

① Nipa Madam

ICs 8086, 8284, 8255, 8254, 8259.

8086 Microprocessor

→ 16 bit
→ Register size: 16 bit → 16 bits can be stored

→ One data $11H$

$0000\ 0000\ 0000\ 0001$ b.
0 0 1 1 H

→ Usage: ① Data transfer ② program flow in a simple instruction. ③ Arithmetic and logical operations

→ 8086 \xrightarrow{WR} ; 8086 \xleftarrow{RD}

→ Bus type:

① Data bus

* which data can be transferred

(16 Data bus)

* (D)

* A₀ - A₁₅

multiplexed
Address and data pin

② Address Bus

Address hold / define

* (A)

* D₀ - D₁₅

AD₀ - AD₁₅

Address and data pin combined.

③ Control bus
Data transfer location.

[sat] [sun] [mon] [tue] [wed] [thu] [fr]

Date

Ⓐ $A_{16} - A_{19} \rightarrow 4$ bit \rightarrow Higher bus

Address bus pin \rightarrow Non multiplexed.

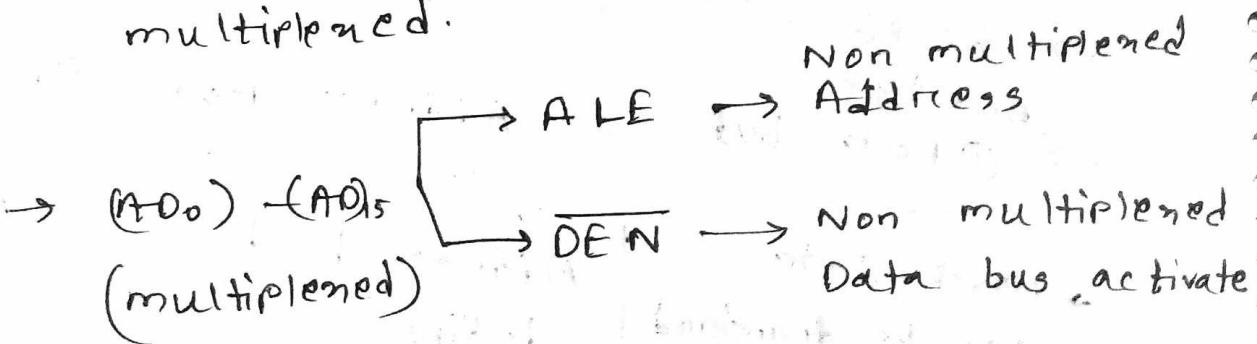
Ⓑ Data bus use of multiplexed Pin:

\rightarrow ALE (Address Latch, Enable)

$\rightarrow (AD)_0 - (AD)_{15} : A_0 - A_{15}$ non multiplexed
multiplexed usage.

$\rightarrow \overline{DEN}$ (Data enable N.)

$(AD)_0 - (AD)_{15} : D_0 - D_{15}$ Non multiplexed
multiplexed.



Ⓒ 8086 \xrightarrow{WR} M / \overline{IO} (Input / Output)
(Memory)

8086 $\xleftarrow{RD.}$ D.T / \overline{R}

- control pins: \overline{RD} (Assigner specified).
 (Read)
- \overline{WR} (write Data);

Pins: $\overline{RD} =$

$\overline{WR} =$

M/I/O

DT (R)

ALE

\overline{DEN} → D₁₆ - D₁₉ (High, enable 4 bit
enable).

\overline{BHE}

- other pins:

CLK pin: clock enables

Reset pin: External to Microprocessor (\leftarrow).

INT#: → Interrupt Request.
 → Running function can be paused at any time.

INTA: → Interrupt Acknowledgement.

INTR ≥ 1 } Data is accepted and
 $\overline{INTA} = 0$ } acknowledged.

④ Hold } used to hold a
HLDA (Hold Accepted) program.

⑤ After
Interrupt / Hold \rightarrow Ready = 1

otherwise, Ready = 0.

⑥ Test: \rightarrow perform AND operation.

\rightarrow does not change the value of the resistor.

⑦ Mn / MX } Takes input from
 Min mode multiple IC's
 Max mode operation.
 operation.

Low speed \downarrow Max High speed.

One IC (other IC's help to perform its own task)
 performs its own calculation

⑧ Ref book:

① Ramesh Gaonkar, Microprocessor Architecture, programming and Application (6th Ed)

② Barry B. Baur, The intel Microprocessor Arch. & Programming

④ Rafiguzzaman → Microevolution theory

Time and applications

Microprocessor and Microcontroller

(P) Douglas F. Halls

Interface 5.

1

fahim sir

8086 assembly language programming

Q Ref book:

① Digital computer electronics - Malvino and Brown.

Microprocessor and Interfacing - Douglas V. Hall

③ Assembly Language for x86 Processors-

Kip R. Irvine

Microprocessor

(in internal storage)

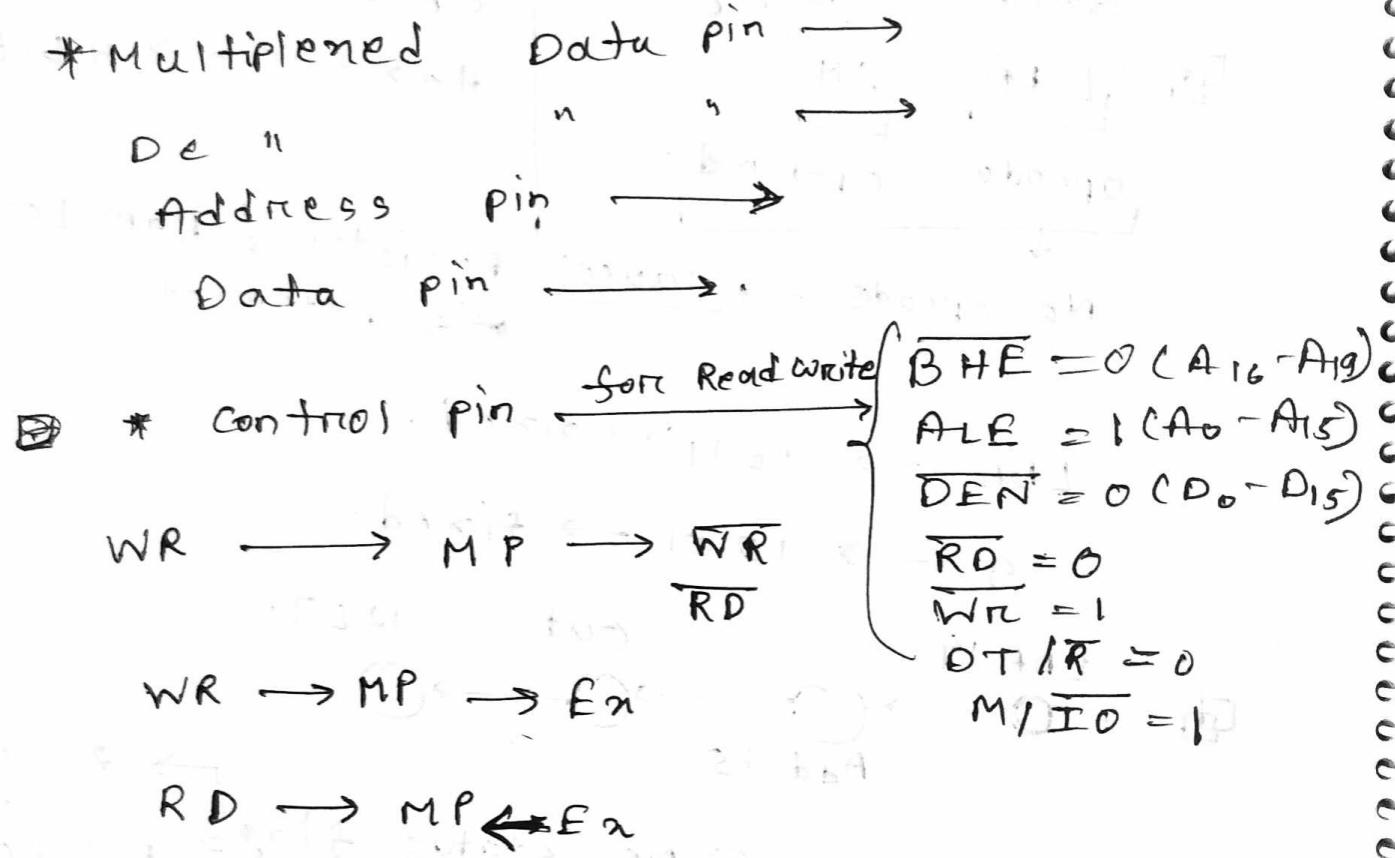
→ fetches instructions → decodes instructions
(from memory) executes ←

Executes instruction.

(2)

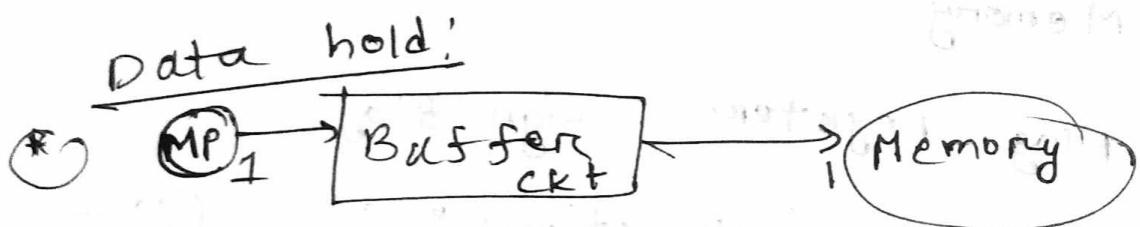
Nipa Madam

Bus Buffering and Latching:



* Data storing time: $t < 0.01 \text{ ms}$.

* control pin → Address pin, Data pin
 \overline{BHE} , ALE others.



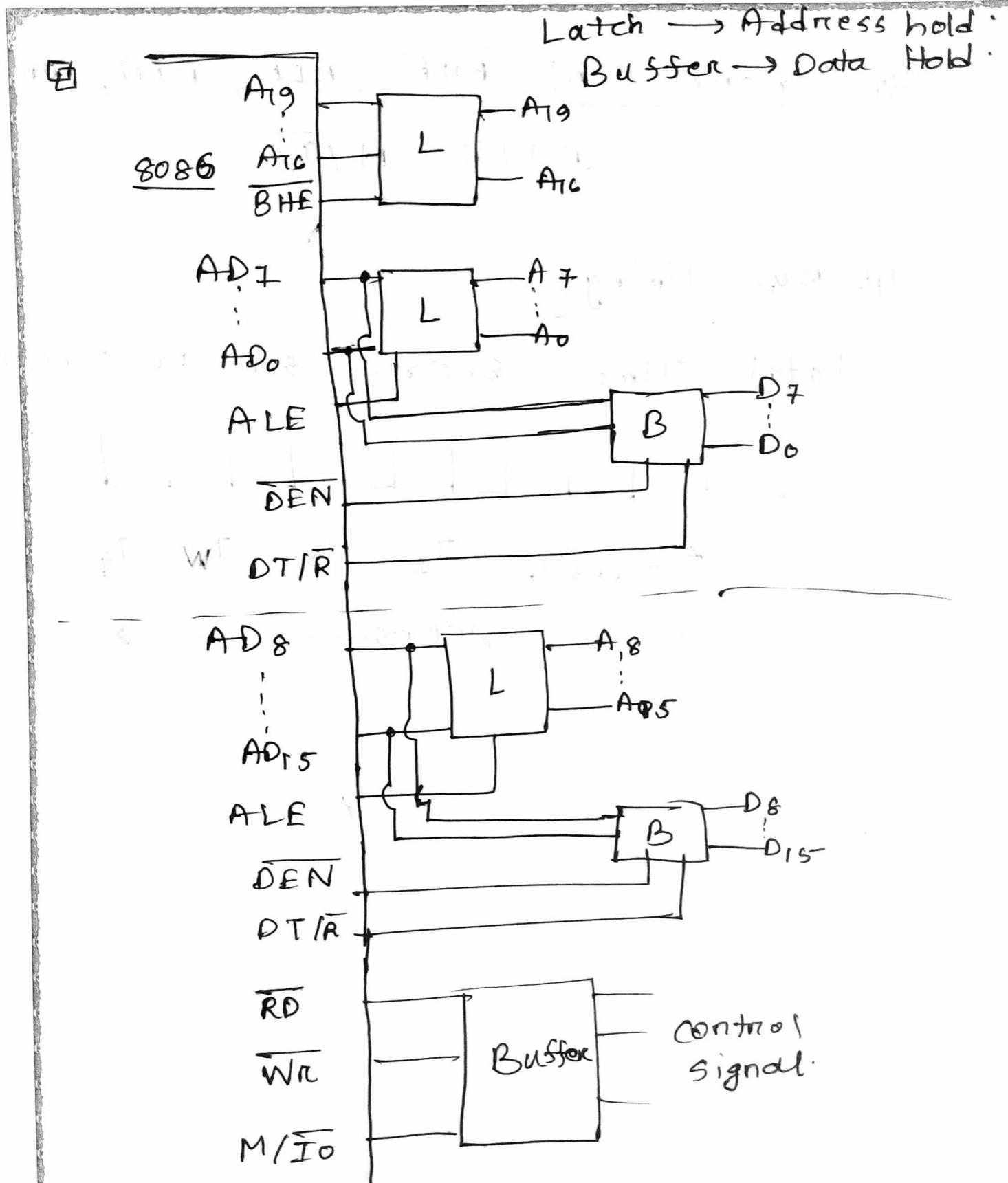


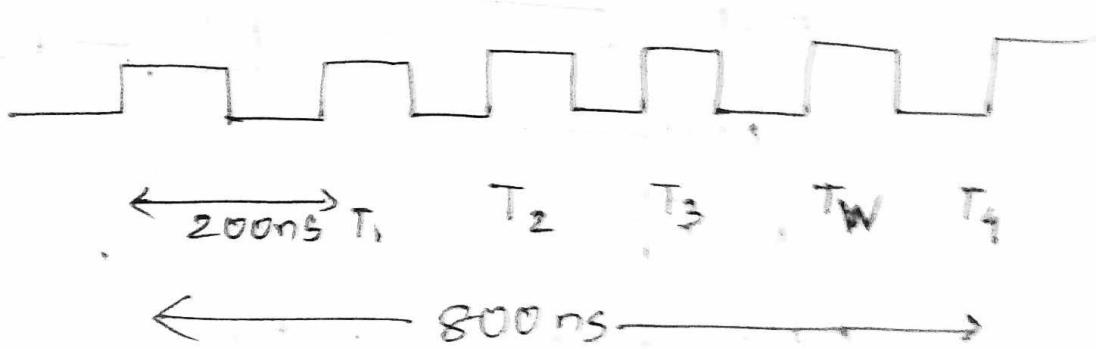
Fig: Schematic of Bus Buffering and Latching.

Control Pin! \overline{BHE} , ALE, \overline{DEN} , RD, \overline{WE}

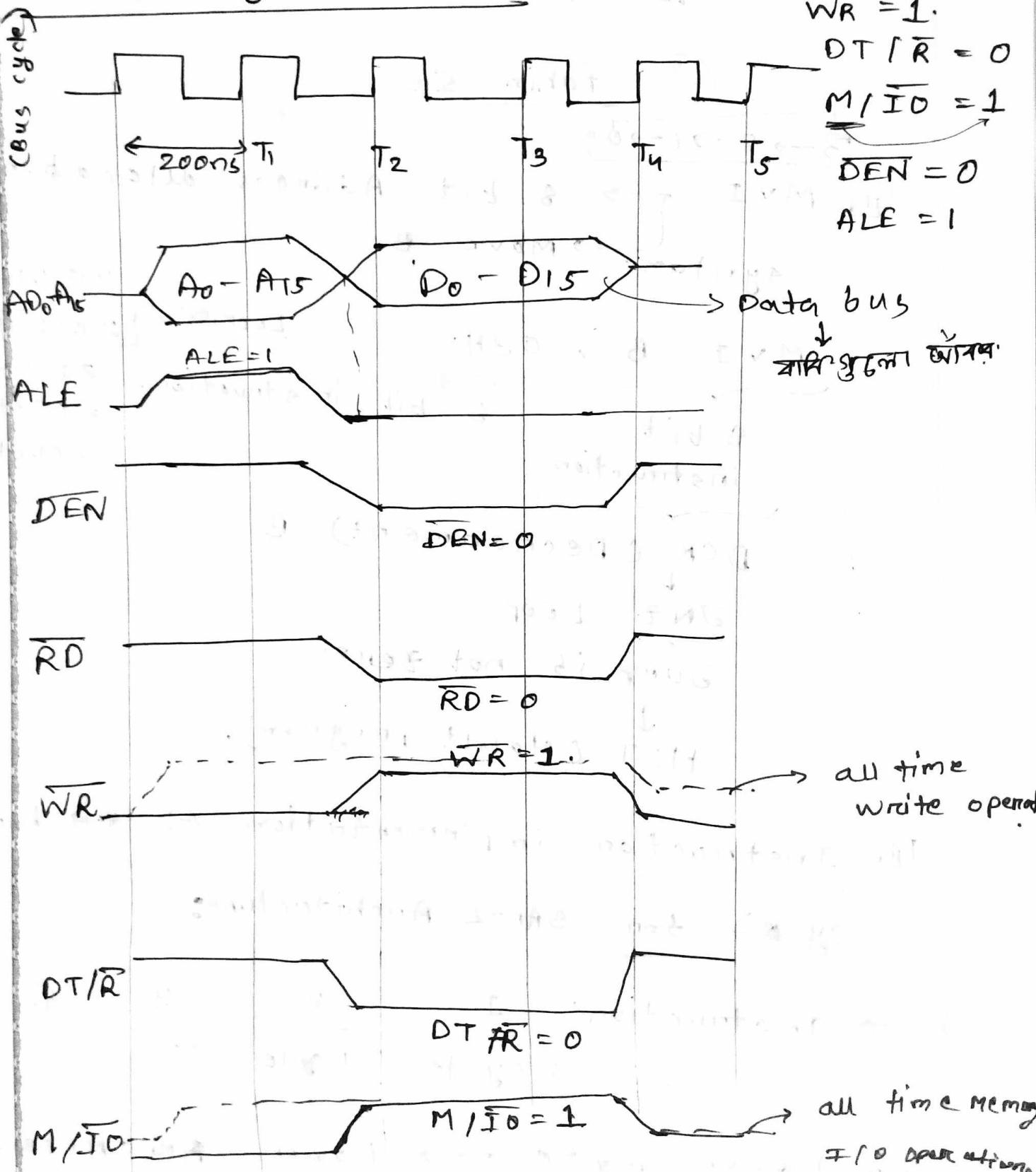
$\overline{DT/R}$, M $\overline{I/O}$

Bus Timing!

Total Time = 800 ns for IC : 8085



Bus timing Diagram: For Read:



* For Write operation \rightarrow HLT.

(03)

fahim sir:

$3 \rightarrow 2 \rightarrow 1 \rightarrow 0$

MVI \rightarrow 8 bit Address allocation

syntax: Move B.

MVI B, 03H

8 bit instruction

DCR (Decrement) B

JNT LOOP

Jump is not zero.

HLT (halt program).

2000H
Loop level + 2002H

2002H

2003H

2006H

□ Instruction implementation (\leftarrow machine cycle) for 8082 Architectures

\Rightarrow Instruction:

1
2 cycle

2
1 cycle

3

□ Lower Byte \rightarrow Lower Addressing Store.

Higher \rightarrow Higher Addressing

Instruction Set

STA 5600H } total 3 byte.
 2 byte of 2 byte word of 3 bytes.

SAP-3: → It has stack pointers (LIFO manner)
 → Push, Pop mechanism for execution.

03

Nipa Madam

Control pins

ALE ($A_0 - A_{15}$)

$\overline{DEN} (D_0 - D_{15}) = 1$

$\overline{RD} = 1$

$\overline{WR} = 0$

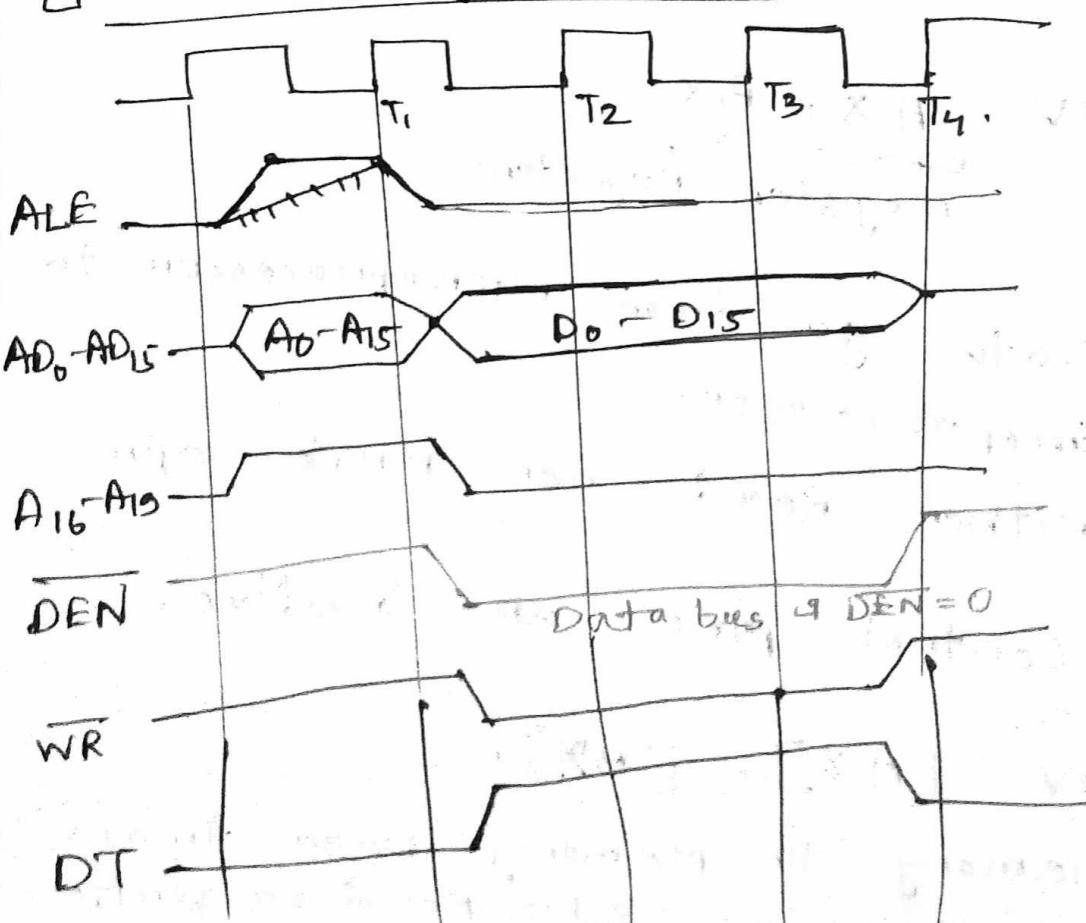
$M/I\overline{O} = 0$

$DT/R = 1$

BHE ($A_{16} - A_{19}$)

$= 0$

Write timing Diagram!



MOV A_n, [B_x]
Destination Source

B_x → Register.

[B_x] → Memory
location addressed
with B_x.

MOV A_n, [2002H]

→ Data go from memory to

microprocessor
(Read operation)

MOV [A_n], B_n
Memory Register

→ Data go from microprocessor to Register
(Write operation)

MOV A_X, B_X
Register Register

→ Data go from microprocessor to
Microprocessor.

→ neither Read nor Write op.

→ Control pins are inactive.

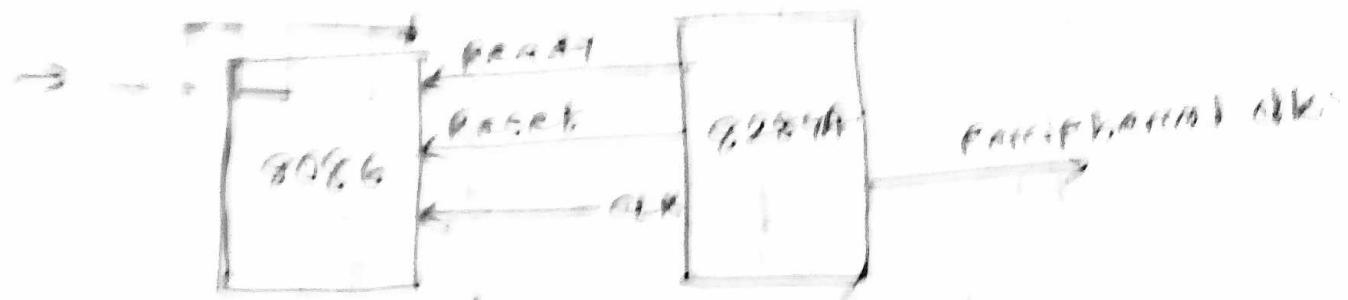
MOV [A_X], [B_X]

→ Memory to Memory Data transfer
→ No Read or Write

→ Control pins are 16 bits

④ clock generation (8086)

→ To supply clock to 8086 peripheral



→ 4 working: ① CLK generation

② ready signal

③ reset

④ peripheral

Developed by

counters and

T-SLIP flops

for clock

frequency division

→ 3 count counter required Alvidar ??

→ blocks: ① reset ② CLK ③ PCLK

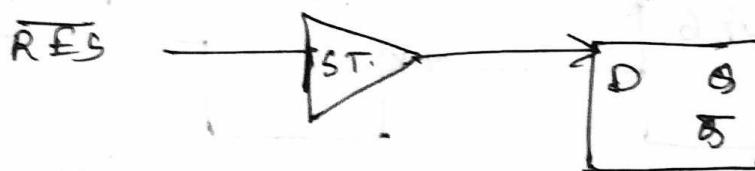
④ ready

① Reset Block:

↳ Smith Trigger

→ op/amp ckt: LT
(Level comparator).

→



D flip-flop is used to transfer data to S. smoothly.

② clk Block:

→ crystal oscillator

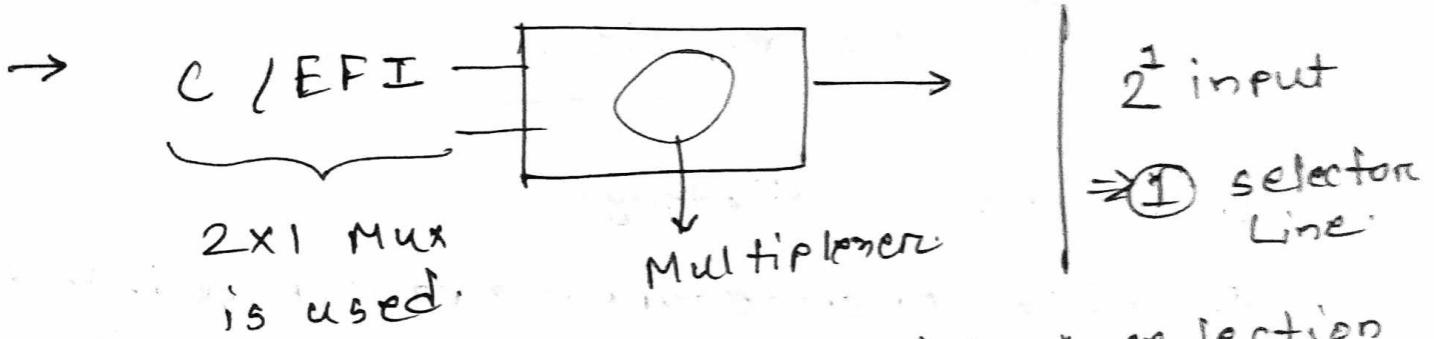
↳ x_1, x_2 : two input.

↳ High frequency is available at opf



③ Peripheral clock:

→ f / EFI (External Frequency Input)
CX-trail op)



Equation of Mux O/P having 1 selection

$$\text{Line} \Rightarrow Y = S_0 I_1 + \bar{S}_0 I_2.$$

- "Divided by 3 counter" is used to convert crystal osc. 15MHz frequency to 5 MHz frequency for 8086 input.
- "Divided by 2 counter" is cascaded to "3 - counter" for convert 5Hz to 2MHz for peripheral clock.

④ Ready block:

- ALE (Address Latch Enable) is used to enable ready signal.
- Ready 1, 2, 3 can be used.
- $\overline{\text{Asy}} \phi \overline{\text{NC}}$ is used to operate for multiple ready block.

Linear vs Non-Linear Execution.

Q4

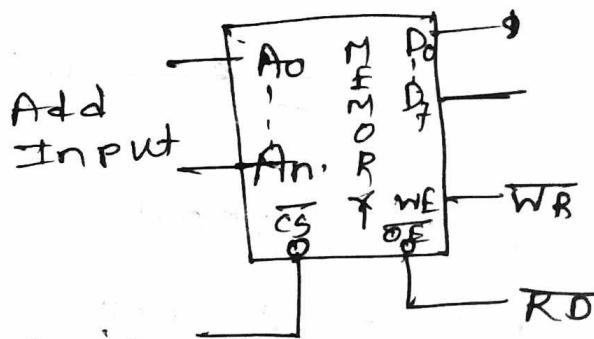
Nipa Madam

Ch-10
Brey

- ↳ Connected Devices
- ↳ Memory Interfacing!

→ Memory Unit!

- ↳ RAM
- ↳ ROM
- ↳ SRAM
- ↳ DRAM
- ↳ EEPROM.



Chip Select
(which chip should be selected (used))

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

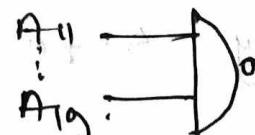
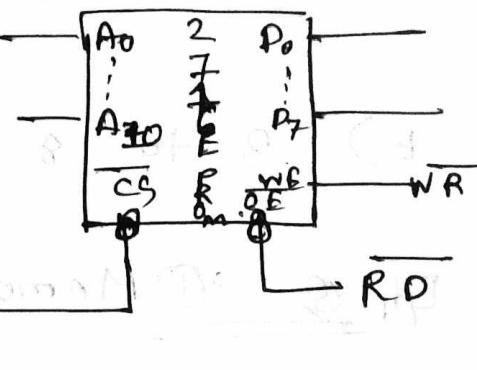
→ Memory Interfacing:

MOV → used

I/O → 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 $\rightarrow 20$
(20bit)
8086 → Memory Device

Bit mismatch.

solution: ① Address Decoding:

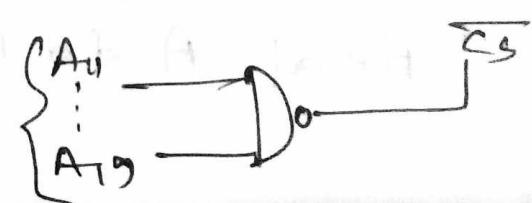
→ a) NAND Gate Decoding.

→ b) 3 to 8 Line Decoder

a) NAND Gate Decoding:

→ only one memory is selected

Loss of
bits



→ Not feasible in high Application.

b). 3 to 8 Line Decoding

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

n = Address Line = 11 (Here)

$$2048 = 2^11 \times 8$$

Q → 2: Starting and final add. of memory

NAND = 0 → CS = 1

A ₁₉	A ₁₈	A ₁₇	A ₁₆	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	
Starting 111111111111111111									1	0	0	0	0	0	0	0	0	0	0	0

F F 8 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

i. Starting Address: ff 800 H ^{Hex notation}

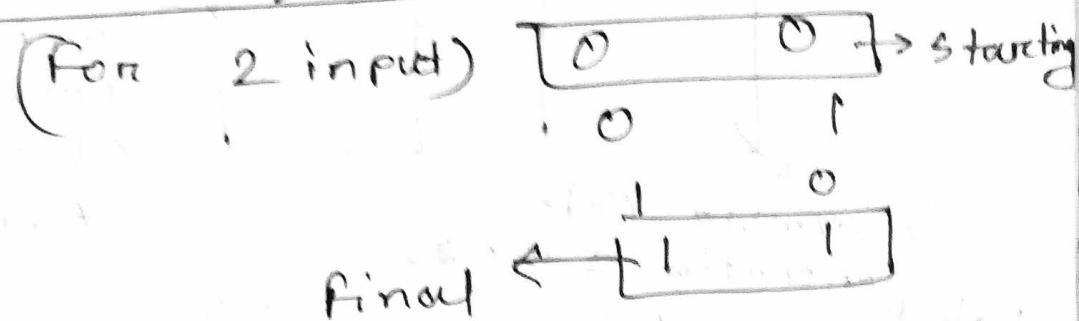
finally:

→ F F F F F F F F F F F F F F F F F F

∴ final Address: ffff ff H

Second Address: $A_0 = 1$, $A_1 = 0$

Sequence Technique: $A_1 \quad A_0$



Alternative to determine final Address:

$$\text{Final Add.} = \text{Starting Add.} + \underbrace{\text{Memory Size}}$$

$$= FF800 + 00800$$

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

Q-3: Segment Add = ??

if offset Address = 0800H, $007FFH$

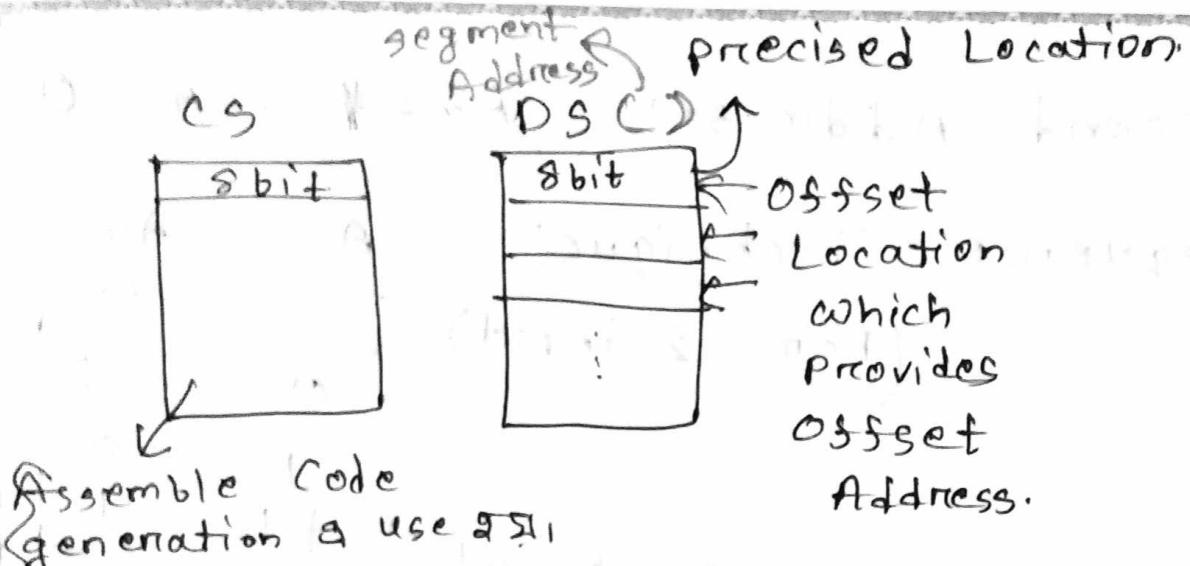
\Rightarrow Segment Register: 4 pieces

(Code, Data, extra, stack)

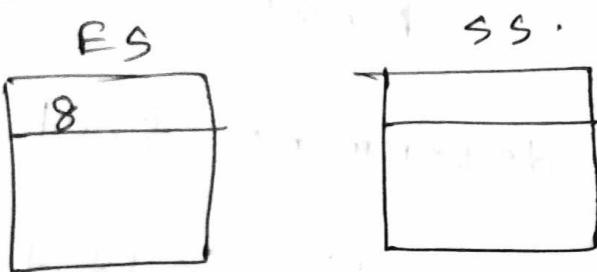
\rightarrow Physical Add?

Segment Add.

Offset Add.



Lab



FF00 * 10
0800 H

$$PA \xrightarrow{\quad} FF800H$$

$$PA = \text{segment : offset}$$

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

$$\Rightarrow \begin{aligned} SA &= FF00H \\ OA &= 0800H \end{aligned}$$

Q-2: If offset Address is not defined

$$FF00 * 10$$

$$+ F800$$

$$FF800H$$

E Practice: Repeat Problem.

① Final memory size.

② Determine starting and final Add. of
memory.

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

95 Nipa Madam

E Q6: Write assemble language program
to [read] [1 Byte] data from [0800H].

Ans:

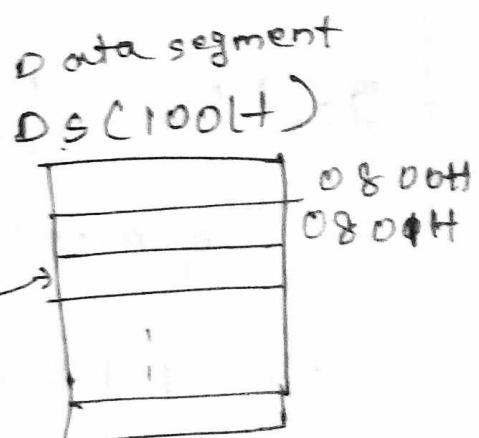
specific location 4000H

offset address 4100H

Steps:

① Segment definition

② Read operation.



MOV
By default Data segment

PUSH, POP
By default Stack
Segment

CODE
↓

By default
Code segment

Additional
↓

Segment registers

A STORE ~~20H~~

Initial Address : FF800H

PA = Segment Add * 10 + offset Add.

∴ PA = FF800H \Rightarrow Segment Add = FF00H
(starting Add)

\Rightarrow Segment Add : offset Add

\Rightarrow FF00H : 0800H

CODE:

DS
segment
Definition

Register
Addressing
for 0800H
allocation
Read operation

MOV AX, FF00H
MOV DS, AX

Error:

MOV DS, FF00H
X ↗

We cannot
direct address
data in
DS, CS, ES

MOV BX, 0800H

Error:

MOV DL, 0800H

[BX]

, DL = 0800H

MOV DL, [0800H]

Note:

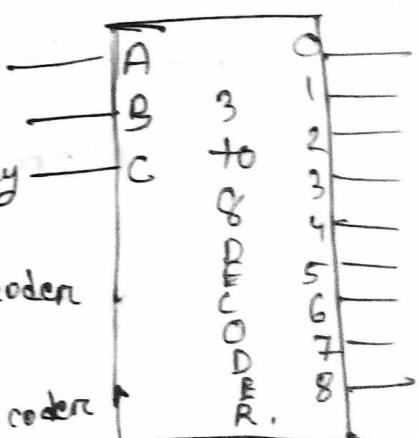
DL → To copy 1 Byte Data.

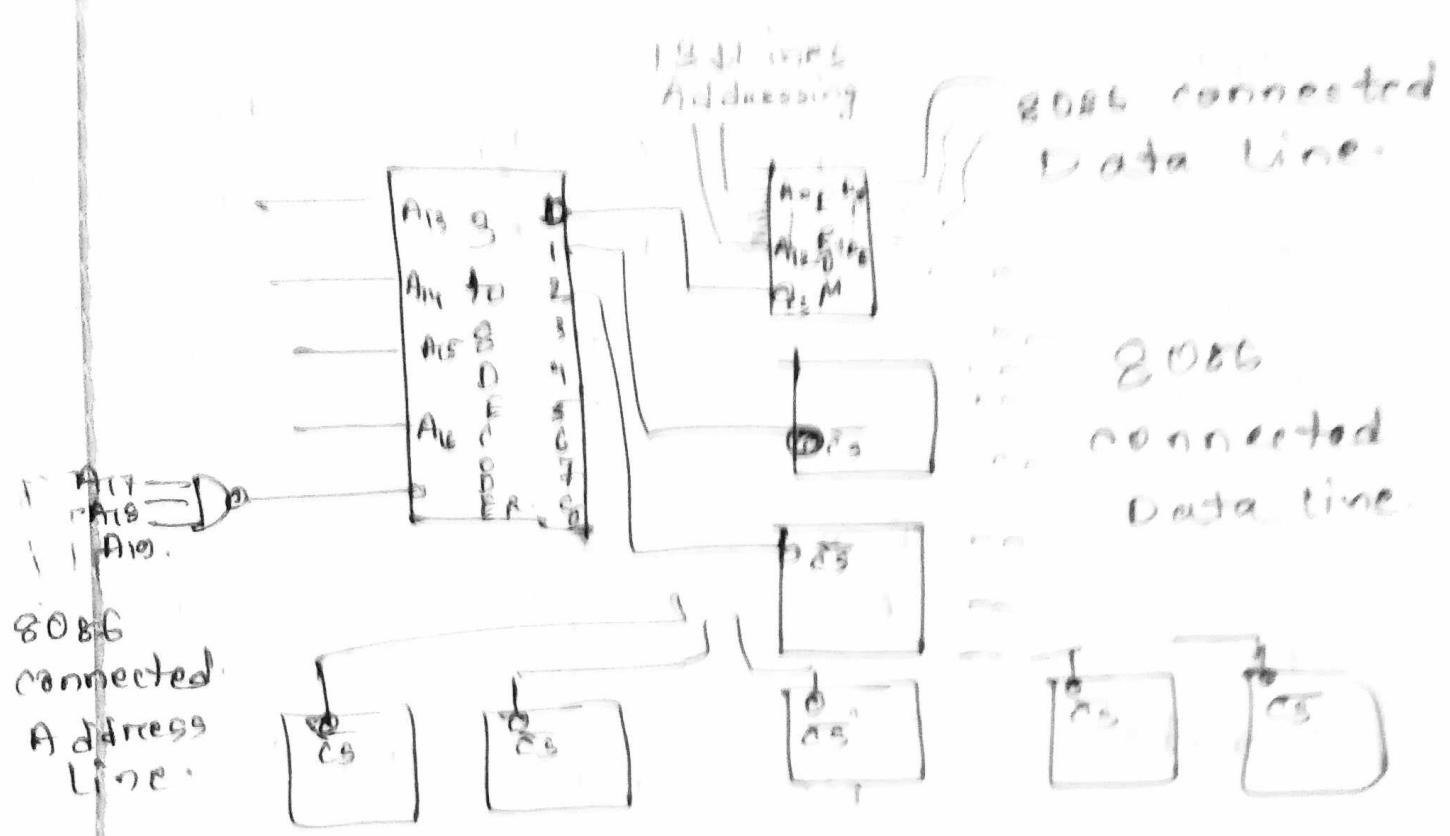
DX → To copy 2 Byte Data.



To work with 8 memory

we can use 3 to 8 decoder
instead of NAND Gate Decoder





Q18 Find memory size?

→ Memory size: 2^n

n = Address Line connected to Memory (i.e. EEPROM).

Here, $n = 13$.

$$\therefore \text{Memory size} = 2^{13}$$

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

$$= 2000H$$

Tracing Data Range:

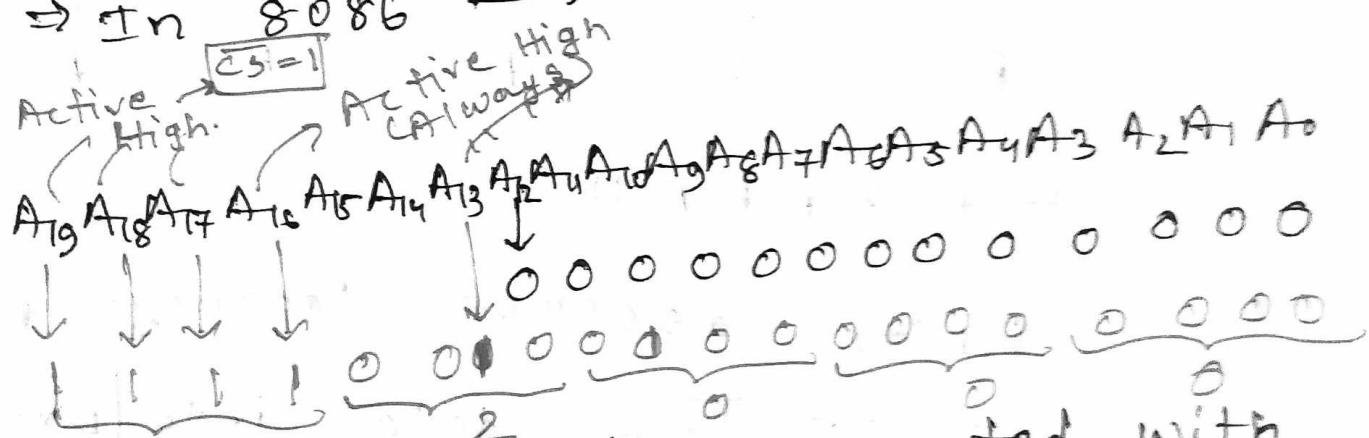
0000H to (2000-1)H

$\rightarrow 0000H$ to $1FFFH$.

Q2: Initial and final Address of

2nd Memory?

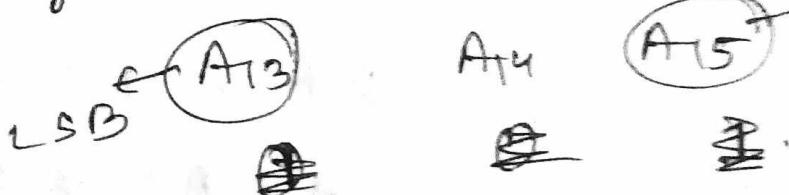
→ In 8086 → Add. Line : 20



→ Second ^F Memory ² is connected with

'1' pin 05 3 to 8 Line Decoder

→ 50, 3 to 8 Line Selection combination → MSB.



→ for 2nd Memory:

A₁₅ A₁₄ A₁₃
② ② 1

→ Initial Address for 2nd Memory

F 2000H

→ Final Address for 2nd Memory

1111 0011 1111 1111 1111
F 3 F F F F

= F 3 F F F H

S.B: Write assemble language program
to read 2B data into DX from
2002H.

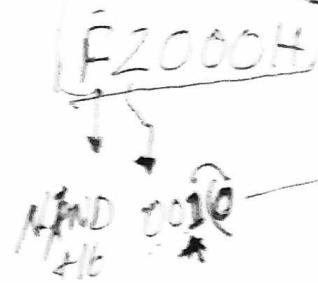
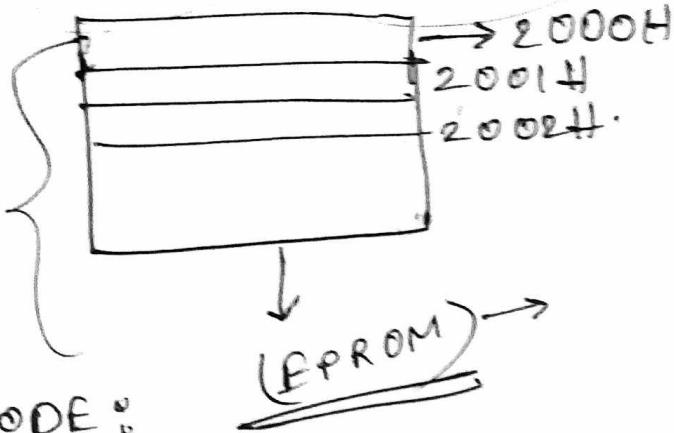
⇒ To fetch data from 2002H

S.A : F 000 * 10

+ O.A : 2002H ← from question

PA : F 2002H

F000H : 2000H



SA {

MOV AX, F000H

MOV DS, AX

MOV DX [2002H].