

IEEE 754 Floating Point Notation

Single Precision Notation

Base 2

1 sign bit

8-bit excess 127 (not 128) notation

Bit patterns 00000000 & 11111111 are reserved for special cases

23-bit fraction

Because the base = 2, the leading digit (1 bit in this case – why?) can always be assumed to be a 1. Therefore it's not necessary to write it, and it exists as an implied bit, an integer, to the left of the fraction. This gives a 24-bit significand, 1.f, where the f's are the 23 fraction bits.

There are 5 kinds of numbers which can be represented in this notation;
Nonzero normalized numbers as described above.

A clean 0 represented by all 0's in the exponent and fraction

Infinity, which is represented by the exponent 11111111 and an all 0 fraction

Nan or Not a Number can result from undefined results, such as $\frac{0}{0}$, or

$\sqrt{-1}$ and is represented by the 11111111 exponent and any non-zero fraction.

Denormalized numbers, which don't have the leading 1 and consequently can be very small. They have the 00000000 exponent, and the fraction which contains the actual bit pattern for the number (no leading 1 is assumed).

Some examples:

Sign	Exponent	Fraction	Value
0	10000100	101000000000000000000000	$+1.101x2^5$
1	00000001	010110000000000000000000	$-1.01011x2^{-126}$
0	11111110	000000000000000000000000	$+1.0x2^{127}$
0	00000000	000000000000000000000000	+0
1	00000000	000000000000000000000000	-0
0	11111111	000000000000000000000000	$+\infty$
0	00000000	010000000000000000000000	$+2^{-128}$
0	11111111	011000000000000000000000	+NaN

Notice that the smallest value represented by the exponent bits is -126 , $1-127$ (excess-127). The all 0 exponent also carries the value -126 . Otherwise there would be no way to represent 2^{-127} .

There are also double precision and extended precision notations. But I won't discuss them here.