

# Introduction to 8086 Assembly

## Lecture 5

**Jump, Conditional Jump, Looping, Compare instructions**

# Labels and jumping (the jmp instruction)



K. N. Toosi  
University of Technology

```
mov eax, 1
```

```
add eax, eax
```

```
jmp label1
```

```
xor eax, eax
```

```
label1:
```

```
sub eax, 303
```

# Labels and jumping (the jmp instruction)



K. N. Toosi  
University of Technology

```
mov eax, 1
```

```
add eax, eax
```

```
jmp label1
```

```
xor eax, eax
```

```
label1: → address of sub eax, 303
```

```
sub eax, 303
```

# Infinite loop



K. N. Toosi  
University of Technology

```
mov eax, 0

loop1:

    call print_int
    call print_nl

    inc eax

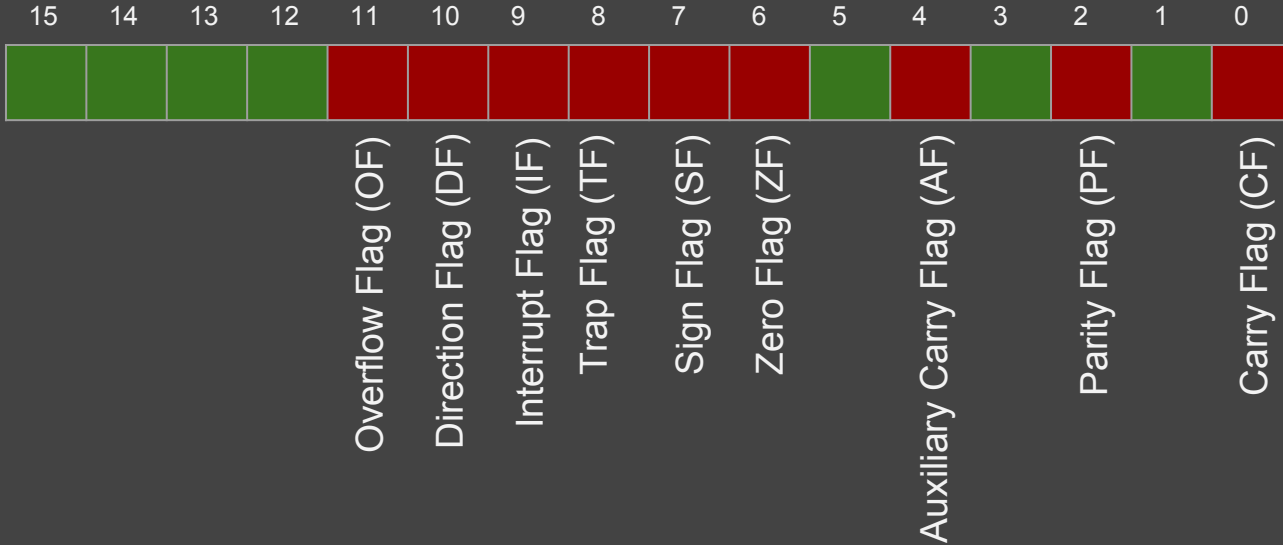
    jmp loop1
```

infinite\_loop.asm

# Remember: the FLAGS Register



K. N. Toosi  
University of Technology



**CF:** carry flag

**OF:** overflow flag

**SF:** sign flag

**ZF:** zero flag

**PF:** parity flag

**DF:** direction flag

**IF:** interrupt flag

# Conditional jumps



<b>JZ</b>	Jump if ZF=1
<b>JNZ</b>	Jump if ZF=0
<b>JO</b>	Jump if OF=1
<b>JNO</b>	Jump if OF=0
<b>JS</b>	Jump if SF=1
<b>JNS</b>	Jump if SF=0
<b>JC</b>	Jump if CF=1
<b>JNC</b>	Jump if CF=0
<b>JP</b>	Jump if PF=1
<b>JNP</b>	Jump if PF=0

# Example

unsigned integer:

```
if (eax == ebx)
    esi = 0
```



K. N. Toosi  
University of Technology

<b>JZ</b>	Jump if ZF=1
<b>JNZ</b>	Jump if ZF=0
<b>JO</b>	Jump if OF=1
<b>JNO</b>	Jump if OF=0
<b>JS</b>	Jump if SF=1
<b>JNS</b>	Jump if SF=0
<b>JC</b>	Jump if CF=1
<b>JNC</b>	Jump if CF=0
<b>JP</b>	Jump if PF=1
<b>JNP</b>	Jump if PF=0

# Example

unsigned integer:

```
if (eax == ebx)
    esi = 0
```

```
sub eax, ebx
jnz next
```

```
mov esi, 0
```

```
next:
```



K. N. Toosi  
University of Technology

JZ	Jump if ZF=1
JNZ	Jump if ZF=0
JO	Jump if OF=1
JNO	Jump if OF=0
JS	Jump if SF=1
JNS	Jump if SF=0
JC	Jump if CF=1
JNC	Jump if CF=0
JP	Jump if PF=1
JNP	Jump if PF=0



# Example

signed integer:

```
if (eax == ebx)
    esi = 0
```

```
sub eax, ebx
jnz next
```

```
mov esi, 0
```

```
next:
```



JZ	Jump if ZF=1
JNZ	Jump if ZF=0
JO	Jump if OF=1
JNO	Jump if OF=0
JS	Jump if SF=1
JNS	Jump if SF=0
JC	Jump if CF=1
JNC	Jump if CF=0
JP	Jump if PF=1
JNP	Jump if PF=0

# Example

signed integer:

```
if (eax == - ebx)
    edi = 4
```



K. N. Toosi  
University of Technology

JZ	Jump if ZF=1
JNZ	Jump if ZF=0
JO	Jump if OF=1
JNO	Jump if OF=0
JS	Jump if SF=1
JNS	Jump if SF=0
JC	Jump if CF=1
JNC	Jump if CF=0
JP	Jump if PF=1
JNP	Jump if PF=0

# Example

signed integer:

```
if (eax == - ebx)
    edi = 4
```

```
add eax, ebx
jnz next
```

```
mov edi, 4
```

```
next:
```



JZ	Jump if ZF=1
JNZ	Jump if ZF=0
JO	Jump if OF=1
JNO	Jump if OF=0
JS	Jump if SF=1
JNS	Jump if SF=0
JC	Jump if CF=1
JNC	Jump if CF=0
JP	Jump if PF=1
JNP	Jump if PF=0

# Example

unsigned integer:

```
if (eax >= ebx)
    esp -= 4
```



JZ	Jump if ZF=1
JNZ	Jump if ZF=0
JO	Jump if OF=1
JNO	Jump if OF=0
JS	Jump if SF=1
JNS	Jump if SF=0
JC	Jump if CF=1
JNC	Jump if CF=0
JP	Jump if PF=1
JNP	Jump if PF=0

# Example

unsigned integer:

```
if (eax >= ebx)
```

```
    esp -= 4
```

```
    sub eax, ebx
```

```
    jc next
```

```
    sub esp, 4
```

```
next:
```



K. N. Toosi  
University of Technology

JZ	Jump if ZF=1
JNZ	Jump if ZF=0
JO	Jump if OF=1
JNO	Jump if OF=0
JS	Jump if SF=1
JNS	Jump if SF=0
JC	Jump if CF=1
JNC	Jump if CF=0
JP	Jump if PF=1
JNP	Jump if PF=0

# Example



K. N. Toosi  
University of Technology

signed integer:

$x - y$

```
if (eax < ebx)
    ebp += 8
```

$x < y \Rightarrow SF = 1$

$x \geq y \Rightarrow SF = 0$

# Example



K. N. Toosi  
University of Technology

signed integer:

$x - y$

```
if (eax < ebx)
```

```
    ebp += 8
```

OF=0

$x < y \Rightarrow SF = 1$

$x \geq y \Rightarrow SF = 0$

# Example



K. N. Toosi  
University of Technology

signed integer:

$x - y$

```
if (eax < ebx)
```

```
    ebp += 8
```

<b>OF=0</b>	$x < y \Rightarrow SF = 1$ $x \geq y \Rightarrow SF = 0$
<b>OF=1</b>	$x < 0 < y \Rightarrow SF = 0$ $x > 0 > y \Rightarrow SF = 1$



# Example



K. N. Toosi  
University of Technology

signed integer:

```
if (eax < ebx) ebp += 8
```

```
sub eax, ebx
```

```
jo overflow
```

```
jns endl
```

```
if_cond:
```

```
add ebp, 8
```

```
jmp endl
```

```
overflow:
```

```
jns if_cond
```

```
endl:
```

$x - y$

OF=0	$x < y \Rightarrow SF = 1$ $x \geq y \Rightarrow SF = 0$
OF=1	$x < 0 < y \Rightarrow SF = 0$ $x > 0 > y \Rightarrow SF = 1$

# Example



K. N. Toosi  
University of Technology

signed integer:

```
if (eax < ebx)
    ebp += 8
else
    ebp -= 8
```

$x - y$

<b>OF=0</b>	$x < y \Rightarrow SF = 1$ $x \geq y \Rightarrow SF = 0$
<b>OF=1</b>	$x < 0 < y \Rightarrow SF = 0$ $x > 0 > y \Rightarrow SF = 1$

# Other conditional jump commands



sub x, y

		unsigned			signed
JE	label	jump if x == y (same as JZ)	JE	label	jump if x == y (same as JZ)
JNE	label	jump if x != y (same as JNZ)	JNE	label	jump if x != y (same as JNZ)
JA	label	jump if x > y	JG	label	jump if x > y
JNBE	label		JNLE	label	
JB	label	jump if x < y	JL	label	jump if x < y
JNAE	label		JNGE	label	
JAE	label	jump if x >= y	JGE	label	jump if x >= y
JNB	label		JNL	label	
JBE	label	jump if x <= y	JLE	label	jump if x <= y
JNA	label		JNG	label	

# Example:



```
    call read_int
    mov ebx, eax

    call read_int

l1:   sub ebx, eax
      jnc l1

      add eax, ebx

      call print_int
      call print_nl
```

# Example:



```
call read_int  
mov ebx, eax  
  
call read_int
```

l1:

```
sub ebx, eax  
jnc l1
```

```
add eax, ebx
```

```
call print_int  
call print_nl
```

rem.asm

```
call read_int  
mov ebx, eax  
  
call read_int
```

l1:

```
sub ebx, eax  
jae l1
```

```
add eax, ebx
```

```
call print_int  
call print_nl
```

rem2.asm

# Example:



K. N. Toosi  
University of Technology

Practice: Also print quotient

```
    call read_int
    mov ebx, eax

    call read_int

l1:  sub ebx, eax
     jnc l1

     add eax, ebx

     call print_int
     call print_nl
```

rem.asm

# Example:



```
call read_int
mov ebx, eax

call read_int

l1:
sub ebx, eax
jnc l1

add eax, ebx

call print_int
call print_nl
```

rem.asm

```
call read_int
mov ebx, eax

call read_int
mov ecx, 0

l1:
sub ebx, eax
inc ecx
jnc l1

dec ecx
add eax, ebx
call print_int
call print_nl

mov eax, ecx
call print_int
call print_nl
```

div.asm

# Example:



```
    call read_int
    mov  ecx, eax

    call read_int

    mov  ebx, 0
l1:  add  ebx, eax

    dec  ecx
    jnz  l1

    mov  eax, ebx
    call print_int
    call print_nl
```



# Example:



```
call read_int
mov ecx, eax

call read_int

mov ebx, 0
l1:
add ebx, eax

dec ecx
jnz l1

mov eax, ebx
call print_int
call print_nl
```

```
call read_int
mov ecx, eax

call read_int

mov ebx, 0
l1:
add ebx, eax

loop l1

mov eax, ebx
call print_int
call print_nl
```

# The loop commands



K. N. Toosi  
University of Technology

```
loop    lbl    ecx--; if (ecx!=0) goto lbl
loopz   lbl    ecx--; if (ecx!=0 && ZF=1) goto lbl
loopnz  lbl    ecx--; if (ecx!=0 && ZF=0) goto lbl
```

# The loop commands



K. N. Toosi  
University of Technology

```
loop      lbl      ecx--; if (ecx!=0) goto lbl
loopz     lbl      ecx--; if (ecx!=0 && ZF=1) goto lbl
loopnz    lbl      ecx--; if (ecx!=0 && ZF=0) goto lbl

loope     ≡ loopz
loopne    ≡ loopnz
```

# Example: Count up to N



```
    call read_int
    mov ebx, eax

    mov eax, 1

l1:  call print_int
     call print_nl

     inc eax

     mov ecx, ebx
     sub ecx, eax
     jnc l1
```

# Example: Count up to N



```
    call read_int
    mov ebx, eax

    mov eax, 1

l1:  call print_int
     call print_nl

     inc eax

     mov ecx, ebx
     sub ecx, eax
     jnc l1
```

```
    call read_int
    mov ebx, eax

    mov eax, 1

l1:  call print_int
     call print_nl

     inc eax

     mov ecx, ebx
     sub ecx, eax
     jae l1
```

# Example: Count up to N



```
    call read_int
    mov ebx, eax

    mov eax, 1

l1:  call print_int
    call print_nl

    inc eax

    mov ecx, ebx
    sub ecx, eax
    jnc l1
```

```
    call read_int
    mov ebx, eax

    mov eax, 1

l1:  call print_int
    call print_nl

    inc eax

    mov ecx, ebx
    sub ecx, eax
    jae l1
```

```
    call read_int
    mov ebx, eax

    mov eax, 1

l1:  call print_int
    call print_nl

    inc eax

    mov ecx, ebx
    sub ecx, eax
    jge l1
```

# using sub before jump; what's wrong?



K. N. Toosi  
University of Technology

```
    call read_int
    mov ebx, eax

    mov eax, 1
l1:  call print_int
     call print_nl

     inc eax

     mov ecx, ebx
     sub ecx, eax
     jae l1
```

# the cmp instruction



```
    call read_int
    mov ebx, eax

    mov eax, 1
l1:  call print_int
     call print_nl

     inc eax

     mov ecx, ebx
     sub ecx, eax
     jae l1
```

```
    call read_int
    mov ebx, eax

    mov eax, 1
l1:  call print_int
     call print_nl

     inc eax

     cmp ebx, eax
     jae l1
```



# The `cmp` instruction



K. N. Toosi  
University of Technology

```
sub eax, ebx
```

```
cmp eax, ebx
```

- `cmp x, y`
- subtracts `y` from `x` (like `sub x, y`)
- does not store the result (`x` is not changed)
- flags are set (as though a subtraction has taken place)

# The cmp instruction



cmp x, y

		unsigned			signed
JE	label	jump if x == y	JE	label	jump if x == y
JNE	label	jump if x != y	JNE	label	jump if x != y
JA	label	jump if x > y	JG	label	jump if x > y
JNBE	label		JNLE	label	
JB	label	jump if x < y	JL	label	jump if x < y
JNAE	label		JNGE	label	
JAE	label	jump if x >= y	JGE	label	jump if x >= y
JNB	label		JNL	label	
JBE	label	jump if x <= y	JLE	label	jump if x <= y
JNA	label		JNG	label	

# Practice



**K. N. Toosi**  
University of Technology

```
if (eax > ebx) {edi=1} else {edi=2}    (signed)
```

# Practice



K. N. Toosi  
University of Technology

if (eax > ebx) {edi=1} else {edi=2} (signed)

```
cmp eax, ebx
jle else_lbl
mov edi, 1
jmp endif
```

```
else_lbl:
    mov edi, 2
```

```
endif:
```