1. Register Transfers

1) To Execute an Instruction, it involves in a sequence of steps where data gets transfer from 1 register to another register

2) For each Register, two control signals are used to place the contents of that register on the bus

3) The Input and output of registers R_i are connected to the Bus By using which are controlled by signals R_i In and R_i out

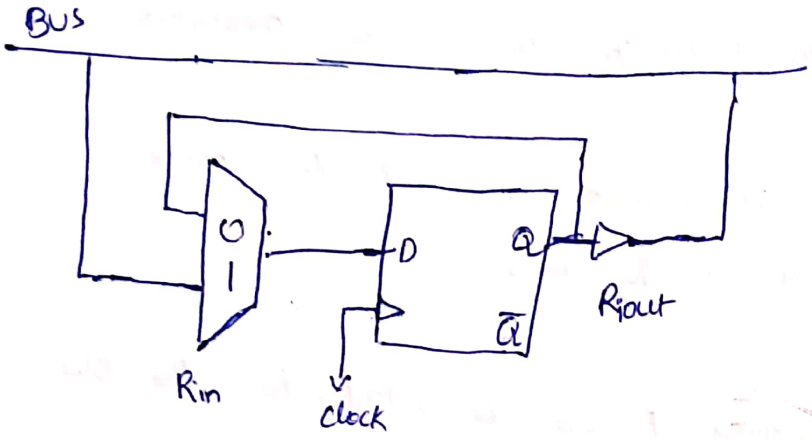
4) when R_i in is set 1, The data on the bus is loaded R_i

5) when R_i out is set 1, The contents of register R_i are placed on the bus

Example:- To transfer the contents of registers R_1 to R_2, we have to follow below steps

1) Enable The output of register R_1 by setting R_1 out to 1
2) This places the content of R_1 on the bus
3) Now Enabling The Input of register R_2 by setting the R_2 In to 1

Register



From The above daigram; Two input multiplexer is used To Select the data deployed to. The input of D. Flip-flo

when The control input Riin = 1. The multiplexer. Selects the data on The bus. and The data is moved to The flip flop based on The clock.

when Riin = 0 The multiplexer feeds back. The value currently store in The flipflop.

2 performing an Arithmetic & Logic operation

ALU:- The ALU is a combinational circuit which performs Arithmetic and Logical operations without internal storagage

*) The ALU perforns Arithmetic ~~and~~ Logical operations on two operands.

*) The Result is prodeced by ALU is stored in a tempeory register

*) To add two values in Two registers, A Seqence of operations are to be performed.

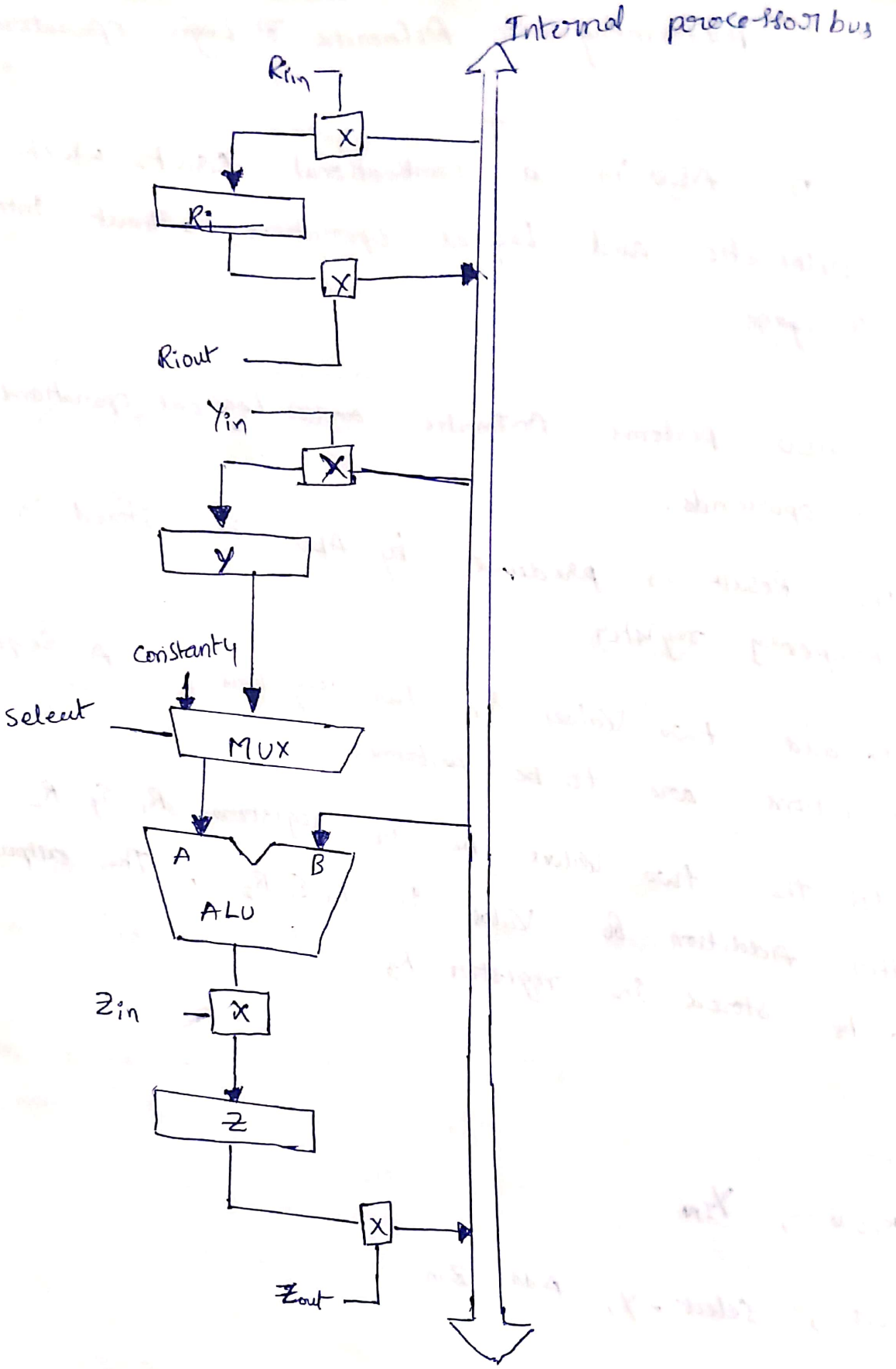
*) Let the two values be in registers R_1 & R_2 and. After Addition of Values in R_1 & R_2, The output is to be stored in register R_3.

Steps:-

(i) R_1 out , Y_{IN}

(ii) R_2 out , Select - Y, Add, Zin

(iii) Zout , R_3 in


Internal processor bus
Rin
X
Ri
X
Riout
Yin
X
Y
Constant4
Select
MUX
A
B
ALU
Zin
X
Z
X
Zout

(i) ~~process~~:-

producer :- In Step1 The output of the register R_1 is Loaded onto the bus and The register Y takes as The ~~The~~ input which is on the bus.

(ii) In Step2, The register R_2 will Load The data into the bus.
~~Here~~ the Register Y is selected to perform the addition operation.

(iii) The addition Operation is performed on register Y value and The Value on the bus.

(iv) The Value on the bus is now stored in register 'Z'

(v) In Step3 The Value in register 'Z' is loaded into the bus and the register R_3 stores The Value from The BUS.

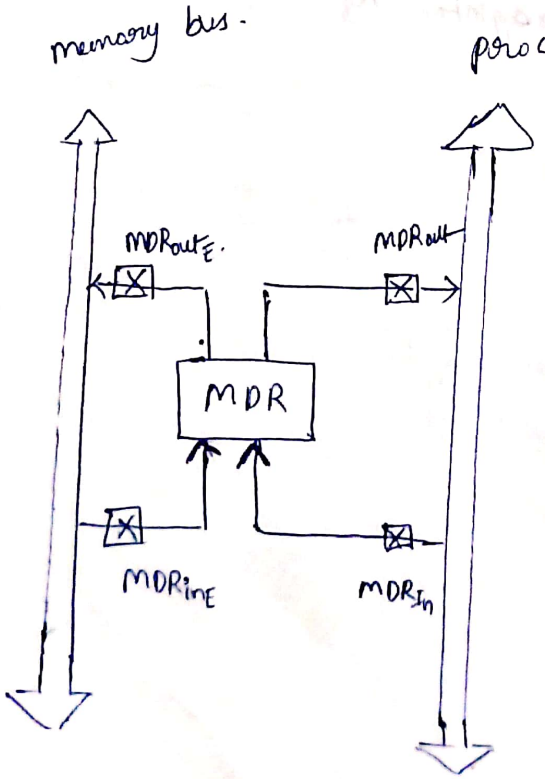
3. Fetching a word from a

*) To fetch a word of information from the memory The processor specifies the address memory location, where The information is stored. It can be done by issuing request to a read operation.

*) To implement, the processor transfers The required Address to 'MAR'

*) The processor uses The control lines of The Memory bus to indicate that a read operation is needed.

*) when The requested data is recived from the memory it is stored in MDR.



Ex:-

a) To perform a read operation, consider the instruction

MOVE (R), R_2.

b) The Actions needed to perform the above instruction are

1. MOVE ← [R_1]
2. Read operation on memory bus.
3. WMFC (wait for memory function complete until its completion.
4. Load the value from the memory to MDR.
5. R_2 ← [MDR]

procedure of Machine instruction :-

a) To load the data from the memory into the processor, the below mentioned instructions gets Executed.

1. R_1OUT, MAR_{In}, Read
2. MDR_{Ein}, WMFC.
3. MDR_{out}, R_2 In.

To store a word in a memory location is similar to fetching a word from a memory.

The operation to be performed is writing a word into a memory location.

Here first, The desired Address is loaded into MAR.

Then, The data which is to be written are loaded into MDR using write Command.

Eg:-

Executing the instruction

MOVE R_2, (R_1) requires following sequence;

1) R_1 out, MAR in

2) R_2 out, MDRin, write.

3) MDRoutE, WMFC.

The write (Command) Control signal causes The memory bus interface hardware to issue a write Command on The memory bus.

The processor remains in step3 until WMFC (The memory operation is completed and an The MFC response is received.).

Execution of a Complete Instruction

(i) Consider an instruction Add (R_3), R_1 which adds the contents of memory location pointed to by (R_3) and the Register value 'R'

(ii) To Executive the above instruction, we need to follow some Actions, They are.

1. Fetch The instruction
2. Fetch The first operand value, if it is in a memory location.
3. perform The Addition operation.
4. LOAD The result into R_1

(iii) The Sequence of control steps to perform the Execution of the above instruction

Add (R_3), R_1

∴ is as follows:

1. PCout MARin Read, Select 4, Add, Zin
2. Zout, PCin Yin WMFC
3. MDRout MARin
4. R_3out, MAR_{in}, Read
5. R_1out, Yin WMFC
6. MDRout, Select Y, Add, Zin
7. Zout R_1in End

*) Executing The above instruction by using branch instruction :-

1. The Branch instruction can reduce the no. of instructions, where the branch instructions replace the PC with branch address.

2. The Branch address is obtained by adding an offset 'x' which is given by the branch instructions.

Ex:-

Control sequence for an unconditional branch instruction.

1. PC_{out} MAR_{in}, Read, Select 4, Add, Z_{in}

2. Z_{out}, PC_{in}, Y_{in} WMFC

3. MDR_{out}, IR_{in}

4. offset-field of IR out, Add, Z_{in}

5. Z_{out}, PC_{in}, End.

5. HARD WIRED CONTROL

*) To Executive instructions, The processor needs control signals in the proper sequence.

*) To generate control signals which follows a proper sequence, we use two approaches, They are.

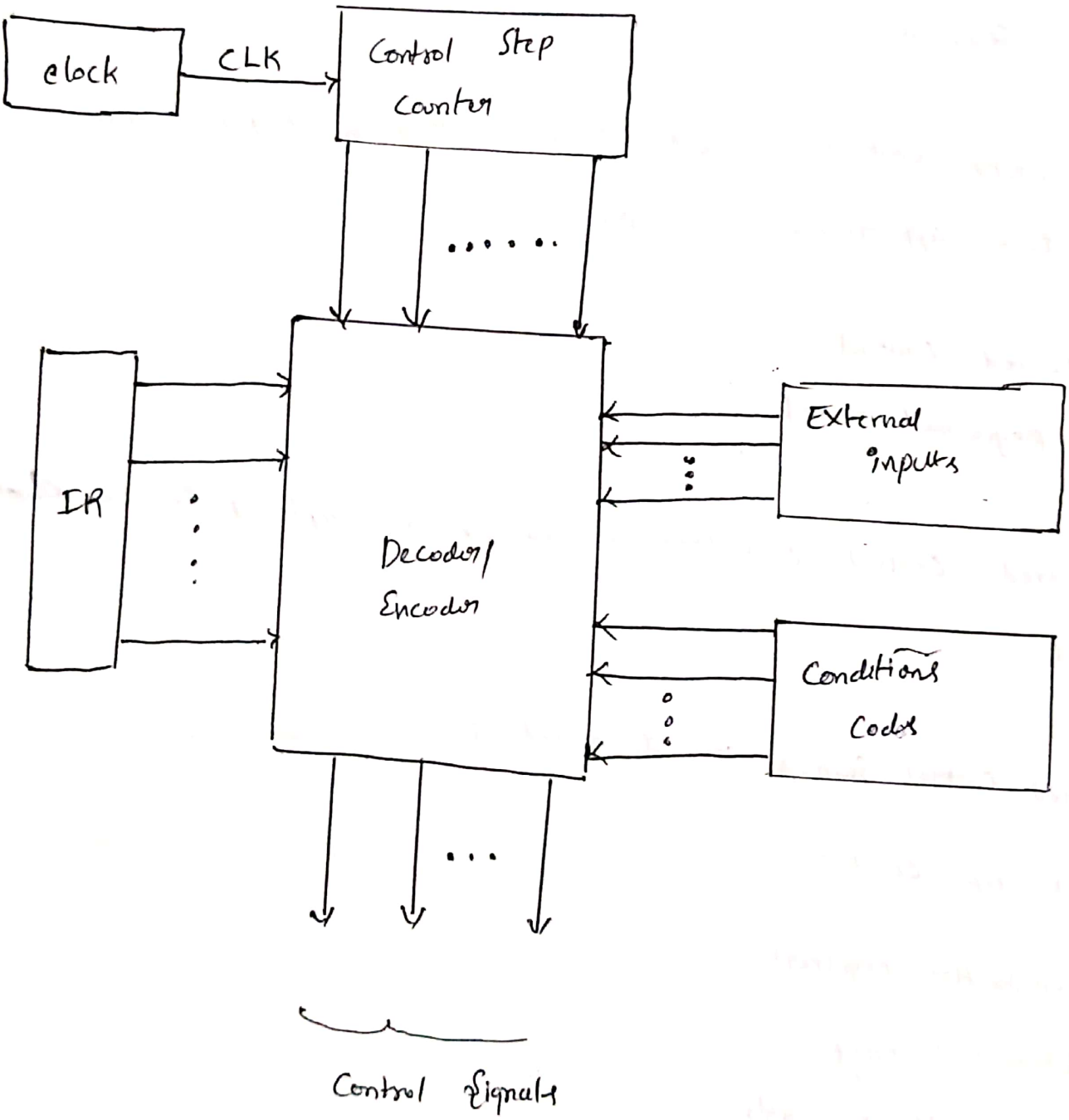
1. Hardwired Control
2. Micro programmed Control

In Hardwired control each step in sequence is completed in one clock period.

The required control signals are determined by;

a. Control Step counter
b. IR (Instruction Register)
c. Condition code flags
d. External ~~Signed~~ input signals

The decoder / Encoder is a Combinational circuit which generates the required control output.


Clock
CLK
Control Step Counter
IR
Decoder/ Encoder
External inputs
Conditions Codes
Control Signals

6. MICRO PROGRAMMED CONTROL

*) In Microprogrammed Control, The control Signals are generated by a programme.

*) The common terms That we use..

*) Control word (CW) :- The individual ~~between~~ bits in word represents. Various control Signals

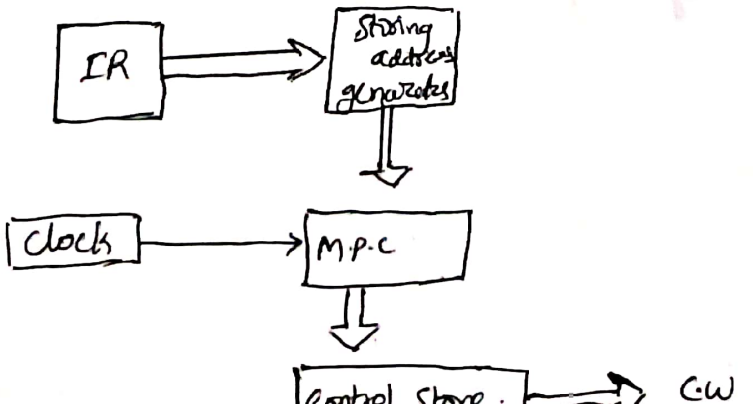
*) Micro Routine:- A Seaurance of cw's Corresponding to The Seauence of a Machine. Instructions constitues. The Micro routine for That instruction.

*) Micro -instruction:- The individual control words in this Micro routine are refered as Micro instructions.

) Control Store:- The Micro routines for all instructions in The instructions Set of a Comp are stuctered in a Special Memory Called Control Store.

) programme Counter:- To read The control words Seauentialy from The Control Store a Micro programme Counter is used.

Basic organization of Micro programmed Control Unit



A Micro programmed control unit which utilizes are Machine Language programme. Constitutes of

a. Micro instructions
b. Micro programme. Sequencing
c. Wide branch addressing.
d. Micro instructions with next address field.

a). Micro Instructions :-

Address micro instructions

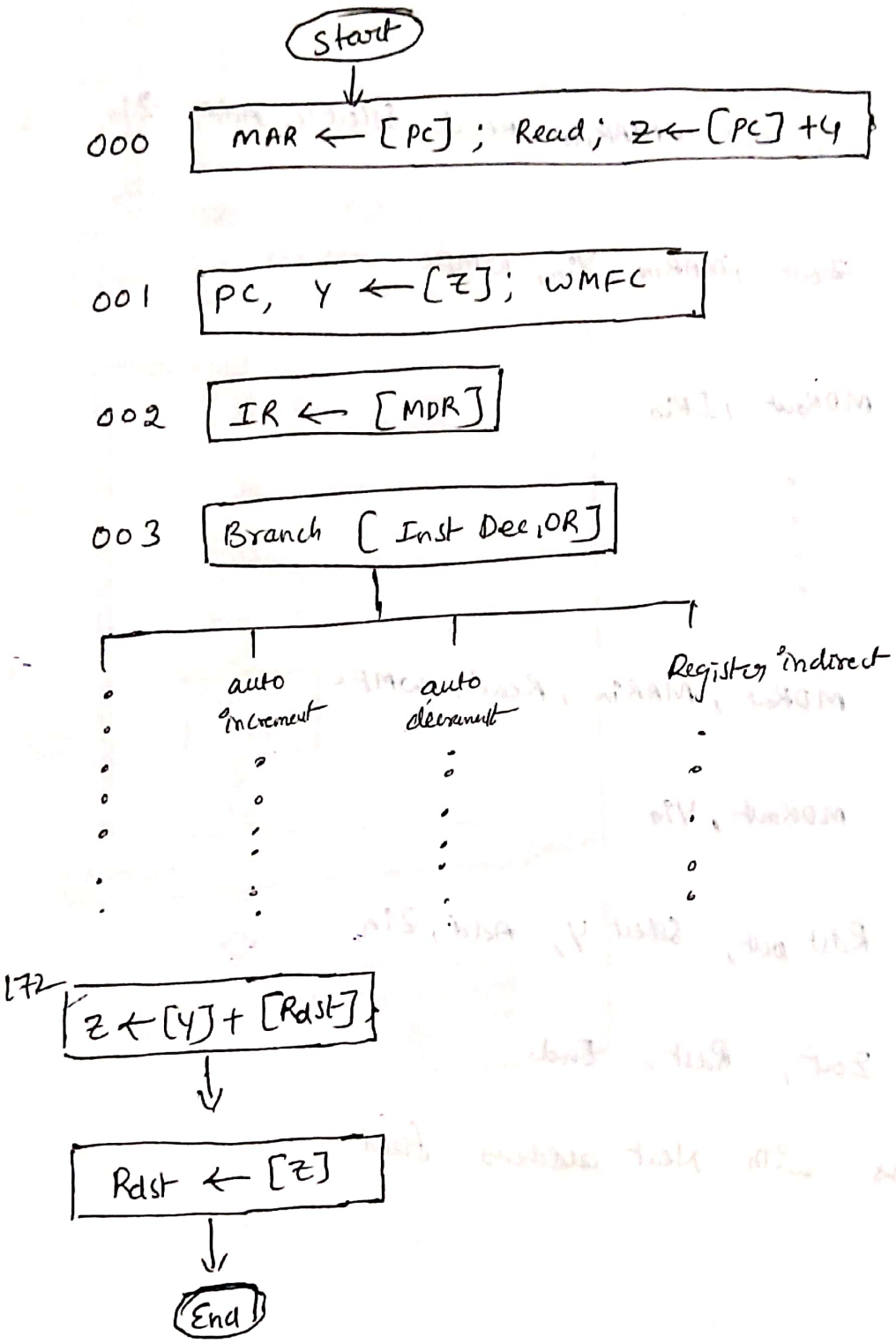
0. PC_{out}, MAR_{IN}, Read, Select4, Add, Zin
1. Zout, PCin, Yin, WMFC
2. MDR_{out}, IRin
3. Branch to address of appropriate micro route.

b) Micro programe Sequence :-

*) Consider an instruction Add Src, Rdst, which address the Source operand to the contents of register. Rdst and places the Sum in Rdst [destination register]

*) Assume That Source Operand is specified in addressing modes like register, auto increment, auto decrement.

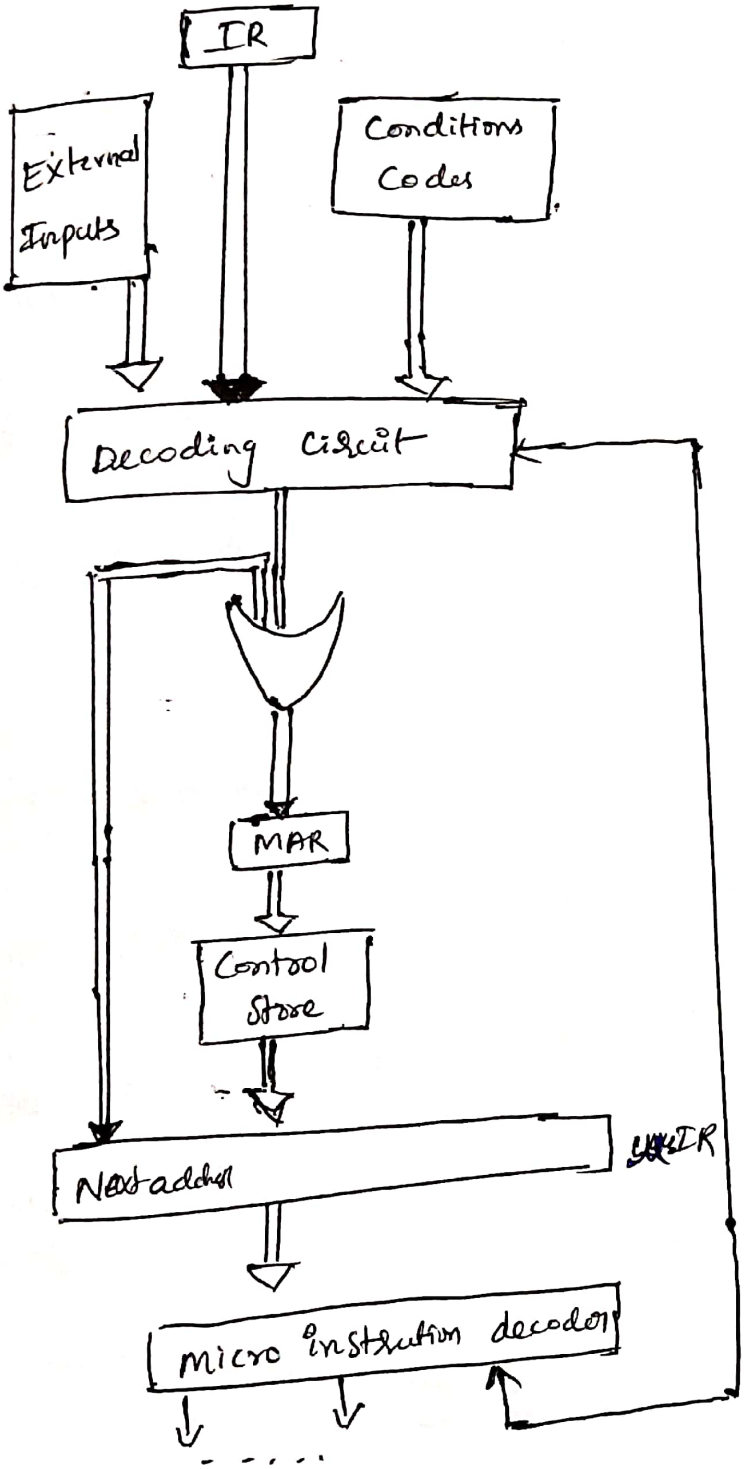
flow chart :-



c. wide - branch Addressing :-

Address (octal)	Micro instructions
000	Pc out, MAR$_{IN}$, Read, Select 4, Add, Zin
001	Zout, MARin, Yin, WMFC.
002	MDRout, IRin
⋮	⋮
170	MDRout, MARin, Read, WMFC.
171	MDRout, Yin
172	Rdst out, Select Y, Add, Zin
173	Zout, Rdst, End.

(d) Micro instructions with Next address field.


IR
External Inputs
Conditions Codes
Decoding Circuit
MAR
Control Store
Next address
IR
Micro instruction decoder