Федеральное агентство по образованию
Государственное образовательное учреждение высшего образования
Московский государственный политехнический университет

Кафедра: Прикладная информатика

Дисциплина: «Информационные технологии»

Лекция №1 Понятия информации и системы счисления

Москва, 2016

Понятие ***количества информации*** в статистической теории информации определяется на основе понятия вероятности, которое применяется для описания систем с неопределенностью.

Информация – это сведения или сообщения, которые снимают существовавшую до их получения неопределенность

Степень неопределенности измеряется величиной, которую в теории информации называют **энтропией** (**H**).

Энтропия является функцией вероятности (**p**): $H=-\log_2 p$

При **p = 1** энтропия равна нулю, неопределенность полностью отсутствует.

В том случае, если в результате получения какого-то сообщения неопределенность полностью исчезает, количество информации (**I**) в этом сообщении равно энтропии **I=H.**

Количество информации, равное единице **H=1**, принято называть **битом**.
Свое название **бит** (bit) единица измерения информации получила от английских слов binary digit (двоичная цифра).

Информация является положительной если неопределенность уменьшается и отрицательной если неопределенность увеличивается.

Рассмотрим в качестве примера сообщение “Do not lean of the window”, данное сообщение содержит 27 позиций и должна быть заполнена 27 символами (20 букв и 7 пробелов).

Следует отметить, что вероятность появления каждого символа на каждой позиции, не равна 1/27. Это связано с тем, что различные буквы латинского алфавита имеют различную частоту повторяемости в английском языке, а следовательно, и вероятность появления на каждой позиции. Для простоты примера будем считать вероятность появления каждого символа равной 1/27.

Количество информации, которое несет каждый символ на каждой позиции, равно:

H = − log2 1/27= − log2 1 + log2 27=log2 27 = 4,75 бита (на позицию).

Общее количество информации, содержащееся в сообщении, можно определить как:

H = 25 log2 27 = 119 бит (всего).

Системы счисления

Совокупность приемов наименования и изображения количественных величин с помощью ограниченного набора знаков будем называть **системой счисления**.

В настоящее время используются два вида систем счисления: **позиционные** и **непозиционные системы счисления**.

Наиболее известной непозиционной системой счисления является римская система счисления. В ней запись различных целых количеств производится с помощью цифр (1, 5, 10, 50, 100, 500 и 1000) **I**, **V**, **X**, **L**, **C**, **D**, **M** и т.д.

Запись **MCMXCVII** расшифровывается следующим образом: **MCMXCVII = M + CM + XC + VII** и означает число одна тысяча девятьсот девяносто семь.

Позиционные системы счисления

Системы счисления, в которых вклад каждой цифры в величину числа зависит от ее положения (позиции) в записи числа, называются **позиционными**.

Пример – десятичная система счисления: в числе 102 цифра 2 означает две единицы, а в числе 120 – два десятка.

Используя позиционный принцип, можно изобразить любое целое десятичное число с помощью десяти цифр (0, 1, 2, 3, 4, 5, 6, 7, 8, 9) в их различных комбинациях.

Общее правило такого представления:

$$z = a_n \cdot 10^n + a_{n-1} \cdot 10^{n-1} + \ldots + a_1 \cdot 10^1 + a_0 \cdot 10^0,$$

a_n, a_{n-1}, ..., a_1, a_0 – целые числа в пределах от нуля до девяти (основание десятичной системы счисления равно 10).

Пример:

3 2 1 0

$3247 = 3 \cdot 10^3 + 2 \cdot 10^2 + 4 \cdot 10^1 + 7 \cdot 10^0$.

Любое целое десятичное число z может быть представлено в системе счисления с основанием p единственным образом $z_p = a_n a_{n-1} \ldots a_1 a_0$, а его десятичное значение определяется по следующему правилу:

$$\mathbf{z = a_n \cdot p^n + a_{n-1} \cdot p^{n-1} + \ldots + a_1 \cdot p^1 + a_0 \cdot p^0}$$

$\mathbf{(0 \le a_i < p, i = 0, 1, \ldots, n)}$, где a_n, a_{n-1}, ..., a_1, a_0 – цифры в представлении числа в p-ичной системе счисления.

Примеры:

$11103 = 1 \cdot \mathbf{3^3} + 1 \cdot \mathbf{3^2} + 1 \cdot \mathbf{3^1} + 0 \cdot \mathbf{3^0} = 39,$

$1739 = 1 \cdot \mathbf{9^2} + 7 \cdot \mathbf{9^1} + 3 \cdot \mathbf{9^0} = 147.$

Основание системы счисления, в которой записано число, обозначается нижним индексом (в случае 10-ичной системы, как правило, не указывается).

Чем больше основание системы счисления, тем короче запись числа в ней.

Наибольший интерес при работе с ЭВМ представляют системы счисления с основаниями 2, 8 и 16 (т.е. равными степени двойки).

Система счисления	Основание	Алфавит
2-ичная	2	0,1
8-ричная	8	0, 1, 2, 3, 4, 5, 6, 7
16-ричная	16	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F A ~ 10 B ~ 11 C ~ 12 D ~ 13 E ~ 14 F ~ 15

Примеры:

$1111011_2 = 1 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 123,$

$173_8 = 1 \cdot 8^2 + 7 \cdot 8^1 + 3 \cdot 8^0 = 123,$

$7B_{16} = 7 \cdot 16^1 + 11 \cdot 16^0 = 123.$

При переводе числа из одной системы счисления в другую количественное значение числа не изменяется, а меняется только форма записи числа.

Перевод чисел из произвольной системы счисления в десятеричную выполняется непосредственным вычислением по формуле:

$$z_p = a_n \cdot p^n + a_{n-1} \cdot p^{n-1} + \ldots + a_1 \cdot p^1 + a_0 \cdot p^0$$

Для перевода целого числа z из 10-ичной системы счисления в произвольную ричную применяется следующий алгоритм:

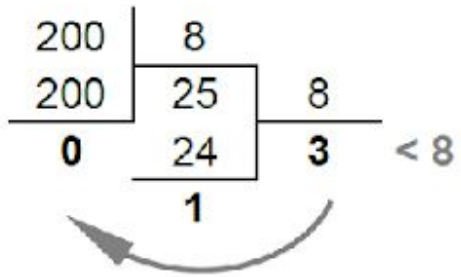
1) Определяется частное и остаток от деления числа z на p. Остаток будет очередной цифрой ai (i = 0, 1, 2, …) записи числа в новой системе счисления.

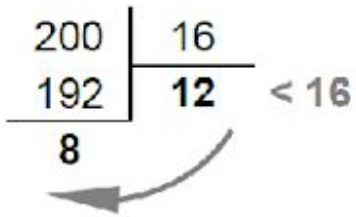
2) Если частное равно нулю, то перевод числа закончен, иначе к частному применяется пункт 1.

3) Цифры ai в записи числа нумеруются справа налево. Если p>10, то для цифр с числовыми значениями, большими или равными 10, вводятся специальные обозначения.

Пример:

$$200 = 310_8 = C8_{16}.$$





Перевод правильных дробей

Для перевода правильных дробей из одной системы счисления в другую, необходимо умножить исходную дробь и вновь полученные дробные части до получения числа заданной точности (Нулевой дробной части).

Пример:

а) 10 →2

0, 6875	x 2
1, 3750	x 2
0, 7500	x 2
1, 5000	x 2
1, 0000	

б) 10 →8

0, 6875	x 8
5, 5000	x 8
4, 0000	

в) 10 →16

0, 6875	x 16
11, 0000	

11=B

Арифметические операции с системами счисления

Арифметические операции в различных системах счисления выполняются по тем же правилам, что и в десятичной системе счисления. Так, если при выполнении операции сложения двух целых чисел результат поразрядного сложения в каждом разряде меньше основания системы счисления, т.е.

$$x(i) + y(i) < p,$$

где **i = -k, …, -1, 0,1,2,...n**, а **p** - основание системы счисления, то в **i**-й разряд суммы **z(i)** записывается цифра, которая отображает количество, равное

$$z(i) = x(i) + y(i).$$

В том случае, если результат поразрядного сложения больше основания системы счисления или равен ему, т.е.

$$x(i) + y(i) \geq p,$$

то в **i-й** разряд суммы записывается цифра, которая отображает количество, равное

$$z(i) = x(i) + y(i) - p.$$

При этом в следующий разряд суммы **z(i+1)** переносится единица, которая должна учитываться при суммировании в **(i+1)-м** разряде. Это правило можно изобразить с помощью алгоритма поразрядного сложения двух чисел. Его блок-схема представлена на **рисунке 1**.

Операция вычитания является обратной к операции сложения. Поэтому при вычитании числа Y из числа X поступают по аналогичным правилам. То есть, если разряд вычитаемого меньше разряда уменьшаемого

$$x(i) > y(i),$$

то в **i-й** разряд частного записывается цифра, которая означает количество

$$r(i) = x(i)-y(i).$$

Если разряд вычитаемого будет больше разряда уменьшаемого, то в **i-й** разряд частного записывается цифра, обозначающая количество, равное

$$r(i) = x(i) + p - y(i),$$

а значение старшего разряда **x(i+l)** уменьшается на единицу.

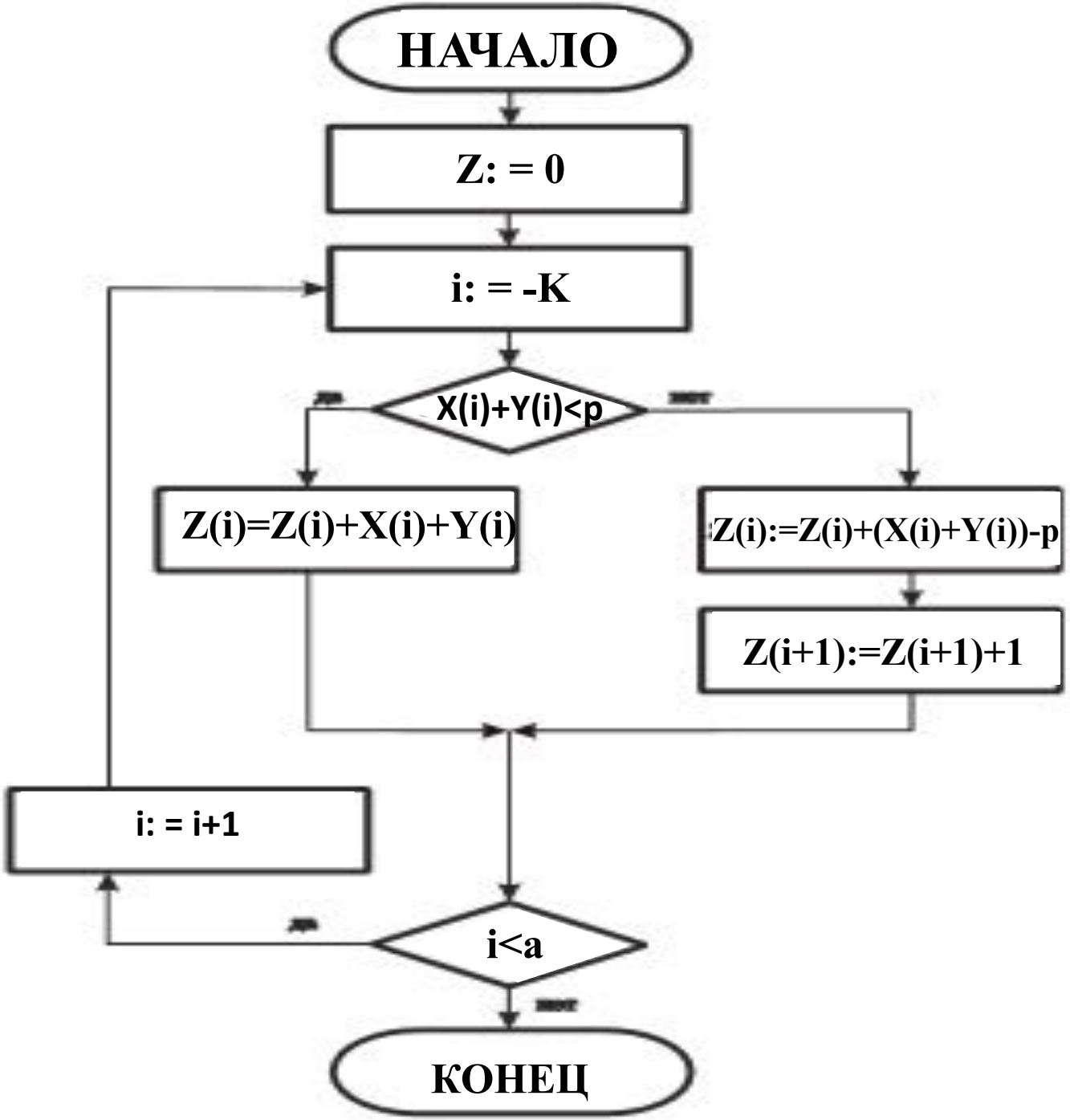


Рисунок 1. Блок-схема алгоритма сложения двух чисел

Далее рассмотрим выполнение операций сложения и вычитания на примере шестнадцатеричной системы счисления.

Пусть X = 2FC(16), а Y = A3F(16).

Тогда сумма **X + Y** будет определяться следующим образом:

$$\begin{array}{rccc} & 2 & F & C \\ + & A & 3 & F \\ \hline & D & 3 & B \end{array}$$

При поразрядном сложении **C и F** получается количество **27**. Это количество не может быть записано с помощью цифр шестнадцатеричной системы счисления, т.е. возникает единица переноса в старший разряд. Следовательно, количество, которое будет записано в младшем разряде, равно **27 - 16 = 11**, т.е. B. Во втором разряде результат поразрядного сложения цифр с учетом единицы переноса из младшего разряда имеет вид:

15 + 3 + 1 = 19

Этот результат также больше основания системы счисления. Поэтому во второй разряд суммы запишется величина **19 - 16 = 3**, и появится единица переноса в третий разряд. С учетом этой единицы переноса в старшем разряде суммы должно быть отображено количество **10 + 2 + 1 = 13**. Это количество меньше основания системы счисления и может быть записано цифрой **D**.

Операции умножения и деления рассмотрим на примере восьмеричной системы счисления. Для определения частных поразрядных произведений составим таблицу умножения **(таблица 2).** В этой таблице номера строк и столбцов соответствуют значениям сомножителей. Значения произведений по заданным значениям сомножителей находятся на пересечении соответствующих строк и столбцов.

Наиболее просто арифметические операции выполняются в двоичной системе счисления. Эта простота вызвана тем, что в двоичной системе счисления всего две цифры **0 и 1**. Следовательно, максимальное количество, которое может быть записано в каждом разряде, равно одному. Количество, равное двум, записывается как **10**.

Таблицы сложения, вычитания и умножения двоичных чисел весьма просты. Каждая из них состоит всего из четырех строчек (таблица 3, 4 и 5).

Таблица 2 Восьмеричная таблица умножения

	0	1	2	3	4	5	6	7	10
0	0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7	10
2	0	2	4	6	10	12	14	16	20
3	0	3	6	11	14	17	22	25	30
4	0	4	10	14	20	24	30	34	40
5	0	5	12	17	24	31	36	43	50
6	0	6	14	22	30	36	44	52	60
7	0	7	16	25	34	43	52	61	70
10	0	10	20	30	40	50	60	70	100

Таблица 3 Двоичная таблица сложения

0	+	0	=	0
0	+	1	=	1
1	+	0	=	1
1	+	1	=	10

Таблица 4 Двоичная таблица вычитания

0	–	0	=	0
1	–	0	=	1
1	–	1	=	0
10	–	1	=	1

Таблица 5 Двоичная таблица умножения

0	•	0	=	0
0	•	1	=	0
1	•	0	=	0
1	•	1	=	1

С помощью этих таблиц операции сложения, вычитания, умножения и деления двоичных чисел выполняются так же, как и в десятичной системе счисления.

Пример :

Сложение:

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

Вычитание:

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

Умножение:

```
    1 1 1 0 1 1
  x     1 1 0 1
  -------------
    1 1 1 0 1 1
  + 1 1 1 0 1 1
+ 1 1 1 0 1 1
-------------------
1 0 1 1 1 1 1 1 1 1
```

Деление:

```
1 1 0 1 1 1 0 1 1 0 1 | 1 0 0 1
                      | ---------
                      | 1 1 0 0 0 1 0 1

  1 1 0 1 1 1 0 1 1 0 1
- 1 0 0 1
  -------
    1 0 0 1
  - 1 0 0 1
    -------
        1 0 1 1
      - 1 0 0 1
        -------
            1 0 0 1
          - 1 0 0 1
            -------
                0 0 0 0
```

Что касается умножения, то его можно выполнять как со сдвигом частичных произведений влево (умножение производить с младших разрядов множителей), так и со сдвигом вправо (получая частичные произведения при умножении начиная со старших разрядов множителей). Общее произведение в этом случае не изменится.

Способы представления чисел в памяти ЭВМ

При естественном представлении чисел для каждой двоичной цифры отводится один разряд в памяти вычислительной машины. *Совокупность разрядов для записи числа носит название* **ячейки памяти**.

Естественное представление числа требует, чтобы некоторое постоянное количество разрядов отводилось для хранения дробной части, а остальные – для хранения целой части числа. Для изображения знака числа в машине приняты следующие обозначения:

" + " = 0;

" – " = 1.

Схематичное изображение ячейки памяти при записи чисел с фиксированной точкой представлено на рисунке 2. В этой ячейке запятая (точка) фиксирована после четвертого разряда. Сама точка в ячейке не записывается.

Фиксированное положение точки при естественном представлении чисел определяет то, что такое представление числа носит название ***"С фиксированной точкой"***.

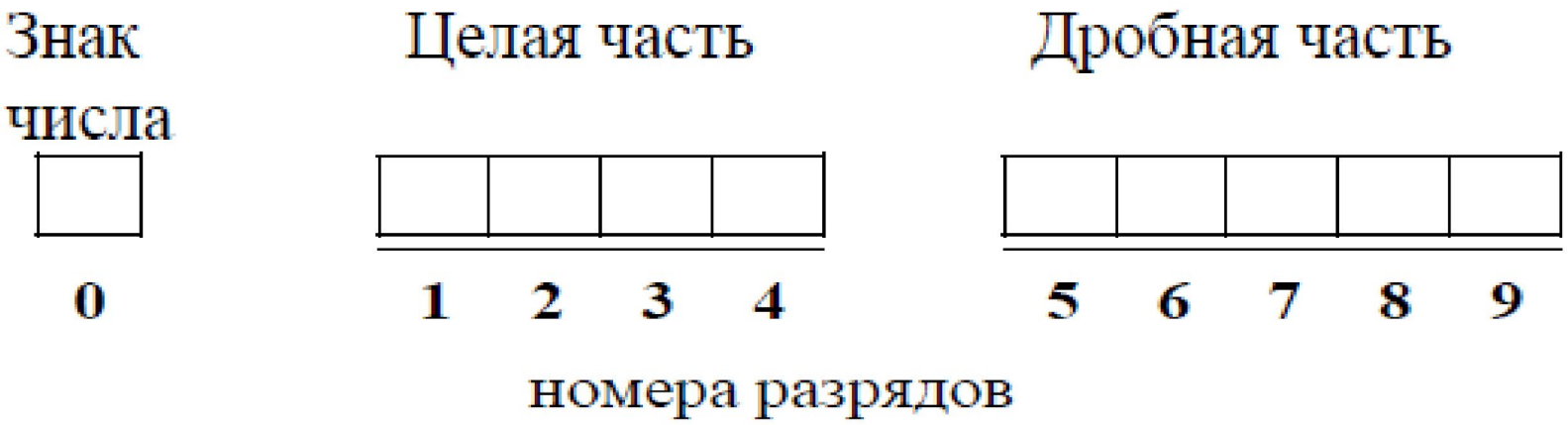


Рисунок 2. Схематичное изображение ячейки памяти при записи чисел с фиксированной точкой.

При выполнении арифметических операций с фиксированной точкой могут получаться результаты, целая часть которых содержит больше цифр, чем число разрядов ячейки, отведенных для хранения целой части.

При этом происходит переполнение разрядной сетки. Чтобы избежать этого, необходимо все величины, входящие в решаемую задачу, умножить заранее на множители, подобранные с таким расчетом, чтобы в процессе вычислений ***разрядная сетка*** не переполнялась. Эти множители носят название ***масштабных коэффициентов.***

Подбор масштабных коэффициентов представляет собой нелегкую работу, для упрощения которой в большинстве ЭВМ запятая фиксируется непосредственно после знакового разряда (рисунок 3). При таком представлении в ячейке могут быть записаны только правильные дроби.

Представление чисел с фиксированной точкой в виде правильных дробей имеет ряд преимуществ при выполнении арифметических операций. Так, при сложении двух чисел переполнение может произойти только на один разряд, возникнет единица переноса в знаковый разряд. Чтобы зафиксировать результат переполнения, для записи знака числа надо отводить не один, а два разряда:

" + " = 00;

" – " = 11.

В этом случае при переполнении в знаковых разрядах результата будут записаны противоположные значения: "**01**" или "**10**". Переполнения разрядной сетки при выполнении операции умножения вообще никогда не произойдет, т.к. перемножение двух правильных дробей всегда в результате дает правильную дробь.

Рисунок 3. Схематичное изображение ячейки памяти при записи чисел с точкой, фиксированной после знакового разряда.

Однако выполнение операций с фиксированной точкой сопровождается необходимостью использования масштабных коэффициентов. Это можно избежать если каждая ячейка памяти машины, кроме знаковых и цифрового разрядов, будет иметь и некоторое количество разрядов, образующих указатель положения точки (рисунок 4).

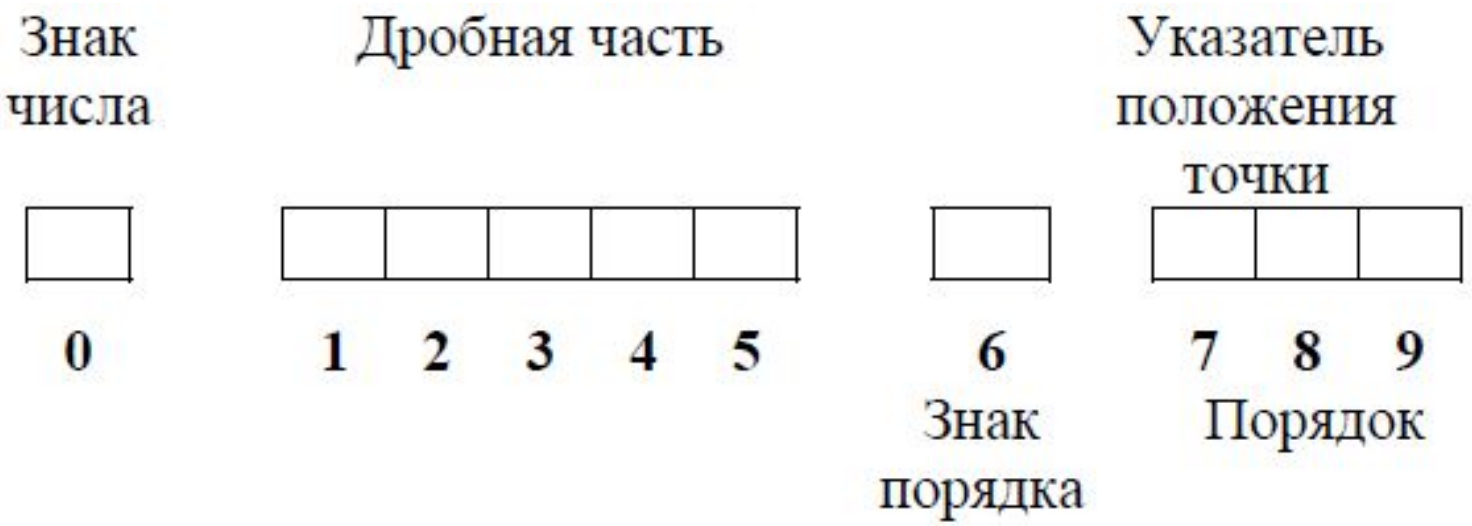


Рисунок 4. Схематичное изображение ячейки памяти при записи чисел с плавающей точкой.

Чтобы понять, как изображаются числа в такой ячейке, рассмотрим произвольное число **N**. Это число всегда может быть представлено в виде:

$$N = m \cdot q\ p,$$

где **q** – основание системы счисления, $|m| < 1$, а **p** – целое.

Например, десятичное число **364,78** может быть представлено как

$$364{,}78 = 0{,}36478 \cdot 10\ 3 = 0{,}036478 \cdot 10\ 4 = 0{,}0036478 \cdot 10\ 5$$

Двоичное число **1101,1001** может также быть представлено в виде:

$$1101{,}1001 = 0{,}11011001 \cdot 10\ 100 = 0{,}011011001 \cdot 10\ 101$$

(основание системы счисления и показатель степени записаны в двоичной системе счисления).

Такое представление чисел носит название записи ***"с плавающей точкой".***

Из приведенных примеров видно, что при записи числа с плавающей точкой его дробная часть, которая носит название ***мантиссы*** (**m**), может содержать много нулей между точкой и первой значащей цифрой. Поэтому, для экономии разрядов ячейки памяти машины для представления **мантиссы**, ее надо записывать так, чтобы после точки стояла значащая цифра, а не нуль, с соответствующей коррекцией ***порядка числа*** **p**. Такая запись числа носит название нормализованной.

Следовательно, запись числа в виде $N = m \cdot q^p$ называется ***нормализованной***, если $1/q \le |m| < 1$.

Все числа с плавающей точкой хранятся в памяти ЭВМ в нормализованном виде. Если в результате выполнения арифметических операций получается ненормализованное число, то оно подлежит обязательной нормализации.

При сложении мантисс двоичных чисел может быть получен ненормализованный результат, который устраняется сдвигом мантиссы влево на такое количество разрядов, при котором первой цифрой после запятой станет единица.