СИСТЕМЫ СЧИСЛЕНИЯ. ПРЕДСТАВЛЕНИЕ ЧИСЕЛ В КОМПЬЮТЕРЕ

Зав. кафедрой, к.ф.-м.наук
Тишков Артем Валерьевич
к.т.н. Никонорова Маргарита Леонидовна

Система счисления

– это совокупность правил именования и изображения чисел с помощью набора символов, называемых цифрами.

Используются три типа систем счисления:

- позиционная – представление числа зависит от порядка записи цифр.
- непозиционная – представление числа не зависит от порядка записи цифр
- смешанная – нет понятия «основание»: либо оснований несколько, либо оно вычисляемое

Позиционная, двоичная

- Логика: истина / ложь
- В повседневной жизни: Да / Нет
- В повседневной жизни: Есть / Нет
- В технике: электрический сигнал есть / нет
- 0 / 1, бит

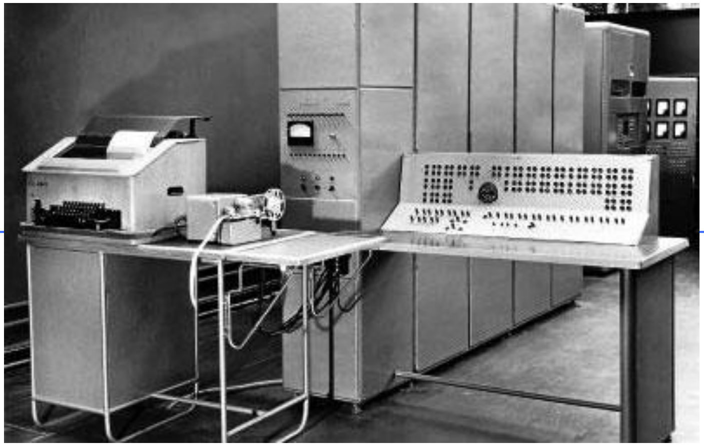
Позиционная, троичная. Симметричная и несимметричная

Десятичная система	−9	−8	−7	−6	−5	−4	−3	−2	−1	0	1	2	3	4	5	6	7	8	9
Троичная несимметричная	−100	−22	−21	−20	−12	−11	−10	−2	−1	0	1	2	10	11	12	20	21	22	100
Троичная симметричная	Ī00	Ī01	Ī1Ī	Ī10	Ī11	Ī Ī	Ī0	Ī1	Ī	0	1	1Ī	10	11	1ĪĪ	1Ī0	1Ī1	10Ī	100

- Трит – троичный триггер
- 1 Трайт = 6 тритов, 729 значений (байт – 256)
- Советская машина Сетунь – первая и единственная серийная троичная машина. 1962-1965 годы. Главный конструктор Николай Петрович Бруснецов

10Ī = 9 − 1 = 8

Ī01 = −9 + 1 = −8



Преимущества троичной симметричной системы (-1, 0, 1)

- естественное представление чисел со знаком (не нужен прямой, обратный или дополнительный код!)
- знак числа - это знак старшей ненулевой цифры и не нужен знаковый бит
- Простое сравнение чисел по величине, при этом не нужно смотреть на знак
  - поэтому команда ветвления по знаку в троичной машине работает вдвое быстрее, чем в двоичной
- усечение длины числа равносильно правильному округлению (округление в двоичных машинах не обеспечивают этого)
- троичный сумматор осуществляет вычитание при инвертировании одного из слагаемых, откуда следует, что троичный счетчик автоматически является реверсивным (обеспечивает и сложение и вычитание
- в трехвходовом троичном сумматоре перенос в следующий разряд возникает в 8 ситуациях из 27, а в двоичном сумматоре - в 4 из 8. В четырехвходовом сумматоре перенос также происходит только в соседний разряд.
- таблицы умножения и деления почти так же просты, как и в двоичной системе
- умножение на -1 инвертирует множимое
- трехуровневый сигнал более устойчив к воздействию помех в линиях передачи. Это означает что специальные методы избыточного кодирования троичной информации проще, нежели двоичной

Позиционные системы счисления

- Десятичная
- Двоичная
- Восьмеричная
- Шестнадцатеричная

Количество цифр называют основанием позиционной системы счисления, а позиции цифр в числе – разрядами.

- Троичная (электроника +,0,-)
- Двенадцатеричная (счет дюжинами)
- Шестидесятеричная (время, углы – широта долгота)

Непозиционные системы счисления

Представление через
биномиальные коэффициенты $x = \sum_{k=1}^{n} \binom{c_k}{k}$ $0 \le c_1 < c_2 < \ldots < c_n$

Система остаточных классов

Определяется набором взаимно простых *модулей* $(m_1, m_2, \ldots, m_n)$ с произведением $M = m_1 * m_2 * \ldots * m_n$ так, что каждому целому числу x из отрезка $[0, M-1]$ ставится в соответствие набор $(x_1, x_2, \ldots, x_n)$ вычетов

$$x \equiv x_1 \pmod{m_1};$$
$$x \equiv x_2 \pmod{m_2};$$
$$x \equiv x_n \pmod{m_n};$$

Римские цифры

I — 1	C — 100,
V — 5	D — 500,
X — 10	M — 1000
L — 50	

Строго говоря,
не является
непозиционной:
IV и VI – разные числа

Смешанные системы счисления

Дата – год, месяц, день

Время – часы, минуты, секунды, миллисекунды

при этом величина h часов, m минут, *s* секунд соответствует

$$T = h * 60 * 60 + m * 60 + s$$

Углы – градусы, минуты, секунды

Смешанной называется система счисления, в которой числа, заданные в некоторой системе счисления с основанием P (например – время) изображаются с помощью цифр другой системы счисления с основанием Q (например – секунды), где $Q<P$.

В такой системе P называется ***старшим основанием***, Q – ***младшим основанием***, *а сама система счисления называется Q–P–ичной.*

Представление числа в позиционной системе счисления

$$a_{n-1}a_{n-2}\ldots a_1a_0,\ a_{-1}\ldots a_{-m}$$

Запись чисел в каждой из систем счисления с основанием q означает сокращенную запись выражения

$$a_{n-1}q^{n-1} + a_{n-2}q^{n-2} + \ldots + a_1q^1 + a_0q^0 + a_{-1}q^{-1} + \ldots + a_{-m}q^{-m},$$

где a_i – цифры численной записи, соответствующие разрядам, i – индекс, n и m – количество разрядов числа целой и дробной части соответственно, q – основание системы счисления

Например, развернутая форма числа 327,46

n = 3, m = 2, q = 10

$$X = \sum_{i=2}^{-2} a_i q^i {}_2 * 10^2 + a_1 * 10^1 + a_0 * 10^0 + a_{-1} * 10^{-1} + a_{-2} * 10^{-2} =$$

$$3*10^2 + 2*10^1 + 7*10^0 + 4*10^{-1} + 6*10^{-2}$$

Алфавит

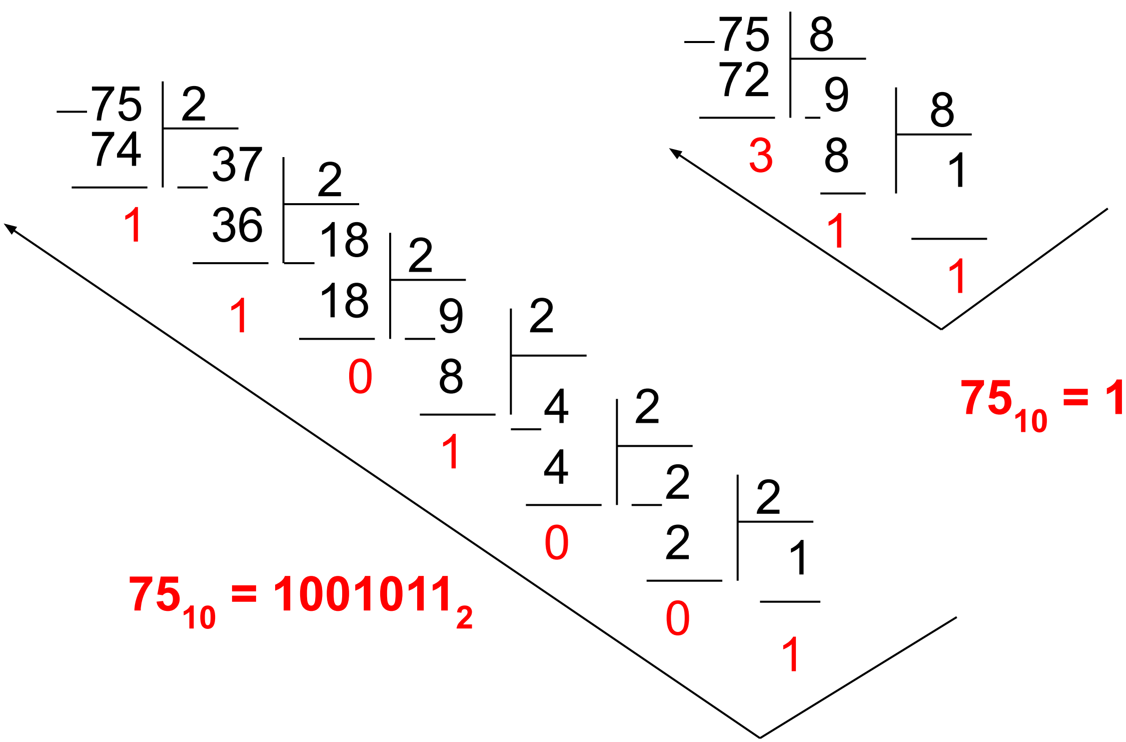
- Двоичная система счисления – 0, 1
- Восьмеричная система счисления – 0, 1, 2, 3, 4, 5, 6, 7
- Шестнадцатеричная система счисления –

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

10, 11, 12, 13, 14, 15

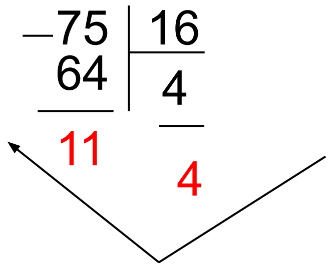
Перевод целых чисел из десятичной системы счисления

Пример. Перевести число 75 из десятичной системы счисления в двоичную, восьмеричную и шестнадцатеричную.



$75_{10} = 1001011_2$

$75_{10} = 113_8$



$75_{10} = 4B_{16}$

Перевод правильной десятичной дроби из десятичной системы счисления

Пример. Перевести число 0,35 из десятичной системы в счисления в двоичную, восьмеричную и шестнадцатеричную.

$$
\begin{array}{r}
\times\, 0{,}35 \\
2 \\
\hline
\color{red}{0},70 \\
\times\quad 2 \\
\hline
\color{red}{1},40 \\
\times\quad 2 \\
\hline
\color{red}{0},80 \\
\times\quad 2 \\
\hline
\color{red}{1},60 \\
\times\quad 2 \\
\hline
\color{red}{1},20
\end{array}
\qquad\qquad
\begin{array}{r}
\times\, 0{,}35 \\
8 \\
\hline
\color{red}{2},80 \\
\times\quad 8 \\
\hline
\color{red}{6},40 \\
\times\quad 8 \\
\hline
\color{red}{3},20
\end{array}
\qquad\qquad
\begin{array}{r}
\times\, 0{,}35 \\
16 \\
\hline
\color{red}{5},60 \\
\times\quad 16 \\
\hline
\color{red}{9},60
\end{array}
$$

$\mathbf{0{,}35_{10} = 0{,}01011_2}$ $\mathbf{0{,}35_{10} = 0{,}263_8}$ $\mathbf{0{,}35_{10} = 0{,}59_{16}}$

Перевод чисел в десятичную систему счисления

Пример. Перевести число 1011,1 из двоичной системы счисления в десятичную.

разряды 3 2 1 0 -1

число $1\ 0\ 1\ 1,\ 1_2 = 1\cdot2^3 + 0\cdot2^2 + 1\cdot2^1 + 1\cdot2^0 + 1\cdot2^{-1} = 11{,}5_{10}$

Пример. Перевести число 276,8 из восьмеричной системы счисления в десятичную.

разряды 2 1 0 -1

число $2\ 7\ 6,\ 5_8 = 2\cdot8^2 + 7\cdot8^1 + 6\cdot8^0 + 5\cdot8^{-1} = 190{,}625_{10}$

Пример. Перевести число 1F3 из шестнадцатеричной системы счисления в десятичную.

разряды 2 1 0

число $1\ F\ 3_{16} = 1\cdot16^2 + 15\cdot16^1 + 3\cdot16^0 = 499_{10}$

Перевод из восьмеричной и шестнадцатеричной системы счисления в двоичную

Заменить каждую цифру восьмеричного/шестнадцатеричного числа соответствующим трехразрядным/четырехразрядным двоичным кодом.

Пример. Перевести число $527{,}1_8$ в двоичную систему счисления.

$$527{,}1_8 = \underbrace{101}_{5}\ \underbrace{010}_{2}\ \underbrace{111}_{7},\ \underbrace{001}_{1}\ {}_2$$

Пример. Перевести число $1A3{,}F_{16}$ в двоичную систему счисления.

$$1A3{,}F_{16} = \underbrace{0001}_{1}\ \underbrace{1010}_{A}\ \underbrace{0011}_{3},{}_{1}\underbrace{111}_{F}\ {}_2$$

Перевод из двоичной системы счисления в восьмеричную и шестнадцатеричную

Для перехода от двоичной к восьмеричной/шестнадцатеричной системе счисления поступают следующим образом: двигаясь от запятой влево и вправо, разбивают двоичное число на группы по 3(4) разряда, дополняя, при необходимости, нулями крайние левую и правую группы. Затем каждую группу из 3(4) разрядов заменяют соответствующей восьмеричной/шестнадцатеричной цифрой.

Пример:

$0\,1\,0\,1\,0\,1\,0\,0\,1{,}1\,0\,1\,1\,1\,0_2 = 251{,}65_8$

2 5 1 5 6

$1\,0\,1\,0\,1\,0\,0\,1{,}1\,0\,1\,1\,1\,000_2 = A9{,}B8_{16}$

A 9 B 8

Перевод из восьмеричной системы счисления в шестнадцатеричную и обратно

При переходе из восьмеричной системы счисления в шестнадцатеричную и обратно вначале производится перевод чисел из исходной системы счисления в двоичную, а затем – в конечную систему .

Пример. Перевести число $527{,}1_8$ в шестнадцатеричную систему счисления.

$$527{,}1_8 = 000\ 101010111{,}011\ 0_2 = 157{,}6_{16}$$

1 5 7 6

Пример. Перевести число $1A3{,}F_{16}$ в восьмеричную систему счисления.

$$1A3{,}F_{16} = 110100011{,}1111\ 00_2 = 643{,}74_8$$

6 4 3 7 4

Арифметические операции в позиционных системах счисления

Правила выполнения основных арифметических операций в любой позиционной системе счисления подчиняются тем же законам, что и в десятичной системе.

При сложении цифры суммируются по разрядам, и если при этом возникает переполнение разряда, то производится перенос в старший разряд. Переполнение разряда наступает тогда, когда величина числа в нем становится равной или большей основания системы счисления.

При вычитании из меньшей цифры большей в старшем разряде занимается единица, которая при переходе в младший разряд будет равна основанию системы счисления

Если при умножении однозначных чисел возникает переполнение разряда, то в старший разряд переносится число кратное основанию системы счисления. При умножении многозначных чисел в различных позиционных системах применяется алгоритм перемножения чисел в столбик, но при этом результаты умножения и сложения записываются с учетом основания системы счисления.

Деление в любой позиционной системе производится по тем же правилам, как и деление углом в десятичной системе, то есть сводится к операциям умножения и вычитания.

Правила выполнения арифметических действий задаются таблицей:

сложение	вычитание	умножение
0 + 0 = 0	0 – 0 = 0	0 * 0 = 0
0 + 1 = 1	1 – 0 = 1	0 * 1 = 0
1 + 0 = 1	1 – 1 = 0	1 * 0 = 0
1 + 1 = 10	10 – 1 = 1	1 * 1 = 1

Сложение в позиционных системах счисления

Цифры суммируются по разрядам, и если при этом возникает избыток, то он переносится влево

двоичная система

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

1+1=2=2+**0**

1+0+0=**1**

1+1=2=2+**0**

1+1+0=2=2+**0**

1+1=2=2+**0**

Ответ: 100010_2

восьмеричная система

```
  1 1 1
  2 1 5 4
+   7 3 6
---------
  3 1 1 2
```

4+6=10=8+**2**

5+3+1=9=8+**1**

1+7+1=9=8+**1**

1+2=3

Ответ: 3112_8

шестнадцатеричная система

```
  1 1
  8 D 8
+ 3 B C
-------
  C 9 4
```

8+12=20=16+4

13+11+1=25=16+**9**

8+3+1=12=$\mathbf{C}_{16}$

Ответ: $C94_{16}$

Вычитание в позиционных системах счисления

При вычитании чисел, если цифра уменьшаемого меньше цифры вычитаемого, то из старшего разряда занимается единица основания

двоичная система

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

1-1=**0**

2-1=**1**

0-0=**0**

2-1=**1**

Ответ: 1010_2

восьмеричная система

$$\begin{array}{r} \scriptstyle 1\ \ \ 1\ \ \ \ \ \ \ \ \\ -\ 4\,3\,5\,0\,6 \\ 5\,0\,4\,2 \\ \hline 3\,6\,4\,4\,4 \end{array}$$

6-2=**4**

8-4=**4**

4-0=**4**

8+3-5=11-5=**6**

Ответ: 36444_8

шестнадцатеричная система

$$\begin{array}{r} \scriptstyle 1\ \ \ 1\ \ \ \ \ \\ -\ C\ 9\ 4 \\ 3\ B\ C \\ \hline 8\ D\ 8 \end{array}$$

16+4-12=20-12=**8**

16+8-11=24-11=**13=D_{16}**

11-3=**8**

Ответ: 848_{16}

Умножение в позиционных системах счисления

При умножении многозначных чисел в различных позиционных системах применяется алгоритм перемножения чисел в столбик, но при этом результаты умножения и сложения записываются с учетом основания системы счисления

двоичная система

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

1+1+1=3=2+**1**

1+1+1=3=2+**1**

1+1=2=2+**0**

Ответ: 101011111_2

восьмеричная система

```
      4 2
      2 1
 x  1 6 3
      6 3
  ---------
      5 3 1
      1
  1 2 6 2
  ---------
  1 3 3 5 1
```

3·3=9=8+**1**

6·3+1=19=16+**3**=2·8+**3**

1·3+2=**5**

6·3=18=16+**2**=8·2+**2**

6+5=11=8+**3**

6·6+2=38=32+**6**=4·8+**6**

6·1+4=10=8+**2**

Ответ: 13351_8

Деление в позиционных системах счисления

Деление в любой позиционной системе производится по тем же правилам, как и деление углом в десятичной системе. При этом необходимо учитывать основание системы счисления.

двоичная система

```
 100011 | 1110
-1110   |------
 ----   | 10,1
  -1110
   1110
   ----
      0
```

Ответ: $10{,}1_2$

восьмеричная система

```
 13351 | 163
-1262  |-----
 ----  | 63
  -531
   531
   ---
     0
```

Ответ: 63_8

Представление чисел в компьютере

формат с фиксированной запятой – целые числа
формат с плавающей запятой – вещественные числа

Целые числа без знака занимают в памяти один или два байта.

Целые числа со знаком занимают в памяти компьютера один, два или четыре байта, при этом самый левый (старший) разряд содержит информацию о знаке числа.

Применяются три формы записи (кодирования) целых чисел со знаком: *прямой код, обратный код и дополнительный код.*

Вещественные числа хранятся и обрабатываются в компьютере в формате с плавающей запятой. Этот формат базируется на экспоненциальной форме записи, в которой может быть представлено любое число.

Представление целых чисел в компьютере

Целые числа в компьютере могут представляться со знаком или без знака.

Целые числа без знака занимают в памяти один или два байта.

Формат числа в байтах	Запись с порядком	Обычная запись
1	$0 \ldots 2^8 - 1$	0 …255
2	$0 \ldots 2^{16} - 1$	0 …65535

Пример. Число $72_{10} = 1001000_2$ в однобайтовом формате

0	1	0	0	1	0	0	0

Целые числа со знаком занимают в памяти компьютера один, два или четыре байта, при этом самый левый (старший) разряд содержит информацию о знаке числа.
Знак «плюс» кодируется нулем, а «минус» - единицей

Формат числа в байтах	Запись с порядком	Обычная запись
1	$-2^{7} \dots 2^{7}-1$	-128 …127
2	$-2^{15} \dots 2^{15}-1$	-32 768 …32 767
4	$-2^{31} \dots 2^{31}-1$	- 2 147 483 648 …2 147 483 647

Пример. Число $-72_{10} = -1001000_2$ в однобайтовом формате

1	1	0	0	1	0	0	0

Пример. Число $62_{10} = 111110_2$ в однобайтовом формате

0	0	1	1	1	1	1	0

В компьютерной технике применяются три формы записи (кодирования) целых чисел со знаком:
прямой код, **обратный** код и **дополнительный** код.

Прямой код – чаше всего отводится 2 байта памяти (16 бит), в старший разряд записывается «0» если число положительное и «1» – если число отрицательное.

Обратный код – для положительных чисел совпадает с прямым кодом, для отрицательных чисел образуется из прямого кода заменой нулей единицами, а единиц – нулями, кроме цифр знакового разряда.

Пример. Число $-57_{10} = -111001_2$

прямой	**1**	0	0	0	0	0	0	0	0	0	1	1	1	0	0	1
обратный	**1**	1	1	1	1	1	1	1	1	1	0	0	0	1	1	0

Дополнительный код используется для представления отрицательных чисел, позволяет заменить арифметическую операцию вычитания операцией сложения, что существенно упрощает работу процессора и увеличивает его быстродействие.

Дополнительный код отрицательного числа A, хранящегося в n ячейках равен $2^n - |A|$. Образуется из обратного кода с последующим прибавлением единицы к его младшему разряду

	13_{10}	-12_{10}
Прямой код	**0**0001101	**1**0001100
Обратный код	–	**1**1110011
Дополнительный код	–	**1**1110011 + 1 **1**1110100

Отрицательные десятичные числа при вводе в компьютер автоматически преобразуются в обратный или дополнительный код и в таком виде хранятся, перемещаются и участвуют в операциях.

При выводе таких чисел из компьютера происходит обратное преобразование в отрицательные десятичные числа.

Определим диапазон чисел, которые могут храниться в оперативной памяти в формате *длинных целых чисел со знаком* (отводится 32 бита памяти): минимальное число $-2^{31} = -2147483648$

максимальное число $2^{31} - 1 = 2147483647$

☺ – простота и наглядность представления чисел, простота алгоритмов реализации арифметических операций

☹ – небольшой диапазон представляемых чисел, недостаточный для решения большинства прикладных задач

Представление вещественных чисел в компьютере

Любое число ***N*** в системе счисления с основанием ***q*** можно записать в виде $\mathbf{N = m \cdot} q^{p}$, где ***m*** называется **мантиссой** числа, а ***p*** – **порядком**.

Такой способ записи чисел называется **представлением числа с плавающей точкой**.

Характеристики форматов вещественных чисел

Формат числа	Диапазон абсолютных значений	Биты мантиссы	Размер в байтах (битах)
одинарный	10^{-45} … 10^{38}	23+1	4 (32)
вещественный	10^{-39} … 10^{38}	используется редко	6 (48)
двойной	10^{-324} … 10^{308}	52+1	8 (64)
расширенный	10^{-4932} … 10^{4932}	64+1	10 (80)

При записи числа выделяют разряды для хранения знака мантиссы, знака порядка, порядка и мантиссы.
Порядок и мантисса определяют диапазон изменения чисел и их точность.
Так, диапазон (порядок) и точность (мантисса) для формата чисел обычной точности (четырехбайтных): из 32 битов выделяют 8 для хранения порядка и 24 бита – для хранения мантиссы и ее знака.

При записи основания числа в десятичной системе можно говорить о нормализованной записи: мантиссу и порядок *q*-ичного числа записывают в системе счисления с основанием *q*.

Различают:

– научная нормализованная запись числа: $1 \le |m| < 10$, $q = 10$, $(3{,}5 \cdot 10^2)$

– инженерная нормализованная запись (информатика):

$$0{,}1 < |m| \le 1,\ q = 10,\ (0{,}35 \cdot 10^2)$$

– компьютерная нормализованная запись: $1 \le |m| < 10$, $q = 10(E)$, (3,5E2)

Нормализованная экспоненциальная запись числа

– это запись вида

$$N = m*p^q,$$

где *q* - целое число (положительное, отрицательное или ноль в десятичной системе счисления), а *m* – *p*-ичная дробь, у которой целая часть состоит из одной цифры.

При этом *m* называется **мантиссой** числа, *q* - **порядком** числа.

Примеры:

$3{,}1415926 = 0,\ 31415926 * 10^1$;

$1000=0{,}1 * 10^4$;

$0{,}123456789 = 0{,}123456789 * 10^0$;

$0{,}0000107_8 = 0{,}1078 * 8^{-4}$; (порядок записан в 10-й системе)

$1000{,}0001_2 = 0,\ 100000012 * 2^4$.

Смещенный порядок

Для того, чтобы не хранить знак порядка используется **смещённый порядок**, который рассчитывается по формуле 2^a*-1+ИП*, где *a* – число разрядов, отводимых под порядок (выделенных для представления порядка числа в формате с плавающей запятой), *ИП* – (истинный) порядок числа.

Пример:

Если истинный порядок равен – 5, тогда смещённый порядок для 4-байтового числа будет равен 127-5=122.

В компьютере, число с плавающей запятой представляется в виде набора отдельных двоичных разрядов, условно разделенных на **знак**, **порядок** и **мантиссу**. Так, в наиболее распространённом формате (стандарт IEEE 754) число с плавающей запятой имеет вид:

Знак (0 или 1)	Порядок + смещение	Мантисса *(m)*

Прибавление смещения позволяет записывать положительные и отрицательные порядки в виде положительных чисел.

Тип данных	Длина (бит)			Смещение порядка
	число	порядок	мантисса	
одинарная точность	32	8	23	127
двойная точность	64	11	52	1023
расширенная точность	80	15	64	16383

Рассмотрим пример записи числа с плавающей точкой:
- ► число +178.25
- ► в двоичной системе счисления +1011 0010.01=+1.0110 0100 1 x 2^{111}

Одинарная точность (32 бита, смещение порядка $127_{10} = 1111111_2$)		
Знак	1 бит	0
Порядок	8 бит	**111**+111 1111=**1000 0110**
Мантисса	23 бита	011 0010 0100 0000 0000 0000
Итого	32 бита	0**100 0011 0**011 0010 0100 0000 0000 0000
В 16-ричном формате		**43324000**

Алгоритм представления числа с плавающей запятой

- Перевести число из р-ичной системы счисления в двоичную;
- представить двоичное число в нормализованной экспоненциальной форме;
- рассчитать смещённый порядок числа;
- разместить знак, порядок и мантиссу в соответствующие разряды сетки.

Пример

- Представить число -25,625 в машинном виде с использованием 4 байтового представления (где 1 бит отводится под знак числа, 8 бит - под смещённый порядок, остальные биты - под мантиссу).
- 1. $25_{10}=100011_2$

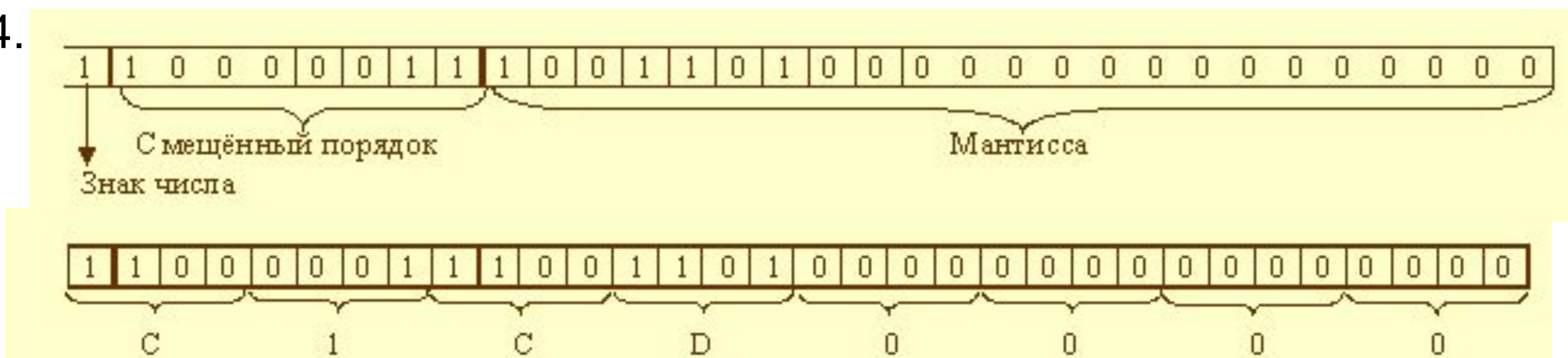
  $0{,}625_{10}=0{,}101_2$

  $-25{,}625_{10}= -100011{,}101_2$

  2. $-100011{,}101_2 = -1{,}00011101_2 * 2^4$

  3. СП=127+4=131

  4.



- Окончательный ответ: C1CD0000

Кодирование текстовой информации

Соответствие между набором символов и набором числовых значений называется **кодировкой символа**. При вводе в компьютер текстовой информации происходит ее двоичное кодирование. Код символа хранится в оперативной памяти компьютера. В процессе вывода символа производится обратная операция – **декодирование**, т.е.преобразование символа в его изображен

Добавлен слайд

Институтом стандартизации США была введена в действие система кодирования **ASCII** (*American Standard Code for Information Interchange*). Каждому символу ASCII соответствует 8–битовый двоичный код

(***1 символ – 1 байт***).

В последнее время широкое распространение получил новый международный стандарт **Unicode**. Стандарт состоит из двух основных разделов: универсальный набор символов (*UCS, universal character set*) и семейство кодировок (*UTF, Unicode transformation format*).
Каждому символу Unicode соответствует 16–битовый двоичный код

(***1 символ – 2 байта***).

Пример

С помощью кодировок ASCII и Unicode закодирована фраза:
«Я поступил в университет!».
Оцените информационный объем этой фразы.

Решение.
В данной фразе содержится 25 символов, включая пробелы и знак препинания.
В кодировке ASCII на 1 символ отводится 1 байт, следовательно для фразы понадобится 25 байт или 200 бит.
В кодировке Unicode 1 символ занимает 2 байта, поэтому вся фраза займет 50 байт или 400 бит.

Формула Хартли

Для измерения количества информации, которое может быть передано при помощи алфавита, существует формула Хартли

$\boxed{n = p^i}$, где *n* – число равновероятных событий, *i* – количество информации, полученной в результате совершения события,
p – количество различных вариантов
или
p – количество используемых символов, *i* – длина строки символов или сигналов.

Добавлен слайд

Пример:
Сколько различных сигналов можно записать с помощью 32-разрядного компьютерного кода.
В этом случае длина строки – 32, количество используемых символов – 2, следовательно $n = 2^{32} = 4\ 294\ 967\ 296$.

Кодирование графической информации

Примеры

1)В палитре растрового графического изображения 8 цветов, его размер 16x16 пикселей. Какой информационный объем имеет изображение?

Решение:

Глубину цвета определим по формуле $8 = 2^i$, I = 3

$In = 16 \times 16 \times 3 = 768$ бит = 96 байт

2)Для хранения растрового изображения размером 64x64 пикселя отвели 1,5 килобайта памяти. Каково максимальное возможное число цветов в палитре изображения?

Решение:

$In = 1{,}5$ Кб $= 1{,}5 \times 2^{10}$ байт $= 1{,}5 \times 2^{10} \times 8$ (бит)

$X*Y = 64 \times 64 = 2^6 \times 2^6 = 2^{12}$. Из формулы найдем глубину цвета

$I = (1{,}5 \times 10^{10} \times 2^3) / 2^{12} = 1{,}5 \times 2 = 3$. Общее количество цветов $2^3 = 8$