

Лабораторна робота №6

Тема: **ВИВЧЕННЯ КЕРУЮЧОЇ ПРОГРАМИ ЕМУЛЯТОРА SCM ТА ВИЗНАЧЕННЯ КОМАНД ПЕРЕДАЧІ ДАНИХ**

Мета роботи: отримання практичних навичок роботи з програмою емулятора SCM та по використанню команд передачі даних в програмах для OEBM

1 Теоретичні відомості

1.1 Програмно-логічна модель МК-51 і робота з нею

Програма SCM (Single-Chip Machine) являє собою систему моделювання однокристальних мікроконтролерів.

Система моделювання Single-Chip Machine 1.22 призначена для дослідження поведінки внутрішніх і зовнішніх сигналів зазначених мікросхем.

Програма SCM (Single-Chip Machine) виконана у вигляді незалежного запускаемого модуля, працездатного під управлінням операційної системи MS Window NT / XP. SCM включає засоби налагодження та редагування програм на асемблері. Виконання програми користувача здійснюється з максимальним наближенням до дійсності за допомогою імітаційної моделі. Крім того, користувачеві надається такі засоби, як тимчасові діаграми внутрішніх і зовнішніх сигналів, імітація зовнішніх сигналів, можливість зміни значень вузлів МК в процесі роботи моделі та ін

Користувач набирає програму в редакторі програм, потім натискає кнопку "компіляція". Текст програми переводиться в машинні коди і записується в однойменний файл (з вихідним текстом) з розширенням ".MPM". Розширення ".MPM", розшифровується як Microcontroller Program Memoгу, проте існує стандартний формат подання пам'яті програм - так званий формат HEX.

Програма забезпечує: виконання прикладної програми для OEBM в покроковому режимі, в режимі прогону з зупинкою по контрольних точках;

доступ до всіх внутрішніх ресурсів OEVM, зовнішньої пам'яті програм і даних.

1.2 Робочий простір емулятора SCM (Single-Chip Machine)

Робочий простір емулятора SCM (Single-Chip Machine представлений на рис.6.1.

Для створення нової програми або редагування готової програми необхідно натиснути на кнопку Редактор.

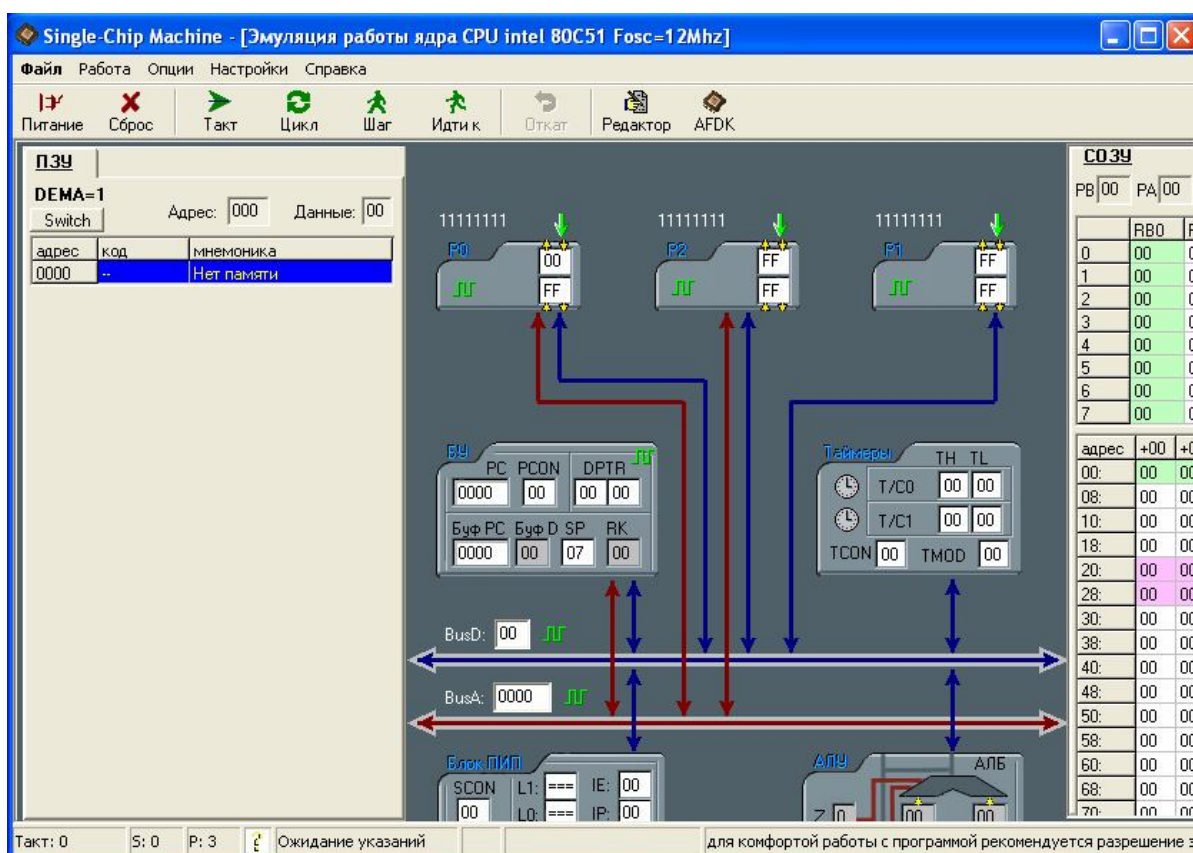


Рис.6.1 - Робочий простір емулятора SCM

Після натиснення кнопки Редактор з'явиться вікно Редактора, зовнішній інтерфейс якого представлений на рис.6.2. В білому просторі, де мигає курсор, необхідно набирати або редагувати текст програми рис.6.3. Після набору або редагування програми необхідно зберегти файл або запустити програму кнопкою Виконати програму (F9), в свою чергу вона також запропонує зберегти програму.

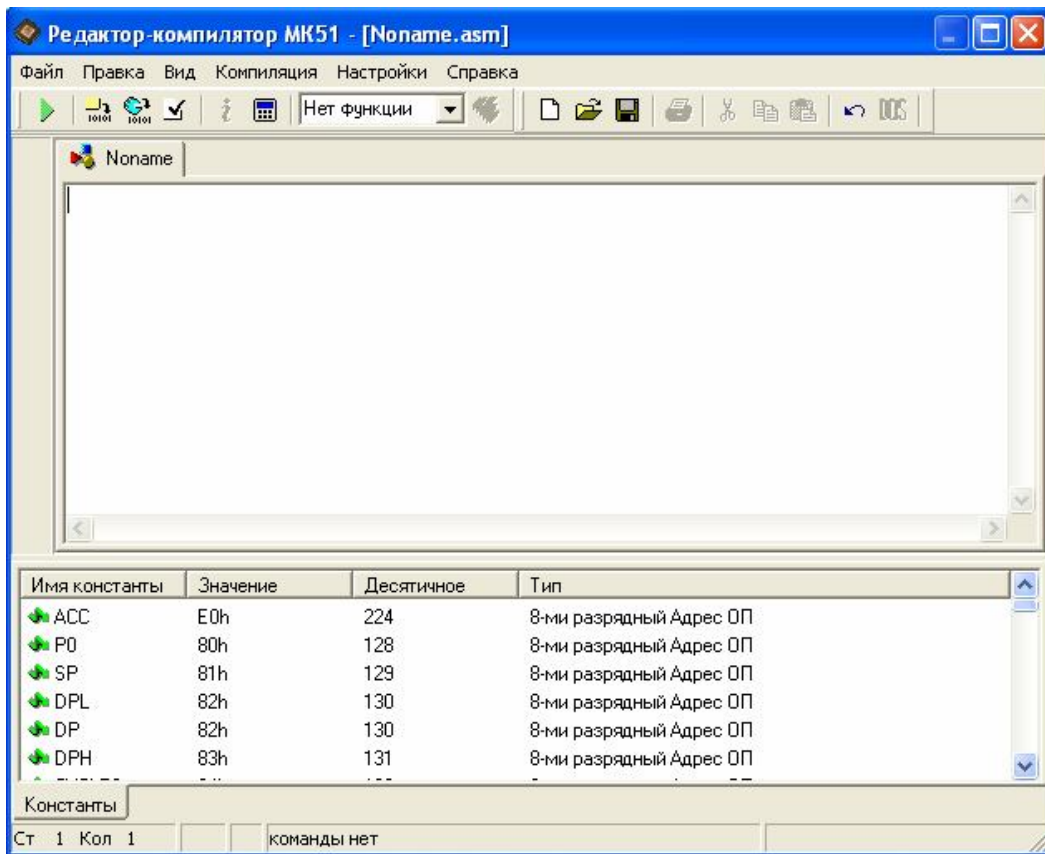


Рис.6.2 – Вікно редактора

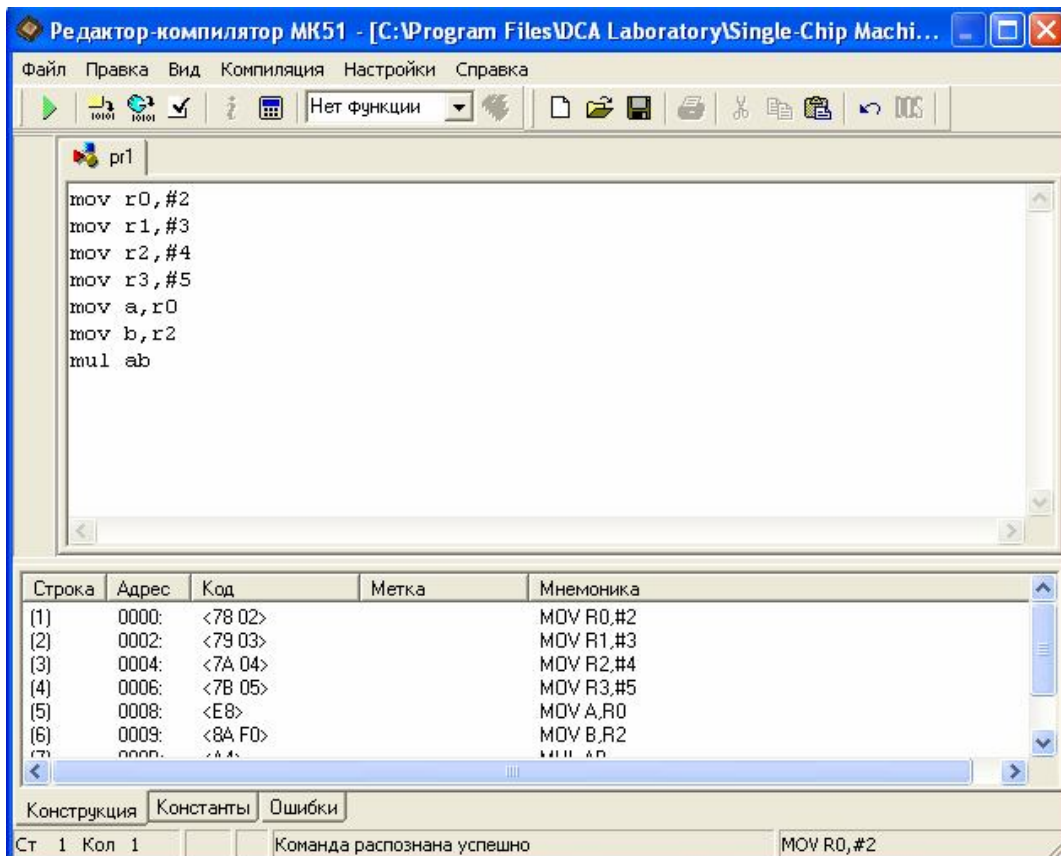


Рис.6.3

Після збереженні програми натиснути кнопку Виконати програму (F9). Якщо не існують помилок в програмі, вона запуситься. Якщо помилки присутні, потрібно їх усунути. Також можливо скористатися кнопкою Компіляція (Ctrl+F9) до запуску програми.

Після запуску програми в вікні Редактора, емулятор запускає програму в режимі емуляції (рис.6.4). З рис.6.4 видно, що в ПЗУ записалась наша програма. В свою чергу бачимо, адрес, код команд в HEX системі та мнемоніку команд.

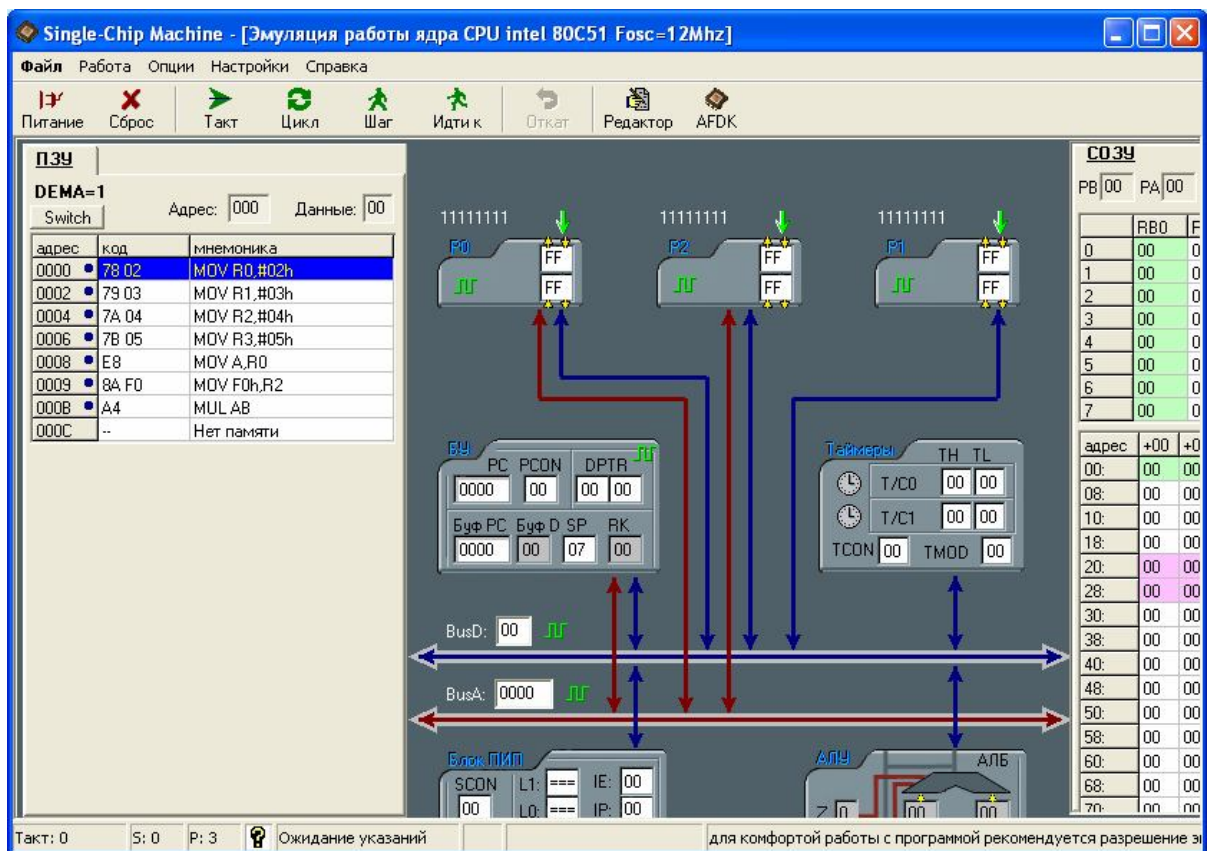


Рис.6.4

Емуляції програму можна запусити кнопкою Такт (F7) – просування на один такт, кнопкою Цикл (Ctrl+F7) – просування на один цикл, кнопкою Крок (F8) – просування на один крок, кнопкою Йти до (F4) – запуск програми до зазначеної команди.

Запустимо програму для емуляції, наприклад кнопкою Цикл (Ctrl+F7) (рис.6.5). З рис.6.5 відслідковуючи HEX код програми, побачимо, що в БУ під адресою 0002 записано 02 07 79, а в АЛУ записалося число 2 з чого видно, що програма просунулась на один цикл, також число 2 записалося в СО ЗУ.

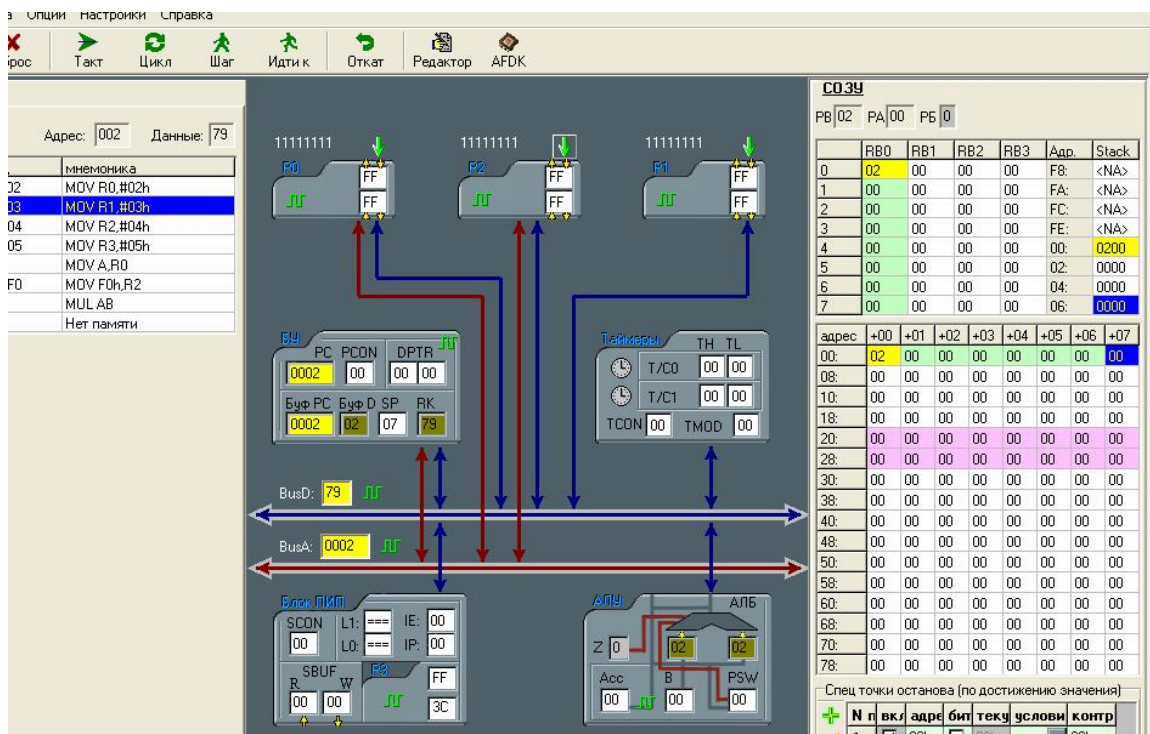


Рис.6.5

Пройшовши програму кнопкою Цикл, в кінці побачимо в АЛУ результат 8 рис.6.6, виходячи з коду програми результат вірний (число в регістрі А дорівнює 8, а в - В дорівнює 4, а їх добуток MUL AB дорівнює 32).

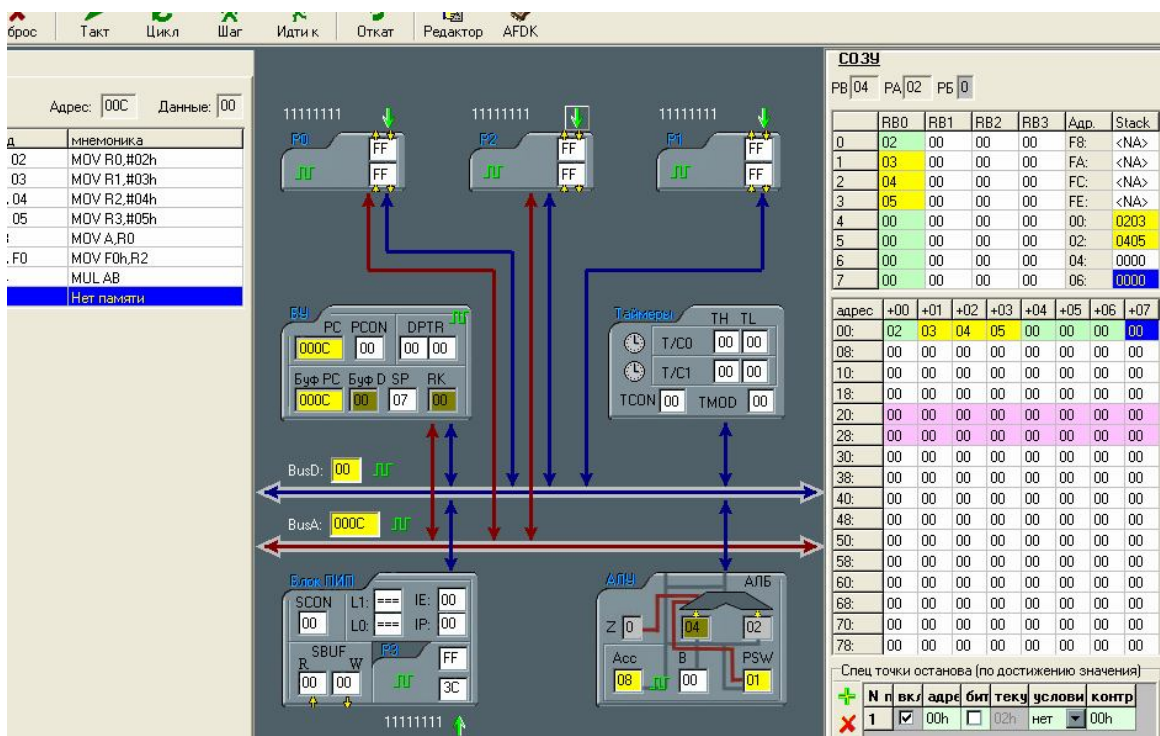


Рис.6.6

2 Завдання

Задача1. Розробіть алгоритм і напишіть програму на мові Асемблера для обчислення функцій, наведеної у табл.6.1 (A, R1, R2, R3, R4, R5, R6 – регістри загального призначення мікропроцесора K1816BE51; M1, M2 – операнди, розташовані у пам'яті; M3 – результат обчислень, розташований у пам'яті). Операнди у регістрах загального призначення та в пам'яті – 8-розрядні (без знаку). Адреси операндів M1 і M2 та результату M3 виберіть самостійно.

Таблиця 6.1

Варіант	Функція
1	$M3=(R4-R5)+(R6-(50)_{16})+(M1+M2)$
2	$M3=(M1-(73)_{16})+M2-(R3+R6)-R5$
3	$M3=R6-((43)_{16}+M1)-R3-(M2+R4)$
4	$M3=(R3+R4)-R5-(M1+(53)_{16})+M2$
5	$M3=M1+(M2-R5)+(57)_{16}+(R6+R3)$
6	$M3=(M2-M1)+R3-((17)_{16}+R4)-R5$
7	$M3=M1+(R3+R4)-(R5-R6)-(83)_{16}$
8	$M3=R5+(R4-M1)+(25)_{16}-R3+R6$
9	$M3=R4+((62)_{16}+M1)+R5-(M2-R6)$
10	$M3=(R6+R5)-(M1+(32)_{16})-(M2-R4)$
11	$M3=(M1-M2)-R3+(R5+(12)_{16})+R6$
12	$M3=(38)_{16}-(R3-R4)+(M1+M2)-R6$
13	$M3=(M1-M2)+(R4-(31)_{16})+(R3+R5)$
14	$M3=M2+(M1-(93)_{16})-(R3+R4)+R6$
15	$M3=R3-(M1+R6)-(48)_{16}+(R4+M2)$
16	$M3=((70)_{16}-R3)-R4+(R5+R6)+M1$
17	$M3=(R6+M1)-((96)_{16}+R3)+(R4+R5)$
18	$M3=(R4-M1)+(R6-(50)_{16})+(R5+M2)$
19	$M3=(R3+R6-(73)_{16})+M2-(M1+(7A)_{16})-R5$

20	$M3=M1-((43)_{16}+R1)-R3-(M2+R2)$
21	$M3=(R3+(5A)_{16})-R5-(M1+R1-R4)+M2$
22	$M3=R4+(M2-R5)+(57)_{16}+(R6-M1)$
23	$M3=M1+(R3+R4)-(R1+R2+(0A)_{16})-(83)_{16}$
24	$M3=R5+(A-M1)+((77)_{16}-R3)+R2$
25	$M3=R4+((22)_{16}+M2)+R5-(R6+M2)$
26	$M3=(R6+R4)-(M1+(32)_{16})-(R5-M2)$
27	$M3=(M1+M2)-R3+(R5-(21)_{16})+A$
28	$M3=(38)_{16}-(R3-M2)+(M1+R4)-R6$
29	$M3=(M1-M2)+(R4-(4A)_{16})+(R6-R5)$
30	$M3=M1+(M2+(93)_{16})-(R3-R4)+R6$

3 Порядок виконання роботи

3.1 Вивчення програми емулятора SCM.

3.2 Запуск програми. Знайомство і вивчення основного меню програми, ознайомлення з можливостями і способами редагування внутрішніх і зовнішніх ресурсів емулятора.

3.3 Вивчення процесу програмування мікроконтролера, введення та налагодження програм та їх виконання.

3.4 Запуск програми - емулятора і практичне закріплення отриманих знань.

3.5 Ознайомлення з командами передачі даних.

3.6 Вивчення особливостей організації пам'яті програм і даних в ОЕВМ типу K1816BE51.

3.7 Вивчення команд пересилки між регістрами мікроконтролера, регістрами і таймерами, акумулятором і портами.

3.8 Складання програм на мові Асемблера ОЕВМ, що реалізують різні способи пересилки і адресації.

4 Зміст звіту

- 4.1 Назва та мета роботи
- 4.2 Лістинг налагодженої програми (з коментарями).
- 4.3 Результат виконання програми
- 4.4 Висновки по роботі

5 Контрольні питання

- 5.1 Архітектура мікроконтролера. Основні вузли і блоки.
- 5.2 Максимальний обсяг ОЗУ і ПЗУ, використовуваний даними контролером.
- 5.3 Способи редагування за допомогою програми - емулятора ОЗУ, ПЗУ, системних ресурсів.
- 5.4 Команди виконання програми. Результати виконання.
- 5.5 Можливості налагодження МПУ.
- 5.6 Між якими частинами мікроконтролера здійснюється передача даних.
- 5.7 Методи адресації, що використовуються в ОЕВМ.
- 5.8 Типи портів мікроконтролера.
- 5.9 Як відбувається адресація зовнішнього ОЗУ і ПЗУ?
- 5.10 Відобразити схему підключення зовнішнього ОЗУ і ПЗУ до мікроконтролера. Які сигнали для цього використовуються?

ДОДАТОК

Таблиця 6.2 - Система команд

Мнемокод	КОП	Мнемокод	КОП	Мнемокод	КОП
ACALL 0xxH	11	AJMP 5XXH	A1	DA A	D4
ACALL 1xxH	31	AJMP 6XXH	C1	DEC A	14
ACALL 2xxH	51	AJMP 7XXH	E1	DEC ad	15
ACALL 3xxH	71	ANL A, ad	55	DEC R0	18
ACALL 4xxH	91	ANL A, R0	58	DEC R1	19
ACALL 5xxH	B1	ANL A, R1	59	DEC R2	1A
ACALL 6xxH	D1	ANL A, R2	5A	DEC R3	1B
ACALL 7xxH	F1	ANL A, R3	5B	DEC R4	1C
ADD A, ad	25	ANL A, R4	5C	DEC R5	1D

ADD A, R0	28	ANL A, R5	5D	DEC R6	1E
ADD A, R1	29	ANL A, R6	5E	DEC R7	1F
ADD A, R2	2A	ANL A, R7	5F	DEC @R0	16
ADD A, R3	2B	ANL A, @R0	56	DEC @R1	17
ADD A, R4	2C	ANL A, @R1	57	DIV AB	84
ADD A, R5	2D	ANL A, #d	54	DJNZ ad, rel	D5
ADD A, R6	2E	ANL ad, A	52	DJNZ R0, rel	D8
ADD A, R7	2F	ANL ad, #d	S3	DJNZ R1, rel	D9
ADD A, @R0	26	ANL C, bit	82	DJNZ R2, rel	DA
ADD A, @R1	27	ANL C, /bit	BO	DJNZ R3, rel	DB
ADD A, #d	34	CJNE A, ad, rel	B5	DJNZ R4, rel	DC
ADDC A, ad	35	CJNE A, #d, rel	B4	DJNZ R5, rel	DD
ADDC A, R0	38	CJNE R0, #d, rel	B8	DJNZ R6, rel	DE
ADDC A, R1	39	CJNE R1, #d, rel	B9	DJNZ R7, rel	DF
ADDC A, R2	3A	CJNE R2, #d, rel	BA	INC a	04
ADDC A, R3	3B	CJNE R3, #d, rel	BB	INC ad	05
ADDC A, R4	3C	CJNE R4, #d, rel	BC	INC DPTR	A3
ADDC A, R5	3D	CJNE R5, #d, rel	BD	INC R0	08
ADDC A, R6	3E	CJNE R6, #d, rel	BE	INC R1	09
ADDC A, R7	3F	CJNE R7, #d, rel	BF	INC R2	0A
ADDC A, @R0	36	CJNE @R0, #d, rel	B6	INC R3	0B
ADDC A, @R1	37	CJNE @R1, #d, rel	B7	INCR4	0C
ADDC A, #d	24	CLR A	E4	INC R5	0D
AJMP 0XXH	01	CLR bit	C2	INC R6	0E
AJMP 1XXH	21	CLR C	C3	INC R7	0F
AJMP 2XXH	41	CPL A	F4	INC @R0	06
AJMP 3XXH	61	CPL bit	B2	INC @R1	07
AJMP 4XXH	81	CPL C	B3	JB bit, rel	20
				JBC bit, rel	10

Продовження таблиці 5.3

Мнемокол	КОП	Мнемокол	КОП	Мнемокол	КОП
JC rel	40	MOV ad, @R0	86	MOV R7, ad	AF
JMP @A+DPTR	73	MOV ad, @R1	87	MOV R7, #d	7F
JNB bit, rel	30	MOV ad, #d	75	MOV @R0, A	F6
JNC rel	50	MOV ad, ads	85	MOV @R0, ad	A6
JNZ rel	70	MOV bit, C	92	MOV @R0, #d	76
JZ rel	60	MOV C, bit	A2	MOV @R1, A	F7
LCALL ad16	12	MOV DPTR, #dl6	90	MOV @R1, ad	A7
LJMP ad 16	02	MOV R0, A	F8	MOV @R1, #d	77
MOV A, ad	E5	MOV R0, ad	A8	MOVC A,	93
MOV A, R0	E8	MOV R0, #d	78	MOVC A, @+PC	83
MOV A, R1	E9	MOV R1, A	F9	MOVX A,	E0
MOV A, R2	EA	MOV R1, ad	A9	MOVX A, @R0	E2
MOV A, R3	EB	MOV R1, #d	79	MOVX A, @R1	E3
MOV A, R4	EC	MOV R2, A	FA	MOVX @DPTR,	F0
MOV A, R5	ED	MOV R2, ad	AA	MOVX @R0, A	F2
MOV A, R6	EE	MOV R2, #d	7A	MOVX @R1, A	F3
MOV A, R7	EF	MOV R3, A	FB	MUL AB	A4
MOV A, @R0	E6	MOV R3, ad	AB	NOP	00
MOV A, @R1	E7	MOV R3, #d	7B	ORL A, ad	45

MOV a , #d	74	MOV R4, A	FC	ORL A , R0	48
MOV ad , A	F5	MOV R4 , ad	AC	ORL A, R1	49
MOV ad , R0	88	MOV R4, #d	7C	ORL A, R2	4A
MOV ad , R1	89	MOV R5, A	FD	ORL A , R3	4B
MOV ad , R2	8A	MOV R5 , ad	AD	ORL A, R4	4C
MOV ad, R3	8B	MOV R5 , #d	7D	ORL A, R5	4D
MOV ad , R4	8C	MOV R6 , A	FE	ORL A, R6	4E
MOV ad , R5	8D	MOV R6, ad	AE	ORL A, R7	4F
MOV ad , R6	8E	MOV R6, #d	7E	ORL A, @R0	46
MOV ad , R7	8F	MOV R7 , A	FF	ORL A , @R0	47
ORL A, #d	44	RRC A	13	SUBB A , R7	9F
ORL ad , A	42	SETB bit	D2	SUBB A , @R0	96
ORL ad , #d	43	setb c	D3	SUBB A, @R1	97
ORL C , bit	72	SJMP rel	80	SWAP A	C4
ORL C, /bit	AO	SUBB A, ad	95	XCH A , ad	C5
POP ad	DO	SUBB A, R0	98	XCH A, R0	C8
PUSH ad	CO	SUBB A, R1	99	XCH A, R1	C9
RET	22	SUBB A , R2	9A	XCH A , R2	CA
RETI	32	SUBB A , R3	9B	XCH A , R3	CB
RL A	23	SUBB A, R4	9C	XCH A , R4	CC
RLC A	33	SUliB A , R5	9D	XCH A , R5	CD
RR A	03	SUBB A, R6	9E	XCH A , R6	CE
XCH A, R7	CF	XRL A, R1	69	XRL A , R7	6F
XCH A, @R0	06	XRL A,	6A	XRL A , @R0	66
XCH A, @R1	C7	XRL A, R3	6B	XRL A, @R1	67
XCHD A, @R0	D6	XRL A, R4	6C	XRL A, #d	64
XCHD A, @R1	D7	XRL A, R5	6D	XRL ad , A	62
XRL A, ad	65	XRL A , R6	6E	XRL ad, #d	63
XRL A, R0	68				