No padrão IEEE754, o número binário $(-1)^s \times M \times 2^E$ é armazenado da seguinte forma:

onde:

s é o bit do sinal

c é a característica do expoente, em base 2

f é o significando (bits significativos), em base 2. O bit mais significativo não é armazenado, sendo determinado pela característica do expoente.

bias é o deslocamento do expoente, que permite a representação de expoentes negativos

A relação destes elementos com o número $(-1)^s \times M \times 2^E$ é a seguinte:

c ($\|c\|$ bits)	f	M(em base 2)	$(E)_{10}$	Observações e casos especiais
$(0\ldots00)_2 = (0)_{10}$	$0\ldots0$	$0.0\ldots0$		$+0 : s = 0$
				$-0 : s = 1$
	$0\ldots01$	$0.0\ldots01$	$1 - bias$	menor número representável
	$b_1\ldots b_n$	$0.b_1\ldots b_n$	$1 - bias$	quando $f \neq 0$:
				números subnormais*
$(0\ldots01)_2 = (1)_{10}$	$0\ldots00$	$1.0\ldots00$	$1 - bias$	menor número normalizado
$(0\ldots01)_2 = (1)_{10}$	$b_1\ldots b_n$	$1.b_1\ldots b_n$	$1 - bias$	
$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$
$(1\ldots10)_2 = (2^{\|c\|} - 2)_{10}$	$b_1\ldots b_n$	$1.b_1\ldots b_n$	$(2^{\|c\|} - 2)_{10} - bias$	
$(1\ldots10)_2 = (2^{\|c\|} - 2)_{10}$	$1\ldots11$	$1.1\ldots1$	$(2^{\|c\|} - 2)_{10} - bias$	maior número representável
$(1\ldots11)_2 = (2^{\|c\|} - 1)_{10}$	$0\ldots0$			$+\infty : s = 0$
$(1\ldots11)_2 = (2^{\|c\|} - 1)_{10}$	$0\ldots0$			$-\infty : s = 1$
$(1\ldots11)_2 = (2^{\|c\|} - 1)_{10}$	$\neq 0$			NaN: “Not a Number”

*significando não é normalizado

O número de bits reservado para cada um destes elementos e o *bias* do expoente varia conforme a precisão estabelecida. As opções são:

Formato	s	c	f	Total de bits	Bias do Expoente	Precisão
Half	1	5	10	16	15	11
Single	1	8	23	32	127	24
Double	1	11	52	64	1023	53
Quad	1	15	112	128	16383	113

Table 4.1: IEEE Single Format

$\pm$	$a_1a_2a_3\ldots a_8$	$b_1b_2b_3\ldots b_{23}$

If exponent bitstring $a_1\ldots a_8$ is	Then numerical value represented is
$(00000000)_2 = (0)_{10}$	$\pm(0.b_1b_2b_3\ldots b_{23})_2 \times 2^{-126}$
$(00000001)_2 = (1)_{10}$	$\pm(1.b_1b_2b_3\ldots b_{23})_2 \times 2^{-126}$
$(00000010)_2 = (2)_{10}$	$\pm(1.b_1b_2b_3\ldots b_{23})_2 \times 2^{-125}$
$(00000011)_2 = (3)_{10}$	$\pm(1.b_1b_2b_3\ldots b_{23})_2 \times 2^{-124}$
$\downarrow$	$\downarrow$
$(01111111)_2 = (127)_{10}$	$\pm(1.b_1b_2b_3\ldots b_{23})_2 \times 2^{0}$
$(10000000)_2 = (128)_{10}$	$\pm(1.b_1b_2b_3\ldots b_{23})_2 \times 2^{1}$
$\downarrow$	$\downarrow$
$(11111100)_2 = (252)_{10}$	$\pm(1.b_1b_2b_3\ldots b_{23})_2 \times 2^{125}$
$(11111101)_2 = (253)_{10}$	$\pm(1.b_1b_2b_3\ldots b_{23})_2 \times 2^{126}$
$(11111110)_2 = (254)_{10}$	$\pm(1.b_1b_2b_3\ldots b_{23})_2 \times 2^{127}$
$(11111111)_2 = (255)_{10}$	$\pm\infty$ if $b_1 = \cdots = b_{23} = 0$, NaN otherwise

now that -0 and 0 are *two different representations for the same number zero*, while $-\infty$ and ∞ represent *two very different numbers*. Another special number is NaN, which stands for "Not a Number" and is accordingly not a number at all, but an error pattern. This too will be discussed later. All of these special numbers, as well as others called subnormal numbers, are represented through the use of a specific bit pattern in the exponent field.

The Single Format

The IEEE standard specifies two basic representation formats, *single* and *double*. *Single format* numbers use a 32-bit word and their representations are summarized in Table 4.1.

Let us discuss Table 4.1 in some detail. The $\pm$ refers to the sign of the number, a zero bit being used to represent a positive sign. The first line shows that the representation for zero requires a special zero bitstring for the exponent field *as well as* a zero bitstring for the fraction field, i.e.,

$\pm$	00000000	00000000000000000000000

No other line in the table can be used to represent the number zero, for all lines except the first and the last represent normalized numbers, with an initial bit equal to 1; this is the one that is hidden. In the case of the first line of the table, the hidden bit is 0, not 1. The 2^{-126} in the first line is confusing at first sight, but let us ignore that for the moment since $(0.000\ldots0)_2 \times 2^{-126}$ is certainly one way to write the number 0. In the case when the exponent field has a zero bitstring but the fraction field has a nonzero bitstring, the number represented is said to be *subnormal*.[9] Let us postpone the discussion of subnormal numbers for the moment and go on to the other lines of the table.

All the lines of Table 4.1 except the first and the last refer to the normalized numbers, i.e., all the floating point numbers that are not special in some way. Note

[9]The word *denormalized* was used in IEEE 754. The word *subnormal* replaced it in IEEE 854.