1. Signed and unsigned overflow-

Signed Overflow:

On addition of numbers with the same sign, signed overflow occurs when the sum has a different sign. In the preceding example, when we were adding 7FFFH and 7FFFH (two positive numbers), but we get FFFEH (a negative number). Subtraction of numbers with different signs is like adding numbers of the same sign. For example, A – (– B) = A + B and – A – (+B) = – A + (– B). Signed overflow occurs if the result has a different sign than expected. In addition of numbers with different signs, overflow is impossible, because a sum like A + (– B) is really A – B. For exactly the same reason, subtraction of numbers with the same sign cannot give overflow.

Unsigned Overflow :

On addition, the unsigned overflow occurs when there is a carry out of the msb. This means that the correct answer is larger than the biggest unsigned number; that is FFFFH for word and FFH for a byte. On subtraction, unsigned overflow occur when there is a borrow into the msb . This means that the correct answer is smaller than 0.

2. Progam Segment Prefix:

When a program is loaded in memory, DOS prefaces it with a 256 byte program segment prefix (PSP). The PSP contains information about the program. So that programs may access this area, DOS places its segment number in both DS and ES before executing the program. The result is that DS does not contain the segment number of the data segment. To correct this, a program containing data segment begins with these two instructions: MOV AX, @DATA
MOV DS, AX

@DATA is the name of the data segment defined by .DATA. The assembler translates the name @DATA into a segment number

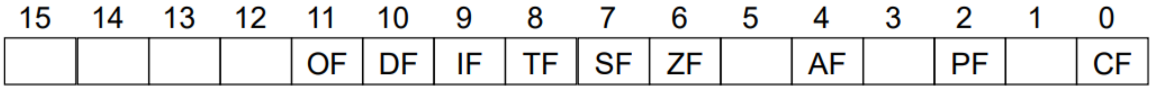
3. What is flag? How flag are affected signed and unsigned overflow?

A flag is a flip-flop that indicates some condition produced by the execution of an instruction. It also controls certain operations of the execution unit. A 16-bit flag register in the execution unit contains nine active flags.

4. Flag register of 8086 with neat diagram-

The circuits in the CPU can perform simple decision making based on the current state of the processor. For the 8086 processor, the processor state is implemented as nine individual bits called flags. Each decision made by the 8086 is based on the values of these flags.

A flag is a flip-flop that indicates some condition produced by the execution of an instruction. It also controls certain operations of the execution unit. A 16-bit flag register in the execution unit contains nine active flags. Six of the nine flags are called status flags and used to indicate some conditions produced by execution of some instructions. For example, when a subtraction operation results in a 0, the ZF (zero flag) is set to 1 (true). The three flags that are used to control certain operation of the processor are called control flags. They are used for interruption (interrupt flag - IF) and some string operations.



15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				OF	DF	IF	TF	SF	ZF		AF		PF		CF

Fig.5.1: The FLAGS Register.

5. Define addressing mode. Discuss based addressing mode and index addressing mode of 8086 processor.

The way an operand is specified is known as its addressing mode. The addressing modes we have used so far are

(1) Register Mode - which means that an operand is a register.
(2) Immediate Mode - when an operand is a constant.
(3) Direct Mode - when an operand is a variable.

For example,
MOV AX, 0 (Destination AX is register mode, source 0 is immediate mode.)
ADD ALPHA, AX (Destination ALPHA is direct mode, source AX is register mode.)

There are four additional addressing modes for the 8086:
(1) Register Indirect Mode
(2) Based Mode
(3) Indexed Mode
(4) Based Indexed Mode
These modes are used to address memory operands indirectly.

6. Discuss how logical instruction of 8086 are used to mask of instruction.

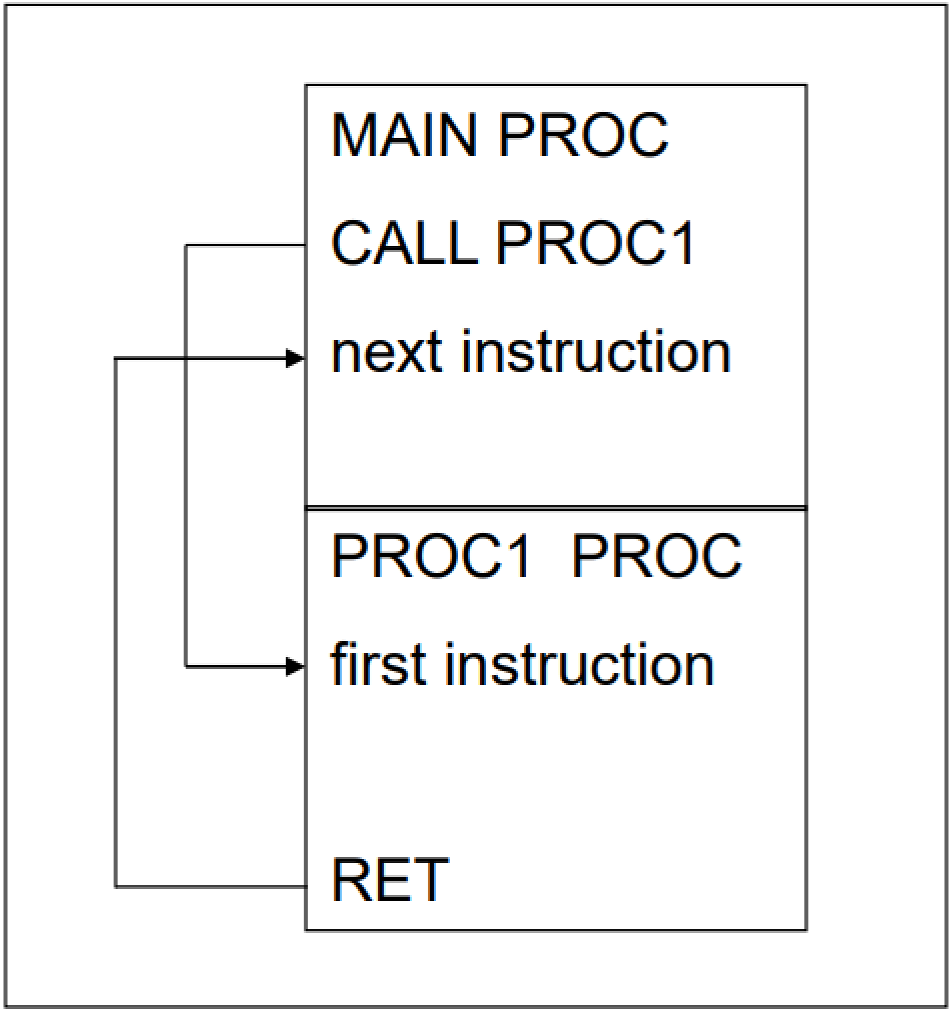
Mask: One use of AND, OR, and XOR is to selectively modify the bits in the destination. To do this, we construct a source bit pattern known as a mask. The mask bits are chosen so that the corresponding destination bits are modified in the desired manner when the instruction is executed.
Properties of AND, OR, and XOR

1. The AND instruction can be used to clear specific destination bits while preserving the others. A 0 mask bit clears the corresponding destination bit; a 1 mask bit preserves the corresponding destination bit.
2. The OR instruction can be used to set specific destination bits while preserving the others. A 1 mask bit sets the corresponding destination bit; a 0 mask bit preserves the corresponding destination bit.
3. The XOR instruction can be used to complement specific destination bits while preserving the others. A 1 mask bit complements the corresponding destination bit; a 0 mask bit preserves the corresponding destination bit.

7. Explain near type and far type procedure used in assembly language.

NEAR Type Procedure : For NEAR type procedure the statement that calls the procedure is in the same segment as the procedure itself.

FAR Type Procedure : For FAR type procedure the calling statement is in a different segment than the called procedure.



8. Define microprocessor. Mention different categories of functions performed by a microprocessor.

   Microprocessor: A microprocessor is a programmable logic device, designed with registers, flip-flops, and timing elements. It is a single chip made by LSI or VLSI technology. Microprocessor is also called as a brain of a computer system. It is the CPU of a microcomputer.

All the various functions performed by the microprocessor can be classified in three categories:

❑ Microprocessor-initiated operations.

❑ Internal data operations.

❑ Peripheral (or externally) initiated operations.

9. Write the steps by which a microprocessor communicates with peripherals or memory locations.

To communicate with a peripheral or a memory location, the MPU needs to perform the following steps:

❑ Step 1: Identify the peripheral or the memory location (with its address)

❑ Step 2: Transfer data.

❑ Step 3: Provide timing or synchronization signals.

10. Shortly discuss different index and pointer registers of 8086 microprocessor.

The SP, BP, SI, and DI registers normally point to memory locations. They contain the offset address of memory locations. Unlike segment registers, the pointer and index registers can be used in normal arithmetic operations.

SP (Stack Pointer) The SP register is used in conjunction with SS for accessing the stack segment.

BP (Base Pointer) The BP register is used primarily to access data on the stack. However, unlike SP, BP can also be used to data in other segments.

SI (Source Index) The SI register is used to point memory locations in the data segment. By incrementing the contents of SI, consecutive memory locations can be accessed.

DI (Destination Index) The DI register perform the same functions as SI. There is a class of instructions called string operations, that use DI to access memory locations addressed by ES.

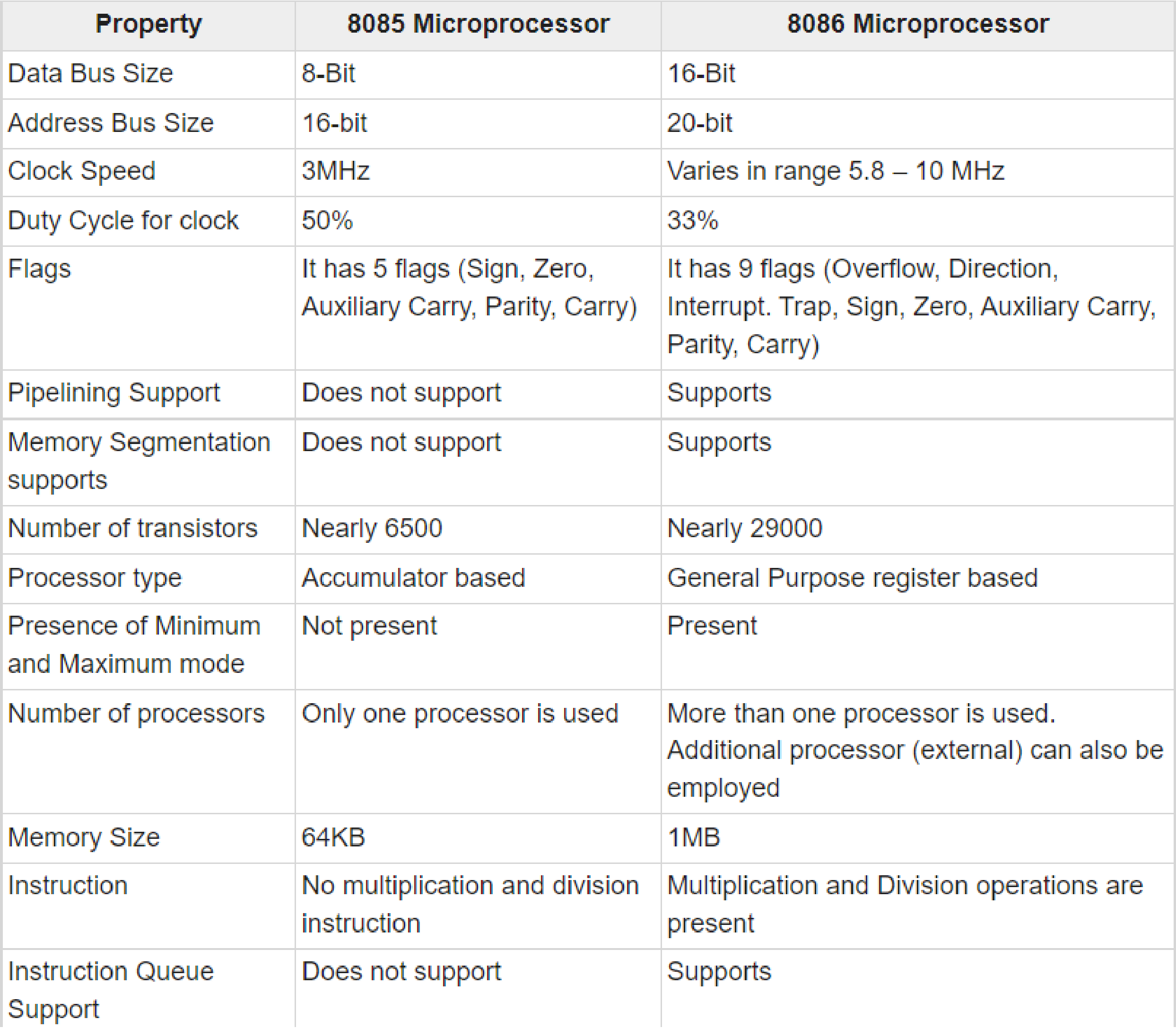
IP (Instruction Pointer) To access instructions, the 8086 uses the register CS and IP. The CS register contains, the segment number of the next instruction, and the IP contains the offset. IP is updated each time an instruction is executed.

11. Briefly discuss different data registers of 8086 microprocessor.

They are: AX, BX, CX, and DX. The high and low bytes of the data registers can be accessed separately. The high bytes are AH, BH, CH, DH and low bytes are AL, BL, CL, DL respectively. This arrangement gives us more registers to use when dealing with byte-size data.

AX (Accumulator Register) AX is preferred for arithmetic, logic and data transfer instructions. In multiplication and division operations, one of the numbers involved must be in AX or AL. Input and output operations also require the use of AL and AX.
BX (Base Register) BX is also serve as an address register.
CX (Count Register) Program loop constructions are facilitated by the use of CX, which serves as a loop counter. This register has important role in string, shift and rotate operations.
DX (Data Register) DX is used in multiplication and division. It is also used in I/O operations.

12. Differentiate 8085 and 8086 microprocessor on basis of data lines, address lines, memory capacity, and speed.



Property	8085 Microprocessor	8086 Microprocessor
Data Bus Size	8-Bit	16-Bit
Address Bus Size	16-bit	20-bit
Clock Speed	3MHz	Varies in range 5.8 – 10 MHz
Duty Cycle for clock	50%	33%
Flags	It has 5 flags (Sign, Zero, Auxiliary Carry, Parity, Carry)	It has 9 flags (Overflow, Direction, Interrupt. Trap, Sign, Zero, Auxiliary Carry, Parity, Carry)
Pipelining Support	Does not support	Supports
Memory Segmentation supports	Does not support	Supports
Number of transistors	Nearly 6500	Nearly 29000
Processor type	Accumulator based	General Purpose register based
Presence of Minimum and Maximum mode	Not present	Present
Number of processors	Only one processor is used	More than one processor is used. Additional processor (external) can also be employed
Memory Size	64KB	1MB
Instruction	No multiplication and division instruction	Multiplication and Division operations are present
Instruction Queue Support	Does not support	Supports