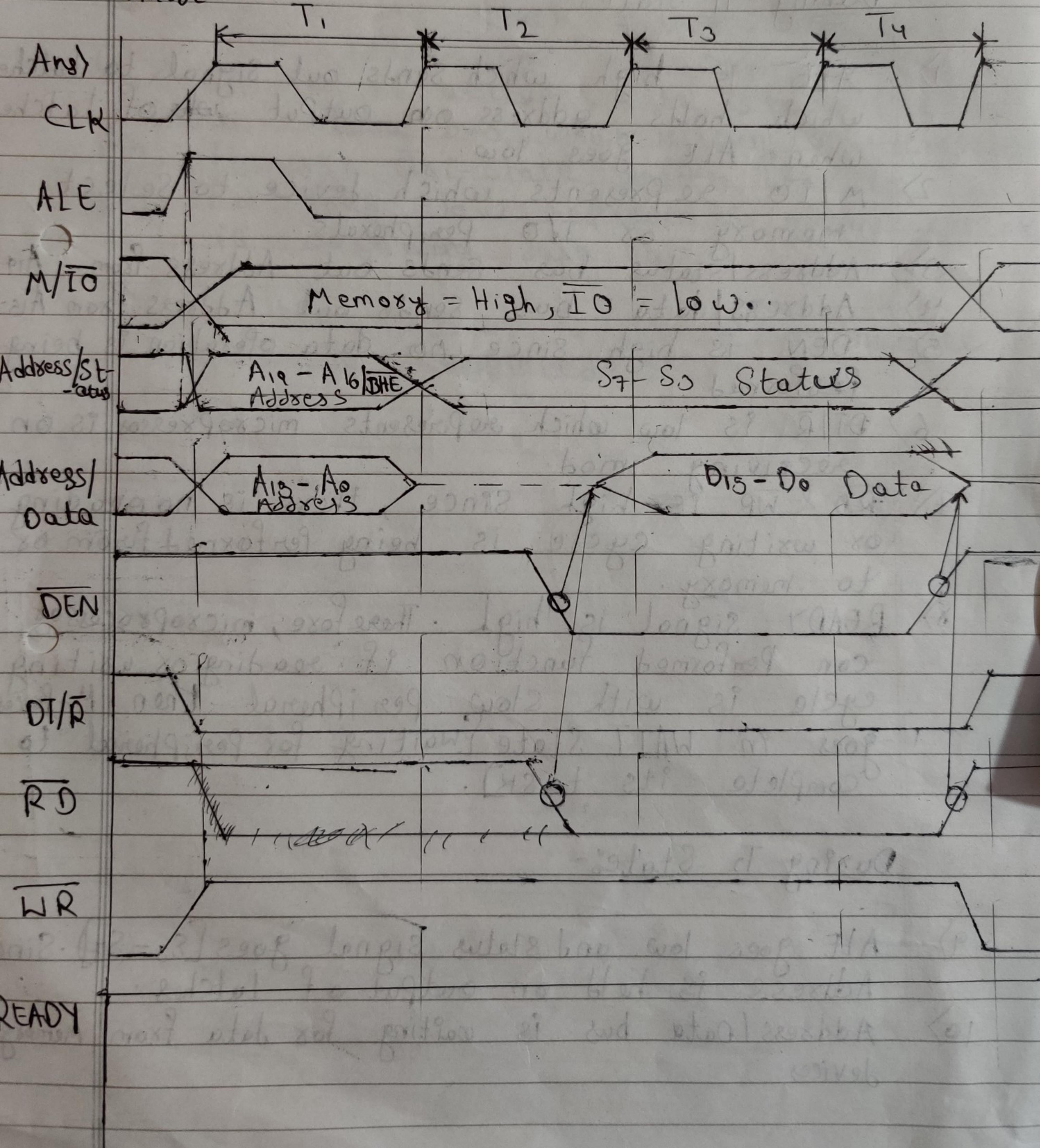


Assignment-1 M.P

Q.1) Draw and explain the read cycle in minimum mode.



During T₁ State:-

- 1) ALE is high which sends out signal to Latches which holds address on output of Latches when ALE goes low.
- 2) M/I/O represents which device to select memory or I/O peripherals.
- 3) Address/Status bus sends out Address from A₁₉-A₁₆
- 4) Address/Data bus sends out Address from A₁₅-A₀
- 5) DEN is high since no data operation is being performed
- 6) DT/R is low which represents microprocessor is on receiving mod.
- 7) RD & WR is high since, there is no reading or writing cycle is being performed from or to memory.
- 8) READY signal is high. Therefore, microprocessor can perform function if reading or writing cycle is with slow peripheral then μ-processor goes in WAIT State [Waiting for peripheral to complete its task].

During T₂ State:-

- 9) ALE goes low and status signal goes [S₃-S₇]. Since, Address is held on output of Latches.
- 10) Address/Data bus is waiting for data from Memory/I/O devices

During T₃ State:-

- ① DEN is low which Data bus is available for Transmission or receiving of Data
- ② DIS-DO is active , Data is being present on data bus , which is given to microprocessor through Tri-State buffer [8286].
- ③ RD goes low represents microprocessor is receiving data now .
- ④ At the End of T₄ State a machine cycle is completed.

During

Q. Explain INT Table.

Ans) ① The Event that causes the Interruption is called interrupt and Special routine executed to service the Interrupt is called ISR (Interrupt Service Routine).

② INT - Interrupt vector table of 8086 uses 1st 1 KB of memory from 0000 to 003FFH.

③ This table gives starting address to 256 ISR.

④ Type 0 to 4: Predefined / dedicated interrupts

⑤ Type 5 to 31: Reserved for higher processor

⑥ Type 32 to 255: Available for users.

⑦ To calculate PA of ISR, microprocessor need base address and offset address. i.e.

i.e. for each ISR four memory location are used.

⑧ Therefore, for 256 ISR 1024 memory location are used.

⑨ These ISR are known as type 0 ISR, type 1 ISR, type 2 ISR, ..., type 255 ISR.

⑩ Interrupt Sequence:-

① Push current content of Flag CS, IP into Stack memory pointed by S.P. Now S.P is decremented by -2, -2, -2 i.e 6

TSR	0000	IPL	Type I	Type 255	user available
	0001	IPH		Type 32	
	0002	CSL		Type 31	
	0003	CSH		Type 5	
				Type 4	
IPL	003FC	IPL	Type 255	Type 3	dedicated Interrupts
	003FD	IPH		Type 2	
	003FE	CSL		Type 1	
	003FF	CSH		Type 0	

2.) Trap flag and interrupt enable flag both are cleared.

3.) Microprocessor calculate PA of IVT as follows:

$$PA = 4 \times 64 - 256 = (00100)_10$$

4.) Now content of $(00100)_10$ & $(00101)_10$ is placed in I.P and content of $(00102)_10$ & $(00103)_10$ is placed in C.S

5.) Microprocessor calculates startup physical address of type 64 ISR by using current content of CS and IP

6.) It executes type 64 ISR and performs a particular task.

7.) At the end of ISR, microprocessor execute instruction RET. Now content is returned to main program by popping contents of stack memory pointed by SP into IP

8.) Now microprocessor continues the main program.

III Interrupt types:-

Type 0 to Type 4 are known as dedicated Interrupts. The reason is these ISR always perform same task.

Dedicated Interrupts:-

- 1) Type 0 (Divide by zero):- After performing division, if microprocessor is enable to store quotient in destination register, the μ P executes type 0 ISR.
- 2) Type 1 (Single Step interrupt):- After executing one instruction μ P checks Trap flag TF. If Trap flag is set, the μ P executes type 1 ISR. Then μ P returns to main program and executes next instruction then μ P executes type 1 ISR. This action is continues till Trap flag is reset.
- 3) Type 2 (NMI):- When μ P receives request at NMI, then μ P executes type 2 ISR. This ISR is used to answer the request at NMI. NMI is used for critical events such as Power Failure and emergency Shut off.

4) Type 3 (Break Point): -

Whenever μP executes instruction INT then μP executes type 3 ISR. This ISR is used for debug debugging it is used replaced single step whenever single step is time consuming.

5) Type 4 (Cover flow): -

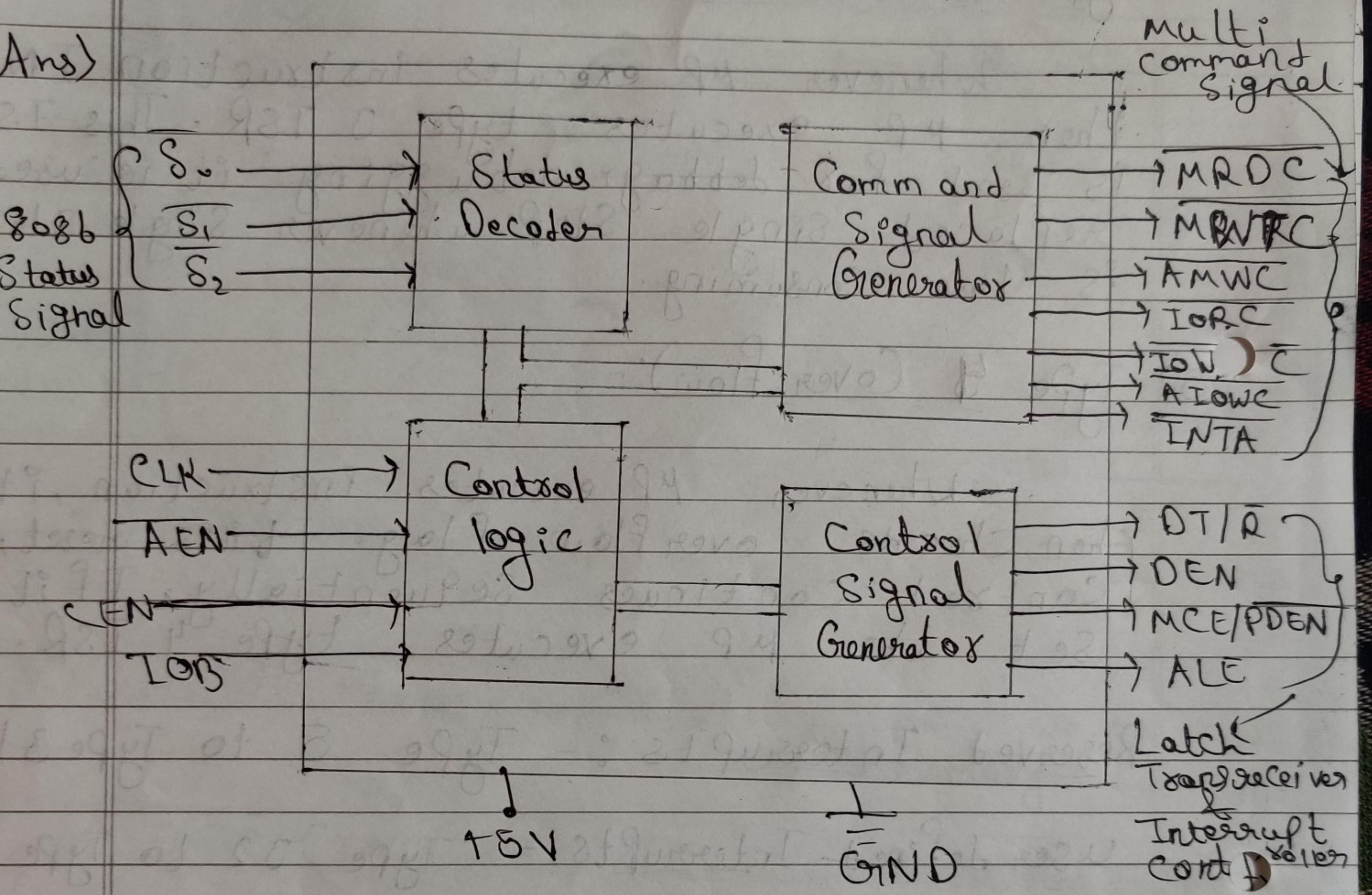
Whenever μP executes instruction, it then checks over flow flag. If it's reset, control continues sequentially, If it is set the μP executes type 4 ISR.

Reserved Interrupts : - Type 5 to Type 31

User defined Interrupts : - Type 32 to Type 255

Q.) Explain 8288 bus controller.

Ans)



- ① 8288 bus controllers are used to produce control signals.
- ② The value of control signals depends on S₂, S₁, S₀.

	S_2	S_1	S_0	
IO	0	0	0	Bus cycle
	0	0	1	INTA
	0	1	0	TORC
	0	1	1	TOWC & ATOWC
MD	1	0	0	None
	1	0	1	MRDC
	1	1	0	MRDC
	1	1	1	MWTC & AMWTC
				None.

- 1.) MRDC (Memory Read):- It is used to read data from memory.
- 2) MWTC (Memory write):- It is used to write data into memory.
- 3.) TORC (IO Read):- It is used to read data from IO.
- 4.) TOWC (IO Write)- It is used to write data into IO
- 5.) AMWTC (Advanced Memory write):- Similar to MWTC except one function i.e, it is activated one clock cycle earlier.
- 6.) ATOWC (Advanced IO write):- Similar to TOWC except one function i.e, it is activated one cycle earlier

7.) ALE :- Enable Latch.

8.) DT/R :- It is given to transceiver it is used to control direction of data flow.

9.) DEN :- It is given to transceiver through NOT gate. It is used to enable data buffer of transceiver.

10.) AEN, TOB & CEN :- They are I/P signals used in multiprocessor environment.

If TOB = 0 & AEN = 0, then 8288 is in System Bus Mode.

If TOB = 0 & AEN = 1, then 8288 is in IO BUS mode.

When CEN = 0, 8288 is disabled.

11.) MCE / PDEN :- It is O/P signal in System. Bus mode, MCE = 1 (Master Cascade Enable). In TO Bus mode, it is known as PDEN (peripheral data enable).

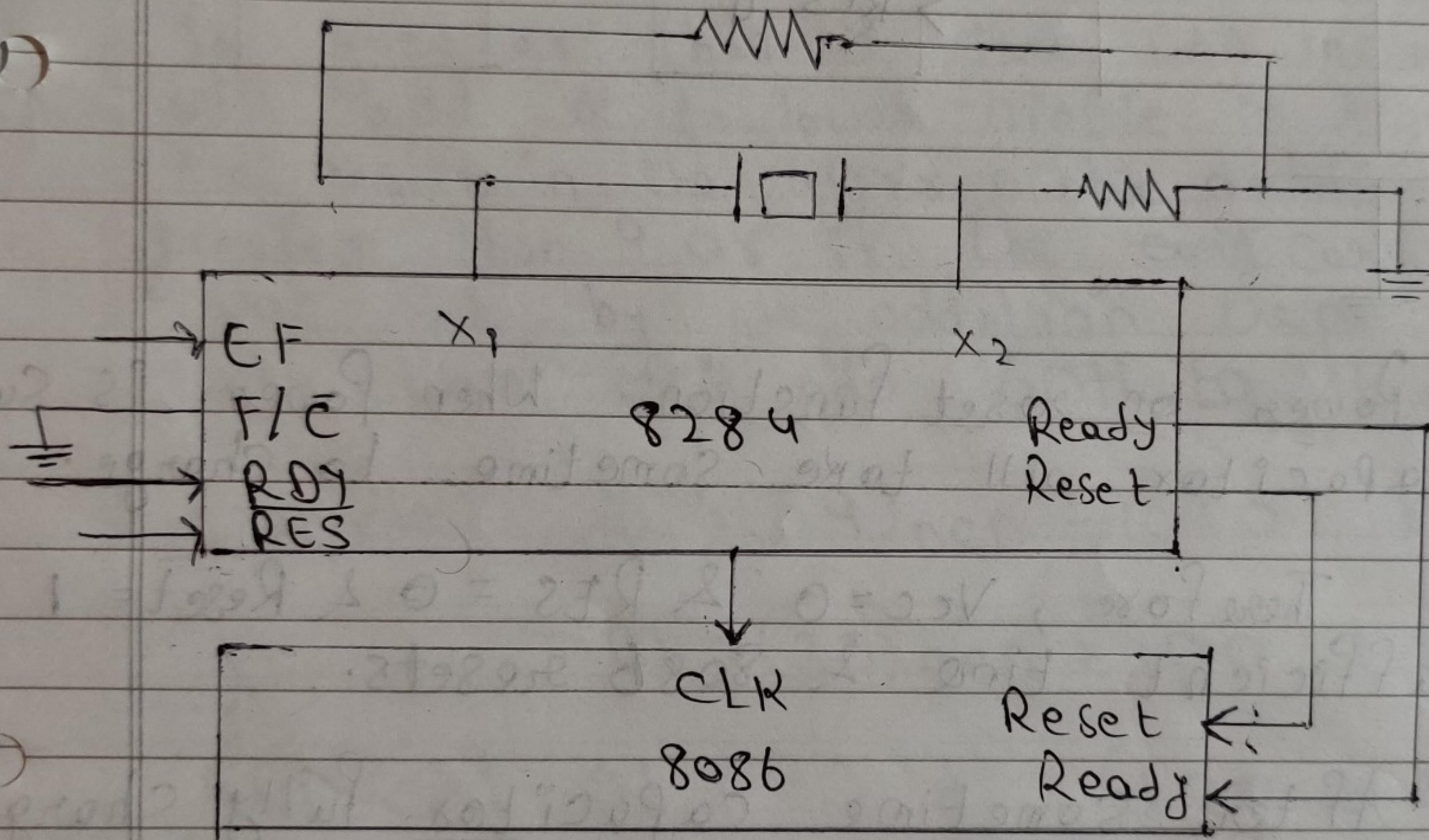
In single Processor System, 8288 is operated in System bus mode.

CEN = 1, TOB = 0, AEN = 0.

Q. Explain how microprocessor resets.

Ans) Clock Generator 8086:-

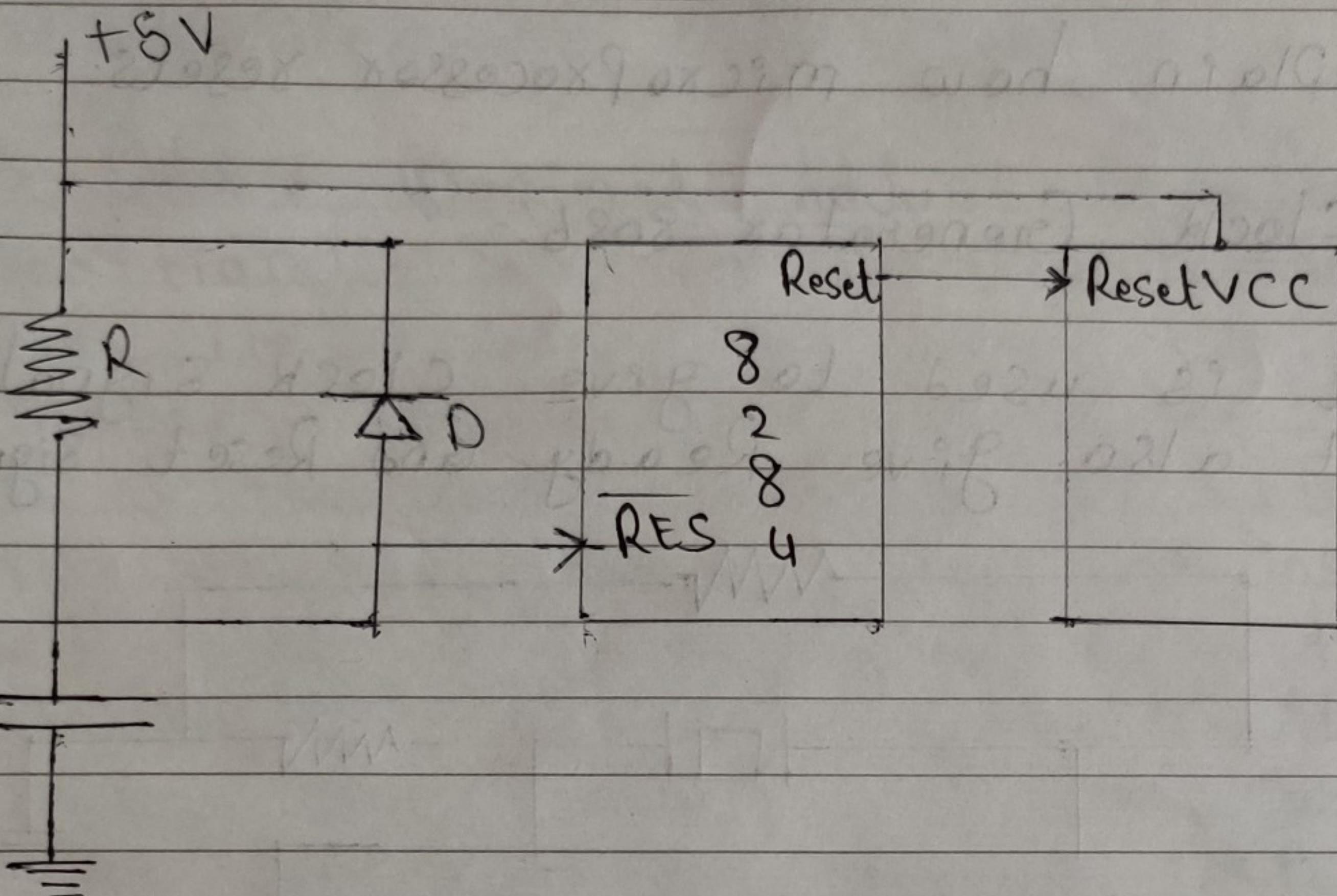
- 1) It is used to give clock signal to 8086.
- 2) It also give Ready and Reset Signal to 8086.



External circuits to Reset 8086:-

8086 can be reset in 2 ways:-

- 1) Power on reset
- 2) Manual Reset



1. Power on reset function:- When Power is supplied, capacitor will take sometime to charge.

Therefore, $V_{CC} = 0$ & $RES = 0$ & $Reset = 1$ for sufficient time & 8086 resets.

- After sometime capacitor fully charges and so $V_{CC} = 1$ & so $RES = 1$ & so $reset = 0$ so 8086 comes out of reset state.

Q. > Explain following instructions:-

(i) DAA (Decimal Adjust AL after BCD Addition):- This instruction is used to make sure the result of adding two packed BCD numbers is adjusted to be a legal BCD number. If the lower nibble in AL after an addition is greater than 9 then DAA instruction will add 6 to lower nibble in AL. If the result in the upper nibble of AL is now greater than 9 or if the ~~cot~~ carry flag was set by the addition then the DAA instruction will add 60H to AL.

Example:- $AL = (59)_{BCD} = (0101\ 1001)_2$,

$BL = (35)_{BCD} = (0011\ 0101)_2$,

$ADD\ AL, BL \quad \therefore AL = (1000\ 01110)_2$

$AL = (8E)_{BCD}$

DAA : Add 0b $\because E > 9$

$AL = (1001\ 0100)_2 = (94)_{BCD}$

$AL = (94)_{BCD}$ which is correct BCD.

(ii)

XLAT:- XLAT / XLATB instruction is used to translate a byte from one code to another code. The instruction replaces a byte in the AL register with a byte pointed to by BX in a lookup table in memory. Before the XLAT instruction can be executed, the lookup table containing the values for new code must be put in memory, and the offset of the starting of the lookup table must be loaded in BX. The code byte to be translated is put in AL. To point to the desired byte in the look up table, the XLAT instruction add the byte in AL to start the offset BX. It then copies the Byte from the address pointed by (BX+AL) back into AL.

Example:-
MOV BX,OFFSET EBCDIC_TABLE
; Point BX at Start of EBCDIC
; table in DS.

XLAT B
; Replace content in AL with
; EBCDIC from table.

The XLAT instruction can be used to convert any code of 8 bits or less to any other code of 8 bits or less.

(iii) CMP:- This instruction compares a byte from the specified source with a byte from the specified destination [word can also be compare]. The comparison is actually done by subtracting the source byte or word from the destination byte or word. The source and the destination are not changed, but the flags are set to indicate the results of the comparison. AF, OF, SF, ZF, PF and CF are updated by CMP instruction.

example: CMP CX, BX

IF $CX = BX$	<u>CF</u>	<u>ZF</u>	<u>SF</u>
	○		○ ; Result of ; subtraction is ○

$CX > BX$	○	○	○ ; No borrow required, so $CF = 0$
-----------	---	---	--

$CX < BX$	1	○	1 ; Subtraction ; required borrow ; so, $CF = 1$
-----------	---	---	--

Source can be register, memory location or immediate number.

destination can be register & memory location

2. Identify the addressing mode of the following instructions:
- i) MOV ax, bx :- Register addressing mode
Data is stored in register.
 - ii) IN al, [2000] :- Displacement or direct addressing mode. Content of Port location 2000 is translated is AL.
 - iii) DAA :- Implied addressing mode
Correct BCD number is form in AL.
 - iv) MOV ax, [bx + si + 4] :- Base indexed addressing mode. After adding bx, si & 4 result of memory location is placed in ax.
 - v) MOV al, 05h :- Immediate addressing mode.
Place 05h in al.