Processor Status & the FLAGES Registers

Covering topics

- Overview
- Flags Registers
- How processor indicates overflow?
- How processor determines that overflow occurred?
- How instructions effects the flags?

Overview

- One important factor of computer is that it makes decision. The circuits in the CPU can perform simple decision making by observing the current state of the processor.
- The 8086 processor state is implemented by using nine individual bits called flags. Decision is based on the value of these flags
- Status Flags and Control Flags.
- How the status Flags are effected by the result of a computation.
- Using control flags to enable and disable certain operations of the processor in next lesson.

The FLAGS Registers

- A register that is used to indicate the state of the processor or to control the process certain operation.
- Status Flags: The status flags are located in bits 0,2,4,6,7 and 11.
- Control Flags: These flags are located in bits 8,9 and 10. Other bits have no significances.
- Note: It is not important to remember which bit is for which flag , the names of the flags and their symbols are listed in the next slide.

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

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

Status Flags

- The process use the status flags to reflect the result of an operation.
- For example SUB AX,AX will set the ZF(zero flag) to 1 indicating that zero result is produced. Now let us discuss each status flags as:

Carry Flag(CF):

- CF is 1 if there is a carry out from the MSB(most significant bit) on addition, or there is a borrow on subtraction, other wise it is zero.
- Also affected by shift and rotate instructions.

Parity Flag(PF):

- PF is 1 if the lower byte of a result has an even number of one bits(even parity).
- It is 0 if the low byte has odd parity.

Auxiliary Carry Flag(AF):

AF is 1 if there is a carry out from bit 3 on addition, or a borrow into bit 3 on subtraction.

Zero Flag(ZF):

ZF is 1 for a zero result and is 0 for a non zero result.

Sign Flag(SF):

SF=1 if the MSB of a result is 1; it means the result is negative if you are giving a signed interpretation. SF is 0 if the MSB is 0.

Overflow Flag(OF):

OF is 1 if signed overflow occurred. Otherwise it is 0.

Auxiliary Carry Flag(AF):

AF is 1 if there is a carry out from bit 3 on addition, or a borrow into bit 3 on subtraction.

Zero Flag(ZF):

ZF is 1 for a zero result and is 0 for a non zero result.

Sign Flag(SF):

SF=1 if the MSB of a result is 1; it means the result is negative if you are giving a signed interpretation. SF is 0 if the MSB is 0.

Overflow Flag(OF):

OF is 1 if signed overflow occurred. Otherwise it is 0.

How processor indicates that overflow has occurred?

- Processor sets OF as 1 for signed overflow and CF as 1 for unsigned overflow.
- In determining overflow the processor does not interpret the result as either signed or unsigned.
- It is the programmer, who interpret the result, in signed interpretation only OF is of interest and CF can be ignored, while in case of unsigned CF is important not OF.

How processor determines that overflow has occurred?

- Many instructions can cause the overflow for the simplicity we will limit the discussion to addition and subtraction.
- **Unsigned Overflow:** ***On addition,*** unsigned overflow occurs when there is a carry out from MSB. ***On subtraction,*** unsigned overflow occurs when there is a borrow into the MSB.
- **Singed Overflow:** On addition of numbers with same sign, signed overflows occurs when the sum has a different sign. Subtraction of two number with different signs is like adding numbers of the same sign. Signed over flow occurs if the result has a different sign then expected.

How instructions effects the Flags?

- Each time the processor execute an instruction, the flags are altered to reflect the result. However some instructions don’t effect any of them, affect only some of them, are may leave them unaffected.
- Seven basic instruction we studied in previous lecture affect the flags as:

Instruction	Flags affected
MOV/XCHG	None
ADD/SUB	All
INC/DEC	All except CF
NEG	All

- How these instructions affect the flags let us examine an example.

Example: ADD AX,BX , where AX=FFFFh and BX=FFFFh

Solution:

$$\begin{array}{r} \text{FFFFh} \\ +\underline{\text{FFFFh}} \\ \text{1FFFEh} \end{array}$$

The result stored in AX is FFFEh=1111 1111 1111 1110

SF=1
PF=0
ZF=0
CF=1
OF=0

Flags Symbols

Table: 5.2 Flag Symbols

Status Flag	Set(1) Symbol	Clear(0) Symbol
CF	CY (carry)	NC(no carry)
PF	PE(even parity)	PO(odd parity)
AF	AC(auxiliary carry)	NA(no auxiliary carry)
ZF	ZR(zero result)	ZX(non zero)
SF	NF(negative result)	PL(plus)
OF	OV(overflow)	NV(no overflow)
Control Flags		
DF	DN(down)	UP(up)
IF	EI(enable interrupts)	DI(disable interrupts)

References

- Assembly Language Programming and Organization of the IBM PC (Ytha Yu, Charles Marut)