

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Матвеев Александр Сергеевич
Должность: И.о. начальника учебно-методического управления
Дата подписания: 13.12.2023 15:58:15
Уникальный программный ключ:
49d4975072043f866eaf354936262c78745e9

Приложение к ППССЗ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Российский государственный аграрный университет –
МСХА имени К.А. Тимирязева»
(ФГБОУ ВО РГАУ-МСХА имени К.А. Тимирязева)
ТЕХНОЛОГИЧЕСКИЙ КОЛЛЕДЖ

МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ ЛАБОРАТОРНЫХ РАБОТ

по дисциплине
ОП.07 «Основы вычислительной техники»

**специальность: 15.02.10 Мехатроника и
мобильная робототехника (по отраслям)**

форма обучения: очная

Пояснительная записка

Методические указания по выполнению лабораторных работ подготовлены на основе рабочей программы учебной дисциплины «Основы вычислительной техники», разработанной на основе ФГОС СПО по специальности 15.02.10 «Мехатроника и мобильная робототехника (по отраслям)» и соответствующих профессиональных (ПК) компетенций:

ПК 1.2. Осуществлять настройку и конфигурирование программируемых логических контроллеров и микропроцессорных систем в соответствии с принципиальными схемами подключения.

ПК 1.3. Разрабатывать управляющие программы мехатронных систем в соответствии с техническим заданием.

ПК 3.1. Составлять схемы простых мехатронных систем в соответствии с техническим заданием.

ПК 3.2. Моделировать работу простых мехатронных систем.

ПК 4.1. Осуществлять настройку и конфигурирование управляющих контроллеров мобильных робототехнических комплексов в соответствии с принципиальными схемами подключения.

ПК 4.2. Разрабатывать управляющие программы мобильных робототехнических комплексов в соответствии с техническим заданием.

ПК 4.3. Осуществлять настройку датчиков и исполнительных устройств мобильных робототехнических комплексов в соответствии с управляющей программой и техническим заданием.

ПК 5.4. Диагностировать неисправности мобильных робототехнических комплексов с использованием алгоритмов поиска и устранения неисправностей.

Целью освоения учебной дисциплины «Основы вычислительной техники» является:

- теоретическая и практическая подготовка студентов в области информационных технологий в такой степени, чтобы они могли выбирать необходимые технические, алгоритмические, программные и технологические решения, уметь объяснить принципы их функционирования и правильно их использовать;
- формирование у студентов знаний по дисциплине, достаточных для самостоятельного освоения вычислительных систем с новыми архитектурами.

При выполнении лабораторных работ студент должен **уметь**:

- настраивать и конфигурировать ПЛК в соответствии с принципиальными схемами подключения;
- программировать ПЛК с целью анализа и обработки цифровых и аналоговых сигналов и управления исполнительными механизмами мехатронных систем;
- применять специализированное программное обеспечение при разработке управляющих программ и визуализации процессов управления и работы мехатронных систем;

- проводить расчеты параметров типовых электрических, пневматических и гидравлических схем узлов и устройств, разрабатывать несложные мехатронные системы;
- составлять структурные, функциональные и принципиальные схемы мехатронных систем;
- применять специализированное программное обеспечение при моделировании мехатронных систем;
- использовать стандартные пакеты (библиотеки) языка для решения практических задач;
- решать исследовательские и проектные задачи с использованием компьютеров;
- решать конфигурационные задачи с использованием компьютеров при построении системы управления мобильным роботом;
- понимание систем программирования и управления мобильными роботами;
- понимание технологии построения беспроводной сети и взаимосвязи робота и компьютера, используя данную технологию;
- использование поставляемого производителем программного обеспечения для анализа передаваемых датчиками данных, и обеспечение диагностики роботом на основе данных, поступающих с датчиков.

При выполнении лабораторных работ студент должен **знать**:

- принципы связи программного кода, управляющего работой ПЛК, с действиями исполнительных механизмов;
- методы непосредственного, последовательного и параллельного программирования;
- алгоритмы поиска ошибок управляющих программ ПЛК;
- промышленные протоколы для объединения ПЛК в сеть;
- языки программирования и интерфейсы ПЛК;
- технологии разработки алгоритмов управляющих программ ПЛК;
- типовые модели мехатронных систем;
- основные факты, базовые концепции и модели информатики; основы технологии работы на ПК в современных операционных средах;
- технологию работы на ПК в современных операционных средах, основные методы разработки алгоритмов и программ, структуры данных, используемые для представления типовых информационных объектов, типовые алгоритмы обработки данных; основные принципы и методологию разработки прикладного программного обеспечения, включая типовые способы организации данных и построения алгоритмов обработки данных, синтаксис и семантику универсального алгоритмического языка программирования высокого уровня;
- современных основ информационно-коммуникационных технологий для решения некоторых типовых задач в проектировании мобильных роботов;
- методов построения современных мобильных роботов.

Содержание лабораторных занятий определено рабочей программой и тематическим планированием, соответствует теоретическому материалу изучаемых разделов учебной дисциплины.

Объём лабораторных занятий по дисциплине определяется учебным планом по данной специальности.

Продолжительность лабораторного занятия - 2 академических часа. Перед проведением лабораторного занятия преподавателем организуется инструктаж, а по его окончании – обсуждение итогов.

Комплект методических указаний по выполнению лабораторных работ дисциплины «Основы вычислительной техники» содержит 8 лабораторных занятий.

**Перечень лабораторных работ
по дисциплине «Основы вычислительной техники»**

ЛАБОРАТОРНАЯ РАБОТА №1.

Тема: Логические элементы электронно-вычислительной техники (ЭВТ).

ЛАБОРАТОРНАЯ РАБОТА №2.

Тема: Логические элементы электронно-вычислительной техники (ЭВТ).

ЛАБОРАТОРНАЯ РАБОТА №3.

Тема: Типовые комбинационные цифровые устройства.

ЛАБОРАТОРНАЯ РАБОТА №4.

Тема: Типовые комбинационные цифровые устройства.

ЛАБОРАТОРНАЯ РАБОТА №5.

Тема: Организация интерфейсов в вычислительной технике.

ЛАБОРАТОРНАЯ РАБОТА №6.

Тема: Организация интерфейсов в вычислительной технике.

ЛАБОРАТОРНАЯ РАБОТА №7.

Тема: Способы адресации.

ЛАБОРАТОРНАЯ РАБОТА №8.

Тема: Способы адресации.

ЛАБОРАТОРНАЯ РАБОТА № 1

Тема: «Логические элементы электронно-вычислительной техники (ЭВТ)»

Цель: научиться представлять логические функции.

Оборудование: справочный материал, персональный компьютер с выходом в Интернет.

Справочный материал

Основные понятия и определения.

Определение. Формула называется тождественно-истинной (тавтологией), если для любых наборов переменных она принимает значение И.

Определение. Формула называется тождественно тождественно-ложной, если для любых наборов переменных она принимает значение Л.

В алгебре высказываний используют две нормальные формы: дизъюнктивную и конъюнктивную нормальные формы формулы (ДНФ и КНФ).

Дизъюнктивной нормальной формой (ДНФ) называется дизъюнкция простых конъюнкций.

Конъюнктивной нормальной формой (КНФ) формулы есть формула, равносильная исходной формуле логики высказываний и записанная в виде конъюнкции элементарных дизъюнкций переменных.

Каждая формула, не равная тождественно Л, может быть приведена СДНФ, которая является единственной с точностью до перестановки дизъюнктивных членов.

Каждая формула, не равная тождественно И, может быть приведена к СКНФ, которая является единственной с точностью до перестановки конъюнктивных членов.

Совершенная дизъюнктивная нормальная форма формулы (СДНФ) это равносильная ей формула, представляющая собой дизъюнкцию элементарных конъюнкций, обладающая свойствами

1. Каждое логическое слагаемое формулы содержит все высказывания, входящие в формулу.
2. Все логические слагаемые формулы различны
3. Ни одно логическое слагаемое не содержит высказывание и его отрицание
4. Ни одно логическое слагаемое формулы не содержит одно и то же высказывание дважды.

Алгоритм получения СКНФ по таблице истинности:

- 1) Отметить те строки, в последнем столбце которых стоят 0:
- 2) Выписать для каждой отмеченной строки дизъюнкцию всех переменных следующим образом: если значение некоторой переменной в данной строке =0, то в дизъюнкцию включают саму эту переменную, если =1, то ее отрицание:
- 3) Все полученные дизъюнкции связать в конъюнкцию.

Образец решения

Построить таблицу истинности для высказывания: $(x \mid \bar{y}) \rightarrow (y \oplus z)$, построить СДФ, СКНФ, найти минимальную ДНФ.

Решение.

Строим таблицу истинности- таблицу, с помощью которой устанавливается истинностное значение сложного высказывания при данных значениях входящих в него простых высказываний.

x	y	z	\bar{y}	$x \mid \bar{y}$	$y \oplus z$
0	0	0	1	0	0
0	0	1	1	0	1
0	1	0	0	1	1
0	1	1	0	1	0
1	0	0	1	1	0
1	0	1	1	1	1
1	1	0	0	0	1
1	1	1	0	0	0

По таблице составляем дизъюнктивную нормальную форму (ДНФ). ДНФ в булевой логике — нормальная форма, в которой булева формула имеет вид дизъюнкции нескольких конъюнктов.

Алгоритм получения СДНФ по таблице истинности:

- 1) Отметить те строки, в последнем столбце которых стоят 1;
- 2) Выписать для каждой отмеченной строки конъюнкцию всех переменных следующим образом: если значение некоторой переменной в данной строке =1, то в конъюнкцию включают саму эту переменную, если =0, то ее отрицание;
- 3) Все полученные конъюнкции связать в дизъюнкцию:

Выбираем в таблице строки, в которых булева функция принимает значение 1. В данном случае – это 2-ая, 3-ая, 4-ая, 6-ая и 7-ая строки.

Для каждой строки составляем конъюнкцию: если значение переменной равно 0, то берем ее отрицание, а если 1, то берем саму переменную. Затем составляем дизъюнкцию полученных конъюнкций:

$$f(x, y, z) = (x \wedge \bar{y} \wedge z) \vee (x \wedge y \wedge \bar{z}) \vee (x \wedge \bar{y} \wedge \bar{z}) \vee (\bar{x} \wedge \bar{y} \wedge z) \vee (x \wedge \bar{y} \wedge z)$$

Выбираем в таблице строки, в которых булева функция принимает значение 0.

В данном случае – это 1-ая, 5-ая, и 8-ая строки:

$$f(x, y, z) = (\bar{x} \wedge \bar{y} \wedge \bar{z}) \vee (x \wedge \bar{y} \wedge \bar{z}) \vee (x \wedge y \wedge z)$$

ДНФ называется минимальной, если она содержит наименьшее число букв среди всех ДНФ ей равносильных.

Метод Квайна основывается на применении двух основных соотношений.

Соотношение склеивания :

$$(a \wedge b) \vee (\bar{a} \wedge b) = b ; (a \vee b) \wedge (\bar{a} \vee b) = b$$

Соотношение поглощения:

$$a \wedge (a \vee b) = a ; a \vee (a \wedge b) = a$$

Используя соотношение склеивания получаем:

$$(\bar{x} \wedge \bar{y} \wedge \bar{z}) \vee (x \wedge \bar{y} \wedge \bar{z}) = \bar{y} \wedge \bar{z} ;$$

$$(x \wedge y \wedge \bar{z}) \vee (x \wedge y \wedge z) = x \wedge y . \text{ Отсюда,}$$

$$(\bar{x} \wedge \bar{y} \wedge \bar{z}) \vee (x \wedge \bar{y} \wedge \bar{z}) \vee (x \wedge y \wedge \bar{z}) \vee (x \wedge y \wedge z) = (\bar{y} \wedge \bar{z}) \vee (x \wedge y) - \text{сокращенная}$$

ДНФ

Порядок выполнения работы

Задание. Построить таблицу истинности, найти СНДФ, найти минимальную ДНФ для высказывания:

Вариант 1

1. $(\bar{z} \vee y) \rightarrow (\bar{z} \oplus \bar{x})$

2. $\left((\overline{A \wedge B}) \Rightarrow A \right) \Rightarrow A \vee B$

3. $(\bar{z} \vee y) \wedge (\bar{z} \oplus \bar{x})$

Вариант 2

1. $\left((\overline{A \wedge B}) \Rightarrow A \right) \Leftrightarrow (A \vee B)$

2. $x \left((y \rightarrow z) \oplus (x|y) \rightarrow (x|z) \right)$

3. $(\bar{z} \Rightarrow y) \Leftrightarrow (\bar{z} \vee \bar{x})$

Вариант 3

1. $(x|y) \rightarrow (x|z)$

2. $(\overline{A \wedge B}) \Leftrightarrow (\bar{B} \oplus \bar{A}) \Leftrightarrow (A \vee B) \oplus (A \oplus \bar{B})$

3. $(\bar{z} \oplus y) \Rightarrow (\bar{z} | (y \vee \bar{x}))$

Вариант 4

1. $(\overline{A \Rightarrow B}) \Leftrightarrow (\bar{B} \wedge \bar{A})$

2. $(x \wedge y) \oplus (x \wedge z) \Leftrightarrow x \wedge (y \oplus z)$

3. $(\bar{z} \oplus x) \vee (\bar{z} | (y \vee \bar{x}))$

Вариант 5

1. $((x \downarrow y) \rightarrow z) \oplus y$

2. $(x|y) \rightarrow (x|z) \oplus ((\bar{z} \vee y) \rightarrow (\bar{z} \oplus \bar{x}))$

3. $(\bar{z} \vee y) \rightarrow (\bar{z} | (y \vee \bar{x}))$

Вариант 6

1. $(x \vee \bar{y}) \rightarrow (\bar{z} \oplus \bar{x})$

2. $(\overline{A \Rightarrow B}) \Leftrightarrow (\bar{B} \wedge \bar{A}) \oplus ((A \Rightarrow B) \wedge \bar{B}) \Rightarrow A$

3.

Вариант 7

1.

2. $(\overline{A \Rightarrow B}) \vee (\bar{B} \wedge \bar{A}) \Rightarrow ((A \Rightarrow B) \wedge \bar{B}) \oplus A$

$$3. (\bar{z} \vee x) \Leftrightarrow (\bar{z} | (y \vee \bar{x}))$$

Вариант 8

$$1. ((A \vee B) \wedge B) \Rightarrow A$$

$$2. x | (y \Rightarrow z) \Leftrightarrow (x | y) \vee (x | z)$$

$$3. (\bar{z} \Leftrightarrow y) \Leftrightarrow (\bar{z} | (y \oplus \bar{x}))$$

Вариант 9

$$1. (\bar{x} | \bar{y}) \oplus (z \rightarrow \bar{x})$$

$$2. (\overline{A \Rightarrow B}) \vee (\bar{B} \wedge \bar{A}) \Leftrightarrow ((A \Rightarrow B) \oplus \bar{B}) \vee A$$

$$3. ((A \vee B) \oplus \bar{B}) \Rightarrow A$$

Вариант 10

$$1. \left(\overline{A \vee B \wedge A} \right) \Leftrightarrow \bar{A}$$

$$2. (x \wedge y) \vee (x \wedge z) \Rightarrow$$

$$3. (x \vee \bar{y}) \rightarrow (\bar{z} \Leftrightarrow \bar{x})$$

Вариант 11

$$1. (\bar{z} \rightarrow x) \Leftrightarrow (y | x)$$

$$2. (\overline{A \vee B}) \vee (\bar{B} \wedge \bar{A}) \Leftrightarrow ((A \vee B) \oplus \bar{B}) \Rightarrow A$$

$$3. (\bar{z} \oplus y) \vee (\bar{z} | (y \vee \bar{x}))$$

Вариант 12

$$1.$$

$$2. x | (y \oplus z) \oplus (x | y) \vee (x | z)$$

$$3. (\overline{A \vee B}) \Leftrightarrow (\bar{B} \wedge \bar{A})$$

Вариант 13

$$1. (x \wedge y) \oplus (x \wedge z)$$

$$2. (\overline{A \Rightarrow B}) \wedge (\bar{B} \Leftrightarrow \bar{A}) \Leftrightarrow ((A \Rightarrow B) \wedge \bar{B}) \oplus A$$

$$3. (\bar{z} \Leftrightarrow y) \vee (\bar{z} | (z \vee \bar{x}))$$

Вариант 14

$$1. (\bar{x} | \bar{y}) \oplus (\bar{z} \rightarrow x)$$

$$2. (\overline{A \oplus B}) \Leftrightarrow (\bar{B} \oplus \bar{A}) \Leftrightarrow A \Rightarrow ((A \vee B) \wedge \bar{B})$$

$$3. (\bar{z} \Rightarrow y) \oplus (\bar{z} | (y \vee \bar{x}))$$

Вариант 15

$$1. \left((\overline{A \wedge B}) \Rightarrow A \right) \Leftrightarrow (A \downarrow B)$$

$$2. x | (y \wedge z) \Rightarrow (x | y) \oplus (x | z)$$

$$3. (\overline{z} \vee y) \rightarrow (\overline{z} \oplus \overline{x})$$

ЛАБОРАТОРНАЯ РАБОТА № 2

Тема: «Логические элементы электронно-вычислительной техники (ЭВТ)»

Цель: научиться выполнять арифметические операции.

Оборудование: справочный материал, персональный компьютер с выходом в Интернет.

Справочный материал

Сложение	Вычитание	Умножение	Деление
$0 + 0 = 0$	$0 - 0 = 0$	$00 = 0$	$0:1=0$
$0 + 1 = 1$	$1 - 0 = 1$	$0 - 1 = 0$	$1:1 = 1$
$1 + 0 = 1$	$1 - 1 = 0$	$1 - 0 = 0$	
$1 + 1 = 0$ и «1» переноса в ст. разряд суммы	$10 - 1 = 1$ необходима «1» замена из ст. разряда	$1 - 1 = 1$	

Сложение. Рассмотрим сложение чисел в двоичной системе счисления. В его основе лежит таблица сложения одноразрядных двоичных чисел:

$$0 + 0 = 0$$

$$0 + 1 = 1$$

$$1 + 0 = 1$$

$$1 + 1 = 10$$

Важно обратить внимание на то, что при сложении двух единиц происходит переполнение разряда и производится перенос в старший разряд. Переполнение разряда наступает тогда, когда величина числа в нем становится равной или большей основания.

Сложение многоразрядных двоичных чисел происходит в соответствии с вышеприведенной таблицей сложения с учетом возможных переносов из младших разрядов в старшие. В качестве примера сложим в столбик двоичные числа 110_2 и 11_2 :

$$\begin{array}{r} +110_2 \\ 11_2 \\ \hline 1001_2 \end{array}$$

Вычитание. Рассмотрим вычитание двоичных чисел. В его основе лежит таблица вычитания одноразрядных двоичных чисел. При вычитании из меньшего числа (0) большего (1) производится заем из старшего разряда. В таблице заем обозначен 1 с чертой:

$$\begin{array}{l} 0 - 0 = 0, \\ 0 - 1 = \bar{1}1, \\ 1 - 0 = 1, \\ 1 - 1 = 0. \end{array}$$

Вычитание многоразрядных двоичных чисел происходит в соответствии с вышеприведенной таблицей вычитания с учетом возможных заемов из старших

$$\begin{array}{r} -110_2 \\ 11_2 \\ \hline 11_2 \end{array}$$

разрядов.
 110_2 и 11_2 :

В качестве примера произведем вычитание двоичных чисел

Умножение. В основе умножения лежит таблица умножения одноразрядных двоичных чисел:

$$\begin{array}{l} 0 \cdot 0 = 0, \\ 0 \cdot 1 = 0, \\ 1 \cdot 0 = 0, \\ 1 \cdot 1 = 1. \end{array}$$

Умножение многоразрядных двоичных чисел происходит в соответствии с вышеприведенной таблицей умножения по обычной схеме, применяемой в десятичной системе счисления, с последовательным умножением множимого на цифры множителя. В качестве примера произведем умножение двоичных чисел 100_2 и 11_2 :

$$\begin{array}{r}
 \times 110_2 \\
 11_2 \\
 \hline
 110 \\
 110 \\
 \hline
 10010_2
 \end{array}$$

Деление. Операция деления выполняется по алгоритму, подобному алгоритму выполнения операции деления в десятичной системе счисления. В качестве примера произведем деление двоичного числа 100, на 11_2 :

$$\begin{array}{r|l}
 -110_2 & 11_2 \\
 \hline
 11 & 10_2 \\
 \hline
 0 &
 \end{array}$$

Арифметические операции над двоичными числами

В примерах для получения результата при умножении и сложении чисел применение таблиц «Сложение» и «Умножение» начинается с младших разрядов. При делении чисел использование таблицы «Вычитание» начинается со старших разрядов.

Таблица 4

Арифметические операции над двоичными числами

Сложение	Умножение	Деление	
$+10110_2$	X^{10111}_2	$\begin{array}{r} -1000 \\ 11_2 \end{array}$	$\begin{array}{r} 11 \\ 1- \end{array}$
1010_2	1010_2	111	$\begin{array}{r} 11 \\ 01_2 \end{array}$
100000_2	10111_2	$\begin{array}{r} -110 \end{array}$	
	10111_2	111	
	$\begin{array}{r} 11100110 \\ 2 \end{array}$	111	
		ш	
		0	

Любая математическая операция над двоичными числами требует выполнения команд «сложить», «вычитать», «сдвинуть» (вправо или влево).

Можно выполнять арифметические действия в восьмеричной и шестнадцатеричной системах счисления. Необходимо только помнить, что величина переноса в следующий разряд при сложении и заем из старшего разряда при вычитании определяются величиной основания системы

$$\begin{array}{r} +37_8 \\ 25_8 \\ \hline 64_8 \end{array} \qquad \begin{array}{r} -9C_{16} \\ 78_{16} \\ \hline 24_{16} \end{array}$$

счисления:

Для проведения арифметических операций над числами, выраженными в различных системах счисления, необходимо предварительно перевести их в одну и ту же систему.

Порядок выполнения работы

Задание. Ответить на вопросы и выполнить задания

- 1. Что такое система счисления?
- 2. Как называется единица измерения количества информации?
- 3. Какая система счисления одной и той же цифре присваивает различное значение, определяющееся позицией в последовательности цифр, изображающей число?
- 4. Какая система счисления построена на позиционном принципе записи чисел с основанием 10?
- 5. Какая система счисления построена на позиционном принципе записи чисел с основанием 8?
- 6. Что такое триада? Где она используется?
- 7. Что такое тетрада? Где она используется?
- 8. В чем различие между триадой и тетрадой?
- 9. Сложить числа в шестнадцатеричной системе счисления: 2ADEC и FAEDA.
- 10. Умножить числа в восьмеричной системе счисления: 761 и 205.
- 11. Переведите число 703,04 из восьмеричной системы счисления в десятичную.
- 12. Переведите число B2E,4 из шестнадцатеричной системы счисления в десятичную.
- 13. Какие три команды выполняются при любых математических операциях над двоичными числами?

ЛАБОРАТОРНАЯ РАБОТА №3

Тема: «Типовые комбинационные цифровые устройства»

Цель: разобраться в арифметико-логических устройствах.

Оборудование: справочный материал, персональный компьютер с выходом в Интернет.

Справочный материал

Подготовка к выполнению практической работы

1. Изучить описание практической работы.
2. Ответить на все контрольные вопросы.
3. Произвести синтез параллельного АЛУ по варианту.
4. Разработать функциональную схему параллельного АЛУ.
5. Разработать алгоритмы выполнения операций.
6. Выполнить операции над двоичными данными (кодами), согласно списка указанного в таблице вариантов (таблица 3).

Порядок выполнения работы

1. Произвести исследование i -го разряда комбинационной части АЛУ.
2. Произвести исследование АЛУ.

Отчет должен содержать

1. Функциональную схему АЛУ со списком операций и таблицей значений управляющих сигналов.
2. Таблицу установки признаков.

Варианты индивидуальных заданий

Таблица

N/N n/n	Число Разрядов АЛУ	Список команд	Коды операндов	
			A	B
0	3	$A+B, A-1, A-B, A \oplus B, \bar{A}$	011	011
1	4	$A-1, A \vee B, A/B, A+B+1, B$	0110	0111
2	2	$A-B, B-1, B+1, \bar{A} \cdot B, A$	01	11
3	4	$A-B+1, A \oplus B, A \downarrow B, \bar{A}$	0111	0011
4	2	$B-1, A+B, A \cdot \bar{B}, A$	10	01
5	3	$B+1, A-B, A \oplus B, A$	011	101
6	4	$A+B, A-B, B-1, A/B, B$	1110	0010
7	3	$A-1, A+B+1, A \vee B, A$	011	011
8	4	$A-B, A+B+1, B+1, \bar{A} \cdot B, B$	1111	0101

9	2	$B-1, A+B+1, A \cdot \bar{B}, B$	11	10
10	3	$A-1, A+B-1, A \cdot B, \bar{A}, \bar{B}$	111	010
11	2	$B-1, A+B+1, A/B, A$	01	01
12	4	$B+1, A-B+1, A \oplus B, \bar{A}$	0101	0111
13	2	$A-B, A+B+1, B+1, A \downarrow B, B$	11	10
14	3	$B+1, A-B, A+B, B$	011	011
15	4	$A+B, A-B, B+1, A \cdot B, \bar{A}$	1010	0101
16	3	$A+1, A-B-1, A \oplus B, \bar{A} \cdot B, A$	011	110
17	4	$B-1, A+B+1, A/B, B$	0011	0101
18	2	$A-B, A+B+1, A \vee B, A-B, B$	11	01
19	4	$A-1, A+B+1, A \downarrow B, A \oplus B, A$	0111	0011
20	3	$A+1, A+B-1, \bar{A} \vee B, A$	101	011

Контрольные вопросы

1. Назовите этапы синтеза АЛУ?
2. Назначение АЛУ.
3. Для чего служат флаговые регистры?
4. Как производится запись признаков во флаговый регистр?
5. Назначение буферных регистров A и A' , B и B' в схеме АЛУ?
6. Для чего используется схема MS при синтезе АЛУ?
7. Как производится использование сумматора при выполнении арифметической операции сложения?
8. Как используется сумматор при выполнении арифметической операции вычитания?
9. Как используется сумматор при выполнении арифметической операции декремент?
10. Как используется сумматор при выполнении арифметической операции инкремент?

ЛАБОРАТОРНАЯ РАБОТА №4

Тема: «Типовые комбинационные цифровые устройства»

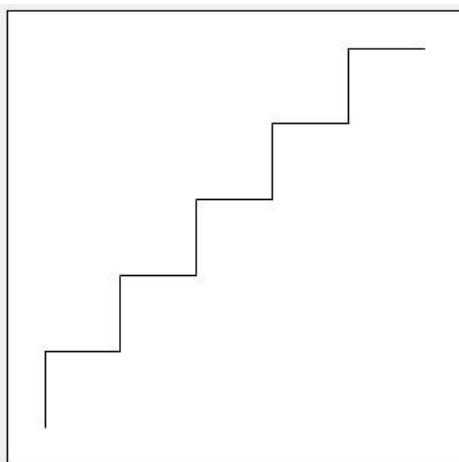
Цель: закрепление навыков программного управления исполнителем алгоритмов.

Оборудование: ПК, исполнитель «Стрелочка».

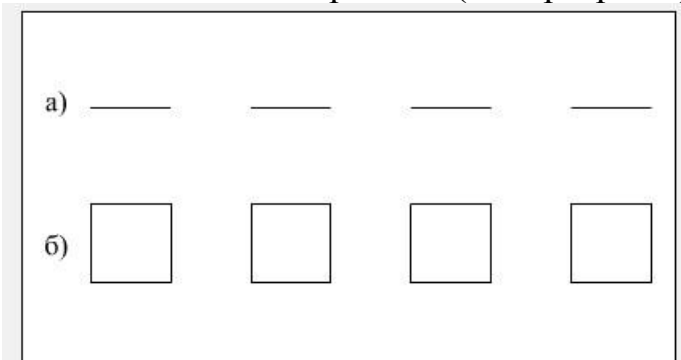
Порядок выполнения работы

Ход работы

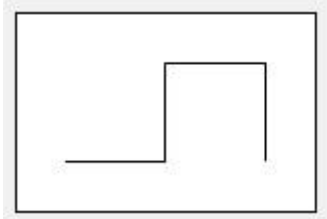
1. С помощью исполнителя алгоритмов стрелочка выполните следующий рисунок.



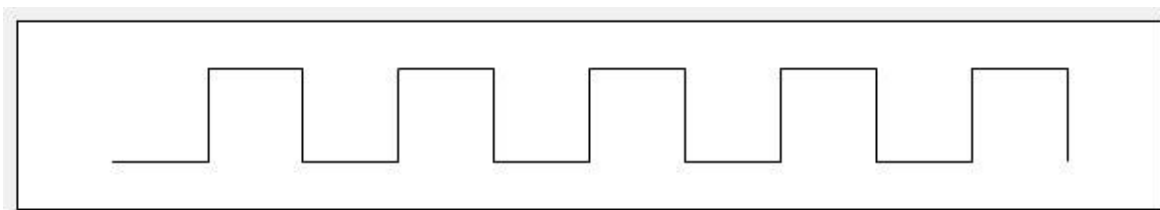
2. Написать программы для рисования следующих рисунков на всю длину листа с использованием вспомогательных алгоритмов (подпрограмм).



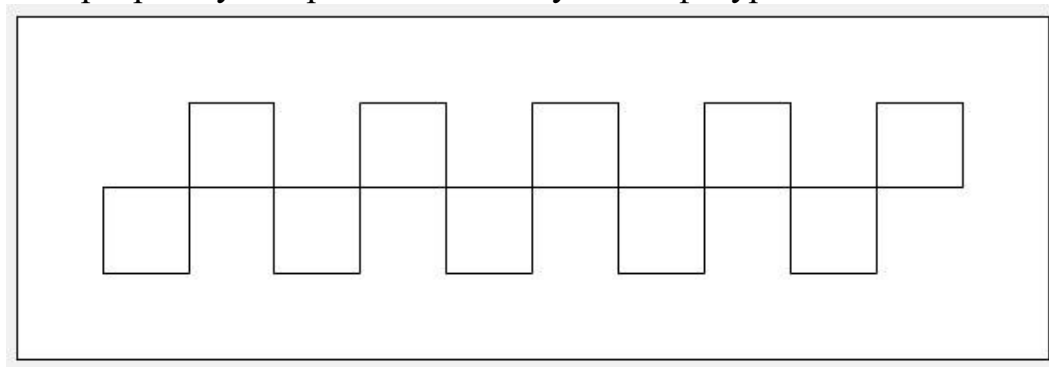
3. Написать подпрограмму для изображения следующей фигуры



4. Используя подпрограмму из предыдущего задания, отобразить следующий элемент.



5. Составить программу для рисования следующей фигуры



ЛАБОРАТОРНАЯ РАБОТА №5

Тема: «Организация интерфейсов в вычислительной технике»

Цель: знакомство с компонентной структурой современного персонального компьютера.

Оборудование: ПК с операционной системой Windows.

Порядок выполнения работы

Задание:

Определение ключевых параметров аппаратного обеспечения рабочего ПК.

Ход работы

Определение типа и характеристик центрального процессора и объёма оперативной памяти.

На рабочем столе найдите иконку **Мой компьютер**. Через контекстное меню вызовите команду **Свойства** и откройте (если она не открыта) вкладку **Общие**.
Мой компьютер □Свойства (в контекстном меню) □Общие
MyComputer □Properties □General

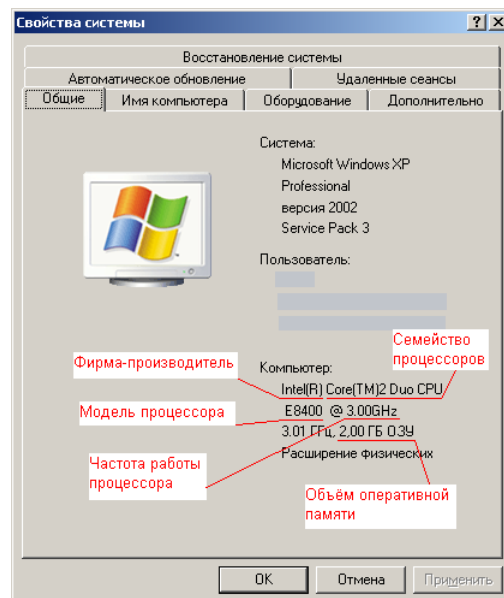
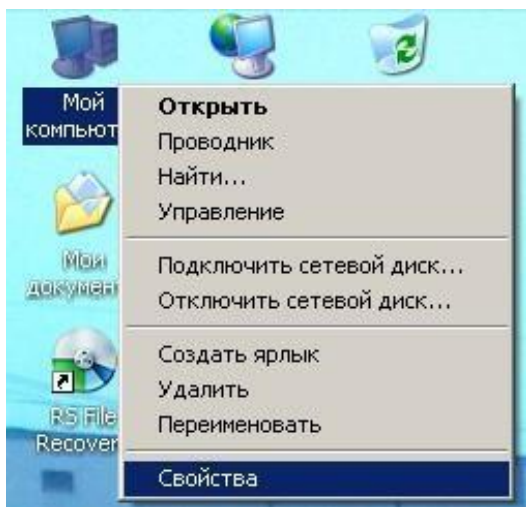
В открывшемся окне найдите информацию о процессоре и оперативной памяти. Для приведённого на рисунках ниже примера:

Фирма производитель процессора: Intel Семейство процессоров: Core2 DUO

Модель процессора: E8400

Частота работы процессора: 3 ГГц Объём оперативной памяти: 2 ГБ

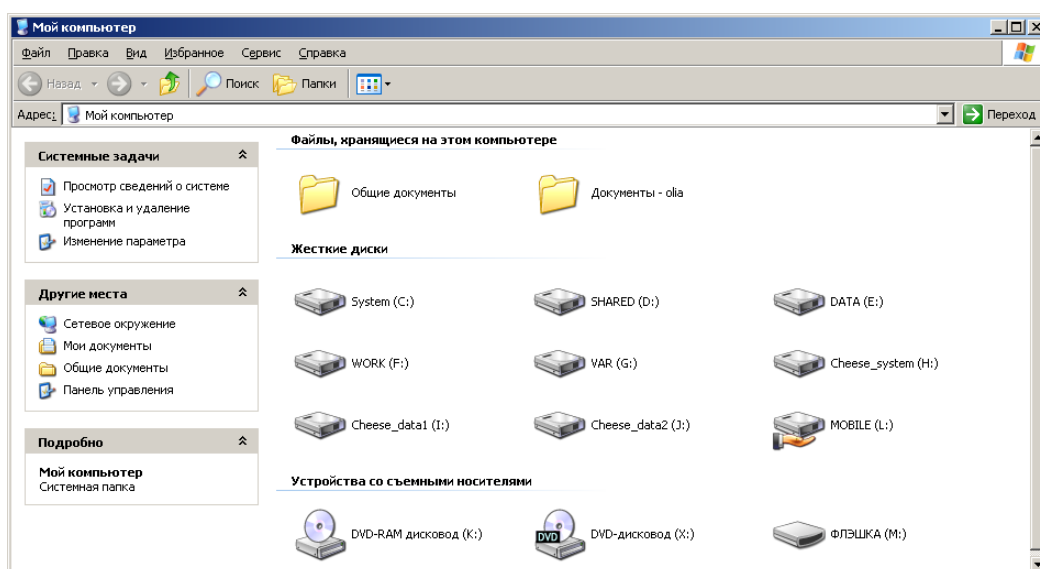
Приведите в отчёте данные о процессоре и оперативной памяти для Вашего рабочего компьютера в лаборатории и для Вашего домашнего компьютера. Сравните полученные характеристики. Какой из компьютеров имеет потенциально бóльшую производительность?



Определение объёма памяти на жестких накопителях

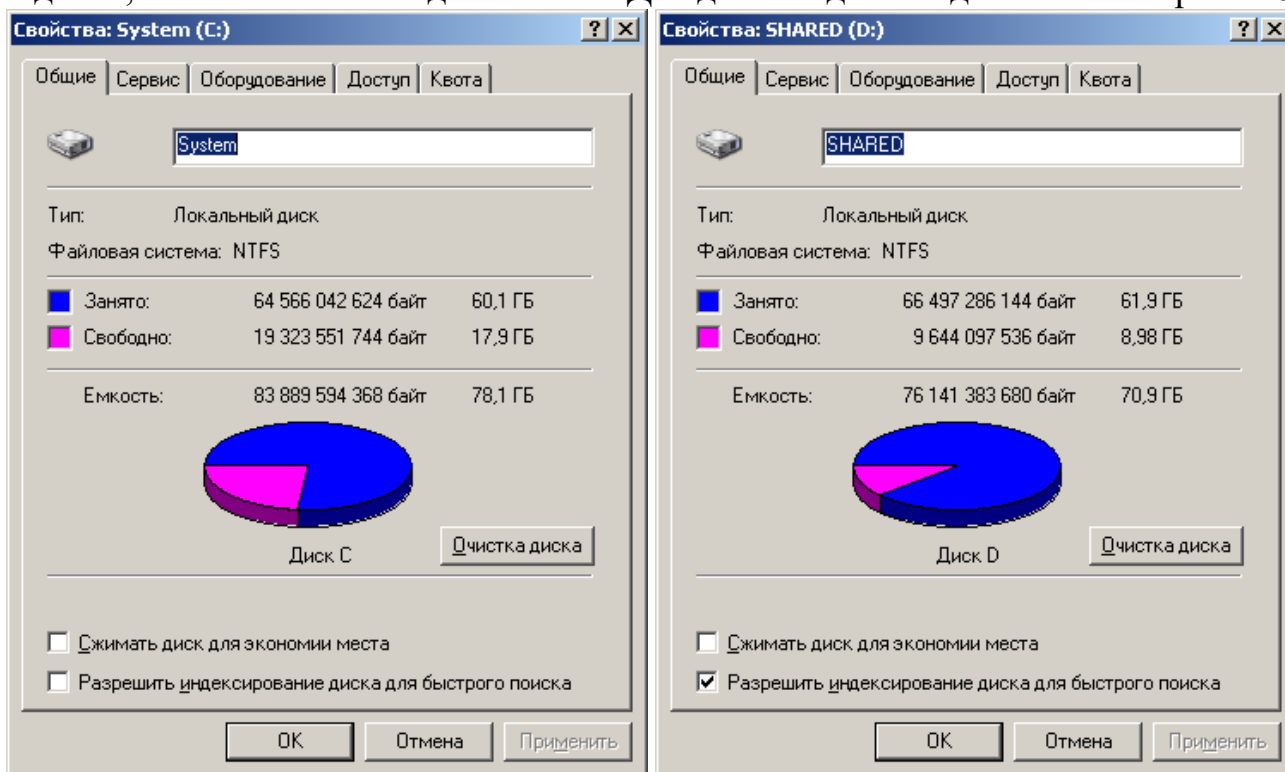
На рабочем столе найдите и дважды щелкните на иконку **Мой компьютер**. В появившемся окне будут показаны иконки для всех внешних накопителей, подключённых в настоящий момент к системе.

My computer (на рабочем столе) контекстное меню жесткого диска Properties



Вызовите окно **Свойства** через пункт меню **Свойства** в контекстном меню одного из дисков. В появившемся окне найдите информацию об общем объёме

диска, о занятом и свободном месте. Для дисководов без дисков объём равен 0.

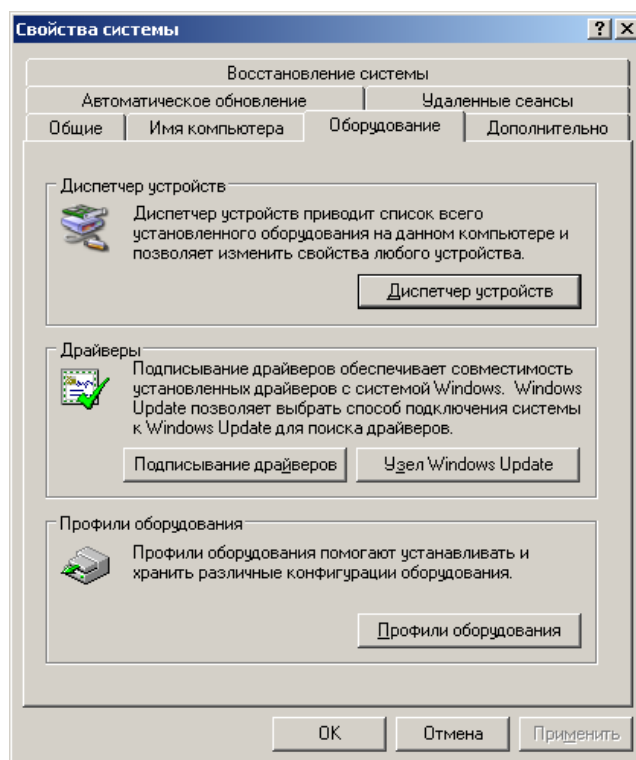


Определение количества физических накопителей, подключённых к компьютеру. Определение модели видеокарты.

Фактическое количество физических накопителей, подключённых к компьютеру, может быть меньше показанного в папке **Мой компьютер**, поскольку один физический накопитель может быть разбит на несколько разделов, отображающихся независимо друг от друга.

На рабочем столе найдите иконку **Мой компьютер**. Через контекстное меню вызовите команду **Свойства**, откройте вкладку **Оборудование** и нажмите кнопку **Диспетчер устройств**.

Мой компьютер □Свойства □Оборудование □Диспетчер устройств
MyComputer □Properties □Hardware □Device manager



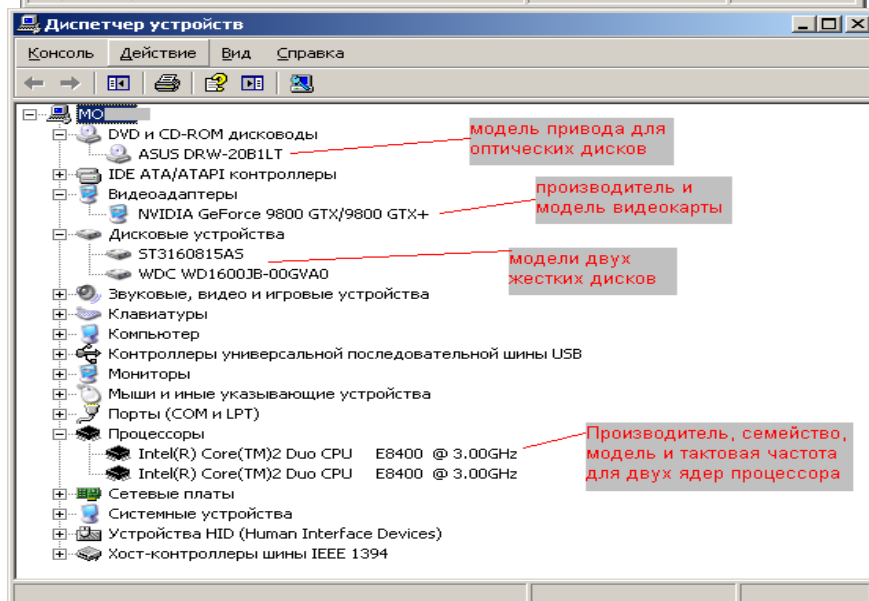
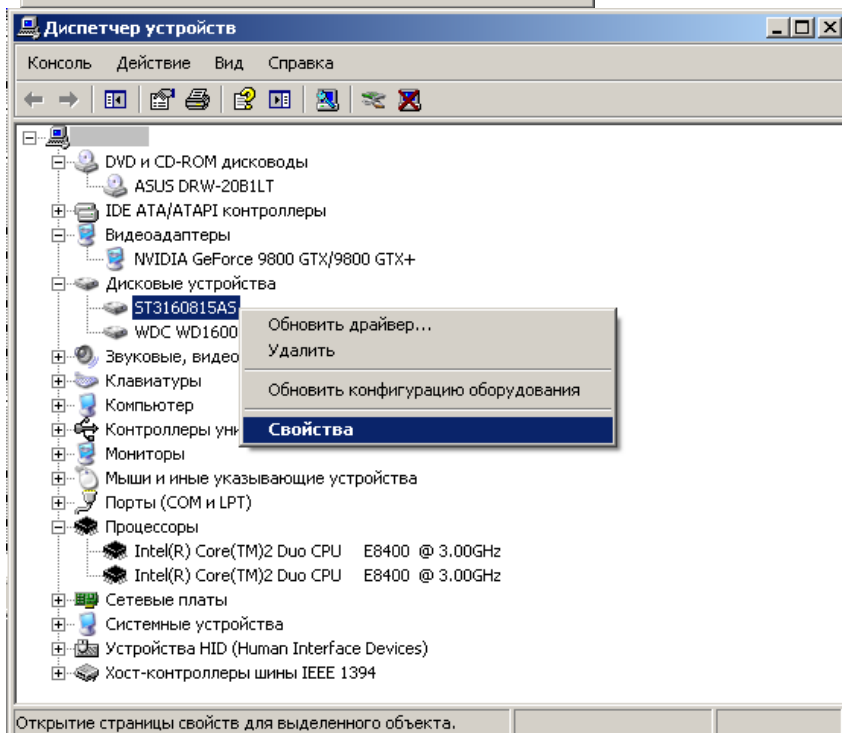
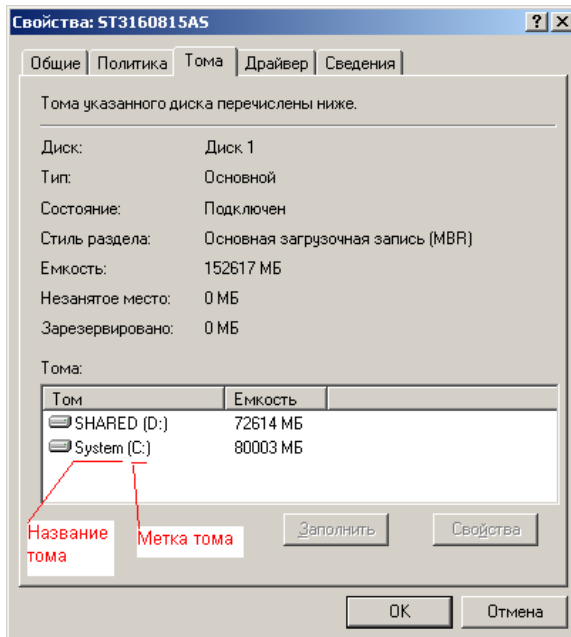
В появившемся окне найдите раскрывающееся меню **Дисковые устройства** (ищите иконку с жёстким диском). Раскройте меню, нажав на плюс. В раскрывшемся меню будут показаны все физически подключённые к компьютеру жесткие диски. Первые две буквы в названии винчестера кодируют название производителя. Остальные символы в зависимости от производителя каким-то образом кодируют в том числе и объём диска. Например, диск ST3160815AS, как и диск WD1600JB-00GVA0, имеет объём 160 ГБ.

Производители винчестеров (жёстких дисков): WDC – Western Digital, ST – Seagate, Samsung, Hitachi.

Вызовите окно **Свойства** через контекстное меню для одного из дисков и откройте вкладку **Тома**. Нажмите кнопку **Заполнить**.

Контекстное меню дискового устройства **Свойства** **Тома** **Заполнить**
Контекстное меню дискового устройства **Properties** **Volumes** **Populate**

В нижней части окна появится информация о виртуальных разделах - томах на физическом диске. Сопоставьте все физические жесткие диски всем виртуальным жестким дискам в папке **Мой компьютер**.



Приведите в отчёте полный список внешних накопителей для Вашего рабочего компьютера в лаборатории и для Вашего домашнего компьютера. Для каждого накопителя укажите принадлежность к физическому жесткому диску, общую ёмкость и процент свободного места. Результаты приведите в таблице по примеру.

Как Вы думаете, достаточно ли имеющегося свободного места на дисках для полноценной работы?

К компьютеру из примера на рисунке подключены следующие внешние накопители:

Физический накопитель	Название тома	Метка тома	Общая ёмкость, ГБ	Процент свободного места, %
ST3160815AS	System	C	78.1	$100 * 17.9 / 78.1 = 23$
ST3160815AS	SHARED	D	70.9	13
	...			

ЛАБОРАТОРНАЯ РАБОТА №6

Тема: «Организация интерфейсов в вычислительной технике»

Цель: знакомство с компонентной структурой современного персонального компьютера.

Оборудование: ПК с операционной системой Windows.

Порядок выполнения работы

Задание:

Подбор аппаратной конфигурации ПК по индивидуальному заданию.

Ход работы

Для подбора компонентов Вы можете воспользоваться сервисом Конфигуратор системного блока на сайте www.ulmart.ru (<http://www.ulmart.ru/configurator.php#configer>) или на сайте key.ru <http://key.ru/shop/devices/>.

Выберите конфигурацию по заданию. Проверьте согласованность параметров выбранных компонентов в строках, помеченных цветом. Обоснуйте выбор каждого компонента в поле для примечания в строке **Выбранная модель**. Приведите значения дополнительных характеристик для выбранных компонентов в остальных строках.

Дополните конфигурацию периферийным оборудованием по желанию - наушники, микрофон, принтер, сканер и пр. Подсчитайте суммарную стоимость выбранной комплектации.

Варианты

0. Intel Core i7, OEM, видео от NVIDIA
1. Intel Core i7, BOX, встроенное видео
2. Intel Core i5, OEM, видео от AMD
3. Intel Core i5, BOX, встроенное видео
4. AMD A10, BOX, видео от NVIDIA
5. AMD A8, BOX, видео от AMD
6. AMD A10, OEM, встроенное видео
7. AMD A8, OEM, видео от NVIDIA

Пример. Сборка для 0 варианта. Жирным цветом отмечены фиксированные заданием параметры. Цветом отмечены параметры, которые требуют согласования между компонентами.

Компонент	Характеристика	Значение	Примечание
Процессор	Производитель	Intel	По заданию
	Модель	Intel Core i7	
	Версия поставки	OEM - без кулера	
	Выбранная модель	Intel Core i7-3820 3.6/10Mb LGA2011	Средний ценовой диапазон
	Ссылка	http://key.ru/shop/devices/processors/intel_core_i7-3820_3_6_10mb_gla2011_box	
	Тип разъёма (Socket)	LGA2011	
	Частота собственная	3.6 ГГц	
	Число ядер	4	
	Кэш память (наличие и размер)	10 МБ	
	Мощность	130 Вт	
	Стоимость	10 190 р	
Кулер процессора	Тип разъёма (Socket)	LGA2011	Необходим, поскольку процессор поставляется без вентилятора
	Выбранная модель	CPU cooler Cooler Master Hyper 412 Slim	
	Ссылка	http://key.ru/shop/devices/computer_cooling/kulery_dlya_processorov/cpu_cooler_cooler_master_hyper_412_slim	

	Уровень шума	8 дБ	
	Стоимость	1 899	
Материнская плата	Тип разъёма (Socket)	LGA2011	
	Выбранная модель	MB Gigabyte GA-X79-UD3	самая недорогая модель без встроенного видео с достаточным запасом под оперативную память
	Ссылка	http://key.ru/shop/devices/motherboards/mb_gigabyte_ga-x79-ud3/	
	Встроенная видеокарта	нет	
	Интерфейс видеокарты	2 шт. PCI Express 3.0 x16	
	Количество слотов памяти	2	
	Тип модулей памяти	DDR3	
	Частота системной шины	2400/2133/1866/1600/1333 /1066 МГц	
	Форм-фактор	ATX	
	Разъёмы	Сетевая LAN-розетка RJ-45 2 порта USB 3.0/2.0 1 порт PS/2 для подключения клавиатуры и мыши 8 портов USB 2.0/1.1 6 аудио разъемов 1 x SPDIF out (коаксиальный) Оптический выход SPDIF-интерфейса	пришлось искать дополнительные характеристики от производителя не совпали с характеристиками на сайте магазина http://www.gigabyte.ru/products/page/mb/ga-x79-ud3rev_10/specs/
Стоимость	6890		
Видеокарта	Производитель	NVIDIA	По заданию
	Интерфейс	до 2 PCI Express 3.0 x16	
	Выбранная модель	2048M Asus GeForce GTX660 DDR5	Средний ценовой диапазон
	Ссылка	http://key.ru/shop/devices/videokarty/2048m_asus_geforce_gtx660_ddr5_2xdvi_hdmi_dp_pci-e/	
	Чипсет	GeForce GTX 660	

	Частота	1072 МГц	
	Объём памяти	2048 Мб	
	Видеовыходы	2xDVI HDMI DP	
	Мощность	150 Вт	информация с сайта производителя
	Стоимость	8090	
	Длина видеокарты	10 дюймов = 255 мм	с сайта производителя
Оперативная память	Тип модулей памяти	DDR3	
	Количество слотов памяти	2	
	Частота системной шины	2400/2133/1866/1600/1333 /1066 МГц	
	Выбранная модель	Модуль памяти DDR3 16Gb 2133MHz Kingston XMP Predator CL11 Kit of 2	Максимальный объём с учетом ограничения на количество слотов памяти
	Ссылка	http://key.ru/shop/devices/memory/ddr3_16gb_2133mhz_ki_ngstn_xmp_predator_non-ecc_cl11_kit_of_2/	
	Стоимость	5090	
Привод CD/DVD	нет		Нет необходимости
Жесткий диск HDD	Интерфейс	SATA	Современный стандарт дефакто, должен совпадать с интерфейсом на материнской плате чем быстрее - тем лучше
	Выбранная модель	WD4001FAEX	Самый дешевый из самых больших
	Ссылка	http://key.ru/shop/devices/ustrojstva_hraneniya_i_chteniya_dannyh/hdd_ssd/vinchester_4tb_wd_caviar_black_wd4001faex/	
	Объём	4 ТБ	
	Стоимость	9290	
Корпус	Минимальная требуемая мощность	300 Вт	сумма мощности процессора и видеокарты

Форм фактор	АТХ	или совместимый
Длина видео-карты	10 дюймов = 255 мм	
Выбранная модель	Codegen Q3339-A2 Black АТХ	По внешнему виду
Наличие блока питания	да	
Ссылка	http://key.ru/shop/devices/korpusa/korpus_codegen_q3339-a2_black_atx/	
Стоимость	1490	

ЛАБОРАТОРНАЯ РАБОТА №7

Тема: «Способы адресации»

Цель: изучить архитектуру и принципы построения микроконтроллера AVR ATMEGA128.

Оборудование: справочный материал, персональный компьютер с выходом в Интернет.

Справочный материал

Основные характеристики микроконтроллера AVR ATMEGA128.

AVR-архитектура объединяет высокопроизводительный RISC-процессор с отдельным доступом к памяти программ и данных, 32 регистра общего назначения, каждый из которых может работать как регистр-аккумулятор, и развитую систему команд с фиксированной (16-бит) длиной. Конвейерная архитектура с одновременным исполнением текущей и выборкой следующей команды позволяет выполнять большинство команд за один машинный цикл, что обеспечивает производительность до 1 MIPS на каждый МГц тактовой частоты.

Ниже приводятся основные характеристики микроконтроллера AVR ATMEGA128:

производство по КМОП-технологии с низким энергопотреблением;
тактовая частота может изменяться в широких пределах от 0 до 16 МГц (полностью статическая архитектура);

ядро микроконтроллера основано на RISC архитектуре с двухступенчатым конвейером, обеспечивающим выполнение одной команды за один машинный цикл;

гарвардская архитектура с отдельной памятью программ и данных;

регистровый файл содержит 32 регистра общего назначения;

все регистры общего назначения непосредственно подключены к АЛУ;

совмещенная архитектура ввода/вывода (регистры общего назначения и порты ввода/вывода находятся в адресном пространстве ОЗУ данных);

наличие программного стека;

наличие в составе АЛУ аппаратного умножителя;

19 источников внутренних прерываний, 8 источников внешних прерываний;

Объем FLASH-памяти программ: 128 кБт;

Объем статической оперативной памяти (ОЗУ) : 4 кБт

Объем памяти данных на основе электрически-стираемого ПЗУ (EEPROM): 4 кБт;

Интерфейсы программирования: SPI и JTAG;

Напряжение питания: 4.5–5.5 В;

Периферийные устройства:

8-разрядные параллельные порты ввода/вывода;

8 и 16 разрядные таймеры-счётчики;

широотно-импульсные модуляторы;

аналоговые компараторы,

10–разрядный 8–канальный АЦП,

Встроенный универсальный асинхронный приемопередатчик (USART).

Высокая производительность, наличие развитой подсистемы ввода/вывода и широкого спектра встроенных периферийных устройств позволяют отнести микроконтроллеры AVR ATMEGA128 к классу наиболее функциональных микроконтроллеров для встроенных систем управления, применяемых в бытовой и офисной технике, мобильных телефонах, контроллерах периферийного оборудования (принтеры, сканеры, приводы CD-ROM), портативных медицинских приборах, интеллектуальных датчиках (охранных, пожарных) и др.

ATmega128 – маломощный 8-разр. КМОП микроконтроллер, основанный на расширенной AVR RISC-архитектуре. За счет выполнения большинства инструкций за один машинный цикл ATmega128 достигает производительности 1 млн. операций в секунду/МГц, что позволяет проектировщикам систем оптимизировать соотношение энергопотребления и быстродействия.

Функциональная схема

Ядро AVR сочетает богатый набор инструкций с 32 универсальными рабочими регистрами. Все 32 регистра непосредственно подключены к арифметико-логическому устройству (АЛУ), который позволяет указать два различных регистра в одной инструкции и выполнить ее за один цикл. Данная архитектура обладает большей эффективностью кода за счет достижения производительности в 10 раз выше по сравнению с обычными CISC-микроконтроллерами.

ATmega128 содержит следующие элементы: 128 кбайт внутрисистемно программируемой флэш-памяти с поддержкой чтения во время записи, 4 кбайт ЭСППЗУ, 4 кбайт статического ОЗУ, 53 линии универсального ввода-вывода, 32 универсальных рабочих регистра, счетчик реального времени (RTC), четыре гибких таймера-счетчика с режимами сравнения и ШИМ, 2 УСАПП, двухпроводной последовательный интерфейс ориентированный на передачу байт, 8-канальный 10-разр. АЦП с опциональным дифференциальным входом с программируемым коэффициентом усиления, программируемый сторожевой таймер с внутренним генератором, последовательный порт SPI, испытательный интерфейс JTAG совместимый со стандартом IEEE 1149.1, который также используется для доступа к встроенной системе отладки и для программирования, а также шесть программно выбираемых режимов уменьшения мощности.

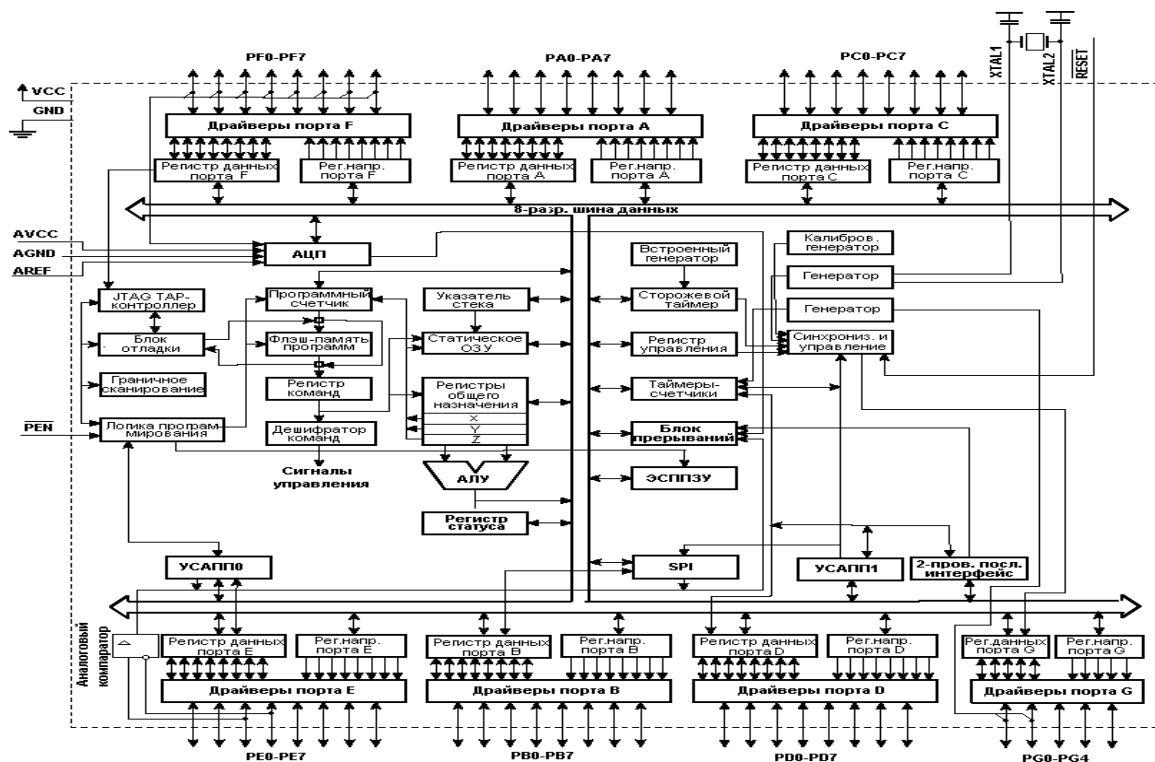


Рисунок – Функциональная схема

Режим холостого хода (Idle) останавливает ЦПУ, но при этом поддерживая работу статического ОЗУ, таймеров-счетчиков, SPI-порта и системы прерываний. Режим выключения (Powerdown) позволяет сохранить содержимое регистров, при остановленном генераторе и выключении встроенных функций до следующего прерывания или аппаратного сброса. В экономичном режиме (Power-save) асинхронный таймер продолжает работу, позволяя пользователю сохранить функцию счета времени в то время, когда остальная часть контроллера находится в состоянии сна. Режим снижения шумов АЦП (ADC Noise Reduction) останавливает ЦПУ и все модули ввода-вывода, кроме асинхронного таймера и АЦП для минимизации импульсных шумов в процессе преобразования АЦП. В дежурном режиме (Standby) кварцевый/резонаторный генератор продолжают работу, а остальная часть микроконтроллера находится в режиме сна. Данный режим характеризуется малой потребляемой мощностью, но при этом позволяет достичь самого быстрого возврата в рабочий режим. В расширенном дежурном режиме (Extended Standby) основной генератор и асинхронный таймер продолжают работать.

Микроконтроллер производится по технологии высокоплотной энергонезависимой памяти компании Atmel. Встроенная внутрисистемно программируемая флэш-память позволяет перепрограммировать память программ непосредственно внутри системы через последовательный интерфейс SPI с помощью простого программатора или с помощью автономной программы в загрузочном секторе. Загрузочная программа может использовать любой интерфейс для загрузки прикладной программы во флэш-память. Программа в загрузочном секторе продолжает работу в процессе обновления прикладной секции флэш-памяти, тем самым поддерживая

двухоперационность: чтение во время записи. За счет сочетания 8-разр. RISC ЦПУ с внутрисистемно самопрограммируемой флэш-памятью в одной микросхеме ATmega128 является мощным микроконтроллером, позволяющим достичь высокой степени гибкости и эффективной стоимости при проектировании большинства приложений встроенного управления.

ATmega128 поддерживается полным набором программных и аппаратных средств для проектирования, в т.ч.: Си-компиляторы, макроассемблеры, программные отладчики/симуляторы, внутрисистемные эмуляторы и оценочные наборы.

Программная модель микроконтроллера avr mega128. Механизм работы с регистрами, памятью и портами ввода/вывода.

В микроконтроллере AVR ATMEGA128 реализована гарвардская архитектура, в соответствие с которой адресные пространства памяти программ и данных физически разделены (доступ к этим областям памяти осуществляется по отдельным шинам). Такая организация позволяет ядру процессора одновременно работать с памятью программ и данных, что повышает быстродействие. Карта распределения памяти в микроконтроллере AVR ATMEGA128 приведена на рисунке 1.1. Память программ представляет собой электрически стираемое перепрограммируемое постоянное запоминающее устройство ППЗУ объемом 128 кБт, выполненное по технологии FLASH – памяти, и предназначена для хранения команд, управляющих функционированием микроконтроллера, а также для хранения констант, не меняющих своих значений в ходе выполнения программы. Так, как длина команды составляет 16 бит, то память программ имеет 16-разрядную организацию. Для адресации памяти программ используется 16-разрядный регистр – программный счетчик РС (Program Counter). Программа исполняется последовательно. Для управления ходом выполнения программы существуют команды перехода, изменяющие соответствующим образом значение РС.

Память данных организована по принципу совмещенной архитектуры ввода/вывода и разделена на 3 части: регистровая память, память портов (регистров) ввода/вывода и статическое ОЗУ (SRAM), расположенные в едином адресном пространстве.

Рисунок – Распределение памяти в микроконтроллере AVR ATMEGA128

Регистровая память (см. рисунок 1.2) включает 32 8-разрядных регистра общего назначения (R0 - R31), объединенных в регистровый файл. Каждый из регистров общего назначения непосредственно связан с АЛУ. АЛУ поддерживает арифметические и логические операции с регистрами, между регистром и константой или непосредственно с регистром. При исполнении арифметической или логической команды за один такт из регистрового файла выбираются два операнда, выполняется действие, и результат возвращается в регистровый файл. Регистровый файл отображается на младшие 32 адреса 0000h-001Fh памяти данных и к его регистрам можно обращаться как к ячейкам памяти. Шесть 8 - разрядных регистров (R26 - R31) могут использоваться как три 16-разрядных регистра-указателя для косвенной адресации (см. рисунок).

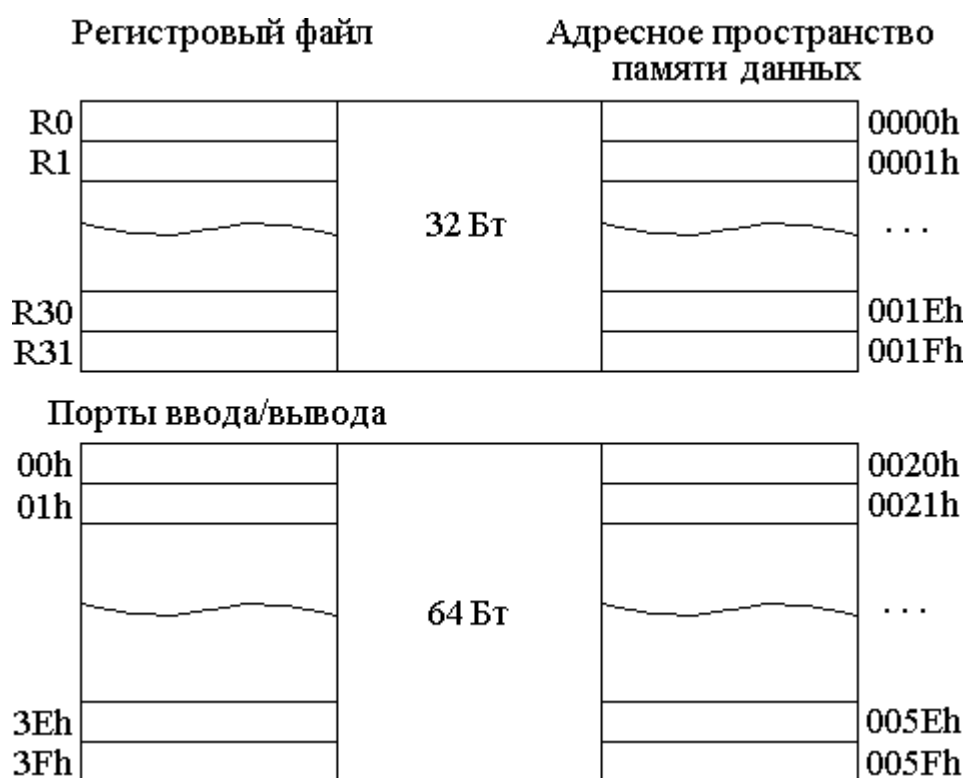


Рисунок – Иллюстрация отображения регистров общего назначения и портов ввода/вывода на адресное пространство памяти данных

X		Y		Z	
27	26	29	28	31	30

Рисунок – 16-разрядные регистры X, Y, Z, используемые для косвенной адресации памяти.

Пространство ввода/вывода состоит из 64 адресов портов 0000h-003Fh, предназначенных для взаимодействия с внутренними и внешними устройствами по отношению к микроконтроллеру. Порты ввода/вывода отображаются на область памяти данных с адресами 0020h-005Fh и допускают возможность обращения к ним как к ячейкам памяти. При доступе к порту ввода/вывода как к ячейке памяти к адресу порта необходимо добавить 20h. Адрес порта ввода/вывода в пространстве памяти часто указывается в скобках после адреса в пространстве портов ввода/вывода. Ввиду того, что основной функцией микроконтроллера является управление внешними устройствами, в таблице 1.1. приводятся названия и адреса (в пространстве портов ввода/вывода) основных интерфейсных портов с указанием режима работы и функций отдельных регистров. Для реализации операций ввода вывода для каждого порта выделяются три регистра PIN*(для ввода данных), PORT* (для вывода данных) и DDR*(для настройки порта).

По адресам памяти 0060h-00FFh расположены 160 дополнительных регистров ввода/вывода.

Непосредственно память данных представляет собой статическое ОЗУ (SRAM) объемом 4 кБт, занимающее диапазон адресов 0100h-10FFh.

Таблица – Порты ввода/вывода микроконтроллера AVR MEGA128 для подключения внешних устройств

Название порта ввода/вывода	Идентификаторы отдельных регистров	Адрес	Режим / функция
PORTA	PINA	19h	IN
	DDRA	1Ah	OUT / DIRECTION
	PORTA	1Bh	OUT
PORTB	PINB	16h	IN
	DDRB	17h	OUT / DIRECTION
	PORTB	18h	OUT
PORTC	PINC	13h	IN
	DDRC	14h	OUT / DIRECTION
	PORTC	15h	OUT
PORTD	PIND	10h	IN
	DDRD	11h	OUT / DIRECTION
	PORTD	12h	OUT
PORTE	PINE	01h	IN
	DDRE	02h	OUT / DIRECTION
	PORTE	03h	OUT
PORTF	PINF	00h	IN
	DDRF	61h	OUT / DIRECTION
	PORTF	62h	OUT
PORTG	PING	63h	IN
	DDRG	64h	OUT / DIRECTION
	PORTG	65h	OUT

Регистр состояния SREG расположен в области ввода/вывода по адресу 3Fh (5Fh) и содержит информацию о текущем состоянии микроконтроллера. Расположение флаговых битов регистра состояния приведено на рисунке.

№ бита	7	6	5	4	3	2	1	0
3Fh (5Fh)	I	T	H	S	V	N	Z	C

Рисунок – Регистр состояния SREG.

Назначение отдельных битов регистра состояния приведено ниже:

- **C** – флаг переноса, устанавливается в 1 при наличии переноса в арифметических операциях;
- **Z** – флаг нуля, устанавливается в 1, если результат операции равен 0;
- **N** – флаг отрицательного результата, устанавливается в 1 при получении отрицательного результата;
- **V** – флаг переполнения, фиксирует выход результата за пределы допустимого диапазона значений;
- **S** – флаг знака, $S = N \text{ xor } V$;
- **H** – флаг дополнительного переноса (из младшей тетрады байта в старшую);
- **T** – флаг для временного хранения бита из регистров общего назначения;
- **I** – управляющий флаг разрешения прерываний, разрешает (1) или запрещает (0) процессору реагировать на аппаратные прерывания.

Система команд микроконтроллера avr mega128.

- Базовый набор команд языка ASSEMBLER для микроконтроллеров AVR содержит 120 инструкций, которые можно разделить на 4 группы: команды пересылки данных; арифметические и логические команды; инструкции для работы с битами; команды управления ходом исполнения программы.
- **Команды пересылки данных.** Группа команд пересылки данных включает в себя инструкции по загрузке значений констант, пересылки данных типа регистр – регистр, регистр – память, регистр – порт ввода/вывода. Команды данной группы являются двух-операндными, причем первым операндом является приемник данных, а вторым – источник данных.
- Команда загрузки констант **ldi R, K** применяется для записи непосредственного значения K в регистр – приемник R. В качестве регистра – приемника могут использоваться регистры общего назначения R16 – R31. Если константа представлена в двоичной или шестнадцатеричной системах счисления, то перед значением константы K необходимо указать спецификатор системы счисления 0b – для двоичной, 0x – для шестнадцатеричной соответственно. Примеры:
 - **ldi R16, 125** загрузка в R16 десятичного числа 125;
 - **ldi R20, 0xFF** загрузка в R20 шестнадцатеричной константы FFh;
 - **ldi R23, 0b11011001** загрузка в R23 двоичной константы 11011001;
- Команда пересылки данных между регистрами **mov Rd, Rs** используется для пересылки значения из регистра-источника Rs в регистр-приемник

Rd. Операнды в команде являются исключительно регистрами общего назначения R0 – R31.

- Примеры:
- **mov R16, R0** загрузка в R16 значения из регистра R0;
- **mov R17, R20** загрузка в R17 значения из регистра R20;
- В командах пересылки данных между регистром и ячейкой памяти используется механизм косвенной адресации, при котором адрес ячейки памяти заносится в один из 16-разрядных регистров X,Y,Z (см. рисунок 1.3). Форматы команд:
- **ld R₈, (R₁₆)** – загрузка данных из ячейки памяти, адрес которой находится в 16-разрядном регистре R₁₆, в регистр общего назначения R₈
- **st (R₁₆), R₈** – загрузка данных из регистра общего назначения R₈ в ячейку памяти, адрес которой находится в 16-разрядном регистре R₁₆,
- **ldd R₈, (R₁₆+Q)** – загрузка данных в регистр общего назначения R₈ из ячейки памяти, адрес которой находится как сумма значения, находящегося в 16-разрядном регистре R₁₆, и смещения Q,.
- **std (R₁₆+Q), R₈** – загрузка данных из регистра общего назначения R₈ в ячейку памяти, адрес которой находится как сумма значения, находящегося в 16-разрядном регистре R₁₆, и смещения Q,.
- **ld R₈, (R₁₆)** – загрузка в регистр общего назначения R₈ данных из ячейки памяти, адрес которой находится в 16-разрядном регистре R₁₆;
- Примеры:
- **ld R2, X** загрузка в R2 значения из памяти по адресу, указанному в X;
- **st Y, R5** загрузка значения из регистра R5 в память по адресу, указанному в Y.
- **ldd R5, Z+1** загрузка в R5 байта из памяти по адресу Z+1;
- **std Y+4, R7** загрузка байта из регистра R7 в память по адресу Y+4.
- Для обращения к портам ввода/вывода в микропроцессоре предусмотрены специальные команды **in** и **out**:
- **in R, P** ввод данных из порта с адресом P в регистр общего назначения R;
- **out P, R** вывод данных из регистра общего назначения R в порт с адресом P;
- Примеры:
- **in R10, 0x15** ввод данных из порта с адресом 15h в регистр общего назначения R10;
- **out 0x2F, R8** вывод данных из регистра общего назначения R8 в порт с адресом 2Fh;
- **Арифметические и логические команды.** Для работы с целыми двоичными числами целочисленное АЛУ микроконтроллера AVR MEGA128 поддерживает более десятка арифметических и логических команд.
- Основными арифметическими командами являются инструкции сложения, вычитания и умножения. Операндами в командах данной группы могут быть только регистры общего назначения. Результат операции (кроме умножения) записывается по адресу первого операнда.

0	0	0	0	0	0	0	0	0
0	1	1	0	1	0	0	1	1
1	0	1	1	0	0	1	0	1
1	1	1	1	1	1	1	1	0

- Указанные команды используются для выполнения операций поразрядного маскирования: **or** – для установки единиц в заданных разрядах, **and** – для установки нулей, **eor** – для выяснения совпадений значений битов первого операнда с маской. Команды изменяют флаги нуля, **Z**, знака **N** и переполнения **V**. Примеры поразрядных логических операций, иллюстрирующие применение механизма маскирования битов, приводятся в таблице.

Таблица – Примеры поразрядных логических операций

Пример поразрядного маскирования or		Пример поразрядного маскирования and		Пример поразрядного маскирования eor	
Rd	xxxxxxx	Rd	xxxxxxx	Rd	10100110
Rs	00010010	Rs	10110101	Rs	00010010
Rd=Rd or Rs	xxx1xx1x	Rd=Rd and Rs	x0xx0x0x	Rd=Rd eor Rs	10110100

- Команда поразрядного инвертирования:
- com R** логическое отрицание; действие: $R = 0b11111111 - R$,
- выполняет изменение значений двоичных разрядов операнда (регистр общего назначения) на противоположные.
- Пример:
- com R3** действие: $R3 = 0b11111111 - R3$.
- Полный перечень арифметических и логических команд микроконтроллера AVR MEGA128 приводится в Приложении 2.
- Команды для работы с битами.** Дополняют совокупность логических операций команды сброса, установки и проверки значений отдельных битов.
- Команды сброса **cbi P, n** и установки **sbi P, n** битов предназначены для присваивания значений 0 и 1 отдельным битам портов ввода/вывода соответственно. Первым операндом в этих командах является адрес порта ввода/вывода, вторым – номер бита (от 0 до 7).
- Примеры:
- cbi 0x17, 5** действие: $0x17_5 = 0$;
- sbi 0x40, 1** действие: $0x40_1 = 1$.
- Команда логического сдвига **lsl R** осуществляет сдвиг влево на одну позицию всех битов операнда, а в младший разряд добавляется ноль. Старший бит операнда поступает в флаг переноса **C**. В качестве операнда могут использоваться только регистры общего назначения. Команда **lsr R** выполняет сдвиг вправо на одну позицию всех битов операнда, а в старший разряд добавляется ноль. Младший бит операнда поступает в флаг переноса **C**. Механизм работы и

синтаксис аналогичен команде **lsl**. Примеры использования команд логического сдвига:

lsl R17	выполнить логический сдвиг влево всех разрядов в R17;
lsr R9	выполнить логический сдвиг вправо всех разрядов в R9.

- Поменять местами младшую и старшую тетрады байта, загруженного в регистр общего назначения, можно с помощью команды **swap R**. Следующий фрагмент иллюстрирует действие команды **swap**:
- **ldi R19, 0b01001101** Загрузить константу 0b01001101 в регистр R19;
- **swap R19** В результате исполнения команды **swap** в регистре R19 будет сохранено значение 0b11010100.
- Дополняют перечень команд для работы с битами инструкции для сброса/установки значений флаговых разрядов в регистре статуса SREG, описание которых приводится в Приложении 2.
- **Команды сравнения, условного и безусловного перехода.** Команда сравнения **cp Rd, Rs** – осуществляет действие **Rd–Rs** и устанавливает флаги нуля **Z**, отрицательного результата **N**, переполнения **V**, переноса **C** и дополнительного переноса **H**. Результат не сохраняется по адресу первого операнда, а только формируются флаги. Операндами могут быть только регистры общего назначения.
- Команды условного перехода вызываются сразу после команд сравнения (или других инструкций, вызывающих изменения битов регистра состояния **SREG**) и на основе анализа флагов осуществляют переход по указанному адресу (метке) в памяти команд.
- Наиболее распространенными среди команд этой группы являются:
- **breq M** переход на M, если равно;
- **brne M** переход на M, если неравно;
- **brlo M** переход на M, если меньше;
- **brsh M** переход на M, если больше или равно.
- Пример совместного использования команд сравнения и условного перехода:
- **cp R1, R5** сравнить значения в регистрах R1 и R5;
- **breq lbl1** выполнить переход на метку lbl1, если значения в регистрах R1 и R5 равны ($R1-R5=0$).
- Команда **rjmp M** осуществляют безусловный переход по указанному 8-разрядному адресу (метке, label) в памяти команд. Пример:
- **rjmp lbl2** безусловный переход на метку lbl2.
- Команда **jmp M** осуществляют безусловный переход по указанному 16-разрядному адресу (метке, label) в памяти команд. Пример:
- **rjmp lbl3** безусловный переход на метку lbl3.
- Полный перечень команд сравнения и перехода приводится в таблицах

Порядок выполнения работы

Занятие 1.

Проработайте теоретические материалы и конспекты лекций, ознакомьтесь со структурой и принципами функционирования микроконтроллера AVR ATMEGA128.

Задание 1. Ответьте на контрольные вопросы с №1 - №36 (согласно заданному варианту).

Вариант выбирается по последней цифре номера строки в журнале группы.

Набор контрольных вопросов для отчета выбирается согласно варианту из таблицы.

№ Вар.	0	1	2	3	4	5	6	7	8	9
Вопрос 1	1	2	3	4	5	6	7	8	9	10
Вопрос 2	10	9	8	7	6	4	5	2	3	1
Вопрос 4	11	12	13	14	15	16	17	18	19	20
Вопрос 5	19	20	11	12	13	14	15	16	17	18
Вопрос 6	21	22	23	24	25	26	27	28	29	30
Вопрос 7	31	32	33	34	35	36	21	22	23	24
Вопрос 8	25	26	27	28	29	30	31	32	33	34

ЛАБОРАТОРНАЯ РАБОТА №8

Тема: «Способы адресации»

Цель: изучить архитектуру и принципы построения микроконтроллера AVR ATMEGA128.

Оборудование: справочный материал, персональный компьютер с выходом в Интернет.

Порядок выполнения работы

Занятие 2.

Перед началом выполнения практической части лабораторной работы проводится экспресс–контроль знаний по принципам функционирования микроконтроллера AVR ATMEGA 128. При подготовке к занятию 2 необходимо составить предварительный вариант листинга программы, в соответствие с индивидуальным заданием (см. таблицу).

Задание 2. Разработать в среде программирования Code Vision AVR программу на языке ASSEMBLER для микроконтроллера AVR ATMEGA 128, выполняющую сложение двух однобайтных чисел.

Порядок выполнения задания:

1. Включить лабораторный макет (установить выключатель электропитания в положение I, и убедиться в свечении индикатора электропитания красным цветом).
2. Запустить компилятор Code Vision AVR.
3. Создать пустой проект.
4. Создать файл ресурса для кода программы и подключить его к проекту.
5. Ввести код исходного модуля программы управления светодиодами в соответствии с вариантом задания, указанным в таблице.
6. Выполнить компиляцию (нажав клавишу **F9**) исходного модуля программы и устранить ошибки, полученные на данном этапе.
7. Настроить параметры программатора.
8. Создать загрузочный модуль программы (нажав комбинацию клавиш **Shift+F9**) и выполнить программирование микроконтроллера.
9. Проверить работоспособность загруженной в микроконтроллер программы и показать результаты работы преподавателю.
10. В случае некорректной работы разработанной программы, выполнить аппаратный сброс микроконтроллера, провести отладку исходного модуля программы и заново проверить функционирование программы, повторив выполнение пункта 9.

Пример выполнения задания

Разработать программу, выполняющую сложение двух однобайтных чисел. Эта программа будет реализовать функцию:

$$f(a,b) = a + b$$

Допустим, что первое слагаемое будет находиться в памяти с адресом указанным в регистре **X**, а второе слагаемое будет располагаться в регистре **R16**. Результат вычислений должен находиться в памяти с адресом указанным в регистре **Y**. Для правильного выполнения программы необходимо выполнить начальную инициализацию слагаемых.

Так как, первое слагаемое находится в памяти с адресом указанным в регистре **X**, выполним инициализацию его. Допустим значение первого слагаемого будет число 125 (0x7Dh) Для этого в программе необходимо внести такой код:

ldi R20, 125; промежуточное внесение числа в регистр R20.

st X, R20; перенесение значения из регистра R20 в область памяти, где располагается первое слагаемое (адрес хранится в двухбайтовом регистре **X**).

Для инициализации второго слагаемого, которое будет находиться в регистре **R16**, используем число 60 (0x3C)

ldi R16, 60

Теперь оба слагаемые имеют значения (125 и 60). Для выполнения сложения двух чисел использует команду **add**. Специфика этой команды в том, что команда может оперировать только с регистрами общего назначения.

Для выполнения сложения необходимо перенести первое слагаемое из памяти, например в регистр **R17**. Операция, а также операция сложения показана в следующем листинге.

ld R17, X

add R16, R17

Результат выполнения команды сложения **add** находится в регистре **R16**. Переноса результата в память, согласно заданию, реализуется такой командой:

st Y, R16

Полный листинг программы сложения двух чисел, показан ниже:

ldi R20, 125.

st X, R20

ldi R16, 60

ld R17, X

add R16, R17

st Y, R16

Задание 3. Разработать в среде программирования Code Vision AVR программу на языке ASSEMBLER для микроконтроллера AVR ATMEGA 128, выполняющую сложение двух двухбайтных чисел. Варианты выполнения задания находятся в таблице.

Таблица – Таблица значений слагаемых

	0	1	2	3	4	5	6	7	8	9
Первое слагаемое	0CFE	565A	AD32	FE63	3CB6	3271	5CA6	78DF	FC17	1978
Второе слагаемое	2A12	AACD	77FF	5657	5287	9FE3	36DE	2317	6E8C	0912

В таблице значения представлены в шестнадцатеричной системе. Номер варианта выбирается, как предпоследняя цифра номера зачетной книжки.

Таблица – Таблица выбора размещения операндов

	0	1	2	3	4	5	6	7	8	9
Первое слагаемое	1	1	1	2	2	2	2	1	1	2
Второе слагаемое	1	2	2	2	2	1	2	2	2	1
Результат	1	1	2	2	1	1	1	1	2	1



Рисунок – Блок-схема алгоритм сложения двух двухбайтных чисел

В таблице значения соответствуют расположению операндов:

- 1 – регистры;
- 2 – память.

Номер варианта выбирается, как последняя цифра номера зачетной книжки.

3. Содержание отчета

Отчет должен содержать:

- Титульный лист;
- Название;
- Цель работы;
- Ответы на контрольные вопросы (согласно заданному варианту).
- Алгоритм работы программы согласно индивидуальному заданию

- Листинг программы согласно индивидуальному заданию

Контрольные вопросы

1. Какая архитектура лежит в основе данных микропроцессоров фирмы AVR?
2. Какая тактовая частота может устанавливаться для микропроцессоров?
3. Сколько каналов АЦП насчитывает микропроцессор?
4. Сколько разрядов в одном канале АЦП?
5. Количество регистров общего назначения?
6. Что такое АЛУ?
7. Объем статической памяти?
8. Объем Flash – памяти?
9. Объем электрически стираемой памяти?
10. Количество источников внутреннего прерывания?
11. Количество источников внешнего прерывания?
12. Какие интерфейсы программирования существуют у данного типа микропроцессоров?
13. Какие периферийные устройства входят в состав микроконтроллера?
14. В чем заключается гарвардская структура микроконтроллера?
15. Что такое программный счетчик?
16. Какие режимы уменьшения мощности существуют для микропроцессора?
17. Какие возможности предоставляет режим холостого хода (Idle)?
18. Какие возможности предоставляет режим выключения (Powerdown)?
19. Какие возможности предоставляет экономичный режим (Power-save)?
20. Какие возможности предоставляет дежурный режим (Standby)?
21. Какие возможности предоставляет режим снижения шумов АЦП (ADC Noise Reduction)?
22. Какие возможности предоставляет расширенный дежурный режим (Extended Standby)?
23. Какой интерфейс позволяет запрограммировать встроенную Flash-память?
24. Назначение регистра SREG
25. Назначение флага C регистра SREG
26. Назначение флага Z регистра SREG
27. Назначение флага N регистра SREG
28. Назначение флага V регистра SREG
29. Назначение флага S регистра SREG
30. Назначение флага H регистра SREG
31. Назначение флага T регистра SREG
32. Назначение флага I регистра SREG
33. Что такое порт ввода вывода?
34. Назначение регистров PIN*.
35. Назначение регистров DDR*.
36. Назначение регистров PORT*.
37. Что такое система команд?

38. Какие виды команд существуют?
39. Какие команды относятся к командам пересылки данных и их назначение?
40. Какие команды относятся к арифметическим командам и их назначение?
41. Какие команды относятся к логическим командам пересылки данных и их назначение?
42. Какие команды относятся к инструкциям для работы с битами и их назначение?
43. Какие команды относятся к командам управления ходом исполнения программы и их назначение?

Информационное обеспечение обучения

Основные литература:

1. Замятина, О. М. Инфокоммуникационные системы и сети. Основы моделирования: учебное пособие для среднего профессионального образования / О. М. Замятина. — Москва: Издательство Юрайт, 2023. — 167 с. — (Профессиональное образование). — ISBN 978-5-534-17558-5. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru>
2. Гаврилов, М. В. Информатика и информационные технологии : учебник для среднего профессионального образования / М. В. Гаврилов, В. А. Климов. — 5-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2023. — 355 с. — (Профессиональное образование). — ISBN 978-5-534-15930-1. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/>

Дополнительные источники:

Казанский, А. А. Программирование на Visual C# : учебное пособие для среднего профессионального образования / А. А. Казанский. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2023. — 192 с. — (Профессиональное образование). — ISBN 978-5-534-14130-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/>

Учебно-методические материалы:

1. Методические указания к практическим/лабораторным работам (Электронный ресурс)/ Коровин Ю.И., Горохов Д.В., – Москва: РГАУ-МСХА, 2021 – ЭБС – «РГАУ-МСХА»

Интернет – ресурсы

1. Электронно-библиотечная система РГАУ-МСХА им. К.А. Тимирязева (далее ЭБС) сайт www.library.timacad.ru
2. Научная электронная библиотека «КиберЛенинка» - <https://cyberleninka.ru/>
3. Сетевая электронная библиотека аграрных вузов - <https://e.lanbook.com/books>