

3. Representação Numérica segundo Padrão IEEE 754

A norma [IEEE 754](#), publicada em 1985, procurou uniformizar a maneira como as diferentes máquinas representam os números em ponto flutuante, bem como devem operá-los.

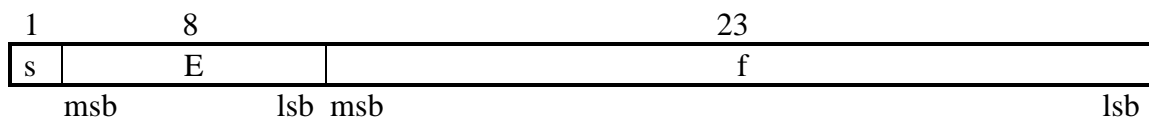
Essa norma define dois formatos básicos para os números em ponto flutuante: o formato simples, com 32 bits e o duplo com 64 bits. O primeiro bit é para o sinal: 0 representa número positivo e 1 representa número negativo. No formato simples o expoente tem 8 bits e a mantissa tem 23 bits; no formato duplo, o expoente tem 11 bits e a mantissa 52 bits.

No formato simples, o menor expoente é representado por 00000001, valendo -126, e o maior expoente é representado por 11111110, valendo +127. Em ambos os casos, o expoente vale o número representado em binário menos 127.

3.1. Precisão Simples

Padrão: 4 bytes ou 32 bits (precisão de 7 a 8 dígitos significativos equivalentes).

Neste padrão um número real v pode ser representado por:



onde: $s = 0 \Rightarrow v$ positivo e $s = 1 \Rightarrow v$ negativo

e = expoente

f = mantissa

polarização = $(127)_{10} = 2^7 - 1 = (01111111)_2$

msb = bit mais significativo e lsb = bit menos significativo

Um número v armazenado no registro acima é interpretado da seguinte forma:

- Se $0 < e < 255$, então $v = (-1)^s \cdot 2^{(e-127)} \cdot (1,f)$
- Se $e = 0$ e $f \neq 0$, então $v = (-1)^s \cdot 2^{-126} \cdot (0,f)$
- Se $e = 0$ e $f = 0$, então $v = (-1)^s \cdot 2^{-126} \cdot (0,) = (-1)^s \cdot 0$ (zero)
- Se $e = 255$, então v pertence a região de overflow.

No formato duplo, o menor expoente é representado por 00000000001, valendo -1022, e o maior expoente é representado por 11111111110, valendo +1023. Em ambos os casos, o expoente vale o número representado em binário menos 1023.

No formato simples, o zero possui, ainda, duas representações 0 00000000 000...00, correspondendo a mais zero e 1 00000000 000...00, correspondendo a menos zero, ambas iguais em qualquer operação de comparação.

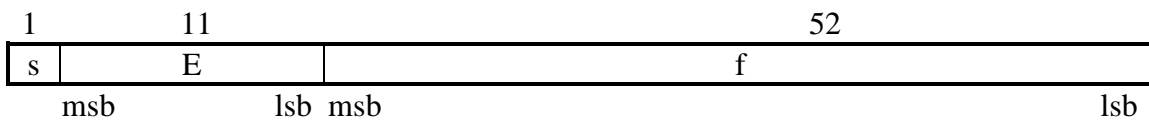
Mais infinito é representado por 0 11111111 000...00 e menos infinito por 1 11111111 000...00.

Indeterminado é representado por 1 11111111 100...00.

As demais combinações não são válidas, sendo consideradas "not a number".

3.2. PRECISÃO DUPLA

Padrão: 8 bytes ou 64 bits (precisão de 16 a 17 dígitos significativos equivalentes).



onde: polarização = $(1023)_{10} = 2^{10} - 1 = (011111111111)_2$

Um número v armazenado no registro acima é interpretado da seguinte forma:

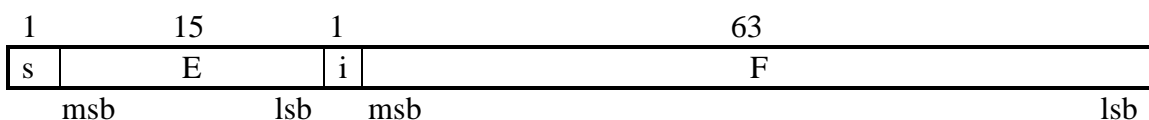
- Se $0 < e < 2047$, então $v = (-1)^s \cdot 2^{(e-1023)} \cdot (1,f)$
- Se $e = 0$ e $f \neq 0$, então $v = (-1)^s \cdot 2^{-1022} \cdot (0,f)$
- Se $e = 0$ e $f = 0$, então $v = (-1)^s \cdot 2^{-1022} \cdot (0,0) = (-1)^s \cdot 0$ (zero)
- Se $e = 2047$, então v pertence a região de overflow.

Em ambos os formatos, a norma IEEE 754 prevê o chamado underflow gradual, permitindo obter números bem mais próximos de zero. Para isso, como mostrado no exemplo hipotético, o expoente representado por 000...000 representa o menor expoente, "-126" no formato simples e "-1022" no formato duplo, e a mantissa deixa de ser normalizada. Dessa maneira podemos representar, como menor número positivo:

no formato simples: 0 00000000 00...01 isto é: $2^{-126} \times 2^{-23} = 2^{-149}$
 no formato duplo: 0 000000000000 00...01 isto é: $2^{-1022} \times 2^{-52} = 2^{-1074}$

3.3. PRECISÃO EXTENDIDA

Padrão: 10 bytes ou 80 bits (precisão de 19 a 20 dígitos significativos equivalentes).



onde

polarização = $(16383)_{10} = 2^{14} - 1 = (0111111111111111)_2$

Um número v armazenado no registro acima é interpretado da seguinte forma:

- Se $0 < e < 32767$, então $v = (-1)^s \cdot 2^{(e-16383)} \cdot (i,f)$ (onde i pode assumir 0 ou 1)
(se $e = 0 \Rightarrow i = 1$)
- Se $e = 32767$ e $f = 0$, então v pertence a região de overflow.

4. ERRO ABSOLUTO E ERRO RELATIVO

Definimos como erro absoluto a diferença entre o valor exato de um número X e de seu valor aproximado X' .

$$EA_x = X - X'$$

Em geral, apenas o valor X' é conhecido e, neste caso, é impossível obter o valor exato do erro absoluto. O que se faz é obter um limitante superior ou uma estimativa para o módulo do erro absoluto.

Por exemplo, sabendo-se que $\pi \in (3.14, 3.15)$ tomaremos para π um valor dentro deste intervalo e teremos, então, $|EA_\pi| = |\pi - \pi'| < 0.01$.

Seja agora o número X representado por $X' = 2112.9$ de tal forma que $|EA_x| < 0.1$, ou seja, X (2112.8, 2113) e seja o número Y representado por $Y' = 5.3$ de tal forma que $|EA_y| < 0.1$, ou seja, Y (5.2, 5.4). Os limitantes superiores para os erros absolutos são os mesmos. Podemos dizer que ambos os números estão representados com a mesma precisão?

É preciso comparar a ordem de grandeza de X e Y . Feito isto, é fácil concluir que o primeiro resultado é mais preciso que o segundo, pois a ordem de grandeza de X é maior que a ordem de grandeza de Y . Então, dependendo da ordem de grandeza dos números envolvidos, o erro relativo é amplamente empregado.

O erro relativo é definido como o erro absoluto dividido pelo valor aproximado:

$$ER_x = \frac{EA_x}{X'} = \frac{X - X'}{X'}$$

No exemplo anterior, temos

$$|ER_x| = \frac{|EA_x|}{|X'|} < \frac{0.1}{2112.9} = 4,7 \cdot 10^{-5}$$

e

$$|ER_y| = \frac{|EA_y|}{|Y'|} < \frac{0.1}{5.3} = 0.02$$

confirmando, portanto, que o número x é representado com maior precisão que o número y .