

государственное бюджетное профессиональное образовательное учреждение  
«Пермский химико-технологический техникум»

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ  
ДЛЯ ОБУЧАЮЩИХСЯ  
ПО ВЫПОЛНЕНИЮ ПРАКТИЧЕСКИХ РАБОТ**

для специальности 09.02.03 Программирование в компьютерных системах  
по дисциплине ОП.02 «Архитектура компьютерных систем»

2020

**СОДЕРЖАНИЕ**

ВВЕДЕНИЕ .....	3
ПРАВИЛА ВЫПОЛНЕНИЯ ПРАКТИЧЕСКИХ РАБОТ .....	5
ОПИСАНИЕ РАБОЧЕГО МЕСТА ОБУЧАЮЩЕГОСЯ.....	6
ПРАКТИЧЕСКИЕ РАБОТЫ .....	7
Практическая работа № 1 .....	7
Практическая работа № 2 .....	9
Практическая работа № 3 .....	13
Практическая работа № 4 .....	15
Практическая работа № 5 .....	18
Практическая работа № 6 .....	24
Практическая работа № 7 .....	26
Практическая работа № 8 .....	27
Практическая работа № 9 .....	29
Практическая работа № 10 .....	30
Практическая работа № 11 .....	32
Практическая работа № 12 .....	34
Практическая работа № 13 .....	36
Практическая работа № 14 .....	39
Практическая работа № 15 .....	43

## ВВЕДЕНИЕ

Учебная дисциплина ОП.02 «Архитектура компьютерных систем» является обязательной в цикле общепрофессиональных дисциплин основной профессиональной образовательной программы по специальности 09.02.03 «Программирование в компьютерных системах».

В результате освоения учебной дисциплины обучающийся должен **уметь**:

- получать информацию о параметрах компьютерной системы;
- подключать дополнительное оборудование и настраивать связь между элементами компьютерной системы;
- производить инсталляцию и настройку программного обеспечения компьютерных систем;

В результате освоения учебной дисциплины обучающийся должен **знать**:

- базовые понятия и основные принципы построения архитектур вычислительных систем;
- типы вычислительных систем и их архитектурные особенности;
- организацию и принцип работы основных логических блоков компьютерных систем;
- процессы обработки информации на всех уровнях компьютерных архитектур;
- основные компоненты программного обеспечения компьютерных систем;
- основные принципы управления ресурсами и организации доступа к этим ресурсам

Формируемые дисциплиной компетенции:

ПК 1.1. Выполнять разработку спецификаций отдельных компонент.

ПК 1.2. Осуществлять разработку кода программного продукта на основе готовых спецификаций на уровне модуля.

ПК 1.5. Осуществлять оптимизацию программного кода модуля.

ПК 2.3. Решать вопросы администрирования базы данных.

ПК 2.4. Реализовывать методы и технологии защиты информации в базах данных.

ПК 3.1. Анализировать проектную и техническую документацию на уровне взаимодействия компонент программного обеспечения.

ПК 3.2. Выполнять интеграцию модулей в программную систему.

ПК 3.4. Осуществлять разработку тестовых наборов и тестовых сценариев.

ОК 1. Понимать сущность и социальную значимость своей будущей профессии, проявлять к ней устойчивый интерес.

ОК 2. Организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество.

ОК 3. Принимать решения в стандартных и нестандартных ситуациях и нести за них ответственность.

ОК 4. Осуществлять поиск и использование информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития.

ОК 5. Использовать информационно-коммуникационные технологии в профессиональной деятельности.

ОК 6. Работать в коллективе и в команде, эффективно общаться с коллегами, руководством, потребителями.

ОК 7. Брать на себя ответственность за работу членов команды (подчиненных), за результат выполнения заданий.

ОК 8. Самостоятельно определять задачи профессионального и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации.

ОК 9. Ориентироваться в условиях частой смены технологий в профессиональной деятельности.

Методические указания предназначены для проведения практических занятий по дисциплину «Архитектура компьютерных систем», закрепления теоретических знаний и получения навыков работы в области прикладного программирования.

Методические указания разработаны в соответствии с рабочей программой дисциплины «Архитектура компьютерных систем» по специальности 09.02.03 «Программирование в компьютерных системах».

Содержание методических указаний по выполнению практических работ соответствует требованиям Федерального государственного стандарта среднего профессионального образования по специальности 09.02.03 «Программирование в компьютерных системах».

По учебному плану, и в соответствии с рабочей программой профессионального дисциплины «Архитектура компьютерных систем», на изучение обучающимися предусмотрено 114 часов, в том числе: обязательной аудиторной учебной нагрузки 76 часов; 30 практических занятий, самостоятельной работы обучающегося 38 часов.

Методические указания включают 15 практических работ. Каждая практическая работа содержит сведения о теме, цели ее проведения и формируемых компетенциях, включает пояснения к работе, содержание отчета, контрольные задания или вопросы, список литературы.

К выполнению практических работ обучаемые приступают после подробного изучения соответствующего теоретического материала и прохождения инструктажа по технике безопасности.

Характер практических работ репродуктивный и частично-репродуктивный.

**Примечание:**

Репродуктивный характер – обучающиеся пользуются подробными инструкциями, в которых указаны: цель работы, пояснения, оборудование, аппаратура, материалы и их характеристики, порядок выполнения работы, таблицы, выводы (без формулировок), контрольные вопросы, литература.

Частично-поисковый характер – обучающиеся не пользуются подробными инструкциями, им не задан порядок выполнения заданий, от студентов требуется самостоятельный подбор оборудования, выбор способов выполнения работы, справочной литературы.

Поисковый характер – обучающиеся должны решить новую для них проблему, опираясь на имеющиеся у них теоретические знания.

Работы частично-поискового и поискового характера выполняются, как правило, при изучении дисциплин, составляющих ядро конкретной специальности, а так же дисциплин, связанных с обслуживанием, эксплуатацией и ремонтом различного оборудования.

## **ПРАВИЛА ВЫПОЛНЕНИЯ ПРАКТИЧЕСКИХ РАБОТ**

1. Студент должен прийти на практическое занятие, подготовленным к выполнению работы. Студент, не подготовленный к работе, не может быть допущен к ее выполнению.
2. Каждый студент выполняет вариант задания, указанного преподавателем в начале занятия.
3. Каждый студент после выполнения работы должен представить отчет о проделанной работе.
4. Таблицы и схемы следует выполнять с помощью чертежных инструментов (линейки, циркуля и т. д.) карандашом с соблюдением ЕСКД.
5. В заголовках граф таблиц обязательно проводить буквенные обозначения величин и единицы измерения в соответствии с ЕСКД.
6. Вспомогательные расчеты можно выполнить на отдельных листах, а при необходимости на листах отчета.
7. Если студент не выполнил практическую работу или часть работы, то он может выполнить работу или оставшуюся часть во внеурочное время, согласованное с преподавателем.
8. Оценку по практической работе студент получает, с учетом срока выполнения работы, если:
  1. работа выполнена правильно и в полном объеме;
  2. студент может пояснить выполнение любого этапа работы;
  3. отчет выполнен в соответствии с требованиями к выполнению работы;
  4. студент ответил на дополнительные теоретические вопросы преподавателя.
9. Для получения допуска к экзамену студент должен выполнить все предусмотренной программой работы после сдачи отчетов при удовлетворительных оценках.

После окончания занятий обучающиеся приводят в порядок рабочее место, показывают преподавателю полученные результаты. После утверждения преподавателем предъявленных результатов, каждый обучающийся оформляет отчет о проделанной работе, представляет его на проверку и подпись преподавателю в тот же день или на следующем практическом занятии.

## **ОПИСАНИЕ РАБОЧЕГО МЕСТА ОБУЧАЮЩЕГОСЯ**

1. Практические работы по дисциплине выполняются в аудитории, компьютерном классе.
2. Для выполнения практических работ необходимы:
  - методические указания;
  - чертежные инструменты;
  - конспект лекций и учебник;
  - персональный компьютер.
  - операционная система Windows;
  - приложения MS Office;
  - Эмулятор УК «Нейман»;
  - Микросхемы для изучения логических устройств;
3. Выполнение расчетов и оформление отчета студент выполняет индивидуально, согласно указанного преподавателем варианта.
4. По выполнению всех этапов задания и их оформления студент представляет отчет по практической работе.
5. Для получения оценки студент защищает практическую работу по предоставленному отчету и отвечая на вопросы преподавателя по теории.

## ПРАКТИЧЕСКИЕ РАБОТЫ

### Практическая работа № 1

Тема: Кодирование символьной информации.

Цель: Закрепление знаний о кодировании символьной информации

Пояснения к работе:

Для представления текстовой (символьной) информации в компьютере используется алфавит мощностью 256 символов. Один символ из такого алфавита несет 8 бит информации, т.к.  $2^8 = 256$ . Но 8 бит = 1 байту, следовательно, двоичный код каждого символа в компьютерном тексте занимает 1 байт памяти.

Пример 1. Сколько бит памяти компьютера займет слово «Микропроцессор»?

Решение. Слово состоит из 14 букв. Каждая буква является символом компьютерного алфавита и поэтому занимает 1 байт памяти. Слово займет 14 байт = 112 бит памяти, т.к. 1 байт = 8 бит.

Таблица кодировки: таблица, в которой устанавливается соответствие между символами и их порядковыми номерами в компьютерном алфавите.

Все символы компьютерного алфавита пронумерованы от 0 до 255. Каждому номеру соответствует 8-разрядный двоичный код от 00000000 до 11111111. Этот код есть порядковый номер символа в двоичной системе счисления.

Для разных типов ЭВМ используются различные таблицы кодировки. С распространением персональных компьютеров типа IBM PC международным стандартом стала таблица кодировки под названием ASCII (American Standard Code for Information Interchange) — Американский стандартный код для информационного обмена.

Стандартными в этой таблице являются только первые 128 символов, т.е. символы с номерами от нуля (двоичный код 00000000) до 127 (01111111). Сюда входят буквы латинского алфавита, цифры, знаки препинания, скобки и некоторые другие символы. Остальные 128 кодов, начиная с 128 (двоичный код 10000000) и кончая 255 (11111111), используются для кодировки букв национальных алфавитов, символов псевдографики и научных символов (например, символы  $>$ ,  $<$  или  $\pm$ ). В русских национальных кодировках в этой части таблицы размещаются символы русского алфавита.

Принцип последовательного кодирования алфавита: в кодовой таблице ASCII латинские буквы (прописные и строчные) располагаются в алфавитном порядке. Расположение цифр также упорядочено по возрастанию значений. Данное правило соблюдается и в других таблицах кодировки. Благодаря этому и в машинном представлении для символьной информации сохраняется понятие «алфавитный порядок».

В Приложении 2, в 1-й таблице приведена стандартная часть (управляющие коды - от 00 до 31, в данную таблицу не включены) кода ASCII. Во 2-й таблице дан фрагмент альтернативной части кода ASCII, содержащий буквы русского алфавита. Здесь в первой колонке - десятичный номер символа, во второй колонке - символ, в третьей - двоичный код.

Пример 2. Буква «i» в таблице кодировки символов имеет десятичный код 105. Что зашифровано последовательностью десятичных кодов: 108 105 110 107?

Решение. При расшифровке данной последовательности кодов не нужно обращаться к таблице кодировки символов. Необходимо учесть принцип последовательного кодирования алфавитов и вспомнить порядок букв в латинском алфавите - ... i, j, k, l, m, n, o .... Буква «j» будет иметь код 106, «k» — код 107 и т.д. Следовательно, закодировано слово «link».

Пример 3. С помощью последовательности десятичных кодов: 99 111 109 112 117 116 101 114 зашифровано слово «computer». Какая последовательность десятичных кодов будет соответствовать этому же слову, записанному заглавными буквами?

Решение. При шифровке слова не обязательно пользоваться таблицей кодировки символов. Необходимо лишь учесть, что разница между десятичным кодом строчной буквы латинского алфавита и десятичным кодом соответствующей заглавной буквы равна 32. Если букве «с» соответствует код 99, то заглавная буква «С» имеет десятичный код  $67=99-32$ . Следовательно, слову «COMPUTER» соответствует последовательность кодов: 67 79 77 80 85 84 69 82.

Текстовая информация, хранящаяся в памяти компьютера в двоичном коде, из-за своей многозначности неудобна для восприятия человеком. На практике внутреннее представление чаще всего перекодируется в шестнадцатеричную форму. Шестнадцатеричный код каждого символа - двузначное число от 00 до FF.

Пример 4. Последовательность двоичных кодов: 01110011 01110100 01101111 01110000 соответствует слову «stop». Построить внутреннее шестнадцатеричное представление этого слова.

Решение. Необходимо учесть, что каждая шестнадцатеричная цифра представима четырехзначным двоичным числом, т.е. двоичному коду 01110011 будут соответствовать две шестнадцатеричные цифры 7 (0111) и 3 (0011). Следовательно, шестнадцатеричный код будет иметь вид: 73 74 6F 70.

Задание:

1. Текст занимает 0,25 Кбайт памяти компьютера. Сколько символов содержит этот текст?
2. Текст занимает полных 5 страниц. На каждой странице размещается 30 строк по 70 символов в строке. Какой объем оперативной памяти (в байтах) займет этот текст?
3. Свободный объем оперативной памяти компьютера 640 Кбайт. Сколько страниц книги поместится в ней, если на странице:  
32 строки по 64 символа в строке;  
64 строки по 64 символа в строке;  
16 строк по 64 символа в строке?
4. Текст занимает полных 10 секторов на дискете объемом 180 Кбайт. Дискета разбита на 40 дорожек по 9 секторов. Сколько символов содержит текст?
5. Десятичный код (номер) буквы «е» в таблице кодировки символов ASCII равен 101. Какая последовательность десятичных кодов будет соответствовать слову 1) file; 2) help?
6. Десятичный код (номер) буквы «о» в таблице кодировки символов ASCII равен 111. Что зашифровано с помощью последовательности десятичных кодов: 1) 115 112 111 114 116; 2) 109 111 117 115 101?
7. Десятичный код (номер) буквы «i» в таблице кодировки символов ASCII равен 105. Какая последовательность десятичных кодов будет соответствовать слову INFORMATION, записанному заглавными буквами?
8. С помощью последовательности десятичных кодов: 66 65 83 73 67 зашифровано слово BASIC. Какая последовательность десятичных кодов будет соответствовать этому слову, записанному строчными буквами?
9. Пользуясь таблицей кодировки символов ASCII, закодируйте с помощью шестнадцатеричных кодов следующий текст: 1) Norton Commander; 2) Computer IBM PC.
10. Пользуясь таблицей кодировки символов ASCII, расшифруйте текст, представленный в виде шестнадцатеричных кодов символов:  
57 69 6E 64 6F 77 73 2D 39 35;  
63 6F 6D 65 2D 4F 4E 2D 6C 69 6E 65.
11. Пользуясь таблицей кодировки символов ASCII, закодируйте с помощью двоичных кодов следующие слова: 1) EXCEL; 2) Word.
12. По шестнадцатеричному коду восстановить двоичный код и, пользуясь таблицей кодировки символов, расшифровать слово:  
1) 42 61 73 69 63; 2) 50 61 73 63 61 6C.
13. По шестнадцатеричному коду восстановить десятичный код и, пользуясь таблицей кодировки символов, расшифровать слово:  
8A 8E 8C 8F 9C 9E 92 85 90.
14. Пользуясь таблицей кодировки символов, получить шестнадцатеричный код слова ИНФОРМАТИКА.
15. Сколько букв и какого алфавита содержит зашифрованный десятичными кодами текст: 128 32 139 32 148 32 128 32 130 32 136 32 146?

Содержание отчета

- Название работы;
- Цель работы;
- Формулировку задания;
- Решенные задания, с пояснением решения в виде таблиц, рассуждений, рисунков.



### Контрольные вопросы

1. В чем состоит принцип последовательного кодирования?
2. В чем состоят особенности кодовой таблицы ASCII?
3. Какие способы кодирования символьной информации существуют?

### Список литературы:

#### Основная литература

1. Информатика. Задачник – практикум в 2 т./ Под ред. И. Г. Семакина, Е. К. Хеннера: Том 1 – М.: Бином. Лаборатория знаний, 2003.-304с.
2. Максимов Н.В., Партыка Т. Л. Попов И. И. Архитектура ЭВМ и вычислительных систем:учебник. – М.:Форум, 2008.
3. Келим Ю.М. Вычислительная техника: учеб. Пособие для студ.сред.проф.образования/ Ю. М. Келим. – М.: Издательский центр «Академия», 2008.

#### Дополнительная литература

4. Таненбаум Э. Архитектура компьютера – СПб: Питер, 2010

## Практическая работа № 2

Тема: Кодирование числовой информации. Системы счисления. Правила десятичной арифметики.

Цель: Закрепление умений перевода из одной системы счисления в другую, вычислений с помощью двоичной арифметики.

Пояснения к работе:

Система счисления — это способ представления чисел и соответствующие ему правила действия над числами. Разнообразные системы счисления, которые существовали раньше и которые используются в наше время, можно разделить на *непозиционные* и *позиционные*. Знаки, используемые при записи чисел, называются *цифрами*.

В *непозиционных системах счисления* от положения цифры в записи числа не зависит величина, которую она обозначает. Примером непозиционной системы счисления является римская система (римские цифры).

В *позиционных системах счисления* величина, обозначаемая цифрой в записи числа, зависит от ее позиции.

Количество используемых цифр называется *основанием* позиционной системы счисления. Система счисления, применяемая в современной математике, является позиционной десятичной системой. Ее основание равно десяти, т. к. запись любых чисел производится с помощью десяти цифр: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

Если требуется указать основание системы, к которой относится число, то оно приписывается нижним индексом к этому числу. Например:  $101101_2$ ,  $3671_8$ ,  $3B8F_{16}$ .

В системе счисления с основанием  $q$  ( $q$ -ичная система счисления) единицами разрядов служат последовательные степени числа  $q$ .  $q$  единиц какого-либо разряда образуют единицу следующего разряда. Для записи числа в  $q$ -ичной системе счисления требуется  $q$  различных знаков (цифр), изображающих числа 0, 1, ...,  $q-1$ .

*Развернутой формой* записи числа называется запись в виде

$$A_q = \pm(a_{n-1} \cdot q^{n-1} + a_{n-2} \cdot q^{n-2} + \dots + a_0 q^0 + a_{-1} q^{-1} + a_{-2} q^{-2} + \dots + a_{-m} q^{-m}).$$

Здесь  $A_q$  — само число,  $q$  — основание системы счисления,  $a_i$  — цифры данной системы счисления,  $n$  — число разрядов целой части числа,  $m$  — число разрядов дробной части числа.

Пример 1. Получить развернутую форму чисел  $112_3$ ,  $15FC_{16}$ ,  $101,11_2$ .

$$112_3 = 1 \times 10^2 + 1 \times 10^1 + 2 \times 10^0.$$

$$15FC_{16} = 1 \times 10^3 + 5 \times 10^2 + F \times 10^1 + C.$$

$$101,11_2 = 1 \times 10^{10} + 0 \times 10^1 + 1 \times 10^0 + 1 \times 10^{-1} + 1 \times 10^{-10}.$$

Обратите внимание, что в любой системе счисления ее основание записывается как 10.

Если все слагаемые в развернутой форме десятичного числа представить в десятичной системе и вычислить полученное выражение по правилам десятичной арифметики, то получится число в десятичной системе, равное данному. По этому принципу производится перевод из десятичной системы в десятичную.

**Пример 2.** Все числа из предыдущего примера перевести в десятичную систему.

$$112_3 = 1 \times 3^2 + 1 \times 3^1 + 2 \times 3^0 = 9 + 3 + 2 = 14_{10}.$$

$$15FC_{16} = 1 \times 16^3 + 5 \times 16^2 + 15 \times 16^1 + 12 = 4096 + 1280 + 240 + 12 = 5628_{10}.$$

$$101,11_2 = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} = 4 + 1 + 1/2 + 1/4 = 5 + 0,5 + 0,25 = 5,75_{10}.$$

*Почему люди пользуются десятичной системой, а компьютеры — двоичной?*

Люди предпочитают десятичную систему, вероятно, потому, что с древних времен считали по пальцам, а пальцев у людей по десять на руках и ногах.

А компьютеры используют двоичную систему потому, что она имеет ряд преимуществ перед другими системами:

- для ее реализации нужны *технические устройства с двумя устойчивыми состояниями* (есть ток — нет тока, намагничен — не намагничен и т.п);
- представление информации посредством только двух состояний *надежно и помехоустойчиво*;
- возможно *применение аппарата булевой алгебры* для выполнения логических преобразований информации;
- двоичная арифметика намного проще десятичной.

*Какие системы счисления используют специалисты для общения с компьютером?*

Кроме десятичной широко используются системы с основанием, являющимся целой степенью числа 2, а именно:

- двоичная (используются цифры 0, 1);
- восьмеричная (используются цифры 0, 1, ..., 7);
- шестнадцатеричная (для первых целых чисел от нуля до девяти используются цифры 0, 1, ..., 9, а для следующих чисел — от десяти до пятнадцати — в качестве цифр используются символы A, B, C, D, E, F).

Полезно запомнить запись в этих системах счисления первых двух десятков целых чисел:

10-я	2-я	8-я	16-я
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9

10-я	2-я	8-я	16-я
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F
16	10000	20	10
17	10001	21	11
18	10010	22	12
19	10011	23	13

1. Для того чтобы произвольное двоичное число записать в системе счисления с основанием  $q=2^n$  (4, 8, 16 и т. д.), нужно:

- 1) данное двоичное число разбить слева и справа (целую и дробную части) на группы по  $n$  цифр в каждой;
- 2) если в последних правой и левой группах окажется меньше  $n$  разрядов, то их надо дополнить справа и слева нулями до нужного числа разрядов;

3) рассмотреть каждую группу как  $n$ -разрядное двоичное число и записать ее соответствующей цифрой в системе счисления с основанием  $q = 2^n$ .

2. Для того чтобы произвольное число, записанное в системе счисления с основанием  $q = 2^n$ , перевести в двоичную систему счисления, нужно каждую цифру этого числа заменить ее  $n$ -разрядным эквивалентом в двоичной системе счисления.

Пример 3. Перевести число  $11101,1011101_2$  в восьмеричную и в шестнадцатеричную системы счисления.

$$\begin{array}{rcccccc}
 11101,1011101_2 = & 011 & 101, & 101 & 110 & 100 & = 35,564_8 \\
 & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \\
 & 3 & 5, & 5 & 6 & 4 & \\
 11101,1011101_2 = & 0001 & 1101, & 1011 & 1010 & & = 1D,BA_{16} \\
 & \downarrow & \downarrow & \downarrow & \downarrow & & \\
 & 1 & D, & B & A & & 
 \end{array}$$

Пример 4. Перевести число  $15FC_{16}$  в двоичную систему. Для решения задачи воспользуемся приведенной таблицей.

$$\begin{array}{cccccc}
 1 & 5 & F & C, & 2 & A \\
 \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\
 0001 & 0101 & 1111 & 1100, & 0010 & 1010
 \end{array}$$

$$\text{Т.е. } 15FC_{16} = 101011111100,00101010_2$$

*Перевод десятичных чисел в другие системы счисления:*

3. *Перевод целых чисел.*

- 1) Основание новой системы счисления выразить в десятичной системе счисления и все последующие действия производить в десятичной системе счисления;
- 2) последовательно выполнять деление данного числа и получаемых неполных частных на основание новой системы счисления до тех пор, пока не получим неполное частное, меньшее делителя;
- 3) полученные остатки, являющиеся цифрами числа в новой системе счисления, привести в соответствие с алфавитом новой системы счисления;
- 4) составить число в новой системе счисления, записывая его, начиная с последнего частного.

Пример 5. Перевести число  $75_{10}$  в двоичную, восьмеричную и шестнадцатеричную системы счисления. Для обозначения цифр в записи числа, используем символику:  $a_n a_{n-1} \dots a_1 a_0$

$$\begin{array}{r}
 \begin{array}{r}
 75 \overline{) 2} \\
 74 \overline{) 37} \quad 2 \\
 \hline
 a_0=1 \quad 36 \overline{) 18} \quad 2 \\
 \hline
 a_1=1 \quad 18 \overline{) 9} \quad 2 \\
 \hline
 a_2=0 \quad 8 \overline{) 4} \quad 2 \\
 \hline
 a_3=1 \quad 4 \overline{) 2} \quad 2 \\
 \hline
 a_4=0 \quad 2 \overline{) 1} \quad 2 \\
 \hline
 a_5=0
 \end{array}
 \end{array}$$

$$\text{Отсюда } 75_{10} = 1001011_2 = 113_8 = 4B_{16}$$

4. *Перевод дробных чисел.*

- 1) основание новой системы счисления выразить в десятичной системе и все последующие действия производить в десятичной системе счисления;
- 2) последовательно умножать данное число и получаемые дробные части произведений на основание новой системы до тех пор, пока дробная часть произведения не станет равной нулю или не будет достигнута требуемая точность представления числа в новой системе счисления;
- 3) полученные целые части произведений, являющиеся цифрами числа в новой системе счисления, привести в соответствие с алфавитом новой системы счисления;

- 4) составить дробную часть числа в новой системе счисления, начиная с целой части первого произведения.

**Пример 6.** Перевести десятичную дробь 0,1875 в двоичную, восьмеричную и шестнадцатеричную системы счисления.

0	1	8	7	5	0	1	8	7	5	0	1	8	7	5
x				2	x				8	x			1	6
0	3	7	5	0	1	5	0	0	0	3	0	0	0	0
x				2	x				8					
0	7	5	0	0	4	0	0	0	0					
x				2										
1	5	0	0	0										
x				2										
1	0	0	0	0										

Здесь вертикальная черта отделяет целые части чисел от дробных частей. Отсюда:  $0,1875_{10} = 0,0011_2 = 0,14_8 = 0,3_{16}$ .

5. **Перевод смешанных чисел**, содержащих целую и дробную части, осуществляется в два этапа. Целая и дробная части исходного числа переводятся отдельно по соответствующим алгоритмам. В итоговой записи числа в новой системе счисления целая часть отделяется от дробной запятой (точкой).

**Пример 7.** Перевести десятичное число 75,1875 в восьмеричную и в шестнадцатеричную системы счисления.

Из рассмотренных выше примеров следует:

$$75,1875_{10} = 113,14_8 = 4B,3_{16}.$$

Задание:

1. Запишите числа в развернутом виде;
2. Переведите числа в десятичную систему;
3. Проверьте результаты, выполнив обратные переводы;
4. Переведите заданное двоичное число в восьмеричную и шестнадцатеричную системы счисления;
5. Переведите заданное восьмеричное число в двоичную систему счисления;
6. Переведите заданное шестнадцатеричное число в двоичную систему счисления.

1) 1010011 <sub>2</sub>	557 <sub>8</sub>	ABC <sub>16</sub>
2) 1001011 <sub>2</sub>	234 <sub>8</sub>	1FB <sub>16</sub>
3) 1001011 <sub>2</sub>	157 <sub>8</sub>	43A <sub>16</sub>
4) 1110111 <sub>2</sub>	356 <sub>8</sub>	9BE <sub>16</sub>
5) 1101001 <sub>2</sub>	443 <sub>8</sub>	8EC <sub>16</sub>
6) 1110111 <sub>2</sub>	655 <sub>8</sub>	7AD <sub>16</sub>
7) 1010001 <sub>2</sub>	416 <sub>8</sub>	6EC <sub>16</sub>
8) 1000001 <sub>2</sub>	712 <sub>8</sub>	54F <sub>16</sub>
9) 1010111 <sub>2</sub>	507 <sub>8</sub>	D64 <sub>16</sub>
10) 1101011 <sub>2</sub>	301 <sub>8</sub>	8AB <sub>16</sub>

Содержание отчета

- Название работы;
- Цель работы;
- Формулировку задания;
- Решенные задания, с пояснением решения в виде таблиц, рассуждений, рисунков.

Контрольные вопросы

1. Чем отличается позиционная система счисления от непозиционной?
2. Сформулируйте правило перевода целого двоичного числа в восьмеричную систему счисления.
3. Сформулируйте правило перевода дробного двоичного числа в шестнадцатеричную систему счисления.

счисления.

4. Приведите пример перевода смешанного десятичного числа в любую другую позиционную систему счисления.

Список литературы:

Основная литература

1. Информатика. Задачник – практикум в 2 т./ Под ред. И. Г. Семакина, Е. К. Хеннера: Том 1 – М.: Бинوم. Лаборатория знаний, 2003.-304с.
2. Максимов Н.В., Партыка Т. Л. Попов И. И. Архитектура ЭВМ и вычислительных систем:учебник. – М.:Форум, 2008.
3. Келим Ю.М. Вычислительная техника: учеб. Пособие для студ.сред.проф.образования/ Ю. М. Келим. – М.: Издательский центр «Академия», 2008.

Дополнительная литература

4. Таненбаум Э. Архитектура компьютера – СПб: Питер, 2010

### Практическая работа № 3

Тема: Кодирование числовой информации. Представление целых чисел в ЭВМ.

Цель: Закрепление знаний о представлении целых чисел в памяти ЭВМ.

Пояснения к работе:

Основные структурные единицы памяти компьютера: бит, байт, машинное слово.

*Бит.* Все данные и программы, хранящиеся в памяти компьютера, имеют вид двоичного кода. Один символ из двухсимвольного алфавита несет 1 бит информации. Ячейка памяти, хранящая один двоичный знак, называется «бит». В одном бите памяти хранится один бит информации.

Битовая структура памяти определяет первое свойство памяти - дискретность.

Байт. Восемь расположенных подряд битов памяти образуют байт. В одном байте памяти хранится один байт информации. Во внутренней памяти компьютера все байты пронумерованы. Нумерация начинается от нуля. Порядковый номер байта называется его адресом. В компьютере адреса обозначаются двоичным кодом. Используется также шестнадцатеричная форма обозначения адреса.

Целые числа могут представляться в компьютере со знаком или без знака.

*Целые числа без знака*

Обычно занимают в памяти компьютера один или два байта. В однобайтовом формате принимают значения от  $00000000_2$  до  $11111111_2$ . В двухбайтовом формате — от  $00000000\ 00000000_2$  до  $11111111\ 11111111_2$ .

*Диапазоны значений целых чисел без знака*

Формат числа в байтах	Диапазон	
	Запись с порядком	Обычная запись
1	$0 \dots 2^8 - 1$	0 ... 255
2	$0 \dots 2^{16} - 1$	0 ... 65535

*Пример1.* Представьте число  $72_{10}$  в однобайтовом формате

Число  $72_{10} = 1001000_2$

**Номера разрядов**  
**Биты числа**

7	6	5	4	3	2	1	0
0	1	0	0	1	0	0	0

*Пример2.* Представьте число  $72_{10}$  в двухбайтовом формате.

**Номера разрядов**  
**Биты числа**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0

*Целые числа со знаком*

Обычно занимают в памяти компьютера один, два или четыре байта, при этом самый левый (старший) разряд содержит информацию о знаке числа.

*Диапазоны значений целых чисел со знаком*

Формат числа в байтах	Диапазон	
	Запись с порядком	Обычная запись
1	$-2^7 \dots 2^7 - 1$	$-128 \dots 127$
2	$-2^{15} \dots 2^{15} - 1$	$-32768 \dots 32767$
4	$-2^{31} \dots 2^{31} - 1$	$-2147483648 \dots 2147483647$

Рассмотрим особенности записи целых чисел со знаком на примере *однобайтового формата*, при котором для знака отводится один разряд, а для цифр абсолютной величины – семь разрядов.

В компьютерной технике применяются три формы записи (кодирования) целых чисел со знаком: прямой код, обратный код, дополнительный код.

Последние две формы применяются особенно широко, так как позволяют упростить конструкцию арифметико-логического устройства компьютера путем замены разнообразных арифметических операций операцией сложения.

*Положительные числа* в прямом, обратном и дополнительном кодах изображаются одинаково — двоичными кодами с цифрой 0 в знаковом разряде.

Пример 3. Представьте положительные числа  $1_{10}$  и  $127_{10}$  в прямом, обратном и дополнительном кодах.

Число  $1_{10} = 1_2$

0 0 0 0 0 0 0 1

Знак числа "+"

Число  $127_{10} = 1111111_2$

0 1 1 1 1 1 1 1

Знак числа "+"

*Отрицательные числа* в прямом, обратном и дополнительном кодах имеют разное изображение.

1. *Прямой код.* В знаковый разряд помещается цифра 1, а в разряды цифровой части числа — двоичный код его абсолютной величины.

2. *Обратный код.* Получается инвертированием всех цифр двоичного кода абсолютной величины числа, включая разряд знака: нули заменяются единицами, а единицы — нулями.

3. *Дополнительный код.* Получается образованием обратного кода с последующим прибавлением единицы к его младшему разряду. (Правило выполнения сложения в двоичной системе счисления:  $0+0=0$ ,  $0+1=1$ ,  $1+0=1$ ,  $1+1=10$ )

Пример 4. Вычислите значения выражения  $100_2 + 1_2$ ,  $101_2 + 1_2$ ,  $111_2 + 1_2$ .

$$\begin{array}{r}
 1 \ 0 \ 0 \\
 + \quad 1 \\
 \hline
 1 \ 0 \ 1
 \end{array}
 \qquad
 \begin{array}{r}
 \phantom{1} +1 \\
 1 \ 0 \ 1 \\
 + \quad 1 \\
 \hline
 1 \ 1 \ 0
 \end{array}
 \qquad
 \begin{array}{r}
 \phantom{1} +1 \ +1 \\
 1 \ 1 \ 1 \\
 + \quad 1 \\
 \hline
 1 \ 0 \ 0 \ 0
 \end{array}$$

Пример 5. Представьте числа  $-1_{10}$  и  $-127_{10}$  в прямом, обратном и дополнительном кодах.

Прямой код числа - 1

1 0 0 0 0 0 0 1

Знак числа "-"

Число: -1

Код модуля числа: 0 0000001

Обратный код числа: 1 1111110

1 1 1 1 1 1 1 0

Дополнительный код числа - 1

1 1 1 1 1 1 1 1

Прямой код числа - 127

1 1 1 1 1 1 1 1

Знак числа "-"

Число: -127

Код модуля числа: 0 1111111

Обратный код числа: 1 0000000

1 0 0 0 0 0 0 0

Дополнительный код числа - 127

1 0 0 0 0 0 0 1

Обычно отрицательные десятичные числа при вводе в машину автоматически преобразуются в обратный или дополнительный двоичный код и в таком виде хранятся,

перемещаются и участвуют в операциях. При выводе таких чисел из машины происходит обратное преобразование в отрицательные десятичные числа.

Задание:

1. Представьте числа в прямом, обратном и дополнительном кодах в однобайтовом формате:

1) 73	-119	6) 83	-129
2) 75	-121	7) 85	-131
3) 77	-123	8) 87	-133
4) 79	-125	9) 89	-135
5) 81	-127	10) 91	-137

Содержание отчета

- Название работы;
- Цель работы;
- Формулировку задания;
- Решенные задания, с пояснением решения в виде таблиц, рассуждений, рисунков.

Контрольные вопросы

1. В каком виде хранятся данные, программы в памяти ЭВМ?
2. Перечислите структурные единицы памяти.
3. Сформулируйте правило представления отрицательных чисел в прямом, обратном и дополнительном кодах.
4. Сформулируйте правило представления положительных чисел в прямом, обратном и дополнительном кодах.

Список литературы:

Основная литература

1. Информатика. Задачник – практикум в 2 т./ Под ред. И. Г. Семакина, Е. К. Хеннера: Том 1 – М.: Бинوم. Лаборатория знаний, 2003.-304с.
2. Максимов Н.В., Партыка Т. Л. Попов И. И. Архитектура ЭВМ и вычислительных систем:учебник. – М.:Форум, 2008.
3. Келим Ю.М. Вычислительная техника: учеб. Пособие для студ.сред.проф.образования/ Ю. М. Келим. – М.: Издательский центр «Академия», 2008.

Дополнительная литература

4. Таненбаум Э. Архитектура компьютера – СПб: Питер, 2010

## Практическая работа № 4

Тема: Кодирование числовой информации. Представление вещественных чисел в ЭВМ.

Цель: Закрепление знаний о представлении вещественных чисел в ЭВМ.

Пояснения к работе:

Формат с плавающей точкой использует представление вещественного числа  $R$  в виде произведения мантиссы  $m$  на основание системы счисления  $p$  в некоторой целой степени  $p$ , которую называют порядком:  $R = mp^p$ .

Представление числа в форме с плавающей точкой неоднозначно. Например, справедливы следующие равенства:

$$25.324 = 2.5324 \times 10^1 - 0.0025324 \times 10^4 = 2532.4 \times 10^{-2} \text{ и т.п.}$$

В ЭВМ используют нормализованное представление числа в форме с плавающей точкой. Мантисса в нормализованном представлении должна удовлетворять условию:  $0.1_n < m < 1_n$ . Иначе говоря, мантисса меньше единицы и первая значащая цифра - не ноль.

В памяти компьютера мантисса представляется как целое число, содержащее только значащие цифры (0 целых и запятая не хранятся). Следовательно, внутреннее представление вещественного числа сводится к представлению пары целых чисел: мантиссы и порядка.

В разных типах ЭВМ применяются различные варианты представления чисел в форме с плавающей точкой. Для примера рассмотрим внутреннее представление вещественного числа в 4-х байтовой ячейке памяти.

В ячейке должна содержаться следующая информация о числе: знак числа, порядок и значащие цифры мантииссы.

± маш.порядок	М	А	Н	Т	И	С	С	А
1-й байт	2-й байт		3-й байт			4-й байт		

В старшем бите 1-го байта хранится знак числа: 0 обозначает плюс, 1 - минус. Оставшиеся 7 бит первого байта содержат машинный порядок. В следующих трех байтах хранятся значащие цифры мантииссы (24 разряда).

В семи двоичных разрядах помещаются двоичные числа в диапазоне от 0000000 до 1111111. Значит, машинный порядок изменяется в диапазоне от 0 до 127 (в десятичной системе счисления). Всего 128 значений. Порядок, очевидно, может быть как положительным так и отрицательным. Разумно эти 128 значений разделить поровну между положительными и отрицательными значениями порядка: от -64 до 63.

*Машинный порядок смещен относительно математического и имеет только положительные значения. Смещение выбирается так, чтобы минимальному математическому значению порядка соответствовал нуль.*

Связь между машинным порядком ( $M_p$ ) и математическим ( $p$ ) в рассматриваемом случае выражается формулой:

$$M_p = p + 64.$$

Полученная формула записана в десятичной системе. В двоичной системе формула имеет вид:  $M_p = p_2 + 100\ 0000_2$ .

Для записи внутреннего представления вещественного числа необходимо:

- 1) перевести модуль данного числа в двоичную систему счисления с 24 значащими цифрами;
- 2) нормализовать двоичное число;
- 3) найти машинный порядок в двоичной системе счисления;
- 4) учитывая знак числа, выписать его представление в 4-х байтовом машинном слове.

**Пример 4.** Записать внутреннее представление числа 250,1875 в форме с плавающей точкой.

**Решение.**

1. Переведем его в двоичную систему счисления с 24 значащими цифрами:  $250,1875_{10} = 11111010,0011000000000000_2$ .

2. Запишем в форме нормализованного двоичного числа с плавающей точкой:  $0,111110100011000000000000 \times 10_2^{1000}$ . Здесь мантиисса, основание системы счисления ( $2_{10} = 10_2$ ) и порядок ( $8_{10} = 1000_2$ ) записаны в двоичной системе.

3. Вычислим машинный порядок в двоичной системе счисления:  $M_p = 1000 + 100\ 0000 = 100\ 1000$ .

4. Запишем представление числа в 4-х байтовой ячейке памяти с учетом знака числа:

0	1001000	11111010	00110000	00000000
31		24	23	0
Шестнадцатеричная форма: 48FA3000.				

**Пример 5.** По шестнадцатеричной форме внутреннего представления числа в форме с плавающей точкой C9811000 восстановить само число. **Решение.**

1. Перейдем к двоичному представлению числа в 4-х байтовой ячейке, заменив каждую шестнадцатеричную цифру 4-мя двоичными цифрами:

1100 1001 1000 0001 0001 0000 0000 0000

1	1001001	10000001	00010000	00000000
31		23		0

2. Заметим, что получен код отрицательного числа, поскольку в старшем разряде с номером 31 записана 1. Получим порядок числа:  $p = 1001001_2 - 1000000_2 = 1001_2 = 9_{10}$ .

3. Запишем в форме нормализованного двоичного числа с плавающей точкой с учетом знака числа:

$-0,100000010001000000000000 \times 2^{1001}$ .

4. Число в двоичной системе счисления имеет вид:  
 $-100000010,001_2$ .



5. Переведем число в десятичную систему счисления:

$$-1000000010,001_2 = -(1 \times 2^8 + 1 \times 2^1 + 1 \times 2^{-3}) = -258,125_{10}.$$

Диапазон вещественных чисел значительно шире диапазона целых чисел. Положительные и отрицательные числа расположены симметрично относительно нуля. Следовательно, максимальное и минимальное числа равны между собой по модулю.

Наименьшее по абсолютной величине число равно нулю. Наибольшее по абсолютной величине число в форме с плавающей точкой - это число с самой большой мантиссой и самым большим порядком.

Для 4-х байтового машинного слова таким числом будет:

$$0,1111111111111111111111111111 \times 10_2^{111111}$$

После перевода в десятичную систему счисления получим:

$$(1 - 2^{-24}) \times 2^{63} \approx 10^{19}.$$

Множество вещественных чисел, представимых в памяти компьютера в форме с плавающей точкой, является ограниченным и дискретным. Количество вещественных чисел, точно представимых в памяти компьютера, вычисляется по формуле:  $N = 2^t \times (U - L + 1) + 1$ . Здесь  $t$  - количество двоичных разрядов мантииссы;  $U$  - максимальное значение математического порядка;  $L$  - минимальное значение порядка. Для рассмотренного нами варианта ( $t = 24$ ,  $U = 63$ ,  $L = -64$ ) получается:  $N = 2\ 146\ 683\ 548$ .

**Задание:**

1. Получить шестнадцатеричную форму внутреннего представления числа в формате с плавающей точкой в 4-х байтовой ячейке.
2. По шестнадцатеричной форме внутреннего представления вещественного числа в 4-х байтовой ячейке восстановить само число.

№ варианта	номера заданий	
	1	2
1	26.315	C5DB0000
2	-27.022	45D24000
3	13.333	C5ED0000
4	-77.611	47B7A000
5	87.022	C5D24000
6	-44.333	488B6000
7	51.315	C7B7A000
8	-63.611	45DB0000
9	60.509	C88B6000
10	-91.509	45ED0000
11	48.611	C6870000
12	-12.022	46870000
13	33.333	C9A6C000
14	-29.611	49A6C000
15	38.315	48E04000

## Содержание отчета

- Название работы;
- Цель работы;
- Формулировку задания;
- Решенные задания, с пояснением решения в виде таблиц, рассуждений, рисунков.

## Контрольные вопросы

1. Перечислите структурные единицы памяти.
2. Сформулируйте правило представления вещественных чисел в 4-х байтовом формате

### Список литературы:

## Основная литература

1. Информатика. Задачник – практикум в 2 т./ Под ред. И. Г. Семакина, Е. К. Хеннера: Том 1 – М.: Бином. Лаборатория знаний, 2003.-304с.
2. Максимов Н.В., Партыка Т. Л. Попов И. И. Архитектура ЭВМ и вычислительных систем:учебник. – М.:Форум, 2008.
3. Келим Ю.М. Вычислительная техника: учеб. Пособие для студ.сред.проф.образования/ Ю. М. Келим. – М.: Издательский центр «Академия», 2008.

- Дополнительная литература  
4. Таненбаум Э. Архитектура компьютера – СПб: Питер, 2010

## Практическая работа № 5

Тема: Кодирование мультимедиа информации.

Цель: формирование представлений о кодировании мультимедиа информации

Пояснения к работе:

Графическая информация может быть представлена в аналоговой и дискретной формах. Примером аналогового представления графической информации может служить живописное полотно, цвет которого изменяется непрерывно, а дискретного - изображение, напечатанное с помощью струйного принтера и состоящее из отдельных точек разного цвета.

Графические изображения из аналоговой (непрерывной) формы в цифровую (дискретную) преобразуются путем пространственной дискретизации. Пространственную дискретизацию изображения можно сравнить с построением изображения из мозаики (большого количества маленьких разноцветных стекол). Изображение разбивается на отдельные маленькие элементы (точки, или пиксели), причем каждый элемент может иметь свой цвет (красный, зеленый, синий и т.д.).

Пиксель - минимальный участок изображения, для которого независимым образом можно задать цвет.

В результате пространственной дискретизации графическая информация представляется в виде растрового изображения, которое формируется из определенного количества строк, содержащих, в свою очередь, определенное количество точек.

Разрешающая способность. Важнейшей характеристикой качества растрового изображения является разрешающая способность.

Разрешающая способность растрового изображения определяется количеством точек как по горизонтали, так и по вертикали на единицу длины изображения.

Чем меньше размер точки, тем больше разрешающая способность (больше строк раstra и точек в строке), и, соответственно, выше качество изображения. Величина разрешающей способности обычно выражается в dpi (dot per inch - точек на дюйм), т.е. в количестве точек в полоске изображения длиной один дюйм (1 дюйм = 2,54 см.)

Пространственная дискретизация непрерывных изображений, хранящихся на бумаге, фото- и киноплёнке, может быть осуществлена путем сканирования. В настоящее время все большее распространение получают цифровые фото- и видеокамеры, которые фиксируют изображения сразу в дискретной форме.

Глубина цвета. В процессе дискретизации могут использоваться различные палитры цветов, т.е. наборы цветов, в которые могут быть окрашены точки изображения. Каждый цвет можно рассматривать как возможное состояние точки, тогда количество цветов в палитре N и количество информации, необходимое для кодирования каждой точки I, связаны между собой и могут быть вычислены по формуле:

$$N=2^I \quad (1.1)$$

В простейшем случае (черно - белое изображение без градаций серого цвета) палитра цветов состоит всего из двух цветов (черного и белого). Каждая точка экрана может принимать одно из двух состояний - "черная" или "белая", следовательно, по формуле (1.1) можно вычислить, какое количество информации необходимо, чтобы закодировать цвет каждой точки:

$$2=2^I \rightarrow 2^1=2^I \rightarrow I=1 \text{ бит.}$$

Количество информации, которое используется для кодирования цвета точки изображения, называется глубиной цвета.

Наиболее распространенными значениями глубины цвета при кодировании цветных изображений являются 4, 8, 16 или 24 бита на точку. Зная глубину цвета, по формуле (1.1) можно вычислить количество цветов в палитре (табл. 1.1).

Таблица 1.1

Глубина цвета, I (битов)	Количество цветов в палитре, N
--------------------------	--------------------------------

4	$2^4=16$
8	$2^8=256$
16	$2^{16}=65\ 536$
24	$2^{24}=16\ 777\ 216$

Графические режимы монитора. Качество изображения на экране монитора зависит от величины пространственного разрешения и глубины цвета.

Пространственное разрешение экрана монитора определяется как произведение количества строк изображения на количество точек в строке. Монитор может отображать информацию с различными пространственными разрешениями (800 x 600, 1024 x 768, 1152 x 864 и выше).

Глубина цвета измеряется в битах на точку и характеризует количество цветов, в которые могут быть окрашены точки изображения. Количество отображаемых цветов также может изменяться в широком диапазоне, от 256 (глубина цвета 8 битов) до более чем 16 миллионов (глубина цвета 24 бита).

Чем больше пространственное разрешение и глубина цвета, тем выше качество изображения.

В операционных системах предусмотрена возможность выбора необходимого пользователю и технически возможного графического режима.

Рассмотрим формирование на экране монитора растрового изображения, состоящего из 600 строк по 800 точек в каждой строке (всего 480 000 точек) и глубиной цвета 8 битов. Двоичный код цвета всех точек хранится в видеопамяти компьютера (рис. 1), которая находится на видеокарте (рис. 2).

Видеопамять	
№ точки	Двоичный код цвета точки
1	01010101
2	10101010
...	...
800	11110000
...	...
480000	11111111



Рис. 1. Формирование растрового

изображения на экране монитора

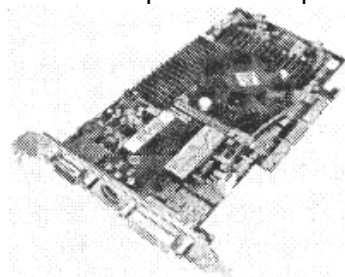


Рис. 2. Видеокарта

Видеокарта устанавливается в слот расширения системной платы PCI или AGP. Монитор подключается к аналоговому выходу VGA или цифровому выходу DVI видеокарты.

Периодически, с определенной частотой, коды цветов точек вчитываются из видеопамяти точки отображаются на экране монитора. Частота считывания изображения влияет на стабильность изображения на экране. В современных мониторах обновление изображения происходит с частотой 75 и более раз в секунду, что обеспечивает комфортность восприятия изображения пользователем компьютера (человек не замечает мерцания изображения). Для сравнения можно напомнить, что частота смены кадров в кино составляет 24 кадра в секунду.

Объем видеопамяти. Информационный объем требуемой видеопамяти можно рассчитать по формуле:

$$I_n = I \times X \times Y,$$

где  $I_n$  - информационный объем видеопамяти в битах;

$X \times Y$  - количество точек изображения ( $X$  - количество точек по горизонтали,  $Y$  - по вертикали);

$I$  - глубина цвета в битах на точку.

*Пример:* необходимый объем видеопамяти для графического режима с пространственным разрешением 800 x 600 точек и глубиной цвета 24 бита равен:

$$I_n = I \times X \times Y = 24 \text{ бита} \times 800 \times 600 = 11\ 520\ 000 \text{ бит} = 1\ 440\ 000 \text{ байт} = 1\ 406,25 \text{ Кбайт} = 1,37 \text{ Мбайт}.$$

Качество отображения информации на экране монитора зависит от размера экрана и размера пикселя. Зная размер диагонали экрана в дюймах (15", 17" и т. д.) и размер пикселя экрана (0,28 мм, 0,24 мм или 0,20 мм), можно оценить максимально возможное пространственное разрешение экрана монитора.

Палитры цветов.

Система цветопередачи RGB (Red - красный, Green - зелёный, Blue - синий).

RGB – излучаемый цвет, используется в мониторах. Наибольшая чувствительность колбочек сетчатки человеческого глаза приходится на красный, зелёный и синий цвета.

Зелёный + Синий = Голубой

Красный + Синий = Пурпурный

Красный + Зелёный = Жёлтый

Красный + Зелёный + Синий = Белый

	Cyan	Magenta	Yellow
Red	0	+	+
Green	+	0	+
Blue	+	+	0

Система цветопередачи CMYK (Cyan- голубой, Magenta - пурпурный, Yellow - жёлтый) – обратная RGB.

CMYK – отображаемый цвет, используется при печати.

Пурпурный + Жёлтый = Красный

Голубой + Жёлтый = Зелёный

Пурпурный + Голубой = Синий

Пурпурный + Жёлтый + Голубой + Чёрный = Чёрный

	Red	Green	Blue
Cyan	0	+	+
Magenta	+	0	+
Yellow	+	+	0

Система цветопередачи HSB (Hue – оттенок цвета, Saturation - насыщенность, Brightness - яркость)

Используется в графических редакторах при настройке палитры цвета.

Параметр Hue позволяет выбрать оттенок цвета из всех цветов оптического спектра, начиная от красного цвета и кончая фиолетовым.

H=0	H=120	H=240	H=360
красный	зелёный	синий	фиолетовый

Параметр Saturation определяет процент «чистого» оттенка и белого цвета: S=0 – белый цвет, S=100 – «чистый» оттенок цвета.

Параметр Brightness определяет интенсивность цвета: B=0 – чёрный цвет, B=100 – максимальная яркость выбранного оттенка цвета.

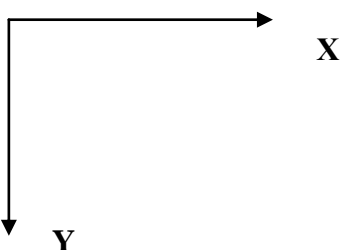
Глубина цвета:

- Монохромный (чёрно-белый) – 1 бит
- Градации серого – 8 бит
- Индексированный – 8 бит
- True Color – 24 бита
- High Color – 16 бит

При векторном подходе изображение рассматривается как описание графических примитивов, прямых, дуг, эллипсов, прямоугольников, окружностей, закрасок и пр.

Описываются положение и форма этих примитивов в системе графических координат.

Таким образом векторное изображение кодируется векторными командами, т.е описывается с помощью алгоритма. Отрезок прямой линии определяется координатами его концов, окружность – координатами центра и радиусом, многоугольник – координатами его углов, закрашенная область - линией границы и цветом закрашки.



Команда	Действие
---------	----------

Линия к X1, Y1	Нарисовать линию от текущей позиции в позицию (X1, Y1).
Линия X1, Y1, X2,Y2	Нарисовать линию с координатами начала X1, Y1 и координатами конца X2, Y2. Текущая позиция не устанавливается.
Окружность X,Y,R	Нарисовать окружность; X,Y – координаты центра, а R – длина радиуса.
Эллипс X1, Y1, X2,Y2	Нарисовать эллипс, ограниченный прямоугольником; (X1, Y1) – координаты левого верхнего, а (X2,Y2) – правого нижнего угла прямоугольника.
Прямоугольник X1, Y1, X2,Y2	Нарисовать прямоугольник; (X1, Y1)- координаты левого верхнего угла, (X2,Y2) - координаты правого нижнего угла прямоугольника.
Цвет рисования Цвет	Установить текущий цвет рисования.
Цвет закрашки Цвет	Установить текущий цвет закрашки
Закрасить X,Y, ЦВЕТ ГРАНИЦЫ	Закрасить произвольную <u>замкнутую</u> фигуру; X, Y – координаты любой точки внутри замкнутой фигуры, ЦВЕТ ГРАНИЦЫ – цвет граничной линии.

Кодирование и обработка звуковой информации

Звуковая информация. Звук представляет собой распространяющуюся в воздухе, воде или другой среде волну с непрерывно меняющейся интенсивностью и частотой.

Человек воспринимает звуковые волны (колебания воздуха) с помощью слуха в форме звука различных громкости и тона. Чем больше интенсивность звуковой волны, тем громче звук, чем больше частота волны, тем выше тон звука (рис. 1).



Рис. 1. Зависимость громкости и высоты тона звука от

интенсивности и частоты звуковой волны

Человеческое ухо воспринимает звук с частотой от 20 колебаний в секунду (низкий звук) до 20 000 колебаний в секунду (высокий звук).

Человек может воспринимать звук в огромном диапазоне интенсивностей, в котором максимальная интенсивность больше минимальной в  $10^{14}$  раз (в сто тысяч миллиардов раз). Для измерения громкости звука применяется специальная единица "децибел" (дБл) (табл. 1). Уменьшение или увеличение громкости звука на 10 дБл соответствует уменьшению или увеличению интенсивности звука в 10 раз.

Таблица 1. Громкость звука

Звук	Громкость в децибелах
Нижний предел чувствительности человеческого уха	0
Шорох листьев	10
Разговор	60
Гудок автомобиля	90
Реактивный двигатель	120
Болевой порог	140

Временная дискретизация звука. Для того чтобы компьютер мог обрабатывать звук, непрерывный звуковой сигнал должен быть преобразован в цифровую дискретную форму с помощью временной дискретизации. Непрерывная звуковая волна разбивается на отдельные

маленькие временные участки, для каждого такого участка устанавливается определенная величина интенсивности звука.

Таким образом, непрерывная зависимость громкости звука от времени  $A(t)$  заменяется на дискретную последовательность уровней громкости. На графике это выглядит как замена гладкой кривой на последовательность "ступенек" (рис. 2).

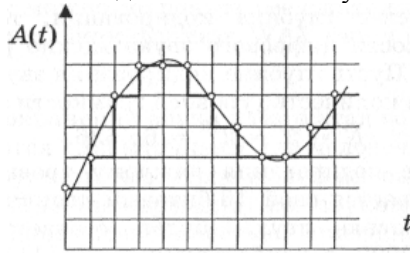


Рис. 2. Временная дискретизация звука

**Частота дискретизации.** Для записи аналогового звука и его преобразования в цифровую форму используется микрофон, подключенный к звуковой плате. Качество полученного цифрового звука зависит от количества измерений уровня громкости звука в единицу времени, т. е. частоты дискретизации. Чем большее количество измерений производится за 1 секунду (чем больше частота дискретизации), тем точнее "лесенка" цифрового звукового сигнала повторяет кривую диалогового сигнала.

*Частота дискретизации звука - это количество измерений громкости звука за одну секунду.*

Частота дискретизации звука может лежать в диапазоне от 8000 до 48 000 измерений громкости звука за одну секунду.

**Глубина кодирования звука.** Каждой "ступеньке" присваивается определенное значение уровня громкости звука. Уровни громкости звука можно рассматривать как набор возможных состояний  $N$ , для кодирования которых необходимо определенное количество информации  $I$ , которое называется глубиной кодирования звука.

*Глубина кодирования звука - это количество информации, которое необходимо для кодирования дискретных уровней громкости цифрового звука.*

Если известна глубина кодирования, то количество уровней громкости цифрового звука можно рассчитать по формуле  $N = 2^I$ . Пусть глубина кодирования звука составляет 16 битов, тогда количество уровней громкости звука равно:

$$N = 2^I = 2^{16} = 65\,536.$$

В процессе кодирования каждому уровню громкости звука присваивается свой 16-битовый двоичный код, наименьшему уровню звука будет соответствовать код 0000000000000000, а наибольшему - 1111111111111111.

**Качество оцифрованного звука.** Чем больше частота и глубина дискретизации звука, тем более качественным будет звучание оцифрованного звука. Самое низкое качество оцифрованного звука, соответствующее качеству телефонной связи, получается при частоте дискретизации 8000 раз в секунду, глубине дискретизации 8 битов и записи одной звуковой дорожки (режим "моно"). Самое высокое качество оцифрованного звука, соответствующее качеству аудио-CD, достигается при частоте дискретизации 48 000 раз в секунду, глубине дискретизации 16 битов и записи двух звуковых дорожек (режим "стерео").

Необходимо помнить, что чем выше качество цифрового звука, тем больше информационный объем звукового файла. Можно оценить информационный объем цифрового стереозвукового файла длительностью звучания 1 секунда при среднем качестве звука (16 битов, 24 000 измерений в секунду). Для этого глубину кодирования необходимо умножить на количество измерений в 1 секунду и умножить на 2 (стереозвук):

$$16 \text{ бит} \times 24\,000 \times 2 = 768\,000 \text{ бит} = 96\,000 \text{ байт} = 93,75 \text{ Кбайт}.$$

**Звуковые редакторы.** Звуковые редакторы позволяют не только записывать и воспроизводить звук, но и редактировать его. Оцифрованный звук представляется в звуковых редакторах в наглядной форме, поэтому операции копирования, перемещения и удаления частей звуковой дорожки можно легко осуществлять с помощью мыши. Кроме того, можно накладывать звуковые дорожки друг на друга (микшировать звуки) и применять различные акустические эффекты (эхо, воспроизведение в обратном направлении и др.).

Звуковые редакторы позволяют изменять качество цифрового звука и объем звукового файла путем изменения частоты дискретизации и глубины кодирования. Оцифрованный звук можно сохранять без сжатия в звуковых файлах в универсальном формате WAV или в формате со сжатием MP3.

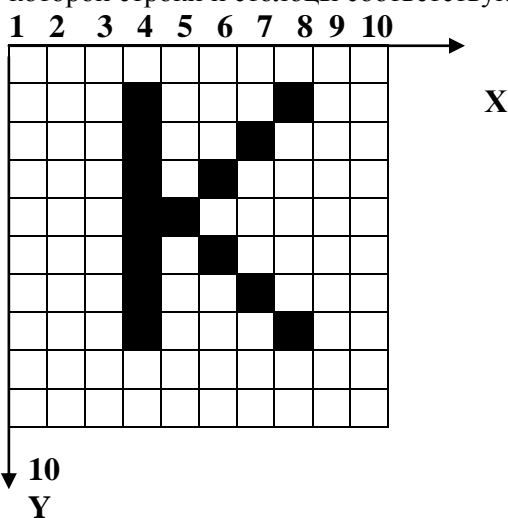
При сохранении звука в форматах со сжатием отбрасываются "избыточные" для человеческого восприятия звуковые частоты с малой интенсивностью, совпадающие по времени со звуковыми частотами с большой интенсивностью. Применение такого формата позволяет сжимать звуковые файлы в десятки раз, однако приводит к необратимой потере информации (файлы не могут быть восстановлены в первоначальном виде).

Задание:

1. Определить требуемый объем видеопамати для различных графических режимов экрана монитора, если известна глубина цвета на одну точку.

Режим экрана	Глубина цвета (бит на точку)				
	4	8	16	24	32
640 на 480					
800 на 600					
1024 на 768					
1280 на 1024					

2. Черно-белое (без градаций серого) растровое графическое изображение имеет размер  $10 \times 10$  точек. Какой объем памяти займет это изображение?
3. Для хранения растрового изображения размером  $128 \times 128$  пикселей отвели 4 КБ памяти. Каково максимально возможное число цветов в палитре изображения.
4. Монитор работает с 16 цветной палитрой в режиме  $640 \times 400$  пикселей. Для кодирования изображения требуется 1250 Кбайт. Сколько страниц видеопамати оно занимает?
5. На «маленьком мониторе» с растровой сеткой размером  $10 \times 10$  имеется черно-белое изображение буквы «К». Представить содержимое видеопамати в виде битовой матрицы, в которой строки и столбцы соответствуют строкам и столбцам растровой сетки.



6. Описать букву «К» последовательностью векторных команд.
7. Сколько секунд потребуется модему, передающему сообщения со скоростью 28800 бит/с, чтобы передать цветное растровое изображение размером  $640 \times 480$  пикселей, при условии, что цвет каждого пикселя кодируется тремя байтами?
8. Производится одноканальная (моно) звукозапись с частотой дискретизации 16 кГц и 24-битным разрешением. Запись длится 1 минуту, ее результаты записываются в файл, сжатие данных не производится. Определите размер полученного файла, выраженному в мегабайтах?
9. Определить размер (в байтах) цифрового аудиофайла, время звучания которого составляет 10 секунд при частоте дискретизации 22,05 кГц и разрешении 8 бит. Файл сжатию не подвержен.
10. В распоряжении пользователя имеется память объемом 2,6 Мб. Необходимо записать цифровой аудиофайл с длительностью звучания 1 минута. Какой должна быть частота дискретизации и разрядность?
11. Объем свободной памяти на диске — 5,25 Мб, разрядность звуковой платы — 16. Какова длительность звучания цифрового аудиофайла, записанного с частотой дискретизации 22,05 кГц?
12. Одна минута записи цифрового аудиофайла занимает на диске 1,3 Мб, разрядность звуковой платы — 8. С какой частотой дискретизации записан звук?

13. Две минуты записи цифрового аудиофайла занимают на диске 5,1 Мб. Частота дискретизации — 22050 Гц. Какова разрядность аудиоадаптера?

#### Содержание отчета

- Название работы;
- Цель работы;
- Формулировку задания;
- Решенные задания, с пояснением решения в виде таблиц, рассуждений, рисунков.

#### Контрольные вопросы

1. Сформулируйте принципы представления растрового изображения.
2. Сформулируйте принципы представления векторного изображения.
3. Сформулируйте принципы представления звуковой информации

#### Список литературы:

##### Основная литература

1. Информатика. Задачник – практикум в 2 т./ Под ред. И. Г. Семакина, Е. К. Хеннера: Том 1 – М.: Бинوم. Лаборатория знаний, 2003.-304с.
2. Максимов Н.В., Партыка Т. Л. Попов И. И. Архитектура ЭВМ и вычислительных систем:учебник. – М.:Форум, 2008.
3. Келим Ю.М. Вычислительная техника: учеб. Пособие для студ.сред.проф.образования/ Ю. М. Келим. – М.: Издательский центр «Академия», 2008.

##### Дополнительная литература

4. Таненбаум Э. Архитектура компьютера – СПб: Питер, 2010

## Практическая работа № 6

Тема: Исследование логических элементов и составление логических схем (микросхемы серии K155)

Цель: изучение логических микросхем серии K155

Оборудование: Микросхемы K155ЛА3, K155ЛЛ1, K155ЛИ1, K155ЛЕ1., светодиоды АЛ07БМ., Комбинированный прибор М92.

Пояснения к работе:

*Перед выполнением работы по учебникам и конспекту повторите следующие вопросы:*

1. Алгебра логики. Основные функции алгебры логики.
2. Логические элементы.
3. Применение логических элементов для синтеза логических схем.
4. Составление логической схемы по логической формуле.
5. Составление логической формулы по логической схеме.
6. Минимизация логических формул.

*Ознакомьтесь с оборудованием:*

1. Микросхемы K155ЛА3, K155ЛЛ1, K155ЛИ1, K155ЛЕ1.
2. Светодиоды АЛ07БМ.
3. Комбинированный прибор М92.

#### План работы:

1. Задание №1. Измерение уровней логических сигналов для микросхем серии K155.
2. Задание №2. Изучение свойств логических элементов.
3. Задание №3. Изучение комбинационных логических схем.

#### 1. Порядок выполнения задания № 1

- 1.1. Ознакомиться с лабораторным блоком № 5.
- 1.2. Подключить к гнездам X19, X20 постоянное напряжение 5 В.
- 1.3. Измерить с помощью цифрового прибора на пределе измерения 20 В напряжение на выходе переключателя SA1, затем SA2, соответствующие логической единице и логическому нулю.
- 1.4. Результаты занести в таблицу 1.1.



Таблица 1.1.

Состояние переключателя	Замкнутое (вкл.)	Разомкнутое (выкл.)
Уровень напряжения, В		
Логический сигнал		
Состояние светодиода		

## 2. Порядок выполнения задания №2.

### 2.1. Изучение микросхемы DD1:

- 2.1.1. Выход переключателя SA1, т.е. гнездо X1 соединить со входом X5 микросхемы DD1.
- 2.1.2. Состояние входа элемента (X5) и выхода (X8) фиксировать светодиодами.
- 2.1.3. Результаты наблюдений и переключений занести в таблицу 1.2.

Таблица 1.2

X5	X8
0	
1	

### 2.2. Изучение микросхемы DD2:

- 2.2.1. Выходы переключателей SA1, SA2 подключить к входам X6, X7 микросхемы.
- 2.2.2. Изменяя состояние входов элемента фиксировать с помощью светодиодов состояния входа X9.
- 2.2.3. Результаты наблюдений занести в таблицу 1.3.

Таблица 1.3

X6	X7	X9
0	0	
0	1	
1	0	
1	1	

- 2.3. Изучение микросхемы DD3: повторить пункты 2.2.1 – 2.2.3.
- 2.4. Изучение микросхемы DD4: повторить пункты 2.2.1 – 2.2.3.
- 2.5. Изучение микросхемы DD5: повторить пункты 2.2.1 – 2.2.3.

## 3. Порядок выполнения задания №3

- 3.1. Составить, затем собрать схему соединения двух логических элементов (произвольную). По аналогии с пунктом 2.2. заполнить таблицу истинности схемы.
- 3.2. Составить и собрать схему соединения четырёх (трёх) любых логических элементов и заполнить таблицу истинности.

## Содержание отчета

1. Название работы
2. Цель работы
3. Условные обозначения всех логических элементов и их названия.
4. Логические схемы, составленные из логических элементов.
5. Таблицы истинности
6. Ответы на вопрос 4 и 5.

## Контрольные вопросы

1. Каковы логические схемы, нумерация выводов микросхем К155ЛА3, -ЛЛ1, -ЛИТ, -ЛЕ1?
2. Запишите логическое выражение для схем, собранных в пунктах 3.1, 3.2 проделанной работы.
3. Проследите прохождение кода "01" по каждой из этих схем.
4. Составьте логическую схему по данному выражению:

$$F = X \vee Y \vee X \wedge Y$$

5. Заполните таблицу истинности для выражения, приведённого в вопросе 4 (теоретическую), предварительно упростив выражение.

## Список литературы:

Основная литература

1. Калиш Г.Г. Основы вычислительной техники. – М.: Высшая школа, 2000.
2. Келим Ю.М. Вычислительная техника. – М.: Издательский центр «Академия», 2005.
3. Максимов Н.В., Партыка Т. Л. Попов И. И. Архитектура ЭВМ и вычислительных систем: учебник. – М.: Форум, 2008.

## Дополнительная литература

1. Кузин А.В. Микропроцессорная техника. – М.: Издательский центр «Академия», 2004.
2. Максимов Н.В., Партыка Т.Л., Попов И.И. Архитектура ЭВМ и вычислительных систем. – М.: ФОРУМ: Инфра-М, 2005.
3. Таненбаум Э. Архитектура компьютера – СПб: Питер, 2010

## Практическая работа № 7

Тема: Исследование двоичного сумматора (микросхема серии К155ИМ2)

Цель: изучение принципа действия двух разрядного двоичного сумматора

Оборудование: Микросхемы К155ИМ2, Светодиоды АЛ307КМ

Пояснения к работе:

*Перед выполнением работы по учебникам и конспекту повторите следующие вопросы:*

1. Двоичный сумматор и его назначение.
2. Полусумматор и его схема на основе логических элементов.
3. Полный сумматор и его схема. Построение n-разрядного сумматора.

*Ознакомьтесь с оборудованием:*

1. Микросхемы К155ИМ2
2. Светодиоды АЛ307КМ

### План работы:

1. Задание № 1. Анализ данного сумматора
2. Задание № 2. Изучение принципа работы сумматора

## 1. Порядок выполнения задания №1

- 1.1. Ознакомиться с блоком №8.
- 1.2. К гнездам X1, X2 блока подать постоянное напряжение 5 В.
- 1.3. Проанализировать; какие сигналы поступают на входы сумматора при замыкании ключей S1 – S5.
- 1.4. То же, при размыкании S1 – S5.
2. Порядок выполнения задания №2
  - 2.1. Установить нулевой сигнал на входе переноса – P1 сумматора.
  - 2.2. Изменяя состояния входов сумматора A1, B1, A2, B2 с помощью ключей S1 – S4 светодиодов VD1 – VD3.
  - 2.3. Результаты наблюдений занести в таблицу 4.1.

Таблица 4.1.

[illegible]

- |      |   |
|------|---|
| 2.4. | Установить единичный сигнал на входе переноса Р1 сумматора. |
| 2.5. | Повторить опыт 2.2. и результаты занести в таблицу 4.2.     |

Таблица 4.2.

[illegible]

	A1															
	B2															
Состояния выходов	1															
	2															
	3															

### Содержание отчета

1. Название работы
2. Цель работы
3. Схему включения сумматора.
4. Таблицы состояний.

### Контрольные вопросы

1. Расшифруйте назначение выводов микросхемы K155ИМ2 и привести её обозначение с нумерацией выводов.
2. Приведите функциональную схему двух разрядного сумматора.
3. Проанализируйте сложение чисел  $A = 10$ ,  $B = 11$  по функциональной схеме.
4. Приведите таблицу состояний полного одноразрядного сумматора и реализовать её на элементах логики.
5. Запишите логическое выражение для переменной P2, используя таблицу 4.1, пункт 2.3.

### Список литературы:

#### Основная литература

1. Калиш Г.Г. Основы вычислительной техники. – М.: Высшая школа, 2000.
2. Келим Ю.М. Вычислительная техника. – М.: Издательский центр «Академия», 2005.
3. Максимов Н.В., Партыка Т. Л. Попов И. И. Архитектура ЭВМ и вычислительных систем: учебник. – М.: Форум, 2008.

#### Дополнительная литература

1. Кузин А.В. Микропроцессорная техника. – М.: Издательский центр «Академия», 2004.
2. Максимов Н.В., Партыка Т.Л., Попов И.И. Архитектура ЭВМ и вычислительных систем. – М.: ФОРУМ: Инфра-М, 2005.
3. Таненбаум Э. Архитектура компьютера – СПб: Питер, 2010

## Практическая работа № 8

Тема: Исследование регистра (микросхема серии ТМ8)

Цель: изучение устройства и принципа действия простейшего регистра

Оборудование: Микросхема K155ТМ8, светодиоды АЛ307БМ

Пояснения к работе: *Перед выполнением работы по учебникам и конспекту повторите следующие вопросы:*

1. Регистр и его назначение.
2. Последовательный и параллельный регистр. Временные диаграммы.
3. Зависимость разрядности регистра от количества триггеров.

Задание:

*Ознакомьтесь с оборудованием:*

1. Микросхема K155ТМ8
2. Светодиоды АЛ307БМ

1. Задание №1. Анализ схемы данного регистра
2. Задание №2. Изучение действий, выполняемых данным регистром

### 1. Порядок выполнения задания №1

- 1.1. Ознакомиться с блоком № 7.
- 1.2. Подать постоянное напряжение 5 В к гнездам X1, X2.
- 1.3. Проанализировать, какие сигналы поступают D – входы триггеров при замкнутых состояниях выключателей SA1 – SA3.

- 1.4. Проанализировать, какой сигнал поступает на R – входы триггеров при нажатии кнопки SB1. Какой уровень сигнала является активным?
- 1.5. Проанализировать, какой перепад напряжения возникает на C – входах триггеров при нажатии кнопки SB2.
2. Порядок выполнения задания № 2
  - 2.1. Нажатием кнопки SB1 «обнулить регистр». Состояния выходов триггеров фиксировать светодиодами VD1 – VD6.
  - 2.2. Записать информацию в регистр:
    - 2.2.1. Установить с помощью выключателей SA1 – SA3 на входе регистра код «111».
    - 2.2.2. Нажать кнопку SB2.
    - 2.2.3. Убедиться, что состояние регистра изменилось.
    - 2.2.4. Результаты наблюдений занести в таблицу 2.1.

Таблица 2.1.

№ п/п	Состояние входов					Состояние выходов					
	$\overline{R}$	$D_1$	$D_2$	$D_3$	$C$	$Q_1$	$Q_2$	$Q_3$	$\overline{Q_1}$	$\overline{Q_2}$	$\overline{Q_3}$
1											
2											
3											
4											
5											

- 2.3. Хранение информации:
  - 2.3.1. Изменить с помощью выключателей SA1 – SA3 код на входе регистра.
  - 2.3.2. Убедиться, что состояние регистра не изменилось.
  - 2.3.3. Результаты занести в таблицу 2.1.
- 2.4. Запись новой информации:
  - 2.4.1. Установить на входе регистра код 101.
  - 2.4.2. Нажать кнопку SB2 и наблюдать состояние регистра.
  - 2.4.3. Результаты занести в таблицу 2.1.
- 2.5. Обнуление регистра:
  - 2.5.1. Нажать кнопку SB1.
  - 2.5.2. Состояние регистра отразить в таблице 2.1.

#### Содержание отчета

- Название работы
- Цель работы
- Схему регистра, изображённую на блоке.
- Таблицу 2.1.
- Выводы, сделанные в пунктах 1.3., 1.4., 1.5.

#### Контрольные вопросы

1. Каковы условное обозначение и назначение выводов микросхемы K155TM8.
2. Какие разновидности регистров Вам известны?
3. Каковы условное обозначение и диаграмма переключений – триггера?
4. Изменить схему соединения триггеров в данном регистре для получения регистра сдвига.
5. Пояснить запись трёхразрядного кода в регистр с помощью временной диаграммы.

#### Список литературы:

##### Основная литература

1. Калиш Г.Г. Основы вычислительной техники. – М.: Высшая школа, 2000.
2. Келим Ю.М. Вычислительная техника. – М.: Издательский центр «Академия», 2005.

##### Дополнительная литература

1. Кузин А.В. Микропроцессорная техника. – М.: Издательский центр «Академия», 2004.

2. Максимов Н.В., Партыка Т.Л., Попов И.И. Архитектура ЭВМ и вычислительных систем. – М.: ФОРУМ: Инфра-М, 2005.

## Практическая работа № 9

Тема: Исследование двоичного счетчика и дешифратора (микросхемы серии K155ИЕ2, K514ИД1)

Цель: изучение принципа работы четырёхзарядного двоичного счётчика и дешифратора

Оборудование: Микросхема K155ИЕ2, микросхема K514ИД1, светодиоды АЛ307КМ, комбинированный прибор М92А, светодиодный индикатор АЛ307КМ.

Пояснения к работе:

*Перед выполнением работы по учебникам и конспекту повторите следующие вопросы:*

1. Счётчик и его назначение.
2. Построение счётчика на основе триггеров. Временные диаграммы.
3. Коэффициент пересчёта счетчика и его зависимость от числа триггеров.

*Ознакомьтесь с оборудованием:*

1. Микросхема K155ИЕ2.
2. Микросхема K514ИД1.
3. Светодиоды АЛ307КМ.
4. Комбинированный прибор М92А.
5. Светодиодный индикатор АЛ307КМ.

1. Задание № 1. Изучение микросхемы K155ИЕ2
2. Задание № 2. Изучение принципа работы счётчика и дешифратора

### 1. Порядок выполнения задания №1

- 1.1. Ознакомиться с блоком №6.
- 1.2. Проанализировать информацию о микросхеме K155ИЕ2:
  - 1.2.1. Микросхема K155ИЕ2 – универсальный четырёхзарядный интегральный счётчик.
  - 1.2.2. Состав микросхемы: триггер со счётным входом и трёхразрядный счётчик с коэффициентом деления – 5; вместе они образуют двоично – десятичный счётчик.
  - 1.2.3. Выводы микросхемы:
    - С1 – вход
    - Р0 - & - а вход установки «0»
    - 1 – С2 – переключатель, устанавливающий коэффициент пересчёта – 9
    - Р9 - & - вход установки «9»
    - 1,2,3,4 – выходы.

### 2. Порядок выполнения задания №2.

- 2.1. К гнездам X1, X3 блока подать постоянное напряжение 5 В.
- 2.2. Нажатием кнопки SB2 установить счётчик в нулевое состояние.
- 2.3. Нажатием кнопки SB1 на вход счётчика подать импульс, при этом должен загореться светодиод, включенный во входную цепь счётчика.
- 2.4. Состояние выходов счётчика также фиксировать светодиодами.
- 2.5. Логические состояния выходов дешифратора контролировать комбинированным прибором на пределе измерения 20 В.
- 2.6. Состояние светодиодного индикатора фиксировать для каждого состояния счётчика.
- 2.7. Подавая на вход счётчика импульсную последовательность многократным нажатием кнопки SB1, заполнить таблицу 3.1. состояний всех элементов.

Таблица 3.1.

№ импульса на входе счетчика	Состояния выходов счетчика				Состояния выходов дешифратора								Состояния индикатора
	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13	X14		

0												
1												
2												
3												
4												
5												
6												
7												
8												
9												
10												

#### Содержание отчета

- Название работы
- Цель работы
- Схему включения счётчика и дешифратора.
- Таблицу состояний 3.1.

#### Контрольные вопросы

1. Как связан номер счётного импульса с состоянием счётчика?
2. Что такое Т – триггер и как его получить? Привести схему и таблицу состояния Т – триггера.
3. Составить функциональную схему четырёхразрядного счётчика.
4. Привести диаграмму переключений счётчика (п.8.3).
5. Какие изменения необходимо ввести в схему счётчика (п.8.3) чтобы коэффициент пересчёта равнялся 9?
6. Сформулировать функцию дешифратора в данной работе.
7. Как построить вычитающий счётчик?
8. Привести обозначение и нумерацию выводов микросхемы K155IE2.

#### Список литературы:

##### Основная литература

1. Калиш Г.Г. Основы вычислительной техники. – М.: Высшая школа, 2000.
2. Келим Ю.М. Вычислительная техника. – М.: Издательский центр «Академия», 2005.

##### Дополнительная литература

1. Кузин А.В. Микропроцессорная техника. – М.: Издательский центр «Академия», 2004.
2. Максимов Н.В., Партыка Т.Л., Попов И.И. Архитектура ЭВМ и вычислительных систем. – М.: ФОРУМ: Инфра-М, 2005.

## Практическая работа № 10

Тема: Изучение системы команд учебного компьютера «Нейман». Программирование алгоритмов на языке машинных команд.

Цель: получить знания о системе команд УК «Нейман», принципах программирования на ЯМК.

Формируемые компетенции: (ПК, ОК)

Оборудование: (программное обеспечение, реактивы, инструменты, материалы... Если разное для каждой работы)

Пояснения к работе:

Описание системы УК «Нейман»:

- Команда процессора - 3-х адресная;
- Длина поля (кода операции) - 1 байт;
- Адресуем каждый байт памяти;
- Длина ячейки памяти = длина машинной команды;
- Системы счисления - 2-ая, 16-ая.

Принцип хранимой программы (Неймана): во время исполнения программа хранится в оперативной памяти вместе с данными. Каждая команда программы занимает ячейку памяти и имеет свой адрес.

Формат команды

Код операции	Адреса величин
--------------	----------------

Формат трёхадресной команды

Код операции	Адрес 1 аргумента	Адрес 2 аргумента	Адрес результата
--------------	-------------------	-------------------	------------------

Система команд УК «Нейман»

Код операции	Мнемо код	Вид команды	Операция	Пояснение
00	MOV	00 a1 00 a3	Пересылка	Перенос содержимого ячейки a1 в a3
01	ADD	01 a1 a2 a3	Сложение	Сложение содержимого ячейки a1 с a2 и перенос рез-та в a3
02	SUB	02 a1 a2 a3	Вычитание	Вычитание содержимого из ячейки a1 a2 и перенос рез-та в a3
03	MUL	03 a1 a2 a3	Умножение	Умножение содержимого ячейки a1 с a2 и перенос рез-та в a3
04	DIV	04 a1 a2 a3	Остаток от деления	Деление содержимого ячейки a1 на a2 и перенос рез-та в a3
05	MOD	05 a1 a2 a3	Целое от деления	Деление содержимого ячейки a1 на a2 и перенос рез-та в a3
0B	GO	0B 00 00 a3	Безусловный переход	Переход на ячейку a3
0A	IFGO	0A 00 00 a3	Условный переход	Если результат предыдущей операции больше нуля, то переход на ячейку a3 иначе – продолжение со следующей ячейки
77	STOP	77 00 00 00	Остановка машины	Завершение программы
FF	NOP	FF 00 00 00	Пустая команда	Команда осуществляет «застой» машины на один рабочий такт (в учебных задачах практически не используется)

Управление учебным компьютером осуществляется по средствам программы в авто-режиме (принцип программного управления) и в пошаговом режиме. Самая 1 команда хранится в ячейке с 0 адресом. команды выполняются в естественном порядке, до тех пор, пока не встретится команда остановки или невыполнимая команды. В регистрах операндах отображаются данные участвующие в операциях. В регистре счетчика команд содержится адрес след команд.

Разберём работу У.К. Неймана на примере с задачей об вычислении Р и S прямоугольника:

Введем последовательно числа: 4(сторона), 3(сторона) и 2(число для формулы)

На выходе получим:  $E_{16}$  и  $C_{16}$  – периметр и площадь

У К "Н Е Й М А Н"					< 16 >	
П А М Я Т Ь						
Адрес	КОП	A1	A2	A3	РЕГИСТРЫ	
0 0	00	FC	00	28	РК	00 38 00 FC
0 4	00	FC	00	24	РОН1	00 00 00 0C
0 8	00	FC	00	2C	РОН2	.. .. .
0 C	01	28	24	30	РР	00 00 00 0C
1 0	03	2C	30	34	W	1
1 4	03	24	28	38		
1 8	00	34	00	FC		
1 C	00	38	00	FC		
2 0	??	00	00	00		
2 4	00	00	00	03		
2 8	00	00	00	04		
2 C	00	00	00	02		
3 0	00	00	00	07		
И Н Д И К А Т О Р						СЧЕТЧИК КОМАНД
FC	00	00	00	0C		1 C
Прочтите индикатор, для продолжения нажмите <Enter>						

Задание:

1. Проверьте работу предложенного алгоритма
2. Составьте программу в УК «Нейман» для вычисления значения выражения  $y = x^2 + x * 5$ .
3. Вычислите площадь круга через радиус или длину круга с диаметром (числом  $\pi = 3$ ).

Содержание отчета

- Название работы;
- Цель работы;
- Формулировку задания;
- Решенные задания, с пояснением решения в виде таблиц, рассуждений, рисунков.

Контрольные вопросы

4. Как распределяется память при трехадресном формате команды?
5. Как целесообразнее распределить память при программировании на ЯМК?
6. В какой системе счисления представлены данные, команды, константы в УК «Нейман»?

Список литературы:

Основная литература

1. Информатика. Задачник – практикум в 2 т./ Под ред. И. Г. Семакина, Е. К. Хеннера: Том 1 – М.: Бином. Лаборатория знаний, 2003.-304с.
2. Максимов Н.В., Партыка Т. Л. Попов И. И. Архитектура ЭВМ и вычислительных систем:учебник. – М.:Форум, 2008.
3. Келим Ю.М. Вычислительная техника: учеб. Пособие для студ.сред.проф.образования/ Ю. М. Келим. – М.: Издательский центр «Академия», 2008.

Дополнительная литература

4. Таненбаум Э. Архитектура компьютера – СПб: Питер, 2010

## Практическая работа № 11

Тема: Программирование алгоритмов на языке машинных команд.

Цель: изучить команды ветвления, отработка умений по программированию разветвляющихся и циклических алгоритмов

Формируемые компетенции: (ПК, ОК)

Оборудование: (программное обеспечение, реактивы, инструменты, материалы... Если разное для каждой работы)

Пояснения к работе:

Как и во всех языках программирования в УК «Нейман» существуют команды ветвления – безусловный и условный переход.

Безусловный переход позволяет перейти на определенную ячейку адреса в программе.

Старайтесь программировать таким образом, чтобы программа не попадала в бесконечный цикл.

Формат команды

ОП	Арг умент1	Арг умент2	Рез ультат
В	00	00	04

Условный переход позволяет перейти на определенную ячейку адреса в программе, если предыдущая операция получила результат больше нуля, иначе программа начнёт выполняться на следующей ячейке с командой.

Формат команды

ОП	Арг умент1	Арг умент2	Рез ультат
А	00	00	04

Условный переход предоставляет возможность организовывать цикл «до тех пор - пока» (в ЯП высокого уровня – Repeat-Until), и так же организовать условие.



Разберём задачу на ветвление:

Вычислить значение выражения  $y = x + 5$ , если  $x > 0$ ;  $y = x - 5$ , если  $x \leq 0$

Введем число  $-3(FFFFFFFD_{16})$ , на выходе получим  $FFFFFFF8_{16}(-8)$

**У К "Н Е Й М А Н"** < 16 >

**П А М Я Т Ь**

Адрес	КОП	A1	A2	A3
0 0	00	FC	00	20
0 4	01	20	28	20
0 8	0A	00	00	14
0 C	02	20	24	20
1 0	0B	00	00	18
1 4	01	20	24	20
1 8	00	20	00	FC
1 C	77	00	00	00
2 0	FF	FF	FF	F8
2 4	00	00	00	05
2 8	00	00	00	00
2 C	..	..	..	..
3 0	..	..	..	..

**И Н Д И К А Т О Р**

F C	FF	FF	FF	F8
-----	----	----	----	----

**Р Е Г И С Т Р Ы**

PK	00	20	00	FC
POH1	FF	FF	FF	F8
POH2	..	..	..	..
PP	FF	FF	FF	F8
w	0			

**С Ч Е Т Ч И К К О М А Н Д**

1 8
-----

Прочтите индикатор, для продолжения нажмите <Enter>

Задание:

1. Проверьте работу предложенного алгоритма
2. Составьте программу в УК «Нейман» для решения формулы:

$$\sum_{i=1}^N (5 * i) + 2$$

Где число N вводится пользователем;

3. Составьте программу в УК «Нейман» для поиска максимального из трёх чисел и вывести его, если самое максимальное число больше 15, то вывести самое минимальное из двух оставшихся.

Содержание отчета

- Название работы;
- Цель работы;
- Формулировку задания;
- Решенные задания, с пояснением решения в виде таблиц, рассуждений, рисунков.

Контрольные вопросы

1. Как реализуется ветвление в ЯМК?.
2. Как реализуют цикл в ЯМК?
3. Перечислите достоинства и недостатки при программировании на ЯМК.

Список литературы:

Основная литература

1. Информатика. Задачник – практикум в 2 т./ Под ред. И. Г. Семакина, Е. К. Хеннера: Том 1 – М.: Бином. Лаборатория знаний, 2003.-304с.
2. Максимов Н.В., Партыка Т. Л. Попов И. И. Архитектура ЭВМ и вычислительных систем:учебник. – М.:Форум, 2008.
3. Келим Ю.М. Вычислительная техника: учеб. Пособие для студ.сред.проф.образования/ Ю. М. Келим. – М.: Издательский центр «Академия», 2008.

Дополнительная литература

4. Таненбаум Э. Архитектура компьютера – СПб: Питер, 2010

## Практическая работа № 12

Тема: Получение информации о параметрах компьютерной системы

Цель: получение информации об архитектуре персонального компьютера с помощью утилит аппаратного мониторинга.

Формируемые компетенции: (ПК, ОК)

Оборудование: (программное обеспечение, реактивы, инструменты, материалы... Если разное для каждой работы)

Пояснения к работе:

Обычно пользователь при приобретении нового компьютера хорошо знает лишь основные его внутренние характеристики, чаще всего это частота и тип процессора, объем системной памяти, видеопамяти и название графического чипа, емкость жесткого диска. Причем в последних трех случаях очень часто неизвестными остаются производители этих устройств, впрочем, как и ряд более тонких технических параметров. Если же компьютер был подарен, куплен по совету сторонних лиц либо предоставлен на работе, то пользователю ведомо о его конфигурации обычно еще меньше. Как тут со всем сразу разберешься, тем более что все и так вроде бы отлично функционирует? Иногда даже специалисты затрудняются дать однозначный ответ о тех или иных характеристиках, особенно, если это относится к столь распространенному виду комплектующих - «NONAME», а системные блоки надежно опломбированы, дабы долго оставались, соблюдены нерушимые гарантийные обязательства, а заодно и секретность их истинного содержания.

Но рано или поздно пользователю приходится разбираться с внутренностями своего электронного друга, и дело здесь не только в одной любознательности. Когда же наступает такой момент? Наиболее распространенным является случай, когда перестала загружаться операционная система и ее приходится переустанавливать с нуля. Естественно, никакая гарантия на ОС не распространяется. Но вот, наконец, пользователь переустановил ее своими силами - это только половина дела в конфигурировании системы. Далее идет черед за драйверами всех тех устройств, которые находятся внутри системного блока: материнской платы, видеокарты, звуковой платы, возможно, внутреннего модема или сетевой карты, иногда чего-то еще. Стандартные средства Windows не всегда могут полностью разрешить эту проблему. Тут уже нужно обладать определенными знаниями, чтобы правильно выбрать и установить драйверы: сначала для чипсета материнской платы, потом и для всех остальных устройств. Если такая информация не была известна изначально, то ее придется добыть самостоятельно. К тому же делать это лучше заранее, а не дожидаться, пока гром грянет, и потом хвататься за все в экстренном порядке.

Если ваша система живет вполне благополучно, то даже в этом случае каждому любознательному пользователю будет интересно узнать, из чего же действительно состоит его компьютер. Выяснить это на самом деле не так уж и сложно, вам помогут специальные программы, предназначенные для определения конфигурации ПК, - и его не придется разбирать по частям.

Большинство подобных утилит предназначены для работы из-под Windows. Часто подобное ПО довольно функционально, предоставляя пользователю, возможность узнать конфигурацию не только аппаратной части, но и выяснить особенности операционной системы. К дополнительным возможностям относятся всевозможные тесты на производительность и работоспособность тех или иных компонентов. Хотя в последнем случае лучше пользоваться иными средствами - целевыми утилитами, специально разработанными для конкретных испытаний. Некоторые программы могут работать из-под OOS. Последнее может понадобиться при исследовании очень старых компьютеров, на которых Windows не установлена, что на сегодняшний день уже большая редкость. Гораздо чаще возникает другая ситуация - неработоспособность Windows. Тогда-то и выручает старенький DOS и диагностические программы под него, запущенные с загрузочной дискеты или диска.

Задание:

1. Получите информацию о параметрах компьютерной системы с помощью разных утилит. Оформите результаты в виде таблицы, позволяющей сравнить полученные параметры с помощью разных утилит.

#### Утилита для диагностики конфигурации ПК SiSoftware Sandra

1. Рассмотрите интерфейс **SiSoftware Sandra**. Все ее инструменты разбиты на четыре категории: информационные модули, бенчмаркинг-модули, просмотрные модули, тестовые модули. Каждому модулю соответствует ярлык в поле окна программы.
2. В верхней строке инструментов Sandra есть соответствующая кнопка с подменю - Переключение отображения модулей. Отобразите ярлыки одного типа модулей, допустим информационных, или же **Вывести Все**. Предназначение того или иного типа модулей понятно по названию.
3. Воспользовавшись модулем **Сводная информация**, посмотрите список устройств и краткие параметры конфигурации компьютера.
4. В разделе **Информация о материнской плате** приведены характеристики, модели и имена производителей материнской платы и ее компонентов, а также устройств, относящихся к ней, в частности, системной памяти.
5. Для системной памяти определите рабочие частоты и текущий список таймингов.
6. Просмотреть информацию о центральном процессоре, видеосистеме, устройствах на шинах PCI, AGP, Card Bas, устройствах связи.
7. Рассмотрите инструменты SiSoftware Sandra, тестирующие производительность компонентов компьютера, средства измерения скоростных показателей.
8. В разделе просмотрных модулей предоставлена информация о системных файлах и настройках программной части системы. Просмотрите тексты основных системных файлов.
9. Результаты диагностики сохраните в виде подробного отчета, воспользовавшись **Мастером создания отчетов**, ярлык которого присутствует во всех разделах.

**Примечание.** По каждому из вышеперечисленных пунктов, полученная информация заносится в отчет по практической работе.

#### Утилита для диагностики конфигурации ПК PC Wizard

1. Найдите в разделе **Hardware** ярлык **Mainboard**. В подразделе **Mainboard** находится информация об установленной системной памяти, основных характеристиках и таймингах.
2. Посмотрите характеристики центрального процессора, показания датчиков напряжения, температуры и оборотов вентилятора CPU.
3. Зайдите в раздел **Drivers**, где представлена информация обо всех физических приводах и логических дисках. Выведите полный список состояния атрибутов SMART, а также температуру интегрированного в привод термодатчика.
4. Перейдите в подраздел **Voltage, Temperature and Fans**, Выведите текущие показания доступных датчиков систем аппаратного мониторинга компьютера: напряжения процессора и материнской платы, обороты вентиляторов, подключенных к материнской плате, температура системных сенсоров, процессора и датчика привода жестких дисков.
5. Перейдите в подраздел **Configurations**. Рассмотрите информацию о Windows, интернет-приложениях, запущенных процессах, установленных в систему динамических библиотеках, шрифтах и ассоциируемых с приложениями расширениях файлов.
6. Перейдите в подраздел **System Files**, где получите информацию и посмотрите тексты системных файлов.

**Примечание.** По каждому из вышеперечисленных пунктов, полученная информация заносится в отчет по практической работе.

#### Утилита для диагностики конфигурации ПК AIDA32

1. В подразделе **Summary** раздела **Computer** рассмотрите перечень основных компонентов, из которых состоит компьютер.
2. В подразделе **Sensor** можно найти перечень доступных датчиков средств аппаратного мониторинга, которыми оснащена материнская плата (но производители могут не задействовать их все), с выводом в окне их текущих показаний. Узнайте температуру

процессора, чипсета, напряжения материнской платы и частоту вращения вентиляторов.

*Для более подробного рассмотрения каждого из перечисленных разделов нужно обратиться к соответствующим пунктам. Так, в подразделах **Motherboards** и **Chipset** дается довольно подробная техническая информация о соответствующих компонентах. Здесь можно найти даже геометрические размеры установленной в компьютере материнской платы.*

3. В подразделе **Display** просмотрите характеристики видеокарты и монитора, видеоакселераторе.
4. Запишите тип и характеристики видеочипа, на котором построена видеокарта, объем видеопамати, ее тип, рабочие частоты графического процессора и памяти.
5. В подразделе **Fonts** просмотрите список установленных в системе шрифтов.
6. В разделе **Operating System** AIDA32 рассмотрите основные характеристики Windows.
7. В разделе **Processes** узнайте обо всех запущенных на текущий момент программах.
8. В разделе **Software** просмотрите список всех проинсталлированных в системе приложений, узнайте о тех из них, что находятся в автозагрузке, проверьте зарегистрированные типы файлов.

**Примечание.** По каждому из вышеперечисленных пунктов, полученная информация заносится в отчет по практической работе.

#### Содержание отчета

- Название работы;
- Цель работы;
- Формулировку задания;
- Решенные задания, с пояснением решения в виде таблиц, рассуждений, рисунков.

#### Контрольные вопросы

1. Какая из изученных утилит наиболее эффективна для вывода информации о настройках компьютера?
2. Какая из изученных утилит наиболее эффективна для вывода информации о настройках операционной системы?
3. Дайте сравнительный анализ рассмотренных утилит по количественным показателям.

#### Список литературы:

##### Основная литература

1. Максимов Н.В., Партыка Т. Л. Попов И. И. Архитектура ЭВМ и вычислительных систем: учебник. – М.: Форум, 2008.
2. Келим Ю.М. Вычислительная техника: учеб. Пособие для студ. сред. проф. образования/ Ю. М. Келим. – М.: Издательский центр «Академия», 2008.

##### Дополнительная литература

3. Таненбаум Э. Архитектура компьютера – СПб: Питер, 2010

##### Интернет ресурсы

<http://tomcomp.com.ua/page/kak-uznat-parametry-svoego-kompjutera>  
<http://dontreg.ru/aida32>

## Практическая работа № 13

Тема: Изучение конфигурации компьютера, аппаратного мониторинга с помощью стандартных утилит.

Цель: изучить интерфейсы ПК, сформировать умения по подключению внешних периферийных устройств, провести аппаратный мониторинг и тестирование ПК

Оборудование: ПК

Пояснения к работе:

Периферийные шины используются в основном для внешних запоминающих устройств.

**Интерфейс IDE** (Integrated Drive Electronics) – интерфейс устройств со встроенным контроллером. Поддерживает несколько способов обмена. Первый способ производит обмен данными через регистры процессора под его непосредственным управлением. Следствием этого является высокая загрузка процессора при операциях ввода/вывода. Вторым способом является использование режима прямого доступа к памяти, при котором контроллер интерфейса IDE и контроллер прямого доступа к памяти материнской платы пересылают данные между диском и оперативной памятью, не загружая центральный процессор. В целях развития возможностей интерфейса IDE была предложена его расширенная спецификация EIDE (синонимы ATA, ATA-2). Она поддерживает накопители емкостью свыше 504 Мбайт, поддерживает несколько накопителей IDE и позволяет подключать к одному контроллеру до четырех устройств, а также поддерживает периферийные устройства, отличные от жестких дисков. Расширение спецификации IDE для поддержки иных типов накопителей с интерфейсом IDE называют также ATAPI.

**Интерфейс SCSI** (Small Computer System Interface) - является стандартным интерфейсом для подключения приводов компакт-дисков, звуковых плат и внешних устройств массовой памяти. Спецификацией SCSI предусматривается параллельная передача данных по 8, 16 или 32 линиям данных. Структура SCSI, по существу, является магистральной, хотя устройства включаются в нее по принципу последовательной цепочки. Каждое SCSI-устройство имеет два разъема – один входной, а другой выходной. Все устройства объединяются в последовательную цепочку, один конец которой подключается к контроллеру интерфейса. Все устройства работают независимо и могут обмениваться данными как с компьютером, так и друг с другом. К шине SCSI можно подключить до 8 устройств, включая основной контроллер SCSI (хост-адаптер). Контроллер SCSI является, по сути, самостоятельным процессором и имеет свою собственную BIOS. К шине Wide SCSI подключается до 15 устройств.

**ACPI** (Advanced Configuration Power Interface – расширенный интерфейс конфигурирования и питания) – интерфейс, представляющий собой единую систему управления питанием для всех компонентов компьютера.

Принтеры, модемы и другое периферийное оборудование подключаются к компьютеру через стандартизированные интерфейсы, называемые портами. В зависимости от способа передачи информации между сопряженными устройствами различают параллельные и последовательные интерфейсы.

**Последовательный порт стандарта RS-232-C.** Является стандартом для соединения ЭВМ с различными последовательными внешними устройствами. В операционных системах каждому порту RS-232 присваивается логическое имя COM1-COM4.

**Параллельный порт** используется для одновременной передачи 8 битов информации. В компьютерах этот порт используется главным образом для подключения принтера, графопостроителей и других устройств. Параллельные порты обозначаются LPT1-LPT4.

**Интерфейс USB** (Universal Serial Bus) – универсальная последовательная шина призвана заменить устаревшие последовательный (COM-порт) и параллельный (LTP-порт) порты. Шина USB допускает подключение новых устройств без выключения компьютера. Шина сама определяет, что именно подключили к компьютеру, какой драйвер и ресурсы понадобятся устройству, после чего выделяет их без вмешательства пользователя. Шина USB позволяет подключить до 127 устройств.

**IEEE 1394** (Institute of Electrical and Electronic Engineers 1394 – стандарт Института инженеров по электротехнике и электронике 1394) - последовательный интерфейс, предназначенный для подключения внутренних компонентов и внешних устройств. IEEE 1394 известен также под именем **FireWire** «огненный провод». Цифровой последовательный интерфейс FireWire характеризуется высокой надежностью и качеством передачи данных, его протокол поддерживает гарантированную передачу критичной по времени информации, обеспечивая прохождение видео- и аудиосигналов в реальном масштабе времени без заметных искажений. При помощи шины FireWire можно подключить до 63 устройств и практически в любой конфигурации, чем она выгодно отличается от трудноконфигурируемых шин SCSI. Этот интерфейс используется для подключения жестких дисков, дисководов CD-ROM и DVD-ROM, а также высокоскоростных внешних устройств, таких как видеокамеры, видеомагнитофоны и т.д.

**Задание:**

1. Подключение внешних устройств

- 1.1. **Отключить электропитание ПК.**
- 1.2. Подключить жесткий диск к системной плате.
- 1.3. Подключить CD-ROM к системной плате.
- 1.4. Определить внешние интерфейсы целевого компьютера.
- 1.5. Подключить к целевому компьютеру принтер.
- 1.6. Подключить к целевому компьютеру монитор
- 1.7. Подключить к целевому компьютеру сканер.
2. Аппаратный мониторинг
  - 2.1. Включить компьютер.
  - 2.2. Услышать и записать звуковые сигналы при запуске компьютера.
  - 2.3. Описать эти сигналы.
  - 2.4. По сигналам определить характер неисправностей.
  - 2.5. Добиться загрузки ос (при необходимости зайти в setup для устранения мелких неисправностей).
  - 2.6. Описать главную файловую систему жесткого диска, для чего воспользоваться только командами dos.
  - 2.7. Найти программу CHECKIT или аналогичную
  - 2.8. Переписать на USB-накопитель.
  - 2.9. Запустить с USB-накопитель программу CHECKIT или аналогичную.
  - 2.10. Описать программу CHECKIT, определяемые параметры, дать пояснения по каждой позиции на русском языке.
  - 2.11. Определить все диагностические параметры компьютера с помощью программы CHECKIT, записать на бумагу и дать пояснения по каждому параметру.

#### Содержание отчета

- Название работы;
- Цель работы;
- Формулировку задания;
- Решенные задания, с пояснением решения в виде таблиц, рассуждений, рисунков.

#### Контрольные вопросы

1. Укажите основные особенности интерфейса IDE.
2. Укажите основные особенности интерфейса SCSI .
3. Укажите основные особенности порта ACPI .
4. Укажите основные особенности последовательного порта стандарта rs-232-с.
5. Укажите основные особенности параллельного порта .
6. Укажите основные особенности порта IEEE 139.
7. Укажите основные особенности порта USB .

#### Список литературы:

##### Основная литература

1. Максимов Н.В., Партыка Т. Л. Попов И. И. Архитектура ЭВМ и вычислительных систем: учебник. – М.: Форум, 2008.
2. Келим Ю.М. Вычислительная техника: учеб. Пособие для студ. сред. проф. образования/ Ю. М. Келим. – М.: Издательский центр «Академия», 2008.

##### Дополнительная литература

3. Таненбаум Э. Архитектура компьютера – СПб: Питер, 2010

##### Интернет ресурсы

1. [http://www.nix.ru/price/price\\_list.html?section=motherboards\\_all](http://www.nix.ru/price/price_list.html?section=motherboards_all)
2. <http://softobase.com/ru/sistema/utility>
3. <http://pc-performance.ru/>
4. <http://winsoft.ru/utilites/>

5. <http://www.parallel.ru>
6. <http://www.citilink.ru/catalog/parts/>
7. <http://www.wikipedia.org>

## Практическая работа № 14

Тема: Установка и настройка программного обеспечения компьютерных систем

Цель: изучить основные термины программного обеспечения; научиться устанавливать программное обеспечение, обновлять и использовать по назначению.

Оборудование: Персональный компьютер

Пояснения к работе:

**Установка программного обеспечения, инсталляция** — процесс установки программного обеспечения на компьютер конечного пользователя. Выполняется особой программой (пакетным менеджером), присутствующей в операционной системе (например, RPM, APT или dpkg в Linux, Установщик Windows в Microsoft Windows), или же входящим в состав самого программного обеспечения средством установки. В операционной системе GNU очень распространено использование системы GNU toolchain и её аналогов для компиляции программного обеспечения непосредственно перед установкой.

Большинство программ поставляются для продажи и распространения в сжатом (упакованном, см.: Архив) виде. Для нормальной работы они должны быть распакованы, а необходимые данные правильно размещены на компьютере, учитывая различия между компьютерами и настройками пользователя. В процессе установки выполняются различные тесты на соответствие заданным требованиям, а компьютер необходимым образом конфигурируется (настраивается) для хранения файлов и данных, необходимых для правильной работы программы.

Установка, как правило, включает в себя размещение всех необходимых программе файлов в соответствующих местах файловой системы, а также модификацию и создание конфигурационных файлов. Пакетные менеджеры также выполняют при установке контроль зависимостей, проверяя, есть ли в системе необходимые для работы данной программы пакеты, а в случае успешной установки регистрируя новый пакет в списке доступных.

Так как данный процесс является различным для каждой программы и компьютера, то многие программы (включая сами операционные системы) поставляются вместе с универсальным или специальным установщиком — программой, которая автоматизирует большую часть работы, необходимой для их установки.

Некоторые программы написаны таким образом, что устанавливаются простым копированием своих файлов в нужное место, а самого процесса установки как такового нет. Про такие программы говорят, что они «не требуют установки». Это распространено среди программ для Mac OS X, DOS и Microsoft Windows. Существуют операционные системы, которые не требуют установки, и, таким образом, могут быть напрямую запущены с загрузочного диска (компакт- или DVD-диск), не оказывая воздействия на другие операционные системы, установленные на компьютере пользователя. Примером такой операционной системы является Knoppix или Mac OS 1-9.

Данный термин также распространяется на плагины, драйверы и программные файлы, которые сами по себе не являются программами.

Обычные операции, выполняемые в процессе установки программного обеспечения, включают создание или изменение:

- Используемых и неиспользуемых совместно программных файлов.
  - Каталогов.
  - Записей конфигурационных файлов, используемых одной программой, или совместно.
  - Переменных среды.
- Возможные варианты установки

- Установка вручную — установка выполняется без установщика или со значительным количеством операций, вручную выполняемых пользователем.
- «Тихая» установка — установка, в процессе которой не отображаются сообщения или окна. «Тихая» установка не является синонимом «автоматическая установка», хотя часто ошибочно используется в этом значении.
- Автоматическая установка — установка, которая выполняется без вмешательства со стороны пользователя, исключая, конечно, сам процесс её запуска. Процесс установки иногда требует взаимодействия с пользователем, который управляет процессом установки, делая выбор: принимая [пользовательское соглашение](#), настраивая параметры, указывая пароли и так далее. В [графических средах](#) могут использоваться инсталляторы, которые предоставляют так называемого Мастера установки, однако и они зачастую предоставляют параметры [командной строки](#), позволяющие выполнить полностью автоматическую установку.
- Самостоятельная установка — установка, которая не требует начального запуска процесса. Например, [Vodafone Mobile Connect USB Modem](#), который устанавливается с [USB-порта](#) компьютера при подключении к нему без необходимости в ручном запуске.
- Удалённая установка — установка, которая выполняется без использования монитора, подсоединённого к компьютеру пользователя (в частности, выполняемая на компьютере без [видеовыхода](#) вообще). Это может быть контролируемая установка с другой машины, соединённой через [локальную сеть](#) или посредством [последовательного кабеля](#). Автоматическая и удалённая установки являются обычными операциями, выполняемыми [системными администраторами](#).
- «Чистая» установка — установка, выполняемая в отсутствие таких факторов, которые могут изменяться от программы к программе. Ввиду сложности типичной установки, имеется множество факторов, влияющих на её успешный исход. В частности, файлы, оставшиеся от предыдущей установки этой же программы, или нестабильное состояние операционной системы могут привести к неправильной установке и работе программы.
- Непосредственная установка — установка программы, выполняемая с её копии на жестком диске (называемой flat copy), а не с самого оригинального носителя (обычно компакт- или DVD-диск). Это может быть полезным в ситуациях, когда целевая машина не способна справиться с произвольным доступом для чтения с [оптических дисководов](#) во время выполнения задач, вызывающих большую загрузку процессора, как, например, при установке программ.

#### Программа установки

**Программа установки, установщик или инсталлятор** ([англ. installer](#)) — это программа, которая устанавливает программное обеспечение на компьютер. Некоторые установщики специально сделаны для установки содержащихся в них файлов, другие же являются универсальными и работают, считывая содержимое комплекта программного обеспечения, которое необходимо установить.

Различия между [системой управления пакетами](#) и установщиком:

Package Management System	Installer
Обычно часть операционной системы	Каждый продукт поставляется вместе со своим установщиком
Использует одну базу данных для установки	Производит свою установку
Может проверять и управлять всеми пакетами в системе	Работает только со связанным продуктом
Один производитель	Много производителей
Один формат пакета	Много форматов установщика

#### Семейство Windows

Наиболее популярным форматом для семейства [Windows NT](#) является установочный пакет [MSI](#), который устанавливается посредством [Установщика Windows](#). Компании, производящие средства для создания инсталляторов: InstallShield ([InstallShield Wizard](#)), Macrovision ([InstallAnywhere](#)), Wise Solutions, Inc., SetupBuilder, InstallBuilders ([Smart Install](#)



[Maker](#)). Большинство из этих средств могут создавать как пакеты MSI, так и свои собственные пакеты.

Бесплатными альтернативами являются [NSIS](#), [Actual Installer Free](#), [Clickteam Install Creator](#), [Inno Setup](#), [Install Simple](#), а также инструментальные средства от Microsoft ([WiX](#)).

#### Семейство UNIX

Большинство дистрибутивов операционных систем на базе [GNU](#), [Linux](#) и [BSD](#) имеет встроенные [системы управления пакетами](#), с помощью которых можно устанавливать как необходимые компоненты операционной системы, так и стороннее программное обеспечение, зачастую даже если оно использует собственный установщик, которым это не предусмотрено.

#### Семейство Mac OS

Mac OS X также использует систему управления пакетами. Некоторые коммерческие приложения для Mac OS X используют отдельный установщик, например, [Installer VISE](#) или Stuffit [InstallerMaker](#). Приложения, которые не нуждаются в установке дополнительных компонентов системы, могут быть установлены посредством простого копирования файлов приложения в нужное место на жестком диске. Mac OS X также включает отдельное приложение для обновления программ [Software Update](#) (также известное как команда [оболочки](#) softwareupdate), но оно поддерживает только программное обеспечение продуктов Apple.

#### Программы установки и архиваторы

Некоторые [архиваторы](#) (например [WinRAR](#), [WinZip](#), [7-Zip](#) и другие) также позволяют создавать установщики (самораспаковывающийся архив).

Некоторые операционные среды — например, Windows NT (по умолчанию) и [Xfce 4](#) (по желанию пользователя), содержат функцию автоматического запуска определённой программы, находящейся на носителе, при его вставке в устройство чтения.

Программы по их юридическому статусу можно разделить на три большие группы:

- Лицензионные
- условно бесплатные (shareware)
- свободно распространяемые программы (freeware).

Дистрибутивы лицензионных программ продаются пользователям. В соответствии с лицензионным соглашением разработчики программы гарантируют ее нормальное функционирование в определенной операционной системе и несут за это ответственность.

Условно бесплатные программы предлагаются пользователям в целях их рекламы и продвижения на рынок. Пользователю предоставляется версия программы с ограниченным сроком действия (после истечения указанного срока программа перестает работать, если за нее не произведена оплата) или версия программы с ограниченными функциональными возможностями (в случае оплаты пользователю сообщается код, включающий все функции).

Производители бесплатного программного обеспечения заинтересованы в его широком распространении. К таким программным средствам можно отнести следующие:

- новые недоработанные (бета) версии программных продуктов
- программные продукты, являющиеся частью принципиально новых технологий
- дополнения к ранее выпущенным программам, исправляющие найденные
- устаревшие версии программ;
- драйверы к новым устройствам или улучшенные драйверы к уже существующим.

Производители программного обеспечения предлагают пользователям лицензионное соглашение. Принимая настоящее соглашение, Вы выражаете свое полное согласие со всеми его положениями и условиями. Если Вас не устраивают условия, описанные в нем, то не устанавливайте программу. Использование программы однозначно подразумевает принятие Вами всех положений и условий данного соглашения. Принятие положений и условий настоящего соглашения не является передачей, каких бы то ни было прав собственности на программы и продукты.

#### Пример



Чтобы ознакомиться с процессом установки программного обеспечения, запустите видео ролик Установка антивируса Avast.mpg



Все осознают необходимость надёжной защиты компьютера от вирусов и других опасных программ. Каждый квартал по миру прокатывается очередная новая волна компьютерной инфекции, вирусы, черви, трояны постоянно ведут свою деятельность целью которой является инфицирование всё новых и новых компьютеров. Только надёжная антивирусная система в состоянии противостоять этой атаке.

Чтобы защитить свой компьютер или ноутбук нужна надёжная антивирусная программа. Но стоимость такого программного обеспечения (особенно надёжного и качественного) довольно высока. Можно использовать бесплатные версии антивирусов. Бесплатных антивирусных программ довольно много, но очень много нареканий на их надёжность, если с устаревшими вирусами они ещё могут справиться, то новые инфекции в большинстве проходят их защиту без проблем.

Основная опасность в инфицировании компьютера опасными программами - эта порча или потеря пользовательских данных, в редких случаях возможен и выход из строя компьютера или его составных частей. Поэтому защищаться обязательно необходимо.

Домашняя версия антивируса AVAST Home Edition отличается от коммерческой только типом лицензии (доступна для не коммерческого домашнего использования) и меньшим количеством настроек (что даже больше плюс чем минус). Она обеспечивает комплексную защиту вашего компьютера от различных видов инфекций распространяющихся различными путями. Кроме того антивирус AVAST Home Edition

имеет обновляемую антивирусную базу (обновление происходит автоматически при подключении к интернету) что даёт возможность защищаться и от новых вирусов. Такие возможности бывают только у платных антивирусов, но AVAST Home Edition – это исключение.

Для бесплатного использования антивируса AVAST Home Edition с возможностью его обновления на протяжении целого года достаточно пройти бесплатную регистрацию и скачать Avast бесплатно Home Edition.

1. Зайдите на сайт Avast! (<http://www.avast.ru>)
2. Установите на свой компьютер антивирусную программу для домашнего использования, размещая пиктограммы этого антивируса на **Рабочем столе**.
3. Сохраните скриншот рабочего стола с пиктограммами установочного файла антивируса и пиктограммой самого антивируса с именем **Virus.jpg**.
4. Отошлите скриншот **Virus.jpg** преподавателю.

#### Содержание отчета

- Название работы;
- Цель работы;
- Формулировку задания;
- Решенные задания, с пояснением решения в виде таблиц, рассуждений, рисунков.

#### Контрольные вопросы

1. Что такое дистрибутив?
2. На какие группы программ делится программное обеспечение ПК?
3. Перечислите программы относящиеся к системным.
4. Какие программы называют прикладными?
5. Какие функции реализуют инструментальные программы?
6. Что такое утилиты?
7. Для чего предназначены драйвера?
8. Какую функцию выполняют упаковщики?
9. С помощью каких устройств программы переводятся на один из языков понятный для ЭВМ?

10. Каково назначение программ тестирования, контроля и диагностики?
11. В чем сходство и в чем различие компилятора и интерпретатора?
12. Какие программы содержит библиотека стандартных подпрограмм?

Список литературы:

Основная литература

1. Максимов Н.В., Партыка Т. Л. Попов И. И. Архитектура ЭВМ и вычислительных систем: учебник. – М.: Форум, 2008.
2. Келим Ю.М. Вычислительная техника: учеб. Пособие для студ. сред. проф. образования/ Ю. М. Келим. – М.: Издательский центр «Академия», 2008.

Дополнительная литература

3. Таненбаум Э. Архитектура компьютера – СПб: Питер, 2010

## Практическая работа № 15

Тема: Установка и настройка программного обеспечения компьютерных систем

Цель: закрепление умений по установке и настройке программного обеспечения компьютерных систем

Оборудование: Персональный компьютер

Задание:

### 1 вариант

1. Установить на ПК программу **Wise Calculator**

Калькулятор для инженерных и научных вычислений с очень большими возможностями.

Сайт программы: <http://www.wisecalculator.chat.ru/rus.html>

2. Опишите этапы процесса установки и удаления программы
3. Удалите программу **Wise Calculator**
4. Опишите этапы процесса удаления программы
5. Изучить основные термины программного обеспечения

### 2 вариант

1. Установить на ПК программу **Stamina**

Программа-тренажёр для тех, кто хочет научиться набирать текст на клавиатуре всеми десятью пальцами. Сайт программы: <http://stamina.ru/>

2. Опишите этапы процесса установки программы
3. Удалите программу **Stamina**
4. Опишите этапы процесса удаления программы
5. Изучить основные термины программного обеспечения

### 3 вариант

1. Установить на ПК программу **CCleaner**

CCleaner - утилита для чистки мусора в операционной системе.

Домашняя страница: <http://www.ccleaner.com/>

2. Опишите этапы процесса установки программы
3. Удалите программу **CCleaner**
4. Опишите этапы процесса удаления программы
5. Изучить основные термины программного обеспечения

### 4 вариант

1. Установить на ПК программу **7-Zip**

Архиватор с высокой степенью сжатия.

Домашняя страница <http://www.7-zip.org/>

2. Опишите этапы процесса установки программы
3. Удалите программу **CCleaner**
4. Опишите этапы процесса удаления программы
5. Изучить основные термины программного обеспечения

## 5 вариант

1. Установить на ПК программу *Avira AntiVir PersonalEdition Classic*  
AntiVir Personal Edition - это антивирусная программа, которая может определять и удалять вирусы.

Домашняя страница: <http://www.free-av.com/>

2. Опишите этапы процесса установки программы
3. Удалите программу *CCleaner*
4. Опишите этапы процесса удаления программы
5. Изучить основные термины программного обеспечения

## 6 вариант

1. Установить на ПК программу *Dr.WEB CureIt!*

Это бесплатная антивирусная утилита на основе сканера Dr.WEB, которая быстро и эффективно проверит и вылечит, в случае необходимости

Домашняя страница: <http://www.freedrweb.com/>

2. Опишите этапы процесса установки программы
3. Удалите программу *Dr.WEB CureIt!*
4. Опишите этапы процесса удаления программы
5. Изучить основные термины программного обеспечения

## 7 вариант

1. Установить на ПК программу *Кто хочет стать ОТЛИЧНИКОМ?!*

Игровая система тестирования.

Сайт программы: <http://otlichnik5.net/>

2. Опишите этапы процесса установки программы
3. Удалите программу *Кто хочет стать ОТЛИЧНИКОМ?!*
4. Опишите этапы процесса удаления программы

## Содержание отчета

- Название работы;
- Цель работы;
- Назначение программного продукта, устанавливаемого на ПК;
- Создайте инструкцию по установке, указанного в варианте ПО;

## Контрольные вопросы

1. К какой группе программ относится установленное вами ПО?
2. Перечислите технические характеристики ПК, на который возможна установка, установленного вами ПО?
3. Перечислите правила ТБ при работе на ПК.

## Список литературы:

## Основная литература

1. Максимов Н.В., Партыка Т. Л. Попов И. И. Архитектура ЭВМ и вычислительных систем: учебник. – М.: Форум, 2008.
2. Келим Ю.М. Вычислительная техника: учеб. Пособие для студ. сред. проф. образования/ Ю. М. Келим. – М.: Издательский центр «Академия», 2008.

## Дополнительная литература

3. Таненбаум Э. Архитектура компьютера – СПб: Питер, 2010