

Автономная некоммерческая организация профессионального образования
«ПЕРМСКИЙ ГУМАНИТАРНО-ТЕХНОЛОГИЧЕСКИЙ КОЛЛЕДЖ»
(АНО ПО «ПГТК»)



ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
ПО УЧЕБНОЙ ДИСЦИПЛИНЕ
ОП.02 Архитектура компьютерных систем
для специальности
09.02.03 Программирование в компьютерных системах
(код и наименование специальности)

Квалификация выпускника
Техник-программист
(базовая подготовка)

Форма обучения
Очная

Пермь, 2020 г

Фонд оценочных средств дисциплины «ОП.02 Архитектура компьютерных систем» составлен в соответствии с требованиями Федерального государственного образовательного стандарта среднего профессионального образования по специальности 09.02.03 Программирование в компьютерных системах (утвержден приказом Министерства образования и науки Российской Федерации от 28.07.2014 г., № 804).
Предназначен для студентов и преподавателей АНО ПО «ПГТК».

Автор – составитель: Долганова Я.А., старший преподаватель.

Фонд оценочных средств учебной дисциплины рассмотрен и одобрен на заседании кафедры математических и естественно-научных дисциплин, протокол, № 06 от «6» февраля 2020 г.

Рекомендован к утверждению педагогическим советом АНО ПО «ПГТК» (протокол от «21» февраля 2020г. №3)

Оглавление

1. Паспорт фонда оценочных средств	6
2. Результаты освоения учебной дисциплины, подлежащие проверке	7
2.1. В результате аттестации по учебной дисциплине осуществляется комплексная проверка следующих умений и знаний, а также динамика формирования общих компетенций:	7
3. Оценка освоения учебной дисциплины:	9
3.1. Формы и методы оценивания.....	9
3.2. Типовые задания для оценки освоения учебной дисциплины	3
3.2.1. Типовые задания для оценки знаний.	3
Эталон ответов к контрольным заданиям.....	19
Эталоны ответов к полугодовой срезовой контрольной работе	23
2) Практические занятия:	26
Арифметические операции над числами с фиксированной точкой	32
Арифметические операции над двоичными числами с плавающей точкой ..	35
Задание на выполнение лабораторной работы	145
Экзаменационные билеты по дисциплине.....	175

1. Паспорт фонда оценочных средств

В результате освоения учебной дисциплины ОП.02 Архитектура компьютерных систем обучающийся должен обладать предусмотренными ФГОС по специальности СПО 09.02.03 «Программирование в компьютерных системах», квалификация – техник-программист следующими умениями, знаниями, которые формируют профессиональную компетенцию, и общими компетенциями:

У1. Получать информацию о параметрах компьютерной системы;

У2. Подключать дополнительное оборудование и настраивать связь между элементами компьютерной системы;

У3. Производить установку и настройку программного обеспечения компьютерных систем.

З1. Базовые понятия и основные принципы построения архитектур вычислительных систем;

З2. Типы вычислительных систем и их архитектурные особенности;

З3. Организацию и принцип работы основных логических блоков компьютерных систем;

З4. Процессы обработки информации на всех уровнях компьютерных архитектур;

З5. Основные компоненты программного обеспечения компьютерных систем;

З6. Основные принципы управления ресурсами и организации доступа к этим ресурсам.

ОК 1	Понимать сущность и социальную значимость своей будущей профессии, проявлять к ней устойчивый интерес
ОК 2	Организовать 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 результаты 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

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

Формой аттестации по учебной дисциплине является **дифференциальный зачет**.

2. Результаты освоения учебной дисциплины, подлежащие проверке

2.1. В результате аттестации по учебной дисциплине осуществляется комплексная проверка следующих умений и знаний, а также динамика формирования общих компетенций:

Таблица 1.1

Результаты обучения: умения, знания и общие компетенции	Показатели оценки результата	Форма контроля и оценивания
Уметь:		
У1. Получать информацию о параметрах компьютерной системы;	Мотивированное обоснование выбора и применения методов и способов решения профессиональных задач. Точность, правильность и полнота выполнения профессиональных задач.	1. Опрос фронтальный, индивидуальный. 2. Тестовое задание. 3. Самостоятельная работа.
У2. Подключать дополнительное оборудование и настраивать связь между элементами компьютерной системы;	Демонстрация способности принимать решения в стандартных и нестандартных ситуациях и нести за них ответственность. Использование специализированных программных средств отладки программных модулей.	1. Опрос фронтальный, индивидуальный. 2. Тестовое задание. 3. Самостоятельная работа.
У3. Управлять учетными записями, настраивать параметры рабочей среды пользователя;	Обоснованность выбора информационных источников для решения профессиональных задач. Оперативность поиска и использования необходимой информации для качественного выполнения профессиональных задач, профессионального и личностного развития.	1. Опрос фронтальный, индивидуальный. 2. Тестовое задание. 3. Самостоятельная работа.
Знать:		
З1. Базовые понятия и основные принципы построения архитектур вычислительных систем;	Выполнять конфигурирование аппаратных устройств; управлять учетными записями, настраивать параметры рабочей среды пользователя; - управлять дисками и файловыми системами	1. Практические занятия 2. Устный опрос 3. Творческие работы (подготовка презентаций, кроссвордов)

32. Типы вычислительных систем и их архитектурные особенности;	Использование специализированных программных средств отладки программных модулей. Разработка компонент программных модулей с использованием современных инструментальных средств и технологий.	1. Практические занятия 2. Устный опрос 3. Творческие работы (подготовка презентаций, кроссвордов)
33. Организацию и принцип работы основных логических блоков компьютерных систем;	Тестирование и отладка программного средства. Использования инструментальных средств на этапе отладки программного продукта. Сопровождение программного обеспечения компьютерных систем	1. Практические занятия 2. Устный опрос 3. Творческие работы (подготовка презентаций, кроссвордов)
34. Процессы обработки информации на всех уровнях компьютерных архитектур;	Выполнять конфигурирование аппаратных устройств; управлять учетными записями, настраивать параметры рабочей среды пользователя; - управлять дисками и файловыми системами	1. Практические занятия 2. Устный опрос 3. Творческие работы (подготовка презентаций, кроссвордов)
35. Основные компоненты программного обеспечения компьютерных систем;	Использование специализированных программных средств отладки программных модулей. Разработка компонент программных модулей с использованием современных инструментальных средств и технологий.	1. Практические занятия 2. Устный опрос 3. Творческие работы (подготовка презентаций, кроссвордов)
36. Основные принципы управления ресурсами и организации доступа к этим ресурсам.	Тестирование и отладка программного средства. Использования инструментальных средств на этапе отладки программного продукта. Сопровождение программного обеспечения компьютерных систем	1. Практические занятия 2. Устный опрос 3. Творческие работы (подготовка презентаций, кроссвордов)

3. Оценка освоения учебной дисциплины:

3.1. Формы и методы оценивания

Предметом оценки служат умения и знания, предусмотренные ФГОС по дисциплине ОП. 02 Архитектура компьютерных систем, направленные на формирование общих и профессиональных компетенций.

Элемент учебной дисциплины	Текущий контроль		Промежуточная аттестация	
	Форма контроля	Проверяемые ОК, У, З, ПК	Форма контроля	Проверяемые ОК, У, З
Раздел 1 Представление информации в вычислительных системах			<i>Дифференциальный зачет</i>	У1, У2, У3 З 1, 32, 33, 34, 35, 36 ОК 1- 9 ПК 1,1, ПК 1,2, ПК 1,5, ПК2,3, ПК 2,4, ПК3,1, ПК3,2, ПК 3,4
Тема 1.1. Базовые понятия и основные принципы построения архитектур вычислительных систем.	Устный опрос Самостоятельная работа Практическая работа №1 Практическая работа №2 Практическая работа №3 Практическая работа №4 Практическая работа №5 Практическая работа №6 Практическая работа №7 Практическая работа №8	У1, У2, У3 З 1, 32, 33, 34, 35, 36 ОК 1- 9 ПК1,1, ПК 1,2, ПК 1,5, ПК 2,3, ПК 2,4, ПК 3,1, ПК 3,2, ПК 3,4		
Тема 1.2. Типы вычислительных систем и их архитектурные особенности.	Устный опрос Практическая работа №9 Практическая работа №10 Практическая работа №11 Практическая работа №12 Самостоятельная работа	У1, У2, У3 З 1, 32, 33, 34, 35, 36 ОК 1- 9 ПК1,1, ПК 1,2, ПК 1,5, ПК 2,3, ПК 2,4, ПК 3,1, ПК 3,2, ПК 3,4		
Раздел 2 Архитектура и принципы работы основных логических блоков вычислительных систем			<i>Дифференциальный зачет</i>	У1, У2, У3 З 1, 32, 33, 34, 35, 36 ОК 1- 9 ПК 1,1, ПК 1,2, ПК 1,5, ПК2,3, ПК 2,4, ПК3,1, ПК3,2, ПК 3,4
Тема 2.1 Организация и принцип работы основных логических блоков компьютерных систем.	Устный опрос Практическая работа №13 Практическая работа №14 Практическая работа №15 Практическая работа №16	У1, У2, У3 З 1, 32, 33, 34, 35, 36 ОК 1- 9 ПК 1,1, ПК 1,2, ПК 1,5, ПК2,3, ПК 2,4, ПК3,1, ПК3,2, ПК 3,4		

	<i>Практическая работа №17 Самостоятельная работа</i>			
Тема 2.2 Логические элементы, узлы, блоки и устройства компьютера	<i>Устный опрос Практическая работа №18 Практическая работа №19 Практическая работа №20 Практическая работа №21 Практическая работа №22 Практическая работа №23 Самостоятельная работа</i>	<i>У1, У2, У3 3 1, 32, 33, 34, 35, 36 ОК 1- 9 ПК 1,1, ПК 1,2,ПК 1,5, ПК2,3, ПК 2,4, ПК3,1, ПК3,2, ПК 3,4</i>		
Тема 2.3. Процессы обработки информации на всех уровнях компьютерных архитектур.	<i>Устный опрос Практическая работа №24 Практическая работа №25 Практическая работа №26 Практическая работа №27 Практическая работа №28 Практическая работа №29 Практическая работа №30 Практическая работа №31 Самостоятельная работа</i>	<i>У1, У2, У3 3 1, 32, 33, 34, 35, 36 ОК 1- 9 ПК 1,1, ПК 1,2,ПК 1,5, ПК2,3, ПК 2,4, ПК3,1, ПК3,2, ПК 3,4</i>		
Тема 2.4 Организация работы памяти компьютера	<i>Устный опрос Практическая работа №32 Практическая работа №33 Практическая работа №34 Практическая работа №35 Практическая работа №36 Практическая работа №37 Практическая работа №38 Практическая работа №39 Практическая работа №40 Практическая работа №41 Практическая работа №42 Самостоятельная работа</i>	<i>У1, У2, У3 3 1, 32, 33, 34, 35, 36 ОК 1- 9 ПК 1,1, ПК 1,2,ПК 1,5, ПК2,3, ПК 2,4, ПК3,1, ПК3,2, ПК 3,4</i>		
Раздел 3 Управление внешними устройствами			<i>Дифференциальный зачет</i>	<i>У1, У2, У3 3 1, 32, 33, 34, 35, 36 ОК 1- 9 ПК 1,1, ПК 1,2,ПК 1,5, ПК2,3, ПК 2,4</i>

				ПК3,1, ПК3,2, ПК 3,4
Тема 3.1 Принципы управление внешними устройствами	Устный опрос Практическая работа №43 Практическая работа №44 Практическая работа №45 Практическая работа №46 Практическая работа №47 Практическая работа №48 Самостоятельная работа	У1, У2, У3 3 1, 32, 33, 34, 35, 36 ОК 1- 9 ПК 1,1, ПК 1,2,ПК 1,5, ПК2,3, ПК 2,4, ПК3,1, ПК3,2, ПК 3,4		
Тема 3.2 Внешние устройства ЭВМ	Устный опрос Практическая работа №49 Практическая работа №50 Практическая работа №51 Практическая работа №52 Самостоятельная работа	У1, У2, У3 3 1, 32, 33, 34, 35, 36 ОК 1- 9 ПК 1,1, ПК 1,2,ПК 1,5, ПК2,3, ПК 2,4, ПК3,1, ПК3,2, ПК 3,4		
Тема 3.3 Внешние запоминающие устройства.	Устный опрос Практическая работа №53 Практическая работа №54 Практическая работа №55 Самостоятельная работа			
Раздел 4. Программное обеспечение.			Дифференциальный зачет	У1, У2, У3 3 1, 32, 33, 34, 35, 36 ОК 1- 9 ПК 1,1, ПК 1,2,ПК 1,5, ПК2,3, ПК 2,4, ПК3,1, ПК3,2, ПК 3,4
Тема 4.1 Основные компоненты программного обеспечения компьютерных систем.	Устный опрос Практическая работа №56 Практическая работа №57 Практическая работа №58 Практическая работа №59 Самостоятельная работа	У1, У2, У3 3 1, 32, 33, 34, 35, 36 ОК 1- 9 ПК 1,1, ПК 1,2,ПК 1,5, ПК2,3, ПК 2,4, ПК3,1, ПК3,2, ПК 3,4		
Тема 4.2 Системы автоматизации программирования	Устный опрос Самостоятельная работа	У1, У2, У3 3 1, 32, 33, 34, 35, 36 ОК 1- 7		
Раздел 5. Вычислительные системы.			Дифференциальный зачет	У1, У2, У3 3 1, 32, 33, 34, 35, 36 ОК 1- 9 ПК 1,1, ПК 1,2,ПК 1,5, ПК2,3, ПК 2,4, ПК3,1, ПК3,2, ПК 3,4

Тема 5.1 Основные принципы управления ресурсами и организация доступа к этим ресурсам.	<i>Устный опрос</i> <i>Практическая работа №60</i> <i>Самостоятельная работа</i>			
--	--	--	--	--

3.2. Типовые задания для оценки освоения учебной дисциплины

3.2.1. Типовые задания для оценки знаний.

1) Тестовые задания:

Вариант № 1

1. Что такое основная память компьютера?
2. Дать определение понятию «триггер»?
3. Перечислить назначение и основные характеристики ОЗУ.
4. Что такое ассоциативная память компьютера?
5. Описать виды больших интегральных схем ПЗУ.

Вариант № 2

1. Что кэш - память компьютера?
2. Дать определение понятию «сумматор»?
3. Перечислить назначение и основные характеристики ПЗУ.
4. Что такое расслоение памяти?
5. Описать виды больших интегральных схем ОЗУ.

Вариант 1

1 задание. Выберите правильные ответы

1. Дощечка покрытая слоем пыли, на которой острой палочкой проводились линии и выкладывались какие-нибудь предметы называлась:

- а. вестоницкая кость;
- б. абак;
- в. соробан;
- г. костяшки Непера.

2. Как называется система счисления, в которой каждая цифра имеет одно и тоже значение независимо от положения в записи числа?

- а. позиционная;
- б. непозиционная;
- в. арабская;
- г. римская.

3. Логический элемент ЭВМ для сложения чисел:

- а. триггер;
- б. сумматор;
- в. дешифратор;
- г. шифратор.

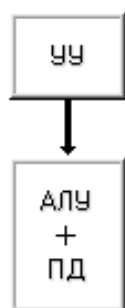
4. Количество бит, обрабатываемых процессором за один прием:

- а. система команд;
- б.быстродействие;
- в.максимальный объем адресуемой памяти;
- г. разрядность.

5. Память ЭВМ – это:

- а. процессор, который является «мозгом» компьютера;
- б. совокупность всех запоминающих устройств ЭВМ;
- в. совокупность триггеров для запоминания информации;
- г. место, для хранения ненужной информации.

6. Вид памяти, которая предназначена для промежуточного хранения информации при обмене данными между устройствами ЭВМ:
- а. ОЗУ;
 - б. ПЗУ;
 - в. БЗУ;
 - г. ППЗУ.
7. Отношение емкости запоминающего устройства к его физическому объему называется:
- а. емкостью;
 - б. удельной емкостью;
 - в. быстродействием;
 - г. оперативностью.
8. Микроканальная архитектура, несовместимая с ISA/EISA, ориентированная на асинхронное функционирование шины и процессора
- а. ISA;
 - б. MCA;
 - в. PCI;
 - г. EISA.
9. Оптическая мышь - ...
- а. движение фиксируется механически и связано с перемещением частей устройств.
 - б. движение шарика отслеживается с помощью двух валиков с прорезями и двух оптических пар светодиод-фотодиод.
 - в. движение отслеживается с помощью двух пар светодиодов и фотоэлементов.
 - г. это стержень-ручка, отклонение которой от вертикального положения приводит к передвижению курсора в соответствующем направлении по экрану монитора.
10. Ручной сканер...
- а. Оригинал автоматически перемещается относительно сканирующей головки, часто имеется автоматическая подача документов
 - б. Внешне напоминают фотоувеличитель: внизу лежит сканируемый документ, а наверху находится сканирующая головка
 - в. Бумажный лист с изображением или текстом кладется на прозрачную стеклянную поверхность, под которой проходит распознающий элемент сканера, и закрывается крышкой
 - г. Прокатывают по поверхности документа рукой
11. Предложения в языке программирования Ассемблер: внутри идентификаторов и чисел пробелы
- а. возможны;
 - б. недопустимы;
 - в. обязательны;
 - г. допустимы.



12.

- а. Классификация Хендлера;
- б. Классификация Шора;
- в. Классификация Хокни;
- г. Классификация Скилликорна.

13. Архитектура суперкомпьютера, в которой каждый процессор имеет свою оперативную память:

- а. параллельная мультимикропроцессорная обработка;
- б. асимметричная мультимикропроцессорная обработка;
- в. симметричная мультимикропроцессорная обработка;
- г. последовательная мультимикропроцессорная обработка.

2 задание. Перевести числа из одной системы счисления в другую:

$$124,35_{10} - X_8;$$

$$1010110101_2 - X_{16};$$

$$46,2_8 - X_{10}$$

Вариант 2

1 задание. Выберите правильные ответы

1. Персональные компьютеры, на которых работали пользователи с общей компьютерной подготовкой, находясь за своим рабочим столом, относятся к:

- а. Первому поколению;
- б. Второму поколению;
- в. Третьему поколению;
- г. Четвертому поколению.

2. Базовые цифры 16-ричной системы счисления:

- а. 0-15;
- б. 0-9, А-Е;
- в. 0-9, А.-G;
- г. 1-16.

3. Комбинационная схема с несколькими входами и выходами, преобразующая код, подаваемый на вход, в сигнал на одном из выходов:

- а. триггер;
- б. сумматор;
- в. дешифратор;
- г. шифратор.

4. Классификация арифметическо-логического устройства по структуре (возможно несколько вариантов):

- а. с непосредственными связями;

- б. многосвязные;
- в. блочные;
- г. многофункциональные.

5. На ЭВМ с памятью в 4 Мбайт можно:

- а. слушать современную музыку;
- б. работать в среде WindowsXP;
- в. работать в среде MS DOS и простейших текстовых редакторах;
- г. ничего нельзя делать.

6. Вид памяти, информация из которой может «стекать»:

- а. динамическая память;
- б. статическая память;
- в. генерированная память;
- г. постоянная память.

7. Запоминающее устройство, включаемое между ОЗУ и процессором:

- а. ПЗУ;
- б. ППЗУ;
- в. СОЗУ;
- г. БЗУ.

8. Шина, поддерживающая режим Plug&Play, скоростной режим пересылки пакетов данных, однозначно определяющая устройства, используемая в основном на файл-серверах:

- а. ISA;
- б. EISA;
- в. PCI;
- г. MCA.

9. Печатающие устройства бывают (возможно несколько вариантов):

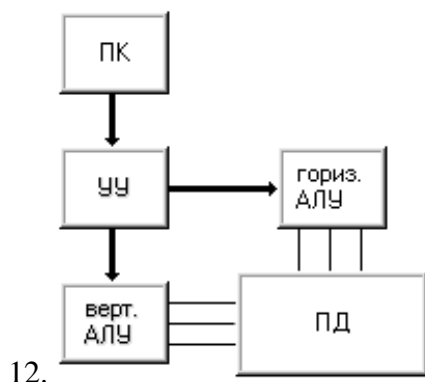
- а. посимвольные,
- б. построчные,
- в. постраничные,
- г. познаковые.

10. Плоттер - ...

- а. устройство для вывода информации из компьютера
- б. устройства для оцифровки и ввода в компьютер изображений с бумажных копий
- в. электронное устройство, преобразующее графический образ, хранящийся, как содержимое памяти компьютера (или самого адаптера), в форму, пригодную для дальнейшего вывода на экран монитора
- г. устройство, которое чертит графики, рисунки или диаграммы под управлением компьютера.

11. Предложения в языке программирования Ассемблер: Переносить предложение на следующую строку или записывать два предложения на одной строке:

- а. возможно;
- б. нельзя;
- в. обязательно;
- г. иногда.



- а. Классификация Хендлера;
- б. Классификация Шора;
- в. Классификация Хокни;
- г. Классификация Скилликорна.

13. Архитектура суперкомпьютера, в которой группа процессоров работает с общей оперативной памятью:

- а. параллельная мультипроцессорная обработка;
- б. асимметричная мультипроцессорная обработка;
- в. симметричная мультипроцессорная обработка;
- г. последовательная мультипроцессорная обработка.

2 задание. Сложить числа в двоичной и десятичной системах счисления:

- а) $100101011_2 + 10011100_2 - X_2$;
- б) $37_8 + 25_8 - X_{10}$

Вариант 3

1 задание. Выберите правильные ответы

1. Большие ламповые ЭВМ с однопрограммным режимом работы, на которых работали инженеры-программисты, находясь в машинном зале, относятся к:

- а. Первому поколению;
- б. Второму поколению;
- в. Третьему поколению;
- г. Четвертому поколению.

2. Десятичное число 0.65 в двоичной системе счисления равно (с точностью -5 знаков после запятой):

- а. 0,10101;
- б. 0,10100;
- в. 0,01101;
- г. 0,00101.

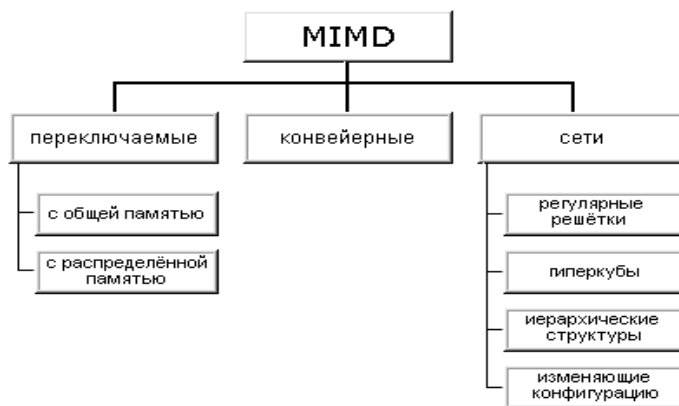
3. Устройство, выполняющее преобразование позиционного кода в n-разрядный двоичный код

- а. триггер;
- б. сумматор;
- в. дешифратор;
- г. шифратор.

4. Классификация арифметическо-логического устройства по организации действий над операндами (возможно несколько вариантов):
- последовательного действия;
 - параллельного действия;
 - блочные;
 - многофункциональные.
5. Память предназначена для:
- хранения данных;
 - удерживания разрядов в состоянии «0»;
 - помещения туда ненужной информации;
 - можно обойтись без нее.
6. Вид памяти, в которой информацию можно изменять с помощью специальной аппаратуры:
- ОЗУ;
 - ПЗУ;
 - ППЗУ;
 - БЗУ.
7. Вид памяти, которая предназначена для размещения данных, видимых на экране:
- кэш-память;
 - местная память;
 - видеопамять;
 - БЗУ.
8. Шина, дающая возможность реализовать многопользовательский и многозадачный режимы работы процессора, получать доступ к контроллеру VGA, имеющая повышенную плотность размещения выводов микросхем и разъемов:
- MCA;
 - EISA;
 - PCI;
 - USB.
9. ...бывают ручные, планшетные, барабанные.
- Принтеры
 - Сканеры
 - Плоттеры
 - Адаптеры
10. Сканер - ...
- устройство для вывода информации из компьютера
 - устройства для оцифровки и ввода в компьютер изображений с бумажных копий
 - электронное устройство, преобразующее графический образ, хранящийся, как содержимое памяти компьютера (или самого адаптера), в форму, пригодную для дальнейшего вывода на экран монитора
 - устройство, которое чертит графики, рисунки или диаграммы под управлением компьютера.
11. Язык программирования Ассемблер: Команды сложения и вычитания (возможно несколько вариантов):
- ADD;

- б. INC;
- в. ADC;
- г. NEG.

12.



- а. Классификация Хендлера;
- б. Классификация Джонсона;
- в. Классификация Хокни;
- г. Классификация Скилликорна.

13. Параллельная архитектура с векторными процессорами:

- а. PVP-архитектура;
- б. асимметричная архитектура;
- в. симметричная архитектура;
- г. последовательная архитектура.

2 задание. Построить таблицу состояния по следующей функции:

$$F = \overline{X \& Y} \vee (\overline{X \vee Y}) \& X$$

Вариант 4

1 задание. Выберите правильные ответы

1. Определите признак, по которому информация делится на элементарную, биологическую, социальную:
 - а. По способу передачи и восприятия;
 - б. По общественному назначению;
 - в. По области возникновения;
 - г. По статусу.
2. В восьмеричной системе счисления для записи чисел используются цифры:
 - а. 0, 1, 2, 3, 4, 5, 6, 7;
 - б. 1, 2, 3, 4, 5, 6, 7, 8;
 - в. -4, -3, -2, -1, 1, 2, 3, 4;
 - г. 0, 8.
3. Электронные схемы с одним или несколькими входами и одним выходом, через которые проходят электрические сигналы, представляющие 0,1 - это
 - а. логические элементы;
 - б. логические узлы;

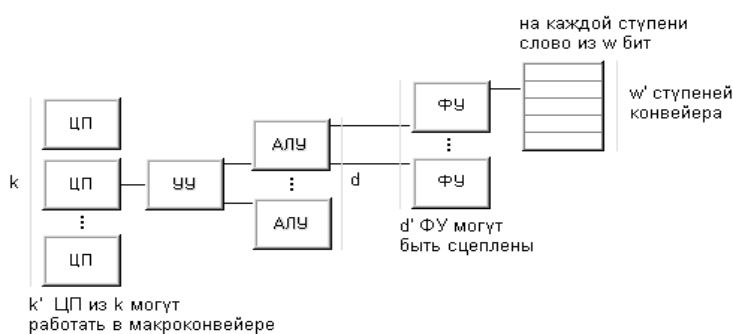
- в. логические блоки;
 - г. логические устройства.
4. Классификация арифметическо-логического устройства по виду обрабатываемых чисел (возможно несколько вариантов):
- а. последовательного действия;
 - б. параллельного действия;
 - в. двоичные;
 - г. двоично-десятичные.
5. На ЭВМ с памятью в 8 Мбайт можно:
- а. работать в среде OS/2 и MSDOS;
 - б. смотреть видеофильмы и играть в сложные игры;
 - в. использовать для серверов локальных сетей;
 - г. ничего нельзя.
6. Метод организации оперативной памяти, при которой память адресуется по границам страниц:
- а. метод страничной организации;
 - б. метод статических колонок;
 - в. метод чередования адресов;
 - г. пограничный метод.
7. Вид памяти, которая представляет собой цепочку регистров и устроена по принципу «последний пришел – первый ушел»:
- а. ПЗУ;
 - б. стек;
 - в. ППЗУ;
 - г. БЗУ.
8. Шина, которая является портом ускоренного графического вывода:
- а. VLB;
 - б. IDE;
 - в. AGP;
 - г. PCI.
9. Проекционные сканеры...
- а. Оригинал автоматически перемещается относительно сканирующей головки, часто имеется автоматическая подача документов
 - б. Внешне напоминают фотоувеличитель: внизу лежит сканируемый документ, а наверху находится сканирующая головка
 - в. Бумажный лист с изображением или текстом кладется на прозрачную стеклянную поверхность, под которой проходит распознающий элемент сканера, и закрывается крышкой
 - г. Прокатывают по поверхности документа рукой
10. Жидкокристаллические мониторы (возможно несколько вариантов)
- а. Основной элемент дисплея — электронно-лучевая трубка.
 - б. Заряды передаются через пассивную матрицу — сетку невидимых нитей, горизонтальных и вертикальных, создавая в месте пересечения нитей точку изображения

в.Его передняя, обращенная к зрителю часть с внутренней стороны покрыта люминофором — специальным веществом, способным излучать свет при попадании на него быстрых электронов

Г.Создаются путем заполнения пространства между двумя стеклянными поверхностями инертным газом, например аргоном или неоном.

11. Язык программирования Ассемблер: Команды умножения и деления(возможно несколько вариантов):

- а. MUL;
- б. DEC;
- в. IMUL;
- г. IDIV.



12.

- а. Классификация Хендлера;
- б. Классификация Джонсона;
- в. Классификация Кришнамарфи;
- г. Классификация Скилликорна.

13. Возможность наращивания числа и мощности процессоров, объемов оперативной и внешней памяти и других ресурсов вычислительной системы:

- а. операбельность;
- б. оперативность;
- в. кластер;
- г. масштабируемость.

2 задание. Перевести числа из одной системы счисления в другую:

- а) $97_{10} - X_{16}$;
- б) $6A,1_{16} - X_{10}$;
- с) $134,2_8 - X_2$

Вариант 5

1 задание. Выберите правильные ответы

1. Определите признак, по которому информация делится на массовую, личную, специальную:

- а. По способу передачи и восприятия;
- б. По общественному назначению;
- в. По области возникновения;
- г. По статусу.

2. Количество различных чисел, используемых для изображения чисел в данной системе счисления:

- а. Основание системы счисления;
- б. Позиция системы счисления;

- в. слово системы счисления;
- г. знаки системы счисления.

3. Комбинированные связи, входящие в состав логических схем - это

- а. Триггеры;
- б. Вентили;
- в. Регистры;
- г. Счетчики.

4. Классификация арифметическо-логического устройства по способу действий над операндами (возможно несколько вариантов):

- а. последовательного действия;
- б. параллельного действия;
- в. двоичные;
- г. двоично-десятичные.

5. На ЭВМ с памятью в 32 Мбайта и более можно:

- а. работать в среде WindowsXP;
- б. работать в любых средах и с любыми приложениями;
- в. использовать для серверов локальных сетей;
- г. ничего нельзя.

6. Метод организации оперативной памяти, при которой информация записывается по блокам:

- а. метод строк/колонок;
- б. метод статических колонок;
- в. метод чередования адресов;
- г. блочный метод.

7. Вид памяти, поиск в которой производится по ее содержимому (по ключевому слову):

- а. ассоциативная память;
- б. поисковая память;
- в. адресная память;
- г. словарная память.

8. Порт для одновременной передачи 8 бит информации:

- а. последовательный;
- б. параллельный;
- в. USB;
- г. IDE.

9. Принтеры можно разделить на группы по принципу действия (возможно несколько вариантов):

- а. матричные,
- б. термические,
- в. струйные,
- г. лазерные.

10. Разрядность сканера измеряется в...

- а. Гц
- б. дюймах
- в. байтах

г. битах

11. Язык программирования Ассемблер: Команды сложения и вычитания(возможно несколько вариантов):

- а. SUB;
- б. DEC;
- в. SBB;
- г. NEG.

12. Для классификации параллельных вычислительных систем предлагает использовать четыре характеристики: степень гранулярности; способ реализации параллелизма; топология и природа связи процессоров; способ управления процессорами.

- а. Классификация Базу;
- б. Классификация Джонсона;
- в. Классификация Кришнамарфи;
- г. Классификация Скилликорна.

13. Наиболее распространенными из систем класса SIMD являются ...системы, которые наиболее приспособлены для решения задач, характеризующихся параллелизмом независимых объектов или данных.

- а. клеточные;
- б. ассоциативные;
- в. матричные
- г. ДНК.

2 задание. Сложить числа в двоичной и десятичной системах счисления:

- а) $11101101_2 + 1010111_2 - X_2$;
- б) $A5_{16} + 2B_{16} - X_{10}$

Вариант 6

1 задание. Выберите правильные ответы

1. Определите признак, по которому информация делится на визуальную, аудиальную, тактильную и т.д.:
 - а. По способу передачи и восприятия;
 - б. По общественному назначению;
 - в. По области возникновения;
 - г. По способу отображения.
2. Перевод чисел из шестнадцатеричной системы счисления в двоичную и наоборот осуществляется
 - а. по тетрадам;
 - б. по триадам;
 - в. слева направо;
 - г. справа налево.
3. Элемент электронных схем, который может находиться в любом из двух устойчивых состояний, а также многократно переходит из одного состояния в другое - это
 - а. Триггер;
 - б. Вентиль;
 - в. Регистр;
 - г. Счетчик.

4. Принцип адресности логического устройства ЭВМ фон Неймана (возможно несколько вариантов):

- а. Программы и данные хранятся в одной и той же памяти.
- б. обеспечивает автоматизацию процессов вычислений на ЭВМ
- в. процессору в произвольный момент времени доступна любая ячейка.
- г. Структурно основная память состоит из пронумерованных ячеек

5. Память характеризуется:

- а. объемом и стоимостью;
- б. количеством информации и скоростью;
- в. емкостью, удельной емкостью, быстродействием;
- г. объемом, скоростью, стоимостью.

6. Метод организации оперативной памяти, при которой информация размещается в определенной колонке:

- а. метод строк/колонок;
- б. метод статических колонок;
- в. метод страничной организации;
- г. страничный метод.

7. ОЗУ относится к:

- а. внешней памяти;
- б. внутренней памяти;
- в. постоянной памяти;
- г. общей памяти.

8. В операционной системе последовательному порту присваивается логическое имя:

- а. COM;
- б. LPT;
- в. USB;
- г. IDE.

9. Дигитайзер - это устройство, главное назначение которого –

- а. печать
- б. расшифровка
- в. оцифровка изображений
- г. обеспечение связи между принтером и компьютером

10. Разрешающая способность сканера

- а. от 75 до 1600 dpi
- б. от 0 до 75 dpi
- в. от 1600 до 3200 dpi
- г. от 100 до 1650 dpi

11. Язык программирования Ассемблер: к логическим командам относятся (возможно несколько вариантов):

- а. команды сложения;
- б. команды логических операций;
- в. команды сравнения;
- г. команды сдвигов.

12. Архитектура любого компьютера – абстрактная структура, состоящая из четырех компонентов: процессор команд, процессор данных, иерархия памяти и переключатель.

- а. Классификация Базу;
- б. Классификация Джонсона;
- в. Классификация Кришнамарфи;
- г. Классификация Скилликорна.

13. Процессоры, в основе работы которых лежит принцип обработки многих данных с помощью одной команды

- а.нейронные;
- б. потоковые;
- в.баз данных;
- г. коммуникационные.

2 задание. Построить таблицу состояния по следующей функции:

$$F = \overline{X \& Y} \vee \overline{X \vee Y} \& (X \vee Y)$$

Вариант 7

1 задание. Выберите правильные ответы

1. Величина, характеризующая процесс, не имеющий перерывов или промежутков:

- а. дискретная форма представления информации;
- б. непрерывная форма представления информации;
- в. полезность информации;
- г. полнота информации.

2. Перевод чисел из восьмеричной системы счисления в двоичную и наоборот осуществляется

- а. по тетрадам;
- б. по триадам;
- в. слева направо;
- г. справа налево.

3. Цифровая схема, выполняющая функции счета, построенная на базе триггеров, - это

- а. Шифратор;
- б. Сумматор;
- в. Регистр;
- г. Счетчик.

4. Принцип однородности памяти логического устройства ЭВМ фон Неймана:

- а. Программы и данные хранятся в одной и той же памяти.
- б. обеспечивает автоматизацию процессов вычислений на ЭВМ
- в. процессору в произвольный момент времени доступна любая ячейка.
- г. Структурно основная память состоит из пронумерованных ячеек

5. Метод организации оперативной памяти, при которой память представляет собой матрицу, состоящую из строк и колонок:

- а. метод строк/колонок;
- б. метод статических колонок;
- в. метод страничной организации;
- г. метод динамических колонок.

6. Вид памяти, в которую можно вносить изменения:
- а. ОЗУ;
 - б. ПЗУ;
 - в. ППЗУ;
 - г. БЗУ.
7. Максимальное количество данных, которое хранится в памяти, называется:
- а. емкостью;
 - б. удельной емкостью;
 - в. быстродействием;
 - г. оперативностью.
8. В операционной системе параллельному порту присваивается логическое имя:
- а. COM;
 - б. LPT;
 - в. USB;
 - г. IDE.
9. Роликовые сканеры...
- а. Оригинал автоматически перемещается относительно сканирующей головки, часто имеется автоматическая подача документов
 - б. Внешне напоминают фотоувеличитель: внизу лежит сканируемый документ, а наверху находится сканирующая головка
 - в. Бумажный лист с изображением или текстом кладется на прозрачную стеклянную поверхность, под которой проходит распознающий элемент сканера, и закрывается крышкой
 - г. Прокатывают по поверхности документа рукой
10. Плазменные мониторы (возможно несколько вариантов)
- а. Основной элемент дисплея — электронно-лучевая трубка.
 - б. Заряды передаются через пассивную матрицу — сетку невидимых нитей, горизонтальных и вертикальных, создавая в месте пересечения нитей точку изображения
 - в. Его передняя, обращенная к зрителю часть с внутренней стороны покрыта люминофором — специальным веществом, способным излучать свет при попадании на него быстрых электронов
 - г. Создаются путем заполнения пространства между двумя стеклянными поверхностями инертным газом, например аргоном или неоном.
11. Язык программирования Ассемблер: Команды логических операций(возможно несколько вариантов):
- а. AND;
 - б. DEC;
 - в. OR;
 - г. XOR.
12. Любую параллельную вычислительную систему можно однозначно описать последовательностью решений, принятых на этапе ее проектирования, а сам процесс проектирования представить в виде дерева.
- а. Классификация Базу;
 - б. Классификация Джонсона;
 - в. Классификация Кришнамарфи;

г. Классификация Скилликорна.

13. Это микрочипы, являющие собой нечто среднее между жесткими специализированными интегральными микросхемами и гибкими процессорами общего назначения.

- а. нейронные;
- б. потоковые;
- в. баз данных;
- г. коммуникационные.

2 задание. Перевести числа из одной системы счисления в другую:

- а) $111011101_2 - X_{16}$;
- б) $75_{10} - X_8$
- в) $125,34_8 - X_{10}$

Вариант 8

1 задание. Выберите правильные ответы

1. Последовательность символов, характеризующая прерывистую, изменяющуюся величину:

- а. дискретная форма представления информации;
- б. непрерывная форма представления информации;
- в. полезность информации;
- г. полнота информации.

2. Система счисления, в которой цифра меняют своего количественного значения при изменении их расположения в числе:

- а. унарная;
- б. позиционная;
- в. непозиционная;
- г. количественная.

3. Узел ЭВМ, который состоит из системы запоминающих элементов и управляющей этой системой логической схемы, - это

- а. Шифратор;
- б. Сумматор;
- в. Регистр;
- г. Счетчик.

4. Принцип программного управления логического устройства ЭВМ фон Неймана:

- а. программы и данные хранятся в одной и той же памяти;
- б. обеспечивает автоматизацию процессов вычислений на ЭВМ;
- в. процессору в произвольный момент времени доступна любая ячейка;
- г. структурно основная память состоит из пронумерованных ячеек.

5. Продолжительность обращения к памяти называется:

- а. емкостью;
- б. удельной емкостью;
- в. быстродействием;
- г. оперативностью.

6. ПЗУ относится к:

- а. внешней памяти;
- б. внутренней памяти;
- в. постоянной памяти;
- г. общей памяти.

7. Вид памяти, которая представляет собой цепочку регистров и устроена по принципу «последний пришел – первый ушел»:

- а. ПЗУ;
- б. стек;
- в. ППЗУ;
- г. БЗУ.

8. Универсальный помехоустойчивый интерфейс, который предназначен для работы с периферийными устройствами любого типа

- а. VLB;
- б. IDE;
- в. PCI;
- г. ESDI.

9. Принтеры по способу формирования символов делятся на (возможно несколько вариантов):

- а. знакопечатающие,
- б. знакосинтезирующие,
- в. матричные,
- г. термические.

10. Видеоадаптер - ...

- а. устройство для вывода информации из компьютера
- б. устройства для оцифровки и ввода в компьютер изображений с бумажных копий
- в. электронное устройство, преобразующее графический образ, хранящийся, как содержимое памяти компьютера (или самого адаптера), в форму, пригодную для дальнейшего вывода на экран монитора
- г. устройство, которое чертит графики, рисунки или диаграммы под управлением компьютера.

11. Язык программирования Ассемблер: Команды логических операций (возможно несколько вариантов):

- а. логическое И;
- б. логическое Не;
- в. логическое Или;
- г. исключающее Или.

12. Совокупность взаимосвязанных и взаимодействующих процессов или ЭВМ, периферийного оборудования и программного обеспечения, предназначенная для сбора, хранения, обработки и распределения информации –

- а. поток данных;
- б. поток команд;
- в. кластер;
- г. вычислительная система.

13. ...системы относятся к классу SIMD и включают некоторое множество операционных устройств, способных одновременно по командам управляющего устройства вести обработку нескольких потоков данных.

- а. нейронные;
- б. потоковые;
- в. баз данных;
- г. ассоциативные.

2 задание.

1. Запишите указанные команды в восьмеричном и двоичном виде.
2. Заполните таблицу, определив, какие значения будут содержаться в ячейках памяти и в регистрах.
3. Объясните, что произойдет после выполнения каждой из команд.

Исходные данные:
INCR1

До выполнения команды		После выполнения команды	
Адреса памяти	Значение данных	Адреса памяти	Значение данных
10004	<код команды>	10004	
Регистры	Значение данных	Регистры	Значение данных
R1	124551	R1	

Эталон ответов к контрольным заданиям

Задание 1.

	1	2	3	4	5	6	7	8
1	б	г	а	в	б	а	б	а
2	б	б	б	а	а	а	б	б
3	б	в	г	а	б	а	г	в
4	г	а, б	в, г	в, г	а, б	в, г	а	б
5	г	в	а	а	в	в	а	в
6	в	а	в	а	в	б	а	б
7	б	в	в	б	а	б	а	б
8	б	б	а	в	б	а	б	г
9	в	а, б, в	б	б	а, б, в, г	в	а	а, б
10	г	г	б	б	г	а	г	в
11	б	б	а, б, в	а	а	б, в, г	а, в, г	а, в, г
12	б	б	в	а	в	г	а	г
13	б	в	а	г	в	б	г	г

Задание 2.

Вариант 1

- 1) $124,35_{10} - X_8$
 $124:8=15(0),$
 $15:8=1(7)$

$$\begin{aligned}
 0,35 \cdot 8 &= 2,8 \\
 0,8 \cdot 8 &= 1,6 \\
 0,6 \cdot 8 &= 4,8 \\
 0,8 \cdot 8 &= 6,4 \\
 0,4 \cdot 8 &= 3,2
 \end{aligned}$$

$$124,35_{10} = 170, 21463_8$$

$$2) 1010110101_2 - X_{16}$$

$$1010110101_2 = 1*2^9 + 0*2^8 + 1*2^7 + 0*2^6 + 1*2^5 + 1*2^4 + 0*2^3 + 1*2^2 + 0*2^1 + 1*2^0 = 512 + 128 + 32 + 16 + 4 + 1 = 693_{10}$$

$$693_{10} : 16 = 43(5)$$

$$43 : 16 = 2(11)$$

$$1010110101_2 - 2B5_{16}$$

$$3) 46,2_8 - X_{10}$$

$$46,2_8 = 4*8^1 + 6*8^0 + 2*8^{-1} = 32 + 6 + 0,25 = 38,25_{10}$$

Вариант 2

$$a) 100101011_2 + 10011100_2 - X_2;$$

$$+ 100101011_2$$

$$\underline{10011100_2}$$

$$111000111_2$$

$$b) 37_8 + 25_8 - X_{10}$$

$$37_8 = 3*8^1 + 7*8^0 = 31_{10}$$

$$25_8 = 2*8^1 + 5*8^0 = 21_{10}$$

$$31_{10} + 21_{10} = 52_{10}$$

Вариант 3

Построить таблицу состояния по следующей функции:

$$F = \overline{X} \& \overline{YV(XVY)} \& X;$$

X	Y	\overline{X}	XVY	\overline{XVY}	$\overline{(XVY)} \& X$	$\overline{X} \& Y$	F
0	0	1	0	1	0	0	0
0	1	1	1	0	0	1	1
1	0	0	1	0	0	0	0
1	1	0	1	0	0	0	0

Вариант 4

$$a) 97_{10} - X_{16}$$

$$97 : 16 = 6(1)$$

$$97_{10} = 61_{16}$$

$$b) 6A,1_{16} - X_{10}$$

$$6A,1_{16} = 6*16^1 + A*16^0 + 1*16^{-1} = 96 + 10 + 0,0625 = 106,0625_{10}$$

$$c) 134,2_8 - X_2$$

$$134,2_8 = 1*8^2 + 3*8^1 + 4*8^0 + 2*8^{-1} = 64 + 24 + 4 + 0,25 = 92,25_{10}$$

Вариант 5

$$a) 11101101_2 + 1010111_2 - X_2$$

$$11101101_2$$

$$+ 1010111_2$$

$$101000100_2$$

$$\begin{aligned}
 & \text{b) } A5_{16} + 2B_{16} - X_{10} \\
 & A5_{16} = A \cdot 16^1 + 5 \cdot 16^0 = 165_{10} \\
 & 2B_{16} = 2 \cdot 16^1 + B \cdot 16^0 = 43_{10} \\
 & 165_{10} + 43_{10} = 208_{10}
 \end{aligned}$$

Вариант 6

Построить таблицу состояния по следующей функции:

$$F = X \oplus Y \vee X \vee Y \oplus (X \vee Y);$$

X	Y	$X \oplus Y$	$X \vee Y$	$Y \oplus (X \vee Y)$	$X \oplus Y \vee X$	F
0	0	0	0	0	0	0
0	1	0	1	1	0	1
1	0	0	1	0	1	1
1	1	1	1	1	1	1

Вариант 7

- a) $111011101_2 - X_{16}$
 $111011101_2 = 1 \cdot 2^8 + 1 \cdot 2^7 + 1 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 477_{10}$
 $477 : 16 = 29(13)$
 $29 : 16 = 1(13)$
 $111011101_2 = 1DD_{16}$
- b) $75_{10} - X_8$
 $75 : 8 = 9(3)$
 $9 : 8 = 1(1)$
 $75_{10} = 113_8$
- c) $125,34_8 - X_{10}$
 $125,34_8 = 1 \cdot 8^2 + 2 \cdot 8^1 + 5 \cdot 8^0 + 3 \cdot 8^{-1} + 4 \cdot 8^{-2} = 85,4375_{10}$

Вариант 8

В восьмеричном коде команда INCR1 имеет вид 005201 (0052 – код команды, 0 – код режима прямой адресации, 1 – номер регистра), в двоичном – 000 000 101 010 000 001.
 0 0 5 2 0 1

До выполнения команды		После выполнения команды	
Адреса памяти	Значение данных	Адреса памяти	Значение данных
10004	005201	10004	005201
Регистры	Значение данных	Регистры	Значение данных
R1	124551	R1	124552

В результате выполнения этой команды содержимое регистра R1 увеличится на 1 (124551+1=124552).

Вариант 1	Вариант 2
1. Кто является основоположником математической логики: а) Аристотель	1. Слово "логика" обозначает... а) форма мышления, в которой отражаются признаки предмета

б) Декарт Рене в) Лейбниц Г.В. г) Джордж Буль	б) совокупность правил, которым подчиняется процесс мышления в) мысль, к которой что-то утверждается или отрицается о предметах г) прием мышления, когда из исходного знания получается новое знание
2. Логическая операция, соответствующая союзу "И" – это... а) импликация б) эквиваленция в) дизъюнкция г) конъюнкция	2. Логическая операция, соответствующая союзу "ИЛИ" – это а) импликация б) эквиваленция в) дизъюнкция г) конъюнкция
3. Логическая операция, соответствующая союзу "ЕСЛИ..., ТО..." – это... а) импликация б) эквиваленция в) дизъюнкция г) конъюнкция	3. Логическая операция, соответствующая союзу "ТОГДА И ТОЛЬКО ТОГДА, КОГДА" – это... а) импликация б) эквиваленция в) дизъюнкция г) конъюнкция
4. Высказывание $A \rightarrow B$ ложно тогда и только тогда, когда ... а) А истинно, а В ложно б) А и В совпадают в) А ложно, а В истинно г) А и В истинны	4. Высказывание $A \leftrightarrow B$ истинно, тогда и только тогда, когда Выберите один из 4 вариантов ответа: а) А истинно, а В ложно б) А и В совпадают в) А ложно, а В истинно г) А и В истинны
5. Что не относится к периферийным устройствам? а) манипуляторы; б) жесткий диск; в) модем; г) ОЗУ.	5. К позиционным системам счисления не относится а) римская СС; б) десятичная СС; в) двоичная СС; г) шестнадцатеричная СС.
6. Что такое микропроцессор? А) интегральная микросхема, которая выполняет поступающие на ее вход команды (например, вычисление) и управление работой машины; В) устройство для хранения той информации, которая часто используется в работе; С) устройство для вывода текстовой или графической информации; Д) устройство для ввода алфавитно-цифровых данных.	6. Назначение процессора: А) управлять работой ПК с помощью электрических импульсов; В) подключать периферийные устройства к магистрали; С) выполнять команды одной программы в данный момент; Д) выполнять арифметико-логические операции и управлять ходом вычислительного процесса.

<p>7. Найдите соответствие: Hardware - это:</p> <p>А) самая популярная система для компьютеров IBM PC;</p> <p>В) аппаратная часть компьютера;</p> <p>С) система, обеспечивающая создание новых программ;</p> <p>Д) модернизация аппаратной или программной части компьютеров</p>	<p>7. Найдите соответствие: Software – это:</p> <p>А) программа вспомогательного назначения;</p> <p>В) система «включил и работай»</p> <p>С) программное обеспечение компьютера;</p> <p>Д) программы для подключения к компьютеру новых устройств</p>
<p>8. Внешняя память необходима для:</p> <p>А) для хранения часто изменяющейся информации в процессе решения задачи;</p> <p>В) для долговременного хранения информации после выключения компьютера;</p> <p>С) для обработки текущей информации;</p> <p>Д) для постоянного хранения информации о работе компьютера.</p>	<p>8. ОЗУ – это память, в которой:</p> <p>А) хранится исполняемая в данный момент времени программа и данные, с которыми она непосредственно работает;</p> <p>В) хранится информация, присутствие которой постоянно необходимо для работы компьютера;</p> <p>С) хранится информация, независимо от того работает компьютер или нет;</p> <p>Д) хранятся программы, предназначенные для обеспечения диалога пользователя с компьютером</p>
<p>9. Что такое КЭШ-память?</p> <p>А) память, в которой обрабатывается программа в данный момент времени;</p> <p>В) память, в которой хранится информация, после выключения ПК;</p> <p>С) сверхоперативная память для хранения часто используемых данных ОЗУ;</p> <p>Д) память, в которой хранятся системные файлы операционной системы.</p>	<p>9. Что такое адресное пространство?</p> <p>А) Максимальное количество разрядов двоичного кода для символа;</p> <p>В) периодичность импульсов, синхронизирующих работу устройств компьютера;</p> <p>С) множество адресов ячеек памяти, к которым обращается процессор;</p> <p>Д) сигнал, определяющий характер обмена информацией.</p>
<p>10. Укажите, в каких случаях высказывание истинно, а в каких ложно:</p> $(\overline{A \Rightarrow B}) \Leftrightarrow (\overline{B} \wedge \overline{A})$	<p>10. Укажите, в каких случаях высказывание истинно, а в каких ложно:</p> $\left(\overline{(A \wedge B) \Rightarrow A} \right) \Leftrightarrow (A \downarrow B)$

Эталоны ответов к полугодовой срезовой контрольной работе

по дисциплине

ОП.02 Архитектура компьютерных систем

№ вопроса	1	2	3	4	5	6	7	8	9
Вариант 1	г	г	а	а	г	а	в	в	д
Вариант 2	б	в	б	г	а	с	с	а	с

Вариант 1

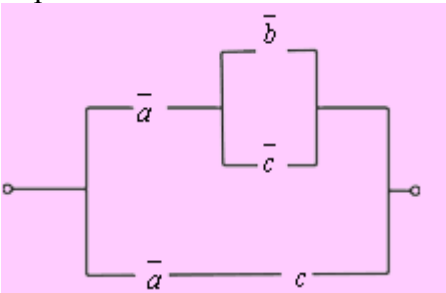
10. Истинно при A=0, B=1 и при A=1 и B=1, Ложно при A=0, B=0 и при A=1, B=0.

Вариант 2.

10. Истинно при A=0, B=1, Ложно при A=0, B=0 и при A=1, B=0, при A=1 и B=1.

Годовая срезовая контрольная работа

Вариант 1	Вариант 2
<p>1. Виды памяти более высокого уровня имеют...</p> <p>а) меньший объем, меньшую скорость доступа;</p> <p>б) меньший объем, большую скорость доступа;</p> <p>в) больший объем, меньшую скорость доступа;</p> <p>г) больший объем, большую скорость доступа</p>	<p>1. К энергонезависимой памяти относится</p> <p>а) ПЗУ;</p> <p>б) ОЗУ;</p> <p>в) жесткий диск;</p> <p>г) flash.</p>
<p>2. Минимальная единица информации в двухуровневой иерархии - это...</p> <p>а) байт;</p> <p>б) ячейка;</p> <p>в) кластер;</p> <p>г) блок.</p>	<p>2. Системная шина не содержит...</p> <p>а) шину адреса;</p> <p>б) шину данных;</p> <p>в) АЛУ;</p> <p>г) шину управления.</p>
<p>3. Использование кэш-памяти в качестве буфера между процессором и памятью – это...</p> <p>а) чередование памяти;</p> <p>б) разбиение памяти на страницы;</p> <p>в) кэширование памяти;</p> <p>г) фрагментация памяти.</p>	<p>3. Скорость работы компьютера зависит от...</p> <p>а) объема жесткого диска;</p> <p>б) наличия периферийных устройств;</p> <p>в) скорости нажатия на клавиши клавиатуры;</p> <p>г) объема оперативной памяти.</p>
<p>4. Узел, предназначенный для приема, временного хранения и выдачи машинного слова - это...</p> <p>а) регистр;</p> <p>б) триггер;</p> <p>в) счетчик;</p> <p>г) сумматор.</p>	<p>4. Запоминающим элементом динамической памяти является</p> <p>а) конденсатор;</p> <p>б) триггер;</p> <p>в) регистр;</p> <p>г) полусумматор.</p>
<p>5. Последовательные и асинхронные сумматоры – это...</p> <p>а) комбинационные сумматоры;</p> <p>б) накапливающие сумматоры;</p> <p>в) дублирующие сумматоры;</p> <p>г) идеальные сумматоры, все разряды которых срабатывают одновременно.</p>	<p>5. Регенерация памяти – это...</p> <p>а) очистка памяти;</p> <p>б) увеличение памяти;</p> <p>в) уменьшение памяти;</p> <p>г) восстановление памяти.</p>
<p>6. Количество выходов мультиплексора называется...</p> <p>а) каналами;</p> <p>б) разрядами;</p> <p>в) количеством каналов мультиплексора;</p> <p>г) количеством разрядов мультиплексора.</p>	<p>6. Сколько существует уровней кэш-памяти?</p> <p>а) только 1;</p> <p>б) только 2;</p> <p>в) только 3;</p> <p>г) 3 и более.</p>
<p>7. Что не относится к периферийным устройствам?</p> <p>а) манипуляторы;</p> <p>б) жесткий диск;</p> <p>в) модем;</p>	<p>7. К позиционным системам счисления не относится</p> <p>а) римская СС;</p> <p>б) десятичная СС;</p> <p>в) двоичная СС;</p>

г) ОЗУ.	г) шестнадцатеричная СС.
8. Специализированная интегральная схема, работающая в содружестве с ЦП, но менее универсальная – это... а) сопроцессор; б) CPU; в) ОЗУ; г) АЛУ.	8. Идентификатор, присвоенный различным объектам, которые должны быть распознаны процессором – это... а) уровень кэш; б) уровень привилегий; в) уровень буферизации; г) уровень контроля.
9. Закончите предложение Унитарным называется двоичный код, содержащий _____.	9. Закончите предложение Число триггеров в регистре определяет _____.
10. По данной формуле составить РКС $a \vee b \wedge c \rightarrow \bar{a} \wedge \bar{b}$	10. Упростить РКС: 

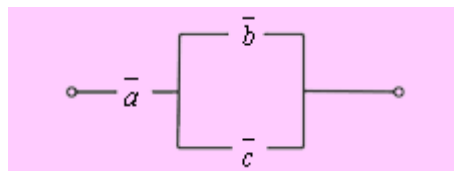
Эталоны ответов к годовой срезовой контрольной работе

№ вопроса	1	2	3	4	5	6	7	8	9
Вариант 1	б	г	в	а	б	г	г	а	Одну и только одну единицу
Вариант 2	б	в	г	а	г	г	а	б	Разрядность регистра.

Вариант 1

10 Решение. Упростим данную формулу с помощью равносильных преобразований:

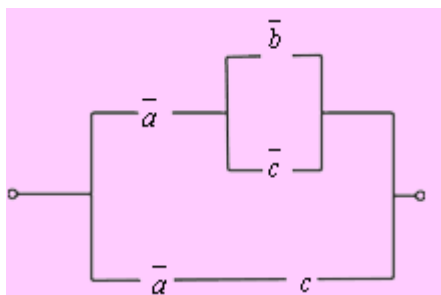
$$\begin{aligned}
 a \vee b \wedge c &\rightarrow \bar{a} \wedge \bar{b} \equiv \overline{a \vee b \wedge c} \vee \bar{a} \wedge \bar{b} \equiv \bar{a} \wedge (\bar{b} \vee \bar{c}) \vee \bar{a} \wedge \bar{b} \equiv \bar{a} \wedge \bar{b} \vee \bar{a} \wedge \bar{b} \vee \bar{a} \wedge \bar{c} \vee \bar{a} \wedge b \equiv \\
 &\equiv \bar{a} \wedge \bar{b} \vee \bar{a} \wedge \bar{c} \equiv \bar{a} \wedge (\bar{b} \vee \bar{c})
 \end{aligned}$$



Тогда РКС для данной формулы имеет вид:

Вариант 2

10. Упростить РКС:

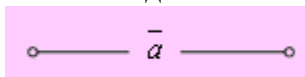


Решение. Составим по данной РКС формулу (функцию проводимости) и упростим ее:

$$\bar{a} \wedge (\bar{b} \vee c) \vee \bar{a} \bar{c} \equiv \bar{a} \bar{b} \vee \bar{a} c \vee \bar{a} \bar{c} \equiv \bar{a} \bar{b} \vee \bar{a} \wedge (\bar{c} \vee c) \equiv \bar{a} \bar{b} \vee \bar{a} \equiv \bar{a} \wedge (\bar{b} \vee 1) \equiv \bar{a}$$

(к последним двум слагаемым применили закон поглощения).

Тогда упрощенная схема выглядит так:



2) Практические занятия:

Практическое занятие № 1 «Перевод чисел из десятичной системы счисления в двоичную систему счисления».

Цель занятия: формировать умения по переводу целых и дробных чисел из десятичной системы счисления в различные системы счисления.

Содержание работы:

Перевести данное число из десятичной системы счисления в двоичную систему счисления.

Вариант 1

1. а) 666; б) 305; в) 153,25; г) 162,25; д) 248,46.

Вариант 2

1. а) 164; б) 255; в) 712,25; г) 670,25; д) 11,89.

Вариант 3

1. а) 273; б) 661; в) 156,25; г) 797,5; д) 53,74.

Вариант 4

1. а) 105; б) 358; в) 377,5; г) 247,25; д) 87,27.

Вариант 5

1. а) 500; б) 675; в) 810,25; г) 1017,25; д) 123,72.

Вариант 6

1. а) 218; б) 808; в) 176,25; г) 284,25; д) 253,04.

Вариант 7

1. а) 306; б) 467; в) 218,5; г) 667,25; д) 318,87.

Вариант 8

1. а) 167; б) 113; в) 607,5; г) 828,25; д) 314,71.

Вариант 9

1. а) 342; б) 374; в) 164,25; г) 520,375; д) 97,14.

Вариант 10

1. а) 524; б) 222; в) 579,5; г) 847,625; д) 53,35.

Вариант 11

1. а) 113; б) 875; в) 535,1875; г) 649,25; д) 6,52.

Вариант 12

1. а) 294; б) 723; в) 950,25; г) 976,625; д) 282,73.

Вариант 13

1. а) 617; б) 597; в) 412,25; г) 545,25; д) 84,82.

Вариант 14

1. а) 1047; б) 335; в) 814,5; г) 518,625; д) 198,91.

Вариант 15

1. а) 887; б) 233; в) 801,5; г) 936,3125; д) 218,73.

Вариант 16

1. а) 969; б) 549; в) 973,375; г) 508,5; д) 281,09.

Вариант 17

1. а) 163; б) 566; в) 694,375; г) 352,375; д) 288,61.

Вариант 18

1. а) 917; б) 477; в) 74,5; г) 792,25; д) 84,33.

Вариант 19

1. а) 477; б) 182; в) 863,25; г) 882,25; д) 75,2.

Вариант 20

1. а) 804; б) 157; в) 207,625; г) 435,375; д) 30,43.

Вариант 21

1. а) 753; б) 404; в) 111,1875; г) 907,0625; д) 62,88.

Вариант 22

1. а) 571; б) 556; в) 696,25; г) 580,375; д) 106,67.

Вариант 23

1. а) 244; б) 581; в) 351,6875; г) 1027,375; д) 151,44.

Вариант 24

1. а) 388; б) 280; в) 833,5625; г) 674,25; д) 159,05.

Вариант 25

1. а) 386; б) 608; в) 398,6875; г) 270,25; д) 317,32.

Практическое занятие № 2 «Перевод чисел из двоичной системы счисления в десятичную систему счисления».

Цель занятия: формировать умения по переводу целых и дробных чисел из двоичной системы счисления в различные системы счисления.

Содержание работы:

Перевести данное число из двоичной системы счисления в десятичную систему счисления.

Вариант 1

1. а) 110011101₂; б) 10000000111₂; в) 10110101,1₂; г) 100000110,10101₂;

Вариант 2

1. а) 100111001₂; б) 1001000₂; в) 1111100111,01₂; г) 1010001100,101101₂;

Вариант 3

1. а) 1100000000₂; б) 1101011111₂; в) 1011001101,00011₂; г) 1011110100,011₂;

Вариант 4

1. а) 1100001001₂; б) 1100100101₂; в) 1111110110,01₂; г) 11001100,011₂;

Вариант 5

1. а) 1101010001₂; б) 100011100₂; в) 1101110001,011011₂; г) 110011000,111001₂;

Вариант 6

1. а) 111000100₂; б) 1011001101₂; в) 10110011,01₂; г) 1010111111,011₂;

Вариант 7

1. а) 1111000111₂; б) 11010101₂; в) 1001111010,010001₂; г) 1000001111,01₂;

Вариант 8

1. а) 110010001₂; б) 100100000₂; в) 1110011100,111₂; г) 1010111010,1110111₂;

Вариант 9

1. а) 1000110110₂; б) 111100001₂; в) 1110010100,1011001₂;

Вариант 10

1. а) 101111111₂; б) 1111100110₂; в) 10011000,1101011₂; г) 1110001101,1001₂;

Вариант 11

1. а) 11101000_2 ; б) 1010001111_2 ; в) $1101101000,01_2$; г) $1000000101,01011_2$;

Вариант 12

1. а) 10000011001_2 ; б) 10101100_2 ; в) $1101100,01_2$; г) $1110001100,1_2$;

Вариант 13

1. а) 110111101_2 ; б) 1110011101_2 ; в) $111001000,01_2$; г) $1100111001,1001_2$;

Вариант 14

1. а) 1101100000_2 ; б) 100001010_2 ; в) $1011010101,1_2$; г) $1010011111,1101_2$;

Вариант 15

1. а) 1010100001_2 ; б) 10000010101_2 ; в) $1011110000,100101_2$; г) $1000110001,1011_2$;

Вариант 16

1. а) 10100010_2 ; б) 1110010111_2 ; в) $110010010,101_2$; г) $1111011100,10011_2$;

Вариант 17

1. а) 1001101001_2 ; б) 110011101_2 ; в) $1000001101,01_2$; г) $1010001001,11011_2$;

Вариант 18

1. а) 1110011100_2 ; б) 1111101111_2 ; в) $111110100,101_2$; г) $110011110,1000011_2$;

Вариант 19

1. а) 101011100_2 ; б) 1000010011_2 ; в) $11100011,1_2$; г) $100101010,00011_2$;

Вариант 20

1. а) 10010000_2 ; б) 11001010_2 ; в) $1110101100,1011_2$; г) $110110101,10111_2$;

Вариант 21

1. а) 11100011_2 ; б) 1111001111_2 ; в) $1011111111,01001_2$; г) $1001011101,011_2$;

Вариант 22

1. а) 110011010_2 ; б) 111001010_2 ; в) $1000010011,00101_2$; г) $11010110,00001_2$;

Вариант 23

1. а) 1001100111_2 ; б) 1100010010_2 ; в) $1100110010,1101_2$; г) $1001011,0101_2$;

Вариант 24

1. а) 11001111_2 ; б) 101001101_2 ; в) $101001101,001001_2$; г) $100101011,101_2$;

Вариант 25

1. а) 11000001_2 ; б) 1111111110_2 ; в) $1110100010,10101_2$; г) $1001011001,011_2$;

Практическое занятие № 3 «Перевод чисел из десятичной системы счисления в восьмеричную и шестнадцатеричную системы счисления».

Цель занятия: формировать умения по переводу целых и дробных чисел из десятичной системы счисления в различные системы счисления.

Содержание работы:

1. Перевести данное число из десятичной системы счисления восьмеричную и шестнадцатеричную системы счисления.

Вариант 1

2. а) 666; б) 305; в) 153,25; г) 162,25; д) 248,46.

Вариант 2

2. а) 164; б) 255; в) 712,25; г) 670,25; д) 11,89.

Вариант 3

2. а) 273; б) 661; в) 156,25; г) 797,5; д) 53,74.

Вариант 4

2. а) 105; б) 358; в) 377,5; г) 247,25; д) 87,27.

Вариант 5

2. а) 500; б) 675; в) 810,25; г) 1017,25; д) 123,72.

Вариант 6

2. а) 218; б) 808; в) 176,25; г) 284,25; д) 253,04.

Вариант 7

2. а) 306; б) 467; в) 218,5; г) 667,25; д) 318,87.

Вариант 8

2. а) 167; б) 113; в) 607,5; г) 828,25; д) 314,71.

Вариант 9

2. а) 342; б) 374; в) 164,25; г) 520,375; д) 97,14.

Вариант 10

2. а) 524; б) 222; в) 579,5; г) 847,625; д) 53,35.

Вариант 11

2. а) 113; б) 875; в) 535,1875; г) 649,25; д) 6,52.

Вариант 12

2. а) 294; б) 723; в) 950,25; г) 976,625; д) 282,73.

Вариант 13

2. а) 617; б) 597; в) 412,25; г) 545,25; д) 84,82.

Вариант 14

2. а) 1047; б) 335; в) 814,5; г) 518,625; д) 198,91.

Вариант 15

2. а) 887; б) 233; в) 801,5; г) 936,3125; д) 218,73.

Вариант 16

2. а) 969; б) 549; в) 973,375; г) 508,5; д) 281,09.

Вариант 17

2. а) 163; б) 566; в) 694,375; г) 352,375; д) 288,61.

Вариант 18

2. а) 917; б) 477; в) 74,5; г) 792,25; д) 84,33.

Вариант 19

2. а) 477; б) 182; в) 863,25; г) 882,25; д) 75,2.

Вариант 20

2. а) 804; б) 157; в) 207,625; г) 435,375; д) 30,43.

Вариант 21

2. а) 753; б) 404; в) 111,1875; г) 907,0625; д) 62,88.

Вариант 22

2. а) 571; б) 556; в) 696,25; г) 580,375; д) 106,67.

Вариант 23

2. а) 244; б) 581; в) 351,6875; г) 1027,375; д) 151,44.

Вариант 24

2. а) 388; б) 280; в) 833,5625; г) 674,25; д) 159,05.

Вариант 25

2. а) 386; б) 608; в) 398,6875; г) 270,25; д) 317,32.

Практическое занятие № 4 «Перевод чисел из восьмеричной и шестнадцатеричной систем счисления в десятичную систему счисления».

Цель занятия: формировать умения по переводу целых и дробных чисел из восьмеричной и шестнадцатеричной систем счисления в десятичную систему счисления.

Содержание работы:

Вариант 1

671,24₈; 41A,6₁₆.

Вариант 2

413,41₈; 118,8C₁₆.

Вариант 3

1017,2₈; 111,B₁₆.

Вариант 4

112,04₈; 334,A₁₆.

Вариант 5

1347,17₈ (8); 155,6C₁₆ (16).

Вариант 6

1665,3₈; FA,7₁₆.

Вариант 7

465,3₈; 252,38₁₆.

Вариант 8

704,6₈; 367,38₁₆.

Вариант 9

666,16₈; 1C7,68₁₆.

Вариант 10

140,22₈; 1DE,54₁₆.

Вариант 11

1600,14; 1E9,4₁₆.

Вариант 12

1053,2₈; 200,6₁₆.

Вариант 13

1471,17₈; 3EC,5₁₆.

Вариант 14

452,63₈; 1E7,08₁₆.

Вариант 15

1034,34₈; 72,6₁₆.

Вариант 16

605,02₈; 3C8,8₁₆.

Вариант 17

247,1₈; 81,4₁₆.

Вариант 18

1446,62₈; 9C,D₁₆.

Вариант 19

1762,7₈; 1B5,6₁₆.

Вариант 20

1164,36₈; 1D5,C8₁₆.

Вариант 21

615,72₈; 3DA,5₁₆.

Вариант 22

1343,66₈; 3C3,6₁₆.

Вариант 23

171,3₈; 3A3,4₁₆.

Вариант 24

750,51₈; 90,8₁₆.

Вариант 25

1335,2₈; 18F,8₁₆.

Практическое занятие № 5 «Выполнение арифметических операций над двоичными числами».

Цель занятия: формировать умения по выполнению арифметических операций в различных системах счисления.

Содержание работы:

Вариант 1

Произвести действия над числами в различных системах счисления:

1. $110,011_2 + 11,1_2$
2. $110,011_2 - 11,1_2$
3. $11,01_2 * 1,1_2$
4. $6A,4_{16} + 82,B_{16}$
5. $82,B_{16} - 6A,4_{16}$
6. $3,1_{16} * B,6_{16}$

Выполнить проверку полученных результатов путем их перевода в десятичную систему счисления

Вариант 2

Произвести действия над числами в различных системах счисления:

1. $101,01_2 + 10,10_2$
2. $101,01_2 - 10,10_2$
3. $11,1_2 * 1,01_2$
4. $A6, F_{16} + 1B, 6_{16}$
5. $A6, F_{16} - 1B, 6_{16}$
6. $6, 3_{16} * A, 1_{16}$

Выполнить проверку полученных результатов путем их перевода в десятичную систему счисления

Вариант 3

Произвести действия над числами в различных системах счисления:

1. $100,101_2 + 10,11_2$
2. $100,101_2 - 10,11_2$
3. $10,11_2 * 1,1_2$
4. $B3, D_{16} + 6A, E_{16}$
5. $B3, D_{16} - 6A, E_{16}$
6. $2, A_{16} * 3, 9_{16}$

Выполнить проверку полученных результатов путем их перевода в десятичную систему счисления

Вариант 4

Произвести действия над числами в различных системах счисления:

1. $101,101_2 + 10,01_2$
2. $101,101_2 - 10,01_2$
3. $11,01_2 * 0,1_2$
4. $C3, 8_{16} + 3A, D_{16}$
5. $C3, 8_{16} - 3A, D_{16}$
6. $3, F_{16} * 6, A_{16}$

Выполнить проверку полученных результатов путем их перевода в десятичную систему счисления

Практическое занятие № 6 «Выполнение арифметических операций над числами с фиксированной точкой».

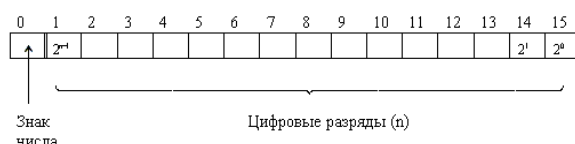
Цель занятия: научиться выполнять арифметических операций над числами с фиксированной точкой.

Содержание работы:

Числа с фиксированной точкой.

Запись числа с фиксированной точкой обычно имеет знаковый и цифровой разряды. Фиксированная точка означает, что на этапе конструирования ЭВМ было определено, сколько и какие разряды машинного слова отведены под изображение целой и дробной частей числа. Пример. Как частный случай числа с фиксированной точкой может быть рассмотрена запись целого числа (в этом случае все разряды, кроме знакового, используются для записи целой части).

Пример. Ячейка с записью целого числа.



К достоинствам использования чисел с фиксированной точкой относятся простота выполнения арифметических операций и высокая точность изображения чисел. К недостаткам - небольшой диапазон представления чисел.

Арифметические операции над числами с фиксированной точкой

Сложение (вычитание). Операция вычитания приводится к операции сложения путем преобразования чисел в обратный или дополнительный код. Пусть числа $A \Rightarrow O$ и $B \Rightarrow O$, тогда операция алгебраического сложения выполняется в соответствии с табл. 1.

Таблица 1

Таблица преобразования кодов при алгебраическом сложении

Требуемая операция	Необходимое преобразование
$A+B$	$A+B$
$A-B$	$A+(-B)$
$-A+B$	$(-A)+B$
$-A-B$	$(-A)+(-B)$

Скобки в представленных выражениях указывают на замену операции вычитания операцией сложения с обратным или дополнительным кодом соответствующего числа. При выполнении сложения цифр необходимо соблюдать следующие правила.

1. Слагаемые должны иметь одинаковое число разрядов. Для выравнивания разрядной сетки слагаемых можно дописывать незначащие нули слева к целой части числа и незначащие нули справа к дробной части числа.

2. Знаковые разряды чисел участвуют в сложении так же, как и значащие.

3. Необходимые преобразования кодов производятся с изменением знаков чисел. Приписанные незначащие нули изменяют свое значение при преобразованиях по общему правилу.

4. При образовании единицы переноса из старшего знакового разряда, в случае использования ОК, эта единица складывается с младшим числовым разрядом. При использовании ДК единица переноса теряется. Знак результата формируется автоматически, результат представляется в том коде, в котором представлены исходные слагаемые.

Пример 1. Сложить два числа $A_{10}=7$ $B_{10}=16$

$$A_2=+11=+0111;$$

$$B_2=+1000=+10000.$$

Исходные числа имеют различную разрядность, необходимо провести выравнивание разрядной сетки:

$$[A_2]_{\Pi}=[A_2]_{\text{ОК}}=[A_2]_{\text{ДК}}=0: 00111;$$

$$[B_2]_{\Pi}=[B_2]_{\text{ОК}}=[B_2]_{\text{ДК}}=0: 10000.$$

Сложение в обратном или дополнительном коде дает один и тот же результат

$$0: 00111$$

$$+0: 10000$$

$$C_2=0: 10111$$

$$C_{10}=+23.$$

Обратим внимание, что при сложении цифр отсутствуют переносы в знаковый разряд и из знакового разряда, что свидетельствует о получении правильного результата.

Пример 2 Сложить два числа $A_{10}=+16$ $B_{10}=\approx 7$ в ОК и ДК. В соответствии с табл. 1 должна быть реализована зависимость $A+(-B)$, в которой второй член преобразуется с учетом знака

$[A_2]_n = 0; 10000 = 0; 10000$ $[A_2]_{ок} = 0; 10000$ $[A_2]_{дк} = 0; 10000;$
 $[B_2]_n = 1; 111 = 1; 00111$ $[B_2]_{ок} = 1; 11000$ $[B_2]_{дк} = 1; 11001.$

Сложение в ОК
 $[A_2]_{ок} = 0; 10000$
 $+ [B_2]_{ок} = 1; 11000$
 \hline
 $\begin{array}{r} 0; 01000 \\ + \quad \quad \quad 1 \\ \hline 0; 01001 \end{array}$

$C_2 = 0; 01001$
 $C_{10} = +9$

Сложение в ДК
 $[A_2]_{дк} = 0; 10000$
 $+ [B_2]_{дк} = 1; 11001$
 \hline
 $\begin{array}{r} 0; 01001 \end{array}$

$C_2 = 0; 01001$
 $C_{10} = +9$

При сложении чисел в ОК и ДК были получены переносы в знаковый разряд и из знакового разряда. В случае ОК перенос из знакового разряда требует дополнительного прибавления единицы младшего разряда (см. п.4 правил). В случае ДК этот перенос игнорируется.

Умножение. Умножение двоичных чисел наиболее просто реализуется в прямом коде. Рассмотрим, каким образом оно приводится к операциям сложения и сдвигам.

Пример 3. Умножить два числа $A_{10}=7$ $B_{10}=5$.

Перемножим эти числа, представленные прямыми двоичными кодами, так же, как это делается в десятичной системе.

Нетрудно видеть, что произведение получается путём сложения частных произведений, представляющих собой разряды множимого, сдвинутые влево в соответствии с позициями разрядов множителя. Частные произведения, полученные умножением на нуль игнорируются. Важной особенностью операции умножения n -разрядных сомножителей является увеличение разрядности произведения до $n+n=2n$. Знак произведения формируется путём сложения знаковых разрядов сомножителей. Возможные переносы из знакового разряда игнорируются.

Деление. Операция деления, как и в десятичной арифметике, является обратной операции умножения. Покажем, что и эта операция приводится к последовательности операций сложения и сдвига.

Пример 4. Разделить два числа $A_{10}=45$ $B_{10}=5$

$[A_2]_n = 101101$
 $[B_2]_n = 101$
Делимое Делитель
 $\begin{array}{r} 101101 \\ \underline{101} \\ 0101 \\ \underline{101} \\ 0 \end{array}$ $\begin{array}{r} 101 \\ \hline 1001 - \text{частное} \end{array}$
 $[C_2]_n = 1001$
 $C_{10} = 9$

Деление произведено так же, как это делается обычно в десятичной системе. Сначала проверяется, можно ли вычесть значение делителя из старших разрядов делимого. Если возможно, то в разряде частного записывается единица и определяется частная разница. В противном случае в частное записывается нуль и разряды делителя сдвигаются вправо на один разряд по отношению к разрядам делимого. К полученной предыдущей разнице сносится очередная цифра делимого, и данный процесс повторяется, пока не будет получена необходимая точность. Если учесть, что все вычитания в ЭВМ заменяются сложением в ОК или в ДК (см. табл.1), то действительно операция деления приводится к операциям сложения и сдвигам вправо разрядов делителя относительно разрядов делимого. Отметим, что делимое перед операцией деления должно быть приведено к $2n$ -разрядной сетке. Только в этом случае при делении на n -разрядный делитель получается n -разрядное частное.

Знак частного формируется также путем сложения знаковых разрядов делимого и делителя, как это делалось при умножении.

Порядок выполнения работы:

Выполните арифметические операции в естественной и нормальной форме:

1. сложение
2. вычитание
3. умножение
4. деление

Практическое занятие № 7 «Выполнение арифметических операций над числами с плавающей точкой»

Цели занятия: научиться выполнять арифметических операций над числами с плавающей точкой.

Содержание работы:

Числа с плавающей точкой.

Для представления чисел с плавающей точкой (ЧПТ) используется полулогарифмическая форма записи числа:

$$N = \pm m q^{\pm p}$$

где q - основание системы счисления, p - порядок числа, m - мантисса числа N .

Положение точки определяется значением порядка p . С изменением порядка точка перемещается (плавает) влево или вправо. Пример.

$$125_{10} = 12.5 \cdot 10^1 = 1.25 \cdot 10^2 = 0.125 \cdot 10^3 = 0.0125 \cdot 10^4 = \dots$$

Для установления однозначности при записи чисел принята **нормализованная форма** записи числа. Мантисса нормализованного числа может изменяться в диапазоне: $1/q \leq |m| < 1$. Таким образом в нормализованных числах цифра после точки должна быть значащей.

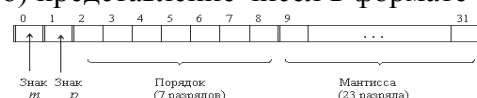
$$\underbrace{0.0832 \cdot 10^3}_{\text{ненормализованное число}} = \underbrace{0832 \cdot 10^2}_{\text{нормализованное число}}$$

Пример.

Для представления чисел в машинном слове выделяют группы разрядов для изображения мантиссы, порядка, знака числа и знака порядка: а) представление чисел в формате полуслова



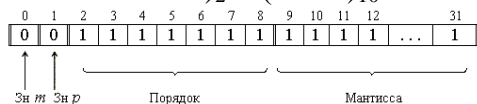
б) представление чисел в формате слова



Наиболее типично представление ЧПТ в формате слова (32 разряда). Пример.
Число $A = -3.5_{10} = -11.1_2 = -0.111 \cdot 10^{10}$

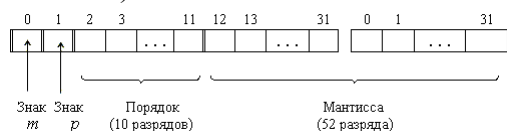


Максимальным числом представимым в формате слова будет $A = (0.1111...1 \cdot 10^{1111111})_2 \approx (1 \cdot 2^{127})_{10}$.



Таким образом числа с плавающей точкой позволяют увеличить диапазон обрабатываемых чисел, но при этом точность изображения чисел определяется только разрядами мантиссы и уменьшается по сравнению с числами с фиксированной точкой. При записи числа в формате слова диапазон представимых чисел будет от $-1 \cdot 2^{127}$ до $1 \cdot 2^{127}$ ($2^{127} \approx 10^{38}$), а точность определяться мантиссой, состоящей из 23 разрядов. Точность может быть повышена путем увеличения количества разрядов мантиссы. Это реализуется

путем представления чисел с так называемой двойной точностью (используется формат двойного слова):



Арифметические операции над двоичными числами с плавающей точкой

В современных ЭВМ числа с плавающей точкой хранятся в памяти машин, имея мантиссу и порядок (характеристику) в прямом коде и нормализованном виде. Все арифметические действия над этими числами выполняются так же, как это делается с ними, если они представлены в полулогарифмической форме (мантисса и десятичный порядок) в десятичной системе счисления. Порядки и мантиссы обрабатываются отдельно.

Сложение (вычитание). Операция сложения (вычитания) производится в следующей последовательности.

1. Сравниваются порядки (характеристики) исходных чисел путем их вычитания $p = p_1 - p_2$. При выполнении этой операции определяется, одинаковый ли порядок имеют исходные слагаемые.
2. Если разность порядков равна нулю, то это значит, что одноименные разряды мантисс имеют одинаковые веса (двоичный порядок). В противном случае должно проводиться выравнивание порядков.
3. Для выравнивания порядков число с меньшим порядком сдвигается вправо на разницу порядков Δp . Младшие выталкиваемые разряды при этом теряются.
4. После выравнивания порядков мантиссы чисел можно складывать (вычитать) в зависимости от требуемой операции. Операция вычитания заменяется операцией сложения в соответствии с данными табл. 2.3. Действия над слагаемыми производятся в ОК или ДК по общим правилам.
5. Порядок результата берется равным большему порядку.
6. Если мантисса результата не нормализована, то осуществляются нормализация и коррекция значений порядка.

Пример 5. Сложить два числа $A_{10} = +1.375$; $B_{10} = -0.625$.

$A_2 = +1.011 = 0:1011 \cdot 10^1$; $B_2 = -0.101 = -0:101 \cdot 10^0$.

В нормализованном виде эти числа будут иметь вид:

Порядок	Мантисса
$[A_2]_{II} = 0:1$	$\uparrow 0:1011$
	знак числа
$[B_2]_{II} = 0:0$	$\downarrow 1:101$

1. Вычитаем порядки $\Delta p = p_1 - p_2 = 1 - 0 = 1$. В машине эта операция требует операции сложения с преобразованием порядка чисел в дополнительный код:

$$\begin{array}{rcl} p_1 = 0:1 & [p_1]_{ДК} = 0:1 & \\ p_2 = 0:0 & + [p_2]_{ДК} = 0:0 & \\ \hline \Delta p = 0:1 & & \end{array}$$

Определяем, что $\Delta p \neq 0$.

2. Порядок первого числа больше порядка второго числа на единицу. Требуется выравнивание порядков.

3. Для выравнивания порядков необходимо второе число сдвинуть вправо на один разряд.

$[B_2]_{исх} = 0:1:101$

после сдвига

$[B_2]_п = 0:11:0101$

$[m_B]_{ДК} = 1:1011$

4. Складываем мантиссы.

$$\begin{array}{r} [m_A]_{ДК} = 0:1011 \\ + [m'_B]_{ДК} = 1:1011 \\ \hline [m_C]_{ДК} = 0:0110 \end{array}$$

Мантисса числа С - положительная.

5. Порядок числа С равен порядку числа с большим порядком, т.е. $p = +1$.

$[C_2]_{II} = 0:10:0110$.

Видно, что мантисса результата не нормализована, так как старшая цифра мантиссы равна нулю.

6. Нормализуем результат путем сдвига мантиссы на один разряд влево и соответственно вычитаем из значения порядка единицу:

$[C_2]_{II} = 0:00:110$

$C_{10} = +0.75$.

Умножение (деление). Операция умножения (деления) чисел с плавающей точкой также требует разных действий над порядками и мантиссами. Алгоритмы этих операций выполняются в следующей последовательности.

1. При умножении (делении) порядки складываются (вычитаются) так, как это делается над числами с фиксированной точкой.

2. При умножении (делении) мантиссы перемножаются (делятся).

3. Знаки произведения (частного) формируются путем сложения знаковых разрядов сомножителей (делимого и делителя). Возможные переносы из знакового разряда игнорируются.

Задание:

	I	II	III	IV	V	VI
1	149,375 ₁₀	711,25 ₁₀	360,25 ₁₀	741,125 ₁₀	597,25 ₁₀	237,73 ₁₀
	65,2 ₁₆	-3BF, A ₁₆	-2FE, 6 ₁₆	24A, 4 ₁₆	24D, 5 ₁₆	BB, 4 ₁₆
2	101010000,10 111	1100110110,001 1	11001010,0 1	1111100100,1101 1	1101111111,1 1	1000101001, 1
	11001100,01	11111110,01	1110001,00 1	101110111,011 1	1100111110,101 1	1111101,1
3	X=69,4 ₁₆ Y=A, D ₁₆	X=2B, A ₁₆ ; Y=36, 6 ₁₆	X=7, 4 ₁₆ ; Y=1D, 4 ₁₆	X=36, 4 ₁₆ ; Y=A, A ₁₆	X=4B, 2 ₁₆ ; Y=3C, 3 ₁₆	X=4A, 3 ₁₆ Y=F, 6 ₁₆ ;
4	X=326,8 ₁₀ ; Y=-15,2 ₁₀	X=-220,15 ₁₀ ; Y=6,29 ₁₀	X=150,46 ₁₀ ; Y=-7,523 ₁₀	X=365,02 ₁₀ ; Y=-18,251 ₁₀	X=-128,3 ₁₀ ; Y=6,415 ₁₀	X=421,2 ₁₀ ; Y=-11,7 ₁₀

Порядок выполнения работы:

Выполните арифметические операции в естественной и нормальной форме:

5. сложение
6. вычитание
7. умножение
8. деление

Практическое занятие № 8 «Выполнение арифметических операций над двоично-десятичными кодами чисел»

Цель занятия: научиться выполнять арифметические операции над двоично-десятичными кодами чисел.

Содержание работы:

Задание: Выполните арифметические операции.

Образец выполнения задания:

Даны два числа 1111₂ и 110₂

1. Выполним сложение чисел:

+	1	1	1	1
		1	1	0
1	0	1	0	1

2. Выполним вычитание чисел:

-	1	1	1	1
		1	1	0
	1	0	0	1

3. Выполним умножение чисел:

	*	1	1	1	1	
				1	1	0
+		1	1	1	1	
		1	1	1	1	
1	0	1	1	0	1	0

4. Выполним деление чисел:

-	1	1	1	1		1	1	0
	1	1	0			1	0,	1
			1	1	0			
		-	1	1	0			
					0			

Задачи для решения.

- Дано $A=A71_6$, $B=251_8$. Какое из чисел C , записанных в двоичной системе, отвечает условию $A < C < B$? 1) 10101100_2 2) 10101010_2 3) 10101011_2 4) 10101000_2
- Вычислите сумму чисел X и Y , если $X=110111_2$, $Y=135_8$. Результат представьте в двоичном виде.
 - 11010100_2
 - 10100100_2
 - 10010011_2
 - 10010100_2
- Вычислите значение выражения $206_8 + AF_{16} - 11001010_2$. Вычисления производите в двоичной с/с. Переведите ответ в десятичную с/с.

1 вариант

Выполните сложение, вычитание, умножение и деление чисел 1011110_2 , 1101_2 в двоичной системе счисления.

2 вариант

Выполните сложение, вычитание, умножение и деление чисел 1011101_2 , 1010_2 в двоичной системе счисления.

Цель занятия: Научиться кодировать текстовую информацию.

Содержание работы:

Напоминание: Если сообщение состоит из символов некоего алфавита (и все символы равно вероятны). То количество информации в сообщении I вычисляется по формуле:

$$I = k \cdot i$$

Где I – количество информации в сообщении, i – найти из формулы $2 = N$

1. Сколько бит памяти займет слово «Микропроцессор»?

Решение:

Слово состоит из 14 букв. Каждая буква – символ компьютерного алфавита, занимает 1 байт памяти. Слово занимает 14 байт $= 14 \cdot 8 = 112$ бит памяти.

Ответ: 112 бит

2. Текст занимает 0, 25 Кбайт памяти компьютера. Сколько символов содержит этот текст? ([1], с.133, №31)

Решение:

Переведем Кб в байты: $0, 25 \text{ Кб} \cdot 1024 = 256$ байт. Так как текст занимает объем 256 байт, а каждый символ – 1 байт, то в тексте 256 символов.

Ответ: 256 символов

3. 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? ([1], с.133, №32)

Решение:

$30 \cdot 70 \cdot 5 = 10500$ символов в тексте на 5 страницах. Текст займет 10500 байт оперативной памяти.

Ответ: 10500 байт

4. Считая, что каждый символ кодируется одним байтом, оцените информационный объем следующего предложения из пушкинского четверостишия:

Певец-Давид был ростом мал, Но повалил же Голиафа! (ЕГЭ_2005. демо, уровень А)

- 1)
400 бит
- 2)
50 бит
- 3)
400 байт
- 4)
5 байт

Решение:

В тексте 50 символов, включая пробелы и знаки препинания. При кодировании каждого символа одним байтом на символ будет приходиться по 8 бит, Следовательно, переведем в биты $50 \cdot 8 = 400$ бит.

Ответ: 400 бит

5. Автоматическое устройство осуществило перекодировку информационного сообщения длиной 48 символов, первоначально записанного в 7–битном коде ASCII, в 16–битную кодировку Unicode.

При этом информационное сообщение увеличилось на

- 1) 48 байт 2) 96 байт 3) 54 байт 4) 432 байт

Решение:

6. Изменение кодировки с 7 бит на 16 бит, равно $16 - 7 = 9$ бит. Следовательно информационный объем каждого символа сообщения увеличился на 9 бит. Так как длина сообщения равна 48 символов, информационное сообщение увеличилось на $48 * 9 = 432$ бита. В предложенных ответах данного ответа нет, следовательно переведем биты в байты - $432/8 = 54$ байта.

Ответ: Информационное сообщение увеличилось на 54 байта.

7. Автоматическое устройство осуществило перекодировку информационного сообщения на русском языке, первоначально записанного в 16-битном коде Unicode, в 8-битную кодировку Windows-1251, при этом информационный объем сообщения составил 60 байт. Определите информационный объем сообщения до перекодировки.

1) 60 бит 2) 120 бит 3) 960 бит 4) 60 байт

Решение:

8. По условиям задачи информационный объем сообщения после кодировки составил 60 байт = 480 бит, следовательно количество символов в сообщении = $480/8 = 60$. До перекодировки сообщение кодировалось 16-битным кодом, следовательно информационный объем сообщения до перекодировки = $60 * 16 = 960$ бит.

Ответ: информационный объем сообщения до перекодировки 960 бит.

9. В кодировке Unicode на каждый символ отводится два байта. Определите информационный объем слова из двадцати четырех символов в этой кодировке.

1) 384 бита 2) 192 бита 3) 256 бит 4) 48 бит

Решение:

Количество символов в сообщении - 24, каждый символ кодируется 2 байтами = 16 бит, следовательно $24 * 16 = 384$ бит.

Ответ: 384 бита.

Практическое занятие № 10 «Кодирование графической информации»

Цель занятия: научиться кодировать графическую информацию.

Содержание работы:

Задание 1. Установить графический режим экрана монитора:

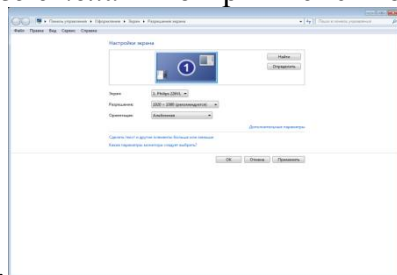
- с наиболее возможным высоким разрешением экрана;
- с наиболее возможной глубиной цвета.

Задание 2. В графическом редакторе последовательно установить цвета (см. табл. 1.3. «Кодировка при глубине цвета 24 бита» на стр. 17 учебника и табл. 1.4. «Формирование цветов в системе цветопередачи CMYK») с использованием палитр цветов RGB и CMYK. Цвета устанавливать путем введения числовых кодов базовых цветов в соответствующие текстовые поля.

Задание 1. Установка графического режима экрана монитора в операционной

системе Windows 7 

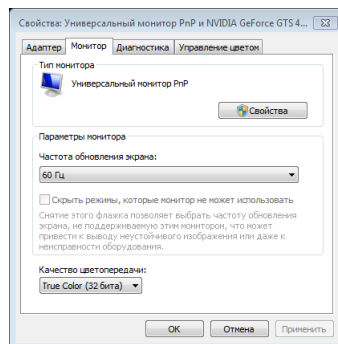
1. Для установки разрешения экрана щелкнуть правой кнопкой мыши в пустом месте *Рабочего стола* и в открывшемся контекстном меню выбрать пункт *Разрешение*



экрана.

Разрешающая способность экрана компьютера устанавливается с помощью раскрывающегося списка *Разрешение:* и ползунок, позволяющего менять параметры разрешения от самого низкого до максимально высокого.

2. Глубина цвета устанавливается при переходе по гиперссылке *Дополнительные параметры* в окне *Разрешение экрана*. При нажатии на данную ссылку открывается диалоговая панель *Свойства монитора*. В раскрывающемся списке *Качество цветопередачи:* на вкладке *Монитор* можно выбрать необходимую глубину цвета экрана.



3. Изменив параметры экрана и просмотрев результаты внесенных изменений, вернуть исходные настройки.

Установка цвета в графическом редакторе с использованием системы цветопередачи

RGB в векторном редакторе OpenOffice.org Draw



1. Запустить интегрированное офисное приложение OpenOffice.org Draw командой [Пуск – Все программы – OpenOffice.org 3.3 – OpenOffice.org Draw].
2. Используя панель *Рисование*, расположенную внизу окна программы, нарисовать восемь одинаковых фигур (например, прямоугольников).

Для каждой фигуры зададим цвет из таблицы кодировки цветов при глубине цвета 24 бита (стр. 17 учебника, табл. 1.3)

3. Выделить фигуру и выполнить одну из команд: [Формат – Область...] или [щелчок правой кнопкой мыши на фигуре – Область...].

На появившемся диалоговом окне *Область* выбрать вкладку *Цвета*.

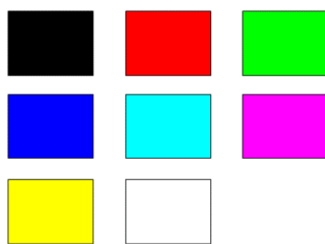
С помощью раскрывающегося списка выбрать систему цветопередачи RGB.

Задать цвет путем установки в полях со счетчиком интенсивностей базовых цветов, вручную вводя числа, данные в таблице 1.3, в текстовые поля **R**, **G** и **B** и нажимая *ОК*.

В случае появления системного сообщения следующего содержания: «Цвет был изменен без сохранения. Измените выбранный или добавьте новый цвет», щелкнуть по кнопке *Изменить*.

4. Выполнить пункт 3 для оставшихся фигур.

В результате будут получены восемь фигур, закрашенные цветами, указанными в таблице.



5. Результат работы сохранить в собственной папке под именем RGB.
6. Создайте новый документ OpenOffice.org Draw командой [*Файл – Создать – Рисунок*].
7. Используя панель *Рисование* нарисовать восемь одинаковых фигур (например, эллипсов).

Для каждой фигуры зададим цвет из таблицы формирования цветов в системе цветопередачи СМΥΚ (стр. 19 учебника, табл. 1.4)

8. Выделить фигуру и выполнить одну из команд: [*Формат – Область...*] или [щелчок правой кнопкой мыши на фигуре – *Область...*].
9. На появившемся диалоговом окне *Область* выбрать вкладку *Цвета*.
10. С помощью раскрывающегося списка выбрать систему цветопередачи СМΥΚ.
11. Задать цвет путем установки в полях со счетчиком интенсивностей базовых цветов, вручную вводя процентные соотношения, данные в таблице 1.4, в текстовые поля **С**, **М**, **Υ** и **Κ** и нажимая *OK*.

Формулы формирования цветов в системе цветопередачи СМΥΚ расшифровываются следующим образом: например, формула **Black** = **Κ** = **С** + **М** + **Υ** = **W** – **G** – **B** – **R** означает, что цвет *Black* («черный») получается в соотношении **Κ** = 100%, или **С** = 100% + **М** = 100% + **Υ** = 100%, или при исключении цветов *White* (белого), *Green* (зеленого), *Blue* (синего) и *Red* (красного).

12. Результат работы сохранить в собственной папке под именем СМΥΚ.

Практическое занятие № 11 «Кодирование звуковой информации»

Цель занятия: научиться кодировать звуковую информацию.

Содержание работы:

Вариант 1.

1. Найдите информационный цифрового звукового 16-ти битного стереофайла (24000 Гц), звучащего 2 минуты.
2. Загрузите в приложение «Звукозапись» файл «Музыка 1».
 - а) Определите объем файла.
 - б) Сравните полученный результат с объемом, который определяет ОС Windows.
 - в) Преобразуйте его в файл с максимально возможными параметрами. Увеличился или уменьшился объем и во сколько раз?
3. Смикшируйте данный файл с файлом «Музыка 2» Ответьте на вопросы.
 - а) равен ли объем смикшированного файла сумме смешанных файлов?
 - б) изменится ли объем файла, Если Вы увеличите скорость и громкость прослушивания?
 - в) на сколько объем файла, если Вы добавите какой- либо эффект прослушивания?

Вариант 2.

1. Найдите информационный цифрового звукового 16-ти битного стереофайла (24000 Гц), звучащего 2 минуты.
2. Загрузите в приложение «Звукозапись» файл «Музыка 1».
 - а) Определите объем файла.
 - б) Сравните полученный результат с объемом, который определяет ОС Windows.
 - в) Преобразуйте его в файл с максимально возможными параметрами. Увеличился или уменьшился объем и во сколько раз?

3. Смикшируйте данный файл с файлом «Музыка 2» Ответьте на вопросы.
- а) равен ли объем смикшированного файла сумме смешанных файлов?
 - б) изменится ли объем файла, Если Вы увеличите скорость и громкость прослушивания?
 - в) на сколько объем файла, если Вы добавите какой- либо эффект прослушивания?

Вариант 3.

1. Найдите информационный цифрового звукового 16-ти битного стереофайла (24000 Гц), звучащего 2 минуты.
2. Загрузите в приложение «Звукозапись» файл «Музыка 1».

 - а) Определите объем файла.
 - б) Сравните полученный результат с объемом, который определяет ОС Windows.
 - в) Преобразуйте его в файл с максимально возможными параметрами. Увеличился или уменьшился объем и во сколько раз?

3. Смикшируйте данный файл с файлом «Музыка 2» Ответьте на вопросы.

 - а) равен ли объем смикшированного файла сумме смешанных файлов?
 - б) изменится ли объем файла, Если Вы увеличите скорость и громкость прослушивания?
 - в) на сколько объем файла, если Вы добавите какой- либо эффект прослушивания?

Практическое занятие № 12 «Кодирование числовой информации»

Цель занятия: научиться кодировать числовую информацию.

Содержание работы:

Пример задания:

Для кодирования букв А, Б, В, Г решили использовать двухразрядные последовательные двоичные числа (от 00 до 11, соответственно). Если таким способом закодировать последовательность символов БАВГ и записать результат шестнадцатеричным кодом, то получится

- 1) $4B_{16}$ 2) 411_{16} 3) $BACD_{16}$ 4) 1023_{16}

Решение:

- 1) из условия коды букв такие: А – 00, Б – 01, В – 10 и Г – 11, код равномерный
- 2) последовательность БАВГ кодируется так: 01 00 10 11 = 1001011
- 3) разобьем такую запись на тетрады справа налево и каждую тетраду переведем в шестнадцатеричную систему (то есть, сначала в десятичную, а потом заменим все числа от 10 до 15 на буквы А, В, С, D, E, F); получаем $1001011 = 0100\ 1011_2 = 4B_{16}$
- 4) правильный ответ – 1.

Пример задания:

Для 5 букв латинского алфавита заданы их двоичные коды (для некоторых букв – из двух бит, для некоторых – из трех). Эти коды представлены в таблице:

A	B	C	D	E
---	---	---	---	---

Определить, какой набор букв закодирован двоичной строкой 0110100011000

- 1) EVCEA 2) BDDEA 3) BDCEA 4) EBAEA

Решение (вариант 1, декодирование с начала):

- 1) здесь используется неравномерное кодирование, при котором декодирование может быть неоднозначным, то есть, заданному коду может соответствовать несколько разных исходных сообщений
- 2) попробуем декодировать с начала цепочки, первой буквой может быть В или Е, эти случаи нужно рассматривать отдельно
- 3) пусть первая буква – Е с кодом 011, тогда остается цепочка 0100011000
 $\frac{3}{4}$ для кода 0100011000 первой буквой может быть только В с кодом 01, тогда остается 00011000 (начало исходной цепочки – EV?)

$\frac{3}{4}$ для кода 00011000 первой буквой может быть только А с кодом 000, тогда остается 11000, а эта цепочка не может быть разложена на заданные коды букв

$\frac{3}{4}$ поэтому наше предположение о том, что первая буква – Е, неверно

4) пусть первая буква – В с кодом 01, тогда остается цепочка 10100011000

$\frac{3}{4}$ для кода 10100011000 первой буквой может быть только D с кодом 10, тогда остается 100011000 (можно полагать, что начало исходной цепочки – BD?)

$\frac{3}{4}$ для кода 100011000 первой буквой может быть только С с кодом 100, тогда остается 011000 (начало исходной цепочки – BDC?)

Несмотря на то, что среди ответов есть единственная цепочка, которая начинается с BDC, здесь нельзя останавливаться, потому что «хвост» цепочки может «не сойтись»

$\frac{3}{4}$ для кода 011000 на первом месте может быть В (код 01) или Е (011); в первом случае «хвост» 1000 нельзя разбить на заданные коды букв, а во втором – остается код 000 (буква А), поэтому исходная цепочка может быть декодирована как BDCEA

5) правильный ответ – 3

Решение (вариант 2, декодирование с конца):

1) для кода 0110100011000 последней буквой может быть только А (код 000), тогда остается цепочка 0110100011

2) для 0110100011 последней может быть только буква Е (011), тогда остается цепочка 0110100

3) для 0110100 последней может быть только буква С (100), тогда остается цепочка 0110

4) для 0110 последней может быть только буква D (10), тогда остается 01 – это код буквы В

5) таким образом, получилась цепочка BDCEA

6) правильный ответ – 3

Решение (вариант 3, кодирование ответов):

1) в данном случае самое простое и надежное – просто закодировать все ответы, используя приведенную таблицу кодов, а затем сравнить результаты с заданной цепочкой

2) получим

1) EBCEA – 01101100011000 2) BDDEA – 011010011000

3) BDCEA – 0110100011000 4) EBAEA – 01101000011000

3) сравнивая эти цепочки с заданной, находим, что правильный ответ – 3.

Пример задания:

Для передачи по каналу связи сообщения, состоящего только из букв А, Б, В, Г, решили использовать неравномерный по длине код: А=0, Б=10, В=110. Как нужно закодировать букву Г, чтобы длина кода была минимальной и допускалось однозначное разбиение кодированного сообщения на буквы?

1) 1 2) 1110 3) 111 4) 11

Решение (вариант 1, метод подбора):

1) рассмотрим все варианты в порядке увеличения длины кода буквы Г

2) начнем с Г=1; при этом получается, что сообщение «10» может быть раскодировано двояко: как ГА или Б, поэтому этот вариант не подходит

3) следующий по длине вариант – Г=11; в этом случае сообщение «110» может быть раскодировано как ГА или В, поэтому этот вариант тоже не подходит

4) третий вариант, Г=111, дает однозначное раскодирование во всех сочетаниях букв, поэтому...

5) ... правильный ответ – 3.

Решение (вариант 2, «умный» метод):

1) для того, чтобы сообщение, записанное с помощью неравномерного по длине кода, однозначно раскодировалось, требуется, чтобы никакой код не был началом другого (более длинного) кода; это условие называют *условием Фано*

- 3) но бит четности нам совсем **не нужен**, важно другое: пятый бит в каждой пятерке **можно отбросить!**
- 4) разобьём заданную последовательность на группы по 5 бит в каждой:
01010, 10010, 01111, 00011.
- 5) отбросим пятый (последний) бит в каждой группе:
0101, 1001, 0111, 0001.
- это и есть двоичные коды передаваемых чисел:
 $0101_2 = 5$, $1001_2 = 9$, $0111_2 = 7$, $0001_2 = 1$.
- 6) таким образом, были переданы числа 5, 9, 7, 1 или число 5971.
- 7) Ответ: 2.

Практическое занятие № 13 «Построение таблиц истинности для логических высказываний»

Цель занятия: Выработать навык построения таблиц истинности для логических высказываний.

Содержание работы:

Таблица истинности – Для каждого составного высказывания (логического выражения) можно построить таблицу истинности, которая определяет его истинность или ложность при всех возможных комбинациях исходных значений простых высказываний (логических переменных).

Для построения таблиц истинности существует определенный алгоритм:

- Определить количество переменных (простых выражений);
- Определить количество логических операций и последовательность их выполнения (инверсия, конъюнкция, дизъюнкция).
- Определить количество строк:

количество строк = 2^n + строка для заголовка,

где n – количество логических переменных.

- Определить количество столбцов: **количество столбцов** = количество переменных + количество логических операций;

- Заполнить столбцы результатами выполнения логических операций в обозначенной последовательности с учетом таблиц истинности основных логических операций.

Рассмотрим примеры построения таблиц истинности.

Пример. Составьте таблицу истинности логического выражения:

$\neg A \& (B \vee C)$.

Решение:

1. Определим количество строк по известной нам формуле: **количество строк** = 2^n + строка для заголовка, т.к. у нас три простых высказывания A, B, C поэтому $n = 3$ и количество строк = $2^3 + 1 = 9$.

2 Определим количество столбцов:

- простые выражения (переменные): A, B, C.

- промежуточные результаты (логические операции): $\neg A$ – инверсия (еще можно обозначить \bar{A}), $B \vee C$ - операция дизъюнкции, $\neg A \& (B \vee C)$ - операция конъюнкции.

3. Заполним столбцы с учетом таблиц истинности логических операций.

A	B	C	\bar{A}	$B \vee C$	$\neg A \& (B \vee C)$
0	0	0	1	0	0
0	0	1	1	1	1
0	1	0	1	1	1
0	1	1	1	1	1
1	0	0	0	0	0
1	0	1	0	1	0

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

Равносильные логические выражения – логические выражения, у которых таблицы истинности совпадают, называются равносильными (эквивалентными). Для обозначения равносильных логических выражений используется знак “ \equiv ”.

Пример. Докажем, что логические выражения $\bar{A} \cdot B$ и $A \vee \bar{B}$ равносильны.

Построим сначала таблицу истинности логического выражения: $\bar{A} \cdot B$

1) Определим количество строк в таблице. Для этого: считаем количество переменных, в нашем случае логическая функция содержит 2 переменные: A и B. Количество строк в таблице истинности должно быть равно $2^2=4$.

2) Определяем количество столбцов. Это количество логических переменных + количество логических операций. В нашем случае количество переменных = 2, а количество логических операций = 1, т.е. количество столбцов таблицы истинности = 3.

3) Строим таблицу истинности с указанным количеством строк и столбцов, обозначаем столбцы и вносим в таблицу возможные наборы значений исходных логических переменных и заполняем таблицу истинности по столбцам.

A	B	$\bar{A} \cdot B$
0	0	0
0	1	1
1	0	0
1	1	0

Теперь построим таблицу истинности логического выражения: $A \vee \bar{B}$

1) Определим количество строк в таблице. Для этого: считаем количество переменных, в нашем случае логическая функция содержит 2 переменные: A и B. Количество строк в таблице истинности должно быть равно $2^2=4$.

2) Определяем количество столбцов. Это количество логических переменных + количество логических операций. В нашем случае количество переменных = 2, а количество логических операций = 1, т.е. количество столбцов таблицы истинности = 3.

3) Строим таблицу истинности с указанным количеством строк и столбцов, обозначаем столбцы и вносим в таблицу возможные наборы значений исходных логических переменных и заполняем таблицу истинности по столбцам.

Сначала необходимо выполнить действие в скобках, а затем логическое отрицание.

A	B	$A \vee B$	$\overline{A \vee B}$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

Построили таблицы. Теперь давайте, сравним значения в последних столбцах таблиц истинности, т.к. именно последние столбцы являются результирующими:

$$\bar{A} \cdot B = \overline{A \vee \bar{B}}$$

Задание 1. Постройте таблицу истинности для логического выражения: $(A \vee B) \& (\neg A \vee \neg B)$.

1) Определим количество строк в таблице истинности. Для этого: считаем количество переменных, в нашем случае логическая функция содержит 2 переменные: A и B. Количество строк в таблице истинности должно быть равно $2^2=4$.

2) Определим количество столбцов. Это количество логических переменных + количество логических операций. В нашем случае количество переменных = 2, а количество логических операций = 2, т.е. количество столбцов таблицы истинности = 4.

3) Строим таблицу истинности с указанным количеством строк и столбцов, обозначаем столбцы и вносим в таблицу возможные наборы значений исходных логических переменных и заполняем таблицу истинности по столбцам.

A	B	$A \vee B$	$\neg A$	$\neg B$	$\neg A \vee \neg B$	$(A \vee B) \& (\neg A \vee \neg B)$
0	0	0	1	1	1	0
0	1	1	1	0	1	1
1	0	1	0	1	1	1
1	1	1	0	0	0	0

Задание 2. (ученики решают самостоятельно, а потом происходит коллективная проверка). Составьте таблицу истинности для логического выражения: $A \vee A \& B$.

A	B	$A \& B$	$A \vee A \& B$
0	0	0	0
0	1	0	0
1	0	0	1
1	1	1	1

Обратите внимание, что последний столбец (результат) совпал со столбцом A. Значит можно сказать, что логическое выражение $A \vee A \& B$ равносильно логической переменной A.

Теперь давайте откроем учебник на странице 39 и выполним упражнение №8. Постройте таблицы истинности для следующих логических выражений:

$$B \& (A \vee B).$$

1) Определим количество строк в таблице истинности. Для этого: считаем количество переменных, в нашем случае логическая функция содержит 2 переменные: A и B. Количество строк в таблице истинности должно быть равно $2^2 + 1 = 5$.

2) Определим количество столбцов. Это количество логических переменных + количество логических операций. В нашем случае количество переменных = 2, а количество логических операций = 2, т.е. количество столбцов таблицы истинности = 4.

3) Строим таблицу истинности с указанным количеством строк и столбцов, обозначаем столбцы и вносим в таблицу возможные наборы значений исходных логических переменных и заполняем таблицу истинности по столбцам.

A	B	$(A \vee B)$	$B \& (A \vee B)$
0	0	0	0
0	1	1	1
1	0	1	0
1	1	1	1

Постройте таблицы истинности для следующих логических выражений: $A \& (B \vee B)$. Это задание ученики выполняют самостоятельно, а потом обмениваются тетрадями и выполняется проверка работы.

1) Количество строк: 5;

2) Количество столбцов: 5

A	B	\overline{B}	$B \vee \overline{B}$	$A \& (B \vee \overline{B})$
0	0	1	1	0
0	1	0	1	0
1	0	1	1	1
1	1	0	1	1

Постройте таблицы истинности для следующих логических выражений:

$$A \& (A \vee B \vee C).$$

1) Определим количество строк в таблице истинности. Для этого: считаем количество переменных, в нашем случае логическая функция содержит 3 переменные: A, B и C. Количество строк в таблице истинности должно быть равно $2^3 = 8$.

2) Определим количество столбцов. Это количество логических переменных + количество логических операций. В нашем случае количество переменных = 3, а количество

логических операций = 2, т.е. количество столбцов таблицы истинности = 5.

3) Строим таблицу истинности с указанным количеством строк и столбцов, обозначаем столбцы и вносим в таблицу возможные наборы значений исходных логических переменных и заполняем таблицу истинности по столбцам.

A	B	C	$A \vee B \vee C$	$A \& (A \vee B \vee C)$
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	1	0
1	0	0	1	1
1	0	1	1	1
1	1	0	1	1
1	1	1	1	1

Постройте таблицы истинности для следующих логических выражений:

$$A \vee B \vee C$$

1) Определим количество строк в таблице истинности. Для этого: считаем количество переменных, в нашем случае логическая функция содержит 3 переменные: A, B и C. Количество строк в таблице истинности должно быть равно $2^3 = 8$.

2) Определим количество столбцов. Это количество логических переменных + количество логических операций. В нашем случае количество переменных = 3, а количество логических операций = 3, т.е. количество столбцов таблицы истинности = 6.

3) Строим таблицу истинности с указанным количеством строк и столбцов, обозначаем столбцы и вносим в таблицу возможные наборы значений исходных логических переменных и заполняем таблицу истинности по столбцам.

A	B	C	\overline{C}	$A \vee B \vee \overline{C}$	$\overline{A \vee B \vee \overline{C}}$
0	0	0	1	1	0
0	0	1	0	0	1
0	1	0	1	1	0
0	1	1	0	1	0
1	0	1	0	1	0
1	1	0	1	1	0
1	1	1	0	1	0

Практическое занятие № 14 «Решение задач алгебры логики»

Цель занятия: научиться решать задачи алгебры логики.

Содержание работы:

1. Упростить логическое выражение.

$$F = (A \vee B) \rightarrow (\overline{B \vee C})$$

Решение (используются законы де Моргана, закон двойного отрицания, распределительный закон):

$$F = (A \vee B) \rightarrow (\overline{B \vee C}) = A \vee B \& (\overline{B \vee C}) = (A \vee B) \& (\overline{B} \& \overline{C}) = (A \vee B) \& (\overline{B} \& \overline{C}) = \overline{B} \vee (A \& \overline{C})$$

2. Проверить правильность упрощения построением таблиц истинности.

Для проверки правильности упрощения мы строим таблицы истинности для исходного и упрощенного логического выражения. Если данные в последних столбцах таблиц истинности совпадают, значит мы правильно упростили логическое выражение.

Решение:

Таблица истинности для исходного логического выражения

A	B	C	$A \vee B$	$B \vee C$	$\overline{(B \vee C)}$	$(A \vee B) \rightarrow \overline{(B \vee C)}$	F
0	0	0	0	0	1	1	0
0	0	1	0	1	0	1	0
0	1	0	1	1	0	0	1
0	1	1	1	1	0	0	1
1	0	0	1	0	1	1	0
1	0	1	1	1	0	0	1
1	1	0	1	1	0	0	1
1	1	1	1	1	0	0	1

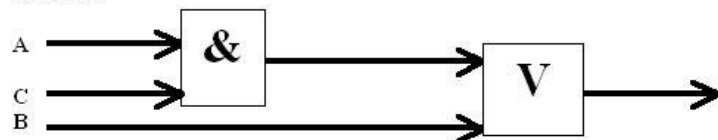
Таблица истинности для упрощенного логического выражения

A	B	C	$A \& C$	$B \vee A \& C$
0	0	0	0	0
0	0	1	0	0
0	1	0	0	1
0	1	1	0	1
1	0	0	0	0
1	0	1	1	1
1	1	0	0	1
1	1	1	1	1

Из таблиц истинности видно, что упрощение верное.

3. Построить логическую схему полученного упрощенного логического выражения.

Решение:



(Продемонстрировать и объяснить работу схемы).

4. Записать следующее высказывание в виде логического выражения: «Если я хорошо подготовлюсь по русскому языку, математике и физике, то я получу пятерки или четверки».

Решение: выделим в составном высказывании простые и обозначим их логическими переменными:

A – хорошо подготовлюсь по русскому языку;

B – хорошо подготовлюсь по математике;

C – хорошо подготовлюсь по физике;

D – получу пятерки;

E – получу четверки.

Тогда составное высказывание будет записано следующим образом:

$$F = (A \& B \& C) \rightarrow (D \vee E)$$

Практическое занятие № 15 «Основы логики и логические основы компьютера»

Цель занятия: работа с основами логики.

Содержание работы:

Вариант 1.

1. Дайте определение логического умножения (конъюнкции). Постройте таблицу истинности.
2. Выделите простые высказывания из следующего сложного высказывания: “Я поеду в Москву, и если встречу там друзей, то мы интересно проведем время.” Запишите сложное высказывание в виде логической формулы.
3. Чему равно значение логического выражения $(A \vee B) \rightarrow (\neg C)$, если $A=0$, $B=0$, $C=1$?

Вариант 2.

1. Дайте определение логического сложения (дизъюнкции). Постройте таблицу истинности.
2. Выделите простые высказывания из следующего сложного высказывания: “Если число делится на 4, то оно делится на 2 и является четным”. Запишите сложное высказывание в виде логической формулы.
3. Чему равно значение логического выражения $\neg(A \wedge B) \rightarrow C$, если $A=1$, $B=0$, $C=1$?

Вариант 3.

1. Дайте определение логического отрицания (инверсии). Постройте таблицу истинности.
2. Выделите простые высказывания из следующего сложного высказывания: “Число является простым, если оно делится на единицу и само себя”. Запишите сложное высказывание в виде логической формулы.
3. Чему равно значение логического выражения $(A \vee B) \wedge (A \vee C)$, если $A=1$, $B=0$, $C=0$?

Вариант 4.

1. Дайте определение логического следования (импликации). Постройте таблицу истинности.
2. Выделите простые высказывания из следующего сложного высказывания: “Спортсмен подлежит дисквалификации, если он некорректно ведет себя по отношению к сопернику или принимал допинг”. Запишите сложное высказывание в виде логической формулы.
3. Чему равно значение логического выражения $\neg A \wedge (B \vee C)$, если $A=1$, $B=1$, $C=0$?

Практическое занятие № 16 «Построение МДНФ и схемы логического устройства по таблицам истинности».

Цель занятия: научиться правилам построения МДНФ и схем логического устройства по таблицам истинности

Содержание работы:

Задание: получить МДНФ логической функции вида $f(x_1, x_2, x_3, x_4) = 0, 1, 5, 8, 9, 13, 14, 15$ и построить схему КУ в базисах:

- И, ИЛИ, НЕ;
- И-НЕ;
- ИЛИ-НЕ.

Составим таблицу истинности

x_1	x_2	x_3	x_4	Y
0	0	0	0	1
0	0	0	1	1
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

Получение МДНФ:

Составляем по таблице истинности **карту Вейча-Карно** и заполняем ее значениями:

$x_1 x_2 \{ x_3 x_4$	00	01	11	10
00	1	1	0	0
01	0	1	0	0
11	0	1	1	1
10	1	1	0	0

1. Для получения **МДНФ** объединяем ячейки, содержащие **1**, количеством не менее 2^n .

Получаем:

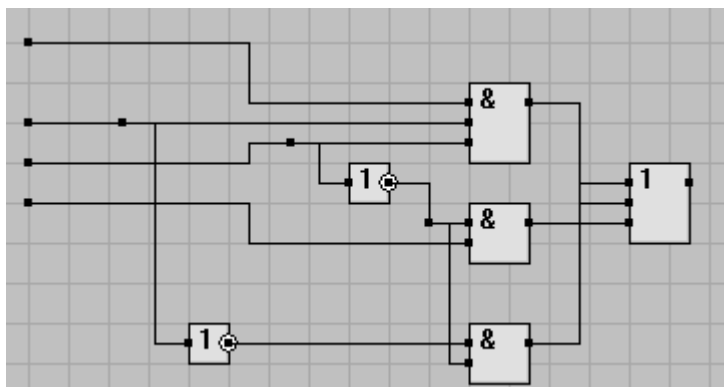
$$f = \overline{x_2} \overline{x_3} \square \overline{x_3} x_4 \square x_1 x_2 x_3$$

Получение логических схем:

Пояснение: сигналы нумеруются сверху вниз, от x_1 до x_4

1. В базисах И, ИЛИ, НЕ:

Структурная схема:

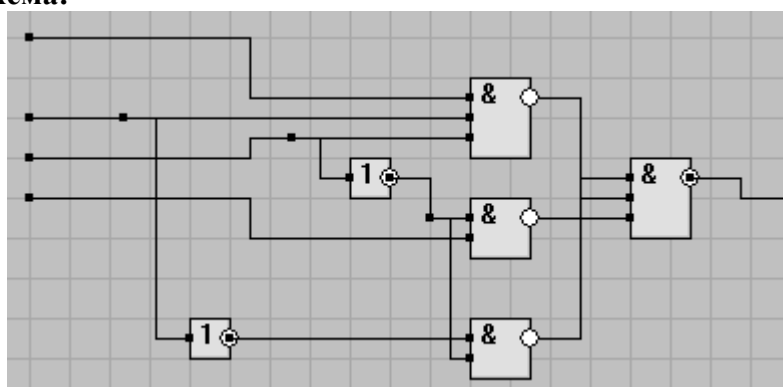


2. В базисах И-НЕ:

Для получения МДНФ в базисе И-НЕ используем **законы Де Моргана** и **закон двойного отрицания**:

$$f = \overline{\overline{x_2 x_3} \vee \overline{x_3 x_4} \vee \overline{x_1 x_2 x_3}} = \overline{\overline{x_2 x_3} \cap \overline{x_3 x_4} \cap \overline{x_1 x_2 x_3}}$$

Структурная схема:



Практическое занятие № 17 «Построение МКНФ и схемы логического устройства по таблицам истинности».

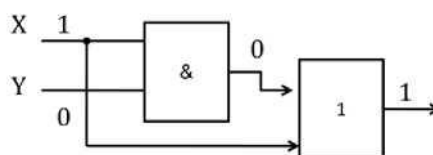
Цель занятия: научиться правилам построения МКНФ и схем логического устройства по таблицам истинности

Содержание работы:

Пример 1. Пусть X = истина, Y = ложь. Составить логическую схему для следующего логического выражения: $F = X \vee Y \& X$.

Две переменные — X и Y
Две логические операции: \vee и $\&$
 $X \vee Y \& X$.

Строим схему:



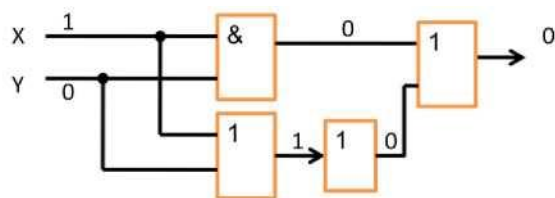
Пример 2. Постройте логическую схему, соответствующую логическому выражению $F = X \& Y \vee \neg(Y \vee X)$. Вычислить значения выражения для X = 1, Y = 0.

1) Переменных две: X и Y;

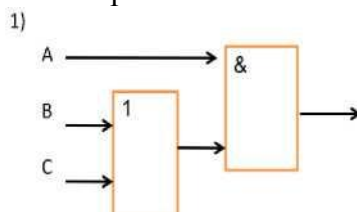
2) Логических операций четыре:

$$F = X \& Y \vee \neg(Y \vee X)$$

Схему строим слева направо в соответствии с порядком логических операций:



Пример 3. Составьте логическое выражения по схеме:



Начинаем составление выражения с последнего элемента - конъюнктора: $F = A \& (B \vee C)$

Разберите упражнения:

1. Постройте логическую схему, соответствующую логическому выражению, и найдите значение логического выражения: $F = A \vee B \& \neg C$, если $A=1, B=1, C=1$
2. Постройте логическую схему, соответствующую логическому выражению, и найдите значение логического выражения: $F = \neg(A \vee B \& C)$, если $A=0, B=1, C=1$.
- 3.

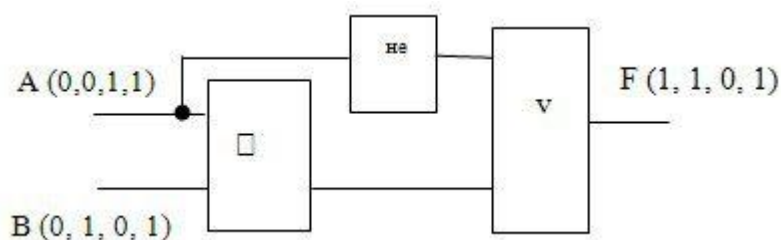
Практическое занятие №18 «Построение структурных схем логических элементов компьютера и их таблиц истинности»

Цель занятия: закрепить умения по составлению схем с использованием логических элементов ЭВМ.

Содержание занятия:

Вариант 1.

1. По заданной логической функции $F(A, B) = (A \wedge B) \vee \bar{A}$ построить логическую схему и таблицу истинности.
- Решение:

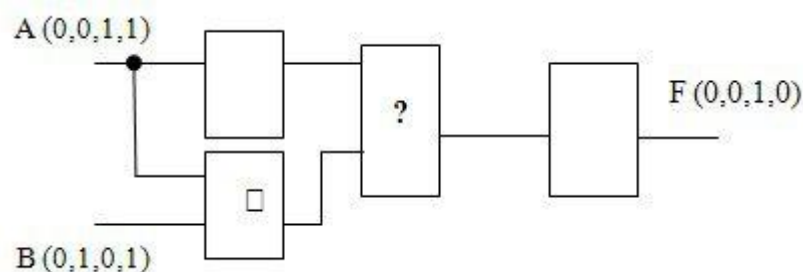


A	B	$A \wedge B$	\bar{A}	$(A \wedge B) \vee \bar{A}$
0	0	0	1	1
0	1	0	1	1
1	0	0	0	0
1	1	1	0	1

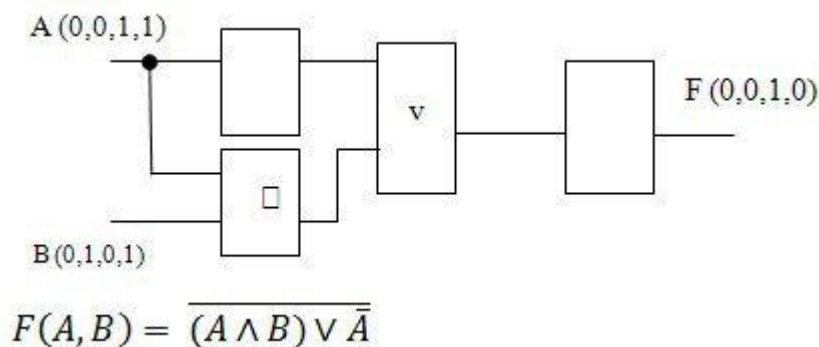
A	B	$A \vee B$	$\overline{A \vee B}$	$(\overline{A \vee B}) \vee B$
0	0	0	1	1
0	1	1	0	1
1	0	1	0	0
1	1	1	0	1

2. Выписать из логической схемы соответствующую ей логическую формулу:

A	B	F
0	0	0
0	1	0
1	0	1
1	1	0



Решение:



Практическое занятие №19 «Получение информации о параметрах компьютерной системы»

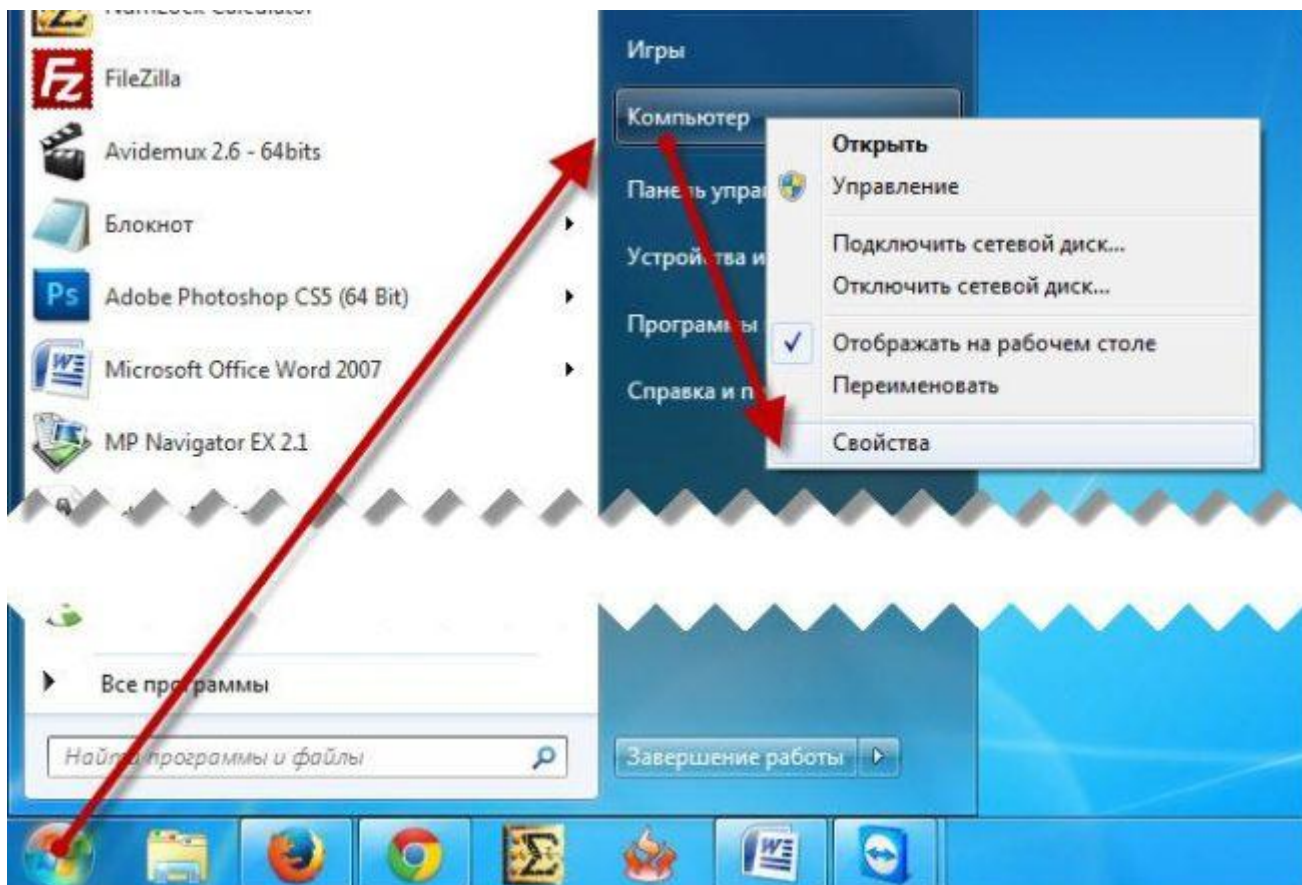
Цель занятия: отработать навык получения информации о параметрах компьютерной системы.

Содержание работы:

1 способ. Через «Панель управления»

Простейший способ получить информацию о свойствах и производительности Windows — с помощью Панели управления:

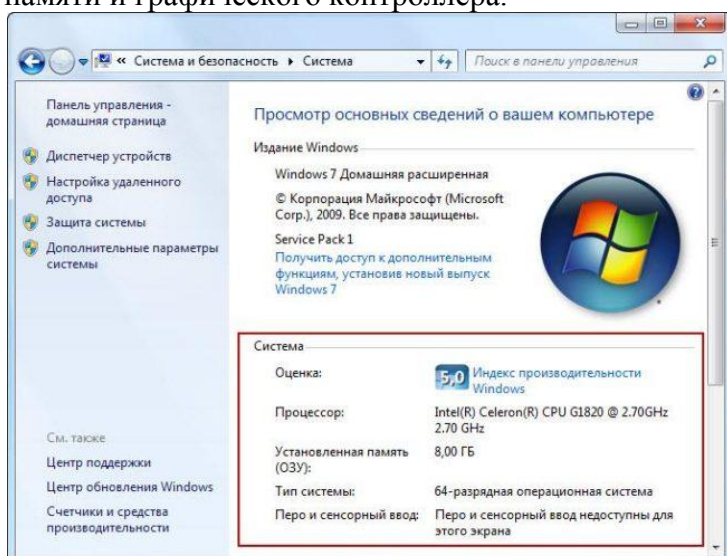
1. Откройте «Пуск» — кнопка слева внизу экрана.
2. Найдите пункт «Компьютер» и кликните по нему правой кнопкой.
3. Выберите «Свойства».



Нажмите правой кнопкой мыши на пункт «Компьютер»

Вызвать окно «Система» можно, нажав сочетание клавиш Win+Pause.

Откроется окно «Система» с информацией о версии Windows, процессоре и оперативной памяти ПК. Также вы увидите индекс производительности от 1 до 10 — чем больше значение, тем лучше работает компьютер. Нажмите на кнопку индекса, чтобы получить детальную информацию о производительности жёсткого диска, процессора, оперативной памяти и графического контроллера.

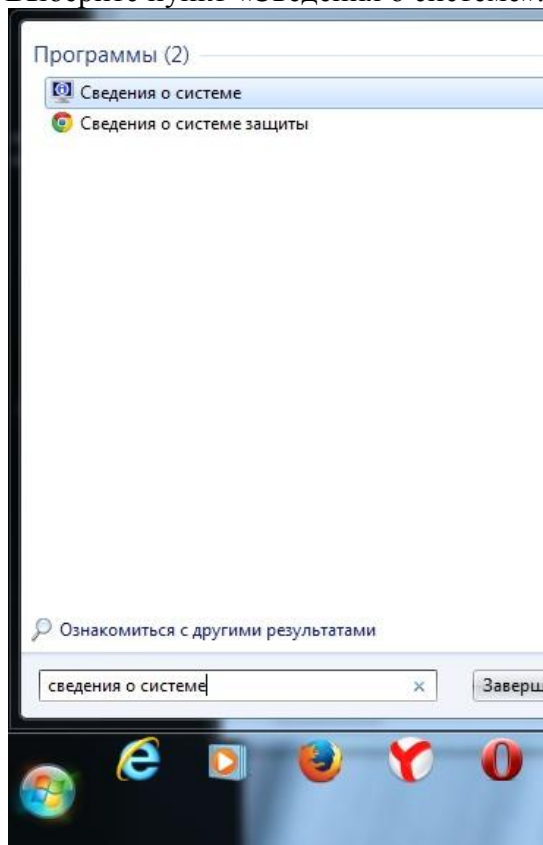


Информация о производительности компьютера

2 Способ. С помощью утилиты «Сведения о системе»

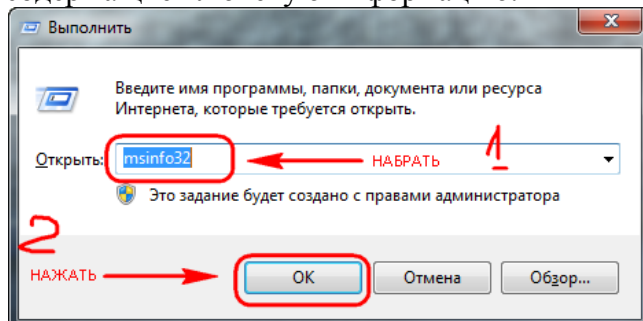
В окне служебной программы Windows «Сведения о системе» вы найдёте информацию о процессоре, версии BIOS и сборке Windows, проверите работу аппаратных ресурсов ПК — CD-ROM, системной платы, видеокарты и других, а также посмотрите наличие ошибок в действии программ и драйверов. Чтобы открыть сведения о системе:

1. Откройте «Пуск».
2. В поисковой строке начните набирать запрос «Сведения о...».
3. Выберите пункт «Сведения о системе».



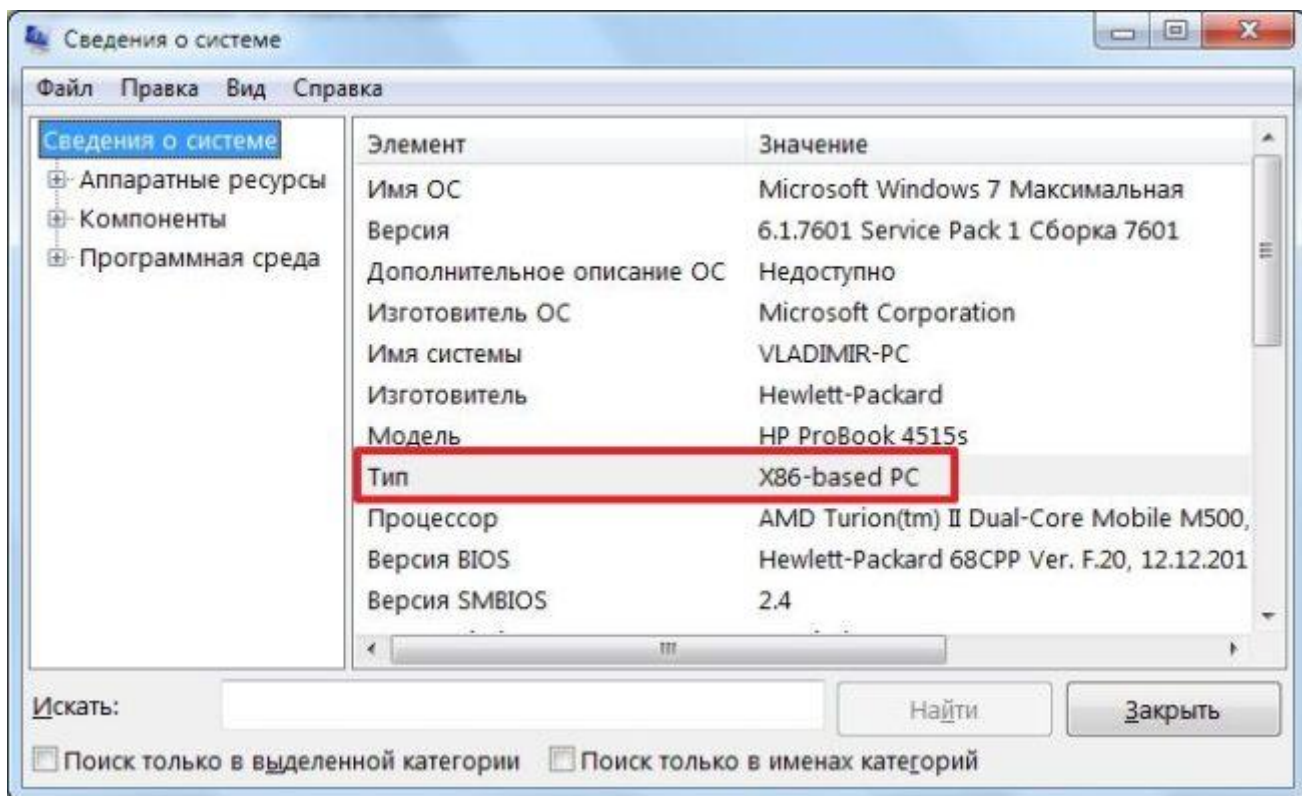
Поиск через поисковую строку меню

4. Вы увидите три категории: «Программная среда», «Компоненты» и «Аппаратные ресурсы». Каждая категория для удобства пользователя разбита на подпункты, содержащие ключевую информацию.



Переход в сведения о системе через команду msinfo32

Меню «Сведения о системе» можно открыть с помощью команды «Выполнить». Для этого нажмите на клавиатуре сочетание Win+R, введите «msinfo32» без кавычек и нажмите Enter.

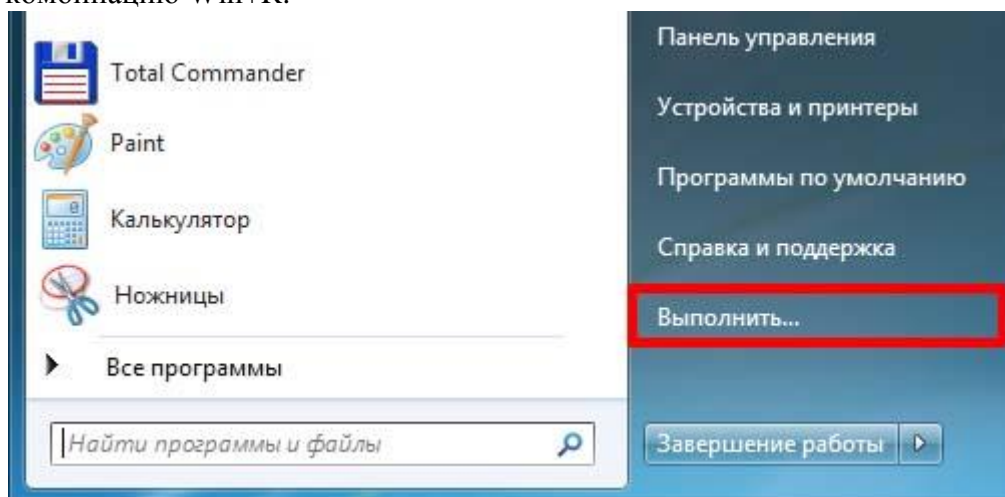


Данные разделены по категориям

3 Способ. Через «Средство диагностики DirectX»

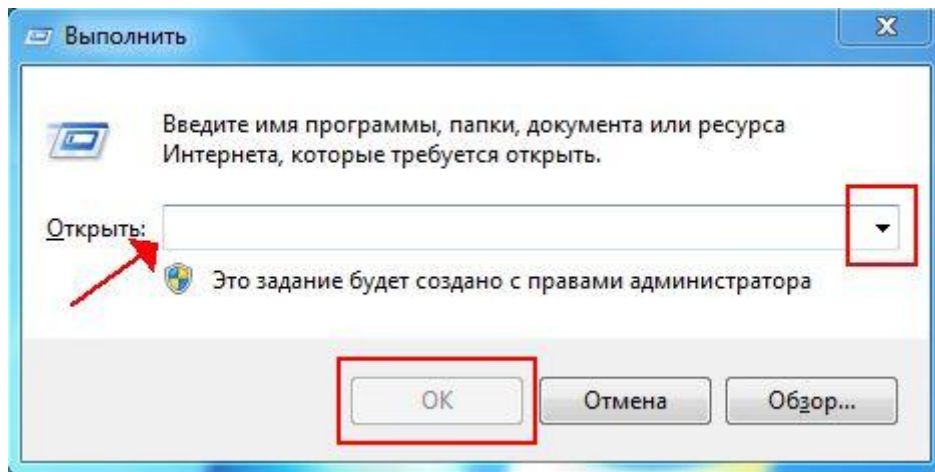
Средство диагностики DirectX предназначено для информирования пользователей об установленных драйверах и проверки их работоспособности. Чтобы открыть диагностику DirectX:

1. Нажмите «Пуск» и выберите пункт «Выполнить». Либо нажмите на клавиатуре комбинацию Win+R.



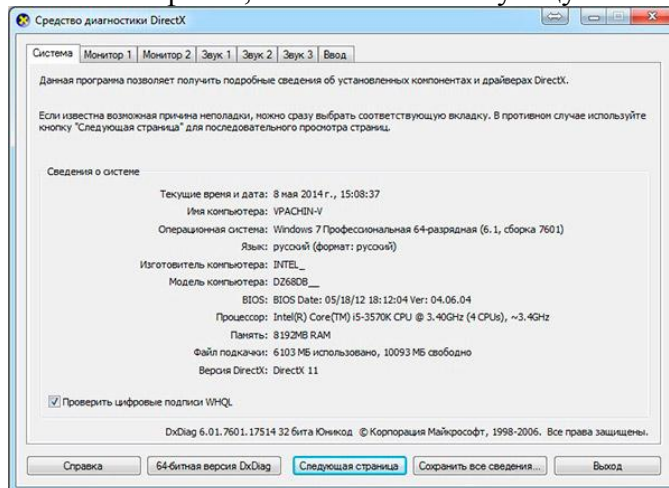
Нажмите, чтобы открыть окно запуска программ

2. В открывшемся окошке введите «dxdiag» без кавычек.
3. Нажмите «ОК».



Введите команду dxdiag

4. Подождите, пока средство диагностики обновит информацию.
5. Вверху окна вы увидите вкладки «Система», «Экран», «Звук», «Ввод». Откройте каждую и отыщите необходимую информацию. Сведения можно сохранить в виде текстового файла, нажав соответствующую кнопку внизу окна.

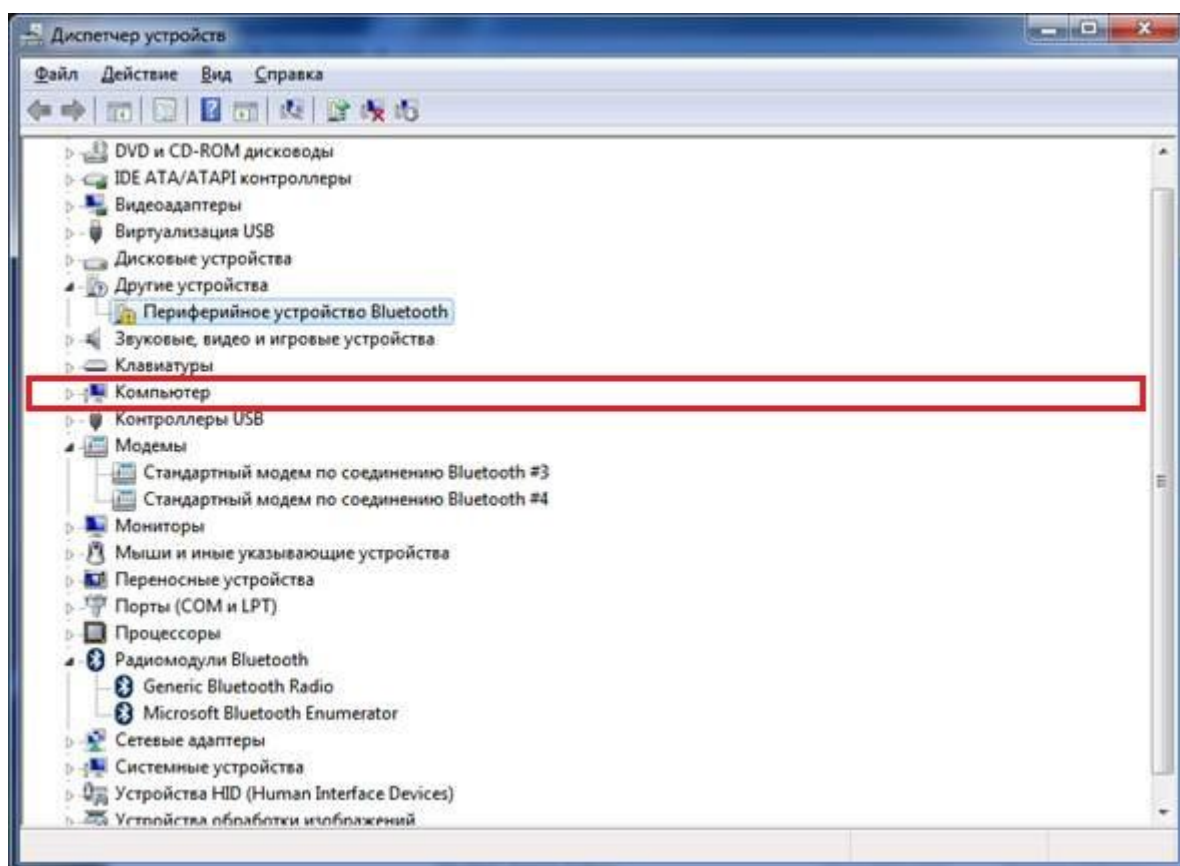


Число вкладок зависит от количества подключенных мониторов и звуковых устройств

4 Способ. С помощью диспетчера устройств

Продвинутым пользователям будет полезно заглянуть в «Диспетчер устройств», где отображается информация обо всех комплектующих к ПК и драйверах для них. Для входа в Диспетчер:

1. Откройте «Пуск».
2. Выберите «Панель управления».
3. Найдите и откройте «Диспетчер устройств».



Древовидная структура позволяет легче ориентироваться в категориях. Просматривая информацию об аппаратных ресурсах, обращайте внимание на поле «Состояние устройства» — там вы увидите, есть ли неполадки в работе.

Практическое занятие №20 «Подключение устройств компьютера»

Цель занятия: изучение основных видов периферийного оборудования, способов их подключения, основных характеристик. Определение по внешнему виду типов разъемов и подключаемого к ним оборудования.

Содержание работы:

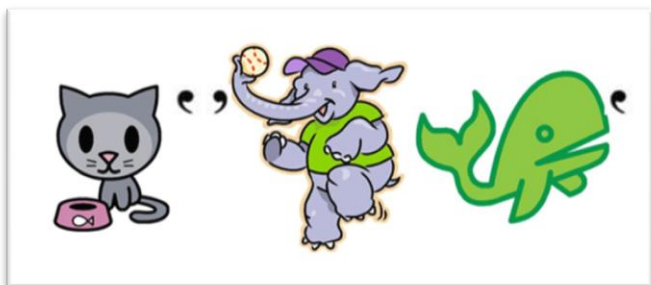
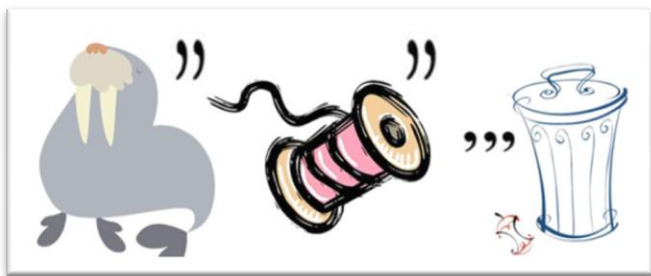
Задание № 1. Определите количество и типы разъёмов вашего персонального компьютера. Данные занесите в таблицу

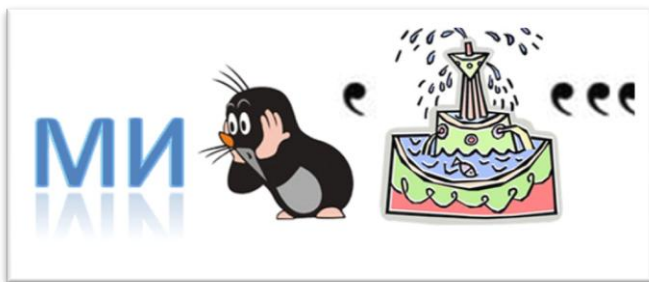
Разъем (изображение)	Тип разъема (название)	Количество в ПК (шт)	Для подключения каких устройств используется	Характеристики разъема
....				

Задание № 2. Установите соответствие между устройствами и их назначением

Клавиатура	Ввод информации
Монитор	Вывод информации
Принтер	
Акустические колонки	
Сканер	
Дигитайзер	
Плоттер	

Задание №3. Разгадайте ребусы. Дайте определения разгаданным понятиям.





Задание №4. Сделать вывод о проделанной лабораторной работе:

Практическое занятие № 21 «Построение автомата намят-триггер»

Содержание работы:

Регистр – внутреннее запоминающее устройство процессора или внешнего устройства, предназначенное для временного хранения обрабатываемой или управляющей информации. Регистры представляют собой совокупность триггеров, количество которых равняется разрядности регистра, и вспомогательных схем, обеспечивающих выполнение некоторых элементарных операций.

Схемы с памятью

Более сложным преобразователем информации являются схемы с памятью. Наличие памяти в схеме позволяет запоминать промежуточные состояния обработки и учитывать их значения в дальнейших преобразованиях. Выходные сигналы $Y = (y_1, y_2, \dots, y_m)$ в схемах данного типа формируются не только по совокупности входных сигналов $X = (x_1, x_2, \dots, x_n)$, но и по совокупности состояний схем памяти $Q = (q_1, q_2, \dots, q_k)$. При этом различают текущий дискретный момент времени t и последующий $(t+1)$ момент.

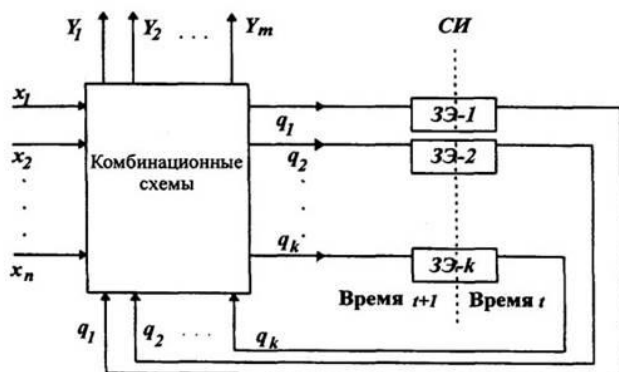


Рис. Обобщенная структура схемы с памятью

Передача значения Q между моментами времени t и $(t+1)$ осуществляется обычно с применением двухступенчатой памяти и синхронизирующих импульсов (СИ).

В качестве простейшего запоминающего элемента (ЗЭ) в современных ЭВМ используют триггеры. В связи с успешным применением микроэлектроники в схемах основных устройств ЭВМ (процессоров и оперативной памяти) исчезли в качестве запоминающихся элементов схемы, использующие остаточную намагниченность – ферритовые сердечники. Самая простейшая схема триггера может быть синтезирована по общим правилам.

Пример. Построить автомат намят - триггер, имеющий вход R (Reset - сброс), Для установки элемента в "нулевое состояние" и вход S (Set - установка) - для установки элемента в "единичное" состояние. При отсутствии сигналов $R=S=0$ элемент должен сохранять свое состояние до тех пор, пока не будут получены новые сигналы на входе К или 8.

Условия работы триггера могут быть представлены в виде таблицы переходов (табл.), представляющей собой модификацию таблицы истинности.

Таблица
Условия работы триггера

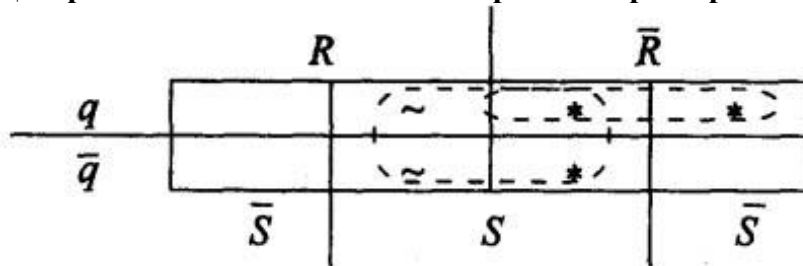
Входы		Состояние q_{t+1}		
R	S	0	1	Режим
0	0	0	1	Хранение
1	0	0	0	Установка 0
0	1	1	1	Установка 1
1	1	?	?	Запрещенное состояние

Содержание таблицы расшифровывается следующим образом. Элемент памяти может сохранять значение $q_t=0$ или $q_t=1$ в зависимости от установки ранее установленного состояния. При отсутствии входных сигналов на входах R и S ($R=0$ и $S=0$) значения q_{t+1} первой строки таблицы в точности повторяют значения q_t . При поступлении сигнала $R=1$ (сигнала установки "нуля") элемент независимо от своего состояния принимает значение, равное нулю, $q_{t+1}=0$. Если же на вход S поступает сигнал установки "единицы" ($S=1$), то $q_{t+1}=1$ независимо от предыдущего состояния q_t . Одновременное поступление сигналов на входы R и S является запрещенной ситуацией, так как она может привести к непредсказуемому состоянию. В схемах формирования сигналов R и S должны быть предусмотрены блокировки, исключающие их совпадения, $S=R=1$.

Для таблицы переходов (табл.) может быть построена диаграмма Вейча.

Таблица

Диаграмма Вейча для таблицы переходов триггера



В этой таблице знаком "~" отмечены запрещенные комбинации входных сигналов. Эти комбинации могут быть использованы для упрощения логических зависимостей. Логическая зависимость, описывающая работу элемента памяти, принимает вид:

$$q_{t+1} = \bar{R}_t \bar{S}_t q_t \vee \bar{R}_t S_t \bar{q}_t \vee \bar{R}_t S_t q_t = \bar{R}_t S_t \vee q_t \bar{R}_t.$$

Уравнение получено путем эквивалентных преобразований. Добавление в него комбинаций, соответствующих запрещенным ситуациям и помеченных знаком "~", т.е.

$$R_t S_t q_t \vee R_t S_t \bar{q}_t = R_t S_t,$$

позволяет еще больше упростить уравнение триггера:

$$q_{t+1} = \bar{R}_t S_t \vee q_t \bar{R}_t \vee R_t S_t = S_t (\bar{R}_t \vee R_t) \vee q_t \bar{R}_t = S_t \vee q_t \cdot \bar{R}_t.$$

Для реализации полученной зависимости в базисе И - НЕ применим правило де Моргана и получим функцию

$$q_{t+1} = \overline{S_t \vee q_t \bar{R}_t} = \bar{S}_t \cdot \overline{q_t \bar{R}_t}.$$

По данной зависимости можно построить схему элемента памяти - асинхронного RS-триггера. В этой схеме следует только соединить выход q_{t+1} со входом q_t . На рис. эта связь отмечена штриховой линией.

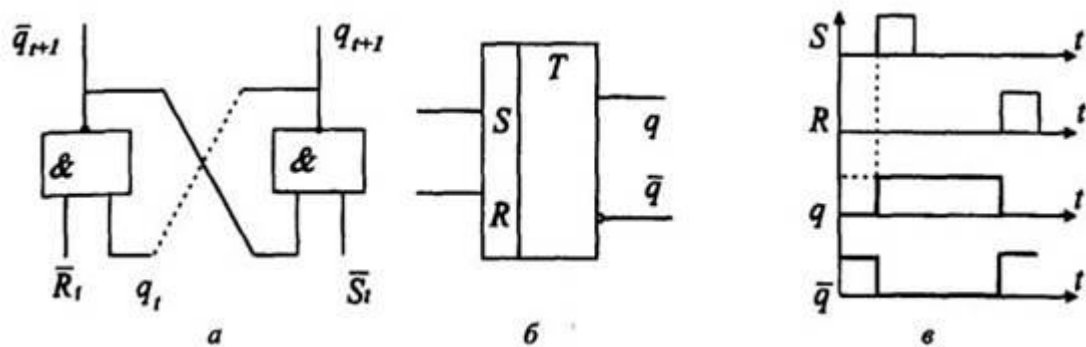


Рис. Схема асинхронного RS-триггера: а- схема; б - обозначение на принципиальных электрических схемах; в - временная диаграмма

RS-триггер нашел широкое распространение в схемах ЭВМ. Одиночные триггеры этого типа часто используются в различных блоках управления. В асинхронных RS-триггерах имеется один существенный недостаток, обусловленный самой логикой их построения (см. табл. 3.5), т.е. в них сигналы R и S должны быть разнесены во времени. Дополнение этого триггера комбинационными схемами синхронизации на входе и выходе позволяет получить триггеры с более сложной логикой работы: синхронные RS-триггеры, Т-, JK-, D- триггеры и целый ряд комбинированных RST-, JKRS-, DRS-трштеров [4].

Прописные буквы в названиях триггеров обозначают:

- R (Reset - сброс) - вход установки триггера в нулевое состояние $Q=0$;
- S (Set - установка) - вход установки триггера в единичное состояние $Q=1$;
- T (Toggle - релаксатор) - счетный вход триггера;
- J (Jerk - внезапное включение) - вход установки ЛС-триггера в единичное состояние $Q=1$;
- K (Kill - внезапное выключение) - $Q=0$;
- D (Delay - задержка) - вход установки триггера в единичное или нулевое состояние на время, равное одному такту;
- C (Clock - часы) - вход синхронизирующих тактовых импульсов. На рис. показаны схемы синхронного одноконтного (а) и двухконтного (б) RS-триггеров.

Двухкаскадная схема RS-триггера нашла наиболее широкое применение для построения n -разрядных схем запоминания - всевозможных регистровых схем. Штриховыми линиями на схеме указаны дополнительные точки подключения сигналов установки и сброса.

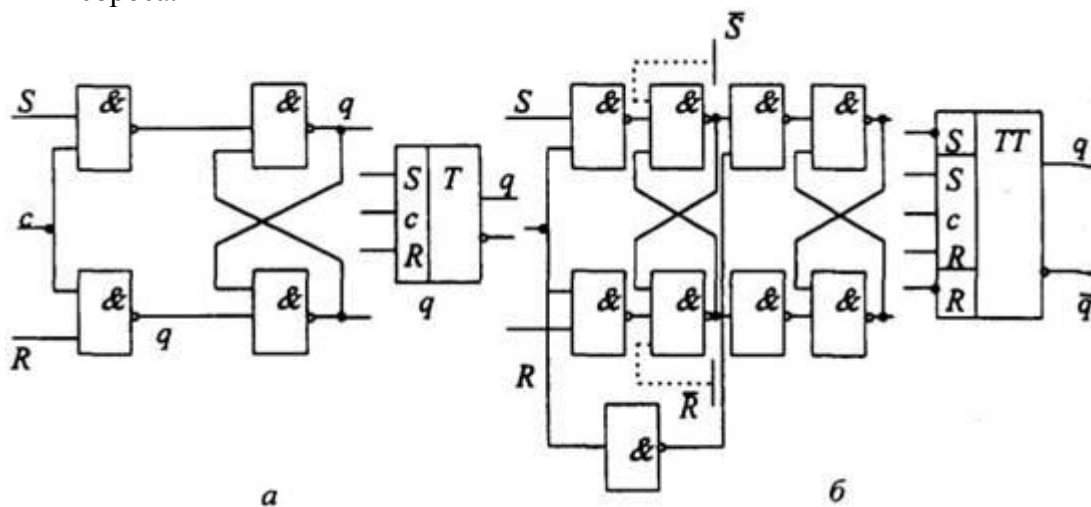


Рис. Электрическая и функциональная схемы синхронных RS-триггеров: а, б - варианты

На рис. приведена схема Т-триггера или иначе - триггера со счетным входом. При значении $T=0$ триггер сохраняет свое ранее установленное состояние - режим хранения

состояния, при $T=1$ триггер переходит в противоположное состояние. Таблица переходов (табл.) и диаграмма работы (рис.) отражают динамику работы этого элемента.

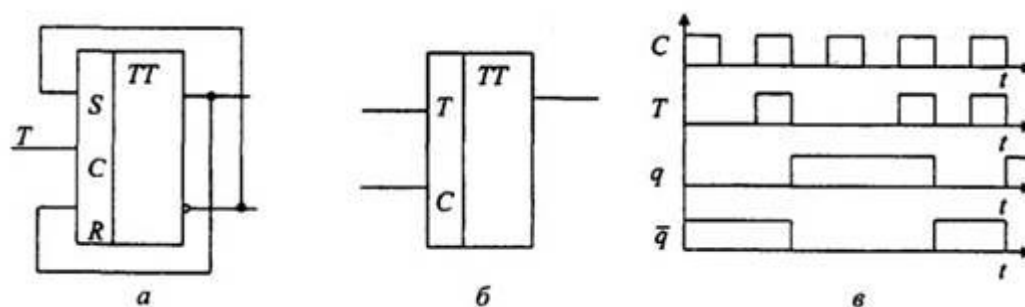


Рис. Схема триггера со счетным входом: а - функциональная; б - условное обозначение; в - временная диаграмма

Таблица переходов Т-триггера

Входные сигналы	Состояние q_t		Режим
	0	1	
X_t	0	1	
0	0	1	Хранение
1	1	0	Инверсия

По таблице переходов можно получить логическую функцию, реализуемую Т-триггером:

$q_{t+1} = \bar{q}_t \cdot X_t \vee q_t \cdot \bar{X}_t = q_t \oplus X_t$. Нетрудно видеть, что зависимость очень похожа на функцию, выведенную для одноразрядного комбинационного полусумматора. На рис., а показано, как двухтактный RS-триггер преобразуется в Т-триггер.

Наиболее сложным типом триггера является JK-триггер. Он, по существу, является объединением двухтактного RS- и Т-триггеров. Этому соответствует его таблица переходов (табл.).

Таблица переходов JK-триггера

Входные сигналы		Состояние q		Реж им
J	K	0	1	
0	1	0	1	Хра нение
0	0	1	0	
0	1	0	1	Уста новка 0
1	1	0	0	
				Уста новка 1
				Инв ерсия

Если первые три строки таблицы переходов полностью повторяют соответствующие строки табл., то последняя строка, с запрещенной комбинацией для RS-триггера, соответствует режиму переключения Т-триггера. Схема JK-триггера изображена на рис.

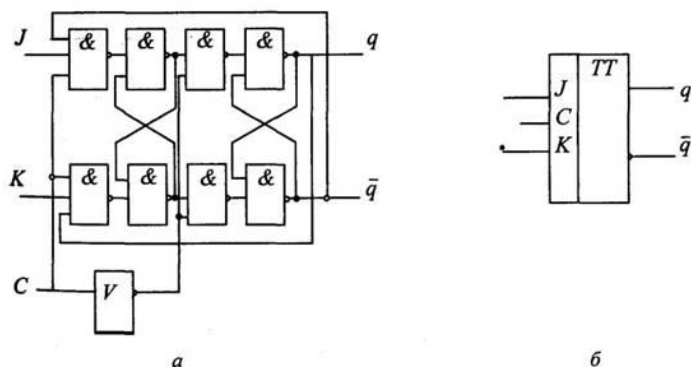


Рис. JK-триггер: *а* - функциональная схема; *б* - условное обозначение

D-триггер обычно строится на основе двухтактного RS- или JK-триггера. Он предназначен для хранения состояния (1 или 0) на один период тактовых импульсов (с задержкой на 1 такт). Таблица его переходов отражена в табл. На рис. представлены варианты его построения, а на рис., *в* - его условное обозначение.

Таблица переходов D-триггера

Входные сигналы	Состояния q		Режим
	0	1	
D			
0	0	0	Установка 0
1	1	1	Установка 1

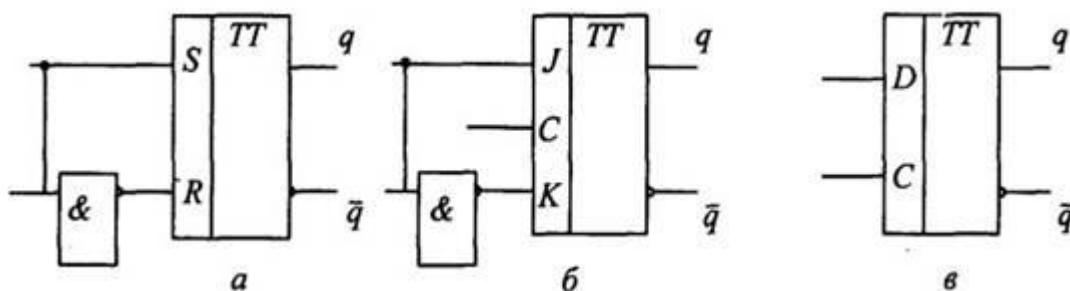


Рис. D-триггер: *а*- функциональная схема на основе RS-триггера; *б*-функциональная схема на основе JK-триггера; *в* - условное обозначение

Все перечисленные элементы памяти позволяют хранить одну единицу информации - бит или одну двоичную цифру.

При построении ЭВМ широко используются функциональные схемы, обеспечивающие операции хранения и преобразования информации над группами битов (машинными словами). Такие сложные схемы называются узлами. К типовым узлам относят: регистры, счетчики, сумматоры. Все они также принадлежат к регулярным структурам, состоящим из одинаковых параллельно работающих одноразрядных схем.

Регистром называется узел, предназначенный для приема, временного хранения и выдачи машинного слова. Регистры могут также использоваться для некоторых операций преобразования данных: для сдвига кода числа (слова) на определенное число разрядов влево или вправо, для преобразования последовательного кода числа в параллельный и наоборот и т.д. Эти дополнительные функции регистров обеспечиваются путем усложнения схем хранения, выбора более сложных триггеров и подключения дополнительных логических схем на их входах и выходах.

Таким образом, регистры представляют собой совокупность триггеров, число которых соответствует числу разрядов в слове, и вспомогательных схем, обеспечивающих выполнение различных операций над словом.

На рис. показана функциональная схема n -разрядного регистра, построенного на RS-триггерах. Информация в регистр записывается под действием сигнала "Запись". Предварительно перед установкой кода на регистр обычно на все разряды R подается

сигнал сброса. На рисунке показано, что подключение к входам R дополнительных инверторов позволяет избежать этой предварительной операции. Здесь на вход каждого разряда поступает парафазный код двоичной цифры (x_i - на вход S_i и \bar{x}_i - на вход R_i), т.е. прямое и инверсное значения кода подаются в противофазе.

На рис. изображена функциональная схема того же регистра, дополненная логическими элементами для преобразования хранящегося на регистре кода. По сигналу "Прямой код" с регистра считывается прямой код хранящихся данных, а по сигналу "Обратный код" - инверсное значение каждого разряда слова. Если оба эти сигнала поступают одновременно, то считывается парафазный код хранящейся информации. Более сложная логика на входе и выходе запоминающих элементов позволяет строить сдвигающие регистры.

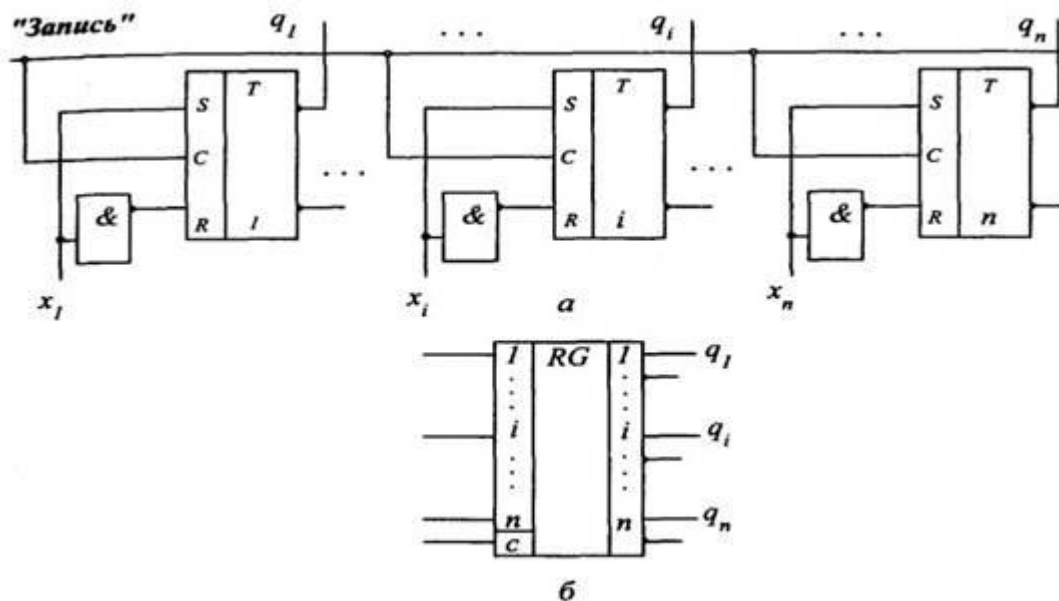


Рис. Схема регистра на RS-триггерах: а - функциональная схема; б - условное обозначение регистра

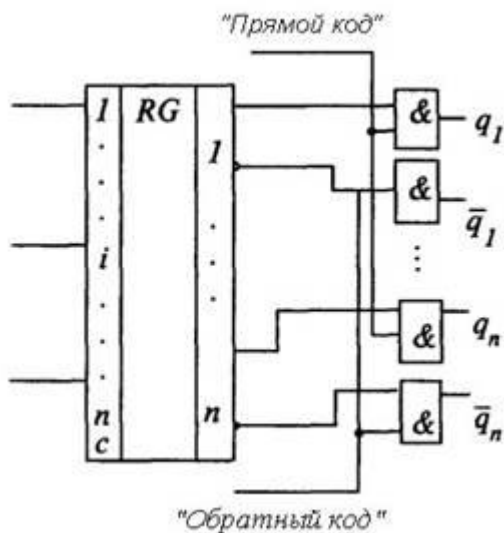


Рис. Схема выдачи информации из регистра

Счетчик - узел ЭВМ, позволяющий осуществлять подсчет поступающих на его вход сигналов и фиксацию результата в виде многоразрядного двоичного числа. Счетчик, состоящий из n -триггеров, дает возможность подсчитывать до N сигналов, связанных зависимостью:

$$n = \log_2 N \text{ или } N = 2^n.$$

В ЭВМ счетчики используются для подсчета импульсов, сдвигов, формирования адресов и т.д. Функционально различают суммирующие, вычитающие, реверсивные счетчики. Они также отличаются друг от друга логикой работы дополнительных логических элементов, подключаемых к триггерам.

В основу построения любого счетчика положено свойство Т-триггеров изменять свое состояние при подаче очередного сигнала на счетный вход Т. На рис. показана схема трех разрядов суммирующего счетчика, построенного на Т-триггерах. Логика его работы представлена в табл.

Таблица переходов трехразрядного счетчика

ход	Состояние								
	00	01	10	11	00	01	10	11	Ре
	0	0	0	0	1				жим
	00	01	10	11	00	01	10	11	Хр
	0	0	0	1	1				анение
	01	10	11	00	01	10	11	00	Сч
									ет

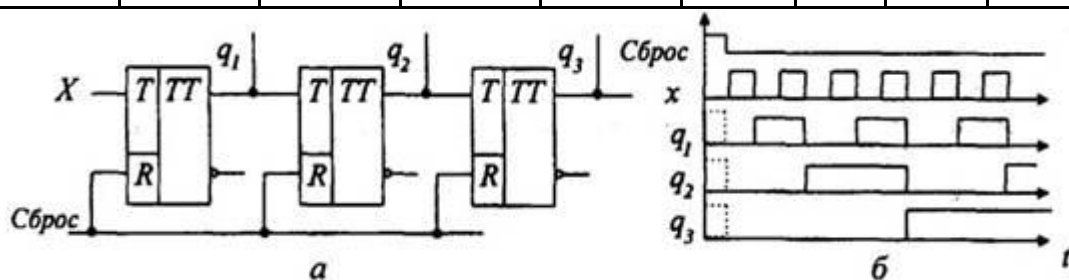


Рис. Организация счетчика на Т-триггерах: *а* - функциональная схема; *б* - временная диаграмма

Сумматор - узел ЭВМ, в котором суммируются коды чисел. Как правило, любой сумматор представляет собой комбинацию одноразрядных сумматоров. Сумматоры различают по принципам построения: накапливающего типа и комбинационного типа. Сумматоры накапливающего типа строят на сложных JKRS-триггерах, дополняя их выходы достаточно сложными схемами формирования и распространения переносов. Процесс сложения при этом осуществляется поэтапно. Сначала на триггерах сумматора фиксируется код первого операнда, затем на счетные коды разрядов подается код второго операнда. По зависимостям на каждом триггере формируются одноразрядные суммы и значения переносов между разрядами. Учет возникающих переносов задерживает формирование окончательного результата суммы и может требовать дополнительных тактов сложения. Из-за этого многоразрядные схемы сумматора накапливающего типа используются достаточно редко.

Более часто для построения сумматоров используются сумматоры комбинационного типа. Логика работы такого сумматора была представлена данными табл. Обычно у такого сумматора на входе и выходе имеются регистры для хранения и преобразования кодов операндов и результата.

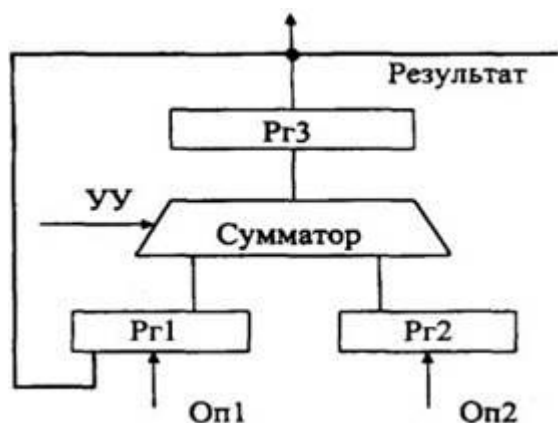


Рис. Упрощенная схема сумматора ЭВМ

Регистр Pr1 предназначен для хранения кода первого операнда, регистр Pr2 - для хранения кода второго операнда. Сумматор по сигналам из устройства управления настраивается на выполнение определенной машинной операции, соответствующей коду операции, находящемуся в коде команды. Результат выполняемой операции фиксируется в регистре Pr3. При необходимости этот результат может использоваться для продолжения вычислений. Для этого предусматривается возможность перезаписи содержимого регистра Pr3 на Pr1 в качестве значения одного из операндов при выполнении очередной операции.

Работа триггерной схемы определяется не таблицей истинности, как для комбинационной логической схемы, а таблицей переходов. Таблица переходов показывает изменение состояния триггера при изменении состояния входных сигналов в зависимости от его текущего состояния.

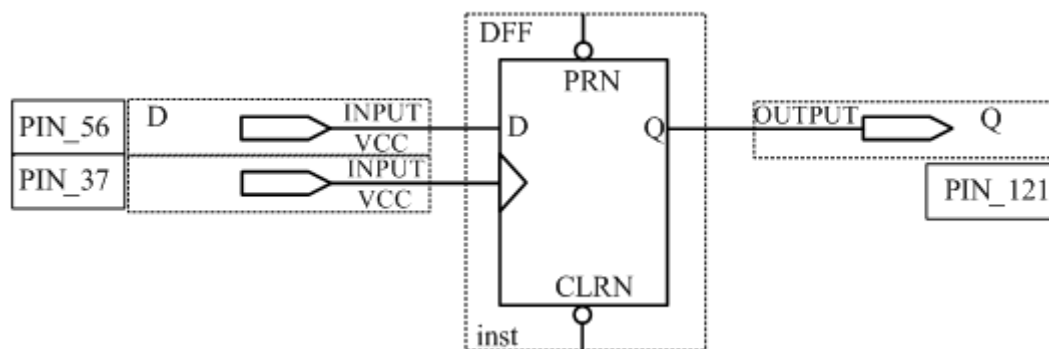
Практическое занятие № 22 «Исследование работы D-триггера и T-триггера»

Цель занятия: Исследование работы D-триггера и T-триггера.

Содержание работы:

1. Исследовать D-триггер

Сконфигурировать ПЛИС в соответствии с рисунком:



Устанавливая с помощью переключателя S8 различные логические уровни на выходе D и затем нажимая на кнопку Button, заполнить таблицу переходов 3.

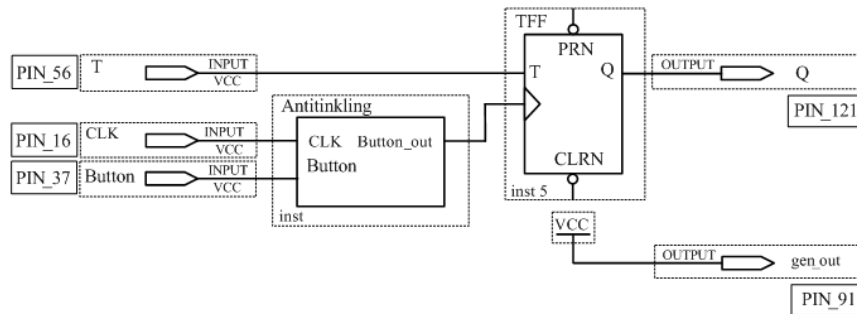
Таблица переходов для синхронного D триггера

D	C	Q(t)	Q(t+1)
x	0	0	
x	0	1	
0	1	0	
0	1	1	

1	1	0	
1	1	1	

2. Исследовать синхронный Т-триггер

Сконфигурировать ПЛИС в соответствии с рисунком:



Установите с помощью переключателя S8 логический уровень "1" на входе Т и затем, нажимая на кнопку Button, заполнить таблицу переходов. Повторите те же действия при логическом "0" на входе Т. Заполните таблицу.

Таблица переходов для синхронного Т триггера:

T	C	Q(t)	Q(t+1)	Q(t+2)	Q(t+3)	Q(t+4)	Q(t+5)
1	1	0					
1	1	1					
0	1	0					
0	1	1					

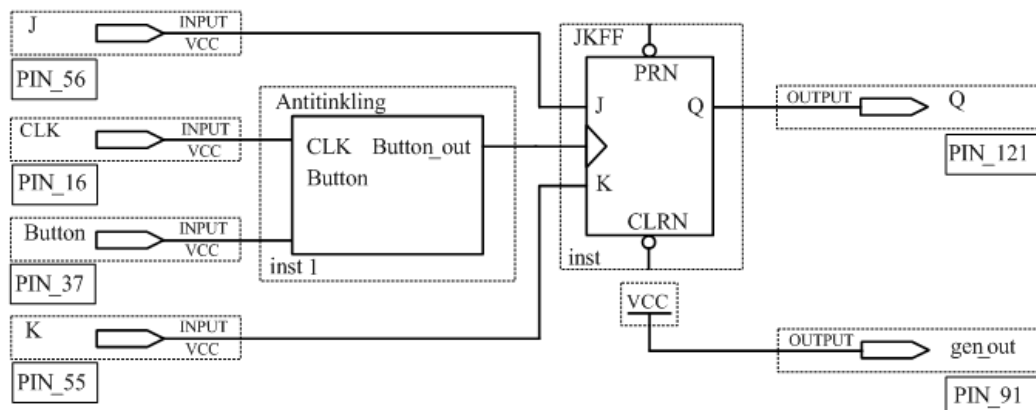
Практическое занятие № 23 «Исследование работы JK -триггера»

Цель занятия: Исследование работы JK–триггера.

Содержание работы:

Исследование синхронного JK триггера

Сконфигурировать ПЛИС в соответствии с рисунком.



Устанавливая с помощью переключателя S7 и S8 различные логические уровни на входах J, K и затем, нажимая на кнопку Button, заполнить таблицу переходов(таблица).

Таблица переходов для JK триггера

J	K	C	Q(t)	Q(t+1)	Название режима работы триггера
0	0	1	0		
0	0	1	1		

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

Практическое занятие № 24 «Исследование архитектуры системной платы».

Цель занятия: ознакомление с составом системной платы, принципами ее работы.

Содержание работы:

1. С помощью программы WinMSD (WinCheckit, SysInfo,)осуществить:
2. определение типа основного микропроцессора;
3. определение тактовой частоты микропроцессора;
4. определение типа BIOS (базовой системы ввода-вывода);
5. определение количества подключенных дисковых устройств (физических и логических);
6. определение размера ОЗУ и кэш-памяти;
7. определение параметров контроллера клавиатуры;
8. определение наличия дополнительных контроллеров и адаптеров (E-IDE, SCSI, SVGA, ETHERNET, MODEM, PCI, и т.д.);
9. определение типа системной и локальной шины и их характеристики.
10. С помощь. Панели Рисование создать рисунок поясняющий работу системной платы компьютера:

Логический схема системной платы



Практическое занятие № 25 «Изучение устройства системного блока ПЭВМ»

Цель занятия – изучение конструктивов корпусов системных блоков ПЭВМ, методов компоновки их элементов и узлов, особенностей работы блоков питания.

Содержание работы:

Задание на подготовку к работе

1. Ознакомиться с основными параметрами выбора корпуса системного блока.
2. Ознакомиться с типами конструкций системных блоков, форм факторами настольных ПЭВМ, способами компоновки системных блоков.

3. Ознакомиться с основными характеристиками блоков питания, используемых в конструкциях системных блоков.

Лабораторное задание

1. Изучить выданный преподавателем корпус системного блока, определить его стандарт, конструктивные особенности.
2. Согласовать с преподавателем варианты компоновки системного блока.
3. Осуществить монтаж тестового оборудования на рабочем месте. Выполнить настройку указанных конфигураций, провести анализ производительности с помощью тестовых утилит, определить пути повышения производительности.
4. Выполнить анализ температурных режимов в указанных преподавателем точках системного блока на основе математического моделирования и экспериментальных исследований.
5. Определить рекомендации по использованию соответствующих конфигураций ПЭВМ для решения прикладных задач.

Практическое занятие № 26 «Исследование работы оперативной памяти компьютера».

Цель работы: Исследовать работу оперативно запоминающего устройства.

Содержание работы:

Оперативные запоминающие устройства (ОЗУ) являются неотъемлемой частью микропроцессорных систем различного назначения. ОЗУ делятся на два класса: статические и динамические. В статических ОЗУ запоминание информации производится на триггерах, а в динамических — на конденсаторах емкостью порядка 0,5 пФ. Длительность хранения информации в статических ОЗУ не ограничена, тогда как в динамических она ограничена временем саморазряда конденсатора, что требует специальных средств регенерации и дополнительных затрат времени на этот процесс. Конструктивно любое ОЗУ состоит из двух блоков — матрицы запоминающих элементов и дешифратора адреса. По технологическим соображениям матрица чаще всего имеет двухкоординатную дешифрацию адреса — по строкам и столбцам. Матрица состоит из 16 ячеек памяти mem_i . Каждая ячейка памяти адресуется по входам X, Y путем выбора дешифраторами адресных линий по строкам $Ax0...Ax3$ и по столбцам $Ay0...Ay3$ и подачи по выбранным линиям сигнала логической единицы. При этом в выбранной ячейке памяти срабатывает двухвходовой элемент И (U1), подготавливая цепи чтения-записи информации на входных $DIO...DI3$ или выходных $DO0...D03$ разрядных шинах. Разрешающим сигналом для выдачи адреса является CS (chip select — выбор кристалла), который подается на вход разрешения счетчика адреса (Addr_cnt) или такой же вход дешифраторов, подключенных к выходам счетчика. При записи в ячейку памяти (см. задание) на соответствующей разрядной шине устанавливается 1 или 0, на входе WR/RD' устанавливается сигнал 1 и после стробирования счетчика или дешифраторов адреса сигналом CS срабатывают элементы 2И U1, U2. Положительный перепад сигнала с элемента U2 поступает на тактовый вход D-триггера U4, в результате чего в нем записывается 1 или 0 в зависимости от уровня сигнала на его D-входе. При чтении из ячейки памяти на входе WR/RD' устанавливается 0, при этом срабатывают элементы U1, U3, U5 и на вход РАЗРЕШЕНИЕ ВЫХОДА буферного элемента U6 поступает разрешающий сигнал, в результате чего сигнал с Q-выхода D-триггера передается на разрядную шину $DO0...D03$. Для проверки функционирования ячейки памяти используется генератор слова. В микропроцессорных системах в качестве ОЗУ чаще всего используются динамические ОЗУ с запоминающим конденсатором, которые отличаются большим многообразием. Приведем данные по наиболее распространенным типам таких ОЗУ. В динамической памяти ячейки выполнены на основе областей с накоплением зарядов, занимающих гораздо меньшую площадь, нежели триггеры, и практически не потребляющих энергии при хранении информации. При записи бита в такую ячейку в ней

формируется электрический заряд, который сохраняется в течение нескольких миллисекунд; для постоянного сохранения заряда ячейки необходимо регенерировать (перезаписывать) ее содержимое. Ячейки микросхем динамической памяти также организованы в виде прямоугольной матрицы; при обращении к микросхеме на ее входы вначале подается адрес строки матрицы, сопровождаемый сигналом RAS (Row Address Strobe — строб адреса строки), затем, через некоторое время — адрес столбца, сопровождаемый сигналом CAS (Column Address Strobe — строб адреса столбца). При каждом обращении к отдельной ячейке регенерируются все ячейки выбранной строки, поэтому для полной регенерации матрицы достаточно перебрать адреса строк. Ячейки динамической памяти имеют сравнительно малое быстродействие (десятки — сотни наносекунд), но большую удельную плотность (порядка нескольких мегабайт на корпус) и меньшее энергопотребление. Обычные ОЗУ называют часто асинхронными, так как установка адреса и подача управляющих сигналов могут выполняться в произвольные моменты времени, необходимо только соблюдение временных соотношений между этими сигналами. В них включены так называемые охранные интервалы, необходимые для установления сигналов. Существуют также синхронные виды памяти, получающие внешний синхросигнал, к импульсам которого жестко привязаны моменты подачи адресов и обмена данными; они позволяют более полно использовать внутреннюю конвейеризацию и блочный доступ. FPM DRAM (Fast Page Mode DRAM — динамическая память с быстрым страничным доступом), активно используется в последнее время. Память со страничным доступом отличается от обычной динамической памяти тем, что после выбора строки матрицы и удержания сигнала RAS допускает многократную установку адреса столбца, стробируемого сигналом CAS, а также быструю регенерацию по схеме "CAS прежде RAS". Первое позволяет ускорить блочные передачи, когда весь блок данных или его часть находятся внутри одной строки матрицы, называемой в этой системе страницей, а второе — снизить затраты времени на регенерацию памяти. EDO (Extended Data Out — расширенное время удержания данных на выходе) фактически представляют собой обычные микросхемы FPM, на выходе которых установлены регистры-защелки данных. При страничном обмене такие микросхемы работают в режиме простого конвейера: удерживают на выходах данных содержимое последней выбранной ячейки, в то время как на их входы уже подается адрес следующей выбираемой ячейки. Это позволяет примерно на 15% по сравнению с FPM ускорить процесс считывания последовательных массивов данных. При случайной адресации такая память ничем не отличается от обычной. BEDO (Burst EDO — EDO с блочным доступом) — память на основе EDO, работающая не одиночными, а пакетными циклами чтения/записи. Современные процессоры благодаря внутреннему и внешнему кэшированию команд и данных обмениваются с основной памятью преимущественно блоками слов максимальной ширины. При наличии памяти BEDO отпадает необходимость постоянной подачи последовательных адресов на входы микросхем с соблюдением необходимых временных задержек, достаточно стробировать переход к очередному слову отдельным сигналом. SDRAM (Synchronous DRAM — синхронная динамическая память) — память с синхронным доступом, работающая быстрее обычной асинхронной (FPM/EDO/BEDO). Кроме синхронного доступа, SDRAM использует внутреннее разделение массива памяти на два независимых банка, что позволяет совмещать выборку из одного банка с установкой адреса в другом. SDRAM также поддерживает блочный обмен. Основное преимущество SDRAM состоит в поддержке последовательного доступа в синхронном режиме, где не требуется дополнительных тактов ожидания. При случайном доступе SDRAM работает практически с той же скоростью, что и FPM/EDO. PB SRAM (Pipelined Burst SRAM — статическая память с блочным конвейерным доступом) — разновидность синхронных SRAM с внутренней конвейеризацией, за счет которой примерно вдвое повышается скорость обмена блоками данных. Кроме основного ОЗУ, устройством памяти снабжается и устройство отображения информации — видеодисплейная система.

Такая память называется видеопамью и располагается на плате видеоадаптера. Видеопамью служит для хранения изображения. От ее объема зависит максимально возможное разрешение видеокарты — $A \times B \times C$, где A — количество точек по горизонтали, B — по вертикали, C — количество возможных цветов каждой точки. Например, для разрешения $640 \times 480 \times 16$ достаточно иметь видеопамью 256 Кбайт, для $800 \times 600 \times 256$ — 512 КБ, для $1024 \times 768 \times 65536$ (другое обозначение — $1024 \times 768 \times 64k$) — 2 Мбайт и т.д. Поскольку для хранения цветов отводится целое число разрядов, количество цветов всегда является целой степенью 2 (16 цветов — 4 разряда, 256 — 8 разрядов, 64k — 16 и т.д.). В видеоадаптерах используются следующие типы видеопамью. FPM DRAM (Fast Page Mode Dynamic RAM — динамическое ОЗУ с быстрым страничным доступом) — основной тип видеопамью, идентичный используемой в системных платах. Активно применялась до 1996 г. Наиболее распространенные микросхемы FPM DRAM — четырехразрядные DIP и SOJ, а также шестнадцатиразрядные SOJ. VRAM (Video RAM — видео-ОЗУ) — так называемая двухпортовая DRAM с поддержкой одновременного доступа со стороны видеопроцессора и центрального процессора компьютера. Позволяет совмещать во времени вывод изображения на экран и его обработку в видеопамью, что сокращает задержки и увеличивает скорость работы. EDO DRAM (Extended Data Out DRAM — динамическое ОЗУ с расширенным временем удержания данных на выходе) — память с элементами конвейеризации, позволяющей несколько ускорить обмен блоками данных с видеопамью. SGRAM (Synchronous Graphics RAM — синхронное графическое ОЗУ) — вариант DRAM с синхронным доступом, когда все управляющие сигналы изменяются одновременно с системным тактовым синхросигналом, что позволяет уменьшить временные задержки. WRAM (Window RAM — оконное ОЗУ) — EDO VRAM, в котором окно, через который обращается видеоконтроллер, сделано меньшим, чем окно для центрального процессора. MDRAM (Multibank DRAM — многобанковое ОЗУ) — вариант DRAM, организованный в виде множества независимых банков объемом по 32 Кбайт каждый, работающих в конвейерном режиме. Увеличение скорости обращения видеопроцессора к видеопамью, кроме повышения пропускной способности адаптера, позволяет повысить максимальную частоту регенерации изображения, что снижает утомляемость глаз оператора. Микросхемы памяти имеют четыре основные характеристики — тип, объем, структуру и время доступа. Тип обозначает статическую или динамическую память, объем показывает общую емкость памяти, а структура — количество ячеек памяти и разрядность каждой ячейки. Например, 28/32-выводные DIP-микросхемы SRAM имеют 8-разрядную структуру ($8k \times 8$, $16k \times 8$, $32k \times 8$, $64k \times 8$, $128k \times 8$), кэш объемом 256 Кбайт состоит из восьми микросхем $32k \times 8$ или четырех микросхем $64k \times 8$ (речь идет об области данных, дополнительные микросхемы для хранения признаков могут иметь другую структуру). Две микросхемы по $128k \times 8$ поставить уже нельзя, так как нужна 32-разрядная шина данных, что могут обеспечить только четыре микросхемы. Распространенные PB SRAM в 100-выводных корпусах PQFP имеют 32-разрядную структуру $32k \times 32$ или $64k \times 32$ и используются по две или по четыре в платах для Pentium.

Ход работы.

1. Изучите краткую теорию и ответьте на контрольные вопросы.
2. Соберите и исследуйте работу ячейки памяти.
3. Объясните принцип работы и полученные результаты.
4. Заполните таблицу. Статическая Динамическая Видео Характеристики
5. Сделайте вывод

Практическое занятие №27 «Исследование работы постоянного запоминающего устройства»

Цель работы: Исследовать работу постоянно запоминающего устройства.

Содержание работы:

Постоянные запоминающие устройства (ПЗУ) делятся на четыре типа : - масочные, программируемые на заводе-изготовителе с применением специальных масок; - однократно программируемые потребителем путем пережигания нихромовых или поликремневых перемычек; - многократно программируемые потребителем со стиранием записанной информации ультрафиолетовым излучением; - многократно программируемые потребителем с электрическим стиранием информации. Рассмотрим ПЗУ второго типа, которое состоит из дешифратора $2^n \times 2^n$ и подключенных к его выходам схем ИЛИ с плавкими перемычками. ПЗУ содержит дешифратор 2×4 в виде подсхемы *pzt1_dcd* (А, В — кодовые входы, Е — вход разрешения, активный сигнал высокого уровня), к выходам которых можно подключить четыре элемента 4ИЛИ с дополнительными устройствами. На рисунке показаны два таких элемента, выполненных в виде отдельных подсхем *pzu_un1* и *pzu_un2*. Хотя эти элементы одинаковы, наращивание их на схеме путем копирования исключено из-за наличия пережигаемых перемычек — при наличии одноименных подсхем пережигание перемычки в одной подсхеме автоматически приведет к пережиганию такой же перемычки в другой. Поскольку программа не позволяет копировать подсхемы с их переименованием, все их приходится выполнять полностью. На схеме DO, D1 — выходы младшего и первого разрядов. Схема дешифратора *pzu_dcd* показана на рисунке. Дешифратор выполнен на трех элементах НЕ и четырех элементах ЗИЛИ-НЕ на транзисторах. Необходимость выполнения элементов дешифратора на транзисторах объясняется тем, что используемые в программе EWB математические модели цифровых ИМС не всегда позволяют подключать к ним обычные транзисторные схемы и, в частности, применяемые в рассматриваемом ПЗУ ячейки памяти в виде подсхемы *pzu_un1*. Ее внутренняя структура аналогична структуре ячейки памяти, используемой в ПЗУ K155PE3. В отличие от ИМС K155PE3, в которой в качестве элемента ИЛИ используется многоэмиттерный транзистор, приведены отдельные транзисторы T1...T4, эмиттеры которых через пережигаемые перемычки S1...S4 (имитируются предохранителями на 10 мА) соединены с формирователем на транзисторах T5, T6 и стабилитроне D. Транзистор T5 и стабилитрон D используются только в режиме программирования и в рабочем режиме не оказывают влияния на работу выходного каскада на транзисторе T6 (каскад с открытым коллектором), поскольку транзистор T5 закрыт низким потенциалом на его базе (напряжение пробоя стабилитрона D выбирается несколько больше напряжения питания транзистора T6, подаваемого во второй подсхеме в точку DO или D1 через резистор нагрузки). Внутренняя структура подсистемы дешифратора. Ячейка ПЗУ работает следующим образом. В исходном состоянии транзисторы T1...T4 и T6 закрыты, и при подключенной к T6 нагрузке на его выходе DO формируется сигнал логической единицы (около +5 В). При подаче на входы А, В дешифратора заданной кодовой комбинации, а на вход разрешения Е — сигнала логической единицы, один из транзисторов T1...T4 откроется и на выходе DO сформируется сигнал логического нуля. Так, например, при А=В=1 откроется транзистор T4 и сигнал логической единицы с его эмиттера через перемычку S4 поступит на делитель на резисторах R2, R3, транзистор T6 откроется, и на его выходе сформируется сигнал логического нуля. Очевидно, что и при любой другой двоичной комбинации будет происходить то же самое до тех пор, пока не будет разрушена соответствующая перемычка. Пережигание перемычек составляет суть программирования и осуществляется отдельно для каждого разряда (каждой ячейки) следующим образом: О на входы А, В (см. рис. 9.48) подается двоичная комбинация, соответствующая адресу пережигаемой перемычки в программируемом разряде (в ячейке *pzu_unx*, где x — номер ячейки); О к выходу ячейки Dx через резистор нагрузки (его сопротивление для конкретных ИМС указывается в документации, для K155PE3 составляет около 300 Ом) подключается источник напряжения 12,5 В, в результате чего стабилитрон D пробивается и транзистор T5 открывается; О на вход разрешения Е на короткое время подается сигнал логической единицы, при этом через один из открытых транзисторов T1...T2 и T5 протекает ток,

достаточный для пережигания соответствующей перемычки (длительность разрешающего сигнала на входе Е в промышленных программаторах может автоматически увеличиваться после нескольких неудачных попыток программирования одной и той же ячейки); О источник 12,5 В отключается, и после раскрытия соответствующей подсхемы можно убедиться, что перемычка действительно разрушена (в промышленных программаторах этот процесс сводится к проверке записи программируемой ячейки, и при отрицательном результате производится повторное программирование при большей длительности разрешающего сигнала). Заключительным этапом программирования серийных микросхем ПЗУ в промышленных условиях является электротермотренировка, которая проводится чаще всего в течение 168 часов при повышенной температуре, после чего производится дополнительный контроль записанной информации. Если при этом обнаруживается ошибка, допускается повторное программирование. Если ошибка снова повторяется, микросхема бракуется. Для моделирования процесса программирования к программируемой схеме необходимо подключить дополнительные элементы. Моделирование целесообразно начинать с одноразрядного ПЗУ. Следует отметить, что рассматриваемая модель ПЗУ достаточно капризна и при некоторых комбинациях входных сигналов моделирование не выполняется. Признаком невозможности моделирования является отсутствие слева от выключателя питания (в верхнем правом углу экрана) окошка с индикацией временных интервалов отсчета. По истечении некоторого времени может быть выдана рекомендация изменить установку погрешности моделирования (по умолчанию она равна 1%). Целесообразно установить ее максимально возможной (10%) в меню Circuit (команда Analysis Options, параметр Tolerance). Целесообразно также поварьировать сопротивлениями входных резисторов и резисторов нагрузки элементов НЕ и ИЛИ-НЕ, а также попробовать изменить параметры транзисторов. В крайнем случае можно ограничиться простейшим случаем — обойтись без дешифратора и использовать только одну ячейку памяти на рисунке, подключив к выходу и к одному из ее входов дополнительные элементы.. ПЗУ с пережигаемыми перемычками используются чаще всего в качестве специализированных дешифраторов, например для селекции У ВВ. ПЗУ с ультрафиолетовым стиранием используются в микропроцессорных системах для хранения управляющих программ, в частности, для размещения BIOS (Basic Input/Output System — основная система ввода/вывода, записанная в ПЗУ, отсюда ее полное название ROM BIOS). BIOS представляет собой набор программ проверки и обслуживания аппаратуры компьютера и выполняет роль посредника между операционной системой (ОС) и аппаратурой. BIOS получает управление при включении системной платы, тестирует саму плату и основные блоки компьютера — видеоадаптер, клавиатуру, контроллеры дисков и портов ввода/вывода, настраивает чипсет платы и загружает внешнюю ОС. При работе под управлением DOS/Windows 3.x/95/98 BIOS управляет основными устройствами, при работе под OS/2, Unix, Windows NT BIOS практически не используется, выполняя лишь начальную проверку и настройку. Обычно на системной плате установлено только ПЗУ с системным (Main System) BIOS, отвечающим за саму плату и контроллеры FDD (флоппи-дисков), HDD (жестких дисков), портов и клавиатуры; в системный BIOS практически всегда входит System Setup — программа настройки системы. Видеоадаптеры и контроллеры HDD с интерфейсом ST-506 (MFM) и SCSI имеют собственные BIOS в отдельных ПЗУ; их также могут иметь и другие платы — интеллектуальные контроллеры дисков и портов, сетевые карты и т.п. Обычно BIOS для современных системных плат разрабатывается одной из специализированных фирм: Award Software, American Megatrends (AMI), реже: Phoenix Technology, Microid Research; в данное время наиболее популярны BIOS фирмы Award. Некоторые производители плат (например, IBM, Intel и Acer) сами разрабатывают BIOS для них. Иногда для одной и той же платы имеются версии BIOS разных производителей, в этом случае допускается копировать прошивки или заменять микросхемы ПЗУ; в общем же случае каждая версия BIOS привязана к конкретной модели

платы. Раньше BIOS помещался в однократно программируемые ПЗУ либо ПЗУ с ультрафиолетовым стиранием; сейчас в основном выпускаются платы с электрически перепрограммируемыми ПЗУ (Flash ROM), которые допускают перепрограммирование BIOS средствами самой платы. Это позволяет исправлять заводские ошибки в BIOS, изменять заводские установки по умолчанию, программировать собственные экранные заставки и т.п. Тип микросхемы ПЗУ обычно можно определить по маркировке: 27xxxx — обычное ПЗУ, 28xxxx или 29xxxx — перепрограммируемые. Если на корпусе микросхемы 27xxxx есть прозрачное окно — это ПЗУ с ультрафиолетовым стиранием; если его нет — это однократно программируемое ПЗУ, которое можно лишь заменить на другое. Видео-ПЗУ (Video ROM) — постоянное запоминающее устройство, в которое записаны видео-BIOS, экранные шрифты, служебные таблицы и т.п. ПЗУ не используется видеоконтроллером напрямую, к нему обращается только центральный процессор, в результате выполнения программ, записанных в ПЗУ, происходят обращения к видеоконтроллеру и видеопамати. На многих современных видеокартах устанавливаются электрически перепрограммируемые ПЗУ (EEPROM, Flash ROM), допускающие перезапись пользователем под управлением специальной программы из комплекта карты. ПЗУ необходимо только для первоначального запуска видеоадаптера и работы в режиме DOS, Novell Netware и других ОС, функционирующих преимущественно в текстовом режиме; ОС Windows, OS/2 и им подобные, работающие через собственные видеодрайверы, не используют ПЗУ для управления адаптером либо используют его только при выполнении программ для DOS. При создании видео-BIOS все разработчики придерживаются рекомендаций VESA и VBE. VESA (Video Electronics Standards Association — ассоциация стандартизации видеоэлектроники) — организация, выпускающая различные стандарты в области электронных видеосистем и их программного обеспечения. VBE (VESA BIOS Extension — расширение BIOS в стандарте VESA) — дополнительные функции видео-BIOS по отношению к стандартному видео-BIOS для VGA, позволяющие запрашивать у адаптера список поддерживаемых видеорежимов и их параметров (разрешение, цветность, способы адресации, развертка и т.п.) и изменять эти параметры для согласования адаптера с конкретным монитором. По сути, VBE является унифицированным стандартом программного интерфейса с VESA-совместимыми картами, при работе через видео-BIOS он позволяет обойтись без специализированного драйвера видеокарты.

Ход работы:

1. Используя схему, проведите моделирование процесса программирования ПЗУ с пережигаемыми перемычками.
2. Составьте схему ПЗУ на базе двухразрядного ПЗУ и проведите моделирование процесса программирования одной из его ячеек памяти.
3. Зарисуйте предложенные схемы в отчёт.
4. Сравните предложенные модели и системы ПЗУ и охарактеризуйте их.
5. Сделайте выводы о проделанной работе.

Практическое занятие № 28 «Исследование работы с кэш-памяти»

Цель занятия: знакомство с организацией кэш-памяти в учебной ЭВМ.

Содержание работы:

1. Изучить теоретические сведения посвященные организации кэш-памяти в модели учебной ЭВМ.
2. Выполнить задание, представленное некоторой короткой "программой" (табл. 1), которую необходимо выполнить с подключенной кэш-памятью (размером 4 и 8 ячеек) в шаговом режиме для следующих двух вариантов алгоритмов замещения указанных в таблице 2). Номер задания определяется номером варианта.

Таблица 1. Варианты задания

№ вари- анта	Номера команд программы						
	1	2	3	4	5	6	7
1	RD#12	WR 10	WR @10	ADD 12	WR R0	SUB 10	PUSH R0
2	RD #65	WR R2	MOV 4,R2	WR 14	PUSH R2	POP R3	CALL 02
3	RD #16	SUB #5	WR 9	WR @9	WR R3	PUSH R3	POP R4
4	RD #99	WR R6	MOVR7,R6	ADD R7	PUSH R7	CALL006	POP R8
5	RD #11	WR R2	WR -@R2	PUSHR2	CALL005	POP R3	RET
6	RD #19	SUB#10	WR9	ADD #3	WR ©9	CALL006	POPR4
7	RD #8	WR R2	WR@R2+	PUSHR2	POP R3	WR-@R3	CALL003
8	RD #13	WR 14	WR @14	WR@13	ADD 13	CALL006	RET
9	RD #42	SUB#54	WR16	WR@16	WRR1	ADD@R1+	PUSH R1
10	RD #10	WR R5	ADD R5	WR R6	CALL 05	PUSH R6	RET
11	JMP006	RD #76	WR 14	WR R2	PUSH R2	RET	CALL001

Примечание. Не следует рассматривать заданную последовательность команд как фрагмент программы*. Некоторые конструкции, например, последовательность команд push R6, ret в общем случае не возвращает программу в точку вызова подпрограммы. Такие группы команд введены в задание для того, чтобы обратить внимание студентов на особенности функционирования стека.

* Напомним, что программа определяется как последовательность команд, выполнение которых позволяет получить некоторый результат.

Таблица 2. Пояснения к вариантам задания

Номера вариантов	Режим записи	Алгоритм замещения
1,7, 11	Сквозная	СЗ, без учета бита записи
	Обратная	О, с учетом бита записи
2,5,9	Сквозная	БИ, без учета бита записи
	Обратная	О, с учетом бита записи
3,6,12	Сквозная	О, без учета бита записи
	Обратная	СЗ, с учетом бита записи
4, 8, 10	Сквозная	БИ, без учета бита записи
	Обратная	БИ, с учетом бита записи

Где:

СЗ – случайное замещение записи;

О – очередь;

БИ – бит использования.

2.1. Ввести в модель учебной ЭВМ текст своего варианта программы (см. табл. 1), ассемблировать его и сохранить на диске в виде txt-файла.

2.2. Установить параметры кэш-памяти размером 4 ячейки, выбрать режим записи и алгоритм замещения в соответствии с первой строкой своего варианта из табл. 2.

2.3. В шаговом режиме выполнить программу, фиксируя после каждого шага состояние кэш-памяти.

2.4. Для одной из команд записи (WR) перейти в режим **Такт** и отметить, в каких микрокомандах происходит изменение кэш-памяти.

Практическое занятие № 29 «Исследование работы виртуальной памяти»

Цель занятия: произвести исследование работы виртуальной памяти.

Содержание работы:

Программа SystemInfo. Назначение программы заключается в выводе системной информации о процессоре, их количестве и виртуальной памяти. Для этого программа вызывает функцию *GetSystemInfo*. После запуска программы на экране открывается окно, показанное на рисунке 1.

По полученной информации определить основные характеристики процессора компьютера, число процессоров в компьютере, активные процессоры компьютера, дискретность (*granularity*) выделения памяти процессу, размер страницы и размер виртуального адресного пространства процесса (по значениям минимального и максимального адреса приложения).

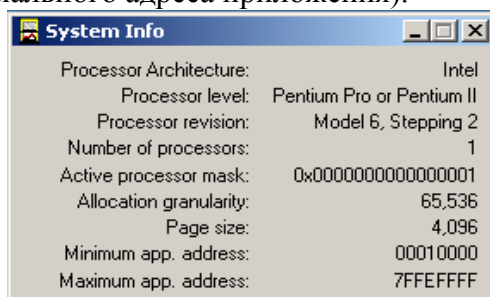


Рисунок 1.

Программа VMStat в отличие от программы SysInfo, дает динамику изменения основных данных о физической и виртуальной памяти, обновляемых каждую секунду. Это делает программу пригодной для мониторинга памяти в системе. Для этого программа вызывает функцию *GlobalMemoryStatus*. После запуска программы на экране открывается окно, показанное на рисунке 2. Элемент Memory load позволяет оценить, насколько занята подсистема управления памятью. Это число может быть любым в диапазоне от 0 до 100 (на практике от значения этого элемента толку немного).

Элемент TotalPhys отражает общий объем физической (оперативной) памяти в байтах. Однако реально память компьютера несколько больше (проверить экспериментально на своем компьютере). Причина, по которой *GlobalMemoryStatus* не сообщает полный объем памяти заключается в том, что система при загрузке резервирует небольшой участок оперативной памяти, недоступный даже ядру. Этот участок никогда не сбрасывается на диск.

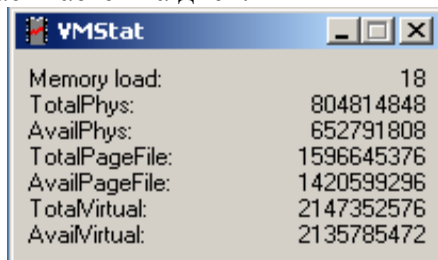


Рисунок 2.

Элемент AvailPhys сообщает число байтов свободной физической памяти. Следующий элемент TotalPageFile дает максимальное количество байтов, которое может содержаться в страничном файле (файлах) на жестком диске (дисках). Свободное число байтов в страничном файле, которое может быть передано любому процессу, показывает элемент AvailPageFile.

Элемент TotalVirtual отражает общее количество байтов, отведенных под закрытое адресное пространство процесса. Значение 2 147 352 576 ровно на 128 Кбайт меньше 2

Гбайт. Два раздела недоступного адресного пространства – от 0x00000000 до 0x0000FFFF и от 0x7FFF0000 до 0x7FFFFFFF – как раз и составляют эту разницу в 128 Кбайт.

Последний элемент, отображаемый на рисунке 2, - AvailVirtual – единственный элемент структуры, специфичный для конкретного процесса, вызывающего *GlobalMemoryStatus* (остальные элементы относятся исключительно к самой системе и не зависят от того, какой именно процесс вызывает эту функцию). При подсчете значения AvailVirtual функция суммирует размеры всех свободных регионов в адресном пространстве вызывающего процесса. В данном случае его значение говорит о том, что в распоряжении программы VMStat имеется 2 135 785 472 байтов свободного адресного пространства. Вычтя из значения TotalVirtual величину AvailVirtual, получим 11 567 104 байтов – такой объем памяти VMStat зарезервировала в своем виртуальном адресном пространстве.

Задание:

1. Используя программу SystemInfo, определите объем виртуальной памяти, доступной процессу. Сравните эти данные с результатом программы VMStat. Совпадают ли эти значения? Если нет, то почему?

Практическое занятие № 30 «Изучение фрагментации реальной памяти»

Цель занятия: изучение фрагментации реальной памяти.

Содержание работы:

Общие сведения об оперативной памяти

Оперативная память компьютера, иначе называемая оперативным запоминающим устройством (ОЗУ), используется для оперативного обмена информацией (командами и данными) между процессором, внешней памятью (например, дисковой) и различными подсистемами (видеоподсистема, подсистема ввода/вывода, коммуникации и т.д.). ОЗУ представляет собой память с произвольным доступом, поэтому для ее обозначения часто применяется аналогичное английское понятие RAM (Random Access Memory). Произвольность доступа подразумевает возможность операций чтения/записи с любой ячейкой ОЗУ в произвольном порядке. Основными требованиями, предъявляемыми к оперативной памяти являются следующие:

- Большой (по меркам электронной памяти) объем, измеряемый десятками и сотнями мегабайт.
- Быстродействие и производительность, позволяющие наиболее эффективно задействовать мощь современных процессоров.
- Высокая надежность хранения данных.

Физически ОЗУ представляет собой просто набор последовательно расположенных ячеек, каждая из которых определяется собственным уникальным адресом. В сумме все ячейки оперативной памяти образуют **адресное пространство (address space)**. В простейшем случае адресом некоторой ячейки памяти является просто ее номер относительно начала адресного пространства - нулевого адреса. В реальности же дело обстоит совсем иначе, поскольку по многим причинам программы не всегда могут использовать простейшую схему адресации оперативной памяти. Поэтому с точки зрения программиста наиболее важна не физическая, а логическая организация памяти. **Логическая организация адресного пространства - это совокупность способов использования программами и операционными системами оперативной памяти компьютера.**

Эффективное использование имеющегося ОЗУ является одной из главных задач, стоящих как перед программистами, так и перед пользователями, стремящимися получить максимальный выигрыш от использования персонального компьютера в своей деятельности. Поэтому для эффективного использования памяти необходимо четкое понимание принципов ее логической организации.

Распределение памяти

С точки зрения использования программами адресного пространства его можно подразделить на следующие участки:

- стандартная память (Conventional Memory)
- расширенная (Extended Memory), либо дополнительная (Expanded Memory), память
- верхняя память (High Memory или Upper Memory)

помимо перечисленных категорий следует также особо рассмотреть так называемую **область верхней памяти** (High Memory Area – HMA), расположенную сразу за пределами 1 Мб, а также области теневой памяти (Shadow Memory). Распределение памяти представлено на рис. 1.

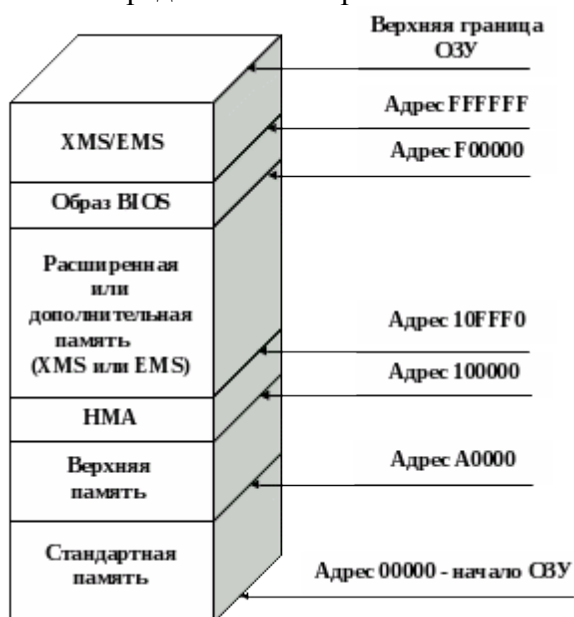


Рис 1. Распределение памяти в IBM PC
Стандартная память

По причинам исторического характера программы, являющиеся совместимыми с операционной системой MS-DOS и ее клонами, могут использовать в своей работе только первый мегабайт из всей памяти, установленной на компьютере, если не применяются специальные средства поддержки. Для объяснения этого обстоятельства следует обратить внимание на особенности архитектуры процессора Intel 8086 – базового процессора семейства 80x86. В процессоре 8086 16 ножек (pins) посылают сигналы, соответствующие 16 битам текущих данных, которыми процессор обменивается с системным ОЗУ. Однако данные не имеют никакого смысла сами по себе, если нет возможности следить за тем, что они собой представляют. Системная шина должна знать, куда направляются конкретные данные или откуда они поступили. Для выполнения этой задачи процессор использует еще 20 из своих ножек для создания уникальных адресов памяти. Это дает возможность процессору семейства 8086/8088 адресовать 2^{20} уникальных байтов, что и соответствует 1 мегабайту памяти. В оригинальной архитектуре персонального компьютера 640 Кб из этого 1 Мб было зарезервировано под DOS и прикладные программы, работающие под ее управлением, а область с 640 Кб до 1 Мб была зарезервирована для системного пользования. Большая часть 640 килобайт обычной памяти используется почти постоянно, но в области служебных адресов существуют участки, которые система не использует и которые могут быть доступны для других целей. Участки системной памяти, расположенные между 640 килобайт и 1 мегабайт интенсивно используются программами-расширителями памяти, такими, как Microsoft EMM386, Quarterdeck QEMM и Qualitas 386MAX.

Предварительно следует дать необходимые понятия о разделении адресного пространства в пределах 1 Мб на регионы. Общепринято делить 1 Мб памяти на 16

последовательных участков по 64 Кб каждый. Эти участки, помечаются шестнадцатеричными целыми числами от 0 до F, т.е., 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F. Из этих регионов DOS получает в свое распоряжение области с номерами 0 - 9 (10 первых участков по 64 Кб, итого 640 Кб), а остальные 6 участков памяти, расположенные между 640 Кб и 1 Мб (участки с номерами A, B, C, D, E, F), относятся к области служебных адресов и поступают в распоряжение системы. Удобство подобного разделения заключается в том, что номеру какого-либо 64Кб-региона может быть легко поставлен в соответствие адрес этого региона в адресном пространстве 1 Мб путем добавления 3 цифр справа к номеру региона.

Рассмотрим это на конкретном примере. Итак, регион A находится в адресном пространстве непосредственно за областью в 640 Кб, то есть, говоря иными словами, адрес его должен превышать 655359 байт (учитывая, что отсчет адресов идет с 0, число 655359 как раз соответствует 640 Кб) в десятичной нотации или 9FFFF в шестнадцатеричной. С учетом вышесказанного, адрес региона с номером A в адресном пространстве будет выглядеть как A000. С точки зрения 16 (10 в×процессора, этот адрес будет выглядеть как A0000, т.е. A000 шестнадцатеричной нотации). Это объясняется тем, что адрес, задаваемый в программе, может быть выражен 16-разрядным числом, т.е. содержать не более 4-х шестнадцатеричных цифр. Адресная шина процессора имеет 20 разрядов, поэтому перед вычислением адреса он переводится из 16-разрядной формы в 20-разрядную путем добавления справа еще одной шестнадцатеричной цифры, что эквивалентно умножению на 10 в шестнадцатеричной нотации или на 16 – в десятичной. Выполнив теперь перевод полученного адреса региона A – A0000 в десятичную форму получим число 655360, что доказывает, что регион с номером A находится как раз над границей памяти DOS в 640 Кб.

Выполнив аналогичные вычисления для региона с номером C, получим следующие результаты:

1. Добавим 3 шестнадцатеричных цифры справа к номеру региона, получим C000.
2. $16 = C0000$.×Для вычисления адреса по правилам процессора 8086 умножим этот адрес на 16: C000
3. Переведем его в десятичную форму: C0000 = 786432
4. Посчитаем сумму адресов, занимаемых предыдущими 12 регионами (0 - B): $12 * 65536 = 786432$ (отсчет идет с 0!).

Таким образом, вновь доказана корректность вычислений и удобство принятого обозначения регионов. Каждый 64-килобайтный регион может быть дополнительно поделен на 16 областей размером в 4 Кб, которые также помечаются шестнадцатеричными числами. Например, область A разбивается на дополнительные области от 0 до F: A0, A1, ..., AF. В отношении вычисления адресов этих дополнительных областей действует то же правило, что и для регионов, с той разницей, что к номеру области добавляются 2 цифры, а не 3, как в случае региона. Например, адрес области A7 может быть вычислен следующим образом:

1. Добавляем 2 цифры к номеру области: A700.
2. $16 = A7000$.×Переводим адрес в эффективный адрес процессора: A700
3. Вычисляем его десятичное значение: A7000 = 684032.

Легко проверить, что если к 640 Кб, которые занимают 10 регионов, идущие перед регионом A, добавить адреса, занимаемые 7 4-килобайтными областями внутри региона A (A0 – A6), то получится:

$$655359 + 4096 * 7 = 684031$$

Все приведенные выше сведения, возможно, грешат излишней детальностью, но являются, тем не менее, исключительно важными для понимания дальнейшего изложения.

Расширенная память

Появление расширенной памяти (Extended Memory Specification – XMS) было обусловлено появлением процессора 80286, который имеет на четыре адресных линии больше, чем 8086/8088, что позволяет адресоваться к количеству адресов, большему в 16 раз (2 в степени 4), т.е. к 16 Мб. Адреса памяти выше предела 1Мб и называются расширенной памятью. Характерной особенностью процессора Intel 80286 является возможность работы в 2-х режимах: в реальном режиме, в котором 80286 работает как более производительный 8086, и в защищенном, позволяющем адресовать память, лежащую за пределами 1-мегабайтного барьера. Работая в реальном режиме, 80286 не может получить доступа к расширенной памяти и по-прежнему ограничен 1 Мб адресуемого пространства. Чтобы использовать расширенную память, он должен работать в режиме виртуальной (защищенной) адресации. Изначально разработчики этого процессора не предусмотрели простого способа переключаться обратно в режим реальной адресации из режима виртуальной адресации (защиты). Это затруднило использование расширенной памяти в программах и она использовалась в основном под буферы ввода – вывода и печати. Однако, впоследствии, благодаря появлению специальных программ-расширителей – драйверов расширенной памяти – эта проблема была решена. Драйверы расширенной памяти обеспечивают полную поддержку расширенной памяти и позволяют программам осуществлять доступ к ней, а также выполнять программы из расширенной памяти. Наиболее популярными драйверами расширенной памяти являются HIMEM.SYS фирмы Microsoft и DOSHI.SYS фирмы Quarterdeck, входящий в состав пакета QEMM.

На компьютере с микропроцессором 80286, имеющем 24 адресных линии, можно получить до 15Мб расширенной памяти (весь объем памяти составляет 16Мб). Микропроцессоры 80386 и выше физически способны адресовать 4 гигабайта памяти, имея 32 адресные линии (2 в степени 32 байтов).

Дополнительная память

Дополнительная память (Expanded memory), часто называемая EMS (Expanded Memory Specification) или LIM (согласно ее разработчикам: Lotus, Intel и Microsoft), представляет собой способ доступа к памяти, лежащей за пределами 1 Мб, посредством окна размером 64 Кбайт, лежащего в области служебных адресов между 640 Кб и 1 Мб. При этом используется так называемая схема коммутации банков памяти (bankswitching) – способ управления памятью, когда физическая память разбита на несколько сегментов (банков) длиной, равной размеру адресного пространства процессора. В каждый момент процессор работает с одним банком. Память EMS логически подразделяется на участки размером по 16 Кбайт, называемые страницами. В процессе работы с дополнительной памятью часть содержимого банка размером в 4 страницы, или в 64 Кб **отображается** в определенный регион, лежащий в пределах 1 Мб, чаще всего в регион Е. Данный регион принято называть EMS-фреймом. EMS-фрейм как бы «скользит» по всему адресному пространству банка, что позволяет программам, физически обращаясь к адресам в пределах 1 Мб, получать, тем не менее, доступ ко всей памяти, которая установлена на компьютере.

Дополнительная память особенно полезна потому, что, в отличие от расширенной памяти, она использует только адреса ниже границы в 1Мб, следовательно, она может использоваться на машинах с процессорами 8086/8088, кроме того, при программировании нет необходимости использовать адреса, лежащие за пределами 1 Мб. Их отображение в EMS-фрейм будет выполнено драйвером дополнительной памяти. Недостатком EMS-памяти является некоторое замедление при обращении к памяти, которое объясняется накладными расходами, связанными с отображением страниц EMS в EMS-фрейм, переключением банков и т.д. Основными драйверами, обеспечивающими доступ к дополнительной памяти, являются EMM386 (Microsoft), QEMM (Quarterdeck) и 386MAX (Qualitas).

Область верхней памяти (НМА)

Начиная с процессора 80286 можно было наблюдать интересный эффект, связанный с тем, как происходит адресация в пределах 1 Мб. Согласно используемой при программировании в реальном режиме сегментной модели, все адресное пространство может быть представлено в виде пересекающихся между собой участков – сегментов размером в 64 Кб, которые должны начинаться с адреса, кратного 16. До тех пор, пока сегмент начинается с адресов, лежащих до 960 Кб включительно, все более или менее понятно. Однако, как быть в случае, когда сегмент начинается с адреса, лежащего выше 960 Кб? Ведь в этом случае все 64 Кб, входящие в сегмент, просто не уместятся в пределах 1 Мб. В случае с процессором 8086 просто-напросто происходил циклический возврат к началу. Например, если сегмент начинался, скажем, по адресу FB00 или 1028096 байт (1 Мб = 1048576 байт), то первые 4FFF или 20480 байт сегмента располагались в самом верху адресного пространства до 1 Мб включительно, а остальные B000 или 45056 байт – начиная опять с 0 адреса, то есть с самого начала адресного пространства процессора. В

процессоре 80286 ситуация изменилась, поскольку стало возможным адресовать физически более 1 Мб ОЗУ. Таким образом, когда процессор 80286 работает как 8086 и пытается получить доступ к одному из сегментов, лежащих близко к вершине адресного пространства, то адрес, полученный в результате операции будет лежать за пределами 1 Мб и, следовательно, возникает возможность обращаться к 64-килобайтной области (точнее, к 64 Кб минус 16 байт), лежащей за пределами 1 Мб, причем делать это в режиме реальной адресации, без каких-либо средств расширения DOS. Адресуемая подобным образом область размером почти в 64 Кб, лежащая в диапазоне 1024 Кб – 1088 Кб, получила название области верхней памяти (High Memory Area – НМА). В настоящее время типичным применением области НМА стало размещение в ней ядра DOS, что позволяет освободить дополнительно часть памяти ниже 640 Кб для пользовательских программ.

Чтобы ликвидировать разногласия в случаях, когда микропроцессор 80286 работает как микропроцессор 8086, в шину были встроены специальные средства (так называемый мост A20 – A20 Gate), которые заставляют память в режиме реального времени переходить циклически на низшие адреса, как это происходит в 8086/8088. Одним из применений драйвера HIMEM.SYS является отмена действия моста A20, что позволяет программам, написанным особым образом, обращаться к области НМА.

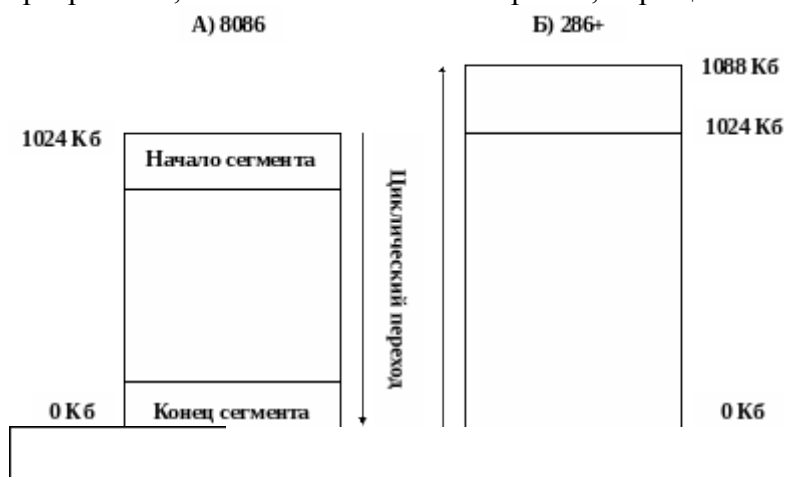


Рис 2. Особенности адресации в процессорах Intel 8086 (а) и Intel 80286 и старших (б)
Верхняя память

Память в высших адресах или верхняя память (High Memory) – это область объемом 384 Кбайт, расположенная между границами 640 Кбайт и 1 Мбайт (адреса A000 до FFFF), зарезервированная IBM для системного аппаратного обеспечения – для видеопамати, BIOS и прочего. Во многих случаях она, однако, используется не

полностью, и в ней образуются «дыры» – свободные участки, которые не используются в служебных целях, но к которым не имеют прямого доступа программы, работающие в 640 Кб основной памяти. Посредством расширителей памяти – таких как EMM386 (производства Microsoft), QEMM (производства Quarterdeck) и 386MAX (производства Qualitas) можно управлять этой областью зарезервированной памяти и перенести туда резидентные программы из основной памяти, тем самым освобождая больше обычной памяти под прикладные программы.

Структура верхней памяти

Для эффективного использования памяти, расположенной в верхних адресах, необходимо, прежде всего, знать, какие именно регионы этой памяти, используются всегда, а какие – зачастую остаются свободными.

Как уже упоминалось, верхняя память представлена 384 Кб, расположенными в адресном пространстве между 640 Кб и 1 Мб и включает в себя рассмотренные ранее регионы А, В, С, D, Е и F. Следует сразу же заметить, что не существует четких стандартов на использование всех этих регионов. В общем случае использование участков верхней памяти определяется конкретной конфигурацией и может быть разным на разных машинах. В то же время существуют определенные рассматриваемые ниже соглашения об их распределении.

Регионы А и В зарезервированы для видеопамати, но в большинстве случаев по крайней мере, часть их не используется. Оригинальный монохромный адаптер берет 4 Кбайт из региона В, чтобы разместить 4000 байт, необходимых для описания текстового экрана (25 строк на 80 столбцов, по 2 атрибута). Эта память начинается с адреса В000 и продолжается почти до адреса В100, хотя возможно использование области вплоть до В200 и даже В400 в зависимости от конкретной BIOS. В случае наличия в системе адаптера CGA для организации видеопамати используется 16 Кбайт, лежащих в адресах от В800 до ВС00. Хотя это оставляет остаток региона В в отрезке ВС00h - С000h при наличии CGA в принципе свободным, им можно пользоваться только на свой риск, поскольку некоторые программы предполагают, что эта неиспользуемая видеопамать доступна и прекрасным образом испортят все, что программа управления памятью поместит в эту область. Адаптеры EGA и VGA используют область В800 - С000 для организации видеопамати текстового режима и тех графических режимов, которые подражают CGA, а также всю область А для графики с высокой разрешающей способностью.

Другой стандартной областью является область F. В ней располагаются процедуры и данные BIOS и некоторая другая важная системная информация. Тем не менее, не все системы полностью занимают весь отведенный участок размером 64 Кб, и не все из этих 64 Кб нужны после загрузки. Например, в некоторых системах память от F000 до F800 используется программой установки и диагностики, которая может быть вызвана во время загрузки. Поскольку этот участок не используется после загрузки, можно позволить программе управления памятью разместить там что-нибудь другое.

Области С и D запутаны больше всего. В системах, имеющих EGA- и VGA-адаптеры, видео-ПЗУ обычно находится на дне области С. Остаток региона С, не используемый для размещения видеоПЗУ (адреса выше С800), а также вся область D применяются для размещения драйверов и прочего программного обеспечения, обеспечивающего поддержку всевозможного специфического оборудования – сетевых карт, адаптеров SCSI и т.д.

Наконец, регион Е в случае наличия в системе диспетчера дополнительной (EMS) памяти используется как EMS-фрейм.

Подытоживая все эти сведения, можно заметить, что в среднестатистической системе гарантированно занятыми являются только регионы А (видеопамать графического режима для адаптеров EGA, VGA и выше) и F (системная BIOS), кроме того, почти всегда (при наличии цветного адаптера) занята область размером 16 Кб,

лежащая в адресах В800 – ВС00 (видеопамять текстового режима), и область размером в 32 Кб в начале региона С в адресах С000 – С800 (видеоПЗУ). Занятость же прочих регионов верхней памяти в общем случае не определена. Таким образом, стандартно занято 64 Кб + 64 Кб + 16 Кб + 32 Кб = 176 Кб верхней памяти. Прочие 208 Кб, вообще говоря, могут быть свободны и использованы для хранения программ, которые обычно располагаются в основной памяти. Реально, во многих случаях занятым оказывается еще и регион Е, содержащий EMS-фрейм, а вторая половина региона С и регион D, напротив, чаще всего свободны. В силу данного обстоятельства в верхней памяти обычно остаются неиспользованными как минимум 96 Кб.

Разумеется, хранение в верхней памяти прикладных программ, работающих в адресном пространстве 640 Кб, было бы сопряжено со слишком большими трудностями, однако, если речь идет о драйверах, которые не должны постоянно выгружаться и загружаться в память, а должны находиться в ней резидентно, то использование верхней памяти в данном случае имеет большие преимущества. Поэтому была изобретена технология, позволяющая переносить драйверы, расположенные в обычной памяти, в свободные блоки верхней памяти, освобождая таким образом основное ОЗУ. Блоки верхней памяти, используемые подобным образом, получили название Upper Memory Blocks – UMB. Организацией блоков UMB и переносом в них драйверов из нижней памяти занимаются все те же менеджеры памяти, о которых уже шла речь выше – QEMM, EMM386, 386MAX и т.д.

В качестве примера, иллюстрирующего преимущество правильного использования верхней памяти и знания структуры памяти вообще, можно привести следующий факт. Если после загрузки DOS и необходимых драйверов устройств остаются свободными примерно 550-570 Кб ОЗУ, то при помещении ядра DOS в НМА с помощью драйвера HIMEM.SYS (или DOSHI.SYS), а также драйверов устройств в UMB посредством EMM386 или QEMM доступными для прикладных программ оказываются уже 620-635 Кб (а иногда – и больше). Таким образом, правильное конфигурирование системы и настройка менеджеров памяти позволяет буквально из ничего выжать дополнительные 50-85 Кб (удавалось получать даже 100 Кб)!

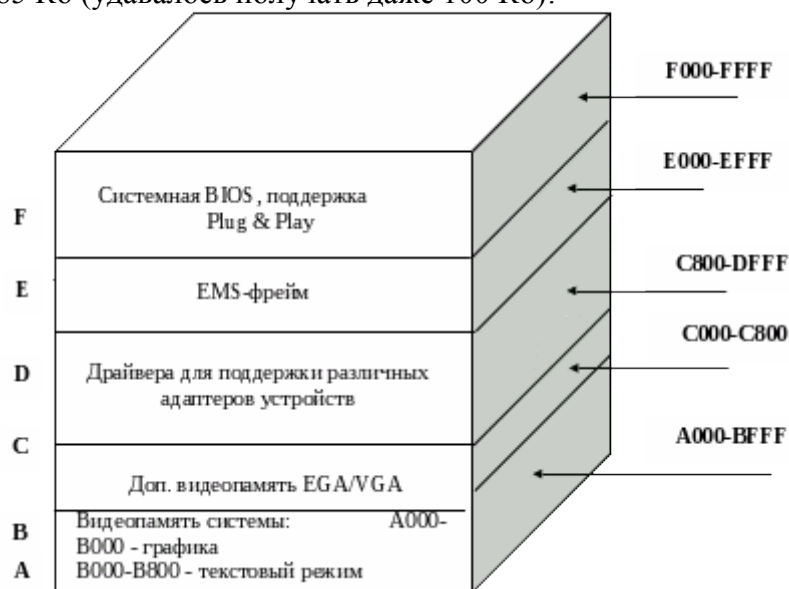


Рис 3. Структура блоков верхней памяти (UMB)

Теневая память

Теневая память (Shadow Memory) ускоряет обращение к обычно медленной постоянной памяти, такой как System BIOS или VideoROM BIOS за счет ее подмены быстрой оперативной памятью, имеющей более высокую разрядность. Теневая память подразделяется на теневую RAM (Shadow RAM) и теневую ROM (Shadow ROM). При инициализации Shadow ROM содержимое затеняемой области копируется в область

теневого памяти и при дальнейшем обращении к затемняемым участкам памяти вместо реального ROM подставляется участок Shadow RAM, запись же в эту область и вовсе блокируется. Shadow ROM отличается тем, что при записи в ROM запись происходит как собственно в ROM, так и в участок затемняющей его теневой памяти. Области, используемые для затемнения постоянной памяти находятся в верхней памяти. Как правило Shadow Memory располагается в регионах C и D и представляет собой, чаще всего, копию содержимого ПЗУ различных адаптеров для убыстрения доступа к ним. Обычно включение или выключение теневой памяти осуществляется через BIOS Setup компьютера. Эту же функцию имеют и некоторые менеджеры памяти, в частности EMM386.

1. Работая в Windows '9x, выполнить запуск утилиты MEM, рассмотреть и проанализировать выводимые сведения.
2. Перезагрузиться в режиме MS-DOS, выполнить запуск MEM, проанализировать текущее распределение памяти.
3. Выполнить полную перезагрузку компьютера, по нажатию клавиши F5 в момент загрузки отключить драйвера. Исследовать распределение памяти.

Практическое занятие № 31 «Построение блока памяти заданной емкости»

Цель занятия: построить блок памяти заданной емкости.

Содержание работы:

Вариант 0

1. Выполнить синтез дешифратора на четыре входа. Составить функциональную схему дешифратора.
2. Построить блок памяти заданной емкостью, используя указанные микросхемы памяти (таблица 1).
3. Реализовать функцию четырех переменных F на мультиплексоре восемь к одному:

$$F(A, B, C, D) = \Sigma(1, 3, 4, 6, 8, 10, 11, 12, 13).$$

Вариант 1

1. Выполнить синтез шифратора для преобразования одноразрядного числа в десятичной системе счисления в число в двоичной системе счисления.
2. Построить блок памяти заданной емкостью, используя указанные микросхемы памяти (таблица 1).
3. Реализовать функцию четырех переменных F на мультиплексоре восемь к одному:

$$F(A, B, C, D) = \Sigma(1, 3, 4, 6, 8, 10, 11, 12, 13).$$

Вариант 2

1. Выполнить синтез компаратора на «равенство» двух двухразрядных чисел. Составить функциональную схему компаратора.
2. Построить блок памяти заданной емкостью, используя указанные микросхемы памяти (таблица 1).
3. Реализовать функцию четырех переменных F на мультиплексоре восемь к одному:

$$F(A, B, C, D) = \Sigma(0, 1, 2, 5, 7, 11, 12, 14, 15).$$

Вариант 3

1. Выполнить синтез компаратора на «неравенство» двух двухразрядных чисел. Составить функциональную схему компаратора.
2. Построить блок памяти заданной емкостью, используя указанные микросхемы памяти (таблица 1).

3. Реализовать функцию четырех переменных F на мультиплексоре восемь к одному:

$$F(A,B,C,D) = \Sigma(0,2,3,4,7,10,11,12,15).$$

Вариант 4

1. Выполнить синтез преобразователя числа в прямом коде в число в обратном коде. Составить функциональную схему преобразователя.

2. Построить блок памяти заданной емкостью, используя указанные микросхемы памяти (таблица 1).

3. Реализовать функцию четырех переменных F на мультиплексоре восемь к одному:

$$F(A,B,C,D) = \Sigma(1,2,5,6,9,10,13,14,15).$$

Вариант 5

1. Описать принципы расширения ПЛМ по входам и выходам. Составить функциональную схему ПЛМ с расширением по входам

2. Построить блок памяти заданной емкостью, используя указанные микросхемы памяти (таблица 1).

3. Реализовать функцию четырех переменных F на мультиплексоре восемь к одному:

$$F(A,B,C,D) = \Sigma(1,2,3,4,5,8,9,12,13).$$

Вариант 6

1. Выполнить синтез RS-триггера в базисах И-НЕ, ИЛИ-НЕ. Составить функциональные схемы в базисах И-НЕ, ИЛИ-НЕ.

2. Построить блок памяти заданной емкостью, используя указанные микросхемы памяти (таблица 1).

3. Реализовать функцию четырех переменных F на мультиплексоре восемь к одному:

$$F(A,B,C,D) = \Sigma(0,1,3,4,5,7,8,9,11,12,13).$$

Вариант 7

1. Описать работу реверсивного регистра сдвига. Составить функциональную схему регистра.

2. Построить блок памяти заданной емкостью, используя указанные микросхемы памяти (таблица 1).

3. Реализовать функцию четырех переменных F на мультиплексоре восемь к одному:

$$F(A,B,C,D) = \Sigma(0,1,5,6,7,9,10,11,15).$$

Вариант 8

1. Построить схему суммирующего счетчика с $K_{сч}=11$. Изобразить временную диаграмму счетчика. Описать работу счетчика.

2. Построить блок памяти заданной емкостью, используя указанные микросхемы памяти (таблица 1).

3. Реализовать функцию четырех переменных F на мультиплексоре восемь к одному:

$$F(A,B,C,D) = \Sigma(0,2,4,7,9,11,12,13,14,15).$$

Вариант 9

1. Выполнить синтез одноразрядного комбинационного сумматора. Составить схему сумматора в базисе И-НЕ.

2. Построить блок памяти заданной емкостью, используя указанные микросхемы памяти (таблица 1).

3. Реализовать функцию четырех переменных F на мультиплексоре восемь к одному:

$$F(A,B,C,D) = \Sigma(1,4,5,7,8,10,13,14,15).$$

Таблица1

В ариант	Емкость блока памяти (Кбит)	Тип микросхемы памяти
0	32	KP565PY1A
1	64	K565PY3A
2	64	K565PY5Д3
3	16	K565PY1
4	16	KP537PY2
5	64	KP537PY9
6	16	KP537PY3
7	64	KP537PY8
8	16	KP537PY6
9	16	KP537PY14

Практическое занятие № 32 «Сравнение CISC – архитектуры и RISC - архитектуры»

Цель занятия: Провести сравнительную характеристику CISC – архитектуры и RISC – архитектуры.

Содержание работы:

Двумя основными архитектурами набора команд, используемыми компьютерной промышленностью на современном этапе развития вычислительной техники, являются архитектуры CISC и RISC. Основоположителем CISC-архитектуры - архитектуры с полным набором команд (CISC - Complete Instruction Set Computer) можно считать фирму IBM с ее базовой архитектурой IBM/360, ядро которой используется с 1964 г. и дошло до наших дней, например, в таких современных мейнфреймах, как IBM ES/9000.

Лидером в разработке микропроцессоров с полным набором команд считается компания Intel с микропроцессорами X86 и Pentium. Это практически стандарт для рынка микропроцессоров. Простота архитектуры RISC-процессора обеспечивает его компактность, практическое отсутствие проблем с охлаждением кристалла, чего нет в процессорах фирмы Intel, упорно придерживающейся пути развития архитектуры CISC. Формирование стратегии CISC-архитектуры произошло за счет технологической возможности перенесения "центра тяжести" обработки данных с программного уровня системы на аппаратный, так как основной путь повышения эффективности для CISC-компьютера виделся, в первую очередь, в упрощении компиляторов и минимизации исполняемого модуля. На сегодняшний день CISC-процессоры почти монополично занимают на компьютерном рынке сектор персональных компьютеров, однако RISC-процессорам нет равных в секторе высокопроизводительных серверов и рабочих станций. Основные черты RISC-архитектуры с аналогичными по характеру чертами CISC-архитектуры отображаются следующим образом (табл. 1):

CISC архитектура	RISC-архитектура
Многобайтовые команды	Однобайтовые команды
Малое количество регистров	Большое количество регистров
Сложные команды	Простые команды
Одна или менее команд за один цикл процессора	Несколько команд за один цикл процессора
Традиционно одно исполнительное устройство	Несколько исполнительных устройств

Таблица 1. Основные черты архитектуры

Одним из важных преимуществ RISC-архитектуры является высокая скорость арифметических вычислений. RISC-процессоры первыми достигли планки наиболее распространенного стандарта IEEE 754, устанавливающего 32-разрядный формат для представления чисел с фиксированной точкой и 64-разрядный формат "полной точности"

для чисел с плавающей точкой. Высокая скорость выполнения арифметических операций в сочетании с высокой точностью вычислений обеспечивает RISC-процессорам безусловное лидерство по быстродействию в сравнении с CISC-процессорами.

Другой особенностью RISC-процессоров является комплекс средств, обеспечивающих безостановочную работу арифметических устройств: механизм динамического прогнозирования ветвлений, большое количество оперативных регистров, многоуровневая встроенная кэш-память. Организация регистровой структуры - основное достоинство и основная проблема RISC. Практически любая реализация RISC-архитектуры использует трехместные операции обработки, в которых результат и два операнда имеют самостоятельную адресацию $-R1 := R2, R3$. Это позволяет без существенных затрат времени выбрать операнды из адресуемых оперативных регистров и записать в регистр результат операции. Кроме того, трехместные операции дают компилятору большую гибкость по сравнению с типовыми двухместными операциями формата "регистр - память" архитектуры CISC. В сочетании с быстродействующей арифметикой RISC-операции типа "регистр - регистр" становятся очень мощным средством повышения производительности процессора. Вместе с тем опора на регистры является ахиллесовой пятой RISC-архитектуры. Проблема в том, что в процессе выполнения задачи RISC-система неоднократно вынуждена обновлять содержимое регистров процессора, причем за минимальное время, чтобы не вызывать длительных простоев арифметического устройства. Для CISC-систем подобной проблемы не существует, поскольку модификация регистров может происходить на фоне обработки команд формата "память - память".

Существуют два подхода к решению проблемы модификации регистров в RISC-архитектуре: аппаратный, предложенный в проектах RISC-1 и RISC-2, и программный, разработанный специалистами IBM и Стэнфордского университета. Принципиальная разница между ними заключается в том, что аппаратное решение основано на стремлении уменьшить время вызова процедур за счет установки дополнительного оборудования процессора, тогда как программное решение базируется на возможностях компилятора и является более экономичным с точки зрения аппаратуры процессора.

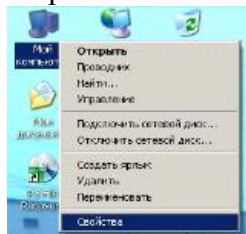
Практическое занятие № 33 «Исследование технического состояния и основных характеристик центрального процессора»

Цель занятия: провести исследование технического состояния и основных характеристик центрального процессора.

Содержание работы:

1. Определить тип и характеристики центрального процессора

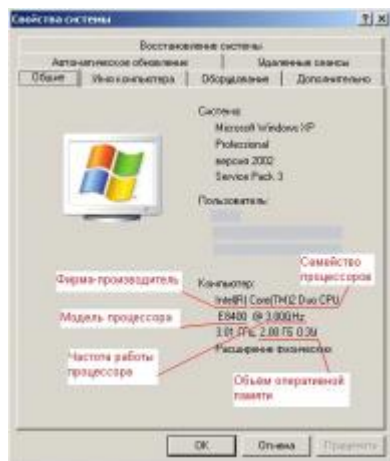
- На рабочем столе найдите иконку **Мой компьютер**.



- Через контекстное меню вызовите команду **Свойства**
- Откройте (если она не открыта) вкладку **Общие**.

Мой компьютер - Свойства (в контекстном меню) – Общие - My Computerà Propertiesà General

- В открывшемся окне найдите информацию о процессоре
- Приведите в отчете данные о процессоре и оперативной памяти для Вашего рабочего компьютера в лаборатории и для Вашего домашнего компьютера.



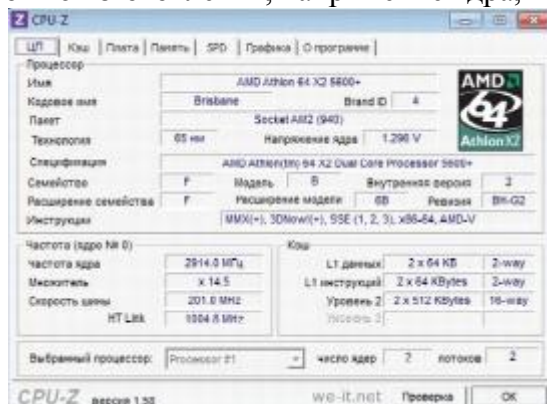
Заполните таблицу 1.

Наименование характеристики	Значение
Фирма производитель процессора:	
Семейство процессоров:	
Модель процессора:	
Частота работы процессора:	
Объем оперативной памяти:	

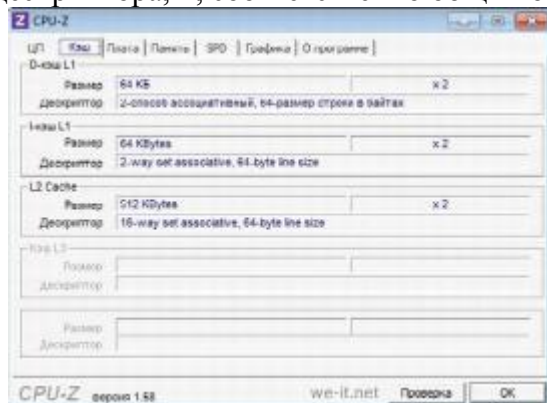
2. Работа с программой CPU-Z

- Запустите эту программу и рассмотрите открывшиеся окно.

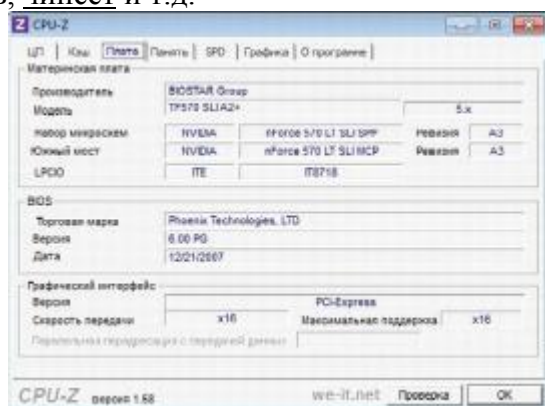
В текущей вкладке (рис.1), можно увидеть подробные характеристики вашего процессора. Здесь присутствуют пункты определяющие имя процессора, сокет, технологию изготовления, напряжение ядра, частоту ядра, множитель и многие другие.



На вкладке «Кэш», можно определить размер каждого из уровней кэша процессора и тип дескриптора, и, соответственно общий объем кэш-памяти процессора.



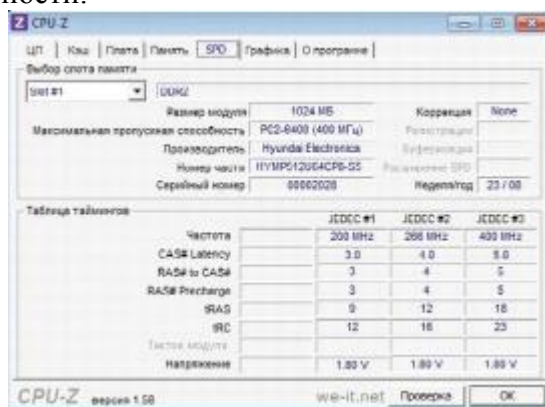
Теперь переходим на вкладку «Плата» и видим пункты, которые определяют характеристики системной (материнской) платы. Производитель платы, модель, чипсет и т.д.



Затем идём на вкладку память, где в CPU-Z выводится несколько пунктов по описанию характеристик памяти: тип, объём, количество каналов и тайминги.



В любом модуле оперативной памяти, присутствует чип SPD (Serial Presence Detected), в котором находится информация о рабочих частотах и таймингах памяти, которые необходимы для нормального функционирования модуля. То есть, по сути, на этой вкладке выводятся подробные характеристики каждого модуля памяти по отдельности.



Ну а на вкладке «Графика» вы можете увидеть несколько пунктов - характеристик вашей видеокарты.



Составьте отчет по программе CPU-Z^

III. Дополните таблицу 1, новыми характеристиками процессора: сокет, технологию изготовления, напряжение ядра, частоту ядра, множитель, размер КЭШа и другие.

Практическое занятие № 34 «Исследование работы процессоров с архитектурой VLIW»

Цель занятия: провести исследование работы процессоров с архитектурой VLIW.

Содержание работы:

Very long instruction word — сверхдлинное командное слово. Архитектура процессоров с явно выраженным параллелизмом вычислений, заложенным в систему команд процессора. Являются основой для архитектуры EPIC. Ключевым отличием от суперскалярных CISC-процессоров является то, что для них загрузкой исполнительных устройств занимается часть процессора (планировщик), на что отводится достаточно малое время, в то время как загрузкой вычислительных устройств для VLIW-процессора занимается компилятор, на что отводится существенно больше времени (качество загрузки и, соответственно, производительность теоретически должны быть выше). Примером VLIW-процессора является IntelItanium.

Центральный процессор (ЦП, CPU – англ. с é ntral pr ó cessing ú nit, дословно – центральное вычислительное устройство) – процессор машинных инструкций, часть аппаратного обеспечения компьютера или программируемого логического контроллера, отвечающая за выполнение арифметических операций, заданных программами операционной системы, и координирующий работу всех устройств компьютера.

Кэш (англ. cache) – промежуточный буфер с быстрым доступом, содержащий копию той информации, которая хранится в памяти с менее быстрым доступом, но с наибольшей вероятностью может быть оттуда запрошена. Доступ к данным в кэше идёт быстрее, чем выборка исходных данных из медленной памяти или их перевычисление, что делает среднее время доступа короче.

Кэширование – это использование дополнительной быстродействующей памяти (кэш-памяти) для хранения копий блоков информации из основной (оперативной) памяти, вероятность обращения к которым в ближайшее время велика.

Различают кэши 1-, 2- и 3-го уровней. Кэш 1-го уровня имеет наименьшую латентность (время доступа), но малый размер, кроме того кэши первого уровня часто делаются многопортовыми. Так, процессоры AMD K8 умели производить 64 бит запись+64 бит чтение либо два 64-бит чтения за такт, процессоры Intel Core могут производить 128 бит запись+128 бит чтение за такт. Кэш 2-го уровня обычно имеет значительно большие латентности доступа, но его можно сделать значительно больше по размеру. Кэш 3-го уровня самый большой по объёму и довольно медленный, но всё же он гораздо быстрее, чем оперативная память.

Многоядерные процессоры содержат несколько процессорных ядер в одном корпусе (на одном или нескольких кристаллах).

Двухядерность процессоров включает такие понятия, как наличие логических и физических ядер: например двухядерный процессор Intel Core Duo состоит из одного физического ядра, которое в свою очередь разделено на два логических. Процессор Intel Core 2 Duo состоит из двух физических ядер, что существенно влияет на скорость его работы. Компания AMD изготовила четырехядерные процессоры единым кристаллом (в отличие от Intel, процессоры которой представляют собой фактически склейку двух двухядерных кристаллов). Процессор получил название AMD Phenom X4. Его отставание от современных ему процессоров конкурента составляло от 5 до 30 и более процентов в зависимости от модели и конкретных задач.

На настоящий момент (1-2 квартал 2009 года) обе компании обновили свои линейки четырехядерных процессоров. Intel представила семейство Core i7, состоящее из трех моделей, работающих на разных частотах. Основными преимуществами данного процессора является использование трехканального контроллера памяти (типа DDR-3) и технологии эмулирования восьми ядер (полезно для некоторых специфических задач). Кроме того, благодаря общей оптимизации архитектуры удалось значительно повысить производительность процессора во многих типах задач. Слабой стороной платформы, использующей Core i7 является ее чрезмерная стоимость, так как для установки данного процессора необходима дорогая материнская плата на чипсете Intel-X58 и трехканальный набор памяти типа DDR3, также имеющий на данный момент высокую стоимость.

Компания AMD представила линейку процессоров Phenom II X4. При её разработке компания учла свои ошибки: был увеличен объем, а производство процессора было переведено на 45 нм техпроцесс, который позволил снизить тепловыделение и значительно повысить рабочие частоты. В целом AMD Phenom II X4 по производительности стоит на одном уровне с процессорами Intel предыдущего поколения (ядро Yorkfield) и весьма значительно отстает от Intel Core i7. Однако, принимая во внимание умеренную стоимость платформы на базе этого процессора, он будет пользоваться большим спросом, чем его предшественник.

Задание: Выписать основные особенности данной архитектуры. Сделать сравнительный анализ.

Практическое занятие № 35 «Исследование работы процессоров с архитектурой EPIC»

Цель занятия: провести исследование работы процессоров с архитектурой EPIC.

Содержание работы:

1.Направление EPIC фактически использует известную технологию **VLIW (Very Large INStruction Word)** - очень длинного командного слова.

Распараллеливание алгоритма между исполнительными модулями производится компилятором на этапе создания машинного кода, когда команды объединяются в связки и не конкурируют между собой за ресурсы микропроцессора. При этом упрощается блок управления на кристалле.

Особенности архитектуры EPIC:

1. Явный параллелизм в машинном коде. Поиск зависимостей между командами проводит не процессор, а компилятор.
2. Большое количество регистров.
3. Масштабируемость архитектуры до большого количества функциональных устройств (ALU, FPU, MMX, SSE и т. п.).
4. Применение предикатов. Предикатный подход исходит из предпосылки, что возросшие мощности микропроцессоров позволяют запускать параллельно команды из разных ветвей условного ветвления вместо того, чтобы ожидать формирования истинных признаков для выбора правильного направления или полагаться на блок предсказания

переходов, рискуя прийти к необходимости перезагрузки достаточно длинных конвейеров в случае неудачного предсказания. При этом каждая команда снабжается специальным полем условия (предикатом) (рис.1). По мере определения истинных признаков ветвления те команды, предикаты которых указывали, что они выбраны из другой ветви, снимаются с обработки в конвейере. Результаты команд не записываются в приемник до определения правильности направления перехода.

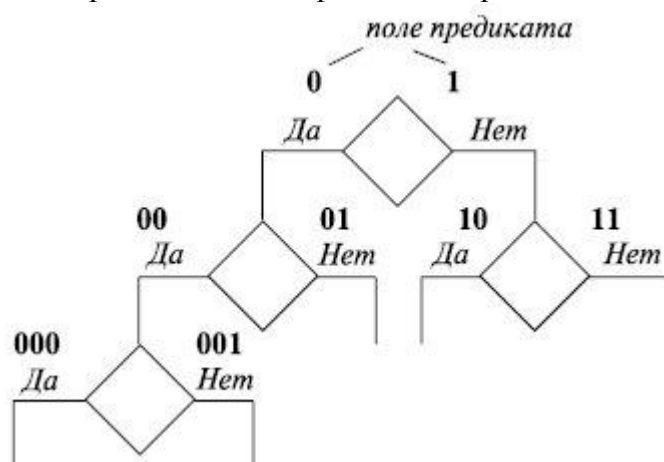


Рис. 1. Предикатное исполнение команд

Отметим **основные достоинства этого подхода**:

1. Упрощается архитектура процессора. Вместо логики распараллеливания на EPIC-процессоре можно разместить больше регистров, функциональных устройств и т. п.
2. Процессор не тратит время на анализ потока команд.
3. Возможности процессора по анализу программы во время выполнения ограничены сравнительно небольшим участком программы, тогда как компилятор способен произвести анализ всей программы.
4. Если некоторая программа должна запускаться многократно (а именно так и бывает в подавляющем большинстве случаев), выгоднее распараллелить ее один раз при компиляции, а не тратить на это время каждый раз, когда она исполняется на процессоре.

Однако архитектуре EPIC присущ и ряд **недостатков**:

1. Компилятор производит статический анализ программы, раз и навсегда планируя вычисления. Однако даже при небольших изменениях исходных данных путь выполнения программы существенно изменяется.
2. Значительно усложняются компиляторы, следовательно, увеличиваются время компиляции программы и число ошибок в самих компиляторах. Если первый фактор, учитывая высокое быстродействие современных компьютеров, не очень существен, то на второй следует обратить определенное внимание. Исследования показывают, что к моменту поставки даже ответственного программного обеспечения в нем содержится примерно 1 ошибка на 10 000 строк исходного кода. Следовательно, программа из 500 тыс. строк будет содержать около 50 ошибок, как бы хорошо ни работали тестировщики. И эти ошибки могут проявиться самым неожиданным образом.
3. Производительность микропроцессора во многом определяется качеством компилятора. Правда, здесь необходимо четко определить, что понимается под производительностью, ведь количество операций, выполняемых микропроцессором в единицу времени, от компилятора не зависит. Но это тема будет особо рассмотрена далее.
4. Увеличивается сложность отладки, так как отлаживается не исходная программа, а оптимизированный параллельный код. Программисту тяжело определить место и причину появления ошибки, так как в процессе трансляции исходной программы ее отдельные команды будут переставлены компилятором для обеспечения оптимальности работы микропроцессора.

Задание: Записать в тетрадь выявленные в процессе исследования достоинства и недостатки процессоров с архитектурой EPIC.

Практическое занятие № 36 «Параллельные порты и особенности их работы»

Цель занятия: установить соединение двух компьютеров через параллельные порты.

Содержание работы:

Задание 1. Установите соединение двух компьютеров через параллельные порты.

1. Выполните физическое подключение кабеля к **LPT-портам** компьютеров.
2. Настройте *ведомый* компьютер:
 - запустите **Мастер новых подключений** (*Сетевые подключения/ Мастер новых подключений*);
 - при необходимости введите код нашего города (8152) и щелкните **ОК**;
 - прочитайте информационное сообщение мастера и закройте окно кнопкой *Далее*;
 - установите **Тип сетевого подключения**, выбрав *Установить прямое подключение к другому компьютеру* и закройте текущее окно кнопкой *Далее*;
 - установите **Дополнительные параметры подключения**. Для этого выберите *Подключиться напрямую к другому компьютеру* и закройте это окно кнопкой *Далее*;
 - выберите **роль вашего компьютера** в создаваемом соединении. Для чего активизируйте радиокнопку *Ведомый компьютер* и закройте окно кнопкой *Далее*;
 - установите порт, через который осуществляется соединение, указав в списке **LPT**;
 - определите/добавьте пользователя, которому будет разрешен доступ к компьютеру (например, с именем *user* и паролем *user*):
 - щелкните по кнопке **Добавить**;
 - введите в поле **Пользователь** – имя пользователя (*user*);
 - введите в поле **Пароль** – пароль пользователя (*user*);
 - введите в поле **Подтверждение** – то же значение, что и в поле **Пароль** и закройте окно кнопкой **ОК**;
 - закройте окно **Разрешения пользователей** кнопкой *Далее*.
3. Установите *диапазон IP-адресов* для вашего соединения:
 - откройте диалоговое окно **Свойства: Протокол Интернета (TCP/IP)**, используя двойной щелчок по элементу списка *Протокол Интернета (TCP/IP)*;
 - выберите *Указать адреса TCP/IP явным образом*;
 - введите в поле **С** начало диапазона - 192.168.1.1;
 - введите в поле **ПО** конец диапазона - 192.168.1.2;
 - закройте окно кнопкой **ОК**;
 - закройте диалоговое окно **Программы работы с сетью** кнопкой *Далее* и завершите работу мастера кнопкой *Готово*.
4. Настройте *ведущий* компьютер:
 - все действия *до установки роли компьютера* в соединении, выполняйте аналогично настройке *ведомого* компьютера;
 - выберите **роль** вашего компьютера в соединении, Для этого активизируйте радиокнопку *Ведущий компьютер* и закройте окно кнопкой *Далее*;
 - введите **имя подключения**, например, имя *ведомого* компьютера;
 - установите **порт**, через который осуществляется соединение. Для этого выберите в списке **LPT** и закройте мастер кнопкой *Готово*;

- закройте диалоговое окно **Подключение** кнопкой **Отмена**.
- 5. Установите *соединение* с ведомым компьютером:
 - откройте диалоговое окно **Подключение (Пуск/Панель управления/Сетевые подключения/Прямой параллельный порт)**;
 - введите в поле **Пользователь** - имя пользователя, которому разрешен доступ к ведомому компьютеру (*user*);
 - введите в поле **Пароль** - пароль пользователя (*user*);
 - инициализируйте подключение кнопкой **Подключение**.

Задание 2. Передайте файл с ПК на ПК через созданное подключение.

1. На *ведомом* компьютере *предоставьте доступ* к папке **Прямое соединение**:
 - создайте в своей папке каталог **Прямое соединение**;
 - откройте диалоговое окно со свойствами папки (**Контекстное меню папки/Свойства**) и перейдите на вкладку **Доступ**;
 - активизируйте радиокнопку **Открыть общий доступ к этой папке**;
 - разрешите *всем пользователям полный доступ* к папке. Для этого:
 - щелкните по кнопке **Разрешения**;
 - установите флажок **Полный доступ – разрешить**;
 - примените параметры, нажав кнопку **ОК.**;
 - закройте диалоговое окно кнопкой **ОК**.
2. На *ведущем* компьютере просмотрите *ресурсы ведомого компьютера*:
 - выполните команду **Пуск/Выполнить**;
 - в появившемся диалоговом окне введите *имя компьютера* в формате **UNC**: *<имя_компьютера>* и нажмите **Enter**; Если возникнет ошибка, то вместо имени ведомого компьютера воспользуйтесь его *IP-адресом*.
 - введите в поле **Пользователь** имя пользователя в одном из следующих форматов:
 - *имя_ведомого_компьютера**имя_пользователя*
 - *или*
 - *имя_пользователя@имя_ведомого_компьютера*
 - в поле **Пароль** введите пароль (*user*).
2. Подготовьте файл размером 3 Мбайт (можно создать архив, содержащий рисунки).
3. Скопируйте этот файл в папку на удаленном компьютере и замерьте время, необходимое на его передачу.
4. Запишите полученный результат для сравнения с другими способами прямого соединения.
5. Разорвите соединение. Для этого откройте окно **Сетевые подключения** и выполните команду контекстного меню соединения – **Разорвать**.

Задание 3. Осуществите передачу файлов на удаленный компьютер с использованием файлового менеджера Total Commander.

1. Запустите на обоих компьютерах программу **Total Commander**.
2. Настройте *ведомый* компьютер:
 - выполните команду **Сеть/Соединение с другим компьютером через порт**;
 - установите **роль компьютера** в соединении, выбрав **Сервер**.
3. Настройте *ведущий* компьютер:
 - выполните команду **Сеть/Соединение с другим компьютером через порт**;
 - выберите **роль компьютера** – **Клиент**.
4. Скопируйте файл размером 3 Мбайт на ведомый компьютер и замерьте время, потраченное на его передачу.
5. Удалите созданные вами подключения на обоих компьютерах.
6. Выключите компьютеры и отсоедините кабель от LPT-портов компьютеров.

Задание 4. Настройте входящее модемное подключение к вашему ПК.

1. Физически подключите модем к компьютеру.

2. Откройте диалоговое окно **Сетевые подключения**, запустите **Мастер новых подключений** (*Файл/Новое подключение*) и закройте информационное окно мастера кнопкой *Далее*.
3. Выберите **тип подключения** – *Установить прямое подключение к другому компьютеру* и закройте окно кнопкой *Далее*.
4. Установите **режим работы** вашего компьютера, выбрав *Принимать входящие подключения* и закройте окно кнопкой *Далее*.
5. Укажите модем и закройте окно кнопкой *Далее*.
6. Установите параметр **Разрешить виртуальные частные соединения** и закройте окно кнопкой *Далее*.
7. Определите/Добавьте нового пользователя (например, *user*).
8. Запретите пользователю ответный вызов сервера:
 - откройте **свойства пользователя User** (выделите пользователя и воспользуйтесь кнопкой *Свойства*);
 - перейдите на вкладку **Ответный вызов сервера**, установите *Запретить ответный вызов* и закройте окно кнопкой **ОК**.
9. Настройте параметры элемента **Протокол Интернета (TCP/IP)**:
 - откройте **окно свойств** этого элемента;
 - выберите *Указать адрес TCP/IP явным образом*;
 - введите в поле **С** начало диапазона IP-адресов (192.168.150.10);
 - введите в поле **ПО** конец диапазона IP-адресов (192.168.150.125);
 - закройте окно кнопкой **ОК**.
10. Закройте диалоговое окно **Программы работы с сетью** кнопкой *Далее* и завершите работу мастера кнопкой *Готово*.
Теперь компьютер готов принимать входящие подключения от других ПК.
11. Проверьте созданное входящее подключение, позвонив с мобильного телефона по телефонному номеру к которому подключен модем.
Если все настроено правильно, то при получении звонка модем снимет трубку, и вы услышите характерный “писк”.

Практическое занятие № 37 «Последовательные порты и особенности их работы»

Цель занятия: установить соединение двух компьютеров через последовательные порты.

Содержание работы:

Задание 1.

Найти рисунок разъёмов системной платы. Указать название разъёмов и для каких устройств они применяются. Найти теоретические сведения по этим разъёмам.



Системная плата – печатная плата, соединяющая все узлы компьютера в одно устройство. Кроме термина "системная плата", используется название "**материнская плата**".

Основные компоненты системной платы:

1. разъем для подключения микропроцессора; Процессор – основная часть ЭВМ, обеспечивающая выполнение процедур обработки данных и взаимодействие всех устройств ЭВМ.

2. *Слот для оперативной памяти* – как Вы уже поняли, непосредственно в данный разъем устанавливаются модули с оперативной памятью (RAM). Сейчас все большую популярность набирает память DDR3, но при этом DDR2 еще используется, что называется «вовсю».

3. **AGP**(отангл.*Accelerated Graphics Port*, ускоренный графический порт) — специализированная 32-разрядная системная шина для видеокарты, разработанная в 1996 году компанией Intel. Появилась одновременно с чипсетами для процессора Intel Pentium MMX; у сторонних производителей появилась в чипсетах MVP3, MVP5 cSuper Socket 7. Основной задачей разработчиков было увеличение производительности и уменьшение стоимости видеокарты, за счёт уменьшения количества встроенной видеопамяти. По замыслу Intel, большие объёмы видеопамяти для AGP-карт были бы не нужны, поскольку технология предусматривала высокоскоростной доступ к общей памяти. Её отличия от предшественницы, шины PCI:

4. *Слот PCI (Peripheral Component Interconnect)* – если перевести название данного слота, то мы получим «разъем для подключения (установки) периферийных устройств». Данный разъем предназначен для подключения различных карт расширения: сетевые и звуковые карты, модемы и так далее. Это уже устаревший компонент, который вытесняется слотами PCI-Express, однако на материнских картах обычно присутствуют и тот и другой.

5. **ATA**(англ.*Advanced Technology Attachment*) —параллельный интерфейс подключения накопителей (жёстких дисков и оптических дисководов) к компьютеру. В 1990-е годы был стандартом на платформе IBM PC; в настоящее время вытесняется своим последователем —SATAи с его появлением получил название **PATA**(Parallel ATA).

6) *Северный мост* – одна из составляющих чипсета, который серьезно влияет на расширения systemboard, да и на производительность компьютера вообще. Такое название данный узел получил из-за своего «географического» расположения на материнской плате: он расположен в верхней части «материнки», рядом с центральным процессором, следовательно, южный мост (о нем читаем ниже) расположился в нижней части

системной платы. В обязанности Северного моста входит обеспечение бесперебойной передачи данных в связке «центральный процессор - оперативная память - графический адаптер».

7) **Южный мост** – на системных платах от Intel южный мост называется ICH (I/O Controller Hub). На южный мост кладется ответственность за передачу данных между такими портами как USB, оптическими приводами и жестким диском, кроме того, южный мост отвечает за устройства ввода: клавиатуру, мышь. Рядом с южным мостом располагается слот PCI.

8) **SATA**(англ.*Serial ATA*) —последовательный интерфейс обмена данными с накопителями информации. SATA является развитием параллельного интерфейса ATA(IDE), который после появления SATA был переименован в PATA (Parallel ATA).

9) **USB-контроллер**в составе платформы персонального компьютера обеспечивает коммуникацию с периферийными устройствами, подключенными к универсальной последовательной шине. USB-контроллер является устройством, способным взаимодействовать с оперативной памятью в обход центрального процессора в режиме прямого доступа к памяти.

По способу интеграции контроллер для USB-шины может быть задействован в составе системной логики или в виде дискретного чипа как на самой на системной плате, так и на плате расширения. По способу подключения USB-контроллер может быть выполнен для PCI-шины, либо для шины PCI Express.

10)USB-порты являются своего рода стандартом для подключения внешних устройств, к которому стремятся все производители этих устройств.

11) Это сетевая карта. Если контроллер интегрированный, т.е. сетевой разъем находится на общей колодке, там где USB, клавиатура и мышь, то для установки драйверов надо использовать диск, который прилагается к материнской плате. Если сетевой контроллер вставлен в один из PCI разъемов, то для его установки надо использовать соответствующие драйверы. Скачать их можно из Интернета, набрав в поисковой строке то, что написано на чипе самого контроллера.

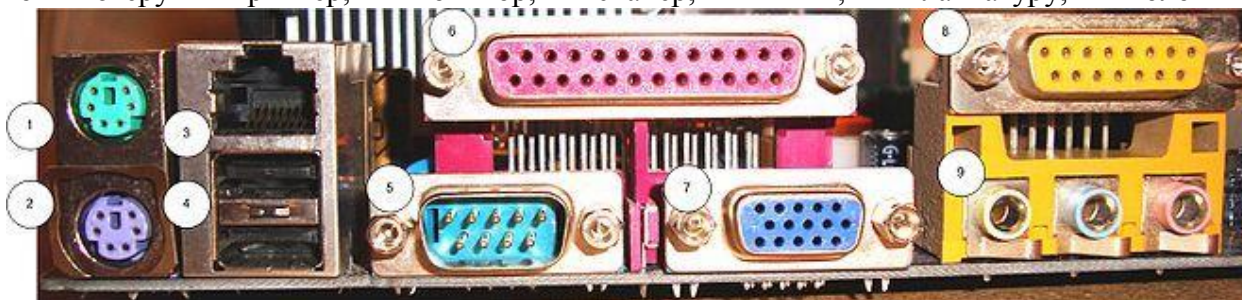
12) Например, наличие в устройстве порта Ethernet 10/100 говорит о том, что через него можно работать по технологиям 10BASE-T и 100BASE-TX, а порт Ethernet 10/100/1000 — поддерживает стандарты 10BASE-T, 100BASE-TX и 1000BASE-T.

13) Гигабитный Ethernet- это стандарт для локальной вычислительной сети (LAN), который обеспечивает скорость передачи данных 1 миллиард битов в секунду (один гигабит). Определен в стандартеIEEE802.3 и есть первые изделия. Гигабитный Ethernet используется как основа (backbone) предприятия.

Гигабитный Ethernet строится прежде всего на волоконном световоде (возможно и на медном кабеле, на очень короткие расстояния). Существующие локальные вычислительные сети Ethernet с платами 10 и 100 Мбит/с могут подключаться к основе Гигабитного Ethernet. Альтернативная технология, которая конкурирует с Гигабитным Ethernet - ATM.

Задание 2.

Определить внешние интерфейсы целевого компьютера. Подключить к целевому компьютеру принтер, монитор, сканер, мышь, клавиатуру, колонки.



Наружные разъёмы материнской платы: PS/2(1 -мышь, 2 -клавиатура), сетевой RJ-45 (3), USB(4), D-subminiature(9-контактный разъём COM-порта) (5), LPT порт(6), VGA порт(7), MIDI (8) и 3.5 мм аудио входы-выходы(разъём TRS)(9)

Пример:

1, 2. PS/2 — компьютерный порт(разъём), применяемый для подключения клавиатуры и мыши. Впервые появился в 1987 году на компьютерах IBM PS/2 и впоследствии получил признание других производителей и широкое распространение в персональных компьютерах и серверах. Скорость передачи данных — от 80 до 300 Кб/с и зависит от производительности подключенного устройства и программного драйвера.

Практическое занятие № 38 «Идентификация и установка процессора»

Цель занятия: Изучение характеристик процессора.

Содержание работы:

Задание 1.

Выберите процессор, подходящий для установки на целевой системной плате. Установите процессор на целевую системную плату

Задание 2.

Идентифицируйте процессор целевого компьютера.

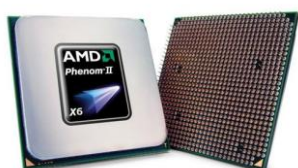
Назовите его основные характеристики процессора и заполните таблицу:

Тип процессора	Поколение	Год выпуска	Разрядность шины данных	Разрядность шины адреса	Первичная кэш-память, Кбайт	Тактовая частота шины, МГц	Тактовая частота процессора, МГц	Количество транзисторов, млн	Размер минимальной структуры, нм
Команды	Данные								

Дайте рекомендации по модернизации целевого компьютера.

Варианты процессоров:

1)



2)



3)



4)



5)



6)



7)



8)



9)



10)



11)



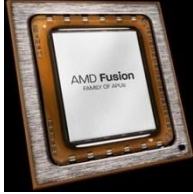
12)



13)



14)



15)



16)



17)



18)



19)



20)



21)



22)



Приложение. Маркировка процессоров.

Процессоры Intel

Само название моделей последних процессоров семейства Core несет в себе смысл. В нем присутствует один символ и четыре цифры. Буква обозначает его энергопотребление:

- «U» – Ultra low voltage (процессоры с TDP ниже 15 ватт);
- «L» – Low voltage (процессоры с TDP от 15 и до 25 ватт);
- «T» – sTandard mobile (процессоры с TDP от 25 и до 55 ватт);
- «E» – standard dEsktop (процессоры с TDP от 55 и до 75 ватт);
- «X» – eXtreme (процессоры с TDP выше 75 ватт).

Первые три литеры относятся к ноутбучным процессорам. Последующее четырехзначное число в общем случае указывает на производительность «камня» (чем больше число – тем производительнее процессор). В совокупности с этим, первая цифра обозначает семейство, к которому относится рассматриваемый процессор. Второе число определяет иерархию чипов внутри семейства. И так далее.

Маркировка наносит прямо на крышку теплораспределителя ЦП. Там же располагаются название компании-изготовителя, иногда основные характеристики ядер, спрятанных под крышкой, и, самое главное, информация sSpec и маркировка конкретного «камня».

По строчке sSpec можно определить все характеристики процессора.

В маркировке конкретного кремниевого чуда присутствует информация о месте и времени его изготовления, а также относительно партии. Первая цифра (или буква) интеловских процессоров означает страну-изготовителя:

- 0 – Коста Рика;
- 1 – Кавита и Филиппины;
- 3 – опять Коста Рика;
- 6 – Аризона;
- 7 – опять Филиппины;
- 8 – Ирландия;
- 9 – Малайзия;
- L – Малайзия;
- Q – Малайзия;
- R – Филиппины;
- Y – Ирландия.

Второй символ означает год выпуска, а третий и четвертый – неделю выпуска. Остальные «иероглифы» определяют порядковый номер изделия. Например, маркировка L708B0220 означает, что этот процессор изготовлен в Малайзии в 2007 году на 8 неделе.

Что означают последние цифры (после второй буквы), точно никто не знает – эта информация не для потребителя и даже далеко не для всех работников Intel. Возможно, это номер партии или кремниевых пластин.

Аналогичная ситуация – с чипсетами материнских плат. На них кроме производителя и кодового имени также указывают неделю выпуска. По неделе выпуска можно найти материнскую плату с более удачным чипсетом, способным работать с более высокой частотой системной шины. Ситуация с отбором партий: самые удачные партии придерживаются для продукции самой Intel и первых ревизий топовых продуктов «любимых» партнеров (особенно хорошо это заметно на примере материнских плат Asus).

Процессоры AMD

CPU производства AMD имеют несколько другую маркировку, нежели у конкурентов. Первыми двумя латинскими буквами компания обозначает название бренда. Так у Athlon 64 X2 он обозначен символикой «AD». У процессоров Phenom и Phenom II, соответственно, «H». Следующая литера обозначает тепловыделение «камня»: A – стандартное; D – 35 ватт; I – 76 ватт; O – 65 ватт; X – 125 ватт и выше; и, наконец, V – 89 ватт. Затем идет номер модели, входящий непосредственно в название процессора. Далее наименование сокета («I» для процессоров под AM2 и «J» для «Феноменов» под AM2+), коды потребляемого напряжения, максимально допустимой температуры, КЭШа второго уровня (цифра 5 означает 1 Мб SRAM, 4 – 512 килобайт и 6 – 2 мегабайта КЭШ-памяти) и ревизии. Рассмотрим, например, процессор Phenom 2 920 с маркировкой HX920XCJ4DGI, где:

H – название бренда, Phenom;

D – означает, что процессор относится к десктопному сегменту;

HX920 – марка «камня»;

XC – тепловыделение, превышающее 125 ватт;

J – сокет AM2+;

4 – число ядер;

D – Размер КЭШа уже третьего уровня;

GI – ревизия.

Также на металлической крышке процессора AMD присутствуют данные о месте производства и времени его изготовления.

Практическое занятие № 39 «Исследование схемы реализации микропрограммного принципа управления процессором»

Цель занятия: Изучение принципов построения микропрограммного устройства управления.

Содержание работы:

1. Изучить описание и назначение органов управления и индикации лабораторного стенда.
2. Осуществить запись в память микропрограмм (ПМП) содержимого таблицы 7. Выполнить и зафиксировать результат выполнения каждой микрокоманды из приведенных примеров в заранее подготовленной таблице 9.
3. Записать в ПМП текст микропрограммы примера 4. Проконтролировать последовательность выполняемых микрокоманд и конечный результат нормализации. Исходные значения числа выбрать из таблицы 10 в соответствии с вариантом задания.
4. Разработать алгоритм и составить микропрограмму выполнения операции обмена информацией между двумя регистрами РОН. Исходные данные выбрать из таблицы 11 в соответствии с вариантом задания. Загрузить микропрограмму в ПМП, выполнить операцию обмена, зафиксировав результат каждой микрокоманды в заранее подготовленной таблице.
5. Разработать алгоритм, составить и выполнить микропрограмму подсчета числа единиц в слове, записанном в одном из регистров РОН. Исходные данные выбрать из таблицы 12 в соответствии с вариантом задания.

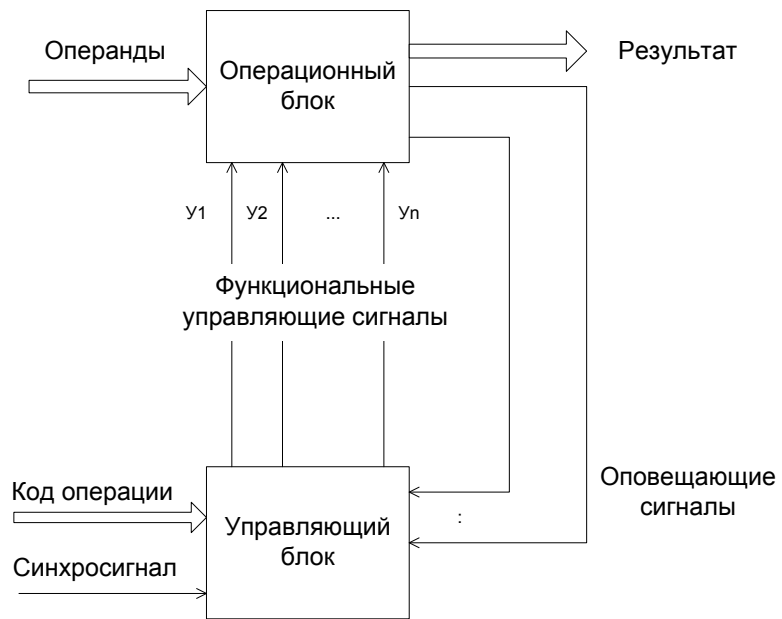


РИС. 1 Декомпозиция вычислительного устройства на операционный и управляющий блок

6.

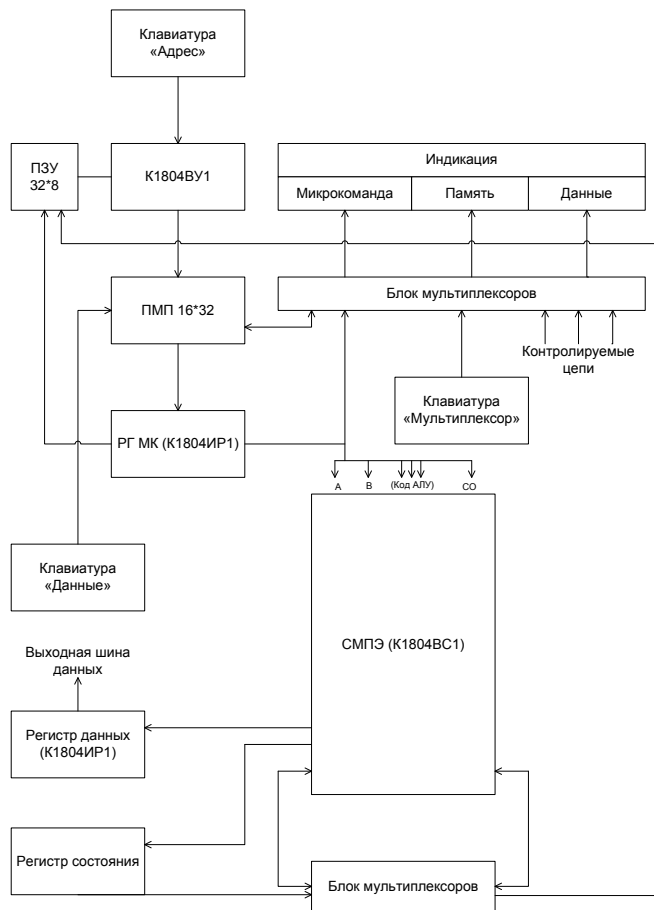
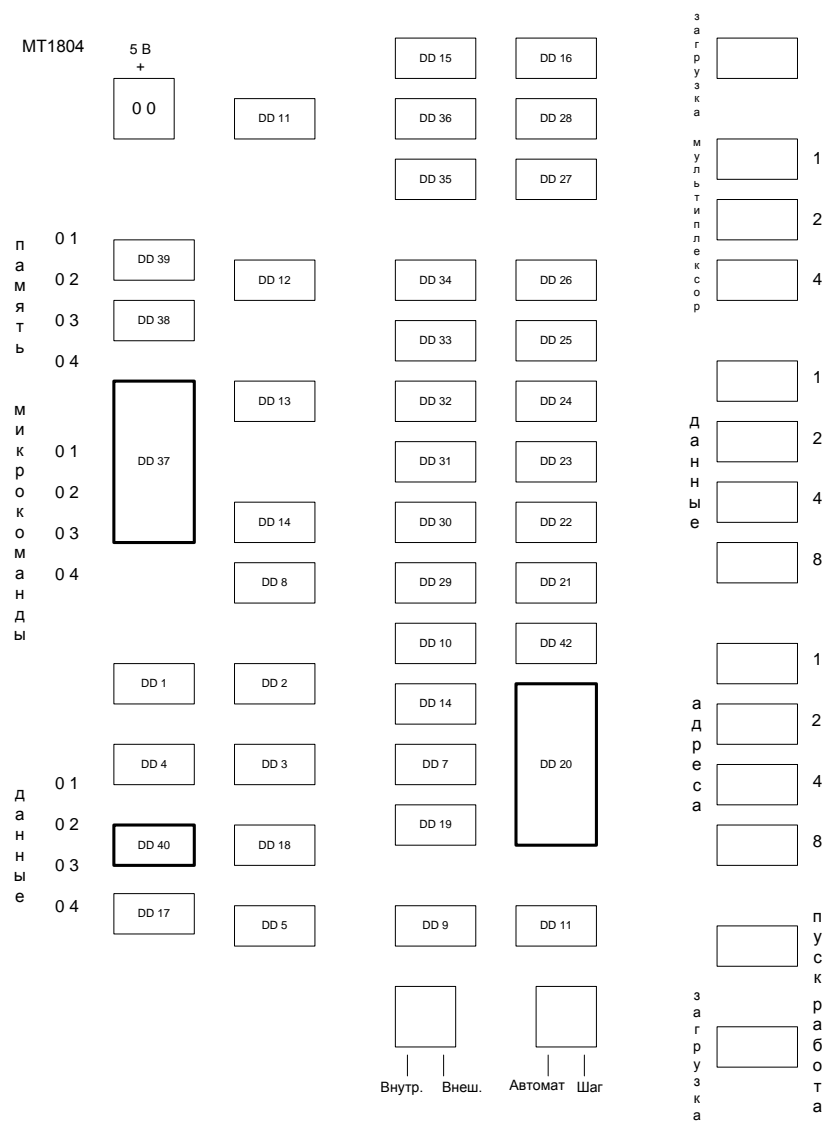


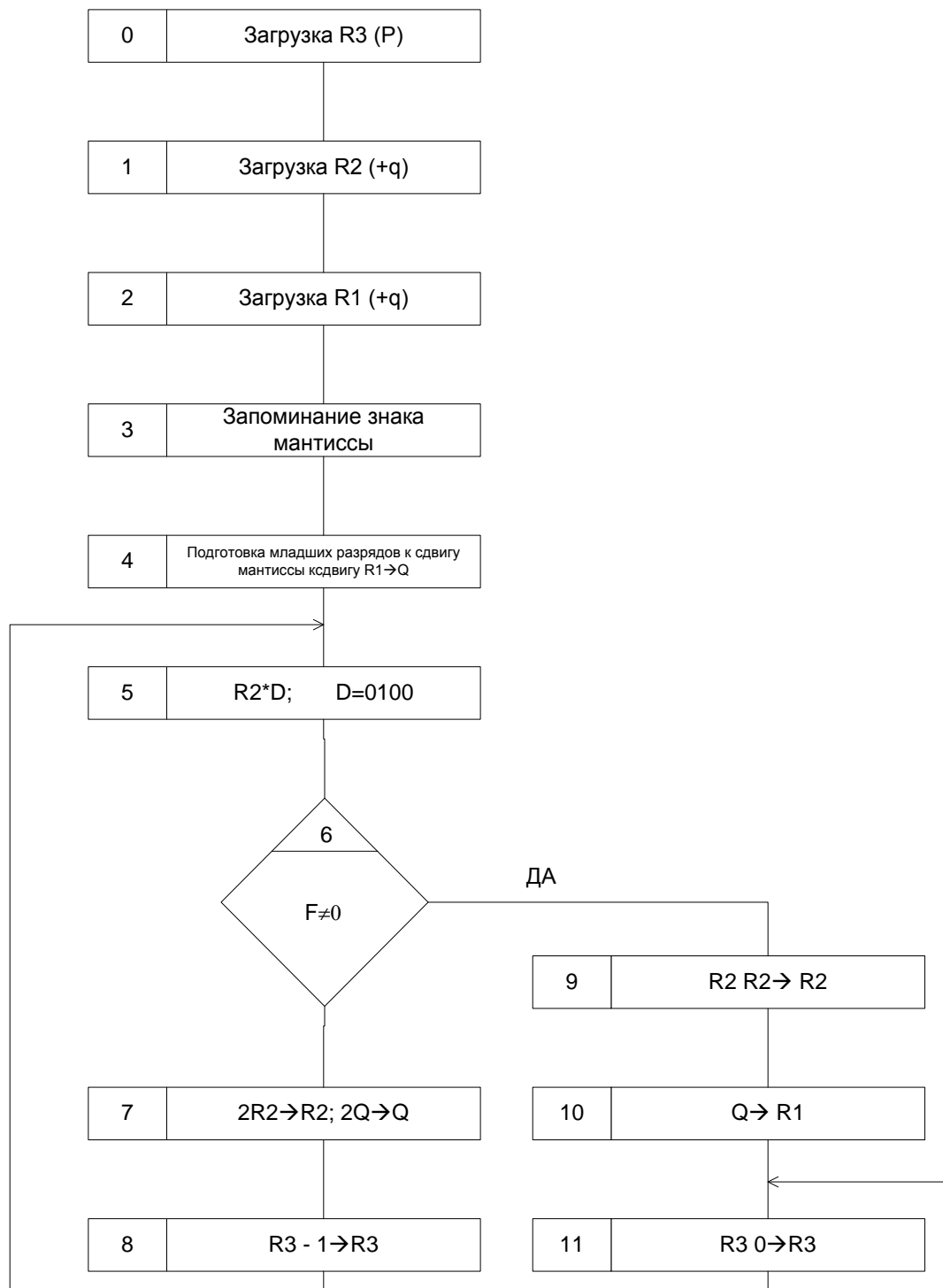
Рис. 2 Структурная схема лабораторного стенда

7.



8.

Рис. 3 Схема расположения элементов на стенде



9.

Рис. 4 Алгоритм операции нормализации

10. Таблица 1

Номер тетрады	Номер бита	Определение поля МК	Назначение
0	0 1 2 3	Данные «D»	Входные данные АЛУ
1	4 5 6 7	Адрес «B»	Адрес РОН по каналу «B»
2	8 9 10 11	Адрес «A»	Адрес РОН по каналу «A»
3	12 13 14 15	АЛУ СО	Функция АЛУ Входной перенос АЛУ
4	16 17 18 19	Источник операнда M0	Определение источник операнда для АЛУ Управляет операцией сдвига
5	20 21 22 23	Управление приемником результата M1	Определяют приемник результата выполнения микрокоманды Управляет операцией сдвига
6	24 25 26 27	Управление следующим адресом	Управляют выборкой следующего адреса микрокоманды в микропрограмме
7	28 29 30 31	Адрес перехода	Определяет адрес перехода

11.

12. Таблица 2

13. Зависимость функции АЛУ от кода поля микрокоманды

Двоичный код поля микрокоманды (разряды МК)			Функция АЛУ – F
14	13	12	
0	0	0	$R + S + C_0$
0	0	1	$S - R - 1 + C_0$
0	1	0	$R - S - 1 + C_0$
0	1	1	$R \vee S$
1	0	0	$R * S$
1	0	1	$\overline{R} * S$
1	1	0	$R \oplus S$

110

1	1	1	$\overline{R \oplus S}$
---	---	---	-------------------------

14.

15. Таблица 3

16. Зависимость источника операнда от кода поля микрокоманды

Двоичный код поля микрокоманды (разряды МК)			Источник операндов АЛУ	
18	17	16	R	S
0	0	0	A	Q
0	0	1	A	B
0	1	0	0	Q
0	1	1	0	B
1	0	0	0	A
1	0	1	D	A
1	1	0	D	Q
1	1	1	D	0

17.

18. Таблица 4

19. Зависимость приемника результата от кода поля микрокоманды

Двоичный код поля микрокоманды (разряды МК)			Загрузка	Выход K1804BC1 Y
22	21	20		
0	0	0	$F \rightarrow Q$	F
0	0	1	Нет загрузки	F
0	1	0	$F \rightarrow B (*)$	A
0	1	1	$F \rightarrow B$	F
1	0	0	$F/2 \rightarrow B; Q/2 \rightarrow Q$	F
1	0	1	$F/2 \rightarrow Q$	F
1	1	0	$2F \rightarrow B; 2Q \rightarrow Q$	F
1	1	1	$2F \rightarrow B$	F

20.

21. Где: F – результат операции АЛУ

22. Y – выходная шина данных БИС K1804BC1

23. (*) – на выходе операционного блока результат по адресу «А» микрокоманды

25. Зависимость сдвиговых операций от кода «M0» и «M1»

Код		Сдвиг вправо	Сдвиг влево
M0	M1		
0	0	<p>РОН</p>	<p>РОН</p>
0	1	<p>РОН</p>	<p>РОН</p>
1	0	<p>РОН</p>	<p>РОН</p>
1	1	<p>РОН</p>	<p>РОН</p>

26.

27. R0,R3 – значения соответствующих разрядов РОН;

28. Q0,Q3 – значения соответствующих разрядов регистра «Q»;

29. F3 – значение старшего разряда результата.

31. Зависимость адреса перехода от кода поля микрокоманды

Двоичный код поля микрокоманды (разряды МК)				Функция
27	26	25	24	
0	0	0	0	Переход на адрес из РГМК, если F=0
0	0	0	1	Переход на адрес из РГМК
0	0	1	0	Переход на следующий адрес
0	0	1	1	Переход на адрес по назначению переключателей адреса
0	1	0	0	Переход к подпрограмме, если F=0
0	1	0	1	Переход к подпрограмме
0	1	1	0	Возврат из подпрограммы
0	1	1	1	Переход по стеку
1	0	0	0	Окончить цикл и вытолкнуть из стека, если F=0
1	0	0	1	Загрузить стек (и продолжить)
1	0	1	0	Вытолкнуть из стека (и продолжить)
1	0	1	1	Окончить цикл и вытолкнуть из стека, если C4
1	1	0	0	Переход на адрес из РГМК, если F=0
1	1	0	1	То же, если F3
1	1	1	0	То же, если OVR
1	1	1	1	То же, если C4

32.

33. Где: F3 – старший (знаковый) разряд результата в АЛУ,

34. OVR – признак переполнения разрядной сетки,

35. C4 – признак выходного переноса

36. Таблица 7

37. Тексты микропрограмм примеров 1-3

Номер тетрады	Определение поля МК	Номер бита	Адрес микрокоманды												
			0	1	2	3	4	5	6	7	8	9	10	11	12
0	Данные D	0	1							1					
		1	0							0					
		2	1							1					
		3	0							0					
1	Адрес В	4	0	0	0	0	0	0	0	0	0				
		5	0	0	0	0	0	0	0	0	0				
		6	0	0	0	0	0	0	0	0	0				
		7	0	0	0	0	0	0	0	0	0				
2	Адрес А	8						0	0						
		9						0	0						
		10						0	0						
		11						0	0						
3	АЛУ	12	1	1	1	0	1	0	1	1	1				
		13	1	1	1	0	0	0	1	1	1				
	C0	14	0	0	0	0	0	0	1	1	1				
		15	0	0	0	1	0	0	0	0	0				
4	Источник операнда	16	1	1	1	1	1	1	1	1	1				
		17	1	1	1	1	1	0	0	1	1				
	M0	18	1	0	0	0	0	0	1	0	0				
		19								0	1				
5	Приемник	20	1	1	0	1	1	1	1	1	1				

[illegible]

39. Микропрограммы примера 4

Номер тетрады	Определение поля МК	Номер бита	Адрес микрокоманды												
			0	1	2	3	4	5	6	7	8	9	10	11	12
0	Данные D	0				0		0							
		1			0		0								
		2	p	+q	q	0		1							
		3				1		0							
1	Адрес B	4	1	0	1	0	1			0	1	0	1	1	
		5	1	1	0	0	0			1	1	1	0	1	
		6	0	0	0	0	0			0	0	0	0	0	
		7	0	0	0	0	0			0	0	0	0	0	
2	Адрес A	8				0		0				0			
		9				1		1				0			
		10				0		0				0			
		11				0		0				0			
3	АЛУ	12	1	1	1	0	1	0		1	1	1	1	1	
		13	1	1	1	0	1	0		1	0	1	1	1	
	C0	14	1	1	1	0	1	0		1	0	1	1	1	
		15	0	0	0	1	0	1		0	0	0	0	0	
4	Источник операнда	16	1	1	1	1	1	1		1	1	1	0	1	
		17	1	1	1	0	1	0		1	1	0	1	1	
	M0	18	1	1	1	1	0	1		0	0	0	0	0	
		19								1					
5	Приемник	20	1	1	1	1	0	1		0	1	1	1	1	

	результата	21	1	1	1	1	0	0		1	1	1	1	1	
		22	0	0	0	0	0	0		1	0	0	0	0	
	M1	23								1					
6	Управление следующим адресом	24	0	0	0	0	0	0	0	0	1	0	0	1	
		25	1	1	1	1	1	1	0	1	0	1	1	0	
		26	0	0	0	0	0	0	0	0	0	0	0	0	
		27	0	0	0	0	0	0	0	0	0	0	0	0	
7	Адрес перехода	28							1		1			1	
		29							0		0			1	
		30							0		1			0	
		31							1		0			1	
Примечания															
R3 := p															
R2 := +q															
R1 := q															
R0 :=SIGN															
Q := R1															
R2 * 0100															
ПЕРЕХОД по F = 0															
R2 := L1(R2,Q(3))															
Q := L1(Q,0)															
R3 := R3-1															
R2 := R2 □ R0															
R1 := Q															
R3 := R3 □ 0															

40. Таблица 9

41. Тексты микропрограмм задания

	Определение поля МК	Номер бита	Адрес микрокоманды												
			0	1	2	3	4	5	6	7	8	9	10	11	12
0	Данные D	0													
		1													
		2													
		3													
1	Адрес В	4													
		5													
		6													
		7													
2	Адрес А	8													
		9													
		10													
		11													
3	АЛУ	12													
		13													
		14													
		15													
4	Источник операнда	16													
		17													
		18													
		19													
5	Приемник	20													

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

47. Таблица 12

Вариант	Номер РОН	Содержимое
1	0	2
2	1	9
3	2	7
4	3	11
5	4	15
6	5	13
7	6	0
8	7	6
9	8	5
10	9	14

Практическое занятие № 40 «Исследование систем охлаждения процессоров»

Цель занятия: Исследование систем охлаждения процессоров

Содержание работы:

Охлаждение процессоров

Известно, что большинство современных процессоров развивают значительную тепловую мощность. Например, *Pentium II-300* потреблял при работе до 43 Вт, а *Pentium 4* (ядро *Willamette*) 2 ГГц — до 75 Вт! Очевидно, что практически вся электрическая мощность процессора преобразуется в тепловую энергию. Это тепло необходимо эффективно отводить, иначе температура компонентов процессора превысит допустимую границу и произойдет сбой в работе.

В качестве пассивного элемента охлаждения широко применяют радиаторы. Производители процессоров стараются рассчитать площадь поверхности радиаторов таким образом, чтобы обеспечивался некоторый запас по эффективности теплоотвода. Однако такой запас невелик и в случае разгона процессора не обеспечивает стабильной работы.

Если радиатор в одиночку не справляется с отводом тепла, на помощь приходят вентиляторы. Сочетание вентилятора и радиатора используют уже довольно давно, начиная с линейки процессоров *Pentium*. Однако в последних моделях процессоров размеры и мощность агрегатов охлаждения достигли чудовищных размеров. Известны модели воздушных систем охлаждения процессоров с весом до 900 грамм!

Как правило, в *Retail-ноставке* процессоры снабжены заводским комплектом устройств охлаждения, уже закрепленным на картридже. Такие стандартные устройства гарантированно обеспечивают стабильную работу процессора в любых режимах.

В системных платах (и корпусах) форм-фактора *AT* вентилятор охлаждения процессора напрямую подключается к разъемам блока питания. Поэтому управлять им невозможно. Современные устройства форм-фактора *ATX* и его модификаций предусматривают подключение вентилятора охлаждения процессора к специальному

разъему на системной плате. Шина питания и управления позволяет контролировать частоту вращения крыльчатки вентилятора и в совокупности с термодатчиками удерживать температуру микросхемы процессора в заданных пределах. К сожалению, в дешевых изделиях термоконтроль заключается в установке в вырез процессорного разъема термодиода, который даже не касается корпуса процессора. В этом случае значения температуры, выдаваемые системой, очень далеки от реальности.

Термоинтерфейс

В OEM-поставке процессоры обычно продают без устройств охлаждения, которые требуется приобретать отдельно. Таким образом, встает проблема выбора комплекта охлаждения и способа его установки. Часто с целью экономии средств покупается первое попавшееся дешевое устройство, обычно китайского производства. Такой подход чреват серьезными неприятностями. Подобные конструкции заявляются как универсальные, якобы рассчитанные на широкий спектр моделей процессоров.

Даже беглый осмотр этих изделий приводит к стойкому убеждению, что их создатели вообще не знакомы с основами термодинамики. Да и качество исполнения таково, что плоскость радиатора, примыкающую к процессору, как минимум пару дней надо шлифовать, а вентилятор грохочет не хуже пропеллера взлетающего «Мессершмитта». Служат такие устройства очень недолго и практически не способны к нормальному охлаждению.

Монтировать радиатор с вентилятором на корпус процессора следует только после обработки поверхности теплопроводной пастой. Отечественная паста КПП-8 стоит копейки и, хотя не имеет выдающихся показателей, достаточно качественно обеспечивает нормальный теплоотвод. Заводские стандартные радиаторы, как правило, имеют прикрытую липкой лентой площадку, на которую заранее нанесена теплопроводная паста.

В профессиональных изделиях в качестве пасты применяют специально подобранные составы, удельная теплопроводность которых на порядок выше, чем у КПП-8. К таковым относятся термопасты фирм *Bergquist (HF225UT)*, *Chromerics (T725)*, *Honeywell (PCM45)*, *ShinEtsu (PCS-TC011T-13)*, *Termagon (T~pcm905C)* и некоторые другие. Их желательно использовать для критичных по надежности систем.

Последствия перегрева

Попробуем оценить последствия перегрева для различных процессоров. В общем случае можно утверждать, что незначительный перегрев выражается в неспровоцированных зависаниях и «слетах» операционной системы и программ «тяжелого» класса: 3Д-графики, игр, обработки изображений или видео. Если встречаются проблемы такого рода, иногда достаточно тщательно сдуть пыль с радиатора и крыльчатки вентилятора, а также других компонентов. Но чаще решение состоит в замене выработавшего ресурс или некачественного вентилятора. Систематические сбои компьютера спустя определенное время после запуска говорят о том, что тепловой режим серьезно нарушен. Обычно это связано с выходом из строя вентилятора (охлаждения процессора или блока питания). Однако такие проблемы могут быть последствием установки новых энергоемких компонентов. В этом случае поставьте на процессор более мощную систему охлаждения или поменяйте размещение компонентов в корпусе.

Не для всех моделей процессоров перегрев проходит безболезненно. Особенно чувствительны к превышению предельной температуры процессоры *Athlon* и *Duron* фирмы *AMD*. Известны и подробно описаны случаи полного выхода из строя этих процессоров при отказе вентилятора. Процессоры *Intel* относятся к категории более стойких изделий. По крайней мере, автор ни разу не слышал о сгоревшем по причине отказа вентилятора процессоре *Intel*.

Такая разница в поведении современных продуктов *Intel* и *AMD* объясняется разным подходом к контролю за термическим режимом процессора. Фирма *AMD* до последнего времени считала, что охлаждение и контроль за температурой — задача системная и потому этим должны заниматься производители материнских плат. Однако

«материнский» датчик обычно крепится к радиатору системы охлаждения, а часто находится в вырезе разъема *Socket* и меряет, по сути, температуру материнской платы в данной точке. Сочетание процессоров *Athlon XP* с современными материнскими платами более надежно в смысле термоконтроля, поскольку в процессор встроен терморезистор, выдающий сведения о текущей температуре.

Фирма *Intel* переложила задачу контроля за температурой процессора на внутренние схемы. Особенно удачно эта проблема решена в процессорах *Pentium 4*, где схема *Thermal Monitor* обеспечивает не только контроль температуры, но и снижение рабочей частоты при перегреве. Тестирование процессора *Pentium 4* со снятой (!) системой охлаждения показало, что процессор сохраняет работоспособность, лишь замедляя работу. Модель *Pentium III* ведет себя более «грубо», но вполне предсказуемо: в случае перегрева процессор «зависает», отключая питание ядра.

Воздушные системы охлаждения

Воздушные системы охлаждения основаны на передаче тепловой энергии от процессора к радиатору, имеющему гораздо большую площадь поверхности, чем кристалл, через которую и отводится излишек тепла. Процессоры малой мощности (например, *VIA C3* или мобильные варианты *Pentium III*) обходятся только радиатором. Однако в большинстве настольных систем используют систему охлаждения, в которой поверхность радиатора обдувается воздушным потоком, создаваемым вентилятором. При тепловыделении мощностью до 30 Вт достаточно применить обычный вентилятор 50x10, обдувающий алюминиевый радиатор. Значительное превышение значения 30 Вт рассеиваемой мощности вызывает необходимость увеличивать площадь радиатора и производительность вентилятора. В сложных случаях (при тепловыделении процессора более 50 Вт) приходится оптимизировать форму ребер радиатора, применять материалы с хорошей теплопроводностью (медь), наращивать производительность вентилятора за счет увеличения диаметра крыльчатки и числа оборотов, а иногда устанавливать два вентилятора.

Главным «двигателем прогресса» в области построения систем охлаждения явилась фирма *AMD*, чьи старшие модели *Athlon* (с рабочими частотами более 950 МГц) выделяют более 50 Вт. Поэтому самые «изысканные» системы охлаждения разработаны именно под интерфейс *Socket A (Socket 462)*. Правда, изделия компании *Intel* серии *Pentium 4* не слишком отстают от *Athlon*. Например, *Pentium 4* с ядром *Willamette* и частотой 2 ГГц имеет тепловую мощность 75 Вт.

Задание:

1. Составить план конспекта.
2. Краткий конспект по плану.
3. Контрольные вопросы.

Практическое занятие № 41 «Исследование работы процессора семейства Intel.»

Цель занятия:изучить структурную организацию ЭВМ, построенную на базе процессоров Intel, набор и доступность регистров процессора Intel 8086, организацию оперативной памяти, способы представления и форматы данных ЭВМ.

Содержание работы:

Центральный процессор

Центральный процессор – основное устройство ЭВМ, выполняющее управление обменом информации по интерфейсу "общая шина" и все необходимые арифметико-логические операции обработки информации. Ниже будут рассмотрены основы архитектуры процессора Intel 8086 – родоначальника семейства процессоров Intelx86 и IntelPentium. На базе этого семейства в настоящее время построено большинство персональных компьютеров (ПК).

В состав центрального процессора входят устройство управления и блок микрокоманд, регистровая память, арифметико-логическое устройство и кэш-память 1-го уровня.

Устройство управления предназначено для организации обмена информации между центральным процессором и остальными частями ЭВМ. С помощью этого устройства процессор считывает команды и данные из оперативной памяти и записывает данные в память. Другой его функцией является дешифрирование кодов команд.

Результат дешифрирования используется блоком микрокоманд, для генерации так называемой микропрограммы – последовательности внутренних сигналов процессора, обеспечивающих совместное функционирование устройств центрального процессора с целью исполнения той или иной команды.

Регистровая память используется центральным процессором для хранения данных и управляющей информации. Регистровая память образуется так называемыми регистрами, которые представляют собой ячейки памяти, входящие в состав процессора и доступные из машинной программы. Доступ к регистрам осуществляется значительно быстрее, чем к ячейкам оперативной памяти, поэтому использование регистров заметно уменьшает время выполнения программ.

Все регистры центрального процессора Intel 8086 имеют размер слова (два байта – 16 битов), за каждым из них закреплено определенное имя (*AX*, *SP* и т.п.).

По назначению и способу использования регистры можно разбить на следующие группы:

- регистры общего назначения (*AX*, *BX*, *CX*, *DX*, *BP*, *SI*, *DI*, *SP*);
- сегментные регистры (*CS*, *DS*, *SS*, *ES*);
- счетчик команд (*IP*);
- регистр флагов (*Flags*).

Запомнить названия регистров совсем не сложно, если учесть, что вторая буква 'X' в названиях регистров означает принадлежность к группе регистров, предоставляющих доступ к отдельным байтам, вторая буква 'S' – принадлежность к группе сегментных регистров. Дальнейшая расшифровка этих названий: *A* – accumulator (аккумулятор); *B* – base (база); *C* – counter (счетчик); *D* – data (данные); *BP* – basepointer (указатель базы); *SI* – sourceindex (индекс источника); *DI* – destinationindex (индекс приемника); *SP* – stackpointer (указатель стека); *CS* – codesegmentregister (сегментный регистр кода команд); *DS* – datasegmentregister (сегментный регистр данных); *SS* – stacksegmentregister (сегментный регистр стека); *ES* – extensionsegmentregister (дополнительный сегментный регистр данных); *IP* – instructionpointer (счетчик команд).

Регистры общего назначения применяются для хранения данных. Их можно использовать во всех арифметических и логических командах. В то же время каждый из них имеет определенную специализацию (некоторые команды "работают" только с определенными регистрами). Например, команды умножения и деления требуют, чтобы один из операндов находился в регистре *AX* или в регистрах *AX* и *DX* (в зависимости от размера операнда), а команды управления циклом используют регистр *CX* в качестве счетчика цикла. Регистры *BX* и *BP* очень часто используются как базовые регистры при обращении к памяти, а *SI* и *DI* – как индексные. Регистр *SP* обычно указывает на вершину стека – специально организованной области памяти, аппаратно поддерживаемой в ЭВМ.

Регистры *AX*, *BX*, *CX* и *DX* конструктивно устроены так, что возможен независимый доступ к их старшей и младшей половинам; можно сказать, что каждый из этих регистров состоит из двух байтовых регистров, обозначаемых *AH*, *AL*, *BH* и т.д. (*H* – high, старший; *L* – low, младший):

<i>AX</i> :	<i>AH</i>	<i>AL</i>
<i>BX</i> :	<i>BH</i>	<i>BL</i>
<i>CX</i> :	<i>CH</i>	<i>CL</i>
<i>DX</i> :	<i>DH</i>	<i>DL</i>

Биты:	158	7 0
-------	-----	-----

Таким образом, с каждым из этих регистров можно работать как с единым целым, а можно работать и с его "половинками". Например, можно записать слово в *AX*, а затем считать только часть слова из регистра *AH* или заменить только часть в регистре *AL* и т.д. Такое устройство регистров позволяет использовать их для работы и с числами, и с символами.

Все остальные регистры на "половинки" не делятся, поэтому считать или записать их содержимое (16 битов) можно только целиком.

Сегментные регистры *CS*, *DS*, *SS* и *ES* используются при формировании адресов команд и данных, расположенных в оперативной памяти. Они не могут быть операндами никаких команд, кроме команд пересылки и стековых команд.

Счетчик команд *IP* всегда содержит адрес (смещение от начала программы) той команды, которая должна быть выполнена следующей (начало программы хранится в регистре *CS*). Содержимое регистра *IP* можно изменить только командами перехода.

В процессоре имеется особый регистр флагов. Флаг – это бит, принимающий значение 1 ("флаг установлен"), если выполнено некоторое условие, и значение 0 ("флаг сброшен") в противном случае. В процессоре Intel 8086 используется 9 флагов, каждому из них присвоено определенное имя (*ZF*, *CF* и т.д.). Все они собраны в 16-разрядном регистре флагов (каждый флаг – это один из разрядов регистра, часть его разрядов не используется). Формат регистра флагов, отражающий распределение отдельных флагов по битам этого регистра, приведен на рис. 1.

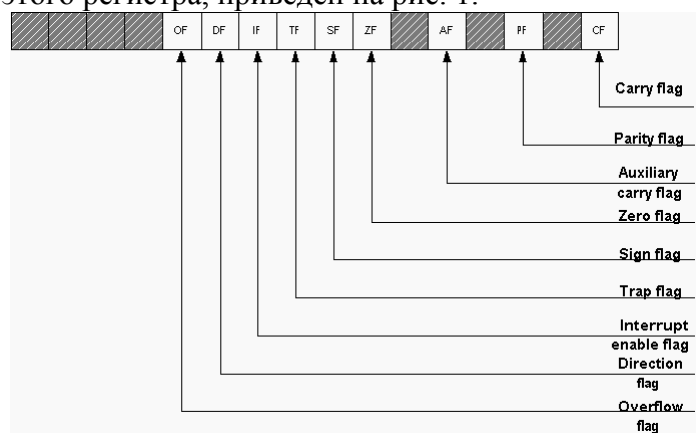


Рис. 1. Формат регистра флагов

Непосредственного доступа как к отдельным флагам, так и к регистру флагов в целом центральный процессор не предоставляет.

Некоторые флаги принято называть флагами условий; они автоматически меняются при выполнении команд и фиксируют те или иные свойства их результата (например, равен ли он нулю). Другие флаги называются флагами состояний; они меняются из программы и оказывают влияние на дальнейшее поведение процессора (например, блокируют прерывания).

К флагам условий относят:

- *CF* (carryflag) – флаг переноса;
- *OF* (overflowflag) – флаг переполнения;
- *ZF* (zeroflag) – флаг нуля;
- *SF* (sign flag) – флагзнака;
- *PF* (parity flag) – флагчетности;
- *AF* (auxiliarycarryflag) – флаг дополнительного переноса.

Флаг переноса (*CF*) принимает значение 1, если при сложении целых чисел появилась единица переноса, не "влезаящая" в разрядную сетку, или если при вычитании чисел без знака первое из них было меньше второго. В командах сдвига в *CF* заносится

бит, вышедший за разрядную сетку. *CF* фиксирует также особенности команды умножения.

Флаг переполнения (*OF*) устанавливается в 1, если при сложении или вычитании целых чисел со знаком получился результат, по модулю превосходящий допустимую величину (произошло переполнение мантиссы и она "залезла" в знаковый разряд).

Флаг нуля (*ZF*) устанавливается в 1, если результат команды оказался равным 0.

Флаг знака (*SF*) устанавливается в 1, если в операции над знаковыми числами получился отрицательный результат.

Флаг четности (*PF*) равен 1, если результат очередной команды содержит четное количество двоичных единиц. Учитывается обычно только при операциях ввода-вывода.

Флаг дополнительного переноса (*AF*) фиксирует особенности выполнения операций над двоично-десятичными числами.

К флагам состояний относят:

- *DF* (direction flag) – флаг направления;
- *IF* (interruptenableflag) – флаг разрешения прерываний;
- *TF* (trap flag) – флаг трассировки.

Флаг направления (*DF*) устанавливает направление просмотра областей памяти в цепочечных командах: при *DF* = 0 области просматриваются "вперед" (от начала к концу), при *DF* = 1 – в обратном направлении.

Флаг разрешения прерываний (*IF*): при *IF* = 0 процессор перестает реагировать на поступающие к нему прерывания, при *IF* = 1 блокировка прерываний снимается.

Флаг трассировки (*TF*): при *TF* = 1 после выполнения каждой команды процессор делает прерывание (с номером 1), чем можно воспользоваться при отладке программы для ее трассировки.

Арифметико-логическое устройство предназначено для организации переработки информации внутри центрального процессора. Оно позволяет выполнять арифметические и логические операции над данными, а также сдвиги. К арифметическим операциям относятся: сложение, вычитание, умножение и деление. В состав логических операций входят: отрицание, операция "И", операция "ИЛИ" и операция "ИСКЛЮЧАЮЩЕЕ ИЛИ".

Кэш-память является "посредником" между процессором и оперативной памятью. В ней хранятся наиболее часто используемые участки оперативной памяти. Время доступа процессора к кэш-памяти в несколько раз меньше, чем к обычной памяти. За счет этого среднее время доступа к памяти значительно сокращается.

1. Ознакомиться с теоретическим материалом.
2. Дать определение архитектуры ЭВМ.
3. Описать состав и назначение основных блоков, составляющих структурную схему ЭВМ.
4. Описать регистровую структуру процессора Intel 8086 и назначение регистров.
5. Описать механизмы организации оперативной памяти.
6. Описать способы представления данных в ЭВМ.

Практическое занятие № 42 «Исследование сигналов и построение временной диаграммы работы процессора Intel.»

Цель занятия: Исследование сигналов и построение временной диаграммы работы процессора Intel.

Содержание работы:

Примем, что в нашем распоряжении имеются 9 Kb оперативной памяти и 5 внешних устройств. Характеристики заданий приведены в таблице 1.

Таблица 1. Характеристики заданий.

Номер	Время	Необходимое	Требования к	Требования к
-------	-------	-------------	--------------	--------------

задания	поступления	процессорное время	ОП	ВУ

Время ввода каждого задания определим по формуле:

$$t_{\text{ввода}} = q \cdot N, \text{ где}$$

q – время работы одного внешнего устройства ($q=5$ ед.времени), N – число внешних устройств.

Таким образом, если бы задания выполнялись по одному, то на каждое задание было бы затрачено время

$$T = t_{\text{ввода}} + t_{\text{проц.}}, \text{ где}$$

T – время выполнения задания,

$t_{\text{проц.}}$ – процессорное время, затраченное для выполнения задания.

В ЭВМ поступает поток заданий, поэтому при одновременном выполнении нескольких заданий процессорное время распределяется между ними равномерно. Т.о. реально затраченное время для выполнения задания \geq необходимого процессорного времени.

Для простоты предположим, что все внешние устройства могут работать параллельно, не задерживая друг друга.

Процесс построения временной диаграммы рассмотрен в таблице 2.

Таблица 2. Описание построения временной диаграммы.

Время	Событие
$t = 0$	Поступило задание 1. Т.к. свободных ресурсов (ОП и ВУ) заданию хватает, оно назначается на выполнение. (Начинается ввод).
$t = 5$	Ввод завершен и все процессорное время отдается для выполнения задания 1.
$t = 7$	Поступило задание 2. Свободных ресурсов для выполнения задания недостаточно, поэтому оно помещается в очередь.
$t = 12$	Поступило задание 3. Свободных ресурсов для выполнения задания недостаточно, поэтому оно также помещается в очередь. В очереди сейчас находятся задания 2 и 3.
$t = 20$	Поступило задание 4. Т.к. свободных ресурсов заданию хватает, оно назначается на выполнение. Первые 5 ед. времени процессорное время заданию 4 не требуется и его полностью получает задание 1.
$t = 25$	Завершился ввод задания 4. С этого момента времени процессор необходим обоим заданиям. Процессор начинает работать в мультипрограммном режиме. Каждое из заданий получает 10 ед. процессорного времени. В сумме процессорного времени затрачено 20 ед.
$t = 45$	Завершено выполнение задания 1. Ресурсы, занятые им, освобождены. Теперь свободных ресурсов достаточно для назначения на выполнение задания 2 или 3 (но не одновременно). В действие вступает дисциплина обслуживания FIFO. На выполнение назначается задание 2 – оно поступило первым (задание 3 остается в очереди).
$t = 50$	Те 5 ед. времени, в течение которых выполнялся ввод задания 2, все процессорное время (5 ед.) получило задание 4. Вновь процессор начинает работать в мультипрограммном режиме. 10 ед. процессорного времени распределяются между заданиями 4 и 2.
$t = 60$	Выполнение задания 4 завершено, но освободившихся ресурсов недостаточно для

	назначения на выполнение задания 3. Задание 3 остается в очереди.
t = 65	Завершилось выполнение задания 2. освободившихся ресурсов достаточно для назначения на выполнение задания 3. Задание 3 выполняется в однопрограммном режиме.
t = 85	Завершилось выполнение задания 3.

Каким образом можно проверить правильность построения временной диаграммы?

Для этого необходимо определить по временной диаграмме, сколько времени работал процессор. Это временные интервалы с t=5 до t=65 и с t=75 до t=85. В интервале с t=65 до t=75 процессор простаивал. Таким образом, процессор работал в течение 70 ед. времени.

Далее необходимо просуммировать процессорное время всех выполненных заданий.

$$T_{\text{процессора}} = 30 + 10 + 10 + 20 = 70 \text{ ед.}$$

Значения равны, следовательно, диаграмма построена верно.

Для оценки эффективности дисциплины обслуживания найдем средневзвешенное время обращения. Согласно формуле . Необходимо найти взвешенные времена выполнения каждого задания (данные берем из временной диаграммы и таблицы):

$$1. W_1 = (t_3 - t_{\Pi}) / T = (t_3 - t_{\Pi}) / (t_{\text{ввода}} + t_{\text{проц}}) = (t_3 - t_{\Pi}) / (q \text{ Ч } N + t_{\text{проц}}) = (45 - 0) / (5 \text{ Ч } 1 + 15) = 1.00$$

$$2. W_2 = (65 - 7) / (5 \text{ Ч } 1 + 15) = 2.90$$

$$3. W_3 = (82 - 12) / (5 \text{ Ч } 2 + 10) = 3.65$$

$$4. W_4 = (60 - 20) / (5 \text{ Ч } 1 + 35) = 1.00$$

$$T.o. W_{\text{ср}} = (W_1 + W_2 + W_3 + W_4) / 4 = 2.14.$$

Практическое занятие № 43 «Изучение внутреннего устройства и программное управление стандартным параллельным портом РС»

Цель занятия: Изучение внутреннего устройства и программное управление стандартным параллельным портом РС.

Содержание работы:

Задание

1) Заполните таблицу компонентов ПК

Компоненты	Описание
Системная плата	
Процессор	
Оперативная память	
Корпус	
Источники питания	
Накопитель на жестких дисках	
Накопитель CD-ROM/DVD-ROM	
Клавиатура	
Мышь	
Видеоадаптер	
Монитор	
Звуковая плата	
Модем	

2) Заполните таблицу в соответствии с теоретическим материалом

Таблица Разъемы подключения

Разъем	Тип разъема	Характеристика	Примечания
Питание системного блока			
Питание монитора			
Параллельный порт			
Последовательный порт			
Mouse			
Keyboard			
USB			
LAN			

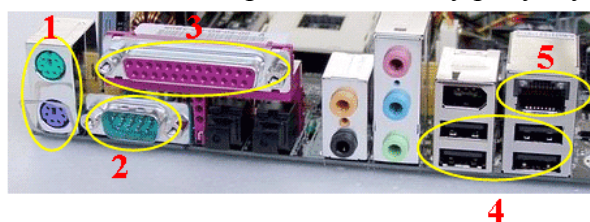
3) Заполните таблицу.

	COM-порт	LPT-порт	USB	IEEE1394	IrDA	Bluetooth
Параллельный или последовательный интерфейс						
Максимальная пропускная способность						
Подключаемые устройства						
Количество одновременно подключаемых устройств						
Проводной или беспроводной интерфейс						

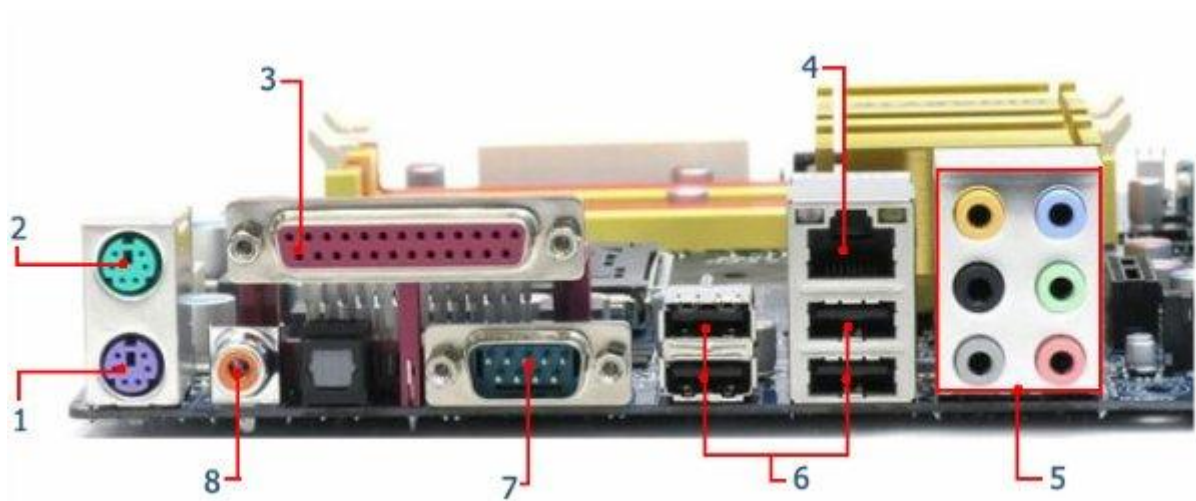
4) К каким интерфейсам ПК относятся разъемы, представленные на этих рисунках?



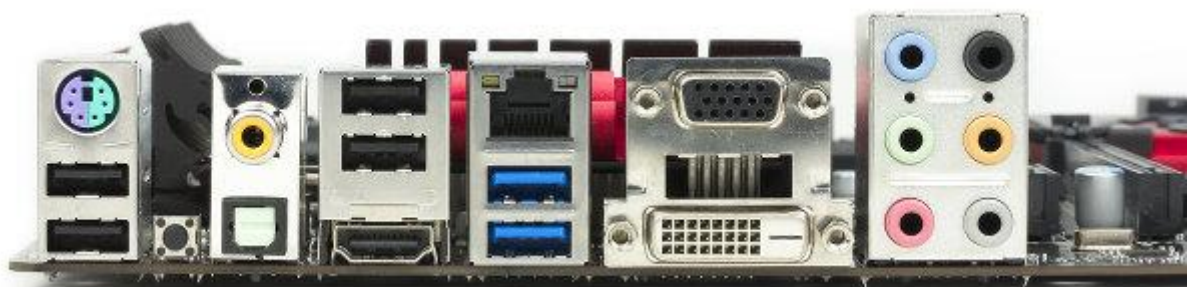
5) По представленному рисунку назовите интерфейс подключения.



а)



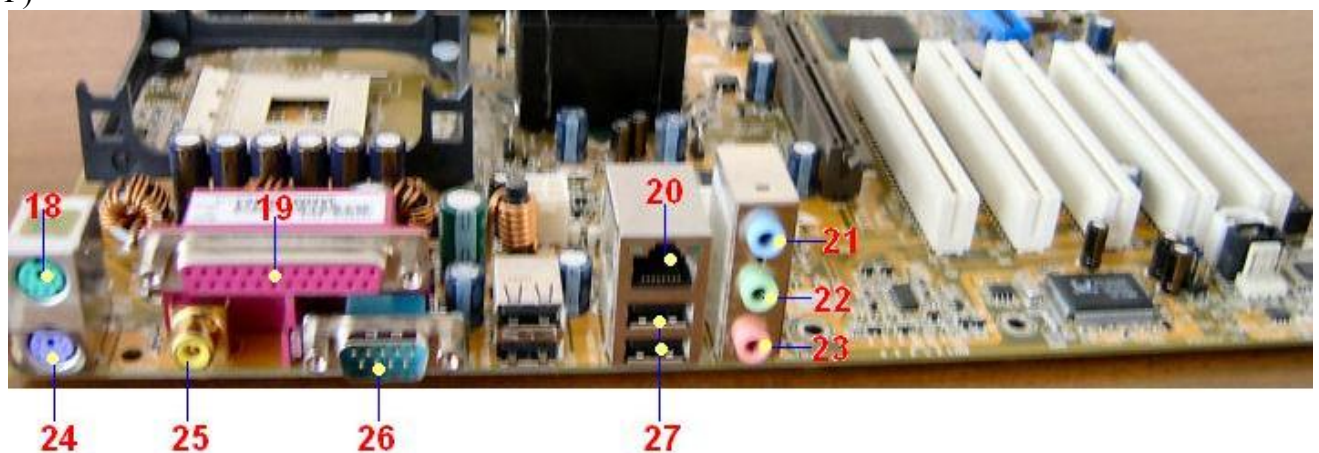
б)



в)



г)



6) Самостоятельно, используя Интернет-ресурсы заполните таблицу

Тип разъема	Характеристика	Примечания(скорость передачи, использование)
-------------	----------------	--

VGA		
SPP (Standard Parallel Port)		
USB		
EPP (Enhanced Parallel Port)		
ECP (Enhanced Capability Port)		
Line Out		
Line In		
Com		

Практическое занятие № 44 «Подключение дополнительного оборудования и настройка связи между элементами компьютерной системы»

Цель занятия: изучить основные блоки и периферийные устройства персонального компьютера, способы их соединения, конструктивы (разъемы), основные характеристики (название, тип разъема, количество контактов, скорость передачи данных, дополнительные свойства); научиться определять по внешнему виду типы разъемов, подключаемое к ним оборудование, знать основные устройства персонального компьютера, их назначение и основные характеристики; научиться определять компоненты системного блока по внешнему виду, уяснить порядок и способы их соединения.

Содержание работы:

В основу устройства компьютера положен **принцип открытой архитектуры**, т.е. возможность подключения к системе дополнительных независимо разработанных устройств для различных прикладных применений. Все устройства подключаются к системе и взаимодействуют друг с другом через **общую шину**.

Минимальный набор аппаратных средств, без которых невозможен запуск, и работа ПК определяет его базовую конфигурацию. В базовую конфигурацию ПК входят: системный блок, монитор, клавиатура и ручной манипулятор - мышь. Включение ручного манипулятора в базовую конфигурацию обусловлено тем, что работа в современных графических операционных системах без этого устройства возможна, но крайне затруднительна.



Рисунок 1.1.

Системный блок. Системный блок является центральной частью ПК. В корпусе системного блока размещены внутренние устройства ПК.

Системные блоки ПК имеют различные дополнительные элементы (вентилятор, динамик) и конструктивные особенности, обусловленные назначением и условиями эксплуатации ПК. Обязательным узлом системного блока является блок питания, который преобразует поступающий из сети переменный ток напряжением 220В в постоянный - 3.3В, -5В и -12В для электропитания всех внутренних устройств компьютера. Основным параметром блока питания, учитываемым при сборке требуемой конфигурации ПК, является его мощность. Питание монитора также возможно через блок питания системного блока.



Рисунок 1.2.

По внешнему виду системные блоки отличаются формой корпуса (рис. 1.3). Наиболее распространенными на сегодняшний день являются системные блоки форм-фактора ATX (на следующем практическом занятии рассмотрим особенности конструкции системных блоков нового перспективного форм-фактора - BTX).



Рисунок 1.3.

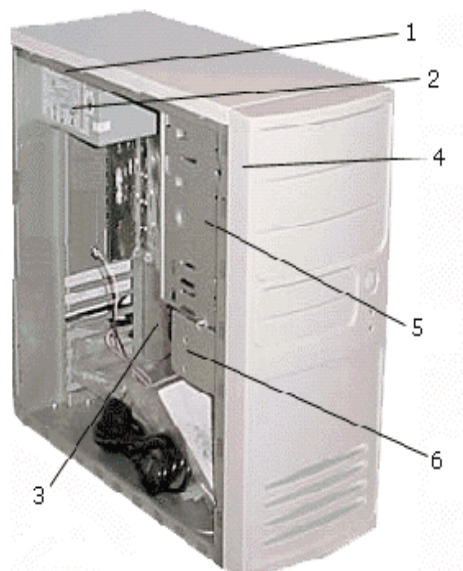
Основой корпуса (рис. 1.4) системного блока является каркас (1), к которому крепятся: блок питания (2), панель крепления материнской платы (3), передняя панель (4), а также секции для дисководов размером 5,25- (5) и 3,5- (6). Оба типа секций можно использовать для накопителей на жестких магнитных дисках.

В состав системного блока входят следующие аппаратные средства ПК:

1. Системная (материнская) плата с микропроцессором.
1. Оперативная память.
1. Накопитель на жестком магнитном диске.
1. Контроллеры или адаптеры для подключения и управления внешними устройствами ПК (монитор, звуковые колонки и др.).
1. Порты для подключения внешних устройств (принтер, мышь и др.).
1. Внешние запоминающие устройства для гибких магнитных дисков и лазерных дисков CD и DVD.



Системный блок Desktop



Системный блок MiniTower

Рисунок 1.4.

Если открыть корпус системного блока, то можно увидеть большую плату, на которой размещаются микросхемы, электронные устройства и разъемы (слоты). В разъемы материнской платы вставлены платы меньшего размера, к которым, посредством кабелей, подключены периферийные устройства. Это и есть системная плата (рис. 1.5).



Рисунок 1.5.

На системной плате помимо процессора расположены (рис. 1.6):

1. **Чипсет** (микропроцессорный комплект) - набор микросхем, которые управляют работой внутренних устройств ПК и определяют основные функциональные возможности материнской платы.

1. **Шины** - набор проводников, по которым происходит обмен сигналами между внутренними устройствами компьютера.

1. **Оперативная память** - набор микросхем, предназначенных для временного сохранения данных, пока включен компьютер.

1. **Постоянное запоминающее устройство** - микросхема, предназначенная для долговременного хранения данных, даже при отключенном компьютере.

1. **Разъемы (слоты)** для подсоединения дополнительных устройств.

Основные элементы системной платы показаны на рис. 1.6, где цифрами обозначены:

1. Разъем для микропроцессора.
2. Слоты для модулей оперативной памяти.
3. Интерфейсы шины PCI.
4. Микросхема системной логики (чипсет, 4.1 - северный мост, а 4.2 - южный мост).
5. Интерфейсы для подключения жестких дисков.
6. Блок портов ввода/вывода.
7. Интерфейс шины AGP для подключения видеоадаптера.

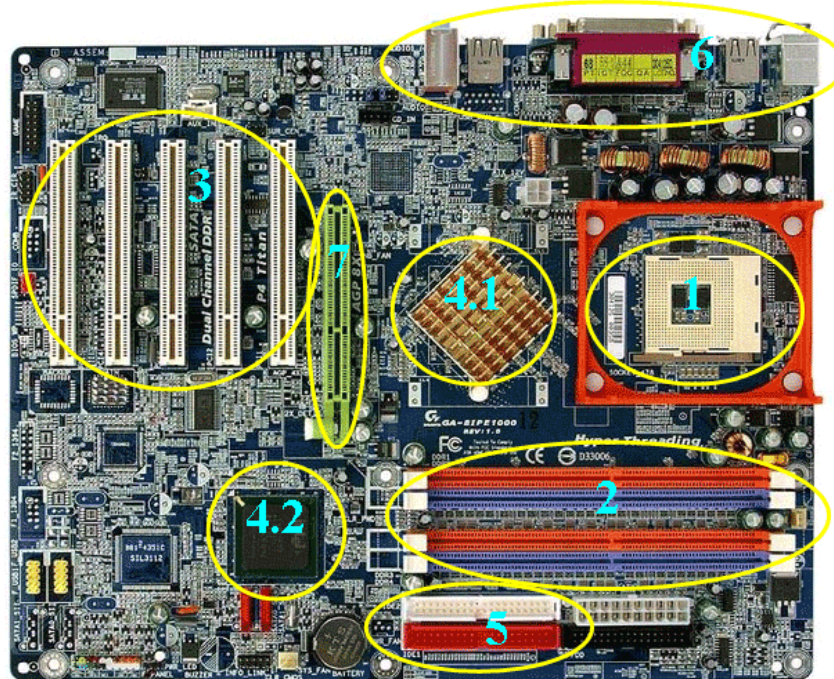


Рисунок 1.6.

Интерфейсы ПК. В общем случае под стандартным интерфейсом понимается совокупность унифицированных аппаратных, программных и конструктивных средств, необходимых для реализации взаимодействия различных функциональных компонентов в системах. Применительно к персональным компьютерам к стандартным интерфейсам относятся все порты ввода/вывода, различные слоты расширения системной платы (PCI, AGP) и другие разъемы, используемые для подключения различных устройств в единое целое.

Рассмотрим набор и внешний вид интерфейсов, размещенных на задней стенке системного блока (рис. 1.7). Все эти интерфейсы предназначены для подключения периферийных устройств к персональному компьютеру.

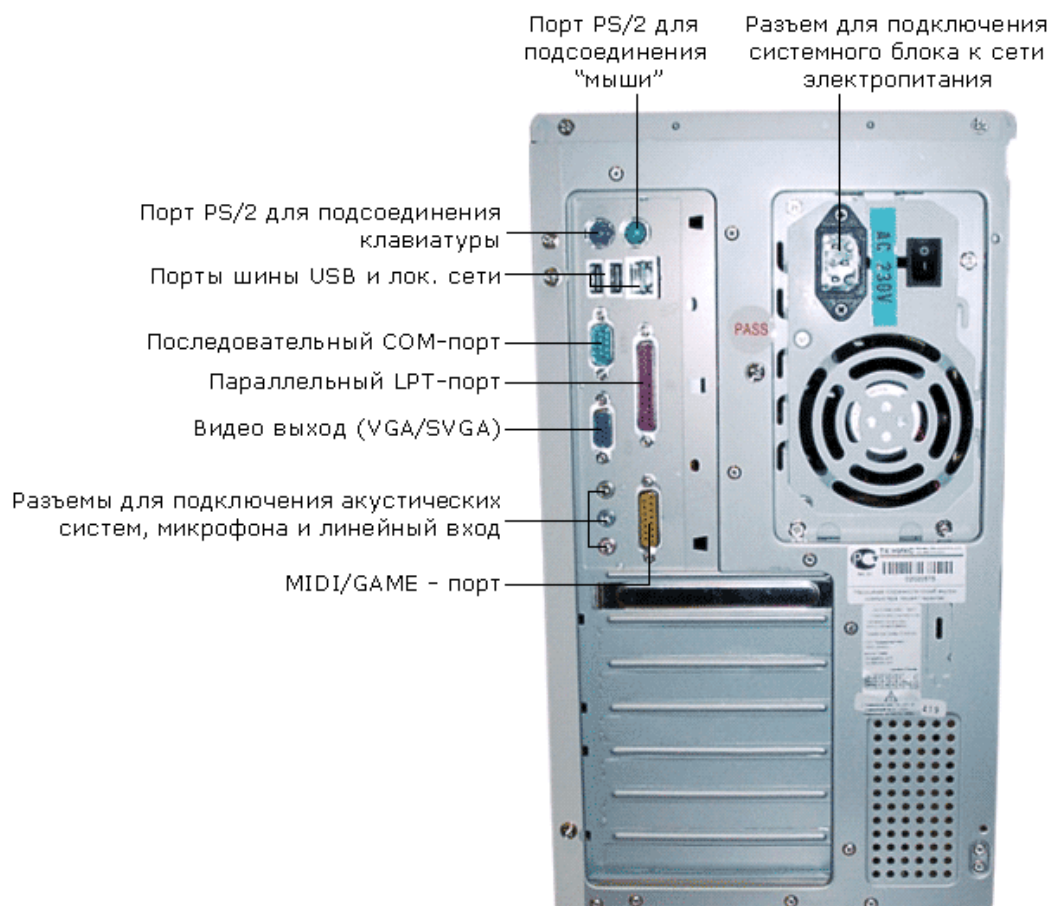
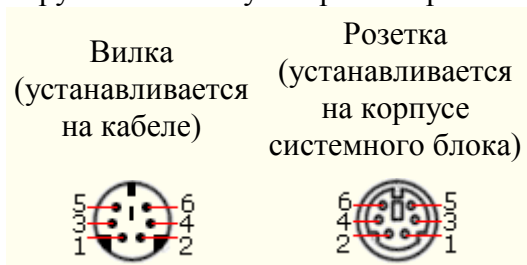
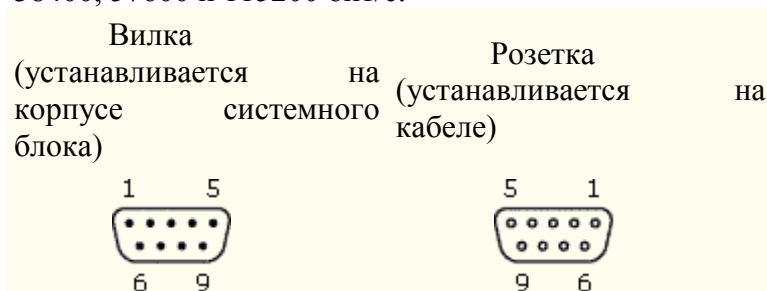


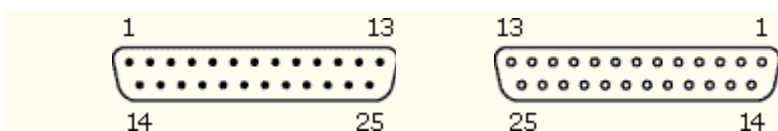
Рисунок 1.7.

Порт PS/2 - шестиконтактный разъем, используемый для подключения клавиатуры и ручного манипулятора. Эти разъемы подключены к единому контроллеру.

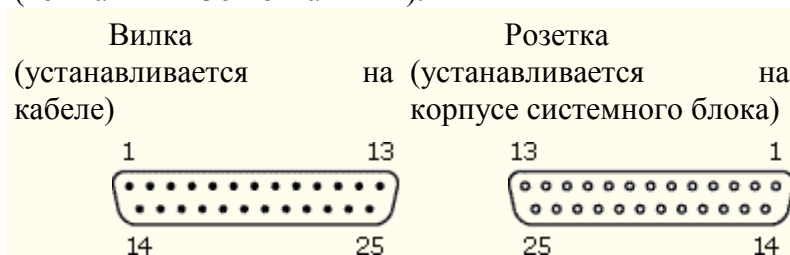


Последовательный COM-порт (RS-232) - данный порт используется для подключения модема. Ранее использовался и для подключения ручного манипулятора ("мыши"). Порт стандартизован в двух вариантах 9 (DB9) и 25-контактный (DB25). Последний вариант практически не реализуется в современных системных блоках. Для асинхронного режима принято несколько стандартных скоростей обмена: 9600, 19200, 38400, 57600 и 115200 бит/с.





Параллельный порт (LPT) - этот порт изначально разрабатывался как интерфейс для подключения принтера. Также может быть использован для подключения сканера или плоттера, имеющего соответствующий интерфейс. Скорость обмена не выше 150 Кбайт/с при значительной загрузке процессора. В 1994 г. был принят стандарт IEEE1284, определивший спецификацию портов SPP, EPP и ECP. Дополнительные режимы EPP (Enhanced Parallel Port - улучшенный параллельный порт) и ECP (Extended Capability Port - порт с расширенными возможностями) позволили ввести поддержку двунаправленного обмена с аппаратным сжатием данных (устанавливается программой Setup BIOS). В качестве разъемов спецификацией определены Тип А (DB-25), Тип В (Centronics) и тип С (компактный 36-контактный).



Видеовыход (15-контактный разъем) - используется для подключения VGA/SVGA монитора к системному блоку, а именно, к видеоадаптеру. В случае интегрированного в системную плату видеоадаптера видеовыход размещается на стандартной панели, как показано на рис. 1.7.

Разъем для подключения к локальной сети (RJ-45) - восьмиконтактный интерфейс для подключения компьютера к локальной сети. В случае интегрированного в системную плату сетевого адаптера интерфейс RJ-45 размещается на стандартной панели интерфейсов (как на рис. 1.7). Другой вариант - размещается на установленном сетевом адаптере.

MIDI/GAME порт - используется для подключения мультимедийных игровых устройств, например, синтезатора и игрового манипулятора "джойстика".

В архитектуре современных персональных компьютеров все большее значение приобретают внешние шины, служащие для подключения различных устройств, таких как внешние накопители flash-памяти и накопители на жестких магнитных дисках, CD/DVD-устройства, сканеры, принтеры, цифровые камеры и др. Основными требованиями к таким шинам и их интерфейсам заключаются в высоком быстродействии, компактности интерфейса и удобстве коммутации устройств пользователем.

В современных ПК к таким внешним шинам и интерфейсам относятся: USB, FireWire, IrDA, Bluetooth. Последние два интерфейса относятся к классу беспроводных интерфейсов.

Шина и интерфейс USB. Архитектура шины USB представляет собой классическую топологию "звезда" с последовательной передачей данных, в соответствии с которой в системе должен быть корневой (ведущий) концентратор USB, к которому подключаются периферийные концентраторы USB (рис. 1.8, внешний концентратор на 4 порта USB), а непосредственно к ним подключаются периферийные устройства с интерфейсом USB. Периферийные концентраторы могут подключаться друг к другу, образуя каскады.



Рисунок 1.8.

Корневой концентратор расположен в одной из микросхем системной логики (как правило, это южный мост чипсета). Всего через один корневой концентратор USB может быть подключено до 127 устройств (концентраторов и устройств USB). Однако, учитывая относительно невысокую пропускную способность шины USB версии 1.1 (до 12 Мбит/с), что с учетом служебных расходов составляет 1 Мбайт/с, оптимальным является подключение 4-5 низкоскоростных устройств (клавиатура, манипулятор, сканер).

Проблема низкой пропускной способности частично решена версией интерфейса USB 2.0, в соответствии с которой пиковая пропускная способность увеличена до 480 Мбит/с (60 Мбайт/с). Этого вполне достаточно для работы типичных современных USB-устройств: принтеров, офисных сканеров, цифровых фотокамер, джойстиков и др. (более скоростные устройства должны подключаться ближе к корневому концентратору).

Все устройства USB соединяются между собой четырехжильным кабелем (рис. 1.9).

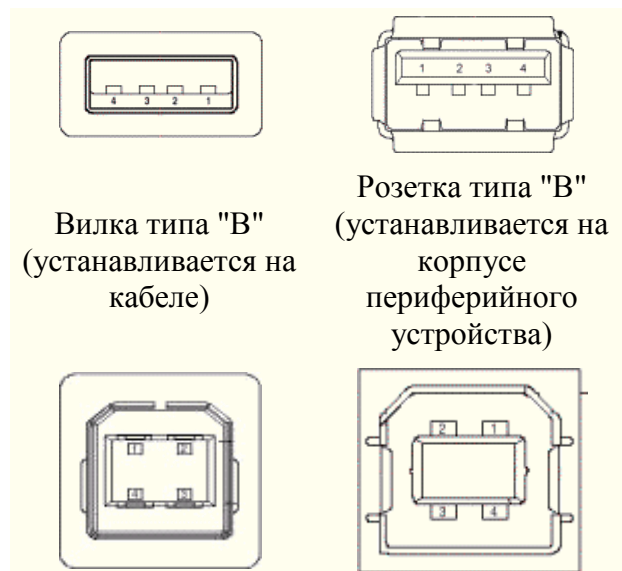


Рисунок 1.9.

По одной паре передаются данные, по другой - электропитание, которое автоматически подключается устройством при необходимости. На концах кабеля монтируются разъемы типов "А" и "В". С помощью разъема "А" устройство подключают к концентратору. Разъем типа "В" устанавливают на концентраторы для связи с другим концентратором и на устройства, от которых кабель должен отключаться (например, сканеры).

Вилка типа "А"
(устанавливается на
кабеле)

Розетка типа "А"
(устанавливается на
корпусе системного
блока)



Вилка типа "B"
(устанавливается на
кабеле)

Розетка типа "B"
(устанавливается на
корпусе
периферийного
устройства)

Спецификация USB определяет две части интерфейса: внутреннюю и внешнюю. Внутренняя часть делится на аппаратную (собственно корневой концентратор и контроллер USB) и программную (драйверы контроллера, шины, концентратора, клиентов). Внешнюю часть представляют устройства (концентраторы и компоненты) USB. Для обеспечения корректной работы все устройства делятся на классы: принтеры, сканеры, накопители и т. д. Разделение устройств на классы происходит не по их целевому назначению, а по единому способу взаимодействия с шиной USB. Поэтому драйвер класса принтеров определяет не его разрешение или цветность, а способ передачи (односторонний или двунаправленный) данных, порядок инициализации при подключении. Также спецификация USB предусматривает интерфейс mini-USB.

В интерфейсе USB реализована процедура подключения периферии к шине "в горячем режиме", т.е. без отключения питания системного блока. Подключенное в свободный порт устройство вызывает перепад напряжения в цепи. Контроллер немедленно направляет запрос на этот порт. Присоединенное устройство принимает запрос и посылает пакет с данными о классе устройства, после чего устройству присваивается уникальный идентификационный номер. Далее происходит автоматическая загрузка и активация драйвера устройства, его конфигурирование и, тем самым, окончательное подключение устройства. Точно так же происходит инициализация уже подсоединенного и включаемого в сеть устройства (например, модема).

Интерфейс IEEE1394 (FireWire). Конкурентом интерфейса USB 2.0 на сегодняшний день является последовательный цифровой интерфейс FireWire, называемый также IEEE1394 (iLink - торговая марка Sony). Этот интерфейс, рассматривающийся по началу как скоростной вариант интерфейса SCSI, был предложен компанией Apple. В начале 90-х годов вышло техническое описание этого интерфейса в виде стандарта IEEE 1394 (Institute of Electrical and Electronic Engineers - института инженеров по электротехнике и электронике).

Спецификация интерфейса IEEE1394 предусматривает последовательную передачу данных со скоростями 100, 200, 400, 800 Мбит/с (последнее значение не стандартизировано). Выбор последовательного интерфейса обусловлен необходимостью связать удаленные внешние устройства, работающие с различными скоростями. В этом случае обеспечивается их работа по одной линии, отсутствие громоздких кабелей и шлейфов, габаритных разъемов. Появление последовательных интерфейсов IEEE1394 и USB привело к вытеснению параллельных интерфейсов для подключения внешних устройств.

Топология интерфейса IEEE1394 "древовидная", при этом система адресации обеспечивает подключение до 63 устройств в одной сети. Для связи между сетями

существуют мосты, для объединения ветвей в один узел - концентраторы. Повторители служат для усиления сигналов при длине соединения более 4.5 метров. Всего может быть связано до 1024 сетей по 63 устройства в каждой. Все устройства IEEE1394 соединяются между собой шестижильным экранированным кабелем, имеющим две пары сигнальных и пару питающих проводников. Подключение осуществляется с помощью стандартной пары "вилка - розетка" (рис. 1.10, рис. 1.11). Корневое устройство интерфейса выполняет функции управления шиной. Первоначально такие устройства разрабатывались в виде плат расширения (рис. 1.11), в дальнейшем поддержка IEEE1394 стала реализовываться в наборе системной логики (чипсете) системной платы.



Рисунок 1.10.

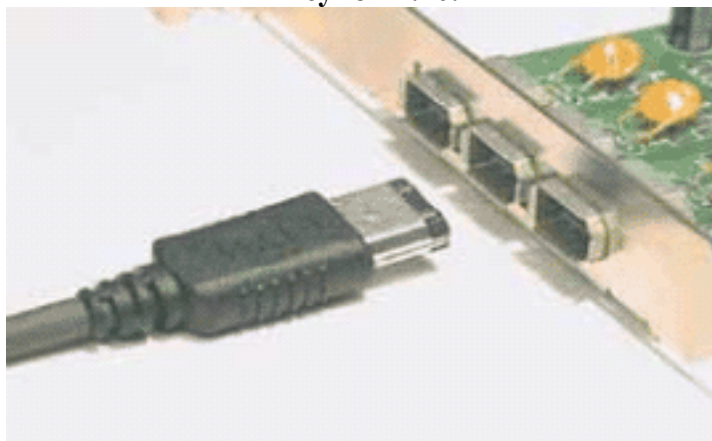


Рисунок 1.11.

Автоматическая конфигурация интерфейса IEEE1394 происходит после включения питания, отсоединения или подключения устройства. При изменении конфигурации подается сигнал сброса и производится новая идентификация дерева.

Как и USB, шина IEEE 1394 обеспечивает возможность переконфигурации аппаратных средств компьютера без его выключения. В соответствии с принятым стандартом IEEE1394 существует два варианта разъемов и кабелей (рис. 1.12).

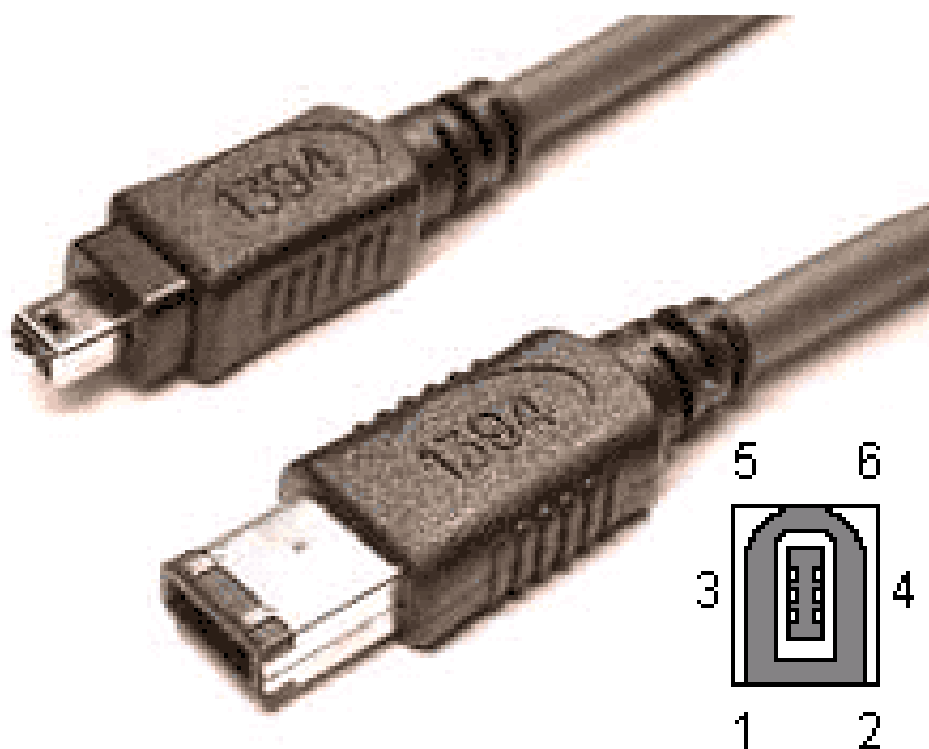


Рисунок 1.12.

Первый вариант с 6-контактным разъемом IEEE1394 предусматривает не только передачу данных, но и подачу электропитания на подключенные к соответствующему контроллеру ПК устройства IEEE1394. При этом общий ток ограничен величиной 1.5 А.

Второй вариант с 4-контактным разъемом IEEE1394 рассчитан только на передачу данных. В этом случае подключаемые устройства должны иметь автономные источники питания. Интерфейс IEEE1394, используемый для подключения различного видео и аудио оборудования (телевизоры, видеомагнитофоны, видеокамеры и т.д.), осуществляющего передачу данных в цифровом коде, широко известен под названием iLink (торговая марка Sony).

Инфракрасный интерфейс IrDA (Infrared Data Association). IrDA относится к категории беспроводных (wireless) внешних интерфейсов, однако, в отличие от радио-интерфейсов, канал передачи информации создается с помощью оптических устройств. Инфракрасный (ИК) открытый оптический канал является самым недорогим и удобным интерфейсом передачи данных на небольшие расстояния (до нескольких десятков метров) среди других беспроводных линий передачи информации.

Технически интерфейс IrDA основан на архитектуре коммуникационного COM-порта ПК, который использует универсальный асинхронный приемопередатчик и работает со скоростью передачи данных 2400-115200 бит/с. В IrDA реализован полудуплексный режим передачи данных, т.е. прием и передача данных происходит по очереди.

Первым вариантом интерфейса IrDA стал стандарт Serial Infrared standart (SIR). Этот стандарт обеспечивает передачу данных со скоростью 115.2 Кбит/с. В 1994 году IrDA была опубликована спецификация на общий стандарт, получивший название IrDA-standart, который включал в себя описание Serial Infrared Link (последовательная инфракрасная линия связи), Link Access Protocol (IrLAP) (протокол доступа) и Link Management Protocol (IrLMP) (протокол управления). С 1995 года компания Microsoft включила поддержку интерфейса IrDA-standart в стандартный пакет операционной системы Windows 95. В настоящее время IrDA-standart? самый распространенный стандарт для организации передачи информации по открытому инфракрасному каналу.

На рис. 1.13 показан интерфейс IrDA, подключаемый к системному блоку через USB порт. В мобильных устройствах такой интерфейс встраивается, как правило, на лицевой стороне корпуса.

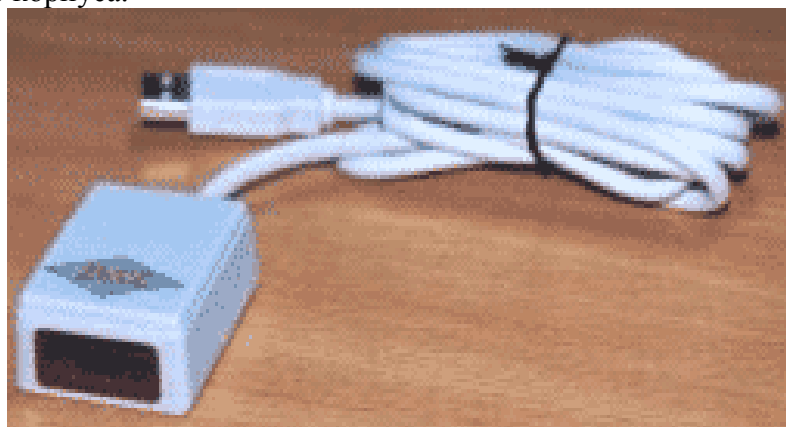


Рисунок 1.13.

Интерфейс Bluetooth относится к перспективным беспроводным интерфейсам передачи данных. Этот интерфейс активно разрабатывается и продвигается консорциумом Bluetooth Special Interest Group (Bluetooth SIG).

Технология Bluetooth разрабатывалась для построения беспроводных персональных сетей (WPAN, Wireless Personal Area Network). В 2001 году был принят стандарт IEEE 802.15.1, описывающий технологию построения таких сетей, а в 2002 году технология получила развитие в стандарте IEEE 802.15.3 (протокол связи для беспроводных частных сетей).

Единичная Bluetooth-система состоит из модуля, обеспечивающего радиосвязь, и присоединенного к нему хоста, в качестве которого может выступать компьютер или любое периферийное устройство. Bluetooth-модули обычно встраивают в устройство, подключают через доступный порт либо PC-карту. Модуль состоит из менеджера соединений (link manager), контроллера соединений и приемопередатчика с антенной. Два связанных по радио модуля образуют пиконет (piconet). Причем один из модулей играет роль ведущего (master), второй - ведомого (slave). В пиконете не может быть больше восьми модулей, поскольку адрес активного участника пиконета, используемый для идентификации, является трехбитным (уникальный адрес присваивается семи ведомым модулям, ведущий модуль не имеет адреса, а нулевой адрес зарезервирован для широковещательных (broadcast) сообщений).

Оптимальный радиус действия модуля - до 10 м (в настоящее время удалось увеличить дальность связи до 100 метров при работе вне помещений). Диапазон рабочих частот 2.402-2.483 ГГц. Коммуникационный канал Bluetooth имеет пиковую пропускную способность 721 Кбит/с. Для уменьшения потерь и обеспечения совместимости пиконетов частота в Bluetooth перестраивается скачкообразно (1600 скачков/с). Канал разделен на временные слоты (интервалы) длиной 625 мс (время между скачками), в каждый из них устройство может передавать информационный пакет. Для полнодуплексной передачи используется схема TDD (Time-Division Duplex, дуплексный режим с разделением времени). По четным значениям таймер передает ведущее устройство данных, а по нечетным - ведомое устройство.

Задания:

1. Убедитесь в том, что компьютерная система обесточена (при необходимости, отключите систему от сети).
2. Разверните системный блок задней стенкой к себе.
3. По наличию или отсутствию разъемов USB установите форм-фактор материнской платы (при наличии разъемов USB - форм-фактор ATX, при их отсутствии - AT).
4. Установите местоположение и снимите характеристики следующих разъемов:
 - питания системного блока;

- питания монитора;
 - сигнального кабеля монитора;
 - клавиатуры;
 - последовательных портов (два разъема);
 - параллельного порта;
 - других разъемов.
5. Убедитесь в том, что все разъемы, выведенные на заднюю стенку системного блока, не взаимозаменяемы, то есть каждое базовое устройство подключается одним единственным способом.
6. Изучите способ подключения мыши.

Мышь может подключаться к разъему последовательного порта или к специальному порту PS/2, имеющему разъем круглой формы. Последний способ является более современным и удобным. В этом случае мышь имеет собственный выделенный порт, что исключает возможность ее конфликта с другими устройствами, подключаемыми к последовательным портам. Последние модели могут подключаться к клавиатуре через разъем интерфейса USB.

7. Заполните таблицу:

Разъем	Тип разъема	Количество контактов	Примечания

8. Определить наличие основных устройств персонального компьютера.
9. Установите местоположение блока питания, выясните мощность блока питания (указана на ярлыке).
10. Установите местоположение материнской платы.
11. Установите характер подключения материнской платы к блоку питания.

Для материнских плат в форм-факторе АТ подключение питания выполняется двумя разъемами. Обратите внимание на расположение проводников черного цвета - оно важно для правильной стыковки разъемов.

12. Установите местоположение жесткого диска.

Установите местоположение его разъема питания. Проследите направление шлейфа проводников, связывающего жесткий диск с материнской платой. Обратите внимание на местоположение проводника, окрашенного в красный цвет (на жестком диске он должен быть расположен рядом с разъемом питания).

13. Установите местоположения дисководов гибких дисков и дисковода CD-ROM.

Проследите направление их шлейфов проводников и обратите внимание на положение проводника, окрашенного в красный цвет, относительно разъема питания.

14. Установите местоположение платы видеоадаптера.

Определите тип интерфейса платы видеоадаптера.

15. При наличии прочих дополнительных устройств выявите их назначение, опишите характерные особенности данных устройств (типы разъемов, тип интерфейса и др.).

16. Заполните таблицу:

Устройство	Характерные особенности	Куда и при помощи чего подключается

Практическое занятие № 45 «Использование прямого доступа к памяти»

Цель занятия:

Содержание работы:

1. Закрепить понятие конечного автомата на примере разработки контроллера прямого доступа к памяти (ПДП).
2. Изучить применение контроллера ПДП (DMA).
3. Изучить принцип обмена данными с использованием DMA.
4. Изучить действие управляющих сигналов HOLD и HLDA, используемых для прямого доступа к памяти.
4. Изучить функции контроллера DMA 8237 при обмене данными в составе ВМ.

Задание к лабораторной работе

Варианты заданий формируются для заданного режима работы контроллера DMA.

№ варианта	Функции ПДП
1	Передача данных «память - память»
2	Передача данных «память - принтер»
3	Передача данных «у.ввода - память»
4	Передача данных «память — у.вывода»
5	Передача данных «память - дисплей»
6	Передача данных «у.ввода — у.вывода»

2. Приведите математическую модель контроллера DMA.
3. Приведите VHDL модель контроллера DMA.
4. Сформируйте тестовые сигналы.
5. Протестируйте контроллер DMA на макете FLEX8000.

Перечень используемого оборудования и программного обеспечения

1. Разработка VHDL модели цифровой системы должна проводиться в среде GHDL.
2. VHDL модель цифровой системы должна быть верифицирована в среде GHDL.
3. Для валидации VHDL модели цифровой системы необходимо использовать целевое устройство - лабораторный макет FLEX8000 (target).
4. Визуальный контроль за формой сигнала и оценки временных параметров проводится с использованием аналогового осциллографа С1-93.
5. Разработка VHDL модели цифровой системы должна проводиться на IBM совместимых ПК с использованием операционной системы WindowsXP (host PC).
6. Отчет по лабораторной работе должен выполняться в OpenOffice.

Практическое занятие № 46 «Использование дальних указателей при работе с видеопамью»

Цели занятия: научиться использование дальних указателей при работе с видеопамью.

Содержание работы:

Для работы может оказаться более удобным иметь указатель на видеопамью как на массив структур. Приведем пример для системы MS DOS:

```
#include <dos.h> /* там определено MK_FP */
char far *screen =
    MK_FP(0xB800 /*сегмент*/, 0x0000 /*смещение*/);
struct symb{
    char chr; char attr;
} far *scr, far *ptr;
#define COLS 80 /* число колонок */
#define LINES 25 /* число строк */
#define SCR(x,y) scr[(x) + COLS * (y)]
/* x из 0..79, y из 0..24 */
void main(){
    int x, y;
    char c;
```

```

scr = (struct symb far *) screen;
/* или сразу
* scr = (struct symb far *) MK_FP(0xB800, 0x0000);
*/
/* переписать строки экрана справа налево */
for(x=0; x < COLS/2; x++)
    for(y=0; y < LINES; y++){
        c = SCR(x,y).chr;
        SCR(x,y).chr = SCR(COLS-1-x, y).chr;
        SCR(COLS-1-x, y).chr = c;
    }
/* сделать цвет экрана: желтым по синему */
for(x=0; x < COLS; x++)
    for(y=0; y < LINES; y++)
        SCR(x,y).attr = (0xE | (0x1 << 4));
/* желтый + синий фон */
/* прочесть любую кнопку с клавиатуры (пауза) */
(void) getch();
}

```

И, наконец, еще удобнее работа с видеопамятью как с двумерным массивом структур:

```

#include <dos.h> /* MS DOS */
#define COLS 80
#define LINES 25
struct symb {
    char chr; char attr;
} (far *scr)[ COLS ] = MK_FP(0xB800, 0);
void main(void){
    register x, y;
    for(y=0; y < LINES; y++)
        for(x=0; x < COLS; ++x){
            scr[y][x].chr = '?';
            scr[y][x].attr = (y << 4) | (x & 0xF);
        }
    getch();
}

```

Учтите, что при работе с экраном через видеопамять, курсор не перемещается! Если в обычной работе с экраном текст выводится в позиции курсора и курсор **автоматически** продвигается, то здесь курсор будет оставаться на своем прежнем месте. Для перемещения курсора в нужное вам место, вы должны его поставить **явным образом** по окончании записи в видеопамять (например, обращаясь к портам видеоконтроллера). Обратите внимание, что спецификатор модели памяти *far* должен указываться перед КАЖДЫМ указателем (именно для иллюстрации этого в первом примере описан неиспользуемый указатель **ptr**).

Практическое занятие № 47 «Организация связи компьютера с внешними устройствами посредством последовательного интерфейса»

Цели занятия: Изучение организации порта последовательной передачи данных (COM-порта) и приобретение практических навыков его использования для ввода и вывода информации.

Содержание работы:

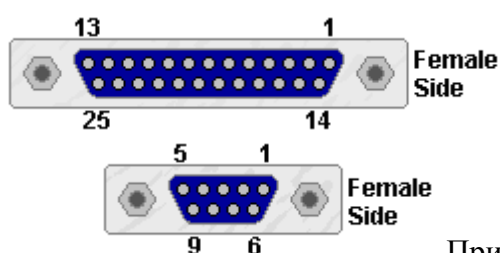
Последовательный порт или COM-порт (от англ. *communication*) — двунаправленный последовательный интерфейс, предназначенный для обмена байтовой

информацией. Порт называется последовательным потому, что информация через него передаётся по одной линии интерфейса бит за битом (в отличие от параллельного порта). Наиболее часто для последовательного порта персональных компьютеров используется стандарт *RS-232C*. Ранее последовательный порт использовался для подключения терминала, позже для модема или мыши. Сейчас он используется для соединения с источниками бесперебойного питания, для связи с аппаратными средствами разработки встраиваемых вычислительных систем.

Хотя некоторые другие интерфейсы компьютера — такие как *Ethernet*, *FireWire* и *USB* — также используют последовательный способ обмена, название «последовательный порт» закрепилось за портом, имеющим стандарт *RS-232C*.

Наиболее часто для реализации *COM*-интерфейса используются D-образные разъёмы: 9- и 25-контактные, (*DB-9* и *DB-25* соответственно). Раньше использовались также *DB-31* и круглые восьмиконтактные *DIN-8*. Максимальная скорость передачи обычно составляет 115200 бод.

Контакты разъемов и обозначение линий интерфейса приведены на рисунке:



Приведем более подробное описание сигналов

последовательного порта *DB-9*:

CD — установленный сигнал показывает, что модемом или другим устройством обнаружена несущая частота. Сигнал *CD#* является сигналом состояния модема, и центральный процессор может определить его состояние путем чтения бита *DCD* из регистра *MSR* для соответствующего последовательного порта. Бит *DCD* в регистре *MSR* показывает, изменился ли сигнал *CD#* со времени последнего чтения регистра *MSR*.

RXD — последовательные входы данных.

TXD — при обычных условиях последовательные выходы данных. Во время аппаратной конфигурации (сигнал *RSTDRV* установлен и некоторое время спустя) эти выводы работают только как входы.

DTR — Установленный сигнал показывает модему или другому устройству, что модуль последовательного порта готов установить соединение. Сигнал *DTR#* может быть установлен через регистр управления модемом (Modem Control Register (*MCR*)). Сигнал *reset* сбрасывает этот сигнал. Во время аппаратной конфигурации (сигнал *RSTDRV* установлен и некоторое время спустя) эти выводы работают только как входы.

GND — общая земля.

DSR — установленный сигнал показывает, что модем или другое устройство готовы произвести соединение с модулем последовательного порта. Сигнал *DSR#* является сигналом состояния модема, и центральный процессор может определить его состояние путем чтения бита *DSR* из регистра *MSR* для соответствующего последовательного канала. Бит *DDSR* в регистре *MSR* показывает, изменился ли сигнал *CD#* со времени последнего чтения регистра *MSR*.

RTS — установленный сигнал информирует модем или другое устройство о том, что модуль последовательного порта готов к обмену данными. Сигнал *RTS#* может быть установлен через бит *RTS* регистра *MCR*. Сигнал *reset* сбрасывает этот бит. Во время аппаратной конфигурации (сигнал *RSTDRV* установлен и некоторое время спустя) эти выводы работают только как входы.

CTS — установленный сигнал показывает, что модем или другое устройство готовы к обмену данными. Сигнал *CTS#* является входом состояния модема, и центральный

процессор может определить его состояние путем чтения бита *CTS* из *MSR* соответствующего последовательного порта. Бит *DCTS* в *MSR* показывает, изменился ли сигнал *CTS#* со времени последнего чтения *MSR*.

RI – установленный сигнал показывает что модемом или другим устройством принят телефонный звонок. Сигнал *RI#* является сигналом состояния модема, и центральный процессор может определить его состояние путем чтения бита *RI* из регистра *MSR* для соответствующего последовательного канала. Бит *TERI* в регистре *MSR* показывает, менялся ли уровень сигнала *RI#* с низкого на высокий со времени последнего чтения регистра *MSR*.

Примечание: здесь *MSR* – регистр статуса модема (*Modem Status Register*)

COM-порты в операционной системе типа *Windows* это именованные каналы для передачи данных, называемые обычно *COM1*, *COM2* и т.д. по порядку обнаружения драйверов соответствующих устройств. Например, для обмена информации через *Bluetooth* многие драйверы представляются операционной системе как *COM*-порт, и резервируют похожее имя.

Последовательный интерфейс *RS-232C* был разработан еще в 1969 г. и введен ассоциацией электронной промышленности (*Electronic Industries Associations — EIA*) как стандарт для связи аппаратных средств передачи данных (*Data Communication Equipment — DCE*), в качестве которых, как правило, используются компьютеры, с аппаратными средствами приема данных (*Data Terminal Equipment — DTE*) — периферийными устройствами (принтерами, модемами и т. п.). Из всего многообразия периферийных устройств, подключаемых к компьютеру, в настоящее время интерфейс *RS-232* остался, пожалуй, лишь у выносных модемов, поскольку возросший на несколько порядков объем информации, передаваемой в периферийные устройства, потребовал новых высокоскоростных интерфейсов, таких, например, как параллельный (*Centronics*) или высокоскоростной последовательный (*USB*). Но сейчас подавляющее большинство модемов выпускаются в виде плат, которые вставляются в материнскую плату компьютера (и сопрягаются с ней по внутренним интерфейсам *ISA* или *PCI*) либо даже интегрируются в материнскую плату.

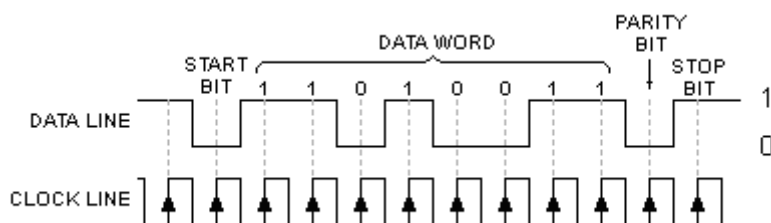
Казалось бы, использованию интерфейса *RS-232* приходит конец. Но нет, интерфейс *RS-232* до сих пор имеет подавляющее большинство компьютеров, поскольку он и сейчас не утратил своего коммуникационного назначения, несмотря на низкую скорость обмена информацией. В чем причина такой поразительной живучести этого интерфейса?

Второе рождение интерфейс *RS-232* пережил около 10 лет назад, когда начался выпуск микроконтроллеров. Поначалу скорость работы микроконтроллеров была такой низкой, что единственным интерфейсом, который позволял производить обмен информацией между компьютером и микроконтроллером, был интерфейс *RS-232*. Но к настоящему времени скорость работы современных микроконтроллеров возросла как минимум на порядок, а интерфейс *RS-232* все равно имеет практически любой микроконтроллер, причем многие микроконтроллеры оборудованы даже двумя такими интерфейсами.

Причины, по которым интерфейс *RS-232* продолжает до сих пор использоваться, по-видимому, состоят в том, что этот интерфейс достаточно надежен, допускает связь на расстоянии до 15 м (при скорости обмена 115200 бод) и более (при соответствующем снижении скорости), легко программируется (как в компьютере, так и в микроконтроллере). Всех этих свойств лишен, например, интерфейс *USB*, который сейчас присутствует практически во всех компьютерах.

Передача данных через *COM*-порт осуществляется побайтно с последовательной передачей каждого бита по одной линии связи. Синхронизация процесса передачи данных требует поддержания заданного значения временного интервала, соответствующего одному биту; длительность временных интервалов между байтами произвольная.

Асинхронный режим передачи байтов реализован на основе обрамления каждого байта специальными стартовыми и стоповыми битами. При отсутствии передаваемых битов на линии удерживается уровень логической «1». Передача байтов данных начинается с посылки стартового бита с уровнем «0», после чего, начиная с младшего (нулевого) выдаются последовательно все биты байта. Затем может следовать бит четности, который обеспечивает четное или нечетное общее количество единиц в пакете битов и используется для обнаружений ошибок. Передача байта завершается одним или двумя стоповыми битами с уровнем «1». Эти стоп-биты начинают отмеченное состояние (нет передачи), которое будет сохраняться до тех пор, пока не начнется передача следующего байта данных. Число стоп-битов существенно, поскольку они устанавливают минимальное время, которое должно пройти перед следующим стартовым битом. Сказанное иллюстрирует следующий рисунок:



Примечание: тактовый сигнал при асинхронной передаче данных – это внутренний сигнал интерфейса.

Бит четности, если он присутствует в передаваемом сообщении, используется для контроля корректности передачи и поиска ошибок. Контроль передачи может проводиться как на четность (контрольный разряд равен сумме по модулю 2 информационных разрядов и общее число единичных разрядов четно), так и на нечетность (контрольный разряд не равен сумме по модулю 2 информационных разрядов и общее число единичных разрядов нечетно).

Скорость передачи битов по каналу измеряется в бодах (изменение числа состояний линии в секунду).

Наименьшей скоростью передачи информации считается 300 бод. Эта скорость передачи использовалась в старых модемах (сейчас большинство модемов позволяют достигать скорости передачи от 1200 до 56 000 бод). Семейство компьютеров IBM PC поддерживают скорость передачи данных в 56000 бод. Некоторые типы интерфейсов позволяют достигать скорости передачи данных в 1000000 бод!

Синхронизация работы приемное и передатчика данных предполагает программирование их на работу с одинаковой скоростью передачи данных, на использование конкретного числа стартовых а стоповых битов и режима контроля.

При программировании COM-портов на низком уровне используются десять регистров. Изменение состояния конкретного регистра производится посылкой байта данных в соответствующий порт.

Задание:

1. Изучить теоретические положения по данной лабораторной работе.
2. Установить в системе эмулятор виртуальных COM-портов (*Virtual Serial Port Driver*).
3. Создать два виртуальных COM-порта и эмулировать связь между ними.
4. Разработать программное обеспечение для передачи данных через последовательный интерфейс. Вариантами заданий могут быть следующие:
 - реализовать чат, функционирующий через COM-порт;
 - организовать пересылку файлов с одного компьютера на другой через COM-порт;
 - прочитать и изменить параметры заданного COM-порта.

Практическое занятие № 48 «Исследование программно управляемого обмена по готовности внешнего устройства»

Цель занятия: изучение принципов программно управляемого обмена по готовности внешних устройств на примере работы с контроллером клавиатуры и контроллером прерываний.

Содержание работы:

Состав рассматриваемой системы

В ходе лабораторной работы рассматривается система совместимая по архитектуре с системой Intel 8086. В ее состав входит центральный процессор, обеспечивающий выполнение программ. Центральный процессор имеет внешнюю шину к которой подключаются все периферийные устройства – контроллеры ввода-вывода, память, шинные контроллеры и т.д. Контроллер клавиатуры совместимый с Intel 8042, обеспечивающий обмен данными с устройством ввода “клавиатура” и их буферизацию. Контроллер прерываний, совместимый с контроллером Intel 8259.

Контроллер клавиатуры и контроллер прерываний располагаются в сокращенном адресном пространстве ввода-вывода периферийных устройств, что позволяет обращаться к ним с использованием команд работы портами ввода-вывода IN/OUT.

Режимы обмена с периферийными устройствами

Наиболее распространенными режимами обмена данными с периферийными устройствами являются:

- Программный обмен по готовности (сканирование)
- Обмен по прерываниям
- Обмен с использованием DMA (Direct Memory Access) канала.

В большинстве периферийных устройств существует три типа регистров: контрольные, для управления режимами работы; статусные – для определения состояния устройства и данных, для непосредственного обмена данными. Программно управляемый обмен состоит в опросе статусных регистров периферийного устройства и выявления состояния готовности к обмену данными. Чаще всего опрос осуществляется непосредственно в основном цикле программы постоянно или в соответствии с временной диаграммой, когда опрос происходит через фиксированные промежутки времени. При обнаружении сигнала готовности программа выполняет обмен данными через регистры данных. Этот способ является наиболее простым, поскольку позволяет иметь линейное построение программного обеспечения и отсутствие параллельных процессов. Но это также является и недостатком этого метода, поскольку затрудняется поддержка нескольких периферийных устройств критичных по времени реакции программной системы, и необходимость иметь поддержку данного вида устройств непосредственно в программе.

Второй способ – обмен по прерываниям, когда устройство формирует специальный сигнал, идущий после некоторой обработки на специальный вход процессора, и говорящий о том, что устройство готово к обмену и процессор должен прервать текущую программу, чтобы перейти к подпрограмме обмена. Достоинством этого метода является то, что существенно упрощается поддержка нескольких периферийных устройств, а также уменьшается время реакции системы.

Обмен данными посредством канала прямого доступа к памяти (DMA) подразумевает что устройство передает данные непосредственно в предопределенную область памяти самостоятельно без прямого участия программного обеспечения (кроме настройки этого канала), а затем сообщает процессору об окончании обмена посредством прерывания или специального бита в статусном регистре контроллера прерывания. Обычно этот режим используется для устройств, которые имеют специализированную поддержку этого режима и его необходимость состоит в существенном отличии скоростей работы процессора и устройства, а также позволяет несколько разгрузить программное обеспечение.

Реализация программного обмена между центральным процессором и контроллером клавиатуры посредством контроллера прерываний.

В данной лабораторной работе рассматривается следующая схема работы. Контроллер клавиатуры принимает скан-код нажатой клавиши и помещает его во внутренний буфер. Обычный скан-код представляет собой байтовое значение, младшие семь битов которого описывают код клавиши (0..127), а старший бит нажатие(0)/отпускание(1). Буфер клавиатуры доступен для чтения через порт 60h сокращенного пространства ввода-вывода. Контроллер клавиатуры также имеет и статусный и контрольный регистр, но в данной лабораторной работе предполагаем, что он настроен BIOSом на режим обмена по прерываниям и его перепрограммирование производиться не будет.

По появлению кода в буфере контроллер формирует сигнал запроса на прерывание, который попадает в контроллер прерываний. Контроллер прерываний необходим для того, чтобы принимать сигналы от пятнадцати источников прерываний и формировать один сигнал запроса на прерывание для процессора, с последующей выдачей номера источника в специальном шинном цикле определения номера вектора. Контроллер клавиатуры подключен к входу IRQ1 ведущего контроллера прерываний.

Контроллер прерываний состоит из двух одинаковых микросхем, включенных каскадно. Каждая микросхема обеспечивает поддержку до восьми источников. В данной лабораторной работе будет использована только входная часть контроллера прерываний которая состоит из регистра IRR (Interrupt Request Register) и IMR (Interrupt Mask Register). Регистр IRR предназначен для запоминания запросов на прерывания от устройств и будет хранить соответствующий запрос. Регистр IMR предназначен для блокирования прохождения запросов на прерывания от соответствующих устройств. Единичное значение соответствующего бита в регистре IMR блокирует прохождение прерывания на дальнейшую обработку по этой линии. Регистр IMR доступен через порт 21h для чтения и записи напрямую. Регистр IRR доступен только для чтения через порт 20h, но этот порт используется для обращения к нескольким регистрам контроллера прерываний, поэтому предварительно в этот порт должна быть отправлена команда на чтение регистра IRR, которая представляет собой код 0Ah.

Для обеспечения режима программного обмена, разрабатываемая программа должна обеспечить блокирование прохождения прерывания от клавиатуры через регистр IMR. Затем в цикле должен анализироваться регистр IRR на предмет появления запроса от клавиатуры. После появления запроса необходимо выбрать очередной символ из буфера клавиатуры и приступить к его анализу и обработке.

Задание на выполнение лабораторной работы

1. Написать программу («com» файл), последовательно выполняющую:
2. Очистку экрана.
3. Чтение и вывод на экран содержимого регистра маски прерываний (адрес 21h) в двоичном или шестнадцатеричном формате.
4. Запрет аппаратных прерываний от клавиатуры.
5. Чтение готовности клавиатуры по изменению соответствующего бита в регистре запросов прерывания (IRR).
6. По готовности клавиатуры, если нажата клавиша “Esc” восстановить прерывания от клавиатуры и выйти из программы.
7. Дополнительно рассмотреть вариант с проверкой готовности клавиатуры по опросу регистра состояния контроллера клавиатуры.

Рекомендации:

1. Данная программа может правильно работать только под управлением операционной системы MS DOS.
2. Для контроля работы программы в циклах опроса готовности и для функций задержки на время обмена с контроллером прерываний использовать

вывод фиксированных символов на экран с переводом строки в конце каждой строки.

Практическое занятие № 49 «Работа с анимационными устройствами ввода-вывода»

Цель занятия: научиться работать с анимационными устройствами ввода-вывода информации.

Содержание работы:

Задание 1. Работа с горячими клавишами клавиатуры. Изучите сочетания горячих клавиш. Откройте любой текстовый файл проверьте работу горячих клавиш.

CTRL + C Копирование

CTRL + Z Отмена

CTRL + S Сохранить

SHIFT + DELETE Удаление элемента без помещения его в корзину и возможности восстановления

Удерживание нажатой клавиши CTRL при перетаскивании элемента Копирование выбранного элемента

Удерживание нажатыми клавиш CTRL + SHIFT при перетаскивании элемента Создание ярлыка для выбранного элемента

F1 Вызов помощи

F2 Переименование выбранного элемента

F3 Поиск файла или папки

F5 Найти и заменить в Word

CTRL+A Выделение всего

ALT+ENTER Просмотр свойств выбранного элемента

ALT+F4 Закрытие текущего элемента или выход из активной программы

ALT+TAB Переход от одного открытого окна к другому

ALT+ESC Переключение между окнами в том порядке, в котором они были открыты

SHIFT+F10 Открытие контекстного меню для выделенного элемента

SHIFT+ стрелка Выделить группу файлов

CTRL+ESC Открытие меню Пуск

SHIFT Печать заглавных букв в Word

DEL (DELETE) Удалить выделенное или знак после курсора

CAPS LOCK Печатать заглавные буквы

NUM LOCK Включает цифровой блок (правая сторона панели)

HOME/END Переход к началу/концу строки

PAGE UP/PAGE DOWN Переход к началу/концу страницы

Горячие клавиши ПК

CTRL+ ALT+ DELETE – КЛАВИШИ ДЛЯ ВЫЗОВА ДИСПЕТЧЕРА ПРОГРАММ, ЕСЛИ КОМПЬЮТЕР ЗАВИС!

Задание 2. Работа со сканером

1. Включите сканер и щелкнув по ярлыку программы откройте программу сканирования Abby FineReader
2. Отсканируйте текст. Сохраните файл с расширением doc.
3. Отсканируйте рисунок. Сохраните файл с расширением jpeg
4. Отсканируйте текст (английский язык). Сохраните файл с расширением doc.

Задание 3. Работа с микрофоном

1. Используя цифровой диктофон вашего телефона запишите на него текст ЗИМА

ЗИМА

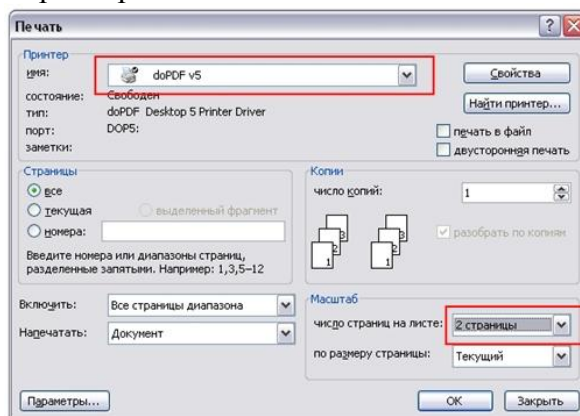
Она явилась закутанной, белой, холодной.

- Кто ты? – спросили дети.
- Я – время года – зима. Я принесла с собою снег и скоро набросаю его на землю. Он все закроет белым пушистым одеялом. Тогда придет мой брат – дедушка Мороз и заморозит поля, луга и реки. А если ребята станут шалить, то отморозит им руки, ноги, щеки и носы.
- Ой, ой, ой! Какая нехорошая зима! Какой страшный дедушка Мороз! – сказали дети.
- Подождите, дети... Но зато я подарю Вам катанье с гор, коньки и салазки. А после придет любимое Рождество с веселой елкой и дедушка Мороз с подарками. Разве Вы не любите зимы?

Задание 3. Работа с виртуальным принтером. Создайте документ в формате .pdf на страницах формата А4, на которых будут размещены по 2 страницы реального документа в формате .doc.

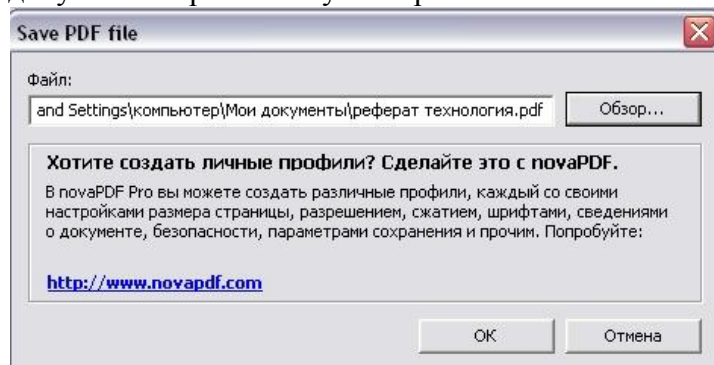
Установите виртуальный принтер. При установке doPDF 5 принтер, укажите принтер, используемый по умолчанию.

- Откройте файл (текстовый документ, документ презентации, лист электронной таблицы и т.д.), для которого создаем PDF файл.
- Напечатаем документ на виртуальном принтере. Файл – Печать – Принтер – Выбираем принтер из списка – doPDF 5.



- Так же как у реального принтера задаем параметры печати. В нашем случае: Страницы – все.
Число копий - 1.
Число страниц на листе – 2 страницы.
Свойства: страница – А4.
Ориентация – книжная. – ОК.

- Откроется окно Save PDF file, выберите место сохранения вашего документа через кнопку Обзор.



Практическое занятие № 50 «Работа с устройствами ввода-вывода звуковых сигналов»

Цель занятия: Знакомство с правилами работы со звукотехническим оборудованием.

Содержание работы:

Устройства вывода звука

1. Колонки
2. Наушники
3. Динамик

С развитием информационных технологий сферы использования звука самые разнообразные: от образовательных, развивающих (чтение, музыка и пр.), бизнес-приложений (тренингов, пресс-конференций, презентаций) до профессиональных программ (прослушивание шумов в сердце, обучение радиотелеграфистов) и профессиональных мультимедиа (озвучивание фильмов, музыкальные редакторы). К устройствам вывода звука в современных ПК относят: колонки, наушники, динамики.

Колонки – периферийное устройство вывода, которое служит для воспроизведения звука. В основном используется акустическая система, которая состоит из двух колонок, но существуют варианты с большим числом. Колонки различаются размерами, формой и мощностью. Колонки (или акустическая система) преобразуют электрический сигнал в звуковое давление. Колонки бывают однополосными (с одним широкополосным излучателем, например, динамической головкой) и многополосными (с двумя и большим количеством головок, которые создают звуковое давление в своей частотной полосе). Также колонки разделяют на: активные (имеют встроенный усилитель, регулятор громкости и тембра, нужны дополнительные источники питания); пассивные (малой мощности).

Наушники являются устройством для персонального прослушивания звуковой информации. По способу передачи звука наушники разделяют на: проводные – соединены с источником с помощью провода, могут обеспечить звук максимального качества; беспроводные – соединяются через беспроводной канал (bluetooth, радио- или инфракрасный). Такие устройства вывода звука мобильны, но имеют привязку к базе и ограниченный радиус действия. Обеспечивают более низкое качество звука, чем проводные. По типу конструкции (виду) наушники делятся на: вставные («вкладыши») – устанавливаются в ушную раковину; канальные (внутриканальные, «затычки») – устанавливаются в ушной канал; накладные – накладываются на ухо; полноразмерные (мониторные) – охватывают все ухо. По акустическому оформлению наушники разделяют на: наушники открытого типа – частично пропускают внешние звуки, при этом достигается более естественное звучание. Преимуществом использования является отсутствие давления на внутреннее ухо; наушники полукрытого типа – обеспечивают частичную звукоизоляцию; наушники закрытого типа – обеспечивают полную звукоизоляцию.

В наушниках используется один из трех типов соединительных разъемов: jack, mini jack или micro jack. Наушники могут крепиться на голове с помощью вертикальной дужки или с помощью затылочной дужки, на ушах с помощью заушины или клипс, или не иметь креплений (вставные или канальные наушники).

Основные технические характеристики наушников: Частотная характеристика влияет на качество звука наушников. Среднее значение частоты от Гц до Гц. Некоторые профессиональные наушники имеют интервал частот от Гц до Гц. Наиболее часто используется частота от Гц до Гц. Чувствительность влияет на громкость звука в наушниках. Обычно наушники обеспечивают чувствительность не менее дБ, если чувствительность меньше, звук может быть слишком тихим. Сопротивление (импеданс). Наушники делятся на низкоомные и высокоомные, причем это разделение зависит от их типа. Например, полноразмерные наушники с сопротивлением до Ом считаются

низкоомными, а наушники внутриканального типа с сопротивлением выше Ом – высокоомные. Большинство современных наушников имеют величину сопротивления Ом. Наушники со значением сопротивления Ом обладают повышенной излучаемой акустической мощностью. Для студийной работы используют наушники с максимальным значением сопротивления. Максимальная входная мощность влияет на громкость звучания. Уровень искажений в наушниках измеряется в процентах, причем чем он меньше, тем выше качество звучания. Для частот от Гц до Гц приемлемым является искажение до , для частот ниже Гц – Динамик

Определение 2 Динамик – является простейшим устройством вывода звука. До появления сравнительно дешевых звуковых плат динамик являлся основным устройством воспроизведения звука. Обеспечивает достаточно низкое качество и примитивность звуков. Динамик все же и сегодня остаётся штатным устройством ПК и в основном используется для подачи сигналов об ошибках, в частности при работе программы . Некоторые программы можно настроить на вывод звуковых сигналов через динамик, что бывает удобно, если к звуковой плате подключены наушники (по умолчанию не надеты).

Задание:

1. Подключить и настроить наушники.
2. Подключить и настроить колонки.
3. Подключить и настроить микрофон.
4. Записать звуковую дорожку и воспроизвести ее.

Практическое занятие № 51 «Генерация компьютерного звука»

Цель занятия: научиться генерировать компьютерный звук.

Содержание работы:

ЗАДАЧА 1.

Можно оценить информационный объем стереоаудиофайла длительностью звучания 1 секунда при высоком качестве звука (16 битов, 48 кГц). Для этого количество битов, приходящихся на одну выборку, необходимо умножить на количество выборок в 1 секунду и умножить на 2 (стерео):

Решение: $16 \text{ бит} \cdot 48\,000 \cdot 2 = 1\,536\,000 \text{ бит} = 192\,000 \text{ байт} = 187,5 \text{ Кбайт}$.

ЗАДАЧА 2.

Оценить информационный объем цифрового стерео звукового файла длительностью звучания 1 минута при среднем качестве звука (16 битов, 24 кГц).

Решение: $16 \text{ бит} \times 24\,000 \times 2 \times 60 = 46\,080\,000 \text{ бит} = 5\,760\,000 \text{ байт} = 5\,625 \text{ Кбайт} \approx 5,5 \text{ Мбайт}$

Стандартное приложение *Звукозапись* играет роль цифрового магнитофона и позволяет записывать звук, то есть дискретизировать звуковые сигналы, и сохранять их в звуковых файлах в формате WAV. Эта программа позволяет редактировать звуковые файлы, микшировать их (накладывать друг на друга), а также воспроизводить.

ЗАДАЧА 3.

1. Определить количество уровней сигнала 24-битной звуковой карты .

ЗАДАЧА 4.

2. Уместится ли песня на дискету размером 1,44 Мбайта, если она имеет следующие параметры: стерео длительностью звучания 3 минуты при качестве звука - 16 битов, 16 кГц.

Практическое занятие № 52 «Ввод в ЭВМ и машинный синтез речи»

Цель занятия: научиться вводить и осуществлять машинный синтез речи.

Содержание работы:

Особое место в системах мультимедиа занимает использование аудиоаппаратуры для речевого общения.

Для распознавания и понимания речи дикторов необходимо ввести речевые сигналы в ЭВМ с помощью акустических устройств ввода и проанализировать вводимую речь.

Системы речевого ввода делятся на два типа по характеру распознаваемой речи:

- системы, ориентированные на восприятие отдельных слов;
- системы, воспринимающие связную речь.

Разница между ними весьма существенна, так как при слитном произношении слов изменяется их звучание.

При анализе отдельных слов (команд) осуществляется их оцифровка, идентификация и инициируется выполнение программы, отрабатывающей принятую команду. Этот же режим используется и для речевого ввода цифровой информации; в этом случае после идентификации введенное слово преобразуется в соответствующий код ASCII (за счет чего достигается существенное сжатие речи). Фирма «Курцвейл» выпускала по этому принципу устройство Voice Writer, которое распознавало около 10000 отдельно произнесенных английских слов и печатало их на принтере.

Сегодня практически отсутствуют устройства для ввода динамически развивающихся звуковых сцен. Устройства ввода и программы-анализаторы не позволяют выделить эмоциональную составляющую речи, которая значительно корректирует смысл (и может даже изменить его до противоположного). Чаще всего эмоциональная составляющая рассматривается как помеха (за исключением систем контроля эмоционального состояния). Как дополнительный источник информации эмоциональная окраска голоса (и введенного сообщения) в настоящее время не используется.

Системы речевого вывода называются *синтезаторами речи*.

Существуют три основных технологически различных подхода к проблеме синтеза речи:

- метод сжатия-восстановления формы сигналов;
- аналоговый метод синтеза формантных частот;
- цифровое моделирование голосового тракта.

Первый метод — самый простой. ЭВМ в этом случае служит как цифровой магнитофон. Фразы и слова записываются отдельно и выбираются для воспроизведения в нужный момент по командам, поступающим от соответствующей программы. В такой системе невозможно воспроизвести слово, которое не было заранее записано.

Для хранения оцифрованной речи требуется память большого объема, так как хранить необходимо каждое слово из лексикона ЭВМ с учетом различных падежных окончаний, рода («пошел-пошла-пошло»), числа и т.д. Но зато качество воспроизведения речи очень высокое.

Разновидностью синтезаторов этого типа являются автоответчики, построенные из ЭВМ и Voice-модема; речевая телепочта (передача речевого сообщения по вычислительным сетям).

Считается, что этот метод эффективен, когда словарный запас невелик — не превышает 10—15 слов (например, говорящий приборный щиток автомобиля, говорящие часы, калькулятор, календарь).

Второй метод использует принципы акустического моделирования голосового тракта человека. Речь составляется из формантных частотных полос, которые создаются полосовыми фильтрами. Суммарный выходной сигнал формантных фильтров достаточно близко соответствует частотному спектру человеческой речи. Но такая речь звучит, как голос робота, разборчивость ее оставляет желать лучшего.

Этот метод универсален: с его помощью можно синтезировать любые слова, иметь неограниченный словарь, так как речь создается из отдельно генерируемых звуков. Синтезатор может быть реализован программным путем.

Наиболее распространенный способ возбуждения синтезатора формантных частот состоит в использовании отдельных, поддающихся идентификации звуков речи, называемых **фонемами**.

Фонемный синтезатор образует последовательность фонем, которая при воспроизведении на акустическом устройстве вывода звучит как речь.

Фонемный синтез речи практически не требует дополнительной аппаратуры; он может быть реализован на ЭВМ стандартной конфигурации программным путем.

Речь разделяется на отдельные элементарные части — фонемы. Например, в английском языке выделяются такие фонемы для гласных звуков, как *ee*, *i*, *eh* и др. (табл.).

Таблица **Фонемы гласных звуков английского языка**

Фонема	Произношение	F_1	F_2	f_3
ee	feet	250	2300	3000
i	hid	375	2150	2800
eh	head	550	1950	2600
ae	had	700	1800	2550
ah	tot	775	1100	2500
aw	talk	575	900	2450
u	took	425	1000	2400
oo	tool	275	850	2400

F_1, F_2, F_3 — три основные формантные частоты, наблюдаемые в спектрограмме при произношении средним мужским голосом.

Однако кроме гласных в речи человека существуют фрикативные, взрывные и носовые согласные. Помимо этого каждая фонема имеет вариации — аллофоны.

В русском языке согласные фонемы бывают мягкие и твердые, глухие и звонкие (шумные, сонорные, губные, зубные, альвеолярные, велярные).

Третий метод использует словарь, который создается голосом человека, но в память записывается не оцифрованный акустический сигнал, а его частотные параметры, при этом уменьшается объем памяти, занимаемый словарем. Синтез же речи производится интегральными микросхемами, генерирующими заданный набор частот с заданными амплитудами и смешивающими их.

Практическое занятие № 53 «Изучение работы гибких магнитных дисков»

Цель занятия: укрепить навыки работы с гибкими дисками.

Содержание работы:

Для получения практических навыков в работе с дискетами сделайте следующее задание.

1) Предварительно убедившись в отсутствии на дискете нужной информации, отформатируйте ее. Для этого сделайте следующие действия:

- вставьте дискету в дисковод;
- дважды щелкните мышью на значке Мой компьютер(My Computer), на Рабочем. Столе (Desktop);
- в появившемся окне папки Мой компьютер (My Computer), щелкните правой клавишей мыши по иконке Диск3,5 (A:) (3.5 Floppy (A:));
- в открывшемся контекстном меню выберите команду Форматировать (Format);
- в появившемся окне Форматирование (Format), в поле Емкость (Capacity) Выберите максимальное значение, составляющее 1,44 МБ (1.44 Mb);
- для системы Windows в группе Способ Форматирования (Format type) установите переключатель Полное (Full)

щелкните левой клавишей мыши по кнопке Начать (Start), и в случае появления предупреждения нажмите кнопку ОК;

- по окончании процедуры форматирования, закройте информационное окно, сообщаемое о результатах форматирования, щелчком левой клавиши мыши либо по кнопке Заккрыть (Close).

2) Скопируйте на дискету любой из созданных вами на предыдущих уроках файлов. Для этого сделайте следующие действия:

- запустите программу Проводник;
- найдите диск и папку, которых находится файл, подлежащий процедуре копирования (например, диск–Z, папка– User);
- откройте папку с файлом;
- щелкните правой клавишей мыши по названию файла;
- в открывшемся контекстном меню переведите указатель мыши на поле Послать (Send To);
- в открывшемся списке выберите Диск3,5 (A:)(3.5 Floppy (A:)) или(3floppy~94) и щелкните левой клавишей;
- по окончании процедуры копирования убедитесь в том, что нужный файл находится на дискете.

3) Повторите процедуру копирования файлов.

4) Удалите все файлы, находящиеся на дискете. Для этого выполните следующие действия:

- в программе Проводник перейдите на диск A;
- выделите файлы, подлежащие удалению;
- нажмите клавишу Delete на клавиатуре.

4) С помощью дискеты перенесите любую информацию с другого компьютера.

Практическое занятие № 54 «Изучение работы накопителей информации на жестких магнитных дисках»

Цель занятия: Изучение конструкции, особенностей функционирования и порядка подготовки к использованию НЖМД.

Содержание работы:

1. Изучить основные теоретические положения, сделав необходимые выписки в конспект лекций.
2. Рассмотреть особенности конструкции НЖМД на представленных образцах.
3. Оформить отчет:
 - состав элементов НЖМД;
 - процесс включения и работы НЖМД – тезисно;
 - этапы подготовки НЖМД к использованию;
 - структура разбиения на разделы (3.4.3);
 - основные параметры дисковых накопителей;
 - описать установку параметров дисководов в BIOS Setup вашего ПК.
4. Ответить на контрольные вопросы.

Практическое занятие № 55 «Определение неисправностей накопителей на жестких магнитных дисках»

Цель занятия: изучить характерные неисправности накопителей HDD.

Содержание работы:

Порядок выполнения работы

1. Ознакомиться с теоретической частью данной работы.
2. Изучить устройство накопителей HDD и подключение интерфейсов.
3. Ознакомиться с контрольными вопросами и продумать ответы на них.

4. Сдать устный или письменный (по вариантам) зачёт по данной лабораторной работе.

Краткие теоретические сведения

Устранение неисправности жесткого диска — достаточно серьезное занятие, которое не всегда заканчивается полным успехом. Дело усложняется еще и тем, что в большинстве случаев на жестком диске записаны важные данные, которые, чаще всего, не были в нужный момент скопированы на внешний носитель информации.

Неисправности жесткого диска могут быть следующих видов:

- логическая неисправность;
- неисправность контроллера жесткого диска;
- потеря или разрушение служебной информации
- физическая неисправность.

Логическая неисправность

При логической неисправности загрузка операционной системы с жесткого диска невозможна, а при подключении его в качестве вторичного не отображается содержимое логических дисков или сами диски. Самое интересное, что такой винчестер корректно распознается в BIOS и при проверке не выявляется никаких физических повреждений диска. Такой диск нельзя исправить проверочными утилитами вроде ScanDisk. Мало того, использовать такие утилиты крайне не рекомендуется, поскольку в процессе проверки они могут записывать на жесткий диск какие-либо данные, что может привести к потере важной информации. Логическую неисправность можно устранить *только с помощью специализированных утилит*, которые на низком уровне восстанавливают разделы и файловую структуру винчестера. Следует помнить о том, что любое вмешательство в структуру диска может привести к потере информации. К такому эффекту могут привести встроенные в жесткий диск средства диагностики, например технология S.M.A.R.T., или переназначение диска. Чтобы избежать этого, при первых признаках неисправности диска эти средства диагностики нужно отключить.

Неисправность контроллера жесткого диска

К данному виду относятся неисправности, связанные с физическими повреждениями компонентов контроллера диска: перегоревшими микросхемами, поврежденными головками, оторванным интерфейсным кабелем и т. п.

Существует два варианта определения таких неисправностей.

Первый из них — самый простой, когда признаки разрушения компонентов контроллера, например дыры на микросхемах, выявляют при внешнем осмотре. В этом случае перед началом каких-либо действий следует заменить сгоревшие компоненты.

Второй вариант, соответственно, самый сложный: когда видимых признаков повреждения нет, однако жесткий диск ведет себя нестандартно. Рассмотрим некоторые ситуации.

- *Двигательне раскручивается, никаких звуков не слышно*. Возможные причины - заклинило шпиндельный двигатель или головки чтения/записи "прилипли" к поверхности диска.

Если попытаться повернуть шпиндель, то это может привести не только к повреждению поверхности диска и, соответственно, пропаже информации, но и к поломке механизма управления головками и самих головок. В этом случае неисправный винчестер лучше отнести в сервисный центр, где, во-первых, вам точно скажут, подлежит ли жесткий диск ремонту, а во-вторых, переписут с него всю важную информацию

- *Двигатель раскручивается, слышен щелчок*. Этот щелчок является следствием неудачной распарковки головок. Наиболее вероятная причина данной неисправности - выход из строя генератора шпиндельного двигателя или системы позиционирования головок. Возможно также повреждение катушки позиционирования, которая размещена на блоке головок.

- *Двигатель раскручивается*, однако *диск не определяется* или *определяется неверно*. Причиной возникновения такой неисправности может быть выход из строя интерфейсной микросхемы контроллера или механическое повреждение контактной группы, к которой подключается шлейф данных (например, согнутый или сломанный металлический вывод).

- Диск раскручивается, слышен *стук*. Данная ситуация может означать очень многое, начиная с неисправной системы позиционирования и заканчивая повреждением головок. Еще один вариант - сбойные секторы в загрузочной области винчестера. Такую ситуацию вы сможете исправить самостоятельно. Определив состояние жесткого диска, подключим шлейф данных. Теперь можно не только услышать издаваемые жестким диском звуки, но и увидеть на экране сообщения об ошибках в ходе работы винчестера. Возможны следующие основные ситуации.

- Появление сообщений типа **Invalid Drive Specification**. Такие сообщения означают, что в BIOS записана неверная информация о параметрах установленного винчестера или же он неверно распознается. Последний вариант говорит о разрушении служебной информации или о повреждении головок. Возможна так же поломка контактов интерфейса на плате контроллера или в самом шлейфе данных.

- Появление сообщений типа **Disk Boot Failure**. Это однозначно говорит о том, что повреждена MBR (Master Boot Record) — основная загрузочная запись.

- Появление сообщений типа **Boot Disk Failed**. Самая вероятная причина — наличие сбойных секторов на нулевой дорожке, вследствие чего загрузка с жесткого диска невозможна. ОС нормально распознают жесткий диск, однако *не отображает логические* диски. Если исключить вариант того, что винчестер просто не разбит на логические диски, то основной причиной такой неисправности является наличие сбойных дисков в системной области или разрушение файловой системы, содержащей информацию о текущей конфигурации дисков. В этом случае следует заново разбить винчестер на логические диски, после чего отформатировать его. Можно также более детально изучить неисправность с помощью низкоуровневых утилит.

- Жесткий диск распознается нормально, однако *ОС загружается не полностью или не загружается* вовсе. Данный факт говорит о том, что область винчестера, в которой записаны файлы операционной системы, содержит сбойные секторы. В данном случае следует обратиться к низкоуровневым утилитам, которые пометят сбойные секторы и в случае необходимости перепланируют винчестер.

Профилактика HDD

Со временем диск начинает работать медленнее, со сбоями, греется, появляются ошибки чтения и т. п. Это означает, что пришло время профилактики, основными мерами которой являются:

- **Проверка логического и физического состояния** диска с помощью диагностических утилит типа **scandisk** (для систем Windows 95/98). Такая проверка позволяет не только исправить логические ошибки, освободив место на жестком диске, но и узнать о появлении сбойных секторов, которые утилита помечает соответствующим образом. ОС Windows XP проверяет диск немного иначе, однако механизм проверки тот же: чтобы запустить утилиту, кликните ПКМ на интересующем вас диске, выберите **Свойства** ► **Сервис** ► **Выполнить проверку...** В открывшемся окне установите требуемые параметры проверки и запустите диагностику на выполнение.

В Windows-7: «Пуск» ► «Все программы» ► «Стандартные» ► «Служебные» ► «Очистка диска» или скачайте и установите **TreeSize Free**.

- **Дефрагментация файлов** с помощью стандартных утилит или утилит стороннего производителя. Фрагментация — это разбиение одного файла на несколько фрагментов и размещение их на разных участках диска. Фрагментация происходит не намеренно, а из-за специфики записи информации. Дефрагментация — соответственно, соединение фрагментов одного файла. Если диск дефрагментирован, то скорость доступа к

информации увеличивается. Со временем фрагментация новых файлов снова приводит к замедлению работы, в этом случае процесс дефрагментации следует повторить.

- **Наблюдение за температурой диска** с помощью специализированных утилит. Температура как ничто другое влияет на состояние производительности жесткого диска. В результате повышения температуры могут пострадать внутренние компоненты контроллера, что приведет к появлению серьезных неисправностей. Этого вполне достаточно для обеспечения, по крайней мере, логического "здоровья" жесткого диска.

Контрольные вопросы

1. Назовите основные компоненты HDD и их назначение.
2. Отличие способа записи информации в HDD и FDD.
3. Среднее время доступа в накопителях, единицы его измерения.
4. Чем определяется надёжность HDD?
5. Что определяет быстродействие HDD?
6. Назовите интерфейсы подключения HDD (внутренних и внешних).
7. Что происходит при форматировании жёстких дисков?
8. Какие виды форматирования вы знаете? Для каких носителей существует возможность создания логических дисков?
9. Назовите виды неисправностей HDD.
10. Назовите меры профилактики HDD, охарактеризуйте их.
11. О чём говорит появление сообщения **Disk Boot Failure**?
12. Какова причина возникновения ситуации *Двигательные раскручивается, никаких звуков не слышно?*
13. Какова причина появления сообщения **Invalid Drive Specification**?
14. Какова причина возникновения ситуации *Двигатель раскручивается, однако диск не определяется* или *определяется неверно?*

Практическое занятие № 56 «Работа с системным программным обеспечением»

Цель занятия: приобретение навыков работы с системным программным обеспечением в среде WINDOWS.

Содержание работы:

Системное программное обеспечение — это комплекс программ, которые обеспечивают управление компонентами компьютерной системы, такими как процессор, оперативная память, устройства ввода-вывода, сетевое оборудование, выступая как «межслойный интерфейс», с одной стороны которого аппаратура, а с другой — приложения пользователя. В отличие от прикладного программного обеспечения, системное не решает конкретные прикладные задачи, а лишь обеспечивает работу других программ, управляет аппаратными ресурсами вычислительной системы и т.д.

Системное программное обеспечение – совокупность программ и программных комплексов для обеспечения работы компьютера и сетей ЭВМ. Системное программное обеспечение подразделяется на базовое и сервисное программное обеспечение.



Базовое программное обеспечение – минимальный набор программных средств, обеспечивающих работу компьютера: к нему относятся операционные системы и операционные оболочки. Базовое ПО – самый низкий уровень программного обеспечения. Оно отвечает за взаимодействие с базовыми аппаратными средствами. Как правило, базовые программные средства непосредственно входят в состав базового оборудования и хранятся в специальных микросхемах, называемых ПЗУ. Программы и данные записываются в микросхемы ПЗУ на этапе производства и не бывают изменены в процессе эксплуатации.

В тех случаях, когда изменение базовых программных средств во время эксплуатации является технически целесообразным, вместо микросхем ПЗУ применяют перепрограммируемые ПЗУ. В этом случае изменение содержания ПЗУ можно выполнять на специальных устройствах, называемых *программаторами*.

Операционная система – это комплекс программ, организующих управление работой компьютера и его взаимодействие с пользователем.



1. Операционная система представляет собой комплекс программ, выполняющих следующие функции:

- загрузка компьютера (проведение проверки работоспособности устройств компьютера, предоставляя системные программы для работы с файлами, директориями и дисками);
- управление памятью и операциями ввода-вывода;
- управление файловой системой;
- обеспечение правильности выполнения задач (программ);
- обеспечение защиты и сохранности информации;
- управление работой прикладных программ;
- обеспечение взаимодействия пользователя с компьютером (интерфейс);

WINDOWS совмещает функции операционной системы и операционной оболочки.

Как

операционная оболочка **WINDOWS** обеспечивает пользователя такими средствами, как:

- графический интерфейс;
- набор стандартных встроенных программ;
- возможность отображать на экране несколько документов и работать с несколькими программами в разных окнах одновременно.

2. Рабочий стол.

Экран **WINDOWS** называется **Рабочим столом**, на котором находятся различные значки- **ярлыки**, окна и **Панель задач**. **Рабочий стол** - это корневая папка файловой системы, которая всегда отображается на экране монитора.

Обязательными элементами рабочего стола являются:

Мой компьютер - программа для работы с файловой системой компьютера, с дисками, папками и файлами. Она позволяет перемещать, копировать, переименовывать и удалять элементы файловой системы.

Корзина - программа, в окне которой хранятся удаленные файлы, папки и другие элементы. До тех пор пока Вы сознательно не «очистите» корзину, Вы можете восстановить любые удаленные элементы.

Сетевое окружение - программа для работы в локальной сети.

Панель задач - прямоугольная полоса, обычно в нижней части Рабочего стола. На Панели задач находятся кнопка Пуск и кнопки всех открытых программ и документов.

3. Понятие окна. Управление окнами.

При двойном нажатии левой кнопки мыши каждый значок раскрывается в окно. **Окно** - это ограниченный рамкой прямоугольник экрана. Все окна имеют стандартную структуру:

Первая строка - **Название окна**. Если строка заголовка высвечена, то окно активно. Нажав левую кнопку мыши на строке заголовка и не отпуская ее, можно переместить окно в любую область рабочего стола.

В конце строки заголовка три прямоугольника, позволяющих управлять окном:

- кнопка свернуть (черточка на ней показывает, что если вы щелкните на кнопке, окно сожмется до линии);
- кнопка развернуть\свернуть в окно (щелчок на ней увеличивает окно до размера экрана и обратно возвращает окну прежний размер);
- кнопка закрыть (щелчок на кнопке закрывает окно).

Вторая строка - Командное меню, которое включают группы команд Файл (команды работы с дисками и папками), Правка (команды работы с файлами), Вид (команды вида экрана), ? (Справка о WINDOWS). Выбранные команды помечаются флажком-галочкой. Для изменения выбора достаточно щелкнуть на команды мышкой.

Если содержащаяся в окне информация превышает размеры окна, то справа и снизу окна появляются **полосы прокрутки** со стрелками-указателями движения.

Если на экране раскрыто несколько окон, перекрывающих друг друга, то щелчок мыши на любой части перекрытого окна выведет его вперед.

Управление окнами - перемещение, изменение размера, использование полос прокрутки, свертывание, закрытие осуществляется с помощью указателя мыши.

Раскрытие свернутого окна или переход из окна в окно осуществляется с помощью **Панели задач**, появляющейся внизу экрана линейки с названиями открытых окон.

Управление расположением окон на экране осуществляется с помощью контекстного меню. Для этого надо щелкнуть правой кнопкой мыши на свободном месте экрана. Появится контекстное меню, включающее команды расположения окон: каскадом (одно за другим с перекрытием), сверху вниз (одно под другим), слева направо (одно за другим в линейку).

4. Организация файловой системы. Файлы и папки.

Операционная система работает с **файлами** информации. Файлы могут быть собраны в **папки** (аналог: каталоги, директории - названия в более ранних операционных системах). В папке может храниться произвольное число файлов. Файл можно перемещать из папки в папку.

В **WINDOWS** можно просмотреть содержимое существующих папок, создать новые папки, переименовать или удалить существующие папки. Можно создавать одну папку внутри другой.

Просмотр файловой системы.

Чтобы **просмотреть** файловую систему компьютера можно использовать программу **Мой компьютер** или программу **Проводник**.

В первом случае необходимо:

- 1) Дважды щелкнуть на значке **«Мой компьютер»**. Все дисководы вашего компьютера представлены в этом окне значками дисководов и именами.
- 2) Выбрать название дисковода и дважды щелкнуть на этом значке.
- 3) Выбрать необходимую папку и дважды щелкнуть на этом значке. Содержимое папки раскроется в новом окне.

Каждый тип файла имеет в окне особое обозначение. Программные файлы обозначаются прямоугольниками с утолщенной верхней чертой, а текстовые файлы - прямоугольник с загнутым левым уголком.

Во втором случае необходимо:

- 1) в стартовом меню выбрать пункт **«Программы»**;
- 2) пункт **«Стандартные»**;
- 3) пункт **«Проводник»**. Откроется окно программы **«Проводник»**, состоящее из двух половин. Для того чтобы просмотреть содержимое диска или папки, необходимо щелкнуть на значке в левой половине, а в правой половине окна откроется его содержимое.

Файлы хранятся на внешних устройствах памяти, как правило, это жесткий диск. Однако для хранения резервных копий и переноса информации необходимы гибкие магнитные диски. Для работы с гибким магнитным диском он должен быть отформатирован.

Работа с папками.

Для создания папки необходимо:

- 1) Дважды щелкните на значке **«Мой компьютер»**, а затем на названии диска и при необходимости папки, куда необходимо будет поместить новую папку.
- 2) В меню **«Файл»** выбрать строку **«Создать»**. В окне будет создана новая папка без названия. Вы должны ввести название папки в поле ввода под значком папки. (С помощью группы команд **Файл** можно переименовать и удалить папки и файлы.)
- 3) После ввода названия нажмите клавишу **«Enter!»**.

Работа с файлами.

WINDOWS позволяет создавать файлы, копировать их в папки и на дискеты, переименовывать их, искать необходимые файлы, удалять и восстанавливать файлы.

Файл - это поименованная область на диске или другом машинном носителе.

Создание файла осуществляется с помощью прикладной программы - текстового или графического редактора и т.д.

Копировать файлы можно одним из следующих способов:

1. С помощью команд **«Правка»** в окне **«Мой компьютер»**:
 - 1.1.откройте папку с необходимым файлом;
 - 1.2.в меню **«Правка»** щелкните на строке **«Копировать»**. Файл будет помещен в Буфер обмена;
 - 1.3.откройте папку, куда необходимо скопировать файл. В меню **«Правка»** выберите команду **«Вставить»**.
2. С помощью «контекстного меню» и «комбинаций горячих клавиш»:
 - 2.1.щелкните на необходимом файле, затем воспользуйтесь комбинацией клавиш **«Ctrl+C»** или щелкните правой кнопкой мыши на значке нужного файла и в контекстном меню выбрав команду **«Копировать»**;
 - 2.2.для вставки необходимо открыть нужную папку, а затем воспользоваться комбинацией клавиш **«Ctrl+V»** или щелкнуть на нужном окне и в контекстном меню выбрать команду **«Вставить»**.
3. С помощью **«мыши»** (частный случай метода *drag-and-drop*-перетащить и оставить):

- 3.1.откройте два окна. Первое, в котором находится необходимый файл, второе - окно, куда надо скопировать данный файл;
- 3.2.нажмите клавишу «**Ctrl**» и левую клавишу мыши на значке файла. Не отпуская этих клавиш, перетащите значок файла в другое окно. При этом к указателю мыши добавится знак «+», показывающий, что файл копируется.

При копировании с жесткого диска на дискету необходимо открыть окно дискеты, щелкнув на ее значке.

Контекстное меню - это меню появляющееся при выделении объекта и нажатии правой кнопки мыши.

Для **переноса** файла в меню «**Правка**» используйте команду «**Вырезать**», а в способе с использованием мыши вместо клавиши «**Ctrl**» клавишу «**Shift**».

Удаление файла осуществляется либо с помощью команды **Меню Правка Вырезать**, либо с помощью «**Корзины**»:

- 1) Откройте необходимую папку.
- 2) Нажмите левую клавишу мыши на значке файла и, не отпуская ее, переместите значок в «**Корзину**».

Поиск файла осуществляется программой «**Поиск**» («**Найти**»):

- 1) В стартовом меню укажите на пункт «Поиск». Откроется меню программы.
- 2) Выберите строку «**Файлы и папки**». Откроется диалоговое окно «**Найти**». Поиск файла можно осуществлять по: имени, дате и дополнительным параметрам таким как - тип файла, текст или размер.
- 3) В строке имя наберите имя файла и нажмите клавишу «**Найти**». Внизу диалогового окна появится перечень всех файлов, в названии которых есть набранное сочетание.

5. Ярлыки и работа с ними

Ярлык - значок, который представляет программу, папку или файл и является их указателем. Ярлык позволяет легко открыть тот элемент, адрес которого в нем прописан (т.е. на который он «наклеен»). Ярлык можно поместить на Рабочий стол, в стартовое меню или в меню Программы.

Создание ярлыка

1 способ:

- 1) установите курсор на файле/папке, для которого хотите создать ярлык;
- 2) нажмите правую кнопку мыши и в контекстном меню пункт «**Создать ярлык**»;
- 3) рядом с Вашим файлом/папкой появиться ярлык;
- 4) перенесите его на рабочий стол.

2 способ:

- 1) установите курсор на файле/папке, для которого хотите создать ярлык;
- 2) нажав и не отпуская правую кнопку мыши, перенесите файл/папку на рабочий стол. Появится контекстное меню;
- 3) в контекстном меню щелкните на пункте «**Создать ярлык**». Ярлык папки появится на рабочем столе.

Замечание: Если вы перенесете папку или документ, удерживая нажатой левую кнопку мыши, на «**Рабочем столе**» появится еще одна копия документа или папки, а не ярлык этого объекта.

Переименование ярлыков.

- 1) Щелкните правой кнопкой мыши на ярлыке программы. Появится контекстное меню.
- 2) В этом меню щелкните на пункте «**Переименовать**». Выделенная надпись изменит цвет, и в ней появится мигающий курсор вставки.
- 3) Наберите новое название и нажмите клавишу «**Enter**».

Работа с ярлыками.

Для того чтобы **«Упорядочить ярлыки»** на **«Рабочем столе»** в требуемом порядке необходимо:

- 1) открыть контекстное меню;
- 2) выбрать в нем строку **«Упорядочить значки» «По имени»** (ярлыки расположатся в алфавитном порядке) или **«По дате»** (ярлыки будут упорядочены по времени их создания).

Для того чтобы **запустить программу** с помощью **ярлыка**, достаточно произвести на нем двойной щелчок левой кнопкой мышки.

Удаление ярлыков.

Удаление ярлыка можно осуществить несколькими способами:

- перенести ярлык на значок **«Корзина»**;
- правой кнопкой мыши щелкнуть на ярлыке, в контекстном меню - на пункте **«Удалить»**. В диалоговом окне **«Подтвердите удаление»** щелкнуть на кнопке **«Да»**;
- щелкните на ярлыке, а затем нажмите клавишу **«Del»**. В диалоговом окне **«Подтвердите удаление»** щелкнуть на кнопке **«Да»**.

6. Стандартные программы WINDOWS.

В комплект поставки **WINDOWS** входит целый набор так называемых «стандартных» компьютерных программ. Это текстовые редакторы - **WordPad** и **Блокнот**, графический редактор - **Paint**, программы - **Калькулятор**, **Номеронабиратель**, **ПроиГрыватель компакт-дисков**, **Просмотр Буфера обмена**, **Проводник**, **Программы связи** и др.

Запуск стандартных программ осуществляется, как правило, с помощью стартового меню - нажатием на кнопке **Пуск**, затем выбора строки **Программы** и в следующем меню - строки **Стандартные**. В открывшемся меню стандартных программ выбирается нужная строка (название программы) и нажатием левой кнопки мыши производится ее запуск. Текстовый редактор Блокнот (Notepad).

Редактор **Блокнот** - позволяет создавать и просматривать короткие неформатируемые текстовые документы, объемом до 64Кб.

При запуске текстового редактора Блокнот открывается стандартное окно, имеющее следующую структуру:

- первая строка включает название документа и название программы;
- вторая строка - командное меню, состоящее из следующей группы команд - **«Файл»**, **«Правка»**, **«Формат»**, **«Вид»**, **«Справка»**;
- ниже **«Рабочее окно»** для набора текста.

Группа команд **Файл** позволяет проводить работу:

- 1) с текстовыми файлами - **«Создать»** новый текстовый файл (выбор этой команды приводит к открытию нового Рабочего окна); **«Открыть»** существующий текстовый документ (в появившемся окне необходимо будет выбрать название папки, набрать имя файла и выбрать тип файла); **«Сохранить»** набранный документ, указав название файла и папки для хранения, изменить уже существующее название, выбрав команду **«Сохранить как»**;
- 2) напечатать готовый документ, подготовив макет страниц. Команды - **«Печать»**, **«Параметры страницы»**.

Группа команд **«Правка»** позволяет отредактировать созданный документ - **«Вырезать»** или **«Удалить»** часть текста, **«Копировать»** или **«Вставить»** часть текста, **«Найти»** и **«Заменить»** необходимое слово в тексте, позволяет вставлять в текст документа текущие дату и время.

Группа команд **«Формат»** позволяет: выбирать шрифт, используя командное меню (команда **«Шрифт»**) и вставлять где необходимо перенос текста по словам.

Группа команд **«Вид»** позволяет иметь только один раздел строки состояния.

Создание и редактирование документа с помощью редактора «Блокнот». При

загрузке текстового редактора открывается чистое рабочее окно, что позволяет сразу же создавать новый текстовый документ.

Отредактированный текст необходимо сохранить с помощью команд **Меню Файл Сохранить** или **Сохранить как**.
Текстовый редактор WordPad.

При запуске текстового редактора **WordPad** открывается стандартное окно, имеющее следующую структуру:

- первая строка состоит из названия документа и названия программы;
 - вторая строка включает командное меню, которое имеет следующие группы команд - «**Файл**», «**Правка**», «**Вид**», «**Вставка**», «**Формат**», «**Справка**»;
 - третьей строкой является «**Панель инструментов**» (каждая клавиша панели соответствует какой-либо команде из командного меню);
 - четвертая строка - строка форматирования, где расположены окна выбора вида шрифта, размера шрифта, кнопки шрифтов: жирный, курсив, подчеркнутый, кнопки выравнивания текста: по левому краю, по середине, по правому краю страницы, кнопка маркеры;
 - под кнопками меню идет измерительная линейка;
 - ниже «**Рабочее окно**» для набора текста.
- Командное меню.

Группа команд «**Файл**» позволяет проводить работу:

- 1) с текстовыми файлами - «**Создать**», «**Открыть**», «**Сохранить**» и «**Сохранить как**»;
- 2) напечатать готовый документ, подготовив макет страниц. Команды - «**Печать**», «**Предварительный просмотр**», «**Макет страницы**» («**Параметры страницы**»);
- 3) открыть те документы, с которыми Вы работали в последний раз, список документов находится в этой группе команд. Открытие документа проводится по выбранному в списке имени.

Группа команд «**Правка**» позволяет отредактировать созданный документ - «**Вырезать**» или «**Удалить**» часть текста, «**Копировать**» или «**Вставить**» часть текста, «**Найти**» и «**Заменить**» необходимое слово в тексте.

Группа команд «**Вид**» позволяет отображать на экране линейки и клавиши по требованию пользователя.

Группа команд «**Вставка**» позволяет вставлять в текст документа текущие дату и время, а также тексты, рисунки, диаграммы и т.д., созданные с помощью других прикладных программ.

Группа команд «**Формат**» позволяет: выбирать шрифт, используя командное меню (команда «**Шрифт**»); помечать строки текста (команда «**Маркер**»); редактировать отдельные абзацы текста, устанавливая отступы от края листа и для красной строки, межстрочные интервалы и расстояния между абзацами (команда «**Абзац**»); набирать текст в несколько колонок (команда «**Табуляция**»).

Создание и редактирование документа с помощью редактора WordPad.

При загрузке текстового редактора открывается чистое рабочее окно, что позволяет сразу же создавать новый текстовый документ.

ПРЕЖДЕ ЧЕМ НАБИРАТЬ ТЕКСТ НА КЛАВИАТУРЕ НЕОБХОДИМО:

- 1) в панели «**Форматирование**» выбрать необходимый шрифт, открыв список шрифтов и выделив нужное название;
- 2) в панели «**Форматирование**» выбрать размер шрифта;
- 3) в измерительной панели выбрать тип отступа;
- 4) в измерительной панели переместить верхний указатель отступа начала абзаца на нужную позицию.

При наборе текста в конце каждого абзаца необходимо нажимать клавишу «**Enter**».

Отредактированный текст необходимо сохранить с помощью команд **Меню Файл Сохранить** или **Сохранить как**.

Для того чтобы *напечатать готовый текст* необходимо:

- 1) установить размеры полей - в меню «**Файл**» выбрать строку «**Макет**» («**Параметры**») страницы, в полях ввода указать размер левого, правого, верхнего и нижнего поля, нажать «**ОК**»;
- 2) просмотреть документ перед печатью - в меню «**Файл**» выбрать пункт «**Предварительный просмотр**»;
- 3) напечатать документ - в меню «**Файл**» выбрать пункт «**Печать**», установить параметры печати, указав, сколько страниц печатать («все», «с ... по...», «выделенный текст»), количество копий при печати, а в группе свойства - размер бумаги, размещение текста на листе и тип подачи бумаги.

Как задавать в данном редакторе **Параметры страницы, Свойства файла и Нумерацию страниц** смотри раздел «Текстовый редактор Word».

Гrafический редактор Paint.

Grafический редактор **Paint** предназначен для рисования несложных эскизов, схем, графиков. Он позволяет создавать, исправлять, масштабировать рисунок и изменять его расположение на листе.

Для запуска программы необходимо открыть стартовое меню, выбрать в нем строку «**Программы**», в следующем меню - «**Стандартные**», затем строку «**Графический редактор Paint**».

Окно редактора состоит из:

- первая строка окна включает название документа и название редактора;
- вторая строка - командное меню;
- в центре окна находится рабочее поле, где можно создавать документы;
- на левом поле - панель инструментов;
- в нижнем поле - палитра цветов.

Командное меню включает следующие группы команд:

«**Файл**» - команды работа с графическими файлами - «**Создать**», «**Открыть**», «**Сохранить**» и «**Сохранить как**» (с новым именем) и команды печати - «**Предварительный просмотр**», «**Параметры страницы**», «**Печать**».

Правка - команды редактирования рисунка, а также команду отмены предыдущего действия и возможность вставки из другого файла текста, рисунка.

Вид - команды настройки вида экрана и подключения панелей, а также масштаб изображения на экране.

Рисунок - команды изменения размеров рисунка («**Атрибуты**»), повороты («**Отразить/Повернуть**») и наклона («**Растянуть/Наклонить**») рисунка, а также изменение цветов на противоположные («**Обратить цвета**») и очистки экрана («**Очистить**»).

Палитра - команды изменения палитры (для создания нужного цвета).

? (Справка) - справочная информация о программы.

Панель инструментов состоит из следующих клавиш: «**Выделение произвольной области**», «**Выделение прямоугольной области**», «**Ластик**», «**Заливка**», «**Выбор цветов**», «**Масштаб**», «**Карандаш**», «**Кисть**», «**Распылитель**», «**Надпись**», «**Линия**», «**Кривая**», «**Прямоугольник**», «**Многоугольник**», «**Эллипс**», «**Прямоугольник со скругленными углами**».

Для того чтобы вставить текст в рисунок необходимо:

- 1) щелкнуть на кнопке «**Надпись**» на «**Панели инструментов**»;
- 2) появившийся значок («**+**») необходимо установить на угол начала надписи и растянуть по ее (надписи) размеру. Область набора называется текстовым блоком;
- 3) в верхней части окна появится текстовое меню, в котором находится окно с названием шрифта, окно с размером шрифта и кнопки для печати жирным

шрифтом, курсивом и подчеркнутым шрифтом. При необходимости Вы можете произвести выбор необходимых параметров надписи

К стандартным программам относятся также следующие программы:

Калькулятор - программа, работающая в режимах простого калькулятора и инженерного для научных расчетов;

Таблица символов - предлагает набор специальных символов, которые можно вставлять в документы;

Просмотр буфера обмена - показывает содержимое буфера обмена - те элементы, которые вы вырезали или скопировали, работая с программами;

Телекоммуникационные средства (*Модем, Прямое соединение, программа связи, Номераонабигатель*) - позволяют соединять между собой компьютеры либо по телефонным линиям, либо с помощью специального кабеля;

Средства для работы с факсом (Создание факсимильного сообщения, Редактор титульных страниц, Запрос факсимильного сообщения);

Средства для работы с мультимедиа (*Лазерный проигрыватель, Универсальный проигрыватель, фонограф, Регулятор уровня*) - позволяют воспроизвести аудио и видео анимационные файлы, создавать, редактировать вставлять и проигрывать звуковые эффекты в документы.

и т. д.

7. Текстовый редактор Word. (первое знакомство).

При запуске текстового редактора **Word**открывается стандартное окно, имеющее следующую структуру:

- первая строка состоит из названия документа и названия программы;
- вторая строка включает командное меню, которое имеет следующие группы команд - «Файл», «Правка», «Вид», «Вставка», «Формат», «Сервис», «Таблица», «Окно», «Справка»;
- третьей строкой является «Панель инструментов» (которую можно установить самостоятельно по своему вкусу «Меню» «Вид» «Панели инструментов»);
 - под кнопками меню идет измерительная линейка;
 - ниже «Рабочее окно» для набора текста;
 - слева от «Рабочего окна» расположена горизонтальная линейка для установки позиций табуляции и размера полосы набора;
 - справа и внизу под окном для набора текста расположены вертикальные и горизонтальные линии прокрутки (соответственно);
 - в нижней части вертикальной полосы прокрутки расположены кнопки перемещения по объектам;
 - слева от горизонтальной полосы прокрутки расположены кнопки режимов просмотра документов (различают следующие режимы: обычный режим, режим веб-документа, режим разметки, режим структуры и режим чтения);

- под горизонтальной полосой прокрутки расположена строка состояния документа. **Параметры страницы.** В текстовом редакторе типа Word параметры страницы

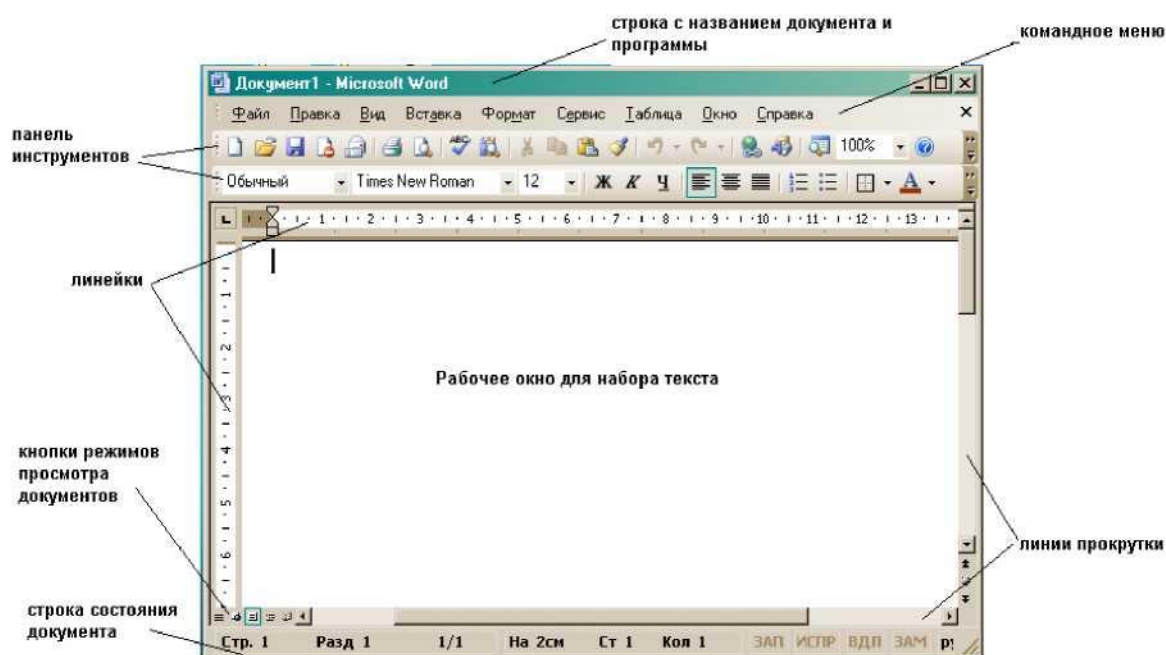


Рис.2

задаются с помощью команд

Меню Файл ^ Параметры Страницы.



Рис. 3

Свойства файла задаются с помощью команды **Меню Файл ^ Свойства. Колонтитул** формируется с помощью команды **Меню ^ Вид ^ Колонтитул.**

Нумерация страниц задается с помощью команды **Меню Вставка ^ Номер Страниц....**

Совет. Если Вы продержите курсор мыши около кнопки на панели инструментов более 3 секунд, то всплывающая подсказка укажет Вам

на ее функции (рис.3.).

Задания:

Часть I

1. Откройте папку «МОИ ДОКУМЕНТЫ».
2. Или на сетевом диске откройте папку «STUDENTS», создайте там папку с номером Вашей группы.
3. Создайте папку, содержащую в её названии Вашу «ФАМИЛИЮ» и «№ ВАШЕГО СТУДЕНЧЕСКОГО БИЛЕТА» в папке «МОИ ДОКУМЕНТЫ». (Вместо символа « / » в вашем номере билета используйте символ « - » тире в создаваемой папке). Например: «Иванова-1245-05»).
4. Создайте папку «Лаб.работа-1» внутри папки «ФАМИЛИЯ-НОМЕР СТУДЕНЧЕСКОГО БИЛЕТА» (далее сокращённо «ФНСБ»).
5. Создайте папку «ПАПКА 1» внутри папки «Лаб.работа-1».
6. Создайте папку «ПАПКА 2» в папке «ПАПКА 1».
7. Создайте файл «Фамилия-НР» в папке «ПАПКА 1» с помощью текстового редактора Блокнот (Notepad) для которого:

- a) выберете фирму или придумайте её сами и напишите текст-резюме, кто Вы и почему Вам хотелось бы работать в этой фирме (не менее пяти предложений примерно на три или более строк);
 - b) найдите «Параметры страницы» в меню «ФАЙЛ» и установите «ПОЛЯ»: верхнее - 35,0 мм, нижнее - 25,0 мм, левое - 35,0 мм, правое - 15,0 мм, «РАЗМЕР БУМАГИ»- 210x297 мм;
 - c) опишите свойства файла «Фамилия-NP», когда создан, когда изменен, время правки, какой его объем и какое у него расширение. Сохраните файл и закройте его.
8. Создайте файл «Фамилия-WP» папке «ПАПКА 1» с помощью текстового редактора WordPad, для которого:
- a) напишите тот же текст, что и в предыдущем случае (не менее пяти предложений примерно на три или более строк);
 - b) найдите «Макет страницы» («Параметры страницы») в меню «ФАЙЛ» и установите «ПОЛЯ»: верхнее - 20,0 мм, нижнее - 25,0 мм, левое - 30,0 мм, правое - 20,0 мм, «РАЗМЕР БУМАГИ»- 210x297 мм;
 - c) опишите свойства файла «Фамилия-WP», когда создан, когда изменен, время правки, какой его объем и какое у него расширение. Сохраните файл и закройте его.
9. Создайте файл «:Фамилия^1» в папке «ПАПКА 1» с помощью текстового редактора WORD, для которого:
- a) напишите то же текст, что и в предыдущих случаях (не менее пяти предложений примерно на три или более строк);
 - b) найдите «Параметры страницы» и установите «ПОЛЯ»: верхнее - 3,0 см, нижнее - 3,7 см, левое - 3,5 см, правое - 2,0 см, «РАЗМЕР БУМАГИ»- 210x297 мм;
 - c) опишите свойства файла «Фамилиями», когда создан, когда изменен, время правки, какой его объем и какое у него расширение. Сохраните файл и закройте его.
10. С помощью команды «Найти» найдите файл «Фамилия-W1.doc».
11. Скопируйте файл «Фамилия-WLdoc» из папки «ПАПКА 1» в папку «ПАПКА 2» одним из известных Вам методов.
12. Переименуйте файл «Фамилия-WLdoc» в папке «ПАПКА 2», назовите его «ФАЙЛ 2.doc».
13. Удалите файл «ФАЙЛ 2.doc» из папки «ПАПКА 2», путём переноса его в корзину.
16. Восстановите файл «ФАЙЛ 2.doc» в папке «ПАПКА 2».
17. Найдите файл «ФАЙЛ 2.doc», с помощью команды «Найти» (или «FIND»), создайте ярлык этого файла на рабочем столе.
18. Переименуйте ярлык файла «ФАЙЛ 2.doc» в Ваше «ИМЯ».
19. Упорядочите ярлыки на рабочем столе по дате (по размеру, по типу).
20. Удалите ярлык файла «ИМЯ-doc» с рабочего стола.

Часть II

1. С помощью текстового редактора Word создайте файл «Фамилия Лаб1^ос» в папке «Лаб.работа-1».
- a) Установите «ПОЛЯ»: верхнее - 3,2 см, нижнее - 3,9 см, левое - 3,5 см, правое - 1,9 см, «РАЗМЕР БУМАГИ»- 210x297 мм).
2. Напишите заголовок: Редактор **WORD**, файл «ИМЯ», шрифт «XXX», размер «XX» и перенесите весь текст из «Фамилия-W1» в этот файл.
 3. Напишите заголовок: Редактор **WordPad**, файл «ИМЯ», шрифт «XXX», размер «XX» и перенесите весь текст из «Фамилия-WP» в этот файл.
 4. Напишите заголовок: Редактор **Notepad**, файл «ИМЯ», шрифт «XXX», размер «XX» и перенесите текст из «Фамилия-NP» в этот файл.
 5. Сравните эти редакторы и укажите, какой из файлов имеет меньший объем памяти.

6. Пункты 3-7 Части II должны уместиться на одной странице. Если они не умещаются на ней, то измените размеры полей в разделе «**Параметры страницы**».
7. На другой странице файла «**Фамилия**» составьте письмо-заказ на создание:
 - а) логотипа фирмы, в которой Вы хотели бы работать;
 - б) визитки этой фирмы.
 - в) примерное содержание письма см. ниже.
8. Пример текста начала письма-заказа:

Кому: **ФАМИЛИЯ ИСПОЛНИТЕЛЯ**, (напр. Исидоров В.Я.)

Отдел дизайна Фирмы «**Картонная коробка**»,

Адрес: Урюпинск, 123468, Забытый тупик, д.12, стр. 12, офис 246.

От: **ВАША ФАМИЛИЯ**, (Иванова Т.И.)

Должность и наименование фирмы,

Адрес: ВАШ АДРЕС.

Посылаю Вам свой набросок логотипа фирмы "Название фирмы". Он должен символизировать "ИДЕЯ"(написать, что символизирует логотип). Прошу развить мою идею и выполнить окончательный вариант логотипа и названия фирмы.

9. В графическом редакторе **Paint** выполните эскиз логотипа (небольшого размера и с ограниченной палитрой цветов, иначе Ваш файл не поместится на дискете) и вставьте рисунок в документ.
10. Продолжение текста письма-заказа:

А также прошу проработать окончательный вариант моей визитки. Эскиз моей визитки прилагается (размер 90X50 мм, см. образец).



11. В графическом редакторе **Paint** выполните эскиз визитки, причём в верхнем левом углу должен быть поставлен Ваш логотип. Вставьте рисунок в документ.
12. Окончание текста письма-заказа (после визитки):
Пожалуйста, доработайте эти эскизы. Учтите, что мы располагаем суммой в XXXXруб.

ПРИМЕЧАНИЕ. Сумма **XXXX руб.** должна составлять 21,75% от 17571 рубля. Откройте калькулятор и подсчитайте сумму, затем скопируйте полученный результат в документ.

13. Закончите свое письмо-заказ таким образом:

Давайте встретимся в XX.XX.X (укажите дату, день недели, время и место) и обсудим этот проект более подробно. Заранее Вам благодарен (на).

Подпись

/Фамилия/

14. Данное письмо с логотипом фирмы и визиткой должно поместиться на 1 странице.
15. Вставьте нижний колонтитул, в котором укажите Автора файла, время его создания и нумерацию страницы.
16. **Покажите созданный отчёт преподавателю.**
17. Выключить компьютер, привести в порядок Ваше рабочее место.

Практическое занятие № 57 «Инсталляция программного обеспечения, его использование и обновление»

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

Содержание работы:

Установка программного обеспечения

Без подходящего, хорошо настроенного программного обеспечения даже самый мощный современный компьютер не будет работать в полную силу, а его реальные возможности останутся не использованными.

Настройка разнообразных программ непосредственно под задачи каждого пользователя является залогом комфортной и уверенной работы на компьютере. Установка программ – широчайшее поле деятельности: количество приложений настолько велико, что сориентироваться в новинках и системных требованиях бывает порой весьма затруднительно.

Запишите в тетрадь:

Установка или инсталляция — процесс установки программного обеспечения на компьютер конечного пользователя.

Деинсталляция - действие, обратное инсталляции; процесс удаления программного продукта с диска, с компьютера

Рассмотрим понятия:

1. Что такое дистрибутив.
2. Типы инсталляции программного обеспечения.
3. Лицензионное соглашение

1. **Дистрибутив** (англ. distribute — распространять) — это набор программ, предназначенный для начальной установки программного обеспечения.

Например, дистрибутив операционной системы обычно содержит программы для начальной инициализации — инициализация аппаратной части, загрузка урезанной версии системы и запуск программы-установщика, программу-установщик (для выбора режимов и параметров установки) и набор специальных файлов, содержащих отдельные части системы (так называемые пакеты).

Запишите в тетрадь:

Дистрибутив - это пакет файлов, изготовленный специально для удобства инсталляции программы в достаточно произвольный компьютер.

Дистрибутив также может содержать **README-файл** (от англ. *read me* — «прочти меня») — текстовый файл, содержащий информацию о других файлах.

2. Инсталляция программного обеспечения

Дистрибутив (ПО) - это комплект (как правило, набор файлов), приспособленный для распространения ПО. Может включать вспомогательные инструменты для автоматической или автоматизированной начальной настройки ПО (установщик).

Так и при использовании дистрибутива программного обеспечения - устанавливаются только необходимые файлы, при чем таким образом, чтобы их правильно видела операционная система. Также конфигурируются начальные параметры, язык, способ подключения, например, к Интернет.

Виды дистрибутивов:

- **Архив** (.zip, .rar, .tar.gz и др.) - неавтоматизированный дистрибутив
- **Исполняемый файл** - дистрибутив с автоматизированным установщиком, позволяет пользователю указать необходимые параметры при установке.
- **Комплект на CD/DVD** - такой дистрибутив, как правило, состоит из нескольких файлов и сопровождается автоматизированным установщиком.

Используется для крупных пакетов ПО и системного программного обеспечения (дистрибутивы ОС, Windows, различные дистрибутивы Linux).

Большинство программ поставляются для продажи и распространения в сжатом (упакованном) 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 включает в себя размещение всех необходимых программе файлов в соответствующих местах файловой системы. Многие программы (включая операционные системы) поставляются вместе с универсальным или специальным инсталлятором — программой, которая автоматизирует большую часть работы, необходимой для их установки.

Запишите в тетрадь:

Инсталлятор — это компьютерная программа, которая устанавливает файлы, такие как приложения, драйверы, или другое ПО, на компьютер. Она запускается из файла SETUP.EXE или INSTALL.EXE

Дистрибутив также может содержать **README-файл** (от англ. *read me* — «прочти меня») — текстовый файл, содержащий информацию о других файлах.

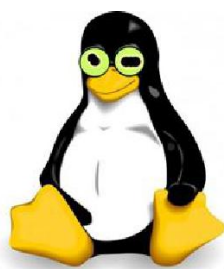
Программы по их юридическому статусу можно разделить на три большие группы:

Запишите в тетрадь:

1. Лицензионные
2. условно бесплатные (shareware)
3. 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.

4. Порядок установки

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



Практическое задание:

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.

Практическое занятие № 58«Изучение пакета прикладных программ»

Цель занятия: отработать практический навык работы с пакетом прикладных программ.

Содержание работы:

Произвести настройку установки MS Office 2007, чтобы она проходила полностью в автоматическом режиме. **Требования:**

1. Компьютер с операционной системой Windows 7.

2. Дистрибутив Microsoft Office 2007.

Задача №1.

1. Запускаем компьютер с установленной операционной системой Windows 7. Заходим в систему.
2. Копируем содержимое дистрибутива в папку "C:\Office2007Distr\"
3. Запускаем программу настройки **C:\Office2007Distr\setup.exe /admin**
4. В окне "Select Product" выбираем "Create a new Setup customization file for the following product" и выбираем "Microsoft Office Enterprise 2007". Жмем "Ок".
5. Переходим в секцию "Setup" и выбираем шаг "Install location and organization name" указываем имя организации ("Organization Name") – Corp.
6. Переходим к шагу "Licensing and user interface" вводим ключ продукта в секцию "ProductKey", принимаем лицензионное соглашение, ставя галочку напротив "I accept the terms in the License Agreement", параметр "Display Level" выставляем в значение "None", оставляем галочку напротив строки "Suppressmodal".
7. Шаг "Remove previous installation" оставляем без изменений.
8. Переходим в секцию "Features" шаг "Set feature installation states". Выбираем "Microsoft Office Groove", жмем правой кнопкой и указываем параметр "Not Available".
9. По окончании настроек жмем File -> SaveAs указываем имя конфигурационного файла enterprisecustom.msp и сохраняем его в папке "C:\Office2007Distr"
10. Выходим из мастера установки.
11. Производим установку приложения в автоматическом режиме, запуская команду C:\Office2007Distr\setup.exe /config
C:\Office2007Distr\Enterprise.ww\config.xml
12. По окончании установки проверяем, т.е. ярлык для установки Microsoft Office Groove.
13. Запускаем файл настройки, выполнив команду msixec /p
C:\Office2007Distr\enterprisecustom.MSP

Практическое занятие № 59«Работа со специализированным программным обеспечением»

Цель занятия: изучить технологию работы со специализированным программным обеспечением.

Содержание работы:

Порядок выполнения работы:

Компьютерные способы хранения и обработки звуковой информации получают в последнее время все большее распространение. Звукооператоры увидели в компьютере мощное средство для более комфортной реализации своих традиционных функций и, самое главное, множество принципиально новых, ранее неизвестных возможностей.

Действительно, современный компьютер может служить хорошей аппаратной основой для обработки звуковой информации: характерная (тактовая) частота последних процессоров превышает максимальные звуковые частоты не менее чем на 5 порядков², так что при такой скорости можно организовать весьма сложную обработку данных, включая автоматические преобразования в масштабе реального времени.

Чаще всего звуковые колебания с помощью микрофона легко преобразуются в электрические. Сигнал от микрофона очень слаб и нуждается в усилении, что на современном уровне развития техники проблемы также не представляет. Форму полученных колебаний, т.е. зависимость интенсивности сигнала от времени, можно наблюдать на экране осциллографа

АЦП — аналого-цифровой преобразователь.

принципы работы АЦП:

Во-первых, АЦП производит дискретизацию записываемого звукового сигнала по времени.

Во-вторых, АЦП производит дискретизацию амплитуды звукового сигнала.

В качестве практической иллюстрации к проблеме выбора параметров цифровой записи звука можно заглянуть в списки форматов и атрибутов программы “Звукозапись”, входящей в состав Windows (для получения изображенного диалогового окна необходимо в

При воспроизведении записанного в компьютерный файл звука производится преобразование в противоположном направлении — из дискретной цифровой формы представления сигнала в непрерывную аналоговую, поэтому вполне естественно соответствующий узел компьютерного устройства называется **ЦАП** — **цифроаналоговый преобразователь**. Процесс реконструкции первоначального аналогового сигнала по имеющимся дискретным данным нетривиален, поскольку *никакой* информации о форме сигнала между соседними отсчетами не сохранилось. В разных звуковых картах для восстановления звукового сигнала могут использоваться различные способы. Наиболее наглядный и понятный из них состоит в том, что по имеющимся соседним точкам рассчитывается некоторая гладкая функция, проходящая через заданные точки, которая и принимается в качестве формы аналогового сигнала. Технические возможности современных микросхем позволяют для реконструкции формы сигнала производить весьма сложные вычисления. Выпускаются даже специализированные микропроцессоры, для которых в технической литературе принято название DSP (*Digital Signal Processor*) — *процессоры цифровой обработки сигналов*.

Звуковые файлы могут иметь различные форматы:

Формат AU. Этот простой и распространенный формат на системах Sun и NeXT (в последнем случае, правда, файл будет иметь расширение SND). Файл состоит из короткого служебного заголовка (минимум 28 байт), за которым непосредственно следуют звуковые данные. Широко используется в Unix-подобных системах и служит базовым для Java-машины.

Формат WAVE (WAV). Стандартный формат файлов для хранения звука в системе Windows. Является специальным типом другого, более общего формата RIFF (*Resource Interchange File Format*); другой разновидностью RIFF служат видеофайлы AVI. Файл RIFF составлен из блоков, некоторые из которых могут, в свою очередь, содержать другие вложенные блоки; перед каждым блоком данных помещается четырехсимвольный идентификатор и длина. Звуковые файлы WAV, как правило, более просты и имеют только один блок формата и один блок данных. В первом содержится общая информация об оцифрованном звуке (число каналов, частота дискретизации, характер зависимости громкости и т.д.), а во втором — сами числовые данные. Каждый отсчет занимает целое количество байт (например, 2 байта в случае 12-битовых чисел, старшие разряды содержат нули). При стереозаписи числа группируются парами для левого и правого канала соответственно, причем каждая пара образует законченный блок — для нашего примера его длина составит 4 байта. Такая, казалось бы, излишняя структурированность позволяет программному обеспечению оптимизировать процесс передачи данных при воспроизведении, но, как в подобных случаях всегда бывает, выигрыш во времени приводит к существенному увеличению размера файла.

Формат MP3 (MPEG Layer3). Это один из форматов хранения аудиосигнала, позднее утвержденный как часть стандартов сжатого видео. Природа получения данного формата во многом аналогична уже рассмотренному нами ранее сжатию графических данных по технологии JPEG (см. билет № 19). Поскольку произвольные звуковые данные обратимыми методами сжимаются недостаточно хорошо, приходится переходить к методам необратимым: иными словами, базируясь на знаниях о свойствах человеческого слуха, звуковая информация “подправляется” так, чтобы возникшие искажения на слух

были незаметны, но полученные данные лучше сжимались традиционными способами. Это называется **адаптивным кодированием** и позволяет экономить на наименее значимых с точки зрения восприятия человека деталях звучания. Приемы, применяемые в MP3, непросты для понимания и опираются на достаточно сложную математику, но зато обеспечивают очень значительный эффект сжатия звуковой информации. Успехи технологии MP3 привели к тому, что ее применяют сейчас и во многих бытовых звуковых устройствах, например, плеерах и сотовых телефонах.

Формат MIDI. Название MIDI есть сокращение от *Musical Instrument Digital Interface*, т.е. цифровой интерфейс для музыкальных инструментов. Это довольно старый (1983 г.) стандарт, объединяющий разнообразное музыкальное оборудование (синтезаторы, ударные, освещение). MIDI базируется на пакетах данных, каждый из которых соответствует некоторому событию, в частности, нажатию клавиши или установке режима звучания. Любое событие может одновременно управлять несколькими каналами, каждый из которых относится к определенному оборудованию. Несмотря на свое изначальное предназначение, формат файла стал стандартным для музыкальных данных, которые при желании можно проигрывать с помощью звуковой карты компьютера безо всякого внешнего MIDI-оборудования. Главным преимуществом файлов MIDI является их очень небольшой размер, поскольку это не детальная запись звука, а фактически некоторый расширенный электронный эквивалент традиционной нотной записи. Но это же свойство одновременно является и недостатком: поскольку звук не детализирован, то разное оборудование будет воспроизводить его по-разному, что в принципе может даже заметно исказить авторский музыкальный замысел.

Формат MOD. Представляет собой дальнейшее развитие идеологии MIDI-файлов. Известные как “модули программ воспроизведения”, они хранят в себе не только “электронные ноты”, но и образцы оцифрованного звука, которые используются как шаблоны индивидуальных нот. Таким способом достигается однозначность воспроизведения звука. К недостаткам формата следует отнести большие затраты времени при наложении друг на друга шаблонов одновременно звучащих нот.

Задание1

Осуществить запись звука.

Процесс редактирования звуков:

- свойства (преобразование формата файла);
- правка (вставка звука в другой документ или связывание с ним);
- вставить в файл (в установленную позицию исходного звука);
- смешать с файлом/буфером (микширование звуков);
- удалить до/после текущей позиции;
- увеличить/уменьшить громкость;
- увеличить/уменьшить скорость;
- добавить эхо;
- обратить.

Практическое занятие № 60 «Инсталляция и настройка программного обеспечения компьютерных систем»

Цель занятия: инсталляция и настройка программного обеспечения компьютерных систем.

Содержание работы:

Установить на индивидуальных компьютерах программу распознавания текста ABBYY FineReader 6.0 Professional с помощью специальной программы-тренажера

ИНСТАЛЛЯЦИЯ.

Для этого выполнить следующие действия:

- Открыть папку *Практические занятия/ Практическое занятие 1/ Инсталлятор_Тренажер* на **Рабочем столе** вашего компьютера

- Запустить файл *setup.exe*

- выбрать полную версию установки

- принять лицензионное соглашение (иначе процесс не пойдет дальше), нажать

Далее

- ввести имя пользователя (свою фамилию), название организации (УМТК)

- выбрать папку для размещения файлов программы (согласиться с предлагаемым вариантом C:/Program Files)

- ввести код инсталляции **ABBY-1234-5678-9012-3456** (вводится только при выборе полной версии)

- выбор типа инсталляции (полная, типичная, выборочная), выбрать *Типичная*, нажать *Далее*

- нажать *Установить*

- подождать, пока пройдет копирование файлов на жесткий диск

- подождать, пока пройдет создание программной группы и ярлыков в главном меню

- подождать, пока пройдет создание записи в реестре для обеспечения возможности удаления программы через **Панель управления**

- нажать *Готово*

ПРОВЕРКА.

Выполнить проверку правильности выполнения задания.

Для этого выполнить следующие действия:

- Вызвать Главное меню (Нажать кнопку «Пуск»)

- В меню "*Программы*" найти программную группу "*ABBY FineReader 6.0 Тренажер*"

- запустить программу *FineReader.exe* через ярлык "*ABBY FineReader 6.0 Тренажер.lnk*"

- Показать результат преподавателю

- нажать *Выход*

ДЕИНСТАЛЛЯЦИЯ.

Выполнить деинсталляцию программы распознавания текста *ABBY FineReader 6.0 Professional*.

Для этого выполнить следующие действия:

- Вызвать Главное меню (Нажать кнопку «Пуск»)

- В меню "*Программы*" найти программную группу "*ABBY FineReader 6.0 Тренажер*"

- запустить программу *FineReader.exe* через ярлык "*Настройка и удаление ABBY FineReader 6.0 Тренажер.lnk*"

- Выбрать *Удалить*

- Подтвердить *Удаление*

- Показать результат преподавателю

- Нажать *Готово*.

Тип программного обеспечения по способу доступа и условиям использования

Задание 1

Законспектировать следующие сведения

Рассмотрим программное обеспечение (ПО) по способу доступа к нему и условиям использования. Вы знаете, что некоторые программы находятся в свободном доступе, их, например, можно бесплатно скачать из интернет, установить на своем компьютере и беспрепятственно пользоваться. Но также есть и такие программы, которые требуют оплаты, их установка по «украденному» коду, как мы уже говорили, незаконна.

Итак, по способу доступа ПО делится на следующие части.

Проприетарное ПО (от англ. proprietary – собственность) – это программы, все права на которые принадлежат собственнику (организации или отдельному лицу), и это закрепляется соответствующим договором (лицензией). Возможность пользования такими программами нужно приобретать, иначе их использование считается незаконным и наказывается. Часто такие программы называют коммерческими или лицензионными, но это не совсем верно. Коммерция может распространяться не только на проприетарное ПО, так же как и лицензионная защита.

Приведите примеры таких программ.

Примеры: _____

Свободное ПО (СПО, free software) - это свободно распространяемое, бесплатное ПО, за использование которого не накажут. Такая его суть должна декларироваться самими разработчиками. Для законного обеспечения этого разработана лицензия *GNU General Public License* (далее — *GPL*), которая не только гарантирует свободу, но и защищает её: она допускает дальнейшее распространение программ только под той же лицензией.

Примеры: _____

Открытое ПО или ПО с открытым кодом (open source software) - это ПО, доступное для редактирования всем желающим (и могущим), дорабатываемое и изменяемое ПО. Часто понятия свободного и открытого ПО относятся к одним и тем же программам, но бывает и иначе, поэтому путать их нельзя.

Примеры: _____

Условно-свободное ПО (shareware software) - это программы, занимающие положение где-то между свободным и проприетарным ПО. К ним относятся версии проприетарных программ «для ознакомления», демоверсии, распространяемые в рекламных целях, программы "для домашнего использования" ("free for home users" или "freeware for personal non-commercial use").

Примеры: _____

Кроссплатформенное свободное ПО - это программы, работающие под различными операционными системами (например, под Windows и под Linux). Примеры: офисный пакет OpenOffice.org, пакет программ для работы в Интернет Mozilla, редактор растровой графики GIMP, ...

Как узнать, какое ПО мы используем? Это мы можем узнать в строке меню программы с помощью пунктов *Помощь/ О программе*

. Задание 2

Установить способ доступа и условия использования указанных программ

<i>Программа</i>	<i>Как войти</i>	<i>Сведения о программе</i>
ОС Windows	ПК мыши на <i>Мой компьютер</i> , см. вкладки <i>Общие</i>	
Internet Explorer	<i>Главное меню</i> или <i>Ярлык на Рабочем столе</i>	
Stamina.exe	<i>Ярлык на Рабочем столе</i>	
Crossword	<i>Ярлык на Рабочем столе</i>	
Hvost.exe	<i>Мои документы/Lesson1</i>	

Переписать в конспект первую колонку таблицы и заполненную вами третью колонку.

Экзаменационные билеты по дисциплине

ОП.02 Архитектура компьютерных систем

Билет 1

1. Основные характеристики ЭВМ
2. Сумматоры
3. Являются ли эквивалентными следующие высказывания:

$$x \mid (y \oplus z) \quad \text{и} \quad (x \mid y) \vee (x \mid z)$$

Билет 2

1. Триггеры
2. Внешняя память
3. Построить таблицу истинности, найти СНДФ, найти минимальную ДНФ.

для высказывания: $\left(\overline{(A \wedge B)} \Rightarrow A \right) \Rightarrow A \vee B$

Билет 3

1. ОЗУ и ПЗУ: назначение и основные характеристики
2. Арифметические операции в алгебре логики
3. Перевести числа 210 и 30 из десятичной системы счисления в двоичную, произвести их сложение и деление в двоичном коде

Билет 4

1. Третье поколение ЭВМ
2. Минимизация логических функций
3. Укажите, в каких случаях высказывание истинно, а в каких ложно:

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

Билет 5

1. Шифраторы и дешифраторы
2. Последовательный интерфейс ввода-вывода
3. Являются ли эквивалентными следующие высказывания:

$$(\overline{A \vee B}) \vee (\overline{B} \wedge \overline{A}) \quad \text{и} \quad ((A \vee B) \oplus \overline{B}) \Rightarrow A$$

Билет 6

1. Классификация средств ЭВТ
2. Параллельный интерфейс ввода-вывода
3. Построить таблицу истинности, найти СНДФ, найти минимальную ДНФ.

для высказывания: $(\overline{z} \Rightarrow y) \Leftrightarrow (\overline{z} \vee \overline{x})$

Билет 7

1. Логические элементы
2. Режимы работы процессора
3. Для чисел 27_{10} и 13_{10} записать прямой, обратный и дополнительный код

Билет 8

1. Системы счисления, применяемые в ЭВМ
2. Модули памяти
3. Укажите, в каких случаях высказывание истинно, а в каких ложно:

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

Билет 9

1. Виртуальная память
2. Регистры

3. Являются ли эквивалентными следующие высказывания:

$$\overline{(A \Rightarrow B)} \wedge (\bar{B} \Leftrightarrow \bar{A}) \text{ и } ((A \Rightarrow B) \wedge \bar{B}) \oplus A$$

Билет 10

1. Центральный процессор
2. Оценка производительности вычислительных систем
3. Построить таблицу истинности, найти СНДФ, найти минимальную ДНФ.

для высказывания: $(x|y) \rightarrow (x|z)$

Билет 11

1. Страничная память
2. Многомашинные и многопроцессорные вычислительные системы
3. Построить таблицу истинности, найти СНДФ, найти минимальную ДНФ.

для высказывания: $\overline{(A \wedge B)} \Leftrightarrow (\bar{B} \oplus \bar{A}) \Leftrightarrow (A \vee B) \oplus (A \oplus \bar{B})$

Билет 12

1. Первое поколение ЭВМ
2. Многопрограммная работа ЭВМ
3. Укажите, в каких случаях высказывание истинно, а в каких ложно:

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

Билет 13

1. Коды чисел
2. Порты USB
3. Являются ли эквивалентными следующие высказывания:

$$\overline{(A \wedge B)} \Leftrightarrow (\bar{B} \oplus \bar{A}) \text{ и } (A \vee B) \oplus (A \oplus \bar{B})$$

Билет 14

1. Сегментированная память
2. Порты SCSI
3. Решить задачу средствами алгебры логики.
Намечаются экскурсии в три города А, В и С. Руководитель фирмы сказал: «Неверно, что если будет экскурсия в город В, то не будет экскурсии в город С. Если будет экскурсия в город С, то не будет экскурсии в город А.» В какие города будет проводиться экскурсия?

Билет 15

1. Системные платы
2. Большие интегральные схемы
3. Построить таблицу истинности, найти СНДФ, найти минимальную ДНФ.

для высказывания: $((x \downarrow y) \rightarrow z) \oplus y$

Билет 16

1. Второе поколение ЭВМ
2. Иерархия памяти
3. Укажите, в каких случаях высказывание истинно, а в каких ложно:

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

Билет 17

1. Модификация памяти типа SRAM
2. Ввод-вывод
3. Упростить логическую формулу: $\bar{x} \wedge \bar{y} \rightarrow x \vee (x \wedge y)$.

Билет 18

1. Машинные коды
2. Классификация и характеристика запоминающих устройств

3. С помощью таблиц истинности проверить, являются ли равносильными формулы

$$x \rightarrow (\bar{x} \wedge \bar{y}) \text{ и } \overline{\bar{x} \vee x \vee y}.$$

Билет 19

1. Формы представления чисел в ЭВМ
2. Энергонезависимая память
3. Укажите, в каких случаях высказывание истинно, а в каких ложно:

$$\overline{(z \rightarrow x) \leftrightarrow (y|x)}$$

Билет 20

1. Большие интегральные схемы
2. Режимы работы процессоров
3. Построить таблицу истинности, найти СНДФ, найти минимальную ДНФ.

для высказывания: $\overline{(x|\bar{y}) \oplus (z \rightarrow \bar{x})}$

Билет 21

1. Основные характеристики процессоров
2. Счетчики
3. Являются ли эквивалентными следующие высказывания:

$$(x|y) \rightarrow (x|z) \text{ и } (\bar{z} \vee y) \rightarrow (\bar{z} \oplus \bar{x})$$

Билет 22

1. Мультиплексоры
2. Шины
3. Перевести числа 35 и 20 из десятичной системы счисления в двоичную, произвести их вычитание и умножение в двоичном коде

Билет 23

1. Четвертое поколение ЭВМ
2. Микросхемы системной логики
3. Начертить схему логического устройства для выражения

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

Билет 24

1. Вычислительные системы и их архитектура
2. Модификация памяти типа DRAM
3. Построить таблицу истинности, найти СНДФ, найти минимальную ДНФ.

для высказывания: $(x \vee \bar{y}) \rightarrow (\bar{z} \oplus \bar{x})$

Билет 25

1. Защита памяти
2. Системные ресурсы
3. Начертить схему логического устройства для выражения

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

КРИТЕРИИ ОЦЕНКИ

Даны полные исчерпывающие ответы на все вопросы – «отлично».

Даны полные ответ, с небольшими недочетами – «хорошо».

Даны ответы на вопросы, раскрывающие только общие понятия – «удовлетворительно».

Количество вариантов задания для экзаменуемого – 25

Время выполнения задания – 20 минут.

Экзаменационная ведомость.

ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

№ п.п.	Содержание изменения	Дата, номер протокола заседания педагогического совета
1	2	3
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		