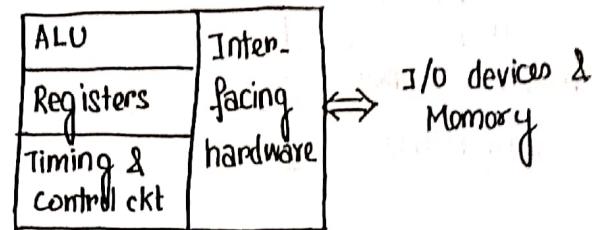


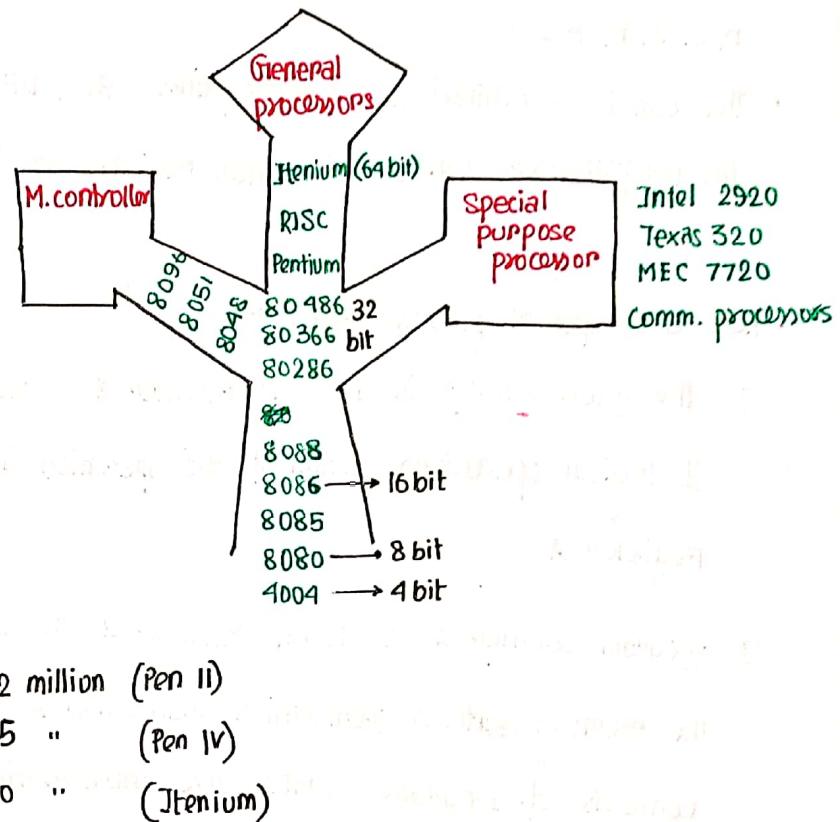
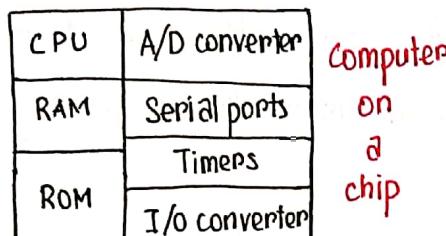
(SB)

MP is basically a CPU with many components —

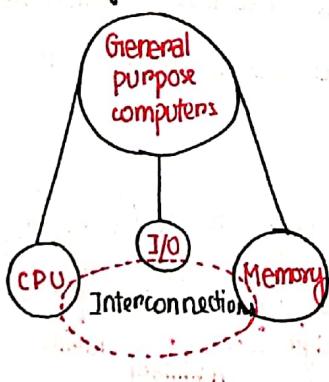
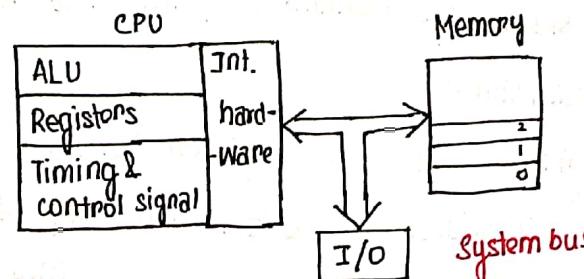
- Arithmetic Logic Unit (ALU)
- Set of Registers
- Timing & control circuit
- Interfacing hardware



CPU on a chip

MicrocontrollerAdvantages

- small size
  - Cost
  - Reliability
  - Power consumption
  - Flexibility
  - Powerful
- [2250 transistors (4004) → 42 million (Pen II)  
55 " (Pen IV)  
320 " (Itanium)]

Architecture & Organization of 8085Von Neumann architectureRegister Section of 8085

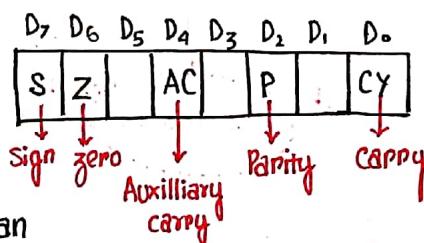
In 8085 most registers are of 8 bit. There are 2 types —

- General purpose register (6)
- Special

Accumulator	Flag Register
B	C
D	E
H	L
Program Counter	
Stack Pointer	

16 bit  $\rightarrow$  16 address lines

- 8085 has 6 general purpose registers to store 8 bit data. They are termed or identified as B, C, D, E, H & L.
- The can be combined as register pairs BC, DE & HL to store address and 16 bit operations. The programmers can use these registers as & when reqd. through instructions.
- Rest are special purpose registers.
  - The accumulator is a 8 bit register & is used to store 8 bit data to perform arithmetic & logical operations. Result of the operation is stored in accumulator & identified as register A.
  - Program counter is a 16 bit register & is used to store addresses. It points to the memory address from which instruction(s) are to be fetched. Once it is fetched the contents of program counter are incremented by 1 which points to the next memory location.
  - Stack pointer is a 16 bit register & it points to another memory location in R/W memory for the stack. The beginning of the stack is defined by loading a 16 bit address in the stack pointer.
  - Flag register is a 8 bit register but only 5 flags are individually used to indicate diff. status of an operation done by the ALU. They are known as flags. These are set or reset after an operation acc. to cond's of the result in the accumulator & other registers. Microprocessors use these flags to test data cond's. These flags decide the next instruction to be executed.



e.g. If 0 flag is set, then a particular instruction is to be executed & some other instruction is to be followed.

Cond's under which flags go to set state :

- The Z flag is set to 1 when an ALU operation results in 0. (All 8 bits of accumulator is zero)
- If an arithmetic op. operation results in carry-the CY flag is set, else it is reset.
- The S flag is set if bit D<sub>7</sub> of the result is 1, else it is reset.
- The P flag is set if result has even no. of 1's and reset for odd no. of 1's.
- In an arithmetic op. when carry is generated by digit D<sub>3</sub> & pushed to digit D<sub>4</sub> The AC flag is set. The flag is used internally for BCD operations.

List of op.s ALU section can perform:

Addition

Subtraction

Increment

Decrement

OR CLEAR

AND COMPARE

XOR SHIFT

NOT ROTATE

} Arithmetic operations

} Logical operations



System bus : The primary 4 op.s done by Microprocessor unit (MPU) —

1. Memory read : Processor reads data from memory

2. " write

3. I/O read

4. I/O write

The 4 major op.s are performed through the foll. steps—

1. Identify the peripheral and memory locations.

2. Transfer the binary info.

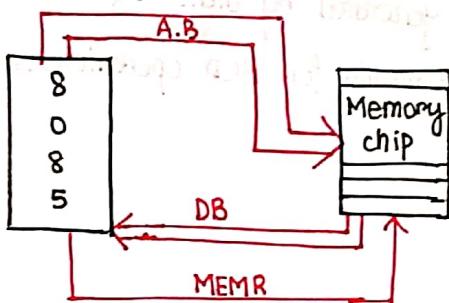
3. Provide timing & synchronisation signals

MPU performs these op.s using 3 states of comm. lines called buses.

Address Bus → It is a group of 16 lines which generally identify A<sub>0</sub> to A<sub>15</sub>. It is unidirectional, bits flow from MPU to peripherals (memory or I/O).

The data bus is a loop of 8 lines used for data flow & are bidirectional in nature.

Control bus though termed as bus actually consists of a no. of discrete signal lines. These lines carry synch. signals & timing signals. The MPU generates pulse for diff. ops like memory read, memory write etc. & these pulses are being carried to appropriate peripherals through these control lines.



### Languages of MP

Machine Language : Instructions given in binary.

8085 has 74 instructions in total & it is called Instruction set of 8085. Instruction can be given using a Hex Keyboard & the instructions written in Hex board is called Machine code.

Assembly Language : Used In order to facilitate programming MP have devised English type words to represent binary instructions. Programs using these words are called assembly language program. Symbolic code for each instruction is called mnemonic.

Instruction : It is a command to the processor to perform a given task on specified data. There are 2 types —

1. Operation code (op code) - Task to be performed.

2. Operand - The data to be operated on

Depending upon word size, 8085 I.S. can be classified into 3 categories —

1. 1 Byte Instruction

2. 2 " "

3. 3 or more " " (Instructions having longer length of operation code)

<u>Task</u>	<u>Op code</u>	<u>operand</u>	<u>Hex code</u>
• Copy contents of accumulator in register C	MOV	C,A	4F H
• Add contents of register B to contents of accumulator	ADD	B	80H
2 bytes • Load the data in the accumulator	MVI	A,32H	3E 32H
3 bytes • Load contents of memory 2050 Hex into accumulator	LDA	2050H	3A 50H
		2050H	20H

### INSTRUCTION SET OF 8085

#### Data Transfer or Copy

This instruction copies data from a source to a destination, without modifying the contents of the source.

<u>Op code</u>	<u>Operand</u>	<u>Description</u>
MOV	Rd, Rs	<ul style="list-style-type: none"> <li>• 1 byte data in R<sub>d</sub> copy to R<sub>s</sub></li> <li>• copies data from source R<sub>s</sub> to destination R<sub>d</sub></li> </ul>
MVI	R, 8 bit	<ul style="list-style-type: none"> <li>• 2 byte data</li> <li>• Loads the 8 bits in the specified register</li> </ul>
OUT	8 bit port address	<ul style="list-style-type: none"> <li>• 2 byte ins.</li> <li>• sends the contents of accumulator to the output port mentioned in the 2nd byte</li> </ul>
IN	"	<ul style="list-style-type: none"> <li>• 2 byte ins.</li> <li>• Accepts data from input port specified in the 2nd byte</li> <li>• loads into A.</li> </ul>
HLT	-	<ul style="list-style-type: none"> <li>• 1 byte ins.</li> <li>• Processor stops executing &amp; enters a wait state</li> </ul>
NOP	-	<ul style="list-style-type: none"> <li>• 1 byte ins.</li> <li>• When an error occurs in a prog. &amp; an inst. needs to be eliminated it is more convenient to substitute NOP than to reassemble the whole program.</li> </ul>

▷ Load Register B with 37H

▷ copy the no. from B to A

▷ Send " " to the O/p port 01H

▷ End of program.

### Memory Address

### Mnemonics

### Hex code

2000

MVI B, 37H

06

37

2001

MOV A,B

37

2002

OUT PORT1

78

2003

D3

2004

01

2005

HLT

76

## ARITHMETIC OPERATIONS

1. Arithmetic instructions accepts JNR & DCR

▷ Assume implicitly that accumulator is one of the operands

▷ Modify all the flags acc. to the data cond's of the result

▷ Place the result in the accumulator.

▷ The ins. INR & DCR affects all flag except carry flag.

### Op Code

### Operand

### Description

**ADD**

R

• Add contents of register R to A.

**ADBI ADI**

R

• Adds the 8 bit data to contents of A.

**SUB**

R

• Subtracts contents of R from contents of A.

**SUI SUI**

R

• 8 bit data from contents of A

**INR**

R

• Increases contents of R by 1.

**DCR**

R

• Decreases ..

A = 93H

CY : D<sub>7</sub> D<sub>6</sub> D<sub>5</sub> D<sub>4</sub> D<sub>3</sub> D<sub>2</sub> D<sub>1</sub> D<sub>0</sub>

C = B7H

: 1 0 0 1 0 0 0 1

: 1 0 1 0 1 0 0 1

Ans: 14A1

1 0 1 0 0 1 0 1 0

CY = 1 (Ans) with Remainder 0, HLL 904

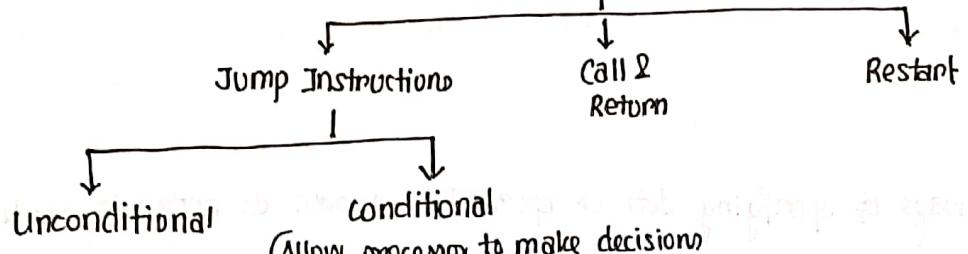
Z = 0 (All digits are not zero)

S = 0 (D<sub>7</sub> = 0)

<u>Op code</u>	<u>Operand</u>	<u>Description</u>
Masking bits $\rightarrow$ ANA	R	• Logically AND's contents of R with A
ANI	8 bit	• " " AND 8 bit data to A
Setting bits $\rightarrow$ ORA	R	• ORS contents of R with A
ORI	8 bit	• " " OR 8 bit data to A
XRA	R	• XORs contents of R with A
XRI	8 bit	• " " XOR 8 bit data to A
CMA		• NO flags are affected

$$\begin{array}{r} A = 1001\ 1000 \\ \text{AND} \quad \underline{1001\ 0000} \\ \hline 1001\ 0000 \rightarrow \text{Masked} \end{array}$$

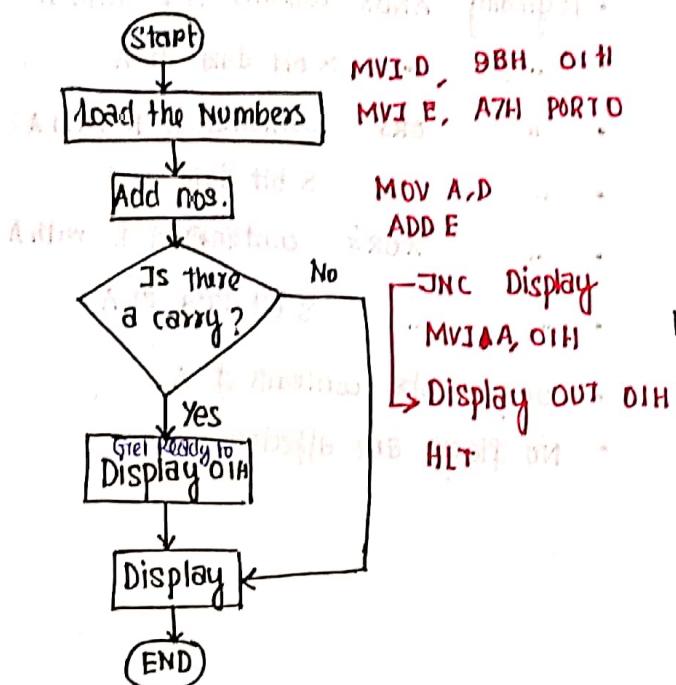
### Branch operations



④ It checks the flag cond's & make decision to change or not to change sequence of the program

	<u>Op code</u>	<u>Operand</u>	<u>Description</u>
	JMP	16 bit (memory address)	Instructs processor to go to memory location specified by 16 bit.
Carry	JC	"	Jump on Carry (if CY=1)
	JNC	"	Jump on NO Carry (if CY=0)
Zero	JZ	"	Jump on Zero (Z=0)
	JNZ	"	Jump on No Zero (Z=0)
Sign	JP	"	Jump on Plus (S=0)
	JM	"	Jump on Minus (S=1)
Parity	JPE	"	Jump on Parity Even (P=1)
	JPO	"	Jump on Parity Odd (P=0)

Load the hexa dec. nos. 9BH & 2 A7H to D & E. Add the sum. If SUM > FFH, display 01H at o/p PORT 1 or display the sum.



MVI D, 9BH, 01H  
MVI E, A7H, PORT 0

MOV A,D  
ADD E

JNC Display

MVI A, 01H

Display OUT 01H  
HLT

9BH → 1001 1011

A7H → 1010 0111

$$\begin{array}{r} 10100010 \\ \hline 142 \end{array}$$

FFH → 1111 1111 H

Q Identify the no. of bytes of the following instructions:

DCX SP → stack pointer → 1 byte

ADI 42H → 2 byte

JNZ 6151 → 3 byte

Addressing Modes : The various ways of specifying data or operand is known as addressing mode

1. Immediate

MVI B, 20

2. Register

ADJ C, 01

3. Discrete Direct

MOV B,C

4. Indirect

IN/OUT

5. Implicit

LDA 16 bit (8050)

STA 16 bit

MOV R,M (HL)

LDAX B (BC)

CMA

RRC

Q Specify the o/p at PORT 1 if foll. prog. is executed -

MVI B, 82H      B ← 82

MOV A,B      A ← 82

MOV C,A      C ← 82

MVI D, 37H      D ← 37H

OUTPORT I      82

HLT

specify the register contents & status of flag registers (S, Z, CY) as the foll instructions are execute

O/p at PORT 0		$D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0$	S $\rightarrow$ 0
MVI A, F2H	A $\leftarrow$ F2H	1111 0010	Z $\rightarrow$ 0
MVI B, 7AH	B $\leftarrow$ 7AH	0111 1010	CY $\rightarrow$ 1
ADD B	A $\leftarrow$ 16CH	<u>10110</u> <u>1100</u>	
OUTPORT 0	16E 6C	1 6 C	
HLT			

Q Write a ALP to add 2 hex nos. A2H & 18H. Both no. should be saved for future use

MVI ~~A~~, A2H  
MVI ~~B~~, 18H  
~~ADD B, C~~ MVI B  
ADD C  
OUTPORT 0  
HLT

Q Write ALP to add 2 hex nos. 34H & 48H & disp. ans. at output port 1

MVI A, 34H  
ADJ, 48H  
OUTPORT 1  
HLT

Q Load the data byte A8H in C. Mask the high order data bits ( $D_7 - D_4$ ) & display low order bits ( $D_3 - D_0$ ) in outport.

MVI C, A8H  
MOV A, C  
ANI #01  
OUTPORT 0  
HLT

$D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0$   
1010 1000  
AND  
0000 1111  
Masked  
0000 1000

16 bit data transfer to register bits

<u>Op Code</u>	<u>Operand</u>	<u>Description</u>
LXI	Rp	Load Register pair
LXI	B1	Byte Address (B1 C1)
LXI	D8 $\Rightarrow$ DC	
LXI	H $\Rightarrow$ HL	

Q Write instruction to load a 16 bit number 2050H in register pair HL using LXI & MVI op codes. Explain the difference b/w the two instructions.

LXI R1 H 2050H	$21 \{ 50 \} 20$	∴ LXI more advantageous
MVI H 20H	$26 \{ 20H \}$	: Time & memory saved
MVI L 50H	$2E \{ 50H \}$	

**MOV R, M :** It copies the data byte from memory location into the register.

Any instruction with operand **M** automatically assumes the **HL** register pair as the memory pointer.

Q Move the contents of **2050** to the **A** using ins. **MOV R, M**.

LXI H 2050H      

H	L
20	50

Mov A, M

**LDAX B/D :** It copies the data byte from the memory location into the **A**. The location is (indirect) specified by the contents of the register **B, C** or **D, E**.

**LDA :** 16 bit (Load Accumulator Direct)

It copies the memory location specified by 16 bit address into **A**.

Data Transfer microprocessor to memory

**MOV M, R :** It copies data from register **R** into the memory location specified by the contents of **HL** register

**STA X B/D :** (Store Accumulator Indirect)

It copies data from **A** into the memory location specified by the contents of either **BC / DE** register

**STA B/D :** (Store Acc. Direct)

Copies data from **A** into

**MVI M, 8 bit :** Copies 8 bit data into the memory location **HL** register.

Arithmetic operations related to 16 bits / register pairs

(This inst. do not affect stacks)

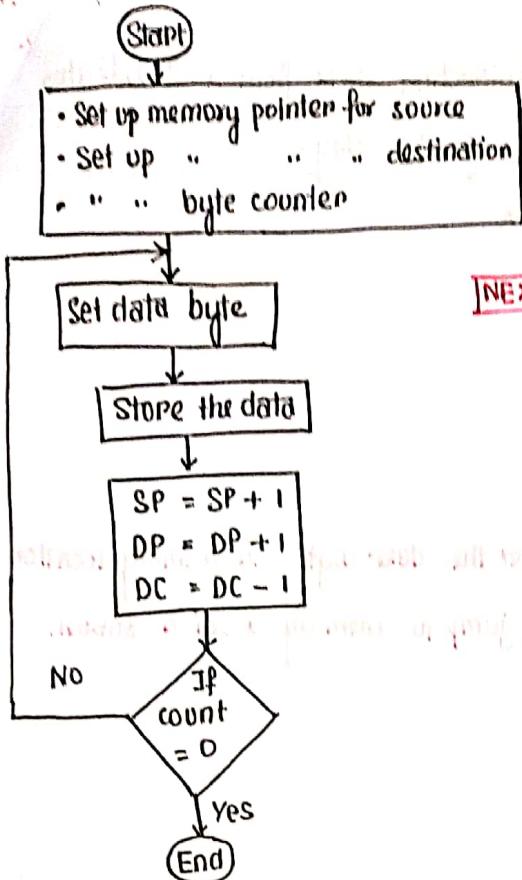
**INX Rp :** (Increment Register Pair)

Inc. the contents of 2 registers as one 16 bit no. & inc. the content by  $\pm 1$ .

**DCX Rp :** Dec. " 16 bit no. of the register pair by 1 .

Q 16 bytes data stored in a memory location **2050H** to **205F H**. Transfer entire block to new memory location starting at **2070H**.

See next page ↗



$\text{LXI H}, 2050\text{H}$   
 $\text{LDX D}, 2070\text{H}$   
 $\text{MVI B}, 10\text{H}$   
 $\text{MOV A, M}$   
 $\text{STAX D}$

$\begin{array}{|c|c|} \hline H & L \\ \hline 20 & 50 \rightarrow 51 \\ \hline \end{array}$   
 $\begin{array}{|c|c|} \hline D & E \\ \hline 20 & 70 \rightarrow 71 \\ \hline \end{array}$

$\begin{array}{|c|} \hline B \\ \hline 10\text{H} \\ \hline \end{array}$   
 $\begin{array}{|c|} \hline A \\ \hline \text{Do} \rightarrow D \\ \hline \end{array}$

$\text{INX H}$   
 $\text{INX D}$   
 $\text{DCH B}$   
 $\text{JNZ NEXT}$   
 $\text{HLT}$

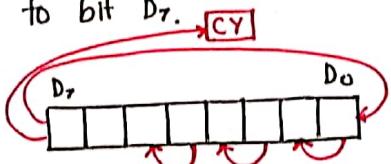
<u>Opcode</u>	<u>Operand</u>	<u>Description</u>
ADD	M	It adds the contents of memory location M specified by contents of HL register to the content of A and stores result in A.
SUB	M	Subtracts M from A & shows the result in A.
INR	M	It increments the contents of M specified by contents of HL register by 1.
DCR	M	Decrements the contents of M by 1.
RLC		

rotate

**RLC** → Rotate Accumulator Left

Each bit is shifted 1 bit left from its pos. i.e.  $D_7$  becomes  $D_0$  & carry flag

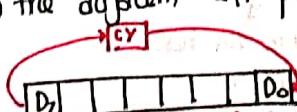
is modified acc. to bit  $D_7$ .



**RRC** → Right

**RAL** → Rotate Accum. Left through carry. Each bit is shifted to the adjacent left pos. &  $D_7$  becomes the cr bit & cr bit is shifted to  $D_0$ .

**RAR** → Right



## Logical Op. Compare

The processor compares data byte with contents of A & by subtracting data from A it indicates whether the data is ~~>~~,  $>$ ,  $=$ ,  $<$  than A by modifying the flags.

CMP R/M - compared data byte of register/Memory with A.

Case 1:  $A < (R/M) \Rightarrow CY=1, Z=0$

" 2:  $A = (R/M) \Rightarrow CY=0, Z=1$

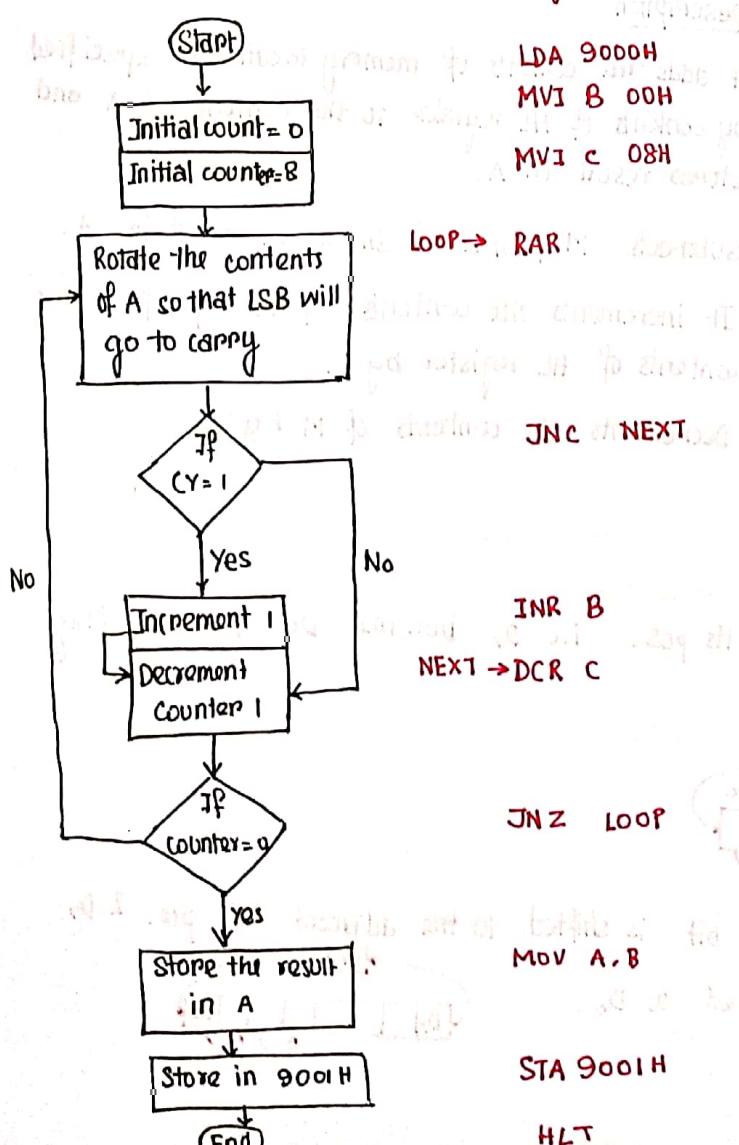
" 3:  $A > (R/M) \Rightarrow CY=0, Z=0$

CMI 8 bit : compared 8 bit data with contents of A.

Q Write inst. to load A with data byte 64H & verify whether the data byte in memory location 2050H is equal to contents of A. If both bytes are equal, jump to memory location 3050H.

<u>LXI</u> A 64H	LXI H 2050H	DRAM MSG
<u>EMP</u> R/M	MVI A 64H	
	CMP M	
	JZ 3050H	End

Q Count the no. of 1's at the memory location 9000H & store it in 9001H.



Add the corresp. no. of 2 array having 10 8bit nos. each & store them in a 3rd array. Use the COMPARE instr. to find the end of the array. Assume no carry is generated during summation process. The 1st, 2nd & 3rd array starts from 9000H, 9100H & 9200H resp.

LXI H 9000H

LXI B 9100 H

LXI D 9200H

LDAX B

Loop → ADD M

STAX D

INX H

INX B

INX D

MVI A,L

CPI 09H

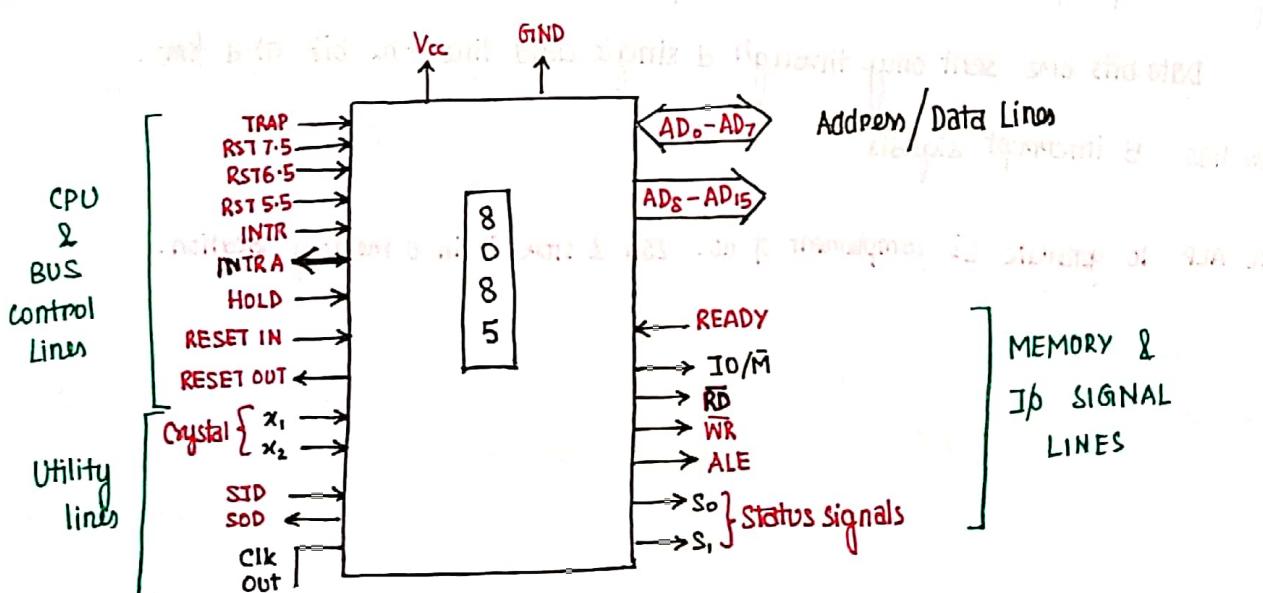
JNZ LOOP

HLT

### MICROPROCESSOR ARCHITECTURE : 40 pin IC Technology

IO/M → If it is high it indicates address for I/O with case of bonds of address bus or low case of bonds of memory

ALE → Address Latch Enable



To optimally use the no. of pins provided by processor sometimes same signal lines are used for dual purpose — address and data. ALE facilitates this func. If ALE is high, address is latched (i.e. locked) which is stored in some buffer & once latched the same lines can be used to transfer data. (AD<sub>0</sub> - AD<sub>7</sub>) are used for this purpose

RD → whenever this signal activated processor understands that selected I/O or Memory device is to be read & data are available on data bus

**WR** → This signal indicates that data on data bus is to be written into a selected memory I/O section.

**READY** → It is a signal that serves to delay the processor R/W signal until a slow responding peripheral is ready to accept/send data. If this signal goes low the processor waits for an integral no. of clock cycles until the READY becomes high.

**Status Signals** These are used to specify the kind of sigf operation being performed

Sometimes the proc needs to be interrupted so that the buses are relinquished for direct memory access data transfer. When this is activated by an ext. signal, the proc. gives up the control over buses & allows external peripherals to use them. The process is managed by a chip called DMA controller

**HOLD Acknowledge** → This signal is sent by proc. to peripheral, indicating that HOLD has been. **RESET IN** → When this pin is activated all internal operations are suspended & program counter is cleared to 0000 state.

**RESET OUT** → This signal indicates that MPU has been reset. It can also be used to reset any other peripheral connected to the proc.

Data bits are sent only through a single data line one bit at a time.

8085 has 5 interrupt signals

Q Write ALP to generate 2's complement of no. 23H & store it in a memory location.



J.R.Y.

Select chip

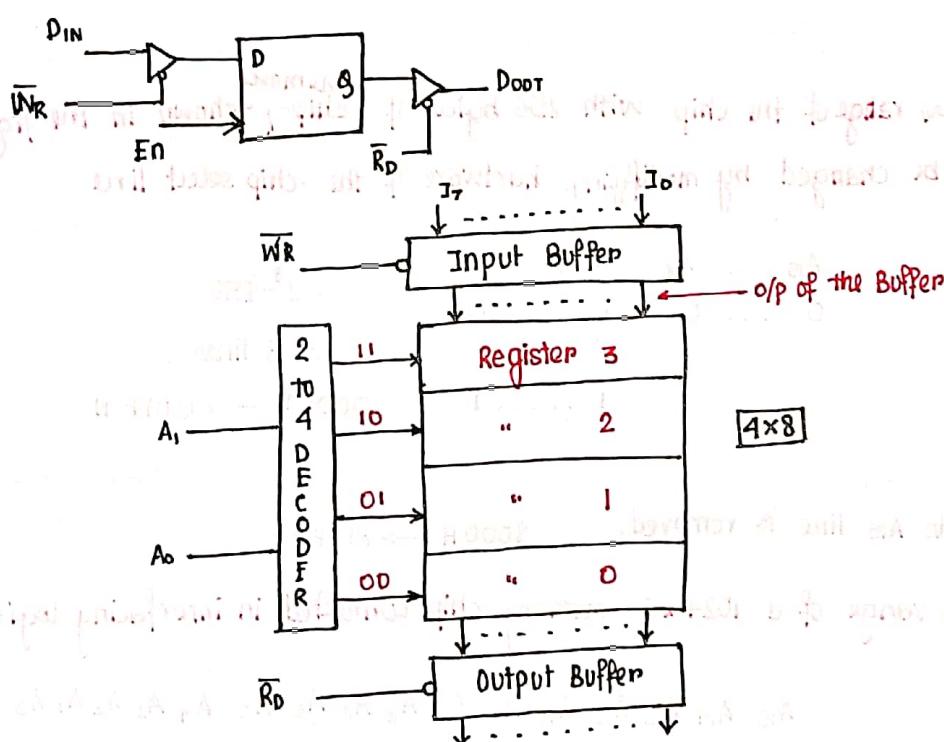
Select register

2 tristate buffers are connected to the latch, one at the i/p line & another at the o/p line. This is done in order to avoid unintentional change in the i/p line & control the availability of the data in the o/p line. Tristate buffers have 3 logic states — 0, 1 and high impedance.

When Enable is high, it is in the high impedance.

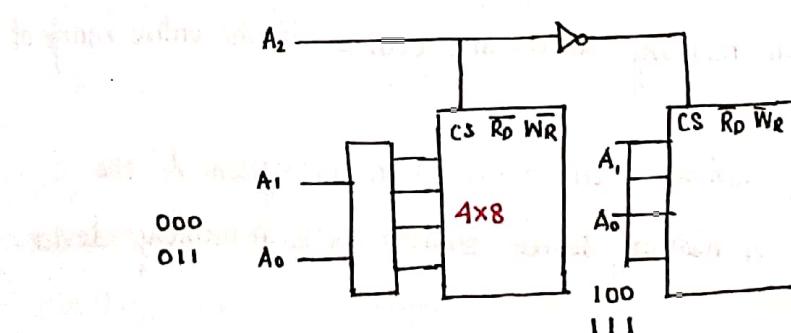
" " " low, it acts as a normal buffer.

The  $\bar{R}_D$  and  $\bar{W}_R$  signals are connected to the Enable pins. so that we can write into the latch by enabling the i/p buffer and read from the latch by enabling the o/p buffer.

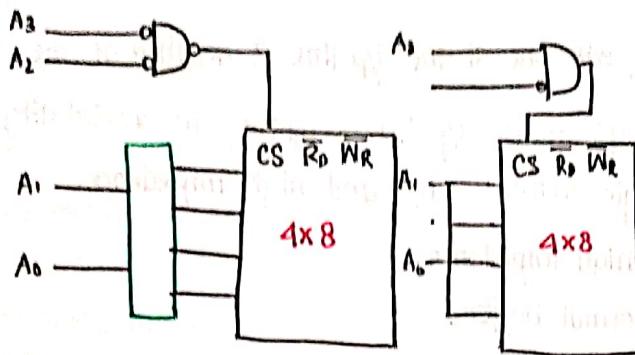


Q Explain the necessity of i/p & o/p buffers in a memory chip, with the help of a 4:8 memory chip.

I/p & o/p buffers are shown by 2 blocks, where the i/p and o/p lines are shown distinctly for both buffers. The  $\bar{R}_D$  and  $\bar{W}_R$  signals are connected in common to all the enable lines of the resp. buffers.



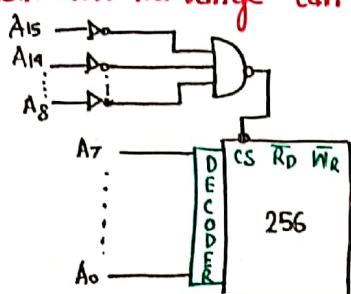
In order to select 2 chips.  $A_3 A_2 A_1 A_0$



The method in which all the address lines are used to generate unique addresses for the memory chip is called absolute decoding.

### ABSOLUTE DECODING (Contd)

Illustrate the memory address range of the chip with 256 bytes of <sup>memory</sup> chip shown in the fig. & explain how the range can be changed by modifying hardware of the chip select line

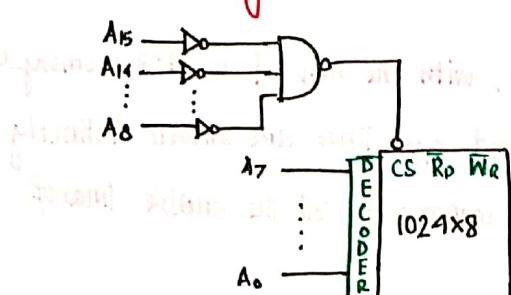


$A_{15} \dots A_8$	$A_7 \dots A_0$	$\therefore 2^8 = 256$
0 ... 0	0 ... 0	
1 ... 1	1 ... 1	$0000H \rightarrow 0FFFH$

The NOT gate connected to A15 line is removed.

$8000H \rightarrow 80FFH$

Explain the memory address range of a  $1024 \times 8$  memory chip connected in interfacing logic shown in diag.



$A_{15} A_{14} A_3 A_{12} A_{11} A_{10} A_9 A_8 A_7 A_6 A_5 A_4 A_3 A_2 A_1 A_0$	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0
	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	
		$0000H \rightarrow D3FFH$

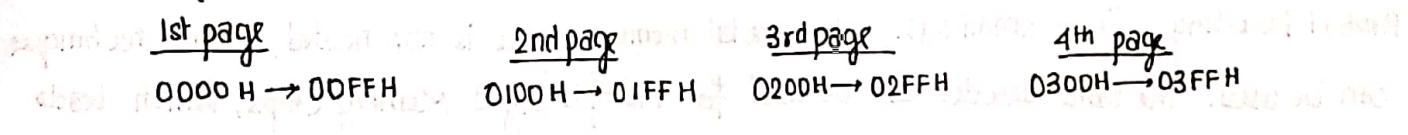
### Memory map & address

Memory map is a pictorial rep. in which memory devices are located in the entire range of addresses.

Memory address provide the location of various memory devices in the system & the interfacing logic defines the range of a memory device address for each memory device.

1K → 1024 registers

256 registers is considered as a page. One register → One line of the page  
1K Memory has 4 pages ( $\because 1024 \div 256 = 4$ ) with each page having 256 registers



$$2^x = 256$$

$$\Rightarrow \log_2^x = \log_2^{256}$$

$$\Rightarrow x \log_2 = \frac{\log 256}{\log 2}$$

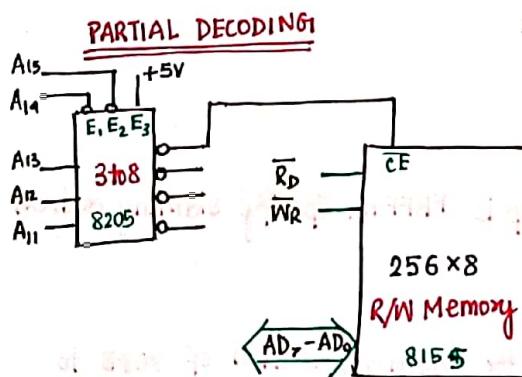
$$\Rightarrow x = 8$$

Q. Calculate address lines reqd. for an 8K byte memory chip

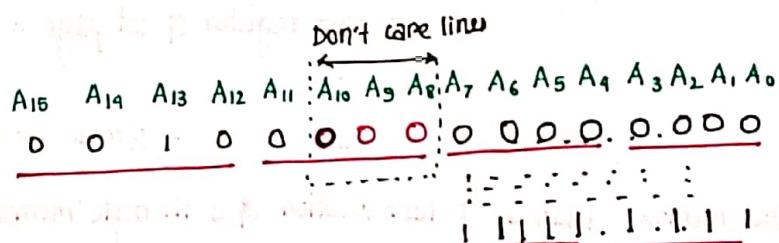
Q. " the no. of memory chips needed to design 8K byte memory if the memory chip

in size  $1024 \times 1$  was chosen for the required program and data memory

$$\begin{array}{c} 8 \\ \downarrow \\ 8 \end{array} \quad \therefore 8 \times 8 = 64$$



$A_8, A_9, A_{10} \rightarrow$  not used



$$2000H \rightarrow 20FFH$$

If other 8 bit comb's assigned to Don't care lines we have other address ranges. (e.g. 2100H → 21FFH ... etc.)  
Foldback Memory / Mirror Memory

In the fig., address lines  $A_8, A_9, A_{10}$  are not connected & thus they are don't care lines capable of assuming any logic state 0 or 1. These 3 don't care address lines can assume any of the 8 comb's from 0000H to 1111H. Thus each comb can generate one set of complete address, the address range given by assuming all don't care lines to be at logic '0' (2000-20FFH) is by convention specified as the memory address range of the system or the primary address. The rest ranges are Foldback / Mirror Memory. (2100H to 27FFH)

Attempting to store an info. in 2100H, 2200H or 2700H is the same as entering the instruction in location 2000H.

The address lines A<sub>8</sub>-A<sub>10</sub> were not decoded, resulting in Multiple Addresses. This is called Partial Decoding. In a small sys., where total memory space is not needed, such a technique can be used. The same decoder can be used for Multiple Sized Memory Chips, which leads to cost saving.

Q If the memory chip size is 1024 x 4 bits, how many chips are reqd. to make 2K bytes of memory?

∴ 8 bits = 1 byte

∴ 4 chips to make 2K bytes

Q How many address lines are reqd. on the chip of 2K byte memory?

$$256 \times 4 = 1024$$

$$1024 \times 2 = 2048$$

$$\therefore 2^n = 2048 \Rightarrow n=11$$

Q The memory map of a 4K byte memory chip begins at the location 2000H. How many no. of pages are there on chip? Find address of last location of the chip.

$$1K \rightarrow 4 \text{ pages}$$

$$4K \rightarrow 16 \text{ pages}$$

256 registers

∴ Last register of 1st page = 20FFH

$$\therefore 16^{\text{th}} = 2FFFH$$

Q The memory address of last location of a 1K byte memory chip is FBFFH. Specify starting address

F800H

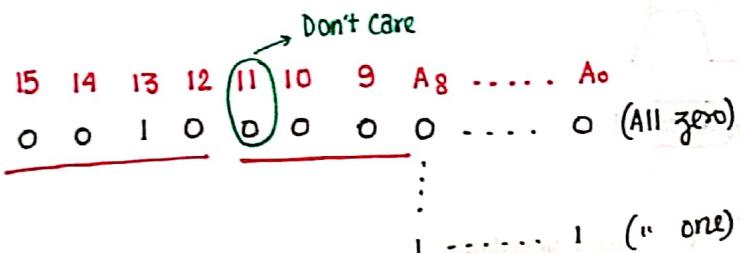
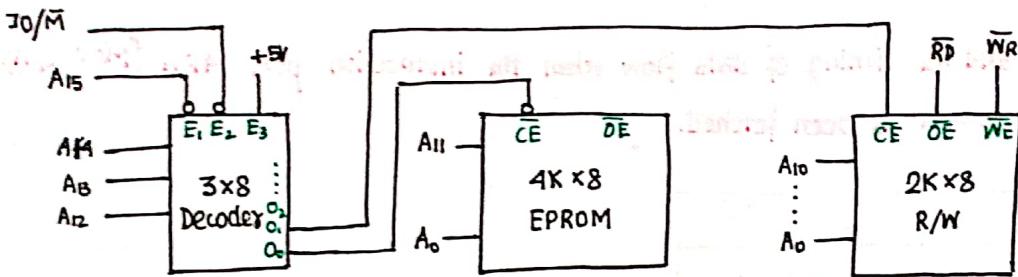
Q Given the components as listed, design an interfacing circuit for memory section of 8085 to meet the foll. specifications

i) 3-8 Decoder (i)

ii) 4x8 EPROM

Address range should begin at 0000H. 2 additional 4K memory space should be available for future expansion.

iii) 2Kx8 CMOS R/W Memory.



2000H — 27FFH

If don't care line = 1 Range: 2800H—2FFFH

### PERIPHERAL MAPPED I/O & MEMORY MAPPED I/O

In this type of I/O the MPU uses 8 address lines to identify an I/O or O/P device. This is known as Peripheral mapped I/O.

In Memory mapped I/O the I/O devices are assigned and identified by 16 bit addresses.

#### Characteristics

1. Device address

#### Memory mapped I/O

16 bits

#### Peripheral mapped I/O

8 bits

2. control signals for I/O

MEMR, MEMW

IN, OUT IOR, IOW

3. Instructions available

STA; LDAX; MOV M,R; ADD M

IN, OUT

4. Data transfer

Between any register & I/O

Only b/w I/O & A

5. Max<sup>m</sup> no. of I/Os possible

64K is shared b/w system  
memory & peripherals

256 I/O devices can  
be connected

6. Execution Time

10/7 T states are reqd.

10 T states reqd.

7. Hardware requirement

More hardware reqd. to decode  
16 bit address

Less hardware reqd. to decode  
8 bit address

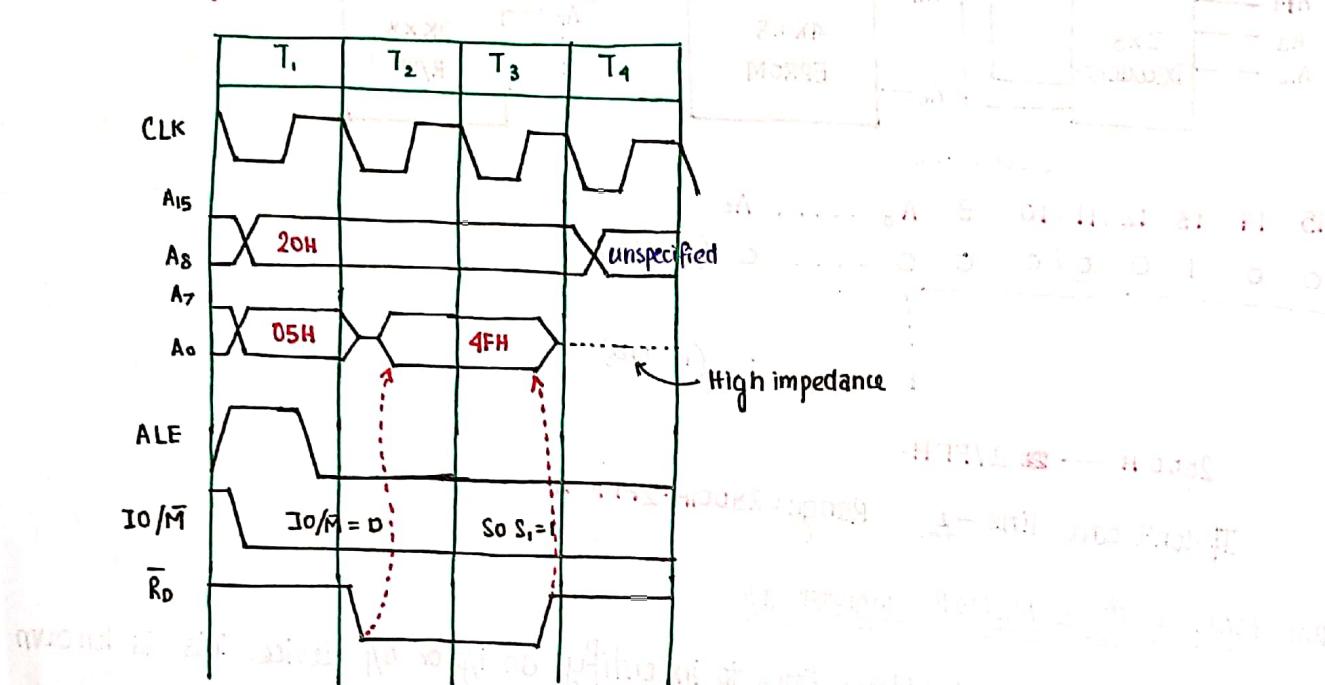
8. Other features

Arithmetic/Logical operations can  
be directly performed with I/O  
data

Not available

## TIMING DIAGRAM

Q) Illustrate the steps and the timing of data flow when the instruction port 4FH (MOV C, A) stored in memory location 2005H is been fetched.



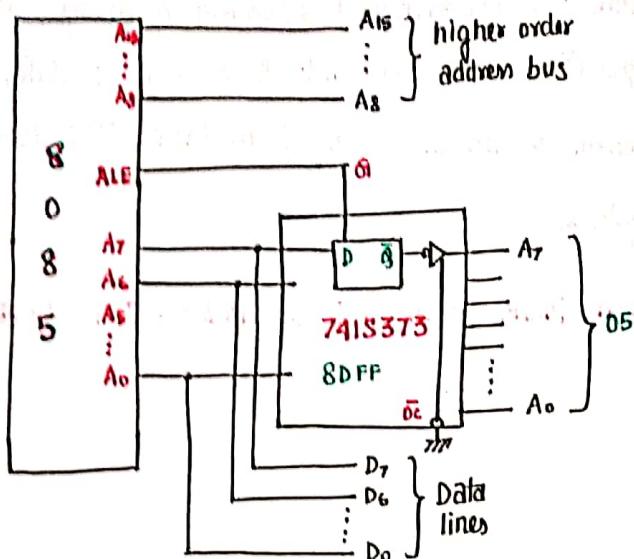
① The processor places 16 bit memory address from the program counter on the address bus. At  $T_1$ , the higher order memory address 20H is placed on the address lines A<sub>15</sub> to A<sub>8</sub> and the lower address bus A<sub>7</sub> to A<sub>0</sub>. The ALE signal also goes high during  $T_1$ . Since it is a memory related operation, the signal IO/M goes low during  $T_1$ .

② The control unit sends the signal RD to the o/p buffers of the memory chips. This signal is sent out during  $T_2$  and remains active for 2 clock periods.

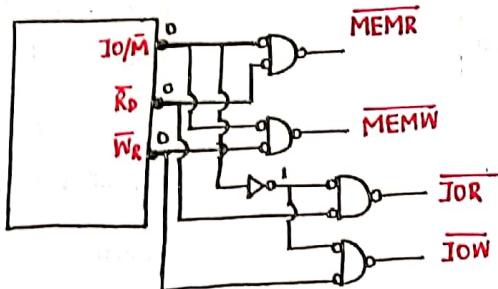
③ The byte 4F from memory location is placed on data bus.

The instruction 4F is placed on the bus A<sub>7</sub> to A<sub>0</sub> & is transferred to the processor. It is decoded by instruction decoder & the contents of the A is placed in register C.

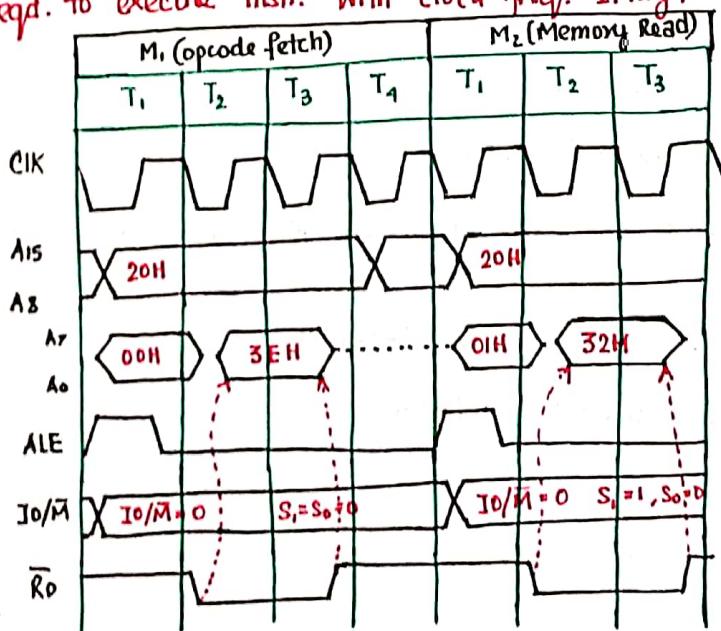
## MULTIPLEXING OF ADDRESS BUS



The Read & Write operations need to be performed both for the memory and the peripherals. In order to distinguish, 4 diff. signals ( $\overline{\text{MEMR}}$ ,  $\overline{\text{MEMW}}$ ,  $\overline{\text{IOR}}$ ,  $\overline{\text{IOW}}$ ) are generated using  $\overline{\text{IO/M}}$ ,  $\overline{\text{RD}}$ ,  $\overline{\text{WR}}$ .



Q. 2 machine codes 3EH and 32H are stored in memory location 2000H & 2001H resp. Op code performs the operation MVI A 32H. Illustrate the bus timings as these machine codes are executed. Calc. time reqd. to execute instr. with clock freq. 2MHz.



$$T = \frac{1}{f} \times 0.5 \mu\text{sec} \Rightarrow \text{Total } T = 7 \times 0.5 = 3.5 \mu\text{sec}$$

During T<sub>4</sub> the 8085 decodes the opcodes & finds out that a 2nd byte needs to be read. After completion of the opcode fetch, the 8085 places the address 2001H on the address bus & increments the program counter to the next address 2002H. The operation is identified as the memory read cycle ( $\overline{\text{IO/M}} = 0$ ,  $S_1 = S_0 = 1$ ;  $\overline{\text{IO/M}} = 0$ ,  $S_1 = 1, S_0 = 0$ ) & ALE is asserted. At T<sub>2</sub>  $\overline{\text{RD}}$  signal becomes active & enables the output buffer of the memory chip. At the rising edge of T<sub>2</sub> memory places data byte 32H on data bus & 8085 reads & stores the byte in A during T<sub>3</sub>.

T state is defined as 1 subdivision of the operation performed in 1 clk period.

Machine cycle is defined as time reqd. to complete one operation of accessing memory, I/O or acknowledging an external request. This cycle operation may consist of 3 to 6 T-states.

Instruction cycle is defined as the time reqd. to complete the execution of an instr. The 8085 " " consist of 1 to 6 Machine cycle(s).

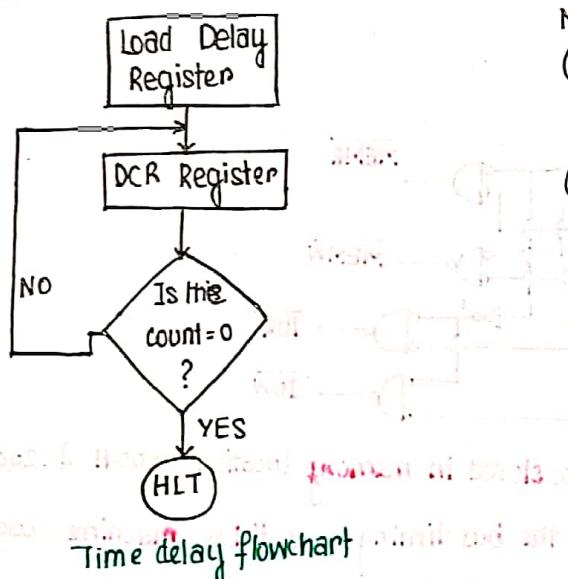
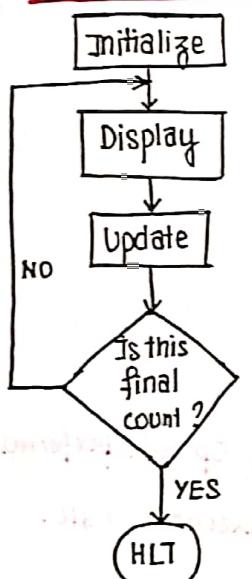
Q Examine the machine cycle of the following 3 byte instr. when it is executed  $\rightarrow$  STA 2065H

This is 3 byte instruction

- STA  $\rightarrow$  32H (4T)
- 65H  $\rightarrow$  RD (3T)
- 20 H  $\rightarrow$  RD (3T)
- Storing in 2065 from A  $\rightarrow$  WR (3T)

Total = 13T

### COUNTERS AND TIME DELAY



MVI B, FFH  
(T-states = 7)

LOOP DCR B  
(T-states = 4)

JNZ LOOP  
(T-states =  $\frac{10}{7}$ )

2 MHz, T = 0.5 μsec

255  $\rightarrow$  since FFH

$16 \times 16 = 256$

$(256 - 1) \rightarrow$  when 0

Time to execute MVI ( $T_1$ ) =  $7 \times 0.5 = 3.5 \mu\text{sec}$

Time to execute loop ( $T_2$ ) =  $0.5 \times 14 \times 255 = 1785 \mu\text{sec}$

The loop is executed 255 times and in the last cycle, JNZ instruction will be executed only in 7 T-states.

Adjusted loop delay is :  $T_D = (1785 - 1.5) \mu\text{sec} = 1783.5 \mu\text{sec}$

$\therefore$  Total delay =  $T_D + T_1 = 1783.5 + 3.5 = 1787 \mu\text{sec}$