Flag Register of 8086

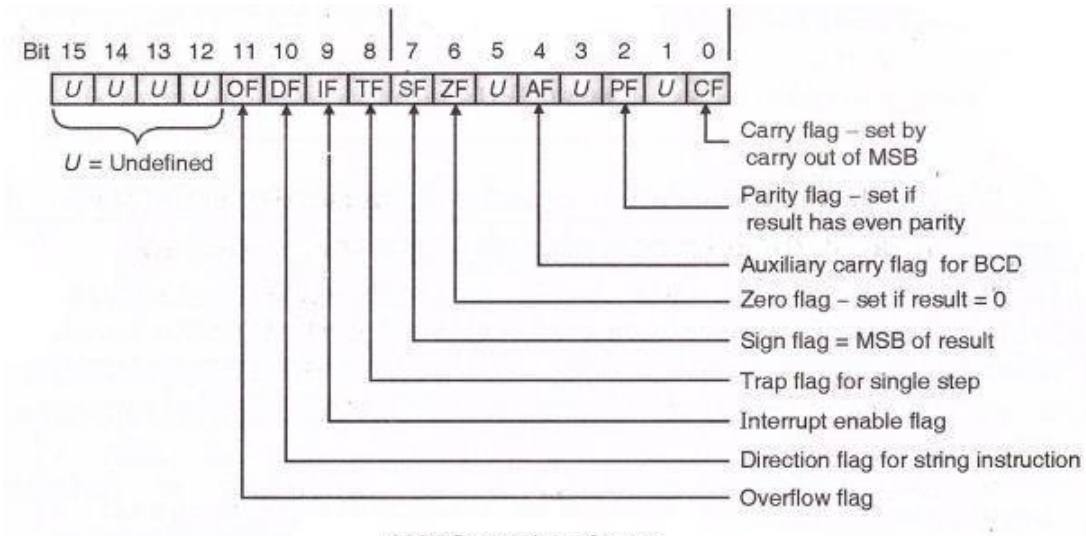
The Flag register is a Special Purpose Register. Depending upon the value of result after any arithmetic and logical operation the flag bits become set (1) or reset (0).

Flag register is a part of EU (Execution Unit). It is a 16 bit register with each bit corresponding to a flip-flop. It indicates some condition produced by the execution of an instruction. For example, the zero flag (ZF) will set if the result of execution of an instruction is zero.

The Flag register is the status register in Intel x86 microprocessors that contains the current state of the processor. Its successors, the EFLAGS and RFLAGS registers, are 32 bits and 64 bits wide, respectively. The wider registers retain compatibility with their smaller predecessors.

All Flags registers contain the condition codes, flag bits that let the results of one machine-language instruction affect another instruction. Arithmetic and logical instructions set some or all of the flags, and conditional jump instructions take variable action based on the value of certain flags.

For example, jz (Jump if Zero), jc (Jump if Carry), and jo (Jump if Overflow) depend on specific flags. Other conditional jumps test combinations of several flags. *Figure below shows the details of the 16-bit flag register of 8086 CPU.*



8086 flag register format

Flags register consists of 9 active flags out of 16. The remaining 7 flags marked ‘U’ are undefined flags. These 9 flags are of two types: 6 Status flags & 3 Control flags. So, you can divide the flag bits into two sections. The Status Flags, and the Control Flags.

A. *Status flags:*

1. *Carry flag (CF):*

It is set whenever there is a carry or borrow out of the MSB (most significant bit) of a result. D7 bit for an 8-bit operation and D15 bit for a 16-bit operation.

The carry flag is only relevant to unsigned numbers, not signed.

we want to add two unsigned numbers and the result is larger than "capacity" of register where it is saved.

- The carry flag is set if the addition of two numbers causes a carry out of the most significant (leftmost) bits added:

  1111 + 0001 = 0000 (carry flag is turned on, CF=1) [add]

  0111 + 0001 = 1000 (carry flag is turned off, CF=0) [add]

- The carry (borrow) flag is also set if the subtraction of two numbers requires a borrow into the most significant (leftmost) bits subtracted.

  0000 - 0001 = 1111 (carry flag is turned on, CF=1) [borrow]

  1000 - 0001 = 0111 (carry flag is turned off, CF=0) [not borrow]

2. *Parity flag (PF):*

This flag indicates whether the number of bits of the result of an operation is odd or even. EX:

If the result of the last operation were 26 (11010_2), the parity flag would be 0 since the number of set bits is odd. Similarly, if the result were 10 (1010_2) then the parity flag would be 1.

3. *Auxiliary carry flag (AF):*

Auxiliary Flag is used as CF but when working with BCD. So, AF will be set when we have overflow or underflow on in BCD** calculations. For example: considering 8-bit ALU unit, Auxiliary flag is set when there is carry from 3rd bit to 4th bit i.e., carry from lower nibble to higher nibble.

** Binary Coded Decimal or BCD is a 4-bit encoding system used to represent the decimal numbers or digits 0 to 9. For example, 123_{10} = 0001 0010 0011_2 as an 8-4-2-1 code. Since decimal digits range from 0 (0000) to 9 (1001), BCD uses the 4-bits to store just 10 combinations of numbers rather than the maximum possible 16, meaning as an encoding system it wastes storage space.

4. *Zero flag (ZF):*

It is set (1), when the result of an arithmetic operation is zero.

5. *Sign flag (SF):*

Sign Flag (SF) or negative flag, set to 1 when the result of the last mathematical operation is negative. set to 0 when the result of the last mathematical operation is positive.

6. *Overflow flag (OF):*

Overflow Flag is used as CF but when we work on signed numbers. Ex: we want to add two 8-bit signed numbers: 127 + 2. the result is 129 but it is too much for 8bit signed number, so OF will be set. Similar when the result is too small like -128 - 1 = -129 which is out of scope for 8-bit signed numbers.

B. Control flags:

1. *Trap flag (TF):*

It is used to set the trace mode i.e., start single stepping mode. Here the microprocessor is interrupted after every instruction so that the program can be debugged. The Trap Flag controls the operation of the microprocessor. (TF=0 Normal operation; TF=1 Single Step operation).

2. *Interrupt enable flag (IF):*

It is used to mask (disable) or unmask (enable) the INTR interrupt. If a user sets the IF flag, the CPU will recognize external interrupt requests. It determines whether or not the CPU will respond to mask-able hardware interruptions.

- If the flag is set to 1, mask-able hardware interrupts will be handled.
- If cleared (set to 0), such interrupts will be ignored.

3. *Direction flag (DF):*

This flag is specifically used in string instructions. If this flag is set, SI and DI are in an auto-decrementing mode in string operations.

- If the directional flag is set (1), then access the string data from higher memory location towards lower memory location.
- If the directional flag is reset (0), then access the string data from lower memory location towards higher memory location.