Лекция 3

ИНФОРМАЦИОННО-ЛОГИЧЕСКИЕ ОСНОВЫ ПОСТРОЕНИЯ ЭВМ

Воспоминания о прошлой лекции

Две формы представления чисел:

- С фиксированной точкой
- С плавающей точкой

$X=M*p^k$, p-основание системы счисления,

M-мантисса, $P^{-1}<=M<1$, k-порядок

Преобразование чисел из естественной формы в нормализованную

- <u>Число больше 1.</u>

Перемещение разделителя по числу влево до тех пор, пока не исчезнет целая часть. Нормализация влево. $N_{\leftarrow}$

$$N_{\leftarrow}[1234,56]=0.123456*10^{4}$$

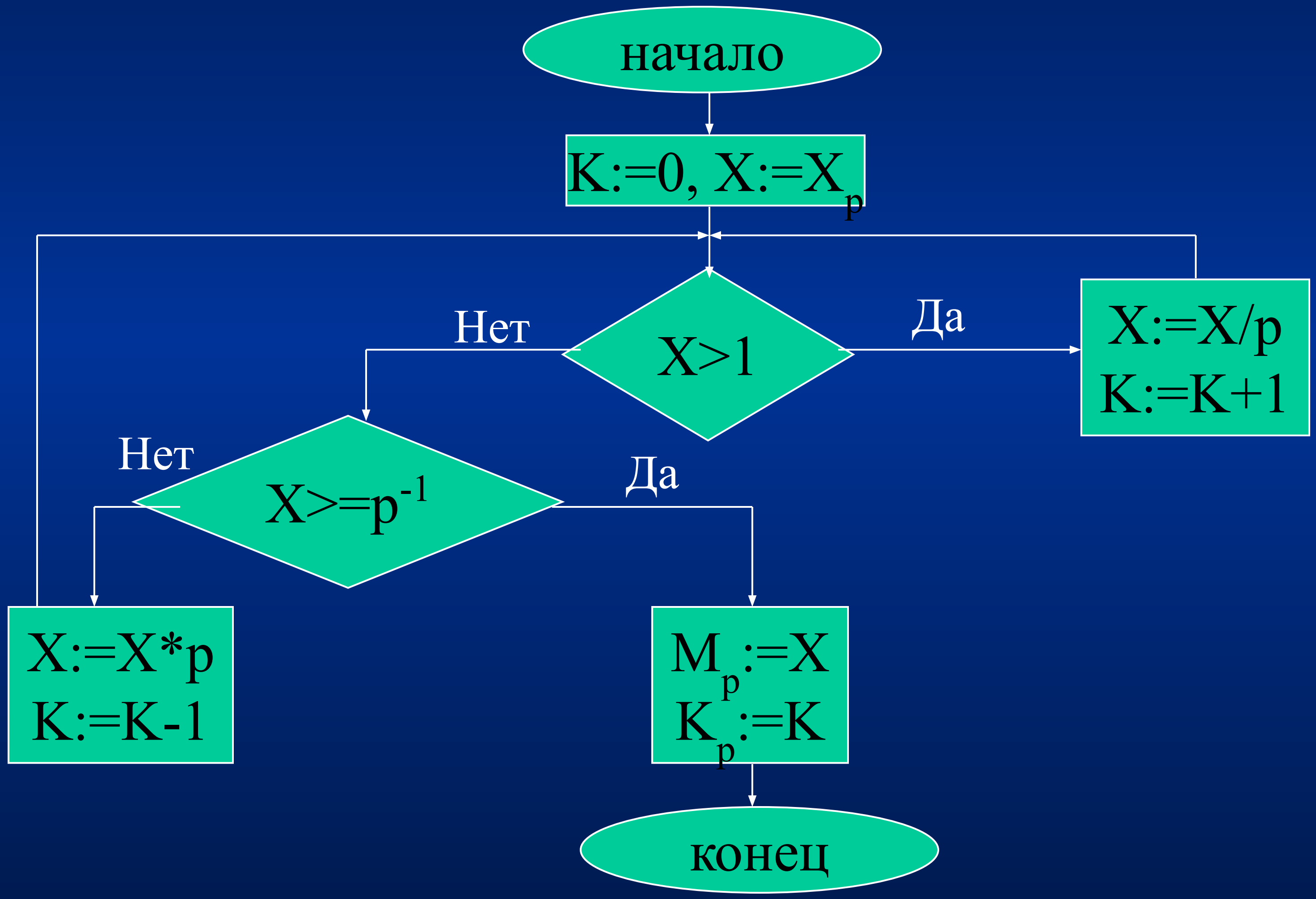
$$N_{\leftarrow}[23,4*10^{6}]=0.234*10^{7}$$

- <u>Число меньше 1.</u>

Перемещение разделителя по числу вправо до тех пор, пока первая цифра после разделителя не станет ненулевой. Нормализация вправо. $N_{\rightarrow}$

$$N_{\rightarrow}[0.0003]=0.3*10^{-3}$$

Общий алгоритм по нормализации числа



Необходимо хранить в ЭВМ

- Вещественное число
  - Знак числа
  - Мантисса
  - Знак порядка
  - Порядок

Способы кодирования чисел и допустимые над ними действия различны для следующих числовых множеств:

- целые положительные числа (без знака)
- целые со знаком
- вещественные нормализованные числа.

В ПК могут обрабатываться поля постоянной и переменной длины.

Поля постоянной длины:

- слово — 2 байта
- двойное слово — 4 байта
- полуслово — 1 байт
- расширенное слово — 8 байт
- слово длиной 10 байт — 10 байт

Числа с фиксированной запятой чаще всего имеют формат слова и полуслова, числа с плавающей запятой — формат двойного и расширенного слова.

Поля переменной длины могут иметь любой размер от 0 до 256 байт, но обязательно равный целому числу байтов.

Целые числа без знака.

$72_{10}=1001000_2$

Нумерация битов в байте

7	6	5	4	3	2	1	0
0	**1**	**0**	**0**	**1**	**0**	**0**	**0**

Размещение разрядов числа в байте

Целые числа без знака

$72_{10}=1001000_2$

Нумерация битов в байте

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

Размещение разрядов числа в байте

Целые числа со знаком

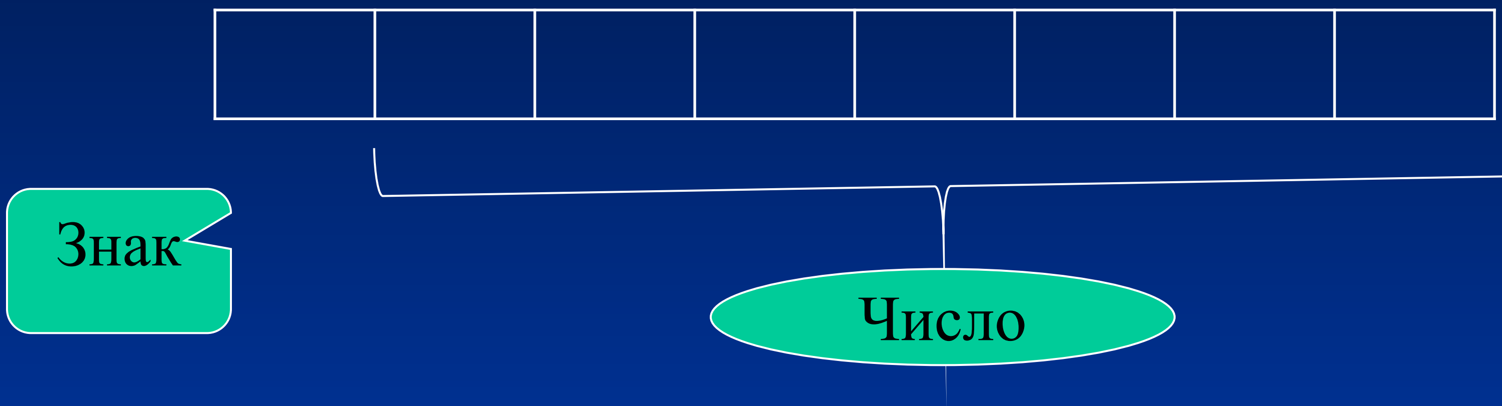
Прямой код

Обратный код

Дополнительный код

Смещенный код

Прямой код



Пример: 1 = 0000 0001, -1 = 1000 0001

$$A_{10} = (-1)^{a_{зн}} \sum_{i=0}^{n-2} a_i 2^i$$

n-разрядность кода, $a_{зн}$ - значение знакового разряда.

Пример: если разрядность кода равна 4, то

1101 = $(-1)^1[1x2^0+0x2^1+1x2^2]=-5$

Прямой, обратный, дополнительные коды

$$Pпр = \begin{cases} 0X, \text{при } x > 0 \\ 1X, \text{при } x < 0 \end{cases}$$

Где знак “+” –0, “–” – 1

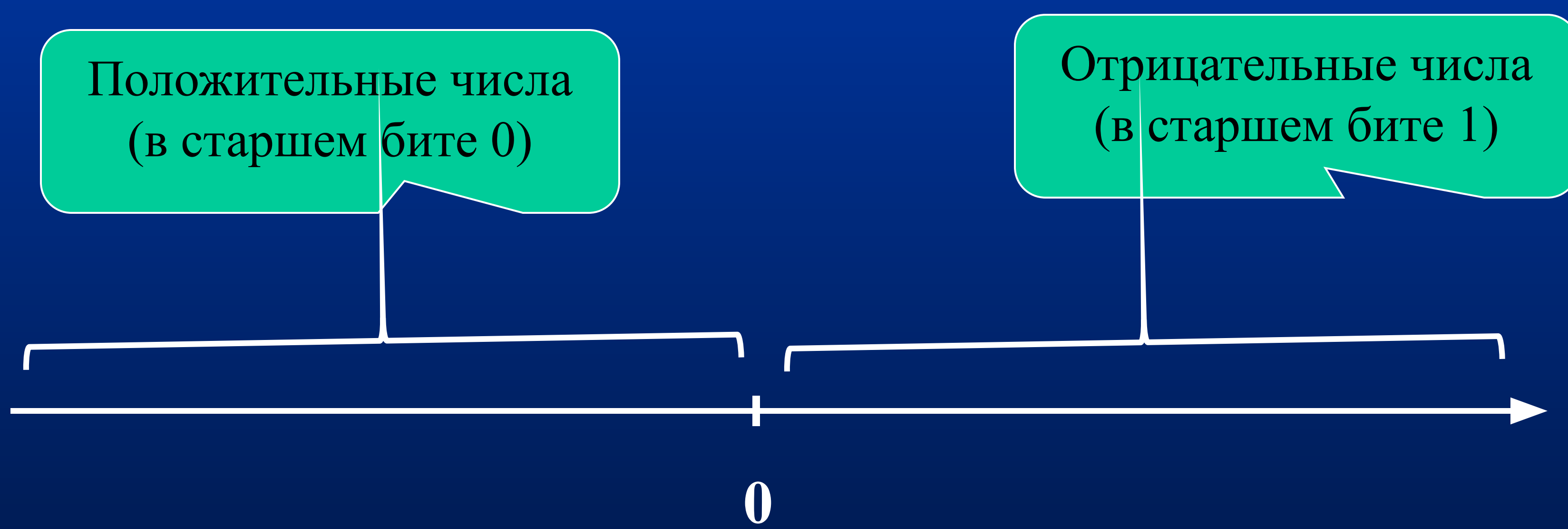
$$Pобр = \begin{cases} Pпр, \text{при } x > 0 \\ 1\overline{X}, \text{при } x < 0 \end{cases}$$

Дополнительный код

Идея: на примере десятичного вычитания **двухразрядных** чисел:

предположим, то надо выполнить вычитание 84-32 /результат **52**/. Дополним 32 до 100 /это «дополнение» равно 68/. Затем выполним сложение 84+68 /результат 1**52**/. Единица «уходит», потому что рассматривает **двухразрядные** десятичные числа.

Идея: в терминах двоичного представления чисел:



Дополнительный код

Представление в двоичном дополнительном коде в случае 8-битного кодирования чисел:

Набор битов	Значение
0000 0011	3
0000 0010	2
0000 0001	1
0000 0000	0
1111 1111	-1
1111 1110	-2
1111 1101	-3
1111 1100	-4

1. Начинают с цепочки от всех нулей 2. затем идут вверх до появления цепочки, состоящей из первого нуля и всех остальных единиц	Это положи-тельные 1,2,3, …
3. Затем вниз – начинают с цепочки из всех единиц, затем в обратном порядке идут до цепочки, состоящей из первой 1 и всех остальных нулей	Это -1,-2,-3,-4, ….

Дополнительный код

Для дополнительного кода справедливо следующее соотношение:

$$A_{10} = a_{зн}(-2^{n-1}) + \sum_{i=0}^{n-2} a_i 2^i$$

где *n*-разрядность машинного слова, $a_{зн}$ =0 для положительных чисел, $a_{зн}$ =1 для отрицательных чисел.

Пример: 1101 = 1*(-2^3)+[1x2^0+1x2^1+0x2^2]=-8+3=-5

Дополнительный код

алгоритм перевода отрицательных чисел в

I вариант. Переписать исходную последовательность битов числа справа налево до первой единицы, включая ее. Остальные биты инвертировать.	$6_{10}=0110_2$ $-6_{10}=\underline{1010}_2$
II вариант. Дополнительный код=логическое дополнение (все биты инвертированы)+1	$-6_{10}=1001_2+1$ $=\underline{1010}_2$

!!! Число + его дополнительный код =0

Обратный код

получается инвертированием всех цифр двоичного кода абсолютной величины числа

Пример: число: -1, модуль 00000001, обратный код 11111110

Для обратного кода справедливо следующее соотношение:

$$\boxed{A_{10} = a_{зн}(-2^{n-1}+1)+\sum_{i=0}^{n-2} a_i 2^i}$$

n-разрядность машинного слова, $a_{\text{зн}}$ =0 для положительных чисел, $a_{\text{зн}}$ =1 для отрицательных чисел.

1010 = 1*(-2^3+1)+[0x2^0+1x2^1+0x2^2]=

-7+2=-5

Число с фиксированной запятой формата слово со знаком:

Структурно запись числа -193(10) = -11000001(2) в разрядной сетке ПК выглядит следующим образом.

-193	Знак числа	Величина числа														
N разряда	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
прямой	1	0	0	0	0	0	0	0	1	1	0	0	0	0	0	1
обратный	1	1	1	1	1	1	1	1	0	0	1	1	1	1	1	0
Дополнительный	1	1	1	1	1	1	1	1	0	0	1	1	1	1	1	1

Смещенный код (с избытком)

Выбирается длина разрядной сетки — *n* и записываются последовательно все возможные кодовые комбинации в обычной двоичной системе счисления. Затем кодовая комбинация с единицей в старшем разряде, имеющая значение 2^{n-1}, выбирается для представления числа 0. Все последующие комбинации с единицей в старшем разряде будут представлять числа 1, 2, 3,... соответственно, а предыдущие комбинации в обратном порядке — числа -1,-2, -3,... .

Номер кодовой комбинации	Код с избытком 4	Десятичн. значение
7	111	3
6	110	2
5	101	1
4	100	0
3	O11	-1
2	010	-2
1	001	-3
0	000	-4

Смещенный код

Различия

между двоичным кодом с избытком и двоичным дополнительным кодом состоит в <u>противоположности значений знаковых битов</u>,

разность значений кодовых комбинаций в обычном двоичном коде и двоичном коде с избытком для 3-разрядных сеток равна 4 (для 4-х разрядных – 8).

Пример: кодовые комбинации 111 и 001 в обычном двоичном коде имеют значения 7 и 1, а в двоичном коде с избытком: 3 и — 3. Таким образом, разность значений кодовых комбинаций в обычном двоичном коде и двоичном коде с избытком: 7-3 = 4 и 1—(—3) = 4. <u>*Код с избытком 4.*</u>

Для *n*-разрядной сетки код будет называться двоичным кодом с избытком 2^{n-1}.

Операции над целыми числами

- Сложение. Особенность:

  0111

  1011

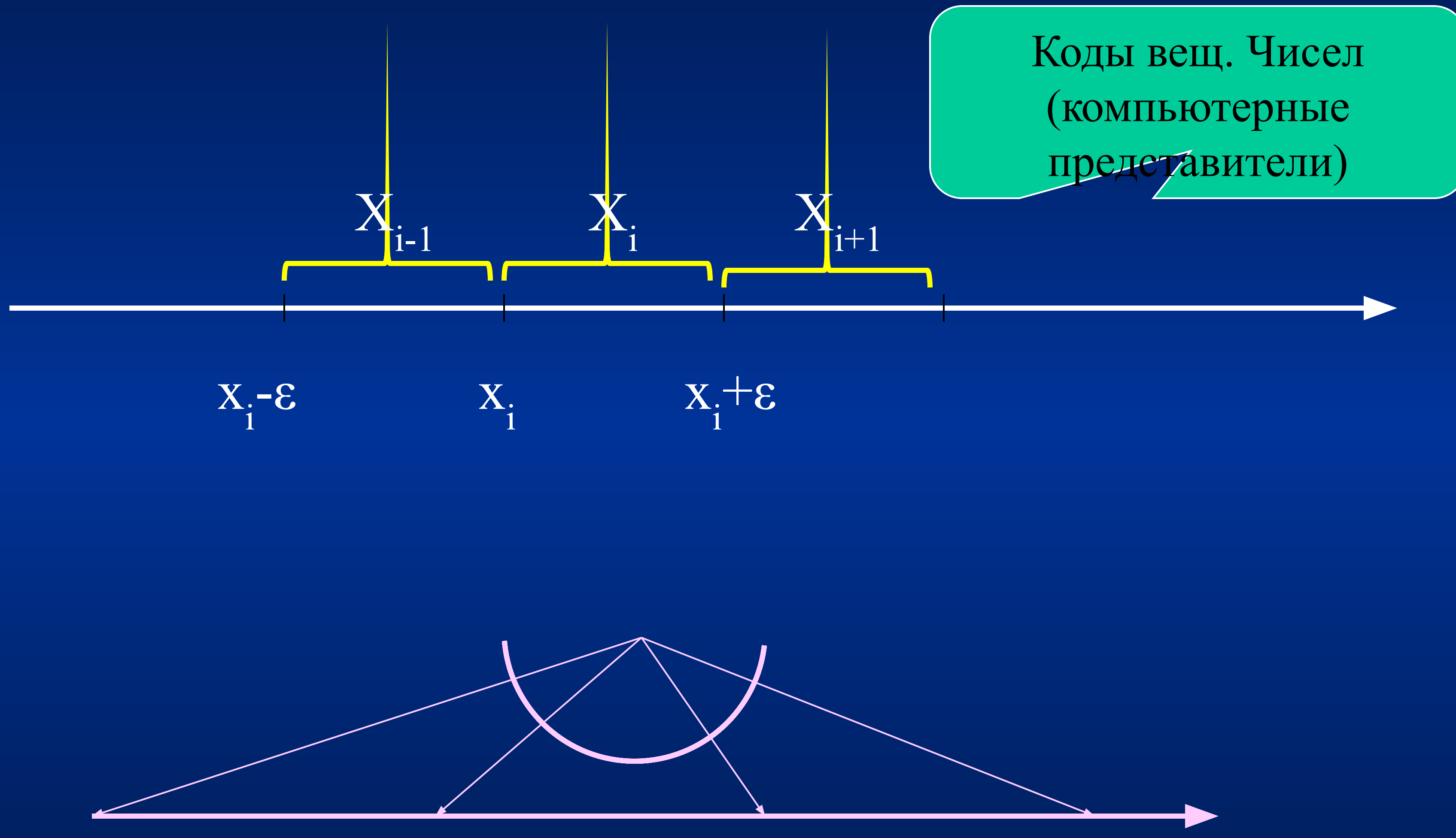
  ———

  [1]0010

  отбрасывается

- Вычитание – сводится к сложению с дополнительным кодом
- Умножение
- Целочисленное деление и нахождение остатка от деления

Вещественные числа



Вещественные числа

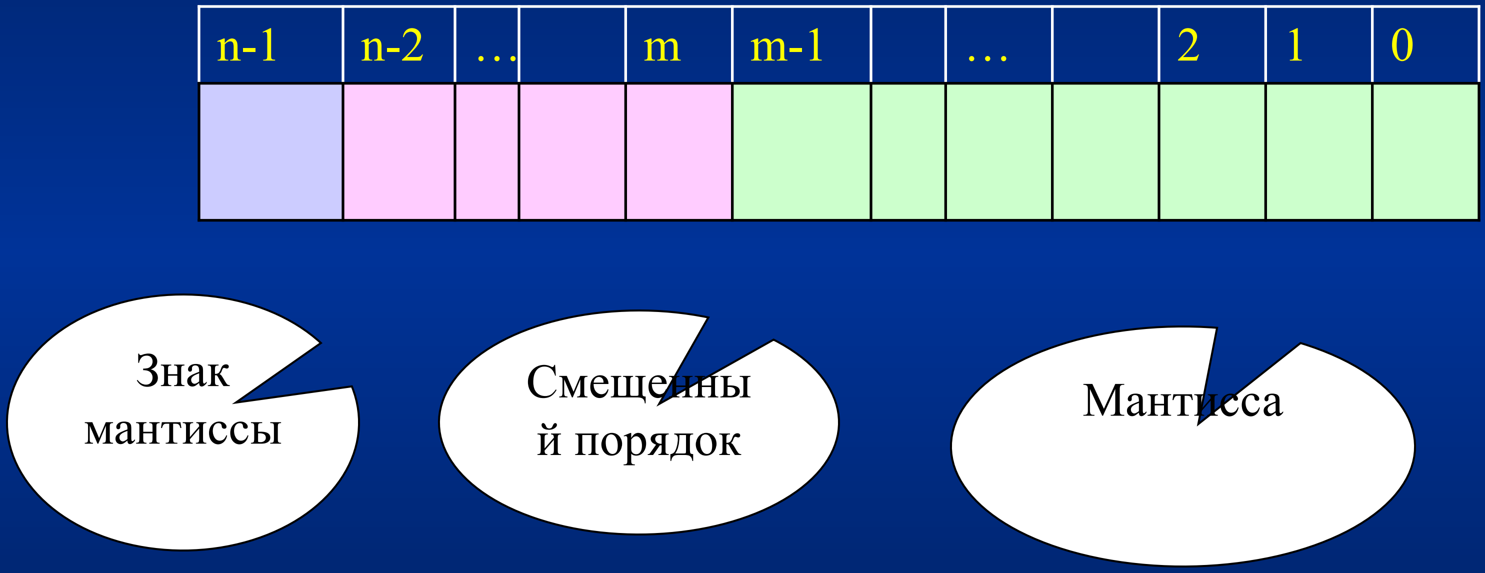
Особенности

Строгие отношения между вещественными числами превращаются в нестрогие для их компьютерных представителей

Результаты вычислений будут заведомо содержать погрешности

«Машинный нуль» и «машинная бесконечность»

Формат представления вещественных чисел



Вещественные числа в компьютерах представляются в нормализованном виде, как правило, в трех форматах – одинарном (32), двойном (64) и расширенном (80 разрядов).

Формат представления вещественных чисел

Нормализованное число одинарной точности, представленное в формате с плавающей запятой, записывается в память следующим образом:

15	14	…	…	7	6	…	0	15	…	…	…	0
31	30			23	22							0

Знак

Порядок

Мантисса (на самом деле у мантиссы 24 разряда !!!)

Формат представления вещественных чисел

Пример. $-49{,}5_{10}=-110001{,}100_2=-1{,}100011_2*10^{(5)10}$

нормализованное число

Порядок числа выражаем двоичным смещенным кодом: $5_{10}=(5+127)_{10}=(101+1111111)_2=10000100_2$.

1	1	0	0	0	0	1	0	0	1	0	0	0	1	1	0	0	0	0	…	0	0	0	0
31	30						24	23	22										…	3	2	1	0
знак	Смещенный порядок								Мантисса														

Арифметические операции с вещественными числами

1. Сложение .

$$X_1 = M_1 \cdot 10^{k_1} \qquad X_2 = M_2 \cdot 10^{k_2}$$

a) $\Delta k = |k_1 - k_2|$

b) если $k1 > k2$, то $M = M_1 + M_2 10^{-\Delta k}$ $k = k_1$

иначе $M = M_2 + M_1 10^{-\Delta k}$ $k = k_2$

c) если $10^{-1} <= M < 1$, то вывод результата в виде $M \times 10^k$, иначе предварительная нормализация

Арифметические операции с вещественными числами

2. Вычитание сводится к сложению с дополнительным кодом.
3. Умножение производится по правилу – мантиссы перемножаются, а порядки складываются. Если нужно, то полученное число нормализуется.
4. Деление производится по правилу – мантиссы делятся (делимое на делитель), а порядки вычитаются (порядок делителя из порядка делимого). Если нужно, то полученное число нормализуется

Арифметические операции с вещественными числами

Пример.

X1=0.87654 * 10^1, X2=0.94567*10^2. Пусть под запись мантиссы отводится 5 разрядов.

a. Δk=1, $k1<k2$ следовательно $k=k_2$=2 (уравняли порядки)
b. мантиссу числа X1 сдвигаем на один разряд влево (пропадет 4)
c. новая мантисса равна 0,94567+0,08765=1,03332

мантисса вышла за допустимый интервал (она >1). Нормализуя, получим мантиссу 0,10333 (теряем 2) и порядок увеличиваем на 1. В итоге получаем X=0,10333*10^3, а точный результат равен 103,3324.

Двоично-десятичные кодированные числа

Двоично-десятичные кодированные числа могут быть представлены в ПК полями переменной длины в так называемых

- упакованном и
- распакованном форматах.

Структура поля двоично-десятичного упакованного формата:

В *упакованном формате* для каждой десятичной цифры отводится по 4 двоичных разряда (полбайта), при этом знак числа кодируется в крайнем правом полубайте числа (1100 — знак "+" и 1101 — знак "-").

Структура поля двоично-десятичного упакованного формата:

Цф	Цф	...	Цф	Знак

Здесь и далее: Цф — цифра. Знак — знак числа

Структура поля распакованного формата:

В распакованном формате для каждой десятичной цифры отводится по целому байту, при этом старшие полубайты (зона) каждого байта (кроме самого младшего) в ПК заполняются кодом 0011 (в соответствии с ASCII-кодом), а в младших (левых) полубайтах обычным образом кодируются десятичные цифры. Старший полубайт (зона) самого младшего (правого) байта используется для кодирования знака числа.

Структура поля распакованного формата:

Зона	Цф	Зона	Цф	...	Зона	Цф	Знак	Цф

Распакованный формат используется в ПК при вводе-выводе информации в ПК, а также при выполнении операций умножения и деления двоично-десятичных чисел.

Пример

Число -193(10) = -000110010011 (2-ю) в ПК будет представлено:

в упакованном формате —

0001	1001	0011	1101
1	9	3	-

в распакованном формате —

0011	0001	0011	1001	1101	0011
зона	1	Зона	9	Знак “-”	3

Операция сложения над двоично-десятичными числами

- Суммирование двоично–десятичных чисел можно производить по правилам обычной двоичной арифметики, а затем производить двоично-десятичную коррекцию. Двоично-десятичная коррекция заключается в проверке каждой тетрады на допустимые коды. Если в какой либо тетраде обнаруживается запрещенная комбинация , то это говорит о переполнении. В этом случае необходимо произвести двоично-десятичную коррекцию. Двоично-десятичная коррекция заключается в дополнительном суммировании числа шесть (число запрещенных комбинаций) с тетрадой, в которой произошло переполнение или произошел перенос в старшую тетраду.
- Рассмотрим два примера:

$$+\begin{array}{r}0001\ 1000\\ 0001\ 0011\\ \hline 0010\ 1011\end{array} \Rightarrow +\begin{array}{r}0010\ 1011\\ 0000\ 0110\\ \hline 0011\ 0001\end{array} \qquad +\begin{array}{r}0001\ 1001\\ 0001\ 1001\\ \hline 0011\ 0010\end{array} \Rightarrow +\begin{array}{r}0011\ 0010\\ 0000\ 0110\\ \hline 0011\ 1000\end{array}$$