

Microprocessors & Microcontrollers

↳ Here sys is embedded

History of Microprocessors

I7 → ... Core2 Duo / Dual Core

Series = 32 bit
(IV-I)

80486
80386
80286

8088 → 16 bit
8086 → 16 bit
8085 → 8 bit
8080
8008
4004 → 4 bit

History of Microcontrollers

8031 → 8051 → 8052

- These are designed by Intel

PIC

Microchip Corporation

Board

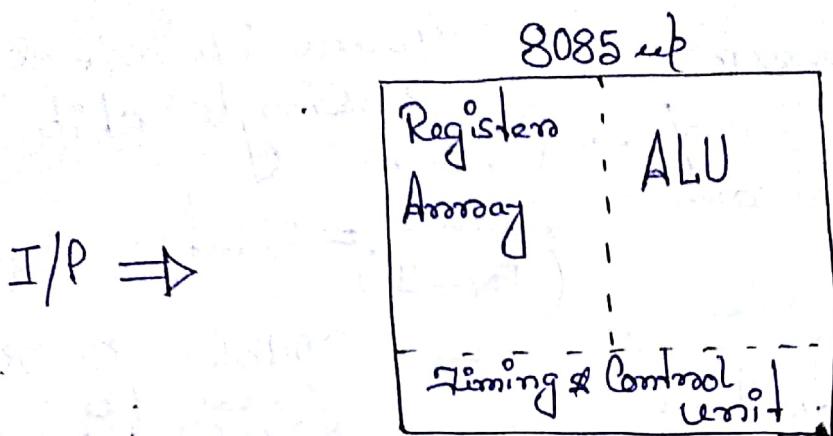
8085 uP

Main Features:

1. It's a 8 bit uP.
2. It's designed by intel in 1976.
3. Its operating frequency / clock frequency is 2.03 megahertz.
4. It has 8 bit data bus.
5. It has 16 bit address bus.
6. It can access maximum 64 KB memory.
7. Its operating voltages +5V.

Internal architecture:

It Instⁿ from memory chip



Registers
array

⇒ It's consists of several registers.
So the funⁿ is these are used during
Program execution

ALU

⇒ Stands on Arithmetic & Logical unit
It performs different type of
Arithmetic & Logical operation

Timing &
Control unit

⇒ It controls different types of
read & write operation.

Registers array:

Registers ⇒

1 bit Registers storage location

0/1

2 bit registers

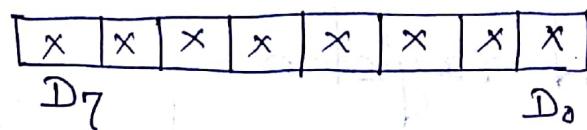
x x

- 0 0 x = 0/1
0 1
1 0
1 1

4 bit registers

x x x x

8 bit register



LSB = D₀

MSB = D₇

16 bit register

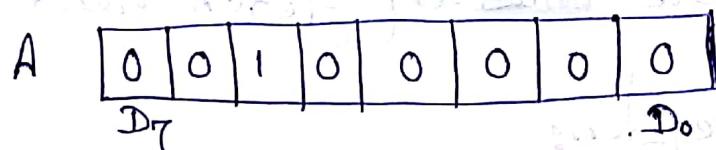


- 1. A registers:
 - ① It's called Accumulators register.
 - ② It's a 8 bit register

example:

$$A = 20H$$

$$i = 20H$$



- ③ All the arithmetic & logical operations are performed with respect to the content of accumulator.

ex: $\underbrace{1^{\text{st}} \text{ data} + 2^{\text{nd}} \text{ data}}_A = \text{Result}$

A

Before

A

After

General Purpose Registers:

- ① In 8085 there are 6 gpr, name - B, C, D, H, L
- ② All these are 8 bit length.
- ③ These 6 gpr combined to form a 16 bit register pair

B

C

H-8

L-8

B C

H L

D E

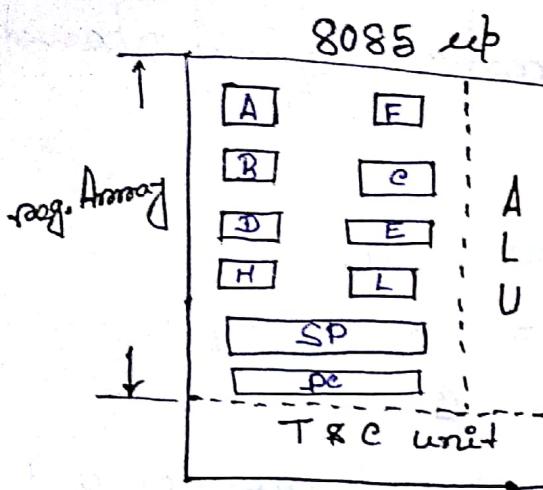
H L

3. SP register: ① full form Stack Pointer reg.
② It's a 16 bit register
③ Points to the top of the stack memory

4. PC register: ① full form Prog. Counter register
② It's a 16 bit register.
③ func: It points to the mem. location from where ~~up~~ fetch next instⁿ.

5. W and Z registers:
① Both the registers are 8 bit
② They are internally used by processor during execution and user can't use it.

6. Flag register: ① It is the status register of 8085
② It is a 8bit register.



8 bit flag registers

S	Z	X	AC	X	P	X	CY
D ₇	D ₆	D ₅	D ₄	-D ₃	D ₂	D ₁	D ₀

D₀ \Rightarrow Carry bit / Store carry information

D₁ \Rightarrow Don't Care bit

D₂ \Rightarrow Parity bit

If the no. of 1's is odd (Result)

Then P flag will = 0

D₃ \Rightarrow Don't Care

D₄ \Rightarrow Auxiliary carry bit

D₅ \Rightarrow Don't Care

$$\begin{array}{r}
 & & D_4 & D_3 \\
 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\
 \hline
 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1
 \end{array}$$

AC = 1 if D₃ generates a carry

D₆ \Rightarrow Zero flag bit

$$\begin{array}{r}
 & & D_4 & D_3 \\
 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
 \hline
 & 1 & 0 & 0 & 0 & 0 & 0 & 0
 \end{array} = \text{Accumulator}$$

CY = 1 here, Z = 1

For any non zero value Z = 0

D₇ \Rightarrow Sign bit

$$\begin{array}{r}
 & & D_4 & D_3 \\
 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\
 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\
 \hline
 & 1 & 1 & 1 & 1 & 1 & 1 & 1
 \end{array} = A$$

S = 1 as D₇ bit is 1

If there is not occurs any arithmetic & logical operation
then FI (flag reg.) will same.

$$20H = A \\ + A0H = B \\ \hline C0H = A \text{ is nope,} \\ B \text{ is same}$$

MVI A, 20H = Same

" B 10H = "

ADD B = Change status

Then the value of addition will
be in Accumulators

if Z = 0 Then A=0

~~MVI M, 05H~~

~~MVI C, 00H~~

~~MVI B, 10H~~

~~ADD B~~

Addition of two 8-bits numbers, Result, 16 bits

Ex: 9001 \Rightarrow A9

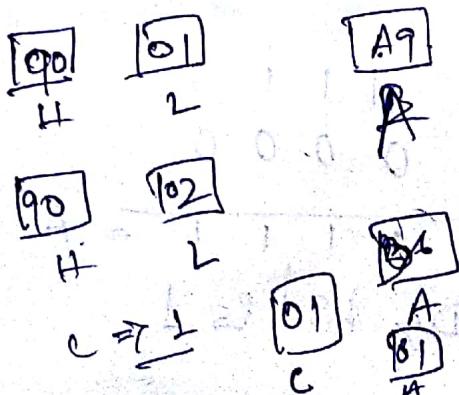
9002 \Rightarrow 8D

9003 = 26 2003 Result

9004 = 01 0000

$$\begin{array}{r} 16124 \\ 16 \\ \hline 8D \end{array}$$

MOV A, A9
MOV B,



MVI C, 00H	2B	2M
LXI H, 9001H	3B	3M
MOV A, M	1B	2M
INX H	1B	1M
ADD M	1B	2M
JNC L1	2B	3M
INR C	1B	1M
LISTA 9003	3B	4M
MOV A, C	1B	1M
STA 9004H	1B	1M
HLT		

8000	(0E)	} MVR C, 00
8001	00	
8002	(21)	} INX H, 9001
8003	01	
8004	90	} ADD M
8005	(1E)	
8006	(23)	INX H
8007	(8C)	ADD M
8008	(D2)	} JNC 800D
8009	00	
800A	80	
800B	(0C)	} INR C
800C	(32)	
4: 800D	03	} STA 9003
800E	90	
800F	(79)	} MOV A, C
8010	(32)	
8011	04	} STA 9004
8012	90	
8013	(76)	HLT

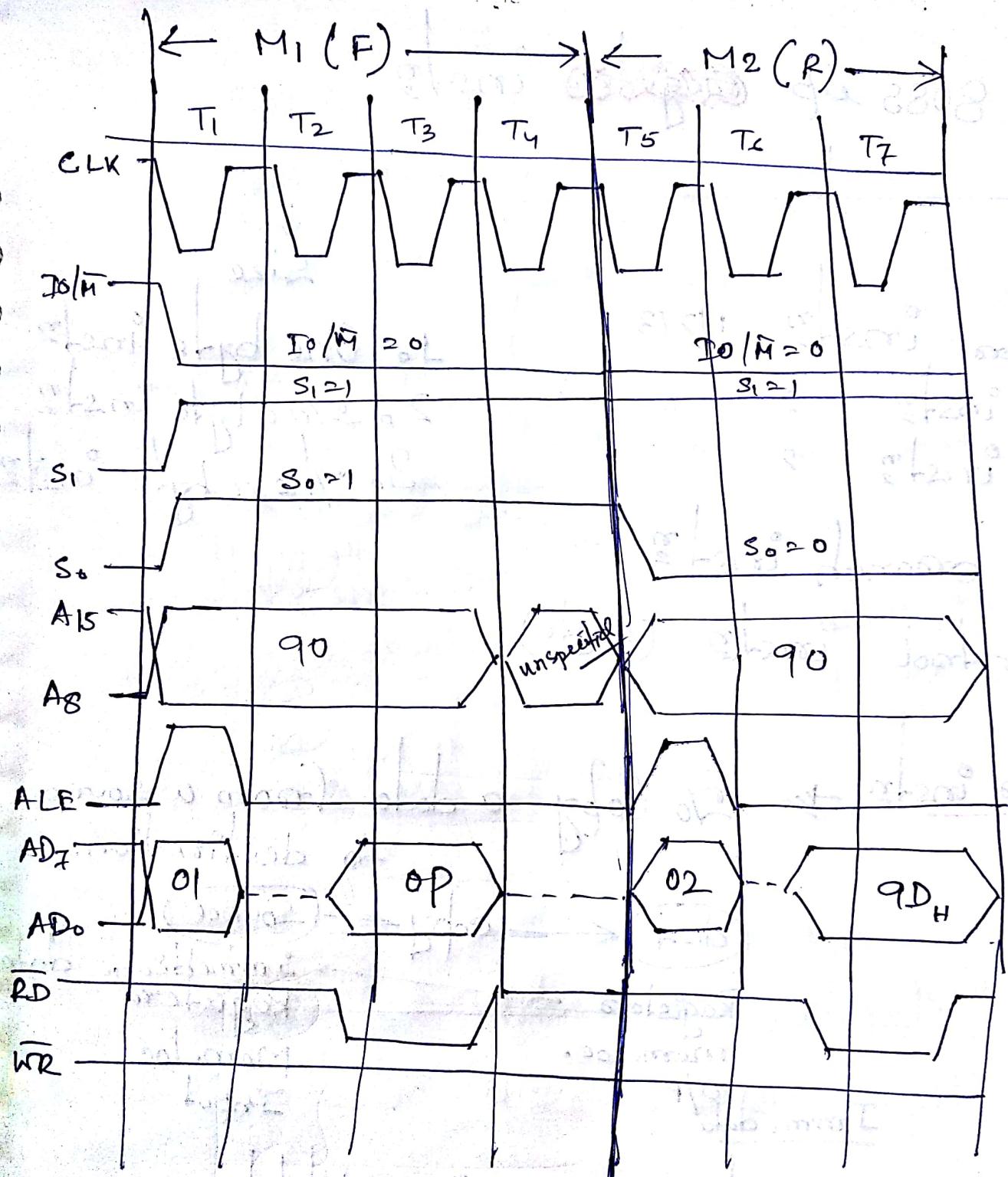
MVI B, 9D₁₆

9001 OP →
9002 RD →

2 N/C
F, R. $\frac{4+3}{7}$

Table : Machine Cycle Status & Control Signal

Opcode Fetch	M/c cycle	I/O/M	S ₁	S ₀	Control signal
opcode fetch	0		1	1	$\overline{RD} = 0$
Memory Read	0		1	0	$\overline{RD} = 0$
Mem. Write	0		0	1	$\overline{WR} = 0$
I/O Read	1		1	0	$RD = 0$
I/O write	1		0	1	$WR = 0$
Interrupt Acknowledge	1		1	1	$INTA = 0$
Hold	Z		0	0	
Hold	Z		X	X	$\overline{RD}, \overline{WR} = Z$
Reset	Z		X	X	$\overline{INTA} = 1$



8085 up ~~instructions~~ instructions

func

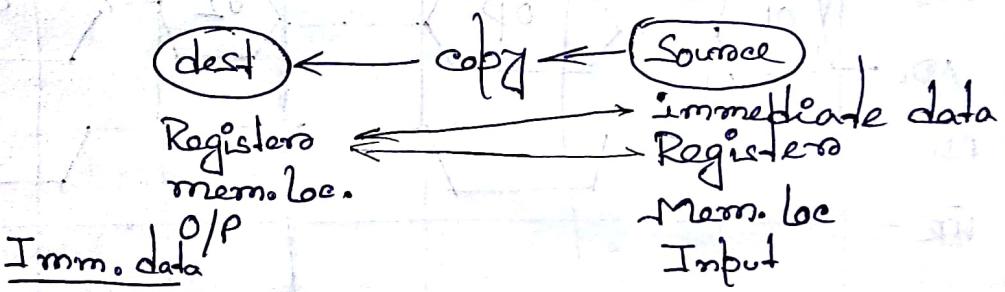
1. Data transfer inst 1/2/3
2. Arithmetic inst 2
3. Logical inst 2
4. Program branch inst 2
5. M/C control inst (1 byte)

Size

1. One byte inst
2. Two byte inst
3. Three byte inst

Data transfers inst

→ To copy a data from a source to destination.



Registers

8 bit

8 bit → MVI R_D, 8 bit data

16 bit

16 bit → LXI R_P, 16 bit data

example: 40H → B

MVI B, 40H

B = 40H

C = XXH

4020H → B,C

LXI B, 4020H

B = 40H

C = 20H

Registers to Registers

8 bit

R_S

8 bit

MOV R_D, R_S

16 bit

16 bit

XCHG

exchange operation DE ↔ HL

Ex:

$$E \Rightarrow A$$

$$E = YY_H$$

$$A = XX_H$$

Mov A, E

$$A = YY_H$$

$$E = YY_H$$

$$D = AA_H \quad E = BB_H$$

} before operation

$$H = CC_H \quad L = DD_H$$

$$HL \rightarrow DE$$

$$XC_HG$$

$$D = CC_H \quad E = DD_H$$

$$H = AA_H \quad L = BB_H$$

Mem. loc. to
(source)

Registers
(dest.)

Registers

8 bit

16 bit

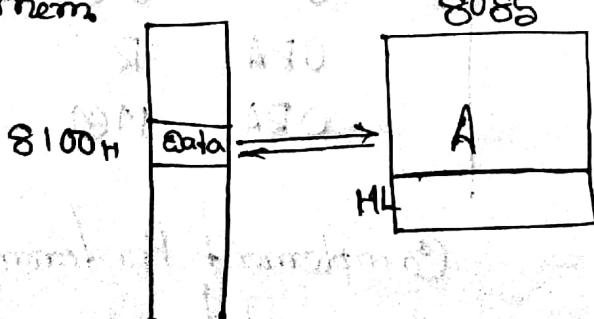
Mem loc

One (8 bit) \rightarrow MOV Ra, M

Two (16 bit)

example:

User mem.



$$HL = 8100_H$$

$L_{X_1} = H, 8100_H$

MOV A, M

Arithmetic Inst:

Add
Sub
Increment
Decrement

1st data 2nd data Result

$$80_H = A + 82_H \Rightarrow ADI \quad 82_H$$

stored in flag reg.

Sub operation:

Sub R

Sub M

SUI 8 bit data

Increment:

8 bit data = INR R

16 " " = INX R

Decrement:

8 bit data = DCR R 16 bit data = Dec R

Logical instruction:

AND, OR, EX-OR, Complement

ANI 8 bit data

ANA R

ANA M \oplus

ORI 8 bit data

ORA R

ORA M \oplus

XRI 8 bit data

XRA R

XRA M \oplus

Complement Performed only in
accumulators

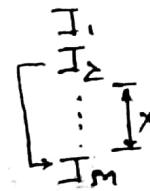
CMA 8 bit data

Program Branching instruction:

Jump / Call Subroutine

Jump operation —
→ unconditional
→ Conditional

Main program



In
Halt

unconditional Jump:

→ JMP 16 bit address of Im

I2 = JMP 8050H

Conditional Jump:

→ if I2 there's Cy = 1/0

S, Z, AC, P, Cy

S = 0 → +ve → JP 16 bit address of Im
= 1 → -ve → JM " " "

Z = 0 → JNZ " " "

= 1 → JZ " " "

P = 0 → JPO " " "

= 1 → JPE " " "

Cy = 0 → JNC " " "

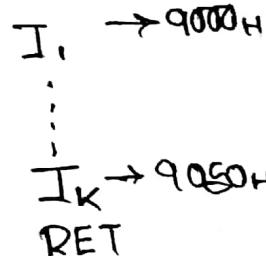
= 1 → JC " " "

Call subroutine:

I2 = CALL 16 bit starting address of subroutine

= CALL 9000H

Sub Program



Problem-2

XRA A
 JZ L₁
 MVI A, 24H $\leftarrow Z=0$
 MVI B, FFH
 L₁: CMA
 ADI 01H
 HLT

\Rightarrow Let A = xxH
 XRA A \rightarrow A = 00H

$$\begin{matrix} S & Z \times Ac & P \times Cy \\ 0 & 1 & 0 \end{matrix}$$

JZ L₁ \rightarrow True \rightarrow

L₁: CMA \rightarrow A = FFH

$$10 \times 0 \times 1 \times 0$$

ADI 01H \rightarrow FFH + 01H = 00H A = 00H

$$0 \times 1 \times 1 \times 1$$

STOP

Problem-3

Main Program

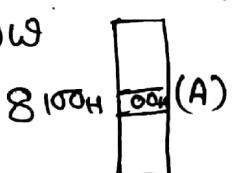
8000H \Rightarrow LXI SP, 820FH \Rightarrow SP = 820FH
 XRA A \Rightarrow 00H
 CALL 9000H \Rightarrow **
 MVI C, F0H \Rightarrow C = F0H
 HLT

Subroutine

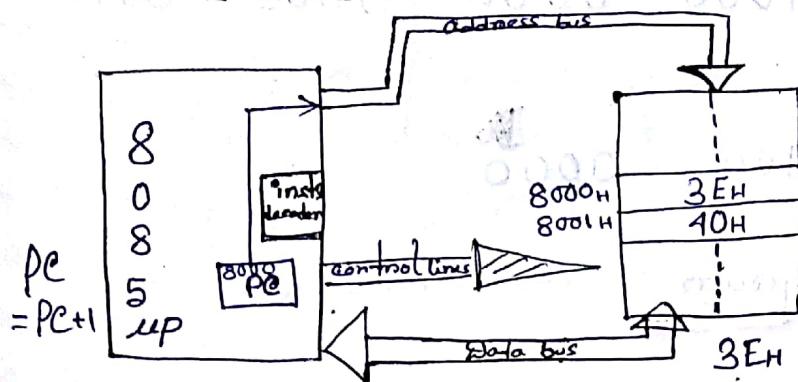
** 9000H \Rightarrow LXI H, 8100H
 MOV M,A
 RET

$$HL = 8100H$$

Now



□ How the Processors Communicate with diff. devices.



There is an add. decoder
which goes to inst? decoders

~~3 types of bus~~

- ① Data bus
 - ② Address bus
 - ③ Control lines
- not unidirectional

① Data bus → ① length 8 bit

② bidirectional

③ D₀ - D₇

② Address bus → ① unidirectional

② binary add. from ~~mem~~ to mem

③ length 16 bit

④ A₀, A₁, ..., A₇, ..., A₁₅

locates 8 bit

add. bus

③ Control Line → To control the operation of eep

like RD, WR

13EH → MVI A, 40H

40H

>6

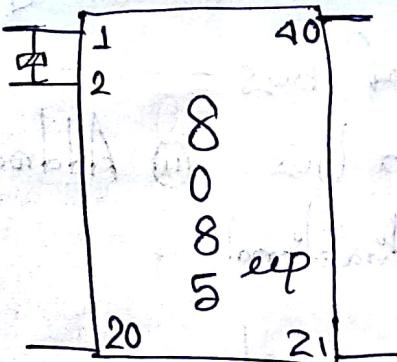
address

8000H

$$8001_H \text{ (add. bus)} = \begin{array}{c|c} A_{15} & A_6 \\ \hline 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{array}$$

$$40H \text{ (data bus)} = \begin{matrix} 1 \\ 0100 \\ 0000 \end{matrix}$$

Pin diagram



These are 4 types

1. Power supply & clock
 2. Address & data bus
 3. Control & status line
 4. ⁽³⁾ Externally initiated
⁽⁸⁾ Signal
 5. Serial data Communication

1. Power Supply & clock

- a. x_1 and x_2 : ① $P_{in \neq 1}$ and 2

⑩ funⁿ: They are crystal oscillators i/p fin.

- One cry. oscil. of frequency 6 MHz (twice the operating frequency of 8085 up) is connected b/w them.

iii) generate clk signal

- b. ~~12~~ Vcc \Rightarrow 0 Rin \Rightarrow 0

⑪ $f_{un} = 0$: +5v powers supply to vcc

- C. GND : ① Pin \Rightarrow 20

⑩ funⁿ: Connect the fm to ground

2. Address bus

A data bus

a. $A_8 - A_{15}$: ① Pin $\Rightarrow 21 - 28$

① fun³ : This lines transfers the higher 8 bit address from up

b. $A_0 - A_7$: ① Pin $\Rightarrow 12 - 19$

D₀ - D₇ ② fun² : Transfers address from esp and (AD₀ - AD₇) ③ data from m₀ to esp

for separate them, they represent an external latch device

3. Control and Status Line

ALE : ① Pin $\Rightarrow 30$

② "Address Latch Enable"

③ To activate the latch we need it

④ It's a small duration Pulse to activate the external latch

$\bar{R}D$: ① Pin $\Rightarrow 32$

② It's active low. $\bar{R}D = 0$ Then it's active

③ This Control Line used to read data from memory and o/p devices.

$\bar{W}R$: ① Pin $\Rightarrow 31$

② It's active low.

③ used to write data to mem. or to o/p devices

IO/\bar{M} : ① Pin \Rightarrow 34

- ② funⁿ : used to Control I/O - O/P / memory operation
if $\text{IO}/\bar{M} = 0$ then it indicate that it is a mem. related operation, for, if it's an I/O - O/P operation

IO/\bar{M}	$\bar{R}\bar{D}$	$\bar{W}\bar{R}$	
0	0	1	M. Read
0	1	0	(M. Write)
1	0	1	I. Read

① Externally initiated signal:

RESET IN : ① Pin \Rightarrow 36

- ② funⁿ : used to reset the system, after reset
PC value = 0 000 (initial state)

RESET OUT : Pin \Rightarrow 3

- ① through this pin, reset other peripheral devices

READY : ① Pin \Rightarrow 25

- ② it's used to inform ~~slow~~ interface slow peripheral devices

HOLD : ① Pin \Rightarrow 39

- ② used in DMA operations (received DMA request)

HLDA : Pin : 38

① Through this line CPU acknowledge the DMA request

② INTR : ① Pin : 10

① funⁿ : It's a hardware interrupt Pin.

INTA : ① Pin : 11

① funⁿ : It's an acknowledgement signal for INTR

RST 7.5, RST 6.5, RST 5.5 : Pins : 7, 8, 9

① funⁿ : These are hardware interrupt Pins

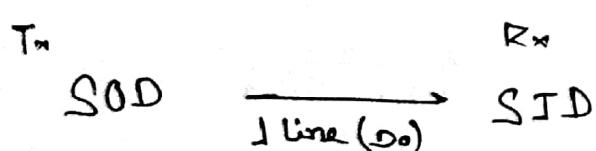
TRAP : Pin : 6

① funⁿ : It is an non maskable hardware interrupt.

5. Serial data communication:

There are 2 Pins for S.D. Comm., They are

SID, SOD



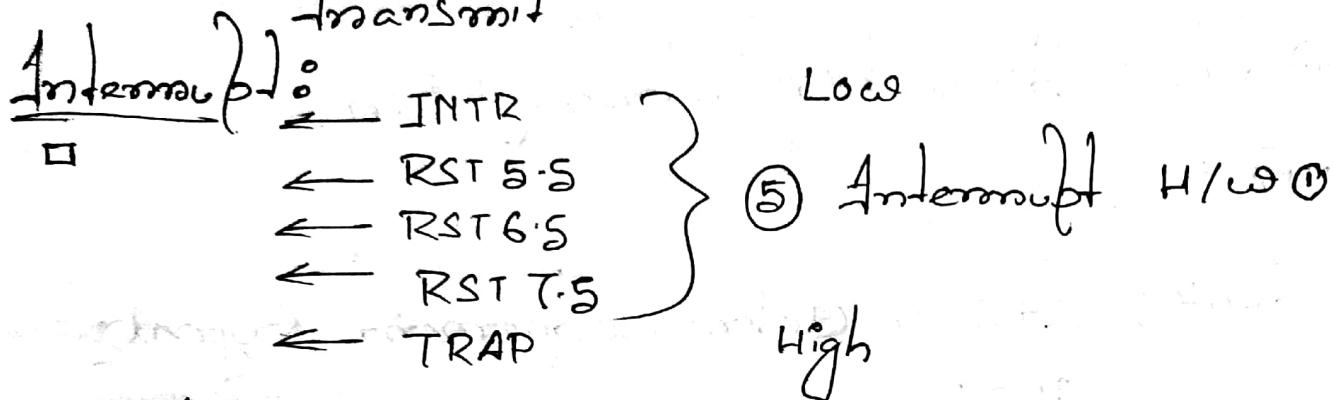
one bit of data at a time via single line
= Serial data Comm.

SID : ① Pin : 5

① funⁿ : It's a serial i/o data line via this line we receive serial data.

SOD : ① Pin: 4

② It's a Serial O/P data; via this line
rep ~~receive~~ serial data
transmit



⑥ Software Interrupt RST 0 — RST 7

□ ① Maskable ② Non-Maskable

↓
INTR

↓
~~receive~~ TRAP

□ □ □

① vectors

↓

RST 5.5 to Trap

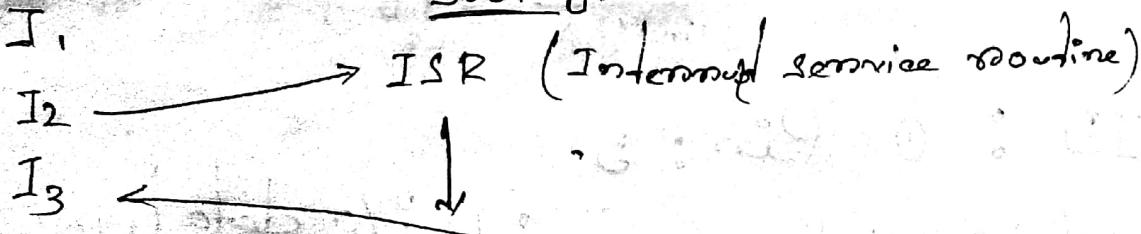
② non-vectors (address is not known to CPU)

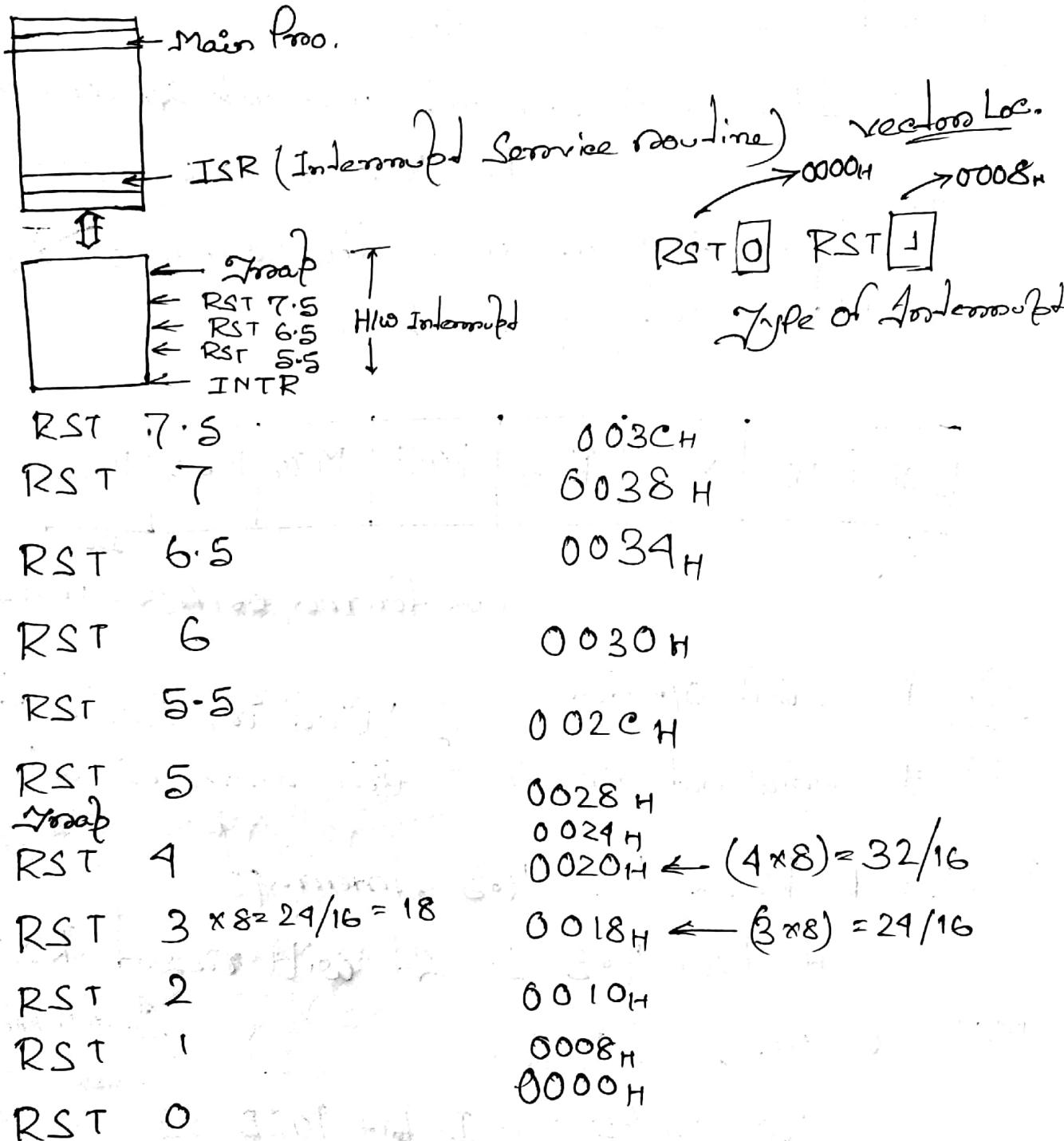
↓

INTR

INTA through ask the unknown address to external devices.

SubProg.





How to enable/disable or mask/unmask an interrupt?

⇒ EI : It's an insⁿ of 1 byte to enable the interrupt in 8085 esp.

DI : It's also 1 byte insⁿ to disable the interrupts in 8085 esp

- SIM \Rightarrow ① Set Interrupt Mask
 ② 1 byte instⁿ to mask/unmask Interrupts.
 (RST 7.5, RST 6.5, RST 5.5)
 ③ This command reads the content of Acc
 and accordingly mask or unmask interrupt

SOD	SDE	X	R7.5	MSE	M7.5	M6.5	M5.5
D ₇							D ₀

8 bit format of Acc. in SIM

SOD \Rightarrow Serial O/P data } Used in Serial data

SDE \Rightarrow Serial data enable } ~~for~~ Communication

RST 7.5 \Rightarrow Reset RST 7.5 Interrupt

if RST 7.5 = 1, AI will reset RST 7.5
Interrupt

MSE \Rightarrow Mask Set Enable

To enable D₀-D₂ bit MSE = 1

To disable D₀-D₂ " " = 0

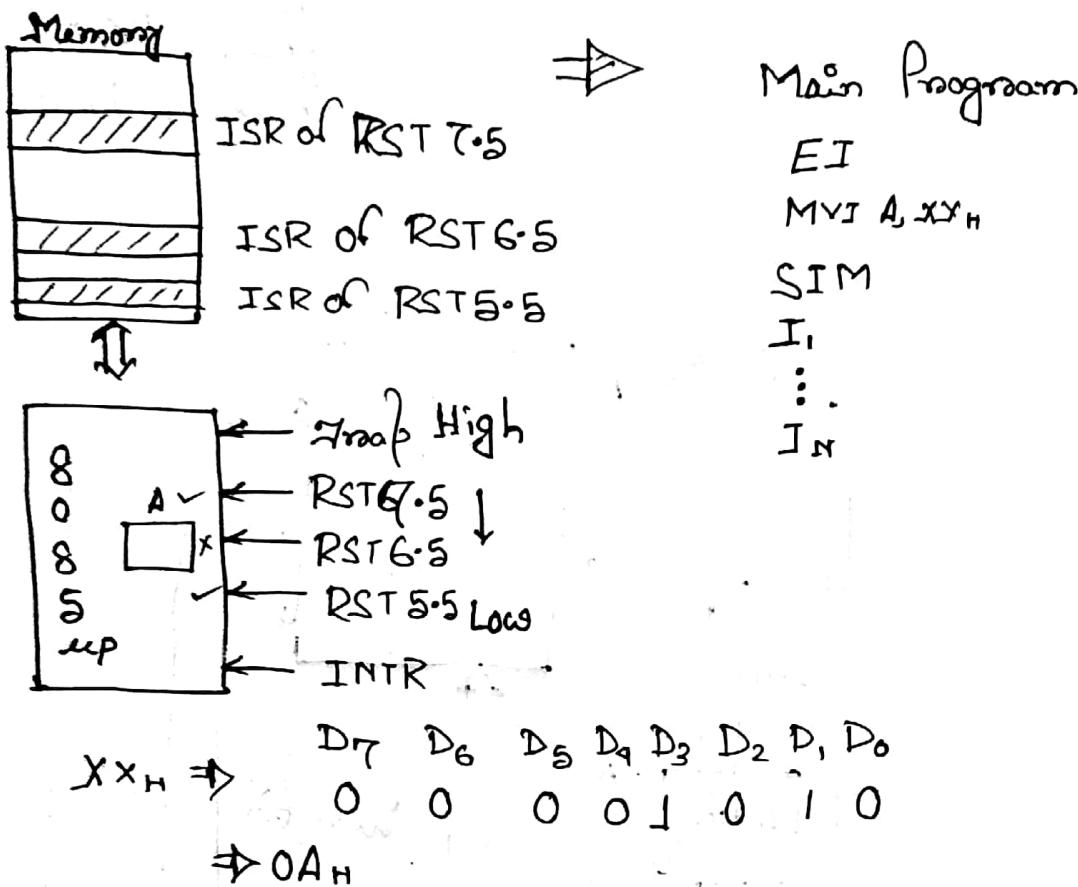
M 7.5 \Rightarrow Mask RST 7.5 interrupt

if M 7.5 = 1, AI will mask RST 7.5
Interrupt

M 6.5 \Rightarrow Same as 7.5

M 5.5 \Rightarrow " " 5.5

Problem 1: Write the instrⁿs to enable interrupts and mask RST 6.5, unmask RST 7.5, RST 5.5 interrupt



Program 2: Write the instrⁿs to check whether RST 6.5 interrupt is masked or unmasked. If it is masked then unmask it (RST 6.5)

→

RIM → Mask ① It's a 1byte instrⁿ to read Pending + Read → Interrupt Status of the interrupt.

② After ^{RIM} read instrⁿ procession loads 8 bit interrupt status into Accumulator

SI	I7.5	I6.5	I5.5	IE	M7.5	M6.5	M5.5
----	------	------	------	----	------	------	------

D₇

D₀

I : Pending (An interrupt); if pending bit value is 1

$\text{IE} \Rightarrow \text{Interrupt Enable}$ $M = \text{Mask}$
 If interrupt core mask
 Then $M=1$.
 Otherwise $M=0$.

Program

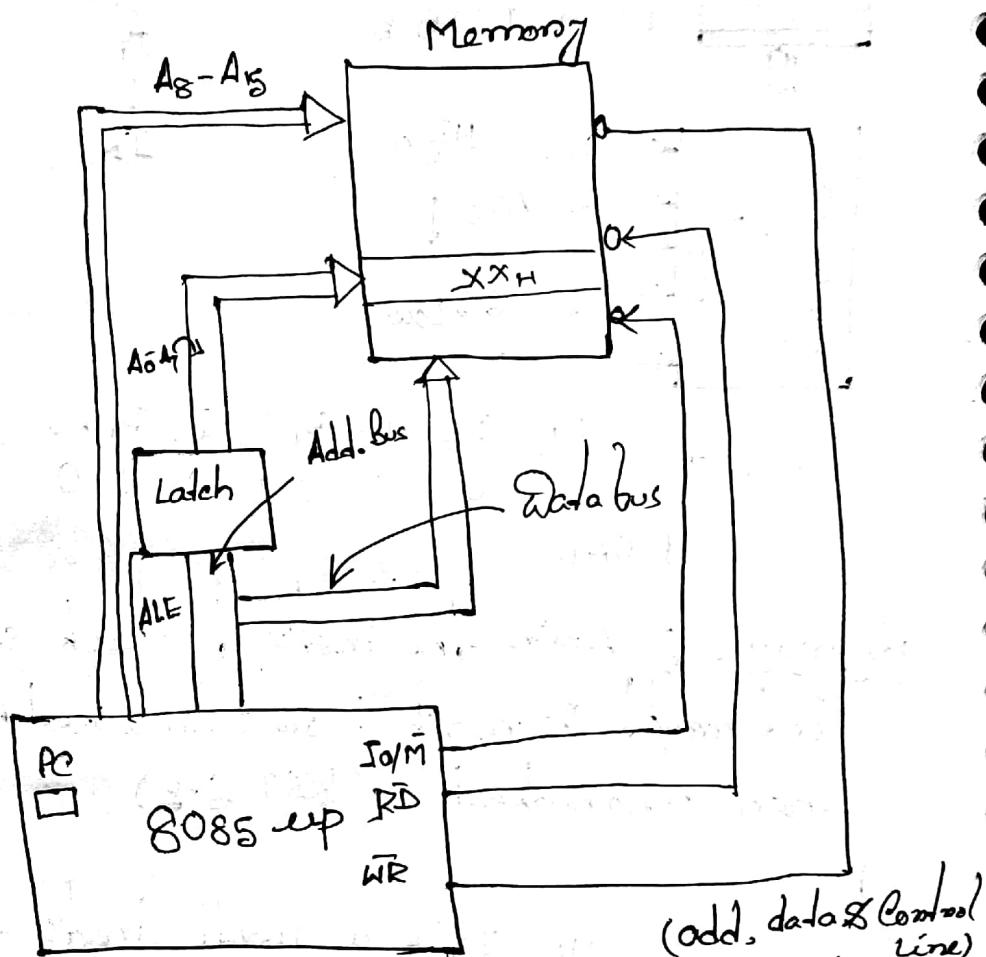
RIM

ANI 02H

JNZ L1

MOV A, XXH

STL
L1



Time = CLK cycle

Timing diagram [Plot of status with respect to time]

Instruction Cycle: The time required to execute a complete instruction. Instruction cycle consists of several m/c cycles.

M/c cycle: The time required to execute 1 part of instⁿ like mem Read, memory write, opcode fetch, etc. It consists of several T state.

MVI A, 23H

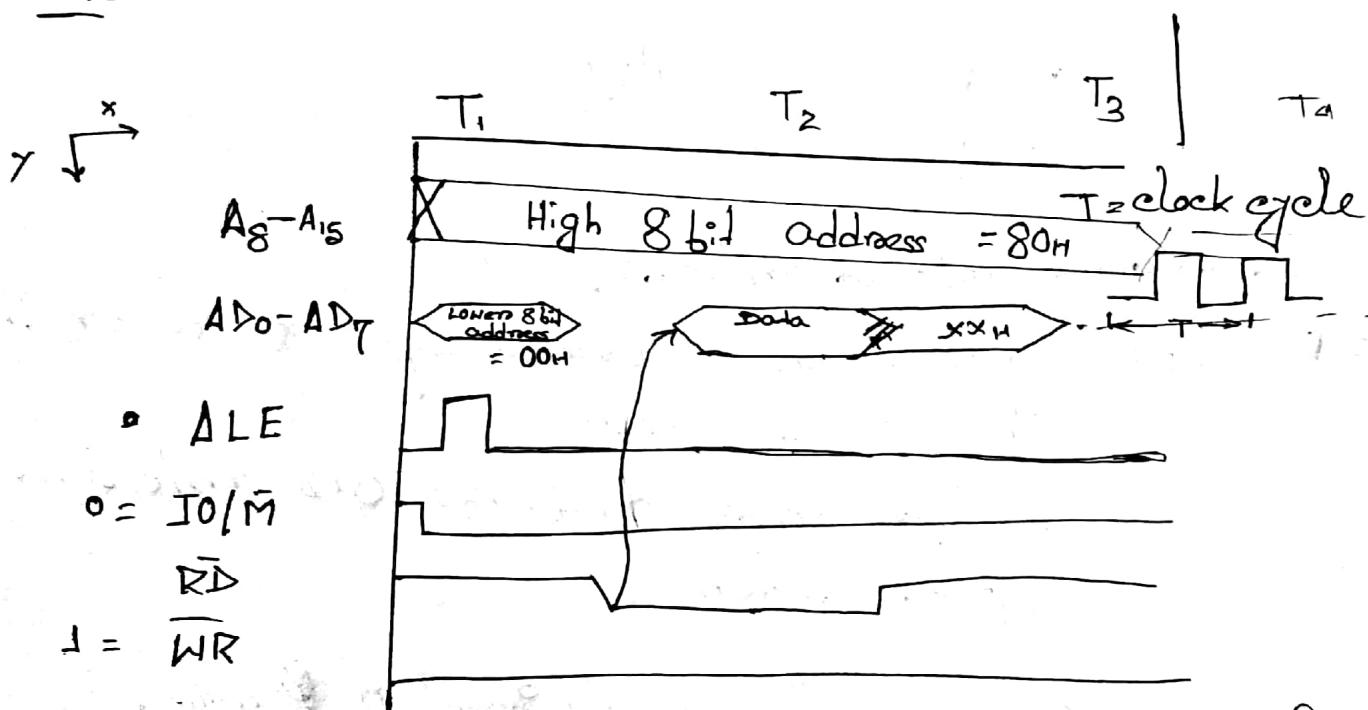
opcode

data

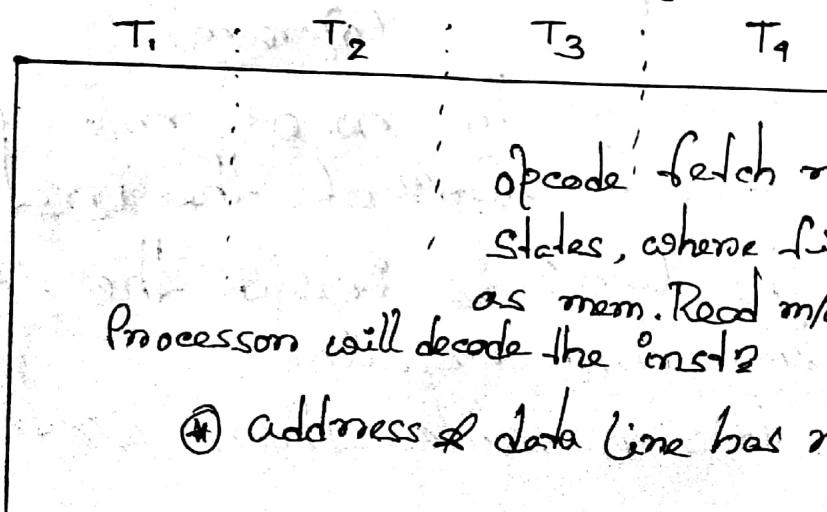
opcode $\Rightarrow 8050H \rightarrow$ one m/c

23H $\Rightarrow 8051H \rightarrow$ mem Read m/c

Memory Read:

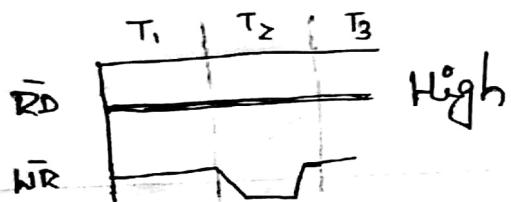


Opcode fetch: In this m/c step reads the opcode of an instⁿ from memory & decode the instⁿ



Memory write \Rightarrow In this m/e up will write data to a mem. loc.

- (ii) It is same as same as mem.read m/e but WR signal will be active during $T_2 T_3$ State in place of RD



Has to draw m/e of an instⁿ.

\Rightarrow Step 1 : calculate the no. of m/e cycle based on the size of the instⁿ
one byte instⁿ requires one m/e i.e.
Op-code fetch

2 byte instⁿ requires 2 m/e. i.e
Op. fetch - mem. Read.

3 byte instⁿ requires 3 m/e i.e.
Op. fetch - mem. Read - mem. Read
(operand)

Step 2 : calculate the no. of m/e require based on the lenⁿ of the instⁿ
if the lenⁿ is inside the if no extra m/e require
if it's outside then extra m/e

reduced based on the funcⁿ of the instⁿ

Prob 1 Draw the timing diagram of MOVA,B.
Let the instⁿ is stored at M.L. = 8000H

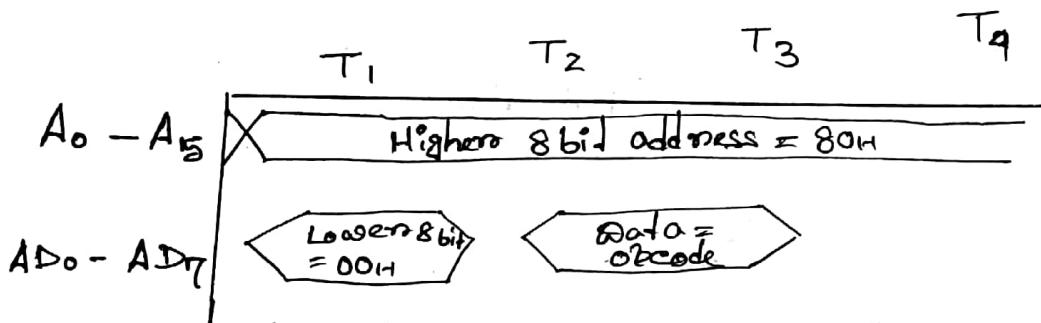
Step:1

MOV A,B \Rightarrow 1 byte instⁿ = opcode fetch = 8000H

op.fetch m/c reducing AT state

Step:2

B,A are inside the reg so no extra m/c



Prob 2 Draw the timing diagram of ADD L.

Let the instⁿ is stored at C050H

Step:1:

ADD L \Rightarrow 1 byte instⁿ = opcode = C050H

opcode fetch m/c

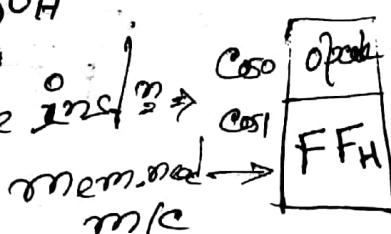
Step:2: no external m/c is here

Prob 3 Draw the timing diagram of MVIC,FFH

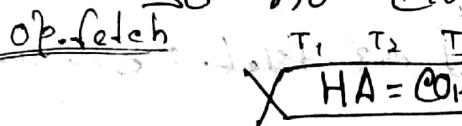
Let the instⁿ is stored at C050H

Step:1 :

MVIC C,FFH \Rightarrow 2 byte instⁿ \Rightarrow C050, C051
mem. read \rightarrow m/c



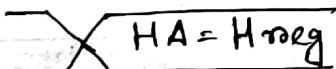
Step 2: As FF_H was outside but now inside rep
so no extra m/e.



Prob 4 ANA M

Step 1 $ANA \ M = 1 \text{ by } 1 \text{ bit inst} \Rightarrow C050_H$ Op. fetch

" 2 Func - Outside - M.R. m/e

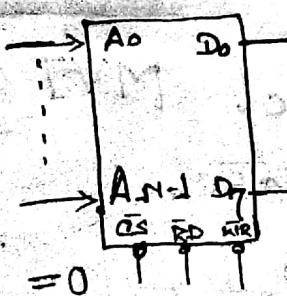


Prob 5

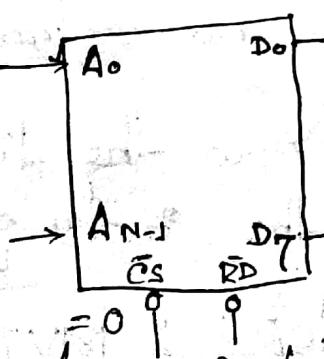
- | | |
|-------------|-----------|
| ① LDA 8050H | ④ IN 20H |
| ② STA C000H | ⑤ OUT FEH |
| ③ ANI F0H | |

Memory Interfacing: (Connect mem. chip to esp)

RAM



ROM



$CS \Rightarrow \text{chip Select}$

$A = A_{N-1}$
= Address line

$D_0 = D_T$
= Data line

To select a RAM or ROM
I/P to $CS = 0$
read $RD = 0$ $\bar{WR} = 0$
write

How to calculate n value?



Array size = $R \times C$ ← Content of first
memory location
No of
memory location

$$RAM \Rightarrow 4 \text{ GB}$$

$$= 4G \times \underbrace{\text{Byte}}_{R} \underbrace{\text{}}$$

$$= x \times 8$$

$$4G = 4 \times 1024 \text{ M}$$

$$= 4 \times 1024 \times 1024 \text{ K}$$

$$= \underbrace{4 \times 1024 \times 1024 \times 1024}_{x}$$

Binary

$$1 \text{ bit} = 0/1 = 2 = 2^1 \quad 3 \text{ bit} = 0-9 = 2^3$$

$$2 \text{ bit} = 2^2 = 4 \quad 2^{16} = 64 \text{ KB} = 65536$$

$$4G \rightarrow 2^{\text{?}} = N$$

max memory size
of 8085 up

16 kB

$$= 16 \text{ K} \times B$$

$$= 16 \times 1024$$

$$= 2^4 \times 2^{10} = 2^{14} \text{ or}$$

Step-1 Draw the block diagram of RAM/ROM memory chip.

Step-2 Draw the address, data & control lines of up
($\text{IOA}, \text{RD}, \text{WR}$)

Step-3 Connect the address lines of RAM/ROM
memory chip to the address bus of 8085 up

Standing from Ao address line for last obit

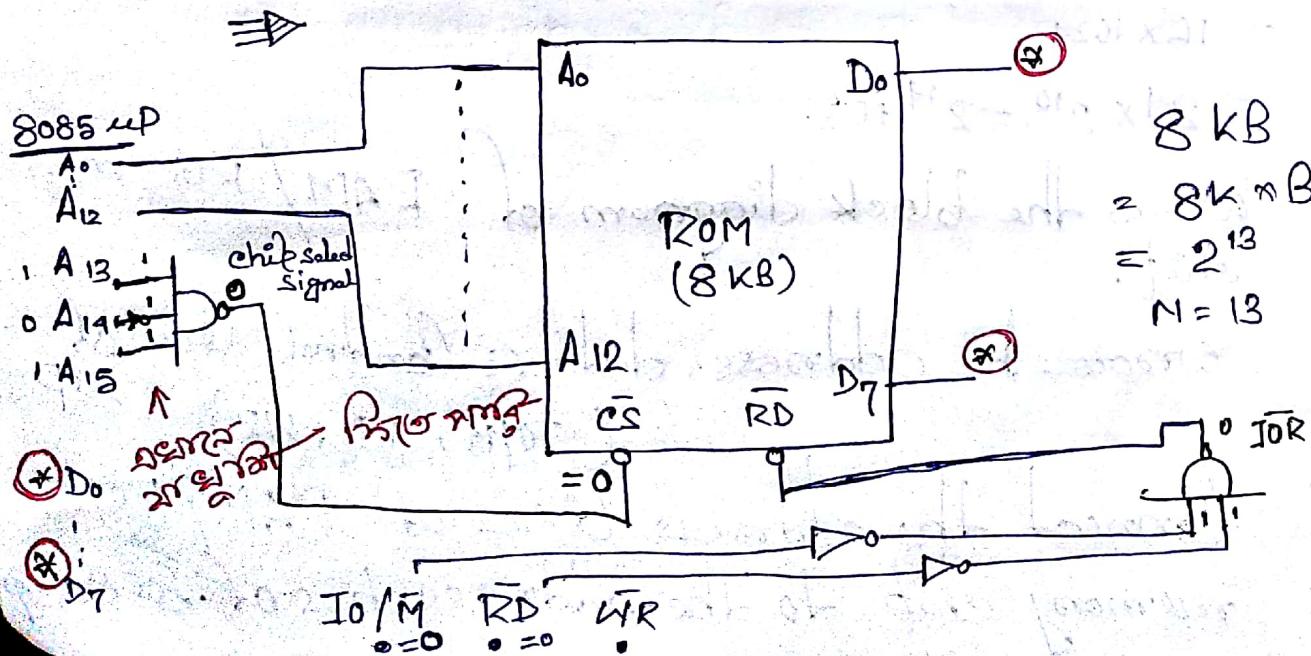
Step-4 Decode or combine the remaining address lines of 8085 up (using NAND gate / 3:8 decoders) to 2⁶ generate chip select signal & connect it to the CS line of ~~the~~ memory chip.

S-5 Combine one decode $\bar{I}O/M$ and \bar{RD} to generate \bar{IOR} signal and connect it to \bar{RD} of memory chip.

S-6 only for RAM memory

Combine one decade I_0/M and \bar{WR} lines to generate I_{0W} control signal & connect it to WR line of RAM chip.

Prob1 Design an interface b/w 8085 uP & One 8 KB ROM mem. chip.



X ₁	1	Vcc
X ₂	2	HOLD
RESET OUT	3	HLDA
SOD	4	CLK (out)
SID	5	RESET IN
TRAP	6	READY
RST7.5	7	I/O/M
RST6.5	8	S ₁
RST5.5	9	RD
INTR	10	WR
INTA	11	ALE
AD ₀	12	S ₀
AD ₁	13	A ₁₅
AD ₂	14	A ₁₄
AD ₃	15	A ₁₃
AD ₄	16	A ₁₂
AD ₅	17	A ₁₁
AD ₆	18	A ₁₀
AD ₇	19	A ₉
Vss	20	A ₈

8085 up

Prob 2 Find the address and data bus of 2048×512 ram chip.



$$2^n = 2048 = 2^11$$

$$n = 11$$

$$2^10 \times 2^10 = 1024 \times 2 = 2048$$

No of address line = 11
Hence 512 data lines needed.

Prob 3 Find the length of address & data bus of ~~2048 × 512~~ ram chip

$$1024 \times 1024$$

$$2^n = 1024 = 2^{10} \quad \text{No of add. line} = 10 \\ \text{data line} = 1024$$

$$\boxed{\text{No of chips} = \frac{\text{m/m to be designed}}{\text{Available Capacity}}}$$

Prob 4 Find no. of 256×8 Rom chip required to design 8 kB of memory.

$$\begin{array}{c} 256 \times 8 \\ \downarrow \quad \text{for data line} \\ 28 \quad 8 \end{array}$$
$$\text{No of chips} = \frac{8 [2^{10} \times 8]}{28 \times 8} = 32$$

512x8
chips

You have to design 1KB m/m. How much m/m chip we're to take.

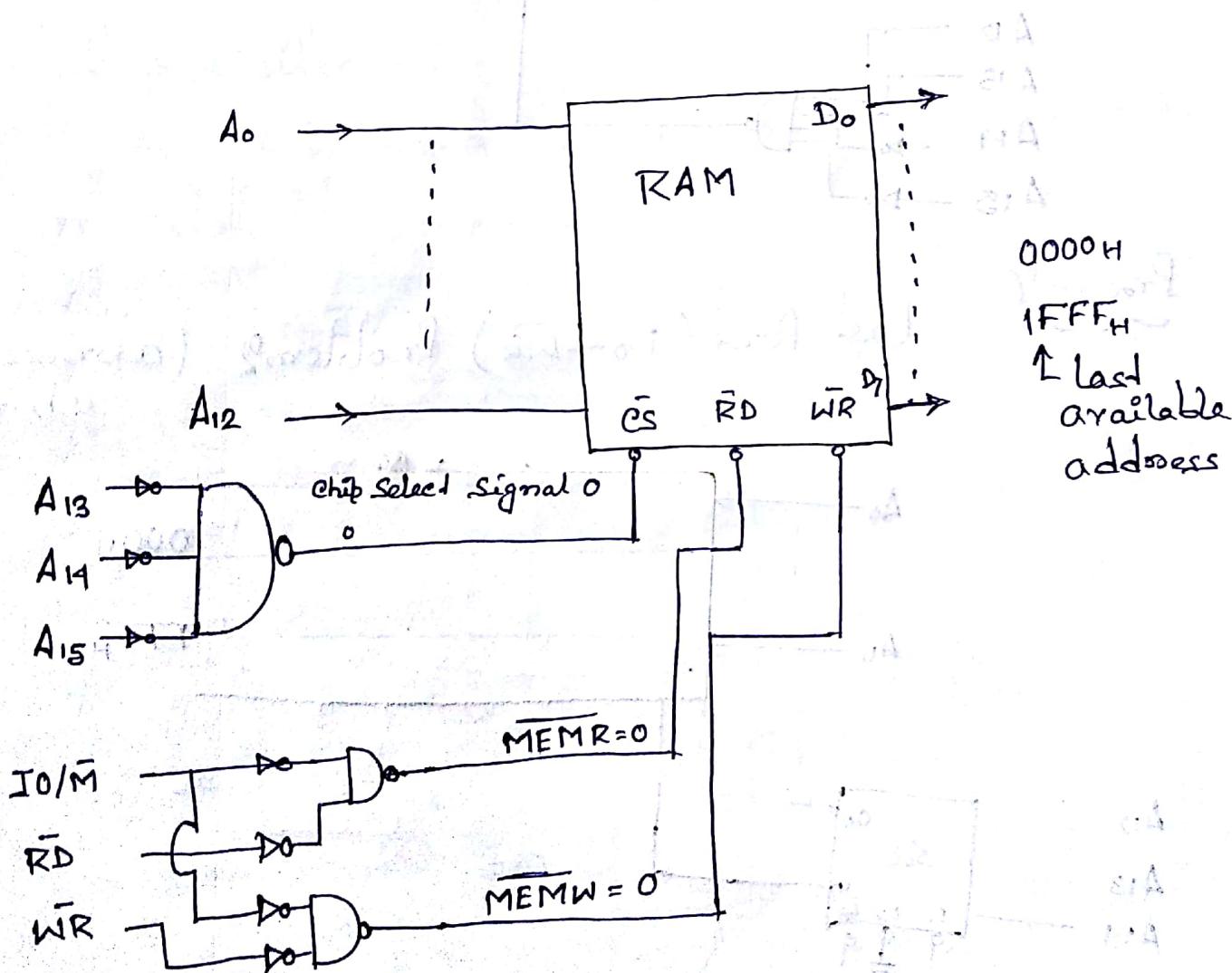
$$\Rightarrow \text{No of chips} = \frac{1024 \times 8}{512 \times 8} = 2$$

Prob-5 Interface one 8 KB RAM with 8085 CPU, where the starting address of memory is $1000H$.

$$\text{Size } 8 \text{ KB} = 8 \text{ K} \times \text{Byte}$$

$$= 2^3 \times 2^{10} = 13$$

~~$N = 13$~~ hence



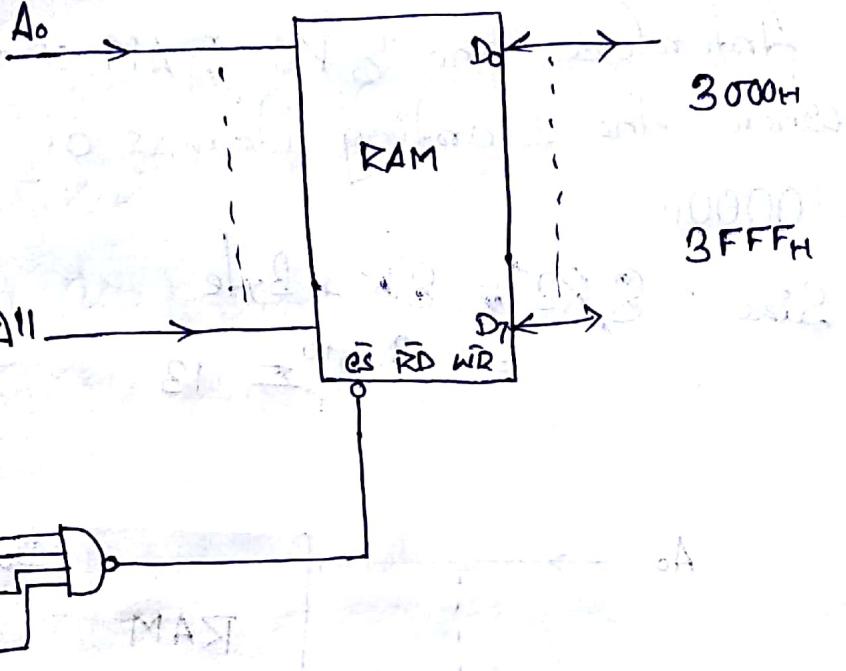
Prob-6

Interface one 4KB RAM with 8085 up.

Calculate the memory map.

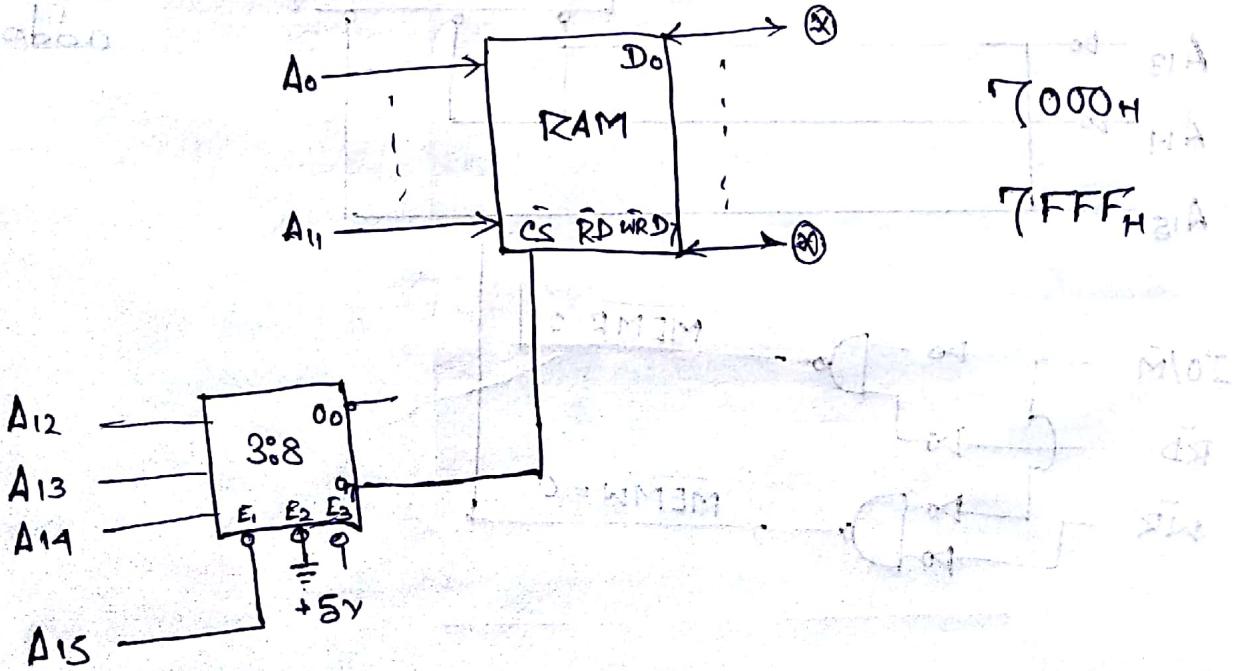


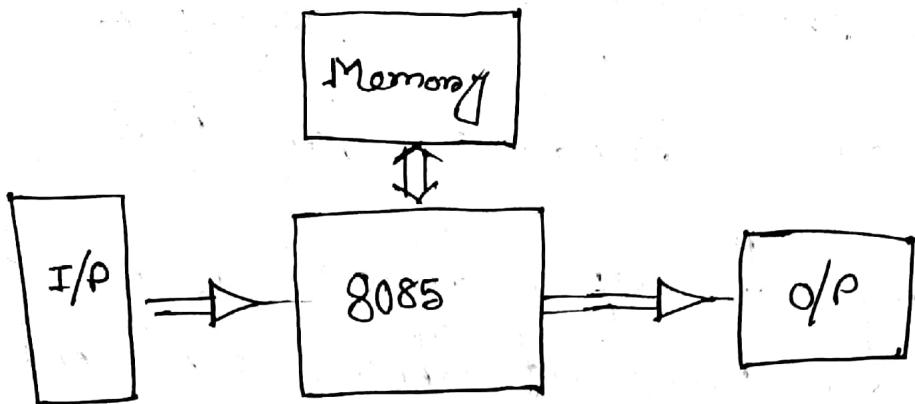
$$4 \text{ KB} = 4 \text{ K} \times 8 = 2^{10} \times 2^3 \Rightarrow 12 = n$$



Prob-7

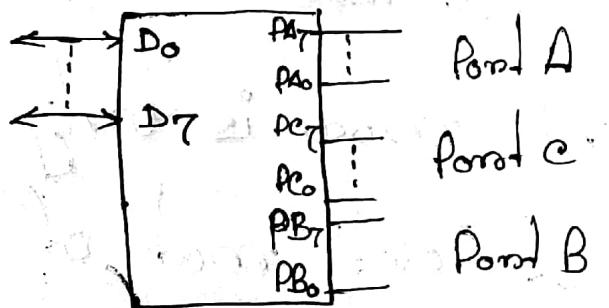
(last part) (D₀ - WR) Problem 2 (address no change)





8255 - Programmable Peripheral Interface

- Main features:
- ① It's designed by Intel
 2. It's called "PPI"
 3. It has 8 bit data bus to receive or to transmit data from or to I/O
 4. It has 24 I/O lines.



The 24 I/O lines are divided into 3 Port —

- ① Port A
- ② Port B
- ③ Port C

where each has 8 I/O lines

⑥ Port C is divided into again 2 Port —

Port lower & Port C upper.

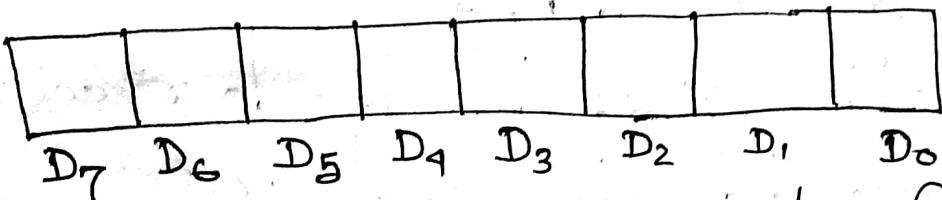
each has 4 lines $PC_0 - PC_3 = PC_L$
 $PC_4 - PC_7 = PC_U$

Pond A & Pond Upper = Group A

Pond Lower & Pond B = Group B

7. To Control the operation of 8255 there is register present there which is called CWR

8. CWR \Rightarrow Control Word Register
It's a 8 bit register.

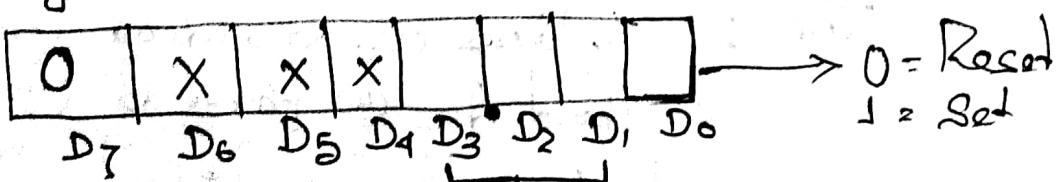


D₇ bit \Rightarrow 0 - It's called BSR mode of operation
1 - I/O mode of operation

BSR mode: 0 \sim This mode is only for Pond

(i) "Bit Set-Reset" mode of operation.

CWR reg.



Select one line

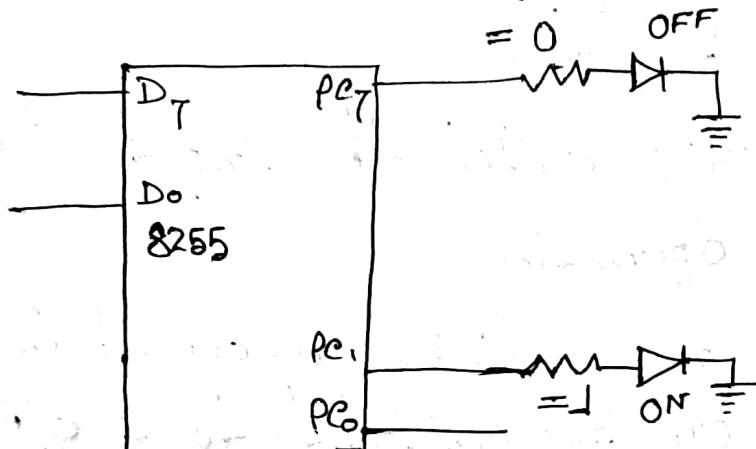
or Pond C

0 0 0 \rightarrow PLO

0 0 1 \rightarrow PC₁

... \rightarrow PC₇

Prob-1 Two Leds are Connected as PC₁ & PC₇ line, write a program to turn on the Led connected at PC₁ & turn off the led " " PC₇.



CWR \rightarrow Set PC₁

0	0	0	0	0	0	1	1
D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
as dont come						for PC ₁	

Reset PC₇

0	0	0	0	1	1	1	0
D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
as dont come						for PC ₇	

Program

MVI A, 03_H \rightarrow Load CWR in Accumulator

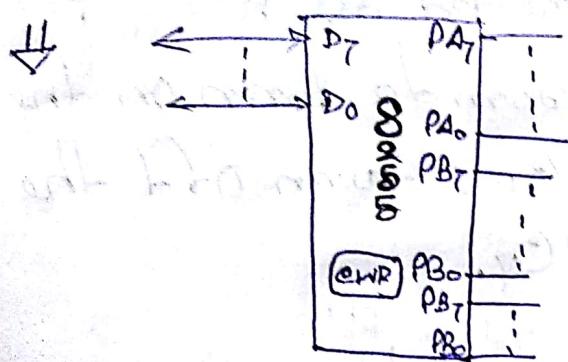
OUT CWR \rightarrow [A] to CWR of 8255

MYI A, 0E_H

OUT CWR

HLT

8085-14P



BSR mode is only for Port C

I/O \Rightarrow It's called the Input-Output mode of operation.

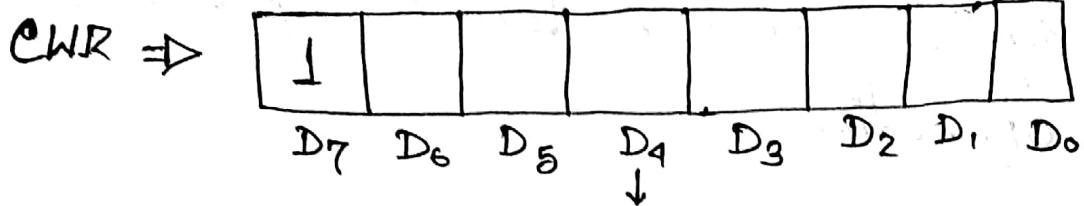
- ① In this mode 8255 received data from i/o device on transmit data to a o/p device.
- ② There are 3 different types of I/O mode - mode 0, mode 1, mode 2

③ Mode 0 \Rightarrow It's called the simple data transfer mode of operation. This mode is for Port A, B, C.

Mode 1 \Rightarrow It's called I/O mode with handshake signal. This mode of operation for Port A & Port B only.

Mode 2 \Rightarrow It is called bidirectional mode of operation. It is only for Port A.

I/O mode:



D₄: Func Selection for Port A.

if $D_4 = 0$: Port A will act as output

$D_4 = 1$; " " " " " " Input

D₅, D₆: Mode Selection for Port A

<u>D₆</u>	<u>D₅</u>	← Mode 0
0	0	← Mode 0
0	1	← Mode 1
1	x	← Mode 2

D₃: Port C upper Func Selection

$D_3 = 0$ — output

= 1 — Input

D₁: Func Selection for Port B

$D_1 = 0$ — o/p

= 1 — I/P

D₂: Mode Selection for Port B.

if $D_2 = 0$ Then Mode 0

$D_2 = 1$ " " 1

D₀: Port C lower Func Selection.

0 — o/p 1 — I/P

Programing the CW for 8255 where Port A is Input, Port B output & Port C Input in Mode 0 of operation

1	0	0	1	1	0	0	1
D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀

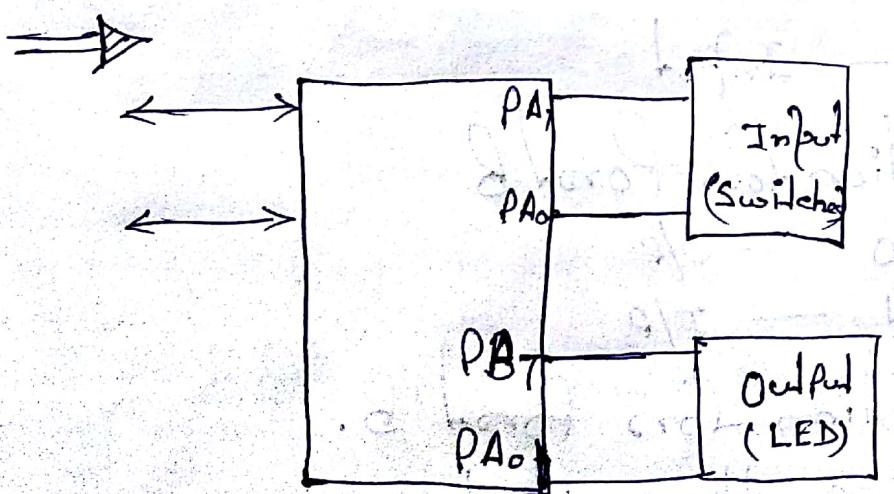
CW \Rightarrow 1 0 0 1 1 0 0 1

MVI A, 99_H 99_H

OUT CWR address ← Transfers data from accumulators to CWR

Mode 0 of operation:

example: Interface 8 no. of switches to Port A and 8 LEDs to Port B. Write a program to read the data (status of the switches) from I/P device. And transfers the result to the LEDs on to the O/P device



CWR Same as the above

MVI A, 99_H
OUT CWR address
IN Port A "
OUT Port B "
HLT