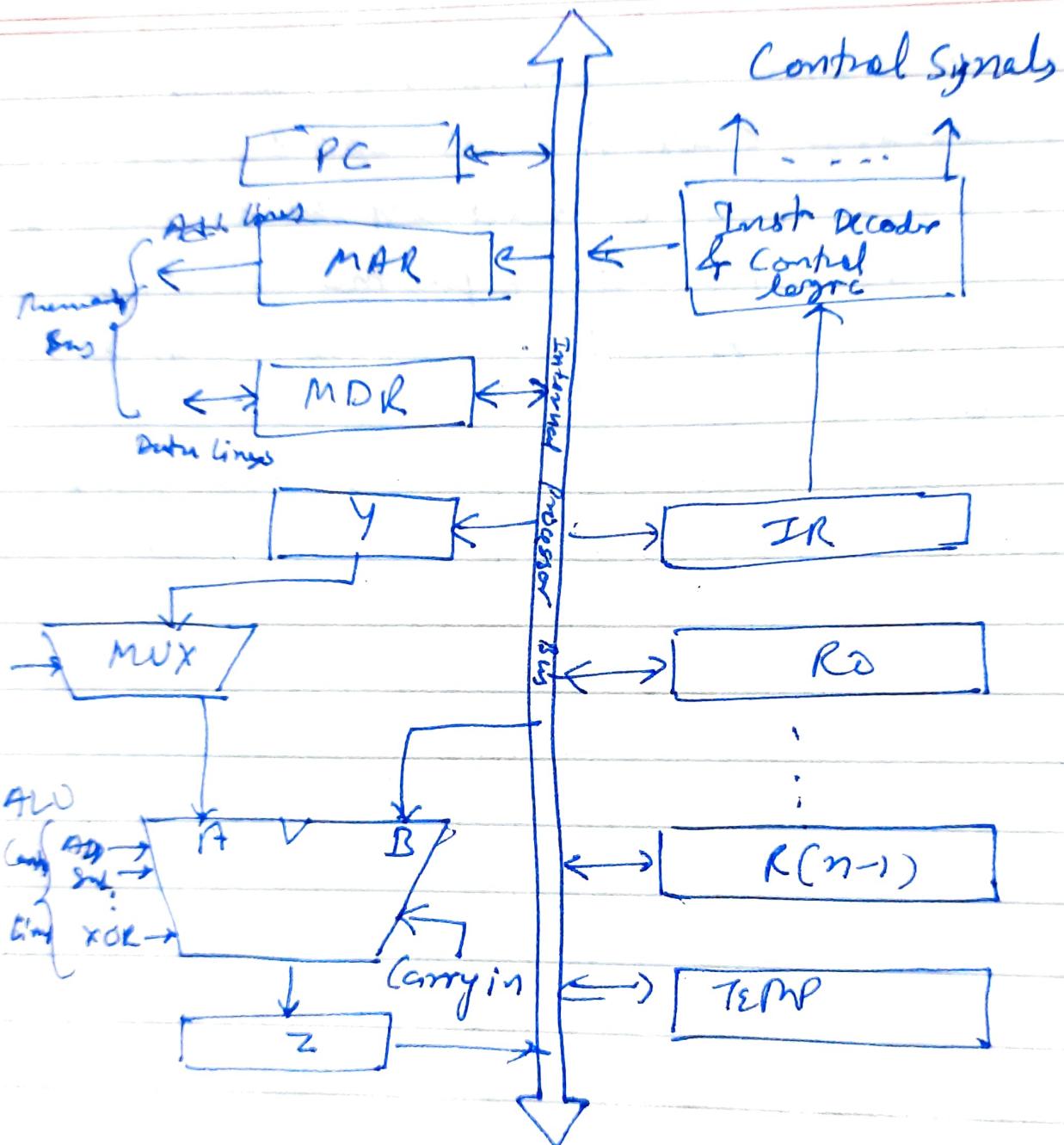


# UNIT-3

## Lecture-17



Sing Bus Arch.  
ADD R1, M[R3]

Ans:

1. PCout MARin, READ, SELECT-4, ADD, Zin
2. Zout, PCin Yin WMFC
3. MDRout, IRin
4. R3out, MARin, READ
5. R1out Yin WMFC
6. MDRout, SELECT-4, ADD, Zin
7. Zout, R1in END

ADD RI, R2

1. PCout, MARin READ, SELECT -cont, ADD, Zi
2. Zout PCin Yin WMFC
3. MDRout, IRin
4. R1out, Yin
5. R2out, ADD, Zin
6. Zout, R1in END

Assuming that each instruction consists of two words. The first word specifies the operation and the second word contains the number NUM. Also assume a single bus structure as discussed. Write the sequence of control steps required for:

- (a) Add the number NUM to register R1.
- (b) Add the contents of memory location NUM to register R1.
- (c) Add the contents of memory whose address is at the memory location NUM to register R1.

Ans.

(g)

1. PCout, MARin, Read, Select 4, Add, Zin
2. Zout, RIn, Yin, WMFC
3. MDRout, IRin
4. PCout, MARin, Read, Select 4, Add, Zin
5. Zout, PCin, Yin
6. ROut, Yin WMFC
7. MDRout, Select 4, Add, Zin
8. Zout, RIn, End.

(g)

- 1.
  - 2.
  - 3.
  - 4.
  - 5.
- Some
- B

## Lecture-18

6. MDRout, MARin, Read
7. Rlout, Yin, WMFC
8. MDRout, Select Y, Add, Zin
9. Zout, Rlin, End

(C)

1.

2.

3.

4.

5.

6. MDRout, MARin Read WMFC

7. MDRout MARin, Read.

8. Rlout, Yin, WMFC

9. MDRout, Select -Y, Add, Zin

10. Zout, Rlin, End

## Branch Instruction

- Replaces the contents of PC with the branch target address.
  - Usually obtained by adding offset  $\delta$ , given in the branch instruction, to the update value of PC.
  - Offset is the difference b/w the branch target address and the address immediately following the branch instruction.
  - If branch instruction is at location 2020 and branch target address is 2052, the value of offset is 46.

## The Fetch Phase

1. PCout, MARin, Read, Select 4, Add, Zin
2. Zout, PCin, Yin, WMFC
3. MDRout, IRin

END of Fetch Phase

The offset value is extracted from the IR by the instruction decoding circuit

4. Offset-field-of-IRout, Select 4, Add, Zin
5. Zout, PCin End

## Conditional Branch Statement

We need to check the status of the condition codes before loading a new address into the PC for example for a branch-on-negative ( $CBranch < 0$ )

1. PCout, MARin, Read, Select 4, Add, Zin
2. Zout, PCin, Yin, WMFC
3. MDRout, IRin
4. Offset-field-of-IRout, Select 4, Add, Zin,  
If  $N=0$  then End
5. Zout, PCin, End

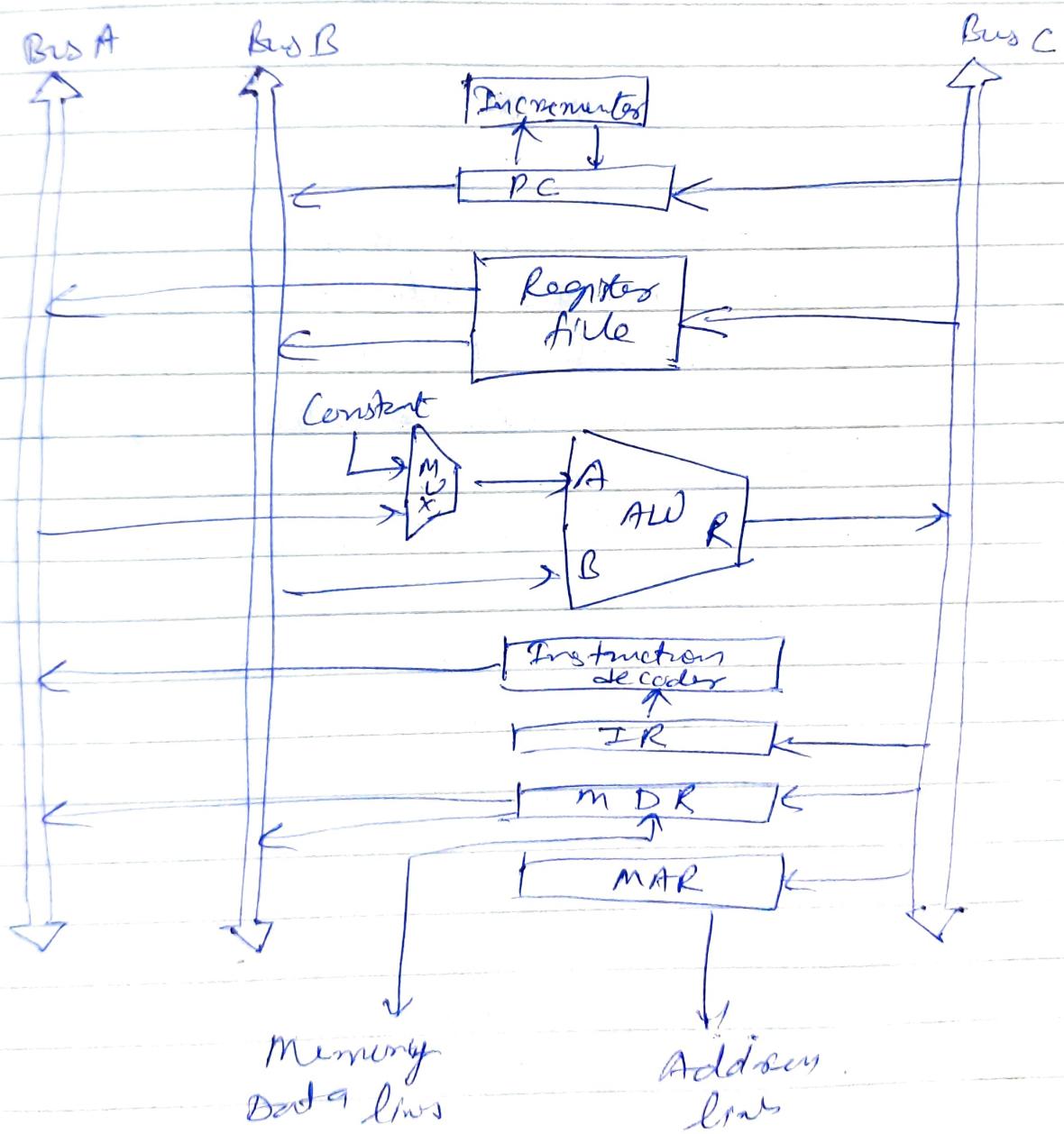
Thus if  $N=0$ , the processor returns to step 1 immediately after step 4. If  $N=1$ , step 5 is performed to load a new value into PC, thus performing the branch operation.

## Lecture 19

ADD R4, R5, R6

Multiple Bus

1. PCout, R=B, MARin, Read, Inc PC
2. WMRCE
3. MDRoutB, R=B, IRin
4. RAoutA, RSoutB, SelectA, Add, R6in End



## Lecture-10

### Control Unit

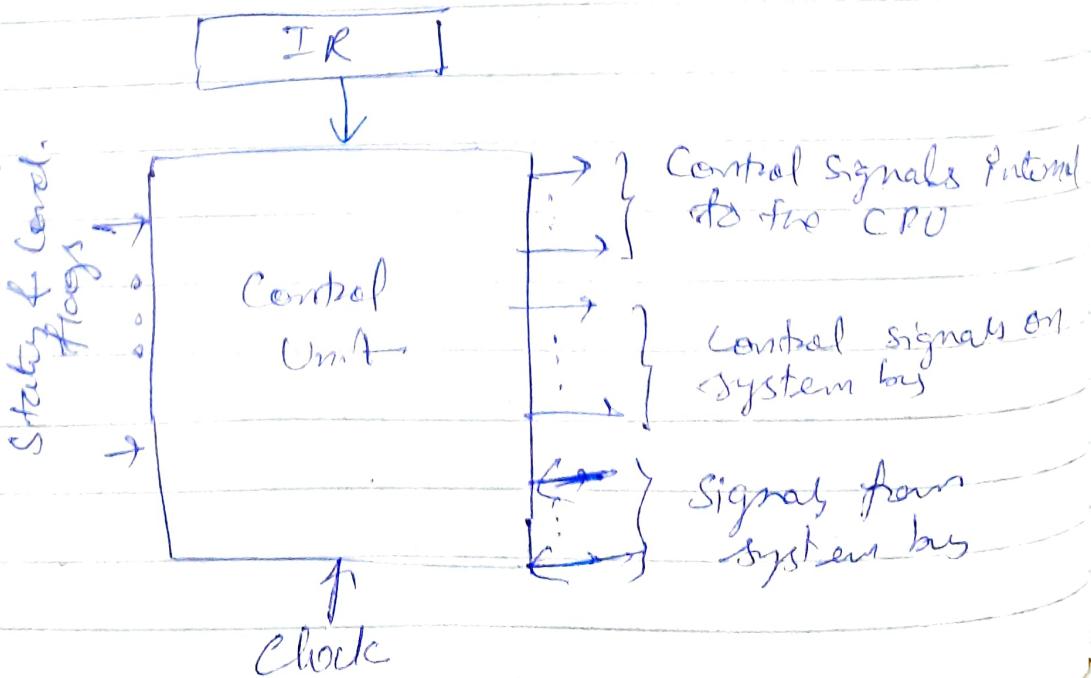
Basic tasks of the control unit →

- for each instruction the control unit causes the CPU to go through a sequence of control steps

- In each control step the control unit issues a set of signals which causes the corresponding microoperations to be executed.

→ The control unit is driven by the processor clock

- The signals to be generated at a certain moment depend on:
  - the actual step to be executed
  - the conditional f-status flags of the processor
  - the actual instruction executed
  - external signals received on the system bus (e.g. interrupt signals)



## Lecture-21

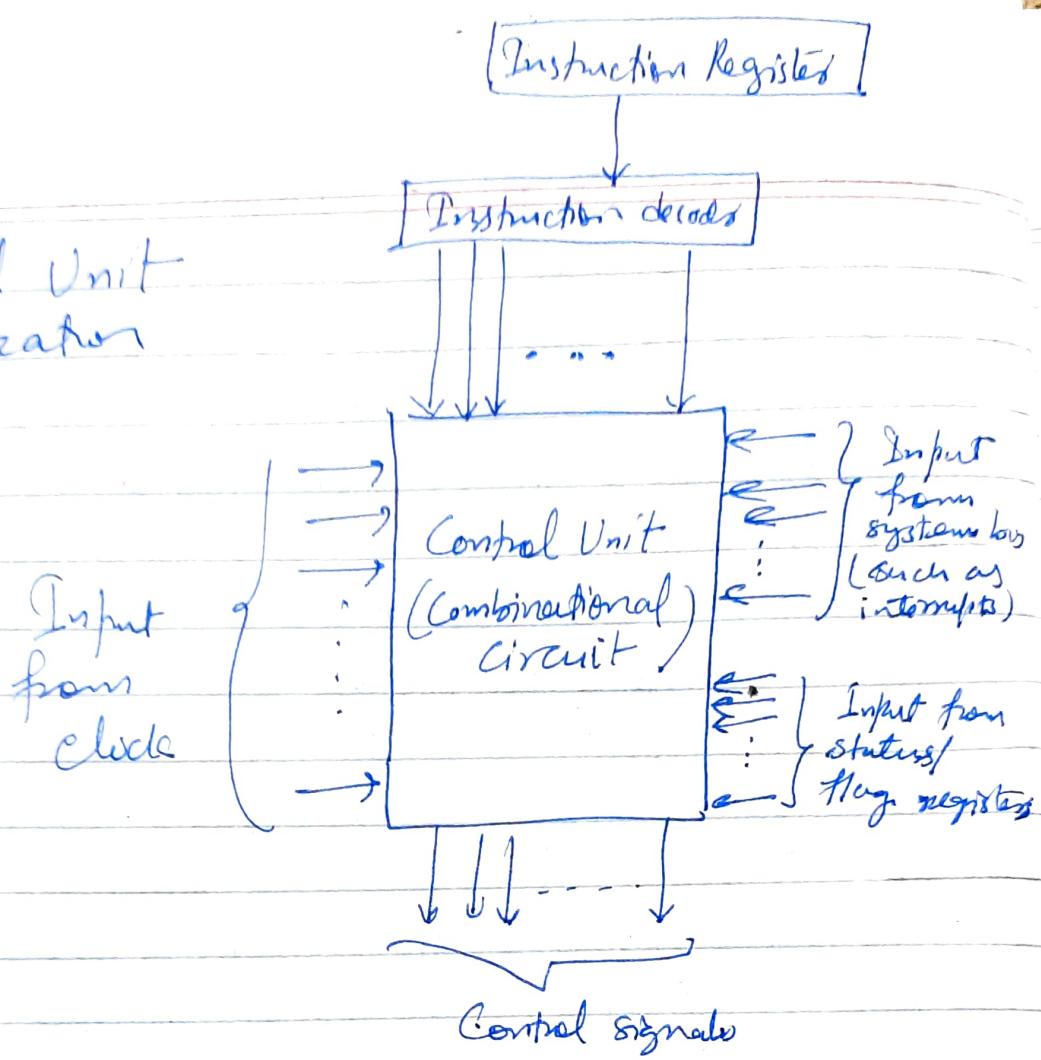
### Control Unit design

- \* To execute instructions, the processor must have some means of generating the control signals needed in proper sequence.
- \* Techniques for implementation of the control unit:
  1. Hardwired control
  2. Microprogrammed control

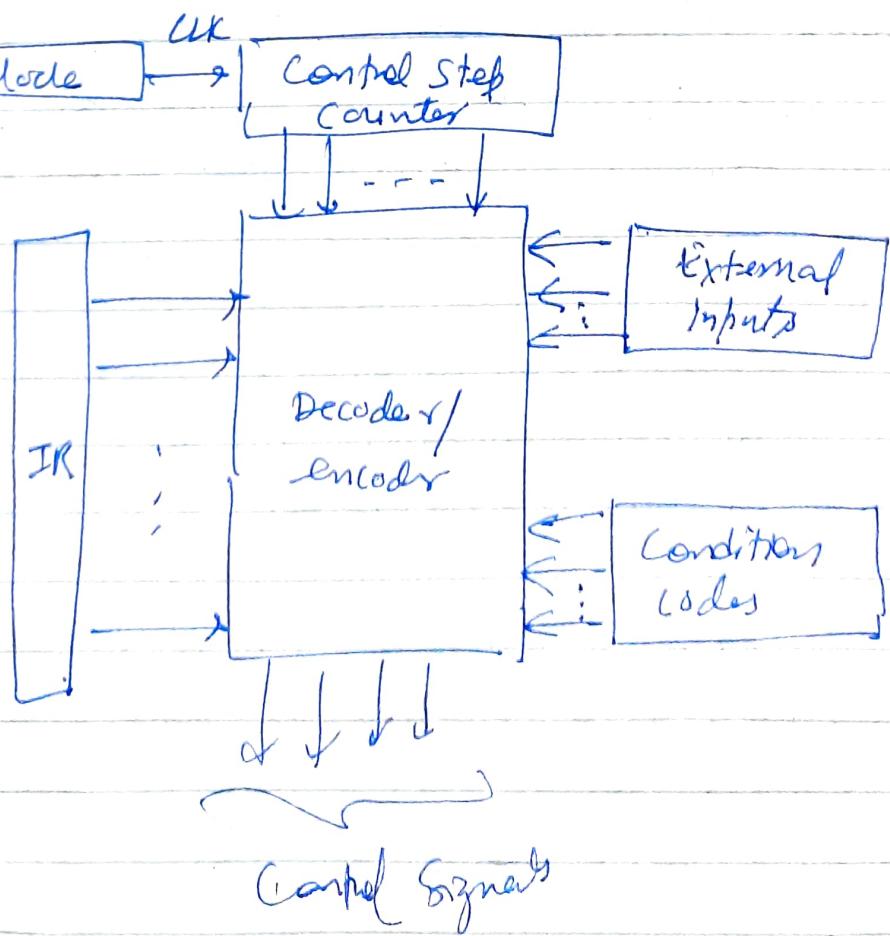
### Hardwired Control Unit

- In hardwired control units, control signals emanate from blocks of digital logic components.
  - These signals direct all of the data and instruction traffic to appropriate part of the system.
  - We need a special digital circuit that uses as inputs, the bits from the opcode field in our instructions, bits from the flag (or status) register, signals from the clock. It should produce, as outputs, the control signals to drive the various components in the computer.
  - All of the control lines are physically connected to the actual machine instructions.
  - The instructions are divided up into fields, and different bits in the instruction are combined through various digital logic components to drive the control lines.

## Control Unit organization



Q2



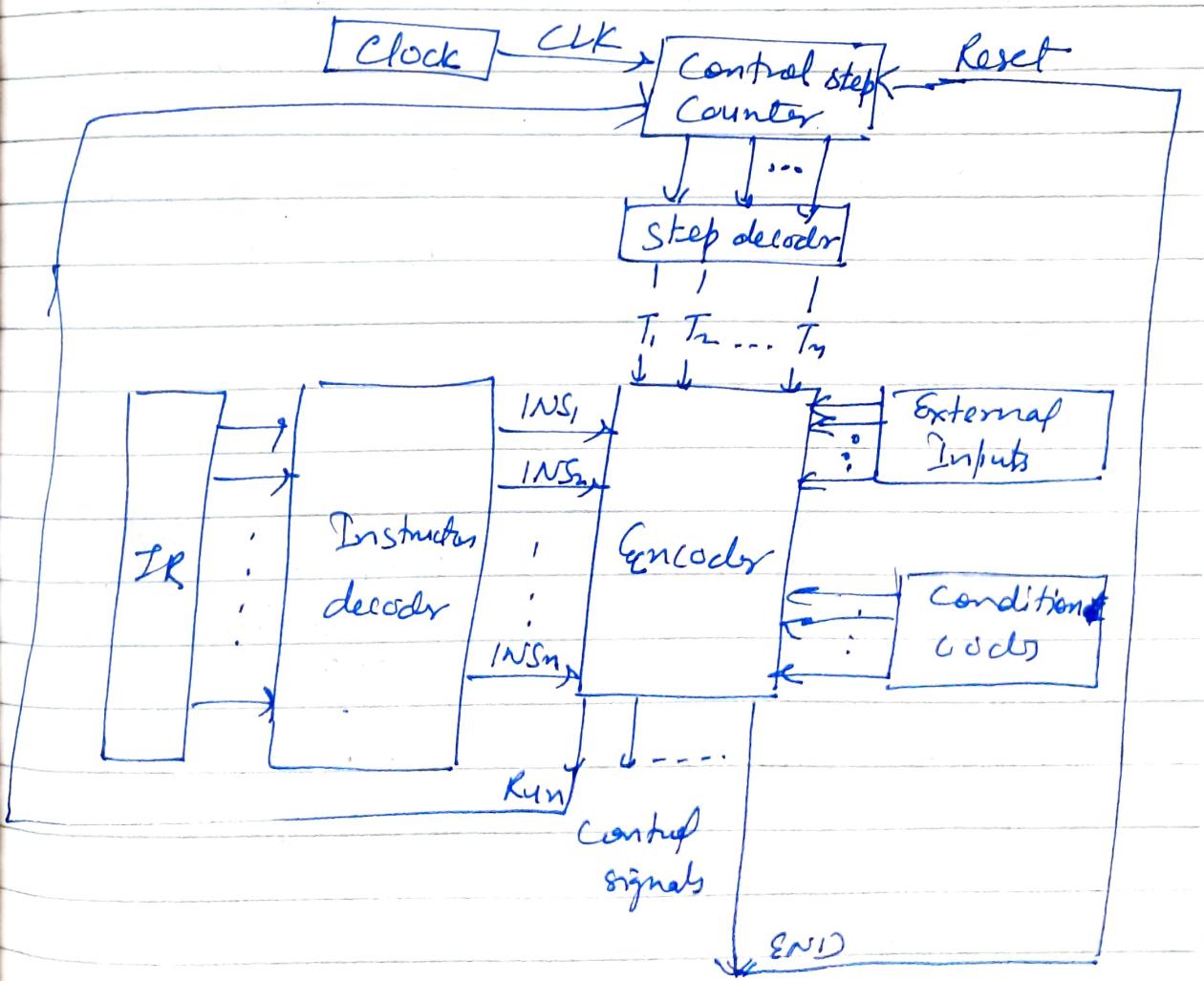
## Lecture-22

The counter keeps track of the control steps.  
The required control signals are determined by

- Contents of control step counter
- Contents of Instruction Registers
- Contents of condition code flags
- External input signals, like MFC and interrupt requests.

### Sequence

Separation of Decoding & Encoding Functions

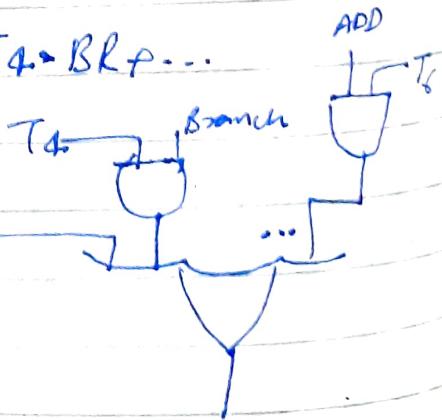


## functioning

- The step decoder provides a set of separate signal line for each step or time slot in the control sequence.
- o/p of the instruction decoder consists of separate line for each machine instruction — for any instruction loaded in the IR, one of the o/p lines  $INS_1$  through  $INS_m$  is set to 1 and all other are set to 0.
- The I/p signals to the encoder block are combined to generate the individual control signals  $Y_{in}$ , PCout, ADD, END etc.
- If  $Rm = 1$ , the counter is incremented by one at end of every clock cycle & if  $Rm = 0$ , the counter stops counting.
  - needed whenever WMEN signal is issued to cause the processor to wait for reply from memory.

e.g.  $Z_{in}$

$$Z_{in} = T_1 + T_6 \cdot ADD + T_4 \cdot BR + \dots$$

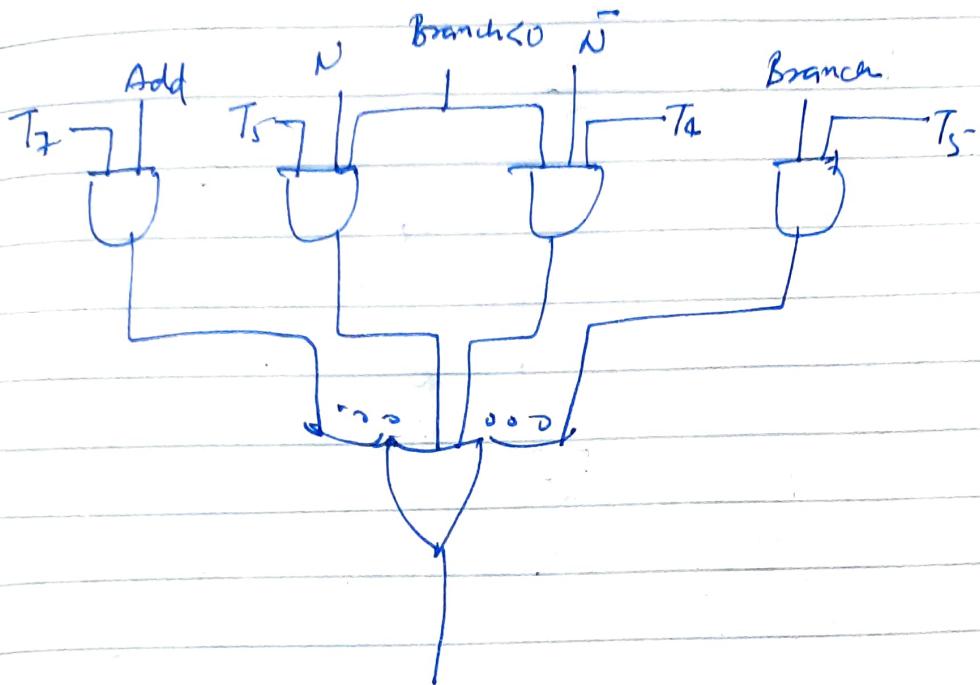


Generation of signal  $Z_{in}$ :

- first step of all instructions  $T_1$  (fetch Instruction)
- step 6 of ADD with register addressing
- step 4 w/ BR
- step 6 of ADD with register indirect addressing

Example : END

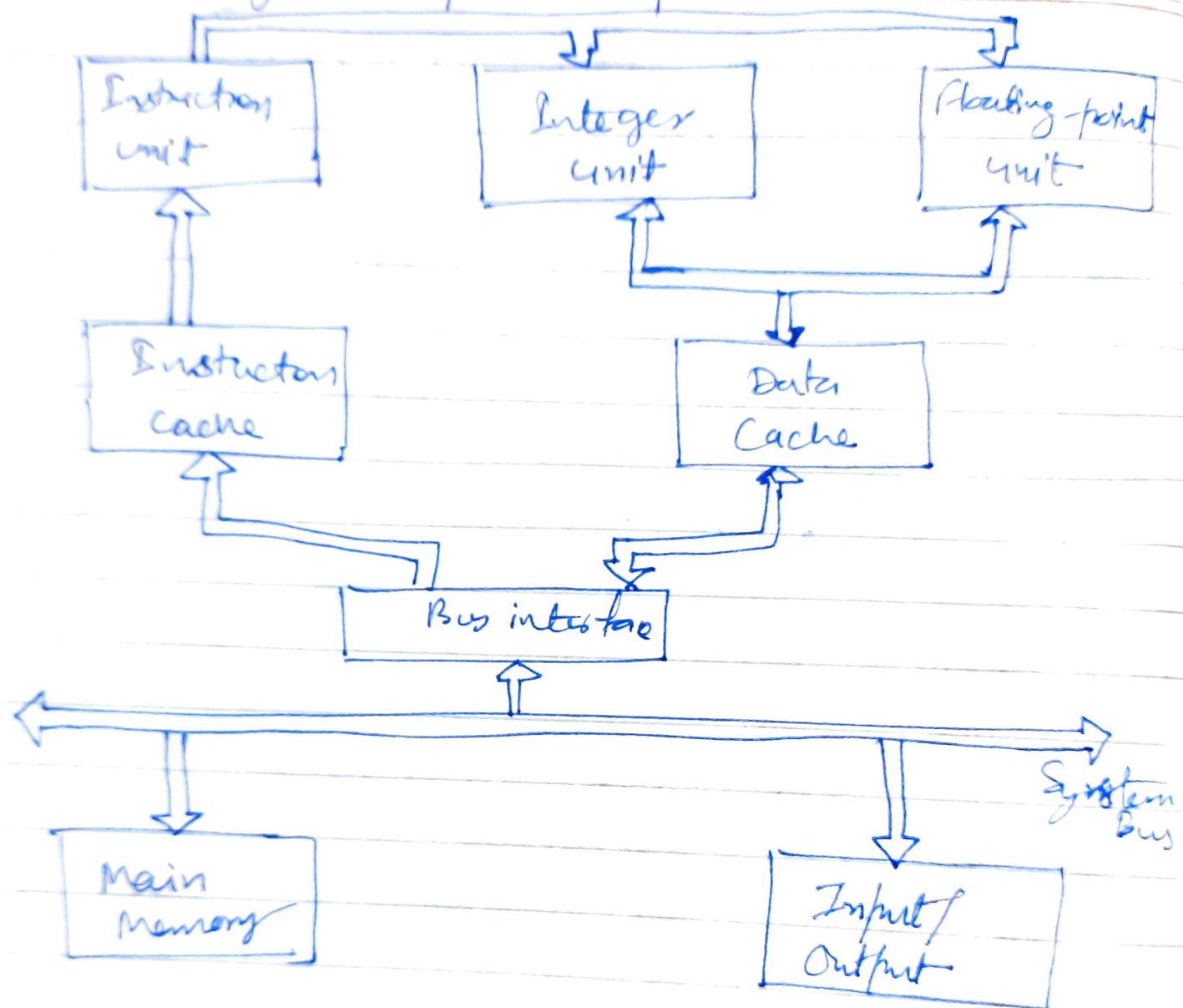
$$END = T_7 \cdot ADD + T_5 \cdot BR + (T_5 \cdot N + T_4 \cdot \bar{N}) \cdot BRV \dots$$

Pros & Cons

- The advantage of hardwired control is that it is very fast.
- The disadvantage is that instruction set and the control logic are directly tied together by special circuits that are complex and difficult to design or modify.
- If someone designs a hardwired computer and later decides to extend the instruction set the physical components in the computer must be changed.
- This is prohibitively expensive because not only must new chips be fabricated but also the old ones must be relocated and replaced.

## Lecture-24

### Block Diagram of a complete Processor



## Micro Programmed Control

### Overview

- The control unit initiates the sequences of microoperations.
- Every microoperation system has a finite number of microinstruction type.
- Complexity of the digital system is derived from the no. of sequences of microoperations that are performed.
- The control function which specifies a micro-operation is a binary variable.
- In a bus organization system, the control signals that specify microoperations are groups of bits that select the paths in multiplexers, decoders and ALU Cnts.
- The control variables at any given time can be represented by a string of 1s & 0s called a control word.
- Control words can be programmed to perform various operations on system components.
- A control unit whose binary control variables are stored in memory is called Microprogrammed control unit.

### Terminology

#### Microprogram:-

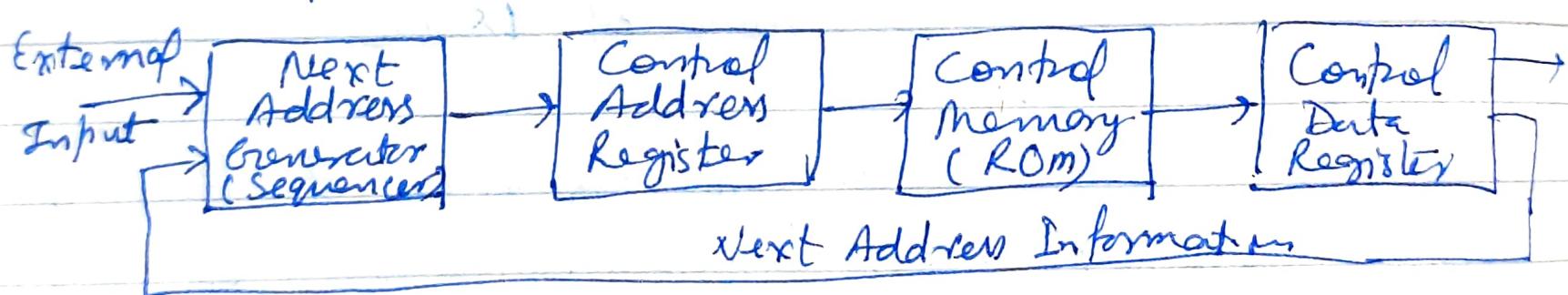
Program stored in memory that generates all the control signals required to execute the instruction set correctly. Consists of microinstructions.

Microinstruction:- Contains a control word & sequencing word.

- Control word → All the control information required for one clock cycle
  - Sequencing word Information needed to decide the next microinstruction address
- Control Memory (CM)
    - (ROM) Storage in the microprogrammed control unit to store the microprogram.
  - Writable Control Memory
    - o Dynamic Microprogramming
      - Computer system where control unit is implemented with a microprogram in writable CM
      - Microprogram can be changed by a systems programmer or a user.

## Lecture-20

### Microprogrammed Control Organizations



#### Microprogram Sequencer

- The next Address Generator is also called the Microprogram sequencer.
- Determines the address sequence to be read from control memory.
- Many ways of specifying addresses depending on sequencer inputs.
- Functions of sequencer include
  - Incrementing control register by one

- Load an address from control memory into CAR
- Transfer an external address
- Load an initial address to start the control operations.

### Control Address Register :-

- Specifies the address of microinstructions
- After execution of the operations specified in the microinstruction, the control must determine the next microinstruction's address.
- The address of next microinstruction is a function of some bits of current instruction and external I/P.
- While the current microinstruction is being executed, the next address is generated in NAG and is transferred to CAR to read the next microinstructions.

### Control Memory :-

- The control memory holds a fixed microprogram.
- The microprogram consists of microinstructions that specify various control signals for execution of register microoperations.
- Each machine instruction in main memory initiates a series of microinstructions in control memory.
- These microinstructions generate the microoperations to
  - Fetch the instructions from main memory
  - Evaluate the effective address

- Execute the operation specified by the instruction
- Return control to fetch phase to repeat the cycle for next instruction.

## Control Data Register

- Holds the present microinstructions while the next address is computed and read from memory
- Also called Pipeline register
  - Allows execution of microinstructions simultaneously with generation of next instruction
- This necessitates a two-phase clock, one clock applied to the address register and the other to the data register each