НАВЧАЛЬНО-МЕТОДИЧНА ЛІТЕРАТУРА

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Тернопільський національний технічний університет
імені Івана Пулюя

Кафедра автоматизації
технологічних
процесів та виробництв

Методичні вказівки для виконання лабораторної роботи №4 “Програмування мікроконтролера I8051 з використанням програмної моделі EdSim51. Реалізація спеціальних функцій МК51” з курсу “Розробка систем керування на основі ОМЕОМ”

Тернопіль 2016

Методичні вказівки для виконання лабораторної роботи №4 «Програмування мікроконтролера I8051 з використанням програмної моделі EdSim51. Реалізація спеціальних функцій МК51» з курсу «Розробка систем керування на основі ОМЕОМ».

Методичні вказівки розглянуті і схвалені кафедрою «Автоматизація технологічних процесів і виробництв», протокол № 4 від 21.11.2016 р.

Рецензент д.т.н. Карпінський М.П.

Відповідальні за випуск доцент, к.т.н. Медвідь В.Р.

асистент Пісьціо В.П.

Лабораторна робота №4

Програмування мікроконтролера I8051 з використанням програмної моделі EdSim51. Реалізація спеціальних функцій МК51

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.

Використання цих команд приведено в наступному прикладі.

Потрібно скласти підпрограму обчислення синуса кута X (X змінюється в межах від 0 до 89 градусів з дискретністю 1°).

Найбільш швидке обчислення функції можна отримати шляхом вибірки готового значення синуса з таблиці. Така таблиця для діапазону 0 - 89 ° займе 90 байтів при похибці 0,4%. Кожен байт таблиці буде містити дробову частину двійкового представлення синуса.

Вихідним параметром для підпрограми служить значення кута X, що знаходиться в акумуляторі:

```
                        ; обчислення SIN (x) по таблиці значень
                        ; вхід: X в A (акумуляторі) в межах від 0 до 89 градусів
                        ; вихід: дрібна частина значення синуса в акумуляторі
        ACALL SINX      ; виклик підпрограми обчислення синуса
        ...
SINX:   INC A           ; інкремент акумулятора
        MOVC A, @ A + PC ; завантаження значення синуса з таблиці в акумулятор
        RET             ; повернення з підпрограми
                        ; таблиця значень синуса
        DB 0            ; sin (0) = 0
DB 00000100B            ; sin (1) = 0.017
DB 00001001B            ; sin (2) = 0.035
        . . .
        . . .
DB 11111111B            ; sin (89) = 0.999.
```

Інкремент акумулятора перед зверненням до таблиці необхідний через наявність однобайтної команди повернення RET, розташованої між командою MOVC і початком таблиці значень синуса.

2. Завдання

1. Дослідити програму зведення в квадрат числа.
2. Дослідити програму перекладу двійкового числа в двійково-десяткове число.
3. Дослідити і виконати програму виводу на семисегментний дисплей інформації:

Програма виводу інфориації на 7-сегментний дисплей

```
; Ця програма мультиплексує номер 1234
; на чотирьох 7-сегментних індикаторах.
; Примітка: логічний 0 запалює сегмент дисплею.

start:
    SETB P3.3           ;
    SETB P3.4           ; ввімкнути дисплей 3
    MOV P1, #11111001B  ; вивести зображення “1” на дисплей
    CALL delay          ; перейти на підпрограму часової затримки
```

```
CLR P3.3                ; ввімкнути дисплей 2
MOV P1, #10100100B      ; вивести зображення “2” на дисплей
CALL delay              ; перейти на підпрограму часової затримки
CLR P3.4                ;
SETB P3.3               ; ввімкнути дисплей 1
MOV P1, #10110000B      ; вивести зображення “3“ на дисплей
CALL delay              ; перейти на підпрограму часової затримки
CLR P3.3                ; ввімкнути дисплей 0
MOV P1, #10011001B      ; вивести зображення “4“ на дисплей
CALL delay              ; перейти на підпрограму часової затримки
JMP start               ; перейти на початок програми

delay:
MOV R0, #200            ; підпрограма часової затримки
DJNZ R0, $
RET
```

Схема підключення семисегментного дисплею показана на рис. 1.

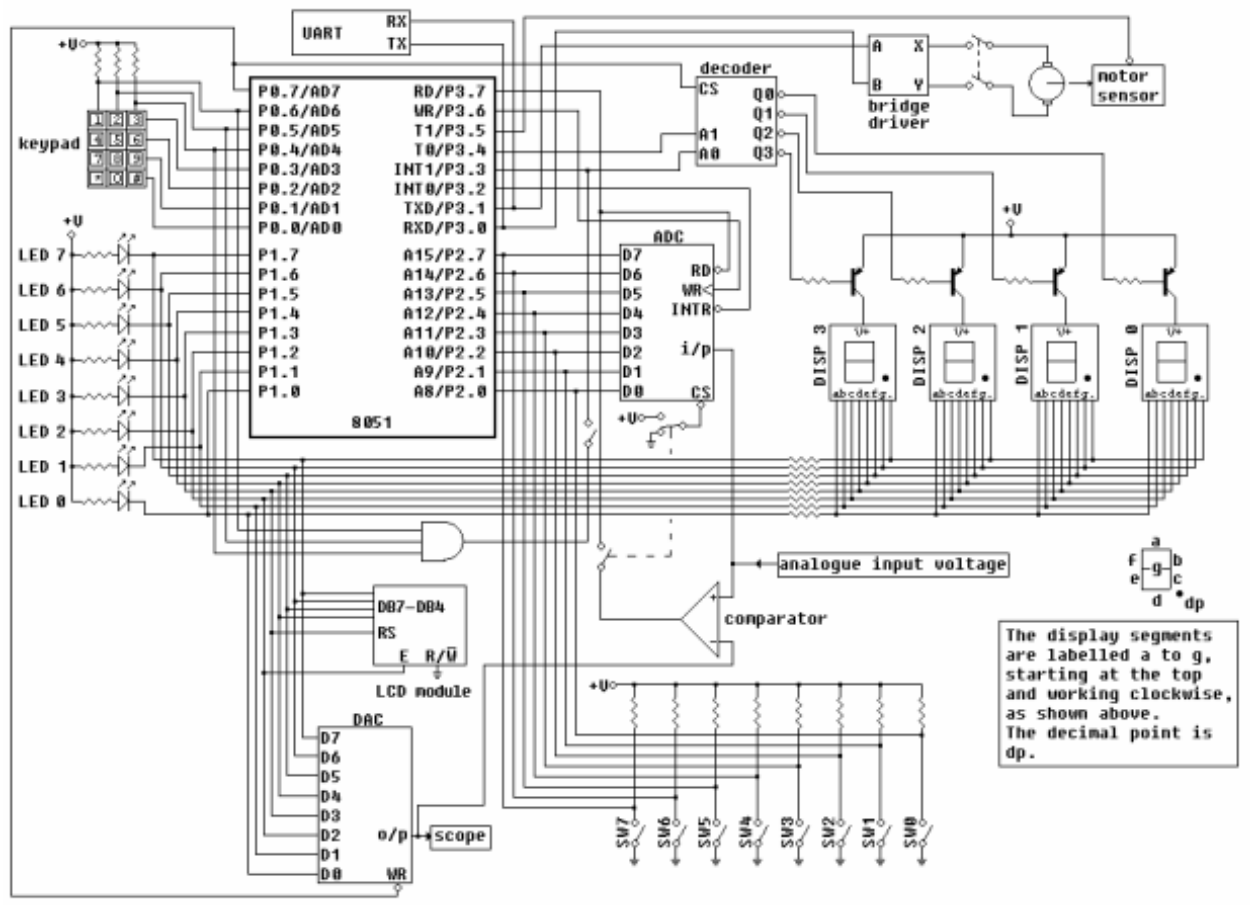

Рис. 1 Схема підключення семисегментного дисплею

4. Розробити і дослідити підпрограму обчислення синуса кута в діапазоні 0-80 ° з кроком 10 °.

5. Розробити програму перекладу двійкового числа в двійковій-десяткове, використовуючи рекурентне співвідношення:

$$d7\ d6\ d5\ d4\ d3\ d2\ d1\ d0 = d0 + 2\ (d1 + 2\ (d2 + 2\ (d3 + 2\ (d4 + 2\ (d5 + 2\ (d6 + 2\ d7))))))).$$

3. Послідовність виконання роботи

3.1. Вивчити команди відповідно до завдання. Вивчення кожної команди проводити наступним чином:

3.1.1. Відкрити інтерфейс емулятора, двічі клацнувши клавішею миші на архівованому файлі «EdSim51.jar”. Відкриється інтерфейс програмного емулятора, зображений на рис. 2.

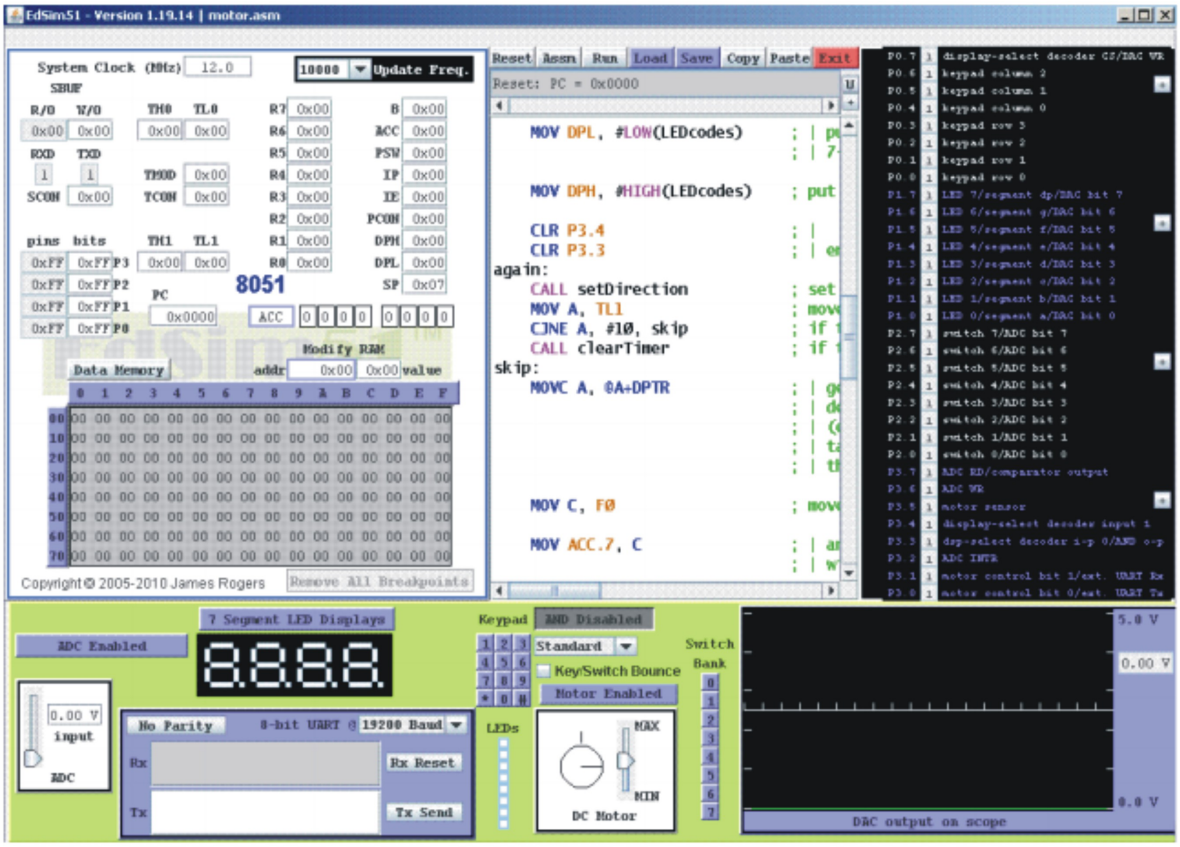

Рис. 2 Інтерфейс програмного емулятора

Середнє поле емулятора, 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. 2);
- **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.

3.1.2. 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.1 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 командою ***“Stop”.***

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 прорами і скинути в початковий стан регістри мікроконтролера емулятора необхідно натиснути кнопку **“*Reset*”.**

3.1.3. Записати в звіт зміни в вікнах регістрів мікроконтролера, які використовуються при виконанні програм, за прикладом, наведеним в табл. 1 (для 5..6 команд з програми).

Таблиця 1. Результати виконання команд

№	Команда	Код	Виконувана операція	Вміст використовуваних регістрів і комірок пам'яті до і після виконання		Пояснення
				До	**Після**	
1	MOV A,R0	E8	Пересилання байту даних з регістру R0 в акумулятор A	A/00 PC/00 PSW/00	A/F2 PC/01 PSW/01	
2	…	…	…	…	…	…
…	…	…	…	…	…	…
…	…	…	…	…	…	…

***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.

4. Контрольні запитання

1. Як формувати таблицю 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-десяткове число.
5. Які значення може приймати вихідне двійкове число?