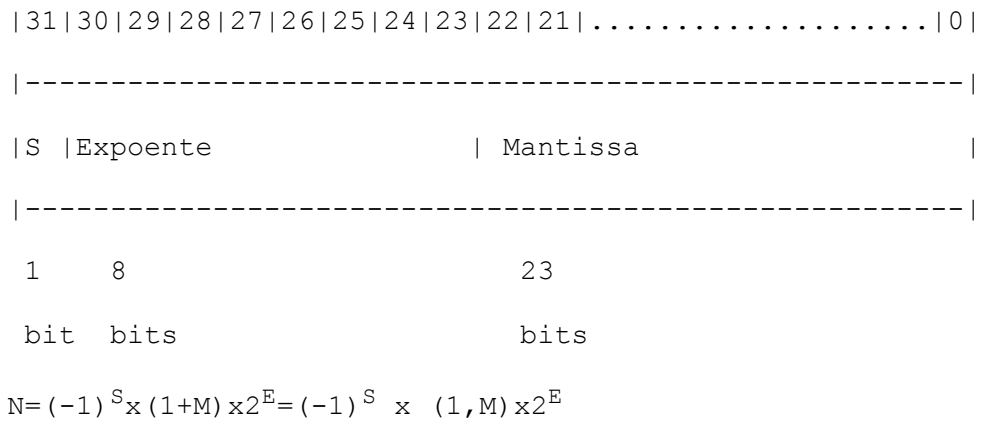
Representação em ponto flutuante

Formato IEEE 754 para ponto flutuante



$N=(-1)^S x(1+M)x2^E=(-1)^S \ x \ (1,M)x2^E$

Precisão simples: 8 bits para expoente e 23 para mantissa
Precisão dupla: 11 bits para expoente e 52 bits para mantissa
Representação em ponto flutuante para expoente, temos 8 bits, o excesso pode ser 2^7 ou 2^7-1. Escolheram o segundo que é 127 para precisão simples e 1023 para precisão dupla.
Exemplo de precisão dupla:
$-0,75=-0,11_2=-1,1_2x\ 2^{-1}$

$N=-1,1_2\ x\ 2^{-1}$

$N=(-1)^S$ x (1+M)x $2^{(E-1023)}=$
$(-1)^1$ x (1+0,10000000000000000000)x $2^{(1022-1023)}$
10111111111010000000000000000000000000000000000000000000000000

C0A00000C0A00000 representa que número em ponto flutuante ?
1100 0000 1010 0000 0000 0000 0000 0000 1100 0000 1010 0000 0000 0000 0000 0000
Sinal = -1
Expoente=E-1023=1034-1023=+11
Mantisssa=00000000000000000000110000001010000000000000000000
$N=-1,00000000000000000000110000001010_2x2^{+11}$

- Casos especiais:

  - E=0(e=-127) e M=0, S=0 ou 1, representa zero
  - E=0(e=-127) e M#0, S=0 ou 1, Não normalizado, e=-126, complicado de usar
  - E=255(e=-128) e M=0, S=0 ou 1, (+/-)infinito, divisão por zero
  - E=255 (e=-128), M#0, S=0 ou 1, NaN, 0/0,infinito -infinito
  - -126 =< e <=+127, número normalizado

Menor não-normalizado, $0,00000000000000000000001_2x2^{-126}=1,0_2x2^{-149}$

Maior não-normalizado, $0,11111111111111111111111_2x2^{-126}$

Menor número normalizado, 1 para expoente e 0 para mantissa, $1,0_2x2^{-126} = 10^{-38}$

Maior número normalizado, bits do expoente todos em 1, com exceção do bit menos significativo, logo e= 254-127=+127, e mantissa toda com 1s, $1,1111..1_2x2^{127}=10^{38}$

Valores para o padrão IEEE 754

Sinal	Expoente	Mantissa	Valor do número
0	0000....0000	0000....0000	+0
0	0000....0000	0000....0001 1111....1111	0,M x 2^{-126}
0	0000....0001 1111....1110	XXXX....XXXX	1,XXXX....XXXX x $2^{(e-b)}$
0	1111....1111	0000....0000	+ infinito
0	1111....1111	0000....0001 1111....1111	NaN
1	0000....0000	0000....0000	-0
1	0000....0000	0000....0001 1111....1111	-0,M x 2^{-126}
1	0000....0001 1111....1110	XXXX....XXXX	-1,XXXX....XXXX x $2^{(e-b)}$
1	1111....1111	0000....0000	- infinito
1	1111....1111	0000....0001 1111....1111	NaN

Soma em ponto flutuante

Somar $9{,}999 \times 10^{1}$ com $1{,}610 \times 10^{-1}$.
Suponha que pode-se utilizar 4 dígitos para o número e 2 dígitos para expoente.

1. Alinha o ponto decimal do número que tem o menor expoente para igualar ao expoente do número que tem maior expoente.
   $1{,}610 \times 10^{-1} = 0{,}1610 \times 10^{0} = 0{,}016 \times 10^{1}$ (só pode usar 4 dígitos)

   Porque alinha menor com maior ? Pode perder digitos, então melhor perder dígitos menos significativos. Maior chance de resultado ser normalizado.

2.

3. 1.101×2^{-6},

4. 1.001001×2^{6}

5. $.000000000001101 \times 2^{6}$

6. 1.001001×2^{6}

7. ----------

8. $1.001001000001101 \times 2^{6}$

9. Soma as mantissas

10.

11. 9,999

12. 0,016

13. --------

14. $10,015x10^1$

15. Coloca em notação científica normalizada
    $10,015x10^1=1,0015x10^2$
16. Ajusta para número de dígitos possível com arredondamento
    $1,0015x10^2=1,002x10^2$
17. Verifica se houve overflow ou underflow (expoente maior ou menor do que é possível ser representado)

Exemplo de adição de números em ponto flutuante em binário:

Somar $0,5_{10}$ com $-0,4375_{10}$

1. Converte para a base 2.
   $0,5=0,1x2^0=1,000x2^{-1}$
   $-0,4375=-0,01110x2^0=-1,110x2^{-2}$
2. Desloca para a direita o número com menor expoente. O número de deslocamentos é igual a diferença entre os expoentes. Se os expoentes são iguais, não há deslocamento.
   $-1,110x2^{-2}=-0,1110x2^{-1}=-0,111x2^{-1}$
3. Soma os números
   $1,000x2^{-1}+(-0,111x2^{-1})=0,001x2^{-1}$
4. Normaliza a soma se necessário
   (ex. resultado igual a 10.__ ou 11.__ deverá sofrer um deslocamento para a direita e um incremento no expoente. Podemos tambem ter um resultado de 0.____ se somarmos um número positivo com negativo. Deverá sofrer um deslocamento para a esquerda e um decremento do expoente.
   $0,001x2^{-1}=0,010x2^{-2}=0.100x2^{-3}=1,000x2^{-4}$
5. Para padrão IEEE 754 precisão simples, ocorre overflow se expoente maior que 127 ou underflow se expoente menor que -126 então será gerada uma exceção
6. Trunca resultado para número de bits que pode ser utilizado.
   $1,000x2^{-4}$.