Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Саратовский государственный технический университет имени Гагарина Ю.А.»

Филиал федерального государственного бюджетного образовательного
учреждения высшего образования
«Саратовский государственный технический университет имени Гагарина Ю.А.»
в г. Петровске

УТВЕРЖДАЮ
Директор филиала СГТУ
имени Гагарина Ю.А. в г.Петровске
Е.А.Бесшапошникова
«30» июня 2021 г.

МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ ЛАБОРАТОРНЫХ РАБОТ

по дисциплине

ОП.11 «Вычислительная техника»

специальности
13.02.07 «Электроснабжение (по отраслям)»

Методические указания рассмотрены
на заседании предметной (цикловой)
комиссии общепрофессиональных дисциплин,
профессиональных модулей специальностей
технического профиля
«14» июня 2021 года, протокол № 13
Председатель ПЦК /Т.А.Лескина/

Петровск 2021

Пояснительная записка

Методические указания по выполнению лабораторных работ разработаны в соответствии с рабочей программой учебной дисциплины «Вычислительная техника», требованиями Федерального государственного образовательного стандарта (далее – ФГОС) по специальности среднего профессионального образования (далее – СПО) 13.02.07 Электроснабжение (по отраслям), утвержденного приказом Министерства образования и науки РФ от 14.12.2017 № 1216 и соответствующих общих (ОК) и профессиональных (ПК) компетенций:

ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам;

ОК 02. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности;

ОК 04. Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами;

ОК 05.Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;

ОК 09. Использовать информационные технологии в профессиональной деятельности;

ОК 10. Пользоваться профессиональной документацией на государственном и иностранном языках;

ПК 1.1. Выполнять основные виды работ по проектированию электроснабжения электротехнического и электротехнологического оборудования;

ПК 1.2. Читать и составлять электрические схемы электроснабжения электротехнического и электротехнлогического оборудования.

ПК 2.5. Разрабатывать и оформлять технологическую и отчетную документацию.

Цели изучения дисциплины «Вычислительная техника»:

– формирование профессиональных компетенций;

– использование полученных знаний, умений и навыков применения вычислительной техники и сетей в процессе изучения профессиональных дисциплин и в дальнейшей профессиональной деятельности;

– ознакомление с основами современных компьютерных коммуникаций и тенденциями их развития.

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

- составлять и минимизировать комбинационные схемы на основе базовых логических элементов;
- использовать типовые средства вычислительной техники и программного обеспечения.

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

- способы представления информации в ЭВМ;
- основные логические элементы цифровой техники;
- составные части, общие принципы организации и функционирования компьютерных систем;
- архитектуру процессоров;
- принцип работы основной памяти и периферийных устройств.

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 ее окончании – обсуждение итогов.

Комплект методических указаний по выполнению лабораторных работ дисциплины «Вычислительная техника», содержит 5 лабораторных занятий.

Перечень лабораторных работ.

ЛАБОРАТОРНАЯ РАБОТА№1

Тема: Система команд микропроцессора

ЛАБОРАТОРНАЯ РАБОТА№2

Тема: Понятия о состоянии процессора.

ЛАБОРАТОРНАЯ РАБОТА№3

Тема: Микроконтроллеры.

ЛАБОРАТОРНАЯ РАБОТА№4

Тема: Программирование на машинном языке.

ЛАБОРАТОРНАЯ РАБОТА№5

Тема: Программирование на языках высокого уровня

ИНСТРУКЦИИ ДЛЯОБУЧАЮЩИХСЯ ПО ВЫПОЛНЕНИЮ ЛАБОРАТОРЫХ РАБОТ

Прежде чем приступить к выполнению заданий, внимательно прочитайте данные рекомендации. Лабораторных работы включают в себя задания следующих видов:

Выполнение тестовых заданий

Для проверки и последующего анализа своих знаний Вам предлагается пройти тестовые задания. Выбор заданий осуществляется тестирующей системой случайным образом.

Тестовые задания интерактивны. По структуре формирования ответа различают следующие типы заданий:

- тесты востановления соответствия - предусматривают востановление соответствия между одинаковыми по величине, но различными по записи числами.
- тесты востановления порядока - предусматривают расстановку чисел в соответствие с указанным порядком.
- тесты единственного выбора - предусматривают выбор одного правильного ответа из нескольких предложенных вариантов,
- тесты открытого типа - предусматривают ввод текстовых данных.

При вводе ответа необходимо соблюдать следующие правила:

- курсор нужно поместить в окно для ввода,
- вписывать слова нужно без сокращения,
- вписывать числовые выражения нужно без пробелов, строго следуя образцу, приведенному в задании.

Несоблюдение правил выполнения тестов открытого типа приведет к обозначению ответа как неверного.

Перед выполнением задания внимательно прочитайте его формулировку и предлагаемые варианты ответа. Отвечайте только после того, как Вы поняли вопрос и проанализировали все варианты ответа.

Выполняйте задания в том порядке, в котором они представлены в тесте. Выбор правильных ответов осуществляется путем выбора правильных ответов из списка.

Тестовые задания оцениваются в баллах. Все вопросы имеют свое балльное значение, что определяется, в первую очередь, сложностью самого вопроса.

Постарайтесь выполнить как можно больше заданий и набрать наибольшее количество баллов. По завершении тестирования баллы суммируются.

После выполнения тестовых заданий обязательно сохраните Ваши ответы и предоставьте их учителю.

Создание презентаций

ПРАВИЛА ПОСТРОЕНИЯ СОДЕРЖАНИЯ.

Правило 1. Содержание должно быть структурировано.

Содержание презентации должно быть четко структурировано: каждый новый слайд должен логически вытекать из предыдущего и одновременно подготавливать появление следующего. Лучший способ проверить, правильно ли построена презентация, — быстро прочитать только заголовки. Если после этого станет ясно, о чем презентация — значит, структура построена верно.

Правило 2. Краткость — сестра убедительности.

После того как содержание презентации собрано, с ним следует аккуратно поработать, сократив его насколько возможно. Оптимальным объемом презентации считается 24 традиционных слайда, если презентация умещается в 16 слайдов — еще лучше, ну а 12 и менее слайдов — это то, что редко встречается и крепко запоминается. В среднем, один слайд - это 1,5 минуты выступления.

ПРАВИЛА СОЗДАНИЯ СЛАЙДОВ.

Правило 1. Думать о зрителе.

При разработке формы презентации всегда следует думать о том, как зритель ее будет видеть. В первую очередь нужно решить, где зрители будут смотреть вашу презентацию: на бумаге, экране монитора или на большом экране с помощью проектора. На конкурс вы создаете презентации для экрана монитора! И возможно, вашу презентацию захотят распечатать. Это следует учитывать при выборе размера и цвета шрифтов.

Правило 2. Последовательность и единство оформления.

Все однотипные элементы должны всегда быть в одном месте: если зритель знает, где ждать заголовок, а где график, он лучше схватывает суть дела. Заголовок – всегда в одном месте экрана. График – всегда в одном месте экрана. И т.д. Однотипные подписи – одинакового цвета и размера. И т.д.

Правило 3. Нет тексту!

«Нет» любому тексту, кроме абсолютно необходимого. Читать страницу за страницей и запоминать текст совсем непросто. Количество текста на слайдах должно составить не более 35% от всего содержимого слайдов. Весь ненужный текст следует оставить либо для устного выступления (для текста доклада, т.к. у нас заочная конференция), либо заменить его графиками, картинками и т.д.

ВАЖНЫЕ ЗАПРЕТЫ.

1. Изображения и текст на слайдах не должны быть мелкими (даже если презентация готовится для экрана).
2. Если презентация будет цветной, то следует избегать ярких, так называемых чистых тонов — алого, ярко–синего, зеленого, фиолетового (они режут глаз). Такие краски следует зарезервировать для выделения действительно ключевых моментов, а для рядовых изображений использовать пастельные тона и контрастные сочетания цветов шрифта и фона.
3. Пестрота на экране (больше четырех цветов одновременно).
4. Самый главный запрет - спецэффекты. Анимации наподобие вращающихся заголовков, переворачивающихся слайдов, любые звуки - все это лишь отвлекает слушателей и необоснованно растягивает время презентации.

ОСНОВНЫЕ ПРАВИЛА ВЫСТУПЛЕНИЯ.

Презентация состоит из двух частей: демонстрация слайдов и сопровождение их текстом. Слайды — поддержка выступления, а не наоборот. Очень часто докладчик вместо выступления просто зачитывает текст на слайдах. Таких ораторов слушатели не уважают, текст они могут и сами прочитать.

Именно поэтому на конкурс мы обязательно требуем ТЕКСТ ДОКЛАДА.

Правило 1. Стройте выступление на аргументах, а не на слайдах.

Если презентация сделана правильно и текст хорошо сбалансирован другими визуальными элементами, то все равно не следует вести свою аудиторию по презентации, как экскурсовод туристов: «посмотрите налево, посмотрите направо». Презентер должен вести аудиторию не от слайда к слайду, а от тезиса к аргументу, от аргумента к примеру, от вывода к выводу. Нельзя говорить «перейдем на страницу 7», надо — «как именно мы решаем эту проблему, рассказывается на слайде 7». Нельзя говорить «посмотрите на следующий слайд», надо «и что же из этого следует? А вот что!» - и показываем слайд.

Правило 2. Готовьтесь к выступлению.

Выступление должно быть подготовлено, прорепетировано и отхронометрировано (подогнано под временные рамки).

Правило 3. Помните, что аудитория — это живые люди. Позволяйте себе эмоции.

Позволяйте себе в тексте восклицательные знаки. Текст вовсе не должен быть сухим! Вы не диктор ТВ, вы живой человек, который свято верит в то, о чем он рассказывает

Работа за компьютером

При любой работе должны соблюдаться определённые правила поведения и безопасности, чтобы сохранить своё здоровье и уберечься от возможных травм или каких-либо заболеваний. Профилактика лучше лечения, поэтому правила работы за компьютером необходимо знать всем, ведь мы всё больше и больше времени проводим именно за компьютером — за ним сидим на работе, и за ним же сидим дома.

Памятка ниже будет весьма полезна для людей всех возрастных категорий, чья жизнь или работа напрямую связана с ПК и на компьютере приходится долго и часто работать.

1. Сидите прямо.
2. Вам должно быть удобно. Но это не значит, что надо подгибать ноги под себя или класть ногу на ногу, сутулиться. Этого делать НЕЛЬЗЯ!
3. Верхняя часть монитора должна быть расположена на уровне глаз или чуть ниже, а нижняя чуть ближе к Вам.
4. Расстояние между монитором и глазами должно быть 45-75 см.
5. Освещение должно падать так же как и при писании с левой стороны, свет не должен быть сильно ярким или тусклым.
6. Не забывайте моргать, при моргании глаз омывается слёзной жидкостью и не пересыхает, а пересыхание глаза вредит зрению.

7. Периодически необходима зарядка для глаз, которую можно делать и на работе, и дома.

8. Каждый час работы за компьютером делайте перерыв на 15-20 минут.

9. Можете купить специальные очки для работы за ПК, их можно найти в каждой оптике.

10. Если Вы устали, началось чувство сонливости или тяжести в глазах, Вы не должны продолжать работу!

11. Обязательно каждый день надо проветривать комнату, вытирать пыль, влажная уборка только на пользу пойдёт.

ЛАБОРАТОРНАЯ РАБОТА№1

Тема: Система команд микропроцессора

Цель: Изучить структуру и группы команд, выполняемые процессором

Оборудование: Справочный материал, карточки с заданиями

Справочный материал:1,2.

Содержание работы

1. Организационный момент
- Проверка готовности учащихся к уроку.
- Приветствие.
- Проверка готовности ребят к уроку
2. Постановка темы и цели урока
3. Повторение изученного материала

ЗАДАНИЕ 1

Программа помещается в оперативную память вместе с данными. Каждая команда хранится в ячейке или группе ячеек и имеет свой *адрес*.

Команда – это элементарная инструкция машине, выполняемая ею автоматически без каких либо дополнительных указаний и пояснений.

Все команды имеют одинаковую *структуру*. Машинная команда состоит из двух частей: операционной и адресной

Код **ОП** ерации (**КОП**)	Адреса

1. Код операции (операционная часть) определяет, какую команду нужно выполнить
2. Операнд (адресная часть) – над чем выполняется операция

Операционная часть команды (КОП) – это группа разрядов в команде, предназначенная для представления кода операции машине.

Адресная часть – это группа разрядов в команде, в которых записывают коды адресов операндов.

Операнд может быть простым и составным. Он может содержать в себе адресную часть, которая определяет, где хранятся данные и куда поместить результат операции.

В зависимости от количества используемых адресов, различают одноадресные, двух-, трех-, четырехадресные и безадресные команды.

- *Одноадресные команды*. Здесь указывается, где находится одно из чисел, второе должно быть помещено в АЛУ. Для этого существуют специальные команды пересылки данных.
- *Двухадресные команды*. Оба операнда в памяти, их адреса указаны. Результат заносится по одному из адресов.
- *Трехадресные команды*. Два адреса являются операндами, а третий служит для помещения туда результата.
- *Четырехадресные команды*. Два адреса – операнды, третий – результат, четвертый – адрес следующей команды.

• *Безадресные команды.* используются для выполнения служебных операций (очистить экран, заблокировать клавиатуру и так далее).

Команды выполняются друг за другом, а также могут быть выполнены ветвления и циклы.

Группы команд микропроцессора

Семейство микропроцессоров фирмы Intel от 8086 до Pentium имеет *базовую систему команд,* в состав которых входят:

Команды пересылки данных:

внутри процессора (Mov, Push, Pop и др.)

ввода-вывода (In, Out)

Арифметические команды:

основные (сложение, вычитание, умножение, деление)

дополнительные (INC, DEC)

Логические команды:

сдвиг, дизъюнкция, конъюнкция, и/или и др.

Обработка строковых данных:

пересылка, сканирование, сравнение, слияние.

Передачи управления:

безусловный переход, условный переход, прерывания, переход с возвратом.

Управления:

нет операции, внешняя синхронизация и так далее.

Каждая команда имеет много модификаций, чаще всего определяемых режимом адресации операндов.

Типы операндов команд

• Регистровые операнды указываются именами используемых регистров процессора.

• Непосредственные являются всегда числовыми величинами и могут быть в различных системах счисления. Числа различаются по последней букве, сопровождающей число:

b – двоичное число;

q – восьмеричное число;

d – десятичное число;

h – шестнадцатеричное число.

• Операнды в памяти могут указываться с помощью адресов ячеек, символическими именами, константами.

Различные комбинации этих элементов в команде называются *способами адресации*. Например, команда mov (переслать числа) может иметь следующие способы адресации:

m ov	r, r
m	r,

ov	m
m ov	m, r
m ov	im ed, r
m ov	im ed, m
m ov	Sr, m
m ov	Sr, r
m ov	m, Sr
m ov	r, Sr

Где r – РОН (регистр общего назначения); m – адрес в памяти;

imed – число; Sr – с

Задание к работе:

1. Изучить материал теоретической части;
2. Составить программу для условия: переписать 10000 байтов начиная с адреса A в другое место памяти начиная с адреса B. Оба эти имени относятся к сегменту данных, на начало которого указывает регистр A_S;
3. Записать микропрограмму суммирования двух чисел;
4. Сделать вывод о проделанной работе.

ЛАБОРАТОРНАЯ РАБОТА№2

Тема: Понятия о состоянии процессора.

Цель работы: Знакомство с компонентной структурой современного персонального компьютера.

Оборудование: Справочный материал, карточки с заданиями

Справочный материал:1,2.

Содержание работы

1. Организационный момент
- Проверка готовности учащихся к уроку.
- Приветствие.
- Проверка готовности ребят к уроку
2. Постановка темы и цели урока
3. Повторение изученного материала

ЗАДАНИЕ 1

Программа работы:

1. Определение ключевых параметров аппаратного обеспечения рабочего ПК

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**.

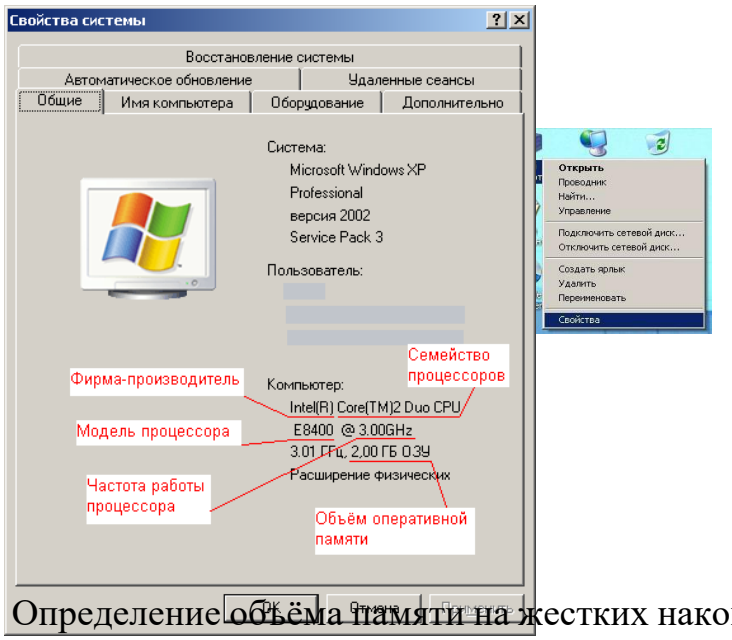
Мой компьютер □Свойства (в контекстном меню) □Общие MyComputer □Properties □General

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

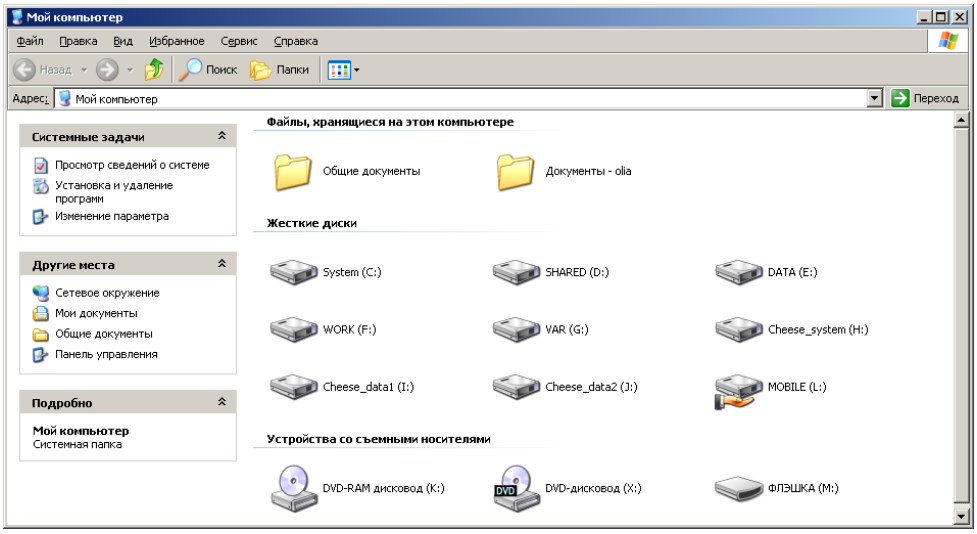
Приведите в отчёте данные о процессоре и оперативной памяти для Вашего рабочего компьютера в лаборатории и для Вашего домашнего компьютера. Сравните полученные характеристики. Какой из компьютеров имеет потенциально бóльшую производительность?



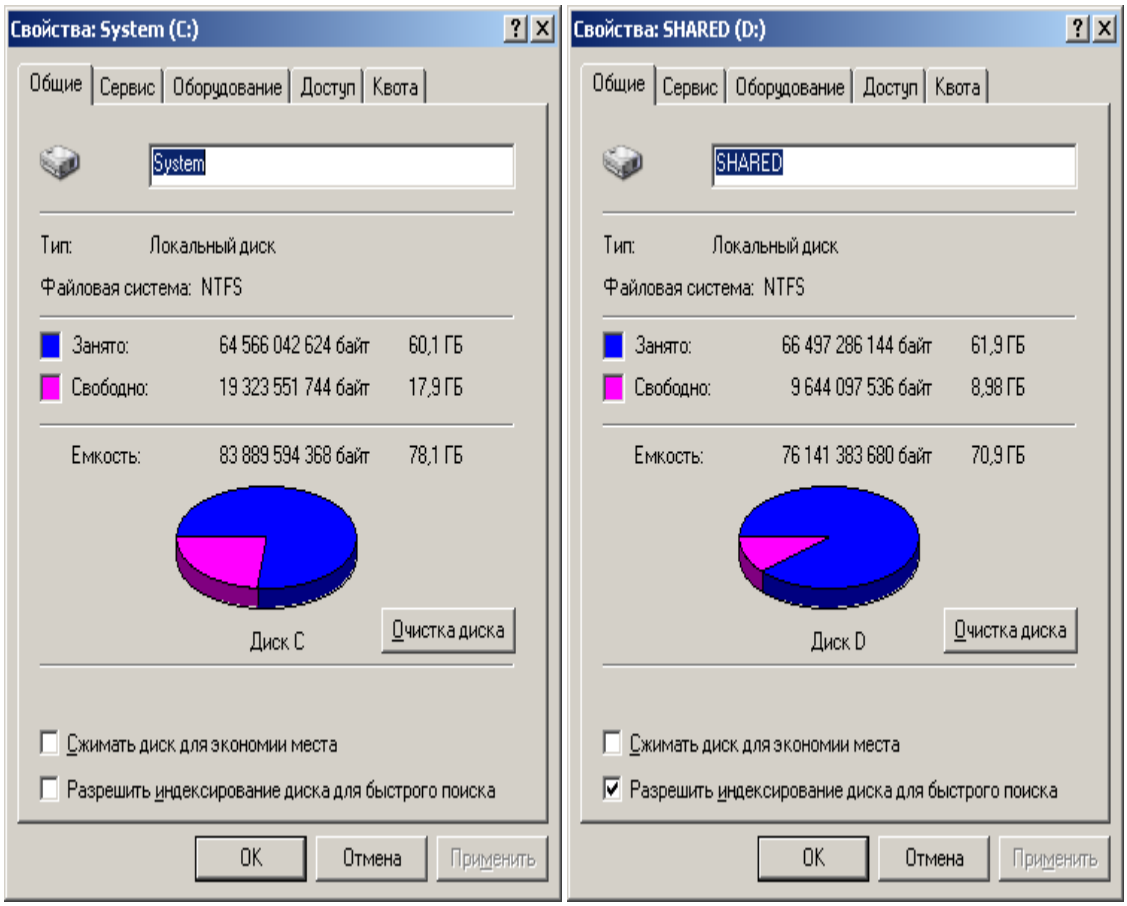
Определение объёма памяти на жестких накопителях

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.

Mycomputer (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.

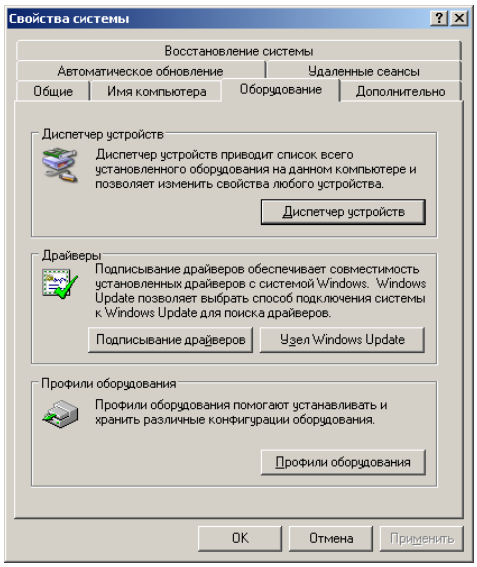


Определение количества физических накопителей, подключённых к компьютеру. Определение модели видеокарты.

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 Devicemanager



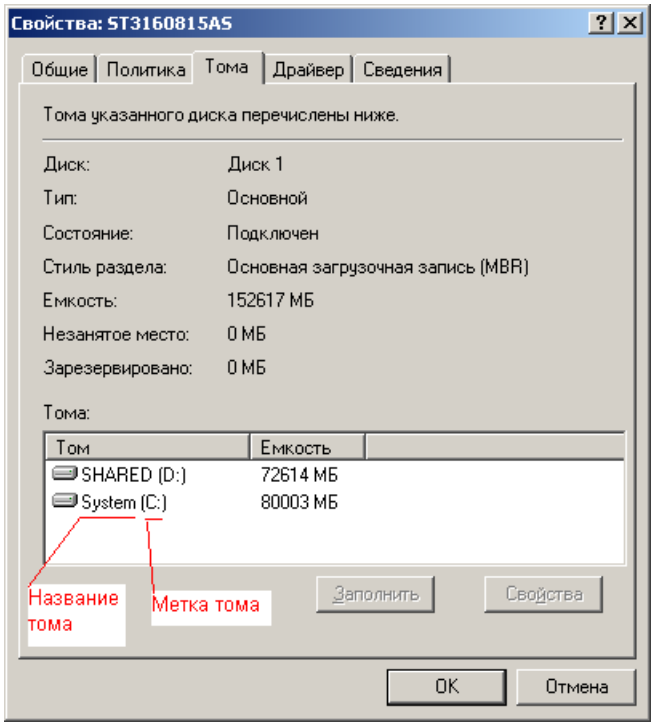
В появившемся окне найдите раскрывающееся меню **Дисковые устройства** (ищите иконку с жёстким диском). Раскройте меню, нажав на плюс. В раскрыв- шемся меню будут показаны все физически подключённые к компьютеру жесткие диски. Первые две буквы в названии винчестера кодируют название производителя. Остальные 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 – WesternDigital, 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 при- меру.

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

К компьютеру из примера на рисунке подключены сле- дующие внешние накопители:

Физический накопитель	Название тома	Метка тома	Общая ёмкость, ГБ	Процент свободного места, %
ST3160815 AS	System	C	78.1	100*17.9/78.1 = 23
ST3160815 AS	SHARE D	D	70.9	13
	…			

2. Подбор аппаратной конфигурации ПК по индивидуальному заданию

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 вы- бранной комплектации.

Варианты

1. Intel Core i7, OEM, видеоот NVIDIA
2. Intel Core i7, BOX, встроенное видео
3. Intel Core i5, OEM, видеоот AMD
4. Intel Core i5, BOX, встроенное видео
5. AMD A10, BOX, видеоот NVIDIA
6. AMD A8, BOX, видео от AMD
7. AMD A10, OEM, встроенное видео
8. AMD A8, OEM, видеоот 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**	**Позаданию**
	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	
	Типразъёма(Socket)	LGA2011	
	Частота собственная	3.6 ГГц	
	Числоядер	4	
	Кэш память (наличие и размер)	10 МБ	
	Мощность	130 Вт	
	Стоимость	10 190 р	
Кулер процессора	Типразъёма(Socket)	LGA2011	Необходим, поскольку процессор поставляется без вентилятора
	Выбранная модель	CPU cooler Cooler Master Hyper 412 Slim	Самый тихий
	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	
Материнска яплата	Типразъёма(Socket)	LGA2011	
	Выбраннаямо- дель	MB Gigabyte GA-X79- UD3	самая недорогая модель без встроенного видео с доста- точным запасом под опера- тивную память
	Ссылка	http://key.ru/shop/devices/motherboards/mb_gigabyte_ga-x79-ud3/	

Компонент	Характери-стика	Значение	Примечание
	Встроенная видеокарта	нет	
	Интерфейсви-деокарты	2 шт. PCI Express 3.0 x16	
	Количество слотовпамяти		
	Типмодулей памяти	DDR3	
	Частотаси-стемнойшины	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 xSPDIFout (коаксиальный) Оптический выход SPDIF-интерфейса	пришлось искать дополнительно характеристики от производителя не совпали с характеристикамина сайте мага-зинаhttp://www.gigabyte.ru/products/page/mb/ga-x79- ud3rev_10/specs/
	Стоимость	6890	
Видеокарта	**Производи-тель**	**NVIDIA**	**Позаданию**
	Интерфейс	до 2 PCI Express 3.0 x16	
	Выбраннаямо-дель	2048M Asus GeForce GTX660 DDR5	Среднийценовойдиапазон
	Ссылка	http://key.ru/shop/devices/videokarty/2048m_asus_geforce_gtx660_ddr5_2xdvi_hdmi_dp_pci-e/	
	Чипсет	GeForce GTX 660	
	Частота	1072 МГц	
	Объёмпамяти	2048 Мб	
	Видеовыходы	2xDVIHDMI DP	
	Мощность	150 Вт	информация с сайта произ-водителя
	Стоимость	8090	
	Длинавидео-карты	10 дюймов = 255 мм	с сайтапроизводителя

Компонент	Характери-стика	Значение	Примечание
Оперативная память	Типмодулей памяти	DDR3	
	Количество слотовпамяти	2	
	Частотаси-стемнойшины	2400/2133/1866/1600/1333/1066 МГц	
	Выбранная модель	Модульпамяти 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_kingston_xmp_predator_non-ecc_cl11_kit_of_2/	
	Стоимость	5090	
Привод CD/DVD	нет		Нет необходимости
Жесткий диск HDD	Интерфейс	SATA	Современный стандарт де- факто, 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
	Формфактор	ATX	или совместимый
	Длинавидео-карты	10 дюймов = 255 мм	
	Выбраннаямо-дель	Codegen Q3339-A2 Black ATX	По внешнему виду
	Наличиеблока питания	да	
	Ссылка	http://key.ru/shop/devices/korpu	
	Стоимость	1490	

Компонент	Характери-стика	Значение	Примечание
Монитор	Разъём	2xDVIHDMI DP	
	Выбранная модель	Dell UltraSharp U2212HM Black	С максимальным углом об-зора, чёрный
	Ссылка	http://key.ru/shop/noutbuki_kompyutery_i_aksessuary/tftmonitory/monitor_215_dell_ultrasharp_u2212hm_black_8ms_gtg_tft/	
	Диагональ	21,5 дюйма	
	Стоимость	8890	
Клавиатура	Выбраннаямо-дель	CBR KB-107 Black_USB	
	Ссылка	http://key.ru/shop/noutbuki_kompyutery_i_aksessuary/mouses_keyboards/klaviatury/cbr_klaviatura_kb-107_107_kl/	
	Стоимость	259	Самая недорогая, черная, с интерфейсом PS/2, как на материнской плате USB тоже подошла бы
Мышь	Выбранная модель	Razer Ouroboros	Производитель Razer, бес-проводная, с индикатором Заряд абатареи
	Ссылка	http://key.ru/shop/noutbuki_kompyutery_i_aksessuary/mouses_keyboards/myshi/mysh_razer_ouroboros/	
	Стоимость	5990	
…			

ЛАБОРАТОРНАЯ РАБОТА№3

Тема: Микроконтроллеры.

Цель работы: изучить архитектуру и принципы построения микроконтроллера AVR ATMEGA128.

Оборудование: Справочный материал , карточки с заданиями

Справочный материал:1,2.

Содержание работы

1. Организационный момент
- Проверка готовности учащихся к уроку.
- Приветствие.
- Проверка готовности ребят к уроку
2. Постановка темы и цели урока
3. Повторение изученного материала

ЗАДАНИЕ 1

Перед работой необходимо проработать теоретический материал и конспект лекций, ознакомиться со структурой и принципами функционирования микроконтроллера AVR ATMEGA128.

1. **Основные характеристики микроконтроллера AVR ATMEGA128.**

AVR-архитектура объединяет высокопроизводительный RISC-процессор с раздельным 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 раздельной 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;

широтно-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 внутри системно 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 отладке и для программирования, а также шесть программно выбираемых режимов уменьшения мощности.

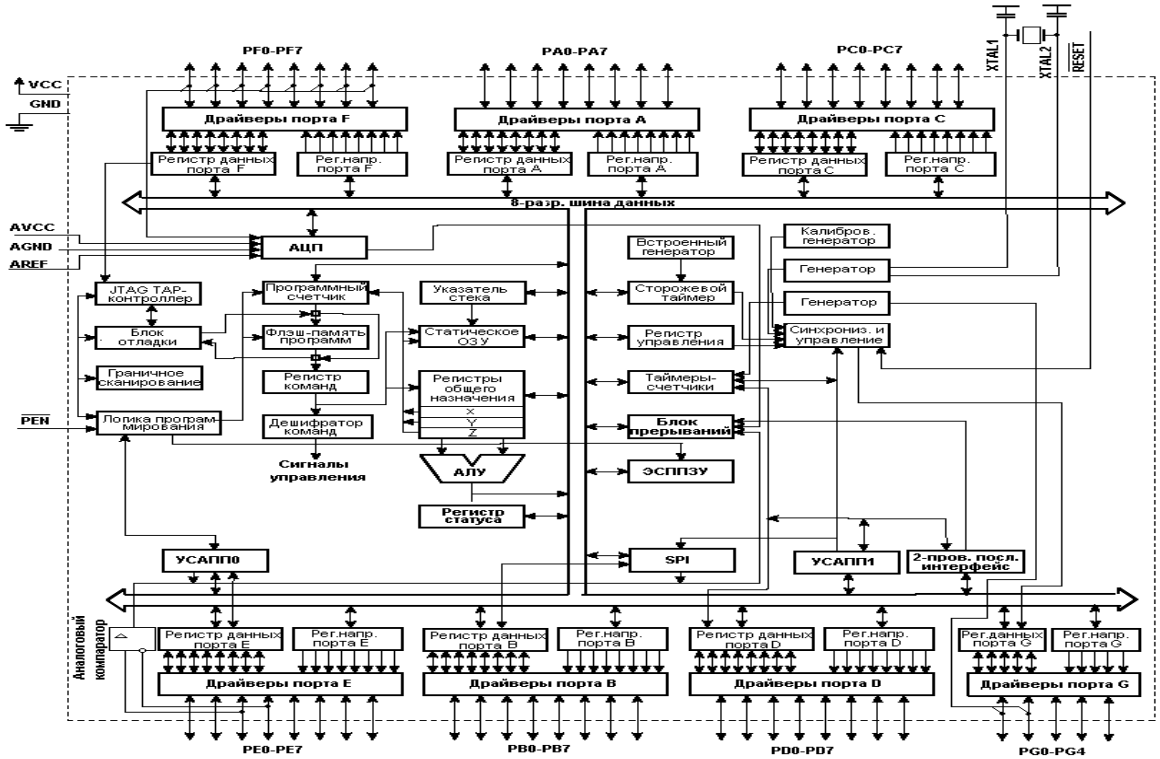


Рисунок 2 – Функциональная схема

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 NoiseReduction) 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 (ExtendedStandby) 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 двух операционность: чтение во время записи. За счет сочетания 8-разр. RISC ЦПУ с внутри систем носамопрограммируемой 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.

1.2. 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 раздельным 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 PC (ProgramCounter). Программа исполняется последовательно. Для управления ходом выполнения программы существуют команды перехода, изменяющие соответствующим образом значение PC.

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 1.1 – 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 1.3).

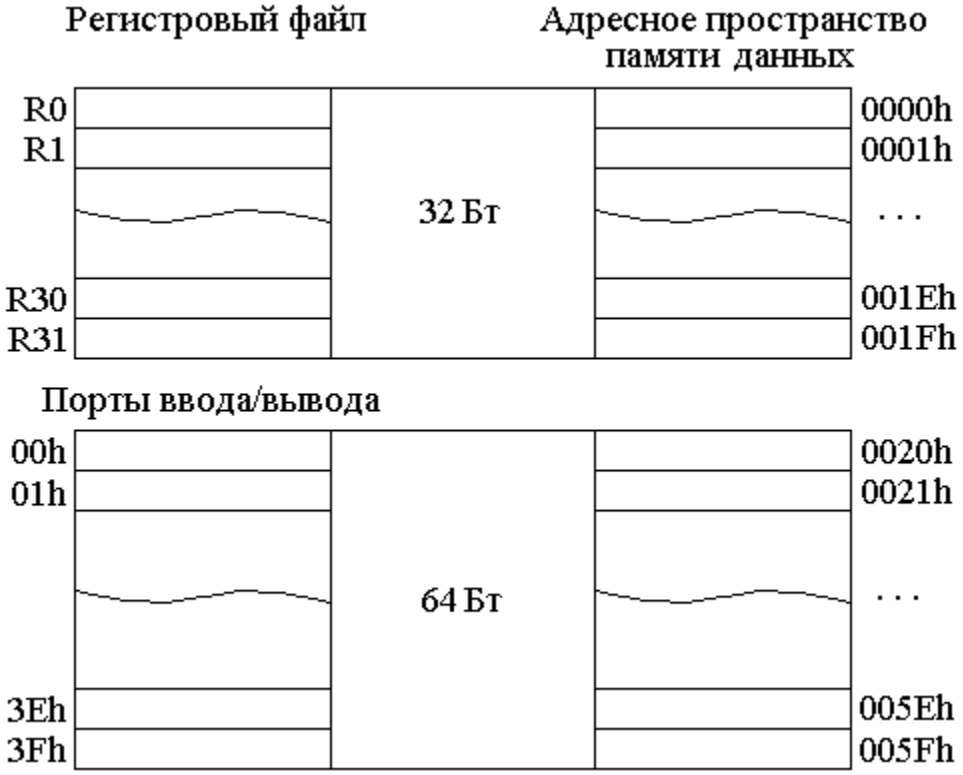


Рисунок 1.2 – Иллюстрация отображения регистров общего назначения и портов ввода/вывода на адресное пространство памяти данных

X		Y		Z	
27	26	29	28	31	30

Рисунок 1.3 – 16-разрядные регистры X, Y, Z, использующиеся 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 1.1 – 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 1.4.

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

Рисунок 1.4 – 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 – флагзнака, S = N 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.

- 1.3 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 двухоперандными, причем первым операндом является приемник данных, а вторым – источник данных.

- Команда загрузки констант **ldi R, K** применяется для записи непосредственного значения К 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 К 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 **movRd, 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 R_8, (R_{16})** – загрузка данных из ячейки памяти, адрес которой находится в 16-разрядном регистре R_{16}, в регистр общего назначения R_8
- **st (R_{16}), R_8** – загрузка данных из регистра общего назначения R_8 в ячейку памяти, адрес которой находится в 16-разрядном регистре R_{16},
- **ldd R_8, (R_{16}+Q)** – загрузка данных в регистр общего назначения R_8 из ячейки памяти, адрес которой находится как сумма значения, находящегося в 16-разрядном регистре R_{16}, и смещения Q,.
- **std (R_{16}+Q), R_8** – загрузка данных из регистра общего назначения R_8 в ячейку памяти, адрес которой находится как сумма значения, находящегося в 16-разрядном регистре R_{16}, и смещения Q,.
- **ld R_8, (R_{16})** – загрузка в регистр общего назначения R_8 данных из ячейки памяти, адрес которой находится в 16-разрядном регистре R_{16};
- 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.

• Основные команды для выполнения операций сложения, вычитания и умножения (для чисел без знака):

• **addRd, Rs** команда сложения (addition), действие: Rd=Rd+Rs;

• **adcRd, Rs** команда сложения (addition) с учетом переноса, действие: Rd=Rd+Rs+C;

• **subRd, Rs** команда вычитания (subtraction), действие: Rd=Rd–Rs.

• **mulRd, Rs** команда умножения (multipl.), действие: R1, R0=Rd*Rs;

• Команды изменяют флаги переноса **C**, переполнения V, знака N, **S,** и нуля **Z**. При выполнении операции умножения **n**-значных чисел местонахождение результата разрядностью 2n фиксировано и не указывается в команде: при умножении двух байтов результат размером в слово заносится в регистровую пару (R1, R0), в R0 – младшее слово, в R1 – старшее слово.

• Примеры:

• **add R10, R15** R10 = R10 + R15;

• **sub R2, R7** R2 = R2 – R7;

• **mul R5, R16** R1,R0 = R5 * R16.

• Команды положительного и отрицательного приращения (инкремента и декремента):

• **inc R** инкремент (increment), действие R=R + 1;

• **dec R** декремент (decrement), действие R=R – 1;

• В качестве операнда в этих командах допускается использовать только регистр общего назначения.

• Примеры:

• **inc R20** действие R20 = R20 + 1;

• **dec R16** действие R16 = R16 – 1;

• Основными логическими командами микроконтроллера AVR MEGA128 являются:

• **or Rd, Rs** логическое “или”; действие: Rd = RdorRs;

• **and Rd, Rs** логическое “и”; действие: Rd = RdandRs;

• **eor Rd, Rs** “исключающее или”; действие: Rd = RdeorRs.

• Данные команды выполняют операции поразрядного логического “или”, логического “и”, “исключающего или” (см. таблицу 1.2) над операндами, находящимися в регистрах общего назначения, причем результат записывается по адресу первого операнда. Примеры:

• **or R7, R11** действие: R7 = R7 or R11;
• **and R10, R11** действие: R10 = R10 and R11;
• **eor R25, R30** действие: R25 = R25 eor R30.
• Таблица 1.2 – Таблицы истинности логических операций or, end, eor

or (или)			and (и)			eor (исключ. или)		
Вход		Вы ход	ход		Вы ход	ход		Вых од
	B	Q		B	Q		B	Q
	0	0		0	0		0	0
	1	1		1	0		1	1
	0	1		0	0		0	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 1.3.

• Таблица 1.3 – Примеры поразрядных логических операций

Пример поразрядного маскирования or				мер ного вания	eor
Rd	xxxxx xxx	Rd xxxxxx xx	Rd		10100 110
Rs	00010 010	Rs 101101 01	Rs		00010 010
Rd=Rdor Rs	xxx1x x1x	Rd=Rdan x(xx0x dRs 0x	Rd=Rdeo rRs		10110 100

• 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 нуль. 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 нуль. 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;

• **swapR19** 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 **cpRd, 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
- 2 Порядок проведения работы и указания по ее выполнению
- **Занятие 1.**
- Проработайте теоретические материалы и конспекты лекций, ознакомитесь 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 2.1.

№ 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	5	6	7	8	9	0	1	2	3	4

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 1.4).

Задание 2. Разработать в среде программирования CodeVision 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 CodeVision 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 соответствие с вариантом задания, указанном в таблице 1.4.
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 указанном в регистре **X**, а второе слагаемое будет располагаться регистре **R16**. Результат вычислений должен находиться в памяти с адресом указанном в регистре **Y**. Для правильного выполнения программы необходимо выполнить начальную инициализацию слагаемых.

Так как, первое слагаемое находиться в памяти с адресом указанном 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 храниться в двухбайтовом регистре **X**).

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

ldi R16, 60

Теперь оба слагаемые имею 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** находиться 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 CodeVision AVR программу на языке ASSEMBLER для микроконтроллера AVR ATMEGA 128, выполняющую сложение двух двухбайтных чисел. Варианты выполнения задания находятся в таблице 2.2.

Таблица 2.2 – 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 2.3 значения представлены в шестнадцатеричной системе. Номер варианта выбирается, как предпоследняя цифра номера зачетной книжки.

Таблица 2.3 – Таблица выбора размещения операндов

Первое слагаемое										
Второе слагаемое										
Результат										

Рисунок 1 – Блок-схема алгоритм сложения двух двухбайтных чисел

В таблице 2.3 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

ЛАБОРАТОРНАЯ РАБОТА№4

Тема: Программирование на машинном языке.

Цель работы:

1) познакомиться с основными понятиями языка Турбо Паскаль, правилами записи арифметических выражений,

2) формировать навыки представления арифметических выражений на Паскале.

Оборудование: Справочный материал , карточки с заданиями

Справочный материал:1,2.

Содержание работы

1. Организационный момент
- Проверка готовности учащихся к уроку.
- Приветствие.
- Проверка готовности ребят к уроку
2. Постановка темы и цели урока
3. Повторение изученного материала

ЗАДАНИЕ 1

1. Ознакомиться с теоретическим материалом «Основные сведения о языке Турбо Паскаль»:

Для того чтобы правильно записывать арифметические выражения, нужно соблюдать следующие правила:

1.Все символы пишутся в строчку на одном уровне. Проставляются все знаки операций (нельзя пропускать знак умножения).

2. Не допускаются два следующих подряд знака операций (нельзя A+-B; можно A+ (-B)).

3. Операции с более высоким приоритетом выполняются раньше операций с меньшим приоритетом. Порядок убывания приоритетов:

- вычисление функции;
- унарная операция смены знака (-);
- *, /, div, mod;
- +, - .

4. Несколько записанных подряд операций одинакового приоритета выполняются последовательно слева направо.

5. Часть выражения, заключенная в скобки, вычисляется в первую очередь.

(Например, (A+B) * (C—D) — умножение производится после сложения и вычитания.)

Не следует записывать выражений, не имеющих математического смысла. Например, деление на нуль, логарифм отрицательного числа и т.п.

В Паскале нет операции или стандартной функции возведения числа в произвольную степень. Для вычисления x^y рекомендуется поступать следующим образом:

- если y – целое значение, то степень вычисляется через умножение;

например, x^3- xxx.

• если y – вещественное значение, то используется следующая математическая формула:

$x^y = e^{y\ln(x)}$.

На Паскале это будет выглядеть так: Exp(y*Ln(x)).

2. Используя правила записи арифметических выражений на Паскале, выполнить следующие задания:

1) Для следующих формул записать соответствующие арифметические выражения на Паскале:

в) $\frac{a+b}{c}+\frac{c}{ab}$; г) $\frac{x+y}{a_1}\cdot\frac{a_2}{x-y}$;

д) $10^4\alpha-3\frac{1}{5}\beta$; е) $\left(1+\frac{x}{2!}+\frac{y}{3!}\right)\Big/\left(1+\frac{2}{3+xy}\right)$.

2) Записать математические формулы, соответствующие следующим выражениям на Паскале:

а) *(p+q)/(r+s)-p*q/(r*s);*

б) *lE3+beta/(x-gamma*delta);*

в) *a/b*(c+d)-(a-b)/b/c+lE-8.*

3) Для следующих формул записать соответствующие арифметические выражения на Паскале:

а) $(1+x)^2$; б) $\sqrt{1+x^2}$; в) $\cos^2 x^2$; г) $\log_2\frac{x}{5}$;

д) $\arcsin x$; е) $\frac{e^x+e^{-x}}{2}$; ж) $x^{\sqrt{2}}$; з) $\sqrt[3]{1+x}$;

и) $\sqrt{x^8+8^x}$; к) $\frac{xyz-3,3\left|x+\sqrt[4]{y}\right|}{10^7+\ln 4!}$; л) $\frac{\beta+\sin^2\pi^4}{\cos 2+|\mathrm{ctg}\,\gamma|}$.

4) Вычислить значения выражений:

а) round(6.9); в) 20 div 6; д) 20 mod 6;

б) round(6.2); г) 2 div 5; ж) 2 mod 5;

и) 3*7 div 2 mod 7/3-trunc(sin(1)).

3. Ответите на контрольные вопросы.

4. Показать работу преподавателю на проверку.

5. Привести рабочее место в порядок.

Контрольные вопросы:

1. Из чего состоит алфавит языка Турбо Паскаль?

2. Как записываются буквы греческого алфавита на Паскале?

3. Почему в Паскале аргумент функции всегда записывают в скобках (например, пишут ln(5), а не ln5)?

4. Запишите самостоятельно арифметическое выражение и соответствующее ему выражение на Паскале.

ЛАБОРАТОРНАЯ РАБОТА№5

Тема: Программирование на языках высокого уровня

Цель работы:

1) разобрать структуру программ в общем виде;

2) сформировать навыки объявления переменных, использование инструкции присваивания, ввода, вывода;

3) учиться составлять простейшие программы.

Оборудование: Справочный материал, карточки с заданиями

Справочный материал:1,2.

Содержание работы

1. Организационный момент
- Проверка готовности учащихся к уроку.
- Приветствие.
- Проверка готовности ребят к уроку
2. Постановка темы и цели урока
3. Повторение изученного материала

Задание 1. Объявление переменных

Приступая к решению задач этого раздела, следует помнить, что:

- каждая переменная программы должна быть объявлена;
- объявления переменных помещают в раздел, который начинается словом **var;**
- инструкция объявления переменной выглядит так: Имя Переменной: Тип;
- в имени переменной можно использовать буквы латинского алфавита и цифры (первым символом должна быть буква);
- основными числовыми типами языка TurboPascal являются: integer (целый), real (вещественный);
- после инструкции объявления переменной рекомендуется указывать назначение переменной.

Основные типы данных языка Турбо Паскаль:

- целые числа (INTEGER и др.);
- действительные, вещественные числа (REAL и др.);
- символы (CHER);
- строки (STRING);
- логический тип (BOOLEAN).

Пример1. Объявите переменные, необходимые для вычисления значения функции $y = x^2$.

Решение: x: real; {аргумент функции}

y: real; {значение функции}

Объявите переменные, необходимые для пересчета веса из фунтов в килограммы.

Объявите переменные, необходимые для вычисления площади кольца.

Объявите переменные, необходимые для вычисления объема параллелепипеда.

Объявите переменные, необходимые для вычисления площади круга.

Объявите переменные, необходимые для вычисления стоимости покупки, состоящей из нескольких тетрадей, карандашей и линейки.

Задание 2. Инструкция присваивания

Приступая к решению задач этого раздела, следует вспомнить, что:

• инструкция присваивания используется для изменения значений переменных, в том числе и для вычислений по формулам;

• тип выражения, находящегося в правой части инструкции присваивания, должен соответствовать типу переменной, имя которой стоит слева от символа инструкции присваивания (при нарушении соответствия типа переменной и выражения компилятор выводит сообщение об ошибке **Typemissmatch**— несоответствие типов).

*Пример 2.*Запишите инструкцию, которая увеличивает на единицу значение переменной n.

*Решение:*n:=n+1;

2.1. Запишите инструкцию, которая присваивает переменной summанулевое значение.

2.2. Запишите инструкцию вычисления среднего арифметического переменных xi и x2.

2.3. Запишите в виде инструкции присваивания формулу вычисления значения функции

$y = -2{,}7x^3 + 0{,}23x^2 - 1{,}4$.

2.4.Запишите в виде инструкции присваивания формулу пересчета веса из фунтов в килограммы (один фунт равен 409,5 г).

2.5. Запишите в виде инструкции присваивания формулу пересчета расстояния из километров в версты (одна верста равна 1066,8 м).

2.6. Запишите в виде инструкции присваивания формулу вычисления площади треугольника:

$s = 1/2\ ah$, где a — длина основания треугольника, h — его высота.

Задание 3. Вывод

Приступая к решению задач этого раздела, следует вспомнить, что:

• инструкции write и writeln предназначены для вывода на экран монитора сообщений и значений переменных;

• одна инструкция write (writeln) может вывести на экран значения нескольких переменных и (или) несколько сообщений;

• инструкция writeln без параметров переводит курсор в начало следующей строки экрана.

Пример 3. Написать инструкции вывода значений переменных a, b и c (тип вещественный, 2 знака после запятой). Значение каждой переменной должно быть выведено на отдельной строке.

Решение: writeln (a:6:2);
writeln (b:6:2);
writeln(c:6:2);

Написать программу, которая выводит на экран имя и фамилию.

Написать программу, которая выводит на экран четверостишие:

Унылая пора! Очей очарованье!
Приятна мне твоя прощальная краса –
Люблю я пышное природы увяданье,
В багрец и золото одетые леса.

А. С. Пушкин

3.3. Написать инструкцию вывода значения переменной *a* (тип*real*) с тремя цифрами в дробной части.

Задание 4. Ввод

Приступая к решению задач этого раздела, следует вспомнить, что:

• для ввода с клавиатуры во время работы программы исходных данных (значений переменных) предназначена инструкция readln;

• используя одну инструкцию readln, можно ввести значения нескольких переменных;

• тип данных, вводимых во время работы программы, должен соответствовать типу переменной, указанной в инструкции readln;

• в случае несоответствия типа введенных данных типу переменной, значение которой вводится с клавиатуры, программа завершает работу и на экран выводится сообщение Error 106: Invalidnumericformat (если программа запущена из среды разработки, т. е. из TurboPascal) или Runtimeerror 106 (если программа запущена из операционной системы).

Пример 4. Написать инструкцию, которая обеспечивает ввод значений переменных u и r. Предполагается, что во время работы программы пользователь будет набирать числа в одной строке.

Решение: readln (u,r);

4.1. Написать инструкции, которые обеспечивают ввод значений переменных u и r. Предполагается, что во время работы программы пользователь будет после набора каждого числа нажимать клавишу <Enter>.

4.2. Объявите необходимые переменные и напишите фрагмент программы вычисления объема цилиндра, обеспечивающий ввод исходных данных.

4.3. Объявите необходимые переменные и напишите инструкции ввода исходных данных для программы вычисления стоимости покупки нескольких тетрадей и карандашей. Предполагается, что во время работы программы пользователь будет вводить данные о каждой составляющей покупки в отдельной строке: сначала цену, затем количество.

3. Оформить работу в тетради и показать преподавателю.
4. Привести рабочее место в порядок.

Информационное обеспечение обучения

Печатные и электронные издания

Основные учебные издания:

1. Алфёров, В. В. Вычислительная техника и сети в отрасли: учебное пособие

/ В. В. Алфёров, Ю. М. Миронов. — Москва: Московская государственная академия водного транспорта, 2018. — 152 с. — ISBN 2227-8397. — Текст: электронный // Электронный ресурс цифровой образовательной среды СПО PROFобразование: [сайт]. — URL: https://profspo.ru/books/67596

Дополнительные учебные издания:

2. Тюрин, И. В. Вычислительная техника: учебное пособие / И. В. Тюрин. — Тамбов: Тамбовский государственный технический университет, ЭБС АСВ, 2019.

— 112 с. — ISBN 978-5-8265-2099-4. — Текст: электронный // Электронный ресурс цифровой образовательной среды СПО PROFобразование: [сайт]. — URL: https://profspo.ru/books/99754

Интернет ресурсы

3. http://www.inf1.info/book/export/html/210 - Логические основы ЭВМ
4. http://cssblok.ru/computer/prinsip.html - Основные принципы функционирования компьютеров
5. http://infolike.narod.ru/logic.html - Основы алгоритмизации

Электронно-библиотечная система:

6. ЭБС «elibrary», ООО «РУНЭБ»
7. ЭБС «IPRbooks», ООО «Ай Пи Ар Медиа»
8. ЭБС «Лань», ООО «Издательство Лань»
9. ЭБС «PROFобразование»
10. ЭБС «Book.ru»