

Linear vs Non-Linear Execution.

04

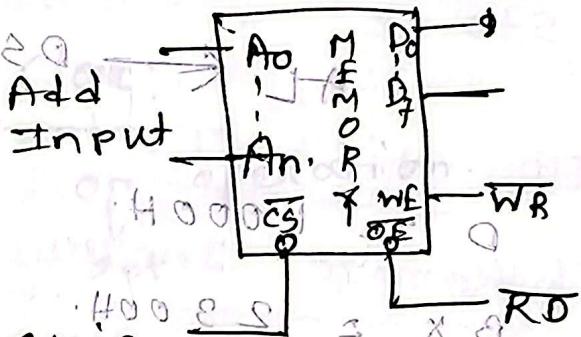
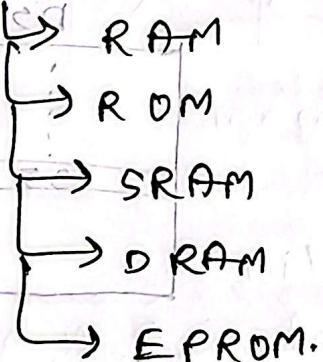
Nipa Maldam A

Ch-10
Very
Brey

Connected Devices

Memory Interfacing

→ Memory Unit:



Chip Select
(which chip should be selected for use)

Defines

→ Address Line: what will be the memory size [$n \rightarrow 2^n$]

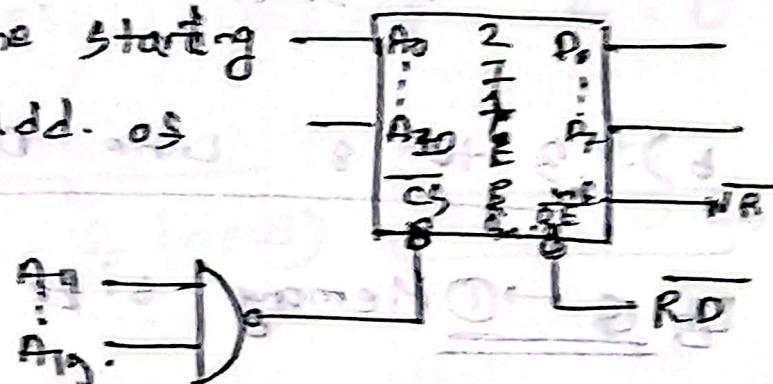
→ Memory Interfacing:

MOV → used

In/out → Not used X

① Find Memory size:

→ ② Determine starting and final add. of memory.



In 8086 → Total Address Line $\rightarrow 20$

In Memory device \rightarrow Total Address Line ≤ 20



Bit mismatch.

Solution: ① Address Decoding:

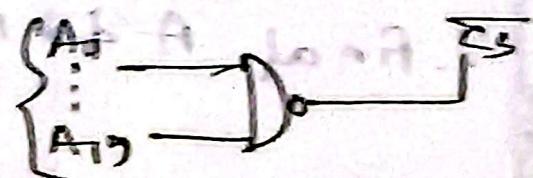
→ NAND Gate Decoding.

→ 3 to 8 Line Decoder

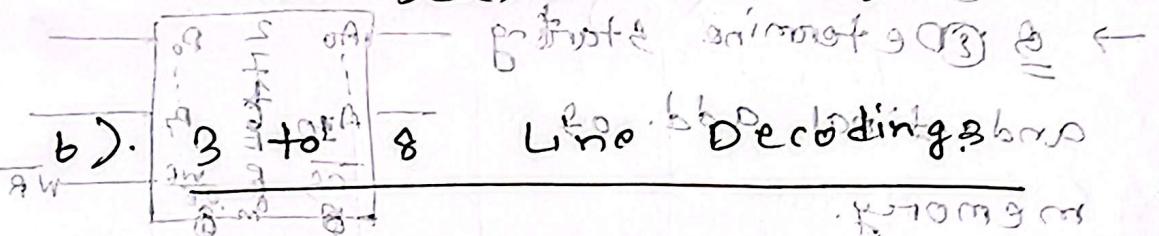
a) NAND Gate Decoding:

⇒ only one memory is selected

Loss of
bits -



→ Not feasible in high Application.



Q → ① Memory size = $2^n = 2^{11} = 2048$

n = Address Line = 11 (Here)

$$2048 = 2^k \times 8$$

Q → 2: if starting address of memory is 1000 then starting and final add. of program will give output ← gives program and final add. of (fido)

$$s = 0 \rightarrow CS = 1$$

$$s = 0 \rightarrow CS = 1$$

$$s = 0 \rightarrow CS = 1$$

→ NAND
indicates the

$A_{12} A_9 A_{10} A_9 A_8 A_7 A_6 A_5$

~~12259~~ 1 0 0 0 0 0 0

W.M.B.P. (1) 18

8
B5C6 9-6 8

Address : 11800 H
Boulevard at 86th

卷之三

fibrosis often

F

21 February 1910

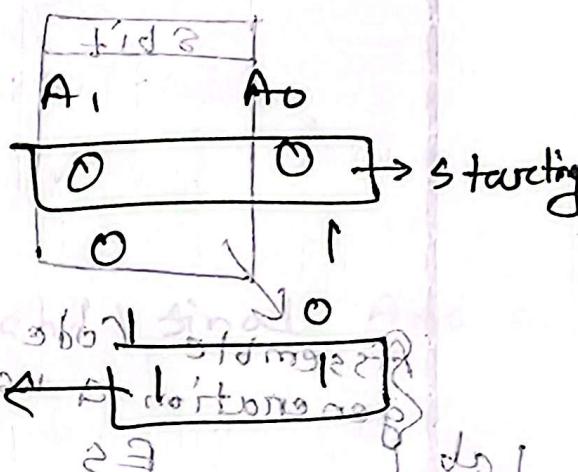
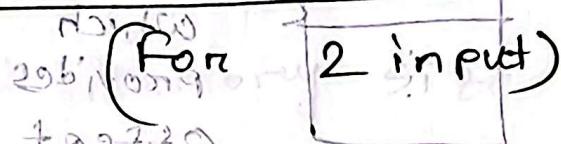
Address: FFFF

1. $\frac{1}{2} \times 10^3$ m^3 min^{-1}

(PA) 10

Second Address: $A_0 = 0, A_1 = 0$

Sequence Technique:



Alternative to determine final Address:

$$\text{Final Address} = \text{Starting Address} + \text{Memory Size}$$

$$= 1F1D800 + 00800$$

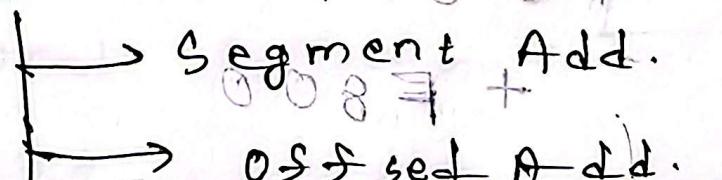
$$+ 00800 + (0) \times = FF800 + 007FF \quad (2^N - 1)$$

Q-3: Segment Address = ??
is offset Address = 0800H. 007FFH

⇒ Segment Register: 14 bits

(Code, Data, extra, stack) 21 bits

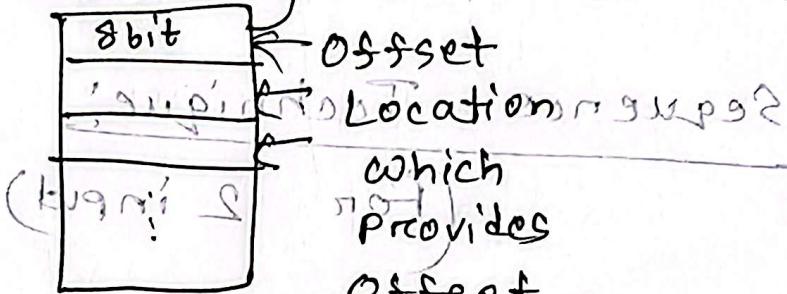
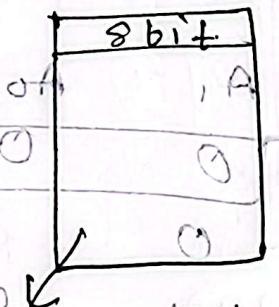
Physical Address = 000007



000877

segment precised Location

0 = Reg. # = SADS C) \rightarrow 16bba bba 0000



which provides offset address.

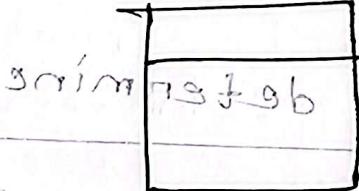
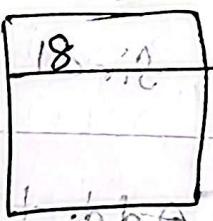
Assembly Code

generation for use of SI, DI, SS.

Lab

ES

DI, SS.



of give FF00H + 0800H

PA = segment + offset

$$PA = (\text{segment Address} \times 10) + \text{offset}$$

SA = 16bba from page 18-8

$$\Rightarrow 16^10$$

0A2 0800H from page 18-8

Q-2: If offset address is not defined

FF00H * 16bba location

$$+ F800$$

$$\underline{\underline{. 16bba 16bba}}$$

FF800H

Practice

Repeat

Problems

① Find memory size.

② Determine starting and final Add. of

memory.

+ Brey (Ref. book, Ch-10).

05

Nipa Madam

QG:

Write assemble language program

to read

1 Byte

data from

0800H

Ans:

(b6A print not DS(100L)C)

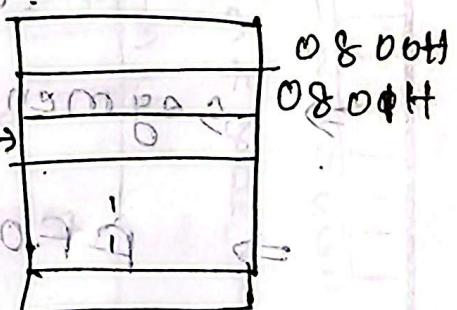
specific location

Offset address

steps:

① Segment definition

② Read operation.



Sat Sun Mon Tue Wed Thu Fri Date:

MOV,
By default Data Segment

PUSH, POP
By default Stack
is program Segment.

16-bit Long CODE pointer.
By default Code segment

Additional Segment registers

A STORE ~~20~~

Initial Address : FF800H

$[\text{HOPA}]=\text{segment Add} * 10 + \text{offset Add.}$

$\Rightarrow \text{PA} = \text{FF } 800H \Rightarrow \text{segment Add} = \text{FF00H}$
(Starting Add)

$\Rightarrow \boxed{\text{segment Add}} : \boxed{\text{offset Add}}$
 $\Rightarrow \boxed{\text{FF00H}} : \boxed{0800H}$

+ offset add from fig 2

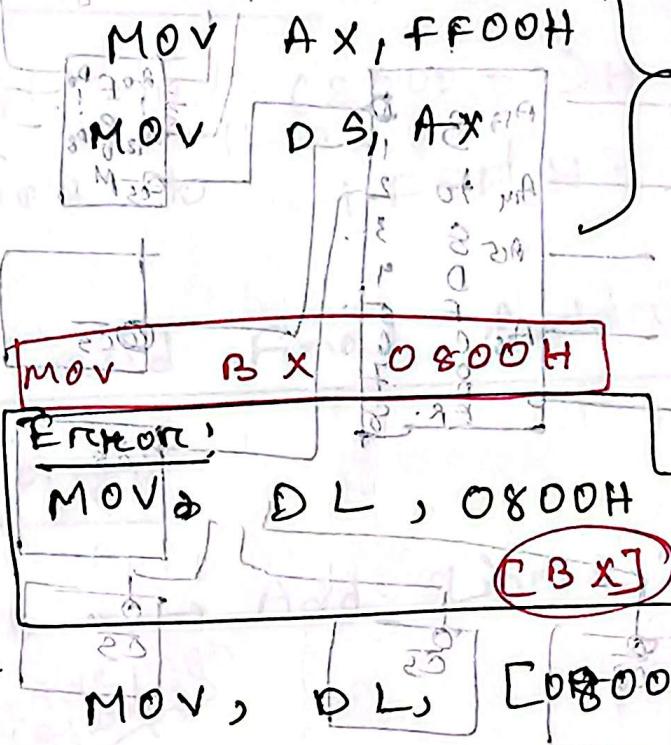
writing to bus

sat sun mon tue wed thu fri Date:

CODE :

DS
segment
Definition

Registers
Addressing
for 0800H
allocation
Read
Operation.



Note:

DL → To copy 1 Byte Data.

BX → To copy 2 Byte Data.

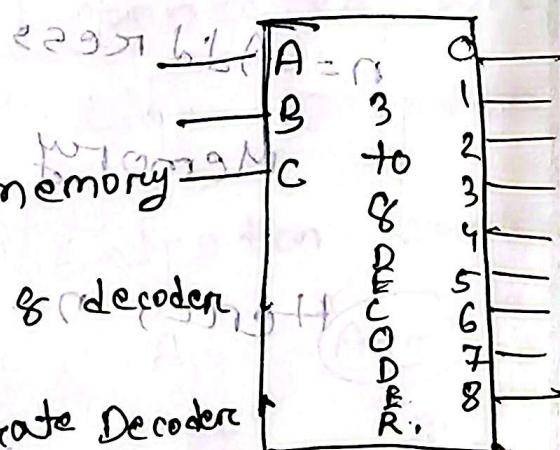


(MORRIS)

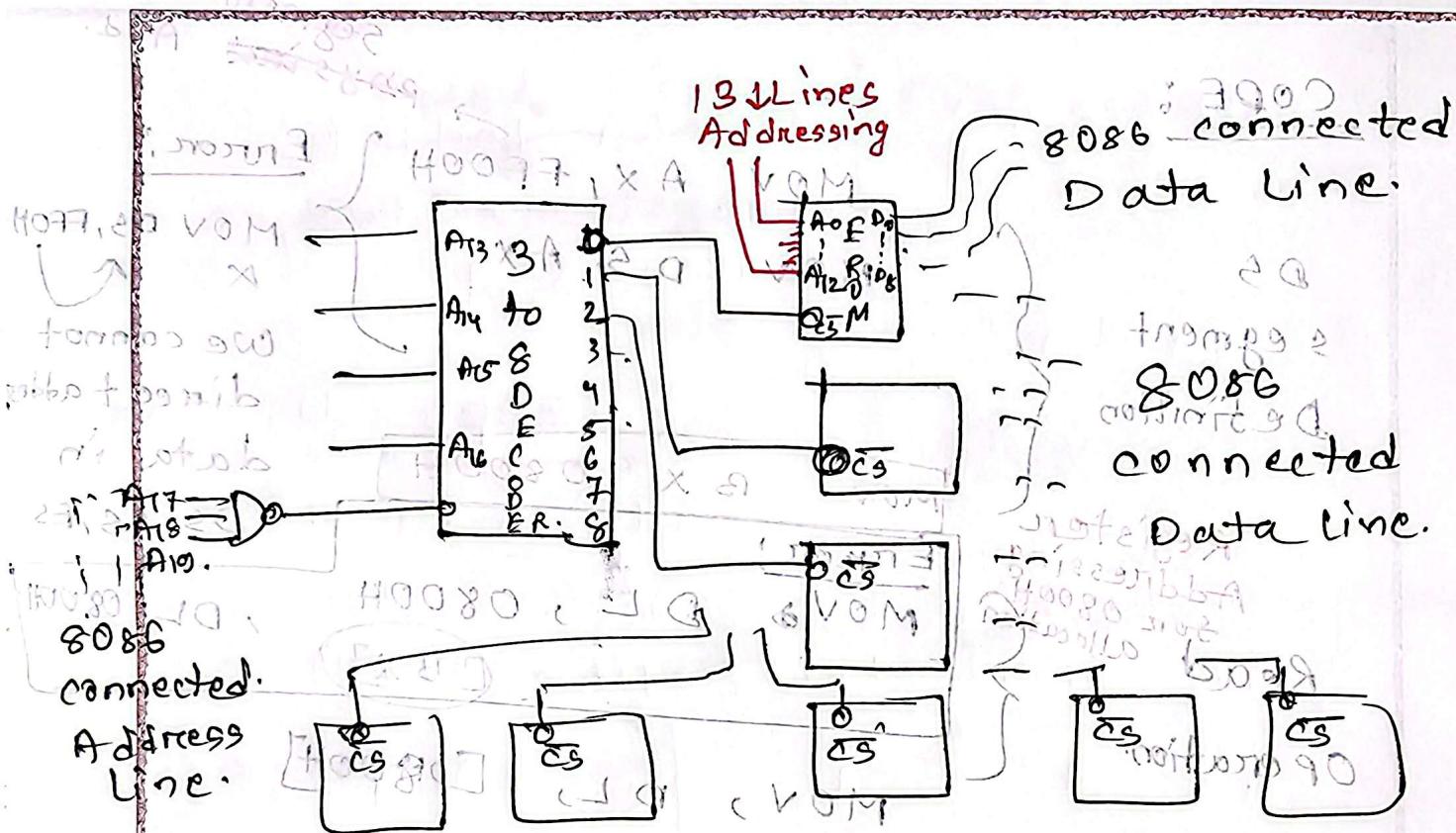
To work with 8 memory

we can use 3 to 8 decoder

instead of NAND Gate Decoder



sat sun mon tue wed thu fri Date:



Q10 Find memory size?

→ Memory size: $\leftarrow 2^n \times q$

n = Address Line connected to

Memory (i.e. EEPROM).

Here, $n = 13$

∴ Memory size $\approx 2^{13}$

$$= (8192)_{10}$$

$$= 2000H$$

Date: 03/11/2015

Tracing Data Range!

0000H to (2000-1)H

$\Rightarrow 0000H \rightarrow 1FFFH$.

Q2: Initial and final Address \leftarrow 0^{th}

2nd Memory?

⇒ In 8086 → Add. Line : 120.

Active High
→ CS = 1

$A_{19} A_{18} A_{17} A_{15} A_{14} A_{13} A_{12} A_{11} A_{10} A_9 A_8 A_7 A_6 A_5 A_4 A_3 A_2 A_1 A_0$

→ Second Memory is connected with
1 pin of 3 to 8 Line Decoder.

→ So, 3 to 8 Line Selection Combination

01 A 000 ← A13
LSB

At 4

A15

→ For 2nd Memory: A9 A15 An A13
② ① ↓

→ Initial Address for 2nd Memory

HCl + 000 S) of 400 mg

F 20004

书 4771 of 40000

Final Address for 2nd Memory:

• 9 BIR 19m 0016A LIII LIII LIII
F 3 BIR 1808 F F

F

3

F

3

$$= F_3 F F F H$$

Q3: Write assembly language program

to be read) 2B data into DX from memory

20024

2 B

data into

Dx

Strom

so it's done \Rightarrow right to fetch data from 2002H.

४५

N/A

S. A. S.

F 000 * 10

144

三

10. A

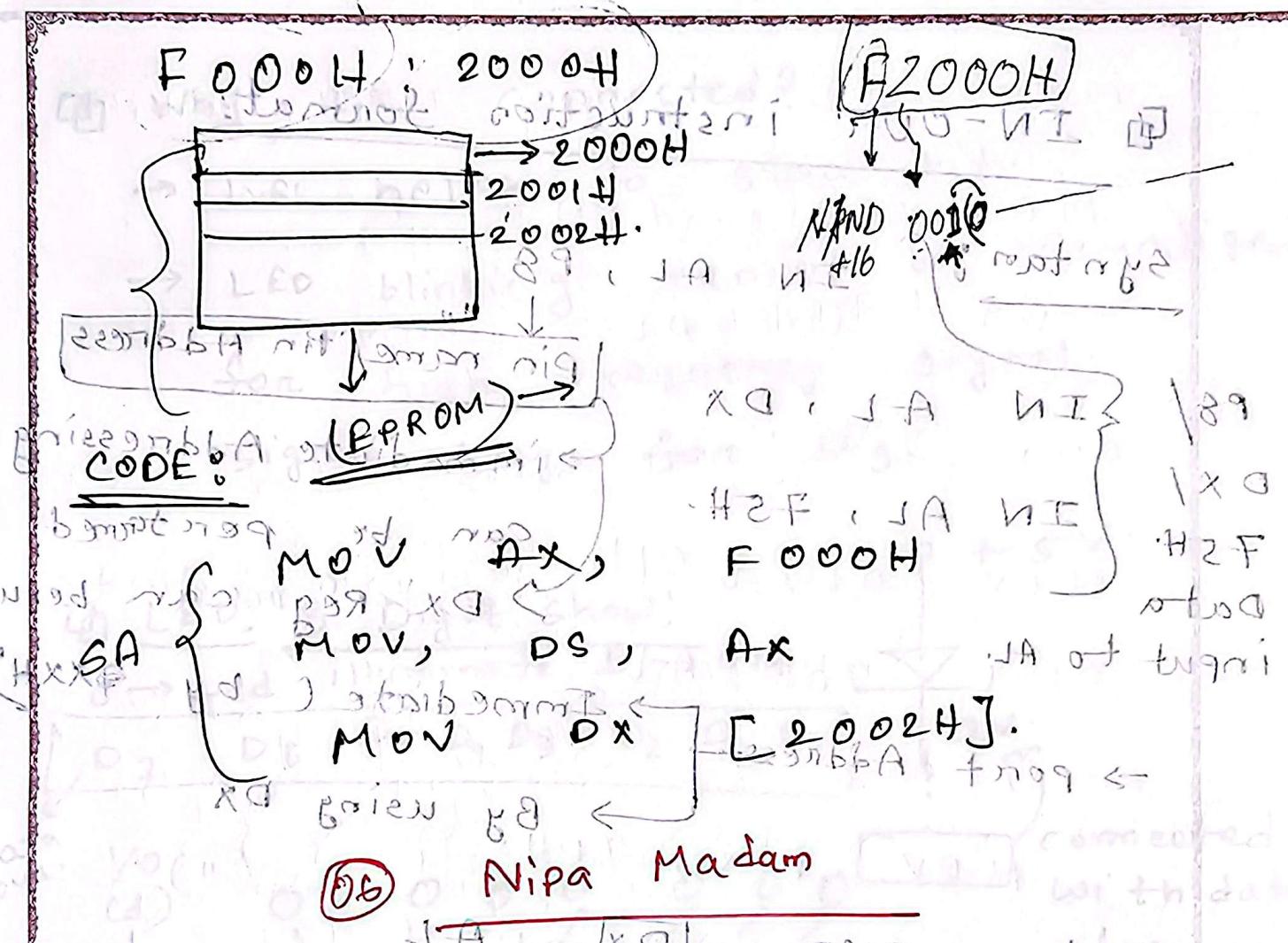
+0. A: 2002 H ← from question

۱۰۸

三

PA: F 2002 H. 579 ←

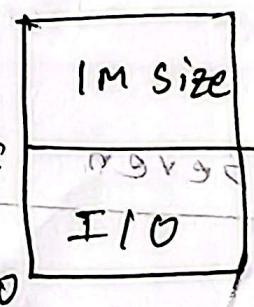
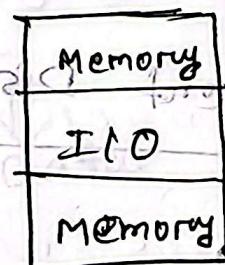
Date: 03/11/2025



Ch 10 Interfacing

→ I/O : Printer, mouse, key board.

→ Two Types:



→ Memory Mapped
book of
I/O

→ Uses MOV instruction

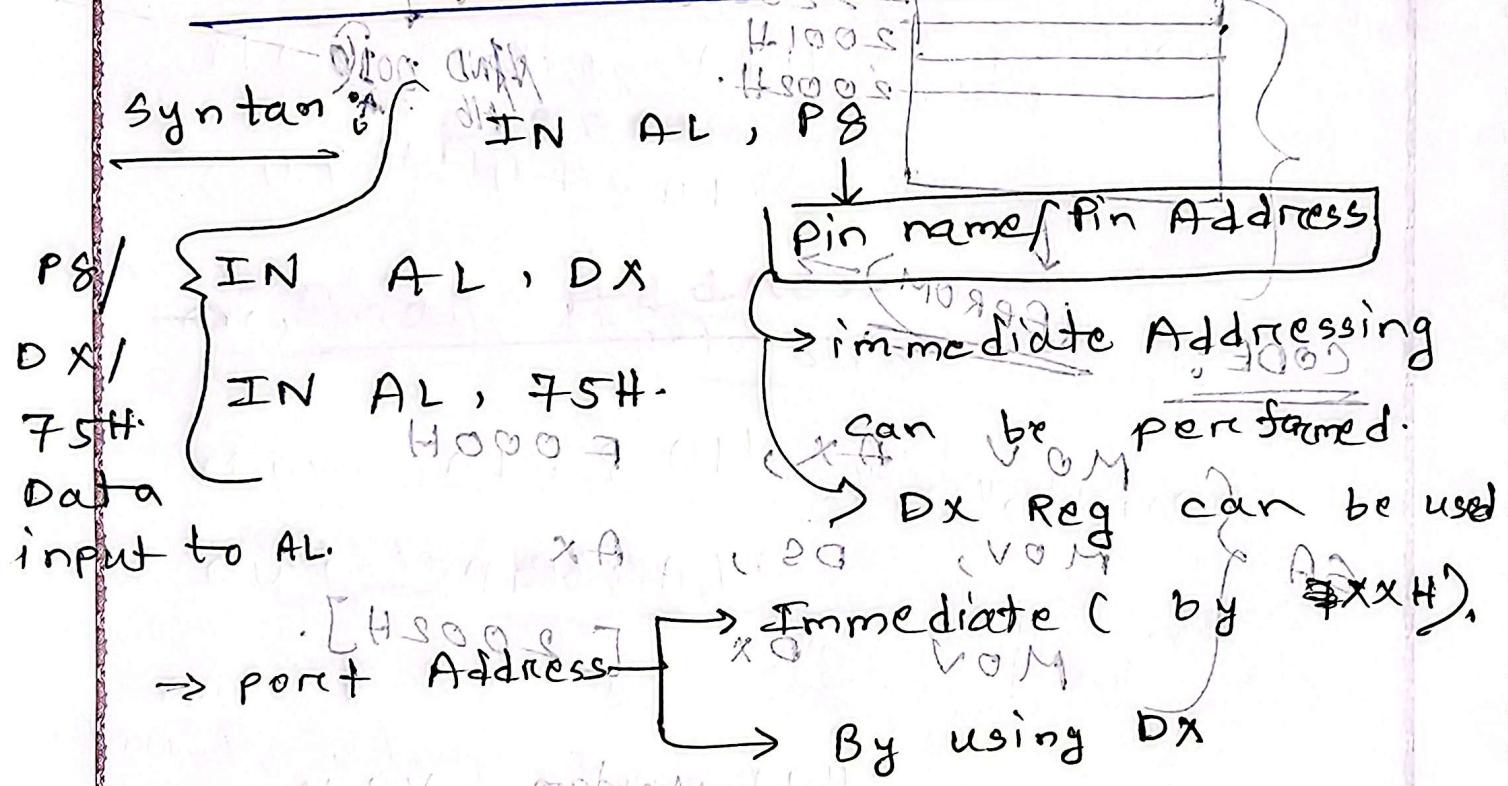
→ Isolated ± 10

\rightarrow T N OUT

— 4 —

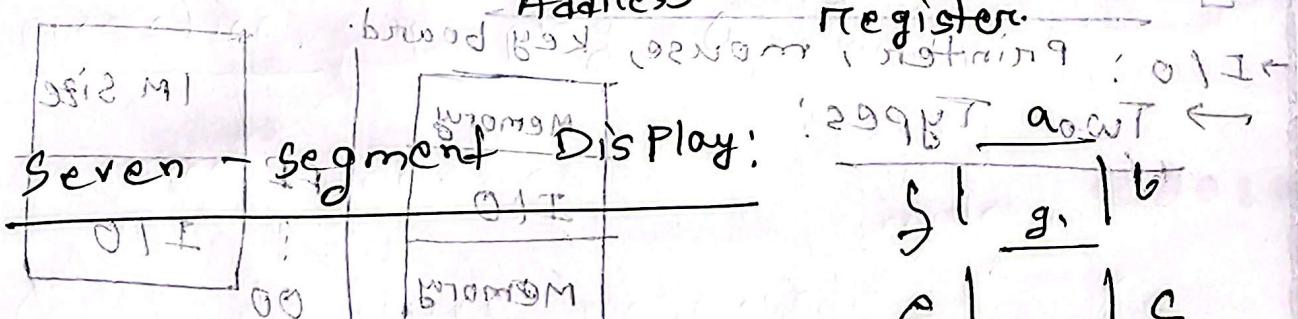
instruction

IN - OUT instruction format:



and, OUT DX, AL

Port Content
Address register



Why B to 8 decoder connected??

V_O, N_I ← to 8 decoder connected??

to 8 decoder connected??

V_O ← 8 decoder connected??

Q Why LEDs connected? a VOM

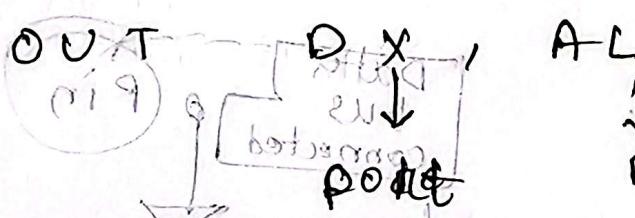
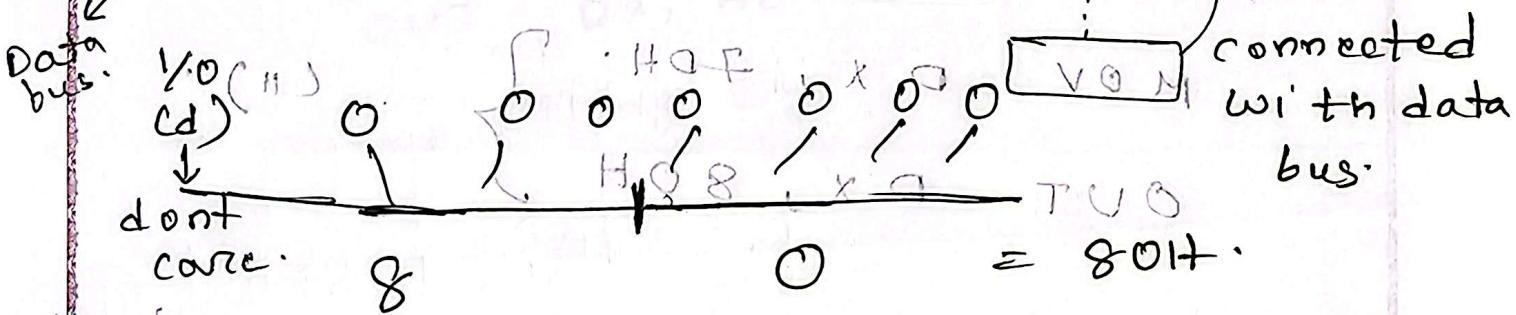
→ LEDs help to store bit.

→ LED blinking cannot be visualized for high frequency signal.
Signal stays for ms.

Q LED 8 digit show:

$0 \rightarrow \text{Led illuminate } 1 \rightarrow 0 \text{ FF}$

D₇ D₆ D₅ D₄ D₃ D₂ D₁ D₀



Activation
Address over 9 bits long 0.5
(last 9 bits 1 bit)

A₉ A₈ A₇ A₆ A₅ A₄ A₃ A₂ A₁ A₀
0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 1
= 7DH

Sat Sun Mon Tue Wed Thu Fri Date:

MOV AL, 80H → Data. Fwd ↗

MOV DX, 7DH → Address. ↙

OUT DX, AL → LED Display ↙

OR, ↗ not equal logic not

MOV AL, 80H { word } Step reduction ↗

OUT 7DH, AL → LED Display ↗

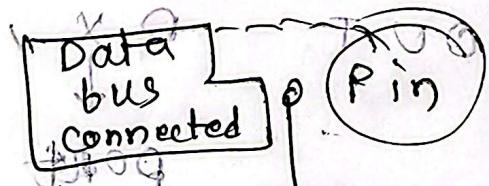
or, ↗ [D6 D5 D4 D3 D2 D1 D0] ↗

MOV DX, 7DH ↗

OUT DX, 80H ↗

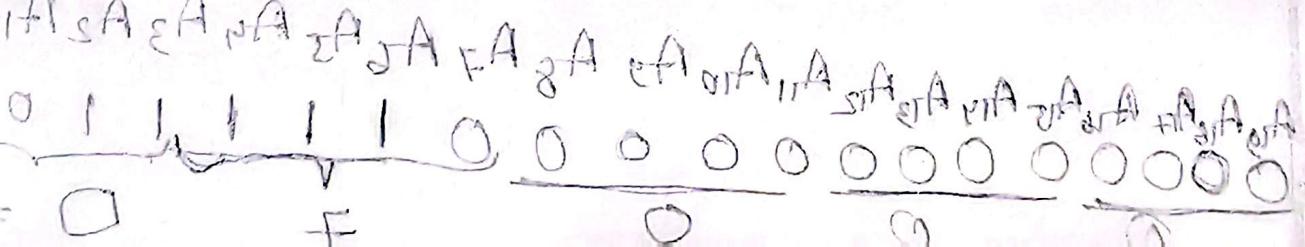
practice: ① Display 0 to 7, 9 in the

same manner.



② Led type reversed
(Anode/Cathode type).

+5V.



③ Switch type
(input type device)

Ref Book : Braunhart
(Ch - M/I/O interface)

Question:

P-A

add = ? ? ?

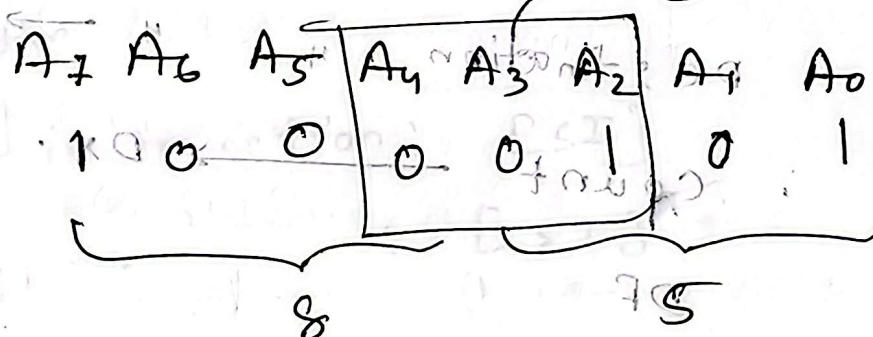
P-B

add = ? ? ?

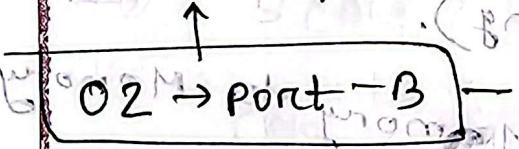
3 to 8 Decoder
reduces

Address
mismatch.

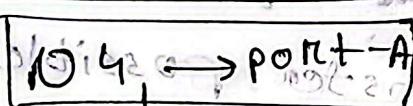
Ans;



So, it is input port.



RD WR I/O/M



RD WR I/O/M

Memory
(Active Low)

So, if read will be performed.
So, it is Read port.

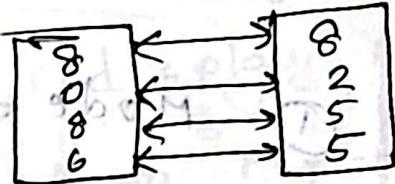
W2 V0 M

R2 V0 M

08

Nipasbo Madam

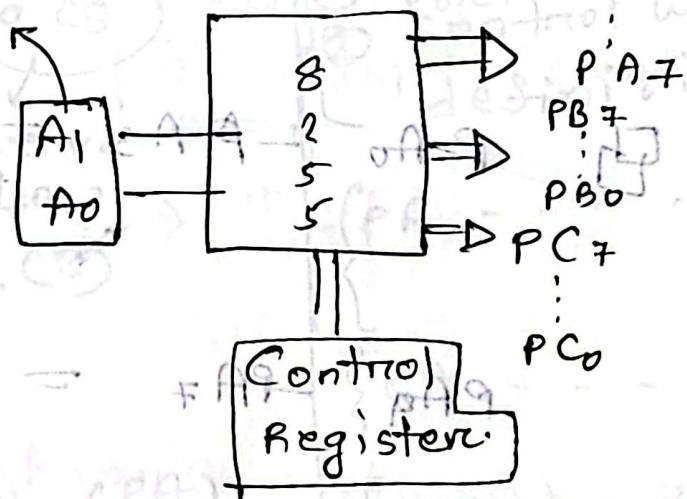
IC: 8255 PPI



\Rightarrow PPI: Programmable Peripheral Interface

Used to interface Port A? Port B?
Port C?

A₁ A₀ & Control



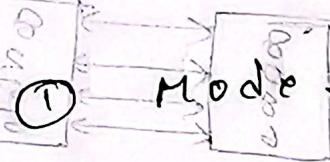
control bar defines what is the purpose of Port A/B/C

Elements: LED, 7 Segment display,

Stepper Motor, Mouse, Key board
can be operated from this Ports

Sat Sun Mon Tue Wed Thu Fri Date:

8255 PPI Modes:

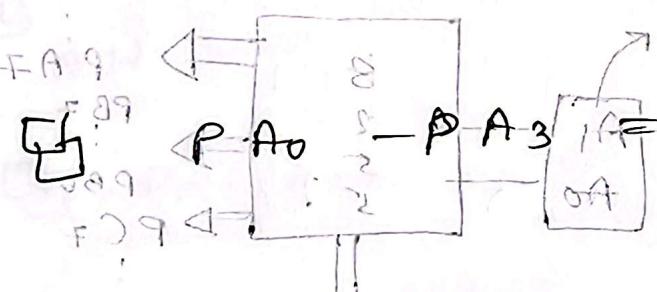


① Mode → 0 (I/O device connected).

L99 Handshake Bit set reset

② Mode → 1 [BSR mode, mode]

③ Mode → 2 [Bidirectional mode PC - PG].



Upper Lower port of
Port - A.

(PA0), OA, A

PA7 = PA0 (upper port of
port - A).

same as PA7 = PA0 (upper port of
port - A).

MEMR Memory map Input / Output
MEMW

Input / Output from F (0E) & E (0F)

Input / Output from M (0D) & N (0E)

Input / Output from S (0C) & T (0D)

Q) Write an assembly language program

INT TWO = A9

+ to read

INT TWO = PA.9

display it

Soln:



Since:

Port B and display it
and to read from Port C and
to SPCU.

Port Address
Port define.

Steps:
① Port definition
② Control Word
definition

CS

Active

Low

(0)

Activated

A₁ A₀

① ② = 8000H (CPA)

0 0 = 8001H (PB)

0 1 = 8002H (PC)

{ PA - - -

{ PB - - -

{ PC - - -

control region

Other address pins (A₂ → A₅) →

Dont care

Sat Sun Mon Tue Wed Thu Fri Date:

Control Word:

D₂ D₆ D₅ D₄ D₃ D₂ D₁ D₀

1 0 0 0 0 0 0 1 = 83H

simple
output / input

selection

Mode = 0

II

20

(0)

start IA

PA = output
port.

P.B₉ = input
port

PC_L = input
section

PCU = upper
section

EQU Directive:

EQU → (49) PPIA (49) 8000H EQU 8000H

PPIA Address eff. = 3.1 Address define

3.1 → 8000H

Similar Look like

#Define PI 3.1K

এখন কোনো PI =
3.1K

Code: $\text{JMP } \text{PortB}$

PPI A EQU 8000H

PPI B EQU 8001H

PPI C EQU 8002H

PPI C EQU 8003H

Port define

define control register port

CW → { MOV AL, 83H }

CR → OUT [PPI C], AL

Same as for all

PPI Device

PBread { IN AL, PPI B }

→ first & data

Input PortB and
Read from Port B and
Display it in Port A.

PA write

OUT [PPI A], AL

first Address
of Port A

Then Data

PortA = I/O

PB = I/O

PortB = I/O

PortC = I/O

IN AL, PPI C

SHL AL, 4

OUT [PPI C], AL

Output at PortC but
initially ...

Read from P_{CO-PI}
 $(C_7 C_6 C_5 C_4 C_3 C_2 C_1)$
 $\overbrace{X \quad X \quad X} = X$
 B 7 A I 99

X X X X
S H L A L, 4

Transfer 'x' from
 $P(C_0) \rightarrow P(B_3)$ to C_6
section.

1) Nipa Madam

09

(Registers)

content

Bit Select

বেগন
পিন
set/reset
৩ টেক্সট

option: $D_7 D_6 D_5 D_4 \overbrace{D_3 D_2 D_1}^{D_0}$

$$S/R = 1/0$$

$$q_1 d = AP_A + 1PB$$

$$g/1 = 39$$

• 81) 8 write

1199

a BSR (set ^{and} reset mode).

~~PC₇~~ and

$\text{PCB} \xrightarrow[\text{JA}]{\text{SII99}} \text{forms wide lay set}$
 JA, SII99 neglect

code:

① Port Address Define:

$\text{HFO} = 0\ 1\ 1\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0 \rightarrow PA = 80H$
 $\text{HDO} = 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0 \rightarrow PB = 81H$
 $\text{HD} = 0\ 1\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0 \rightarrow PC = 82H$
 $\text{CR} = 0\ 1\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0 \rightarrow CR = 83H$

[If only $PA = 8000H$, no ckt given,

then we will determine address by

+ increment]

② C.W: For $PC = 80H$ set $[PC + CS]$

$D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0$

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

B SR
Mode
Set

$HFO \rightarrow A$ $FU \rightarrow M$
 $HDO \rightarrow A$ VOM
 $JA \rightarrow T$ TUO

Similarly,

9363

3. 39769 23798 A 709

$$\begin{matrix} D_7 & D_6 & D_5 & D_4 & D_3 & D_2 & D_1 & D_0 \\ P C_7(s) & 0 & A & 0 & A & 0 & A & 0 \\ P C_3(s) & 0 & 0 & 0 & 0 & A & A & A \end{matrix} = 074.$$

$$PC_1(R) \otimes \mathbb{Z}^n = 0$$

$$P(C_3(R)) = 0.00000110 = 0.64$$

This is the control Bar condition for

Code:

PPJA EOU 80H Engrossed

PPE B EQU F 81 H

PPI C. FOM 1824.

PPI (P) Eos u, 83H

PCF
Set

A small rectangular logo for "Floral products" with "a concern of BEARIS".

{ am }

OUT

M PV

OUT

~~A1~~ OFF

$\text{Al}_2 \cdot \text{OFH}$

PPIC, AL

Because processor gives instruction

1929
W.C. M.
P.C.P.

PC₃ set { tMOV AL, 0FH. event not
fringe 23 } OUT PPI C, AL

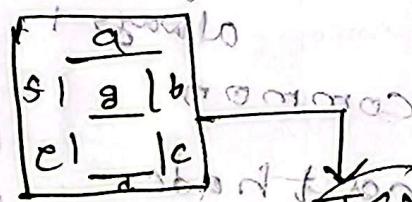
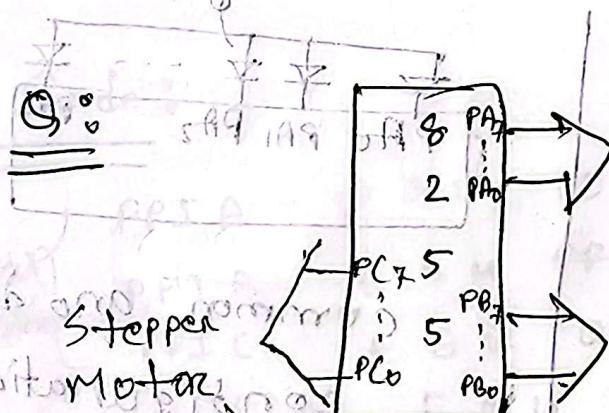
DELAT → operated by timer IC.
Gone lines will be included

PC₇ Reset { MOV AL, 0EH }

OUT PPI C AL

PC₉ R off { MOV AL, 06H }

OUT PPI C, AL



Key board (off) → (off)

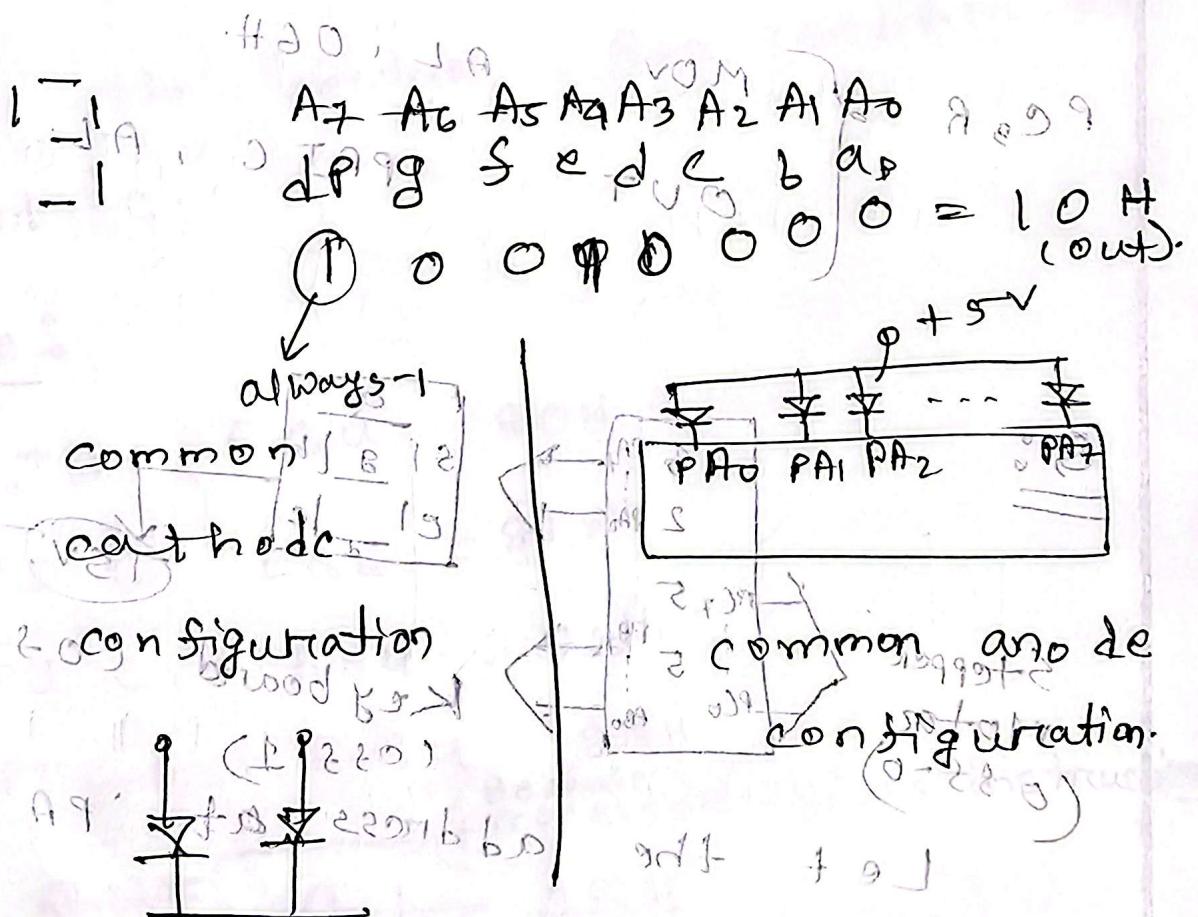
Let the address at PA = 8000H.

Write instructions to display digit 9 using 7 segment display

You have to turn off the applications at PB and I_{PIC} TWO for efficient

I want to use microcontroller because it has better memory and faster processing.

Ans: The 7 segment display is connected with port A; so, we deal with PA only.



first program of environment of PIC

program frame for brief

Sat Sun Mon Tue Wed Thu Fri Date:

$$PA = 16H \text{ (Output)}$$

$$PB = FFH \text{ (Input)}$$

$$PC = 00H \text{ (Output)}$$

HO O C A VOM

Control bar (CW) T U O

HO D₇ D₆ D₅ D₄ D₃ D₂ D₁ D₀

T A (A I O) 0 0 0 0 0 0 1 0 = 82H

Code

PPI A	EQU	10H
PPI B	EQU	FFH
PPI C	EQU	00H

PPI C-C EQU 82H

CW of MOV AL, 82H

OUT PPI C-C, AL

PB off } MOV AL, FFH } A9

OUT PPIB_H 4 AC = A9

Though keyboard is input device.
But we hold keyboard function
by processor

PC off } MOV AL, 00H

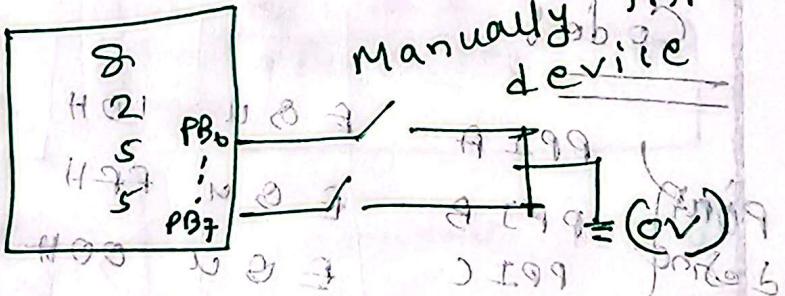
OUT PPIC_(W), AL

Output port position

PA → display } MOV AL, 010H

OUT PPIA, AL

For this



⇒ If we add manually input

switch, then J1 → I99 two } w

Program F
PB off {
PMOV 90L FFH,
IN }
emitter logic
**
PPIB, AL

With manually input 1 to processor via
register

no switch

latch nos

110 : 2295660

HAI, WEA-A I99

Nipa Madam

HDA

3 TO DA (IO)

910

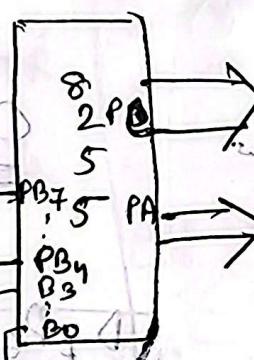
0 = 09

0 = 19

□

Stepper
Motor

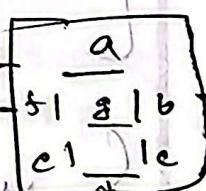
035:1



H08 D/A converter (On) → 1

7 segment
display

H77 DA VOM



+5V

Write

A. L.PA to display digits

⇒ +5V → 0V → ON So, to keep 0 for
B = 1, D = 1

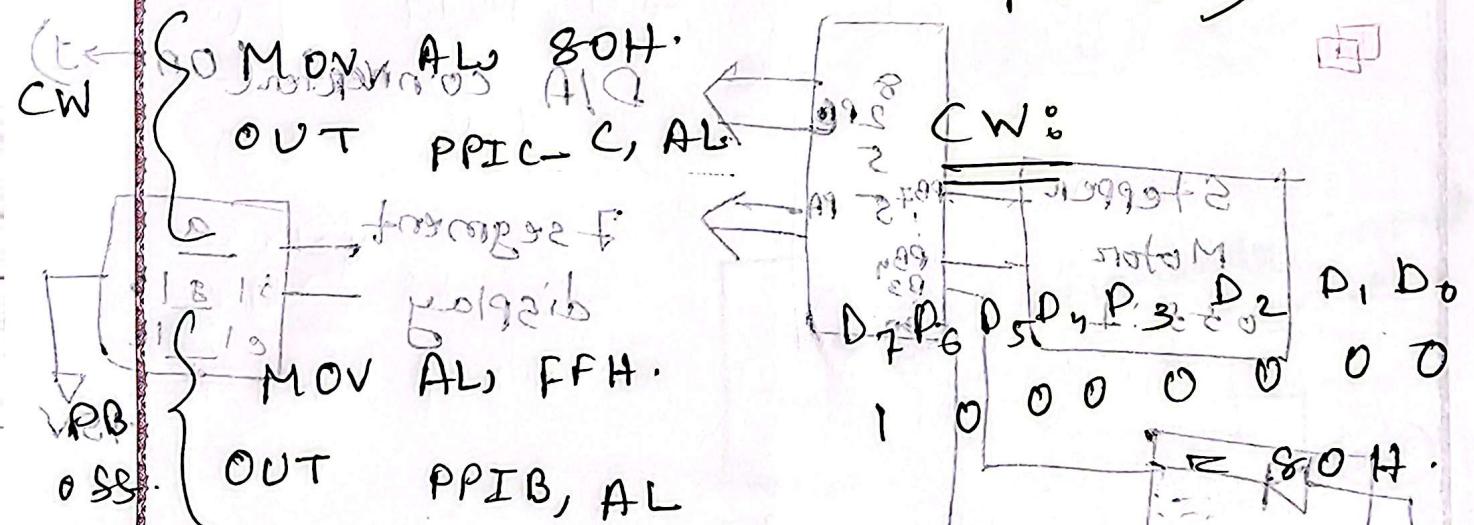
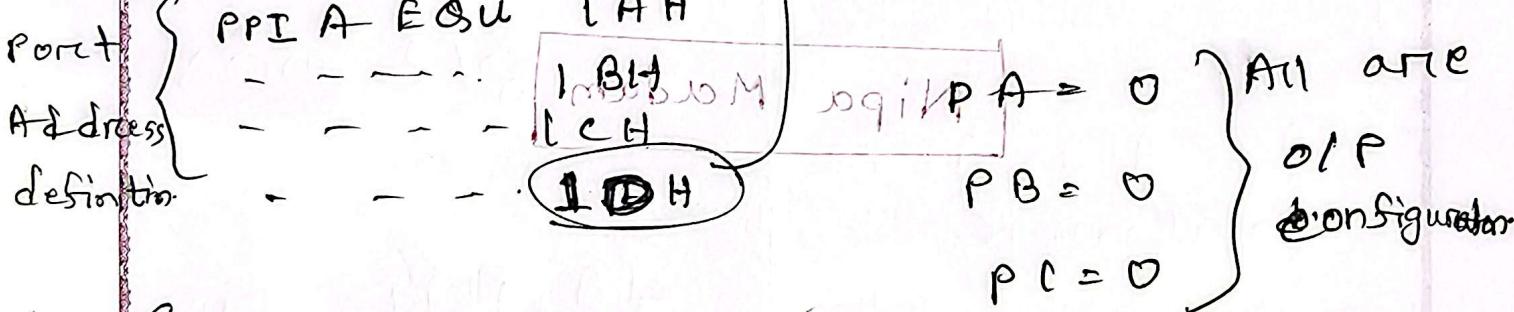
0 → a in a loop using 7 segment
 ↓ A, B I99
 display. Turns off
 other applications.

All necessary of Engg. P. No. 10
 Let address of PA = 1AH

Code

control register

address: 1DH



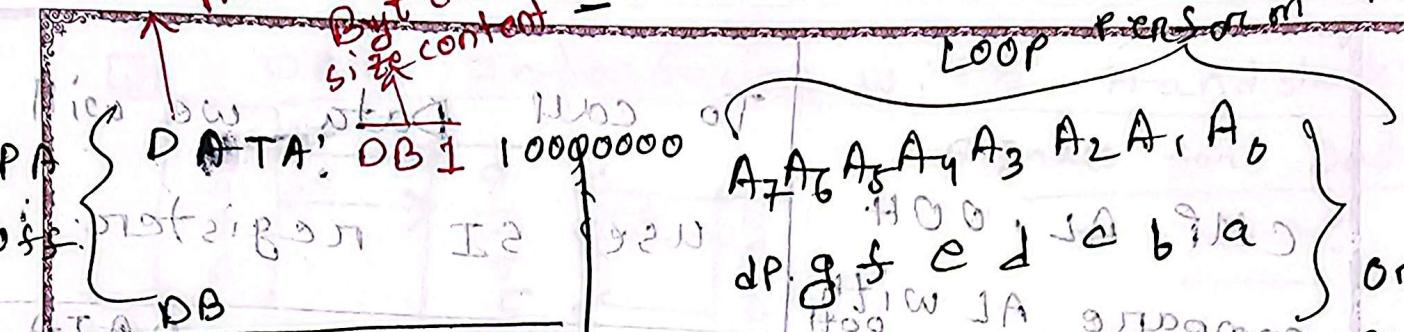
PC { MOV AL, 00H.

Offset: OUT PPI Conf AL. A

B₇ B₆ B₅ B₄ B₃ B₂ B₁ B₀
 1 1 1 1 1 1 1 1

start CW

BW → Word sized data
 Data level notation.



DATA offset 1000H
 DATA offset 1001H
~~DATA offset 1002H~~

L10 MOV SI OFFSET

DATA ← AT9, DATA

avif garrish

MOV AL, BYTE PTR

CS:[SI]

Ref compared value → A1000H stored in code
 segment memory

AL = 10000000
 00000000
 NZFZ = 0

Code segment

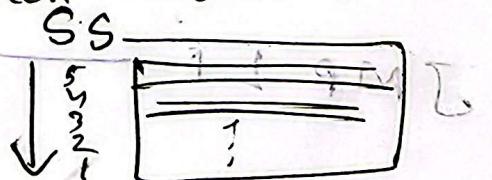
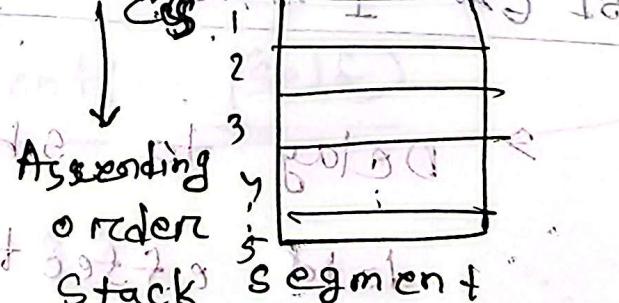
CMP AL, 00H

JB L2

OUTPPIA, AL

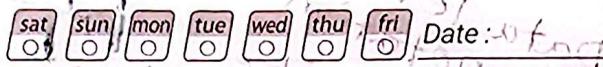
INC ST

JMB L1.



Descending Order

date 220 3 bed &  10005 655-12 10005-12
10005 655-12 10005-12



$\int_A A_1 A_2 \in A \cap A_1 A_2$
CMP AL, 00H
compare AL with 00H

To call Data we will
use SI register.

Jump condition
to L2 after
! checking with
zero flag!

MOV SI, OFFSET DATA

COA TIA Address copy
RECEIVED 12 NOV 1967
TO S.I.A. VOMBAL
U.S. AIR FORCE

~~MOV AL, BY TF P TR CS%~~
~~[SI]~~

BYTE PTR → Assembler
directive.

To copy data, SI ~~segment~~
RTF ETR, JA ~~VOM~~ from
To & Code segment

OUT PPIA, AL
B57072 8.6.
A Lengot J
1

delay
INC (SI) 832 360)

SI 670 । राष्ट्रिय +

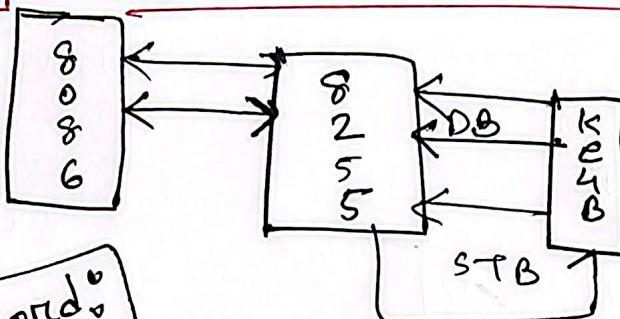
Show the next data
100, 101, 102

→ Delay to show the change of
data effectively.

Diagram illustrating the stack frame for function f . The stack grows downwards. The frame contains local variables l_1 , l_2 , and l_3 , and parameters p_1 and p_2 .

Jump to 21. ~~DE100~~

DATA Interfacing with Handshake



signals for Data
input?

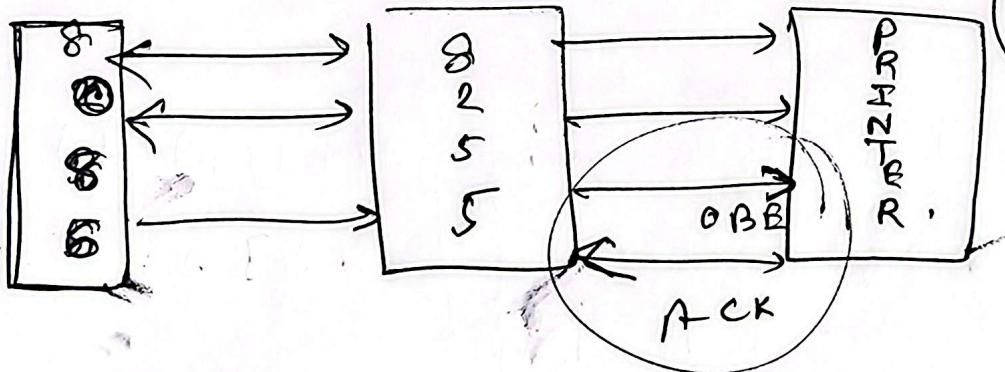
Keyboard

Keyboard
sends a
STrobe

communicate
Input (STB)

IBF

Input Buffer full (IBF)



* *
Output
handshake

Problem Statement: (self)

Mode - 1? 2? 3?