Processor Status and Flags Register

Computer Organization and Assembly Language

Agenda

- The microprocessor status
- The FLAGS register
- Signed and unsigned overflow
- Instructions affecting FLAGS register

Processor Status

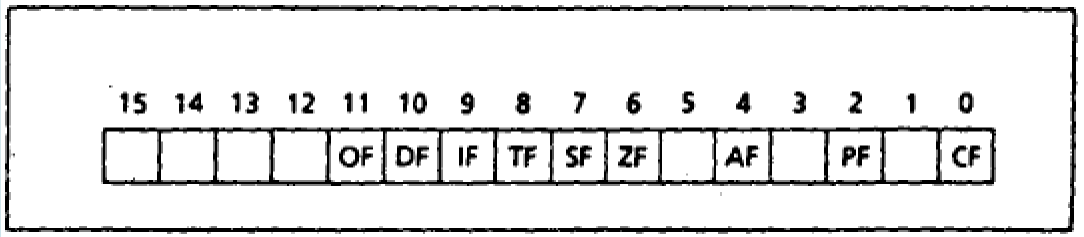
- The circuits in the CPU perform simple decision making based on the current state of the processor.
- In 8086 processor, the processor state is implemented as nine individual bits called flags.
- Each decision made by the 8086 is based on the values of these flags.

Flags Register

- The flags are placed in the FLAGS register and are either status flags or control flags.
- Status flags reflect the result of a computation.

Flags Register

- The status flags are bits 0, 2, 4, 6, 7, and 11.
- The control flags are located in bits 8, 9, and10.
- The other bits have no significance.



Flags Register

- *Flag Numbers, Names and Symbols*

Status Flags

Bit	Name	Symbol
0	Carry flag	CF
2	Parity flag	PF
4	Auxiliary carry flag	AF
6	Zero flag	ZF
7	Sign flag	SF
11	Overflow flag	OF

Control Flags

Bit	Name	Symbol
8	Trap flag	TF
9	Interrupt flag	IF
10	Direction flag	DF

Status Flags - Carry Flag (CF)

- The status flags to reflect the result of an operation.
  - Example, If SUB AX,AX is executed, the zero flag becomes 1, indicating that a zero result was produced.
- CF is 1 if there is a carry out from the most significant bit (msb) on addition, or there Is a borrow into the msb on subtraction; otherwise, its 0.
- CF is also affected by shift and rotate Instructions

Status Flags - Parity Flag (PF)

- PF = 1 if the low byte of a result has an even number of one bits (even parity).
- It is 0 if the low byte has odd parity.
- For example, if the result of a word addition is FFFEh, then the low byte contains 7 one bits, so PF = 0.

Status Flags - Auxiliary Carry Flag (AF)

- AF = 1 if there is a carry out from bit 3 on addition, or a borrow into bit 3 on subtraction.
- AF is used in binary-coded decimal (BCD) operations.

Status Flags - Zero Flag (ZF)

- ZF = 1 for a zero result, and ZF = 0 for a nonzero result.

Status Flags - Sign Flag (SF)

- SF = 1 if the msb of a result is 1; it shows the result is negative if a signed interpretation is used.
- SF = 0 if the msb is 0 .

Status Flags - Overflow Flag (OF)

- OF = 1 if signed overflow occurred, othcrwise it is 0.

Overflow

- The range of signed 16 bit numbers or word is -32768 to 32767; for 8 bit the range is -128 to 127.
- For unsigned numbers, the range for a word is 0 to 65535; for a byte, it is 0 to 255.
- If the result of an operation falls outside these ranges, overflow occurs and the truncated result will be incorrect.
- For an arithmetic operation such as addition, there are four possible outcomes: (1) no overflow, (Z) signed overflow only, (3) unsigned overflow only, and (4) both signed and unsigned overflows.

Overflow - Unsigned

- The example below is an unsigned overflow but not signed overflow, suppose AX contains FFFFh, BX contains 0001h, and ADD AX,BX is executed.
- The binary result is-

```
  1111 1111 1111 1111
+ 0000 0000 0000 0001
---------------------
1 0000 0000 0000 0000
```

- The result is 10000h=65536, but it is out of range for a word. A 1 is carried out of the msb and wrong answer stored in AX, 0000h, so unsigned overflow occurred.
- But the stored answer is correct as a signed number, as FFFFh = -1 0001h = 1, and FFFFh + 0001h = -1 + 1 = 0, so signed overflow did not occur.

Overflow - Signed

- Let AX and BX both contain 7FFFh, and instruction ADD AX, BX is executed

```
  0111 1111 1111 1111
+ 0111 1111 1111 1111
---------------------
  1111 1111 1111 1110 = FFFEh
```

- The signed and unsigned decimal interpretation of 7FFFh IS 32767.
- Thus for both signed and unsigned addition, 7FfFh + 7FFfh = 32767 + 32767 = 65534.
- This is out of range for signed numbers; the signed interpretation of the stored answer FFFEh is -2. so signed overflow occurred.
- Unsigned interpretation of FFFEh is 65534, which is correct answer, so there is no unsigned overflow.

Overflow and the Flags

- The processor sets OF= 1 for signed overflow and CF = 1 for unsigned overflow.
- It is then up to the program to take appropriate action,
- The result of a subsequent instruction may cause the overflow flag to be turned off.
- Processor turns on CF or OF for unsigned overflow or signed overflow, respectively.

Unsigned Overflow and the Flags

- On addition, unsigned overflow occurs when there is a carry out of the msb.
  - Tha is, the correct answer is larger than the biggest unsigned number; that is, FFFFh for a word and FFh for a byte.
- On subtraction, unsigned overflow occurs when there is a borrow into the msb.
  - That is, the correct answer is smaller than 0.

Signed Overflow and the Flags

- On addition of numbers with the same sign, signed overflow occurs when the sum has a different sign.
- This happens when adding 7FFFh and 7FFFh (two positive numbers), getting FFFEh (a negative result).
- Subtraction of numbers with different signs is like adding numbers of the same sign.
  - Example, A - (-B) = A + B and -A -(+B) = -A -B.
- Signed overflow occurs if the result has a different sign than expected.

Instructions Affecting Flags

- Each time the an instruction is executed, the flags are altered to reflect the result.
- Here is an example instruction and the flags affected upon its execution.
- ADD AX, BX ; AX has FFFFh, BX contains FFFFh.
- Now AX contains result FFFEh and the status of flags is:
- SF = 1 because the msb is 1.
- PF = 0 since there are 7 (odd number) or 1 hits in the low byte of the result.
- ZF = 0 because the result is nonzero.
- CF = 1 because there is out of the msb on addition.
- OF = 0 because the sign of the stored result is the same as that of the number being added.
- [See details on page 85 of the book]