

Федеральное государственное бюджетное образовательное учреждение высшего образования
**«Петербургский государственный университет путей сообщения
Императора Александра I»**
(ФГБОУ ВО ПГУПС)

Петрозаводский филиал ПГУПС

ОДОБРЕНО

на заседании цикловой комиссии

протокол № 11 от 10.06.2017

Председатель цикловой комиссии:

 Петрозаводский филиал ПГУПС

УТВЕРЖДАЮ

Начальник УМО



А.В. Калько

«13» 06 2017г.

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
по организации и проведению практических
занятий/лабораторных работ

По дисциплине/МДК/ПМ: ОП.03 Архитектура аппаратных средств

Специальность: 09.02.02 Компьютерные сети

Выполнил (а): Усков А.А. – Преподаватель Петрозаводского
филиала ПГУПС.

2017г.

Практическое занятие №1

Тема: Представление логических выражений в виде схем из логических элементов.

Цель: Научиться строить схемы логических элементов в программе Logisim.

Краткие теоретические сведения.

Logisim — это образовательный инструмент для разработки и моделирования цифровых логических схем. Её написал и продолжает активную разработку преподаватель Hendrix College, США, Carl Burch. Logisim является свободным программным обеспечением и распространяется на условиях лицензии GNU GPL. Это означает, что вы можете свободно использовать и распространять её в любых целях, читать и редактировать её исходный код, и распространять модифицированные версии программы на тех же условиях.

Интерфейс Logisim полностью переведён на русский язык, программа распространяется с подробнейшей справочной системой на русском языке.

Интерфейс программы состоит из строки меню, панели инструментов, панели проводника, таблицы атрибутов и холста.

Панель проводника содержит список схем проекта и все инструменты загруженных библиотек. Любые инструменты можно выносить на панель инструментов. Состав панели инструментов настраивается в окне параметров проекта, которое вызывается из пункта «Параметры» меню «Проект».

На холсте располагаются все компоненты редактируемой в данный момент схемы. Холст покрыт сеткой, по которой выравниваются все компоненты и провода схемы. Таблица атрибутов содержит атрибуты выделенного в данный момент инструмента или компонента на холсте.

Лучший способ быстро освоить базовые навыки работы с Logisim пройти все шаги «Пособия начинающего» из меню «Справка».

Logisim позволяет не только редактировать цифровые схемы, но и моделировать их поведение. Если флаг «Моделирование включено» из меню «Моделировать» установлен, то Logisim просчитывает процессы, происходящие в схеме прямо во время её редактирования: меняются значения на входах и выходах устройств, обновляется внутреннее состояние устройств памяти, устройства вывода отображают соответствующую информацию, а провода меняют цвет в зависимости от проходящих по ним значений.

Для схем, на которых присутствуют тактовые генераторы, можно либо осуществлять моделирование потактово (пункт «Один такт» из меню «Моделировать»), либо установить максимальную тактовую частоту (пункт «Тактовая частота») и запустить продолжительное моделирование схемы (пункт «Такты включены»). Вернуть схему в исходное состояние можно с помощью пункта «Сбросить моделирование».

Подсхемы

Одна из важнейших возможностей Logisim — создание подсхем. Идея подсхем аналогична идее подпрограмм в программировании: они служат для разбиения большой работы на более мелкие части, для повторного использования уже спроектированных частей схем и для упрощения процесса отладки.

Добавить схему можно с помощью пункта «Добавить схему...» из меню «Проект». Одна из схем в проекте обязательно является главной — она открывается первой при загрузке проекта. После запуска моделирования будет выполняться моделирование схемы, просматриваемой в данный момент. Чтобы включить схему в качестве подсхемы в другую схему, нужно выделить её и расположить на холсте, как и любой другой элемент. При этом образуется иерархия вложенных подсхем, верхним уровнем которой является просматриваемая схема.

Чтобы редактировать схему, нужно дважды щёлкнуть на её названии в панели обозревателя. Располагать несколько копий одной схемы — совершенно нормальная практика. При этом редактирование этой схемы будет приводить к изменениям во всех копиях, но моделирование поведения каждой копии происходит автономно.

Передача сигналов между какой-либо схемой и расположенной в ней подсхемой происходит через контакты. Контакты бывают входные и выходные. По умолчанию на панели инструментов присутствуют оба вида контактов. Также компонент «Контакт» есть во встроенной библиотеке «Проводка».

В цифровой технике при соединении элементов проводами всегда определено, какой элемент передаёт сигнал, а какой принимает.

Входной контакт подсхемы будет принимать сигнал из внешней схемы и передавать его в подсхему. Выходной контакт подсхемы будет передавать сигнал из подсхемы во внешнюю схему.

Контакты подсхемы при расположении её копии на холсте другой схемы будут располагаться с той стороны «корпуса» подсхемы, в какую сторону направлен контакт (атрибут «Направление») внутри подсхемы.

Очень важно давать чёткие осмысленные имена контактам (атрибут «Метка»), тогда подключать к расположенной на холсте копии подсхемы провода будет очень легко.

Для верхней схемы в иерархии всегда показывается реальное текущее состояние во время моделирования. Однако чтобы наблюдать текущее состояние конкретной копии подсхемы, расположенной в иерархии ниже, нужно щёлкнуть правой кнопкой мыши на этой копии на холсте и выбрать в контекстном меню пункт «Рассмотреть...».

Logisim имеет небольшой встроенный редактор векторной графики для изменения внешнего вида подсхем при добавлении их в другие схемы. Он также позволяет изменять расположение контактов подсхемы.

Этот редактор вызывается через пункт «Редактировать внешний вид схемы» из меню «Проект» или нажатием соответствующей кнопки на панели инструментов.

Провода

Располагать на холсте провода можно инструментами «Правка» или «Проводка». Провод и контакт элемента считаются соединёнными, если они располагаются строго в одной и той же точке сетки. То же относится и к соединению провода с другим проводом. При перемещении по холсту уже подключенных элементов, Logisim будет сохранять соединения, если это возможно. Этого можно избежать, если при перетаскивании нажать клавишу Shift.

Ещё одна важная функция Logisim — возможность собирать провода в пучки. Пучки обозначаются на схемах чёрным цветом.

Очень многие компоненты библиотеки имеют атрибут «Разрядность данных». Этот атрибут может принимать значение от 1 до 32, и если он больше 1, то подключение провода к соответствующему контакту компонента автоматически создаст пучок проводов. Естественно, нужно соединять контакты с одинаковой разрядностью данных.

Разрядность данных можно назначать также входным и выходным контактам схем, что значительно уменьшает количество проводов на схеме, а значит, повышает её удобочитаемость.

Когда возникает потребность разделить пучок проводов на отдельные провода или более мелкие пучки (или наоборот — собрать их в один пучок), нужно применить компонент «Разветвитель» из встроенной библиотеки «Проводка». Этот компонент также позволяет назначать в каком порядке провода входят в пучок — это настраивается в таблице атрибутов разветвителя.

Провода в Logisim могут иметь один из семи цветов, каждый из которых несёт

определённую информацию о проводе.

- Серый — разрядность провода неизвестна (он не подключен ни к каким компонентам)
- Синий — провод несёт однобитный сигнал, но его состояние не определено.
- Тёмно-зелёный — провод несёт однобитный сигнал, в данный момент этот сигнал — «0».
- Светло-зелёный — провод несёт однобитный сигнал, в данный момент этот сигнал — «1».
- Чёрный — провод несёт многобитный сигнал (его разрядность больше единицы). При этом любое количество отдельных однобитных проводов в пучке могут иметь неопределённое состояние.
- Красный — провод содержит значение «ошибка». Это происходит, когда провод замыкает сразу несколько выходов компонентов, и на выходах разное значение сигнала. На практике это означает короткое замыкание.
- Оранжевый — провод подключён к выходам компонентов, имеющим разную разрядность.

Выделение цветом проводов, несущих «0» или «1» позволяет наблюдать изменение значений на выходах компонентов и эффективно отлаживать схему.

Комбинационный анализ

Ещё одна замечательная функция Logisim — возможность проводить анализ и синтез комбинационных логических устройств.

Если схема не содержит последовательностных логических устройств, контактов разрядностью больше 1 и количество входных и выходных контактов не больше 12, то для схемы можно провести комбинационный анализ. Это можно сделать через пункт «Анализировать схему» из меню «Проект». В открывшемся окне «Комбинационный анализ» присутствуют вкладки «Входы», «Выходы», «Таблица», «Выражение» и «Минимизация».

Вкладка «Таблица» содержит полную таблицу истинности, полученную для схемы; вкладка «Выражение» содержит список выражений каждой из булевых функций для выходных контактов.

Вкладка «Минимизация» содержит минимизированные таблицы истинности и выражения для булевых функций. И выражения и таблицы истинности можно свободно редактировать, копировать и вставлять.

Открыв то же окно «Комбинационный анализ» из меню «Окно», можно ввести во вкладках «Входы» и «Выходы» имена входных и выходных контактов для будущей схемы; ввести таблицу истинности или выражения булевых функций, нажать кнопку «Построить схему», и указать название новой схемы. Logisim синтезирует схему, используя логические элементы.

Библиотека компонентов

Огромным преимуществом программы Logisim является обширная библиотека компонентов. Далее даётся краткое описание каждого компонента библиотеки. Если компонент является реализацией типичного для цифровой техники устройства, то никакого описания не приводится. Для получения подробной информации по каждому компоненту следует обратиться ко встроенной справке (пункт «Справка по библиотеке...» из меню «Справка»).

Библиотека «Проводка»

- Разветвитель — позволяет разделять пучки на отдельные провода и объединять отдельные провода в пучки.
- Контакт — передаёт значения между подсхемой и содержащей её внешней схемой.
- Датчик — отображает значение в точке схемы, к которой подключен. Позволяет выбирать основание системы счисления для отображения.
- Тоннель — позволяет соединять удалённые точки схемы без протягивания провода.
- Согласующий резистор — меняет в определённую сторону значение на проводе, к которому подключен, если значение на нём не определено.
- Тактовый генератор — формирует прямоугольные импульсы.
- Константа — источник постоянного значения²⁹
- Питание — источник логической единицы
- Земля — источник логического нуля
- Транзистор — полевой транзистор, работающий в режиме электронного ключа
- Передаточный вентиль — комбинация двух комплементарных полевых транзисторов
- Расширитель битов — преобразует значение в значение с другой разрядностью. При увеличении разрядности позволяет выбирать, чем заполнять старшие биты.

Библиотека «Элементы»

- Элемент НЕ
- Буфер — повторитель сигнала («элемент «ДА»)
- Элемент И
- Элемент ИЛИ
- Элемент И-НЕ
- Элемент ИЛИ-НЕ
- Элемент Исключающее ИЛИ
- Элемент Исключающее ИЛИ-НЕ
- Нечётность — на его выходе единица, если единица на нечётном количестве входов
- Чётность — на его выходе единица, если единица на чётном количестве входов
- Управляемый буфер — позволяет создавать на его выходе неопределённое значение
- Управляемый инвертор — то же, что и управляемый буфер, но выполняющий инвертирование выхода.

Библиотека «Плексоры»

- Мультиплексор
- Демультиплексор
- Декодер
- Шифратор приоритетов — выдаёт номер старшего входа, на котором «1»
- Селектор битов — разделяет многоразрядный вход на несколько групп битов, и пускает на выход выбранную управляющим входом группу

Библиотека «Арифметика»

- Сумматор
- Вычитатель
- Множитель
- Делитель
- Отрицатель — выдаёт значение в дополнительном коде, противоположное по знаку входному значению
- Компаратор — сравнивает два числа
- Сдвигатель — реализация левого и правого сдвига
- Сумматор битов — выдаёт количество единиц, поступающих на его

многоразрядные входы

- Искатель битов — выдаёт номер старшего или младшего нуля или единицы в многоразрядном входном значении.

Библиотека «Память»

- D-триггер
- T-триггер
- JK-триггер
- RS-триггер
- Регистр
- Счётчик
- Сдвиговый регистр
- Генератор случайных чисел
- ОЗУ
- ПЗУ

Библиотека «Ввод/вывод»

- Кнопка
- Джойстик
- Клавиатура — выдаёт ASCII коды введённых пользователем символов
- Светодиод
- 7-сегментный индикатор
- Шестнадцатеричный индикатор — 7-сегментный индикатор, выводящий шестнадцатеричное значение, поступающее на его 4-разрядный вход.
- Светодиодная матрица
- Терминал — отображает поступающие на его вход ASCII коды в31 виде строк текста

Библиотека «Базовые»

- Инструмент Нажатие — инструмент для взаимодействия с компонентами схемы: нажатия кнопок, движения джойстика, установки значений контактов, просмотра состояния пучков проводов, и т.д.
- Инструмент Правка — основной инструмент для редактирования схем: перемещения компонентов и добавления проводов. Этот инструмент совмещает в себе функции четырёх инструментов, описанных ниже (можно считать их устаревшими).

- Инструмент Выбор — инструмент для выделения элементов схемы.
- Инструмент Проводка — инструмент для добавления проводов.
- Инструмент Текст — инструмент для добавления текстовых полей и редактирования меток компонентов.
- Инструмент Меню — инструмент для вызова контекстного меню компонентов.
- Метка — инструмент для добавления текстовых меток в любую точку холста.

Различные возможности

В этом параграфе приводится список различных дополнительных возможностей Logisim с краткими описаниями. Более подробную информацию о каждой возможности можно получить в соответствующем разделе справки.

Любой файл проекта Logisim можно загрузить в другой проект в качестве библиотеки. В таком случае все его схемы будут доступны как компоненты, но изменять их внутреннюю структуру будет нельзя. Это осуществляется через пункт «Загрузить библиотеку: Библиотека Logisim...» из меню «Проект».

Существует возможность писать на языке Java собственные библиотеки компонентов, обладающие всем набором функций, доступных в Logisim. Руководство по написанию таких библиотек — в разделе «Библиотеки JAR» руководства пользователя Logisim. Моделировать поведение схем можно без запуска графического интерфейса пользователя Logisim.

Существует множество параметров для запуска и автоматизированного моделирования схем из командной строки. Подробнее — в разделе «Проверка из командной строки» руководства пользователя Logisim.

Растровые изображения схем можно экспортировать в файлы — как индивидуально, так и группами. Это осуществляется через пункт «Экспортировать изображение...» из меню «Файл».

Любой инструмент можно привязать к определённой комбинации клавиш Shift, Control и Alt и кнопок мыши. Это настраивается на вкладке «Мышь» окна параметров проекта (пункт «Параметры...» из меню «Проект»).

Любой файл проекта Logisim можно использовать как шаблон для создания новых проектов. Это удобно, если у вас есть файл проекта с настроенной панелью инструментов, поведением мыши, и т.д. Это осуществляется на вкладке «Шаблон» окна настроек приложения (пункт «Настройки...» из меню «Файл»).

Графические обозначения логических элементов можно выбрать на вкладке

«Международные» окна настроек приложения.

Отображение фактической тактовой частоты моделирования можно включить на вкладке «Окно» окна настроек приложения.

Полную статистику по типам и количеству компонентов, содержащихся в схеме и входящих в её состав подсхемах можно получить через пункт «Получить статистику схемы» из меню «Проект».

Существует возможность просматривать в виде дерева иерархию подсхем и перемещаться по ней. Для перехода в такой режим просмотра нужно выбрать пункт «Показать дерево моделирования» из меню «Проект», или нажать соответствующую кнопку на панели инструментов. Можно проследивать распространение сигналов по схеме; это может быть удобно при отладке схем. Для этого нужно в меню «Моделировать» снять флажок «Моделирование включено», затем выбрать пункт «Сбросить моделирование», и далее последовательно выбирать пункт «Шаг моделирования», наблюдая распространение сигнала.

Изменение во времени значений на контактах и некоторых компонентах схемы можно автоматически записывать в виде таблицы в файл. Подробнее — в разделе «Запись в журнал» руководства пользователя Logisim.

Задачи:

По структурной формуле построить схему и составить таблицу истинности:

1. $\text{НЕ } A$
2. $A * B$
3. $A + B$
4. $A * \text{НЕ } B$
5. $A * \text{НЕ } B + \text{НЕ } A * B$. Какое название носит эта логическая ф-я?
6. $\sim a \sim b + \sim a b + a \sim b + b a$

Практическое занятие №2

Тема: Цифровой логический уровень. Комбинационные устройства.

Цель: Спроектировать в Logisim комбинационное устройство, имеющее несколько входов и несколько выходов.

Количество входов и выходов, поведение устройства приведено в таблице ниже. Вариант определяется по последней цифре номера студента в списке группы.

Под «значением на входах» и «значением на выходах» понимается некоторое целое число, двоичное представление которого формируется отдельными входными или выходными битами.

Каждая из реализаций устройства должна быть составлена в Logisim на логических элементах в виде схемы с осмысленными названиями входов и выходов.

Отчет

Отчет по работе содержит схему в файле с расширением *.circ (программа Logisim), таблицу истинности для разрабатываемого комбинационного устройства.

Вариант	Номера в списке группы	Количество входов	Количество выходов	Поведение устройства
1	0,1	4	1	На выходе «1» если значение на входах в двоичном представлении симметрично. В остальных случаях «0»
2	2,3	3	1	На выходе «0» когда на входе число больше 3. В остальных случаях «1»
3	4,5	2	4	Когда на входах значение «3» — на всех выходах единицы. В любом другом случае единица только на одном выходе. Номер этого выхода задаётся значением на входах.
4	6,7	3	1	На выходе «1», когда на входах нет ни одной пары нулей, идущих подряд. В остальных случаях — «0».
5	8,9	3	2	Значение на выходах — номер самого младшего входа, несущего «0».

Практическое занятие №3

Тема: Простые устройства памяти. Триггеры. Защелки

Цель: Понять принципы работы устройств памяти. Реализовать простейшие устройства памяти.

Все задания выполняются в одном файле проекта Logisim; каждое самостоятельное

устройство должно быть оформлено в виде отдельной схемы с осмысленным названием входов и выходов, а также самой схемы.

Теоретическая часть:

Для временного хранения информации в цифровых схемах применяют различные устройства памяти. Из простейших устройств памяти мы рассмотрим три класса: триггеры, регистры и счётчики. Все эти устройства есть во встроенной библиотеке Logisim «Память». Они могут быть и асинхронными, но в рамках данного курса мы будем рассматривать только синхронные.

У каждого синхронного устройства есть синхронизирующий (тактовый) вход. Любое изменение внутреннего состояния устройства происходит только в тот момент, когда уровень сигнала на тактовом входе меняется (в таком случае говорят, что тактовый вход срабатывает).

Триггер — простейшее устройство памяти, хранящее один бит информации. Иными словами, триггер может иметь только два разных внутренних состояния — «0» или «1». В англоязычной литературе триггер называют «flip-flop». Существует четыре вида триггеров: D (data), T (toggle), JK (jump-kill) и RS (reset-set). Эти названия даны по названиям входов триггеров. Кроме этих входов каждый триггер имеет два выхода — Q (прямой) и Q' (инверсный); значение на прямом выходе всегда совпадает со внутренним состоянием триггера, а значение на инверсном — противоположное.

Каждый из четырёх типов триггеров имеет разное поведение. Таблицы истинности для них приведены в таблице 3.2. Q' означает значение, противоположное значению, хранимому в триггере в данный момент.

Таблица 3.2. *Триггеры*

D-триггер		T-триггер		JK-триггер			RS-триггер		
D	Q	T	Q	J	K	Q	S	R	Q
0	0	0	Q	0	0	Q	0	0	Q
1	1	1	Q	0	1	0	0	1	0
				1	0	1	1	0	1
				1	1	Q	1	1	?

Можно дать словесное описание поведения триггеров:

- D-триггер: когда тактовый вход срабатывает, значение, хранящееся в триггере, мгновенно становится значением входа D (данные).
- T-триггер: когда тактовый вход срабатывает, значение, хранящееся в триггере, меняется или остаётся прежним в зависимости от того, какое значение на входе T (переключение): «1» или «0».
- JK-триггер: когда тактовый вход срабатывает, значение, хранящееся в триггере, меняется, если на входах J и K единица; остаётся прежним, если на них 0; если значения на них различны, то значение становится единицей, если на входе J (прыжок) — «1»; или нулём, если на входе K (забой) — «1».
- RS-триггер: когда тактовый вход срабатывает, значение, хранящееся в триггере, остаётся неизменным, если на входах R и S — «0»; становится «0», если на входе R (сброс) — «1», и становится «1», если на входе S (установка) — «1». Поведение не определено, если на обоих входах «1». (В Logisim значение триггера остается неизменным.)

Физически триггеры реализуются на логических элементах (то есть в конечном итоге на транзисторах в составе интегральных схем), включенных, как правило, не совсем обычным для них способом — их выходы так или иначе соединяются с их входами. Как говорилось выше, триггеры могут быть синхронными и асинхронными. Тактовый вход триггеров и других устройств памяти в Logisim обозначается треугольником; в

случае, когда вход нужно пометить буквой или строкой, используют «С» или «Clock». Синхронные триггеры, как правило, содержат большее количество логических элементов.

Иногда синхронными ошибочно называют также особую разновидность триггеров, имеющих разрешающий вход. Изменение внутреннего состояния такого триггера происходит, когда на разрешающем входе «1».

Этот вход иногда называют синхронизирующим, но на самом деле он таковым не является. На схемах ниже разрешающий вход отмечен буквой «Е» (от англ. enable).

Асинхронные триггеры (с разрешающим входом или без него) иногда называют «прозрачными» (чаще в англоязычной литературе — “transparent”), а синхронные — «непрозрачными» (англ. “non-transparent” или “opaque”). Это связано с тем, что если на разрешающий вход (если таковой имеется) «прозрачного» триггера подать единицу, то помимо записи в память триггера, входной сигнал будет непосредственно подаваться на выход триггера (то есть можно сказать, что триггер будет работать в качестве повторителя). Если при этом сигнал на входе зависит от сигнала на выходе (то есть выход и вход триггера связаны через внешнюю схему, не содержащую синхронных устройств), то схема начнёт возбуждаться. Иными словами, если в схеме есть своего рода «замкнутый круг», то чтобы предотвратить возбуждение схемы, нужно «разорвать» этот круг хотя бы в одном месте синхронным устройством (например, синхронным триггером).

На схемах представлены различные реализации триггеров на логических элементах (рис. 3.1, рис. 3.2, рис. 3.3, рис. 3.4). На рисунке 3.4 показан D-триггер, использующий специальную «надстройку» над RS-триггером, делающую его синхронным.

Однако существует общая техника построения синхронных триггеров из триггеров с разрешающим входом — двухступенчатые триггеры (англ. master-slave flip-flop). В таком триггере соединены два триггера, на разрешающие входы которых подаются противоположные значения. Двухступенчатый синхронный D-триггер показан на рисунке 3.5. При поступлении на его тактовый вход переднего фронта, обновится только состояние на выходе первой ступени, а обновление состояния всего триггера произойдёт при заднем фронте.

При моделировании триггеров на логических элементах в Logisim, следует установить флажок «Добавить шум к задержкам компонентов» в окне параметров проекта. Это нужно сделать для того, чтобы имитировать неравномерность реальных схем.

Например, RS-триггер, построенный с использованием двух элементов ИЛИ-НЕ, будет возбуждаться без этой случайности, так как оба элемента будут обрабатывать свои входы «нога в ногу». До подачи первого значения на входы, состояние некоторых триггеров будет неопределённым, и

Logisim будет показывать значение ошибки на выходах. Это особенности работы Logisim, на практике таких проблем не возникает.

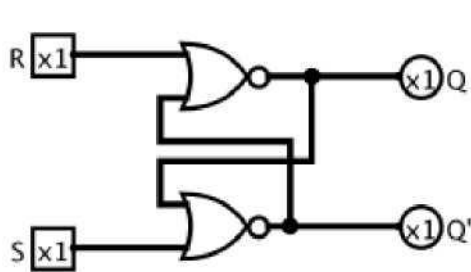


Рис. 3.1. Асинхронный RS-триггер

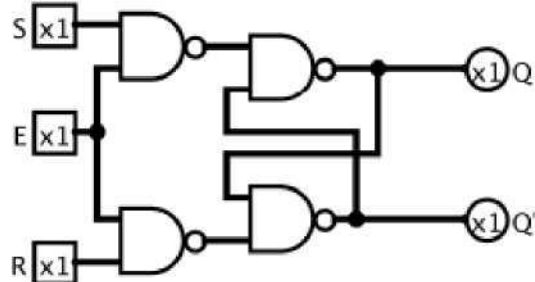


Рис. 3.2. RS-триггер с разрешающим входом

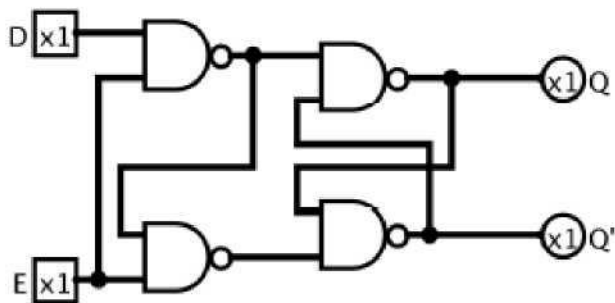


Рис. 3.3. D-триггер с разрешающим входом

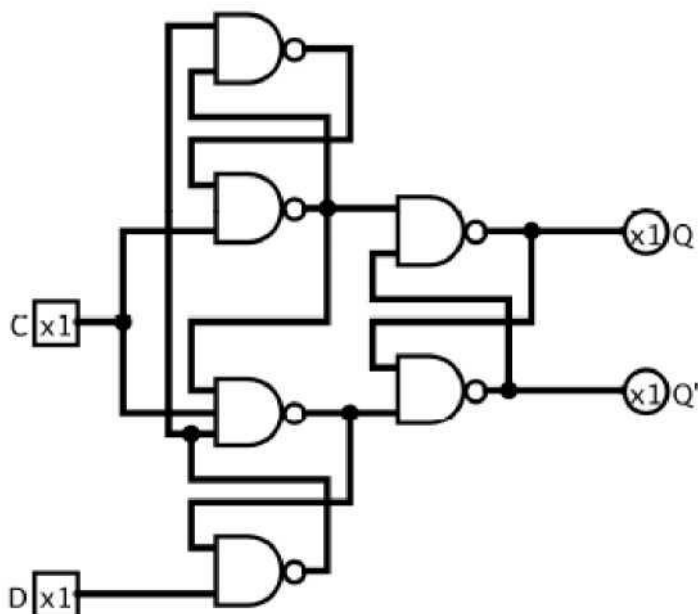


Рис. 3.4. Синхронный D-триггер

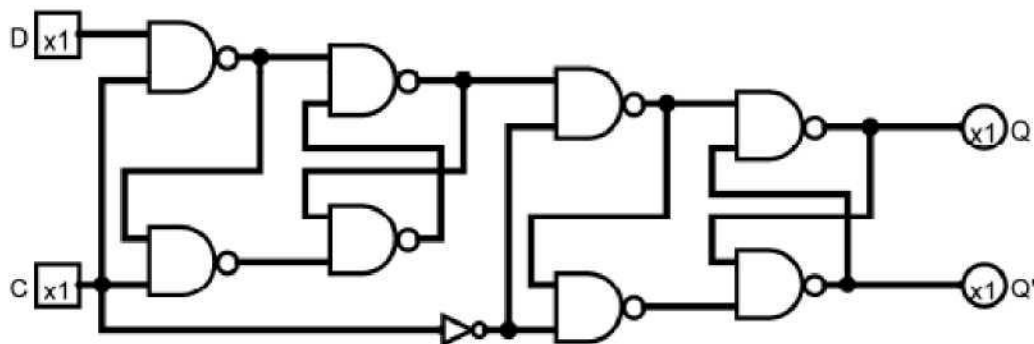


Рис. 3.5. Двухступенчатый синхронный D-триггер

Регистр — последовательностное логическое устройство, используемое для хранения n -разрядных (многобитных) двоичных слов (чисел) и выполнения преобразований над ними. Мы будем использовать регистры только для хранения информации (компонент Logisim «Регистр»). Такой регистр имеет многобитный вход для загрузки данных, тактовый вход и многобитный выход, на который всегда поступает значение, сохранённое в регистре. В первом приближении можно представить себе синхронный регистр для

хранения n -разрядных слов как группу из n синхронных D-триггеров с соединёнными тактовыми входами.

Счётчик — последовательностное логическое устройство, на выходы которого поступает двоичный код (многобитное значение), определяемый числом поступивших на его тактовый вход импульсов. Компонент Logisim «Счётчик» по сути является регистром, который меняет хранимое значение на единицу при поступлении очередного тактового импульса. Кроме того, у него имеются два дополнительных входа («загрузка» и «счёт»), которые позволяют выбирать его поведение — увеличивать значение, уменьшать его, или работать как обычный регистр.

Чтобы сделать в Logisim триггер, регистр или счётчик асинхронным, нужно установить для его атрибута «Срабатывание» значение «Высокий уровень». В таком случае тактовый вход будет работать как Рис. 3.5.

Двухступенчатый синхронный D-триггер разрешающий, и устройство станет «прозрачным».

Более подробную информацию о работе устройств памяти вообще и в Logisim в частности можно найти на страницах справки по библиотеке Logisim (библиотека «Память»).

Задание:

1. Реализовать в Logisim асинхронный RS-триггер на логических элементах. Убедиться в том, что его поведение соответствует описанному в таблице и совпадает с поведением RS-триггера из встроенной библиотеки Logisim. Значения на входных контактах можно изменять инструментом «Нажатие».
2. Повторить задание 1 для RS-триггера с разрешающим входом.
3. Спроектировать асинхронный 8-разрядный регистр на основе D-триггера из встроенной библиотеки компонентов Logisim.
4. *(повышенной сложности). Спроектировать 4-разрядный счётчик на основе D-триггера и комбинационного устройства собственной разработки, имеющего 4 входа и 4 выхода. Это комбинационное устройство выдаёт на выходах 4-разрядное значение на единицу большее, чем 4-разрядное значение на его входах, а при подаче максимального значения выдаёт ноль.

Лабораторная работа №1

Тема: Идентификация основных узлов и логических блоков персонального компьютера и разъемов для подключения внешних устройств. Объяснение их назначения, принципов работы и взаимодействия.

Цель: Идентифицировать основные узлы персонального компьютера.

Краткие теоретические сведения

Шина (bus) –это общий канал связи, используемый в компьютере и позволяющий соединить два и более системных компонента. Название «Шина» пришло из электротехники. Физически шина –это совокупность линий (проводников на материнской плате) по которым обмениваются информацией компоненты и устройства PC.

Обычно шина имеет места для подключения внешних устройств.

Существует иерархия шин PC : более медленная шина соединена с более быстрой шиной.

Современные компьютерные системы включают в себя 3, 4 и более шин.

Каждое системное устройство соединено с какой либо шиной.

Архитектура шины

Линии шины делятся на 3 группы в зависимости от типа передаваемых данных:

- Линии данных (шина данных)
- Линии адреса (шина адреса)
- Линии управления (шина управления)

Шина данных

Контроллер шины осуществляет управление процессом обмена данными

По этой шине происходит обмен данными между CPU (центральным процессором), картами расширения и памятью.

Режим DMA (Direct Memory Access) – режим прямого доступа к памяти.. В этом режиме обмен данными с памятью осуществляется специальным контроллером минуя CPU

Шина адреса

Каждый компонент PC, каждая ячейка RAM (памяти) имеют свой адрес и входят в общее адресное пространство PC.

Для адресации к какому либо устройству РС служит шина адреса по которому передается уникальный идентификационный код (адрес).

Для ускорения обмена данными используется оперативная память - RAM .

Объём адресуемой памяти = 2^n

Где n- число линий шины адреса

Шина управления

Для того, чтобы данные были записаны (считаны) в регистры устройств подключенных к шине, необходим ряд служебных сигналов записи/считывания, готовности к приёму/ передаче данных, подтверждения приема/ передачи и др.

Все эти сигналы передаются по шине управления

Основные характеристики шины

Шина характеризуется разрядностью и частотой работы.

Разрядность (ширина) шины- определяется количеством данных параллельно проходящих через неё. Первая шина ISA была 8 разрядной. Это означало, что для передачи данных использовалось 8 линий и одновременно можно было передать 8 бит информации.

Под разрядностью шины понимается разрядность шины данных.

Пропускная способность шины

Пропускная способность шины определяется количеством бит информации передаваемых по шине за секунду.

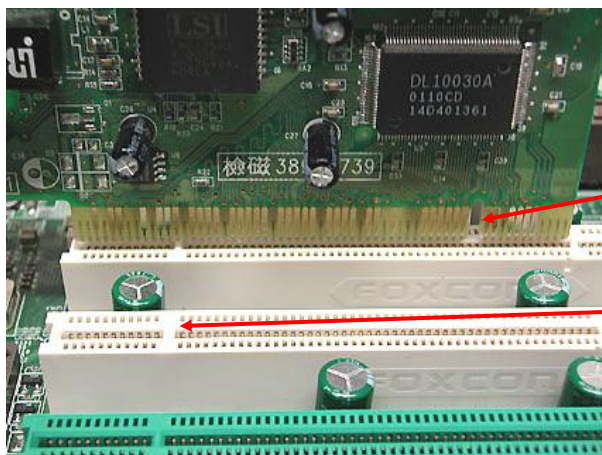
Пропускная способность шины равна произведению разрядности шины на тактовую частоту шины.

Например для шины ISA

$(16 \text{ бит} \times 8,33 \text{ МГц}) : 8 = 133,28 \text{ Мбит/сек} =$

16,6 Мбайт/сек

Шина PCI

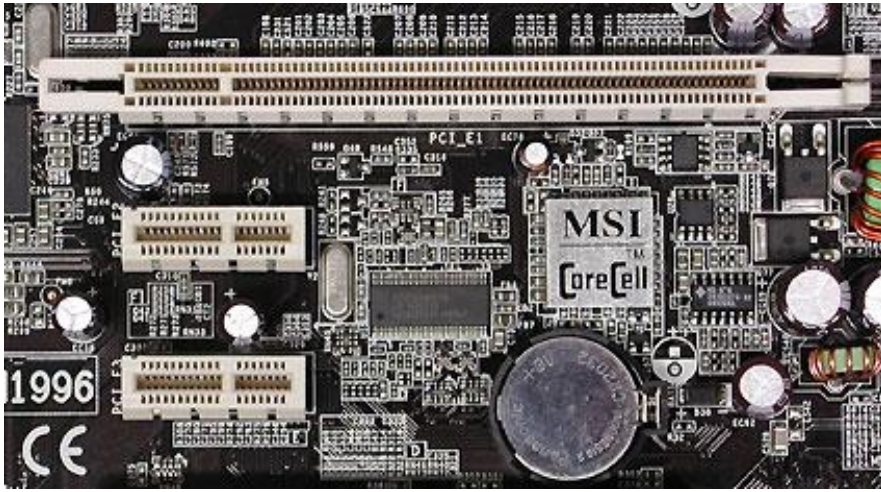


Ключ (вырез) для
питания 5 Вольт

Ключ (вырез) для
питания 3.3 Вольт

Спецификация PCI 2.1 сегодня предусматривает напряжение питания 3,3 В. Левый вырез/выступ предотвращает установку старых 5-В карт, которые показаны на иллюстрации. Универсальные карты имеют два выреза

Шина PCI Express



Слоты расширения материнской платы: PCI Express x16 линий (сверху) и 2 PCI Express x1 линия (снизу).

Задания:

1. Ознакомьтесь с предложенными материнскими платами и определите, какие шины используют слоты расширения на материнской плате.
2. Ознакомьтесь с предложенными платами расширения. Определите напряжение питания и используемую шину.
3. Включите компьютер. Зайдите в Setup. Найдите в меню опции , относящиеся к шине PCI. Объясните их.
4. Подготовьте отчет о проделанной работе.

Контрольные вопросы:

1. Что такое «Шина» компьютера?
2. Какую архитектуру имеет шина? Назначение компонентов шины.
3. Что такое разрядность шины? Как рассчитать производительность шины?
4. Приведите основные технические характеристики шины ISA (разрядность и частоту)
5. Чем отличалась шина EISA от шины ISA? Почему на материнской плате для ISA EISA использовался один и тот же разъем?
6. Что означает термин «локальная шина»? Для каких целей использовалась локальная шина на материнской плате?
7. Приведите основные характеристики шины PCI
8. Чем принципиально отличается шина PCI express от шины PCI

Лабораторная работа №2

Тема: Выполнение сборки и разборки системного блока. Занятие 1

Цель: Ознакомиться с узлами и компонентами системного блока

Научиться: устанавливать в корпус блок питания, материнскую плату, дисковод, CD-привод жёсткий диск, подключать указанные устройства

Краткие теоретические сведения

Системный блок состоит из следующих узлов: корпуса (кейса), блока питания, материнской (системной) платы, дисководов гибких дисков (FDD), жесткого диска (HDD), CD-привода и плат расширения. Платы расширения - сетевая карта, звуковая карта, видеокарта устанавливаются в разъемы на материнской плате. К разъемам материнской платы подключаются коннекторы индикаторов и кнопок передней панели корпуса.

Общий вид системного блока в разобранном состоянии показан на рис 1.

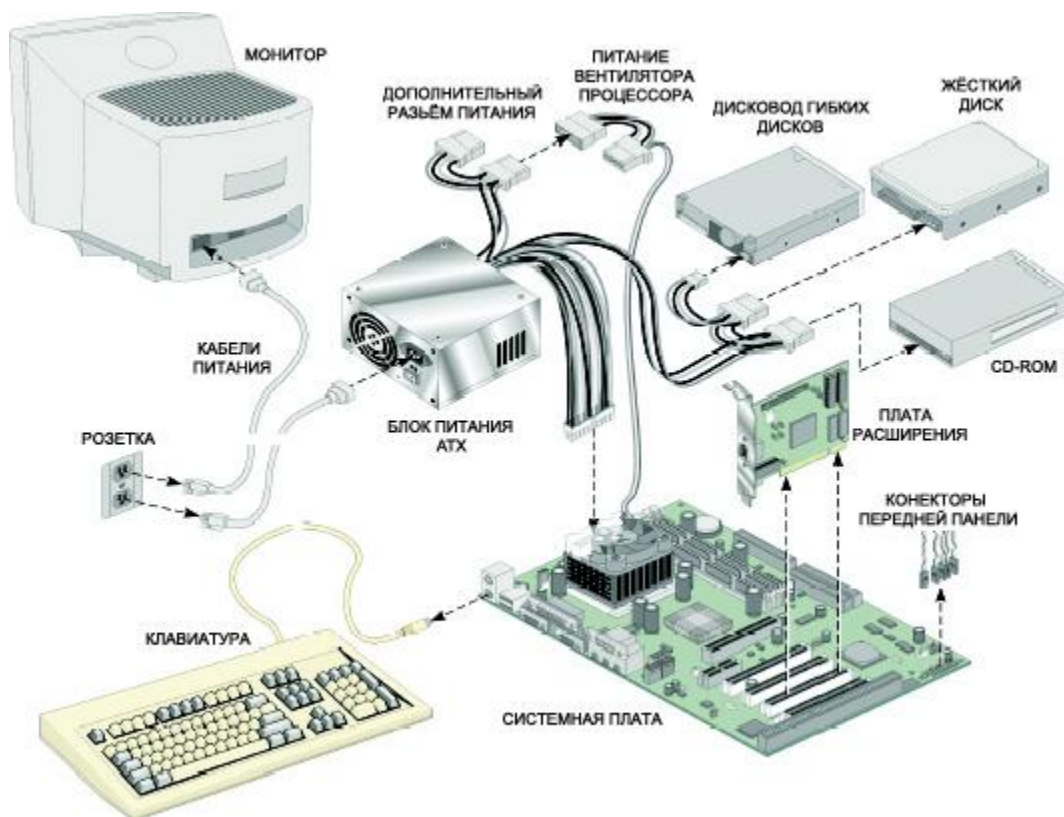


Рис.1 Узлы системного блока

Корпус системного блока

Существует огромное количество корпусов системных блоков отличающихся размерами, формой и другими параметрами. Важнейшей характеристикой корпуса является его форм-фактор, определяющий, какие материнские платы и блоки питания могут быть установлены в данный корпус. Существует несколько форм-факторов:

AT. Впервые использован фирмой IBM в 1984 году. В корпусе форм-фактора AT можно устанавливать полноразмерные платы форм-фактора AT и уменьшенные платы Baby AT. У всех корпусов этого класса сзади имеется круглое отверстие для подключения разъема клавиатуры к материнской плате. Корпуса AT выпускались в двух модификациях рассчитанных на работу с разными источниками питания: Desktop AT и Tower AT. В настоящее время такие корпуса с блоками питания больше не выпускаются, но в эксплуатации находится значительное количество таких компьютеров.

Baby AT Потребность в уменьшении размеров систем привела к тому, что появились миниатюрные корпуса и, соответственно уменьшенные материнские платы стандарта AT. Подобные корпуса выпускались и продавались до 2000 года

LPX. Корпоративные потребители требовали выпуска плоских системных блоков. В результате появился новый форм фактор LPX. Однако он имел ряд недостатков. Потребовалось установить дополнительные вентиляторы, был затруднен доступ к материнской плате. Выпуск указанных корпусов в настоящее время прекращен.

ATX. Недостатки указанных выше стандартов вынудили компанию Intel разработать новый форм фактор ATX. Расположение компонентов в форм факторе ATX было изменено таким образом, чтобы обеспечить удобство доступа и качество охлаждения. Важнейшее качество ATX заключается в том что корпуса

избавились от отверстий для клавиатуры и портов и были заменены единой панелью ввода вывода стандартного размера и расположения.

В зависимости от назначения выпускаются различные по размерам платы стандарта ATX. В таблице 2 показаны размеры плат и совместимость их с корпусами.

Материнская плата	ширина мм	высота мм	подходит к корпусам					
ATX	305	244						
mini- ATX	284	208						
microATX	244	244						
FlexATX	229	191						
NLX	346	229						
WTX	425	356						

Размер и ориентация корпуса

Корпуса могут иметь различные размеры и ориентацию. Плоский настольный корпус (low-profile desktop), стандартный настольный корпус (standart desktop), микротуэр (microtower) , мини-туэр (mini-tower), мидитауэр (miditower), и просто туэр (full tower).

Количество и расположение отсеков для накопителей (drive bays)

может быть разным. Во всех современных корпусах имеется по крайней мере один 3-дюймовый отсек с внешним доступом, предназначенный для установки дисководов, один 5-дюймовый отсек для CD-ROM и один 3-дюймовый внутренний отсек для жёсткого диска.

Блоки питания

Блоки питания также должны быть совместимы с выбранным корпусом (case). Мощность блока питания зависит от используемой материнской платы и плат расширения. Обычно для модернизации кейс поставляется вместе с блоком питания. Блоки питания подразделяются на две большие группы: стандарта AT и стандарта ATX. Выходные напряжения и разъёмы у этих блоков отличаются .

Материнские платы

Материнская (системная) плата является главным устройством в системном блоке, так как именно она формирует компьютер. Она обеспечивает связь всех компонентов компьютера, включая процессор, память, диски, видеокарты, звуковые карты, клавиатуру, мышь, и другие периферийные устройства.

Платы расширения.

Большинство современных материнских плат имеют встроенные контроллеры видео, звука, сетевую карту. Однако в случае необходимости можно поставить дополнительные платы расширения. Чаще всего это видеокарта. В этом случае внутренний видеоконтроллер отключается и используется видеокарта установленная в слот расширения. Чаще всего это AGP или PCI express.

В этой работе необходимо провести сборку системного блока из предложенных комплектующих и проверить его на работоспособность. Особое внимание необходимо обратить на подключение блока питания. В случае ошибочного включения разъёма блока AT (это возможно) будет выведена из строя материнская плата.

Не применяйте больших усилий при подключении шлейфов. Убедитесь, что штырьки (pin) точно попадают в отверстия разъёма. Питание FDD не допускает обратного включения. Но сам разъём

выполнен из пластика и применив усилие можно подать питание наоборот. Тщательно проверьте правильность подключения питания. Порядок сборки системного блока показан на рис. 2,3,4,5

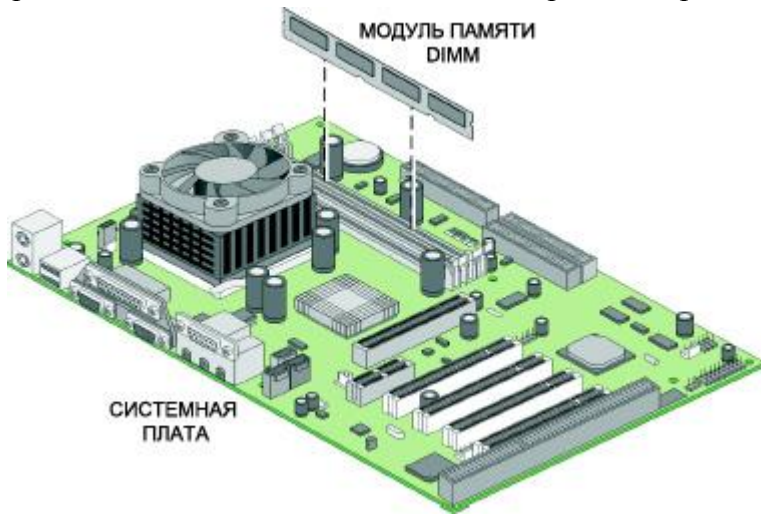


Рис.2 Установка модуля памяти в материнскую плату

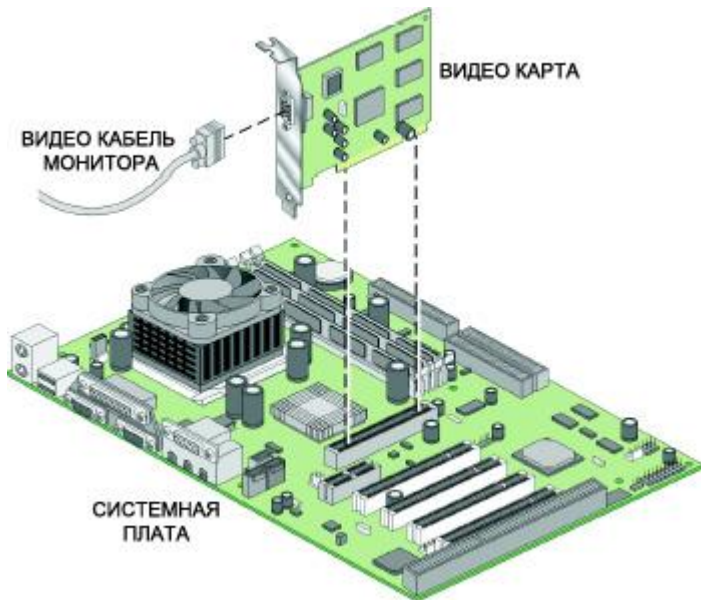


Рис 3 Установка видеокарты в материнскую плату.

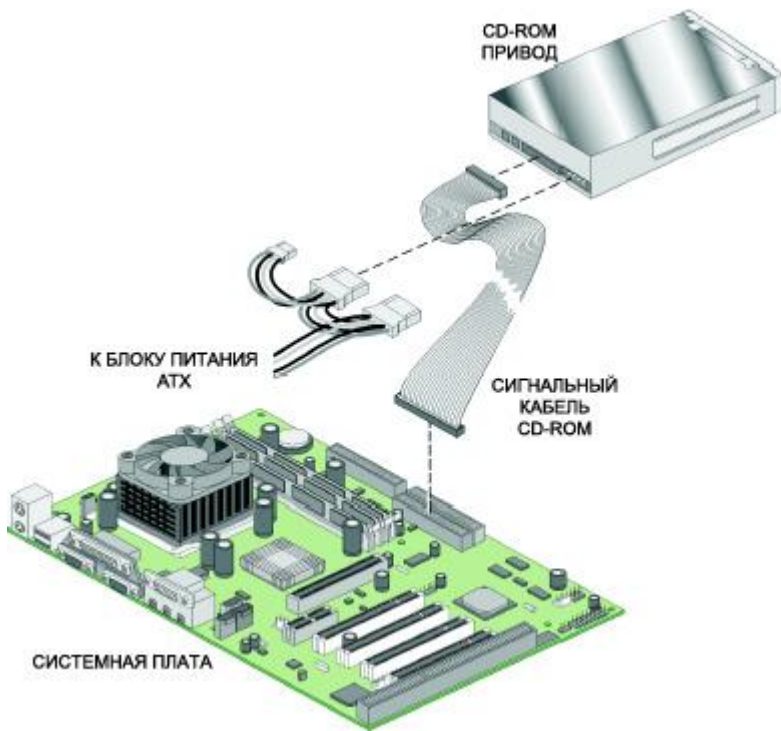


Рис. 4 Подключение CD-ROM привода к материнской плате.

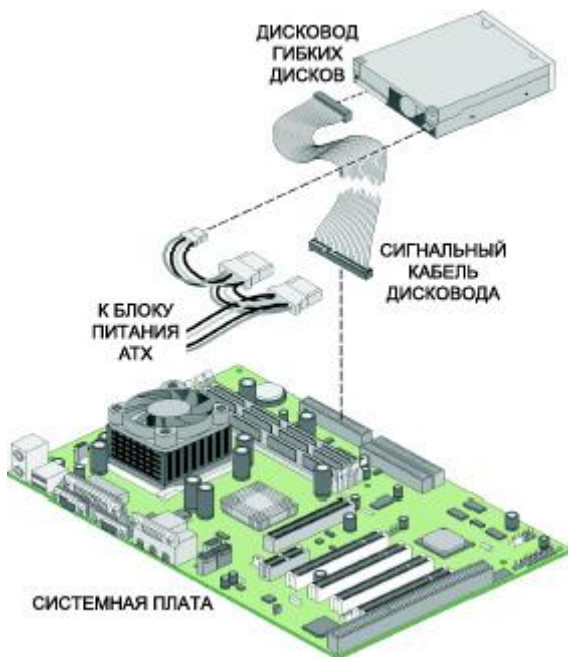


Рис. 5 Подключение дисковод к материнской плате

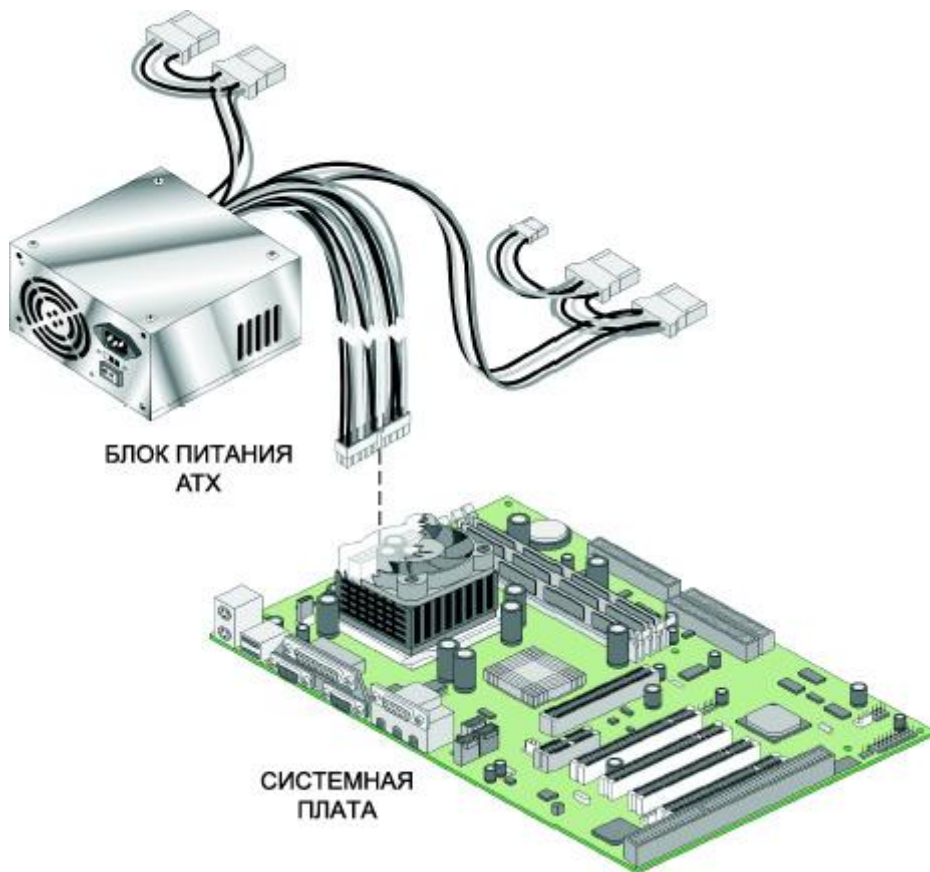


Рис. 6 Подключение блока питания к материнской плате



Рис. 7 Подключение жёсткого диска к материнской плате

Задания

1. В пустой кейс установите и закрепите материнскую плату. Внимательно ознакомьтесь с маркировкой разъемов для подключения индикаторов : Power Led, HDD LED; выключателя Power on, Reset, динамика Speaker, разъемов для подключения HDD, FDD. Обратите внимание на положение первого контакта на

разъёмах. Обычно он маркируется белым треугольником или на разъёме указываются номера контактов. Иногда он выделяется пайкой в виде квадрата.

2. Подключите к разъёмам материнской платы проводники от устройств размещенных на передней панели

3. Закрепите в кейсе FDD и HDD.

4. Подключите кабеля данных от материнской платы к HDD и FDD

5. Закрепите в кейсе CD-привод

6. Подключите CD-привод к материнской плате. (если материнская плата имеет два разъёма с маркировкой IDE1 и IDE2, то CD-привод подключается к разъёму IDE 2. Причём джампер на задней панели CD-привода выставляется в положение «Master». Если на материнской плате один разъём IDE, то жёсткий диск выставляется «Master», а CD-привод в положение «Slave» и подключаются одним шлейфом IDE

7. Закрепите в кейсе блок питания. ВНИМАНИЕ! Соблюдайте осторожность! Падение блока питания на материнскую плату может вывести её из строя.

8. Подключите разъёмы источника питания к материнской плате и к остальным устройствам .

9. Подключите к системному блоку монитор, мышь и клавиатуру. Включите компьютер . Правильно собранный компьютер должен загрузить операционную систему.

10. Подготовьте отчёт

Лабораторная работа №3

Тема: Изучение BIOS и CMOS Setup компьютера

Цель: Ознакомиться с аппаратной реализацией BIOS

Научиться устанавливать основные параметры CMOS Setup

Перечень необходимых средств обучения :

компьютер, материнская плата, микросхема ПЗУ

Краткие теоретические сведения

BIOS (Basic Input Output System) – базовая система ввода-вывода. BIOS представляет собой «промежуточный слой » между программной и аппаратной частями системы. Для начального запуска компьютера необходимо провести тестирование и загрузить основные драйверы устройств. Bios- это набор программ небольшого размера в функции которых входит:

- Тестирование компьютера при включении питания (POST- Power On Self Test) с помощью специальных тестовых программ.
- Поиск и подключение к системе других BIOS, расположенных на платах расширения.
- Распределение ресурсов между компонентами компьютера.

После завершения начальной стадии загрузки BIOS загружает операционную систему и передаёт ей управление. После загрузки операционной системы BIOS может продолжать различные действия на низком уровне по запросам поступающим от операционной системы и приложений. Активно использовали BIOS операционные системы DOS , windows 3.11. Современные операционные системы мало зависят от сервисов BIOS.

Аппаратное обеспечение BIOS

BIOS в ПК обычно можно найти в следующих компонентах системы:

- ПЗУ системной платы
- ПЗУ плат расширения (например видеоадаптер)
- данные на диске загружаемые в ОЗУ

Системная BIOS содержит драйверы основных компонентов- клавиатуры, дисковод, жесткого диска, последовательного и параллельного портов необходимых для начального запуска компьютера

Для отображения информации на экране монитора требуется активизировать видеоадаптер, но его поддержка не встроена в системную BIOS. Существует огромное количество видеоадаптеров и все их драйверы невозможно поместить в системную BIOS. В таких случаях драйверы помещаются в микросхему BIOS на плате этого устройства, а системная BIOS при загрузке ищет BIOS видеоадаптера и загружает его до запуска операционной системы. Компания IBM предложила следующее решение : ПЗУ системной платы сканирует разъёмы на предмет наличия плат с собственным ПЗУ. Если такой адаптер найден, код ПЗУ выполняется на этапе первоначальной загрузки, до загрузки операционной системы.

BIOS и CMOS RAM

Иногда пользователи путают эти понятия, но фактически это разные компоненты.

Обычно BIOS находится в отдельной микросхеме ПЗУ на материнской плате (Рис 1).



Рис 1 Микросхема BIOS

Кроме того, на материнской плате находится микросхема RTC/ NVRAM , которая содержит часы истинного времени и энергонезависимую память. Как правило она называется CMOS (Complimentary Metall Oxide Semiconductor) микросхемой. Она потребляет при работе ничтожно мало энергии. (ток потребления приблизительно 1 мка). Для питания используется небольшой аккумулятор напряжением 3 вольта (Рис 2).



Рис 2 Аккумулятор CMOS

Аккумулятор работает в режиме заряд-разряд: при включенном компьютере аккумулятор заряжается, а при выключенном микросхема питается от аккумулятора.

При загрузке BIOS Setup и последующем конфигурировании установочные параметры системы записываются в соответствующую область памяти микросхемы. При каждой загрузке системы для определения её конфигурации проводится считывание параметров хранящихся в микросхеме.

Процессы, происходящие после включения компьютера

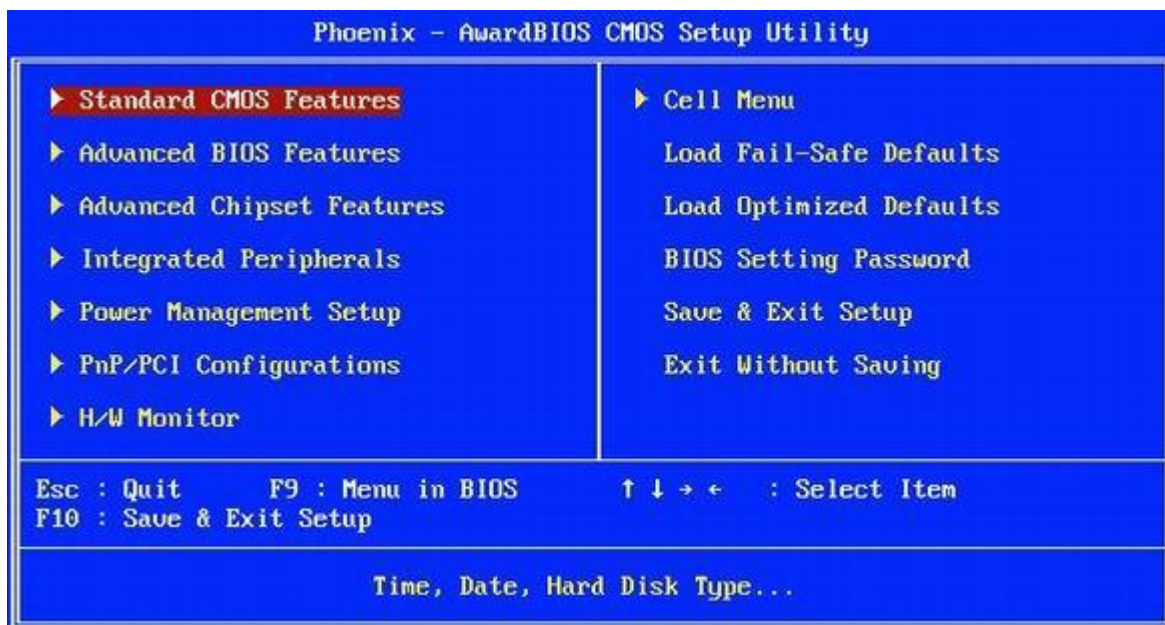
1. Спустя 0.5 – 1 сек после включения вырабатывается сигнал Power Good и на вход процессора поступает сигнал «Reset».
2. Процессор обнуляется и приступает к работе. Выполняется обращение к BIOS
3. Тестирование процессора.
4. Проверка контрольной суммы ROM BIOS.
5. Проверка и инициализация контроллеров DMA, IRQ и таймера 8254.
После этой стадии становится доступной звуковая диагностика.
6. Проверка операций регенерации памяти.
7. Тестирование первых 64 кБайт памяти. (В эту часть памяти будет загружаться BIOS)
8. Загрузка векторов прерываний.
9. Инициализация видеоконтроллера.
После этого этапа диагностические сообщения выводятся на экран.
10. Тестирование полного объема ОЗУ.
11. Тестирование клавиатуры.
12. Тестирование CMOS памяти.
13. Инициализация COM и LPT портов.
14. Инициализация и тест контроллера FDD.
15. Инициализация и тест контроллера HDD.
16. Поиск дополнительных модулей ROM BIOS и их инициализация.
17. Вызов загрузчика операционной системы (INT 19h, Bootstrap), при неудаче - остановка системы (HALT).

- Загрузчик операционной системы – подпрограмма выполняющая поиск действующего основного загрузочного сектора на дисковых устройствах. При обнаружении сектора соответствующего определенному критерию выполняется код начальной загрузки. Загружается первый файл инициализации операционной системы (NTLDR для WIN XP) отвечающий за управление этапом её загрузки.

- При запуске системы в режиме защищенном от сбоев используются драйверы BIOS.

Изменение настройки BIOS CMOS Setup

Для вызова меню CMOS Setup необходимо во время загрузки компьютера нажать клавишу “ Del” (Некоторые производители используют другую комбинацию клавиш). На экране появляется меню.



Настройка BIOS: с помощью курсора выберите нужную строку и нажмите "Enter".

Чтобы выбрать какое-либо меню в BIOS, используйте курсор и с помощью стрелок подведите его к нужному пункту. Нажав клавишу "Enter", вы перейдете к разделу или получите окно выбора настройки (как на иллюстрации ниже). Чтобы изменять указанную настройку, следует нажимать клавиши "плюс" [+] или "минус" [-], либо другую комбинацию вроде [Page Up] и [Page down]. Из главного меню настройки BIOS вы будете попадать на различные разделы настройки, которые могут разделяться и на собственные подразделы.

Задания:

1. На материнской плате найдите микросхему BIOS. Обычно она имеет наклейку с логотипом фирмы-производителя. Определите тип микросхемы (EPROM или EEPROM), запишите маркировку и определите по справочнику её характеристики.
2. Найдите на видеокarte BIOS видеоадаптера. Определите тип и параметры микросхемы.
3. Включите компьютер. Во время загрузки нажмите на клавишу «Del» и войдите в меню Setup.
4. Найдите в меню и измените параметры «Дата», «Время», «порядок загрузки». Определите параметры жесткого диска с помощью программы «Autodetect HDD»
5. Найдите в меню пункты относящиеся к защите информации и запишите их в отчет
6. Найдите в меню Setup пункты относящиеся к включению и отключению портов LPT и COM. Отключите LPT порт.
7. Найдите в меню пункты определяющие защиту загрузочного сектора от вирусов и запишите их
8. Измените порядок загрузки компьютера. Сделайте первой загрузку с CD привода
9. Оформите отчет о проделанной работе

Лабораторная работа №4

Тема: Определение оптимальной конфигурации оборудования и характеристик устройств для конкретной задачи. Поиск моделей комплектующих, их цен и поставщиков в Интернете.

Цель: Подобрать оптимальную конфигурацию персонального компьютера под конкретную задачу

1. Профессиональный видеомонтаж
2. Игровой компьютер
3. Обработка текстовой информации (Word, Excel)
4. Обработка звуковой информации (обратить внимание на звуковые карты и ПО)
5. Серфинг в интернете
6. Для построения чертежей и графиков

Предложить для каждого типа несколько различных конфигураций.(учитывать все характеристики ОЗУ, HDD, Видеокарта и т.д.) Сокеты, частоты и т.д.

Предложить ПО как бесплатное так и платное.

Предложить оптимальную ОС и ее разрядность для того или иного вида деятельности.

Лабораторная работа №5

Теоретическая часть

Архитектура компьютера

Архитектура компьютера (АК) - абстрактное представление компьютера, которое отражает его схемотехническую, логическую и функциональную организацию. В понятие архитектуры компьютера входят:

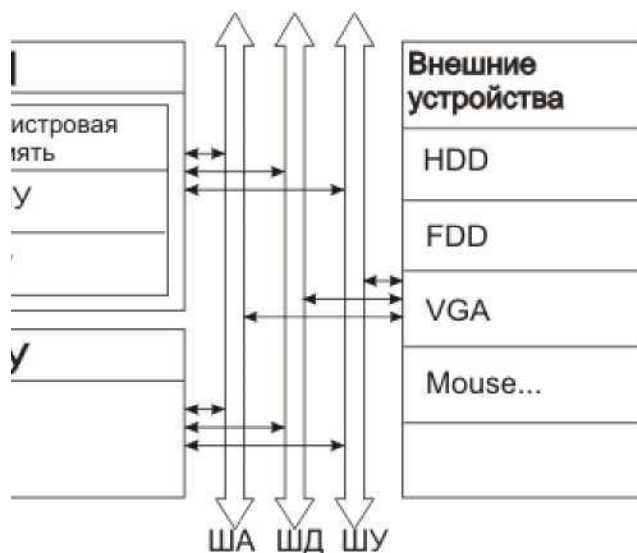
Структурная схема

Организация и разрядность интерфейсов

Набор и доступность регистров

Организация памяти

Структурная схема



где МП - микропроцессор, АЛУ - арифметико-логическое устройство, УУ - устройство управления, ОЗУ - оперативное запоминающее устройство, ША - шина адреса, ШД - шина данных, ШУ - шина управления.

Организация и разрядность интерфейсов

При рассмотрении общей схемы ЭВМ под интерфейсами подразумевают шины обмена информацией (на схеме ША - шина адреса, ШД - шина данных, ШУ - шина управления). Посредством этих шин организован интерфейс (связь, взаимодействие) между МП и другими частями компьютера.

Для ПК на базе процессора i8086 эти шины имеют следующие разрядности:

шина адреса - 20;

шина данных - 8;

шина управления - 1.

Набор и доступность регистров

Регистр — 16-ти-разрядная ячейка памяти, расположенная непосредственно в МП. Доступ к таким ячейкам осуществляется

на много

ah al si

быстрее, чем к ячейкам памяти в ОЗУ МП содержит большое количество регистров, и каждый из них имеет собственное название. Не все регистры доступны для практического использования. Ниже приведена классификация по функциональному назначению доступных регистров.

bh bl di

ch cl ip

dh dl bp

Особенностью данной группы регистров является то, что они могут использоваться при различных арифметико-логических операциях (регистры арифметико-логического устройства).

Для регистров *ax*, *bx*, *cx*, *dx* можно получить доступ отдельно к старшему или младшему байту слова. Старший байт обозначается *h* (*high*), младший - *l* (*low*). Так регистр *ax* может использоваться и целиком, и по частям: *ah* - старший байт регистра *ax*, *al* - младший байт регистра *ax* (см. Рис 2).

Регистры *si* и *di* называют индексными регистрами, их можно использовать по своему усмотрению, но официально они предназначены для поддержки цепочечных команд.

Регистры *sp* и *bp* лучше не использовать без особой необходимости на начальном этапе изучения ассемблера.

Данные регистры используются при работе со специальной областью памяти - стеком, неправильное их использование может внести в программу труднообнаруживаемые ошибки.

Сегментные регистры
Название "сегментные" данные регистры получили от применяемого в МП Intel способа адресации памяти. Этот способ состоит в поблочном использовании памяти, блоки называются сегментами. Каждая программа может одновременно использовать 4 сегмента (блока) памяти:

сегмент с кодом программы;

сегмент с данными;

сегмент стека;

дополнительный сегмент данных.

Для задания местоположения этих сегментов в памяти (для адресации) и применяются сегментные регистры.

В данную группу входят 4 регистра:

cs (code segment) - сегмент кода;

ds (data segment) - сегмент данных;

ss (stack segment) - сегмент стека;

es (extension data segment) - дополнительный сегмент данных.

Регистры состояния и управления

Регистры данной группы предоставляют доступ к некоторой дополнительной информации о состоянии МП и о результате выполнения последней арифметико-логической операции.

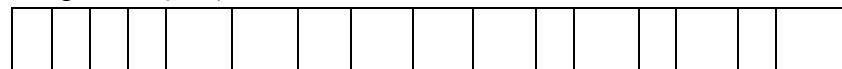
Регистр указатель (*ip*)

В регистре *ip* содержится адрес команды, которая будет выполняться следующей. Фактически этот адрес содержится в паре регистров *cs:ip*.

содержит адрес сегмента кода, а *ip* - смещение (положение относительно начала сегмента) нужной команды (см **организация памяти**).

Флаговый регистр (*flags*)

Флаговый регистр особо выделяется среди всех остальных. У этого регистра каждый бит (для флагового регистра биты именуется *флагами*) имеет определённое назначение и собственное название. Флаги отражают состояние микропроцессора и особенности результата выполнения последней арифметической или логической операции. Флаг либо установлен, либо сброшен (0, 1).



и состояния:

CF (Carry) - флаг переноса. Используется арифметическими командами и командами сдвигов;

PF (Parity) - флаг контроля четности. Содержит 1, если сумма единиц в восьми младших разрядах регистра является четным числом, или 0, если она нечетна;

AF (Auxiliary) - дополнительный флаг переноса. Содержит значение переноса из 3-го бита восьмибитовых данных;

ZF (Zero) - флаг нуля. Содержит 1, если результат арифметической операции или операции сравнения нулевой, в противном случае содержит 0;

SF (Sign) - знаковый флаг. Содержит знак результата (старший бит) после выполнения арифметических операций (0 = '+', 1 = '-');

OF (Overflow) - флаг переполнения. Указывает, что в результате выполнения арифметической команды возникло переполнение старшего разряда результата;

управления:

• DF (Direction) - флаг направления. Определяет, в каком направлении будут обрабатываться данные цепочечными командами - от младших адресов к старшим или наоборот;

Системные флаги:

IF (Interrupt) - флаг разрешения прерываний. Указывает на возможность внешних прерываний;

а TF (Trap) - флаг пошагового выполнения. Устанавливает выполнение команд процессора в пошаговом режиме (для отладки программ).

Организация памяти

Физическая память, к которой МП имеет доступ, называется оперативной памятью (ОП) или оперативным запоминающим устройством (ОЗУ). Память можно условно представить в виде простой последовательности пронумерованных байт. Порядковый номер байта называется его физическим адресом.

Доступ к любому байту ОЗУ осуществляется по шине адреса, которая, как мы уже говорили, является 20-разрядной (для i8086). Разрядность шины ограничивает объем доступной памяти, так 20 разрядов шины позволяют записать адреса от 0 до $2^{20}-1$ (1 Мбайт). Все адреса удобнее записывать в шестнадцатеричном виде (диапазон физических адресов от 00000h до FFFFFh).

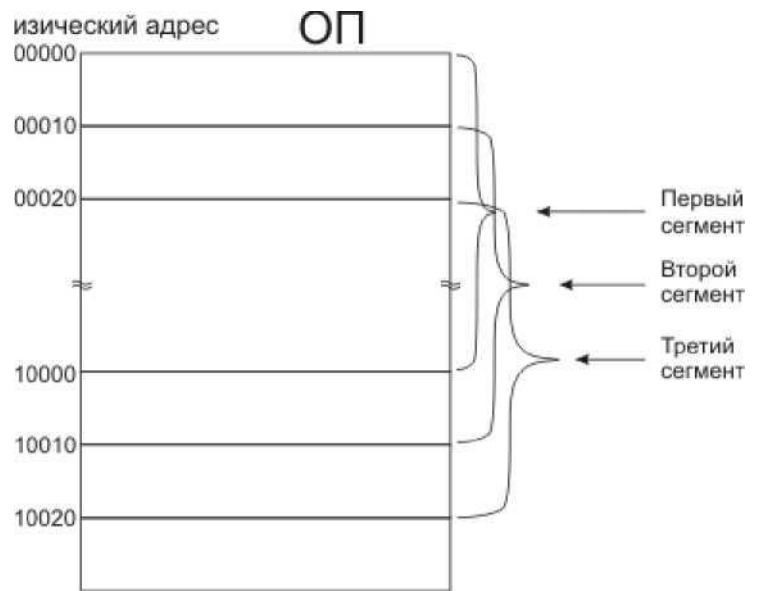
Для записи любого физического адреса необходимо 20 разрядов. Но все регистры МП 16-разрядные. Для получения 20-разрядного адреса используется следующий механизм:

1. Всю физическую память условно можно разбить на параграфы (1 параграф = 16 байт);

2. Память будем выделять программам только небольшими блоками, данные блоки будем в дальнейшем называть сегментами памяти или просто сегментами.

3. Программа может состоять из любого количества сегментов;

Каждый сегмент может начинаться только с начала параграфа, т.е. адресом сегмента будет номер его первого параграфа (такой адрес уже 16-разрядный);



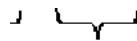
Такой механизм называется сегментацией, он поддерживается на аппаратном уровне (мы не можем его заменить или изменить). В МП имеются специальные **сегментные регистры**, которые содержат адреса сегментов в памяти.

Так как под адресом сегмента понимается номер параграфа, с которого он начинается, то для получения физического адреса начала сегмента необходимо умножить его на 16. Например, сегмент 0032h имеет физический адрес $0032h \times 10h = 00320h$.

Для определения адреса конкретной ячейки памяти недостаточно знать адрес сегмента, в котором она находится. Необходимо знать позицию этой ячейки относительно начала сегмента. Этот относительный адрес называют эффективным адресом или смещением. Поскольку максимальный размер сегмента ограничен 64 килобайтами, то смещение будет записываться 16-разрядным значением.

Таким образом, от 20-разрядного физического адреса мы перешли к адресу, состоящему из двух 16-разрядных частей: адрес сегмента и смещение. Такие составные адреса называются логическими и записываются так:

96AF : E305



сегмент смещение

Механизм сегментации служит не только для перехода от 20-разрядных физических адресов к 16-разрядным логическим, он необходим для защиты программ от взаимного влияния.

Таблица кодов символов ASCII

ASCII - сокращение от American Standard Code for Information Interchange (Американский стандартный код для обмена информацией). ASCII каждому символу и некоторым операциям соответствует свой числовой код.

Расширенная таблица ASCII состоит из двух частей. Первая (символы с кодами 0-127) является

ол		ол		ол		ол		ол	

универсальной (см. табл.), а в' предназначена для специальных символов и букв национальных алфавитов и на компьютерах разных типов может меняться.

Символы с кодами 0-31 относятся к служебным. Они представляют собой команды, вывод которых на стандартное выходное устройство выполняется определёнными действиями.

Функции DOS (Int 21h)

Функции DOS и BIOS

Назначение	№ Функции	Данные на входе	Данные на выход
Ввод символа с ожиданием и эхосопровождением	01	—	al - ASCII код
Вывод символа	02	dl - ASCII ко	—
Ввод символа с ожиданием и без эхосопровождения	08	—	al - ASCII код
Вывод строки на экран	09	ds:dx - адрес строки с символом '\$' на конце	—
Ввод строки с клавиатуры	0ah	ds:dx - адрес буфера со следующим форматом: <ul style="list-style-type: none"> • 1й байт - размер буфера; • 2й байт - число фактически введенных символов; • 3й байт и следующие за ним - буфер под 	Введенная строка в буфере с 0dh на конце

Функции BIOS (Int 10h)

Назначение	№ Функции	Данные на входе	Данные на выходе
Установить положение курсора	02	dh - строка dl - колонка	—
Прочитать символ и атрибут с экрана	08	bh - № экранной страницы	атрибут al - символ
Вставить символ с атрибутом	09	bh - № экранной страницы; bl - атрибут; al - ASCII код;	—
		cx - число повторений	
Получить параметры видеорежима	0fh	—	al - видеорежим bh - № экранной страницы
Установить параметры видеорежима	00h	al - видеорежим bh - № экранной страницы	—

Создание

программы

на

языке ассемблера

В отличие от языков программирования высокого уровня, язык ассемблера поставляется без среды разработки программ. Поэтому разработчику приходится самостоятельно контролировать весь процесс создания программы. Для языка ассемблера этот процесс включает в себя следующие этапы:

Постановка задачи (точное и подробное описание функциональности будущей программы, а также описание

всех входных и выходных данных и способа их передачи программе);

Разработка алгоритма программы (построение блок-схемы, граф-схемы или текстовое или математическое описание решения);

Формализация алгоритма (запись алгоритма на языке программирования).

Создание текстового файла программы с расширением .asm (например my.asm). Отсутствие среды разработки позволяет программисту самостоятельно выбрать текстовый редактор для написания кода программы. Для этой цели подойдет любой текстовый редактор с нумерацией строк, мы рекомендуем редактор, встроенный в оболочку "FAR Manager".

Компиляция программы;

Компиляция - процесс перевода программы из текстового вида в машинный код. При использовании компилятора фирмы Borland необходимо выполнить: `tasm my.asm`

т.е. запускаем компилятор `tasm` и передаем с командной строки имя файла, содержащего программу. Если программа имеет синтаксические ошибки, компилятор выдаст сообщение об ошибке с указанием номера строки

и описанием для каждой ошибки (нужно вернуться на этап №3 и исправить синтаксические ошибки). В случае успешной компиляции будет создан файл, содержащий объектный код программы my.obj, который ещё не является исполняемым модулем.

Компоновка программы; Компоновка - создание из файла объектного кода исполняемого модуля.

tlink my.obj

В качестве параметра компоновщик tlink принимает имя файла содержащего объектный код программы (в нашем случае - my.obj). В случае успешной компоновки будет создан исполняемый модуль my.exe

Запуск и тестирование исполняемого модуля программы. На данном этапе необходимо проверить, соответствует ли написанная программа постановке задачи, сделанной нами на этапе №1. Неправильная работа программы говорит об алгоритмической ошибке (семантическая ошибка), поэтому для успешного её устранения нужно вернуться на этап разработки алгоритма (этап №2).

Написание первой программы на языке ассемблера

Постановка задачи. Написать программу, которая выводит на экран строчку "Привет!".

Разработка алгоритма программы. Алгоритм линейный, разработки не требует.

Формализация (запись) алгоритма

В текстовом редакторе создаем файл privet.asm и записываем в него следующий код (без номеров строк) :

```
;описание  
data segment
```

```
сегмента данных ;строка для
```

```
mes db вывода на экран. 'Привет!$$' - признак
```

```
конца строки
```

```

3  data ends
4
5  de segment ;конец сегмента данных
;начало сегмента кода
;метка start - начало нашей программы
6  start: ;директива компилятора
;настройка сегмента данных
assume ;функция №9 - 9 вывод строки на экран
;берём адрес
7  cs:code, строки
ds: ;вызов
;прерывания для вывода строки
8  mov ax, ;функция
data
9  mov ds, . завершения
10 ;программы
;завершаем
11 mov ah, программу
; конец сегмента
12 lea dx, mes ;конец
;программы с точкой входа start
13 int 21h 4. Компиляция программы
tasm privet.asm
14 5. Компоновка программы tLink privet.obj
6. Запуск и тестирование
15 mov ax, 4c00h ;программы privetasm
16 int 21h ;Строки 1 - 3 программы privet.asm содержат описание
;сегмента данных. Сегмент данных - область память, в
;которой будут храниться данные для наших программ.
17 code ends ;Строки 5 - 17 — это код программы, её исполняемая часть.
18 end start ;В 8 и 9 строках выполняется настройка сегмента данных
;программы.
;Строки 11 - 13 — вывод строки на экран при помощи
;функции №9 прерывания 21h (подробнее о функциях и
;работе с ними на следующей лабораторной работе).
;15 и 16 строки — стандартное завершение программы.

```

После символа ';' пишутся комментарии, они не обрабатываются компилятором.

Переход на новую строку

Для организации перехода на новую строку достаточно вывести на экран символы перевода строки и возврата каретки (CR/LF). Эти символы имеют коды 10 и 13. Если в нашей программе необходимо после вывода строки перейти на новую, то для этого достаточно переписать вторую строку программы:

```
mes2 db ' Выводим строку и переходим на новую...', 10, 13, '$'
```

Переход на новую строку можно выполнить и до вывода сообщения на экран:

```
mes3 db 10, 13, 'Выводим с новой строки...$'
```

Задание для выполнения

Написать программу, которая выводит одно под другим следующие сообщения:

Привет!

Меня зовут компьютер!

До свидания!

Применение

функций

DOS и BIOS

В лабораторной работе приведена

программа вывода строки на экран. Вывод строки в этой программе осуществляется посредством обращения к функции операционной системы (ОС). Кроме этой функции ОС содержит большое количество других функций, которые можно использовать при разработке программ. Для вывода строки на экран использовалась функция № 9 прерывания № 21h. Прерывание - специальный набор готовых процедур, постоянно доступный для использования.

Общий алгоритм применения функций любого прерывания

Поместить номер вызываемой

функции в регистр ah;

Подготовить, если необходимо, входные данные для функции;

Вызвать прерывание;

Сохранить или обработать выходные данные

Пример 1: Вывести на экран символ '\$'

```
mov ah, 02 ;номер функции mov dl, '$' ;входные данные int 21h ;вызов прерывания
```

Пример 2: Очистить экран

```
mov ah, 0fh ; получить параметры видеорежима int 10h
```

```
mov ah, 00 ; установить параметры
```

```
видеорежима
```

```
int 10h
```

Задания для выполнения

В центре чистого экрана вывести слово "Привет"; (5 баллов);

Вывести по углам экрана (с отступом по 3 символа от углов) символы от '1' до '4'; (5 баллов);

Задания для самостоятельного выполнения

В центре чистого экрана вывести слово "Привет" *вертикально*; (5 баллов);

Вывести приглашение к вводу символа, ввести символ, вывести введенный символ в центре очищенного экрана. (8 баллов)

Архитектура компьютера

Лабораторная работа №3

Линейные алгоритмы

Вопросы для повторения:

1. Какие алгоритмы называются линейными?
2. Какие действия обычно выполняются на линейных участках алгоритмов?
3. Что такое программа-отладчик? Назовите примеры таких программ.
4. Для чего необходимо кодирование информации? Приведите примеры кодирования информации, которые вы применяете.
5. Что называется ASCII-кодами?

Линейными называются такие алгоритмы, в которых все действия выполняются поочередно, друг за другом.

На линейных участках алгоритмов обычно выполняются манипуляции данными, вычисление выражений, перекодировка информации.

Команда пересылки данных:

mov <приемник>, <источник>

команда **mov** - команда пересылки данных, результатом ее исполнения является копирование данных из операнда <источник> в операнд <приемник>.

mov ah, 09: записать в регистр ah число 09

mov cl, ah: скопировать байт из регистра ah в ch

mov bx, cx; скопировать слово из регистра cx в регистр bx

mov ch, 0fh

mov ch, bl

Вопрос: чему будет равно содержимое регистра ch после выполнения данного участка программы?

Простейшие арифметические команды:

<слагаемое 1>, <слагаемое 2>	д ать в операнд <слагаемое 1> значение суммы обоих
операнд >	д читать <операнд > на единицу;
<операнд 1>, <операнд 2>	д ать в операнд <операнд 1> значение разности
операнд >	д вычитать <операнд> на единицу.
множитель	д ожает <сомножитель 1> на <сомножитель 2> если <сомн2> - байт, то <сомн1> - регистр al (результат слово в ax); <сомн2> - слово, то
делитель>	д т <делимое> на <делитель> если <делитель> - байт, то <делимое> - регистр ax (результат частное в al , остаток в ah); если <делитель> -

Отладчик Turbo Debugger

Отладчик Turbo Debugger (TD) - оконная среда отладки программ, позволяет производить отладку программ, написанных на языках Pascal, C, assembler.

TD позволяет определить:

- место логической ошибки (семантической);
- причину логической ошибки.

Для использования TD необходимо выполнить следующие действия:

tasm /zi *.asm откомпилировать с добавлением отладочной информации
tlink / v *. obj ^компоновать с добавлением отладочной информации td *. exe

Файл с исходным кодом программы должен находиться в директории отладчика.

Задания для выполнения

1. Вывести на экран сумму двух введенных с клавиатуры цифр (значение вводить так, чтобы сумма была меньше 10); (10 баллов).
2. Изучите работу отладчика TD на примере задачи из первого задания.

Необходимо изучить:

- a) . Систему меню программы TD.
- b) . Выполнение программы в четырех режимах
м
ловного
тремя:
ле по шагам;
кущего
кения
та:
установкой точек
прерывания.
тему
вспомогательных окон
).

Задания для самостоятельного выполнения

1. С клавиатуры вводятся два символа. Вывести их в обратном порядке (Например: '1gg1', 'assa', '=='); (5 баллов);
2. Имеется два числа (цифры) a и b. Вычислить $y = 2a - b + 1$; (8 баллов);
3. Вывести на экран сумму двух любых введенных с клавиатуры цифр (сумма может быть больше 10) (12 баллов);

Десятичная арифметика

Вопросы для повторения:

1. Что такое позиционные системы счисления?
2. Что называется двоичной (десятичной) арифметикой?
3. Каково назначение BCD-кода?
4. Какие команды работают с BCD-числами?
5. Как работают команды daa и das?

Команды десятичной арифметики

Рассматриваемые ниже команды предназначены для работы с неупакованными BCD-числами. Изучите их работу по приведенным в таблице примерам.

Описание	Алгоритм работы	Ассемблер
Процедура вычитания двух упакованных BCD- чисел	<p>проанализировать значение младшего полубайта регистра al и значение флага</p> <p>если (значение младшего полубайта регистра al >9) или (AF=1), то выполнить следующие действия:</p> <p>прибавить значение al на 6; очистить старший полубайт регистра al; прибавить значение ah на 1; установить флаги: af = 1, иначе сбросить флаги af = 0 и cf = 0.</p>	<pre> al, 7 mov ch, 6 add al, ch aaa ; al=3 </pre>
Процедура вычитания двух упакованных BCD- чисел	(младший полубайт регистра al меньше 9) или	<pre> al, 5 mov ch, 6 </pre>

	<p>разрядных чисел BCD-</p>	<p>af=1), то выполнить следующие действия: уменьшить значение младшего полубайта регистра al на 6; обнулить значение старшего полубайта регистра al; установить флаги af и cf в 1; установить флаги af и cf в 1</p>	<p>1=9</p>
	<p>преобразование упакованных чисел; преобразование двузначного числа (≤ 99) из двоичного вида в его упакованный BCD-эквивалент</p>	<p>делить значение регистра al на 10; записать частное в регистр ah, остаток — в регистр al.</p>	<p>1, 7 ah, 6 ah</p>
	<p>подготовка упакованных чисел для операции; преобразование двузначного упакованного BCD-числа в двоичное представление</p>	<p>умножить значение регистра ah на 10; сложить полученное значение с значением младшим регистром al: $(ah*10)+al$; присвоить регистру al значение $(ah*10)+al$; обнулить регистр ah</p>	<p>ah, 4 1, 2 2 1, 6</p>

Задание:

Вывести на экран сумму двух введенных с клавиатуры двузначных чисел (Вводить посимвольно функцией 01 прерывания 21h, значение вводить так, чтобы сумма была

меньше 100). Для решения задачи применить команды десятичной арифметики. Проследить за выполнением операций с использованием отладчика. (10 баллов)

Задания для самостоятельного выполнения

1. Вывести на экран байт в десятичном виде (10 баллов);

Лабораторная работа №6

Тема: Изучение оперативной памяти компьютера

Цель: изучить устройство и принцип работы оперативной памяти. Выполнить тестирование оперативной памяти

Перечень необходимых средств обучения: модули оперативной памяти, материнские платы, компьютер, программа MemTest 86+

Краткие теоретические сведения

Память компьютера – это устройства для записи, хранения и чтения информации.

Исторически сложилось, что памятью называют оперативную память компьютера, работающую на микросхемах.

Устройства: дискеты, жесткие диски и т.п. называют накопителями.

Структура памяти компьютера приведена на рис 1

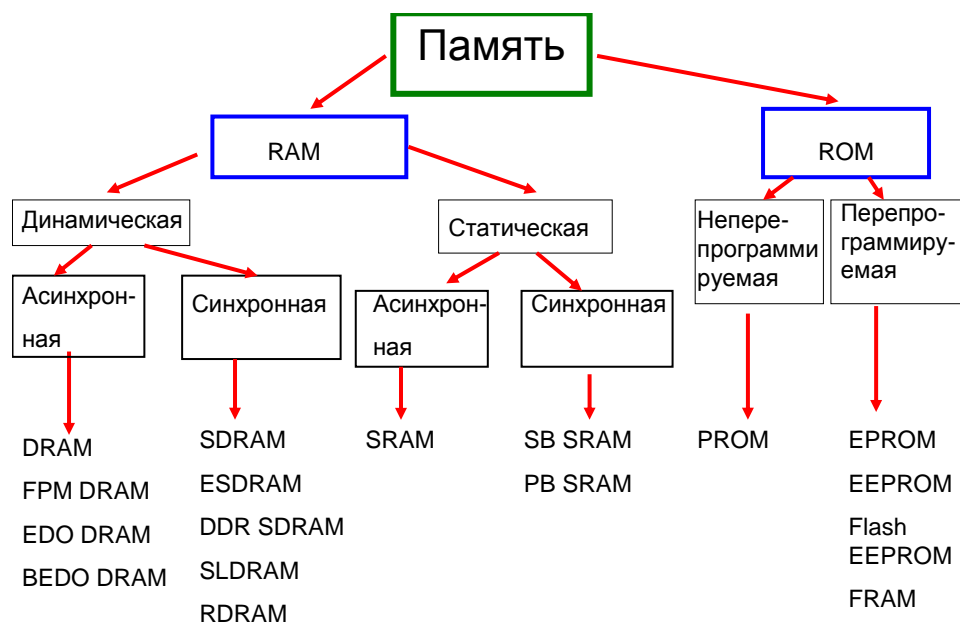


Рис 1 Структура памяти компьютера

Оперативная память (RAM)- это рабочая область для процессора компьютера. Во время работы в ней хранятся программы и данные.

Информация в оперативной памяти сохраняется только при включенном питании или до нажатия кнопки «Reset».

ROM- (Read only Memory)- энергонезависимая память только для чтения. Предназначена для долговременного хранения информации. Физически представляет из себя микросхемы ПЗУ.

RAM- (Random Access Memory)- память с произвольным доступом. Это означает, что обращение к данным не зависит от их расположения в памяти

Пространство памяти представлено в виде ячеек (прямоугольники), которые состоят из определённого количества строк и столбцов. Один такой "прямоугольник" называется страницей, а совокупность страниц называется банком.

Принцип работы модулей памяти

Микросхемы памяти собирают на отдельных платах –модулях. В зависимости от типа памяти, частоты и др. параметров модули имеют различный тип разъема для подключения (чтобы избежать ошибочного включения).

Виды модулей памяти

SIMM (Single In line Memory Module)- модули с однорядным расположением выводов- устаревшая модель. Использовались 30- pin и 72 pin модули. Контакты с обеих сторон соединены попарно. Это значительно повышает надежность работы



Рис 2 Модуль памяти SIMM

DIMM (Dual In line Memory Module) . В связи с возрастанием объёма памяти и , следовательно , увеличением количества контактов производители памяти отказались от дублирования контактов. Контакты с двух сторон модуля и электрически независимы. Используются 168 и 184 контактные и модули.

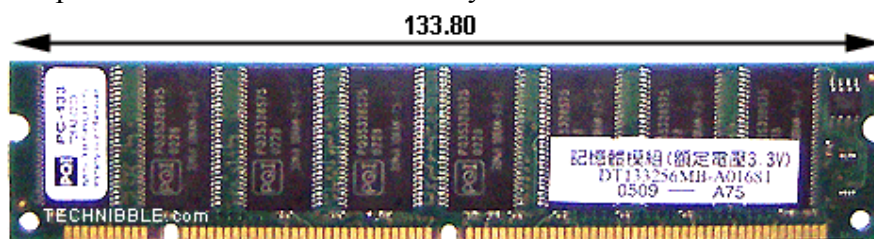


Рис 3 Модуль памяти DIMM

RIMM (Rambus In line Memory Module)- модуль оперативной памяти, разработанный компанией Rambus совместно с Intel.

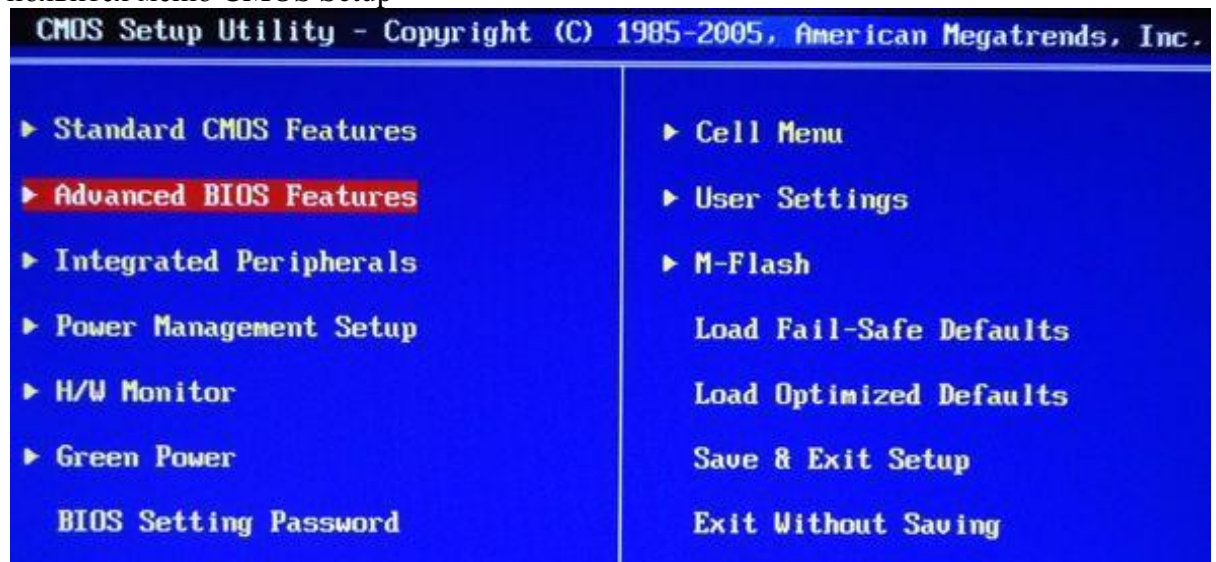
Конструктивно состоит из двух модулей (184 контакта), которые должны применяться вместе. В связи с высокой стоимостью, выпуск памяти подобного типа прекращен.



Рис 4 Модули памяти RIMM

Выполнение работы

1. Определите характеристики предложенных модулей памяти с помощью справочной таблицы и фотографий в методичке.
2. Включите компьютер. В момент загрузки нажмите кнопку Del и на экране монитора появится меню CMOS Setup



3. Выберите Advanced BIOS Features. И найдите опцию порядок загрузки (Boot Sequence)



Выберите в качестве первого загрузочного устройства дисковод А. Это означает, что загрузка системных файлов и программы будет выполняться с дискеты 3.5 дюйма.

4. Вставьте в дисковод загрузочную дискету с программой тестирования памяти
5. Сохраните сделанные изменения в Setup (выходите через опцию Save and Exit Setup)
6. Перезагрузите компьютер. На экране появится меню программы Mem Test86, как показано на скриншоте ниже. Порядок выполнения тестов, процент выполнения выделены на скриншоте красной линией

```

Memtest86+ v4.20
AMD K10 (65nm) @ 4013 MHz
L1 Cache: 16K 71668 MB/s
L2 Cache: 2048K 35834 MB/s
L3 Cache: 8192K 9488 MB/s
Memory : 512M 6330 MB/s
Chipset : AMD K10 IMC (ECC : Detect / Correct - Chipkill : On)
Settings: RAM : 887 MHz (DDR1775) / CAS : 19-12-12-30 / DDR3 (128 bits)

Pass 4% #
Test 48% #####
Test #3 [Moving inversions, 8 bit pattern]
Testing: 184K - 512M 512M
Pattern: efefefef
-----
WallTime  Cached  RsvdMem  MemMap  Cache  ECC  Test  Pass  Errors  ECC Errs
-----
0:00:07  512M      OK      e820    on  off  Std    0    0
-----

(ESC)Reboot (c)configuration (SP)scroll_lock (CR)scroll_unlock

```

Программа тестирует оперативную память циклически. Один полный проход = 9 тестам. В таблице будет информация Pass 1 и указаны выявленные ошибки (выделено зеленой линией).

```

Pass100% #####
Test 42% #####
Test #8 [Modulo 20, Random pattern]
Testing: 184K - 512M 512M
Pattern: 817df558-11
-----
Detect / Correct - Chipkill : On)
75) / CAS : 19-12-12-30 / DDR3 (128 bits)

MemMap  Cache  ECC  Test  Pass  Errors  ECC Errs
-----
e820    on  off  Std    1    0
-----

```

- Содержание этих 9 тестов приведено ниже
- **Test 0 [Address test, walking ones, no cache]**, - тест для определения проблем с адресацией памяти;
 - **Test 1 [Address test, own address]**, - более углубленный тест для определения проблем с адресацией памяти;
 - **Test 2 [Moving inversions, ones&zeros]**, - быстрая проверка на аппаратные или трудноуловимые ошибки;
 - **Test 3 [Moving inversions, 8 bit pat]**, - тоже самое, только используется 8 битный алгоритм прохода нулей и единиц. Использует 20 схем для теста;
 - **Test 4 [Moving inversions, random pattern]**, - этот тест особенно эффективен для выявления проблем с data sensitive. Использует 60 схем для теста;
 - **Test 5 [Block move, 64 moves]**, - тест для поиска проблем в схемах памяти;
 - **Test 6 [Moving inversions, 32 bit pat]**, - эффективен для определения data sensitive errors. Очень долгий тест;
 - **Test 7 [Random number sequence]**, - тест, проверяющий ошибки записи памяти;

- **Test 8 [Modulo 20, ones&zeros]**, - тест для определения скрытых ошибок при помощи кеша и буферизации, которые не выявили предыдущие тесты;
- **Test 9 [Bit fade test, 90 min, 2 patterns]**, - особый тест, который можно запустить вручную. Запоминает адреса в памяти, после чего засыпает на полтора часа. После этого проверяет не изменились ли биты в адресах. Требуется 3 часа для прохождения и ручного запуска через меню конфигурации (клавиша c).

Если по окончании теста выдается сообщение, как приведено на рисунке ниже, то память исправна.

```

CPU: 2560 MHz (X64 Mode) | Pass 0%
Cache: 64K 16732 MB/s | Test 19% #####
Cache: 6144K 16732 MB/s | Test #3 [Moving inversions, 1s & 0s Pa
Cache: None | Testing: 1024K - 2048M 2047M of 2048M
Memory : 2048M 3849 MB/s | Pattern: 00000000 | Time:
-----
Core#: 0 (SMP: Disabled) | Chipset: Intel i440FX
State: i Running... | RAM Type: EDO DRAM
Cores: 1 Active / 1 Total (Run: All) | Pass: 1 Errors:
-----
** Pass complete, no errors, press Esc to exit **

```

В случае неисправности памяти на экране появятся сообщения:

```

CPU: 2560 MHz (X64 Mode) | Pass 0%
L1 Cache: 64K 16203 MB/s | Test 15% #####
L2 Cache: 6144K 16733 MB/s | Test #9 [Random number sequence]
L3 Cache: None | Testing: 0K - 32M 32M of 1024M
Memory : 1024M 3849 MB/s | Pattern: de4bd78d R | Time: 0:03:12
-----
Core#: 0 (SMP: Disabled) | Chipset: Intel i440FX
State: i Running... | RAM Type: EDO DRAM
Cores: 1 Active / 1 Total (Run: All) | Pass: 0 Errors:
-----
Test Pass Failing Address Good Bad Err-Bits Count CPU
-----
9 0 00000100028 - 1.0MB 6453cad6 182f0cdf 7c7cc609 11 0
9 0 0000010002c - 1.0MB 475d6138 a05dd233 e700b30b 12 0
9 0 00000100030 - 1.0MB d69e1ead 91b43086 472a2e2b 13 0
9 0 00000100034 - 1.0MB 58f93383 ec4be2e8 b4b2d16b 14 0
9 0 00000100038 - 1.0MB 1ec2491c 81740ab9 9fb643a5 15 0
9 0 0000010003c - 1.0MB c1107b4e 6d26cd3e ac36b670 16 0
9 0 00000100040 - 1.0MB bd72d73b a36ecc60 1e1c1b5b 17 0
9 0 00000100044 - 1.0MB 94a6bd51 445a7967 d0fcc436 18 0
9 0 00000100048 - 1.0MB 0be8b264 25037100 2eebc364 19 0
9 0 0000010004c - 1.0MB 5154bec1 75b6003f 24e2befe 20 0
(ESC)exit (c)configuration (SP)scroll_lock (CR)scroll_unlock Loc

```

7. Подготовьте отчет о проделанной работе

Контрольные вопросы

1. Какие существуют виды модулей памяти?
2. Чем отличаются конструктивно SIMM и DIMM модули?
3. Чем отличаются синхронная и асинхронная память?
4. Чем отличаются статическая и динамическая память?
5. Что такое КЭШ память? Что означают индексы L1, L2 ?

6. Что означает аббревиатура SDRAM ? SRAM? DDR SDRAM ?
7. Чем отличается память DDR, DDR2 DDR3 DDR4?
8. Как работает программа Mem Test 86+

Лабораторная работа №7

Тема: Управление режимами энергопотребления персонального компьютера

Цель: Изучить: установки BIOS компьютера ,обеспечивающие режим экономии электроэнергии .

Научиться : выбирать и устанавливать оптимальные режимы работы компьютера

Перечень необходимых средств обучения:

компьютер

Краткие теоретические сведения

Компания Intel совместно с Microsoft разработали стандарт усовершенствованной системы управления питанием (Advanced Power Management- APM). Полностью реализованный стандарт APM позволяет автоматически переключать компьютер между пятью состояниями, в зависимости от текущего состояния системы. Каждое последующее состояние в приведенном ниже списке характеризуется уменьшением потребления электроэнергии.

- Full On (система включена) Система полностью включена.
- APM enabled (активизирован режим приостановки). Система работает, но неиспользуемые устройства могут быть выключены. Тактовая частота генератора может быть замедлена.
- APM Standby (резервный режим). Система не работает. Большинство устройств находятся в состоянии потребления малой мощности. Работа тактового генератора процессора может быть замедлена или остановлена.Необходимые параметры функционирования хранятся в памяти. Пользователь или операционная система могут запустить компьютер из этого состояния почти мгновенно.
- APM Suspend (режим приостановки). Система не работает. Большинство устройств пассивны. Тактовый генератор центрального процессора остановлен, а параметры функционирования хранятся на диске и при необходимости могут быть считаны в память для восстановления работы системы. Чтобы запустить систему из этого состояния требуется некоторое время.
- Off (система отключена). Система не работает. Источник питания выключен.

В различных версиях BIOS могут встретиться следующие опции:

Раздел Power Management Setup

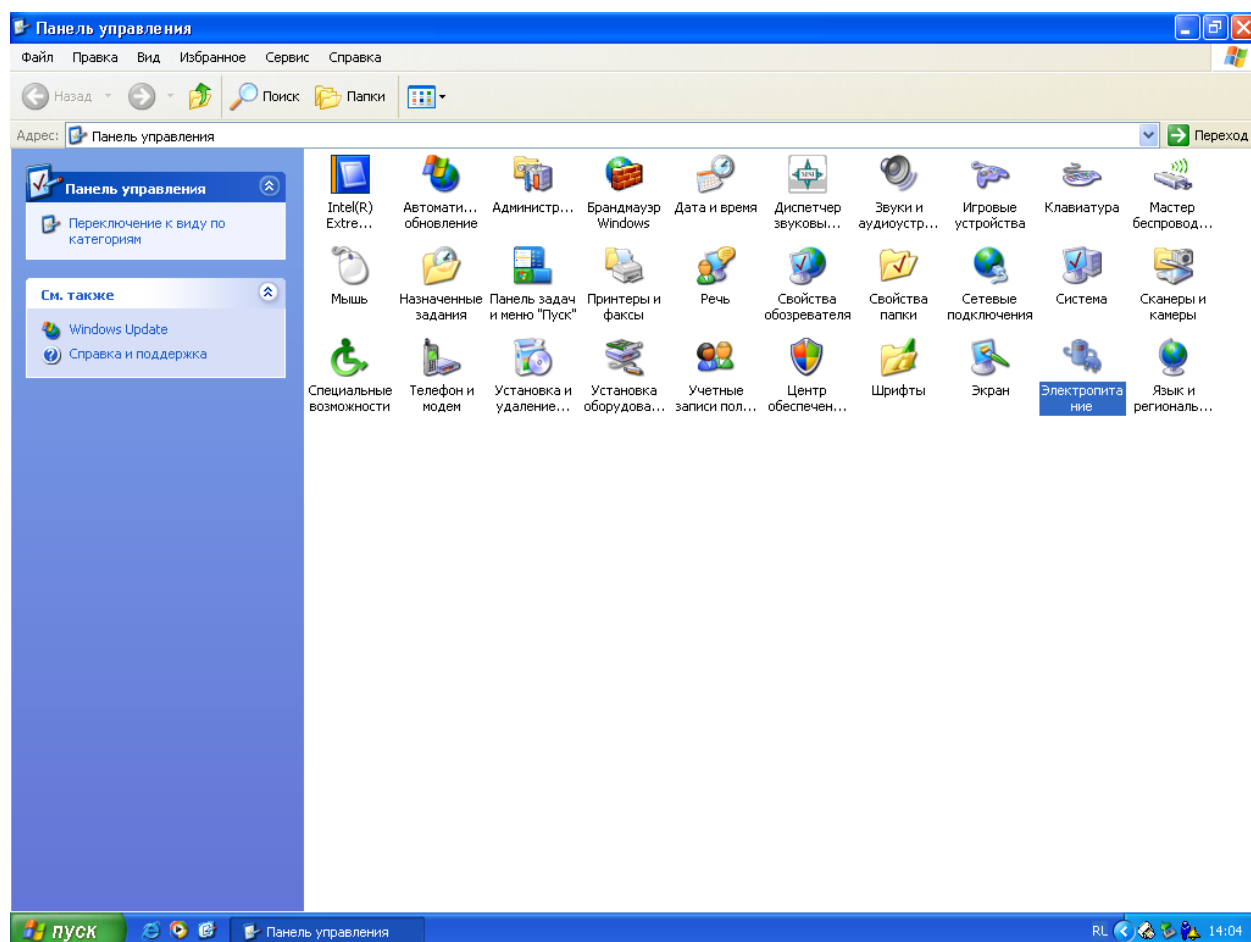
- **Power Management** (управление энергопотреблением) - позволяет либо разрешать BIOS'у снижать энергопотребление компьютера, если за ним не работают, либо запрещать. Может принимать значения:
 - **User Define** (определяется пользователем) - при установке этого параметра вы можете самостоятельно установить время перехода в режим пониженного энергопотребления.

- **Min Saving** (минимальное энергосбережение) - при выборе этого параметра компьютер будет переходить в режим пониженного энергопотребления через время от 40 мин. до 2 часов (зависит от конкретного BIOS материнской платы)
 - **Max Saving** (максимальное энергосбережение) - компьютер перейдет в режим пониженного энергопотребления через 10 - 30 с. после прекращения работы пользователя с ним.
 - **Disable** (запрещение энергосбережения) - запрещает режим энергосбережения.
 - **ACPI function** (функционирование ACPI) - разрешает или запрещает поддержку BIOS стандарта ACPI. Следует помнить, что по состоянию на конец 1998 года только Windows 98 поддерживает этот стандарт. Может принимать значения:
 - **Enabled** - разрешено
 - **Disabled** - запрещено
 - **Video Off Option** (в каком режиме выключать монитор) - позволяет устанавливать, на какой стадии "засыпания" компьютера переводить монитор в режим пониженного энергопотребления. Может принимать значения:
 - **Susp, Stby -> Off** (выключение в режиме Suspend И Standby) - монитор перейдет в режим пониженного энергопотребления при наступлении либо режима Suspend, либо Standby.
 - **All modes -> Off** (выключение во всех режимах) - монитор будет переведен в режим пониженного энергопотребления в любом режиме.
 - **Always On** (всегда включен) - монитор никогда не будет переведен в режим пониженного энергопотребления
 - **Suspend -> Off** (выключение в режиме Suspend) - монитор перейдет в режим пониженного энергопотребления при наступлении режима Suspend.
 - **Video Off Method** (способы выключения монитора) - устанавливается способ перехода монитора в режим пониженного энергопотребления. Может принимать значения:
 - **DPMS OFF** - снижение энергопотребления монитора до минимума
 - **DPMS Reduce ON** - монитор включен и может использоваться
 - **DPMS Standby** - монитор в режиме малого энергопотребления
 - **DPMS Suspend** - монитор в режиме сверхмалого энергопотребления
 - **Blank Screen** - экран пуст, но монитор потребляет полную мощность
 - **V/H SYNC+Blank** - снимаются сигналы разверток - монитор переходит в режим наименьшего энергопотребления.
 - **Suspend Switch** (переключатель режима Suspend) - параметр разрешает или запрещает переход в режим suspend (временной остановки) с помощью кнопки на системном блоке. Для этого необходимо соединить джампер SMI на материнской плате с кнопкой на лицевой панели. Как правило, для этого используется либо специальная кнопка Sleep, либо кнопка Turbo. Режим suspend является режимом максимального снижения энергопотребления компьютером. Может принимать значения:
 - **Enabled** - разрешено
 - **Disabled** - запрещено
 - **Doze Speed** (частота процессора в режиме Doze) - определяет коэффициент деления тактовой частоты в режиме Doze (засыпание).
 - **Stby Speed** (частота процессора в режиме Standby) - определяет коэффициент деления тактовой частоты в режиме Standby (ожидания работы).
- PM Timers** - в этой секции устанавливаются времена перехода в различные стадии снижения энергопотребления.
- **HDD Power Down** (выключение жесткого диска) - устанавливает либо время, через которое при отсутствии обращения жесткий диск будет выключен, либо запрещает

такое выключение вообще. Параметр не оказывает влияние на диски SCSI. Может принимать значения:

- От 1 до 15 минут
- **Disabled** - запрещено
- **Doze Mode** (режим засыпания) - устанавливает время перехода или запрещает переход в первую стадию снижения энергопотребления. Может принимать значения:
 - **30 Sec, 1 Min, 2 Min, 4 min, 8 Min, 20 Min, 30 Min, 40 Min, 1 Hour** - время перехода (Sec - секунды, Min - минуты, Hour - час)
 - **Disabled** - запрещено
- **Standby Mode** (режим ожидания работы) - устанавливает время перехода или запрещает переход во вторую стадию снижения энергопотребления. Может принимать значения:
 - **30 Sec, 1 Min, 2 Min, 4 min, 8 Min, 20 Min, 30 Min, 40 Min, 1 Hour** - время перехода (Sec - секунды, Min - минуты, Hour - час)
 - **Disabled** - запрещено
- **Suspend Mode** (режим временной остановки)

В современных операционных системах предусмотрено управление электропитанием непосредственно через панель управления компьютера. Для этого необходимо в меню «Пуск» выбрать «Панель управления». В раскрытом окне выберите пункт меню «Электропитание», как показано на рис. 1



Задания:

1. Включить компьютер
2. В момент загрузки нажать клавишу « Del »

3. В меню Setup найдите раздел ,относящийся к управлению энергоснабжением. (В разных версиях BIOS этот раздел может называться по-разному).
4. Изучите опции энергоснабжения ,которые можно изменять в Setup.
Запишите основные параметры энергоснабжения, которые можно изменять в Setup
5. Загрузите компьютер. В данной лабораторной работе используется операционная система Windows . Найдите меню «Электропитание» и изучите основные характеристики. Запишите основные параметры, которые можно регулировать
6. Подготовьте отчет о проделанной работе

Лабораторная работа №8

Тема: Система охлаждения компьютера.

Цель: Изучить: устройство и принцип работы системы охлаждения компьютера

Ознакомиться: с устройством кулеров, вентиляторов, радиаторов.

Научиться выполнять: замену кулеров, смазку кулеров,

установку дополнительного вентилятора, заменять термопасту.

Перечень необходимых средств обучения :

Системный блок компьютера, материнская плата, кулеры,
вентиляторы, термопаста, инструменты

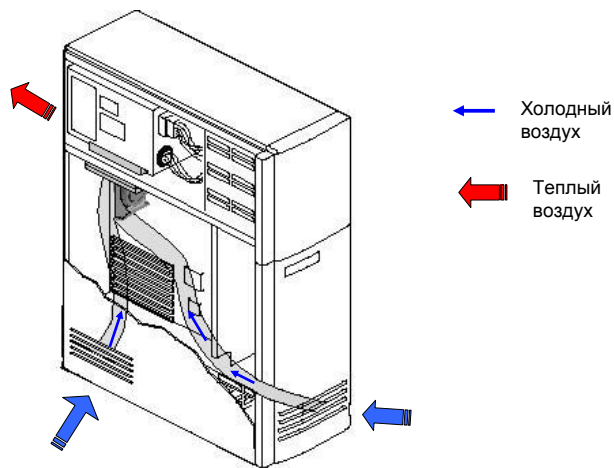
Краткие теоретические сведения

Система воздушного охлаждения

Современные стандарты по конструированию корпусов компьютеров регламентируют способ построения системы охлаждения. Начиная с 1997 года внедряется технология охлаждения компьютера сквозным воздушным потоком, направленным от передней стенки корпуса к задней (дополнительно воздух для охлаждения всасывается через левую стенку), рисунок 1

В любом месте внутри корпуса компьютера можно устанавливать дополнительные вентиляторы для усиления потоков воздуха. Обязательно нужно следовать правилу: на передней и левой боковой стенке воздух нагнетается внутрь корпуса, на задней стенке горячий воздух выбрасывается наружу.

Не рекомендуется работа компьютера со снятой крышкой т.к. нарушается движение воздушных потоков



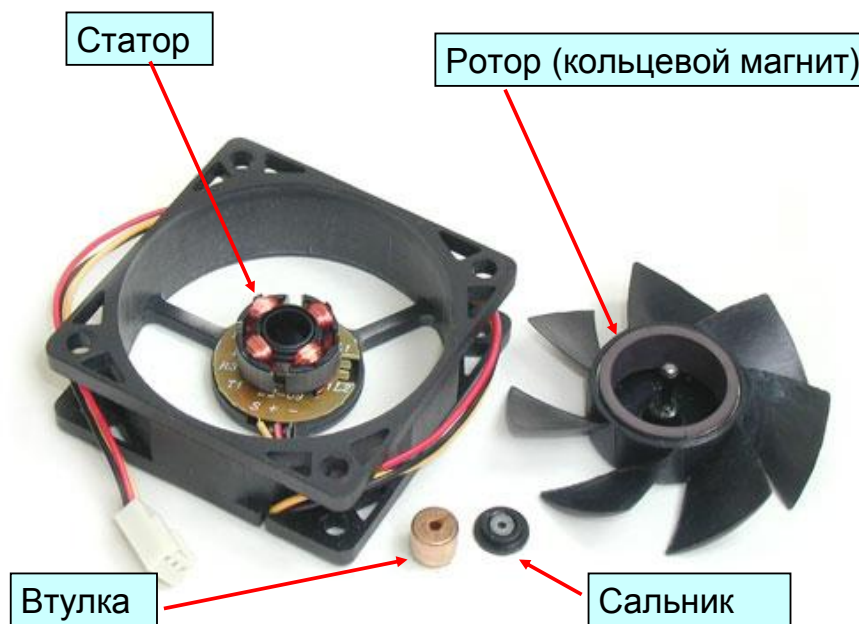
Направление движения воздушных потоков

Рис 1 Сквозной воздушный поток в корпусе компьютера

Движение воздушного потока обеспечивается многочисленными вентиляторами (кулерами). Вентиляторы устанавливаются в блок питания, на радиатор центрального процессора, на видеокарту и т.д.

Устройство вентилятора

Вентилятор состоит из корпуса (обычно в виде рамки), электродвигателя и крыльчатки, закреплённой при помощи подшипников на одной оси с двигателем. От типа установленных подшипников зависит надёжность вентилятора. Вентилятор с подшипником скольжения приведен на рис 2



Подшипник скольжения представляет собой бронзовую втулку. Стальной вал ротора опирается на втулку и имеет вращающийся стальной вал ротора. Рис 2 Вентилятор с подшипником скольжения. Дополнительно к этому втулка закрывается с торцов двумя резиновыми прокладками

(сальниками). Сальники служат в качестве препятствия вытеканию смазки из зазора вал-подшипник.

- Достоинства- низкая стоимость
- Недостатки:
 1. Со временем отверстие во втулке приобретает форму эллипса. Возрастает уровень шума
 2. Смазка из зазора вал-подшипник постепенно вытекает. Возрастает шум. Снижается надежность
 3. При возрастании температуры до 50 – 60 градусов срок службы уменьшается.

Вентилятор (1 подшипник скольжения, 1 подшипник качения)

Дополнительно к подшипнику скольжения (втулке) добавляется подшипник качения (рис 3). В этом случае:

- Основная нагрузка ложится на шариковый подшипник. Так как трение качения меньше трения скольжения, старт двигателя облегчается, рассеваемая вентилятором мощность уменьшается.
- Комбинированная конструкция менее восприимчива к весовому дисбалансу крыльчатки. Биения вала гасятся подшипником качения и срок службы втулки увеличивается.
- «Комбинированные» вентиляторы могут нормально функционировать в сложных эксплуатационных условиях (при высоких температурах окружающей среды и повышенной влажности воздуха).

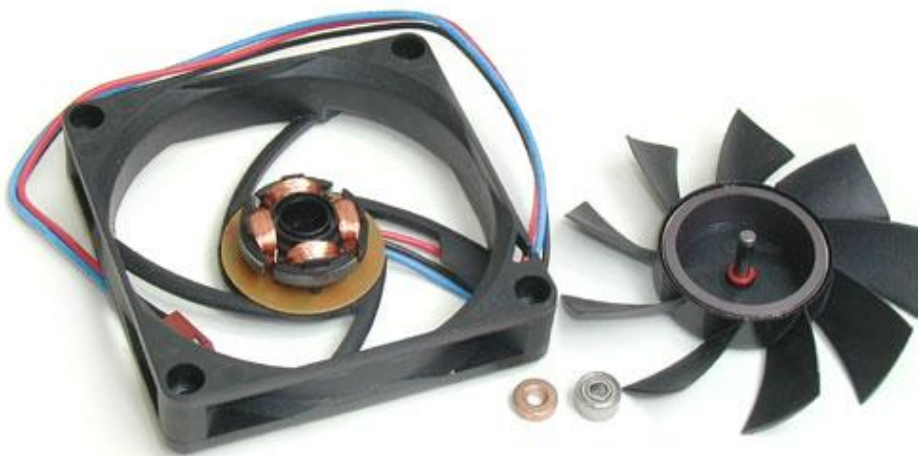


Рис 3. Вентилятор с комбинированными подшипниками

Недостаток комбинированных вентиляторов- вытекание смазки из втулки

Разборка и смазка вентилятора

Инструменты, которые могут понадобиться при сборке-разборке и чистке вентилятора.



Для разборки вентилятора сначала удаляем фирменную этикетку. Затем удаляем резиновую заглушку. (В мелких вентиляторах её функции может выполнять этикетка).



Далее острым скальпелем расширяем зазор стопорной шайбы. Вставляем в зазор тонкую отвёртку и раздвигаем концы шайбы в разные стороны.

Радиаторы

Радиатор является устройством ускоряющим теплообмен полупроводниковых (п/п) элементов с окружающей средой.

Площадь поверхности п/п элементов чрезвычайно мала и недостаточна для эффективного отвода тепловой мощности. Благодаря своей оребренной поверхности, радиатор в сотни раз увеличивает площадь его теплового контакта с окружающей

средой, способствуя тем самым усилению интенсивности теплообмена и снижению рабочей температуры. Фотографии радиаторов различных типов приведены на рис 5.

«Экструзионные» (прессованные) радиаторы.



Изготавливаются из алюминия. Наиболее распространенные.

«Складчатые» радиаторы



Ребра для рассеивания тепла изготовлены из металлической ленты. Материал – алюминий или медь.

Жидкостное охлаждение компьютера

Более эффективным и менее шумным является жидкостное охлаждение компьютера. Вместо воздуха через радиатор прокачивается жидкость. Вода отличается хорошей теплопроводностью и большой теплоемкостью. Циркулирующая жидкость обеспечивает гораздо лучший теплоотвод, чем поток воздуха. Жидкостный радиатор для процессора обычно меньше воздушного. Общая схема жидкостного охлаждения приведена на рис 7

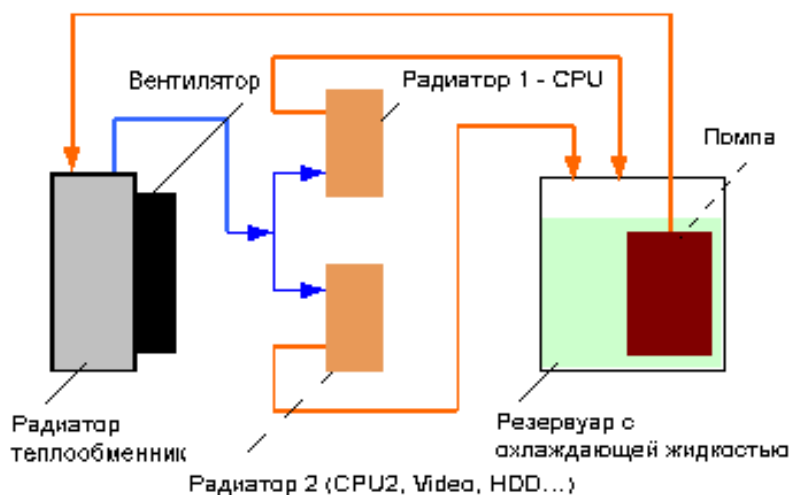


Рис 7 Общая схема жидкостного охлаждения

Задания

1. Ознакомьтесь с устройством охлаждения системного блока компьютера. Установите дополнительный вентилятор на заднюю панель системного блока и подключите его к питанию
2. Ознакомьтесь с устройством кулера. Выполните его разборку и смазку. Определите тип подшипника.
3. Выполните разборку охлаждающего устройства процессора. Определите тип и материал радиатора. Удалите старую термопасту и нанесите новую. Выполните сборку охлаждающего устройства.

Внимание! Аккуратно отсоединяйте пластиковые защелки. Не применяйте чрезмерных усилий. Паста наносится очень тонким слоем для лучшего контакта процессора и радиатора.

Пригласите преподавателя и продемонстрируйте разобранное и собранное устройство для охлаждения процессора. В случае повреждения оценка за работу снижается.

4. Подготовьте отчет о проделанной работе

Вопросы для проверки:

1. Что такое «хладагент» и какую функцию он выполняет?
2. Какие типы вентиляторов вы знаете? Чем они отличаются друг от друга?
3. Как организован воздушный поток охлаждения в блоках АТХ? Почему не рекомендуют снимать боковую крышку системного блока?
8. Что такое радиатор? Какие типы радиаторов вы знаете?
9. Как устроено жидкостное охлаждение компьютера?
10. Для чего предназначены помпа и радиатор-теплообменник в жидкостной системе охлаждения?
11. Чем отличаются радиаторы жидкостной и воздушной систем охлаждения?