1. 8086 – INTRODUCTION AND ARCHITECTURE

1.0 Introduction:

A microprocessor is one of the most exciting technological innovation in electronics. Since the appearance of transistor in 1948. A Microprocessor is the chip containing some control and logical circuits that is capable of making arithmetic and logical decisions. Based on input data and produces the corresponding arithmetic and logical output. It is the brain of any computer, whether it is a desktop machine, a server or a laptop. The microprocessor not only forms the very basis of computers, but also many other devices such as cell phones, satellites, many projects, control equipment toys and many other hand held devices.

1.1 Define microprocessor and microcomputer

A microprocessor is one of the most exciting technological innovation in electronics.

"**Microprocessor** is a programmable controlled semiconductor device (IC), which fetches, decodes, and executes the instructions. It is used as CPU in computers."

(or)

The word micro in microprocessor refers to its small size and the processor refers to the device that performs computational and control operations. The basic functional block diagram of a microprocessor is given below

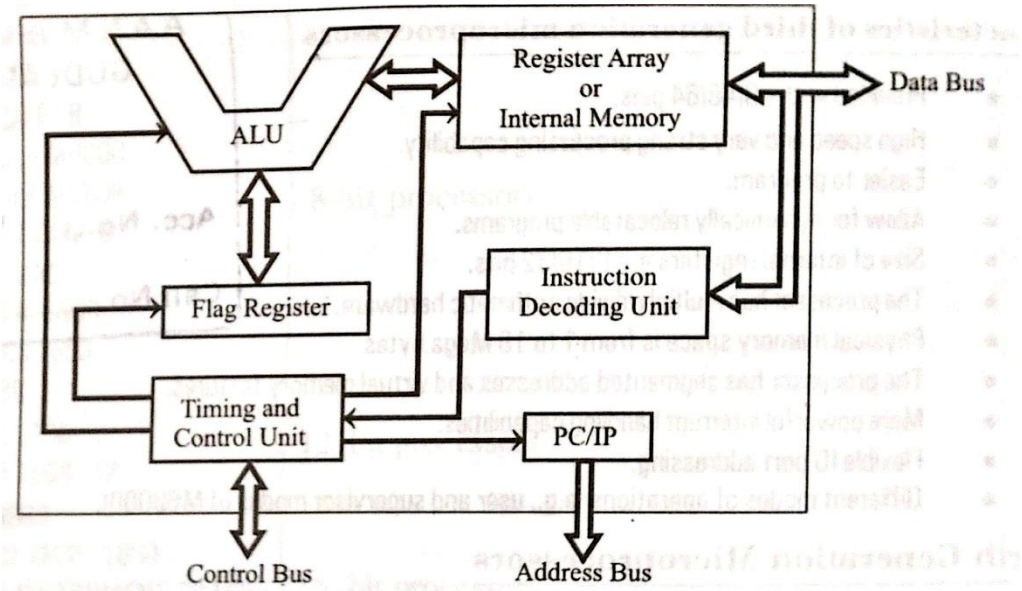


Fig : Block Diagram of Microprocessor

The block diagram consists of

ALU: it is the computational unit of the microprocessor, which performs arithmetic and logical operations on binary data.

Register array: the register array is the internal storage for microprocessor. The input data for ALU, the output data from ALU and any other binary information needed for processing are stored in the register array.

- For any microprocessor, the instructions are stored in a memory, which is external to the microprocessor.

Instruction pointer/Program counter: The IP generates the address of the next instruction to be fetched from the memory and sends it through the address bus to the memory.

Flag register: flag register contents are used to show the status of the operation which is done by ALU. Based on the operation performed ALU the flag bits will be set or reset.

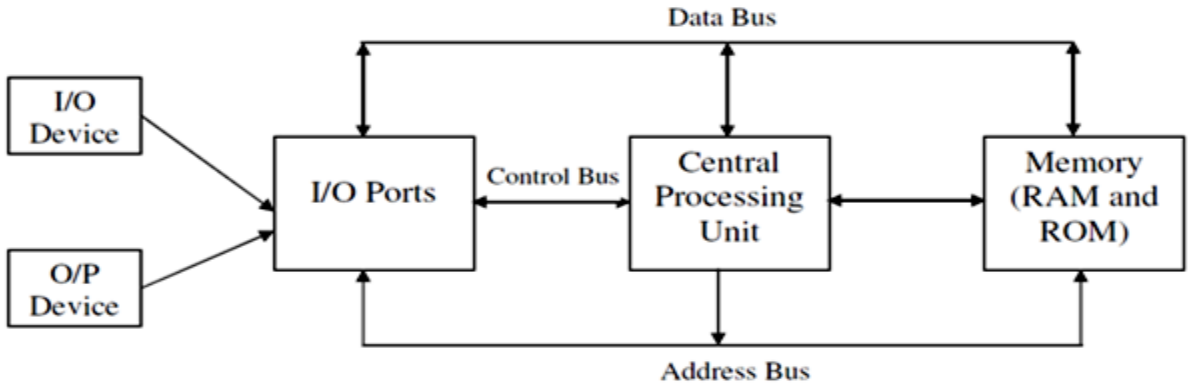
Instuction decoding unit: It is used to decode the instructions or seperates the opcode from the operands.

Control unit: The control unit will generate the necessary control signals for internal and external operations of microprocessor.

Definition of Microcomputer:

"Microcomputer is defined as a system designed using the **microprocessor** as its CPU and also contains semiconductor memories, EPROM, RAM, input devices, output devices and interfacing devices."

The block diagram of a microcomputer is given below



The block diagram consists of

CPU: the CPU controls the operation of computer. In a microcomputer the CPU is a microprocessor. It fetches binary coded instructions from memory, translates into a series of simple actions and carries out these actions into a sequence of steps.

Memory: The memory is a device to store data.Memory can be viewed as set of memory locations, each location has an address to identify it by CPU. It is classified into two types. They are

- **Primary memory :**it can directly accessed by the CPU. examples are RAM and ROM.
- **Secondary memory:** it can not be directly accessed by the CPU. Examples are floppy disks, magnetic tapes and hard disks.

I/O devices: Input devices are used for feeding data and instructions to the CPU fro the outside world.
Examples are keyboard, mouse and scanner.

The output devices are used for delivering the results of operations to the outside world.
Examples are printers, monitors, LED's.

- The input and output devices are connected to the CPU by using **I/O ports.**

System Bus: A bus is a group of conducting lines, that connects the parts of microcomputer. Buses are generally three types, they are

- **Address Bus**:Generally the address is an identification number used by microprocessor to access memory location or I/O devices. The group of conducting lines that carries address are called **Address Bus.** It is a uni-directional bus.
- **Data Bus**: The group of conducting lines that carries data are called **Data Bus.** It is a Bi-directional bus.
- **Control Bus**: The group of conducting lines that carries control signals are called **Control Bus.** Typical control signals are memory read, memory write, I/O read and I/O write.

1.2 Describe how a microcomputer fetches and executes an Instruction?

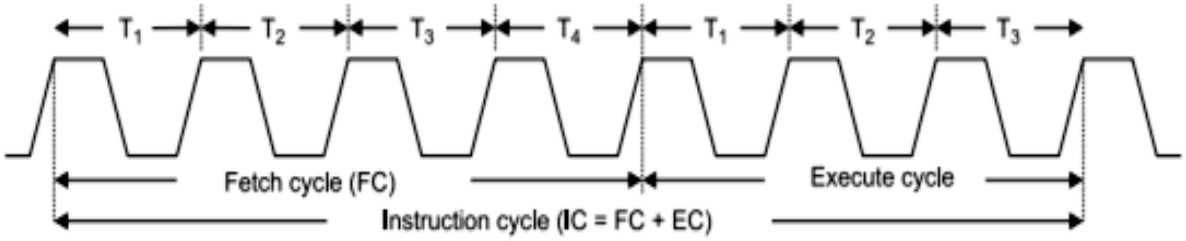
For any microprocessor there will be a set of instructions given by the manufacturer of the microprocessor. For doing any useful work with the microprocessor we have to write a program using these instructions and are stored in a memory device, which is external to the microprocessor.

Once a program is in the memory, it has to be executed by fetching, decoding the instructions from the memory. This constitutes an "instruction cycle".

An Instruction cycle is defined as the time required to complete the execution of an instruction.

Instruction cycle is the combination fetch cycle and execute cycle.

Instruction cycle(IC) = fetch cycle(FC)+ execute cycle(EC)



Fetch cycle: The fetch cycle is the time required to fetch an instruction from the memory. Fetching is nothing but retrieving the instruction from memory and transferring it to the CPU.

Execute cycle: The execute cycle is defined as the time required to decode and execute an instruction. Decoding is nothing but observing the opcodes.

8086 FEATURES:

It is a third generation processor, which is successor of 8085 introduced by Intel in 1978. It is 16-bit processor (that means, ALU can able to perform 16-bit operations at a time). It is a 40 pin DIP (Dual in line pin package) IC.

- It has 16-bit data bus and control bus.
- It has 20-bit address bus. So, it can able to address 1Megabyte memory locations.
- It has 6-byte prefetch instruction queue. By this pipelining can be possible.
- It has 16-bit flag register.
- It can generate 16-bit I/O address.
- It is operated in two modes, i.e.,minimum mode and maximum mode.
- It has fourteen 16-bit registers.
- It supports multiprogramming.
- It requires an external asymmetric clock source with 33% duty cycle.

1.3 Internal Architecture of 8086

The 8086 has a pipelined architecture. In pipelined architecture, the processor will executes and fetches instructions side by side. The architecture of 8086 can be divided into two separate functional units.

I. Bus Interface unit (BIU)
II. Execution Unit (EU)

Bus Interface Unit (BIU) : The BIU communicates with peripheral devices which are outside of the microprocessor. The BIU is resposible for performing all external bus operations as given below.

- ✓ It sends out address of the memory or I/O ports.
- ✓ It fetches instructions from memory.
- ✓ It reads data from memory or I/O ports.
- ✓ It writes data to memory or I/O ports.
- ✓ It supports instruction queuing.

BIU contains following functional blocks.

- **Segment registers:** The BIU has four 16-bit segment registers. They are Code segment register (CS), Stack segment register (SS), Data segment register (DS) and Extra segment register (ES). Segment registers are used to store the segment address of the memory segments.

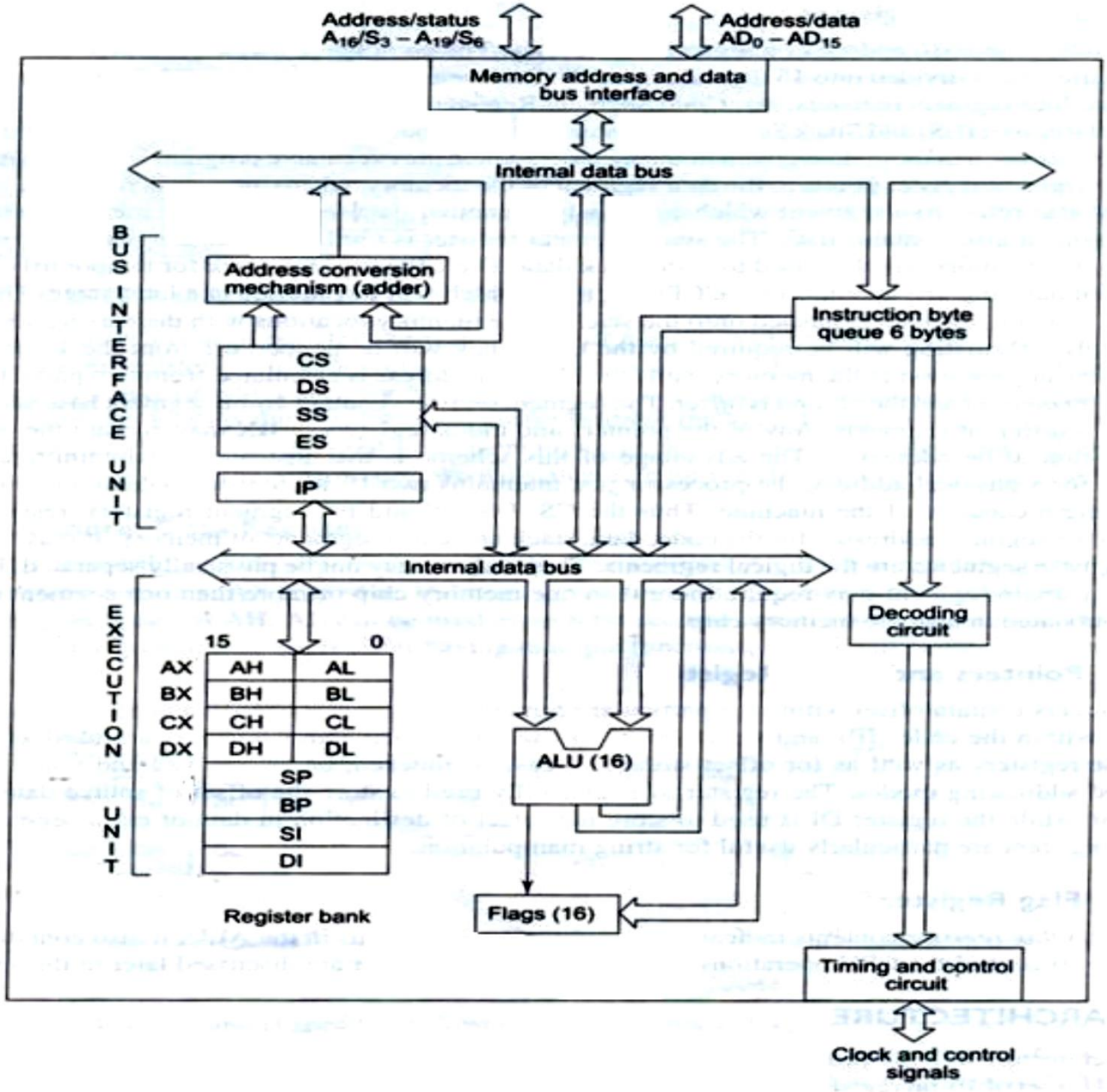
- **Address Conversion mechanism:** It generates 20-bit physical address by using the segment address (or) base address and offset address.

$$\text{Physical address} = \text{segment address} \times 10_H + \text{offset address}$$

(note: base address stored in segment registers and offset address stored in IP, SP, BP, SI and DI.**)**

- **Insrtuction Queue:** The 8086 has 6-bytes pre-fetch instruction queue. The instructions are stored in the First –in-first-out (FIFO) manner. By using prefetch queue, the 8086 processor supports pipelining. Pipelining means , fetching the next instruction while the current instruction executes.

Execution Unit (EU): The execution unit fetches instruction codes from the queue and decodes that instructions and executes them. It has ALU, Flag register, general purpose registers, pointers, index registers, decoding circuit and timing and control unit.



- **ALU**: ALU is used to perform arithmetic and logical operations.
- **Decoding Circuit**: Decoding circuit is used to decode the instruction opcode bytes which are retrieved from the queue.

- **Timing and control unit:** It is used to generate necessary control signals required for the processor.
- **General purpose registers:** General purpose registers used by the programmer and these are used to storing the data temperarly.
- **Pointer registers:** These are used to point the offset address of the memory segments.
- **Index registers:** used to store the offset addresses of segment memories and is used to store the contents of the operands.
- **Flag register:** 8086 has 16-bit flag register. Flag register is used to show the status of the current result produced by the ALU and some of the flags are also used to control the processor operation. 8086 has nine active flags, six of them are called condition code flags and the remaining three are called machine control flags. There are two types of flags in 8086. They are
    I. Condition code flags (shows the status of the result)
    II. Machine control flags (used to control processor operation)

1.4 List the registers and other parts in 8086

The functional block diagram of 8086 contains

I. ALU
II. Prefetch instruction queue (6- bytes)
III. Timing and control unit
IV. Instruction decoding unit
V. Adderess conversion mechanism
VI. Timing and control unit
VII. It has fourteen(14)16-bit register set

- ❖ The fourteen(14) 16-bit register can be classified as follows

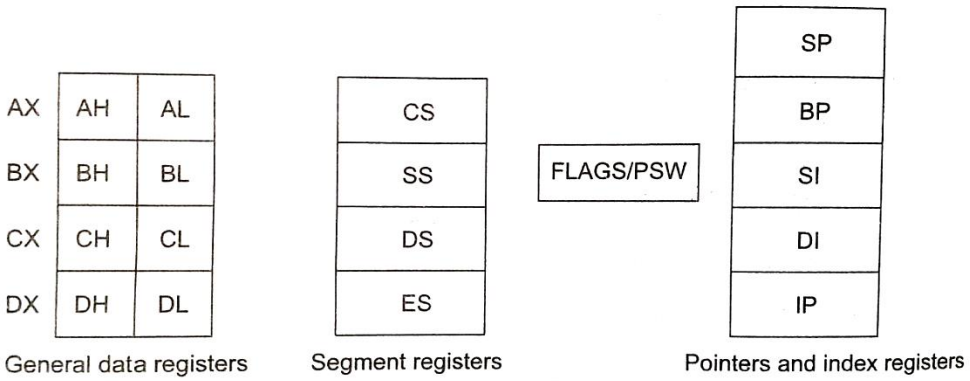
1. General purpose registers
    a) AX(accumulator register)
    b) BX(base register)
    c) CX(count register)
    d) DX(data register)
2. special purpose registers
    a) stack pointer(SP)
    b) base pointer(BP)
    c) program counter/Instruction pointer(PC/IP)
    d) source index(SI)
    e) destination index(DI)
3. segment registers
    a) Code segment register(CS)
    b) Stack segment register(SS)
    c) Data segment register(DS)
    d) Extra segment register(ES)
4. flag register

1.5 Describe the function of each block in 8086

The functional block diagram of 8086 contains 16-bit register set, ALU,Instruction Decoding unit, Timing and control unit and 6-bytes prefetch instruction queue.

8086 16-bit register set, it can be classified as follows

1. General purpose registers 2. Special function registers 3. Flag register



1. General purpose registers: 8086 has **four** general purpose registers. They are AX, BX, CX, DX. These registers can be used as two 8-bit registers like AH-AL, BH-BL, CH-CL and DH-DL. 'H' represents the higher byte and 'L' represents lower byte and X represents total 16-bit register.

- **AX register:** This register is also called as accumulator register. When the processor performs any arithmetic and logical operations, it provides one of the operand and it also hold the result of operation.
- **BX register:** BX register is used to store the offset address for certain addressing modes. It is called as base register.
- **CX register:** CX register is the default counter in 8086. It is used to store the count value for loop instructions. It is called as counter register.
- DX register: It is also called as data register. In multiplication and division operations, if the data is of 32-bit, the DX register is used to store the MSB 16 bits and AX is used to store the LSB 16bits.

2. Special function registers: The special function registers are **Segment registers, pointers and index registers**.

a) Segment registers: There are four 16-bit segment registers, namely Code Segment register (CS), Stack Segment register (SS), Data Segment register (D) and Extra Segment (ES) The segment registers are used to store the base address or segment address of the memory segments.

- **Code segment register (CS):** Code segment register is used to store the base address of the code segment memory, where the executable program is stored.
- **Data segment register(DS):** Data segment register is used to store the base address of the data segment memory, where data is resided.
- **Stack Segment register (SS):** Stack segment register is used to store the base address of the Stack segment memory, where stack data is stored.
- **Extra Segment memory (ES):** Extra segment register is used to store the base addres of the extra segment memory which is also used to store data.

b) Pointer registers: There are three 16-bit pointer registers. They are, Base pointer, Stack pointer and Instruction pointer.

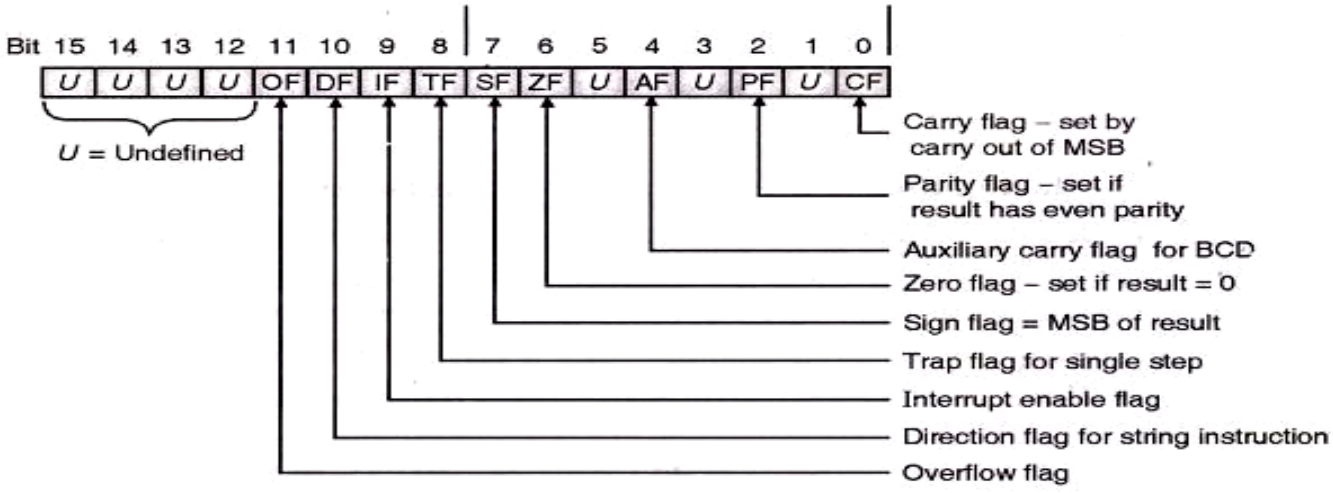
- **Base pointer and Stack pointer:** the base pointer and stack pointer is used to hold the offset address of stack segment memory.
- **Instruction pointer:** It holds the offset address of the next instruction to be fetched from memory.

c) Index registers: There are two types of index registers in 8086, they are Source Index and Destination Index register. These Index registers are used to store the offset address for data and extra segments respectively.

3. Flag registers/PSW: 8086 has 16-bit flag register/PSW, in that only 9-bits are currently used to represent the different flags, the remaining 7 bits are undefined. Among the **nine flags**, six flags are status flags and three are control flags.

Status/conditional code flags: Used to show the status of the processor after completion of each instruction.

- **Carry flag (CF):** Carry flag is set if there is a carry is generated from the MSB bit after adding two numbers or barrow from subtraction.
- **Parity Flag (PF):** It is set, when the lower byte of result has an even parity (i.e., even no of 1's) and is reset for odd parity of the result (i.e., odd no of 1's).
- **Auxiliary Carry Flag (AF):** It is set, if there is a carry from lower nibble(D_3) to (D_4)higher nibble.
- **Zero Flag (ZF):** It is set, when the result of Arithmetic and Logical operation is zero.
- **Sign flag (SF):** Sign Flag is set, if the MSB of the result is '1', otherwise '0'.
- **Overflow Flag (OF):** It is set, if there is an arithmetic overflow, i.e., if the size of the result is exceeds the capacity of the destination, otherwise it is zero.



8086 flag register format

Control flags: These are used to control the processor activities.

- **Trap Flag (TF):** Setting the trap flag to one, places the 8086 in single step execution mode. In this mode, 8086 generates an internal interrupt, after execution of each instruction. Trap flag is mainly used for debugging process.
- **Direction Flag (DF):** It is set to '1' for auto decrement and it is reset for auto increment of SI and DI registers during string instructions.
- **Interrupt Flag (IF):** It is set when the 8086 recognise an externalon its interrupt pins of 8086 and clearing IF to '0', disables the interrupts.

6-bytes prefetch instruction queue: 8086 has a prefetch instruction queue with 6-bytes long. While execution unit is decoding or executing an instruction, the BIU fetches upto 6-bytes. The BIU stores these prefetched bytes in a first-in-first-out register called queue. This prefetching scheme greatly speeds up the processing.

Instruction Decoding Unit: Decoding circuit is used to decode the instruction opcode bytes which are retrieved from the queue.

Timing and control unit: It is used to generate necessary control signals required for the processor.

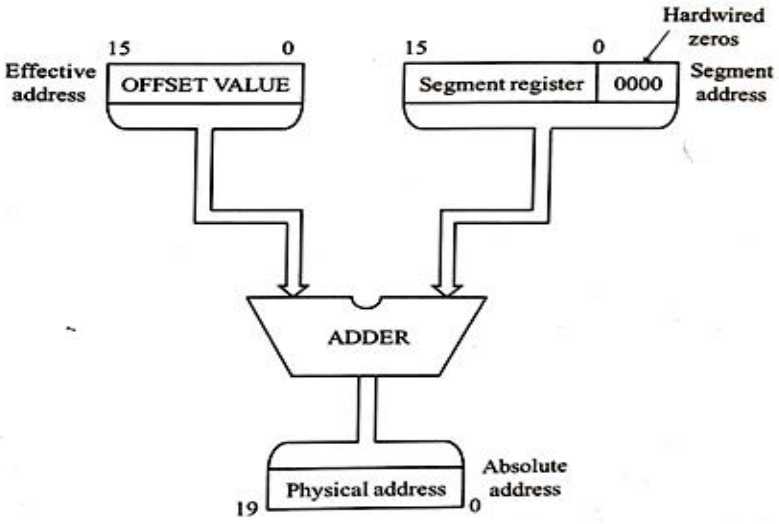
ALU: ALU is used to perform arithmetic and logical operations. The arithmetic operations such as addition, subtraction etc. the logical operations such as AND, OR etc.

1.6 Demonstrate how 8086 calculates memory address/physical address

The 8086 microprocessor has the address bus of size 20-bit. So, the physical memory (1Mega Byte) can be addressed from 00000H to FFFFFH. The 20-bit physical address of different segments can be calculated by using base address and offset address.

The **offset address** is a 16-bit value, which is the displacement of desired operand from the segment base.
The **base address** is a starting address of the memory segments.

Physical address= base address×10_H + offset address



For example: If we want to calculate the physical address of code segment, the base address is 548BH and offset address is 1242H, then the physical address is calculated as follows:

Physical address = base address : offset address

PA = CS : IP

548BH 1242H

PA = (CS × 10_H)+IP

= (548B × 10_H)+1242H

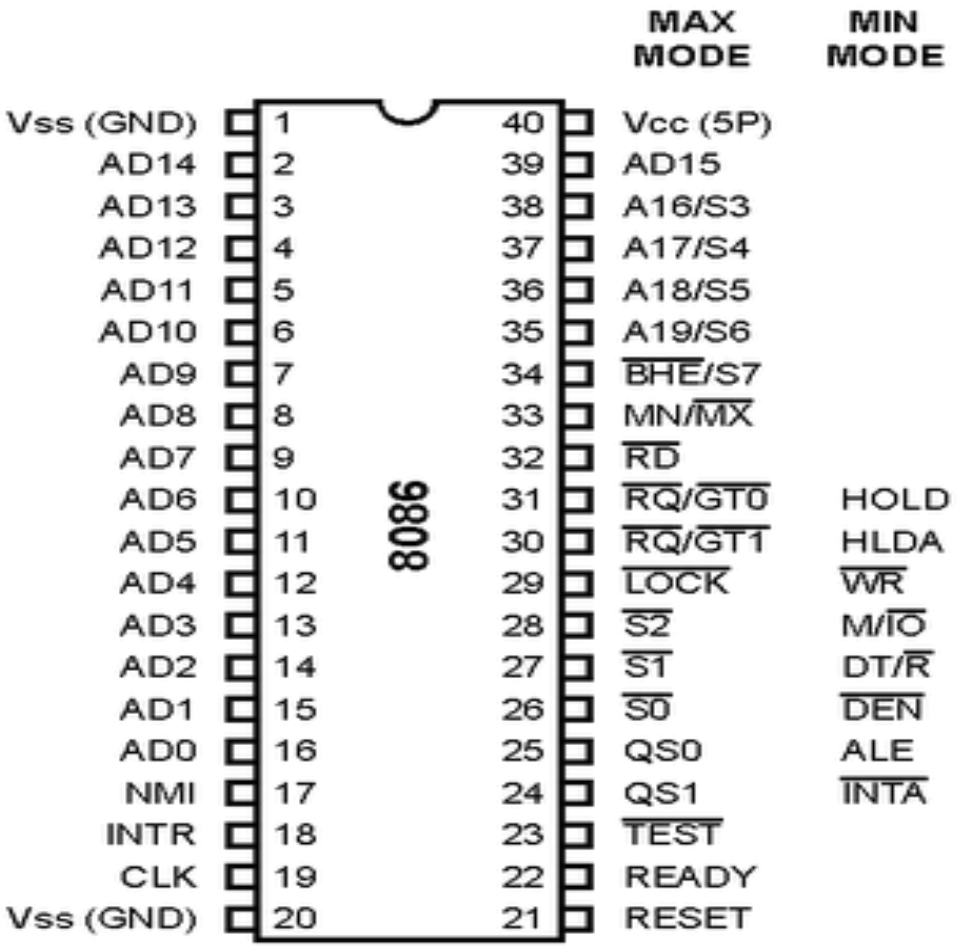
PA = 55AF2H

1.7 Describe Pins and Signals of 8086

The 8086 microprocessor is a 16-bit processor introduced by Intel in 1978. It is a 40-pin DIP (dual in line pin package) IC.

The 8086 operates in two modes, minimum mode and maximum mode. The signals are categorised into three types. They are

a) Common mode signals or pins (**13**)
b) Minimum mode signals (**7**)
c) Maximum mode signals (**4**)



Common signals for minimum mode and maximum mode:

1) AD_0-AD_{15} (address/Data): These are the time multiplexed memory I/O address and data lines.The address is available for T1 state and data will be available during T2, T3, Tw and T4 states. Tw is a wait state.

2) A_{16}/S_3,A_{17}/S_4, A_{18}/S_5,A_{19}/S_6 (Address/Status): These are the time multiplexed address and status lines. During T1, the higher order (A16-A19) address is available on these lines and during T2, T3 and T4 status information is available on these lines.

S4	S3	Indication
0	0	Extra
0	1	Stack
1	0	Code or none
1	1	Data

The status lines S_3 and S_4 are used to indicate memory segment which is presently accessing by the processor. These are also used to expand the memory upto 4Mb.

S_5 is used to indicate the status of the Interrupt flag bit and it is updated for every clock cycle.

S_6 pin is always low, which is not using presently.

3) $\overline{BHE}$/S_7(Bus High Enable) : This signal is used to indicate the transfer of data over D8-D15 and used in conjuction with address bit A0 (AD0) to select memory banks.

$\overline{BHE}$	A_0	Indication
0	0	Whole Word
0	1	Upper byte from or to odd address
1	0	Upper byte from or to even address
1	1	None

4) **$\overline{RD}$ (Read):** When this pin is low, it indicates that the processor reads data from memory or I/O devices.

5) READY: This is the acknowledge from the slow devices or memory that they have completed the data transfer.

6) INTR (Interrupt Request): This is a level triggered input. When the level triggered input present on this pin, the processor checks for the interrupt requests, if any interrupt request is pending, the processor enters interrupt acknowledge cycle. This can be internally masked by resetting the interrupt flag.

7) **$\overline{TEST}$:** This input is examined by WAIT instruction. If the **$\overline{TEST}$** goes low, execution will continue, else the processor remains in an idle state.

8) NMI (Non maskable Interrupt): This is a edge triggered input, which causes type2 interrupt. The NMI is not maskable by software. It is a high priority interrupt.

9) RESET: This input causes the processor to stop current activity and start execution from the location FFFF0H. It actually executes the program from the first instruction.

10) CLK (Clock): Clock signal provides the basic timing for processor operation and bus control activity. It is Asymmetric square wave with 33% duty cycle.

11) Vcc: +5V power supply for operation of the internal circuit of 8086.

12) GND(Ground) : Ground for internal circuit.

13) MN/$\overline{MX}$ (Minimum/Maximum): The logic level at this pin decides whether the processor is to operate in either minimum mode or maximum mode.

Minimum mode signals: If a single 8086 processor is used to design a microcomputer, then it is called minimum mode. In minimum mode of operation the 8086 microprocessor **MN/$\overline{MX}$** pin is connected to the Vcc (high level) and the processor minimum mode pins are given below.

1) **$\overline{INTA}$**: INTA stands for Interrupt Acknowledgement.It is a response for INTR.It is a active low signal.

2) ALE: ALE stands for Address Latch Enable when this output signal is high the 20bit address is available on time multiplexed address /data lines.

During T1 state address lines A0-A15 are seperated from data lines and appeared on the bus. During T2,T3,T4 states data is available.

3) DT/$\overline{R}$ (Data transmit/receive): This pin is used to decide the direction of data flow through buffers.

4) $\overline{\text{DEN}}$: DEN stands for data enable and is a active low signal.This indicates availability of a valid data over the address by data lines.It is used to enable the transeivers to separate the data from multiplexed address/data lines.

5) M/$\overline{\text{IO}}$: Whenever the microprocessor wants to communicate with memory then this pin connected to "1".If the microprocessor wants to communicate with IO ports this pin connected to "0".

6) **$\overline{\text{WR}}$**: This pin is active low pin,when 8086 writes data to memory or output device then it gives WR=0,otherwise WR=1.

7) HOLD&HLDA:When the hold signal goes high it indicates to the processor that another master is requesting the bus access.The processor after receiving the hold request,issues the hold acknowledgement signal on HLDA pin.

Maximum mode signals: If multiple processors along with 8086 are used to design a microcomputer, then it is called maximum mode.The 8086 processor work in maximum mode when the MN/MX pin is connected to ground.

1) **$\overline{\text{LOCK}}$**: LOCK signal is activated by using the name LOCK as prefix to any instruction.It is a active low signal when LOCK is activated then the system bus is not to be used by another master.

$\overline{\text{LOCK}}$ = 0 then bus is locked for only 8086

$\overline{\text{LOCK}}$ = 1 the bus is not locked,as that any system can share the bus.

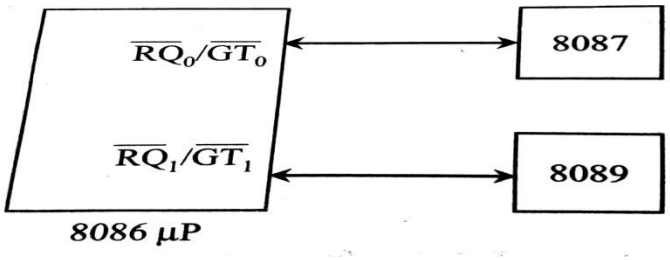
2) QS_0&QS_1: These pins are used to show the status of the queue.

QS_1	QS_0	Queue Status
0	0	No operation
0	1	First byte of opcode from queue
1	0	Empty queue
1	1	Subsequent byte from queue

3) **$\overline{\text{S0}}$/$\overline{\text{S1}}$/$\overline{\text{S2}}$**: These are staus signals.In maximum mode,the different control signals and command signals are generated using these status signals.

Status Inputs			Indication
$\overline{S}_2$	$\overline{S}_1$	$\overline{S}_0$	
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	Instruction Fetch
1	0	1	Read Memory
1	1	0	Write Memory
1	1	1	Passive

4) $\overline{\mathbf{RQ0}}/\overline{\mathbf{GT0}}$ &$\overline{\mathbf{RQ1}}/\overline{\mathbf{GT1}}$:RQ/GT stands for Request/Grant ,and is given below



The operation of RQ_1 signal is same as HOLD pin and operation of GT is same as HLDA pin.By using this pins the 8086 processor can share system bus with two other co-processors like 8087&8089.

- If any co-processor wants to share system bus it will send request signal($\overline{\mathbf{RQ}}$)to 8086.Then 8086 sends out grant signal ($\overline{\mathbf{GT}}$) to co-processor on the same pin.

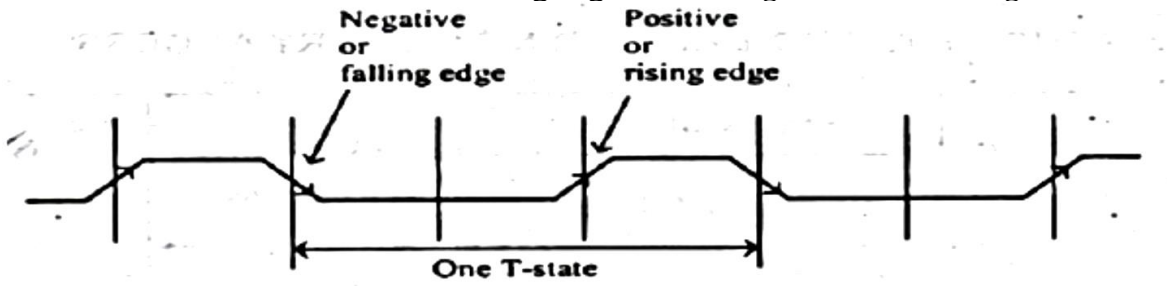
1.8 Illustrate the bus cycles of 8086

The 8086 processor has two functional units called Bus Interface Unit (BIU) And Execution Unit (EU). Most of the time each unit work independently. The BIU takes care of fetching instruction codes from memory, and data from memory and I/O devices. The EU takes care of executing the instructions pre fetched by BIU.

The **BIU initiates** all external operations which are also called **bus activity**. The external bus activities are **repetitions of certain basic operations**. The basic operations performed by the CPU bus are called **bus cycles**. Depending on the activities of 8086, the bus cycles can be classified as follows.

1. Memory read cycles (Four T-state)
2. Memory write cycle (Four T-state)
3. I/O read cycle (Four T-state)
4. I/O write cycle (Four T-state)
5. Interrupt acknowledge cycle (Eight T-state)

- The processor takes a definite time to perform a bus cycle.
- **The time taken to perform a bus cycle is specified in terms of T-states**.
- In 8086 processor the time duration of T-state is equal to one time period of the internal clock of the processor.
- The T-state **starts in the middle of falling edge of clock** signal as shown in figure.



Timing diagram:

The timing diagram **provides information about the various condition** (**high** state / **low** state / **high impedance** state) of the signals while a bus cycle is executed. The timing diagrams are essential for system designer. Only from the knowledge of timing diagrams, the matched peripheral device like memories, ports, etc., can be selected to form a system with microprocessor as CPU.

1. Memory read cycle:

- The memory read cycle is initiated by BIU of 8086 to read a program code or data from memory.
- The normal time taken by memory read cycle is four clock period.

The timings of various signals involved in reading a word (16-bit) starting from even addresses of memory in minimum mode are shown in fig., the activities of the bus in each T-state are given below.

Activities During T_1 state:

- During T_1 state the **ALE** pin is high and then it is low. The address is demultiplexed from the data lines and appered in the first state as $\mathbf{A_{15}}$ to $\mathbf{A_0}$.
- The **M/$\overline{\mathbf{IO}}$** is set to high to indicate memory access.
- The **DT/$\overline{\mathbf{R}}$** is set to low to receive the data.
- The $\overline{\boldsymbol{BHE}}$ is asserted low to enable odd upper memory bank.

Activities During T_2 state:

- The address is removed from the data and with a wait state data is available.
- The address is removed from the status lines and status information is available.
- The $\overline{\mathbf{RD}}$ asserted to low to read the data from the memory.
- The $\overline{\mathbf{DEN}}$ is asserted to low to enable the data bus buffers.

Activities During T_3 state:

- No activities are performed during T_3 state. The activities are remain same for T_3 as T_2.

Activities During T_4 state:

- The $\overline{\mathbf{RD}}$ is asserted high and at this time, The data is latched into 8086.
- The is$\overline{\mathbf{DEN}}$ made high to disable the data buffer.

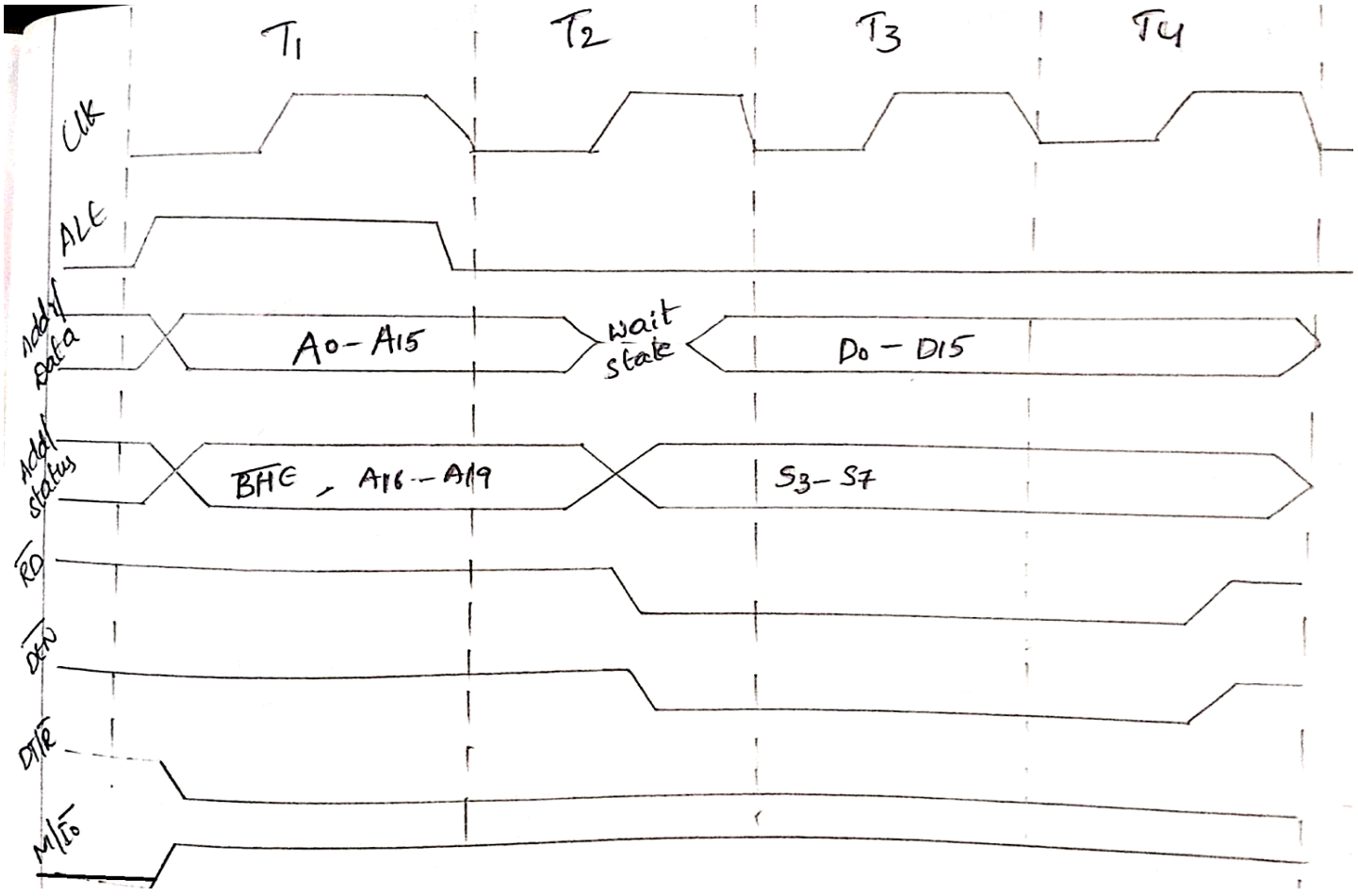


Fig: Memory read cycle

2. Memory write cycle:

- The memory write cycle is initiated by BIU of 8086 to write a data in memory.
- The normal time taken by memory write cycle is four clock periods.
- The timings of various signals is involved in writing a word(16-bit) starting from even address of memory in the minimum mode are shown in fig ., the activities of thebe's in each T-state are explains below.

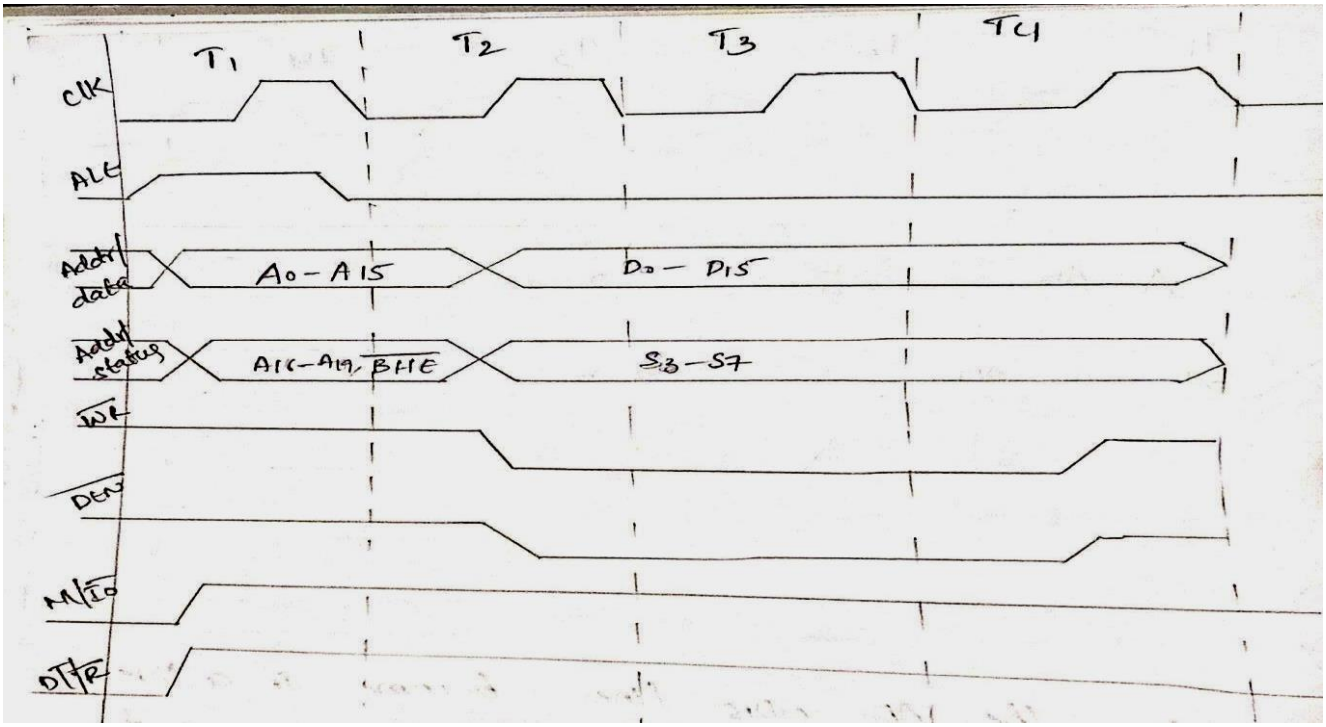


Fig: Memory write cycle

Activities During T_1 state:

- During T_1 state the **ALE** pin is high and the address is demultiplexed from the data lines and appered in the first state as $\mathbf{A_{15}}$ to $\mathbf{A_0}$.
- The **M/$\overline{\mathbf{IO}}$** is set to high to indicate memory access.
- The **DT/$\overline{\mathbf{R}}$** is set to high to transmit the data.
- The $\overline{\boldsymbol{BHE}}$ is asserted low to enable odd upper memory bank.

Activities During T_2 state:

- The address is removed from the data and with a wait state data is available.
- The address is removed from the status lines and status information is available.
- The $\overline{\mathbf{WR}}$ asserted to low to write the data on to the memory.
- The $\overline{\mathbf{DEN}}$ is asserted to low to enable the data bus buffers.

Activities During T_3 state:

- No activities are performed during T_3 state. The activities are remain same for T_3 as T_2.

Activities During T_4 state:

- The $\overline{\mathbf{WR}}$ asserted high and at this time, The data is latched into memory.
- The is$\overline{\mathbf{DEN}}$ made high to disable the data buffer.

3. I/O Read Cycle:

- The I/O read cycle is initiated by BIU of 8086 to read a data from I/O mapped device or I/O port.
- The normal time taken by I/P read cycle is four clock periods.
- The timings of various signals involved in reading an I/O port in the minimum mode are shown in the figure. The activities of the bus in each T-state are given below.

Activities during T1:

- The 8086 output a 16-bit I/O address on **AD_0-AD_{15}** lines. Logic low is output on the $\overline{BHE}$ and **ADDR/STATUS** lines.
- The **ALE** is asserted high and then low. This enables to latch the address and keep on their output lines.

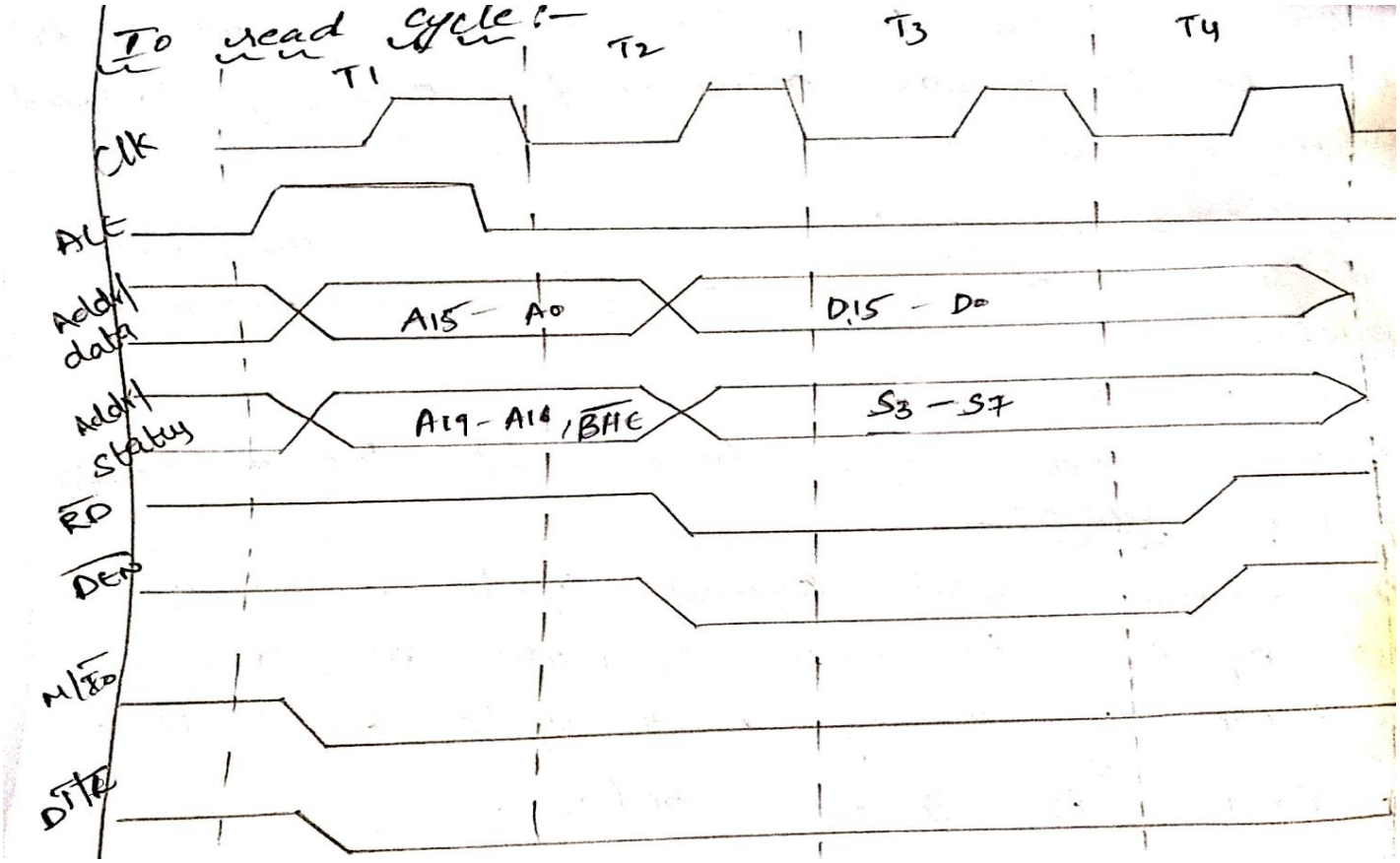


Fig: IO read cycle

- The **DT/$\overline{R}$** signal is asserted low to inform the external bi-directional data buffer that the processor has to receive data.
- The **M/$\overline{IO}$** signal is asserted low to indicate I/O access

Activities during T_2:

- The **AD_{15} –AD_0** lines becomes inactive
- The status singles **S_7-S_3** are issued on **ADDR/STATUS** lines
- At the end of T_2 the read control signal **$\overline{RD}$** is asserted low to enable the I/O device for read operating the time during which **$\overline{RD}$** remains low time allowed I/O device to load data in the bus
- The **$\overline{DEN}$** single is asserted low to enable the external bi-directional data buffer.
- The 8086 samples **READY** signal during T_2 (if ready is high then T_3 and T_4 are executed otherwise wait states are introduced)

Activities during T_3:

- No activities are performed during T_3 the status of the signal at the end of T_2 is maintained throughout T_3.

Activities during T_4:

- The **$\overline{RD}$** is asserted high and at this time the data is latched into 8086.
- The **$\overline{DEN}$** is mode high to disable the data buffer. \

4. I/O WRITE CYCLE:

- The I/O write cycle is initiated by BIU of 8086 to send a data to I/O device.
- The normal time take by I/O write cycle is four clock periods
- The timings of various signals involved in sending a word to I/O devious in the minimum mode are shown in fig:

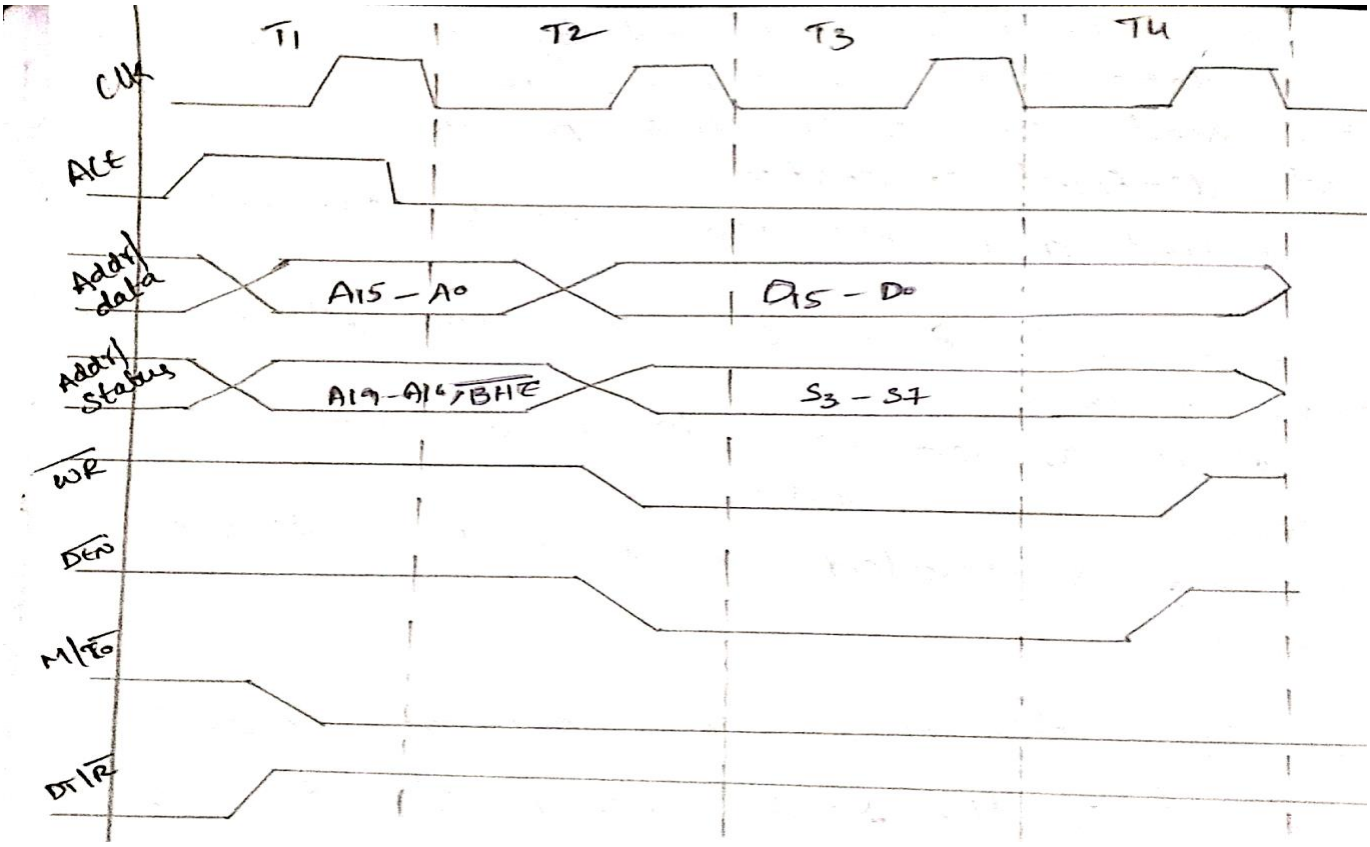


Fig: I/O Write Cycle

❖ The activities during I/O write cycle will be same as I/O read cycle except the following.

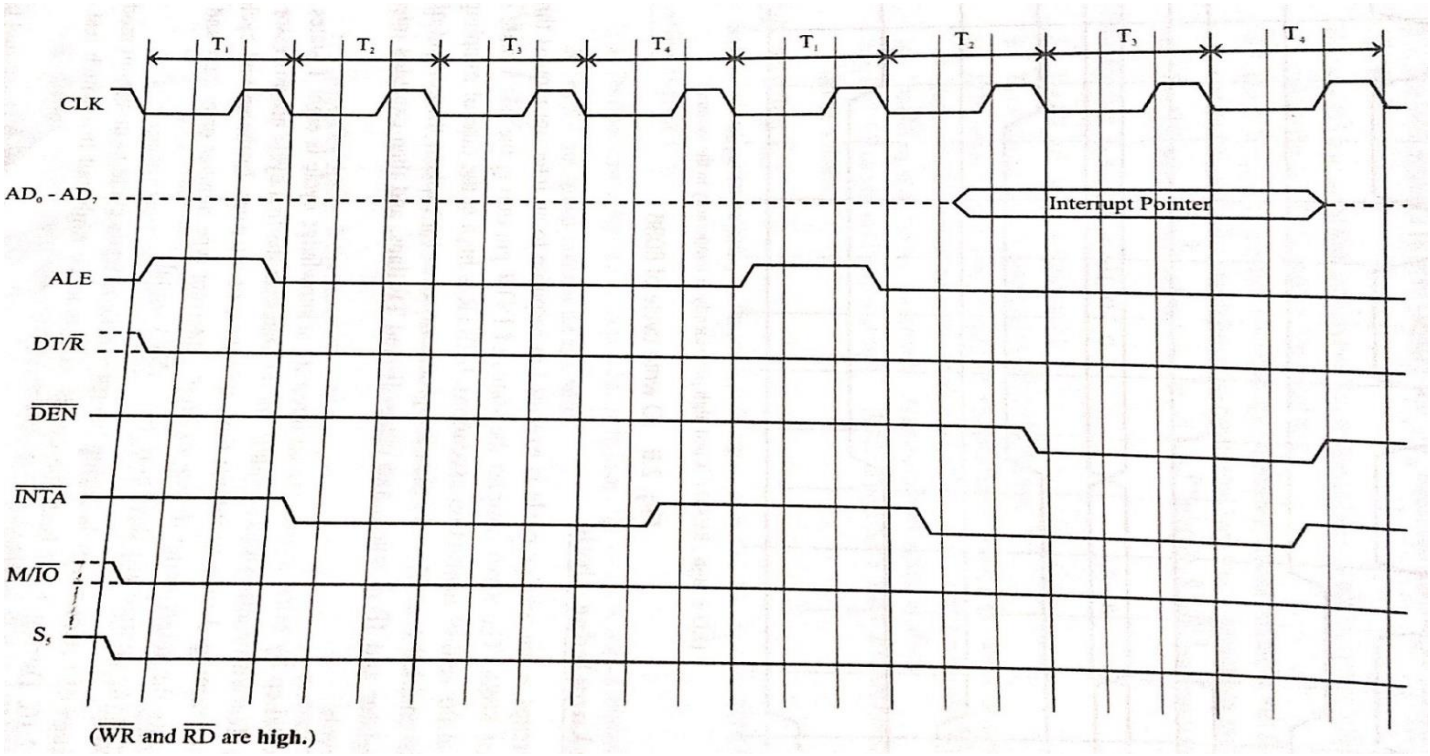
- During T_1, **DT/$\overline{R}$** is asserted high to inform the external bi-direction data buffer that the processor is going to transmit data
- During T_2 the address is withdrawn from **AD_{15}-AD_0** lines and data is output on these lines.
- The **$\overline{WR}$** is low to enable the I/O device for write operation.
- The **$\overline{DEN}$** is asserted low to enable the data buffer on the bus.
- During T_4 **$\overline{WR}$** is asserted high and at this time (i.e. at the rising edge of WR) the data latched into I/O devices.

5. INTERRUPT ACKNOWLEDGE CYCLE:

- The interrupt acknowledge cycle is executed in response to an interrupt request through the INTR pin 8086.
- The 8086 samples the status of INTR pin during the last T-state of an instruction (or at the end of instruction execution).
- If INTR is high at the time of sampling and interrupt flag is enable (i.e., if=1) then the processor save (or push) the content of flag register, register and IP in stack and clears IF and TF flags and then executes interrupt acknowledge cycle
- The time taken by 8086 to execute an interrupt acknowledge cycle is ***Eight*** **T-states**. It is actually two cycle with each cycle extending for **4 T-states.**

- In the first cycle the processor send **INTA** to the interrupting device to inform the acceptance of interrupt.
- In the second cycle the processor request the interrupting device to supply ***Interrupt Type Number or Interrupt Pointer*** and read type number form the interrupting device by INTA signal.

The timing of various signals during interrupt acknowledge cycle in minimum mode are shown in fig



- During T_1 of both the cycle ale is made high and low which results loading a junk value in address latches.
- During T_1 of first cycle **DT/R**, **M/IO** and S5 are asserted low the **DT/R** is asserted low to inform the data buffer that the processor has to receive data the **M/IO** is asserted low indicate I/O operating the S5 is asserted low the peripheral devices that the intern system is disabled (actually S5 is the status of interrupt flag).
- In both the cycles the **INTA** is asserted low during T_2 and then high during T_4 in second cycle when **INTA** is low the processor expects an 8-bit interrupt pointer on the low eight lines (**AD0-AD7**) of the data bus the time allowed to the interrupting device to load point is the time during which **INTA** remains low the processor samples the interrupt pointer on the rising edge of **INTA** signal in the second cycle.

Assignment Questions:

1) Explain Internal Architecture of 8086 with specific blocks?
2) Explain Pin description of 8086?

NOTE: Last date to submit : 05 – 12- 2019