Unit-4:- Microprocessors-I

8085 microprocessor Review (brief details only), 8086 microprocessor, Functional Diagram, register organization 8086, Flag register of 8086 and its functions, Addressing modes of 8086, Pin diagram of 8086, Minimum mode & Maximum mode operation of 8086, Interrupts in 8086.

---

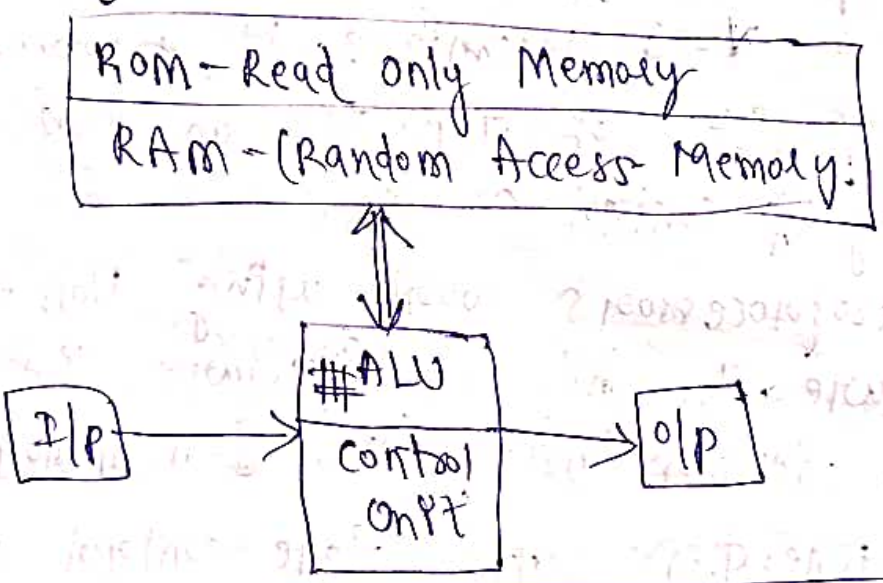
8085 Microprocessor:-

The microprocessor is a semiconductor device consisting of electronic logic circuit manufactured using LSI or VLSI technique. Microprocessor logically consists of registers on ALU, flipflops and timing & control circuits.

* All microprocessors work using Von Neumon Architecture. In this architecture the CPU or processor fetches instructions from memory decode it and generates appropriate control signal and finally executes it.
* The program is stored in consecutive memory locations. The execution steps are repeated for all the instructions of the program. Until execution is terminated by hardware or software.

VLSI = Very Large Scale Integration.

* A program is a list of instructions for the microprocessor to execute.
* Intel 8085 is an 8-bit microprocessor manufactured by intel corporation and is usually called a general purpose 8 bit processor.
* A microprocessor system consists of 3 functional blocks.
  ⇒ Central processing unit. (CPU)
  ⇒ Input & output units. (IO Unit)
  ⇒ Memory units. (MU)




## 8086 Microprocessor features:-

(i) 16 bit ALU
(ii) 16 bit data bus
(iii) 20 bit address bus
(iv) 14 sixteen bit registers
(v) 16 bit flag register.

(vi) Intel Corporation Manufactured this microprocessor (MP). (μP).

(vii) Rich in instruction set

(viii) Instruction byte queue present.

(ix) Clock - frequency 5MHz - 10MHz

(x) 40 pin Integrated circuit DIP structure.

## Register Organisation of 8086 μP:-

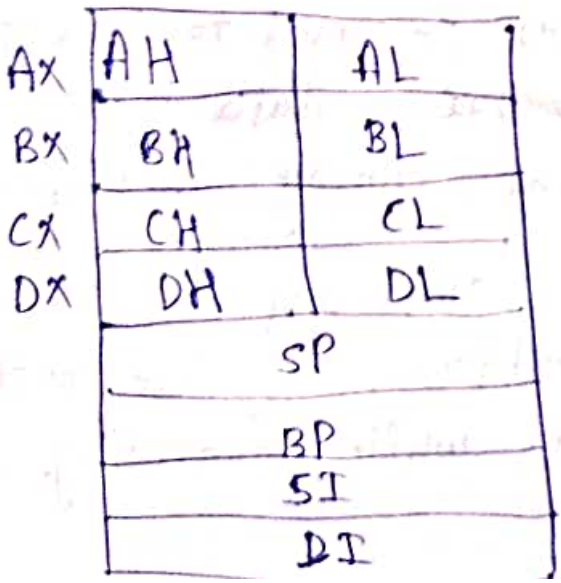
There are two types of registers present 8086 microprocessor.

(i) General purpose Registers.

(ii) Special purpose register (or) Segment Registers.

## General Purpose Registers:-

General purpose registers are in execution unit (EU). These are eight 16-bit registers and are used to store temporary data in different operations in micro processor. 8086 has the following eight general purpose registers.



AX:- This is the accumulator. It has 16-bit register but it is devided into two 8-bit registers. These are AH and AL.
Ax is generally used for arithmetic (or) logical instructions.

BX:- This is another register pair consisting of BH and BL. This is used to store the half-set values. (Offset

CX:- CX is generally has counter register. It has two parts CH and CL for different looping and counting purposes.

DX:- This is data register. The two parts are DH and DL and can be used in multiplication and division input, output addressing.

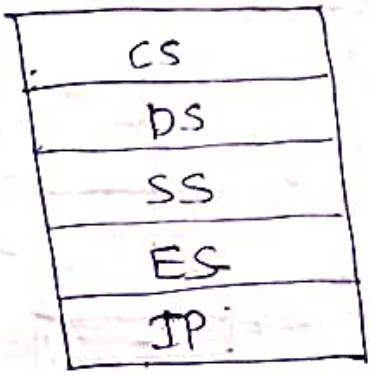
SP:- stack pointer points the top most element of the stack.

BP:- Base pointer is 16-bit register used to hold the half set address of the data to be read from (or) written into the stack segment.

SI:- Source Index register is used to hold the half set address of source data in the data segment, while executing string instructions.

DI:- The destination index register is used to hold the half set address of the destination data in extra segment, while executing string instructions.

~~Special Purpose Registers~~




## Segment Registers

There are 4 segment registers ( CS, DS, SS, ES)

In 8086 BIU - Bus Interface Unit. The function of these registers is to indicate the starting or base address of the code segment, data segment, stack segment, respectively in the memory. The code segment contains the instructions of a program and data segment contains the data for a program. The stack holds the stack of the program, which is needed while executing the call and return instructions and also to handle interrupts. The extra segment is the additional data segment that is used by some string instructions.

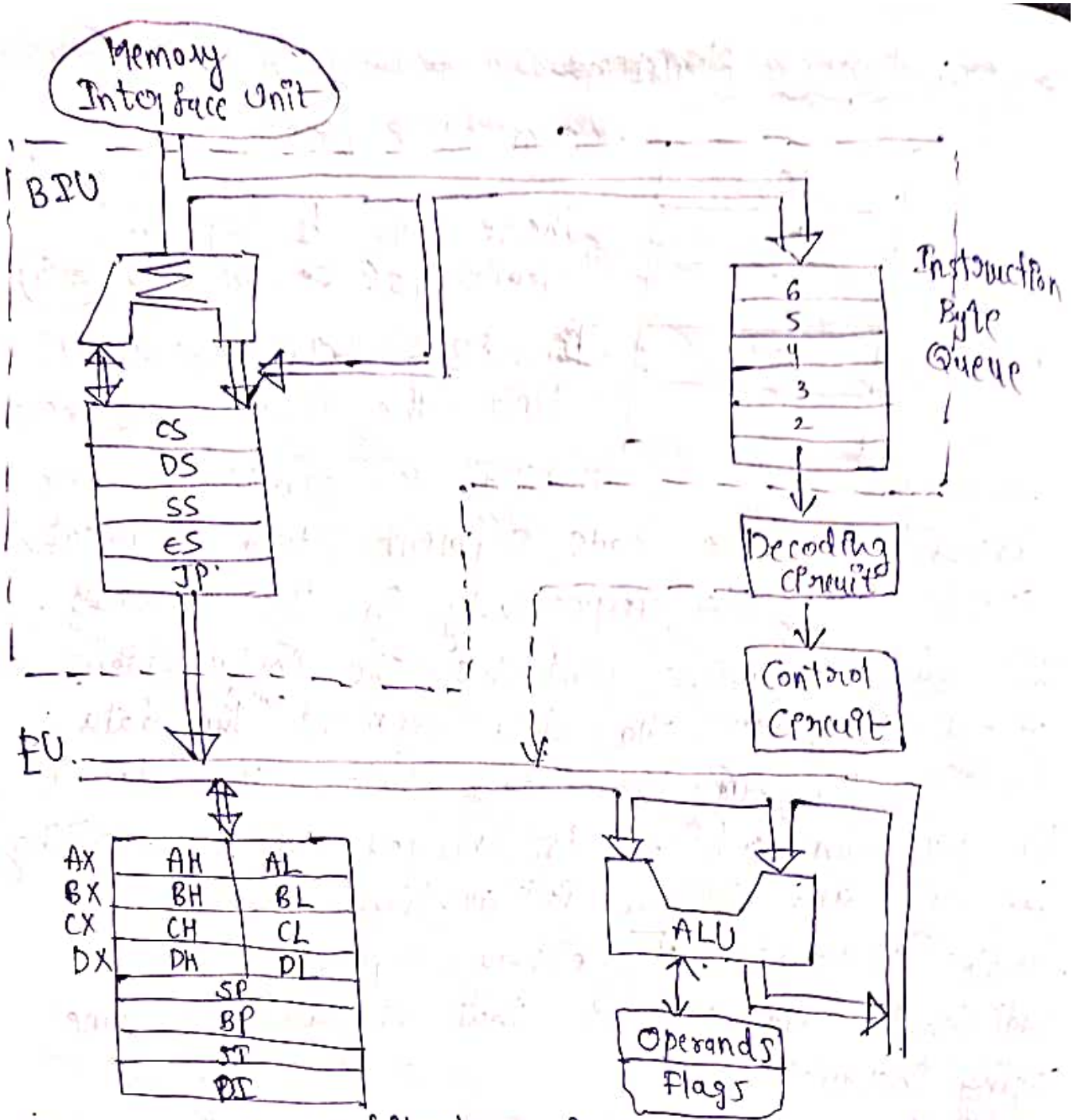
---

## 8086 Architecture:-

CS + IP

10H * CS + IP

1. Pipeline structure.




The complete architecture of 8086 can be divided into 2 parts. (i) BIU - Bus Interface Unit
(ii) EU - Execution Unit.

The BIU contains the circuit for physical address calculations and a free coding instruction byte queue (6-bytes long).

The BIU makes the system bus signals available for external interfacing of the device and peripherals including memory via bus.

The physical address calculated as

segment Address = 1005H

Offset Address = 5555H

segment address shifted by 4-bit position towards left

| | | | |
|---|---|---|---|
| 0001 | 0000 | 0000 | 0101 |
| 0101 | 0101 | 0101 | 0101 |

| | | | | | |
|---|---|---|---|---|---|
| Segment Address : | 0001 | 0000 | 0000 | 0101 | 0000 |
| Offset Address : | 0000 | 0101 | 0101 | 0101 | 0101 |
| Effective Address : | 0001 | 0101 | 0101 | 1010 | 0101 |
| physical Address. | 1 | 5 | 5 | A | 5 H |

(a) The instructions from the queue are taken for decoding sequentially, once a byte is decoded the queue is rearranged by pushing it out.

(b) While the opcode is fetched by BIU. The execution unit forms a decoded instructions concurrently. Thus BIU along with EU (execution unit) forms a pipeline.

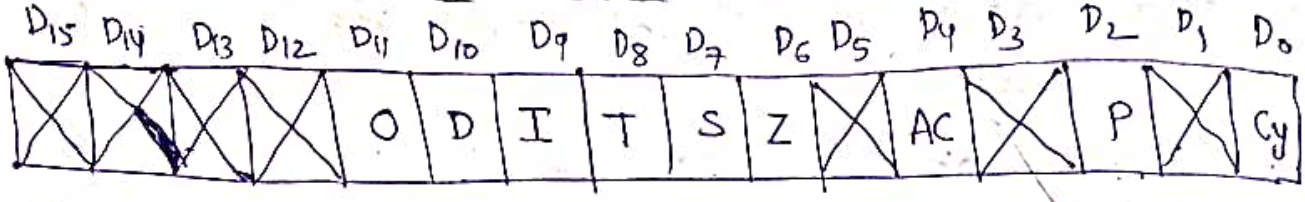
Register Organization :- (same as previous).

Flag Registers :- (same as explanation of all flags).

(5) The timing and control unit derives the necessary control signals to execute the instruction opcode received from the queue. Depending upon decoding circuit information.

(iv) The EU may pass the results to the BIU for storing them in memory.

## Flag Registers in 8086 Microprocessor:-




The flags in the flag register can be classified into status flags and control flags.

The flags CF (Cy) (Carry Flag), PF, AF, ZF, SF and OF are called status flags. as they indicate the status of the result that is obtained after the execution of an arithmetic (or) logic instructions.

The flags DF, IF and TF are called control flags. as they control the operation of the CPU. The function of these as follows.

### CF (Carry Flag):-

CF becomes set after an 8-bit (or) 16-bit addition (or) the borrow after an 8-bit (or) 16-bit subtraction.

### PF (Parity Flag):-

If the lower 8-bits of the result have an odd parity then PF is set to '0'. Otherwise it is set to '1'. (i.e, even parity)

### AF (Auxiliary Carry Flag):-

AF is set when the carry after addition (or

borrow after subtraction of the bits in the bit position - 3. This flag is used by the DAA and DAS instruction to adjust the value in AL after a BCD addition (or subtraction.

DAA = Data After Addition
DAS = Data After Subtraction.

ZF (Zero Flag):-

ZF indicates that the result of an arithmetic (or logical operation is '0'.

If ZF = 1, the result is '0'.

If ZF=0 the result is not '0'.

SF (Sign Flag):-

SF holds the arithmetic sign of the result after an arithmetic (or logic instruction is executed.

If S=1, the result is -ve.

If S=0, the sign bit is '0', the result is +ve.

TF (Trap Flag):-

TF is used to debug a program using the single step technique. If TF is set TF=1 the 8086 gets interrupted after execution of each instruction in the program. If TF is cleared TF=0 the debugging is disable.

DF (Direction Flag):-

DF selects the increment mode (or decrement mode for the DI (or SI register during the execution of string instructions.

D=0 register incremented.

D=1 register decremented.

The flag set (or) cleared using STD and CLD respectively.

IF (Interrupt Flag):-

It controls the operation of the INTR (Interrupt) Pin of the 8086. If IF=0 the INTR pin is disabled and if IF=1 the INTR pin is enabled. The flags can be set and clear using the STI and CLI instruction.

OF (Overflow Flag):-

Signed -ve numbers are represented in the 2's compliment form. In the microprocessor, when signed numbers are added (or) subtracted an overflow may occur. An overflow indicates that the result has exceeded capacity of the machine.

---

① Define opcode and operands in the 8086 microprocessor.

Opcode:-

Opcode (or) Operation code field which indicates the type of the operation to be performed by the CPU.

Operand:-

Operand is the information (or) data by which CPU executes the instructions.

19. What is meant by addressing modes? Explain sequential flow instructions and control transfer inst$^n$s with examples.

Addressing modes indicates a way of locating data (or operands depending upon the data types used in the instruction. There are different addressing modes. Any instruction may belong to one (or more or none of the addressing modes

Addressing modes describe the type of the operands and the way they are accessed by executing an instruction.

According to the flow of instruction and execution the instructions may be categorized. Sequential control flow instructions:-

These are the instructions each after execution transfer control to the next instruction appearing immediatly after it. i.e, in the sequence in the program.

Ex:- Arithmetic logical, data transfer and processor control instructions. are sequential control flow instructions.

Control Transfer Instruction:-

This instructions transfer control to some pre-defined address (or the address some how specified in the instruction after their execution.

INT, CALL, RET and JUMP instructions.

③ List various addressing modes in 8086 MP with examples.

Sol:- There are mainly 2 types of addressing mode
(i) Sequential Control Transfer Addressing modes
(ii) Control Transfer Addressing modes

Sequential Control Transfer Addressing Modes:-

(i) Immediate Addressing Mode:-

In this type immediate data is a part of instruction and appears in the form of successive bytes.

Eg:- MOV AX, 0005H

(ii) Direct Addressing Mode:-

In this addressing mode a 16-bit memory address offset is directly specified in the instruction as a part of it.

Eg:- MOV AX, [5000H] Here the data resides in a memory location in the data segment whose effective address may be computed using 5000H as the offset address and content of DS as segment address the affective address is 10H * DS + 5000H.

(iii) Register Addressing Mode:-

In register addressing mode the data is stored in a register and it is referred using the particular register.

Eg:- MOV AX, BX

(iv) Register Indirect Addressing Mode:-
In this data (or operand is determined in an indirect way using the offset register. This is register Indirect mode in this offset address of data is either BX. The default segment is DS (or ES. The effective address of data is

10H * DS + [BX].

Eg:- MOV AX, [BX]

(v) Indexed Addressing Mode:-
In this addressing mode offset of the operand is stored in one of the index registers. SI (or DI. The default segments are DS (or ES.

The effective address is 10H * DS + [SI]

Eg:- MOV AX, [SI]

(vi) Register Relative Addressing Mode:-
The effective address is formed by adding 8 (or 16-bit displacement with the contents of any one of the registers.

Eg:- MOV AX, 50H [BX]

The effective address is 10H * DS + 50H + [BX]

(vii) Based Indexed Addressing Mode:-
The effective address of data is formed in this addressing mode by adding content of a base register to the content of index register.

10H * DS + [BX] + [SI].

The default segment register ES to DS.
Eg:- MOV AX, [BX][SI]. (viii) Relative Based Indexed:-
(***). The effective address is formed by adding 8-or 16-bit displacement with some of contents of any one of the base register and any one of the Index register
Effective address is 10H * DS + 50H + [BX] + [SI]
MOV AX, 50H [BX][SI]

## Control Transfer Instruction:-

In this the addressing modes depend upon weather the destination location is within the same segment (or) a different one. There are 2 types of control transfer instruction.

(i) Intra Segment AM → Direct Intra Segment AM
→ Indirect Intra Segment AM

(ii) Inter Segment AM → Direct Inter Segment AM
→ Indirect Inter Segment AM

### Direct Intra Segment AM:-

In this mode control is to be transferred, lies in the same segment and appears directly in the instruction as an immediate displacement value. If displacement is of 8-bits. we term as short jump. If it is 16-bits, it is termed as long jump.

Indirect Intra segment AM:-
In this mode the displacement to which the control is transformed in the same segment. But it is passed to the instruction indirectly.

Direct Inter segment AM:-
In this segment the address to which the control is to be transformed in a different segment. i.e, one code segment to the another code segment Here, another code segment CS and IP of destination address are specified directly in the instruction.

Indirect Inter segment AM:-
In this also the address to which control is to be transformed, lies in a different segment and it is passed to the instruction indirectly.

The contents of different registers are given below. Find the effective addresses for different AMs.

Offset displacement = 5000H

[AX] = 1000 H
[BX] = 2000H
[BP] = 5000H
[SP] = 6000H
[SI] = 3000H
[DI] = 4000 H
[CS] = 0 00H
[DS] = 1000H
[SS] = 2000H
[IP] = 7000H

Sol:- (i) Direct : MOV AX, [5000H]

EA (or) phy. add = 10H * DS + 5000H

= 10H * 1000 + 5000

= 15000H

(ii) Register Indirect : MOV AX, [BX]

EA (or) phy. add = 10H * DS + BX

= 10H * 18000 + 9000

= 19000H

(iii) Register Relative : MOV AX, 5000 [BX]

EA = 10H * DS + 5000 + BX

= 10H * 1000 + 5000 + 2000

= 17000H

(iv) Based Indexed : MOV AX, [SI] [BX]

EA = 10H * DS + SI + BX

= 10H * 1000 + 3000 + 2000

= 15000H

(v) Relative Based Indexed AM :

MOV AX, 5000H [BX] [SI]

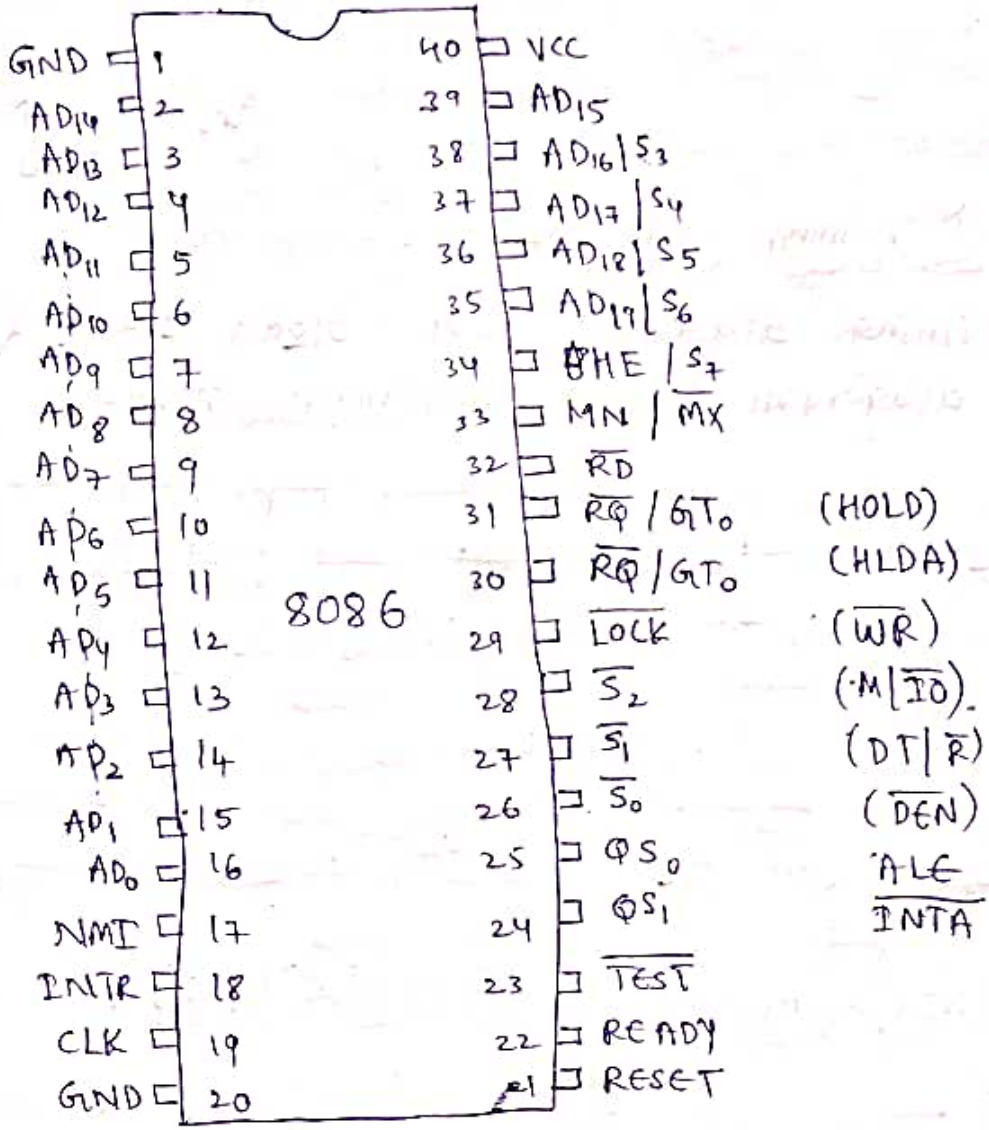
EA = 10H * DS + 5000H + BX + SI

= 10H * 1000 + 5000H + 2000 + 3000

= 1A000 ⇐ Hexadecimal

# pin diagram of 8086 Micro processor :-

Minimum Mode.




The micro processor - 8086 is a 16-bit CPU available in 3-clock rates (5, 8 and 10 MegaHz) and 40-pin plastic packages.

8086 - operates in single processor (min mode) (or) multi-processor max mode configurations to achieve high performance.

8086 signals can be categorized in to 3-groups.

(i) The signals having common function. in min as well as max mode. (1 to 23 & 33 to 40).

(ii) The signals which have special functions for min mode. (24 to 31)

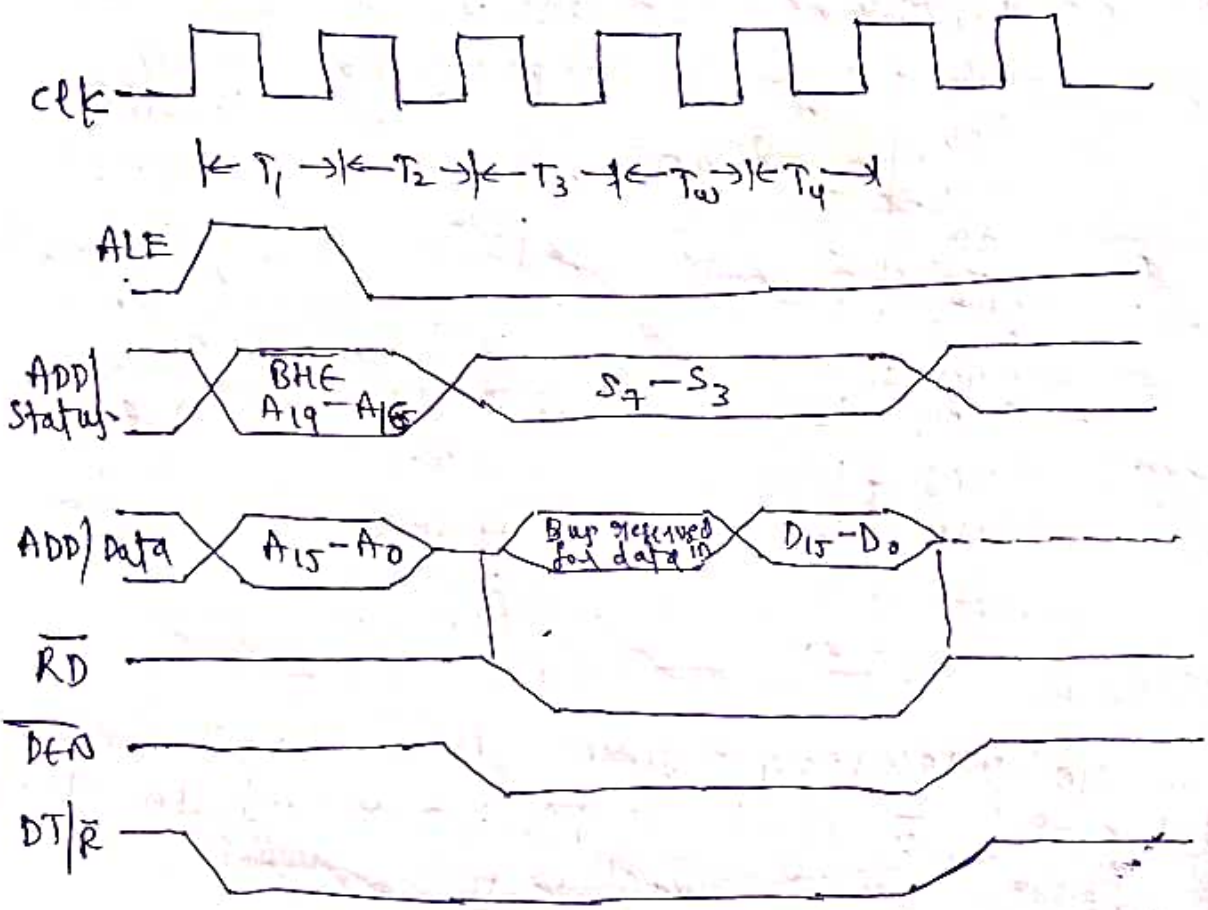
(iii) The signals which have special functions for max mode. (24 to 31)

Answer Recorded
ಉತ್ತರ ದಾಖಲಿಸಲಾಗಿದೆ.

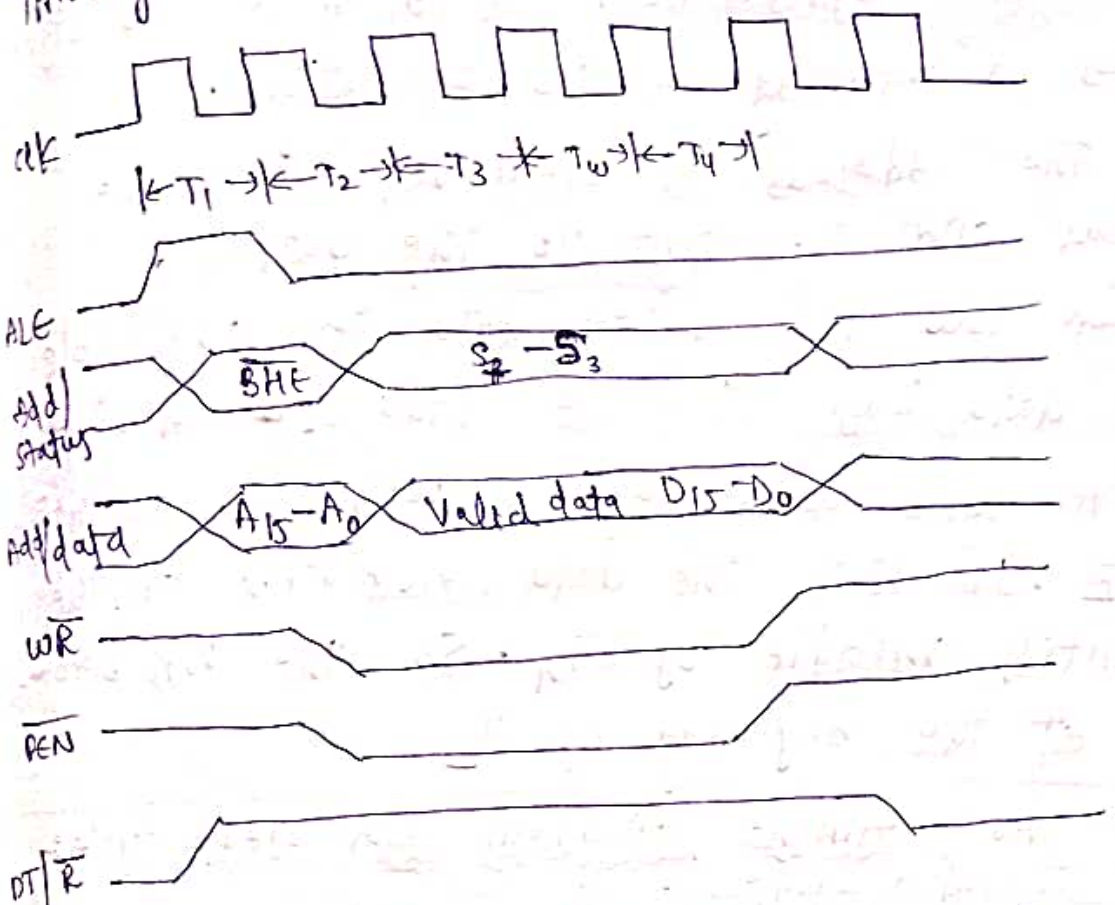
---

(i) Minimum Mode:-

Timing diagram for read cycle (Timing diagram for minimum mode).



Timing diagram for minimum operation (write cycle)




In the min mode all the control signals are given out by the micro processor chip it self. There is a single micro processor in the min mode system. The working of the min mode configuration system can be described by above timing diagram for read cycle and for write cycle.

The read cycle begins in $T_1$ with the assertion of the address latch enable (ALE) signal and M/$\overline{IO}$ signal. The valid address is latch on the local bus during $T_1$ state.

The $\overline{BHE}$ and $A_0$ signals are address low, high for both bytes. From $T_1$ to $T_H$ M/$\overline{IO}$ signal indicates a memory or IO operation.

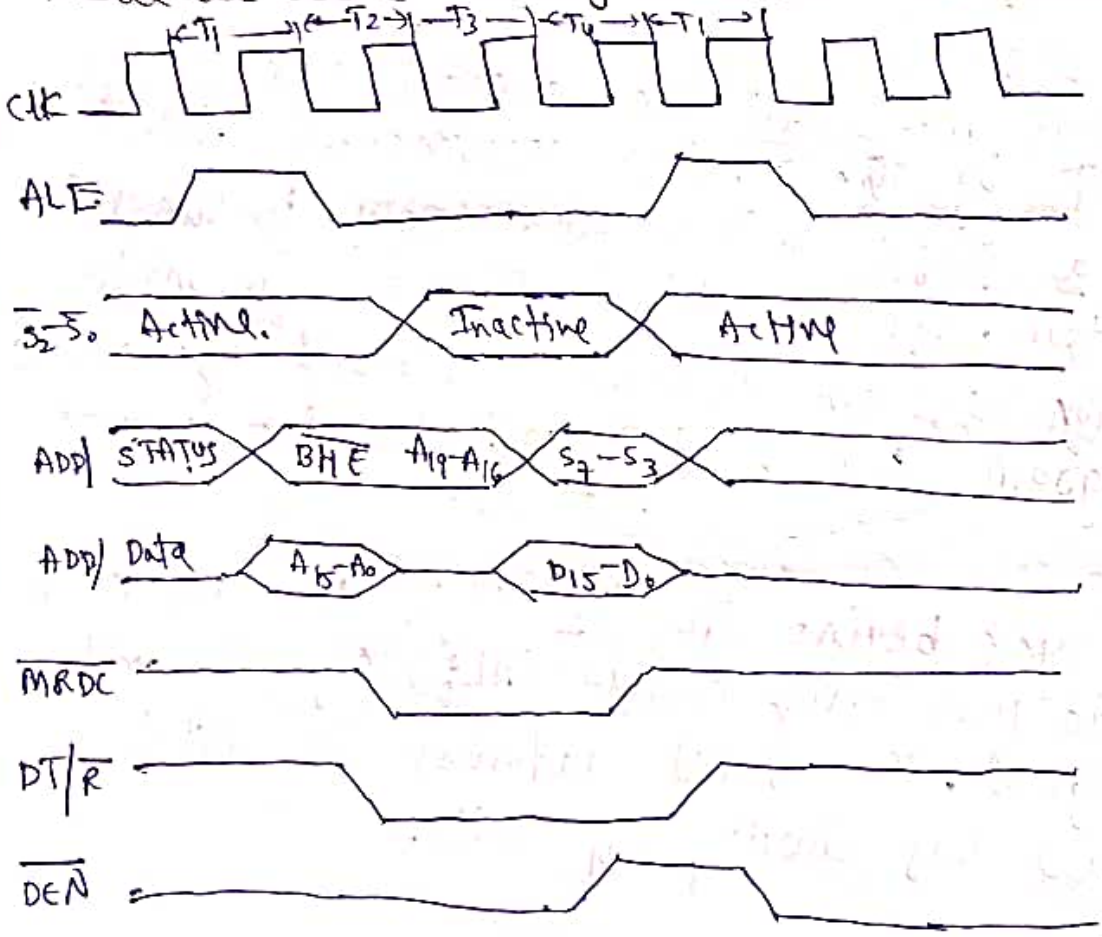
At $T_2$ the address is removed from the local bus and is sent to the output.

$\overline{RD}$ goes low the valid data is available on the data bus for the addressed device.

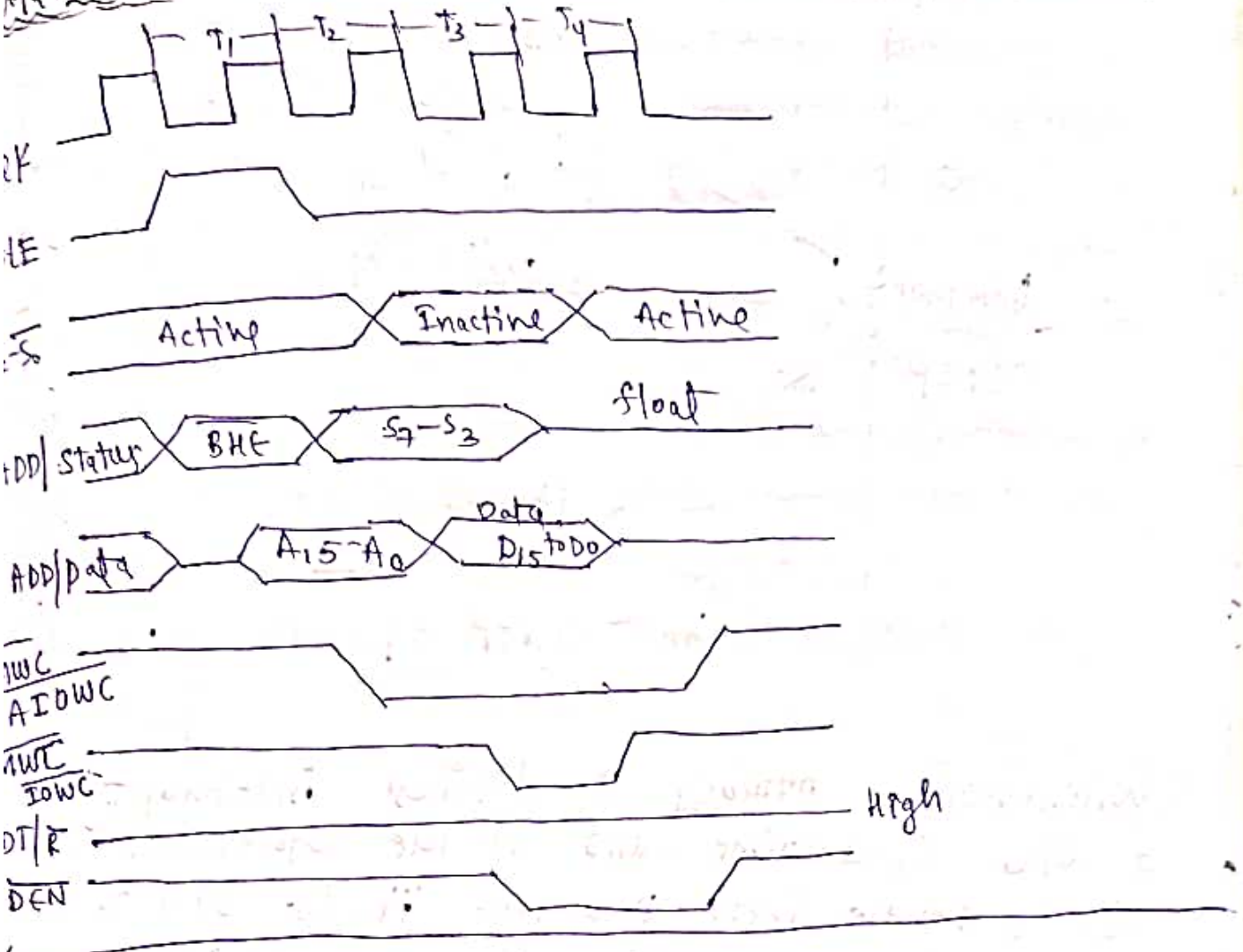
A write cycle also begins with the assertion of ALE. In this the data remains on the bus untill middle of $T_H$. So, the WR becomes active at the begining of $T_2$.

---

Maximum mode timing diagram for read cycle:-



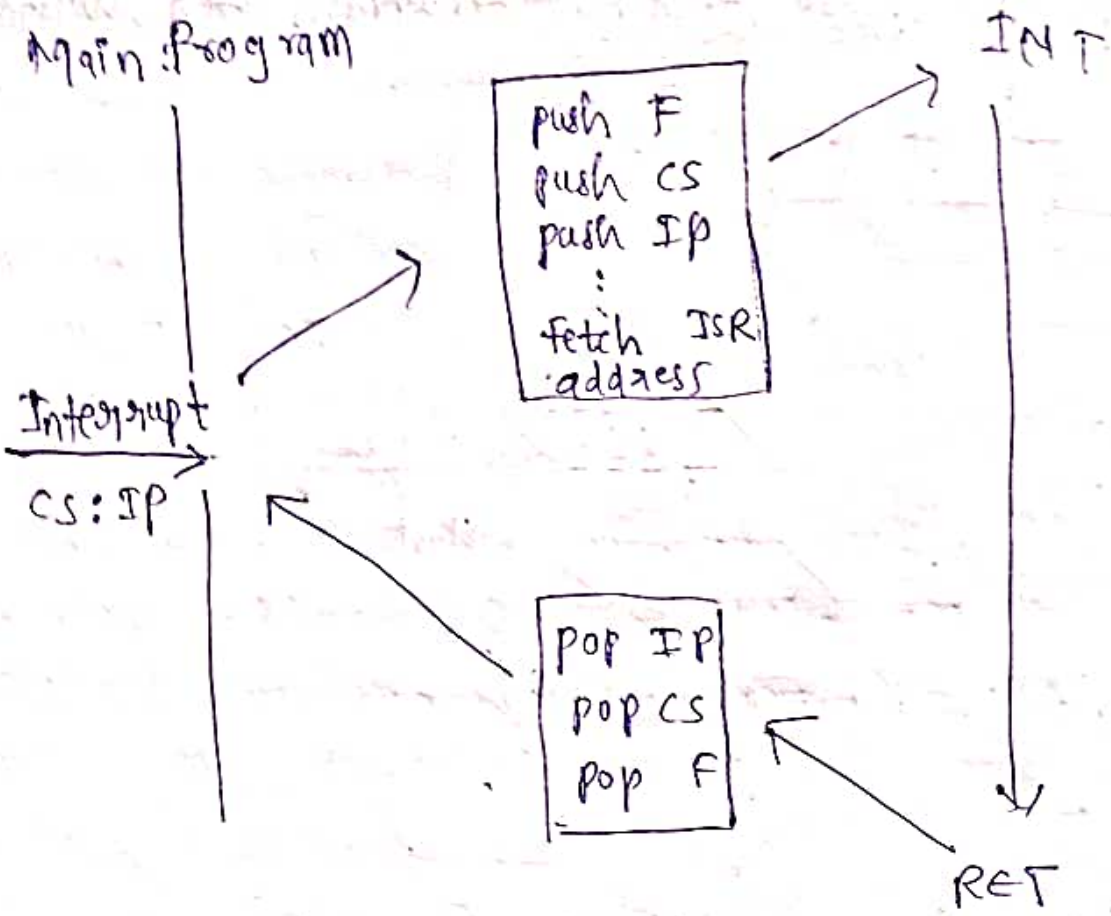
## Maximum mode timing diagram for write cycle:-




## Interrupts in 8086 Micro processor:-

1) Interrupt is to break the sequence of operation while the CPU is executing a program. An Interrupt breaks the normal sequence of execution of instructions diverse its execution to some other program called interrupt service routine.
2) After executing ISR the control is transfered back again to the main program which was being executed at the time of interruption.

ISR = Interrupt service Routine.

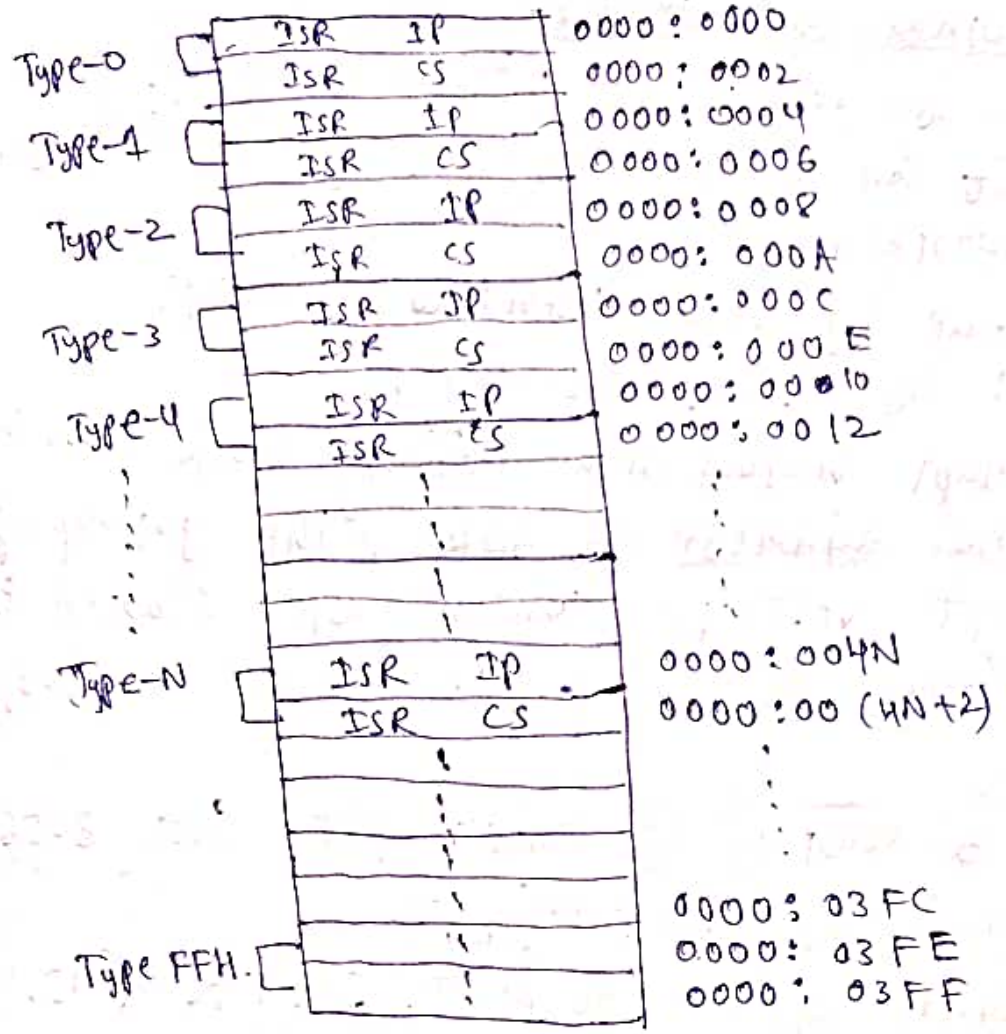



(i) Whenever a number of devices interrupt a CPU at a time and if the processor is able to handle them properly, it is said to be multiple interrupt processing capability.

(ii) In 8086 two interrupt pins NMI and INTR. The NMI is Non Maskable Interrupt input pin which means that any interrupt request at NMI input can not be masked or disabled by any means.

(iii) The INTR however may be masked using the interrupt flag. The INTR is of 00h to FFh [256 types. i.e 00h : 255]

## Structure of Interrupt Vector Table:-




(v) If more than one type of INTR (Interrupt) occurs at a time then an external chip called programmable interrupt controller is required to handle them.

(vi) ISR's are the programs to be executed by interrupting the main program execution of CPU. After execution of ISR, the main program continues its execution further from the point at which it was interrupted.

(vii) There are 2 types of interrupts

⇒ External Interrupt:
This is generated outside the processor

Ex:- A keyboard interrupt

⇒ Internal Interrupt:-
This is generated internally by the processor circuit (or) by the execution of an interrupt instruction.

Ex:- Divide by zero, Overflow Interrupt, INT instruction execution.

(vii) Interrupt vector is a memory block, which contains address of ISR, the group of interrupt vectors which are stored in a table form is called interrupt vector table.

(viii) The 3 sources of Interrupts for 8086

⇒ Hardware Interrupt:-
Signal applied to NMI, INTR pins.

⇒ Software Interrupts:-
By executing INT instructions.

⇒ Error in program execution:-
Divide by zero error, Overflow.