



Microprocessor (CSC 405)

Assignment -1

Q3.1 Explain flag register of 8086.

→ Flag register of 8086 contains one 16-bit register and 9 flags or flip flops. These flags are then bifurcated into status / condition flags and control flags in the number of 6 and 3.

Condition or Status flags are set or reset by the processor and indicate the result condition after ALU operations whereas Control flags are set or reset by the user and are used to control certain operation of the processor.

The 6 status flags with their function are listed below:-

i) Carry Flag (CF)

It indicates the carry generated out of MSB in addition or the borrow taken in the MSB in Subtraction.

OR

ii) Parity Flag (PF)

This flag indicates the even parity result.

iii) Auxiliary Carry (AC) Flag:

This shows the internal carry out of the lower nibble.

iv) Zero Flag (ZF)

This shows us the result when zero



iv) Sign Flag (SF)

This indicates the sign of the operation i.e. whether the result is positive or negative.

v) Overflow Flag (OF)

This indicates when the result is signed Arithmetic Overflow.

Control Flags

i) Trap Flags (TF):

At ($TF = 1$) ; set by the user then it enables "Single Stepping" during program execution.

At ($TF = 0$), then there won't be any single stepping but normal execution of the program.

ii) Interrupt Flag :- (IF)

At ($IF = 1$) ; set by the user, it enables hardware interrupt line INTR.

At ($IF = 0$) ; disables the INTR line.

iii) Direction Flag (DF):-

At ($DF = 1$), set by the user, it enables hardware interrupt - the processor performs "Auto-Decrement" IR offset after SI.

At ($DF = 0$), performs "Auto-increment" the IR after string op.



Q2. What is memory segmentation in 8086? What are its advantages?

→ In 8086, memory segmentation is a process in which the main memory of the computer is logically divided into different segments and each segment has its own base address.

This segmentation happens in the way of Segment Registers where there are 4 segment registers each of 16 bit. They are i) Data Segment (DS)
ii) Code Segment (CS)
iii) Stack Segment (SS)
iv) Extra Segment (ES).

Here size of each register is 64 bit fixed and size of segment register is 16 bit.

The ~~pro~~ advantages of segmentation are-

- i) No interference between code, data and stack information in case of totally disjoint segments.
- ii) It is possible to access 1 MB of memory using 2, 16 bit registers but without any segmentation 20 bit register is required.
- iii) It is possible to relocate the program by reloading the CS register with other value.
- iv) One code segment (program) can process multiple data segments one after another. by reloading the DS register within that code segment one after another.
- v) Data (DS) can be shared among multiple codes [CS] by loading DS within the active CS.



Q3 Write addressing modes of following instructions :-

a. MOV BL,[BX+SI]

→ "Base plus Index" Addressing Mode

b. AND CL,[3000]

→ "Direct" Addressing Mode

c. IN AL,DX

→ "Register" Addressing Mode.

d. ADD AX,[BX+SI + R34]

→ "Base relative plus index" Addressing Mode.

e. POP CX

→ "Register Indirect" Addressing mode.

f. SCASB

→ "Indirect Indirect" Addressing mode.

g. STC

→ "Direct" Addressing mode

h. JNZ address

→ "Register Indirect" Addressing mode.



Q4. Explain following instruction in 8086 with example.

1. LEA :- (Load Effective Address).

It loads the address of operand into provided register

Ex:- LEA DX, [SI].

2. PUSH:- It's a stack related instruction where it pushes 16 bit data from a register to the stack.

Ex:- PUSH BX :-
SP \leftarrow SP - 1
SS:[SP] \leftarrow BH
SP \leftarrow SP - 1
SS:[SP] \leftarrow BL

3. POP :- It's a stack register to 16 bit data from the stack into a register

Ex:- POP BX :-
SP \leftarrow SP + 1
BH \leftarrow SS[SP], BL \leftarrow SS[SP]

4. SCASB :- Search for a byte of accumulator into the destination string by subtraction.

∴ Ex:- Scasb : AL - ES:[DI] & update flags.
if DF = 0, DI \leftarrow DI + 1
if DF = 1, DI \leftarrow DI - 1

5. JNZ / JNE :- Used to jump if not equal or zero and uses zero flag

JNZ [1000H]



6. XOR :- It is used to perform logical Exclusive-OR opⁿ over each bit in a byte / word with the corresponding bit in another byte / word.

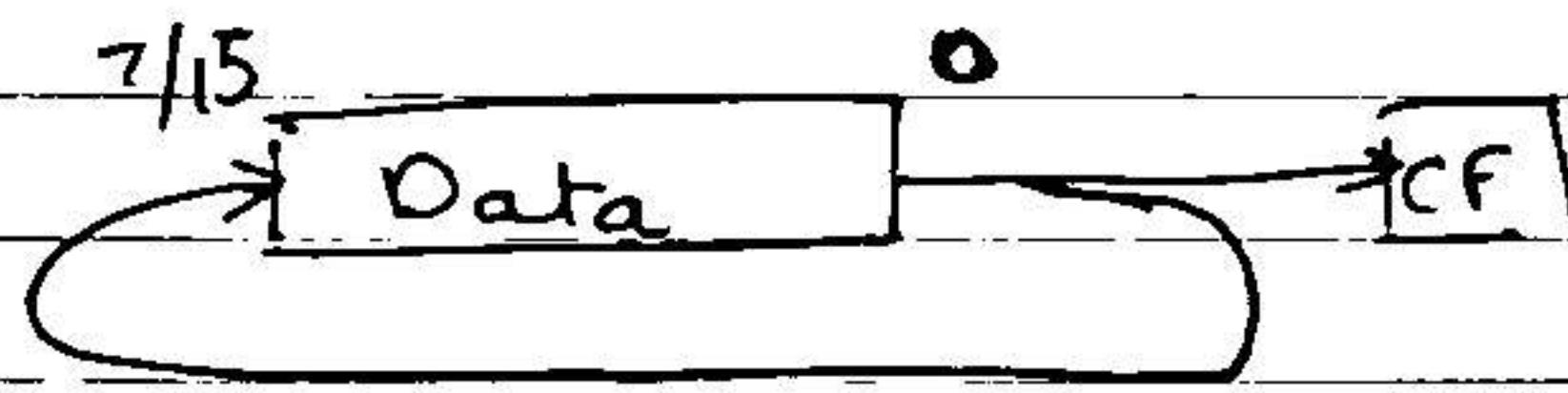
Ex:- XOR AL, 012h

7. CMP :- This is used to compare the two provided byte / word and subtract.

Ex:- CMPSB : DS : [SI] - ES : [DI] and flag update
If DF = 0 , SI ← SI + 1 and DI ← DI + 1
If DF = 1 SI ← SI - 1 and DI ← DI - 1

8. ROR :- This is used to rotate bytes of byte / words towards right i.e. LSB to MSB and to Carry Flag [CF].

Ex:- ROR AX, 01



9. Test :- This adds operands to update flags, without affecting operand.

Ex:- Test AX, [SI].

10. DAA :- This used to adjust the decimal after the add ALU operation.

Ex:- ADD AL, CL , let AL = 53 , CL = 29 .

$$\text{DAA} \quad : \quad AL \leftarrow (AL) + (CL)$$

$$AL \leftarrow 53 + 29$$

$$AL \leftarrow 82$$

$$AL \leftarrow 7C + 06$$

$$AL \leftarrow \underline{82}$$



Q5. Analyze the following program and answer the following

Assume data byte $3000 = 79H$, at $20000H = 55H$.

1. $MOV SI, 3000H$
2. $MOV BP, 2000H$
3. $MOV AX, 1000H$
4. $MOV DS, AX$
5. $MOV SS, AX$
6. $MOV AX, 5678H$
7. $MOV CY, 3909H$
8. $MOV BX, 1111H$
9. $MOV [BX], CX$
10. $MOV CX, 1000$
11. $MOV [BP+8], CH$
12. $MOV SP, 1234H$
13. $PUSH CX$
14. $PUSH AX$
15. $POP BX$
16. $YCHG CX, BX$
17. $XLAT$
18. $SAHF$

1. What is the physical address (P.A.) formed after execution of instruction no. 9? At that PA what is the value stored.

$$\rightarrow P.A. = DS^* 10H + [CX] \quad CX \rightarrow 3909$$
$$= DS^* 10H + [3909] \quad [BX] \rightarrow 1111H$$
$$P.A. = 10000H + 3909$$
$$P.A. = \underline{13909}$$

$$P.A. = DS^* 10H + [BX]$$
$$= 10000 + 1111H$$
$$P.A. = \underline{11111H}$$



The value stored at P.A is 1111H would be

~~[1111H + 3909H]~~

~~[5020H]~~

3909H

2. What is P.A formed after execution of instruction 11?
What is value stored at P.A.

$$\rightarrow P.A = DS * 10n + [2000 + 8] \\ = 10000 + 2008 \\ = 12008H$$

Value stored :- 39H.

3. What is value of SP after instruction 13 and what contents are pushed onto stack?

→ Value of SP is '1234'

~~1000~~
CX - ~~1000~~H is pushed onto stack.

4. At what address are the contents of CX are stored after execution of instruction no. 13

→ 1238H

5. What is the value of SP after execution of instruction no. 14

→ 1238H

6. What is value of BX after 15?

→ 5768H



7. What are the contents of CX and BX after 16?

→ BX → 1000H, CX → 5768H.

8. What are the contents of A after 17?

→ 5600H

9. What are the contents in flag register after 18?

→ 0028H.

Q6. With an example various addressing modes.

→ ~~Addressing~~

Example 1.

1. MOV AX, 0005H
2. MOV AX, [5000H]
- 3.

Addressing modes are the ways of locating data or operands. This addressing mode depends upon instruction.

Example:

1. MOV AX, 0005H
2. MOV BX, [5000H]
3. ~~Register~~ ADD AX, BX
4. MOV AX, [BX]
5. MOV BX, [SI]
6. MOV AL, [BX+4]
7. MOV BL, [BX+SI]
8. MOV BL, [AY+BI+05]



In Instruction no.1, $MOV AX, 0005H$, is immediate addressing mode where immediate data is part of instruction. \leftrightarrow

In 2nd instruction, $MOV BX, [5000H]$, is direct addressing mode where the address is mentioned.

In 3rd instruction, $ADD AX, BX$, is register addressing mode where the data stored in a register and it is referred using registers excluding IP.

In 4th, $MOV AX, [BX]$ is register indirect addressing mode where the address of the memory location which contains data or operand is determined in an indirect way using offset registers.

5th, $MOV AX, [SI]$ is register indirect as well as indexed addressing mode as the location offset of an IP is referred here.

6th, $MOV AL, [BX + 4]$ is register relative addressing mode where the effective address is formed by adding an 8-bit or 16-bit displacement and offset of register.

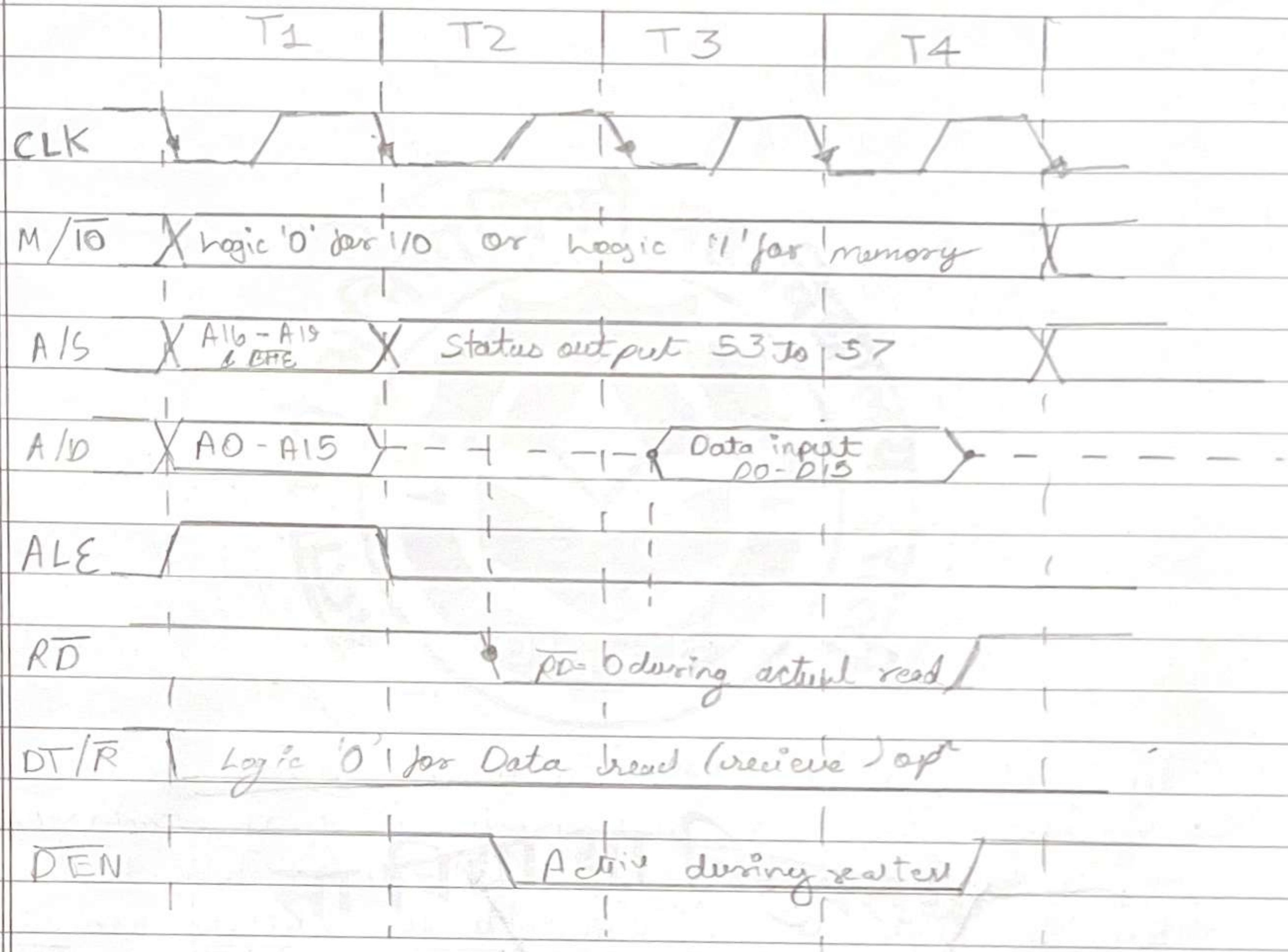
7th, $MOV BL, [BX + SI]$ is based indexed addressing mode where the effective address is addition of offset of register and an IP.

The 8th instruction, $MOV AL, [AX + DI + 05H]$ is the ~~base~~ relative based indexed addressing mode where the effective address contains offset of a register, an IP and a 8-bit or 16-bit displacement.



Q.9. Explain operation of 8086 in minimum mode with timing diagram.

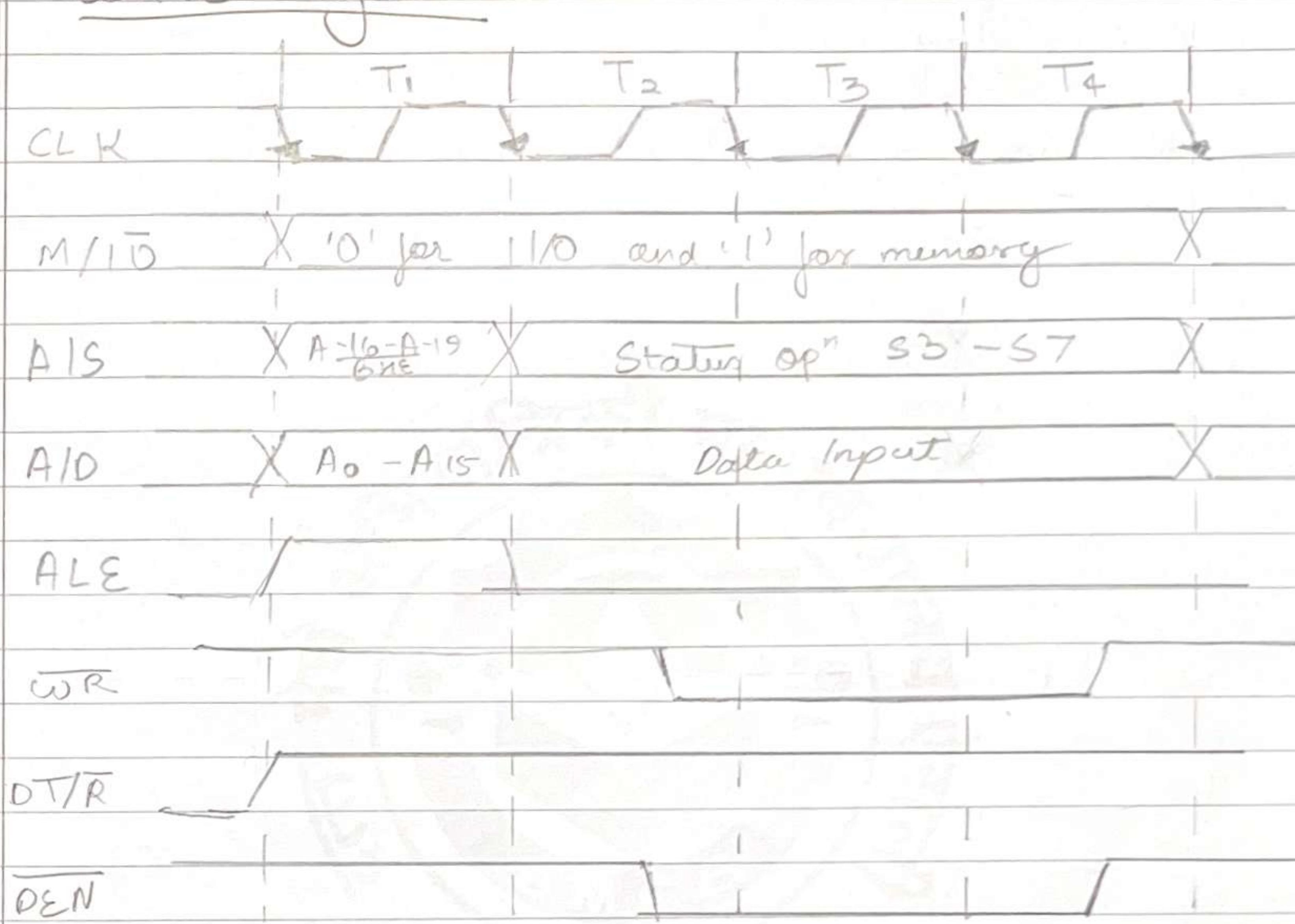
→ Minimum mode read cycle.



The read cycle in 8086 minimum mode begins at T₁, with assertion of Address Latch Enable (ALE) which is the Address Latch Enable. Clock is given to 8086. And M/IO signal is also asserted with '0' for I/O and '1' for memory from T₁ to T₄. At T₂ the address is removed from local bus and is sent to the output, this can be seen at Address/Status. The Read (RD) control signal is also activated in T₂. This enables its data bus drivers. After RD gets low, the data gets available in data bus when its high the bus driver gets activated again.



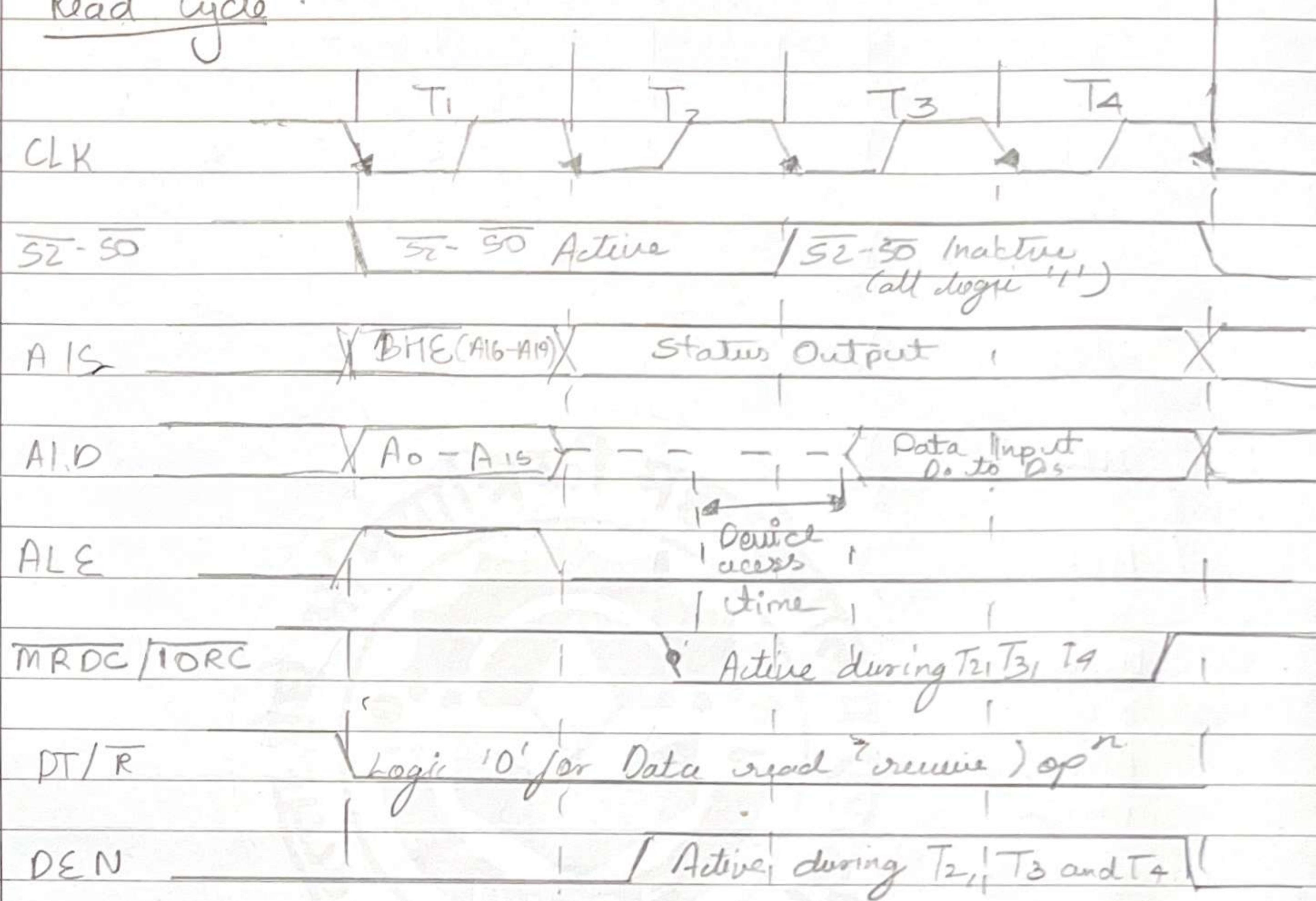
Minimum mode in
Write Cycle :-



The write cycle of 8086 minimum mode begins with assertion of ALE and permission of the address in T₁, as well as enabling the clock. Then the processor sends the data to be written to the address location. The data remains until the middle of T₄. The WR becomes active at the beginning of T₃. The B8E signals are used to select the proper bytes of memory or I/O.

- Q10. Explain operation of 8086 in maximum mode with timing diagram.



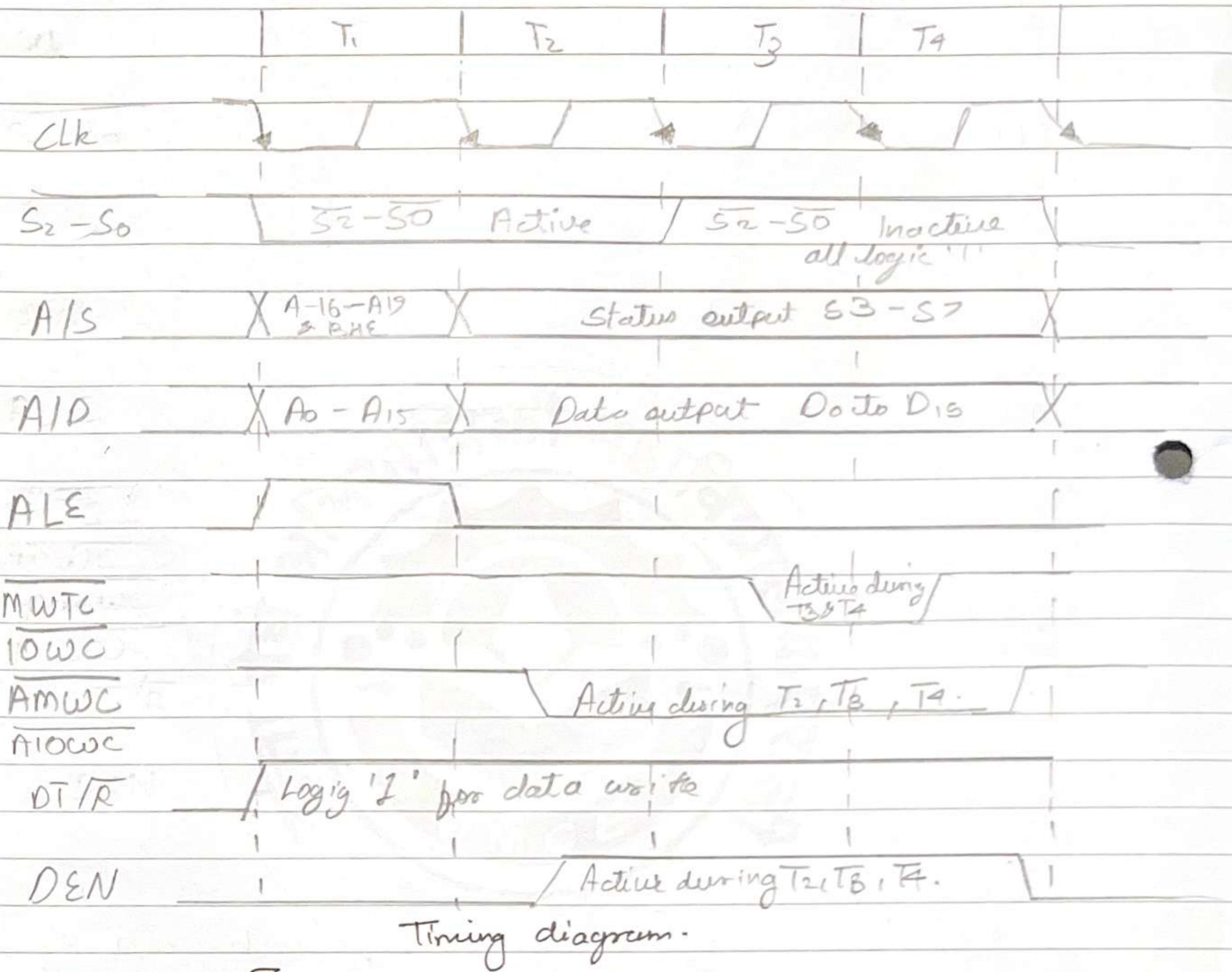
Read Cycle

The read cycle of 8086 maximum mode begins with assertion of Enable, ALE, clock and Selection bus S₂-S₀, who become inactive from T₃ to T₄.

The BHE and address bytes are read during T₁ and MRDC (memory read command) is also issued till the middle of T₂ state.

Write Cycle

The write cycle of 8086 maximum mode begins with asserting ALE, clock and Selection bus. The working is similar to read cycle, the difference being MRWC signal, status signals and available control signals.



MODULE 3

Q4. Explain operating (transfer) modes of 8257 DMA controller.

→ The operating modes of 8257 DMA C are:-

(i) Fixed Priority mode:-

DMA channels priorities are fixed and can't be changed.

CN0 → Top priority (first)

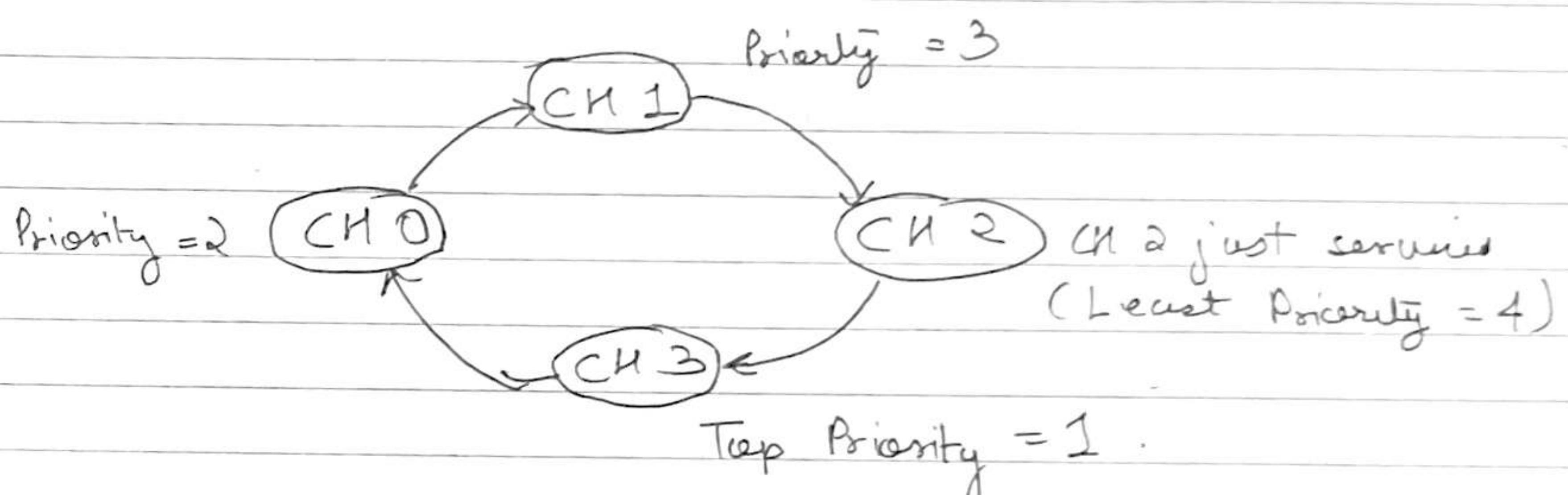
CN1 → Next to CN0 (second)



- (i) CH₂ → Went to CH₁ (third)
CH₃ → least (fourth)

(ii) Rotating Priority mode:

Priority gets rotated after serving a particular CH_0 , DMA channel. Just serviced DMA channel is assigned the least priority, while next sequentially numbered DMA - C is assigned the top priority as shown below:



(iii) Extended Write mode

When $\text{EW} = 1$ in mode set register, then DMA C generates extended write signals ("low" & memw^+). These signals that write signals are activate 1 clock cycle earlier. Slower speed I/O devices may use this extra clock cycle to complete the write operation, so that the possibility of "wait" state can be reduced.

(iv) Autoload Mode:

When $\text{AL} = 1$ in mode set register, it's the autoload mode: DMA operation take place through CH₂ only, with all the initial parameters are copied into the CH₃ registers. DMA CH₂ performs the DMA data transfer operation between memory and I/O device.



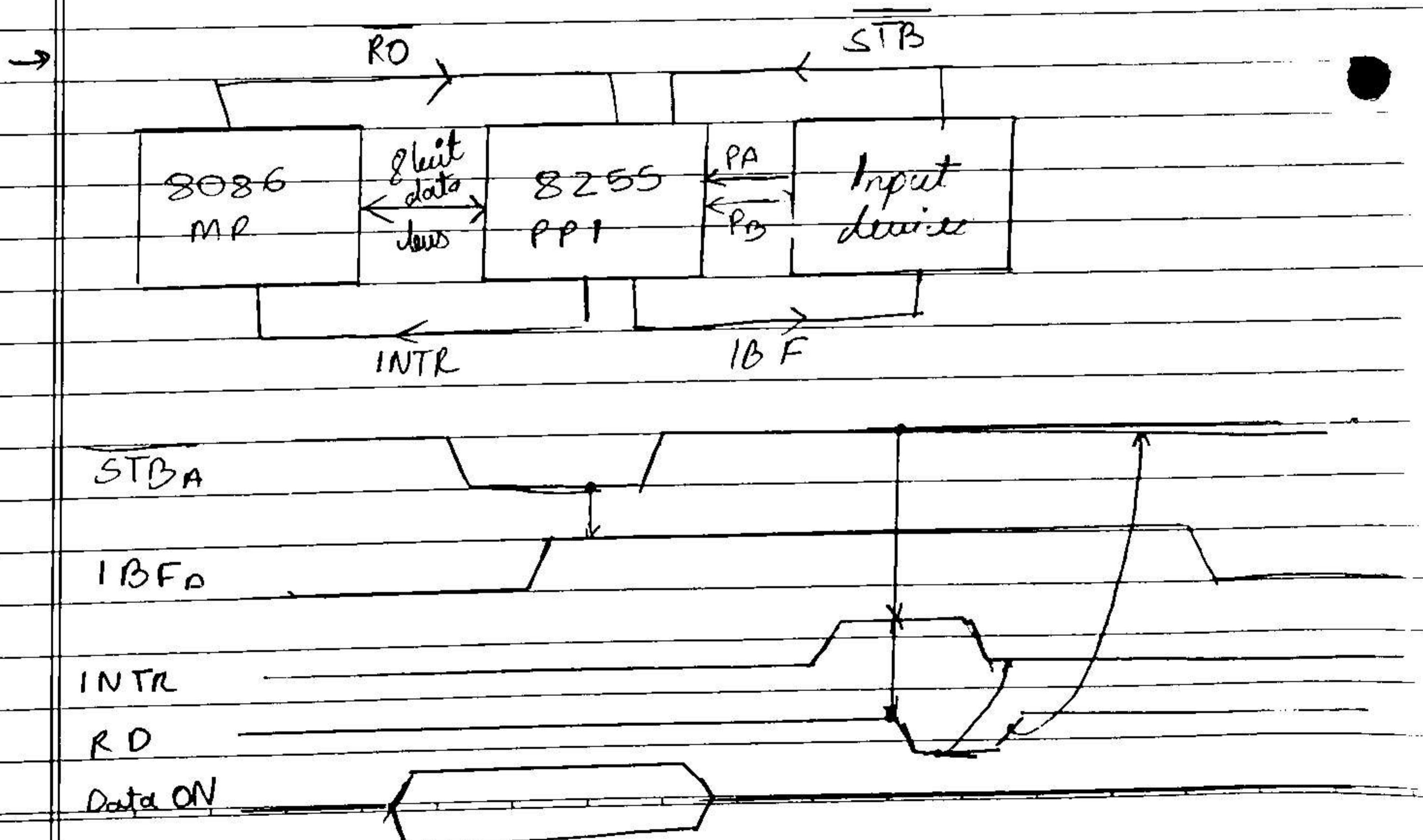
Upon Terminal count ($T_C = 1$), then DMA C performs the "Update cycle" to perform the reload of initial parameters from CH_3 to CH_2 registers. Thus CH_2 transfers same block of data in repetitive manner. This "Repeat Block op" may be used in refreshing the display system.

Q2. Explain mode 1 of opⁿ of 8255 PPI.

→ MODE 1 (strobed I/O mode).

This mode is also called as strobed input / output mode. In this mode the handshaking signals control the input or output action. Here actual data is transferred through PA and PB. Strobe / handshake signals - here 3 PC pins are used as strobe signals for PA data transfer, 3 PC pins as strobe signals for PB data transfer. Remaining 2 PC pins are used as simple I/O pins.

Input with handshake.



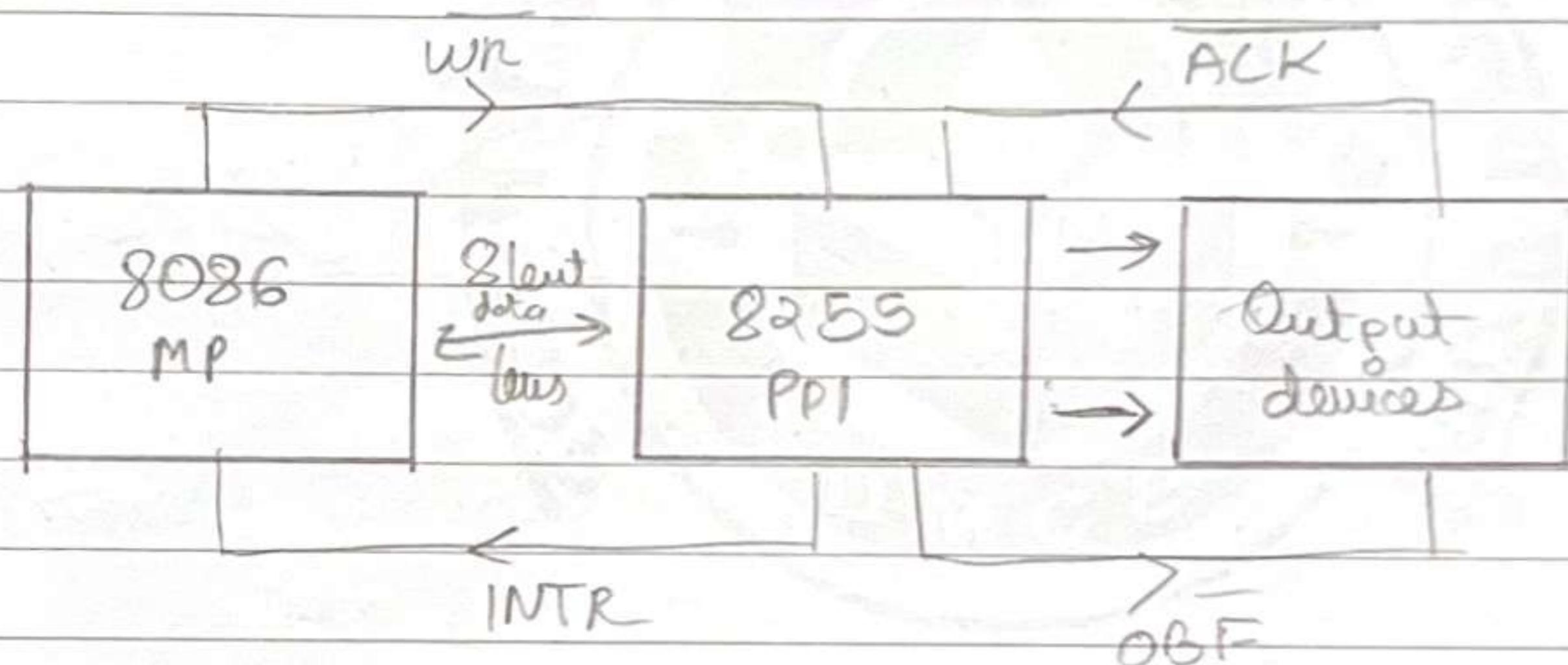


STB (Slave input) - If this line falls to logic low level, the data available at 8-bit input port is loaded.

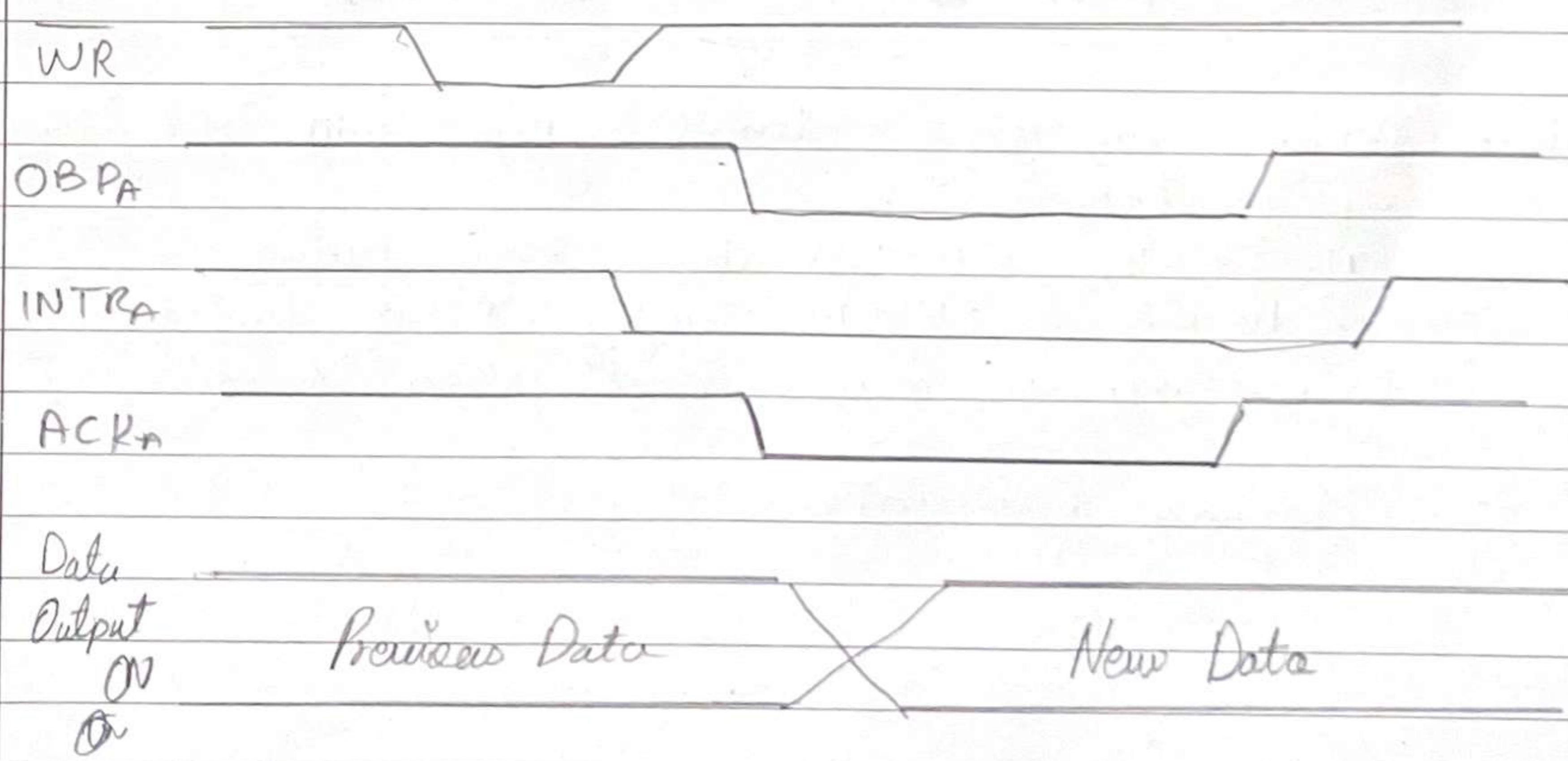
IBF :- (Input buffer full) → If logic is '1', it indicates data has been loaded into the latches and when '0' on STP and is reset by the rising edge of RD input.

INTR (Interrupt request) :- This is set high at STB pin and at IBF pin. This can be controlled by set/reset mode of by falling edge on RD input.

Output with handshakes:



Timing diagram:

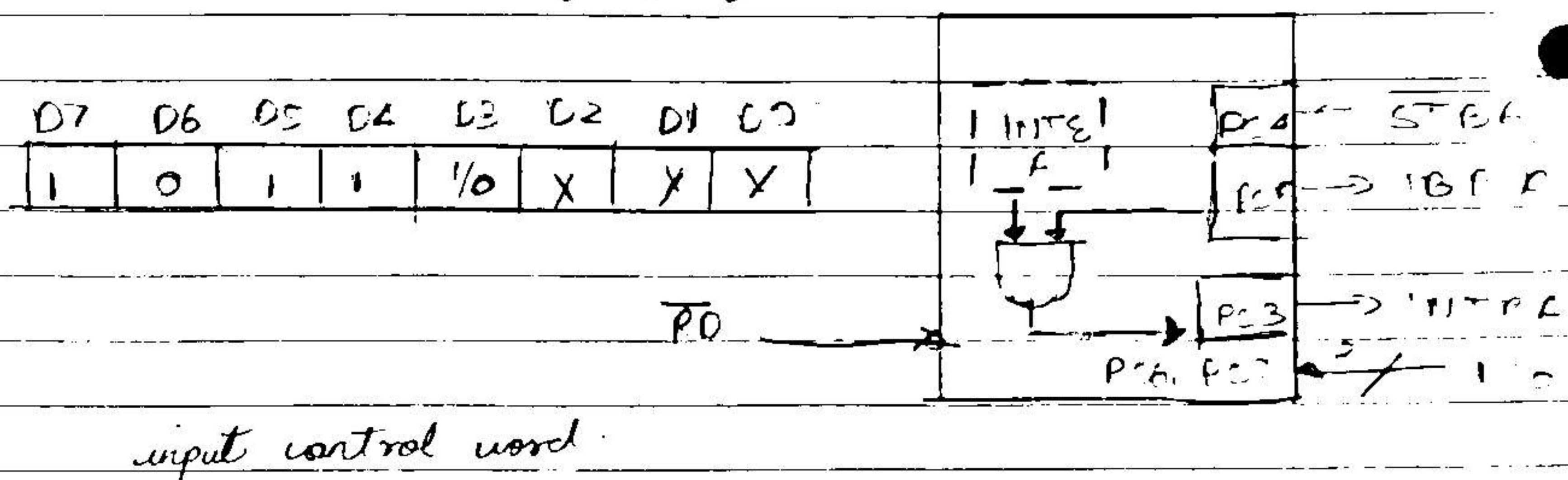




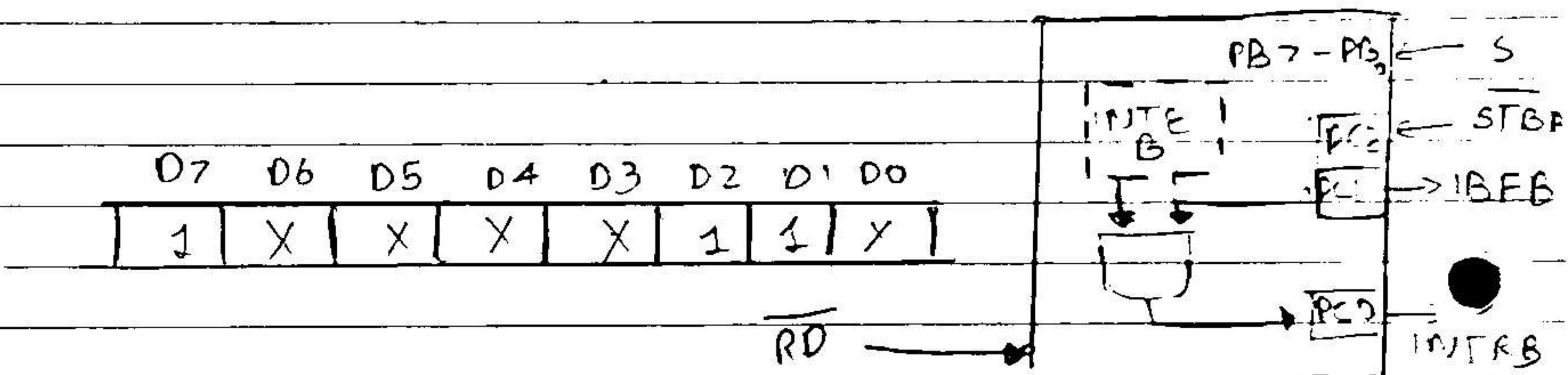
OBF (Output buffer full) - This status signal, whenever falls to logic low, indicates that the CPU has written data to output port.

ACK (Acknowledgement input): ACK acts as an acknowledgement to the given by an output driver, when low it informs the CPU that the data transferred by the CPU.

HSIR Control word group for input and output:



input control word



Q 7. Design an 8086 based system with the following specifications;

i) 32 KB EPROM using 8 KB devices

ii) 16 KB RAM using 8 KB devices.

Draw memory map and show decoding.

→ Memory Calculations :-

E PROM

Required = 32 Kb

Available = 8 kb

∴ no. of chips required = 4

Space required

No. of set required : ~~≥~~ 4 / 2Set 1: Starting address = ~~00 000 H~~
Ending = 2 : FFFFHSet Size = Chip size \times 2 = 8 Kb \times 2 = 16 Kb.
(ie.)

∴ Starting address of set 1

$$\begin{array}{r} \text{FFFF FH} \\ - \underline{3FFF FH} \\ \text{FC000H} \end{array}$$

Starting address of set 2.

$$\begin{array}{r} \text{FBFFFH} \\ - \underline{03FFFH} \\ \text{F8000H} \end{array}$$

Size of a single EPROM chip = 8 Kb.

$$\begin{aligned} &= 8 \times 1 \text{ Kb} \\ &= 2^3 \times 2^{10} \\ &= 2^13 \\ &= 13 \text{ address lines} \\ &= (A13 \dots A1) \end{aligned}$$

For RAM

Require - 16 Kb, Available = 8

No. of Chips required = 2.

Starting address of RAM is 00000 H.



Size of RAM chip = 8 kB
= 13 address lines.

Memory Map

Q-6: Show interfacing of 8259 PIC in cascaded mode

→ In It's cascaded mode when more than one microprocessor 8259 is connected to a microprocessor. The cascaded mode increases the number of interrupts handled by the system. As The maximum number of 8259s which can be interfaced is 9 i.e. 1 master and 8 slaves, the maximum number of interrupts which could be handled is 61. Each slaves INT output is connected to the IR input of the Master. The INT output is connected to the IR input of the Master. The INT output of the Master is connected to INTR input of the microprocessor. The master addresses the individual slaves through the CAS₂, CAS₁, CAS₀ lines connected



from the master of each slave. First the INT_R signal of the microprocessor should be enabled using the STI instruction. Each 8259 (Master or Slave) has its own address and has to be initialized separately by giving IC_{LSB}s as per requirement.

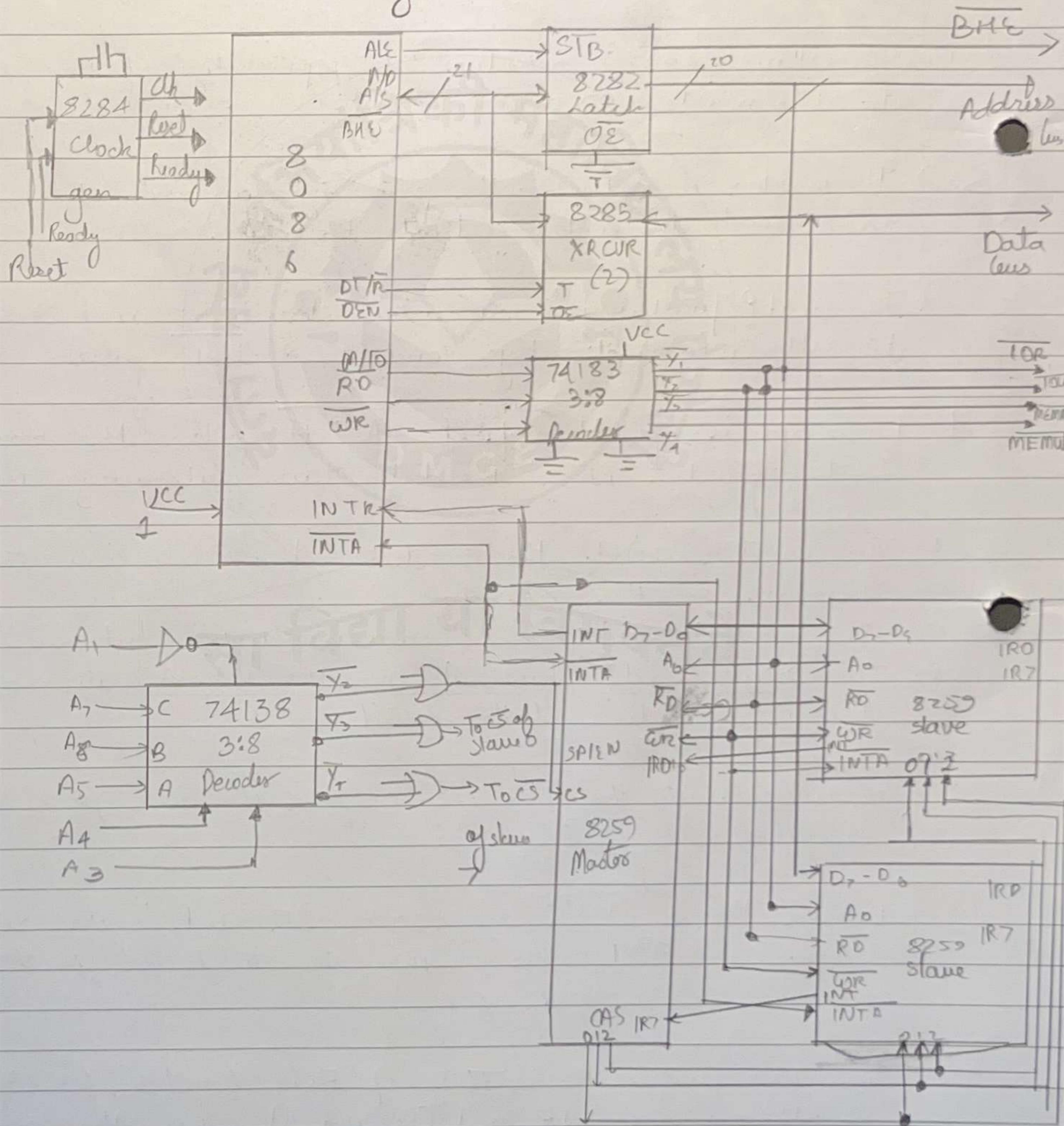
- When an interrupt request occurs on a SLAVE, :-
 - (1) The slave 8259 resolves the priority of the interrupt to the master 8259.
 - (2) The master resolves the priority among its slaves and sends the interrupt to the microprocessor.
 - (3) The MP finishes the current instruction and responds to the interrupt by sending 2 INTA pulses.
 - (4) In response of the first INTA pulse the slave places Vector number N on the data bus. following events :-
 - * The master sends the 3-bit slave identification number on the CAS line.
 - * The master sets the corresponding bit in its ISR.
 - * The slave identifies its number on CAS lines and sets the corresponding bits in its ISR.
 - (5) In response to the 2nd INTA signal the slave places N above data lines.
 - (6) During the 2nd INTA signal the ISR bit of the slave is cleared in AEOI mode, otherwise it's cleared by the B01 command at the end of the ISR.
 - (7) The MP pushes the contents of Flag Register, CS, IP, onto the stack, clears IF and TF and transfers the program.



To the address of ISR.

- (8) The ISR thus begins.

- (9) At the end of the ISR, B01 commands are given to the master and slaves to make the corresponding to "0" in the Service Register



Interfacing of Cascaded 8259.