

## 5 laboratorinis darbas

### Operandų adresavimas mikroprocesoriuje *Intel 8086 (K1810BM86)*

#### 1. Darbo tikslas

Susipažinti su mikroprocesoriaus *Intel 8086 (K1810BM86)* komandų formatais, operandų adresavimo būdais bei komandų kodų formavimu.

#### 2. Bendrosios žinios

##### 2.1. Komandų formatai

Mikroprocesoriaus *Intel 8086 (K1810BM86)* komandų sistemoje 133 bazinės komandos, naudojančios ir 8, ir 16 bitų duomenis. Komandos gali būti be operandų arba turėti 1 arba 2 operandus. Bendrasis komandos formatas pateiktas 1 paveiksle.

Kartojimo prefiksas	Segmento pakeitimo prefiksas	Operacijos kodas (COP)	Mod Reg R / M	Adresas	Duomenys
0 arba 1	0 arba 1	1	0 arba 1	0, 1, 2, 4	0, 1, 2
Ilgis baitais					

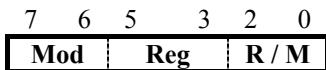
1 pav. Bendrasis komandos formatas

Kartojimo prefiksas (jo gali ir nebūti) taikomas kai kuriose ciklinėse procedūrose (pvz., simbolių eilučių apdorojimo komandose), kartojant vieną arba kitą komandą, kol bus įvykdyta iš anksto nustatyta sąlyga.

Segmento pakeitimo prefiksas taikytinas tuomet, kai vietoje segmento, kuris naudojamas pagal nutylėjimą, reikia taikyti kitą segmentą, pvz., vietoje **DS** – **CS**.

Operacijos kodo baitas (**OKB**) yra visose komandose. Jei komanda vieno baito, tai šis baitas yra **OKB**.

Tolesnis komandos kodo baitas (jo gali ir nebūti) yra vadinamas adresavimo baitu (**AB**). Jo formatas parodytas 2 paveiksle.



2 pav. Adresavimo baito formatas

Šio baito laukas **Mod** nustato komandos taikomą adresavimo režimą. Nuo jo turinio priklauso, kaip bus interpretuojamas laukas **R/M**, kuris gali reikšti arba registrą, arba atminties adresą. Laukas **Reg** nurodo registrą, kaip vieną iš komandos operandų. Šis laukas gali būti taikomas ir operacijos kodo išplėtimui.

Jeigu **Mod=11**, komandos operandai yra procesoriaus registruose, kuriuos nurodo laukai **Reg**, **R/M** ir žemiausias **OKB** bitas **W** (žr. 1 lentelę).

1 lentelė. MP registrų adresai

Laukas Reg arba R/M	Registrai		Segmentų registrai
	W=0	W=1	
000	AL	AX	ES
001	CL	CX	CS
010	DL	DX	SS
011	BL	BX	DS
100	AH	SP	-
101	CH	BP	-
110	DH	SI	-
111	BH	DI	-

Segmentų registrai adresuojami tik dviem bitais, t. y. 00, 01, 10 ir 11.

Kitais atvejais laukus **Mod**, **R/M** ir komandos kodą lemia adresavimo režimas (žr. 2 lentelę).

2 lentelė. Operandų adresavimo režimai

R/M	Adresavimo režimas		
	Mod=00	Mod=01	Mod=10
000	BX+SI	BX+SI+Disp L	BX+SI+ Disp H Disp L
001	BX+DI	BX+DI+Disp L	BX+DI+ Disp H Disp L
010	BP+SI	BP+SI+Disp L	BP+SI+ Disp H Disp L
011	BP+DI	BP+DI+Disp L	BP+DI+ Disp H Disp L
100	SI	SI+Disp L	SI+ Disp H Disp L
101	DI	DI+Disp L	DI+ Disp H Disp L
110	Disp H DipL	BP+Disp L	BP+Disp H Disp L
111	BX	BX+Disp L	BX+Disp H Disp L

Šioje lentelėje **Disp L** yra poslinkio segmento pradžios atžvilgiu žemesnysis baitas, **Disp H** – šio poslinkio aukštesnysis baitas.

Adresavimo baitas nereikalingas, jeigu:

- ♦ komanda neturi operandų (pvz., **PUSH, RET**);
- ♦ pirmasis komandos operandas – registras, antrasis – komandos kodo dalis (tiesioginis operandas);
- ♦ vienas operandas – atminties adresas, esant tiesioginiam adresavimui; antras – registras **AX (AL)**;
- ♦ pereigos komandose taikomas tiesioginis adresavimas;
- ♦ naudojamos įvesties / išvesties komandos (**IN, OUT**).

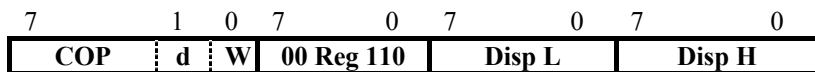
Komandos kodo adreso laukas gali būti iki keturių baitų ilgio. Čia įrašomas poslinkis **Disp L**, **Disp H** ir bazinis segmento adresas **Seg L**, **Seg H**.

Paskutiniame komandos kodo lauke įrašomi vieno arba dviejų baitų ilgio tiesioginiai duomenys.

Komandos kodo ilgis baitais lygus atskirų laukų ilgių sumai ir paprastai būna 1–6 baitų.

## 2.2. Tiesioginis adresavimas

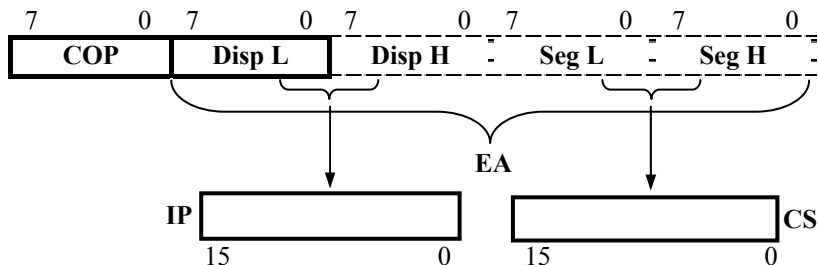
Tai adresavimas, kai adresas nurodomas pačiame komandos kode. Duomenų apdorojimo komandose tiesioginis adresas nurodomas po adresavimo baito einančiu poslinkiu (3 pav.).



3 pav. Komandos su tiesioginiu adresavimu formatas

Bitas "**d**" rodo duomenų perdavimo kryptį. Jeigu **d**=1, operandas perduodamas į registrą, nustatomą lauke **Reg**, t. y. kai **d**=1 – registras yra imtuvas, kai **d**=0 – registras yra siųstuvas.

Tiesioginis adresas besąlyginės tarpsegmentinės pereigos ir kreipimosi į paprogramius komandose formuojamas iš 16 bitų poslinkio **Disp** ir 16 bitų segmento adreso **Seg** (4 pav.).



4 pav. Besąlyginės pereigos ir kreipimosi į paprogramius komandų formatas, kai adresavimas tiesioginis

Komandos vykdymo metu poslinkis perduodamas į komandų skaitiklį **IP**, o segmento adresas – į registrą **CS**. Šiuo atveju komandos ilgis yra 5 baitai. Jeigu pereiga vykdoma segmento viduje, baitai **Seg L** ir **Seg H** netaikomi, o esant artimai pereigai tampa nereikalingas ir baitas **Disp H**. Tokiu atveju efektyvusis pereigos adresas apskaičiuojamas pagal formulę

$$EA = FA - (IP) + 1,$$

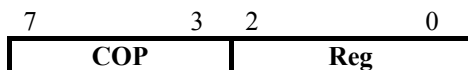
čia **FA** – fizinis adresas, prie kurio reikia pereiti; **(IP)** – komandų skaitiklyje nurodytas einamosios komandos paskutiniojo baito adresas.

### 2.3. Registrinis adresavimas

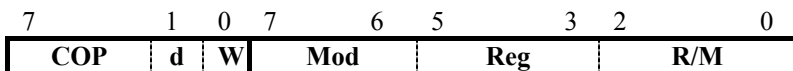
Komandose su registriniu adresavimu operandas yra viename iš bendrosios paskirties registru. Kadangi registrams adresuoti pakanka 2–3 komandos kodo skilčių, tai tokių komandų kodai yra neilgi (1–2 baitai). Be to, jiems vykdyti reikalingas minimalus kompiuterio laikas, nes operandai yra procesoriaus viduje.

Galimi tokie komandų formatai:

1) registro adresas nurodomas operacijos kodo baite;



2) registro (registru) adresas nurodomas adresavimo baite;

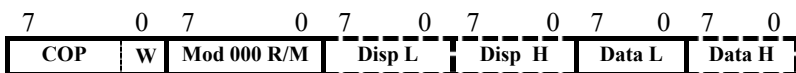


Kai **Mod**=11, abu operandai yra registruose, kuriuos nurodo laukai **Reg** ir **R/M**. Bitų **d** ir **W** paskirtis tokia pati kaip ir tiesioginio adresavimo atveju, t. y. kai **d**=1, registras, nurodytas lauke **Reg**, yra informacijos imtuvas, kai **W**=1, operacijos vykdomos su 16 bitų operandais.

### 2.4. Tiesioginis operandas

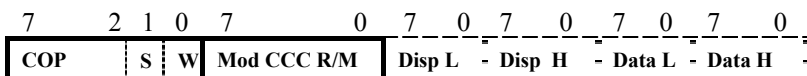
Tiesioginis operandas **Data** užima 1 arba 2 baitus komandos kodo pabaigoje. Žemesnysis baitas **Data L** visuomet nurodomas kaip pirmasis. Vieno baito tiesioginis operandas, jeigu reikia įvertinti ženklą, užrašomas papildomuoju kodu ir gali reikšti skaičius nuo –128 iki +127.

Vienos iš galimų komandų su tiesioginiu operandu formatų yra toks:



Punktyru pažymėti nebūtinai komandos kodo baitai. Pvz., jeigu **W=0** tuomet **DataH=0**, ir turėsime 5 baitų ilgio komandą. Taikant šį formatą galima sudaryti kodą komandos, kuri tiesioginį operandą persiųs į atmintį arba atliks operaciją su kitu operandu, kurio adresas nurodomas adresavimo baite.

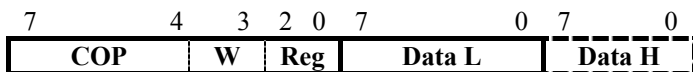
Aritmetinių komandų su tiesioginiu operandu formatas irgi gali būti iki 6 baitų ilgio.



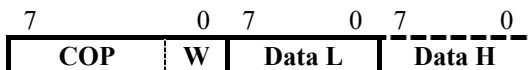
Pagal trijų skilčių lauką **CCC** kartu su lauku **COP** identifikuojamos aritmetinės operacijos. Jeigu **SW=01**, tiesioginis operandas yra 2 baitų. Kai **SW=11**, tiesioginis operandas yra 1 baito, užrašomas papildomuoju kodu ir gali reikšti skaičius nuo -128 iki +127.

Taikomos ir trumpo formato (1–3 baitai) specializuotos komandos su tiesioginiu operandu:

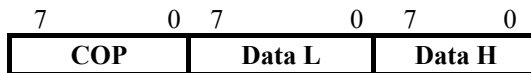
1) tiesioginiam operandui persiųsti į registrą:



2) aritmetinėms ir loginėms operacijoms atlikti, jeigu vienas iš operandų yra kaupiklyje;



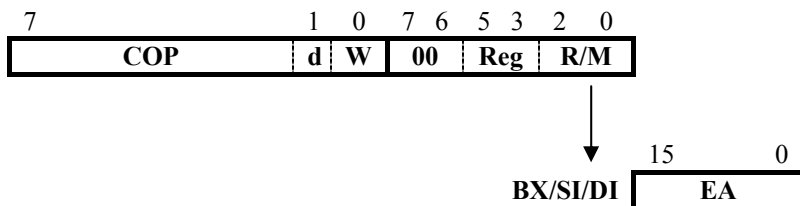
3) grįžimo iš paprogramių komandose, kai prie dėklo rodyklės reikia pridėti konstantą;



Pastarosioms trims komandoms būdingas vykdymo spartumas, nes nereikia kreiptis į atmintį.

## 2.5. Šalutinis registrinis adresavimas

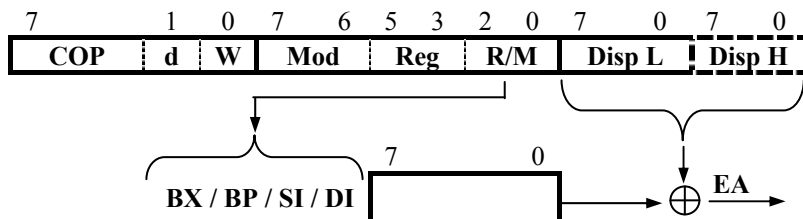
Komandose su šalutiniu registriniu adresavimu 16 bitų vykdymo (efektyvusis) adresas **EA** yra viename iš registrų, kurį nustato adresavimo baido laukas **R/M**. Šiam tikslui gali būti panaudotas vienas iš registrų **BX, SI** arba **DI**



Toks adresavimo būdas naudojamas besąlyginių perėjimų ir kreipimosi į paprogrames komandose.

## 2.6. Bazinis ir indeksinis adresavimas

Esant baziniam adresavimui vykdymo adresas EA gaunamas sudedant registrų **BX** arba **BP** turinius su 8 arba 16 skilčių poslinkiu.



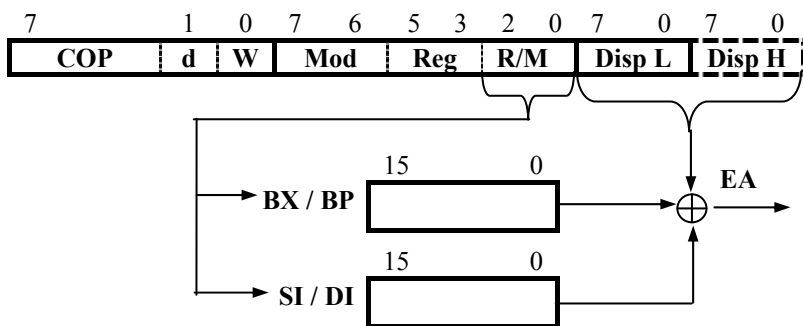
Jeigu poslinkis yra 8 skilčių, tai jis pateikiamas papildomuoju kodu iš skaičių intervalo nuo –128 iki +127.

Kai adresavimas bazinis, galima dirbti su duomenimis, esančiais įvairiose atminties vietose.

Indeksinis adresavimas skiriasi nuo bazinio tik tuo, kad vietoje bazinio registro **BX** ir bazės rodiklio **BP** panaudojami indeksų registrai **SI** ir **DI**. Toks adresavimo būdas patogus apdorojant duomenų masyvus, kai poslinkis rodo bazinį masyvo adresą, o indeksų registro turinys – masyvo elemento indeksą.

## 2.7. Bazinis-indeksinis adresavimas

Šiuo atveju vykdymo adresas **EA** formuojamas sumuojant bazinio ir indeksų registrų turinius su postūmiu:



8 skilčių postūmis pateikiamas papildomuoju kodu iš skaičių diapazono –128 ... +127.

Šio tipo adresavimas naudojamas dirbant su sudėtingos struktūros duomenimis, nes leidžia vienu metu keisti dvi adreso komponentes.

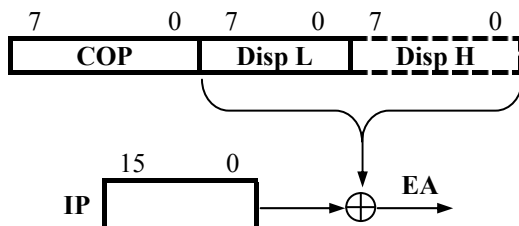
## 2.8. Santykinis adresavimas

Esant šiam adresavimui vykdymo adresas **EA** apskaičiuojamas



sumuojant komandų skaitiklio **IP** einamąjį turinį (pirmojo kitos komandos baito adresas) su postūmiu, nurodytu komandoje.

8 arba 16 skilčių postūmis imamas iš skaičių su ženklų diapazonuose  $-128 \dots +127$  arba  $-32768 \dots +32767$ :



Santykinis adresavimas taikomas tik pereigų, kreipinių į paprogramius ir ciklų valdymo komandose.

## 2.9. Simbolių eilučių apdorojimo komandoms taikomas adresavimas

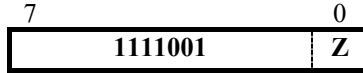
Simbolių eilučių elementais gali būti baitai (**W=0**) arba žodžiai (**W=1**). Pirminės simbolių eilutės elementai adresuojami taikant registrą **SI**, o patys elementai pagal nutylėjimą talpinami duomenų segmente **DS**. Apdorota simbolių eilutė visuomet talpinama papildomame duomenų segmente **ES**, o jos elementams adresuoti taikomas registras **DI**.

Jei prieš eilutės elementų apdorojimo komandą įterpiamas vieno baito kartojimo prefiksas **REP**, tai atitinkama operacija (primityvas) bus kartojama tol, kol ciklų skaitiklio **CX** turinys taps lygus 0. Vykdamas eilutės apdorojimo komandą automatiškai koreguojamas registrų **SI** ir **DI** turinys, atsižvelgiant į požymio registro **FL** 11-os skilties reikšmę **DF**. Jei **DF=0**, po kiekvienos operacijos atitinkamų indeksavimo registrų turinys padidinamas vienetu, kai veiksmai atliekami su baitais, ir dviem – jeigu operuojama žodžiais. Jei **DF=1**, indeksų registrų turiniai mažinami vienetu arba dviem.

Eilučių apdorojimo komandos yra 1 baito.



Kartojimo prefiksas taip pat yra vieno baid.

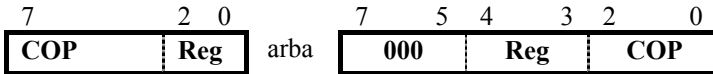


Jeigu požymio registro požymis **ZF** (6 bitas) nesutampa su bitu **Z**, kartotinė eilutės apdorojimo operacija nutraukiama.

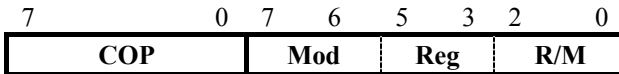
## 2.10. Dėklinis adresavimas

Dėklinis adresavimas taikomas komandose **PUSH** ir **POP** (rašyti į dėklą ir skaityti iš dėklo). Operando adresas šiuo atveju talpinamas dėklo rodyklėje **SP** ir automatiškai mažinamas arba didinamas 2 vienetais įrašant į dėklą arba skaitant iš jo. Dėklas užpildomas adresų mažėjimo kryptimi.

Dėklas gali keistis duomenimis su bendros paskirties registrais ir segmentų registrais. Tokios komandos yra vieno baid.



Komandos, vykdančios informacijos keitimasi informacija tarp dėklo ir atminties, yra 2 baidų.



3 skilčių laukas **Reg** kartu su lauku **COP** identifikuoja komandą.

## 3. Užduotis

1. Pagal toliau pateiktą formą nusibraižyti lentelę, į kurią bus rašomi mikroprocesoriaus *Intel 8086* (K1810BM86) komandų vykdymo rezultatai.

### Mikroprocesoriaus *Intel 8086* komandų vykdymo rezultatai

Adresas	Komandų šešiolyktainiai kodai	Atliekamos funkcijos	Operandai	Vykdymo rezultatai
Tiesioginio ir registrinio adresavimo komandos				
0100	OKB =			
0101	AB =			
0102	Disp L =			
0103	Disp H =			
...	...	...	...	...
Komandos, taikančios šaltinį registrinį adresavimą ir tiesioginį operandą				
	OKB =			
	AB =			
...	...	...	...	...

2. Įjungti kompiuterį, paleisti mokomąją programą (I8086.bat) ir susipažinti su mikroprocesoriaus komandų sistema ir operandų adresavimo būdais.

3. Įjungti **MPS** ir patikrinti klavišinio (displėjaus) monitoriaus funkcionavimą.

4. Taikant tiesioginį ir registrinį adresavimą sudaryti ir įvykdyti programas, kad būtų galima:

- a) persiųsti žodį iš atminties į kaupiklį;
- b) susumuoti du, atmintyje esančius 16 bitų operandus;
- c) inkrementuoti kaupiklio turinį, besąlygiškai pereiti į kito segmento adresą ir dar kartą inkrementuoti kaupiklio turinį;
- d) atlikti keitimąsi informacija tarp kaupiklio ir registrų **CX**, **DX** ir **BX**;
- e) inkrementuoti (dekrementuoti) bet kurio registro turinį;
- f) bet kurio registro turinį persiųsti į kaupiklį;
- g) neįvertinant ženklo operandą, esantį kaupiklyje, sudauginti su operandu registre **BX**;

5. Taikant šaltinį registrinį adresavimą ir tiesioginį operandą sudaryti ir įvykdyti programas, kad galima būtų:

- a) dviejų (vieno) baitų tiesioginį operandą persiųsti į atmintį;

- b) sudėti 2 baitų tiesioginį operandą su kaupiklio turiniu;
  - c) iš atminties ląstelės turinio atimti kitos ląstelės turinį;
  - d) inkrementuoti atminties ląstelės turinį;
  - e) palyginti tiesioginį operandą su operandu atmintyje;
  - f) rasti tiesioginio operando ir operando kaupiklyje loginę sandaugą;
6. Taikant bazinį, indeksinį, bazinį-indeksinį ir santykinį adresavimą sudaryti ir įvykdyti programas, kad galima būtų:
- a) persiųsti 2 baitų operandą iš atminties į kaupiklį;
  - b) sudauginti operandą, esantį kaupiklyje, su operandu atmintyje;
  - c) jeigu **(CX)  $\neq$  0**, prie akumulatoriaus pridėti vieneta ir stabdyti programos darbą, jei **(CX)=0** – programos darbą nutraukti iškart;
  - d) atlikti 4 c punkto veiksmus taikant santykinį adresavimą.
7. Duomenų eilutę iš segmento **DS** persiųsti į segmentą **ES**.
8. Taikant dėklinį adresavimą sudaryti ir įvykdyti programas, kad galima būtų:
- a) operandą iš bet kurio registro persiųsti į dėklą nurodytu adresu;
  - b) operandą iš atminties persiųsti į dėklą nurodytu adresu;
  - c) operandą iš dėklo persiųsti į registrą;
  - d) operandą iš dėklo persiųsti į operatyviają atmintį.

#### 4. Ataskaitos turinys

1. Darbo tikslas.
2. Tiesioginio ir registrinio adresavimo komandų vykdymo pavyzdžiai.
3. Komandų su tiesioginiu operandu ir šalutiniu registriniu adresavimu vykdymo pavyzdžiai.
4. Bazinio, indeksinio, bazinio-indeksinio ir santykinio adresavimo komandų vykdymo pavyzdžiai.
5. Simbolių eilutes apdorojančių komandų vykdymo pavyzdžiai.
6. Dėklinio adresavimo komandų vykdymo pavyzdžiai.

## 5. Kontroliniai klausimai

1. Nubraižyti bendrąjį mikroprocesoriaus *Intel 8086* (K1810BM86) komandos formatą.
2. Nubraižyti adresavimo baido struktūrą ir paaiškinti jo laukų paskirtį.
3. Paaiškinti tiesioginio adresavimo principą.
4. Paaiškinti registrinio adresavimo principą.
5. Pateikti komandos su tiesioginiu operandu pavyzdį.
6. Paaiškinti bazinio, indeksinio ir bazinio-indeksinio adresavimo principą.
7. Kaip apskaičiuojamas fizinis adresas santykinio adresavimo atveju?
8. Komandų, apdorojančių simbolių eilutes, formatas.
9. Paaiškinti dėklinio adresavimo principą.

## Literatūra

1. Ю. М. Казаринов, В. Н. Номоконов, Г. С. Подклетнов, Ф.В.Филипов. Микропроцессорный комплект К1810. Структура, программирование, применение. М.: Высшая школа, 1990. с 5 - 73.
2. С. Т. Хвощ и др. Микропроцессоры и микро-ЭВМ в системах автоматического управления. Справочник под общ. ред. С.Т. Хвоща. - Л.: Машиностроение. Ленингр. отд-ние, 1987. 640 с.
3. Ю. Чжен Лю, Г. Гибсон. Микропроцессоры семейства 8086/8088. М.: Радио и связь, 1987. 510 с.
4. Г. В. Майко. Assembler для IBM PC. М.: Бизнес - Информ, Сирин, 1997. 212 с.
5. Использование Turbo Assembler при разработке программ / Составитель А. А.Чекатков. - Киев; Диалектика, 1995. 288 с.
6. В. Е. Буйкевич, А. И. Трофимов, А. Р.Тедеев, А.Ю. Федотов. Обучающая система i8086. Кафедра ЭВМ Минского радиотехнического института, 1989. Программа для персонального компьютера.