

Тернопільський національний технічний  
університет імені Івана Пулюя

---

Кафедра автоматизації  
технологічних процесів  
і виробництв

Лабораторна робота № 7  
з курсу  
”Мікропроцесорні та програмні  
засоби автоматизації”

Ввід-вивід даних через  
порти PIC-контролера

Тернопіль 2018

Методичні вказівки до лабораторної роботи №7 “Ввід-вивід даних через порти РС-контролера” з курсу “Мікропроцесорні та програмні засоби автоматизації”  
Медвідь В.Р., Пісьціо В.П., - Тернопіль: ТНТУ, 2018 - 14 с.

Відповідальні за випуск

доцент, к.т.н. Медвідь В.Р.,

асистент Пісьціо В.П.

Для студентів напрямку: 151 "Автоматизація та комп'ютерно-інтегровані технології"

## Лабораторна робота №7

### Вивід/вивід даних через порти PIC-контролера на програмному симуляторі PIC Simulator IDE

#### 1. Послідовність роботи з програмним симулятором PIC Simulator IDE

Основне вікно програми PIC Simulator IDE має вигляд, показаний на (рис. 1).

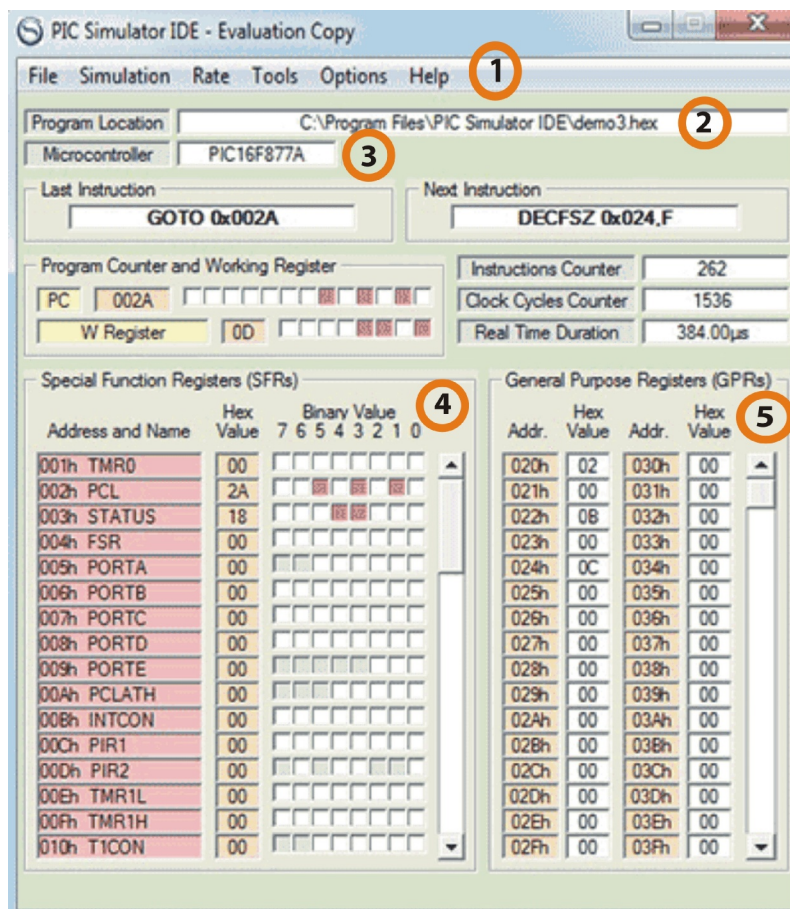


Рис. 1. Основне вікно програми PIC Simulator IDE

У верхній частині знаходяться меню, через які можна отримати доступ до основних і додаткових модулів програми (поз. 1) (рис. 1).

В рядку Program Location вказано шлях до обраної програми і її ім'я (поз. 2).

В рядку Microcontrollers, відображається тип обраного мікроконтролера (поз. 3).

У нижній частині вікна є дві панелі (поз.4 і поз.5), в яких відображаються стан програми, вміст регістрів спеціальних функцій (PCФ) і керуючих регістрів обраного МК.

#### Послідовність роботи з програмним симулятором наступний:

- запуск програми PIC Simulator IDE;
- вибір типу мікроконтролера, для якого написана програма;
- вибір частоти кварцового генератора (впливає тільки на відображувані програмою дані про час виконання програми або команди, але не на швидкість роботи програми, що налагоджуються в PIC Simulator IDE);
  - завантаження програми у вигляді HEX-файлу або запуск вбудованого компілятора мови асемблера і написання в ньому потрібної програми;
- вибір потрібних модулів віртуальних пристроїв;
- вибір швидкості і режиму роботи програми симулятора;
- запуск процесу симуляції роботи програми на обраному МК.

Якщо потрібно скористатися для роботи з симулятором власною програмою або внести зміни у вже розроблену, необхідно створити або завантажити для цього файл асемблера, з якого після компіляції буде створений необхідний для роботи з симулятором hex-файл.

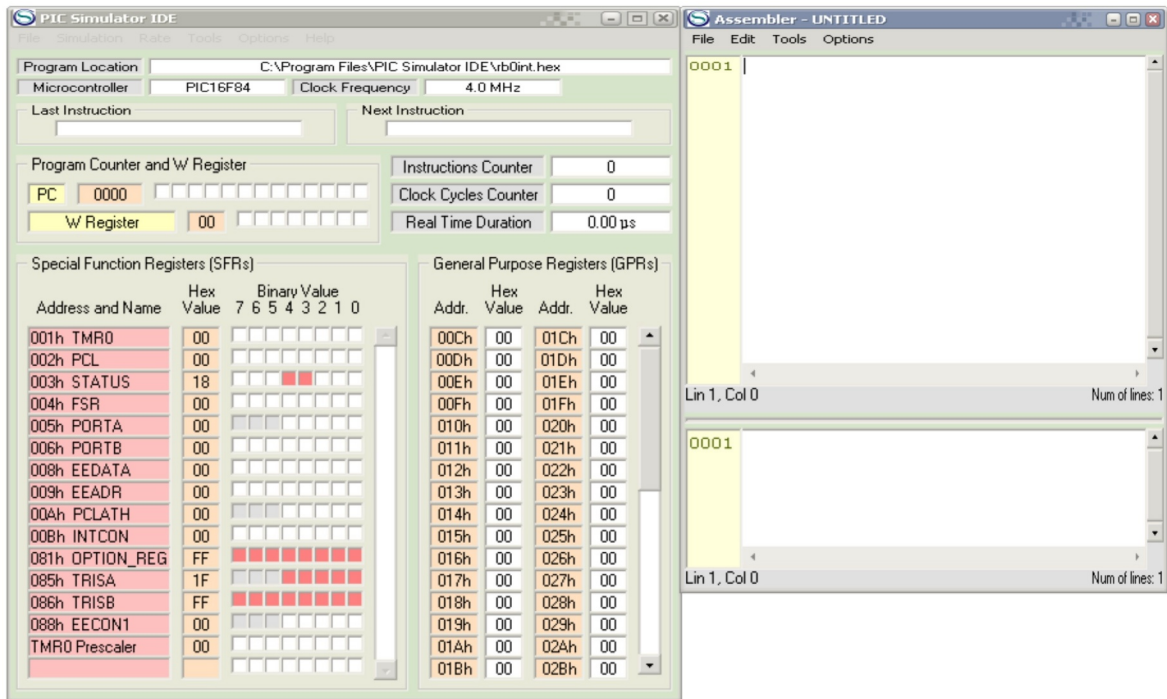


Рис. 2 Вікно симулятора з відкритим вікном Assembler

Для цього:

1. Натиснути Options | Assembler. Відкриється вікно компілятора Assembler – UNTITLED (рис. 2);
2. У вікні Assembler натиснути опцію File. Розкриється закладка (рис. 3), з якої для створення нового файлу потрібно натиснути New, а для завантаження вже створеного – OPEN.

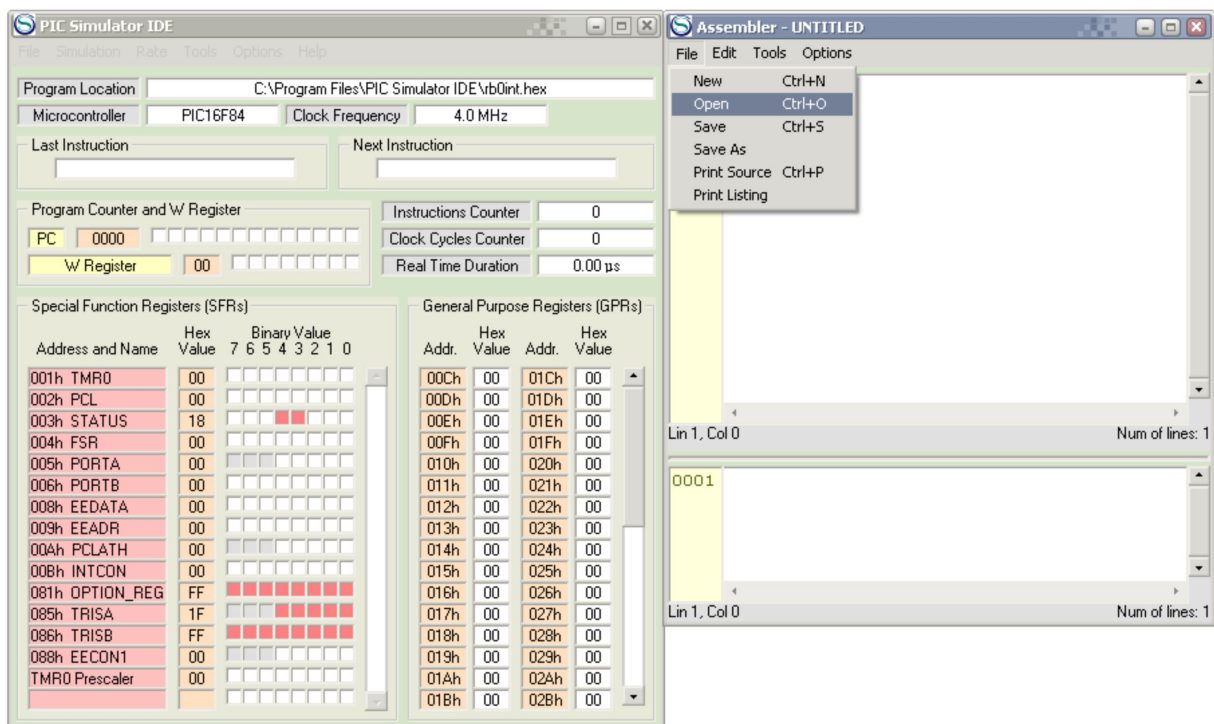


Рис. 3 Завантаження існуючого або створення нового файлу асемблера

3. Після вибору і завантаження файлу (наприклад, rb0int.asm), його текст з'явиться у вікні Assembler (рис. 4).

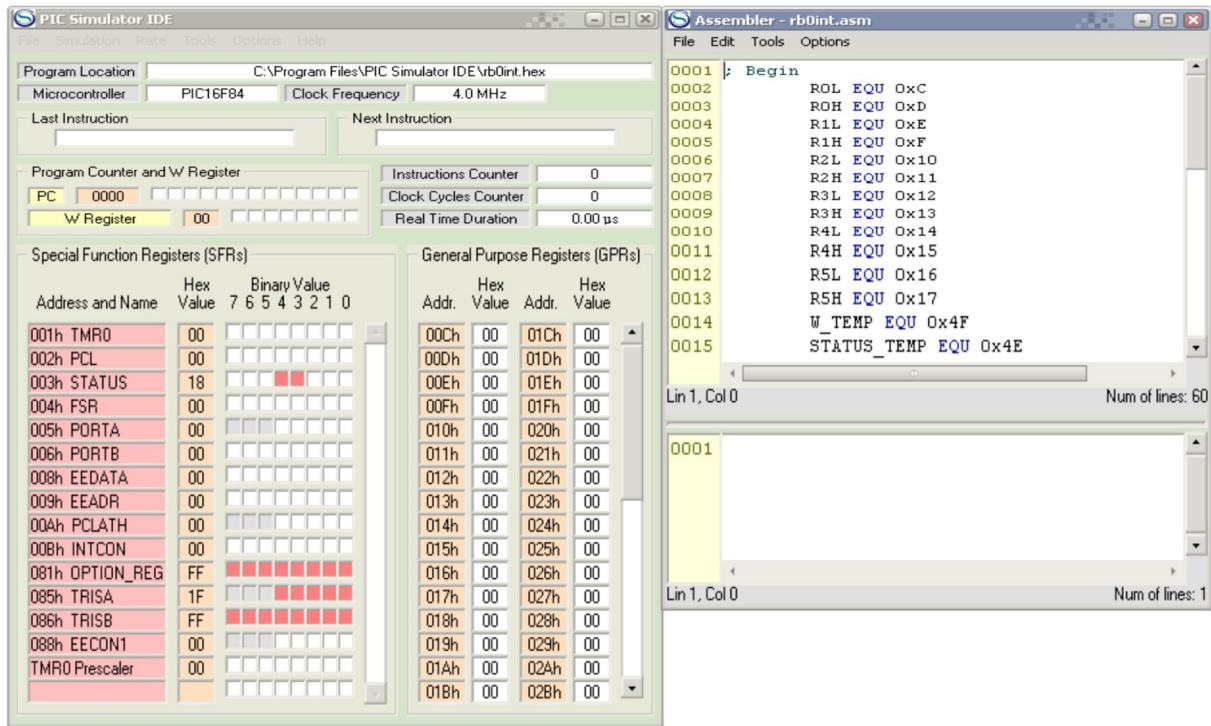


Рис. 4 Завантаження файлу rb0int.asm

4. Для компіляції створеного або завантаженого і потім зміненого файлу, натисніть Tools і у вікні, що розкриється – Assemble. В нижній половині вікна Assembler з'явиться лістинг відкомпільованого файлу і, одночасно, при відсутності помилок, буде створений одноіменний hex-файл.

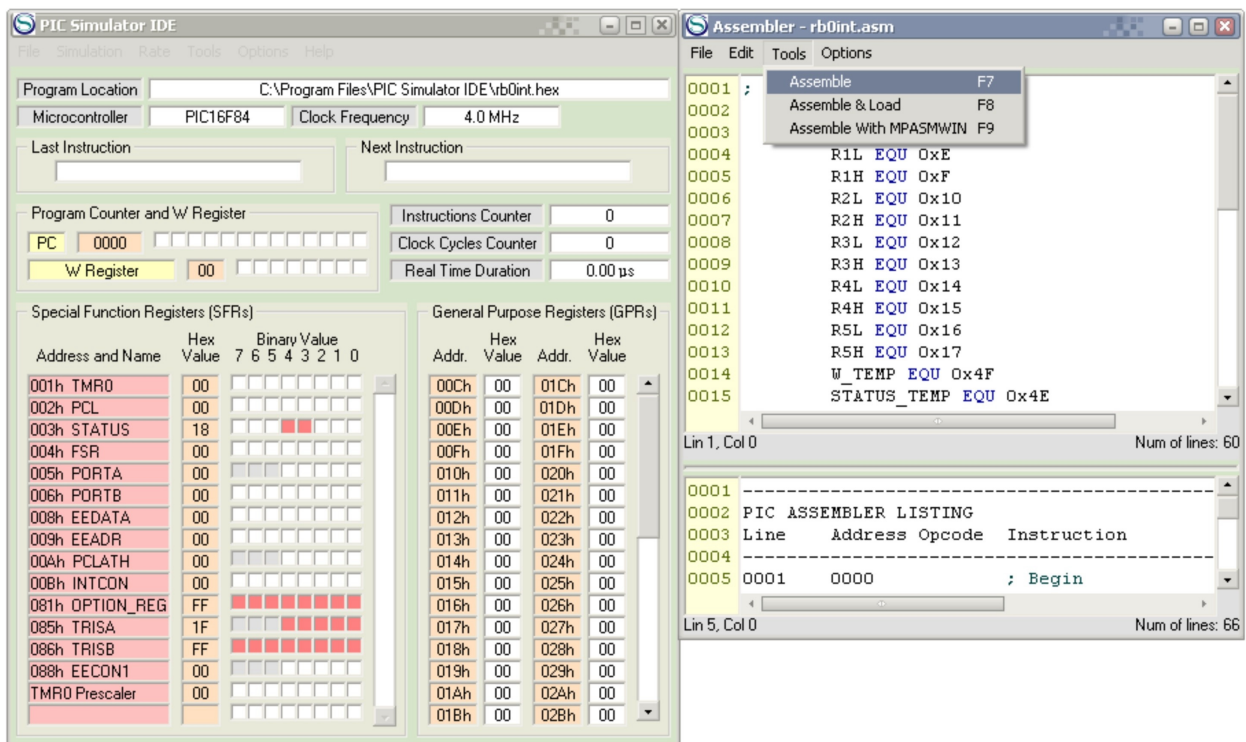


Рис. 5

**2. Завдання на лабораторну роботу:** ввід-вивід даних через порти PIC-контролера.

1. Вивчити програмну модель PIC Simulator IDE.
2. Вивчити команди програмування та обміну даними через порти PIC – контролера.
3. Дослідити роботу програми з Прикладу 1 в режимі роботи стимулятора «Normal» та вміст регістрів контролера (W, STATUS...), які використовуються при виконанні цієї програми, в покроковому режимі роботи.
4. Записати для вибраних команд асемблера коментар щодо їх призначення (див. Приклад 1).

### Приклад 1

Програма забезпечує ввід даних з чотирьох молодших розрядів порту В, зсув їх на чотири розряди вліво і вивід через чотири старші розряди цього ж порту.

Текст програми з файлу «Input\output.asm» має наступний вигляд:

```
; Begin
    ORG 0x0000
    BCF PCLATH,3
    BCF PCLATH,4
    BSF STATUS,RP0
    MOVLW 0x80
    MOVWF OPTION_REG
    MOVLW 0x0F
    MOVWF TRISB
    BCF STATUS,RP0
SHIFTING:
    SWAPF PORTB,W
    MOVWF PORTB
    GOTO SHIFTING
END
```

### 3. Послідовність роботи з симулятором при виконанні програми

Виконаємо цю програму в PIC Simulator ID, для чого необхідно:

1. Запустити PIC Simulator IDE;
2. Натиснути Options | Select Microcontroller;
3. Вибрати PIC16F84 і натиснути кнопку Select;
4. Натиснути Tools і у вікні, що розкриється, вибрати «Assembler». Відкриється вікно компілятора «Assembler – UNTITLED» (рис. 2);
5. Набрати текст програми Прикладу 1 у вікні «Assembler»;
6. Натиснути Tools і у вікні, що розкриється – Assemble. В нижній половині вікна Assembler з'явиться лістинг відкомпільованого файлу (рис. 6);
7. Одночасно, при відсутності помилок, буде створений файл «Input\output.hex», для якого можна вибрати ім'я та шлях для запису. Записати його на «Робочий стіл» комп'ютера;
8. Вибрати File | Load Program і завантажити створений файл «Input\output.hex»;
9. Натиснути Tools | 8 x LED Board. Відкриється вікно з панеллю, що містить вісім світлодіодів (рис. 7);

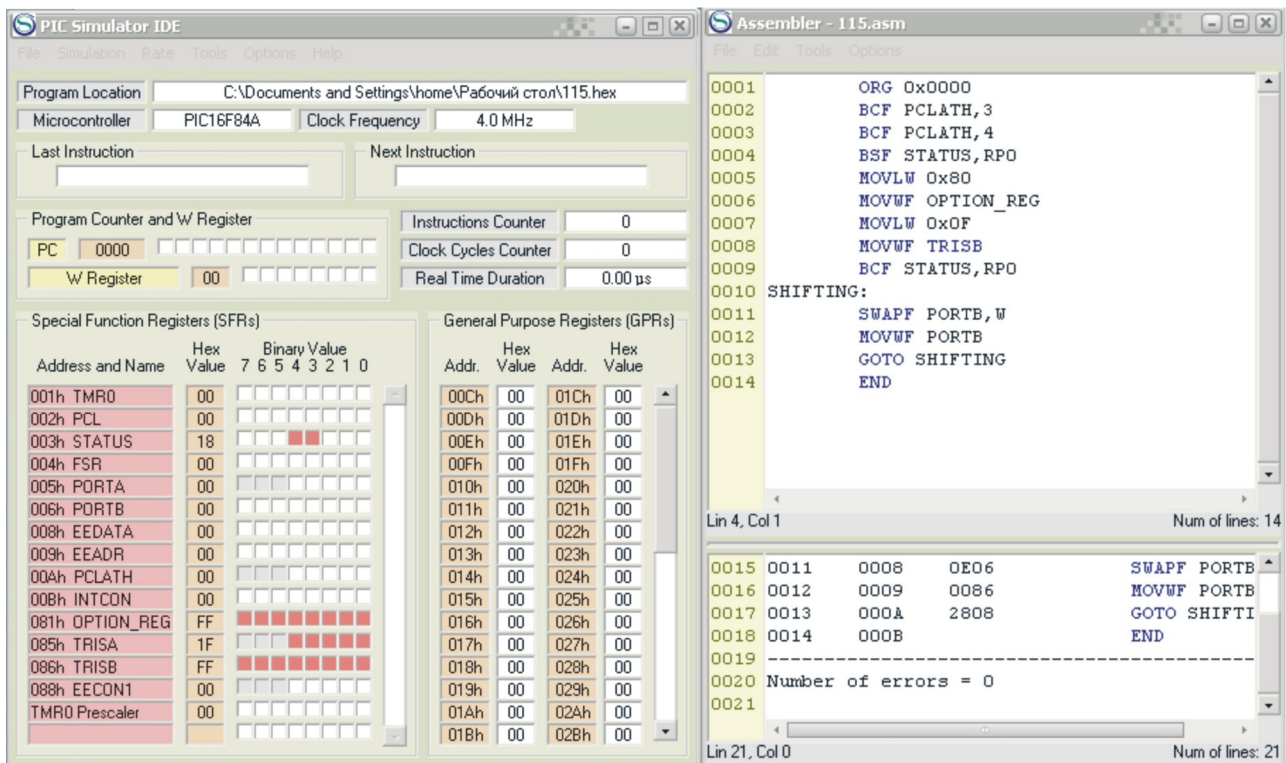


Рис. 6 Вигляд інтерфейсу симулятора з програмою «Input/output»

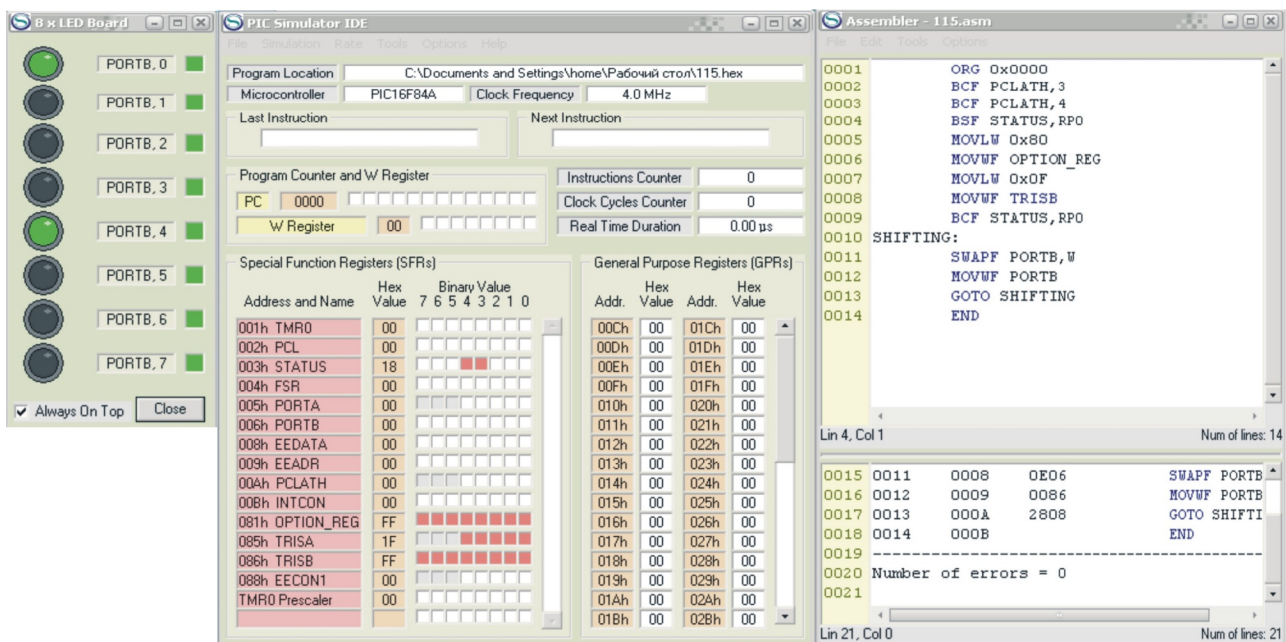


Рис. 7 Вигляд інтерфейсу симулятора з програмою та панеллю «8 x LED Board»

10. Натиснути Tools | Microcontroller View PIC16F84. Відкриється вікно з виводами мікроконтролера (рис. 8);

11. У вікні «Select Pin» по чергово натиснути поле «PORTB» і далі «0», після чого натиснути на поле «Select», яке розташоване внизу вікна. Таким чином, вибрано порт B та його вивід 0. Це повторити для всіх ліній вибраного порту;

12. Вибрати Rate | Normal;

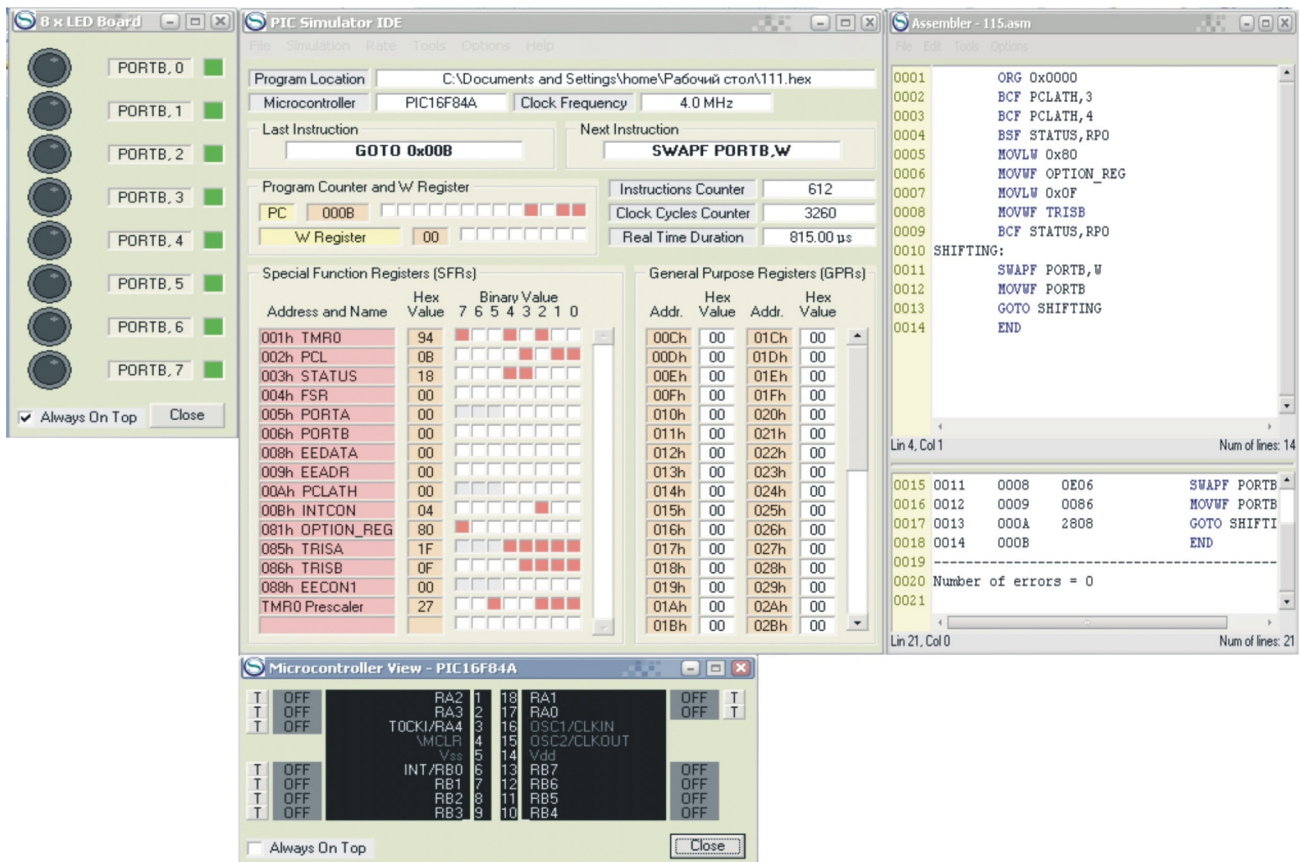


Рис. 8 Вигляд інтерфейсу симулятора з програмою та панеллю «Microcontroller View PIC16F84»

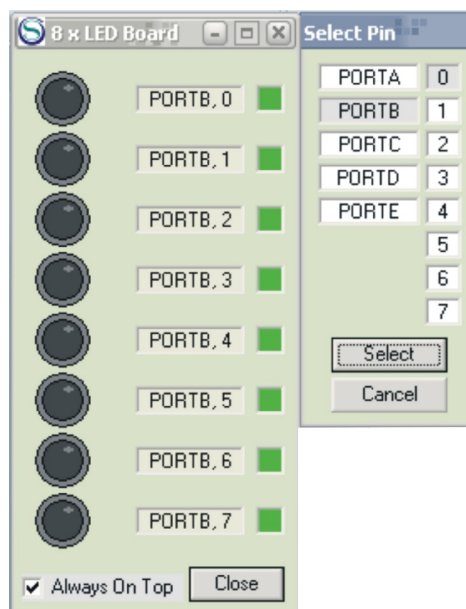


Рис. 9 Налаштування виводів порту В

13. Натиснути Simulation | Start (почнеться виконання програми). Якщо при цьому курсором клацнути на одному з виводів мікросхеми (панель «Microcontroller View PIC16F84») за номером  $n$  (це відповідає появі на цьому виводі логічної «1» - світлодіод світиться), то засвітиться світлодіод на виводі мікросхеми з номером  $n+4$  (рис. 10);

14. Щоб зупинити виконання програми, потрібно натиснути Simulation | Stop.



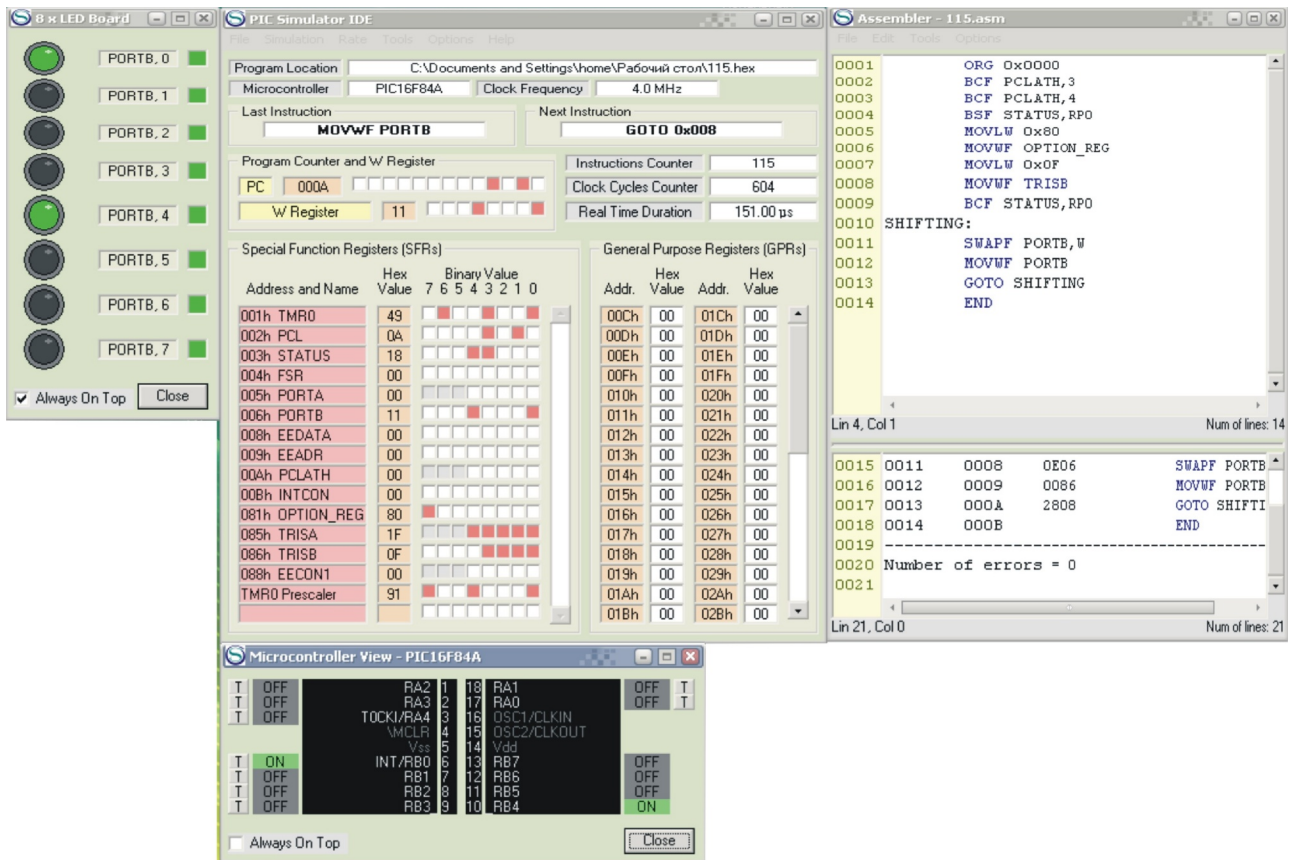


Рис. 10 Зображення інтерфейсу стимулятора з виконуваною програмою

Для того, щоб мати змогу контролювати вміст регістрів після виконання стимулятором кожної команди, перейти на виконання програми в кроковому режимі роботи.

Для цього:

1. В основному вікні симулятора натиснути Rate | Step By Step, а далі вибрати опцію Simulation і натиснути Start. Симулятор готовий до виконання програми в кроковому режимі;
2. Для виконання наступної команди програми потрібно натиснути на закладку STEP, яка з'явиться справа від закладки HELP вгорі основного вікна симулятора після вибору крокового режиму його роботи.

Вміст регістрів контролера, які використовуються при виконанні команд програми, знайти в області регістрів Address and Name, яка розташована в лівій нижній частині основного вікна симулятора (виділені рожевим кольором). Всі регістри восьмирозрядні.

В процесі виконання програми по зміні кольору комірок видно, вміст яких регістрів змінюється. Забарвлення комірки відповідного розряду регістру помаранчевим кольором означає наявність "1", білим - "0".

#### 4. Завдання для виконання лабораторної роботи

##### Завдання 1

1. Виконати програму (Приклад 1) в режимі «Normal». В процесі виконання програми клацнути курсором на одному з виводів чотирьох молодших розрядів порту В мікросхеми (панель «Microcontroller View PIC16F84») за номером  $n=0...3$  (це відповідає появі на цьому виводі логічної «1» - світлодіод світиться). При цьому засвітиться світлодіод на виводі мікросхеми з номером  $n+4$  (рис. 10);

2. Виконати програму в кроковому режимі виконання програми. Вміст тих регістрів, значення яких змінюється в процесі виконання команд програми, записати в шістнадцятковому коді в табл.1.

Таблиця 1

Регістр	PC	W	TMR0	STATUS	PCL	TRISA	TRISB	PCLATH	EEADR	EEDATA	FSR
Команда 1											
Команда 2											
.....											
Команда n											

3. З програми Прикладу 1 вибрати десять команд і за таблицею команд асемблера для PIC-контролера (табл. 1) записати коментар щодо призначення цих команд (див. Приклад 2, де наведено такий запис для однієї команди).

### Приклад 2

#### Код команди

118A

#### Команда

BCF PCLATH, 3

#### Виконувана операція (коментар)

; скинути в "0" 3-ій біт регістра PCLATH

і т.д.

### Завдання 2

1. Скласти програму, яка забезпечує ввід даних з **чотирьох старших розрядів порту В**, зсув їх на чотири розряди вправо і вивід через **чотири молодші розряди цього ж порту**.

2. Виконати програму в режимі «Normal».

3. В процесі виконання програми клацнути курсором на одному з виводів чотирьох старших розрядів порту В мікросхеми (панель «Microcontroller View PIC16F84») за номером  $n=4...7$ . При цьому повинен засвітитися світлодіод на виводі мікросхеми з номером  $n-4$  (рис. 10);

4. Виконати програму в кроковому режимі виконання програми.

5. Вміст тих регістрів, значення яких змінюється в процесі виконання команд програми, записати в шістнадцятковому коді в табл. 2.

6. З виконуваної програми вибрати десять команд і за таблицею команд асемблера для PIC-контролера (таблиця 3) записати коментар щодо призначення цих команд (див. Приклад 2, де наведено такий запис для однієї команди).

### Завдання 3

1. Скласти програму, яка забезпечує ввід даних з **чотирьох молодших розрядів порту А**, та вивід через **чотири молодші розряди порту В**.

2. Виконати програму в режимі «Normal».

3. В процесі виконання програми клацнути курсором на одному з виводів чотирьох молодших розрядів порту А мікросхеми (панель «Microcontroller View PIC16F84») за номером  $n=0...3$ . При цьому повинен засвітитися світлодіод на виводі мікросхеми з номером  $n$  порту В (рис. 10);

4. Виконати програму в кроковому режимі виконання програми.

5. Вміст тих регістрів, значення яких змінюється в процесі виконання команд програми, записати в шістнадцятковому коді в табл. 2.

6. З виконуваної програми вибрати десять команд і за таблицею команд асемблера для PIC-контролера (таблиця 3) записати коментар щодо призначення цих команд (див. Приклад 2, де наведено такий запис для однієї команди).

### Завдання 4

Програма забезпечує ввід даних з чотирьох молодших розрядів порту А і вивід через чотири старші розряди порту В.

Текст програми має наступний вигляд:

```

; Begin
    ORG 0x0000
    BCF PCLATH,3
    BCF PCLATH,4
    BSF STATUS,RP0
    MOVLW 0x80
    MOVWF OPTION_REG
    MOVLW 0x0F
    MOVWF TRISA
    MOVLW 0xF0
    MOVWF TRISB
    BCF STATUS,RP0
SHIFTING:
    MOVF PORTA,W
    MOVWF PORTB
    GOTO SHIFTING
    END

```

### 5. Контрольні запитання

1. Будова PIC-контролерів серії PIC16X8X.
2. Призначення регістрів спеціальних функцій мікроконтролера.
3. Як запрограмувати порти мікроконтролера на ввід або вивід?
4. Формат регістра STATUS. Призначення бітів в процесі програмування портів мікроконтролера.
5. Які файли створюються в результаті компілювання програми на асемблері?

### 5. Література

1. Данилин А. Программа-симулятор PIC Simulator IDE / Данилин А. // Современная электроника. 2006.- №4. -С. 68-76.
2. Тавернье К. PIC-микроконтроллеры. Практика применения. М.: ДМК, 2002.
3. Предко М. Создайте работа своими руками на PIC- контроллере./ Майкл Предко; Пер. с английского Земского Ю.В. – М.: ДМК Пресс, 2006. – 408 с.: ил.
4. Кениг А. и М. Полное руководство по PIC-микроконтроллерам.: Пер. с нем.-К.: МК- Пресс”, 2007.-256 с., ил.

Таблиця 3. Команди контролера PIC16X8X

Позначення	Функція	Цикли	Код команди	Біти стану	Прим.
<b>ADDLW</b>	Додавання константи і W	1	11111x	C, DC, Z	
<b>ADDWF</b>	Додавання W с f	1	00 0111 dfff ffff	C, DC, Z	1, 2
<b>ANDLW</b>	Логічне І константи і W	1	11 1001 kkkk kkkk	Z	
<b>ANDWF</b>	Логічне І W і f	1	00 0101 dfff ffff	Z	1, 2
<b>BCF</b>	Скидання біту в регістрі f	1	01 00bb bfff ffff		1, 2
<b>BSF</b>	Встановлення біту в регістрі f	1	01 01bb bfff ffff		1, 2
<b>BTFSZ</b>	Пропустити команду, якщо біт у f дорівнює нулю	1 (2)	01 10bb bfff ffff		3
<b>BTFS</b>	Пропустити команду, якщо біт у f дорівнює одиниці	1 (2)	01 11bb bfff ffff		3
<b>CALL</b>	Виклик підпрограми	2	10 0kkk kkkk kkkk		
<b>CLRF</b>	Скидання регістру f	1	00 0001 1fff ffff	Z	2
<b>CLRW</b>	Скидання регістра W	1	00 0001 0xxx xxxx	Z	
<b>CLRWDT</b>	Скидання сторожового таймера WDT	1	00 0000 0110 0100		
<b>COMF</b>	Інверсія регістру f	1	00 1001 dfff ffff	Z	1, 2
<b>DECF</b>	Декремент регістру f	1	00 0011 dfff ffff	Z	1, 2
<b>DECFSZ</b>	Декремент f, пропустити команду, якщо 0	1 (2)	00 1011 dfff ffff		1, 2, 3
<b>GOTO</b>	Перехід за адресою	2	10 1kkk kkkk kkkk		
<b>INCF</b>	Інкремент регістру f	1	00 1010 dfff ffff	Z	1, 2
<b>INCFSZ</b>	Інкремент f, пропустити команду, якщо 0	1 (2)	00 1111 dfff ffff		1, 2, 3
<b>IORLW</b>	Логічне АБО константи і W	1	11 1000 kkkk kkkk	Z	
<b>IORWF</b>	Логічне АБО W і f	1	00 0100 dfff ffff	Z	1, 2
<b>MOVF</b>	Пересилання регістру f	1	00 1000 dfff ffff	Z	1, 2
<b>MOVLW</b>	Пересилання константи в W	1	11 00xx kkkk kkkk		
<b>MOVWF</b>	Пересилання W у f	1	00 0000 1fff ffff		
<b>NOP</b>	Пуста команда	1	00 0000		

<b>OPTION</b>	Завантаження регістру OPTION	1	00 0000 0110 0010		
<b>RETFIE</b>	Повернення з переривання	2	00 0000 0000 1001		
<b>RETLW</b>	Повернення з підпрограми з заванта- женням константи в W	2	11 01xx kkkk kkkk		
<b>RETURN</b>	Повернення з підпрограми	2	00 0000 0000 1000		
<b>RLF</b>	Зсув f вліво через перенесення	1	00 1101 dfff ffff	C	1, 2
<b>RRF</b>	Зсув f вправо через перенесення	1	00 1100 dfff ffff	C	1, 2
<b>SLEEP</b>	Перехід у режим SLEEP	1	00 0000 0110 0011		
<b>SUBLW</b>	Вирахування W з константи	1	11 110x kkkk kkkk	C, DC, Z	
<b>SUBWF</b>	Вирахування W з f	1	00 0010 dfff ffff	C, DC, Z	1, 2
<b>SWAPF</b>	Обмін місцями тетрад в f	1	00 1110 dfff ffff		1, 2
<b>TRIS</b>	Завантаження регістру TRIS	1	00 0000 0110 0fff		
<b>XORLW</b>	Виключаюче АБО константи і W	1	11 1010 kkkk kkkk	Z	
<b>XORWF</b>	Виключаюче АБО W і f	1	00 0110 dfff ffff		1, 2

\*Якщо в результаті виконання команди змінюється лічильник команд, або виконується перехід по перевірці умови, то команда виконується за два цикли. Другий цикл виконується як NOP.

**Завдання 3**

Програма забезпечує ввід даних з чотирьох молодших розрядів порту А і вивід через чотири старші розряди порту В.

Текст програми має наступний вигляд:

```
; Begin
    ORG 0x0000
    BCF PCLATH,3
    BCF PCLATH,4
    BSF STATUS,RP0
    MOVLW 0x80
    MOVWF OPTION_REG
    MOVLW 0x0F
    MOVWF TRISA
    MOVLW 0xF0
    MOVWF TRISB
    BCF STATUS,RP0
SHIFTING:
    MOVF PORTA,W
    MOVWF PORTB
    GOTO SHIFTING
END
```