Chapter 2: 16 bit Microprocessor: 8086 [24 M]

Salient Features of 8086 Microprocessor

-16 bit Microprocessor
- 20 bit Address Bus Hence it can access 2^20=1 MB memory location
- Operating Clock Frequencies are 5 MHz, 8 MHz or 10 MHz
- Operate in maximum and minimum mode to achieve high performance level
- Can operate in single processor and multiprocessor configuration i.e operating modes
-Supports 24 addressing modes
- Supports Multiprogramming
-The instruction set is powerful, flexible and can be programmed in high level language like C language
-Provides 256 software interrupts
-Provide 6 byte instruction queue for Pipelining of instruction execution
-Arithmetic operation can be performed on 8-bit or 16-bit signed and unsigned data including multiplication and division
-Generate 16 bit I/O address so it can access maximum 64K I/O devices (2^16=65535)

Differentiate between 8085 and 8086

8085	8086
1) 8 bit microprocessor	1) 16 bit Microprocessor
2) 16 Address Lines	2) 20 Address lines
3) Total Memory supported 64 KB	3) Total Memory supported 1 MB
4) 8 bit operations can be performed using 8 bit Registers and ALU	4) 16 bit operations as well as 8 bit operations can be performed using 16 bit registers and ALU.
5) Operates in single mode	5) Designed to operate in two modes, maximum and minimum
6) Does not support Multiprogramming	6) Supports Multiprogramming
7) Does not support Pipelining	7) Supports Pipelining
8) Memory Segmentation is not implemented	8) Memory Segmentation is implemented
9) Supports 5 Hardware interrupts	9) Supports 2 Hardware Interrupts
10) Has 5 Addressing Modes	10) Has 24 Addressing Modes
11) Single Unit Microprocessor	11) Architecture has 2 units Execution Unit and Bus Interface Unit

Architecture or Block Diagram of 8086 Microprocessor

8086 has two blocks BIU (Bus Interface Unit) and EU (Execution Unit)

Functions of Bus Interface Unit

• The BIU performs all bus operations such as instruction fetching, reading and writing operands for memory and calculating the addresses of the memory operands.
• The instruction bytes are transferred to the instruction queue. BIU contains Instruction queue, Segment registers, Instruction pointer, and Address adder.
• BIU controls the address, data and control buses.
• It provides a full 16 bit bidirectional data bus and 20 bit address bus.
• The bus interface unit is responsible for performing all external bus operations.
• Instruction fetch, Instruction queuing, Operand fetch and storage, Address relocation and Bus control are the operations performed by BIU
• The BIU uses a mechanism known as an instruction stream queue to implement a pipeline architecture.

Functions of Execution Unit

• EU executes instructions from the instruction system byte queue.
• Both units operate asynchronously to give the 8086 an overlapping instruction fetch and execution mechanism which is called as Pipelining. This results in efficient use of the system bus and system performance.
• EU contains Control circuitry, Instruction decoder, ALU, Pointer and Index register, Flag register.
• The Execution unit is responsible for decoding and executing all instructions.
• The EU extracts instructions from the top of the queue in the BIU, decodes them, generates operands if necessary, passes them to the BIU and requests it to perform the read or write bus cycles to memory or I/O and perform the operation specified by the instruction on the operands.
• During the execution of the instruction, the EU tests the status and control flags and updates them based on the results of executing the instruction.
• If the queue is empty, the EU waits for the next instruction byte to be fetched and shifted to top of the queue.
• When the EU executes a branch or jump instruction, it transfers control to a location corresponding to another set of sequential instructions.
• Whenever this happens, the BIU automatically resets the queue and then begins to fetch instructions from this new location to refill the queue.
• EU has a 16 bit ALU, which can perform arithmetic and logical operations on 8 bit as well as 16 bit data.

The Instruction Queue:

To implement any instruction first it is to be fetched, then decoded and then executed. The fetching of an instruction involves its address to be sent out to the system memory and then the memory sending back the instruction.
While the EU is busy decoding or executing certain instructions which do not need the buses, the BIU fetches next six instruction bytes and stores them in a first-in-first-out (FIFO) register set called queue.
The processor doesn't have to wait for the next instruction to be fetched as it is already made available in the queue registers. Thus, the speed of operation is enhanced.

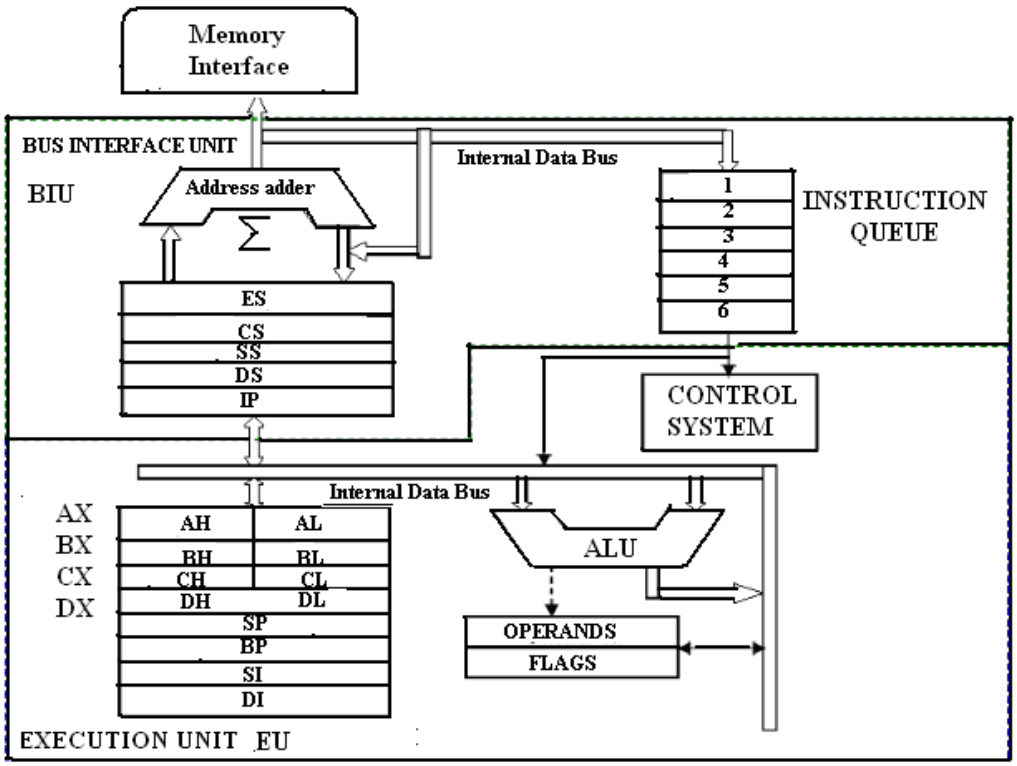
This prefetching of next instruction while another instruction is still being executed is known as pipelining.

General-purpose Registers:
There are eight 8-bit general-purpose registers: AL, AH, BL, BH, CL, CH, DL and DH. These can be used for temporary storage of 8-bit data. They can also be used for storage for 16-bit data words as groups: AX register (AH and AL), BX register (BH and BL),CX register (CH and CL) and DX register (DH and DL).

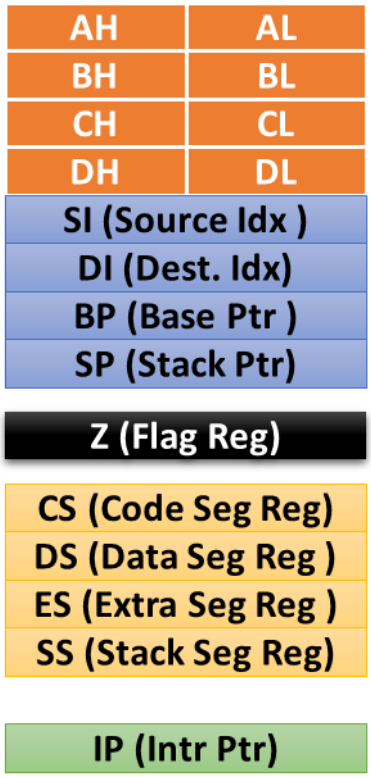
Flags:
The 16-bit flag register of 8086 contains 9 active flags (six conditional & 3 control), other 7 flags are undefined.
Conditional Flags: indicate certain condition that arises during the execution. They are controlled by the processor.
Control Flags: control certain operations of the processor. They are deliberately set/reset by the user.

Other Registers:
The Segment Registers- SS, DS, CS and ES, the pointer and index registers -BP, SP, SI and DI. and the Instruction Pointer (IP) will be discussed along with 8086 memory.
ALU:- ALU is used to perform arithmetic and logical operations

Address generator: - This unit is used to generate 20 bit physical address by adding 16 bit logical address displacement with base address
Instruction Decoder:- This unit is used convert or decode the instructions and provides signals to various units in the EU.



Register Organization in 8086 Microprocessor



- **AX** - Accumulator register (divided into **AH / AL**)
- **BX** - Base address register (divided into **BH / BL**)
- **CX** - Count register (divided into **CH / CL**)
- **DX** - Data register (divided into **DH / DL**)
- **SI** - Source index register.
- **DI** - Destination index register.
- **BP** - base pointer.
- **SP** - stack pointer.

Above four 16-bit registers can also be used as eight bit general purpose registers AH,AL ,BH, BL, CH, CL, DH, DL. These are used for 8-bit operations

Pointer and Index Group Registers

There are two 16 bit pointer registers:
Stack Pointer (SP) and Base Pointer (BP)

There are two Index Registers:
Source Index (SI) and Destination Index (DI)
These four registers can be used as general purpose registers in arithmetic and logic operations. SP and BP are pointer register which holds 160bit offset within the particular segment.

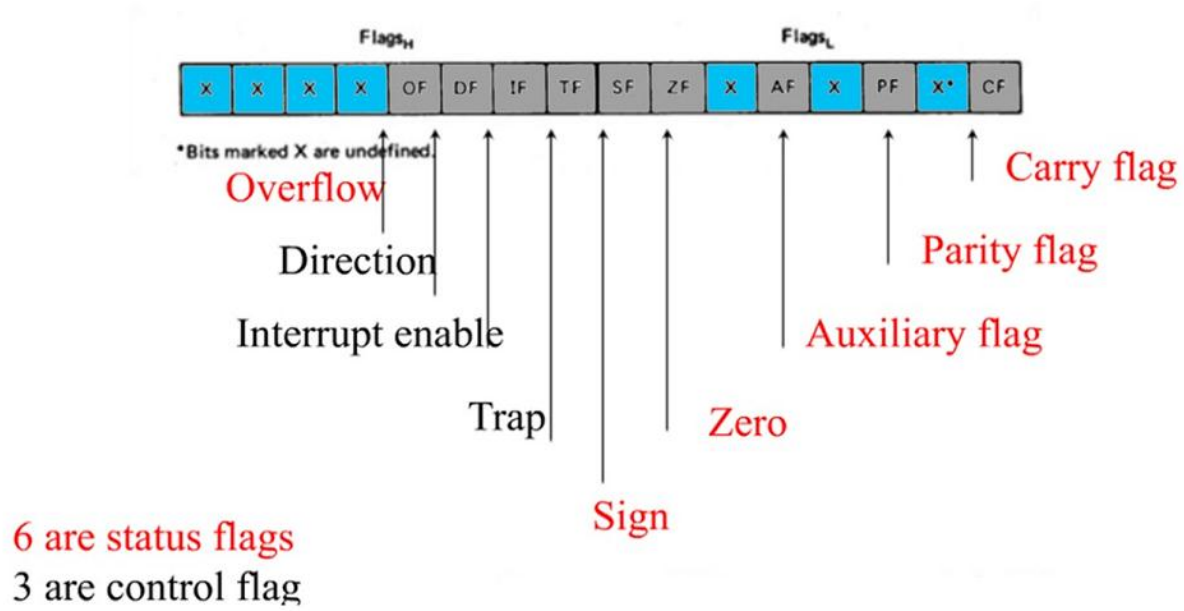
Instruction Pointer(IP) : The Instruction Pointer always points to the next instruction to be carried out from the program memory. It is 16-bit register.

SI and DI
During the execution of string related instructions, register SI is used to store the offset of source data/string in data segment while the register DI is used to store the offset of destination in Data/Extra Segment.

Segment Registers:
BIU has 4 segment registers of 16 bits each i.e CS, DS, SS and ES
-The memory pointers are used to point/address particular memory location in memory.

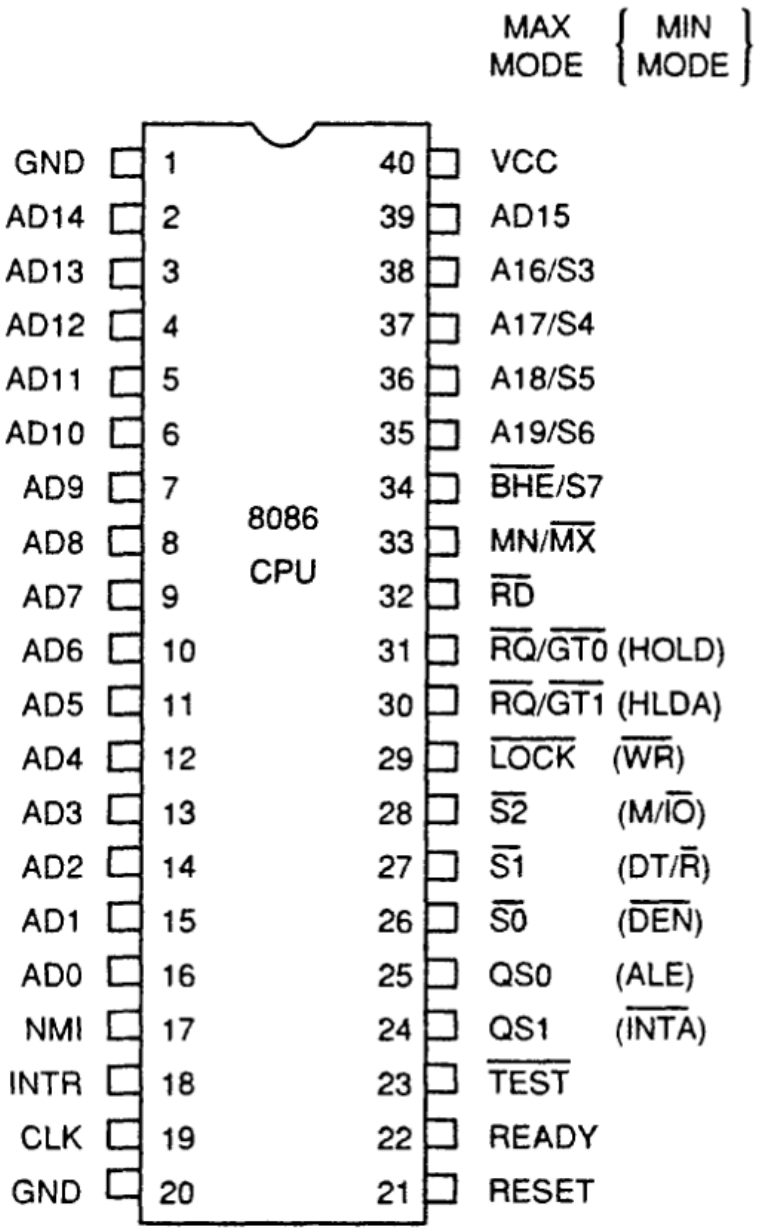
Code Segment (CS): CS register is used to address a memory location in the code segment of the memory where opcode of program is stored.
Data Segment(DS) : DS register points to the data segment of the memory where the data is stored.
Extra Segment (ES): ES register is used to address the segment which is additional data segment
Stack Segment (SS) : SS Register is used to point Stack location in stack segment of the memory and used to store data temporarily on the stack such as the contents of the CPU Registers which will be required later stage of execution. The default segment base and offset pair registers are CS: IP and SS: SP

Flag Register



1. **Carry flag**: This flag is set when there is a carry out of MSB in case of addition or a borrow in case of subtraction.
2. **Parity flag**: This flag is set to 1if the lower byte of the result contains even numbers of 1s
3. **Auxiliary carry flag** : this is set if 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.
4. **Zero flag:** This flag is set if the result of the computation or comparison performed by previous instruction is zero.
5. **Sign flag:** This flag is set when the result of any computation is negative. For signed computations the sign flag equals the MSB of the result.
6. **Trap flag:** If this flag is set the processor enters the single step execution mode.
7. **Interrupt flag:** If this flag is set the maskable interrupts are recognized by the cpu, otherwise they are ignored.
8. **Direction flag:** This flag is used by string manipulation instructions. If this flag bit is „0" the string is processed beginning from the lowest address to the highest address, otherwise the string is processed from the highest address towards lowest address.
9. **Overflow flag:** This flag is set if an overflow occurs.

8086 PIN Diagram:



$\overline{BHE}$ (Active Low)/ **S_7** (Output)**: Bus High Enable/Status**.

When $\overline{BHE}$=0 it indicates that AD15-AD8 are involved in data transfer. $\overline{BHE}$ & A0 are used to select even/odd memory banks or I/O address.

S7 always remain high.

$\overline{BHE}$	**A_0**	**Indication**
0	**0**	Whole word (16-bits)
0	**1**	High byte to/from odd address
1	**0**	Low byte to/from even address
1	**1**	No selection

AD_0-AD_{15} (Bidirectional): Address/Data bus. These pins acts as multiplexed address and data bus of the microprocessor. Whenever the ALE pin is high these pins carry the address, when the ALE pin is low it carry the data.

A_{16}/S_3, A_{17}/S_4, A_{18}/S_5, A_{19}/S_6 : These pins are multiplexed to provide the address signals A19-A16 and the status bits S6-S3. When ALE=1 these pins carry the address and when ALE=0, they carry the status lines.

$\overline{RD}$ **(Read)** (Active Low)**:** The signal is used for read operation with memory or I/O depending on the status of M/$\overline{IO}$ signal.

READY: This is the acknowledgement from the slower I/O device or memory.When high it indicates that peripheral device is ready to transfer data.

INTR-Interrupt Request: This is a level triggered interrupt request input and is checked during the last clock cycles of each instruction to determine the availability of the request.

NMI –NON-MASKABLE INTERRUPT: It is an edge triggered input which causes a non maskable interrupt. The non-maskable interrupt input is a hardware interrupt. It cannot be disabled by software.

$\overline{INTA}$**: Interrupt acknowledge:**

When microprocessor receive INTR signal, processor complete m/c cycle and acknowledge the interrupt by generating this signal.

MN/ $\overline{MX}$ MINIMUM / MAXIMUM: This pin signal indicates what mode the processor is to operate in.

$\overline{RQ}/\overline{GT_0}$ **,** $\overline{RQ}/\overline{GT_1}$ **: REQUEST/GRANT**:

- These pins are used by other local bus master in maximum mode to gain the control of local buses at the end of the processors current bus cycle.
- The pins $\overline{RQ}/\overline{GT_0}$ and $\overline{RQ}/\overline{GT_1}$ are bidirectional and $\overline{RQ}/\overline{GT_0}$ have higher priority than $\overline{RQ}/\overline{GT_1}$**.**
- After receiving request on these lines, CPU sends acknowledge signal on the same lines.

RESET (Input) : RESET: It is system reset. When this signal goes high, the processor enter into reset state.

$\overline{LOCK}$**:**

It indicates that other system bus masters are not to allowed to gain control of the system bus while $\overline{LOCK} = 0$ ($\overline{LOCK}$ is LOW).

When it goes low, all interrupts are masked and HOLD request is not granted.

$\overline{TEST}$:

- This signal is used to test the status of math co-processor 8087
- This input signal is examined by a 'WAIT' instruction.
- If the $\overline{TEST}$ =0, execution will continue WAIT instruction (i.e NOP instruction), If the $\overline{TEST}$ =1, the processor will remain in an idle state.

CLK- Clock Input : The clock input provides the basic timing for processor operation and bus control activity. Its an asymmetric square wave with 33% duty cycle.

Vcc – Power Supply (+5V D.C.)

GND – Ground

QS_1, QS_0 (Queue Status) These signals indicate the status of the internal 8086 instruction queue according to the table shown below:

QS_I	QS_0	Status
0	**0**	No Operation
0	**1**	First Byte of Op Code from Queue
1	**0**	Empty the Queue
1	**1**	Subsequent Byte from Queue

DT/$\overline{R}$: DATA TRANSMIT/RECEIVE:
This output signal is used to decide the direction of data flow through the transceivers (bidirectional buffers) 8286/74LS245
DT/$\overline{R}$ =1 indicates transmitting the data
DT/$\overline{R}$=0 indicates processor receives data.

$\overline{DEN}$: DATA ENABLE

- This signal informs the transceiver that the microprocessor is ready to send or receive the data.
- This signal indicates the availability of valid data over the address/data lines.

M/$\overline{IO}$: (Memory / Input Output)

The signal on this pin indicates a memory or I/O operation.

If M/$\overline{IO}$ =1, then the data transfer operation is between memory and microprocessor and if

M/$\overline{IO}$ =0 then the data transfer operation is between I/O devices and microprocessor.

ALE :(Address Latch Enable):

- It indicate the availability of valid address on AD0-AD15.
- This pin is connected to latch enable pin of latches 8282 or 74LS373.

$\overline{WR}$: **(WRITE)**

This is active low signal issued by processor to write data to memory or I/O depending on the status of M/$\overline{IO}$ signal.

HOLD:
When another master device needs the use of address, data and control bus, it sends a HOLD request to the processor through this line.

HLDA (Hold Acknowledge)
The processor receiving the hold request will issue HLDA as an acknowledgement.

$\overline{S_0}$, $\overline{S_1}$, $\overline{S_2}$: (Status Signals)
These status signals reflect the type of operation being carried out by the processor and required by the bus controller. Intel 8288 to generate all memory / I/O access control signals.

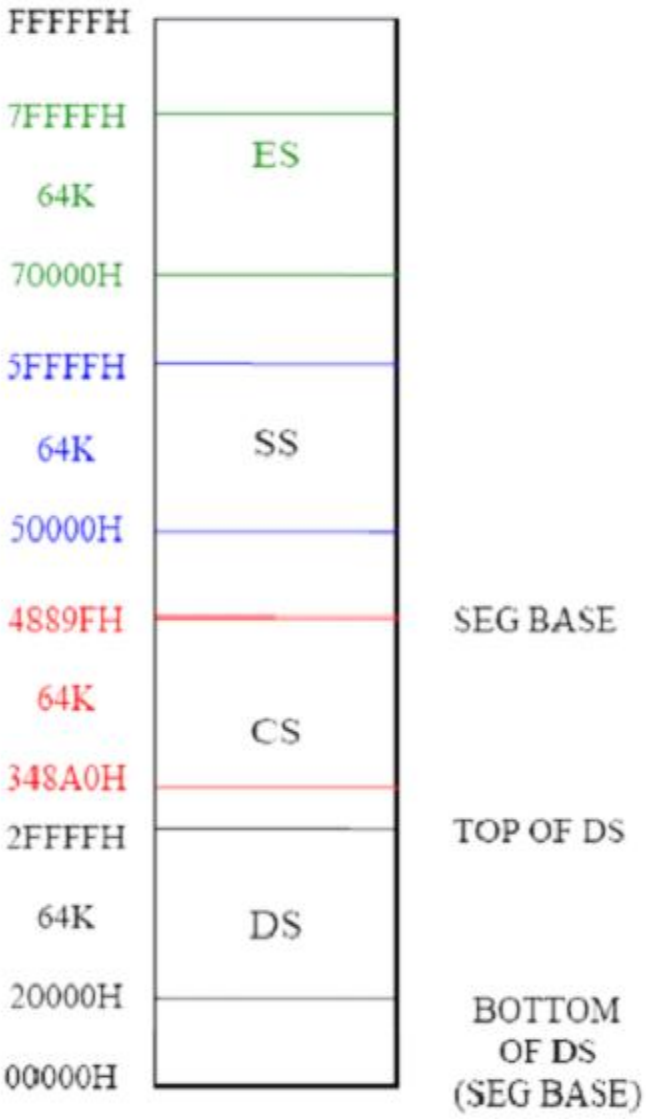
$\overline{S_2}$	$\overline{S_1}$	$\overline{S_0}$	Status
0	0	0	INTA
0	0	1	I/O Read
0	1	0	I/O Write
0	1	1	Halt
1	0	0	Opcode Fetch
1	0	1	Memory Read
1	1	0	Memory Write
1	1	1	Passive

Memory Segmentation in 8086

- The memory in an 8086 based system is organized as segmented memory and this memory management technique is called as segmentation.
- The complete physically memory is divided into a number of logical segments in segmentation
- Size of each segment is 64 KB and addressed by one of the segment register i.e. CS, DS, ES or SS.
- In 8086, 20 bit address bus 2^20 =1 MB Memory
- The 16-bit content of the segment register holds the starting address of a particular segment. So we need an offset address / displacement to address a specific memory location within a segment.
- The offset address is 16-bit so the maximum offset value will be FFFFH and hence the maximum size of any segment is 2^16=64 KB locations Since there are only 16 bit registers hence size=64 KB.
- The complete 1 MB memory can be divided into 16 segments each of 64 KB size.

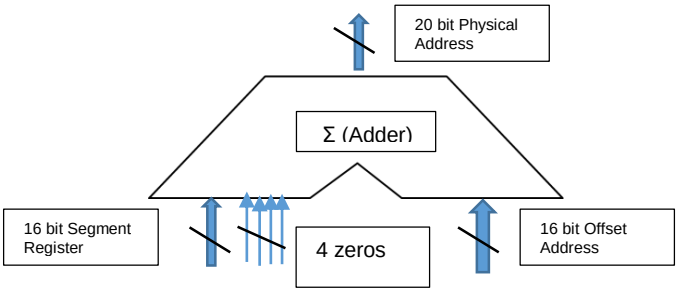
Advantages of segmentation:

- Segmentation can be used in multi-user time shared system
- Programs and data can be stored separately.
- Segmentation allows two processes to share data
- The address associated with any instruction or data is only 16 bits(though 8086 has 20 bits physical address)



Physical Memory address Generation

- The address associated with any instruction or data byte is only 16-bit called as Effective Address /Offset/Displacement/Logical Address.
- The logical addresses are used to calculate physical address
- The address generated by BIU is 20-bit called as Physical address



Calculate the physical address generated by:

i) 4370 : 561E

ii) 7A32 : 0028

4370 0 - Segment Base Address
\+ 561 E - Offset Address

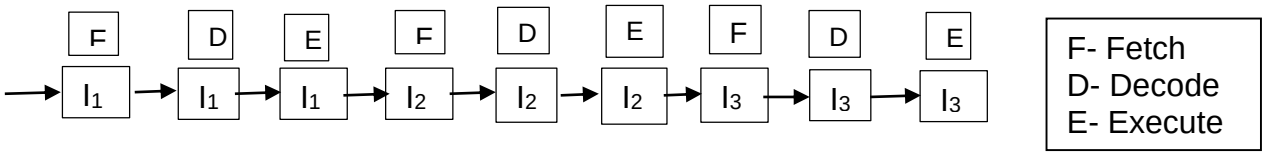
4 8 D1 E – Physical Address

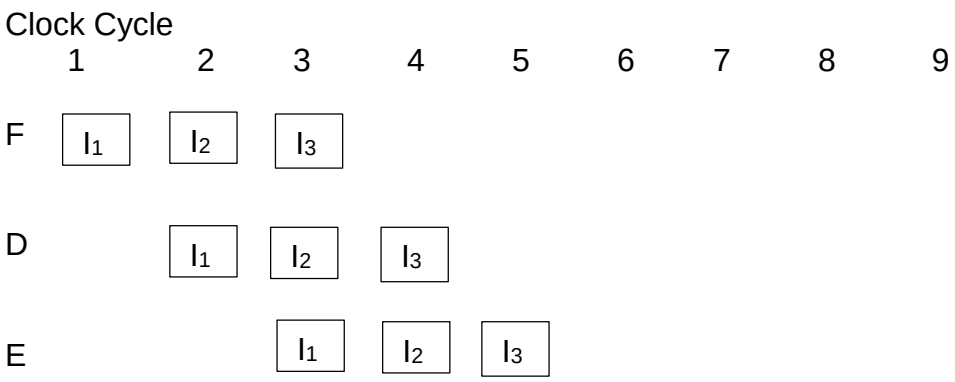
iii) 7A32 0 - Segment Base Address
\+ 002 8 - Offset Address

7A34 8 – Physical Address

Concepts of Pipelining

- The technique used to enable an instruction to complete with each clock cycle is called as Pipelining
- Normally, on a non-pipelined processor, nine clock cycles are required for fetch, decode and execute cycles for the 3 instructions (a)
- But on a pipelined processor, the fetch, decode and execute operations are performed in parallel only five clock cycles are required to execute the same 3 instructions (b)





Pipelined execution of three instructions

- Feature of fetching the next instruction while the current instruction is executing is called Pipelining which will reduce the execution time. So pipelining improve the execution speed of the processor.
- In pipelined processor fetch, decode and execute operation are performed simultaneously or in parallel.
- In 8086, pipelining is implemented by providing 6 byte queue where 6 one byte instructions can be stored well in advance and then one by one instruction goes for decoding and execution.
- In this way 8086 perform fetch, decode and execute operation in parallel.

Advantages of Pipelining:

1) Pipelining enables many instructions to be executed at the same time
2) It allows execution to be done in fewer cycles.
3) Speed up the execution speed of the processor.
4) More efficient use of Processor.

Sr. No	Minimum Mode	Maximum Mode
1	**MN/ $\overline{MX}$ =1** for Minimum Mode	**MN/ $\overline{MX} = 0$** for maximum mode
2	Single Microprocessor in minimum mode	Multiprocessor Configuration
3	Bus Controller IC 8288 is not present	Bus Controller IC 8288 is used to generate control signal.
4	Advanced I/O write signal is not there.	Advanced I/O write command is there.
5	No Status pins $\bar{S}_0$, $\bar{S}_1$, $\bar{S}_2$	Status pins $\bar{S}_0$, $\bar{S}_1$, $\bar{S}_2$ are there.