

Міністерство освіти і науки України
Національний гірничий університет

Кафедра систем електропостачання

Методичні вказівки
до виконання лабораторної роботи МП – 4
„Програмування мікроконтролера K1-20”

Дніпропетровськ
2005

Міністерство освіти і науки України
Національний гірничий університет

Кафедра систем електропостачання

Методичні вказівки
до виконання лабораторної роботи МП – 4
„Програмування мікроконтролера К1-20”
для студентів спеціальностей:
7.000008 “Енергетичний менеджмент
7.090603, “Електротехнічні системи електроспоживання”
7.092204 „Електромеханічне обладнання енергоємних виробництв”

Дніпропетровськ
2005

Методичні вказівки до виконання лабораторної роботи МП – 4 „Програмування мікроконтролера К1-20” для студентів спеціальностей 7.000008 “Енергетичний менеджмент; 7.090603, “Електротехнічні системи електроспоживання”; 7.092204 „Електромеханічне обладнання енергоємних виробництв” / Укл.: Г.М. Бажін, А.В. Рухлов – Дніпропетровськ: НГУ, каф. систем електропостачання, 2005.– 10 с.

1. Мета роботи.

Вивчити систему команд мікропроцесора KP580BB80. Вивчити методику складання простих програм, їх асемблювання і рішення на мікроконтролері.

У результаті виконання роботи студент повинен:

- знати архітектуру мікропроцесора KP580BB80, систему його команд;
- уміти складати прості лінійні програми, асемблювати їх і виконувати за допомогою мікроконтроллера „Електроніка К1 -20”.

2. Теоретичні положення роботи

2.1. Архітектура мікропроцесора

Мікропроцесор має складну внутрішню структуру, однак з боку зору програміста він складається з декількох регістрів (акумулятора, шести 8-розрядних регістрів загального призначення, регістра признаків та двох 16-розрядних регістрів – вказівника стеку (SP) і лічильника команд (PC)).(Рис. 1.)



Рис. 1. Архітектура мікропроцесора серії KP580

Акумулятор та регістри загального призначення використовуються для тимчасового зберігання даних або адреси. Акумулятор (позначається літерою А) призначено для зберігання одного з операндів перед виконанням арифметико-логічних операцій, а також результату що отримано після їх виконання. Восьми-розрядні регістри загального призначення прийнято позначати латинськими літерами В, С, D, Е, Н, та L. Вони призначаються для тимчасового зберігання проміжних даних. Регістри програмно доступні. При виконанні деяких команд регістри В і С, D і Е, Н і L об'єднуються у регістрові пари. Це дозволяє використовувати їх для зберігання адреси або 16-розрядних операндів.

Регістр ознак (F) призначено для зберігання характеристик (ознак), які характеризують результат виконання арифметичних та логічних команд. Відображаються такі ознаки:

- C – ознака наявності переносу із старшого розряду;
- P – ознака парності (парної кількості одиниць) в отриманому результаті;
- AC – ознака наявності допоміжного переносу (з розряду 3 в розряд 4);
- Z – ознака наявності нульового результату;
- S – ознака знаку (позитивний або негативний).

Вказівник стеку (SP) – програмно доступний регістр, вказує на верхівку стека. (Стек – область пам'яті, яка виділяється для тимчасового зберігання вмісту регістрових пар або слова стану процесора).

Лічильник команд (PC) – програмно доступний регістр, в якому зберігається адреса команди, що буде виконуватись у наступному кроці.

2.2. Система команд мікропроцесора.

Усі команди мікропроцесора наведені в табл. 1. Сутність кожної команди зазначена після крапки з комою. Умовні позначення розшифровані в нижній частині таблиці в розділі „Умовні позначки”. Коди команд наведені в табл. 2.

ПРИКЛАД 1. Користуючись табл.1 визначити призначення команди STAX D. У розділі „однобайтові пересилання” знаходимо мнемоніку команди STAX. У відповідному рядку записано STAX YZ⁺; A → M(YZ⁺). В умовних позначках читаємо: YZ⁺ – вміст регістрової пари BC або DE. Запис після крапки з комою означає, що вміст акумулятора переміщується або ще кажуть, завантажується в пам'ять (літера M – від слова memory, що означає пам'ять) за адресою, що знаходиться в регістровій парі BC або DE.

Таким чином, команда STAX D означає пересилання вмісту акумулятора в комірку пам'яті. Адреса цієї комірки міститься в регістровій парі DE.

За таблицею 2 легко визначити шістнадцятирічний код операції команди. Наприклад, код команди STAX D– 12H.

2.3. Приклади складання лінійних програм

Лінійні програми характеризуються послідовним записом і виконанням команд, що вміщуються в цих програмах.

ПРИКЛАД 2. Скласти програму додавання двох шістнадцятирічних чисел 4E и 2F. Результат розмістити в комірці пам'яті з адресою 23FH.

Оскільки операнди задані в явному виді, необхідно, помістити їх в операційні регістри загального призначення. При цьому один з операндів

Табл. 1. Система команд мікропроцесора

Однобайтові пересилання		Двобайтові пересилання	
MOV R1,R; R → R1		SPHL; HL → SP	
MVI R,D8; D8 → R		LXI YZ,D16; D16 → YZ	
STA adr; A → M(adr)		SHLD adr; HL → M(adr); M(adr+1)	
LDA adr; M(adr) → A		LHLD adr; M(adr) → L, M(adr+1) → H	
STAX YZ ⁺ ; A → M(YZ ⁺)		PUSH YZ ⁺⁺ ; YZ ⁺⁺ → M(SP-1); M(SP-2)	
LDAX YZ ⁺ ; M(YZ ⁺) → A		PUSH PSW; A → M(SP-1); F → M(SP-2); SP-2 → SP	
		POP YZ ⁺⁺ ; M(SP), M(SP+1) → YZ ⁺⁺	
		POP PSW; M(SP) → F; M(SP+1) → A; SP+2 → SP	
Команди вводу та виводу		Обмін байтами	
IN N (N) → A		XCHG; HL ↔ DE	
OUT N; A → (N)		XTHL; H ↔ M(SP+1), L ↔ M(SP)	
Арифметичні та логічні операції з одним операндом			
CMC ^{''} ; C → C	CMA; $\bar{A} \rightarrow A$	INR ^{'''} R; R+1 → R	INX YZ; YZ+1 → YZ
STC ^{''} ; 1 → C	DAA ['] ; Десятична корекція	DCR ^{'''} R; R-1 → R	DCX YZ; YZ-1 → YZ
Арифметичні і логічні операції з двома операндами			
ADD ['] R; A+R → A	ADI ['] D8; A+D8 → A	CPI ['] D8; Установка ознак у відпо- CMP ['] R; відності з A-D8 або A-R	
ADC ['] R; A+(R+C) → A	ACI ['] D8; A+(D8+C) → A		
SUB ['] R; A-R → A	SUI ['] D8; A-D8 → A	16-бітові операції	
SBB ['] R; A-(R+C) → A	SBI ['] D8; A-(D8+C) → A		
ANA ['] R; A ∩ R → A	ANI ['] D8; A ∩ D8 → A	DAD ^{''} YZ; HL+YZ → HL	
ORA ['] R; A ∪ R → A	ORI ['] D8; A ∪ D8 → A		
XRA ['] R; A ⊕ R → A	XRI ['] D8; A ⊕ D8 → A		
Команди здвигу вмісту акумулятора		Команди передачі керування	
RLC ^{''} ; Здвиг вліво		PCHL; HL → PC	
RAL ^{''} ; Здвиг вліво через біт ознаки C		JMP adr; adr → PC	
RRC ^{''} ; Здвиг вправо		J-CON adr; adr → PC	
RAR ^{''} ; Здвиг вправо через біт ознаки C			
Спеціальні команди		Команди виводу та повертання з підпрограми	
EI; Дозвіл переривання		CALL adr; PC → M(SP-1)M(SP-2)	
DI; Заборона переривання		C-CON adr; adr → PC	
HLT; Остановка		RST X; PC → M(SP-1)M(SP-2)	
NOP; Холоста операція		де X=0,1,...,7	adr → PC
Формат регістра F		Для кожного X adr, відповідно, рівний 0H; 8H; 10H; 18H; 20H; 28H; 30H; 38H;	
D7 D6 D5 D4 D3 D2 D1 D0		RET; M(SP) M(SP+1) → PC	
S Z 0 AC 0 P 1 C		R-CON; SP+2 → SP	
Умовні позначки			
' – команда діє на всі ознаки регістра F;			
'' – команда діє тільки на ознаку наявності переносу із старшого розряду (C);			
''' – команда діє на всі ознаки регістра F, крім ознаки C;			
R, R1 – вміст регістрів A, B, C, D, E, H, L або комірки пам'яті M(HL)			
YZ – вміст регістрової пари BC, DE, HL або регістру SP (YZ в мнемониці замінюється на B, D, H або SP);			
YZ ⁺ – вміст регістрової пари BC або DE (YZ ⁺ в мнемониці замінюється на B або D);			
YZ ⁺⁺ – вміст регістрової пари BC, DE, HL або PSW (YZ ⁺⁺ в мнемониці замінюється на B, D, H або PSW);			
SP – вміст вказівника стеку перед виконанням команди;			
D8 – 8-розрядний операнд D (вміст другого байту команди);			
(N) – вміст порта вводу або виводу з номером N (N = 0; 1; ...; 255);			
D16 – 16-розрядний операнд (другий та третій байти команди);			
adr – 16-розрядна адреса в трьохбайтовій команді;			
M() – вміст комірки пам'яті (адреса вказується у скобках);			
-CON – частина мнемоники команди, яка визначає умови виконання команди. (-CON в мнемониці замінюється на NZ, Z, NC, C, PO, PE, P або M)			

Таблиця 2. Коди команд мікропроцесора K580BM80

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NOP	LXI B,&	STAX B	INX B	INR B	DCR B	MVI B,#	RLC	-	DAD B	LDAX B	DCX B	INR C	DCR C	MVI C,#	RRC
1	-	LXI D,&	STAX D	INX D	INR D	DCR D	MVI D,#	RAL	-	DAD D	LDAX D	DCX D	INR E	DCR E	MVI E,#	RAR
2	-	LXI H,&	SHLD *	INX H	INR H	DCR H	MVI H,#	DAA	-	DAD H	LHLD *	DCX H	INR L	DCR L	MVI L,#	CMA
3	-	LXI SP&	STA *	INX SP	INR M	DCR M	MVI M,#	STC	-	DAD SP	LDA *	DCX SP	INR A	DCR A	MVI A,#	CMC
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
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
6	MOV H,B	MOV H,C	MOV H,D	MOV H,E	MOV H,H	MOV HL	MOV H,M	MOV H,A	MOV L,B	MOV L,C	MOV LD	MOV L,E	MOV L,H	MOV LL	MOV LM	MOV L,A
7	MOV M,B	MOV M,C	MOV M,D	MOV M,E	MOV M,H	MOV M,L	HLT	MOV MA	MOV A,B	MOV A,C	MOV A,D	MOV A,E	MOV A,H	MOV A,L	MOV A,M	MOV A,A
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
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
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
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
C	RNZ	POP B	JNZ *	JMP *	CNZ *	PUSH B	ADI #	RST O	RZ	RET	JZ *	-	CZ *	CAL L	ACI #	RST 1
D	RNC	POP D	JNC *	OUT N	CNC *	PUSH D	SUI #	RST 2	RS	-	JC *	IN N	CC *	* -	SBI *	RST 3
E	RPO	POP H	JPO *	XTH L	CPO *	PUSH H	ANI #	RST 4	RPE	PCHL	JPE *	XCHG	CPE *	-	XRI #	RST 5
F	RP	POP PSW	JP *	DI	CP *	PUSH PSW	ORI #	RST 6	RM	SPHL	JM *	EI	CM *	-	CPI #	RST 7
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

Примітки:

N – номер порта вводу-виводу;

& – двухбайтовий операнд D16;

* – двухбайтовий операнд adr;

– однобайтовий операнд D8;

PSW – вміст акумулятора та регістра признаков (слово стану процесора).

ПРИКЛАД. Код операції – C3; операція – JMP adr.

перед виконанням додавання обов'язково повинен бути розташований в акумуляторі А. Результат додавання також буде в акумуляторі. Користуючись командою мікропроцесора виконаємо пересилання вмісту акумулятора в комірку пам'яті з адресою 23FЕН.

З урахуванням цього програма прийме вид:

MVI A, 4E; – завантаження числа 4E в регістр А;

MVI C, 2F; – завантаження числа 2F у регістр С;

ADD C ; – додавання вмісту регістра С до вмісту регістра А, результат у регістрі А;

STA 23FE ; – пересилання вмісту регістра А в комірку пам'яті з адресою 23FЕН;

HLT ; – остановка.

Користуючись таблицею 2 виконаємо асемблювання програми і розмістимо її в пам'яті мікро-ЕОМ починаючи з адреси 2150Н. (таблиця 3).

Таблиця 3. Проасембльована програма

Адреса	Код операції	Команда	Коментар
2150	3E	MVI A,4E	завантаження числа 4E в регістр А
2151	4E		
2152	0E	MVI C,2F	завантаження числа 2F у регістр С
2153	2F		
2154	81	ADD C	додавання вмістів регістрів С и А
2155	32	STA 23FE	запис результату в комірку пам'яті з адресою 23FЕН
2156	FE		
2157	23		
2158	76	HLT	остановка

Користуючись командою монітора S, завантажимо програму в пам'ять мікроЕОМ:

S 2150 □ 3E □ 4E □ 0E □ 2F □ 81 □ 32 □ FE □ 23 □ 76.

За допомогою команди монітора G виконаємо програму:

G 2150 □ 2158 CR.

Користуючись командою монітора E перевіримо наявність в акумуляторі результату додавання:

E □

За допомогою команди S перевіримо результат у комірці пам'яті з адресою 23FЕН:

S 23FE □ .

В акумуляторі А и комірці пам'яті 23FЕН повинне бути отримане число $4EH + 2FH = 7DH$.

ПРИКЛАД 3. Скласти вміст двох комірок пам'яті з адресами 2200H і 2210H, з отриманого числа відняти число 1AH, результат проінвертувати і додати до нього одиницю, після чого результат переслати в комірку пам'яті з адресою 2300.

Вихідні дані: в комірку 2200H за допомогою команди S завантажити число 4DH, а в комірку 2210H – число 4AH.

1-й ВАРІАНТ ВИРІШЕННЯ ЗАДАЧІ:

LDA 2200 – завантаження вмісту комірки пам'яті з адресою 2200H в акумулятор;

MOV B, A – пересилання його в регістр B;

LDA 2210 – завантаження вмісту комірки пам'яті з адресою 2210H в акумулятор;

ADD B – додавання до вмісту акумулятора вмісту регістра B;

CMA – інвертування акумулятора (заміна одиниць нулями, а нулів одиницями);

ADI 01 – додаток одиниці до вмісту акумулятора;

STA 2300 – пересилання результату в комірку з адресою 2300H;

HLT – зупинення роботи.

2-й ВАРІАНТ РІШЕННЯ ЗАДАЧІ (з використанням адресації через реєстрову пару HL):

LXI H,2200 – завантаження в реєстрову пару HL адреси першого операнда;

LDA 2210 – завантаження в акумулятор другого операнда;

ADD M – додавання до вмісту акумулятора вмісту комірки пам'яті, адреса якої знаходиться в реєстровій парі HL (тобто 2200H);

SUI 1A – вирахування з вмісту акумулятора числа 1AH;

CM A – інвертування акумулятора;

INR A – інкремент (збільшення на одиницю) вмісту акумулятора;

STA 2300 – пересилання результату в комірку пам'яті з адресою 2300H;

HLT – зупинення роботи.

3. Порядок виконання роботи.

1. По конспекту лекцій і літературних джерелах вивчити архітектуру і систему команд мікропроцесора.

2. Вивчити методичні вказівки до виконання даної лабораторної роботи, розібрати наведені приклади складання програм.

3. Відповідно до індивідуального завдання скласти програму його виконання і проасемблювати її.

4. Виконати вручну розрахунки завдання у двоїчній формі. Результат перевести у 16-річну форму. Вміст комірок пам'яті, які використовуються у розрахунках приймати довільно.

5. Занести прийняті значення вмісту комірок пам'яті, що використано при розрахунках у пам'ять мікроЕОМ.

6. Занести складену програму у пам'ять мікроконтролера „Електроніка К1-20”, вирішити завдання за його допомогою і отримати результат. Завдання вважається виконаним, якщо результат розрахунків і результат виконання програми на мікроконтролері співпадають.

7. Показати результати виконання програми і зроблені розрахунки викладачу та отримати його підпис, який свідчить про виконання завдання.

4. Зміст звіту.

Звіт повинний містити:

- найменування і мету роботи;
- умови завдання свого варіанту;
- проасембльоване завдання свого варіанту, оформлене у вигляду таблиці;
- розрахунки завдання у двоїчній формі і результат у 16-річній формі.

5. Контрольні питання.

1. Скільки розрядів мають регістри загального призначення?
2. Який спосіб використовується, якщо у регістрах загального призначення потрібно розмістити 16-розрядний операнд?
3. Скільки у мікропроцесорі серії КР580 регістрів загального призначення?
4. Що таке стек та яке його призначення?
5. Яке призначення вказівника стеку?
6. Яке призначення лічильника команд та скільки розрядів він має?
7. З скількох байтів може складатися команда мікропроцесора?
8. Що таке мнемоніка команди і для чого вона використовується?
9. Що означає термін „операнд”? Що може бути операндом?
10. З яких двох частин складається команда? Яке призначення кожної частини?
11. Призначення регістру признаків та які ознаки в ньому зберігаються?
12. Які функції виконують команди пересилання?
13. Які функції виконують арифметичні команди?
14. Які функції виконують логічні команди?
15. Що може бути умовою переходу в іншу частину програми?
16. Який формат має регістр признаків?

6. Рекомендована література.

1. Балашов Е.П., Пузанков Д.В. Мікропроцесори і мікропроцесорні системи. -М.: Радіо і зв'язок, 1981.-с.53-55,119-125.

2. Вершинин О.Е. Застосування мікропроцесорів для автоматизації технологічних процесів. -Л.: Энергоатомиздат, 1986.-с.102-130.

7. Індивідуальні завдання

№ варіанту	Індивідуальне завдання
1	Вміст комірки пам'яті 2200Н скласти з числом 2FH і отриманий результат відняти з вмісту комірки 2201Н
2	Скласти вміст комірок пам'яті 2200Н і 2201Н і з отриманого результату відняти число 1ВН
3	З вмісту комірки 2200Н відняти число 01 і до отриманого результату додати одиницю
4	Проінвертувати вміст комірки 2200Н і до отриманого результату додати одиницю, після чого отримане число скласти з вмістом комірки 2201Н
5	Скласти число 4FH із проінвертованим вмістом комірки пам'яті 2200Н і від отриманої суми відняти одиницю
6	Проінвертувати вміст комірок пам'яті 2200Н і 2201Н після чого скласти їх вміст
7	З вмісту комірки пам'яті 2200Н відняти вміст комірки 2201Н, результат проінвертувати
8	Вміст комірки 2200Н переслати в комірку 2201Н, потім проінвертувати суму вмісту цих комірок
9	Скласти два числа, розташовані в комірках 2200Н і 2201Н, до результату додати число 03
10	Проінвертувати вміст комірки 2201Н, додати до результату 1 і скласти результат з числом F0H
11	До вмісту комірки з адресою 2200Н додати одиницю, проінвертувати результат і відняти з нього число 1ВН
12	Скласти проінвертований вміст комірок пам'яті 2200Н і 2201Н, до результату додати число 1СН
13	З вмісту комірки пам'яті 2200Н відняти одиницю, результат проінвертувати і скласти з числом 2DH
14	До проінвертованого вмісту комірки 2200Н додати 1 і з результату відняти вміст комірки -2201Н
15	З числа FFH відняти проінвертований вміст комірки 2201Н, результат проінвертувати
16	З числа FFH відняти число 05, від результату відняти проінвертований вміст комірки 2201Н
17	Із суми числа OFH і проінвертованого вмісту комірки 2201Н відняти число 4EH.