

Unit I : 8086 - 16 bit Microprocessor

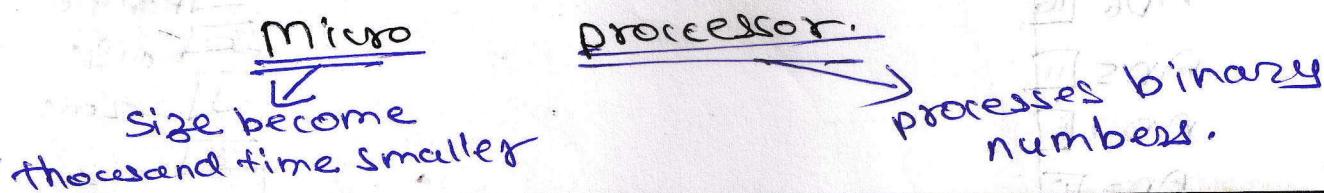
(Weightage - 14 marks)

Microprocessor

The word

- The word micro processor comes from combination of "micro" and "processor".

- Microprocessor is a multipurpose programmable device that accepts digital data as input, processes it accordingly to instructions stored in its memory and provides result as output. It is an IC.



1.1. 8086 Microprocessor : Salient features, Pin description

Questions . . .

① Function of following Pin. (2m)

- i) ALE
- ii) DT/R

② Function pin of 8086 : (4m)

- i) BHE
- ii) READY
- iii) ALE
- iv) RESET

③ State function. (2m)

i) READY & ii) INTR.

④ State function of BHE and its pins of 8086. (2m)

⑤ MN/MX and TEST. (2m)

⑥ READY i) ALE ii) TEST iii) DEN (4m).

Features of 8086

- * 16 bit CPU & 16 bit Data bus.
- * 20 bit Address bus with 2^{20} locations memory.
- * Works on single +5V power supply.
- * Supports pipelining architecture.
- * There are 14 16-bit registers.
- * Total 29000 transistors.
- * Has flag registers.
- * Supports two mode max 8 min.

| | | | | |
|------------|----|----------------------|----|------------|
| <u>GND</u> | 1 | (2700mV H - 920mV L) | no | VCC |
| AD14 | 2 | | sg | ADIS |
| AD13 | 3 | | sg | A16/S3 |
| AD12 | 4 | | sg | A17/S4 |
| AD11 | 5 | | sg | A18/S5 |
| AD10 | 6 | 8086 | 35 | A19/A8 |
| AD9 | 7 | | 34 | BHE/IS7 |
| AD8 | 8 | | 33 | PIN1/MX |
| AD7 | 9 | | 32 | RD |
| AD6 | 10 | | 31 | R>O |
| AD5 | 11 | | 30 | R/GTI |
| AD4 | 12 | | 29 | CLOCK |
| AD3 | 13 | | 28 | S&Z |
| AD2 | 14 | | 27 | EST |
| AD1 | 15 | | 26 | DEFL |
| AD0 | 16 | (ms) | 25 | SSP |
| AT&T | 17 | | 24 | TEST, INTA |
| INTR | 18 | | 23 | TEST |
| CLK | 19 | (mV) ~ 2800 to 3000 | 22 | READY |
| <u>GND</u> | 20 | MAX 3.6V | 21 | RESET |

- ① GND = pin 1, 20 = Ground Pin.
- ② VCC = pin 40 = +5V power supply.
- ③ CLK = pin 19 → This maximum clock frequency ranges from 5 to 10 MHz. This clock provides the basic timing for the processor & bus controller. It's asymmetric wave with duty cycle 33%.

- ④ AD0 - AD15 (2-16, 39)
- Address and data bus 8086 has 20-bit address-bus AD0-A15 and 16-bit data bus D0-D15. Lower 16 address lines AD0 to AD15 are multiplexed with data bus. Address bus and Data Bus are separated by using ALE signal.

NMI (Non-maskable interrupt) (Pin 1)

It is a single pin non-maskable hardware interrupt that cannot be disabled. It is highest priority interrupt in 8086.

An edge triggered signal on this pin causes 8086 to interrupt the program it is executing and execute interrupt service procedure corresponding to type 2 interrupt.

⑥ INTR (Pin 18)

→ A signal on this input causes 8086 to interrupt the program. It is executing and execute specified interrupt service procedure.

(F=0 (not occurred)) If F=1 (occurred).

⑦ ALE (Pin 25)

→ Address latch enable. It is control signal that enables the address bus.

This signal indicate the availability of valid address on AD lines.

⑧ BHE (Pin 34)

→ It is Bus high Enable. It is used to indicate transfer of data using data bus. (D8-D15)

This is used to read signal for read operation.

⑨ PT/R (Pin 27)

→ Data transmit/Receive. The output line is used to decide direction flow of data flow through transceivers.

When processor sends the data, the signal is high, when receive data low.

⑩ READY (Pin 22)

→ Whenever signal is high, it indicates that device is ready to transfer the data. It carries normal operations. When signal is low, freezes it power and enter in wait state.

⑪ RESET (Pin 21)

This input is used to reset 8086.

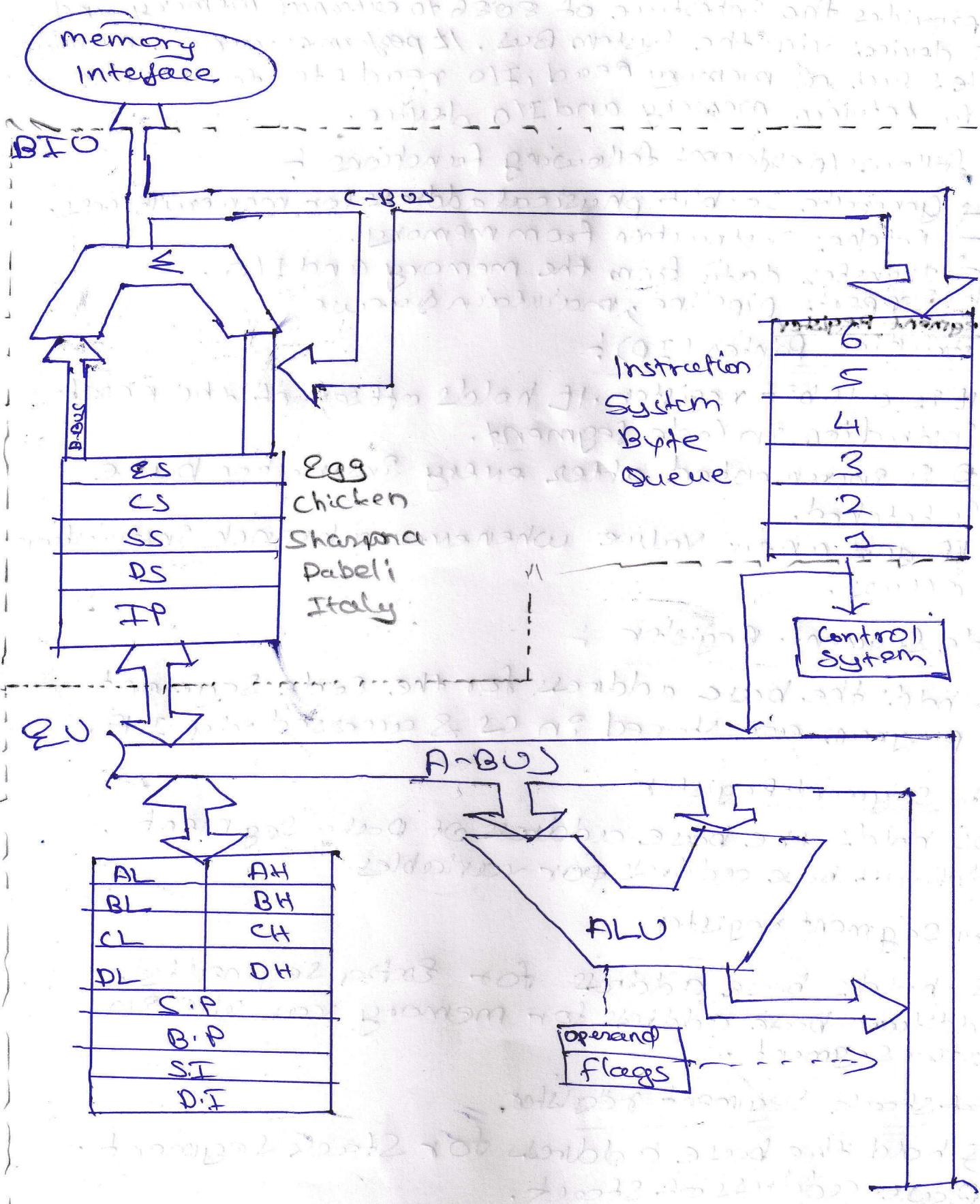
It terminate its operation when signal is high, resets all register.

- (12) TEST (Pin 23)
- This input is examined by 8086 'wait' instructions
 - If the TEST i/p is low, i.e. 0, execution continues.
 - If high, processor waits on an idle state.
- (13) DEN (Pin 26)
- Data Enable Signal.
 - Indicate availability of valid address (A₀₀-A₁₅) data.
 - Used to enable transceiver 8286.
 - Separate address / data bus.
- (14) MN/MX (Pin -33)
- It will select which mode is going to be executed.
 - When it is high \Rightarrow minimum mode, low \Rightarrow maximum mode.
- (15) INTA (Pin 24)
- Interrupt acknowledgement signal in response to INT R input signal.

2.2 Architecture of 8086: Functional Block Diagram, Register Organization.

- Questions --
- (Q1) Draw functional Block Diagram of 8086 microprocessor. (4m)
- OR
6 marks with Explain (Co=23)
- (Q2) Describe register organisation in 8086 CPU (4m)
- (Q3) Draw tabulated format of 8086 flag register. (2m)
- (Q4) Draw flag register of 8086 and explain any four flag registers. (4m)

Architecture / functional Block diagram of 8086 microprocessor



Working of BIU (Bus Interface Unit) ?

It provides the interface of 8086 to external memory and I/O devices via the System Bus. It performs various machine cycles such as memory Read, I/O read etc to transfer data between memory and I/O device.

It follows/perform following functions :-

- Generate 20-bit physical address for memory access.
- Fetches instruction from memory.
- Transfer data from the memory and I/O.
- Supports pipeline, maintain queue

Instruction Pointer (IP) :

- * It is a 16 bit register. It holds offset of the next instruction in Code Segment.
- * IP is incremented after every instruction byte is fetched.
- * IP gets a new value whenever a branch instruction occurs.

Code Segment Register :

- * CS holds the base address for the Code Segment.
- * All programs are stored in CS & accessed via IP.

Data Segment Register

- * DS holds the base address of Data Segment.
- * default base address for variables.

Extra Segment Register

- * ES holds base address for Extra Segments.
- * additional base address for memory variable in extra segment.

Stack Segment Register

- * SS hold the base address for Stack Segment.
- * Base address of Stack.

EU (Execution Unit)

- * The main components of EU are general purpose registers, the ALU, Special Purpose registers, Instruction registers and Instruction Decoder and Flag Status Registers.
- * Fetches instruction from Queue (in BIU), decodes and execute arithmetic and logic operations using ALU.
- * Sends control signal for internal data transfer operations with UP.
- * Sends request to BIU to access the external module.

8086 has four 16-bit general purpose register

AX, BX, CX & DX. Store information values during execution. Each of this have two 8-bit parts. (higher & lower).

General Data Registers

AX register

Used as 16-bit accumulator. The lower 8-bit is designated as AL and higher 8-bit is designated as AH. AL can be used as 8-bit accumulator for 8-bit operation.

BX register = 16 bit

BL BH
8

CX register

acts as counter for repeating or looping instruction

DX register: Used with AX to hold 32-bit values during multiplication, division operations. Used to hold I/O port address.

Special Purpose Register

SP (Stack Pointer) SP (Stack Pointer) points to stack top. Stack is in stack segment used during instruction like push, pop, call, ret etc. used to access memory locations.

BP (Base Pointer) Contains the offset of stack top. SP used with SS register to calculate 20-bit physical address

Base Pointer BP

BP holds offset address of any location in stack segment. It is used to access random locations on stack.

SI (Source Index)

It holds offset address in DS during operation. Used in string movement instruction. It holds offset address

DI (Destination Index)

acts as the destination for string movement instruction used to hold offset address of ES.

IP (Instruction Pointer)

Contains 16-bit offset address of instruction that is to be executed in CS.

Pointer & Index Registers

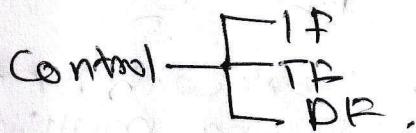
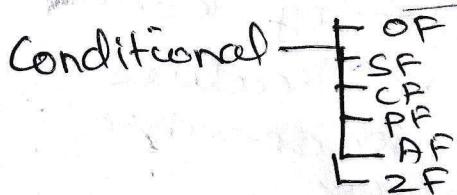
The pointer contains offset within the particular segments.

- The pointer register IP contain offset within CS.
- BP → DS
- SP → SS.

Flag register (Imp)

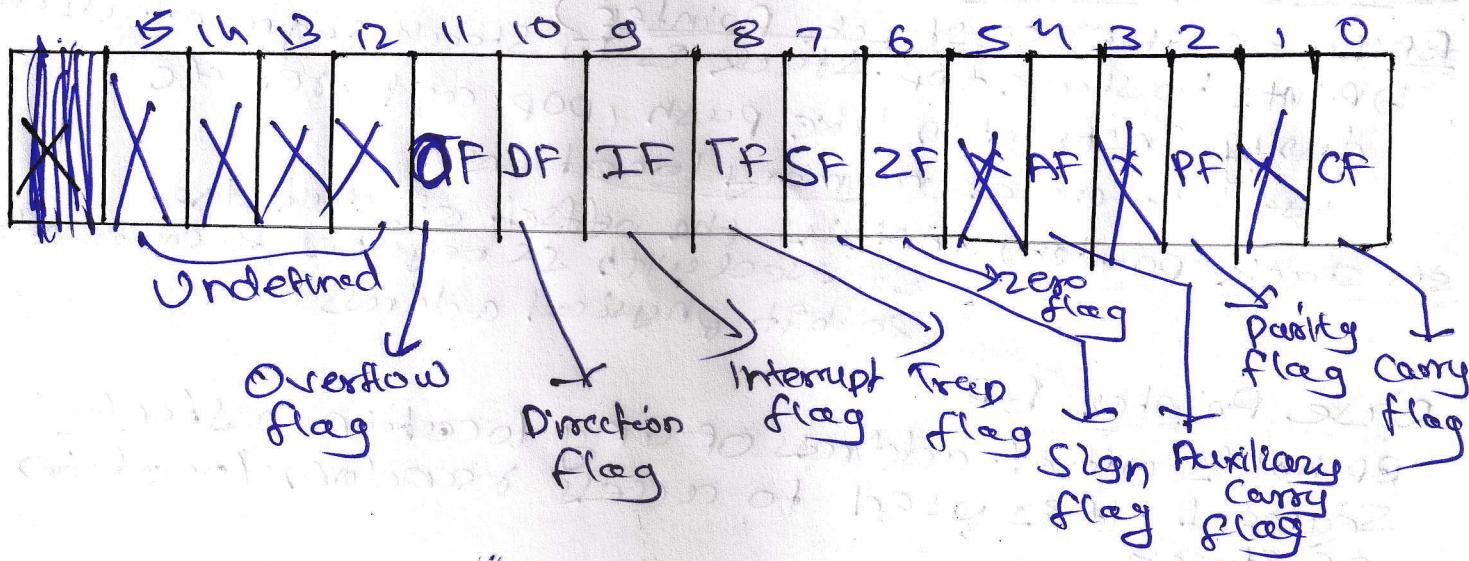
It is a flip flop which indicates some condition produced by execution of an instruction or control certain operations of CPU. It is 16 bit.

9 active flags, 6 are conditional & 3 control



Format

16 BITS 2 APC



① CF (Carry flag)

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

1 = set carry at msb
0 = No carry

* After certain arithmetic operations if carry is generated CF will be set to 1.

Example 1

$$\begin{array}{r} 0010 \\ + 0000 \\ \hline 0010 \end{array}$$

So, CF = 0

Example 2

$$\begin{array}{r} 0011 \\ - 1111 \\ \hline 0010 \end{array}$$

So, CF = 1

② Parity Flag

* The flag is set, if the lower byte/nibble of result contains even numbers of 1's.

* It's known as even parity.

* Odd parity then reset.

Example

$$\begin{array}{r} \text{Highest} \quad \text{lowest} \\ 00110011 \\ \cancel{\text{+}} \quad \cancel{\text{+}} \\ 00100000 \\ \text{Lp = even.} \end{array}$$

So, PF = 1

③ AF (Auxiliary carry flag)

During and computations or operations on two 8 bit or 16-bit numbers, if there is there carry from lower nibble to higher nibble the AF is Set.

Example

$$\begin{array}{r} \text{Highest} \quad \text{lowest} \\ \cancel{0000} \quad 1111 \\ + \cancel{0001} \quad 1111 \\ \hline 0010 \quad 1100 \end{array}$$

Carry

So, AF = 1

$$\begin{array}{r} 0000 \quad 0001 \\ + 0001 \quad 0101 \\ \hline 0001 \quad 0110 \end{array}$$

So, AF = 0

④ Zero flag

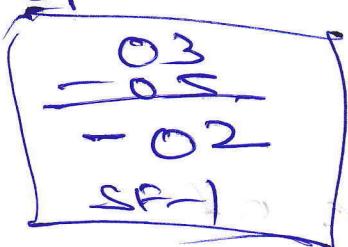
The flag is set if result of computation or comparison performed by previous instruction is zero.
Or, else, the flag is set to 0.

$$\begin{array}{r} \text{FFFF} \\ + \text{0001} \\ \hline \text{0000} \end{array}$$

$$ZF=1$$

⑤ SF (Sign flag)

It's set to indicate that result of computation or operation is negative. Example (2's complement) -



⑥ Overflow flag

The flag is set if overflow occurs.

* OF=1 i.e. overflow occurred
OF=0 i.e. overflow not occurred.

* Sum of two operands exceeds.

⑦ Trap flag

If the flag is set, the processor enters the single step execution else, whole program executes once. Act as debugger.

⑧ PF (Parity flag)

It selects either increment or decrement mode for DI or SI register during string instruction.

⑨ IF (Interrupt flag)

It is used to mask (disable) or enable the INT.

When IF=1, recognized INT.

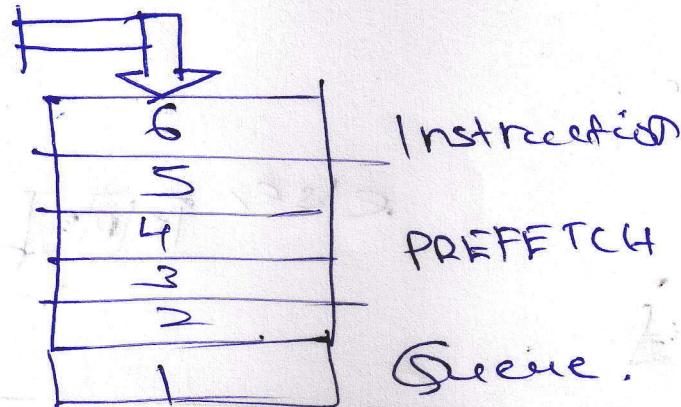
When IF=0, masked. (No interrupt).

1.3 Concepts of pipelining

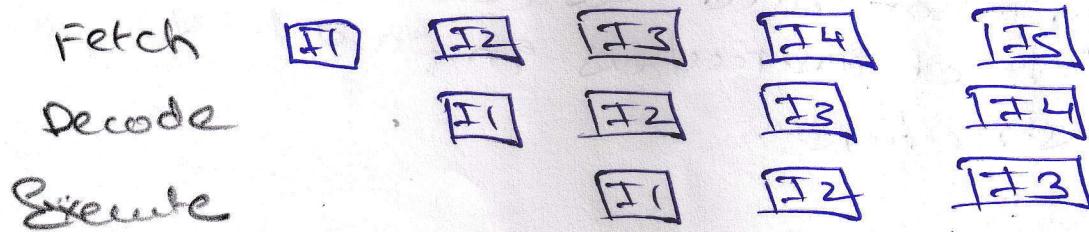
Questions.....

- Q1) Explain concept of pipelining in 8086. State advantages of pipelining (any two). (4m) 19-S
- Q2) What is pipelining? How it improves processing speed. (4m)
- Q3) Pipelining Concept. (2m) 19-W

- * In pipelined processor, fetch, decode and execute operation are performed simultaneously in parallel.
- * When first instruction is being decode, same time next fetches.
- * When first is getting execute, second one is decoded and third instruction code is fetched to memory.
- * This process is known as pipelining.
- * It improves speed upto great extent.
- * Fetching the next instruction while the recent instruction execute is known as pipelining.
- * To speed up program execution, BIU fetches as many 6 instruction bytes ahead of time from memory are held for EU unit in FIFOs or group register called QUEUE.



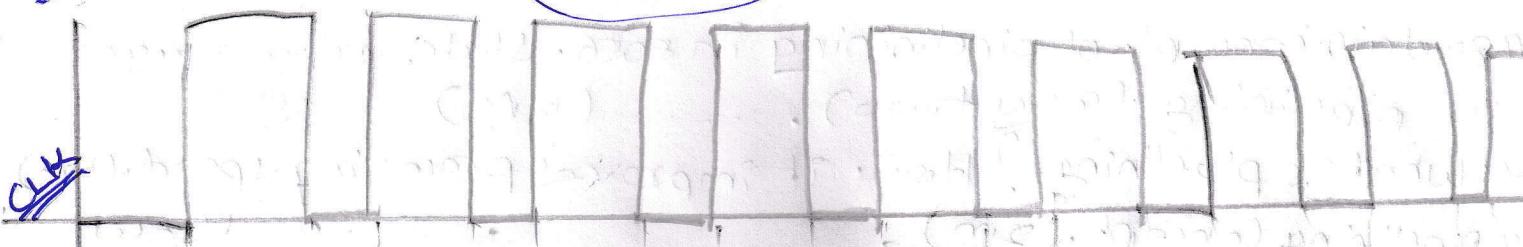
- * In 8086, pipeline technique of overlapping instruction fetch and execute mechanism.
- * It improves overall speed of CPU.



Without Pipeline

9 CLK cycles

P = PETCH
D = DECODE
E = EXECUTE



P D E

P D E

P D

with pipeline

6 CLK cycles

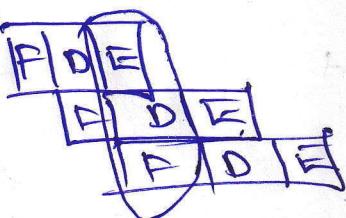
F D E

F D E

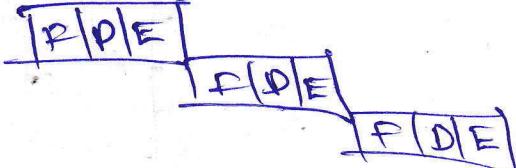
F D E

Saves time.

i.e.



else,



Advantages

- * performance improvement.
- * Saves time.
- * We are able to pump more instructions & get improved processor speed and we are able to execute parts of instruction in parallel to parts of other instructions.

1.4 Memory segmentation, Physical memory address generation.

Questions

Q1/ What is memory segmentation? Explain it with reference to 8086 CPU. (4m) ~~Advantages~~

Q2/ Define logical and effective address. Describe physical address generation process in CPU. Calculate physical address using DS, CS & IP. (6m)

Q3/ How 20bit physical address is generated in 8086. (4m)

Q4/ Define logical and effective address. Describe physical address generation process in 8086. If DS = 345A H and SI = 13 DC. Calculate physical address. (6m)

Q5/ Define logical & effective address. Describe how physical address generated in 8086. If CS = 348AH and IP = 4214H calculate physical address. (6m)

Memory Segmentation

* Segmentation is the process in which the main memory of the computer is logically divided into segments/ blocks and each segment has its own base address.

* It is basically used to enhance the speed of execution of computer system. So that the processor is able to fetch and execute the data from memory easily and fast.

* Total memory size is divided into segments of various size.

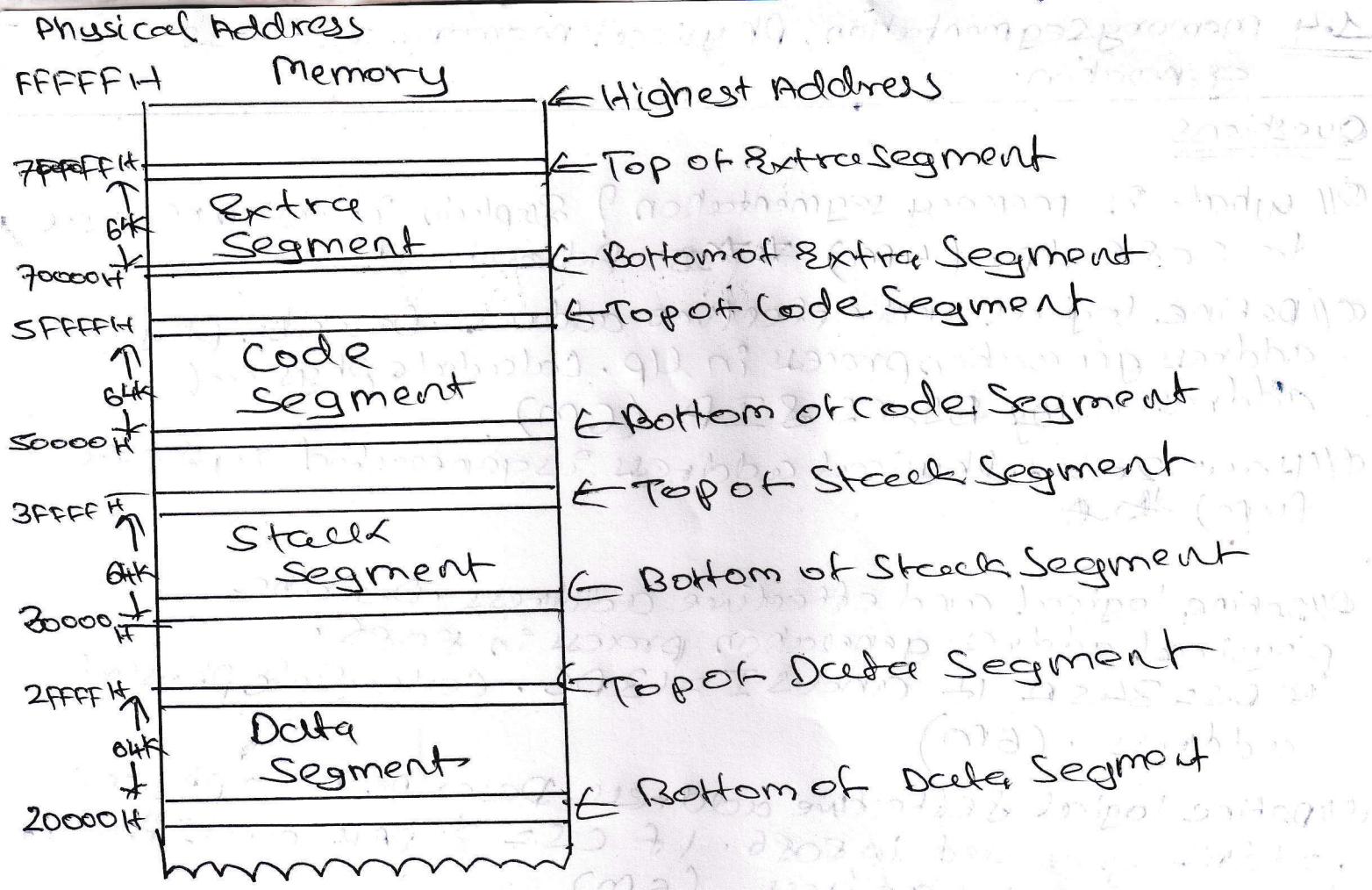
* Segment is nothing but just an area in memory. It has 20 lines address bus. 2^{20} bytes = 1 MB. (Total memory capacity = 1 MB)

* It can access by address ranging from 00000 H to FFFF H.

* At any given time, 8086 can only work with 4, 64 KB Segment

(CS, DS, ES, SS (16 bit each))

* They are used to hold upper 16 bit of starting address i.e. base address.



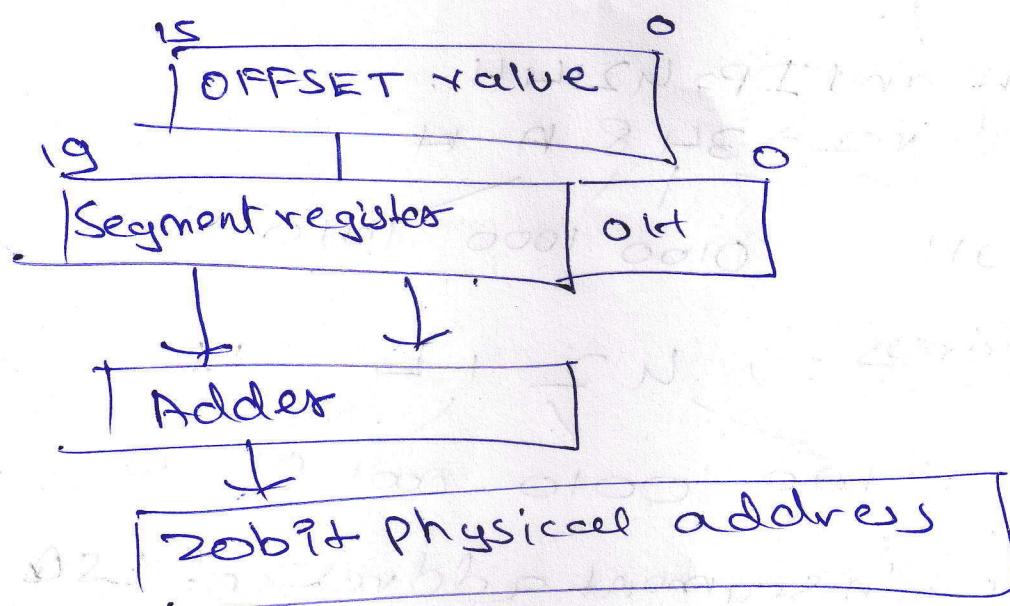
* The memory segmentation in 8086 CPU is organized as Segmented memory. This physical memory is divided into 4 segments namely - Data segment, Code Segment, Stack Segment and Extra Segment.

* Data Segment used to hold data, Code segment for executable program, Stack segment is used to store stack data and Extra Segment holds data specifically in strings ~~and stack~~.

- Each segment is 64 K bytes & addressed by one segment register.
 - The 16 bit segment register holds the starting address of segment.
 - The offset address to this segment is specified as 16-bit displacement (offset) between 0000 to FFFF H.
 - Since the memory size of 8086 is 1 MB, total 16 segments are possible with each having 64 KB.

Physical Address generation (20-bit)

- * Segment register carry 16 bit data which is also known as base address.
- * BIU appends four bits to LSB of the base/Segment address.
- * So thereafter, address (Base) becomes 20-bit.
- * Any pointer/base or index register carries 16 bit offset address.
- * Offset address is added into 20-bit base address which finally forms 20-bit physical address of memory location.



i.e

OFFSET address (16bit) + Segment address
+ 4bit(6)

Adding
physical address (20bit)

i) CS:1200H IP:0E001H

Segment address \rightarrow 1200H \rightarrow 0001 0010 0000 0000

Offset address \rightarrow 0E00H \rightarrow 1101 1110 0000 0000

Segment \rightarrow 0001 0010 0000 0000 0000
+ 1101 1110 0000 0000

0001 1111 1110 0000 0000

T F E 0 0 0

Q1) CS=348AH and IP=4214H.

\Rightarrow Segment address \rightarrow 348 AH

0011 0100 1000 1010

Offset address \rightarrow 4214H
0100 0010 0001 0100

Adding 4 0's in segment address at LSD to make it 20-bit.

0011 0100 1000 1010 0000
0100 0010 0001 0100

0011 1000 1010 1011 0100

3 A B 4

else

Physical Address = Segment address * 10 + offset address

$$= CS * 10 + IP$$

$$= 348AH * 10H + 4214H$$

$$= 38A84H$$

Q1) DS = 345AH, SI = 13DCH.

⇒ Segment address (16-bit)

3 4 5 A
↓ ↓ ↓ ↓
0010 0100 0101 1010

(Convert to 20-bit by adding 4 bits of 0's at LSB)

0010 0100 0101 1010 0000

Offset address

1 3 D C
↓ ↓ ↓ ↓
0001 0010 1101 1100

Adder

00101 0100 0101 1010 0000
+ 0001 0010 1101 1100
00101 0101 1000 0111 1100

3 5 8 7 C

Physical Address = 2587C H

logical Address

- Also known as virtual Address.
- It generated during/by CPU during program execution
- Logical address provide access for CPU access different locations in memory without needing to know the physical organisation of memory.

It is address at which item (memory cell, storage element) appears to reside from the perspective of execution application program.

A logical address is different from physical address due to the operation of address translator or mapping function.

Effective Address

- Also known as Offset Address
- The offset address for memory operand is called Operand Effective address or EA.
- It is an unsigned 16 bit number that express the operand's distance in bytes the beginning of the segment in which it reside.

{ location of an operand which stored in memory }