

# CÁLCULO NUMÉRICO

## UERJ/2023

### 01 - Representação de números reais em binário

Rodrigo Madureira  
rodrigo.madureira@ime.uerj.br  
IME-UERJ

# Sumário

- 1 Sistema Decimal - Números inteiros
- 2 Sistema Binário - Números inteiros
- 3 Conversão de bases
- 4 Representação de Números Reais no Computador
- 5 Bibliografia

# Sistema Decimal - Números inteiros

Números na base 10:

Cada posição digital de um número inteiro  $N$  representa uma potência de dez.

$$\begin{aligned} N &= (a_n a_{n-1} \dots a_1 a_0)_{10} \\ &= a_n \times 10^n + a_{n-1} \times 10^{n-1} + \dots + a_1 \times 10^1 + a_0 \times 10^0, \end{aligned}$$

onde  $a_i \in \{0, 1, 2, \dots, 9\}$ .

**Exemplos:**

$$(21)_{10} = 2 \times 10^1 + 1 \times 10^0.$$

$$(2001)_{10} = 2 \times 10^3 + 0 \times 10^2 + 0 \times 10^1 + 1 \times 10^0.$$

$$(23457)_{10} = 2 \times 10^4 + 3 \times 10^3 + 4 \times 10^2 + 5 \times 10^1 + 7 \times 10^0.$$

# Sistema Decimal - Números inteiros

Números na base 10:

Cada posição digital de um número inteiro  $N$  representa uma potência de dez.

$$\begin{aligned} N &= (a_n a_{n-1} \dots a_1 a_0)_{10} \\ &= a_n \times 10^n + a_{n-1} \times 10^{n-1} + \dots + a_1 \times 10^1 + a_0 \times 10^0, \end{aligned}$$

onde  $a_i \in \{0, 1, 2, \dots, 9\}$ .

**Exemplos:**

$$(21)_{10} = 2 \times 10^1 + 1 \times 10^0.$$

$$(2001)_{10} = 2 \times 10^3 + 0 \times 10^2 + 0 \times 10^1 + 1 \times 10^0.$$

$$(23457)_{10} = 2 \times 10^4 + 3 \times 10^3 + 4 \times 10^2 + 5 \times 10^1 + 7 \times 10^0.$$

# Sistema Decimal - Números inteiros

Números na base 10:

Cada posição digital de um número inteiro  $N$  representa uma potência de dez.

$$\begin{aligned} N &= (a_n a_{n-1} \dots a_1 a_0)_{10} \\ &= a_n \times 10^n + a_{n-1} \times 10^{n-1} + \dots + a_1 \times 10^1 + a_0 \times 10^0, \end{aligned}$$

onde  $a_i \in \{0, 1, 2, \dots, 9\}$ .

**Exemplos:**

$$(21)_{10} = 2 \times 10^1 + 1 \times 10^0.$$

$$(2001)_{10} = 2 \times 10^3 + 0 \times 10^2 + 0 \times 10^1 + 1 \times 10^0.$$

$$(23457)_{10} = 2 \times 10^4 + 3 \times 10^3 + 4 \times 10^2 + 5 \times 10^1 + 7 \times 10^0.$$

# Sistema Decimal - Números inteiros

Números na base 10:

Cada posição digital de um número inteiro  $N$  representa uma potência de dez.

$$\begin{aligned} N &= (a_n a_{n-1} \dots a_1 a_0)_{10} \\ &= a_n \times 10^n + a_{n-1} \times 10^{n-1} + \dots + a_1 \times 10^1 + a_0 \times 10^0, \end{aligned}$$

onde  $a_i \in \{0, 1, 2, \dots, 9\}$ .

**Exemplos:**

$$(21)_{10} = 2 \times 10^1 + 1 \times 10^0.$$

$$(2001)_{10} = 2 \times 10^3 + 0 \times 10^2 + 0 \times 10^1 + 1 \times 10^0.$$

$$(23457)_{10} = 2 \times 10^4 + 3 \times 10^3 + 4 \times 10^2 + 5 \times 10^1 + 7 \times 10^0.$$

# Sistema Binário - Números inteiros

Base 2: usada nos computadores binários.

Cada posição digital de um número inteiro  $N$  representa uma potência de dois.

$$\begin{aligned} N &= (a_n a_{n-1} \dots a_1 a_0)_2 \\ &= a_n \times 2^n + a_{n-1} \times 2^{n-1} + \dots + a_1 \times 2^1 + a_0 \times 2^0, \end{aligned}$$

onde  $a_i \in \{0, 1\}$ .

**Exemplos:**

$$(101)_2 = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0.$$

$$(1010)_2 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0.$$

$$(110101)_2 = 1 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0.$$

# Sistema Binário - Números inteiros

Base 2: usada nos computadores binários.

Cada posição digital de um número inteiro  $N$  representa uma potência de dois.

$$\begin{aligned} N &= (a_n a_{n-1} \dots a_1 a_0)_2 \\ &= a_n \times 2^n + a_{n-1} \times 2^{n-1} + \dots + a_1 \times 2^1 + a_0 \times 2^0, \end{aligned}$$

onde  $a_i \in \{0, 1\}$ .

## Exemplos:

$$(101)_2 = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0.$$

$$(1010)_2 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0.$$

$$(110101)_2 = 1 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0.$$



# Sistema Binário - Números inteiros

Base 2: usada nos computadores binários.

Cada posição digital de um número inteiro  $N$  representa uma potência de dois.

$$\begin{aligned} N &= (a_n a_{n-1} \dots a_1 a_0)_2 \\ &= a_n \times 2^n + a_{n-1} \times 2^{n-1} + \dots + a_1 \times 2^1 + a_0 \times 2^0, \end{aligned}$$

onde  $a_i \in \{0, 1\}$ .

## Exemplos:

$$(101)_2 = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0.$$

$$(1010)_2 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0.$$

$$(110101)_2 = 1 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0.$$

# Sistema Binário - Números inteiros

Base 2: usada nos computadores binários.

Cada posição digital de um número inteiro  $N$  representa uma potência de dois.

$$\begin{aligned} N &= (a_n a_{n-1} \dots a_1 a_0)_2 \\ &= a_n \times 2^n + a_{n-1} \times 2^{n-1} + \dots + a_1 \times 2^1 + a_0 \times 2^0, \end{aligned}$$

onde  $a_i \in \{0, 1\}$ .

## Exemplos:

$$(101)_2 = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0.$$

$$(1010)_2 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0.$$

$$(110101)_2 = 1 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0.$$

# Conversão de bases

Um número na base  $\beta$  pode ser convertido para base decimal como

$$\begin{aligned} N &= (a_n a_{n-1} \dots a_1 a_0)_{10} \\ &= a_n \times \beta^n + a_{n-1} \times \beta^{n-1} + \dots + a_1 \times \beta^1 + a_0 \times \beta^0, \end{aligned}$$

onde  $a_i$  são os dígitos do número representado na base  $\beta$ .

**Exemplo 1:** Converta  $(110)_2$  para a base decimal.

$$(110)_2 = 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = (6)_{10}.$$

**Exemplo 2:** Converta  $(1001)_2$  para a base decimal.

$$(1001)_2 = 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = (9)_{10}.$$

# Conversão de bases

Um número na base  $\beta$  pode ser convertido para base decimal como

$$\begin{aligned} N &= (a_n a_{n-1} \dots a_1 a_0)_{10} \\ &= a_n \times \beta^n + a_{n-1} \times \beta^{n-1} + \dots + a_1 \times \beta^1 + a_0 \times \beta^0, \end{aligned}$$

onde  $a_i$  são os dígitos do número representado na base  $\beta$ .

**Exemplo 1:** Converta  $(110)_2$  para a base decimal.

$$(110)_2 = 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = (6)_{10}.$$

**Exemplo 2:** Converta  $(1001)_2$  para a base decimal.

$$(1001)_2 = 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = (9)_{10}.$$

# Conversão de bases

Um número na base  $\beta$  pode ser convertido para base decimal como

$$\begin{aligned} N &= (a_n a_{n-1} \dots a_1 a_0)_{10} \\ &= a_n \times \beta^n + a_{n-1} \times \beta^{n-1} + \dots + a_1 \times \beta^1 + a_0 \times \beta^0, \end{aligned}$$

onde  $a_i$  são os dígitos do número representado na base  $\beta$ .

**Exemplo 1:** Converta  $(110)_2$  para a base decimal.

$$(110)_2 = 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = (6)_{10}.$$

**Exemplo 2:** Converta  $(1001)_2$  para a base decimal.

$$(1001)_2 = 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = (9)_{10}.$$

# Conversão de bases

Um número na base  $\beta$  pode ser convertido para base decimal como

$$\begin{aligned} N &= (a_n a_{n-1} \dots a_1 a_0)_{10} \\ &= a_n \times \beta^n + a_{n-1} \times \beta^{n-1} + \dots + a_1 \times \beta^1 + a_0 \times \beta^0, \end{aligned}$$

onde  $a_i$  são os dígitos do número representado na base  $\beta$ .

**Exemplo 1:** Converta  $(110)_2$  para a base decimal.

$$(110)_2 = 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = (6)_{10}.$$

**Exemplo 2:** Converta  $(1001)_2$  para a base decimal.

$$(1001)_2 = 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = (9)_{10}.$$

# Conversão de bases

Um número na base  $\beta$  pode ser convertido para base decimal como

$$\begin{aligned} N &= (a_n a_{n-1} \dots a_1 a_0)_{10} \\ &= a_n \times \beta^n + a_{n-1} \times \beta^{n-1} + \dots + a_1 \times \beta^1 + a_0 \times \beta^0, \end{aligned}$$

onde  $a_i$  são os dígitos do número representado na base  $\beta$ .

**Exemplo 1:** Converta  $(110)_2$  para a base decimal.

$$(110)_2 = 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = (6)_{10}.$$

**Exemplo 2:** Converta  $(1001)_2$  para a base decimal.

$$(1001)_2 = 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = (9)_{10}.$$

# Conversão de bases

Conversão da base decimal para a base  $\beta$ :

Divisões sucessivas do número em base decimal por  $\beta$  até que o quociente seja igual a zero.

O número na base  $\beta$  é formado pela concatenação em ordem inversa dos restos das divisões.

**Exemplo 1:** Converta  $(25)_{10}$  para a base 2.

$$\begin{array}{r}
 25 \quad | \quad 2 \\
 \hline
 24 \quad | \quad 12 \quad | \quad 2 \\
 \hline
 1 \quad | \quad 12 \quad | \quad 6 \quad | \quad 2 \\
 \hline
 \quad 0 \quad | \quad 6 \quad | \quad 3 \quad | \quad 2 \\
 \hline
 \quad \quad 0 \quad | \quad 2 \quad | \quad 1 \\
 \hline
 \quad \quad \quad 1
 \end{array}$$

←

$$25 = 11001_2$$



# Conversão de bases

Conversão da base decimal para a base  $\beta$ :

Divisões sucessivas do número em base decimal por  $\beta$  até que o quociente seja igual a zero.

O número na base  $\beta$  é formado pela concatenação em ordem inversa dos restos das divisões.

**Exemplo 1:** Converta  $(25)_{10}$  para a base 2.

$$\begin{array}{r}
 25 \quad | \quad 2 \\
 \hline
 24 \quad | \quad 12 \quad | \quad 2 \\
 \hline
 1 \quad | \quad 12 \quad | \quad 6 \quad | \quad 2 \\
 \hline
 \quad 0 \quad | \quad 6 \quad | \quad 3 \quad | \quad 2 \\
 \hline
 \quad \quad 0 \quad | \quad 2 \quad | \quad 1 \\
 \hline
 \quad \quad \quad 1
 \end{array}$$

←

$$25 = 11001_2$$

# Conversão de bases

Conversão da base decimal para a base  $\beta$ :

Divisões sucessivas do número em base decimal por  $\beta$  até que o quociente seja igual a zero.

O número na base  $\beta$  é formado pela concatenação em ordem inversa dos restos das divisões.

**Exemplo 1:** Converta  $(25)_{10}$  para a base 2.

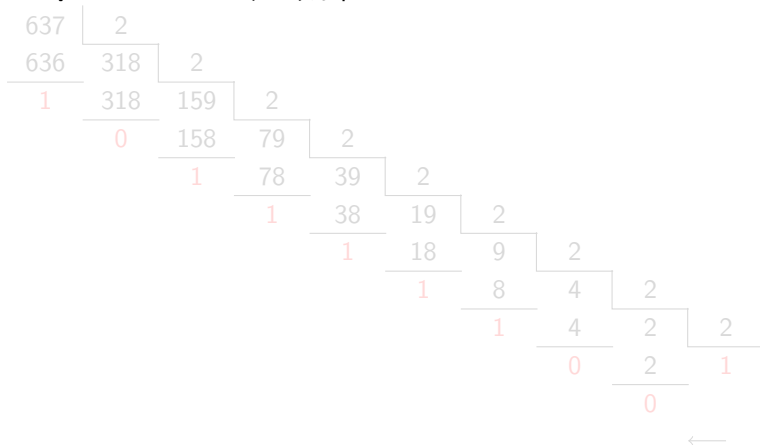
$$\begin{array}{r}
 25 \overline{) 2} \\
 \underline{24} \phantom{0} \\
 1 \phantom{0} \\
 12 \overline{) 12} \\
 \underline{12} \phantom{0} \\
 0 \phantom{0} \\
 6 \overline{) 6} \\
 \underline{6} \phantom{0} \\
 0 \phantom{0} \\
 3 \overline{) 3} \\
 \underline{2} \phantom{0} \\
 1 \phantom{0} \\
 2 \overline{) 2} \\
 \underline{2} \phantom{0} \\
 0 \phantom{0} \\
 1 \phantom{0}
 \end{array}$$

←

$$25 = 11001_2$$

# Conversão de bases

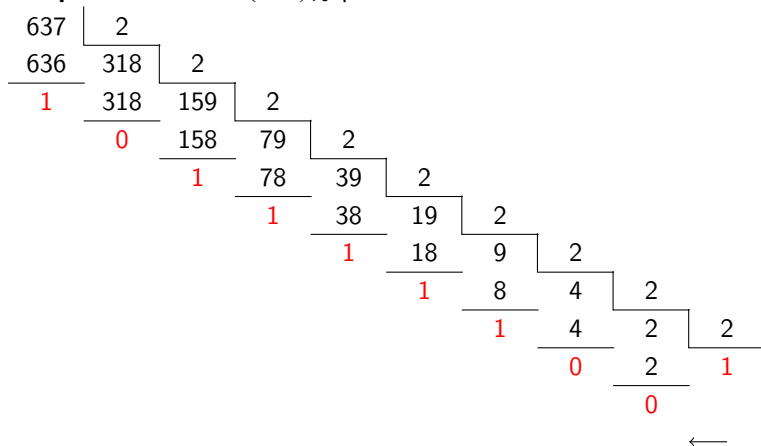
**Exemplo 2:** Converta  $(637)_{10}$  para a base 2.



$$637 = 1001111101_2$$

# Conversão de bases

**Exemplo 2:** Converta  $(637)_{10}$  para a base 2.



$$637 = 1001111101_2$$

# Representação de Números Reais no Computador

Um número real positivo  $N$  na base  $\beta$  pode ser escrito como

$$\begin{aligned}
 N &= (\underbrace{a_n a_{n-1} \dots a_1 a_0}_{N_{\text{int}}}, \underbrace{b_1 b_2 b_3 \dots}_{N_{\text{frac}}})_{\beta} \\
 &= a_n \times \beta^n + a_{n-1} \times \beta^{n-1} + \dots + a_1 \times \beta^1 + a_0 \times \beta^0 \\
 &\quad + b_1 \times \beta^{-1} + b_2 \times \beta^{-2} + b_3 \times \beta^{-3} \dots \\
 &= \underbrace{\sum_{i=0}^n a_i \times \beta^i}_{N_{\text{int}}} + \underbrace{\sum_{i=1}^{\infty} b_i \times \beta^{-i}}_{N_{\text{frac}}},
 \end{aligned}$$

onde  $N_{\text{int}}$  é a parte inteira de  $N$  e  $N_{\text{frac}}$  é a parte fracionária de  $N$ .

**Exemplo 1:**

$$(123,45)_{10} = 1 \times 10^2 + 2 \times 10^1 + 3 \times 10^0 + 4 \times 10^{-1} + 5 \times 10^{-2}.$$

**Exemplo 2:**

$$(101,101)_2 = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} = (5,625)_{10}$$

# Representação de Números Reais no Computador

Um número real positivo  $N$  na base  $\beta$  pode ser escrito como

$$\begin{aligned}
 N &= (\underbrace{a_n a_{n-1} \dots a_1 a_0}_{N_{\text{int}}}, \underbrace{b_1 b_2 b_3 \dots}_{N_{\text{frac}}})_{\beta} \\
 &= a_n \times \beta^n + a_{n-1} \times \beta^{n-1} + \dots + a_1 \times \beta^1 + a_0 \times \beta^0 \\
 &\quad + b_1 \times \beta^{-1} + b_2 \times \beta^{-2} + b_3 \times \beta^{-3} \dots \\
 &= \underbrace{\sum_{i=0}^n a_i \times \beta^i}_{N_{\text{int}}} + \underbrace{\sum_{i=1}^{\infty} b_i \times \beta^{-i}}_{N_{\text{frac}}},
 \end{aligned}$$

onde  $N_{\text{int}}$  é a parte inteira de  $N$  e  $N_{\text{frac}}$  é a parte fracionária de  $N$ .

## Exemplo 1:

$$(123,45)_{10} = 1 \times 10^2 + 2 \times 10^1 + 3 \times 10^0 + 4 \times 10^{-1} + 5 \times 10^{-2}.$$

## Exemplo 2:

$$(101,101)_2 = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} = (5,625)_{10}$$

# Representação de Números Reais no Computador

Um número real positivo  $N$  na base  $\beta$  pode ser escrito como

$$\begin{aligned}
 N &= (\underbrace{a_n a_{n-1} \dots a_1 a_0}_{N_{\text{int}}}, \underbrace{b_1 b_2 b_3 \dots}_{N_{\text{frac}}})_{\beta} \\
 &= a_n \times \beta^n + a_{n-1} \times \beta^{n-1} + \dots + a_1 \times \beta^1 + a_0 \times \beta^0 \\
 &\quad + b_1 \times \beta^{-1} + b_2 \times \beta^{-2} + b_3 \times \beta^{-3} \dots \\
 &= \underbrace{\sum_{i=0}^n a_i \times \beta^i}_{N_{\text{int}}} + \underbrace{\sum_{i=1}^{\infty} b_i \times \beta^{-i}}_{N_{\text{frac}}},
 \end{aligned}$$

onde  $N_{\text{int}}$  é a parte inteira de  $N$  e  $N_{\text{frac}}$  é a parte fracionária de  $N$ .

## Exemplo 1:

$$(123,45)_{10} = 1 \times 10^2 + 2 \times 10^1 + 3 \times 10^0 + 4 \times 10^{-1} + 5 \times 10^{-2}.$$

## Exemplo 2:

$$(101,101)_2 = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} = (5,625)_{10}.$$

# Representação de Números Reais no Computador

- Conversão de base binária para decimal
  - ▶ Similar ao caso inteiro
- Conversão de base decimal para binária
  - ▶ Converte-se a parte inteira
    - ★ Divisões sucessivas
  - ▶ Converte-se a parte fracionária
    - ★ Multiplicações sucessivas

**Exemplo:** Converta  $(12,625)_{10}$  para a base binária

Parte inteira:  $(12)_{10} \Rightarrow$  Divisões sucessivas

Parte fracionária:  $(0,625)_{10} \Rightarrow$  Multiplicações sucessivas



# Representação de Números Reais no Computador

- Conversão de base binária para decimal
  - ▶ Similar ao caso inteiro
- Conversão de base decimal para binária
  - ▶ Converte-se a parte inteira
    - ★ Divisões sucessivas
  - ▶ Converte-se a parte fracionária
    - ★ Multiplicações sucessivas

**Exemplo:** Converta  $(12,625)_{10}$  para a base binária

Parte inteira:  $(12)_{10} \Rightarrow$  Divisões sucessivas

Parte fracionária:  $(0,625)_{10} \Rightarrow$  Multiplicações sucessivas

# Representação de Números Reais no Computador

- Conversão de base binária para decimal
  - ▶ Similar ao caso inteiro
- Conversão de base decimal para binária
  - ▶ Converte-se a parte inteira
    - ★ Divisões sucessivas
  - ▶ Converte-se a parte fracionária
    - ★ Multiplicações sucessivas

**Exemplo:** Converta  $(12,625)_{10}$  para a base binária

Parte inteira:  $(12)_{10} \Rightarrow$  Divisões sucessivas

Parte fracionária:  $(0,625)_{10} \Rightarrow$  Multiplicações sucessivas

# Representação de Números Reais no Computador

- Conversão de base binária para decimal
  - ▶ Similar ao caso inteiro
- Conversão de base decimal para binária
  - ▶ Converte-se a parte inteira
    - ★ Divisões sucessivas
  - ▶ Converte-se a parte fracionária
    - ★ Multiplicações sucessivas

**Exemplo:** Converta  $(12,625)_{10}$  para a base binária

Parte inteira:  $(12)_{10} \Rightarrow$  Divisões sucessivas

Parte fracionária:  $(0,625)_{10} \Rightarrow$  Multiplicações sucessivas

# Representação de Números Reais no Computador

Parte inteira:  $(12)_{10} \Rightarrow$  Divisões sucessivas

$$\begin{array}{r}
 12 \overline{) 2} \\
 \underline{12} \phantom{0} \\
 0 \phantom{0} \phantom{0} \phantom{0} \phantom{0} \\
 6 \overline{) 2} \\
 \underline{6} \phantom{0} \phantom{0} \phantom{0} \\
 0 \phantom{0} \phantom{0} \phantom{0} \\
 3 \overline{) 2} \\
 \underline{3} \phantom{0} \phantom{0} \\
 2 \phantom{0} \\
 1 \overline{) 2} \\
 \underline{2} \\
 0
 \end{array}$$

←

$$12 = 1100_2$$

Parte fracionária:  $(0,625)_{10} \Rightarrow$  Multiplicações sucessivas

# Representação de Números Reais no Computador

Parte inteira:  $(12)_{10} \Rightarrow$  Divisões sucessivas

$$\begin{array}{r}
 12 \overline{) 2} \\
 \underline{12} \phantom{0} \\
 0 \phantom{0} \phantom{0} \phantom{0} \phantom{0} \\
 6 \overline{) 2} \\
 \underline{6} \phantom{0} \phantom{0} \phantom{0} \\
 0 \phantom{0} \phantom{0} \phantom{0} \\
 3 \overline{) 2} \\
 \underline{3} \phantom{0} \phantom{0} \\
 0 \phantom{0} \phantom{0} \\
 2 \overline{) 1} \\
 \underline{2} \\
 0
 \end{array}$$

←

$$12 = 1100_2$$

Parte fracionária:  $(0,625)_{10} \Rightarrow$  Multiplicações sucessivas

# Representação de Números Reais no Computador

Parte fracionária:  $(0,625)_{10} \Rightarrow$  Multiplicações sucessivas

$0,625 \times 2 = 1,35$  (Parte inteira: 1  $\Rightarrow$  Subtraio 1 de 1,35 e multiplico por 2)

$0,35 \times 2 = 0,70$  (Parte inteira: 0  $\Rightarrow$  Apenas multiplico 0,70 por 2)

$0,70 \times 2 = 1,40$  (Parte inteira: 1  $\Rightarrow$  Subtraio 1 de 1,40 e multiplico por 2)

$0,40 \times 2 = 0,80$  (Parte inteira: 0  $\Rightarrow$  Apenas multiplico 0,80 por 2)

$0,80 \times 2 = 1,60$

$0,60 \times 2 = 1,20$

$0,20 \times 2 = 0,40$

$0,40 \times 2 = 0,80$

$0,80 \times 2 = 1,60$

$0,60 \times 2 = 1,20$

$0,20 \times 2 = 0,40$

$\dots \Rightarrow (0,625)_{10} = (0,101011001100110\dots)_2 = (0,1010110)_2$

Logo,  $(12,625)_{10} = (1100,101011001100110\dots)_2 =$

# Representação de Números Reais no Computador

Parte fracionária:  $(0,625)_{10} \Rightarrow$  Multiplicações sucessivas

$0,625 \times 2 = 1,35$  (Parte inteira: 1  $\Rightarrow$  Subtraio 1 de 1,35 e multiplico por 2)

$0,35 \times 2 = 0,70$  (Parte inteira: 0  $\Rightarrow$  Apenas multiplico 0,70 por 2)

$0,70 \times 2 = 1,40$  (Parte inteira: 1  $\Rightarrow$  Subtraio 1 de 1,40 e multiplico por 2)

$0,40 \times 2 = 0,80$  (Parte inteira: 0  $\Rightarrow$  Apenas multiplico 0,80 por 2)

$0,80 \times 2 = 1,60$

$0,60 \times 2 = 1,20$

$0,20 \times 2 = 0,40$

$0,40 \times 2 = 0,80$

$0,80 \times 2 = 1,60$

$0,60 \times 2 = 1,20$

$0,20 \times 2 = 0,40$

$\dots \Rightarrow (0,625)_{10} = (0,101011001100110\dots)_2 = (0,1010110)_2$

Logo,  $(12,625)_{10} = (1100,101011001100110\dots)_2 =$

# Representação de Números Reais no Computador

Parte fracionária:  $(0,625)_{10} \Rightarrow$  Multiplicações sucessivas

$0,625 \times 2 = 1,35$  (Parte inteira: 1  $\Rightarrow$  Subtraio 1 de 1,35 e multiplico por 2)

$0,35 \times 2 = 0,70$  (Parte inteira: 0  $\Rightarrow$  Apenas multiplico 0,70 por 2)

$0,70 \times 2 = 1,40$  (Parte inteira: 1  $\Rightarrow$  Subtraio 1 de 1,40 e multiplico por 2)

$0,40 \times 2 = 0,80$  (Parte inteira: 0  $\Rightarrow$  Apenas multiplico 0,80 por 2)

$0,80 \times 2 = 1,60$

$0,60 \times 2 = 1,20$

$0,20 \times 2 = 0,40$

$0,40 \times 2 = 0,80$

$0,80 \times 2 = 1,60$

$0,60 \times 2 = 1,20$

$0,20 \times 2 = 0,40$

$\dots \Rightarrow (0,625)_{10} = (0,101011001100110\dots)_2 = (0,1010110)_2$

Logo,  $(12,625)_{10} = (1100,101011001100110\dots)_2 =$



# Representação de Números Reais no Computador

Parte fracionária:  $(0,625)_{10} \Rightarrow$  Multiplicações sucessivas

$0,625 \times 2 = 1,35$  (Parte inteira: 1  $\Rightarrow$  Subtraio 1 de 1,35 e multiplico por 2)

$0,35 \times 2 = 0,70$  (Parte inteira: 0  $\Rightarrow$  Apenas multiplico 0,70 por 2)

$0,70 \times 2 = 1,40$  (Parte inteira: 1  $\Rightarrow$  Subtraio 1 de 1,40 e multiplico por 2)

$0,40 \times 2 = 0,80$  (Parte inteira: 0  $\Rightarrow$  Apenas multiplico 0,80 por 2)

$0,80 \times 2 = 1,60$

$0,60 \times 2 = 1,20$

$0,20 \times 2 = 0,40$

$0,40 \times 2 = 0,80$

$0,80 \times 2 = 1,60$

$0,60 \times 2 = 1,20$

$0,20 \times 2 = 0,40$

$\dots \Rightarrow (0,625)_{10} = (0,101011001100110\dots)_2 = (0,1010110)_2$

Logo,  $(12,625)_{10} = (1100,101011001100110\dots)_2 =$

# Representação de Números Reais no Computador

Parte fracionária:  $(0,625)_{10} \Rightarrow$  Multiplicações sucessivas

$0,625 \times 2 = 1,35$  (Parte inteira: 1  $\Rightarrow$  Subtraio 1 de 1,35 e multiplico por 2)

$0,35 \times 2 = 0,70$  (Parte inteira: 0  $\Rightarrow$  Apenas multiplico 0,70 por 2)

$0,70 \times 2 = 1,40$  (Parte inteira: 1  $\Rightarrow$  Subtraio 1 de 1,40 e multiplico por 2)

$0,40 \times 2 = 0,80$  (Parte inteira: 0  $\Rightarrow$  Apenas multiplico 0,80 por 2)

$0,80 \times 2 = 1,60$

$0,60 \times 2 = 1,20$

$0,20 \times 2 = 0,40$

$0,40 \times 2 = 0,80$

$0,80 \times 2 = 1,60$

$0,60 \times 2 = 1,20$

$0,20 \times 2 = 0,40$

$\dots \Rightarrow (0,625)_{10} = (0,101011001100110\dots)_2 = (0,1010110)_2$

Logo,  $(12,625)_{10} = (1100,101011001100110\dots)_2 =$

# Representação de Números Reais no Computador

Parte fracionária:  $(0,625)_{10} \Rightarrow$  Multiplicações sucessivas

$0,625 \times 2 = 1,35$  (Parte inteira: 1  $\Rightarrow$  Subtraio 1 de 1,35 e multiplico por 2)

$0,35 \times 2 = 0,70$  (Parte inteira: 0  $\Rightarrow$  Apenas multiplico 0,70 por 2)

$0,70 \times 2 = 1,40$  (Parte inteira: 1  $\Rightarrow$  Subtraio 1 de 1,40 e multiplico por 2)

$0,40 \times 2 = 0,80$  (Parte inteira: 0  $\Rightarrow$  Apenas multiplico 0,80 por 2)

$0,80 \times 2 = 1,60$

$0,60 \times 2 = 1,20$

$0,20 \times 2 = 0,40$

$0,40 \times 2 = 0,80$

$0,80 \times 2 = 1,60$

$0,60 \times 2 = 1,20$

$0,20 \times 2 = 0,40$

$\dots \Rightarrow (0,625)_{10} = (0,101011001100110\dots)_2 = (0,1010110)_2$

Logo,  $(12,635)_{10} = (1100,101011001100110\dots)_2 =$

# Representação de Números Reais no Computador

Parte fracionária:  $(0,625)_{10} \Rightarrow$  Multiplicações sucessivas

$0,625 \times 2 = 1,35$  (Parte inteira: 1  $\Rightarrow$  Subtraio 1 de 1,35 e multiplico por 2)

$0,35 \times 2 = 0,70$  (Parte inteira: 0  $\Rightarrow$  Apenas multiplico 0,70 por 2)

$0,70 \times 2 = 1,40$  (Parte inteira: 1  $\Rightarrow$  Subtraio 1 de 1,40 e multiplico por 2)

$0,40 \times 2 = 0,80$  (Parte inteira: 0  $\Rightarrow$  Apenas multiplico 0,80 por 2)

$0,80 \times 2 = 1,60$

$0,60 \times 2 = 1,20$

$0,20 \times 2 = 0,40$

$0,40 \times 2 = 0,80$

$0,80 \times 2 = 1,60$

$0,60 \times 2 = 1,20$

$0,20 \times 2 = 0,40$

$\dots \Rightarrow (0,625)_{10} = (0,101011001100110\dots)_2 = (0,1010110)_2$

Logo,  $(12,625)_{10} = (1100,101011001100110\dots)_2 =$

# Representação de Números Reais no Computador

Parte fracionária:  $(0,625)_{10} \Rightarrow$  Multiplicações sucessivas

$0,625 \times 2 = 1,35$  (Parte inteira: 1  $\Rightarrow$  Subtraio 1 de 1,35 e multiplico por 2)

$0,35 \times 2 = 0,70$  (Parte inteira: 0  $\Rightarrow$  Apenas multiplico 0,70 por 2)

$0,70 \times 2 = 1,40$  (Parte inteira: 1  $\Rightarrow$  Subtraio 1 de 1,40 e multiplico por 2)

$0,40 \times 2 = 0,80$  (Parte inteira: 0  $\Rightarrow$  Apenas multiplico 0,80 por 2)

$0,80 \times 2 = 1,60$

$0,60 \times 2 = 1,20$

$0,20 \times 2 = 0,40$

$0,40 \times 2 = 0,80$

$0,80 \times 2 = 1,60$

$0,60 \times 2 = 1,20$

$0,20 \times 2 = 0,40$

$\dots \Rightarrow (0,625)_{10} = (0,101011001100110\dots)_2 = (0,1010110)_2$

Logo,  $(12,625)_{10} = (1100,101011001100110\dots)_2 =$

# Representação de Números Reais no Computador

Parte fracionária:  $(0,625)_{10} \Rightarrow$  Multiplicações sucessivas

$0,625 \times 2 = 1,35$  (Parte inteira: 1  $\Rightarrow$  Subtraio 1 de 1,35 e multiplico por 2)

$0,35 \times 2 = 0,70$  (Parte inteira: 0  $\Rightarrow$  Apenas multiplico 0,70 por 2)

$0,70 \times 2 = 1,40$  (Parte inteira: 1  $\Rightarrow$  Subtraio 1 de 1,40 e multiplico por 2)

$0,40 \times 2 = 0,80$  (Parte inteira: 0  $\Rightarrow$  Apenas multiplico 0,80 por 2)

$0,80 \times 2 = 1,60$

$0,60 \times 2 = 1,20$

$0,20 \times 2 = 0,40$

$0,40 \times 2 = 0,80$

$0,80 \times 2 = 1,60$

$0,60 \times 2 = 1,20$

$0,20 \times 2 = 0,40$

$\dots \Rightarrow (0,625)_{10} = (0,101011001100110\dots)_2 = (0,1010110)_2$

Logo,  $(12,635)_{10} = (1100,101011001100110\dots)_2 =$

# Representação de Números Reais no Computador

Parte fracionária:  $(0,625)_{10} \Rightarrow$  Multiplicações sucessivas

$0,625 \times 2 = 1,35$  (Parte inteira: 1  $\Rightarrow$  Subtraio 1 de 1,35 e multiplico por 2)

$0,35 \times 2 = 0,70$  (Parte inteira: 0  $\Rightarrow$  Apenas multiplico 0,70 por 2)

$0,70 \times 2 = 1,40$  (Parte inteira: 1  $\Rightarrow$  Subtraio 1 de 1,40 e multiplico por 2)

$0,40 \times 2 = 0,80$  (Parte inteira: 0  $\Rightarrow$  Apenas multiplico 0,80 por 2)

$0,80 \times 2 = 1,60$

$0,60 \times 2 = 1,20$

$0,20 \times 2 = 0,40$

$0,40 \times 2 = 0,80$

$0,80 \times 2 = 1,60$

$0,60 \times 2 = 1,20$

$0,20 \times 2 = 0,40$

$\dots \Rightarrow (0,625)_{10} = (0,101011001100110\dots)_2 = (0,1010110)_2$

Logo,  $(12,625)_{10} = (1100,101011001100110\dots)_2 =$

# Representação de Números Reais no Computador

Parte fracionária:  $(0,625)_{10} \Rightarrow$  Multiplicações sucessivas

$0,625 \times 2 = 1,35$  (Parte inteira: 1  $\Rightarrow$  Subtraio 1 de 1,35 e multiplico por 2)

$0,35 \times 2 = 0,70$  (Parte inteira: 0  $\Rightarrow$  Apenas multiplico 0,70 por 2)

$0,70 \times 2 = 1,40$  (Parte inteira: 1  $\Rightarrow$  Subtraio 1 de 1,40 e multiplico por 2)

$0,40 \times 2 = 0,80$  (Parte inteira: 0  $\Rightarrow$  Apenas multiplico 0,80 por 2)

$0,80 \times 2 = 1,60$

$0,60 \times 2 = 1,20$

$0,20 \times 2 = 0,40$

$0,40 \times 2 = 0,80$

$0,80 \times 2 = 1,60$

$0,60 \times 2 = 1,20$

$0,20 \times 2 = 0,40$

$\dots \Rightarrow (0,625)_{10} = (0,101011001100110\dots)_2 = (0,1010\overline{110})_2$

Logo,  $(12,625)_{10} = (1100,101011001100110\dots)_2 =$



# Representação de Números Reais no Computador

Parte fracionária:  $(0,625)_{10} \Rightarrow$  Multiplicações sucessivas

$0,625 \times 2 = 1,35$  (Parte inteira: 1  $\Rightarrow$  Subtraio 1 de 1,35 e multiplico por 2)

$0,35 \times 2 = 0,70$  (Parte inteira: 0  $\Rightarrow$  Apenas multiplico 0,70 por 2)

$0,70 \times 2 = 1,40$  (Parte inteira: 1  $\Rightarrow$  Subtraio 1 de 1,40 e multiplico por 2)

$0,40 \times 2 = 0,80$  (Parte inteira: 0  $\Rightarrow$  Apenas multiplico 0,80 por 2)

$0,80 \times 2 = 1,60$

$0,60 \times 2 = 1,20$

$0,20 \times 2 = 0,40$

$0,40 \times 2 = 0,80$

$0,80 \times 2 = 1,60$

$0,60 \times 2 = 1,20$

$0,20 \times 2 = 0,40$

$\dots \Rightarrow (0,625)_{10} = (0,101011001100110\dots)_2 = (0,1010\overline{110})_2$

Logo,  $(12,625)_{10} = (1100,101011001100110\dots)_2 = (1100,1010\overline{110})_2$

# Representação de Números Reais no Computador

Parte fracionária:  $(0,625)_{10} \Rightarrow$  Multiplicações sucessivas

$0,625 \times 2 = 1,35$  (Parte inteira: 1  $\Rightarrow$  Subtraio 1 de 1,35 e multiplico por 2)

$0,35 \times 2 = 0,70$  (Parte inteira: 0  $\Rightarrow$  Apenas multiplico 0,70 por 2)

$0,70 \times 2 = 1,40$  (Parte inteira: 1  $\Rightarrow$  Subtraio 1 de 1,40 e multiplico por 2)

$0,40 \times 2 = 0,80$  (Parte inteira: 0  $\Rightarrow$  Apenas multiplico 0,80 por 2)

$0,80 \times 2 = 1,60$

$0,60 \times 2 = 1,20$

$0,20 \times 2 = 0,40$

$0,40 \times 2 = 0,80$

$0,80 \times 2 = 1,60$

$0,60 \times 2 = 1,20$

$0,20 \times 2 = 0,40$

$\dots \Rightarrow (0,625)_{10} = (0,101011001100110\dots)_2 = (0,1010\overline{110})_2$

Logo,  $(12,625)_{10} = (1100,101011001100110\dots)_2 = (1100,1010\overline{110})_2$

# Representação de Números Reais no Computador

Parte fracionária:  $(0,625)_{10} \Rightarrow$  Multiplicações sucessivas

$0,625 \times 2 = 1,35$  (Parte inteira: 1  $\Rightarrow$  Subtraio 1 de 1,35 e multiplico por 2)

$0,35 \times 2 = 0,70$  (Parte inteira: 0  $\Rightarrow$  Apenas multiplico 0,70 por 2)

$0,70 \times 2 = 1,40$  (Parte inteira: 1  $\Rightarrow$  Subtraio 1 de 1,40 e multiplico por 2)

$0,40 \times 2 = 0,80$  (Parte inteira: 0  $\Rightarrow$  Apenas multiplico 0,80 por 2)

$0,80 \times 2 = 1,60$

$0,60 \times 2 = 1,20$

$0,20 \times 2 = 0,40$

$0,40 \times 2 = 0,80$

$0,80 \times 2 = 1,60$

$0,60 \times 2 = 1,20$

$0,20 \times 2 = 0,40$

$\dots \Rightarrow (0,625)_{10} = (0,101011001100110\dots)_2 = (0,1010\overline{110})_2$

Logo,  $(12,625)_{10} = (1100,101011001100110\dots)_2 = (1100,1010\overline{110})_2$

# Representação de Números Reais no Computador

Parte fracionária:  $(0,625)_{10} \Rightarrow$  Multiplicações sucessivas

$0,625 \times 2 = 1,35$  (Parte inteira: 1  $\Rightarrow$  Subtraio 1 de 1,35 e multiplico por 2)

$0,35 \times 2 = 0,70$  (Parte inteira: 0  $\Rightarrow$  Apenas multiplico 0,70 por 2)

$0,70 \times 2 = 1,40$  (Parte inteira: 1  $\Rightarrow$  Subtraio 1 de 1,40 e multiplico por 2)

$0,40 \times 2 = 0,80$  (Parte inteira: 0  $\Rightarrow$  Apenas multiplico 0,80 por 2)

$0,80 \times 2 = 1,60$

$0,60 \times 2 = 1,20$

$0,20 \times 2 = 0,40$

$0,40 \times 2 = 0,80$

$0,80 \times 2 = 1,60$

$0,60 \times 2 = 1,20$

$0,20 \times 2 = 0,40$

$\dots \Rightarrow (0,625)_{10} = (0,101011001100110\dots)_2 = (0,101\overline{0110})_2$

Logo,  $(12,625)_{10} = (1100,101011001100110\dots)_2 = (1100,101\overline{0110})_2$

# Ponto fixo e ponto flutuante

Em todos esses exemplos, a posição da vírgula está fixa, separando a casa das unidades da primeira casa fracionária.

Entretanto, pode-se variar a posição da vírgula, corrigindo-se o valor com a potência da base, seja dez ou dois, dependendo do sistema que se use.

## Exemplo 1 (Base decimal):

$$(45,31)_{10} = 4,531 \times 10^1 = 0,4531 \times 10^2 = 453,1 \times 10^{-1}$$

## Exemplo 2 (Base binária):

$$(1110,01)_2 = 1,11001 \times 2^3 = 0,111001 \times 2^4 = 11100,1 \times 2^{-1}$$

Chama-se a isso **ponto flutuante (floating point)**, pois no lugar de se deixar sempre a posição da vírgula entre a casa das unidades e a primeira casa decimal, flutua-se a posição da vírgula e corrige-se com a potência da base.

## Ponto fixo e ponto flutuante

Em todos esses exemplos, a posição da vírgula está fixa, separando a casa das unidades da primeira casa fracionária.

Entretanto, pode-se variar a posição da vírgula, corrigindo-se o valor com a potência da base, seja dez ou dois, dependendo do sistema que se use.

### Exemplo 1 (Base decimal):

$$(45,31)_{10} = 4,531 \times 10^1 = 0,4531 \times 10^2 = 453,1 \times 10^{-1}$$

### Exemplo 2 (Base binária):

$$(1110,01)_2 = 1,11001 \times 2^3 = 0,111001 \times 2^4 = 11100,1 \times 2^{-1}$$

Chama-se a isso **ponto flutuante (floating point)**, pois no lugar de se deixar sempre a posição da vírgula entre a casa das unidades e a primeira casa decimal, flutua-se a posição da vírgula e corrige-se com a potência da base.

## Ponto fixo e ponto flutuante

Em todos esses exemplos, a posição da vírgula está fixa, separando a casa das unidades da primeira casa fracionária.

Entretanto, pode-se variar a posição da vírgula, corrigindo-se o valor com a potência da base, seja dez ou dois, dependendo do sistema que se use.

### Exemplo 1 (Base decimal):

$$(45,31)_{10} = 4,531 \times 10^1 = 0,4531 \times 10^2 = 453,1 \times 10^{-1}$$

### Exemplo 2 (Base binária):

$$(1110,01)_2 = 1,11001 \times 2^3 = 0,111001 \times 2^4 = 11100,1 \times 2^{-1}$$

Chama-se a isso **ponto flutuante (floating point)**, pois no lugar de se deixar sempre a posição da vírgula entre a casa das unidades e a primeira casa decimal, flutua-se a posição da vírgula e corrige-se com a potência da base.

## Ponto fixo e ponto flutuante

Em todos esses exemplos, a posição da vírgula está fixa, separando a casa das unidades da primeira casa fracionária.

Entretanto, pode-se variar a posição da vírgula, corrigindo-se o valor com a potência da base, seja dez ou dois, dependendo do sistema que se use.

### Exemplo 1 (Base decimal):

$$(45,31)_{10} = 4,531 \times 10^1 = 0,4531 \times 10^2 = 453,1 \times 10^{-1}$$

### Exemplo 2 (Base binária):

$$(1110,01)_2 = 1,11001 \times 2^3 = 0,111001 \times 2^4 = 11100,1 \times 2^{-1}$$

Chama-se a isso **ponto flutuante (floating point)**, pois no lugar de se deixar sempre a posição da vírgula entre a casa das unidades e a primeira casa decimal, flutua-se a posição da vírgula e corrige-se com a potência da base.



## Ponto fixo e ponto flutuante

Em todos esses exemplos, a posição da vírgula está fixa, separando a casa das unidades da primeira casa fracionária.

Entretanto, pode-se variar a posição da vírgula, corrigindo-se o valor com a potência da base, seja dez ou dois, dependendo do sistema que se use.

### Exemplo 1 (Base decimal):

$$(45,31)_{10} = 4,531 \times 10^1 = 0,4531 \times 10^2 = 453,1 \times 10^{-1}$$

### Exemplo 2 (Base binária):

$$(1110,01)_2 = 1,11001 \times 2^3 = 0,111001 \times 2^4 = 11100,1 \times 2^{-1}$$

Chama-se a isso **ponto flutuante (floating point)**, pois no lugar de se deixar sempre a posição da vírgula entre a casa das unidades e a primeira casa decimal, flutua-se a posição da vírgula e corrige-se com a potência da base.

## Ponto fixo e ponto flutuante

Em todos esses exemplos, a posição da vírgula está fixa, separando a casa das unidades da primeira casa fracionária.

Entretanto, pode-se variar a posição da vírgula, corrigindo-se o valor com a potência da base, seja dez ou dois, dependendo do sistema que se use.

### Exemplo 1 (Base decimal):

$$(45,31)_{10} = 4,531 \times 10^1 = 0,4531 \times 10^2 = 453,1 \times 10^{-1}$$

### Exemplo 2 (Base binária):

$$(1110,01)_2 = 1,11001 \times 2^3 = 0,111001 \times 2^4 = 11100,1 \times 2^{-1}$$

Chama-se a isso **ponto flutuante (floating point)**, pois no lugar de se deixar sempre a posição da vírgula entre a casa das unidades e a primeira casa decimal, flutua-se a posição da vírgula e corrige-se com a potência da base.

## Ponto fixo e ponto flutuante

Em todos esses exemplos, a posição da vírgula está fixa, separando a casa das unidades da primeira casa fracionária.

Entretanto, pode-se variar a posição da vírgula, corrigindo-se o valor com a potência da base, seja dez ou dois, dependendo do sistema que se use.

### Exemplo 1 (Base decimal):

$$(45,31)_{10} = 4,531 \times 10^1 = 0,4531 \times 10^2 = 453,1 \times 10^{-1}$$

### Exemplo 2 (Base binária):

$$(1110,01)_2 = 1,11001 \times 2^3 = 0,111001 \times 2^4 = 11100,1 \times 2^{-1}$$

Chama-se a isso **ponto flutuante (floating point)**, pois no lugar de se deixar sempre a posição da vírgula entre a casa das unidades e a primeira casa decimal, flutua-se a posição da vírgula e corrige-se com a potência da base.

## Ponto fixo e ponto flutuante

Em todos esses exemplos, a posição da vírgula está fixa, separando a casa das unidades da primeira casa fracionária.

Entretanto, pode-se variar a posição da vírgula, corrigindo-se o valor com a potência da base, seja dez ou dois, dependendo do sistema que se use.

### Exemplo 1 (Base decimal):

$$(45,31)_{10} = 4,531 \times 10^1 = 0,4531 \times 10^2 = 453,1 \times 10^{-1}$$

### Exemplo 2 (Base binária):

$$(1110,01)_2 = 1,11001 \times 2^3 = 0,111001 \times 2^4 = 11100,1 \times 2^{-1}$$

Chama-se a isso **ponto flutuante (floating point)**, pois no lugar de se deixar sempre a posição da vírgula entre a casa das unidades e a primeira casa decimal, flutua-se a posição da vírgula e corrige-se com a potência da base.

## Ponto fixo e ponto flutuante

Em todos esses exemplos, a posição da vírgula está fixa, separando a casa das unidades da primeira casa fracionária.

Entretanto, pode-se variar a posição da vírgula, corrigindo-se o valor com a potência da base, seja dez ou dois, dependendo do sistema que se use.

### Exemplo 1 (Base decimal):

$$(45,31)_{10} = 4,531 \times 10^1 = 0,4531 \times 10^2 = 453,1 \times 10^{-1}$$

### Exemplo 2 (Base binária):

$$(1110,01)_2 = 1,11001 \times 2^3 = 0,111001 \times 2^4 = 11100,1 \times 2^{-1}$$

Chama-se a isso **ponto flutuante (floating point)**, pois no lugar de se deixar sempre a posição da vírgula entre a casa das unidades e a primeira casa decimal, flutua-se a posição da vírgula e corrige-se com a potência da base.

# Forma normalizada

Como se vê, há diferentes maneiras de escrever o mesmo número.

Chama-se forma normalizada aquela que apresenta um único dígito diferente de zero antes da vírgula.

Exemplo 1 na forma normalizada:  $(45,31)_{10} = 4,531 \times 10^1$

Exemplo 2 na forma normalizada:  $(1110,01)_2 = 1,11001 \times 2^3$

Outros exemplos:

$$(110101)_2 = 1,10101 \times 2^5 = (53)_{10}.$$

$$(0,0011)_2 = 1,1 \times 2^{-3} = (0,1875)_{10}.$$

$$(0,1001)_2 = 1,001 \times 2^{-1} = (0,5625)_{10}.$$

# Forma normalizada

Como se vê, há diferentes maneiras de escrever o mesmo número.

Chama-se **forma normalizada** aquela que apresenta **um único dígito diferente de zero** antes da vírgula.

Exemplo 1 na forma normalizada:  $(45,31)_{10} = 4,531 \times 10^1$

Exemplo 2 na forma normalizada:  $(1110,01)_2 = 1,11001 \times 2^3$

Outros exemplos:

$$(110101)_2 = 1,10101 \times 2^5 = (53)_{10}.$$

$$(0,0011)_2 = 1,1 \times 2^{-3} = (0,1875)_{10}.$$

$$(0,1001)_2 = 1,001 \times 2^{-1} = (0,5625)_{10}.$$

# Forma normalizada

Como se vê, há diferentes maneiras de escrever o mesmo número.

Chama-se **forma normalizada** aquela que apresenta **um único dígito diferente de zero** antes da vírgula.

**Exemplo 1 na forma normalizada:**  $(45,31)_{10} = 4,531 \times 10^1$

**Exemplo 2 na forma normalizada:**  $(1110,01)_2 = 1,11001 \times 2^3$

**Outros exemplos:**

$$(110101)_2 = 1,10101 \times 2^5 = (53)_{10}.$$

$$(0,0011)_2 = 1,1 \times 2^{-3} = (0,1875)_{10}.$$

$$(0,1001)_2 = 1,001 \times 2^{-1} = (0,5625)_{10}.$$



# Forma normalizada

Como se vê, há diferentes maneiras de escrever o mesmo número.

Chama-se **forma normalizada** aquela que apresenta **um único dígito diferente de zero** antes da vírgula.

**Exemplo 1 na forma normalizada:**  $(45,31)_{10} = 4,531 \times 10^1$

**Exemplo 2 na forma normalizada:**  $(1110,01)_2 = 1,11001 \times 2^3$

Outros exemplos:

$$(110101)_2 = 1,10101 \times 2^5 = (53)_{10}.$$

$$(0,0011)_2 = 1,1 \times 2^{-3} = (0,1875)_{10}.$$

$$(0,1001)_2 = 1,001 \times 2^{-1} = (0,5625)_{10}.$$

## Forma normalizada

Como se vê, há diferentes maneiras de escrever o mesmo número.

Chama-se **forma normalizada** aquela que apresenta **um único dígito diferente de zero** antes da vírgula.

**Exemplo 1 na forma normalizada:**  $(45,31)_{10} = 4,531 \times 10^1$

**Exemplo 2 na forma normalizada:**  $(1110,01)_2 = 1,11001 \times 2^3$

**Outros exemplos:**

$$(110101)_2 = 1,10101 \times 2^5 = (53)_{10}.$$

$$(0,0011)_2 = 1,1 \times 2^{-3} = (0,1875)_{10}.$$

$$(0,1001)_2 = 1,001 \times 2^{-1} = (0,5625)_{10}.$$

# Representação de Números Reais no Computador

O computador representa os números em sistema binário.

- A representação é finita
  - ▶ Números como o  $\pi = 3,1415\dots$  são aproximados.

Vamos usar a representação em ponto flutuante na forma normalizada.

**Exemplo:**  $110101 = 1,10101 \times 2^5$

Chama-se *mantissa* ao número  $1,10101$  e *expoente* ao número  $101$ , que é  $5_2$ , deste exemplo.

Para se definir a maneira como o computador armazenará o número real em ponto flutuante, é preciso definir o número de bits que ele usará para representar a mantissa e o número de bits para o expoente.

# Representação de Números Reais no Computador

O computador representa os números em sistema binário.

- A representação é finita
  - ▶ Números como o  $\pi = 3,1415\dots$  são aproximados.

Vamos usar a representação em ponto flutuante na forma normalizada.

**Exemplo:**  $110101 = 1,10101 \times 2^5$

Chama-se **mantissa** ao número  $1,10101$  e **expoente** ao número  $101$ , que é  $5_2$ , deste exemplo.

Para se definir a maneira como o computador armazenará o número real em ponto flutuante, é preciso definir o número de bits que ele usará para representar a mantissa e o número de bits para o expoente.

# Representação de Números Reais no Computador

O computador representa os números em sistema binário.

- A representação é finita
  - ▶ Números como o  $\pi = 3,1415\dots$  são aproximados.

Vamos usar a representação em ponto flutuante na forma normalizada.

**Exemplo:**  $110101 = 1,10101 \times 2^5$

Chama-se **mantissa** ao número  $1,10101$  e **expoente** ao número  $101$ , que é  $5_2$ , deste exemplo.

Para se definir a maneira como o computador armazenará o número real em ponto flutuante, é preciso definir o número de bits que ele usará para representar a mantissa e o número de bits para o expoente.

# Representação de Números Reais no Computador

O computador representa os números em sistema binário.

- A representação é finita
  - ▶ Números como o  $\pi = 3,1415\dots$  são aproximados.

Vamos usar a representação em ponto flutuante na forma normalizada.

**Exemplo:**  $110101 = 1,10101 \times 2^5$

Chama-se **mantissa** ao número  $1,10101$  e **expoente** ao número  $101$ , que é  $5_2$ , deste exemplo.

Para se definir a maneira como o computador armazenará o número real em ponto flutuante, é preciso definir o número de bits que ele usará para representar a mantissa e o número de bits para o expoente.

# Representação de Números Reais no Computador

O computador representa os números em sistema binário.

- A representação é finita
  - ▶ Números como o  $\pi = 3,1415\dots$  são aproximados.

Vamos usar a representação em ponto flutuante na forma normalizada.

**Exemplo:**  $110101 = 1,10101 \times 2^5$

Chama-se **mantissa** ao número  $1,10101$  e **expoente** ao número  $101$ , que é  $5_2$ , deste exemplo.

Para se definir a maneira como o computador armazenará o número real em ponto flutuante, é preciso definir o número de bits que ele usará para representar a mantissa e o número de bits para o expoente.

# Representação de Números Reais no Computador

O computador representa os números em sistema binário.

- A representação é finita
  - ▶ Números como o  $\pi = 3,1415\dots$  são aproximados.

Vamos usar a representação em ponto flutuante na forma normalizada.

**Exemplo:**  $110101 = 1,10101 \times 2^5$

Chama-se **mantissa** ao número  $1,10101$  e **expoente** ao número  $101$ , que é  $5_2$ , deste exemplo.

Para se definir a maneira como o computador armazenará o número real em ponto flutuante, é preciso definir o número de bits que ele usará para representar a mantissa e o número de bits para o expoente.



# Representação de Números Reais no Computador

Para armazenar a mantissa, é dispensável representar o "1," por estar sempre presente, sendo também desnecessário armazenar o 2, base do sistema.

Suponha-se que um determinado computador reserve **1 byte**, isto é, **8 bits**, para representar os números reais.

Admita-se que usa o primeiro bit para sinal do número, três bits seguintes para o expoente e os últimos quatro bits para o restante da mantissa.



Tabela: Representação em ponto flutuante

# Representação de Números Reais no Computador

Para armazenar a mantissa, é dispensável representar o "1," por estar sempre presente, sendo também desnecessário armazenar o 2, base do sistema.

Suponha-se que um determinado computador reserve **1 byte**, isto é, **8 bits**, para representar os números reais.

Admita-se que usa o primeiro bit para sinal do número, três bits seguintes para o expoente e os últimos quatro bits para o restante da mantissa.



Tabela: Representação em ponto flutuante

# Representação de Números Reais no Computador

Para armazenar a mantissa, é dispensável representar o "1," por estar sempre presente, sendo também desnecessário armazenar o 2, base do sistema.

Suponha-se que um determinado computador reserve **1 byte**, isto é, **8 bits**, para representar os números reais.

Admita-se que usa o primeiro bit para sinal do número, três bits seguintes para o expoente e os últimos quatro bits para o restante da mantissa.



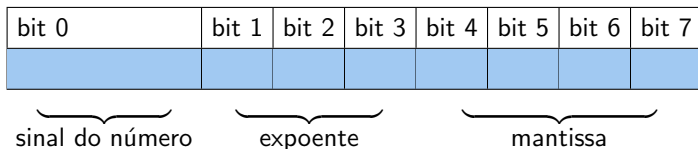
Tabela: Representação em ponto flutuante

# Representação de Números Reais no Computador

Para armazenar a mantissa, é dispensável representar o "1," por estar sempre presente, sendo também desnecessário armazenar o 2, base do sistema.

Suponha-se que um determinado computador reserve **1 byte**, isto é, **8 bits**, para representar os números reais.

Admita-se que usa o primeiro bit para sinal do número, três bits seguintes para o expoente e os últimos quatro bits para o restante da mantissa.



**Tabela:** Representação em ponto flutuante

# Representação de Números Reais no Computador

O bit 0 indica o sinal do número: 0 positivo, 1 negativo.

Os bits 1, 2 e 3 constituem o expoente e precisam representar tanto expoentes positivos quanto expoentes negativos.

bits	expoente	valor em decimal
000	Desnormalizado	
001	-2	1
010	-1	2
011	0	3
100	+1	4
101	+2	5
110	+3	6
111	NaN, Inf., Indet.	

Tabela: Tabela para os bits do expoente

# Representação de Números Reais no Computador

O bit 0 indica o sinal do número: 0 positivo, 1 negativo.

Os bits 1, 2 e 3 constituem o expoente e precisam representar tanto expoentes positivos quanto expoentes negativos.

bits	expoente	valor em decimal
000	Desnormalizado	
001	-2	1
010	-1	2
011	0	3
100	+1	4
101	+2	5
110	+3	6
111	NaN, Inf., Indet.	

Tabela: Tabela para os bits do expoente

# Representação de Números Reais no Computador

O bit 0 indica o sinal do número: 0 positivo, 1 negativo.

Os bits 1, 2 e 3 constituem o expoente e precisam representar tanto expoentes positivos quanto expoentes negativos.

bits	expoente	valor em decimal
000	Desnormalizado	
001	-2	1
010	-1	2
011	0	3
100	+1	4
101	+2	5
110	+3	6
111	NaN, Inf., Indet.	

**Tabela:** Tabela para os bits do expoente

# Representação de Números Reais no Computador

Dessa maneira, o número que representa o expoente será o valor em decimal menos três.

Os bits da mantissa representam os dígitos que aparecem depois da vírgula na forma normalizada.

## Exemplo 1:

$$(3,5)_{10} = (11,1)_2 = 1,11 \times 2^1 = 1,1100 \times 2^1 \Rightarrow$$

bit de sinal: 0; bits do expoente: 100; mantissa: 1100



Tabela: Exemplo 1



# Representação de Números Reais no Computador

Dessa maneira, o número que representa o expoente será o valor em decimal menos três.

Os bits da mantissa representam os dígitos que aparecem depois da vírgula na forma normalizada.

## Exemplo 1:

$$(3,5)_{10} = (11,1)_2 = 1,11 \times 2^1 = 1,1100 \times 2^1 \Rightarrow$$

bit de sinal: 0; bits do expoente: 100; mantissa: 1100



Tabela: Exemplo 1

# Representação de Números Reais no Computador

Dessa maneira, o número que representa o expoente será o valor em decimal menos três.

Os bits da mantissa representam os dígitos que aparecem depois da vírgula na forma normalizada.

## Exemplo 1:

$$(3,5)_{10} = (11,1)_2 = 1,11 \times 2^1 = 1,1100 \times 2^1 \Rightarrow$$

bit de sinal: 0; bits do expoente: 100; mantissa: 1100

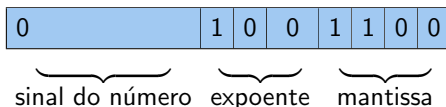


Tabela: Exemplo 1

# Representação de Números Reais no Computador

## Exemplo 2:

$$(-7,25)_{10} = (-111,01)_2 = -1,1101 \times 2^2 \Rightarrow$$

bit de sinal: 1; bits do expoente: 101; mantissa: 1101

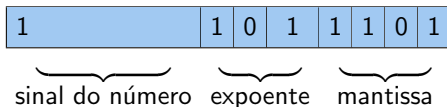


Tabela: Exemplo 2

## Exemplo 3 - Maior número positivo:



Tabela: Exemplo 3

$$1,1111 \times 2^3 = (15,5)_{10}$$

# Representação de Números Reais no Computador

## Exemplo 2:

$$(-7,25)_{10} = (-111,01)_2 = -1,1101 \times 2^2 \Rightarrow$$

bit de sinal: 1; bits do expoente: 101; mantissa: 1101

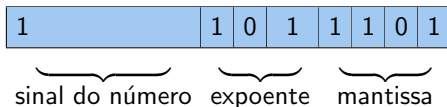


Tabela: Exemplo 2

## Exemplo 3 - Maior número positivo:



Tabela: Exemplo 3

$$1,1111 \times 2^3 = (15,5)_{10}$$

# Representação de Números Reais no Computador

## Exemplo 2:

$$(-7, 25)_{10} = (-111, 01)_2 = -1, 1101 \times 2^2 \Rightarrow$$

bit de sinal: 1; bits do expoente: 101; mantissa: 1101

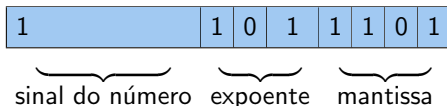


Tabela: Exemplo 2

## Exemplo 3 - Maior número positivo:

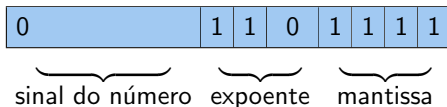


Tabela: Exemplo 3

$$1,1111 \times 2^3 = (15,5)_{10}$$

# Representação de Números Reais no Computador

## Exemplo 2:

$$(-7,25)_{10} = (-111,01)_2 = -1,1101 \times 2^2 \Rightarrow$$

bit de sinal: 1; bits do expoente: 101; mantissa: 1101

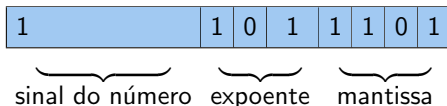


Tabela: Exemplo 2

## Exemplo 3 - Maior número positivo:

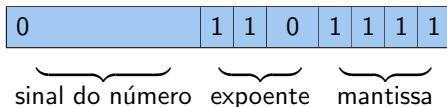


Tabela: Exemplo 3

$$1,1111 \times 2^3 = (15,5)_{10}$$

# Representação de Números Reais no Computador

## Exemplo 4 - Menor número positivo:

Devemos lembrar que os expoentes 000 e 111 possuem tratamento especial.

Usa-se o expoente 000 para indicar que o número não está normalizado.

Assim, 000 será configurado como o menor expoente, isto é,  $-2$  neste exemplo.

Na forma desnormalizada, a mantissa passa a ser  $0, b_1 b_2 b_3 b_4$ , onde  $b_1, \dots, b_4$  são os bits da mantissa neste exemplo.

Assim, o menor número positivo armazenado pelo computador de 8 bits recebe a seguinte configuração:



Tabela: Exemplo 4

$$0,0001 \times 2^{-2} = (0,000001)_2 = 2^{-6} = (0,015625)_{10}$$

# Representação de Números Reais no Computador

## Exemplo 4 - Menor número positivo:

Devemos lembrar que os expoentes 000 e 111 possuem tratamento especial.

Usa-se o expoente 000 para indicar que o número não está normalizado.

Assim, 000 será configurado como o menor expoente, isto é,  $-2$  neste exemplo.

Na forma desnormalizada, a mantissa passa a ser  $0, b_1 b_2 b_3 b_4$ , onde  $b_1, \dots, b_4$  são os bits da mantissa neste exemplo.

Assim, o menor número positivo armazenado pelo computador de 8 bits recebe a seguinte configuração:



Tabela: Exemplo 4

$$0,0001 \times 2^{-2} = (0,000001)_2 = 2^{-6} = (0,015625)_{10}$$



# Representação de Números Reais no Computador

## Exemplo 4 - Menor número positivo:

Devemos lembrar que os expoentes 000 e 111 possuem tratamento especial.

Usa-se o expoente 000 para indicar que o número não está normalizado.

Assim, 000 será configurado como o menor expoente, isto é,  $-2$  neste exemplo.

Na forma desnormalizada, a mantissa passa a ser  $0, b_1 b_2 b_3 b_4$ , onde  $b_1, \dots, b_4$  são os bits da mantissa neste exemplo.

Assim, o menor número positivo armazenado pelo computador de 8 bits recebe a seguinte configuração:



Tabela: Exemplo 4

$$0,0001 \times 2^{-2} = (0,000001)_2 = 2^{-6} = (0,015625)_{10}$$

# Representação de Números Reais no Computador

## Exemplo 4 - Menor número positivo:

Devemos lembrar que os expoentes 000 e 111 possuem tratamento especial.

Usa-se o expoente 000 para indicar que o número não está normalizado.

Assim, 000 será configurado como o menor expoente, isto é,  $-2$  neste exemplo.

Na forma desnormalizada, a mantissa passa a ser  $0, b_1 b_2 b_3 b_4$ , onde  $b_1, \dots, b_4$  são os bits da mantissa neste exemplo.

Assim, o menor número positivo armazenado pelo computador de 8 bits recebe a seguinte configuração:



Tabela: Exemplo 4

$$0,0001 \times 2^{-2} = (0,000001)_2 = 2^{-6} = (0,015625)_{10}$$

# Representação de Números Reais no Computador

## Exemplo 4 - Menor número positivo:

Devemos lembrar que os expoentes 000 e 111 possuem tratamento especial.

Usa-se o expoente 000 para indicar que o número não está normalizado.

Assim, 000 será configurado como o menor expoente, isto é,  $-2$  neste exemplo.

Na forma desnormalizada, a mantissa passa a ser  $0, b_1 b_2 b_3 b_4$ , onde  $b_1, \dots, b_4$  são os bits da mantissa neste exemplo.

Assim, o menor número positivo armazenado pelo computador de 8 bits recebe a seguinte configuração:



Tabela: Exemplo 4

$$0,0001 \times 2^{-2} = (0,000001)_2 = 2^{-6} = (0,015625)_{10}$$

# Representação de Números Reais no Computador

## Exemplo 4 - Menor número positivo:

Devemos lembrar que os expoentes 000 e 111 possuem tratamento especial.

Usa-se o expoente 000 para indicar que o número não está normalizado.

Assim, 000 será configurado como o menor expoente, isto é,  $-2$  neste exemplo.

Na forma desnormalizada, a mantissa passa a ser  $0, b_1 b_2 b_3 b_4$ , onde  $b_1, \dots, b_4$  são os bits da mantissa neste exemplo.

Assim, o menor número positivo armazenado pelo computador de 8 bits recebe a seguinte configuração:

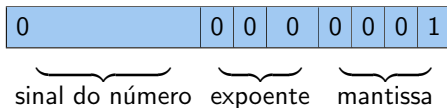


Tabela: Exemplo 4

$$0,0001 \times 2^{-2} = (0,000001)_2 = 2^{-6} = (0,015625)_{10}$$

# Representação de Números Reais no Computador

## Exemplo 4 - Menor número positivo:

Devemos lembrar que os expoentes 000 e 111 possuem tratamento especial.

Usa-se o expoente 000 para indicar que o número não está normalizado.

Assim, 000 será configurado como o menor expoente, isto é,  $-2$  neste exemplo.

Na forma desnormalizada, a mantissa passa a ser  $0, b_1 b_2 b_3 b_4$ , onde  $b_1, \dots, b_4$  são os bits da mantissa neste exemplo.

Assim, o menor número positivo armazenado pelo computador de 8 bits recebe a seguinte configuração:

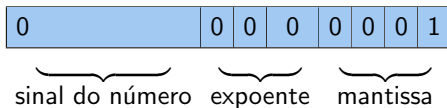


Tabela: Exemplo 4

$$0,0001 \times 2^{-2} = (0,000001)_2 = 2^{-6} = (0,015625)_{10}$$

# Representação de Números Reais no Computador

A configuração

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

representa  $+0$ ,

enquanto a configuração

1	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

representa  $-0$ , devendo ambos serem reconhecidos como iguais nas comparações.

O expoente 111 é reservado para representar  $+\infty$ ,

0	1	1	1	0	0	0	0
---	---	---	---	---	---	---	---

e  $-\infty$ , que é representado por

1	1	1	1	0	0	0	0
---	---	---	---	---	---	---	---

bastando trocar o bit de sinal do número para 1 por ser negativo.

As demais combinações com o expoente 111 não são válidas, sendo consideradas (NaN, ou Not a Number).

# Representação de Números Reais no Computador

A configuração

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

representa  $+0$ ,

enquanto a configuração

1	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

representa  $-0$ , devendo ambos serem reconhecidos como iguais nas comparações.

O expoente 111 é reservado para representar  $+\infty$ ,

0	1	1	1	0	0	0	0
---	---	---	---	---	---	---	---

e  $-\infty$ , que é representado por

1	1	1	1	0	0	0	0
---	---	---	---	---	---	---	---

bastando trocar o bit de sinal do número para 1 por ser negativo.

As demais combinações com o expoente 111 não são válidas, sendo consideradas (NaN, ou Not a Number).

# Representação de Números Reais no Computador

A configuração

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

representa  $+0$ ,

enquanto a configuração

1	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

representa  $-0$ , devendo ambos serem reconhecidos como iguais nas comparações.

O expoente 111 é reservado para representar  $+\infty$ ,

0	1	1	1	0	0	0	0
---	---	---	---	---	---	---	---

e  $-\infty$ , que é representado por

1	1	1	1	0	0	0	0
---	---	---	---	---	---	---	---

bastando trocar o bit de sinal do número para 1 por ser negativo.

As demais combinações com o expoente 111 não são válidas, sendo consideradas (NaN, ou Not a Number).



# Representação de Números Reais no Computador

E quando a mantissa não cabe nos 4 bits?

Somos obrigados a arredondar a mantissa para que ela caiba nos 4 bits.

Vamos, assim, perder precisão no número e ele não mais representará, exatamente, o número desejado.

**Exemplo 5:**

$$(9,375)_{10} = (1001,011)_2 = 1,001011 \times 2^3$$

⇒ Possui 6 dígitos depois da vírgula, mas o computador só tem reservados 4 bits para a mantissa.

⇒ Deve ser feito o arredondamento para 4 casas decimais: se o dígito abandonado após as 4 casas decimais for menor que 1 (ou seja, se for 0), os dígitos anteriores devem ser mantidos. Se o dígito abandonado for igual a 1, somo 1 ao dígito anterior.

# Representação de Números Reais no Computador

E quando a mantissa não cabe nos 4 bits?

Somos obrigados a arredondar a mantissa para que ela caiba nos 4 bits.

Vamos, assim, perder precisão no número e ele não mais representará, exatamente, o número desejado.

**Exemplo 5:**

$$(9,375)_{10} = (1001,011)_2 = 1,001011 \times 2^3$$

⇒ Possui 6 dígitos depois da vírgula, mas o computador só tem reservados 4 bits para a mantissa.

⇒ Deve ser feito o arredondamento para 4 casas decimais: se o dígito abandonado após as 4 casas decimais for menor que 1 (ou seja, se for 0), os dígitos anteriores devem ser mantidos. Se o dígito abandonado for igual a 1, somo 1 ao dígito anterior.

# Representação de Números Reais no Computador

E quando a mantissa não cabe nos 4 bits?

Somos obrigados a arredondar a mantissa para que ela caiba nos 4 bits.

Vamos, assim, perder precisão no número e ele não mais representará, exatamente, o número desejado.

## Exemplo 5:

$$(9,375)_{10} = (1001,011)_2 = 1,001011 \times 2^3$$

⇒ Possui 6 dígitos depois da vírgula, mas o computador só tem reservados 4 bits para a mantissa.

⇒ Deve ser feito o arredondamento para 4 casas decimais: se o dígito abandonado após as 4 casas decimais for menor que 1 (ou seja, se for 0), os dígitos anteriores devem ser mantidos. Se o dígito abandonado for igual a 1, somo 1 ao dígito anterior.

# Representação de Números Reais no Computador

E quando a mantissa não cabe nos 4 bits?

Somos obrigados a arredondar a mantissa para que ela caiba nos 4 bits.

Vamos, assim, perder precisão no número e ele não mais representará, exatamente, o número desejado.

## Exemplo 5:

$$(9,375)_{10} = (1001,011)_2 = 1,001011 \times 2^3$$

⇒ Possui 6 dígitos depois da vírgula, mas o computador só tem reservados 4 bits para a mantissa.

⇒ Deve ser feito o arredondamento para 4 casas decimais: se o dígito abandonado após as 4 casas decimais for menor que 1 (ou seja, se for 0), os dígitos anteriores devem ser mantidos. Se o dígito abandonado for igual a 1, somo 1 ao dígito anterior.

# Representação de Números Reais no Computador

E quando a mantissa não cabe nos 4 bits?

Somos obrigados a arredondar a mantissa para que ela caiba nos 4 bits.

Vamos, assim, perder precisão no número e ele não mais representará, exatamente, o número desejado.

## Exemplo 5:

$$(9,375)_{10} = (1001,011)_2 = 1,001011 \times 2^3$$

⇒ Possui 6 dígitos depois da vírgula, mas o computador só tem reservados 4 bits para a mantissa.

⇒ Deve ser feito o arredondamento para 4 casas decimais: se o dígito abandonado após as 4 casas decimais for menor que 1 (ou seja, se for 0), os dígitos anteriores devem ser mantidos. Se o dígito abandonado for igual a 1, somo 1 ao dígito anterior.

# Representação de Números Reais no Computador

E quando a mantissa não cabe nos 4 bits?

Somos obrigados a arredondar a mantissa para que ela caiba nos 4 bits.

Vamos, assim, perder precisão no número e ele não mais representará, exatamente, o número desejado.

## Exemplo 5:

$$(9,375)_{10} = (1001,011)_2 = 1,001011 \times 2^3$$

⇒ Possui 6 dígitos depois da vírgula, mas o computador só tem reservados 4 bits para a mantissa.

⇒ Deve ser feito o arredondamento para 4 casas decimais: se o dígito abandonado após as 4 casas decimais for menor que 1 (ou seja, se for 0), os dígitos anteriores devem ser mantidos. Se o dígito abandonado for igual a 1, somo 1 ao dígito anterior.

# Representação de Números Reais no Computador

## Exemplo 5:

$$(9,375)_{10} = (1001,011)_2 = 1,0010\mathbf{11} \times 2^3 = 1,0011 \times 2^3$$



$$\text{Exemplo 6: } (9,125)_{10} = (1001,001)_2 = 1,0010\mathbf{01} \times 2^3 = 1,0010 \times 2^3$$



# Representação de Números Reais no Computador

## Exemplo 5:

$$(9,375)_{10} = (1001,011)_2 = 1,0010\textcolor{red}{1} \times 2^3 = 1,0011 \times 2^3$$



$$\text{Exemplo 6: } (9,125)_{10} = (1001,001)_2 = 1,0010\textcolor{red}{0}1 \times 2^3 = 1,0010 \times 2^3$$

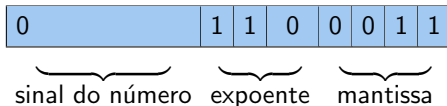




# Representação de Números Reais no Computador

## Exemplo 5:

$$(9,375)_{10} = (1001,011)_2 = 1,0010\mathbf{11} \times 2^3 = 1,0011 \times 2^3$$



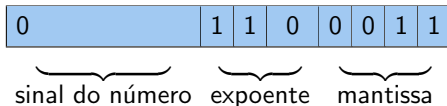
$$\text{Exemplo 6: } (9,125)_{10} = (1001,001)_2 = 1,0010\mathbf{01} \times 2^3 = 1,0010 \times 2^3$$



# Representação de Números Reais no Computador

## Exemplo 5:

$$(9,375)_{10} = (1001,011)_2 = 1,0010\mathbf{11} \times 2^3 = 1,0011 \times 2^3$$



## Exemplo 6:

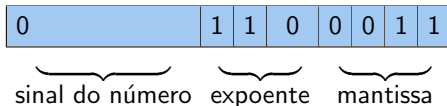
$$(9,125)_{10} = (1001,001)_2 = 1,0010\mathbf{01} \times 2^3 = 1,0010 \times 2^3$$



# Representação de Números Reais no Computador

## Exemplo 5:

$$(9,375)_{10} = (1001,011)_2 = 1,0010\mathbf{11} \times 2^3 = 1,0011 \times 2^3$$



## Exemplo 6:

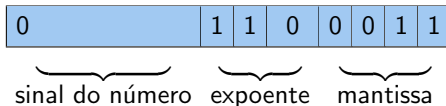
$$(9,125)_{10} = (1001,001)_2 = 1,0010\mathbf{01} \times 2^3 = 1,0010 \times 2^3$$



# Representação de Números Reais no Computador

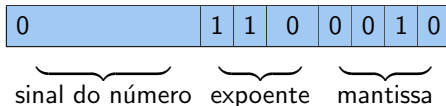
## Exemplo 5:

$$(9,375)_{10} = (1001,011)_2 = 1,0010\mathbf{11} \times 2^3 = 1,0011 \times 2^3$$



## Exemplo 6:

$$(9,125)_{10} = (1001,001)_2 = 1,0010\mathbf{01} \times 2^3 = 1,0010 \times 2^3$$





Oliveira, R. **Capítulo 2 - Representação binária de números inteiros e reais** - Site: [www.raymundodeoliveira.eng.br/binario.html](http://www.raymundodeoliveira.eng.br/binario.html)



Oliveira, R. **Representação de ponto flutuante IEEE** - Site: <https://learn.microsoft.com/pt-br/cpp/build/ieee-floating-point-representation?view=msvc-170>