Module1

CONTENTS

- 8086 Microprocessor
- Memory Segmentation

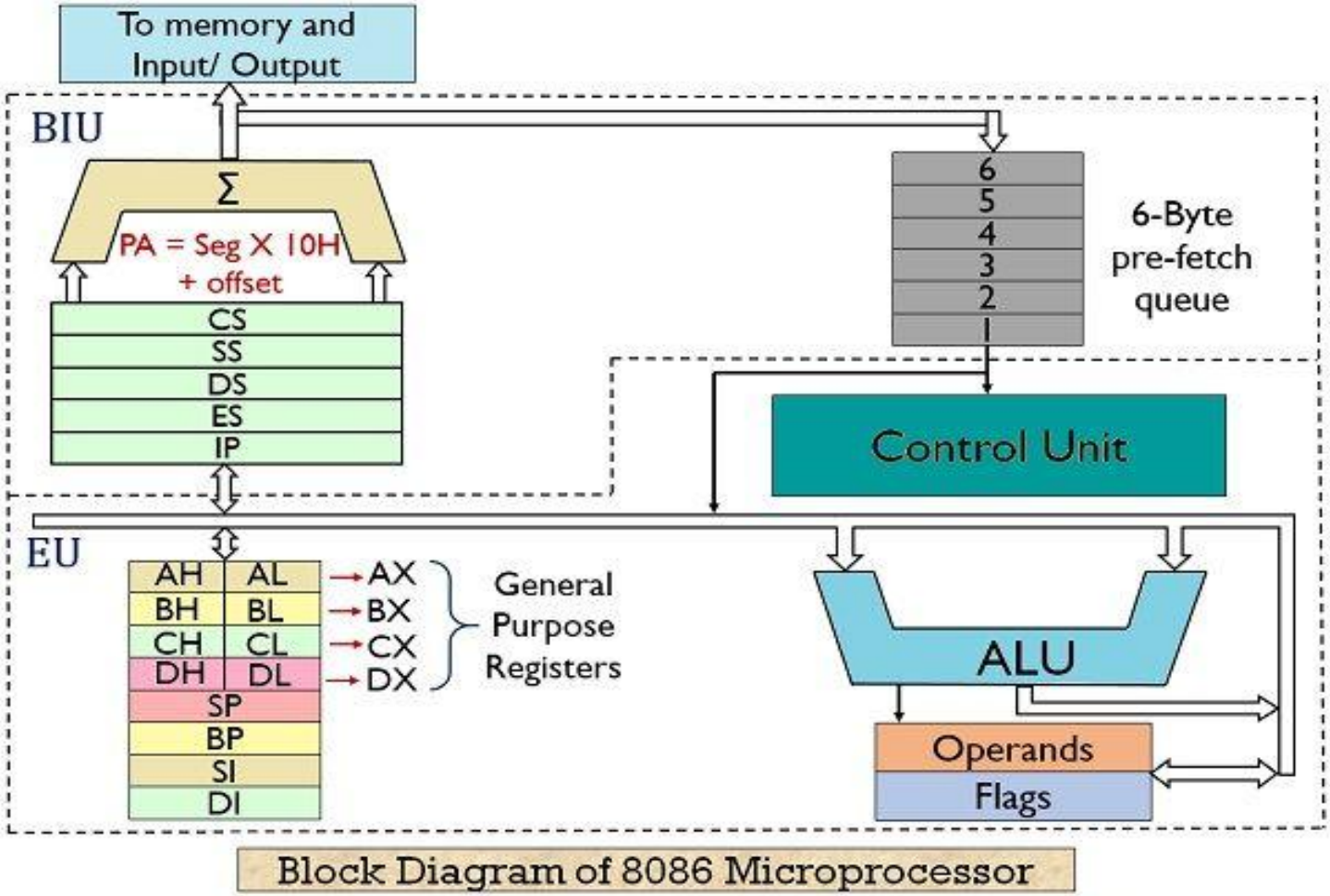
8086 Microprocessor

- 8086 Microprocessor is an enhanced version of 8085
- Designed by Intel in 1978
- It is a 16-bit Microprocessor having 20 address lines and16 data lines that provides up to 1MB storage
- It consists of powerful instruction set, which provides operations like multiplication and division easily.
- It supports two modes of operation, i.e. Maximum mode and Minimum mode.
- Maximum mode is suitable for system having multiple processors and Minimum mode is suitable for system having a single processor.

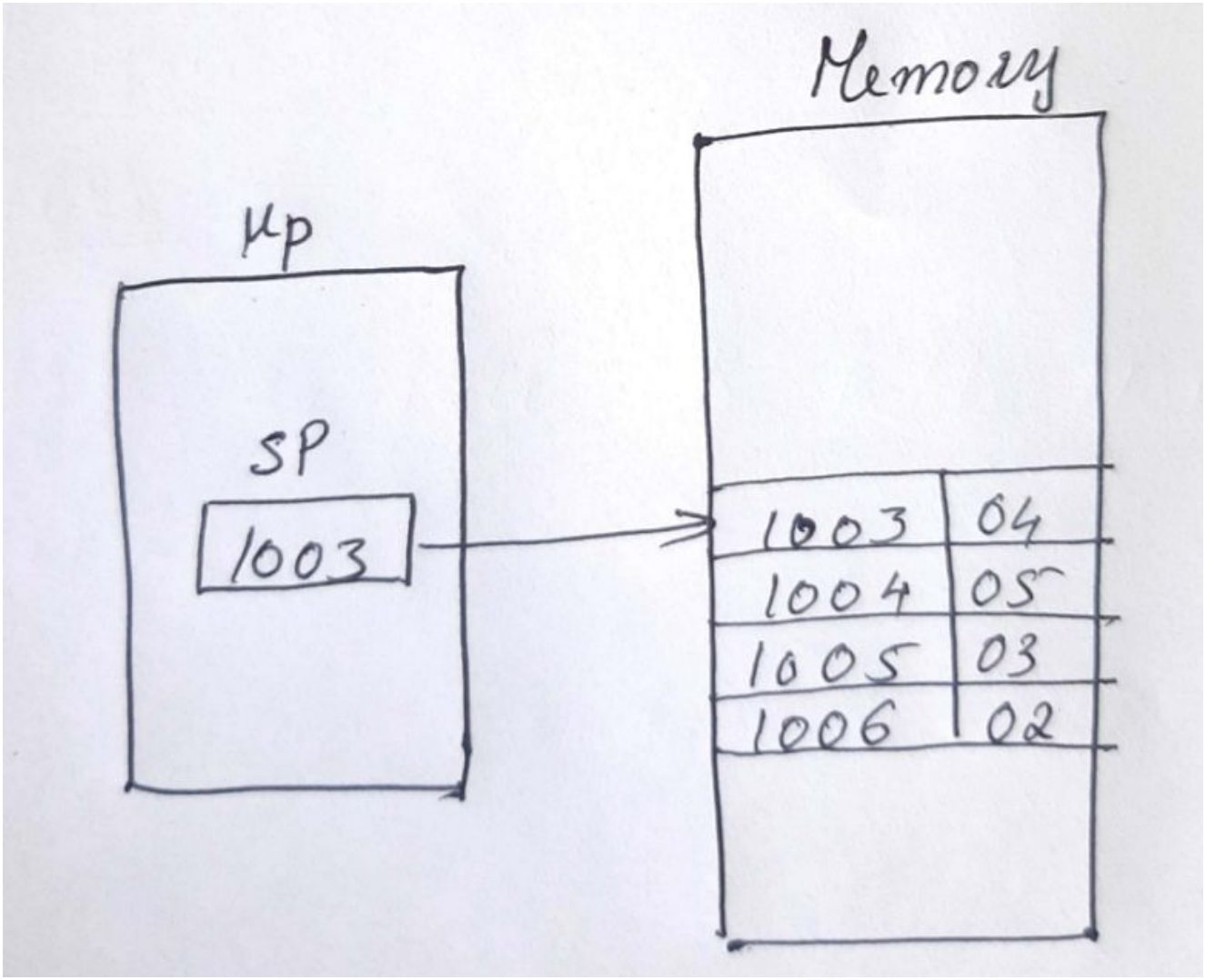
Features of 8086

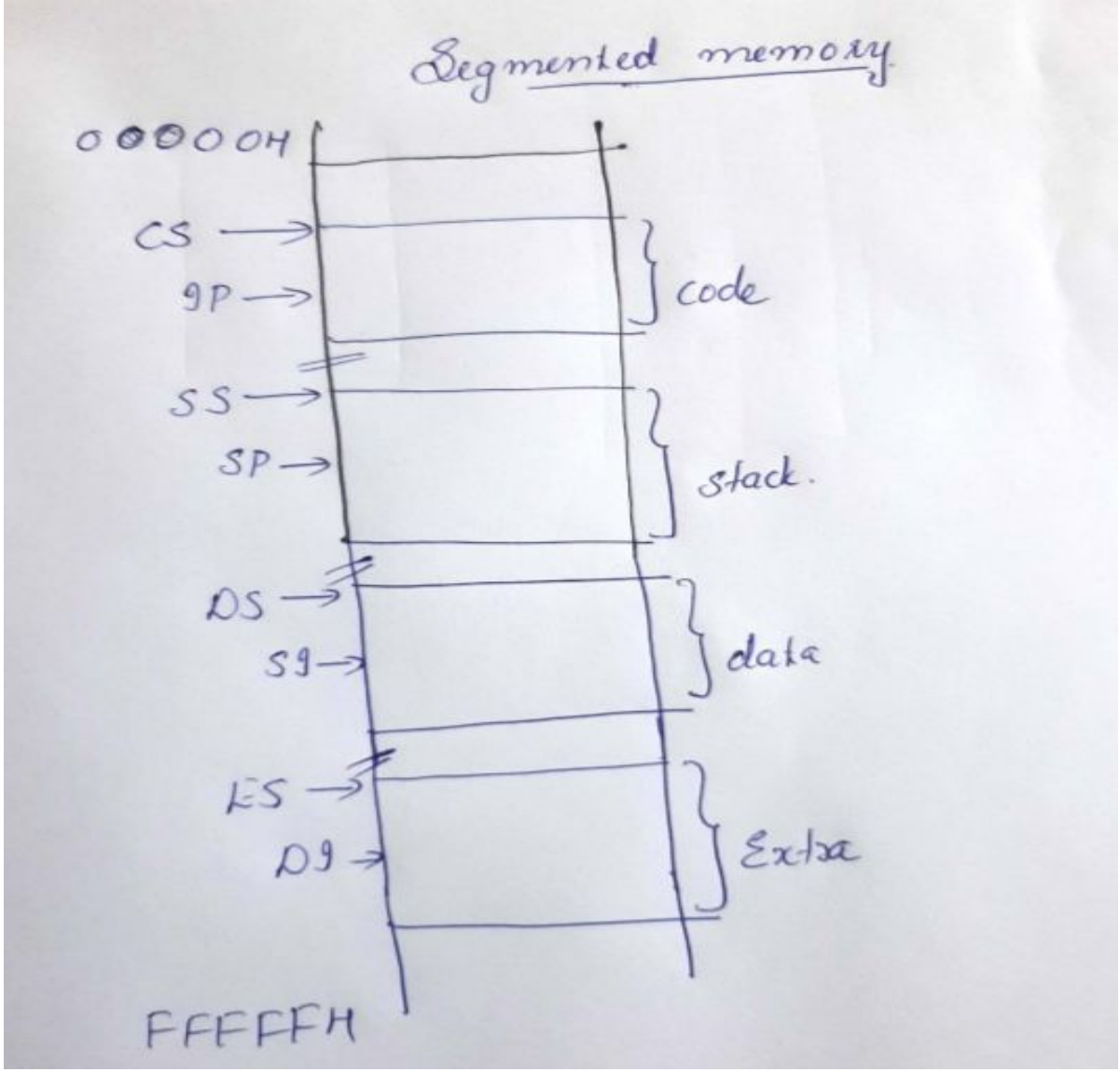
- It has an instruction queue, which is capable of storing six instruction bytes from the memory resulting in faster processing.
- It was the first 16-bit processor having 16-bit ALU, 16-bit registers, internal data bus, and 16-bit external data bus resulting in faster processing.
- It is available in 3 versions based on the frequency of operation −

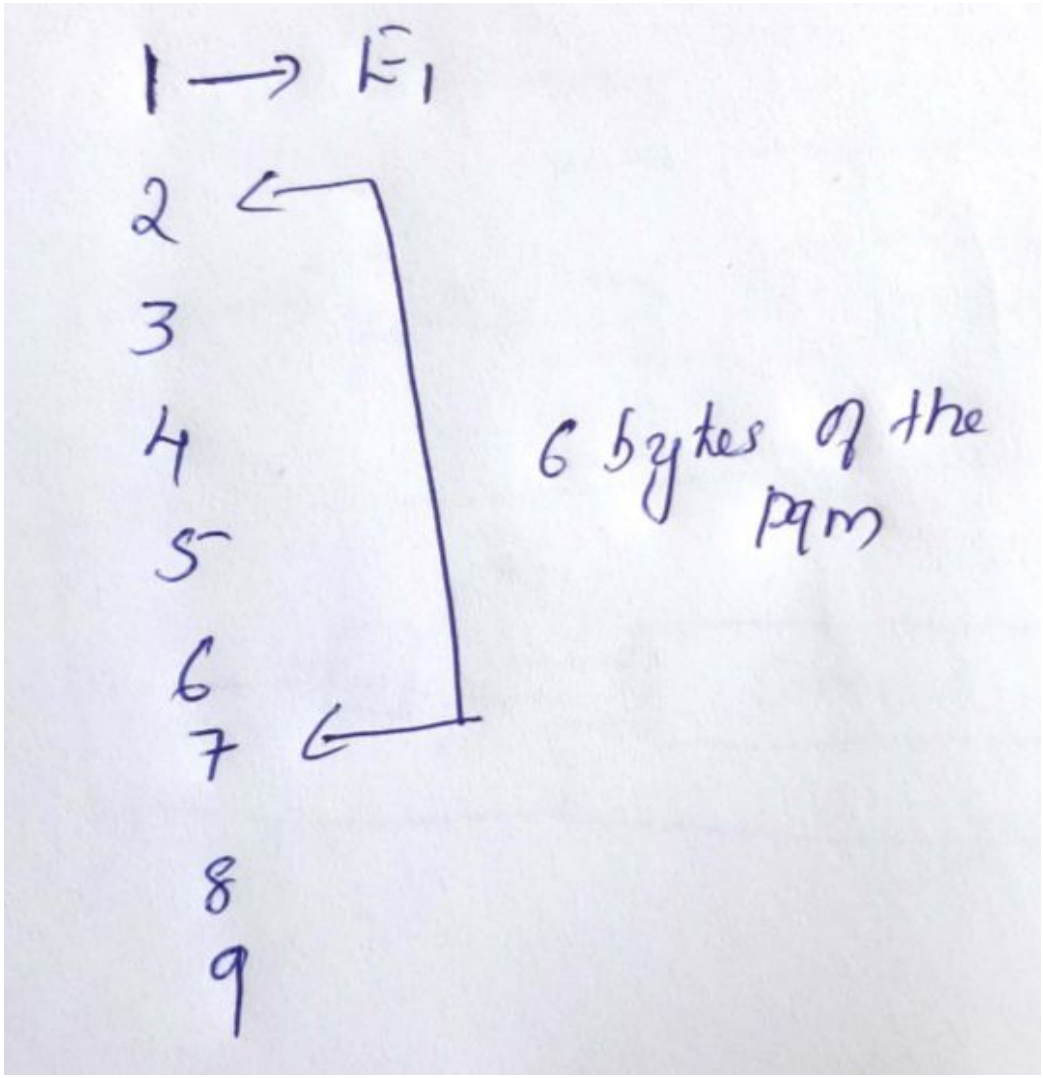
  5MHz,8MHz,10 MHz
- It has 256 vectored interrupts.
- It consists of 29,000 transistors.



Block Diagram of 8086 Microprocessor


Memory
μp
SP
1003
1003
04
1004
05
1005
03
1006
02


Segmented memory
00000H
CS
IP
code
SS
SP
stack.
DS
SI
data
ES
DI
Extra
FFFFFH


1 → E1
2
3
4
5
6
7
8
9
6 bytes of the PAM

MOV BL, CL 10001010

MOV BL, 25H

MOV BX, 2000H.

MOV CL, 34H CL=34.

MOV CH, 12H CH =12

MOV CX, 1234H CH =12, CL=34

Architecture of 8086(explanation)

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

EU (Execution Unit)

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

Execution Unit

ALU

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

Flag Register

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

Conditional Flags

- It represents the result of the last arithmetic or logical instruction executed.

Following is the list of conditional flags −

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

Control Flags

- Control flags controls the operations of the execution unit. Following is the list of control flags −
- Trap flag − It is used for single step control and allows the user to execute one instruction at a time for debugging. If it is set, then the program can be run in a single step mode.
- Interrupt flag − It is an interrupt enable/disable flag, i.e. used to allow/prohibit the interruption of a program. It is set to 1 for interrupt enabled condition and set to 0 for interrupt disabled condition.
- Direction flag − It is used in string operation. As the name suggests when it is set then string bytes are accessed from the higher memory address to the lower memory address and vice-a-versa.

Flag Register

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.

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.

Sign Flag

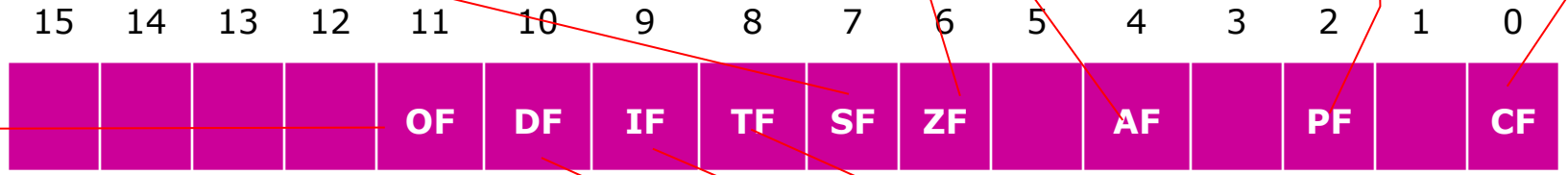
This flag is set, when the result of any computation is negative

Zero Flag

This flag is set, if the result of the computation or comparison performed by an instruction is zero

Parity Flag

This flag is set to 1, if the lower byte of the result contains even number of 1's ; for odd number of 1's set to zero.



Over flow Flag

This flag is set, if an overflow occurs, i.e, if the result of a signed operation is large enough to accommodate in a destination register. The result is of more than 7-bits in size in case of 8-bit signed operation and more than 15-bits in size in case of 16-bit sign operations, then the overflow will be set.

Trap Flag

If this flag is set, the processor enters the single step execution mode by generating internal interrupts after the execution of each instruction

Direction Flag

This 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, i.e., auto incrementing mode. Otherwise, the string is processed from the highest address towards the lowest address, i.e., auto incrementing mode.

Interrupt Flag

Causes the 8086 to recognize external mask interrupts; clearing IF disables these interrupts.

General purpose register

- There are 8 general purpose registers, i.e., AH, AL, BH, BL, CH, CL, DH, and DL. These registers can be used individually to store 8-bit data and can be used in pairs to store 16bit data. The valid register pairs are AH and AL, BH and BL, CH and CL, and DH and DL. It is referred to the AX, BX, CX, and DX respectively.
- AX register − It is also known as accumulator register. It is used to store operands for arithmetic operations.
- BX register − It is used as a base register. It is used to store the starting base address of the memory area within the data segment.
- CX register − It is referred to as counter. It is used in loop instruction to store the loop counter.
- DX register − This register is used to hold I/O port address for I/O instruction.

Stack pointer register

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

BIU (Bus Interface Unit)

- BIU takes care of all data and addresses transfers on the buses for the EU like sending addresses, fetching instructions from the memory, reading data from the ports and the memory as well as writing data to the ports and the memory. EU has no direct connection with System Buses so this is possible with the BIU. EU and BIU are connected with the Internal Bus

Bus Interface unit

BIU has the following functional parts

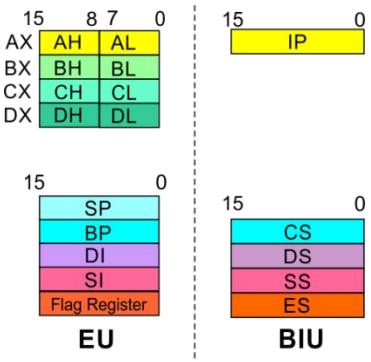
- **Instruction queue** − BIU gets upto 6 bytes of next instructions and stores them in the instruction queue. When EU executes instructions and is ready for its next instruction, then it simply reads the instruction from this instruction queue resulting in increased execution speed.
- Fetching the next instruction while the current instruction executes is called pipelining.

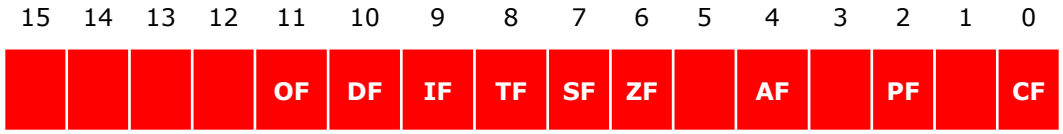
- Segment register − BIU has 4 segment registers i.e. CS, DS, SS& ES.
- CS − It stands for Code Segment. It is used for addressing a memory location in the code segment of the memory, where the executable program is stored.
- DS − It stands for Data Segment. It consists of data used by the program and is accessed in the data segment by an offset address or the content of other register that holds the offset address.
- SS − It stands for Stack Segment. It handles memory to store data and addresses during execution.

- ES − It stands for Extra Segment. ES is additional data segment, which is used by the string to hold the extra destination data.
- Instruction pointer(IP) − It is a 16-bit register used to hold the address of the next instruction to be executed.

Register Organization

8086 registers categorized into 4 groups





Sl.No.	Type	Register width	Name of register
1	General purpose register	16 bit	AX, BX, CX, DX
		8 bit	AL, AH, BL, BH, CL, CH, DL, DH
2	Pointer register	16 bit	SP, BP
3	Index register	16 bit	SI, DI
4	Instruction Pointer	16 bit	IP
5	Segment register	16 bit	CS, DS, SS, ES
6	Flag (PSW)	16 bit	Flag register

Registers and Special Functions

Register	Name of the Register	Special Function
AX	16-bit Accumulator	Stores the 16-bit results of arithmetic and logic operations
AL	8-bit Accumulator	Stores the 8-bit results of arithmetic and logic operations
BX	Base register	Used to hold base value in base addressing mode to access memory data
CX	Count Register	Used to hold the count value in SHIFT, ROTATE and LOOP instructions
DX	Data Register	Used to hold data for multiplication and division operations
SP	Stack Pointer	Used to hold the offset address of top stack memory
BP	Base Pointer	Used to hold the base value in base addressing using SS register to access data from stack memory
SI	Source Index	Used to hold index value of source operand (data) for string instructions
DI	Data Index	Used to hold the index value of destination operand (data) for string operations

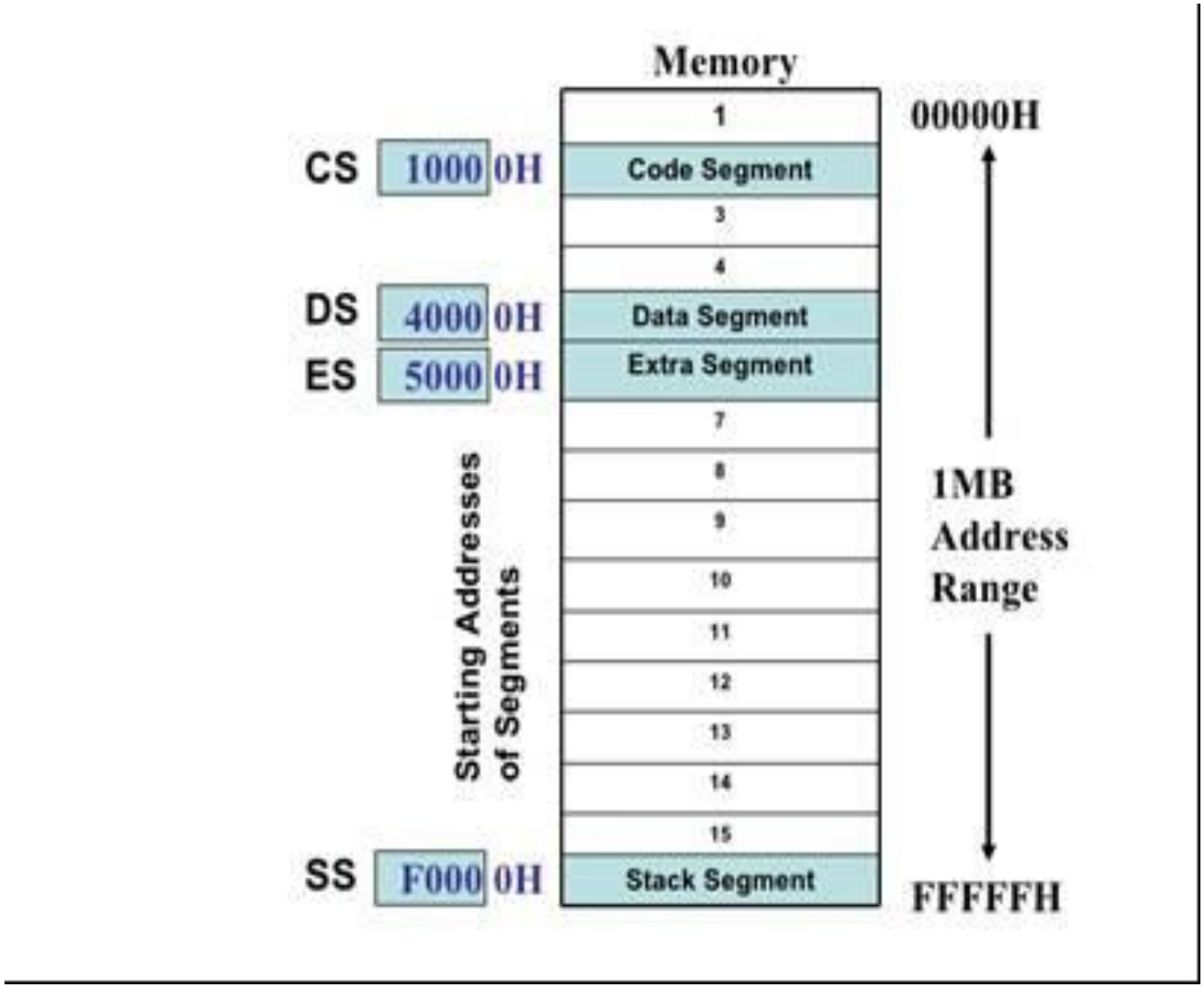
Memory Segmentation

- Main memory is divided into different segments and each segment has its own base address.
- Increase the execution speed of computer system so that processor can able to fetch and execute the data from memory easy and fastly

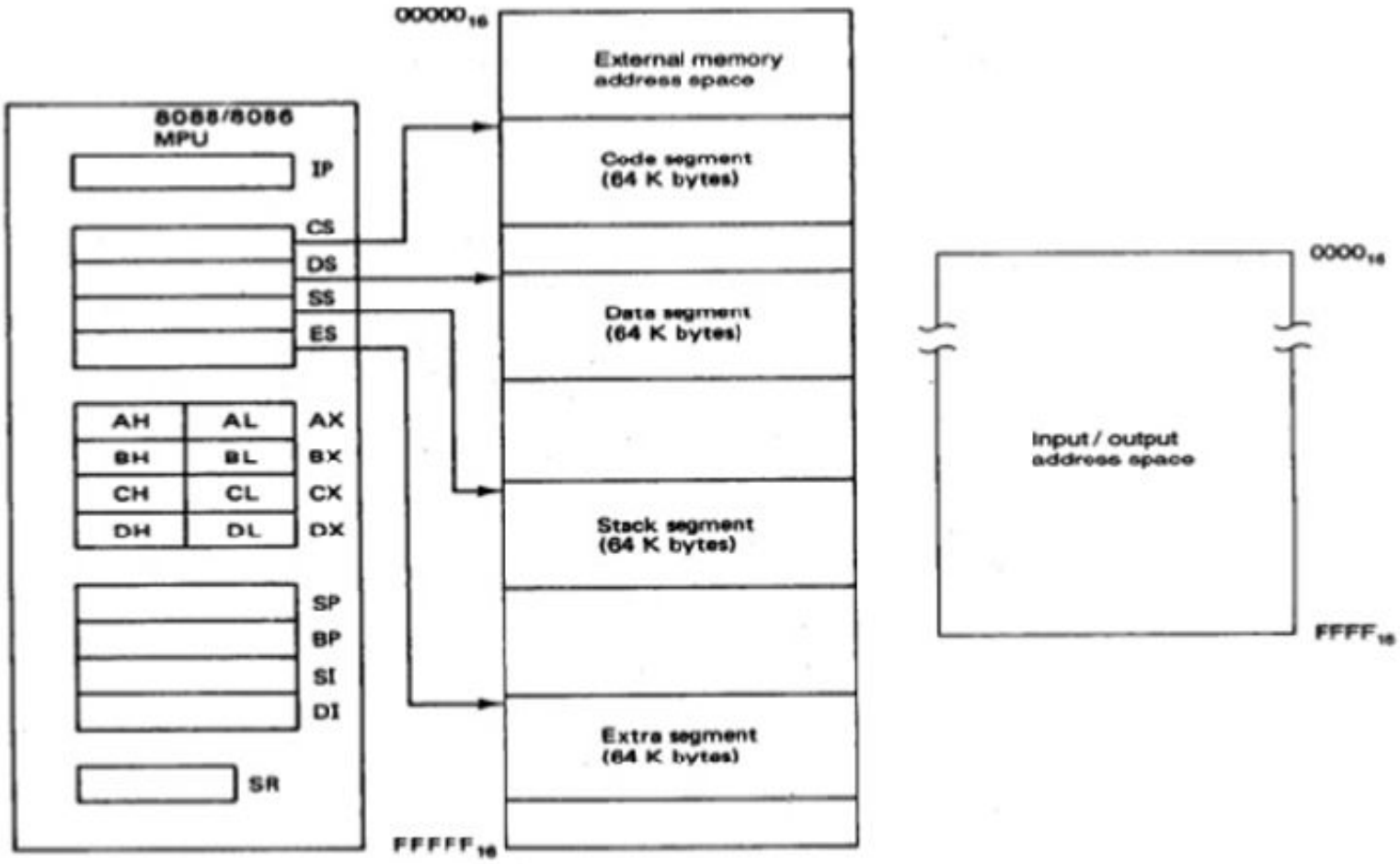
Segmentation in 8086

- The size of address bus of 8086 is 20 and is able to address 1 Mbytes of physical memory.
- The compete 1 Mbytes memory can be divided into 16 segments, each of 64 Kbytes size.
- Only four segments can be addressed at a time

Segmentation in 8086



Segmented Memory Representation



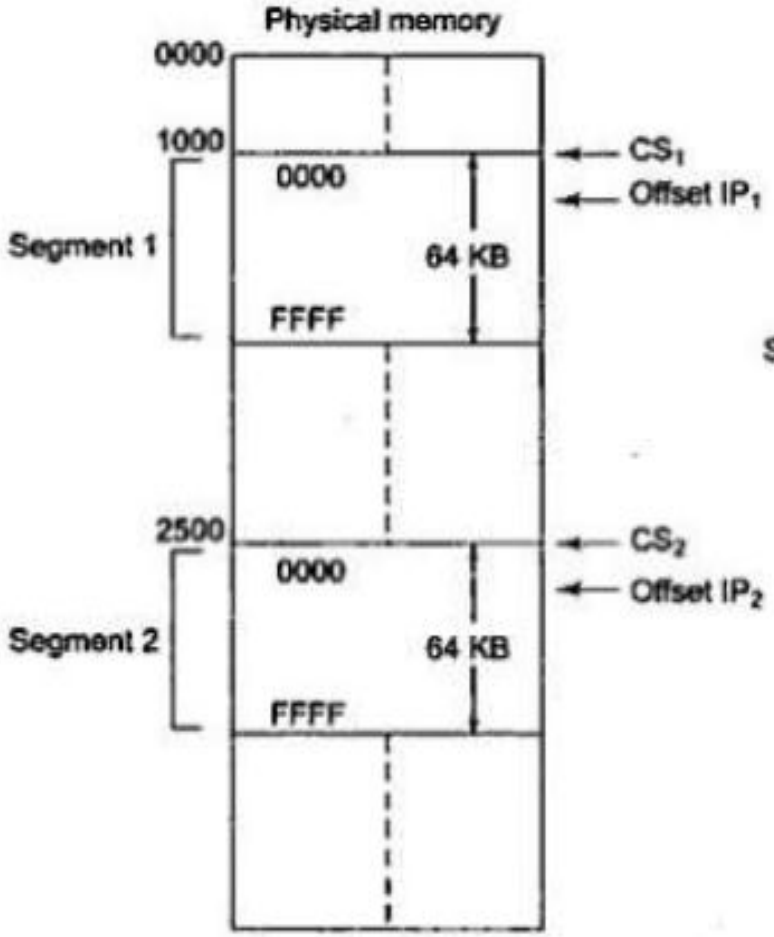
Types of Segmentation

Overlapping segment

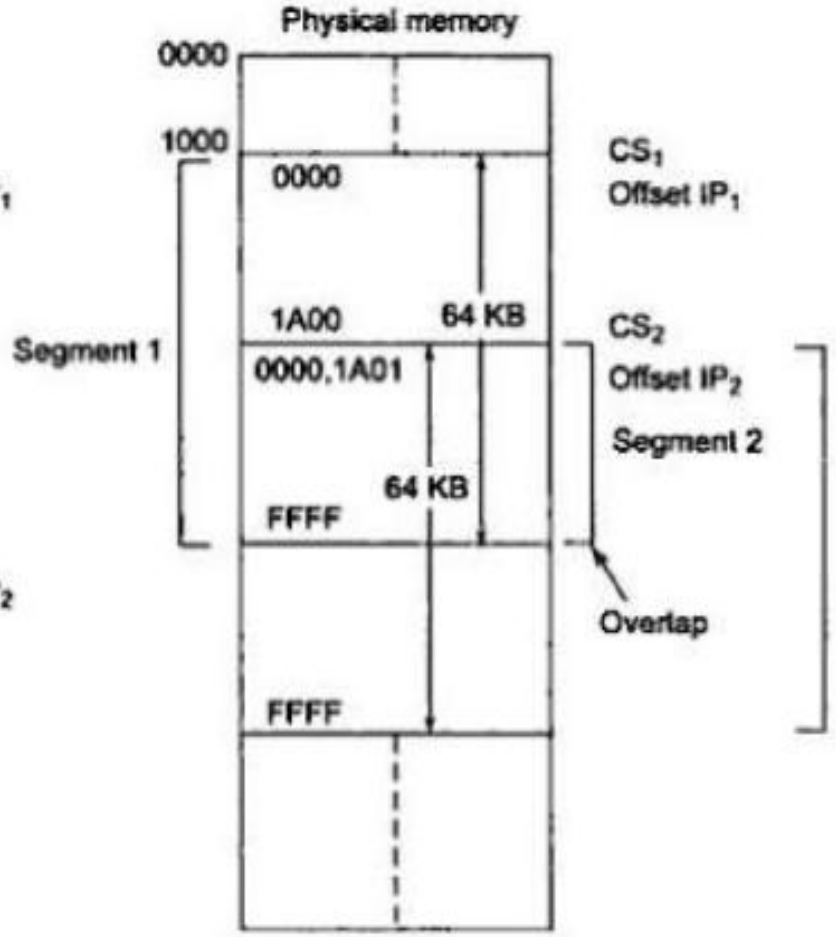
- A segment starts at a particular address and its maximum size can go up to 64 Kbytes. But if another segment starts along this 64 Kbytes location of the first segment, the two segments are said to be overlapping segment.
- The area of memory from the start of the second segment to the possible end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

Non Overlapped Segment

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



Non-overlapping Segments



Overlapping Segments

Physical Address Calculation in 8086

- Segment address ----- 1005H
- Offset address -------5555H
- Segment address-----1005------ 0001 0000 0000 0101
- Shifted by 4 bit positions---- 0001 0000 0000 0101 0000

  +
- Offset address ---------- 0101 0101 0101 0101

- Physical address ---------- 0001 0101 0101 1010 0101

  1 5 5 A 5

Question

1) The value of Code Segment (CS) Register is 4042H and the value of offset is IP: 0580H.Calculate the effective address of the memory location pointed by the CS register.

Ans: 409A0