Министерство образования Республики Беларусь

Учреждение образования

Белорусский государственный университет информатики и радиоэлектроники

Кафедра РТС

Отчет по лабораторной работе №1

«ПРОГРАММИРОВАНИЕ И ОТЛАДКА ПРОЦЕДУР АРИФМЕТИЧЕСКИХ И ЛОГИЧЕСКИХ ПРЕОБРАЗОВАНИЙ ДАННЫХ»

|  |  |
| --- | --- |
| Выполнили: | Проверил: |
| ст.гр.240102 |  |
|  | Каленкович Е.Н. |

Минск 2015

**Цель работы**

Изучить архитектуру микропроцессорного устройства и приемы работы в интегрированной среде MPLAB на примерах составления и отладки программ процедур арифметических и логических преобразований данных.

**Принципиальные схемы**



Рисунок *1* – Принципиальная схема макетной платы



Рисунок *2* – Структурная схема однокристального микроконтроллера PIC16F886



Рисунок *3* – Однокристальный микроконтроллер PIC16F886



Рисунок *4* – Принципиальная схема модуля бинарных периферийных устройств

**Выполнение работы**

Для следующих комбинаций, подаваемых на вход микроконтроллера, выполнили определенные арифметические и логические операции.

Таблица *1* – Комбинации на входе и соответствующие им операции

|  |  |  |
| --- | --- | --- |
| Комбинации на входе | Операции над переменными | Ожидаемый результат в системе счисления |
| 10-ой | 2-ой | 16-ой |
| *1100* | *Вывод переменной А* | *101* | *01100101* | *0x65* |
| *1110* | *Вывод переменной В* | *49* | *00110001* | *0x31* |
| *1111* | *Вывод переменной С* | *25* | *00011001* | *0x19* |
| *1000* | *(А - С)/2* | *38* | *00100110* | *0x26* |
| *0100* | *!(А +В)* | *105* | *01101001* | *0x69* |
| *0010* | *(ВС)4* | *160* | *10100000* | *0xA0* |
| *0001* | *(!С)/2 + 10* | *125* | *01111101* | *0x7D* |

|  |
| --- |
| Действия при выполнении операции *(А - С)/2* |
|  |

|  |
| --- |
| Действия при выполнении операции *!(А +В)* |
|  |

|  |
| --- |
| Действия при выполнении операции *(ВС)4* |
|  |

|  |
| --- |
| Действия при выполнении операции *(!С)/2 + 10* |
|  |

**Код программы:**

list p=16f886 ; list directive to define processor

 #include <p16f886.inc> ; processor specific variable definitions

; '\_\_CONFIG' directive is used to embed configuration data within .asm file.

; The labels following the directive are located in the respective .inc file.

; See respective data sheet for additional information on configuration word.

 \_\_CONFIG \_CONFIG1, \_LVP\_OFF & \_FCMEN\_ON & \_IESO\_OFF & \_BOR\_OFF & \_CPD\_OFF & \_CP\_OFF & \_MCLRE\_ON & \_PWRTE\_ON & \_WDT\_OFF &\_HS\_OSC

 \_\_CONFIG \_CONFIG2, \_WRT\_OFF & \_BOR21V

;\*\*\*\*\* VARIABLE DEFINITIONS

w\_temp EQU 0x7D ; variable used for context saving

status\_temp EQU 0x7E ; variable used for context saving

pclath\_temp EQU 0x7F ; variable used for context saving

\_A EQU 0x20 ; Регистр для хранения числа A

\_B EQU 0x21 ; Регистр для хранения числа B

\_C EQU 0x22 ; Регистр для хранения числа С

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

 ORG 0x000 ; processor reset vector

 nop

 goto main ; go to beginning of program

 ORG 0x004 ; interrupt vector location

 movwf w\_temp ; save off current W register contents

 movf STATUS,w ; move status register into W register

 movwf status\_temp ; save off contents of STATUS register

 movf PCLATH,w ; move pclath register into w register

 movwf pclath\_temp ; save off contents of PCLATH register

; isr code can go here or be located as a call subroutine elsewhere

 movf pclath\_temp,w ; retrieve copy of PCLATH register

 movwf PCLATH ; restore pre-isr PCLATH register contents

 movf status\_temp,w ; retrieve copy of STATUS register

 movwf STATUS ; restore pre-isr STATUS register contents

 swapf w\_temp,f

 swapf w\_temp,w ; restore pre-isr W register contents

 retfie ; return from interrupt

main

 banksel PORTA ;Директива выбора банка памяти, где расположен указанный

;регистр

 clrf PORTA ;Инициализировать порт А

 clrf PORTB ;Инициализировать порт B

 clrf PORTC ;Инициализировать порт C

 clrf \_A ;Очистить регистр \_А

 clrf \_B ;Очистить регистр \_В

 clrf \_C ;Очистить регистр \_С

 clrf \_pop ;Очистить регистр \_pop

 banksel ANSEL ;Выбрать банк памяти

 clrf ANSEL ;Очистить регистры ANSEL и ANSELH

 clrf ANSELH ;Установить порты ввода/вывода как цифровые

 banksel TRISA ;Выбрать банк памяти

 bcf TRISA,5 ;Установить вывод PORTA RA5 как выход

 clrf TRISB ;Настроить все выводы PORTB как выходы

 movlw b'11110000' ;Загрузить константу в W

 movwf TRISC ;Установить выводы PORTC RC4,RC5,RC6,RC7 как вход,

;остальные как выходы

 banksel PORTC ;Выбрать банк памяти

metka: ;Метка основного бесконечного цикла программы

 movlw .101 ;Присвоить переменной А значение 111

 movwf \_A ;Записать число 111 в регистр \_А

 movlw .49 ;Присвоить переменной B значение 39

 movwf \_B ;Записать число 39 в регистр \_В

 movlw .25 ;Присвоить переменной С значение 25

 movwf \_C ;Записать число 25 в регистр \_С

 subwf PORTC,w ;Вычесть w из PORTC

 btfsc STATUS,Z ;Если результат равен нулю, то пропустить следующюю команду,

 goto C7 ;иначе перейти к подпрограмме вычисления функции (А-С)/2

 subwf PORTC,w ;Вычесть w из PORTC

 btfsc STATUS,Z ;Если результат равен нулю, то пропустить следующюю команду,

 goto C6 ;иначе перейти к подпрограмме вычисления функции !(А+В)

 subwf PORTC,w ;Вычесть w из PORTC

 btfsc STATUS,Z ;Если результат равен нулю, то пропустить следующюю команду,

 goto C5 ;иначе перейти к подпрограмме вычисления функции

;(В"исключающее или"А)\*4

 subwf PORTC,w ;Вычесть w из PORTC

 btfsc STATUS,Z ;Если результат равен нулю, то пропустить следующюю команду,

 goto C4 ;иначе перейти к подпрограмме вычисления функции (!С)/2+10

 subwf PORTC,w ;Вычесть w из PORTC

 btfsc STATUS,Z ;Если результат равен нулю, то пропустить следующюю команду,

 goto A1 ;иначе перейти к подпрограмме вывода переменной А

 subwf PORTC,w ;Вычесть w из PORTC

 btfsc STATUS,Z ;Если результат равен нулю, то пропустить следующюю команду,

 goto B1 ;иначе перейти к подпрограмме вывода переменной В

 subwf PORTC,w ;Вычесть w из PORTC

 btfsc STATUS,Z ;Если результат равен нулю, то пропустить следующюю команду,

 goto C1 ;иначе перейти к подпрограмме вывода переменной С

 goto metka ;Вернуться к программе

C7: ;Метка подпрограммы вывода числа (А-C)/2

 movf \_C,w ;Занести число С в w

 subwf \_A,f ;Вычесть число C из w, результат сохранить в регистре \_С

C6: ;Метка подпрограммы вывода числа !(А+В)

 movf \_A,w ;Занести число A в W

 addwf \_B,f ;Сложение А и В, результат сохранить в регистре \_В

 comf \_B,w ;Инверсия содержимого регистра \_В, результат сохранить в w

 movwf PORTB ;Вывести на PORTB число В

 goto metka ;Вернуться к выполнению программы

C5: ;Метка подпрограммы вывода значения функции

;(В"исключающее или"А)\*4

 movf \_B,w ;Переслать содержимое \_В в w

 xorwf \_C,f ;Выполнить операцию "сумма по модулю 2" w и \_C

 bcf STATUS,C ;Установать бит С в регистре STATUS в 0

 rlf \_C,f ;Умножить содержимое \_C на 2

 bcf STATUS,C ;Установать бит С в регистре STATUS в 0

 rlf \_C,w ;Умножить содержимое \_C на 2

 movwf PORTB ;Переслать содержимое w в PORTB

 goto metka ;Вернуться к выполнению программы

C4: ;Метка подпрограммы вывода значения функции (!С)/2+10

 comf \_C,f ;Инвертировать \_С

 bcf STATUS,C ;Установать бит С в регистре STATUS в 0

 rrf \_C,w ;Умножить содержимое \_C на 2

A1:

 movf \_A,w ;Записать константу в W

 movwf PORTB ;Переслать содержимое w в PORTB

 goto metka ;Вернуться к выполнению программы

B1:

 movf \_B,w ;Записать константу в W

 movwf PORTB ;Переслать содержимое w в PORTB

 goto metka ;Вернуться к выполнению программы

C1:

 movf \_C,w ;Записать константу в W

 movwf PORTB ;Переслать содержимое w в PORTB

 goto metka ;Вернуться к выполнению программы

 END ;Конец текста программы

**Вывод**

В ходе данной лабораторной работы были изучены архитектура микропроцессорного устройства и приемы работы в интегрированной среде MPLAB на примерах составления и отладки программ процедур арифметических и логических преобразований данных.

По поводу остальных лабораторных обращаться https://vk.com/strittrance