

Types

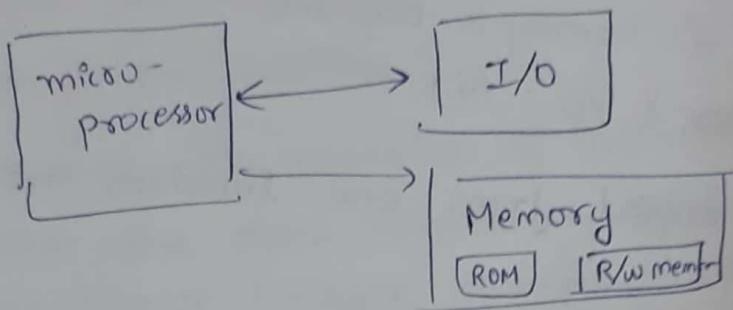
① Reprogrammable

user directly can use the m.p. and give instructions.

② Embedded System

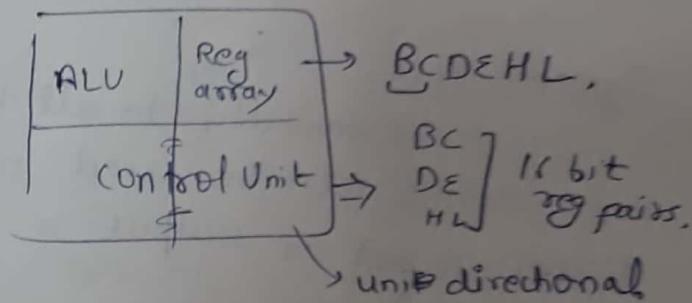
part of a bigger system ex - washing machine car dash board

Instruction set is converted to binary data by MP.

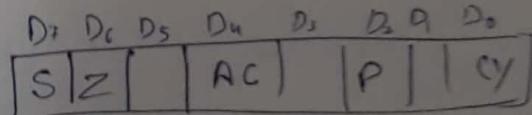
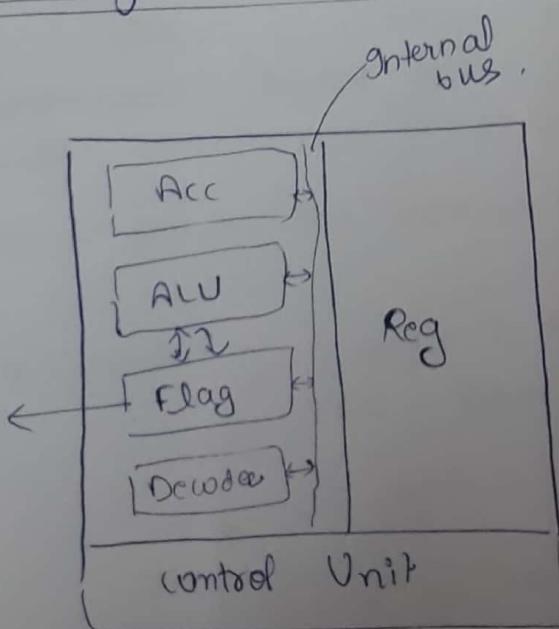


2 types of program counter, sequence counter.

Microprocessor →



→ Function of control unit



✓ S → Sign

✓ Z → Zero

AC → Auxiliary carry

P → Parity

✓ CY → Carry

g → JC (Jump on carry)

① Jump

Conditional jumps are important in a decision making process in programming.

This instruction test for certain conditions i.e. 0, carry, flag etc to alter the program sequence. In addition, we also have an unconditional jump.

② Call, return & Restart

These instructions change the sequence of the program either by calling the subroutine or returning from subroutine, these conditionally call, return can also ~~test~~ test the condition flags.

Instructions

Three types

① 1 byte instruction

For these instruction, operand and opode require one memory location.

Opode	Operand	Hexcode
MOV	C, A	4FH
ADD	B	80H
CMA		2FH

② 2 byte instruction

These instructions require 2 memory location each to store the binary code for the opode and the operand.

Opode	Operand	Hex
MVI	A, 32H	3E 32H
MVI	B, F2H	06 F2

③ 3 byte instruction

They require 3 memory locations to store the binary code. The first byte stores the binary code for the opode, and the second and third byte specifies 16 bit address.

(Instruction Set) X

Class-8, 4, 5
15/1/2018

Arithmetic Operations.

Addition and Subtraction,

- 8 bit data
 - contents of memory location,
 - contents of a register
-]} ± Accumulator.

① DAD

Adding the contents of a register directly.
(16 bit data in a reg).

Logical Operations

① AND, OR, XOR

- memory location
 - reg
 - 8 bit data
- } with Accumulator

② Rotate

Each bit of the ac is shifted either left or right to the next position.

③ Compare

→ any 8 bit number or the contents of a register or a memory location can be compared with the accumulator. Contents for equality greater than or less than

④ Complement

CMA
→ the contents of the ac can be complemented i.e. all 0's replaced by 1 and vice-versa.

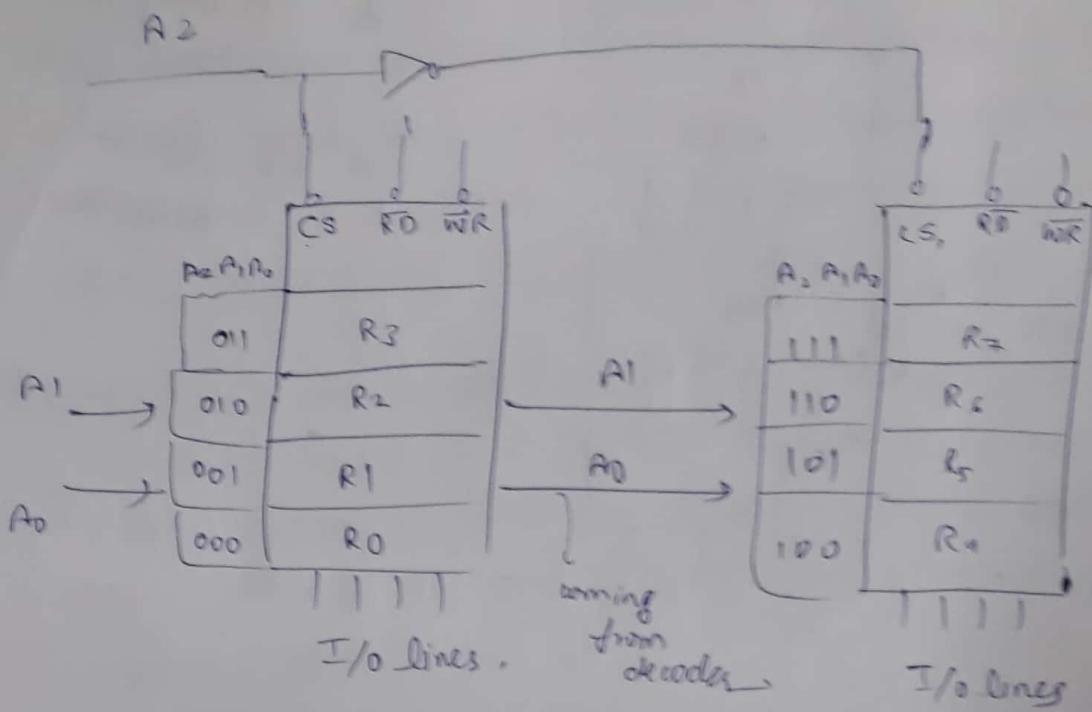
Branching operations.

can either be conditional or unconditional.

Two types

①

- ② The memory chip requires, \bar{CS} to enable the chip and address lines of the memory chip called register. Thus the memory address of a register is determined by the logic levels 0/1. of all the address lines.
- ③ The control signal \overline{RD} enables the output buffer and the data is selected from a register and is made available at the output.
- ④ \overline{WR} enables the input buffer and the data on the input lines is written into the memory cell.



- ① When A_2 is active low, the chip M_1 is selected, when A_2 is active high, chip M_2 is selected.
- ② Address line A_1 and A_0 determines individual address of registers.
- ③ A_2, A_1 and A_0 determines complete address from 000 to 111.
- ④ \bar{CS} gives more flexibility in designing the chip and expanding the memory size.

Conclusion

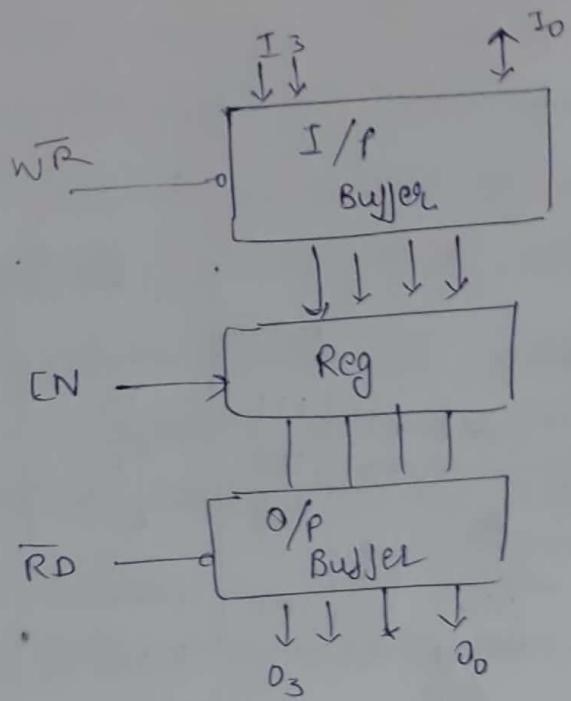
- ① A memory ^{chip} requires address lines to identify a memory register.

8085 - 16 address lines.

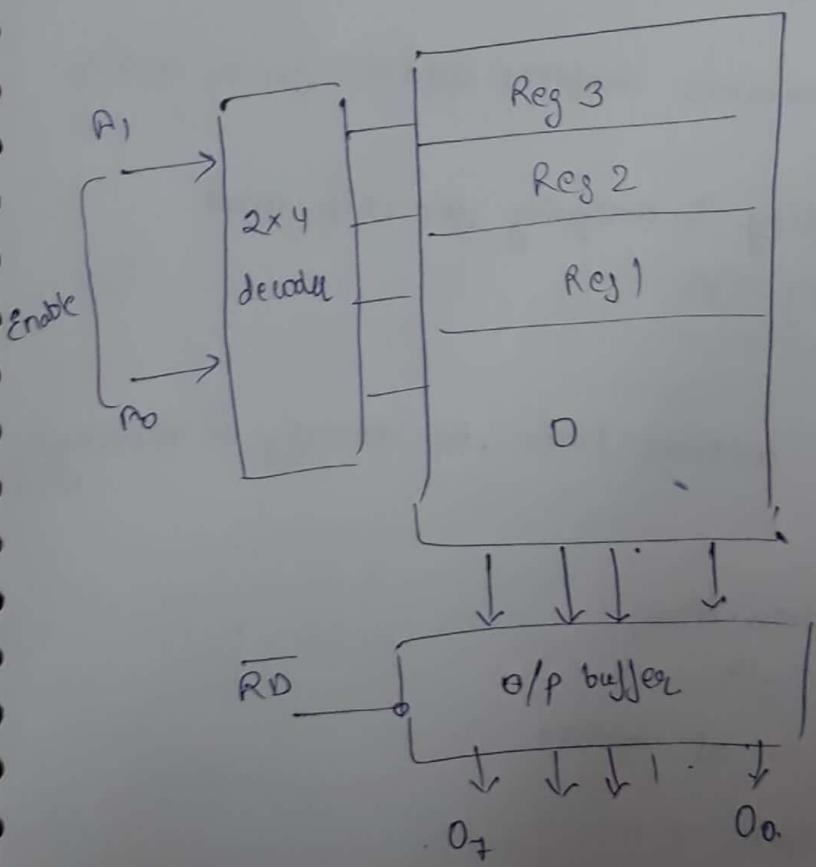
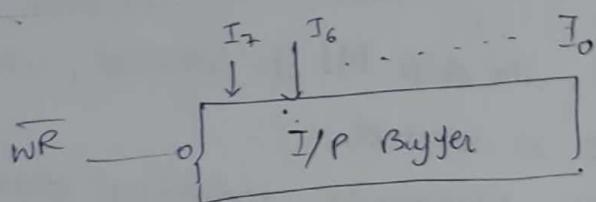
No. of registers - 2^{16}

$n \rightarrow$ No. of registers

Address lines $\rightarrow n$,



- * 1 Binary Bit Stored.
- * $\overline{RD}/\overline{WR}$ active low



Peripherals

RESET :- When activated by external devices, all the internal operations are suspended and program counter is cleared to 0000 H, so that the program execution can begin from memory address ~~not be~~ 0000 H.

INTERRUPT :- The microprocessor can interrupt from its normal routine execution and can be asked to execute some other instruction called the service routine and MPU resumes its normal operations after completing service routine.

READY :- After ^(active low pin) goes low, MPU enters into wait state and this is used to synchronize with slower peripherals.

HOLD :- When activated, MPU relinquishes the control of the buses and allow some external peripheral to use them.

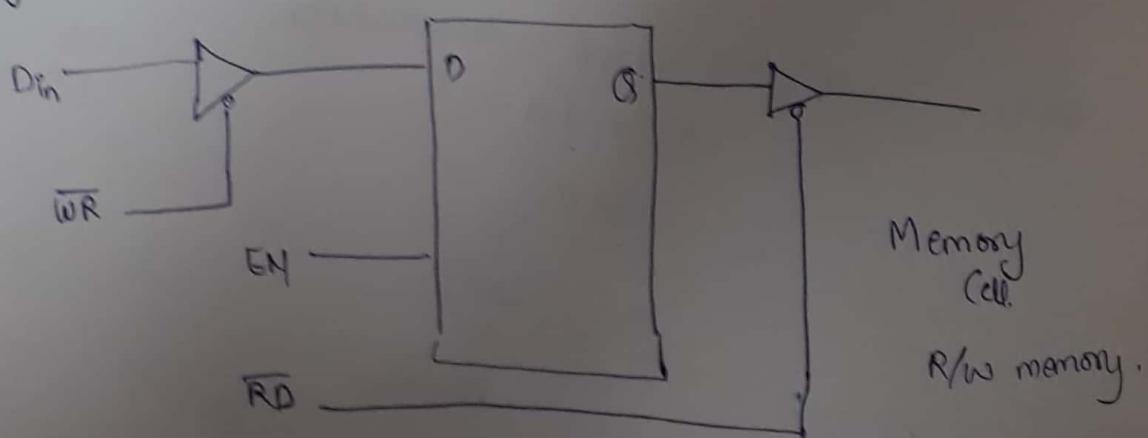
DMA data transfer

Direct Memory access.

RAM :- flip-flop.

ROM :- Diode (forward or reverse biased)
contents can't be changed

Memory Cell



Chapter-1 is done.

(Ch-2 (instructions))

Goonkar 8085
6th edition

class 2,3

8/11/2018

Internal architecture decides the operation to be performed.

- ① stores the 8-bit data.
- ② perform arithmetic and the logical operation
- ③ test for the condition.
- ④ sequence the execution of instruction.
- ⑤ store the data temporarily during execution in the read/write memory location called stack.

PC → 2000	opcode 06	MVI(B) 7,8H	(MVI → More immediate)
1600 byte instruction	78	register	
2001	3E		
2002	F2		
2003	80	MVI(A) F2H	
2004		ADD B	
2005	76	HLT	

Address 2000 is loaded to PC, transferred to address bus, so the microprocessor can go to that address.

2000 is put on the address bus, PC increments.

Add B:- content of B added to Accumulator.

78H + F2H

= 16AH

= (1) 6AH
carry Register

78
F2

=

Addition operation can not be performed directly, but use AC.
[DAD instruction] allows content of register directly

Diffr b/w Program counter, and stack pointer.

Tines

→ Stack pointer is present in R/W memory

→ PC is present in register

Chapter 1

Ch 2 (1)

internal
operat

① stores

② perform

③ test

④ sequ

⑤ sto

re

Four types of Data Transfer

- ① Register
- ② Memory & Register
- ③ Accumulator & I/O
- ④ - ?

Data is always copied from one location to another.

Acc is allocated to register 'A'.

System Bus

↳ connected to all the peripherals.

I/O device can be used only one at a time.

Either input or output is used to interact with peripheral.

PC —

two
byte
register

Address
so th

2000

Add B :-

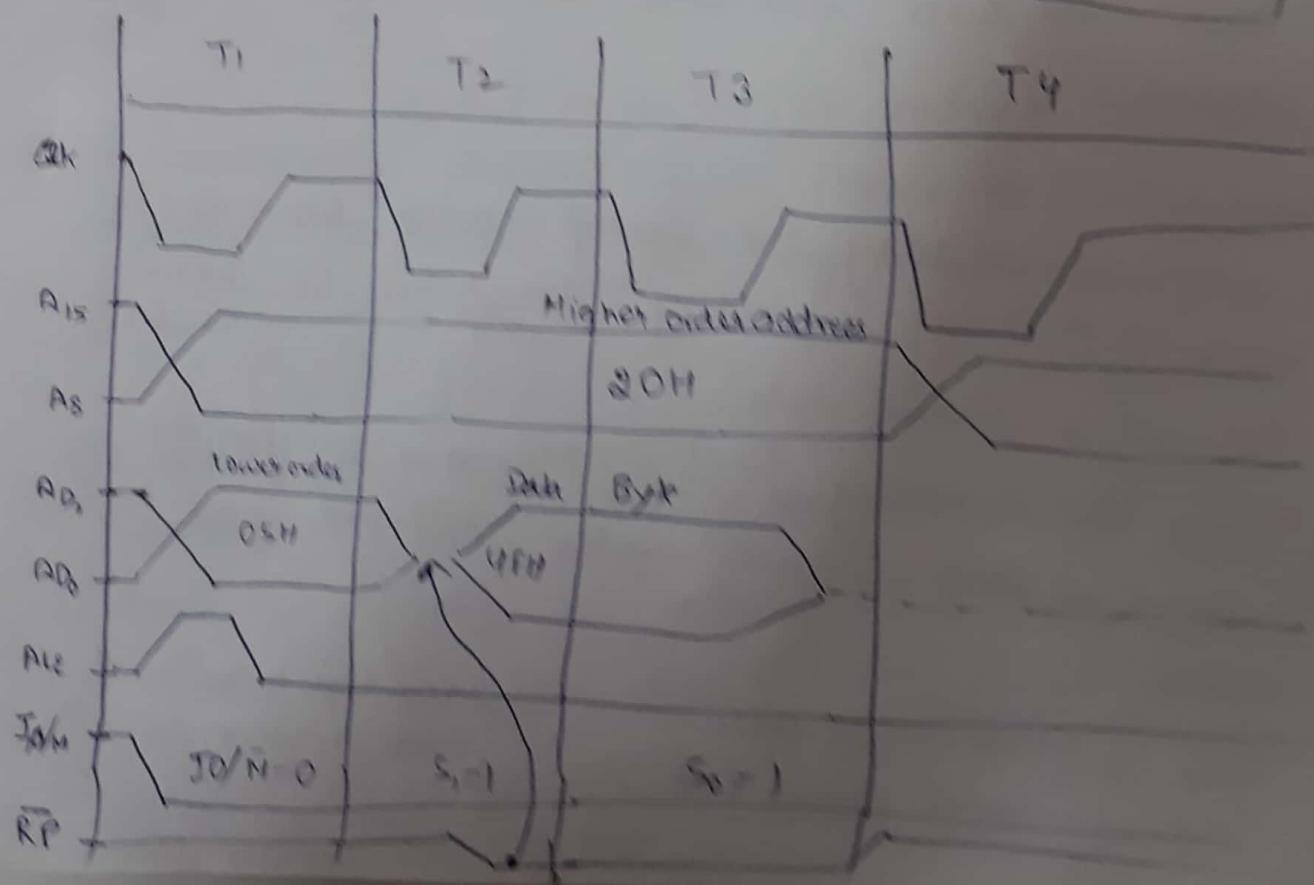
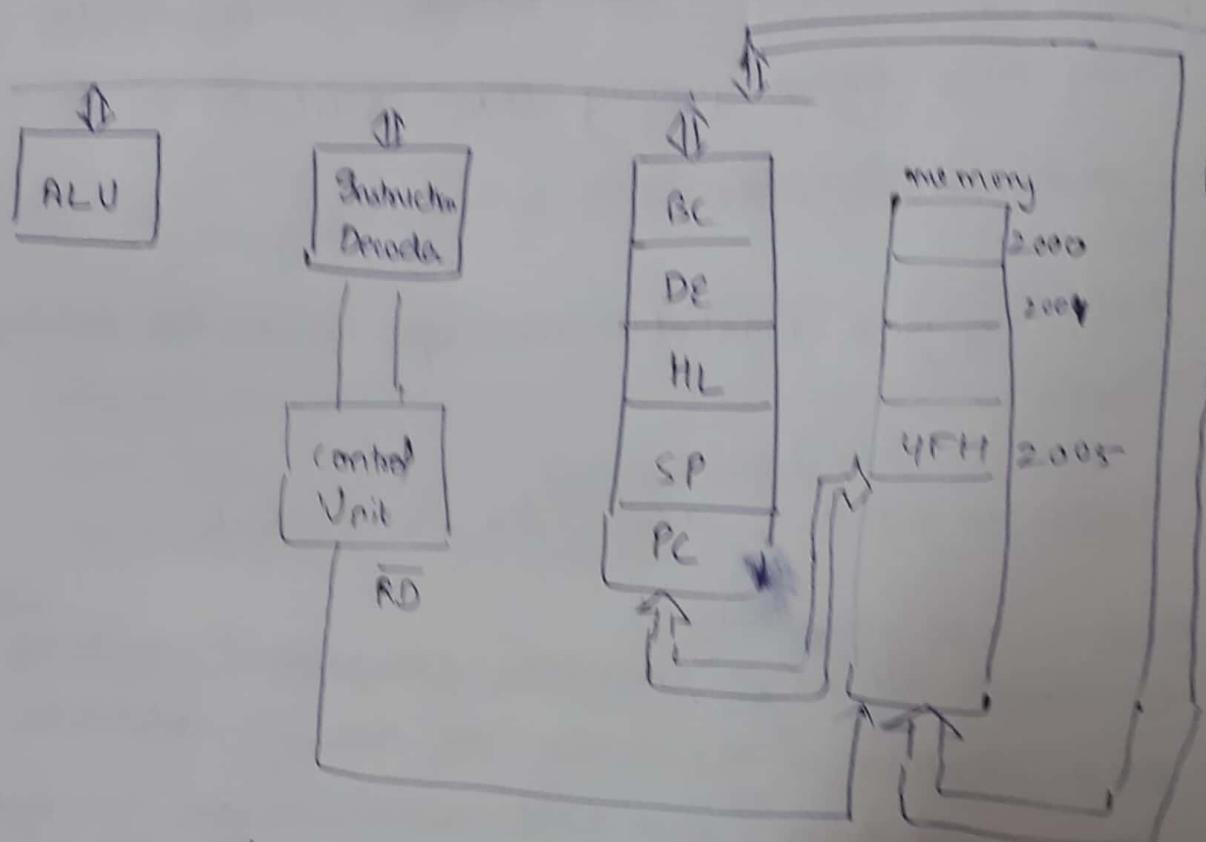
Steps to fetch data from the location 2005H.

Binary code = 4FH

Instruction: MOV C, A.

class-9
July 16
(Not attended)

How this happen in the system?



When interrupt occurs, interrupt acknowledge done.
INTR and $\overline{\text{INTA}}$ coordinates with each other

TRAP are unmaskable interrupt

RST Restart Interrupt

- these are vectored interrupts that transfer the program counter to the specific memory location
- they have higher priority than INTR
- order is: 7.5, 6.5, 5.5

TRAP is non-maskable interrupt and has the highest priority

INTR is input, and $\overline{\text{INTA}}$ is output.

HLD (Hold acknowledge)

The signal indicates that the peripheral such as DMA controller is requesting the use of address and Data Buses, and hold acknowledges the hold signal.

Ready

- 1) It is used to delay the microprocessor, RD or write cycles till a slow responding peripheral is ready to send or receive data
- 2) When the signal goes low, MP waits for an integral number of clock cycles until it goes high again

SID → Serial Input Data

SOD → Serial Output Data

Data has to be sent and retrieved serially.

Reset → PC goes to initial state
Register class

ALE Address latch enable.

This is a positive going pulse generated every time a 8085 begins the operation. It indicates that A7 and A0 are address bits. This is used to latch the lower order address from the multiplex bus and generate a separate set of 8 bit address lines A7 to A0.

I/O M, is data fetched from output peripherals or from memory.

This is used to differentiate between ~~I/OAT~~ I/O and Memory operation. High indicates I/O operation and low indicates memory operation and is combined with RD and WR

B/W

X₁ and X₀, there is a crystal oscillator or RC/CC network

X₁, X₀

It is a crystal or an RC/CC network. The frequency is internally divided by 2. Hence to operate the system at 3 MHz. The crystal should have a frequency of 6 MHz.

Clk out is a system clock

Externally related signals → Interrupts has certain priority.

→ There is a priority of interrupts.

There are 5 interrupts

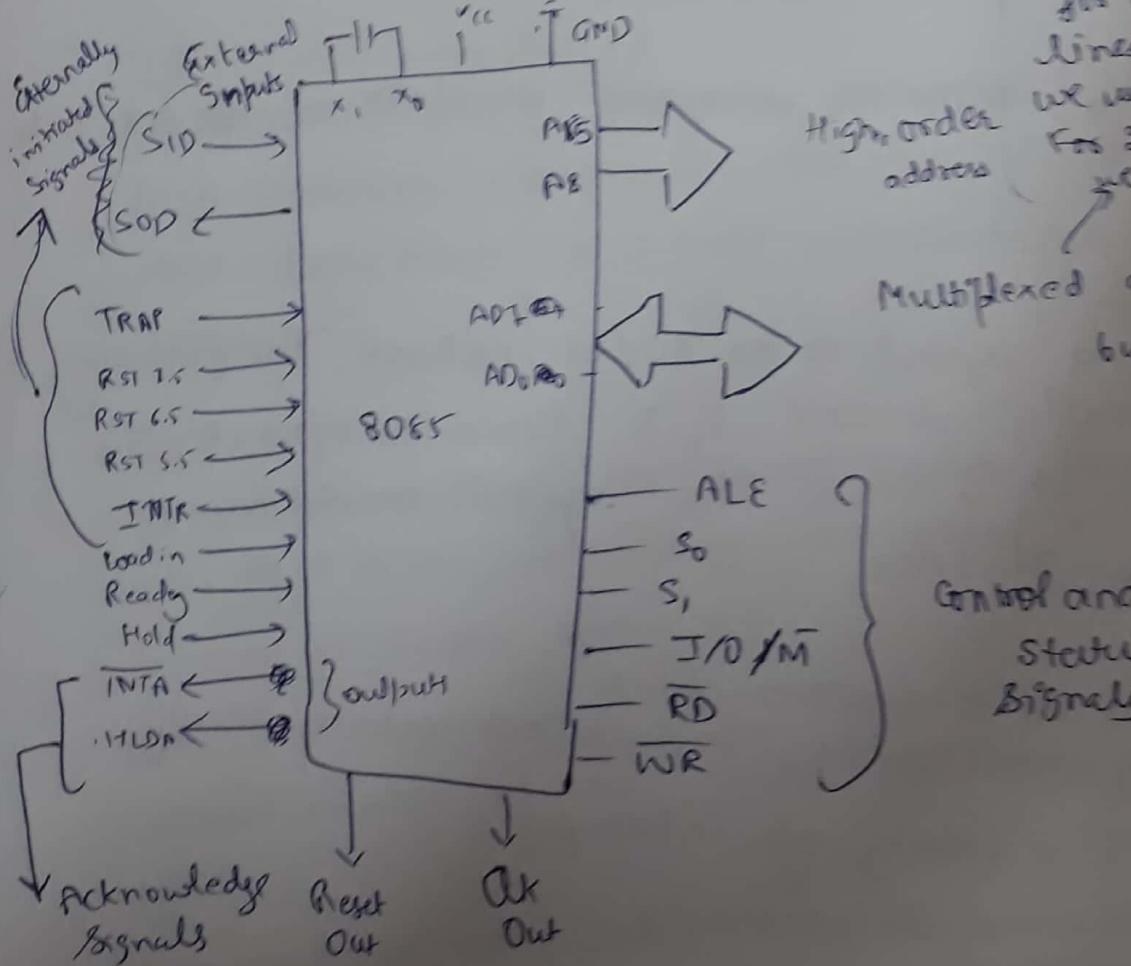
TRAP is highest priority

INTF is lowest priority

Basic Information about 8085

- 1) It is 8 bit microprocessor (Data stored is 16 bit)
- 2) It has 16 address lines and hence capable of addressing 64 K of memory.
- 3) It has 40 pins.
- 4) Power Supply is +5V.
- 5) It operates with a 3MHz single phase clock.
- 6) All signals can be classified into address bus, data bus
 → Control and status signals.
 → Externally initiated signals.

7) Serial I/O data.



Using same set of lines for address or data lines for 1st half cycle, we use it as address line. For 2nd half cycle, we use it as data line.

Multiplexed add/data bus

Control and Status Signals

Addition Subtraction
 $20534 \rightarrow 4874$ $20534 \rightarrow 4874$ Add and store the sum by
 20534 20534 20534

stacking R $\rightarrow 2030H$
 point

Ans

4874	2030	LDA	2051H
1120	2031		
3498	2032		
1000	2033		

2033	MOV R,A	
2034	LDA 2050H	
2035		
2036		
2037	ADD B	
2038	STA 2053H	
2039		
2040	HLT	
2041	HLT	

Pin Configuration

8085 is 40 pin IC

Q13 is
done

n address lines,
 8 n registers

⑥ CMP R

⑦ CPI 8 bit

CMP B

CPI 4FH

Branch Instructions

① JMP

16 bit add

JMP 3050H

② JZ

16 bit add

JZ 2080H

③ JNZ

16 bit add

JNZ 2030H

④ JC

16 bit add

JNC 3070H

⑤ JNC

"

CALL 3075H

⑥ CALL

→ call for a service routine

RET

⑦ RET

→ Return

Machine Control Instructions

① HLT → Halt

② NOP → No operation.

Length of each instruction - ? 1 byte, 2 byte.

Find it by yourself

All instructions are at the back of the book
(Appendix)

Arithmetic Operations.

Mnemonics

- ① ADD R
- ② ADD 8-bit
- ③ ADD M
- ④ SBSRR, SUB 8bit, SUB M

- ⑤ INR R
- ⑥ INR M
- ⑦ DCR R
- DCR M

- ⑧ INX Rp

- ⑨ INX R_p

E.g.

- ADD B
- ADD 3FH
- ADD M

- INR D

- INR M - HL

- INX D

Congruence
Immediate I

- DCX B,

Logical Operations

AND, OR, XOR.

- ① ANA R
 - ② ANI 8 bit
 - ③ ANA M
 - ④ F0R OR
 - ⑤ F0R XOR
- ANA B
 - ANI 3FH
 - ANA M
 - OR A, OR I, ORA M
 - XOR A, XRI, XRM

⑥

⑦

Branch

①

②

③

④

⑤

⑥

⑦

Mach

①

②

⑨ MOV R,M

MOV R,M

⑩ MOV M,R

MOV M,R

③ → LXI

Load the 16 bit data in the register pair
B, 3030H → 50H in C
30H in B.

Load Immediate

④ IN

→ Microcontroller accepts the data from the input device
and places it in the accumulator.
02H → stored in A.

⑤ ST copies the data into A from the memory location
specified by the 16 bit address.

⑥ STA is reverse of LD.

→ pushing the data from Accumulator to 16 bit
address

⑦ LDAX

Load accumulator direct

This copies the data byte into A from the memory
location specified in the register pair.

⑧ STAX

Data from accumulator is shift to Register Pair

⑨ Copies the data byte into register from the memory
location specified by HL Register Pair

⑩ →

Second byte stores the lower order of the address, third byte stores the higher order of the address

Opcode
LDA

Operand
2050H

Hx
3A
50H
20H
opcode
lower
higher order

JMP

2055H

C3
85H
20H

Instruction Set.

8085

R - 8 bit Register

M - Memory Location

Rs - Source Register

Rd - Destination Register

R_p - Register Pair

Data Transfer Operations

Mnemonics

① MVI @ R, 8 bit

② MOV @ Rd, Rs

③ LXI R_p, 16 Bit

④ IN 8 bit

⑤ LDA 16 Bit

⑥ STA 16 Bit

⑦ LDAX R_p

⑧ STAX R_p

Eg.

MVI B, 4FH

MOV B, A

LXI B, 2050H

IN 0FH,

LDA , 2050H

STA 2070H

LDAX , B

~~STAX~~, D

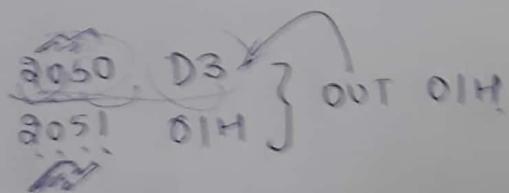
- ① The don't care address lines can be assumed to have, any one of the 8 input combinations, 000 to 111. Thus each combination can generate one set of complete address.
- ③ Address range given by ^{addr}, assuming top don't care 000 is by convention. This is specified as the primary address. The remaining address range is known as the foldback memory or the mirror memory.

2000H - 20FFH] Primary 2100H - 27FFH] Mirror memory

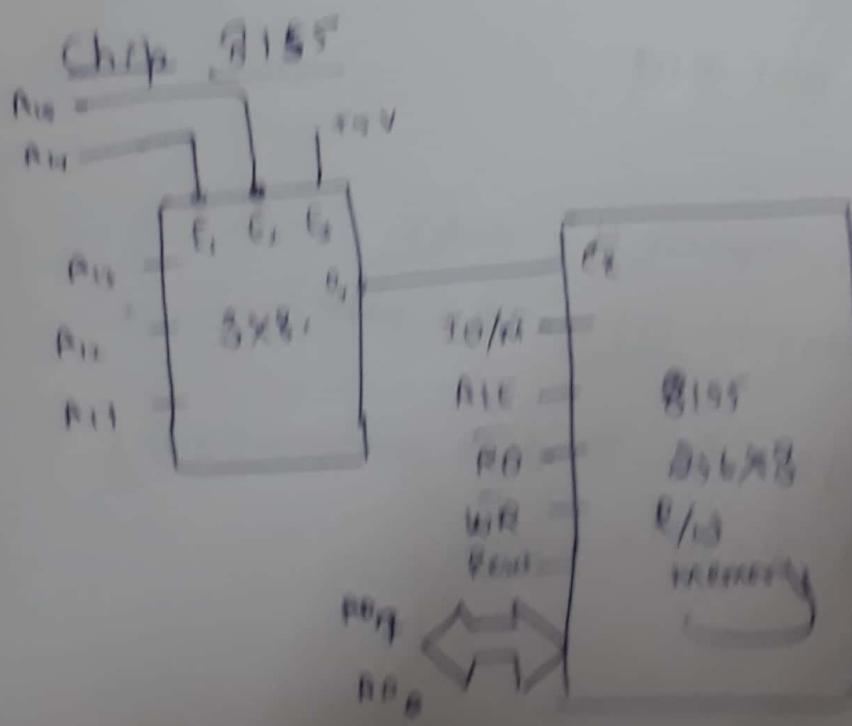
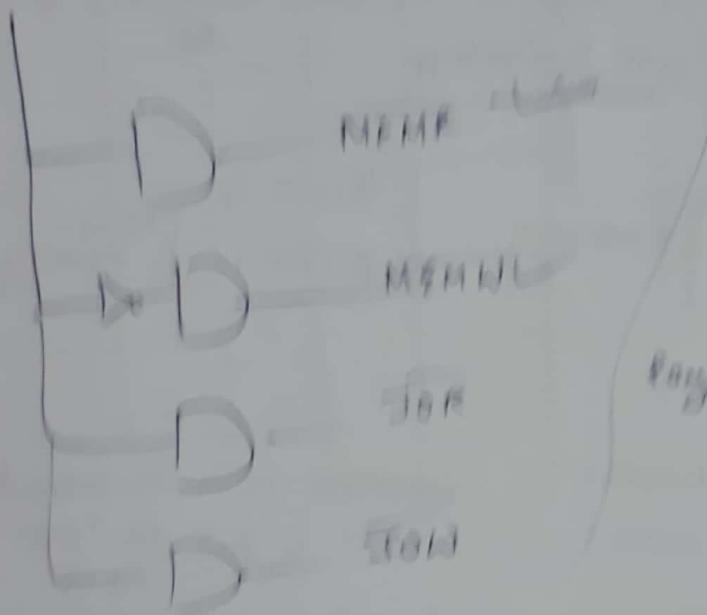
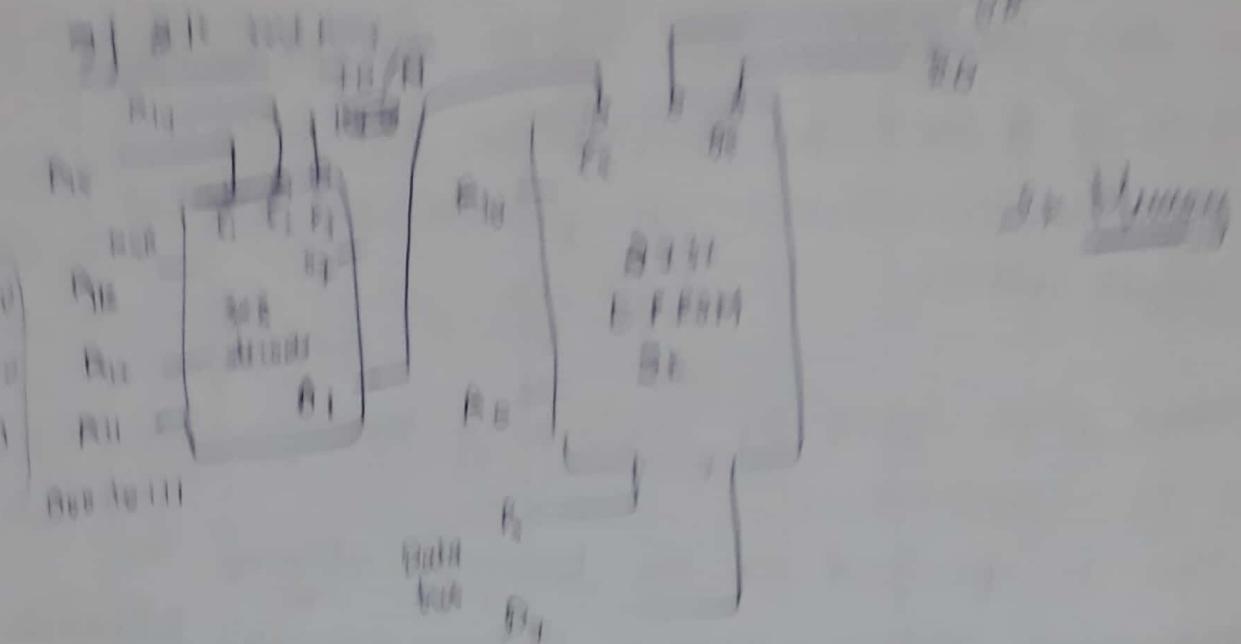
→ End of 4th chapter

⇒ OUT 8 bit address
→ (10 T states)

whatever is in accumulate
is put to port address
01H

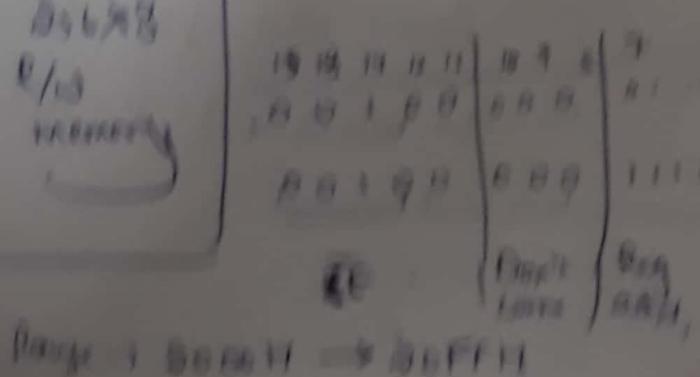


- ① Decode D3, requires (1T state)
- ② Get the address 01H (1T state)
- ③ Load data from AC to this address (01H)
It requires 3 machine cycles.



$R15, R14, R13 \rightarrow R10/R9$
fast
 $0\ 0\ 0 \rightarrow 0\ 0\ 0$
long

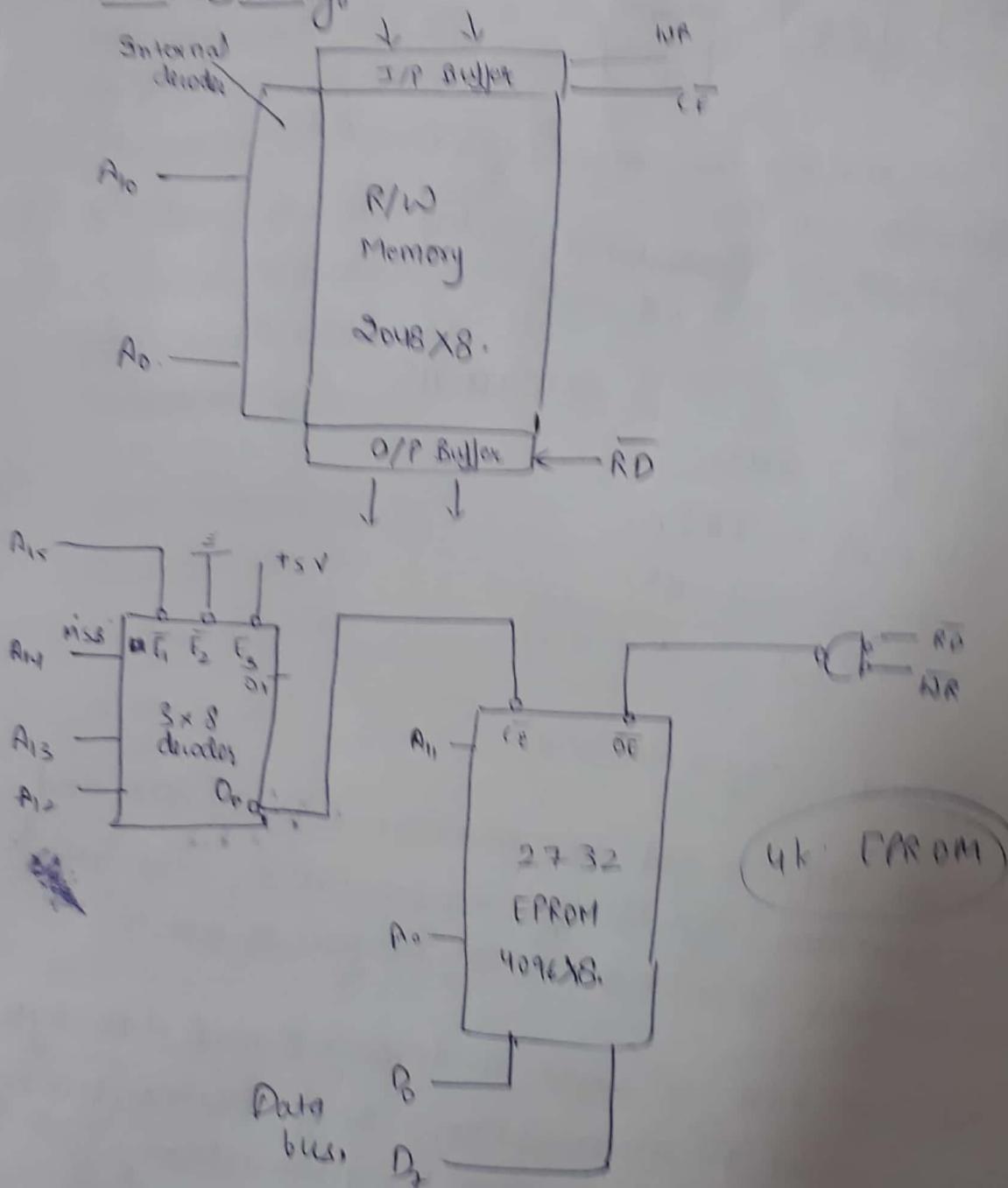
$f = add R10, R9$



$4x \rightarrow 12$ Address lines

$2x \rightarrow 11$

Address Decoding



Chip Enable

A_{15}	A_{14}	A_3	A_{12}
----------	----------	-------	----------

0000H	0	0	0
-------	---	---	---

0FFFH	0	0	0
-------	---	---	---

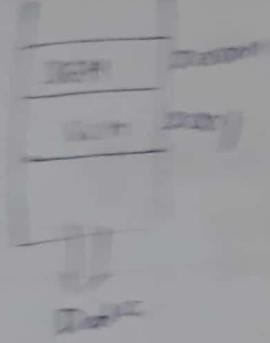
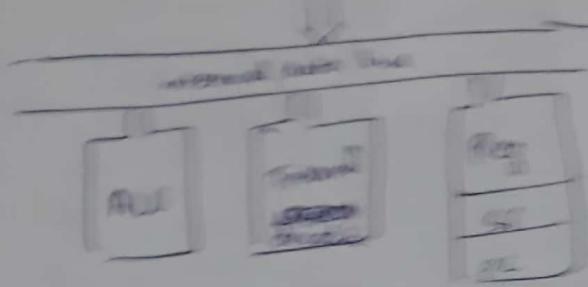
Memory range

Register Address

$A_{11} \dots$	$\dots A_0$
----------------	-------------

0	\dots	\dots	0
---	---------	---------	---

1	1	1	1
---	---	---	---



we do not perform any decoding operation in MZ
Therefore MZ is not needed
To subtract

$$T = \frac{1}{f} = \frac{1}{200\text{ns}} = 0.5 \mu\text{s}$$

$$ML = 0 \times 0.5 = 0 \mu\text{s}$$

$$MZ = 3 \times 0.5 = 1.5 \mu\text{s}$$

$$\text{Total} = 2.5 \mu\text{s}$$

Assignment-1 (After First Solved)

Q.1. Write the code to add two numbers 0001 0101 and 0111, save the sum to the accumulator, also both the numbers should be saved for future use.

Q.2) Data byte 0011 is stored in Register B and data byte 0111 is stored in accumulator. Show the contents of the registers B and C after the instruction,

Now R, B

Now C, A

have been executed.

Q). 2 machine codes,
3EH and 32H are stored in
memory locations 200H & 2001H.

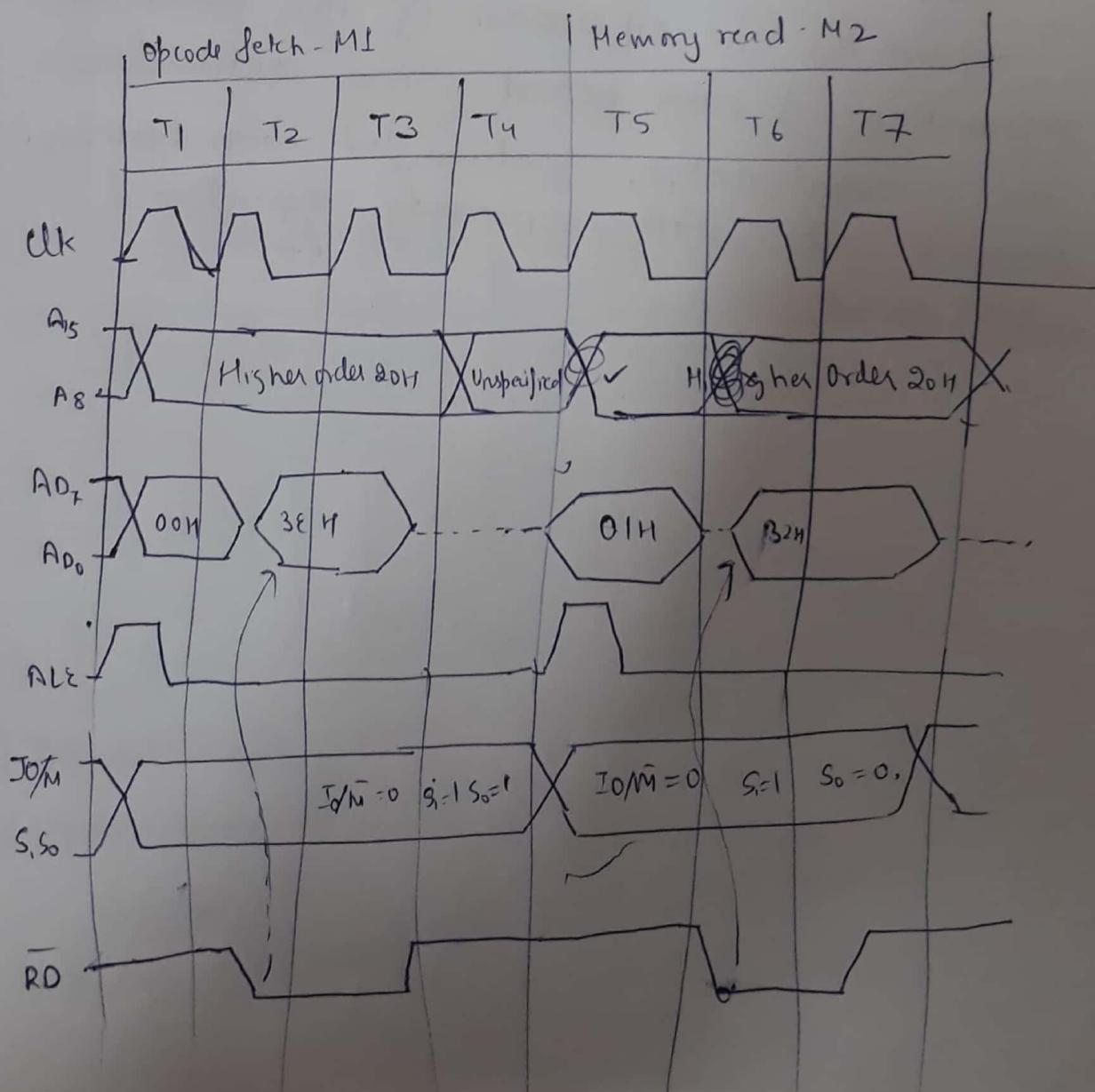
3EH represents the opcode to locate the data byte
in the accumulator and 32H is the byte to
be loaded in the accumulator.

- ① Illustrate the bus timing.
- ② Calculate the time required to execute the opcode
fetch and the memory read cycles and the entire
instruction cycle if the clock frequency is 2 MHz.

Soln

3EH → 2000H

32H → 2001H



* Every machine cycle require no. of states
(T_1, T_2, \dots, T_7)

① It is defined as time required to complete the execution of an instruction 8085 instruction cycle contain 1-6 Machine cycle (1-6 operations)

② Machine cycle

Defined as time required to complete an operation of accessing the memory, IO or acknowledge any external request. Each Machine cycle consists of (3-6 states).

③ T-state

defined as one subdivision of the operation performed in 1 clock period. These subdivisions are internal states synchronized with system clock & each T-state is precisely equal to 1 clock period.

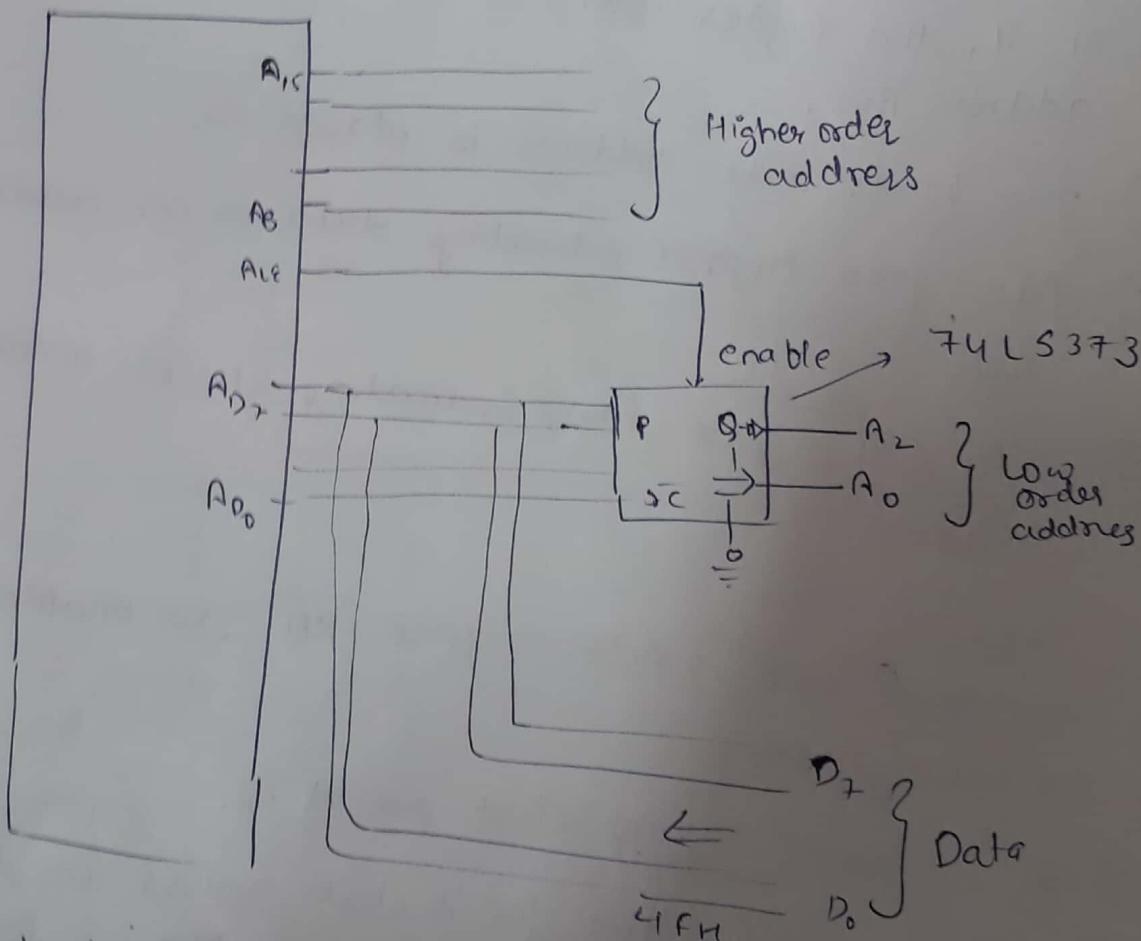
bus.

When RD goes high, it causes the bus to go in the high impedance state.

Step 4 : The byte is placed in instruction decoder and task is carried out according to the instructions

The byte 4FH is decoded and contents of AC are output into register C. ^(By decoded)

From 3 to 6 clock periods, all instructions executed,



- ① Instruction cycle
- ② Machine cycle
- ③ T states.

(set of operations performed (1-6))

~~8~~
Borrower: Address Bus is waiting for address to be loaded onto it.

---: High impedance state, waiting for next byte to come.

ALE is high: to Enable.

Steps:

Step 1: MP place 16 bit address from PC on to address Bus.

- (a) At T1, the higher order address is placed on address Bus.
- (b) The lower order address is placed on.
- (c) ALE goes high (indicating data is on address bus)
- (d) I/O / \bar{M} goes low/ indicating it is a memory location.

Step 2:

The control unit sends signal \bar{RD} to enable the memory chip.

- (a) \bar{RD} is send during clock period T2

- (b) \bar{RD} remains active for 2 clock cycles i.e. to read the data from memory location & load it onto data bus.

Step 3:

The byte from memory location is placed on data

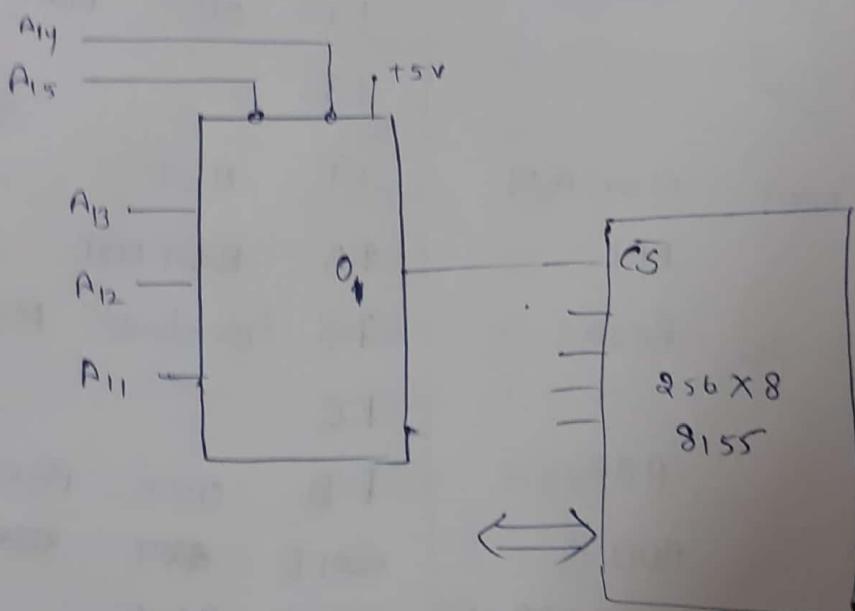
Assignment - 2

Q-3 LDA \rightarrow 2050 H copies the content to the accumulator. Identify the no. of machine cycles, and the no. of T states required, and draw the timing diagram.

Q-4 Identify the no. of machine cycles and T states for the following.

- ① SUB B
 - ② ADI 47H
 - ③ STA 2050 H
 - ④ PUSH B,
- (No. timing diagram
for this que)

Q-5



- ① Specify the memory address range if \bar{O}_0 of the decoder is connected to \bar{CS} . Specify the hold back memory.
- ② Specify the memory range if O_7 is connected to chip enable bar signal of the 2K memory.
- ③ Specify the relationship b/w storage of the feedback memory and the don't care lines.

- 13,14
- ④ out FFH to port 1 at any time when the sum exceeds 8 bits to indicate an overload, otherwise displayed the sum.
 - ⑤ if the output code is not available in the system, then go to step ⑥
 - ⑥ Store FFH in the memory location XX70, when sum exceeds 8 bits, otherwise stored the sum.

			, Reset (content)
Step 1	00	MVI B, 00H	13 JNZ NEXT
	01		14
	02	MVI C, 0AH	15
	03		16 MOV A,B
	04	LXI D, XX60	17 OUT PORT 1
	05	'	18
	06		19 HLT
	07	NXT MOV A,M	\$A REJECT NOP
	08	RAL	\$B (overload) MVI A, FFH
3 byte instruction (leave addressing) W~	09	JC REJECT	\$C
	0A		\$D OUT PORT 1
	0B	RAR	\$E - REJECT PORT
	0C	ADD B,	\$F HLT.
	0D	JC OVERLOAD>FFH	20
	0E		21
	0F		
	10	MOV B,A	
	11	(Reject) INX H	
	12	DCR C	

Q) 16 bytes of data are stored in the memory location 5/2/2019
Class 13, 14
XX50 to XX5FH.

Transfer the entire block of data to a new memory
location XX70H

Ans:

Label,

00

LXI H, XX50H

01

02

03

LXI D, XX70H

04

05

06

MVI B, 10H. → 16 byte counter.

08

MOV A, M

09

STAX D

0A

INX H

0B

INX D

0C

DCR B

0D

JNZ NEXT

0E

0F

10

HLT,

Q) A set of 10 readings is stored in a memory location,
starting at XX60H, readings are expected to be
positive. Write a program,

- ① To check each reading to determine whether it is positive or negative
- ② Reject all negative readings.
- ③ Add all the positive readings.

UNCONDITIONAL JUMP

Q) Load the two hex numbers 9BH & A7H in reg D & E respectively, Add the numbers.

① If sum > FFH

displays 01H at the output

else display the sum.

Not
have
if
Statement

Soln

00	MVI	D, 9BH,
01		
02	MVI	E, A7H
03		
04	MOV	A, D,
05	ADD	E
06	JNC	DISPLAY
07		
08		
09	MVI	A, 01H
0A		
Display	0B	OUT
0C		01H
0D	HLT	

1000 0000

Write a program,

00	- MVI	A, Data
01		9FH
02	- ANI	80H
03		
04	- OUT	01 H
05		
06	- HALT	

Q-2) B → 93H

R → 15H

ORAB, XRAB and CMA

These are three different programs.

Q-3) ^{previous} In the figure, keep the radio on continuously, without affecting the function of other appliances, even if someone turns off Scotch Sy.

Ans - DR with 10 H. 0001 0000

Q-4) In the winters turn off the AC., without affecting other appliances.

Ans, AND with 7FH

$B \rightarrow 30H$

$C \rightarrow 59H$

Sum $0011\ 1001$

$S^* 1100\ 0110$

Result $1100\ 0111$

$30H\ 0011\ 0000$

$\textcircled{D}\ 0111\ 0111$

$C_7=0, S=1, Z=0.$

0 0

0 1

0 2

0 3

0 4

0 5

0 6

0 7

0 8

MVI

~~MOV~~

MVI

~~MOV~~

MOV

SUB

OUT

HLT

16, 30H

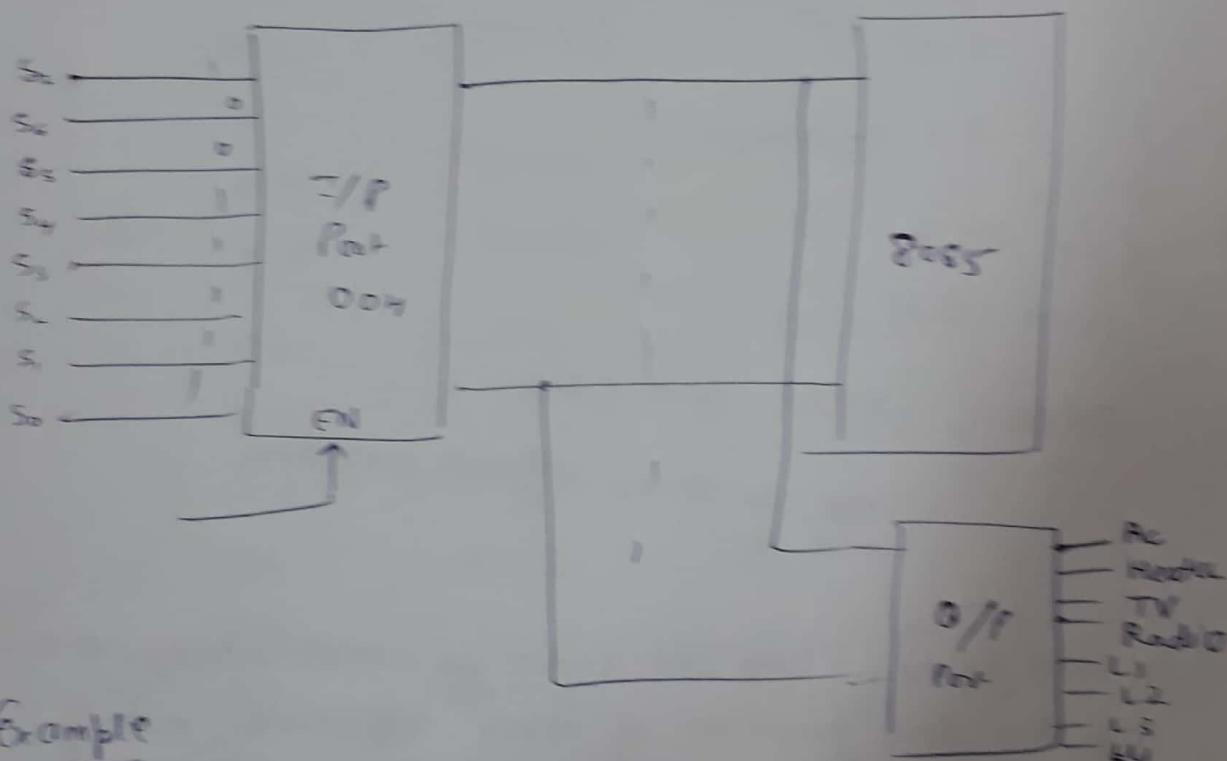
C, 59H

A,B

C

PORT A

Logical Operation



Example

To conserve the energy in the house,

(a) Turn on the air conditioning.
(Switch S7 of input port)

(b) Ignore all the other switches of the input port even if someone attempts to turn them on to use any other appliance.

31/1/18
Class 12

(b) Write the program and verify the output.

Load,

- ① D → 8BH
- ② C → 6FH

Increment the content of register C with 1 and add the contents of C & D and display the sum at the output port 1.

③
 1000 1011
 0111 0000

 0 D111 1011
 9 Da
 G=1 S=1 Z=0

xx00	MVI	D, 8BH
01		
02	MVI	C, 6FH
03		
04	INR C	
05	MOV/A,C	
06	ADD D	
07	OUT B	
08		
09	HLT	

Since the addition of two signed numbers is not taking place. Bit D7 = 1 resulted in the sign flag can be ignored.

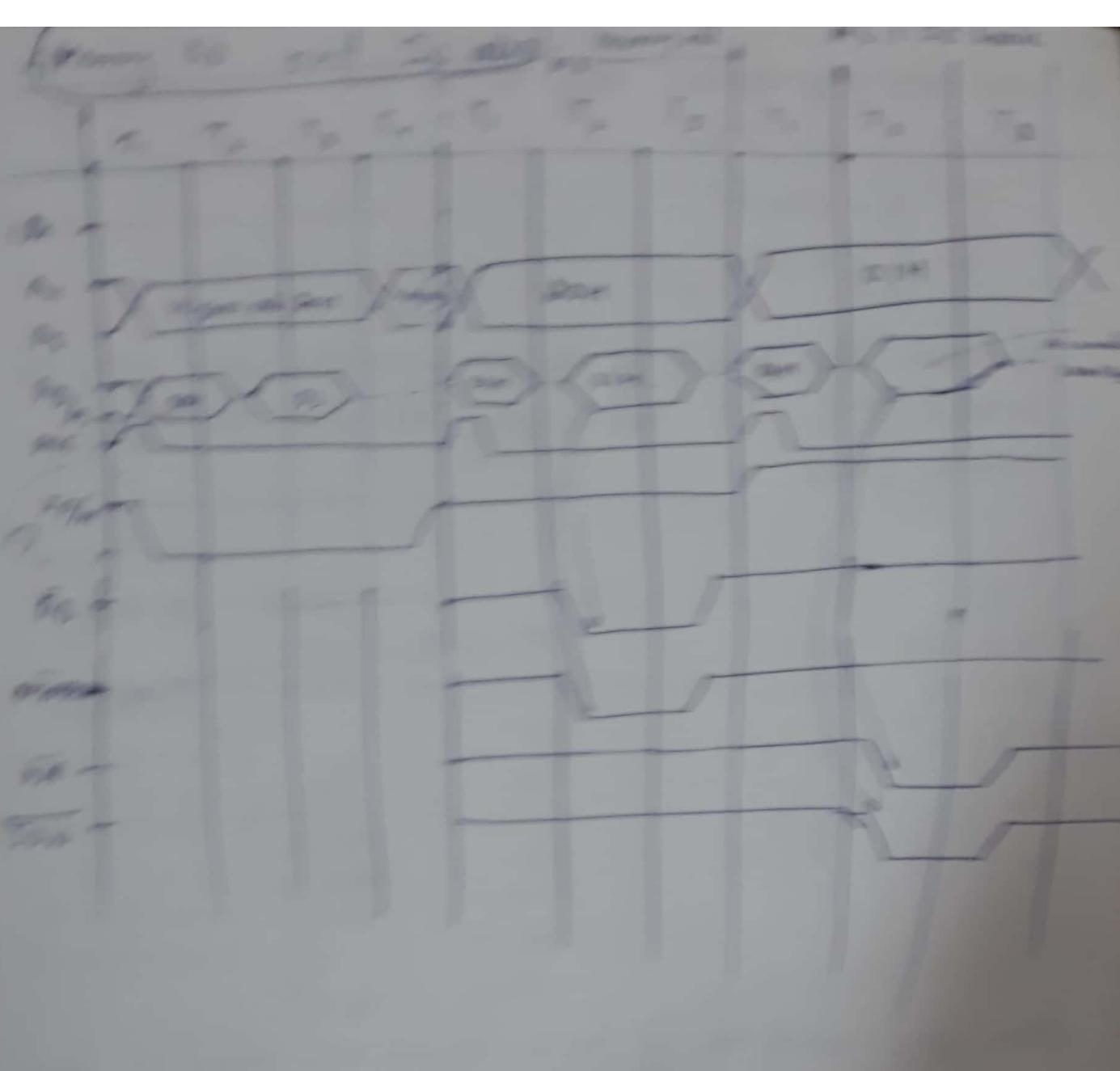
Q. 11

① Load 30H in B

B → 30H

C → 39H

Subtract 39H from 30H and display the contents at port 1.



$$T_{loop} = \left(0.54S \times \underbrace{14}_{T_{stall}} \times \text{count} \right)$$

$$T_0 = \frac{T_{loop}}{\text{clock f}}$$

$UT = DCR B$

$7T - MVI C, \text{count}$

$4T - MOV A, B$

$10T - OUT Port I$

$10T - JMP$

35T

$$T_0 = 175 \mu S,$$

$$1ms = 17.5 \times 10^{-6} + (7 \times 10^{-6}) \times \text{count}$$

$$\text{Count} = \frac{1 - 17.5 \times 10^{-6}}{7 \times 10^{-6}}$$

$$= \frac{(1000 - 17.5)}{7}$$

$$= (140.35)_{10}$$

Q2. Write a program to count from 0 to 9 with 18.8 delay b/w each count. At the count of 9, the counter should reset itself to 0 and repeat the sequence continuously. Use registers pair HL to set up the delay and display each count at one of the output ports. Assume the clock frequency of the microprocessor to be ~~1 MHz~~ 1 MHz

01	Start	MVI B, 00H	(count)
02	Display	OUT Port I	$\frac{10^3 \times 10^6 \times 10}{7 \times 8}$
03			$\frac{10^3 \times 2}{7 \times 8}$
04			A2-8 H.
05			LXI H, 16 bit - Delay (reg pair) \rightarrow (10T)
06			
07			
08	loop	DCX H.	ST

Ex-

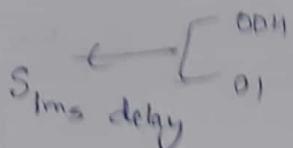
- Q) Write a program to count continuously a hexadecimal number from FFH to 00H in a system with a 0.5 us clock period. Use the register C to set up a 1 ms delay b/w each count and display the number at one of the output ports.

SOP

FFH



00H,



Program

01 MVI B, 00H ← counter for hex (71)
02
03 Next DCR B
04 MVI C, count ← delay count (71)
05
06 delay DCRC
07 JNZ delay. (4T)
08 (10T)
09
0A - MOV A, B
0B OUT PORT 1 (4T)
0C (10T)
0D JMP Next (10T)
0E

loop within a loop

MVI C, FFH



Loop 1



DCR B



JZN loop 2

$$T_{L1} = 1783.5 \mu\text{s.} \quad (\text{last clock})$$

$$T_{L2} = 56(T_{L1} + 0.5 \times 21T) \\ = 56(1783.5 + 0.5 \times 210) \\ = 56100.46 \mu\text{s.}$$

$$= 56100.46 \mu\text{s.}$$

Additional delay
by adding
NO operation

Disadvantages of Software generated Delay.

- 1) The accuracy of the time delay depends upon the accuracy of the system clock.
- 2) The microprocessor is occupied simply in waited loop, otherwise it would be employed to perform other functions.
- 3) The task of calculating the accurate time delays is tedious.

Solution

→ INTEL 8254 programmable Time chip can be interfaced by the microprocessor but it costs.

1) If there are required to execute a unconditional jump
 then the jumps are change the sequence of operations & if
 there are required when the jump goes to the instruction
 following the jump.

16/01/2018

class 14/01

When doing using a single table

LST	B	3384	H	10
lef	DCA	6		6
Mov	A, C			4
Ors	B			4
Sum	leap			10
				10

$$(3384)_{10} \rightarrow (\quad)_{10}$$

$$3 \times 10^3 + 3 \times 10^2$$

$$+ 8 \times 10 + 6 \times 1$$

$$(3473)_{10}$$

$$= 3473 + 7 \cdot 60 + 1 \cdot 6$$

$$= (11276)_{10}$$

$$T_0 = T_1 + T_2$$

D

$$T_1 = 2 \times 1.7 \times 0.5 \times \frac{0.5 \times 13188}{1394} = (3.1 \times 0.5)$$

$$= 15.88545$$

$$T_2 = 10 \times 0.5$$

$$= 5$$

$$T_0 = 5 + 15.88545$$

$$= 21.88545$$

Assignment - 3

- (Q1) The set of 8 readings is stored in a memory location starting at XX50H. Sort the readings in the Ascending order and write the program.
- (Q2) The following block of data is stored in a memory location XX58H to XX5AH. Transfer the data to the location XX80 to XX85 in the reverse order.

Counters and Time Delays

T states

MVI C FFH 4T

loop DCR C 4T

SNE loop 10T/7T

* Clock frequency = 2 MHz

System clock period = $T_c = \frac{1}{f} = 0.5 \mu s$,

Total delay = Toutside loop + Tloop

$$T_{\text{loop}} = 7 \times 0.5 = 3.5 \mu s,$$

Tloop PF₁₀ $\rightarrow (055)_{10}$.

$$\begin{aligned} T_c &= T_s \times 14T \times 3T \\ &= 0.5 \times 14 \times 3T \\ &= 17.85 \mu s \end{aligned}$$

*

$$= 17.85 = (3T \times 0.5)$$

D 0015 = 5

$$= 17.85 \mu s = T_c \left(\frac{3T}{0.5} \right)$$

(adding 2 T since)

$$T_{\text{delay}} = 17.85 + 3.5$$

$$= 17.87 \mu s$$

$$\approx 1.8 \text{ ms.}$$

Assignment-3

(Q-6) The set of 3 readings is stored in a memory location starting at XX50H. Sort the readings in the ascending order and write the program.

(Q-7) The following block of data is stored in a memory location XX55H to XX5AH. Transfer the data to the location XX80 to XX85 in the reverse order

Counters and Time Delays

Ex MVI C, FFH \downarrow states. 7T
 Loop DCR C 4T
 JNZ Loop. 10T/7T.

$$\approx \text{Clock frequency} = 2 \text{ MHz}$$

$$\text{System Clock Period} = T_c = \frac{1}{2} = 0.5 \mu\text{s},$$

$$\text{Total delay} = T_{\text{outside loop}} + T_{\text{loop}}$$

$$T_{\text{outside loop}} = 7 \times 0.5 = 3.5 \mu\text{s}.$$

$$T_{\text{loop}}: FFH \rightarrow (255)_{10}.$$

$$\begin{aligned} T_{\text{delay}} &= 1783.5 + 3.5 \\ &= 1787 \mu\text{s} \end{aligned}$$

$$T_L = T_S \times 14T \times 255$$

$$= 0.5 \times 14 \times 255$$

$$= 1785 \mu\text{s}$$

∴

$$= 1785 - (3T \times 0.5)$$

~~0.5 μs~~

$$= 1783.5 \mu\text{s} = T_L \left(\frac{3T}{\text{starts}} \right)$$

Q1 A set of 10 readings is stored in a memory location XX50H. The end of the data string is indicated by the byte 00H. Add this set of readings. The answer may be longer than FFH. Display the sum at port 1 and port 2.

Sol/

00 — Start LXI H XX50H
01
02
03 — MVI C,00H Clear for sum
04
05 — MOV B,C Clear for count
06 — Next Byte MOV A,M
07 — CPI 00H
08
09 — ~~JZ~~ Display DA
0A
0B
0C — ADD C
0D — JNC Save
0E
0F
10 — INR B
11 — Save MOV C,A
12 — IN XH
13 — JMP Next Byte
14
15
16 — Display MOV A,C
17 — OUT Port 1
18
19 — MOV A,B
1A — OUT Port 2
1B
1C — HLT

Assignment

Q6 The location in the

Q7 The location in the l

Count

Er

A

Syllabus.

(Goonakar) (6th edition)

Chapter-1 (Introduction + basics)

Chapter-2 (Entire thing).

Ch-3 (upto, 3.2.6, 3.2.7 (read),

Ch-4) till 4.4.

4.5, 4.6, 4.7 just read through this.

Ch-5 (~~if you go through all 4 chapters,~~

~~you find it easy.~~

~~in case you find any topic difficult, tell me.~~

till(5.4)

Ch-6 → / Entire chapter)

Ch-7 → Full

Ch-8 → Full

Ch-9 Full

8086 → ~~internal~~ architecture

Pin diagram.

Comparison b/w 8085 & 8086.

The entire process requires 5 machine cycles and 18 T states

(A) CALL 2070 H

opcode fetch cycle

- ① for M₁, it is opcode fetch.
- ② The contents of PC placed on the address bus.
- ③ The instruction is fetched
- ④ PC is updated to 2041 H
- ⑤ It is decoded and executed
- ⑥ SP is decremented by one.

(B) M₂ → memory read cycle.

M₃.
read the contents and put them in
W2 register pair.

(C) M₄ & M₅

contents of W2 will be placed in PC

Read from notes given by Ma'am

(Sep PC)

We can have conditional calls also,

call on carry, zero, sign etc

comparison call & PUSH / POP

✓ ↓
stack goes to pointer → register pair goes

Syllabus

Architecture of 8086

Pin configuration

Comparison b/w 8085 & 8086

(Ch-9)

Ex- # of subroutines

2030 LXI SP 2400 H



2040 CALL 2070 H

2041

2042

2043

Next instruction



205F MLT

2070

1st subroutine instruction



RET

2080



2098
2BFF

Other subroutine

23FF

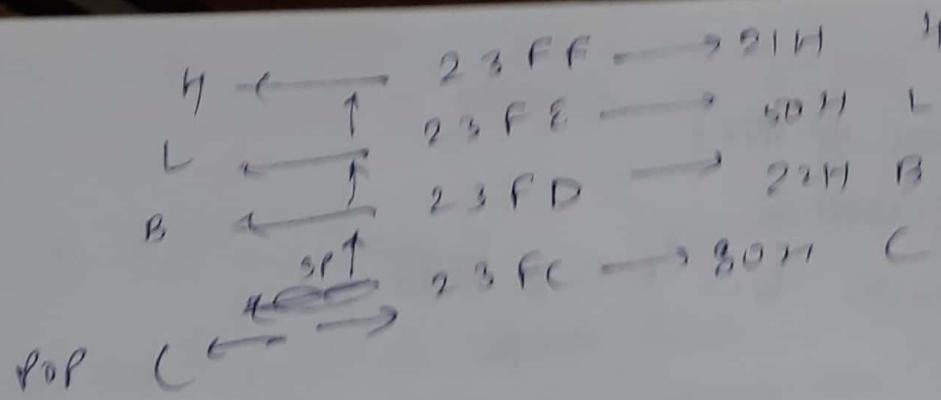
empty space ↑

2400

→ SP is initialized

we put
the contents
in this,

w, z
Register
Pair
Temp



Subroutines

(Call, Return (RET))

(Call)
① It is a 3 byte instruction.

② It stores the content of the program counter PC on the stack.

③ It decrements the stack pointer register by 2.

④ Jump unconditionally to the memory location specified by second & third byte.

(Return)

① It is a 1 byte instruction.

② It inserts the 2 bytes from the top of the stack into PC.

③ It unconditionally returns from the subroutine.

(traffic light example)

07	MOV	LXI	BP	2400 H
08		LXI	H	2150 H
09		LXI	B	2280 H
0A		Mov	A, M	
0B	PUSH		H	
0C	PUSH		B	
0D	PUSH		PSW	
		,		
		,		
)		

1 F.

2020	POP	PSW
2021	POP	B
2022	POP	H,

	Acc	Flag	
B	22	80	C
D			E
H	21	50	L

Q) The available memory range is from 2000 to 23ffH.
A program of data transfer and arithmetic operation
is stored in the memory location from 2000H to
2050H, & SP is initialised at 2400H.

To generate 2 sets of data stored.

2150 H

380 H

To use the unused which are generally in lost.

Program status word mean

Pop @

- ① It is a one byte instruction
- ② Copies the contents of the top two memory location of the stack into specified register pair,
- ③ AND places the contents of the memory located indicated by the stack pointer is copied into the lower address register (C).
- ④ Stack pointer is incremented by one.
- ⑤ Contents of the next memory location are copied into the higher order register.
- ⑥ Stack pointer is again incremented by one

Ex -

POP R_P

POP B

POP

POP H

POP PSW

Simple Instruction

Q000 LXI SP 2099H

Q003 LXI H 42F2H

(RJ) Q006 PUSH H

About Delay

Q010 P0H H

Push

Pop

SP → 2099

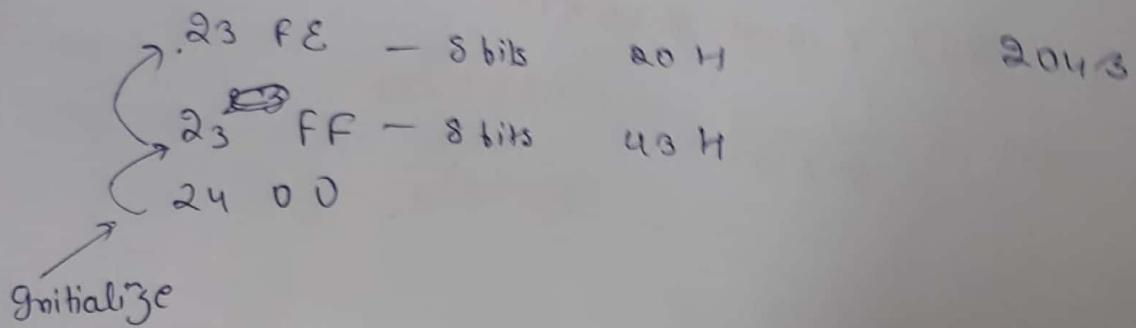
2098 → H → 42H

2097 → L → F8H ← SP → L

Stack & SubroutineStack - PUSH & POP.

PUSH :- The data bytes in the register pair of the microprocessor can be stored to a stack at a time in the reverse order by using the PUSH operation.

POP :- The data byte can be transferred from the stack to the register pair.



In a single instruction, SP is decremented or incremented by 2.

PUSH instruction is one byte instruction.

Ex. LXI SP, 16 bit

① PUSH R_p

Push B

Push D

Push H

Push PSW (Program Status Word)

- ① Stack pointer register is decremented by one, and the content of the higher order register B are copied into the location ② The stack pointer is again decremented by one. The contents of lower order register C in memory are given in the stack pointer.

The contents of ZER flag directly; therefore the contents have to be pushed into the accumulator and then ORed.

CPI M0H

It compares the contents of Register B to every cylinder. When the register B reaches the number ON H, the program sequence is redirected to reset the counter 0, without displaying ON H.

Note: $T_L = \text{SUT} \times \text{count} \times 1.0 \times 10^{-6}$

$$\text{count} = \frac{1}{84 \times 10^{-6}} = (41666)_{10}$$
$$= (A8C3)_{16}$$

$$T_{\text{total}} = T_0 + T_L$$
$$16 = (43 \times 1 \times 10^{-6}) + (84 \times 1 \times 10^{-6} \times \text{count})$$
$$C = (41665)_{16}$$

short assignment (give in one week)

09	MOV	A, L	4T
0A	OR A	H	4T
0B	JNZ	loop	10T
0C			
0D			
0E	JNR Mov	B A, B.	4T
0F	CPI	0AH	7T
10			
11	JNZ	display	10T
12			
13			
14	JZ	start	10T

$$T_D = 1 \text{ sec}$$

$$T_C = 1 \mu\text{s},$$

HL register pair is required because delay is too large (1 second), wouldn't be able to store in a single register.

ORA H

It is used to OR the contents of H and L registers in brackets to catch the zero flag. Since it can not be checked directly, the contents of L have to be moved to A. The zero flag is set only when the contents of both H and L are zero.

DEC H

This inst. does not recognize and acknowledge

the content
contents h
and the
CPI # of
of comp
cycle.
the pr
counter

T

short