

Министерство сельского хозяйства РФ

ФГБОУ ВПО «КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ  
АГРАРНЫЙ УНИВЕРСИТЕТ»

К. С. Галиев, Е. К. Печурина

**Двоичная система  
и представление информации  
в компьютере**

Учебно-методическое пособие

Под редакцией доктора технических наук,  
профессора В. И. Лойко

Краснодар  
КубГАУ  
2014

**УДК 004.032.8:681.3.042 (078)**

**ББК 32.81**

**Г15**

**Рецензенты:**

**Г. А. Аршинов** – доктор технических наук, профессор  
(Кубанский государственный аграрный университет);

**Е. В. Луценко** – доктор экономических наук, профессор  
(Кубанский государственный аграрный университет)

**Галиев К. С.**

**Г15** Двоичная система и представление информации в компьютере : учеб.-метод. пособие / К. С. Галиев, Е. К. Печурина; под ред. д-ра техн. наук, проф. В. И. Лойко. – Краснодар : КубГАУ, 2014. – 107 с.

Учебно-методическое пособие посвящено одному из разделов дисциплины «Информатика»: основам системы счисления и представлению информации в памяти компьютера. Рассматриваются вопросы видов систем счисления, перевод чисел из одной системы в другую, двоичная арифметика, представление (кодирование) чисел, текста и графических изображений.

Предназначено студентам–бакалаврам, обучающимся по направлению 140400 «Электроэнергетика и электротехника»; 110800 «Агроинженерия»; 270800 «Строительство».

**УДК 004.032.8:681.3.042 (078)**

**ББК 32.81**

© Галиев К. С.,  
Печурина Е. К., 2014  
© ФГБОУ ВПО «Кубанский  
государственный аграрный  
университет», 2014

## Оглавление

ВВЕДЕНИЕ .....	5
1 ОСНОВНЫЕ ПОНЯТИЯ О КОМПЬЮТЕРЕ .....	6
1.1 Понятие компьютера по Джону фон Нейману .....	6
1.2 Понятие информации .....	9
1.3 Представление информации .....	11
1.4 Двоичная система .....	13
2 ПОНЯТИЯ В СИСТЕМЕ СЧИСЛЕНИЯ .....	14
2.1 Понятие системы счисления .....	14
2.2 Примеры систем счисления .....	20
2.3 Взаимное соответствие чисел .....	22
2.4 Полиномиальная форма записи числа .....	25
2.5 Перевод недесятичного числа в десятичное .....	26
2.6 Перевод десятичного числа в недесятичное методом деления и умножения на основание .....	27
2.7 Перевод десятичного числа в недесятичное методом разложения по степеням основания .....	29
2.8 Взаимный перевод чисел между $D_2$ , $D_8$ и $D_{16}$ .....	31
2.9 Используемые системы счисления в компьютерной технологии .....	33
2.10 Вопросы и задания: .....	33
3 ДВОИЧНАЯ АРИФМЕТИКА .....	35
3.1 Сложение .....	35
3.2 Умножение .....	37
3.3 Вычитание .....	40
3.4 Деление .....	43
3.5 Арифметика с дробной частью .....	44
3.6 Замечание об арифметических операциях .....	49
3.7 Примеры арифметических операций в $D_8$ и $D_{16}$ .....	49
3.8 Таблицы сложения и умножения в $D_8$ и $D_{16}$ .....	52

3.9 Вопросы и задания.....	57
4 КОДИРОВАНИЕ ЧИСЕЛ.....	59
4.1 Замечание о кодировании информации .....	59
4.2 Замечание об отрицательном числе.....	60
4.3 Кодирование целого числа .....	61
4.4 Арифметические операции с кодированными целыми числами .....	65
4.5 Кодирование вещественного числа .....	69
4.6 Вопросы и задания.....	73
5 КОДИРОВАНИЕ ТЕКСТА .....	75
5.1 Кодирование текстовой информации .....	75
5.2 Кодовые страницы .....	80
5.3 Кодировка Unicode .....	81
5.4 Кодировка текста в документе .....	84
5.5 Вопросы и задания.....	86
6 КОДИРОВАНИЕ ГРАФИКИ.....	87
6.1 Кодирование графической информации .....	87
6.2 Кодирование цвета.....	90
6.3 Особенности растрового кодирования .....	99
6.4 Замечание о кодировании файла.....	100
6.5 Вопросы и задания.....	101
Список использованных источников.....	103
Приложение .....	104

## ВВЕДЕНИЕ

В настоящее время компьютеры используются практически во всех сферах деятельности человека. В пособии речь идёт о том, в каком виде представлять информацию для её обработки компьютером. Рассматриваются основные (элементарные) вопросы представления информации путем её кодирования на основе двоичной системы счисления.

В первой главе учебного пособия приводятся пояснения слов, входящих в название книжки.

Во второй главе приводятся основные понятия систем счисления и перевод чисел из одной системы счисления в другую систему.

В третьей главе приводятся двоичная арифметика.

В четвертой главе описываются основные методы представления (кодирования) числовой информации в компьютере на основе двоичной системы счисления.

В пятой главе описываются основные методы представления (кодирования) текстовой информации в компьютере на основе двоичной системы счисления.

В шестой главе описываются основные методы представления (кодирования) графической информации в компьютере на основе двоичной системы счисления.

Подробно рассматриваются примеры, поясняющие приведенные правила. Приводятся вопросы и задания для проверки степени усвоения учебного материала.

Успешные результаты, впервые достигнутые в компьютерной технологии по представлению (кодированию) разного вида информации, послужили основой для появления и развития *цифровой технологии* (мобильная связь, оцифровка музыки и фильмов для записи на компакт диски, цифровой фотоаппарат, цифровое телевидение и т. п.).

# 1 ОСНОВНЫЕ ПОНЯТИЯ О КОМПЬЮТЕРЕ

## 1.1 Понятие компьютера по Джону фон Нейману

Каждый слышал про компьютер и видел его. А что такое компьютер на самом деле? Попробуйте дать такое определение компьютера, чтобы отличить это техническое устройство от другого технического устройства, например, от велосипеда, автомобиля, трактора, стиральной машины, телевизора и т. п. Вопрос весьма не простой, как кажется на первый взгляд. Таким вопросом впервые люди озадачились 70 лет тому назад, когда появились первые компьютеры. Не видя компьютер, не разбираясь в его принципиальном устройстве, а только зная, что компьютер может «думать» в тысячи раз быстрее человека, работать сутками без усталости, некоторые люди, с ограниченным взглядом на окружающий мир, начали испытывать страх перед компьютером, полагая, что компьютеры, будучи «сильнее» человека, поработят человечество. Писатели-фантасты стали публиковать рассказы на эту тему.

Самое простое и достаточно грамотное определение понятия «компьютер» дал в 1945 г. Джон фон Нейман инженер и ученый, принимавший участие в разработке компьютера EDVAC – Electronic Discrete Variable Automatic Computer (электронный дискретный переменный автоматический вычислитель). Его определение компьютера попало в книги по «Информатике» как **принципы** Джона фон Неймана. Эти принципы составлены из нескольких пунктов. Отметим самые главные:

**1. Состав компьютера.** Компьютер, как техническое устройство, состоит из пяти частей:

1.1. Арифметико-логическое устройство (АЛУ) – выполняет арифметические операции над числами (сложение, вычитание, умножение и деление), проверяя логические условия типа «истина/ложь». Отсюда следует, что компьютер обрабатывает числа. Полное название компьютера – это ЦЭВМ – **цифровая** электронная вычислительная машина. Позднее стали говорить короче – ЭВМ. Компьютер (computer – вычислитель) принято называть в

англоязычных странах. Слова *компьютер* и *ЭВМ* – это синонимы, одно и то же.

1.2. Устройство управления (УУ) – управляет вычислениями, направляет потоки информации, в разные моменты времени (такты времени) включает в работу соответствующее устройство. Отсюда следует, что компьютер является не аналоговым (непрерывным) устройством, а **дискретным**, т. е. работает (выполняет операции) только в дискретные моменты времени. Чем короче промежуток времени между тактами (чем больше тактовая частота), тем быстрее работает компьютер. К слову сказать, тактовая частота составляет мегагерцы, сейчас гигагерцы.  $1 \text{ МГц} = 10^6 \text{ Гц}$ ,  $1 \text{ ГГц} = 10^9 \text{ Гц}$ . Отметим, что в современных компьютерах устройства АЛУ и УУ объединены в один блок и этот блок называется **процессор**.

1.3. Устройство памяти – хранит все обрабатываемые данные и программу по обработке данных.

1.4. Устройство ввода информации.

1.5. Устройство вывода информации.

На рисунке 1.1 показана схема компьютера по Джону фон Нейману.

2. Принцип **программного управления**. Этот принцип говорит о том, что компьютер управляется программой, а не человеком. Программа – это инструкция понятная ЭВМ. Вообще, инструкция, описывающая ход выполнения конкретной работы, называется **алгоритмом**. В нашем случае речь идет о выполнении конкретного расчета с помощью компьютера. Роль человека сводится к разработке алгоритма на понятном для человека языке (на алгоритмическом языке) и переводе этого алгоритма на машинный язык (в компьютерную программу). После этого компьютер будет управляться самостоятельно с помощью программы, без непосредственного участия человека.

Таким образом, получается, что компьютер – как аппаратно-программный комплекс – является автоматизированным средством обработки информации. Автоматизация – есть замена человека машиной для выполнения конкретной работы. Здесь отметим, что дисциплина «**Информатика**» изучает методы автоматизированной обработки информации.

3. Принцип **адресности памяти**. Этот принцип говорит о том, что устройство памяти состоит из отдельных ячеек, все ячейки пронумерованы (имеют адреса) и к каждой ячейке имеется доступ для записи или чтения информации.

4. Принцип **однородности памяти**. Это значит, что для ячейки памяти безразлично, что в ней записано – часть данных или часть программы. В памяти нет двух разных видов ячеек, отдельно для данных и отдельно для программы. Память однородна.

5. Принцип **хранимой программы**. Это говорит о том, что программа хранится в памяти. Программа вводится в память компьютера и остается там надолго. Исходные же данные для расчета должны вводиться каждый раз для разных вариантов расчета.

6. Принцип **двоичного кодирования**. Это значит, что обрабатываемая информация (числа), а также программа должны быть закодированы на основе двоичной системы счисления. Данные и программа должны быть представлены в памяти компьютера набором нулей и единиц. 0 и 1 есть базис двоичной системы. Именно поэтому выполняется принцип однородности памяти – в ячейке памяти находится набор нулей и единиц, и для ячейки неважно, что представляет собой этот набор из 0 и 1, данные или программу. Здесь отметим, что 0 и 1 – это математический образ физического явления, который представляет информацию. Например, электрический ток: 1 – есть ток, 0 – нет тока; или 1 – значимое напряжение (например, 5 В), 0 – малое напряжение (1,5 В); или магнитный слой: 1 – участок слоя намагничен, 0 – не намагничен; поверхность компакт диска: 1 – участок поверхности отражает падающий лазерный луч, 0 – участок поверхности рассеивает падающий лазерный луч; перфокарта или перфолента: 1 – есть отверстие, 0 – нет отверстия и т. п.

Таким образом, мы дали определение понятию *компьютер* по Джону фон Нейману (рисунок 1.1).



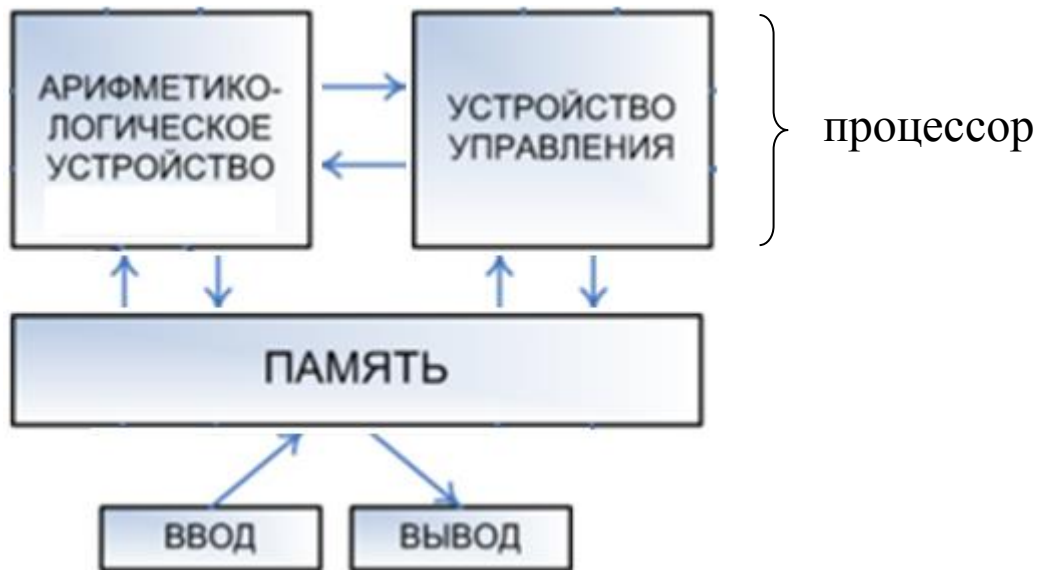


Рисунок 1.1 – Состав ЭВМ по Джону фон Нейману (1945)

## 1.2 Понятие информации

Каждый человек по-своему понимает понятие информации. Информация обладает большим разнообразием свойств: свежая информация, устаревшая, ценная, не очень важная, ложная, правдивая, много информации или мало; информацию можно сохранить, передать, испортить, купить или продать и т. п. Информация обладает одним уникальным свойством, которое присуще только ей, а именно, *информация не уменьшается*. Если информацию передать куда-либо, то информация у источника не убывает. Другие субстанции, такие как вещество и энергия, не обладают этим свойством: при их передаче они уменьшаются у источника. Вообще, до сих пор нет научного определения понятия информации. Для дальнейшего рассуждения, в рамках дисциплины «Информатика», примем следующее определение:

*информация – это сведения о чём-либо или о ком-либо, передаваемые в форме сигналов или знаков (символов).*

**Количество** информации можно измерить. Есть два подхода к понятию количества информации.

1. Количество информации можно рассматривать как меру уменьшения неопределенности знания при получении сообщения

(Хартли [5], Шеннон [6]). Если обозначить:  $N$  – количество равновероятных сообщений,  $i$  – количество информации, которое содержится в этих сообщениях, то существует следующая формула:

$$N = 2^i; \quad \text{или} \quad i = \log_2 N.$$

Здесь за единицу количества информации принимается такое количество информации, которое уменьшает неопределенность знания в два раза. Такая единица называется **бит**.

2. Другой подход – это количество информации, заполняющее некоторое пространство на носителе информации, не вдаваясь в смысл содержимого. Например, числа, буквы, текст, рисунок, записанные на листе бумаги. Или некая информация, сохраненная на магнитной ленте, диске и т. п. В таких случаях принято рассматривать знакоместо, в котором находится или отсутствует сообщение. Не важно, о чём сообщение, важно лишь то, что есть или нет в этом месте сообщение. Значение ответа: есть или нет (вернее да-нет), можно обозначить символами 1 и 0 (один и ноль). Такая единица информации, отвечающая на вопрос «да» или «нет», называется **бит**. Слово **бит** – это произношение английского слова *bit*, образованного из *binare digit* (двоичная цифра).

В дальнейшем будет рассматриваться только этот подход к количеству информации, а именно, объём информации (количество знакомест, количество бит), занятое сообщением. Если числовая информация записана в ячейки одномерной сетки, то в таком случае ячейка (знакоместо) называется разрядом.

Принято использовать следующие единицы для измерения объёма информации:

✓ бит (*bit*) – наименьшая единица, занимающая одно знакоместо, 1 бит = 1 или 0;

✓ байт (byte); 1 байт = 8 бит =  $2^3$  бит;

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

✓ мегабайт (Мб); 1 Мб =  $2^{10}$  Кбайт = 1024 Кбайт =  $2^{20}$  байт;

✓ гигабайт (Гб); 1 Гб =  $2^{10}$  Мбайт = 1024 Мбайт =  $2^{30}$  байт;

✓ терабайт (Тб); 1 Тб =  $2^{10}$  Гбайт = 1024 Гбайт =  $2^{40}$  байт.

В информатике (в информационной технологии) объем информации измеряется в единицах, кратных биту. 1 бит = 1 или 0, что составляет базис двоичной системы счисления. Поэтому в информатике приставка кило, мега, гига, тера к слову байт имеет значение кратное  $2^{10} = 1024$ . В распространенной метрической системе единиц, например, в международной системе единиц СИ (смотри последнюю страницу обложки школьной тетради) приставка кило имеет значение  $1000 = 10^3$ , мега =  $10^6$ , гига =  $10^9$  и т. д., потому что за основу счета принята десятичная система счисления. Например,  $1 \text{ кг} = 1000 \text{ г} = 10^3 \text{ г}$ ,  $1 \text{ Мл} = 10^6 \text{ л}$ .

### 1.3 Представление информации

Здесь речь идёт о том, в каком виде представлять информацию в процессе обработки. Кстати, что понимать под обработкой информации? В других сферах, например, обработка дерева – это пилить, строгать, сверлить, шлифовать и т. п.; обработка поля сельскохозяйственного назначения – это пахать, бороновать, вносить удобрение, поливать и т. п. В информатике под обработкой информации будем понимать четыре действия:

✓ *получение* информации (введение в компьютер в качестве исходных данных);

✓ *сохранение* информации (в памяти компьютера или на машинном носителе информации);

✓ *преобразование* информации (или представление, или кодировка информации для удобства работы с ней внутри компьютера);

✓ *выдача* (или передача информации из компьютера пользователю, например, на экран, на принтер).

Получение и выдача информации особых пояснений не требует – это числа, написанные в десятичной системе; текст, написанный на русском или иностранном языке. Преобразование информации является важнейшей составляющей её обработки. Привычные для нас десятичные числа, буквы, знаки препинания и т. п. в своём естественном виде не могут обрабатываться АЛУ (арифметико-логическим устройством). Такая информация должна быть закодирована двоичными символами (принцип

двоичного кодирования по Джону фон Нейману). Сохранение информации в памяти компьютера или на машинном носителе (магнитный диск, компакт диск, флешка) также выполняется в преобразованном (кодированном) виде. Основная часть данной книжки посвящена представлению (кодированию) информации в компьютере.

При представлении информации в компьютере следует отметить ещё понятие «данные». Иногда говорят «данные», иногда – «информация». Это одно и то же, или есть разница между этими понятиями? Выше было сказано, что информация - это сведения в виде символов, знаков, сигналов.

Информация – это осмысленные сведения.

*Информация = данные + комментарий.* Например,

325, к, р – это число, буквы (это данные, здесь нет смысла);

5 руб. – это деньги (это информация);

5 км – это расстояние (это информация).

Компьютер обрабатывает только **данные**. Данные подразделяются на:

✓ числа – 5, 527, 14.35;

✓ текст – «Вася», «x1y5»;

✓ графика – линия, рисунок, фотография.

Компьютер не понимает, что означает 527, x1y5, 5, что изображено на рисунке – дерево, кошка или деталь станка. Для компьютера это всё равно. Для компьютера это просто данные, которые надо обработать по заранее составленному правилу (алгоритму).

Возникает вопрос: а где же смысловое содержание обрабатываемых данных? А смысл данных знает только программист, который разработал программу. Точнее сказать, смысл данных содержится в программе. В программе всегда в начале обработки указывается *тип* данных, например, *integer* – целое число, *real* – вещественное число, *string* – строка (символы) и т. п.

То, что компьютер обрабатывает только данные, было хорошо видно в эпоху компьютеров 1-го поколения. Информация в компьютер вносилась с помощью перфокарт, а результаты обработки выводились печатающим устройством в виде чисел на узкую бумажную ленту. Перфокарта – это тонкий картонный пря-

моугольный лист с отверстиями; на перфокарте нет никаких комментариев для компьютера (рисунок 1.2).

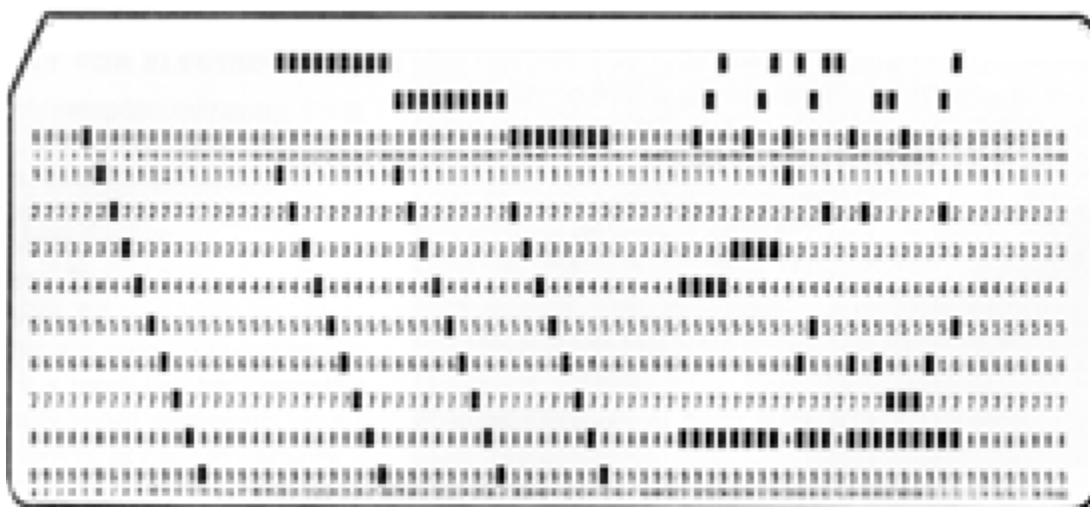


Рисунок 1.2 – Перфокарта

## 1.4 Двоичная система

Как было сказано выше, представление информации в компьютере выполняется её кодированием на основе двоичной системы счисления. Далее будут рассмотрены основные понятия систем счисления, перевод чисел из одной системы счисления в другую систему, принятые в компьютерной технологии системы счисления, двоичная арифметика и, наконец, представление информации (числовой, текстовой и графической) в компьютере.

Отметим, что успешные результаты, впервые достигнутые в компьютерной технологии по представлению разного вида информации, стали в дальнейшем использоваться в *цифровой технологии* (мобильная связь, оцифровка музыки и фильмов для записи на компакт диски, цифровой фотоаппарат, цифровое телевидение и т. п.).

## 2 ПОНЯТИЯ В СИСТЕМЕ СЧИСЛЕНИЯ

### 2.1 Понятие системы счисления

#### 2.1.1 Понятие числа и цифры

**Число** обозначает меру количества чего-либо, например, 25; 47,38; XXVII; XIV; 5AD4. Число состоит из набора символов, называемых *цифрами*. В приведенных примерах использованы следующие цифры (символы): 2; 5; 4; 7; 3; 8; X; V; I; A; D. Над числами можно выполнять арифметические операции (сложение, вычитание, умножение, деление) и числа можно сравнивать между собой. Иногда число может содержать только одну цифру, например, 5; X; 0; 1.

**Цифра** – это просто значок, символ, иногда может иметь буквенное начертание. Сделаем два замечания.

Во-первых, если цифра входит в состав числа, то эта цифра тоже является числом (обозначает меру количества). Например, вышеприведенные цифры являются отдельными числами.

Во-вторых, если символ, имеющий начертание общепринятой цифры, входит в состав слова (как грамматического понятия), то это уже не число, это просто символ. Над такими символами (цифрами) нельзя выполнять арифметических операций. Например, в словах BA32105 и BA32106 присутствуют цифры, как символы, а не как числа.

Человечество в процессе эволюции использовало различные начертания цифр (символов) для записи числа (рисунки 2.1 – 2.3).

#### 2.1.2 Понятие системы счисления

Выше было сказано, что число состоит из цифр и над числами можно выполнять арифметические операции.

*Системой счисления называются правила написания чисел с помощью цифр и выполнения арифметических операций над числами.*

Core- menная	Египет- кая (иероглифич.)	Египет- кая (иероглифич.)	Египет- ская (иероглифич.)	«Реческая» (латинская)	«Реческая» (ионическая)	Римская	Древне- реческая	Индийцев майя	Древнеки- тайская (палочек)	Древнеки- т (иероглифическая)	Индийск. (девана- гари)	Арабская (алфавит)	Арабская (core- menная)	Арабская (иероглифич.)
1	1	1	1	I	A	I	𐤀	•	—	1	1	1	1	1
2	11	11	11	II	B	II	𐤁	••	==	11	2	2	2	2
3	111	111	111	III	Г	III	𐤂	•••	===	111	3	3	3	3
4	1111	1111	1111	IV	Δ	IV	𐤃	••••	====	1111	4	4	4	4
5	11111	11111	11111	V	E	V	𐤄	—	=====	11111	5	5	5	5
6	111111	111111	111111	VI	F	VI	𐤅	—•	=====	111111	6	6	6	6
7	1111111	1111111	1111111	VII	Z	VII	𐤆	•••	=====	1111111	7	7	7	7
8	11111111	11111111	11111111	VIII	H	VIII	𐤇	••••	=====	11111111	8	8	8	8
9	111111111	111111111	111111111	IX	Θ	IX	𐤈	•••••	=====	111111111	9	9	9	9

Рисунок 2.1 — Обозначения чисел (единицы)

Совре- менная	Египетс- кая (иеро- глифы)	Египетс- кая (иеро- глифическая)	Египетс- кая (иеро- глифическая)	Бавилон- ская	Греческая (латинес- кая)	Греческая (ионичес- кая)	Римская	Древнеев- рейская	Индийцев майя	Древнеки- тайская (палочк )	Древнеки- т (иерогли- фическая)	Индийск. (дewan- гари)	Арабская (алфавит)	Арабская (совре- менная)	Арабская (робари)
10	𐤀	𐤀	Δ	Ι	Ι	Ι	X	𐤁	≡	—	+	10	9	1.	10
20	𐤁	𐤁	ΔΔ	Κ	Κ	Κ	XX	𐤂	⦿	≡	++	20	𐤃	2.	20
30	𐤂	𐤂	ΔΔΔ	Λ	Λ	Λ	XXX	𐤃	⦿	≡	+++	30	𐤄	3.	30
40	𐤃	𐤃	ΔΔΔΔ	Μ	ΔΔΔΔ	ΔΔΔΔ	XL	𐤄	⦿	≡	++++	40	𐤅	4.	40
50	𐤄	𐤄	𐤀	Ν	𐤀	𐤀	L	𐤅	⦿	≡	++++	50	𐤆	5.	50
60	𐤅	𐤅	𐤁	Ξ	𐤁	𐤁	LX	𐤆	⦿	⊥	++++	60	𐤇	6.	60
70	𐤆	𐤆	𐤂	Ο	𐤂	𐤂	LXX	𐤇	⦿	⊥	++++	70	𐤈	7.	70
80	𐤇	𐤇	𐤃	Π	𐤃	𐤃	LXXX	𐤈	⦿	⊥	++++	80	𐤉	8.	80
90	𐤈	𐤈	𐤄	Ρ	𐤄	𐤄	XC	𐤉	⦿	⊥	++++	90	𐤊	9.	90

Рисунок 2.2 – Обозначения чисел (десятки)



Соре- менная	Египет- ская (иероглифич.)	Египет- ская (иероглифическая)	Вавилон- ская	Греческая (латинес- кая)	Греческая (ионичес- кая)	Римская	Древнеев- рейская	Индиев мая	Древнеки- тайская (палочк )	Древнеки- т (персид- ская)	Индийск. (девана- гари)	Арабская (алфавит)	Арабская (соре- менная)	Арабская (робари)
100	𐤀	𐤁	𐤂	Η	Ρ	C	𐤃	𐤄	一	𐎠	100	١٠٠	1.	100
200	𐤂𐤂	𐤂𐤂	𐤂𐤂	ΗΗ	Σ	CC	𐤄𐤄	𐤄𐤄	二	𐎠𐎠	200	٢٠٠	2.	200
300	𐤂𐤂𐤂	𐤂𐤂𐤂	𐤂𐤂𐤂	ΗΗΗ	Τ	CCC	𐤄𐤄𐤄	𐤄𐤄𐤄	三	𐎠𐎠𐎠	300	٣٠٠	3.	300
400	𐤂𐤂𐤂𐤂	𐤂𐤂𐤂𐤂	𐤂𐤂𐤂𐤂	ΗΗΗΗ	Υ	CD	𐤄𐤄𐤄𐤄	𐤄𐤄𐤄𐤄	四	𐎠𐎠𐎠𐎠	800	٨٠٠	4.	400
500	𐤂𐤂𐤂𐤂𐤂	𐤂𐤂𐤂𐤂𐤂	𐤂𐤂𐤂𐤂𐤂	Ρ	Φ	D	𐤄𐤄𐤄𐤄𐤄	𐤄𐤄𐤄𐤄𐤄	五	𐎠𐎠𐎠𐎠𐎠	500	٥٠٠	5.	500
600	𐤂𐤂𐤂𐤂𐤂𐤂	𐤂𐤂𐤂𐤂𐤂𐤂	𐤂𐤂𐤂𐤂𐤂𐤂	ΡΗ	Χ	DC	𐤄𐤄𐤄𐤄𐤄𐤄	𐤄𐤄𐤄𐤄𐤄𐤄	𐎠	𐎠𐎠𐎠𐎠𐎠𐎠	500	٦٠٠	6.	600
700	𐤂𐤂𐤂𐤂𐤂𐤂𐤂	𐤂𐤂𐤂𐤂𐤂𐤂𐤂	𐤂𐤂𐤂𐤂𐤂𐤂𐤂	ΡΗΗ	Ψ	DCC	𐤄𐤄𐤄𐤄𐤄𐤄𐤄	𐤄𐤄𐤄𐤄𐤄𐤄𐤄	𐎠𐎠	𐎠𐎠𐎠𐎠𐎠𐎠𐎠	700	٧٠٠	7.	700
800	𐤂𐤂𐤂𐤂𐤂𐤂𐤂𐤂	𐤂𐤂𐤂𐤂𐤂𐤂𐤂𐤂	𐤂𐤂𐤂𐤂𐤂𐤂𐤂𐤂	ΡΗΗΗ	Ω	DCCC	𐤄𐤄𐤄𐤄𐤄𐤄𐤄𐤄	𐤄𐤄𐤄𐤄𐤄𐤄𐤄𐤄	𐎠𐎠𐎠	𐎠𐎠𐎠𐎠𐎠𐎠𐎠𐎠	800	٨٠٠	8.	800
900	𐤂𐤂𐤂𐤂𐤂𐤂𐤂𐤂𐤂	𐤂𐤂𐤂𐤂𐤂𐤂𐤂𐤂𐤂	𐤂𐤂𐤂𐤂𐤂𐤂𐤂𐤂𐤂	ΡΗΗΗΗ	Λ	CM	𐤄𐤄𐤄𐤄𐤄𐤄𐤄𐤄𐤄	𐤄𐤄𐤄𐤄𐤄𐤄𐤄𐤄𐤄	𐎠𐎠𐎠𐎠	𐎠𐎠𐎠𐎠𐎠𐎠𐎠𐎠𐎠	900	٩٠٠	9.	900

Рисунок 2.1 – Обозначения чисел (сотни)

Различают два типа систем счисления: позиционная и непозиционная. Позиция – это номер знакоместа цифры в числе. Исторически первой появилась непозиционная система.

**Непозиционная система счисления** – это когда значение цифры *не зависит* от её позиции в числе. Например:

*Счетные палочки:* I I I I I I I – каждая палочка означает единицу; количество палочек показывает значение числа.

*Римская система* счета состоит из цифр I, V, X, L, C, D, M. Каждая цифра имеет постоянное значение независимо от её позиции в числе, а именно, 1, 5, 10, 50, 100, 500, 1000.

Например, правило написания числа:

$$XXVI = 10+10+5+1 = 26; \quad XXVIII = 10+10+5+4 = 29.$$

Правило написания и чтения числа такое же, как со счетными палочками: сложить все цифры с учетом их значений. При этом цифры располагать, начиная со старших. Если младшая цифра расположена слева от старшей, то значение этой младшей цифры вычитается, например:

$$XXIV = 10+10-1+5 = 24; \quad XXIX = 10+10-1+10 = 29.$$

Арифметическая операция сложения:

$$\begin{array}{r} X \ X \ V \ I \ + \ X \ X \ I \ V \ = \ 26 + 24 = 50 \\ + \ X \ X \ V \ I \\ + \ X \ X \ I \ V \\ \hline = \ X \ X \ X \ X \ V \ I \ I \ V \\ = \ X \ X \ X \ X \ X \ = \ L \ = \ 50 \end{array}$$

При операции сложения надо сложить значения всех цифр.

Арифметическая операция умножения:

$$\begin{array}{r} X \ X \ V \ I \ * \ X \ X \ I \ V \ = \ 26 * 24 = 624 \\ \left\{ \begin{array}{l} + \ C \ C \ L \ X \\ + \ C \ C \ L \ X \\ - \ X \ X \ V \ I \\ + \ L \ L \ X \ X \ V \ V \end{array} \right. \\ \hline = \ C \ C \ C \ C \ L \ L \ L \ X \ X \ I \ V \ = \\ = \ C \ D \ C \ C \ X \ X \ I \ V \ = \\ = -100+500+100+100+10+10-1+5 = 624 \end{array}$$

При операции умножения надо первое число умножать на каждую цифру второго числа, затем сложить все слагаемые с учетом знака слагаемого.

Другие примеры непозиционных систем счисления показаны на рисунках 2.1 – 2.3. Можно отметить следующие две особенности непозиционных систем счисления:

1) Правила записи и чтения числа, а также выполнение арифметических операций – не сложные; правила просто трудоемкие – число состоит из множества цифр, которые надо сложить; при этом цифра в любой позиции имеет постоянное значение.

2) В непозиционной системе отсутствует цифра, имеющая нулевое значение. Понятие нуля (0) – это абстракция. Как записать количество того, чего нет в природе? Поэтому в непозиционной системе счисления, в общем-то, не было потребности в нуле.

Позже для удобства выполнения вычислений человечество перешло к позиционной системе, в которой появилась потребность в значке (цифре), обозначающей положение позиции, где отсутствует значащая цифра.

**Позиционная система счисления** – это когда значение цифры *зависит* от её позиции. Например: число 555 состоит из трёх одинаковых цифр, однако значение каждой цифры разное, а именно, первая цифра обозначает пятьсот, вторая цифра – пятьдесят, третья цифра – пять.

$$\begin{array}{ccccccc}
 5 & 5 & 5 & = & 500 & + & 50 & + & 5 \\
 \downarrow & \downarrow & \downarrow & & & & & & \\
 & & & & & & 5 & & \\
 & & & & & & \downarrow & & \\
 & & & & & & 50 & & \\
 & & & & & & \downarrow & & \\
 & & & & & & 500 & & 
 \end{array}$$

Дальше будем рассматривать только позиционную систему счисления. В системе счисления существуют два важных понятия: базис и основание.

**Базис** системы счисления – это *набор* цифр, применяемых в данной системе. Иногда говорят «алфавит» системы счисления.

**Основание** системы счисления – это *количество* цифр, применяемых в данной системе.

Примеры позиционных систем счисления:

- ✓ Десятичная система – 0,1,2,3,4,5,6,7,8,9 – (основание = 10).
- ✓ Восьмеричная система – 0,1,2,3,4,5,6,7 – (основание = 8).
- ✓ Двоичная система – 0,1 – (основание = 2).
- ✓ Шестнадцатеричная система – 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F (основание = 16).

Номер позиции (номер места цифры) в позиционной системе указывается справа налево, начиная с 0 от запятой:

3 2 1 0 -1 -2 -3	← номер позиции
7 9 3 5, 4 8 6	← число

В позиционной системе, например, в десятичной системе, позиция 0 означает количество единиц (значение цифры базиса); позиция 1 означает количество десятков (значение цифры, умноженное на десять); позиция 2 означает количество сотен (значение цифры, умноженное на сто) и т. д.

Например, число  $555 = 500 + 50 + 5$ .

### 2.1.3 Замечание о нуле

Запишем в десятичной позиционной системе счисления число, например, равное «четыреста семь». Здесь значение «семь» будет обозначаться цифрой 7 и занимать нулевую позицию, а значение «четыреста» будет обозначаться цифрой 4 и занимает вторую позицию. Первая позиция остается пустой, в этом месте нет значащей цифры. *Поэтому принято пустое место (пустую позицию) в записи числа обозначать круглым значком 0, который стал называться нулём.*

Тогда число «четыреста семь» запишется как 407.

## 2.2 Примеры систем счисления

В позиционной системе счисления можно написать число с любым целым основанием больше единицы. Например, троичная система счисления, или пятеричная, или двадцатеричная и т. п. В практической деятельности приходится пользоваться довольно часто некоторыми системами счисления, приспособленными не

столько для счета, сколько для обозначения некоторых событий. Например:

1) Упомянутая выше *римская* система. Используется для обозначения глав в книгах, года возведения некоторых уникальных зданий, указания столетия (века) исторического события и т. п.

2) *Семеричная* система. Базис = 1, 2, 3, 4, 5, 6, 7. Названия цифр: понедельник, вторник, ..., воскресенье. Используется для обозначения дня недели.

3) *Двенадцатеричная* система. Базис = 1, 2, 3, ..., 12. Иногда цифрам присваивается название: январь, февраль, ..., декабрь. Используется для указания месяца года, или для указания времени суток в часах. Некоторые предметы принято считать дюжинами (дюжина = 12). До сих пор вилки, ножи, ложки продают дюжинами, а посудные сервизы (чайные и столовые) по традиции все еще составляют из 12 комплектов. Школьные тетради имеют количество страниц кратных 12. Английский фунт состоит из 12 шиллингов, а шиллинг из 12 пенсов. В футе содержится 12 дюймов. Числительные от 1 до 12 в немецком языке являются одинарными словами, а с 13 – составными словами. В частности, 1 – eins, 2 – zwei, 3 – drei, ..., 10 – zehn, 11 – elf, 12 – zwölf, 13 – dreizehn и т. д.

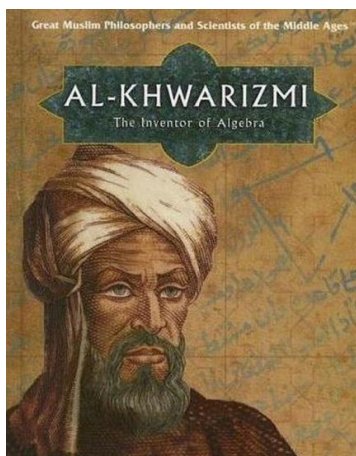


Принято считать, что число 12 равно количеству фаланг на всех пальцах одной руки, не считая большого пальца. А большой палец предназначен отмечать отсчитанные фаланги.

4) *Шестидесятеричная* система. Используется для указания количества минут и секунд. 1 ч = 60 мин, 1 мин = 60 с. Иногда приходится выполнить перерасчет времени заданного в часах, минутах и секундах, например, 3 ч 47 мин 38 с в количество секунд.

5) *Десятичная* система. Используется повсеместно для счета и удобна для выполнения арифметических операций. Цифры десятичной системы принято называть «*арабскими цифрами*», хотя такое название исторически не совсем правильное.

История появления названия «арабские *цифры*» связана с именем среднеазиатского ученого IX в. аль-Хорезми. Полное имя ученого: Абу Абдуллах Мухаммад ибн Муса аль-Хорезми.



Аль-Хорезми считается автором девяти сочинений на арабском языке [7].

Одной из самых известных и значительных его работ является «Книга о восстановлении и противопоставлении» («Китаб аль-джебр валь-мукабала»), в которой были заложены основы алгебры как самостоятельной науки.

Само слово «алгебра» происходит от начала названия сочинения аль-Хорезми («аль-джебр»), которое в XII в. было переведено на латинский язык. Китаб переводится как книга.

Другой важной работой аль-Хорезми является «Книга об индийском счете». В ней было впервые дано систематическое изложение арифметики, основанной на индийской десятичной позиционной системе счисления с девятью цифрами и нулём. Перевод этого труда на латынь был выполнен в XII в., и в латинской транскрипции имя автора – аль-Хорезми – звучало как «Алгоризми». Прозвище аль-Хорезми в видоизмененной форме превратилось в нарицательное слово "алгоритм" и сначала означало всю систему десятичной позиционной арифметики. Поскольку книга была очень популярна среди европейцев, то такую арифметику стали называть алгоритмом. Впоследствии термин «алгоритм» приобрел более широкий смысл в математике, как правило выполнения операций в определенном порядке.

Латинский перевод арабского сочинения начинается со слов: "...и сказал Алгоризми", что послужило поводом для европейцев называть эти цифры *арабскими*.

## 2.3 Взаимное соответствие чисел

В компьютерной технологии используются четыре типа позиционных систем счисления. Почему именно эти четыре типа си-

стем счисления будут объяснены ниже в п. 2.5, а сейчас познакомимся с ними подробнее. Для удобства сокращенной записи приняты следующие обозначения:

$D_{10}$  – десятичная система счисления;

$D_2$  – двоичная система счисления;

$D_8$  – восьмеричная система счисления;

$D_{16}$  – шестнадцатеричная система счисления.

Рассмотрим несколько строк чисел этих систем счисления (таблица 2.1). В таблице цифры базиса занимают одну позицию (цифры базиса выделены жирным шрифтом). Следующие числа после цифр базиса занимают 2 позиции, затем 3 позиции и т. д.

Рассмотрим, например, как записываются числа в  $D_{10}$ . После 9 следующими значениями являются числа 10, 11, ..., 19, 20, ..., 29, 30, 31 и т. д. По какому правилу записаны эти числа? Ноль (0) возглавляет *базис* и все цифры базиса занимают одну позицию. Затем берётся очередная цифра базиса 1 и к нему справа приставляются цифры базиса от 0 до 9. Получаем 10, 11, ..., 19. Затем берётся очередная цифра базиса 2 и к нему приставляются цифры базиса от 0 до 9, получаем 20, 21, ..., 29 и т.д. до 99. Две позиции заполнены старшими цифрами базиса – это 99. Затем берётся очередное число 10 и к нему приставляются цифры базиса от 0 до 9. Получаем 100, 101, 102, ..., 109. Затем берётся очередное число 11, приставляется базис и получаем 110, 111, 112, ..., 119 и т. д.

Можно по-другому рассматривать правило написания числа. Как известно, ноль (0) обозначает пустое место в числе, или пустую позицию, в которой нет значащей цифры. Значащиеся цифры базиса от 1 до 9 занимают одну позицию. После самой старшей цифры 9, значащие цифры начинают поочередно заполнять левую позицию, а правая позиция сначала становится пустой, т. е. обозначается цифрой 0, а затем опять заполняются значащими цифрами от 1 до 9 и т. д.

Аналогично записываются (формируются) числа в  $D_8$ . После старшей цифры базиса 7, очередное число занимает две позиции и пишется 10 (читается «один ноль в восьмеричной системе»).

Таблица 2.1– Взаимное соответствие чисел

<b>D<sub>10</sub></b>	<b>D<sub>8</sub></b>	<b>D<sub>16</sub></b>	<b>D<sub>2</sub></b>
<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>
<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>
<b>2</b>	<b>2</b>	<b>2</b>	10
<b>3</b>	<b>3</b>	<b>3</b>	11
<b>4</b>	<b>4</b>	<b>4</b>	100
<b>5</b>	<b>5</b>	<b>5</b>	101
<b>6</b>	<b>6</b>	<b>6</b>	110
<b>7</b>	<b>7</b>	<b>7</b>	111
<b>8</b>	10	<b>8</b>	1000
<b>9</b>	11	<b>9</b>	1001
10	12	<b>A</b>	1010
11	13	<b>B</b>	1011
12	14	<b>C</b>	1100
13	15	<b>D</b>	1101
14	16	<b>E</b>	1110
15	17	<b>F</b>	1111
16	20	10	10000
17	21	11	10001
18	22	12	10010
19	23	13	10011
20	24	14	10100
21	25	15	10101
22	26	16	10110
23	27	17	10111
24	30	18	11000
25	31	19	11001
26	32	1A	11010
27	33	1B	11011
28	34	1C	11100
29	35	1D	11101
30	36	1E	11110
31	37	1F	11111
32	40	20	100000



$10_8$  – это не «десять»; это «восемь относительно десятичной системы»; или «один ноль в восьмеричной системе». Также  $11_8$  – это не «одиннадцать»; это «девять относительно десятичной системы»; или «один один в восьмеричной системе».

Аналогично записываются числа в  $D_{16}$ . После старшей цифры базиса F, очередное число занимает две позиции и пишется 10 (читается «один ноль в шестнадцатеричной системе»).  $10_{16}$  – это не «десять»; это «шестнадцать относительно десятичной системы» или «один ноль в шестнадцатеричной системе». Также  $11_{16}$  – это не «одиннадцать»; это «семнадцать относительно десятичной системы» или «один один в шестнадцатеричной системе».

Аналогично записываются числа в  $D_2$ . После старшей цифры базиса 1, очередное число занимает две позиции и пишется 10 (читается «один ноль в двоичной системе»).  $10_2$  – это не «десять»; это «два относительно десятичной системы» или «один ноль в двоичной системе». Также  $11_2$  – это не «одиннадцать»; это «три относительно десятичной системы» или «один один в двоичной системе». Ещё пример записи:  $100_2$  – это не «сто»; это «четыре относительно десятичной системы» или «один ноль ноль в двоичной системе».

## 2.4 Полиномиальная форма записи числа

Слово «полином» – греческое (poly – много, nomos – часть), переводится как многочлен. Это математическое понятие, представляющее собой сумму, состоящую из многих слагаемых. Оказывается, что любое позиционное число можно записать в виде суммы слагаемых, в которой цифры являются коэффициентами при слагаемых. При записи исходного числа будем указывать основание системы счисления, чтобы видеть, к какой системе принадлежит число, а также номера позиций над цифрами.

Например:

$$\begin{array}{c} 2 \ 1 \ 0 \\ 555_{10} = 500 + 50 + 5 = 5 \cdot 10^2 + 5 \cdot 10^1 + 5 \cdot 10^0 ; \end{array}$$

$$\begin{array}{cccc} 3 & 2 & 1 & 0 \\ 2573_{10} = 2*10^3 + 5*10^2 + 7*10^1 + 3*10^0 ; \end{array}$$

$$\begin{array}{cccccc} 1 & 0 & -1 & -2 & -3 & -4 \\ 75,4098_{10} = 7*10^1 + 5*10^0 + 4*10^{-1} + 0*10^{-2} + 9*10^{-3} + 8*10^{-4} . \end{array}$$

Аналогично запишется число в другой системе счисления:

$$\begin{array}{cccc} 3 & 2 & 1 & 0 \\ 2573_8 = 2*8^3 + 5*8^2 + 7*8^1 + 3*8^0 ; \end{array}$$

$$\begin{array}{cccccc} 1 & 0 & -1 & -2 & -3 & -4 \\ 75,4098_{16} = 7*16^1 + 5*16^0 + 4*16^{-1} + 0*16^{-2} + 9*16^{-3} + 8*16^{-4} ; \end{array}$$

$$\begin{array}{cccccc} 1 & 0 & -1 & -2 & -3 \\ 48,A0D_{16} = 4*16^1 + 8*16^0 + 10*16^{-1} + 0*16^{-2} + 13*16^{-3} ; \end{array}$$

$$\begin{array}{ccccccccc} 4 & 3 & 2 & 1 & 0 & -1 & -2 & -3 \\ 11011,101_2 = 1*2^4 + 1*2^3 + 0*2^2 + 1*2^1 + 1*2^0 + 1*2^{-1} + 0*2^{-2} + 1*2^{-3} . \end{array}$$

Таким образом, правило записи числа в полиномиальной форме следующее:

- 1) Написать исходное число с указанием основания системы счисления.
- 2) Записать номера позиций над цифрами числа.
- 3) Каждую цифру умножать на основание, возведённую в степень номера позиции и составить многочлен из этих слагаемых.

## 2.5 Перевод недесятичного числа в десятичное

Рассмотрим перевод чисел из систем  $D_8$ ,  $D_{16}$  и  $D_2$  в  $D_{10}$ . Такой перевод выполняется очень просто: исходное число надо записать в форме полинома и вычислить сумму всех слагаемых.

Например:

$$\begin{array}{cccc} 3 & 2 & 1 & 0 \\ 2573_8 = 2*8^3 + 5*8^2 + 7*8^1 + 3*8^0 = \\ = 2*512 + 5*64 + 7*8 + 3*1 = 1403_{10} ; \end{array}$$

$$\begin{array}{cccccc} 1 & 0 & -1 & -2 & -3 & -4 \\ 75,4098_{16} = 7*16^1 + 5*16^0 + 4*16^{-1} + 0*16^{-2} + 9*16^{-3} + 8*16^{-4} = \\ = 7*16 + 5*1 + 4/16 + 0/256 + 9/4096 + 8/65536 = \\ = 117,2523193359375_{10} ; \end{array}$$

$$\begin{array}{r}
 1\ 0\ -1\ -2\ -3 \\
 48, A0D_{16} = 4 \cdot 16^1 + 8 \cdot 16^0 + 10 \cdot 16^{-1} + 0 \cdot 16^{-2} + 13 \cdot 16^{-3} = \\
 = 4 \cdot 16 + 8 \cdot 1 + 10/16 + 13/4096 = 72,628173828125_{10};
 \end{array}$$

$$\begin{array}{r}
 4\ 3\ 2\ 1\ 0\ -1\ -2\ -3 \\
 11011,101_2 = 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3} = \\
 = 16 + 8 + 2 + 1 + 1/2 + 1/8 = 27,625_{10}.
 \end{array}$$

## 2.6 Перевод десятичного числа в недесятичное методом деления и умножения на основание

Речь идёт о переводе числа из системы  $D_{10}$  в  $D_8$ ,  $D_{16}$  и  $D_2$ . Такой перевод выполняется отдельно для целой части десятичного числа и отдельно для дробной части десятичного числа. Рассмотрим, например, число  $75,25_{10}$ . Здесь 75 – целая часть числа; 0,25 – дробная часть числа.

**Правило для целой части десятичного числа.** Целая часть десятичного числа *делится* уголком на основание переводимой системы; получаем частное и остаток. Затем частное опять делится на основание, и так до тех пор, пока последнее частное не станет меньше делителя. Результатом перевода является последнее частное и остатки, записанные в обратном порядке.

Перевод целого числа  $75_{10}$  в  $D_2$ ,  $D_8$  и  $D_{16}$  производится следующим образом:

$D_2$	$D_8$	$D_{16}$
$  \begin{array}{r}  75 \overline{) 2} \\  \underline{1} \phantom{0} \\  13 \overline{) 2} \\  \underline{1} \phantom{0} \\  18 \overline{) 2} \\  \underline{0} \phantom{0} \\  9 \overline{) 2} \\  \underline{1} \phantom{0} \\  4 \overline{) 2} \\  \underline{0} \phantom{0} \\  2 \overline{) 2} \\  \underline{0} \phantom{0} \\  0 \phantom{0}  \end{array}  $	$  \begin{array}{r}  75 \overline{) 8} \\  \underline{3} \phantom{0} \\  9 \overline{) 8} \\  \underline{1} \phantom{0} \\  1  \end{array}  $	$  \begin{array}{r}  75 \overline{) 16} \\  \underline{11} \phantom{0} \\  4  \end{array}  $

Ответ:  $75_{10} = 1001011_2 = 113_8 = 4B_{16}$ .

**Правило для дробной части** десятичного числа. *Дробная часть десятичного числа умножается на основание переводимой системы; получим целую и дробную части. Затем дробную часть результата опять умножаем на основание, и так до тех пор, пока не настанет одна из трёх возможных ситуаций:*

- 1) дробная часть при умножении получилась  $=0$ ;
- 2) дробная часть стала повторяться;
- 3) прекращаем умножение, ограничившись требуемой точностью.

*Результатом перевода являются записанные сверху вниз числа целой части, полученные при умножении.*

Перевод дробной части числа  $0,25_{10}$  в  $D_2$ ,  $D_8$  и  $D_{16}$  производится следующим образом:

$D_2$	$D_8$	$D_{16}$
$0,25 \cdot 2 =$	$0,25 \cdot 8 =$	$0,25 \cdot 16 =$
$\downarrow \overline{0,50} \cdot 2 =$	$\downarrow \overline{2,00} \cdot 8 =$	$\downarrow \overline{4,00} \cdot 16 =$
$1,00 \cdot 2 =$	$0,00$	$0,00$
$\downarrow 0,00$		

Ответ:  $0,25_{10} = 0,01_2 = 0,2_8 = 0,4_{16}$ .

Рассмотрим ещё два примера перевода десятичных дробных чисел в двоичную систему  $0,725_{10}$  и  $0,672_{10}$ :

$0,725 \cdot 2 =$	$0,672 \cdot 2 =$
$\downarrow \overline{1,450} \cdot 2 =$	$\downarrow \overline{1,344} \cdot 2 =$
$0,900 \cdot 2 =$	$0,688 \cdot 2 =$
$1,800 \cdot 2 =$	$1,376 \cdot 2 =$
$1,600 \cdot 2 =$	$0,752 \cdot 2 =$
$1,200 \cdot 2 =$	$1,504 \cdot 2 =$
$0,400 \cdot 2 =$	$1,008 \cdot 2 =$
$0,800 \cdot 2 =$	$0,016 \cdot 2 =$
$1,600 \cdot 2 =$	$0,032 \cdot 2 =$
$\downarrow 1,200 \cdot 2 =$	$\downarrow 0,064 \cdot 2 =$

Ответ:  $0,725_{10} = 0,101(1100)(1100)_2$ .  
 $0,672_{10} = 0,101011_2$ .

Здесь в первом примере видна повторяющаяся часть, а во втором примере ограничиваемся требуемой точностью (шесть цифр после запятой).

Таким образом, исходное число  $75,25_{10}$  в других системах счисления запишется так:

$$75,25_{10} = 1001011,01_2 = 113,2_8 = 4B,4_{16}.$$

## 2.7 Перевод десятичного числа в недесятичное методом разложения по степеням основания

Рассмотрим другой метод перевода числа из системы  $D_{10}$  в  $D_8$ ,  $D_{16}$  и  $D_2$ . Для этого воспользуемся следующей простой таблицей возведения основания системы счисления в целые степени (таблица 2.2).

Таблица 2.2 – Возведение основания в целые степени

x	$2^x$	$2^{-x}$	$8^x$	$8^{-x}$	$16^x$	$16^{-x}$
0	1	1	1	1	1	1
1	2	0,5	8	0,125	16	0,0625
2	4	0,25	64	0,0015625	256	0,0039062
3	8	0,125	512	0,00195312	4096	0,0002441
4	16	0,0625	4096	0,00024414	65536	и т.д.
5	32	0,03125	32768	0,00003060	и т.д.	
6	64	0,015625	262144	и т.д.		
7	128	0,0078125	и т.д.			
8	256	0,00390625				
9	512	и т.д.				
10	1024					

Пусть задано десятичное число, например,  $75,25_{10}$ . Здесь 75 – целая часть числа; 0,25 – дробная часть числа.

Рассмотрим перевод в  $D_2$ . Разложим целую часть на составляющие, пользуясь числами из столбца  $2^X$ , а дробную часть – из столбца  $2^{-X}$ :

$$\begin{aligned} 75_{10} &= 64 + 8 + 2 + 1 = 2^6 + 2^3 + 2^1 + 2^0 = \\ &= 1 \cdot 2^6 + 0 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 1001011_2. \end{aligned}$$

Здесь набор слагаемых с коэффициентами, есть не что иное, как полиномиальная форма записи двоичного числа  $1001011_2$ .

Дробная часть:  $0,25_{10} = 0,25 = 2^{-2} = 0 \cdot 2^{-1} + 1 \cdot 2^{-2} = 0,01_2$ .

$$\begin{aligned} \text{Таким образом, } 75,25_{10} &= 2^6 + 2^3 + 2^1 + 2^0 + 2^{-2} = \\ &= 1 \cdot 2^6 + 0 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 + 0 \cdot 2^{-1} + 1 \cdot 2^{-2} = \\ &= 1001011,01_2. \end{aligned}$$

Рассмотрим перевод в  $D_8$ . Разложим целую часть на составляющие, пользуясь числами из столбца  $8^X$ , а дробную часть – из столбца  $8^{-X}$ :

$$\begin{aligned} 75_{10} &= 64 + 8 + 1 + 1 + 1 = 8^2 + 8^1 + 8^0 + 8^0 + 8^0 = \\ &= 1 \cdot 8^2 + 1 \cdot 8^1 + 3 \cdot 8^0 = 113_8. \end{aligned}$$

Здесь набор слагаемых с коэффициентами, есть не что иное, как полиномиальная форма записи восьмеричного числа  $113_8$ .

Дробная часть:

$$0,25_{10} = 0,125 + 0,125 = 8^{-1} + 8^{-1} = 1 \cdot 8^{-1} + 1 \cdot 8^{-1} = 2 \cdot 8^{-1} = 0,2_8.$$

$$\begin{aligned} \text{Таким образом, } 75,25_{10} &= 1 \cdot 8^2 + 1 \cdot 8^1 + 3 \cdot 8^0 + 2 \cdot 8^{-1} = \\ &= 1 \cdot 8^2 + 1 \cdot 8^1 + 3 \cdot 8^0 + 2 \cdot 8^{-1} = 113,2_8. \end{aligned}$$

Рассмотрим перевод в  $D_{16}$ . Разложим целую часть на составляющие, пользуясь числами из столбца  $16^X$ , а дробную часть – из столбца  $16^{-X}$ :

$$75_{10} = 16 + 16 + 16 + 16 + 11 = 4 \cdot 16 + 11 = 4 \cdot 16^1 + 11 \cdot 16^0 = 4B_{16}.$$

Здесь набор слагаемых с коэффициентами, есть не что иное, как полиномиальная форма записи шестнадцатеричного числа  $4B_{16}$ .

Дробная часть:  $0,25_{10} = 0,0625 \cdot 4 = 4 \cdot 16^{-1} = 0,4_{16}$ .

Таким образом,

$$75,25_{10} = 4 \cdot 16^1 + 11 \cdot 16^0 + 4 \cdot 16^{-1} = 4B,4_{16}.$$

Видно, что перевод десятичного числа в недесятичное, выполненное методом деления и умножения на основание перево-

димой системы (п. 2.6) совпадает с результатом по методу разложения числа на составляющие по степеням основания (п. 2.7).

## 2.8 Взаимный перевод чисел между $D_2$ , $D_8$ и $D_{16}$

Здесь вообще не упоминается десятичная система счисления. Речь идёт о взаимном переводе чисел *только между системами*  $D_2$ ,  $D_8$  и  $D_{16}$ . Известно, что  $2 = 2^1$ ;  $8 = 2^3$ ;  $16 = 2^4$ . Это показывает, что 8 и 16 можно выразить через 2, т. е. через  $D_2$ . Обратимся к таблице взаимного соответствия чисел (таблица 2.1) и запишем базис  $D_8$  и  $D_{16}$  через  $D_2$ , выделяя для чисел, соответственно, *три* и *четыре* позиции. Напоминаем, что если в позиции нет значащей цифры, то она указывается нулём (таблица 2.3).

Таблица 2.3 – Выражение базиса  $D_8$  и  $D_{16}$  через  $D_2$

$D_2$	$D_8$		$D_{16}$	
<b>0</b>	<b>0</b>	000	<b>0</b>	0000
<b>1</b>	<b>1</b>	001	<b>1</b>	0001
10	<b>2</b>	010	<b>2</b>	0010
11	<b>3</b>	011	<b>3</b>	0011
100	<b>4</b>	100	<b>4</b>	0100
101	<b>5</b>	101	<b>5</b>	0101
110	<b>6</b>	110	<b>6</b>	0110
111	<b>7</b>	111	<b>7</b>	0111
1000			<b>8</b>	1000
1001			<b>9</b>	1001
1010			<b>A</b>	1010
1011			<b>B</b>	1011
1100			<b>C</b>	1100
1101			<b>D</b>	1101
1110			<b>E</b>	1110
1111			<b>F</b>	1111

В таблице видно, что цифру базиса  $D_8$  можно выразить через *тройку* двоичных чисел, а цифру базиса  $D_{16}$  можно выразить через *четверку* двоичных чисел. Тогда, если задано число  $D_8$  из нескольких цифр, то каждую цифру  $D_8$  можно заменить *тройкой* цифр  $D_2$ . Также, если задано число  $D_{16}$  из нескольких цифр, то каждую цифру  $D_{16}$  можно заменить *четвёркой* цифр  $D_2$ .

Аналогично, если задано число  $D_2$ , то нужно выделить по три позиции для  $D_8$  или по четыре позиции для  $D_{16}$  и каждую выделенную двоичную группу заменить соответствующей цифрой базиса. Отметим, что группировка по три или четыре выполняется от запятой. Например:

$$\begin{array}{ccccccc} 2 & 5 & 1 & 1 & 4 & 4 & 1_{(8)} \\ \underbrace{\quad} & \underbrace{\quad} & \underbrace{\quad} & \underbrace{\quad} & \underbrace{\quad} & \underbrace{\quad} & \underbrace{\quad} \\ \underbrace{101} & \underbrace{010} & \underbrace{010} & \underbrace{010} & \underbrace{100} & \underbrace{100} & \underbrace{01}_2 \end{array}$$

$$5 \quad 4 \quad 9 \quad 9 \quad 0 \quad 8_{(16)}$$

$$10101001001,100100001_2 = 2511,441_8 = 549,908_{16}.$$

Пример перевода числа из  $D_8$  в  $D_2$ :

$$\begin{array}{ccccccc} 1 & 6 & 7 & , & 5 & 4_{(8)} \\ \underbrace{\quad} & \underbrace{\quad} & \underbrace{\quad} & & \underbrace{\quad} & \underbrace{\quad} & \\ 001 & 110 & 111 & , & 101 & 100_{(2)} \end{array} \quad 167,54_8 = 1110111,1011_2.$$

Пример перевода числа из  $D_{16}$  в  $D_2$ :

$$A6F,D4_{16} = 101001101111,110101_2.$$

$$\begin{array}{ccccccc} A & 6 & F & , & D & 4_{(16)} \\ \underbrace{\quad} & \underbrace{\quad} & \underbrace{\quad} & & \underbrace{\quad} & \underbrace{\quad} & \\ 101 & 001 & 101 & 111 & , & 110 & 10100_{(2)} \end{array}$$

При переводе числа из  $D_8$  в  $D_{16}$  или, наоборот, из  $D_{16}$  в  $D_8$  следует выполнить промежуточный перевод в  $D_2$ . Например,

$$549,908_{16} = 10101001001,100100001_2 = 2511,441_8.$$

$$\begin{array}{ccccccc} 5 & 4 & 9 & , & 9 & 0 & 8_{(16)} \\ \underbrace{01010100} & \underbrace{1001} & \underbrace{1000} & , & \underbrace{1001} & \underbrace{0000} & \underbrace{1000}_{(2)} \\ 2 & 5 & 1 & 1 & , & 4 & 4 & 1 & 0_{(8)} \end{array}$$



## 2.9 Используемые системы счисления в компьютерной технологии

Выше в п. 2.3 было сказано, что используются четыре вида систем счисления:  $D_{10}$ ,  $D_2$ ,  $D_8$  и  $D_{16}$ .

Привычная для нас *десятичная* система используется для записи числовых данных при постановке задачи и для ввода исходных данных в компьютер и вывода результатов расчета.

*Двоичная* система используется для кодирования любой информации в компьютере: числовой, текстовой, графической, звуковой, а также компьютерной программы. Компьютер обрабатывает информацию, представленную только двоичным кодом (принцип двоичного кодирования по Джону Нейману).

Как видно из приведенных выше примеров двоичных чисел, они (двоичные числа) занимают много позиций в записи. Поэтому при обсуждении некоторых параметров в компьютерной технологии, принято использовать более короткий эквивалент двоичного кода. Для этого наиболее удобными являются *восьмеричная* и *шестнадцатеричная* системы счисления, потому что взаимный перевод чисел между  $D_2$ ,  $D_8$  и  $D_{16}$  выполняется очень просто (см. п. 2.7).

## 2.10 Вопросы и задания

1. Какие цифры составляют алфавит (базис)  $D_{10}$  ?
2. Какие цифры составляют алфавит (базис)  $D_2$  ?
3. Какие цифры составляют алфавит (базис)  $D_8$  ?
4. Какие цифры составляют алфавит (базис)  $D_{16}$  ?
5. Сколько существует различных двузначных двоичных чисел?
6. Сколько существует различных восьмизначных двоичных чисел?
7. Сколько существует различных двузначных восьмеричных чисел?
8. Переведите десятичные числа 27, 35, 54, 127, 128 в  $D_2$ .
9. Переведите десятичные числа 27, 35, 54, 127, 128 в  $D_8$ .
10. Переведите десятичные числа 27, 35, 54, 127, 128 в  $D_{16}$ .

11. Переведите двоичные числа 100101, 101100, 111001, 1010101 в десятичные.
12. Переведите двоичные числа 100101, 101100, 111001, 1010101 в восьмеричные.
13. Переведите двоичные числа 100101, 101100, 111001, 1010101 в шестнадцатеричные.
14. Как по записи числа в  $D_2$  определить, что оно четное?
15. Переведите в  $D_2$  и  $D_8$  числа:  $8F4A_{16}$ ,  $CA4F_{16}$ ,  $8245_{16}$ .
16. Переведите в  $D_2$  и  $D_{16}$  числа:  $5342_8$ ,  $1720_8$ ,  $7061_8$ .

### 3 ДВОИЧНАЯ АРИФМЕТИКА

В этой главе будет показано выполнение арифметических операций (сложение, умножение, вычитание и деление) в двоичной системе счисления. Отметим, что правила двоичной арифметики, такие же как в обычной арифметике с десятичной системой счисления. В двоичной системе, также как и в десятичной системе, сложение, вычитание и умножение выполняется столбиком, а деление уголком. К тому же, и это самое главное, двоичная система в 25 раз проще, чем десятичная.

Поясним сказанное. В арифметике оперируют с двумя числами  $a$  и  $b$ . Множество возможных значений базисных цифр таково:

$\{a,b\}_2 = \{ (0,0); (0,1); (1,0); (1,1) \}$  – всего 4 варианта.

$\{a,b\}_{10} = \{ (0,0); (0,1); \dots (8,9); (9,9) \}$  – 100 вариантов.

#### 3.1 Сложение

Таблица сложения двоичных чисел очень простая (проще не бывает). Сложение выполняется столбиком. Каждая цифра занимает одну позицию. Покажем сложение столбиком.

$0 + 0 = 0;$	0	0	1	1	10	11
$0 + 1 = 1;$	+0	+1	+0	+1	+ 1	+ 1
$1 + 0 = 1;$	--	--	--	--	--	--
$1 + 1 = 10_2$ (два)	0	1	1	10	11	100

Рассмотрим сложение более «длинных» двоичных чисел подробно по шагам, например  $a + b = 111 + 1$ .

$$\begin{array}{r}
 \begin{array}{c} 2 \ 1 \ 0 \\ 1 \ 1 \ 1 \\ + \ 1 \\ \hline 0 \\ \hline 1 \end{array}
 \end{array}$$

← позиция

} слагаемые

Шаг 1 (позиция 0): Складываем цифры в позиции. Получаем  $1+1=10$ . 0 пишем в этой позиции, а 1 «в уме» в следующей позиции (под пунктирной чертой).

Шаг 2 (позиция 1): Складываем цифры в 1-й позиции. Имеем вверху 1 и «в уме» 1, получаем  $1+1=10$ . 0 пишем, а 1 «в уме» в следующей позиции (под пунктирной чертой).

$$\begin{array}{r} 2\ 1\ 0 \\ 1\ 1\ 1 \\ +\ 1 \\ \hline 0 \\ \text{---} \\ 1 \end{array}$$

было

стало

$$\begin{array}{r} 2\ 1\ 0 \\ 1\ 1\ 1 \\ +\ 1 \\ \hline 0\ 0 \\ \text{---} \\ 1 \end{array}$$

Шаг 3 (позиция 2): Складываем цифры во 2-й позиции. Имеем вверху 1 и «в уме» 1, получаем  $1+1=10$ . Эти цифры пишем в 3-й и 2-й позициях.

$$\begin{array}{r} 2\ 1\ 0 \\ 1\ 1\ 1 \\ +\ 1 \\ \hline 0\ 0 \\ \text{---} \\ 1 \end{array}$$

было

стало

$$\begin{array}{r} 3\ 2\ 1\ 0 \\ 1\ 1\ 1 \\ +\ 1 \\ \hline 1\ 0\ 0\ 0 \end{array}$$

Результат сложения:  $111 + 1 = 1000$ .

Рассмотрим подробно по шагам ещё один пример:  $111+11$ .

$$\begin{array}{r} 2\ 1\ 0 \\ 1\ 1\ 1 \\ +\ 1\ 1 \\ \hline 0 \\ \text{---} \\ 1 \end{array}$$

← позиция  
} слагаемые

Шаг 1 (позиция 0): Складываем цифры в позиции. Получаем  $1+1=10$ . 0 пишем в этой позиции, а 1 «в уме» в следующей позиции (под пунктирной чертой).

Шаг 2 (позиция 1): Складываем цифры в 1-й позиции. Имеем вверху  $1 + 1$  и «в уме» 1, получаем  $1+1+1=11$ . Младшую цифру 1 пишем в 1-й позиции, а старшую цифру 1 запоминаем «в уме» в следующей 2-й позиции (под пунктирной чертой).

$$\begin{array}{r} 2\ 1\ 0 \\ 1\ 1\ 1 \\ +\ 1\ 1 \\ \hline 0 \\ \text{---} \\ 1 \end{array}$$

было

стало

$$\begin{array}{r} 2\ 1\ 0 \\ 1\ 1\ 1 \\ +\ 1\ 1 \\ \hline 1\ 0 \\ \text{---} \\ 1 \end{array}$$

Шаг 3 (позиция 2): Складываем цифры во 2-й позиции. Имеем вверху 1 и «в уме» 1, получаем  $1+1=10$ . Эти цифры пишем в 3-й и 2-й позициях.

$$\begin{array}{r} 2\ 1\ 0 \\ \underline{1\ 1\ 1} \\ + 1\ 1 \\ \hline \underline{\underline{1\ 0}} \\ 1 \end{array}$$

было

стало

$$\begin{array}{r} 3\ 2\ 1\ 0 \\ \underline{1\ 1\ 1} \\ + 1\ 1 \\ \hline 1\ 0\ 1\ 0 \end{array}$$

Результат сложения:  $111 + 11 = 1010$ .

Ещё примеры:

$$\begin{array}{r} 1\ 0\ 1\ 1\ 1\ 1\ 1\ 0\ 1\ 1 \\ + 1\ 1\ 0\ 0\ 0\ 1\ 1\ 1\ 1 \\ \hline 1\ 1\ 1\ 1\ 0\ 0\ 0\ 1\ 0\ 1\ 0 \end{array}$$

$$\begin{array}{r} 1\ 1\ 1\ 1\ 0\ 1\ 1\ 1\ 1\ 0\ 1\ 1\ 1 \\ + 1\ 1\ 1\ 1\ 1\ 0\ 1\ 1\ 1\ 1\ 1 \\ \hline 1\ 0\ 0\ 1\ 1\ 0\ 1\ 1\ 0\ 1\ 0\ 1\ 1\ 0 \end{array}$$

Обратим внимание, что если несколько позиций подряд заполнены 1 и затем к числу прибавляется 1, то в результате получается число с 1 в новой позиции (дополнительно слева) и подряд 0 в остальных позициях:

$$\begin{array}{r} 1\ 1 \\ + 1 \\ \hline 1\ 0\ 0 \end{array} \quad \begin{array}{r} 1\ 1\ 1 \\ + 1 \\ \hline 1\ 0\ 0\ 0 \end{array} \quad \begin{array}{r} 1\ 1\ 1\ 1 \\ + 1 \\ \hline 1\ 0\ 0\ 0\ 0 \end{array} \quad \begin{array}{r} 1\ 1\ 1\ 1\ 1 \\ + 1 \\ \hline 1\ 0\ 0\ 0\ 0\ 0 \end{array} \quad \begin{array}{r} 1\ 1\ 1\ 1\ 1\ 1 \\ + 1 \\ \hline 1\ 0\ 0\ 0\ 0\ 0\ 0 \end{array}$$

Или, если число состоит из 1 с несколькими 0, то такое число равно сумме инверсии всех цифр плюс 1. Инверсия – это замена значения на противоположное, в данном случае замена 0 на 1 и 1 на 0. Например,  $100 = 11+1$ ;  $1000 = 111+1$ ;  $10000 = 1111+1$  и т. д.

## 3.2 Умножение

Таблица умножения двоичных чисел очень простая (проще не бывает). Умножение выполняется столбиком.

$0 * 0 = 0$ ;  
 $0 * 1 = 0$ ;  
 $1 * 0 = 0$ ;  
 $1 * 1 = 1$ ;

Первое число умножается на каждую цифру второго числа и затем выполняется сложение с соблюдением позиции.

Покажем на примере процедуру умножения по шагам. Слева показаны исходные числа и результат умножения. Справа показан шаг 1 (позиция 1).

$$\begin{array}{r}
 110111 * 1111 \\
 \hline
 110111 \\
 110111 \\
 + 110111 \\
 110111 \\
 \hline
 1100111001
 \end{array}$$

$$\begin{array}{r}
 9876543210 \leftarrow \text{позиции} \\
 \hline
 110111 \\
 110111 \\
 + 110111 \\
 110111 \\
 \hline
 01 \\
 \hline
 1
 \end{array}
 \quad \left. \begin{array}{l} \\ \\ \\ \end{array} \right\} \text{слагаемые}$$

Шаг 2 (позиция 2). Складываем цифры слагаемых и «в уме» под пунктирной чертой. Получаем  $1+1+1+1=100$ . Младшую цифру 0 оставляем во 2-й позиции, а цифры 10 пишем ниже «в уме», при этом 1 будет в 4-й позиции.

$$\begin{array}{r}
 9876543\mathbf{2}10 \\
 \hline
 110\mathbf{1}11 \\
 1101\mathbf{1}1 \\
 + 1101\mathbf{1}1 \\
 110111 \\
 \hline
 01 \\
 \hline
 1
 \end{array}$$

было

стало

$$\begin{array}{r}
 9876543\mathbf{2}10 \\
 \hline
 110111 \\
 110111 \\
 + 110111 \\
 110111 \\
 \hline
 001 \\
 \hline
 10
 \end{array}$$

Шаг 3 (позиция 3). Складываем цифры слагаемых и «в уме». Получаем  $0+1+1+1+0=11$ . Младшую цифру 1 оставляем в 3-й позиции, а старшую цифру 1 пишем ниже «в уме» в 4-й позиции.

$$\begin{array}{r}
 987654\mathbf{3}210 \\
 \hline
 11\mathbf{0}111 \\
 110\mathbf{1}11 \\
 + 1101\mathbf{1}1 \\
 1101\mathbf{1}1 \\
 \hline
 001 \\
 \hline
 10
 \end{array}$$

было

стало

$$\begin{array}{r}
 987654\mathbf{3}210 \\
 \hline
 110111 \\
 110111 \\
 + 110111 \\
 110111 \\
 \hline
 1001 \\
 \hline
 1 \\
 1
 \end{array}$$

Шаг 4 (позиция 4). Складываем цифры слагаемых и «в уме». Получаем  $1+0+1+1+1+1=101$ . Младшую цифру 1 оставляем в 4-й позиции, а цифры 10 пишем ниже «в уме» в 6-й и 5-й позициях.

$$\begin{array}{r}
 98765\mathbf{4}3210 \\
 \hline
 1\mathbf{1}0111 \\
 11\mathbf{0}111 \\
 + 110\mathbf{1}11 \\
 1101\mathbf{1}1 \\
 \hline
 1001 \\
 \hline
 \mathbf{1} \\
 \mathbf{1}
 \end{array}$$

было

стало

$$\begin{array}{r}
 98765\mathbf{4}3210 \\
 \hline
 110111 \\
 110111 \\
 + 110111 \\
 110111 \\
 \hline
 \mathbf{1}1001 \\
 \hline
 \mathbf{10}
 \end{array}$$

Шаг 5 (позиция 5). Складываем цифры слагаемых и «в уме». Получаем  $1+1+0+1+0=11$ . Младшую цифру 1 оставляем в 5-й позиции, а старшую цифру 1 пишем ниже «в уме» в 6-й позиции.

$$\begin{array}{r}
 9876\mathbf{5}43210 \\
 \hline
 1\mathbf{1}0111 \\
 1\mathbf{1}0111 \\
 + 11\mathbf{0}111 \\
 110\mathbf{1}11 \\
 \hline
 11001 \\
 \hline
 \mathbf{10}
 \end{array}$$

было

стало

$$\begin{array}{r}
 9876\mathbf{5}43210 \\
 \hline
 110111 \\
 110111 \\
 + 110111 \\
 110111 \\
 \hline
 \mathbf{1}11001 \\
 \hline
 \mathbf{1}
 \end{array}$$

Шаг 6 (позиция 6). Складываем цифры слагаемых и «в уме». Получаем  $1+1+0+1+1=100$ . Младшую цифру 0 оставляем в 6-й позиции, а цифры 10 пишем ниже «в уме» в 8-й и 7-й позициях.

$$\begin{array}{r}
 987\mathbf{6}543210 \\
 \hline
 110111 \\
 \mathbf{1}10111 \\
 + 1\mathbf{1}0111 \\
 11\mathbf{0}111 \\
 \hline
 111001 \\
 \hline
 \mathbf{1} \\
 \mathbf{1}
 \end{array}$$

было

стало

$$\begin{array}{r}
 987\mathbf{6}543210 \\
 \hline
 110111 \\
 110111 \\
 + 110111 \\
 110111 \\
 \hline
 \mathbf{0}111001 \\
 \hline
 \mathbf{10}
 \end{array}$$

Шаг 7 (позиция 7). Складываем цифры слагаемых и «в уме». Получаем  $1+1+0=10$ . Цифру 0 оставляем в 7-й позиции, а старшую цифру 1 пишем ниже «в уме» в 8-й позиции.

$$\begin{array}{r}
 9876543210 \\
 \hline
 \phantom{98}110111 \\
 \phantom{98}110111 \\
 + \phantom{98}110111 \\
 \hline
 \phantom{98}110111 \\
 \hline
 \phantom{98}0111001 \\
 \hline
 10
 \end{array}$$

было

стало

$$\begin{array}{r}
 9876543210 \\
 \hline
 \phantom{98}110111 \\
 \phantom{98}110111 \\
 + \phantom{98}110111 \\
 \hline
 \phantom{98}110111 \\
 \hline
 \phantom{98}00111001 \\
 \hline
 1 \\
 \mathbf{1}
 \end{array}$$

Шаг 8 (позиция 8). Складываем цифры слагаемых и «в уме». Получаем  $1+1+1=11$ . Цифры 11 пишем в 9-й и 8-й позициях.

$$\begin{array}{r}
 9876543210 \\
 \hline
 \phantom{98}110111 \\
 \phantom{98}110111 \\
 + \phantom{98}110111 \\
 \hline
 \phantom{98}110111 \\
 \hline
 \phantom{98}00111001 \\
 \hline
 1 \\
 1
 \end{array}$$

было

стало

$$\begin{array}{r}
 9876543210 \\
 \hline
 \phantom{98}110111 \\
 \phantom{98}110111 \\
 + \phantom{98}110111 \\
 \hline
 \phantom{98}110111 \\
 \hline
 1100111001
 \end{array}$$

В итоге получаем следующий результат при умножении:

$$110111 * 1111 = 1100111001.$$

Таким образом, в двоичной системе счисления операция умножения превращается в сложение со сдвигами первого числа; количество сдвигов равно количеству единиц второго числа.

### 3.3 Вычитание

Следует сразу оговориться, что при вычитании первое число всегда равно или больше второго числа. *Нельзя вычитать бóльшее число из мёньшего числа.*



Таблица вычитания двоичных чисел следующая.

$$\begin{array}{rcl}
 0 - 0 = 0; & \text{Вычитание выполняется столбиком.} \\
 1 - 0 = 1; & 0 & 1 & 1 & 10 & 11 & 100 \\
 1 - 1 = 0; & \frac{-0}{0} & \frac{-0}{1} & \frac{-1}{0} & \frac{-1}{1} & \frac{-1}{1} & \frac{-1}{11} \\
 10 - 1 = 1; & & & & & & 
 \end{array}$$

Вычитание выполняется справа налево по позициям.

При выполнении вычитания типа  $100 - 1 = 11$  вспомним, что  $100 = 11 + 1$  (см. последний абзац п. 3.1). Рассмотрим пример.

9 8 7 6 5 4 3 2 1 0 ← позиция;  
 1 1 1 1 0 0 0 1 0 1 0  
 - 1 1 0 0 0 1 1 1 1  
 -----  
 было стало

Шаг 1 (позиция 0). Из 0 нельзя вычесть 1, поэтому зачеркиваем две исходные цифры (до ближайшей 1) и заменяем их на 01 и 1. Теперь в нулевой позиции имеем  $1+1-1=1$ .

Шаг 2 (позиция 1). Из 0 нельзя вычесть 1, поэтому зачеркиваем три цифры 100 (до ближайшей 1) и заменяем их на 011 и 1. Теперь в 1-й позиции имеем  $1+1-1=1$ , а во 2-й позиции  $1-1=0$ .

было

стало

Шаг 3 (позиция 3). Из 0 нельзя вычесть 1, поэтому зачеркиваем пять цифр 10000 (до ближайшей 1) и заменяем их на 01111

и 1. Теперь в 3-й позиции имеем  $1+1-1=1$ , в 4-й позиции  $1-0=1$ , в 5-й позиции  $1-0=1$ , в 6-й позиции  $1-0=1$ .

$$\begin{array}{r}
 9\ 8\ 7\ 6\ 5\ 4\ \mathbf{3}\ 2\ 1\ 0 \\
 \hline
 \phantom{11110000}11 \\
 11110000\mathbf{0}111 \\
 -\ 11000\mathbf{1}111 \\
 \hline
 \phantom{11110000}011
 \end{array}$$

было

стало

$$\begin{array}{r}
 9\ 8\ 7\ \mathbf{6}\ 5\ 4\ \mathbf{3}\ 2\ 1\ 0 \\
 \hline
 \phantom{11110000}1 \\
 \phantom{11110000}\mathbf{0}1111\ 11 \\
 111\mathbf{1}0000111 \\
 -\ 11000\mathbf{1}111 \\
 \hline
 \phantom{11110000}1111011
 \end{array}$$

Шаг 4 (позиция 7). Из 0 нельзя вычесть 1, поэтому зачеркиваем две цифры 10 (до ближайшей 1) и заменяем их на 01 и 1. Теперь в 7-й позиции имеем  $1+1-1=1$ .

$$\begin{array}{r}
 9\ 8\ \mathbf{7}\ 6\ 5\ 4\ 3\ 2\ 1\ 0 \\
 \hline
 \phantom{1110111111}1\ 11 \\
 111\mathbf{0}1111111 \\
 -\ 1\mathbf{1}0001111 \\
 \hline
 \phantom{11101111}1111011
 \end{array}$$

было

стало

$$\begin{array}{r}
 9\ 8\ \mathbf{7}\ 6\ 5\ 4\ 3\ 2\ 1\ 0 \\
 \hline
 \phantom{1110111111}1 \\
 \phantom{1110111111}\mathbf{0}1\phantom{11111111}1\ 11 \\
 111\mathbf{1}0111111 \\
 -\ 1\mathbf{1}0001111 \\
 \hline
 \phantom{11101111}11111011
 \end{array}$$

Шаг 5 (позиция 8). Из 0 нельзя вычесть 1, поэтому зачеркиваем две цифры 10 (до ближайшей 1) и заменяем их на 01 и 1. Теперь в 8-й позиции имеем  $1+1-1=1$ . В 9-й позиции будет 0, в 10-й позиции будет 1.

$$\begin{array}{r}
 9\ \mathbf{8}\ 7\ 6\ 5\ 4\ 3\ 2\ 1\ 0 \\
 \hline
 \phantom{1101111111}1\phantom{11111111}1\ 11 \\
 11\mathbf{0}1111111 \\
 -\ \mathbf{1}10001111 \\
 \hline
 \phantom{11011111}11111011
 \end{array}$$

было

стало

$$\begin{array}{r}
 9\ 8\ 7\ 6\ 5\ 4\ 3\ 2\ 1\ 0 \\
 \hline
 \phantom{1101111111}1 \\
 \phantom{1101111111}\mathbf{0}11\phantom{11111111}1\ 11 \\
 11\mathbf{1}0111111 \\
 -\ \mathbf{1}10001111 \\
 \hline
 \phantom{11011111}1011111011
 \end{array}$$

Таким образом, вычитание двух заданных чисел дает следующий результат:

$$\begin{array}{r}
 11110001010 \\
 -\ 110001111 \\
 \hline
 10111111011
 \end{array}$$

### 3.4 Деление

При операции деления будем считать, что делимое больше делителя, т.е. результат всегда больше 1. Операция деления в двоичной системе такая же, как в обычной десятичной системе.

Как правило, деление выполняется уголком. Рассмотрим пример деления двух двоичных чисел.

$$101000101 / 1101 = 11001$$

Отметим последовательность действий по шагам.

1) На делимом слева отсчитывается такое количество цифр, чтобы их значение превосходило делитель. В данном примере пять цифр – это 10100. При этом результат приобретает цифру 1. Затем делитель вычитается из 10100. Получаем остаток, равный 111.

$$\begin{array}{r}
 101000101 \mid 1101 \\
 -1101 \phantom{000000} \\
 \hline
 1110 \phantom{00000} \\
 -1101 \phantom{0000} \\
 \hline
 1101 \phantom{000} \\
 -1101 \phantom{00} \\
 \hline
 0
 \end{array}$$

2) К остатку приписывается очередная цифра делимого. В данном случае 0. Получаем 1110. Значение этих цифр больше делителя, поэтому результат опять приобретает цифру 1 и делитель вычитается из этих цифр. Получаем остаток, равный 1.

3) К остатку приписывается очередная цифра делимого. В данном случае 1. Получаем 11. Значение этих цифр меньше делителя, поэтому результат приобретает цифру 0.

4) Затем опять к остатку приписывается очередная цифра делимого. В данном случае 0. Получаем 110. Значение этих цифр меньше делителя, поэтому результат опять приобретает цифру 0.

5) Опять к остатку приписывается очередная цифра делимого. В данном случае 1. Получаем 1101. Значение этих цифр равно делителю, поэтому результат приобретает 1 и делитель вычитается из этих цифр. Получаем остаток, равный 0.

В данном случае делимое разделилось на делитель точно.

Если после использования всех цифр делимого получился остаток, отличный от 0, то результат деления будет иметь дробную часть. Дробная часть получается продолжением деления путем присоединения нулей к остатку. При этом целая часть отделяется запятой от дробной части.

### 3.5 Арифметика с дробной частью

Выполнение арифметических операций двоичных чисел с дробной частью такое же, как с десятичными числами. Для примера рассмотрим два числа: 10,101 и 1,011.

#### 3.5.1 Сложение

$$\begin{array}{r}
 \begin{array}{cccc}
 1 & 0 & -1 & -2 & -3 \\
 \hline
 10, & 101 & & & \\
 +1, & 011 & & & \\
 \hline
 & & 0 & & \\
 \hline
 & & & & 1
 \end{array}
 \end{array}$$

← позиция  
слагаемые

Шаг 1 (позиция «-3»): Складываем цифры в позиции. Получаем  $1+1=10$ . 0 пишем в этой позиции, а 1 «в уме» в следующей позиции (под пунктирной чертой).

Шаг 2 (позиция «-2»): Складываем цифры в позиции «-2». Имеем вверху 1 и «в уме» 1, получаем  $1+1=10$ . 0 пишем, а 1 «в уме» в следующей позиции (под пунктирной чертой).

$$\begin{array}{r}
 \begin{array}{cccc}
 1 & 0 & -1 & -2 & -3 \\
 \hline
 10, & 101 & & & \\
 +1, & 011 & & & \\
 \hline
 & & 0 & & \\
 \hline
 & & & & 1
 \end{array}
 \end{array}$$

было

стало

$$\begin{array}{r}
 \begin{array}{cccc}
 1 & 0 & -1 & -2 & -3 \\
 \hline
 10, & 101 & & & \\
 +1, & 011 & & & \\
 \hline
 & & 00 & & \\
 \hline
 & & & & 1
 \end{array}
 \end{array}$$

Шаг 3 (позиция «-1»): Складываем цифры в позиции «-1». Имеем вверху 1 и «в уме» 1, получаем  $1+1=10$ . 0 пишем, а 1 «в уме» в следующей позиции (под пунктирной чертой).

$$\begin{array}{r}
 \begin{array}{cccc}
 1 & 0 & -1 & -2 & -3 \\
 \hline
 10, & 101 & & & \\
 +1, & 011 & & & \\
 \hline
 & & 00 & & \\
 \hline
 & & & & 1
 \end{array}
 \end{array}$$

было

стало

$$\begin{array}{r}
 \begin{array}{cccc}
 1 & 0 & -1 & -2 & -3 \\
 \hline
 10, & 101 & & & \\
 +1, & 011 & & & \\
 \hline
 & & 000 & & \\
 \hline
 & & & & 1
 \end{array}
 \end{array}$$

Шаг 4 (позиция 0): Складываем цифры в позиции 0. Имеем вверху 1 и «в уме» 1, получаем  $1+1=10$ . 0 пишем, а 1 «в уме» в следующей позиции (под пунктирной чертой).

$$\begin{array}{r} 1\ 0\ -1\ -2\ -3 \\ \hline 10,101 \\ +1,011 \\ \hline \text{---}000 \\ \text{---}1 \end{array}$$

было

стало

$$\begin{array}{r} 1\ 0\ -1\ -2\ -3 \\ \hline 10,101 \\ +1,011 \\ \hline \text{---}0,000 \\ \text{---}1 \end{array}$$

Шаг 5 (позиция 1): Складываем цифры в позиции 1. Имеем вверху 1 и «в уме» 1, получаем  $1+1=10$ . Эти цифры пишем под 2-й и 1-й позициями.

$$\begin{array}{r} 1\ 0\ -1\ -2\ -3 \\ \hline \mathbf{1}0,101 \\ +1,011 \\ \hline \text{---}0,000 \\ \text{---}1 \end{array}$$

было

стало

$$\begin{array}{r} 2\ 1\ 0\ -1\ -2\ -3 \\ \hline 10,101 \\ +1,011 \\ \hline 100,000 \end{array}$$

Таким образом, сложение  $10,101_2 + 1,011_2 = 100,000_2 = 100_2$ ;

Проверим полученный результат, переводя двоичные числа в десятичную систему.

10-1-2-3 ← позиции

$$10,101_2 = 1 \cdot 2^1 + 1 \cdot 2^{-1} + 1 \cdot 2^{-3} = 2 + 0,5 + 0,125 = 2,625_{10};$$

$$1,011_2 = 1 \cdot 2^0 + 1 \cdot 2^{-2} + 1 \cdot 2^{-3} = 1 + 0,25 + 0,125 = 1,375_{10};$$

$$100_2 = 1 \cdot 2^2 = 4_{10}; \quad 2,625_{10} + 1,375_{10} = 4_{10}.$$

### 3.5.2 Умножение

При умножении двух чисел  $10,101_2 * 1,011_2$  имеем по три цифры в дробной части. Умножение выполняем без запятых, затем в результате (в ответе) поставим запятую, перед 6-ю цифрами дробной части.

Шаг 1 (позиции 0...3). В позициях 0,1,2 имеем по цифре 1. В позиции 3 имеем  $1+1=10$ . Цифру 0 пишем в позиции 3, а цифру 1 пишем «в уме» под пунктирной чертой в позиции 4.

$$\begin{array}{r}
 10101 * 1011 \\
 \hline
 10101 \\
 10101 \\
 + 00000 \\
 \hline
 10101 \\
 \hline
 11100111
 \end{array}$$

$$\begin{array}{r}
 7\ 6\ 5\ \mathbf{4}\ 3\ 2\ 1\ 0 \leftarrow \text{позиции} \\
 \hline
 10101 \\
 + 10101 \\
 10101 \\
 \hline
 0111 \\
 \hline
 \mathbf{1}
 \end{array}
 \quad \left. \begin{array}{l} \\ \\ \end{array} \right\} \text{слагаемые}$$

Шаг 2 (позиция 4). Складываем цифры слагаемых и «в уме» под пунктирной чертой. Получаем  $1+0+0+1=10$ . Цифру 0 оставляем во 4-й позиции, а цифру 1 пишем ниже «в уме» в 5-й позиции.

$$\begin{array}{r}
 7\ 6\ 5\ \mathbf{4}\ 3\ 2\ 1\ 0 \\
 \hline
 10101 \\
 + 10101 \\
 10101 \\
 \hline
 0111 \\
 \hline
 \mathbf{1}
 \end{array}$$

было

стало

$$\begin{array}{r}
 7\ 6\ \mathbf{5}\ 4\ 3\ 2\ 1\ 0 \\
 \hline
 10101 \\
 + 10101 \\
 10101 \\
 \hline
 00111 \\
 \hline
 \mathbf{1}
 \end{array}$$

Шаг 3 (позиция 5). Складываем цифры слагаемых и «в уме». Получаем  $1+1+1=11$ . Младшую цифру 1 оставляем в 5-й позиции, а старшую цифру 1 пишем ниже «в уме» в 6-й позиции.

$$\begin{array}{r}
 7\ 6\ \mathbf{5}\ 4\ 3\ 2\ 1\ 0 \\
 \hline
 10101 \\
 + 10101 \\
 10101 \\
 \hline
 00111 \\
 \hline
 \mathbf{1}
 \end{array}$$

было

стало

$$\begin{array}{r}
 7\ \mathbf{6}\ 5\ 4\ 3\ 2\ 1\ 0 \\
 \hline
 10101 \\
 + 10101 \\
 10101 \\
 \hline
 100111 \\
 \hline
 \mathbf{1}
 \end{array}$$

Шаг 4 (позиции 6 и 7). Складываем цифры в позициях. Получаем 11. Младшую цифру 1 оставляем в 6-й позиции, а старшую цифру 1 пишем 7-й позиции.

$$\begin{array}{r}
 \mathbf{7}\ \mathbf{6}\ 5\ 4\ 3\ 2\ 1\ 0 \\
 \hline
 10101 \\
 + 10101 \\
 10101 \\
 \hline
 100111 \\
 \hline
 \mathbf{1}
 \end{array}$$

было

стало

$$\begin{array}{r}
 7\ 6\ 5\ 4\ 3\ 2\ 1\ 0 \\
 \hline
 10101 \\
 + 10101 \\
 10101 \\
 \hline
 11100111
 \end{array}$$

Таким образом, в полученных цифрах 11100111 ставим запятую перед 6-ю цифрами справа, и результатом умножения является

$$10,101_2 * 1,011_2 = 11,100111_2.$$

Проверим полученный результат, переводя двоичные числа в десятичную систему.

$$\begin{aligned}
& 10-1-2-3-4-5-6 \leftarrow \text{ПОЗИЦИИ} \\
& 11,100111_2 = 1 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-1} + 1 \cdot 2^{-4} + 1 \cdot 2^{-5} + 1 \cdot 2^{-6} = \\
& = 2 + 1 + 1/2 + 1/16 + 1/32 + 1/64 = \\
& = 3 + 0,5 + 0,0625 + 0,03125 + 0,015625 = 3,609375_{10};
\end{aligned}$$

$$2,625_{10} * 1,375_{10} = 3,609,375_{10}.$$

### 3.5.3 Вычитание

Шаг 1. Позиция «-3»: имеем  $1-1=0$ . Позиция «-2»: из 0 нельзя вычесть 1, поэтому зачеркиваем две исходные цифры (до ближайшей 1) и заменяем их на 01 и 1. Теперь в позиции «-2» имеем  $1+1-1=1$ . Позиция «-1»: имеем  $0-0=0$ .

$$\begin{array}{r} \underline{1\ 0\ -1\ -2\ -3} \leftarrow \text{позиция;} \\ 10, 101 \\ - 1, 011 \end{array} \left. \vphantom{\begin{array}{r} 10, 101 \\ - 1, 011 \end{array}} \right\} \begin{array}{l} \text{исходные} \\ \text{цифры;} \end{array}$$

было

стало

$$\begin{array}{r}
 \underline{1\ 0\ \cancel{1}\ \cancel{2}\ \cancel{3}} \leftarrow \text{позиция;} \\
 \phantom{1\ 0\ }\phantom{\cancel{1}\ \cancel{2}\ \cancel{3}} \\
 \phantom{1\ 0\ }\phantom{\cancel{1}\ \cancel{2}\ \cancel{3}} 1 \\
 \phantom{1\ 0\ }\phantom{\cancel{1}\ \cancel{2}\ \cancel{3}} 01 \\
 \hline
 10, \cancel{10}1 \\
 - 1,011 \\
 \hline
 010
 \end{array}
 \left. \vphantom{\begin{array}{r} 10, \cancel{10}1 \\ - 1,011 \end{array}} \right\} \begin{array}{l} \text{исходные} \\ \text{цифры} \end{array}$$

Шаг 2 (позиция 0). Из 0 нельзя вычесть 1, поэтому зачеркиваем две цифры 10 (до ближайшей 1) и заменяем их на 01 и 1. Теперь в позиции 0 имеем  $1+1-1=1$ , а во 1-й позиции имеем 0.

$$\begin{array}{r} 10-1-2-3 \\ \hline 1 \\ \hline 10,011 \\ - 1,011 \\ \hline 010 \end{array}$$

было

стало

$$\begin{array}{r} 10-1-2-3 \\ \hline 1 \\ 01 \quad 1 \\ \hline 10,011 \\ - 1,011 \\ \hline 1,010 \end{array}$$

Таким образом, результатом вычитания является

$$10,101_2 - 1,011_2 = 1,01_2.$$

Проверим полученный результат, переводя двоичные числа в десятичную систему.

0-1-2 ← позиции

$$1,01_2 = 1 \cdot 2^0 + 1 \cdot 2^{-2} = 1 + 0,25 = 1,25_{10};$$

$$2,625_{10} - 1,375_{10} = 1,25_{10}.$$

### 3.5.4 Деление

При делении имеем дробную часть в числителе и знаменателе. Передвигаем запятую параллельно в числителе и знаменателе вправо, чтобы освободиться от дробной части. Это, то же самое, что и умножение числителя и знаменателя на одинаковое число.

$$\frac{10,101}{1,011} = \frac{10101}{1011} \quad \begin{array}{r} 10101 \overline{) 1011} \\ -1011 \phantom{000000000000} \\ \hline 10100 \\ -1011 \phantom{0000000000} \\ \hline 10010 \\ -1011 \phantom{000000000} \\ \hline 1110 \\ -1011 \phantom{00000000} \\ \hline 11 \\ \hline \end{array}$$

и т.д.

Проверим полученный результат, переводя двоичные числа в десятичную систему.

$$1,111010001011_2 = 1 + 1/2 + 1/4 + 1/8 + 1/32 + 1/512 + 1/2048 + \\ + 1/4048 + 1/8096 = 1,9091_{10};$$

$$2,625_{10} / 1,375_{10} = 1,9091_{10}.$$



### 3.6 Замечание об арифметических операциях

Приведенные выше арифметические операции с двоичными числами наглядно показывают, что, во-первых, умножение – есть сложение со сдвигом, во-вторых, деление – есть вычитание со сдвигом. Получается, что основными операциями являются сложение и вычитание.

Математики, рассуждающие с абстрактными понятиями, говорят, что операция вычитания – есть сложение с обратным знаком, т.е.  $a - b = a + (-b)$ . Однако, в реальном мире, в представлении людей древнего и среднего веков, невозможно найти предметы с обратным (отрицательным) знаком, которые при *добавлении* (сложении) к «нормальным» предметам уменьшали бы их количество. Напоминаем, что число – это мера количества. Поэтому, на самом деле, вычитание – это *удаление* из количества «нормальных» предметов некоторого количества таких же «нормальных» предметов. Именно так построена таблица вычитания.

Таким образом, в реальном мире, вычитание – это операция *уменьшения* (удаления) из имеющегося множества некоторого подмножества. При абстрактных рассуждениях допускается утверждать, что вычитание – это операция *добавления* (сложения) в имеющееся множество некоторого «отрицательного» подмножества.

### 3.7 Примеры арифметических операций в $D_8$ и $D_{16}$

Отметим, что арифметические операции в  $D_8$  и  $D_{16}$  выполняются по таким же правилам, как в  $D_{10}$  и  $D_2$ . При этом всегда будем помнить, как пишется число после базиса (см. таблица 2.1).

Рассмотрим несколько примеров. Отметим, что здесь не привлекается на помощь система  $D_{10}$ .

**Примеры в  $D_8$ .** Базис = 0, 1, ..., 7. Затем 10, 11, ..., 17, 20, 21, ...

**Пример 1.** Сложение:  $356_8 + 4662_8 = 5240_8$ .

$$\begin{array}{r}
 \underline{3\ 2\ 1\ 0} \leftarrow \text{позиции} \\
 356_8 \\
 +4662_8 \\
 \hline
 5240 \\
 \hline
 111
 \end{array}$$

*Пояснения:*

Позиция 0)  $(6+2)_8 = (7+1)_8 = 10_8$ . 0 пишем, а 1 в уме в позиции 1.

Позиция 1)  $5+6+1 = (7+1)+4 = 10+4 = 14_8$ . 4 пишем, а 1 в уме в позиции 2.

Позиция 2)  $3+6+1 = (7+1)+2 = 10+2 = 12_8$ . 2 пишем, а 1 в уме в позиции 3.

Позиция 3)  $4+1 = 5$ .

**Пример 2.** Вычитание:  $5240_8 - 356_8 = 4662_8$ ;

$$\begin{array}{r}
 \underline{3\ 2\ 1\ 0} \leftarrow \text{позиции} \\
 5240_8 \\
 -356_8 \\
 \hline
 4662_8
 \end{array}$$

*Пояснения:*

Позиция 0) Из 0 нельзя вычесть 6. Занимаем 1 из левой позиции, было 4, остается 3 в позиции 1, т. е.  $40_8 = (30+10)_8$ . Получаем  $(10-6)_8 = (7+1) - 6 = 2_8$ .

Позиция 1) Было 4, осталось 3. Из 3 нельзя вычесть 5. Занимаем 1 из левой позиции; было 2, остается 1 в позиции 2, т. е.  $23_8 = (10+13)_8$ . Получаем  $(13-5)_8 = 10+3-5 = (7+1)+3-5 = 6_8$ .

Позиция 2) Было 2, остался 1. Из 1 нельзя вычесть 3. Занимаем 1 из левой позиции; было 5, остается 4 в позиции 3, т. е.  $51_8 = (40+11)_8$ . Получаем  $(11-3)_8 = 10+1-3 = (7+1)+1-3 = 6_8$ .

Позиция 3) Было 5, остается 4.

**Пример 3.** Вычитание:  $456_8 - 277_8 = 157_8$ .

$$\begin{array}{r}
 \begin{array}{ccc} 2 & 1 & 0 \end{array} \leftarrow \text{позиции} \\
 \hline
 456_8 \\
 -277_8 \\
 \hline
 157_8
 \end{array}$$

$$0) \quad 16-7 = 10+6-7 = (7+1)+6-7 = 7;$$

$$1) \quad 14-7 = 10+4-7 = (7+1)+4-7 = 5;$$

$$2) \quad 3-2 = 1;$$

*Пояснения:*

Позиция 0) Из 6 нельзя вычесть 7. Занимаем 1 из левой позиции, было 5, остается 4 в позиции 1, т. е.  $56_8 = (40+16)_8$ .

Получаем  $(16-7)_8 = 10+6-7 = (7+1)+6-7 = 7_8$ .

Позиция 1) Было 5, осталось 4. Из 4 нельзя вычесть 7. Занимаем 1 из левой позиции; было 4, остается 3 в позиции 2, т. е.

$44_8 = (30+14)_8$ . Получаем  $(14-7)_8 = 10+4-7 = (7+1)+4-7 = 5_8$ .

Позиция 2) Было 4, осталось 3. Получаем  $3-2 = 1$ .

**Примеры в  $D_{16}$ .** Базис = 0, 1, ..., 9, A, B, C, D, E, F. Затем 10, 11, ..., 19, 1A, 1B, 1C, 1D, 1E, 1F, 20, ..., 2F, 30... и т. д.

Здесь  $A=10_{10}$ ,  $B=11_{10}$ ,  $C=12_{10}$ ,  $D=13_{10}$ ,  $E=14_{10}$ ,  $F=15_{10}$ .

Напоминаем, что  $15_{10}+1 = F+1 = 10_{16}$ ;  $10_{16}+7 = 17_{16}$ .

**Пример 4.** Сложение:  $A5B_{16} + C7E_{16} = 16D9_{16}$ .

$$\begin{array}{r}
 \begin{array}{ccc} 2 & 1 & 0 \end{array} \leftarrow \text{поз} \\
 \hline
 A5B_{16} \\
 +C7E_{16} \\
 \hline
 \mathbf{16D9} \\
 \hline
 1
 \end{array}$$

*Пояснения:*

Позиция 0)  $(B+E)_{16} = 9+2+E+2-2 = 10_{16} + 9 = 19_{16}$ . 9 пишем, а 1 в уме в позиции 1.

Позиция 1)  $5+7+1 = 13_{10}$ , пишем D.

Позиция 2)  $(A+C)_{16} = 9+1+C+4-4 = 10_{16} + 6 = 16_{16}$ . 6 пишем в позиции 2, и 1 в позиции 3.

**Пример 5.** Вычитание:  $C5B_{16} - A7E_{16} = 1DD_{16}$ .

$$\begin{array}{r}
 2\ 1\ 0 \leftarrow \text{поз} \\
 \hline
 C5B_{16} \\
 -A7E_{16} \\
 \hline
 1DD
 \end{array}$$

*Пояснения:*

Позиция 0) Из В нельзя вычесть Е. Занимаем 1 из левой позиции; было 5, остается 4 в позиции 1, т. е.  $5B_{16} = (40+1B)_{16}$ . Получаем  $(1B - E)_{16} = 10_{16} + B - E = 10_{16} + (9+2) - (10_{16} - 2) = 9+4 = D$ .

Позиция 1) Было 5, осталось 4. Из 4 нельзя вычесть 7. Занимаем 1 из левой позиции; было С, остается В в позиции 2, т. е.  $C4_{16} = (B0+14)_{16}$ . Получаем  $(14-7)_{16} = (10_{16} + 4) - 7 = 10_{16} - 3 = D$ .

Позиция 2) Было С, осталось В. Получаем  $B - A = 1$ .

### 3.8 Таблицы сложения и умножения в $D_8$ и $D_{16}$

Для выполнения арифметических операций в  $D_8$  и  $D_{16}$  можно воспользоваться таблицами 3.1– 3.4 .

Таблица 3.1 – Сложение в  $D_8$

+	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	1	2	3	4	5	6	7	10
2	2	3	4	5	6	7	10	11
3	3	4	5	6	7	10	11	12
4	4	5	6	7	10	11	12	13
5	5	6	7	10	11	12	13	14
6	6	7	10	11	12	13	14	15
7	7	10	11	12	13	14	15	16

Таблица 3.2 – Умножение в  $D_8$ 

*	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	1	2	3	3	5	6	7
2	0	2	4	6	10	12	14	16
3	0	3	6	11	14	17	22	25
4	0	4	10	14	20	24	30	34
5	0	5	12	17	24	31	36	43
6	0	6	14	22	30	36	44	52
7	0	7	16	25	34	43	52	61

Таблица 3.3 – Сложение в  $D_{16}$ 

+	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0															
1	1	2														
2	2	3	4							Симметрично относительно главной диагонали						
3	3	4	5	6												
4	4	5	6	7	8											
5	5	6	7	8	9	A										
6	6	7	8	9	A	B	C									
7	7	8	9	A	B	C	D	E								
8	8	9	A	B	C	D	E	F	10							
9	9	A	B	C	D	E	F	10	11	12						
A	A	B	C	D	E	F	10	11	12	13	14					
B	B	C	D	E	F	10	11	12	13	14	15	16				
C	C	D	E	F	10	11	12	13	14	15	16	17	18			
D	D	E	F	10	11	12	13	14	15	16	17	18	19	1A		
E	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	
F	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E

Таблица 3.4 – Умножение в  $D_{16}$ 

*	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0															
1	0	1														
2	0	2	4							Симметрично относительно главной диагонали						
3	0	3	6	9												
4	0	4	8	C	10											
5	0	5	A	F	14	19										
6	0	6	C	12	18	1E	24									
7	0	7	E	15	1C	23	2A	31								
8	0	8	10	18	20	28	30	38	40							
9	0	9	12	1B	24	2D	36	3F	48	51						
A	0	A	14	1E	28	32	3C	46	50	5A	64					
B	0	B	16	21	2C	37	42	4D	58	63	6E	79				
C	0	C	18	24	30	3C	48	54	60	6C	78	84	90			
D	0	D	1A	27	34	41	4E	5B	68	75	82	8F	9C	A9		
E	0	E	1C	2A	38	46	54	62	70	7E	8C	9A	A8	B6	C4	
F	0	F	1E	2D	3C	4B	5A	69	78	87	96	A5	B4	C3	D2	E1

Рассмотрим те же самые примеры из пункта 3.7. Отметим, что здесь также не привлекается на помощь система  $D_{10}$ .

### Примеры в $D_8$ .

**Пример 1.** Сложение:  $356_8 + 4662_8 = 5240_8$ .

$$\begin{array}{r}
 \begin{array}{cccc}
 3 & 2 & 1 & 0 \\
 \hline
 & & & \leftarrow \text{позиции}
 \end{array} \\
 \begin{array}{r}
 356_8 \\
 + 4662_8 \\
 \hline
 5240 \\
 \hline
 111
 \end{array}
 \end{array}$$

*Пояснения:* пользуемся таблицей сложения  $D_8$ . Слагаемые берутся из вертикального столбца и горизонтальной строки, а результат сложения находится внутри таблицы на пересечении столбца и строки:

Позиция 0)  $(6+2)_8 = 10_8$ . 0 пишем, а 1 в уме в позиции 1.

Позиция 1)  $5+6+1 = 14_8$ . 4 пишем, а 1 в уме в позиции 2.

Позиция 2)  $3+6+1 = 12_8$ . 2 пишем, а 1 в уме в позиции 3.

Позиция 3)  $4+1 = 5$ .

**Пример 2.** Вычитание:  $5240_8 - 356_8 = 4662_8$ .

$$\begin{array}{r}
 \begin{array}{cccc}
 3 & 2 & 1 & 0 \\
 \hline
 5 & 2 & 4 & 0_8 \\
 - & 3 & 5 & 6_8 \\
 \hline
 4 & 6 & 6 & 2_8
 \end{array}
 \end{array}
 \leftarrow \text{позиции}$$

*Пояснения:* пользуемся таблицей сложения  $D_8$ . Второе число берется из вертикального столбца, первое число находится правее внутри таблицы, результат находится вверху на горизонтальной строке.

Позиция 0) Из 0 нельзя вычесть 6. Занимаем 1 из левой позиции, было 4, остается 3 в позиции 1, т. е.  $40_8 = (30+10)_8$ . Получаем  $(10-6)_8 = 2_8$ .

Позиция 1) Было 4, осталось 3. Из 3 нельзя вычесть 5. Занимаем 1 из левой позиции; было 2, остается 1 в позиции 2, т. е.  $23_8 = (10+13)_8$ . Получаем  $(13-5)_8 = 6_8$ .

Позиция 2) Было 2, остался 1. Из 1 нельзя вычесть 3. Занимаем 1 из левой позиции; было 5, остается 4 в позиции 3, т. е.  $51_8 = (40+11)_8$ . Получаем  $(11-3)_8 = 6_8$ .

Позиция 3) Остается 4.

**Пример 3.** Вычитание:  $456_8 - 277_8 = 157_8$ .

$$\begin{array}{r}
 \begin{array}{ccc}
 2 & 1 & 0 \\
 \hline
 4 & 5 & 6_8 \\
 - & 2 & 7 & 7_8 \\
 \hline
 1 & 5 & 7_8
 \end{array}
 \end{array}
 \leftarrow \text{позиции}$$

*Пояснения:*

Позиция 0) Из 6 нельзя вычесть 7. Занимаем 1 из левой позиции, было 5; остается 4 в позиции 1, т. е.  $56_8 = (40+16)_8$ . Получаем  $(16-7)_8 = 7_8$ .

Позиция 1) Было 5, осталось 4. Из 4 нельзя вычесть 7. Занимаем 1 из левой позиции; было 4, остается 3 в позиции 2, т. е.  $44_8 = (30+14)_8$ . Получаем  $(14-7)_8 = 5_8$ .

Позиция 3) Было 4, осталось 3. Получаем  $3-2 = 1$ .

**Примеры в  $D_{16}$ .** Базис = 0, 1, ..., 9, A, B, C, D, E, F. Затем 10, 11, ..., 19, 1A, 1B, 1C, 1D, 1E, 1F, 20, ..., 2F, 30... и т. д.

Здесь  $A = 10_{10}$ ,  $B = 11_{10}$ ,  $C = 12_{10}$ ,  $D = 13_{10}$ ,  $E = 14_{10}$ ,  $F = 15_{10}$ .

Напоминаем, что  $15_{10} + 1 = F + 1 = 10_{16}$ ;  $10_{16} + 7 = 17_{16}$ .

**Пример 4.** Сложение:  $A5B_{16} + C7E_{16} = 16D9_{16}$ .

$$\begin{array}{r}
 \text{---} 2\ 1\ 0 \text{---} \leftarrow \text{поз} \\
 A5B_{16} \\
 + C7E_{16} \\
 \hline
 16D9 \\
 \text{---} 1 \text{---}
 \end{array}$$

*Пояснения:*

Позиция 0)  $(B+E)_{16} = 19_{16}$ . 9 пишем, 1 в уме в следующей позиции.

Позиция 1)  $5+7+1 = D$ .

Позиция 2)  $(A+C)_{16} = 16_{16}$ . 6 пишем в позиции 2, и 1 в позиции 3.

**Пример 5.** Вычитание:  $C5B_{16} - A7E_{16} = 1DD_{16}$ .

$$\begin{array}{r}
 \text{---} 2\ 1\ 0 \text{---} \leftarrow \text{поз} \\
 C5B_{16} \\
 - A7E_{16} \\
 \hline
 1DD
 \end{array}$$

*Пояснения:*

Позиция 0) Из  $B = 11_{10}$  нельзя вычесть  $E = 14_{10}$ . Занимаем 1 из левой позиции, было 5, остается 4 в позиции 1, т. е.

$5B_{16} = (40+1B)_{16}$ . Получаем  $(1B - E)_{16} = D$ .

Позиция 1) Было 5, осталось 4. Из 4 нельзя вычесть 7. Занимаем 1 из левой позиции, было C, остается B в позиции 2, т. е.

$C4_{16} = (B0+14)_{16}$ . Получаем  $(14-7)_{16} = D$ .

Позиция 3) Было C, осталось B. Получаем  $B-A = 1$ .



### 3.9 Вопросы и задания

1. Выполните сложение следующих двоичных чисел:

- а)  $1010101 + 110110$ ;      г)  $11011 + 1101101$ ;  
 б)  $1011011 + 111101$ ;      д)  $111011 + 100111101$ ;

Для проверки повторите вычисления, переходя к  $D_{10}$ , а потом преобразуя результат обратно в  $D_2$ .

2. Выполните вычитание следующих двоичных чисел:

- а)  $1010101 - 110110$ ;      г)  $1101101 - 11011$ ;  
 б)  $1011011 - 111101$ ;      д)  $100111101 - 111011$ ;

Для проверки повторите вычисления, переходя к  $D_{10}$ , а потом преобразуя результат обратно в  $D_2$ .

3. Выполните умножение следующих двоичных чисел:

- а)  $1010101 * 110110$ ;      г)  $1101101 * 11011$ ;  
 б)  $1011011 * 111101$ ;      д)  $100111101 * 111011$ ;

Для проверки повторите вычисления, переходя к  $D_{10}$ , а потом преобразуя результат обратно в  $D_2$ .

4. Выполните деление следующих двоичных чисел:

- а)  $1010101 : 110110$ ;      г)  $1101101 : 11011$ ;  
 б)  $1011011 : 111101$ ;      д)  $100111101 : 111011$ ;

Для проверки повторите вычисления, переходя к  $D_{10}$ , а потом преобразуя результат обратно в  $D_2$ .

5. Чему равна сумма двух чисел  $a_8 + b_{16} = 438 + 56_{16}$  ?

*Решение.*  $a = 43_8 = 100011_2 = 23_{16} = 35_{10}$ .

$$b = 56_{16} = 1010110_2 = 126_8 = 86_{10}.$$

$$a + b = 121_{10} = 1111001_2 = 171_8 = 79_{16}.$$

6. Заданы два числа (например:  $a = 45_{10}$  и  $b = 17_{10}$ ). Надо:

6.1. Оба исходных числа и результаты перевести в  $D_2$ .

6.2. Сложить, вычесть, умножить числа **a**, **b** в  $D_2$ .

6.3. Числа **a**, **b** сложить, вычесть, умножить в  $D_{10}$ .

6.4. Перевести результат сложения, вычитания, умножения в десятичную систему.

6.5. Сравнить результаты.

*Решение.*

$$(6.1) 45 + 17 = 62; \quad 45 - 17 = 28; \quad 45 * 17 = 765;$$

$$(6.2) 45 = 32 + 8 + 4 + 1 = 1 * 2^5 + 1 * 2^3 + 1 * 2^2 + 1 * 2^0 = 101101;$$

$$17 = 16 + 1 = 1 \cdot 2^4 + 1 \cdot 2^0 = 10001;$$

$$62 = 32 + 16 + 8 + 4 + 2 = 2^5 + 2^4 + 2^3 + 2^2 + 2^1 = 111110;$$

$$28 = 16 + 8 + 4 = 2^4 + 2^3 + 2^2 = 11100;$$

$$765 = 512 + 128 + 64 + 32 + 16 + 8 + 4 + 1 = 2^9 + 2^7 + 2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 2^0 =$$

$$= 1011111101;$$

(6.3)

$\begin{array}{r} 101101 \\ + 10001 \\ \hline 111110 \end{array}$	$\begin{array}{r} 101101 \\ - 10001 \\ \hline 11100 \end{array}$	$\begin{array}{r} 101101 \\ * 10001 \\ \hline 101101 \\ +101101 \\ \hline 1011111101 \end{array}$
---	--	---

Обратите внимание, результаты двоичной арифметики совпали с переводами по п. (6.2).

(6.4)

$$111110 = 2^5 + 2^4 + 2^3 + 2^2 + 2^1 = 32 + 16 + 8 + 4 + 2 = 62 = 45 + 17;$$

$$11100 = 2^4 + 2^3 + 2^2 = 16 + 8 + 4 = 28 = 45 - 17;$$

$$1011111101 = 2^9 + 2^7 + 2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 2^0 =$$

$$= 512 + 128 + 64 + 32 + 16 + 8 + 4 + 1 = 765 = 45 \cdot 17.$$

(6.5) Все результаты совпали.

## 4 КОДИРОВАНИЕ ЧИСЕЛ

### 4.1 Замечание о кодировании информации

Кодирование информации связано с её *передачей*. Например.

Речь: Любой предмет или любое явление кодируется различным набором звуков в разных языках. Речь *передает* сообщение от одного человека другому.

Перевод текста: Одно и то же содержание кодируется разными словами для *передачи* информации от одного человека другому.

Азбука Морзе: Текстовое сообщение кодируется набором точек и тире для *передачи* информации.

Кодирование информации также связано с *сохранением* информации на носителе. Например.

Письменность: речь, как специфический набор звуков, закодирована с помощью букв и *сохранена* на носителе (бумажном).

Математические выражения: системой графических знаков кодируются закономерности и *сохраняются* в виде формул.

Нотное письмо: системой графических знаков кодируется музыка и *сохраняется* на бумажном носителе.

Звукозапись: закодированная запись музыки или речи и её *сохранение* на каком-нибудь носителе (виниловая пластинка, магнитная лента, компакт диск).

В последнее время для обработки информации используется компьютер, при этом информация также кодируется для её обработки.

В этой части книжки показываются некоторые методы кодирования трёх видов информации: числовой, текстовой и графической. При этом методы показаны сами по себе, в виде идей, без всякой связи с компьютером. Однако именно эти методы будут использованы в дальнейшем при представлении информации в компьютере. Напомним также, что кодирование преследует цель получения двоичного кода информации.

## 4.2 Замечание об отрицательном числе

Рассмотрим пример: набор сосудов с водой (стаканы, банки, вёдра, бочки и т. п.). Учитывается вид сосуда и объём воды в нём. Если говорить о количестве сосудов и об объеме воды, то следует использовать цифры, среди которых может быть ноль (например, сосуд пустой). В этом примере также можно встретиться с некоторыми особенностями операций сложения и вычитания чисел. Например, нельзя из малого объема воды вычесть большой объем воды, т. е. при вычитании первое число должно быть не меньше второго числа. Или в сосуд нельзя поместить такой объем воды, который больше объема сосуда, т. е. возникает ситуация переполнения.

Продолжим рассуждения. Допустим, учитывается также температура воды в сосуде. Здесь уже операции сложения и вычитания числа, характеризующего температуру воды, требуют более внимательного рассмотрения. Например,

5 литров воды с  $60^{\circ}\text{C}$  + 3 литра воды с  $10^{\circ}\text{C}$ .

Что получим в итоге? Объем воды = 8 литров. Чему равна температура воды? Очевидно, что  $\neq 70^{\circ}\text{C}$  (будет  $41,25^{\circ}\text{C}$ ).

Если же говорить только об изменении температуры, то следует использовать принцип моделирования явления, т.е. отвлечься от предмета (от объема воды) и учитывать только существенное свойство, а именно, температуру. Например, при увеличении температуры на  $10^{\circ}\text{C}$  путем нагрева, получим  $60 + 10 = 70$ . При уменьшении температуры на  $40^{\circ}\text{C}$  путем охлаждения, получим  $60 - 40 = 20$ . Если дальше продолжать охлаждение, то можно получить лёд, т.е. *отрицательную* температуру.

Таким образом, понятие *отрицательного* числа применяется только к некоторым свойствам реальных предметов. При этом количественная характеристика этого свойства делится на отрезки с обозначением значения и точка отсчета одного из отрезков принимается за 0. Значения больше нуля считаются *положительными*, значения меньше нуля считаются *отрицательными*. Поэтому понятие *отрицательного* числа – это абстракция, условность, принятая для удобства рассуждения.

Кстати, если говорить о температуре, то её измеряют в разных градусных единицах: Цельсия, Фаренгейта, Кельвина, Ньютона, Реомюра и т. д. Отличие в измерениях в том, что (?) принимается за 0 и что означает 1 градус? Например:

✓ градусы Цельсия<sup>1</sup>:  $0^{\circ}\text{C}$  означает температуру замерзания воды (превращения воды в лёд), а  $100^{\circ}\text{C}$  – температуру закипания воды (превращения воды в пар). Тогда  $1^{\circ}\text{C}$  означает сотую часть температуры воды от состояния замерзания до состояния закипания.

✓ градусы Фаренгейта<sup>2</sup>:  $0^{\circ}\text{F}$  означает температуру замерзания смеси воды, соли и нашатыря (1:1:1). Температура таяния льда из чистой воды равна  $+32^{\circ}\text{F}$ , а точка кипения воды  $+212^{\circ}\text{F}$ . Тогда  $10^{\circ}\text{F}$  равен  $1/180$  разности этих температур.

✓ градусы Кельвина<sup>3</sup>:  $0^{\circ}\text{K}$  означает температуру абсолютного холода (минимальный предел температуры, которую может иметь физическое тело во Вселенной). Температура тройной точки воды принята в качестве другой опорной величины и её значение составляет  $273,16^{\circ}\text{K}$ . Тройная точка воды – это строго определенные значения температуры и давления, при которых вода может одновременно существовать в виде трёх фаз - в твердом, жидком и газообразном состояниях. Тогда  $1^{\circ}\text{K}$  равен  $1/273,16$ . Таким образом, в градусах Кельвина нет *отрицательной* температуры, только положительные. Пример измерения температуры в градусах Кельвина показывает применение принципа «смещения», когда точка отсчета смещается так, чтобы не было отрицательных значений.

### 4.3 Кодирование целого числа

Рассматриваются целые числа со знаком (как положительные, так и отрицательные). Числа могут быть небольшие, средние или

---

<sup>1</sup> швед. физик Цельсий (A. Celsius; 1701 - 1744).

<sup>2</sup> нем. физик Фаренгейт (G. D. Fahrenheit; 1686-1736).

<sup>3</sup> англ. физик Уильям Томсон, лорд Кельвин - William Thomson, lord Kelvin (1824–1907).

большие. Поэтому в зависимости от значения, целые числа могут быть подразделены на следующие типы:

*short integer* (короткое целое) =  $\pm(2^7-1) = \pm 127_{10}$ ;

*integer* (целое) =  $\pm(2^{15}-1) = \pm 32\,767_{10}$ ;

*long integer* (длинное целое) =  $\pm(2^{31}-1) = \pm 2\,147\,483\,647_{10}$ ;

Если число только положительное (без знака) тогда:

*unsigned short integer* =  $0, \dots, 2^8-1 = 0, \dots, 255_{10}$ ;

*unsigned integer* =  $0, \dots, 2^{16}-1 = 0, \dots, 65\,535_{10}$ ;

*unsigned long integer* =  $0, \dots, 2^{32}-1 = 0, \dots, 4\,294\,967\,295_{10}$ ;

Для размещения числа, принадлежащего указанному типу, при двоичном кодировании отводится следующее количество позиций (бит):

*(unsigned) short integer*: 8 бит = 1 байт;

*(unsigned) integer*: 16 бит = 2 байта;

*(unsigned) long integer*: 32 бит = 4 байта.

В дальнейшем будем всегда оговаривать тип целого числа и примеры будем приводить применительно к числам типа *short integer* или *integer*. Напомним также, что исходное число задаётся в десятичной системе счисления.

### 4.3.1 Прямой код числа

Правило кодирования:

- 1) Исходное десятичное число перевести в двоичную систему.
- 2) В зависимости от типа числа выделить необходимое количество позиций (бит) для кодирования.
- 3) Старший бит отвести для знака числа (0 – положительное число, 1 – отрицательное).
- 4) Двоичное число записать в соответствующие позиции.
- 5) Оставшиеся позиции заполнить нулём.

**Пример 1.** Число = 75.

- 1) Перевод  $D_{10}$  в  $D_2$ :  $75_{10} = 64 + 8 + 2 + 1 = 1001011_2$ .
- 2) Число принадлежит к типу *short integer*, поэтому для кодирования отводится 8 бит.
- 3,4,5) Заполняем 8 позиций двоичным числом:

знак числа	7	6	5	4	3	2	1	0	позиции
	0	1	0	0	1	0	1	1	прямой код

**Пример 2.** Число =  $-75$ .

- 1) Перевод  $D_{10}$  в  $D_2$ :  $-75_{10} = -1001011_2$ .
- 2) Число принадлежит к типу *short integer*, поэтому для кодирования отводится 8 бит.
- 3,4,5) Заполняем 8 позиций двоичным числом:

знак числа	7	6	5	4	3	2	1	0	позиции
	1	1	0	0	1	0	1	1	прямой код

**Пример 3.** Заранее неизвестно значение числа, но известно, что число может изменяться в пределах  $\pm 15000$ . В таком случае для числа надо отвести 16 бит (2 байта). Тогда число  $-75$  будет кодировано следующим образом:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	позиции
1	0	0	0	0	0	0	0	0	1	0	0	1	0	1	1	прямой код

знак числа

**Пример 4.** Число =  $-1975$ .

- 1) Перевод  $D_{10}$  в  $D_2$ :  $-1975_{10} = -11110110111_2$ .
- 2) Число принадлежит к типу *integer*, поэтому для кодирования отводится 16 бит.
- 3,4,5) Заполняем 16 позиций двоичным числом:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	позиции
1	0	0	0	0	1	1	1	1	0	1	1	0	1	1	1	прямой код

знак числа

### 4.3.2 Дополнительный код числа

Для положительного числа дополнительный код совпадает с прямым кодом. Для отрицательного числа формируется дополнительный код следующим образом:

- 1) Исходное десятичное число перевести в двоичную систему.
- 2) В зависимости от типа числа выделить необходимое количество позиций (бит) для кодирования.
- 3) Двоичное число **без знака минуса** записать в соответствующие позиции. Оставшиеся позиции заполнить нулём.
- 4) Цифры во всех позициях инвертировать, т. е. 0 заменить на 1, а 1 заменить на 0.
- 5) К нулевой позиции прибавить 1, (таким способом учитываем отрицательный знак исходного числа).

**Пример 1.** Число =  $-75$ .

- 1) Перевод  $D_{10}$  в  $D_2$ :  $-75_{10} = -1001011_2$ .
- 2) Число принадлежит к типу *short integer*, поэтому для кодирования отводится 8 бит.
- 3,4,5) Заполняем 8 позиций двоичным числом:

7	6	5	4	3	2	1	0	позиции
0	1	0	0	1	0	1	1	двоичное число
1	0	1	1	0	1	0	0	инвертируем
							1	прибавляем
<b>1</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>1</b>	дополнительный код

**Пример 2.** Число =  $-1975$ .

- 1) Перевод  $D_{10}$  в  $D_2$ :  $-1975_{10} = -11110110111_2$ .
- 2) Число принадлежит к типу *integer*, поэтому для кодирования отводится 16 бит.
- 3,4,5) Заполняем 16 позиций двоичным числом:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	позиции
0	0	0	0	0	1	1	1	1	0	1	1	0	1	1	1	двоичное число
1	1	1	1	1	0	0	0	0	1	0	0	1	0	0	0	инвертируем
															1	прибавляем
<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>1</b>	дополнит. код



В дополнительном коде отрицательного числа видно, что старший бит всегда равен 1. Дополнительный код положительного числа совпадает с прямым кодом, в котором старший бит всегда равен 0. Таким образом, старший бит дополнительного кода для исходного числа с любым знаком всегда правильно отражает знак числа.

#### **4.4 Арифметические операции с кодированными целыми числами**

Ранее было сказано, что из четырёх арифметических операций основными являются сложение и вычитание. При этом при вычитании должно соблюдаться дополнительное условие, чтобы второе число было не больше первого. При операции сложения такого условия нет, поэтому будем считать, что сложение является более простой операцией, чем вычитание.

Кроме того, самое главное в кодирование – это не просто перевод числа в двоичную систему, а, во-первых, заполнение двоичным кодом *заранее принятое количество позиций* (1 байт, 2 байта, или 4 байта и т.п.), и, во-вторых, выполнение арифметических операций *с кодами, как с числами*. При этом, если при операции сложения возникает ситуация переполнения, то это не фатальная ошибка, просто надо было для результата сложения выделить заранее большее количество бит. Если при вычитании второе число оказывается больше первого числа, хотя по модулю (без знака) второе число меньше первого, то в такой ситуации предполагается наличие у первого кода слева одного дополнительного разряда с 1, которое не учитывается в расчетах.

Рассмотрим сложение и вычитание с кодированными числами. Напоминаем, что речь пойдёт об арифметических операциях с двоичными числами, заполняющие полностью блок из 8 бит или 16 бит и т. д.

##### **4.4.1 Прямой код**

В качестве примера возьмем следующие числа:

позиции	7	6	5	4	3	2	1	0
45	0	0	1	0	1	1	0	1
75	0	1	0	0	1	0	1	1
120	0	1	1	1	1	0	0	0
-45	1	0	1	0	1	1	0	1
-75	1	1	0	0	1	0	1	1
-120	1	1	1	1	1	0	0	0

Отметим очевидное, числа 45, 75, 120 – положительные, назовем их «нормальными», числа -45, -75, -120 – отрицательные, назовем их «абстрактными».

Сложение чисел  $45 + 75 = 120$  (сложение нормальных чисел):

	45	0	0	1	0	1	1	0	1
+	75	0	1	0	0	1	0	1	1
	120	0	1	1	1	1	0	0	0

Вычитание чисел  $75 - 45 = 30$  (вычитание нормальных чисел):

	75	0	1	0	0	1	0	1	1
-	45	0	0	1	0	1	1	0	1
	30	0	0	0	1	1	1	1	0

Вычитание – как сложение положительного и отрицательного (абстрактного) числа:  $75 + (-45) = 30$ :

	75	0	1	0	0	1	0	1	1
+	-45	1	0	1	0	1	1	0	1
	ответ	1	1	1	1	1	0	0	0

Здесь выполняется сложение двоичных кодов. При сложении нет ограничений на слагаемые, нет переполнения, и, тем не менее, ответ  $= -120$ , это неверно (см. таблицу исходных чисел прямого кода). Значит, вычитание как сложение отрицательного (абстрактного) числа дает неверный ответ.

Рассмотрим сложение двух отрицательных чисел

$$(-75) + (-45) = -120:$$

	-75	1	1	0	0	1	0	1	1
+	-45	1	0	1	0	1	1	0	1
	ответ	0	1	1	1	1	0	0	0

Здесь при сложении происходит переполнение в старшем (знаковом) разряде, а восьмибитный код результата дает ответ, равный +120 (см. таблицу исходных чисел прямого кода), это неверно. Значит, при сложении двух отрицательных чисел также получается неверный ответ.

Рассмотрим вычитание из положительного числа отрицательного числа:  $75 - (-45) = 120$ :

	75	0	1	0	0	1	0	1	1
-	-45	1	0	1	0	1	1	0	1
	ответ	1	0	0	1	1	1	1	0

Вычитать эти два числа, как восьмибитные коды, нельзя, так как второе число больше первого, хотя по модулю (без знака) второе число меньше первого. В таком случае, в первом числе предполагается наличие слева одного дополнительного разряда с 1, тогда вычитание становится возможным. Тем не менее, при этом результат восьмибитного кода равен -30 (см. таблицу исходных чисел прямого кода), это неверно. Значит, вычитание из положительного числа отрицательного также дает неверный ответ.

**Вывод:** арифметические операции сложения и вычитания, выполняемые с числами в *прямом* коде, дают *неверный* результат, когда одним из слагаемых является «абстрактное» отрицательное число.

#### 4.4.2 Дополнительный код

В качестве примера возьмем те же числа, только представленные в дополнительном коде:

позиции	7	6	5	4	3	2	1	0
45	0	0	1	0	1	1	0	1
75	0	1	0	0	1	0	1	1
120	0	1	1	1	1	0	0	0
-45	1	1	0	1	0	0	1	1
-75	1	0	1	1	0	1	0	1
-120	1	0	0	0	1	0	0	0

Сложение чисел  $45 + 75 = 120$ :

	45	0	0	1	0	1	1	0	1
+	75	0	1	0	0	1	0	1	1
	120	0	1	1	1	1	0	0	0

Вычитание чисел  $75 - 45 = 30$ :

	75	0	1	0	0	1	0	1	1
-	45	0	0	1	0	1	1	0	1
	30	0	0	0	1	1	1	1	0

Вычитание – как сложение положительного и отрицательного (абстрактного) числа:  $75 + (-45) = 30$ :

	75	0	1	0	0	1	0	1	1
+	-45	1	1	0	1	0	0	1	1
	30	0	0	0	1	1	1	1	0

Здесь при сложении происходит переполнение в старшем разряде, а восьмибитный код результата дает ответ  $= 30$ , это верно.

Сложение двух отрицательных чисел

$$(-75) + (-45) = -120:$$

	-75	1	0	1	1	0	1	0	1
+	-45	1	1	0	1	0	0	1	1
	-120	1	0	0	0	1	0	0	0

Здесь при сложении происходит переполнение в старшем разряде, а восьмибитный код результата дает ответ, равный  $-120$ , это верно.

Вычитание из положительного числа отрицательного

$$75 - (-45) = 120:$$

	75	0	1	0	0	1	0	1	1
-	-45	1	1	0	1	0	0	1	1
	120	0	1	1	1	1	0	0	0

Вычитать эти два числа, как восьмибитные коды, нельзя, так как второе число больше первого. В таком случае, в первом коде предполагается наличие слева одного дополнительного разряда с 1, тогда вычитание становится возможным. При этом результат восьмибитного кода равен 120, это верно.

**Вывод:** арифметические операции сложения и вычитания, выполняемые с числами в дополнительном коде, всегда дают верный результат.

Таким образом, вычисления с целыми двоичными числами следует выполнять в *дополнительном* коде.

## 4.5 Кодирование вещественного числа

### 4.5.1 Понятия, связанные с вещественным числом

Число называется *вещественным*, если имеется дробная часть числа. Например,  $75,25_{10}$ . Здесь 75 – целая часть числа; 0,25 – дробная часть.

Число можно записать по-разному, например,

$$75,25_{10} = 7,525 \cdot 10 = 0,7525 \cdot 10^2 = 0,07525 \cdot 10^3 = 752,5 \cdot 10^{-1} \text{ и т. д.}$$

Видно, что в такой записи, запятая ставится в любое место среди цифр. Поэтому употребляется понятие *плавающей запятой* или *плавающей точки* (иногда дробная часть отделяется точкой).

Среди возможных форм записи числа с плавающей запятой надо выбрать наиболее приемлемую форму. Мы примем форму записи вида  $0,7525 \cdot 10^2$ , в которой значащиеся цифры записаны справа от запятой после 0. Принятая форма называется **нормальной** формой записи числа с плавающей запятой.

В нормальной форме записи числа применяются термины:

✓ **мантисса** числа – это все значащиеся цифры в данном числе;

✓ **порядок** числа – количество цифр в целой части исходного числа, или количество сдвигов запятой;

✓ **основание** числа – количество цифр в принятой системе счисления (известный термин).

Таким образом, число в общем виде записывается  $0, m \cdot D^p$ ,

где  $m$  – мантисса числа, в нашем примере  $m = 7525$ ;

$p$  – порядок числа, в нашем примере  $p = 2$ ;

$D$  – основание числа, в нашем примере  $D = 10$ .

Так как в записи числа перед мантиссой всегда присутствует 0 и запятая, то иногда разрешается их не указывать и записывать число короче, как  $m \cdot D^p$ .

Вещественное число встречается в природе гораздо чаще, чем целое число. Вещественным числом можно описать очень широкий диапазон величин, от микромира до макромира. Для указания весьма малых и весьма больших величин используются приставки: нано-, микро-, милли-, кило-, мега- и т.п., что говорит о значительном отрицательном или положительном порядке числа. Чтобы в дальнейших вычислениях не оперировать с отрицательным порядком числа, принято сдвинуть точку отсчета числа в меньшую сторону на некоторую величину  $c$ , называемую **смещением** числа, и вместо истинного порядка  $p$  рассматривать искусственный порядок  $h$ , называемый **характеристикой** числа, т.е.  $h = p + c$ . В качестве смещения принимается некоторое число, в соответствии с общепринятым стандартом, чтобы характеристика числа при расчетах всегда оставалась *положительной* (см. пример измерения температуры в градусах Кельвина).

Например,  $c = 64_{10}$  или  $c = 128_{10}$ . Если принято смещение  $c = 64_{10}$  и даны числа:  $a = 0, m \cdot 10^{25}$  и  $b = 0, m \cdot 10^{-34}$ , тогда *характеристики* чисел будут равны, соответственно,

для числа  $a$ :  $h = p + c = 25 + 64 = 89$  ;

для числа  $b$ :  $h = p + c = -34 + 64 = 30$  ;

т. е. характеристика всегда положительная, хотя истинный порядок числа  $b$  является отрицательным.

Так как конечной целью является кодирование вещественного числа, то в соответствии с принятым значением смещения, принимается также некоторое количество позиций (бит) для размещения двоичного кода вещественного числа. Например, для  $c = 64_{10}$  принимается 32 бита (4 байта), из них 7 бит для характеристики числа.

#### 4.5.2 Правило кодирования вещественного числа

Правило кодирования вещественного числа следующее:

- 1) Исходное десятичное число перевести в двоичную систему.
- 2) Полученное двоичное число нормализовать.
- 3) В нормализованном числе выделить мантиссу и порядок.
- 4) Принять значение смещения и вычислить характеристику числа.

5) Заполнить принятые для кодирования позиции следующими двоичными данными:

- ✓ старший бит (31-й бит) для знака числа: 0 – положительное число, 1 – отрицательное число;
- ✓ 7 бит (с 30 по 24) для характеристики числа;
- ✓ 24 бита (с 23 по 0) для мантиссы числа.

При этом заполнять позиции слева направо.

**Пример 1.** Дано число  $75,25_{10}$ .

- 1) Перевод  $D_{10}$  в  $D_2$ :  $75,25_{10} = 1001011,01_2$ .
- 2) Нормализация числа:  $1001011,01_2 = 0,100101101 \cdot 2^7 = 0,100101101 \cdot 2^{111}$ . Здесь  $7_{10} = 111_2$ .
- 3) Мантисса  $m = 100101101$ ; порядок  $p = 111$ .
- 4) Смещение  $c = 64_{10} = 1000000_2$ .  
Характеристика  $h = p + c = 7_{10} + 64_{10} = 1000111_2$ .

5) Заполняем 32 бита двоичными данными:

3	3	2	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	1	1	9	8	7	6	5	4	3	2	1	0
1	0	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	1	1	1	1	0	0	1	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

↑  
 знак                      характеристика                      мантисса  
 числа

**Пример 2.** Дано число  $-1975,625_{10}$ .

1) Перевод  $D_{10}$  в  $D_2$ :  $-1975,625_{10} = -11110110111,101_2$ .

2) Нормализация числа:

$$-11110110111,101_2 = -0,11110110111101 \cdot 2^{11} =$$

$$= -0,11110110111101 \cdot 2^{1011}. \text{ Здесь } 11_{10} = 1011_2.$$

3) Мантисса  $m = 11110110111101$ ; порядок  $p = 1011$ .

4) Смещение  $c = 64_{10} = 1000000_2$ .

$$\text{Характеристика } h = p + c = 11_{10} + 64_{10} = 1001011_2.$$

5) Заполняем 32 бита двоичными данными:

3	3	2	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

### Замечания:

1) Понятие *скрытого* бита. Обратим внимание на нормализованное двоичное число, в котором мантисса всегда начинается с 1. Поэтому можно эту 1 не помещать в разрядную сетку, но её наличие всегда учитывать при выполнении расчетов. Использование *скрытого* бита позволяет увеличить мантиссу двоичного числа на один разряд, что должно повысить точность представления числа при выполнении расчетов. Отметим, в вышеприведенных примерах *не использовался* принцип скрытого бита.

2) Приведенные выше значения смещения  $c$ , количество бит для всего числа и для характеристики  $h$ , надо рассматривать как



идею кодирования чисел. Эти значения не должны интересовать пользователя. Конкретные значения этих величин приняты производителем процессора компьютера в соответствии с их стандартами.

3) Бóльшая часть вычислений в компьютере выполняется с вещественными числами. Поэтому для обработки вещественных чисел в компьютерах прежних поколений использовалось специальное устройство – *математический сопроцессор*. В современных компьютерах обработка вещественных чисел выполняется основным, центральным процессором, в который встроены действия математического сопроцессора.

#### 4.6 Вопросы и задания

1. Чем отличается представление (кодирование) целых чисел со знаком и без знака?
2. Приведите примеры величин, которые всегда имеют положительные значения?
3. Как кодируются целые числа?
4. Сколько бит (позиций) выделяется для целого числа при кодировании?
5. Что такое прямой код целого числа?
6. Что такое дополнительный код целого числа?
7. Чем отличается прямой и дополнительный код для целого положительного числа?
8. Чем отличается прямой и дополнительный код для целого отрицательного числа?
9. Какое максимальное целое положительное число можно записать в 1 байт (8 бит)?
10. Какое минимальное целое отрицательное число можно записать в 1 байт (8 бит)?
11. В чем главное преимущество дополнительного кода при кодировании отрицательных чисел?
12. Что означает термин вещественное число?
13. Что такое плавающая запятая (плавающая точка)?
14. Что такое нормализованная форма записи вещественного числа?

15. Что такое мантисса числа?
16. Что такое порядок числа?
17. Что такое характеристика числа?
18. Какие параметры являются основными в стандартах по кодированию вещественного числа?
19. Как кодируется вещественное число в формате 4 байта?
20. Запишите в нормализованном виде следующие десятичные вещественные числа:  $27 \cdot 10^{15}$ ; 2047; 3,45; 0,00075;  $0,345 \cdot 10^{-17}$ .
21. Запишите в нормализованном виде следующие двоичные вещественные числа (значащая часть и порядок даны в двоичной форме):  $101 \cdot 2^{1011}$ ;  $110011$ ;  $11,101$ ;  $0,000101$ ;  $0,011 \cdot 2^{-11011}$ .
22. Выполните кодирование вещественного десятичного числа 40,17 в формате 4 байта.
23. Двоичный код предидущего числа запишите в  $D_{16}$ .

## 5 КОДИРОВАНИЕ ТЕКСТА

### 5.1 Кодирование текстовой информации

Текстовая информация представляет собой набор символов в виде букв, цифр, скобок, знаков препинания, пробела и т. п. Целью кодирования является преобразование текста в набор из единиц и нулей. Идея преобразования в общем простая:

- 1) надо составить таблицу,
- 2) в каждую ячейку таблицы записать один символ,
- 3) кодом символа считать адрес ячейки,
- 4) адрес ячейки написать двоичным кодом.

Для этого разработаны общепринятые стандарты по размеру таблицы и расположению символов в таблице. Известны стандарты КОИ–7, КОИ–8. Здесь КОИ – код обмена информацией, цифры 7 или 8 обозначают количество бит для записи двоичного кода адреса ячейки таблицы. Эти стандарты морально устарели.

Сейчас применяется кодировочная таблица ASCII – American Standart Code for Information Interchange (американский стандарт код для обмена информацией). Аббревиатура ASCII читается «аски». Эта таблица состоит из 256 ячеек, из 16 строк и 16 столбцов. Нумерация ячеек сверху вниз, начиная с первого столбца. *Код символа в таблице ASCII равен восьмибитному двоичному коду адреса ячейки.* Номер ячейки может указываться в  $D_{10}$ ,  $D_8$ ,  $D_{16}$ . Первая половина таблицы – 128 ячеек, называется «международной», вторая половина таблицы – 128 ячеек, называется «альтернативной» (таблица 5.1 и 5.2). В этих таблицах адреса ячеек указаны в  $D_{16}$ .

Например, буква R имеет адрес  $52_{16}$ . Код буквы R равен  $01010010_2 = 52_{16}$ . Код буквы Д равен  $10000100_2 = 84_{16}$ .

Таблица 5.1– Первая половина таблицы *ASCII*  
(основная или «международная»)

	00	10	20	30	40	50	60	70
0		▸		0	@	P	`	p
1	☒	◀	!	1	A	Q	a	q
2	☒	‡	"	2	B	R	b	r
3	♥	!!	#	3	C	S	c	s
4	♦	¶	\$	4	D	T	d	t
5	♣	§	%	5	E	U	e	u
6	♠	=	&	6	F	V	f	v
7	•	⊕	'	7	G	W	g	w
8	◼	↑	<	8	H	X	h	x
9	◻	↓	>	9	I	Y	i	y
A	⊕	→	*	:	J	Z	j	z
B	♂	←	+	;	K	[	k	{
C	♀	└	,	<	L	\	l	:
D	♂	↕	–	=	M	]	m	}
E	♂	▲	.	>	N	^	n	~
F	※	▼	/	?	O	_	o	△

Таблица 5.2 – Вторая половина таблицы *ASCII*  
(кириллица или «альтернативная» CP866)










	<b>80</b>	<b>90</b>	<b>A0</b>	<b>B0</b>	<b>C0</b>	<b>D0</b>	<b>E0</b>	<b>F0</b>
<b>0</b>	<b>А</b>	<b>Р</b>	<b>а</b>		<b>Л</b>	<b>Ц</b>	<b>Р</b>	<b>≡</b>
<b>1</b>	<b>Б</b>	<b>С</b>	<b>б</b>		<b>⊥</b>	<b>⌒</b>	<b>С</b>	<b>±</b>
<b>2</b>	<b>В</b>	<b>Т</b>	<b>в</b>		<b>Г</b>	<b>П</b>	<b>Т</b>	<b>&gt;</b>
<b>3</b>	<b>Г</b>	<b>У</b>	<b>г</b>	<b> </b>	<b>└</b>	<b>Ц</b>	<b>У</b>	<b>&lt;</b>
<b>4</b>	<b>Д</b>	<b>Ф</b>	<b>д</b>	<b>┌</b>	<b>—</b>	<b>Е</b>	<b>Ф</b>	<b>ѓ</b>
<b>5</b>	<b>Е</b>	<b>Х</b>	<b>е</b>	<b>‡</b>	<b>+</b>	<b>Г</b>	<b>х</b>	<b>J</b>
<b>6</b>	<b>Ж</b>	<b>Ц</b>	<b>ж</b>	<b>  </b>	<b>ƒ</b>	<b>П</b>	<b>ц</b>	<b>÷</b>
<b>7</b>	<b>З</b>	<b>Ч</b>	<b>з</b>	<b>П</b>	<b>  </b>	<b>  </b>	<b>Ч</b>	<b>≈</b>
<b>8</b>	<b>И</b>	<b>Ш</b>	<b>и</b>	<b>‡</b>	<b>⌒</b>	<b>‡</b>	<b>Ш</b>	<b>о</b>
<b>9</b>	<b>Й</b>	<b>Щ</b>	<b>й</b>	<b>‡ </b>	<b>Г</b>	<b>└</b>	<b>Щ</b>	<b>.</b>
<b>A</b>	<b>К</b>	<b>Ь</b>	<b>к</b>	<b>  </b>	<b>⌒</b>	<b>Г</b>	<b>Ь</b>	<b>.</b>
<b>B</b>	<b>Л</b>	<b>Ы</b>	<b>л</b>	<b>└</b>	<b>⌒</b>		<b>Ы</b>	<b>Ј</b>
<b>C</b>	<b>М</b>	<b>Ъ</b>	<b>м</b>	<b>└</b>	<b>└</b>		<b>Ъ</b>	<b>п</b>
<b>D</b>	<b>Н</b>	<b>Э</b>	<b>н</b>	<b>Ц</b>	<b>=</b>		<b>Э</b>	<b>2</b>
<b>E</b>	<b>О</b>	<b>Ю</b>	<b>о</b>	<b>└</b>	<b>⌒</b>		<b>Ю</b>	
<b>F</b>	<b>П</b>	<b>Я</b>	<b>п</b>	<b>└</b>	<b>⌒</b>		<b>Я</b>	

Таблица *ASCII* может быть представлена в другом виде, например, таблица 5.3 или таблица 5.4.

Таблица 5.3 – Первая половина таблицы *ASCII*

<i>Неотображаемые символы</i>				<i>Отображаемые символы</i>					
Название	Упр. симв.	Сим- вол	D <sub>16</sub>	Сим- вол	D <sub>16</sub>	Сим- вол	D <sub>16</sub>	Сим- вол	D <sub>16</sub>
Нуль	^ @	<b>NUL</b>	00	<b>SP</b>	20	@	40	'	60
Начало заголовка	^A	<b>SOH</b>	01	!	21	<b>A</b>	41	<b>a</b>	61
Начало текста	^B	<b>STX</b>	02	“	22	<b>B</b>	42	<b>b</b>	62
Конец текста	^C	<b>ETX</b>	03	#	23	<b>C</b>	43	<b>c</b>	63
Конец передачи	^D	<b>EOT</b>	04	\$	24	<b>D</b>	44	<b>d</b>	64
Запрос	^E	<b>ENQ</b>	05	%	25	<b>E</b>	45	<b>e</b>	65
Подтверждение	^F	<b>ACK</b>	06	&	26	<b>F</b>	46	<b>f</b>	66
Звонок	^G	<b>BEL</b>	07	'	27	<b>G</b>	47	<b>g</b>	67
Шаг назад	^H	<b>BS</b>	08	(	28	<b>H</b>	48	<b>h</b>	68
Гориз. табуляция	^I	<b>HT</b>	09	)	29	<b>I</b>	49	<b>i</b>	69
Перевод строки	^J	<b>LF</b>	0A	*	2A	<b>J</b>	4A	<b>J</b>	6A
Вертик. табуляция	^K	<b>VT</b>	0B	+	2B	<b>K</b>	4B	<b>k</b>	6B
Перевод стр-цы	^L	<b>FF</b>	0C	,	2C	<b>L</b>	4C	<b>l</b>	6C
Возврат каретки	^M	<b>CR</b>	0D	-	2D	<b>M</b>	4D	<b>m</b>	6D
Нижний регистр	^N	<b>SO</b>	0E	.	2E	<b>N</b>	4E	<b>n</b>	6E
Верхний регистр	^O	<b>SI</b>	0F	/	2F	<b>O</b>	4F	<b>o</b>	6F
Завершен. связи	^P	<b>DLE</b>	10	0	30	<b>P</b>	50	<b>p</b>	70
Упр. устр-вом 1	^Q	<b>DC1</b>	11	1	31	<b>Q</b>	51	<b>q</b>	71
Упр. устр-вом 2	^R	<b>DC2</b>	12	2	32	<b>R</b>	52	<b>r</b>	72
Упр. устр-вом 3	^S	<b>DC3</b>	13	3	33	<b>S</b>	53	<b>s</b>	73
Упр. устр-вом 4	^T	<b>DC4</b>	14	4	34	<b>T</b>	54	<b>t</b>	74

Ошибка передачи	^U	<b>NAK</b>	15	<b>5</b>	35	<b>U</b>	55	<b>u</b>	75
Хол. синхрониз.	^V	<b>SYN</b>	16	<b>6</b>	36	<b>V</b>	56	<b>v</b>	76
Конец пер. блока	^W	<b>ETB</b>	17	<b>7</b>	37	<b>W</b>	57	<b>w</b>	77
Отмена	^X	<b>CAN</b>	18	<b>8</b>	38	<b>X</b>	58	<b>x</b>	78
Конец носителя	^Y	<b>EM</b>	19	<b>9</b>	39	<b>Y</b>	59	<b>y</b>	79
Подстановка	^Z	<b>SUB</b>	1A	<b>:</b>	3A	<b>Z</b>	5A	<b>z</b>	7A
Выход	^[	<b>ESC</b>	1B	<b>;</b>	3B	<b>[</b>	5B	<b>{</b>	7B
Раздел. файлов	^\ ^]	<b>FS</b>	1C	<b>&lt;</b>	3C	<b>\</b>	5C	<b> </b>	7C
Разделение групп	^]	<b>GS</b>	1D	<b>=</b>	3D	<b>]</b>	5D	<b>}</b>	7D
Раздел, записей	^^	<b>RS</b>	1E	<b>&gt;</b>	3E	<b>^</b>	5E	<b>~</b>	7E
Раздел. элемент.	^_ ^_	<b>US</b>	1F	<b>?</b>	3F	<b>_</b>	5F	<b>DEL</b>	7F

Таблица 5.4 – Вторая половина таблицы *ASCII*  
(приводится в сокращении, исключены символы псевдографики)

Сим- вол	D <sub>10</sub>	Сим- вол	D <sub>10</sub>	Символ	D <sub>10</sub>	Сим- вол	D <sub>10</sub>	Сим- вол	D <sub>10</sub>
<i>A</i>	128	<i>P</i>	144	<i>a</i>	160	<i>p</i>	224	<i>Ě</i>	240
<i>Б</i>	129	<i>С</i>	145	<i>б</i>	161	<i>с</i>	225	<i>ě</i>	241
<i>В</i>	130	<i>Т</i>	146	<i>в</i>	162	<i>т</i>	226	<i>€</i>	242
<i>Г</i>	131	<i>У</i>	147	<i>г</i>	163	<i>у</i>	227	<i>€</i>	243
<i>Д</i>	132	<i>Ф</i>	148	<i>д</i>	164	<i>ф</i>	228	<i>ĭ</i>	244
<i>Е</i>	133	<i>Х</i>	149	<i>е</i>	165	<i>х</i>	229	<i>ĭ</i>	245
<i>Ж</i>	134	<i>Ц</i>	150	<i>ж</i>	166	<i>ц</i>	230	<i>Ÿ</i>	246
<i>З</i>	135	<i>Ч</i>	151	<i>з</i>	167	<i>ч</i>	231	<i>Ÿ</i>	247
<i>И</i>	136	<i>Ш</i>	152	<i>и</i>	168	<i>ш</i>	232	°	248
<i>Й</i>	137	<i>Щ</i>	153	<i>й</i>	169	<i>щ</i>	233	♦	249
<i>К</i>	138	<i>Ъ</i>	154	<i>к</i>	170	<i>ъ</i>	234	•	250
<i>Л</i>	139	<i>Ы</i>	155	<i>л</i>	171	<i>ы</i>	235	√	251
<i>М</i>	140	<i>Ь</i>	156	<i>м</i>	172	<i>ь</i>	236	<i>№</i>	252
<i>Н</i>	141	<i>Э</i>	157	<i>н</i>	173	<i>э</i>	237		253
<i>О</i>	142	<i>Ю</i>	158	<i>о</i>	174	<i>ю</i>	238		254
<i>П</i>	143	<i>Я</i>	159	<i>п</i>	175	<i>я</i>	239	□	255

Удобнее пользоваться таблицей *ASCII*, в которой адреса ячеек указаны в  $D_{16}$ , так как перевод из  $D_{16}$  в  $D_2$  выполняется очень просто. Например, кодирование слова «Вася»:

$$\begin{aligned}\text{Вася} &= 82 \text{ A0 E1 EF} = \\ &= 1000 \ 0010 \ 1010 \ 0000 \ 1110 \ 0001 \ 1110 \ 1111 = \\ &= 10000010101000001110000111101111.\end{aligned}$$

Таким образом, текстовая информация кодируется набором нулей и единиц.

## 5.2 Кодовые страницы

Ранее было сказано, что таблица *ASCII* условно разделена на две части: международная (американская, иногда называется *US-ASCII*) и альтернативная (например, кириллица). Альтернативная часть предназначена для кодирования букв национального алфавита страны. Для различения таблиц, им присвоены стандартные номера, которые называются кодовыми страницами (code page, коротко CP). Например,

- ✓ CP 866            Кириллица (DOS).
- ✓ CP 28595        Кириллица (ISO).
- ✓ CP 20866        Кириллица (KOI8-R).
- ✓ CP 21866        Кириллица (KOI8-U).
- ✓ CP 10007        Кириллица (Mac).
- ✓ CP 1251        Кириллица (Windows).

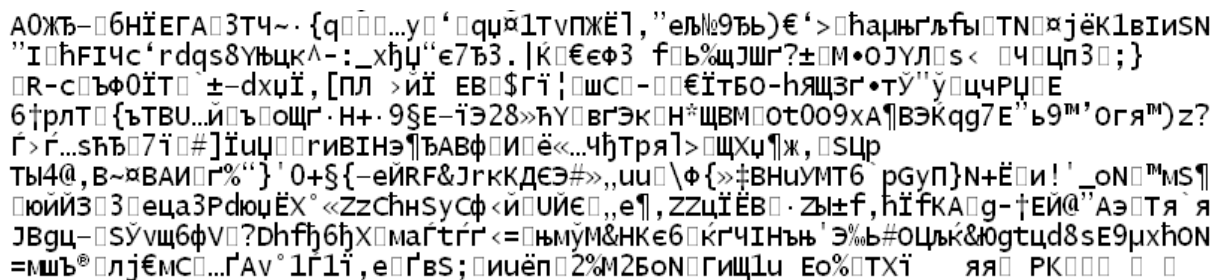
Здесь внутри скобок указывается название операционной системы или название стандарта. Приняты следующие сокращения:

- DOS – Disk Operating System (дискровая операционная система);
- ISO – International Standardization Organization (международная организация по стандартам);
- KOI8-R – КОИ8 (русский язык);
- KOI8-U – КОИ8 (украинский язык);
- Mac – Macintosh (линейка компьютеров фирмы Apple, управляемых операционной системой MacOS);
- Windows – операционная система.



Разные номера кодовых страниц для символов кириллицы указывают на различное расположение (разные адреса) символов кириллицы в альтернативной таблице.

Для всех языков, имеющих алфавитные системы письма, разработаны десятки разных кодовых страниц. Использование различных кодовых страниц создает много неудобств, как для пользователей, так и для программистов. При попытке прочесть текстовый файл при помощи кодовой страницы, не совместимой с той, в которой он был создан, возникают «кракозябры» - отображение документов в неправильной кодировке (рисунок 5.1).



А0ЖЪ-бнїЕГА3ТЧ~·{q...y'qм1твпЖЕ1,"ель9тъ)€'>һарыгльфыТНхјёк1вIиSN  
 "IһfIчс'rdqs8Yыцк^:-\_xђұ"е7ѳ3.|Кёеф3 fь%щJшг?±М•OJYЛs< чЦп3;}  
 R-сѳ0İT±-dxұİ,[пл>йİ ЕВ\$Гі!шс-İтбо-һящзг•тУ'ўцчРцЕ  
 6†рлТ{ътВU...йъощг·Н+·9§Е-іэ28»ҺYвгэкН\*щВМот009хА¶ВЭКqg7Е"ь9"огя")z?  
 Г>г...sђѳ7і#]İuцгивІнэ¶ѳавфИиё«...чђТряl>щХу¶ж,сцр  
 ты4@,в~хВАИг%"}'0+§{-ейRF&JгккдЕэ#»„uu\ф{»ѳВнуУМТ6`pgуп}N+ёи!'\_оN"MS¶  
 юйИЗ3ецаЗРдоуЕХ°«ZzсҺnsусф<йУИЕ„е¶,ZZцİЕВ·Zы±f,ҺİfкАg-†ЕЙ@"АэТя`я  
 JВgц-сSŸvщбфV?Dhfh6ђXмагтгг<=ньмУМ&нкє6кГЧІнь'э%ь#оцък&юgtcd8sE9мхҺon  
 =мшъ®лјЄМС...ГAv°1Г1і,еГвS;иуёп2%М2бонГищЦи Eo%ТХї яя PK

Рисунок 5.1 - Отображение документа в неправильной кодировке

Кроме того, существуют такие языки и системы письменности, в которых столь много знаков, что одного байта (8 бит), позволяющего кодировать не более 256 символов, недостаточно. Например, письменность типа слогового письма (языки: китайский, японский, корейский, вьетнамский) имеют количество символов более 6000.

### 5.3 Кодировка Unicode

В последние годы получил широкое распространение стандарт **Unicode**, как альтернатива традиционным кодовым страницам. Unicode – это уникальный код для любого символа. Применение этого стандарта (с 1991 г.) позволяет закодировать очень большое число символов из разных письменностей: китайские иероглифы, буквы греческого алфавита, латиницы и кириллицы, математические символы, символы валют, шахматных фигур и др.

Стандарт Unicode не такой простой, как 8-битные кодовые страницы. Стандарт Unicode имеет два раздела: первый (UCS) – набор символов, второй (UTF) – кодирование символов последовательностью нулей и единиц.

### 5.3.1 Универсальный набор символов

UCS (universal character set – универсальный набор символов). Это 17 таблиц символов, в каждой таблице количество ячеек равно  $16 \cdot 4096_{10} = 65536_{10} = 2^{16}$ . В каждой ячейке находится один символ. Во все таблицы UCS можно поместить  $17 \cdot 65536 = 1114112$  символов (более одного миллиона). Чаще всего используется первая («нулевая») таблица, называемая **базовой** или **основной** таблицей символов – Basic Multilingual Plane (BMP) – Основная Многоязычная Плоскость, содержащая почти все символы (65536) современных языков мира. В остальных 16 таблицах (плоскостях) находятся редко используемые символы. Например, первая плоскость используется для исторических письменностей, вторая — для редко используемых иероглифов, третья зарезервирована для архаичных китайских иероглифов (древнекитайское письмо содержит десятки тысяч разных иероглифов). Плоскости 15 и 16 выделены для частного употребления.

Каждую таблицу удобно представлять из 16 столбцов и 4096 строк, с адресацией в  $D_{16}$ . Ячейку таблиц символов UCS обозначают с приставкой буквы U. Например:

- U+0000,..., U+FFFF – ячейки базовой таблицы BMP;
- U+10000,..., U+FFFFF – ячейки с 1-й по 15 таблицы UCS;
- U+100000,..., U+10FFFF – ячейки 16-й таблицы UCS.

Фрагмент базовой таблицы BMP приводится на рисунке 5.2 (см. <http://unicode-table.com/ru>). Каждый символ (каждая ячейка) в таблицах UCS Unicode имеет свой адрес. Например, русская буква «Д» имеет адрес U+0414<sub>16</sub>. Отметим, что в Unicode это **адрес** буквы (символа), а не **код** символа. Напомним, что код – это последовательность нулей и единиц, заменяющая символ (букву) в памяти компьютера.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0390	İ	Α	Β	Γ	Δ	Ε	Ζ	Η	Θ	Ι	Κ	Λ	Μ	Ν	Ξ	Ο
03A0	Π	Ρ		Σ	Τ	Υ	Φ	Χ	Ψ	Ω	Ϊ	Ϋ	ά	έ	ή	ί
03B0	ϐ	α	β	γ	δ	ε	ζ	η	θ	ι	κ	λ	μ	ν	ξ	ο
03C0	π	ρ	ς	σ	τ	υ	φ	χ	ψ	ω	ϊ	ϋ	ό	ύ	ώ	ϙ
03D0	ϖ	ϗ	Ϙ	ϙ	Ϛ	ϛ	Ϝ	ϝ	Ϟ	ϟ	Ϡ	ϡ	Ϣ	ϣ	Ϥ	ϥ
03E0	Ϧ	ϧ	Ϩ	ϩ	Ϫ	ϫ	Ϭ	ϭ	Ϯ	ϯ	ϰ	ϱ	ϲ	ϳ	ϴ	ϵ
03F0	Ϸ	ϸ	Ϲ	Ϻ	ϻ	ϼ	Ͻ	Ͼ	Ͽ	Ϡ	ϡ	Ϣ	ϣ	Ϥ	ϥ	Ϧ
0400	Ѐ	Ё	Ђ	Ѓ	Є	Ѕ	І	Ї	Ј	Љ	Њ	Ћ	Ќ	Ў	Џ	Ц
0410	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П

Рисунок 5.2 – Фрагмент базовой таблицы Unicode

### 5.3.2 Код представления символа

UTF (Unicode transformation format – формат преобразования Unicode). UTF – это преобразование любого символа в уникальную последовательность байтов. Стандарт Unicode имеет несколько форм представления: UTF–8, UTF–16 и UTF–32. Семейство форматов UTF определяет машинное представление символов таблиц UCS. Числа в UTF означают количество бит для кодирования символа. Другими словами, один и тот же символ из таблицы может быть закодирован разным количеством бит (1,2,3,4 байта) в зависимости от принятого формата UTF.

UTF–8 – представление Юникода, обеспечивающее наилучшую совместимость со старыми системами, использовавшими 8-битные символы. Символы с номером меньше 128, при записи в UTF–8, превращается в обычный код US–ASCII. Остальные символы Юникода изображаются последовательностями длиной от 2 до 4 байт.

В UTF-16 символы представлены двумя байтами (16 битами). Эта кодировка используется в Windows, поскольку 16-битными значениями можно представить символы, составляющие алфавиты большинства языков мира.

В UTF-32 все символы представлены 4 байтами (32 битами). С точки зрения использования памяти эффективность UTF-32 далека от идеала. Поэтому данную кодировку редко применяют для передачи строк по сети и сохранения их в файлы. Как правило, UTF-32 используется как внутренний формат представления данных в программе.

## 5.4 Кодировка текста в документе

До сих пор рассматривалась кодировка текста, *как сообщения*. Символы, образующие текстовое сообщение, представляют собой некоторую графическую конструкцию, в которой не учитываются такие атрибуты символа, как размер, цвет, толщина линий т.п.

Например, текст, который указан в названии данной книжки и набран встроенным редактором, известной в прошлом, программы Norton Commander (рисунок 5.3), имеет, в соответствии с кодировкой ASCII (DOS), самый короткий размер файла, равный числу символов, умноженных на 1 байт (рисунок 5.4).

### Двоичная система и представление информации в компьютере

Рисунок 5.3 – Текст, набранный простейшим редактором

```
00000000: 84 A2 AE A8 E7 AD A0 EF 20 E1 A8 E1 E2 A5 AC A0 | Двоичная система
00000010: 20 A8 20 AF E0 A5 A4 E1 E2 A0 A2 AB A5 AD A8 A5 | и представление
00000020: 20 A8 AD E4 AE E0 AC A0 E6 A8 A8 20 A2 20 AA AE | информации в ко
00000030: AC AF EC EE E2 A5 E0 A5 | мпьютере
```

Рисунок 5.4 – Кодировка текста по ASCII

Если тот же текст, приведенный на рисунке 2.7, представить в кодировке Windows (CP 1251), то получим «кракозябры» (рисунок 5.5).

```
00000000: 84 A2 AE A8 E7 AD A0 EF|20 E1 A8 E1 E2 A5 AC A0 | „ÿÿÿÿ- п бёбвг-
00000010: 20 A8 20 AF E0 A5 A4 E1|E2 A0 A2 AB A5 AD A8 A5 | Ё ЁаГ*бв ÿ«Г-ЁГ
00000020: 20 A8 AD E4 AE E0 AC A0|E6 A8 A8 20 A2 20 AA AE | Ё-д@а- *ЁЁ ÿ ё@
00000030: AC AF EC EE E2 A5 E0 A5| | -ЇмовГаГ
```

Рисунок 5.5 – Тот же текст в другой кодировке

Совершенно другая картина получается при работе с текстовым редактором, который может привнести в начертание символа и текста некоторый *стиль*. В таком тексте, сохраненном на бумажном носителе, или в текстовом файле, или на экране монитора, каждый символ всегда обладает некоторым набором атрибутов, составляющих стиль. Например, на рисунке 5.6 одно и то же сообщение представлено четырьмя разными стилями (редактор MS Word 2007).

Двоичная система и представление информации в компьютере

*Двоичная система и представление информации в компьютере*

**Двоичная система и представление информации в компьютере**

**Двоичная система  
и представление информации  
в компьютере**

Рисунок 5.6 – Разные стили для одного и того же сообщения

При таком представлении текста, мы получаем не только сообщение, но и его некоторое оформление, которое увеличивает машинное представление символа. Например, размеры файлов, составленных из одного и того же сообщения, но с разными стилями оформления, имеют следующие величины (таблица 5.7).

Таблица 5.7 – Размер файла в зависимости от оформления

Оформление сообщения разными стилями (в рисунке 2.10)	Размер файла, байты
<i>стиль 1: Times New Roman,</i> 16 пт., по лев. краю.	10273
<i>стиль 2: Courier New, 18 пт.</i> курсив, по лев. краю.	10308
<i>стиль 3: Arial, 18 пт.</i> полужирный, по лев. краю.	10248
<i>стиль 4: Arial, 22 пт.</i> полужирный, по центру.	10337

## 5.5 Вопросы и задания

1. Как означает аббревиатура ASCII?
2. Что представляет собой кодировка ASCII?
3. Сколько символов включает кодировка ASCII?
4. Что является кодом символа в таблице ASCII?
5. Какой символ имеет ячейка  $100_{10}$  в таблице ASCII?
6. Какой код имеет цифра (символ) 5 в кодировке ASCII?
7. Что означает кодовая страница?
8. Назовите основные кодовые страницы, содержащие русские буквы?
9. Почему при просмотре текста могут появиться проблемы с распознаванием («кракозябры»)?
10. В чем состоит ограничение 8-битных кодировок?
11. Чему равен информационный объем сообщения «Двоичная система и представление информации в компьютере», закодированного с помощью 16-битной кодировки Unicode?

## 6 КОДИРОВАНИЕ ГРАФИКИ

### 6.1 Кодирование графической информации

Графической информацией является чертёж, рисунок, картина, фотография и т.п. Различают два вида графической информации: *векторную* и *растровую*.

*Векторная* графика составлена из линий, соединяющих заданные точки. Например, чертёж, или рисунок, выполненный пером.

*Растровая* графика составлена из сплошных областей с полутонами без резких переходов. Например, картина, выполненная кистью с масляными или акварельными красками.

В любом случае, элементом изображения является «точка», которую принято называть **пиксель** (иногда пишут пиксел - без мягкого знака). Слово пиксель иногда пишут английскими буквами *pixel*, образованного от сокращения *piccel* (picture cell – ячейка рисунка). Пиксель на экране дисплея отображается в виде светящейся точки, на бумаге (распечатке) – в виде точечного следа краски. Рассмотрим, как кодируются векторное и растровое изображение.

#### 6.1.1 Кодирование векторного изображения

Векторная графика состоит из простейших геометрических фигур, называемых *графическими примитивами* (линий, кривых, эллипсов, многоугольников). Каждый графический примитив характеризуется (описывается) своими параметрами: координаты вершин, размеры, углы наклона, цвет контура и цвет заливки. Линии описываются математическими формулами (формулы аналитической геометрии). Параметры задаются числовыми значениями. Пример векторного рисунка приведен на рисунке 6.1.



Рисунок 6.1 – Пример векторного рисунка

Векторный рисунок можно разобрать на части (примитивы) или собрать из частей (рисунок 6.2).

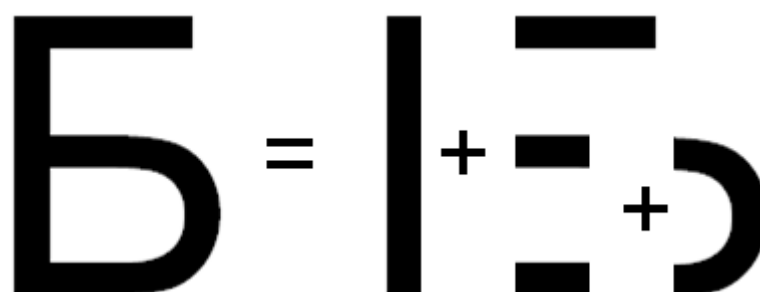


Рисунок 6.2 – Элементы векторного рисунка

Векторный способ кодирования рисунка обладает следующими преимуществами:

- ✓ Рисунок можно масштабировать (изменять размеры, поворачивать, растягивать) не теряя качества (четкости) изображения.

- ✓ Объем числовых данных, описывающих параметры графических примитивов, невелик. Объем файла векторного рисунка зависит от количества примитивов и их сложности. Как правило, этот объем меньше по сравнению с растровым способом кодирования (см. ниже).

Векторный способ кодирования имеет недостаток – он непригоден для рисунков с нечеткими (расплывчатыми) краями, например, фотография пейзажа.

Файлы векторных изображений имеют следующие расширения-форматы:



**.wmf** (англ. Windows Metafile – метафайл Windows, иногда расширение .emf) – стандартный файл векторных рисунков в операционной системе Windows);

**.cdr** – формат векторных рисунков программы Corel DRAW;

**.ai** – формат векторных рисунков программы Adobe Illustrator;

**.svg** (англ. Scalable Vektor Graphics – масштабируемая векторная графика) – векторная графика для веб-страниц в Интернете.

### 6.1.2 Кодирование растрового изображения

Растр – это сетка (чаще всего, прямоугольная). Растр просматривается слева направо и вниз (рисунок 6.3, а). Идея растрового кодирования состоит в наложении сетки на рисунок и моделирование рисунка набором прямоугольников. Пример растрового рисунка приведен на рисунке 6.3, б).

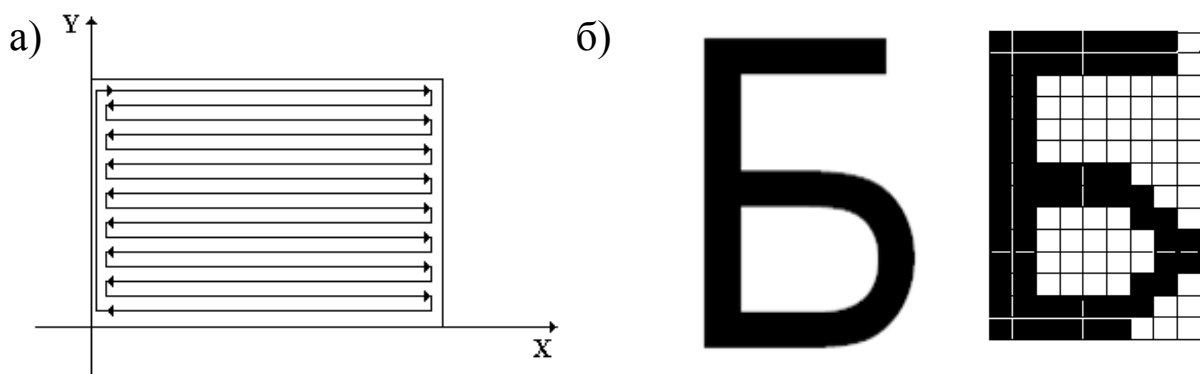


Рисунок 6.3 – Растровый рисунок  
а) растр; б) моделирование рисунка

Растровый рисунок – это набор пикселей-квадратиков. Размер квадратика определяется *разрешением*. Разрешение – это количество пикселей, приходящихся на единицу длины изображения. Разрешение измеряется в пикселях на дюйм, **ppi** – pixels per inch. Например, разрешение 254 ppi означает, что на 1 дюйм (25,4 мм) приходится 254 пикселя, так что каждый пиксель имеет размер  $0,1 \cdot 0,1$  мм. Чем больше разрешение, тем точнее (чётче) моделируется рисунок.

## 6.2 Кодирование цвета

Каждый пиксель кодируется указанием цвета. Например, для черно-белого рисунка, белый цвет – 1, черный цвет – 0. Тогда рисунок 6.3 можно закодировать матрицей 4•9, первые три строки которой, заполненные кодами, показаны ниже:

0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	1
0	0	1	1	1	1	1	1	1
...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...

Как быть, если рисунок цветной? Например, рисунок флага, в котором используются 4 цвета – черный, белый, красный, синий (рисунок 6.4, а). В приложении приводится рисунок в цвете.

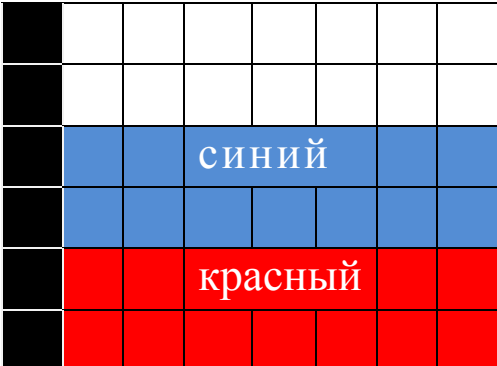
а)									б)								
		00	11	11	11	11	11	11		00	11	11	11	11	11	11	11
		00	11	11	11	11	11	11		00	11	11	11	11	11	11	11
		00	10	10	10	10	10	10		00	10	10	10	10	10	10	10
		00	10	10	10	10	10	10		00	10	10	10	10	10	10	10
		00	01	01	01	01	01	01		00	01	01	01	01	01	01	01
		00	01	01	01	01	01	01		00	01	01	01	01	01	01	01

Рисунок 6.4 – Кодирование цветного растрового рисунка;  
а) растровый рисунок; б) матрица кодирования рисунка

Для кодирования одного из четырех вариантов цвета нужно 2 бита, поэтому код каждого цвета (и код каждого пикселя) будет состоять из двух бит. Пусть 00 обозначает черный цвет, 01 – красный, 10 – синий и 11 – белый. Тогда получаем таблицу кодов (рисунок 6.4, б).

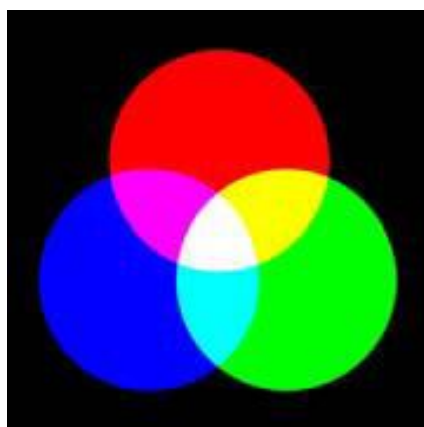
### 6.2.1 Кодирование цвета на экране

Считается, что любой *излучаемый* цвет можно имитировать, используя только три световых луча (красный, зеленый и синий) разной яркости. Следовательно, любой цвет (в том числе и «белый») приближенно раскладывается на три составляющих – красную, зеленую и синюю. Меняя яркость этих составляющих, можно составить любые цвета. Эта модель цвета получила название RGB по начальным буквам английских слов *red* – красный, *green* – зеленый и *blue* – синий (рисунок 6.5,а). Рисунок в цвете приведен в приложении.

В модели RGB яркость каждой составляющей чаще всего кодируется целым числом от 0 до 255. При этом код цвета – это тройка чисел (R,G,B), яркости отдельных составляющих. Цвет (0,0,0) – это черный цвет, а (255,255,255) – белый. Если все составляющие имеют равную яркость, получаются оттенки серого цвета, от черного до белого. Например, (75,75,75) – темно-серый, а (175,175,175) – светло-серый.

Чтобы сделать светло-красный (розовый) цвет, нужно в красном цвете (255,0,0) одинаково увеличить яркость зеленого и синего цветов, например, (255, 155, 155) – это розовый цвет. Это можно проверить в редакторе MSWord, *инструмент – цвет текста – другие цвета... – спектр* (рисунок 6.5,б).

а)



б)

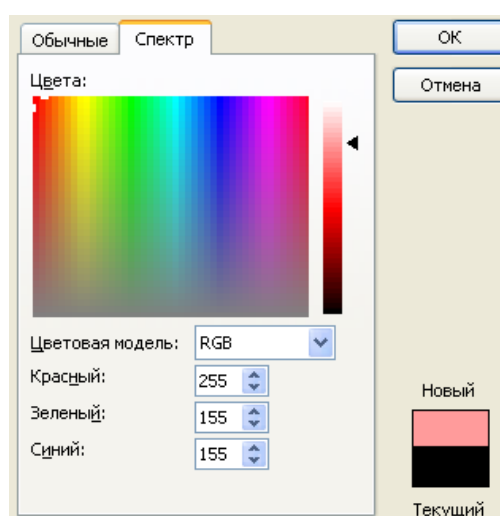


Рисунок 6.5 – Модель цвета RGB;

а) модель RGB; б) инструмент «цвет текста» в MSWord

Коды некоторых цветов представлены ниже в таблице 6.1.

Таблица 6.1 – Коды цветов

Цвет	Код RGB
красный; зеленый; синий;	(255,0,0); (0,255,0); (0,0,255);
белый; черный;	(255,255,255); (0,0,0);
фиолетовый; голубой; желтый;	(255,0,255); (0,255,255); (255, 255,0);
светло-желтый	(255,255,128)
тёмно-фиолетовый	(128,0,128)

Всего есть по 256 вариантов яркости каждого из трех цветов. Это позволяет закодировать  $256^3 = 16\,777\,216$  оттенков (более 16 миллионов), что более чем достаточно для человека. Так как  $256 = 2^8$ , каждая из трех составляющих занимает в памяти 8 бит или один байт, а вся информация о каком-то цвете – 24 бита (или три байта). Эта величина называется *глубиной цвета*.

**Глубина цвета** – это количество бит, используемое для кодирования цвета пикселя.

Каждому пикселю отводится от 1 бита до 3 байтов видеопамяти (*изображение формируется в видеопамяти*). Например:

- Монохромный режим, 2 цвета (черно-белый) – 1 бит (рисунок 6.3, б).

- Цветной режим, 8 цветов – 3 бит. Red=0; 1. Green=0; 1. Blue=0; 1. RGB =  $2^3 = 8$ .

- Цветной режим, 16 цветов – 4 бит;  $i = 0; 1$  – интенсивность (яркий, тусклый);  $i \text{ RGB} = 2 \cdot 2^3 = 2^4 = 16$  (таблица 6.2).

- Цветной режим, 256 цветов – 8 бит;  $i = 00000, \dots, 11111 = 2^5 = 32$  градации интенсивности;  $i \text{ RGB} = 2^5 * 2^3 = 2^8 = 256$ .

Или по 2 градации интенсивности и по 2 цвета RGB  $i^2 R^2 G^2 B^2 = 4 * 4 * 4 * 4 = 2^8 = 256$  (таблица 6.3).

- Цветной режим, 16 млн. цветов – 3 байта = 24 бит (рисунок 6.5, б).

Таблица 6.2 – Коды для формирования 16 цветов

<b>№</b>	<b>Цвет</b>	<b>i RGB</b>
1	Black (черный)	0000
2	Blue (синий)	0001
3	Green (зеленый)	0010
4	Cyan (сине-зеленый)	0011
5	Red (красный)	0100
6	Magenta (фиолетовый)	0101
7	Brown (коричневый)	0110
8	Light Gray (светло-серый)	0111
9	Dark Gray (темно-серый)	1000
10	Light Blue (ярко-синий)	1001
11	Light Green (ярко-зеленый)	1010
12	Light Cyan (бирюзовый)	1011
13	Light Red (ярко-красный)	1100
14	Light Magenta (лиловый)	1101
15	Yellow (желтый)	1110
16	White (белый)	1111

Таблица 6.3 – Коды для формирования 256 цветов

<b><i>i</i></b>	<b>R</b>	<b>G</b>	<b>B</b>
00	00	00	00
00	00	00	01
00	00	00	10
00	00	00	11
...	...	...	...
11	11	11	00
11	11	11	01
11	11	11	10
11	11	11	11

24-битное кодирование цвета часто называют режимом *истинного цвета* (англ. **True Color** – истинный цвет). Для вычисления объема рисунка в байтах при таком кодировании нужно определить общее количество пикселей (перемножить ширину и высоту) и умножить результат на 3, так как цвет каждого пикселя кодируется тремя байтами. Например, рисунок размером  $20 \times 30$  пикселей, закодированный в режиме истинного цвета, будет занимать  $20 \times 30 \times 3 = 1800$  байт.

Кроме режима истинного цвета используется также 16-битное кодирование (англ. **High Color** – «высокий» цвет), когда на красную и синюю составляющую отводится по пять бит, а на зеленую, к которой человеческий глаз более чувствителен – шесть бит (итого 16 бит). В режиме High Color можно закодировать  $2^{16} = 65536$  различных цветов. В мобильных телефонах 12-битное кодирование цвета (4 бита на составляющую,  $2^{12} = 4096$  цветов).

Связь между глубиной цвета и количеством формируемых цветов можно показать таблицей 6.4.

Таблица 6.4 – Глубина цвета и количество цветов

Глубина цвета, бит	8	15	16	24
		High Color		True Color
Кол-во цветов	256	32768	65536	16,7 млн

Как правило, чем меньше цветов используется, тем больше будет искажаться цветное изображение. Таким образом, при кодировании цвета тоже есть неизбежная потеря информации, которая «добавляется» к потерям, вызванным дискретизацией. Дискретизация возникает при замене рисунка набором пиксель-квадратиков. Однако при увеличении количества используемых цветов одновременно растет объем файла. Например, в режиме *истинного цвета* файл получится в два раза больше, чем при 12-битном кодировании.

Очень часто (например, в схемах, диаграммах и чертежах) количество цветов в изображении невелико (не более 256). В этом случае применяют *кодирование с палитрой*.

**Цветовая палитра** – это таблица, в которой каждому цвету, заданному в виде составляющих в модели RGB, сопоставляется числовой код.

**Размер палитры** – это количество байт, указывающие цвета палитры.

Например, черно-белая палитра, всего 2 цвета (рисунок 6.3):

- ✓ черный: RGB-код (0,0,0); двоичный код  $0_2$ ;
- ✓ белый: RGB-код (255,255,255); двоичный код  $1_2$ .

0	0	0	255	255	255
цвет черный, код $0_2$			цвет белый, код $1_2$		

Здесь размер палитры 6 байт.

Кодирование изображения флага, четыре цвета (рисунок 6.4):

- ✓ черный: RGB-код (0,0,0); двоичный код  $00_2$ ;
- ✓ красный: RGB-код (255,0,0); двоичный код  $01_2$ ;
- ✓ синий: RGB-код (0,0,255); двоичный код  $10_2$ ;
- ✓ белый: RGB-код (255,255,255); двоичный код  $11_2$ .

0	0	0	255	0	0	0	0	255	255	255	255
черный = $00_2$			красный = $01_2$			синий = $10_2$			белый = $11_2$		

Здесь размер палитры 12 байт.

Ниже приведены данные по некоторым вариантам кодирования с палитрой (Таблица 6.5).

Таблица 6.5 – Варианты кодирования с палитрой

Количество цветов	Размер палитры (байт)	Глубина цвета (бит на пиксель)
2	6	1
4	12	2
16	48	4
256	768	8

При известных характеристиках экрана монитора (разрешение экрана и количество цветов пикселя) можно вычислить минимальный объем видеопамати для формирования качественного изображения (таблица 6.6).

Таблица 6.6 – Объем видеопамати

Разрешение, пиксели	Кол-во пикселей	Объем видеопамати, Мбайт			
		256 цветов	32,7 тыс. цветов	65,5 тыс. цветов	16,7 млн. цветов
640*480	307200	0,3	0,55	0,58	0,88
800*600	480000	0,46	0,86	0,92	1,37
1024*768	786432	0,75	1,41	1,50	2,25
1280*1024	1310729	1,25	2,34	2,50	3,75

### **6.2.2 Кодирование цвета на бумаге**

RGB-кодирование лучше всего описывает цвет, который излучается некоторым устройством, например, монитором или экраном ноутбука. Когда же мы смотрим на изображение, отпечатанное на бумаге, ситуация совершенно другая. Мы видим не прямые лучи источника, попадающие в глаз, а отраженные от поверхности. «Белый свет» от какого-то источника (солнце, лампочка), содержащий волны во всем видимом диапазоне, попадает на бумагу, на которой нанесена краска. Краска поглощает часть лучей (их энергия уходит на нагрев бумаги), а оставшиеся отраженные цвета попадают в глаз, это и есть тот цвет, который мы видим.



Например, если краска поглощает красные лучи, отражаются только синие и зеленые – мы видим голубой цвет. В этом смысле красный и голубой цвета дополняют друг друга, так же, как и пары «зеленый – фиолетовый» и «синий – желтый». Действительно, если из белого цвета «вычесть» зеленый, то получится цвет фиолетовый, а если «вычесть» синий, то получится цвет желтый. Отметим синонимы цветов: фиолетовый = пурпурный.

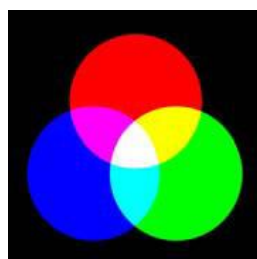
Ниже показаны соотношения падающего и отраженного цветов (Таблица 6.7).

Таблица 6.7 – Соотношения падающего и отраженного цветов

Излучаемый цвет	На бумаге	
	поглощение	отражение
(255,255,255) белый	(255,0,0) красный	$(0,255,255) = (0,255,0) + (0,0,255)$ ; голубой = зеленый + синий;
	(0,255,0) зеленый	$(255,0,255) = (255,0,0) + (0,0,255)$ ; фиолетовый = красный + синий;
	(0,0,255) синий	$(255,255,0) = (255,0,0) + (0,255,0)$ ; желтый = красный + зеленый;

На трех дополнительных цветах – голубом, фиолетовом и желтом – строится цветовая модель **СМУ** (англ. *Cyan* – голубой, *Magenta* – фиолетовый, *Yellow* – желтый), которая применяется для вывода на печать (рисунок 6.6,б). Таким образом, модели цветов RGB и CMY – обратимы (рисунок 6.7). Рисунок в цвете показан в приложении.

а)



б)



Рисунок 6.6 – Модели цветов;  
а) модель RGB (для монитора); б) модель CMY (для принтера)

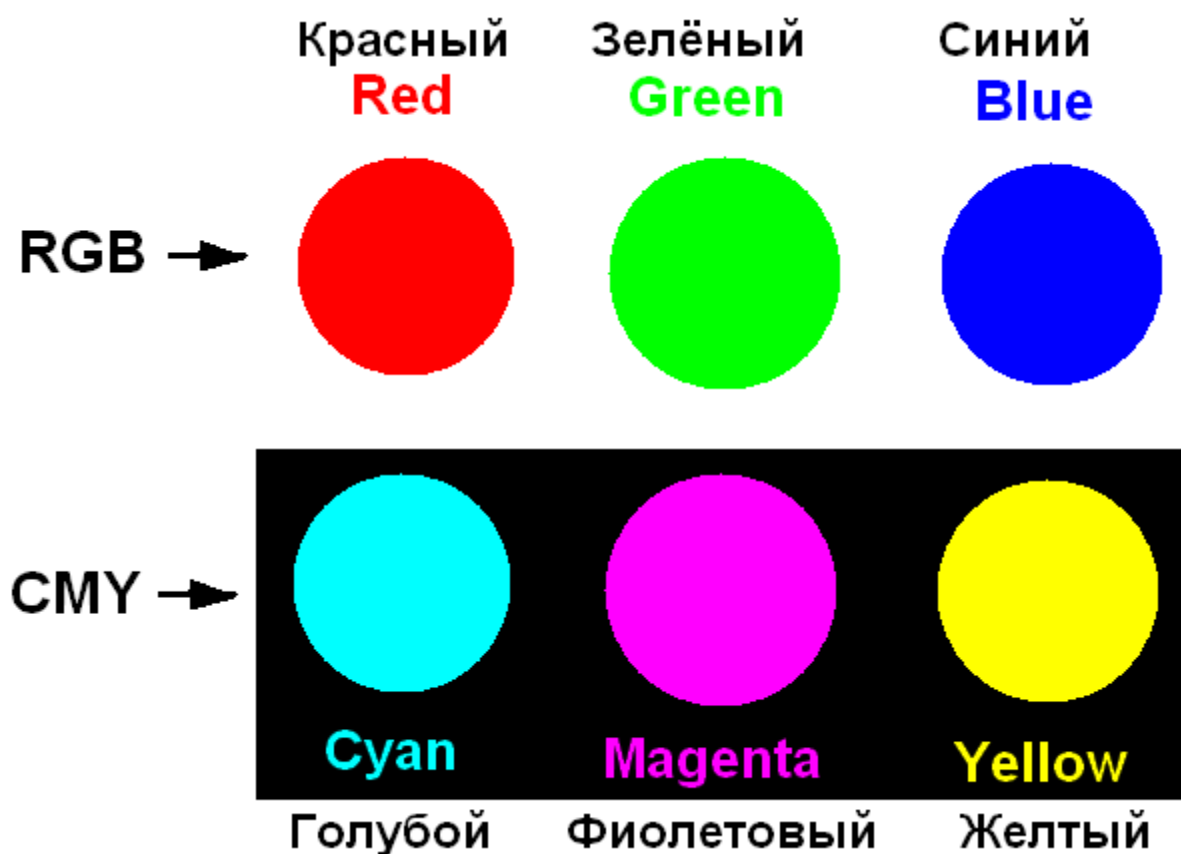


Рисунок 6.7 – Обратимые модели цветов

Значения  $C=M=Y=0$  говорят о том, что на белую бумагу не наносится никакая краска, поэтому все лучи отражаются, это белый цвет.

При наложении голубой, фиолетовой и желтой красок теоретически должен получиться черный цвет (рисунок 6.6,б), все лучи поглощаются. Однако на практике краски не идеальны, поэтому вместо черного цвета получается грязно-коричневый. Кроме того, при печати черных областей приходится «выливать» тройную порцию краски в одно место. Нужно также учитывать, что обычно на принтерах часто распечатывают черный текст, а цветные чернила значительно дороже черных.

Чтобы решить эту проблему, в набор красок добавляют черную, это так называемый *ключевой* цвет (англ. **Keycolor**), поэтому получившуюся модель обозначают **CMYK**.

Кроме цветовых моделей RGB и CMY (CMYK), существуют и другие. Наиболее интересная из них – модель **HSB** (англ. Hue – тон, оттенок; Saturation – насыщенность, Brightness – яркость),

которая ближе всего к естественному восприятию человека. Тон – это, например, синий, зеленый, желтый. Насыщенность – это чистота тона, при уменьшении насыщенности до нуля получается серый цвет. Яркость определяет, насколько цвет светлый или темный. Любой цвет при снижении яркости до нуля превращается в черный.

### 6.3 Особенности растрового кодирования

При растровом кодировании рисунок разбивается на пиксели (дискретизируется). Для каждого пикселя определяется цвет, который чаще всего кодируется с помощью RGB-кода.

Растровое кодирование имеет **достоинства**:

- ✓ универсальный метод (можно закодировать любое изображение);
- ✓ единственный метод для кодирования и обработки размытых изображений, не имеющих четких границ, например, фотографий;

и **недостатки**:

- ✓ при дискретизации всегда есть потеря информации;
- ✓ при изменении размеров изображения искажается цвет и форма объектов на рисунке, поскольку при увеличении размеров надо как-то восстановить недостающие пиксели, а при уменьшении – заменить несколько пикселей одним;
- ✓ размер файла не зависит от сложности изображения, а определяется только разрешением и глубиной цвета; как правило, растровые рисунки имеют большой объем.

Существует много разных форматов растровых рисунков. Чаще всего встречаются следующие расширения в именах файлов:

**.bmp** (англ. bitmap – битовая карта) – стандартный формат в операционной системе Windows; поддерживает кодирование с палитрой и в режиме истинного цвета;

**.jpg** или **.jpeg** (англ. Joint Photographic Experts Group – объединенная группа фотографов-экспертов) – формат, разработанный специально для кодирования фотографий; поддерживает только режим истинного цвета; для уменьшения объема файла

используется сильное сжатие, при котором изображение немного искажается, поэтому не рекомендуется использовать его для рисунков с четкими границами;

**.gif** (англ. Graphics Interchange Format – формат для обмена изображениями) – формат, поддерживающий только кодирование с палитрой (от 2 до 256 цветов); в отличие от предыдущих форматов, части рисунка могут быть прозрачными; в современном варианте можно хранить анимированные изображения; используется сжатие без потерь, то есть при сжатии изображение не искажается;

**.png** (англ. Portable Network Graphics – переносимые сетевые изображения) – формат, поддерживающий как режим истинного цвета, так и кодирование с палитрой; части изображения могут быть прозрачными и даже полупрозрачными (32-битное кодирование RGBA, где четвертый байт задает прозрачность); изображение сжимается без искажения; анимация не поддерживается.

## 6.4 Замечание о кодировании файла

Ранее было сказано, что все виды информации хранятся в памяти компьютера в виде двоичных кодов, то есть цепочек из нулей и единиц. Получив такую цепочку, абсолютно невозможно сказать, что это – текст, рисунок, звук или видео. Например, код  $11001000_2$  может обозначать число 200, букву «И», одну из составляющих цвета пикселя в режиме истинного цвета, номер цвета в палитре для рисунка с палитрой 256 цветов, цвета 8 пикселей черно-белого рисунка и т. п. Как же компьютер разбирается в двоичных данных? В первую очередь нужно ориентироваться на расширение имени файла. Например, чаще всего файлы с расширением .txt содержат текст, а файлы с расширениями .bmp, .gif, .jpg, .png – рисунки.

Однако расширение файла можно менять как угодно. Например, можно сделать так, что текстовый файл будет иметь расширение .bmp, а рисунок в формате JPEG – расширение .txt. Поэтому в начало всех файлов специальных форматов (кроме простого текста, .txt) записывается заголовок, по которому можно «узнать» тип файла и его характеристики. Например, файлы в формате BMP начинаются с символов «BM», а файлы в формате GIF – с символов «GIF». Кроме того, в заголовке указывается размер рисунка и его характеристики, например, количество цветов в палитре, способ сжатия и т.п. Используя эту информацию, программа декодирует (расшифровывает) основную часть файла и выводит его на экран.

## 6.5 Вопросы и задания

1. Какие два принципа кодирования рисунков используются в компьютерной технике?
2. Почему не удастся придумать единый метод кодирования рисунков, пригодный во всех ситуациях?
3. В чем состоит идея растрового кодирования?  
Что такое растр?
4. В чем состоит идея векторного кодирования? Что такое графический примитив?
5. Что такое пиксель? Как образовалось такое слово?
6. Что такое дискретизация рисунка? Почему она необходима?
7. Что теряется при дискретизации рисунка?
8. Что такое разрешение (экрана, принтера)? В каких единицах оно измеряется?
9. Что такое глубина цвета? В каких единицах она измеряется?
10. Что такое режим истинного цвета (True Color)?
11. Что такое режим «высокого» цвета (High Color)?
12. Что такое кодирование с палитрой? В чем его принципиальное отличие от режима истинного цвета?

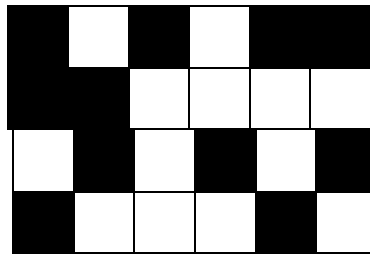
13. В чем состоят достоинства и недостатки растрового кодирования?

14. В чем состоят достоинства и недостатки векторного кодирования?

15. В каких форматах целесообразно сохранять фотографии?

16. В каких форматах целесообразно сохранять чертежи, рисунки с четкими границами?

17. Как запишется код следующего рисунка? Черно-белое растровое изображение кодируется построчно, начиная с левого верхнего угла и заканчивается в правом нижнем углу. При кодировании 0 обозначает черный цвет, а 1 – белый.



Решение. Запись кода изображения будет следующая:

$$010100 \ 011111 \ 101010 \ 011101 = 010100011111101010011101_2 = \\ = 24375235_8 = 51FAD_{16}.$$

### **Список использованных источников**

1. Андреева Е. В. Математические основы информатики: учеб. пособие / Е. В. Андреева, Л. Л. Босова, И. Н. Фалина. – М. : БИНОМ. Лаборатория знаний, 2007.
2. Пospelов Д. А. Арифметические основы вычислительных машин дискретного действия / Д. А. Пospelов. – М. : Энергия, 1970.
3. Савельев А. Я. Арифметические и логические основы цифровых автоматов / А. Я. Савельев. – М. : Высшая школа, 1980.
4. Поздняков С. Н. Дискретная математика: учебник / С. Н. Поздняков, С. В. Рыбин. – М. : Академия, 2008.
5. Хартли Р. В. Л. Передача информации / Р. В. Л. Хартли // Теория информации и ее приложения. – М. : Физматгиз, 1959.
6. Шеннон К. Математическая теория связи. (Shannon C.E. A Mathematical Theory of Communication. Bell System Technical Journal. – 1948. – с. 379-423, 623-656).
7. Юшкевич А. П. История математики в средние века / А. П. Юшкевич. – М. : Физматгиз, 1961.

## **Приложение**



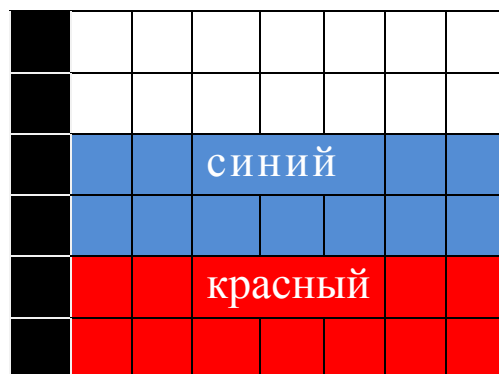
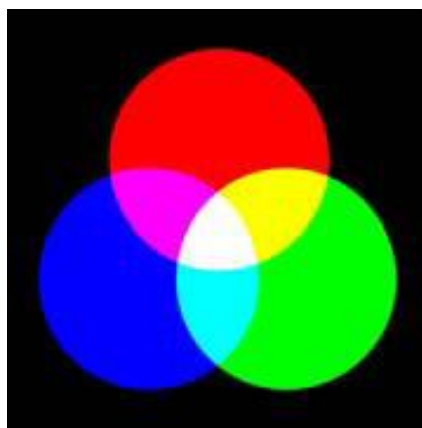


Рисунок 6.4 – Цветной растровый рисунок

а)



б)

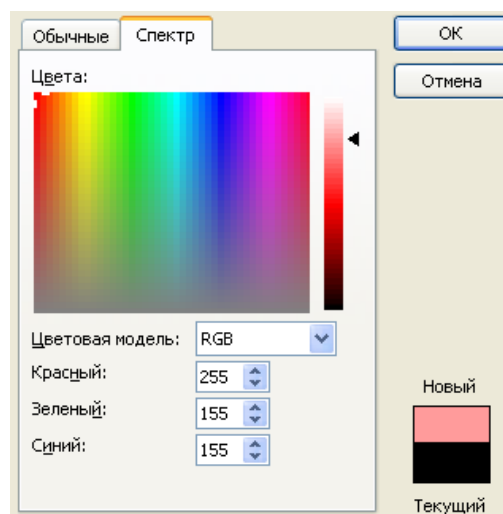
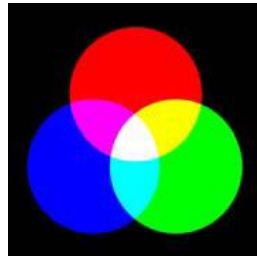


Рисунок 6.5 – Модель цвета RGB;  
 а) модель RGB; б) инструмент “цвет текста” в MSWord

а)



б)

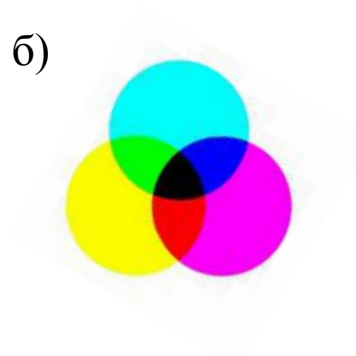


Рисунок 6.6 – Модели цветов;  
а) модель RGB (для монитора); б) модель CMY (для принтера)

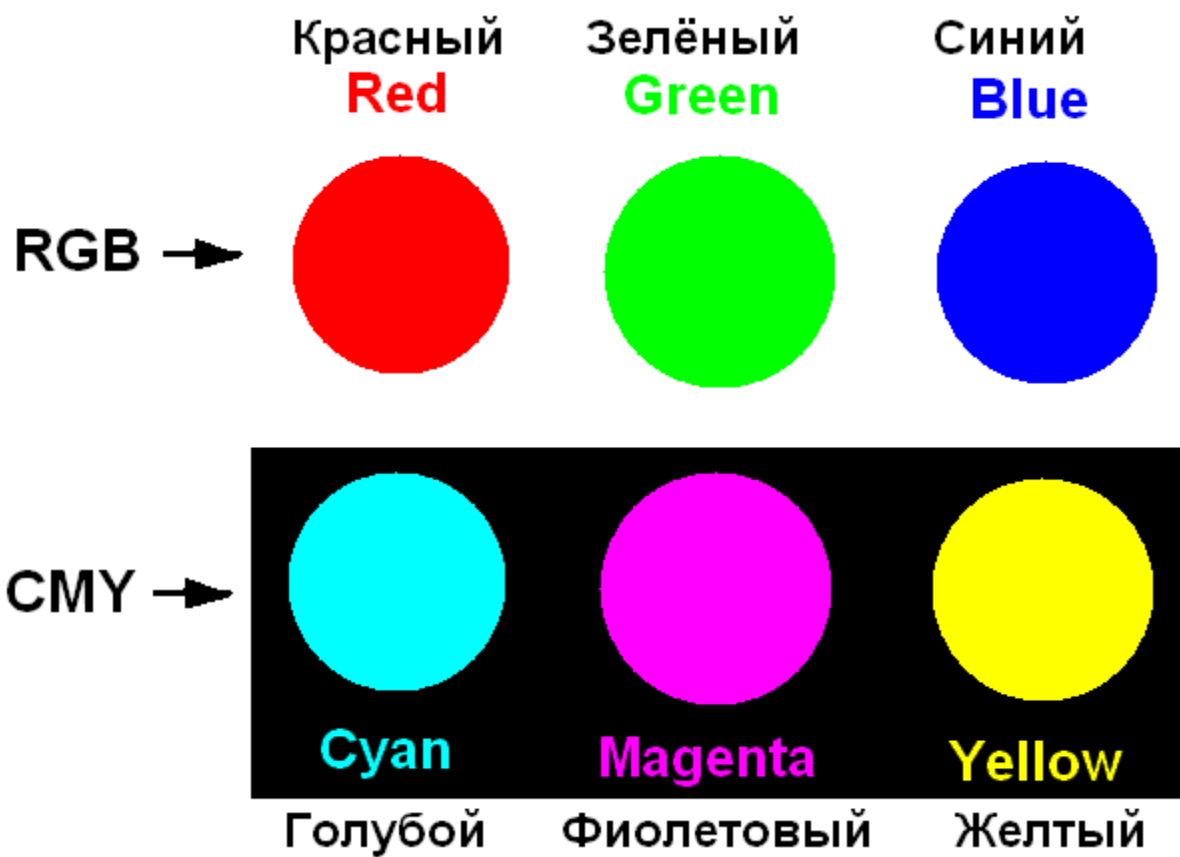


Рисунок 6.7 – Обратимые модели цветов

Учебное издание

Галиев Карим Сулейманович  
Печурина Елена Каримовна

**Двоичная система  
и представление информации  
в компьютере**

*Учебно-методическое пособие*

Под редакцией д-ра техн. наук, профессора В. И. Лойко

Подписано в печать 19.12.2014. Формат  $60 \times 84 \frac{1}{16}$ .

Усл. печ. л. – 6,2. Уч.-изд. л. – 4,9.

Тираж 200 экз. Заказ № \_\_\_\_\_

Типография Кубанского государственного  
аграрного университета.

350044, г. Краснодар, ул. Калинина, 13