



# **Microprocesorul Intel8086**

## **Setul de instructiuni I8086**

### **Curs 8**

Arhitectura sistemelor de calcul

Lect. Dr. Ozten CHELAI

Facultatea de Matematica si Informatica  
Universitatea Ovidius Constanta

# Instrucțiuni pentru transferuri de date



## b) Instrucțiuni specifice cu acumulatorul

- instrucțiuni de transfer de la/la porturile de intrare/ieșire.
  - Un operand se află întotdeauna în registrul acumulator, iar celălalt reprezintă portul. Dacă portul este pe 8 biți se folosește registrul **AL**, iar dacă portul este pe 16 biți se folosește registrul **AX**;
  - Adresa portului:
    - Dacă adresa portului este între 0 și 255 se specifică direct în instrucțiune.
    - Dacă adresa >255, se încarcă registrul DX cu adresa respectivă, iar instrucțiunea are sintaxa:
  - Instrucțiuni de citire (intrare)
    - in ac, port** ; citește un caracter în registrul acumulator de la portul specificat.
    - in al, dx** ; pentru un port de 8 biți
    - in ax, dx** ; pentru un port de 16 biți
  - Instrucțiuni de scriere (ieșire):
    - out port, ac** ; scrie un caracter din registrul acumulator în portul specificat.
    - out dx,al** ; pentru un port de 8 biți
    - out dx,ax** ; pentru un port de 16 biți
- Instrucțiunea **xlat**. Această instrucțiune realizează o translație folosind registrul **bx** ca pointer al adresei de bază a unui tabel de 256 octeți, **al** ca index în acest tabel, iar valoarea pointată din tabel este întoarsă în registrul **al**.

## c) Instrucțiuni cu obiect adresă.

- instrucțiuni pentru încărcarea regiștrilor segment.
  - lds reg, mem32** – încarcă registrul **reg** cu cuvântul de la adresa de memorie **mem32**, iar registrul segment DS cu valoarea de la locația **mem32+2**
  - les reg, mem32** - încarcă registrul segment ES
- Aceste instrucțiuni sunt folosite pentru setarea pointerilor de tip far.
- instrucțiuni pentru încărcarea adresei efective
  - lea reg, mem** - care încarcă în registrul specificat adresa efectivă de memorie **mem**.

## d) Transferuri de flaguri

**lahf** – încarcă registrul **ah** cu conținutul octetului mai puțin semnificativ al registrului de stare;  
**sahf** - încarcă octetul mai puțin semnificativ al registrului de stare cu conținutul registrului **ah**;  
**pushf** - salvează în stivă registrul de stare  
**popf** - încarcă registrul de stare cu valoarea din vârful stivei.

# Instrucțiuni aritmetice



- Instrucțiunile aritmetice lucrează cu patru tipuri de date: binare fără semn, binare cu semn, BCD neîmpachetate și BCD împachetate.
- Instrucțiunile aritmetice sunt:
  - instrucțiuni pentru adunare
  - instrucțiuni pentru scădere
  - instrucțiuni pentru înmulțire
  - instrucțiuni pentru împărțire
- Instrucțiunile aritmetice afectează indicatorii de condiție din registrul de stare.
- **a) Instrucțiuni pentru adunare:**
  - add dest, sursă** - realizează operația  $\text{dest} + \text{sursă}$  cu depunere în dest
  - adc dest, sursă** - realizează operația  $\text{dest} + \text{sursă} + \text{CF}$  cu depunere în dest, unde CF=carry flag
  - inc dest** - realizează operația  $\text{dest} + 1$  cu depunere în dest
  - aaa** – (ASCII Adjust for Addition) realizează o corecție ASCII la adunare a rezultatului stocat în registrul AL după o operație de adunare ASCII
  - daa** - (Decimal Adjust for Addition) realizează o corecție BCD la adunare a rezultatului stocat în registrul AL după o operație de adunare BCD
- **b) Instrucțiuni pentru scădere:**
  - sub dest, sursă** - realizează operația  $\text{dest} - \text{sursă}$  cu depunere în dest
  - sbb dest, sursă** - realizează operația  $\text{dest} - \text{sursă} - \text{CF}$  cu depunere în dest, unde CF=carry flag
  - dec dest** - realizează operația  $\text{dest} - 1$  cu depunere în dest
  - neg dest** - realizează operația  $0 - \text{dest}$  cu depunere în dest
  - cmp dest, sursă** - compară operanzii dest și sursă prin scădere ( $\text{dest} - \text{sursă}$ ), poziționează corespunzător indicatorii de condiție din registrul de stare fără să modifice destinația.
  - aas** – (ASCII Adjust for Subtraction) realizează o corecție ASCII la scădere a rezultatului stocat în registrul AL după o operație de scădere ASCII
  - das** - (Decimal Adjust for Subtraction) realizează o corecție BCD la scădere a rezultatului stocat în registrul AL după o operație de scădere BCD.

# Instrucțiuni aritmetice



- **c) Instrucțiuni pentru înmulțire**

În general un operand se află în registrul acumulator (AX sau AL), iar rezultatul se încarcă tot în registrul acumulator (AX) și în registrul DX partea mai semnificativă dacă operandii sunt de doi octeți.

**mul dest, sursă** - realizează operația  $\text{dest} * \text{sursa}$  fără semn cu depunere în dest. Dest este AL sau AX

**imul dest, sursă** - realizează operația  $\text{dest} * \text{sursa}$  cu semn cu depunere în dest. Dest este AL sau AX

**aam** – (ASCII Adjust for Multiplication) realizează o corecție ASCII la înmulțire a rezultatului stocat în registrul A după o operație de înmulțire ASCII

- **d) Instrucțiuni pentru împărțire**

Deîmpărțitul se află se află în registrul acumulator AX sau în AX și DX iar rezultatul se încarcă în registrul acumulator (AL, AX). Restul împărțirii se întoarce în AH sau DX

**div dest, sursă** - realizează operația  $\text{dest} / \text{sursa}$  fără semn cu depunere în dest.

**idiv dest, sursă** - realizează operația  $\text{dest} / \text{sursa}$  cu semn cu depunere în dest.

**aad** – (ASCII Adjust for Division) realizează o corecție ASCII la împărțire a rezultatului stocat în registrul A după o operație de împărțire ASCII

**cbw** – (Convert Byte to Word) – permite o extensie de semn a lui AL în AH

**cwd** – (Convert Word to DoubleWord) – permite o extensie de semn a lui AX în DX