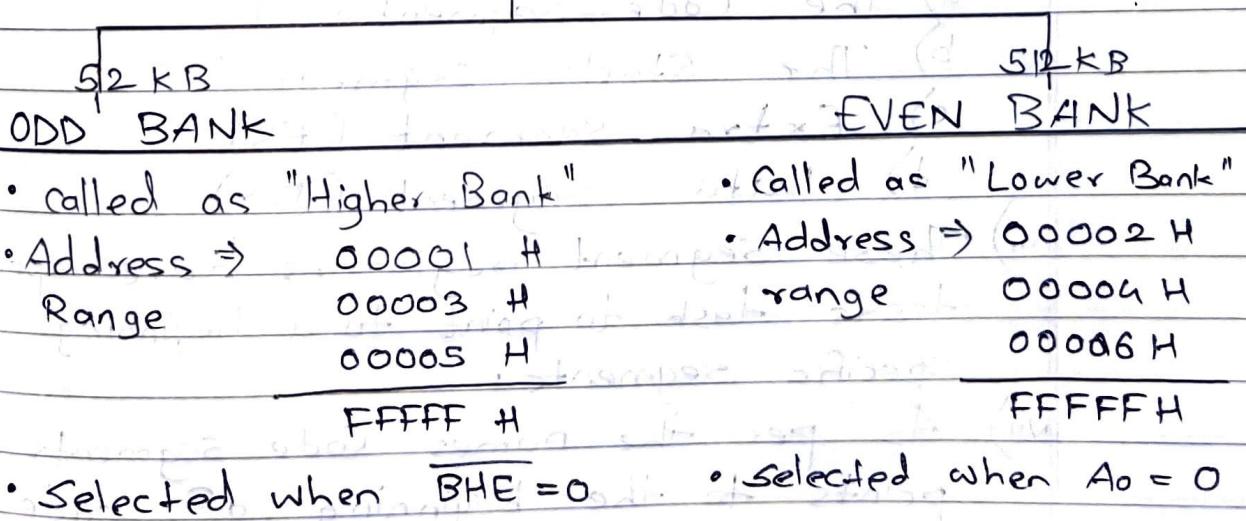


# 1. The Intel 8086 Architecture

Explain memory banking in 8086.

8.1  
 $\Rightarrow$  i) 8086 has 16-bit data bus hence it should be able to access 16-bit data in one operation.

- ii) But the memory chips available normally are such that each location has 8-bits.
- iii) Hence to read 16-bit it needs to access 2 memory locations.
- iv) However, if both these memory locations are in same memory chip then the address bus will have to provide two addresses sequentially and requires the double time also 16-bits could not be accessed simultaneously.
- v) To solve this problem, the memory of 8086 is divided into two banks each bank provides 8-bits.
- vi) One bank contains all even addresses called "Even Bank" and other bank contains all odd addresses called as "Odd Bank".



BHE	A <sub>0</sub>	Action
0	0	Access 16-bit word
0	1	Access odd byte D <sub>8</sub> -D <sub>15</sub>
1	0	Access even byte D <sub>0</sub> -D <sub>7</sub>
1	1	No Action

Q.2 Explain Segmentation of 8086 microprocessor.  
Give its advantages.

- ⇒ i) 8086 has 20-bit address bus while the registers are of 16-bit.
- ii) To access a memory location, we thus need to provide 20-bit addresses while the registers are 16-bit, this is made possible using segmentation.
- iii) Segmentation in 8086 refers to division of the 1MB main memory into segments or blocks of 64 KB each.
- iv) There are four segment registers that point to the beginning of a segment and are called as segment registers.
- v) They are-
- The Code Segment (CS)
  - The Stack segment (SS)
  - Extra Segment (ES)
  - Data segment (DS)
- vi) These segment registers have specific tasks task to point to a beginning of specific segments.
- vii) As per the names Code segments (CS) points to the beginning of code

segments (that stores the programs or codes).

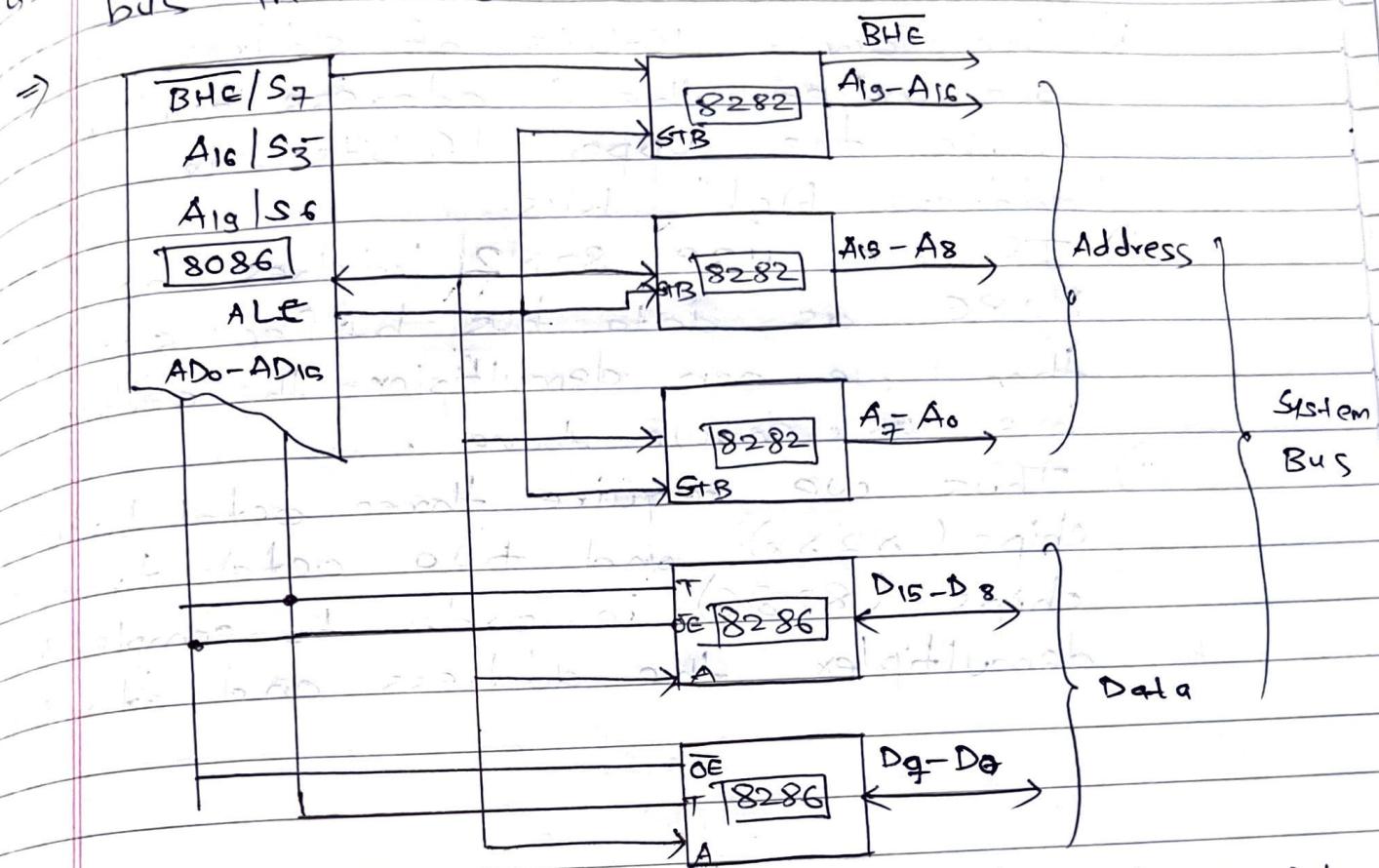
- vii) Stack segment points to the beginning of stack segment. Data segment (DS) and Extra segment (ES) are used to point data segments, wherein ES is used only during the string instruction execution.
- viii) Thus the segment register do the task of pointing to the starting of segment while the pointer or index registers point to a location within the segment.

<u>Index Registers</u>	<u>Default Segment registers</u>
BX	DS
SI	DS
DI	DS
SP	SS
BP	SS
IP	ES

- Advantages of Memory Segmentation in 8086 :

- i) The most important advantage is that programmer can access a memory that required 20-bit address, by using 16-bit registers only.
- ii) The program, data and stack are stored in separate blocks in memory and hence the three are organized in a modular fashion.
- iii) It also helps in object-oriented programming to store data of an object.
- iv) Finally, sharing of data or passing of data from one program to another is easily possible due to segmentation.
- v) The 8086's segmentation architecture allows for a large address space of up to 1MB.
- vi) Also segmentation makes the data relocatable as the program uses only offset registers pointers while segment points to the base of segment.

Q.3 Draw and explain demultiplexing of address bus in 8086.



- i) In 8086 up, the address bus is 20-bit wide, which means it can address up to  $2^{20}$  memory locations.
- ii) However 8086 has 16 data lines, which means it can transfer data in 16-bit chunks. This represent a problem when trying to address memory locations that require more than 16 bits of address bus.
- iii) To overcome this limitation, 8086 uses a technique called address demultiplexing.
- iv) Address demultiplexing splits the 20-bit address bus into two 16-bit

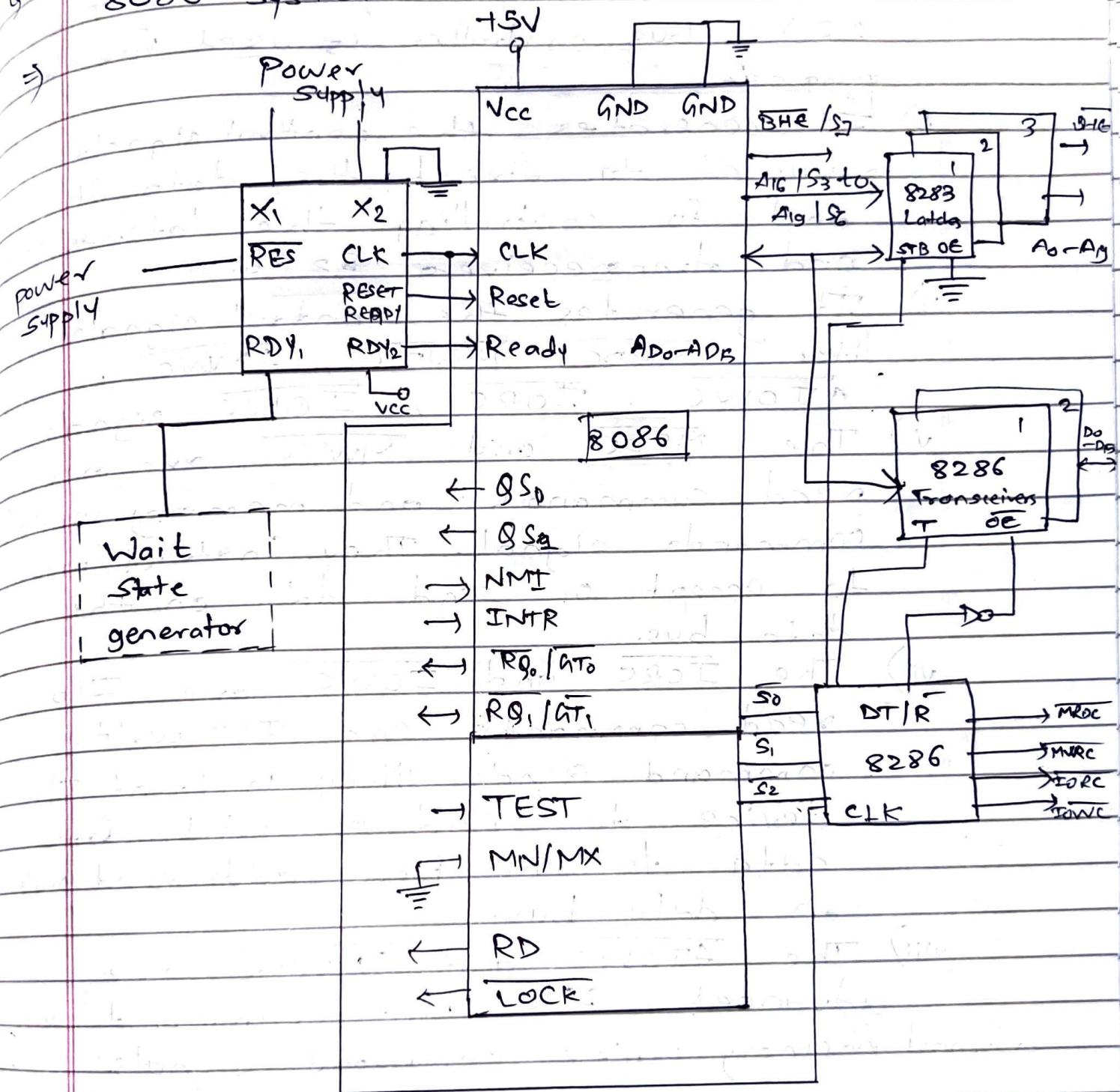
buses, which are used to address different parts of memory system.

v) The lower 16-bits of address bus are called the "address low" bus, while the upper 16-bits are called "address high" bus.

vi) If we use 8288 latches and 8286 as data bus buffer simultaneously then we can demultiplex the address bus as shown above.

vii) Thus we require three octal latch chips (8288) and two octal transceiver chips (8286) in order to completely demultiplex the address and data bus.

Q.4 Draw and explain maximum mode configuration 8086 system.



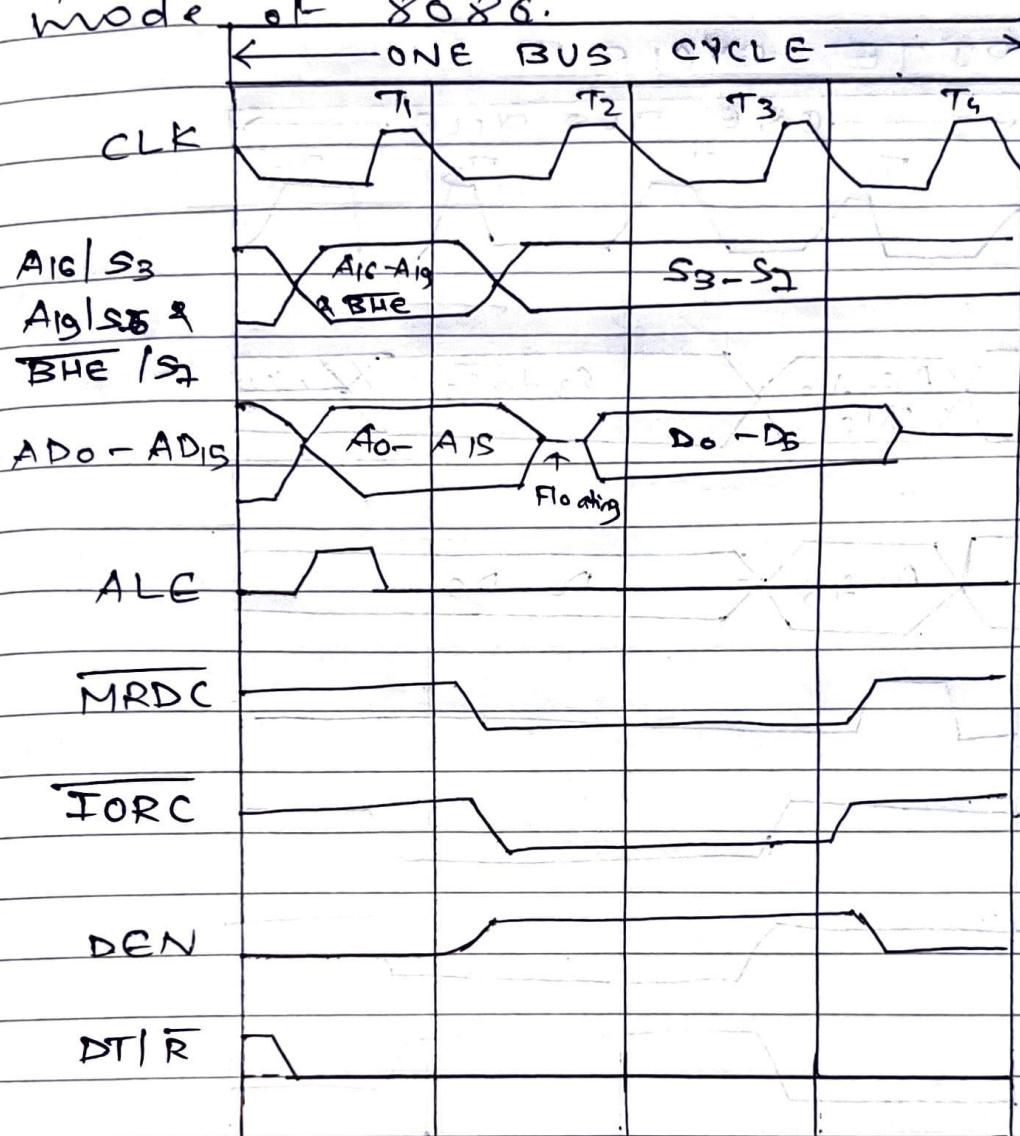
- i) The above block diagram of the 8086 system in maximum mode.
- ii) Additional circuitry is required to generate control signals.
- iii) The additional circuitry from status

signals ( $S_2 - S_0$ ) to produce  $I/O$  and memory transfer signals. The Intel 8288 bus controller is used for this purpose.

- iv) It generates the control signals required to direct the data flow and for controlling the latches 8282 and transreceivers 8286.
- v) It generates the control signals like MRDC, MWTC, AMWC, AIOWC, IORC, IOWC signals.
- vi) The MRDC and MWTC are memory read commands and memory write commands signal. They instruct memory to accept or send data on the data bus.
- vii) The IORC and IOWC are  $I/O$  read commands and  $I/O$  write command signals. They instruct  $I/O$  device to read or write data to and from addressed port on data bus.
- viii) The AIOWC and AMWTC are advanced  $I/O$  write command and advanced memory write command signals.

Q.5

Draw and explain memory and machine cycle timing graph / diagrams in maximum mode of 8086.



### • READ CYCLE

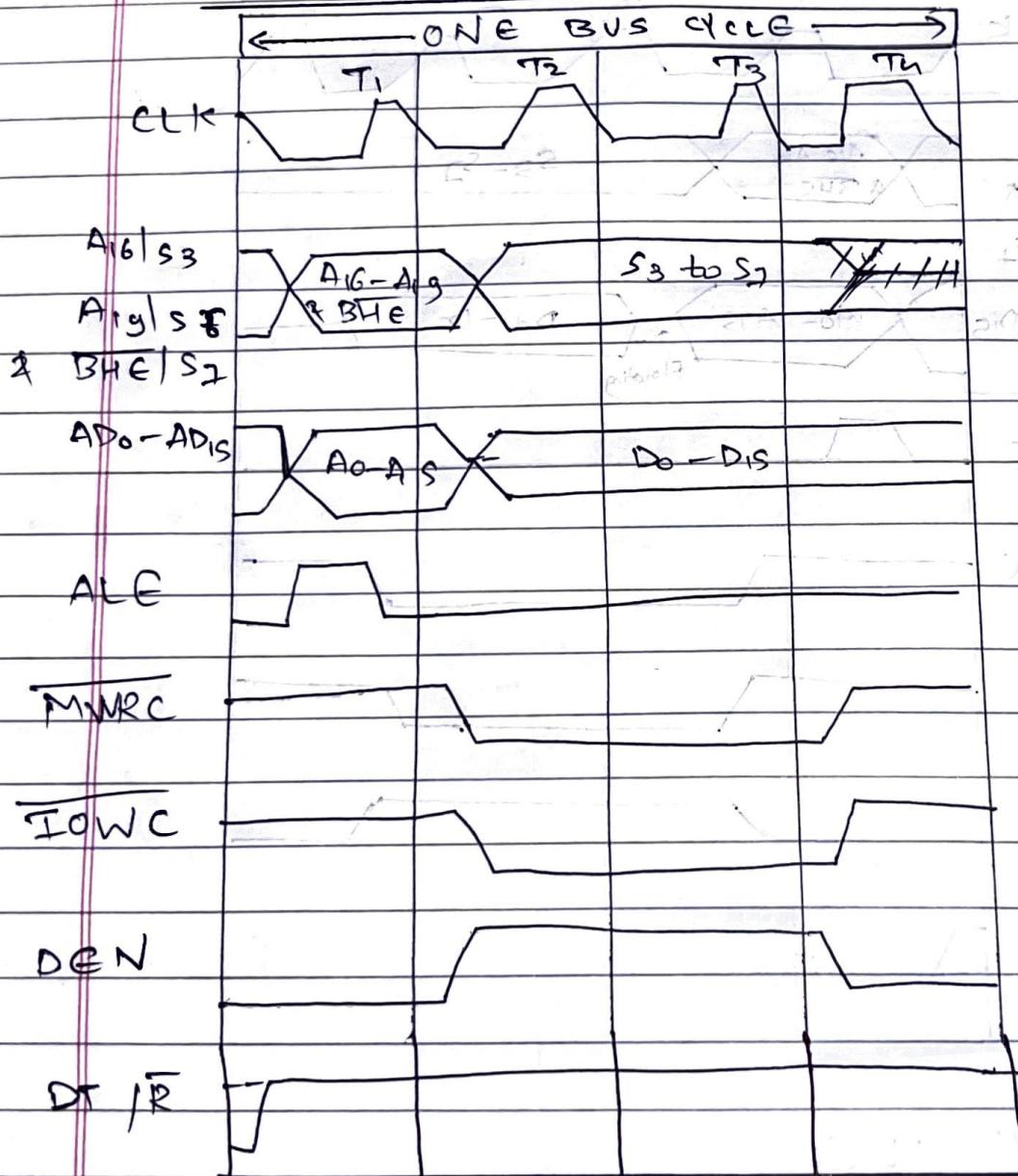
→ The above figure shows read cycle in maximum mode of 8086.

i) Status line  $S_2$   $S_0$  links are taken into account. These lines are active for  $T_1$  and  $T_2$  cycle. After that they are inactive.

ii) ALE, Memory read, I/O read, DT/R, DEN are generated by 8288 bus

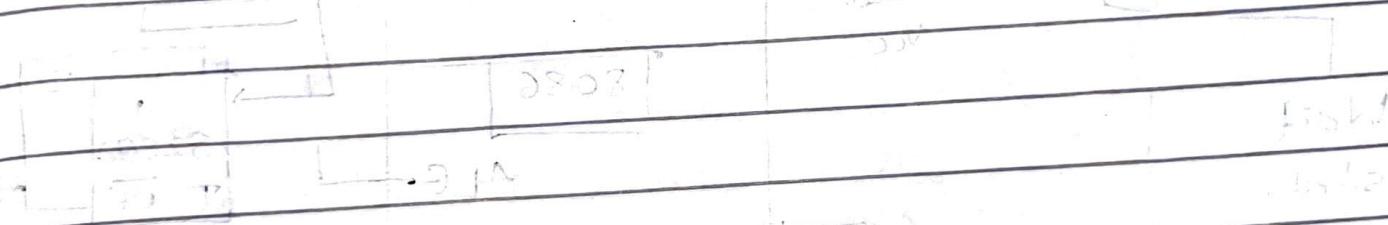
controller. They are not generated by the CPU directly.

### • WRITE CYCLE :



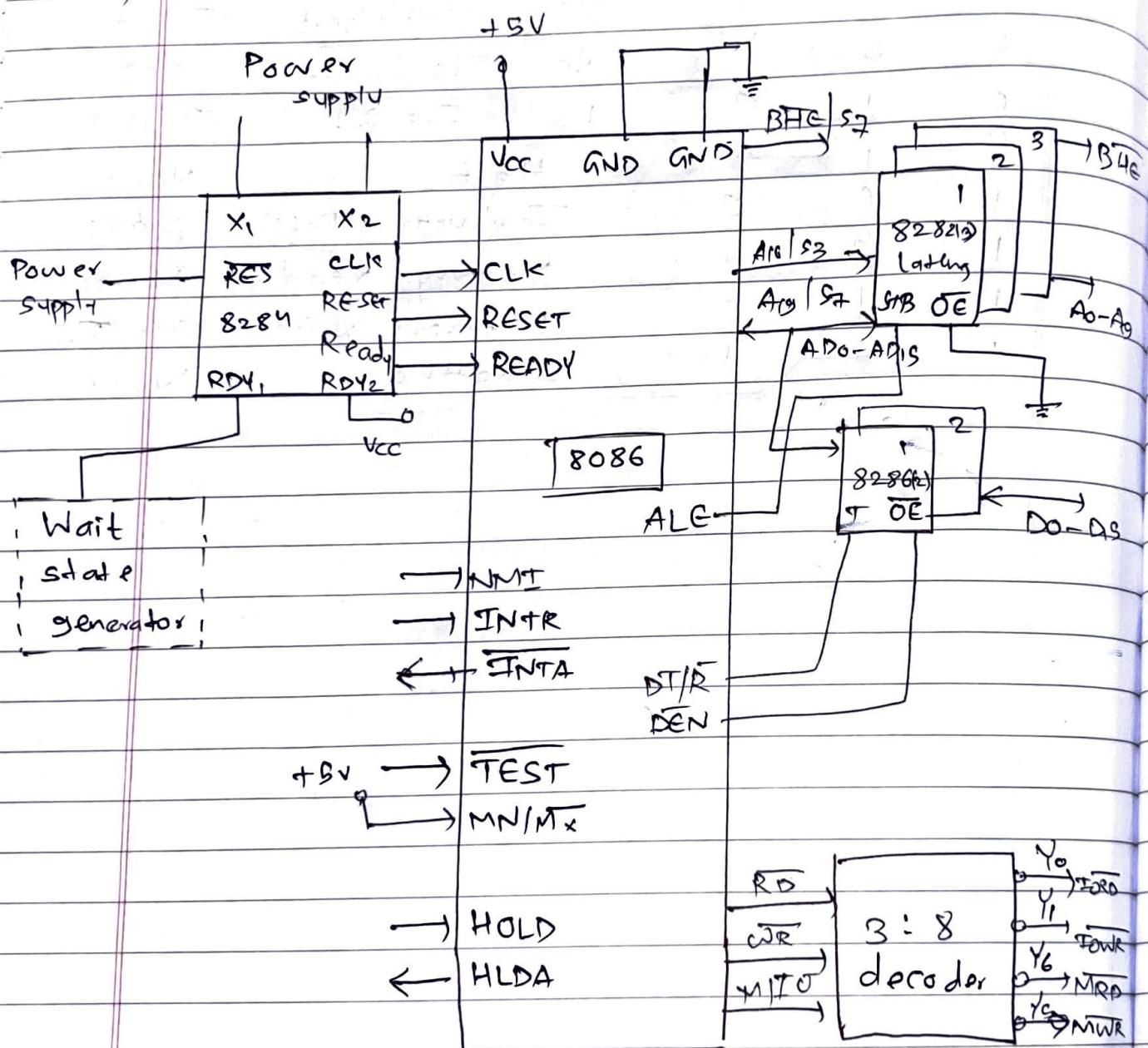
S<sub>2</sub>    S<sub>1</sub>    S<sub>0</sub>    Processor State    8288 Active Output

0	0	0	INT Acknowledgment	INTA
0	0	1	Read I/O Port	IORC
0	1	0	Write I/O Port	IOWC
0	1	1	Halt	None
1	0	0	Instruction Fetch	MRDC
1	0	1	Memory Read	MRDC
1	1	0	Memory write	MWPC
1	1	1	Inactive	None



Processor State Organization and Control Unit

Q.6 Explain minimum mode configuration of 8086 MP.

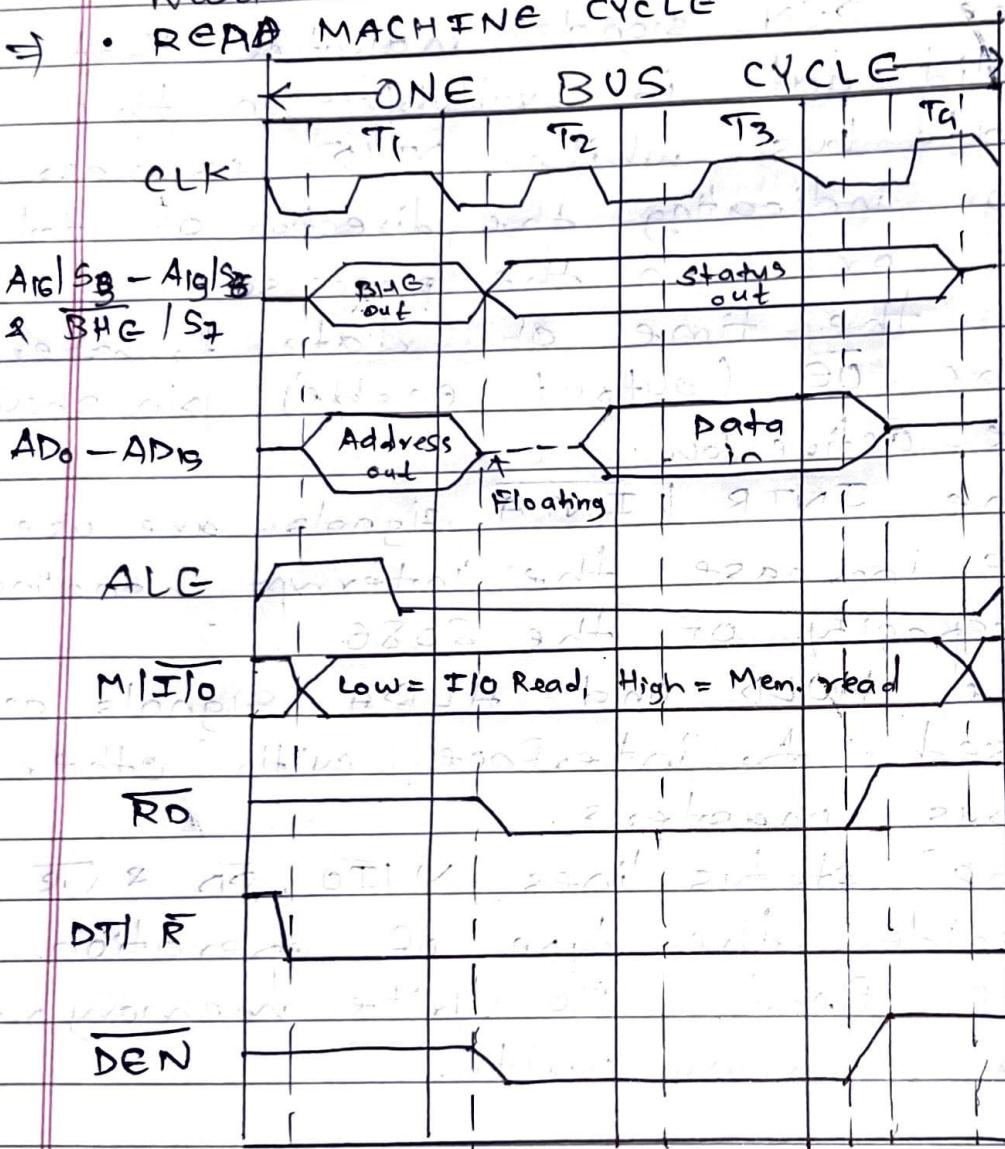


- i) The system in minimum mode contains the support chips such as 8282, 8284 and if necessary, 8286 data buffers.
- ii) These signals AD<sub>0</sub>-AD<sub>15</sub>, A<sub>10</sub>/S<sub>3</sub>, - A<sub>9</sub>/S<sub>2</sub> and BHE/S<sub>7</sub> are multiplexed.

- iii) These signals are demultiplexed by external latches and the ALE signal.
- iv) The DEN signals indicates that valid data is available on the data bus, while DT/R is responsible for indicating the direction of data to or from the processor.
- v) At the time of data transfer the OE (output enable) pin should be active low.
- vi) The INTR | INTA signals are used to increase the interrupt handling capacity of the 8086.
- vii) The HOLD and HLDA signals are used to interface with other bus masters.
- viii) The status lines M1<sup>o</sup>, RD & WR will decide the type of operation I<sup>o</sup> Read, I<sup>o</sup> Write, memory read or memory write.

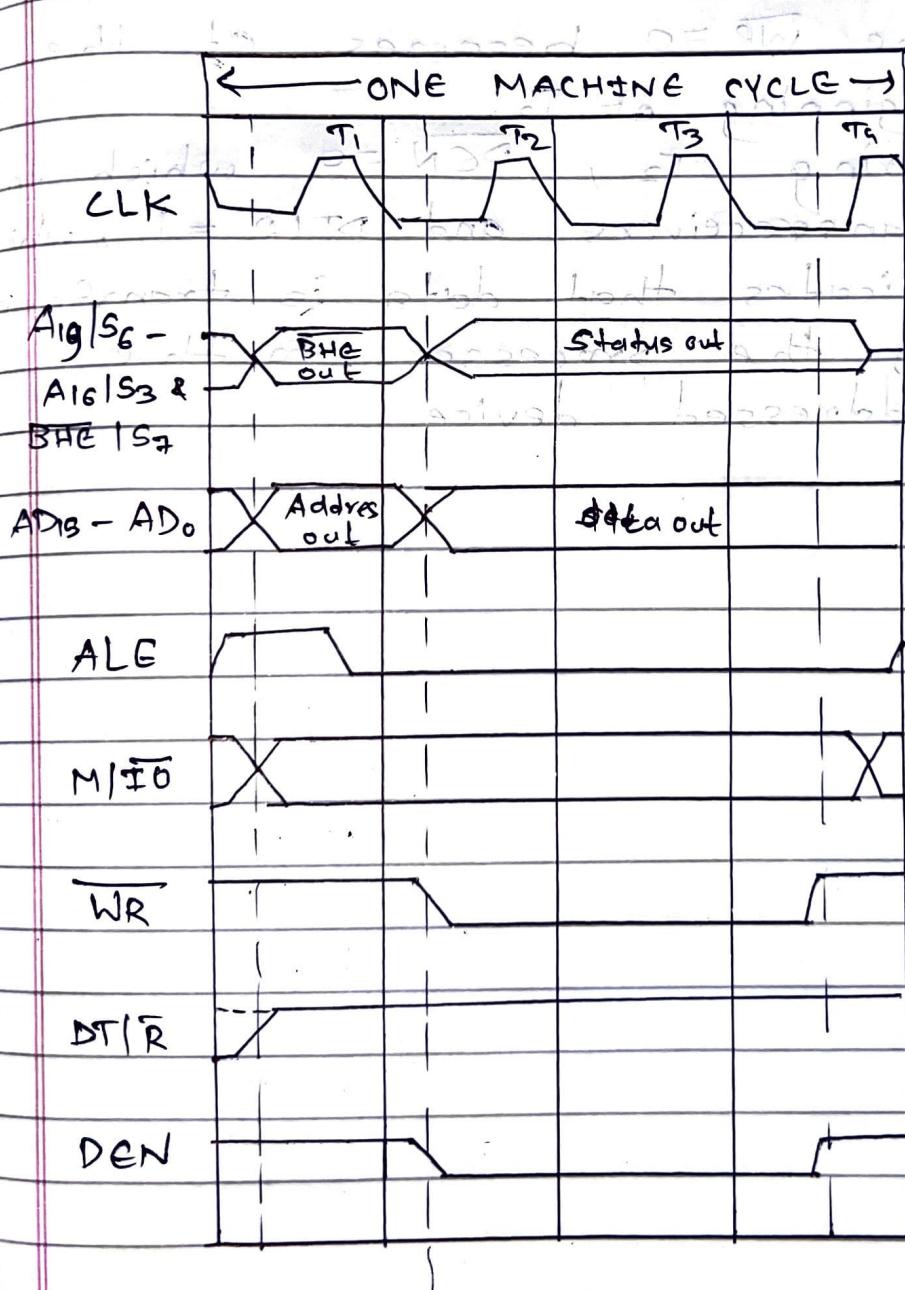
Q.7 Draw and explain memory read machine cycle timing graph diagram in minimum mode of 8086.

⇒ • READ MACHINE CYCLE



- i) All processor bus cycle is of at least 4 T-state (T<sub>1</sub>, T<sub>2</sub>, T<sub>3</sub>, T<sub>4</sub>).
- ii) The address is given by processor in the T<sub>1</sub> state.
- iii) In T<sub>2</sub>, bus cycle is tristated for changing the direction of bus.
- iv) The data transfer takes place between T<sub>3</sub> and T<sub>4</sub>.

- v) At  $T_1$ ,  $\text{--state ALE} = 1$ , this indicates that valid address is latched on the address bus and  $M/I\bar{O} = 1$  which indicates that memory operation is in progress.
- vi) When  $\overline{RD} = 0$ , indicates that valid address is present on data bus.
- vii) During  $T_2$   $\overline{DEN} = 0$ , which enables transreceivers and  $DT/R = 0$  (which indicates that data is received).



- i) At state  $T_1$ ,  $ALE = 1$  indicates that a valid address is latched on the address bus and  $M/I/O = 1$ , which indicates the memory operation is in progress.
- ii) In  $T_2$ , processor sends the data to be written to the addressed location.
- iii) The data is buffered on the bus until the middle of  $T_3$  state.
- iv) The  $\overline{WR} = 0$  becomes at the beginning of  $T_2$ .
- v) During  $T_2$ ,  $\overline{DEN} = 0$  which enables transceivers and  $DT/R = 1$ , which indicates that data is transferred by the processor to the addressed device.

16 KB = 03FFF H

32 KB = 07FFF H

64 KB = OFFFH

128 KB = 1FFFFH

classmate

Date \_\_\_\_\_

Page \_\_\_\_\_

Q.9 Design 8086 with following specification  
1) 32 KB EPROM using 16 KB chip  
2) 64 KB SRAM using 32 KB chip.  
⇒ Soln:-

2) 64 KB SRAM using 32 KB chip.

Step 1:

Total SRAM required = 64 KB

chip sizes available = 32 KB

$$\therefore \text{No. of chips required} = \frac{\text{total available}}{32} = 2$$

$$\therefore \text{No. of set required} = \frac{\text{chip required}}{2} = 1$$

Step 2:

Starting address of SRAM = 00000H

$$\therefore \text{SET SIZE} = \text{available} \times \text{no. of chip}$$
$$= 32 \times 2$$
$$= 64 KB = 0FFFFH$$

$$\therefore \text{Ending address of SRAM} = \text{SA} + \text{SET size}$$
$$= 00000 + 0FFFF$$
$$= 0FFFFH$$

EVEN	ODD
SA EA	00000H 0FFFFH
	(+1) → 00001H (-1) ← 0FFFFH

• Step 2: 32KB EPROM using 16KB chip.

Total EPROM required = 32KB

Chip size available = 16KB.

∴ No. of chips required =  $\frac{\text{total}}{\text{available}}$

$$\frac{\text{Total EPROM required}}{\text{chip size available}} = \frac{32}{16} = 2.$$

∴ No. of sets required =  $\frac{\text{chip required}}{\text{chip size}}$

Ending address of EPROM = FFFFFFFH

∴ SET SIZE = available  $\times$  no. of chips

$$\text{Available size} = 16 \times 2 = 32 \text{ KB}$$

Initial address = 07FFFH

∴ STARTING ADDR. = EA - SET SIZE

$$= FFFFFF - 07FFF$$

$$= F8000H$$

	EVEN	ODD
SA	F8000 H	F8001 H
EA	FFFFE H	FFFFF H

### Step 3: Memory Map

		A <sub>9</sub> A <sub>8</sub> A <sub>7</sub> A <sub>6</sub>	A <sub>5</sub> A <sub>4</sub> A <sub>3</sub> A <sub>2</sub>	A <sub>11</sub> A <sub>10</sub> A <sub>9</sub> A <sub>8</sub>	A <sub>7</sub> A <sub>6</sub> A <sub>5</sub> A <sub>4</sub>	A <sub>3</sub> A <sub>2</sub> A <sub>1</sub> A <sub>0</sub>	
E		F8000	1111	1000	0000	0000	0000
P	EB	FFFFE	11111	1111	1111	1111	1110
R		EA					
O	OB	SA	1111	1000	0000	0000	0001
M		F8001					
		EA					
		FFFFF	1111	1111	1111	1111	1111
S	EB	SA					
R		00000	0000	0000	0000	0000	0000
A		EA					
M		0FFF	0000	1111	1111	1111	1110
OB		SA					
		00001	0000	0000	0000	0000	0001
		EA					
		0FFF	0000	1111	1111	1111	1111

$$ICG \frac{512}{8} \rightarrow \frac{512}{8} = 64$$

classmate

Date \_\_\_\_\_

Page \_\_\_\_\_

