

---

## Module I

Evolution of microprocessors, 8086 Microprocessor - Architecture and signals, Memory organisation, Minimum and maximum mode of operation, Minimum mode Timing Diagram. Comparison of 8086 and 8088.

### What is Microprocessor?

A single chip processor is called **microprocessor**. It is also called Central Processing Unit (CPU). It is the brain of a computer system. It is also used in many electronic devices.

Microprocessor is a semiconductor device consisting of electronic logic circuits, capable of performing computing functions and making decisions. A computer built with a microprocessor is called micro computer.

### **Examples**

Intel 8086, Intel 80286, Pentium II

A microprocessor in which most instructions operate on n-bit data and its registers are of n-bit size is said to be a n-bit microprocessor.

Eg:- 8 bit microprocessor can operate 8 bit data..

### Evolution of microprocessors

#### 4-bit Microprocessors

The first microprocessor was introduced in 1971 by Intel Corp. It was named Intel 4004 as it was a 4 bit processor. It was a processor on a single chip. It could perform simple arithmetic and logic operations such as addition, subtraction, boolean AND and boolean OR. Intel introduced the enhanced version of 4004, the 4040.

#### 8-bit Microprocessors

The first 8 bit microprocessor which could perform arithmetic and logic operations on 8 bit words was introduced in 1973 again by Intel. This was Intel 8008 and was later followed by an improved version, Intel 8088.

#### 16-bit Microprocessors

The 8-bit processors were followed by 16 bit processors. They are Intel 8086 and 80286.

#### 32-bit Microprocessors

---

The 32 bit microprocessors were introduced by several companies but the most popular one is Intel 80386.

### Pentium Series

Instead of 80586, Intel came out with a new processor namely Pentium processor. Its performance is closer to RISC performance. Pentium was followed by Pentium Pro CPU. Pentium Pro allows multiple CPUs in a single system in order to achieve multiprocessing. The MMX extension was added to Pentium Pro and the result was Pentium II. The low cost version of Pentium II is celeron.

The Pentium III provided high performance floating point operations for certain types of computations by using the SIMD extensions to the instruction set. These new instructions makes the Pentium III faster than high-end RISC CPUs.

Interestingly Pentium IV could not execute code faster than the Pentium III when running at the same clock frequency. So Pentium IV had to speed up by executing at a much higher clock frequency.

NAME	YEAR	TRANSISTORS	DATA WIDTH	CLOCK SPEED
8080	1974	6,000	8 bits	2 MHz
8085	1976	6,500	8 bits	5 MHz
8086	1978	29,000	16 bits	5 MHz
8088	1979	29,000	8 bits	5 MHz
80286	1982	134,000	16 bits	6 MHz
80386	1985	275,000	32 bits	16 MHz
80486	1989	1,200,000	32 bits	25 MHz
PENTIUM	1993	3,100,000	32/64 bits	60 MHz
PENTIUM II	1997	7,500,000	64 bits	233 MHz
PENTIUM III	1999	9,500,000	64 bits	450 MHz
PENTIUM IV	2000	42,000,000	64 bits	1.5 GHz

### 8086 Microprocessor

8086 Microprocessor is an enhanced version of 8085Microprocessor that was designed by Intel in 1976. It is a 16-bit Microprocessor having **20 address lines and 16 data lines that provides up to 1MB storage**. It consists of powerful instruction set, which provides operations like multiplication and division easily.

**It supports two modes of operation**, i.e. 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.

---

## Features of 8086

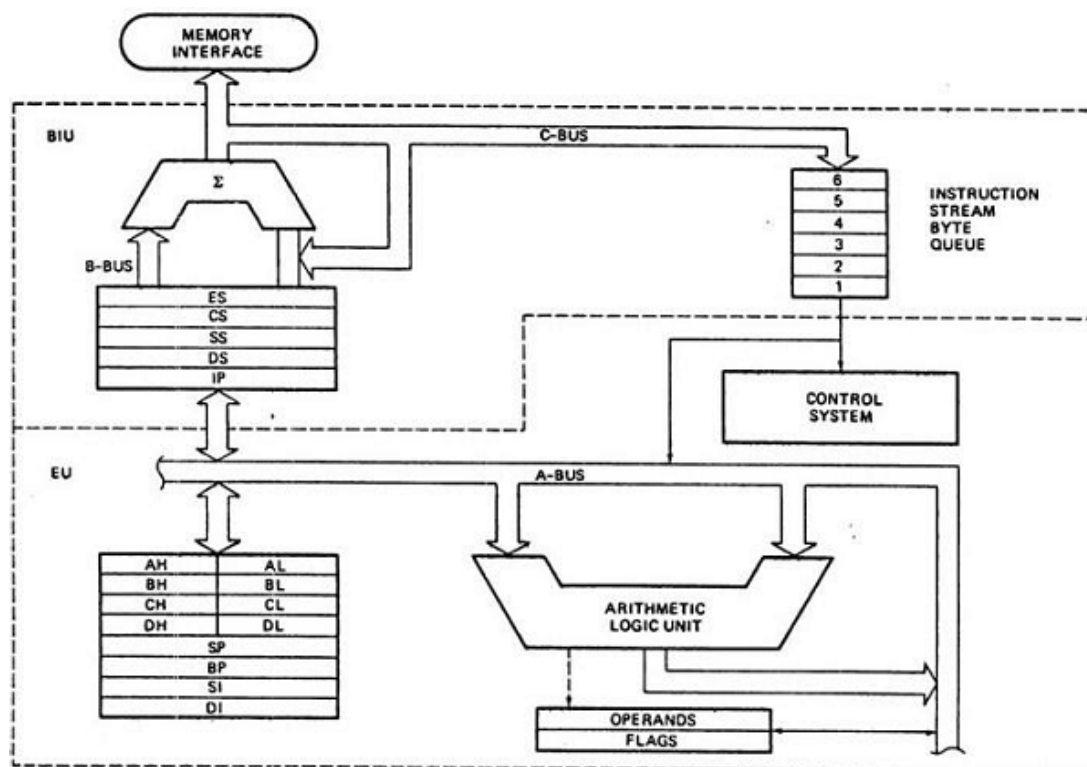
The features of 8086 microprocessor are as follows –

- It has a 20 bit address bus can access upto  $2^{20}$  memory location
- Support upto 64 KB I/O Ports
- Word size is 16 bits
- 40 pin dual in line package
- Address ranges from 00000H to FFFFFH
- Memory is addressable- every byte has a separate address
- It has an instruction queue, which is capable of storing six instruction bytes from the memory resulting in faster processing.
- 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.
- It is available in 3 versions based on the frequency of operation –
  - o 8086 → 5MHz
  - o 8086-2 → 8MHz
  - o 8086-1 → 10 MHz
- It uses two stages of pipelining, i.e. Fetch Stage and Execute Stage, which improves performance.
  - o Fetch stage can prefetch up to 6 bytes of instructions and stores them in the queue.
  - o Execute stage executes these instructions.
- It has 256 vectored interrupts.
- It consists of 29,000 transistors.

## Architecture of 8086

The following diagram depicts the architecture of a 8086 Microprocessor –

8086 Microprocessor is divided into two functional units, i.e., **EU** (Execution Unit) and **BIU** (Bus Interface Unit).



### **BIU (Bus Interface Unit)**

BIU takes care of all data and addresses transfers on the buses. Its functions are **sending addresses, fetching instructions from the memory, reading data from the ports and the memory as well as writing data to the ports and the memory.**

EU and BIU are connected with the Internal Bus.

It has the following functional parts –

- **Instruction queue** – BIU contains the instruction queue. BIU reads 6 bytes of next instructions from memory and stores them in the instruction queue. After the execution of current instructions, EU reads next from instruction queue resulting in increased execution speed.
  - Fetching the next instruction while the current instruction executes is called **pipelining**.
- **Segment register** – BIU has 4 segment registers, i.e. CS, DS, SS& ES. It holds the addresses of instructions and data in memory, which are used by the processor to access memory locations.
  - **CS** – It stands for Code Segment register. It stores the starting addressing of code segment

- 
- o **DS** – It stands for Data Segment register. It stores the starting addressing of data segment
  - o **SS** – It stands for Stack Segment register. It stores the starting addressing of stack segment
  - o **ES** – It stands for Extra Segment register. It stores the starting addressing of extra segment. ES is additional data segment, which is used by the string to hold the extra destination data.
  - **Instruction pointer** – It is a 16-bit register used to hold the address of the next instruction to be executed.

BIU contains a **separate adder** to perform physical address calculation.

### **EU (Execution Unit)**

Execution unit gives instructions to BIU stating from where to fetch the data and then decode and execute those instructions. Its function is to control operations on data using the **instruction decoder** & ALU. EU has no direct connection with system buses as shown in the above figure, it performs operations over data through BIU.

### **ALU**

It handles all arithmetic and logical operations, like +, -, ×, /, OR, AND, NOT operations.

### **Flag Register**

It is a 16-bit register that behaves like a flip-flop, i.e. it changes its status according to the result stored in the accumulator. It has 9 flags and they are divided into 2 groups – Conditional Flags and Control Flags.

### **Conditional Flags**

It represents the result of the last arithmetic or logical instruction executed. Following is the list of conditional flags –

- **Carry flag** – This flag indicates an overflow condition for arithmetic operations.
- **Auxiliary flag** – When an operation is performed at ALU, it results in a carry/borrow from lower nibble (i.e. D0 – D3) to upper nibble (i.e. D4 – D7), then this flag is set, i.e. carry given by D3 bit to D4 is AF flag. The processor uses this flag to perform binary to BCD conversion.
- **Parity flag** – This flag is used to indicate the parity of the result, i.e. when the lower order 8-bits of the result contains even number of 1's, then the Parity Flag is set. For odd number of 1's, the Parity Flag is reset.
- **Zero flag** – This flag is set to 1 when the result of arithmetic or logical operation is zero else it is set to 0.

- 
- **Sign flag** – This flag holds the sign of the result, i.e. when the result of the operation is negative, then the sign flag is set to 1 else set to 0.
  - **Overflow flag** – This flag represents the result when the system capacity is exceeded.

### Control Flags

Control flags controls the operations of the execution unit. Following is the list of control flags –

- **Trap flag** – It is used for single step control and allows the user to execute one instruction at a time for debugging. If it is set, then the program can be run in a single step mode.
- **Interrupt flag** – It is an interrupt enable/disable flag, i.e. used to allow/prohibit the interruption of a program. It is set to 1 for interrupt enabled condition and set to 0 for interrupt disabled condition.
- **Direction flag** – It is used in string operation. As the name suggests when it is set then string bytes are accessed from the higher memory address to the lower memory address and if it is zero then access the string data from lower memory location to higher memory location

### General purpose register

There are 8 general purpose registers, i.e., AH, AL, BH, BL, CH, CL, DH, and DL. These registers can be used individually to store 8-bit data and can be used in pairs to store 16bit data. The valid register pairs are AH and AL, BH and BL, CH and CL, and DH and DL. It is referred to the AX, BX, CX, and DX respectively.

- **AX register** – It is also known as accumulator register. It is used to store operands for arithmetic operations.
- **BX register** – It is used as a base register. It is used to store the starting base address of the memory area within the data segment.
- **CX register** – It is referred to as counter. It is used in loop instruction to store the loop counter.
- **DX register** – This register is used to hold I/O port address for I/O instruction.

### Stack pointer register

It is a 16-bit register, which holds the address from the start of the segment to the memory location, where a word was most recently stored on the stack.

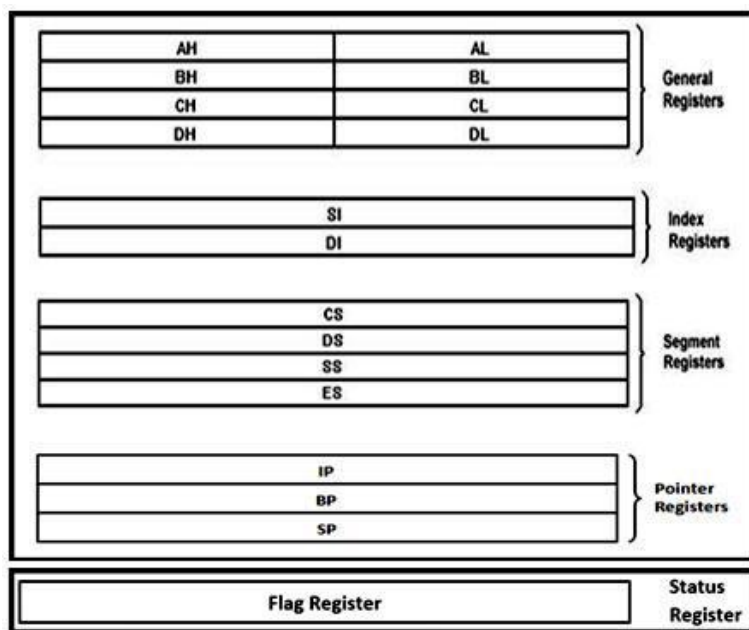
---

## Register Organization

A register is a very small amount of fast memory that is built in the CPU (or Processor) in order to speed up the operation. Register is very fast and efficient than the other memories like RAM, ROM, external memory etc.,

The 8086 microprocessor has a total of fourteen registers that are accessible to the programmer. All these registers are 16-bit in size. The registers of 8086 are categorized into 5 different groups.

- a) General registers
- b) Index registers
- c) Segment registers
- d) Pointer registers
- e) Status Register(Flag register)



**Figure:** Register groups of 8086 micro-processor

### General Registers

All general registers of the 8086 microprocessor can be used for arithmetic and logic operations. These all general registers can be used as either 8-bit or 16-bit registers. The general registers are:

---

**AX (Accumulator):**

AX is used as 16-bit accumulator. The lower 8-bits of AX are designated to use as AL and higher 8-bits as AH. AL can be used as an 8-bit accumulator for 8-bit operation.

This Accumulator used in arithmetic, logic and data transfer operations. For multiplication and division operations, one of the numbers must be placed in AX or AL.

**BX (Base Register):**

BX is a 16 bit register, but BL indicates the lower 8-bits of BX and BH indicates the higher 8-bits of BX.

The register BX is used as address register to form physical address in case of certain addressing modes

**CX (Count Register):**

The register CX is used default counter in case of string and loop instructions. Count register can also be used as a counter in string manipulation and shift/rotate instruction.

**DX (Data Register):**

DX register is a general purpose register which may be used as an implicit operand or destination in case of a few instructions. Data register can also be used as a port number in I/O operations.

**Segment Register:**

The 8086 architecture uses the concept of segmented memory. 8086 can able to access a memory capacity of up to 1 megabyte. This 1 megabyte of memory is divided into 16 logical segments. Each segment contains 64 Kbytes of memory. This memory segmentation concept will discuss later in this document.

There are four segment registers to access this 1 megabyte of memory. The segment registers of 8086 are:

**CS (Code Segment):**

Code segment (CS) is a 16-bit register that is used for addressing code segment of the memory (64Kb), where the executable program is stored.

**Stack segment (SS)**

Stack Segment (SS) is a 16-bit register that used for addressing stack segment of the memory (64kb) where stack data is stored.

**Data segment (DS)**

Data Segment (DS) is a 16-bit register that points the data segment of the memory (64kb) where the program data is stored.

**Extra segment (ES):**

Extra Segment (ES) is a 16-bit register that also points the data segment of the memory (64kb) where the program data is stored.



---

## Index Registers

The index registers can be used for arithmetic operations but their use is usually concerned with the memory addressing modes of the 8086 microprocessor (indexed, base indexed and relative base indexed addressing modes).

The index registers are particularly useful for string manipulation.

### **SI (Source Index):**

SI is a 16-bit register. Source Index Register is used to point the memory locations in the data segment.

### **DI (Destination Index):**

DI is a 16-bit register. This is destination index register performs the same function as SI.

## Pointer Registers:

8086 microprocessor contains three pointer registers.

### **SP (Stack Pointer):**

Stack Pointer register points the program stack that means SP stores the base address of the Stack Segment. We can use the stack pointer to access data from stack segment

### **BP (Base Pointer):**

Base Pointer register also points the same stack segment. Unlike SP, we can use BP to access data in the other segments also.

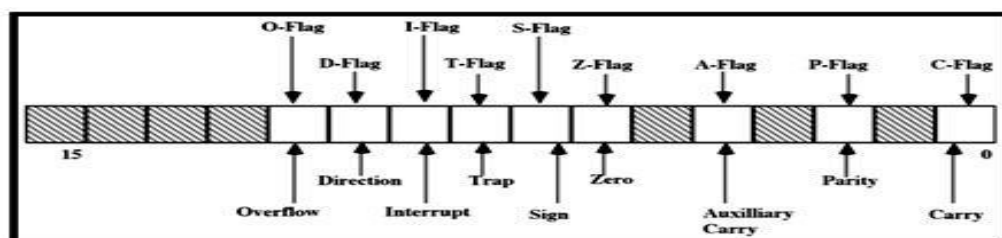
### **IP (Instruction Pointer):**

The Instruction Pointer is a register that holds the address of the next instruction to be fetched from memory. It contains the **offset of the next word of instruction code instead of its actual address**

## Status Register:

The status register also called as flag register. The 8086 flag register contents indicate the results of computation in the ALU.

Flag register is 16-bit register with only nine bits that are implemented. Six of these are status flags. The complete bit configuration of 8086 is shown in the figure.



---

**SF (Sign Flag):** This flag represents sign of the result.

0-Result is Positive

1-Result is Negative

**ZF (Zero Flag):** ZF is set if the result produced by an instruction is zero. Otherwise, ZF is reset.

**PF (Parity Flag):** This flag is set to 1, if the lower byte of the result contains even number of 1's.

0- Odd parity

1- Even parity

**CF (Carry Flag)**

This flag is set, when there is a carry out of MSB in case of addition or borrow in case of subtraction.

0- No Carry/ Barrow

1- Carry/ Barrow

**TF (Trap Flag):** It is used for single step control and allows the user to execute one instruction at a time for debugging. If it is set, then the program can be run in a single step mode.

**IF (Interrupt Flag):**

If this flag is set, the maskable interrupts are recognized by the CPU, otherwise they are ignored.

**DF (Direction Flag):**

This is used by string manipulation instructions.

0- The string is processed beginning from the lowest address to the highest address, i.e., auto incrementing mode.

1- The string is processed from the highest address towards the lowest address, i.e., auto decrementing mode.

**AC (Auxiliary Carry Flag):**

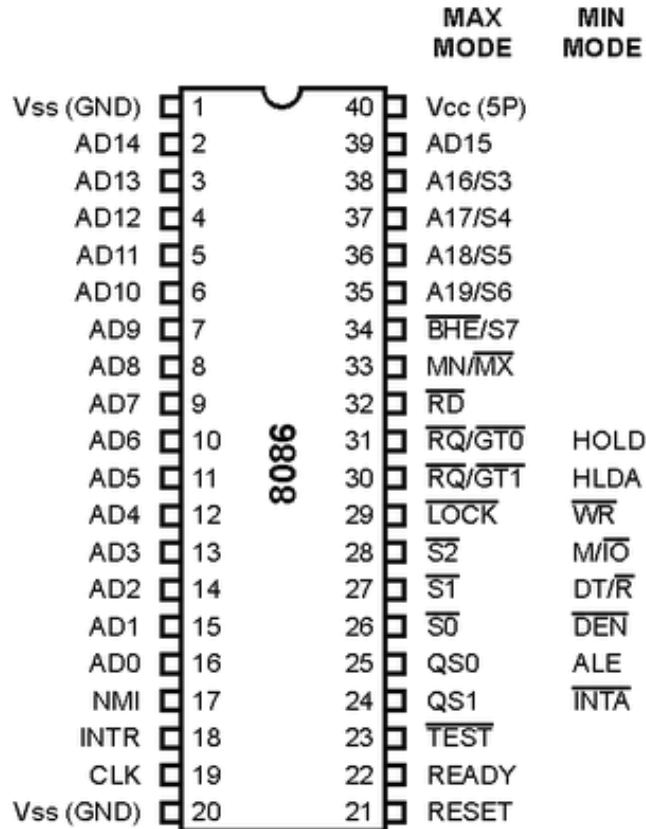
This is set when there is a carry from the lowest nibble (i.e, bit three during addition), or borrow for the lowest nibble (i.e, bit three, during subtraction).

**OF(Over flow Flag):**

This flag is set, if an overflow occurs, i.e, if the result of a operation is large enough to accommodate in a destination register.

### **8086 Pin Diagram**

8086 was the first 16-bit microprocessor available in 40-pin DIP (Dual Inline Package) chip. Here is the pin diagram of 8086 microprocessor –



- **Power supply and frequency signals**

**V<sub>cc</sub>**- +5V power supply for the operation of internal circuit

**GND** – ground for internal circuits

- **Clock signal**

Clock signal is provided through Pin-19. It provides timing to the processor for operations. Its frequency is different for different versions, i.e. 5MHz, 8MHz and 10MHz.

- **Address/data bus**

AD0-AD15. These are 16 address/data bus. AD0-AD7 carries low order byte data and AD8-AD15 carries higher order byte data. During the first clock cycle, it carries 16-bit address and after that it carries 16-bit data.

During T1 state, address are on the lines and data is available on the data bus during T2, T3, Tw and T4. Tw is a wait state

- **Address/status bus(A19/S6,A18/S5,A17/S4,A16/S3)**

---

During T1 state the four address lines for memory operations. During I/O operations these lines are low. During T2, T3 and T4 states status information is available on these lines.

S5: The status of the interrupt flag

S6: is always low

A17/S4	A16/S3	Function
0	0	Extra segment access
0	1	Stack segment access
1	0	Code segment access
1	1	Data segment access

- **S7/BHE**

BHE stands for Bus High Enable. It is available at pin 34 and used **to indicate the transfer of data** using data bus D8-D15. This signal is low during the first clock cycle, thereafter it is active.

S7: not currently used

- **Read**

It is available at pin 32 and is used to read **signal for Read operation.**

- **Ready**

It is available at pin 32. It is an acknowledgement signal from I/O devices that data is transferred. It is an active high signal. When it is high, it indicates that the device is ready to transfer data. When it is low, it indicates wait state.

- **RESET**

It is available at pin 21 and is used to restart the execution. It causes the processor to immediately terminate its present activity. This signal is active high for the first 4 clock cycles to RESET the microprocessor.

- **INTR**

It is available at pin 18. It is an interrupt request signal, which is sampled during the last clock cycle of each instruction to determine the availability of interrupt request.

- **NMI**

---

It stands for non-maskable interrupt and is available at pin 17, indicate immediate interrupt response at the end of current instruction.

- **TEST**

This signal is like wait state and is available at pin 23. If the TEST pin is Low, execution continues. Otherwise the processor waits in an "idle" state.

**MN/MX**

It stands for Minimum/Maximum and is available at pin 33. It indicates what mode the processor is to operate in; when it is high, it works in the minimum mode and vice-versa.

- **INTA**

It is an interrupt acknowledgement signal and is available at pin 24. When the microprocessor receives this signal, it acknowledges the interrupt.

- **ALE**

It stands for address enable latch and is available at pin 25. This signal indicates the availability of a valid address on the address/data lines.

- **DEN**

It stands for Data Enable and is available at pin 26. It is used to indicate the availability of valid data over address/data lines

- **DT/R**

It stands for Data Transmit/Receive signal and is available at pin 27. When it is high, data is transmitted out and when it is low, data is received by processor

- **M/IO**

This signal is used to distinguish between memory and I/O operations. When this pin is Low, it accesses I/O and when high it access memory

- **WR**

It stands for write signal and is available at pin 29. It is used to write the data into the memory or the output device depending on the status of M/IO signal.

- **HLDA**

It stands for Hold Acknowledgement signal and is available at pin 30. This signal acknowledges the HOLD signal.

- **HOLD**

This signal indicates to the processor that external devices are requesting to access the address/data buses. It is available at pin 31.

- **QS<sub>1</sub> and QS<sub>0</sub>**

These are queue status signals and are available at pin 24 and 25. These signals provide the status of instruction queue. Their conditions are shown in the following table –

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

**S<sub>0</sub>, S<sub>1</sub>, S<sub>2</sub>**

These are the status signals that provide the status of operation, which is used by the Bus Controller 8288 to generate memory & I/O control signals. These are available at pin 26, 27, and 28. Following is the table showing their status –

S <sub>2</sub>	S <sub>1</sub>	S <sub>0</sub>	Status
0	0	0	Interrupt acknowledgement
0	0	1	I/O Read
0	1	0	I/O Write
0	1	1	Halt
1	0	0	Opcode fetch
1	0	1	Memory read
1	1	0	Memory write
1	1	1	Passive

---

## LOCK

It indicates to another system bus master, not to gain control of the system bus while LOCK is active Low.

## RQ/GT<sub>1</sub> and RQ/GT<sub>0</sub>

These are the Request/Grant signals used by the other processors requesting the CPU to release the system bus. When the signal is received by CPU, then it sends acknowledgment. RQ/GT<sub>0</sub> has a higher priority than RQ/GT<sub>1</sub>.

## Memory Segmentation

### Segmentation

It is the process in which the main memory of computer is divided into different segments and each segment has its own base address.

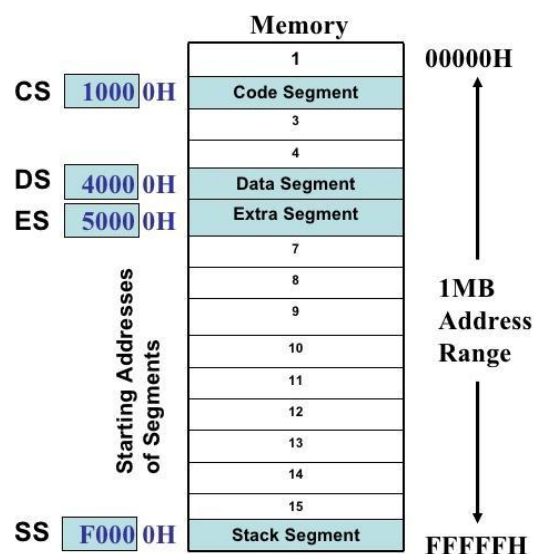
\* Segmentation is used to increase the execution speed of computer system so that processor can able to fetch and execute the data from memory easily and fastly.

### Segmentation in 8086

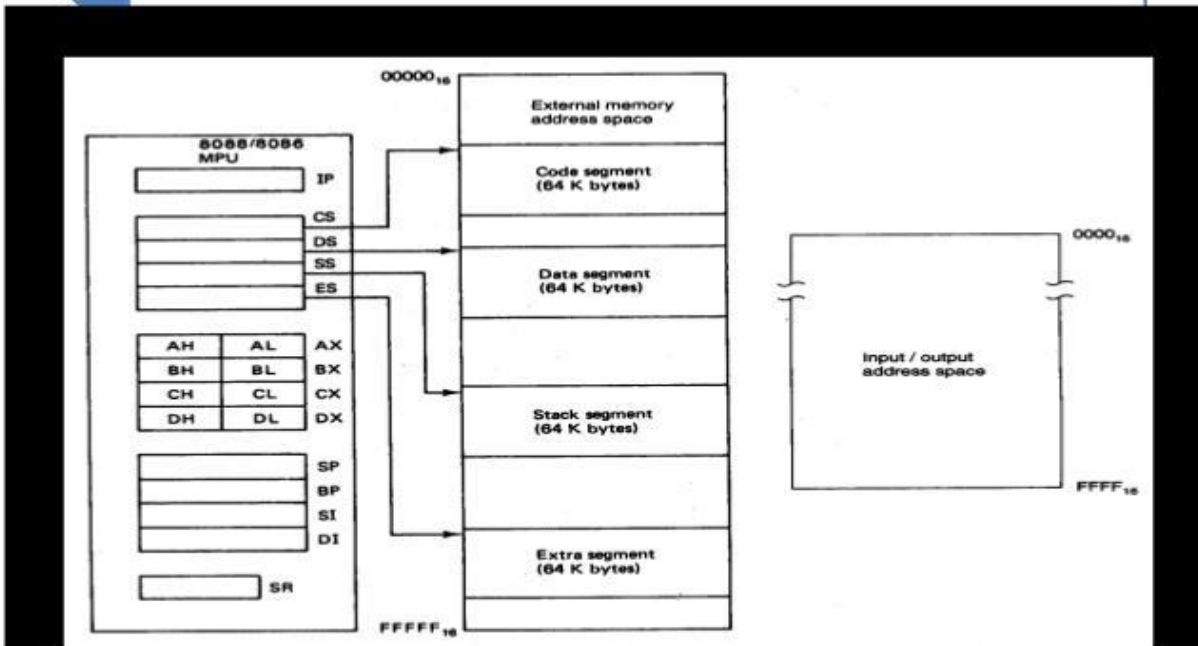
The size of address bus of 8086 is 20 and is able to address 1 Mbytes of physical memory.

The complete 1 Mbytes memory can be divided into 16 segments, each of 64 Kbytes size.

Only four segments can be addressed at a time



## Segmented Memory Representation



### Types of Segmentation

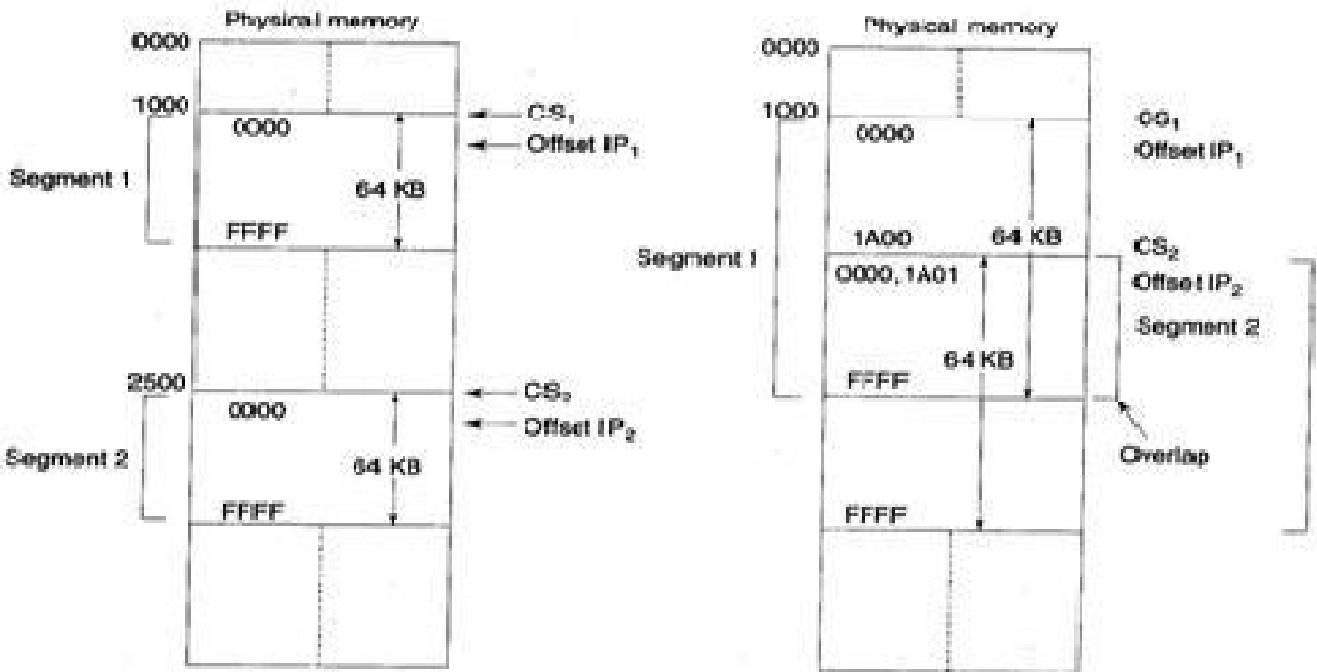
#### Overlapping segment

- A segment starts at a particular address and its maximum size can go up to 64 Kbytes. But if another segment starts along this 64 Kbytes location of the first segment, the two segments are said to be overlapping segment.
- The area of memory from the start of the second segment to the possible end of the first segment is called as overlapped segment.

#### Non Overlapped Segment

- A segment starts at a particular address and its maximum size can go up to 64 Kbytes. But if another segment starts after this 64 Kbytes location of the first segment, the two segments are said to be Non-overlapping segment.





### Advantages of segmentation

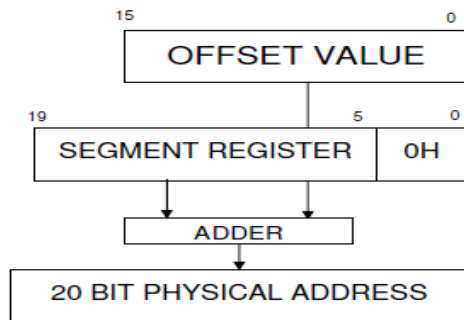
The main advantages of the segmented memory scheme are as follows: Allows the memory capacity to be 1 Mbyte although the actual addresses to be handled are of 16-bit size Allows the placing of code data and stack portions of the same program in different parts (segments) of memory, for data and code protection. Permits a program and/ or its data to be put into different areas of memory each time program is executed, ie, provision for relocation may be done.

### Memory Address Generation

- The BIU has a dedicated adder for determining physical memory addresses. Physical Address (20 Bits)  
Adder Segment Register (16 bits) 0 0 0 0 Offset Value (16 bits)

Logical Address is specified as segment: offset

- Physical address is obtained by shifting the segment address 4 bits to the left and adding the offset address.



CS                      IP

2	5	0	0	:	9	5	F	3
---	---	---	---	---	---	---	---	---

1. Start with CS.

2	5	0	0
---	---	---	---

2. Shift left CS.

2	5	0	0	0
---	---	---	---	---

3. Add IP.

2	E	5	F	3
---	---	---	---	---

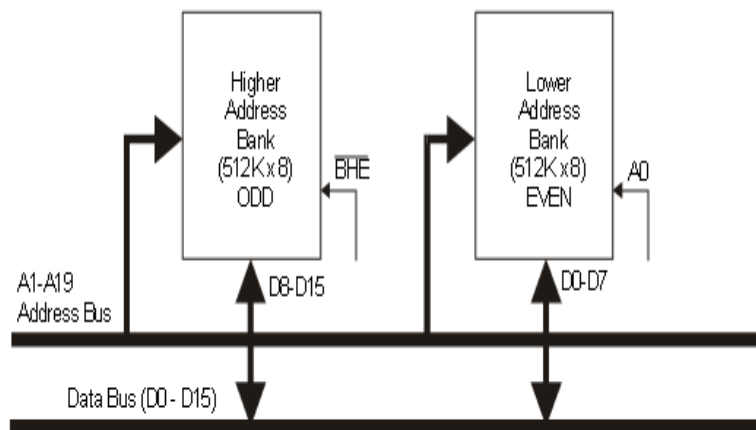
### Physical Memory Organization

- The total memory (1mb) of 8086 is arranged in two banks. An odd bank and an even bank. Both the banks have equal no. of locations.
- The odd bank contains odd numbered memory Locations. It is known as upper bank.
- The even bank contains only even numbered memory. Locations. It is known as lower bank.
- This arrangement is done in order to speed up the operation

Every memory location has two kinds of address – physical and logical.

- A **physical address** is the 20-bit value that uniquely identifies each byte location in the Mega byte memory space.
- These may range from 0 to FFFFF Hex.
- All exchanges between the CPU and memory components use this physical address
- Programs deal with logical, rather than physical, addresses.
- A **logical address consists** of a segment base value and an offset value.

Byte data with even address is transferred on  $D_7 - D_0$  and byte data with odd address is transferred on  $D_{15} - D_8$ .



The processor provides two enable signals,  $\overline{BHE}$  and  $A_0$  for selecting of either even or odd or both the banks.

FIG

$\overline{BHE}$	$A_0$	Function
0	0	Whole word
0	1	Upper byte/ odd address
1	0	Lower byte/even address
1	1	none

### General Bus Operation

Basically, all the processor bus cycles consist of at least four clock cycles. These are referred to as T1, T2, T3, T4.

The 8086 has a combined **address and data bus** commonly referred as a time multiplexed address and data bus. The main reason behind multiplexing address and data over the same pins is the maximum utilization of processor pins and it facilitates the use of 40 pin standard DIP package.

The address is transmitted by the processor during T1. During T2, T3 and T4, data is transmitted over these lines.  $T_w$  is wait state, is introduced between T3 and T4.

**ALE-** is emitted during T1 by the processor or bus controller. Used to separate address or data or status information.

In maximum mode, the status lines **S0, S1 and S2** are used to indicate the type of operation.

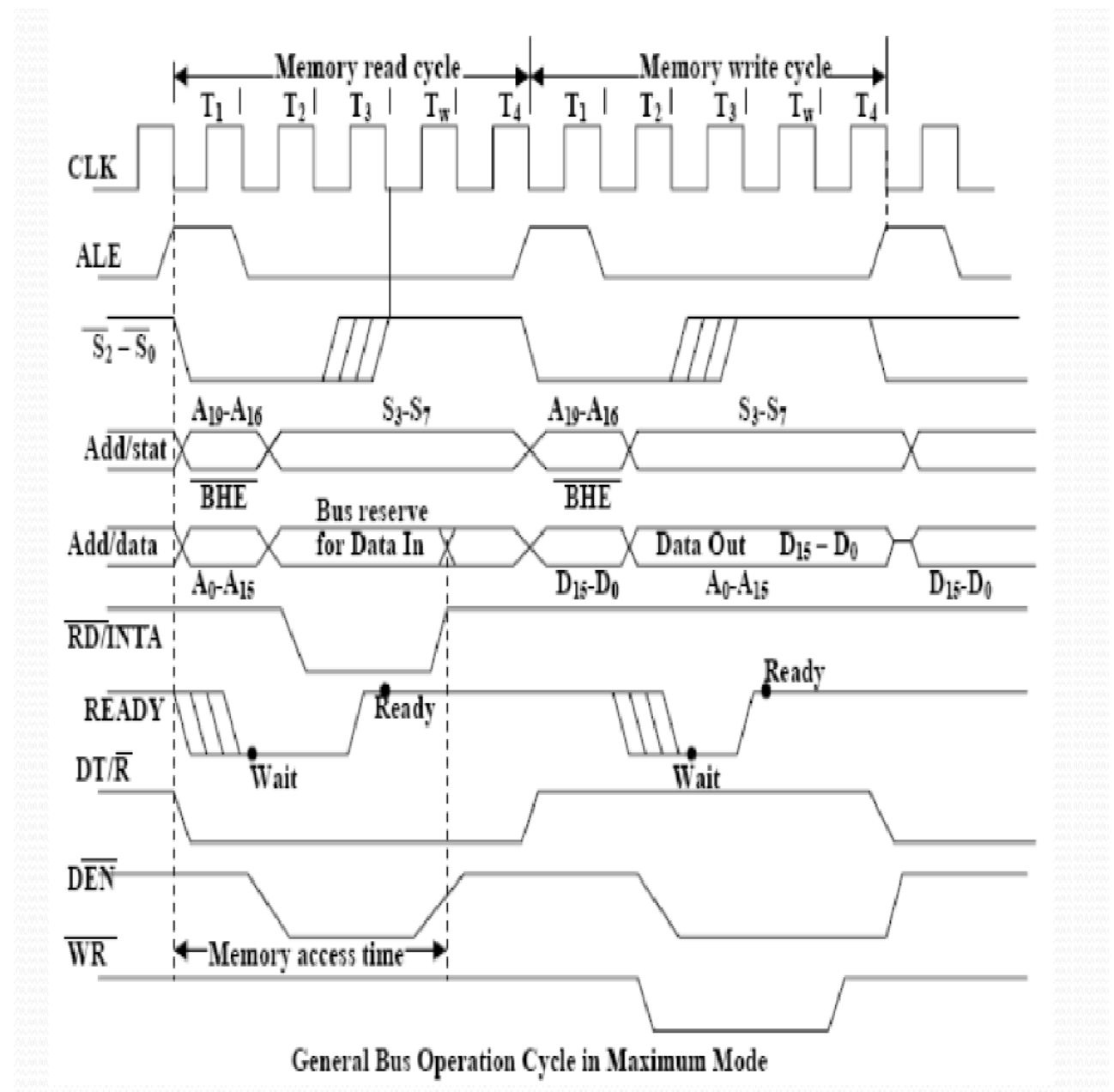
Status bits **S3 to S7** are multiplexed with higher order address bits and **the BHE signal**.

Address is valid during T1 while status bits S3 to S7 are valid during T2 through T4.

General

Bus

Cycle



### Minimum Mode 8086 System

The microprocessor 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 **latches** are generally buffered output D-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 signal generated by 8086.

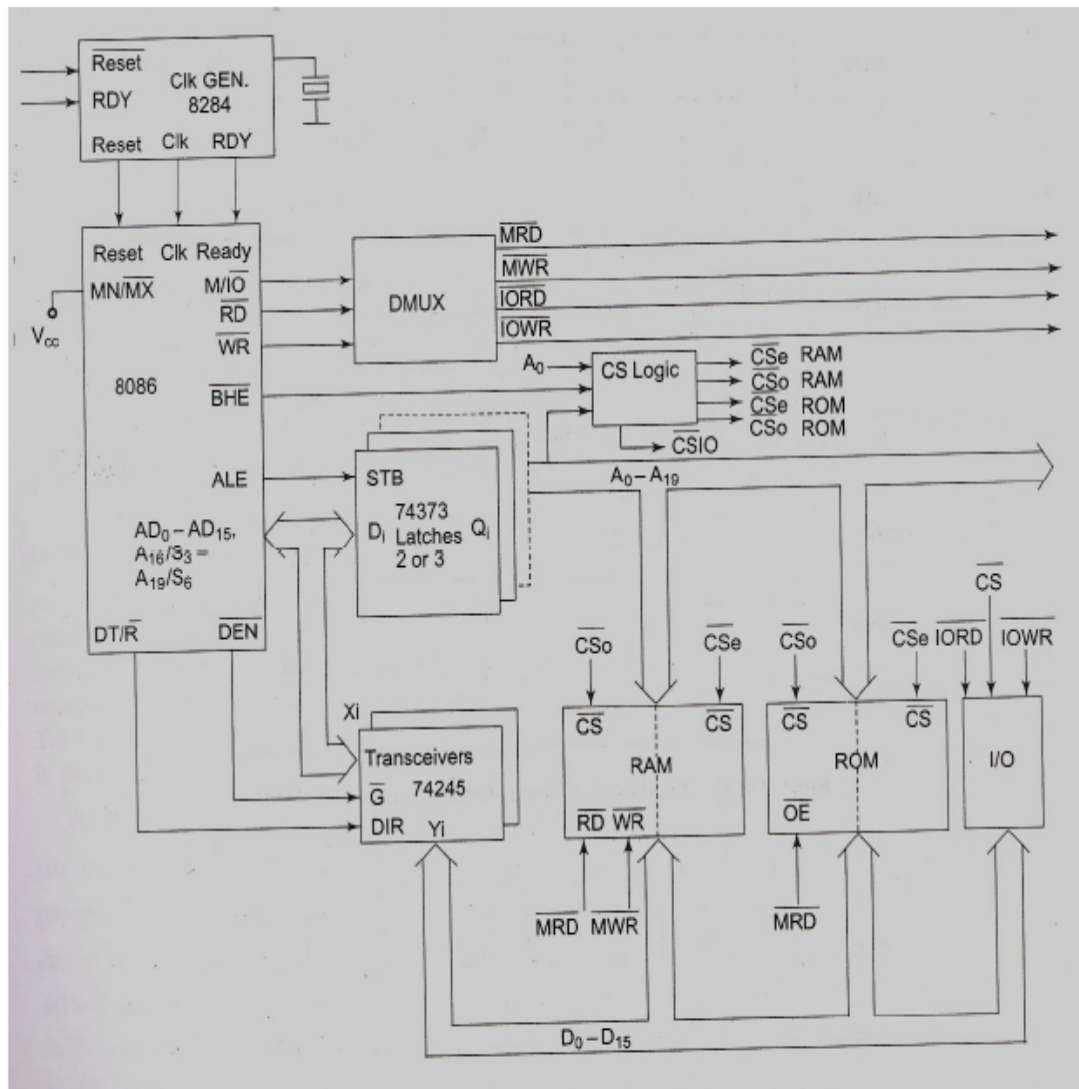


Fig 1.1. Minimum Mode 8086 System

---

**Transreceivers** are the bidirectional buffers they are required to separate the valid data from the time multiplexed address/data signal.

They are controlled by two signals, namely, DEN and DT/R. The DEN signal indicates that the valid data is available on the data bus, while DT/R indicates the direction of data, i.e. from or to the processor.

Usually, **EPROMS** are used for monitor storage, while **RAMs** for users program storage. A system may contain I/O devices for communication with the processor as well as some special purpose I/O devices.

The **clock generator** generates the clock from the crystal oscillator for the system.

The timing diagram can be categorized in two parts, the first is the timing diagram for read cycle and the second is the timing diagram for write cycle.

Fig shows the read cycle timing diagram. The read cycle begins in T1 with the assertion of the address latch enable (ALE) signal and also M/IO signal. During the negative going edge of this signal, the valid address is latched on the local bus.

The BHE and A0 signals address low, high or both bytes. From T1 to T4, the M/IO signal indicates a memory or I/O operation.

At T2 the address is removed from the local bus and is sent to the output. The read (RD) control signal is also activated in T2

The read (RD) signal causes the addressed device to enable its data bus drivers. After RD goes low, the valid data is available on the data bus.

The addressed device will drive the READY line high, when the processor returns the read signal to high level, the addressed device will again tristate its bus drivers.

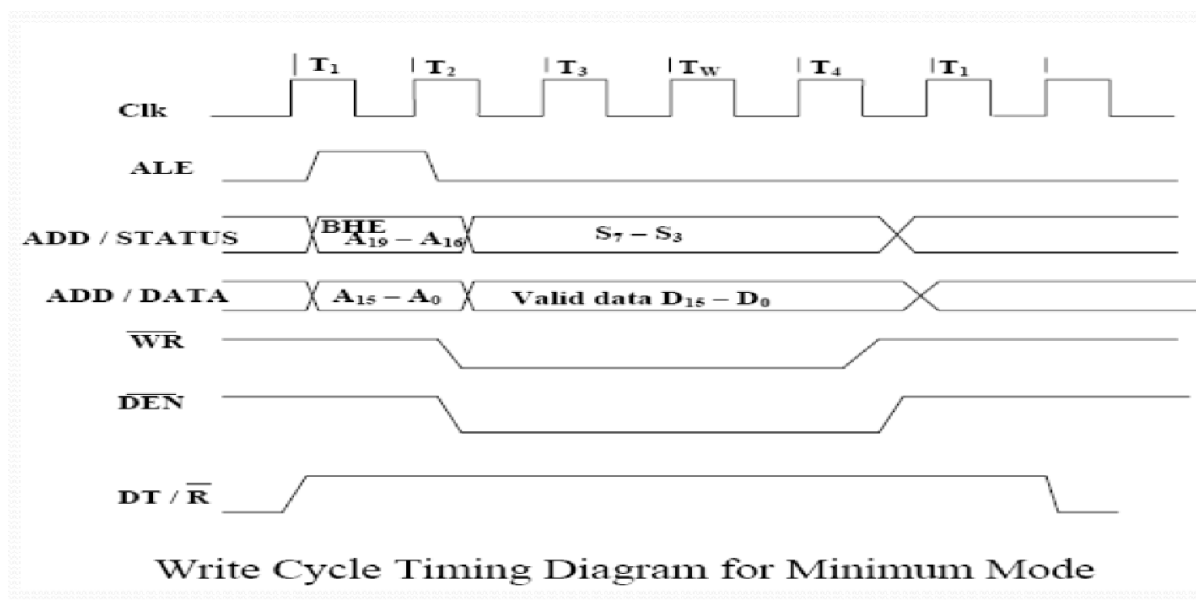
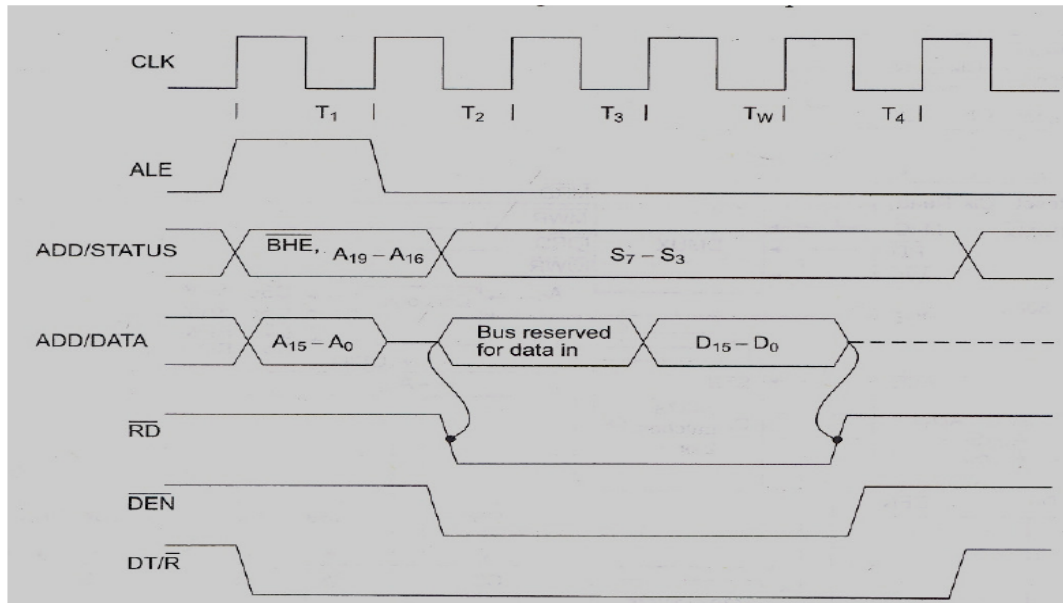
A write cycle also begins with the assertion of ALE and the emission of the address. The M/IO signal is again asserted to indicate a memory or I/O operation. In T2 after sending the address in T1 the processor sends the data to be written to the addressed location.

The data remains on the bus until middle of T4 state. The WR\* becomes active at the beginning of T2 (unlike RD is somewhat delayed in T2 to provide time for floating).

The BHE and A0 signals are used to select the proper byte or bytes of memory or I/O word to be read or written. The M/IO, RD and WR signals indicate the types of data transfer as specified in Table

M/I/O	RD	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

Read cycle timing diagram



---

### Maximum Mode 8086 System

In the maximum mode, the 8086 is operated by strapping the MN/MX pin to ground.

In this mode, the processor derives the status signals S2, S1 and S0.

Another chip called **bus controller drivers** the control signals using this status information. In the maximum mode, there may be more than one microprocessor in the system configuration.

The other components in the system are the same as in the minimum mode system. The general system organization is as shown in the fig

The basic functions of the **bus controller chip IC8288**, is to derive control signals like RD and WR (for memory and I/O devices), DEN, DT/R, ALE, etc. using the information made available by the processor on the status lines.

The bus controller chip has input lines S2, S1 and S0 and CLK. These inputs to 8288 are driven by the CPU. It derives the outputs ALE, DEN, DT/R, MWTC, AMWC, IORC, IOWC and AIOWC. The AEN, IOB and CEN pins are specially useful for multiprocessor systems. AEN and IOB are generally grounded. CEN pin is usually tied to +5V.

The significance of the MCE/PDEN output depends upon the status of the IOB pin. If IOB is grounded, it acts as master cascade enable to control cascaded 8259A; else it acts as peripheral data enable used in the multiple bus configurations

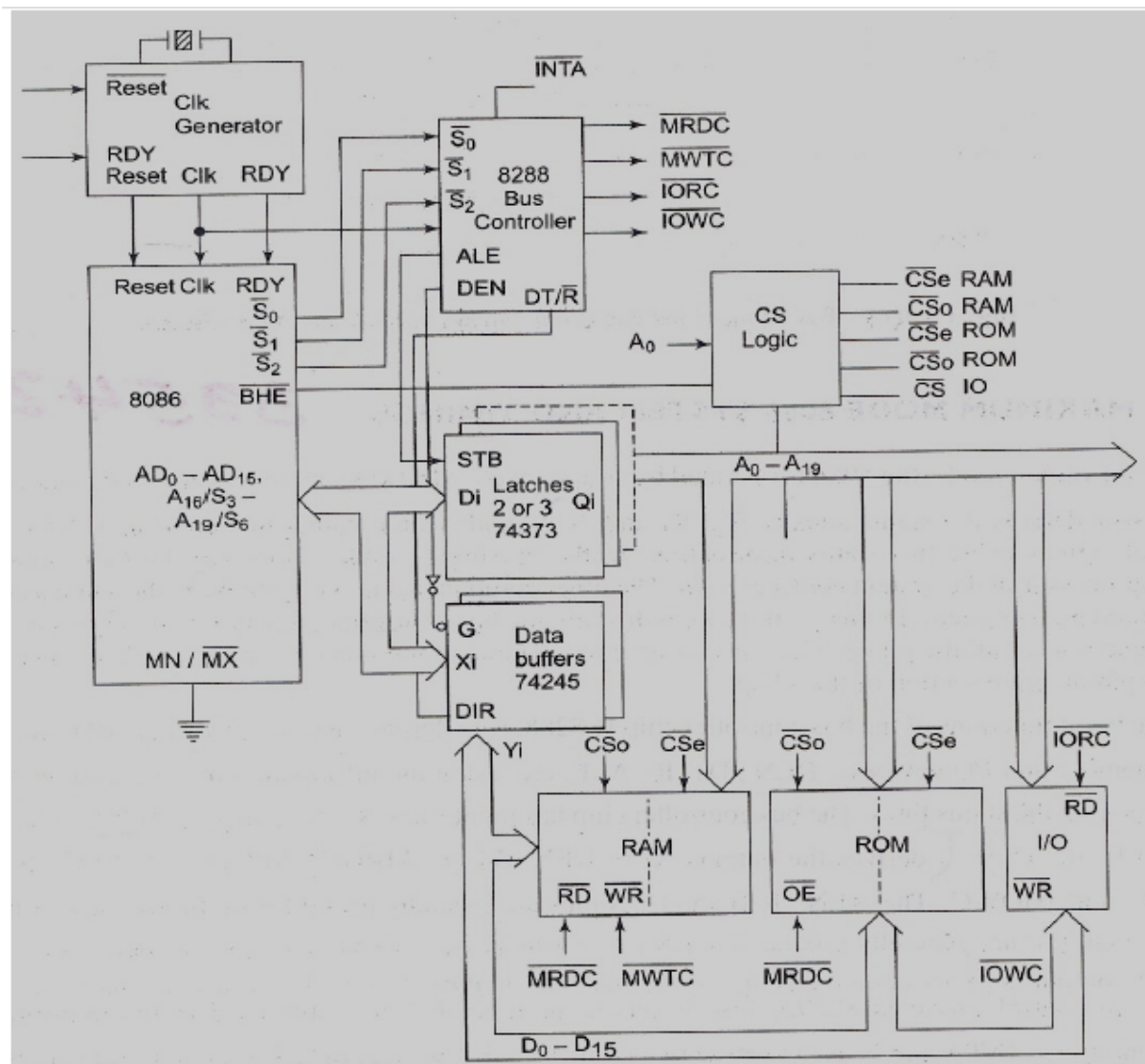
INTA pin is used to issue two interrupt acknowledge pulses to the interrupt controller or to an interrupting device.

IORC, IOWC are I/O read command and I/O write command signals respectively. These signals enable an IO interface to read or write the data from or to the addressed port.

The MRDC, MWTC are memory read command and memory write command signals respectively and may be used as memory read and write signals.

All these command signals instruct the memory to accept or send data from or to the bus. For both of these write command signals, the advanced signals namely AIOWC and AMWTC are available. They also serve the same purpose, but are activated one clock cycle earlier than the IOWC and MWTC signals, respectively.





**Fig Maximum Mode 8086 System**

The maximum mode system timing diagrams are also divided in two portions as read (input) and write (output) timing diagrams. The address/data and address/status timings are similar to the minimum mode. ALE is asserted in T1, just like minimum mode. The only difference lies in the status signals used and the available control and advanced command signals.

Fig. Memory Read Timing in Maximum Mode

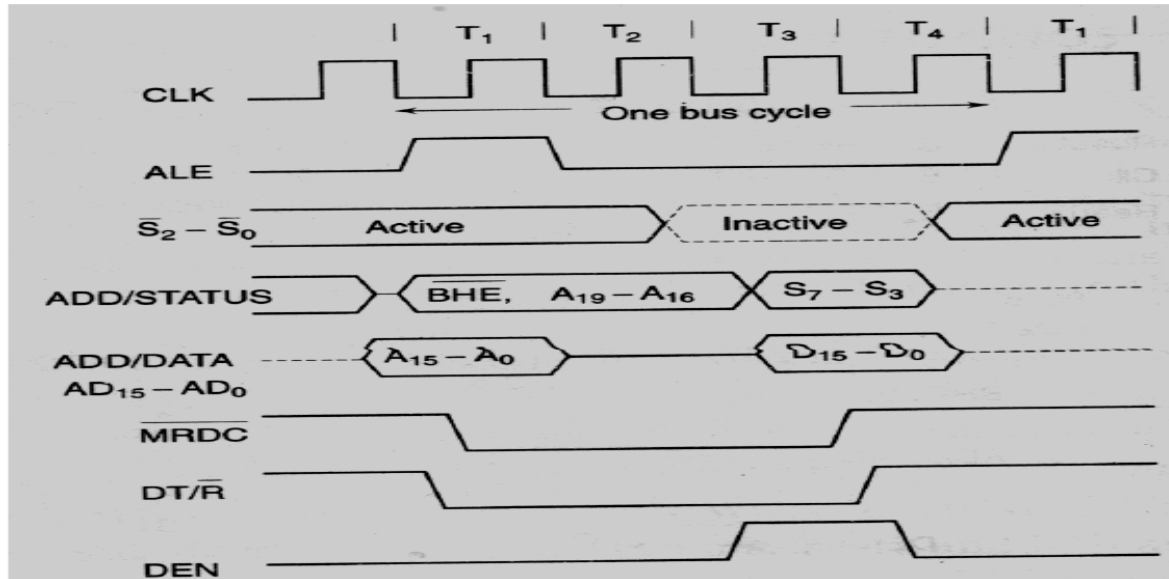
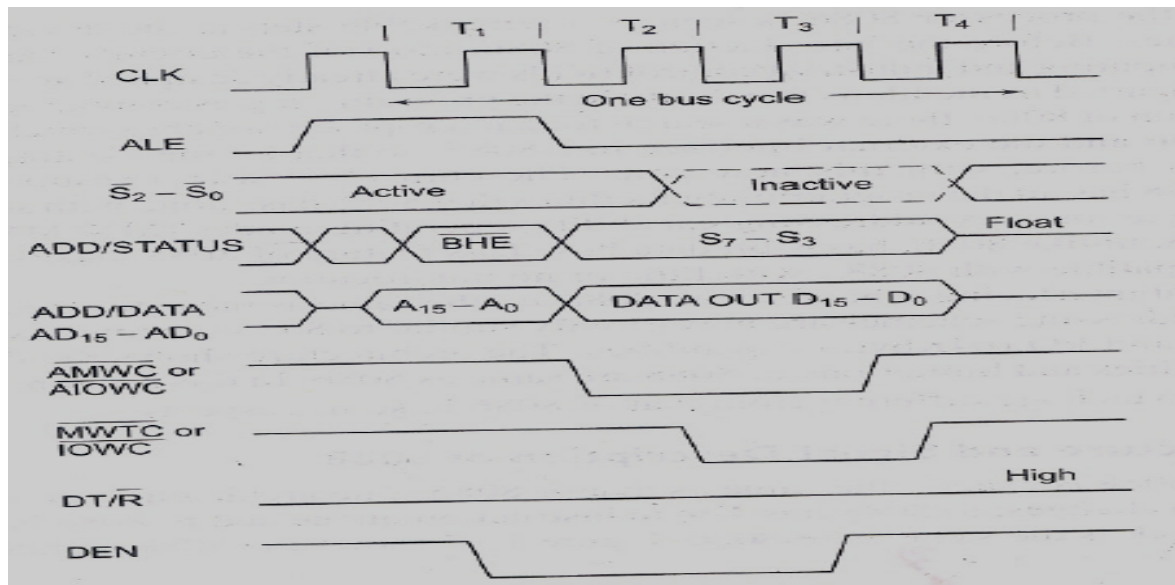
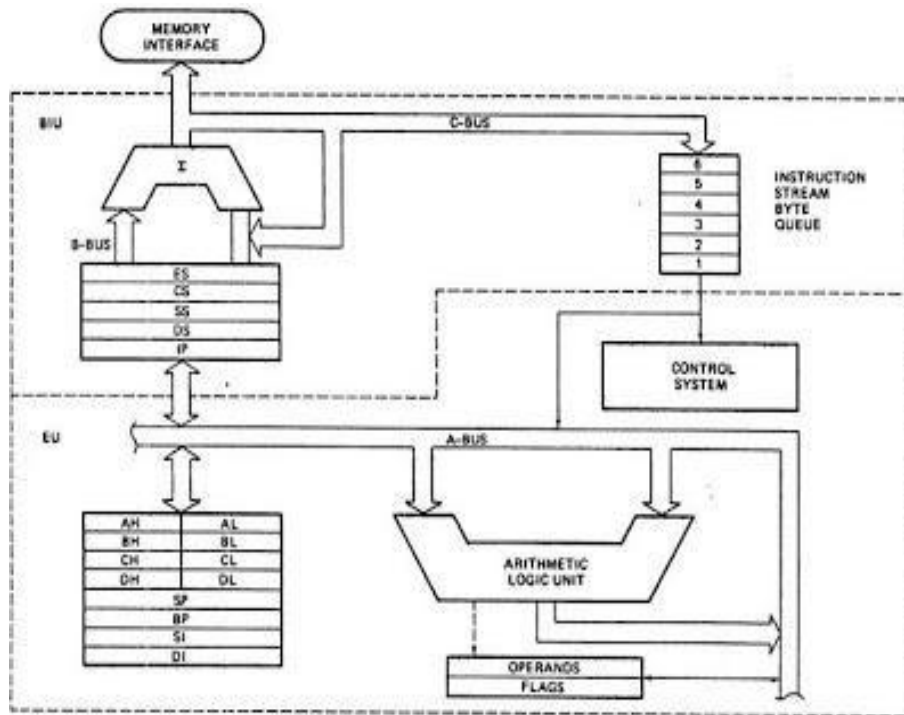


Fig. Memory Write Timing in Maximum Mode



## Comparison between 8086 and 8088

### 8088 Architecture



8086 Microprocessor	8088 Microprocessor
8086 has 16-bit data lines.	8088 has 8-bit data lines.
8086 is available in three clock speed 5 MHz, 8 MHz and 10 MHz	Whereas 8088 is available in two clock speed 5 MHz and 8 MHz
The memory space of 8086 is organized as two 512KB banks.	The memory space of 8088 is implemented as single 1M*8 Memory bank.
8086 has 6-bit instruction queue.	8088 has 4-bit instruction queue.
The 8086 has BHE (Bank high enable)	The 8088 has SSO status signal.
The 8086 can read or write 8-bit or 16-bit data at a time.	The 8088 can read/write 8-bit data at a time.
The I/O voltages level for 8086 is measured at 2.5 mA.	The I/O voltages level for 8086 is measured at 2 mA.
The 8086 draws maximum supply current of 360mA.	The 8086 draws maximum supply current of 340mA.

### 8088 Pin Diagram

**AD7-AD0(Address/Data bus):-**These lines constitute address or data time multiplexed

**A15-A8(Address):-**These lines provide address bits A8-A15 in the entire bus cycle.

**SS0:-**A new SS0 is introduced in 8088 instead of BHE pin in 8086. In minimum mode, the pin ss0 is equivalent to s0 in the maximum mode. In maximum mode it is always high

## IO/M

This pin is similar to M/IO pin of 8086 , but it offers an 8085 compatible, memory I/O bus interface

## 8086/8088 Pin Configuration

