

Департамент образования Вологодской области
Бюджетное профессиональное образовательное учреждение
Вологодской области
«Череповецкий химико-технологический колледж»



ПОРТЫ ВВОДА/ВЫВОДА

МИКРОКОНТРОЛЛЕРА АТМЕГА 8535

***Методические указания
к выполнению лабораторной работы***

*для студентов 3 курса по
специальности 15.02.07
Автоматизация
технологических процессов и
производств (по отраслям)*

Череповец, 2020

Методические указания предназначены для выполнения лабораторной работы по междисциплинарному курсу МДК.03.01 Теоретические основы технического обслуживания и эксплуатации автоматических и мехатронных систем управления для студентов БПОУ ВО «Череповецкий химико-технологический колледж» специальности 15.02.07 Автоматизация технологических процессов и производств (по отраслям)

Разработчик:

Кузнецова Наталья Гаевна – преподаватель БПОУ ВО «Череповецкий химико-технологический колледж»

Рассмотрены и рекомендованы
к использованию на заседании

ЦМК по ППССЗ

Протокол № 7

от «26» февраля 2020 г.

Председатель ЦМК

_____/Н.Г. Кузнецова/

Содержание

	стр.
Пояснительная записка	5
1 Правила работы на лабораторном стенде «Микроконтроллер»	6
2 Порядок работы в лаборатории	8
3 Описание модуля «Микроконтроллер»	10
4 Лабораторная работа «Порты ввода/вывода микроконтроллера Atmega8535»	14
5 Критерии оценивания лабораторной работы	32
Приложение А	34
Список литературы	42

Пояснительная записка

Методические указания предназначены для выполнения лабораторной работы по междисциплинарному курсу МДК.03.01 Теоретические основы технического обслуживания и эксплуатации автоматических и мехатронных систем управления.

Методические указания разработаны в соответствии с программой профессионального модуля ПМ.3 Эксплуатация систем автоматизации и направлены на формирование профессиональных компетенций (ПК):

Код	Наименование результата обучения
ПК 3.1.	Выполнять работы по эксплуатации систем автоматического управления с учетом специфики технологического процесса
ПК 3.2.	Контролировать и анализировать функционирование параметров систем в процессе эксплуатации
ПК 3.3.	Снимать и анализировать показания приборов

В методических указаниях приведены правила работы на лабораторном стенде, описание модуля стенда. Указаны этапы создания проекта в среде AVR Studio. Представлен пример набора и отладки программы на симуляторе AVR Studio, что позволит освоить студентам методику написания программ на языке ассемблера микроконтроллера ATmega8535.

1 Правила работы на лабораторном стенде «Микроконтроллер»

При работе в лаборатории «Автоматизация технологических процессов» во избежание несчастных случаев, а также преждевременного выхода из строя приборов и электрооборудования обучающийся, при выполнении лабораторных работ должен строго соблюдать следующие правила:

- 1) Приступая к работе, необходимо убедиться в том, что к стенду не подано напряжение.
- 2) На металлических корпусах приборов может, при нарушении изоляции, возникнуть опасное для жизни человека напряжение. В связи с этим **запрещается** самостоятельно **вскрывать** переднюю панель стенде и производить электромонтажные работы.
- 3) Подавать питание на стенд следует только после проверки правильности собранной схемы преподавателем.
- 4) **Запрещается** подключать к стенду оборудование, непредусмотренное техническим описанием и методическими указаниями.
- 5) **Запрещается** производить какие-либо переключения на стенде при включенном питании (кроме предусмотренных в указаниях к выполнению лабораторных работ).
- 6) Сборку электрической цепи производят соединительными проводами при выключенном напряжении питания стенда в строгом соответствии с указаниями, представленными в лабораторной

работе, обеспечивая при этом надежность электрических контактов всех разъемных соединений.

- 7) При сборке электрической цепи необходимо следить затем, чтобы соединительные провода не перегибались и не скручивались петлями.
- 8) Собранная электрическая цепь предъявляется для проверки преподавателю.
- 9) Включение стенда под напряжение (после проверки соединительных линий) производится только с разрешения и в присутствии преподавателя.
- 10) Переключения и исправления в собранной электрической цепи разрешается производить только при отключенном напряжении питания.
- 11) При обнаружении повреждений электрического оборудования и приборов стенда, а также при появлении дыма, специфического запаха или искрения необходимо немедленно выключить напряжение питания стенда и известить об этом преподавателя.
- 12) После выполнения лабораторной работы необходимо выключить напряжение питания стенда, разобрать исследуемую электрическую цепь и привести в порядок рабочее место.
- 13) В случае поражения человека электрическим током необходимо немедленно обесточить стенд, выключив напряжение питания. При потере сознания и остановке дыхания необходимо немедленно освободить пострадавшего от

стесняющей его одежды и делать искусственное дыхание до прибытия врача.

2 Порядок работы в лаборатории

Вход в лабораторию допускается только по разрешению преподавателя.

При работе в лаборатории категорически запрещается приносить с собой вещи и предметы, загромождающие рабочие места, способствующие созданию условий, могущих привести к нарушению правил техники безопасности. В лаборатории запрещается громко разговаривать, покидать рабочие места и переходить от одного стенда к другому, без разрешения преподавателя. Приступая в лаборатории к работе, обучающийся должен ознакомиться с правилами работы и порядком работы в лаборатории, расписаться в ведомости.

Приступая к работе в лаборатории, группа делится на подгруппы (3-4 человека), которые затем последовательно выполняют лабораторные работы. Лабораторная работа, пропущенная студентом, выполняется по согласованию с преподавателем.

Лабораторная работа выполняется в соответствии с методическими рекомендациями:

- Прочитать название работы и выяснить смысл всех непонятных слов.

- Прочитать описание работы от начала до конца, не задерживаясь на чтении программ. Задача первого прочтения состоит в том, чтобы выяснить, какова цель лабораторной работы.
- Прочитать теоретический материал, относящийся к данной работе. Разобрать примеры программ.
- Изучить устройство и описание модуля «Микроконтроллер».

Лабораторная работа оформляется в тетради в соответствии с ходом выполнения работы.

В требования к оформлению работы входит: название лабораторной работы и её цель; используемое оборудование; ход выполнения лабораторной работы; исходные данные (схемы, значения параметров, типы устройств и элементов), краткие выводы по каждой работе, анализ полученных данных; обобщающий вывод по всей лабораторной работе — краткое заключение о результатах работы, согласующееся с её целью.

Лабораторная работа, оформленная в тетради, сдается для проверки и оценивания преподавателю.

3 Описание модуля «Микроконтроллер»

Модуль «Микроконтроллер» предназначен для программирования и изучения функций микроконтроллера ATmega8535 семейства AVR, выпускаемого фирмой Atmel. Внешний вид модуля приведен на рис. 1.

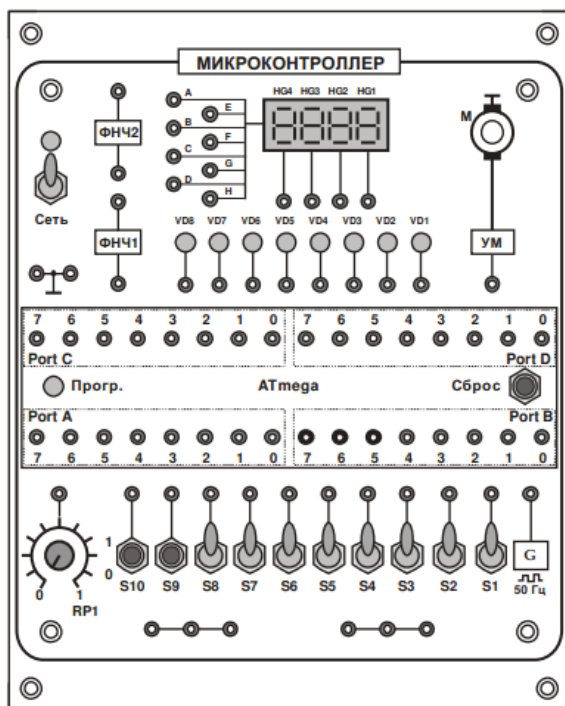


Рис. 1. Внешний вид модуля «Микроконтроллер»

На лицевой панели модуля расположены: – переключатель «Сеть» со светодиодом индикации наличия напряжения.

Переключатель осуществляет коммутацию напряжения, подаваемого на модуль;

- мнемосхему микроконтроллера с клеммами, связанными с портами ввода/вывода микроконтроллера;
- переключатели S1-S8 с выходными клеммами для подачи логических сигналов на микроконтроллер;
- кнопки S9, S10 с выходными клеммами для подачи логических сигналов на микроконтроллер;
- потенциометр RP1 с выходной клеммой для подачи аналогового напряжения на микроконтроллер;
- мнемосхема генератора низкочастотного прямоугольного сигнала 50 Гц и клемма выхода генератора;
- светодиоды VD1 – VD8 с клеммами для их подключения к источнику напряжения (например, к микроконтроллеру);
- электродвигатель постоянного тока М с усилителем мощности и клеммой для подачи на него управляющего напряжения;
- семисегментный четырехсимвольный светодиодный индикатор с клеммами подачи напряжения на сегменты А, В, С, D, Е, F, G, H, а также на общую точку каждого сегмента индикатора;
- два фильтра низкой частоты для фильтрации ШИМ-сигналов на выходе микроконтроллера.

С тыльной стороны модуля располагается разъем для подачи напряжения ~220В 50 Гц на модуль, а также разъем для подключения модуля к персональному компьютеру по интерфейсу USB.

Табл. 1. Краткая характеристика микроконтроллера ATmega8535

Параметр	Значение
Частота установленного кварцевого резонатора	8 МГц
Напряжение электропитания	2,7 – 5,5 В
Объем внутренней Flash – памяти	8 кБайт
Объем энергонезависимой памяти	512 Байт
Объем внутренней ОЗУ	512 Байт
32 программируемых входа/выхода	32 на 4 портах
JTAG – интерфейс	нет
8-битные таймеры/счетчики с ШИМ	2 шт.
16-битный таймер/счетчик с ШИМ	1 шт.
10-разрядный аналогово-цифровой преобразователь	есть
Количество каналов АЦП	8
Аналоговый компаратор	есть
Источники внешних прерываний	3 шт.
Универсальный приемопередатчик USART	есть
SPI – интерфейс	есть
I ² C – интерфейс	есть

В состав лабораторного стенда входит 8-разрядный микроконтроллер AVR ATmega8535 и необходимые элементы для исследования его периферийных устройств и большинства функциональных возможностей. Поскольку микроконтроллер является программируемым, пользователь должен освоить его программирование в различных программных средах и на различных языках высокого и низкого уровня. Среди наиболее популярных методов написания программ отметим:

– написание программ на машинном коде микроконтроллера. Программы, написанные таким способом, являются наиболее быстродействующими,

однако для их написания требуется высокая квалификация программиста и глубокое знание архитектуры процессора;

– написание программы на ассемблере. Написание программ на ассемблере существенно проще, чем на машинном коде, однако также требует высокой квалификации программиста. Следует отметить, что язык ассемблера обычно жестко привязан к конкретному типу микропроцессора и может существенно отличаться для разных микропроцессоров одного производителя;

– написание программы на языке высокого уровня (например, Си). Такие программы обычно являются кросс-платформенными, то есть практически не отличаются по синтаксису для микропроцессоров разных типов. Пользователь пишет программу на языке высокого уровня, а компилятор преобразует ее на ассемблер и машинный код конкретного микропроцессора. Однако обычно использование языка высокого уровня при написании программ для микроконтроллеров может существенно ограничить их быстродействие.

4 Лабораторная работа

«Порты ввода/вывода микроконтроллера Atmega8535»

Цель работы: Освоить работу с портами ввода/вывода микроконтроллеров AVR.

План работы

1. Изучить необходимый теоретический материал, быть готовым ответить на вопросы преподавателя.
2. Изучить представленные в лабораторной работе примеры программ по работе с портами ввода/вывода.
3. Написать и отладить собственную программу в соответствии с вариантом.

Ход выполнения лабораторной работы

Порты ввода/вывода микроконтроллера предназначены для передачи и приема информации и последующей ее обработки. Микроконтроллеры различных типов содержат различное количество портов ввода/вывода. Изучаемый в данном курсе микроконтроллер Atmega8535 содержит 4 порта ввода/вывода, выполненных 8-разрядными: PORTA, PORTB, PORTC, PORTD.

Порты ввода/вывода непосредственно связаны с выводами микросхемы микроконтроллера, при этом каждый конкретный вывод микроконтроллера жестко «привязан» к конкретному разряду порта ввода/вывода. На рис. 1 представлено расположение выводов микроконтроллера Atmega8535, выполненного в DIP-

корпусе.

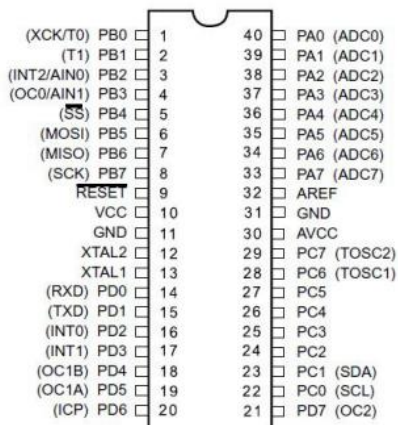


Рис. 1. Расположение выводов микросхемы контроллера ATmega8535 (DIP-40)

По умолчанию выводы микросхемы контроллера предназначены для выполнения функций ввода/вывода информации в соответствии с настройками регистров портов ввода/вывода. Однако функции большинства выводов микросхемы могут быть программно изменены. При этом к выводам микросхемы могут быть присоединены выходы таймеров, приемопередатчиков, входы аналогово-цифрового преобразователя, контроллера внешних прерываний. Альтернативные функции выводов микроконтроллера представлены на рис. 1 (в скобках).

Каждый порт состоит из трех регистров, с помощью которых осуществляется установка направления работы порта и выдача/сбор информации (табл. 1).

Табл. 1. Регистры портов ввода/вывода

Наименование порта	Регистры		
	Регистр данных	Регистр направления	Регистр состояния
Порт А	PORTA	DDRA	PINA
Порт В	PORTB	DDRB	PINB
Порт С	PORTC	DDRC	PINC
Порт D	PORTD	DDRC	PIND

Регистры направления определяют режим работы портов ввода/вывода. Если в каком-либо разряде регистра установлена логическая «1», то соответствующий вывод микросхемы контроллера (рис. 1) работает на вывод информации из микроконтроллера. В противном случае соответствующий вывод микросхемы (рис. 1) работает на ввод информации в микроконтроллер.

Регистры данных предназначены для передачи данных на выводы микросхемы контроллера. Если в каком-либо разряде регистра установлена логическая «1», а соответствующий вывод микросхемы сконфигурирован как выход с помощью регистра направления, то на вывод микросхемы контроллера подается сигнал, соответствующий логической «1». В противном случае на вывод микросхемы контроллера подается сигнал логического «0».

Регистры состояния предназначены для отображения текущего состояния сигналов на выводах микросхемы контроллера. Так, если на выводе микросхемы находится сигнал логической «1», то соответствующий разряд регистра направления находится в состоянии логической «1».

Инициализация порта на ввод и на вывод информации:

Для того, чтобы освоить принципы установки порта на ввод или на вывод информации, необходимо иметь представление о реализации его структуры. Каждый вывод порта выполнен по схеме, представленной на рис. 2. При инициализации порта на ввод информации микроконтроллер принимает сигналы, поступающие от внешнего объекта. Эти сигналы подаются на выводы микросхемы (рис. 2, вывод PIN).

Диоды VD1 и VD2 выполняют функцию защиты микропроцессора от сигналов, величина которых находится за пределами диапазона 0...5В. Так, если напряжение на входе микроконтроллера превышает +5В, то открывается диод VD1, а если напряжение оказывается меньше 0В, то открывается диод VD2 (рис. 2).

Конденсатор C1 выполняет защиту микроконтроллера от импульсных помех.

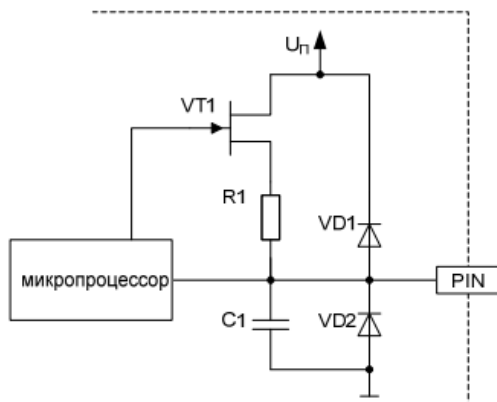


Рис. 2. Структура реализации вывода портов микроконтроллера

Поскольку микропроцессор имеет высокое входное сопротивление, его вход является восприимчивым к воздействию помех. По этой причине важно, чтобы сигнал, подаваемый на микропроцессор, имел однозначное значение логического «0» или логической «1». Для этого в микроконтроллере присутствуют так называемые подтягивающие резисторы (Pull Up).

Подтягивающий резистор имеет высокое сопротивление, измеряемое десятками кОм, и не оказывает влияния на подключаемые к контроллеру сигналы.

Через резистор R1 к порту ввода/вывода подключается напряжение питания с помощью транзистора VT1. Если при этом к выводу микросхемы контроллера не подключена внешняя цепь и вывод «висит в воздухе», то на микропроцессор через подтягивающий резистор R1 подается сигнал логической «1». Это надежно защищает вывод микроконтроллера от воздействия внешних помех.

Для инициализации порта на ввод информации и подключения подтягивающих резисторов необходимо:

- задать в регистре направления DDR работу порта на ввод информации установкой нулевых значений в его разрядах;

- включить подтягивающие резисторы порта ввода/вывода установкой сигналов логической «1» в разрядах регистра PORTX.

Считывание данных с порта в данном режиме осуществляется путем опроса регистра состояния PIN.

Пример инициализации порта A на ввод информации:

`ldi r16,0x00 ;в POH r16 записывается число 0000 0000`

out DDRA,r16 ;командой out значение r16 посылается в DDRA

ldi r16,0xFF ;в POH записывается число 1111 1111

out PORTA,r16 ;командой out значение r16 посылается в PORTA

in r16, PINA ;командой in значение PINA записывается в POH r16

При инициализации порта на вывод информации код, выдаваемый микропроцессором, поступает непосредственно на вывод микросхемы контроллера.

Подтягивающие резисторы при этом должны быть отключены.

Для инициализации порта на вывод информации необходимо:

- задать в регистре направления DDR работу порта на вывод информации установкой его разрядах сигналов логической «1»;
- вывести необходимую информацию на порт записью в регистр данных PORT необходимых значений.

Пример инициализации порта A на вывод информации:

ldi r16,0xFF ;в POH r16 записывается число 1111 1111

out DDRA,r16 ;командой out значение r16 посылается в DDRA

ldi r16,0x35 ;в r16 записывается любое число, например 0011 0101

out PORTA,r16 ;командой out значение r16 посылается в PORTA

Пример №1. Написать программу, осуществляющую сложение двух младших и двух старших бит порта С с выводом результата на порт D.

```

;-----
;Программа для сложения двух чисел PC7:PC6 и PC1:PC0
с последующим
выводом результата на PORTD
;Входы: PINC7:PINC6 и PINC1:PINC0
;Выходы: PORTD
.include "m8535def.inc" ;Подключение стандартной
библиотеки ATmega8535
.cseg ;Начало сегмента кода
.org 0
reset: ;Инициализация портов ввода/вывода
    ldi r16,0xFF
    out DDRD,r16 ;Установка PORTD на вывод информации
    clr r16
    out DDRC,r16 ;Установка PORTC на ввод информации
main:
    in r16,PINC ;Опрос PINC с занесением результата в r16
    mov r17,r16 ;Копирование результата в r17
    andi r16,0x03 ;Обнуление всех бит, кроме 0 и 1
    andi r17,0xC0 ;Обнуление всех бит, кроме 6 и 7
    clr r18 ;Очистка r18
shift:
    lsr r17 ;Сдвиг r17 влево
    inc r18 ;Инкремент r18
    cpi r18,0x06 ;Если r18≠6,
    brne shift ;то возврат на shift
    add r16,r17 ;Иначе - сложение r16 и r17
    out PORTD,r16 ;Вывод результата на PORTD
    rjmp main ;возврат на main
;-----

```

Рассмотрим программу более подробно.

1. В начале программы подключается стандартная библиотека контроллера Atmega8535, которая содержит всю необходимую информацию о контроллере, а именно список регистров с их адресами, объемы памяти, список периферийных устройств и другие особенности:

```
.include "m8535def.inc"
```

2. Далее устанавливается начало сегмента кода на нулевой адрес:

```
.cseg
```

```
.org 0
```

3. После этого необходимо произвести инициализацию портов ввода/вывода. Порт C инициализируется на ввод, а порт D – на вывод информации:

reset:

```
ldi r16,0xFF
```

```
out DDRD,r16
```

```
clr r16
```

```
out DDRC,r16
```

Поскольку порт D необходимо инициализировать на вывод, в регистр DDRD требуется записать 0xFF (1111 1111 в двоичном коде). Для этого сначала в регистр общего назначения r16 записывается число 0xFF (ldi r16,0xFF), а затем содержимое этого регистра переписывается в регистр DDRD (out DDRD,r16).

Аналогичная операция производится с портом C.

4. В главной программе необходимо опросить состояние регистра PINC и выделить два младших и два старших бита (согласно заданию):

main:

```
in r16,PINC
mov r17,r16
andi r16,0x03
andi r17,0xC0
```

Для этого содержимое регистра PINC переносится в регистр общего назначения (РОН) r16 (in r16,PINC) и копируется в регистр r17 (mov r17,r16). Далее осуществляется побитовое умножение содержимого регистров r16 и r17 с константами 0x03 (0000 0011) и 0xC0 (1100 0000) соответственно. Таким образом, в регистре r16 остаются только два младших бита PINC, а в регистре r17 – два старших бита PINC.

5. Для проведения операции сложения r16 и r17 необходимо преобразовать число XX00 0000, содержащееся в r17, в число вида 0000 00XX. Для этого в программе использован циклический сдвиг содержимого r17 на 6 бит:

```
clr r18
shift:
lsr r17
inc r18
cpi r18,0x06
brne shift
```

РОН r18 очищается (clr r18), происходит сдвиг r17 на один бит вправо (lsr r17), регистр r18 инкрементируется (inc r18), затем его содержимое сравнивается с числом 6 (cpi r18,0x06). Если r18 меньше 6, происходит возврат на метку «shift», иначе – выполнение программы далее (brne shift).

6. После выполнения операции сдвига r17 можно произвести сложение r16 и r17 и вывод результата сложения на PORTD:

```
add r16,r17
```

```
out PORTD,r16
```

7. В конце программы необходимо вернуться на метку «main» для создания циклического опроса портов:

```
rjmp main
```

Пример 2. Реализовать на микроконтроллере расчет логического уравнения

$D = A \cdot (\bar{A} \cdot B + C)$, где A, B, C логические сигналы, поступающие на входы PINA0, PINA1, PINA2, а D – результат решения логического уравнения, который выводится на PORTD0.

```
;-----
;Программа для решения логического уравнения
D=A*(NA*B+C)
;Входы: A=PINA0
; B=PINA1
; C=PINA2
;Выход: D=PORTD0

.include "m8535def.inc" ;Подключение стандартной
библиотеки Atmega8535
.def A=r20 ;Объявление переменных и присвоение их
имен
.def B=r21 ;регистрам общего назначения r20 ... r23
.def C=r22
.def D=r23
.cseg ;начало сегмента кода
```

```

.org 0
reset:      ;инициализация портов ввода/вывода
    ldi r16,0x01
    out DDRD,r16    ;PORTD0 – на вывод информации
    clr r16
    out DDRA,r16    ;PINA0 ...PINA2 – на ввод информации
main:
    in r16,PINA     ;Ввод значения PINA в POH r16
    mov A,r16       ;Копирование r16 в регистры A, B, C
    mov B,r16
    mov C,r16
    andi A,0x01     ;Выделение бита числа A
    andi B,0x02     ;Выделение бита числа B
    andi C,0x04     ;Выделение бита числа C
    lsr B           ;Сдвиг B вправо на 1 бит
    lsr C
    lsr C           ;Сдвиг C вправо на 2 бита
    mov r16,A       ;Копирование A в POH r16
    com r16         ;Инверсия содержимого r16
    andi r16,0x01   ;Формирование числа NA
    and r16,B       ;Расчет r16=NA*B
    or r16,C        ;Расчет r16=r16+C
    and r16,A       ;Расчет r16=r16*A
    out PORTD,r16   ;Вывод результата на PORTD0
    rjmp main      ;Возврат на main
;-----

```

Рассмотрим программу более подробно.

1. Инициализация контроллера и портов ввода/вывода в данной программе производится аналогично примеру 1:

```

.include "m8535def.inc"
.def A=r20
.def B=r21
.def C=r22
.def D=r23

```



```
.cseg
.org 0
reset:
    ldi r16,0x01
    out DDRD,r16
    clr r16
    out DDRA,r16
```

Здесь для удобства пользователя регистрам r20 ... r23 присваиваются имена переменных A, B, C, D (.def A=r20 и т.д.).

2. В основной программе сначала в регистр r16 переписывается комбинация сигналов, находящаяся на выводах PINA2...PINA0. Затем производится выделение бита A, B и C:

```
in r16,PINA
mov A,r16
mov B,r16
mov C,r16
andi A,0x01
andi B,0x02
andi C,0x04
```

3. В младших битах регистров r20...r22 формируются биты A, B, C. Для этого регистр r21 сдвигается на 1 бит, а r22 – на 2 бита вправо. В случае единичных сдвигов это целесообразно выполнить командой lsl:

```
lsl B
lsl C
lsl C
```

4. Для формирования числа \bar{A} значение регистра A копируется в РОН r16, в котором оно инвертируется:

```
mov r16,A
com r16
```

```
andi r16,0x01
```

5. Далее производятся логические операции над числами A, B, C, NA согласно логическому уравнению $D = A \cdot (\bar{A} \cdot B + C)$, вывод результата на индикацию и заикливание программы:

```
and r16,B
```

```
or r16,C
```

```
and r16,A
```

```
out PORTD,r16
```

```
rjmp main.
```

После набора программы необходимо произвести ее ассемблирование, то есть сборку, при которой ассемблер производит проверку синтаксиса написанной программы и при отсутствии ошибок преобразует код ассемблера в машинный код микропроцессора. При этом формируется файл с расширением *.hex, который далее записывается в микроконтроллер.

Для ассемблирования написанной программы необходимо нажать кнопку F7 «Assemble». При отсутствии ошибок в окне текущих сообщений «built» AVR Studio отображается сообщение об успешном завершении операции, а при наличии ошибок выводится их список и положение в тексте программы.

Отладка программы на симуляторе:

В программное обеспечение AVR Studio встроен симулятор микроконтроллеров, с помощью которого можно отладить программу, найти в ней ошибки и исправить неточности в алгоритме, не осуществляя непосредственно программирования микросхемы

контроллера. Также симулятор может быть полезен при написании программ в домашних условиях.

Симулятор AVR Studio осуществляет симуляцию микроконтроллера, его портов, таймеров, АЦП, прерываний и т.д. Поскольку симулятор работает с hex-файлом проекта, то для запуска эмулятора необходимо написать программу и исключить из нее явные ошибки, с которыми создание hex-файла невозможно.

Для отладки программы в симуляторе необходимо после ее написания нажать сочетание клавиш Ctrl+F7 «Assemble and Run», после чего будет произведено ассемблирование программы и запуск эмулятора.

После успешной компиляции начало программы будет отмечено желтой стрелкой. Для управления процессами отладки программы в строке меню AVR Studio располагается меню «debug», а также панель функциональных кнопок управления симулятором, назначение которых приведено в табл. 1.

Таблица 1.1. Назначение кнопок управления эмулятором

Название	Сочетание клавиш	Назначение
Start Debugging (начать отладку)	Ctrl+Shift+Alt+F5	Начать процесс отладки программы в эмуляторе.
Stop Debugging (остановить отладку)	Ctrl+Shift+F5	Остановить процесс отладки программы в эмуляторе.
Название	Сочетание клавиш	Назначение
Run (запуск эмуляции)	F5	Эмулятор запускается и циклически производит эмуляцию программы без отображения текущих изменений регистров контроллера на экране.
Break (приостановить эмуляцию)	Ctrl+F5	Команда временно приостанавливает эмулятор без потери данных.
Reset (остановить эмуляцию)	Shift+F5	Команда полностью останавливает эмулятор с потерей данных эмуляции.
Step into (Сделать один шаг вперед)	F11	Команда извлекает только одну инструкцию. После ее завершения все рабочие экраны эмулятора обновляются.
Auto Step (Автовыполнение)	Alt+F5	Команда выполняет эмуляцию программы в пошаговом автоматическом режиме с оперативным обновлением информации на экранах эмулятора после каждого шага.
Toggle breakpoint (Точка остановки)	F9	Команда добавляет пользовательскую точку остановки, при достижении которой программа будет приостановлена.
Remove all program breakpoints (удалить все точки остановки)	—	Команда удаляет все активные точки остановки программы.

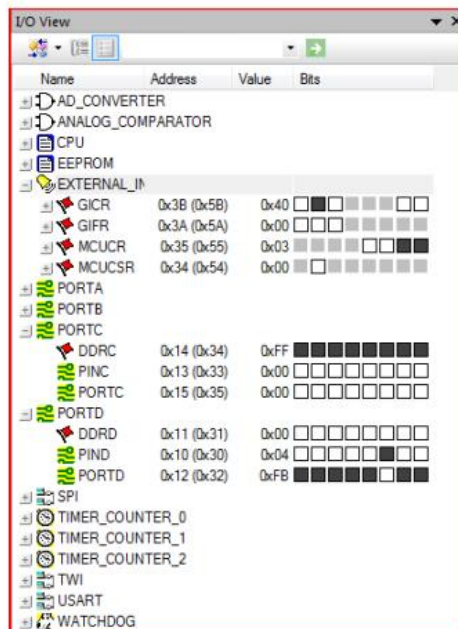


Рис. 5. Контроль состояния регистров микроконтроллера при симуляции

При симуляции программы рекомендуется выполнять пошаговое выполнение инструкций, пользуясь командой Step into (табл. 1). При этом необходимо контролировать содержимое регистров микроконтроллера, а также портов ввода/вывода. Контроль состояния регистров и отдельных устройств микроконтроллера осуществляется в окне «4» (рис. 4). Каждое устройство можно развернуть и увидеть содержимое его регистров управления и контроля (рис. 5).

Запись программы в микроконтроллер:

После успешной отладки программы необходимо проверить ее работу на микроконтроллере. Для этого необходимо:

- включить переключатель «Сеть» модуля «Микроконтроллер» для подачи на него напряжения питания;
- в меню «Tools» AVR Studio выбрать пункт «Program AVR», в котором указать способ соединения «Auto Connect». При правильном подключении к персональному компьютеру модуль микроконтроллера инициализируется как USB COM-порт. Необходимо, чтобы номер этого порта был не более COM9. В противном случае необходимо найти этот порт в диспетчере оборудования Windows и переназначить номер порта.
- если номер USB COM-порта соответствует указанным требованиям, то происходит подключение микроконтроллера к среде AVR-Studio и появляется окно программирования контроллера (рис. 6);
- в окне программирования необходимо выбрать вкладку main, в которой необходимо выбрать тип используемого контроллера, после чего перейти на вкладку «Program», в которой выбрать графу «Flash». В этой графе требуется указать путь к .hex файлу проекта, после чего произвести запись программы в микроконтроллер нажатием кнопки «Program». По завершении записи программы необходимо проверить ее корректную работу на микроконтроллере и показать результат преподавателю.

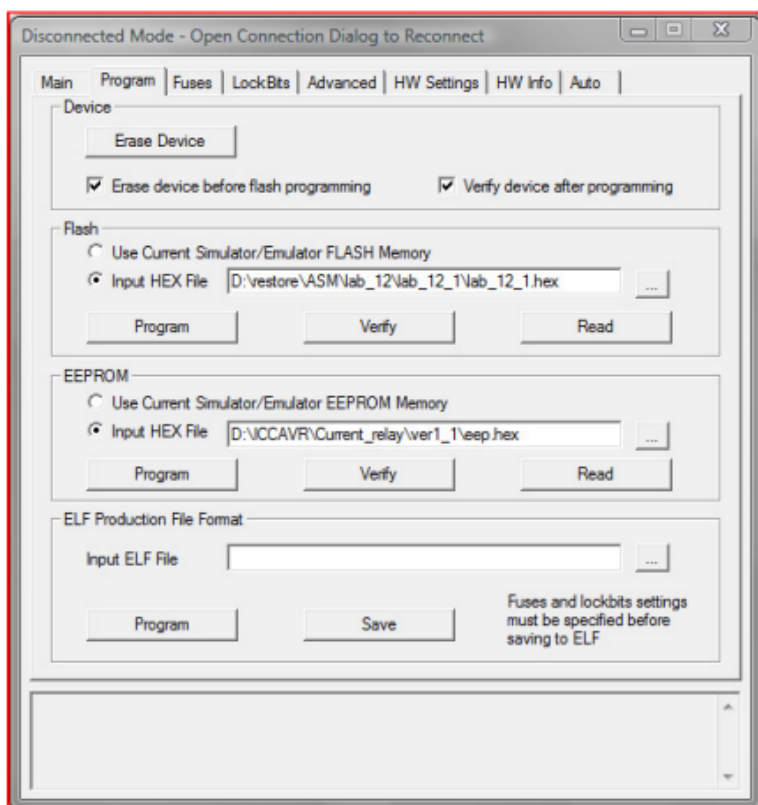


Рис. 6. Окно программирования микроконтроллера

5 Критерии оценивания лабораторной работы

Оценка «5» ставится в том случае, если обучающийся:

- а) выполнил работу в полном объеме с соблюдением необходимой последовательности выполнения лабораторной работы в соответствии с методическими указаниями к ней;
- б) верно написал программу для микроконтроллера и представил ее работу на стенде преподавателю;
- в) в представленном отчете правильно и аккуратно выполнил все записи и сделал выводы;
- г) соблюдал правила работы на лабораторном стенде.

Оценка «4» ставится в том случае, если выполнены требования к оценке «5», но:

- а) в написании программы было допущено два-три недочета, или не более одной негрубой ошибки и одного недочета.

Оценка «3» ставится, если работа выполнена не полностью, но объем выполненной части таков, что позволяет получить правильные результаты и выводы, или если в ходе выполнения работы были допущены следующие ошибки:

- а) или в отчете были допущены в общей сложности не более двух ошибок
- б) или работа выполнена не полностью, однако объем выполненной части таков, что позволяет получить

правильные результаты и выводы по основным, принципиально важным задачам работы.

Оценка «2» ставится в том случае, если:

- а) работа выполнена не полностью, и объем выполненной части работы не позволяет сделать правильных выводов,
- б) или в ходе работы и в отчете обнаружилось в совокупности все недостатки, отмеченные в требованиях к оценке «3».

Во всех случаях оценка снижается, если ученик не соблюдал правила работы на лабораторном стенде!

ПРИЛОЖЕНИЕ А

Система команд микроконтроллеров семейства AVR

Арифметические и логические команды

Мнемоника	Операнды	Описание	Операция	Флаги	Кол-во циклов
ADD	Rd, Rr $0 \leq d \leq 31$ $0 \leq r \leq 31$	Сложить без переноса	$Rd \leftarrow Rd + Rr$	Z, C, N, V, H	1
ADC	Rd, Rr $0 \leq d \leq 31$ $0 \leq r \leq 31$	Сложить с переносом	$Rd \leftarrow Rd + Rr + C$	Z, C, N, V, H	1
ADIW	Rd, K dE{24, 26, 28, 30} $0 \leq K \leq 63$	Сложить непосредственное значение со словом	$Rdh:Rdl \leftarrow Rdh:Rdl + K$	Z, C, N, V	2
SUB	Rd, Rr $0 \leq d \leq 31$ $0 \leq r \leq 31$	Вычитать без заема	$Rd \leftarrow Rd - Rr$	Z, C, N, V, H	1
SUBI	Rd, K $16 \leq d \leq 31$ $0 \leq K \leq 255$	Вычитать непосредственное значение	$Rd \leftarrow Rd - K$	Z, C, N, V, H	1
SBC	Rd, Rr $0 \leq d \leq 31$ $0 \leq r \leq 31$	Вычитать с заемом	$Rd \leftarrow Rd - Rr - C$	Z, C, N, V, H	1
SBCI	Rd, K $16 \leq d \leq 32$ $0 \leq K \leq 255$	Вычитать непосредственное значение с заемом	$Rd \leftarrow Rd - K - C$	Z, C, N, V, H	1
SBIW	Rd, K dE{24, 26, 28, 30} $0 \leq K \leq 63$	Вычитать непосредственное значение из слова	$Rdh:Rdl \leftarrow Rdh:Rdl - K$	Z, C, N, V	2
AND	Rd, Rr $0 \leq d \leq 31$ $0 \leq r \leq 31$	Выполнить логическое AND	$Rd \leftarrow Rd \cdot Rr$	Z, N, V	1
ANDI	Rd, K $16 \leq d \leq 31$ $0 \leq K \leq 255$	Выполнить логическое AND	$Rd \leftarrow Rd \cdot K$	Z, N, V	1
OR	Rd, Rr $0 \leq d \leq 31$ $0 \leq r \leq 31$	Выполнить логическое OR	$Rd \leftarrow Rd \vee Rr$	Z, N, V	1
ORI	Rd, K $16 \leq d \leq 31$ $0 \leq K \leq 255$	Выполнить логическое OR с непосредственным значением	$Rd \leftarrow Rd \vee K$	Z, N, V	1
EOR	Rd, Rr $0 \leq d \leq 31$ $0 \leq r \leq 31$	Выполнить исключающее OR	$Rd \leftarrow Rd \oplus Rr$	Z, N, V	1

Продолжение Приложения А

Мнемоника	Операнды	Описание	Операция	Флаги	Кол-во циклов
COM	Rd $0 \leq d \leq 31$	Выполнить дополнение до единицы	$Rd \leftarrow \text{SFF} - Rd$	Z, C, N, V	1
NEG	Rd $0 \leq d \leq 31$	Выполнить дополнение до двух	$Rd \leftarrow S00 - Rd$	Z, C, N, V, H	1
SBR	Rd, K $16 \leq d \leq 31$ $0 \leq K \leq 255$	Установить биты в регистре	$Rd \leftarrow Rd \vee K$	Z, N, V	1
Мнемоника	Операнды	Описание	Операция	Флаги	Кол-во циклов
CBR	Rd, K $16 \leq d \leq 31$ $0 \leq K \leq 255$	Очистить биты в регистре	$Rd \leftarrow Rd \cdot (\text{SFF} - K)$	Z, N, V	1
INC	Rd $0 \leq d \leq 31$	Инкрементировать	$Rd \leftarrow Rd + 1$	Z, N, V	1
DEC	Rd $0 \leq d \leq 31$	Декрементировать	$Rd \leftarrow Rd - 1$	Z, N, V	1
TST	Rd $0 \leq d \leq 31$	Проверить на ноль или минус	$Rd \leftarrow Rd.Rd$	Z, N, V	1
CLR	Rd $0 \leq d \leq 31$	Очистить регистр	$Rd \leftarrow Rd \oplus Rd$	Z, N, V	1
SER	Rd $16 \leq d \leq 31$	Установить все биты регистра	$Rd \leftarrow \text{SFF}$	нет	1
CP	Rd, Rr $0 \leq d \leq 31$ $0 \leq r \leq 31$	Сравнить	$Rd - Rr$	Z, C, N, V, H	1
CPC	Rd, Rr $0 \leq d \leq 31$ $0 \leq r \leq 31$	Сравнить с учетом переноса	$Rd - Rr - C$	Z, C, N, V, H	1
CPI	Rd, K $16 \leq d \leq 31$ $0 \leq K \leq 255$	Сравнить с константой	$Rd - K$	Z, C, N, V, H	1

Команды сдвигов и операций с битами

Мнемоника	Операнды	Описание	Операция	Флаги	Кол-во циклов
LSL	Rd $0 \leq d \leq 31$	Логически сдвинуть влево	$Rd(n+1) \leftarrow Rd(n)$, $Rd(0) \leftarrow 0$, $C \leftarrow Rd(7)$	Z, C, N, V, H	1
LSR	Rd $0 \leq d \leq 31$	Логически сдвинуть вправо	$Rd(n) \leftarrow Rd(n+1)$, $Rd(7) \leftarrow 0$, $C \leftarrow Rd(0)$	Z, C, N, V	1
ROL	Rd $0 \leq d \leq 31$	Сдвинуть влево через перенос	$Rd(0) \leftarrow C$, $Rd(n+1) \leftarrow Rd(n)$, $C \leftarrow Rd(7)$	Z, C, N, V, H	1

Продолжение Приложения А

Мнемоника	Операнды	Описание	Операция	Флаги	Кол-во циклов
ROR	Rd $0 \leq d \leq 31$	Сдвинуть вправо через перенос	$Rd(7) \leftarrow C$, $Rd(n) \leftarrow Rd(n+1)$, $C \leftarrow Rd(0)$	Z,C,N,V	1
ASR	Rd $0 \leq d \leq 31$	Арифметически сдвинуть вправо	$Rd(n) \leftarrow Rd(n+1)$, $n=0...6$, $Rd(0) \leftarrow C$	Z,C,N,V	1
SWAP	Rd $0 \leq d \leq 31$	Поменять байты местами	$Rd(3...0) \leftrightarrow Rd(7...4)$	Нет	1
BSET	s $0 \leq s \leq 7$	Установить флаг	$SREG(s) \leftarrow 1$	SREG(s)	1
BCLR	s $0 \leq s \leq 7$	Очистить флаг	$SREG(s) \leftarrow 0$	SREG(s)	1
SBI	P,b $0 \leq P \leq 31$ $0 \leq b \leq 7$	Установить бит в регистр I/O	$I/O(P,b) \leftarrow 1$	Нет	2
CBI	P,b $0 \leq P \leq 31$ $0 \leq b \leq 7$	Очистить бит в регистре I/O	$I/O(P,b) \leftarrow 0$	Нет	2
BST	Rd,b $0 \leq d \leq 31$ $0 \leq b \leq 7$	Переписать бит из регистра во флаг T	$T \leftarrow Rd(b)$	T	1
BLD	Rd,b $0 \leq d \leq 31$ $0 \leq b \leq 7$	Загрузить T флаг в бит регистра	$Rd(b) \leftarrow T$	Нет	1
SEC		Установить флаг переноса	$C \leftarrow 1$	C	1
CLC		Очистить флаг переноса	$C \leftarrow 0$	C	1
SEN		Установить флаг отрицательного значения	$M \leftarrow 1$	N	1
CLN		Очистить флаг отрицательного значения	$N \leftarrow 0$	N	1
SEZ		Установить флаг нулевого значения	$Z \leftarrow 1$	Z	1
CLZ		Очистить флаг нулевого значения	$Z \leftarrow 0$	Z	1
SEI		Установить флаг глобального прерывания	$I \leftarrow 1$	I	1
CLI		Очистить флаг глобального прерывания	$I \leftarrow 0$	I	1
SES		Установить флаг знака	$S \leftarrow 1$	S	1

Продолжение Приложения А

Мнемоника	Операнды	Описание	Операция	Флаги	Кол-во циклов
CLS		Очистить флаг знака	$S \leftarrow 0$	S	1
SEV		Установить флаг переполнения	$V \leftarrow 1$	V	1
CLV		Очистить флаг переполнения	$V \leftarrow 0$	V	1
SET		Установить флаг T	$T \leftarrow 1$	T	1
CLT		Очистить флаг T	$T \leftarrow 0$	T	1
SEH		Установить флаг полу переноса	$H \leftarrow 1$	H	1
CLH		Очистить флаг полу переноса	$H \leftarrow 0$	H	1
NOP		Выполнить холостую команду		Нет	1
SLEEP		Установить режим SLEEP		Нет	1
WDR		Сбросить сторожевой таймер		Нет	1

Команды пересылки данных

Мнемоника	Операнды	Описание	Операция	Флаги	Кол-во циклов
ELPM		Расширенная загрузка из памяти программ в регистр RO	$R0 \leftarrow (Z+RAMPZ)$	Нет	3
MOV	Rd,Rr $0 \leq d \leq 31$ $0 \leq r \leq 31$	Копировать регистр	$Rd \leftarrow Rr$	Нет	1
LDI	Rd,k $16 \leq d \leq 31$ $0 \leq k \leq 255$	Загрузить непосредственное значение	$Rd \leftarrow K$	Нет	1
LDS	Rd,k $0 \leq d \leq 31$ $0 \leq k \leq 65535$	Загрузить из ОЗУ	$Rd \leftarrow (k)$	Нет	3
LD	Rd,X $0 \leq d \leq 31$	Загрузить косвенно	$Rd \leftarrow (X)$	Нет	2
LD	Rd,X+ $0 \leq d \leq 31$	Загрузить косвенно с постинкрементом	$Rd \leftarrow (X),$ $X \leftarrow X+1$	Нет	2
LD	Rd,X- $0 \leq d \leq 31$	Загрузить косвенно с преддекрементом	$X \leftarrow X-1,$ $Rd \leftarrow (X)$	Нет	2

Продолжение Приложения А

Мнемоника	Операнды	Описание	Операция	Флаги	Кол-во циклов
LD	Rd,Y $0 \leq d \leq 31$	Загрузить косвенно	$Rd \leftarrow (Y)$	Her	2
LD	Rd,Y+ $0 \leq d \leq 31$	Загрузить косвенно с постнакрементом	$Rd \leftarrow (Y), Y \leftarrow Y+1$	Her	2
LD	Rd,Y $0 \leq d \leq 31$	Загрузить косвенно с преддекрементом	$Y \leftarrow Y-1, Rd \leftarrow (Y)$	Her	2
LDD	Rd,Y+q $0 \leq d \leq 31$ $0 \leq q \leq 63$	Загрузить косвенно со смещением	$Rd \leftarrow (Y+q)$	Her	2
LD	Rd,Z $0 \leq d \leq 31$	Загрузить косвенно	$Rd \leftarrow (Z)$	Her	2
LD	Rd,Z+ $0 \leq d \leq 31$	Загрузить косвенно с постнакрементом	$Rd \leftarrow (Z), Z \leftarrow Z+1$	Her	2
LD	Rd,-Z $0 \leq d \leq 31$	Загрузить косвенно с преддекрементом	$Z \leftarrow Z-1, Rd \leftarrow (Z)$	Her	2
LDD	Rd,Z+q $0 \leq d \leq 31$ $0 \leq q \leq 63$	Загрузить косвенно со смещением	$Rd \leftarrow (Z+q)$	Her	2
STS	k,Rr $0 \leq d \leq 31$ $0 \leq k \leq 65535$	Загрузить непосредственно в ОЗУ	$(k) \leftarrow Rr$	Her	3
ST	X,Rr $0 \leq r \leq 31$	Записать косвенно	$(X) \leftarrow Rr$	Her	2
ST	X+,Rr $0 \leq r \leq 31$	Записать косвенно с постнакрементом	$(X) \leftarrow Rr, X \leftarrow X+1$	Her	2
ST	-X,Rr $0 \leq r \leq 31$	Записать косвенно с преддекрементом	$X \leftarrow X-1, (X) \leftarrow Rr$	Her	2
ST	Y,Rr $0 \leq r \leq 31$	Записать косвенно	$(Y) \leftarrow Rr$	Her	2
ST	Y+,Rr $0 \leq r \leq 31$	Записать косвенно с постнакрементом	$(Y) \leftarrow Rr, Y \leftarrow Y+1$	Her	2
ST	-Y,Rr $0 \leq r \leq 31$	Записать косвенно с преддекрементом	$Y \leftarrow Y-1, (Y) \leftarrow Rr$	Her	2
STD	Y+q,Rr $0 \leq r \leq 31$ $0 \leq q \leq 63$	Записать косвенно со смещением	$(Y+q) \leftarrow Rr$	Her	2
ST	Z,Rr $0 \leq r \leq 31$	Записать косвенно	$(Z) \leftarrow Rr$	Her	2
ST	Z+,Rr $0 \leq r \leq 31$	Записать косвенно с постнакрементом	$(Z) \leftarrow Rr, Z \leftarrow Z+1$	Her	2

Продолжение Приложения А

Мнемоника	Операнды	Описание	Операция	Флаги	Кол-во циклов
ST	-Z,Rr $0 \leq r \leq 31$	Записать косвенно с преддекрементом	$Z \leftarrow Z-1, (Z) \leftarrow Rr$	Нет	2
STD	Z+q,Rr $0 \leq r \leq 31$ $0 \leq q \leq 63$	Записать косвенно со смещением	$(Z+q) \leftarrow Rr$	Нет	2
LPM		Загрузить байт из памяти программ	$R0 \leftarrow (Z)$	Нет	3
IN	Rd,P $0 \leq d \leq 31$ $0 \leq P \leq 63$	Загрузить данные из порта I/O в регистр	$Rd \leftarrow P$	Нет	1
OUT	P,Rr $0 \leq r \leq 31$ $0 \leq P \leq 63$	Записать данные из регистра в порт I/O	$P \leftarrow Rr$	Нет	1
PUSH	Rr $0 \leq r \leq 31$	Сохранить регистр в стеке	$STACK \leftarrow Rr$	Нет	2

Команды переходов

Мнемоника	Операнды	Описание	Операция	Флаги	Кол-во циклов
RJMP	k $-2K < k < 2K$	Перейти относительно	$PC \leftarrow PC + k + 1$	Нет	2
LJMP		Перейти косвенно	$PC \leftarrow Z$	Нет	2
JMP	k $0 < k < 4M$	Перейти	$PC \leftarrow k$	Нет	3
RCALL	k $-2K \leq k \leq 2K$	Вызвать подпрограмму относительно	$PC \leftarrow PC + k + 1$	Нет	3
ICALL		Вызвать подпрограмму косвенно	$PC \leftarrow Z$	Нет	3
CALL	k $0 \leq k \leq 64K$	Выполнить длинный вызов подпрограммы	$PC \leftarrow k$	Нет	4
RET		Вернуться из подпрограммы	$PC \leftarrow STACK$	Нет	4
RETI		Вернуться из прерывания	$PC \leftarrow STACK$	I	4
CPSE	Rd,Rr $0 \leq d \leq 31$, $0 \leq r \leq 31$	Сравнить и пропустить, если равно	If $Rd=Rr$ then $PC \leftarrow PC + 2$ (or 3)	Нет	1/2/3
SBRC	Rr,b $0 \leq r \leq 31$ $0 \leq b \leq 7$	Пропустить, если бит в регистре очищен	if $Rr(b)=0$ then $PC \leftarrow PC + 2$ (or 3)	Нет	1/2/3

Продолжение Приложения А

Мнемоника	Операнды	Описание	Операция	Флаги	Кол-во циклов
SBRS	Rr,b 0≤r≤31 0≤b≤7	Пропустить, если бит в регистре установлен	If Rr(b)=1 then PC ← PC + 2 (or 3)	Нет	½/3
SBIC	P,b 0≤P≤31 0≤b≤7	Пропустить, если бит в регистре I/O очищен	if I/O P(b)=0 then PC ← PC + 2 (or 3)	Нет	½/3
SBIS	P,b 0≤P≤31 0≤b≤7	Пропустить, если бит в регистре I/O установлен	if I/O P(b)=1 then PC ← PC + 2 (or 3)	Нет	½/3
BRBS	s,k 0≤s≤7 -64≤k≤+63	Перейти, если бит в регистре статуса установлен	if SREG(s)=1 then PC ← PC + k + 1	Нет	½
BRBC	s,k 0≤s≤7 -64≤k≤+63	Перейти, если бит в регистре статуса очищен	if SREG(s)=0 then PC ← PC + k + 1	Нет	½
BREQ	k -64≤k≤+63	Перейти, если равно	if Rd=Rr (Z=1) then PC ← PC + k + 1	Нет	½
BRNE	k -64≤k≤+63	Перейти, если не равно	if Rd≠Rr (Z=0) then PC ← PC + k + 1	Нет	½
BRCS	k -64≤k≤+63	Перейти, если флаг переноса установлен	if C=1 then PC ← PC + k + 1	Нет	1/2
BRC C	k -64≤k≤+63	Перейти, если флаг переноса очищен	if C=0 then PC ← PC + k + 1	Нет	1/2
BRSH	K -64≤k≤+63	Перейти, если равно или больше (без знака)	if Rd<Rr (C=0) then PC ← PC + k + 1	Нет	1/2
BRL O	k -64≤k≤+63	Перейти, если меньше (без знака)	if Rd<Rr (C=1) then PC ← PC + k + 1	Нет	1/2
BRMI	k -64≤k≤+63	Перейти, если минус	if N=1 then PC ← PC + k + 1	Нет	1/2
BRPL	k -64≤k≤+63	Перейти, если плюс	if N=0 then PC ← PC + k + 1	Нет	1/2
BRG E	k -64≤k≤+63	Перейти, если больше или равно (с учетом знака)	if Rd>Rr (N⊕V=0) then PC ← PC + k + 1	Нет	1/2
BRLT	k -64≤k≤+63	Перейти, если меньше чем (со знаком)	if Rd<Rr (N⊕V=1) then PC ← PC + k + 1	Нет	1/2

Продолжение Приложения А

Мнемоника	Операнды	Описание	Операция	Флаги	Кол-во циклов
BRHS	K $-64 \leq k \leq +63$	Перейти, если флаг полупереноса установлен	if H=1 then $PC \leftarrow PC + k + 1$	Нет	1/2
BRH C	k $-64 \leq k \leq +63$	Перейти, если флаг полупереноса очищен	if H=0 then $PC \leftarrow PC + k + 1$	Нет	1/2
BRTS	k $-64 \leq k \leq +63$	Перейти, если флаг T установлен	if T=1 then $PC \leftarrow PC + k + 1$	Нет	1/2
BRTC	k $-64 \leq k \leq +63$	Перейти, если флаг T очищен	if T=0 then $PC \leftarrow PC + k + 1$	Нет	1/2
BRVS	k $-64 \leq k \leq +63$	Перейти, если флаг переполнения установлен	if V=1 then $PC \leftarrow PC + k + 1$	Нет	1/2
BRV C	k $-64 \leq k \leq +63$	Перейти, если флаг переполнения очищен	if V=0 then $PC \leftarrow PC + k + 1$	Нет	1/2
BRIE	k $-64 \leq k \leq +63$	Перейти, если глобальное прерывание разрешено	if I=1 then $PC \leftarrow PC + k + 1$	Нет	1/2
BRID	k $-64 \leq k \leq +63$	Перейти, если глобальное прерывание запрещено	if I=0 then $PC \leftarrow PC + k + 1$	Нет	1/2

Список литературы

1. Водовозов А.М. Элементы систем автоматики. - М.: Академия, 2006. - 224с.
2. Голубцов М.С. Микроконтроллеры AVR: от простого к сложному. – М.: СОЛОН-Пресс, 2003. – 288 с.
3. Программирование микроконтроллера Atmega 8535 на ассемблере: Методические указания к выполнению лабораторных работ. – Челябинск, Учтех-Профи, 2011.
4. Ревич Ю.В. Практическое программирование микроконтроллеров Atmel AVR на языке ассемблера. – СПб.: БВХ-Петербург, 2008. – 384 с.