Министерство образования и науки Российской Федерации

Санкт-Петербургский политехнический университет Петра Великого

—

**Институт** **кибербезопасности и защиты информации**

Отчёт

по лабораторной работе №1

**Основы программирования МК ATmega328р**

**Вариант 3**

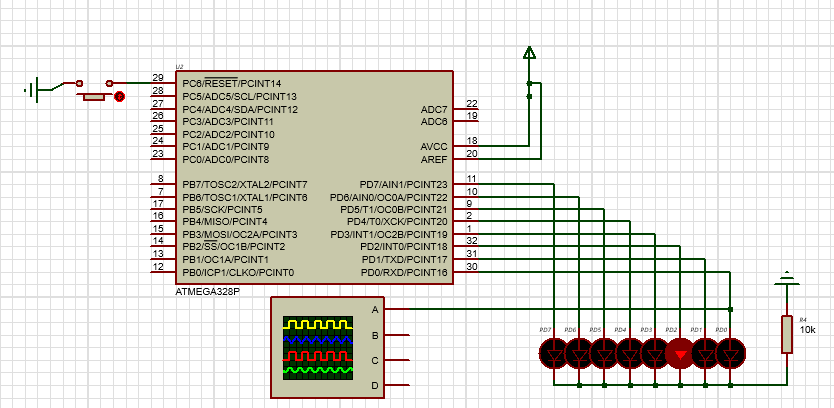
по дисциплине «Аппаратные средства вычислительной техники»

|  |  |  |  |
| --- | --- | --- | --- |
| Выполнили: | студенты группы 4831001/10003 |  | Г. А. Улановский |
|  |  | (подпись, дата) |  |
|  |  |  | Г. Г. Фидаров |
|  |  | (подпись, дата) |  |
|  |  |  |  |
|  |  |  |  |
| Проверил: | доцент, к.т.н. |  | П. О. Семенов |
|  |  | (подпись, дата) |  |

# Формулировка задания

1. В базовом примере заменить фрагмент кода, выделенный зелёным, на фрагмент из своего варианта.
2. Создать новый проект в Atmel Studio на языке ассемблера, разместить исходный код из п.1 в файле проекта. Сформировать «.lss»-файл и «.hex»-файл. Записать «.hex»-файл в контроллер с помощью программы AVRDUDE.
3. Изучить архитектуру и систему команд микроконтроллера ATmega328p. Разобраться в алгоритме работы текущей программы. Убедиться в правильности работы программы.
4. Определить зависимость количества тактов, за которое выполняется заменённый блок кода, от констант x и y.
5. Вычислить значения констант x и y при которых блок кода «delay» будет выполняться ровно 0,1 секунды (при тактовой частоте 8 МГц). При невозможности обеспечения точной величины задержки необходимо дополнить блок кода соответствующим количеством команд NOP (на месте закомментированной команды NOP в базовом примере).
6. Изучить сформированный «.lss»-файл, выписать адреса всех меток программы, перечислить используемые форматы команд в части состава и размерности операндов.
7. Изучить структуру сформированного «.hex»-файла, определить количество записей в файле и количество машинных слов программы.
8. Взять команду ассемблера в соответствии с вариантом и описать порядок её выполнения внутри центрального процессора: определить этапы командного цикла, задействованные узлы МК, состав пересылаемых данных и управляющих сигналов.

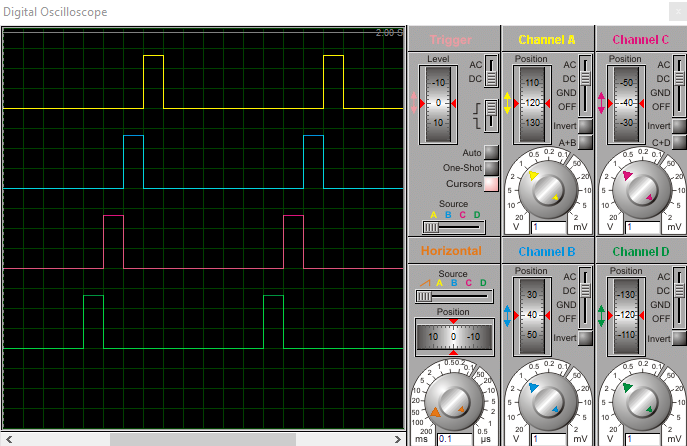
# Схема лабораторной установки



# Блок-схема алгоритма работы программы

СМ .drawio файл

# Временные диаграммы логических сигналов на портах МК (фрагмент)



+ см pdf с примером оформления

# Результаты работы

### Определение зависимости количества тактов от констант x и y

СМ .ру файл

### Анализ lss-файла

СМ lss-файл и первый лист .xlsx

### Анализ hex-файла

CM hex-файл

### Алгоритм выполнения команды ассемблера

# Ответы на контрольные вопросы

1. Укажите, в чём проявляются признаки RISC-архитектуры в микроконтроллере ATmega32. В чём преимущества и недостатки приведённых особенностей?
2. От чего зависит время выполнения команд SBIS и BRLT? Приведите примеры кода (до 3-10 команд каждый), приводящие к различной продолжительности выполнения указанных команд
3. Укажите команды и их аргументы в виде имён регистров общего назначения, портов ввода-вывода и констант в десятичной или шестнадцатеричной системе счисления для следующих машинных слов: 1001 0110 0111 1111, 1111 0011 1110 1101, 1010 1010 0011 1010 и 1001 1010 0011 0110?
4. Приведите пример выполнения циклического сдвига вправо 8-разрядного числа (с переходом младшего разряда в старший) без использования флага Т. Приведите пример из трёх машинных команд, обеспечивающих сложение 24-разрядного числа с 24-разрядной константой.
5. Обоснуйте, чем вызваны ограничения допустимых значений номеров регистров и диапазонов констант в некоторых командах микроконтроллера ATmega32?

# Выводы по лабораторной работе

# Приложение 1 Комментированный листинг программы для МК на языке ассемблера

.device atmega328p ; Необходимо для компилятора

.def TMP = R20 ; Установка R20 как регистра для временного хранения

.org $000

**JMP** reset ; Указатель на начало программы

; Функция паузы

delay:

LDI R29, 20 ; Установка регистра R29 (x) в значение 20

LDI R30, 250 ; Установка регистра R30 (y) в значение 250

delay\_sub:

**INC** R29 ; Инкрементируем x

**NOP** ; Много nop

**NOP**

**NOP**

**NOP**

**NOP**

**NOP**

**NOP**

**NOP**

**NOP**

**NOP**

**NOP**

**NOP**

**NOP**

**NOP**

**NOP**

**NOP**

**NOP**

**NOP**

**NOP**

**NOP**

**NOP**

**NOP**

BRNE delay\_sub ; Если x стал 0, пропускаем. Иначе переход

**NOP**

**DEC** R30 ; �"екриментируем y

BRNE delay\_sub ; Если y стал 0, пропускаем. Иначе переход

**RET** ; �'озврат по адресу возврата в основной цикл

; Начальная настройка

reset:

; настройка исходных значений

LDI TMP, 0x01 ; Положили 1 в temp

**MOV** R0, TMP ; Положили temp в R0

CLR TMP ; Очистили temp

; настройка портов ввода-вывода

SER TMP ; Положили 0xFF в temp (-//- LDI 0xFF)

**OUT** DDRD, TMP ; Открыли все восемь пинов порта D на запись

; Установка вершины стека в конец ОЗУ

LDI TMP, HIGH(RAMEND) ; Старшие разряды адреса

**OUT** SPH, TMP ; Установили старшие разряды ESP

LDI TMP, LOW(RAMEND) ; Младшие разряды адреса

**OUT** SPL, TMP ; Установили младшие разряды ESP

; Основной цикл

loop:

; Циклический сдвиг 8-разрядного числа R0

**ROR** R0 ; циклический сдвиг вправо

; Вывод 8-разрядного числа R0 на порт PORTD

**OUT** PORTD, R0

; Пауза

**CALL** delay;

; Возврат в начало основного цикла

RJMP **loop** ;

# Приложение 2 Фрагмент lss-файла

# Приложение 3 hex-файл