

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 “Робота з
аналого-цифровим перетворювачем на програмному
симуляторі PIC Simulator IDE”
з курсу “Проектування мікропроцесорних систем
керування технологічними процесами”

Тернопіль 2017

Методичні вказівки для виконання лабораторної роботи №4 «Робота з аналого-цифровим перетворювачем на програмному симуляторі PIC Simulator IDE» з курсу «Проектування мікропроцесорних систем керування технологічними процесами».

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

Відповідальні за випуск

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

Лабораторна робота №4

Робота з аналого-цифровим перетворювачем на програмному симуляторі PIC Simulator IDE

1. Робота з програмним симулятором PIC Simulator IDE

Запустивши на виконання PIC Simulator IDE, побачимо основне вікно цієї програми (рис. 1).

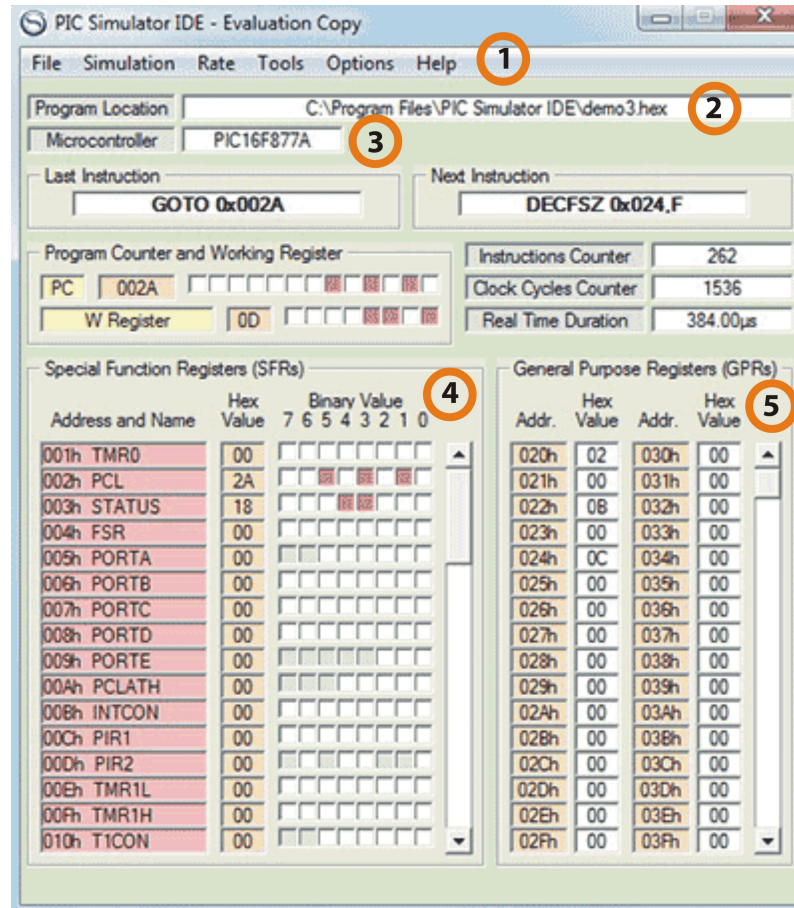


Рис. 1. Основне вікно програми PIC Simulator IDE

У верхній частині знаходяться різні меню, через які можна отримати доступ до різних основних і додаткових модулів програми (на рис. 1 позначено як «1»).

Далі, в рядку Program Location вказано шлях до обраної програми і її ім'я (на рис. 1 - «2»).

Нижче, в рядку Microcontrollers, відображається тип обраного мікроконтролера (на рис. 1 - «3»).

У нижній частині вікна є дві панелі (позначені як «4» і «5»). У них відображаються стан програми, вміст спеціальних 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 асемблер 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.

Для цього:

1. Натиснути Options | Assembler. Відкриється вікно компілятора Assembler – UNTITLED (рис. 2);

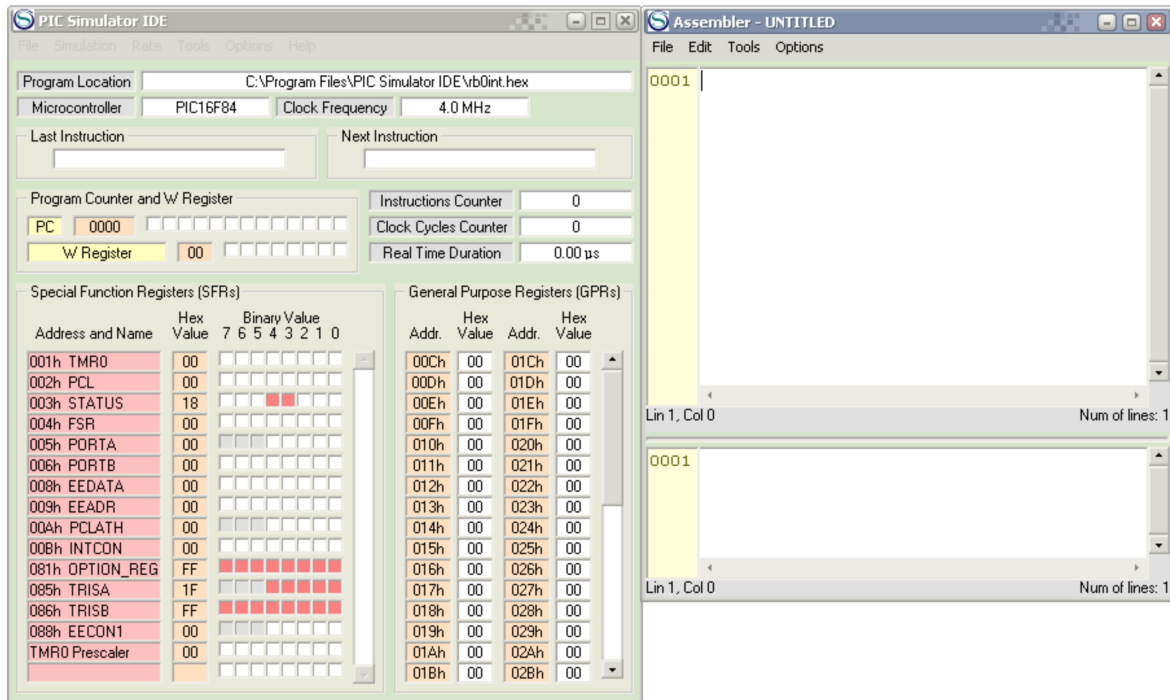


Рис. 2 Вікно симулятора з відкритим вікном Assembler

2. У вікні Assembler натисніть опцію File. Розкриється закладка (рис. 3), з якої для створення нового файлу потрібно натиснути New, а для завантаження вже створеного – OPEN.

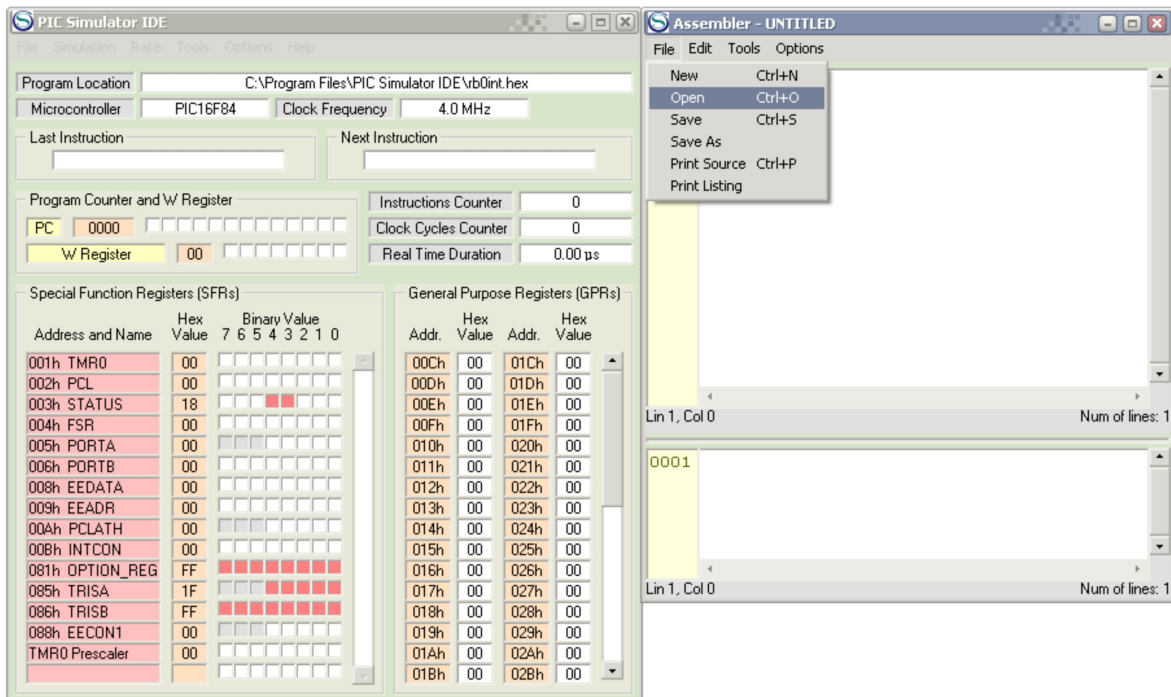


Рис. 3

3. Після вибору і завантаження файлу (наприклад. rb0int.asm), його текст з'явиться в верхній половині вікна Assembler (рис. 4).

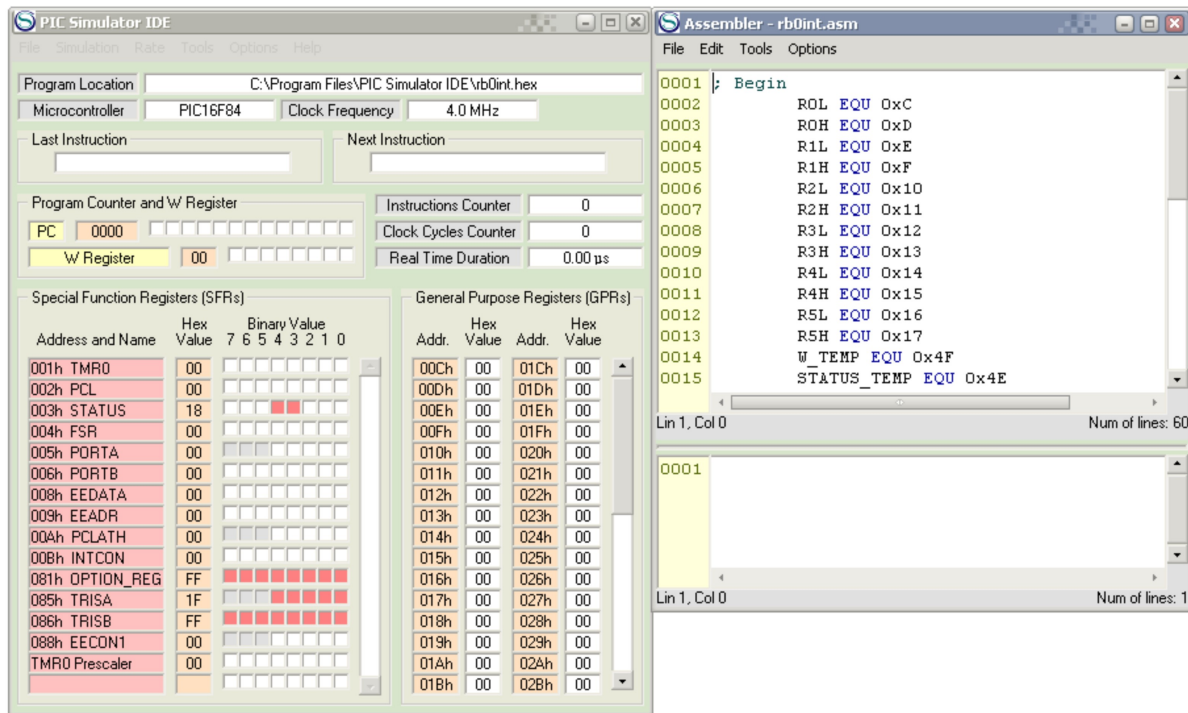


Рис. 4 Завантаження файлу rb0int.asm

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 відкомпільований файл і одночасно, при відсутності помилок, буде створений одноіменний hex-файл.

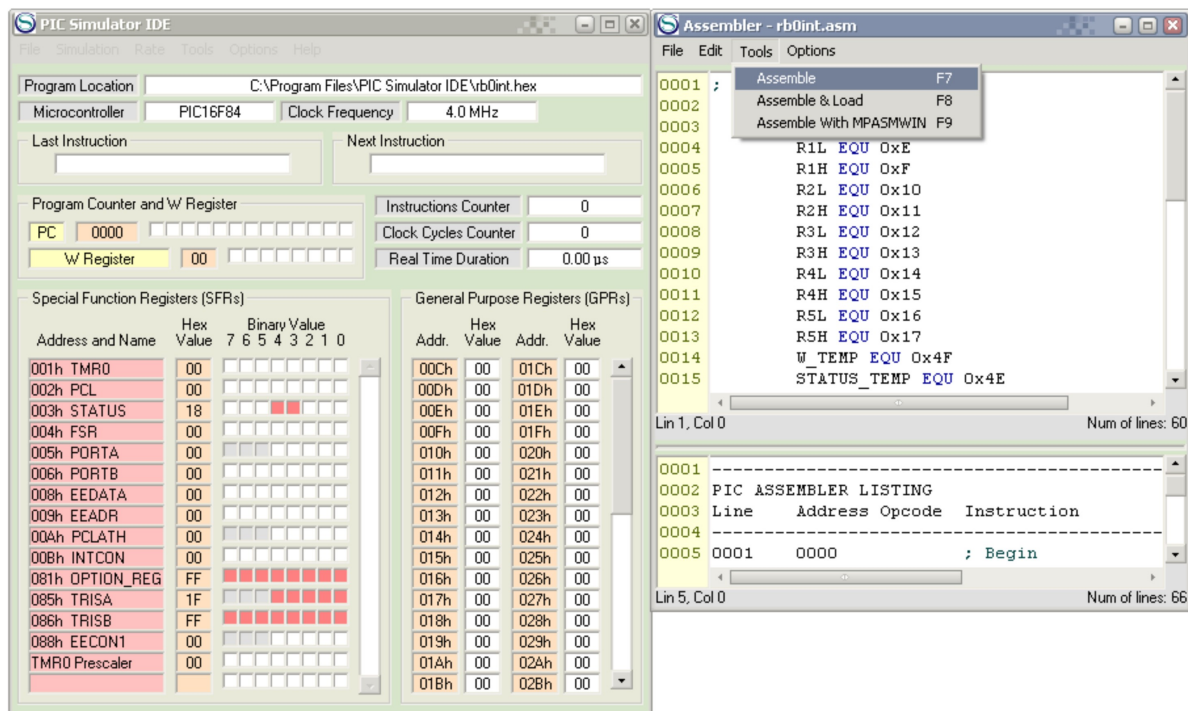


Рис. 5

2. Завдання на лабораторну роботу: математичні операції: підпрограма множення..

1. Вивчити програмну модель PIC Simulator IDE.

2. Вивчити команди арифметичних операцій PIC – контролера.

3. Написати і дослідити роботу програми з Прикладу 1 та дослідити вміст регістрів контролера (PC, W, STATUS), які використовуються при виконанні цієї програми, а також значення на виводах порту RB (RB0...RB7).

4. Користуючись вікном “Program Memory Editor”, записати перші 7 команд виконуваної програми на Асемблері та в шістнадцяткових кодах.

5. Записати для вибраних команд коментар щодо їх призначення (див. Приклад 2).

Приклад 1.

Ця програма читає значення на аналоговому вході AN0 і відображає данні вимірювань на виводи порту PORTB як 8-бітове значення.

Текст програми з файлу adc.asm має наступний вигляд:

; Begin

R0L EQU 0x20

R0H EQU 0x21

R1L EQU 0x22

R1H EQU 0x23

R2L EQU 0x24

R2H EQU 0x25

R3L EQU 0x26

R3H EQU 0x27

R4L EQU 0x28

R4H EQU 0x29

R5L EQU 0x2A

R5H EQU 0x2B

ORG 0x0000

BCF PCLATH,3

BCF PCLATH,4

GOTO L0003

ORG 0x0004

RETFIE

L0003:

; 1: Symbol ad_action = ADCON0.GO_DONE 'set new name for A/D conversion start bit

; The address of 'ad_action' is 0x1F,2

; 2: Symbol display = PORTB 'set new name for PORTB used to display the conversion result

; The address of 'display' is 0x6

display EQU 0x6

; 3:

; 4: TRISB = %00000000 'set PORTB pins as outputs

BSF STATUS,RP0

CLRF 0x06

BCF STATUS,RP0

; 5: TRISA = %111111 'set PORTA pins as inputs

BSF STATUS,RP0

MOVLW 0x3F

MOVWF 0x05

BCF STATUS,RP0

; 6: ADCON0 = 0xc0 'set A/D conversion clock to internal source

MOVLW 0xc0

MOVWF 0x1F


```

; 7: ADCON1 = 0 'set PORTA pins as analog inputs
    BSF STATUS,RP0
    CLRF 0x1F
    BCF STATUS,RP0
; 8: High ADCON0.ADON 'turn on A/D converter module
    BSF 0x1F,0
; 9:
; 10: main:
L0001:
; 11: Gosub getadresult 'go to conversion routine
    CALL L0002
; 12: display = ADRESH 'display the result of the conversion
    MOVF 0x1E,W
    MOVWF 0x06
; 13: Goto main 'repeat forever
    GOTO L0001
; 14: End
L0004:GOTO L0004
; 15:
; 16: getadresult: 'conversion routine
L0002:
; 17: High ad_action 'start the conversion
    BSF 0x1F,2
; 18: While ad_action 'wait until conversion is completed
L0005:
    BTFSS 0x1F,2
    GOTO L0006
; 19: Wend
    GOTO L0005
L0006:MOVLW 0x1F
    ANDWF STATUS,F
; 20: Return
    RETURN
; End of program
L0007:GOTO L0007
; End of listing
    END

```

3. Послідовність роботи з симулятором при виконанні програми

Переглянемо результати роботи цієї програми в PIC Simulator IDE.

Для цього виконаємо наступне, вибравши модель МК PIC16F877:

1. Запустити PIC Simulator IDE;
2. Натиснути Options | Select Microcontroller;
3. Вибрати **PIC16F877A**, який містить АЦП, і натиснути кнопку Select;
4. Натиснути File | Load Program;
5. Вибрати файл adc.hex і натиснути кнопку Open;
6. Натиснути Tools | Microcontroller View (відкриється вікно Microcontroller View);
7. Вибрати Rate | Extremely Fast simulation rate;
8. Натиснути Simulation | Start (почнеться робота програми);
9. Натиснути кнопку «A» перед виводом RA0/AN0. З'явиться білий прямокутник в середній частині панелі «Microcontroller View» з «панеллю прокрутки», повзунком якої

можна змінювати аналогове значення напруги на цьому виводі. Це значення відображається у вікні під написом «AN0» та одночасно перед виводом «AN0/RA0» (на рис. 2 це «435»).

Цифровий код, що відповідає цьому значенню, відображується на лініях порту RB0...RB7 («ON» відповідає логічній «1», «OFF»- логічному «0»);

10. Натиснути кнопку «OK» на полі прокрутки і подивитися, як зміниться стан виводів порту PORTB.

Останні три кроки можна повторити кілька разів і подивитися на результати.

Вигляд екрану з виконуваною програмою показано на рис. 6.

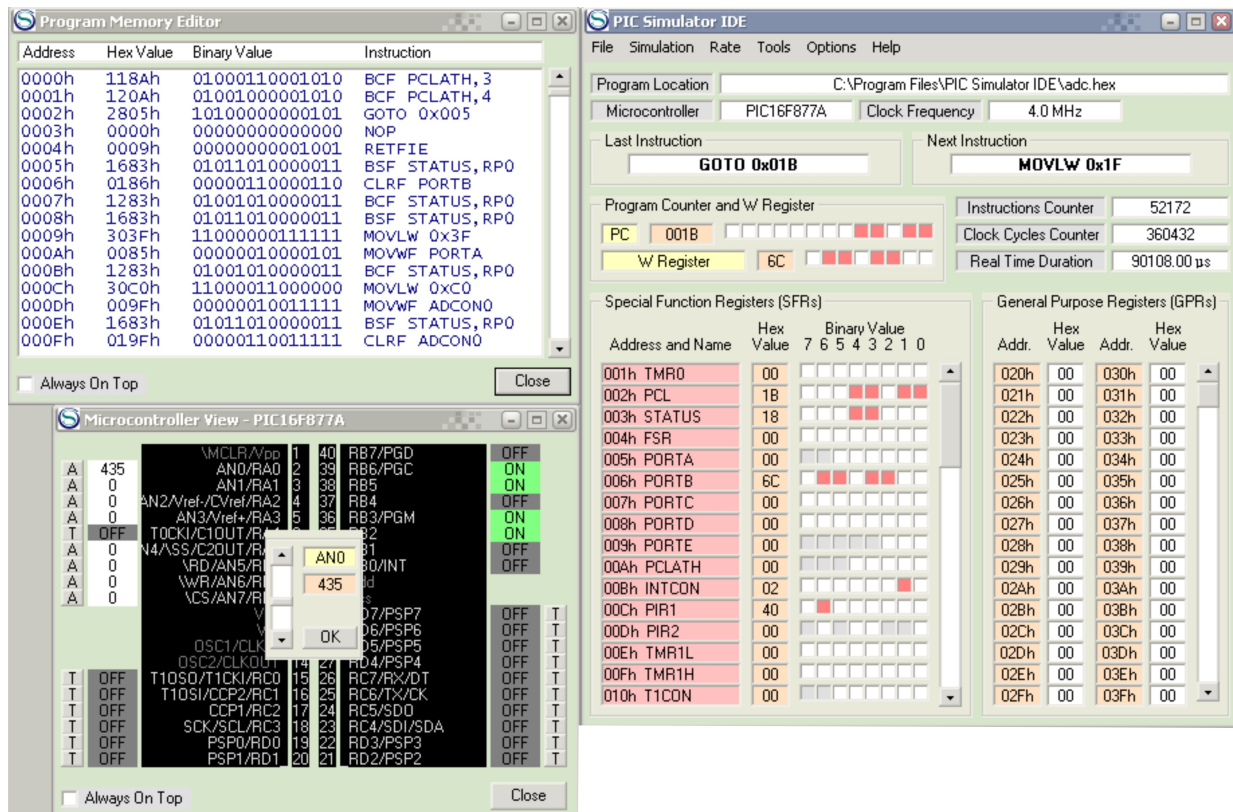


Рис. 6 Вигляд екрану з виконуваною програмою «Робота з аналого-цифровим перетворювачем»

З вікна “Program Memory Editor” вибираємо сім перших команд з їх шістнадцятковими кодами і знаходимо з таблиці кодів асемблера PIC-контролера 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.

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

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”.

Вміст регістрів записуємо в шістнадцятковому коді за Прикладом 3.

Приклад 3.
Регістр
PORTB

Вміст регістра
6C

і т. д.

4. Контрольні запитання

1. Послідовність роботи АЦП мікроконтролера.
2. Аналогові та цифрові виводи мікроконтролера.
3. Формат регістра STATUS.
4. Призначення та позначення основних елементів програмної моделі мікроконтролера

5. Література

1. Данилин А. Програма-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. – (В помощь радиолюбителю).
4. Кениг А. и М. Полное руководство по PIC-микроконтроллерам.: Пер. с нем.-К.: “МК-Пресс”, 2007.-256 с., ил.