

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

Кафедра автоматизації
технологічних
процесів та виробництв

Методичні вказівки
для виконання лабораторної роботи №1
“Програмування мікроконтролера I8051 з
використанням програмної моделі EdSim51.
Команди персилання даних”
з курсу “Розробка систем керування на основі
OMEOM”

Методичні вказівки для виконання лабораторної роботи №1 «Програмування мікроконтролера I8051 з використанням програмної моделі EdSim51. Команди персилання даних» 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.

Відповідальні за випуск

доцент, к.т.н. Медвідь В.Р.
асистент Пісьціо В.П.

Лабораторна робота №1

Програмування мікроконтролера I8051 з використанням програмної моделі EdSim51. Команди пересилання даних

1. Команди МК51

Система команд мікроконтролера МК51 містить 111 базових команд, які зручно розділити за функціональною ознакою на п'ять груп: команди пересилання даних, арифметичних операцій, логічних операцій, передачі управління і операцій з бітами.

Більшість команд мають формат один або два байти і виконуються за один або два машинних циклу. При тактовій частоті 12 МГц тривалість машинного циклу складає 1 мкс.

Склад операндів МК51 включає в себе операнди чотирьох типів: біти, 4-бітові цифри, байти і 16-бітові слова. Є також можливість адресації окремих бітів блоку регістрів спеціальних функцій (РСФ) і портів. Для адресації бітів використовується пряма 8-бітова адреса (bit).

Чотирибітні операнди використовуються тільки під час операції обміну (команди SWAP і XCHD).

Восьмибітним операндом може бути комірка пам'яті програм або даних (резидентної або зовнішньої), константа (безпосередній операнд), регістри спеціальних функцій (РСФ), а також порти вводу/виводу.

Порти і РСФ адресуються тільки прямим способом. Байти пам'яті можуть адресуватися також і непрямим чином через адресні регістри (RO, RI, DPTR і PC).

Двобайтні операнди - це константи і прямі адреси, для подання яких використовуються другий і третій байти команди.

2. Група команд пересилання даних

Більшу частину команд даної групи (див. табл. 1 і меню «Довідка») складають команди пересилання і обміну байтів. Команди пересилання бітів представлені в групі команд бітових операцій. Всі команди даної групи не модифікують флаги результату, за винятком команд завантаження PSW і акумулятора (флаг паритету).

Звернення до зовнішньої пам'яті даних (ЗПД). У МК51 реалізований режим непрямої адресації ЗПД. При використанні команд MOVX @Ri забезпечується доступ до 256 байтів зовнішньої пам'яті даних.

Існує також режим звернення до розширеної ВПД, коли для доступу використовується 16-бітова адреса, що зберігається в регістрі-покажчику даних (DPTR). Команди MOVX @DPTR забезпечують доступ до 65 536 байтів ЗПД.

Таблиця 1. Команди пересилання даних

Назва команди	Мнемокод	КОП	Т	Б	Ц	Операція
Пересилання в акумулятор з регістру ($n=0\div7$)	MOV A, Rn	11101rrr	1	1	1	(A) ← (Rn)
Пересилання в акумулятор прямоадресованого байту	MOV A, ad	11100101	3	2	1	(A) ← (ad)
Пересилання в акумулятор байту з РПД ($i=0,1$)	MOV A, @Ri	1110011i	1	1	1	(A) ← ((Ri))
Завантаження в акумулятор константи	MOV A, #d	01110100	2	2	1	(A) ← #d
Пересилання в регістр з акумулятора	MOV Rn, A	11111rrr	1	1	1	(Rn) ← (A)
Пересилання в регістр прямоадресованого байту	MOV Rn, ad	10101rrr	3	2	2	(Rn) ← (ad)
Завантаження в регістр константи	MOV Rn, #d	01111rrr	2	2	1	(Rn) ← #d
Пересилання за прямою адресою акумулятора	MOV ad, A	11110101	3	2	1	(ad) ← (A)
Пересилання за прямою адресою регістру	MOV ad, Rn	10001rrr	3	2	2	(ad) ← (Rn)

Назва команди	Мнемокод	КОП	Т	Б	Ц	Операція
Пересилання прямоадресованого байту за прямою адресою	MOV ad, ads	10000101	9	3	2	(add) ← (ads)
Пересилання байту з РПД за прямою адресою	MOV ad, @Ri	1000011i	3	2	2	(ad) ← ((Ri))
Пересилання за прямою адресою константи	MOV ad, #d	01110101	7	3	2	(ad) ← #d
Пересилання в РПД з акумулятора	MOV @Ri, A	1111011i	1	1	1	((Ri)) ← (A)
Пересилання в РПД прямоадресованого байту	MOV @Ri, ad	0110011i	3	2	2	((Ri)) ← (ad)
Пересилання в РПД константи	MOV @Ri, #d	0111011i	2	2	1	((Ri)) ← #d
Завантаження покажчика даних	MOV DPTR, #d16	10010000	13	3	2	(DPTR) ← #d16
Пересилання в акумулятор байту з ПП	MOVC A, @A+DPTR	10010011	1	1	2	(A) ← ((A) + (DPTR))
Пересилання в акумулятор байту з ПП	MOVC A, @A+PC	10000011	1	1	2	(PC) ← (PC)+1, (A) ← ((A)+(PC))
Пересилання в акумулятор байту з ЗПД	MOVX A, @Ri	1110001i	1	1	2	(A) ← ((Ri))
Пересилання в акумулятор байту з розширеної ЗПД	MOVX A, @DPTR	11100000	1	1	2	(A) ← ((DPTR))
Пересилання в ЗПД з акумулятора	MOVX @Ri, A	1111001i	1	1	2	((Ri)) ← (A)
Пересилання в розширену ЗПД з акумулятора	MOVX @DPTR, A	11110000	1	1	2	((DPTR)) ← (A)
Завантаження в стек	PUSH ad	11000000	3	2	2	(SP) ← (SP) + 1, ((SP)) ← (ad)
Витяг із стеку	POP ad	11010000	3	2	2	(ad) ← (SP), (SP) ← (SP) - 1
Обмін акумулятора з регістром	XCH A, Rn	11001rrr	1	1	1	(A) ↔ (Rn)
Обмін акумулятора з прямоадресованим байтом	XCH A, ad	11000101	3	2	1	(A) ↔ (ad)
Обмін акумулятора з байтом з РПД	XCH A, @Ri	1100011i	1	1	1	(A) ↔ ((Ri))
Обмін молодших тетрад акумулятора і байту РПД	XCHD A, @Ri	1101011i	1	1	1	(A _{0...3}) ↔ ((Ri) _{0...3})

3. Завдання

1. Вивчити програмну модель мікроконтролера.
2. Вивчити команди пересилання: MOV A, #10H; MOV 01H, #64H; MOV P1, R1.
3. Вивчити команди пересилання: MOV A, @R1; XCHD A, @R1; MOV DPTR, #0121H.
4. Написати і дослідити роботу програми пересилання байту з комірки пам'яті програм з адресою 71H в регістр R1 банку 1.

4. Послідовність виконання роботи

4.1. Вивчити команди пересилання. Вивчення кожної команди проводити наступним чином:

4.1.1. Відкрити інтерфейс емулятора, двічі клацнувши клав'яшею миші на архівованому файлі «EdSim51.jar». Відкриється інтерфейс програмного емулятора, зображений на рис.1.

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.1);
- **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.1 виділений сірим кольором) з'явиться повідомлення про помилку, а колір рядка зміниться на червоний. Червоним кольором буде виділена також

невірно написана команда.

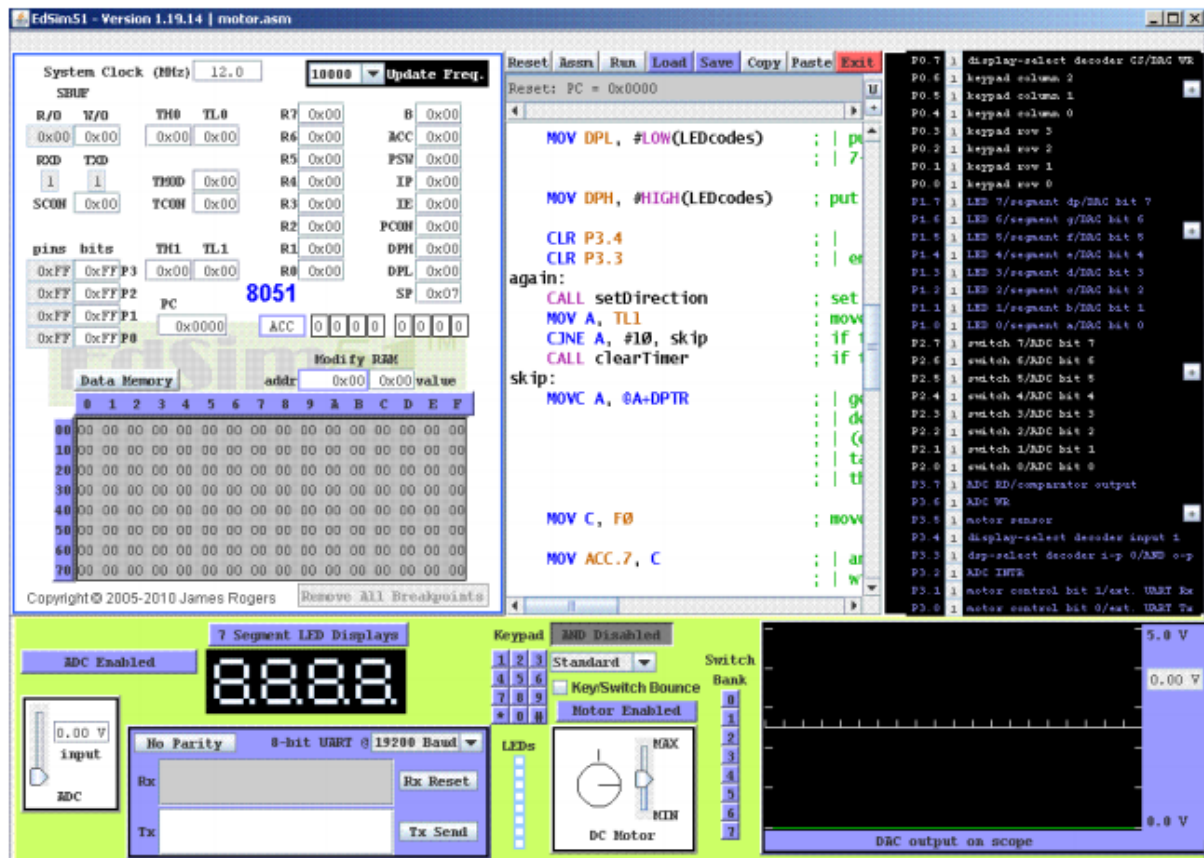


Рис.1 Інтерфейс програмного емулятора EdSim51

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

4.1.3. Записати в звіт зміни в вікнах регістрів мікроконтролера відповідно до табл. 2.

4.2. Написати і дослідити роботу програми пересилання байту з комірки пам'яті програм за адресою 71H в регістр R1 банку 1 з використанням регістру DPTR (для варіанту 1) і в регістр R2 банку 3 з використанням регістру PC (для варіанту 2).

Таблиця 2 - 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 пересилання, наприклад, з регістра 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 проограм.

5. Контрольні запитання

- Структура резидентної пам'яті програм.
- Структура резидентної пам'яті даних.
- Структура блоку регістрів спеціальних функцій.
- Формат слова стану програми.
- Призначення та позначення основних елементів програмної моделі мікроконтролера.
- Пояснити вміст вікон **"Reset"**, **"Assm"**.
- Пояснити вміст вікон **"Run"**, **"Load"**.
- Пояснити вміст вікон **"Copy"**, **"Past"**.
- Пояснити вміст вікна **"Save"**. **"Step"**.
- Як здійснюється введення команди і виконання програми?
- Пояснити виконання команд пересилання.