Министерство образования и науки Российской Федерации

Севастопольский государственный университет

Институт информационных технологий и управления в технических системах

Кафедра ИС

ОТЧЕТ

по лабораторной работе №2

Исследование системы команд микроконтроллеров AVR

Выполнил:

ст. гр. ИС/б-21-о

Куркчи А. Э.

Проверил:

Кудрявченко И. В.

Севастополь

2016

1. ЦЕЛЬ РАБОТЫ

Изучить основные команды микроконтроллеров семейства AVR и исследовать влияние команд различного типа на флаги регистра состояния и регистры управления портами ввода/вывода, а также на работу функциональных узлов устройства. Приобрести практические навыки разработки программ средней сложности на языке Ассемблера и отладки их в специализированной инструментальной среде.

1. ПОСТАНОВКА ЗАДАЧИ
   1. Ознакомиться с системой команд микроконтроллеров AVR и изучить наиболее часто используемые команды.
   2. Повторить материал по архитектуре микроконтроллера и схеме организации памяти и портов ввода/вывода микроконтроллера.
   3. Составить алгоритм программы согласно заданному варианту.
   4. Составить текст программы на языке ассемблера и откомпилировать программу.
   5. Исправить ошибки, при их обнаружении компилятором.
   6. Провести исследование процесса выполнения программы в среде AVR Studio.

Вариант №3:

Составить программу сложения 2 однобайтных без знаковых чисел, находящихся в ОЗУ. Затем полученную сумму умножить на 7. Если в результате сложения происходит переполнение разрядной сетки, вывести в порт В число $FF. Если переполнения нет, то выводить непрерывно в порт D поочередно числа $55 и $AA.

1. ОПИСАНИЕ УСТАНОВКИ

Используемый МК *ATmega16* имеет следующие характеристики:

* Тактовая частота – 16 МГц;
* 32 восьмиразрядных рабочих регистра общего назначения;
* 130 команд;
* 16 Кбайт внутрисистемная программируемая *Flash*-память;
* Встроенный сторожевой таймер, тактовый генератор;
* Питание 2.7 – 5.5 В.

1. КОД ПРОГРАММЫ

.include "m16def.inc"

.def a**=**r16

.def b**=**r17

.def **c=**r18

.def d**=**r19

.def count**=**r20

init**:**

ldi a**,** $FF ; Запись первого числа

ldi b**,** $FF ; Запись второго числа

ldi r21**,** $FF ; Константа для вывода в случае переполнения

ldi r22**,** $55 ; Константа для мигания в случае удачи

ldi r23**,** $AA ; Константа для мигания в случае удачи

ldi r24**,** $00 ; Константа для обнуления

**jmp** main

mult**:**

ldi count**,** 7 ; Множитель/количество итераций сложения

mult\_for**:**

**adc** d**,** **c**

brcs fail ; Если произошел перенос в процессе умножения

**dec** count

brne mult\_for ; Цикл for на count итераций

**jmp** success ; Когда цикл закончился перейти к помигиванию

fail**:** ; Вывод $FF если переполнение

ldi count**,** 32 ; Цикл на 32 помигивания

**out** PORTB**,** r21 ; Вывод на порт B $FF

fail\_for**:**

**nop**

**dec** count

brne fail\_for

**out** PORTB**,** r24 ; Обнулить вывод в порт B

**ret**

success**:** ; Помигивание если переполнения не было

ldi count**,** 32 ; Цикл на 32 помигивания

success\_for**:**

**out** PORTD**,** r22 ; Вывод на порт D $55

**nop**

**out** PORTD**,** r23 ; Вывод на порт D $AA

**nop**

**dec** count

brne success\_for ; Цикл for на count итераций

**out** PORTD**,** r24 ; Обнулить вывод в порт D

**ret**

main**:**

**mov** **c,** a

**adc** **c,** b

brcs fail ; Если произошел перенос при первичном сложении

ldi d**,** 0

**call** mult

1. ОПИСАНИЕ РАБОТЫ ПРОГРАММЫ

В соответствии с заданием программа складывает 2 числа из ОЗУ, затем полученную сумму умножает на 7. Если в процессе сложения или умножения происходит переполнение разрядкой сетки в порт B выводится число $FF (рисунок 1), иначе в порт D попеременно выводятся числа $55 и $AA (рисунок 2). При переполнении разрядкой сетки устанавливается флаг переноса C и происходит переход по оператору brcs fail.

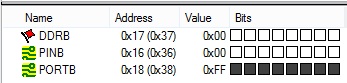


Рисунок 1 – Вывод $FF в порт B

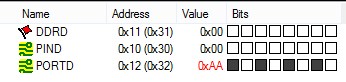
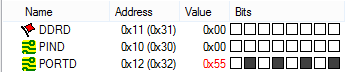


Рисунок 2 – Попеременный вывод $55 и $AA в порт D

ВЫВОДЫ

В ходе выполнения данной лабораторной работы была изучены основные команды микроконтроллеров семейства AVR и исследовано влияние команд различного типа на флаги регистра состояния и регистры управления портами ввода/вывода, а также на работу функциональных узлов устройства. Приобретены практические навыки разработки программ средней сложности на языке Ассемблера и отладки их в специализированной инструментальной среде.