Представление чисел в ЭВМ

Способы представления чисел

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

Целые числа без знака

Целые числа без знака обычно **занимают в памяти** один, два или четыре байта и принимают:

в однобайтовом формате значения от 00000000_2 до 11111111_2 ,

в двубайтовом формате — от $00000000\ 00000000_2$ до $11111111\ 11111111_2$,

и т.д.

Диапазоны значений целых чисел без знака

Формат числа в байтах	Диапазон	
	Запись с кол-вом разрядов	Обычная запись
1	$0 \dots 2^8-1$	0 ... 255
2	$0 \dots 2^{16}-1$	0 ... 65535
4	$0 \dots 2^{32}-1$	0... 4294967295

Целые числа со знаком

<u>Целые числа со знаком</u> обычно 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**.

Диапазоны значений целых чисел без знака

Формат числа в байтах	Диапазон	
	Запись с кол-вом разрядов	Обычная запись
1	$-2^7 \dots 2^7-1$	-128 ... 127
2	$-2^{15} \dots 2^{15}-1$	-32768 ... 32767
4	$-2^{31} \dots 2^{31}-1$	-2147483648 ... 2147483647

Пример (числа без знака)

$72_{10}=1001000_2$

Номера разрядов

а) однобайтовый формат

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

Биты числа

б) двубайтовый формат

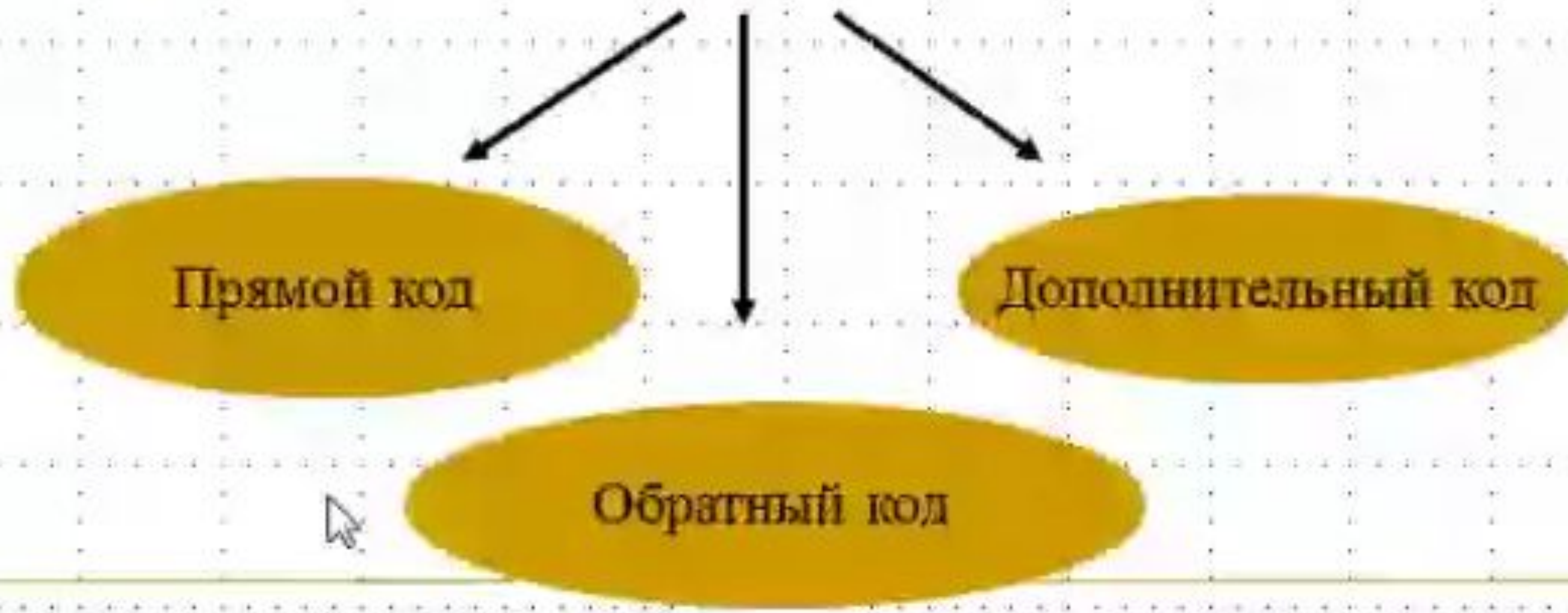
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0

в) число 65535 в двубайтовом формате

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

Все операции в ЭВМ выполняются над числами, представленными специальными машинными кодами. Их использование позволяет обрабатывать знаковые разряды чисел так же, как и значащие разряды, а также заменять операцию вычитания операцией сложения.

для представления целых чисел со знаком



Прямой код

В знаковый разряд помещается цифра знака, а в разряды цифровой части числа — двоичный код его абсолютной величины.

Прямой код числа: 1

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

Прямой код числа: -127

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

знак модуль числа

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

n-разрядность кода (машинного слова), $a_{зн}$ - значение знакового разряда.

<u>Пример:</u> если разрядность кода равна 4, то

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

Прямой код

Сложение в прямом коде чисел, имеющих одинаковые знаки, достаточно просто:

- числа складываются, их сумме присваивается знак слагаемых.

Значительно более сложным является алгебраическое сложение в прямом коде чисел с разными знаками.

- В этом случае приходится определять большее по модулю число, производить вычитание модулей и присваивать разности знак большего по модулю числа. Такую операцию значительно проще выполнять, используя *обратный* и *дополнительный* коды.

В прямом коде число 0 имеет два представления «+0» и «–0».

Обратный код

Обратный код положительных чисел совпадает с их прямым кодом. Обратный код отрицательного числа получается инвертированием всех цифр двоичного кода абсолютной величины (модуля) числа, включая разряд знака: нули заменяются единицами, а единицы — нулями.

Пример:

число: -1, модуль 0 0000001, обратный код 1 1111110

число: -127, модуль 0 1111111 , обратный код 1 0000000

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

n-разрядность кода, $a_{зн}=0$ для положительных чисел, $a_{зн}=1$ для отрицательных чисел.

$$1010 = 1*(-2^3+1)+[0x2^0+1x2^1+0x2^2] = -7+2=-5$$

В обратном коде число 0 также имеет два представления «+0» и «–0».

Дополнительный код

Дополнительный код положительных чисел совпадает с их прямым кодом. Для отрицательных получается из обратного кода прибавлением единицы к его младшему разряду. Это дает единое представление числа 0 в дополнительном коде.

-1 = обрат код 1 1111110
+1

Дополнительный код числа: -1

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

-127 = обрат код 1 0000000
+1

Дополнительный код числа: -127

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

Обычно *отрицательные* целые (десятичные) числа при вводе в машину *автоматически* преобразуются в *обратный* или *дополнительный* двоичный код и в таком виде хранятся, перемещаются и участвуют в операциях. При выводе таких чисел из машины происходит *обратное преобразование* в отрицательные числа.

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 кода, $a_{старший}=0$ для положительных чисел. $a_{старший}=1$ для отрицательных чисел.

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

положительное число + дополнительный код отрицательного числа =0

Операции над целыми числами

- Сложение.
- Вычитание. В большинстве случаев операция вычитания не используется, вместо нее производится сложение обратных или дополнительных кодов уменьшаемого и вычитаемого.
- Умножение
- Целочисленное деление и нахождение остатка от деления

Сложение обратных кодов

При сложении может возникнуть ситуация, когда старшие разряды результата операции не помещаются в отведенной для него области памяти.

Такая ситуация называется **переполнением разрядной сетки** формата числа.

Случай переполнения возможен и для обратных и для дополнительных кодов.

Сложение дополнительных кодов

А и В положительные

Десятичная запись	Двоичные коды
$$\begin{array}{r} + \begin{array}{r} 3 \\ 7 \end{array} \\ \hline 10 \end{array}$$	$$\begin{array}{r} + \begin{array}{r} 0\ 0000011 \\ 0\ 0000111 \end{array} \\ \hline 0\ 0001010 \end{array}$$

Сложение дополнительных кодов

А положительное, **В** отрицательное и по абсолютной величине больше, чем **А**

Десятичная запись	**Двоичные коды**	
3	0 0000011	
+ -10	+ 1 1110110	**Дополнительный код -10**
-7	1 1111001	**Дополнительный код -7**

Сложение дополнительных кодов

А положительное**, В** отрицательное и по абсолютной величине меньше, чем **А**

Десятичная запись	**Двоичные коды**	
10	0 0001010	
+ -3	+ 1 1111101	**Дополнительный код -3**
7	1 00000111	
0		

Перенос отбрасывается

Сложение дополнительных кодов

А и **В** отрицательные

Десятичная запись	**Двоичные коды**	
-3	1 1111101	**Дополнительный код -3**
+ -7	+ 1 1111001	**Дополнительный код -7**
-10	11 1110110	**Дополнительный код -10**

Перенос отбрасывается

Формы представления вещественных чисел

С фиксированной точкой (естественная форма)

С плавающей точкой (нормализованный вид)

Естественная форма

С фиксированной точкой все числа изображаются в виде последовательности цифр с постоянным для всех чисел положением запятой, отделяющей целую часть от дробной.

1,25; -10,44; +0,9781

Пример: Диапазон положительного вещественного числа N в системе счисления с основанием P при наличии n разрядов в целой части и s разрядов в дробной (без учета знака числа) будет:

$$P^{-s} \leq N \leq P^{n} - P^{-s}$$

При p=10, n=1 и s=2 $\quad 0,01 \leq N \leq 9,99$.

Если в результате некоторых операций получится число, выходящее за допустимый диапазон, происходит переполнение разрядной сетки и дальнейшее вычисления теряют смысл.

Для удобства отображения чисел, принимающих значения из достаточно широкого диапазона, используется форма записи чисел с **порядком основания системы счисления**:

$$\ldots = 12.5 * 10^{-1} = 1.25 * 10^{0} = 0.125 * 10^{1} = \ldots$$

Представление с плавающей точкой

Определение: Число X_{10} называется **нормализованным**, если оно представлено в виде

$$X_{10} = \pm M_{10} * 10^{\pm K},$$

где M_{10} – мантисса, $0{,}1 \le M_{10} < 1$, К-порядок, целое положительно десятичное число.

Пример: $-1234_{10} = -0{,}1234 * 10^{4}$, $0{,}003 = 0{,}3 * 10^{-2}$

При нормализации выполняется деление числа на 4 составляющих: знак числа, мантисса, знак порядка, порядок.

Для произвольной системы счисления.

$$X_p = \pm M_p * P^{\pm K}, \quad P^{-1} \le M < 1$$

Преобразование чисел из естественной формы в нормализованную

- Число больше 1.

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

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

$$N_{\leftarrow}[23{,}4*10^{6}]=0.234*10^{8}$$

- Число меньше 1.

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

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

Мантиссу и порядок *p*-ичного числа принято записывать в системе с основанием *p*, а само основание — в десятичной системе.

Десятичная система

$753.15 = 0.75315*10^3$ $-0.000034 = -0.34*10^{-4}$

$753.15 = 0.75315*E+03$

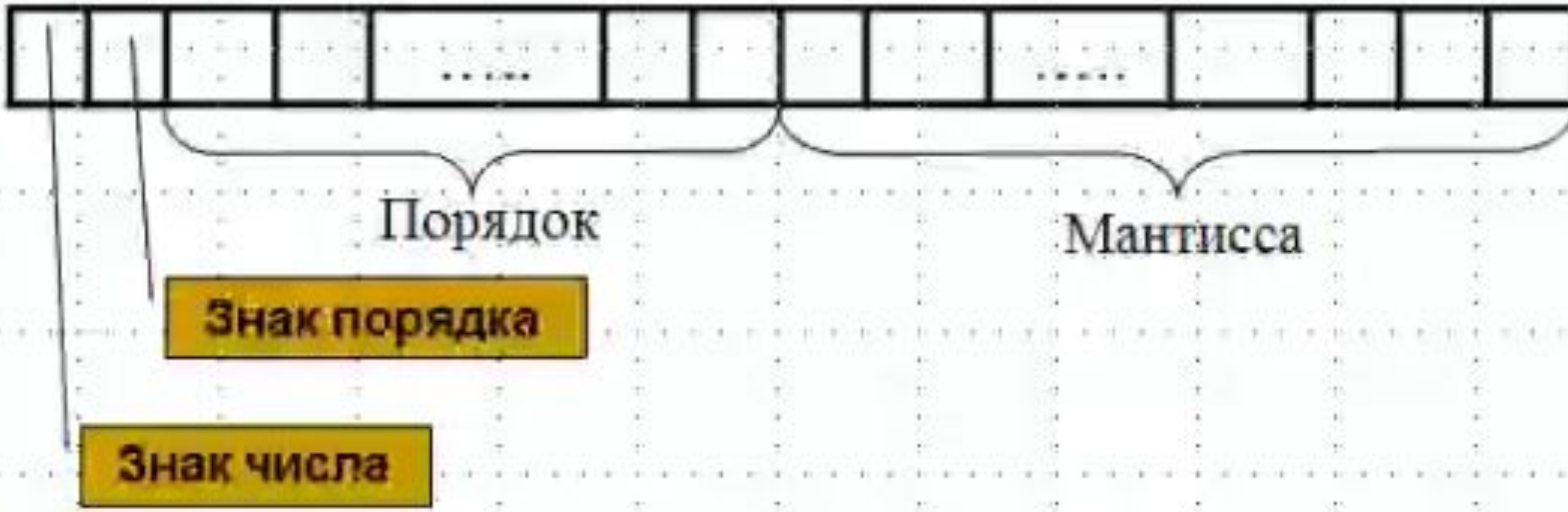
Двоичная система

$-101.01 = -0.10101*2^{11}$ (порядок $11_2 = 3_{10}$)

$-0.000011 = 0.11*2^{-100}$ (порядок $-100_2 = -4_{10}$)

Формат представления вещественных чисел

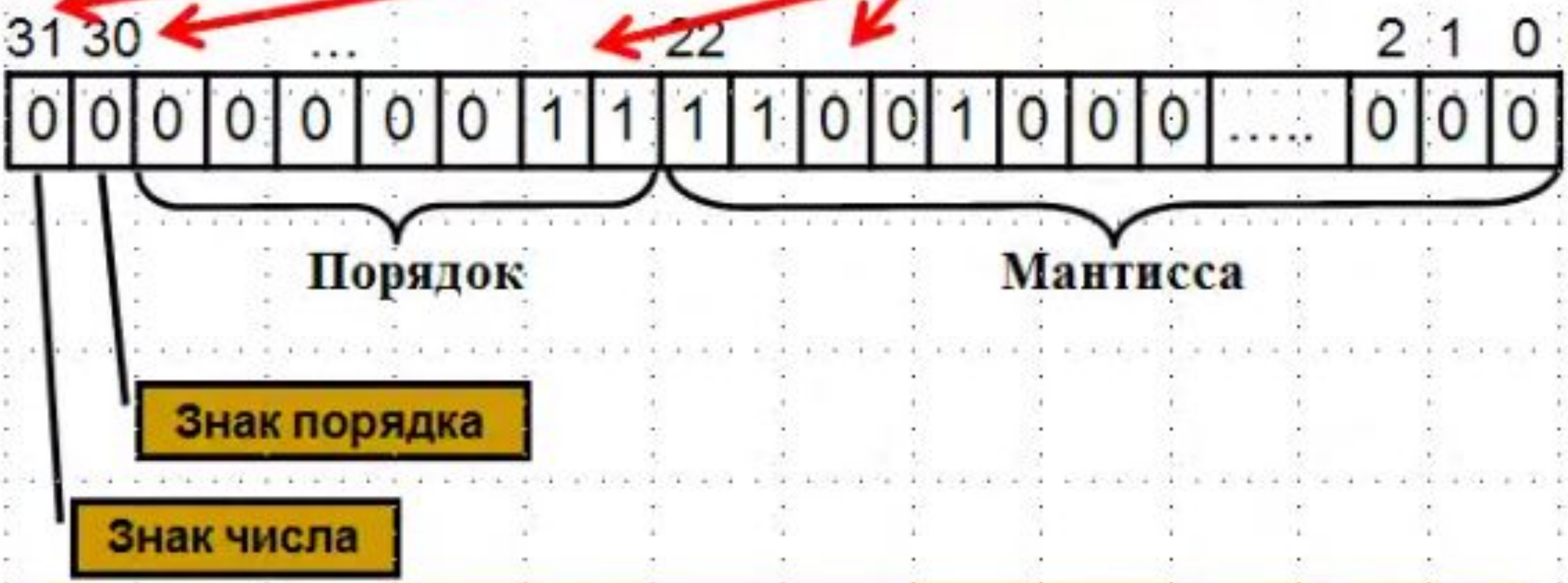
При хранении числа с плавающей точкой отводятся **разряды для мантиссы, порядка, знака числа и знака порядка:**



1. Чем больше разрядов отводится под запись мантиссы, тем выше точность представления числа.
2. Чем больше разрядов занимает порядок, тем шире диапазон от наименьшего отличного от нуля числа до наибольшего числа, представимого в машине при заданном формате.

Пример записи чисел в нормализованном виде в четырехбайтовом формате с семью разрядами для записи порядка.

Число $6.25_{10} = 110.01_2 = +0.11001 \cdot 2^{+11}$



Пример записи чисел в нормализованном виде в четырехбайтовом формате с семью разрядами для записи порядка.

2. Число $-0.125_{10} = -0.001_2 = -0.1*2^{-10}$

(отрицательный порядок записывается в дополнительном коде)

31	30								22								2	1	0
1	1	1	1	1	1	1	0	1	0	0	0	0	0	0	0	.....	0	0	0

Порядок

Мантисса

Знак порядка

Знак числа

Характеристики форматов вещественных чисел

Форматы вещественных чисел	Размер в байтах	Примерный диапазон абсолютных значений	Количество значащих десятичных цифр
Одинарный	4	$10^{-45} \dots 10^{38}$	7 или 8
Вещественный	6	$10^{-39} \dots 10^{38}$	11 или 12
Двойной	8	$10^{-324} \dots 10^{308}$	15 или 16
Расширенный	10	$10^{-4932} \dots 10^{4932}$	19 или 20

Форма представления чисел с плавающей точкой позволяет записывать числа с высокой точностью и из весьма широкого диапазона.

- Если из n разрядов, отводимых под изображение чисел, m двоичных разрядов отвести под мантиссу, k – под порядок, один разряд – под знак числа и один разряд – под знак порядка (например, 0 – плюс, 1 – минус), то диапазон представимых в форме с плавающей запятой чисел резко увеличивается ($m + k + 2 = n$):

$$-(0.111\ldots1)_2 \times (10)_2^{+(111\ldots1)_2} \le x \le +(0.111\ldots1)_2 \times (10)_2^{+(111\ldots1)_2}$$

(многоточие соответствует k единицам).

- Числа, меньшие нижней границы положительных чисел и большие верхней границы отрицательных чисел, считаются равными нулю, не различаются между собой.
- Числа, большие верхней границы положительных чисел, полагаются равными положительной бесконечности (меньшие нижней границы отрицательных – отрицательной бесконечности).
- Сравнение двух разных по величине чисел в арифметике с ограниченной разрядностью может приводить к неверному результату, как и сравнение двух равных в таких *системах* чисел с точки зрения математической.

- Такое представление очень удобно для хранения в ЭВМ, так как на самом деле необходимо хранить не само число, а его знак, мантиссу, порядок и знак порядка, и все операции с числами сводятся к операциям с этими объектами.
- Операции же с этими объектами просты: сравнение знаков, увеличение, уменьшение порядка, сложение мантисс, нормализация, то есть в конечном итоге сводятся к достаточно просто реализуемым операциям сдвига, выравнивания, сравнения разрядов.

К "неудобствам" этой формы представления чисел можно отнести возможность возникновения следующих "особо опасных" ситуаций:

- если число достаточно мало, например, $a = 0.12E + 00$, то оно может быть представлено любым числом из наименьшего интервала включающего a, в частности числом 0.120000001 или 0.199999999 и в этом случае сравнивать на равенство "в лоб" нельзя (вещественные числа в форме с плавающей запятой опасно сравнивать на совпадение);
- порядок выполнения операций может влиять на результат, например, в 4-разрядной арифметике с фиксированной запятой $20.0000 + 0.0001 = 20.0001$, но при этом

  $0.2000E+02 + 0.1000E-05 = 0.2000E+02.$

- может возникнуть так называемая ситуация "переполнения порядка" при сложении (умножении) очень больших чисел или "исчезновения порядка" при сложении (умножении) "очень малых чисел", в частности, 0.6000E + 39 ×0.1200E + 64 = 0.9999E + 99 (или же не определено), а также 0.6000E – 35 × 0.0200E – 65 = 0.9999E – 99 (или же не определено), при соответствующим образом определенной разрядности десятичной арифметики;
- при сложении чисел с плавающей запятой (а, в конечном счете, все операции выполняются через сложение) происходит выравнивание порядков для последующего сложения мантисс, а при выравнивании степеней может происходить потеря (усечение) младших разрядов, например, такая ситуация может возникнуть при сложении одного "очень большого числа" с одним "очень малым числом"