

Кафедра автоматизації
технологічних процесів
і виробництв

Лабораторна робота № 4

з курсу

Проектування мікропроцесорних
систем керування технологічними
процесами

Програмування мікроконтролера
MCS51 з використанням програмної
моделі EdSim51.

Команди звернення до пам'яті
програм MCS51.

Вивід інформації на 7-сегментний
дисплей

Методичні вказівки для виконання лабораторної роботи № 4 «Програмування мікроконтролера MCS51 з використанням програмної моделі EdSim51. Команди звернення до пам'яті програм MCS51. Вивід інформації на 7-сегментний дисплей» з курсу «Проектування мікропроцесорних систем керування технологічними процесами»/Укл.: Медвідь В.Р., Пісьціо В.П. - Тернопіль ТНТУ, 2023 - 11 с.

Розглянуто і затверджено на засіданні кафедри автоматизації технологічних процесів і виробництв (протокол № 1 від 30.08.2023 року)

Лабораторна робота № 4

Програмування мікроконтролера MCS51 з використанням програмної моделі EdSim51. Команди звернення до пам'яті програм MCS51. Вивід інформації на 7-сегментний дисплей.

1. Теоретичні відомості

Часто (при реалізації функціональних залежностей, при роботі з експериментальними даними) необхідно мати в пам'яті програм таблиці готових рішень. Для можливості роботи з такими таблицями, що зберігаються в РПП і ЗПП, є спеціальні команди звернення до пам'яті програм - MOVC A, @A+PC та MOVC A, @A+DPTR.

Використання цих команд приведено в Прикладі 1 в підпрограмі обчислення синуса кута X (X змінюється в межах від 0 до 89 градусів з дискретністю, наприклад, 10°).

Найбільш швидко обчислення функції можна отримати шляхом вибірки готового значення синуса з таблиці. Кожен байт таблиці буде містити дробову частину двійкового представлення синуса. Вихідним параметром для підпрограми служить значення кута X, що знаходиться в акумуляторі. Поточне значення синуса записується в регістр R3.

Приклад 1

```
MOV DPTR,#30H          ; Обчислення SIN (x) по таблиці значень
                        ; вхід: X в A (акумуляторі) в межах від 0 до 89 градусів,
                        ; встановити початкову адресу для таблиці синусів
SINX:
INC A                  ; інкремент акумулятора
MOV R2,A              ; зберегти поточне значення A
MOVC A,@A+DPTR       ; завантажити значення синуса з таблиці в акумулятор
MOV R3,A              ; зберегти в R3 значення синуса
MOV A,R2              ; відновити значення A для адресації таблиці синусів
JMP SINX              ; перейти за міткою SINX
                        ; таблиця значень синусів

ORG 30H
DB 0                  ; sin (0) = 0
DB 00100111B         ; sin (10) = 0.156
DB 01011000B         ; sin (20) = 0.342
.....
DB 11111110B         ; sin (80) = 0.985.
DB 11111111B         ; sin (89) = 0.999.
```

Інкремент акумулятора ненобхідний для адресації наступного значення з таблиці синусів в області постійної пам'яті.

2. Завдання

1. Дослідити і виконати програму зведення в квадрат числа (Завдання 1).
2. Дослідити і виконати програму виводу на семисегментний дисплей інформації (Завдання 2).

Завдання 1 та Завдання 2 виконувати з використанням програмної моделі відповідно до «Варіантів індивідуальних завдань»:

Завдання 1

```
Програма зведення в квадрат числа
MOV 00H, #04H        ; пересилання числа 04H для зведення в квадрат в комірку
                        ; пам'яті РЗП з адресою 0x00
ACALL SQUARING_UP    ; виклик підпрограми
SQUARING_UP:         ; підпрограма
```

```

MOV A, 00H      ; пересилання числа з комірки пам'яті РЗП з адресою 0x00H
                ; в акумулятор А
MOV B, A        ; пересилання числа з акумулятора А в регістр В
MUL AB          ; обчислення добутку двох цілих беззнакових чисел, що
                ; зберігаються в акумуляторі А (молодший байт результату)
                ; та регістрі В (старший байт результату)
ACALL RESULT    ; виклик підпрограми
RESULT:
MOV 10H, B      ; пересилання результату з регістра В в комірку пам'яті РЗП
                ; з адресою 0x10
MOV 11H, A      ; пересилання результату з акумулятора А в комірку пам'яті
                ; РЗП з адресою 0x11
END             ; директива, що повідомляє про кінець програми

```

Приклад 2

Програма виводу інформації на 7-сегментний дисплей

Позначення сегментів індикатора дисплею показано на рис. 1,а.

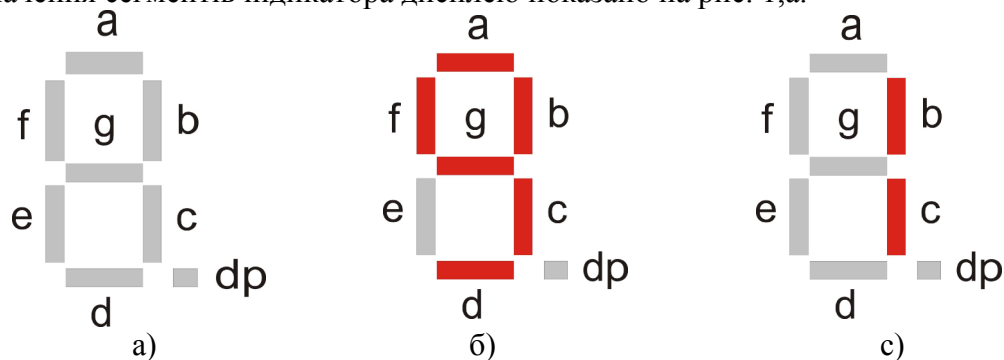


Рис. 1. Позначення сегментів розряду семисегментного індикатора

Відповідно до схеми електричної принципової, роботу якої симулює EdSim51 (рис. 2), сегменти всіх чотирьох його розрядів з'єднані між собою паралельно і під'єднані до ліній порту P1 мікроконтролера MCS51 наступним чином:

Сегменти дисплею: a b c d e f g dp
Лінії порту P1: P1.0 P1.1 P1.2 P1.3 P1.4 P1.5 P1.6 P1.7

Кожен сегмент дисплею запалюється логічним «0».

Щоб вивести на один розряд дисплею цифру «9» (рис. 1,б), необхідно записати в порт P1 наступний байт:

a	b	c	d	e	f	g	dp	
P1.0	P1.1	P1.2	P1.3	P1.4	P1.5	P1.6	P1.7	(1)
0	0	0	0	1	0	0	1	

Якщо на дисплей виводиться «1», то потрібно сформувати байт на лініях порту P1 таким:

a	b	c	d	e	f	g	dp
P1.0	P1.1	P1.2	P1.3	P1.4	P1.5	P1.6	P1.7
1	0	0	1	1	1	1	1

Якщо потрібно засвітити крапку («dp»), необхідно на лінію P1.7 подати логічний «0».

Потрібний розряд дисплею вибирається двійковим кодом на двох лініях P3.4, P3.3 порту P3, які з'єднані з виходом дешифратора «decoder», та транзисторними ключами, через які подається живлення на чотири розряди дисплею DISP 0...DISP 3 (рис. 2). Сегменти вибраного через порт P3 розряду засвітяться, якщо на відповідному виході дешифратора з'явиться логічний «0».

Для вводу в порт P1 потрібного числа використовуються перемикачі з «Switch Bank» симулятора. Натисканням перемикача з позначенням «0» на лінію порту P2.0 подається логічний нуль. Натискання на перемикач «1» подає логічний «0» на лінію P2.1 і т.д.

Одночасно з виводом числа на дисплей засвічується світлодіод з лінійки світлодіодів «LADs» симулятора, що під'єднаний до лінії порту P1, на який виводиться логічний нуль. Світлодіод нульового розряду знаходиться вгорі лінійки.

Завдання 2

Вивести на семисегментний дисплей чотиризначне число відповідно до заданого варіанту завдання в наступній послідовності:

1. Записати задане десяткове число у вигляді чотирьох байтів коду для виводу на семисегментний дисплей по аналогії з виразом (1) Прикладу 2.

2. Скопіювати та завантажити приведену нижче програму в симулятор EdSim51. Асемблювати програму, натиснувши кнопку «*Assm*» симулятора.

3. Почати виконання програми в автоматичному режимі, натиснувши один раз кнопку «*Run*» на панелі коду симулятора.

4. Ввести код для виводу в старший розряд дисплею DISP 3 за допомогою перемикачів «*Switch Bank*». Натиснута курсором кнопка перемикача відповідає логічному «0» на вході порту P2.

5. Натиснути курсором на клавіатурі симулятора «*Keypad*» клавішу «*», після чого натиснути на ній ще раз, щоб відпустити її. Одночасно на семисегментному індикаторі засвітиться число, яке виводиться, а на лінійці світлодіодів LEDs – його код.

6. Ввести за допомогою перемикачів код числа наступного розряду і повторити дії з п.4 та п.5. На дисплей і світлодіоди виведеться значення наступного числа і т.д.

7. Після введення числа останнього числа молодшого розряду (DISP 0) і виведення його на дисплей, знову натиснути на клавіатурі симулятора EdSim51 клавішу «*».

Після її натискання на чотири розряди дисплею послідовно, починаючи з старшого (DISP 3), буде постійно виводитися введене чотиризначне число.

```

setb p3.3           ; вибирається індикатор старшого розряду дисплею DISP 3
setb p3.4
clr p0.0           ; встановлюється на лінії P0.0 логічний «0». Якщо на клавіатурі
                  ; симулятора буде натиснута клавіша «*», на лінії P0.6 також
                  ; з'явиться логічний «0»
m1:                ; цикл, в якому перевіряється, чи клавіша «*» натиснута
jb p0.6,m1         ; якщо натиснута, то код числа для виводу в розряд DISP 3
mov r3,p2         ; завантажується з перемикачів Switch Bank, під'єднаних до порту P2,
                  ; в регістр R3
mov p1,r3         ; та з регістра R3 через порт P1 виводиться на розряд дисплею DISP 3
m11:
jnb p0.6,m11      ; перевіряється, чи клавіша «*» відпущена
setb p0.0         ; на лінії P0.0 встановлюється логічна «1»

clr p3.3         ; вибирається індикатор розряду DISP 2
clr p0.0         ; встановлюється на лінії P0.0 логічний «0»
m2:                ; цикл, в якому перевіряється, чи клавіша «*» натиснута
jb p0.6,m2       ; якщо натиснута, то код числа для виводу в розряд DISP 2
mov r4,p2       ; завантажується з перемикачів Switch Bank, під'єднаних до порту P2,
                  ; в регістр R4
mov p1,r4       ; та з регістра R4 через порт P1 виводиться на розряд дисплею DISP 2
m21:
jnb p0.6,m21    ; перевіряється, чи клавіша «*» відпущена
setb p0.0       ; на лінії P0.0 встановлюється логічна «1»

setb p3.3       ; вибирається індикатор розряду DISP 1

```

```

clr p3.4
clr p0.0
m3:
jb p0.6,m3
mov r5,p2
mov p1,r5
m31:
jnb p0.6,m31
setb p0.0

```

```

clr p3.3 ; вибирається індикатор розряду DISP 0
clr p3.4
clr p0.0
m4:
jb p0.6,m4
mov r6,p2
mov p1,r6
m41:
jnb p0.6,m41
setb p0.0

```

; підпрограма засвічування на дисплеї чотирьох розрядів числа
; відповідно до завдання. Для його виведення потрібно натиснути
; клавішу «*»

```

mov p1,#0FFh
clr p0.0
m5:
jb p0.6,m5
start:
SETB P3.3
SETB P3.4
MOV P1, R3
CALL delay

```

; встановлюється на лінії P0.0 логічний «0»
; цикл, в якому перевіряється, чи клавіша «*» натиснута

; вмикається розряд DISP 3 дисплею

; виводиться зображення числа з регістра R3 на дисплей DISP 3
; перейти на підпрограму часової затримки

```

CLR P3.3
MOV P1, R4
CALL delay

```

; вмикається розряд DISP 2 дисплею

; виводиться зображення числа з регістра R4 на дисплей DISP 2
; перехід на підпрограму часової затримки

```

SETB P3.3
CLR P3.4
MOV P1, R5
CALL delay

```

; вмикається розряд DISP 1 дисплею

; виводиться зображення числа з регістра R5 на дисплей DISP 1
; перехід на підпрограму часової затримки

```

CLR P3.3
CLR P3.4
MOV P1, R6
CALL delay
JMP start

```

; вмикається розряд DISP 0 дисплею

; виводиться зображення числа з регістра R6 на дисплей DISP 0
; перехід на підпрограму часової затримки

```

delay:
MOV R0, #200
DJNZ R0,$
RET

```

; підпрограма часової затримки

; цикл, якщо R0 не нуль

; повернення з підпрограми

\$ (знак доллара) позначає поточне значення лічильника команд PC.

Підключення семисегментного дисплею в електричній схемі, яку симулює EdSim51, показано на рис. 2.

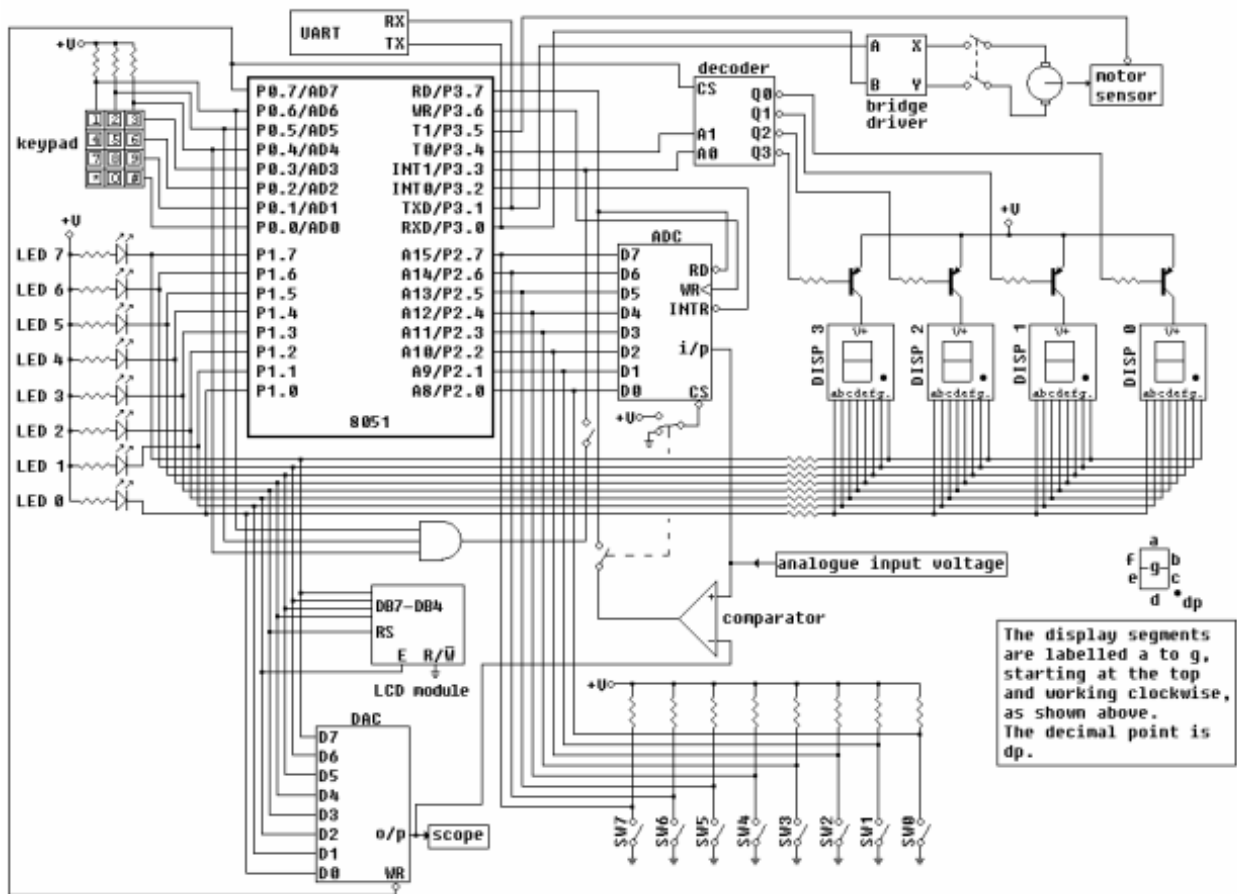


Рис. 2 Схема електрична принципова підключення семисегментного дисплею до MCS51

Варіанти індивідуальних завдань

№	Зміст індивідуального завдання
1	<p>1. Виконати програму зведення в квадрат (Завдання 1) для числа 05H. Записати результат виконання програми.</p> <p>2. Вивести на семисегментний дисплей відповідно до Завдання 2 чотиризначне число в форматі «AB.CD», де A-старший розряд числа, D – молодший розряд: A=5, B=E, C=8, D=1.</p>
2	<p>1. Виконати програму зведення в квадрат (Завдання 1) для числа 04H. Записати результат виконання програми.</p> <p>2. Вивести на семисегментний дисплей відповідно до Завдання 2 чотиризначне число в форматі «A.BCD», де A-старший розряд числа, D – молодший розряд: A=0, B=8, C=4, D=C.</p>
3	<p>1. Виконати програму зведення в квадрат (Завдання 1) для числа 04H. Записати результат виконання програми.</p> <p>2. Вивести на семисегментний дисплей відповідно до Завдання 2 чотиризначне число в форматі «ABC.D», де A-старший розряд числа, D – молодший розряд: A=8, B=0, C=6, D=4.</p>
4	<p>1. Виконати програму зведення в квадрат (Завдання 1) для числа 09H. Записати результат виконання програми.</p>

	<p>2. Вивести на семисегментний дисплей відповідно до Завдання 2 чотиризначне число в форматі «AB.CD», де А-старший розряд числа, D – молодший розряд: A=0, B=8, C=8, D=4.</p>
5	<p>1. Виконати програму зведення в квадрат (Завдання 1) для числа 08H. Записати результат виконання програми.</p> <p>2. Вивести на семисегментний дисплей відповідно до Завдання 2 чотиризначне число в форматі «AB.CD», де А-старший розряд числа, D – молодший розряд: A=1, B=0, C=8, D=4.</p>
6	<p>1. Виконати програму зведення в квадрат (Завдання 1) для числа 10. Записати результат виконання програми.</p> <p>2. Вивести на семисегментний дисплей відповідно до Завдання 2 чотиризначне число в форматі «ABC.D», де А-старший розряд числа, D – молодший розряд: A=E, B=1, C=8, D=0.</p>
7	<p>1. Виконати програму зведення в квадрат (Завдання 1) для числа 12. Записати результат виконання програми.</p> <p>2. Вивести на семисегментний дисплей відповідно до Завдання 2 чотиризначне число в форматі «AB.CD», де А-старший розряд числа, D – молодший розряд: A=8, B=0, C=8, D=0.</p>
8	<p>1. Виконати програму зведення в квадрат (Завдання 1) для числа 11H. Записати результат виконання програми.</p> <p>2. Вивести на семисегментний дисплей відповідно до Завдання 2 чотиризначне число в форматі «A.BCD», де А-старший розряд числа, D – молодший розряд: A=4, B=4, C=8, D=0.</p>
9	<p>1. Виконати програму зведення в квадрат (Завдання 1) для числа 14. Записати результат виконання програми.</p> <p>2. Вивести на семисегментний дисплей відповідно до Завдання 2 чотиризначне число в форматі «AB.CD», де А-старший розряд числа, D – молодший розряд: A=C, B=0, C=8, D=A.</p>
10	<p>1. Виконати програму зведення в квадрат (Завдання 1) для числа 15H. Записати результат виконання програми.</p> <p>2. Вивести на семисегментний дисплей відповідно до Завдання 2 чотиризначне число в форматі «ABC.D», де А-старший розряд числа, D – молодший розряд: A=C, B=E, C=8, D=0.</p>
11	<p>1. Виконати програму зведення в квадрат (Завдання 1) для числа 20. Записати результат виконання програми.</p> <p>2. Вивести на семисегментний дисплей відповідно до Завдання 2 чотиризначне число в форматі «AB.CD», де А-старший розряд числа, D – молодший розряд: A=5, B=4, C=8, D=1.</p>
12	<p>1. Виконати програму зведення в квадрат (Завдання 1) для числа 20H. Записати результат виконання програми.</p> <p>2. Вивести на семисегментний дисплей відповідно до Завдання 2 чотиризначне число в форматі «A.BCD», де А-старший розряд числа, D – молодший розряд: A=8, B=A, C=8, D=0.</p>

3. Додати у звіт копію екрану з виконаною програмою до Завдання 1 та Завдання 2 на програмному симуляторі відповідно до вказаного варіанту.

4. Послідовність виконання роботи

4.1. Виконати завдання відповідно до вказаного варіанту на програмному симуляторі EdSim51 наступним чином:

4.1.1. Відкрити інтерфейс симулятора, двічі клацнувши клавішею миші на архівованому

файлі «EdSim51.jar». Відкриється інтерфейс програмного симулятора, зображений на рис. 3.

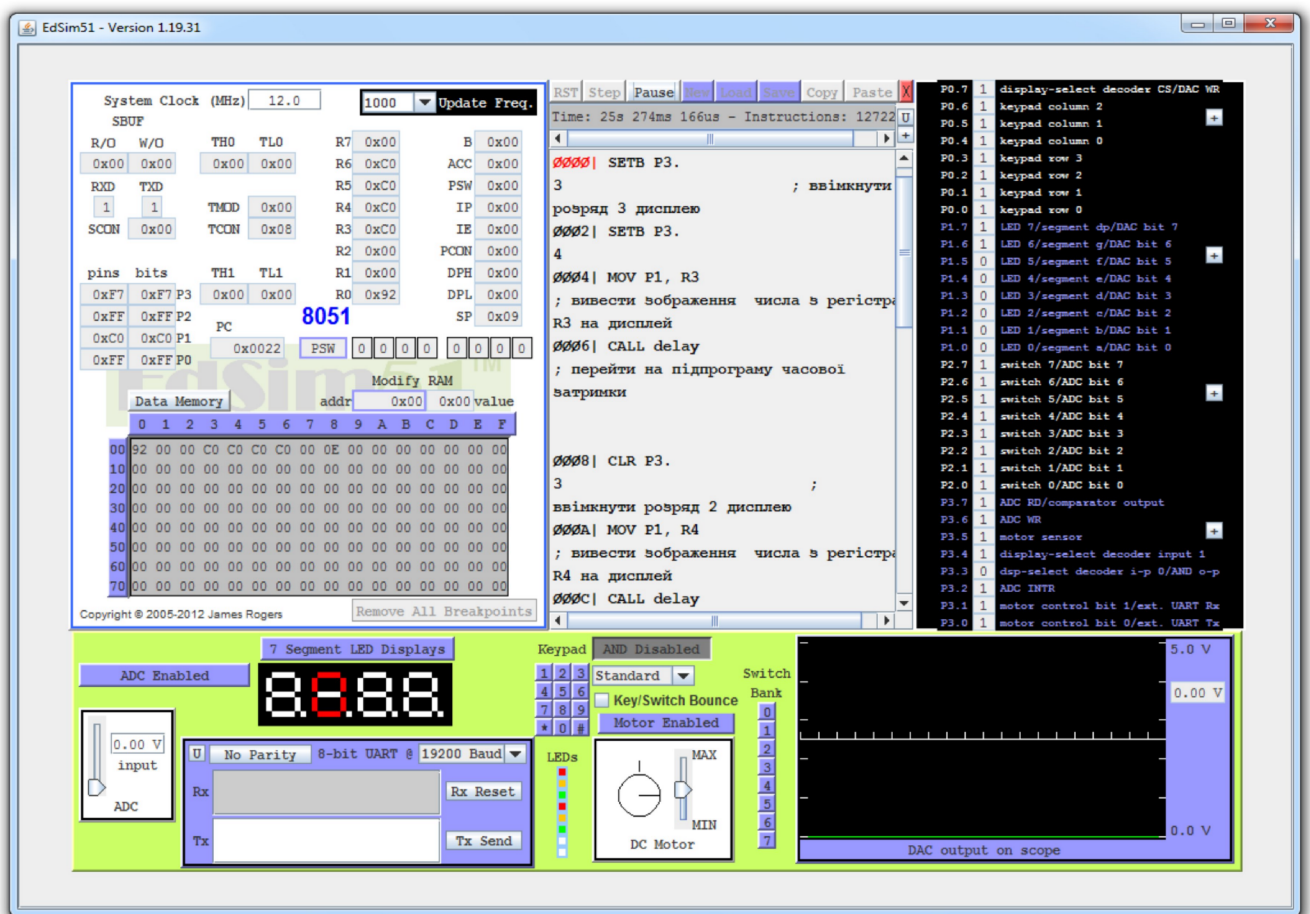


Рис. 3 Інтерфейс програмного симулятора

Середнє поле симулятора, що називається “Панель коду Асемблера”, в верхній частині містить кнопки “Reset”, “Assm”, “Run”, “Load”, “Save”, “Copy”, “Past”.

Панель коду використовується для:

- **набору команд** програми з клавіатури. Для цього курсор встановлюється в верхній частині панелі і вводиться програма по одній команді в рядку (при потребі, з міткою та коментарем) (див. рис. 3);
- **завантаження** вже існуючої програми. Для цього необхідно на панелі вгорі натиснути кнопку “Load” і вказати шлях до потрібного файлу;
- **запису** набраного файлу. Для цього потрібно натиснути кнопку “Save” і вказати шлях для збереження файлу.

4.1.2. Перед виконанням програми необхідно натиснути кнопку “Assm” панелі для асемблювання програми. Після цього, якщо команда записана невірно, в рядку під верхнім рядом кнопок панелі (на рис. 3 виділений сірим кольором) з’явиться повідомлення про помилку, а **колір рядка зміниться на червоний**. Червоним кольором буде виділена також невірно написана команда.

Якщо помилки відсутні, зліва від команд набраної програми з’являться адреси, і сама програма буде готова до виконання. Після асемблювання кнопка “Assm” зміниться на кнопку “Step”. Таким чином, є можливим виконувати програму покомандно в **кроковому режимі**, натискаючи кнопку “Step” після виконання кожної команди, або в **автоматичному режимі**, коли виконується вся програма, натиснувши один раз кнопку “Run”. В останньому випадку програму слід закінчувати директивою “End”.

При написанні програми можна користуватися для копіювання її фрагментів та вставки

в будь-якому місці “Панелі коду Асемблера” кнопками “Copy” та “Past”.

Щоб зупинити виконання програми і скинути в початковий стан регістри мікроконтролера симулятора необхідно натиснути кнопку “Reset”.

*Примітка

1. Якщо Ви хочете виконати якусь з команд над вмістом регістру чи комірки пам'яті, наприклад, команду пересилання з регістру в регістр, необхідно в регістр, з якого буде здійснене пересилання, командою MOV попередньо записати якесь значення операнду (адресу чи константу).

2. Програма, що виконується, буде записана в пам'ять програм, вміст якої можна побачити, натиснувши на кнопку “Data memory” в нижній частині “Панелі пам'яті даних та програмної пам'яті”, що знаходиться зліва від “Панелі коду Асемблера”. Після натискання кнопка “Data memory” зміниться на кнопку “Code memory”, тобто буде висвічуватися в полі пам'яті вміст пам'яті програм.

5. Контрольні запитання

1. Як формувати таблицю з даними в пам'яті програм?
2. Пояснити роботу програми зведення числа в квадрат.
3. Які значення в програмі зведення в квадрат може приймати вихідне число?
4. Пояснити роботу програми переведення двійкового числа в двійковій-десятькове число.
5. Які значення може приймати вихідне двійкове число?

Додаток 1

Перетворення цілих десяткових чисел на двійкові

Припустимо, потрібно перетворити число 21 на двійкове. Можна скористатися такою процедурою:

$$21 / 2 = 10 \text{ з залишком } 1$$

$$10 / 2 = 5 \text{ без залишку } 0$$

$$5 / 2 = 2 \text{ з залишком } 1$$

$$2 / 2 = 1 \text{ без залишку } 0$$

$$1 / 2 = 0 \text{ з залишком } 1$$

Отже, ділиться кожна частка від ділення на 2 і записується залишок на кінець двійкового запису. Продовжується поділ до тих пір, поки в частці не буде 0. Результат записується *справа наліво*. Тобто, нижня остання цифра (1) буде самою лівою і т. д.

В результаті отримуємо число 21 в двійковій формі: 10101.

Перетворення дробових десяткових чисел на двійкові

Якщо у вихідному числі є ціла частина, то вона перетвориться окремо від дробової. Переклад дробового числа з десяткової системи числення в двійкову здійснюється за таким алгоритмом:

- Дріб множиться на основу двійкової системи числення (2);
- В отриманому добутку виділяється ціла частина, яка приймається як старший розряд числа в двійковій системі числення;
- Алгоритм завершується, якщо дробова частина отриманого добутку дорівнює нулю або якщо досягнута необхідна точність обчислень. В іншому випадку обчислення тривають над дробовою частиною добутку.

Приклад: Потрібно перевести десяткове дробове число 206,116 в дробове двійкове число.

Переведення цілої частини дає $206_{10} = 11001110_2$ за раніше описаним алгоритмом. Дробову частину 0,116 множимо на основу 2, заносючи цілі частини добутку у розряди після коми шуканого дробового двійкового числа:

$$0,116 \cdot 2 = 0,232$$

0,232 • 2 = 0,464
0,464 • 2 = 0,928
0,928 • 2 = 1,856
0,856 • 2 = 1,712
0,712 • 2 = 1,424
0,424 • 2 = 0,848
0,848 • 2 = 1,696
0,696 • 2 = 1,392
0,392 • 2 = 0,784

і т. д.

Таким чином $0,116_{10} \approx 0,0001110110_2$

Разом з цілим числом отримуємо: $206,116_{10} \approx 11001110,0001110110_2$

Рекомендована література

1. Проектування мікропроцесорних систем керування : навчальний посібник, перевидання / Медвідь В.Р., Пісьціо В.П., Козбур І.Р. – Тернопіль : Вид-во ТНТУ імені Івана Пулюя, 2015. – 360 с.
2. Handbook of Microcontrollers/Predko Michael. NYс. McGraw-Hill. 1998. 861 p.
3. Бойко В. І., Гуржій А. М., Жуйков В. Я. та ін.Схемотехніка електронних схем: У 3 кн. Кн.3 Мікропроцесори та мікроконтролери: підручник. 2-ге вид., допов. і переробл. К.: Вища шк., 2004. 399 с.
- 4 Мілих В. І., Шавьолкін О. О. Електротехніка, електроніка та мікропроцесорна техніка: підручник; за ред. В. І. Мілих. 2-е вид. К.: Каравела, 2008. 688 с.