

The Processor Status And The FLAGS Register

Anika Sayara
anikasayara@outlook.com

Flags Register

- In 8086, the processor state is implemented as *nine individual bits called flags*.
- Each decision made by 8086 is *based on the values of these flags*
- The flags are *placed in the FLAGS register*
- Two types of flags: Status flags and control flags
- *Status flags* reflect the result of a computation. They are *located in bits 0,2,4,6,7 and 11*
- *Control flags* enable or disable certain operations of the processor. They are *located in bits 8,9 and 10*
- The other bits have no significance

Flags Register

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

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
8	Trap Flag	TF
9	Interrupt Flag	IF
10	Direction Flag	DF

Status Flag

Control Flag

The Status Flags

Anika Sayara
anikasayara@outlook.com

Carry Flag

- **CF=1**
 - if there is a *carry out in the MSB* on addition
 - if there is a *borrow into the MSB* on subtraction
- **Otherwise CF=0**
- It is also affected by *shift and rotate instructions*

Parity Flag

- **PF=1** if the low byte of a result *has even parity*
- **PF=0** if the low byte of a result *has odd parity*

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
- **Otherwise AF=0**
- **AF is used in BCD operations**

Zero Flag(ZF)

- ZF=1 for *a zero result*
- ZF=0 for *a non-zero result*

Sign Flag(SF)

- **SF=1** if the *MSB of a result is 1* that means the result is negative
- **SF=0** if the *MSB of a result is 0* that means the result is positive

Overflow Flag(OF)

- OF=1 if *signed overflow* occurred
- Otherwise OF=0

Overflow

Anika Sayara
anikasayara@outlook.com

Overflow Flag(OF)

- The range of numbers that can be represented in a computer is limited.
- The range of signed numbers that can be represented by a
 - 16-bit word is -32768 to 32767
 - 8-bit byte is -128 to 127
- The range of unsigned numbers that can be represented by a
 - 16-bit word is 0 to 65535
 - 8-bit byte is 0 to 255
- If the *result of an operation falls out of these ranges then overflow occurs* and the truncated result that is saved will be incorrect

Overflow Flag(OF)

When we perform an arithmetic operation, there are *four possible outcomes*:

- *No overflow*
- *Signed overflow only*
- *Unsigned overflow only*
- *Both signed and unsigned overflows*

Suppose *AX contains FFFFh(-1)*
BX contains 0001h(1)

ADD AX,BX

Example of Unsigned
Overflow Only

Suppose *AX contains FFFFh(-1)*

BX contains 0001h(1)

ADD AX,BX

	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
+	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
<hr/>																
	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Example of Unsigned
Overflow Only

Example of Signed Overflow Only

Suppose *AX contains FFFFh(-1)*

BX contains 0001h(1)

ADD AX,BX

	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
+	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
<hr/>																	
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Unsigned interpretation

the correct answer 10000h=65535 but this is out of range for a word operation. 1 is carried out of the MSB and the answer stored in AX is 0000h which is wrong *so unsigned overflow occurs*

Signed interpretation

But *signed overflow does not occur* as the stored answer is correct as a signed number (+1-1=0).

Suppose *AX contains 7FFFh(32767)*

BX contains 7FFFh(32767)

ADD AX,BX

Example of Unsigned
Overflow Only

Suppose *AX* contains 7FFFh(32767)

BX contains 7FFFh(32767)

ADD AX,BX

	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
+	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
<hr/>																
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0

Example of Unsigned
Overflow Only

Suppose *AX contains 7FFFh(32767)*

BX contains 7FFFh(32767)

ADD AX,BX

$$\begin{array}{r} 0\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1 \\ +\ 0\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1 \\ \hline 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 0 \end{array}$$

Unsigned interpretation

The unsigned interpretation of FFFEh is 65534, which is the right answer. So *unsigned overflow does not occur*

Signed interpretation

If it is a signed interpretation the answer is FFFEh = -2 but this is incorrect as the result should be 65534. So *signed overflow occurs*

Example of Unsigned
Overflow Only

How the processor indicates overflow?

Anika Sayara
anikasayara@outlook.com

How the processor
indicates overflow?

- **The processor sets OF=1 for signed overflow**
- **The processor sets CF=1 for unsigned overflow**

How the Processor Determines that Overflow Occured

Anika Sayara
anikasayara@outlook.com

Unsigned Overflow

- *On addition when there is a carry out in the MSB.* This means the result is larger than the biggest unsigned number
- *On subtraction when there is a borrow in the MSB.* This means the correct answer is smaller than 0

Signed Overflow

If the *carries into and out of the msb don't match*-that is, there is a carry into the msb but no carry out, or if there is a carry out but no carry in –then signed overflow has occurred

How instructions affect the flag

Anika Sayara
anikasayara@outlook.com

Effect of instructions on flags

Instruction	Affected flags
MOV/XCHG	None
ADD/SUB	all
INC/DEC	All except CF
NEG	All CF=1 unless result is 0, OF=1 if word operand is 8000h or byte operand is 80h

Examples

Anika Sayara
anikasayara@outlook.com

Example 5.1

ADD AX,BX. where AX contains FFFFh. BX contains FFFFh.

Example 5.2

ADD AL,BL, where AL contains 80h, BL contains 80h.

Example 5.3

SUB AX,BX, where AX contains 8000h and BX contains 0001h.