

Addition and Subtraction Instructions

Outline of the Lecture

- INC and DEC Instructions.
- ADD and SUB Instructions.
- NEG Instruction.
- Implementing Arithmetic Expressions.
- Flags Affected by Arithmetic Operations.
 - Zero
 - Carry
 - Parity
 - Auxiliary
 - Sign
 - Overflow
- Programming Example
- Programming Exercises

INC and DEC Instructions

```
INC destination; destination ← destination + 1
```

```
INC reg/mem
```

```
DEC destination; destination ← destination - 1
```

```
DEC reg/mem
```

- Operand may be register or memory
- Overflow, Sign, Zero, Auxiliary Carry, and Parity Flags changed as needed. Does NOT affect Carry flag.

INC and DEC Examples

```
.data
myWord  WORD 1000h
myDword DWORD 10000000h
.code
inc myWord    ; 1001h
dec myWord    ; 1000h
inc myDword   ; 10000001h
mov ax,00FFh
inc ax; AX = 0100h
mov ax,00FFh
inc al; AX = 0000h
```

Show the value of the destination operand after each of the following instructions executes:

```
.data
myByte  BYTE 0FFh, 0
.code
mov al,myByte      ; AL = FFh
mov ah,[myByte+1]  ; AH = 00h
dec ah; AH = FFh
inc al; AL = 00h
dec ax; AX = FEFF
```

ADD and SUB Instructions

ADD destination, source; destination ← destination + source

SUB destination, source; destination ← destination - source

Same operand rules as for the MOV instruction

ADD and SUB Examples

```
.data
var1 DWORD 10000h
var2 DWORD 20000h
.code
mov eax,var1 ; 00010000h
add eax,var2 ; 00030000h
add ax,0FFFFh ; 0003FFFFh
add eax,1 ; 00040000h
sub ax,1 ; 0004FFFFh
```

NEG (negate) Instruction

NEG reg
NEG mem

- Reverses the sign of an operand. Operand can be a register or memory operand.
- Carry, Overflow, Sign, Zero, Auxiliary Carry, Parity Flags changed as needed.

Example1

```
.data
valB BYTE -1
valW WORD +32767
.code
mov al,valB ; AL = -1
neg al ; AL = +1
neg valW ; valW = -32767
```

Example1

```
.data
valB BYTE 1,0
valC SBYTE -128
.code
neg valB ; CF = 1, OF = 0
neg [valB + 1] ; CF = 0, OF = 0
neg valC ; CF = 1, OF = 1
```

Implementing Arithmetic Expressions

Example:

$Rval = -Xval + (Yval - Zval)$

```
.data
Rval SDWORD ?
Xval SDWORD 26
Yval SDWORD 30
Zval SDWORD 40
```

```

.code
mov eax,Xval
neg eax      ; EAX = -26
mov ebx,Yval
sub ebx,Zval ; EBX = -10
add eax,ebx
mov Rval,eax ; -36

```

Flags Affected by Arithmetic Operation

- The ALU has a number of status flags that reflect the outcome of arithmetic (and bitwise) operations based on the contents of the destination operand
 - Essential flags:
 - **Unsigned:**
 - Zero flag (**ZF**) – set when destination equals zero
 - The Parity flag (**PF**)
 - Auxiliary flag (**AF**)
 - Carry flag (**CF**) – set when unsigned value is out of range
 - **Signed:**
 - Sign flag (**SF**) – set when destination is negative
 - Overflow flag (**OF**) – set when signed value is out of range
 - ⊕ The Carry flag indicates **unsigned integer overflow**. For example, if an instruction has an 8-bit destination operand but the instruction generates a result larger than 11111111 binary, the Carry flag is set.
 - ⊕ The Overflow flag indicates **signed integer overflow**. For example, if an instruction has a 16-bit destination operand but it generates a negative result smaller than -32,768 decimal, the Overflow flag is set.
 - ⊕ The Zero flag indicates that an operation produced zero. For example, if an operand is subtracted from another of equal value, the Zero flag is set.
 - ⊕ The Sign flag indicates that an operation produced a negative result. If the most significant bit of the destination operand is set, the Sign flag is set.
 - ⊕ The Parity flag counts the number of 1 bits in the least significant byte of the destination operand.
 - ⊕ The Auxiliary flag is set when a 1 bit carries out of position 3 in the least significant byte of the destination operand.
- The **data transfer instruction** never affects the flags.

Unsigned - Addition and subtraction of *unsigned* numbers is invalid whenever there is a *carry out*, CF = 1.

Signed - Addition and subtraction of *signed* numbers is invalid whenever there is an *overflow*, OF = 1. The result is valid as a signed number when OF=0.

Zero Flag (ZF) Example:

```

mov cx,1
sub cx,1 ; CX = 0, ZF = 1
mov ax,0FFFFh
inc ax   ; AX = 0, ZF = 1
inc ax   ; AX = 1, ZF = 0

```

Carry Flag (CF) Example

```

mov al,0FFh

```

```

add al,1 ; CF = 1, AL = 00
; Try to go below zero:
mov al,0
sub al,1 ; CF = 1, AL = FF

```

Sign Flag (SF) Example:

```

mov cx,0
sub cx,1 ; CX = -1, SF = 1
add cx,2 ; CX = 1, SF = 0
;The sign flag is a copy of the destination's highest bit:
mov al,0
sub al,1 ; AL = 11111111b, SF = 1
add al,2 ; AL = 00000001b, SF = 0

```

Overflow Flag (OF) Example:

```

; Example 1
mov al,+127
add al,1 ; OF = 1, AL = ??
; Example 2
mov al,7Fh ; OF = 1, AL = 80h
add al,1

```

- When adding two integers, remember that the Overflow flag is only set when . . .
 - Two positive operands are added and their sum is negative
 - Two negative operands are added and their sum is positive

Signed and Unsigned Integers: A Hardware Viewpoint

- All CPU instructions operate exactly the same on signed and unsigned integers
- The CPU cannot distinguish between signed and unsigned integers
- YOU, the programmer, are solely responsible for using the correct data type with each instruction

OF = CF XOR MSB

Programming Example:

```

TITLE Addition and Subtraction (AddSub3.asm)
; Chapter 4 example. Demonstration of ADD, SUB,
; INC, DEC, and NEG instructions, and how
; they affect the CPU status flags.
INCLUDE Irvine32.inc
.data
    Rval SDWORD ?
    Xval SDWORD 26
    Yval SDWORD 30
    Zval SDWORD 40
.code
main PROC
    ; INC and DEC
    mov ax,1000h
    inc ax ; 1001h
    dec ax ; 1000h
    ; Expression: Rval = -Xval + (Yval - Zval)
    mov eax,Xval

```

```

neg    eax                ; -26
mov    ebx,Yval
sub    ebx,Zval           ; -10
add    eax,ebx
mov    Rval,eax          ; -36
; Zero flag example:
mov    cx,1
sub    cx,1              ; ZF = 1
mov    ax,0FFFFh
inc    ax                ; ZF = 1
; Sign flag example:
mov    cx,0
sub    cx,1              ; SF = 1
mov    ax,7FFFh
add    ax,2              ; SF = 1
; Carry flag example:
mov    al,0FFh
add    al,1              ; CF = 1,  AL = 00
; Overflow flag example:
mov    al,+127
add    al,1              ; OF = 1
mov    al,-128
sub    al,1              ; OF = 1
exit
main ENDP
END main

```

Programming Exercises

1. Indicate whether or not each of the following instructions is valid.
 - a. `add ax,bx`
 - b. `add dx,b1`
 - c. `add ecx,dx`
 - d. `sub si,di`
 - e. `add bx,90000`
 - f. `sub ds,1`
 - g. `dec ip`
 - h. `dec edx`
 - i. `add edx,1000h`
 - j. `sub ah,126h`
 - k. `sub al,256`
 - l. `inc ax,1`
2. What will be the value of the Carry flag after each of the following instruction sequences has executed?
 - a. `mov ax,0FFFFh`
`add ax,1`
 - b. `mov bh,2`
`sub bh,2`
 - c. `mov dx,0`
`dec dx`

- d. `mov al,0DFh`
`add al,32h`
- e. `mov si,0B9F6h`
`sub si,9874h`
- f. `mov cx,695Fh`
`sub cx,A218h`

3. What will be the value of the Zero flag after each of the following instruction sequences has executed?

- a. `mov ax,0FFFFh`
`add ax,1`
- b. `mov bh,2`
`sub bh,2`
- c. `mov dx,0`
`dec dx`
- d. `mov al,0DFh`
`add al,32h`
- e. `mov si,0B9F6h`
`sub si,9874h`
- f. `mov cx,695Fh`
`add cx,96A1h`

4. What will be the value of the Sign flag after each of the following instruction sequences has executed?

- a. `mov ax,0FFFFh`
`sub ax,1`
- b. `mov bh,2`
`sub bh,3`
- c. `mov dx,0`
`dec dx`
- d. `mov ax,7FFEh`
`add ax,22h`
- e. `mov si,0B9F6h`
`sub si,9874h`
- f. `mov cx,8000h`
`add cx,A69Fh`

5. What will be the values of the Carry, Sign, and Zero flags after the following instructions have executed?

```
mov ax,620h
sub ah,0F6h
```

6. What will be the values of the Carry, Sign, and Zero flags after the following instructions have executed?

```
mov ax,720h
sub ax,0E6h
```

7. What will be the values of the Carry, Sign, and Zero flags after the following instructions have executed?

```
mov ax,0B6D4h
add al,0B3h
```

8. What will be the values of the Overflow, Sign, and Zero flags after the following instructions have executed?

```
mov bl,-127
```

```
dec bl
```

9. What will be the values of the Carry, Overflow, Sign, and Zero flags after the following instructions have executed?

```
mov cx,-4097
```

```
add cx,1001h
```

10. What will be the values of the Carry, Overflow, Sign, and Zero flags after the following instructions have executed?

```
mov ah,-56
```

```
add ah,-60
```