

2 laboratorinis darbas

Aštuonių skilčių mikroprocesoriaus komandų sistema ir programavimas

1. Darbo tikslas

Susipažinti su 8 skilčių mikroprocesoriaus Intel[®] 8085 komandų sistema. Įgyti programavimo assembleriu įgūdžių.

2. Bendrosios žinios

2.1. Mikroprocesoriaus charakteristikos

Mikroprocesorius Intel[®] 8085 yra tobulesnis Intel[®] 8080 mikroprocesoriaus variantas. Vidinė šio mikroprocesoriaus struktūra yra panaši į Intel[®] 8080: analogiška atminties adresų erdvė, apdorojamų žodžių ilgis, komandų sistema (papildyta dvejomis komandomis) ir adresavimo būdai.

Pagrindiniai mikroprocesoriaus Intel[®] 8085 parametrai:

- taktinis dažnis – iki 6 MHz;
- tranzistorių skaičius – 6500;
- gamybos technologija – 3 μm;
- adresuojamoji atmintis – iki 64 KB;
- adresų magistralė – 16 linijų;
- duomenų magistralė – 8 linijų;
- komandų skaičius – 246.

Mikroprocesorius Intel[®] 8085 buvo taikomas įvairiuose valdymo įrenginiuose, asmeniniuose kompiuteriuose ir netgi NASA daugkartinio naudojimo pilotuojamuose kosminiuose laivuose bei dirbtiniuose žemės palydovuose. Dabar dėl nesudėtingos architektūros ir paprastos komandų sistemos mikroprocesorius plačiai taikomas mokymo tikslams.

2.2. Registrų blokas

Duomenims saugoti mikroprocesorius Intel® 8085 turi 7 aštuonių skilčių registrus. Registras **A** vadinamas kaupikliu ir skirtas keitimuisi informacija su išoriniais įtaisais. Atliekant aritmetines, logines ir poslinkio operacijas, kaupiklyje laikomas vienas iš operandų. Jame talpinamas operacijos vykdymo rezultatas. Kiti šeši registrai – **B, C, D, E, H** ir **L** – sudaro vadinamąjį bendros paskirties registrų (BPR) bloką ir gali būti naudojami ir duomenims ir adresams laikyti. Jai reikia saugoti 16 skilčių dvejetainius skaičius šie pavieniai 8 skilčių registrai jungiami į poras **BC, DE, HL**. Asemblerio komandose šios poros identifikuojamos kaip **B, D** ir **H**.

16 skilčių dėklo rodyklė **SP** skirta dėklo atminties ląstelėms adresuoti. Programuotojui yra prieinamos atskirai žemesnioji **SPL** ir aukštesnioji **SPH** 8 skilčių registro dalys.

16 skilčių komandų skaitiklio **PC** paskirtis – laikyti komandos adresą. Išrinkus iš atminties eilinę komandą, skaitiklio turinys padidinamas vienetu, t. y. adresuojamas kitas komandos baitas (jeigu programoje nėra sąlyginių arba besąlyginių pereinimų).

8 skilčių požymių registras **F** skirtas tam tikriems operacijos vykdymo rezultato požymiams fiksuoti. Požymis užfiksuojamas į atitinkamą registro skiltį įrašant 1 arba 0:

7	6	5	4	3	2	1	0
S	Z	0	AC	0	P	1	CY

- Nulinė – žemiausioji skiltis **CY** (angl. *Carry*) – pernašos: **CY** = 1, jei operacijos rezultatas netelpa į 8 skiltis (įvyksta pernaša iš aukščiausiosios skilties arba buvo skolintasi atliekant atimties veiksmą), **CY** = 0 – pernašos nėra;
- Antroji skiltis **P** (angl. *Parity*) – lyginumo: **P** = 1, jei rezultato dvejetainiame kode vienetų skaičius yra lyginis, **P** = 0 – jei nelyginis;

- Ketvirtoji skiltis **AC** (angl. *Auxiliary Carry*) – pagalbinės pernašos: **AC** = 1, kai įvyksta pernaša iš dvejetainio skaičiaus trečiosios skilties, **AC** = 0 – pernašos nėra;
- Šeštoji skiltis **Z** (angl. *Zero*) – nulio arba nulinio rezultato: **Z** = 1, jei operacijos rezultatas lygus nuliui, **Z** = 0 – jei nelygus nuliui;
- Septintoji – aukščiausioji skiltis **S** (angl. *Sign*) – ženklas: **S** = 1, jei operacijos rezultatas yra neigiamas, **S** = 0 – jei teigiamas.
Pirmoji, trečioji ir penktoji požymių registro **F** skiltys nekinta, nes netaikomos.

2.3. Aritmetinis-loginis įtaisas

Aštuonių skilčių aritmetinis-loginis įtaisas (ALJ) gali atlikti 4 aritmetines operacijas (sudėtį ir atimtį, su pernaša ir be jos), 4 logines ir 4 slinkties operacijas. Atliekant aritmetines ir logines operacijas, vienas iš operandų talpinamas kaupiklyje. Operacijos vykdymo rezultatas pasilieka kaupiklyje. Vykdoma tik kaupiklio ciklinė slinktis.

Numatyta galimybė aritmetinius veiksmus atlikti ir su dešimtainiais skaičiais. Dešimtainiam skaičiui saugoti registro skiltys dalijamos į dvi grupes po 4 ir kiekvienoje grupėje laikomas vienas dešimtainis skaitmuo, užkoduotas 8421 kodu.

ALJ tiesiogiai susietas su valdymo bloku, susidedančiu iš komandų registro, į kurį siunčiamas pirmasis komandos baitas, ir valdymo signalų formavimo įtaiso. Pastarajame yra valdančioji atmintis, kurioje saugomos įvairių operacijų mikroprogramos. Valdančioji atmintis, taigi ir operacijų mikroprogramos, vartotojui neprieinamos.

2.4. Duomenų ir komandų formatai

Duomenys – tai apdorojama dvejetainė informacija ir jos apdorojimo rezultatai. Dažniausiai duomenys laikomi operatyviojoje atmintyje bei mikroprocesoriaus registruose. Aštuonių skilčių dveje-

tainis skaičius vadinamas baitu (D8). Adresai yra 16 bių, todėl, kaip ir 16 bitų duomenys, laikomi dalimis: žemesnysis baitas – D16L, arba ADRL, ir aukštesnysis baitas – D16H, arba ADRH.

Komandų formatai gali būti vieno, dviejų ir trijų baitų. Baitų reikšmės pateiktos 1 lentelėje. Vieno baito komandose operacijos kodo baitas (OKB) yra vienintelis.

1 lentelė. Komandų formatai

Formatas	1 baitas	2 baitas	3 baitas
Vieno baito	OKB	–	–
Dviejų baitų	OKB	D8 arba prievado Nr. (PORT)	–
Trijų baitų	OKB	D16L arba ADRL	D16H arba ADRH

2.5. Operandų adresavimo būdai

Operacijai vykdyti komandoje, be operacijos tipo, turi būti nurodyti ir operandai. Aptarsime taikomus operandų adresavimo būdus.

Registrinis adresavimas. Operandai, esant šiam adresavimui, yra bendrosios paskirties registruose, kurių adresai nurodomi vieninteliame OKB. Kadangi registrų mažai, jiems adresuoti pakanka trijų dvejetainių skilčių, kaip parodyta 2 lentelėje.

2 lentelė. Registrų adresai

Registras	Adresas ₂ komandoje	Registras	Adresas ₂ komandoje
B	000	H	100
C	001	L	101
D	010	M	110
E	011	A	111

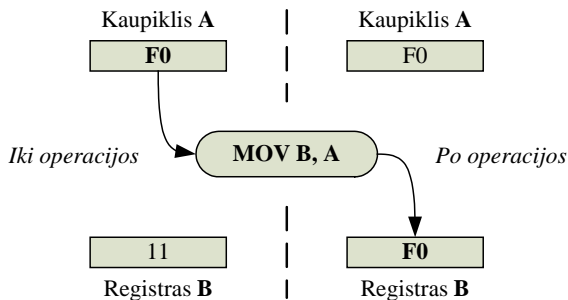
Šioje lentelėje **M** – atminties ląstelė, kurios adresas nurodomas registrų poroje **H** (**H** ir **L** registrai).

Pavyzdžiui, komandos **MOV R1, R2** formatas yra toks:

7	6	5	4	3	2	1	0
0	1	R1			R2		

Irašę vietoje **R1** (registas, į kurį siunčiami duomenys) ir **R2** (registas, iš kurio siunčiami duomenys) konkrečių registrų adresus, gauname dvejetainį komandos kodą.

Tarkime, **R1** yra registras **B**, o **R2** – registras **A**. Įrašius į komandos formatą atitinkamus registrų adresus 000 ir 111 (1.1 lentelė), gauname dvejetainį komandos kodą: $01000111_2 = 47_{16}$. Šis šešioliktainis kodas atitinka komandą **MOV B, A**. **B ← A**:



Tai reiškia, kad kaupiklio **A** turinys persiunčiamas į registrą **B**.

Taigi komandos su registriniu operandų adresavimu yra vieno baido ilgio:

1 baitas

C7	C6	C5	C4	C3	C2	C1	C0
----	----	----	----	----	----	----	----

 – operacijos kodo baitas (OKB)

Operandų adresai nurodomi vieninteliame operacijos kodo baite tiesiogiai.

Tiesioginis adresavimas. Komandos su tiesioginiu operandų adresavimu yra dviejų arba trijų baitų ilgio. Čia prievado numeris (adresas) yra antrasis komandos baitas, o operando adresas yra antrasis ir trečiasis baitai:

1 baitas

C7	C6	C5	C4	C3	C2	C1	C0
----	----	----	----	----	----	----	----

 – operacijos kodo baitas (OKB)

2 baitas

A7	A6	A5	A4	A3	A2	A1	A0
----	----	----	----	----	----	----	----

 – prievado numeris (adresas) (PORT)

1 baitas

C7	C6	C5	C4	C3	C2	C1	C0
----	----	----	----	----	----	----	----

 – operacijos kodo baitas (OKB)

2 baitas

A7	A6	A5	A4	A3	A2	A1	A0
----	----	----	----	----	----	----	----

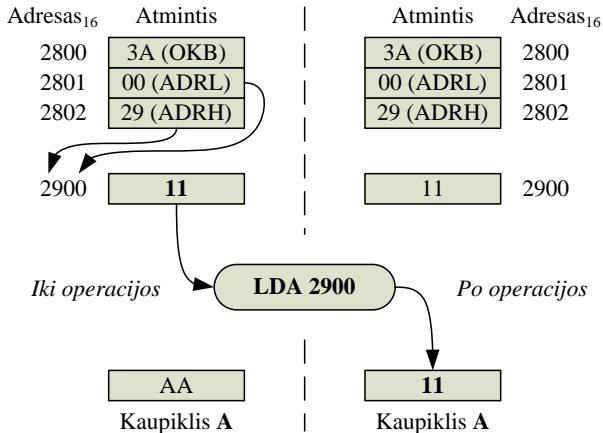
 – žemesnysis adreso baitas (ADRL)

3 baitas

A7	A6	A5	A4	A3	A2	A1	A0
----	----	----	----	----	----	----	----

 – aukštesnysis adreso baitas (ADRH)

Atminties adresai yra 16 bitų, todėl, kaip ir 16 bitų duomenys, laikomi dalimis: ADRL ir ADRH – žemesnysis ir aukštesnysis adreso, kuriame yra operandas, baitai. Pavyzdžiui, **LDA ADR. A ← M[ADR]**. Čia **M[ADR]** – operando adresas atmintyje (antrasis ir trečiasis komandos baitai).



Tai reiškia, kad į kaupiklį **A** persiunčiami duomenys iš atminties ląstelės, kurios adresą nurodo antrasis ir trečiasis komandos baitai.

Šalutinis registrinis adresavimas. Šalutinis registrinis operandų adresavimo būdas skiriasi nuo registrinio tuo, kad registruose laikomas ne pats operandas, o šio operando adresas atmintyje. Kadangi adresas yra 16 bitų, jam saugoti taikoma registrų pora. Pavyzdžiui, komanda **MOV R,M** reiškia, kad į registrą **R** bus persiųsti

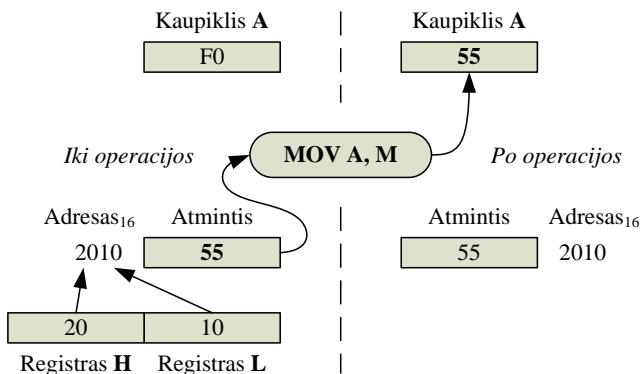
duomenys iš atminties ląstelės, kurios adresą nurodo registrų pora **H** (**H** ir **L** registrai), **M** šiuo atveju reiškia registrų porą **H**.

Komandos **MOV R, M** formatas yra toks:

7	6	5	4	3	2	1	0
0	1	R			M		

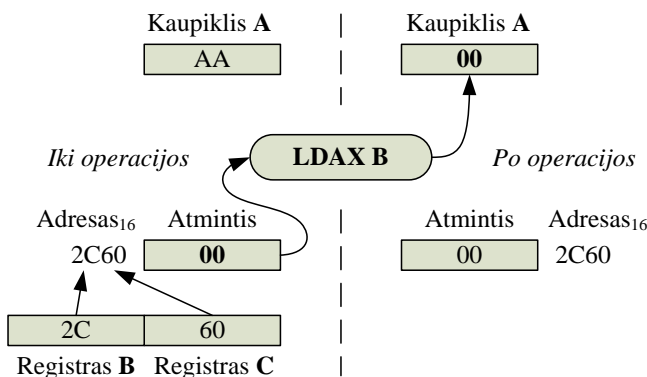
Įrašę vietoje **R** (registas, į kurį siunčiami duomenys) ir **M** (atminties ląstelės, kurios adresą nurodo registrų pora **H**, turinys) konkrečius adresus, gauname dvejetainį komandos kodą.

Tarkime, **R** yra registras **A**. Įrašius į komandos formatą atitinkamus adresus 111 ir 110 (2 lentelė), gauname dvejetainį komandos kodą: $01111110_2 = 7E_{16}$. Šis šešiolyktainis kodas atitinka komandą **MOV A, M. A** \leftarrow **M[HL]**:



Tai reiškia, kad į kaupiklį **A** persiunčiami duomenys iš atminties ląstelės, kurios adresą nurodo registrų pora **H** (**H** ir **L** registrai), **M** šiuo atveju reiškia registrų porą **H**.

Taip pat šiam tikslui gali būti taikomos **B** (**B** ir **C** registrai) arba **D** (**D** ir **E** registrai) registrų poros. Pavyzdžiui, **LDAX B. A** \leftarrow **M[BC]**; **STAX B. M[BC]** \leftarrow **A**. Čia **M[BC]** reiškia atminties ląstelę, kurios adresu yra registrų poros **B** turinys.



Tai reiškia, kad į kaupiklį **A** persiunčiami duomenys iš atminties ląstelės, kurios adresą nurodo registrų pora **B** (**B** ir **C** registrai).

Dėklo komandos irgi taiko šalutinį registrinį adresavimą. Dėklas yra specialioji operatyviosios atminties sritis, skirta grįžties iš pa-programių adresams, tarpiniams skaičiavimų rezultatams, konstantoms ar duomenims ir pan. laikinai saugoti, t. y. laikina informacijos saugykla. Dėklą programuotojas gali susikurti (adresuoti) taikydamas specialųjį 16 skilčių registrą **SP**, vadinamą dėklo rodykle arba dėklo viršūne.

Dėklas veikia LIFO (angl. *Last In, First Out*) principu (paskutinis į, pirmas iš), t. y. duomenys į dėklą įrašomi iš eilės tokia tvarka, kokia pateikiami, tad skaityti iš dėklo galima tik paskiausiai įrašytus duomenis. Juos nuskaičius bus galima skaityti prieš juos buvusius duomenis ir t. t. Toki dėklo veikimo principą galima vaizdžiai paaiškinti lėkščių rietuvės pavyzdžiu (1 pav.).

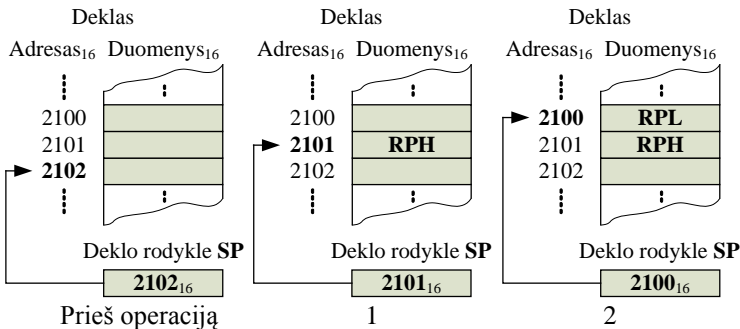


1 pav. Lėkščių rietuvė

Norint paimiti pirmąją lėkštę pirmiausiai reikia nukelti trečiąją, o paskui antrąją lėkštę, t. y. pirmiausiai į dėklą įrašytus duomenis bus galima nuskaityti paskutiniaisiais.

Dėklo rodyklėje **SP** saugomas dėklo pradžios adresas, kuris nurodo kur turi būti įrašyti arba iš kur turi būti nuskaityti duomenys.

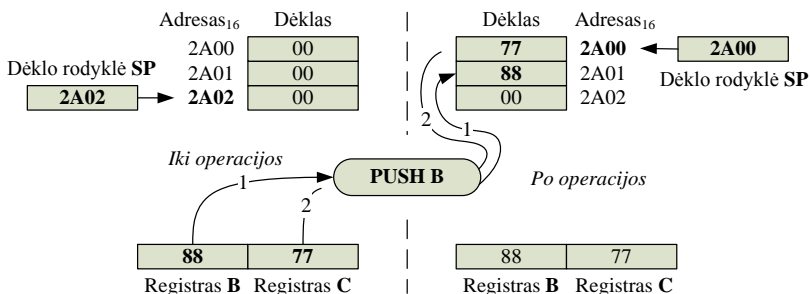
Duomenims rašyti į dėklą yra skirta komanda **PUSH**. Jos veikimo principas parodytas 2 paveiksle.



2 pav. Komandos **PUSH** veikimas

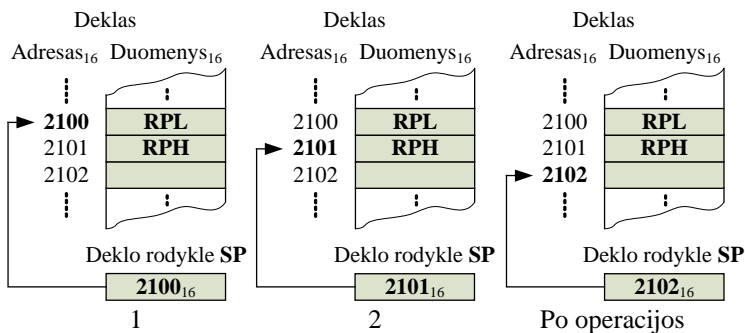
Komandoje **PUSH** nurodytos registrų poros **RP** turinys įrašomas dviejose dėklo atminties ląstelėse, kurių adresus nurodo dėklo rodyklė **SP**. Pirmojo registrų poros registro **RPH** turinys įrašomas į dėklo atminties ląstelę, kurios adresas vienetu mažesnis nei esama dėklo rodyklės **SP** reikšmė (1), o antrojo registrų poros registro **RPL** turinys įrašomas į dėklo atminties ląstelę, kurios adresas dviem vienetais mažesnis nei esama dėklo rodyklės **SP** reikšmė (2). Dėklo rodyklės **SP** turinys įvykdžius **PUSH** komandą sumažėja dviem vienetais. Taigi dėklas užpildomas, t. y. duomenys į dėklą įrašomi adresų mažėjimo kryptimi.

Pavyzdžiui, komanda **PUSH B** veikia taip:



Tai reiškia, kad pirmojo registrų poros registro **B** turinys persiunčiamas į atminties ląstelę, kurios adresas vienetu mažesnis nei esama dėklo rodyklės **SP** reikšmė, o antrojo registrų poros registro **C** turinys persiunčiamas į atminties ląstelę, kurios adresas dviem vienetais mažesnis nei esama dėklo rodyklės **SP** reikšmė. Dėklo rodyklės **SP** turinys įvykdžius **PUSH** komandą sumažėja dviem vienetais.

Duomenims skaityti iš dėklo yra skirta komanda **POP**. Jos veikimo principas parodytas 3 paveiksle.

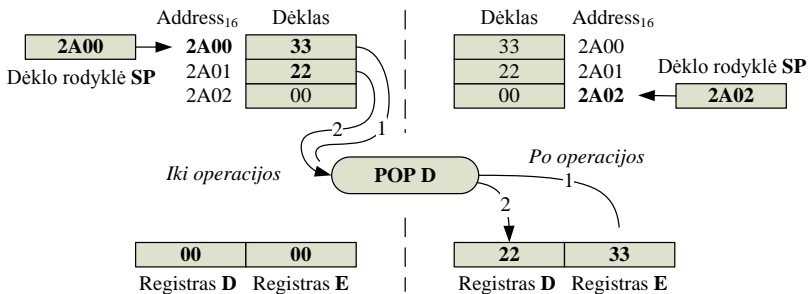


3 pav. Komandos POP veikimas

Dėklo atminties ląstelių, kurių adresus nurodo dėklo rodyklė **SP**, turiniai įrašomi į komandoje **POP** nurodytą registrų porą **RP**. Atminties ląstelės, kurios adresas sutampa su esama dėklo rodyklės **SP** reikšme, turinys įrašomas į antrąją registrų poros registrą **RPL** (1), o

atminties ląstelės, kurios adresas vienetu didesnis nei esama dėklo rodyklės **SP** reikšmė, turinys įrašomas į pirmąjį registrų poros registrą **RPH** (2). Dėklo rodyklės **SP** turinys įvykdžius **POP** komandą didėja dviem vienetais. Taigi duomenys iš dėklo skaitomi adresų didėjimo kryptimi.

Pavyzdžiui, komanda **POP D** veikia taip:



Tai reiškia, kad atminties ląstelės, kurios adresas sutampa su esama dėklo rodyklės **SP** reikšme, turinys persiunčiamas į antrąjį registrų poros registrą **E**, o atminties ląstelės, kurios adresas vienetu didesnis nei esama dėklo rodyklės **SP** reikšmė, turinys persiunčiamas į pirmąjį registrų poros registrą **D**.

Taigi komandos su šalutiniu registriniu operandų adresavimu yra vieno baido ilgio:

1 baitas

C7	C6	C5	C4	C3	C2	C1	C0
----	----	----	----	----	----	----	----

 – operacijos kodo baitas (OKB)

Tiesioginis operandas. Tiesioginis operandas, tai vieno arba dviejų baitų duomenys, prijungti prie operacijos kodo baito (antrasis arba antrasis ir trečiasis komandos baitai). Komandos su tiesioginiu operandu yra dviejų arba trijų baitų ilgio:

1 baitas

C7	C6	C5	C4	C3	C2	C1	C0
----	----	----	----	----	----	----	----

 – operacijos kodo baitas (OKB)

2 baitas

D7	D6	D5	D4	D3	D2	D1	D0
----	----	----	----	----	----	----	----

 – 8 bitų duomenys (D8)

1 baitas

C7	C6	C5	C4	C3	C2	C1	C0
----	----	----	----	----	----	----	----

 – operacijos kodo baitas (OKB)

2 baitas

D7	D6	D5	D4	D3	D2	D1	D0
----	----	----	----	----	----	----	----

 – žemesnysis 16 bitų duomenų baitas (D16L)

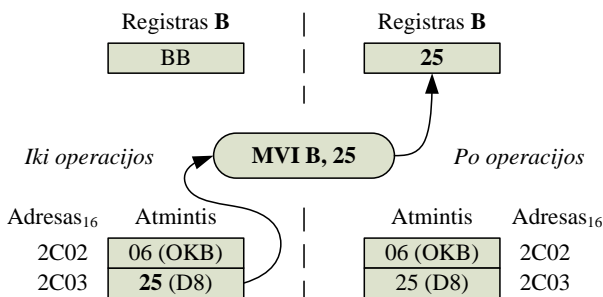
3 baitas

D7	D6	D5	D4	D3	D2	D1	D0
----	----	----	----	----	----	----	----

 – aukštesnysis 16 bitų duomenų baitas (D16H)

16 bitų duomenys, laikomi dalimis: žemesnysis baitas – D16L, aukštesnysis baitas – D16H.

Komandų su tiesioginiu operandu formatas yra panašus į komandų formatą su tiesioginiu operandų adresavimu. Skirtumas tik tas, kad esant tiesioginiam operandui po operacijos kodo baido seka ne adreso, o duomenų baitai. Pavyzdžiui, komanda **MVI R, D8**. **R ← D8**. Čia **D8** – duomenų baitas, einantis po operacijos kodo baido (antrasis komandos baitas).



Tai reikia, kad į registrą **B** persiunčiamas 8 bitų tiesioginis operandas (antrasis komandos baitas).

3. Užduotis

1. Įjungti ir parengti darbui mikroprocesorinę mokomąją sistemą M85-01.

2. Užprogramuoti ir įvykdyti duotas komandas kiekvienam operandų adresavimo būdui:

a) registrinis operandų adresavimas

ADD B; SUB L; MOV D, H; INR E; DCR C; RLC; RRC.

b) tiesioginis operandų adresavimas

LDA 2200; STA 203A; LHLD 2015; SHLD 200B.

c) šalutinis registrinis operandų adresavimas

ADD M; SUB M; MOV M, C; LDAX D; STAX B; PUSH D; POP B.

d) tiesioginis operandas

ADI 0C; SUI 01; MVI L, 05; MVI M, 55; LXI H, 2010; LXI SP, 2012.

Eksperimento rezultatus surašyti į 3 lentelę.

3 lentelė. Operandų adresavimo būdai

Adresas ₁₆	Komandos kodas		Pradiniai duomenys	Rezultatai	Pastabos
	šešiolyktainis	mnemoninis			
a) registrinis operandų adresavimas					
2000	80	ADD B	A = 05 B = 06	A = 0B	A ← A + B
2001	EF	RST 5			
...
b) tiesioginis operandų adresavimas					
...

Pastaba: komandų aprašymai ir šešioliktainiai kodai pateikti priede.

3. Sudaryti ir įvykdyti programą, kuri susumuotų du 8 bitų operandus, esančius skirtingose atminties ląstelėse ir rezultatą įrašytų į trečiąją atminties ląstelę. Operandams adresuoti parinkti skirtingus operandų adresavimo būdus. Užrašant programą ir jos vykdymo rezultatus pasinaudoti 3 lentele.

4. Sudaryti ir įvykdyti programą, kuri skaičiuotų nurodytą reiškinį, ir gautą rezultatą w įrašytų į pasirinktą atminties ląstelę. 8 bitų operandai x , y ir z , yra pasirinktose skirtingose atminties ląstelėse. Operandams adresuoti parinkti skirtingus operandų adresavimo būdus. Užrašant programą ir jos vykdymo rezultatus remtis 3 lentele.

$$w = 2x + (y - z) / 4.$$

5. Sudaryti ir įvykdyti programą, kuri sudėtų du 8 bitų operandus, esančius dėkle, ir gautą rezultatą įrašytų atgal į dėklą. Užrašant programą ir jos vykdymo rezultatus remtis 3 lentele.

6. Sudaryti ir įvykdyti programą, kuri palygintų du 8 bitų operandus, iš didesniojo atimtų mažesnį ir skirtumą įrašytų į dėklą pasirinktu adresu. Užrašant programą ir jos vykdymo rezultatus remtis 3 lentele.

4. Ataskaitos turinys

1. Darbo tikslas.
2. Komandų su skirtingais operandų adresavimo būdais vykdymo rezultatai (3 lentelė).
3. Užduoties 3–6 p. programos ir jų vykdymo rezultatai (3 lentelė).
4. Darbo rezultatų apibendrinimas.

5. Kontroliniai klausimai

1. Pagrindinės mikroprocesoriaus Intel® 8085 charakteristikos?

2. Išvardinti mikroprocesoriaus Intel® 8085 bendrosios paskirties registrus.

3. Mikroprocesoriaus Intel® 8085 požymių registro skilčių reikšmės.

4. Kokia komandų skaitiklio paskirtis?

5. Kas vadinama dėklu ir kokia dėklo rodyklės paskirtis?

6. Kokius veiksmus gali atlikti mikroprocesoriaus Intel® 8085 aritmetinis-loginis įtaisas?

7. Išvardinti ir paaiškinti mikroprocesoriaus Intel® 8085 komandų formatus.

8. Koks operandų adresavimas vadinamas registriniu?

9. Koks operandų adresavimas vadinamas šalutiniu registriniu?

10. Koks operandų adresavimas vadinamas tiesioginiu?

11. Koks operandas vadinamas tiesioginiu?

Literatūra

GRAŽULEVIČIUS, G. 2008. *Mikroprocesorinė technika*: mokomoji knyga. I dalis. Vilnius: Technika, 224 p. ISBN 978-9955-28-280-8.

ROUTT, W. A. 2007. *Microprocessor Architecture, Programming, and Systems Featuring the 8085*. USA, New York: Thomson Delmar Learning, 271 p. ISBN 1-4180-3241-7.

COFFRON, J. W. *Microprocessor Programming, Troubleshooting and Interfacing: The Z80, 8080 and 8085*. USA, New Jersey: Pearson Prentice Hall; 1st Edition. 1997, October 28. 528 p. ISBN 0-13-581976-8.

UFFENBECK, J. *Microcomputers and Microprocessors: The 8080, 8085, and Z-80 Programming, Interfacing and Troubleshooting*. USA, New Jersey: Pearson Prentice Hall; 3rd Edition. 1999, June 18. 729 p. ISBN 0-13-209198-4.

Priedas

Mikroprocesoriaus Intel® 8080 / 8085 komandų sistema

Vieno baito persiuntimai		Dviejų baitų persiuntimai	
MOV R1, R2.	R1 ← R2.	LXI RP, D16.	RPL ← D16L, RPH ← D16H.
MOV R, M.	R ← M[HL].	SHLD ADR.	M[ADR] ← L, M[ADR+1] ← H.
MOV M, R.	M[HL] ← R.	LHLD ADR.	L ← M[ADR], H ← M[ADR+1].
MVI R, D8.	R ← D8.	PUSH RP.	M[SP - 1] ← RPH, M[SP - 2] ← RPL, SP ← SP - 2.
MVI M, D8.	M[HL] ← D8.	PUSH PSW.	M[SP - 1] ← A, M[SP - 2] ← F, SP ← SP - 2.
STAX RP.	M[RP] ← A.	POP RP.	RPL ← M[SP], RPH ← M[SP + 1], SP ← SP + 2.
LDAX RP.	A ← M[RP].	POP PSW.*	F ← M[SP], A ← M[SP + 1], SP ← SP + 2.
STA ADR.	M[ADR] ← A.	SPHL.	SP ← HL.
LDA ADR.	A ← M[ADR].	Apsikeitimas baitais	
Išvesties ir išvesties komandos		XCHG.	H ↔ D, L ↔ E.
IN PORT.	A ← PORT.	XTHL.	L ↔ M[SP], H ↔ M[SP + 1].
OUT PORT.	PORT ← A.		
Aritmetinės ir loginės operacijos su vienu operandu			
8 bitų operacijos			
CMC.**	CY ← ¬CY.	INR R.***	R ← R + 1.
STC.**	CY ← 1.	DCR R.***	R ← R - 1.
CMA.	A ← ¬A.	INR M.***	M[HL] ← M[HL] + 1.
		DCR M.***	M[HL] ← M[HL] - 1.
		INX RP.	RP ← RP + 1.
		DCX RP.	RP ← RP - 1.
		Dešimtainė korekcija	
		DAA.*	Jei A ₃₋₀ > 9 arba AC = 1, tai A ₃₋₀ ← A ₃₋₀ + 6; jei A ₇₋₄ > 9 arba CY = 1, tai A ₇₋₄ ← A ₇₋₄ + 6.
Aritmetinės ir loginės operacijos su dviem operandais			
8 bitų operacijos			
ADD R.*	A ← A + R.	ADI D8.*	A ← A + D8.
ADC R.*	A ← A + R + CY.	ACI D8.*	A ← A + D8 + CY.
SUB R.*	A ← A - R.	SUI D8.*	A ← A - D8.
SBB R.*	A ← A - (R + CY).	SBI D8.*	A ← A - (D8 + CY).
ANA R.*	A ← A ∧ R.	ANI D8.*	A ← A ∧ D8.
ORA R.*	A ← A ∨ R.	ORI D8.*	A ← A ∨ D8.
XRA R.*	A ← A ∨ R.	XRI D8.*	A ← A ∨ D8.
CMP R.*	A - R.		
CMP M.*	A - M[HL].		
CPI D8.*	A - D8.	Kaupiklio turinio poslinkio komandos	
16 bitų operacijos		RLC.**	A _{n+1} ← A _n , n = 0-6, A ₀ ← A ₇ , CY ← A ₇ . Poslinkis į kairę.
		RRC.**	A _n ← A _{n+1} , n = 0-6, A ₇ ← A ₀ , CY ← A ₀ . Poslinkis į dešinę.
		RAL.**	A _{n+1} ← A _n , n = 0-6, A ₀ ← CY, CY ← A ₇ . Poslinkis į kairę per CY skiltį.
		RAR.**	A _n ← A _{n+1} , n = 0-6, A ₇ ← CY, CY ← A ₀ . Poslinkis į dešinę per CY skiltį.
DAD RP.**	HL ← HL + RP.		
Pereigų komandos		Kreipties į paprogramį ir grįžties iš jo komandos	
PCHL.	PC ← HL.	CALL ADR.	M[SP] ← PC + 3, SP ← SP - 2, PC ← ADR.
JMP ADR.	PC ← M[ADR].	Ccc ADR.	M[SP] ← PC + 3, SP ← SP - 2, PC ← ADR.
Jcc ADR.	PC ← M[ADR].	RST N.	PC ← 8 × N (N = 0, 1, ..., 7), 8 × N = ADR. ADR = 0 ₁₆ , 8 ₁₆ , 10 ₁₆ , 18 ₁₆ , 20 ₁₆ , 28 ₁₆ , 30 ₁₆ , 38 ₁₆ .
Mikroprocesoriaus valdymo komandos		RET.	PC ← M[SP], SP ← SP + 2.
EI	Leisti pertrauktis.	Rcc.	PC ← M[SP], SP ← SP + 2.
DI	Drausti pertrauktis.	Požymių registro F formatas	
HLT	Stabdyti mikroprocesoriaus darbą.	D7	D6 D5 D4 D3 D2 D1 D0
NOP	PC ← PC + 1. Nėra operacijos.	S	Z 0 AC 0 P 1 CY

Sutartiniai žymėjimai:

* – komanda keičia visus požymius.

** – komanda keičia požymį **CY**.

*** – komanda keičia visus požymius išskyrus **CY**.

R, R1, R2 – vieno iš registrų **A, B, C, D, E, H** arba **L** turinys (8 bitai);

M – atminties ląstelė, kurios adresą nurodo registrų pora **HL**;

M[HL] – atminties ląstelės, kurios adresą nurodo registrų pora **HL**, turinys (8 bitai);

D8 – 8 bitų tiesioginis operandas (antrasis komandos baitas);

D16 – 16 bitų tiesioginis operandas (antrasis ir trečiasis komandos baitai);

D16L ir **D16H** – žemesnysis ir aukštesnysis 16 bitų tiesioginio operando baitai;

ADR – operando 16 bitų adresas (antrasis ir trečiasis komandos baitai);

M[ADR] – 8 bitų operandas, kurio 16 bitų adresas antrasis ir trečiasis komandos baitai;

RP – registrų poros **BC, DE, HL** arba dėklo rodyklės **SP** turinys (16 bitų);

RPL ir **RPH** – žemesnysis ir aukštesnysis registrų poros registrai;

M[RP] – atminties ląstelės, kurios adresą nurodo registrų pora **BC** arba **DE**, turinys (8 bitai);

M[SP] – dėklas;

SP – 16 bitų dėklo rodyklė;

PC – 16 bitų komandų skaitiklis;

PSW – 16 bitų procesoriaus būsenos žodis (kaupiklio **A** ir požymių registro **F** turiniai);

PORT – įvesties arba išvesties prievado 8 bitų numeris (antrasis komandos baitas);

N – pertraukties aptarnavimo paprogramio numeris;

n – registro bito numeris (bitai numeruojami iš dešinės į kairę nuo 0 iki 7);

cc – nurodo sąlyginę pereigą, kreiptį į paprogramį arba grįžtį iš jo (mnemonikoje **cc** pakeičiamas **NZ, Z, NC, C, PO, PE, P** arba **M**);

CY – požymių registro **F** pernašos skiltis;

P – požymių registro **F** lyginumo skiltis;

AC – požymių registro **F** pagalbinės pernašos skiltis;

Z – požymių registro **F** nulio skiltis;

S – požymių registro **F** ženklų skiltis;

\wedge – loginis IR;

\vee – loginis ARBA;

∇ – loginis išskirtinis (griežtasis) ARBA;

\neg – inversija;

\leftarrow – perdavimas;

\leftrightarrow – sukeitimas.

Mikroprocesoriaus Intel® 8080 / 8085 komandų šešiolyktainiai kodai

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0	NOP	LXI B,D16	STAX B	INX B	INR B	DCR B	MVI B,D8	RLC	–	DAD B	LDAX B	DCX B	INR C	DCR C	MVI C,D8	RRC	0
1	–	LXI D,D16	STAX D	INX D	INR D	DCR D	MVI D,D8	RAL	–	DAD D	LDAX D	DCX D	INR E	DCR E	MVI E,D8	RAR	1
2	RIM ¹	LXI H,D16	SHLD ADR	INX H	INR H	DCR H	MVI H,D8	DAA	–	DAD H	LHLD ADR	DCX H	INR L	DCR L	MVI L,D8	CMA	2
3	SIM ¹	LXI SP, D16	STA ADR	INX SP	INR M	DCR M	MVI M,D8	STC	–	DAD SP	LDA ADR	DCX SP	INR A	DCR A	MVI A,D8	CMC	3
4	MOV B,B	MOV B,C	MOV B,D	MOV B,E	MOV B,H	MOV B,L	MOV B,M	MOV B,A	MOV C,B	MOV C,C	MOV C,D	MOV C,E	MOV C,H	MOV C,L	MOV C,M	MOV C,A	4
5	MOV D,B	MOV D,C	MOV D,D	MOV D,E	MOV D,H	MOV D,L	MOV D,M	MOV D,A	MOV E,B	MOV E,C	MOV E,D	MOV E,E	MOV E,H	MOV E,L	MOV E,M	MOV E,A	5
6	MOV H,B	MOV H,C	MOV H,D	MOV H,E	MOV H,H	MOV H,L	MOV H,M	MOV H,A	MOV L,B	MOV L,C	MOV L,D	MOV L,E	MOV L,H	MOV L,L	MOV L,M	MOV L,A	6
7	MOV M,B	MOV M,C	MOV M,D	MOV M,E	MOV M,H	MOV M,L	HLT	MOV M,A	MOV A,B	MOV A,C	MOV A,D	MOV A,E	MOV A,H	MOV A,L	MOV A,M	MOV A,A	7
8	ADD B	ADD C	ADD D	ADD E	ADD H	ADD L	ADD M	ADD A	ADC B	ADC C	ADC D	ADC E	ADC H	ADC L	ADC M	ADC A	8
9	SUB B	SUB C	SUB D	SUB E	SUB H	SUB L	SUB M	SUB A	SBB B	SBB C	SBB D	SBB E	SBB H	SBB L	SBB M	SBB A	9
A	ANA B	ANA C	ANA D	ANA E	ANA H	ANA L	ANA M	ANA A	XRA B	XRA C	XRA D	XRA E	XRA H	XRA L	XRA M	XRA A	A
B	ORA B	ORA C	ORA D	ORA E	ORA H	ORA L	ORA M	ORA A	CMP B	CMP C	CMP D	CMP E	CMP H	CMP L	CMP M	CMP A	B
C	RNZ	POP B	JNZ ADR	JMP ADR	CNZ ADR	PUSH B	ADI D8	RST 0	RZ	RET	JZ ADR	–	CZ ADR	CALL ADR	ACI D8	RST 1	C
D	RNC	POP D	JNC ADR	OUT PORT	CNC ADR	PUSH D	SUI D8	RST 2	RC	–	JC ADR	IN PORT	CC ADR	–	SBI D8	RST 3	D
E	RPO	POP H	JPO ADR	XTHL	CPO ADR	PUSH H	ANI D8	RST 4	RPE	PCHL	JPE ADR	XCHG	CPE ADR	–	XRI D8	RST 5	E
F	RP	POP PSW	JP ADR	DI	CP ADR	PUSH PSW	ORI D8	RST 6	RM	SPHL	JM ADR	EI	CM ADR	–	CPI D8	RST 7	F
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	

¹ Komanda palaikoma tik mikroprocesoriaus Intel® 8085.