

Министерство образования и науки Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Магнитогорский государственный технический университет  
им. Г.И. Носова»  
Многопрофильный колледж



**МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ  
ПРАКТИЧЕСКИХ И ЛАБОРАТОРНЫХ РАБОТ**

по учебной дисциплине  
**ОПЦ. 02 АРХИТЕКТУРА АППАРАТНЫХ СРЕДСТВ**

для студентов специальности  
**09.02.07 Информационные системы и программирование**

**Квалификация: разработчик web и мультимедийных приложений**

Магнитогорск, 2018

**ОДОБРЕНО:**

Предметно-цикловой комиссией  
Информатики и вычислительной техники  
Председатель И.Г.Зорина  
Протокол № 6 от 21.02.2018

Методической комиссией МпК

Протокол №4 от «01» марта 2018г

**Составитель:**

преподаватель ФГБОУ ВО «МГТУ им. Г.И. Носова» МпК Алексей Александрович Андре

Методические указания по выполнению практических и лабораторных работ разработаны на основе рабочей программы учебной дисциплины «Архитектура аппаратных средств».

Содержание практических и лабораторных работ ориентировано на подготовку обучающихся к освоению профессионального модуля программы подготовки специалистов среднего звена по специальности 09.02.07 Информационные системы и программирование и овладению общими компетенциями.

## СОДЕРЖАНИЕ

1 ПОЯСНИТЕЛЬНАЯ ЗАПИСКА	4
2 ПЕРЕЧЕНЬ ПРАКТИЧЕСКИХ/ЛАБОРАТОРНЫХ ЗАНЯТИЙ	6
3 МЕТОДИЧЕСКИЕ УКАЗАНИЯ	8
Практическое занятие 1	8
Лабораторное занятие 1	11
Лабораторное занятие 2	19
Лабораторное занятие 3	28
Лабораторное занятие 4	35
Лабораторное занятие 5	39
Лабораторное занятие 6	52
Лабораторное занятие 7	55
Лабораторное занятие 8	67
Лабораторное занятие 9	90
Лабораторное занятие 10	97
Лабораторное занятие 11	107
Лабораторное занятие 12	117
Лабораторное занятие 13	119

## 1 ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Состав и содержание практических и лабораторных занятий направлены на реализацию Федерального государственного образовательного стандарта среднего профессионального образования.

Ведущей дидактической целью практических занятий является формирование профессиональных практических умений (умений выполнять определенные действия), необходимые в последующем в профессиональной деятельности.

Ведущей дидактической целью лабораторных занятий является экспериментальное подтверждение и проверка существенных теоретических положений (законов, зависимостей).

В соответствии с рабочей программой учебной дисциплины «Архитектура аппаратных средств» предусмотрено проведение практических и лабораторных занятий. В рамках практического/лабораторного занятия обучающиеся могут выполнять одну или несколько практических/лабораторных работ.

В результате их выполнения, обучающийся должен:

**уметь:**

- У1. получать информацию о параметрах компьютерной системы;
- У2. подключать дополнительное оборудование и настраивать связь между элементами компьютерной системы;
- У3. производить инсталляцию и настройку программного обеспечения компьютерных систем;
- У01.3 определять этапы решения задачи;
- У01.4 выявлять и эффективно искать информацию, необходимую для решения задачи и/или проблемы;
- У02.2 определять необходимые источники информации;
- У02.3 планировать процесс поиска;
- У02.4 структурировать получаемую информацию;
- У04.8 эффективно работать в команде;
- У05.2 использовать навыки устного общения в профессиональной деятельности;
- У05.4 использовать стандартный набор коммуникационных технологий;
- У09.1 применять средства информационных технологий для решения профессиональных задач;
- У09.2 использовать современное программное обеспечение;
- У09.3 проявлять культуру информационной безопасности при использовании информационно-коммуникационных технологий;
- У10.6 понимать тексты на базовые профессиональные темы;
- У10.7 читать, понимать и находить необходимые технические данные и инструкции в руководствах в любом доступном формате.

Содержание практических и лабораторных занятий ориентировано на подготовку обучающихся к освоению профессионального модуля программы подготовки специалистов среднего звена по специальности и овладению **профессиональными компетенциями:**

ПК 5.2 Разрабатывать проектную документацию на разработку информационной системы в соответствии с требованиями заказчика.

ПК 5.3 Разрабатывать подсистемы безопасности информационной системы в соответствии с техническим заданием.

ПК 5.6 Разрабатывать техническую документацию на эксплуатацию информационной системы.

А также формированию **общих компетенций:**

ОК 1. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.

ОК 2. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности.

ОК 4. Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами.

ОК 5. Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста.

ОК 9. Использовать информационные технологии в профессиональной деятельности.

ОК 10. Пользоваться профессиональной документацией на государственном и иностранном языке.

Выполнение обучающихся практических и лабораторных работ по учебной дисциплине «Архитектура аппаратных средств» направлено на:

- обобщение, систематизацию, углубление, закрепление, развитие и детализацию полученных теоретических знаний по конкретным темам учебной дисциплины;

- формирование умений применять полученные знания на практике, реализацию единства интеллектуальной и практической деятельности;

- формирование и развитие умений: наблюдать, сравнивать, сопоставлять, анализировать, делать выводы и обобщения, самостоятельно вести исследования, пользоваться различными приемами измерений, оформлять результаты в виде таблиц, схем, графиков;

- приобретение навыков работы с различными приборами, аппаратурой, установками и другими техническими средствами для проведения опытов;

- развитие интеллектуальных умений у будущих специалистов: аналитических, проектировочных, конструктивных и др.;

- выработку при решении поставленных задач профессионально значимых качеств, таких как самостоятельность, ответственность, точность, творческая инициатива.

Практические и лабораторные занятия проводятся после соответствующей темы, которая обеспечивает наличие знаний, необходимых для ее выполнения.

## 2 ПЕРЕЧЕНЬ ПРАКТИЧЕСКИХ/ЛАБОРАТОРНЫХ ЗАНЯТИЙ

Разделы/темы	Темы практических/лабораторных занятий	Количество часов	Требования ФГОС СПО (уметь)
<b>Раздел 2. АРХИТЕКТУРА И ПРИНЦИПЫ РАБОТЫ ОСНОВНЫХ ЛОГИЧЕСКИХ БЛОКОВ СИСТЕМЫ</b>		<b>26</b>	
2.1 Логические основы ЭВМ, элементы и узлы	Практическая работа № 1 Решение арифметических и логических задач	4	У1
	Лабораторная работа № 1 Моделирование логических элементов	2	У1, У05.4,
	Лабораторная работа 2 Моделирование триггеров	2	У1, У09.1, У09.2
	Лабораторная работа 3 Моделирование регистров	2	У1, У09.1, У09.2
	Лабораторная работа 4 Моделирование работы счетчиков	2	У1, У09.1, У09.2
2.4 Технологии повышения производительности процессоров	Лабораторная работа 5 Системы команд процессора. Модель учебной ЭВМ	2	У1, У04.8, У09.1, У09.2, У09.3
	Лабораторная работа 6 Компиляция и запуск программ в машинных кодах	2	У1, У09.1, У09.2
2.5 Компоненты системного блока	Лабораторная работа 7 Сборка, установка, подключение комплектующих в корпус ПК	2	У2, У3, У01.3, У01.4, У02.2, У02.3, У02.4, У10.6
	Лабораторная работа 8 Установка конфигурации системы при помощи утилиты CMOS Setup	2	У3, У01.3, У01.4, У02.2, У02.4, У10.6
	Лабораторная работа 9 Сбор информации об установленном процессоре. Тестирование процессора на производительность и отказоустойчивость	2	У2, У01.4, У02.2, У10.7
2.6 Запоминающие устройства ЭВМ	Лабораторная работа 10 Тестирование оперативной памяти. Тестирование НЖМД	4	У2, У01.3, У01.4, У10.6, У10.7
<b>Раздел 3. ПЕРИФЕРИЙНЫЕ УСТРОЙСТВА</b>		<b>6</b>	
3.1 Периферийные устройства вычислительной техники	Лабораторная работа 11 Конструкция, подключение и инсталляция видеоадаптера. Тест монитора	2	У2, У3, У01.3, У01.4, У02.2, У02.3, У02.4, У10.6
	Лабораторная работа 12 Конструкция, подключение и инсталляция принтера	2	У2, У3, У01.3, У01.4, У02.2, У02.3, У02.4, У10.6
	Лабораторная работа 13 Конструкция, подключение и инсталляция сканера	2	У2, У3, У01.3, У01.4, У02.2, У02.3, У02.4,

			У10.6
ИТОГО		<b>32</b>	

### 3 МЕТОДИЧЕСКИЕ УКАЗАНИЯ

#### Тема 2.1 Логические основы ЭВМ, элементы и узлы

#### Практическое занятие № 1

#### Решение арифметических и логических задач

**Цель:** научиться решать арифметические и логические задачи

**Выполнив работу, Вы будете:**

**уметь:**

- выполнять перевод чисел из одной системы счисления в другую;
- выполнять арифметические операции над числами в разных системах счисления.

**Материальное обеспечение:**

Методические указания для выполнения практических работ

**Задание:**

1. Выполнить перевод из одной системы счисления в другую.
2. Выполнить арифметические операции в разных системах счисления.

**Порядок выполнения работы:**

**Задание 1.**

#### ВАРИАНТ № 1

1. Переведите данное число из десятичной системы счисления:
  - a)  $860_{10} \rightarrow X_8$ ;
  - b)  $785_{10} \rightarrow X_{16}$ ;
  - c)  $149,375_{10} \rightarrow X_2$ ;
  - d)  $461,25_{10} \rightarrow X_2$ ;
  - e)  $572_{10} \rightarrow X_8$ ;
2. Переведите данное число в десятичную систему счисления:
  - a)  $1001010_2 \rightarrow X_{10}$ ;
  - b)  $775,11_8 \rightarrow X_{10}$ ;
  - c)  $111001010_2 \rightarrow X_{10}$ ;
  - d)  $414,1_8 \rightarrow X_{10}$ ;
  - e)  $1010110011_2 \rightarrow X_{10}$ ;
3. Переведите данное число из двоичной системы счисления:
  - a)  $1000111011_2 \rightarrow X_8$ ;
  - b)  $1011001001_2 \rightarrow X_{16}$ ;
  - c)  $1110001_2 \rightarrow X_8$ ;
  - d)  $110111010_2 \rightarrow X_{16}$ ;

#### ВАРИАНТ № 2

1. Переведите данное число из десятичной системы счисления:
  - a)  $250_{10} \rightarrow X_8$ ;
  - b)  $757_{10} \rightarrow X_{16}$ ;
  - c)  $711,25_{10} \rightarrow X_2$ ;
  - d)  $413,5625_{10} \rightarrow X_2$ ;
  - e)  $485_{10} \rightarrow X_8$ ;
2. Переведите данное число в десятичную систему счисления:
  - a)  $1111000_2 \rightarrow X_{10}$ ;
  - b)  $1233,5_8 \rightarrow X_{10}$ ;
  - c)  $10101000_2 \rightarrow X_{10}$ ;
  - d)  $721,2_8 \rightarrow X_{10}$ ;
  - e)  $1010000_2 \rightarrow X_{10}$ ;
3. Переведите данное число из двоичной системы счисления:
  - a)  $10110111_2 \rightarrow X_8$ ;
  - c)  $100101000_2 \rightarrow X_8$ ;

- b)  $1001000011_2 \rightarrow X_{16}$ ;      d)  $1000010101_2 \rightarrow X_{16}$ ;

ВАРИАНТ № 3

1. Переведите данное число из десятичной системы счисления:  
a)  $759_{10} \rightarrow X_8$ ;      d)  $284.375_{10} \rightarrow X_2$ ;  
b)  $265_{10} \rightarrow X_{16}$ ;      e)  $772_{10} \rightarrow X_8$ ;  
c)  $79.4375_{10} \rightarrow X_2$ ;
2. Переведите данное число в десятичную систему счисления:  
a)  $1001101_2 \rightarrow X_{10}$ ;      d)  $675.2_8 \rightarrow X_{10}$ ;  
b)  $1461.15_8 \rightarrow X_{10}$ ;      e)  $1000001111_2 \rightarrow X_{10}$ ;  
c)  $1111011011_2 \rightarrow X_{10}$ ;
3. Переведите данное число из двоичной системы счисления:  
a)  $1100110010_2 \rightarrow X_8$ ;      c)  $111101101_2 \rightarrow X_8$ ;  
b)  $1001101101_2 \rightarrow X_{16}$ ;      d)  $101111010_2 \rightarrow X_{16}$ ;

ВАРИАНТ № 4

1. Переведите данное число из десятичной системы счисления:  
a)  $216_{10} \rightarrow X_8$ ;      d)  $597.25_{10} \rightarrow X_2$ ;  
b)  $336_{10} \rightarrow X_{16}$ ;      e)  $530_{10} \rightarrow X_8$ ;  
c)  $741.125_{10} \rightarrow X_2$ ;
2. Переведите данное число в десятичную систему счисления:  
a)  $1100000110_2 \rightarrow X_{10}$ ;      d)  $176.5_8 \rightarrow X_{10}$ ;  
b)  $1537.22_8 \rightarrow X_{10}$ ;      e)  $1011000011_2 \rightarrow X_{10}$ ;  
c)  $110001111_2 \rightarrow X_{10}$ ;
3. Переведите данное число из двоичной системы счисления:  
a)  $110011110_2 \rightarrow X_8$ ;      c)  $1111111_2 \rightarrow X_8$ ;  
b)  $1010101101_2 \rightarrow X_{16}$ ;      d)  $1000101110_2 \rightarrow X_{16}$ ;

**Задание 2.**

Вариант 1

- 1)  $1011101000+1111001110=$
- 2)  $1101110011-1101010110=$
- 3)  $100000111+1010100000=$
- 4)  $1110111110-1110101111=$
- 5)  $100110010+10011101=$
- 6)  $1011111000-101010100=$
- 7)  $110100*100101=$
- 8)  $111010*1000111=$
- 9)  $1011111000100/1101=$
- 10)  $100101001100/111=$

Вариант 2

- 1)  $1001010000+1100011011=$
- 2)  $1010000010-100000110=$
- 3)  $10100001+1010110111=$
- 4)  $100011010-100010010=$
- 5)  $1100110110+1000001010=$

- 6)  $11111000-10100101=$
- 7)  $111001*100111=$
- 8)  $1001011*1010011=$
- 9)  $101101111100/110=$
- 10)  $1001111110111/1101=$

#### Вариант 3

- 1)  $1001011001+1100110000=$
- 2)  $1000100111-101101010=$
- 3)  $1011101001+1110011001=$
- 4)  $1010011001-1001101011=$
- 5)  $1101101010+11101100=$
- 6)  $110110101-11100111=$
- 7)  $1000000*1010010=$
- 8)  $111111*1001111=$
- 9)  $1100100011111/10011=$
- 10)  $100011101000/101=$

#### Вариант 4

- 1)  $10001001+10000110=$
- 2)  $1110001001-101001010=$
- 3)  $1110110011+1110110001=$
- 4)  $100001001-1011010=$
- 5)  $1110111110+1000101000=$
- 6)  $1001110110-100011111=$
- 7)  $101000*10000=$
- 8)  $1000110*100000=$
- 9)  $10100001011100/10010=$
- 10)  $110110001100/1100=$

#### **Форма представления результата:**

Отчет о проделанной работе.

#### **Критерии оценки:**

Оценка «отлично» ставится, если задание выполнено верно.

Оценка «хорошо» ставится, если ход выполнения задания верный, но была допущена одна или две ошибки, приведшие к неправильному результату.

Оценка «удовлетворительно» ставится, если приведено неполное выполнение задания.

Оценка «неудовлетворительно» ставится, если задание не выполнено.

## Тема 2.1 Логические основы ЭВМ, элементы и узлы

### Лабораторное занятие № 1

#### Моделирование логических элементов

**Цель:** Изучение интерфейса пользователя. Ознакомление с терминами и понятиями. Построение простейших электронных и логических схем.

#### Выполнив работу, Вы будете:

##### *уметь:*

- моделировать схемы на логических элементах;
- использовать коммуникационные технологии.

#### Материальное обеспечение:

Методические указания для выполнения лабораторных работ, компьютер, программное обеспечение: Multisim Education.

#### Задание:

Собрать схему логического элемента "И".

#### Краткие теоретические сведения:

##### Основные команды меню:

#### 1. Меню *File*:

первые четыре команды меню типовые и пояснений не требуют.

- **Revert to Saved** -стирание всех изменений, внесенных в текущем сеансе редактирования, и восстановление схемы в первоначальном виде.

- **Install** - установка дополнительных программ с жёстких дисков.

- **Import** - импорт текстовых файлов описания схемы .

- **Export** - составление текстового описания схемы и задания на моделирование в формате SPICE.

#### 2. Меню *Edit*:

- **CUT** - стирание (вырезание) выделенной части схемы с сохранением в буфере обмена. Выделение одного компонента производится щелчком мыши на изображении компонента. Для выделения части схемы или нескольких компонентов курсор мыши в левый угол воображаемого прямоугольника, охватывающего выделяемую часть, нажать левую кнопку мыши и, не отпуская её, протянуть курсор по диагонали этого прямоугольника, контуры которого появляются уже в начале движения мыши, и затем отпустить кнопку. Выделенные компоненты окрашиваются в красный цвет.

- **COPY** - копирование выделенной части схемы в буфер обмена.

- **PASTE**- вставка содержимого буфера обмена на рабочее поле программы.

Фрагмент затем ещё будучи отмеченным перетаскивается с помощью мыши в нужное место.

- **DELETE** - стирание выделенной части схемы.

- **SELECT ALL** - выделение всей схемы.

- **Show Clipboard**- показать содержимое буфера обмена.

- **Copy as Bitmap** - копирует выделенный участок в буфер обмена.

3. Меню *Circuit* и *Analysis* - используется при подготовке схем, а также для задания параметров моделирования.

- **Rotate**- вращение выделенного компонента.

- **Subcircuit** - преобразование предварительно выделенной части схемы в подсхему.

- **Zoom** - раскрытие (развёртывание) выделенной подсхемы или контрольно-измерительного прибора, команда выполняется также двойным щелчком мыши по иконке компонента или прибора.

- **Activate** - запуск моделирования.

- **Pause** - прерывание моделирования.

- **Stop** - остановка моделирования. Эти две команды дублируются нажатием кнопки выключателя, расположенного в правом верхнем углу экрана.

### Технология создания схем

Для создания схем, рассматриваемых в рамках лабораторных работ по курсу "Архитектура аппаратных средств" достаточно воспользоваться имеющимися типовыми компонентами.

Для открытия нужной библиотеки компонентов нужно подвести курсор мыши к соответствующей иконке и нажать один раз её левую кнопку. В выпадающем множестве выбирается необходимый значок, и передвигается при удержании левой клавиши мыши на рабочее поле программы. Для установки параметров необходимо двойным нажатием левой кнопкой мыши раскрыть меню настройки параметров компонента. Выбор подтверждается нажатием кнопкой **Accept** и клавишей **Enter**.

После размещения компонентов производится соединение их выводов проводниками. При этом необходимо учитывать, что к выводу компонента можно подключить только один проводник.

Для выполнения подключения курсор мыши подводится к выводу компонента и после появления прямоугольной площадки синего цвета, нажимается левая кнопка и появляющийся при этом проводник протягивается к выводу другого компонента до появления на нём такой же прямоугольной площадки, после чего кнопка мыши отпускается и соединение готово. При необходимости подключения к этим выводам других проводников в библиотеке **Passive** выбирается точка (символ соединения) и переносится на ранее установленный проводник. После удачной постановки точки к проводнику подсоединяется ещё два проводника.

Точка соединения может быть использована не только для подключения проводников, но и для введения надписей.

Если необходимо переместить отдельный сегмент проводника, к нему подводится курсор, нажимается левая кнопка и после появления в вертикальной или горизонтальной плоскости двойного курсора производятся нужные перемещения.

Подключение к схеме контрольно-измерительных приборов производится аналогично. Причём для таких приборов, как осциллограф или логический анализатор, соединения целесообразно проводить цветными проводниками, поскольку их цвет определяет соответствующую осциллограмму.

### Основные компоненты

#### 1. Компонент **Выход из программы**.



#### 2. Вспомогательные компоненты - группа **SOURCES**:

- заземление (метка), точка нулевого потенциала в схеме.



- источник фиксированного напряжения +5 вольт
- генератор однополярных прямоугольных импульсов (амплитуда, частота, коэффициент заполнения).

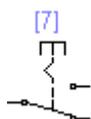


### 3 Основные пассивные элементы - группа BASIC:

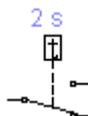
- точка соединения проводников, используется также для введения на схему надписей длиной не более 14 символов (других способов введения текста в EWB не существует).



- переключатель, управляемый нажатием задаваемой клавишей клавиатуры (в квадратных скобках), по умолчанию - клавиша пробела.



- переключатель, автоматически срабатывающий через заданное время на включение и выключение (время в секундах).



### 4. Индикаторные приборы - группа INDICATORS.



- светоиндикатор (свет свечения может быть настроен красным, зелёным и синим)



- семисегментный индикатор с дешифратором



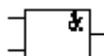
- семисегментный индикатор

10 W 12 V

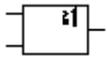


- лампа накаливания.

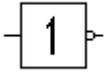
### 5. Логические элементы - группа LOGIC GATES



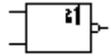
- логический элемент "И"



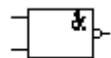
- логический элемент "ИЛИ"



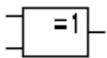
- логический элемент "НЕ"



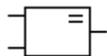
- логический элемент "ИЛИ-НЕ"



- логический элемент "И-НЕ"

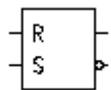


- логический элемент исключающее "ИЛИ"

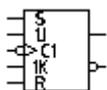


- логический элемент импликация

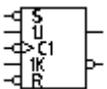
#### 4.6. Комбинированные цифровые компоненты.



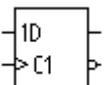
- асинхронный RS-триггер



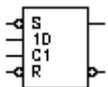
- универсальный JK-триггер с прямым тактовым входом и входами предустановки



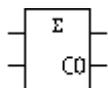
- универсальный JK-триггер с инверсным тактовым входом и инверсными входами предустановки



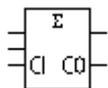
- D-триггер без предустановки



- D- со входами предустановки



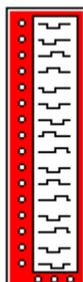
- полусумматор



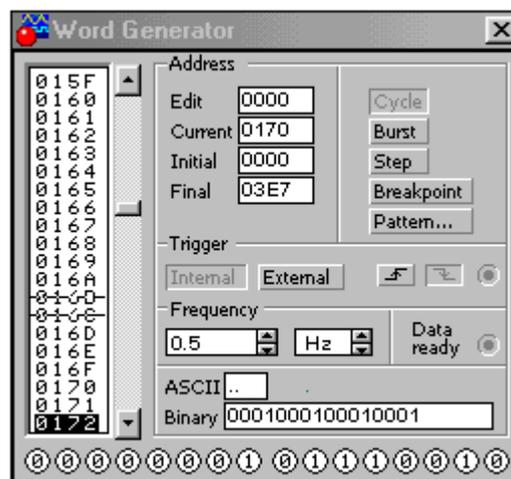
- полный сумматор

## 7. Приборы, группа **INSTRUMENTS**.

- логический анализатор



- генератор слова - **Word Generator**



На рисунке генератор слова показан в развёрнутом виде. Генератор (или кодовый генератор) предназначен для генерации 16-ти разрядных двоичных слов, которые набираются пользователем на экране, расположенном в левой части лицевой панели. Для набора двоичных комбинаций необходимо щёлкнуть мышью на соответствующем разряде и затем ввести с клавиатуры число в десятичном коде.

Сформированные слова выдаются на шестнадцать расположенных в нижней части прибора выходных клемм-индикаторов:

- с индикацией в двоичном коде в строке окна binary;
- в пошаговом (step), циклическом (cycle) или с выбранного слова до конца (при нажатии кнопки BURST) при заданной частоте посылок (установка- заданием частоты в окнах FREQUENCY);

- при внутреннем или внешнем запуске ( при нажатии кнопки EXTERNAL, справа верхняя клемма служит для подключения сигнала синхронизации);

- при запуске по переднему или заднему фронту сигнала синхронизации служит кнопка

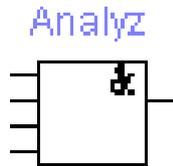


- на правую нижнюю клемму выдается выходной синхронизирующий импульс.

**Порядок выполнения работы:**

**В группе Logic Gates, выбирается логический элемент "И".**

Двумя щелчками мыши на изображении логического элемента переходим к

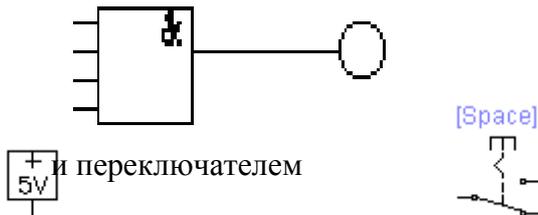


настройкам параметров логического элемента "И". Выбираем количество входов, например 4.

Можно присвоить название логическому элементу.

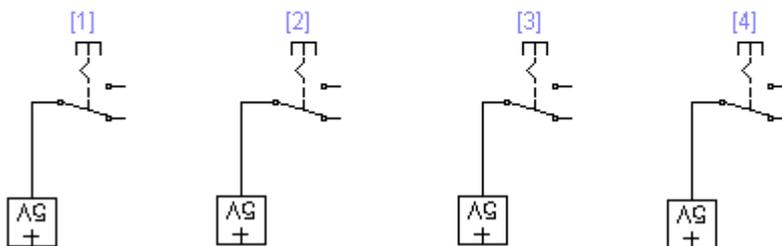
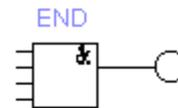
К выходу логического элемента присоединяем из группы **INDICATORS** красный светодиод.

Для получения логического сигнала (0 или 1) удобно воспользоваться источником



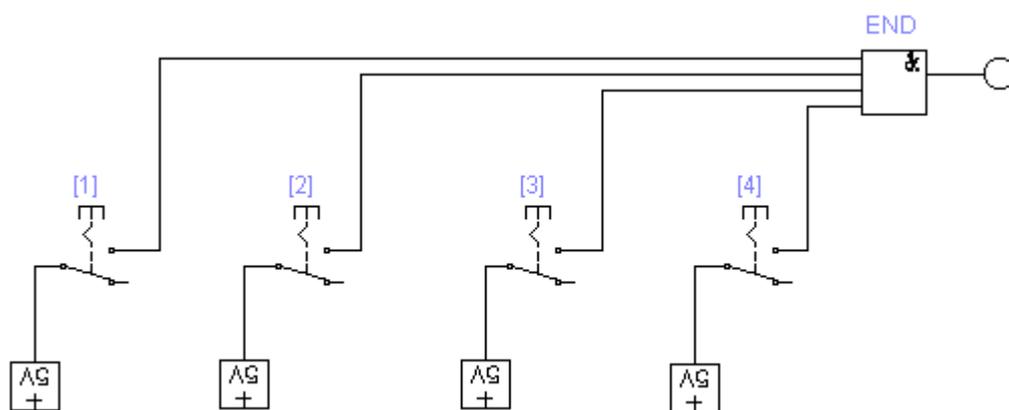
напряжения и переключателем

Затем набираем 4 источника и 4 переключателя

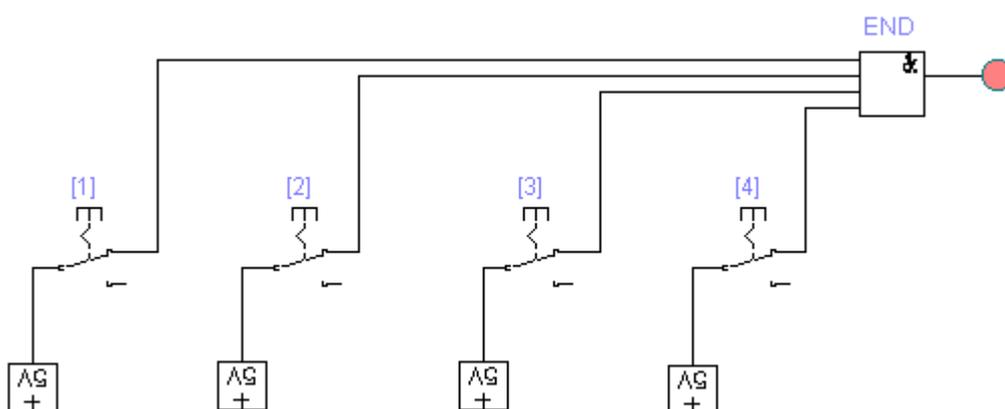


При этом присваиваем каждому переключателю клавишу переключения

Затем соединяем входы логической схемы "И" с каждым из переключателей.



Проверка состоит в подаче различных кодовых комбинаций на вход логической схемы.



На выходе логической схемы "И" появляется логическая 1 (горит светодиод) только при подаче логических 1 (потенциал 5 вольт) на все четыре входа логической схемы "И".

### Контрольные вопросы

1. Какие логические элементы имеются в библиотеке EWB?
2. Какие параметры являются настраиваемыми в генераторе прямоугольных импульсов.
3. Как производится соединение более двух входов или выходов между собой?
4. С помощью каких элементов можно смоделировать подачу логической 1 и логического 0?

### Форма представления результата:

1. Правила запуска и настройки программного моделирующего комплекса EWB.
2. Перечень основных элементов из библиотеки EWB, необходимый для моделирования логических схем и цифровых устройств ЭВМ.
3. Примеры, выполненные по заданию преподавателя. Последовательность составления моделируемой схемы.

### Критерии оценки:

- Оценка «отлично» ставится, если задание выполнено верно.
- Оценка «хорошо» ставится, если ход выполнения задания верный, но была допущена одна или две ошибки, приведшие к неправильному результату.
- Оценка «удовлетворительно» ставится, если приведено неполное выполнение задания.
- Оценка «неудовлетворительно» ставится, если задание не выполнено.

## Тема 2.1 Логические основы ЭВМ, элементы и узлы

### Лабораторное занятие № 2

#### Моделирование триггеров

**Цель:** Ознакомление с возможностями моделирования работы схем триггеров. Исследование работы схем триггеров различных типов, исследование временных диаграмм работы триггеров.

#### Выполнив работу, Вы будете:

##### *уметь:*

- строить схемы триггеров;
- строить временные диаграммы триггеров;
- использовать современное программное обеспечение при решении профессиональных задач.

#### Материальное обеспечение:

Методические указания для выполнения лабораторных работ, компьютер, программное обеспечение: Multisim Education.

#### Задание:

1. Исследовать асинхронные и синхронных R-S триггеры.
2. Исследовать работу D-триггера.

#### Краткие теоретические сведения:

##### Основные понятия

Функциональные узлы и устройства ЭВМ синтезируются на основе двух типов логических схем: комбинационных схем (КС), и элементов памяти (ЭП), в которых хранятся результаты этих операций для использования в последующих операциях. В качестве ЭП в узлах и устройствах ЭВМ наибольшее распространение получили триггерные устройства или, как их чаще называют - триггеры.

##### Триггеры как цифровые автоматы.

Триггер представляет собой устройства с двумя устойчивыми состояниями, содержащее элемент памяти (собственно триггер) и схему управления, выполненную, как правило, с помощью КС. Схема управления преобразует поступающую на ее входы  $X_1, X_2 \dots X_n$  информацию в комбинацию сигналов, действующих непосредственно на входы собственно триггера. При этом информационные входы триггера отождествляются с входными переменными и имеют следующие обозначения:

$S$  (от английского Set - установка) - вход для асинхронной установки триггера в состояние 1 ( $S$  - вход);

$R$  (от англ. Reset - сброс) - вход для асинхронной установки триггера в состояние 0 ( $R$  - вход);

$D$  (от англ. Delay - задержка) - информационный вход для установки триггера в состояние 1 или 0 ( $D$  - вход);

$T$  (от англ. Toggle - кувыркаться) - счетный вход ( $T$  - вход);

$J$  - вход для синхронной установки состояния 1 в универсальном J-K - триггере ( $J$  - вход);

К - вход для синхронной установки состояния 0 в универсальном J-K - триггере (К - вход);

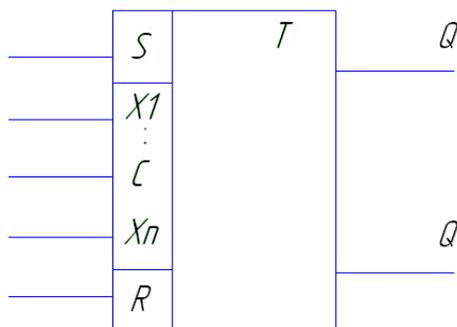
С - вход синхронизации (С - вход).

Асинхронные входы R и S могут быть прямыми и инверсными. Инверсные входы обозначаются знаком инверсии. Для инверсных входов информационным сигналом является уровень логического "0" (низкий уровень) на уровень логической "1" (высокий уровень) триггер не реагирует. Для прямых входов наоборот информационным сигналом является уровень логического "1".

Выход же триггера принято отождествлять с его внутренним состоянием и обозначать символом Q. Подавляющее число схем триггеров имеет два выхода: прямой Q и инверсный  $\bar{Q}$ . В установившемся состоянии всегда, если  $Q = 1$ , то  $\bar{Q} = 0$  и, если  $Q = 0$ , то  $\bar{Q} = 1$ . При этом считают, что триггер находится в состоянии "1", если на прямом выходе Q имеется высокий уровень напряжения ( $Q = 1$ ,  $\bar{Q} = 0$ ), и в состоянии "0", если на прямом выходе имеется низкий уровень напряжения ( $Q = 0$ ,  $\bar{Q} = 1$ ).

#### Обозначение триггеров на функциональных схемах.

Примеры условного обозначения триггеров на функциональных схемах в соответствии с ГОСТ 2.743-91 (ЕСКД) показаны на рисунке. Если триггер содержит входную логику, управляющую процессом занесения в него информации, то в прямоугольнике, условно изображающем триггер, отделяется чертой левее дополнительное поле, а в нем показывается вход "С" синхронизирующего сигнала и отмечаются функциональные назначения информационных входных сигналов X1, X2 . . . Xn. В основном поле прямоугольника ставится символ T для обозначения триггера. Дополнительное поле может быть разделено на две части: асинхронную и синхронную. В первой проставляются символы S и R входов асинхронной установки триггеров в 1 и 0, во второй на местах X1, X2 . . . Xn - символы, относящие данный триггер к тому или иному функциональному типу.



#### Классификация триггеров.

В настоящее время в интегральной микросхемотехнике наиболее распространенными являются триггеры и логические элементы потенциального типа. В основу классификации этих триггеров положены два основных признака:

1) Функциональный - этот признак определяет назначение триггера и в ряде случаев является решающим при выборе типа триггера для проектируемого вычислительного устройства или узла. По указанному признаку различают триггеры R-S, D-, T-, J-K и др. типов.

2) Способ записи информации в триггер - этот признак характеризует способ записи информации и временную диаграмму работ триггера, т.е. определяет ход процесса записи информации в триггер. По этому признаку триггеры подразделяются на две группы:

а) асинхронные;

б) синхронные;

Запись информации в асинхронный триггер осуществляется в произвольный момент времени непосредственно с поступлением информационного сигнала на вход триггера.

Синхронные триггеры помимо информационных входов содержат один или несколько синхронизирующих входов (вход С на рис. 2.1). Запись информации в такие триггеры осуществляется только при подаче синхронизирующего импульса (СИ). В свою очередь, синхронные триггеры подразделяются на триггеры, работающие по уровню СИ (без задержки) и на триггеры с внутренней задержкой. В первых срабатывание происходит одновременно с поступлением СИ, а во вторых - после окончания действия СИ.

#### Триггеры R-S - типа

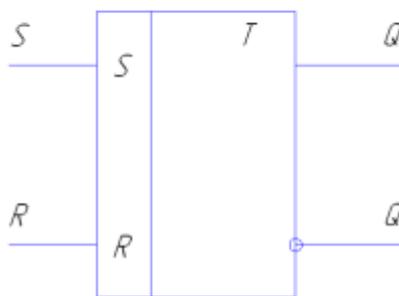
Асинхронным триггером R-S типа (R-S - триггером) называется логическое устройство с двумя устойчивыми состояниями, имеющее два информационных входа R и S, такие что, при  $S = 1$  и  $R = 0$  триггер принимает состояние 1 ( $Q = 1$ ), а при  $R = 1$  и  $S = 0$  триггер принимает состояние 0 ( $Q = 0$ ). Закон функционирования R - S - триггера с прямыми входами отобран в таблице переходов (таблица 1).

Таблица 1

R	S	Q
0	0	не меняется
0	1	1
1	0	0
1	1	запрещено

Как следует из табл.1 состояние R - S -триггера не изменяется, если на обоих входах схемы действуют сигналы с уровнем логического нуля. В случае одновременного поступления на входы R и S логических 1, триггер принимает неопределенное состояние. Поэтому логические устройства на основе триггеров должны строиться с учетом исключения комбинаций сигналов  $R = S = 1$ .

Обозначение триггера показано на рисунке.



#### Синхронный R-S -триггер с прямыми входами.

В отличие от асинхронного этот триггер на каждом информационном входе имеет дополнительные схемы совпадения, первые входы которых объединены и на них подаются синхронизирующие сигналы. Вторые входы схем сравнения являются информационными.

Таким образом, наличие схем совпадения определяет то обстоятельство, что триггер будет срабатывать от сигналов R и S только при наличии синхронизирующего импульса.

На рисунке представлено обозначение синхронного R-S - триггера с прямыми входами. Закон функционирования синхронного R-S - триггера с прямыми входами представлен в таблице переходов (таблица 2).

Таблица 2.

<b>C</b>	<b>S</b>	<b>R</b>	<b>Q</b>
<b>0</b>	<b>0</b>	<b>0</b>	<b>не меняется</b>
<b>0</b>	<b>0</b>	<b>1</b>	<b>не меняется</b>
<b>0</b>	<b>1</b>	<b>0</b>	<b>не меняется</b>
<b>0</b>	<b>1</b>	<b>1</b>	<b>не меняется</b>
<b>1</b>	<b>0</b>	<b>0</b>	<b>не меняется</b>
<b>1</b>	<b>0</b>	<b>1</b>	<b>0</b>
<b>1</b>	<b>1</b>	<b>0</b>	<b>1</b>
<b>1</b>	<b>1</b>	<b>1</b>	<b>запрещено</b>

Комбинация R=S=C=1 является запрещенной и должна быть исключена при функционировании схем с использованием синхронных R-S - триггеров.

#### Триггеры D-типа (D-триггеры).

Триггером D-типа (D-триггером) называется логическое устройство с двумя устойчивыми состояниями и одним информационным входом D. В триггерах D-типа значение переменной в момент времени  $t^n$  совпадает со значением входной переменной в момент времени  $t^{n+1}$ , в связи с чем, такой триггер часто в литературе называют триггером задержки.

Следует отметить, что асинхронный триггер D-типа не нашел практического применения.

Из этого уравнения видно, что при наличии синхронизирующего сигнала (C=1) триггер, переходит в состояние  $Q^{n+1} = D_n$ , а при отсутствии тактирующего сигнала (C=0) триггер сохраняет предыдущее состояние.

На рисунке представлено обозначение синхронного триггера D-типа без задержки управляемого уровнем синхронизирующего импульса. Вход D является информационным, а вход C - синхронизирующим.

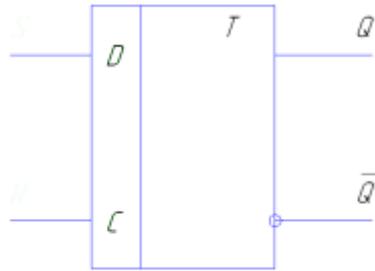
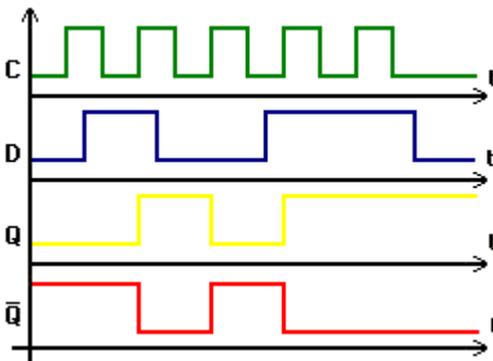


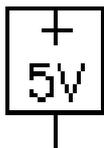
Таблица переходов.

C	D	$Q_{t+1}$	Примеч.
0	*	$Q_t$	Хранен.
1	0	0	Устан. 0
1	1	1	Устан. 1

Пример.



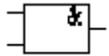
## 2Используемые элементы программы



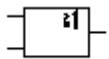
1) Voltage Source. С помощью этого источника на вход триггеров и логических элементов подается логическая единица.



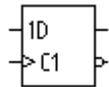
2) Переключатель (Basic->Switch). Переключение производится нажатием на клавишу, указанную в скобках над этим элементом.



3) Логический элемент "И" (Logic gates->2-Input AND gate).



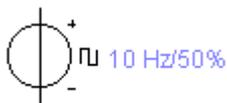
4) Логический элемент "ИЛИ" (Logic gates->2-Input OR gate).



5) Триггер типа D (Digital->D Flip-Flop)



6) Светоиндикатор (Indicators->red Prob). При подаче на этот элемент логической единицы светодиод загорается красным цветом.



7) Источник прямоугольных сигналов (Sources->Clock). Выполняет функцию синхронизатора, который обеспечивает одновременность подачи сигналов на входы триггеров.

### Порядок выполнения работы:

#### Исследование асинхронных и синхронных R-S триггеров.

1. Простейшие схемы асинхронных R-S триггеров составить с использованием логических элементов "И-НЕ", "ИЛИ-НЕ".

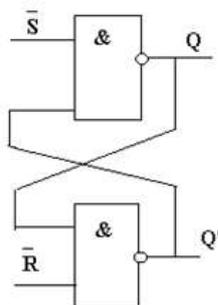


Рис. Асинхронный триггер на логических элементах И-НЕ

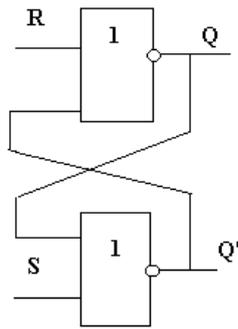


Рис. Асинхронный R-S триггер на логических элементах "ИЛИ-НЕ"

Для подачи на входы триггера значений логического "0" и "1" воспользоваться 5-вольтовым источником и переключателем. Для контроля значений на прямом выходе триггера воспользоваться светоиндикатором.

Проконтролировать работу триггера по таблице состояний. Найти и объяснить отличия в управлении асинхронного триггера на элементах "ИЛИ-НЕ".

Составить таблицы состояний работы триггеров.

2. Для построения синхронного R-S триггера необходимо добавить логические схемы "И-НЕ" для анализа синхросигнала и добавить источник синхросигнала. В качестве синхросигнала можно взять 5-вольтовый источник и переключатель.

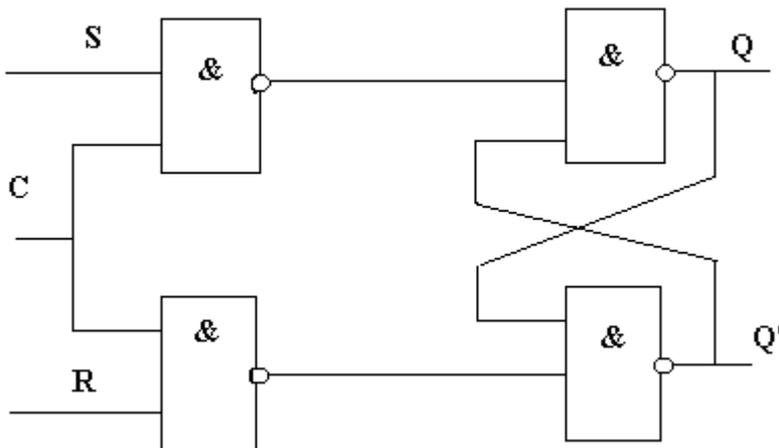
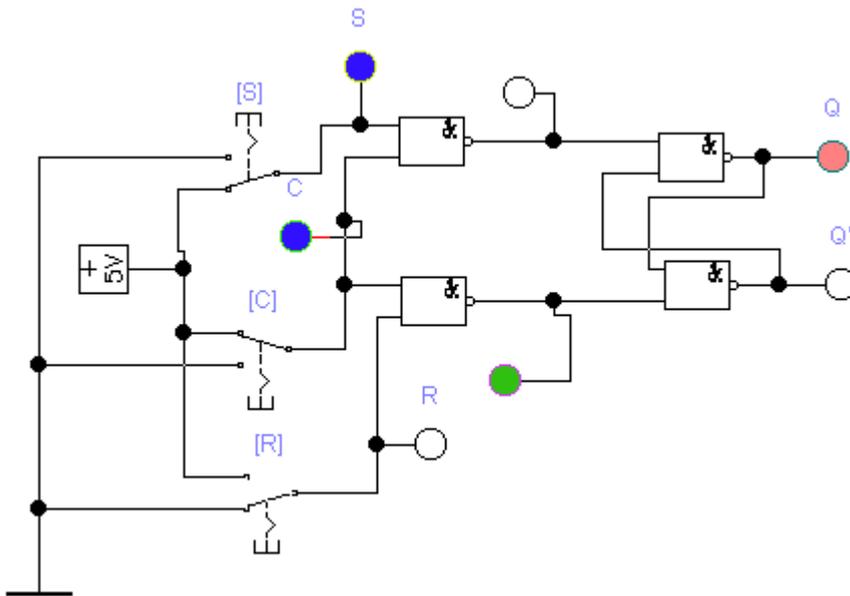


Рис. Синхронный R-S триггер на логических элементах "И-НЕ"

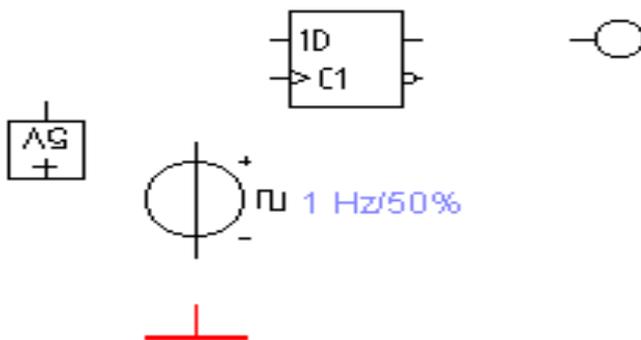
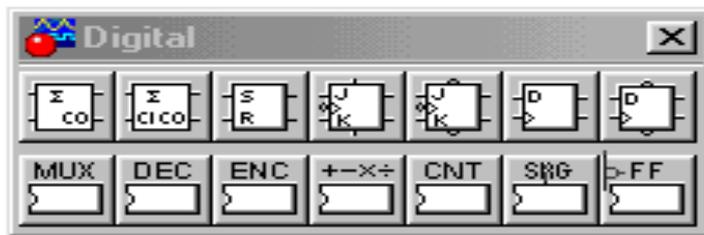
Пример построения модели схемы синхронного R-S триггера.



Составить таблицу состояний синхронного R-S триггера.

### Исследование работы D-триггера

Собрать схему, моделирующую работу D-триггера из библиотечного набора.



Из данного набора элементов собирается схема проверки работы D-триггера. Составить таблицу переходов D-триггера.

### Контрольные вопросы

1. К какому классу устройств относятся триггеры: к комбинационным схемам или цифровым автоматам и в чём основное отличие?
2. По каким признакам классифицируются триггерные устройства?

3. Каковы отличия синхронных триггеров от асинхронных триггеров?

**Форма представления результата:**

1. Схемы, моделирующие работу триггеров по всем пунктам задания.
2. Таблицы переходов для каждой моделируемой схемы триггера.

**Критерии оценки:**

Оценка «отлично» ставится, если задание выполнено верно.

Оценка «хорошо» ставится, если ход выполнения задания верный, но была допущена одна или две ошибки, приведшие к неправильному результату.

Оценка «удовлетворительно» ставится, если приведено неполное выполнение задания.

Оценка «неудовлетворительно» ставится, если задание не выполнено.

## Тема 2.1 Логические основы ЭВМ, элементы и узлы

### Лабораторное занятие № 3

#### Моделирование регистров

**Цель:** Ознакомление с возможностями моделирования работы схем регистров. Исследование схем регистров различного назначения, исследование преобразования параллельного и последовательного кода.

**Выполнив работу, Вы будете:**

**уметь:**

- строить схемы регистров;
- использовать современное программное обеспечение при решении профессиональных задач.

**Материальное обеспечение:**

Методические указания для выполнения лабораторных работ, компьютер, программное обеспечение: Multisim Education.

**Задание:**

1. Составить схему регистра параллельного занесения двоичного кода
2. Составить схему сдвигающего регистра

**Краткие теоретические сведения:**

**Основные понятия**

Регистр - это электронное устройство для запоминания (хранения) слова, а также для выполнения над словами некоторых логических операций.

Регистры выполняют следующие основные операции:

- установка регистра в нулевое состояние (сброс);
- приём слова из другого регистра;
- передача слова в другой регистр;
- сдвиг вправо и влево на требуемое число разрядов;
- преобразование последовательного кода в параллельный и параллельного в последовательный.

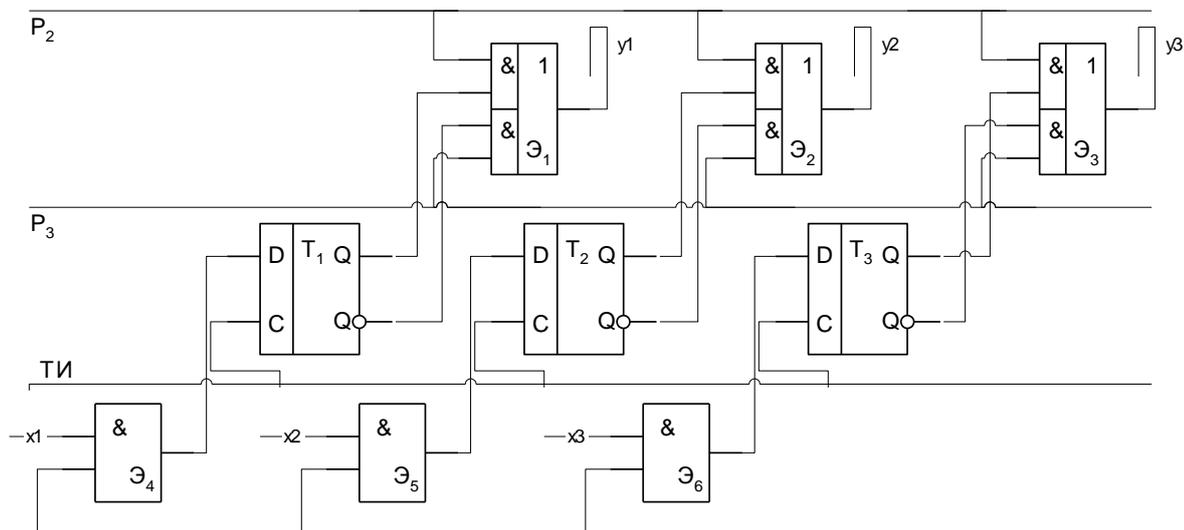
Регистры, как правило, строятся на D-триггерах, т.к. эти триггеры позволяют производить запись информации однофазным кодом без предварительного "обнуления" по входу D (см. лаб. раб.2).

Приём информации в регистр и выдача может осуществляться параллельно и последовательно. В первом случае слово представляется в виде параллельного кода. При записи и при считывании все разряды кода слова передаются одновременно, каждый разряд по своей кодовой шине. При последовательной передаче кода слова, все разряды кода слова передаются последовательно во времени один за другим и строго в определенные дискретные моменты времени, совпадающие с управляющими сигналами.

Различают, сдвигающие регистры, в которых возможен сдвиг хранимого кода и регистры без сдвига с приемом информации параллельным кодом. Рассмотрим принцип работы регистров различного назначения.

Регистр с приёмом информации параллельным кодом.

Эти регистры предназначены для приёма, хранения и выдачи информации кода одного  $n$ -разрядного слова, т.е. приём и выдача информации осуществляется в параллельном коде, рассматриваемый  $n$ -разрядный, регистр можно представить как совокупность одноразрядных регистров, имеющих общие шины управления. На рис. 1.1 представлен фрагмент схемы (3 разряда) регистра с приёмом информации параллельным однофазным кодом. Схема данного, регистра имеет выходную логику на элементах Э<sub>1</sub>-Э<sub>3</sub>, которая позволяет считывать информацию из регистра как в прямом, так и обратном коде в зависимости от значений разрешающего сигнала  $P_2$  и  $P_3$



Регистр (рис.) выполнен на D-триггерах (Т<sub>1</sub>-Т<sub>3</sub>). Запись информации осуществляется по входам D.

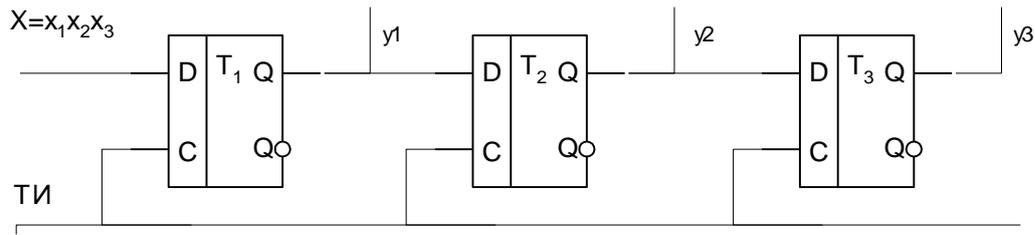
Входное слово  $X=x_1x_2x_3$  поступает на входы D через конъюнкторы Э<sub>4</sub>-Э<sub>6</sub> и при  $P_1=[TI]=1$  записывается в регистр.

Считывание выходного слова  $Y=y_1y_2y_3$  производится через логические элементы Э<sub>1</sub>-Э<sub>3</sub> - с прямых выходов Q<sub>1</sub> Q<sub>2</sub> Q<sub>3</sub> или с инверсных выходов !Q<sub>1</sub>, !Q<sub>2</sub>, !Q<sub>3</sub>. При  $P_2=1, P_3=0, Y=X$  т.е. считывание происходит в прямом коде, а при  $P_2=0, P_3=1 Y=!X$ , т.е. считывание происходит в обратном коде.

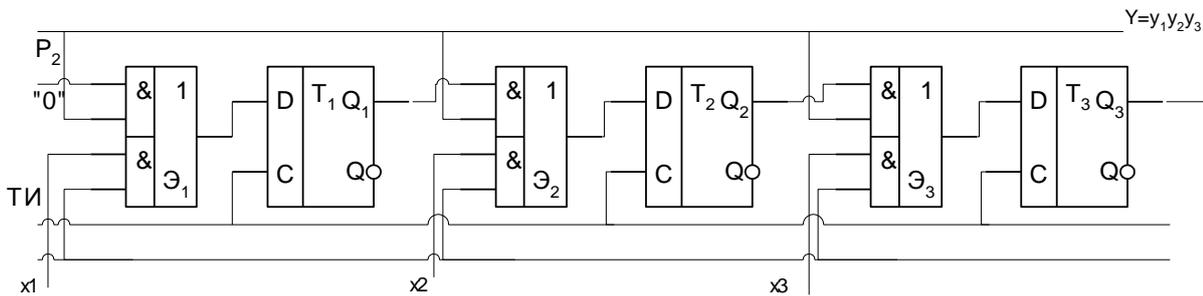
### Сдвигающие регистры.

В регистрах этого типа осуществляется сдвиг слова влево или вправо на заданное число разрядов. За один такт происходит сдвиг на один разряд. Применяются эти регистры в основном для преобразования параллельного кода в последовательный и наоборот, а также в арифметических устройствах при выполнении операций над специальными кодами. Сдвигающие регистры выполняются на D-триггерах.

На рис. 1.2 представлена схема сдвигающего регистра, выполняющего операцию преобразования последовательного кода в параллельный. Число  $X=x_1x_2x_3$  в последовательном коде поступает на вход D-триггера Т<sub>1</sub> старшим разрядом вперёд. При поступлении 1-го ТИ  $x_3$  записывается в Т<sub>1</sub>. По второму ТИ  $x_3$  переписывается в Т<sub>2</sub>, а в Т<sub>1</sub> записывается  $x_2$  и т.д.



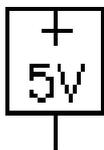
Таким образом, после трёх тактирующих импульсов всё число окажется записанным в регистр и может быть считано в параллельном коде с выходов  $Q_1, Q_2, Q_3$  ( $Y=X$ ) или  $!Q_1, !Q_2, !Q_3$  ( $Y=!X$ ). Процесс считывания организуется точно также, как и в схеме рис. 1



На рисунке изображена схема сдвигающего регистра, выполняющего преобразования параллельного кода в последовательный. Входное слово  $X=x_1x_2x_3$  при  $P_1=[\text{ТИ}]=1$  и  $P_2=0$  записывается в  $T_1T_2T_3$  точно также как и в схеме рис. 1.1. Затем при  $P_2=[\text{ТИ}]=1$  и  $P_1=0$  происходит перезапись информации из  $T_1$  в  $T_2$ , из  $T_2$  в  $T_3$  и т.д. Из младшего разряда в старший по каждому тактирующему импульсу. Таким образом, на входе последнего разряда  $Q_3$  будут последовательно появляться  $x_3, x_2, x_1$ , т.е. входное слово  $X$ , представленное в параллельном коде будет преобразовано в выходное слово  $Y$ , представленное последовательным кодом.

В регистрах на схемах количество триггеров в регистре зависят от разрядности преобразуемого машинного слова. Процесс записи, считывания и преобразования остаётся таким же, как было описано выше.

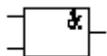
### Используемые элементы программы



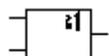
1) Voltage Source. С помощью этого источника на вход триггеров и логических элементов подается логическая единица.



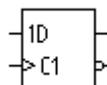
2) Переключатель (Basic->Switch). Переключение производится нажатием на клавишу, указанную в скобках над этим элементом.



3) Логический элемент "И" (Logic gates->2-Input AND gate).



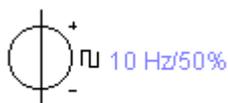
4) Логический элемент "ИЛИ" (Logic gates->2-Input OR gate).



5) Триггер типа D (Digital->D Flip-Flop)



6) Светоиндикатор (Indicators->red Prob). При подаче на этот элемент логической единицы светодиод загорается красным цветом.



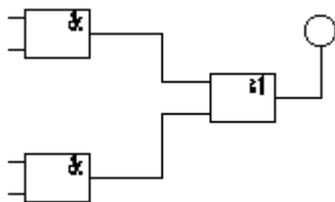
7) Источник прямоугольных сигналов (Sources->Clock). Выполняет функцию синхронизатора, который обеспечивает одновременность подачи сигналов на входы триггеров.

### Порядок выполнения работы:

Составить схему трехразрядного регистра для занесения слова параллельным кодом и возможностью вывода в прямом и обратном кодах и контролем по индикаторам.

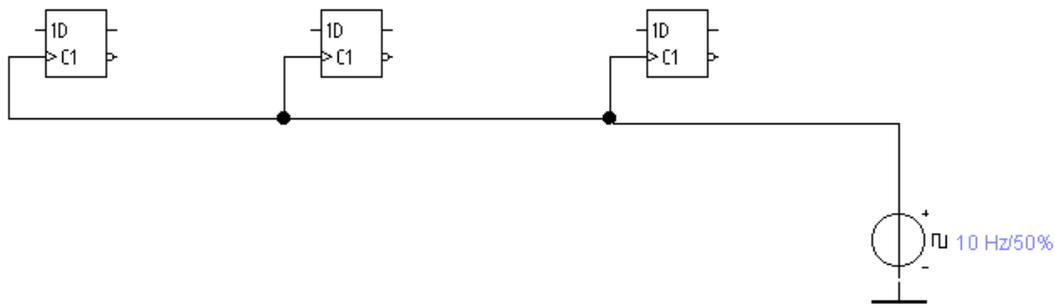
Порядок построения схемы регистра.

1. Сборка переключающих схем для управления считыванием в прямом и обратном кодах:



Для реализации данной схемы необходимо три таких соединения.

2. К входу С1 каждого из трёх D-триггеров подключим источник прямоугольных импульсов с частотой 10 Гц, заземлив его на выходе минусового сигнала:



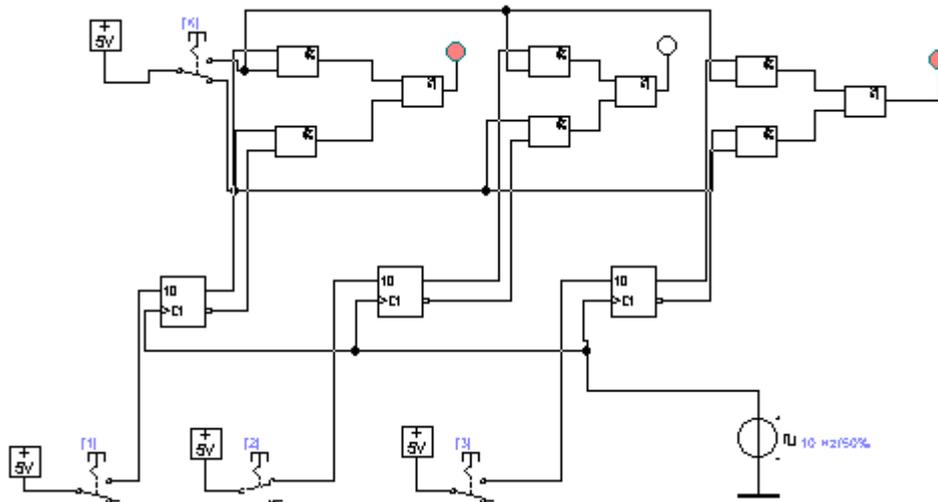
3. Три пятивольтовых источника напряжения подсоединяются через переключатели к входам D1 триггеров каждого разряда.

Для настройки управлением переключателя необходимо назначить клавишу для переключения данного переключателя. Для этого нужно двойным щелчком мыши кликнуть на изображении переключателя и меню настройки в форме KEY указать эту клавишу.

4. Инверсный выход триггера подключить к входу нижнего логического элемента "И". Прodelать эту процедуру с каждым триггером и соответствующим соединением.

5. Пятивольтовый источник напряжения подключить к входу переключателя, выбирающего прямое или инверсное считывание. Верхний вывод переключателя подключить к верхнему входу верхнего логического элемента "И" каждой схемы.

6. Итоговая схема регистра, готовая к проведению проверки работы выглядит следующим образом:



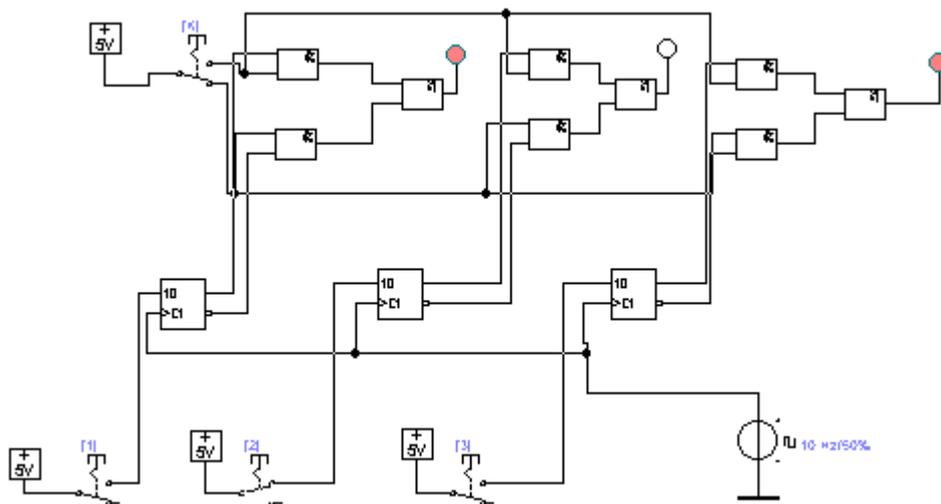
### Порядок проведения исследования работы регистра параллельного занесения

При исходном положении ключа К (верхнее положение) индикация подключается к прямым выходам регистра.

Положение нижних переключателей 1,2,3 определяет код, подаваемый на вход регистра.

Включение 1 и 3 выключателей (нажатие клавиш 1 и 3) соответствует заносимому коду 101.

В результате загорятся соответствующие светоиндикаторы.

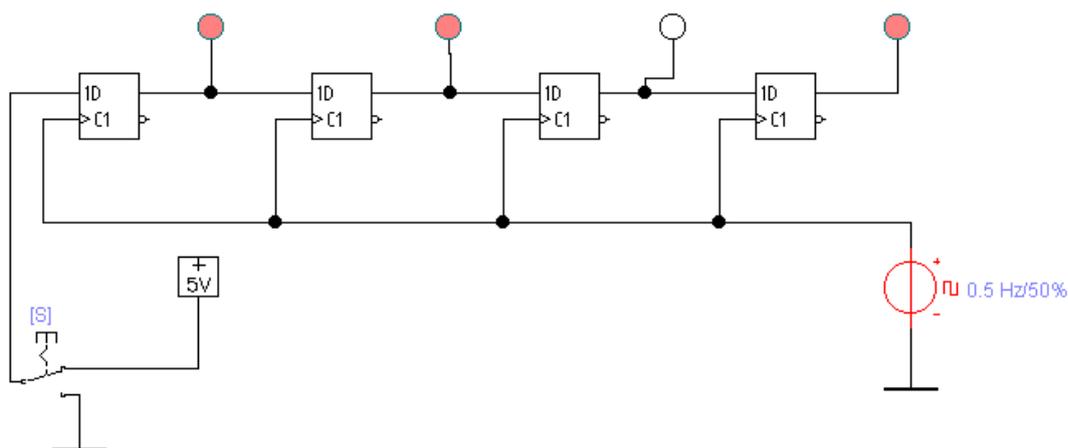


При изменении положения переключателя К на индикаторах отображается инверсный код.

Составить временную диаграмму работы регистра параллельного занесения.

## 2. Составить схему сдвигающего регистра

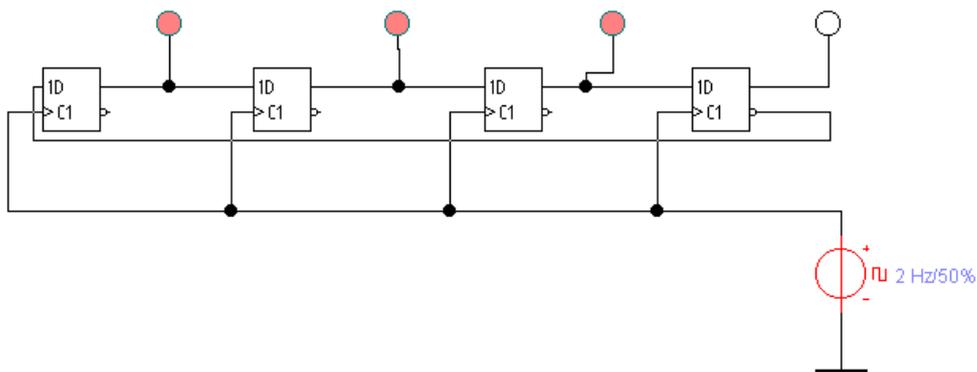
В основе схемы сдвигающего регистра используется та же элементная база.



Изменяя частоту генератора синхроимпульсов в пределах 0.1 до 1 Гц, с помощью переключателя S (клавиша S) можно заносить последовательный код.

## Моделирование работы счетчика Джонсона

Для получения схемы счетчика Джонсона необходимо соединить инверсный выход старшего разряда регистра с входом младшего. Изменяя частоту синхроимпульсов, можно добиться разной скорости изменения состояний.



### Задание для самостоятельной работы

Собрать схему реверсивного сдвигающего регистра. Организовать управление направлением сдвига с помощью переключателя и обеспечить индикацию выполнения операций сдвига.

#### Контрольные вопросы

1. Назначение различных типов регистров в вычислительных устройствах.
2. Какие типы триггеров могут быть использованы для построения регистров?
3. Назначение синхросигнала в работе регистров.
4. Какие дополнительные устройства необходимы для построения универсального регистра.
5. Последовательность действий, выполняемых на регистрах с целью преобразования параллельного кода в последовательный код и обратно.

#### Форма представления результата:

1. Схемы, моделирующие работу регистров по всем пунктам задания.
2. Временные диаграммы, поясняющие работу регистров.

#### Критерии оценки:

- Оценка «отлично» ставится, если задание выполнено верно.
- Оценка «хорошо» ставится, если ход выполнения задания верный, но была допущена одна или две ошибки, приведшие к неправильному результату.
- Оценка «удовлетворительно» ставится, если приведено неполное выполнение задания.
- Оценка «неудовлетворительно» ставится, если задание не выполнено.

## Тема 2.1 Логические основы ЭВМ, элементы и узлы

### Лабораторное занятие № 4 Моделирование работы счётчиков

**Цель:** Ознакомление с возможностями моделирования работы схем счетчиков.

**Выполнив работу, Вы будете:**

**уметь:**

- строить схемы счетчиков;
- использовать современное программное обеспечение при решении профессиональных задач.

**Материальное обеспечение:**

Методические указания для выполнения лабораторных работ, компьютер, программное обеспечение: Multisim Education.

**Задание:**

Составить схему асинхронного суммирующего счётчика на D-триггерах.

**Краткие теоретические сведения:**

#### Основные понятия

**Счётчик** - это электронное устройство для подсчета количества импульсов. По мере поступления входных сигналов счётчик последовательно изменяет свои состояния в определённом для данного типа счётчика порядке.

Число используемых состояний счётчика называют *модулем пересчёта* или *основанием счёта*.

По типу выполняемых операций счётчики подразделяются на *суммирующие*, *вычитающие* и *реверсивные*.

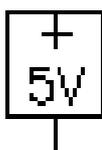
Счётчики, для работы которых достаточно одного счётного сигнала, называют *асинхронными* счётчиками. Счётчики, которые для переключения в следующее состояние используют помимо счётного сигнала ещё и сигнал синхронизации, называются *синхронными* счётчиками.

В большинстве схем счётчиков предусматривается общая шина сброса (обнуления) счётчика.

В некоторых схемах счётчиков может быть предусмотрена возможность загрузки значения начала счёта, отличного от нулевого.

В основе схем счётчиков в большинстве случаев находят применение счетные триггеры.

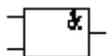
#### Используемые элементы программы



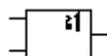
1) Voltage Source. С помощью этого источника на вход триггеров и логических элементов подается логическая единица.



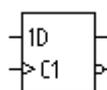
2) Переключатель (Basic->Switch). Переключение производится нажатием на клавишу, указанную в скобках над этим элементом.



3) Логический элемент "И" (Logic gates->2-Input AND gate).



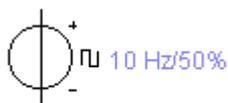
4) Логический элемент "ИЛИ" (Logic gates->2-Input OR gate).



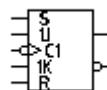
5) Триггер типа D (Digital->D Flip-Flop)



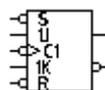
6) Светоиндикатор (Indicators->red Prob). При подаче на этот элемент логической единицы светодиод загорается красным цветом.



7) Источник прямоугольных сигналов (Sources->Clock). Выполняет функцию синхронизатора, который обеспечивает одновременность подачи сигналов на входы триггеров.



8) Универсальный J-K триггер.



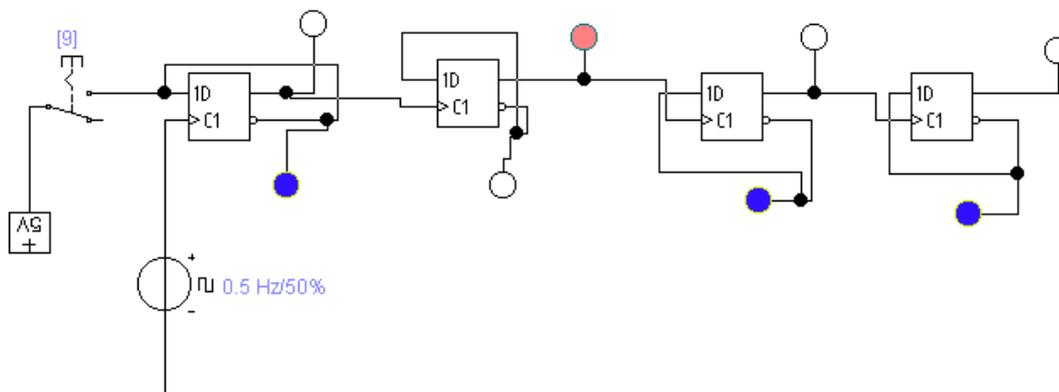
9) Универсальный J-K триггер с установочными входами.



10) Семисегментный цифровой индикатор.

### Порядок выполнения работы:

Составить схему асинхронного суммирующего счётчика на D-триггерах.



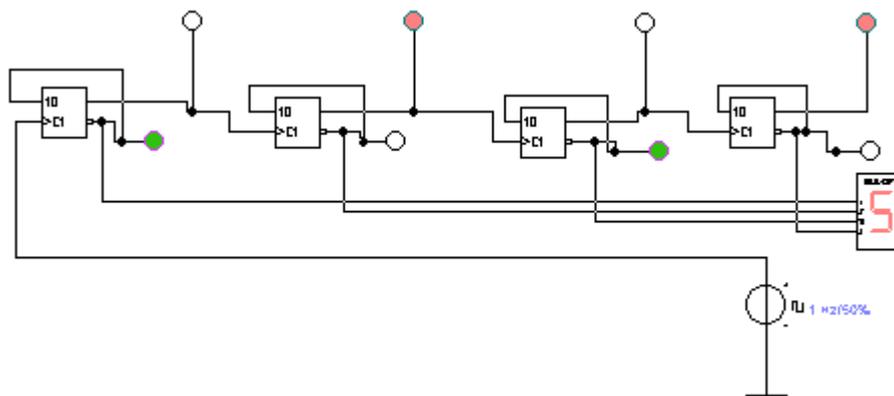
Пятивольтовый источник напряжения подсоединяется через переключатель к входу D1 триггера младшего (левого) разряда счетчика.

Для настройки управлением переключателя необходимо назначить клавишу для переключения данного переключателя. Для этого нужно двойным щелчком мыши кликнуть на изображении переключателя и меню настройки в форме KEY указать эту клавишу.

Инверсный выход триггера подключить к D-входу триггера, чтобы обеспечить работу в счётном режиме. Такое соединение провести на всех триггерах всех разрядов.

К прямым выходам разрядов подключить семисегментный индикатор с дешифратором.

Итоговая схема счетчика может быть такой:



### Контрольные вопросы

1. Назначение счётчиков в вычислительных устройствах.
2. Признаки классификации счётчиков.
3. Коэффициент пересчёта счётчика.
4. Чем определяется максимальная частота поступления входных сигналов на счётчик?

### Форма представления результата:

1. Схемы, моделирующие работу счётчиков по всем пунктам задания.
2. Временные диаграммы, поясняющие работу счётчиков.

**Критерии оценки:**

Оценка «отлично» ставится, если задание выполнено верно.

Оценка «хорошо» ставится, если ход выполнения задания верный, но была допущена одна или две ошибки, приведшие к неправильному результату.

Оценка «удовлетворительно» ставится, если приведено неполное выполнение задания.

Оценка «неудовлетворительно» ставится, если задание не выполнено.

## Тема 2.4 Технологии повышения производительности процессоров

### Лабораторное занятие № 5

#### Системы команд процессора. Модель учебной ЭВМ.

**Цель:** знакомство с интерфейсом модели ЭВМ, методами ввода и отладки программы, действиями основных классов команд и способов адресации.

#### Выполнив работу, Вы будете:

##### уметь:

- выполнять команды в машинных кодах;
- работать в команде;
- использовать современное программное обеспечение при решении профессиональных задач.

#### Материальное обеспечение:

Методические указания для выполнения лабораторных работ, компьютер, программное обеспечение: Multisim Education.

#### Краткие теоретические сведения:

##### Введение

Для выполнения лабораторных работ будет использована программная модель ЭВМ, которая была разработана Жмакиным А.П. Эта модель позволяет реализовать доступ к различным элементам ЭВМ.

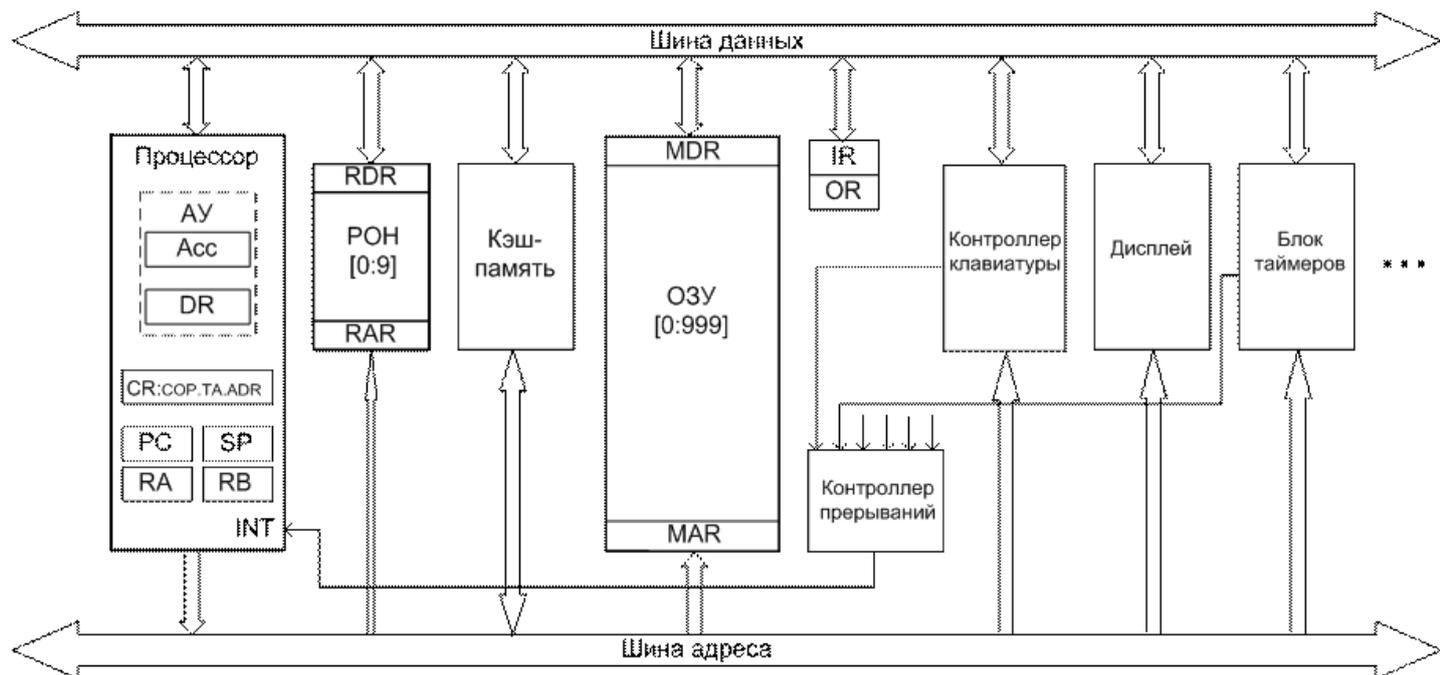


Рис.1 Общая структура учебной ЭВМ.

Модель ЭВМ включает процессор, оперативную память (ОЗУ) и сверхоперативную память (регистры общего назначения (РОН) и кэш-память), устройство ввода и устройство вывода. Процессор в свою очередь состоит из устройства управления (УУ), арифметического устройства (АУ) и системных регистров (CR, PC, SP и др.). Структурная схема ЭВМ показана на рисунке 1.

В ячейках ОЗУ хранятся команды и данные. Емкость ОЗУ составляет 1000 ячеек. По сигналу Wt выполняется запись содержимого регистра данных (MDR) в ячейку памяти с адресом, указанным в регистре адреса (MAR). По сигналу Rd происходит считывание - содержимое ячейки памяти с адресом, содержащимся в MAR, передается в MDR.

Сверхоперативная память с прямой адресацией содержит десять регистров общего назначения R0-R9. Доступ к ним осуществляется (аналогично доступу к ОЗУ) через регистры RAR и RDR.

АУ осуществляет выполнение одной из арифметических операций, определяемой кодом операции (COP), над содержимым аккумулятора (Acc) и регистра операнда (DR). Результат операции всегда помещается в Acc. При завершении выполнения операции АУ вырабатывает сигналы признаков результата: Z (равен 1, если результат равен нулю); S (равен 1, если результат отрицателен); OV (равен 1, если при выполнении операции произошло переполнение разрядной сетки). В случаях, когда эти условия не выполняются, соответствующие сигналы имеют нулевое значение.

В модели ЭВМ предусмотрены внешние устройства двух типов. Во-первых, это регистры IR и OR, которые могут обмениваться с аккумулятором с помощью безадресных команд in (Acc := IR) и out (OR := Acc). Во-вторых, это набор моделей внешних устройств, которые могут подключаться к системе и взаимодействовать с ней в соответствии с заложенными в моделях алгоритмами. Каждое внешнее устройство имеет ряд программно-доступных регистров, может иметь собственный обозреватель (окно видимых элементов).

УУ осуществляет выборку команд из ОЗУ в последовательности, определяемой естественным порядком выполнения команд (т. е. в порядке возрастания адресов команд в ОЗУ) или командами передачи управления; выборку из ОЗУ операндов, задаваемых адресами команды; инициирование выполнения операции, предписанной командой; останов или переход к выполнению следующей команды.

В качестве сверхоперативной памяти в модель включены регистры общего назначения (РОН), и может подключаться модель кэш-памяти. В состав УУ ЭВМ входят:

1. PC-счетчик адреса команды, содержащий адрес текущей команды;
2. CR-регистр команды, содержащий код команды;
3. RB-регистр базового адреса, содержащий базовый адрес;
4. SP-указатель стека, содержащий адрес верхушки стека;
5. RA-регистр адреса, содержащий исполнительный адрес при косвенной адресации.

Регистры Acc, DR, IR, OR, CR и все ячейки ОЗУ и РОН имеют длину 6 десятичных разрядов, регистры PC, SP, RA и RB - 3 разряда.

Данные в ЭВМ представляются в формате, показанном на рисунке 2. Это целые десятичные числа, изменяющиеся в диапазоне "-99 999...+99 999", содержащие знак и 5 десятичных цифр.



Рис. 2

Старший разряд слова данных используется для кодирования знака: плюс (+) изображается как 0, минус (-) как 1. Если результат арифметической операции выходит за пределы указанного диапазона, то говорят, что произошло переполнение разрядной сетки. АЛУ в этом случае вырабатывает сигнал переполнения  $OV = 1$ . Результатом операции деления является целая часть частного. Деление на ноль вызывает переполнение.

## Система команд

– совокупность всех команд, которые способна выполнить данная ЭВМ. Система команд характеризуется тремя аспектами: форматами, способами адресации и системой операций.

Таблица 1. Система команд.

№п/п	Мнемокод	Название
	NOP	Пустая операция
	IN	Ввод
	OUT	Вывод
	RET	Возврат из подпрограммы
	HLT	Стоп
	JMP	Безусловный переход
	JZ	Переход, если 0.
	JNZ	Переход, если не 0
	JS	Переход, если отрицательно
	JNS	Переход, если положительно
	JO	Переход, если переполнение
	JNO	Переход, если нет переполнения
	CALL	Вызов п/п
	RD	Чтение (в аккумулятор)
	WR	Запись (из аккумулятора)
	ADD	Сложение
	SUB	Вычитание
	MUL	Умножение
	DIV	Деление
	ADI	Сложение с данными
	SBI	Вычитание с данными
	MULI	Умножение с данными
	DIVI	Деление с данными
	MOV	Пересылка R1← R2

### 1. Форматы команд.

Команда определяет операцию, которую выполняет ЭВМ над данными. Команда содержит в явной или неявной форме информацию о том, где будет помещен результат операции, а также об адресе следующей команды. Код команды состоит из нескольких частей, которые называются полями.

Длина команды, количество, размер, положение, назначение и способ кодировки ее полей называется форматом команды.

В общем случае формат команды содержит операционную и адресную части. Операционная часть содержит код операции (например, сложение, умножение, передача

данных). Адресная часть состоит из нескольких полей и содержит информацию об адресах операндов, результате операции и адресе следующей команды. Адресная часть, в свою очередь, может состоять из двух полей (типа адресации и адреса операнда). Такой формат команды представлен на рис. 3.

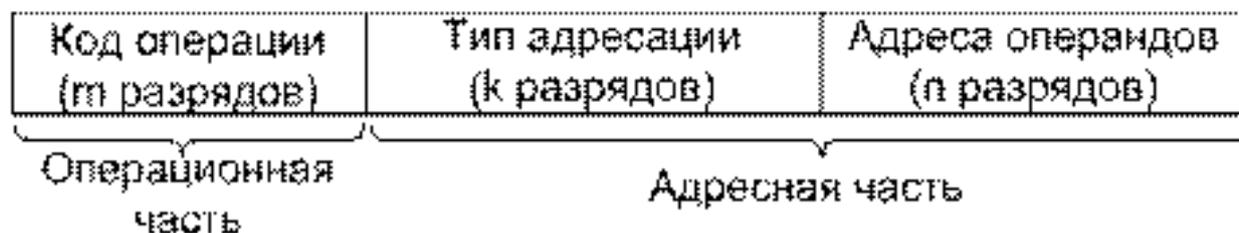


Рис. 3. Формат команды.

Поле «тип адресации» определяет способ адресации операнда. Биты полей «тип адресации» и «адреса операндов» в совокупности определяют ячейки памяти, в которых хранятся операнды.

В современных ЭВМ выделяют следующие форматы команд: трехадресный, двухадресный, одноадресный и безадресный. Эти форматы показаны на рис. 4, в котором приняты следующие обозначения:

КОП – код операции,  $A_i$  – адрес операнда/результата.

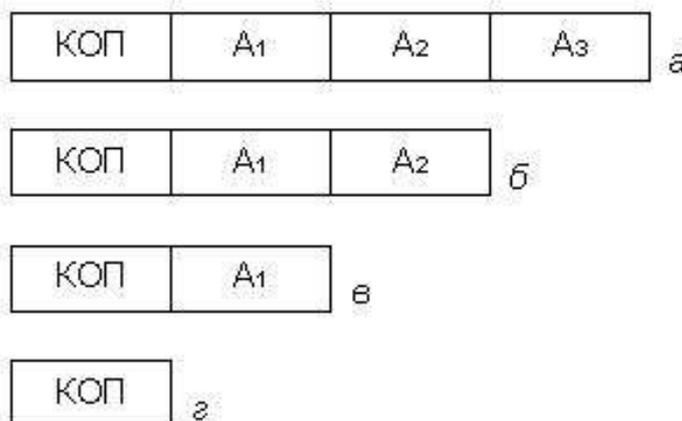


Рис. 4. Форматы команд ЭВМ:

а – трехадресный; б – двухадресный; в – одноадресный; г – безадресный

Для реализации одноадресных форматов (рис. 4, в) в процессоре предусмотрен специальный регистр – аккумулятор. Первый операнд и результат всегда размещаются в аккумуляторе, а второй операнд адресуется полем A.

В аккумуляторе большинство команд – одноадресные и безадресные, длиной в одно машинное слово (6 разрядов). Исключением являются двухсловные команды с непосредственной адресацией и команда MOV, являющаяся двухадресной. Форматы команд учебной ЭВМ показаны на рис. 5.

В форматах команд выделены три поля:

- 1) разряды [0:1] определяют код операции COP и являются старшими;
- 2) разряд 2 может определять тип адресации (в случае формата 5а он определяет номер регистра);

3) разряды [3:5] могут определять прямой или косвенный адрес памяти, номер регистра (в команде MOV номера двух регистров), адрес перехода или короткий непосредственный операнд.

В двухсловных командах непосредственный операнд занимает разряды [6:11]

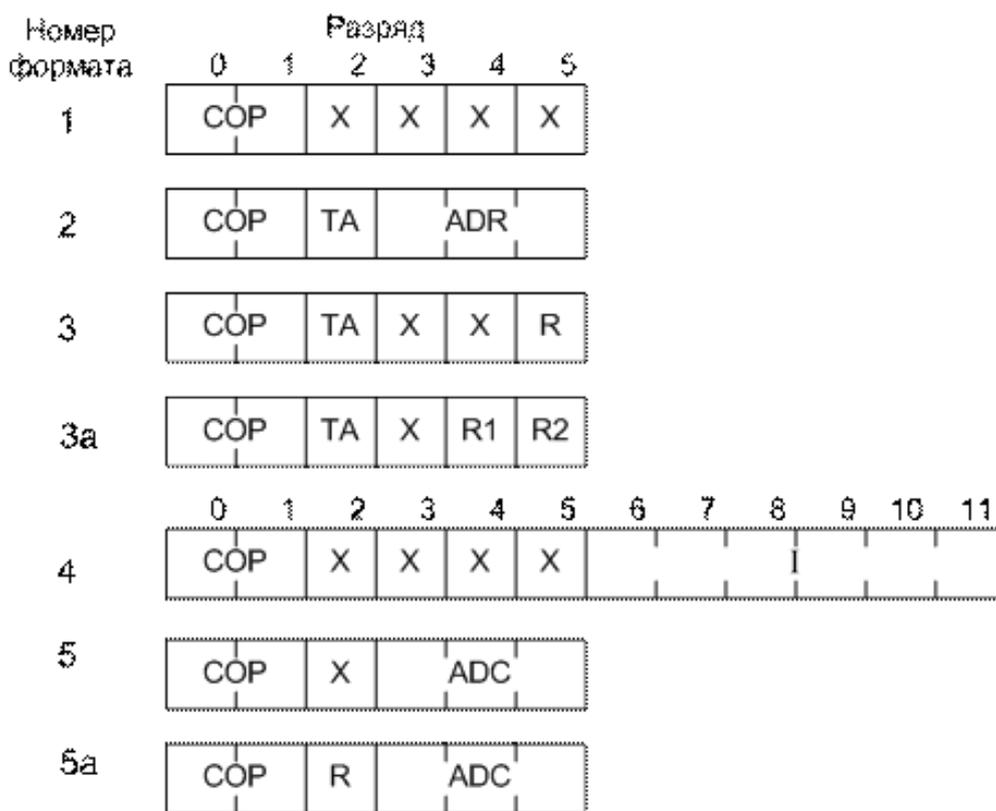


Рис. 5. Форматы команд учебной ЭВМ. COP – код операции; ADR – адрес операнда в памяти;

ADC – адрес перехода; I – непосредственный операнд; R, R1, R2 – номер регистра; TA – тип адресации; X – разряд не используется.

### **Способы адресации**

Способ адресации определяет, каким образом следует использовать информацию, размещенную в поле адреса команды.

В ЭВМ различают пять основных способов адресации: прямая, косвенная, непосредственная, относительная и безадресная. Каждый из перечисленных способов имеет свои разновидности. В модели учебной ЭВМ реализованы семь способов адресации, которые приведены в табл. 2.

Код ТА	Тип адресации	Исполнительный адрес
0	Прямая (регистровая)	ADR (R)
1	Непосредственная	-
2	Косвенная	ОЗУ(ADR)[3:5]
3	Относительная	ADR + RB
4	Косвенно-регистровая	РОН(R)[3:5]
5	Индексная с постинкрементом	РОН(R)[3:5], R:=R + 1
6	Индексная с предскрементом	R:=R-1, РОН(R)[3:5]

**Прямая адресация** – исполнительный адрес размещается в адресном поле команды. В этом способе непосредственно используется счетчик команд и аккумулятор. При прямой адресации содержимое счетчика команд передается на шину адреса и из адресуемой ячейки выбирается очередная команда, поступающая в устройство управления.

**Непосредственная адресация**- данные включены в команду и собственно обращения к памяти при этом способе загрузки регистров не требуется. Команда вместо указания адреса операнда содержит этот операнд непосредственно. Непосредственная адресация используется при операциях с константами, исполнение команд с непосредственной адресацией отличается быстротой.

**Косвенная адресация**– адресное поле команды указывает не адрес операнда, а лишь адрес ячейки, в которой хранится косвенный адрес, по которому можно получить операнд.

**Косвенно-регистровая**- в этом случае команда предписывает выполнить операцию над операндом, адрес которого содержится в регистрах общего назначения.

Таблица 3. Типы адресации

Обозначение	Код	Тип адресации	Пример
		Прямая (регистровая)	ADD 23 (ADD R3)
#		Непосредственная	ADD #33
@		Косвенная	ADD @33
@R		Косвенно - регистровая	ADD @R3

### **Система операций**

Системой операций называется список операций, непосредственно выполняемый техническими средствами ЭВМ. Система операций ЭВМ определяется областью ее применения, требованиями к стоимости, производительности и точности вычислений. Все операции, выполняемые в командах ЭВМ, принято делить на пять классов. Основные операции всех пяти классов представлены в эмуляторе ЭВМ.

Система команд учебной ЭВМ включает команды следующих классов:

1. арифметико-логические и специальные: сложение, вычитание, умножение, деление;
2. пересылки и загрузки: чтение, запись, пересылка (из регистра в регистр), помещение в стек, извлечение из стека, загрузка указателя стека, загрузка базового регистра;
3. ввода/вывода: ввод, вывод;
4. передачи управления: безусловный и шесть условных переходов, вызов подпрограммы, возврат из подпрограммы, цикл, программное прерывание, возврат из прерывания;
5. системные: пустая операция, разрешить прерывание, запретить прерывание, стоп.

## Состояния и режимы работы учебной ЭВМ

Ядром устройства управления ЭВМ является управляющий автомат (УА). Это устройство предназначено для выработки сигналов управления, которые инициируют работу АЛУ, РОН, ОЗУ и УВВ, передачу информации между регистрами устройств ЭВМ и действия над содержимым регистров УУ.

Учебная ЭВМ может находиться в двух состояниях Останов и Работа. В состоянии Работа модель переходит по действию команд **Пуск** или **Шаг**. Команда **Пуск** запускает программу на выполнение. Программа представляет собой последовательность команд, записанных в ОЗУ. Программа выполняется в автоматическом режиме до команды **HLT** (Стоп) или **точки останова**. Программа выполняется по командам, начиная с ячейки ОЗУ, на которую указывает счетчик команд (**РС**), причем изменение состояний объектов модели отображается в окнах компонентов.

В состоянии **Останов** модель учебной ЭВМ переходит в результате действия команды **СТОП** или автоматически в зависимости от выбранного режима работы.

Команда **Шаг** запускает выполнение одной команды либо одной микрокоманды (если установлен **Режим микрокоманд**) после чего модель переходит в состояние **Останов**.

В состоянии **Останов** пользователь может просмотреть или изменить основные компоненты модели: регистры ЦП и РОН, ячейки ОЗУ, устройства ввода/вывода. В процессе изменения содержимого ОЗУ и РОН можно вводить данные для программы, в ячейки ОЗУ – программу в кодах. В данном состоянии пользователю доступны для изменения параметры модели и режимы ее работы, ввод и/или редактирование программы в мнемосокодах, ассемблирование мнемосокодов, выполнение основных операций с файлами.

### Интерфейс пользователя и основные компоненты учебной модели ЭВМ.

#### 1. Окно Модель учебной ЭВМ

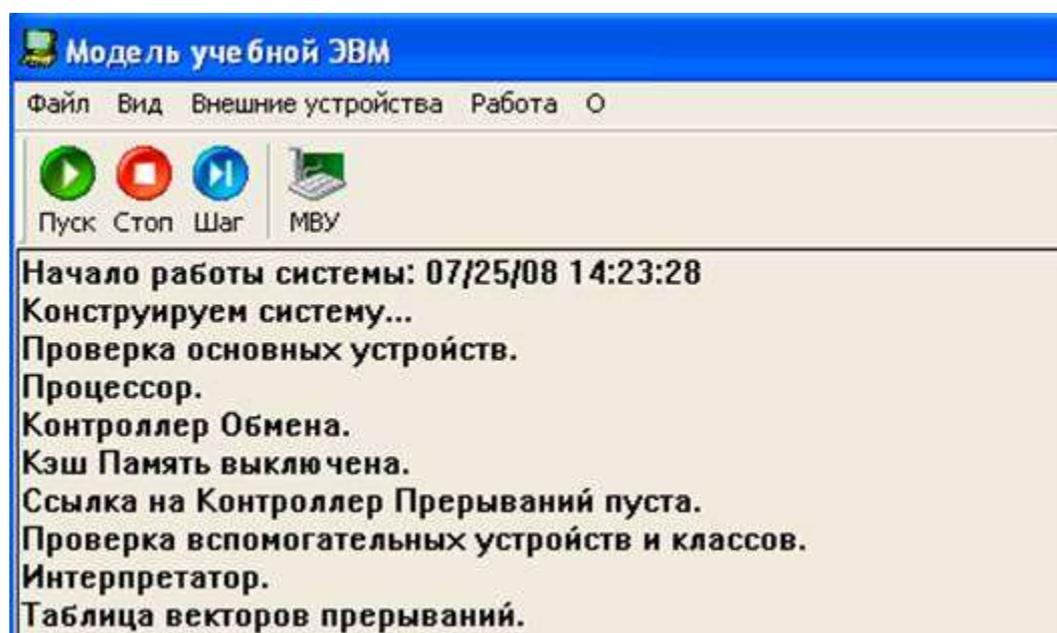


Рис. 6. Основное окно эмулятора.

В рабочее поле окна выводятся сообщения о функционировании системы в целом. Эти сообщения группируются в файл logfile.txt (по умолчанию), автоматически сохраняющемся на диске, и могут быть проанализированы после завершения работы с моделью.

Меню содержит следующие пункты: **Файл, Вид, Внешние устройства, Работа.**

Пункт меню **Вид** содержит следующие команды: **Показать все, Скрыть все, Процессор, Микрокомандный уровень, Память, Кэш-память, Программа, Текст программы.** Эти команды открывают окна соответствующих компонентов, описанных далее.

Пункт меню **Файл** содержит команду **Выход**, позволяющую закончить работу с моделью.

Пункт меню **Работа** содержит команды: **Пуск, Стоп, Шаг, Режим микрокоманд, Кэш-память, Настройки.** Команда **Пуск** позволяет запустить программу в автоматическом режиме, команда **Шаг** запускает программу в шаговом режиме, а команда **Стоп** позволяет остановить выполнение программы в модели процессора. Эти команды могут выполняться путем нажатия одноименных клавиш на панели управления.

Команда **Режим микрокоманд** включает или выключает микрокомандный режим работы процессора, а команда **Кэш-память** позволяет подключить или отключить модель кэш-памяти.

Команда **Настройки** открывает диалоговое окно **Параметры системы**, позволяющие установить задержку реализации командного цикла (если команда выполняется в автоматическом режиме), а также установить параметры файла logfile.txt.

## 2. Компонент Процессор.

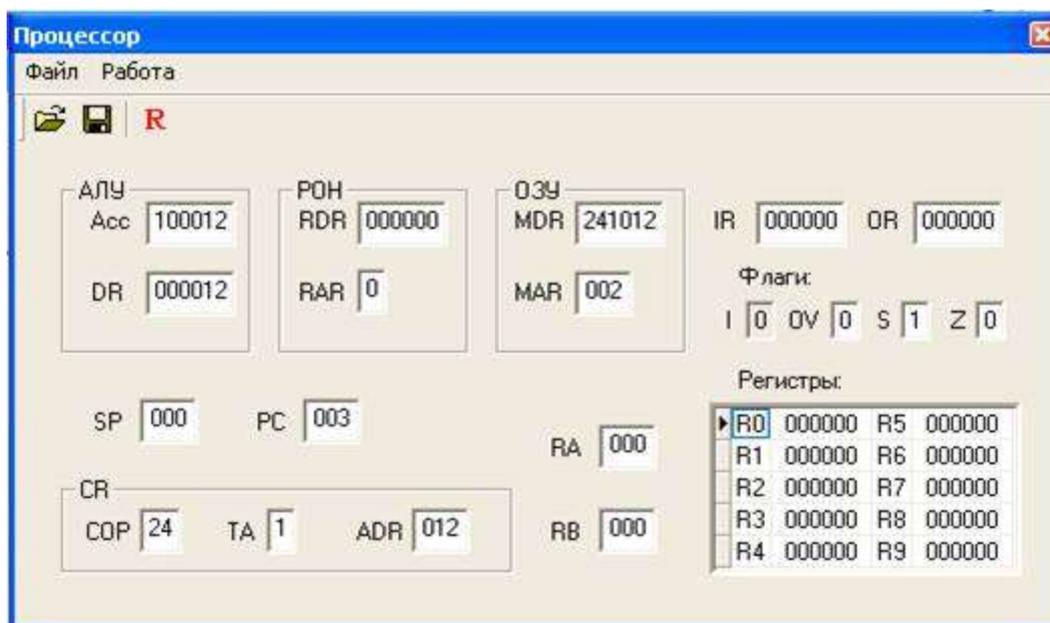


Рис. 7. Окно компонента **Процессор**

Окно **Процессор** обеспечивает доступ ко всем регистрам и флагам процессора.

Программно-доступные регистры и флаги:

- Асс – аккумулятор;
- РС – счетчик адреса текущей команды;
- SP – указатель стека;
- RB – регистр базового адреса, содержащий адрес базы при относительной адресации;
- RA – регистр адреса, содержащий исполнительный адрес при косвенной адресации;
- IR – входной регистр;

- OR – выходной регистр;
- I - флаг разрешения прерываний.

Системные регистры и флаги:

- DR – регистр данных АЛУ, содержащий второй операнд;
- MDR – регистр данных ОЗУ;
- MAR – регистр адреса ОЗУ;
- RDR – регистр данных блока РОН;
- RAR – регистр адреса блока РОН;
- CR – регистр команд, содержащий поля:
  - COP – код операции;
  - TA - тип адресации
  - ADR – адрес или непосредственный операнд.
- Z – флаг нулевого значения Асс;
- S – флаг отрицательного значения Асс;
- OV – флаг переполнения

Регистры Асс, DR, CR, IR, OR и все ячейки ОЗУ и РОН имеют длину шесть десятичных разрядов, а длина SP, PC, RA и RB три десятичных разряда. Компонент **Процессор** отражает текущие значения регистров и флагов ЦП. В состоянии **Останов** все регистры, включая регистры блока РОН, и флаги, кроме флага I, доступны для редактирования.

Меню компонента **Процессор** содержит пункты **Файл** и **Работа**. Пункт меню **Файл** содержит команды **Сохранить** и **Загрузить**, которые позволяют сохранить текущие значения регистров и флагов процессора в файле и восстановить состояние ЦП из файла.

Команда **Reset**, находящаяся в пункте меню **Работа** позволяет установить все регистры (в том числе блок РОН) в нулевое значение. Пункт меню **Работа** содержит также команду **Reset R0 – R9**, при выполнении которой очищаются только регистры блока РОН.

### 3. Компонент Память

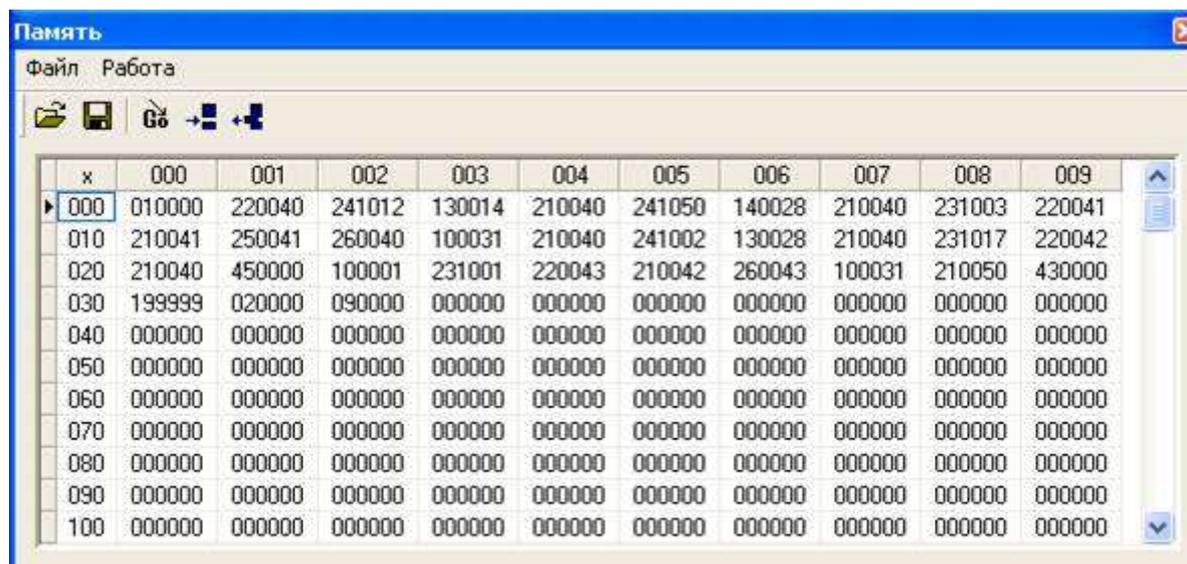


Рис. 8. Окно компонента **Память**

Окно компонента **Память** (рис. 8) отображает текущее состояние ячеек ОЗУ. Этот компонент допускает возможность редактирования содержимого ячеек и выполнения пяти команд: **Сохранить**, **Загрузить**, **Перейти к**, **Вставить** и **Убрать**.

#### 4. Компонент Текст программы

Окно компонента Текст программы (рис. 9) содержит поле текстового редактора, в котором можно редактировать текст, загружать в него текстовые файлы и сохранять текст в виде файлов.

Меню данного окна содержит стандартные для всех компонентов модели пункты **Файл** и **Работа**. В пункте меню **Файл** расположены следующие команды: **Новая**, **Загрузить**, **Сохранить**, **Сохранить как** и **Вставить**. Данные команды позволяют выполнять следующие действия:

**Новая** – открыть новый сеанс редактирования;

**Загрузить** – открыть диалоговое окно загрузки файла в окно редактора;

**Сохранить** – сохранить файл под текущим именем;

**Сохранить как** – открыть диалоговое окно сохранения файла.

Эти команды дублированы кнопками на панели управления.

Команда **Вставить** позволяет вставить выбранный файл в позицию курсора.

На панели управления присутствует также кнопка **Компилировать**, которая запускает процедуру ассемблирования текста в поле редактора. Эту же команду можно запустить в пункте меню **Работа**, который содержит также команду **Адрес вставки**. Команда **Адрес вставки** позволяет задать адрес ячейки ОЗУ, начиная с которой команда будет записана в память. По умолчанию этот адрес равен 0.

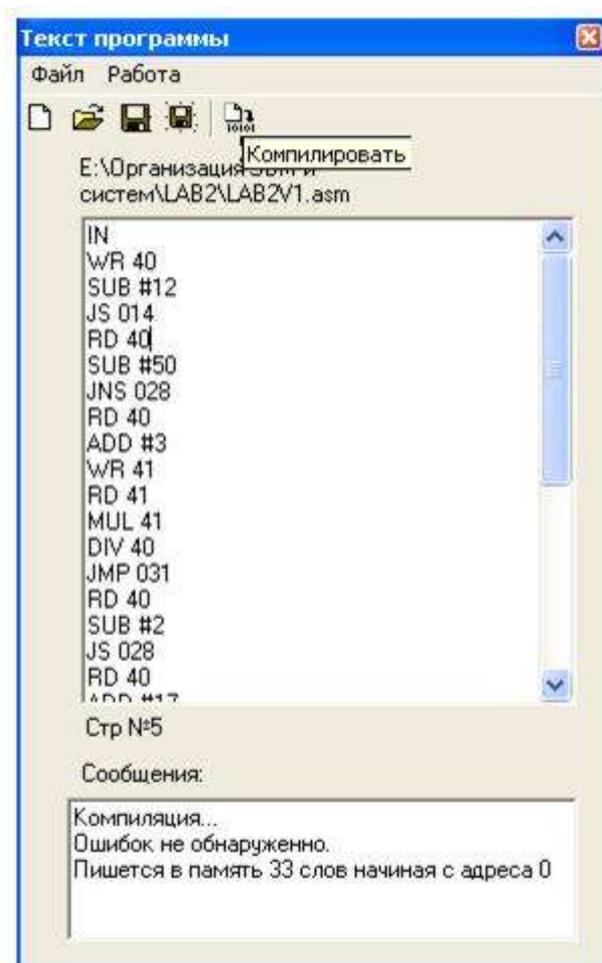


Рис. 9. Компонент Текст программы

Ниже области редактирования находится строка состояния, в которую выводится номер строки, определяющий положение курсора.

Если в процессе компиляции программы обнаруживаются синтаксические ошибки, то сообщения о них выводятся в окно сообщений и запись в память кода программы не производится.

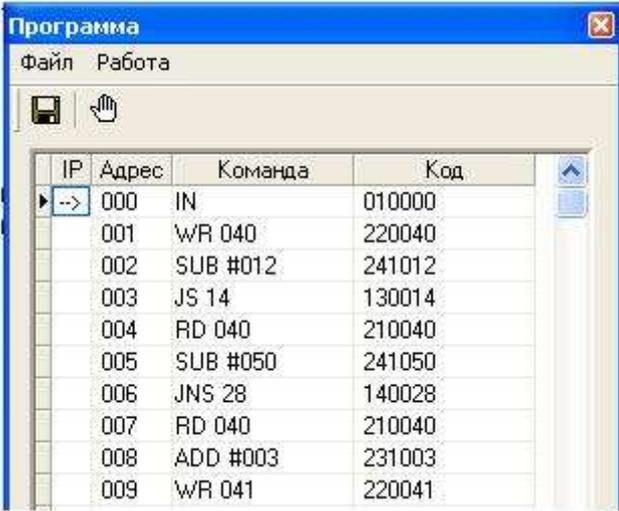
После исправления ошибок и повторной компиляции выдается сообщение об отсутствии ошибок, расположении программы в области памяти и количестве слов, занятых под ассемблированную программу.

Набор текста программы производится по стандартным правилам языка ассемблера. В каждой строке может содержаться метка, одна команда и комментарий. Метка отделяется от команды двоеточием, символы после знака «;» до конца строки игнорируются компилятором и рассматриваются как комментарии. Строка может начинаться с «;» и, следовательно, содержать только комментарии.

**Внимание!** Откомпилированная программа размещается, начиная с адреса 000. Данные надо располагать в памяти так, чтобы они не пересекались с кодом программы.

## 5. Компонент Программа

Окно компонента **Программа** состоит из трех составляющих: стандартного для эмулятора меню, панели управления и таблицы, которая имеет 300 строк и 4 столбца (рис. 1.10). Каждая строка соответствует дизассемблированной ячейке памяти. Второй столбец содержит адрес ячейки ОЗУ, третий – дизассемблированный мнемокод, четвертый – машинный код команды. В первом столбце помещается указатель «→» на текущую команду (текущее значение счетчика команд) и точка останова – красная заливка ячейки.



The screenshot shows a window titled 'Программа' with a menu bar containing 'Файл' and 'Работа'. Below the menu bar is a toolbar with a disk icon and a hand icon. The main area contains a table with the following data:

IP	Адрес	Команда	Код
→	000	IN	010000
	001	WR 040	220040
	002	SUB #012	241012
	003	JS 14	130014
	004	RD 040	210040
	005	SUB #050	241050
	006	JNS 28	140028
	007	RD 040	210040
	008	ADD #003	231003
	009	WR 041	220041

Рис. 10. Окно Программа

Окно компонента **Программа** позволяет наблюдать процесс прохождения программы. Содержимое этого окна нельзя редактировать.

Используя пункты меню, можно сохранить содержимое окна в виде текстового файла, выбрать начальный адрес области ОЗУ, которая будет дизассемблироваться (размер области постоянный – 300 ячеек), а также установить или снять точку останова. Установить точку останова можно тремя способами: командой **Точка останова** из пункта меню **Работа**, одноименной кнопкой на панели управления или двойным щелчком кнопки мыши в первой

ячейке соответствующей строки. Прочитать в окне компонента **Программа** ничего нельзя. Сохраненный ранее файл с расширением .asm можно загрузить в окно компонента **Текст программы**, ассемблировать его и тогда дизассемблированное значение заданной области памяти автоматически появится в окне компонента **Программа**. Такую процедуру удобно использовать, если программа изначально пишется или редактируется в памяти в машинных кодах.

### 6. Компонент Микрокомандный уровень

Компонент Микрокомандный уровень (рис. 11) используется только в режиме микрокоманд, который устанавливается командой Режим микрокоманд в пункте меню Работа основного окна Модель учебной ЭВМ. В окно Микрокомандный уровень выводится мнемокод выполняемой команды, список микрокоманд, ее реализующих и указатель на текущую выполняемую команду.

Чтобы просмотреть процесс выполнения программы на уровне микрокоманд, необходимо чтобы модель ЭВМ работала либо в шаговом режиме, либо в автоматическом режиме с задержкой командного цикла.

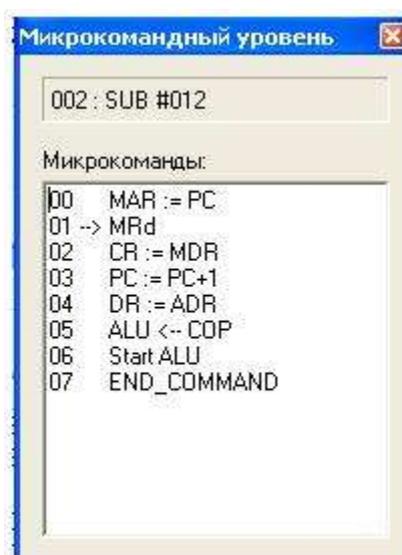


Рис. 11. Окно Микрокомандный уровень

Если открыть окно **Микрокомандный уровень**, не установив режим микрокоманд в меню **Работа**, то после начала выполнения программы в режиме **Шаг** (или в автоматическом режиме) в строке сообщений окна будет выдано сообщение «**Режим микрокоманд неактивен**».

#### Порядок выполнения работы:

**Задание 1.** Выполнить программу из Табл.4

Таблица 4.

Адрес	Команда	Комментарий
	rd 070	Загрузка в аккумулятор первого числа.
	wr R1	Запись первого числа в регистр R1
	rd 071	Загрузка в аккумулятор второго числа

	add r1	Сложение чисел, результат в аккумуляторе
	wr 072	Запись результата в ОЗУ
	hlt	Стоп

Выполнение программы осуществляется нажатием на кнопку **Пуск**. На рис. 12 представлены результаты выполнения программы сложения чисел 5 и 7, которые располагаются в ячейках памяти с адресами 070 и 071. Результат выполнения программы, число 12 записывается в ячейку ОП с адресом 072.

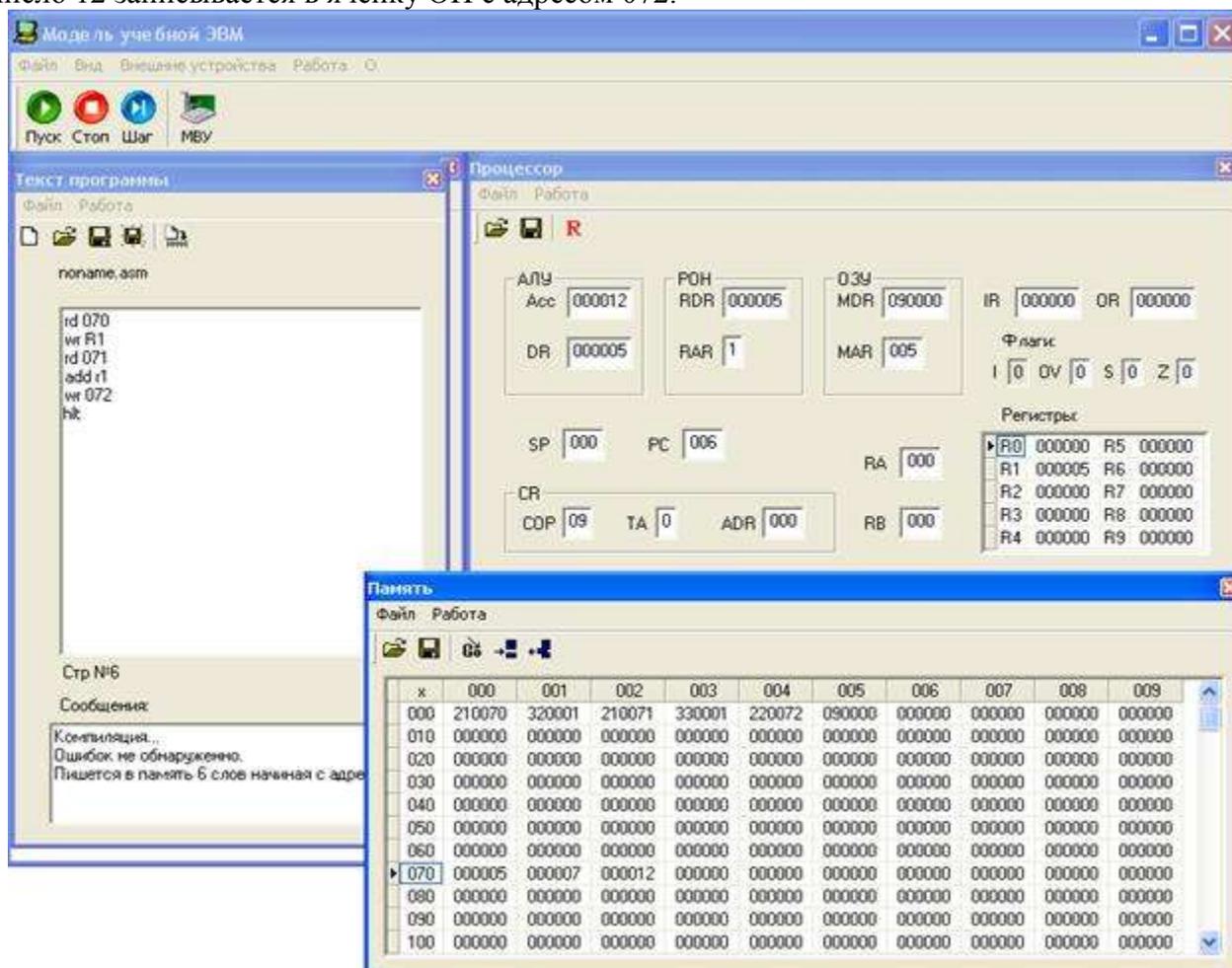


Рис. 12

### Контрольные вопросы

1. Из каких основных элементов состоит ЭВМ и какие из них представлены в модели?
2. Что такое система команд ЭВМ?
3. Какие классы команд представлены в модели?
4. Какие действия выполняют команды передачи управления?
5. Какие способы адресации использованы в модели ЭВМ? В чем отличия между ними?
6. Какие ограничения накладываются на способ представления данных в модели?
7. Какие режимы работы предусмотрены в модели и в чем отличия между ними?
8. Как записать программу в машинных кодах в память модели ЭВМ?
9. Какие способы адресации операндов применяются в командах ЭВМ?
10. Какие команды относятся к классу передачи управления?

11. Как просмотреть содержимое регистров процессора и изменить содержимое некоторых регистров?

12. Как просмотреть и, при необходимости, отредактировать содержимое ячейки памяти? Как запустить выполнение программы в режиме приостановки работы после выполнения каждой команды?

**Форма представления результата:**

Отчет о проделанной работе.

**Критерии оценки:**

Оценка «отлично» ставится, если задание выполнено верно.

Оценка «хорошо» ставится, если ход выполнения задания верный, но была допущена одна или две ошибки, приведшие к неправильному результату.

Оценка «удовлетворительно» ставится, если приведено неполное выполнение задания.

Оценка «неудовлетворительно» ставится, если задание не выполнено.

## Тема 2.4 Технологии повышения производительности процессоров

### Лабораторное занятие № 6

Компиляция и запуск программ в машинных кодах.

**Цель:** научиться компилировать и запускать программы в машинных кодах.

**Выполнив работу, Вы будете:**

**уметь:**

- выполнять компиляцию и запуск программ в машинных кодах;
- использовать современное программное обеспечение при решении профессиональных задач.

**Материальное обеспечение:**

Методические указания для выполнения лабораторных работ, компьютер.

**Задание:**

Выполнить программу, используя машинные коды.

**Порядок выполнения работы:**

#### Задание 1.

Дана последовательность мнемочкодов, которую необходимо преобразовать в машинные коды, занести в ОЗУ ЭВМ, выполнить в режиме **Шаг** и зафиксировать изменение состояний программно-доступных объектов ЭВМ (табл. 5).

Таблица 5.

Последовательность	Значения				
Команды	RD#20	WR30	ADD #5	WR@30	JNZ 002
Коды	21 1 020	22 0 030	23 1 005	22 2 030	12 0 002

Введем полученные коды последовательно в ячейки ОЗУ, начиная с адреса 000. Выполняя команды в режиме **Шаг**, будем фиксировать изменения программно-доступных объектов (в данном случае это Асс, РС и ячейки ОЗУ 020 и 030) в табл. 6.

Таблица 6. Содержимое регистров

РС	Асс	М(30)	М(20)

#### Задание 2.

1. Ознакомиться с архитектурой модели учебной ЭВМ.
  2. Записать в ОЗУ "программу", состоящую из пяти команд — варианты задания выбрать из табл. 7. Команды разместить в последовательных ячейках памяти.
  3. При необходимости установить начальное значение в устройство ввода IR.
  4. Определить те программно-доступные объекты ЭВМ, которые будут изменяться при выполнении этих команд.
  5. Выполнить в режиме Шаг введенную последовательность команд, фиксируя изменения значений объектов, определенных в п. 4, в таблице (см. форму табл. 6).
  6. Если в программе образуется цикл, необходимо просмотреть не более двух повторений каждой команды, входящей в тело цикла.
- Таблица 7. Варианты задания.

№	IR	Команда 1	Команда 2	Команда 3	Команда 4	Команда 5
1	000007	IN	MUL #2	WR10	WR @10	JNS 001
2	X	RD #17	SUB #9	WR16	WR@16	JNS 001
3	100029	IN	ADD #16	WR8	WR@8	JS 001
4	X	RD #2	MUL #6	WR 11	WR @11	JNZ 00
5	000016	IN	WR8	DIV #4	WR @8	JMP 002
6	X	RD #4	WR 11	RD @11	ADD #330	JS 000
7	000000	IN	WR9	RD @9	SUB#1	JS 001
8	X	RD 14	SUB #8	WR 8	WR @8	JNZ 001
9	100005	IN	ADD #12	WR 10	WR @10	JS 004
10	X	RD 4	ADD #15	WR 13	WR @13	JMP 001
11	000315	IN	SUB #308	WR11	WR @11	JMP 001
12	X	RD #988	ADD #19	WR9	WR @9	JNZ 001
13	000017	IN	WR11	ADD 11	WR @11	JMP 002
14	X	RD #5	MUL #9	WR10	WR @10	JNZ 001

#### Форма представления результата:

1. Формулировка варианта задания.
2. Машинные коды команд, соответствующих варианту задания.
3. Результаты выполнения последовательности команд в форме табл. 6.

#### Критерии оценки:

- Оценка «отлично» ставится, если задание выполнено верно.
- Оценка «хорошо» ставится, если ход выполнения задания верный, но была допущена одна или две ошибки, приведшие к неправильному результату.
- Оценка «удовлетворительно» ставится, если приведено неполное выполнение задания.
- Оценка «неудовлетворительно» ставится, если задание не выполнено.

## Тема 2.5 Компоненты системного блока

### Лабораторное занятие № 7

Сборка, установка, подключение комплектующих в корпус ПК

**Цель:** 1. Научиться собирать системный блок компьютера с помощью стенда на базе: EP-5VKM3I / Celeron D 2.8 / DDR 256Mb PC2100/ FDD / HDD 160Gb / SVGA GV-N55128D,  
2. изучить правила и порядок сборки, знать назначение шлейфов и разъемов.

#### Выполнив работу, Вы будете:

##### *уметь:*

- собирать системный блок компьютера;
- выполнять поиск необходимой информации и структурировать ее;
- понимать профессиональный текст.

#### Материальное обеспечение:

Методические указания для выполнения лабораторных работ, компьютер, стенд.

#### Задание:

Собрать системный блок компьютера.

#### Краткие теоретические сведения:

Состав стенда: (Характеристики заполнить САМОСТОЯТЕЛЬНО)



#### 1. Материнская плата:

Производитель	EPoX
Модель	EP-5VKM3I
Чипсет мат. платы	
Формат платы	
Гнездо процессора	
Макс. кол-во процессоров на материнской	

плате	
Поддержка типов процессоров	
Частота шины	
Видео M/B	
Звук	
Количество разъемов DDR DIMM	
Тип поддерживаемой памяти	
Официально поддерживаемые стандарты памяти	
Мах объем оперативной памяти	
Интегрированный RAID-контроллер	
Serial ATA	
Поддержка UDMA/66	
Поддержка UDMA/100	
Поддержка UDMA/133	
Сеть	
AGP	
Количество разъемов PCI	
Клавиатура/мышь	
Порты	

## 2. Процессор:



Производитель	Intel
Модель	Celeron D 336
Назначение	
Тип оборудования	
Описание	
Корпус	
Частота шины CPU	
Рассеиваемая мощность	
Критическая температура	
Частота работы процессора	
Гнездо процессора	
Макс. кол-во процессоров на материнской плате	
Ядро	

Кэш L1	
Кэш L2	
Поддержка Hyper Threading	
Поддержка 64 бит	
Количество ядер	
Умножение	
Видеоядро процессора	
Техпроцесс	

### 3. Оперативная память:

Производитель	HYNIX
Тип оборудования	Модуль памяти DDR SDRAM
Частота функционирования	
Пропускная способность памяти	

### 4. НЖМД:

Производитель	Seagate
Модель	160 Gb Barracuda 7200.9 3160812AS
Актуальная емкость	
Скорость вращения шпинделя	
Буфер HDD	
Среднее время доступа	
Время перехода с дорожки на дорожку	
Интерфейс HDD	
Пропускная способность интерфейса	

### 5. Видеокарта:



Производитель	GIGABYTE
Модель	GV-N55128D
Тип оборудования	
Поддержка API	
RAMDAC	
GPU	
Частота GPU	
Тип видеопамяти	
Разрядность шины видеопамяти	

Частота видеопамати	
Кол-во пиксельных конвейеров	
Интерфейс	
TV out	
Порты	
<b>6. Блок питания:</b>	
Производитель	MICROLAB
Тип оборудования	Блок питания ATX
Максимальная нагрузка	
Охлаждение блока питания	
Входное напряжение	220 ~ 240 В +/-10%
Мощность блока питания	360 Вт
Коннектор питания мат.платы	
Разъемы для подключения MOLEX/FDD/SATA	

## Порядок выполнения работы:

### 1. Установка процессора

1. Отведите рычаг чуть в сторону от разъема и поднимите его вертикально.



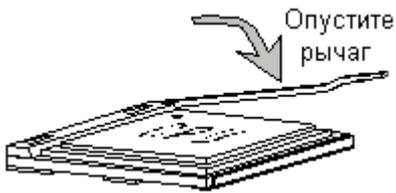
2. Обратите внимание на скошенный угол на процессоре. Он указывает на основание рычага. Процессор войдет в разъем только при правильном его расположении.



Если процессор не удастся установить в разъем, обратитесь к руководству по установке процессора. Никогда не применяйте чрезмерные усилия для установки процессора, так как это может привести к сминанию ножек и необратимому повреждению процессора.

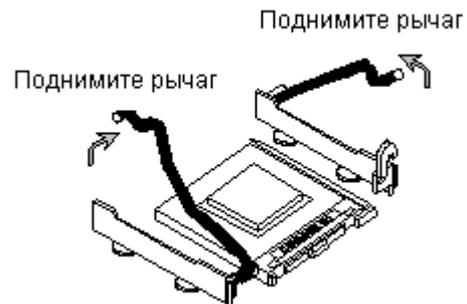


3. Опустите процессор в разъем до упора и затем опустите рычаг. Установка процессора на этом закончена.



## 2. Установка вентилятора процессора

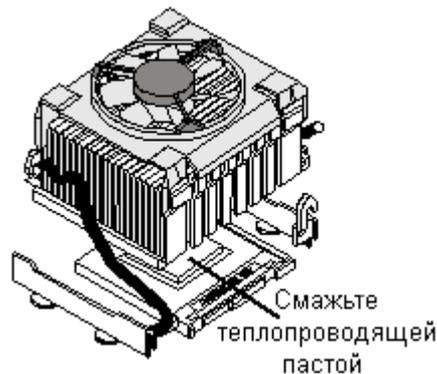
Для примера рассмотрим процессор Intel® Pentium® 4 (socket 423) и вентилятор к нему, чтобы показать процедуру установки. Уточните у преподавателя, какой вам нужен процессор и вентилятор



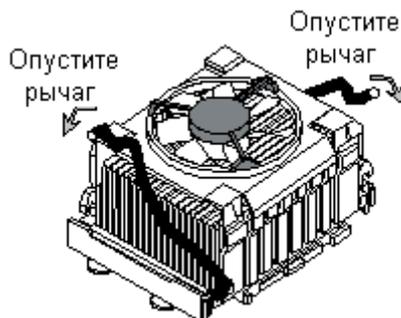
1. Поднимите рычаги механизма крепления.
2. Расположите вентилятор процессора над механизмом крепления.



Мы советуем нанести небольшое количество теплопроводящей пасты на корпус процессора. Это позволит более эффективно рассеивать выделяющееся тепло.



3. Опустите оба рычага, чтобы прижать вентилятор к процессору.
4. Подключите вентилятор к разъему питания вентилятора, расположенному на системной плате.

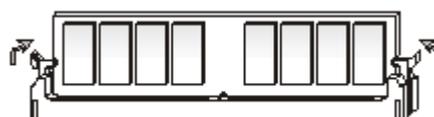
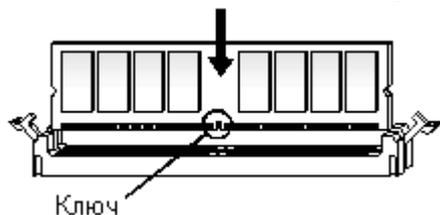


### 3. Установка модулей памяти

1. Откройте пластмассовые защелки с обеих сторон разъема DIMM.



2. Вставьте модуль памяти вертикально в разъем DIMM. Нажмите на него сверху.



3. Защелки автоматически закроются.



Не прикасайтесь к контактам модулей памяти. Загрязнение контактов может вызвать неправильную работу модулей.

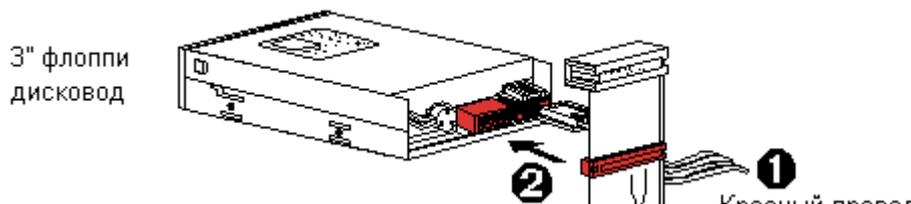
### 4. Подключение дисководов

Флоппи дисковод, подключенный к одному из этих разъемов, будет восприниматься системой как устройство "А".

Флоппи дисковод, подключенный к этому разъему, будет восприниматься системой как устройство "В".

Подключите эту часть кабеля к разъему на системной плате.

1. Подключите разъем питания к флоппи дисководу.
2. Подключите сигнальный кабель к флоппи дисководу.





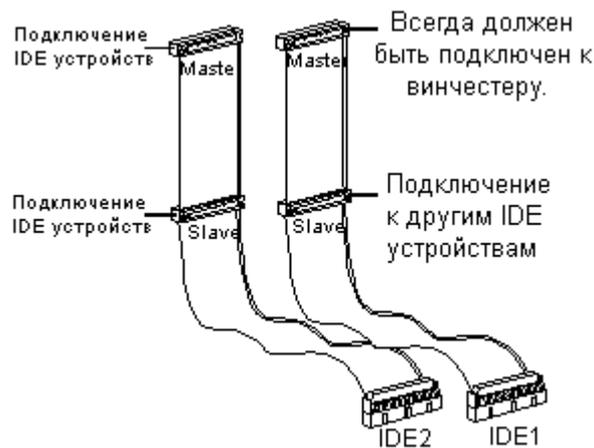
Не предусмотрено конструктивных решений, предотвращающих неверное подключение сигнального кабеля флоппи дисководов. Если флоппи дисковод после сборки не будет работать, а светодиод дисководов будет при этом постоянно светиться, переверните сигнальный кабель.

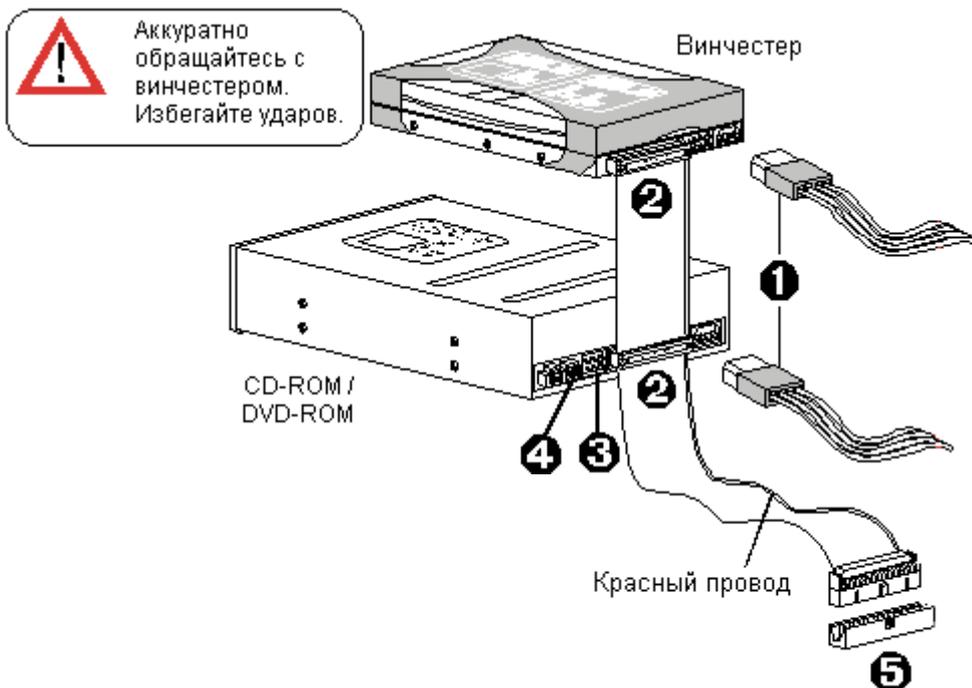
3. Подключите другую часть кабеля к системной плате.

### 5. Подключение IDE устройств

Первый винчестер в системе следует подключать к кабелю IDE1. К этому кабелю можно подключить винчестеры в режиме Master и Slave.

Если вы подключаете два IDE устройства одним кабелем, следует переключить второе устройство в режим Slave с помощью установленных на нем переключателей.





1. Подключите разъемы питания IDE устройств.
2. Подключите сигнальные кабели IDE устройств.
3. Подключите аудио кабели CD-ROM/DVD-ROM к разъемам на системной плате.
4. Переключите устройства в режим Slave, если это необходимо. Обратитесь к Руководству пользователя вашего CD-ROM/DVD-ROM.
5. Подключите кабели к системной плате.

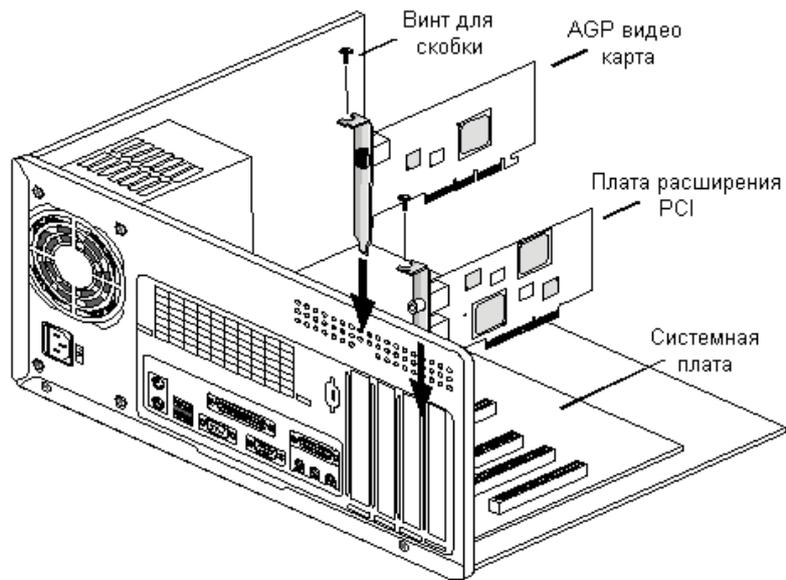
### 6. Установка плат расширения (видеоадаптера)

1. Откройте крышку корпуса.
2. Выньте заглушку задней панели напротив выбранного разъема расширения.



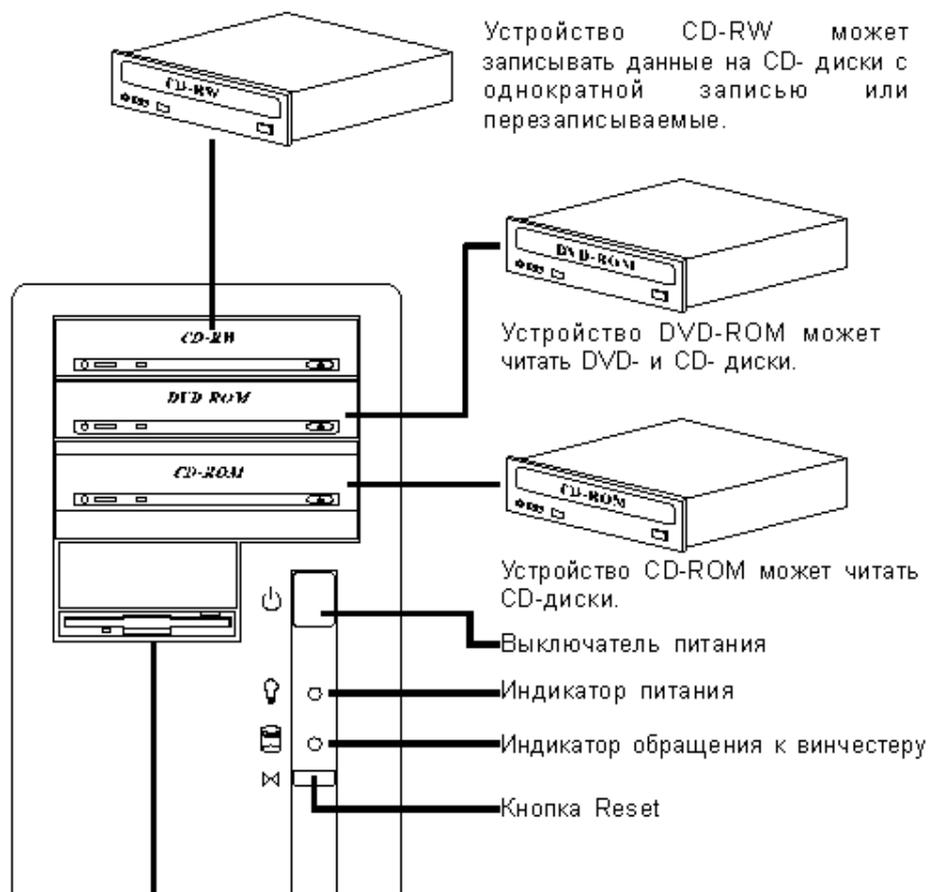
Не следует выполнять эту работу незащищенными руками, так как острые края металла могут порезать кожу. Используйте защитные перчатки и пассатижи.

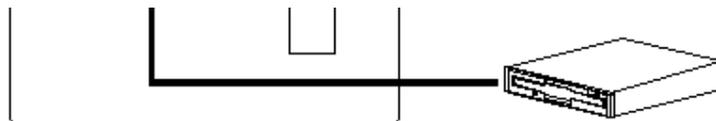
3. Найдите свободный разъем PCI (AGP) на вашей системной плате.
4. Поместите плату над соответствующим разъемом и слегка вставьте ее. Аккуратно, но плотно прижмите ее таким образом, чтобы плата надежно встала в соответствующий разъем.
5. Зафиксируйте плату с помощью винта.
6. Установите остальные платы расширения и устройства, подсоедините все кабели и закройте крышку корпуса.



## 7. Передняя панель

Внешний вид передней панели вашего корпуса может отличаться от нижеприведенного.

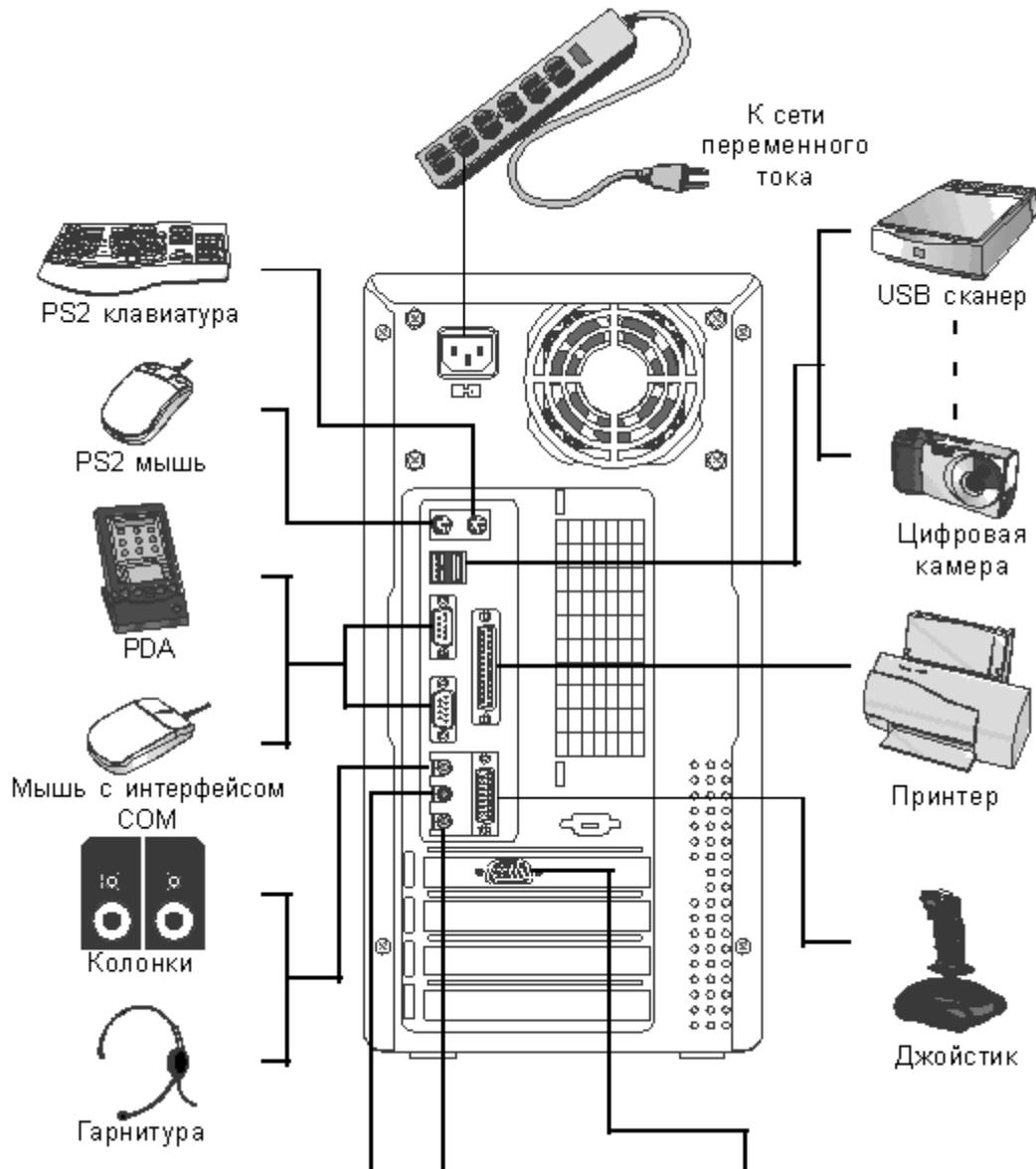




3" флоппи дисковод может читать или записывать данные с/на гибкие диски 3 1/2".

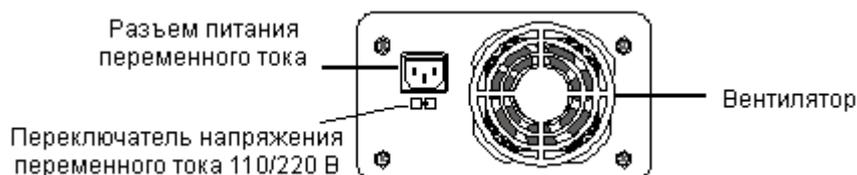
## 8. Задняя панель

На рисунке показана задняя панель системного блока в корпусе ATX. В стенде использован корпус форм-фактора AT.





## 9. Подключение разъема питания



Запрещается подключать шнур питания до завершения сборки компьютера.



Запрещается самостоятельно ремонтировать блок питания или использовать его нецелевым образом. Проконсультируйтесь с фирмой поставщиком, если у вас возникли проблемы с блоком питания.

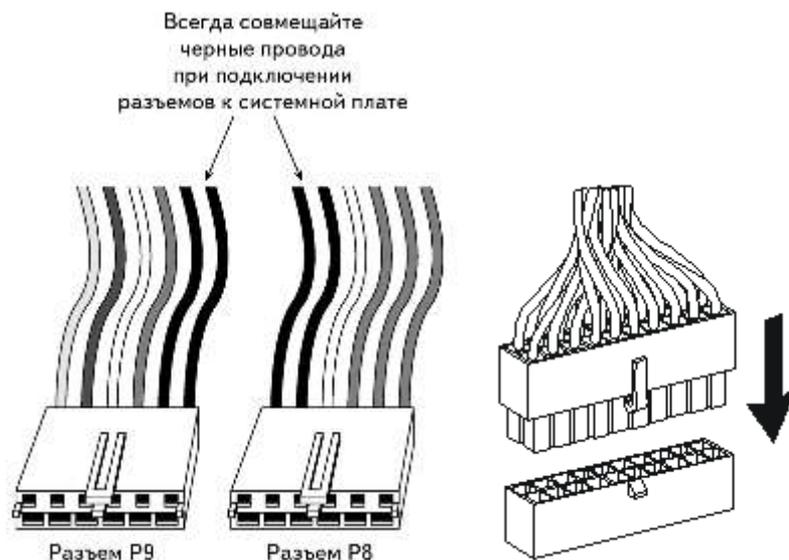


Запрещается прикасаться влажными руками к корпусу блока питания даже в том случае, если он не подсоединен к сети. Капельки воды с рук могут вызвать короткое замыкание.



Необходимо проверить переключатель 110/220 В перед включением в сеть по окончании сборки. Неправильная установка этого переключателя может вызвать повреждение блока питания или даже возгорание.

1. Подключите сдвоенный разъем стандарта АТ (или 20-контактный разъем питания стандарта АТХ) к соответствующей розетке, расположенной на системной плате.
2. Для надежного крепления следует вставить разъем до щелчка.



### Контрольные вопросы

1. В каком порядке осуществляется сборка системного блока компьютера?
2. Какие правила необходимо соблюдать при сборке системного блока?
3. Для чего на всех шлейфах, прилагаемых к стенду один проводник помечен красным цветом?
4. К какому форм-фактору относится данная системная плата? По каким признакам можно это определить?
5. Что необходимо сделать, если на одном IDE шлейфе подключены два устройства?
6. Какой тип памяти используется на данном стенде?
7. Что необходимо учитывать при подключении дисководов?
8. Поясните назначение всех разъемов на задней панели системного блока.

### Форма представления результата:

Отчет о проделанной работе:

1. Тема и цель работы
2. Описание хода проведения работы, ошибок, возникших в результате выполнения работы, аппаратных характеристик компьютера
3. Ответы на контрольные вопросы

### Критерии оценки:

Оценка «отлично» ставится, если задание выполнено верно.

Оценка «хорошо» ставится, если ход выполнения задания верный, но была допущена одна или две ошибки, приведшие к неправильному результату.

Оценка «удовлетворительно» ставится, если приведено неполное выполнение задания.

Оценка «неудовлетворительно» ставится, если задание не выполнено.

## Тема 2.5 Компоненты системного блока

### Лабораторное занятие № 8

#### Установка конфигурации системы при помощи утилиты CMOS Setup

**Цель:** изучить настройки программы BIOS Setup, уметь пользоваться параметрами BIOS для настройки компьютера на оптимальную работу.

**Выполнив работу, Вы будете:**

**уметь:**

- выполнять установку конфигурации системы;
- понимать профессиональный текст.

**Материальное обеспечение:**

Методические указания для выполнения лабораторных работ, компьютер.

**Задание:**

Выполнить установку конфигурации системы при помощи утилиты CMOS Setup

**Краткие теоретические сведения:**

**Общие сведения**

**BIOS (Basic Input/Output System)**, что в переводе на русский означает: «Базовая система ввода/вывода». Такое название получил набор микропрограмм, отвечающих за работу базовых функций видеоадаптеров, дисплеев, дисковых накопителей, дисководов, клавиатур, мышей и других основных устройств ввода/вывода информации.

Основными функциями BIOS являются начальный запуск ПК, тестирование и первичная настройка оборудования, распределение ресурсов между устройствами и активация процедуры загрузки операционной системы.

С учетом того, что BIOS отвечает за самый начальный этап загрузки компьютера вне зависимости от его конфигурации, то эта программа должна быть доступна для базовых устройств сразу же после нажатия на кнопку включения ПК. Именно поэтому она хранится не на жестком диске, как большинство обычных приложений, а записывается в специальную микросхему флэш-памяти, расположенную на системной плате.

Помимо флэш-памяти, в которой хранится сама BIOS, на системной плате существует и еще один вид памяти, который предусмотрен для хранения настроек конфигурации этой программы. Изготавливается он с применением комплементарного метало-оксидного полупроводника или CMOS (Complementary Metal Oxide Semiconductor). Именно этой аббревиатурой и называют специализированную память, в которой содержатся данные о запуске компьютера, используемые BIOS.

CMOS-память питается от батарейки, установленной на материнской плате. Благодаря этому, при отключении компьютера от розетки все настройки BIOS сохраняются. На старых компьютерах функции CMOS-памяти были возложены на отдельную микросхему. В современных же ПК она является частью чипсета.

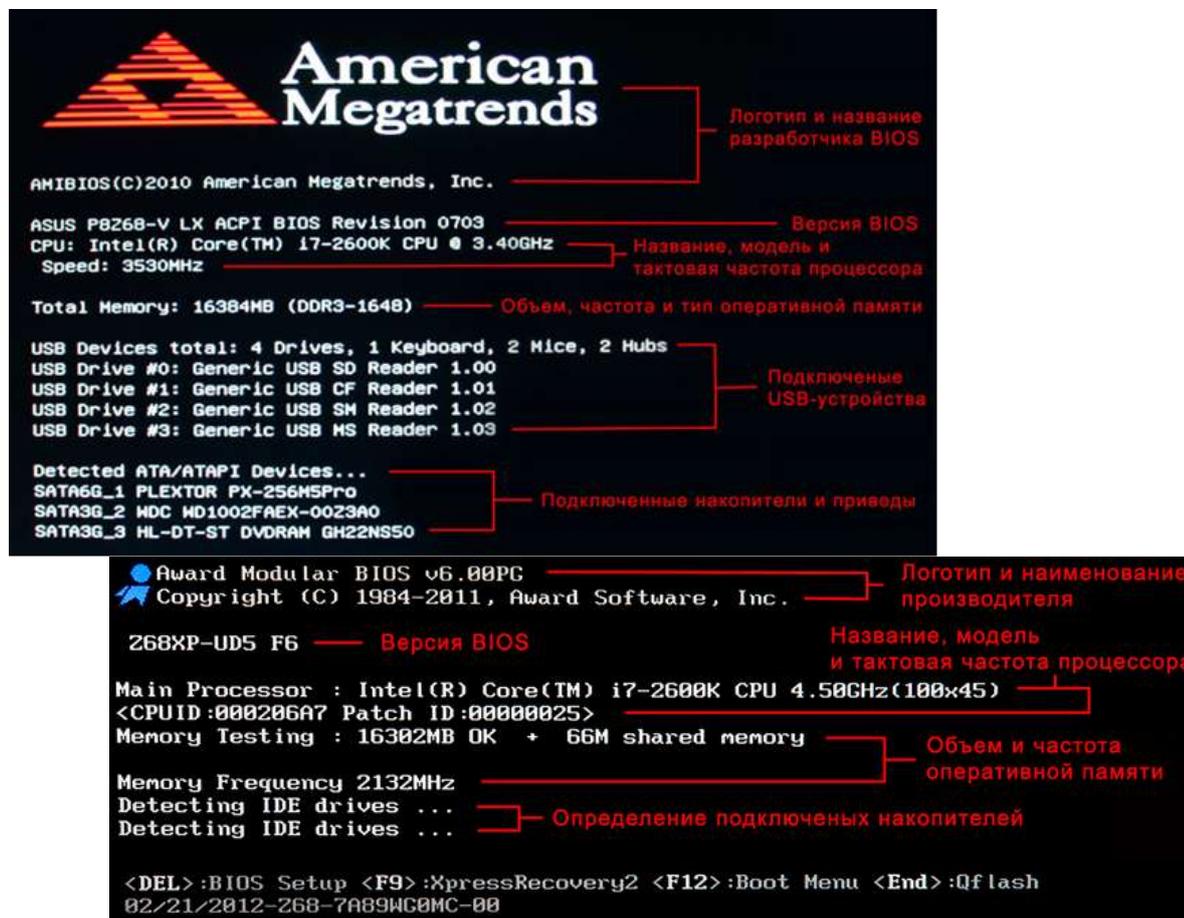
#### Процедура POST и первоначальная загрузка ПК

После нажатия кнопки включения компьютера, первым запускается блок питания, начиная подавать напряжение на материнскую плату. Если оно в норме, то чипсет дает команду на сброс внутренней памяти центрального процессора и его запуск. После

этого процессор начинает последовательно считывать и выполнять команды, записанные в системной памяти, роль которой выполняет как раз микросхема BIOS.

В самом начале процессор получает команду на проведение самотестирования компонентов компьютера (POST - Power-On Self-Test). Процедура POST включает в себя несколько этапов, прохождение большинства которых вы можете наблюдать на экране ПК сразу после его включения. Последовательность происходящего перед началом загрузки операционной системы такова:

1. Сначала происходит определение основных системных устройств.
2. Далее осуществляется инициализация установленной оперативной памяти, запись в нее BIOS и проверка микропрограммы на целостность данных.
3. Третий шаг - настройка набора системной логики, или проще говоря, чипсета.
4. Затем происходит поиск и определение видеокарты. Если в компьютере установлен внешний (самостоятельный) видеоадаптер, то он будет иметь собственную BIOS, которую основная системная BIOS будет искать в определенном диапазоне адресов памяти. Если внешний графический адаптер будет найден, то первое, что вы увидите на экране, будет изображение с названием видеокарты, сформированное ее BIOS.
5. После нахождения графического адаптера, начинается проверка целостности параметров BIOS и состояния батарейки. В этот момент на экране монитора одна за другой начинают появляться те самые таинственные белые надписи, вызывающие трепет у неопытных пользователей из-за непонимания происходящего. Но на самом деле ничего сверхъестественного в этот момент не происходит, в чем вы сами сейчас убедитесь. Первая, самая верхняя надпись, как правило, содержит логотип разработчиков BIOS и информацию об ее установленной версии.



6. Затем запускается тестирование центрального процессора, по окончании которого на дисплей выводятся данные об установленном чипе: название производителя, модели и его тактовая частота.

7. Следом начинается тестирование оперативной памяти. Если все проходит удачно, то на экран выводится общий установленный объем ОЗУ с надписью ОК.

8. По окончании проверки основных компонентов ПК, начинается поиск клавиатуры и тестирование других портов ввода/вывода. В некоторых случаях, на этом этапе загрузка компьютера может остановиться, если системе не удастся обнаружить подключенную клавиатуру. При этом на экран сразу же будет выведено об этом предупреждение.

9. Далее начинается определение подключенных к компьютеру накопителей, включая оптические приводы, жесткие диски и флэш-диски. Сведения о найденных устройствах выводятся на экран. В том случае, если на системной плате установлено несколько контроллеров от разных производителей, то процедура их инициализации может быть отображена на разных экранах.

```
Serial ATA AHCI BIOS, Version iSrc 1.20E
Copyright (c) 2003-2008 Intel Corporation
** This version supports only Hard Disk and CDROM drives **
Please wait. This will take few seconds.

Controller Bus#00, Device#1F, Function#02: 06 Ports, 06 Devices
Port-00: Hard Disk, OCZ-VERTEX3 MI
Port-01: Hard Disk, WDC WD6402AAEX-00Y9A0
Port-02: Hard Disk, ST31500341AS
Port-03: Hard Disk, Hitachi HDS724040ALE640
Port-04: CDROM, Optiarc DVD RW AD-5260S
Port-05: Hard Disk, WDC WD20EARS-00MVWB0
```

10. На завершающем этапе осуществляется распределение ресурсов между найденными внутренними устройствами ПК. В старых компьютерах, после этого осуществляется вывод на дисплей итоговой таблицы со всем обнаруженным оборудованием. В современных машинах таблица на дисплей уже не выводится.

11. Наконец, если процедура POST прошла успешно, BIOS начинает поиск в подключенных накопителях Главной Загрузочной Области (MBR), где содержатся данные о запуске операционной системы и загрузочном устройстве, которому необходимо передать дальнейшее управление.

В зависимости от установленной на компьютер версии BIOS, прохождение процедуры POST может проходить с небольшими изменениями от вышеописанного порядка, но в целом, все основные этапы, будут выполняться при загрузке каждого ПК.

## UEFI BIOS

Базовые принципы работы системной BIOS для настольных компьютеров были сформированы в далеких 80-ых годах прошлого века. За прошедшие десятилетия компьютерная индустрия бурно развивалась и за это время постоянно случались ситуации, когда новые модели устройств оказывались несовместимыми с определенными версиями BIOS. Что бы разрешать эти проблемы, разработчикам постоянно приходилось модифицировать код базовой системы ввода/вывода, но в итоге целый ряд программных ограничений так и остался неизменным со времен первых домашних ПК. Такая ситуация привела к тому, что BIOS в своем классическом варианте окончательно перестал удовлетворять требованиям современного компьютерного железа, мешая его

распространению в массовом секторе персоналок. Стало понятно, что необходимо что-то менять.

В 2011 году, с запуском в производство материнских плат для процессоров Intel поколения Sandy Bridge, устанавливаемых в разъем LGA1155, началось массовое внедрение нового программного интерфейса для начальной загрузки компьютера - UEFI.

На самом деле первая версия данной альтернативы обычной BIOS была разработана и успешно использована компанией Intel в серверных системах еще в конце 90-ых годов. Тогда, новый интерфейс для начальной загрузки ПК назывался EFI (Extensible Firmware Interface), но уже в 2005 году его новая спецификация получила название UEFI (Unified Extensible Firmware Interface). На сегодняшний день эти две аббревиатуры считаются синонимами.

Как видите, производители системных плат не особо спешили переходить к новому стандарту, до последнего пытаясь совершенствовать традиционные вариации BIOS. Но очевидная отсталость этой системы, включая ее 16-битный интерфейс, не возможность использовать более 1 Мб адресного пространства памяти, отсутствие поддержки накопителей объемом более 2 Тб и другие постоянные неразрешимые проблемы совместимости с новым оборудованием все же стали серьезным аргументом для перехода на новое программное решение.

Какие же изменения принес с собой новый загрузочный интерфейс, предложенный Intel и в чем его отличия от BIOS? Как и в случае с BIOS, основной задачей UEFI является корректное определение оборудования сразу после включения ПК и передача управления компьютером операционной системе. Но при этом, перемены в UEFI настолько глубоки, что сравнивать ее с BIOS было бы просто некорректно.

BIOS – это практически неизменяемый программный код, вшитый в специальную микросхему и взаимодействующий напрямую с компьютерным оборудованием с помощью собственных программных средств. Процедура загрузки компьютера с помощью BIOS проста: сразу после включения компьютера производится проверка оборудования и загрузка простых универсальных драйверов для основных аппаратных компонентов. После этого BIOS находит загрузчик операционной системы и его активирует. Далее происходит загрузка ОС.

Систему UEFI можно назвать прослойкой между аппаратными компонентами компьютера, с их собственными микропрограммами-прошивками, и операционной системой, что позволяет ей так же выполнять функции BIOS. Но в отличие от BIOS, UEFI представляет собой модульный программируемый интерфейс, включающий тестовые, рабочие и загрузочные сервисы, драйверы устройств, протоколы коммуникаций, функциональные расширения и собственную графическую оболочку, что делает его

похожим на сильно облегченную операционную систему. При этом пользовательский интерфейс в UEFI современен, поддерживает управление мышью и может быть локализован на несколько языков, включая русский.

Важным преимуществом EFI является ее кроссплатформенность и

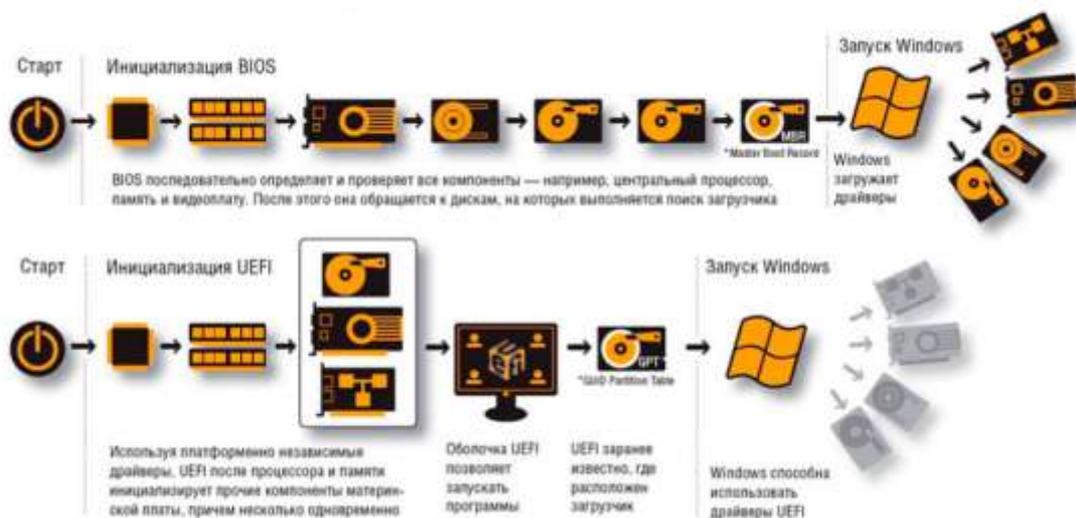


независимость от процессорной архитектуры. Спецификации этой системы позволяют работать ей практически с любой комбинацией чипов, будь то архитектура x86 (Intel, AMD) или ARM. Более того UEFI имеет прямой доступ ко всему аппаратному обеспечению компьютера и платформенно независимые драйверы, что дает возможность без запуска ОС организовать, например, выход в интернет или резервное копирование дисков.

В отличие от BIOS, код UEFI и вся ее служебная информация может храниться не только в специальной микросхеме, но и на разделах как внутренних, так и внешних жестких дисков, а так же сетевых хранилищах. В свою очередь, тот факт, что загрузочные данные могут размещаться на вместилищах накопителях, позволяет за счет модульной архитектуры наделять EFI богатыми функциональными возможностями. Например, это могут быть развитые средства диагностики, или полезные утилиты, которые можно будет использовать как на этапе начальной загрузки ПК, так и после запуска ОС.

Еще одной ключевой особенностью UEFI является возможность работы с жесткими дисками огромных объемов, размеченных по стандарту GPT (Guid Partition Table). Последний не поддерживается ни одной модификацией BIOS, так как имеет 64-битные адреса секторов.

Загрузка ПК на базе UEFI, как и в случае с BIOS, начинается с инициализации устройств. Но при этом, данная процедура происходит гораздо быстрее, так как UEFI может определять сразу несколько компонентов одновременно в параллельном режиме (BIOS инициализирует все устройства по очереди). Затем, происходит загрузка самой системы UEFI, под управлением которой выполняется какой-либо набор необходимых действий (загрузка драйверов, инициализация загрузочного накопителя, запуск загрузочных служб и т.д.), и только после этого осуществляется запуск операционной системы.



Может показаться, что такая многоступенчатая процедура должна увеличить общее время загрузки ПК, но на самом деле все происходит наоборот. С UEFI система запускается гораздо быстрее, благодаря встроенным драйверам и собственному загрузчику. В итоге, перед стартом, ОС получает исчерпывающую информацию об аппаратной начинке компьютера, что позволяет запускаться ей в течение нескольких секунд.

Несмотря на всю прогрессивность UEFI, все же существует ряд ограничений, сдерживающих активное развитие и распространение этого загрузчика. Дело в том, что для реализации всех возможностей нового загрузочного интерфейса требуется полноценная его поддержка со стороны операционных систем. На сегодняшний день в полной мере использовать возможности UEFI позволяет только Windows 8. Ограниченную поддержку нового интерфейса имеют 64-разрядные версии Windows 7, Vista и Linux на ядре 3.2 и выше.

Так же возможности UEFI используются в загрузочном менеджере BootCamp компанией Apple в собственных системах Mac OS X.

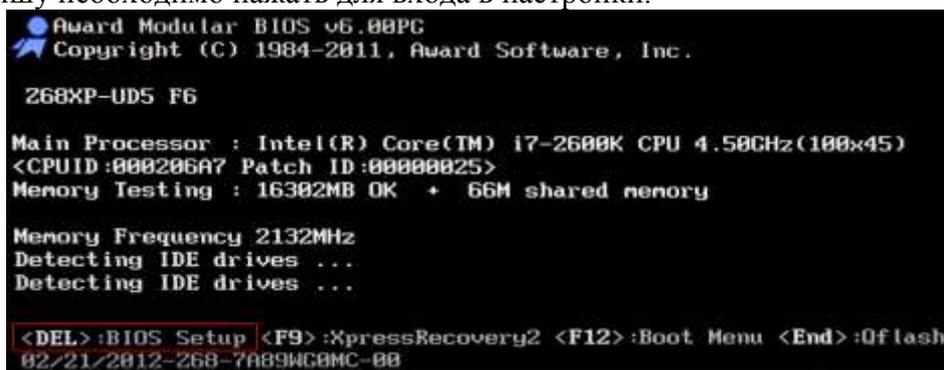
Ну а как же происходит загрузка компьютера с UEFI, если на нем используются неподдерживаемая операционная система (WindowsXP, 32-битная Windows 7) или файловая разметка (MBR)? Для таких случаев в новый загрузочный интерфейс встроены **модуль поддержки** совместимости (Compatibility Support Module), по сути, представляющий из себя традиционную BIOS. Именно поэтому, можно видеть, как многие современные компьютеры, оснащенные системными платами с UEFI, загружаются традиционным способом в режиме эмуляции BIOS. Чаще всего это происходит потому, что их владельцы продолжают использовать разделы HDD с традиционной MBR и не хотят переходить к разметке GPT.

### Программа настройки BIOS

Для того, что бы запустить программу настройки BIOS необходимо во время проведения процедуры первоначального тестирования ПК нажать определенную клавишу или их сочетание. В подавляющем большинстве случаев в настольных компьютерах для входа в BIOS Setup используется клавиша Del, реже F1 или F2. В ноутбуках наоборот, наиболее часто для этих целей задействуются именно функциональные клавиши (F1, F2, F11, F12).



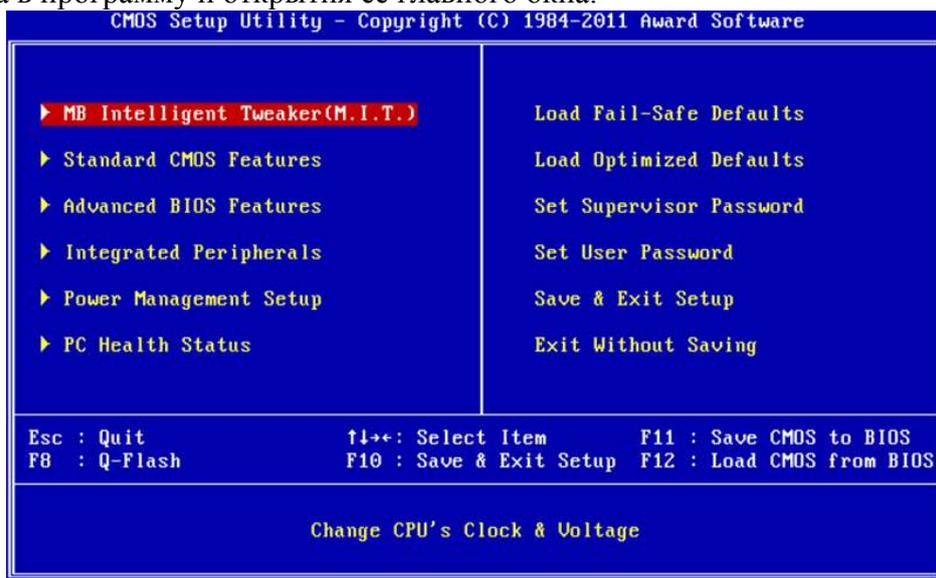
Узнать точно, какие из клавиш используются для запуска BIOS Setup можно из инструкции к компьютеру или системной плате. Так же в некоторых случаях во время прохождения процедуры POST на экран монитора выводится подсказка, о том какую клавишу необходимо нажать для входа в настройки.



Помимо необходимости знать нужную клавишу, для попадания в BIOS Setup, не менее важно выбрать правильный момент ее нажатия. Чтобы не опоздать, лучше сразу после начала загрузки ПК многократно нажимать клавишу входа. В большинстве случаев такой способ гарантировано обеспечивает запуск настроек BIOS.

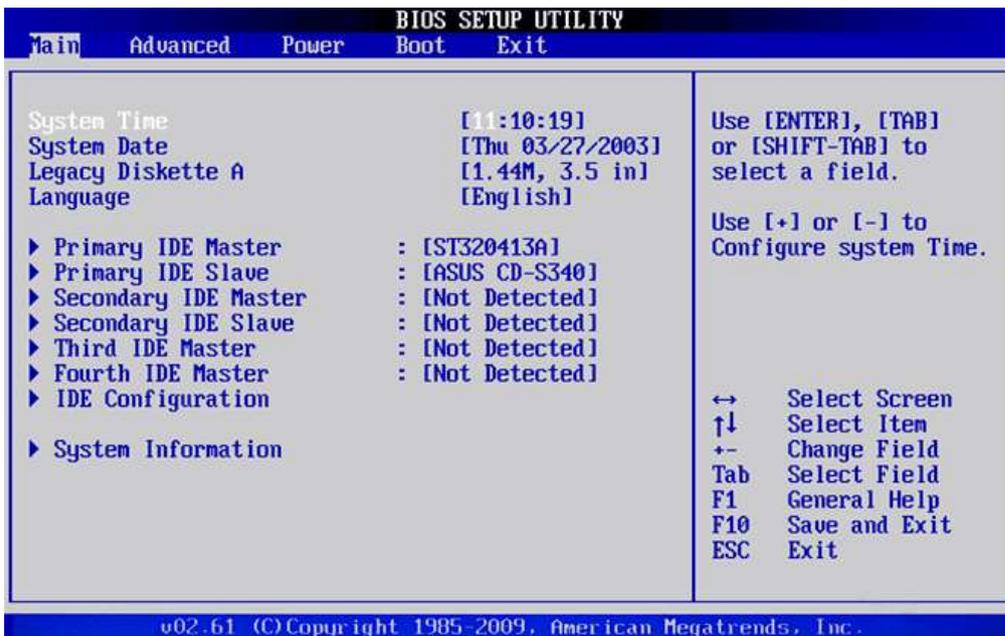
Программа Bios Setup имеет текстовый интерфейс без каких-либо дизайнерских ухищрений и управляется исключительно с помощью клавиатуры. Объясняется это тем, что графическая оболочка этого приложения практически не менялась с 80-ых годов, поэтому все выглядит очень просто и аскетично.

В общем случае интерфейс BIOS Setup бывает двух типов: с расположением главного меню в два столбца или горизонтально. Понять какой тип перед вами можно сразу после входа в программу и открытия ее главного окна.



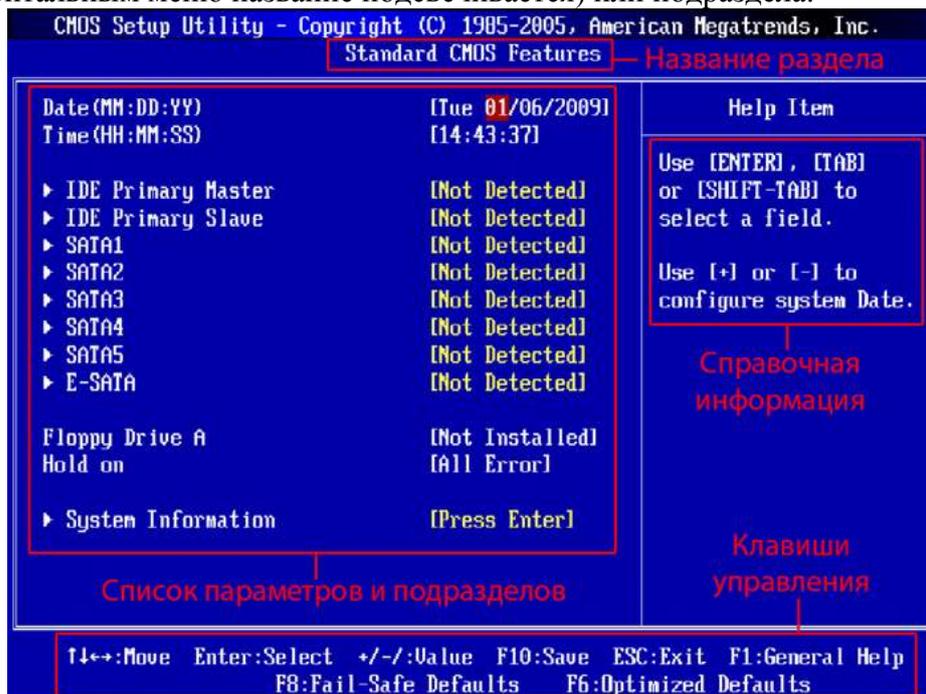
В первом случае вы увидите на синем фоне список разделов, размещенных в два столбца. Такой вариант характерен для версий BIOS, разработанных компанией Phoenix Technologies (AwardBIOS, Award Modular BIOS, Award WorkstationBIOS). Их традиционно в своих системных платах используют такие производители, как MSI, Gigabyte, Foxconn, ECS и другие.

Во втором случае перед вами появится окно с серым фоном, в котором меню с основными разделами будет размещаться сверху экрана, в виде синей горизонтальной полосы. Такой интерфейс, как правило, присущ для BIOS компании American Megatrends (AMIBIOS, Aptio AMIBIOS), использующихся в материнских платах ASUS, Intel, ASRock и некоторых других.

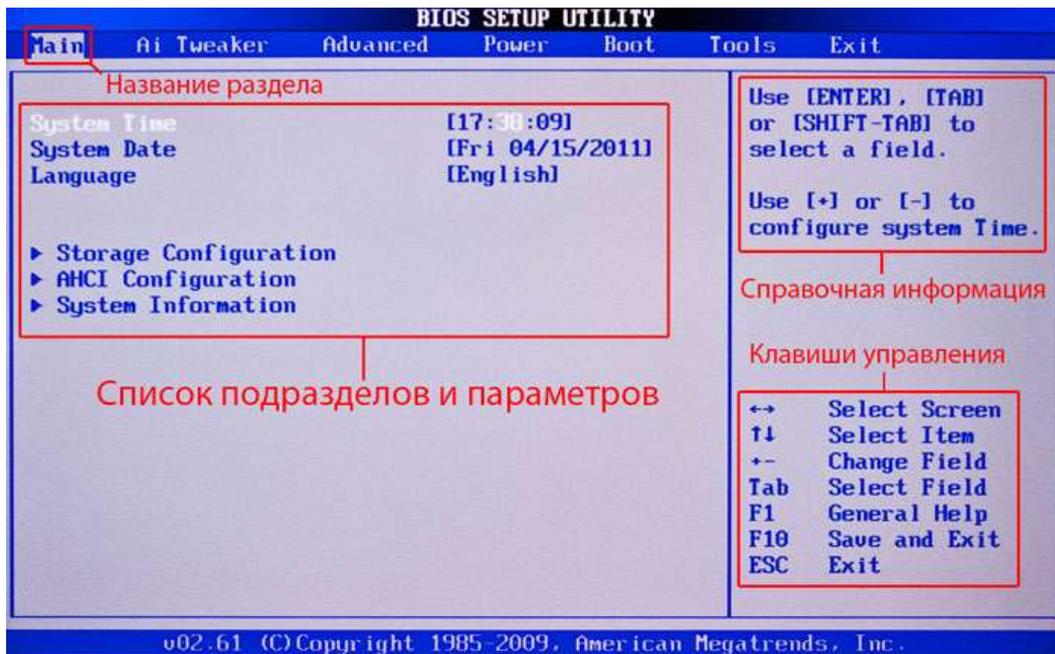


Несмотря на такие различия в интерфейсе этих двух вариантов, все разделы BIOS Setup имеют схожее представление. Что бы убедиться в этом, давайте посмотрим на структуру окон программы в обоих случаях.

В верхней части экрана вы всегда найдете название текущего раздела (в случае с горизонтальным меню название подсвечивается) или подраздела.



Основную часть экрана занимает область, в которой размещается список подразделов (обозначаются треугольными стрелками) и параметров выбранного раздела. Справа от наименования параметров располагаются их значения. При этом стоит учесть, что если параметр выделен бледным цветом (голубым или светло-серым), то он либо имеет статус «только для чтения» и несет исключительно информационный характер, либо для его редактирования необходимо изменить другой, связанный с ним параметр.



Правую часть экрана обычно занимает столбец, в котором выводится краткая справочная информация по выделенному параметру или подразделу, а так же подсказки по возможным действиям и использованию клавиш управления (American Megatrends). В программе настройки BIOS с синим фоном, подсказка по использованию функциональных клавиш располагается обычно в нижней части экрана.

Не смотря на разное цветовое оформление и небольшие различия в расположении на экране рабочих элементов, по своей сути оба интерфейса очень похожи, и преподносят информацию пользователям практически одинаковым образом. Именно поэтому приемы работы с параметрами BIOS в обоих случаях практически одинаковы.

Для навигации по меню и выбора нужных параметров, подразделов или разделов используются клавиши со стрелками, а для их открытия – клавиша «Ввод» («Enter»). За возврат к предыдущему экрану и выход из текущих настроек отвечает клавиша «ESC». Так же при помощи этой клавиши вы можете выйти из BIOS Setup без внесения изменений в настройки, нажав ее в главном меню. Помимо этого неизменными являются функции клавиш «F1», вызывающей справку и «F10», инициализирующей выход из BIOS Setup, из любого места программы с сохранением сделанных изменений. Клавиши «PageUP»/«PageDown» или «+»/«-» традиционно используются для последовательного перебора доступных значений изменяемых параметров.

Кроме вышеперечисленных клавиш, для работы с настройками BIOS могут использоваться и другие функциональные клавиши («F2» - «F9», «F11», «F12»), но их назначение в зависимости от модели платы и ее производителя, может отличаться. Впрочем, что бы понять, за что каждая из них отвечает, несложно. Достаточно обратиться к подсказкам, возникающим на экране или полистать руководство к системной плате.

### Основные разделы BIOS Setup с колоночным главным меню (синий фон)

Каждая модель материнской платы во многих случаях имеет свой уникальный набор настраиваемых параметров, но при этом названия и тематическая направленность основных разделов BIOS Setup обычно остаются неизменными.

## Standard CMOS Futures

В данном разделе сосредоточены основные (стандартные) настройки компьютера, к которым относятся: установка системных даты и времени (Date, Time), параметры дисковых накопителей (IDE Channel), а так же различная информация о системе (сведения об установленном процессоре, количестве оперативной памяти и другие).



### Advanced BIOS Features

Этот раздел содержит расширенные настройки BIOS. К наиболее распространенным из них можно отнести:

Управление кэш-памятью центрального процессора

Параметры, связанные с нюансами загрузки компьютера. Например, здесь можно включить/отключить режим NumLock, режим ускоренной загрузки (Quick Boot), а так же показ логотипа производителя платы во время проведения процедуры самотестирования (Full Screen LOGO Show).

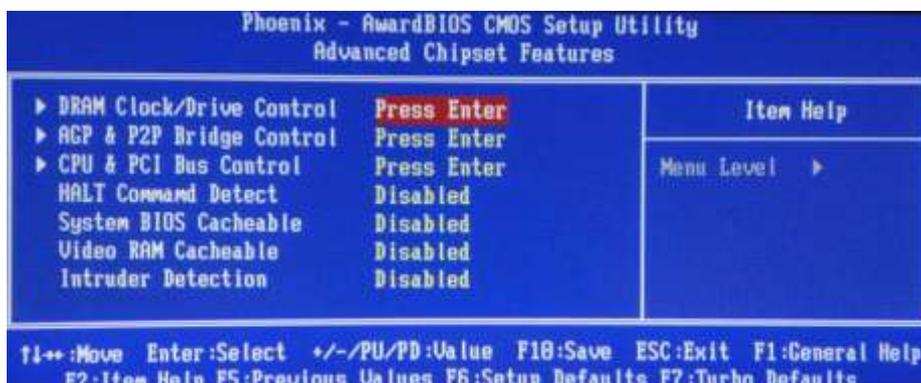
Выбор последовательности опроса загрузочных устройств (First/Second/Third Boot Device). Еще одна самая востребованная функция в BIOS Setup, наряду с установкой даты и времени.

Включение/отключение технологии самоконтроля жесткого диска S.M.A.R.T.



## Advanced Chipset Features

В этом разделе описываются настройки чипсета, установленного в системную плату, вследствие чего набор параметров здесь напрямую зависит от его типа и модификации. В большинстве случаев здесь собраны опции, отвечающие за работу оперативной памяти (регулировка частоты и таймингов), шины обмена данными между процессором и ОЗУ, графической шины AGP/PCI-E и видеоадаптера.



Следует отметить, что в некоторых ситуациях именно с помощью изменения параметров этого раздела можно повысить скорость работы компьютера или как говорят, совершить разгон. Правда, в последнее время опции, отвечающие за увеличение скорости работы ПК, чаще всего выносятся производителями в отдельный специализированный раздел BIOS.

## Integrated Peripherals

Данный раздел содержит параметры, отвечающие за работу, интегрированных в материнскую плату, периферийных устройств, таких как: контроллеры жестких дисков, USB- портов, звуковых и сетевых адаптеров, и прочих.



Например, здесь вы можете включить/отключить встроенную звуковую карту, поддержку USB-устройств ввода или выбрать режим RAID для создания массива жестких дисков.

## Power Management Setup

Здесь собраны опции, отвечающие за электропитание и режимы энергосбережения компьютера. Практически все современные компьютеры позволяют осуществлять управление электропитанием непосредственно из операционной системы, но для этого требуется поддержка со стороны BIOS специализированных стандартов ACPI, режим, и функции которого, регулируются как раз в этом разделе.

Так же здесь вы можете указать, какие действия должны происходить при нажатии на кнопку питания, настроить условия включения ПК и его перехода к пониженному потреблению энергии или выхода из «спячки».



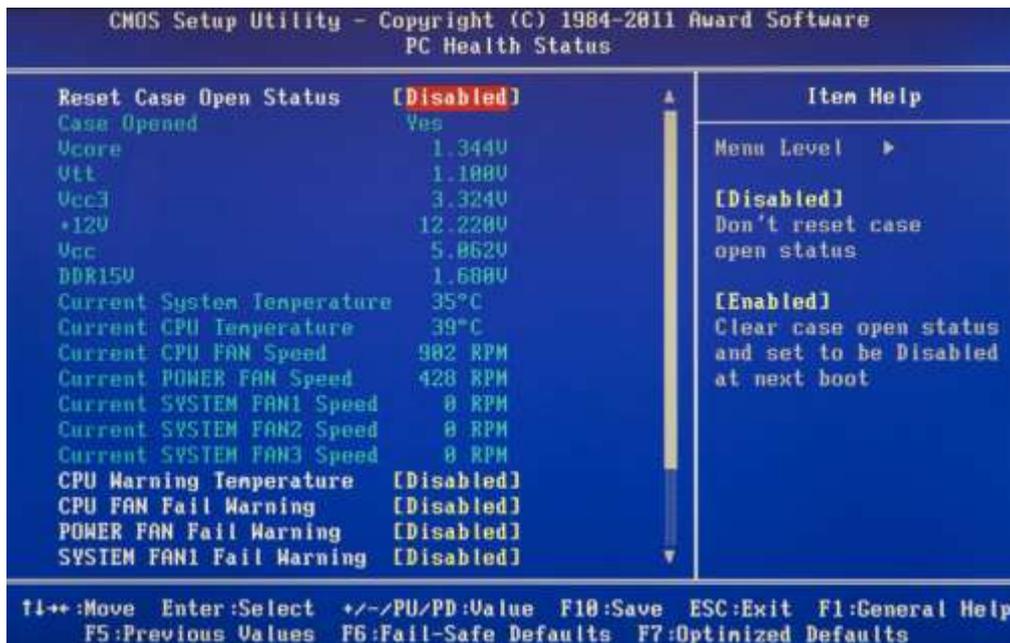
## PnP/PCI Configurations

В этом разделе находятся параметры управления технологии Plug and Play, отвечающей за распределение ресурсов между устройствами ПК и их быстрое конфигурирование, а так же настройки работы шины PCI. Как правило, данные функции с успехом выполняются системой и не требуют ручного вмешательства. Поэтому в современных компьютерах данный раздел может отсутствовать вовсе.

## PC Health Status (H/W Monitor)

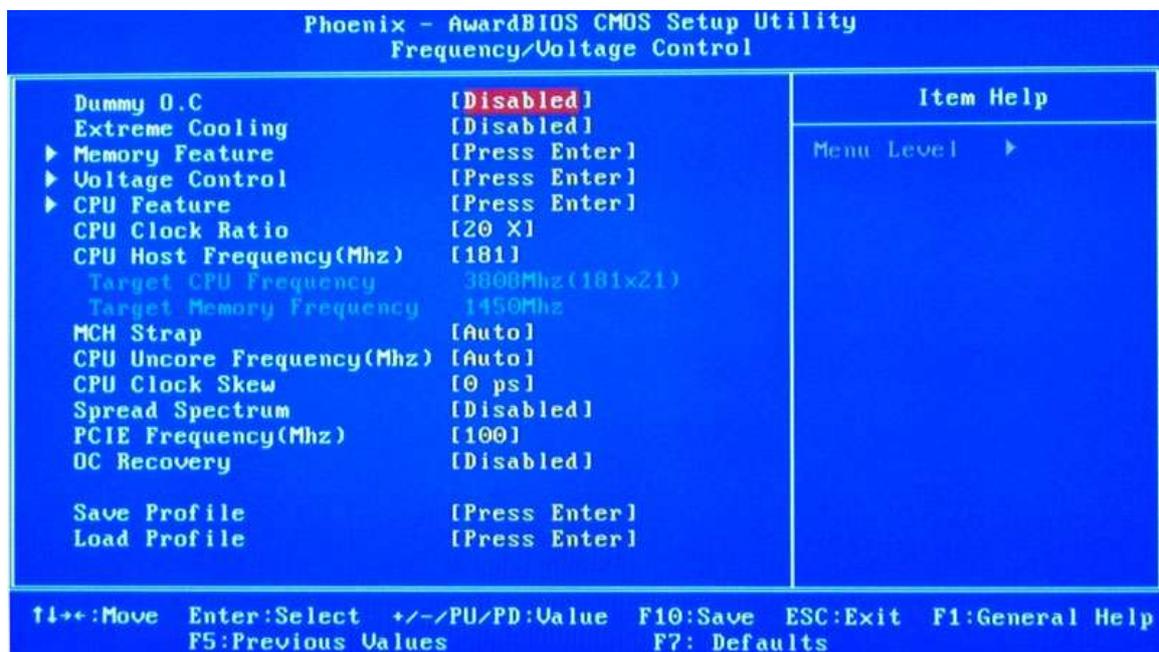
Современные материнские платы всегда оснащаются датчиками, контролирующими рабочие температуры и напряжения основных устройств, а так же скорости вращения вентиляторов системы охлаждения. Все их показатели как раз и отображаются в данном разделе.

Помимо этого в PC Health Status можно управлять режимами работы вентиляторов и настраивать варианты оповещений на случаи возникновения перегрева, остановки кулера или открытия крышки корпуса.

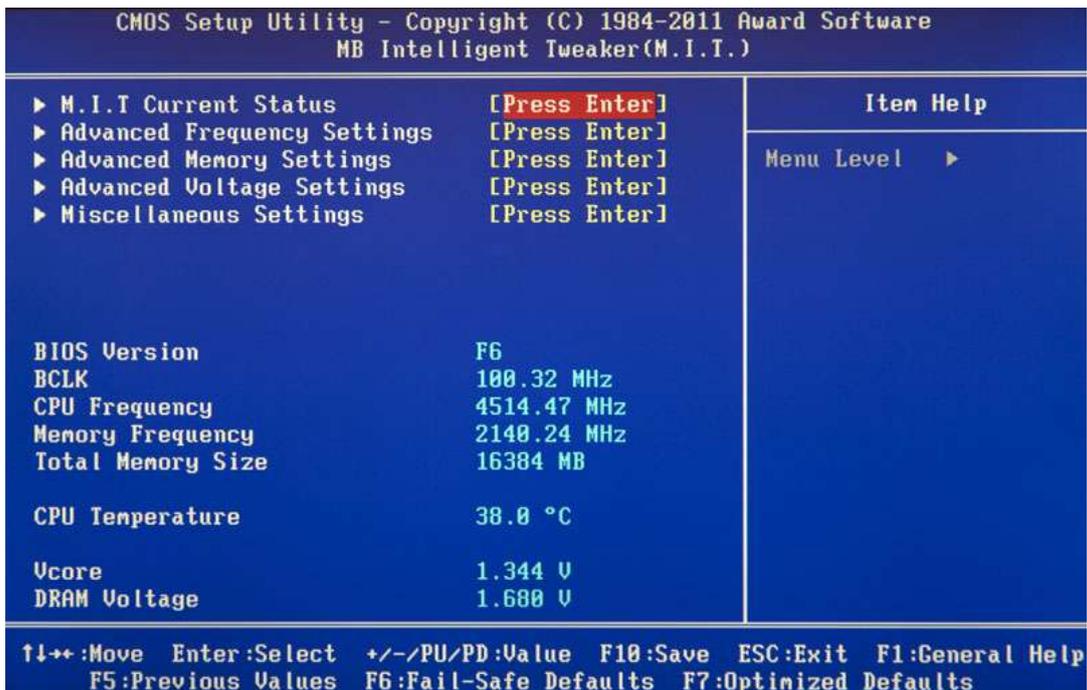


## Frequency/Voltage Control

В данном разделе собраны параметры, отвечающие за установку рабочих частот и значений напряжений для процессора, оперативной памяти, видеокарты и других устройств. По умолчанию все частоты и напряжения имеют рекомендованные значения и настраиваются автоматически, что гарантирует надежную работу системы.



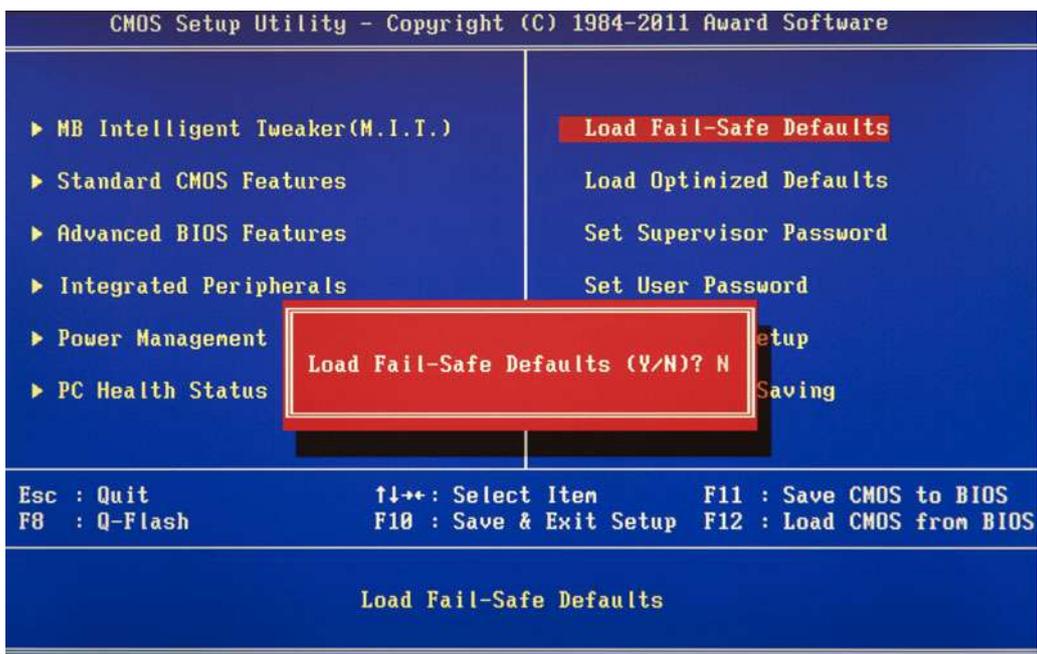
Тем не менее, значение некоторых параметров этого раздела можно изменять вручную. Это дает возможность разогнать процессор, память и прочие компоненты, заставив их работать на повышенных частотах. Только необходимо помнить, что с одной стороны, разгон позволяет увеличить общую производительность системы, а с другой – может вызвать сбои в работе ПК и стать причиной выхода из строя разогнанного железа (например, при установке завышенных значений напряжений). Так что здесь следует быть очень осторожными.



Стоит отметить, что многие крупные производители материнских плат опции по настройке частот и напряжений выносят в специальный раздел с оригинальным названием, например **MB Intelligent Tweaker (M.I.T.)** или **Cell Menu**.

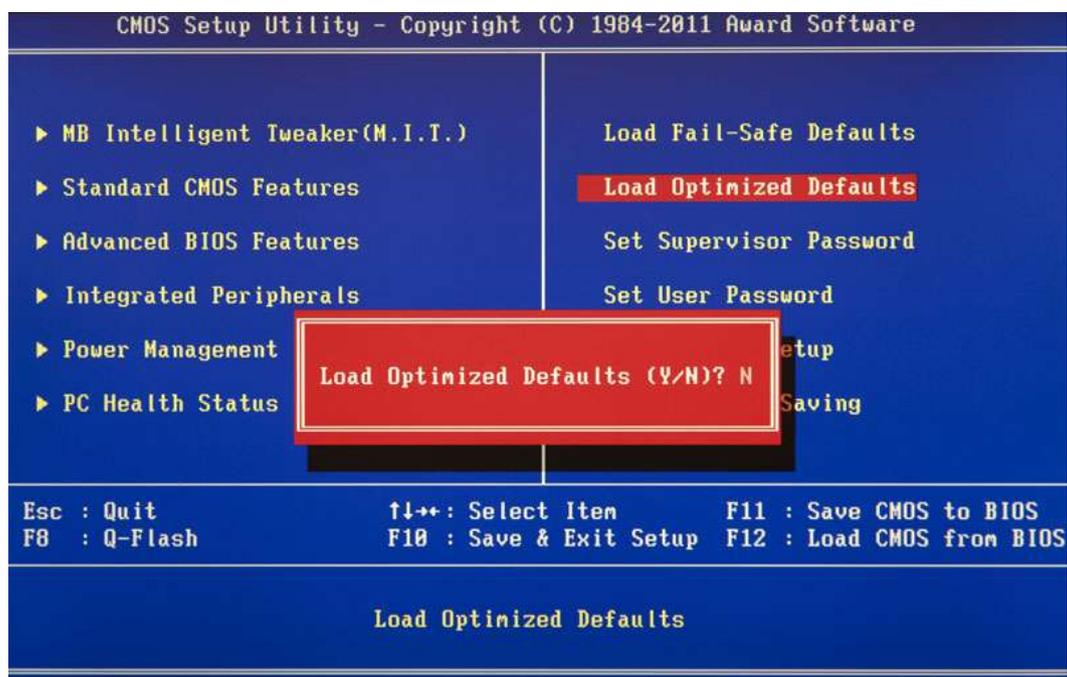
### Load Fail-Safe Defaults

Это не раздел, а команда, сбрасывающая все настройки BIOS к значениям по умолчанию, при которых гарантируется стабильная работа всей системы. После выбора этого пункта перед вами откроется окно, в котором потребуется подтвердить сброс настроек нажатием клавиши «Y».



### Load Optimized Defaults

Команда, устанавливающая значения настроек BIOS таким образом, чтобы бы была обеспечена оптимальная производительность компьютера с сохранением стабильности работы всех его компонентов. При этом параметры, которые подвергаются автоматическому изменению, зависят от модели системной платы и могут различаться.



Однако учтите, что такая оптимизация настроек в некоторых случаях может привести к нестабильной работе системы из-за несовместимости установленного оборудования. Тогда следует вернуться к настройкам по умолчанию при помощи команды **Load Fail-Safe Defaults** и попробовать настроить нужные параметры вручную.

### Set Supervisor Password

Команда, которая позволяет установить, снять или изменить административный пароль, который используется для полного доступа ко всем настройкам BIOS, а так же при загрузке ПК.

### Set User Password

Команда, устанавливающая пользовательский пароль, позволяющий получить доступ к просмотру значений параметров BIOS. То есть большинство настроек будет закрыто для редактирования. Так же данный пароль можно использовать при загрузке компьютера.

### Основные разделы BIOS Setup с горизонтальным главным меню (серый фон)

#### Main

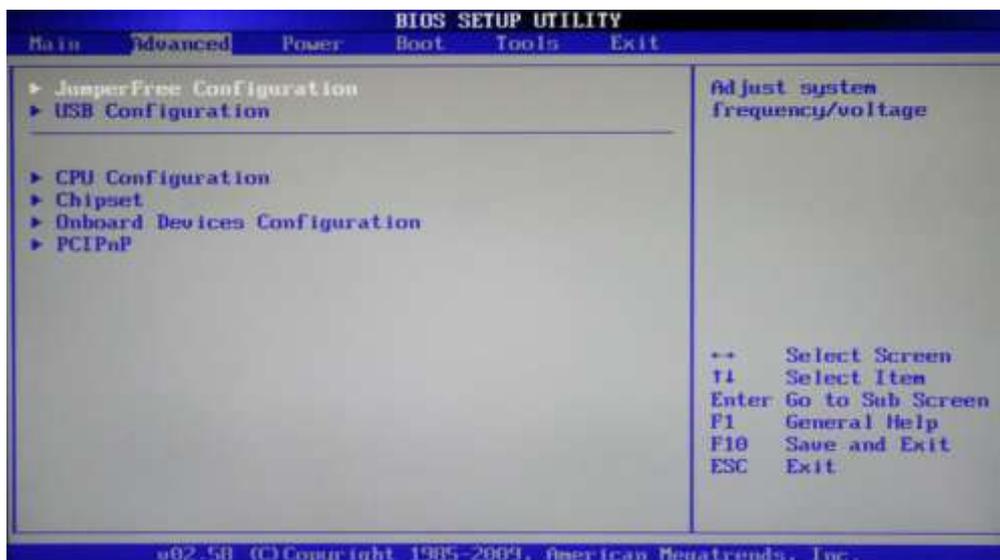
Исходя из названия, по мнению разработчиков, в этом разделе собраны главные настройки BIOS, к которым относятся время и дата, параметры установленных дисковых накопителей и общая системная информация (версия BIOS, модель процессора, объем установленной памяти). Таким образом, *Main* является практически полным аналогом уже знакомого нам раздела *Standard CMOS Futures*.



Как вы, наверное, уже догадались, наиболее востребованной опцией в данном разделе, является настройка системной даты и времени.

## Advanced

Как правило, этот раздел имеет наибольшее количество опций для настройки компонентов и ПК и включает в себя сразу несколько значимых подразделов. Здесь находятся параметры, отвечающие за работу центрального процессора (*CPU Configuration*), оперативной памяти, видеоадаптера, чипсета (*Chipset*), шины передачи данных PCI и технологии Plug and Play (*PnP/PCI Configuration*, *PCI PnP*), встроенных периферийных устройств (*Onboard Device Configuration*), портов USB (*USB Configuration*) и другого оборудования.



Так же в этом разделе можно найти опции разгона, позволяющие вручную задавать значения частот и напряжений процессора, памяти, а так же шины PCI-E. В некоторых случаях, дополнительно пользователям доступна регулировка задержек ОЗУ (тайминги/латентность). Во многих моделях материнских плат, параметры, отвечающие за разгон, выносятся в отдельный подраздел (например, JumperFree Configuration) или даже самостоятельный раздел главного меню (AITweaker, Overclocking или ExtremeTweaker).

Из-за достаточного большого набора компонентов и разнообразия параметров, раздел *Advanced* практически не имеет унифицированной структуры. В зависимости от модели платы и разработчика BIOS, количество подразделов/настроек и их названия могут сильно различаться. Ведь если сравнивать с версией BIOS Setup, которая имеет синий фон, то получается, что в разделе *Advanced* собрано содержимое сразу пяти разделов: *Advanced BIOS Features*, *Advanced Chipset Features*, *Integrated Peripherals*, *Frequency/Voltage Control* и *PnP/PCI Configurations*.

## Power

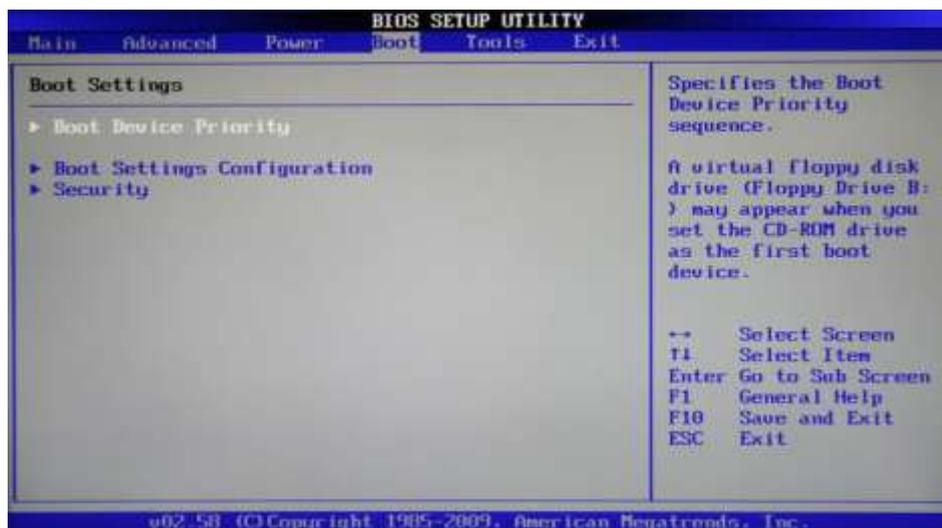
Данный раздел по своему содержанию и сути идентичен разделам *Power Management Setup* и *PC Health Status (H/W Monitor)*.



Здесь находятся параметры, отвечающие за электропитание и энергосбережение ПК, мониторинг рабочих температур и напряжений его основных компонентов, а так же контроля скоростей вращения вентиляторов.

## Boot

Уже из названия видно, что данный раздел отвечает за конфигурирование параметров загрузки компьютера. Именно здесь размещены, востребованные многими пользователями, настройки определения последовательности опроса загрузочных устройств и включение/отключение клавиши «Num Lock» (подраздел *Boot Settings Configuration*).



Во многих случаях раздел *Boot* включает в себя подраздел *Security*, содержащий команды установки, снятия или изменения административного и пользовательского паролей. В некоторых же версиях BIOS Setup параметры управления паролями могут быть вынесены в отдельный одноименный раздел.

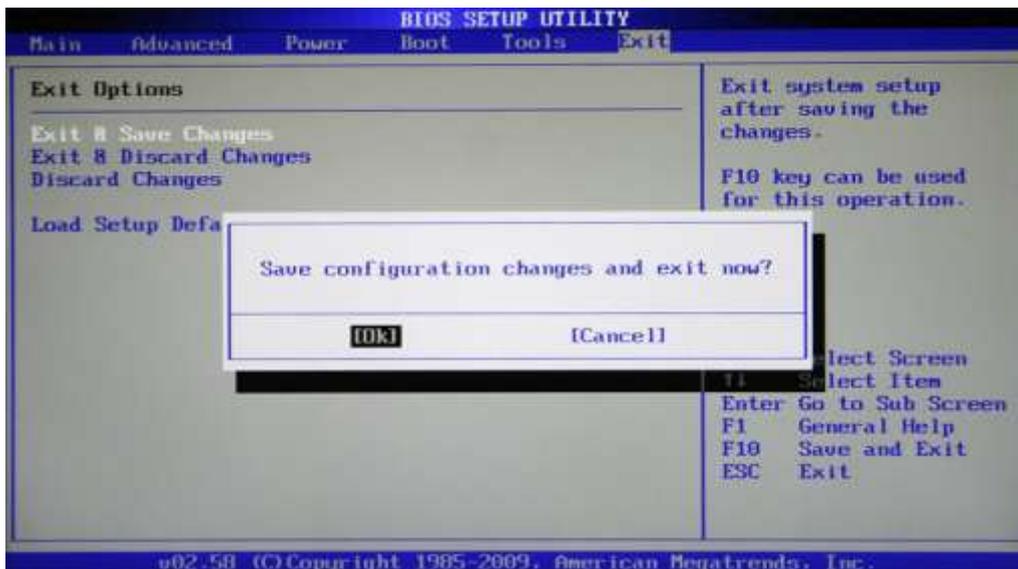
## Tools

Большинство системных плат от популярного производителя ASUS, содержит дополнительный раздел, в который помещаются вспомогательные инструменты для обновления BIOS (*EZ Flash 2*), отключения/включения мини-ОС на ядре Linux (*Express Gate*), создания профилей индивидуальных настроек BIOS (*O.C. Profile*), а так же проверки подключения сетевого кабеля во время загрузки ПК (*AI NET 2*).

## Exit

Этот раздел отвечает за выход из меню настроек BIOS и объединяет в себе такие команды как:

- *Exit & Save Changes* – обеспечивает выход из программы с сохранением всех сделанных вами изменений.
- *Exit & Discard Changes* - осуществляет выход из программы без сохранения всех сделанных изменений.
- *Load Setup Defaults* – возвращает настройки BIOS к значениям по умолчанию (сброс к заводским настройкам).
- *Discard Changes* – отмена сделанных изменений без выхода из программы.

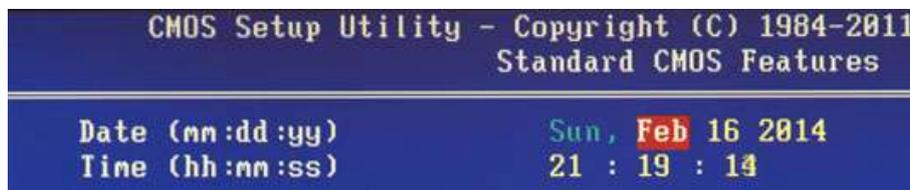


После выбора любой из вышеуказанных команд перед вами появится окно, в котором необходимо подтвердить ее выполнение, нажав клавишу «Y», а затем «Ввод»

### Установка времени и даты

#### **BIOS Setup с синим фоном**

С помощью клавиш со стрелочками перемещаем курсор на раздел *Standard CMOS Futures* и нажимаем «Ввод» («Enter»). Часто этот раздел стоит первым и ничего куда перемещать не надо, но бывают и исключения.



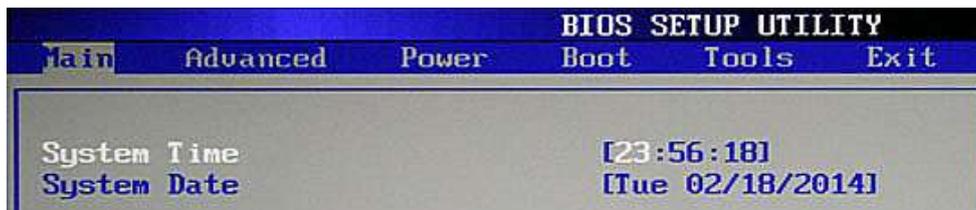
В открывшемся окне с опциями, сверху находим два нужных нам параметра – Date (Дата) и Time (Время). Для перемещения между значениями параметров используйте стрелочки. Для установки значений можно применять как клавиши «+»/«PgUp» или «-»/«PgDn», так и непосредственный ввод цифр с клавиатуры. Для фиксации установленных значений предназначена клавиша «Ввод» («Enter»).

Общий алгоритм действий здесь достаточно прост: устанавливаем курсор на нужное поле (подсвечивается красным), вводите или выбираете его значение и нажимаете «Ввод». Далее переходим на следующее поле и все повторяем до тех пор, пока все параметры не будут установлены.

После того как все значения введены, для сохранения изменений нажимаем клавишу «F10». В открывшемся красном окне вводим букву «Y», нажав на клавиатуре одноименную клавишу. После перезагрузки новые значения времени и даты вступят в силу.

#### **BIOS Setup с серым фоном**

При помощи клавиш «←» и «→» выбираем раздел *Main*, хотя в большинстве случаев этого делать не придется, так как он практически всегда располагается первым и открывается по умолчанию сразу после входа в BIOS Setup.



Находим в этом разделе параметры System Date (Системная дата) и System Time (Системное время) и перемещаем туда курсор, используя клавиши «↓» и «↑». Далее для ввода значений используем либо непосредственно клавиши с цифрами, либо клавиши «+» и «-». Для перемещения между полями внутри одного параметра здесь предназначена клавиша «Tab». После ввода требуемого значения нажимаем «Ввод».

Как и в предыдущем случае, чтобы сохранить внесенные изменения, нажимаем клавишу «F10», а затем «Y». Сразу после этого произойдет перезагрузка ПК, и новые параметры вступят в силу.

### Смена загрузочного устройства

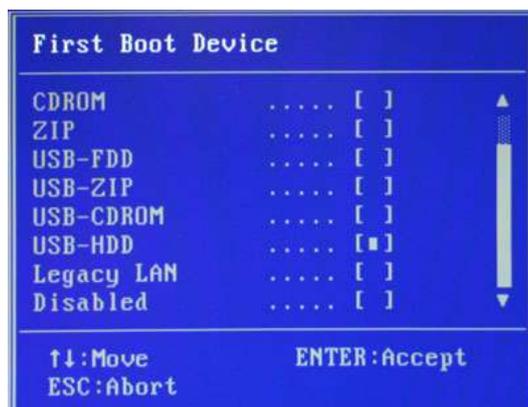
При установке операционной системы или проведении профилактических работ с уже установленной ОС, часто необходимо обеспечить загрузку компьютера не с жесткого диска, а с оптического носителя, USB-флэшки или какого-либо другого устройства хранения данных. Поэтому одной из самых востребованных задач, ради которой рядовым пользователям приходится «лезть» в настройки BIOS, является необходимость смены загрузочного устройства.

### **BIOS Setup с синим фоном**

После открытия программы BIOS Setup, перемещаем стрелочками курсор на раздел *Advanced BIOS Features* и нажимаем «Ввод».



Клавишей «↓» переходим к параметру *First Boot Device* (Первое загрузочное устройство) и вновь ждем «Ввод».



Далее перед вами откроется окно со списком устройств, которые можно выбрать в качестве загрузочных. Если планируется запуск ПК с оптического диска, то выбираем при помощи стрелок значение CDROM и далее как обычно «Ввод». Если необходимо загрузиться с флэшки или внешнего портативного диска, то выбираем опцию USB-HDD. Таким же образом можно выбрать второе и третье загрузочные устройства (Second Boot Device и Third Boot Device).

При этом стоит учесть, что если в компьютере установлено сразу несколько жестких дисков или твердотельных накопителей, содержащих систему и являющихся загрузочными, то для указания последовательности их опроса, предназначен специальный пункт Hard Disk Boot Priority.

Для того, чтобы все сделанные вами настройки вступили в силу, не забудьте нажать клавишу «F10», затем «Y» и наконец «Ввод».

## BIOS Setup с серым фоном

После открытия окна настроек BIOS с помощью клавиши «←→» выбираем пункт *Boot* и нажимаем «Ввод». Далее вас может ожидать два варианта, в зависимости от версии BIOS.



В первом случае вы увидите сразу список назначения загрузочных устройств. Обозначаются они как 1st, 2nd и 3rd Boot Devices (соответственно первое, второе и третье загрузочные устройства). Перемещение по списку производится клавишами «↑↓», выбор значений (HDD, CDROM, USB, Removable) – клавишами «Ввод» или «+/-».



Во втором случае раздел *Boot* будет содержать несколько подразделов, среди которых в данной ситуации нас интересует пункт *Boot Device Priority*. Перемещаем на него курсор, и нажимаем «Ввод». Сразу после этого перед вами откроется окно со списком загрузочных устройств, выбор которых осуществляется точно таким же образом, как было описано выше.

Владельцем нескольких накопителей стоит обратить внимание на подраздел *Hard Disk Drives*. Именно в нем осуществляется выбор приоритетного загрузочного диска среди установленных в компьютере винчестеров. Если же у вас установлено несколько оптических приводов, то в этом случае выбор среди них приоритетного устройства может быть организован в подразделе *CDROM Drives*.

После завершения настроек, остается нажать клавишу «F10», а затем «Ввод», чтобы сохранить внесенные изменения.

#### **Порядок выполнения работы:**

После включения компьютера нажать и удерживать клавишу Del, для того чтобы запустилась программа CMOS Setup BIOS

1. Установить настройки BIOS по умолчанию
2. Установить текущие время и дату
3. Установить дисковод
4. Установить последовательность загрузки: жесткий диск, CD-ROM, дисковод
5. Отключить опрос дисковода при включении компьютера
6. Включить внутренний и внешний кэш
7. Включить автонабор нажатой клавиши 30 симв./сек.
8. Включить задержку перед автонабором 250 мсек.
9. Включить функцию ACPI
10. Отключить второй канал IDE
11. Отключить контроллер USB
12. Отключить последовательный порт
13. Сделать автоматическое определение параметров жесткого диска

После выполнения работы показать результат преподавателю.

#### **Контрольные вопросы**

1. Для чего нужна программа CMOS Setup BIOS ?
2. Какие параметры можно изменить с помощью этой программы?
3. Для чего необходимо устанавливать последовательность устройств, с которых может осуществляться загрузка операционной системы?
4. Какой параметр напрямую влияет на производительность системы?

5. Если какие – либо параметры BIOS настроены неправильно, как вернуться к первоначальным заводским настройкам?
6. Какие способы защиты от несанкционированного доступа можно установить средствами BIOS?
7. Как в BIOS отключить дисковод или жесткий диск?

**Форма представления результата:**

Отчет о проделанной работе:

1. Тема и цель работы
2. Описание хода проведения работы, ошибок, возникших в результате выполнения работы, аппаратных характеристик компьютера
3. Ответы на контрольные вопросы

**Критерии оценки:**

Оценка «отлично» ставится, если задание выполнено верно.

Оценка «хорошо» ставится, если ход выполнения задания верный, но была допущена одна или две ошибки, приведшие к неправильному результату.

Оценка «удовлетворительно» ставится, если приведено неполное выполнение задания.

Оценка «неудовлетворительно» ставится, если задание не выполнено.

## Тема 2.5 Компоненты системного блока

### Лабораторное занятие № 9

Сбор информации об установленном процессоре. Тестирование процессора на производительность и отказоустойчивость

**Цель:** 1. Изучить основные характеристики центрального процессора.

2. Ознакомиться и произвести измерение быстродействия процессора с помощью тестовых программ.

**Выполнив работу, Вы будете:**

**уметь:**

- устанавливать и модернизировать центральный процессор;
- тестировать процессор на производительность и отказоустойчивость;
- выполнять поиск необходимой информации.

**Материальное обеспечение:**

Методические указания для выполнения лабораторных работ, компьютер.

**Задание:**

1. Ознакомиться и получить навыки работы по установке и модернизации центрального процессора.
2. Ознакомиться и получить навыки измерения быстродействия процессора с помощью тестовых программ

**Краткие теоретические сведения:**

1. Микросхема, реализующая функции центрального процессора персонального компьютера, называется микропроцессором. Обязательными компонентами микропроцессора является арифметико – логическое устройство (АЛУ) и устройство управления (УУ).

АЛУ отвечает за выполнение арифметических и логических операций, а УУ координирует работу всех компонентов и выполнение процессов, происходящих в компьютере.

Процессор компьютера предназначен для обработки информации. Каждый процессор имеет определенный набор базовых операций (команд), например, одной из таких операций является операция сложения двоичных чисел.

Технически процессор реализуется на большой интегральной схеме, структура которой постоянно усложняется, и количество функциональных элементов (типа диод или транзистор) на ней постоянно возрастает (от 30 тысяч в процессоре 8086 до 5 миллионов в процессоре Pentium II).

Под тактом мы понимаем промежуток времени, в течение которого может быть выполнена элементарная операция. Тактовую частоту можно измерить и определить ее значение. Единица измерения частоты - МГц – миллион тактов в секунду.

Другой характеристикой процессора, влияющей на его производительность, является разрядность. В общем случае производительность процессора тем выше, чем больше его разрядность. В настоящее время используются 18,16, 32- и 64-разрядные процессоры, причем практически все современные программы рассчитаны на 32- и 64-разрядные процессоры.

Часто уточняют разрядность процессора и пишут, например, 16/20, что означает, что процессор имеет 16-разрядную шину данных и 20-разрядную шину адреса. Разрядность адресной шины определяет адресное пространство процессора, т.е. максимальный объем оперативной памяти, который может быть установлен в компьютере.

В первом отечественном персональном компьютере «Агат» (1985 г.) был установлен процессор, имевший разрядность 8/16, соответственно его адресное пространство составляло 64 Кб. Современный процессор Pentium II имеет разрядность 64/32, т.е. его адресное пространство составляет 4 Гб.

Производительность процессора является интегральной характеристикой, которая зависит от частоты процессора, его разрядности, а также особенностей архитектуры (наличие кэш-памяти и др.). Производительность процессора нельзя вычислить, она определяется в процессе тестирования, т.е. определения скорости выполнения процессором определенных операций в какой-либо программной среде.

2. Программа CPU-Z бесплатная, скачать ее можно на сайте разработчика. Автор программы - Franck Delattre. Программа регулярно обновляется.

После распаковки и запуска CPU-Z открывается окно с восемью вкладками по основным параметрам центрального процессора и памяти компьютера (рис.1):



Рис.1. Окно параметров процессора программы CPU-Z

При сворачивании в лоток программа выводит в нем текущее значение частоты процессора; на закладке SPD, кроме данных об установленных модулях ОЗУ, можно узнать о занятых слотах (рис.2). Таким образом, вы можете, не вскрывая компьютер, определить возможность наращивания памяти, что особенно удобно для ноутбуков.

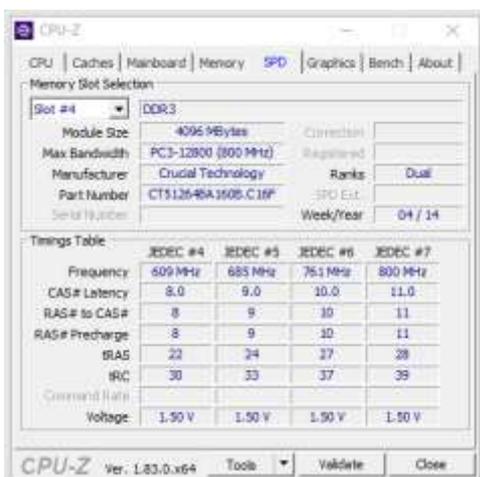


Рис.2. Окно основных параметров модулей памяти

2. Программа AIDA64. Утилита FinalWire Ltd. для тестирования и идентификации компонентов персонального компьютера под управлением операционных систем Windows, предоставляющая детальные сведения об аппаратном и программном обеспечении. Является последовательницей Everest, принадлежащей с 2004 по 2010 года компании Lavalys, которая в свою очередь основана на AIDA32.

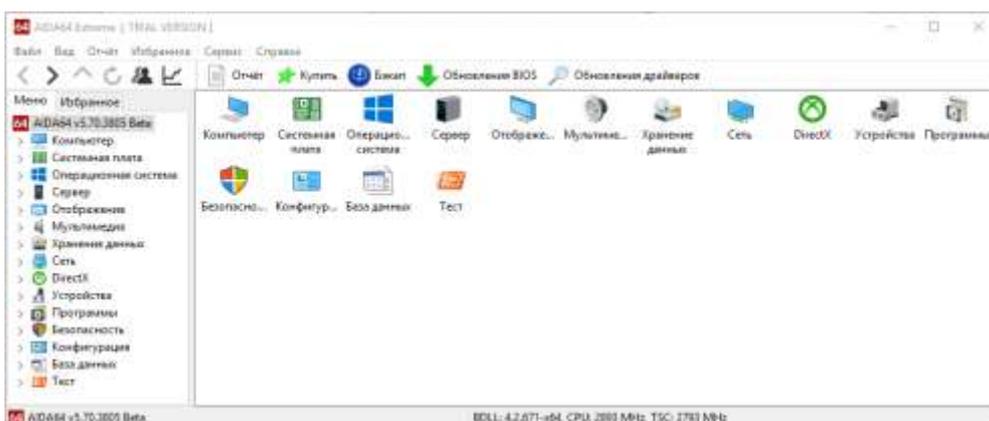


Рис.3. Окно программы AIDA64

Данная программа позволяет получить в мельчайших подробностях всю информацию обо всех имеющихся комплектующих и установленных программных продуктах. Кроме того, в ней имеются несколько бенчмарк-тестов:

- **Чтение из памяти** — этот тест использует максимум оперативной памяти, доступной для чтения. Код теста написан на ассемблере и оптимизирован для всех популярных процессоров от AMD и Intel; при этом используются не только стандартные команды x86, но и наборы инструкций MMX, 3DNow!, SSE и SSE2. Во время теста непрерывно производится прямое обращение к памяти. Данные читаются из блока размером 16 Мб;
- **Запись в память** — этот тест использует максимум оперативной памяти, доступной для записи. Код теста также написан на ассемблере и оптимизирован для всех популярных процессоров от AMD и Intel; в том числе с использованием стандартных команд x86 и наборов инструкций MMX, 3DNow!, SSE и SSE2. Во время теста запись производится непрерывно, путем прямого обращения к памяти. Данные записываются в блок размером 16 Мб;

- **Копирование в памяти** — этот тест использует максимум оперативной памяти, доступной для копирования. Код теста также написан на ассемблере, оптимизирован для всех процессоров AMD и Intel с использованием стандартных команд x86 и наборов инструкций MMX, 3DNow!, SSE и SSE2. Во время теста производится запись блока памяти размером 8 Мб в другой блок памяти размером 8 Мб. Операции выполняются непрерывно путем прямого обращения к памяти;
- **Задержка памяти** — этот тест измеряет типичную задержку при чтении данных из системной памяти. Время задержки памяти означает интервал времени между запуском команды на чтение и их поступлением в регистры процессора. Код теста написан на ассемблере и использует 1 Мб данных, к которым обращается с использованием прямого доступа к памяти. При работе теста используются только стандартные команды x86, один поток на одном ядре процессора;
- **CPU Queen** — простой целочисленный тест процессора, фокусирующийся на возможностях прогнозирования ветвления при решении классической задачи «проблемы ферзя» на 100-клеточной доске;
- **CPU PhotoWorxx** — целочисленный тест для оценки выполнения различных задач общего характера, выполняемых при цифровой обработке фотоизображений. Во время теста с очень большим изображением в палитре RGB выполняются следующие операции: заливка, отражение, поворот на 90° по и против часовой стрелки, заполнение изображения точками со случайным цветом, перевод в оттенки серого, получение негатива. Этот тест сильно загружает не только арифметические узлы процессора, но и подсистему доступа к памяти. Фактически он генерирует огромное количество обращений к памяти и выявляет неэффективность систем, содержащих более двух ядер;
- **CPU ZLib** — целочисленный тест, измеряющий производительность процессора и памяти при выполнении операций сжатия с использованием общедоступной библиотеки Zlib;
- **FPU Julia** — измеряет производительность при выполнении операций одинарной точности (32 бита) с плавающей запятой для расчета нескольких фрагментов популярного фрактала Julia. Код теста написан на ассемблере и оптимизирован для всех популярных процессоров от AMD и Intel; при этом используются не только стандартные команды x87, но и расширенные наборы инструкций MMX, 3DNow!, SSE;
- **FPU Mandel** — измеряет производительность при выполнении операций двойной точности (64 бита) с плавающей запятой для расчета нескольких фрагментов популярного фрактала Мандельброта. Код теста также написан на ассемблере и оптимизирован для всех популярных процессоров AMD и Intel с использованием наборов инструкций x87 и SSE2;
- **FPU SinJulia** — измеряет производительность при выполнении операций расширенной точности (80 бит) с плавающей запятой для расчета одного фрагмента модифицированного фрактала Julia. Код теста также написан на ассемблере и оптимизирован для процессоров AMD и Intel с использованием тригонометрических и экспоненциальных инструкций набора x87.

3. **OCCT** (от «OverClock Checking Tool) — программа для тестирования состояния центрального процессора.

Характеристики программы:

- 3 различных типа тестов
  - CPU = оперативная память не тестируется, замеряется нагрев процессора CPU, регистрируются ошибки процессора.

- RAM = осуществляет максимальное выявление ошибок, тестируется вся цепочка процессор-чипсет-оперативная память.
- Смешанный = тестирование процессора и оперативной памяти чередуются.
- 3 режима тестирования
  - Автоматический = один смешанный тест.
  - Настраиваемый (бесконечный) = бесконечная цепочка тестов по вашему выбору.
  - Настраиваемый (фиксированный) = проверка с типом теста по вашему выбору (длительность вы также определяете самостоятельно).
- Интегрированная система мониторинга и получения системной информации, разработанная на движке таких программ, как CPU-Z и Hwmonitor.
- Внешняя программа мониторинга поддерживает множество плагинов (например, Everest, Speedfan и многие другие).
- Графики иллюстрируют температуру и напряжение вашего компьютера.



Рис.4. Окно программы ОССТ

При помощи программы ОССТ мы сможем провести тест следующих компонентов ПК:

Процессор, Оперативная память, и Материнская плата – CPU:OCST

Процессор – CPU:LINPACK

Видеокарту – GPU:3D

Блок питания – POWER SUPPLY

Программа ОССТ при прохождении теста даёт максимальную нагрузку на тестируемые компоненты ПК. И если тестирование закончилось без ошибок, то ПК и система охлаждения полностью исправны.

В окне настроек самое главное проставить температуры, при достижении которых тест будет остановлен, это необходимо для предотвращения выхода из строя какого либо узла от перегрева.

Если вашему ПК 5 и более лет, то выставляйте температуру 80°C. Более позднего выпуска детали очень чувствительны к перегреву.

Рекомендованное автором программы время для прохождения теста -30 минут, но самое оптимальное время – это 1 час. Если с железом есть проблемы, то за 1 час они точно покажут себя!



Рис.5. Окно настроек программы ОССТ

Режим тестирования:

Большой набор – Тестируются на ошибки Процессор, Оперативная память, и Материнская плата (чипсет).

Средний набор – Тестируются на ошибки Процессор и Оперативная память.

Малый набор – Тестируется на ошибки только Процессор.

Если же при прохождении теста всё-таки возникли ошибки, то: в окошке программы будет описана причина остановки теста и если остановкой теста послужила высокая температура, то стоит проверить, и почистить от пыли систему охлаждения, если тест прерван из-за ошибки аппаратной части какого-либо узла ПК (процессора, видеокарты, памяти, материнской платы) то готовьтесь к скорой замене этой составляющей. Если ошибка вышла при тестировании БП (блока питания), то это ещё не говорит о его неисправности, скорее всего БП просто не вытягивает всю систему при максимальных нагрузках. Жить с этим можно, работать компьютер будет, только при включении, например, какой-нибудь мощной игры, БП опять может не вытянуть систему. Так что задуматься о замене БП на более мощный всё же стоит.

### Порядок выполнения работы:

1. Заполнить следующие параметры:

1. ФИО студента
2. Тип процессора
3. Частота (МГц)
4. Множитель
5. Мощность (Вт)
6. Кодовое имя (для процессоров Intel)
7. Наборы инструкций
8. Тип разъема
9. Техпроцесс
10. Напряжение ядра
11. Количество ядер и потоков

12. Кэш-память всех уровней
13. Модель материнской платы
14. Чипсет
15. Разрядность шины данных
16. Разрядность шины адреса
17. Адресное пространство

2. Выполнить тестирование процессора: **CPU Queen, CPU PhotoWorxx, CPU ZLib, CPU AES, FPU Julia, FPU Mandel, FPU SinJulia**. Результаты записать в тетрадь.

3. Выполнить тестирование процессора в программе ОССТ. В настройках теста указать максимальную температуру не больше 80°C. Указать время теста 30 мин. Результаты записать в тетрадь.

**Форма представления результата:**

Отчет о проделанной работе.

**Критерии оценки:**

Оценка «отлично» ставится, если задание выполнено верно.

Оценка «хорошо» ставится, если ход выполнения задания верный, но была допущена одна или две ошибки, приведшие к неправильному результату.

Оценка «удовлетворительно» ставится, если приведено неполное выполнение задания.

Оценка «неудовлетворительно» ставится, если задание не выполнено.

## Тема 2.6 Запоминающие устройства ЭВМ

### Лабораторное занятие № 10

Тестирование оперативной памяти. Тестирование НЖМД

**Цель:** научиться тестировать компоненты системного блока. Уметь запускать тестовые программы, делать выводы об исправности компонентов. Разбивать на логические разделы и форматировать жесткий диск.

#### Выполнив работу, Вы будете:

##### *уметь:*

- тестировать компоненты системного блока;
- анализировать работоспособность компонентов;
- понимать тексты на профессиональную тему.

#### Материальное обеспечение:

Методические указания для выполнения лабораторных работ, компьютер.

#### Задание:

1. Выполнить тестирование оперативной памяти
2. Выполнить тестирование жесткого диска
3. Выполнить разбивку жесткого диска
4. Выполнить форматирование жесткого диска

#### Краткие теоретические сведения:

##### Общие сведения

#### Программа Memtest86+ v1.65

Полезная утилита для тестирования оперативной памяти компьютера. Причём, тестирование осуществляется из "чистой среды", а не из Windows. При инсталляции Memtest86 создаётся загрузочная дискета или загрузочный CD, загрузившись с которых, можно начать тестирование оперативной памяти. Программа известна высокой достоверностью результатов, но и достаточно длительным временем, необходимым для тестирования. Работа происходит циклически, все тесты повторяются неограниченное количество раз, пока пользователь не отменит их соответствующей командой. Такой подход способствует тому, что можно основательно протестировать оперативную память. Обычный тест проходит за 20-30 минут, если же вы хотите основательно протестировать память, то погоняйте утилиту хотя бы сутки. Ошибок быть не должно. Регулярные обновления позволяют использовать утилиту на новейших аппаратных платформах без опасений о потенциальных проблемах совместимости.

#### Bad-block

Под bad-блоком понимается обычно конкретный участок диска, нормальная работа с которым не гарантируется или невозможна вовсе. На таких участках может содержаться различная информация; это могут быть данные пользователя или служебная информация. Появление таких секторов может быть обусловлено разными причинами, в одном случае такие секторы можно восстановить, в другом нельзя, в одном нужно использовать одни методы лечения и переназначения в другом другие.

Технология изготовления пластин была несовершенная тогда, и несовершенна сейчас. Не существует методов создания идеальной поверхности, не содержащей ни одного плохого блока, вопреки встречающемуся мнению, что с завода винчестер поставляется без них. С

ростом объема дисков росло количество сбойных секторов при выходе с завода, и, понятно, что только до определенного момента процедура их регистрации в FAT могла выполняться в ручную, нужно было найти способ маркировать бэды, даже не смотря на то, что не известно, какая файловая система будет использоваться. Изобретение транслятора позволило решить эти проблемы. На винчестере выделялась специальная защищенная область, куда записывался транслятор, в котором устанавливалось соответствие каждого логического блока непрерывной цепочке и реального физического адреса.

Если вдруг на поверхности обнаруживался сбойный блок, то он просто пропускался, а данному логическому блоку присваивался адрес следующего физически доступного блока. Транслятор считывался с диска при включении. Создание его выполнялось (и выполняется) на заводе, и именно поэтому, а не от того, что производителем применяется какая - то технология, новые диски как бы не содержат бэд-блоков. Физические параметры оказались скрыты (и они слишком рознились, так как у фирм оказались развязаны руки в производстве своих собственных форматов низкого уровня, и пользователя это не заботило), дефекты помечались на заводе, универсальность увеличилась.

В зависимости от природы происхождения все BAD-блоки можно подразделить на две большие группы: логические и физические.

**Физические:** Дефекты поверхности могут быть связаны: с постепенным износом магнитного покрытия дисков; просочившимся через фильтр мельчайшим частицам пыли, кинетическая энергия которых, разогнанных внутри накопителя до колоссальных скоростей, оказывается достаточной для повреждения поверхности, результатом механических повреждений при ударе, при котором из поверхности могут выбиться маленькие частицы, которые потом в свою очередь также будут выбивать другие частички, и процесс пойдет лавинообразно (такие частицы будут скатываться с пластин под действием центробежных сил, но значительно дольше и тяжелее, так как будут удерживаемы силами магнитного притяжения). Это еще чревато тем, что с ними будет происходить столкновение головки, парящей на очень малой высоте, что вызовет ее нагрев и ухудшение рабочих характеристик - будут возникать искажения сигнала (результат - ошибки чтения). К физическим бэдам можно также отнести сбойные сектора, появление которых обусловлено неисправностями электронной или механической части накопителя, например обрыв головок, серьезные механические повреждения в результате удара - заклинивание катушки позиционера или дисков, смещение дисков. В целом же для этого вида повреждений характерен катастрофический характер.

**Логические:** Можно выделить следующие категории:

1. Самый простой случай: ошибки файловой системы. Сектор помечен в FAT как сбойный, но на самом деле таковым не является (раньше таким приемом пользовались некоторые вирусы, когда на небольшом объеме винчестера требовалось найти себе укромное местечко, не доступное простыми средствами).

2. Неисправимые логические бэды - характерны для старых винчестеров использующих запись с полями идентификаторов. Если у вас такой диск, то вполне можете с ними столкнуться. Обусловлено неверным форматом физического адреса, записанного для данного сектора, ошибка контрольной суммы для него и т.д. Соответственно, невозможно обращение к нему. На самом деле они восстанавливаемы, но на заводе. Поскольку сейчас используется технология записи без полей идентификаторов, то эту разновидность можно считать неактуальной.

3. Исправимые логические бэды. Не так уж редко встречаемый, особенно на некоторых типах накопителей тип сбойных блоков. Происхождением в основном обязаны ошибкам записи на диск. Чтение произвести с такого сектора не удастся, так как обычно в нем ЕСС код не соответствует данным, а запись обычно невозможна, так как перед записью осуществляется предварительная проверка подлежащего записи пространства, и поскольку с ней уже обнаружены проблемы, запись в данную область отклоняется. Т.е. получается блок

невозможно использовать, хотя физически поверхность, им занимаемая в полном порядке. Дефекты подобного рода могут быть иногда вызваны ошибками в микропрограмме винчестера, могут быть спровоцированы программным обеспечением или техническими причинами.

4. Появления на винчестерах BAD-блоков этого вида обязано особенностям технологии производства: никогда не существует двух абсолютно одинаковых устройств, какие-то их параметры непременно отличаются. При подготовке винчестеров на заводе, для каждого определяется набор параметров, обеспечивающих наилучшее функционирование данного конкретного экземпляра, так называемые адаптивы. Эти параметры сохраняются, и в случае если они каким то загадочным образом оказываются повреждены, то результатом может быть полная неработоспособность диска, нестабильная его работа или большое количество сбойных секторов появляющихся и исчезающих то в одном, то в другом месте. В домашних условиях с этим сделать ничего нельзя, но все можно настроить на заводе или в сервисном центре.

### Программа MHDD

Программа MHDD является очень удобным инструментом для диагностики и ремонта винчестеров. Программа является универсальной, она работает со всеми накопителями с интерфейсом АТА независимо от производителя и модели. Работать программа способна из под чистого DOS, и прямо из Windows. Утилита способна запускаться с любого, не защищенного от записи носителя на котором есть немного места для записи временных файлов и протоколов работы (минимально нужно 70 кбайт, рекомендуется 20 Мбайт). Не используйте программу с того же диска, который диагностируете и лечите.

После запуска при наличии более чем одного накопителя в системе предлагается выбрать тот, с которым мы хотим работать.

```
[ Drive parameters - PRESS F2 to DETECT ] [ Current position ]
-- -- -- -- -- Device Select -- -- -- -- --
-[key]-----[device info]-----
port 1F0h
1. [ST34311A ]
2. [ ]
port 170h
3. [ST320011A ]
4. [ ]
port 100h
5. [ ]

No PCI controllers found.
-----
Enter HDD number [3]:

! (c) maysoft aka Dmitry Postrigan, http://mhdd.net | 2.9 | | 18:14:55
```

На экран будет выведена подсказка об основных командах программы.

```

ERR INDX CORR DREQ DRSC WRFT DRDY BUSY          AMNF TENF ABRT IDNF UNCR BBK
[ Drive parameters - PRESS F2 to DETECT ] [          Current position          ]

■ This version is WITHOUT PCI bus support
  for Windows NT compatible

*****
* Online HELP *
*****
* id      hpa    pwd    fdisk  rx      *
* scan   fu1st  unlock i      randombad *
* aerase rhpa   dispwd cx    fuj     *
* erase  r      ff     erase  makebad *
* batch  cls   aam   pciscan *
* rpm    nhpa  init  tof    wdrd wdrw wdrw *
* smart  port  stop  wait   wdfmtold *
* killfuj akillfuj fuckfuj ibmø  wdwcp wdrsp *
*        wdfmt wdfmtsa *
* QU     FUJ          wdm     *
*****[ Press <F2> ]*****
MHDD>
| (c) maysoft aka Dmitry Postrigan, http://mhdd.net | 2.9 | | 17:56:49

```

Инициализировав выбранный накопитель нажатием F2 получаем исчерпывающую о нем информацию: название модели; логические геометрию диска (логическое число цилиндров/головок/секторов на дорожке); серийный номер; версию микрокода; число адресуемых через LBA блоков; поддерживаемые функции ATA и максимальный поддерживаемый режим; состояние SMART (включена ли); состояние системы защиты (включен/выключен аппаратный пароль); полный размер винчестера; результат прохождения основных тестов. Часть этой информации отображается во второй сверху строке на экране и видна при работе.

```

ERR INDX CORR DREQ DRSC WRFT DRDY BUSY          AMNF TENF ABRT IDNF UNCR BBK
CE 8944] HE15] SE63] [ 8452000] [ --LBA-- ]--SE ]--HE ] CE ]

ST34311A 8944/15/63
SN:5BFGY2BC FW:8.01 LBAs:8452000
Support: DLCode LBA HPA DMA (UDMA2, MUDMA2)
SMART: Enabled
Security: high, OFF. Size = 4126Mb
Device Reset... OK
Setting Drive Parameters... OK.
Recalibrate... OK
MHDD>
| (c) maysoft aka Dmitry Postrigan, http://mhdd.net | 2.9 | | 17:57:04

```

Кроме того, о состоянии диска и его занятии в данный момент времени сигнализируют индикаторы в самой верхней строке. Они несут следующую информацию:

- BUSY** - накопитель занят и на команды не реагирует;
- WRFT** - ошибка записи;
- DREQ** - накопитель жаждет обменяться данными с внешним миром;

**ERR** - в результате какой-либо операции возникла ошибка. Когда загорается этот бит, обратите внимание на правую верхнюю часть экрана. Там будет отображен тип последней ошибки: (действительно только при зажженной лампочке "ERR"):

**AMNF** - Adress Mark Not Found - Обращение к какому-то конкретному сектору не удалось. Вероятной причиной является повреждение этого сектора, повреждение его полей идентификации (упоминание об этом есть здесь). Но сразу после включения накопителя, как раз наоборот, свидетельствует об отсутствии проблем и сообщает об успешном выполнении внутренней диагностики .

**TONF** - Track 0 Not Found - не найден нулевой трек.

**ABRT** - Abort, команда отвергнута;

**IDNF** - Sector ID Not found;

**UNCR** - Uncorrectable Error - Ошибка не скорректированная кодом ECC. Вероятно, в данном месте имеет место быть логический BAD-блок.

Кроме этих вверху могут гореть еще два индикатора - PWD - сигнализирующая об установленном аппаратном пароле, и HPA - если размер накопителя был изменен с помощью команды HPA (для скрытия BAD-блоков в конце диска).

Для обнаружения BAD-блоков в программе есть возможность проверки поверхности жесткого диска. Причем полученная информация будет очень подробно отражать состояние поверхности. Вызов осуществляется консольной командой SCAN. Проверить можно как весь диск целиком, так и какой-то отдельный участок поверхности. После набора команды SCAN появляется следующего вида меню:

```
ERR INDX CORR DREQ DRSC WRFT DRDY BUSY AMNF TONF ABRT IDNF UNCR BBK
C[ 8944] H[15] S[63] [ 8452079] [ --LBA-- ]-S[ ]-H[ ] C[ ]

ST34311A 8944/15/63
SN:5BF0Y2BC FW:8. Scan Parameters; [SPACE or ENTER]-change
Support: DLMCode LB Scan in : LBA
SMART: Enabled Starting CYL : 0
Security: high, OFF Starting LBA : 0
Device Reset... OK LOG : OFF
Setting Drive Param Remap : OFF
Recalibrate... OK Ending CYL : 8943
Ending LBA : 8452079
Timeout(sec) : 30
Advanced log : OFF
Standby after scan : OFF
Loop the test/repair : OFF
Erase wAITS : OFF
[A,D,S,W]-move; [CTRL+ENTER,F4]-finish

-- > Press the arrow keys for navigate on HDD during scan.
```

Верхняя строка - это режим работы с винчестером. По умолчанию теперь там стоит LBA, менять что-то вряд ли понадобится. В строках Starting Cyl/LBA, Ending Cyl/LBA выставляются начальный и конечный блок для LBA трансляции и начальный/конечный цилиндр для CHS (для проверки конкретного участка). Значения по умолчанию соответствуют полной поверхности. Можно включить переназначение сбойных секторов при обнаружении оных (функция Remap). Timeout - время отводимое накопителю для выполнения операции. Результаты проверки можно занести в протокол. При тесте поверхности на экран будут выводиться квадратики (один такой квадратик равен 255 секторам (при тестировании в режиме LBA либо числу секторов в строке параметров HDD (обычно 63 - при тестировании в режиме CHS)), цвет которых несет определенную смысловую нагрузку.

```

ERR INDX CORR DREQ DRSC WRFT DRDY BUSY          AMNF TENF ABRT IDNF UNCR BBK
C[ 8944] H[15] S[63] [ 8452080] [ 289425]—S[ ]—H[ ] C[ 306]
M#HDD>scan
Scan...
[ 15937 kb/s]
- 1132
- 1
- 2
-
-
?-TIME -
x=UNC -
!-ABRT -
S-IDNF -
A-AMNF -
0-TENF -
*-BBK -
[ 3.1% ] [ 3.1% ]
Start : 16:23:41
Time : 5.07
End : 16:28:57
-- > Press the arrow keys for navigate on HDD during scan. | 16:23:50

```

В окошке справа ведется их счет. Самая верхняя строка показывает скорость считывания данных в данный момент (не зависит от используемого режима DMA), в нижней - два значения в процентах: первое значение показывает процент выполнения текущего теста в заданном промежутке, а второе отображает, насколько далеко головки "ушли" от 0 цилиндра и "пришли" к последнему. Ниже часы показывают время начала проверки и прогнозируемое время выполнения. Разъяснения по цветам приводятся в документации: "Чем "мутнее" квадратик, тем больше накопителю потребовалось времени для чтения этого блока секторов. Если пошли цветные блоки - значит, накопитель не "вписался" в отведенный ему для работы промежуток времени. Цветными блоками отображается ненормальное состояние поверхности, но еще без BAD'ов. Однако, зеленые блоки - нормальное явление, они отражают естественный износ поверхности, ничего страшного в них нет (старые накопители работают медленнее, и поэтому для них график вполне может окраситься в зеленый цвет целиком: это нормально). Также зеленые блоки могут появляться при вибрации самого накопителя. Чем ниже по меню цвет - тем больше накопителю понадобилось времени для чтения этого трудночитаемого участка. Красный цвет - признак того что на этом месте уже почти "сформировался" BAD block. Вопросительный знак появляется при превышении максимального времени ожидания готовности. То есть, при появлении [?] можно считать, что накопитель "подвис" на этом месте и здесь явно присутствует либо серьезный дефект поверхности, либо неисправен блок магнитных головок. Всё, что ниже вопросительного знака - это ошибки (BAD-блок). Если они появляются в процессе тестирования, значит, на поверхности есть физические дефекты или проблемы с CRC секторов, т.н. «софт-бэды».

При большом количестве бэдов полезно будет воспользоваться опцией ERASE WAITs (на приведенном выше рисунке она выключена): тогда при обнаружении сбойного сектора диск не будет долго над ним страдать, пытаясь его вычитать, а сразу же предпримет попытку записи.

С помощью MHDD можно проверить и надежность системы позиционирования: по команде SX будут проводиться многократные попытки позиционирования в различные области, некоторые накопители такой режим выдержать не могут в силу в большинстве случаев конструктивных особенностей и заводских дефектов (но не нужно бояться - из строя ничего не выйдет, так как программа отслеживает такую возможность и переводит диск в ждущее состояние, если что-то не так).

Есть еще ряд команд для диагностики, но это первые которые следует знать. Более полно описывать нет смысла - для этого есть документация к программе.

Из очень интересных функций реализуемых программой нужно отметить возможность управления шумом. По команде ААМ становится доступным изменение производимого уровня шума (нужно ли напоминать, что в ущерб производительности?), при этом результат можно оценить сразу же.

```
ERR INDX CORR DREQ DRSC WRFT DRDY BUSY          AMNF TENF ABRT IDNF UNCR BBK
C[ 8944] H[15] S[63] [ 8452000] [ --LBA-- ]--S[ ]--H[ ] C[ ]
MDD>aam
ST34311A 8944/15/63
SN:5BF0Y2BC FW:8.01 LBAs:8452000
Support: DLMCode LBA HPA DMA (UDMA2,MWDMA2)
SMART: Enabled
Security: high, OFF. Size = 4126Mb
Device Reset... OK
Setting Drive Parameters... OK.
Recalibrate... OK
Automatic Acoustic Management
AAM is not supported by this HDD.
Done.
MDD>
```

! (c) maysoft aka Dmitry Postrigan, http://mhdd.net | 2.9 | 17:59:27

```
ERR INDX CORR DREQ DRSC WRFT DRDY BUSY          AMNF TENF ABRT IDNF UNCR BBK
C[ 38792] H[16] S[63] [ 39102336] [ --LBA-- ]--S[ ]--H[ ] C[ ]
MDD>aam
ST320011A 38792/16/63
SN:3HT1700Z FW:3.10 LBAs:39102336
Support: DLMCode LBA HPA AAM DMA (UDMA5,MWDMA2)
SMART: Enabled
Security: high, OFF. Size = 19092Mb
Device Reset... OK
Setting Drive Parameters... OK.
Recalibrate... OK
Automatic Acoustic Management
Current setting : Maximum performance.
Press: <D> - disable AAM
      <P> - maximum performance
      <L> - middle performance (try if +/- not work)
      <M> - minimum acoustic emanation level
           (lowest speed)
      <+>, <-> - intermediate acoustic management levels
                126 - max. performance (disable AAM);
                0 - minimum acoustic emanation level.
      <ENTER> - finish.
Seeking... Now you can adjust AAM.
! (c) maysoft aka Dmitry Postrigan, http://mhdd.net | 2.9 | 18:02:16
```

## Программа FDISK

### *Разбивка жёсткого диска*

Утилита fdisk, очень простая и не требует много времени для освоения. Для запуска этой утилиты загружаемся с системного диска.

1. После выхода в режим DOS в командной строке пишем команду: fdisk и нажимаем Enter.
2. Появляется меню с вопросом «Включить поддержку больших дисков N-нет, Y- да». Выбираем Y и жмём Enter.
3. Появляется вот такое меню....

```
Опции FDISK

Текущий фиксированный диск: 1

Выберите один из следующих вариантов:

1. Создание раздела DOS или логических устройств DOS
2. Установка активного раздела
3. Удаление раздела DOS или логического устройства
4. Просмотр информации о разделах

Выберите: [ 4 ]

Нажмите Esc для выхода из FDISK
```

Для вызова нужного пункта меню необходимо набрать в командной строке необходимый номер...

Для начала посмотрим на наш жёсткий диск, вводим цифру 4 и жмём Enter... Появляется полная информация о содержании диска - типы файловых систем, объёмы и имена логических дисков (если разбиение уже имеется, то для создания нового разбиения следует удалить имеющиеся разделы, если на жёстком диске нет логических разделов, то можно сразу приступить к их созданию).

Нажмите Esc для возврата в главное меню...

Теперь нажмите 1 и Enter - появится меню создания разделов...

```
Создание раздела DOS либо логического диска DOS

Текущий жесткий диск:
Выберите действие:

1. Создание основного раздела DOS
2. Создание дополнительного раздела DOS
3. Создание логических дисков DOS в дополнительном разделе DOS

Введите номер выбранного действия: [ 1 ]

Для возврата к выбору режима работы FDISK нажмите клавишу Esc
```

Снова нажимаем 1 и Enter, появляется счётчик и программа начнёт создавать основной раздел, а когда она (программа) спросит у Вас сделать основной раздел на весь диск, то ответьте отказом - N. Далее Вам будет предложено ввести размер первого логического диска в процентах или мегабайтах. Сколько считаете нужным, столько и вводите...

Опять Esc и мы в главном меню...

В главном меню выбираем пункт 1 (создание раздела), а потом 2(создание дополнительного раздела). Создание дополнительного раздела точно такое же как и основного, только на предложение программы занять под дополнительный раздел весь имеющийся объём диска необходимо ответить - Y(Да) После того как раздел будет создан жмём Esc и снова выходим в главное меню.

Раздел, с которого Ваш компьютер будет загружаться называется активным. Для выбора активного раздела в главном меню выбираем цифру 2 - установка активного раздела и в появляющейся информации о диске вводим номер раздела.

Опять возвращаемся в главное меню(Esc), выбираем пункт 4 - просмотр информации о разделах и напротив раздела, который мы сделали активным будет стоять буква A

После того, как создание разделов будет завершено, необходимо перезагрузиться и отформатировать новые разделы перед их использованием.

## Порядок выполнения работы:

### 1. Проверка оперативной памяти

- 1.1 Установить в BIOS Setup загрузку с CD:
- 1.2 После загрузки найти и выбрать программу Memtest86+ и запустить ее.
- 1.3 Внимательно рассмотреть экран программы
- 1.4 Следить за процессом тестирования, необходимо, чтобы как минимум закончился один проход тестирования (столбец PASS)
- 1.5 Если были обнаружены ошибки (столбец ERRORS), то надо отразить это в отчете.
- 1.6 Сделать вывод об исправности или неисправности оперативной памяти.

## **2. Проверка жесткого диска**

- 2.1 Загрузится с загрузочного CD, найти и запустить «MHDD»
- 2.2 На некоторых компьютерах необходимо ввести код доступа (цифра\*2+1)
- 2.3 С помощью команды “id” или “F2” получить данные о жестком диске, записать основные в отчет
- 2.4 По команде “scan” проверить жесткий диск на наличие BAD-блоков, записать в отчет среднюю скорость передачи данных
- 2.5 Если будут найдены «красные» или «коричневые» блоки, попробуйте от них избавиться с помощью команды “erase waits” или “remap”
- 2.6 Оцените состояние вашего жесткого диска (износ, быстродействие, плохие блоки и т.д.)

## **3. Разбивка жесткого диска**

- 3.1 Загрузится с загрузочного CD
- 3.2 Запустить программу FDISK
- 3.3 Создать основной раздел равный 50% от всего объема диска
- 3.4 Создать дополнительный раздел равный второй половине жесткого диска
- 3.5 Создать логический диск D равный по объему дополнительному разделу
- 3.6 Диск C сделать активным, т.к. с него будет загружаться операционная система
- 3.7 Показать результат преподавателю
- 3.8 Выйти из программы и перезагрузить компьютер

## **4. Форматирование жесткого диска**

- 4.1 Загрузится с загрузочного CD
- 4.2 Запустить программу FORMAT с параметрами, для форматирования диска C (параметры изучить самостоятельно, команда “format /?” покажет все возможные параметры; определите какие нужны вам)
- 4.3 Когда форматирование закончится, покажите результат преподавателю

### **Контрольные вопросы**

1. Каким образом можно проверить оперативную память на наличие ошибок?
2. Каким образом информация записывается на жесткий диск?
3. Что такое BAD-блоки?
4. Можно ли от них избавиться и каким образом?
5. Для чего нужна разбивка жесткого диска на разделы?
6. Сколько разделов можно создать на жестком диске?
7. Для чего необходимо форматирование жестких дисков?
8. Объясните, почему реальный объем диска меньше, чем по паспорту?

**Форма представления результата:**

Отчет о проделанной работе:

1. Тема и цель работы.
2. Описание хода проведения работы, параметры тестируемых комплектующих.
3. Выводы о состоянии комплектующих.
4. Ответы на контрольные вопросы.

**Критерии оценки:**

Оценка «отлично» ставится, если задание выполнено верно.

Оценка «хорошо» ставится, если ход выполнения задания верный, но была допущена одна или две ошибки, приведшие к неправильному результату.

Оценка «удовлетворительно» ставится, если приведено неполное выполнение задания.

Оценка «неудовлетворительно» ставится, если задание не выполнено.

## Тема 3.1 Периферийные устройства вычислительной техники

### Лабораторное занятие № 11

Конструкция, подключение и инсталляция видеоадаптера. Тест монитора

**Цель:** изучить современные LCD-мониторы. Протестировать LCD-мониторы с помощью утилиты TFT Монитор Тест.

**Выполнив работу, Вы будете:**

**уметь:**

- подключать и устанавливать видеоадаптеры;
- тестировать монитор.

**Материальное обеспечение:**

Методические указания для выполнения лабораторных работ, компьютер.

**Задание:**

1. Запустить утилиту для тестирования мониторов.
2. При помощи утилиты протестировать монитор.
3. Изучить все параметры монитора.
4. Оформить отчет о проделанной работе. Сделать вывод.
5. Ответить на контрольные вопросы.

**Краткие теоретические сведения:**

*Тестирование LCD-мониторов*

**TFT Монитор тест**

Программа TFT Монитор тест разработана отечественными программистами. В отличие от существующих аналогов, она, во-первых, совершенно бесплатна, а во-вторых, снабжена подробной справкой с описанием имеющихся шаблонов на русском языке и советами по решению типичных проблем. Программа состоит из одного файла, не требует инсталляции.

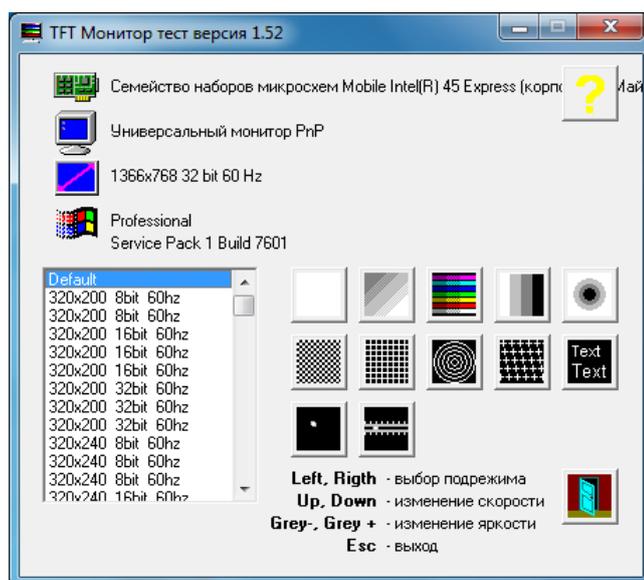


Рисунок 1 – Главное окно программы TFT Монитор тест

Небольшое начальное окно позволяет выбрать разрешение (для тестирования интерполяции), либо оставить разрешение по умолчанию, а также запустить один из 12 тестов. Во время просмотра тестовых шаблонов можно регулировать яркость, перемещаться между шаблонами в пределах теста, а также переключать цвет либо направление градиента. Тестовые шаблоны позволяют проверить:

- равномерность подсветки (1 тест);
- интерполяцию и фокусировку – отсутствие дрожания из-за преобразования аналогового сигнала (5 тестов);
- цветопередачу и разрешение цвета (3 теста);
- контрастность (1 тест);
- время отклика и инерционность (2 теста).

Разработчики даже попытались реализовать возможность получения количественного, а не только качественного результата.

В частности, в тестах на послесвечение можно попытаться с помощью фотографии оценить длину шлейфа и по этому показателю вычислить время отклика. А в тесте на интерполяцию можно зафиксировать, на каком шаге концентрические окружности перестают сливаться. Однако это будет лишь приблизительный результат. Главное предназначение программы – провести визуальное сравнение нескольких мониторов, на которые выводятся одни и те же шаблоны одновременно.

Таким образом, программа "TFT Монитор тест" является одной из лучших в своем классе. К ее плюсам можно причислить не только большой набор статических и анимированных шаблонов, но и простоту, бесплатность, небольшие размеры дистрибутива.

Все тесты, кроме проверки интерполяции, желательно проводить при родном разрешении монитора. Для 15" это обычно 1024x768 точек, для 17"-19" обычно 1280x1024 точек, для 20" обычно 1600x1200 точек. Разрядность цвета 24 или 32 бита. При 8 и 16 битных цветах тесты на градиент цвета будут показывать явные полосы не связанные с разрешением матрицы монитора.

Частота кадровой развертки используется в тестах для оценки быстродействия матрицы. Желательно выставлять максимальную частоту для данного видеорежима и способа подключения монитора.

При тестировании одновременно нескольких мониторов желательно выставить одинаковые настройки по яркости, контрастности и цветовой температуре.

Все тесты запускаются в полноэкранный режим в соответствии с выбранным разрешением. При выборе режима **Default** используется текущий установленный видеорежим. Для операционных систем Windows отдельно задается разрешение экрана, отдельно частота кадров. При этом не все частоты кадров для данного разрешения могут поддерживаться монитором.

### Описание функций

	<p>Тест равномерности подсветки матрицы и отсутствия “битых точек”          Изменение цвета – <b>Left, Right, LeftMouse click</b>          Изменение яркости – <b>Plus, Minus</b>          Выход из теста – <b>Esc RightMouse click</b></p>
	<p>Набор тестов из точек, расположенных в шахматном порядке, вертикальных линий и горизонтальных линий, для проверки интерполяции “неродных” разрешений. Перед проведением теста рекомендуется выполнить настройку монитора для каждого видеорежима.          Тесты выявляют неоднородность изображения в виде горизонтальных и(или) вертикальных полос, по сравнению с изображением в “родном” разрешении.          Изменение средней яркости экрана при переходе от вертикальных линий к горизонтальным при родном разрешении, служит качественной оценкой ширины</p>

	<p>полосы видеоканала при подключении через D-Sub.  Изменение режима – <b>Left, Right, LeftMouse click</b>  Изменение яркости – <b>Plus, Minus</b>  Выход из теста – <b>Esc RightMouse click</b></p>
	<p>Тест градиента цвета. Проверяется равномерность нарастания яркости от левого верхнего угла к правому нижнему.  На матрицах с низким цветовым разрешением наблюдается ступенеобразное изменение яркости.  На ряде матриц наблюдаются характерные диагональные полосы повышенной или пониженной яркости, из-за нелинейной характеристики матрицы. При запуске следует обратить внимание на установку 24 или 32 битных цветов.  Изменение направления градиента - <b>Up, Down</b>  Изменение цвета – <b>Left, Right, LeftMouse click</b>  Изменение яркости – <b>Plus, Minus</b>  Выход из теста – <b>Esc RightMouse click</b></p>
	<p>Тест аналогичен предыдущему, но выводит одновременно несколько горизонтальных или вертикальных цветовых клиньев. По тесту проверяется синхронность увеличения яркости клиньев и баланс цветов.  Изменение режима – <b>Left, Right, LeftMouse click</b>  Изменение яркости – <b>Plus, Minus</b>  Выход из теста – <b>Esc RightMouse click</b></p>
	<p>Тест ступенчатого градиента цвета. Проверяется равномерность нарастания яркости и разрешение перехода по градациям яркости. При запуске следует обратить внимание на установку 24 или 32 битных цветов.  Изменение режима - <b>Left, Right, LeftMouse click</b>  Выход из теста - <b>Esc RightMouse click</b></p>
	<p>Тест концентрических окружностей, для проверки интерполяции "неродных" разрешений. Перед проведением теста рекомендуется выполнить настройку монитора для каждого видеорежима. Тесты выявляют неоднородность изображения по сравнению с изображением в "родном" разрешении.  Изменение режима - <b>Left, Right, LeftMouse click</b>  Выход из теста - <b>Esc RightMouse click</b></p>
	<p>Тест градиента цвета. Проверяется равномерность нарастания яркости по радиусу окружности. На матрицах с низким цветовым разрешением наблюдается ступенеобразное изменение яркости. При запуске следует обратить внимание на установку 24 или 32 битных цветов.  Изменение направления градиента - <b>Up, Down</b>  Изменение цвета - <b>Left, Right, LeftMouse click</b>  Изменение яркости - <b>Plus, Minus</b>  Выход из теста - <b>Esc RightMouse click</b></p>
	<p>Набор узоров для проверки интерполяции "неродных" разрешений. Перед проведением теста рекомендуется выполнить настройку монитора для каждого видеорежима. Тесты выявляют неоднородность изображения в виде горизонтальных и(или) вертикальных полос, по сравнению с изображением в "родном" разрешении.  Изменение режима - <b>Left, Right, LeftMouse click</b>  Выход из теста - <b>Esc RightMouse click</b></p>
	<p>Тест для проверки читаемости шрифтов различного размера и насыщенности при "родных" и "неродных" разрешениях.  Изменение шрифтов – <b>Left, Right, LeftMouse click</b>  Изменение яркости – <b>Plus, Minus</b></p>

	<b>Выход из теста – Esc RightMouse click</b>
	Тест для проверки размытости и толщины линий. Как и предыдущий тест, характеризует читаемость шрифтов и мелких деталей. Изменение режима – <b>Left, Right, LeftMouse click</b> Изменение яркости – <b>Plus, Minus</b> Выход из теста – <b>Esc RightMouse click</b>
	Тест для визуальной оценки быстродействия матрицы. Наиболее нагляден при одновременном тестировании нескольких мониторов на матрицах разных типов, и при сравнении с CRT мониторами. Параметры для сравнения – яркостьдвигающегося квадрата, длина “хвоста”, размытость границы, цвет и геометрия квадрата. Изменение скорости – <b>Up, Down</b> Изменение яркости – <b>Plus, Minus</b> Выход из теста – <b>Esc RightMouse click</b>
	Тест аналогичен предыдущему, но квадрат движется по горизонтали или вертикали, и нанесена масштабная сетка. Изменение скорости – <b>Up, Down</b> Изменение режима – <b>Left, Right, LeftMouse click</b> Изменение яркости – <b>Plus, Minus</b> Выход из теста – <b>Esc RightMouse click</b>
	Выход из программы

### **Порядок выполнения работы:**

Технические характеристики дисплея (**заполняется самостоятельно**):

#### **Экран и видеокарта**

Размер экрана (дюйм) :

Разрешение дисплея (pix) :

Тип ЖК-матрицы:

Тип видеоадаптера (встроенный или внешний) :

Видеокарта :

Яркость:

Контрастность:

Время отклика:

Запустить утилиту TFT Монитор Тест для тестирования монитора. Результаты тестирования отразить в отчете.

#### **Пример результатов**



Тест1.

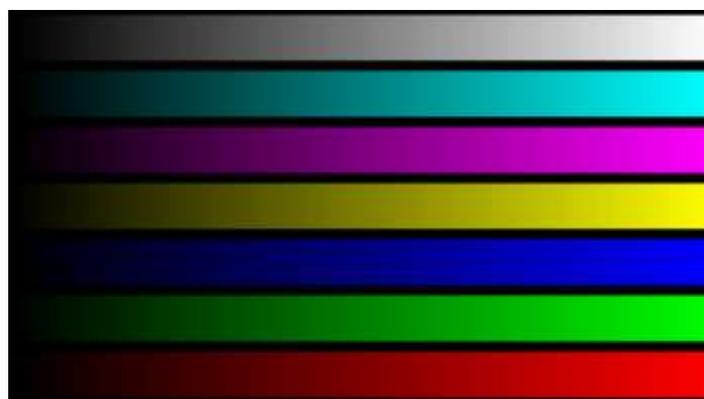
Тест равномерности подсветки матрицы и отсутствия “битых точек”. Была произведена заливка следующими цветами: белый, красный, зеленый, синий, голубой, розовый, желтый, черный. По всему экрану цвет распределен равномерно, искажений не наблюдается. Тест показал, что дисплей не поврежден и “битых точек” нет.

Тест 2.



Тест градиента цвета. У дисплея яркость изменяется плавно. Не наблюдаются диагональные полосы повышенной или пониженной яркости. Таким образом, равномерность нарастания яркости от левого верхнего угла к правому нижнему нормальная, никаких отклонений нет.

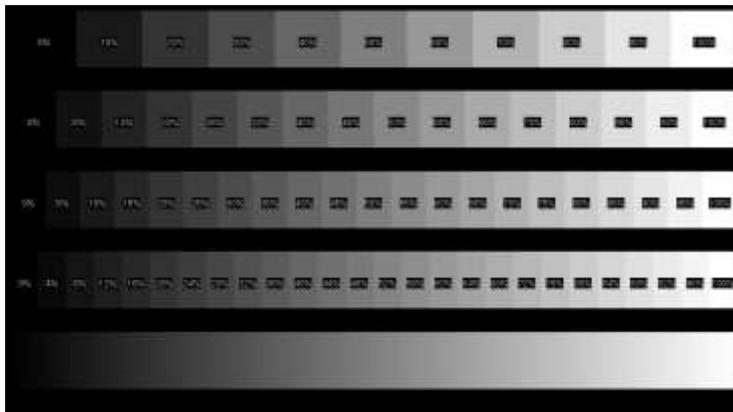
Тест 3.



Тест выводит одновременно несколько горизонтальных или вертикальных цветовых клиньев. По тесту видно, что для дисплея синхронность увеличения яркости клиньев и

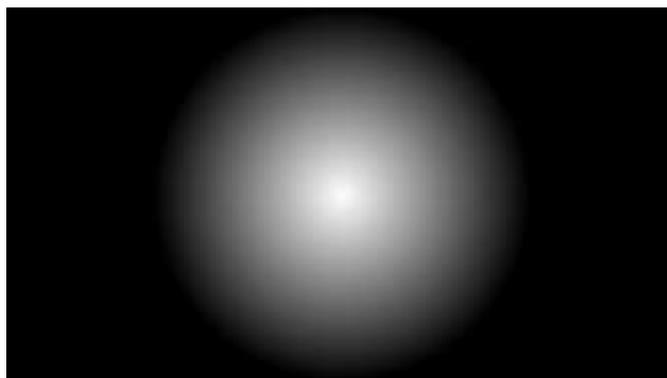
баланс цветов не нарушен. Все линии цветов изменяются плавно, достаточно насыщенные цвета, искажений нет.

Тест4



Тест ступенчатого градиента цвета. Равномерность нарастания яркости и разрешение перехода по градациям яркости в норме. Тест проводился для 4 типов градиентов: с шагом 10%, 6%, 5%, 4%. Во всех границы перехода плавные.

Тест 5.



Тест градиента цвета. У дисплея не наблюдается ступенчатое изменение яркости по радиусу окружности. Яркость от центра к периферии изменяется плавно по всей окружности.

Тест 6.



Набор тестов из точек, расположенных в шахматном порядке, вертикальных линий и горизонтальных линий, для проверки интерполяции «неродных» разрешений. Перед

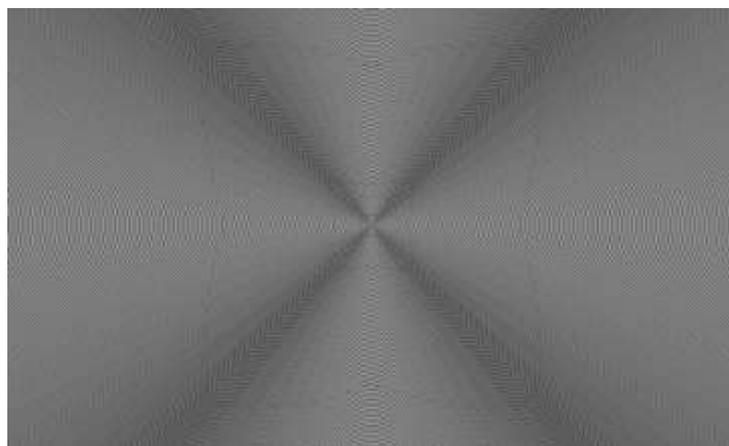
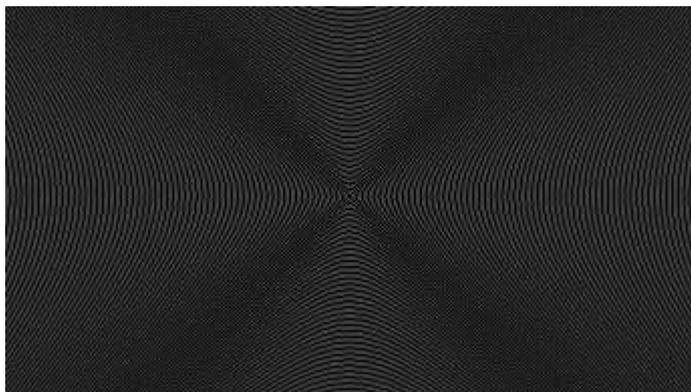
проведением теста выполнили настройку монитора для каждого видеорежима. Для дисплея тест выполнялся при «родном» разрешении \_\_\_\_\_, а также при «неродных» разрешениях: 1280x720, 1280x600, 800x600. При «родном» разрешении размытости и толщины линий не наблюдается. При «неродных» разрешениях 1280x600 и 800x600 наблюдалось утолщение линий и их размытость.

Тест 7.



Тест для проверки размытости и толщины линий. Результат теста показал, что линии параллельные, одинаковой толщины. Линии четкие и прямые. Также тест характеризует читаемость шрифтов и мелких деталей. Дисплей показал хорошие результаты теста, отклонений не выявлено.

Тест 8.



Тест концентрических окружностей. Он предназначен проверки интерполяции «неродных» разрешений. Перед проведением теста выполнили настройку монитора для каждого видеорежима. Тест выполнялся на \_\_\_\_\_ при «родном» разрешении

\_\_\_\_\_, а также при «неродных» разрешениях: 1280x720, 1280x600, 800x600. Результаты тестов показали, что при «родном» разрешении окружности ровные, а при «неродных» выявляют искажения изображения, возникают пересечения линий.

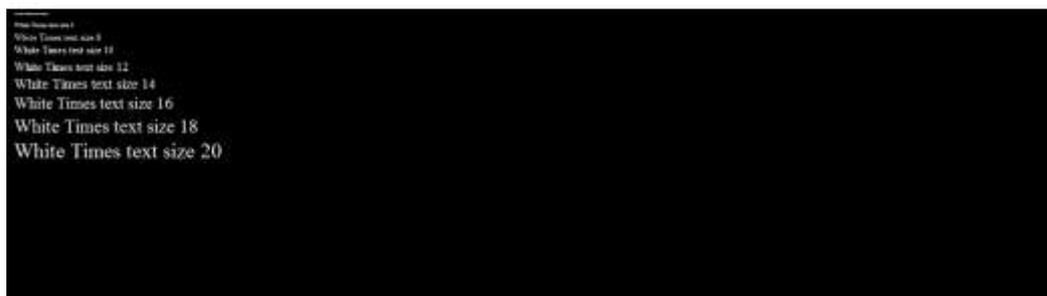
Тест 9.



Набор узоров для проверки интерполяции "неродных" разрешений. Перед проведением теста выполнили настройку мониторов для каждого видеорежима: «родное» разрешение для дисплея \_\_\_\_:\_\_\_\_, «неродные» разрешения 1280x720, 1280x600, 800x600. При «неродных» разрешениях наблюдалось частичное искажение изображения в виде вертикальных полос, размытости изображений.

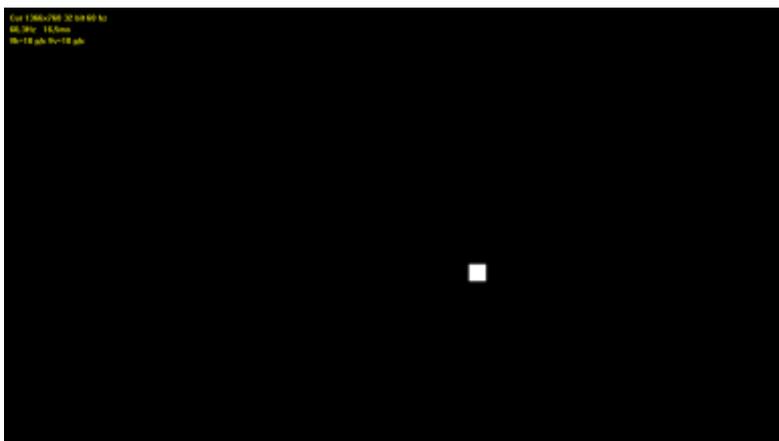
Тест 10.

Black Times text  
Black Times text size 8  
Black Times text size 10  
Black Times text size 12  
Black Times text size 14  
Black Times text size 16  
Black Times text size 18  
Black Times text size 20



Тест для проверки читаемости шрифтов различного размера и насыщенности при «родных» и «неродных» разрешениях. Тест выполнялся при «родном» разрешении, а также при «неродных» разрешениях: 1280x720, 1280x600, 800x600. При «родном» разрешении текст читается хорошо, все буквы понятные. При остальных разрешениях изображение становится менее четким, буквы расплывчатые, что становится особенно заметно при мелком шрифте.

Тест 11.



Тест для визуальной оценки быстродействия матрицы. Параметры для сравнения – яркость двигающегося квадрата, длина «хвоста», размытость границы, цвет и геометрия квадрата.

Информация в левом верхнем углу экрана:

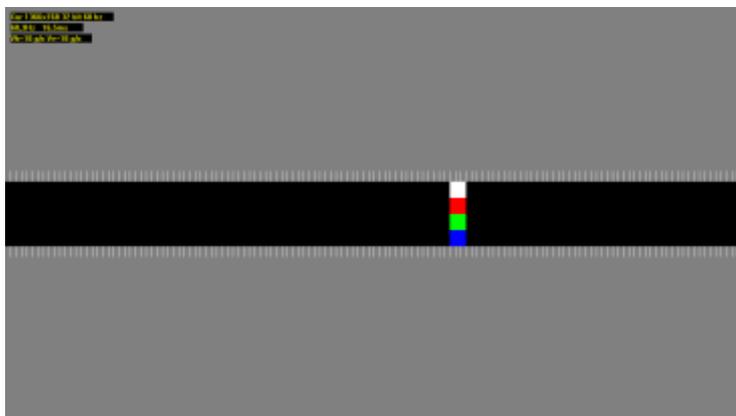
Cur 1366x768 32 bit 60 Hz - текущий видеорежим

60,3Hz 16,5ms - частота и интервал обновления экрана

Vh=10 p/c Vv=10 p/c - скорость смещения квадрата (точек/кадр)

Дисплей показал хорошие результаты теста: границы квадрата были четкими и яркими, цвет белый, частота обновления почти одинакова.

Тест 12.



Тест аналогичен предыдущему, но квадрат движется по горизонтали или вертикали, и нанесена масштабная сетка.

У дисплея границы квадрата четкие, цвета яркие и хорошо различимы, размытости нет.

**Вывод:** в ходе данной работы была изучена программа TFT Монитор Тест v.1.52 и проведено тестирование этой программой дисплея: \_\_\_\_\_.

Исходя из проведенного тестирования, можно сказать, что дисплей не имеет серьезных дефектов. Изображения на нем яркие, линии четкие, тесты равномерности подсветки и градиента цвета никаких отклонений не выявили.

Однако при проведении тестов на интерполяцию нестандартных разрешений были выявлены некоторые дефекты: при тестировании в разрешениях 1154x864, 1280x800, 800x600 мелкий текст становился нечетким, тест для проверки читаемости выявил размытость линий, при тесте с движущимися объектами был замечен, при тесте конических окружностей на дисплее появлялись искажения в виде линий. Поэтому для корректной работы монитора желательно использовать то разрешение, которое предусмотрено производителем.

**Форма представления результата:**

Отчет о проделанной работе.

**Критерии оценки:**

Оценка «отлично» ставится, если задание выполнено верно.

Оценка «хорошо» ставится, если ход выполнения задания верный, но была допущена одна или две ошибки, приведшие к неправильному результату.

Оценка «удовлетворительно» ставится, если приведено неполное выполнение задания.

Оценка «неудовлетворительно» ставится, если задание не выполнено.

## Тема 3.1 Периферийные устройства вычислительной техники

### Лабораторное занятие № 12

#### Конструкция, подключение и инсталляция принтера

**Цель:** научиться подключать матричный принтер к компьютеру, устанавливать драйвер на принтер в ОС Windows, пользоваться меню принтера для изменения настроек.

**Выполнив работу, Вы будете:**

**уметь:**

- подключать принтер;
- проверять работоспособность принтера.

**Материальное обеспечение:**

Методические указания для выполнения лабораторных работ, компьютер.

**Задание:**

Подключить принтер, проверить его работоспособность.

**Порядок выполнения работы:**

**1. Подключение принтера к системному блоку**

1.1 Для соединения принтера с компьютером используется специальный шнур. Подсоедините шнур одним концом к разъему LPT на системном блоке, другим к принтеру. Включите принтер в сеть.

1.2 После загрузки Windows зайдите в «Панель управления» >>> «Принтеры»

1.3 Запустите «Установку принтера»

1.4 Выбрать «Локальный принтер»

1.5 В списке выберите принтер Epson LQ-300 ESC/P2

1.6 Порт принтера выбрать LPT1

1.7 Распечатать пробную страницу

1.8 На этом установку принтера в операционной системе Windows можно считать завершенной.

2. Чтобы проверить работу принтера без подключения компьютера можно запустить печать теста. Для этого выключите принтер, затем нажмите клавишу «NO» и не отпуская ее включите принтер. Отпустите клавишу и принтер начнет печатать тест.

3. Чтобы изменить текущие настройки принтера, необходимо зайти в меню. Для этого выключите принтер, нажмите клавишу «YES» и не отпуская включите принтер. На бумаге распечатается вопрос на каком языке выводить меню, выберите английский. Далее напечатаются текущие настройки принтера. Обратите внимание на номер кодовой страницы, она должна быть 866, иначе принтер не сможет правильно отобразить русский шрифт. Остальные настройки можно оставить без изменений.

**Форма представления результата:**

Отчет о проделанной работе.

**Критерии оценки:**

Оценка «отлично» ставится, если задание выполнено верно.

Оценка «хорошо» ставится, если ход выполнения задания верный, но была допущена одна или две ошибки, приведшие к неправильному результату.

Оценка «удовлетворительно» ставится, если приведено неполное выполнение задания.

Оценка «неудовлетворительно» ставится, если задание не выполнено.

## Тема 3.1 Периферийные устройства вычислительной техники

### Лабораторное занятие № 13

#### Конструкция, подключение и инсталляция сканера

**Цель:** научиться подключать планшетный сканер к компьютеру, устанавливать драйвера на сканер в ОС Windows, пользоваться программами сканирования и распознавания текстов.

#### Выполнив работу, Вы будете:

##### *уметь:*

- подключать и настраивать сканер;
- выполнять поиск необходимой информации.

#### Материальное обеспечение:

Методические указания для выполнения лабораторных работ, компьютер.

#### Задание:

Подключите планшетный сканер к компьютеру, установите драйвера на сканер.

#### Краткие теоретические сведения:

### Системы распознавания (OCR)

**Системы распознавания или OCR-системы** (OCR - аббревиатура слов Optical Character Recognition) предназначены для автоматического ввода документов в компьютер. Это может быть книга, газета, журнал, факс - любой документ, который надо перенести в компьютер, перевести в электронный редактируемый формат.

У пользователя есть выбор:

**Первое:** перепечатать документ, восстановив его первоначальное оформление (таблицы, колонки, картинки и т. д.) или

**Второе:** использовать сканер и OCR-приложение (обычно в России говорят "система распознавания"), что гораздо быстрее и проще, в случае если OCR-приложение распознаёт документ аккуратно.

Работать с такой системой очень просто. Положите документ в сканер, нажмите на кнопку "Мастер Scan&Read" (при работе с FineReader), проверьте результат, нажмите кнопку "Сохранить в MS Word" (также можно сохранить в MS Excel или любом другом из поддерживаемых форматов - HTML, PDF, TXT и т. д.). Самое трудоёмкое в процессе ввода документа в компьютер с помощью OCR-системы - это проверка результата распознавания и воссоздание оформления исходного документа. Поэтому пользователи так высоко ценят точность распознавания (количество ошибок, допущенных при распознавании букв) и качество передачи оформления исходного документа. Этот факт подтверждают результаты проведенного нами исследования.

Ускорить проверку результата и сделать ее более надёжной пользователю поможет встроенная программа проверки орфографии. ABBYY FineReader поддерживает распознавание текста на 179 языках и имеет встроенную проверку орфографии для 36 из них.

ABBYY FineReader является лидером по точности распознавания во всех сравнительных тестах, проведённых в России, начиная с апреля 1995 года (выход версии FineReader 2.0). С момента своего выхода продукты FineReader получили **более 150 наград** от авторитетных международных компьютерных изданий.

У системы ABBYY FineReader есть ещё одно большое преимущество - это сохранение оформления документа. В распознанном документе будут точно переданы таблицы, картинки, разбиение текста по колонкам.

Таким образом, весь процесс переноса текста с бумаги на компьютер - от сканирования до сохранения результата - займёт меньше одной минуты (время зависит от документа, возможностей используемого компьютера и сканера). И "электронный" документ будет выглядеть точно также, как бумажный оригинал!

### **Что считают важным пользователи при работе с OCR-системой?**

Недавнее исследование, проведенное среди пользователей системы FineReader, показало, что пользователи считают важным при работе с OCR-системой следующие параметры:

- Точность распознавания - 95%
- Точность сохранения оформления в документах для текстовых процессоров (в форматах MS Word, MS Excel, Word Pro, Word Perfect) - 89%
- Точность сохранения оформления для последующей электронной публикации (в форматах PDF, HTML) - 87%
- Работа с таблицами и многоколоночными текстами - 87%
- Простота использования - 85%
- Надёжность работы - 82%
- Удобный поиск ошибок и сверка с оригиналом - 80%
- Работа с цветом (сохранение цветных картинок и цвета шрифта, фона) - 63%
- Прямой экспорт в другие приложения - 61%
- Скорость - 55%
- Многоязычное распознавание - 25%

### **Технические характеристики сканера HP ScanJet 2400c (Самостоятельно найти и выписать характеристики сканера HP ScanJet 3800)**

<b>Система сканирования</b>	
Оптическое разрешение при сканировании	До 1200 т/д
Аппаратное разрешение сканирования	До 1200 x 1200 т/д
улучшенное разрешение при сканировании	Улучшенное разрешение до 999999 т/д
Разрядность	48 бит
Оттенки серого	256
Диапазон масштабирования или увеличения изображений	от 10 до 2000% с шагом 1%
Скорость сканирования в режиме предварительного просмотра	12 секунд
Скорость сканирования (цветное фото 4x6)	< 45 сек.
Скорость сканирования (оптическое распознавание символов, А4)	< 45 сек.
Максимальный размер сканируемого оригинала	21,59 x 29,72 см

<b>Другая техническая информация</b>	
Режимы ввода при сканировании	Кнопки "сканировать" и "копировать" на лицевой панели; ПО от HP для создания и обработки фотоизображений
Формат файла сканирования	Bitmap (BMP), TIFF, GIF, PDF, HTML, JPEG, FlashPix (FPX), сжатый TIFF, PCX, PNG, RTF, TXT
Совместимые операционные системы	Microsoft® Windows® (98, 2000, Me, XP Home и Professional Edition), Mac OS X 9.1 или 10.1 и выше
Стандартная подсоединяемость	USB (совместим со спецификациями USB 2.0).
Внешние порты ввода-вывода	1 интерфейс Hi-Speed USB – совместим со спецификациями USB 2.0
Минимальные системные требования	ПК с портом USB и ОС Microsoft® Windows® 98, Me, XP Professional, XP Home Edition, 2000 или выше, привод CD-ROM; Microsoft® Windows®: ПК с процессором Intel® Pentium® II, Celeron® или другим совместимым процессором; 225 Мб свободного пространства на жёстком диске, ОЗУ 64 Мб; монитор SVGA 800 x 600 (16 бит); порт USB на ПК; IE 5.01 SP2 или выше; программа для создания мультимедиа-альбомов HP требует наличия устройства CD-Writer и дополнительно 700 Мб свободного пространства на жёстком диске
Сертификат Energy Star	Да
Размеры (Ш x Г x В)	458 x 275 x 615 мм
Вес	2,2 кг
Вес пакета	2,4 кг
Электромагнитная совместимость	ЕС (сертификат соответствия CE), Россия (ГОСТ)
Потребляемая мощность	17 Вт (максимум)
Тип блока питания	Внешний блок питания
Требования к питанию	100-240 В перем. тока (+/-10%), 50/60 Гц (+/-3 Гц)
Безопасность	ЕС (соответствие требованиям IEC 60950 3-е издание; сертификация безопасности на работу с решениями других производителей), Польша (PCBC) и Россия (ГОСТ)

**Порядок выполнения работы:**

1. Для работы используется сканер HP scanjet 2400 (3800)
2. Запустите программу установки с диска, который идет в комплекте с сканером
3. Выберите в меню пункт «Установка программ»
4. В следующем окне выберите «Программу обработки изображений и фотографий HP» и нажмите «Далее»

5. После этого нажмите далее и в главном меню выберите пункт «Инструкция по подключению сканера»
6. После ознакомления с инструкцией нажмите «Закрыть» и «Выход»
7. Подключите сканер к компьютеру при помощи USB кабеля и включите сканер в сеть
8. Операционная система сама найдет новое устройство, если этого не произошло, перезагрузите компьютер
9. Сканером можно управлять с помощью программы «Директор обработки фотографий и изображений» (Пуск-Программы), или непосредственно из программы, поддерживающей работу со сканером.
10. Запустите программу «Fine Reader» («CuneiForm»). Если она не установлена, то установите ее
11. Нажмите кнопку «Сканировать», при этом запустится программа сканирования и автоматически будет выполнено предварительное сканирование документа
12. В окне размеров все оставьте как есть, в меню «Формат вывода» выберите «черно-белое 1 бит», в меню «Дополнительные» выберите «Разрешение». В открывшемся окне выберите разрешение 300.
13. Нажмите кнопку «Принять». Будет выполняться сканирование документа непосредственно в программу «Fine Reader» («CuneiForm»).
14. Нажмите кнопку «Распознать»
15. Нажмите кнопку «Передать страницы в Word»
16. Проверьте документ на наличие ошибок

#### **Контрольные вопросы**

1. Как устроен сканер?
2. Какие разрешения используются при сканировании?
3. Что значит оптическое (физическое) и программное (интерполяционное) разрешения?
4. Какие типы сканеров вы знаете?
5. От чего зависит быстродействие сканера?

#### **Форма представления результата:**

Отчет о проделанной работе.

#### **Критерии оценки:**

Оценка «отлично» ставится, если задание выполнено верно.

Оценка «хорошо» ставится, если ход выполнения задания верный, но была допущена одна или две ошибки, приведшие к неправильному результату.

Оценка «удовлетворительно» ставится, если приведено неполное выполнение задания.

Оценка «неудовлетворительно» ставится, если задание не выполнено.