

Тернопільський національний технічний
університет імені Івана Пулюя

Кафедра автоматизації
технологічних процесів
і виробництв

Лабораторна робота № 1
з курсу
Проектування мікропроцесорних
систем керування технологічними
процесами

Програмування мікроконтролера
MCS51 з використанням програмної
моделі EdSim51.

Команди передачі даних

Тернопіль 2023

Методичні вказівки для виконання лабораторної роботи № 1 «Програмування мікроконтролера MCS51 з використанням програмної моделі EdSim51. Команди передачі даних» з курсу «Проектування мікропроцесорних систем керування технологічними процесами»/ Укл.: Медвідь В.Р., Пісьціо В.П. - Тернопіль ТНТУ, 2023 - 8 с.

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

Лабораторна робота № 1
Програмування мікроконтролера I8051 з використанням програмної моделі
(програмного симулятора) EdSim51. Команди передачі даних

1. Команди MCS51

Система команд мікроконтролера MCS51 містить 111 базових команд, які зручно розділити за функціональною ознакою на п'ять груп: команди передачі 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 MCS51 включає в себе операнди чотирьох типів: біти, 4-бітові цифри, байти і 16-бітні слова. Є також можливість адресації окремих бітів блоку реєстрів спеціальних функцій (РСФ) і портів. Для адресації бітів використовується пряма 8-бітна 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), реєстри 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 реєстри (R0, R1, 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 (табл. 1) складають команди пересилання і обміну байтів. Команди пересилання бітів представлені в групі команд бітових операцій. Всі команди даної групи не модифікують флажки результату, за винятком команд завантаження PSW і акумулятора (флажок паритету).

Звернення до зовнішньої пам'яті даних (ЗПД). У МК51 реалізований режим непрямой 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 реєстрі-покажчику даних (DPTR). Команди MOVX@DPTR забезпечують доступ до 65 536 байтів ЗПД.

Таблиця 1. Команди пересилання даних

Назва команди	Мнемокод	КОП	Операція
Пересилання в акумулятор з реєстру (n=0÷7)	MOV A, Rn	11101rrr	(A) ← (Rn)
Пересилання в акумулятор прямоадресованого байту	MOV A, ad	11100101	(A) ← (ad)
Пересилання в акумулятор байту з РПД (i=0,1)	MOV A, @Ri	1110011i	(A) ← ((Ri))
Завантаження в акумулятор константи	MOV A, #d	01110100	(A) ← #d
Пересилання в реєстр з акумулятора	MOV Rn, A	11111rrr	(Rn) ← (A)
Пересилання в реєстр прямоадресованого байту	MOV Rn, ad	10101rrr	(Rn) ← (ad)
Завантаження в реєстр константи	MOV Rn, #d	01111rrr	(Rn) ← #d
Пересилання за прямою адресою акумулятора	MOV ad, A	11110101	(ad) ← (A)
Пересилання за прямою адресою реєстру	MOV ad, Rn	10001rrr	(ad) ← (Rn)
Пересилання прямоадресованого байту за прямою адресою	MOV add, ads	10000101	(add) ← (ads)
Пересилання байту з РПД за прямою адресою	MOV ad, @Ri	1000011i	(ad) ← ((Ri))
Пересилання за прямою адресою константи	MOV ad, #d	01110101	(ad) ← #d

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

3. Завдання на самостійну підготовку

1. Вивчити роботу програмної моделі мікроконтролера MCS51.
2. Пояснити використання команд пересилання: MOV A, #10H; MOV 01H, #64H; MOV P1, R1.
3. Пояснити використання команд: MOV A, @R1; XCHD A, @R1; MOV DPTR, #0121H.

ЗАВДАННЯ

1. Завантажити в EdSim51 та виконати команди з Прикладу 1 та Прикладу 2 в покроковому режимі роботи симулятора. Звернути увагу на зміну вмісту регістрів після виконання команд програми.

Приклад 1: програма, що містить команди пересилання:

LOAD:

MOV R0,#41H	;завантаження в R0 покажчика даних
MOV @R0,#1CH	;завантаження в пам'ять числа 1CH
INC R0	;інкремент покажчика
MOV @R0,#3FH	;записати в пам'ять число 3FH
MOV A,#0F1H	;завантаження в акумулятор числа 41H
MOV A,0F1H	;завантаження в акумулятор байту з пам'яті з адресою 41H
MOV A,#01H	;завантаження в акумулятор числа 01H
MOV DPTR,#0001H	;завантаження в DPTR числа 0001H
MOVC A,@A+DPTR	;завантаження в акумулятор числа з пам'яті програм
MOV 45H,A	;завантаження в РПД числа з акумулятора
PUSH ACC	;завантаження в стек операнд з пам'яті з адресою 10H
MOV A,#05H	;завантаження в акумулятор числа 05H
POP ACC	;повернення із стеку операнда в акумулятор
POP 46H	;повернення із стеку операнда в пам'ять за адресою 46H
MOV A,20H	;завантаження в акумулятор числа 20H
XCH A, R0	;обмін вмісту акумулятора з вмістом регістра R0
MOV A,#0FFH	;завантаження в акумулятор числа FFH
MOV R0,#18H	;завантаження в R0 банку 0 покажчика даних
MOV PSW,#00001000b	;вибір 1-го банку РЗП

```

MOV R0,#22H      ;завантаження в R0 банку 1 покажчика даних
MOV PSW,#00010000b ;вибір 2-го банку РЗП
MOV R1,#40H      ;завантаження в R0 банку 1 покажчика даних
MOV PSW,#00001100b ;вибір 3-го банку РЗП
MOV R0,#10H      ;завантаження в R0 банку 1 покажчика даних

END              ;директива закінчення трансляції

```

Приклад 2: операції із стеком

START:

```

MOV R1,#02H      ;завантаження регістрів
MOV A,#30H
MOV R2,#00H
LCALL SUB        ;перехід на підпрограму
SJMP START

```

SUB:

```

PUSH PSW         ;зберігання в стеку вмісту PSW
PUSH ACC         ;зберігання вмісту акумулятора
PUSH B          ;зберігання вмісту B в стеку
ADD A,R1        ;обробка даних
MOV R2,A
POP B           ;відновлення B
POP ACC        ;відновлення акумулятора
POP PSW        ;відновлення PSW
RET            ;повернення з підпрограми
END

```

2. Написати і дослідити роботу команд з використанням програмної моделі відповідно до заданого варіанту:

Варіанти індивідуальних завдань (виконати відповідно до Прикладу 1)

№	Зміст індивідуального завдання
1	Записати і виконати з використанням програмної моделі в покроковому режимі команди: MOV R0,#45H; MOV @R0,#1EH; MOV R1,#46H; MOV @R1,#1FH; MOV A, #10H; MOV 01H, #64H; MOV P1, A; XCHD A, @R1; MOV PSW, 00010000b; MOV R3,#15H.
2	Записати і виконати з використанням програмної моделі в покроковому режимі команди: MOV 35H,#22H; MOV R1,#35H; MOV A, @R1; XCHD A, @R1; MOV DPTR, #0121H; MOV A, #20H; MOV 10H, #40H; XCHD A,@R0; MOV PSW, 00001000b; MOV R1,#25H.
3	Записати і виконати з використанням програмної моделі в покроковому режимі команди: MOV R1,#02H; MOV A,#30H; MOV R2,#10H; MOV A,R2; MOV B,R1; MOV DPL,#21H; MOV DPH,#14H; MOV A,10H; XCHD A, @R0; MOV P2, A.
4	Записати і виконати з використанням програмної моделі в покроковому режимі команди: MOV R0,#15H; MOV @R0,#10H; MOV R1,#12H; MOV @R1,#12H; MOV A,R0; MOV 13H,R0; MOV 15H, #15H; MOV A,15H; MOV PSW, 00011000b; MOV R7,#30H.
5	Записати і виконати з використанням програмної моделі в покроковому режимі команди: MOV A,#15H; MOV P1,A; MOV P2,A; MOV R0,#30H; MOV @R0,#20H; MOV A,@R0; MOV R2,R0; MOV 10H, #10H; MOV B,30H; MOV B,#30H .
6	Записати і виконати з використанням програмної моделі в покроковому режимі команди: MOV R0,#11H; MOV A,#10H; MOV R2,#3FH; MOV A,R0; MOV B,R0; MOV DPL,#10H; MOV DPH,#10H; MOV A,#1FH; MOV PSW, 00001000b; MOV R0,#45H.
7	Записати і виконати з використанням програмної моделі в покроковому режимі команди: MOV R1,#15H; MOV A,#20H; MOV R0,#1FH; MOV A,R0; MOV B,R1; MOV DPTR,#1020H; MOV

№	Зміст індивідуального завдання
	R4,#10H; MOV A,#29H; XCHD A, @R0; MOV P1, A.
8	Записати і виконати з використанням програмної моделі в покроковому режимі команди: MOV R1,#15H; MOV @R0,#10H; MOV R6,#46H; MOV @R1,#15H; MOV A, #10H; MOV 21H, #46H; MOV P2, A; XCHD A, @R1; MOV B,#11H, XCH A,R1.
9	Записати і виконати з використанням програмної моделі в покроковому режимі команди: MOV R1,#20H; MOV @R0,#33H; MOV R7,#12H; MOV @R1,#12H; MOV A,R0; MOV A,R1; MOV R5,A; MOV R6,#46H; MOV @R0,#35H; XCHD A, @R1; XCH A,R0; XCH A,@R0.
10	Записати і виконати з використанням програмної моделі в покроковому режимі команди: MOV 15H,#12H; MOV R1,#11H; MOV A, @R1; XCHD A, 15H; MOV DPTR, #2121H; MOV B, #20H; MOV R5, #04H; MOV P3, A; XCHD A, @R1; MOV PSW, 00011000b; MOV R5,#10H.
11	Записати і виконати з використанням програмної моделі в покроковому режимі команди: MOV R1,#0FH; MOV A,#0FH; MOV R4,#3FH; MOV B,R4; MOV 15H,R0; MOV DPL,#F0H; MOV DPH,#F0H; MOV R7,#1FH, MOV DPTR,#2325H; MOV A,#0A7.
12	Записати і виконати з використанням програмної моделі в покроковому режимі команди: MOV R1,#0FH; MOV A,#0FH; MOV R4,#3FH; MOV B,R4; MOV 15H,R0; MOV DPL,#F0H; MOV DPH,#F0H; MOV R6,#1FH; MOV A,#0F9; XCH A,R1.

3. Записати в звіт зміни в вікнах реєстрів мікроконтролера при виконанні програми завдання вибраного варіанту для перших пяти команд відповідно до табл. 2.

Таблиця 2 - Результати виконання команд

№	Команда	Виконувана операція	Вміст використовуваних реєстрів та комірок пам'яті до і після виконання		Пояснення
			До	Після	
1	MOV A,R0	Пересилання байту даних з реєстру R0 в акумулятор A	A/00 PC/00 PC/01	A/F2 PSW/00 PSW/01	
2
...
5

4. Додати у звіт копію екрану з виконаною програмою відповідно до обраного варіанту на програмному симуляторі.

Послідовність виконання роботи

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

4.1.1. Відкрити інтерфейс симулятора, двічі клацнувши клавішею миші на архівованому файлі «EdSim51.jar». Відкриється інтерфейс програмного симулятора, 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.

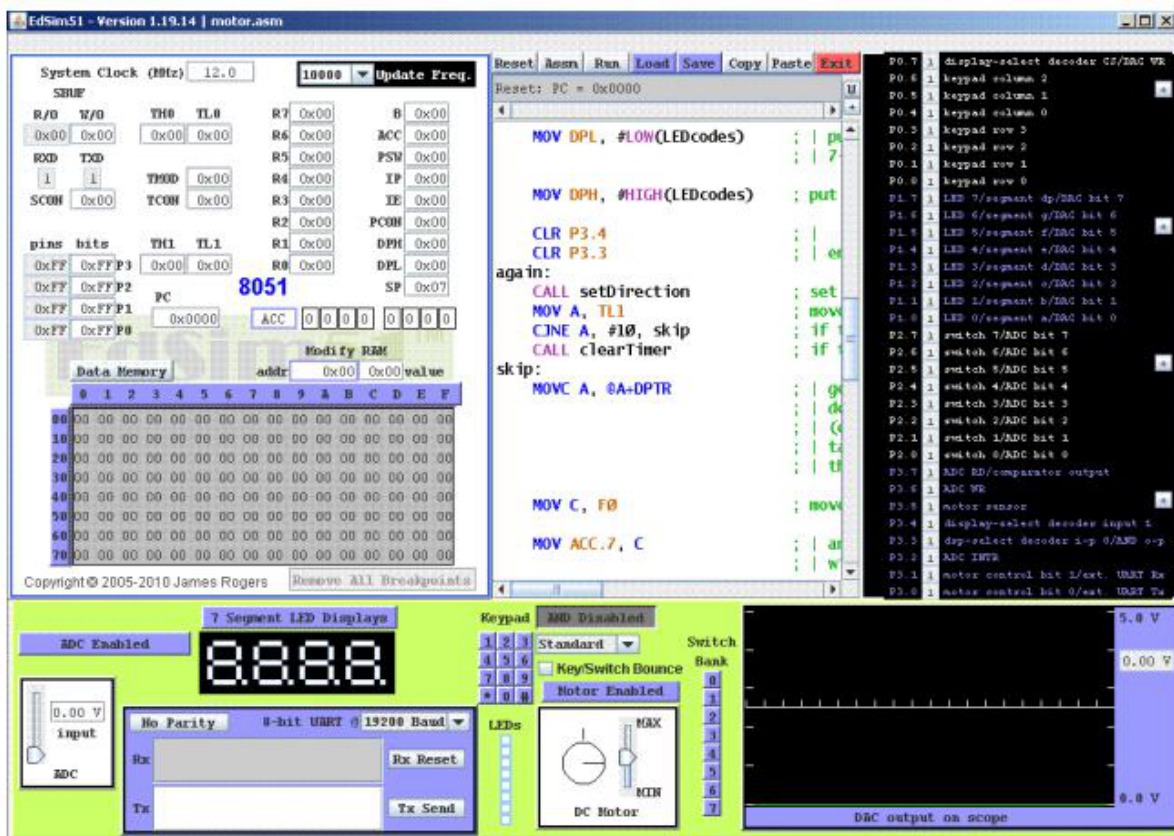


Рис. 1. Інтерфейс програмного симулятора EdSim51

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 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 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 програм.

5. Контрольні запитання

1. Структура резидентної пам'яті програм.
1. Структура резидентної пам'яті даних.
2. Структура блоку реєстрів спеціальних функцій.
3. Формат слова стану програми.
4. Призначення та позначення основних елементів програмної моделі мікроконтролера.
5. Пояснити вміст вікон **“Reset”**, **“Assm”**.
6. Пояснити вміст вікон **“Run”**, **“Load”**.
7. Пояснити вміст вікон **“Copy”**, **“Past”**.
8. Пояснити вміст вікна **“Save”**, **“Step”**.
9. Як здійснюється введення команди і виконання програми?

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.