

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І  
ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

Кафедра комп'ютерних систем і мереж

**МЕТОДИЧНІ ВКАЗІВКИ**

**до виконання лабораторних робіт  
з дисципліни**

**«Мікропроцесорні системи керування»**

для студентів спеціальності 123 «Комп'ютерна інженерія»

всіх форм навчання

Методичні вказівки до лабораторних робіт з дисципліни "Мікропроцесорні системи керування" для студентів спеціальності 123 "Комп'ютерна інженерія" всіх форм навчання / Укл.: В.В. Шкарупило. – Київ: НУБіП, 2019. – \_\_ с.

Укладачі: В.В. Шкарупило, к.т.н., доцент,  
доцент каф. комп'ютерних  
систем і мереж НУБіП України

Рецензенти: В.А. Лахно, д.т.н., професор  
каф. комп'ютерних систем і  
мереж НУБіП України,  
В.Г. Тракай, к.т.н., доцент  
каф. електротехніки,  
електромеханіки та  
електротехнологій  
НУБіП України

Відповідальний  
за випуск: В.В. Шкарупило, к.т.н., доцент,  
доцент каф. комп'ютерних  
систем і мереж НУБіП України

Затверджено:  
на засіданні кафедри

"Комп'ютерні системи і мережі"  
Протокол № \_ від \_\_.\_\_.2019 р.

Рекомендовано до видання:

Вченою радою факультету ІТ  
Протокол № \_ від \_\_.\_\_.2019 р.

**ЗМІСТ**

Вступ.....	4
Лабораторна робота №1.....	5
1 Програмування мікроконтролера i8051.....	5
Лабораторна робота №2.....	14
2 Арифметичні і логічні операції.....	14
Лабораторна робота №3.....	18
3 Програмування таймерів-лічильників.....	18
Лабораторна робота №4.....	
4 .....	
Література.....	20

## ВСТУП

Сучасний стан розвитку суспільства характеризується інтенсивним залученням різноманітних засобів автоматизації процесів управління, що мають місце, зокрема, у складі виробничих процесів. Поширеним шляхом реалізації названих засобів є використання спеціалізованих пристроїв на основі мікропроцесорних систем. Поширеними представниками таких систем є мікроконтролери сімейства MCS51: i8031, i8048, i8051. Найбільшого поширення здобув мікроконтролер i8051. Він та похідні від нього пристрої знайшли застосування у якості контролерів клавіатур, мікрохвильових пристроїв тощо. На пострадянському просторі поширення здобув аналог KP1816BE51.

Ознайомленню з аспектами функціонування і програмування мікроконтролерів сімейства MCS51 і присвячено даний комплекс лабораторних робіт: лабораторна робота №1 – «Програмування мікроконтролера i8051» – вивчення складу мікроконтролера, набуття базових навичок програмування мікроконтролера; лабораторна робота №2 – «Виконання арифметичних операцій» – набуття навичок виконання арифметичних операцій; лабораторна робота №3 – «Програмування таймерів-лічильників» – навчитися маніпулювати часовими затримками.

Виконання кожної із лабораторних робіт регламентується відповідним порядком виконання, а саме наступними позиціями: проводиться ознайомлення із необхідними для виконання роботи теоретичними відомостями; здійснюється послідовне виконання вказаних завдань роботи; за результатами виконання завдань роботи готується звіт, який має оформлюватися згідно наведених у кожній із робіт вимог, включаючи, зокрема, відповіді на контрольні питання.

## **ЛАБОРАТОРНА РОБОТА №1**

### **1 ПРОГРАМУВАННЯ МІКРОКОНТРОЛЕРА I8051**

**Мета роботи** – набути навичок програмування мікроконтролера i8051.

#### **1.1 Теоретичні відомості**

Мікроконтролер i8051 був створений компанією Intel у 1980 р. і набув широкого практичного використання. Він являє собою велику інтегральну схему (ВІС), що включає в собі мікропроцесор, а також периферійні пристрої: таймери, паралельний та послідовний інтерфейси, пам'ять програм, даних, контролер переривань. Такий підхід дозволив здешевити виробництво подібних систем [9].

Відповідне сімейство мікроконтролерів має назву MCS-51, а одним з найпоширеніших його представників є мікроконтролер i8051 – представник гарвардської архітектури. Вітчизняним аналогом названої системи є мікроконтролер КР1816ВЕ51.

Представники сімейства MCS-51 будуються з наступних складових [9]: 8-розрядний мікропроцесор, генератор тактових імпульсів, схеми керування й синхронізації, внутрішня пам'ять програм і внутрішня пам'ять даних, два таймери, паралельні порти введення/виведення, контролер переривань.

Мікросхема живиться від одного джерела напругою +5В та потужністю до 1,5 Вт, допускає експлуатацію в діапазоні температур від -40 до +100 °С.

Загальні характеристики усіх представників сімейства MCS-51:

- 32 двонаправлені лінії введення/виведення, представлені у якості чотирьох 8-розрядних портів;
- два 16-розрядних таймери-лічильники;
- контролер переривань;
- 16-розрядний лічильник команд;
- 16-розрядний показчик даних;
- синхронно-асинхронний приймач-передавач послідовного порту зі змінюваною швидкістю передачі;
- генератор тактових імпульсів (ГТІ).

Виводи мікроконтролера i8051 подано на рис. 1.1.

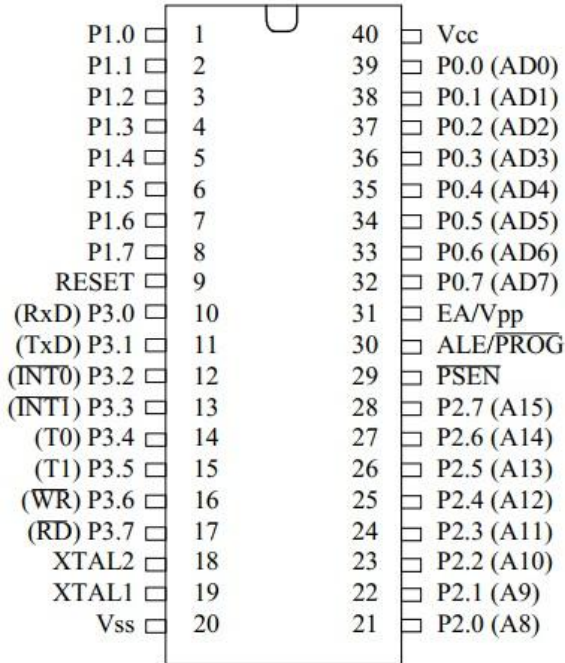


Рисунок 1.1 – Схема портів мікроконтролера i8051

Призначення портів мікроконтролера i8051 подано у табл. 1.1 [10].

Таблиця 1.1 – Призначення портів мікроконтролера i8051

Умовне позначення	Призначення
V <sub>ss</sub>	Потенціал «землі»;
V <sub>cc</sub>	Основна напруга живлення: +5 В;
XTAL1, XTAL2	Виводи для підключення кварцового резонатора;
RESET (RST)	Вхід загального скидання мікроконтролера;
PSEN	Дозвіл зовнішньої пам'яті програм; видається лише при зверненні до зовнішнього постійного запам'ятовуючого пристрою (ПЗП);

ALE	Строб адреси зовнішньої пам'яті;
EA	Відключення внутрішньої програмної пам'яті; рівень 0 на вході змушує мікроконтролер виконувати програму тільки із зовнішнього ПЗП; внутрішній ПЗП ігнорується;
P0	8-бітний двонапрявлений порт введення/виведення даних: при роботі із зовнішніми ОЗП і ПЗП лініями порту (у режимі часового мультиплексування) видається адреса зовнішньої пам'яті, після чого здійснюється передавання чи приймання даних;
P1	8-бітний квазі-двонапрявлений порт введення/виведення: кожен розряд порту може бути запрограмований як на введення, так і на виведення даних, не залежачи від станів інших розрядів;
P2	8-бітний квазі-двонапрявлений порт, аналогічний P1: окрім того лінії порту використовуються для видачі адреси при зверненні до зовнішньої пам'яті програм чи даних (якщо використовується 16-бітна адресація);
P3	8-бітний квазі-двонапрявлений порт, аналогічний P1, але виводи порту можуть виконувати ряд альтернативних функцій, що використовуються при роботі таймерів, порту послідовного введення/виведення, контролера переривань, а також зовнішньої пам'яті програм і даних.

Схему мікроконтролера на базі мікропроцесора i8051 подано на рис. 1.2.

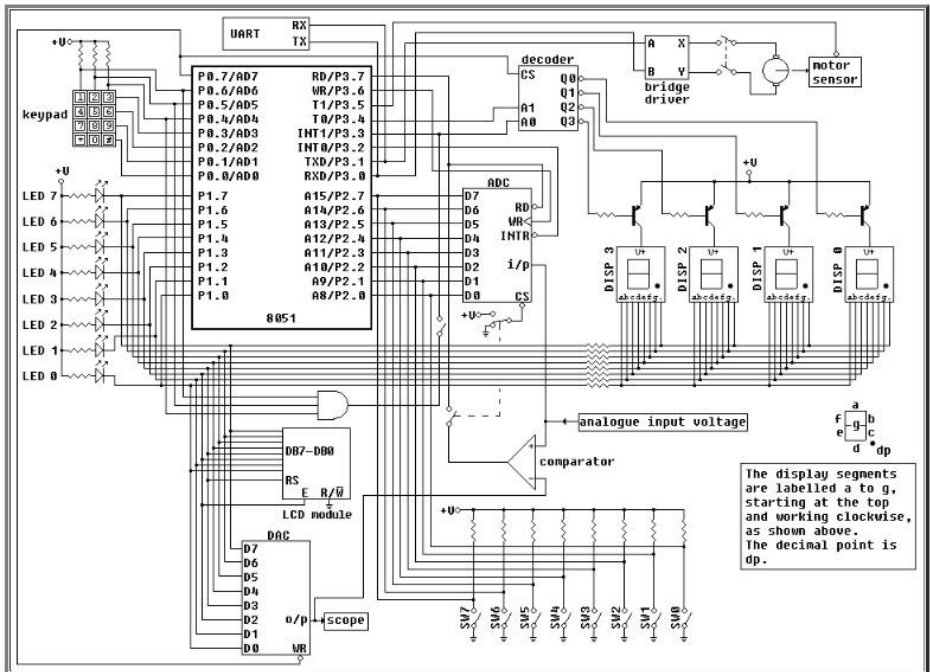


Рисунок 1.2 – Схема мікроконтролера на базі мікропроцесора i8051

На рис. 1.2 зображено, зокрема, наступні компоненти:

- 4 семисегментні індикатори – DISP 0 – DISP 3;
- аналогово-цифровий перетворювач (АЦП) – ADC (Analogue-digital Converter);
- цифро-аналоговий перетворювач (ЦАП) – DAC (Digital-analogue Converter);
- дешифратор (decoder);
- універсальний асинхронний приймач-передавач (UART, Universal Asynchronous Receiver/Transmitter);
- тощо.

Приклад асемблерної програми для i8051, що демонструє використання стеку, подано у лістингу 1.1.

Лістинг 1.1 – Робота з регістрами і стеком



```

start:          ; мітка входження в програму

; завантаження регістрів значеннями
MOV R1, #02H
MOV A, #30H
MOV R2, #00H

; виклик підпрограми
LCALL SUB
SJMP START
SUB:
PUSH PSW ; збереження у стеку значення PSW
PUSH ACC ; збереження у стеку акумулятора
PUSH B ; збереження додаткового акумулятора
ADD A, R1 ; занесення до A значення A+R1
MOV R2, A ; занесення до R2 значення A

POP B ; відновлення зі стеку значення B
POP ACC ; відновлення ACC
POP PSW ; відновлення PSW
RET ; вихід з підпрограми
END

```

Адресний простір регістрів спеціальних функцій (SFR, Special Function Register) мікропроцесора i8051 подано у табл. 1.2 [10].

Таблиця 1.2 – Призначення регістрів мікропроцесора i8051

Адреса	Позначення	Призначення
0E0h	ACC	Акумулятор;
0F0h	B	Розширювач акумулятора;
0D0h	PSW	Слово стану програми (Program Status Word);
080h	P0	Порт 0 (SFR P0);
090h	P1	Порт 1 (SFR P1);
0A0h	P2	Порт 2 (SFR P2);
0B0h	P3	Порт 3 (SFR P3);
081h	SP	Показчик стеку (Stack Pointer);
083h	DPH	Старший байт регістра показчика даних DPTR (Data Pointer High);
082h	DPL	Молодший байт регістра показчика даних

		DPTR (Data Pointer Low);
08Ch	TH0	Старший байт таймера 0 (Timer High 0);
08Ah	TL0	Молодший байт таймера 0 (Timer Low 0);
08Dh	TH1	Старший байт таймера 1 (Timer High 1);
08Bh	TL1	Молодший байт таймера 1 (Timer Low 1);
089h	TMOD	Регістр режимів таймерів-лічильників (Timer/Counter Mode Control Register);
088h	TCON	Регістр керування станом таймерів (Timer/Counter Control Register);
0B8h	IP	Регістр керування пріоритетом переривань (Interrupt Priority Control Register);
0A8h	IE	Регістр маски переривання (Interrupt Enable Register);
087h	PCON	Регістр керування потужністю (Power Control Register);
098h	SCON	Регістр керування приймачем/передавачем (Serial Port Control Register);
099h	SBUF	Буфер приймача/передавача (Serial Data Buffer).

Приклад роботи з регістрами і пам'яттю даних подано у лістингу 1.2.

### Лістинг 1.2 – Програма роботи з регістрами і пам'яттю даних

```
START: JMP LOAD
      ORG 30h ; розміщення програми з адреси 30h
LOAD: MOV R0, #41H ; завантаження в R0 адреси
      MOV @R0, #1CH ; запис у пам'ять числа 1CH
      INC R0 ; інкремент покажчика
      MOV @R0, #3FH ; запис у пам'ять числа 3FH
      JMP START ;
END
```

Схема використання арифметичних операцій подана у лістингу 1.3.

### Лістинг 1.3 – Приклад додавання з урахуванням переносу

```

CLR C ; скидання прапорця переносу
LOOP: MOV A, @R0 ; завантаження до A поточного
; байту першого аргументу
ADDC A, @R1 ; додавання байтів з урахуванням
; переносу
MOV @R0, A ; розміщення результату
INC R0 ; просування покажчика
DJNZ R2, LOOP ; лічильник ітерацій

```

При додаванні беззнакових чисел на переповнення вкаже прапорець C, а при додаванні знакових чисел – OV.

Формат регістру слова стану процесора (PSW) подано на рис. 1.3.



Рисунок 1.3 – Формат регістру слова стану процесора

Для роботи із семисегментним індикатором використовуються біти 3 і 4 порту 3. Приклад програми подано у лістингу 1.4.

Лістинг 1.4 – Робота із семисегментним індикатором

```
start:
```

```

SETB P3.3          ;
SETB P3.4          ; дозволити 3й індикатор
MOV P1, #11111001B ; шаблон 1
CALL delay
CLR P3.3          ; дозволити 2й індикатор
MOV P1, #10100100B ; шаблон 2
CALL delay
CLR P3.4          ;
SETB P3.3          ; дозволити 1й індикатор
MOV P1, #10110000B ; шаблон 3
CALL delay
CLR P3.3          ; дозволити 0й індикатор
MOV P1, #10011001B ; шаблон 4
CALL delay
JMP start

```

delay:

```

MOV R0, #20
DJNZ R0, $
RET

```

## 1.2 Виконання роботи

**Зауваження:** роботу виконувати у середовищі edsim51di.

1.2.1 Робота з пам'яттю і стеком.

1.2.1.1 Керуючись лістингом 1.1, обміняти значеннями регістри R0–R3 із регістрами R4–R7, використовуючи стек.

1.2.1.2 Керуючись лістингом 1.2, послідовно завантажити 10 комірок пам'яті, що починаються з адреси 0x50h, значеннями 10h, 15h і т.д.

1.2.2 Виконання арифметичних операцій.

1.2.2.1 Керуючись лістингом 1.3, накопичити результат складання 10 комірок пам'яті, заповнених при виконанні завдання

1.2.1.2. Одержаний результат зберегти.

1.2.2.2 Скласти два 4-байтні числа.

1.2.3 Створити програму відображення залишку від ділення номеру за списком на 10 на кожному із 4х семи сегментних індикаторів. Шаблон програми подано у лістингу 1.4. Залишок виводити із затримкою. У якості значення затримки вказати власний номер за списком.

### **1.3 Зміст звіту**

1.3.1 Титульний лист.

1.3.2 Мета роботи.

1.3.3 Код програми, створених за результатами виконання завдань 1.2.1, 1.2.2, 1.2.3.

1.3.4 Відповіді на контрольні питання.

### **1.4 Контрольні питання**

1.4.1 Призначення мікроконтролера і8051.

1.4.2 Архітектура мікроконтролера і8051.

1.4.3 Аналоги мікроконтролера і8051.

1.4.4 Схема мікроконтролера і8051.

1.4.5 Призначення портів мікроконтролера і8051.

1.4.6 Регістри мікроконтролера і8051.

1.4.7 Режими роботи таймера мікроконтролера і8051.

1.4.8 Адресація семисегментних індикаторів мікроконтролера і8051.

## ЛАБОРАТОРНА РОБОТА №2

### 2 АРИФМЕТИЧНІ І ЛОГІЧНІ ОПЕРАЦІЇ

**Мета роботи** – набути навичок виконання арифметичних і логічних операцій засобами асемблера i8051.

#### 2.1 Теоретичні відомості

Специфіку виконання арифметичної операції додавання вже було охоплено у попередній роботі. Дана робота, у свою чергу, присвячена виконанню операцій множення, ділення, логічних операцій.

Для здійснення множення призначено команду `MUL`. При цьому перемножуються цілі числа без знаку, які зберігаються у регістрах `A` і `B` [3]. Старший байт результату зберігається у регістрі `B`, молодший – у регістрі `A`. Прапорець переповнення встановлюється, якщо результат множення перевищує значення `0FFh`.

Для ділення цілих чисел без знаку застосовується команда `DIV`. Ділене при цьому має розміщуватися у регістрі `A`, а дільник – у регістрі `B`. У результаті виконання команди ділення ціла частина частки розміщується у регістрі `A`, а залишок – у регістрі `B`. У результаті ділення прапорці `C` і `OV` скидаються. У результаті ділення на 0 прапорець `OV` встановлюється. Застосування команди `DIV` зручне для одержання двійково-десятькового коду числа (Binary-coded Decimal, BCD). Таке представлення числа має ряд переваг, зокрема наступні: зручність виведення розрядів числа на семисегментні індикатори; спрощення введення чисел з цифрової клавіатури; спрощення множення/ділення на 10, округлення. Мають, проте, місце і наступні недоліки: BCD-код менш компактний, ніж HEX-код; ускладнюється виконання арифметичних операцій.

Формат BCD знайшов застосування, зокрема, у телефонії, де незалучені коди (1010 – 1111) задіяно під спеціальні символи: 1010 – \*, 1011 – # тощо.

## 2.2 Виконання роботи

### Зауваження:

- роботу виконувати у середовищі edsim51di;
- для успішного виконання завдання 2.2.1 знадобиться код програми, одержаний у результаті виконання завдання 1.2.2.1 попередньої лабораторної роботи. При цьому його потрібно модифікувати таким чином, щоб програма заповнювала не 10, а 2 комірки пам'яті даних. Він необхідний для заповнення значеннями комірок пам'яті регістра R0, що виступатиме джерелом наповнення значеннями співмножника, розміщеного у регістрі A.

2.2.1 Множення цілих чисел. У програмі лістингу 2.1 виконується множення 2-байтного цілого числа, що зберігається у пам'яті даних за адресою 50H, на константу.

2.2.1.1 Проаналізувати код лістингу 1.2 і покроково виконати відповідний код у емуляторі edsim51di.

### Лістинг 2.1 – Шаблон програми множення чисел

```

START: JMP m1
ORG 30h ; розміщення програми з адреси 30h
m1: mov r1,#2h
MOV R0, #50H ; завантаження в R0 адреси
MOV @R0, #10H ;
LOAD:
mov a,@r0
add a,#5h
INC R0 ; інкремент покажчика
mov @r0,a
djnz r1,load ;

MOV R0, #50H
mov r1,#2h
MOV A,#0 ; скидання акумулятора
m2:   ADD A,@R0           ; завантаження множеного
MOV B,#11h             ; завантаження множника
MUL AB                 ; множення
MOV @R0,A              ; запис молодшого байту
; часткового добутку

```

```

INC R0 ; приріст адреси
MOV A,B ; пересилання старшого байту
; часткового добутку в акумулятор
XCH A,@R0 ; попереднє формування чергового
; байту добутку
DJNZ R1,m2
;
END

```

### **Зауваження:**

– у результаті виконання програми одержуваний результат множення займає у резидентній пам'яті даних на один байт більше.

2.2.1.2 Зафіксувати у звіті зміну значень регістрів А і В у процесі множення – двох проходів циклу. Занести до звіту одержуване значення результату множення, що розміщуватиметься у пам'яті даних із заданої початкової адреси – #50h. Вміти пояснити одержане значення. Занести до звіту назви прапорців, що змінювали значення у процесі множення.

2.2.1.3 Виконати множення цілого бінарного 4-байтного числа на константу, що рівна номеру студента за списком, помноженому на 2. Шаблон програми подано у лістингу 2.1. Результат роботи програми занести до звіту.

2.2.2 Ділення цілих чисел. Програма перетворює HEX-число у VCD-формат.

2.2.2.1 Проаналізувати код лістингу 2.2 і покроково виконати відповідний код у емуляторі edsim51di.

### **Лістинг 2.2 – Шаблон програми ділення чисел**

```

START: JMP m1
ORG 30h ; розміщення програми з адреси 30h
m1:
mov a,#248 ; число, що перетворюємо
mov b,#100 ; для підрахунку сотень
div ab
mov r0,a ; зберегли кількість сотень
xch a,b ; обміняли операнди значеннями

```



```
mov b, #10 ; для підрахунку десятків
div ab
swap a ; обміняти місцями тетради регістра
add a, b
end
```

**Зауваження:**

– результат роботи програми лістингу 2.2 збережеться наступним чином: старша частина – у регістрі r0, молодша – у регістрі a.

2.2.2.2 Модифікувати код лістингу 2.2 таким чином, щоб результат роботи програми зберігався у резидентній пам'яті даних.

2.2.2.3 Модифікувати код лістингу 2.2 таким чином, щоб у якості числа, що підлягає перетворенню, фігурувало наступне число: номер\_студента\_за\_списком \* 10.

## 2.3 Зміст звіту

2.3.1 Титульний лист.

2.3.2 Мета роботи.

2.3.3 Результати виконання завдань 2.2.1.2, 2.2.1.3, 2.2.2.2 і 2.2.2.3.

2.3.4 Відповіді на контрольні питання.

## 2.4 Контрольні питання

2.4.1 Призначення регістрів A і B при множенні операндів.

2.4.2 Перелічити прапорці, які змінюють своє значення при виконанні програми лістингу 2.1. Пояснити.

2.4.3 Проаналізувати зауваження до лістингу 2.1. Пояснити, чому результат виконання команди MUL займає у пам'яті на один байт більше.

2.4.4 Призначення регістрів A і B при виконанні ділення.

2.4.5 Прокоментувати поведінку прапорців при діленні.

2.4.6 Переваги і недоліки VCD-формату представлення чисел.

2.4.7 Призначення команди swap лістингу 2.2.

## ЛАБОРАТОРНА РОБОТА №3

### 3 ПРОГРАМУВАННЯ ТАЙМЕРІВ-ЛІЧИЛЬНИКІВ

**Мета роботи** – набути навичок програмування системного таймера мікроконтролера i8051.

#### 3.1 Теоретичні відомості

Мікроконтролер i8051 включає два таймери-лічильники – Timer 0 і – Timer 1. Кожен з названих таймерів може функціонувати в одному з чотирьох режимів:

- 0 – 13-бітний таймер;
- 1 – 16-бітний таймер;
- 2 – 8-бітний лічильник;
- 3 – Timer 0 функціонує як два 8-бітних таймери.

Таймери-лічильники можуть використовуватися для наступного:

- генерації періодичних часових затримок;
- підрахунку подій (режим лічильника);
- задання бодової швидкості (кількості біт, що передаються за секунду).

Приклад програми генерації таймером затримки подано у лістингу 3.1.

#### Лістинг 3.1 – Шаблон генерації затримки таймером

```
start:
    MOV TMOD,#01H ; ініціалізувати TMOD
    MOV TL0,#0faH ; ініціалізувати TL0
    MOV Th0,#0ffH ; ініціалізувати TH0
    SETB TR0 ; запустити таймер
    Wait: JNB TF0,Wait ; wait for TF0
    CLR TR0 ; stop timer
    CLR TF0 ; clear TF0
RET
```

### 3.2 Виконання роботи

#### Зауваження:

- роботу виконувати у середовищі edsim51di.
- для визначення результуючої затримки програму рекомендується виконувати покроково.

3.2.1 Визначити значення затримки, яка задається програмою, поданою, у лістингу 3.1.

3.2.2 Створити програму генерування затримки у 100 мкс таймером-лічильником Timer 0.

3.2.3 Створити програму генерування затримки згідно варіанту, де час затримки визначається як добуток номеру студента за списком і 10.

3.2.4 На панелі інструментів емулятора (лівий верхній кут) змінити частоту тактового генератора із базових 12 МГц на 6 МГц. Знову виконати завдання 3.2.3. Пояснити нове значення затримки.

### 3.3 Зміст звіту

3.3.1 Титульний лист.

3.3.2 Мета роботи.

3.3.3 Значення результуючої затримки для лістингу 3.1. Код програм, створених за результатами виконання завдань 3.2.2 і 3.2.3.

3.3.4 Відповіді на контрольні питання.

### 3.4 Контрольні питання

3.4.1 Прокоментувати кожен із чотирьох режимів роботи системних таймерів.

3.4.2 Назвати шляхи застосування таймерів-лічильників.

3.4.3 Призначення і використання регістру TMOD.

3.4.4 Призначення і використання регістру TCON.

3.4.5 Призначення байтів TL0, TH0 і TL1, TH1 таймерів-лічильників.

3.4.6 Прокоментувати результат виконання завдання 3.2.4.

## ЛІТЕРАТУРА

1. Programming Timers on 8051 [Електронний ресурс]. – Режим доступу: [http://www.8051projects.net/files/public/1252055169\\_5507\\_FT25871\\_12\\_timers\\_and\\_counters.pdf](http://www.8051projects.net/files/public/1252055169_5507_FT25871_12_timers_and_counters.pdf). – Заголовок з екрану.
2. 5.8051 Timer programming [Електронний ресурс]. – Режим доступу: [https://explorembedded.com/wiki/5.8051\\_Timer\\_programming](https://explorembedded.com/wiki/5.8051_Timer_programming). – Заголовок з екрану.
3. Жабін В.І., Жуков І.А., Ткаченко В.В., Клименко І.А. Мікропроцесорні системи: навч. посібн. К.: НАУ, 2009. 492 с.
4. Куліковський Б.Б., Николайчук Я.М., Шатний С.В. Мікропроцесорні системи. Практикум: навч. посібн. Рівне: НУВГП, 2016. 191 с.
5. Герасимяк Р.П. Теорія автоматичного керування. Одеса: Наука і техніка, 2003. 108 с.
6. Попович М.Г., Ковальчук О.В. Теорія автоматичного керування: підручник. 2-ге вид., перероб. і доп. К.: Либідь, 2007. 656 с.
7. Петров И.В. Программируемые контроллеры. Стандартные языки и приемы прикладного проектирования. М.СОЛОН-Пресс, 2004.
8. Савин М.М., Елсуков В.С., Пятина О.Н. Теория автоматического управления: учеб. пособие. Ростов Н/Д: Феникс, 2007. 469 с.
9. Електронний посібник з дисципліни "Мікропроцесорні системи" [Електронний ресурс]. – Режим доступу: <http://ms.ptngu.com/01.html>. – Заголовок з екрану.
10. Горюнов А.Г., Ливенцов С.Н. Архитектура микроконтроллера Intel 8051: учебн. пособие. Томск: ТПУ, 2005. – 86 с.