Parte 2

Representação de Números Reais

CI1164 - Introdução à Computação Científica

Profs. Armando Delgado e Guilherme Derenievicz
Departamento de Informática - UFPR

Sistemas de Numeração

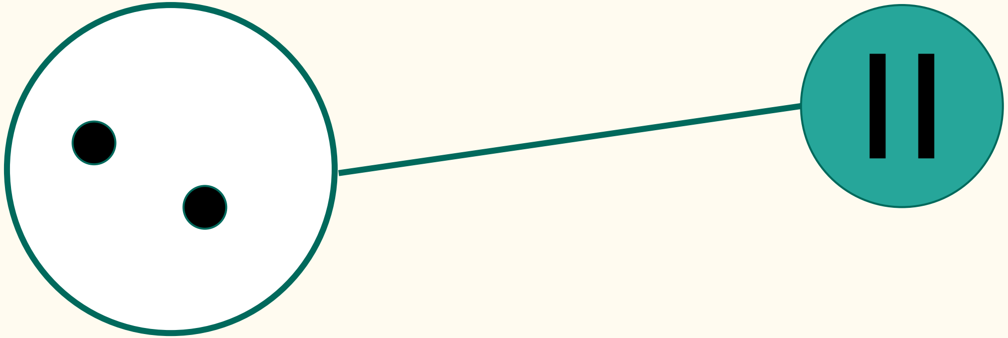
Formas de representar quantidades ou números

Sistemas de Numeração

Formas de representar quantidades ou números

Sistemas de Numeração

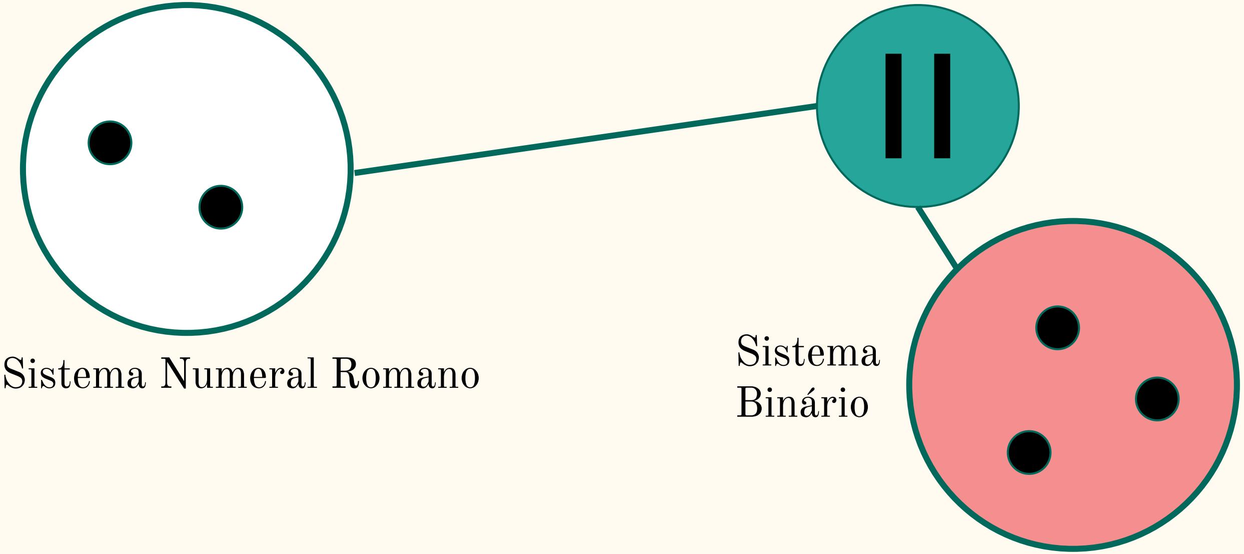
Formas de representar quantidades ou números



Sistema Numeral Romano

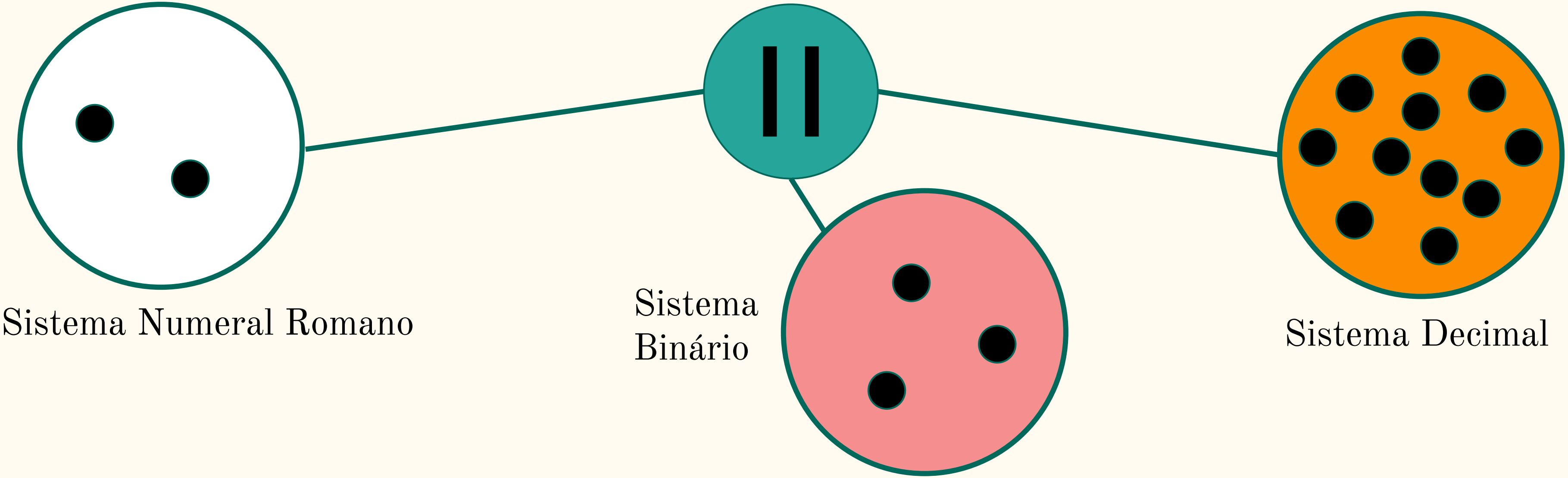
Sistemas de Numeração

Formas de representar quantidades ou números

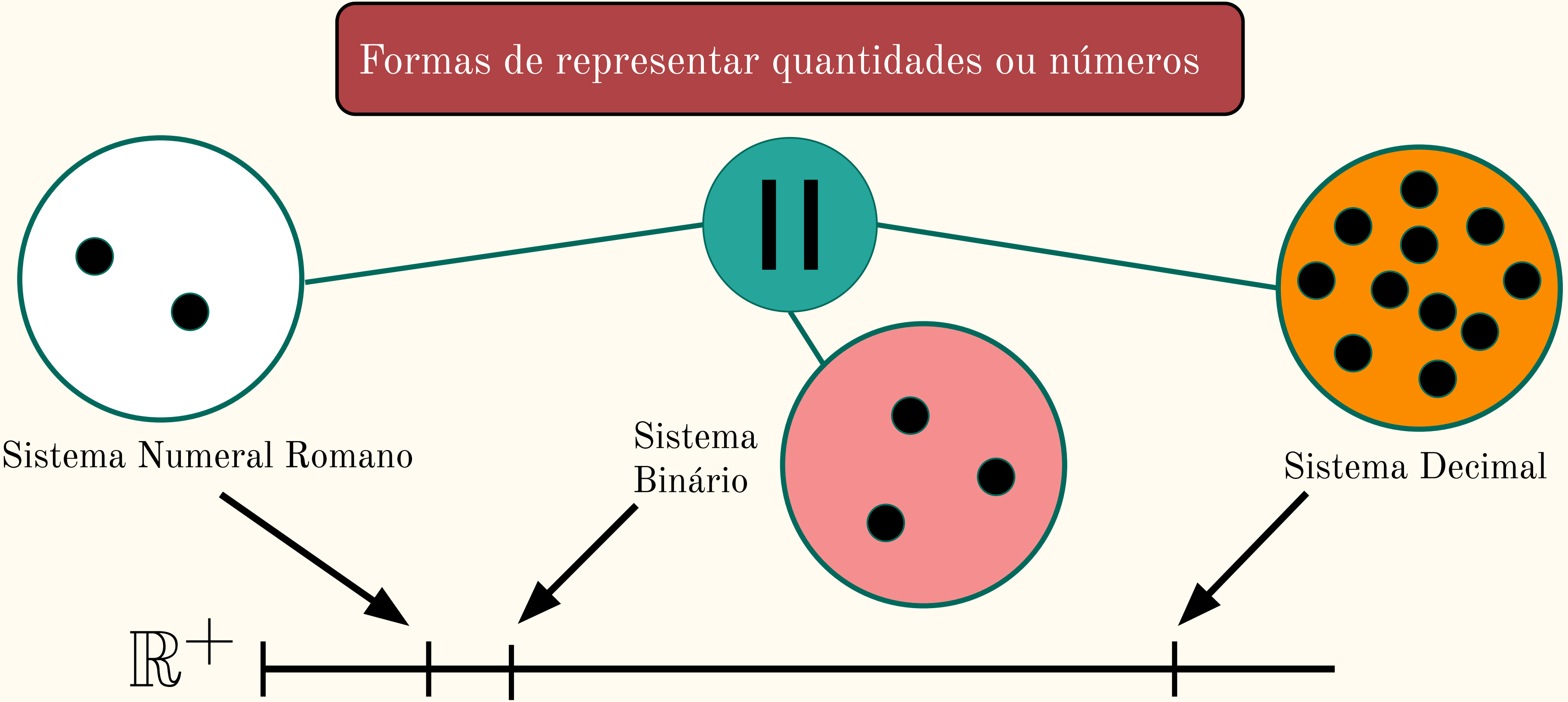


Sistemas de Numeração

Formas de representar quantidades ou números

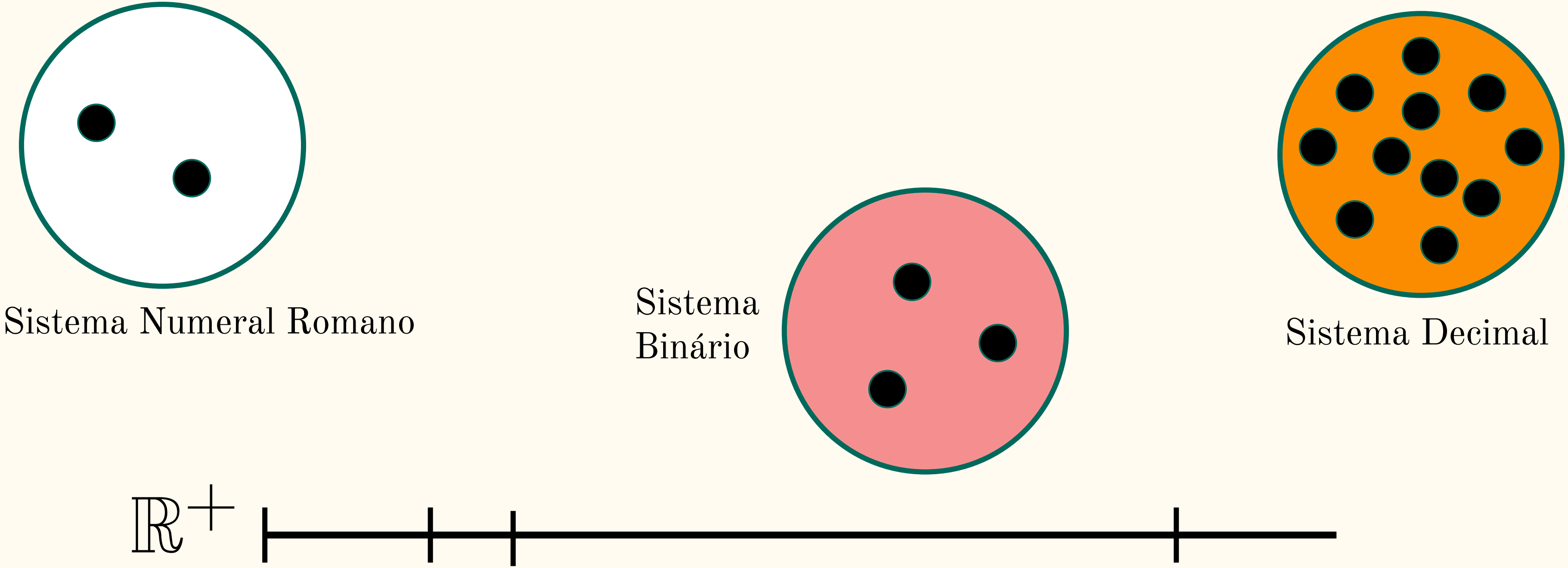


Sistemas de Numeração



Sistemas de Numeração

Base: quantidade β de símbolos que podem ser utilizados para representar um número em dado sistema de numeração.



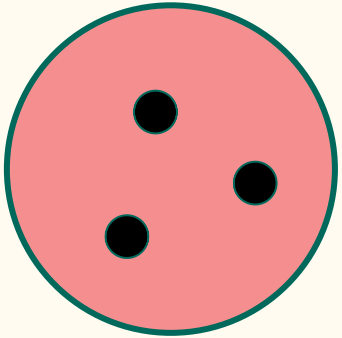
Sistemas de Numeração

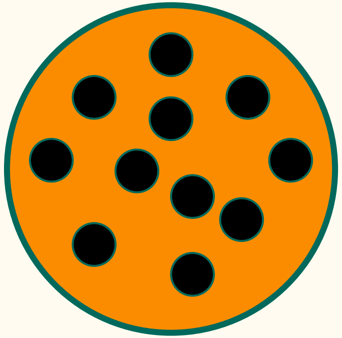
Base: quantidade β de símbolos que podem ser utilizados para representar um número em dado sistema de numeração.

$\beta = 7$
$\{I, V, X, L, C, D, M\}$
Ex: VI

Sistema Numeral Romano

Sistema Binário





Sistema Decimal

Sistemas de Numeração

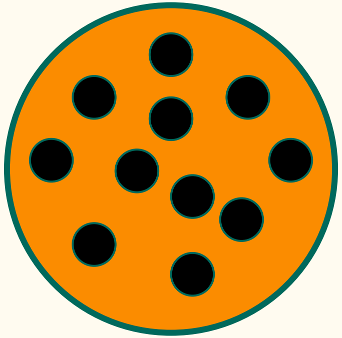
<u>Base</u>: quantidade β de símbolos que podem ser utilizados para representar um número em dado sistema de numeração.

$\beta = 7$
$\{I, V, X, L, C, D, M\}$
Ex: VI

Sistema Numeral Romano

Sistema Binário

$\beta = 2$
$\{0, 1\}$
Ex: 110



Sistema Decimal

$\mathbb{R}^+$

Sistemas de Numeração

<u>Base</u>: quantidade β de símbolos que podem ser utilizados para representar um número em dado sistema de numeração.

$\beta = 7$
$\{I, V, X, L, C, D, M\}$
Ex: VI

Sistema Numeral Romano

Sistema Binário

$\beta = 2$
$\{0, 1\}$
Ex: 110

$\beta = 10$
$\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$
Ex: 6

Sistema Decimal

$\mathbb{R}^+$

Sistemas de Numeração

Base: quantidade β de símbolos que podem ser utilizados para representar um número em dado sistema de numeração.

$\beta = 7$
$\{I, V, X, L, C, D, M\}$
Ex: VI

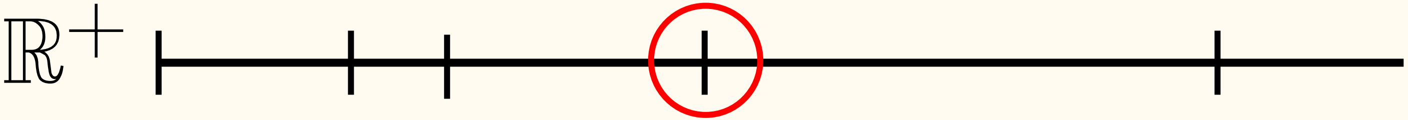
Sistema Numeral Romano

Sistema Binário

$\beta = 2$
$\{0, 1\}$
Ex: 110

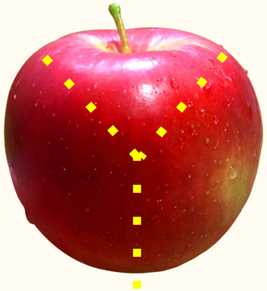
$\beta = 10$
$\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$
Ex: 6

Sistema Decimal



Sistemas de Numeração

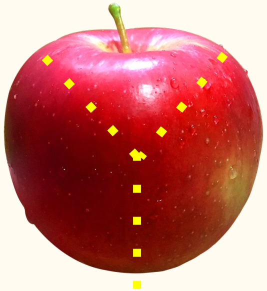
<u>Base</u>: quantidade β de símbolos que podem ser utilizados para representar um número em dado sistema de numeração.

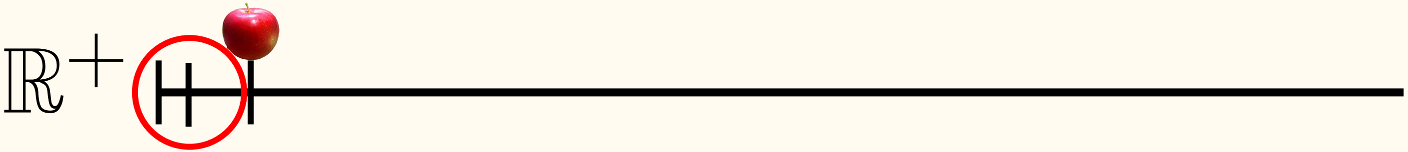


Sistemas de Numeração

<u>Base</u>: quantidade β de símbolos que podem ser utilizados para representar um número em dado sistema de numeração.

Sistema Decimal: 0,33333...



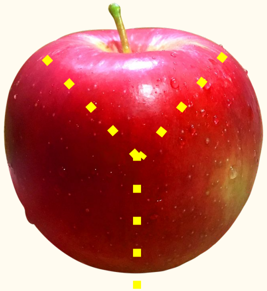


Sistemas de Numeração

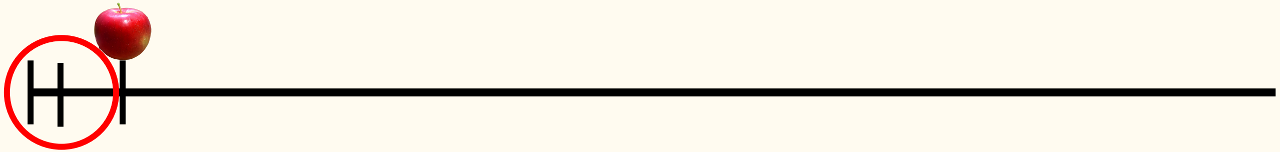
<u>Base</u>: quantidade β de símbolos que podem ser utilizados para representar um número em dado sistema de numeração.

Sistema Decimal: 0,33333...

0,99999... = 1



$\mathbb{R}^+$

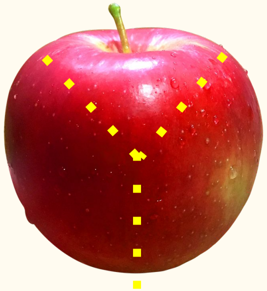


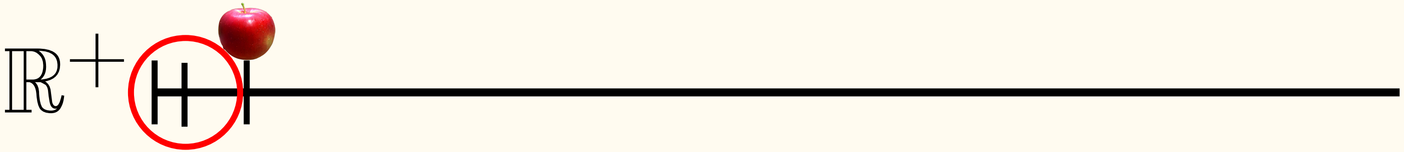
Sistemas de Numeração

<u>Base</u>: quantidade β de símbolos que podem ser utilizados para representar um número em dado sistema de numeração.

Sistema Decimal: 0,33333...

Sistema Numeral Romano: ∷





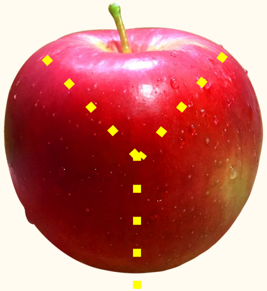
Sistemas de Numeração

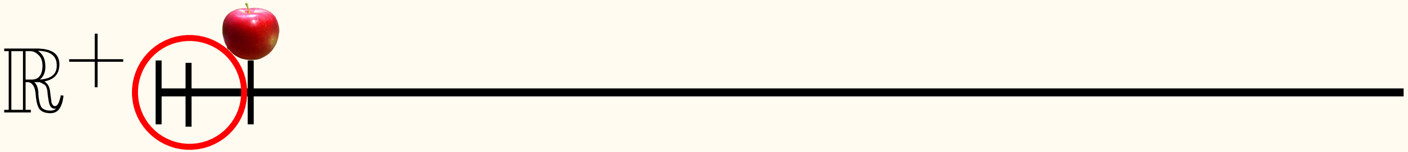
Base: quantidade β de símbolos que podem ser utilizados para representar um número em dado sistema de numeração.

Sistema Decimal: 0,33333...

Sistema Numeral Romano: ∷

Sistema Binário: 0,010101...





Sistemas de Numeração

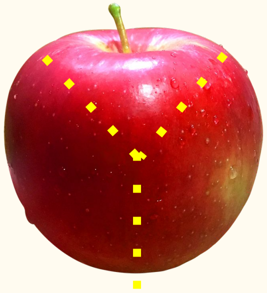
<u>Base</u>: quantidade β de símbolos que podem ser utilizados para representar um número em dado sistema de numeração.

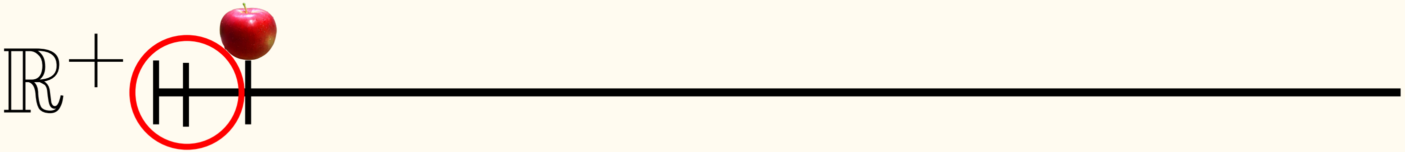
Sistema Decimal: 0,33333...

Sistema Numeral Romano: ∷

Sistema Binário: 0,010101...

Sistema Ternário: 0,1





Sistemas de Numeração

<u>Base</u>: quantidade β de símbolos que podem ser utilizados para representar um número em dado sistema de numeração.

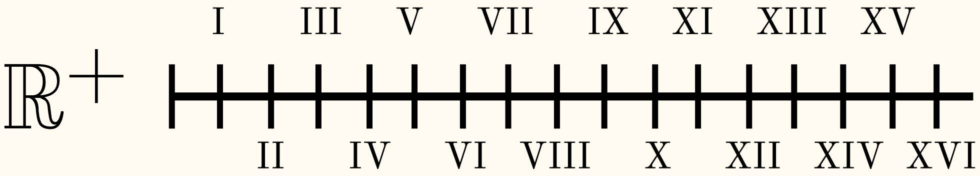
Sistema Decimal: 0,33333...

<u>Sistema Numeral Romano</u>: ∷

Sistema Binário: 0,010101...

Sistema Ternário: 0,1

Sistema Aditivo



Sistemas de Numeração

<u>Base</u>: quantidade β de símbolos que podem ser utilizados para representar um número em dado sistema de numeração.

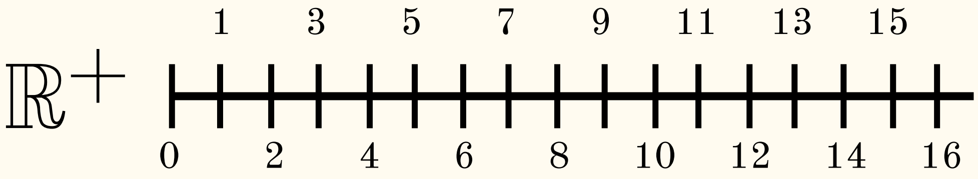
Sistema Posicional

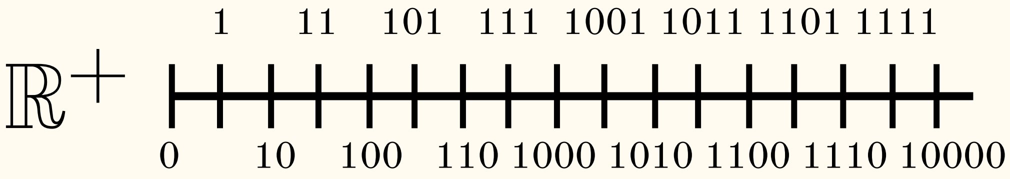
<u>Sistema Decimal</u>: 0,33333...

Sistema Numeral Romano: ∷

<u>Sistema Binário</u>: 0,010101...

<u>Sistema Ternário</u>: 0,1





Sistemas de Numeração

Base: quantidade β de símbolos que podem ser utilizados para representar um número em dado sistema de numeração.

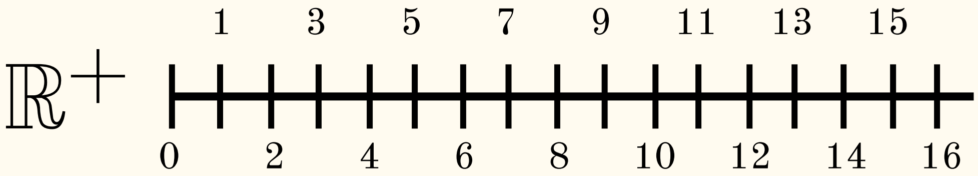
Sistema Posicional

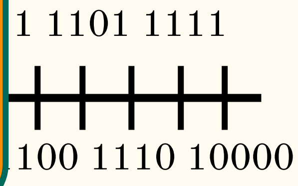
Sistema Decimal: 0,33333...

Sistema Numeral Romano: ∷

Sistema Binário: 0,010101...

Sistema Ternário: 0,1





$$n = a_2 a_1 a_0{,}b_1 b_2 b_3$$

$$n = a_2 * \beta^2 + a_1 * \beta^1 + a_0 * \beta^0 + b_1 * \beta^{-1} + b_2 * \beta^{-2} + b_3 * \beta^{-3}$$

Conversão de Base

Converter da base β para decimal

$$n = (a_2 a_1 a_0, b_1 b_2 b_3)_\beta$$

$$n = a_2 * \beta^2 + a_1 * \beta^1 + a_0 * \beta^0 + b_1 * \beta^{-1} + b_2 * \beta^{-2} + b_3 * \beta^{-3}$$

Conversão de Base

Converter da base β para decimal

$$n = (a_2 a_1 a_0{,}b_1 b_2 b_3)_\beta$$

$$n = a_2 * \beta^2 + a_1 * \beta^1 + a_0 * \beta^0 + b_1 * \beta^{-1} + b_2 * \beta^{-2} + b_3 * \beta^{-3}$$

Exemplo:

$$n = (101010{,}11)_2$$

$$n = 1 * 2^5 + 0 * 2^4 + 1 * 2^3 + 0 * 2^2 + 1 * 2^1 + 0 * 2^0 + 1 * 2^{-1} + 1 * 2^{-2}$$

$$= 32 + 8 + 2 + 0{,}5 + 0{,}25$$

$$= (42{,}75)_{10}$$

Conversão de Base

Converter de decimal para a base β

$$n = (a_2a_1a_0,b_1b_2b_3)_{10}$$

Parte inteira:

$$\begin{array}{llllll}
a_2a_1a_0 & \underline{\big|\ \beta\ } & & & \\
r_0 & q_1 & \underline{\big|\ \beta\ } & & \\
 & r_1 & q_2 & \underline{\big|\ \beta\ } & \\
 & & \ldots & & \\
 & & & q_{n-1} & \underline{\big|\ \beta\ } \\
 & & & r_{n-1} & 0
\end{array}$$

Conversão de Base

Converter de decimal para a base β

$$n = (a_2a_1a_0,b_1b_2b_3)_{10}$$

Parte inteira:

$$\begin{array}{l|l} a_2a_1a_0 & \beta \\ \hline r_0 & q_1 \end{array} \quad \begin{array}{l|l} q_1 & \beta \\ \hline r_1 & q_2 \end{array} \quad \begin{array}{l|l} q_2 & \beta \\ \hline & \end{array} \quad \dots \quad \begin{array}{l|l} q_{n-1} & \beta \\ \hline r_{n-1} & 0 \end{array}$$

Parte fracionária:

$$0,b_1b_2b_3 \times \beta$$

$$d_1,b^1{}_1b^1{}_2b^1{}_3\ldots \Rightarrow 0,b^1{}_1b^1{}_2b^1{}_3\ldots \times \beta$$

$$d_2,b^2{}_1b^2{}_2b^2{}_3\ldots \Rightarrow 0,b^2{}_1b^2{}_2b^2{}_3\ldots \times \beta$$

$$d_3,b^3{}_1b^3{}_2b^3{}_3\ldots$$

$$\dots$$

Conversão de Base

Converter de decimal para a base β

$$n = (a_2a_1a_0,b_1b_2b_3)_{10}$$

Parte inteira:

$$a_2a_1a_0 \;\big|\underline{\;\beta\;}$$

$$r_0 \qquad q_1 \;\big|\underline{\;\beta\;}$$

$$r_1 \qquad q_2 \;\big|\underline{\;\beta\;}$$

$$\cdots$$

$$q_{n-1} \;\big|\underline{\;\beta\;}$$

$$r_{n-1} \qquad 0$$

Parte fracionária:

$$0,b_1b_2b_3 \quad \times \; \beta$$

$$d_1,b^1{}_1b^1{}_2b^1{}_3\ldots \rightarrow 0,b^1{}_1b^1{}_2b^1{}_3\ldots \times \; \beta$$

$$d_2,b^2{}_1b^2{}_2b^2{}_3\ldots \rightarrow 0,b^2{}_1b^2{}_2b^2{}_3\ldots \times \; \beta$$

$$d_3,b^3{}_1b^3{}_2b^3{}_3\ldots$$

$$\cdots$$

$$n = (r_{n-1}\ldots r_1r_0,d_1d_2d_3\ldots)_\beta$$

Conversão de Base

Converter de decimal para a base β

Exemplo: $n = (6{,}34)_{10}$

Parte inteira:

$$\begin{array}{lllll}
6 & \llcorner\, 2 & & & \\
0 & 3 & \llcorner\, 2 & & \\
 & 1 & 1 & \llcorner\, 2 & \\
 & & 1 & 0 &
\end{array}$$

Conversão de Base

Converter de decimal para a base β

Exemplo: $n = (6{,}34)_{10}$

Parte inteira:

$$
\begin{array}{llllll}
6 & \underline{|\ 2\quad} & & & & \\
0 & 3 & \underline{|\ 2\quad} & & & \\
 & 1 & 1 & \underline{|\ 2\quad} & & \\
 & & 1 & 0 & &
\end{array}
$$

Parte fracionária:

$0{,}34 \times 2$

$0{,}68 \Rightarrow 0{,}68 \times 2$

$1{,}36 \Rightarrow 0{,}36 \times 2$

$0{,}72 \Rightarrow 0{,}72 \times 2$

$1{,}44 \Rightarrow 0{,}44 \times 2$

$0{,}88 \Rightarrow 0{,}88 \ldots$

Conversão de Base

Converter de decimal para a base β

Exemplo: $n = (6{,}34)_{10}$

Parte inteira:

$$
\begin{array}{l|l}
6 & 2 \\ \hline
0 & 3 \;|\; 2 \\
 & 1 \quad 1 \;|\; 2 \\
 & \quad\quad 1 \quad 0
\end{array}
$$

$n = (110{,}01010...)_2$

Parte fracionária:

$0{,}34 \times 2$

$0{,}68 \Rightarrow 0{,}68 \times 2$

$1{,}36 \Rightarrow 0{,}36 \times 2$

$0{,}72 \Rightarrow 0{,}72 \times 2$

$1{,}44 \Rightarrow 0{,}44 \times 2$

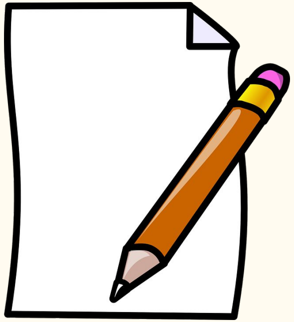
$0{,}88 \Rightarrow 0{,}88 \ldots$

Conversão de Base

Converter da base β_1 para a base β_2

1. Converter da base β_1 para decimal
2. Converter de decimal para a base β_2

Representação em Ponto Flutuante

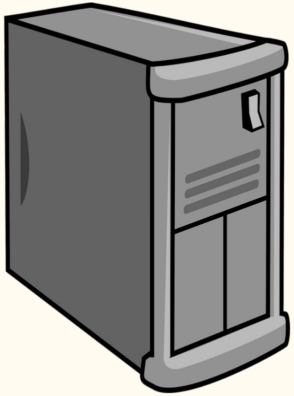


$⅓ = 0{,}33333333333333333333333333333333...$

$\sqrt{2} = 1{,}4142135623730950488016887242096...$

$\pi = 3{,}1415926535897932384626433832795 0...$

Representação em Ponto Flutuante



$$\tfrac{1}{3} = 0{,}33333333333333333333333333333333\ldots$$

$$\sqrt{2} = 1{,}4142135623730950488016887242096\ldots$$

$$\pi = 3{,}14159265358979323846264338327950\ldots$$

Representação em Ponto Flutuante

Representação em Ponto Flutuante

000,00
000,01
000,02
000,03
...
999,99

Representação em Ponto Flutuante

000,00
000,01
000,02
000,03
...
999,99

$\mathbb{R}^+$

Representação em Ponto Flutuante

000,00
000,01
000,02
000,03
...
999,99

$\mathbb{R}^+$: 0 | 0,01 | 0,02 | 0,03 | 0,04 | 0,05 | 0,06 | 0,07 | 0,08

Representação em Ponto Flutuante

000,00
000,01
000,02
000,03
...
999,99

0 999,99

$\mathbb{R}^+$ ....

Representação em Ponto Flutuante

000,00
000,01
000,02
000,03
...
999,99

$$0{,}08 \div 2 = 0{,}04$$

$\mathbb{R}^+$: 0 | 0,01 | 0,02 | 0,03 | 0,04 | 0,05 | 0,06 | 0,07 | 0,08

Representação em Ponto Flutuante

000,00
000,01
000,02
000,03
...
999,99

$$0{,}07 \div 2 = 0{,}035$$

?

$\mathbb{R}^+$: 0 | 0,01 | 0,02 | 0,03 | 0,04 | 0,05 | 0,06 | 0,07 | 0,08

Representação em Ponto Flutuante

000,00
000,01
000,02
000,03
...
999,99

$0{,}07 \div 2 = 0{,}03$

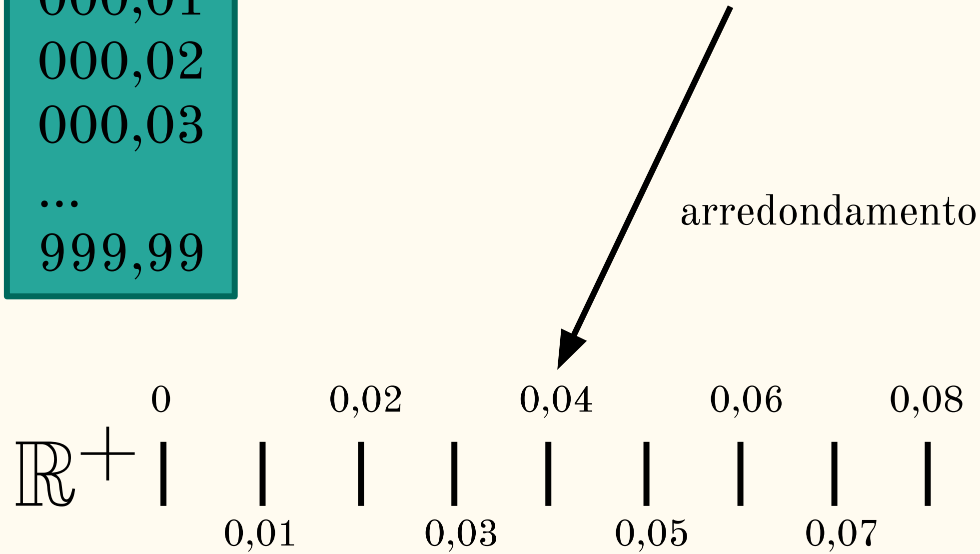
truncamento

	0		0,02		0,04		0,06		0,08
$\mathbb{R}^+$	|	|	|	|	|	|	|	|	|
		0,01		0,03		0,05		0,07	

Representação em Ponto Flutuante

000,00
000,01
000,02
000,03
...
999,99

$$0{,}07 \div 2 = 0{,}04$$



Representação em Ponto Flutuante

000,00
000,01
000,02
000,03
...
999,99

$$0{,}07 \div 2 = 0{,}04$$

$$\boxed{\text{EA} \leq 0{,}01}$$

$\mathbb{R}^+$: 0, 0,01, 0,02, 0,03, 0,04, 0,05, 0,06, 0,07, 0,08

Representação em Ponto Flutuante

000,00
000,01
000,02
000,03
...
999,99

856,786 → 856,79

$$ER = 0{,}004/\ 856{,}786 = 0{,}000005$$

$\mathbb{R}^+$

0 | 0,01 | 0,02 | 0,03 | 0,04 | 0,05 | 0,06 | 0,07 | 0,08

Representação em Ponto Flutuante

000,00
000,01
000,02
000,03
...
999,99

856,786 → 856,79

$$\begin{aligned} ER &= 0{,}004/\ 856{,}786 \\ &= 0{,}000005 \end{aligned}$$

23,546 → 23,55

$$\begin{aligned} ER &= 0{,}004/23{,}546 \\ &= 0{,}0002 \end{aligned}$$

$\mathbb{R}^+$

0 0,02 0,04 0,06 0,08

0,01 0,03 0,05 0,07

Representação em Ponto Flutuante

$$856{,}786 \longrightarrow 8{,}56786 \times 10^2$$

$$23{,}546 \longrightarrow 2{,}3546 \times 10^1$$

Representação em Ponto Flutuante

$856{,}786 \longrightarrow 8{,}56786 \times 10^2$

$23{,}546 \longrightarrow 2{,}3546 \times 10^1$

Representação em Ponto Flutuante

$856{,}786 \rightarrow 8{,}56786 \times 10^2 \rightarrow 8{,}568 \times 10^2$

$23{,}546 \rightarrow 2{,}3546 \times 10^1 \rightarrow 2{,}355 \times 10^1$

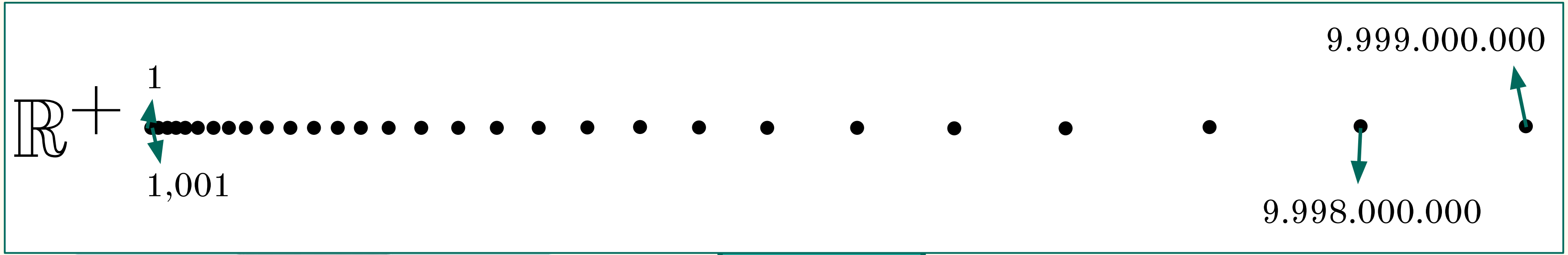
Representação em Ponto Flutuante

$856{,}786 \rightarrow 8{,}56786 \times 10^2 \rightarrow 8{,}568 \times 10^2$

$23{,}546 \rightarrow 2{,}3546 \times 10^1 \rightarrow 2{,}355 \times 10^1$

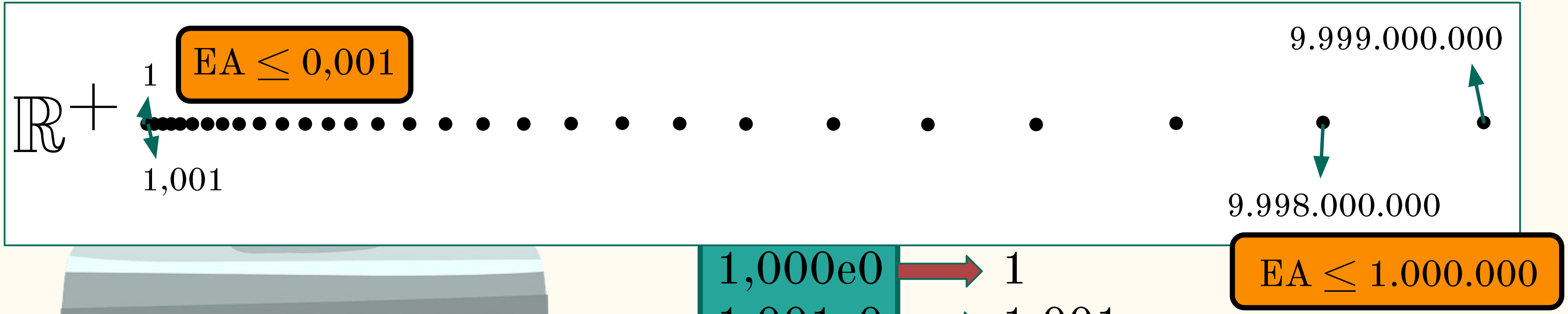
1,000e0 →	1
1,001e0 →	1,001
1,002e0 →	1,002
...	
9,999e0 →	9,999
1,000e1 →	10
1,001e1 →	10,01
...	...
9,999e9 →	9.999.000.000

Representação em Ponto Flutuante



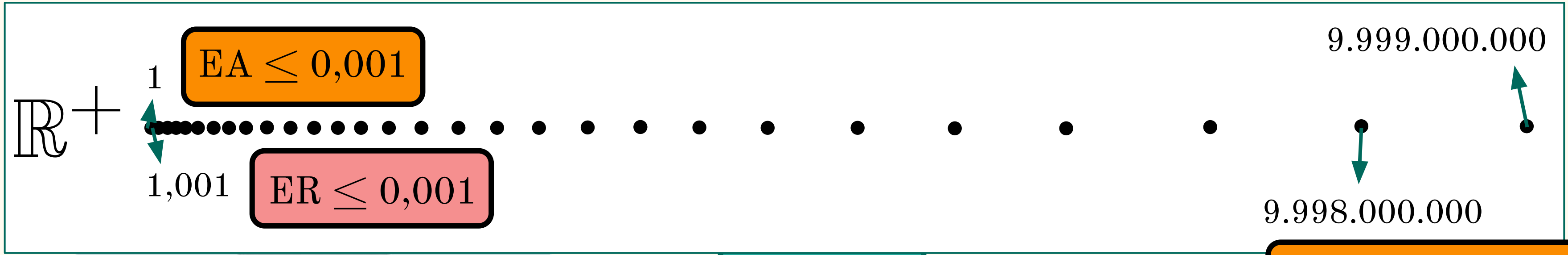
1,000e0 → 1
1,001e0 → 1,001
1,002e0 → 1,002
...
9,999e0 → 9,999
1,000e1 → 10
1,001e1 → 10,01
... ...
9,999e9 → 9.999.000.000

Representação em Ponto Flutuante



1,000e0	1
1,001e0	1,001
1,002e0	1,002
...	
9,999e0	9,999
1,000e1	10
1,001e1	10,01
...	...
9,999e9	9.999.000.000

Representação em Ponto Flutuante



1,000e0	1
1,001e0	1,001
1,002e0	1,002
...	
9,999e0	9,999
1,000e1	10
1,001e1	10,01
...	...
9,999e9	9.999.000.000

$EA \leq 1.000.000$

$ER \leq 0{,}001$

Representação em Ponto Flutuante

$$8{,}56786 \times 10^2 \longrightarrow 8{,}568 \times 10^2$$

$$\boxed{\begin{aligned} \mathrm{ER} &= 0{,}014/\ 856{,}786 \\ &= 0{,}00002 \end{aligned}}$$

$$2{,}3546 \times 10^1 \longrightarrow 2{,}355 \times 10^1$$

$$\boxed{\begin{aligned} \mathrm{ER} &= 0{,}004/23{,}546 \\ &= 0{,}0002 \end{aligned}}$$

Representação em Ponto Flutuante

$d_1, d_2 d_3 \ldots d_t \times 10^e$

$(d_1 d_2 d_3 \ldots d_t)$ é a mantissa com t dígitos

$0 \leq d_i \leq 9$, para todo $i = 1 \ldots t$

$e \in [m, M]$ é o expoente (geralmente $m = -M$)

$d_1 > 0$ (números <u>normalizados</u>)

Representação em Ponto Flutuante

$d_1, d_2 d_3 \ldots d_t \times 10^e$

$(d_1 d_2 d_3 \ldots d_t)$ é a mantissa com t dígitos

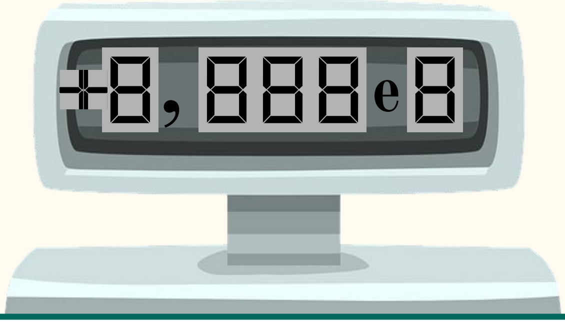
$0 \leq d_i \leq 9$, para todo $i = 1 \ldots t$

$e \in [m, M]$ é o expoente (geralmente $m = -M$)

$d_1 > 0$ (números normalizados)

Como representar o zero?

Representação em Ponto Flutuante



- **Números negativos?**

  “display” de sinal ($\pm$)

$$\pm(d_1,d_2d_3...d_t) \times 10^e$$

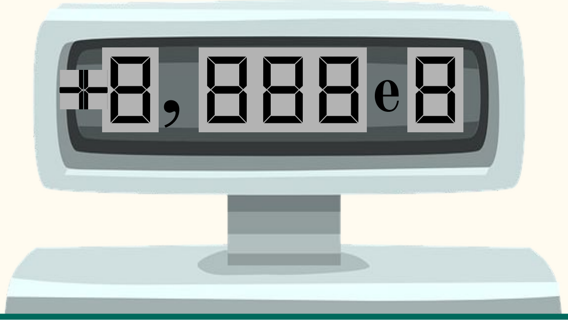
$(d_1d_2d_3...d_t)$ é a mantissa com t dígitos

$0 \leq d_i \leq 9$, para todo $i = 1...t$

$e \in [m, M]$ é o expoente (geralmente $m = -M$)

$d_1 > 0$ (números <u>normalizados</u>)

Representação em Ponto Flutuante



$$\pm(d_1,d_2d_3\ldots d_t) \times 10^{e'-\Delta}$$

- **Números negativos?**

  "display" de sinal ($\pm$)

- **Números menores que 1?**

  expoente negativo: define-se um desvio Δ e subtrai do expoente

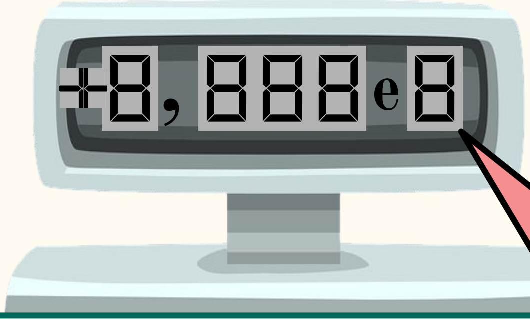
$(d_1d_2d_3\ldots d_t)$ é a mantissa com t dígitos

$0 \leq d_i \leq 9$, para todo $i = 1\ldots t$

$e \in [m, M]$ é o expoente (geralmente $m = -M$)

$d_1 > 0$ (números <u>normalizados</u>)

Representação em Ponto Flutuante



$$\pm(d_1,d_2d_3\ldots d_t) \times 10^{e'-\Delta}$$

$\Delta = 4$

$e' = 0$	10^{-4}	$e' = 5$	10^{1}
$e' = 1$	10^{-3}	$e' = 6$	10^{2}
$e' = 2$	10^{-2}	$e' = 7$	10^{3}
$e' = 3$	10^{-1}	$e' = 8$	10^{4}
$e' = 4$	10^{0}	$e' = 9$	10^{5}

os

m =

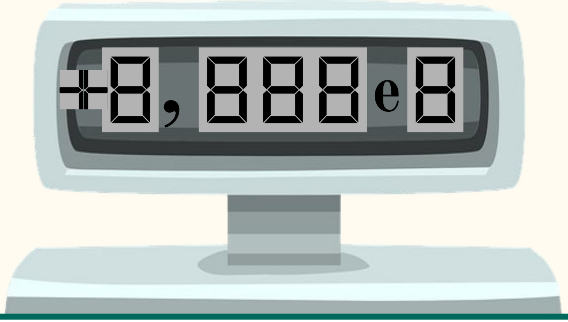
- **Números negativos?**

  "display" de sinal ($\pm$)

- **Números menores que 1?**

  expoente negativo: define-se um desvio Δ e subtrai do expoente

Representação em Ponto Flutuante



- **Números negativos?**

  "display" de sinal ($\pm$)

- **Números menores que 1?**

  expoente negativo: define-se um desvio Δ e subtrai do expoente

- **Outras bases?**

$$\pm(d_1, d_2 d_3 \ldots d_t) \times \beta^{e' - \Delta}$$

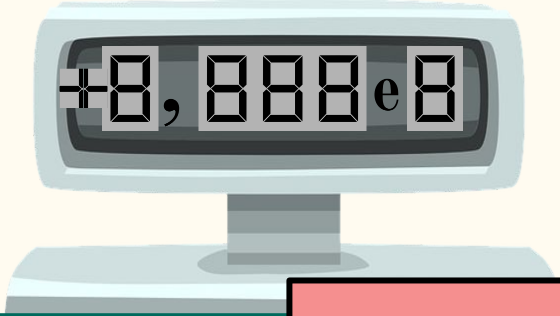
$(d_1 d_2 d_3 \ldots d_t)$ é a mantissa com t dígitos

$0 \leq d_i \leq \beta\text{-}1$, para todo $i = 1 \ldots t$

$e \in [m, M]$ é o expoente (geralmente $m = -M$)

$d_1 > 0$ (números <u>normalizados</u>)

Representação em Ponto Flutuante



$$\pm(d_1,d_2d_3...d_t) \times \beta^{e'-\Delta}$$

$(d_1d_2d_3...d_t)$ é a mantissa com t dígitos

1...t

ralmente m =

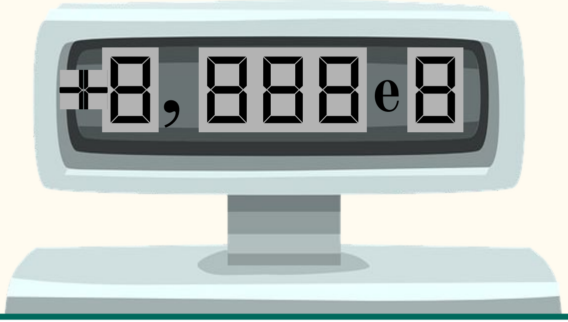
-M)

$d_1 > 0$ (números <u>normalizados</u>)

$$1101{,}011101 \times 2^0 = 11{,}01011101 \times 2^2$$

- **Números nega**
  "display" de si
- **Números men**
  expoente negativo: define-se um desvio Δ e subtrai do expoente
- **Outras bases?**

Representação em Ponto Flutuante



$$\pm(d_1,d_2d_3...d_t) \times \beta^{e'-\Delta}$$

$(d_1d_2d_3...d_t)$ é a mantissa com t dígitos

$0 \leq d_i \leq \beta\text{-}1$, para todo $i = 1...t$

$e \in [m, M]$ é o expoente (geralmente $m = -M$)

$d_1 > 0$ (números normalizados)

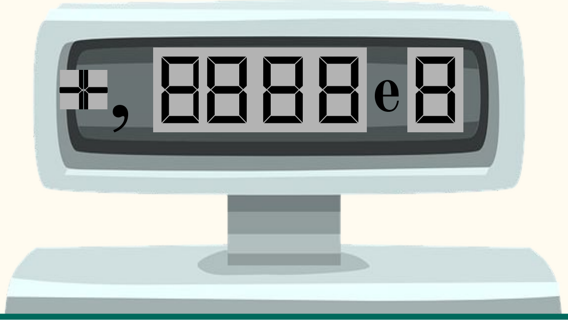
SPF(β, t, m, M)

- **Números negativos?**

  "display" de sinal (±)
- **Números menores que 1?**

  expoente negativo: define-se um desvio Δ e subtrai do expoente
- **Outras bases?**

Representação em Ponto Flutuante



- **Números negativos?**

  “display” de sinal ($\pm$)
- **Números menores que 1?**

  expoente negativo: define-se um desvio Δ e subtrai do expoente
- **Outras bases?**

$$\pm(0,d_1 d_2 d_3 \ldots d_t) \times \beta^{e' - \Delta}$$

$(d_1 d_2 d_3 \ldots d_t)$ é a mantissa com t dígitos

$0 \leq d_i \leq \beta\text{-}1$, para todo $i = 1 \ldots t$

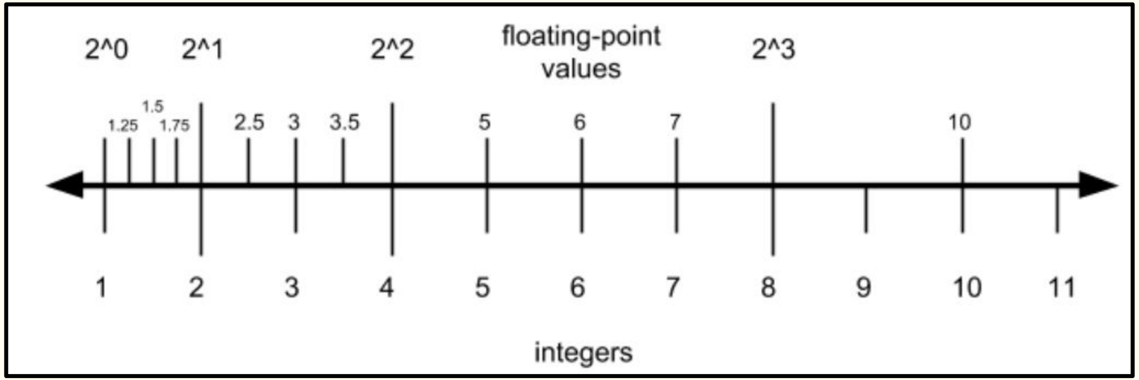
$e \in [m, M]$ é o expoente (geralmente $m = -M$)

$d_1 > 0$ (números normalizados)

SPF(β, t, m, M)

Exemplo

SPF(β=2, t=3, m=0, M=3)



$e = 0$	$e = 1$	$e = 2$	$e = 3$
$1{,}00 \times 2^0 = 1_{10}$	$1{,}00 \times 2^1 = 2_{10}$	$1{,}00 \times 2^2 = 4_{10}$	$1{,}00 \times 2^3 = 8_{10}$
$1{,}01 \times 2^0 = 1{,}25_{10}$	$1{,}01 \times 2^1 = 2{,}5_{10}$	$1{,}01 \times 2^2 = 5_{10}$	$1{,}01 \times 2^3 = 10_{10}$
$1{,}10 \times 2^0 = 1{,}5_{10}$	$1{,}10 \times 2^1 = 3_{10}$	$1{,}10 \times 2^2 = 6_{10}$	$1{,}10 \times 2^3 = 12_{10}$
$1{,}11 \times 2^0 = 1{,}75_{10}$	$1{,}11 \times 2^1 = 3{,}5_{10}$	$1{,}11 \times 2^2 = 7_{10}$	$1{,}11 \times 2^3 = 14_{10}$

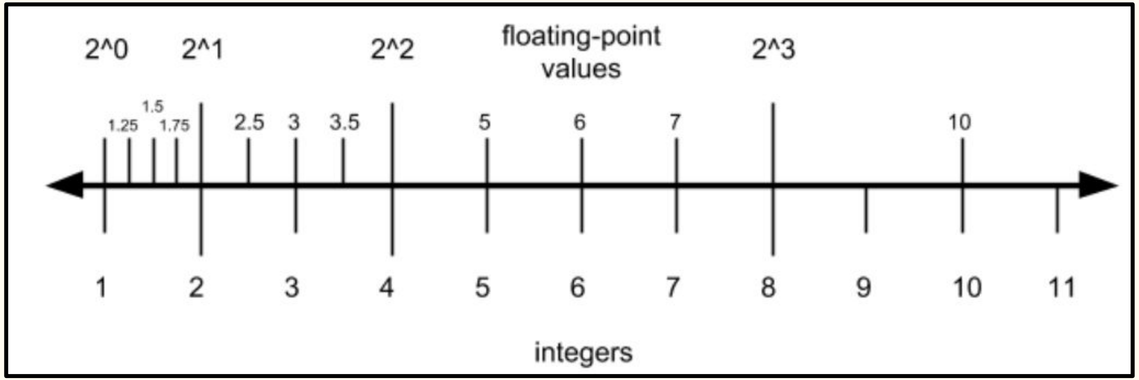
Exemplo

SPF(β=2, t=3, m=0, M=3)

0	1	1	0	1	0

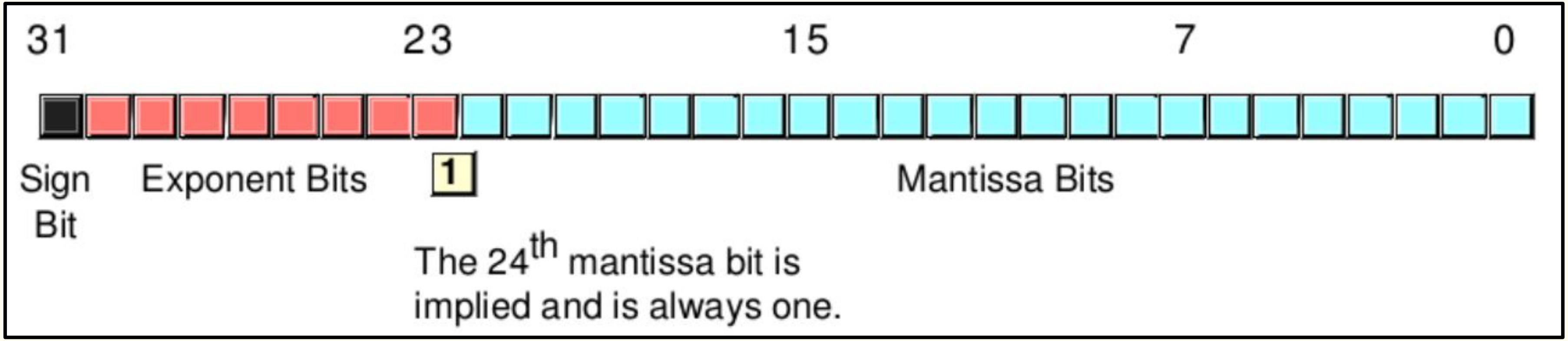
sinal — mantissa — expoente

$(-1)^0 \times 1{,}10 \times 2^{(10\ -\ \Delta)}$



	$e = 1$	$e = 2$	$e = 3$
$1{,}00 \times 2^0 = 1_{10}$	$1{,}00 \times 2^1 = 2_{10}$	$1{,}00 \times 2^2 = 4_{10}$	$1{,}00 \times 2^3 = 8_{10}$
$1{,}01 \times 2^0 = 1{,}25_{10}$	$1{,}01 \times 2^1 = 2{,}5_{10}$	$1{,}01 \times 2^2 = 5_{10}$	$1{,}01 \times 2^3 = 10_{10}$
$1{,}10 \times 2^0 = 1{,}5_{10}$	$1{,}10 \times 2^1 = 3_{10}$	$1{,}10 \times 2^2 = 6_{10}$	$1{,}10 \times 2^3 = 12_{10}$
$1{,}11 \times 2^0 = 1{,}75_{10}$	$1{,}11 \times 2^1 = 3{,}5_{10}$	$1{,}11 \times 2^2 = 7_{10}$	$1{,}11 \times 2^3 = 14_{10}$

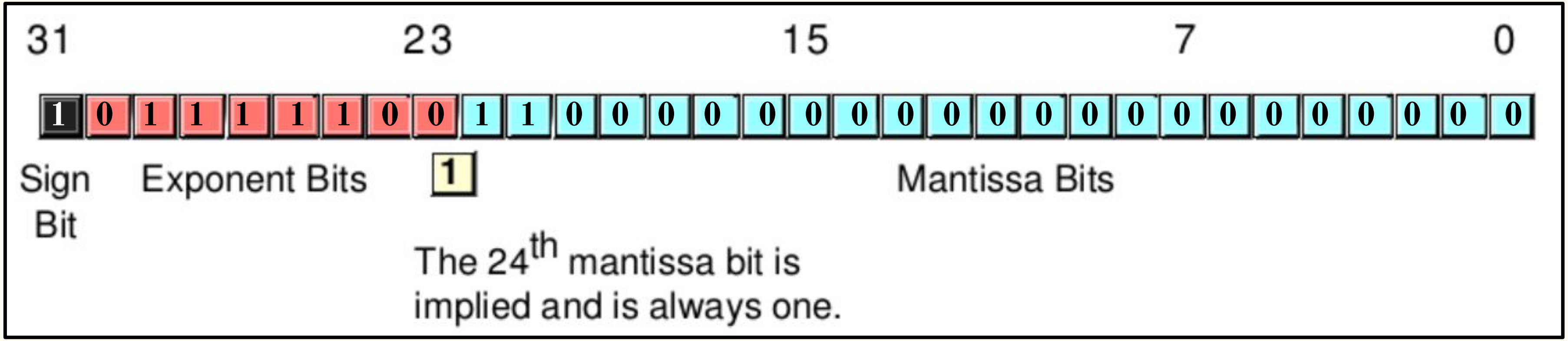
Formato IEEE 754



$$(-1)^{\text{Sign}} \times (1 + \text{Mantissa}) \times 2^{(\text{Exponent - }\Delta)}$$

, onde $\Delta = 127$ para **float** e
$\Delta = 1023$ para **double**

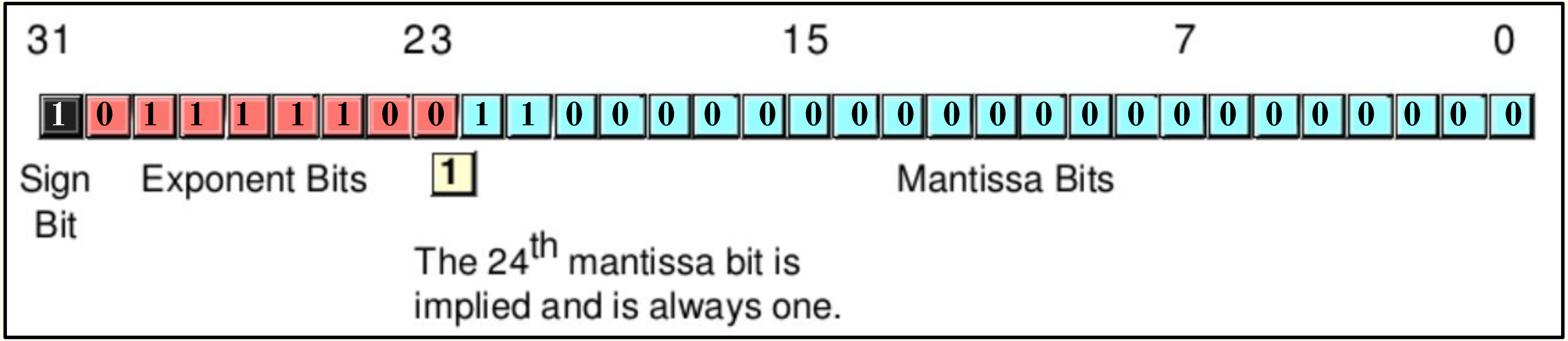
Formato IEEE 754



$$(-1)^{\text{Sign}} \times (1 + \text{Mantissa}) \times 2^{(\text{Exponent - } \Delta)}$$

, onde $\Delta = 127$ para **float** e
$\Delta = 1023$ para **double**

Formato IEEE 754

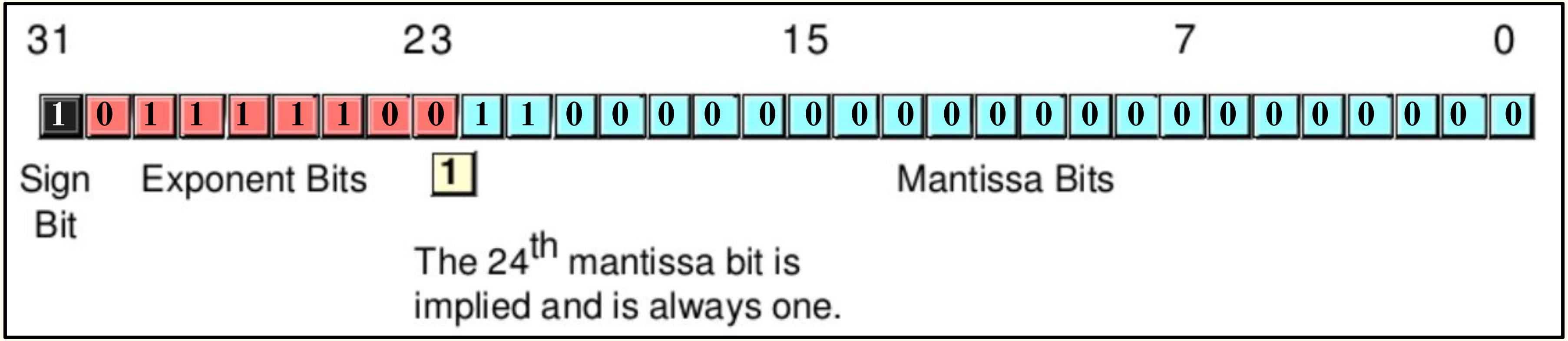


$$(-1)^{\text{Sign}} \times (1 + \text{Mantissa}) \times 2^{(\text{Exponent} - \Delta)}$$

, onde $\Delta = 127$ para **float** e
$\Delta = 1023$ para **double**

Exemplo: $(-1)^1$

Formato IEEE 754



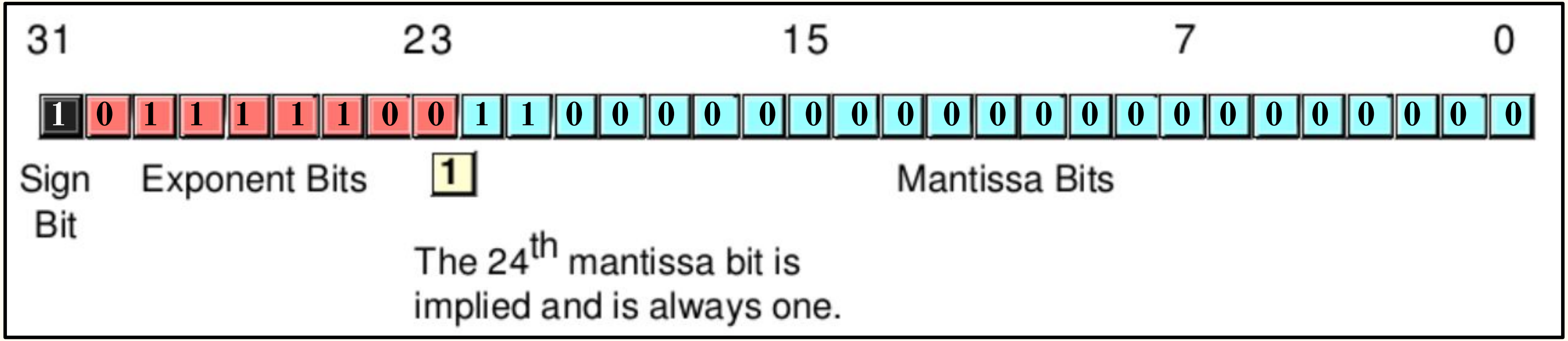
$$(-1)^{\text{Sign}} \times (1 + \text{Mantissa}) \times 2^{(\text{Exponent - }\Delta)}$$

, onde $\Delta = 127$ para **float** e

$\Delta = 1023$ para **double**

Exemplo: $(-1)^1 \times (1 + 0{,}75)$

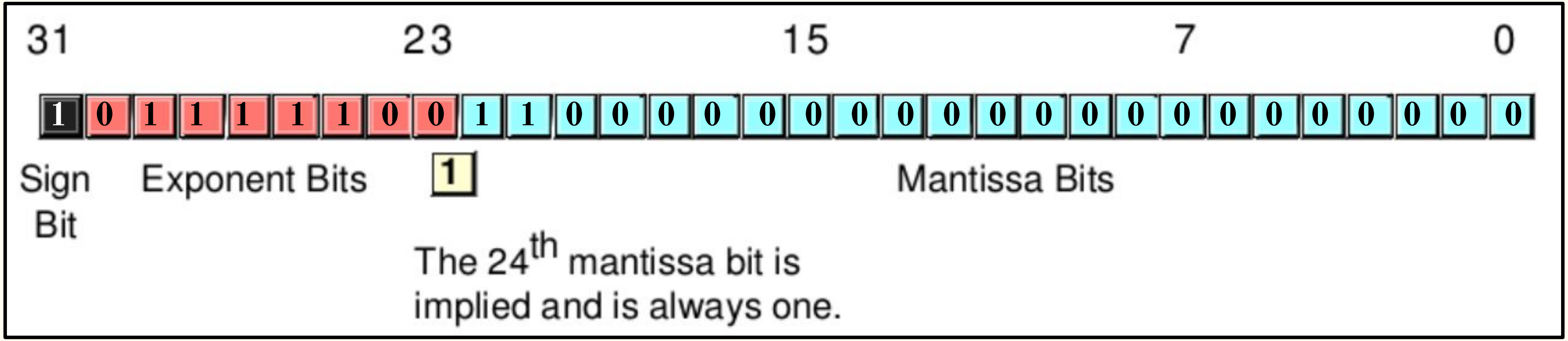
Formato IEEE 754



$(-1)^{\text{Sign}} \times (1 + \text{Mantissa}) \times 2^{(\text{Exponent - }\Delta)}$, onde $\Delta = 127$ para **float** e $\Delta = 1023$ para **double**

Exemplo: $(-1)^{1} \times (1 + 0{,}75) \times 2^{(124\text{ - }127)}$

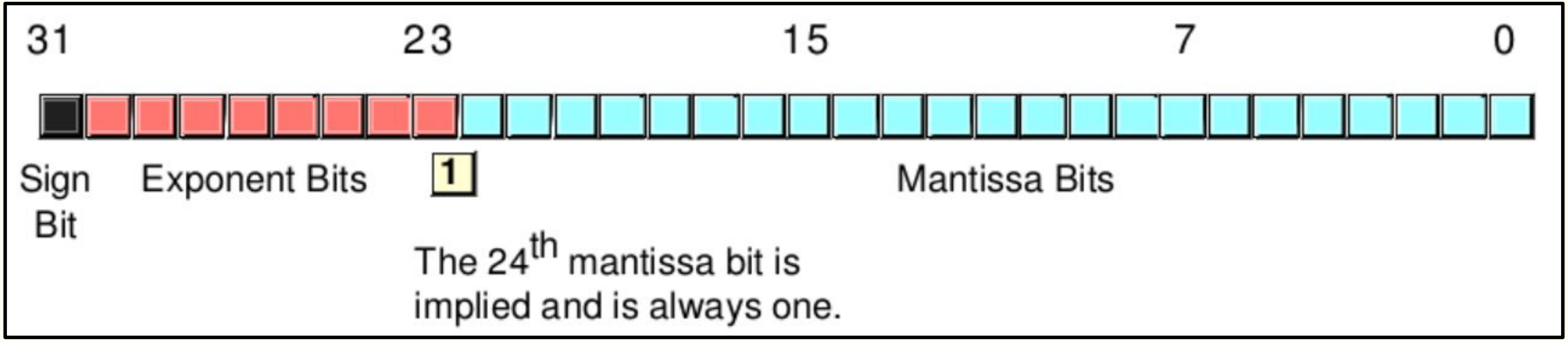
Formato IEEE 754



$$(-1)^{\text{Sign}} \times (1 + \text{Mantissa}) \times 2^{(\text{Exponent - }\Delta)}$$, onde $\Delta = 127$ para **float** e $\Delta = 1023$ para **double**

Exemplo: $(-1)^1 \times (1 + 0{,}75) \times 2^{(124\text{ - }127)} \;=\; -1{,}75 \times 2^{-3} \;=\; -0{,}21875$

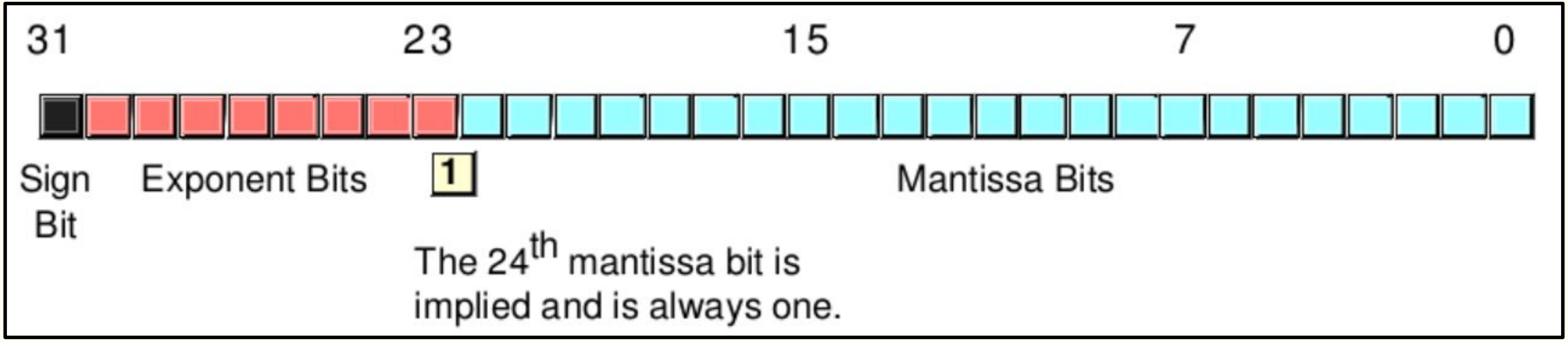
Formato IEEE 754



$$(\text{-}1)^{\text{Sign}} \times (1 + \text{Mantissa}) \times 2^{(\text{Exponent - }\Delta)}$$, onde $\Delta = 127$ para **float** e $\Delta = 1023$ para **double**

Exemplo: $639{,}6875 \;=\; 1001111111{,}1011_2 \;=\; 1{,}0011111111011 \times 2^9$

Formato IEEE 754

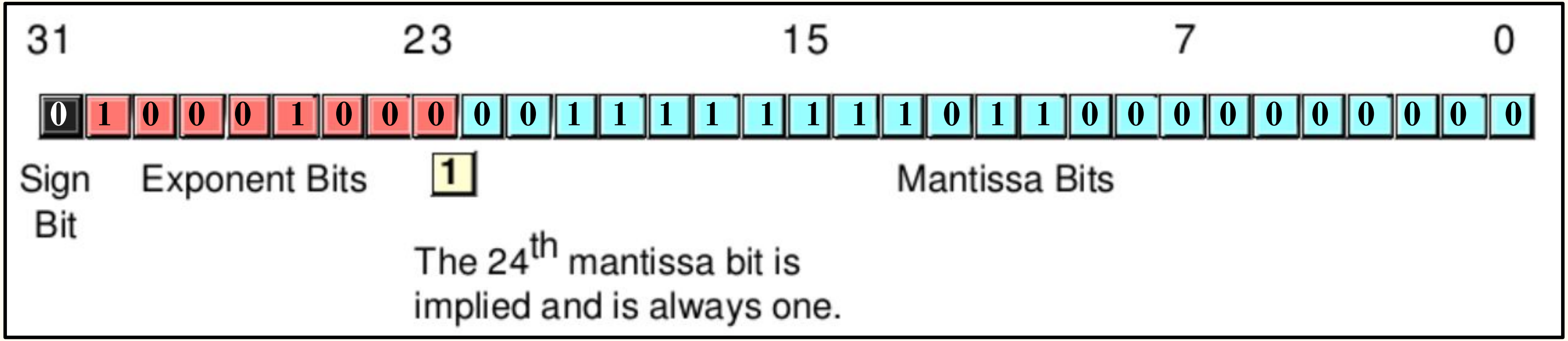


$$(\text{-}1)^{\text{Sign}} \times (1 + \text{Mantissa}) \times 2^{(\text{Exponent - }\Delta)}$$, onde $\Delta = 127$ para **float** e $\Delta = 1023$ para **double**

Exemplo: $639{,}6875 \;=\; 1001111111{,}1011_2 \;=\; 1{,}0011111111011 \times 2^9$

$\text{Sign} = 0 \qquad \text{Exponent} = 9 + 127 = 136 = 10001000_2 \qquad \text{Mantissa} = 0011111111011$

Formato IEEE 754

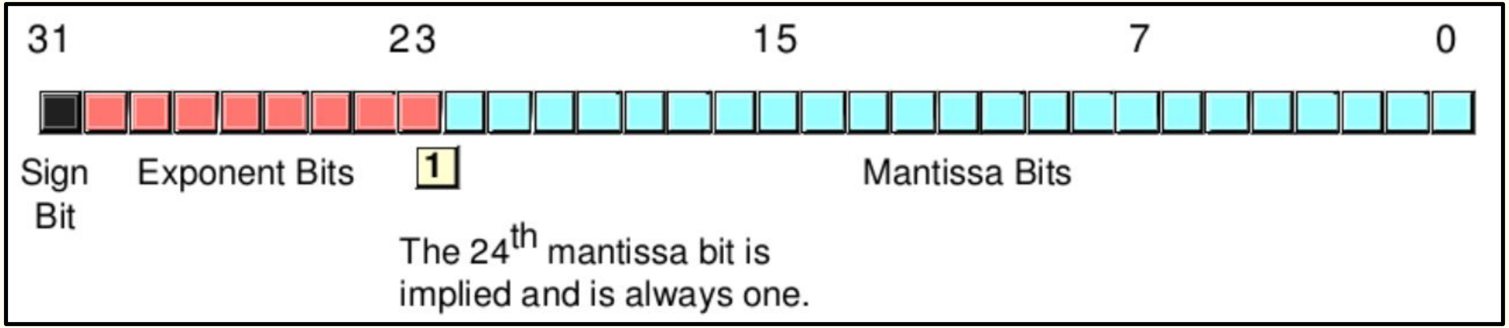


$$(-1)^{\text{Sign}} \times (1 + \text{Mantissa}) \times 2^{(\text{Exponent - }\Delta)} \quad \text{, onde} \quad \begin{array}{l} \Delta = 127 \text{ para } \textbf{float} \text{ e} \\ \Delta = 1023 \text{ para } \textbf{double} \end{array}$$

Exemplo: $639{,}6875 \ = \ 1001111111{,}1011_2 \ = \ 1{,}0011111111011 \times 2^9$

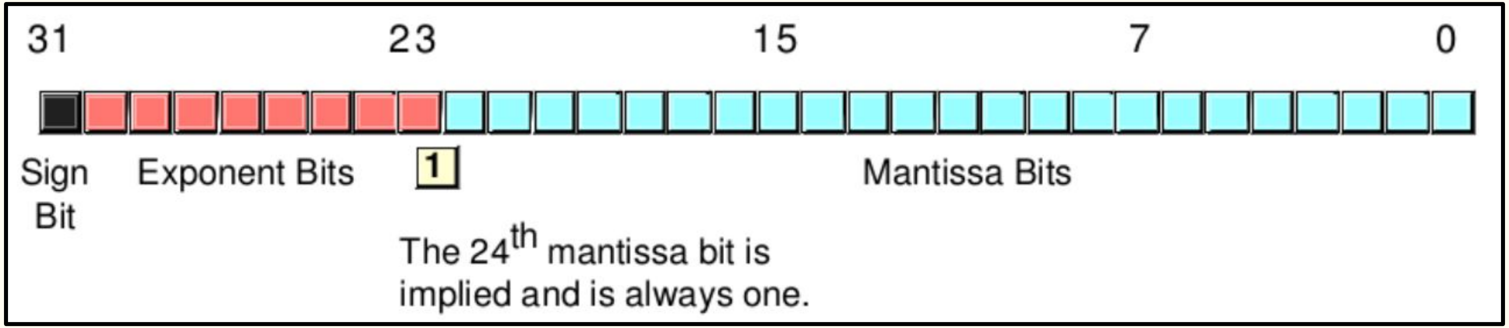
$\text{Sign} = 0 \qquad \text{Exponent} = 9 + 127 = 136 = 10001000_2 \qquad \text{Mantissa} = 0011111111011$

Formato IEEE 754



	float	double
Bits de precisão (mantissa)	23+1	52+1
Precisão decimal (dígitos decimais)	6,5	14,5
Bits do expoente	8	11
Expoente máximo	127	1023
Expoente mínimo	-126	-1022
Deslocamento do expoente	127	1023
Maior/menor número	$10^{\pm 38}$	$10^{\pm 308}$

Formato IEEE 754



	float	double
Bits de precisão (mantissa)	23+1	52+1
Precisão decimal (dígitos decimais)	6,5	14,5
Bits do expoente	8	11
Expoente máximo	127	1023
Expoente mínimo	-126	-1022
Deslocamento do expoente	127	1023
Maior/menor número	$10^{\pm 38}$	$10^{\pm 308}$

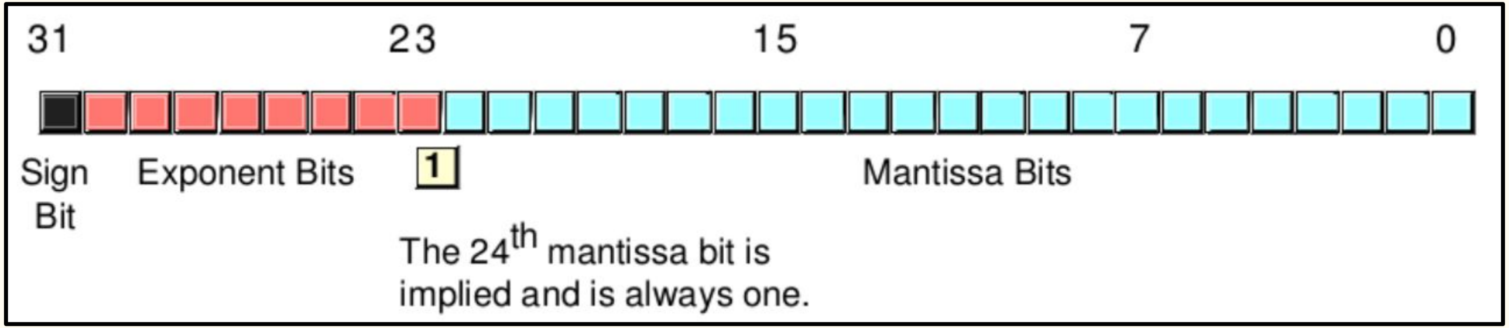
$m = 00000000_2 - 127_{10}$

$M = 11111111_2 - 127_{10}$

$m = -127$

$M = 128$

Formato IEEE 754



Se todos os bits do expoente são 1

Se todos os bits da mantissa são 0

Representa $\pm$**infinity**

Ex: 1/0

Senão

Representa $\pm$**NaN**

Ex: log(-1)

Se todos os bits do expoente são 0

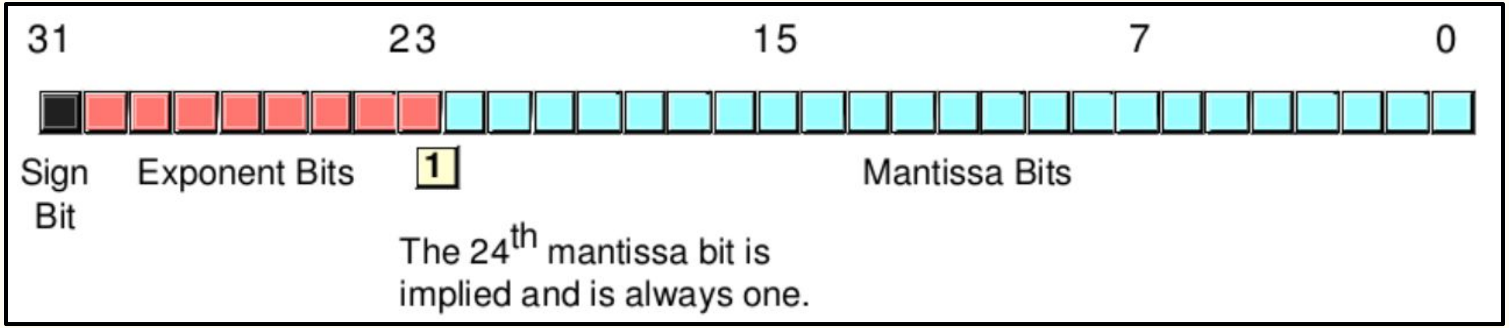
Se todos os bits da mantissa são 0

Representa $\pm$**0**

Senão

Representa um número não normalizado

Formato IEEE 754



<u>Se</u> todos os bits do expoente são 1

<u>Se</u> todos os bits da mantissa são 0

Representa **±infinity** — **Ex:** 1/0

<u>Senão</u>

Representa **±NaN** — **Ex:** log(-1)

<u>Se</u> todos os bits do expoente são 0

$$\cancel{(-1)^{\text{Sign}} \times (1 + \text{Mantissa}) \times 2^{(\text{Exponent} - \Delta)}}$$

$$(-1)^{\text{Sign}} \times (0 + \text{Mantissa}) \times 2^{(-126)}$$

<u>Senão</u>

Representa um número não normalizado

Atenção!

A ordem das operações pode afetar a acurácia do resultado (operações sucessivas que acumulam erros)

$SPF(\beta, t, m, M) = SPF(10, 3, -9, 9)$

- Ex: $x = 1{,}23 \times 10^3$, $y = 1{,}00 \times 10^0$

`z = 0;`
`for (int i=0; i<10; ++i)`
`    z += y`
`z += x`
	`z = x;`
	`for (int i=0; i<10; ++i)`
	`    z += y`

Atenção!

A ordem das operações pode afetar a acurácia do resultado (operações sucessivas que acumulam erros)

$SPF(\beta, t, m, M) = SPF(10,3,-9,9)$

- Ex: $x = 1{,}23 \times 10^3$, $y = 1{,}00 \times 10^0$

`z = 0;`
`for (int i=0; i<10; ++i)`
`    z += y`
`z += x`
	`z = x;`
	`for (int i=0; i<10; ++i)`
	`    z += y`

$i = 0$

$$z = 0{,}00 \times 10^0 + 1{,}00 \times 10^0$$

$$z = 1{,}00 \times 10^0$$

$$z = 1{,}23 \times 10^3 + 1{,}00 \times 10^0$$

$$z = 1{,}23 \times 10^3$$

Atenção!

A ordem das operações pode afetar a acurácia do resultado (operações sucessivas que acumulam erros)

$SPF(\beta, t, m, M) = SPF(10, 3, -9, 9)$

- Ex: $x = 1{,}23 \times 10^3$, $y = 1{,}00 \times 10^0$

`z = 0;`
`for (int i=0; i<10; ++i)`
`    z += y`
`z += x`
	`z = x;`
	`for (int i=0; i<10; ++i)`
	`    z += y`

$i = 0$

$z = 0{,}00 \times 10^0 + 1{,}00 \times 10^0$

$z = 1{,}00 \times 10^0$

$z = 1{,}23 \times 10^3 + 1{,}00 \times 10^0$

$z = 1{,}23 \times 10^3$

$(1{,}23 + 0{,}001) \times 10^3$

$1{,}231 \times 10^3$

Atenção!

A ordem das operações pode afetar a acurácia do resultado (operações sucessivas que acumulam erros)

$SPF(\beta, t, m, M) = SPF(10, 3, -9, 9)$

- Ex: $x = 1{,}23 \times 10^3$, $y = 1{,}00 \times 10^0$

`z = 0;`
`for (int i=0; i<10; ++i)`
`    z += y`
`z += x`
	`z = x;`
	`for (int i=0; i<10; ++i)`
	`    z += y`

$\mathrm{i} = 1$

$$z = 1{,}00 \times 10^0 + 1{,}00 \times 10^0$$

$$z = 2{,}00 \times 10^0$$

$$z = 1{,}23 \times 10^3 + 1{,}00 \times 10^0$$

$$z = 1{,}23 \times 10^3$$

Atenção!

A ordem das operações pode afetar a acurácia do resultado (operações sucessivas que acumulam erros)

$SPF(\beta, t, m, M) = SPF(10, 3, -9, 9)$

- Ex: $x = 1{,}23 \times 10^3$, $y = 1{,}00 \times 10^0$

`z = 0;`
`for (int i=0; i<10; ++i)`
`    z += y`
`z += x`
	`z = x;`
	`for (int i=0; i<10; ++i)`
	`    z += y`

$i = 2$

$$z = 2{,}00 \times 10^0 + 1{,}00 \times 10^0$$

$$z = 3{,}00 \times 10^0$$

$$z = 1{,}23 \times 10^3 + 1{,}00 \times 10^0$$

$$z = 1{,}23 \times 10^3$$

Atenção!

A ordem das operações pode afetar a acurácia do resultado (operações sucessivas que acumulam erros)

$SPF(\beta, t, m, M) = SPF(10, 3, -9, 9)$

- Ex: $x = 1{,}23 \times 10^3$, $y = 1{,}00 \times 10^0$

`z = 0;`
`for (int i=0; i<10; ++i)`
`    z += y`
`z += x`
	`z = x;`
	`for (int i=0; i<10; ++i)`
	`    z += y`

$\mathrm{i} = 3$

$$\mathrm{z} = 3{,}00 \times 10^0 + 1{,}00 \times 10^0$$

$$\mathrm{z} = 4{,}00 \times 10^0$$

$$\mathrm{z} = 1{,}23 \times 10^3 + 1{,}00 \times 10^0$$

$$\mathrm{z} = 1{,}23 \times 10^3$$

Atenção!

A ordem das operações pode afetar a acurácia do resultado (operações sucessivas que acumulam erros)

$SPF(\beta, t, m, M) = SPF(10{,}3, -9{,}9)$

- Ex: $x = 1{,}23 \times 10^3$, $y = 1{,}00 \times 10^0$

`z = 0;`
`for (int i=0; i<10; ++i)`
`    z += y`
`z += x`
	`z = x;`
	`for (int i=0; i<10; ++i)`
	`    z += y`

$i = 4$

$$z = 4{,}00 \times 10^0 + 1{,}00 \times 10^0$$

$$z = 5{,}00 \times 10^0$$

$$z = 1{,}23 \times 10^3 + 1{,}00 \times 10^0$$

$$z = 1{,}23 \times 10^3$$

Atenção!

A ordem das operações pode afetar a acurácia do resultado (operações sucessivas que acumulam erros)

$SPF(\beta, t, m, M) = SPF(10{,}3, -9{,}9)$

- Ex: $x = 1{,}23 \times 10^3$, $y = 1{,}00 \times 10^0$

`z = 0;`
`for (int i=0; i<10; ++i)`
`    z += y`
`z += x`
	`z = x;`
	`for (int i=0; i<10; ++i)`
	`    z += y`

$\mathrm{i} = 5$

$$\mathrm{z} = 5{,}00 \times 10^0 + 1{,}00 \times 10^0$$

$$\mathrm{z} = 6{,}00 \times 10^0$$

$$\mathrm{z} = 1{,}23 \times 10^3 + 1{,}00 \times 10^0$$

$$\mathrm{z} = 1{,}23 \times 10^3$$

Atenção!

A ordem das operações pode afetar a acurácia do resultado (operações sucessivas que acumulam erros)

$SPF(\beta, t, m, M) = SPF(10,3,-9,9)$

- Ex: $x = 1{,}23 \times 10^3$, $y = 1{,}00 \times 10^0$

`z = 0;`
`for (int i=0; i<10; ++i)`
`    z += y`
`z += x`
	`z = x;`
	`for (int i=0; i<10; ++i)`
	`    z += y`

$i = 6$

$$z = 6{,}00 \times 10^0 + 1{,}00 \times 10^0$$

$$z = 7{,}00 \times 10^0$$

$$z = 1{,}23 \times 10^3 + 1{,}00 \times 10^0$$

$$z = 1{,}23 \times 10^3$$

Atenção!

A ordem das operações pode afetar a acurácia do resultado (operações sucessivas que acumulam erros)

$SPF(\beta, t, m, M) = SPF(10, 3, -9, 9)$

- Ex: $x = 1{,}23 \times 10^3$, $y = 1{,}00 \times 10^0$

`z = 0;`
`for (int i=0; i<10; ++i)`
`    z += y`
`z += x`
	`z = x;`
	`for (int i=0; i<10; ++i)`
	`    z += y`

$\mathrm{i} = 7$

$z = 7{,}00 \times 10^0 + 1{,}00 \times 10^0$

$z = 8{,}00 \times 10^0$

$z = 1{,}23 \times 10^3 + 1{,}00 \times 10^0$

$z = 1{,}23 \times 10^3$

Atenção!

A ordem das operações pode afetar a acurácia do resultado (operações sucessivas que acumulam erros)

$SPF(\beta, t, m, M) = SPF(10, 3, -9, 9)$

- Ex: $x = 1{,}23 \times 10^3$, $y = 1{,}00 \times 10^0$

`z = 0;`
`for (int i=0; i<10; ++i)`
`    z += y`
`z += x`
	`z = x;`
	`for (int i=0; i<10; ++i)`
	`    z += y`

$i = 8$

$$z = 8{,}00 \times 10^0 + 1{,}00 \times 10^0$$

$$z = 9{,}00 \times 10^0$$

$$z = 1{,}23 \times 10^3 + 1{,}00 \times 10^0$$

$$z = 1{,}23 \times 10^3$$

Atenção!

A ordem das operações pode afetar a acurácia do resultado (operações sucessivas que acumulam erros)

$SPF(\beta, t, m, M) = SPF(10, 3, -9, 9)$

- Ex: $x = 1{,}23 \times 10^3$, $y = 1{,}00 \times 10^0$

`z = 0;`
`for (int i=0; i<10; ++i)`
`    z += y`
`z += x`
	`z = x;`
	`for (int i=0; i<10; ++i)`
	`    z += y`

$\mathrm{i} = 9$

$$\mathrm{z} = 9{,}00 \times 10^0 + 1{,}00 \times 10^0$$

$$\mathrm{z} = 1{,}00 \times 10^1$$

$$\mathrm{z} = 1{,}23 \times 10^3 + 1{,}00 \times 10^0$$

$$\mathrm{z} = 1{,}23 \times 10^3$$

Atenção!

A ordem das operações pode afetar a acurácia do resultado (operações sucessivas que acumulam erros)

$SPF(\beta, t, m, M) = SPF(10{,}3, -9{,}9)$

- Ex: $x = 1{,}23 \times 10^3$, $y = 1{,}00 \times 10^0$

`z = 0;`
`for (int i=0; i<10; ++i)`
`    z += y`
`z += x`
	`z = x;`
	`for (int i=0; i<10; ++i)`
	`    z += y`

$$z = 1{,}00 \times 10^1 + 1{,}23 \times 10^3$$

$$z = 1{,}24 \times 10^3$$

$$z = 1{,}23 \times 10^3$$

Atenção!

Cadeia de cálculos envolvendo $+$, - , $\times$ e $\div$ ⇒ efetuar $\times$ e $\div$ primeiro.

Atenção!

Cadeia de cálculos envolvendo $+$, - , $\times$ e $\div$ → efetuar $\times$ e $\div$ primeiro.

$x = 1{,}14 \times 10^1$
$y = 3{,}18 \times 10^0$
$z = 5{,}05 \times 10^0$

SPF(β=10, t=3, m=-5, M=5)

Atenção!

Cadeia de cálculos envolvendo $+$, $-$, $\times$ e $\div$ ⟶ efetuar $\times$ e $\div$ primeiro.

$x = 1{,}14 \times 10^1$
$y = 3{,}18 \times 10^0$
$z = 5{,}05 \times 10^0$

SPF(β=10, t=3, m=-5, M=5)

$y \times (z + x)$

$y \times (0{,}505 \ + 1{,}14) \times 10^1$

$(3{,}18 \times 10^0) \times (1{,}64 \times 10^1)$

$(3{,}18 \times 1{,}64) \times 10^{(0+1)}$

$5{,}21 \times 10^1$

Atenção!

Cadeia de cálculos envolvendo $+$, $-$, $\times$ e $\div$ $\Rightarrow$ efetuar $\times$ e $\div$ primeiro.

$x = 1{,}14 \times 10^1$
$y = 3{,}18 \times 10^0$
$z = 5{,}05 \times 10^0$

SPF(β=10, t=3, m=-5, M=5)

$y \times (z + x)$

$y \times (0{,}505 \ + 1{,}14) \times 10^1$

$(3{,}18 \times 10^0) \times (1{,}64 \times 10^1)$

$(3{,}18 \times 1{,}64) \times 10^{(0+1)}$

$5{,}21 \times 10^1$

$(y \times z) + (y \times x)$

$(3{,}18 \times 10^0) \times (5{,}05 \times 10^0) + (3{,}18 \times 10^0) \times (1{,}14 \times 10^1)$

$(3{,}18 \times 5{,}05) \times 10^{(0+0)} + (3{,}18 \times 1{,}14) \times 10^{(0+1)}$

$1{,}60 \times 10^1 + 3{,}62 \times 10^1$

$(1{,}60 + 3{,}62) \times 10^1$

$5{,}22 \times 10^1$

Atenção!

Cadeia de cálculos envolvendo $+$, $-$, $\times$ e $\div$ ⟶ efetuar $\times$ e $\div$ primeiro.

$x = 1{,}14 \times 10^1$
$y = 3{,}18 \times 10^0$
$z = 5{,}05 \times 10^0$

52,311

SPF(β=10, t=3, m=-5, M=5)

$y \times (z + x)$

$y \times (0{,}505 + 1{,}14) \times 10^1$

$(3{,}18 \times 10^0) \times (1{,}64 \times 10^1)$

$(3{,}18 \times 1{,}64) \times 10^{(0+1)}$

$5{,}21 \times 10^1$

$(y \times z) + (y \times x)$

$(3{,}18 \times 10^0) \times (5{,}05 \times 10^0) + (3{,}18 \times 10^0) \times (1{,}14 \times 10^1)$

$(3{,}18 \times 5{,}05) \times 10^{(0+0)} + (3{,}18 \times 1{,}14) \times 10^{(0+1)}$

$1{,}60 \times 10^1 + 3{,}62 \times 10^1$

$(1{,}60 + 3{,}62) \times 10^1$

$5{,}22 \times 10^1$

Atenção!

Multiplicação e Divisão de conjuntos de números ⇨ multiplicar números grandes com número pequenos e dividir números com magnitude semelhantes

Comparando números em Ponto Flutuante

SPF(β=10, t=3, m=-5, M=5)

$x = 1{,}14 \times 10^1$
$y = 3{,}18 \times 10^0$
$z = 5{,}05 \times 10^0$

$a = y \times (z + x)$

$a = 5{,}21 \times 10^1$

$b = (y \times z) + (y \times x)$

$b = 5{,}22 \times 10^1$

Comparando números em Ponto Flutuante

SPF(β=10, t=3, m=-5, M=5)

$x = 1{,}14 \times 10^1$
$y = 3{,}18 \times 10^0$
$z = 5{,}05 \times 10^0$

$a = y \times (z + x)$

$a = 5{,}21 \times 10^1$

$b = (y \times z) + (y \times x)$

$b = 5{,}22 \times 10^1$

```
float a = y*(z+x);
float b = (y*z)+(y*x);

if (a == b)
  return 1;
return 0;
```

Comparando números em Ponto Flutuante

SPF(β=10, t=3, m=-5, M=5)

$x = 1{,}14 \times 10^1$
$y = 3{,}18 \times 10^0$
$z = 5{,}05 \times 10^0$

$a = y \times (z + x)$

$a = 5{,}21 \times 10^1$

$b = (y \times z) + (y \times x)$

$b = 5{,}22 \times 10^1$

```
float a = y*(z+x);
float b = (y*z)+(y*x);

if (a == b)
  return 1;
return 0;
```

```
float a = y*(z+x);
float b = (y*z)+(y*x);

if (abs(a-b) <= erro)
  return 1;
return 0;
```

Comparando números em Ponto Flutuante

- = if abs(x-y) <= error
- ≠ if abs(x-y) > error
- < if (x-y) < error
- ≤ if (x-y) <= error
- > if (x-y) > error
- ≥ if (x-y) >= error

Comparando números em Ponto Flutuante

- = if abs(x-y) <= error
- ≠ if abs(x-y) > error
- < if (x-y) < error
- ≤ if (x-y) <= error
- > if (x-y) > error
- ≥ if (x-y) >= error

Usando erro proporcional aos números

- = if abs(x-y) <= abs(x+y)*erro

Epsilon da Máquina

É a diferença entre 1,0 e o menor valor representável maior que 1,0.

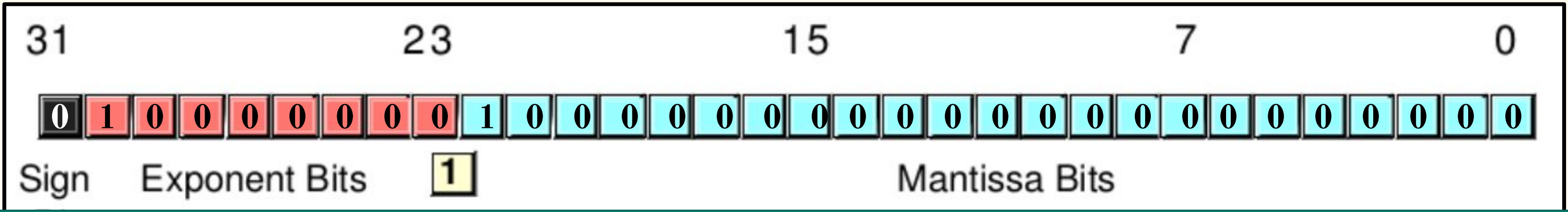
- `float.h: FLT_EPSILON, DBL_EPSILON`

```c
int AlmostEqualRelative(float A, float B)
{
    // Calculate the difference.
    float diff = fabs(A - B);
    A = fabs(A);
    B = fabs(B);
    // Find the largest
    float largest = (B > A) ? B : A;

    if (diff <= largest * FLT_EPSILON)
        return 1;
    return 0;
}
```

ULP

(Units in the Last Place) - Quantos números podem ser representados entre dois números quaisquer.

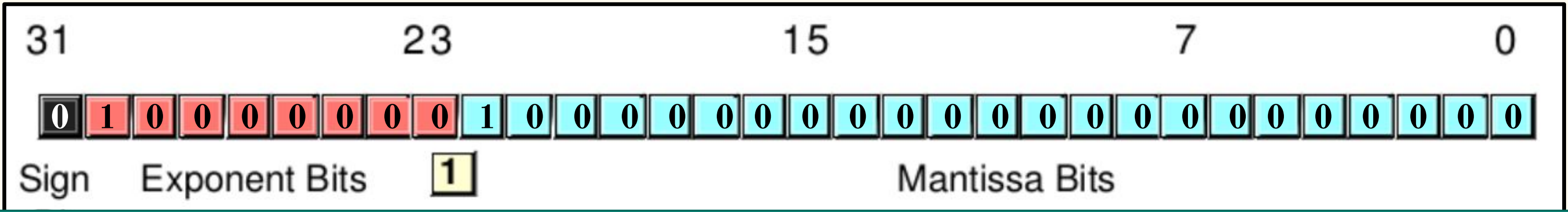


<u>Ponto Flutuante:</u> $1{,}1 \times 2^{128\text{-}127} = 3_{10}$

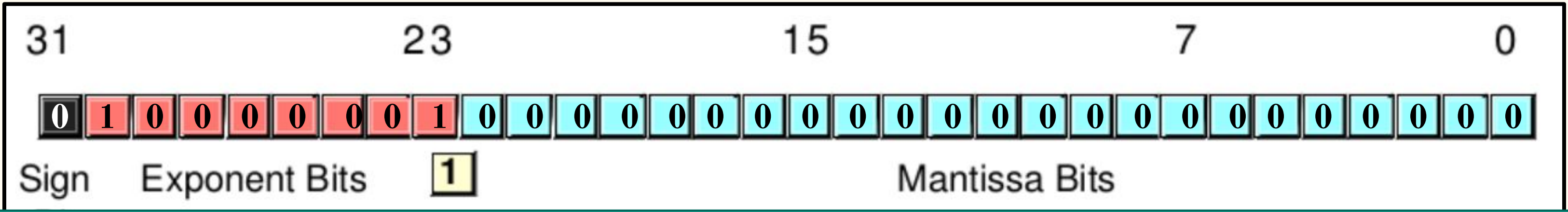
<u>Inteiro:</u> $2^{30} + 2^{22} = 1077936128_{10}$

ULP

(Units in the Last Place) - Quantos números podem ser representados entre dois números quaisquer.



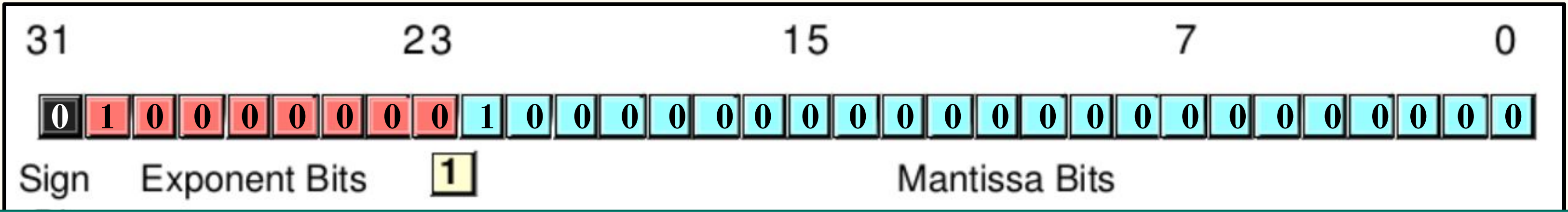
<ins>Ponto Flutuante:</ins> $1{,}1 \times 2^{128\text{-}127} = 3_{10}$ <ins>Inteiro:</ins> $2^{30} + 2^{22} = 1077936128_{10}$



<ins>Ponto Flutuante:</ins> $1{,}0 \times 2^{129\text{-}127} = 4_{10}$ <ins>Inteiro:</ins> $2^{30} + 2^{23} = 1082130432_{10}$

ULP

(Units in the Last Place) - Quantos números podem ser representados entre dois números quaisquer.

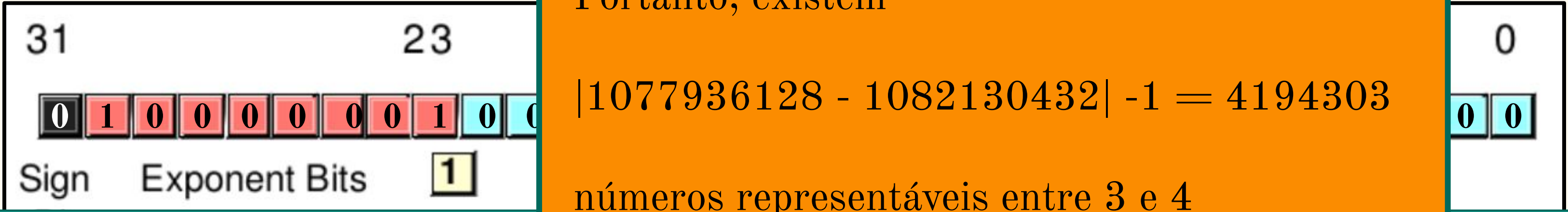


<u>Ponto Flutuante:</u> $1{,}1 \times 2^{128\text{-}127} = 3_{10}$ <u>Inteiro:</u> $2^{30} + 2^{22} = 1077936128_{10}$

Portanto, existem

$|1077936128 - 1082130432| - 1 = 4194303$

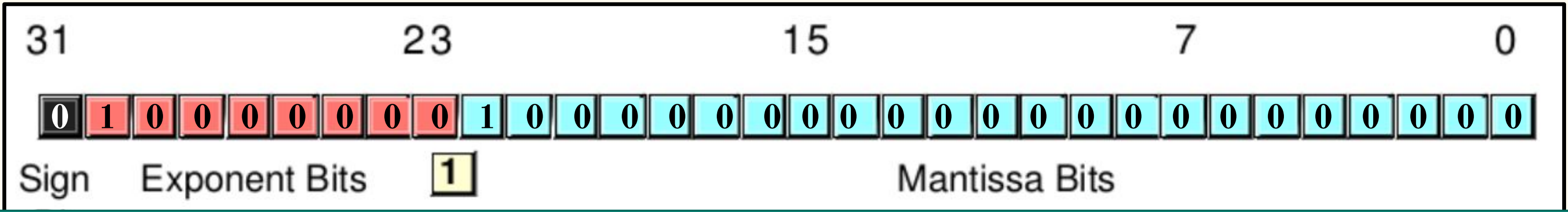
números representáveis entre 3 e 4



<u>Ponto Flutuante:</u> $1{,}0 \times 2^{129\text{-}127} = 4_{10}$ <u>Inteiro:</u> $2 + 2 = 1082130432_{10}$

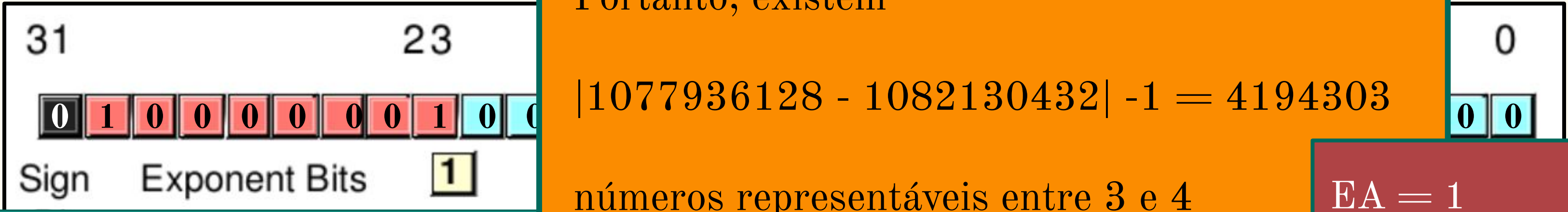
ULP

(Units in the Last Place) - Quantos números podem ser representados entre dois números quaisquer.



Ponto Flutuante: $1{,}1 \times 2^{128\text{-}127} = 3_{10}$

Inteiro: $2^{30} + 2^{22} = 1077936128_{10}$



Ponto Flutuante: $1{,}0 \times 2^{129\text{-}127} = 4_{10}$

Inteiro: $2^{..} + 2^{..}$ 1082130432_{10}

Portanto, existem

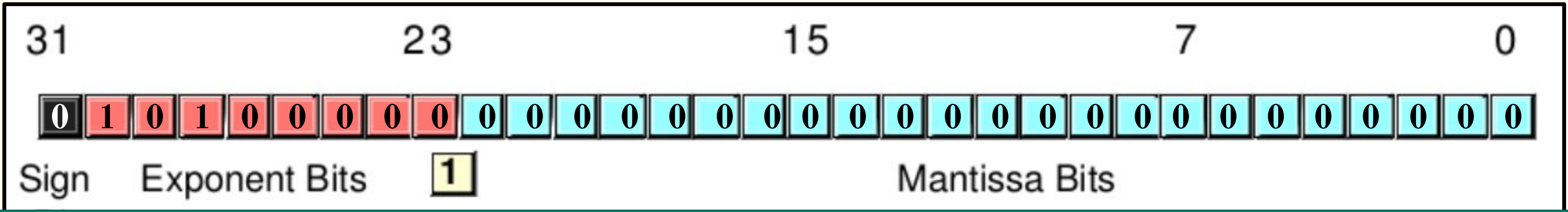
$|1077936128 - 1082130432| - 1 = 4194303$

números representáveis entre 3 e 4

EA = 1
ER = 0,33333

ULP

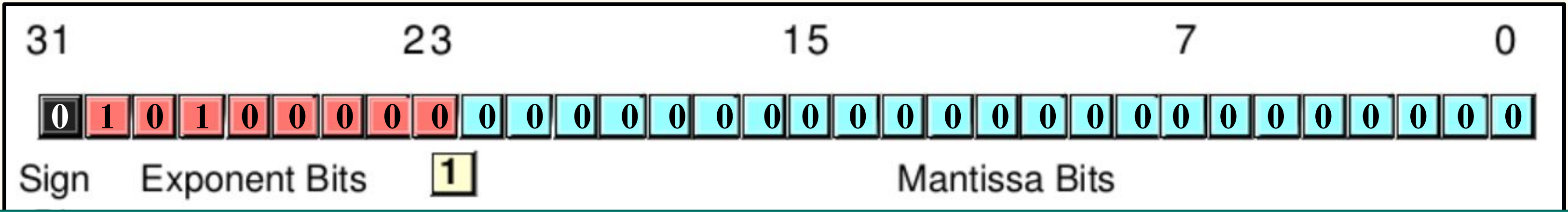
(Units in the Last Place) - Quantos números podem ser representados entre dois números quaisquer.



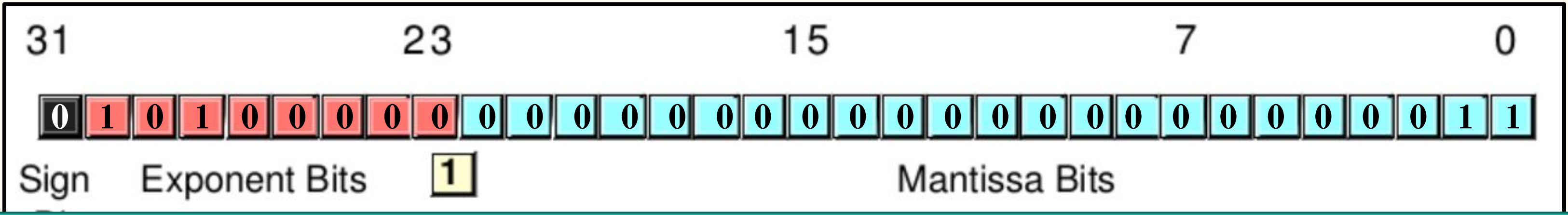
<u>Ponto Flutuante:</u> $1{,}0 \times 2^{160\text{-}127} \cong 8589934592$ <u>Inteiro:</u> $2^{30} + 2^{28} = 1342177280$

ULP

(Units in the Last Place) - Quantos números podem ser representados entre dois números quaisquer.



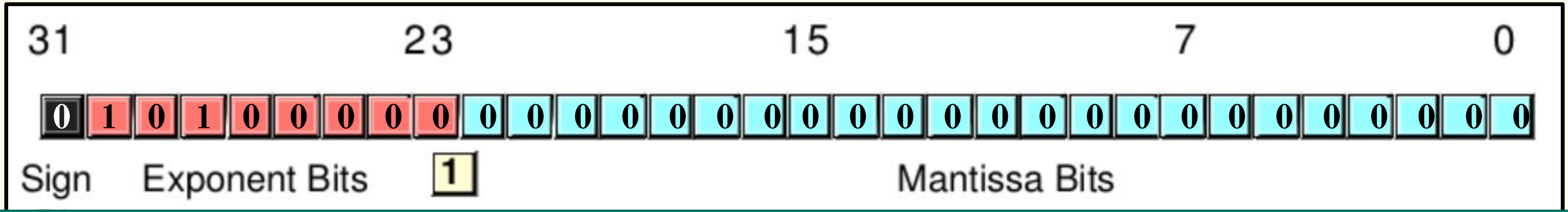
<ins>Ponto Flutuante:</ins> $1{,}0 \times 2^{160-127} \cong 8589934592$ <ins>Inteiro:</ins> $2^{30} + 2^{28} = 1342177280$



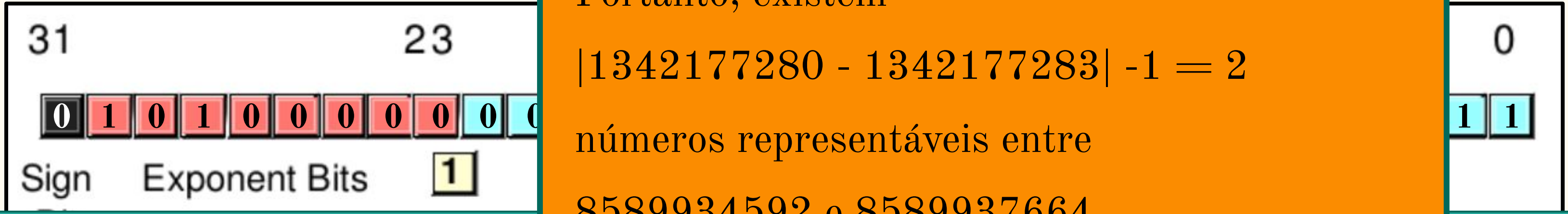
<ins>Ponto Flutuante:</ins> $1{,}0...11 \times 2^{160-127} \cong 8589937664$ <ins>Inteiro:</ins> $2^{30}+2^{28}+2^{1}+2^{0} = 1342177283$

ULP

(Units in the Last Place) - Quantos números podem ser representados entre dois números quaisquer.



<u>Ponto Flutuante:</u> $1{,}0 \times 2^{160\text{-}127} \cong 8589934592$ <u>Inteiro:</u> $2^{30} + 2^{28} = 1342177280$



Portanto, existem

$|1342177280 - 1342177283| -1 = 2$

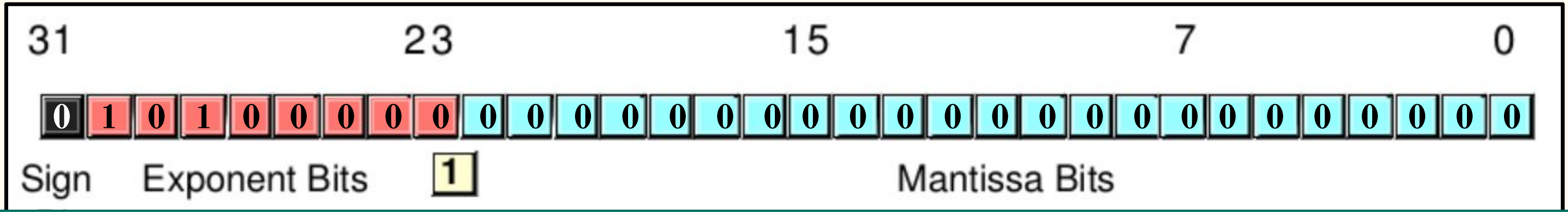
números representáveis entre

8589934592 e 8589937664

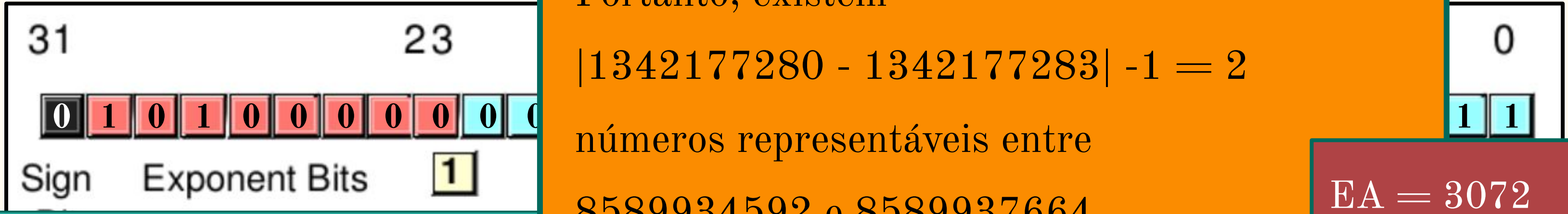
<u>Ponto Flutuante:</u> $1{,}0\ldots11 \times 2^{160\text{-}127} \cong 8589937664$ <u>Inteiro:</u> $2^{30}+2^{28}+2^{1}+2^{0} = 1342177283$

ULP

(Units in the Last Place) - Quantos números podem ser representados entre dois números quaisquer.



<u>Ponto Flutuante:</u> $1{,}0 \times 2^{160\text{-}127} \cong 8589934592$ <u>Inteiro:</u> $2^{30} + 2^{28} =$ 1342177280



<u>Ponto Flutuante:</u> $1{,}0...11 \times 2^{160\text{-}12}$ ... <u>Inteiro:</u> ...
1342177283

Portanto, existem

$|1342177280 - 1342177283| -1 = 2$

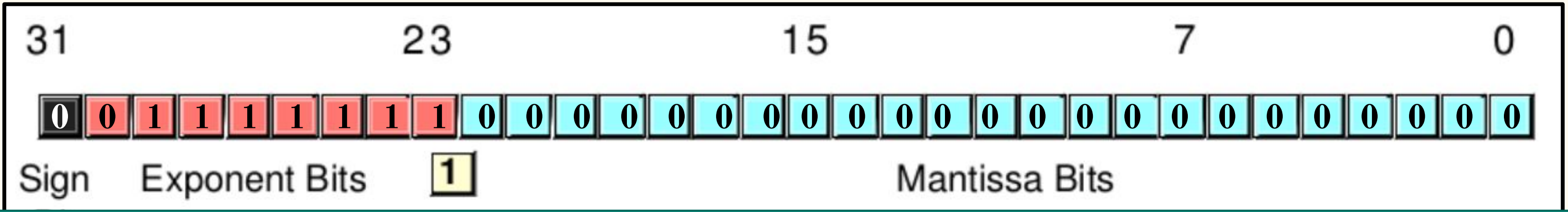
números representáveis entre

8589934592 e 8589937664

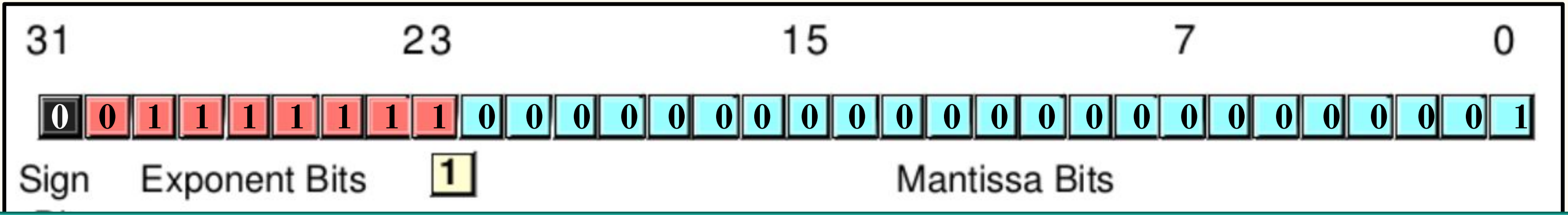
EA = 3072
ER < 0,000001

Epsilon da Máquina

É a diferença entre 1,0 e o menor valor representável maior que 1,0.



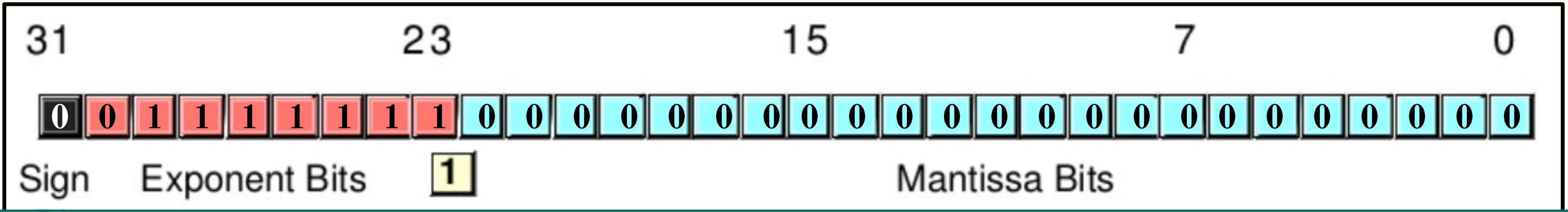
<u>Ponto Flutuante:</u> $1{,}0 \times 2^{127\text{-}127} = 1{,}0$



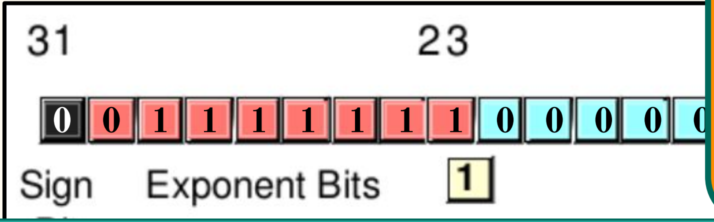
<u>Ponto Flutuante:</u> $1{,}0...01 \times 2^{127\text{-}127} = 1{,}00000011920928955078125$

Epsilon da Máquina

É a diferença entre 1,0 e o menor valor representável maior que 1,0.



Ponto Flutuante: $1{,}0 \times 2^{127\text{-}127} = 1{,}0$

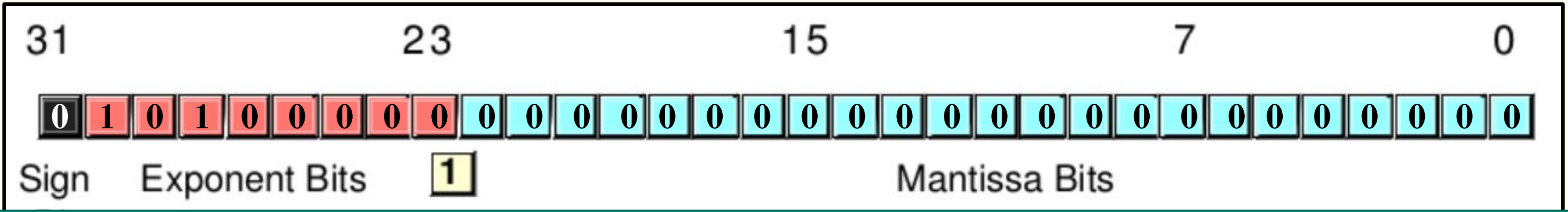


Portanto, $\varepsilon = 0{,}00000011920928955078125$

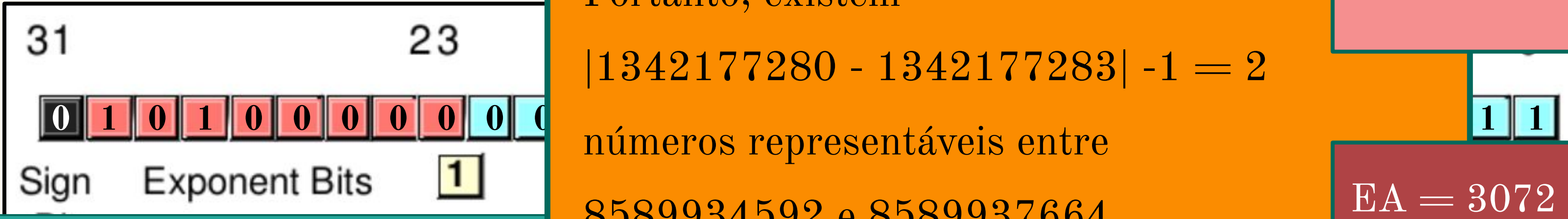
Ponto Flutuante: $1{,}0\ldots01 \times 2^{127\text{-}127} = 1{,}00000011920928955078125$

ULP

(Units in the Last Place) - Quantos números podem ser representados entre dois números quaisquer.



<u>Ponto Flutuante:</u> $1{,}0 \times 2^{160\text{-}127} \cong 8589934592$ <u>Inteiro:</u> $2^{30} + 2^{28} =$ 1342177280



<u>Ponto Flutuante:</u> $1{,}0\ldots11 \times 2^{160\text{-}12}$ ... 1342177283

Portanto, existem

$|1342177280 - 1342177283|$ $-1 = 2$

números representáveis entre

8589934592 e 8589937664

ER $<$ 0,00000012

EA $=$ 3072
ER $<$ 0,000001

Referências

- Daniel Weingaertner; notas de aula da disciplina **Introdução à Computação Científica** (UFPR/DINF)

- A. Kaw, E. Kalu; **Numerical Methods with Applications.** Disponível em https://nm.mathforcollege.com/textbook-numerical-methods-with-applications/

- Bruce Dawson; **Comparing Floating Point Numbers, 2012 Edition.** Disponível em https://randomascii.wordpress.com/2012/02/25/comparing-floating-point-numbers-2012-edition/

- Sérgio Peters e Julio Felipe Szeremeta; **Cálculo Numérico Computacional.** Editora UFSC. Disponível em http://sergiopeters.prof.ufsc.br/livro-calculo-numerico-computacional/

Créditos

Este documento é de autoria do Prof. Guilherme Alex Derenievicz (UFPR/DINF), para uso na disciplina Introdução à Computação Científica (CI1164).