

Flag register of 8086 microprocessor

Last Updated : 05 May, 2023

Prerequisite – [Flag register in 8085 microprocessor](#) The Flag register is a Special

Introduction :

The flag register is a 16-bit register in the Intel 8086 microprocessor that contains information about the state of the processor after executing an instruction. It is sometimes referred to as the status register because it contains various status flags that reflect the outcome of the last operation executed by the processor.

The flag register is an important component of the 8086 microprocessor because it is used to determine the behavior of many conditional jump and branch instructions. The various flags in the flag register are set or cleared based on the result of arithmetic, logic, and other instructions executed by the processor.

The flag register is divided into various bit fields, with each bit representing a specific flag. Some of the important flags in the flag register include the carry flag (CF), the zero flag (ZF), the sign flag (SF), the overflow flag (OF), the parity flag (PF), and the auxiliary carry flag (AF). These flags are used by the processor to determine the outcome of conditional jump instructions and other branching instructions.

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

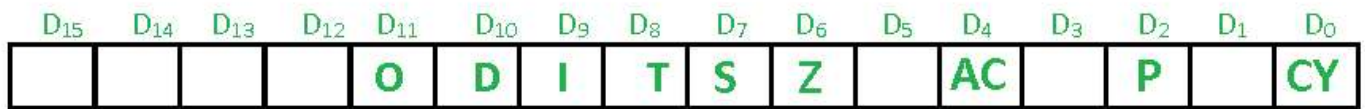


Figure – Format of flag register There are total 9 flags in 8086 and the flag register is divided into two types: **(a) Status Flags** – There are 6 flag registers in 8086 microprocessor which become set(1) or reset(0) depending upon condition after either 8-bit or 16-bit operation. These flags are conditional/status flags. 5 of these flags are same as in case of 8085 microprocessor and their working is also same as in 8085 microprocessor. The sixth one is the overflow flag. The 6 status flags are:

1. **Sign Flag (S)**
 2. **Zero Flag (Z)**
 3. **Auxiliary Carry Flag (AC)**
 4. **Parity Flag (P)**
 5. **Carry Flag (CY)** These first five flags are defined here
 6. **Overflow Flag (O)** – This flag will be set (1) if the result of a signed operation is too large to fit in the number of bits available to represent it, otherwise reset (0). After any operation, if D[6] generates any carry and passes to D[7] OR if D[6] does not generate carry but D[7] generates, overflow flag becomes set, i.e., 1. If D[6] and D[7] both generate carry or both do not generate any carry, then overflow flag becomes reset, i.e., 0.
0. **Example:** On adding bytes 100 + 50 (result is not in range -128...127), so overflow flag will set.

```
MOV AL, 50 (50 is 01010000 which is positive)
MOV BL, 32 (32 is 00110010 which is positive)
ADD AL, BL (82 is 10000010 which is negative)
```

1. Overflow flag became set as we added 2 +ve numbers and we got a -ve number.

(b) Control Flags – The control flags enable or disable certain operations of the microprocessor. There are 3 control flags in 8086 microprocessor and these are:

1. **Directional Flag (D)** – This flag is specifically used in string instructions. If directional flag is set (1), then access the string data from higher memory location towards lower memory location. If directional flag is reset (0), then access the string data from lower memory location towards higher memory location.
2. **Interrupt Flag (I)** – This flag is for interrupts. If interrupt flag is set (1), the microprocessor will recognize interrupt requests from the peripherals. If interrupt flag is reset (0), the microprocessor will not recognize any interrupt requests and will ignore them.
3. **Trap Flag (T)** – This flag is used for on-chip debugging. Setting trap flag puts the microprocessor into single step mode for debugging. In single stepping, the microprocessor executes a instruction and enters into single step ISR. If trap flag is set (1), the CPU automatically generates an internal interrupt after each instruction, allowing a program to be inspected as it executes instruction by instruction. If trap flag is reset (0), no function is performed.

Uses of Flag register in 8086 microprocessor :

The flag register in the 8086 microprocessor has several important uses, including:

1. **Conditional branching:** The flags in the flag register can be used to control conditional branching in assembly language programming. Conditional jump instructions allow a program to take different paths based on the state of the flags in the flag register.
2. **Arithmetic and logic operations:** The flag register is used to store the results of arithmetic and logic operations. The flags in the flag register provide information about the outcome of these operations, such as whether a result is negative or zero, or whether there was an overflow or carry.
3. **Error detection and handling:** The flag register can be used to detect errors and exceptions, such as overflow or divide-by-zero errors. This allows programs to handle these errors gracefully and to take appropriate corrective action.
4. **Debugging:** The flag register provides a convenient way to access important information about the status of the processor after executing an instruction. This information can be used to debug programs and to optimize performance.
5. **Optimization:** The flag register can be used to optimize the performance of assembly language programs by avoiding unnecessary instructions or reducing the number of conditional jumps required.

Advantages :

The flag register in the 8086 microprocessor provides several advantages, including:

1. **Efficient conditional branching:** The flag register enables efficient conditional branching in assembly language programming. Programmers can use conditional jump instructions to make decisions based on the state of the flags in the flag register, allowing for more efficient and optimized code.
2. **Improved arithmetic and logic operations:** The flag register is used to store the results of arithmetic and logic operations, allowing for more complex calculations to be performed efficiently. The various flags in the flag register provide information about the outcome of these operations, such as whether a result is negative or zero, or whether there was an overflow or carry.
3. **Easy access to processor status information:** The flag register provides a convenient way to access important information about the status of the processor after executing an instruction. This information can be used to debug programs and to optimize performance.
4. **Improved error handling:** The flag register can be used to detect errors and exceptions, such as overflow or divide-by-zero errors. This allows programs to handle these errors gracefully and to take appropriate corrective action.

Dis-advantages :

There are not many disadvantages of the flag register in the 8086 microprocessor, but some potential drawbacks include:

1. **Limited number of flags:** The 8086 flag register has a limited number of flags, which can make it difficult to handle complex calculations or to detect certain types of errors or exceptions.
2. **Limited precision:** The flags in the flag register are often limited in precision, which can lead to inaccuracies or errors in certain types of calculations or operations.
3. **Difficulty in understanding and using:** The flag register can be difficult for beginners to understand and use effectively, which can lead to mistakes or errors in programming.
4. **Overhead in execution time:** Accessing the flag register can add overhead to program execution time, which can impact performance in certain types of applications.