

Types

① Reprogrammable

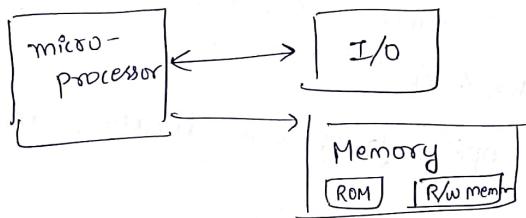
user directly can use
the mp. and give
instructions.

Class-1
31/12/2018

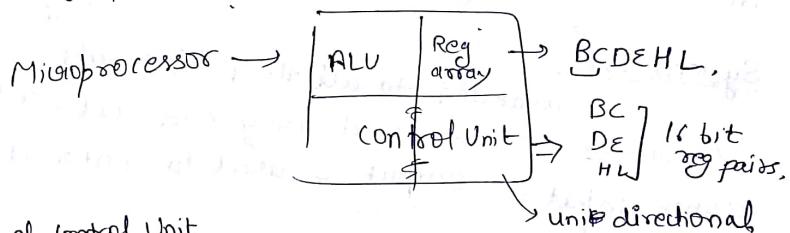
② Embedded System

↓
part of a bigger system Ex-Washing machine
Car-dash board

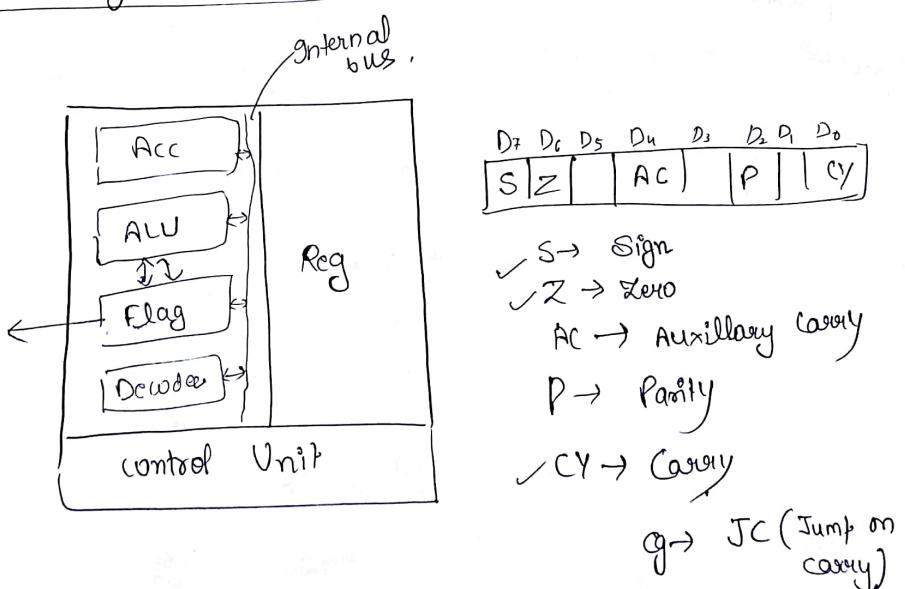
Instruction set is converted to binary data by MP.



2 types of program counter, Sequence counter.



→ Function of control Unit



Addition operation can not be performed directly, but use AC.

PAD instruction, allows content of register directly

Diff b/w Program counter, and stack pointer.

→ Stack pointer is present in R/W memory

→ PC is present in register

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.

Chapter-1 is done.

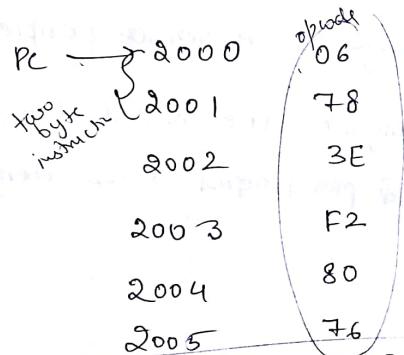
Ch-2 (instructions)

Gyankar
6th edit by
8085

class 2,3
8/12/2018

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.



MVI(B) 78H

MVI(A) F2H

ADD B

HLT

(MVI) →
More
immediate

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, 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

=

Peripherals

at microprocessor \rightarrow Upr.

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 0 memory address ~~go the~~.

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

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

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

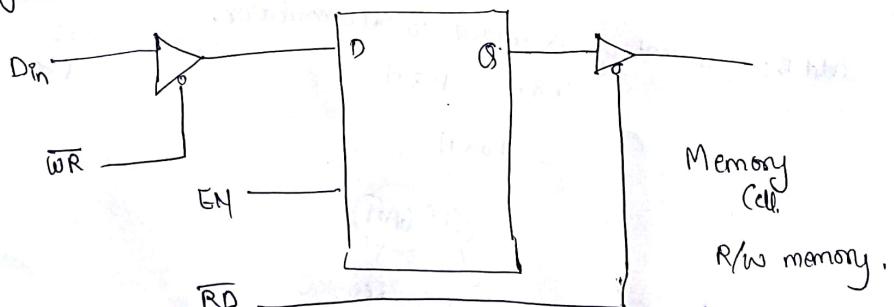
DMA data transfer

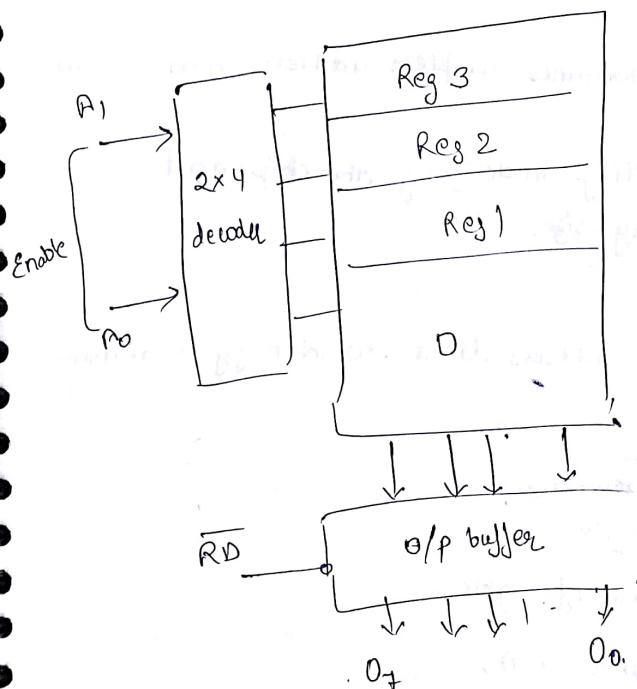
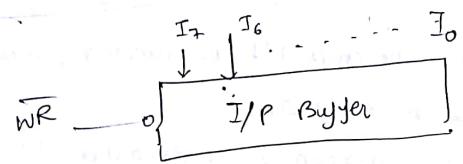
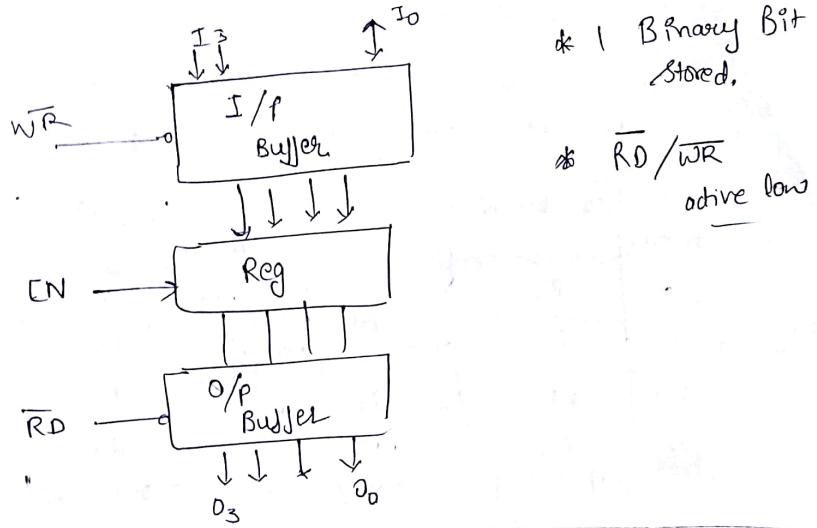
Direct Memory access:

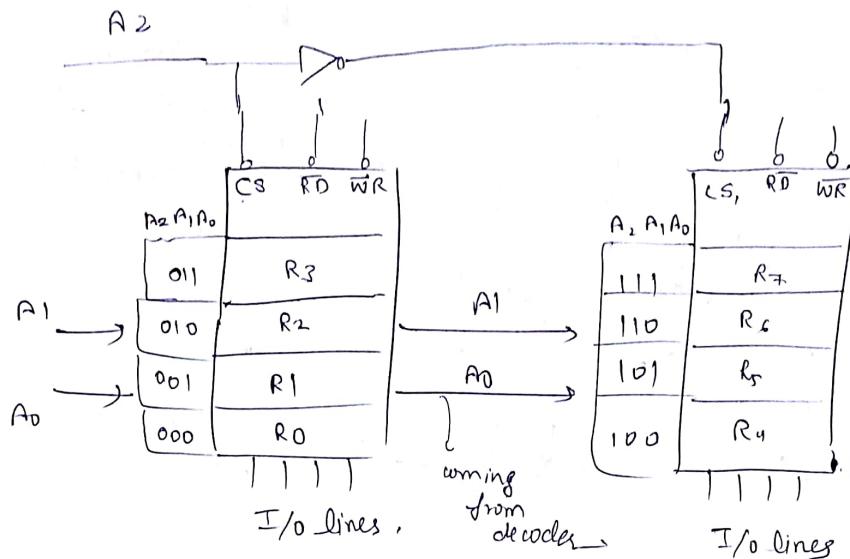
RAM :- flip-flop.

ROM \rightarrow Diode (forward or reverse biased)
contents can't be changed

Memory Cell







- ① When A₂ is active low, the chip M₁ is selected, when A₂ is active high, chip M₂ is selected.
- ② Address line A₁ and A₀ determines individual address of registers.
- ③ A₂, A₁ and A₀ determines complete address from 000 to 111.
- ④ 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}

\Rightarrow No. of registers

address lines $\rightarrow n$,

- (2) The memory chip requires, 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 of all the address lines.
- (3) The control signal RD enables the output buffer and the data is selected from a register and is made available at the output.
- (4) WR enables the input buffer and the data on the input lines is written into the memory cell.

Arithmetic Operations.

Addition and Subtraction,

- 8 bit data
→ contents of memory location,
→ contents of a register
- 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
acc is shifted
either left or right
to the next position.

→ any 8 bit number
or the contents of
a register or
a memory location

③ Compare

→ the contents of the
acc can be complemented
i.e., all bits get replaced
by 1 and vice-versa

④ Complement

contents for equality
greater than or less
than

Branching Operations

can either be conditional or unconditional.
Two types

(1) Jump

(2) Conditional jumping are

(3) Call, return
2. Branch

These instructions change the sequence of the program by 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.

Instructions

Three types

(1) 1 byte instruction

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

| Opcode | 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 opcode and the operand.

| Opcode | 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 opcode, and the second and third byte specifies 16 bit address.

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

Opcode

8085H

~~(3A)~~ Hx
~~(50H)~~ lower
~~(00H)~~ higher order

JMP

8085H

C3

85H

00H

Instruction Sets

8085

R - 8 bit Register

M - Memory Location

Rs - Source Register

Rd - Destination Register

Rp - Register pair.

Data Transfer Operations

→ Mnemonics

Eg.

| | | | |
|--------|------------|------|----------|
| ① MVI | R, 8 bit | MVI | B, 4FH |
| ② MOV | ③ Rd, Rs | MOV | B, A |
| ④ LXI | Rp, 16 Bit | LXI | B, 8050H |
| ⑤ IN | 8 bit | IN | 0FH, |
| ⑥ LDA | 16 Bit | LDA | 8050H |
| ⑦ STA | 16 Bit | STA | 8070H |
| ⑧ LDAX | Rp | LDAX | B |
| ⑨ STAX | Rp | STAX | D |

⑨ MOV R,M

MOV B,M

⑩ MOV M,R

MOV M,C

③ → LXi

Load the 16 bit data in the register pair
B, 8050H → 50H in C

load immediate

50H in B.

④ JN

~~get~~ device gt accepts the data from the input device
and places it in the accumulator.
07H → stored in A.

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

⑥ STA is reverse of LDA,

↳ pushing the data from Accumulator to 16 bit
address

⑦ LDX

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 →

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

Find it by yourself

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

Arithmetic Operations.

| Mnemonics | E.g. |
|-----------------------------|---|
| ① ADD R | ADD B |
| ② ADD 8-bit | ADD 3FH |
| ③ ADD M | ADD M |
| ④ SUB & R, SUB 8 bit, SUB M | |
| ⑤ INR R | INR D |
| ⑥ INR M | INR M — HL |
| ⑦ DCR R | |
| DCR M | |
| ⑧ INX Rp | INX D <small>Concurrent Immediate</small> |
| ⑨ INX R _p | DCX B, |

Logical Operations

| | AND, OR, XOR. |
|-------------|--------------------|
| ① ANA, R | ANA B, |
| ② ANI 8 bit | ANI 3FH |
| ③ ANA M | ANA M |
| ④ For OR | OR A, OR I, ORA M |
| ⑤ For XOR | XOR A, XRI, XRA M. |

⑥ CMP R

⑦ CPI 8bit

CMP B

CPI 4FH,

Branch Instructions

① JMP

16 bit add

JMP 2050H

② JZ

16 bit add

JZ 2080H

③ JNZ

16 bit add

JNZ 2080H

④ JC

16 bit add

JNC 2070H

⑤ JNC

"

JNC 2070H

⑥ CALL
→ call for a service routine

CALL 2075H.

⑦ RET

→ return

RET

Machine Control Instructions

① HLT

→ Halt

② NOP

→ No operation.

B

| <u>Address</u> | <u>Data</u> |
|----------------|--|
| 2051H | → 49H |
| 2052H | → 4FH] Add and store the sum in 2053H. |

Class-1
17/11/2018

Stacking Point → 2030H

Ans

Ans { 2030
15A } 2031
3 byte
143H } 2032

| | | |
|------|----------|--------|
| 2033 | LDA | 2051H, |
| 2034 | MOV B,A. | |
| 2035 | LDA | 2052H |
| 2036 | ADD B | |
| 2037 | STA | 2053H |
| 2038 | | |
| 2039 | | |
| 2040 | HLT | |
| 2041 | | |

Ch-4 Pin Configuration

8085 is ~~16 pin~~ 40 pin IC

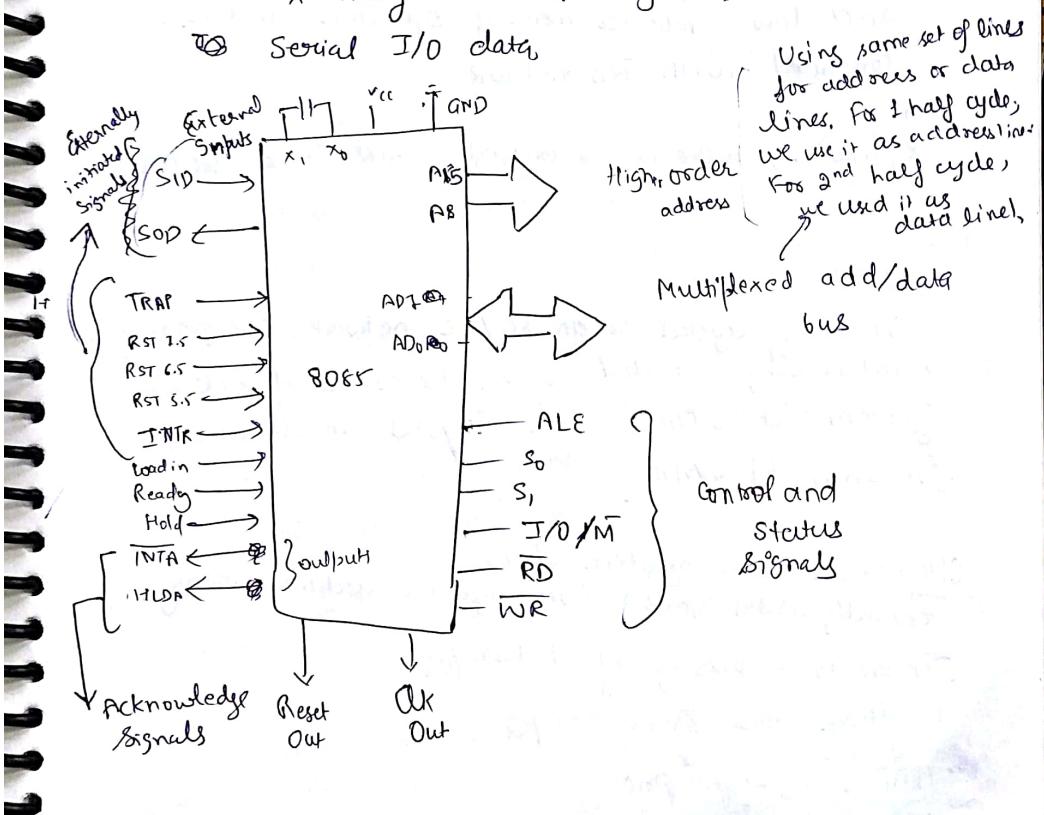
Ch-1, 3 is
done

n - address lines,

2^n - registers.

Basic Information about 8085

- 1) It is 8 bit microprocessor (Data stored is 8 bit)
- 2) It has 16 address lines and hence is 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.
 → Serial I/O data



ALE Address Latch Enable

This is a positive going pulse generated every time 8085 begins the operation. It indicates that AD7 and ADO are address bits. This is used to latch the lower order address from the multiplex bus and generate a separate set of 8 address lines A7 to A0.

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

This is used to differentiate between ~~IO/M~~ IO and Memory operation. High indicates IO operation and low indicates memory operation and is combined with \overline{RD} and \overline{WR} .

B/w

X_1 and X_0 , there is a crystal oscillator or RC/LC network.

X_1, X_0

It is a crystal or an RC/LC 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 interrupt.

There are 5 interrupts

TRAP is highest priority

INTR is lowest priority

When interrupt occurs, interrupt acknowledge done.
INTR and 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 INTA is output.

HLD (Hold acknowledge)
This 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 cycle 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

Steps to fetch data from the location 2005H.

Binary Code - 4FH

Instruction - Mov C, A.

Class-7, 8

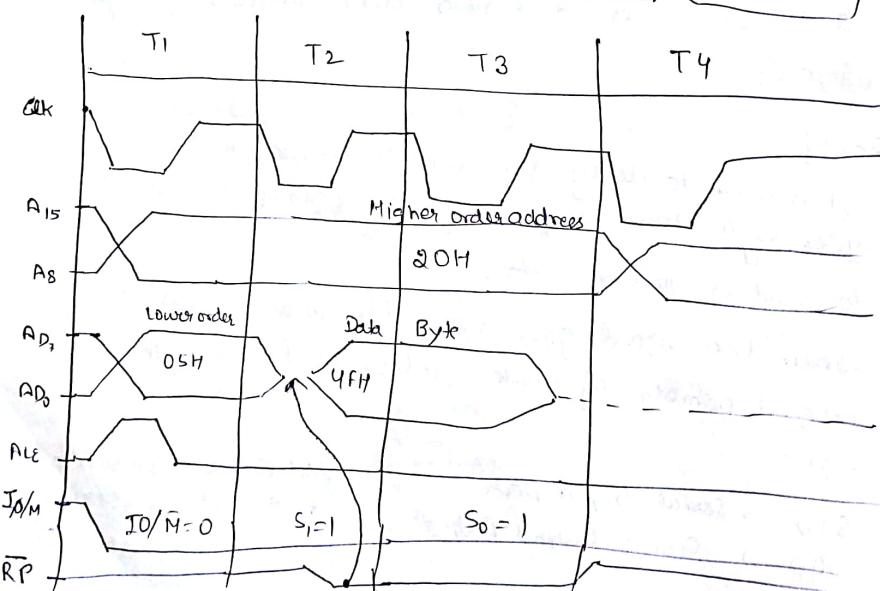
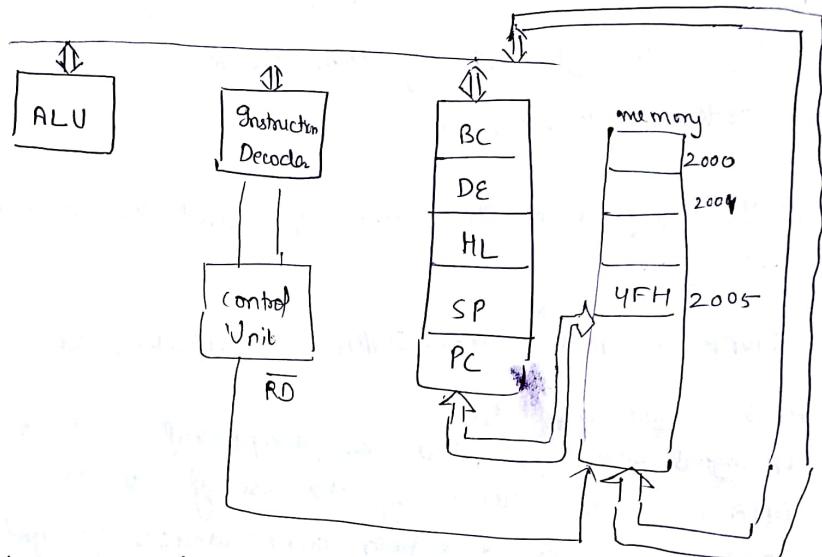
22/12/2018

Class-9

23/12/18

(Not attended)

How this happen in the system?



Crossover: 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 & ie. to read the data from memory location & load it onto data bus.

Step 3:

The byte from memory location is placed on data

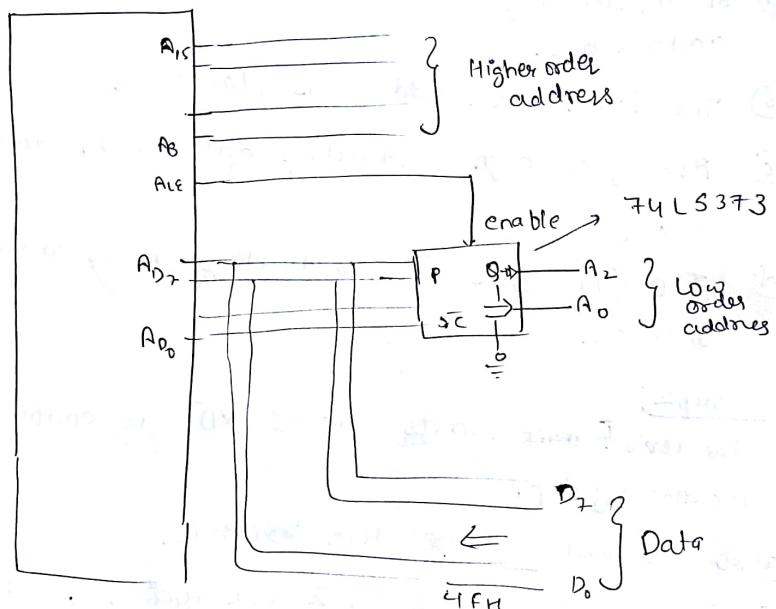
bus.

When \overline{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 (~~by decoded~~)

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

From 3 to 6 clock periods, all instructions executed,



- ① Instruction cycle
- ② Machine cycle (set of operations performed 1-6)
- ③ T states

(every machine cycle require no. of states
(T_1, T_2, \dots, T_r)

- 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, I/O 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,

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

29/11/18
Class-10, 11

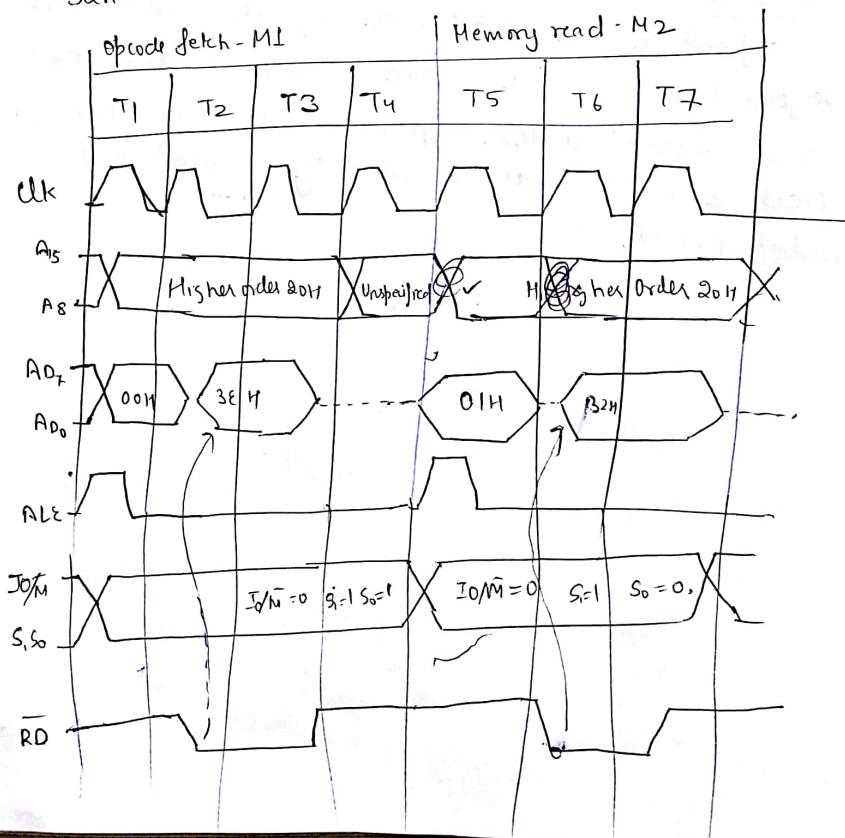
3EH represents the opcode to locate the data byte
in the accumulator and 32H is the byte to
be loaded in the accumulator.

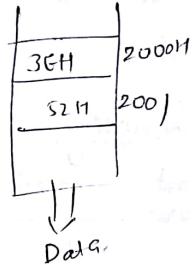
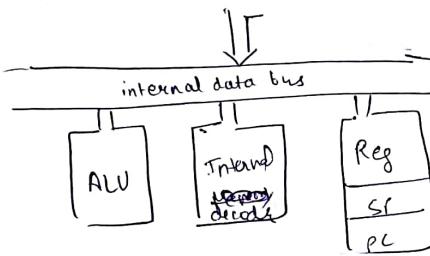
- ① Illustrate the bus timing.
- ② Calculate the time required to execute the opcode
fetch and the memory read cycles and the entire
instruction cycle if the clock frequency is 2 MHz.

Soln:

3EH → 2000H

32H → 2001H





We do not perform any decoding operation in M₂, therefore ~~join~~ not needed
T₄ state

b) $T = \frac{1}{f} = \frac{1}{2 \text{ MHz}} = 0.5 \text{ } \cancel{\mu\text{s}}$.

M₁ = 4 × 0.5 = 2 μs ,

M₂ = 3 × 0.5 = 1.5 μs ,

$\boxed{\text{Total} = 3.5 \text{ } \mu\text{s.}}$

Assignment 1 (After Mid Sem.)

Q31. Write the steps to add two numbers ~~both~~ A2H and 18H, save the sum to the accumulator, also both the numbers should be saved for future use.

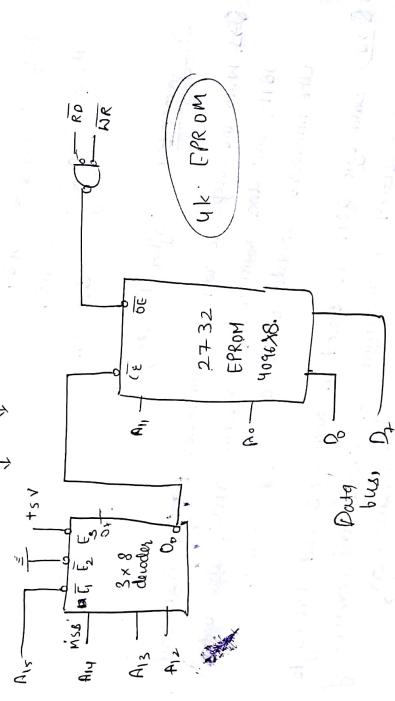
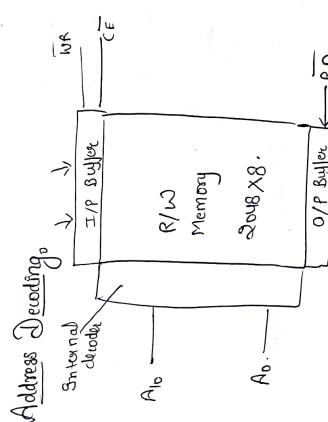
Q32) Data byte 28H is stored in Register B and data byte 97H is stored in accumulator. Show the contents of the registers B and C after the instruction,

MOV A,B

MOV C,B,

have been executed,

$4k \rightarrow 12$ Address lines
 $2^k = 11$



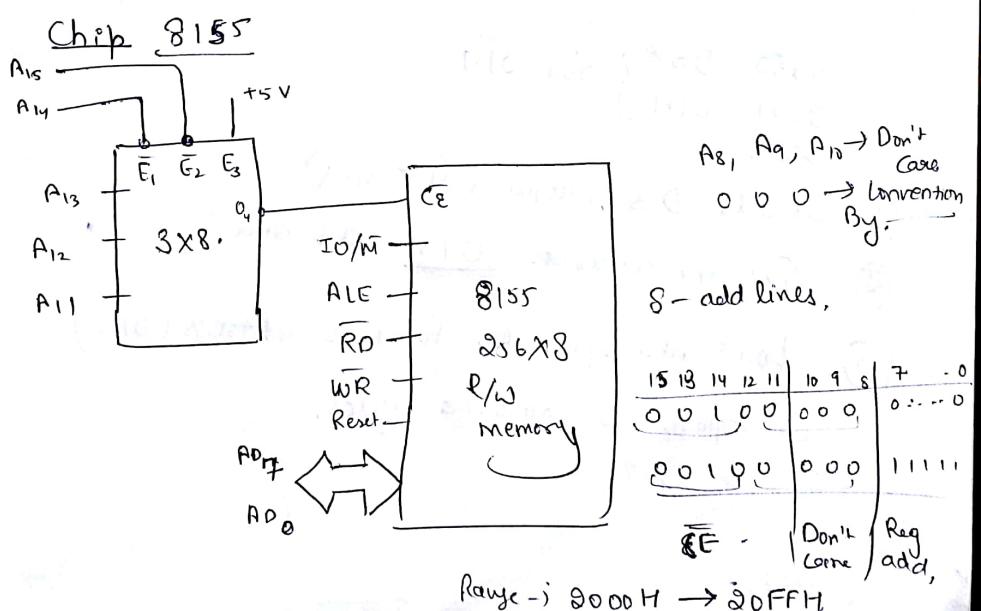
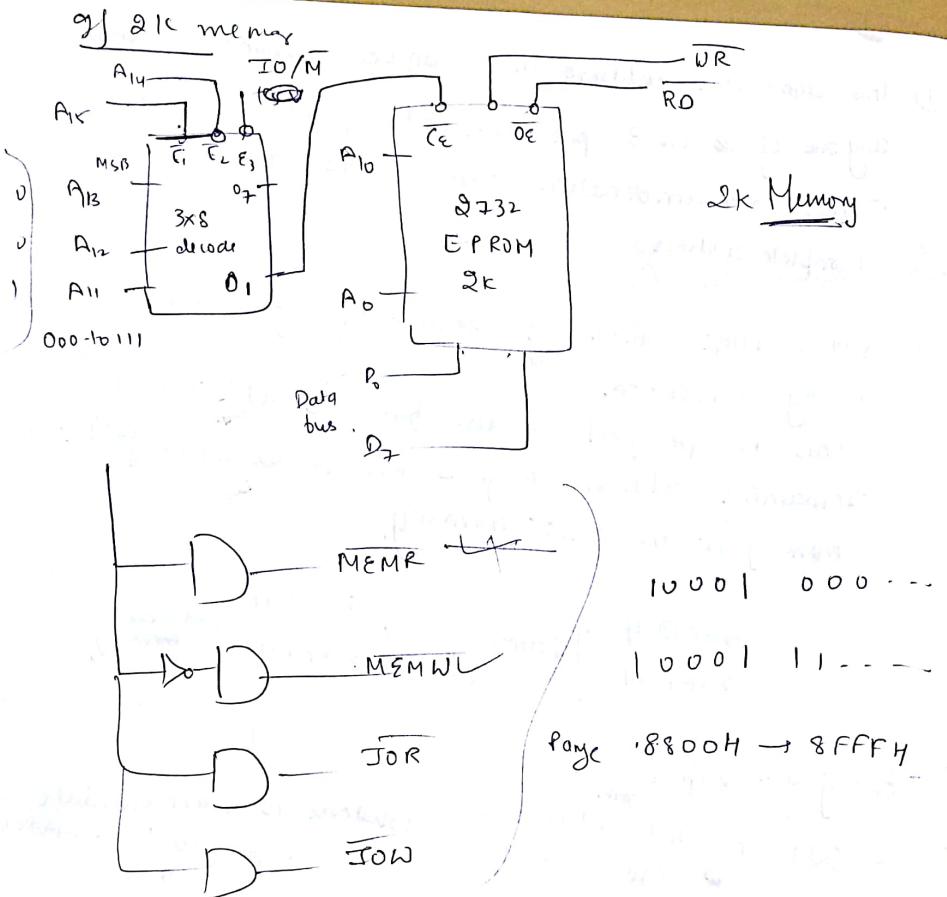
Chip Enable

| | A ₁₅ | A ₁₄ | A ₁₃ | A ₁₂ |
|-------|-----------------|-----------------|-----------------|-----------------|
| 0000H | 0 | 0 | 0 | 0 |
| 0FFFH | 0 | 0 | 0 | 1 |

Memory range

Register address

| A ₁₁ | A ₁₀ | A ₉ |
|-----------------|-----------------|----------------|
| 0 | - | - |
| 1 | 1 | 1 |



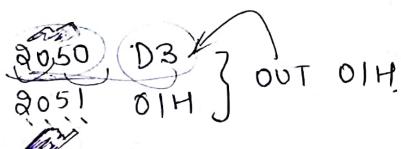
- ① The don't care address lines can be assumed to have any one of the 8 input combinations, 000 to 111. Thus each combination can generate one set of complete address.
- ③ Address range given by ^{don't} assuming to don't cover 000 is by convention. This is specified as the primary address. The remaining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

2000 H 2100 H
20FF H 27FFH] Primary] Mirror memory,

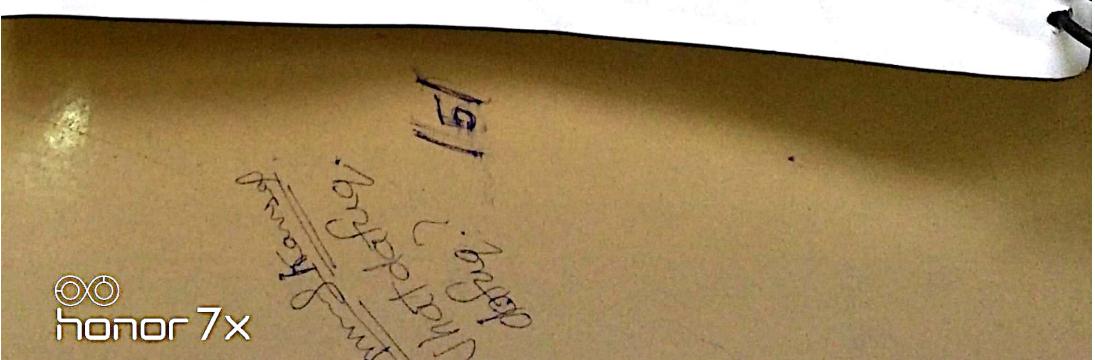
→ End of 4th chapter

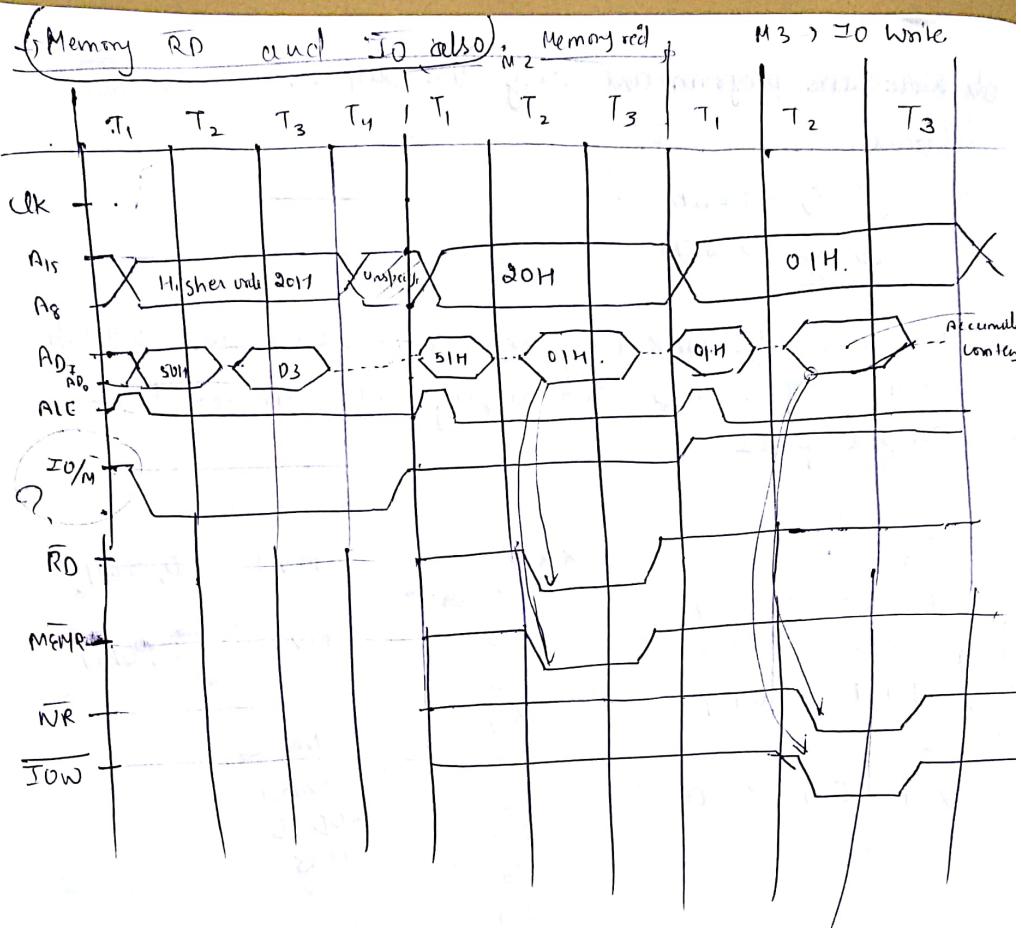
Q → OUT 8 bit address
(10 T states)

whatever is in accumulate
is put to post address
01 H



- ① Decode D3, requires (4 T state)
- ② Get the address 01H (3 T state)
- ③ Load data from AC to this address (01H)
It require 3 machine cycles.





Digitized by srujanika@gmail.com

Digitized by srujanika@gmail.com

 honor 7x

31/1/18
class-12

Q) 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.

| | | | | | |
|---|----------------|-------|--------|----------|--------|
| ③ | 1 000 | 1 011 | X X 00 | MVI | D, 8BH |
| | 0 111 | 0 000 | 01 | | |
| ④ | <u>1</u> 111 | 1 011 | 02 | MVI | C, 6FH |
| | D ₇ | | 03 | | |
| | G = 1 | S = 1 | 04 | INR C | |
| | | Z = 0 | 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) WAP

- ① load 30H in B

$$\begin{aligned} B &\rightarrow 30H \\ C &\rightarrow 39H \end{aligned}$$

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

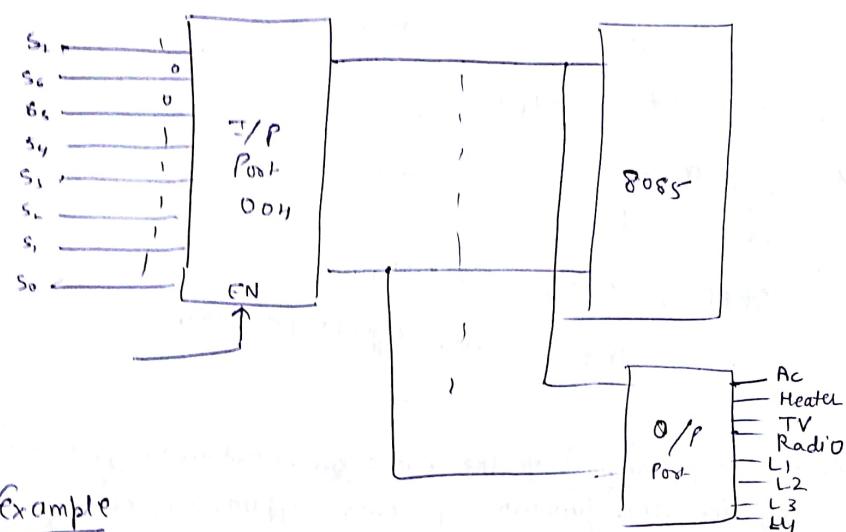
151

Computer
Architecture
W.M.
M.T.
W.M.
W.M.
W.M.
W.M.
W.M.

$B \rightarrow 30H$
 $C \rightarrow 39H$
 Bin: 0011 1001
 1's 1100 0110
 Add: 1100 0111
 30H 0011 0000
 10 0111 0111
 Cy (x=0, s=1, z=0)

Logical Operation

| | | |
|-----|----------------|--------|
| 0 0 | MVI | B, 30H |
| 0 1 | MVI | . |
| 0 2 | MVI | C, 39H |
| 0 3 | | |
| 0 4 | MOV | A, B |
| 0 5 | SUB | @C |
| 0 6 | OUT | PORT A |
| 0 7 | | |
| 0 8 | HLT | |



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

$$\begin{array}{r}
 9FH \\
 80H
 \end{array}
 \begin{array}{r}
 \text{S7} \\
 \text{D001} \quad 1111 \\
 1000 \quad 0000 \\
 \hline
 1000 \quad 0000
 \end{array}
 \text{AND} \quad \boxed{\text{masking}}$$

Write a program,

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

Q \rightarrow) B \rightarrow 93H
 R \rightarrow 15H

ORAB, XRB and CMA

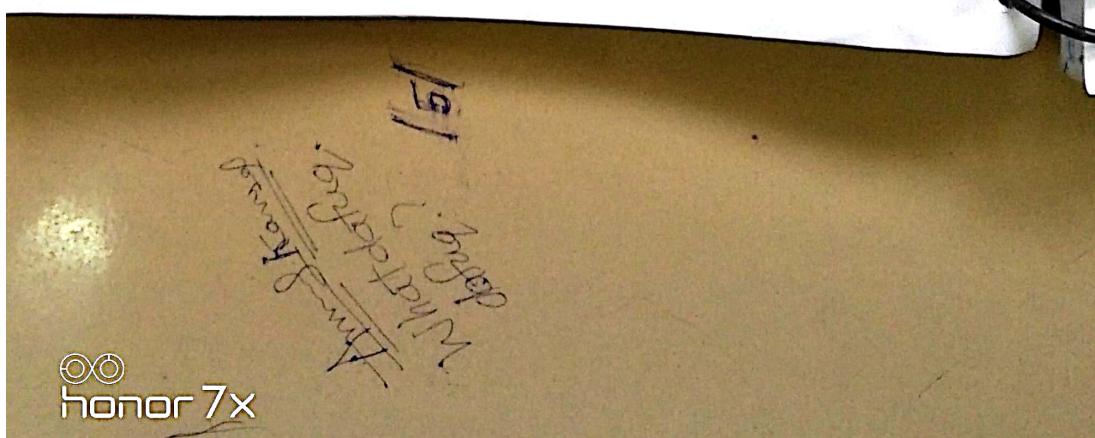
These are three different programs,

Q \rightarrow ^{previous} In the figure, keep the radio on continuously, without affecting the function of other appliances, even if someone turns off switch S₄.

Ans - DR with 10H, 0001 0000

Q.3 In the winter turn off the AC, without affecting other appliances.

Ans, AND with 7FH



UNCONDITIONAL JUMP

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

If sum > FFH displays 01H else the output

| | | |
|----|-----|---------|
| 00 | MVI | 0, 98H |
| 01 | MVI | 5, A#H |
| 02 | MVI | 5, A#H |
| 03 | MOV | A, D. |
| 04 | ADD | E |
| 05 | JNC | DISPLAY |
| 06 | | |
| 07 | | |
| 08 | MVI | A, 01H |
| 09 | | |
| 0A | OUT | 01H |
| 0B | OUT | 01H |
| 0C | HLT | |

Q) 16 bytes of data are stored in the memory location 5/2/2015
class-13,14

XX50 to XX5FH.

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

Opn:

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.

- (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.

Soln:

| | | Next (contents) |
|----|-----------------|--------------------------------------|
| 00 | MVI B, 00H | 13 JNZ NEXT |
| 01 | | 14 |
| 02 | MVI C, 0AH | 15 |
| 03 | | 16 MOV A,B |
| 04 | LXI XX60 | 17 OUT PORT 1 |
| 05 | | 18 |
| 06 | | 19 HLT |
| 07 | NXT MOV A,M | #A HLT NOP |
| 08 | RAL | #B for overload MVI A,FFH |
| 09 | JC REJECT | #E |
| 0A | | #D OUT PORT 1 |
| 0B | RAR | #E HLT |
| 0C | ADD B | #F HLT. |
| 0D | JC OVERLOAD FFH | 20 |
| 0E | | 21 |
| 0F | | |
| 10 | MOV BA | |
| 11 | REJECT NX H | |
| 12 | DCR C | |

*3 type C instruction
1 move reg reg
1 move reg mem
1 move mem mem*

*Overflow detection
Overflow handling*

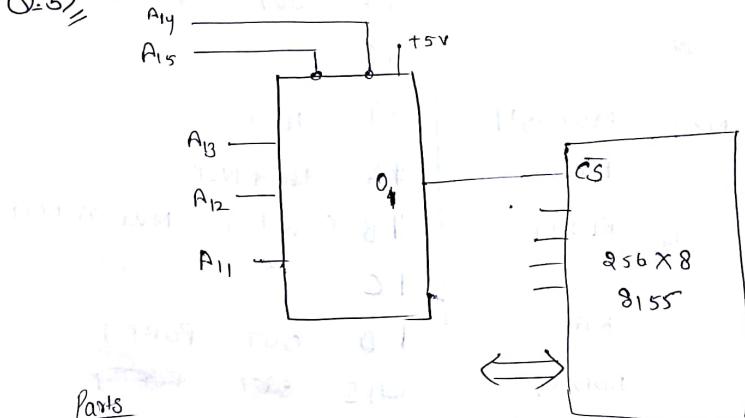
Assignment - 2

Q-3 LDA \rightarrow 2050H 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 (No timing diagram for this query)
- (2) ADI 47H
- (3) STA 2050H
- (4) PUSH B.

Q-5 //



- (1) Specify the memory address range if O_4 of the decoder is connected to \overline{CS} . Specify the hold back memory.
- (2) Specify the memory range if O_7 is connected to chip enable bar signal of the 8K memory.
- (3) Specify the relationship b/w storage of the feedback memory and the don't care lines.

Syllabus.

(Goonkar | (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 → ~~the~~ architecture

Pin diagram.

Comparison b/w 8085 & 8086.

Q → A set of 10 readings is stored in a memory location XX50H. 7/2/2018
class 15

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/ny

```
00 ← Start LXI 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
0A
0B
0C ← ADD C
0D ← INC Save
0E
0F
10 ← INR B
11 ← Save MOV C,A
12 ← IN X H
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
```

151

Wish you all
a very happy
new year

@@
honor 7x

Assignment-3

- Q-6) The set of 3 readings is stored in a memory location starting at $\text{XX}80\text{H}$. Sort the readings in the ascending order and write the program.
- Q-7) The following block of data is stored in a memory location $\text{XX}55\text{H}$ to $\text{XX}5A\text{H}$. Transfer the data to the location $\text{XX}80$ to $\text{XX}85$ in the reverse order.

Counters and Time Delays

Ex- MVI. C, FFH T states.
 loop DCR C 4 T
 JNZ loop. 10T/7T

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

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

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

$$T_0 = 7 \times 0.5 = 3.5 \mu\text{s}$$

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

$$\begin{aligned} T_L &= T_S \times 14T \times 255 \\ &= 0.5 \times 14 \times 255 \\ &= 1785 \mu\text{s} \\ &= 1785 - (3T \times 0.5) \end{aligned}$$

$$\cancel{= 1785} = 1783.5 \mu\text{s} = T_L \left(\begin{array}{l} 3T \\ \text{states} \end{array} \right)$$

$$\begin{aligned} T_{\text{delay}} &= 1783.5 + 3.5 \\ &= 1787 \mu\text{s} \\ &\approx 1.8 \text{ ms} \end{aligned}$$

10T states are required to execute a conditional jump when it jumps to change the sequence whereas 7T states are required when it goes to the instruction following the jump.

18/2/2018
class-16, 17

Time delay using a register pair

| | | |
|----------------|---------|------------------------------------|
| LXI B , 3384 H | 10 | 0.16 |
| loop DCX B | 6 | $(3384)_{16} \rightarrow ()_{10}$ |
| Mov A, C | 4 | $3 \times 16^3 + 3 \times 16^2$ |
| ORA B | 4 | $+ 8 \times 16 + 4 \times 1$ |
| JNZ loop | 10 7 | $= (4092)_{10}$ |
| | | $3 \times 4096 + 7 \times 64 + 28$ |
| | | $= (3188)_{10}$ |

$$T_D = T_0 + T_L$$

$$T_L = 24T \times \frac{0.5 \times 13188}{6594} - (3T \times 0.5)$$

$$= 158254.5$$

$$T_0 = 10 \times 0.5$$

$$= 5$$

$$T_D = 5 + 158254.5$$

$$= 158259.5$$

loop within a loop

MVI C, FFH



Loop 1

↓

DCR B



JZN loop 2

MVI A, 38H

C, FFH

C

↓

Loop 2

MVI

DCR

↓

JNZ

Loop 1

DCR A

JNZ

Loop 2.

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

$$\begin{aligned} 38H \\ = (56)_{10.} \end{aligned}$$

$$T_{L2} = 56(T_{L1} + 0.5 \times 21T),$$

7+4+10

$$= 100.46 \text{ ms.}$$

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 adds to the cost.

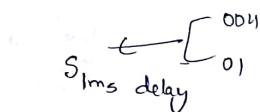


honor 7x

Ex-

- (Q) Write a program to count continuously a hexadecimal number from FFH to 00H in a system with a 0.5 μs 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.

Som . 29 FFH
↓
00H.



Program

01 MVI B, 00H ← counter for hex (7T)
02
03 Next DCR B (4T)
04 MVI C, count ← delay count (7T)
05
06 delay DCRC (4T)
07 JNZ delay. (10T)
08
09
0A - MOV A, B (4T)
0B OUT PORT 1 (10T)
0C
0D JMP Next (10T)
0E

151

$$T_{loop} = \left(0.5 \mu s \times \underbrace{14 \times \text{count}}_{T_{clock}} \right)$$

$$T_0 = \text{Clock} \#$$

$$\hookrightarrow T_0 = 35 \times 0.5 \times 10^{-6}$$

$4T \neq \text{PDRB}$

$$T_0 = 175 \mu s,$$

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

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

$4T - \text{MOV A,B}$

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

$10T - \text{OUT Port 1}$

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

$10T - \text{JMP}$

$$= (140, 35)_{10}.$$

Q2. Write a program to count from 0 to 9 with 18 μ s 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

| | | | | |
|----|----|---------|------------|-----------------------------------|
| 80 | 01 | Start | MVI B, 00H | (Count) |
| | 02 | Display | OUT Port 1 | $(10T) \frac{10^3}{10} \times 10$ |
| | 03 | | | 0.5 μs |
| | 04 | | | 0.5 μs |
| | 05 | | | 0.5 μs |
| | 06 | | | 0.5 μs |
| | 07 | | | 0.5 μs |
| | 08 | Loop | DCX H. | $\xrightarrow{\quad} 6T$ |

$LXI H, 16\text{bit-Delay (reg pair)} \rightarrow 10T$

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

$$T_D = 1 \text{ sec} \quad T_C = 1 \mu\text{s},$$

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.

DCX H

This instr. does not recognize and acknowledge

the contents of zero flag directly. Therefore the contents have to be pushed into the accumulator and then ORed

CPI #0AH.

It compares the contents of registers B in every cycle. If then the register B matches the number DA H, the program sequence is redirected to reset the counter 0, without displaying DA H,

$$\text{Assume } T_L = \text{aux} \times \text{count} \times 1.0 \times 10^{-6}$$

$$\text{Count} = \frac{1}{\text{aux} \times 10^{-6}} = (41666)_10 \\ = (A2C2)_H$$

$$T_{\text{total}} = T_0 + T_L \\ = (45 \times 1 \times 10^{-6}) + (84 \times 1 \times 10^{-6} \times \text{count})$$

$$C = (41665)_10$$

short assignment (give in one week)

Understand memory hierarchy
and how cache performance is affected during memory access
by changing cache size, cache replacement policy
and memory placement policy
and then write a C program which finds out the effect of each of these factors on memory access time.

Syllabus is upto 9th chapter

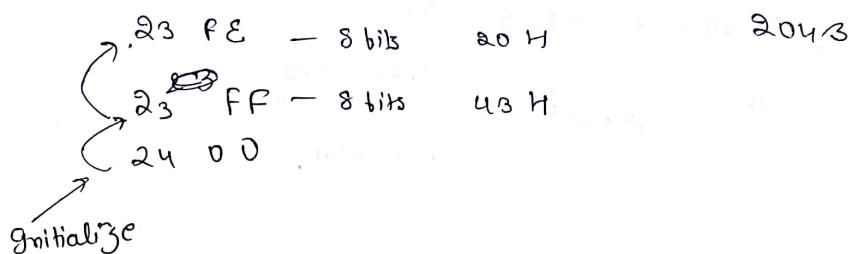
21 Feb
class '18

Stack & Subroutine

Stack - 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 bytes 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 copied into the given location (in the stack pointer).

Program status word

mean

Pop Ⓛ

- ① Ⓛ is a one byte instruction
- ② Copies the contents of the top two memory location of the stack into specified register pair.
- ③ Ⓛ pushes 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
 POP B
 POP
 POP H
 POP PSW

Simple Instruction

2000 LX1 SP 2099H

2003 LX1 H 42F2H

2006 PUSH H

2007 Delay

2010 P0H H

Push

Pop

SP → 2099

2098 → H → 42H

2097 → L → F8H

SP → L

Eg:

| | | | |
|------|------|------|--------|
| 2000 | LXI | SP | 2400 H |
| 2003 | LXI | H | 2150 H |
| 06 | LXI | B | 2280 H |
| 09 | MOV | A, M | |
| 0A | PUSH | H | |
| 0B | PUSH | B | |
| 0C | PUSH | PSW | |
| 0D | | | |

IF.

| | | | | | |
|------|-----|-----|---|-----|------|
| 2020 | POP | PSW | B | Acc | Flag |
| 2021 | POP | B | D | 22 | 80 |
| 2022 | POP | H, | H | 21 | 50 |

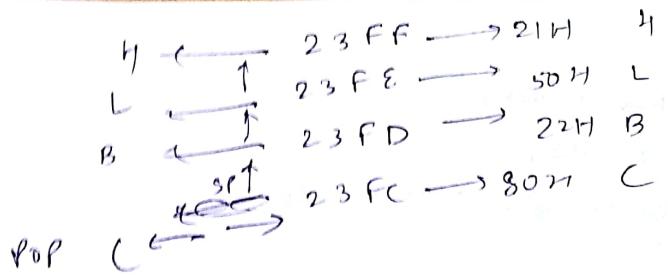
| B | C | E | L |
|---|---|---|---|
| | | | |
| | | | |

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

~~There~~ 2 sets of data stored.

2150 H
2280 H

To use the ~~unused~~ which are generally in lost.



Subroutines

(Call, Return (RET))

(Call)

- ① It is a 3 byte instruction.
- ② It stores the content of the program counter 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)

Syllabus

Architecture of 8086

Pin configuration

Comparison b/w 8085 & 8086

(Ch-9)

Ex- Inv of Subroutines

2020 LXI SP 2400 H

2040 CALL 2070 H

2041

2042

2043

Next Instruction

205F

HLT

2070

1st subroutine instruction

207F

RET

2080

↓

2098

Other subroutine

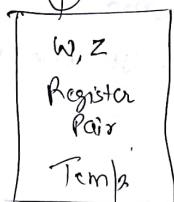
23FF

empty space

2400

→ SP is initialized.

we put
the contents
in this,



151

Microprocessor
Microcontroller
Microchip
Microsystem

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 & M₃ → memory read cycle.

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

(C) M₄ & M₅

contents of WZ will be placed in PC

Read from notes given by Malam

(See PIC)

We can have conditional calls also,

call on carry, zero, sign etc

Comparison call & PUSH / POP

→ register pair goes

stack goes to pointer