



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ**  
**«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**  
**(ДГТУ)**

Колледж экономики, управления и права

УТВЕРЖДАЮ  
Директор колледжа  
\_\_\_\_\_ В.И. Мигаль  
личная подпись

«30» июня 2022 г.

Рег. № \_\_\_\_\_

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ**

для обучающихся по практическим занятиям по дисциплине  
ОП.02 Архитектура аппаратных средств  
для специальности: 09.02.07 Информационные системы и программирование

Ростов-на-Дону 2023 г.

## ОГЛАВЛЕНИЕ

Практическая работа №1 Изучить принципы формы представления чисел в ЭВМ .....	6
Практическая работа №2 Кодирование информации в компьютере. ....	16
Практическая работа №3 Логической основой компьютера.....	27
Практическая работа №4 Разработка программ для машины Тьюринга .....	32
Практическая работа №5 Конфигурации вычислительной машины .....	36
Практическая работа №6 Установка различных модулей ПК.....	41
Практическая работа № 7 Написание программ для эмулятора ЭВМ «ЛамПанель».....	47
Практическая работа № 8 Изучение и диагностика блока питания ПК .....	58
Практическая работа № 9 Подключение и настройка мониторов и видеоадаптеров .....	61
Практическая работа №10 Устройство ввода информации в ЭВМ .....	63
Практическая работа № 11 Подключение и инсталляция принтеров.....	69
Практическая работа № 12 Инсталляция драйверов устройств.....	71
ЛИТЕРАТУРА .....	74

## ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Методические указания по выполнению практических работ учебной дисциплины ОП.02 Архитектура аппаратных средств предназначены для упорядочения работы обучающихся специальности 09.02.07 Информационные системы и программирование.

Программой учебной дисциплины ОП.02 Архитектура аппаратных средств предусмотрено проведение практических занятий в количестве 24 часа по специальности: 09.02.07 Информационные системы и программирование.

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

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

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

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

В результате освоения учебной дисциплины обучающийся должен овладеть общими и профессиональными компетенциями, включающими в себя способность:

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.

Характерная черта практических занятий – индивидуальное выполнение заданий, самостоятельное приобретение знаний. В связи с этим предусмотрены работы по всем основным разделам курса. Перед выполнением практической работы обучающийся получает опережающее теоретическое домашнее задание. На занятии объясняются вопросы, уточняются определения, которые помогают выполнению заданий. Обучающийся может просмотреть запись объяснения любой примерной работы по всем темам. И только после этого обучающийся приступает к выполнению практической работы.

При выполнении работы обучающийся должен самостоятельно изучить методические рекомендации по проведению практической работы, подготовить ответы на контрольные вопросы. Все практические задания выполняются за компьютером, теоретические вопросы сдаются устно или письменно.

После выполнения работы обучающийся должен представить отчет о проделанной работе с полученными результатами и в устной форме защитить.

При отсутствии по неуважительной причине обучающийся выполняет работу самостоятельно во внеурочное время и защищает на консультации по расписанию.

### **Структура практических работ:**

1. Тема.
2. Цель.
3. Теоретическое обоснование.
4. Ход работы.
5. Контрольные вопросы.
6. Содержание отчета.
7. Литература.

### **Ход работы**

1. Изучит теоретическое обоснование.
2. Выполнить практические задания по вариантам.
3. Ответить на контрольные вопросы по указанию преподавателя.
4. Оформить отчет.

### **Содержание отчета**

1. Тему, цель.
2. Выполненные задания по вариантам.
3. Вывод.

При изучении дисциплины необходимо постоянно обращать внимание студентов на ее прикладной характер, показывать, где и когда изучаемые теоретические положения и практические навыки могут быть использованы в будущей профессиональной деятельности.

Темы практических занятий

<b>№ п/п</b>	<b>Темы практических занятий</b>	<b>Ко- личе- ство часов</b>
1	Практическое занятие №1 Изучить принципы формы представления чисел в ЭВМ	2
2	Практическое занятие №2 Кодирование информации в компьютере	2
3	Практическое занятие №3 Логической основой компьютера	2
4	Практическое занятие №4 Разработка программ для машины Тьюринга	2
5	Практическое занятие №5 Анализ конфигурации вычислительной машины	2
6	Практическое занятие №6 Установка различных модулей ПК.	2
7	Практическое занятие №7 Написание программ для эмулятора ЭВМ «ЛамПанель»	2
8	Практическое занятие №8 Изучение и диагностика блока питания ПК	2
9	Практическое занятие №9 Изучение интерфейсов подключение и настройка мониторов и видеоадаптеров.	2
10	Практическое занятие №10 Устройство клавиатуры и мыши, настройка параметров работы клавиатуры и мыши	2
11	Практическое занятие №11 Подключение и инсталляция принтеров	2
12	Практическое занятие №12 Установка драйверов устройств	2
Итого:		24

## Практическая работа №1 Изучить принципы формы представления чисел в ЭВМ

**Тема:** взаимный перевод чисел.

**Цели:** изучить принципы формы представления чисел в ЭВМ и правила выполнения арифметических действий в ЭВМ, построения правил десятичной арифметики.

**Студент должен**

**знать:**

- общие сведения о системах счисления;
- формы представления чисел;
- формы кодирования целых чисел;

**уметь:**

- представлять целые числа в различных формах;
- выполнять операцию сложения с целыми числами в двоичных кодах;
- разрабатывать арифметические правила для любой системы счисления.

### Теоретическое обоснование

**Счислением** называется совокупность приемов наименования и обозначения чисел.

Сегодня человек использует различные системы счисления. Как правило, эти системы различаются между собой количеством используемых для обозначения различных чисел цифр. В зависимости от способа изображения чисел, системы счисления делятся на:

- позиционные;
- непозиционные.

В **позиционной** системе счисления цифры изменяют своё количественное значение в зависимости от их расположения в числе.

Например, в числе 1841,21 имеются три цифры «1». Все они имеют разные значения.

Первая обозначает тысячи, вторая единицы, а третья сотые доли соответственно. Указанное число является сокращенной записью следующей суммы:

$$1841,21_{10} = 1 \cdot 10^3 + 8 \cdot 10^2 + 4 \cdot 10 + 1 + 2 \cdot 10^{-1} + 1 \cdot 10^{-2}$$

Количество (P) различных цифр, используемых для изображения числа в позиционной системе счисления, называется **основанием системы счисления**. Значения цифр лежат в диапазоне от 0 до P-1.

Для предотвращения разночтений основание системы счисления будем записывать в виде нижнего индекса справа от числа.

В общем случае запись любого смешанного числа в системе счисления с основанием P будет представлять собой сумму вида:

$$N_P = a_{m-1} \cdot P^{m-1} + a_{m-2} \cdot P^{m-2} + \dots + a_k \cdot P^k + \dots + a_1 \cdot P^1 + a_0 \cdot P^0 + a_{-1} \cdot P^{-1} + a_{-2} \cdot P^{-2} + \dots + a_{-s} \cdot P^{-s} \quad (1)$$

Нижние индексы определяют местоположение цифры в числе (разряд):

- положительные значения индексов – для целой части числа (m разрядов);
- отрицательные значения индексов – для дробной части числа (s разрядов).

Максимальное целое число, которое может быть представлено в m разрядах:

$$N_{\max} = P^m - 1 \quad (2)$$

- Минимальное значащее, не равное 0 число, которое можно записать в s разрядах дробной части:

$$N_{\min} = P^{-sm} \quad (3)$$

- Имея в целой части числа m, а в дробной – s разрядов, можно записать всего  $P^{m+s}$  разных чисел.

### 1. Перевод чисел из одной системы счисления в другую

1.1. **Перевод чисел из двоичной системы счисления в восьмеричную и шестнадцатеричную** и обратный перевод чисел. Правила перевода чисел из двоичной системы в восьмеричную, шестнадцатеричную и обратно достаточно просты, поскольку основания восьмеричной и шестнадцатеричной систем счисления выражаются целой степенью двойки:  $8 = 2^3$ ,  $16 = 2^4$ .

**Таблица перевода восьмеричных чисел в двоичные**

восьмеричное число	двоичное число
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

**Таблица перевода шестнадцатеричных чисел в двоичные**

шестнадцатеричное число	двоичное число
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
A (10)	1010
B (11)	1011
C (12)	1100
D (13)	1101
E (14)	1110
F (15)	1111

1.2. **Перевод чисел из восьмеричной системы счисления в двоичную** осуществляется представлением каждой цифры восьмеричного числа трехразрядным двоичным числом – *триадой*.

$$762,35_8 = 111\ 110\ 010, 011\ 101_2$$

1.3. **Перевод шестнадцатеричных чисел в двоичную систему счисления** осуществляется представлением каждой цифры шестнадцатеричного числа четырехразрядными двоичными числами – *тетрадами*.

$$A7B,C7_{16} = 1010\ 0111\ 1011,11000111_2$$

1.4. **Перевод чисел из двоичной системы счисления в восьмеричную или шестнадцатеричную систему** осуществляется представлением разрядов двоичного числа, которые разбиваются на группы по три разряда при переводе в восьмеричную систему или по четыре разряда при переводе в шестнадцатеричную систему, отсчитывая от запятой влево и вправо.

При этом неполные крайние группы дополняются нулями. Затем каждая двоичная группа представляется цифрой той системы счисления, в которую переводится число.

$$001\ 111, 101010_2 = 17,52_8$$

$$0101\ 1100, 1011011_2 = 5C,B6_{16}$$

## 2. Арифметические операции для двоичных и шестнадцатеричных чисел

Арифметические операции для двоичных и шестнадцатеричных чисел выполняются по тем же правилам, что и для десятичных чисел. Рассмотрим на примерах выполнение таких арифметических операций, как сложение, вычитание и умножение для целых чисел.

Все арифметические действия, которые применимы к двоичным числам, выполняются аналогично как в десятичной системе. Удобнее всего двоичные числа складывать, вычитать, умножать и делить столбиком.

Числа записываются друг под другом с учетом разрядов. При необходимости производится перенос в старший разряд или заем из старшего разряда.

### 2.1. Сложение

Вычисление суммы двоичных чисел производится следующим образом: числа записываются в столбик. Затем производится поразрядное суммирование цифр, начиная с младшего разряда, как в десятичной системе. Если сумма цифр текущего разряда превышает его размер, то происходит перенос единицы в старший разряд.

**Поразрядная сумма** формируется по следующим правилам:

$$0 + 0 = 0$$

$$0 + 1 = 1$$

$$1 + 0 = 1$$

$$1 + 1 = 0 \text{ – осуществляется перенос } 1 \text{ в старший разряд}$$

Например, выполните суммирование двух двоичных чисел 1000111 и 110011 как показано в таблице

Первое слагаемое	1	0	0	0	1	1	1
Второе слагаемое		1	1	0	0	1	1
Сумма	1	1	1	1	0	1	0

На примере видно, как происходит перенос в старший разряд. При сложении единиц самого младшего разряда получается 10. Ноль остается на своем месте, а единица переносится в старший разряд слева, где уже складываются две единицы. Получается 11. И снова, младшую единицу оставляют, а старшую переносят влево.

Выполните суммирование двоичных чисел:

а)  $1111000111_2 + 11010101_2$ ;

в)  $1001111010,010001_2; + 1000001111,01_2$

### 2.2. Вычитание

Действие разности следует также выполнять столбиком. Вычитание производится поразрядно. Если возникает ситуация, что приходится вычитать из нуля единицу, то происходит заем из старшего разряда.

**Поразрядная разность** формируется по следующим правилам:

$$0 - 0 = 0$$

$$1 - 0 = 1$$

$$1 - 1 = 0$$

$$0 - 1 = 1 \text{ – после заёма } 1 \text{ из старшего разряда}$$

Все как в десятичной системе. Только следует помнить, что в двоичной системе  $10 - 1 = 1$ .

Например, разность чисел:  $1000111 - 110011 = 10100$

Уменьшаемое	1	0	0	0	1	1	1
Вычитаемое		1	1	0	0	1	1
Разность			1	0	1	0	0

На примере видно, как производится заем в старшем разряде. В пятом справа разряде производится вычитание  $0 - 1$ . Здесь следует занять единицу из ближайшего старшего разряда слева.

Выполните вычитание двоичных чисел:



- а)  $1101000111_2 - 110101_2$ ;  
 в)  $1001011010,010001_2; - 100000111,01_2$

### 2.3. Умножение

**Поразрядные произведения** формируются по следующим правилам:

- $0 \times 0 = 0$   
 $0 \times 1 = 0$   
 $1 \times 0 = 0$   
 $1 \times 1 = 1$

Произведение выполняется также поразрядно, каждый разряд второго числа умножается на каждую цифру первого числа, результат суммируется

Произведение двоичных чисел  $1101 * 11 = 100111$

Первый множитель			1	1	0	1
Второй множитель					1	1
			1	1	0	1
		1	1	0	1	
Итог (произведение)	1	0	0	1	1	1

Выполните умножение двоичных чисел:

- а)  $1101000111_2 - 110101_2$ ;  
 в)  $1001011010,010001_2; - 100000111,01_2$

### 2.4. Деление

Операция деления выполняется столбиком, аналогично как в десятичной системе счисления.

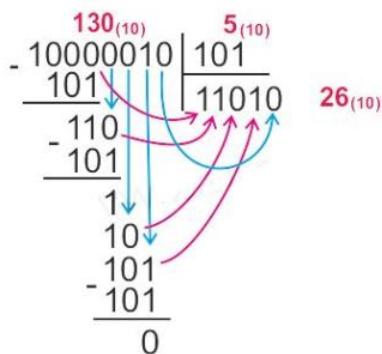


Рис. 2. Деление двоичных чисел

### 2.5. Операция сдвига по разрядной сетке.

В компьютерах, кроме операции алгебраического суммирования двоичных чисел, к которой относятся операции сложения и вычитания, выполняется **операция сдвига числа** по разрядной сетке влево и вправо, осуществляющая, фактически, умножение и деление двоичных чисел.

**В случае сдвига влево** осуществляется **умножение** двоичного числа на  $2^j$ , а **при сдвиге вправо** – **деление** на  $2^j$ , где  $j$  – количество разрядов, на которое сдвигается двоичное число.

Например, выполните сдвиг в двоичном числе на 2 разряда влево и вправо:

- 1)  $000011_2 = 3_{10}$  влево  
 $001100_2 = 12_{10}$   
 т. е.  $3 \times 4(2^2) = 12_{10}$   
 2)  $001000_2 = 8_{10}$  вправо  
 $000010_2 = 2_{10}$   
 т. е.  $8 : 4(2^2) = 2_{10}$

В компьютерах часто используется *циклический сдвиг*, при выполнении которого разрядная сетка, отведенная для *операнда* (числа, над которым производится действие), представляется замкнутой в кольцо. Тогда *при сдвиге влево* содержимое старшего разряда попадает в младший разряд операнда, а *при сдвиге вправо* содержимое младшего разряда попадает в старший разряд операнда.

Алгебраическое представление двоичных чисел, то есть представления чисел с учетом их знака включает в себя арифметическое представление числа и знаковый разряд. При этом код «0» означает знак «+» (плюс), код «1» – знак «-» (минус). Для алгебраического представления чисел в вычислительных машинах используют специальные коды:

- прямой код числа;
- обратный код числа;
- дополнительный код числа.

Два последних кода позволяют заменить неудобную для ЭВМ операцию вычитания на операцию сложения с соответствующим отрицательным числом.

Дополнительный код обеспечивает более быстрое выполнение операций, поэтому чаще всего в ЭВМ применяется именно он.

Прямой код числа  $N = [N]_{пр}$ . Пусть  $N = a_1 a_2 a_3 \dots a_m$ . Тогда:

- если  $N > 0$ , то  $[N]_{пр} = 0, a_1 a_2 a_3 \dots a_m$ ;
- если  $N < 0$ , то  $[N]_{пр} = 1, a_1 a_2 a_3 \dots a_m$ ;
- если  $N = 0$ , то имеет место неоднозначность:  $[0]_{пр} = 0, 0 \dots$  или  $1, 0 \dots$ ;

В общем виде:

$$[N]_{пр} = \begin{cases} N, & \text{если } N \geq 0, \\ 1 - N, & \text{если } N < 0 \end{cases}$$

Обратный код числа  $N = [N]_{обр}$ .

Обозначение  $\bar{a}$  означает величину, обратную  $a$  (инверсию  $a$ ), то есть если  $a = 1$ , то  $\bar{a} = 0$ , и наоборот.

- если  $N > 0$ , то  $[N]_{обр} = [N]_{пр} = 0, a_1 a_2 a_3 \dots a_m$ ;
- если  $N < 0$ , то  $[N]_{обр} = 1, \bar{a}_1, \bar{a}_2, \bar{a}_3 \dots \bar{a}_m$ ;
- если  $N = 0$ , то имеет место неоднозначность:  $[0]_{обр} = 0, 0 \dots$  или  $1, 11 \dots 1$ .

Обратный код положительного числа совпадает с прямым кодом. Для того чтобы получить обратный код отрицательного числа, необходимо цифры этого числа инвертировать, то есть во всех значащих разрядах нули заменить единицами, а единицы нулями, в знаковом разряде поставить 1.

Например, число  $N = 0,1011$ ,  $[N]_{обр} = 0,1011$ .

Число  $N = -0,1011$ ,  $[N]_{обр} = 1,0100$ . В случае, когда  $N < 0$ ,  $[N]_{обр} = 10 - 1 \cdot 10^{-n} + N$ , то есть  $[N]_{обр} = 1,1111 + N$ .

Обобщая результаты, получим:

$$[N]_{обр} = \begin{cases} N, & \text{если } N \geq 0, \\ 10 - 1 \cdot 10^{-n} + N, & \text{если } N < 0 \end{cases}$$

Дополнительный код числа  $N = [N]_{доп}$ .

- если  $N \leq 0$ , то  $[N]_{доп} = [N]_{пр} = 0, a_1 a_2 a_3 \dots a_m$ ;
- если  $N > 0$ , то  $[N]_{доп} = 1, \bar{a}_1, \bar{a}_2, \bar{a}_3 \dots \bar{a}_m, +0, 0 \dots 1$ .

Дополнительный код положительного числа совпадает с прямым кодом. Для того чтобы получить дополнительный код отрицательного числа, необходимо все его цифры инвертировать (в знаковом разряде поставить единицу, во всех значащих разрядах нули заменить единицами, а единицы нулями) и затем к младшему разряду прибавить единицу. В случае возникновения переноса из первого после запятой разряда в знаковый разряд, к числу следует прибавить единицу в младший разряд.

Например, число  $N = 0,1011$ ,  $[N]_{доп} = 0,1011$ .

Число  $N = -0,1011$ ,  $[N]_{доп} = 1,0100$ .  $N = 0,0000$ ;  $[N]_{доп} = 10,0000 = 0,0000$  (1 исчезает). Неоднозначности в изображении 0 нет.

## Практические задания

1. Произвести последовательный перевод целого числа из десятичной системы счисления в системы счисления с основаниями, заданными степенями числа. Системы счисления определяются вариантом задания.

2. Произвести перевод дробного числа из десятичной системы счисления в системы счисления с основанием, заданными степенями числа. При иррациональном представлении числа использовать первые десять разрядов. Системы счисления определяются вариантом задания.

3. Произвести перевод целого числа из десятичной системы счисления в систему счисления, определенную вариантом задания.

4. Произвести перевод дробного числа из десятичной системы счисления в систему счисления, определенную вариантом задания. При иррациональном представлении числа использовать первые десять разрядов. Система счисления определяется вариантом задания.

Оформление работы должно отражать и иллюстрировать принципы перевода чисел из одной системы счисления в другую.

Например:  $100010 / 2 = 500 (0) / 2 = 250 (0) / 2 = 125 (0) / 2 = 62 (1) / 2 = 31 (0) / 2 = 15 (1) / 2 = 7 (1) / 2 = 3 (1) / 2 = 1 (1) / 2 = 0 (1)$ .

Таким образом:  $100010 = 11111010002$

$11111010002 \rightarrow 11\ 1110\ 10002 \rightarrow 0011\ 1110\ 10002 \rightarrow 3E816$

$11111010002 \quad X = 1 * 29 + 1 * 28 + 1 * 27 + 1 * 26 + 1 * 25 + 0 * 24 + 1 * 23 + 0 * 22 + 0 * 21 + 0 * 20 = 100010$

5. Получить прямой код целого числа.

6. Получить обратный код целого числа.

7. Получить дополнительный код целого числа.

Оформление работы должно отражать и иллюстрировать принципы выполнения заданий.

## Варианты заданий

**Задание 1.** Выполнить перевод чисел:

1. Из десятичной системы счисления в двоичную, восьмеричную и шестнадцатеричную системы счисления.

2. Перевести данное число в десятичную систему счисления.

**Задание 1.**

Вариант 1

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

2. а)  $1100111011_2$ ; в)  $10110101,1_2$ ; г)  $100000110,10101_2$ ; д) 671,24<sub>8</sub>; е) 41A,6<sub>16</sub>.

**Вариант 2**

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

2. а)  $1001110011_2$ ; б)  $1111100111,01_2$ ; в)  $1010001100,101101_2$ ; г) 413,41<sub>8</sub>; д) 118,8C<sub>16</sub>.

**Вариант 3**

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

2. а)  $1100000000_2$ ; б)  $1101011111_2$ ; в)  $1011110100,011_2$ ; г) 1017,2<sub>8</sub>; д) 111,В<sub>16</sub>.

**Вариант 4**

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

2. а) 1100001001<sub>2</sub>; б) 1100100101<sub>2</sub>; в) 111110110,01<sub>2</sub>; г) 11001100,011<sub>2</sub>; д) 112,04<sub>8</sub>; е) 334,А<sub>16</sub>.

### Вариант 5

1. а) 500; б) 675; в) 810,25; г) 1017,25; д) 123,72.  
2. а) 1101010001<sub>2</sub>; б) 100011100<sub>2</sub>; в) 1101110001,011011<sub>2</sub>; г) 110011000,111001<sub>2</sub>; д) 1347,17<sub>8</sub> (8); е) 155,6С<sub>16</sub> (16).

### Вариант 6

1. а) 218; б) 808; в) 176,25; г) 284,25; д) 253,04.  
2. а) 111000100<sub>2</sub>; б) 1011001101<sub>2</sub>; в) 10110011,01<sub>2</sub>; г) 1010111111,011<sub>2</sub>; д) 1665,3<sub>8</sub>; е) FА,7<sub>16</sub>.

### Вариант 7

1. а) 306; б) 467; в) 218,5; г) 667,25; д) 318,87.  
2. а) 1111000111<sub>2</sub>; б) 11010101<sub>2</sub>; в) 1001111010,010001<sub>2</sub>; г) 1000001111,01<sub>2</sub>; д) 465,3<sub>8</sub>; е) 252,38<sub>16</sub>.

### Вариант 8

1. а) 167; б) 113; в) 607,5; г) 828,25; д) 314,71.  
2. а) 110010001<sub>2</sub>; б) 100100000<sub>2</sub>; в) 111001100,111<sub>2</sub>; г) 1010111010,1110111<sub>2</sub>; д) 704,6<sub>8</sub>; е) 367,38<sub>16</sub>.

### Вариант 9

1. а) 342; б) 374; в) 164,25; г) 520,375; д) 97,14.  
2. а) 1000110110<sub>2</sub>; б) 111100001<sub>2</sub>; в) 1110010100,1011001<sub>2</sub>; г) 1000000110,00101<sub>2</sub>; д) 666,16<sub>8</sub>; е) 1С7,68<sub>16</sub>.

### Вариант 10

1. а) 524; б) 222; в) 579,5; г) 847,625; д) 53,35.  
2. а) 101111111<sub>2</sub>; б) 1111100110<sub>2</sub>; в) 10011000,1101011<sub>2</sub>; г) 1110001101,1001<sub>2</sub>; д) 140,22<sub>8</sub>; е) 1DE,54<sub>16</sub>.

### Вариант 11

1. а) 113; б) 875; в) 535,1875; г) 649,25; д) 6,52.  
2. а) 11101000<sub>2</sub>; б) 1010001111<sub>2</sub>; в) 1101101000,01<sub>2</sub>; г) 1000000101,01011<sub>2</sub>; д) 1600,14; е) 1E9,4<sub>16</sub>.

### Вариант 12

1. а) 294; б) 723; в) 950,25; г) 976,625; д) 282,73.  
2. а) 10000011001<sub>2</sub>; б) 10101100<sub>2</sub>; в) 1101100,01<sub>2</sub>; г) 1110001100,1<sub>2</sub>; д) 1053,2<sub>8</sub>; е) 200,6<sub>16</sub>.

### Вариант 13

1. а) 617; б) 597; в) 412,25; г) 545,25; д) 84,82.  
2. а) 1101111 01<sub>2</sub>; б) 1110011101<sub>2</sub>; в) 111001 000,01<sub>2</sub>; г) 1100111001,1001<sub>2</sub>; д) 171,17<sub>8</sub>; е) 3EC,5<sub>16</sub>.

### Вариант 14

1. а) 1047; б) 335; в) 814,5; г) 518,625; д) 198,91.  
2. а) 1101100000<sub>2</sub>; б) 100001010<sub>2</sub>; в) 1011010101,1<sub>2</sub>; г) 1010011111,1101<sub>2</sub>; д) 452,63<sub>8</sub>; е) 1E7,08<sub>16</sub>.

### Вариант 15

1. а) 887; б) 233; в) 801,5; г) 936,3125; д) 218,73.
2. а) 1010100001<sub>2</sub>; б) 10000010101<sub>2</sub>; в) 1011110000,100101<sub>2</sub>; г) 1000110001,1011<sub>2</sub>; д) 1034,34<sub>8</sub>; е) 72,6<sub>16</sub>.

### Вариант 16

1. а) 969; б) 549; в) 973,375; г) 508,5; д) 281,09.
2. а) 10100010<sub>2</sub>; б) 1110010111<sub>2</sub>; в) 110010010,101<sub>2</sub>; г) 1111011100,10011<sub>2</sub>; д) 605,02<sub>8</sub>; е) 3С8,8<sub>16</sub>.

### Вариант 17

1. а) 163; б) 566; в) 694,375; г) 352,375; д) 288,61.
2. а) 1001101001<sub>2</sub>; б) 110011101<sub>2</sub>; в) 1000001101,01<sub>2</sub>; г) 1010001001,11011<sub>2</sub>; д) 247,1<sub>8</sub>; е) 81,4<sub>16</sub>.

### Вариант 18

1. а) 917; б) 477; в) 74,5; г) 792,25; д) 84,33.
2. а) 1110011100<sub>2</sub>; б) 1111101111<sub>2</sub>; в) 111110100,101<sub>2</sub>; г) 110011110,1000011<sub>2</sub>; д) 1446,62<sub>8</sub>; е) 9С, D<sub>16</sub>.

### Вариант 19

1. а) 477; б) 182; в) 863,25; г) 882,25; д) 75,2.
2. а) 101011100<sub>2</sub>; б) 1000010011<sub>2</sub>; в) 11100011,1<sub>2</sub>; г) 100101010,00011<sub>2</sub>; д) 1762,7<sub>8</sub>; е) 1В5,6<sub>16</sub>.

### Вариант 20

1. а) 804; б) 157; в) 207,625; г) 435,375; д) 30,43.
2. а) 10010000<sub>2</sub>; б) 11001010<sub>2</sub>; в) 1110101100,1011<sub>2</sub>; г) 110110101,10111<sub>2</sub>; д) 1164,36<sub>8</sub>; е) 1D5, C8<sub>16</sub>.

### Вариант 21

1. а) 753; б) 404; в) 111,1875; г) 907,0625; д) 62,88.
2. а) 11100011<sub>2</sub>; б) 1111001111<sub>2</sub>; в) 1011111111,01001<sub>2</sub>; г) 1001011101,011<sub>2</sub>; д) 615,72<sub>8</sub>; е) 3DA,5<sub>16</sub>.

### Вариант 22

1. а) 571; б) 556; в) 696,25; г) 580,375; д) 106,67.
2. а) 110011010<sub>2</sub>; б) 111001010<sub>2</sub>; в) 1000010011,00101<sub>2</sub>; г) 11010110,00001<sub>2</sub>; д) 1343,66<sub>8</sub>; е) 3С3,6<sub>16</sub>.

### Вариант 23

1. а) 244; б) 581; в) 351,6875; г) 1027,375; д) 151,44.
2. а) 1001100111<sub>2</sub>; б) 1100010010<sub>2</sub>; в) 1100110010,1101<sub>2</sub>; г) 1001011,0101<sub>2</sub>; д) 171,3<sub>8</sub>; е) 3A3,4<sub>16</sub>.

### Вариант 24

1. а) 388; б) 280; в) 833,5625; г) 674,25; д) 159,05.
2. а) 11001111<sub>2</sub>; б) 101001101<sub>2</sub>; в) 101001101,001001<sub>2</sub>; г) 100101011,101<sub>2</sub>; д) 750,51<sub>8</sub>; е) 90,8<sub>16</sub>.

### Вариант 25

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

2. а)  $11000001_2$ ; б)  $111111110_2$ ; в)  $1110100010,10101_2$ ; г)  $1001011001,011_2$ ;  
д)  $1335,2_8$ ; е)  $18F,8_{16}$ .

**Выполнить арифметические операции с двоичными, восьмеричными и шестнадцатеричным числами**

**Задание 2**

**Вариант 1** Произвести действия над числами в различных системах счисления:

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

Выполнить проверку полученных результатов путем их перевода в десятичную систему счисления

**Вариант 2** Произвести действия над числами в различных системах счисления:

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

Выполнить проверку полученных результатов путем их перевода в десятичную систему счисления

**Вариант 3** Произвести действия над числами в различных системах счисления:

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

Выполнить проверку полученных результатов путем их перевода в десятичную систему счисления

**Вариант 4** Произвести действия над числами в различных системах счисления:

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

Выполнить проверку полученных результатов путем их перевода в десятичную систему счисления

### Варианты заданий для самостоятельной работы.

№	Целое число	Дробное число	Основания систем (для задания 1)	Основания систем (для задания 2)	Основание системы (для задания 3)	Основание системы (для задания 4)
1	2	3	4	5	6	7
1	678	0,343	10-2-16-10	10-2-16-10	3	5
2	267	0,666	10-8-2-10	10-8-2-10	4	6
3	762	0,988	10-16-8-10	10-16-8-10	5	3
4	233	0,685	10-16-2-10	10-16-2-10	6	4
5	288	0,222	10-2-8-10	10-2-8-10	3	5
6	532	0,807	10-8-16-10	10-8-16-10	4	6
7	769	0,879	10-2-16-10	10-2-16-10	5	3
8	797	0,788	10-8-2-10	10-8-2-10	6	4
9	155	0,493	10-16-8-10	10-16-8-10	3	5
10	519	0,433	10-16-2-10	10-16-2-10	4	6

### Контрольные вопросы

1. Что такое система счисления?
2. Что называется основанием системы счисления?
3. Приведите правила перевода чисел из десятичной системы счисления в двоичную.
4. Приведите правила перевода чисел из восьмеричной системы счисления в двоичную.
5. Приведите правила перевода чисел из двоичной системы счисления в шестнадцатеричную.
6. Что называется прямым кодом числа?
7. Что называется обратным кодом числа?
8. Что называется дополнительным кодом числа?

## Практическая работа №2 Кодирование информации в компьютере.

**Тема:** способы кодирования информации.

**Цели:** ознакомится с видами информации, и изучить способы представления ее в ЭВМ.

**Студент должен *знать*:**

- виды информации;
- способы представления символьной информации;
- способы представления графической информации;

**уметь:**

- представлять текстовую, графическую информации в двоичном коде;
- определять количество информации для разных типов сообщений.

### Теоретическая часть

#### 1. Представление символов в вычислительных машинах

В вычислительных машинах символы не могут храниться иначе, как в виде последовательностей бит. Для передачи символа и его корректного отображения ему должна соответствовать уникальная последовательность нулей и единиц. Для этого были разработаны таблицы кодировок.

Количество символов, которые можно задать последовательностью бит длины  $n$ , задается простой формулой  $C(n)=2^n$ . Таким образом, от нужного количества символов напрямую зависит количество используемой памяти.

#### 2. Таблицы кодировок

На заре компьютерной эры на каждый символ было отведено по пять бит. Это было связано с малым количеством оперативной памяти на компьютерах тех лет. В эти 32 символа входили только управляющие символы и строчные буквы английского алфавита.

С ростом производительности компьютеров стали появляться таблицы кодировок с большим количеством символов.

Первой семибитной кодировкой стала ASCII7. В нее уже вошли прописные буквы английского алфавита, арабские цифры, знаки препинания.

Затем на ее базе была разработана ASCII8, в котором уже стало возможным хранение 256 символов.

Существуют следующие кодировки стандарта ASCII (8бит):

- ASCII — первая кодировка, в которой стало возможно использовать символы национальных алфавитов.
- КОИ8-R — первая русская кодировка. Символы кириллицы расположены не в алфавитном порядке. Их разместили в верхнюю половину таблицы так, чтобы позиции кириллических символов соответствовали их фонетическим аналогам в английском алфавите. Это значит, что даже при потере старшего бита каждого символа, например, при проходе через устаревший семибитный модем, текст остается "читаемым".
- CP866 — русская кодировка, использовавшаяся на компьютерах IBM в системе DOS.
- Windows-1251 — русская кодировка, использовавшаяся в русскоязычных версиях операционной системы Windows в начале 90-х годов. Кириллические символы идут в алфавитном порядке. Содержит все символы, встречающиеся в типографике обычного текста (кроме знака ударения).

Таблицы символов ASCII имеют 3 колонки:

- номер символа в десятичном формате
- номер символа в шестнадцатичном формате
- представление самого символа.



Первые от 0 до 127 символа, из 256, всегда одинаковые и отданы под символы латинского алфавита, управляющие символы (такие как переносы строк, табуляция и т.д.) и грамматические символы, а национальный язык использует оставшуюся от 128 по 255 строки таблицы. Каждый символ кодируется 8 битами.

## Принципы кодирования по таблицам стандарта ASCII

Цифры 0-9 представляются своими двоичными значениями (например,  $5=0101_2$ ), перед которыми стоит  $0011_2$ . Таким образом, двоично-десятичные числа (BCD) превращаются в ASCII-строку с помощью простого добавления слева  $0011_2$  к каждому двоично-десятичному полубайту.

Буквы A-Z верхнего и нижнего регистров различаются в своём представлении только одним битом, что упрощает преобразование регистра и проверку на диапазон. Буквы представляются своими порядковыми номерами в алфавите, записанными в двоичной системе счисления, перед которыми стоит  $0100_2$  (для букв верхнего регистра) или  $0110_2$  (для букв нижнего регистра).

Например, закодируем слово «ок» в кодировке ASCII. Символ «о» (англ.) имеет позицию 111 в десятичном виде и  $6F$  в шестнадцатеричном. Переведем это в двоичную систему — **01101111**. Символ «к» (англ.) — позиция 107 в десятичной и  $6B$  в шестнадцатеричной. переводим в двоичную — **01101011**. Итого слово «ок» закодированная в ASCII будет храниться в памяти ПК в виде — **01101111 01101011**. Процесс декодирования будет обратный. Берем по 8 бит, переводим их в десятичную кодировку, получаем номер символа, смотрим по таблице что это за символ.

С ростом интернет национальных кодировок стало очень много. Было два выхода, указывать для каждой страницы кодировки, либо создать одну общую для всех символов в мире таблицу символов. Победил второй вариант, так создали **Unicode** таблицу символов.

Юникод или Уникод (англ. Unicode) — это промышленный стандарт обеспечивающий цифровое представление символов всех письменностей мира, и специальных символов.

Стандарт состоит из двух основных разделов:

- универсальный набор символов (англ. UCS, universal character set)
- семейство кодировок (англ. UTF, Unicode transformation format).

Универсальный набор символов задаёт однозначное соответствие символов кодам — элементам кодового пространства, представляющим неотрицательные целые числа.

Семейство кодировок определяет машинное представление последовательности кодов UCS.

Коды в стандарте Unicode разделены на несколько областей. Область с кодами от  $U+0000$  до  $U+007F$  содержит символы набора ASCII с соответствующими кодами. Далее расположены области знаков различных письменностей, знаки пунктуации и технические символы.

Под символы кириллицы выделены области знаков с кодами от  $U+0400$  до  $U+052F$ , от  $U+2DE0$  до  $U+2DFF$ , от  $U+A640$  до  $U+A69F$ . Часть кодов зарезервирована для использования в будущем.

Unicode — это таблица символов (не кодировка, а именно таблица символов). Она состоит из 1 114 112 позиций. Большинство этих позиций пока не заполнены символами, зарезервированы на будущее.

Таблица разделена на 17 блоков, по 65 536 символов в каждом. Каждый блок содержит свою группу символов.

Нулевой блок — базовый, там собраны наиболее употребляемые символы всех современных алфавитов.

Во втором блоке находятся символы вымерших языков. Есть два блока отведенные под частное использование. Большинство блоков пока не заполнены.

Третья зарезервирована для архаичных китайских иероглифов. 15 и 16 блоки выделены для частного употребления. Остальные в резерве.

Итого емкость символов юникода составляет от 0 до 10FFFF (в шестнадцатичном виде).

## Принципы кодирования по таблицам стандарта Unicode

1. Если размер символа в кодировке UTF-8 = 1 байт

Код имеет вид (0aaa aaaa), где «0» — просто ноль, остальные биты «а» — это код символа в кодировке ASCII;

2. Если размер символа в кодировке в UTF-8 >1 байта (то есть от 2 до 6):

Первый байт содержит количество байт символа, закодированное в единичной системе счисления;

- 2 – 11
- 3 – 111
- 4 – 1111
- 5 – 1111 1
- 6 – 1111 11

Символ «0» — бит означающий завершение кода размера

Далее идут значащие байты кода, которые имеют вид (10xx xxxx), где «10» — биты признака продолжения, а «х» — значащие биты.

В общем случае варианты представления одного символа в кодировке UTF-8 выглядят так:

- 1 байт – 0aaa aaaa
- 2 байта – 110x xxxx 10xx xxxx
- 3 байта – 1110 xxxx 10xx xxxx 10xx xxxx
- 4 байта – 1111 0xxx 10xx xxxx 10xx xxxx 10xx xxxx
- 5 байт – 1111 10xx 10xx xxxx 10xx xxxx 10xx xxxx 10xx xxxx
- 6 байт – 1111 110x 10xx xxxx 10xx xxxx 10xx xxxx 10xx xxxx 10xx xxxx

Определение длины кода в UTF-8

Количество байт UTF-8	Количество значащих бит
1	7
2	11
3	16
4	21
5	26
6	31

В общем случае количество значащих бит  $C$ , кодируемых  $n$  байтами UTF-8, определяется по формуле:

$$C = 7 \text{ при } n=1$$

$$C = n \cdot 5 + 1 \text{ при } n > 1$$

Для вставки символов в текст часто используют альт код. Альт-код — это символы, которые выводятся при нажатии комбинации клавиш на клавиатуре Alt + X, где X это набор цифр (определенного числа) на NumPad'e.

NumPAD это блок цифр на клавиатуре как правило расположенных справа. Символы альт на ПК с операционными системами Windows не доступны при использовании обычной клавиатуры. Их можно вызывать только при помощи дополнительных команд.

Зажимаем кнопку ALT и набираем число в десятичной системе счисления. Надеюсь здесь все понятно. Таблица Alt кодов поможет вам сориентироваться и найти нужный символ.

## Принцип кодирования

Существует равномерное и неравномерное кодирование. При равномерном кодировании сообщение декодируется однозначно.

При неравномерном кодировании для однозначного декодирования сообщения нужно, чтобы выполнялось прямое и обратное условие Фано (прямое: никакой код не должен быть началом другого кода, обратное: никакой код не должен быть концом другого кода)

Понимать, что мы можем закодировать сообщение, даже если условие Фано не выполняется, но возможно не сможем его однозначно декодировать.

Однозначно декодировать – получить один единственный точный вариант.

**Пример равномерного кодирования.** Пусть передано сообщение 1100000100110 в котором закодировано некая последовательность 5 букв латинского алфавита с заданными для них двоичные коды (для некоторых букв - из двух бит, для некоторых - из трех). Нужно раскодировать полученное сообщение. Коды представлены в таблице:

a	b	c	d	e
000	110	01	001	10

Решение.

1. Для начала убеждаемся, что выполняется условие Фано: никакое кодовое слово не является началом другого кодового слова, поэтому однозначно можем раскодировать сообщение с начала.
2. Разобьём код слева направо ориентируясь на данные таблицы: 110\ 000\ 01\ 001\ 10.
3. Подставим значение каждого кода получим b\ a \c \d \e.

**Пример неравномерного кодирования.** Для 6 букв латинского алфавита заданы их двоичные коды (для некоторых букв из двух бит, для некоторых – из трех). Эти коды представлены в таблице:

A	B	C	D	E	F
00	100	10	011	11	101

Определите, какая последовательность из 6 букв закодирована двоичной строкой 011111000101100.

Решение.

1. Условия Фано и обратное условие Фано не выполняются, значит код можно раскодировать неоднозначно.
2. Поэтому пробуем разные варианты, отбрасывая те, в которых получаются повторяющиеся буквы:
  - a. Разобьём код слева направо 011\ 11 \100 \0101100
  - b. Первая буква определяется однозначно D код 011
  - c. Вторая буква также определится однозначно как E код 11
  - d. Далее варианты. Пусть третья буква B код 100, тогда следующая начинается с кода 010, но таких букв в таблице нет, значит деление не верно.
  - e. Разобьём код слева направо с учетом пункта d: 011\ 11\ 10\ 00 \101 \100
  - f. Третья буква — C код 10, потом — A код 00, тогда следующая буква — F код 101, и последняя — B код 100.
3. Окончательно получили ответ: DECAFB.

Одним из центральных приёмов при решении задач двоичного кодирования является построение дерева Фано. **Дерево Фано** для двоичного кодирования начинается с двух направлений, которые означают 0(ноль) и 1(единицу) (цифры двоичного кодирования). От каждого направления можно также рисовать только два направления: 0(ноль) и 1(единицу) и т.д. Для удобства рисуют 1(единицу) только вправо, а 0(ноль) только влево. Получается структура похожая на дерево!

В конце каждой ветки можно располагать букву, которую мы хотим закодировать, но если мы расположили букву, от этой ветки больше нельзя делать новых ответвлений.

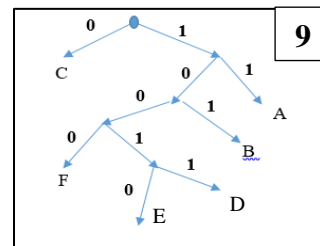
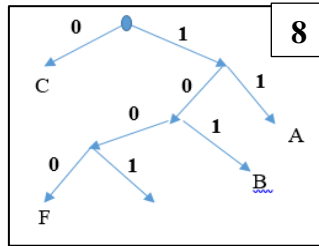
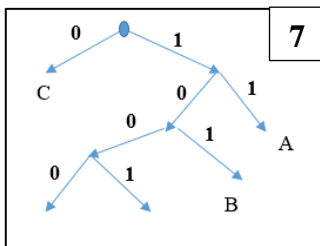
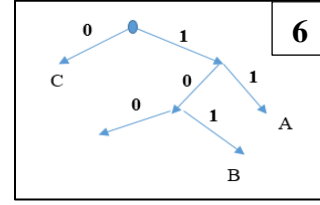
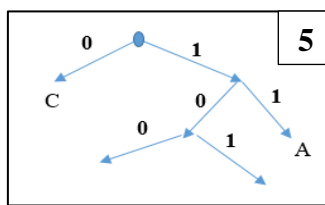
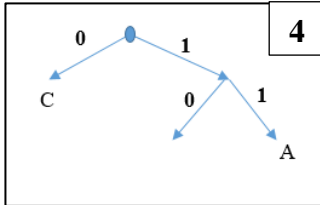
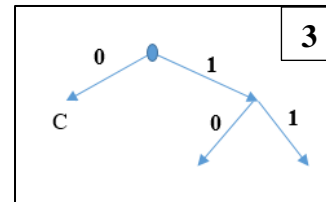
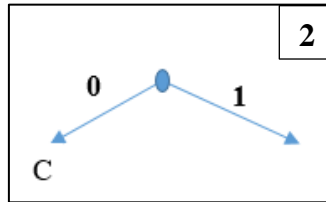
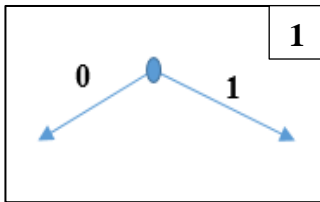
Такой подход позволяет однозначно декодировать сообщение, состоящее из этих букв.

**Пример построения дерева Фано.**

Дана таблица 5-ти символов. 3 символа имеют двоичные коды для кодирования. Требуется определить коды еще 3-м символам.

A	B	C	D	E	F
11	101	0			

Решение.



1. Из начальной точки строим две линии левую обозначаем числом 0, правую 1
2. Линия 0 заблокируется так как у нас есть код 0 для буквы С
3. От конца линии 1 делаем новое ветвление влево 0 вправо 1
4. Путь от начальной точки 1-1 в конце также блокируется буквой А с кодом 11
5. В конце пути от начальной точки 1-0 выполним ветвление по установленному правилу
6. Путь от начальной точки 1-0-1 в конце также блокируется буквой В с кодом 101
7. В конце пути от начальной точки 1-0-0 выполним ветвление по установленному правилу
8. Путь от начальной точки 1-0-0-0 в конце можно присвоить код для новой буквы, например, F, ее код будет 1000, занесем его в таблицу.
9. Для присвоения двум оставшимся буквам кодов необходимо в конце пути от начальной точки 1-0-0-1 выполним ветвление по установленному правилу.

10. В конце пути от начальной точки 1-0-0-1-1 ставим букву D и ее код будет 10011, в конце пути от начальной точки 1-0-0-1-0 ставим букву E и ее код будет 10010. Окончательно заполненная таблица

A	B	C	D	E	F
11	101	0	10011	10010	1000

11. Для уменьшения количество двоичных разрядов при кодировании букв нужно использовать две ветки начального дерева, в нашем примере буква С должна кодироваться двумя цифрами, например, 01.

### Представление графической и ауди информации.

Все многообразие красок на экране получается путем смешивания трех базовых цветов: красного, синего и зеленого. Каждый пиксель на экране состоит из трех близко расположенных элементов, светящихся этими цветами. Если все три цвета имеют одинаковую интенсивность, то из их сочетаний можно получить 8 различных цветов.

Большое количество цветов получается при раздельном управлении интенсивностью базовых цветов. Причем интенсивность может иметь более двух уровней, если для кодирования каждого из базовых цветов выделять больше одного бита.

Число цветов, воспроизводимых на экране дисплея (K) и число бит, отводимых в видеопамяти по каждый пиксель (b), связаны формулой:

$$K=2^b$$

**Пиксель** - наименьший элемент изображения на экране (точка на экране)

**Растр** - прямоугольная сетка пикселей на экране.

**Разрешающая способность экрана** - размер сетки растра, задаваемого в виде произведения M\*N, где M-число точек по горизонтали, N- число точек по вертикали.

**Пример решение задач**

1. Сколько бит видео памяти занимает информация об одном пикселе на черно-белом экране ( без полутонов).

**Решение:** Для черно - белого экрана без полутонов K=2. Следовательно,  $2^b=2$ . Отсюда - N=1 бит на пиксель.

2. Какой объем видеопамяти необходим для хранения двух страниц изображения при условии, сто разрешающая способность дисплея равна 640\*350 пикселей, а количество используемых цветов -16.

**Решение:** Найдем битовую глубину данного изображения: K=16. Отсюда  $2^N=16$ , N=4. Можно вывести формулу для расчета объема видеопамяти графического файла: (разрешение экрана) \*(битовая глубина N) \*(количество страниц)/8  $640*350*4*2/8=218,75$  Кбайт

Ответ: объем видеопамяти необходимый для хранения двух страниц изображения составляет 218,75 Кбайт

3. Физическая природа звука - колебания в определенном диапазоне частот, передаваемые звуковой волной через воздух.

**Аудиоадаптер** (звуковая плата) - специальное устройство, подключаемое к компьютеру, предназначенное для преобразования электрических колебаний звуковой частоты в числовой двоичный код при вводе звука и для обратного преобразования при воспроизведении.

В процессе записи звука аудиоадаптер с определенным периодом измеряет амплитуду электрического тока и заносит в регистр двоичный код полученной величины. Качество компьютерного звука определяется характеристиками аудиоадаптора: частотой дискретизации и разрядностью.

**Частота дискретизации**- это количество измерений входного сигнала за 1 секунду. Частота измеряется в герцах (Гц). Характерные частоты дискретизации: 11 кГц, 22кГц, 44,1кГц.

**Разрядность регистра** -число бит в регистре аудиоадаптера. Разрядность определяет точность измерения входного сигнала. Если разрядность равна 8 (16), то при измерении входного сигнала может быть получен  $2^8=256$  различных значений.

**Звуковой файл**- файл, хранящий звуковую информацию в числовой двоичной форме.

**Пример решения задач:**

1. Определить размер (в байтах) цифрового аудиофайла, время звучания которого составляет 100 секунд при частоте дискретизации 22,05кГц и разрешении 8 бит. Решение: Формула для расчета размера (в байтах) цифрового аудиофайла (монофонического звучания):

$$(частота дискретизации в Гц)*(время записи в секундах)*(разрешение в битах)/8.$$

Таким образом, размер файла вычисляется так:  $22050*10*8/8=220500$  байт.

**Практическая часть**

**Задание 1.** Ниже закодировано слово. Определите вид кодировки и декодируйте слово. (смотри таблицы кодировок):

Hex	D1	E8	F1	F2	E5	EC	E0
Dec	209	232	241	242			
Bin							
Символ	С	и	с	т			

**Задание 2.** Ниже закодировано одно слово. Определите вид кодировки и декодируйте слово. (смотри таблицы кодировок):

Hex	438	43D	444	43E	440	43C	430	446	438	44E
Dec										

Bin															
Сим-вол															

**Задание 3.** Откройте файлы с таблицами кодировок, прилагаемых к практическому заданию.

Заполните таблицы кодов всех букв, входящих в слова: Штора, Лампа, Карта, Ручка, Стена, по стандартам ASCII, UTF-8 и КОИ-8, используя таблицы кодировок.

Сравните коды букв. Сделать вывод и записать его в тетрадь.

ASCII.

Сим-вол															
Hex															
Dec															
Bin															

UTF-8.

Сим-вол															
Hex															
Dec															
Bin															

КОИ-8.

Сим-вол															
Hex															
Dec															
Bin															

**Задание 4.** Определить тип кодировки через командную строку ОС Windows 7. Написать синтаксис команды для смены кодировки через командную строку ОС Windows 7

**Задание 5.** Открыть окно «Таблица символов» через команду «charmap.exe».

1. Определить коды символов, указанных в таблице в различных шрифтах через окно «Таблица символов».

Символ	У	о	д	h	o	l
Arial						
Times New Roman						
SimSun						

2. Создать текстовый файл в редакторе Word. Введите текст: «Учусь кодировать» и «hello house». 3. Скопируйте фразу 3 раза и отформатируйте их разным шрифтом из таблицы.

4. Используя сочетание клавиш «Alt+X» на клавиатуре определите коды указанных букв в таблице. Заполните таблицу.

Символ	У	о	д	h	o	l
Arial						
Times New Roman						
SimSun						

5. Сравните две таблицы и сделайте вывод. Вывод записать в тетрадь.

**Задание 6.** Выполните следующие действия:

1. Открыть текстовый редактор «Блокнот» и определить тип кодировки по умолчанию. Набрать фразу: «Учусь кодировать» и «hello house» сохранить файл с этой фразой в кодировке ANSI, используя окно «Сохранить как» текстовый редактор «Блокнот».
2. Открыть сохраненный текстовый файл в редакторе Word и записать что предлагается выполнить.
3. Сохранить в редакторе Word в формате KOI-U. Записать что предлагается выполнить и каковы результаты выполненных действий.

4. Открыть сохраненный файл в текстовом редакторе «Блокнот». Записать что происходит при его открытии.
5. Создать текстовый файл с фразой: «Учусь кодировать» и «hello house» в редакторе Word. Сохранить с расширением txt. (как обычный тест). Записать что произойдет в ходе сохранения
6. Открыть созданный файл в редактор «Блокнот». Записать что происходит при его открытии.
7. Записать вывод о проделанных действиях и их результатах.

**Задание 7.** Создать текстовый файл в редакторе Word. Используя таблицу «Alt – кодов» создайте текст с вставленными символами как показано: ♥→сердце клиента, ☀ освещение помещения, ¶ символ абзаца, @ знак электронной почты, √ знак выполнения, ☒ затемнение на 25%, ¯ отрицание А.

**Задание 8.** Для 5 букв латинского алфавита заданы их двоичные коды (для некоторых букв - из двух бит, для некоторых - из трех). Эти коды представлены в таблице:

a	b	c	d	e
000	110	01	001	10

Определите, какой набор букв закодирован двоичной строкой 1100000100110

**Задание 9.** Для кодирования сообщения, состоящего только из букв А, В, С, D и Е, используется неравномерный по длине двоичный код:

A	B	C	D	E
000	11	01	001	10

Какое (только одно!) из четырех полученных сообщений было передано без ошибок и может быть декодировано: 1) 110000010011110; 2) 110000011011110; 3) 110001001001110; 4) 110000001011110

**Задание 10.** Для передачи по каналу связи сообщения, состоящего только из букв А, Б, В, Г, решили использовать неравномерный по длине код: А=0, Б=100, В=101. Как нужно закодировать букву Г, чтобы длина кода была минимальной и допускалось однозначное разбиение кодированного сообщения на буквы? (использовать дерево Фано). 1) 1; 2) 11; 3) 01; 4) 010;

**Задание 11.** Для передачи по каналу связи сообщения, состоящего только из символов А, Б, В и Г используется посимвольное кодирование: А-00, Б-11, В-010, Г-011. Через канал связи передается сообщение: ВАГБГВ. Закодируйте сообщение данным кодом. Полученную двоичную последовательность переведите в шестнадцатеричный вид.

### Самостоятельное задание

*Задание №1 по определению информационного объема сообщения.*

1. Метеорологическая станция ведет наблюдение за влажностью воздуха. Результатом одного измерения является целое число от 0 до 100 процентов, которое записывается при помощи минимально возможного количества бит. Станция сделала 80 измерений. Определите информационный объем результатов наблюдений.
2. В велокроссе участвуют 678 спортсменов. Специальное устройство регистрирует прохождение каждым из участников промежуточного финиша, записывая его номер с использованием минимально возможного количества бит, одинакового для каждого спортсмена. Каков информационный объем сообщения, записанного устройством, после того как промежуточный финиш прошли 200 велосипедистов?
3. В некоторой стране автомобильный номер длиной 6 символов составляется из заглавных букв (всего используется 19 букв) и десятичных цифр в любом порядке. Каждый символ кодируется одинаковым и минимально возможным количеством бит, а каждый номер – одинаковым и минимально возможным количеством байт. Определите объем памяти, необходимый для хранения 40 автомобильных номеров.

4. Два текста содержат одинаковое количество символов. Первый текст составлен в алфавите мощностью 16 символов, а второй текст – в алфавите из 256 символов. Во сколько раз количество информации во втором тексте больше, чем в первом?
5. Для регистрации на сайте некоторой страны пользователю необходимо придумать пароль длиной ровно 15 символов. В пароле можно использовать десятичные цифры и 11 различных символов местного алфавита, причем все буквы используются в двух начертаниях – строчные и прописные. Каждый символ кодируется одинаковым и минимально возможным количеством бит, а каждый пароль – одинаковым и минимально возможным количеством байт. Определите объем памяти, необходимый для хранения 30 паролей.
6. Объем сообщения – 7,5 Кбайт. Известно, что данное сообщение содержит 7680 символов. Какова мощность алфавита?
7. Дан текст из 600 символов. Известно, что символы берутся из таблицы размером 16 на 32. Определите информационный объем текста в битах.
8. В школе 800 учащихся, коды учащихся записаны в школьной информационной системе с помощью минимального количества бит. Каков информационный объем сообщения о кодах 320 учащихся, присутствующих на конференции?

*Задание №2 по кодированию информации.*

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

a	b	c	d	e
100	110	011	01	10

Определите, какой набор букв закодирован двоичной строкой 1000110110110, если известно, что все буквы в последовательности – разные:

2. Для передачи по каналу связи сообщения, состоящего только из букв А, Б, В, Г, решили использовать неравномерный по длине код: А=1, Б=01, В=001. Как нужно закодировать букву Г, чтобы длина кода была минимальной и допускалось однозначное разбиение кодированного сообщения на буквы? Построить дерево Фано.

3. Для передачи по каналу связи сообщения, состоящего только из букв А, Б, В, Г, решили использовать неравномерный по длине код: А=0, Б=100, В=101. Как нужно закодировать букву Г, чтобы длина кода была минимальной и допускалось однозначное разбиение кодированного сообщения на буквы?

4. Для кодирования сообщения, состоящего только из букв О, К, Л, М и Б, используется неравномерный по длине двоичный код:

О	К	Л	М	Б
00	01	11	010	0110

Какое (только одно!) из четырех полученных сообщений было передано без ошибок и может быть декодировано: 1) 110001001001110 2) 10000011000111010 3) 110001001101001 4) 1000110001100010

5. Для передачи по каналу связи сообщения, состоящего только из букв А, Б, В, Г, решили использовать неравномерный по длине код: А=01, Б=1, В=001. Как нужно закодировать букву Г, чтобы длина кода была минимальной и допускалось однозначное разбиение кодированного сообщения на буквы? Построить дерево Фано.



6. Для передачи по каналу связи сообщения, состоящего только из символов А, Б, В и Г, используется неравномерный (по длине) код: А-10, Б-11, В-110, Г-0. Через канал связи передается сообщение: ВАГБААГВ. Закодируйте сообщение данным кодом. Полученное двоичное число переведите в восьмеричный вид.

7. По каналу связи передаются сообщения, содержащие только шесть букв: А, В, С, D, E, F. Для передачи используется неравномерный двоичный код, удовлетворяющий условию Фано. Для букв А, В, С используются такие кодовые слова: А — 011, В — 101, С — 11. Укажите код для оставшихся букв с наименьшей возможной длиной. Построить дерево Фано.

8. Для кодирования букв О, В, Д, П, А решили использовать двоичное представление чисел 0, 1, 2, 3 и 4 соответственно (с сохранением одного незначащего нуля в случае одноразрядного представления). Если закодировать последовательность букв ВОДОПАД таким способом и результат записать восьмеричным кодом.

### Контрольные вопросы

1. Какие разновидности данных могут быть записаны в памяти компьютера?
2. Что называется форматом данных?
3. Как в компьютерах кодируются текстовые данные?
4. Для чего используются кодовые таблицы?
5. Какие кодовые таблицы вам известны?
6. Как в компьютерах кодируется графическая информация?
7. Дайте определения понятий «пиксель», «растр», «разрешающая способность», «сканирование».
8. Опишите принципы кодирования монохромных изображений.
9. Опишите принципы кодирования цветных изображений.
10. Какие форматы используются для представления числовых данных в компьютерах?
11. Чем отличаются с точки зрения информатики целые числа от вещественных?
12. Какие поля могут быть использованы для хранения целых чисел?

### Варианты контрольных работ

#### Вариант 1

1. Объем оперативной памяти компьютера содержит 163 840 машинных слов, что составляет 0,625 Мбайт. Сколько бит содержит каждое машинное слово?
2. Современный монитор позволяет получить на экране 262144 различных цветов, сколько бит памяти занимает один пиксель.
3. На экране с разрешающей способностью 640\*200 высвечивается только двухцветное изображение. Какой минимальный объем видеопамати необходим для хранения изображения.
4. Объем свободной памяти на диске - 6 Мб, разрядность звуковой платы -8. Какова длительность звучания цифрового аудиофайла, записанного с частотой дискретизации 22,05 кГц?
5. Цифровой аудиофайл содержит запись звука низкого качества. Какова длительность звучания файла, если его объем составляет 360 Кб?

#### Вариант 2.

1. Объем сообщения, содержащего 2048 символов, составил 1/512 часть Мбайта. Каков размер алфавита, с помощью которого записано сообщение (количество бит для кодирования одного символа)
2. Объем видео памяти равен 1024 Кбайт, количество используемых цветов - 64. Вычислить варианты разрешающей способности дисплея при условии, что число страниц может быть равно 1,2 и 4.
3. Видео память имеет объем, в котором храниться 8-х цветное изображение размером 650\*480. Какого размера изображение может храниться в этом же объеме видеопамати, если использовать 64- цветную палитру?
4. Одна минута записи цифрового аудиофайла занимает на диске 1,5 Мб, разрядность звуковой платы — 16. С какой частотой дискретизации записан звук?

5. Две минуты записи цифрового аудиофайла занимают на диске 8,8 Мб. Частота дискретизации - 22050 Гц. Какова разрядность аудиоадаптера?

### **Вариант 3.**

1. Объем оперативной памяти компьютера составляет 1/8 часть Мбайта. Сколько машинных слов составляют оперативную память, если одно машинное слово содержит 64 бита.
2. Какой объем видеопамати необходим для хранения четырех страниц изображения, если битовая глубина равна 16, а разрешающая способность экрана - 830\*640 пикселей.
3. Объем видео памяти равен 1.5 МБ, битовая глубина - 16, разрешающая способность дисплея - 800\*600. Какое максимальное количество цветов можно использовать при условии, что видеопамать делится на две страница.
4. Каков объем памяти требуется для хранения цифрового аудиофайла с записью звука высокого качества при условии, что время звучания составляет 4.8 минуты?
5. Объем свободной памяти на диске - 0,1 Гб, разрядность звуковой платы - 16. Какова длительность звучания цифрового аудиофайла, записанного с частотой дискретизации 44100 Гц?

### **Вариант 4.**

1. Сообщение занимает 3 страницы по 25 строк. В каждой строке записано по 60 символов. Сколько символов в использованном алфавите, если все сообщение содержит 1125 байтов?
2. Объем видео памяти равен 4 Мб, битовая глубина - 64, разрешающая способность дисплея - 640\*800. Какое максимальное количество страниц может использовать при этих условиях?
3. Видеопамать имеет объем, в котором может храниться 2-х цветное изображение размером 300\*200. Какого размера изображение может храниться в том же объеме видеопамати, если оно будет использовать 16-цветную палитру?
4. Объем свободной памяти на диске — 10,5 Мб, разрядность звуковой платы - 16. Какова длительность звучания цифрового аудиофайла, записанного с частотой дискретизации 44,1 кГц?
5. Пять минуты записи цифрового аудиофайла занимают на диске 5.4 Мб. Частота дискретизации - 22050 Гц. Какова разрядность аудиоадаптера?

### **Контрольные вопросы**

1. Какой используется алфавит в компьютере для представления символьной информации?
2. Расскажите о таблице кодировки под названием ASCII.
3. Какую форму представления символьной информации используют на практике?
4. Что такое разрешающая способность экрана?
5. Что такое видеопамать?
6. Как связаны число цветов, воспроизводимых на экране и число бит, обводимых под каждый пиксель?
7. Что называется аудиоадаптером?
8. Что такое частота дискретизации?

## Практическая работа №3 Логической основой компьютера

**Тема:** Изучение логических функций и их свойства.

**Цель работы:** познакомиться с основными логическими элементами: И, ИЛИ, НЕ, ИЛИ-НЕ, И-НЕ; изучить режимы работы логических элементов; строить схемы из логических элементов по заданному аналитическому выражению функции.

**Студент должен знать**

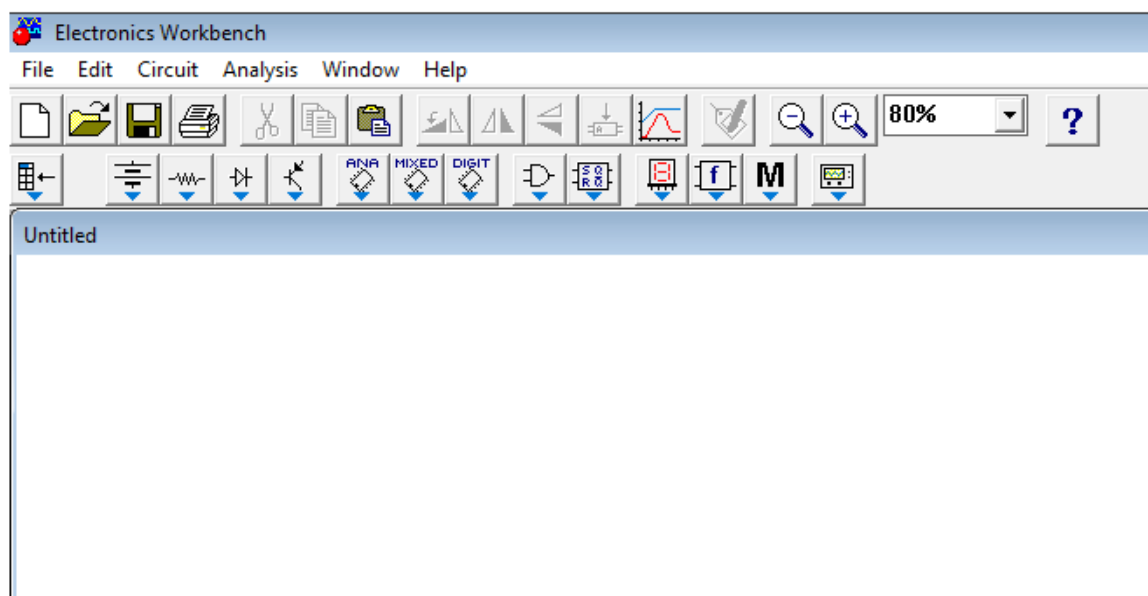
- основные логические элементы;
- таблицы истинности логических элементов;

**уметь**

- объяснять режимы работы логических элементов;
- строить логические схемы.

### Теоретическое обоснование


При построении логических схем будет использоваться программа моделирования электронных схем Electronics Workbench. (рис.3.1).



**Рисунок 3.1** Программа Electronics Workbench.

При построении цифровых схем нам в первую очередь понадобятся следующие элементы:



Заземление, имеющее напряжение 0 В и представляющее нулевое значение двоичного сигнала для цифровых схем. Находится в наборе пассивных компонентов, выбираемом нажатием кнопки . В схеме, собранной с помощью Electronics Workbench, как и практически для любой реальной схемы, требуется указать точку нулевого потенциала, относительно которой определяются напряжения во всех других точках схемы. Именно для этой цели служит элемент заземление. Его единственный вывод подключается к той точке схемы, потенциал которой принимается равным нулю. Допускается и даже целесообразно, особенно для сложных схем, использовать несколько элементов заземления. При этом считается, что все точки, к которым подсоединены заземления, имеют один общий потенциал, равный нулю.




Источник напряжения 5 В, представляющий единичное значение двоичного сигнала для цифровых схем.

[Space]



Переключатель. Находится в наборе переключательных компонентов, выбираемом нажа-

тием кнопки . Этот переключатель допускает два возможных положения, в которых один общий вход соединяется с одним из двух возможных выходов. По умолчанию переключение осуществляется клавишей [Space]. Чтобы назначить какому-либо переключателю другую клавишу, нужно дважды щелкнуть мышью на этом переключателе, ввести требуемый символ в появившемся диалоговом окне и нажатием кнопки "Ассерт" подтвердить сделанный выбор. После этого переключение данного переключателя будет осуществляться с помощью выбранной клавиши.



Индикатор, позволяющий определить значение цифрового сигнала в данной точке схемы. Индикатор загорается при единичном значении сигнала и гаснет при нулевом значении.

Для реализации входного двоичного сигнала, как правило, используется триада, состоящая из источника напряжения, заземления и переключателя, включенных по схеме, представленной на рисунке. Выбор того или иного значения двоичного сигнала в такой схеме осуществляется изменением положения переключателя, для чего нужно нажать соответствующую клавишу на клавиатуре. Индикатор показывает получаемое при этом значение двоичного сигнала, то есть индикатор горит при единичном значении двоичного сигнала и гаснет при нулевом значении. (рис. 3.2).

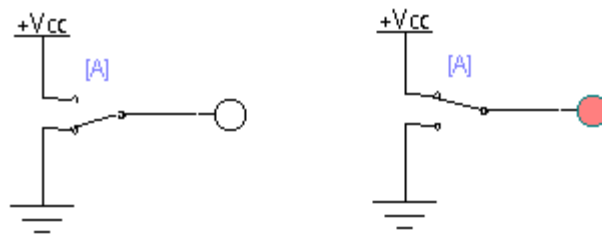


Рисунок 3.2 . Реализация двоичного сигнала и его индикация

Основой построения ЭВМ является алгебра логики, которая оперирует двумя понятиями: истинности и ложности высказывания. Условие истинности обозначают логической единицей "1", а ложности – логическим нулем "0". Логические переменные, принимающие значение 0 или 1, могут образовывать логические функции. Значения аргументов и логических функций удобно изображать в виде таблицы соответствия всех возможных комбинаций входных логических переменных и выходной логической функции, называемой таблицей истинности. Каждой логической функции соответствует свой логический элемент. Основные логические функции и соответствующие им логические элементы в САП "ELECTRONICS WORKBENCH V5.12" приведены в таблице 3.1.

Таблица 3.1. Логические элементы WorkBench

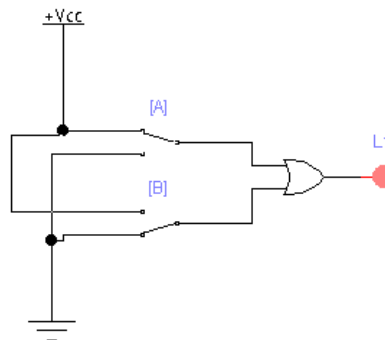
Функция	Название функции	X <sub>1</sub>	0	0	1	1	Элемент, соответствующий функции в EWB
		X <sub>2</sub>	0	1	0	1	
X <sub>1</sub> ∧X <sub>2</sub>	Конъюнкция – логическое умножение (И)		0	0	0	1	
X <sub>1</sub> ∨X <sub>2</sub>	Дизъюнкция – логическое сложение (ИЛИ)		0	1	1	1	

$x_1 \sim x_2$	Эквивалентность	1	0	0	1	
$x_1 \neq x_2$	Неравнозначность (сложение по модулю два)	0	1	1	0	
$x_1   x_2$	Штрих Шеффера (И-НЕ)	1	1	1	0	
$x_1 \downarrow x_2$	Стрелка Пирса (ИЛИ-НЕ)	1	0	0	0	
$\overline{x_1}$	Инверсия (НЕ)	1	1	0	0	

**Последовательность выполнения логических операций определяется следующими правилами:**

1. В первую очередь выполняется операция отрицания, поскольку она имеет наивысший приоритет.
2. Операция логического умножения имеет приоритет над операцией логического сложения.
3. Для задания требуемой последовательности действий могут использоваться скобки, при этом в первую очередь вычисляются выражения, стоящие в скобках.

**Пример 1.** Собрать схему для конъюнкции и заполнить таблицу истинности. (рис. 3.3).



**Рисунок 3.3** Схема для дизъюнкции.

Включив схему, поочередно переключаем А и В, наблюдаем за лампой L1: если на выходе лампа загорается, то на выходе 1, если лампа не горит, то на выходе 0. Таким образом заполняем таблицу истинности для логического элемента ИЛИ.

A	B	L1
0	0	0
0	1	1
1	0	1
1	1	1

**Пример 2.** Собрать схему для логического выражения NOT(A OR B) AND NOT C и заполнить таблицу истинности. (рис. 3.4).

Для удобства логическое выражение перепишем в более удобный для нас вид:

$$\overline{(A + B)} * \bar{C}$$

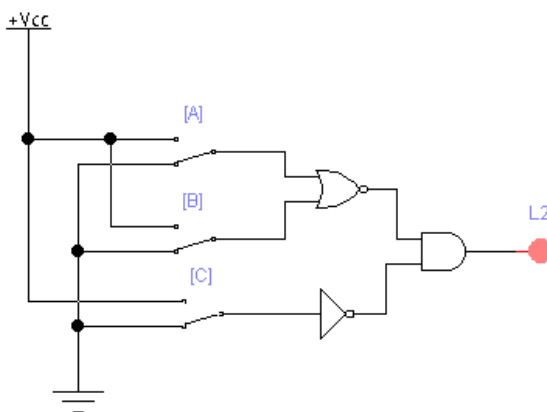


Рисунок 3.4 Схема для логического выражения примера 2

A	B	C	L2
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

Из таблицы состояний для логической схемы делаем вывод, что на выходе будет 1, когда все три переключателя выключены. Во всех других случаях на выходе будет 0.

### Практические задания

**Задание 1.** Собрать схемы для проведения анализа логических элементов (согласно варианту, табл.3.2.). Задать все возможные наборы аргументов  $x_1$ ,  $x_2$ , и получить выходные значения сигналов в виде временной диаграммы на логическом анализаторе.

Таблица 3.2.

№ варианта	Элементы		
1	Сложение по модулю 2	Штрих Шеффера	Стрелка Пирса
2	Инверсия	Дизъюнкция	Эквивалентность
3	Стрелка Пирса	Инверсия	Сложение по модулю 2
4	Эквивалентность	Дизъюнкция	Стрелка Пирса
5	Эквивалентность	Сложение по модулю 2	Штрих Шеффера
6	Сложение по модулю 2	Штрих Шеффера	Стрелка Пирса
7	Инверсия	Дизъюнкция	Эквивалентность
8	Стрелка Пирса	Инверсия	Сложение по модулю 2
9	Эквивалентность	Дизъюнкция	Стрелка Пирса
10	Эквивалентность	Сложение по модулю 2	Штрих Шеффера

## Задание 2

Построить логические схемы по формулам. Сделать минимизацию логических функций по возможности. Схемы реализовать на компьютере в программе и заполнить таблицу истинности.

1.  $A \text{ AND } B \text{ OR NOT } C$ ;
2.  $A \text{ AND NOT } B \text{ OR } C$ ;
3.  $\text{NOT}(A \text{ AND NOT } B) \text{ OR } C$ ;
4.  $A \text{ OR NOT } B \text{ AND } C$ ;
5.  $A \text{ OR NOT}(\text{NOT } B \text{ AND } C)$ ;
6.  $\text{NOT}(A \text{ OR } B) \text{ AND NOT } C$ ;
7.  $\text{NOT}(A \text{ AND } B) \text{ OR NOT } C$ ;
8.  $\text{NOT } A \text{ OR } B \text{ AND } C$ ;
9.  $\text{NOT}(\text{NOT } A \text{ OR } B \text{ OR } C)$ ;
10.  $\text{NOT}(\text{NOT } A \text{ OR } B \text{ AND NOT } C)$ .

## Контрольные вопросы

1. Назначение и использование индикаторов в цифровых схемах.
2. Сформулируйте основные законы алгебры логики.
3. Составьте таблицу истинности для функции трех аргументов:  
а) конъюнкции, б) дизъюнкции, в) ИЛИ-НЕ, г) И-НЕ.
4. Для приведенных схем составить таблицы истинности и записать связь между выходной и входными переменными с помощью операций булевой алгебры.

## Практическая работа №4 Разработка программ для машины Тьюринга

**Тема:** Формирование умений и навыков по разработке программ машин Тьюринга.

**Цель:**

- Изучить устройство машины Тьюринга
- научиться читать и выполнять программы, написанные для машины Тьюринга
- научиться разрабатывать программы для машины Тьюринга.

### Теоретическая часть

**Методические указания:**

**Программы** для машин Тьюринга записываются в виде таблицы, где первый столбец и строка содержат буквы внешнего алфавита и возможные внутренние состояния автомата (внутренний алфавит). Содержимое таблицы представляет собой команды для машины Тьюринга.

Буква, которую считывает головка в ячейке (над которой она находится в данный момент), и внутренне состояние головки определяют, какую команду нужно выполнить. Команда определяется пересечением символов внешнего и внутреннего алфавитов в таблице.

Чтобы задать конкретную машину Тьюринга, требуется **описать для нее следующие составляющие:**

#### 1. Внешний алфавит.

Конечное множество (обозначают буквой  $A$ ), элементы которого называются буквами (символами). Одна из букв этого алфавита (например,  $a_0$ ) должна представлять собой пустой символ. Например, алфавит машины Тьюринга, работающей с двоичными числами, задается в виде  $A = \{0, 1, a_0\}$ . Непрерывную цепочку символов на ленте называют словом.

Автоматом называют устройство, работающее без участия человека. Автомат в машине Тьюринга имеет несколько состояний и при определенных условиях переходит из одного состояния в другое. Множество состояний автомата называют внутренним алфавитом.

#### 3. Внутренний алфавит.

**Конечное множество состояний каретки (автомата).** Обозначается буквой  $Q = \{q_1, q_2, \dots\}$ . Одно из состояний -  $q_1$  - должно быть начальным (запускающим программу). Еще одно из состояний ( $q_0$ ) должно быть конечным (завершающим программу) – состояние остановка.

**Таблица переходов. Описание поведения автомата (каретки) в зависимости от состояния и считанного символа.**

Автомат машины Тьюринга в процессе своей работы управляется программой, во время каждого шага которой выполняются последовательно следующие **действия:**

- Записывать символ внешнего алфавита в ячейку (в том числе и пустой), заменяя находившийся в ней (в том числе и пустой).
- Передвигаться на одну ячейку влево или вправо.
- Менять свое внутреннее состояние.

Поэтому при составлении программы для каждой пары (символ, состояние) нужно определить три параметра:

- символ  $a_i$  из выбранного алфавита  $A$ ,
- направление перемещения каретки (" $\leftarrow$ " — влево, " $\rightarrow$ " — вправо, "точка" — нет перемещения)
- новое состояние автомата  $q_k$ . Например, команда 1 " $\leftarrow$ "  $q_2$  обозначает "заменить символ на 1, переместить каретку влево на одну ячейку и перейти в состояние  $q_2$ ".

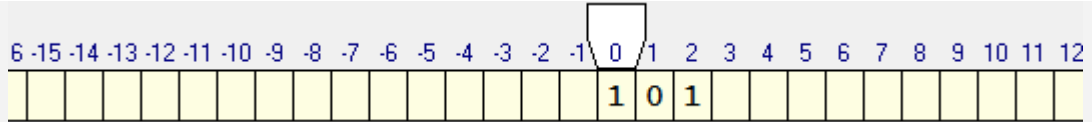
#### 4. Работу Машины Тьюринга.



Машина Тьюринга представляет собой бесконечную ленту, поделенную на ячейки, и каретку (считывающе-печатающее устройство), которая движется вдоль ленты.

Таким образом Машина Тьюринга формально описывается набором двух алфавитов:

- $A = \{a_1, a_2, a_3, \dots, a_n\}$  — внешний алфавит, служит для записи исходных данных
- $Q = \{q_1, q_2, q_3, \dots, q_m\}$  — внутренний алфавит, описывает набор состояний считывающе-печатного устройства.



Машина Тьюринга

Каждая ячейка ленты может содержать символ из внешнего алфавита  $A = \{a_0, a_1, \dots, a_n\}$  (В нашем случае  $A = \{0, 1\}$ )

### 5. Допустимые действия Машины Тьюринга

- записать какой-либо символ внешнего алфавита в ячейку ленты (символ, бывший там до того, затирается)
- сместиться в соседнюю ячейку
- сменить состояние на одно из обозначенных символом внутреннего алфавита  $Q$

Машина Тьюринга — это автомат, который управляется таблицей.

Строки в таблице соответствуют символам выбранного алфавита  $A$ , а столбцы — состояниям автомата  $Q = \{q_0, q_1, \dots, q_m\}$ . В начале работы машина Тьюринга находится в состоянии  $q_1$ . Состояние  $q_0$  — это конечное состояние, попав в него, автомат заканчивает работу.

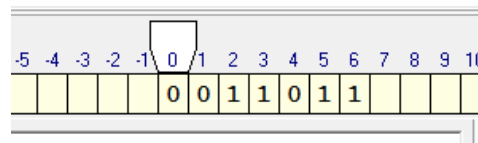
В каждой клетке таблицы, соответствующей некоторому символу  $a_i$  и некоторому состоянию  $q_j$ , находится команда, состоящая из трех частей: символ из алфавита  $A$  направление перемещения: «>» (вправо), «<» (влево) или «.» (на месте) новое состояние автомата

	$Q_1$	$Q_2$	$Q_3$	$Q_4$
0				
1				
_				

В приведенной выше таблице алфавит  $A = \{0, 1, _\}$  (содержит 3 символа), а внутренний алфавит  $Q = \{q_1, q_2, q_3, q_4, q_0\}$ ,  $q_0$  — состояние, заставляющее каретку остановиться.

### Практическая часть.

**Задача 1.** Пусть внешний алфавит состоит из  $A = \{0, 1, _\}$ . На ленте в ячейках находятся символы из алфавита в следующем порядке 0011011, каретка находится над первым символом.



Необходимо составить программу, которая заменит 0 на 1, 1 на 0 и вернет каретку в первоначальное положение.

### Решение

Теперь определимся с состояниями каретки.

Состояние  $Q_1$ . Каретка должна пойти вправо: если видит 0 меняет его на 1 и остается в состоянии  $Q_1$ , если видит 1 — меняет его на 0 и остается в состоянии  $Q_1$ , если видит \_ (пусто)— возвращается назад на 1 ячейку «желает что-то другое», т.е. переходит в состояние  $Q_2$ . Запишем наши рассуждения в таблицу исполнителя. Синтаксис смотрите в справке к программе)

	$Q_1$
0	1 → $Q_1$
1	0 → $Q_1$
_	_ ← $Q_2$

Состояние  $Q_2$ . Теперь опишем «желание каретки»  $Q_2$ . Мы должны вернуться в первоначальное положение. Для этого: если видим 1 оставляем ее и остаемся в состоянии  $Q_2$  (с тем же желанием дойти до конца ряда символов); если видим 0 — оставляем его и продолжаем двигаться влево в состоянии  $Q_2$ ; видим \_ — сдвигается вправо на 1 ячейку.

Вот вы оказались там, где требуется в условии задачи. переходим в состояние  $Q_0$ .

	$Q_1$	$Q_2$
0	1 → $Q_1$	0 ← $Q_2$
1	0 → $Q_1$	1 ← $Q_2$
_	_ ← $Q_2$	_ →

**Задача 2.** Дано: конечная последовательность 0 и 1 (001101011101). Необходимо выписать их после данной последовательности, через пустую ячейку, а в данной последовательности заменить их на 0. Например:

Из 001101011101 получим 000000000000 1111111.

Как видите, семь единиц записались после данной последовательности, а на их местах стоят нолики.

**Решение.** Определим, какие состояния необходимы каретке и сколько.

Для  $Q_1$  : увидел 1 — исправь на нолик и перейди в другое состояние  $Q_2$  (новое состояние вводится, чтобы каретка не поменяла на нули все единицы за один проход)

Для  $Q_2$  : ничего не менять, двигаться к концу последовательности

Для  $Q_3$  : как только каретка увидела пустую ячейку, она делает шаг вправо и рисует единичку, если она видит единичку — то движется дальше, чтобы подписать символ в конце. Как только нарисовал единичку, переходим в состояние  $Q_4$ .

Для  $Q_4$  : проходим по написанным единицам, ничего не меняя. Как только доходим до пустой ячейки, разделяющей последовательность от единиц, переходим с новое состояние  $Q_5$ .

Для  $Q_5$  : в этом состоянии идем начало последовательности, ничего не меняя. Доходим до пустой ячейки, разворачиваемся и переходим в состояние  $Q_1$

Состояние  $Q_0$  каретка примет в том случае, когда она пройдет в состоянии  $q_1$  до конца данной последовательности и встретит пустую ячейку.

Получим такую программу:

	$Q_1$	$Q_2$	$Q_3$	$Q_4$	$Q_5$
0	0 → $Q_1$	0 → $Q_2$			0 ← $Q_5$
1	0 → $Q_2$	1 → $Q_2$	1 → $Q_3$	1 ← $Q_4$	1 ← $Q_5$
_	_ ↓	_ → $Q_3$	1 ↓ $Q_4$	_ ← $Q_5$	_ → $Q_1$

## Решение задач с использованием машины Тьюринга

Задание выполнить на эмуляторе машины Тьюринга Полякова.

### Задание 1

Написать программу на машине Тьюринга, прибавляющую число 2 к введенному числу.

### Задание 2

Написать на машине Тьюринга программу, прибавляющую 3 к введенному числу.

### Задание 3

Перенести первый символ непустого слова  $P$  в его конец. Алфавит:  $A=\{a,b,c\}$ .

### Задание 4

Если первый и последний символы (непустого) слова  $P$  одинаковы, тогда это слово не менять, а иначе заменить его пустым словом. Алфавит:  $A=\{a,b,c\}$ .

### Задание 5

Удалить из слова  $P$  его второй символ, если такой есть. Алфавит:  $A=\{a,b\}$ .

### Задание 6

Удалить из слова  $P$  первое вхождение символа  $a$ , если такое есть. Алфавит:  $A=\{a,b,c\}$ .

### Задание 7

Если  $P$  - непустое слово, то за его первым символом вставить символ  $a$ . Алфавит:  $A=\{a,b,c\}$ .

### Задание 8

Вставить в слово  $P$  символ  $a$  за первым символом  $c$ , если такое есть. Алфавит:  $A=\{a,b,c\}$ .

### Задание 9

Удалить из  $P$  все вхождения символа  $a$ .  $A=\{a,b,c\}$ .

### Задание 10

Удвоить слово  $P$ , поставив между ним и его копией знак  $=$ . Алфавит:  $A=\{a,b\}$ .

### Задание 11

1. Построить таблицу машины Тьюринга, которая заменяет все единицы на нули, а все нули на единицы. Пример. Исходное число 111001. Результат – 000110.

### Задание 12

2. Построить таблицу машины Тьюринга, которая удаляет из числа все нули, например, число 1001110 преобразует к виду 1111. Эта задача уже сложнее и требует ввести в рассмотрение более двух состояний.

### Задание 13

3. Построить машину, имеющую два конечных состояния, условно обозначаемых как YES и NO. Машина должна завершить работу в состоянии YES, если число единиц в записи числа нечетное, и в состоянии NO – в противном случае.

### Задание 14

4. Построить машину, имеющую два конечных состояния, условно обозначаемых как YES и NO. Машина должна завершить работу в состоянии YES, если в записи числа имеется три подряд идущих единицы, и в состоянии NO – в противном случае.

### Задание 15

5. Построить машину Тьюринга, которая получает обратный порядок записи числа, например, исходное число 111001, результат 100111.

### Задание 16

6. Построить машину Тьюринга, которая меняет местами соседние два элемента попарно. Пример. Исходное число 011001 заменяется на 100110.

## Контрольные вопросы

1. Что такое машина Тьюринга
2. Что такое внешний алфавит и алфавит внутренних состояний машины Тьюринга
3. Как записываются команды машины Тьюринга
4. Из какого состояния машины Тьюринга начинает работу
5. Как машина Тьюринга перерабатывает одно слово в другое.

## Практическая работа №5 Конфигурации вычислительной машины

**Тема:** анализ конфигурации вычислительной машины.

**Цели:** закрепить знания по устройству и назначению элементов ПК; приобрести практические навыки анализа конфигурации ПК.

**Студент должен знать:** конфигурацию вычислительной машины.

**уметь:** проводить анализ конфигурации вычислительной машины.

### Теоретическое обоснование

Под конфигурацией вычислительной машины понимают набор аппаратных и программных средств, входящих в ее состав. Минимальный набор аппаратных средств, без которых невозможен запуск, и работа вычислительной машины определяет ее базовую конфигурацию.

Анализ конфигурации вычислительной машины (рассмотрим на примере персонального компьютера) целесообразно проводить в следующей последовательности:

- внешний визуальный осмотр компьютера;
- анализ аппаратной конфигурации компьютера встроенными средствами операционной системы;
- анализ программной конфигурации компьютера;
- анализ конфигурации вычислительной сети, в случае если компьютер к ней подключен.
- В результате внешнего визуального осмотра компьютера определяются следующие данные по его конфигурации:
- тип корпуса системного блока (форм-фактор);
- виды и количество интерфейсов для подключения периферийных устройств, размещенные на задней стенке и лицевой панели системного блока;
- тип клавиатуры и способ ее подключения к компьютеру (количество клавиш, наличие специальных клавиш);
- тип ручного манипулятора (мышь) и способ ее подключения к компьютеру (манипулятор с механической или оптической системой позиционирования, проводной или беспроводной интерфейс подключения);
- тип монитора (ЭЛТ или жидкокристаллический).

Анализ аппаратной конфигурации компьютера, т.е. состава подключенных аппаратных средств, можно проанализировать специальными тестовыми программами, либо встроенными средствами операционной системы, включающей такое понятие как диспетчер устройств.

Для просмотра содержимого диспетчера устройств найдите на рабочем столе ярлык **Компьютер**, далее выделите его и нажмите правую клавишу мыши. В открывшемся контекстном меню выберите пункт **Свойства** (рис. 5.1). В результате этого действия откроется окно **Свойства системы** (рис. 5.2).

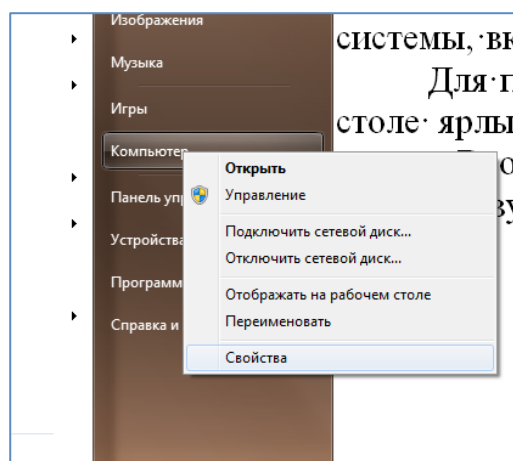


Рисунок 5.1. Свойства

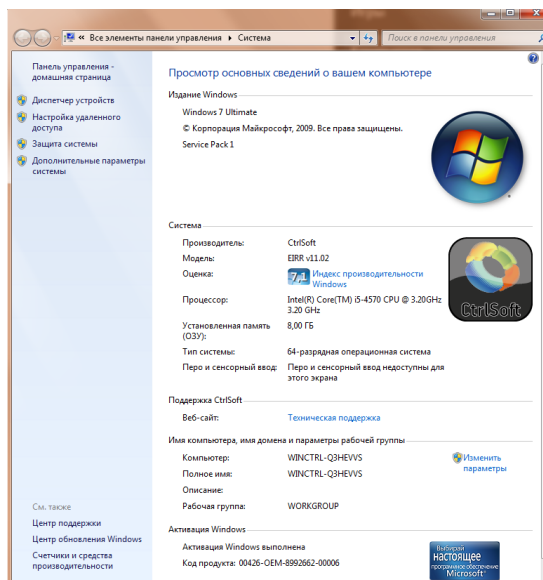


Рисунок 5.2. Свойства системы

В окне *Система* просмотрите и зафиксируйте версию операционной системы, тип процессора и его тактовую частоту, а также объем оперативной памяти (ОЗУ). Далее перейдите к закладке *Диспетчер устройств* (рис. 5.3).

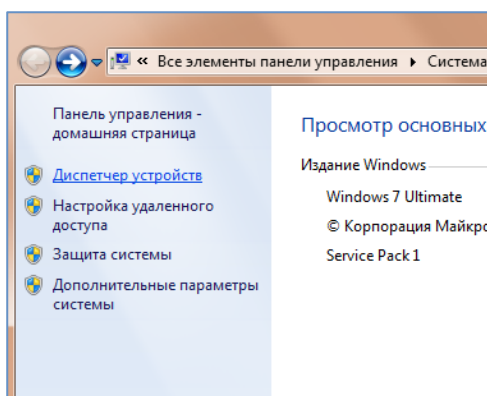


Рисунок 5.3. Диспетчер устройств

В открывшемся окне *диспетчера устройств* (рис. 5.4) представлено графическое отображение перечня оборудования компьютера.

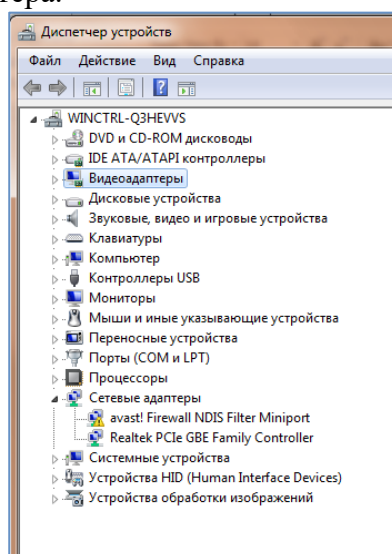


Рисунок 5.4. Перечень оборудования компьютера

Диспетчер устройств можно использовать для обновления драйверов (или программного обеспечения) оборудования, изменения настроек оборудования, а также для устранения неполадок и даже выключения оборудования из конфигурации компьютера.

Для получения доступа к указанным возможностям необходимо выделить из перечня оборудования требуемое устройство и щелкнуть дважды мышью (рис. 5.5). Для просмотра содержимого каждого пункта перечня оборудования необходимо дважды нажать на названии соответствующей группы оборудования.

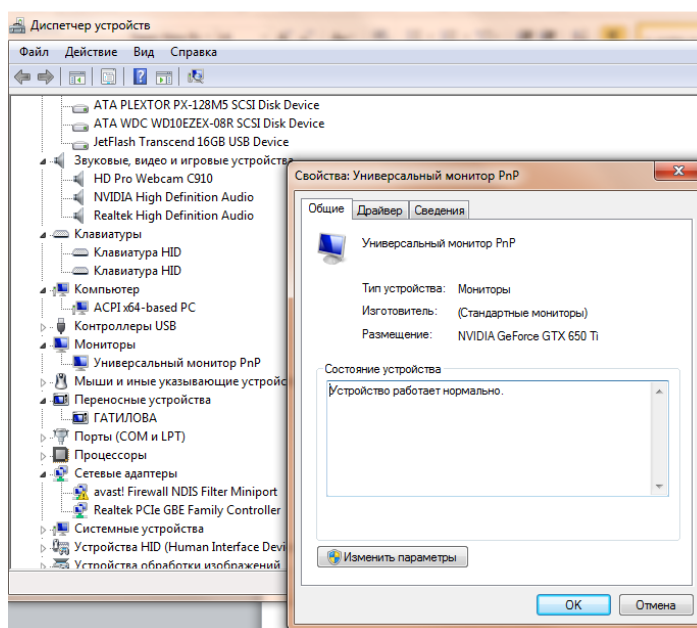


Рисунок 5.5. Свойства конкретного устройства

Диспетчер устройств также позволяет:

- определять правильность работы оборудования компьютера;
- изменять параметры конфигурации оборудования;
- определять драйверы устройств, загружаемые для каждого устройства, и получать сведения о каждом драйвере;
- изменять дополнительные параметры и свойства устройств;
- устанавливать обновленные драйверы устройств;
- отключать, включать и удалять устройства;
- осуществлять возврат к предыдущей версии драйвера;
- распечатывать список устройств, установленных на компьютер.

Современные **Операционные системы** предоставляют пользователю возможность настройки и загрузки различных конфигураций аппаратных средств в рамках одного компьютера. С этой целью введено понятие **Профиль оборудования**.

**Профиль оборудования** - это набор инструкций, используемых Windows для определения устройств, которые должны загружаться при запуске компьютера, или параметров для каждого устройства. При первой установке Windows создается профиль оборудования "Profile 1". По умолчанию все устройства, присутствующие на компьютере на момент установки Windows, включены в "Profile 1".

Вновь создаваемый пользователем профиль оборудования может не включать какое-то из устройств, например, модем или сетевой адаптер, или накопитель гибких магнитных дисков и др.

Если в системе имеется несколько профилей оборудования, можно указать среди них тот, который будет использоваться по умолчанию при каждом запуске компьютера. Windows позволяет также отображать при запуске вопрос, какой профиль следует использовать. После создания профиля оборудования устройства, входящие в него, можно отключать и включать с помощью диспетчера устройств. **При отключении устройства в профиле оборудования драйверы устройства не загружаются при запуске компьютера.**

Более широкие возможности по анализу конфигурации компьютера, в том числе и программной среды, предоставляет модуль **Сведения о системе**.

**Чтобы открыть окно «Сведения о системе», нажмите сочетание «Windows + R», и в открывшемся диалоге «Выполнить» введите «msinfo32» и кликните «Enter».**

В результате этого действия откроется окно *Сведения о системе* (рис. 5.6).

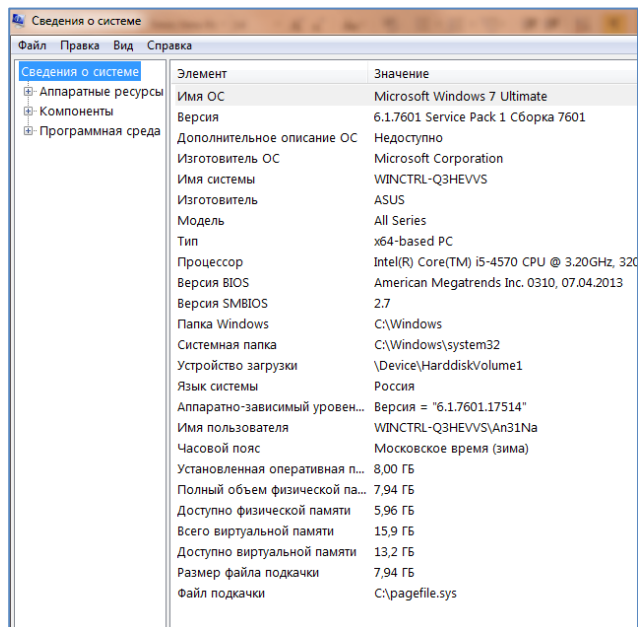


Рисунок 5.6. Сведения о системе

Пример использования модуля *Сведения о системе* иллюстрируется на рис. 5.7, где показаны свойства из подпункта *Дисплей* группы *Компоненты*.

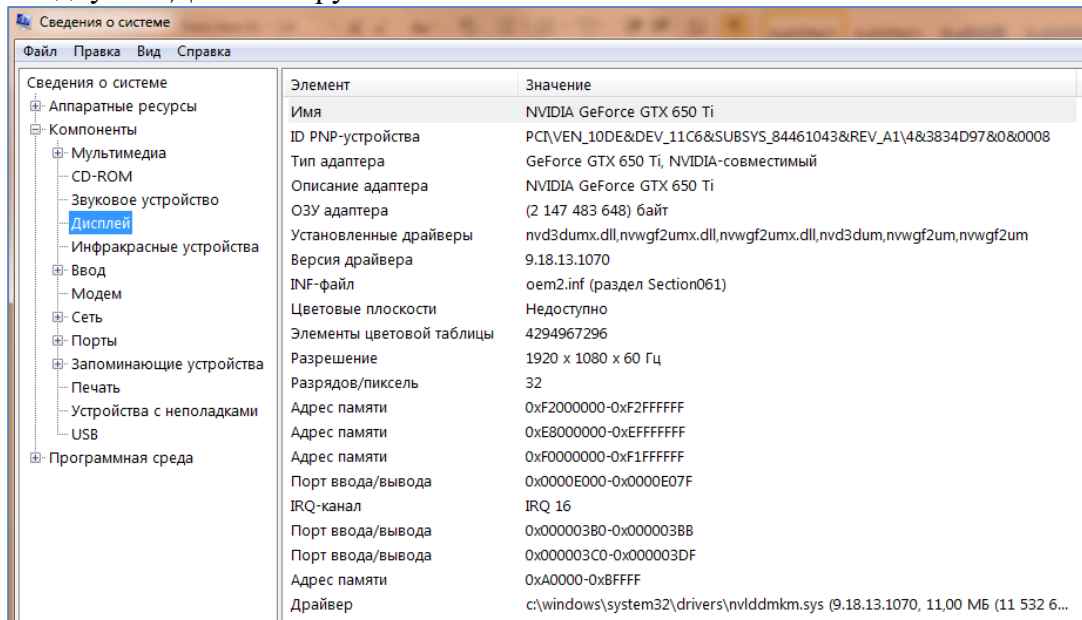


Рисунок 5.7. Компоненты

В данном случае можно получить полную информацию о видеоадаптере, что отображается в правой части открытого окна. Аналогично может быть получена информация о других устройствах, а также о программной среде компьютера. Для этого необходимо выбрать соответствующие пункты в левой части окна *Сведения о системе*.

Для анализа программной среды вычислительной машины помимо модуля *Сведения о системе* можно непосредственно просмотреть полный перечень установленного программного обеспечения, который вызывается последовательным выбором команд *Пуск* и далее *Все программы*.

Для анализа конфигурации вычислительной сети необходимо выбрать на рабочем столе ярлык *Сетевое окружение* или команду *Сетевое окружение* после выбора команды *Пуск*. В открывшемся окне в случае подключения компьютера к локальной сети можно проанализировать конфигурацию сети.

### Практические задания

1. Заполните таблицу (в таблицу следует заносить только реальные данные по конфигурации Вашего компьютера, в случае отсутствия какого-либо устройства ставится прочерк).

п/п	Наименование параметра	Значение параметра
1.	Тип и модель монитора	
2.	Форм-фактор корпуса системного блока	
3.	Клавиатура, интерфейс подключения	
4.	Вид манипулятора "мыши", интерфейс ее подключения	
5.	Интерфейсы подключения периферийных устройств на задней панели системного блока (наименование и количество)	
6.	Интерфейсы подключения периферийных устройств на лицевой панели системного блока (наименование и количество)	
7.	Процессор, модель и тактовая частота	
8.	Объем оперативной памяти	
9.	Тип модема и сетевого интерфейса	
10.	Наименование и скорость привода для чтения оптических дисков	
11.	Модель и объем памяти накопителя на жестких магнитных дисках	
12.	Видеоадаптер, модель и объем видеопам- яти	
13.	Модель звукового адаптера	
14.	Версия операционной системы	
15.	Другие периферийные устройства (прин- тер, сканер и т.д.)	

### Контрольные вопросы

1. Что понимается под конфигурацией вычислительной машины?
2. Какова последовательность анализа конфигурации вычислительной машины?
3. Как проводят анализ аппаратной конфигурации компьютера?
4. Для чего используется Диспетчер устройств?
5. Что понимается под профилем оборудования?
6. Каковы преимущества системы с настраиваемым профилем оборудования?
7. В каком случае используются компонент Сведения о системе?



## Практическая работа №6 Установка различных модулей ПК

**Тема:** идентификация и установка различных модулей ПК: процессора, ОЗУ, видеопамяти, внешней памяти.

**Цели:** получить навыки определения основных характеристик ЦП, тестирования параметров ЦП, определения основных характеристик видеокарты, мониторинга параметров видеокарты.

**Студент должен знать:** характеристики ЦП;

**уметь:** определять основные характеристики ЦП, видеокарты.

### Теоретическое обоснование

ЦП – это центральный блок персонального компьютера, предназначенный для управления работой всех остальных блоков и выполнения арифметических и логических операций над информацией.

#### Основные характеристики процессора:

- количество вычислительных ядер;
- количество потоков;
- размер кеша;
- частота процессора;
- скорость шины процессора;
- техпроцесс;
- энергопотребление.

**CPU-Z** - утилита, которая предоставит вам самую подробную информацию об установленном в системе процессоре, памяти, кэше и материнской плате.

Программа обладает небольшим размером, удобным выводом сведений о компонентах и поддерживает практически все типы процессоров и материнских плат.

Выводит следующую информацию:

#### *Процессор*

- Название процессора.
- Степпинг ядра и техпроцесс.
- Корпусировка.
- Напряжение ядра.
- Внутренняя и внешняя частоты, множитель процессора.
- Поддерживаемые наборы инструкций.
- Информация о кэше.

#### *Материнская плата*

- Производитель, модель и ревизия.
- Производитель, дата и модель BIOS.
- Чипсет (северный и южный мост) и датчики.
- Графический интерфейс.

#### *Память*

- Частота и тайминги.
- Количество каналов памяти.
- Спецификации модулей, записанные в SPD (Схема Последовательного Детектирования ): производитель, серийный номер, таблица таймингов.

#### *Видеокарта*

- Название видеокарты
- Имя GPU
- Техпроцесс
- Поддержка DirectX
- Поддержка пиксельного шейдера

- Тип памяти
- Количество памяти
- Пропускная способность памяти
- Тип и ширина шины
- Частота памяти, GPU (стандартная/разогнанная)
- Версия драйвера и BIOS'a

#### *Датчики*

- Частота ядра GPU
- Частота памяти GPU
- Температура GPU
- Скорость кулера
- Загрузки GPU в реальном времени

**Open Hardware Monitor** - бесплатное программное обеспечение с открытым исходным кодом для мониторинга температуры ядер процессора, скорости вращения вентиляторов, напряжения, нагрузки и тактовой частоты ЦП.

Утилита поддерживает большинство микросхем аппаратного мониторинга, может работать с процессорами Intel и AMD, видеокартами ATI и Nvidia, умеет отображать параметры жесткого диска SMART. Информацию можно просматривать в главном окне приложения, в настраиваемом виджете рабочего стола или в системном tree Windows. Программа не требует инсталляции.

**CPU Stress Test (CST)** - небольшая портативная утилита, которая предназначена для проверки процессора в экстремальных условиях, путем нагрузки его SSE командами (используется алгоритм трансформации матриц). С помощью программы можно определить стабильность ЦП, отсутствие ошибок в работе канала процессор-память, провести тестирование эффективности системы охлаждения и подсистемы питания процессора, а также опытным путем найти предел разгона. Программа включает в себя три теста, которые выполняются последовательно. Для каждого теста можно выбирать свой размер матрицы, количество вычислений и количество повторений теста на один проход. Все настройки осуществляются через файл cst.ini.

**FurMark** - полезная утилита, позволяющая проводить стресс-тестирование видеокарт совместимых с API OpenGL.

В настройках можно задавать разрешение экрана, активировать оконный или полноэкранный режим и настраивать сглаживание, а также указывать время проведения теста. Для оверклокеров будет полезен режим испытания на принудительный отказ (burn-in test), с помощью которого можно проверить стабильность графической платы и возможность системы охлаждения (например при разгоне). Результат теста можно опубликовать на официальном сайте разработчика для занесения в базу полученных результатов.

Основные возможности программы:

- программа создает высокую нагрузку на видеокарту компьютера, при этом постоянно проверяя и отображая на мониторе температуру графического ядра, количество кадров в секунду и некоторые другие параметры;
- настройки тестирования можно изменять (разрешение экрана, сглаживание и др.);
- предусмотрен оконный или полноэкранный режимы тестирования;
- есть возможность испытания видеокарты на принудительный отказ (режим burn-in test), что позволяет проверить не только стабильность графической платы в условиях максимальной нагрузки, но и эффективность ее системы охлаждения;
- результаты тестирования можно добавить в базу на официальном сайте разработчика программы.

Особенности:

в состав программы FurMark входят утилиты GPU-Z и GPU Shark, предоставляющие подробную информацию о видеокарте компьютера

## Практические задания

### Задание 1. Определение основных характеристик центрального процессора с помощью свойств компьютера.

Зайдите в свойства компьютера.

В открывшемся окне найдите информацию о процессоре.

Приведите в отчёте данные о процессоре Вашего рабочего компьютера.

Заполните таблицу 6.1.

Таблица 6.1 – Характеристики ЦП

Наименование характеристики ЦП	Рабочий ПК
Фирма-производитель процессора:	
Семейство процессоров:	
Модель процессора:	
Частота работы процессора:	

### Задание 2. Определение основных характеристик центрального процессора с помощью программы CPU-Z.

С помощью программы CPU-Z просмотрите подробные характеристики ЦП рабочего ПК.

Перепишите основные данные в тетрадь.

### Задание 3. Тестирование температурных показателей ЦП.

Первое, что нужно сделать после установки процессора или сборки нового компьютера, это проверить температуру процессора. Если перебрали с термопастой или наоборот намазали ее слишком мало, то процессор будет перегреваться. Также возможен перегрев из-за неправильной установки системы охлаждения.

Протестируйте температуру ЦП с помощью утилиты Open Hardware Monitor или HWMonitor .

Перепишите основные данные в тетрадь.

### Задание 4. Стресс-тестирование ЦП.

Стресс тестирование – это тестирование с максимальной нагрузкой. Такой тип тестирования используют для того чтобы убедиться, что система охлаждения справится при высокой нагрузке на процессор.

Проведите стресс тестирование ЦП с помощью утилиты CPU Stress Test (CST) или S&M.

Перепишите основные данные в тетрадь.

### Задание 5. Изучение возможностей Furmark.

Установите программу Furmark.

После запуска в окне основных настроек можно отдельно запустить утилиты GPU-Z и GPU Shark, которые позволяют получить дополнительную информацию о работе видеокарты. Также в выпадающем меню может настраиваться нужное разрешение, сглаживание и функция полного экрана (рисунок 6.1). Изучите настройки.

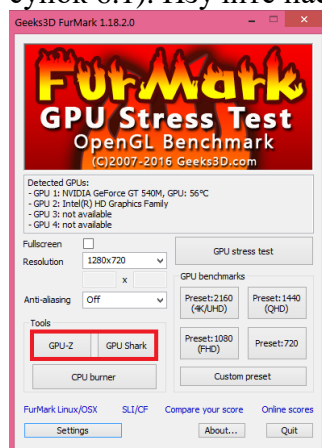


Рисунок 6.1 – Интерфейс программы Furmark

Нажмите на кнопку «**Setting**», переводит на основные настройки теста, где рекомендуется установить «**GPU temperature alarm**» и ввести предельно допустимое значение температуры (не более 80 °C). Это нужно сделать для того, чтобы исключить возможность повреждения процессора видеокарты в процессе перегрева. Ниже можно выбрать продолжительность проводимого теста в миллисекундах, а при желании можно активировать ведение лога мощности, температуры и других характеристик (рисунок 6.2).

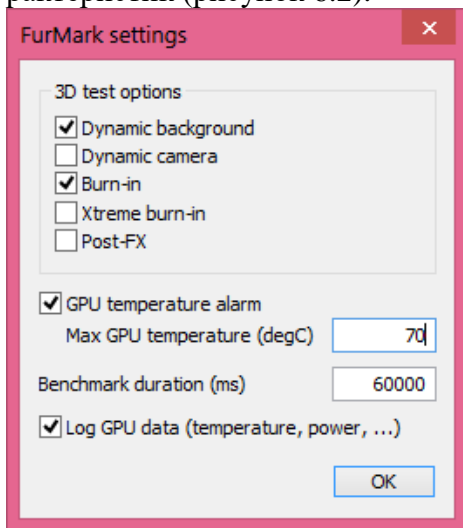


Рисунок 6.2 – Задание активации характеристик

Перед проведением теста проверьте текущую температуру видеокарты при помощи закладки «**Sensors**» в окне «**GPU-Z**» (рисунок 6.3). Не рекомендуется запускать процедуру тестирования в том случае, если на данный момент температура процессора видеокарты составляет 70-90 °C.

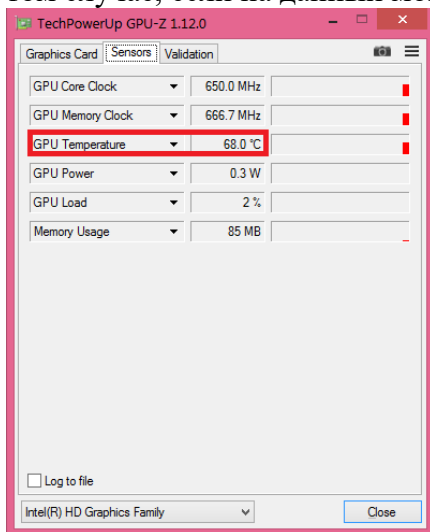


Рисунок 6.3 – Определение текущей температуры видеокарты

Перейдите на вкладку Graphics Card изучите характеристики (рисунок 6.4), перепишите основные характеристики.

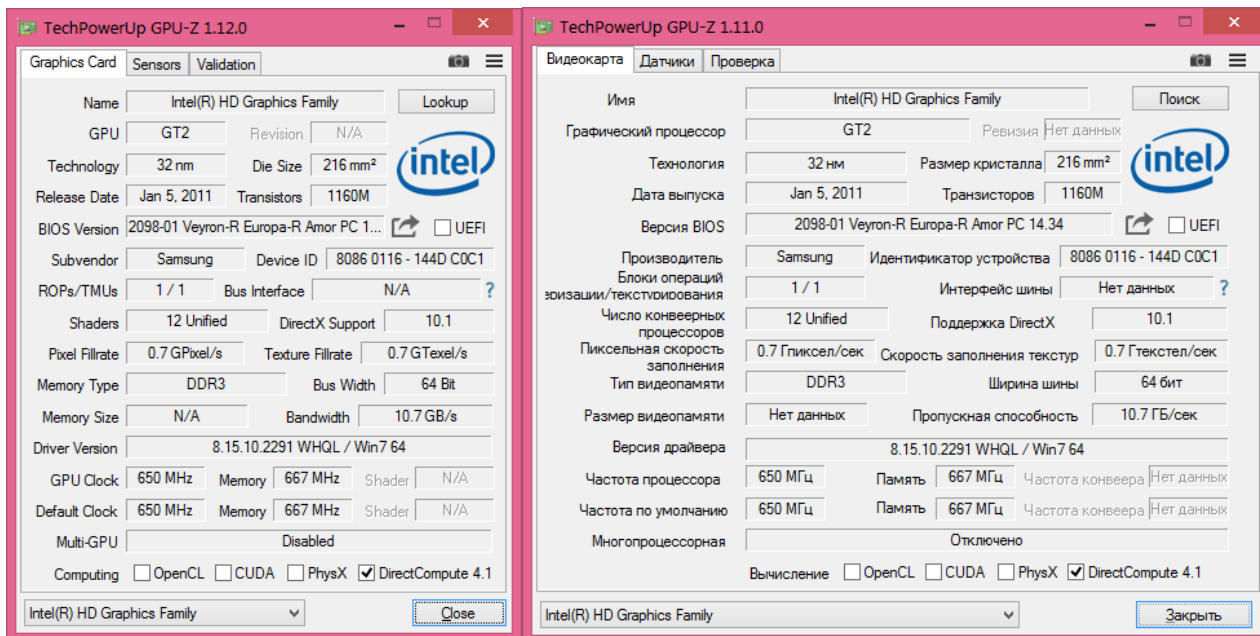


Рисунок 6.4 – Характеристики видеокарты

Изучите программу GPU Shark, отображающую в реальном времени основные параметры видеокарты, такие как минимальное и максимальное значение текущей температуры, напряжение питания, частота и процент использования графического процессора и видеопамяти, запишите основные характеристики.

### Задание 7. Проведение теста в Furmark.

Для того чтобы запустить тестирование, необходимо выбрать одну из пяти кнопок. Для активации проверки с введенными вами настройками кликайте на «**Custom preset**». После запуска тестирования появится сообщение о том, какие риски могут присутствовать в процессе разгона данного оборудования, и для продолжения процедуры кликните на кнопку «**GO!**». В самом окне теста отображается основная информация о режиме, частоте и загрузке графического процессора, а в нижней части находится график роста температуры в зависимости от времени (рисунок 6.5).

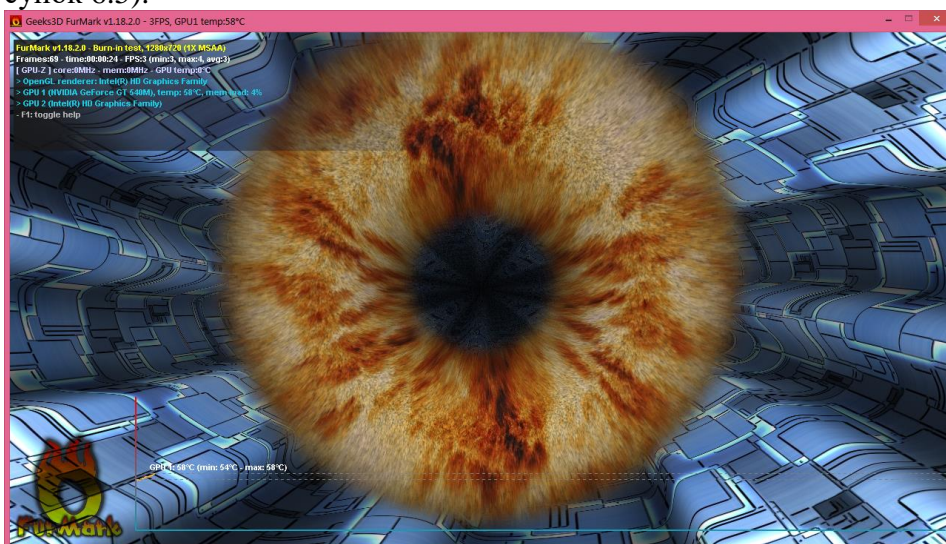


Рисунок 6.5 – Окно тестирования

Даже при установленной кнопке «GPU temperature alarm» программа не будет останавливаться при достижении определенного значения, а только даст предупреждение о том, что такое значение было достигнуто. По этой причине крайне не рекомендуется отходить от компьютера до конца процедуры тестирования, и при значениях критической температуры завершите тестирование.

Изучите окно с основными результатами проверки, такие как предельная температура и основные сведения о системе. При отсутствии каких-либо дефектов изображения, резких скачков температуры, а также соответствии характеристик с теми, которые были указаны производителем, можно сказать о том, что тестирование было проведено успешно. Сделайте выводы.

### **Контрольные вопросы**

- 1 Дайте определение понятию «Центральный процессор».
- 2 Перечислите функции ЦП.
- 3 Какими способами можно определить характеристики ЦП?
- 4 Перечислите и опишите основные технические характеристики ЦП.
- 5 Перечислите основные неисправности ЦП и их причины.
- 6 Дайте определение понятию «Видеокарта».
- 7 Перечислите функции видеокарты.
- 8 Опишите структуру видеокарты.
- 9 Перечислите и опишите основные технические характеристики видеокарты.

## Практическая работа № 7 Написание программ для эмулятора ЭВМ «ЛамПанель»

Тема: **Написание программ для эмулятора ЭВМ «ЛамПанель»**

Цель:

- для изучения принципов работы компьютера (процессор, ОЗУ, ПЗУ);
- для начального изучения программирования на языке ассемблера;
- для изучения операций с целыми числами, в том числе поразрядных логических операций и сдвигов.

Программа-тренажёр «ЛамПанель» – это учебная модель компьютера, который управляет панелью лампочек.

Модель компьютера включает:

- процессор,
- оперативную память (ОЗУ),
- постоянную память (ПЗУ)
- устройство вывода – панель лампочек размером 8×16.

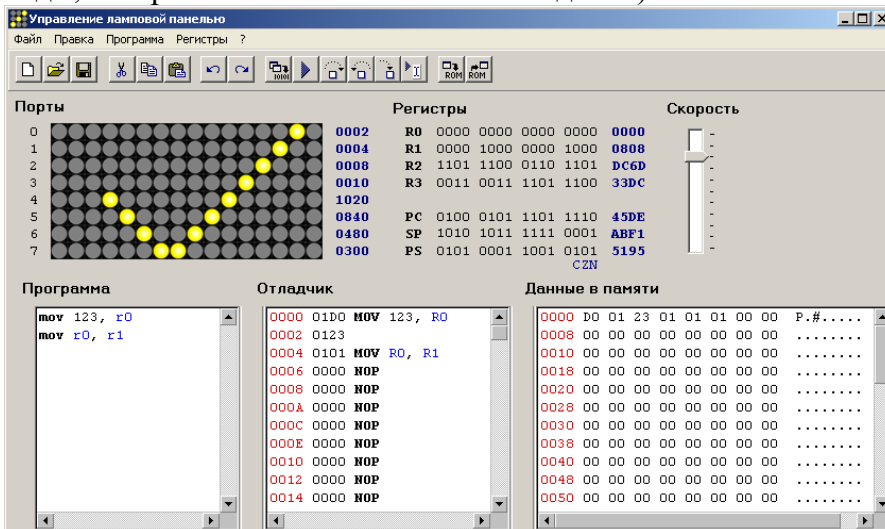
Справочник по языку ассемблера «ЛамПанель»

Система команд процессора в тренажёре «ЛамПанель» – 16-ти -разрядная.

Программа для процессора составляется на **языке ассемблера**, в котором каждая символьная команда соответствует одной (числовой) команде процессора.

Программа набирается в окне «Программа». Программу можно сохранять в виде файла на диске, а потом загружать в память из файла (с помощью меню «Файл»).

При запуске (по клавише F9) набранная программа транслируется (переводится в машинные коды, которые появляются в окне «Отладчик») и начинается ее выполнение.



В окне «Данные в памяти» показывается содержимое оперативной памяти, в которой расположены программа и данные (принцип однородности).

**Структура процессора тренажёра «ЛамПанель»**

Процессор имеет 4 регистра общего назначения (РОН), которые называются **R0, R1, R2, R3**. Кроме того, есть еще три внутренних регистра, недоступные программисту (но видимые в окне программы):

- **PC** (англ. *program counter*) – программный счетчик, счётчик команд, указывающий на следующую выполняемую команду;
- **SP** (англ. *stack pointer*) – указатель стека, адрес вершины стека; стек размещается в нижней части оперативной памяти, его содержимое можно просмотреть в нижней части окна «Данные в памяти»;
- **PS** (англ. *processor state*) – регистр состояния процессора; используются только три младших бита: 0 – бит **N** (англ. *negative*, отрицательный результат), 1 – бит **Z** (англ. *zero*, нулевой результат) и 2 – бит **C** (англ. *carry*, перенос).



В качестве устройства вывода используется **панель лампочек** размером 8×16. Каждый ряд лампочек управляется через отдельный порт вывода.

Всего используется **восемь** 16-разрядных портов с именами **P0, P1, P2, P3, P4, P5, P6** и **P7**.

Постоянное запоминающее устройство (ПЗУ) предназначено для хранения системных подпрограмм. Код ПЗУ загружается при старте программы из текстового файла, поэтому пользователь может изменять содержимое ПЗУ: добавлять, удалять и изменять любые процедуры.

#### **Основные правило написания программ:**

- Программа должна заканчиваться командой **stop**. Например, самая простая программа: **stop**
- Команда **NOP** (*no operation*, нет операции) – это пустая команда, она ничего не делает.
- Комментарий начинается символом «точка с запятой»: **nop ; пустая команда, stop**

#### *Инструменты отладки программы*

Программу можно выполнить всю целиком (клавиша F9) или в пошаговом режиме (F8). В пошаговом режиме в окне отладчика зелёным цветом выделяется текущая строка, которая будет выполнена при следующем нажатии F8. Сочетание клавиш Ctrl+F8 позволяет отменить только что сделанную команду.

Клавиша F7 (вместо F8) позволяет войти в подпрограмму и выполнить ее пошагово (см. раздел «Подпрограммы» ниже).

Если установить курсор в какую-то строчку программы и нажать клавишу F4, программа будет выполняться до этой строчки и затем остановится.

Движок «Скорость» изменяет скорость выполнения программы.

Все команды отладки включены в меню «Программа».

Кроме того, они могут выполняться с помощью кнопок панели инструментов:

	Трансляция в машинные коды без выполнения (Ctrl+F9).
	Трансляция и выполнение (F9).
	Выполнить один шаг (F8).
	Отменить один шаг (Ctrl+F8).
	Войти в подпрограмму (F7).
	Выполнить до курсора (F4).

С помощью меню «Регистры» можно изменить значения любого регистра во время выполнения программы в пошаговом режиме.

#### *Работа с регистрами и портами*

Для простейшей обработки данных можно использовать 4 регистра процессора и 8 портов ламповой панели.

#### **Основные операции:**

- записать данные в регистр, например,

<i>ассемблер</i>	<i>псевдокод</i>
<b>mov 1234, R0</b>	<b>R0:=1234<sub>16</sub></b>

Все числа записываются в шестнадцатеричной системе счисления.

- скопировать значение из одного регистра в другой, например,

<i>ассемблер</i>	<i>псевдокод</i>
<b>mov R0, R1</b>	<b>R1:=R0</b>

- вывести значение регистра в порт

<i>ассемблер</i>	<i>псевдокод</i>
<b>out R0, P1</b>	<b>P1:=R0</b>

- прочитать значение из порта в регистр

<i>ассемблер</i>	<i>псевдокод</i>
<b>in P2, R0</b>	<b>R0:=P2</b>

#### **Пример программы:**

<i>ассемблер</i>	<i>псевдокод</i>
<b>mov 1234, R0</b>	<b>R0:= 1234<sub>16</sub></b>
<b>mov R0, R2</b>	<b>R2:= R0</b>



<b>out R2, P1</b> <b>stop</b>	<b>P1:= R2</b> <b>стоп</b>
----------------------------------	-------------------------------

### Арифметические операции

Арифметические операции могут выполняться с числами (константами) и значениями регистров. Результат записывается по адресу второго операнда-регистра (это не может быть число).

- сложение

<i>ассемблер</i>	<i>псевдокод</i>
<b>add 1, R1</b> <b>add R2, R3</b>	<b>R1:= R1 + 1</b> <b>R3:= R3 + R2</b>

- вычитание

<i>ассемблер</i>	<i>псевдокод</i>
<b>sub 2, R1</b> <b>sub R2, R3</b>	<b>R1:= R1 - 2</b> <b>R3:= R3 - R2</b>

- умножение

<i>ассемблер</i>	<i>псевдокод</i>
<b>mul 5, R1</b> <b>mul R2, R3</b>	<b>R1:= R1 * 5</b> <b>R3:= R3 * R2</b>

- деление

<i>ассемблер</i>	<i>псевдокод</i>
<b>div 12, R1</b> <b>div R2, R3</b>	<b>R1:= R1 div 12</b> <b>R3:= R3 div R2</b>

### Логические операции

Логические операции могут выполняться с числами (константами) и значениями регистров. Результат записывается по адресу второго операнда-регистра (это не может быть число).

- отрицание («НЕ»)

<i>ассемблер</i>	<i>псевдокод</i>
<b>not R1</b>	<b>R1:= not R1</b>

- логическое умножение («И»)

<i>ассемблер</i>	<i>псевдокод</i>
<b>and R0, R1</b> <b>and 1234, R1</b>	<b>R1:= R1 and R0</b> <b>R1:= R1 and 1234<sub>16</sub></b>

- логическое сложение («ИЛИ»)

<i>ассемблер</i>	<i>псевдокод</i>
<b>or R0, R1</b> <b>or 1234, R1</b>	<b>R1:= R1 or R0</b> <b>R1:= R1 or 1234<sub>16</sub></b>

- сложение по модулю 2 («исключающее ИЛИ»)

<i>ассемблер</i>	<i>псевдокод</i>
<b>xor R0, R1</b> <b>xor 1234, R1</b>	<b>R1:= R1 xor R0</b> <b>R1:= R1 xor 1234<sub>16</sub></b>

### Операции сдвига

В командах сдвига первый операнд – это величина сдвига (от 1 до 10<sub>16</sub>), а второй – регистр.

- логический сдвиг влево и вправо

<i>ассемблер</i>	<i>псевдокод</i>
<b>shl 2, R1</b> <b>shr F, R1</b>	<b>R1:= R1 shl 2<sub>16</sub></b> <b>R1:= R1 shr F<sub>16</sub></b>

- арифметический сдвиг вправо

<i>ассемблер</i>	<i>псевдокод</i>
<b>sar 2, R1</b>	<b>R1:= R1 sar 2<sub>16</sub></b>

- циклический сдвиг влево и вправо

<i>ассемблер</i>	<i>псевдокод</i>
<b>rol 2, R1</b> <b>ror F, R1</b>	<b>R1:= R1 rol 2<sub>16</sub></b> <b>R1:= R1 ror F<sub>16</sub></b>

- циклический сдвиг влево и вправо через бит переноса

ассемблер	псевдокод
<b>rcl 2, R1</b>	<b>R1:= R1 rcl 2<sub>16</sub></b>
<b>rcr F, R1</b>	<b>R1:= R1 rcr F<sub>16</sub></b>

Операции: метки, сравнения и переходы

Команды перехода используются для выполнения разветвляющихся алгоритмов. Различают безусловный переход (переходить всегда) и условные переходы (переход при выполнении какого-то условия).

Чтобы обозначить место перехода, необходимо создать метку. **Метка** – это произвольное имя, за которым следует двоеточие. **После двоеточия не должно быть никаких символов** (метка – это отдельная строка программы).

Безусловный переход имеет формат

**jmp метка**

Пример программы (бесконечный цикл):

```
qq:
por
jmp qq
```

Условные переходы зависят от битов состояния процессора, которые определяются результатом последней операции:

```
jge метка ; если больше или равно
jl метка ; если меньше
jnz метка ; если не ноль
jz метка ; если ноль
jle метка ; если меньше или равно
jg метка ; если больше
```

Пример программы (цикл из 5 шагов):

ассемблер	псевдокод
<b>mov 5, R1</b>	<b>R1:=5</b>
<b>qq:</b>	<b>нц пока R1&lt;&gt; 0</b>
<b>sub 1, R1</b>	<b>R1:= R1 - 1</b>
<b>jnz qq</b>	<b>кц</b>

Существует команда сравнения, которая изменяет только биты состояния процессора:

ассемблер	значение
<b>cmp 2, R1</b>	установка битов состояния по значению <b>2<sub>16</sub>-R1</b>
<b>cmp R0, R1</b>	установка битов состояния по значению <b>R0-R1</b>

Пример программы:

ассемблер	псевдокод	
<b>cmp 5, R0</b>	<b>если R0=5 то</b>	
<b>jnz aaa</b>		<b>R0:= R0 + 1</b>
<b>add 1, R0</b>		<b>все</b>
<b>aaa:</b>		

Подпрограммы

Подпрограммы – это вспомогательные алгоритмы, которые можно вызывать по имени. В языке ассемблера имя подпрограммы – это метка. Для вызова подпрограммы используется команда

**call метка**

Подпрограмма должна заканчиваться командой возврата из подпрограммы

**ret**

Подпрограммы располагаются в программе ниже основной программы, после команды **stop**.

Пример программы, которая использует подпрограмму **divMod** для деления с остатком:

ассемблер	псевдокод
<b>mov 1234, R0</b>	<b>R0:= 1234<sub>16</sub></b>
<b>mov 10, R1</b>	<b>R1:= 10<sub>16</sub></b>
<b>call divMod</b>	<b>вызвать divMod</b>
<b>stop</b>	<b>стоп</b>
<b>divMod:</b>	

<b>mov R0, R2</b>	<b>R2:= R0</b>
<b>div R1, R0</b>	<b>R0:= R0 div R1</b>
<b>mul R0, R1</b>	<b>R1:= R1 * R0</b>
<b>sub R1, R2</b>	<b>R2:= R2 – R1</b>
<b>mov R2, R1</b>	<b>R1:= R2</b>
<b>ret</b>	<b>возврат</b>

Чтобы при отладке выполнять по шагам не только основную программу, но и подпрограмму, при выполнении команды **call** нужно нажать не F8, а F7.

#### Работа со стеком

Стек – это структура типа LIFO (англ. *Last In – First Out*, последним пришел – первым ушел). В современных компьютерах стек размещается в памяти, специальный регистр SP (англ. *stack pointer*) указывает на начало стека. Для работы со стеком используются всего две команды:

ассемблер	псевдокод
<b>push R0</b>	<b>сохранить R0 в стеке</b>
<b>pop R0</b>	<b>«снять» данные с вершины стека в R0</b>

Конечно, сохранять в стеке можно не только **R0**, но и другие регистры.

Стек используется:

- для временного хранения данных
- для хранения адресов возврата из подпрограмм
- для размещения локальных переменных подпрограмм

Пример программы (обмен значений регистров R0 и R1):

ассемблер	псевдокод
<b>push R0</b>	<b>R0 – в стек</b>
<b>push R1</b>	<b>R1 – в стек</b>
<b>pop R0</b>	<b>со стека – в R0 (старое значение R1)</b>
<b>pop R1</b>	<b>со стека – в R1 (старое значение R0)</b>

Если подпрограмма использует какой-то регистр, которые не содержит исходные данные и не предназначен для записи результата, она должна сохранить его стеке при входе и восстановить старое значение из стека при выходе. Например:

ассемблер	псевдокод
<b>proc:</b>	<b>начало подпрограммы</b>
<b>push R0</b>	<b>R0 – в стек</b>
<b>...</b>	<b>основное тело подпрограммы</b>
<b>pop R0</b>	<b>со стека – в R0 (старое значение R0)</b>
<b>ret</b>	<b>возврат из подпрограммы</b>

Заметьте, что подпрограмма, приведенная в предыдущем пункте, не совсем грамотно написана – она не сохраняет значение регистра R2, хотя «портит» его во время работы.

#### Вызов подпрограмм из ПЗУ


ПЗУ в данной модели компьютера – это набор подпрограмм, каждая из которых заканчивается командой **ret**. Всего в ПЗУ может быть до 256 подпрограмм.

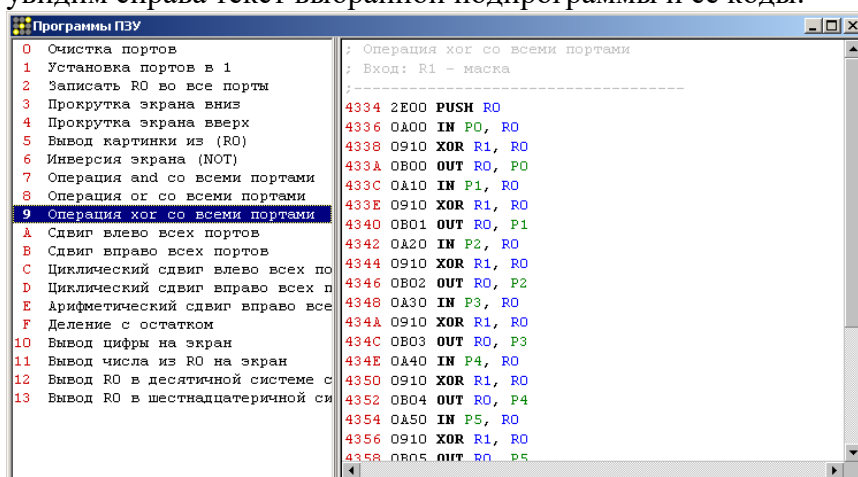
ПЗУ загружается при запуске тренажёра «ЛамПанель» из файла **lampanel.rom**, который должен находиться в том же каталоге, что и сама программа. Это обычный текстовый файл, который можно редактировать в редакторах типа *Блокнота* (если, конечно, вы понимаете, что вы делаете). В настоящей версии в ПЗУ включены следующие подпрограммы:

Номер	Описание
0	Очистить все порты панели (выключить все лампочки).
1	Установить в FF <sub>16</sub> все порты панели (включить все лампочки).
2	Записать значение R0 во все порты панели.
3	Прокрутить изображение на панели вниз.
4	Прокрутить изображение на панели вверх.
5	Вывести на панель массив данных, адрес которого находится в R0.
6	Выполнить инверсию экрана (применить NOT).
7	Операция «И» со всеми портами (R1 – маска).
8	Операция «ИЛИ» со всеми портами (R1 – маска).

9	Операция «исключающее ИЛИ» со всеми портами (R1 – маска).
A <sub>16</sub>	Логический сдвиг влево всех портов (R1 – величина сдвига).
B <sub>16</sub>	Логический сдвиг вправо всех портов (R1 – величина сдвига).
C <sub>16</sub>	Циклический сдвиг влево всех портов (R1 – величина сдвига).
D <sub>16</sub>	Циклический сдвиг вправо всех портов (R1 – величина сдвига).
E <sub>16</sub>	Арифметический сдвиг вправо всех портов (R1 – величина сдвига).
F <sub>16</sub>	Деление с остатком (R0:=R0 div R1, R1:=R0 mod R1).
10 <sub>16</sub>	Вывод цифры на экран (R0 – цифра, R1 – позиция, от 0 до 2)
11 <sub>16</sub>	Вывод числа из R0 на экран (R1 – система счисления, от 2 до 16).
12 <sub>16</sub>	Вывод числа из R0 на экран в десятичной системе счисления.
13 <sub>16</sub>	Вывод числа из R0 на экран в шестнадцатеричной системе счисления.

Просмотреть содержимое ПЗУ можно с помощью пункта меню «Программа-Просмотр ПЗУ»

или кнопки  на панели инструментов. Выделив какую-нибудь строчку в левой части окна, мы увидим справа текст выбранной подпрограммы и ее коды:



Для вызова подпрограмм из ПЗУ нужно использовать команду:

**system номер\_подпрограммы**

Пример программы:

ассемблер	псевдокод
system 0	выключить панель
system 1	включить все лампочки
mov 123, R0	R0:= 123 <sub>16</sub>
system 12	вывести R0 в десятичной системе
system 6	инверсия
mov 1, R1	R1:= 1 ; величина сдвига
system A	сдвиг экрана влево
system B	сдвиг экрана вправо
system 13	вывести R0 в шестнадцатеричной системе
stop	стоп

*Байтовые команды*

Все рассмотренные выше команды работают с 16-битными данными (словами). Часто, например, при обработке текстов, нужно использовать однобайтные данные. Для этого предназначены следующие команды, которые полностью аналогичны соответствующим командам без буквы «b» (от англ. *byte*) на конце:

команда	значение
movb	копирование байта
cmpb	сравнение двух байтов
shlb	логический сдвиг влево
shrb	логический сдвиг вправо
sarlb	арифметический сдвиг вправо
rolb	циклический сдвиг влево
rorb	циклический сдвиг вправо

<b>rclb</b>	циклический сдвиг влево через бит переноса
<b>rcrb</b>	циклический сдвиг вправо через бит переноса

Команда **movb** очищает старший байт регистра, в который копируются данные.

Например,

ассемблер	псевдокод
<b>mov 1234, R0</b>	<b>R0:= 1234<sub>16</sub></b>
<b>movb 12, R0</b>	<b>R0:= 12<sub>16</sub></b>

Остальные команды никак не изменяют старший байт регистра-приемника.

Существует специальная команда для обмена старшего и младшего байтов слова:

**swapb regистр**

Пример программы:

ассемблер	псевдокод
<b>mov 1234, R0</b>	<b>R0:= 1234<sub>16</sub></b>
<b>swapb R0</b>	<b>R0:= 3412<sub>16</sub></b>

Работа с данными

Согласно принципу однородности памяти фон Неймана, данные размещаются в той же области памяти, что и программа (обычно сразу после команды **stop**).

В тренажере «ЛамПанель» данные – это 16-битные слова (вводятся как числа в шестнадцатеричной системе счисления) или символьные строки, заключенные в двойные кавычки. Для размещения данных в памяти применяется команда **data**.

Например:

```
... ; основная программа
stop
ddd: ; метка начала блока данных
data 1234 ; слово 123416
data 5678 ; слово 567816
data "Ехал Грека через реку" ; строка
```

Для того, чтобы работать с этими данными, нужно как-то к ним обратиться. Для этого используется косвенная адресация – в регистре находятся не сами данные, а их адрес в памяти.

Рассмотрим пример:

ассемблер	псевдокод
<b>mov @ddd, R0</b>	<b>R0:= адрес метки ddd</b>
<b>swapb (R0)</b>	<b>переставить байты слова под адресу ddd</b>
<b>add 2, R0</b>	<b>увеличить адрес на 2 (байта)</b>
<b>swapb (R0)</b>	<b>переставить байты слова под адресу ddd+2</b>
<b>stop</b>	<b>стоп</b>
<b>ddd:</b>	<b>начало блока данных</b>
<b>data 1234</b>	<b>здесь будет 3412<sub>16</sub></b>
<b>data 5678</b>	<b>здесь будет 7856<sub>16</sub></b>

Запись **@метка** означает «адрес метки». Запись **(R0)** означает «данные, адрес которых находится в R0» – это и есть косвенная адресация.

Косвенную адресацию можно использовать и в других командах, работающих с регистрами.

Обработка массивов

Пусть в блоке данных, который начинается на метке **ddd**, записан массив, который нужно обработать в цикле. В этом случае удобно использовать косвенную адресацию с автоматическим увеличением адреса. Запись «**(R0)+**» означает «работать с данными, адрес которых находится в R0, и после выполнения операции увеличить R0». Если команда работает со словом, R0 увеличится на 2, а если с байтом – на 1.

Пример программы:

ассемблер	псевдокод
<b>mov @ddd, R0</b>	<b>R0:= адрес метки ddd</b>
<b>mov 3, R1</b>	<b>записать в R1 количество шагов цикла</b>
<b>loop:</b>	<b>начало цикла</b>
<b>swapb (R0)+</b>	<b>переставить байты слова под адресу из R0</b>
<b>sub 1, R1</b>	<b>уменьшить счетчик оставшихся шагов</b>

<b>jnz loop</b>	если счетчик не ноль, перейти в начало цикла
<b>stop</b>	стоп
<b>ddd:</b>	начало блока данных
<b>data 1234</b>	здесь будет 3412 <sub>16</sub>
<b>data 5678</b>	здесь будет 7856 <sub>16</sub>
<b>data 9ABC</b>	здесь будет BC9A <sub>16</sub>

Пример программы обработки байтов:

<i>ассемблер</i>	<i>псевдокод</i>
<b>mov @ddd, R0</b>	<b>R0:=</b> адрес метки ddd
<b>loop:</b>	начало цикла
<b>movb (r0),r1</b>	<b>R1:=</b> байт слова под адресу из R0
<b>or 20,r1</b>	из заглавной буквы сделать строчную
<b>movb r1,(r0)+</b>	записать результат в память
<b>cmpb 0,(r0)</b>	сравнить код следующего байта с 0
<b>jnz loop</b>	если не ноль, перейти в начало цикла
<b>stop</b>	стоп
<b>ddd:</b>	начало блока данных
<b>data "ABCD"</b>	здесь будет "abcd"

Самомодифицирующиеся программы


Поскольку данные находятся в той же области памяти, что и программы, программа может изменять свой код во время выполнения.

Например, для защиты от взлома может быть использовано шифрование: основной код программы зашифрован, и она сама себя расшифровывает при запуске.

Пример самомодифицирующейся программы:

<i>ассемблер</i>	<i>псевдокод</i>
<b>jmp decode</b>	переход на блок расшифровки
<b>main:</b>	начало основной части
<b>data ba6b</b>	в этом и следующем словах будет "mov 123, R0"
<b>data ba98</b>	
<b>data 27a8</b>	здесь будет "system 13"
<b>data 4444</b>	здесь будет "stop"
<b>decode:</b>	начало блока расшифровки
<b>mov @main,r0</b>	<b>R0:=</b> начало зашифрованного блока
<b>mov 4,r1</b>	<b>R1:= 4 ;</b> нужно расшифровать 4 слова
<b>loop:</b>	начало цикла
<b>xor bbbb,(r0)+</b>	расшифровка: хог с маской B <sub>16</sub>
<b>sub 1, r1</b>	уменьшить счетчик
<b>jnz loop</b>	если счетчик не ноль, перейти на начало цикла
<b>jmp main</b>	перейти на основную программу

Расширение ПЗУ

Пользователь может добавить свои подпрограммы в ПЗУ. Для этого нужно сначала отладить подпрограмму, а затем сохранить ее в специальном формате с помощью кнопки  или пункта меню «Программа – Сохранить как ПЗУ».

Например, напишем подпрограмму, которая переставляет биты числа в обратном порядке, используя циклический сдвиг через бит переноса:

<i>ассемблер</i>	<i>псевдокод</i>
<b>mov 1234, R0</b>	<b>R0:= 1234<sub>16</sub></b>
<b>call reverse</b>	вызов подпрограммы
<b>stop</b>	стоп
<b>reverse:</b>	начало подпрограммы
<b>push R1</b>	сохранить R1 в стеке
<b>push R2</b>	сохранить R2 в стеке
<b>mov 10, R2</b>	<b>R2:= 16 = 10<sub>16</sub></b>
<b>xor R1, R1</b>	<b>R1:= 0</b>
<b>next-bit:</b>	



<pre> rcl 1, R0 rcr 1, R1 sub 1, R2 jnz next-bit mov R1, R0 pop R2 pop R1 ret </pre>	<p>старший бит R0 попадает в бит переноса бит переноса попадает в старший бит R1 R2:= R2 – 1 если R2≠0, перейти к метке next-bit R0:= R1 восстановить R2 из стека восстановить R1 из стека возврат из подпрограммы</p>
--	--

Отладив эту программу, уберем верхние три строчки, оставив только процедуру, и применяем команду меню «Программа – Сохранить как ПЗУ». Полученный файл (он будет иметь расширение .rom) открываем в любом текстовом редакторе (например, в Блокноте) и добавляем в начало комментарий:

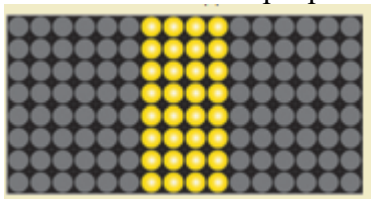
```

; Перестановка битов R0
; в обратном порядке
;-----
2E10 PUSH R1
2E20 PUSH R2
01D2 MOV 10, R2
0010
0911 XOR R1, R1
9E00 RCL 1, R0
AE01 RCR 1, R1
03D2 SUB 1, R2
0001
4D0D JNZ 000A
FFF4
0110 MOV R1, R0
3E20 POP R2
3E10 POP R1
0D00 RET

```

Теперь остается добавить (также в текстовом редакторе) этот фрагмент в конец файла **lampanel.rom**. Новая процедура будет доступна при следующем запуске программы «ЛамПанель». Практическая часть

1. Составьте программу, после выполнения которой ламповая панель выглядит так



2. Как вы думаете, что выведет приведенная выше программа, которая вызывает системную процедуру с номером 5? Проверьте ваш ответ с помощью программы ЛамПанель.
3. Закодируйте изображение домика и выведите его на экран.
4. Напишите программу, которая делает “бегущую строку” из рисунка-домика. Подсказка: используйте команды циклического сдвига.
5. Напишите программу, которая организует “обратный отсчет” от 100 до 0, а затем выводит рисунок с домиком и останавливается. Подсказка: для вывода чисел используйте системную подпрограмму с номером 12<sub>16</sub>.
6. Используя команду MOV, напишите программу, которая заполнит регистры так, как на рисунке. Не забудьте закончить программу командой STOP

```

Регистры
R0 1111 0000 0000 0000
R1 1111 1111 0000 0000
R2 1111 1111 1111 0000
R3 1111 1111 1111 1111

```

Запишите, какие десятичные числа были только что записаны в регистры:

Регистр	Десятичные значения	
	без учета знака	с учетом знака
R0		
R1		
R2		
R3		

7. Выполните программу

SUB 1, R0

NOT R0

STOP

при различных начальных значениях регистра R0 и запишите десятичные значения, полученные в R0 после выполнения программы:

До	После	
	без учета знака	с учетом знака
5		
10		
25		

Какую операцию выполняет этот алгоритм?

8. Используя программу ЛамПанель, вычислите арифметические выражения и запишите результаты в таблицу. Объясните полученные результаты.

Выражение	Результат	
	без учета знака	с учетом знака
65 530 + 9		
32 760 + 9		
8 - 10		

Подсказка:  $65\ 535 = FFFF_{16}$ ,  $32\ 767 = 7FFF_{16}$

9. Вычислите приведенные выражения с помощью программы. Запишите в таблицу результаты, значения знакового (старшего) бита полученного числа и битов состояния:

Выражение	Результат		Знаковый бит	Биты состояния			
	без учета знака	с учетом знака		O	C	Z	N
32 760 + 32 752							
-32 760 - 32 752							
256 - 256							

10. С помощью программы, приведенной в теоретической части, вычислите сумму натуральных чисел от 1 до 100.

11. Напишите программу, которая вычисляет значение факториала — произведения всех натуральных чисел от 1 до заданного числа. Например, факториал числа 5 равен  $5! = 1 \times 2 \times 3 \times 4 \times 5$ . С помощью программы заполните таблицу:

N	N!	
	без учета знака	с учетом знака
5		
6		
7		
8		
9		

Объясните полученные результаты.

12. Напишите программу, которая решает следующую задачу, используя логические операции: В регистрах R1, R2 и R3 записаны коды трех десятичных цифр, составляющих трехзначное число (соответственно сотни, десятки и единицы). Построить в регистре R0 это число. Например, если  $R1=31_{16}$ ,  $R2=32_{16}$  и  $R3=33_{16}$ , в регистре R0 должно получиться десятичное число 123.

13. Используя программу ЛамПанель, определите и запишите в таблицу значения регистра R0 после выполнения каждой из следующих команд, которые выполняются последовательно:



	Команда	R0
1	MOV 1234, R0	
2	XOR ABCD, R0	
3	XOR ABCD, R0	

14. Запишите в таблицу десятичные числа, которые будут получены в регистре R0 после выполнения каждой команды этой программы при разных начальных значениях R0 (две команды выполняются последовательно одна за другой):

Начальное значение	255	254	252	-255	-254	-252
SHR 2, R0						
SHL 2, R0						

В каком случае последовательное выполнение этих двух команд не изменяет данные?

15. Напишите программу, которая решает следующую задачу, используя логические операции и сдвиги:

При кодировании цвета используются 4-битные значения составляющих R (красная), G (зеленая) и B (синяя). Коды этих составляющих записаны в регистрах R1, R2 и R3.

Построить в регистре R0 полный код цвета.

Например, если  $R1 = A_{16}$ ,  $R2 = B_{16}$  и  $R3 = C_{16}$ , в регистре R0 должно получиться число  $ABC_{16}$

16. Напишите программу, которая умножает число в регистре R0 на 10, не применяя команду умножения. Используйте арифметические операции и сдвиги.

Самостоятельная работа

Заполните таблицу обозначение элементов и команд программы

Элемент	Назначение	Принадлежность
R0, R1, R2, R3		
PC;		
PS		
SP		
Команда		
NOP		
STOP		
MOV		
ADD		
SUB		
CMP		
MUL		
DIV		
NOT		
AND		
OR		
XOR		
SHL		
SHR		
SAR		
ROL		
ROR		
RCL		
RCR		
JMP		
JGE		
SYSTEM		
CALL		
RET		

## Практическая работа № 8 Изучение и диагностика блока питания ПК

**Тема:** Виды, характеристики, форм-факторы блоков питания

**Цель:** обобщение, систематизация, углубление, закрепление полученных теоретических знаний о форм-факторах блоков питания, развитие интеллектуальных аналитических умений.

### Теоретическая часть

Компьютерный блок питания (или сокращенно - блок питания, БП) - вторичный источник электропитания, предназначенный для снабжения узлов компьютера электроэнергией постоянного тока путём преобразования сетевого напряжения до требуемых значений. Блок питания в компьютере является основным компонентом из составляющих узлов ПК. От него напрямую зависит правильность и стабильность работы всего компьютера.

Блок питания - это преобразователь электрической энергии поступающей из сети переменного тока в энергию, которая предназначена для питания всей аппаратной части компьютера. Блок питания преобразует сетевое переменное напряжение 220 В, 50 Гц (120 В, 60 Гц) в постоянные напряжения +3,3, +5 и +12 В.

Как правило, для питания цифровых схем (системной платы, плат адаптеров и дисковых накопителей) используется напряжение +3,3 или +5 В, а для двигателей (дисководов и различных вентиляторов) – +12 В. Компьютер работает надежно только в том случае, если значения напряжения в этих цепях не выходят за установленные пределы.

В каждом блоке питания перед получением разрешения на запуск системы выполняется внутренняя проверка и тестирование выходного напряжения. После этого на системную плату посылается специальный сигнал Power\_Good (питание в норме).

Если такой сигнал не поступил, компьютер работать не будет. Напряжение сети может оказаться слишком высоким (или низким) для нормальной работы блока питания, и он может перегреться.

БП имеет множество электрических параметров, большинство из которых не отмечаются в паспорте. На боковой наклейке блока питания отмечается обычно только несколько основных параметров - рабочее напряжение и мощность.

Мощность блока питания часто обозначают на этикетке большим шрифтом. *Мощность блока питания* характеризует, сколько он может отдать электрической энергии приборам, которые подключают к нему (материнская плата, видеокарта, жесткий диск и др.).

*Стабильность напряжений.* Увеличение напряжения вызывает в первую очередь увеличения токов покоя всех схем, а также изменение параметров схем.

*Коэффициент полезного действия* показывает сколько потребленной энергии превратилось в полезную энергию. Чем выше КПД, тем меньше надо платить за электроэнергию потребляемую компьютером.

Основные неэлектрические параметры БП:

- размеры (форм-фактор)
- диапазон рабочих температур;
- надежность (время наработки на отказ);
- уровень шума, создаваемого БП при работе;
- частота обращения вентилятора БП;
- вес;
- длина питательных кабелей;
- удобство в использовании;
- экологичность;

Форм-фактор (от англ. *form factor*) — это стандарт, который задаёт габариты и другие параметры технического изделия. Применительно к персональному компьютеру это количество и размещение комплектующих, разъёмов, элементов корпуса системного блока и прочие технические моменты. Эти стандарты не являются обязательными, но, в целом, производители корпусов и комплектующих для системных блоков стараются их соблюдать.

Блоки питания стандарта АТ: АТ был первым стандартом, который использовался в компьютерных блоках питания.

Блоки питания стандарта АТХ: АТХ новый стандарт, где было увеличено число линий напряжения на выходе. Стандарт АТ необходимо было напряжение в +3,3 В

Блоки питания стандарта ATX 12V (ATX 2.03): Переход на новый стандарт был необходим, так как в 2000 году, появляется новый процессор Intel Pentium 4, который требовал большей мощности блока питания. До него стабилизатор процессора питался от шины +5 Вольт, и если процессор имел мощность 50 Ватт, то сила тока получалась 10 А. При такой нагрузке появились некоторые проблемы с монтажом. Поэтому было принято решение питать стабилизатор процессора от +12V. Соответственно пришлось добавить четырех контактный разъем, который имел два питающих контакта по +12V.

Блоки питания стандарта ATX 12V 2.0 (ATX 2.2): В этом стандарте были внесены некоторые изменения: убрано напряжение питания-5В, и разъем AUX. Этот разъем предполагал дополнительное питание для материнской платы по шинам +5 и +3.3 В и немного напояминал по внешнему виду АТ-разъем. В новом стандарте появились сразу 2-е шины +12 Вольт. Это было сделано для того чтобы «разгрузить» эту шину.

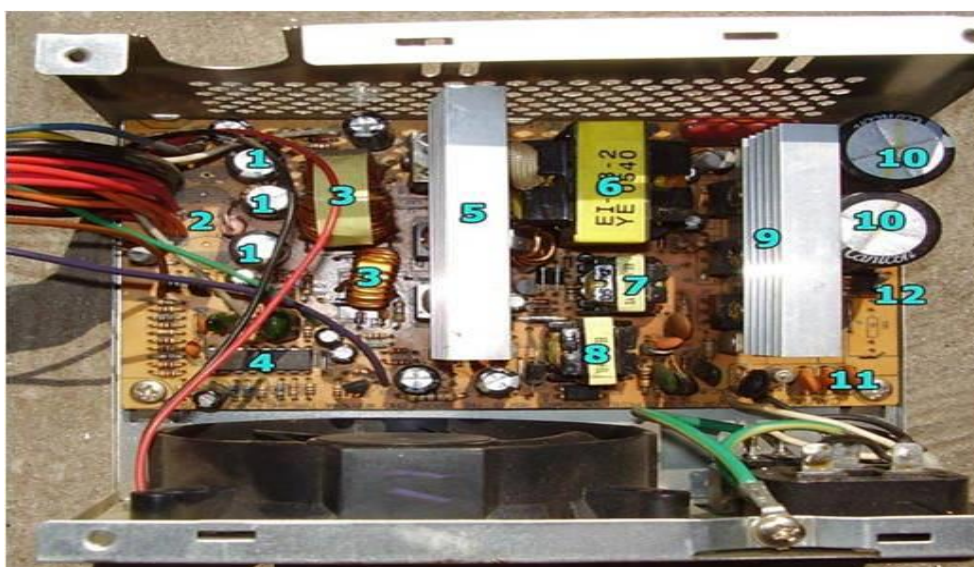
Стандарты массово - выпускаемых БП:

- БП форм-фактора SFX;
- БП форм-фактора TFX;
- БП форм-фактора Flex-ATX.

От форм-фактора зависят габариты БП по ширине, высоте и глубине. Большинство идут в форм-факторе ATX, для стандартных корпусов. В небольших системных блоках стандарта microATX, FlexATX, десктопов и других, устанавливаются блоки меньших размеров, такие как SFX, Flex-ATX и TFX.



На рисунке представлено строение блока питания на 200В форм-фактора ATX



Конденсаторы, которые выполняют фильтрацию исходных напряжений.

Место нераспаянных конденсаторов фильтра исходных напряжений.

Катушки индуктивности, которые выполняют фильтрацию исходных напряжений. Большая катушка играет роль не только фильтра, но еще работает как ферромагнитный стабилизатор. Это позволяет кое-что снизить перекося напряжений при неравномерной нагрузке разных исходных напряжений.

Микросхема ШИМ-Стабилизатора WT7520.

Радиатор, на котором установлены диоды Шоттки для напряжений +3.3В и +5В, а для напряжения +12В обычные диоды.

Основной трансформатор, который осуществляет формирование всех напряжений, а также связь с сетью.

Трансформатор, который формирует управляющие напряжения для исходных транзисторов преобразователя.

Трансформатор преобразователя, который формирует напряжение +5В.

Радиатор, который формирует напряжение.

Конденсаторы фильтра сетевого напряжения.

Элементы фильтра сети от гармоник (препятствий), которые генерируются блоком питания.

Диоды диодного моста, которые осуществляют выпрямление сменного напряжения сети.

**Задание к практической работе:**

**Задание 1.** Блок питания, его предназначение

**Задание 2.** Представьте в виде таблицы характеристику БП форм-факторов SFX, TFX, FLEX, ATX.

Таблица 1- Характеристика форм-факторов блоков питания

Форм-фактор блоков питания	Характеристика

**Задание 3.**  
Изучите строение

блока питания на 200В форм-фактора ATX и представьте в виде схемы, составляющие данного блока питания

**КОНТРОЛЬНЫЕ ВОПРОСЫ:**

1. Назначение блока питания
2. Форм-факторы блоков питания

## Практическая работа № 9 Подключение и настройка мониторов и видеоадаптеров

**Тема:** Изучение способов подключения и настройка мониторов и видеоадаптеров

**Цель:** изучение мониторов и видеоадаптеров интерфейсов подключения.

### Теоретическая часть

Монитор - это устройство оперативной визуальной связи пользователя с управляющим устройством и отображением данных, передаваемых с клавиатуры, мыши или центрального процессора. Современный монитор состоит из экрана (дисплея), блока питания, плат управления и корпуса. Информация для отображения на мониторе поступает с электронного устройства, формирующего видеосигнал (в компьютере — видеокарта или графическое ядро процессора).

Монитор является жизненно важным посредником в обмене информацией между человеком и компьютером, таким же, как клавиатура и мышь.

Система отображения компьютера состоит из двух главных компонентов:

- монитор (дисплей) обычно представляет собой жидкокристаллический экран или переднюю панель электронно-лучевой трубки, но может быть и широкоформатным телевизором, плазменной панелью и проектором, использующими технологии LCD и DLP.
- видеоадаптер (графический адаптер или видеокарта) в большинстве систем представляет собой карту расширения, вставляемую в один из разъемов материнской платы. В некоторых системах он интегрирован в саму системную плату или в ее набор микросхем системной логики, однако и такие компьютеры можно дополнить обособленным и более производительным видеоадаптером AGP, PCI или PCI-Express.

Компьютерный монитор обычно базируется на одной из двух основных технологий: жидкокристаллический дисплей LCD (LiquidCrystalDisplay) или электронно-лучевая трубка CRT (Cathode-RayTube). Проекторы базируются на технологии LCD или DLP (DigitalLightProcessing — цифровая обработка света).

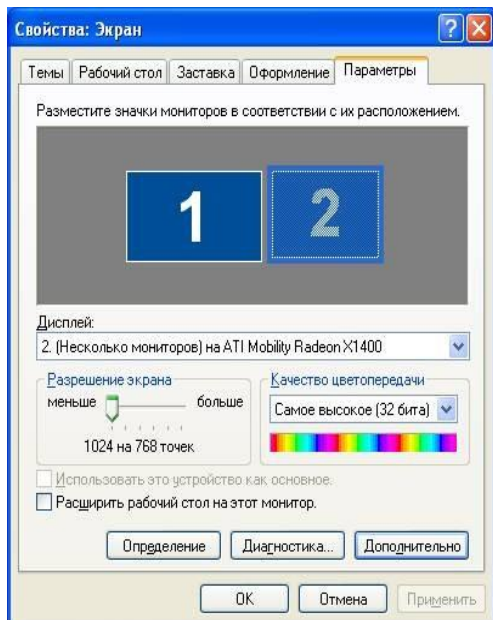
Жидкокристаллические (ЖК, LCD) мониторы благодаря своему малому весу, размерам и цветопередаче в настоящее время практически вытеснили с рынка мониторы на электронно-лучевой трубке (ЭЛТ, CRT). Настольные LCD-мониторы во многом похожи на экраны ноутбуков. По сравнению с классическими ЭЛТ-мониторами у них есть целый ряд преимуществ: плоский экран без бликов и очень низкий уровень энергопотребления (5 Вт по сравнению со 100 Вт, характерными для обычного ЭЛТ-монитора).

Хотя жидкокристаллические мониторы по своей природе являются цифровыми, многие компьютеры содержат только аналоговые порты для подключения монитора. Поэтому стандартный жидкокристаллический монитор может быть оснащен аналоговым портом VGA, цифровым портом DVI или и тем, и другим. Как правило, бюджетные модели жидкокристаллических мониторов с размером экрана от 15 до 19 дюймов оснащены традиционным аналоговым разъемом VGA, а значит, им приходится снова преобразовывать аналоговые сигналы в цифровые. В то же время более дорогие модели мониторов оснащены аналоговым разъемом VGA и цифровым разъемом DVI, который сейчас имеют практически все видеоадаптеры.

Видеоадаптер обеспечивает интерфейс между компьютером и монитором, передавая сигналы, которые превращаются в изображение, которое мы видим на экране.

Видеоадаптер (от англ. videocard) — устройство, преобразующее изображение, находящееся в памяти компьютера, в видеосигнал для монитора.

В состав видеоподсистемы персонального компьютера входят видеоадаптер и монитор.



Компонент «Экран» используется для настройки параметров рабочего стола и экрана. Здесь можно выбрать тему, определяющую общий внешний вид рабочего стола. Тема задает фон, заставку, используемые в окнах системные шрифты, цвета и объемные эффекты, внешний вид значков и указателей мыши, а также звуковые эффекты. Пользователи могут настраивать темы, изменяя отдельные элементы. На закладке «Параметры» можно установить требуемые параметры видеоадаптера: количество цветов, разрешение экрана, а также частоту обновления для монитора.

К персональному компьютеру может быть подключено два и более мониторов. Сделать это можно двумя способами:

1. Второй монитор может быть подключен к видеоадаптеру, имеющему два видеопорта. О возможности такого подключения необходимо узнать из документации на видеоадаптер. Наличие двух видео разъемов (особенно если они разные – VGA и DVI) может быть связано с поддержкой

двух разных интерфейсов и такой адаптер имеет только один видеопорт.

2. Можно установить на материнскую плату персонального компьютера дополнительный видеоадаптер и подключить к нему дополнительный монитор. При загрузке компьютера с двумя подключенными мониторами диалоговое окно входа в систему будет выводиться на мониторе, заданном в качестве основного. Кроме того, при открытии большинства приложений их окна будут также отображаться на основном мониторе.

При использовании нескольких мониторов можно задать параметры для каждого из них в отдельности. Подключив дополнительные мониторы, можно создать большой рабочий стол, способный вместить огромное число приложений и окон. В такой конфигурации можно работать над несколькими задачами одновременно, перемещая элементы с одного монитора на другой или растягивая их на несколько мониторов.

### Практическая часть

**Задание 1.** Назовите и охарактеризуйте компоненты системы отображения компьютера

**Задание 2.** На панели управления откройте компонент «Экран» и изучите настройку разрешения экрана, настройку яркости, настройки параметров экрана, изменение размеров текста

**Задание 3.** Установка разрешения экрана

3.1. На панели управления откройте компонент «Экран».

3.2. На вкладке Настройка перетащите ползунок в группе Разрешение экрана, установив требуемое разрешение, а затем нажмите кнопку Применить.

3.3 Сделайте скриншот

3.4. Когда появится запрос на применение новой настройки, нажмите кнопку ОК. Экран ненадолго станет черным.

3.5. После изменения разрешения экрана необходимо в течение 15 секунд подтвердить его. Для этого следует нажать кнопку Да; если нажать кнопку Нет или ждать, ничего не нажимая, будет восстановлен предыдущий уровень разрешения.

**Задание 4.** Установка частоты обновления изображения на мониторе

4.1. На панели управления откройте компонент «Экран».

4.2. На вкладке Параметры нажмите кнопку Дополнительно.

4.3. На вкладке Монитор выберите в списке Частота обновления экрана требуемое значение частоты обновления.

4.4. Сделайте скриншот

**Задание 5.** Изменение размера объектов и текста на экране

5.1. На панели управления откройте компонент «Экран».

5.2. На вкладке Параметры нажмите кнопку Дополнительно.

5.3. На вкладке Общие в списке Масштабный коэффициент выберите нужное значение величины dpi (dotsperinch - точек на дюйм).

5.4. Сделайте скриншот



## КОНТРОЛЬНЫЕ ВОПРОСЫ:

1. Что такое монитор?
2. Что такое видеоадаптер?
3. Перечислите виды мониторов
4. Назовите и охарактеризуйте компоненты системы отображения компьютера
5. Назовите способы подключения компьютера к монитору

## Практическая работа №10 Устройство ввода информации в ЭВМ

**Тема:** устройство клавиатуры и мыши, настройка параметров работы клавиатуры и мыши.

**Цели:** изучение основных компонентов персонального компьютера и основных видов периферийного оборудования, способов их подключения, основных характеристик (название, тип разъема, скорость передачи данных, дополнительные свойства).

**Студент должен знать:**

- по внешнему виду типов разъемов и подключаемого к ним оборудования;
- **уметь:**
- получать сведения по настройке пользовательского интерфейса периферийных устройств средствами операционной системы **Microsoft Windows**.

## Теоретическое обоснование

### Подключения устройств к системному блоку

Все периферийные устройства подключаются только к системному блоку. Для работы конкретного устройства в составе конкретного комплекта ПЭВМ необходимо иметь:

Контроллер (адаптер) – специальную плату, управляющую работой конкретного периферийного устройства. Например, контроллер клавиатуры, мыши, адаптер монитора, портов и т.п.

Драйвер – специальное программное обеспечение, управляющее работой конкретного периферийного устройства. Например, драйвер клавиатуры, драйвер принтера и т.п.

Для управления работой устройств в компьютерах используются электронные схемы – *контроллеры*. Различные устройства используют разные способы подключения к контроллерам:

некоторые устройства (дисковод для дискет, клавиатура и т. д.) подключаются к имеющимся в составе компьютера стандартным контроллерам (интегрированным или встроенным в материнскую плату);

некоторые устройства (звуковые карты, многие факс-модемы и т. д.) выполнены как электронные платы, т. е. смонтированы на одной плате со своим контроллером;

некоторые устройства используют следующий способ подключения: в системный блок компьютера вставляется электронная плата (контроллер), управляющая работой устройства, а само устройство подсоединяется к этой плате кабелем;

на сегодняшний день большинство внешних устройств подключаются к компьютеру через USB-порт.

Платы контроллеров вставляются в специальные разъемы (слоты) на материнской плате компьютера.

С помощью добавления и замены плат контроллеров пользователь может модифицировать компьютер, расширяя его возможности и настраивая его по своим потребностям. Например, пользователь может добавить в компьютер факс-модем, звуковую карту, плату приема телепередач и т. д.

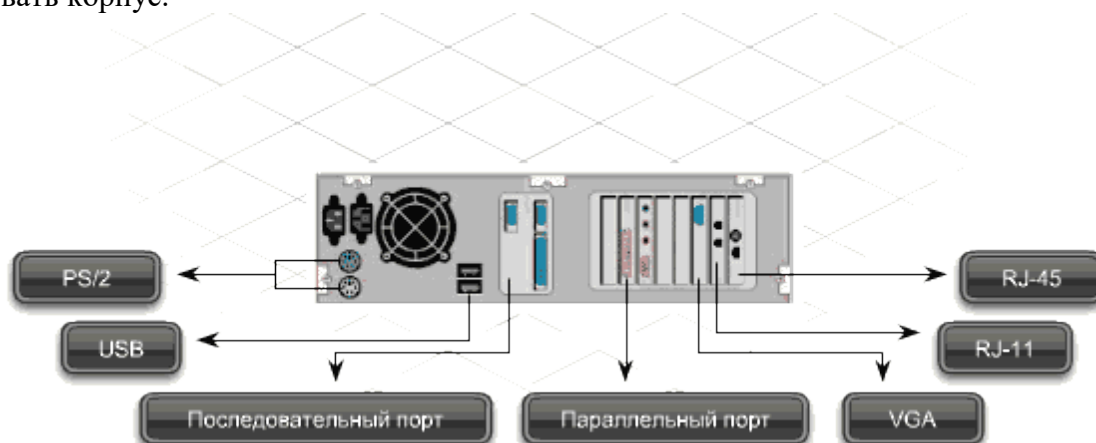
Одним из видов контроллеров, которые присутствуют почти в каждом компьютере, является контроллер портов ввода-вывода. Часто этот контроллер интегрирован в состав материнской платы.

Контроллер портов ввода-вывода соединяется кабелями с разъемами на задней стенке компьютера, через которые к компьютеру подключаются принтер, мышь и некоторые другие устройства.

Кроме контроллеров портов ввода-вывода в системном блоке присутствуют разъемы шины USB – универсальной последовательной шины, к которой можно подключить клавиатуру, мышь, принтер, модем, дисковод компакт-дисков, сканер и т. д. Основное требование возможность подключения к

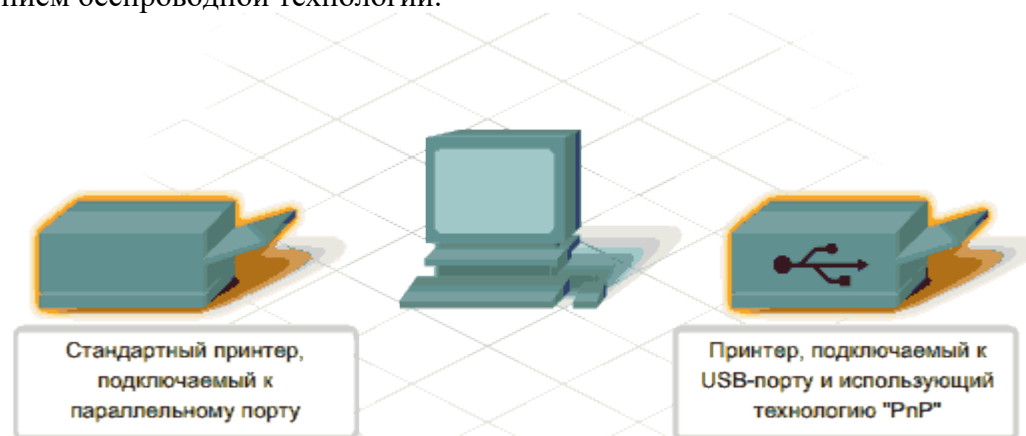
данной шине устройства. Особенность шины USB – возможность подключения к ней устройств во время работы компьютера (не выключая его).

В отличие от внутренних компонентов, для установки периферийных устройств не нужно открывать корпус.



Периферийные устройства подключаются к разъемам на внешней части корпуса с помощью проводов или беспроводной связи. Исторически периферийные устройства разработаны в расчете на определенный тип портов. Например, в конструкции принтеров для персональных компьютеров предусмотрено подключение к параллельному порту, который передает от компьютера к принтеру данные в определенном формате.

Разработанный интерфейс универсальной последовательной шины (USB) сильно упростил использование проводных периферийных устройств. USB-устройства не требуют сложных процедур конфигурации. Они просто подключаются к соответствующему порту (при наличии нужного драйвера). Кроме того, все чаще появляются устройства, которые подключаются к узлу с использованием беспроводной технологии.



**Установка периферийного устройства** выполняется в несколько этапов. Порядок и тип этих шагов зависит от типа физического подключения и от того, относится ли устройство к типу автоматически настраивающихся (PnP). Предусмотрены следующие шаги:

подсоединение периферийного устройства к узлу с помощью соответствующего кабеля или беспроводного соединения;

подключение устройства к источнику питания;

установка соответствующего драйвера.

Некоторые устаревшие устройства, так называемые "обычные устройства", не предусматривают самонастройки. Драйверы таких устройств устанавливаются после того, как устройство подключается к компьютеру и включается питание.

Драйверы самонастраивающихся USB-устройств в системе уже имеются. В таком случае при подключении и включении операционная система распознает устройство и устанавливает соответствующий драйвер.

### **Организация электропитания устройств компьютера**

Все устройства, имеющиеся внутри системного блока (как внутренние, так и внешние) питаются через блок питания системного блока.

Монитор поддерживает две схемы подключения:



- через системный блок (в этом случае при включении/выключении системного блока одновременно включается/выключается монитор);
- через отдельную розетку (тогда при включении системного блока монитор нужно включить через отдельную кнопку).

Все остальные периферийные устройства, расположенные в отдельных корпусах, имеют отдельные кабели питания, подключаемые к собственным розеткам.

Несмотря на высокую надежность и безопасность, компьютерное оборудование должно быть заземлено.

Во избежание нежелательных последствий скачков напряжения электросети (потеря и порча информации, выход из строя компьютерного оборудования и пр.), ПК обычно подключают к электросети через источник бесперебойного питания — UPS, который стабилизирует подаваемое на аппаратуру напряжение от сети и, при его полном отключении, поддерживает питание компьютера в течение некоторого (обычно 15—20 минут) времени, чтобы пользователь успел завершить работу запущенных программ, сохранить необходимую информацию и выключить компьютер по стандартной схеме.

Если в составе комплекта компьютерного оборудования нет UPS, то желательно использовать хотя бы обычный стабилизатор напряжения или специальный сетевой фильтр.

### Разъемы для подключения электропитания и внешних устройств

**Расположение разъемов.** Обычно разъемы для подключения электропитания и внешних устройств находятся на задней стенке системного блока компьютера. В портативных компьютерах эти разъемы могут находиться и с боковых сторон системного блока.

Подключение электропитания и внешних устройств к компьютеру выполняется с помощью специальных проводов (кабелей). Для защиты от ошибок разъемы для вставки этих кабелей сделаны разными, так что кабель, как правило, просто не вставится в неподходящее гнездо.

**Разъемы для электропитания.** Обычно на задней стенке системного блока компьютера имеется разъем для подсоединения к электросети и разъем для подачи электропитания на монитор.

**Разъемы для подключения внешних устройств.** Разъемы для подключения внешних устройств достаточно унифицированы. Распространены разъемы со штырьками и с гнездами.

**Порядок вставки кабелей.** Вставлять и вынимать кабели из разъемов можно только при выключенном компьютере, в противном случае компьютер и устройства могут быть испорчены.

Вставлять кабели надо аккуратно, чтобы не погнуть и не поломать содержащиеся в разъемах штырьки.

Некоторые кабели (например, для монитора или принтера) закрепляются с помощью винтов, эти винты надо завернуть рукой или отверткой (только не слишком туго), чтобы кабель не выпадал из разъема во время работы компьютера.

### Периферийные устройства

**К внешним или периферийным устройствам** персонального компьютера относятся устройствами, которые предназначены для **ввода информации** в компьютер для обработки, и устройствами, для **вывода информации** из него.

Есть еще и устройства для **хранения информации** вне системного блока (внешние накопители).

#### Клавиатура и мышь

Это основные устройства для ввода информации, необходимость в них очевидна и не обсуждается, пока сенсорные мониторы не вытеснили обычные : )

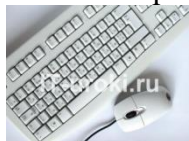


Рисунок 7.1. Клавиатура и мышь

*Подключаться клавиатура и мышь к компьютеру может с помощью двух видов разъемов:*

**PS/2**

**USB.**



Рисунок 7.2. Разъемы на задней панели системного блока

Клавиатуру и мышь с разъемом **PS/2** можно **подключать или отключать только при выключенном компьютере**. Если они выполнены с **USB**-разъемами, то их можно **подключать или отключать** в любой момент, **даже если компьютер работает**.

Акустические системы (колонки)

Акустические системы **преобразуют электрический сигнал**, полученный со звуковой карты компьютера в звуковые колебания (т.е. **в звук**) и относятся к устройствам вывода информации.

На сегодняшний день акустические системы можно назвать обязательными устройствами ПК.

Чтобы слушать музыку, смотреть фильмы, слушать аудиокниги, а также изучать IT-уроки, без «колонок» не обойтись.



Рисунок 7.3. Акустические системы (колонки)

Акустические системы подключаются с помощью **разъема «мини-джек» зеленого цвета** (на фотографии один из разъемов, обозначенных **цифрой 3**). Подключать и отключать можно при работающем компьютере.

Если динамики встроенные в ноутбук не устраивают по качеству звучания, то Вы также можете использовать внешние акустические системы, только подключаются они в разъем для наушников (о нем смотрите ниже.)

Наушники

**Наушники**, можно считать разновидностью **акустических систем**, только предназначены они **для одного пользователя** ПК. Фактически, наушники – это миниатюрные акустические системы для индивидуального использования.



**Наушники**

*Подключаются наушники к компьютеру так же, как и акустические системы.*

На ноутбуках разъем не выделен зеленым цветом, но возле него присутствует значок «наушники».

Микрофон

Микрофон может понадобиться в двух случаях:

**Общение через Интернет** (например, через Skype);

**Запись звука** с целью дальнейшего хранения и обработки на ПК.

Как можно догадаться, микрофон является устройством ввода информации.



### **Микрофон**

*Микрофон подключается к ПК с помощью разъема «мини-джек» розового цвета.*

Веб-камера (Web-камера)

**Веб-камера** может понадобиться, если Вы планируете часто общаться через Интернет, и при этом хотелось бы **передавать** не только голос, но и своё **изображение** (следует учитывать, что скорость Интернета должна быть достаточной для передачи видеосигнала).



### **Веб-камера на ноутбуке (встроенная)**

Видео-звонки очень увлекательны, но не обязательны при живом общении. Лично я считаю это периферийное устройство одним из последних в очереди на покупку, но некоторые со мной не согласятся :).

Веб-камера может иметь встроенный микрофон, что освобождает Вас от покупки «лишнего» устройства.

*Подключается к компьютеру камера через **USB-разъем** (цифра 2 на рисунке 7.2).*

USB-флэш-накопитель (флэшка)

**Флэш-накопители** относятся к устройствам хранения данных, т.е. предназначены для **хранения и переноса информации** между компьютерами.



### **USB-флэш-накопитель (флэшка)**

Кстати, рекомендую всем пользователям компьютера использовать флэшку не только по прямому назначению, но и как одно из средств **резервного копирования данных** (об этом мы поговорим в одном из ближайших IT-уроков).

Конечно же, приобрести флэшку желательно уже на первых шагах освоения компьютера.

*Из названия можно догадаться, что **подключается USB-флэш-накопитель с помощью USB-разъема**.*

Не забывайте про правило, что извлекать флэшку из разъема можно после программного отключения (значок возле системных часов).

Кардридер (он же картридер)

Если у Вас есть **фотоаппарат или видеокамера** (или **смартфон**), то кардридер значительно упростит работу с фотографиями и видеофайлами при переносе на компьютер.



### **Внешний кардридер**

Не обязательно искать «родной» провод для подключения устройства к USB, достаточно **извлечь карту памяти (флэш-карту)** и вставить в нужный разъем на **кардридере**. Кроме того, некоторые устройства (например мобильные телефоны) не позволяют свободно (напрямую) работать с файлами на карте памяти, но с помощью кардридера вы вольны **работать с данными как с обычной флэшкой**.

Кардридер может быть:

**внешний** (указан на фотографии выше);

**внутренний** (устанавливается в системный блок) или

**встроенный** (обязателен в современных ноутбуках).

***Внешний кардридер** подключается к системному блоку **через USB-разъем**.*

Обычно нельзя подключить к компьютерной системе новое устройство, не установив или не настроив соответствующее программное обеспечение. К счастью, операционная система Windows обычно делает это автоматически.

Несмотря на автоматическое распознавание, пожалуйста, внимательно изучите инструкции по установке, которые поставляются с каждым устройством. Иногда перед установкой нужно устанавливать специальное программное обеспечение. Как об этом узнать? Прочитать инструкцию! Способность Windows автоматически распознавать новые устройства называется **plug-and-play** (подключай и играй). Программа, управляющая работой устройства, называется **драйвером**.

### Практические задания

**Задание 1:** Осуществить подключение периферийного оборудования, используя различные интерфейсы: жесткий диск, привод на компакт-дисках, флоппи-дисковод, плату видеоадаптера, сетевую карту, звуковую карту, монитор, принтер, сканер.

#### Изучение разъемов для подключения электропитания и внешних устройств

Посмотрите на обратную сторону системного блока с подключенными кабелями. Поочередно вынимая кабель конкретного устройства зарисуйте разъемы системного блока. Результаты изучения разъемов занесите в таблицу.

Устройство	Схема разъема, к которому подключено
Электропитание	
Клавиатура	
Мышь	
Питание монитора	
Монитор	
Сетевой кабель	
USB порт	

### Контрольные вопросы

- 1 Расскажите, как осуществляется питание различных устройств ПК.
- 2 Посчитайте количество необходимых розеток электропитания, если в состав ПК входят сканер и принтер.
- 3 В чем состоит особенность электропитания мониторов?
- 4 Какие устройства используются в ПК для стабилизации напряжения в сети?
- 5 Какое устройство в составе ПК позволяет стандартно завершить работу аппаратуры при внезапном отключении электропитания?
- 6 Как осуществляется подключение электропитания и внешних устройств в компьютере?

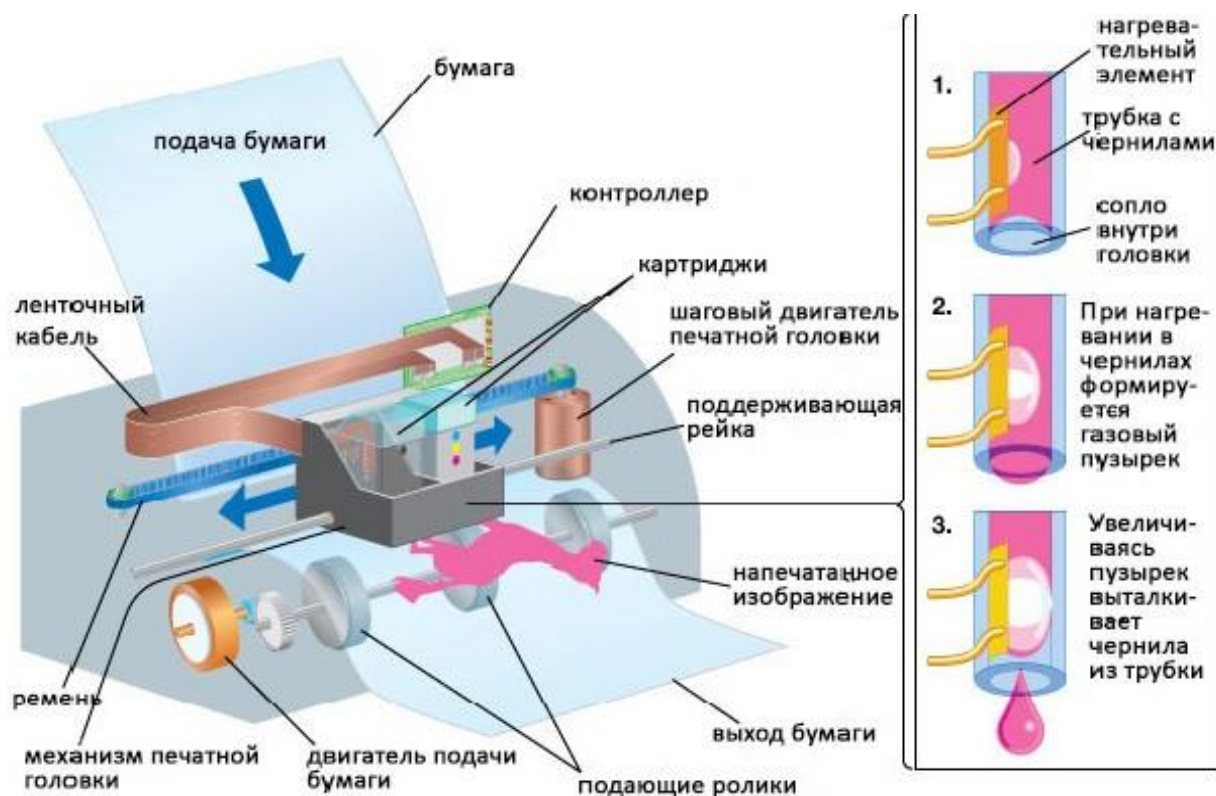
## Практическая работа № 11 Подключение и инсталляция принтеров

### Тема: конструкция, подключение и инсталляция принтера

**Цель занятия:** обобщение, систематизация, углубление, закрепление полученных теоретических знаний о принтерах, развитие интеллектуальных аналитических умений

Краткие теоретические сведения, необходимые для выполнения практической работы:

Первой фирмой, изготовившей струйный принтер, является Hewlett-Packard. Основным принципом работы струйного принтера чем-то напоминает работу игольчатых принтеров, только вместо иголок здесь применяются тонкие, как волос, сопла, которые находятся в головке принтера. В этой головке установлен резервуар с жидкими чернилами, которые через сопла, как микрочастицы, переносятся на материал носителя. Число сопел (от 16 до 64) зависят от модели принтера и изготовителя. Некоторые последние модели имеют гораздо большее число сопел. Например, головка HP DeskJet 1600 имеет 300 сопел для черных чернил и 416 для цветных.



Для хранения чернил применяются два метода:

- головка принтера объединена с резервуаром для чернил; замена резервуара с чернилами одновременно связана с заменой головки;
- используется отдельный резервуар, который через систему капилляров обеспечивает чернилами головку принтера.

Современные модели струйных принтеров в своей работе могут использовать следующие методы:

- пьезоэлектрический метод;
- метод газовых пузырей;
- метод drop-on-demand.

Пьезоэлектрический метод. Для реализации этого метода в каждое сопло установлен плоский пьезокристалл, связанный с диаграммой. Как известно, под воздействием электрического поля происходит деформация пьезоэлемента. При печати, находящийся в трубке пьезоэлемент, сжимая и разжимая трубку, наполняет капиллярную систему чернилами. Чернила, которые отжимают назад, перетекают обратно в резервуар, а чернила, которые «выдавлились» наружу, оставляют на бумаге точку. Подобные устройства выпускают компании Epson, Brother и др.

Метод газовых пузырей. Этот способ является термическим и больше известен под названием Bubblejet (инжектируемые пузырьки). При использовании этого метода каждое сопло оборудовано нагревательным элементом, который при пропускании через него тока за несколько микросекунд нагревается до температуры около 500. возникающее при резком нагревании газы образуют пузырьки (bubbles) стараются вытолкнуть через выходное отверстие сопла необходимую каплю жидких чернил, которая переносится на бумагу. При отключении тока нагревательный элемент остывает, паровой пузырь уменьшается и через входное отверстие поступает новая порция чернил. Подобную технологию использует фирма сапон.

Разрез сопла термоэлектрической печатной головки.

1. Сопло. Для повышения качества и скорости печати одна головка может содержать более ста сопел. Как правило, они располагаются в несколько рядов так, чтобы каждое сопло печатало свою строку точек на носителе. Наличие нескольких рядов и строк сопел позволяет за один проход напечатать больше точек, что ведет к увеличению скорости печати.

2. Канал для подвода чернил к соплу. Именно в подводящем канале, выполненном в стеклянной или кремниевой основе печатающего блока, происходит образование пузырьков пара, выталкивающих чернила через сопла. В канал чернила попадают из резервуара картриджа (чернильницы).

3. Проводники управляющих сигналов. Для работы элемента, нагревающего чернила, необходимо подать на него определенное управляющее напряжение при помощи проводников. Одними концами проводники соединяются с нагревательным элементом, другие концы выводятся на контактную пластину картриджа, которая позволяет соединить непрочные проводники с контактами гнезда картриджа без повреждений.

4. Нагревательный элемент. Нагревательные элементы, являются важнейшими компонентами термоэлектрической печатающей головки. Именно они нагревают чернила в подводящем канале до кипения растворителя и образования пузырьков пара.

Метод drop-on-demand. Данный метод очень схож с вышеописанным, отличие состоит в том, что для подачи чернил используется бумагопротяжный механизм. Бумагопротяжный вал не имеет ручной протяжки бумаги.

Подключение струйных принтеров к ПК производится через IP-порт или через порт USB, которым, как правило, оснащены все компьютеры. Для подключения принтера необходимо наличие в системном блоке порта, через который принтер должен работать, соединительный кабель, а также непосредственно драйвер принтера. Связь беспроводного принтера с компьютером осуществляется

посредством радиоволн с использованием технологии Bluetooth или Wi-Fi. Большинство адаптеров Bluetooth подключаются к USB-порту.

### Задание к практической работе:

**Задание 1.** Охарактеризуйте струйный принтер

**Задание 2.** Заполните таблицу «Методы работы струйного принтера»

Таблица 1- Методы работы струйного принтера

Методы работы струйного принтера	Характеристика

**Задание 3.** Перечислите достоинства и недостатки струйного принтера

**Задание 4.** Подключите принтер к ПК, изучите параметры настройки принтера, произведите печать в различных режимах, сделайте анализ отпечатанных результатов

#### КОНТРОЛЬНЫЕ ВОПРОСЫ:

1. Что такое струйный принтер?
2. Назовите методы работы струйного принтера

### Практическая работа № 12 Инсталляция драйверов устройств

Тема: Установка драйверов устройств

Цель занятия: обобщение, систематизация, углубление, закрепление полученных теоретических знаний об драйверах устройств, развитие интеллектуальных аналитических умений.

Основные требования по технике безопасности при выполнении практической работы: изучить правила техники безопасности, руководствоваться ими и обеспечить их строгое соблюдение при проведении учебного процесса

Краткие теоретические сведения, необходимые для выполнения практической работы:

Драйвер (от английского Driver – водитель) – программа, отвечающая за работу данного устройства или оборудования, которая обеспечивает связь между компьютером и устройством. Драйверы еще иногда называют «дрова», а любое устройство называют еще «девайсом».

В Windows предусмотрена возможность автоматического поиска и установки необходимых драйверов через центр онлайн-обновления Windows Update.

Порядок установки драйвера через центр обновления Windows следующий:

1. Открыть "Диспетчер устройств Windows"
2. В диспетчере щелкнуть правой кнопкой мышки по устройству, для которого устанавливается драйвер.
3. В появившемся контекстном меню выбрать пункт "Обновить драйвер".

Дальше порядок действий зависит от версии Windows:



– в Windows XP – в первом открывшемся окне выбрать пункт "Только в этот раз" и нажать "Далее"; во втором – выбрать "Автоматически", нажать "Далее" и дождаться окончания процесса установки;

– в Windows Vista, Windows 7, Windows 8 - в открывшемся окне нажать на "Автоматический поиск драйверов" и дождаться окончания процесса установки.

Во многих случаях драйвер устройства не удастся установить вообще. Такие ситуации возникают при попытках использования новых операционных систем на старом оборудовании. Дело в том, что для новых версий Windows часто не существует драйверов старых устройств. Их производители перешли на выпуск более современной продукции, полностью прекратив техническую поддержку своих старых изделий. В таком случае придется либо приобретать новое оборудование, либо использовать старую версию Windows.

Драйверы иногда находит операционная система Windows 7 самостоятельно в Интернете и сама устанавливает их, так же драйверы можно установить вручную с CD или DVD диска, либо установить вручную с официального сайта производителя устройства. Однако для некоторых устройств могут потребоваться специальные драйверы, обычно предоставляемые на лицензионном CD или DVD диске производителем конкретного устройства. Драйверы необходимы не только для внешних устройств типа принтера, веб-камеры или мышки. Драйверы необходимы и для внутренних устройств, например, для материнской (системной) платы, для видеокарты, для звуковой карты. Если видеокарта или звуковая карта встроены в материнскую плату, то драйвера для них не нужны, потому что они встроены в драйвера для материнской платы. Если видеокарта или звуковая карта устанавливаются отдельно, то для каждой карты нужен отдельный диск с драйверами.

**Чтобы установить драйвер устройства с оптического диска** поместите диск в привод компьютера. Через непродолжительное время откроется окно автозапуска диска, в котором необходимо нажать мышкой на кнопку "Установить драйвер", "Установить", "Install", "Setup Driver" или с каким-нибудь похожим названием. Драйвер устанавливается автоматически. В процессе установки необходимо ознакомиться с лицензионным соглашением и ответить на несколько дополнительных вопросов (обычно во всех диалоговых окнах достаточно просто нажимать кнопку продолжения). После установки драйвера нужно извлечь диск из дисковода и перезагрузить компьютер.

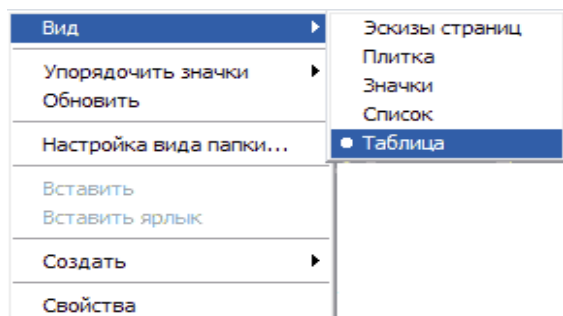
**Порядок установки драйвера устройства из исполняемого файла** зависит от того, находится он в архиве или нет. Чтобы это узнать, откройте папку с драйвером и переведите отображение находящихся в ней файлов в табличную форму.

Для этого необходимо навести указатель мышки на пустое место в папке и щелкнуть правой кнопкой мышки. Откроется контекстное меню, в котором нужно навести указатель мышки на пункт



"Вид". Рядом откроется подменю. В нем нужно активировать пункт "Таблица". Затем посмотрите, что отображается в столбце "Тип файла" напротив файла драйвера:

- если это приложение – дважды щелкаем по файлу левой кнопкой мышки. Запустится мастер установки, в котором нужно подтвердить свое согласие с лицензионным соглашением и ответить на

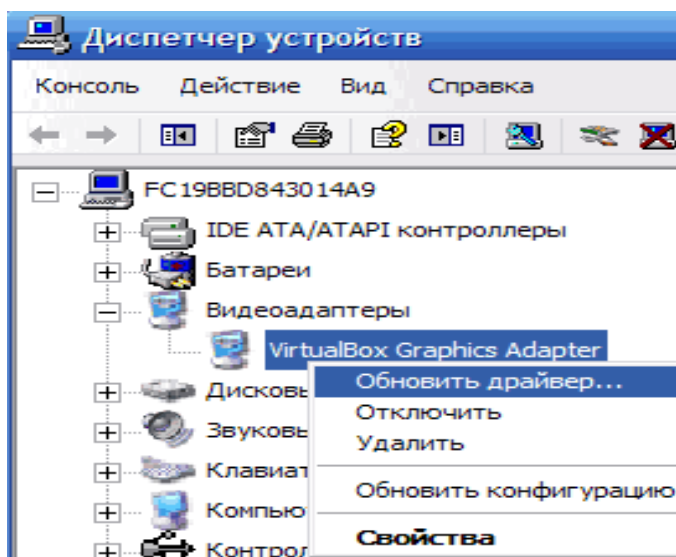


несколько вопросов, нажимая в каждом диалоговом окне кнопку продолжения. После установки драйвера перезагрузите компьютер;

- если это архив – распаковываем его. После распаковки заходим в папку с извлеченными из архива файлами, переводим отображение находящихся в ней файлов

в табличную форму, находим среди них файл-приложение. Как правило, нужный файл имеет название setup, driver, install или что-то похожее. Далее запускаем процесс установки драйвера двойным щелчком левой кнопки мышки по файлу-приложению.

Если среди файлов, извлеченных из архива, отсутствуют приложения, драйвер придется устанавливать через "Диспетчер устройств Windows".



Установку драйвера через Диспетчер устройств целесообразно проводить, если среди файлов драйвера отсутствует исполняемый файл (файл-приложение). В диспетчере нужно щелкнуть правой кнопкой мышки по устройству, для которого устанавливается драйвер, и в открывшемся контекстном меню выбрать пункт "Обновить драйвер".

**Задание к практической работе:**

**Задание 1.** Драйвер устройств, его назначение.

**Задание 2.** Заполните таблицу «Способы установки драйверов».

Таблица 1- Способы установки драйверов

Способ установки драйвера	Описание способа установки драйвера

**КОНТРОЛЬНЫЕ ВОПРОСЫ:**

1. Драйвер устройств
2. Способы установки драйверов

## ЛИТЕРАТУРА

### Основные источники:

1. Сенкевич, А. В. Архитектура аппаратных средств: учебник / А. В. Сенкевич. Архитектура аппаратных средств: учебник для студентов учреждений среднего профессионального образования / А. В. Сенкевич. - 3-е изд., перераб. - Москва: Академия, 2020. - 254, [1]с.: ил. - (Профессиональное образование.ТОП 50). - ISBN 978-5-4468-8689-0. - Текст : непосредственный.
2. Степина, В. В. Архитектура ЭВМ и вычислительные системы: учебник / В.В. Степина. — Москва: КУРС: ИНФРА-М, 2021. — 384 с. — (Среднее профессиональное образование). - ISBN 978-5-906923-07-3. - URL: <https://znanium.com/catalog/product/1423169> (дата обращения: 17.02.2022). – Текст: электронный.

### Дополнительные источники:

1. Информатика и образование: научно – методический журнал / учредители Российская академия образования, Издательство «Образование и информатика». – Ежемес. – 2018 – 2021. – ISSN 0234 – 0453. – Текст: непосредственный.
2. Колдаев, В. Д. Архитектура ЭВМ: учебное пособие / В.Д. Колдаев, С.А. Лупин. — Москва: ФОРУМ: ИНФРА-М, 2021. — 383 с. — (Среднее профессиональное образование). - ISBN 978-5-16-105885-5. - URL: <https://znanium.com/catalog/product/1136788> (дата обращения: 17.02.2022). – Текст: электронный.
3. Максимов, Н. В. Архитектура ЭВМ и вычислительных систем: учебник / Н.В. Максимов, Т.Л. Партыка, И.И. Попов. — 5-е изд., перераб. и доп. — Москва: ФОРУМ: ИНФРА-М, 2022. — 511 с. — (Среднее профессиональное образование). - ISBN 978-5-16-106243-2. - URL: <https://znanium.com/catalog/product/1856720> (дата обращения: 17.02.2022). – Текст: электронный.
4. Методические указания по выполнению практических работ по дисциплине ОП.02 Архитектура аппаратных средств специальность 09.02.07 Информационные системы и программирование/ составитель В.А. Алексеев; Лянторский нефтяной техникум. - Лянтор: ЛНТ, 2019.- 44 с. – Режим доступа: Полнотекстовая коллекция учебно-методических изданий ЛНТ (филиал) ФГБОУ ВО «ЮГУ»