

# Module-1

Prepared by EBIN P.M APCSE, MEA Engineering College,  
Malappuram

## Introduction to 8086 microprocessor :-

### A: Microprocessor:-

- \* It is a program controlled device, which fetches (from memory), decodes and executes instructions.
- \* It is used as CPU in computers.
- \* Microprocessor consists of an ALU (Arithmetic Logic Unit), an array of registers and a Control Unit (CU).  
ALU performs arithmetical and logical operations on the data received from the memory or an input device. Register array consists of registers and accumulators. The control unit controls the flow of data and instructions within the computer.
- \* Mys follows a sequence : Fetch, Decode and Execute. Initially the instructions are stored in the memory in a sequential order. The Mys fetches those instructions from the memory, then decodes it and executes those instructions till STOP instruction is reached. Later it sends the result in binary to the output port. Between these processes, the register stores the temporarily data and ALU performs the computing Functions.

\* The MP is identified with the size of data, the ALU of the processor can work with at a time.

The 8085 processor has 8-bit ALU, hence it is called 8-bit processor. The 8086 processor has 16-bit ALU. Hence it is called 16-bit processor.

## B: 8086 MP

— \* —

\* First 16 bit processor released by INTEL in the year 1978.

\* 8086 is designed using HMOS technology and now it is manufactured using HMOS III Technology, Contains approximately 29,000 transistors.

\* 8086 is packed in a 40 pin DIP (Dual. In-line package) and requires 5 volt supply.

\* 8086 does not have internal clock circuit. The 8284 clock generator is used to generate the required clock for 8086.

\* The maximum internal clock of 8086 is 5MHz.

\* Other ~~Variations~~ <sup>Versions</sup> of 8086 with different clock rate are

8086 - 1 → 10MHz

8086 - 2 → 8MHz

8086 - 4 → 4MHz.

- \* It consists of powerful instruction set, which provides operations like multiplication and division easily.
- \* It supports 2 modes of operations ie, maximum mode and minimum mode. Maximum mode is suitable for system having multiple processors and minimum mode is suitable for system having a single processor.
- \* It was the first 16-bit processor having 16-bit ALU, 16-bit registers, internal data bus and 16-bit external data bus, resulting in faster processing.

### C: Register Organisation of 8086

- \* 8086 contains general purpose and special purpose registers. All registers are 16-bit registers.
- \* General purpose registers used for holding data, variables and intermediate results temporarily.
- \* Special purpose registers are used as segment registers, pointers, index registers or as offset-storage registers for particular addressing modes.

#### (a) General Data Registers

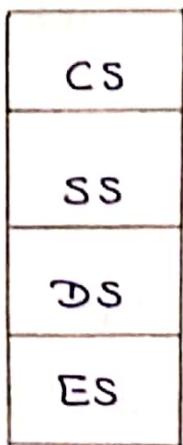
Prepared by EBIN P.M APCSE, MEA Engineering College, Malappuram

AX	AH	AL
BX	BH	BL
CX	CH	CL
DX	DH	DL

Fig:- general data registers.

- \* The registers AX, BX, CX and DX are the general purpose 16-bit registers.
- \* AX is used as 16-bit Accumulator, with the lower 8 bits of AX designated as AL and the higher 8 bits as AH. The letters L and H specify the lower and higher bytes of a particular register. The letter X is used to specify the complete 16-bit register.
- \* The register BX is used as offset-storage for forming Physical addresses. Used as base register. Store starting base address of the memory area within the data segment.
- \* CX is used as a default counter. (store loop counter)
- \* DX is used as an implicit operand or destination in case of a few instructions. (bold I/O port addresses for I/O instruction)

(b) Segment registers:-



segment is a logical unit of memory. minimum size of segment can be 16 bytes and maximum can be 64 KB.

Fig: Segment registers

- \* CS - Code segment- register
- \* DS - Data segment- register
- \* ES - Extra segment- register
- \* SS - Stack segment- register.

Segmentation :- Segmentation is the process in which the main memory of the computer is logically divided into different segments and each segment has its own base address. It is basically used to enhance the speed of execution of the computer system, so that the processor is able to fetch and execute the data from the memory easily and fast.

- \* CS contains the segment- address for the code, where the executable program is stored.
- \* DS points to the data segment of the memory where the data is stored.
- \* ES <sup>points to</sup> another data segment of the memory. It also contains data.

- \* SS is used for addressing stack segment of the memory. Stack segment is used to store stack data.
- \* The segment register hold the upper 16 bits of the base addresses starting addresses of 4 memory segments that 8086 is working with at any particular time.
- \* While addressing any location in the memory bank, the physical address is calculated from 2 parts. The first is segment address second is offset.
- \* Any of the pointers and index registers or BX may contain the offset of the location to be addressed.

### (C) Pointers and Index Registers:-

SP
BP
SI
DI
IP

Pointers  $\Rightarrow$  IP, BP, SP.

Index registers  $\Rightarrow$  SI, DI

SP  $\rightarrow$  stack pointer

BP  $\rightarrow$  Base pointer

IP  $\rightarrow$  Instruction pointer

SI  $\rightarrow$  Source Index

DI  $\rightarrow$  Destination Index.

(Fig: Pointers & index registers)

- \* The pointer IP contains offsets with in the code segment.
- \* BP contains offsets with in the data segment
- \* SP contains offsets with in the stack segment.

- \* Index registers are useful for string manipulations.
- \* SI is used to store the offset of source data in data segment. & DI is used to store offset of destination in data segment or extra segment.

### (d) Flag Register:

- \* Contains the results of Computations in the ALU. It also contains some flag bits to control the CPU operations.

D:

### 8086 Architecture:

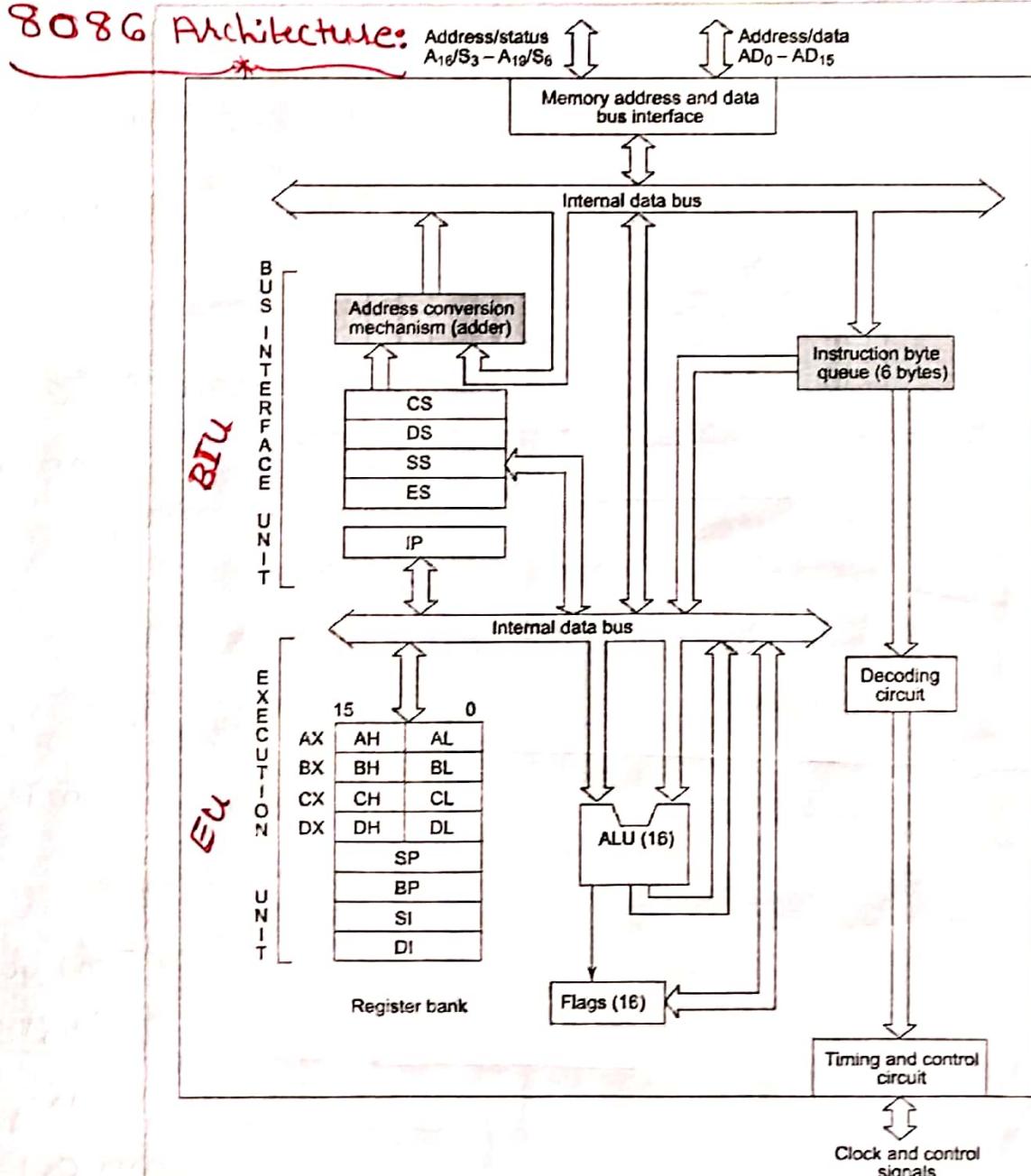


Fig. 1.2 8086 Architecture

- \* 8086 provides 16 bit ALU, 16-bit registers and segmented memory addressing capability.
- \* 8086 has a rich instruction set, powerful interrupt structure. 8086 has Pipelined architecture.
- \* The architecture of 8086 can be internally divided into 2 separate functional units.
  - Bus Interface unit - (BIU)
  - Execution unit - (EU)
- \* The BIU fetches instructions, reads data from memory and I/O ports, writes data to memory and I/O ports.
- \* The BIU contains the circuit for physical address calculations (address generation unit), bus control unit, segment registers, instruction pointer and instruction queue.
- \* The EU executes instructions that have already been fetched by the BIU. The BIU and EU functions independently.
- \* The instruction queue is a FIFO (First in First out) group of registers. The size of queue is 6 bytes. The BIU fetches instruction code from memory and store in queue. The EU fetches instruction codes from the queue.

- \* BIU is responsible for establishing communications with external devices and peripherals including memory via bus.
- \* The complete physical address which is 20-bit long is generated using segment- and offset-registers, each 16 bit-long. The content of segment register is called segment address and offset register content is called offset address.
- \* For generating a physical address, segment address is shifted left bit-wise four times, and to this result, offset address is added, to produce a 20-bit physical address.

Eg: Segment address is 1005H

Offset address is 5555H

Binary of segment address 0001 0000 0000 0101

Shifted by 4 bit position 0001 0000 0000 0101 0000

+

Offset address → 0101 0101 0101 0101

Physical address → 0001 0101 0101 1010 0101

1    5    5    A    5

\* The segment register indicates the base address of a particular segment, while the offset indicates the distance of the required memory location in the segment from the base address.

\* In execution unit, 16-bit ALU performs arithmetic & logical operations. 16-bit flag register contains the results of execution by ALU.

### ⇒ Memory segmentation :

\* To address a specific memory location within a segment, we need an offset address. The offset address is also 16 bit long so that the maximum offset value can be FFFFH, and the maximum size of any segment is thus 64K locations.

\* 2 types of segments are overlapping segments and nonoverlapping segments.

**Overlapping Segment** ⇒ A segment starts at a particular address and its maximum size can go up to 64 KB. But if another segment starts before this 64 KB bytes locations of the first segment, the two segments are said to be overlapping segments.

**Non overlapping segment** ⇒ A segment starts at a particular address and its maximum size can go up to 64 Kilobytes. If another segment starts along with

this 64kbytes location of the first segment, then the two are said to be non-overlapping segments.

**advantages of segmented memory :-**

- \* It provides a powerful memory management mechanism.
- \* Data related or stack related operations can be performed in different segments.
- \* Code related operations can be done in separate code segments.
- \* It allows to processes to easily share data.
- \* It allows to extend the address capability of a processor. i.e., segmentation allows the use of 16-bit registers to give an addressing capability of 1MB. Without segmentation, it would require 20 bit registers.

⇒ **Flag Register:**

- \* A flag is a flip flop used to store the information about the status of the processor and the status of the instruction executed most recently.

8086 has 9 Flag.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
X	X	X	X	0	D	I	T	S	Z	X	Ac	X	P	X	Cy

Fig: Flag register of 8086.

1. **O - Overflow Flag**  $\Rightarrow$  This flag is set, if an overflow occurs.
2. **D - Direction Flag**  $\Rightarrow$  Used by string manipulation instructions. If this Flag bit is "0", the string is processed beginning from the lowest address to the highest address. (ie, autoincrementing mode). Otherwise the string is processed from the highest address towards the lowest address (ie, auto-decrementing mode).
3. **I - Interrupt Flag**:  $\Rightarrow$  If this flag is set, the maskable interrupts are recognised by the CPU, otherwise they are ignored.
4. **T - Trap Flag**  $\Rightarrow$  If this flag is set, the Trap interrupt is generated after execution of each instruction.
5. **S - Sign Flag**  $\Rightarrow$  This flag is set, when the result of any computation is negative.
6. **Z - Zero Flag**  $\Rightarrow$  This flag is set, the result of the computation is zero or comparison performed by the previous instructions is zero.
7. **AC**  $\Rightarrow$  **Auxiliary Carry Flag**  $\Rightarrow$  This is set, if there is a carry from the lowest nibble.

8. P - Parity Flag  $\rightarrow$  This flag is set to '1', if the result has even parity and parity flag is cleared to "zero". For odd parity of the result.
9. Cy - Carry Flag  $\rightarrow$  Is set, if there is a carry from addition or borrow from subtraction.

## E: Signal descriptions of 8086

- \* The microprocessor 8086 is a 16-bit CPU available in three clock rates. i.e., 5, 8 and 10MHz; and it operates in a single processor and multiprocessor configurations, to achieve high performance.

		Maximum mode	Minimum mode
GND	1	40	VCC
AD <sub>14</sub>	2	39	AD <sub>15</sub>
AD <sub>13</sub>	3	38	A <sub>16/S<sub>3</sub></sub>
AD <sub>12</sub>	4	37	A <sub>17/S<sub>4</sub></sub>
AD <sub>11</sub>	5	36	A <sub>18/S<sub>5</sub></sub>
AD <sub>10</sub>	6	35	A <sub>19/S<sub>6</sub></sub>
AD <sub>9</sub>	7	34	BHE/S <sub>7</sub>
AD <sub>8</sub>	8	33	MN/MX
AD <sub>7</sub>	9	32	RD
AD <sub>6</sub>	10	31	RD/GT <sub>0</sub> (HOLD)
AD <sub>5</sub>	11	30	RD/GT <sub>1</sub> (HLDA)
AD <sub>4</sub>	12	29	LOCK (WR)
AD <sub>3</sub>	13	28	S <sub>2</sub> (M/R)
AD <sub>2</sub>	14	27	S <sub>1</sub> (DT/R)
AD <sub>1</sub>	15	26	S <sub>0</sub> (DEN)
AD <sub>0</sub>	16	25	QS <sub>0</sub> (ALE)
NMI	17	24	QS <sub>1</sub> (INTA)
INTR	18	23	TEST
CLK	19	22	READY
GND	20	21	RESET

Fig. 1.5 Pin Configuration of 8086

- \* The signals can be categorised into 3 groups.
  - Signals having common Functions in minimum as well as maximum mode
  - Signals having special Functions for minimum mode
  - Signals having special functions for maximum mode.

\*  $AD_{15} - AD_0 \rightarrow$  (Address / data bus) These are 16 address data bus.  $AD_0$  to  $AD_7$  carries low order byte data and  $AD_8$  to  $AD_{15}$  carries higher order byte data. During the first clock cycle, it carries 16-bit address and after that it carries 16-bit data.

\*  $A_{19}/S_0, A_{18}/S_5, A_{17}/S_4, A_{16}/S_3 \rightarrow$  (Address / status bus). These are the 4 address / status buses.

During the first clock cycle, it carries 4-bit address and later it carries status signals.

$A_{17}/S_4$	$A_{16}/S_3$	Function
0	0	Extra segment access
0	1	Stack segment access
1	0	Code segment access
1	1	Data segment access

- \* **BHE / S7**  $\Rightarrow$  (Bus High Enable / status) It is used to indicate the transfer of data using data bus  $D_{15} - D_8$ . This signal is low during the first clock cycle, thereafter it is active.
- \* **RD**  $\Rightarrow$  (Read) used to read signal for read operation. It is an output signal. It is active when low.
- \* **READY**  $\Rightarrow$  This is the acknowledgement from the slow devices or memory that they have completed the data transfer. It is an active high (1) signal.
- \* **INTR**  $\Rightarrow$  (Interrupt Request) It is an interrupt request signal, which is sampled during the last clock cycle of each instruction to determine if the processor considered this as an interrupt or not. This signal is active high and internally synchronized.
- \* **TEST**  $\Rightarrow$  This input is examined by a "WAIT" instruction. If the TEST pin goes low (0), execution will continue; else the processor remains in an idle state.
- \* **NMI**  $\Rightarrow$  (Non-maskable Interrupt) This is an edge-triggered input which causes a Type 2 interrupt. NMI is non-maskable internally by software. This input is internally synchronized.

- \* **RESET**  $\Rightarrow$  It is used to restart the execution. It causes the processor to immediately terminate its present activity. This signal is active high and must be active for at least 4 clock cycles. RESET is also internally synchronized.
- \* **CLK**  $\Rightarrow$  (clock input) clock input provides the basic timing for processor operation and bus control activity. It is an asymmetric square wave with a 33% duty cycle.
- \* **Vcc**  $\Rightarrow$  power supply (+5VDC)
- \* **GND**  $\Rightarrow$  Ground for the internal circuit
- \* **MN/MX**  $\Rightarrow$  (minimum / maximum). This pin signal indicates which mode of the processor will operate in.
- \* **M/I<sub>O</sub>**  $\Rightarrow$  (memory / I<sub>O</sub>) This signal is used to distinguish between memory and I/O operations. When it is low, it indicates the CPU is having an I/O operation. When it is high, it indicates that the CPU is having a memory operation.
- \* **INTA**  $\Rightarrow$  (Interrupt Acknowledge) When the microprocessor receives this signal, it acknowledges the interrupt. When it goes low, it means that the processor has accepted the interrupt.

- \* **ALE**  $\Rightarrow$  (Address Latch Enable) - This output signal indicate the availability of the valid address on the address / data line. This signal is active high. A positive pulse is generated each time the processor begins any operation.
- \* **DT/R**  $\Rightarrow$  (Data Transmit/Receive) It decides the direction of data flow through the transreceivers (bidirectional buffers). The signal is high when the data is transmitted and the signal is low when the processor is receiving data.
 

High  $\rightarrow$  DT  
Low  $\rightarrow$  Receiving
- \* **DEN**  $\Rightarrow$  (Data Enable) . This signal indicate the availability of valid data over the address / data lines. It is used to enable the transreceivers to separate data from the address / data bus.
- \* **HOLD**  $\Rightarrow$  This signal indicates to the processor that external devices are requesting to access the address / data buses.
- \* **HLDA**  $\Rightarrow$  (Hold Acknowledge) This signal acknowledges the HOLD signal.
- \*  **$S_2, S_1, S_0$**   $\Rightarrow$  (status lines) These are the status lines (signals) that provides the status of operation, which is used by the bus controller 8288 to generate memory and I/O control signals. Following table shows the status lines.

$\bar{S}_a$	$\bar{S}_1$	$\bar{S}_0$	Indication
0	0	0	Interrupt acknowledge
0	0	1	Read I/O port
0	1	0	Write I/O port
0	1	1	Halt
1	0	0	Code Access
1	0	1	Read memory
1	1	0	Write memory
1	1	1	Passive / inactive

\* LOCK  $\Rightarrow$  It is an active low pin. This indicate that other system bus masters will be prevented from joining the system bus, while the lock signal is low.

\* QS<sub>1</sub>, QS<sub>0</sub>  $\Rightarrow$  (Queue status). These signals indicate the status of the internal 8086 instruction queue according to the table shown below.

QS <sub>1</sub>	QS <sub>0</sub>	Status
0	0	No operation
0	1	First byte of opcode from the queue
1	0	Empty queue
1	1	Subsequent byte from the queue.

- \*  $\overline{RQ}/\overline{UIT_0}, \overline{RQ}/\overline{UIT_1} \Rightarrow$  (Request / Urant). It is used by other processors requesting the you to release the system bus. When the signal is received by CPU, then it sends acknowledgement.  $RQ/UIT_0$  has a higher priority than  $RQ/UIT_1$ . Each of the pin is bidirectional.

## F: Minimum Mode 8086 System and Timings

- \* 8086 is operated in minimum mode by strapping its MN/MX pin to logic 1.
  - \* In this mode, all the control signals are given out by the microprocessor chip itself. There is a single microprocessor in the minimum mode system.
  - \* The remaining components in the systems are latches, transceivers, clock generators, memory and I/O devices.
  - \* It has 20 address lines and 16 data lines, the 8086 CPU requires three octal address latches and two octal data buffers for the complete address and data separation.
  - \* Latches:  $\Rightarrow$  They are generally buffered output-type flip-flops like 74LS373 or 8282. They are used for separating the valid address from the multiplexed address/data signals and are controlled by the ALE signals generated by 8086.
- separating address from data bus

Separate Valid data  
from address/dataline

\* Trans-receivers  $\Rightarrow$  They are the bidirectional buffers and sometimes they are called as data amplifiers. They are required to separate the valid data from the time multiplexed address / data signals. They are controlled by 2 signals namely,  $\overline{DEN}$  and  $\overline{DT/R}$ . The  $\overline{DEN}$  signal indicate the availability of valid data over the address / data lines. The  $\overline{DT/R}$  signal indicate direction of data, ie from or to the processor.

\* Usually, EPROM are used for monitor storage, while RAM for users program storage.

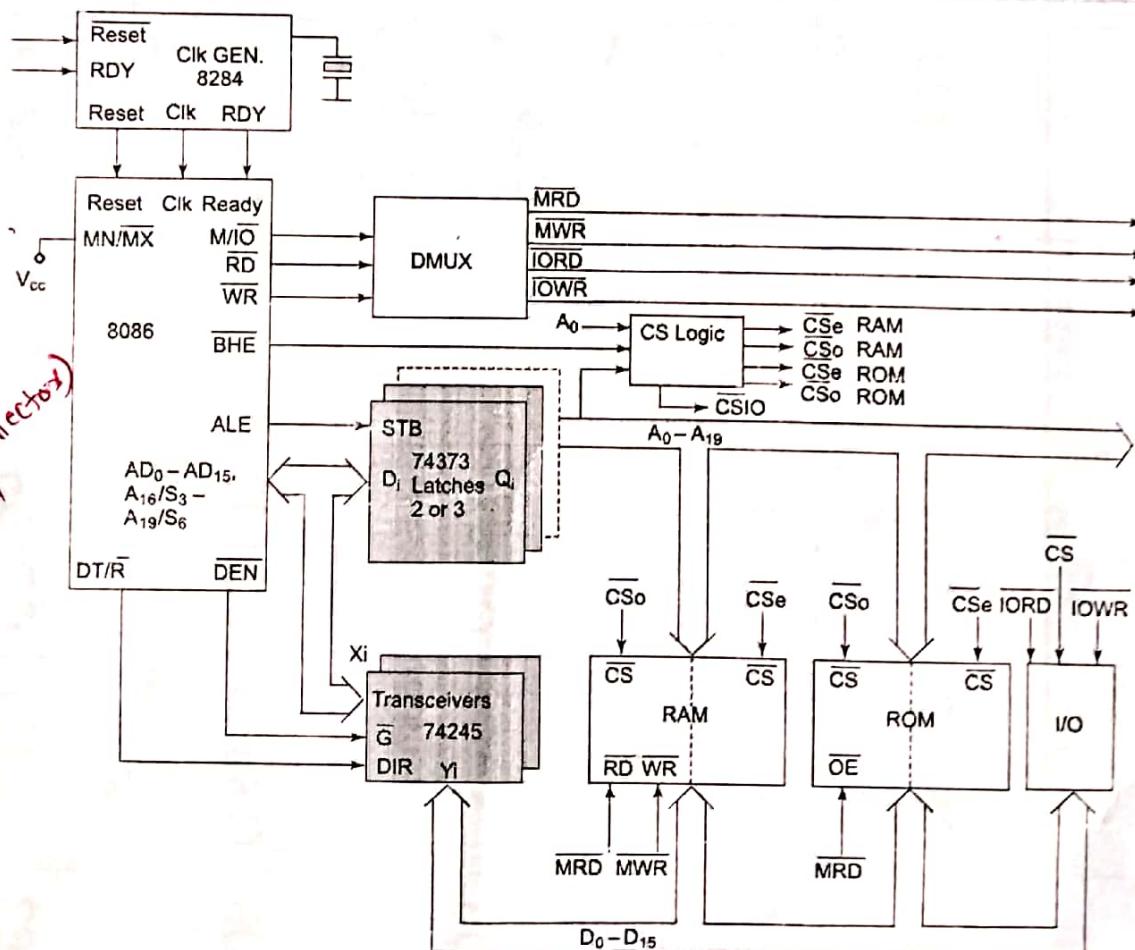


Fig. 1.13 Minimum Mode 8086 System

\* Read Cycle timing diagram for minimum mode:-

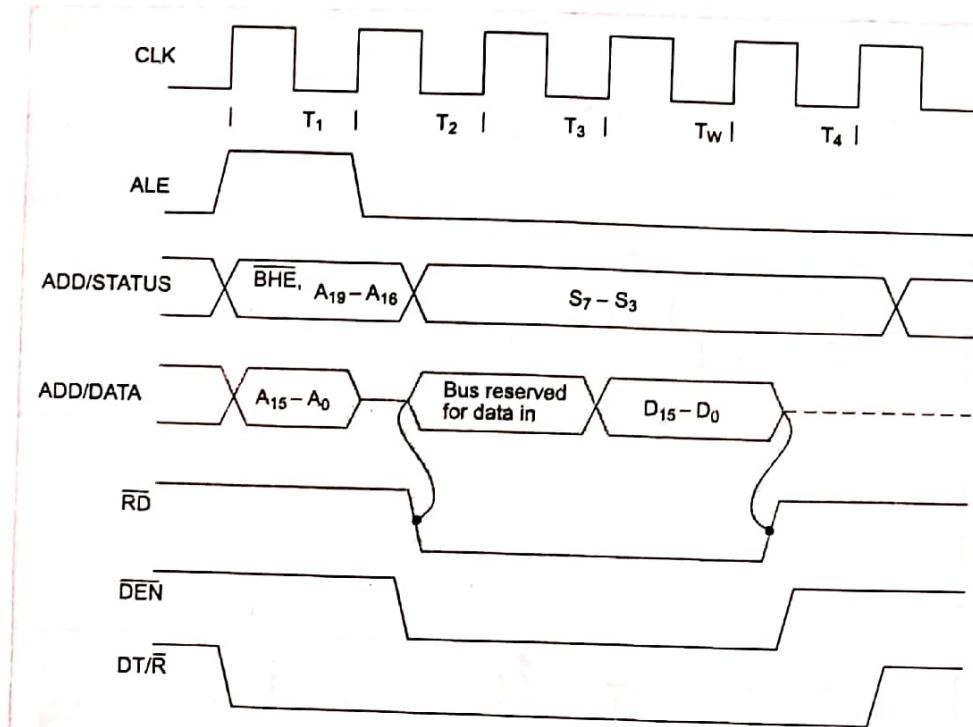


Fig. 1.14(a) Read Cycle Timing Diagram for Minimum Mode

T<sub>1</sub>, T<sub>2</sub>, T<sub>3</sub>, T<sub>4</sub> are clock pulse. one timing cycle having 4 clock pulse including waiting time (T<sub>W</sub>).

During T<sub>1</sub> clock pulse, address latch enable become high. So whenever ALE is high during T<sub>1</sub> cycle, then we can send the address and data on to the buses.

During T<sub>1</sub> clock pulse, address will be active, during the remaining pulse, the status will be active.

During first clock pulse, address is placed on the address bus, after that the data will be placed on the data bus (D<sub>15</sub>-D<sub>0</sub>). Here AD<sub>0</sub>-AD<sub>15</sub> is multiplexed as A<sub>15</sub>-A<sub>0</sub> and D<sub>0</sub>-D<sub>15</sub>. First address will be placed, after that the bus is reserved for data in.

that is ie,  $\overline{D_o}$ -Dis- during remaining clock pulse.

$\overline{RD} \Rightarrow (\overline{RD} = \overline{0} = 1(\text{active}))$  During T<sub>2</sub> to T<sub>4</sub> the data bus is active. So we are performing read operation.

$\overline{DEN} \Rightarrow (\overline{DEN} = \overline{0} = 1(\text{active}))$  Data Enable is also active during T<sub>2</sub> to T<sub>4</sub> clock pulse.

$DT/\overline{R} \Rightarrow$  Reading means receiving the data (R) and data transmitting (DT) is Writing operation.

To make  $\overline{R}$  as active (1) we have to put 0.

The M/ $\overline{IO}$ ,  $\overline{RD}$  and  $\overline{WR}$  signals indicate the type of data transfer.

M/ $\overline{IO}$	$\overline{RD}$	$\overline{WR}$	Transfer type
0	0	1	I/O read
0	1	0	I/O write
1	0	1	memory read
1	1	0	memory write

\* Write cycle timing diagram for minimum mode:-

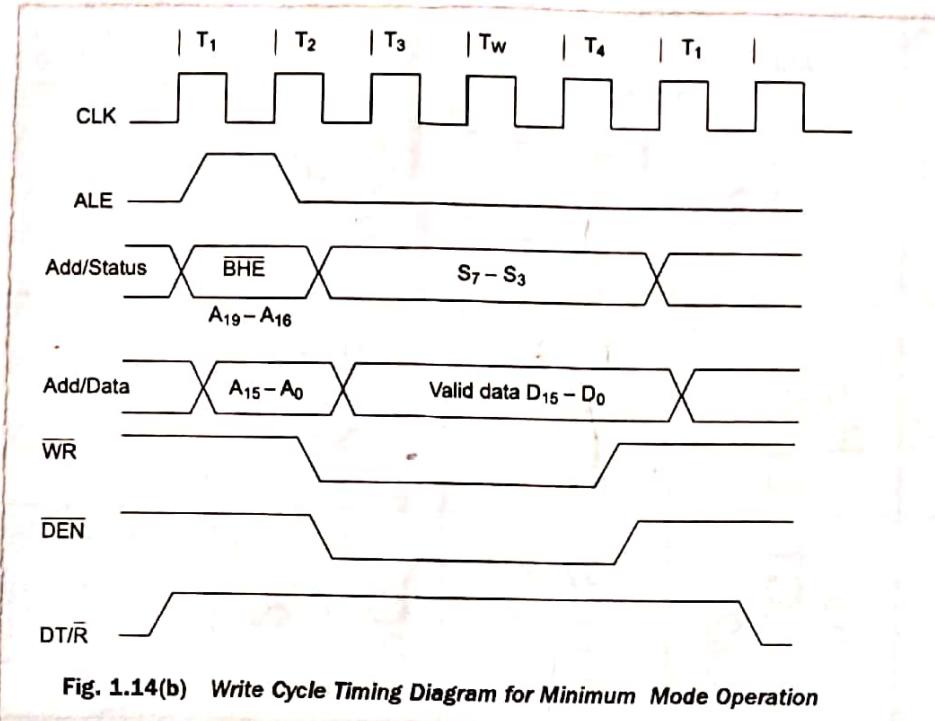


Fig. 1.14(b) Write Cycle Timing Diagram for Minimum Mode Operation

Whatever the valid data that is placed on the data bus that we have to be write to the external devices.

Q1: Maximum mode 8086 system:-

- \* In maximum mode 8086 is operated by strapping the MN/MX pin to ground. (ie, MN/MX is tied to logic low).
- \* In this mode the processor derives the status signals  $\overline{S_2}$ ,  $\overline{S_1}$ , and  $\overline{S_0}$ . Another chip called bus controller derives the control signals using this status information.
- \* In the maximum mode there may be more than one microprocessor in the system configuration.

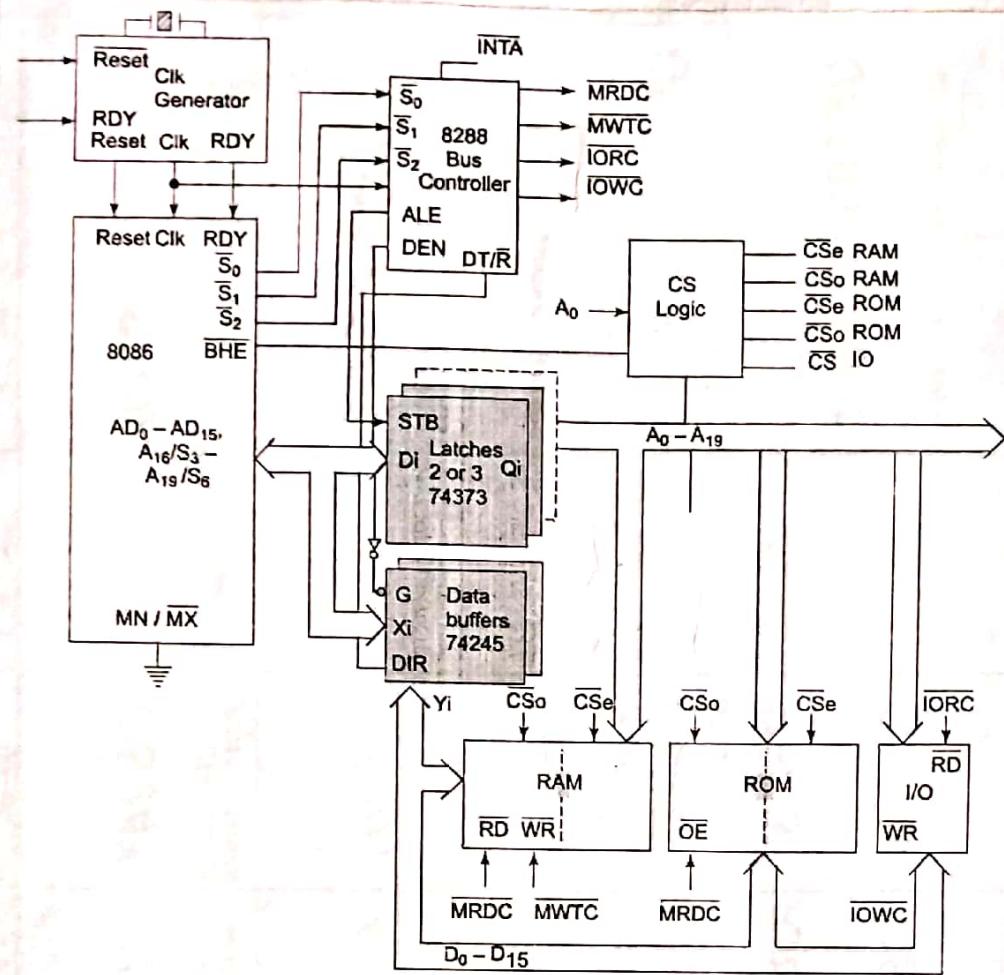
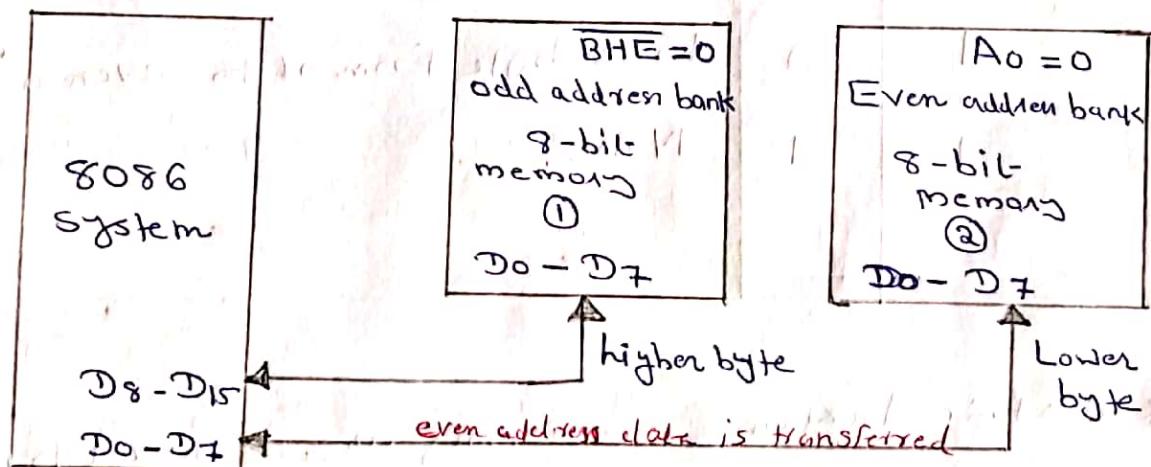


Fig. 1.15 Maximum Mode 8086 System

- \* The basic function of the bus controller chip IC 8288 is to derive control signals like  $\overline{RD}$  and  $\overline{WR}$ ,  $\overline{DEN}$ ,  $\overline{DT/R}$ ,  $\overline{ALE}$  etc; using the information made available by the processor on the status line.
- \* The  $\overline{REN}$  and  $\overline{IOB}$  and  $\overline{CEN}$  pins are specially useful for multiprocessor systems.  $\overline{REN}$  and  $\overline{IOB}$  are generally grounded.  $\overline{CEN}$  pin is usually tied to +5V.
- \* The  $\overline{IORC}$ ,  $\overline{IOWC}$  are IO read command and IO write command signals respectively.
- The  $\overline{MRDC}$ ,  $\overline{MWTC}$  are memory read command and memory write command signals respectively.

## Memory Organisation

- \* In 8086, the 1MB memory is physically organized as odd bank and even bank, each of 512 Kbytes.
- \* Byte data with even address is transferred on  $D_7 - D_0$  and byte data with odd address is transferred on  $D_{15} - D_8$  bus lines.
- \* The processor provides 2 enable signals, BHE, and A<sub>0</sub> for selection of either even or odd or both the banks.
- \* If the processor fetches a word (consecutive two bytes) from memory, the opcodes and operands are identified by the ~~memory~~ internal decoder circuit.



(Fig:-Physical memory organization)

It is better to locate the word data at an even address. To read or write a complete word from/to memory, if it is located at an even address, only one read or write cycle is required. If the word is located at an odd

address, the first read or write cycle is required for accessing the lower byte while the second one is required for accessing the higher bytes. Thus two bus cycles are required, if a word is located at an odd address.

- \* If 8086 transfers a 16-bit data to or from memory both of the memory banks must be selected for the 16-bit operation. The two signals  $A_0$  and  $BHE$  solve the problem of selection of appropriate memory bank as presented in the following table.

$BHE$	$A_0$	Indication
0	0	Whole word (2 bytes)
0	1	Upper byte from or to odd address
1	0	Lower byte from or to even address
1	1	None

- \* Certain locations in memory are reserved for specific CPU operations. The location  $FFFF0H \xleftarrow{\text{zero}}$  to  $FFFFF_H$  are reserved for jump to initialisation programme and I/O processor initialisation.

The location  $00000H$  to  $003FF_H$  are reserved for interrupt vector table.

## Comparison between 8086 & 8088

	8086	8088
1.	It has 16-bit data bus.	It has 8-bit data bus.
2.	8086 can read / write 8/16 bit data at a time.	8088 can only do so for 8-bit data.
3.	8086 memory space is organized as two 512KB (2 * 512KB = 1MB) banks.	In 8088 it is implemented as a single 1MB bank, with each byte of the bank being addressed.
4.	8086 can operate at three clock speeds i.e., 5MHz, 8MHz, 10MHz.	8088 is available in only two clock speeds i.e., 5MHz and 8MHz.
5.	Instruction queue is 6-bit long.	4-bit long.
6.	8086 has Bus High Enable (BHE) pin.	On 8088, this pin is replaced by Status Output (SSO) since it can send or write only 8-bit data.
7.	It draws a maximum supply current of 360mA.	It draws a maximum supply current of 340mA.

## University Questions

1. List the registers used in 8086 microprocessor? (3m)
2. Describe Function of the following signals of 8086  
INTR, READY, HOLD (3m)
3. Draw & explain the internal block diagram of 8086 (9m)
4. What are the different Flag bits available within the Flag register of 8086? (3m)
5. With the help of timing diagram show the transition of control signals involved in the I/O read operation of 8086 in minimum mode? (3m)
6. Explain the physical and logical memory organization of 8086? (9m)
7. How does the 8086 processor access a word from an odd memory location? (3m)
8. Draw memory read timing diagram of 8086 in minimum mode. Describe the status of the relevant signals during each clock period? (9m)
9. With a neat diagram describe how 8086 memory is organised at physical level? (5m)
10. List any 6 Features of 8088 microprocessor
11. What are the different information conveyed by the Queue status Signals Q50 and Q51 of 8086 in max mode? (4m)