Представление вещественных чисел в памяти компьютера

Число с плавающей запятой состоит из набора отдельных двоичных разрядов, условно разделенных на так называемые **знак** (англ. *sign*), **порядок** (англ. *exponent*) и **мантиссу** (англ. *mantis*). В стандарте IEEE 754 число с плавающей запятой представляется в виде набора битов, часть из которых кодирует собой мантиссу числа, другая часть — показатель степени, и ещё один бит используется для указания знака числа (0 — если число положительное, 1 — если число отрицательное). При этом порядок записывается как целое число в коде со сдвигом, а мантисса — в нормализованном виде, своей дробной частью в двоичной системе счисления. Количество бит, выделяемых под порядок и мантиссу, определяет тип данных в котором хранится число. При этом лишь некоторые из вещественных чисел могут быть представлены в памяти компьютера точным значением, в то время как остальные числа представляются приближёнными значениями.

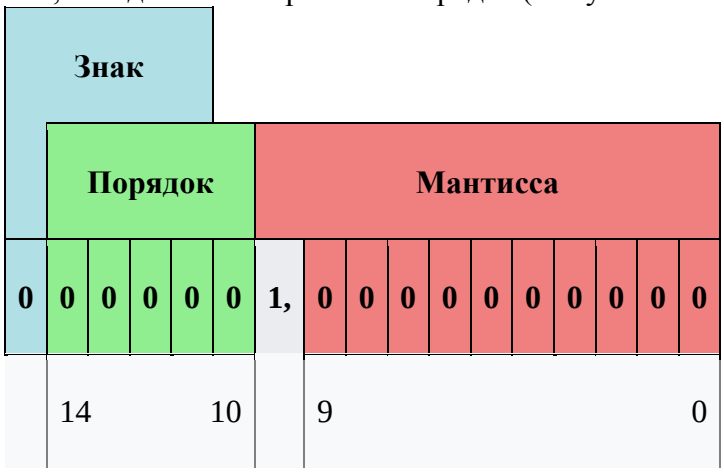
Нормализованная форма

Нормализованная (англ. *normalized*) форма – форма записи числа в которой мантисса десятичного числа принимает значения от 1(включительно) до 10 (не включительно), а мантисса двоичного числа принимает значения от 1 (включительно) до 2 (не включительно). То есть в мантиссе слева от запятой до применения порядка находится ровно один знак. В такой форме любое число (кроме 0) записывается единственным образом. Ноль же представить таким образом невозможно, поэтому стандарт предусматривает специальную последовательность битов для задания числа 0 (а заодно и некоторых других полезных чисел, таких как −∞ и +∞). Так как старший двоичный разряд (целая часть) мантиссы вещественного числа в нормализованном виде всегда равен «1», то его можно не записывать, сэкономив таким образом один бит, что и используется в стандарте IEEE 754.

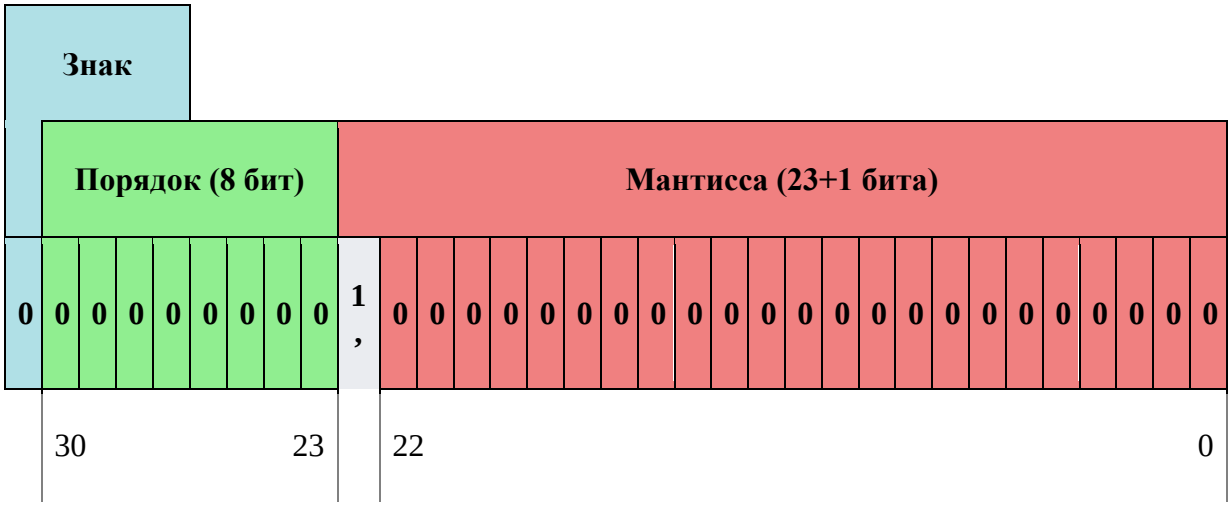
Типы чисел с плавающей запятой

Число́ половинной точности — компьютерный формат представления чисел, занимающий в памяти половину машинного слова (в случае 32-битного компьютера — 16 бит или 2 байта). В силу невысокой точности этот формат представления чисел с плавающей запятой обычно используется в видеокартах, где небольшой размер и высокая скорость работы важнее точности вычислений.

Порядок записан со сдвигом −15. То есть чтобы получить актуально значение порядка нужно вычесть из него сдвиг. Сдвиг можно получить по формуле 2^(b−1)−1, где b — число бит, отведенное на хранение порядка (в случае числа половинной точности b=5).

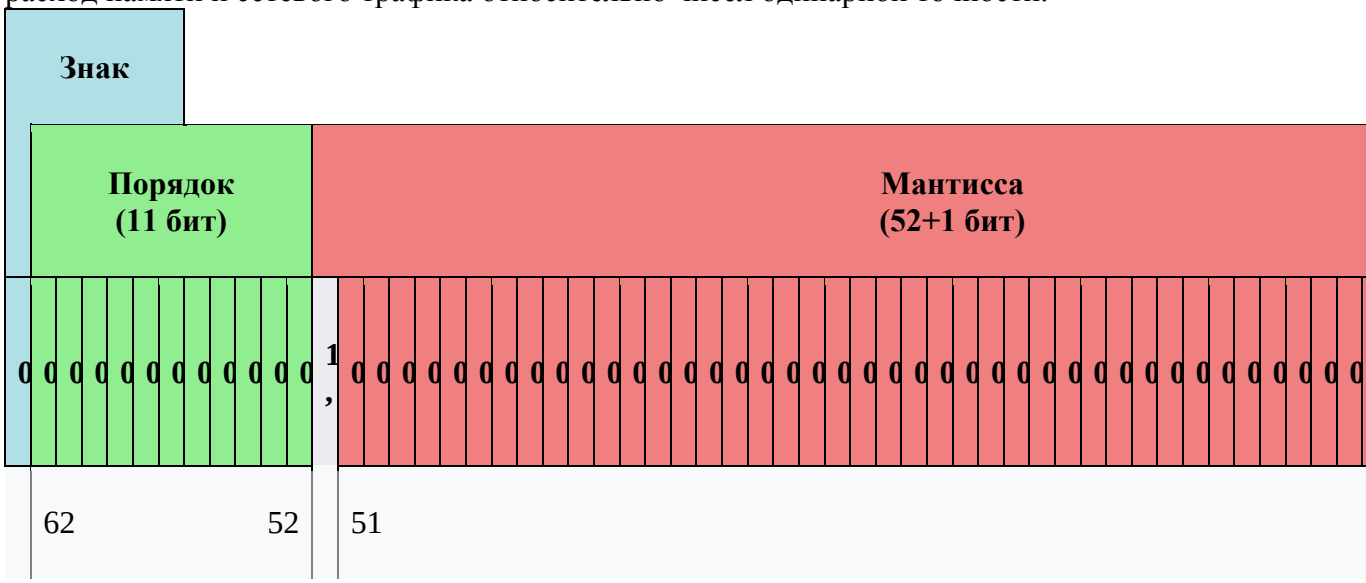


Число одинарной точности — компьютерный формат представления чисел, занимающий в памяти одно машинное слово (в случае 32-битного компьютера — 32 бита или 4 байта). Используется для работы с вещественными числами везде, где не нужна очень высокая точность.



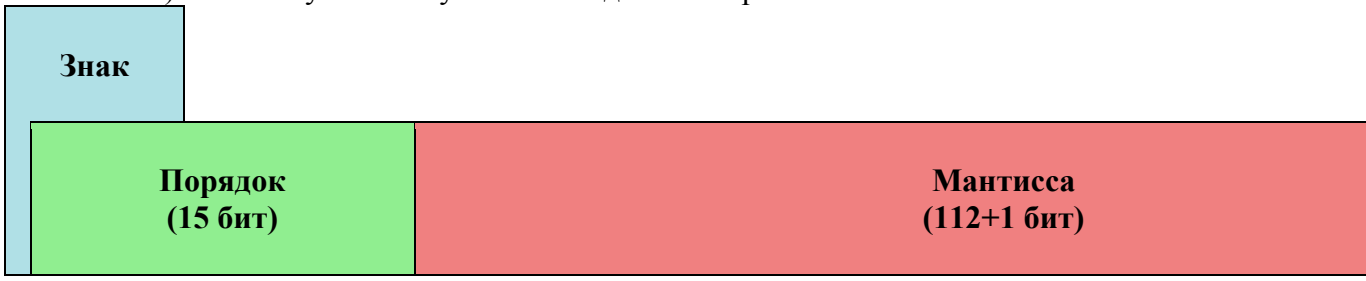
Порядок записан со сдвигом −127.

Числ́о двойно́й то́чности — компьютерный формат представления чисел, занимающий в памяти два машинных слова (в случае 32-битного компьютера — 64 бита или 8 байт). Часто используется благодаря своей неплохой точности, даже несмотря на двойной расход памяти и сетевого трафика относительно чисел одинарной точности.



Порядок записан со сдвигом −1023.

Числ́о четверно́й то́чности — компьютерный формат представления чисел, занимающий в памяти четыре машинных слова (в случае 32-битного компьютера — 128 бит или 16 байт). Используется в случае необходимости крайне высокой точности.



0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1,	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	126														112		111																																			

Мантисса (112+1 бит)																																																				
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Порядок записан со сдвигом −16383.

Обычно этот формат реализуется программно, случаи аппаратной реализации крайне редки.

Ноль (со знаком)

Как уже было оговорено выше, в нормализованной форме числа с плавающей точкой невозможно представить ноль. Поэтому для его представления зарезервированы специальные значения мантиссы и порядка — число считается нулём, если все его биты, кроме знакового, равны нулю. При этом в зависимости от значения бита знака ноль может быть, как положительным, так и отрицательным.

Знак																	
	Порядок					Мантисса											
$^0/_1$	0	0	0	0	0	1,	0	0	0	0	0	0	0	0	0	0	= ±0
	14				10		9									0	

Арифметика нуля со знаком

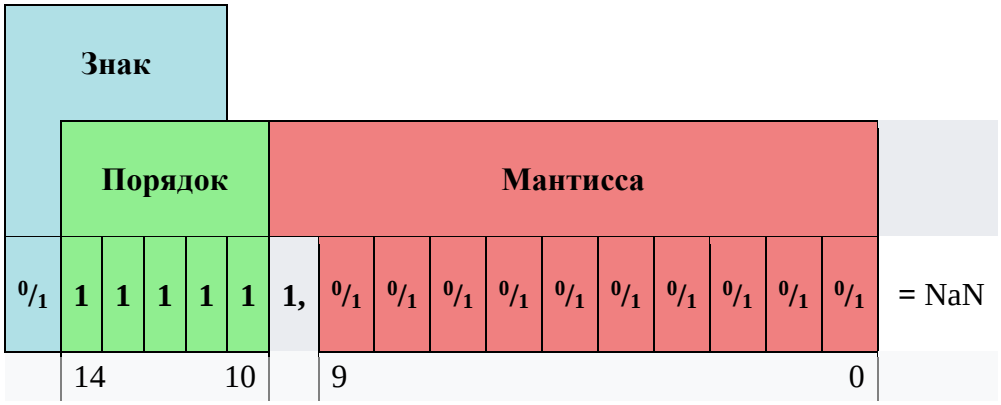
Арифметика отрицательного нуля аналогична таковой для любого отрицательного числа и понятна интуитивно. Вот несколько примеров:

- $-0/|x|=-0$ (если $x\neq 0$)
- $(-0)\cdot(-0)=+0$

- |x|·(−0)=−0
- x+(±0)=x
- (−0)+(−0)=−0
- (+0)+(+0)=+0
- −0/−∞=+0
- |x|/−0=−∞ (если x≠0)

Неопределенность (*NaN*)

NaN — это аббревиатура от фразы "*not a number*". NaN является результатом арифметических операций, если во время их выполнения произошла ошибка (примеры см. ниже). В IEEE 754 NaN представлен как число, в котором все двоичные разряды порядка — единицы, а мантисса не нулевая.



Любая операция с NaN возвращает NaN. При желании в мантиссу можно записывать информацию, которую программа сможет интерпретировать. Стандартом это не оговорено и мантисса чаще всего игнорируется.

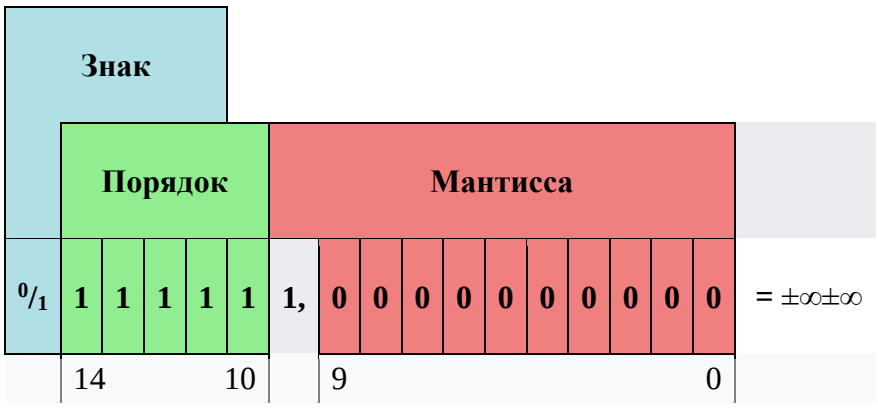
Как можно получить NaN?

- ∞+(−∞)=NaN
- 0×∞=NaN
- ±0/±0=±∞/±∞=NaN
- Sqrt(x)=NaN, где x<0

По определению NaN ≠ NaN, поэтому, для проверки значения переменной нужно просто сравнить ее с собой.

Бесконечности

В число с плавающей запятой можно записать значение +∞ или −∞. Как и нули со знаком, бесконечности позволяют получить хотя бы близки к правильному результат вычисления в случае переполнения. Согласно стандарту IEEE 754 число с плавающей запятой считается равным бесконечности, если все двоичные разряды его порядка — единицы, а мантисса равна нулю. Знак бесконечности определяется знаковым битом числа.



Получить бесконечность можно при переполнении и при делении ненулевого числа на ноль.

Алгоритм получения представления вещественного числа в памяти компьютера

1. Перевести модуль данного числа в двоичную систему счисления;
2. Нормализовать двоичное число, т.е. записать в виде $M \times 2^p$, где M мантисса (ее целая часть равна $1_{(2)}$) и p порядок, записанный в десятичной системе счисления;
3. Прибавить к порядку смещение и перевести смещенный порядок в двоичную систему счисления;
4. Учитывая знак заданного числа (0 положительное; 1 отрицательное), выписать его представление в памяти ЭВМ.

Пример. Запишем код числа −10,75. Тип данных с половинной точностью.

1. Двоичная запись модуля этого числа имеет вид 1010,11.
2. Нормализуем полученное число $1010{,}11 == 1{,}01011 \times 2^3$.
3. Получаем смещенный порядок 3 + 15 = 18. Далее имеем $18_{(10)} == 10010_{(2)}$.
4. Окончательно

1	10010	1100000000
15	14..10	9..0