

diff b/w

Q Computer Architecture and Computer organization

i

parameters | Computer Architecture | Computer organization

rs,
ce
gic

r,

o

Work

What the computer do

How the computer will do.

Design issues

Design issues

Deal with low level design issue

High level design issue

Attribute

Designing ~~CA~~ is attribute of Computer Architecture.

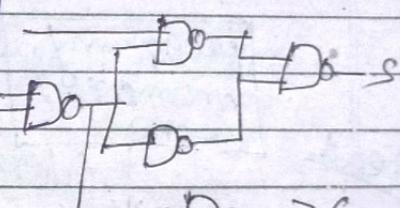
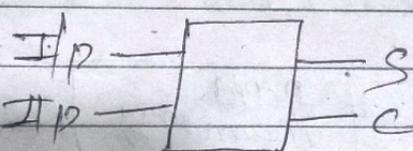
utilization of the resource in proper and planned manner.

what involves

CA involves instruction sets, instruction codes and modes.

CA organization involves circuit design, ALU, CPU de

Example



View

It provides External view of the system

It provides internal view of the system.

* CA and DO provide us with systematic approach to deriving solution to any problem.

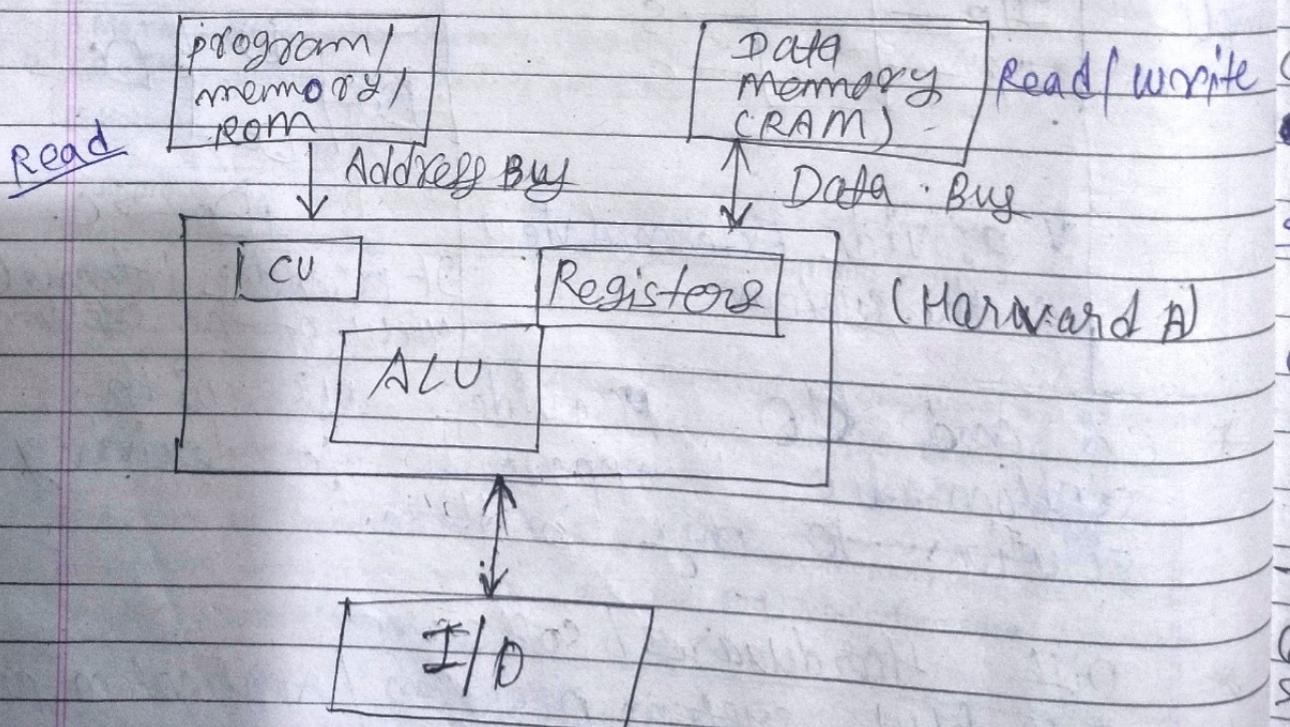
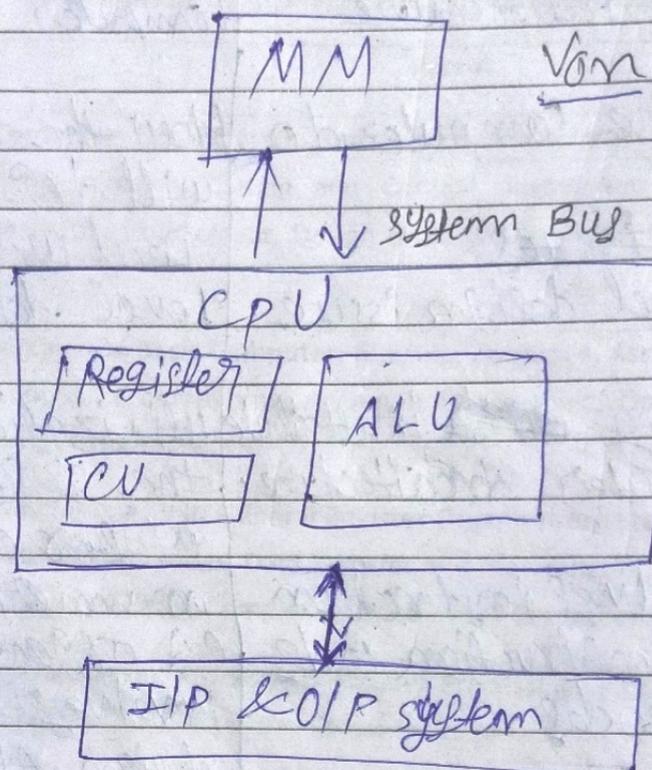
* Diff Hardware / software

* Diff b/w system program / Application program

* what is CA and CO

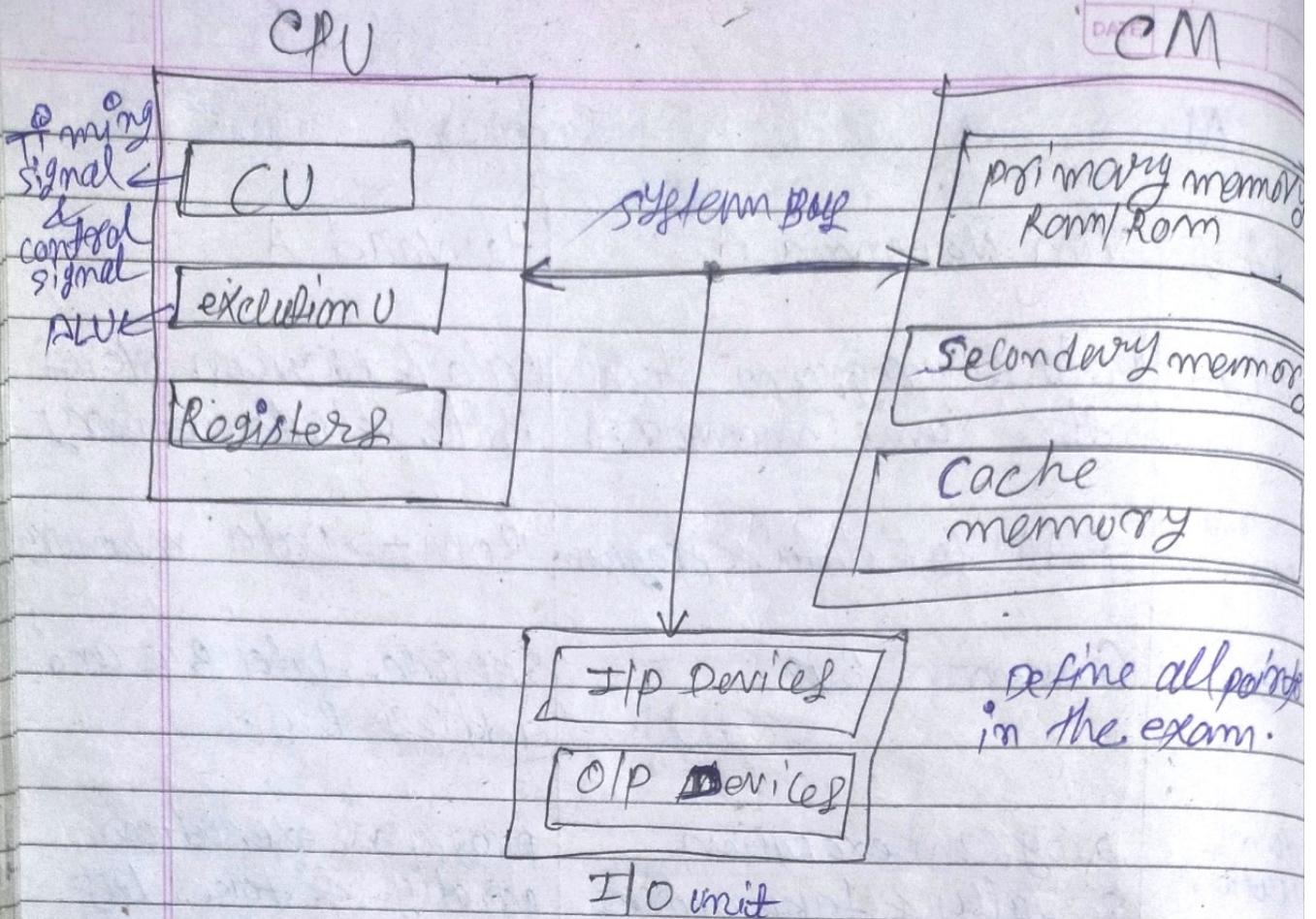
Diff b/w

2m VON Neumann Architecture and Harvard Architecture



M.

	Von Neumann A	Harvard A
Memory	Data & program stored in the same memory	Data & program stored in the separate memory
Memory type	RAM for data & program	ROM → Data memory
Bus	Common Bus	Separate data Bus and address Bus.
Program execution	program execution serially & take more cycles.	program execution parallel & take less cycle.
Machine	Machine takes more machine cycle	Machine takes less machine cycle.
Control signal	less control signal	more control
Space	less space	more space
Cost	less cost	more cost
Ex →	Digital signal processors	Modern microcontrollers
Type of Control Signals	Memory read read Memory write	DATA → Memory Read & W Program → memory Read



* why we are using register if we have main memory.

Ans ⇒ main memory:

Registers : fastest memory available till yet, it stores temporary data, it is smallest size memory 8 bits/1 byte. It is sequence of bits or sequence of flipflop, it stores intermediate or temporary results.

→ Data is present in the main memory and operation is performed on CPU that is ALU (Speed has exec speed is very fast).

main memory comparable to ALU is very slow.

↓
Speed mismatch

↓
Burden → to overcome instead

register

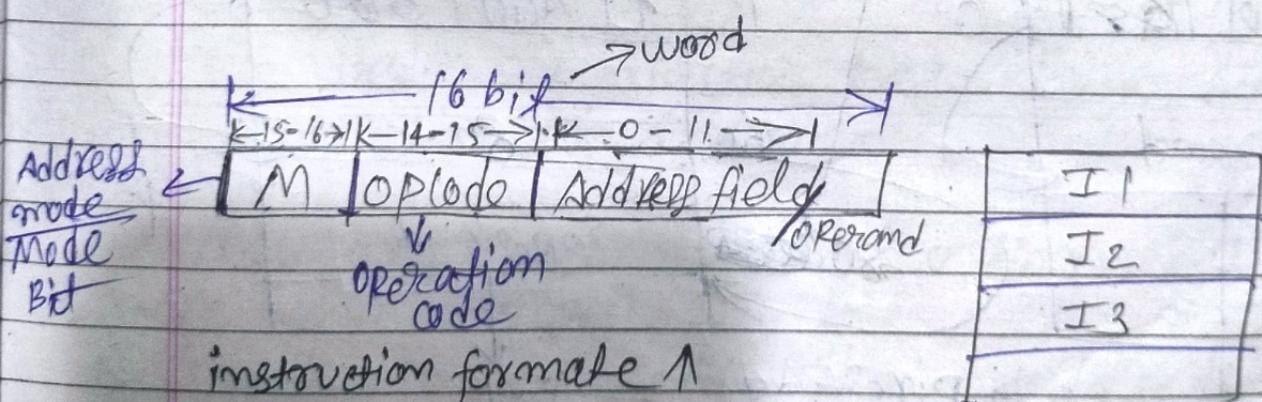
- Control unit
 - ① → Timing signal
 - ② → Control signal

i) Timing signal: it will let you know which instruction will execute first.

ii) Control signal: it will control, handle multiple registers.

Computer Instructions

* Instruction: Instructions comprises of groups called as fields, field include



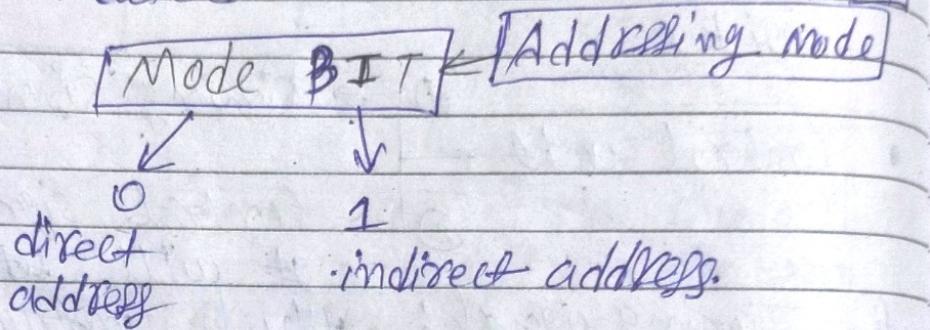
WORD: A 16 bit instruction is called as WORD"

* OPCODE: It specifies to the what operation to performed.

$$\text{OPCODE} \neq 111 = 3 \quad \text{Ex} \Rightarrow 2 + 3 \quad \hookrightarrow \text{OPCODE } (0000 - 110)$$

Address field : It contains the location of operand that is register/memory.

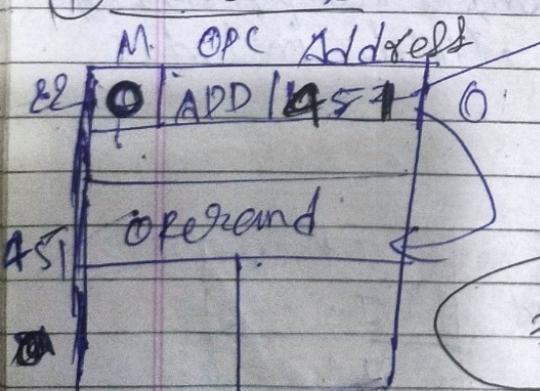
* Mode Bit : M specifies how the operand will be located.



* Instruction Code :

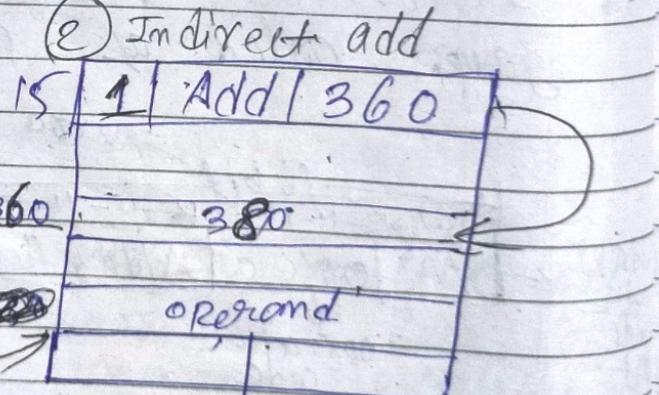
There are three addressing modes used for address portion of instruction code:

① Direct add



0 → Direct Ad
1 → indirect

② Indirect add



360 → 380

(III)

Immediate operand

ADD / Operand | Ex - 9 + 5 ;

~~Q~~ What is effective address

PAGE No.	
DATE	

J/MY

* What is inside in effecive address.

ans

In effective address there is target address of the operand.

* How to write a instruction

ans

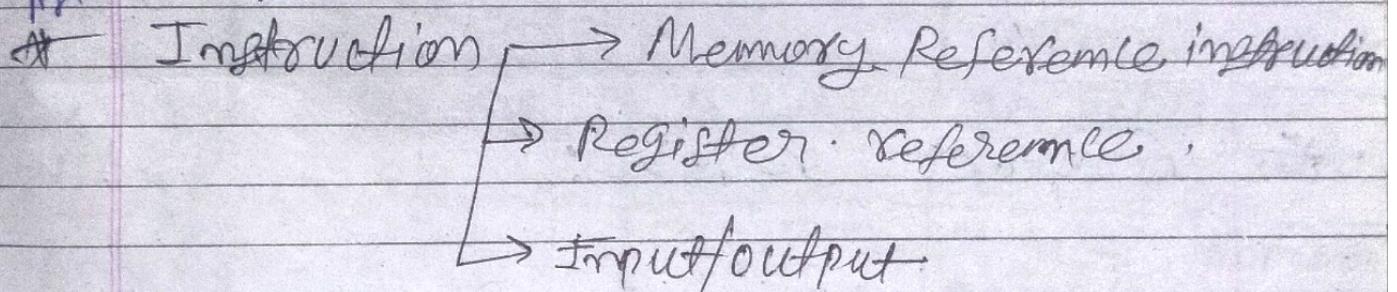
$$AC \leftarrow AC + M[AR]$$

✓

Accumulator memory address of address register

NOT

~~ANSWER~~



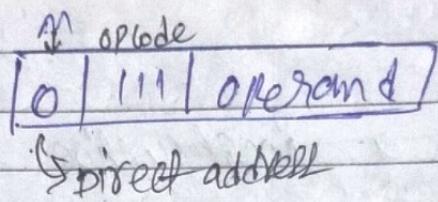
A basic Computer has three types of instruction code formats

i) Memory reference instruction : In memory reference instruction 12 bits of memory is used to specify the address and 1 bit is used to specify the addressing mode I.

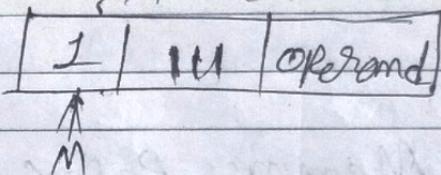
$0,1 \quad (0000-110)$
[M | 111x | operan]

ii) Register reference instruction : The register reference instruction is represented by

the opcode 111 with 0 in the left most bit with the 0 in the left bit of the instruction

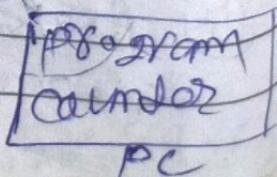
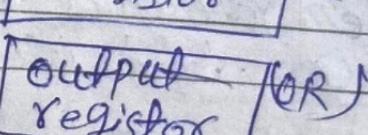
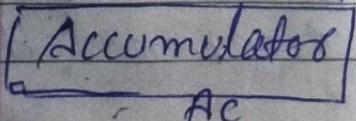
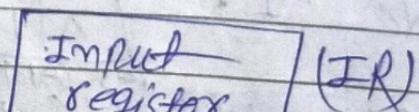
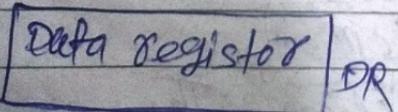
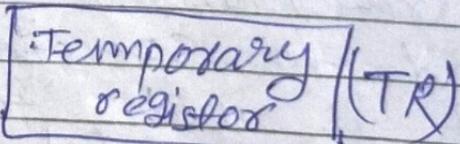
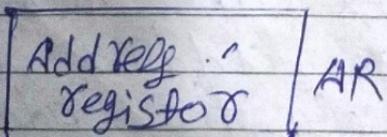
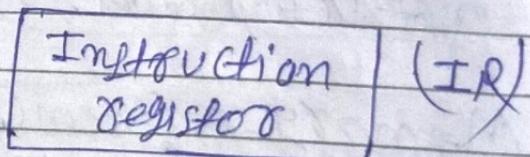
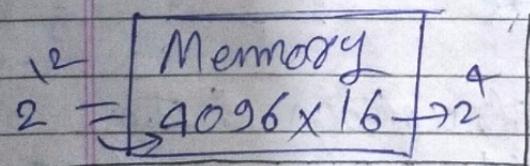


(iii) I/P : The I/P is represented by ~~as~~ the opcode to be 111 and the 1 in the left most bit.
 ↓ indirect address



* Computer register :

Types of Computer register



Reg.name	Reg-symbol	No of Bits	function
Address register	AR	12 Bits (0-11)	It stores the address of operand
Data register	DR	16 Bits (0-15)	It provides data whole instruction
Accumulator	AC	16 Bits (0-15)	It holds the intermediate data
program Counter	PC	12 Bits (0-11)	It stores address of the instruction
Instruction register	IR	16 Bits (0-15)	It stores the complete instruction format
Temporary register	TR	16 bits	It holds the temporary data
Input register	IR	minimum 8 bit max of per requirement	It takes information from the user
Output register	OR	minimum(3) max of per requirement	It prints the output on the monitor, printer

Q. (Remarks): Write a program to add two bit no, where sum is 16 bits.

PAGE NO.	
DATE	

Address	Mnemonic code	opcode/data address	Mnemonic	Comments
9000	3E	opcode.	MVI A 02H	Moves the data to be inputted to AC.
9001	02	Data	—	The data entered here will be inputted to the AC.
9002	06	opcode.	MVI B, 02H	This moves the inputted data to register B.
9003	02	Data	—	Data entered here will be stored in register B.
9004	80	opcode.	ADD B	A = A + B adds contents of ACC and B then stores it in Acc.
9005	32	opcode	STA 9502	Stores the contents of Acc to specific memory location.
9006	02	Address L	—	Address of lower Bits
9007	95	Address H	—	Address Higher Bits
9008	76	opcode	HLT	Stops the program.

Explain :

Step-1 : Press RESET

Step-2 : Press Exec menu

Step 3 : insert 9000 address to move the data

~~Step 4~~ to AC and press next.

Step 5 : put the data⁽⁰⁷⁾ which you want to perform operating at 9001 address this data will

be go to the AC. and press next. (07)

Step 6 : at 9002 this moves the second data to register B.

at

Step 6 : at 9003 enter second data that will be stored in register B.

Step 7 : at 9003 chose the Mnemonic code what do you want to perform with that data for exam addition enter ~~80~~ Mnemonic code 80 for subtraction enter 90. After that this will add content of AC and B and again stores in AC and press next.

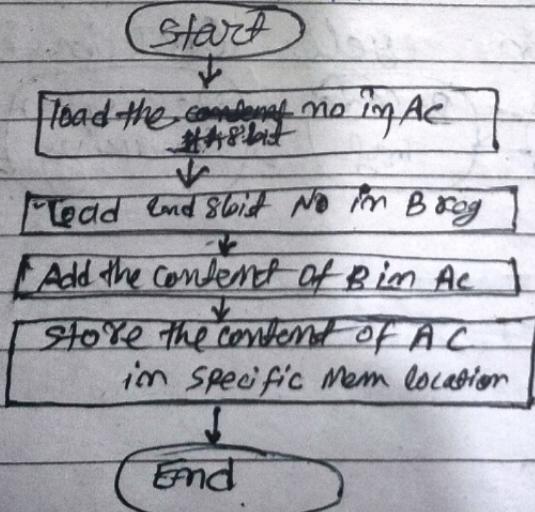
Step 8 : at 9005 ~~enter address~~ where you wants to store result at specific memory location then press next.

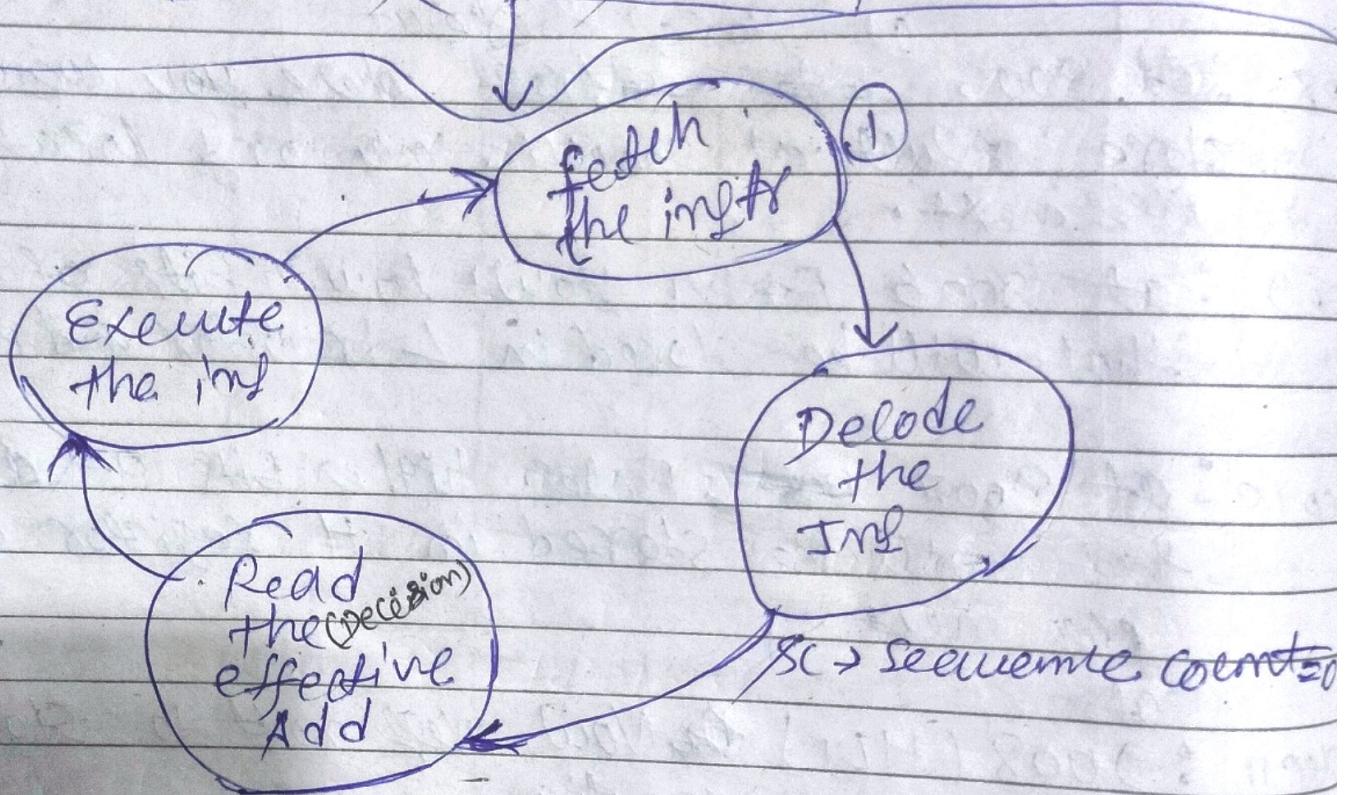
Step 9 : at 9006 Enter your lower bits of address that will be stored in L register and press next.

Step 10 : at 9007 ~~press~~ Enter higher bits of address that will be stored in H register and press next.

at

Step 11 : 9008 (HLT) Now press next to stop the program execution.



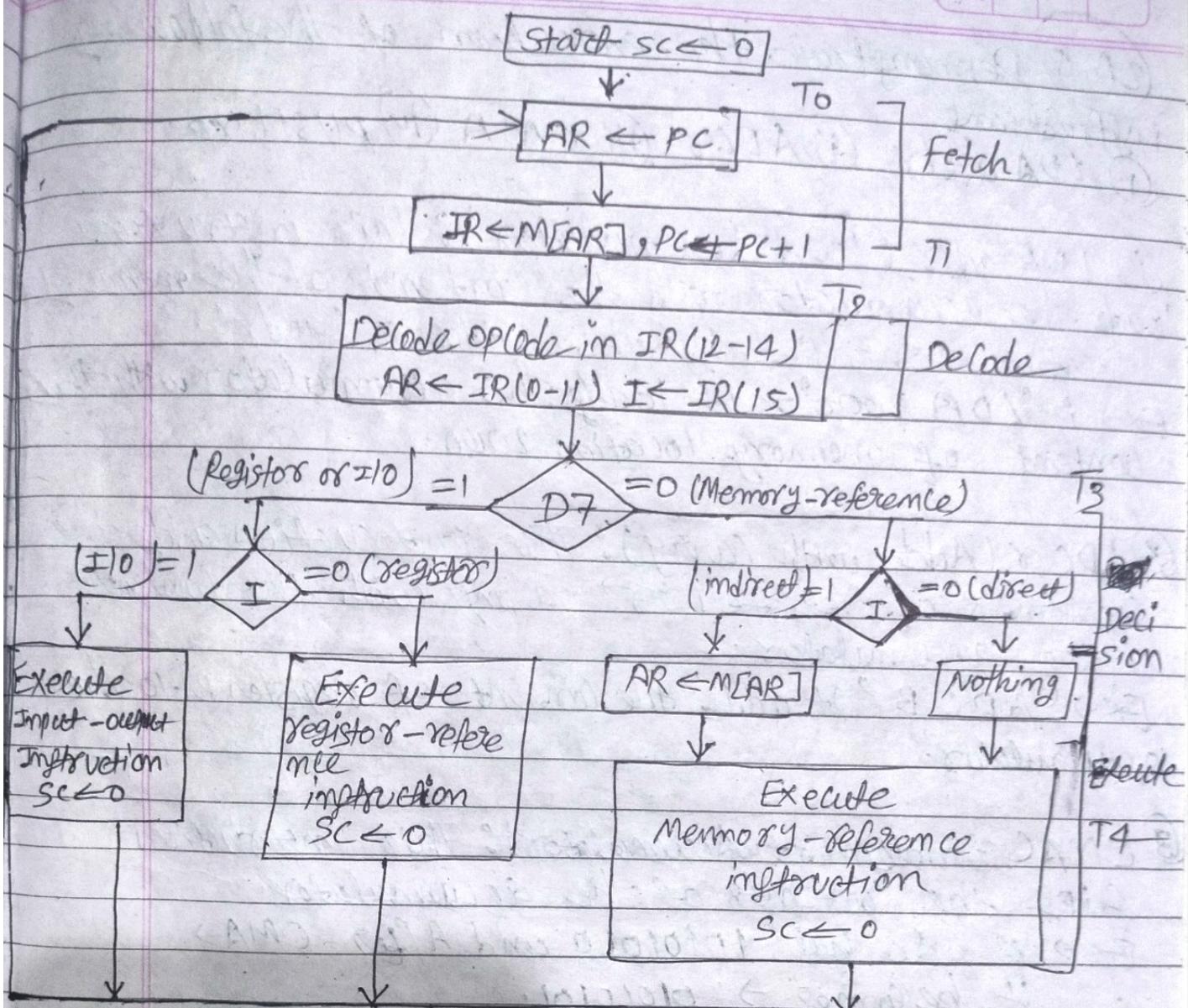


Instruction cycle Contains some steps

Fetch Img → Decode Img → Decision making → Execute Img

Instruction cycle

PAGE NO.	
DATE	



Step 1: Start $SC \leftarrow 0$

Step 2: $T_0 : AR \leftarrow PC$

↳ fetch from memory

Step 3: $T_1 : IR \leftarrow M[AR], PC \leftarrow PC + 1$

Step 4: $T_2 : \begin{array}{l} \text{decode opcode } IR(12-14) \\ AR \leftarrow IR(0-11), I \leftarrow IR(15) \end{array}$

Memory reference
register reference instruction
I/O reference instruction

Step 5: $T_3 : \begin{array}{l} \text{decision} \\ \xrightarrow{\text{indirect}} \\ \xrightarrow{\text{PF}} \xrightarrow{\text{direct}} T_3 \text{ cycle} \end{array}$

Step 6: $T_4 : \text{execute}$

(2M) Demonstrate the execution of the following instructions
 ① LDA addr ② ADCx ③ CMA ④ PUSH xp.

① LDA addr (Load Accumulator): This instruction loads the accumulator with the content of the memory location specified by the address 'addr'.
 Ex → if 'LDA 2000' → it loads the accumulator with content of memory location 2000.

② ADCx (Add with carry): This instruction adds the contents of register 'x' along with the content of the accumulator.
 Ex → 'ADC B' it adds the content of register B to the accumulator.

③ CMA (Complement Accumulator): This instruction flips the contents of the accumulator.
 Ex → if Ac has 10101010 and After CMA → it becomes → 01010101.

④ PUSH xp (Push register pair onto the stack):
 This instruction pushes the content of the register pair 'xp' onto the stack.
 Ex → if 'PUSH BC' it pushes the content of registers B and C onto the stack.

fetch cycle

- start sequence counter to zero before fetching.
- and timing pulse start from T0 during T0 the instruction which are present in program counter place that instruction into address register.
- After that program counter will be increased by 1.

Decode cycle

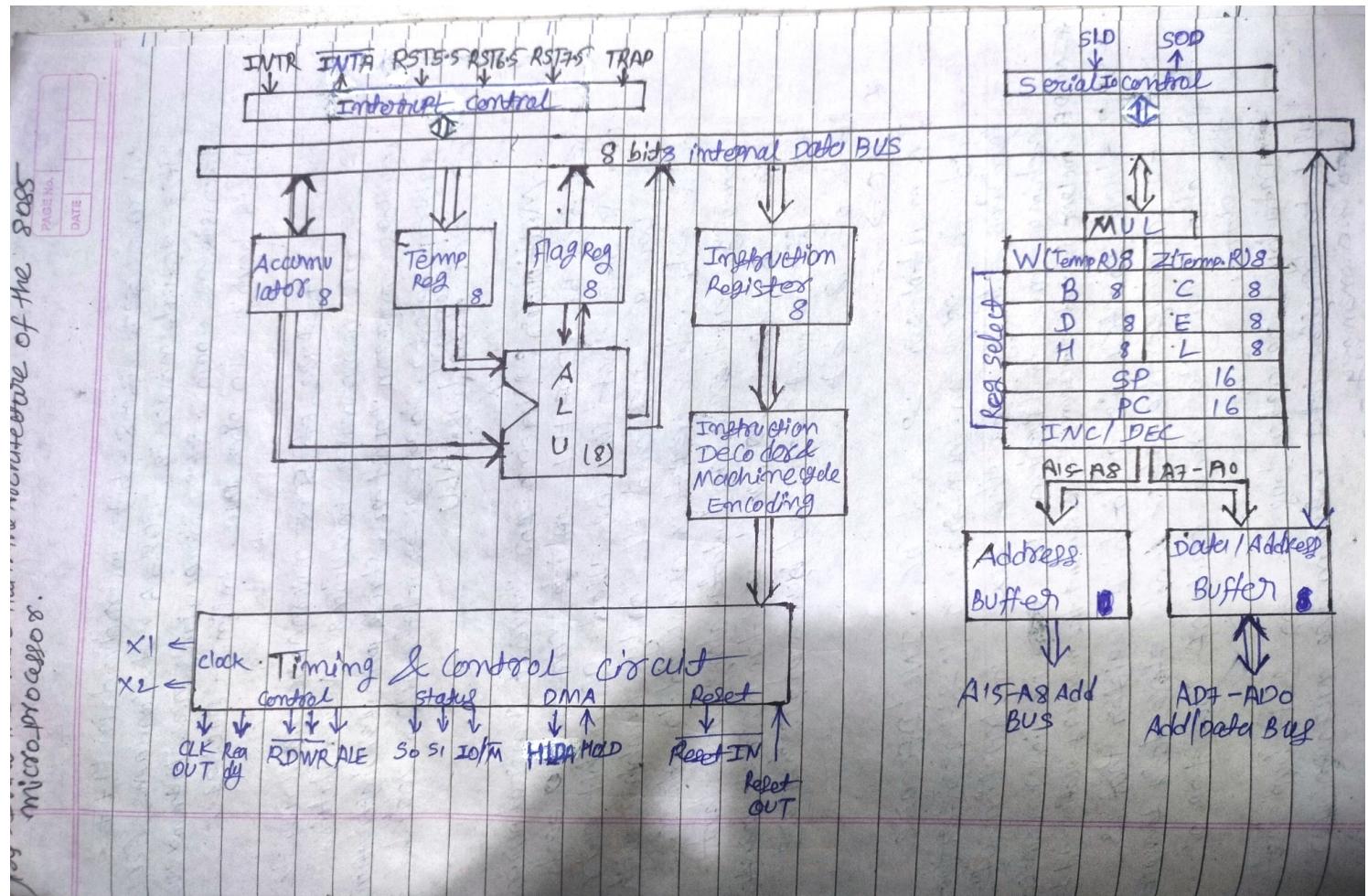
- After fetching the instruction, decode the instruction during pulse T2.
- what operation has to be performed that is stored in IR(12-14) and address of operand at IR(0-11) and whether the address is direct or indirect that is in IR(15).
- decode the opcode of instruction.

Decision cycle & Execute cycle

at T3

- Before Execute we have to take decision that whether the data is 0 or 1 from 000-110 if it will 0 and if it will 111 so Data will be D7 memy 1
- i.e. consider D7 = 0
- D7 = 0 mean it is memory reference
- After that it will check whether I is 0 or 1
- if I = 0 memy direct address so do nothing
- execute the memory reference instruction
- make sequence counter to SC \rightarrow 0

- If the $I=1$ means indirect address then make the effective address into address register.
 - After that execute memory reference instruction and make sequence counter to zero.
- Again let's consider when $D7 = 1$
- If $D7 = 1$ CPU will check I whether it is 0 or 1
 - If the $I=0$ means it is register reference
 - After that execute register-reference instruction and make sequence counter to 0 →
- If $I=1$ means Input/Output instruction
 - After that Execute the input-output instruction and make the sequence counter to zero.



Microprocessor is of 8-bit introduced in 1971.
Its actual name is 8085A, it contains 6200 transistors approx. Its dimensions are 16.4mm x 22mm.
It is having 40 pins dual inline - package.

The architecture of 8085 microprocessor consists of several components including Accumulator, registers, program counter, stack pointer, instruction register, decoder, flag register, data bus, address bus and control bus. ALU.

* **Multiplexer:** Microprocessors have 6 general purpose registers B, C, D, E, H, L which can be ~~be~~ each of 8 bits that can be used as 16 bit pair register BC, DE and HL only combine with each other. These registers are used to store memory addresses and data. Along with that there is also W and Z temporary register but it cannot be used by programmers each of 8 bit.

* **PC →**

In multiplexer SP and PC, INC/DEC are used. The program counter is a 16-bit register that contains address of the next instruction after fetching one instruction. PC ~~is~~ is incremented ~~inc/dec~~ is used to increment & decrement the memory location by 1.

* **SP →** Stack pointer is a 16-bit register that is used to manage the stack and store data temporarily. It is also used to keep track of the top of the stack.

* **Data bus:** The data bus is an 8-bit bus that is used to transfer data between the microprocessor and memory.

* Address buffer & address-data buffer: The content stored in the stack pointer and program counter is loaded into the address buffer and address-data buffer to communicate with CPU.

DATE: _____

* Address buf: Address buf is a 16 bit bus that is used to address memory and other devices. also it is used to select the memory location that the microprocessor wants to access.
(A₁₅-A₈)AB

* Control buf: Control buf is a set of signals that controls the operations of MP. including read, write, reset, interrupt operations.
(AD₇-AD₀)DB

* Instruction register: Instruction register is an 8 bit register that contains the current instruction that is being executed.

* Instruction Decoder: Instruction decoder is used to decode the instruction coming from instruction register.

* Timing and control unit: It provides timing and control to the microprocessor to perform operations. timing and control signals are: READY, RD, WR, ALE, S₀, S₁, IO/M, RESET IN, RESET OUT.

* Accumulator: It is an 8-bit register used to perform arithmetic, logical I/O & load/store operations. It is connected to internal data bus & ALU.

* Temp register: It is an 8-bit register, which holds or any → the temporary data of arithmetic and logical operations.

* Flag register: Flag register is an 8-bit register that contains status of the result. These flags include
① sign flag (positive, negative) ② zero flag (zero) ③ parity flag (even/no of 1 bits) ④ carry flag (carry)

* Arithmetic and logic unit (ALU): As the name suggest, it performs arithmetic and logic operations like addition, subtraction, AND, OR etc. ALU is of 8 bit.

* Interrupt control: As the name suggests it controls the interrupt process during a process. When a microprocessor is executing a main program and whenever an interrupt occurs, the microprocessor shifts the control from the main program to process the incoming request. After the request is completed, the control goes back to the main program.

There are 5 interrupt signals ① INTR ② RST 7.5
③ RST 6.5 ④ RST 5.5, ⑤ TRAP and INTA

Priorities of interrupts \rightarrow TRAP > RST 7.5 > RST 6.5 > RST 5.5 > INTR

* Serial Input/Output Control: It controls the serial data communication by using SID and SOD. The SID pin is used for serial communication. The SOD pin is used for serial output.

Q PIN Diagram of 8085 microprocessor:

Microprocessors introduced in 1977 it is of 8-bit UP, its actual name is 8085A, it contains 6200 transistors approx. Its dimensions are 164mm x 222mm, it is having 40 pins Dual inline package.

crystal	$X_1 \rightarrow$	1	40	$\leftarrow V_{CC}$
Input	$X_2 \rightarrow$	2	39	$\rightarrow HLDA$
Reset out	\leftarrow	3	38	$\leftarrow HOLD$
Serial I/O	$SIP \rightarrow$	4	37	$\rightarrow CLK OUT$
	$SOP \leftarrow$	5	36	$\leftarrow RESET IN$
	$TRAP \rightarrow$	6	35	$\leftarrow READY$
	$RST75 \rightarrow$	7	34	$\rightarrow IO/M$
Interrupts	$RST6.5 \rightarrow$	8	33	$\rightarrow S_1$
	$RST5.5 \rightarrow$	9	32	$\rightarrow RD$
	$INTR \rightarrow$	10	31	$\rightarrow WR$
	$INTA \leftarrow$	11	30	$\rightarrow ALE$
AD ₀ - AD ₇	$AD_7 \leftrightarrow$	12	29	$\rightarrow S_0$
Address & data Bus	$AD_6 \leftrightarrow$	13	28	$\rightarrow A_{15}$
	$AD_5 \leftrightarrow$	14	27	$\rightarrow A_{14}$
	$AD_4 \leftrightarrow$	15	26	$\rightarrow A_{13}$
	$AD_3 \leftrightarrow$	16	25	$\rightarrow A_{12}$
	$AD_2 \leftrightarrow$	17	24	$\rightarrow A_{11}$
	$AD_1 \leftrightarrow$	18	23	$\rightarrow A_{10}$
	$AD_0 \leftrightarrow$	19	22	$\rightarrow A_9$
	$VSS \rightarrow$	20	21	$\rightarrow A_8$

1. Data Bus (AD₀ - AD₇): This 8-bit bidirectional bus transmits and receives data between the microprocessor and memory or I/O device.

2. Address Bus ($A_0 - A_{15}$): These 16-pins carry memory addresses and I/O port address during different operations.

(X_1, X_2)

3. Crystal input (I_{T1}, I_{T2}): These two pins are connected to an external crystal or resonator that maintain internal frequency of microprocessor.

4. Reset out (3): This signal indicates that the microprocessor unit is being reset. The signal can be used to reset other devices.

5. Serial Input/Output (4-5):

SIP: Serial input data is a data line for serial input. It takes data from AC using RIM.

SOP: Serial output data is a data line for serial output. It takes 1 bit from AC to Serial port 8085.

6. Power supply & clock frequency:

- V_{CC} (40): +5V power supply to active the IC of UP.
- V_{SS} (20): Ground reference or ground connection for UP.

7. Interrupts: 8085 has five interrupt signals that are used to interrupt a program execution.

~~INT~~ ① TRAP ② RST 7.5 ③ AST 6.5 ④ RST 5.5
⑤ INTR

Priority of interrupts.

TRAP > RST 7.5 > RST 6.5 > RST 5.5 > INTR

- ① ~~INTR~~ TRAP: It is non maskable interrupt.
- It has highest priority.
 - TRAP is used for power failure and emergency shutdown.

- RST 7.5: It is maskable interrupt, it has the second highest priority, it is positive edge triggered only.
- RST 6.5: It is maskable interrupt, it has third highest priority, it is level triggered only.
- RST 5.5: It is a maskable interrupt. It has the fourth highest priority. It is also level triggered. This interrupt is very similar to RST 6.5.
- INTR: It is a maskable interrupt, it has the lowest priority, it is a general purpose interrupt and request signal interrupt.
- INTA: It is an interrupt acknowledgment sent by the microprocessor after INTR is received.

Timing & Control signals:

- HLDA: HLDA stands for Hold Acknowledge. The CPU uses this pin to acknowledge the HOLD signal.
- HOLD: It indicates that another device is requesting the use of address and data bus. The microprocessor stops using the address and data bus and resumes after finishing the current cycle.
- clock out: This signal can be used as the system clock for other devices.

- Reset IN: It is active low signal used to reset the microprocessor.
- READY: It indicates that the device is ready to send or receive data. If READY is low then the CPU has to wait for READY to go high.
- IO/M: This signal is used to differentiate between IO and memory operations. When it is high it indicates IO and when it is low then it indicates memory operation.
- SI, SO: It indicates the current operation being performed by the microprocessor.
- RD: This is the active-low signal that indicates that the microprocessor is ready to read data from memory or I/O device.
- WR: This is the active-low signal that indicates that the microprocessor wants to write data to memory or I/O device.
- ALE: It is an Address Latch Enable signal, when it is high it indicates address and when it is down it indicates data.

Maskable interrupt: Interrupts which can be ignored by CPU for sometime which can be disable or enable.

non-maskable \rightarrow TRAP.

Q Difference b/w Assembly language and machine language.

Assembly language

Assembly language is a low-level language that uses mnemonics and symbols to represent machine instructions. It's a human readable machine code.

Machine language is the lowest level programming language that uses binary digits either 0 or 1 to represent instructions. It is machine readable.

i) Only understood by ^{human} computer.

only understood by computer

ii) It is represented by mnemonics such as MOV, ADD, SUB.

It is represented by 0 or 1.

iii) Easy to understand.

difficult to understand.

iv) Execution is slow of ML.

Execution is fast of AL.

v) It is machine dependent and it is not portable.

It is hardware dependent

vi) No need of translator.

Need of translator to convert mnemonic into machine language.

vii) Modification and errors debugging can be done.

Modification and errors debugging cannot be done.

Q18 → diff b/w Microprocessor and microcontroller

Microprocessor

Definition → Microprocessor is a processor that execute instruction one by one and control I/O devices.

Microcontroller

A microcontroller is a small computer on a single integrated circuit that is designed to control specific task.

①

Speed → Has a high clock speed

Has a lower clock speed.

② Cost → Too costly.

Less costly.

③ Size → Larger in size

small in size

④ Applic → Computers, laptops

Microwave, washing machine

⑤ It is 32 bit or 64 bit architecture.

It is of 8, 16 or 32 bit architecture.

⑥ It requires More memory

It requires less memory.

⑦ Memory, I/O ports, timers etc. connected to the CPU externally

CPU and all other elements are integrated into a single chip.

⑧ Use → widely use in computer system

widely use in embedded system.

~~diff~~

* Addressing Modes: Addressing The different ways of Specifying the location of an Operand in an instruction are called Addressing Mode.

Types of Addressing Mode

- ① No address field is required
- ②
 - ① Implied Mode / Implicit
 - ② Immediate Mode
- ③ Address specifies the memory location
- ④ Direct Mode
- ⑤ Indirect Mode
- ⑥ Address specifies the processor register
- ⑦ Register direct Mode
- ⑧ Register indirect Mode
- ⑨ Auto increment and auto decrement Mode
- ⑩ Address field plus content of CPU register
[Displacement Addressing]
- ⑪ Relative addressing Mode
- ⑫ Index addressing Mode
- ⑬ Base register mode Addressing Mode

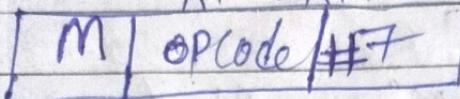
- ① Implied Mode: Operands are specified implicitly in the definition of instruction.
Ex \Rightarrow CMA, CLA
Complement the content of Accumulator
- clear the content of Accumulator

② Immediate Mode : Operand is specified in the instruction itself.

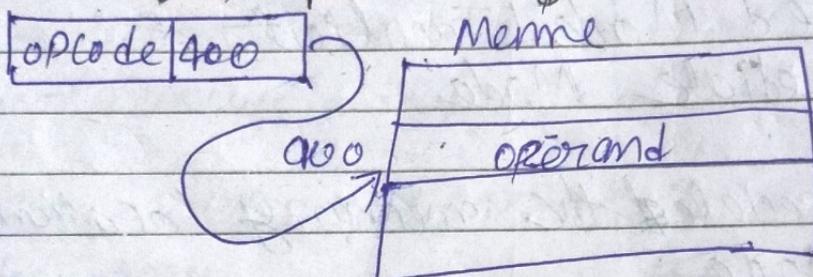
Example →

ADD R1, #7

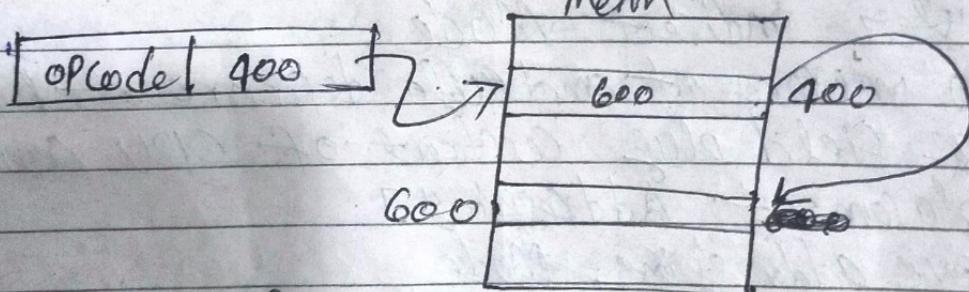
$$R1 \rightarrow R1 + 7$$



③ Direct Mode : In this actual effective address of operand is directly given in the instruction.



④ Indirect Mode : In this effective address of operand is not directly given in instruction, there is next a memory address where operand is present.

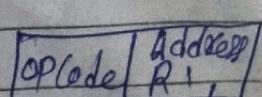


⑤ Register Direct Mode : Operand in the register address field of the instruction refers to the CPU register that contain the operand.

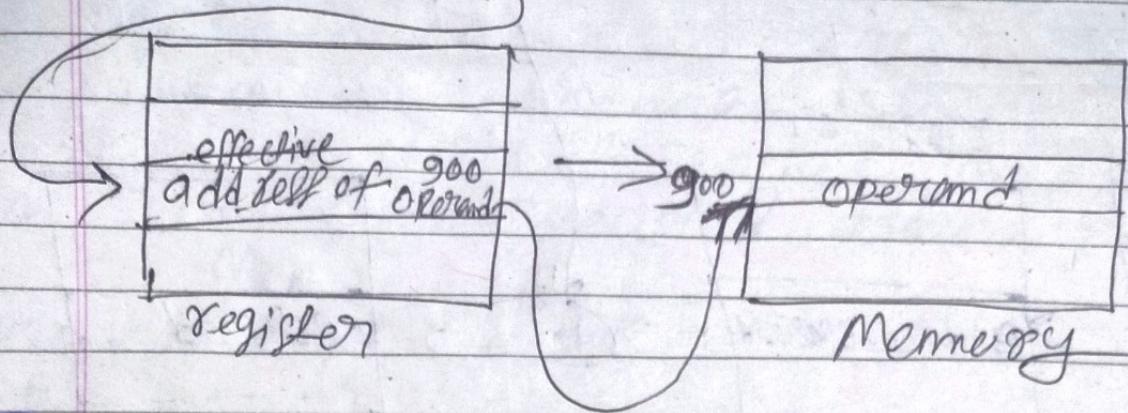
Ex → MOV R1, R2

⑥ Register Indirect Mode : Register containing address of operand rather than operand itself.

LD(R1)



Opcode [add reg 900]



(7) Auto increment and auto decrement Mode
It is similar to register indirect Mode, effective address of the operand is a content of the register specified in the instruction.

In auto increment Mode register is incremented after its value is used to access the memory.

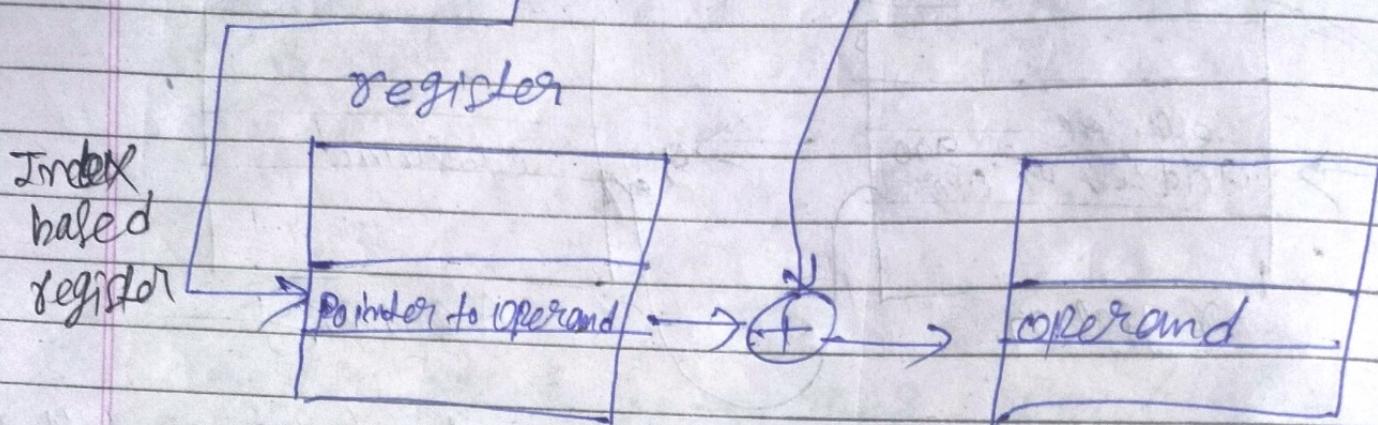
$$\text{Ex} \rightarrow EA = (\text{RI}) +$$

$$EA = -(\text{RI})$$

In auto decrement Mode the register is decremented before its value is used to access the memory.

(8) Displacement Addressing:

[OPcode | Reg | Address A]



~~Dis~~ = Add + offset value.

Note \rightarrow relative addressing Mode:

\rightarrow reg \rightarrow program counter.

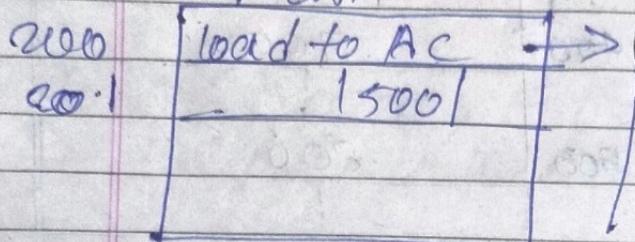
\rightarrow index \rightarrow index reg

\rightarrow base \rightarrow base reg

Effective Address = PC + offset

Q The two word ~~is~~ing there in address 200 at address 201 there is a instruction named as load to Accumulating, the first word of specified load opode and second word specified address. PC has value 200

Mean



the first word of instruction specified the operation code and mode and second word specified the address.

PC is the value of 200 for the fetching the instruction.

the content of program ~~reg~~ \rightarrow 400
and the content of index reg is 100

Find the off for the each possible Mod
calculate the effective add and
the operand loaded in the memory.

		PAGE NO.	
		DATE	
$PC = 200$	200	Load to AC Mode	
	201	Address = 500	
$R_1 = 400$	202	Next instruction	
$XR = 100$	399	450	
	400	700	
AC	500	800	
	600	900	
	702	325	
	800	300	

Q For each possible mode we calculate the effective address and the operand that must be loaded into Ac.

① immediate mode \rightarrow operand value in address field
 $EA = 201, AC = 500$

② Direct address mode \rightarrow Address field containing effective address
 $EA = 500, AC = 800$

③ Indirect address mode : Address field specifies the address where the effective address of operand is stored in memory
 $EA = 800, AC = 300$

④ Register direct mode : Register containing operand
 $EA = \underline{\underline{R_1}}, AC = 700$

⑤ Register indirect mode : Register containing effective address of operand.

$$EA = 400 \quad AC = 700$$

(6)

Auto-increment Mode: Like ~~in register indirect~~
except register value is incremented after it
is used.

$$EA = 400 \quad AC = 700$$

(7)

Auto-decrement Mode: value is decremented before it is used.

$$EA = 399 \quad AC = 450$$

(8)

Relative Address: PC + address post

~~200 + 500 = 700~~

$$202 + 500 = 702$$

$$EA = 702 \quad AC = 325$$

(10) Base reg =
+ address
~~field~~

(9)

Indexed address mode:

~~XR + add 200 & post~~

$$100 + 500 = 600$$

$$EA = 600 \quad AC = 900$$

EA AC

(11)

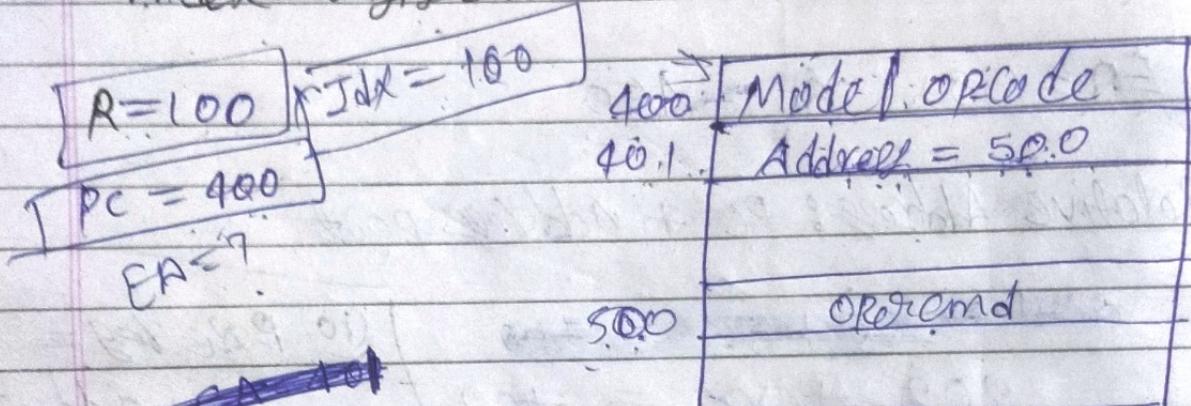
① Immediate address	201	500
② Direct address	500	800
③ Indirect address	800	300
④ Register direct Mode	—	400
⑤ Register Indirect Mode	900	700
⑥ Auto increment Mode	900	700
⑦ Auto decrement Mode	399	450
⑧ Relative address	702	325
⑨ Indexed address mode	600	900

(P18)

PAGE NO. _____

DATE _____

what are addressing Modes - An instruction is stored at location 400 with its address field at location 401. The address field has the value 500. A processor register R containing the number 100. Evaluate the effective address if the addressing mode of the instruction is
① direct ② immediate ③ relative ④ register ⑤ indirect ⑥ index with R of the index register.



① $\rightarrow EA = 500$
② $\rightarrow EA = 401 + AC = 800$
③ $\rightarrow EA = 100 + 500 = 600$
④ $\rightarrow EA = 100$
⑤ $\rightarrow EA = 100 + 500 = 600$

9 An instruction is stored at location 300 with its address field at location 301. The address field has the value 400. A processor register R1 containing the number 200. Evaluate the effective address if the addressing mode of the instruction.

- ① Direct
② Immediate
③ Relative
④ Register direct
⑤ Index with R1 of the index register.

$$P_C = 300$$

四

$$R_1 = 200$$

300

Mode / opcode
Adddress = 400

BA =

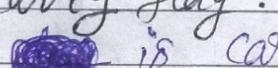
- ⑨ 400 ⑩ BA = 301
⑪ 302 + 400 = 702
⑫ 200
⑬ 200 + 400 = 600

400

operam

flag register: flag reg tells about status of current result, flag is changed by ALU after every operation.

S | z | Ac | P | cix

- ① Carry flag: It tells about whether these
 is carry or not
 $A = 10H$, $B = 20H$ ADD B $A \leftarrow A + B$

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0
<hr/>							

$cy \rightarrow 0$ $\frac{1}{1} \overline{1}$ $cy = 1$

② Parity flag: It is calculated by
 no of 1's

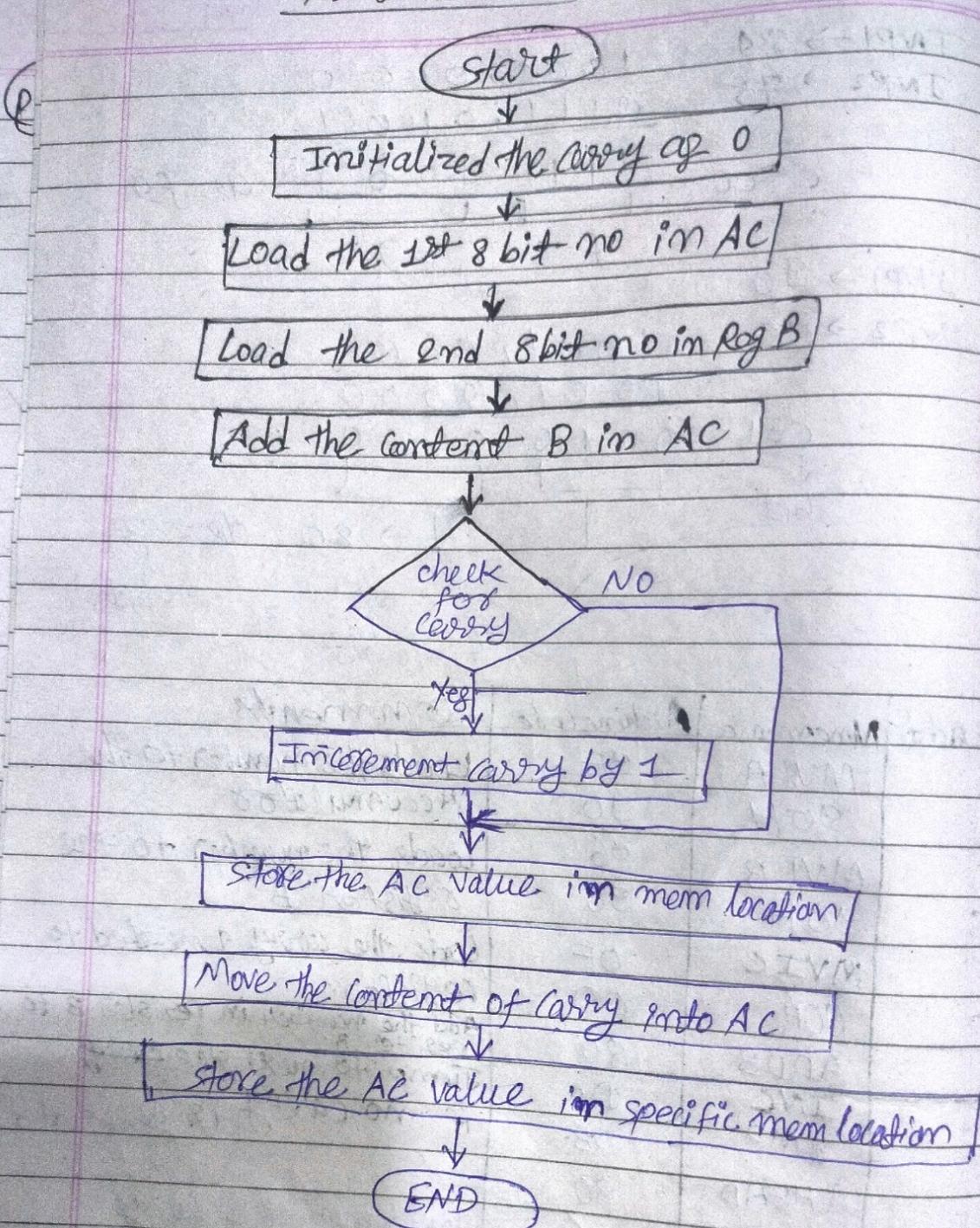
$$\begin{aligned} \text{Ex} \rightarrow & \quad 100010000 \rightarrow \text{No of } L'B \rightarrow 2 \rightarrow p=1 \\ & 100000000 \rightarrow \text{No of } L'R \rightarrow 1 \rightarrow p=0 \end{aligned}$$

- ③ Auxiliary flag : carry from lower nibel to higher nibel is there

000	000	000	AC → 1
000	000	000	
001	010	000	

zero flag: It tells whether the flag is zero or not, so if, 80 H, 80 H

Flowchart



Explain → $Imp_1 = 90, Imp_2 = 90$
move 90 in $\rightarrow AC$ (MVI A)
move 90 in $\rightarrow B Reg$ (MVI B)
initialize carry by 0 (MOVIC 00)
Add B in AC (ADD B)
If No carry jump for ~~store~~ (INC)
INC carry by 1. (INR)

~~and store~~

and move carry into AC \rightarrow (MOV A, C)
store the content of AC in memory (STA)
HLT the program execution (HLT)

Output = 20
Carry = 01