

# Arhitectura sistemelor de calcul

## Comparații între operații

### Instrucțiunea CMP

Aceasta efectuează o scădere între operații dar nu salvează rezultatul, în schimb setează flagurile.

**cmp** **eax, ebx** ( $eax - ebx$ )

caz 1:  $EAX = EBX$ ,  $ZF = 1$

caz 2:  $EAX < EBX$ ,  $CF = 1$

caz 3:  $EAX > EBX$ ,  $CF = 0$

## Salturi condiționate

Salturile permit să executăm instrucțiuni în funcție de rezultatul unei comparații.

**JE** (Jump if equal, dacă  $ZF = 1$ )

**JNE** (Jump if not equal, dacă  $ZF = 0$ )

**JG** (Jump if greater, dacă  $ZF = 0$  și  $SF = OF$ )

**JL** (Jump if less, dacă  $SF \neq OF$ )

**JGE** (Jump if greater or equal, dacă  $SF = OF$ )

**JLE** (Jump if less or equal, dacă  $SF \neq OF$  sau  $ZF = 1$ )

Relația între operanzi ce se dorește a fi testată	Comparație cu semn	Comparație fără semn
$d = s$	JE	JE
$d \neq s$	JNE	JNE
$d > s$	JG	JA
$d < s$	JL	JB
$d \geq s$	JGE	JAE
$d \leq s$	JLE	JBE

Ex:

```
MOV AX, 2
```

```
MOV BX, 5
```

```
CMP AX, BX ; 2 - 5 = -3
```

```
JL MAIMIC
```

```
MAIMIC:
```

## Instrucțiuni de ciclare

Fiecare instrucțiune se bazează pe registrul CX/ECX care scade automat la fiecare iterație.

Sintaxa:

[instrucțiune] etichetă

**LOOP**, scade ECX cu 1 până când  $ECX = 0$ , funcționează dacă  $ECX \neq 0$

**LOOPE**, scade ECX cu 1 până când  $ECX = 0$ , funcționează dacă  $ECX \neq 0, ZF = 1$

**LOOPNZ**, scade ECX cu 1 până când  $ECX = 0$ , funcționează dacă  $ECX \neq 0, ZF = 0$

**JECXZ**, scade ECX cu 1 până când  $ECX = 0$  funcționează dacă  $ECX \neq 0$

și sare la o etichetă specifică doar dacă  $ECX = 0$

Ex:

```
mov ECX, 5
```

```
start_loop:
```

```
; codul
```

```
loop start_loop
```

```
MOV ECX, 0      ; Setează ECX la 0  
JECXZ eticheta  ; Săritura se va face deoarece ECX este 0  
; Codul buclei  
eticheta:  
; Continuarea codului
```