



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

**КАФЕДРА «МАТЕМАТИКА И ИНФОРМАТИКА»**

## **Конспект лекций (часть 1)**

для обучающихся заочной формы обучения

по дисциплине

**«Информатика и программирование»**

Ростов-на-Дону

2025

УДК 004

Составитель: канд. техн. наук, доцент Литвинов В.Н.

Конспект лекций (часть 1) по дисциплине «Информатика и программирование». ДГТУ, г. Ростов-на-Дону, 2025 г.

В конспекте лекций кратко изложены теоретические вопросы, необходимые для успешного освоения дисциплины «Информатика и программирование» рабочее задание и контрольные вопросы для самопроверки.

Предназначено для обучающихся заочной формы обучения по направлениям подготовки 15.03.01 Машиностроение, 15.03.05 «Конструкторско-технологическое обеспечение машиностроительных производств».

**СОДЕРЖАНИЕ**

<b>Введение .....</b>	<b>4</b>
<b>Лекция № 1. Позиционные системы счисления.</b>	
Операции над числами в позиционных системах счисления .....	5
<b>Лекция № 2. Количество информации.....</b>	<b>14</b>
<b>Лекция № 3. Логические основы работы компьютера.....</b>	<b>25</b>
<b>Лекция № 4. Основы алгоритмизации.....</b>	<b>32</b>
<b>Приложения.....</b>	<b>35</b>
<b>Литература.....</b>	<b>38</b>

## ВВЕДЕНИЕ

Конспект лекций соответствуют Федеральному государственному образовательному стандарту высшего образования (ФГОС ВО) и являются частью учебно-методического комплекса дисциплины «Информатика и программирование».

Лекция № 1 включает теоретический материал по позиционным системам счисления, применяемым для вычислений на компьютере, задачи на действия с числами в позиционных системах счисления. Лекция № 2 содержит теоретический материал по представлению информации различных видов в компьютере и задачи на вычисление информационного объема различных видов информации. Лекция № 3 включает теоретический материал по логическим основам работы компьютера и задачи на операции с логическими выражениями, функциями и логическими элементами. Лекция № 4 включает теоретический материал по основам алгоритмизации

Для облегчения восприятия теоретического материала по ходу практических занятий расположены примеры и решения с пояснениями.

Для закрепления теоретического материала конспект лекций содержит задания для индивидуальной работы.

Основные понятия в теоретической части каждого практического занятия выделены жирным шрифтом.

Приложение 1 содержит таблицу правил перевода чисел из одной системы счисления в другую.

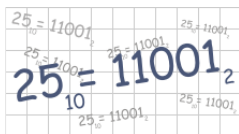
Приложение 2 содержит таблицу соответствия между системами счисления, правила выполнения арифметических операций в двоичной системе счисления.

Приложение 3 содержит таблицу основных и дополнительных логических функций.

В результате изучения дисциплины «Информатика и программирование» студент должен:

- уметь работать на компьютере;
- знать принципы применения современных информационных технологий в науке и предметной деятельности;
- уметь использовать информационные технологии при изучении естественнонаучных дисциплин и решении различных прикладных задач.

## ЛЕКЦИЯ № 1



### Тема: ПОЗИЦИОННЫЕ СИСТЕМЫ СЧИСЛЕНИЯ. ОПЕРАЦИИ НАД ЧИСЛАМИ В ПОЗИЦИОННЫХ СИСТЕМАХ СЧИСЛЕНИЯ

#### Цель лекции

Ознакомиться с основными позиционными системами счисления, применяемыми для вычислений в компьютере. Научиться переводить числа из одной системы счисления на другую, выполнять арифметические операции сложения, вычитания, умножения в позиционных системах счисления.

#### 1 Основные понятия

**Система счисления** – это совокупность правил для обозначения и наименования чисел.

Системы счисления делят на две большие группы: непозиционные и позиционные.

**Непозиционными** системами счисления называются системы счисления, в которых каждой цифре соответствует величина, не зависящая от её места в записи числа. Примером непозиционной системы счисления может служить римская система счисления, символы алфавита которой и обозначаемое ими количество представлено в таблице 1.

Таблица 1 – Римская система счисления

Римские цифры	I	V	X	L	C	D	M
Значение (обозначаемое количество)	1	5	10	50	100	500	1000

**Позиционными** системами счисления называются системы счисления, в которых вклад каждой цифры в величину числа зависит от её положения (позиции) в последовательности цифр, изображающей число. Примером может служить десятичная система счисления, считать в которой учат в школе. Значение каждой цифры в числе зависит от ее местоположения. Сравните два числа: 50 и 500. В первом числе пять стоит в разряде десятков, во втором числе – в разряде сотен.

Основными достоинствами любой позиционной системы счисления являются:

- во-первых, ограниченное количество символов для записи чисел;
- во-вторых, простота выполнения арифметических операций.

**Основанием** позиционной системы счисления ( $q$ ) называют количество символов, используемое для записи числа.

Основание системы счисления показывает, во сколько раз изменится количественное значение цифры при перемещении ее на соседнюю позицию.

При записи числа основание системы счисления указывают в виде нижнего индекса:  $54_{10}$ ,  $1011_2$ ,  $A1E_{16}$ .

В компьютере используют двоичную, десятичную, восьмеричную и шестнадцатеричную системы счисления.

От системы счисления будет зависеть:

- скорость вычислений;
- емкость памяти;
- сложность алгоритма выполнения арифметических и логических операций.

Двоичная система счисления используется для организации машинных операций по преобразованию информации:

$q = 2$ , алфавит: 0, 1.

Десятичная система счисления – для ввода и вывода информации:

$q = 10$ , алфавит: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

Восьмеричная и шестнадцатеричная системы счисления – для составления программ.

В восьмеричной системе счисления:

$q = 8$ , алфавит: 0, 1, 2, 3, 4, 5, 6, 7.

В шестнадцатеричной системе счисления:

$q = 16$ , алфавит: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.

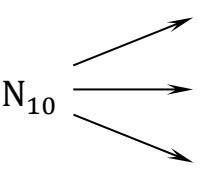
## 2 Правила перевода чисел из одной системы счисления в другую

### 2.1 Перевод чисел из десятичной системы счисления в двоичную, восьмеричную и шестнадцатеричную системы счисления

В таблице 2 приведены правила перевода целой и дробной частей чисел десятичной системы счисления в двоичную, восьмеричную и шестнадцатеричную системы счисления (приложение 1).

Таблица 2 – Перевод чисел десятичной системы счисления в двоичную, восьмеричную и шестнадцатеричную системы счисления

Направление перевода	Правила перевода
<div style="display: flex; align-items: center;"> <div style="margin-right: 20px;"> <math>N_{10}</math> <div style="display: flex; flex-direction: column; align-items: center;"> <div style="margin-bottom: 5px;">↗</div> <div style="margin-bottom: 5px;">→</div> <div style="margin-bottom: 5px;">↘</div> </div> </div> <div> <math>N_2</math>  <math>N_8</math>  <math>N_{16}</math> </div> </div>	<div style="display: flex; flex-direction: column;"> <div>деление на 2</div> <div>деление на 8</div> <div>деление на 16</div> </div>

Дробная часть числа	
$N_{10}$ 	$N_2$ умножение на 2
	$N_8$ умножение на 8
	$N_{16}$ умножение на 16

Поясним правила на примерах.

Для перевода числа из десятичной системы счисления в двоичную (восьмеричную, шестнадцатеричную) системы счисления необходимо:

- целую часть числа разделить на основание системы счисления, в которую производится перевод (в двоичную – на 2, в восьмеричную – на 8, в шестнадцатеричную – на 16). Деление целочисленное, с остатком. Деление продолжить до тех пор, пока делимое не станет меньше делителя (2, 8, 16). Затем выписать остатки, начиная с последнего. Таким образом, получим целую часть числа в новой системе счисления;

- дробную часть числа умножаем на основание системы счисления, в которую производится перевод (в двоичную – на 2, в восьмеричную – на 8, в шестнадцатеричную – на 16). Если при умножении получается целое число, то в следующем действии оно не учитывается, а умножается дробная часть и т.д. Умножение заканчивается тогда, когда дробная часть равна 0. Если этого не происходит, то умножение производить до третьего знака после запятой. В дробную часть переведенного числа выписываем полученные целые числа.

### Пример 1

Переведите число  $67,25_{10}$  из десятичной системы счисления в двоичную.

Перевод целой части	Перевод дробной части
$  \begin{array}{r}  \underline{67} \quad 2 \\  \underline{66} \quad \underline{33} \quad 2 \\  \underline{1} \quad \underline{32} \quad \underline{16} \quad 2 \\  \quad \underline{1} \quad \underline{16} \quad \underline{8} \quad 2 \\  \quad \quad \underline{0} \quad \underline{8} \quad \underline{4} \quad 2 \\  \quad \quad \quad \underline{0} \quad \underline{4} \quad \underline{2} \quad 2 \\  \quad \quad \quad \quad \underline{2} \quad 1 \\  \quad \quad \quad \quad \quad \underline{0}  \end{array}  $	$  \begin{array}{r l}  0, & 25 \\  \hline  & \underline{2} \\  0 & 50 \\  \hline  & \underline{2} \\  1 & 00  \end{array}  $

Ответ:  $67,25_{10} = 1000011,01_2$

### Пример 2

Переведите число  $71,25_{10}$  из десятичной системы счисления в восьмеричную.

### Пример 3

Перевод целой части	Перевод дробной части
$  \begin{array}{r}  \begin{array}{r l}  \_317 & 16 \\  304 & \_19 \\  \hline  13 & 16 \\  & \underline{\quad} \\  & 3  \end{array}  \end{array}  $	$  \begin{array}{r}  \downarrow \quad \begin{array}{r l}  0, & 25 \\  4 & 16 \\  \hline  & 00  \end{array}  \end{array}  $

## 2.2 Перевод чисел из двоичной, восьмеричной и шестнадцатеричной систем счисления в десятичную систему счисления

Таблица 3 – Перевод чисел из двоичной, восьмеричной и шестнадцатеричной систем счисления в десятичную систему счисления

Направление перевода	Правила перевода
$N_2 \rightarrow$ $N_8 \longrightarrow$ $N_{16} \nearrow$ $N_{10}$	<p>представить число в развернутой форме, например:</p> $1011_2 = 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 8 + 2 + 1 = 11_{10}$ $75_8 = 7 \cdot 8^1 + 5 \cdot 8^0 = 56 + 5 = 61_{10}$ $A3_{16} = A \cdot 16^1 + 3 \cdot 16^0 = 10 \cdot 16 + 3 = 163_{10}$



Поясним правила на примерах.

Для перевода числа из двоичной (восьмеричной, шестнадцатеричной) системы счисления в десятичную необходимо:

- над каждой цифрой числа, которое необходимо перевести, поставить номер разряда, на котором стоит цифра. Разряды выставляются в целой части числа справа налево (начиная с 0), в дробной части – слева направо (начиная с -1);
- умножаем каждую цифру переводимого числа на основание системы счисления (из которой происходит перевод), возведенное в степень, равную номеру разряда, на котором стоит соответствующая цифра. Затем складываем эти произведения;
- если число переводится из шестнадцатеричной системы счисления, то буква, при умножении, заменяется соответствующим этой букве значением в десятичной системе счисления (см. приложение 2).

#### **Пример 4**

Переведите число  $57,4_8$  из восьмеричной системы счисления в десятичную.

$$\begin{matrix} 1 & 0 & -1 \\ 57,4_8 = 5 \cdot 8^1 + 7 \cdot 8^0 + 4 \cdot 8^{-1} = 40 + 7 + \frac{4}{8} = 47\frac{1}{2} = 47,5_{10}. \end{matrix}$$

**Ответ:**  $57,4_8 = 47,5_{10}$

#### **Пример 5**

Переведите число  $A7,48_{16}$  из шестнадцатеричной системы счисления в десятичную.

$$\begin{matrix} 1 & 0 & -1 & -2 \\ A7,48_{16} = 10 \cdot 16^1 + 7 \cdot 16^0 + 4 \cdot 16^{-1} + 8 \cdot 16^{-2} = 160 + 7 + \frac{4}{16} + \frac{8}{256} = 167\frac{9}{32}_{10}. \end{matrix}$$

**Ответ:**  $A7,48_{16} = 167\frac{9}{32}_{10}$

### **2.3 Перевод чисел из двоичной системы счисления в восьмеричную и шестнадцатеричную системы счисления**

В таблице 4 приведены правила перевода чисел из двоичной системы счисления в восьмеричную и шестнадцатеричную системы счисления (приложение 1).

Таблица 4 – Перевод чисел из двоичной системы счисления в восьмеричную и шестнадцатеричную системы счисления

Направление перевода	Правила перевода
$N_2 \begin{matrix} \nearrow N_8 \\ \searrow N_{16} \end{matrix}$	замена триад цифрами восьмеричной системы счисления замена тетрад цифрами шестнадцатеричной системы счисления

Поясним правила на примерах.

Для перевода числа из двоичной системы счисления в восьмеричную (шестнадцатеричную) системы счисления необходимо:

- число разбить на тройки цифр – триады (для восьмеричной системы счисления) или на четверки цифр – тетрады (для шестнадцатеричной системы счисления). Разбиение начинать от запятой, разделяющей целую и дробную части, влево и вправо соответственно;
- если в отделенной последовательности количество цифр меньше трёх (четырёх), то недостающие цифры заменить нулями. Для целой части нули приписать перед числом, для дробной части – после числа;
- выписать соответственно каждой отделенной группе (триаде или тетраде) по таблице соответствия между системами счисления восьмеричную или шестнадцатеричную цифру (см. приложение 2).

### **Пример 6**

Переведите число  $1111101,01011_2$  из двоичной в восьмеричную и шестнадцатеричную системы счисления.

$$1111101,01011_2 = \begin{array}{ccccc} 001 & 111 & 101, & 010 & 110_2 \\ 1 & 7 & 5 & 2 & 6 \end{array} = 175,26_8$$

$$1111101,01011_2 = \begin{array}{cccc} 0111 & 1101, & 0101 & 1000_2 \\ 7 & D & 5 & 8 \end{array} = 7D,58_{16}$$

**Ответ:**  $1111101,01011_2 = 175,26_8$ ;  $1111101,01011_2 = 7D,58_{16}$

## **2.4 Перевод чисел из восьмеричной и шестнадцатеричной систем счисления в двоичную систему счисления**

В таблице 5 приведены правила перевода чисел из двоичной системы счисления в восьмеричную и шестнадцатеричную системы счисления (приложение 1).

Таблица 5 – Перевод чисел из двоичной системы счисления в восьмеричную и шестнадцатеричную системы счисления

Направление перевода	Правила перевода
$N_8 \rightarrow$ $N_{16} \nearrow$ <div style="text-align: right;"><math>N_2</math></div>	<p>замена восьмеричных цифр триадами</p> <p>замена шестнадцатеричных цифр тетрадами</p>

Поясним правила на примерах.

Для перевода числа из восьмеричной (шестнадцатеричной) системы счисления в двоичную систему счисления необходимо:

- выписать соответственно каждой цифре восьмеричного числа тройку цифр – триаду по таблице соответствия между системами счисления триаду (см. приложение 2). Аналогично поступаем и с шестнадцатеричным числом, но выписываем четверку цифр – тетраду;
- в записи полученного двоичного числа нули в начале целой части и в конце дробной части опустить.

### **Пример 7**

Переведите числа  $567,561_8$  и  $4FE,AD_{16}$  в двоичную систему счисления.

$$567,561_8 = \begin{array}{ccc|ccc} 5 & 6 & 7, & 5 & 6 & 1 \\ 101 & 110 & 111 & 101 & 110 & 001 \end{array} = 101110111,101110001_2$$

$$4FE,AD_{16} = \begin{array}{ccc|ccc} 4 & F & E, & A & D \\ 0100 & 1111 & 1110 & 1010 & 1101 \end{array} = 10011111110,10101101_2$$

**Ответ:**  $567,561_8 = 101110111,101110001_2$ ;  
 $4FE,AD_{16} = 10011111110,10101101_2$ .

## **2.5 Перевод чисел из восьмеричной системы счисления в шестнадцатеричную систему счисления (или из шестнадцатеричной системы счисления в восьмеричную систему счисления)**

Для перевода числа из восьмеричной системы счисления в шестнадцатеричную (из шестнадцатеричной в восьмеричную) необходимо:

- осуществить перевод восьмеричного (или шестнадцатеричного) числа в двоичную систему счисления (см. п.2.2.4);
- осуществить перевод числа из двоичной системы счисления в шестнадцатеричную (или восьмеричную) систему счисления (см. п.2.2.3).

### **Пример 8**

Переведите число  $4FE,AD_{16}$  в восьмеричную систему счисления, а число  $567,561_8$  в шестнадцатеричную систему счисления.

$$4FE,AD_{16} = 0100\ 1111\ 1110,1010\ 1101_2 = 10011111110,10101101_2 \\ = 010\ 011\ 111\ 110,101\ 011\ 010_2 = 2376,532_8$$

$$567,561_8 = 101\ 110\ 111,101\ 110\ 001_2 = 101110111,101110001_2 \\ = 0001\ 0111\ 0111,1001\ 1001_2 = 177,B9_{16}$$

### 3 Арифметические операции в позиционных системах счисления

Арифметические операции в двоичной системе счисления опираются на несколько правил (см. приложение 2).

Арифметические операции в восьмеричной и шестнадцатеричной системе счисления сходны с аналогичными операциями в десятичной системе счисления.

Все операции производятся столбиком. Рассмотрим несколько примеров.

#### Пример 9

Выполните сложение, вычитание и умножение двоичных чисел  $1100,11_2$  и  $101,10_2$ .

сложение	вычитание	умножение
$\begin{array}{r} 1100,11 \\ + 101,10 \\ \hline 10010,01 \end{array}$	$\begin{array}{r} 1100,11 \\ - 101,10 \\ \hline 111,01 \end{array}$	$\begin{array}{r} 1100,11 \\ \times 101,1 \\ \hline 110011 \\ + 110011 \\ 110011 \\ \hline 10000110,001 \end{array}$

#### Пример 10

Выполните сложение, вычитание и умножение чисел в восьмеричной системе счисления:  $215,4_8$  и  $73,6_8$ .

Необходимо помнить:

- если при сложении или умножении двух цифр в разряде получилось число большее восьми, то полученное число надо разделить на 8 (целочисленное деление). Остаток от деления оставить в текущем разряде, а результат деления прибавить к старшему разряду;

- если при вычитании двух цифр уменьшаемая цифра меньше вычитаемой, то занимаем в старшем разряде 1 (для текущего разряда это будет 8). 8 прибавляем к уменьшаемому и из полученной суммы отнимаем вычитаемое.

сложение	вычитание	умножение
$\begin{array}{r} 111 \\ 215,4 \\ + 73,6 \\ \hline 311,2 \end{array}$ <div style="margin-left: 100px;"> <math>4+6=10=8+2</math>  <math>5+3+1=9=8+1</math>  <math>1+7+1=9=8+1</math>  <math>2+1=3</math> </div>	$\begin{array}{r} 111 \\ 215,4 \\ - 73,6 \\ \hline 121,6 \end{array}$ <div style="margin-left: 100px;"> <math>8+4-6=6</math>  <math>4-3=1</math>  <math>1+8-7=2</math> </div>	$\begin{array}{r} 215,4 \\ \times 73,6 \\ \hline 15210 \\ + 6504 \\ \hline 17364 \\ \hline 20406,50 \end{array}$

### Пример 11

Выполните сложение и вычитание чисел в шестнадцатеричной системе счисления:  $8D,8_{16}$  и  $3B,C_{16}$ .

Необходимо помнить:

- если при сложении или умножении двух цифр в разряде получилось число большее восьми, то полученное число надо разделить на 16 (целочисленное деление). Остаток от деления оставить в текущем разряде, а результат деления прибавить к старшему разряду;
- если при вычитании двух цифр уменьшаемая цифра меньше вычитаемой, то занимаем в старшем разряде 1 (для текущего разряда это будет 16). 16 прибавляем к уменьшаемому и из полученной суммы отнимаем вычитаемое.

сложение	вычитание	умножение
$\begin{array}{r} 11 \\ 8D,8 \\ + 3B,C \\ \hline C9,4 \end{array}$ <div style="margin-left: 100px;"> <math>8+12=20=16+4</math>  <math>13+11+1=25=16+9</math>  <math>8+3+1=12=C</math> </div>	$\begin{array}{r} 11 \\ 8D,8 \\ - 3B,C \\ \hline 51,C \end{array}$ <div style="margin-left: 100px;"> <math>8+16=24-12=12=C</math>  <math>12-11=1</math>  <math>8-3=5</math> </div>	$\begin{array}{r} 8D,8 \\ \times 3B,C \\ \hline 6A20 \\ + 6148 \\ \hline 1A88 \\ \hline 2106,A0 \end{array}$

## ЛЕКЦИЯ № 2



### Тема: КОЛИЧЕСТВО ИНФОРМАЦИИ

#### Цель лекции

Изучить основные формы представления информации, методы и правила вычисления объема информации. Научиться вычислять информационный объем различных видов информации.

#### 1 Единицы измерения и методы измерения количества информации

**Информация** – сведения об объектах и явлениях окружающей среды, их параметрах, свойствах и состоянии, которые уменьшают имеющуюся о них степень неопределенности, неполноты знаний.

В компьютере для представления информации используется двоичное кодирование, так как удалось создать надежно работающие технические устройства, которые могут со стопроцентной надежностью сохранять и распознавать не более двух различных состояний: наличие или отсутствие электрического импульса.

**Код** – система условных знаков (символов) для представления различной информации.

Информация в компьютере представлена в двоичном коде, состоящем из двух цифр – 0 и 1.

При записи двоичной цифры реализуется выбор одного из двух возможных состояний (одной из двух цифр) и, следовательно, она несет объем информации, равный 1 биту (binary digit – двоичный разряд). Таким образом, за минимальную единицу измерения количества информации принят 1 бит.

Для измерения информации используются и более крупные единицы:

1 байт = 8 бит =  $2^3$  бит;

1 Кбайт (килобайт) =  $2^{10}$  байт = 1024 байт;

1 Мбайт (мегабайт) =  $2^{10}$  Кбайт = 1024 Кбайт;

1 Гбайт (гигабайт) =  $2^{10}$  Мбайт = 1024 Мбайт;

1 Тбайт (терабайт) =  $2^{10}$  Гбайт = 1024 Гбайт.

#### 1.1 Алфавитный подход к измерению количества информации

**Алфавитный подход** к измерению информации позволяет определить количество информации, заключенной в тексте. Алфавитный подход является объективным, т.е. он не зависит от субъекта (человека), воспринимающего текст.

Множество символов, используемых при записи текста, называется *алфавитом*. Полное количество символов в алфавите называется *мощностью (размером) алфавита*.

Если допустить, что все символы алфавита встречаются в тексте с одинаковой частотой (равновероятно), то количество информации, которое несет каждый символ, вычисляется по формуле

$$i = \log_2 N,$$

где  $N$  – мощность алфавита.

Следовательно, в 2-символьном алфавите каждый символ «весит» 1 бит ( $\log_2 2 = 1$ ); в 4-символьном алфавите каждый символ несет 2 бита информации ( $\log_2 4 = 2$ ); в 8-символьном – 3 бита ( $\log_2 8 = 3$ ) и т.д.

Один символ из алфавита мощностью 256 несет в тексте 8 бит информации. Такое количество информации называется байт. Алфавит из 256 символов используется для представления текстов на компьютере.

Если весь текст состоит из  $K$  символов, то при алфавитном подходе размер содержащейся в нем информации равен:

$$I = K \cdot i,$$

где  $i$  – информационный вес одного символа в используемом алфавите.

### **Пример 1**

В зрительном зале две прямоугольные области зрительных кресел: одна 10 на 5, а другая 4 на 8. Какое минимальное количество бит потребуется для кодирования каждого места в автоматизированной системе?

**Решение:**

$$N = 10 \cdot 5 + 4 \cdot 8 = 82;$$

$$64(2^6) < 82 < 128(2^7);$$

$$N' = 128 = 2^7;$$

$$i = \log_2 N' = \log_2 2^7 = 7 \text{ бит.}$$

**Ответ:** 7 бит.

### **Пример 2**

В велокроссе участвуют 230 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 20 велосипедистов?

**Решение:**

$$N = 230;$$

$$128(2^7) < 230 < 256(2^8);$$

$$N' = 256 = 2^8;$$

$$i = \log_2 N' = \log_2 2^8 = 8 \text{ бит} - \text{это минимальное количество бит для записи номера спортсмена.}$$

Поскольку была записана информация о 20 спортсменах, объем записанного сообщения составил  $20 \times 8 = 160$  бит.

**Ответ:** 160 бит.

### **Пример 3**

Сколько существует различных последовательностей из символов «+» и «-», длиной ровно в шесть символов?

**Решение:**

В данном случае алфавит состоит из двух элементов, потому информационный объем одного символа – 1 бит. 6 бит позволяют закодировать множество из  $2^6 = 64$  элементов.

**Ответ:** 64.

### **Пример 4**

Световое табло состоит из лампочек. Каждая лампочка может находиться в одном из трех состояний («включено», «выключено» или «мигает»). Какое наименьшее количество лампочек должно находиться на табло, чтобы с его помощью можно было передать 18 различных сигналов?

**Решение:**

С помощью  $n$  лампочек, каждая из которых может находиться в трех состояниях, можно закодировать  $3^n$  сигналов.  $3^2 < 18 < 3^3$ , поэтому двух лампочек недостаточно, а трех хватит.

**Ответ:** 3.

### **Пример 5**

Автоматическое устройство осуществило перекодировку информационного сообщения на русском языке, первоначально записанного в 16-битном коде Unicode, в 8-битную кодировку КОИ-8. При этом информационное сообщение уменьшилось на 480 бит. Какова длина сообщения в символах?

**Решение:**

После перекодировки из 16-битного кода в 8-битный каждый символ сообщения стал занимать на 8 бит меньше, а все сообщение уменьшилось на 480 бит, следовательно, сообщение состояло из  $480:8 = 60$  символов.

**Ответ:** 60.

### **Пример 6**

Книга, набранная с помощью компьютера, содержит 150 страниц. На каждой странице – 40 строк, в каждой строке – 60 символов. Каков объем информации в книге?

**Решение:**

Мощность компьютерного алфавита равна 256. Значит, один символ несет  $256 = 2^8$  – 1 байт информации. Значит, страница содержит



$40 \cdot 60 = 2400$  байт информации. Объем всей информации в книге (в разных единицах):

$$2400 \cdot 150 = 360000 \text{ байт};$$

$$\frac{360000}{1024} = 351,5625 \text{ Кбайт};$$

$$\frac{351,5625}{1024} = 0,34332275 \text{ Мбайт.}$$

**Ответ:** 0,34332275 Мбайт.

## 1.2 Вероятностный подход к измерению количества информации

Рассмотрим способ вычисления информационного объема сообщения с использованием вероятностного подхода на конкретном примере.

### *Пример 7*

В пруду живут 32500 карасей, 5000 щук и 12500 пескарей. Вычислить объем сообщения «рыбак поймал пескаря».

**Решение:**

Всего в пруду живет 50000 рыб ( $32500 + 5000 + 12500 = 50000$ ). Вероятность попадания на удочку каждого из вида рыб равна его доле в общем количестве. Отсюда:

$$p = \frac{12500}{50000} = 0,25.$$

Качественную связь между вероятностью события и количеством информации в сообщении об этом событии можно выразить так: чем меньше вероятность некоторого события, тем больше информации содержит сообщение об этом событии.

Количественная зависимость между вероятностью события ( $p$ ) и количеством информации в сообщении о нем ( $I$ ) выражается формулой

$$I = \log_2(1/p).$$

В рассматриваемом примере количество информации в сообщении «рыбак поймал пескаря» вычисляется по формуле

$$I = \log_2(1/0,25) = 2 \text{ бита.}$$

**Ответ:** 2 бита.

**Пример 8**

В группе 16 студентов – 14 юношей и 2 девушки. Найти объем сообщения «из группы выбрана одна девушка».

**Решение:**

Вероятность выбора одной девушки из группы студентов равна:

$$\frac{2}{16} = \frac{1}{8}.$$

Информационный объем сообщения равен:

$$\log_2 \left( \frac{1}{1/8} \right) = \log_2 2^3 = 3 \text{ бита.}$$

**Ответ:** 3 бита.

### 1.3 Представление и измерение звуковой информации

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

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

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

Человек может воспринимать звук в огромном диапазоне интенсивностей, в котором максимальная интенсивность больше минимальной в  $10^{14}$  раз (в сто тысяч миллиардов раз). Для измерения громкости звука применяется специальная единица «децибел».

**Временная дискретизация звука.** Для того чтобы компьютер мог обрабатывать звук, непрерывный звуковой сигнал должен быть преобразован в цифровую дискретную форму с помощью временной дискретизации. Непрерывная звуковая волна разбивается на отдельные маленькие временные участки, для каждого такого участка устанавливается определенная величина интенсивности звука. Таким образом, непрерывная зависимость громкости звука от времени  $A(t)$  заменяется на дискретную последовательность уровней громкости. На графике это выглядит как замена гладкой кривой на последовательность «ступенек» (рисунок 2.1).

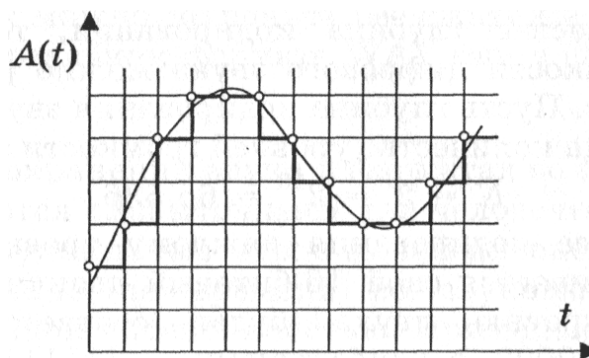


Рисунок 2.1 – Временная дискретизация звука

**Частота дискретизации  $\nu$ .** Для записи аналогового звука и его преобразования в цифровую форму используется микрофон, подключенный к звуковой плате. Качество полученного цифрового звука зависит от количества измерений уровня громкости звука в единицу времени, т.е. частоты дискретизации. Чем большее количество измерений производится за 1 секунду, тем больше частота дискретизации.

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

Частота дискретизации звука может лежать в диапазоне от 8 000 до 48 000 измерений громкости звука за одну секунду.

Одно измерение в секунду соответствует частоте 1 Гц, 1 000 измерений в секунду – 1 кГц.

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

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

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

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

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

Различные звуковые карты сегодня могут обеспечить 8- или 16-разрядную глубину кодирования звука.

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

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

$$16 \text{ бит} \cdot 24000 \cdot 2 \text{ (канала)} = 768000 \text{ бит} = 96000 \text{ байт} = 93,75 \text{ Кбайт}.$$

Качество звука в дискретной форме может быть очень плохим (при 8 битах и 5,5 кГц) и очень высоким (при 16 битах и 44,1 кГц).

Рассмотрим примеры вычисления информационного объема звукового файла.

**Пример 9**

Оцените объем моно аудиофайла длительностью звучания 10 с при частоте дискретизации 22,05 кГц и разрешении 8 бит. Ответ запишите в Кбайтах.

**Решение:**

- 1) Для расчета объема моно аудиофайла используем формулу:

$$N = \nu \cdot t \cdot R,$$

где  $\nu$  – частота дискретизации, Гц;

$R$  – глубина кодирования звука, бит;

$t$  – время звучания звукового файла, с.

- 2) Вычислим объем файла:

$$N = 22,05 \cdot 1000 \cdot 10 \cdot 8 = 1\,764\,000 \text{ бит} = 215,33 \text{ Кбайт.}$$

**Ответ:** 215,33 Кбайт.

**Пример 10**

Подсчитать время звучания звукового файла объемом 3,5 Мбайт, содержащего стерео запись с частотой дискретизации 44 100 Гц и разрядностью кода 16 бит на 1 измерение.

**Решение:**

- 1) Для расчета объема стерео аудиофайла используем формулу

$$N = 2 \cdot \nu \cdot t \cdot R,$$

где  $\nu$  – частота дискретизации, Гц;

$R$  – глубина кодирования звука, бит;

$t$  – время звучания звукового файла, с.

- 2) Вычислим время звучания звукового файла:

$$t = \frac{N}{2 \cdot \nu \cdot R} = \frac{3,5 \cdot 1024}{2 \cdot 44100 \cdot 16} = 20,805 \text{ секунд.}$$

**Ответ:** 20,805 секунд.

**2 Скорость передачи информации**

Основной характеристикой процесса передачи информации является *скорость передачи информации*.

**Скорость передачи данных** – количество бит информации, передаваемой за единицу времени. Измеряется в битах в секунду (бит/с) и кратных единицах – Кбит/с и Мбит/с.

Объем переданной информации  $I$  вычисляется по формуле

$$I = q \cdot t,$$

где  $q$  – пропускная способность канала (бит/с),

$t$  – время передачи (с).

**Пример 11**

Скорость передачи данных через ADSL-соединение равна 128 000 бит/с. Передача файла через это соединение заняла 2 минуты. Определите размер файла в килобайтах.

**Решение:**

1)  $128000 \cdot 120 = 1536000$  (бит) – количество бит, переданных за 2 минуты.

2)  $\frac{1536000}{8 \cdot 1024} = 1875$  (Кбайт).

**Ответ:** 1 875 Килобайт.

**Пример 12**

Скорость передачи данных через ADSL-соединение равна 128 000 бит/с. Через данное соединение передают файл размером 625 Кбайт. Определите время передачи файла в секундах.

**Решение:**

1)  $625 \cdot 1024 \cdot 8 = 512\,000$  (бит) – размер файла в битах;

2)  $\frac{512000}{12800} = 40$  (с).

**Ответ:** 40 секунд.

### 3 Обработка графической информации

Для успешного решения задач этой темы необходимо понимать взаимосвязь информационного объема растрового изображения, его пространственного и цветового разрешения, а также принципы кодирования цвета элементов раstra.

Графическая информация в компьютере представляется в трех форматах: растровая графика, векторная графика и фрактальная графика. С точки зрения вычисления объема информации, наибольший интерес представляет растровая графика. Рассмотрим этот графический формат более подробно.

**Растровое изображение** на экране монитора формируется из точек (пикселей).

**Растр** – прямоугольная сетка пикселей на экране. Объем растрового изображения зависит от глубины цвета.

**Глубина цвета** – количество бит, отводимое под пиксель.

Число цветов, воспроизводимых на экране монитора  $K$ , и глубину цвета  $N$ , можно найти по формуле

$$K = 2^N.$$

Таблица 3 – Расчет глубины цвета

Глубина цвета	Количество цветов	1 пиксель
1	2 (черный, белый)	1 бит
3	8	3 бита
4	16	4 бита
8	256	8 бит
16	65 536 (high color)	16 бит
24	16 777 216 (true color)	24 бита
32	4 294 967 296 (true color)	32 бита

Код цвета пикселя содержит информацию о доле каждого базового цвета. Чем больше глубина цвета, тем большее количество бит потребуется для кодирования одного пикселя и тем большее количество цветов можно использовать в изображении.

Для успешного решения задач этой темы необходимо понимать взаимосвязь информационного объема растрового изображения, его пространственного и цветового разрешения, а также принципы кодирования цвета элементов раstra.

### **Пример 13**

Для хранения растрового изображения размером  $64 \times 64$  пикселя отвели 1,5 килобайта памяти. Каково максимально возможное число цветов в палитре изображения?

#### **Решение:**

Вычислим, сколько бит отводится для хранения одного пикселя, учитывая, что в одном килобайте  $1024 = 2^{10}$  байта, а в одном байте  $8 = 2^3$  бита. Для этого информационный объем изображения, выраженный в битах, разделим на количество пикселей:

$$1,5 \cdot 1024 \cdot \frac{8}{64 \cdot 64} = 1,5 \cdot 2^{10} \cdot \frac{2^3}{2^6 \cdot 2^6} = 1,5 \cdot 2^{10+3-6-6} = 1,5 \cdot 2 = 3 \text{ (бит)}.$$

Итак, на один пиксель приходится три бита. Три бита позволяют закодировать максимум  $2^3 = 8$  различных значений. Поэтому максимально возможное число цветов в палитре изображения равно 8.

**Ответ:** 8.

### **Пример 14**

Укажите минимальный объем памяти (в килобайтах), достаточный для хранения любого растрового изображения размером  $256 \times 256$  пикселей, если известно, что в изображении используется палитра из 216 цветов. Саму палитру хранить не нужно.

**Решение:**

Найдем минимальный объем памяти, необходимый для хранения одного пикселя. В изображении используется палитра из  $2^{16}$  цветов, следовательно, одному пикселю может быть сопоставлен любой из  $2^{16}$  возможных номеров цвета в палитре. Поэтому минимальный объем памяти для одного пикселя будет равен  $\log_2 2^{16} = 16$  бит. Минимальный объем памяти, достаточный для хранения всего изображения будет равен

$$16 \cdot 256 \cdot 256 = 2^4 \cdot 2^8 \cdot 2^8 = 2^{20} \text{бит} = 2^{17} \text{байт} = 2^7 \text{Кбайт} = 128 \text{Кбайт}.$$

**Ответ:** 128 Кбайт.

**Пример 15**

Для кодирования цвета фона страницы Интернет используется атрибут `bgcolor="#XXXXXX"`, где в кавычках задаются шестнадцатеричные значения интенсивности цветовых компонент в 24-битной RGB-модели. Какой цвет будет у страницы, заданной тэгом `<body bgcolor="#FFFFFF">`?

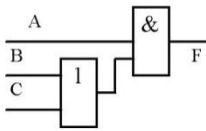
**Решение:**

В 24-битной системе кодирования цветов RGB, используемой для мониторов каждая из трех цветовых составляющих (красная, зеленая, синяя) может принимать значение от 0 до 255 и кодируется одним байтом (две шестнадцатеричные цифры). 0 соответствует полному отсутствию данной цветовой составляющей в пикселе, 255 ( $FFFFFF_{16}$ ) – максимальной яркости данного цвета. Максимальная яркость всех трех цветовых сигналов  $FFFFFF_{16}$  в RGB обеспечивает белый цвет пикселя. Минимальная (все цвета выключены)  $000000_{16}$  – черный.

**Ответ:** белый.



## ЛЕКЦИЯ № 3



### Тема: ЛОГИЧЕСКИЕ ОСНОВЫ РАБОТЫ КОМПЬЮТЕРА

#### Цель лекции

Ознакомиться с понятием высказывание и основными логическими функциями. Научиться составлять простые и составные высказывания и определять их истинность, считать таблицы истинности для различных логических функций, упрощать логические выражения, строить логические схемы.

#### Основные понятия алгебры логики

Основу компьютера составляют элементарные логические схемы. Работа этих схем основана на законах и правилах алгебры логики, которая оперирует двумя понятиями: истинности и ложности высказывания.

**Алгебра логики (Булева алгебра)** – это раздел математики, изучающий высказывания, рассматриваемые со стороны их логических значений (истинности или ложности) и логических операций над ними.

**Логическое высказывание** – это любое повествовательное предложение, в отношении которого можно однозначно сказать, истинно оно или ложно.

Высказывания бывают простыми и сложными.

**Простое высказывание (логическая переменная)** содержит только одну простую мысль.

**Сложное высказывание (логическая функция)** содержит несколько простых мыслей, соединенных между собой с помощью логических операций.

Логические переменные обычно обозначаются буквами латинского алфавита: **A, B, C ...**

**Основными логическими операциями** являются:

- отрицание (инверсия);
- умножение (конъюнкция);
- сложение (дизъюнкция);

**Дополнительными логическими операциями** являются:

- следование (импликация);
- равнозначность (эквиваленция).

Таблица с основными и дополнительными логическими операциями находится в приложении 3.

Значение сложного высказывания (логической функции) можно определить с помощью таблицы истинности.

**Таблица истинности** – это таблица, в которой перечислены все возможные значения входящих логических переменных и соответствующие им значения функции.

Порядок выполнения логических операций:

- 1) операция в скобках;
- 2)  $\bar{A}$ ,  $\neg A$  – логическое отрицание;
- 3)  $\times$ ,  $\&$ ,  $\wedge$  – логическое умножение;
- 4)  $+$ ,  $\vee$  – логическое сложение;
- 5)  $\rightarrow$ ,  $\Rightarrow$  – импликация;
- 6)  $=$ ,  $\Leftrightarrow$ ,  $\leftrightarrow$  – эквиваленция.

### Тождества и законы алгебры логики

Для упрощения сложных логических выражений применяются тождества и законы алгебры логики.

#### Тождества алгебры логики

Тождества логического сложения	Тождества логического умножения
$A \vee 0 = A$	$A \wedge 0 = 0$
$A \vee 1 = 1$	$A \wedge 1 = A$
$A \vee A = A$	$A \wedge A = A$
$A \vee \bar{A} = 1$	$A \wedge \bar{A} = 0$
$\bar{\bar{A}} = A$ – (двойное отрицание)	

#### Законы алгебры логики

Переместительный закон	$A \vee B = B \vee A$ $A \wedge B = B \wedge A$
Сочетательный закон	$(A \vee B) \vee C = A \vee (B \vee C)$ $(A \wedge B) \wedge C = A \wedge (B \wedge C)$
Распределительный закон	$(A \vee B) \wedge C = A \wedge C \vee B \wedge C$ $A \wedge B \vee C = (A \vee C) \wedge (B \vee C)$
Законы де Моргана	$\overline{A \vee B} = \bar{A} \wedge \bar{B}$ $\overline{A \wedge B} = \bar{A} \vee \bar{B}$ $A \rightarrow B = \bar{B} \rightarrow \bar{A} = \bar{A} \vee B$ $A \Leftrightarrow B = A \wedge B \vee \bar{A} \wedge \bar{B} = (\bar{A} \vee B) \wedge (A \vee \bar{B})$

## Логические элементы

Средством обработки двоичных сигналов в ЭВМ являются логические элементы.

Для реализации любых логических операций над двоичными сигналами достаточно элементов трех типов – элементов, реализующих три основные логические операции: **И**, **ИЛИ**, **НЕ**.

**Логические элементы (вентили)** – это электронные схемы с одним или несколькими входами и одним выходом, через которые проходят электрические сигналы, представляющие цифры 0 и 1.

1. Элемент **НЕ (инвертор)**. У инвертора один вход и один выход. Сигнал на выходе появится, если на входе его нет, и наоборот. Инвертор реализует логическую операцию **инверсия (логическое отрицание)**.

Функция	Таблица истинности	Логический элемент						
$F = \bar{A}$	<table><tr><td>A</td><td>F</td></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table>	A	F	0	1	1	0	
A	F							
0	1							
1	0							

2. Элемент **И (конъюнктор)**. Элемент **И** имеет не менее двух входов и один выход. **A** и **B** – входные сигналы, **F** – выходной сигнал. Конъюнктор реализует логическую операцию **конъюнкция (логическое умножение)**.

Функция	Таблица истинности	Логический элемент															
$F = A \wedge B$	<table border="1"> <tr> <th>A</th><th>B</th><th>F</th></tr> <tr> <td>0</td><td>0</td><td>0</td></tr> <tr> <td>0</td><td>1</td><td>0</td></tr> <tr> <td>1</td><td>0</td><td>0</td></tr> <tr> <td>1</td><td>1</td><td>1</td></tr> </table>	A	B	F	0	0	0	0	1	0	1	0	0	1	1	1	
A	B	F															
0	0	0															
0	1	0															
1	0	0															
1	1	1															

3. Элемент **ИЛИ (дизъюнктор)**. Элемент **ИЛИ** имеет не менее двух входов и один выход. **A** и **B** – входные сигналы, **F** – выходной сигнал. Дизъюнктор реализует логическую операцию **дизъюнкция (логическое сложение)**.

Функция	Таблица истинности	Логический элемент															
$F = A \vee B$	<table border="1"> <tr> <th>A</th><th>B</th><th>F</th></tr> <tr> <td>0</td><td>0</td><td>0</td></tr> <tr> <td>0</td><td>1</td><td>1</td></tr> <tr> <td>1</td><td>0</td><td>1</td></tr> <tr> <td>1</td><td>1</td><td>1</td></tr> </table>	A	B	F	0	0	0	0	1	1	1	0	1	1	1	1	
A	B	F															
0	0	0															
0	1	1															
1	0	1															
1	1	1															

С помощью логических элементов **НЕ**, **И**, **ИЛИ** можно реализовать (собрать как из конструктора) типовые функциональные узлы (блоки) ЭВМ:

- триггеры;
- сумматоры;
- шифраторы;
- регистры;
- счетчики;
- дешифраторы.

Чтобы понять, как работает интересующее нас устройство, необходимо понять логику его работы, т.е. найти соответствие между входными и выходными сигналами, для этого:

1. Составить таблицу истинности.
2. По таблице записать логическую функцию.
3. Построить логическую схему.

### **Пример 16**

Постройте таблицу истинности для логической функции  $F = \bar{X} \vee X \vee Y$ .

**Решение:**

1) Подсчитаем количество переменных в функции. В нашей функции две переменные –  $X$  и  $Y$ .

2) Построим таблицу для двух переменных и впишем все возможные сочетания значений двух переменных:

$X$	$Y$
0	0
0	1
1	0
1	1

3) Расставим в логической функции порядок выполнения операций.

$$\overset{1}{\bar{X}} \vee \overset{2}{X} \vee \overset{3}{Y}.$$

4) Согласно полученному порядку впишем операции в таблицу истинности и подсчитаем значения. Значения логических операций выписываем из таблицы в приложении 3.

$X$	$Y$	$\bar{X}$	$\bar{X} \vee X$	$\bar{X} \vee X \vee Y$
0	0	1	1	1
0	1	1	1	1
1	0	0	1	1
1	1	0	1	1

5) Согласно полученному результату делаем вывод.

**Вывод:** функция тождественно истинна.

Так как в последнем столбце значения функции – все единицы, то функция тождественно истинна. Таким образом, функция принимает значение истина (1) при любых значениях входящих в нее переменных  $X$  и  $Y$ .

Если в последнем столбце будут все нули, то функция тождественно ложна. Таким образом, функция принимает значение ложь (0) при любых значениях входящих в нее переменных  $X$  и  $Y$ .

Если в последнем столбце значения функции – набор нулей и единиц, то функция не является ни тождественно ложной, ни тождественно истинной. В таком случае выписываются значения входящих в функцию переменных, при которых функция принимает значения истина (1).

### **Пример 17**

Используя логические законы и тождества, упростите логическое выражение:  $A \wedge (A \vee B) \wedge (B \wedge \bar{B})$ .

**Решение:**

1) Первую скобку раскроем по распределительному закону, во второй скобке воспользуемся тождеством логического умножения, получим:

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

Можно и далее преобразовывать выражение в скобках. Но согласно тождеству  $A \wedge 0 = 0$ , можно заключить, что

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

### **Построение логических схем по заданной таблице истинности функции**

1. Выписывается таблица истинности функции.
2. По данной таблице определяется логическая функция (формула) с помощью следующего метода, называемого дизъюнктивная совершенная нормальная форма (ДСНФ):
  - 2.1. В заданной таблице выбираются наборы переменных, при которых значение функции равно 1.
  - 2.2. Для каждого такого набора записываются конъюнкции ( $\wedge$ ) всех входных переменных, имеющие значение 1. При этом те переменные, которые имеют значение 0, записываются с отрицанием.
  - 2.3. Все полученные конъюнкции объединяются знаком дизъюнкции ( $\vee$ ). Это и будет искомая логическая функция, которую можно будет упростить (минимизировать) по законам Булевой алгебры.
3. По упрощенной логической функции строится логическая схема.

### **Пример 18**

По заданной таблице истинности записать логическую функцию, упростить её и построить логическую схему.

$X$	$Y$	$F$
0	0	1
0	1	1
1	0	1
1	1	0

**Решение:**

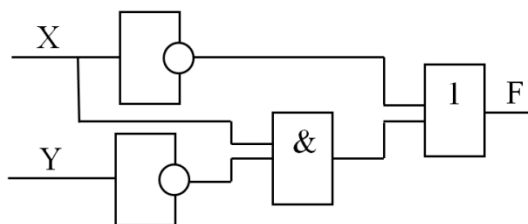
1. Запишем конъюнкцию для каждой строки, где значение функции равно 1 (первая, вторая и третья строки). Переменные, значения которых равны 0, запишем с отрицанием.

2. Объединив полученные конъюнкции дизъюнкцией, получим следующую логическую функцию.

$$F = (\bar{X} \wedge \bar{Y}) \vee (\bar{X} \wedge Y) \vee (X \wedge \bar{Y}).$$

3. Упростим:  $F = (\bar{X} \wedge \bar{Y}) \vee (\bar{X} \wedge Y) \vee (X \wedge \bar{Y}) = \bar{X} \vee (X \wedge \bar{Y})$ .

4. По полученной функции построим логическую схему:

**Пример 19**

Составить логическую схему, работа которой задана таблицей истинности:

$A$	$B$	$C$	$F$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

**Решение:**

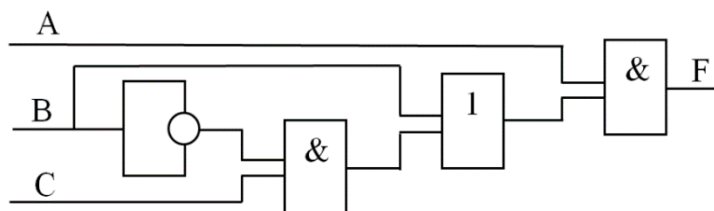
1. Составим логическую функцию по данной таблице:

$$F = (A \wedge \bar{B} \wedge C) \vee (A \wedge B \wedge \bar{C}) \vee (A \wedge B \wedge C).$$

2. Упростим полученную функцию:

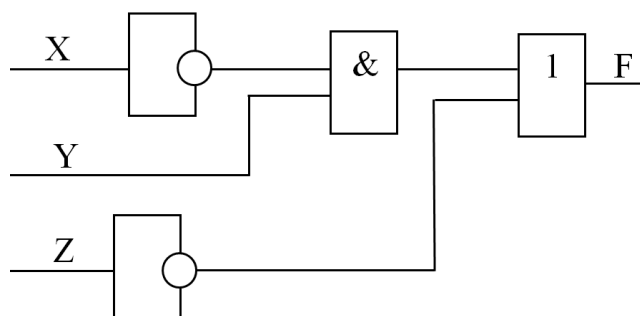
$$F = (A \wedge \bar{B} \wedge C) \vee (A \wedge B \wedge \bar{C}) \vee (A \wedge B \wedge C) = (A \wedge \bar{B} \wedge C) \vee (A \wedge B \wedge (\bar{C} \vee C)) = A \wedge \bar{B} \wedge C \vee A \wedge B = A \wedge (\bar{B} \wedge C \vee B).$$

3. По полученной функции построим логическую схему:



### Пример 20

По заданной логической схеме запишите логическую функцию и постройте таблицу истинности:



### Решение:

1. Для записи функции необходимо записать значения для каждого элемента схемы:

- I.  $\bar{X}$
- II.  $\bar{X} \wedge Y$
- III.  $\bar{Z}$
- IV.  $(\bar{X} \wedge Y) \vee \bar{Z}$

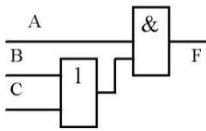
2. Следовательно, получится функция:

$$F = (\bar{X} \wedge Y) \vee \bar{Z}.$$

3. По полученной функции построим таблицу истинности:

$X$	$Y$	$Z$	$\bar{X}$	$\bar{X} \wedge Y$	$\bar{Z}$	$(\bar{X} \wedge Y) \vee \bar{Z}$
0	0	0	1	0	1	0
0	0	1	1	0	0	0
0	1	0	1	1	1	0
0	1	1	1	1	0	0
1	0	0	0	0	1	0
1	0	1	0	0	0	1
1	1	0	0	0	1	1
1	1	1	0	0	0	1

## ЛЕКЦИЯ № 4



### Тема: ОСНОВЫ АЛГОРИТМИЗАЦИИ

#### Цель лекции

Ознакомиться с понятием алгоритма и основными способами разработки алгоритмов. Научиться составлять простые алгоритмы в виде блок-схем.

#### Понятие алгоритма

**Алгоритм** — это система точно сформулированных правил, определяющая процесс преобразования допустимых исходных данных за конечное число шагов в искомый результат.

Разработать алгоритм решения задачи означает разбить задачу на последовательно выполняемые шаги.

#### Свойства алгоритма

**Дискретность** — это возможность разбиения алгоритма на отдельные элементарные действия, выполнение которых не вызывает сомнения.

**Однозначность (или определенность) алгоритма** — это когда в алгоритме отсутствуют шаги, действия, смысл которых может восприниматься неоднозначно, то есть разными исполнителями по-своему.

**Массовость.** Под массовостью понимается, что алгоритм разрабатывается так, что его можно применить для целого класса задач, различающихся лишь наборами исходных данных.

**Конечность.** Под конечностью алгоритма понимают завершение работы алгоритма за конечное число шагов.

**Понятность.** Алгоритм строится для конкретного исполнителя человеком и должен быть ему понятен. Это облегчает его проверку и модификацию при необходимости.

**Результативность.** Алгоритм всегда должен приводить к результату.

#### Способы описания алгоритма

Алгоритмы, исполнителем которых является ЭВМ, описывают:

- В текстовой (*вербальной*) форме.
- Графически в виде схемы.
- На искусственном языке, то есть языке программирования



## Графическая форма записи алгоритма

Правила выполнения, и условные обозначения схем алгоритмов представляются соответствующими ГОСТами Единой Системы Программной Документации (ЕСПД):

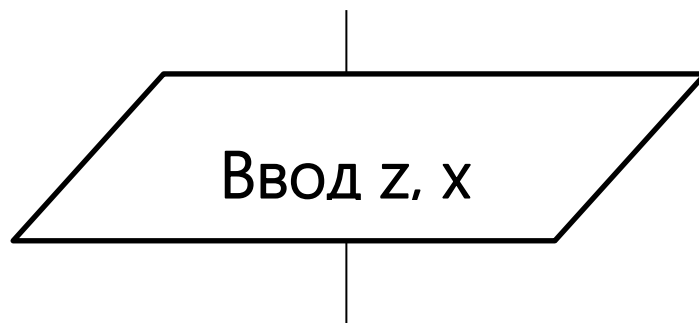
**ГОСТ 19.001 – 77.** Общие положения.

**ГОСТ 19.002 – 80.** Схемы алгоритмов и программ. Правила выполнения.

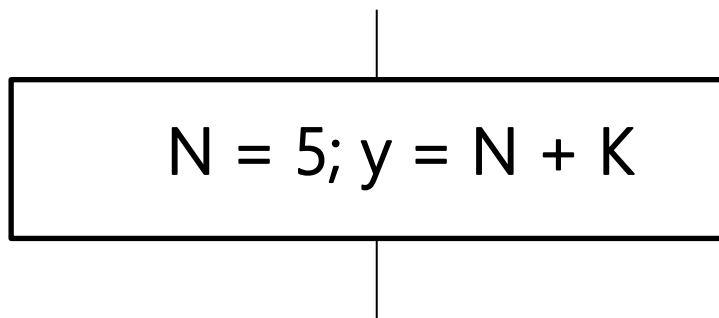
**ГОСТ 19.003 – 80.** Схемы алгоритмов и программ. Обозначения условные графические.

### Основные условные графические обозначения

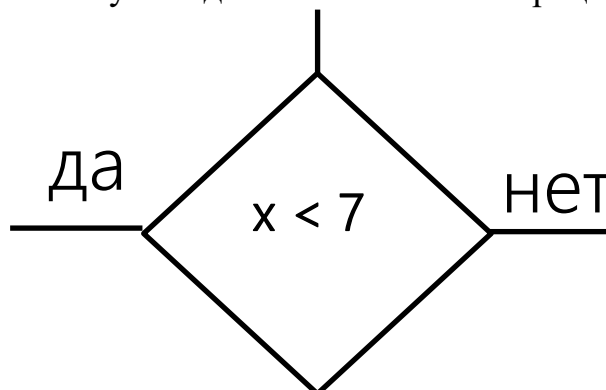
#### 1. Ввод/вывод



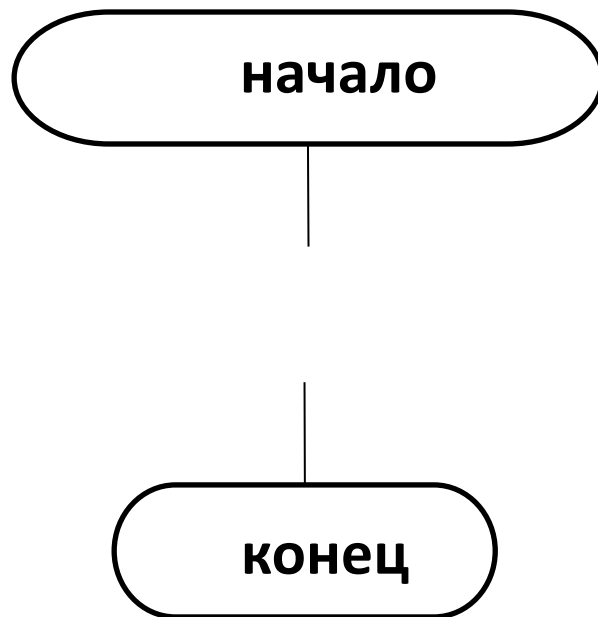
2. Процесс - Выполнение одной или группы операций. Часто используется для обозначения операции присваивания.



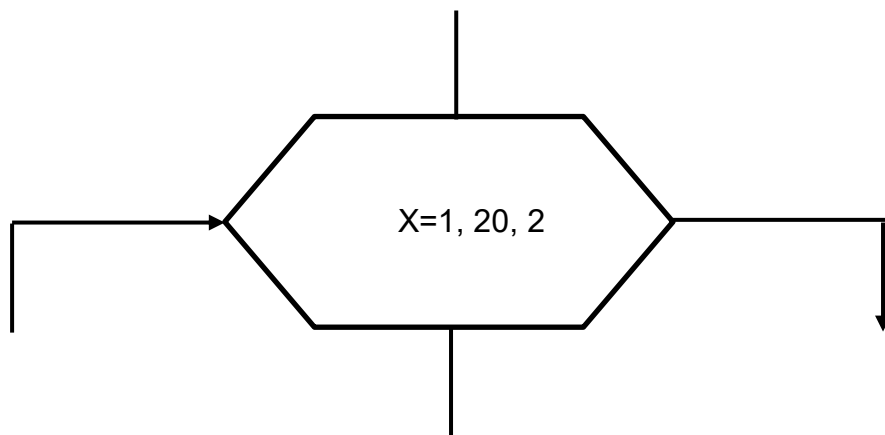
3. Решение - Используется для обозначения операции условного перехода.



4. Терминатор - используется для обозначения начала и окончания процесса решения задачи.

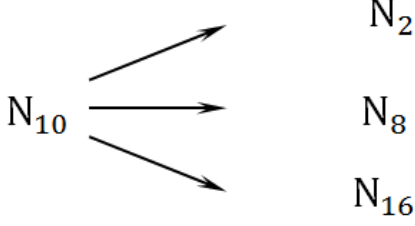
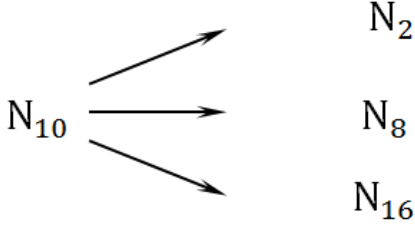
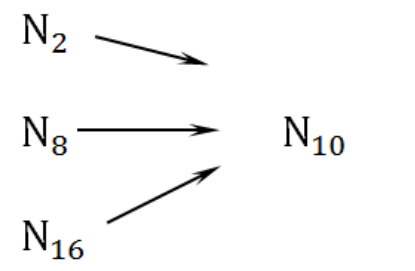
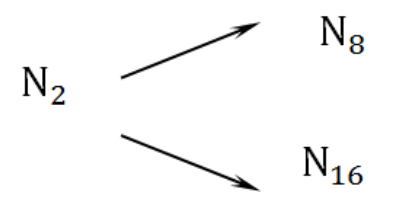
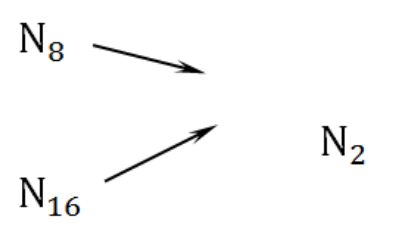


5. Подготовка - используется для организации циклических конструкций.



Приложение 1  
(справочное)

**Таблица правил перевода чисел из одной системы счисления в другую**

Направление перевода	Правила перевода
<p>Целая часть числа</p> 	<p>деление на 2</p> <p>деление на 8</p> <p>деление на 16</p>
<p>Дробная часть числа</p> 	<p>умножение на 2</p> <p>умножение на 8</p> <p>умножение на 16</p>
	<p>представить число в развернутой форме, например:</p> $\begin{matrix} 3 & 2 & 1 & 0 \\ 1011_2 = 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 8 + 2 + 1 = 11_{10} \end{matrix}$ $\begin{matrix} 1 & 0 \\ 75_8 = 7 \cdot 8^1 + 5 \cdot 8^0 = 56 + 5 = 61_{10} \end{matrix}$ $\begin{matrix} 1 & 0 \\ A3_{16} = A \cdot 16^1 + 3 \cdot 16^0 = 10 \cdot 16 + 3 = 163_{10} \end{matrix}$
	<p>замена триад цифрами восьмеричной системы счисления</p> <p>замена тетрад цифрами шестнадцатеричной системы счисления</p>
	<p>замена восьмеричных цифр триадами</p> <p>замена шестнадцатеричных цифр тетрадами</p>

**Приложение 2**  
**(справочное)**

**Таблица соответствия между системами счисления**

Десятичная	Двоичная	Восьмеричная	Шестнадцатеричная
0	0	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

**Правила выполнения арифметических операций  
в двоичной системе счисления**

**Правила сложения**

$$0 + 0 = 0$$

$$0 + 1 = 1$$

$$1 + 0 = 1$$

$$1 + 1 = 10$$

**Правила умножения**

$$0 \cdot 0 = 0$$

$$0 \cdot 1 = 0$$

$$1 \cdot 0 = 0$$

$$1 \cdot 1 = 1$$

**Правила вычитания**

$$0 - 0 = 0$$

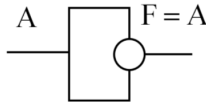
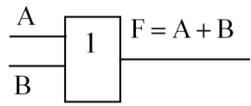
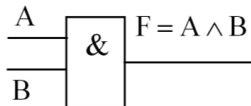
$$1 - 0 = 1$$

$$1 - 1 = 0$$

$$10 - 1 = 1$$

## Приложение 3 (справочное)

### Логические функции

Функция	Соответствие речевым оборотам	Обозначение	Таблица истинност и	Истинность	Логическая схема															
Отрицание (инверсия)	НЕ НЕВЕРНО, ЧТО	не $A$ , $\bar{A}$ , $\neg A$ , $not$	<table><tr><td><math>A</math></td><td><math>F</math></td></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table>	$A$	$F$	0	1	1	0	Инверсия логической переменной истинна, если сама переменная ложна, и, наоборот, инверсия ложна, если переменная истинна.	 $F = \bar{A}$									
$A$	$F$																			
0	1																			
1	0																			
Логическое сложение (дизъюнкция ).	ИЛИ	$+$ , или, $or$ , $\vee$	<table><tr><td><math>A</math></td><td><math>B</math></td><td><math>F</math></td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	$A$	$B$	$F$	0	0	0	0	1	1	1	0	1	1	1	1	Дизъюнкция ложна тогда и только тогда, когда оба высказывания ложны.	 $F = A + B$
$A$	$B$	$F$																		
0	0	0																		
0	1	1																		
1	0	1																		
1	1	1																		
Логическое умножение (конъюнкция ).	И	$\times$ , $\&$ , и, $and$ , $\wedge$	<table><tr><td><math>A</math></td><td><math>B</math></td><td><math>F</math></td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	$A$	$B$	$F$	0	0	0	0	1	0	1	0	0	1	1	1	Конъюнкция истинна тогда и только тогда, когда оба высказывания истинны.	 $F = A \wedge B$
$A$	$B$	$F$																		
0	0	0																		
0	1	0																		
1	0	0																		
1	1	1																		
Логическое следование (импликация )	ЕСЛИ...ТО	$\rightarrow$ , $\Rightarrow$	<table><tr><td><math>A</math></td><td><math>B</math></td><td><math>F</math></td></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	$A$	$B$	$F$	0	0	1	0	1	1	1	0	0	1	1	1	Импликация истинна всегда, за исключением случая, когда $A$ истинно, а $B$ ложно.	
$A$	$B$	$F$																		
0	0	1																		
0	1	1																		
1	0	0																		
1	1	1																		
Равнозначно сть (эквиваленц ия)	ЭКВИВАЛЕНТ НО	$=$ , $\leftrightarrow$ , $\Leftrightarrow$	<table><tr><td><math>A</math></td><td><math>B</math></td><td><math>F</math></td></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	$A$	$B$	$F$	0	0	1	0	1	0	1	0	0	1	1	1	Эквиваленция истинна тогда и только тогда, когда оба высказывания одновременно либо истинны, либо ложны.	
$A$	$B$	$F$																		
0	0	1																		
0	1	0																		
1	0	0																		
1	1	1																		

## Литература

1. Макарова, Н.В. Информатика: учебник для вузов / Н.В. Макарова, В.Б. Волков. – Санкт-Петербург: Питер, 2011. – 576 с.: ил.
2. Акулов, О.А. Информатика: базовый курс: учебник для втузов, бакалавров, магистров по направлению «Информатика и вычислительная техника» / О.А. Акулов, Н.В. Медведев. – 7-е изд., стер. – Москва: Омега-Л, 2012 . – 574 с. – (Высшее техническое образование).