

# Microprocessors & Microcontrollers

Second Year  
Electrical Engineering Department  
College of Engineering  
Basrah University  
2024-2025

**1: Introduction to Microprocessors**

## **COURSE OBJECTIVES:**

- To understand the basics of microprocessors and microcontrollers architectures and its functionalities.
- To develop an in-depth understanding of the operation of microprocessors and microcontrollers, machine language programming & interfacing techniques.
- To design and develop Microprocessor/ Microcontroller based systems for real time applications using low level language.
- To understand the concepts of ARM processor.

# Syllabus

- 1) Introduction To Microprocessor.
- 2) Microprocessor 8086
- 3) Instruction Set and Assembly Language Programming Of 8086
- 4) Stack and Interrupts of 8086
- 5) Introduction To Microcontrollers
- 6) ARM Processor

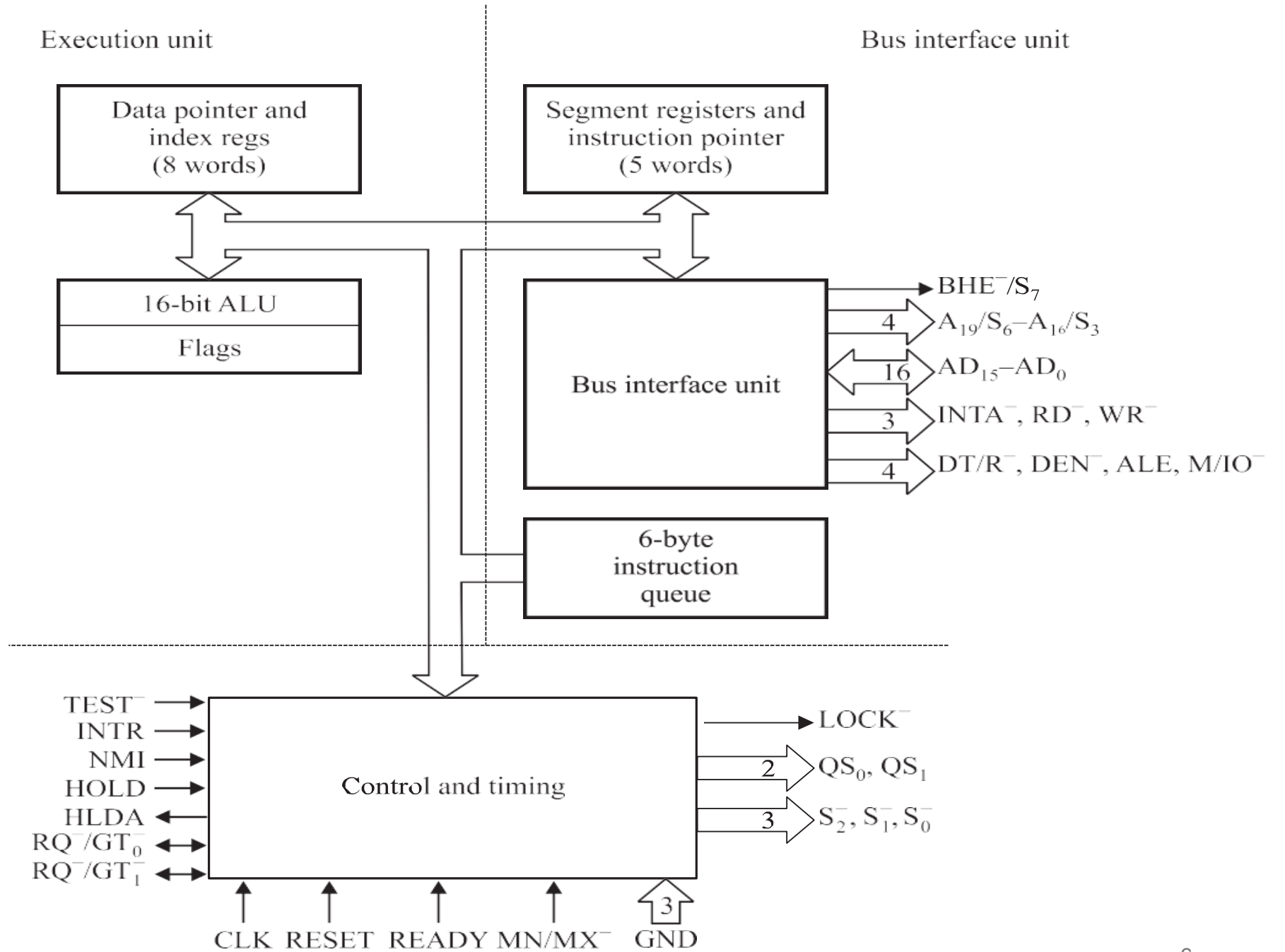
# Introduction to Microprocessors

- The word *microprocessor* was introduced by Intel Corporation. The microprocessor is a single chip microcomputer, i.e. all the functional components of a computer are inbuilt on a single chip using VLSI technology. It consists of an arithmetic and logic unit, register unit and control unit. These three units are fabricated on a single chip.
- The microprocessors are generally characterized by speed, word length, architecture and instruction set.
- 8-bit, 16-bit and even 32-bit microprocessors are quite matured today. Several independent microprocessor families are available in the market. These microprocessors have been widely used in the design of new electronic equipment's and computers.

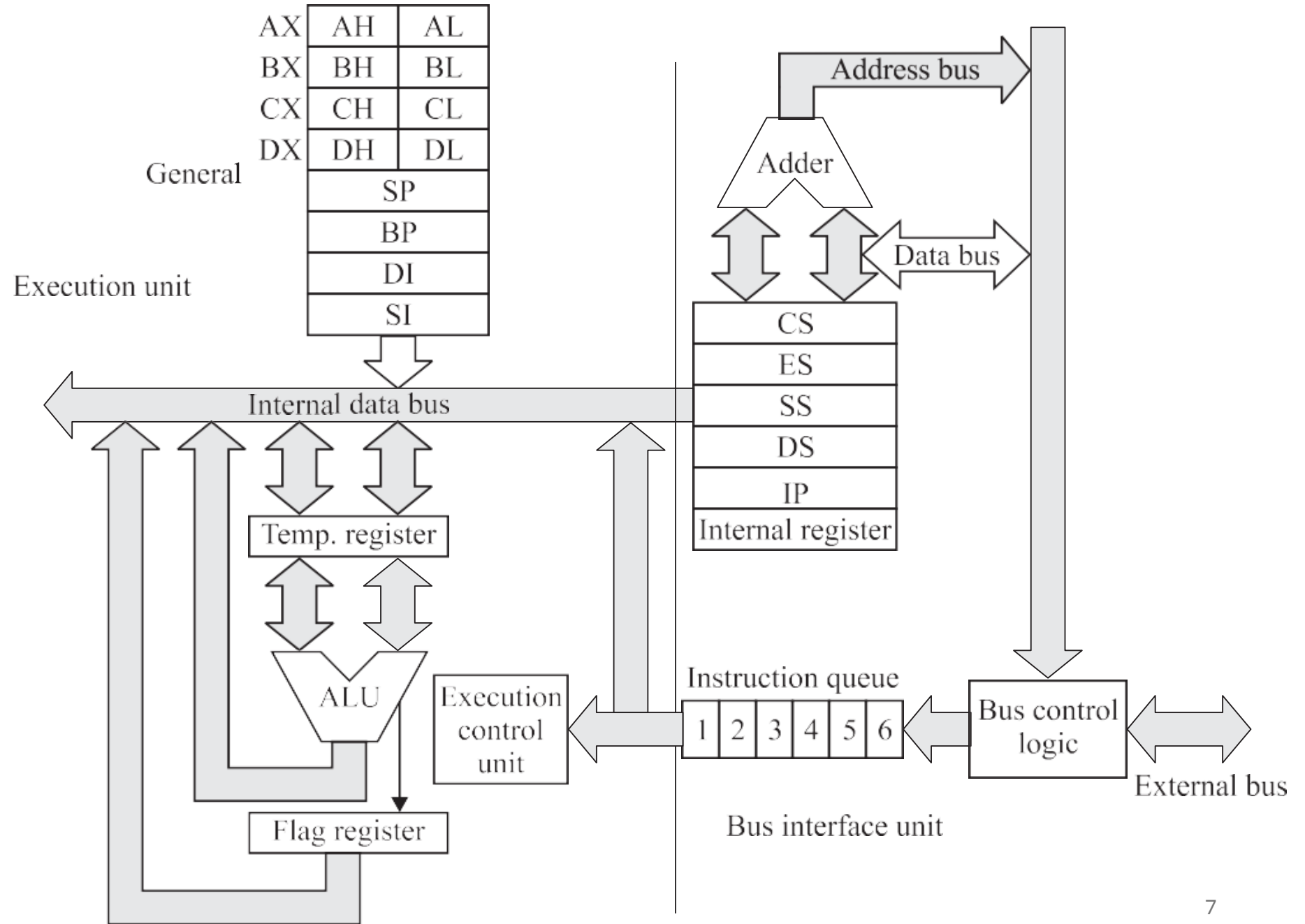
# Microprocessor 8086

- The 8086 was the first 80X86 families and is the basis for all Intel microprocessors that followed. It was a 16-bit microprocessor. It has 20 address lines and capable of addressing 1 Mbyte memory space. The various versions of the 8086 are operated on 5, 8, and 10 MHz clock frequencies.
- The block diagram of 8086 consists of two main sections, the bus interface unit (BIU) and the execution unit (EU). These two units are independent of each other and behave as separate asynchronous operational processors. The EU contains the ALU, flags and general purpose registers.
- The EU carries out all the arithmetic and logical operations. All the registers in the 8086 are 16-bits wide, although 16-bit data registers can be used as two 8-bit data registers.
- The BIU controls the address, data and control buses. The instruction fetching and queuing, operand fetch and store, and address relocation are the operations performed by the bus interface unit.
- When the EU is decoding an instruction or executing instructions inside the microprocessor, the BIU prefetches instructions from memory and stores them in the instruction queue for faster processing.

Block diagram of 8086



Block diagram of 8086  
with more details



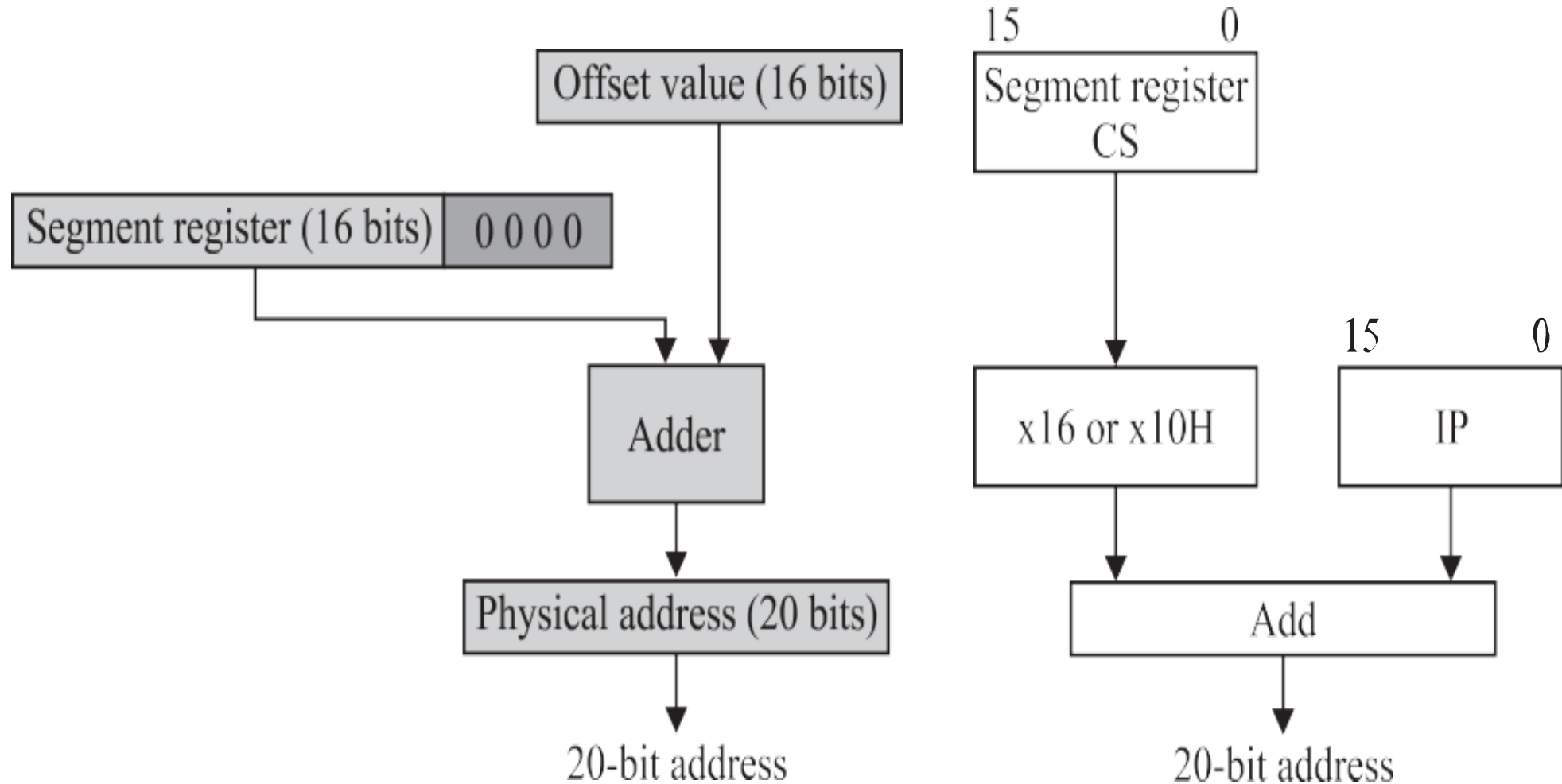
# Bus Interface Unit (BIU)

The Bus Interface Unit (BIU) consists of the following:

- **Queue:** Microprocessor 8086 consists of a FIFO (first in first out) registers set arranged like a pipe and called queue. The BIU continuously fetch operations from the memory while the processor is executing the current instruction. BIU unit stores the fetched bytes in the queue and the EU will read these bytes from the queue as and when it requires.
- **Segment registers:** The memory of 8086 is of 1 MB which is divided into segments or we can say that the memory of 8086 is segmentized. Microprocessor 8086 can at a time access four segments. These segments are named Code Segment (CS), Stack Segment (SS), Data Segment (DS) and Extra Segment (ES). Each segment is up to 64 K bytes long. Each segment is independent and separately addressable unit. Each segment is assigned a base address, which is its starting location in the memory space. All segments start on 16-bit memory boundaries.
- **Instruction pointer (IP):** The IP contains the offset or logical address of the next byte to be read from the code segment. In fact, it shows the distance of the current location, in bytes, from the base address given by the current Code Segment (CS) register. The following Figure shows how this is done. The contents of the CS are shifted left by four. Bit 15 moves to the bit 19 position. The lowest four bits are filled with zeros or the CS register value is multiply by decimal 16 or hexa decimal 10 H.



- The resulting value is added to the Instruction Pointer contents to make up a 20-bit physical address. The CS makes up a segment base address and the IP is looked as an offset into this segment. This segmented model also applies to all the other general registers and segment registers in the 8086 device. For example, the SS and SP are combined in the same way to address the stack area in physical memory.



## Execution Unit (EU)

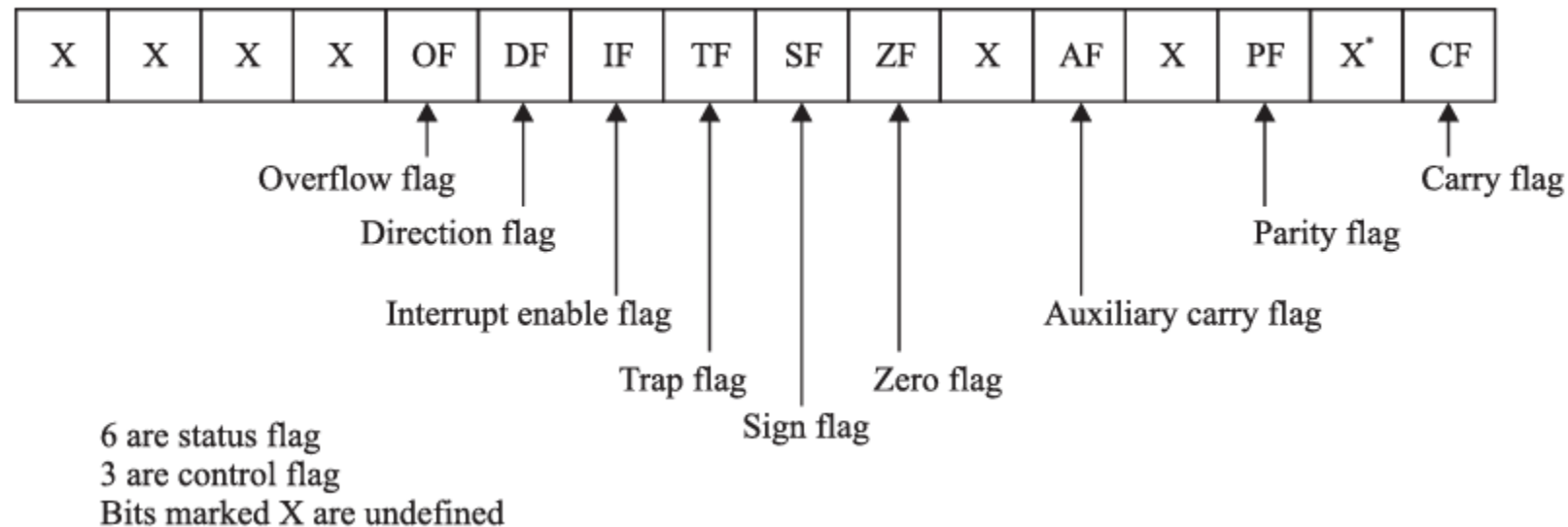
- The execution unit consists of four 16-bit general purpose data registers which can be used as eight 8-bit data registers, four 16-bit pointers and base registers and one 16-bit flag register.
- **General purpose data registers**
- Microprocessor 8086 consists of four 16-bit data registers AX, BX, CX and DX. Each of these registers can be divided into two parts as higher and lower part to store 8-bit data. These are shown in following Table.

AX	AH	AL
BX	BH	BL
CX	CH	CL
DX	DH	DL

- AX register is used as a 16-bit accumulator in 16-bit operations whereas AH and AL are used as accumulator in 8-bit operations. These accumulator registers are used in division, multiplication, and shift and rotate instructions. All the IO and most of the string operations involve accumulator as one of the two operands.

- The 16-bit base register (BX) is supposed to be made up of two 8-bit registers BL and BH. Apart from one 16-bit data register, BX register can be used as a memory pointer in data segment. It can be used for based, based indexed or register indirect addressing modes.
- Count register (CX) or CH and CL can be used as a counter in string manipulation and shift/rotate instructions. These registers are the default counter in loop instruction.
- Data register (DX) is used in division instructions to hold higher word of the operand and the remainder after division. It is used in multiplication operation to hold the higher word of the result. When the IO address is of 16-bit than DX register is, by default, used to hold the 16-bit IO address.
- **Pointers and base registers:** Microprocessor 8086 consists of four 16-bit pointers and base registers. These are SI, DI, SP and BP. These registers are used to hold offset or the logical addresses within a segment.
- Stack Pointer (SP) is a 16-bit register pointing to program stack.
- Base Pointer (BP) is a 16-bit register pointing to data in stack segment. BP register is usually used for based, based indexed or register indirect addressing.
- Source Index (SI) is a 16-bit register. SI is used for indexed, based indexed and register indirect addressing, as well as a source data address in string manipulation instructions.
- Destination Index (DI) is a 16-bit register. DI is used for indexed, based indexed and register indirect addressing, as well as a destination data address in string manipulation instructions.

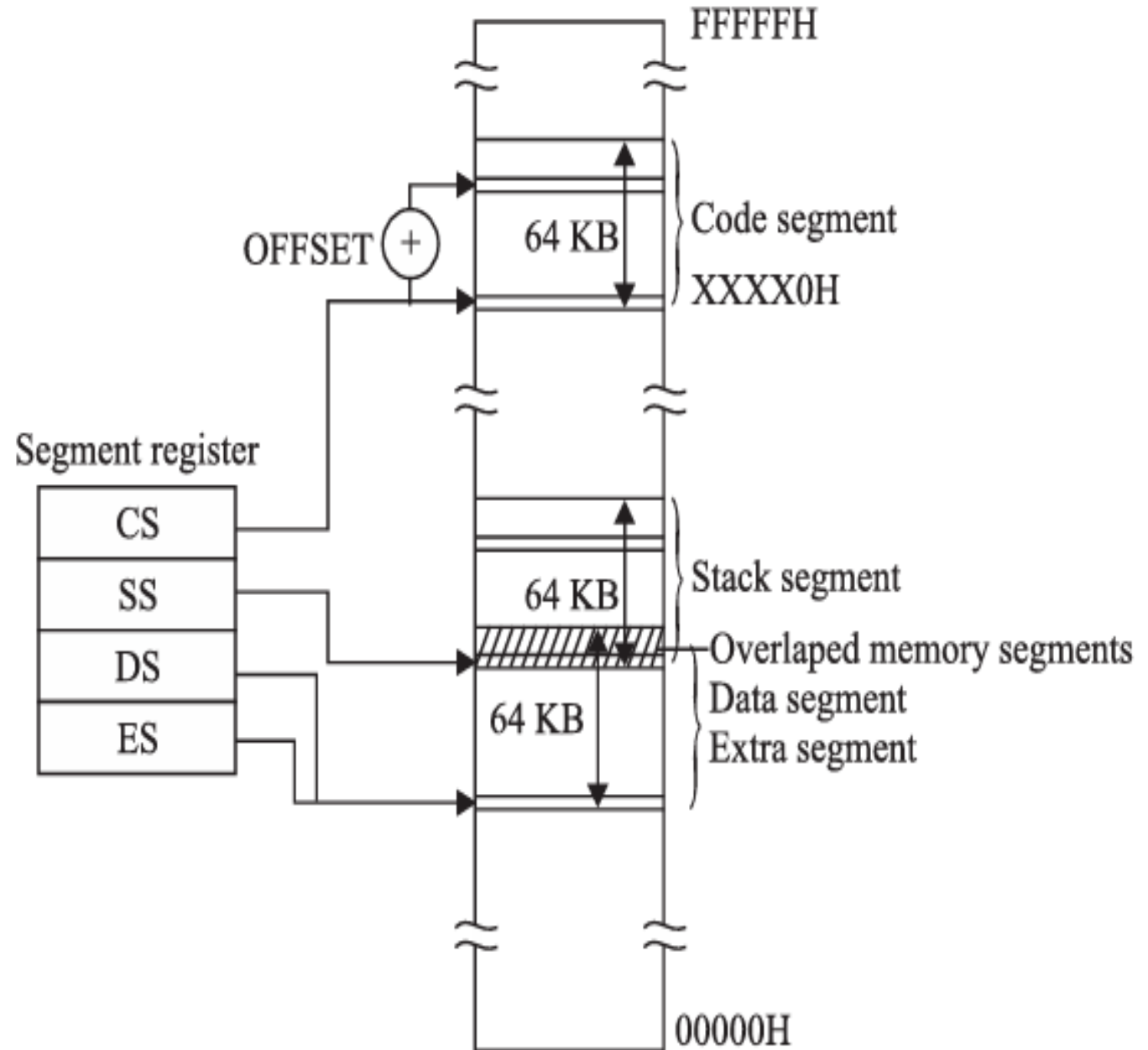
- *Flag register*
- Flags Register determines the current state of the processor. They are modified automatically by CPU after mathematical operations, this allows to determine the type of the result, and to determine conditions to transfer control to other parts of the program.
- A flag is a flip flop that indicates some condition produced by execution of an instruction or controls certain operation of the EU. 6 flip-flops are used to indicate some data status ( sign, zero, auxiliary carry, parity and carry flags) and 3 flip-flops are used to control (direction, interrupt enable and trap flags).



- **Arithmetic Logic Unit:**
- 16 bit ALU
- Used to carry the operations
  - ADD
  - SUBTRACT
  - XOR
  - INCREMENT
  - DECREMENT
  - COMPLEMENT
  - SHIFT BINARY NUMBERS

# MEMORY ORGANIZATION OF 8086

- The 8086 microprocessor provides a 20-bit address to memory. The memory is organized as a linear array of up to 1 MB, addressed from 00000H to FFFFFH.
- The memory is logically divided into code, data, extra and stack segments each of 64 KB. These four segments can partially or fully overlap with each other.
- All memory references are made relative to base addresses contained in segment registers. The segment types were chosen based on the addressing needs of programs.



- That means the code segment always contains the programs or the instructions whereas the stack segment is used to store the data or the addresses during the execution of the PUSH, POP, CALL or the interrupt operations. The data segment and the extra segments are used to store the data.
- Certain locations in memory are reserved for specific microprocessor operations. Table 2.6 shows these reserved locations.

Type of memory references	Default segment base	Alternate segment base	Offset
Instruction fetch	CS	None	IP
Stack operation	SS	None	SP
Variable (except following)	DS	CS, ES, SS	Effective address
String source	DS	CS, ES, SS	SI
String destination	ES	None	DI
BP used as base register	SS	CS, DS, ES	Effective address

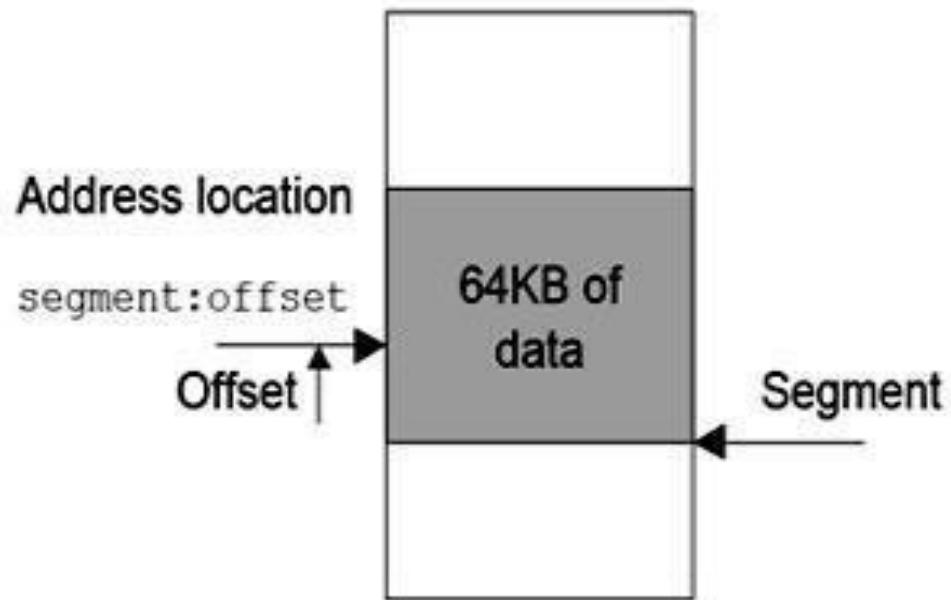
- From software architecture of the 8086 microprocessor, we see that it includes fourteen 16-bit internal registers: the instruction pointer (IP), four data registers (AX, BX, CX, and DX), two pointer registers (BP and SP), two index registers (SI and DI), four segment registers (CS, DS, SS, and ES) and status register (SR), with nine of its bits implemented as status and control flags.
- The point to note is that the beginning segment address must begin at an address divisible by 16. Also note that the four segments need not be defined separately. It is allowable for all four segments to completely overlap (CS = DS = ES = SS).



## Logical and Physical Address

- Addresses within a segment can range from address 00000h to address 0FFFFh. This corresponds to the 64K-byte length of the segment. An address within a segment is called an offset or logical address.
- A logical address gives the displacement from the base address of the segment to the desired location within it, as opposed to its "real" address, which maps directly anywhere into the 1 MByte memory space. This "real" address is called the physical address.
- What is the difference between the physical and the logical address? The physical address is 20 bits long and corresponds to the actual binary code output by the BIU on the address bus lines. The logical address is an offset from location 0 of a given segment.
- You should also be careful when writing addresses on paper to do so clearly. To specify the logical address XXXX in the stack segment, use the convention SS:XXXX, which is equal to  $[SS] * 16 + XXXX$ .

- Logical address is in the form of: Base Address: Offset is the displacement of the memory location from the starting location of the segment. To calculate the physical address of the memory, BIU uses the following formula:
- **Physical Address = Base Address of Segment \* 16 + Offset**



# PIN CONFIGURATION OF 8086

H. W. : Write a report about this subject ?

