

# Eilutinės komandos (1)

Eilutinės komandos skirtos operacijoms su baitų ar žodžių eilutėmis.

Eilutinės komandos dirba su eilutėmis, esančiomis DS arba ES segmentuose (nebent yra keitimo prefiksas) ir/arba akumuliatoriumi (AX/AL registru).

Eilutinės komandos gali padėti greitai pakeisti tam tikrus atminties blokus norimomis reikšmėmis.

Baito ilgio eilutė.  
Dabar joje yra



Žodžio ilgio eilutė.  
Dabar joje yra



Atminties gabalas

12345

11
12
45
89
BA
CC
A5
57
7E
AC
DC
AA
AA
AA
AA

12353

# Eilutinės komandos (2)

Visos eilutinės komandos mašininiame kode užima 1 baitą.

Paskutinis eilutinės komandos mašininio kodo bitas yra **w (word)** bitas – jis nurodo, ar bus dirbama su baitais (**w=0**), ar bus dirbama su žodžiais (**w=1**).

Ar eilutinė komanda dirba su baitu, ar su žodžiu, galima nuspręsti ir iš komandos pavadinimo: jei paskutinė raidė B, dirbama su baitais, jei W – su žodžiais.

---

*Pvz.: MOVSB – gale B, reiškia bus dirbama su baitais.*

*Pvz.: Eilutinės komandos operacijos kodas: 1010 0101 – w=1, reiškia bus dirbama su žodžiais.*

# Eilutinių komandų klasifikacija

Eilutinės komandos gali būti **palyginimo** arba **duomenų persiuntimo**.

## **Palyginimo komandos:**

- 1) Palygina dviejų baitų/žodžių reikšmes
- 2) Atitinkamai pakeičia SF registrą
- 3) Koreguoja SI ir/ar DI registrus

## **Duomenų persiuntimo komandos:**

- 4) Keičia atminties baitų arba AX/AL reikšmę
- 5) Koreguoja SI ir/ar DI registrus

# Palyginimo komandos

SCASB – Scan string byte

SCASW – Scan string word

CMPSB – Compare string byte

CMPSW – Compare string word

---

*Pvz.: atliekama komanda SCASB. Kas vyksta:*

- 1) Paimama AL reikšmė*
- 2) Paimama **baito**, esančio absoliučiu adresu  $ES*10h + DI$ , reikšmė*
- 3) Atliekamas CMP (AL – baitas)*
- 4) Pakeičiami atitinkami SF registro flag'ai*

Komandos pavadinimas	Atliekamas veiksmas
SCASB	cmp al, es:[di]
SCASW	cmp ax, es:[di]
CMPSB	cmp ds:[si], es:[di]
CMPSW	

# Duomenų persiuntimo komandos

MOVSB – Move string byte

MOVSW – Move string word

LODSB – Load string byte

LODSW – Load string word

STOSB – Store string byte

~~STOSW – Store string word~~

*Pvz.: atliekama komanda **MOVSW**. Kas vyksta:*

- 1) Paimama **žodžio**, esančio absoliučiu adresu  $DS*10h + SI$ , reikšmė*
- 2) Paimta reikšmė įrašoma į atminties vietą, kurios absoliutus adresas yra  $ES*10h + DI$*
- 3) Priklausomai nuo DF (Direction Flag), didinamos arba mažinamos SI ir DI registrų reikšmės*

Komandos pavadinimas	Atliekamas veiksmas
MOVSB	mov es:[di], ds:[si]
MOVSW	
LODSB	mov al, ds:[si]
LODSW	mov ax, ds:[si]
STOSB	mov es:[di], al
STOSW	mov es:[di], ax

# SI ir DI eilutinėse komandose

Po kiekvienos eilutinės komandos yra keičiamos SI ir/arba DI reikšmės. Tačiau, jei kažkuris iš registrų komandoje nebuvo panaudotas, jo reikšmė išlieka nepakitusi.

SI ir DI keičiamos pagal šias taisykles:

- $DF = 1 \rightarrow$  reikšmė yra mažinama
  - $DF = 0 \rightarrow$  reikšmė yra didinama
  - Komanda dirba su žodžiais  $\rightarrow$  reikšmė keičiama per 2
  - Komanda dirba su baitais  $\rightarrow$  reikšmė keičiama per 1
- 

*Pvz.: Vykdoma komanda SCASW, o  $DF=1$ . Kaip pasikeis SI ir DI?*

- 1) SCASW  $\rightarrow$  naudojamas tik DI registras, SI nesikeis.
- 2) SCAS<sup>W</sup>  $\rightarrow$  dirbama su žodžiais, reikšmė keisis per 2.
- 3)  $DF = 1 \rightarrow$  reikšmė bus mažinama.

*Ats.: SI nesikeis,  $DI:=DI-2$ .*

# Pakartojimo prefiksai

Pakartojimo prefiksai leidžia pakartoti eilutinę komandą.

Visi pakartojimo prefiksai užima vieną baitą mašiniame kode. Paskutinis baito bitas yra **z** (**zero**) bitas.

Palyginimo komandos su pakartojimo prefiksu yra kartojamos tol, kol  $CX \neq 0$  ir  $ZF = z$ .

Duomenų persiuntimo komandos su pakartojimo prefiksu yra kartojamos tol, kol  $CX \neq 0$ .

Prefiksas	Prefikso mašininis kodas	z bito reikšmė
REP REPE REPZ	F3	1
REPNE REPNZ	F2	0

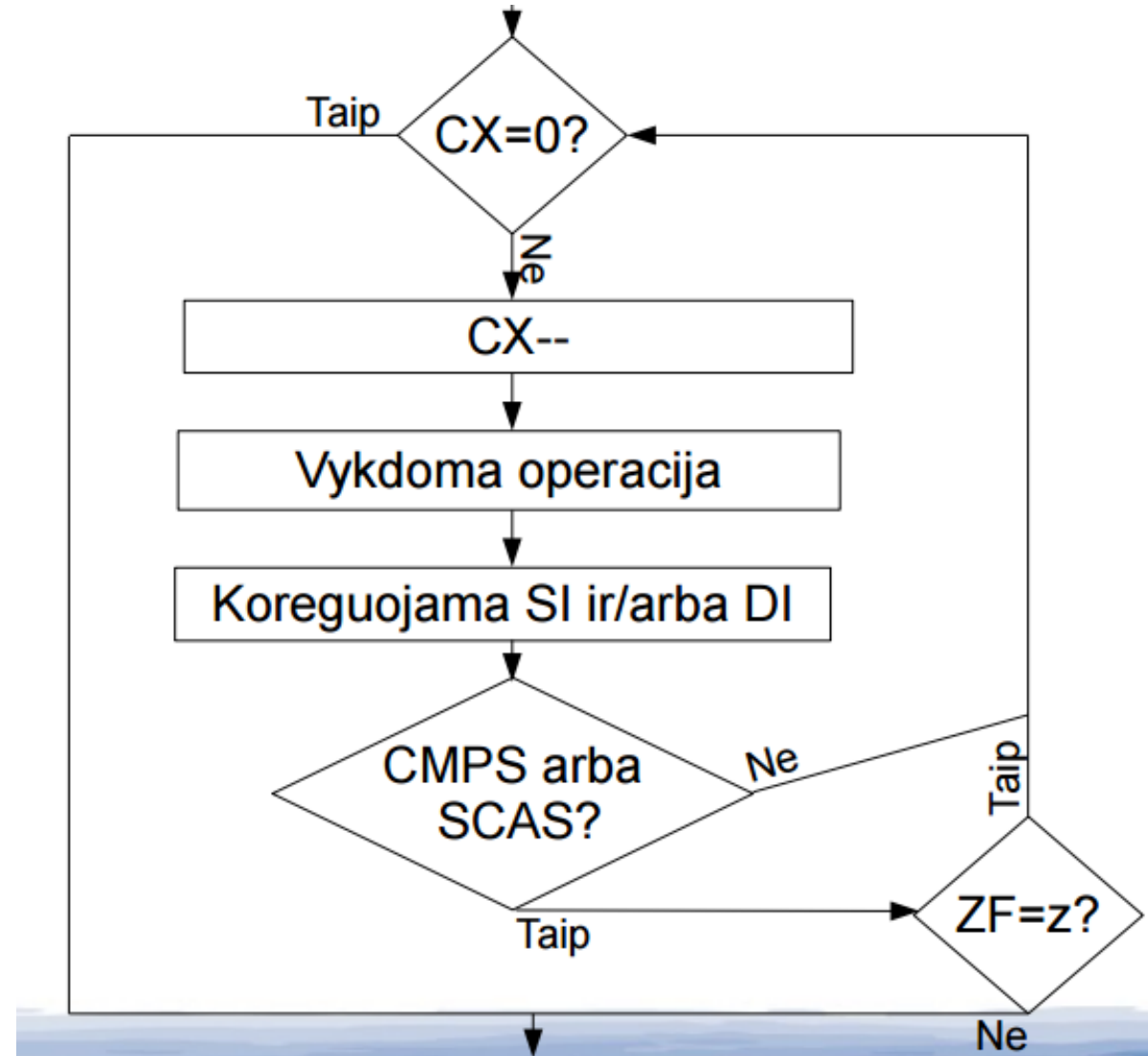
# Eilutinių komandų vykdymo schema

Pvz.: CX=0005, SF=0000, SI = 1234, DI = ABCD, visų ES ir DS baitų reikšmės yra nuliai. Vykdoma komanda REP NZ CMPSW. Kokios bus SI, DI ir CX reikšmės įvykdžius komandą?

- 1) Ar CX = 0? Ne.
- 2) CX:=0005-1=0004;
- 3) Vykdom operaciją. Kadangi visi ES ir DS baitai užpildyti nuliais, bus atliekamas toks veiksmas: CMP 0000h, 0000h. Reiškia, ZF tampa 1 (nes atėmę gavom nulį, reiškia ZF=1).
- 4) Komanda SCASW → dirbam su **žodžiais**, dalyvauja ir SI, ir DI registrai → DF=0, reikšmės didės → SI:=1234+2=1236, DI:=ABCD+2=ABCF
- 5) Ar vykdoma komanda yra CMPS arba SCAS? Taip.
- 6) Ar ZF=z? ZF=1, z=0 → Ne.

Eilutinė komanda įvykdyta, einama vykdyti kitos komandos.

Ats.: SI=1236, DI=ABCF, CX=0004.





# Pavyzdinis uždavinys (1)

Registų reikšmės SI=0000, DI=0000, DS=1234, ES=3333, AX=BABA, CX = 0014, SF = 000F. Duomenų segmento pirmųjų 20-ies baitų reikšmės yra atitinkamai 01h, 02h, 03h, ..., 14h. Vykdoma komanda REP MOVSB. Kokia bus SI, DI ir baitų, atmintyje esančių adresais 33330...33344, reikšmių suma įvykdžius komandą?

Tikslas yra rasti sumą iš trijų skaičių:

- 1) SI registro reikšmės įvykdžius komandą
- 2) DI registro reikšmės įvykdžius komandą
- 3) Baitų, esančių nurodytais adresais, reikšmių sumos

## Pavyzdinis uždavinys (2)

Pradėkime nuo baitų, esančių nurodytais adresais. Nusibraižykime, kaip atrodo atmintis prieš vykdant komandą.

Baitai, kurių mums reikia, vis dar nežinomi, tačiau pirmojo reikalingo baito adresas sutampa su ES segmento pradžia. Kadangi komandoje MOVSB rašoma į **ES** segmentą (MOVSB atlieka veiksmą MOV **ES**:[DI], DS:[SI]), pradėkime vykdyti komandą ir žiūrėkime, kaip keisis atminties baitai.

Absolūtus adresas	Baito reikšmė atmintyje
00000	XXh
	...
<i>DS pradžia</i> 12340	01h
	02h
	...
12354	14h
	...
<i>DS pabaiga</i> 2233F	XXh
	...
<i>ES pradžia</i> 33330	XXh
33331	XXh
	...
33344	XXh
	...
<i>ES pabaiga</i> 4332F	XXh
	...
FFFFF	XXh

# Pavyzdinis uždavinys (3)

Vykdomė komandą REP MOVSB:

- 1) Ar  $CX = 0$ ?  $CX=0014$ , reiškia ne.
- 2)  $CX:=0014-1=0013$ .
- 3) Vykdom operaciją. Atliekamas veiksmas:  $MOV ES:[DI], DS:[SI]$ . Vadinasi, baid, esančio adresu 12340 ( $1234*10h+0000$ ), reikšmę įrašysime į adresą 33330 ( $3333*10h+0000$ ).
- 4) Komandoje naudojamas ir SI, ir DI → keisis ir SI, ir DI.  
Komandoje dirbama su baidais → keisis per 1.  
 $DF=0$  (iš  $SF=000F$ ) → SI ir DI didės.  
 $SI:=0000+1=0001$ ;  $DI:=0000+1=0001$
- 5) Ar komanda SCAS/CMPS? Ne.

Dabar reiktų eiti vykdyti visko iš naujo, bet pažiūrėkime, kaip atrodo atmintis po vienos iteracijos.

Kas bus po dviejų iteracijų? (Baid, esančio adresu 12341 reikšmė įrašyta į 33331 adresu esantį baidą)

Kas bus po visų iteracijų? Iš viso yra vykdoma 14h (CX reikšmė) kartų, kas yra 20d kartų. Reiškia, visos mūsų žinomos reikšmės duomenų segmente yra perkopijuojamos į pirmus 20 ekstra segmento baidų.

Absoliutus adresas	Baid reikšmė atmintyje
00000	XXh
	...
<i>DS pradžia</i> 12340	01h
	02h
	...
12354	14h
	...
<i>DS pabaiga</i> 2233F	XXh
	...
<i>ES pradžia</i> 33330	01h
33331	02h
	...
33344	14h
	...
<i>ES pabaiga</i> 4332F	XXh
	...
FFFFFF	XXh

# Pavyzdinis uždavinys (4)

Prisimenam sąlygą: mums reikia baitų, esančių adresais 33330...33344, reikšmių sumos. Įvykdę komandą, jau turime šias reikšmes. Sudedam jas (nepamirštam, kad tai šešioliktainiai skaičiai):

$$01+02+03+\dots+09+0A+0B+\dots+13+14=D2h$$

Trūksta: SI ir DI reikšmių.

Anksčiau išsiaiškinom, kad ir SI, ir DI didėja per vieną (nes DF=0, dirbam su baitais, naudojami abu registrai). Kadangi eilutinę komandą vykdom 14h kartų, reikia 14h kartų pridėti 01h:

$$SI:=0000+0014=0014h$$

$$DI:=0000+0014=0014h$$

$$\text{Sudedam viską: } D2+14+14=FAh$$

Ats.: FAh

Absoliutus adresas	Baito reikšmė atmintyje
00000	XXh
	...
<i>DS pradžia</i> 12340	01h
	02h
	...
12354	14h
	...
<i>DS pabaiga</i> 2233F	XXh
	...
<i>ES pradžia</i> 33330	01h
33331	02h
	...
33344	14h
	...
<i>ES pabaiga</i> 4332F	XXh
	...
FFFFFF	XXh

# Uždaviniai

1. Registrų SI ir DI reikšmės yra 000A, registras CX=0002, registras SF=0C00. Kokia bus registrų SI ir DI reikšmių suma, įvykdžius komandą: REP STOSW?
2. Registrų SI ir DI reikšmės yra ABCD, registras CX=0000, registras SF=0000. Kokia bus registrų SI ir DI reikšmių suma, įvykdžius komandą: REP LODSW?
3. DI=FFFF, SI=ABBA, CX=0010, SF=FFFF, AX=1234. Visas ekstra segmentas užpildytas baitais, kurių reikšmės 34h. Vykdoma komanda REPNE SCASB. Kokia bus AX ir DI registrų reikšmių suma įvykdžius komandą?
4. SF=FFFF, SI=0013, DI=0009, CX=0003.  
Pirmieji 32 DS baitų atrodo taip:  
0000: ---abcdefghijk—  
0010:---ABCDEFGHIIK—  
Pirmieji 32 ES baitų atrodo taip:  
0000: lmnoprstuv-----  
0010: -----LMNOPRSTUV  
Kaip atrodys pateikti DS ir ES fragmentai ir kokios bus CX, SI ir DI reikšmės įvykdžius REPNE MOVSW ?

# Atsakymai

1. 0010h
2. 1579h
3. 11232h
4. 0000: Imnop----AB-----  
0010: -----LMNOPRSTUV