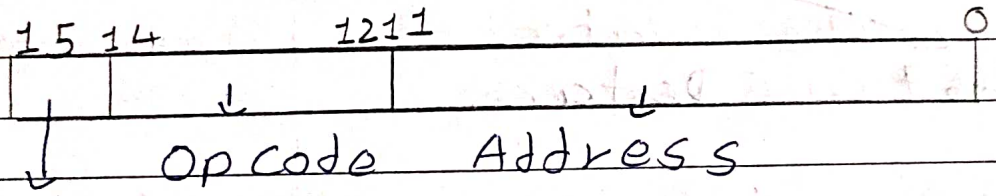
Unit - 2 - Basic Computer Organization and Design

1 Explain Store Program organization using direct and indirect address.

=> Instruction mode divided into three part.



Address mode can be represent in two method.

1 Direct Address
2 Indirect Address

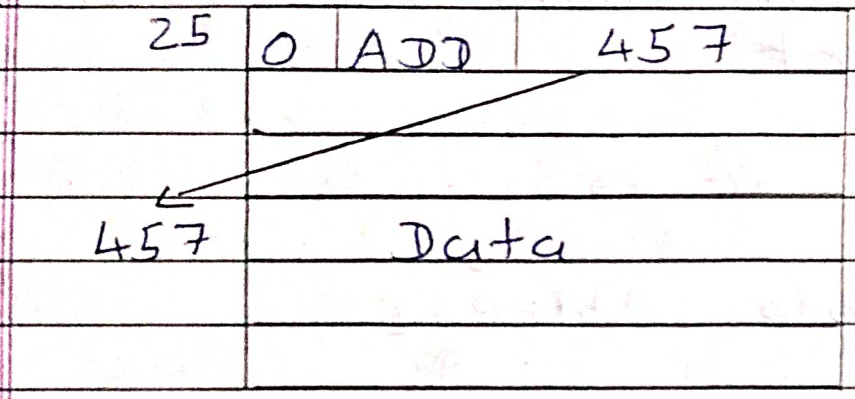
1 Direct Address:

In direct address, instruction 15th bit contain zero bit.

In direct addressing mode the actual address of the data is mentioned in a instruction.

In this addressing mode, direct address is equal to effective address.

Ex.



In this example, Intruction 15th bit is zero, So, we get direct address.

Here, ADD is a operational code and 457 is a address.

At 457 address location we get the data for perform the operation.

2 Indirect Address:

In indirect address, 15th bit of instruction contain 1 bit.

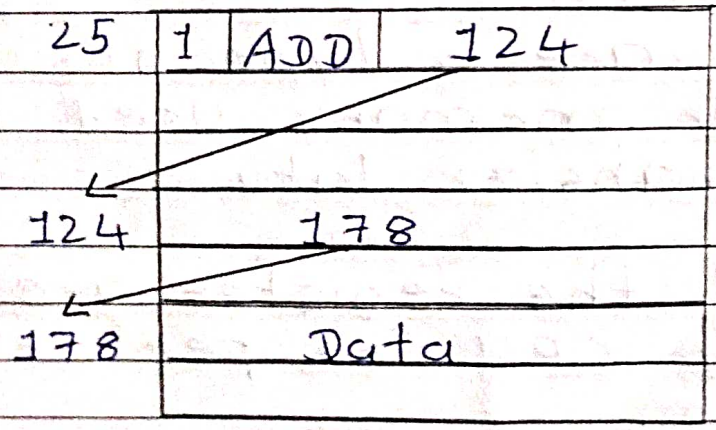
In indirect address mode, the actual address of the data is not placed in ~~the no~~ instruction.

In Indirect address mode we get First address in instruction and after that at First address location we get secound address.

So, somtime we get our data at secound address, either go to next address for data.

In indirect address, when we get our data this address is called effective address.

Ex.



In this example, Instruction 15th bit is one, So, we get indirect address.

Here, ADD is a opcode and 124 is a indirect address.

At 124 address location, we get other address 178.

At 178 address location, we get our data for perform the operation.

Here, 178 is a effective address because at 178 address location we get our actual address of data.

2 Explain Basic Computer Registers.

=> Register are a type of computer memory used to store, transfer data.

This all the register are used by CPU for perform operation.

There are eight computer register.

1 Data Register : DR:

Data Register is a 16 bit of CPU Register.

Data Register is used to holds memory data.

2 Accumulator : AC

Accumulator is a 16 bit of CPU Register.

Accumulator is used to store every operation first operand in the CPU.

3 Instruction Register : IR

Instruction Register is a 16 bit of CPU Register.

Instruction Register is used to hold instruction code in the CPU.

4 Temporary Register : TR

Temporary Register is a 16 bit of CPU Register.

Temporary Register is used to store temporary data.

5 Address Register : AR

Address Register is a 12 bit of CPU Register.

Address Register is used to store address of memory.

6 Program Counter : PC

Program Counter is a 12 bit of CPU Register.

Program Counter is used to store next line address of instruction.

7 Input Register : INPR

Input Register is a 8 bit of CPU Register.

Input Register is used to store input character.

8 Output Register : OUTR

Output Register is 8 bit of CPU Register.

Output Register is used to store output character.

3 Explain and Design of Control unit for computer.

=> Control unit is the part of Computer's CPU which control the processor.

Control unit is responsibility of the execute the instruction.

A control unit is convert input signal to the control signals.

After that compert's processor tell hardware that what operation is perform.

Control unit are implemented by two ways.

1 Hardwired Control:

In Hardwired control signal, control signal are control using hardwired logical circuits.

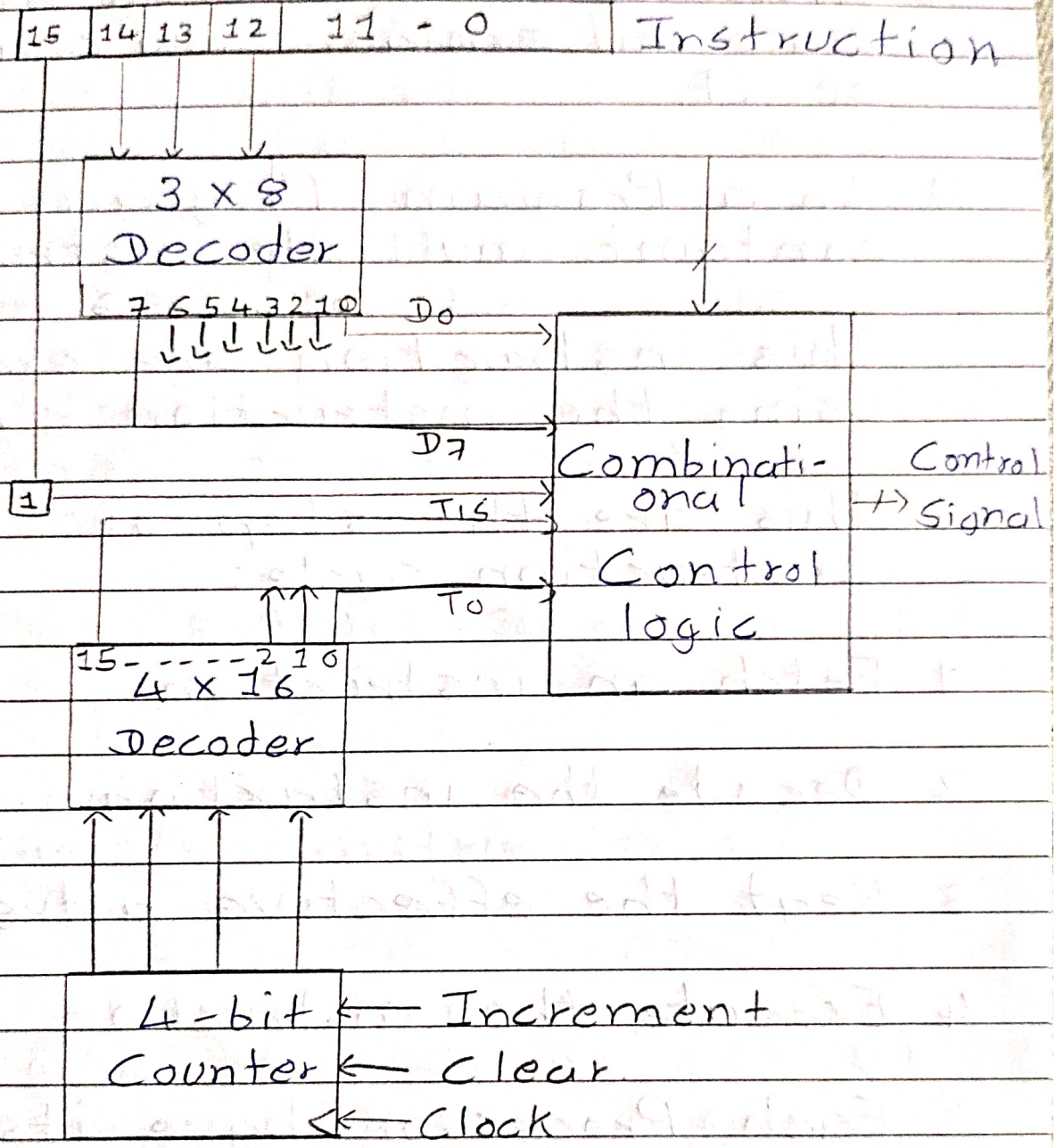
Hardwired control signal is made using Sequential and combinational circuit.

2 Microprogrammable control:

In Microprogrammable control control signal are control using microprams.

Microprograms is active the necessary control signals.

- Diagram of Hardwired Control Unit:


15 14 13 12 11 - 0
Instruction
3 x 8
Decoder
7 6 5 4 3 2 1 0
D0
D7
1
I15
T0
Combinati-
onal
Control
logic
Control
Signal
15 - - - - - 2 1 0
4 x 16
Decoder
4-bit
Counter
Increment
Clear
Clock

4 What is an Instruction Cycle? Draw and explain Flow chart of it.

=> In a Program, Program contaiant multiple instruction.

This instruction are executed using the instruction cycle.

This are the step of Instruction cycle:

1 Fetch an instruction

2 Decode the instruction

3 Read the effective address

4 Execute the instruction.

Each Processor have its own instruction cycle to execute the instruction.

1 Fetch and Decode Cycle:

This are the step of Fetch and Decode Cycle.

Step-1 - The address of Program counter is moved into the Address Register.

$$AR \leftarrow PC$$

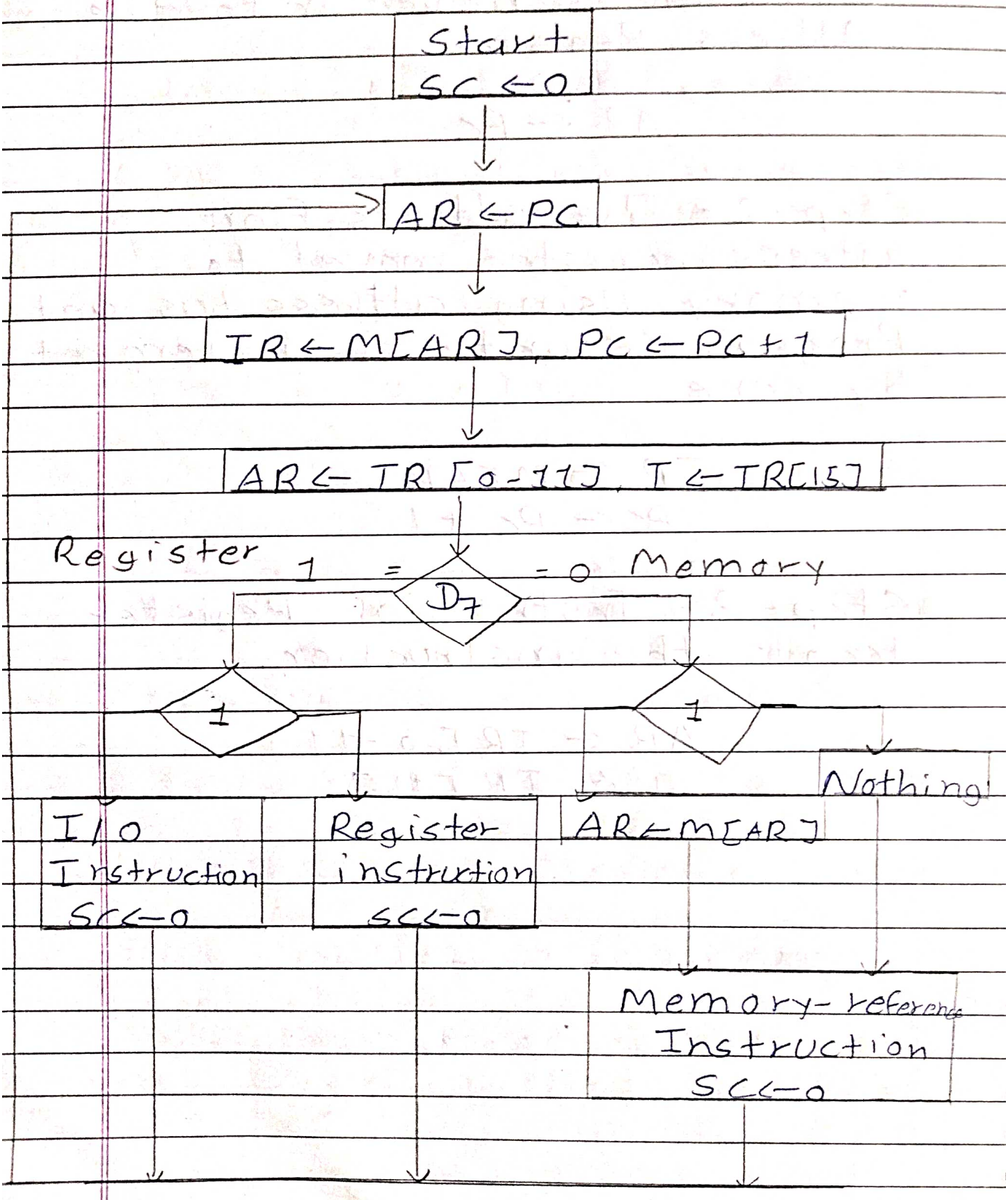
Step-2 - The address From address register moved to memory Using address bus and Program counter is increment by one.

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

Step-3 - Instruction Register decode the instruction.

$$AR \leftarrow IR[0-11]$$
$$I \leftarrow IR[15]$$

→ Flowchart:


Start
SC ← 0
AR ← PC
IR ← M[AR], PC ← PC + 1
AR ← IR [0-11], I ← IR[15]
Register
1
=
D7
=0
Memory
1
1
Nothing
I/O Instruction SC ← 0
Register instruction SC ← 0
AR ← M[AR]
Memory-reference Instruction SC ← 0

5 Explain Memory - Reference instruction

=> Memory Reference instruction are use to perform operation on memory.

Memory Reference instruction use to transfer or store operation data on memory.

This are the seven Memory - Reference instruction.

1 AND:

The AND Memory Reference instruction implements the AND logic operation.

$$DR \leftarrow M[AR]$$
$$AC \leftarrow AC \wedge DR$$

2 ADD:

The ADD Memory Reference instruction implements the addition operation.

DR ← M[AR]
AC ← AC + DR

3 LDA : Load Accmulator.

LDA Memory Reference instruction is use to data in the Accmulator register.

DR ← M[AR]
AC ← DR, SC ← 0

4 STA : Store From Accmulator.

STA memory Reference instruction use to store data From Accmulator to other register.

M[AR] ← AC, SC ← 0

5 BUN : Branch Unconditionally

BUM memory reference instruction use to implement multiple instruction that are not next in the sequence.

PC ← AR, SC ← 0

6 BSA : Branch and Save Return address.

BSA Memory Reference instruction is used to store the address of the instruction from the Program counter into a memory location.

M[AR] ← PC
AR ← AR + 1

7 ISZ : Increment and Skip-if-zero.

ISZ memory reference instruction is used to incremented cost is zero then Program Counter incremented by 1.

DR ← M[AR]
DR ← DR + 1
M[AR] ← DR,
if (DR = 0) then
PC ← PC + 1, SC ← 0

SMVS
Page No.
Date : / /

6 Differentiate LDA and STA Instruction of memory reference

	LDA	STA
1	LDA Stands for Load Accmulator.	STA stands for ~~Stan~~ Store from Accmulator.
2	Use to store data in Accmulator	Use to store Accmulatore data From other location.
3	LDA is copy data from memory location to Accmulator.	STA is copy data from Accmulator to Memory location.
4	LDA is occupies 3 bytes in the memory.	STA is a 2-byte instructi require 3 byte.
5	LDA can store any data from location.	STA can transfer only accomulator data.

7 Explain BSA and ISZ instruction for memory reference.

=> * Write Q-5, BSA and ISZ instruction.

8 Explain Input and Output configuration with suitable block diagram.

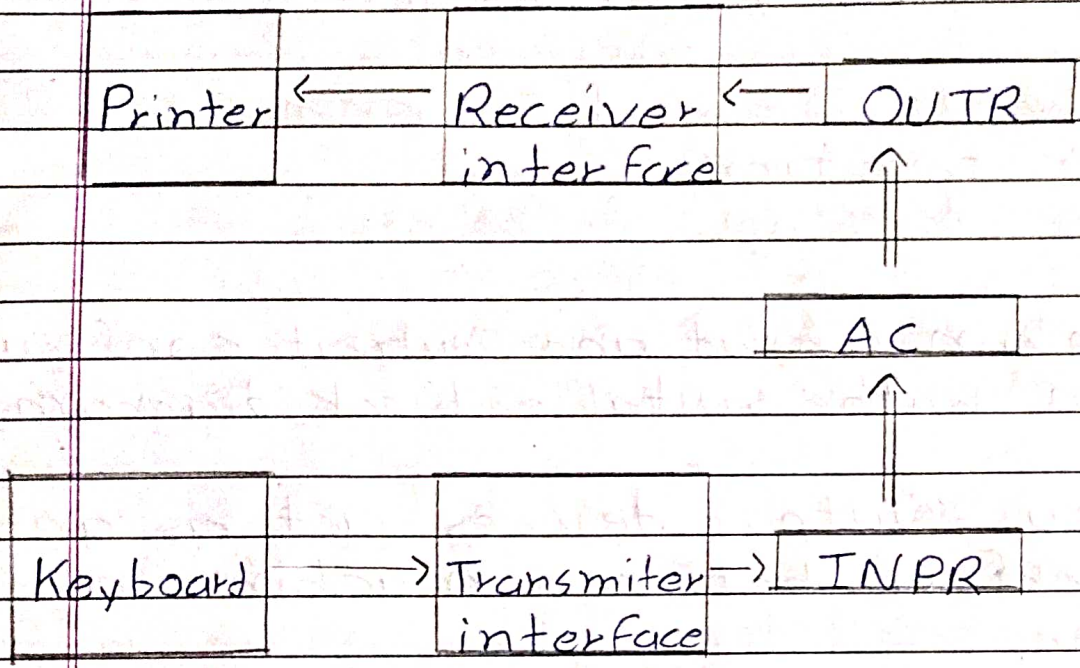
=> Input-Output devices act as an interface between machine and user.

Intructions and data stored in the memory must come from input devices.

The result are displayed to the user the user through some output devices.

Input and Output devices can connect Serial communication and Parallel communication path.

=> Block Diagram:



The input-output terminals send and receive information.

The information generated through keyboard is shifted into input register INPR.

The output of information is store in printer is shifted into output register OUTR.

Transmitter interface receive information through keyboard and transmits it to INPR.

Reciver interface receive information from OUTR and sends to printer.

9 Explain input-output reference instructions.

=> Input-output reference instructions is use to communicate with input-output devices.

Input-output instruction can transfer data to or from the AC register.

This are the basic input-outpu instructions:

~~1 INPR:~~

1 INP: This instruction transfer information from the INPR to AC Register.

This instruction clears the input flag to 0.

AC(0-7) ← INPR, FGI ← 0

2 OUT : This instruction can sed send information from AC to OUTR Register.

This instruction can clears the output flag to 0.

OUTR ← AC(0-7)
FGO ← 0

3 SKI : This instruction use when FGI flag is One.

This instruction can increment Program counter by 1.

If(FGI = 1) then (PC ← PC + 1)

4 SKO : This instruction is use when FGO Flag is one.

This instruction can increment Program counter by 1

If(FGO = 1) then (PC ← PC + 1)

5 ION : This instruction is use to set interrupt.

IEN ← 1

6 I O F : This instruction is use to clear the interrupt

IEN <— 0

10 What is an Interrupt Cycle? Draw and explain with flow chart.

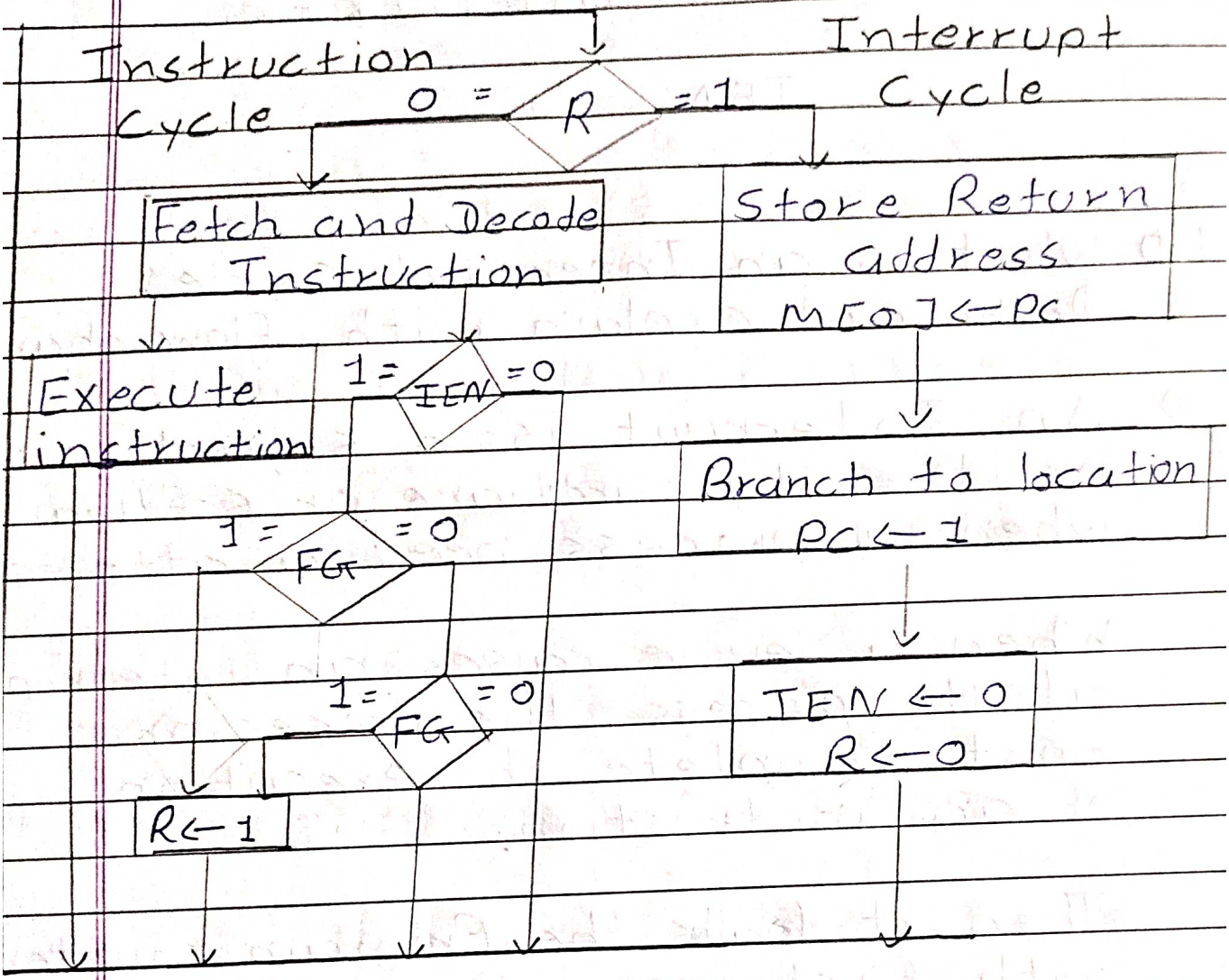
=> An Interrupt is a signal emitted by hardware or software when a process need a attention.

When a device raises an interrupt at the process, the processor first complete the execution of an instruction.

Then it loads the Program Counter with first instruction. After that address of the interrupted instruction is moved to temporary location.

After handling the interrupt the processor can continue with the process.

→ Flowchart:



There are two types of Interrupt

1) Hardware Interrupt

2) Software Interrupt.

SMVS
Page No.
Date : / /

1 Hardware Interrupt:

A hardware interrupt can arise when hardware devices signal may be enable.

Hardware interrupt is generated by external devices.

Hardware interrupt does not increment the program counter.

Does not get higher priority.

2 Software Interrupt:

Software interrupt caused by an instruction in the program

Software interrupt is generated by executing instruction.

Software interrupt does increment the program counter.

Get Higher Priority.

11 List hardware component required to design a basic computer.

=> This are the hardware component required to design a basic computer

1 A Memory unit : 4096 x 16

2 Registers : AR, PC, DR, AC, IR, TR, OUTR, INPR, SC

3 Flip Flops : I, S, E, R, IEN, FGI, FGO

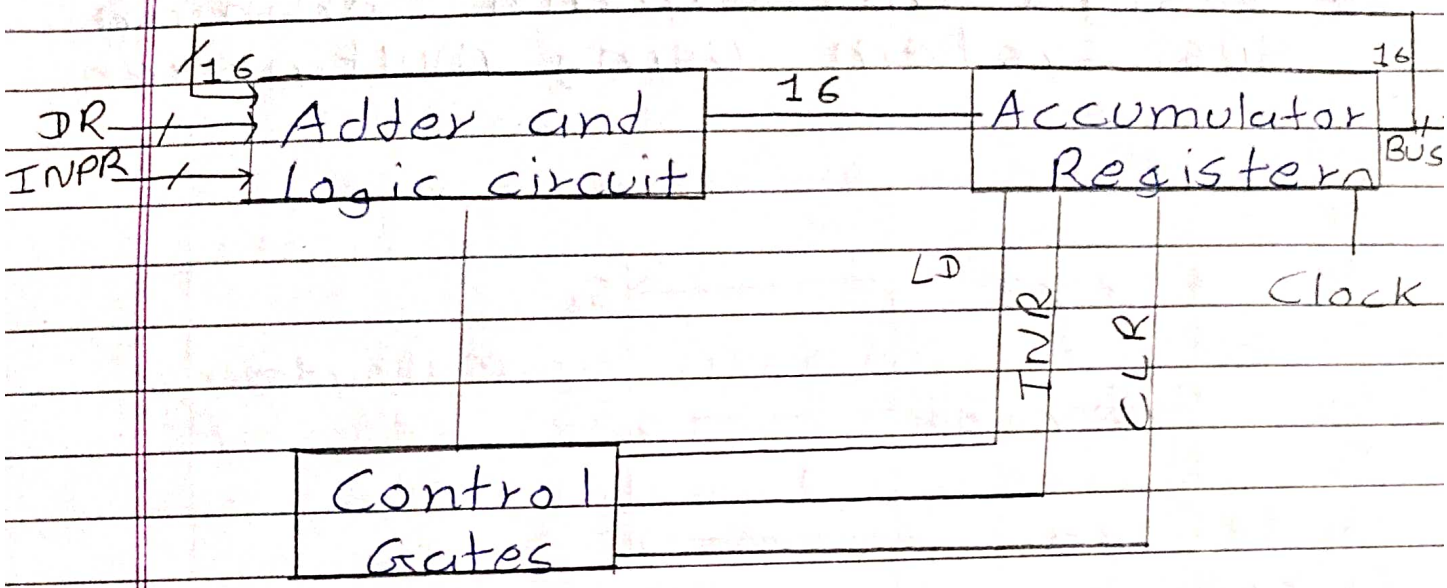
4 Comman bus : 16 bits

5 Control logic Gates

6 Adder and Logic circuit

12 Write a short note on Design of Accumulator logic.

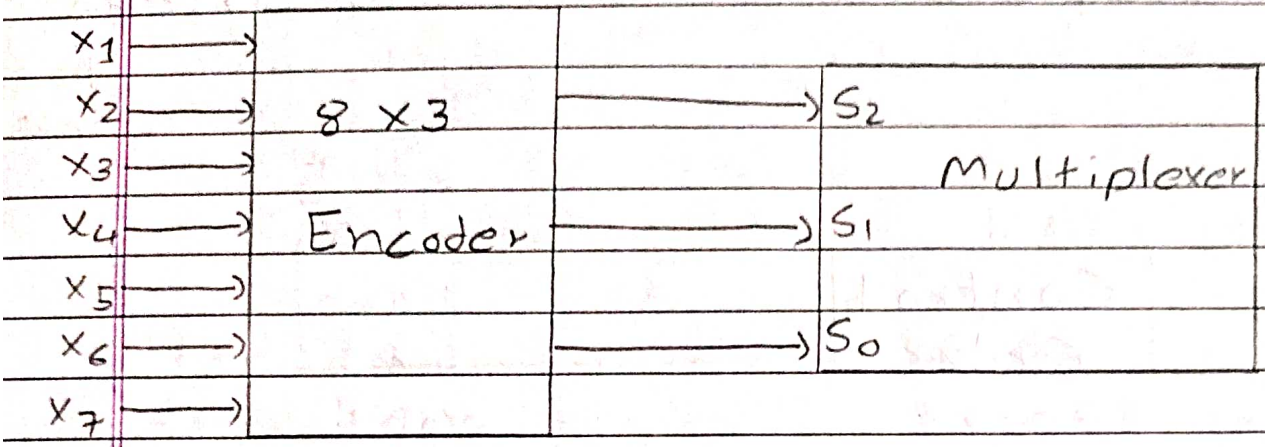
=> Accumulator is special type of register of a Computer System.



Accmulator Circuit Perform this type of operation.

- AND with DR : AC ← AC ∧ DR
- ADD with DR : AC ← AC + DR
- Transfer with DR : AC ← DR
- Transfer From INPR : AC(0-7) ← INP
- Complement : AC ← AC'
- Clear : AC ← 0
- Increment : AC ← AC + 1

13 Design and explain Common Bus system using multiplexer.



x_1	x_2	x_3	x_4	x_5	x_6	x_7	S_2	S_1	S_0	Seleted Register
0	0	0	0	0	0	0	0	0	0	None
1	0	0	0	0	0	0	0	0	1	Address Register
0	1	0	0	0	0	0	0	1	0	Program Counter
0	0	1	0	0	0	0	0	1	1	Data Register
0	0	0	1	0	0	0	1	0	0	Accumulator
0	0	0	0	1	0	0	1	0	1	Instruction Regi.
0	0	0	0	0	1	0	1	1	0	Temorary Regi.
0	0	0	0	0	0	1	1	1	1	memory.

14 Write Short note on Subroutines.

=> Subroutines are programs that are used by other micro program execution.

Subroutines executes the other microprogram in the main program.

Sequence of instruction can called any time the subroutines

Micro-program that use subroutin must have a provision return address of main program.

Provision Return address is follow Stack Organization to perform subroutines.

Subroutine call is implement with the following Operation:

SP ← SP - 1 // Decrement Stack Pointer
M[SP] ← PC // Push PC content onto stack
PC ← effective // Transfer control address of Subroutines.

Ex.

20	0 BSA 135	
21	Next Instruction	
AR = 135	~~135~~	
136	Subroutine ↓	
	1 BUN 135	

20	0 BSA 135	
21	Next Instruction	
	135	21
136	Subroutine ↓	
	1 BUN 135	

Before execution

After execution.

At the execution of 20 address line there are BSA instruction.

BSA instruction call the subroutine at 135 address ~~call~~ and Increment Program Counter by 1.

After the execution at 136 address line, Program back to the main address line.