
Introdução à Ciência da Computação: armazenamento de dados

Parte 5 – Números Reais

Prof. Danilo Medeiros Eler
danilo.eler@unesp.br

Conteúdo

- Representação e Armazenamento de Dados
 - Texto
 - Imagem
 - Números Inteiros
 - Números Reais

Armazenamento de Números

- Armazenamento do ponto decimal
 - Ponto fixo
 - Números inteiros
 - Número sem uma parte fracionária
 - Ponto flutuante
 - Números reais
 - Números com uma parte fracionária

Armazenamento de Números

- O tamanho do padrão binário irá definir o intervalo de valores que conseguirmos representar
- Por exemplo, se usarmos 1 byte conseguiremos representar números inteiros com valores de -128 a 127

Números Inteiros

- Usando o tipo de variável byte

```
byte a;
```

```
a = 100;
```

```
exibir("Valor da variável a: "+a);
```

```
a = 127;
```

```
exibir("Valor da variável a: "+a);
```

```
a = 128;
```

```
exibir ("Valor da variável a: "+a);
```

```
a = 200;
```

```
exibir ("Valor da variável a: "+a);
```

Números Inteiros

- Usando o tipo de variável byte

```
byte a;
```

```
a = 100;
```

```
exibir("Valor da variável a: "+a);
```

```
a = 127;
```

```
exibir("Valor da variável a: "+a);
```

```
a = 128;
```

```
exibir ("Valor da variável a: "+a);
```

```
a = 200;
```

```
exibir ("Valor da variável a: "+a);
```

Valor da variável a: 100

Números Inteiros

- Usando o tipo de variável byte

```
byte a;
```

```
a = 100;
```

```
exibir("Valor da variável a: "+a);
```

```
a = 127;
```

```
exibir("Valor da variável a: "+a);
```

```
a = 128;
```

```
exibir ("Valor da variável a: "+a);
```

```
a = 200;
```

```
exibir ("Valor da variável a: "+a);
```

Valor da variável a: 100

Valor da variável a: 127

Números Inteiros

- Usando o tipo de variável byte

```
byte a;
```

```
a = 100;
```

```
exibir("Valor da variável a: "+a);
```

```
a = 127;
```

```
exibir("Valor da variável a: "+a);
```

```
a = 128;
```

```
exibir ("Valor da variável a: "+a);
```

```
a = 200;
```

```
exibir ("Valor da variável a: "+a);
```

```
Valor da variável a: 100  
Valor da variável a: 127  
Valor da variável a: -128
```


Números Inteiros

- Usando o tipo de variável byte

```
byte a;
```

```
a = 100;
```

```
exibir("Valor da variável a: "+a);
```

```
a = 127;
```

```
exibir("Valor da variável a: "+a);
```

```
a = 128;
```

```
exibir ("Valor da variável a: "+a);
```

```
a = 200;
```

```
exibir ("Valor da variável a: "+a);
```

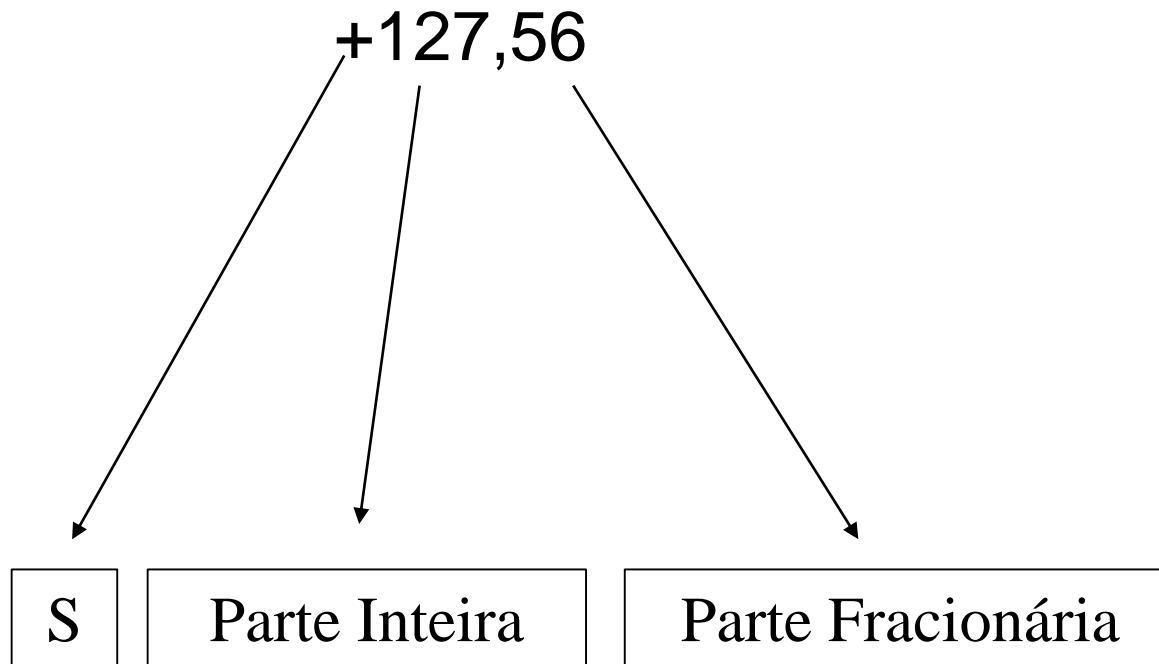
```
Valor da variável a: 100  
Valor da variável a: 127  
Valor da variável a: -128  
Valor da variável a: -56
```

Armazenamento de Números Reais

- Como podemos armazenar um número real em um padrão binário?
- Exemplos de números reais
 - 127,56
 - -127,56
 - 123465321654651,5
 - 65465134165,2465465416
 - -2,23165465416

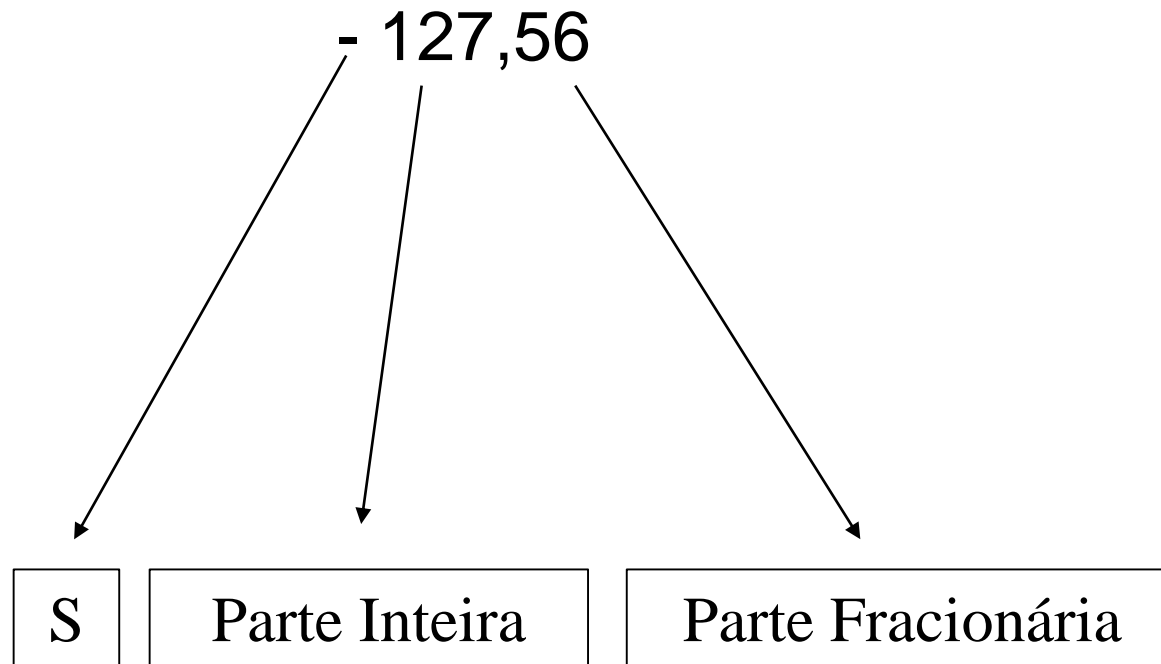
Armazenamento de Números Reais

- Exemplos de números reais



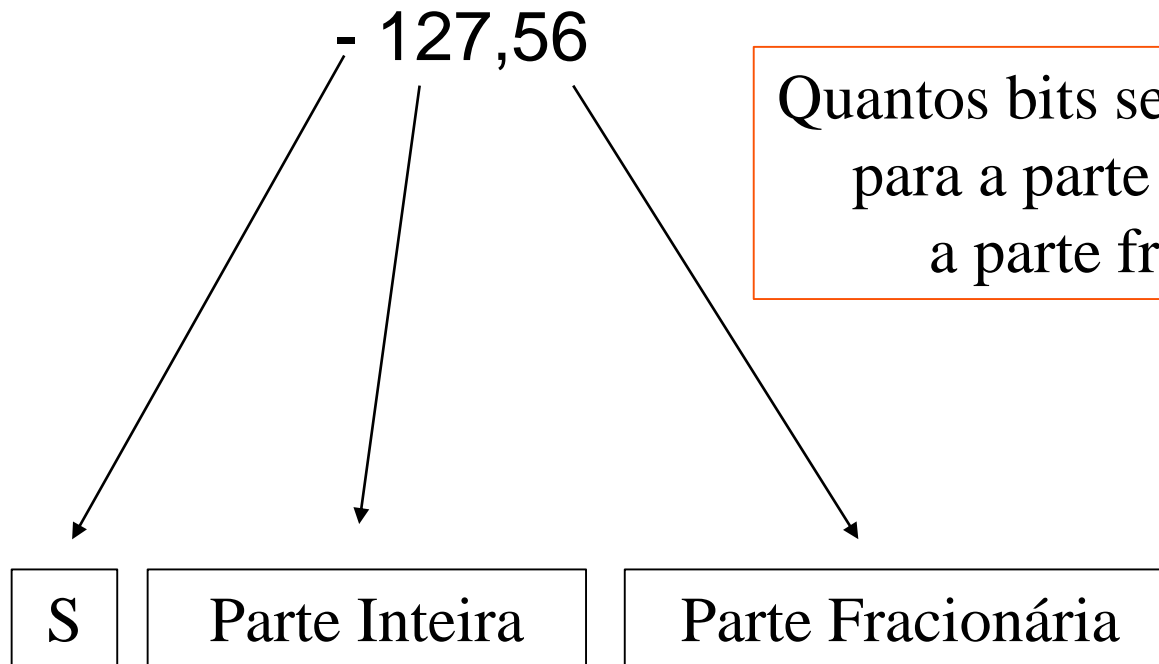
Armazenamento de Números Reais

- Exemplos de números reais



Armazenamento de Números Reais

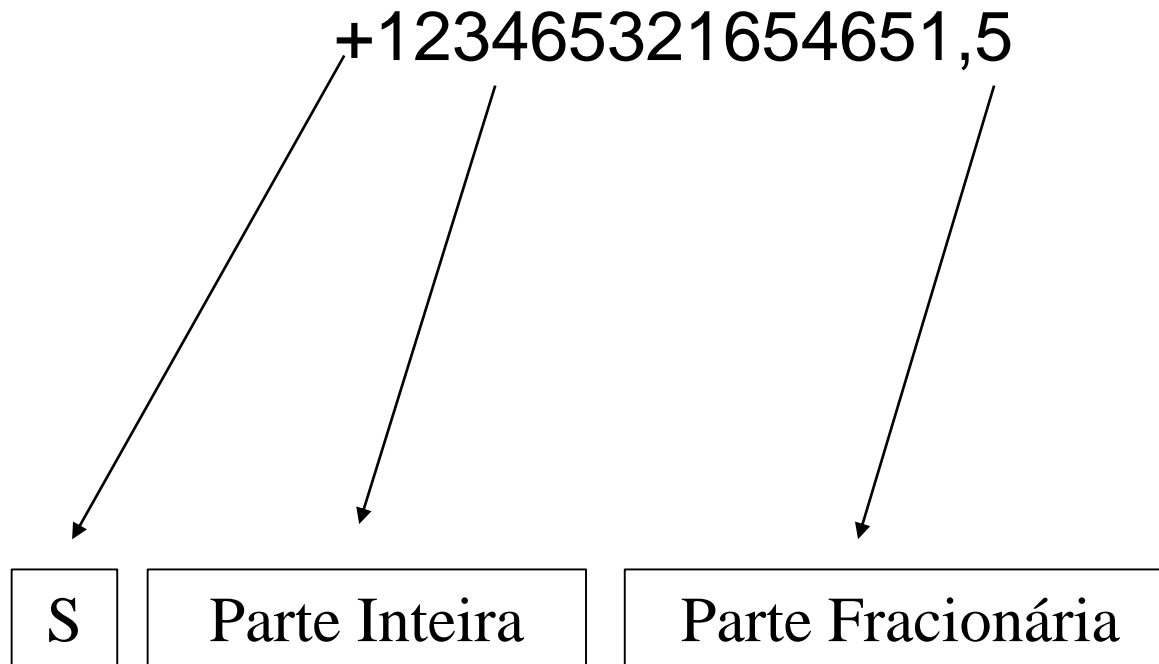
- Exemplos de números reais



Quantos bits seriam reservados para a parte inteira e para a parte fracionária?

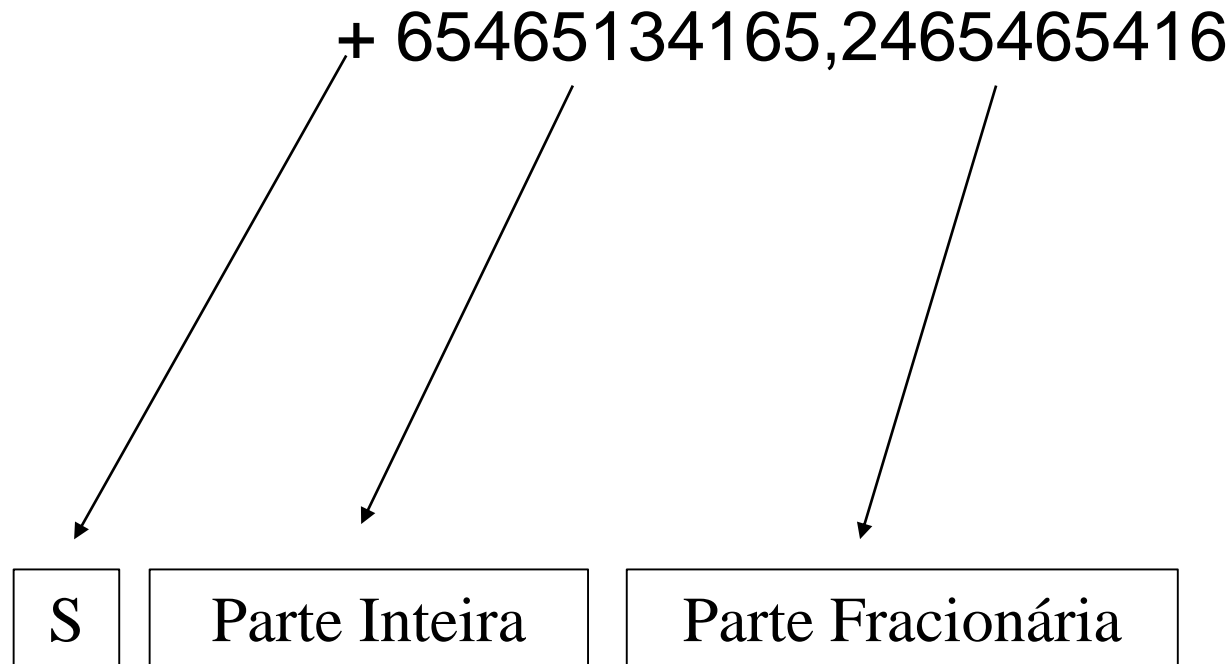
Armazenamento de Números Reais

- Exemplos de números reais



Armazenamento de Números Reais

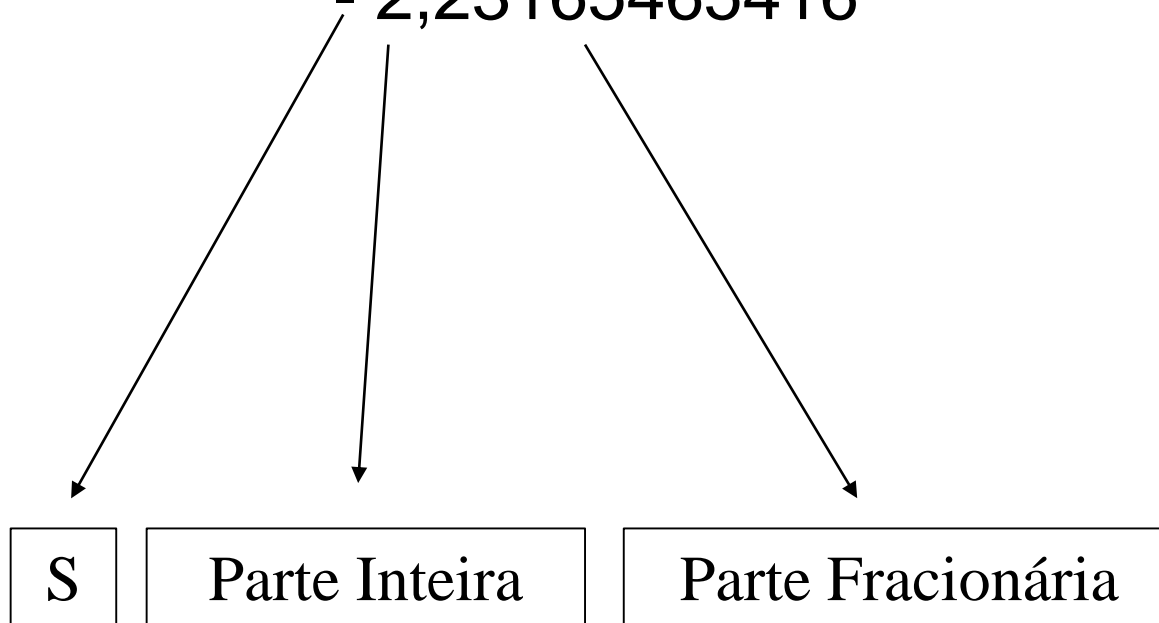
- Exemplos de números reais



Armazenamento de Números Reais

- Exemplos de números reais

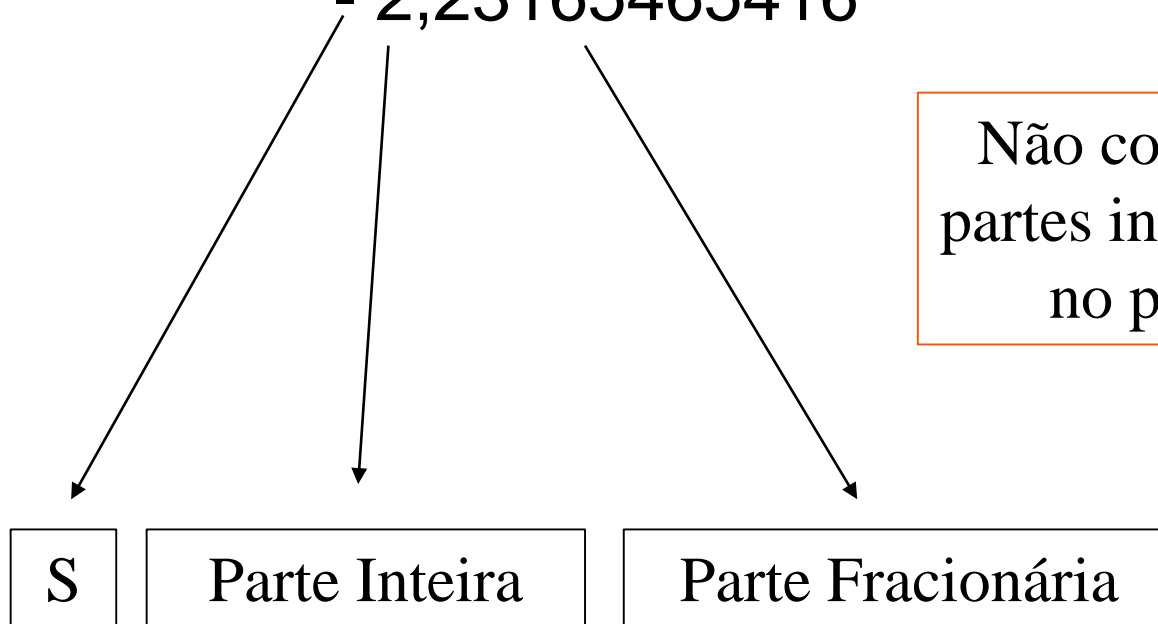
- 2,23165465416



Armazenamento de Números Reais

- Exemplos de números reais

- 2,23165465416



Não convém separar as partes inteira e fracionária no padrão binário

Armazenamento de Números Reais

- A estratégia adotada será baseada no ponto flutuante para que o 'ponto' flutue
- Essa representação permite varia a quantidade necessária para representar a parte inteira e fracionária
- Essa representação será facilitada utilizando uma representação de ponto fixo

Armazenamento de Números Reais

- Representação de ponto fixo

123456789.00

Armazenamento de Números Reais

- Representação de ponto fixo

123456789.00

1.23456792E8 (notação científica)

Armazenamento de Números Reais

- Representação de ponto fixo

123456789.00

1.23456792E8 (notação científica)

1.23456792×10^8 (notação científica)

Armazenamento de Números Reais

- Representação de ponto fixo

123456789.8765

1.234567898765E8 (notação científica)

1.234567898765 x 10⁸ (notação científica)

Armazenamento de Números Reais

- Representação de ponto fixo

0.00000123456789

1.23456789E-6 (notação científica)

$1.23456789 \times 10^{-6}$ (notação científica)

- Essa é a representação utilizada para armazenar no padrão binário

Armazenamento de Números Reais

- Representação de ponto fixo

+123456789.00

+1.23456792E8 (notação científica)

+1.23456792 x 10⁸



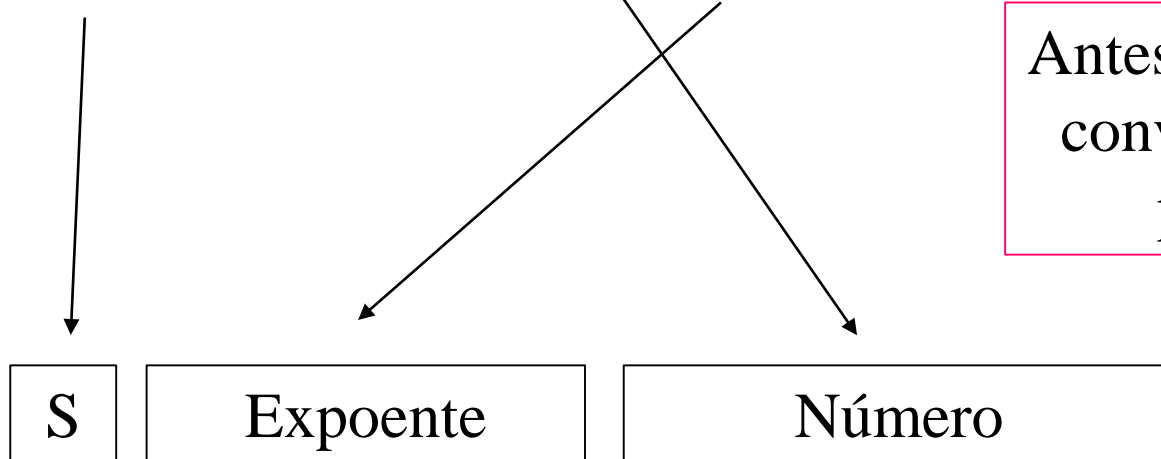
Armazenamento de Números Reais

- Representação de ponto fixo

+123456789.00

+1.23456792E8 (notação científica)

+1.23456792 x 10⁸



Antes, será necessário converter o número para binário

Converter parte Fracionária para Binário

- Parte fracionária: 0.625

$$\begin{array}{r} 0,625 \\ \times \quad 2 \\ \hline \end{array}$$

Decimal:

Binário:



Converter parte Fracionária para Binário

- Parte fracionária: 0.625

Decimal:

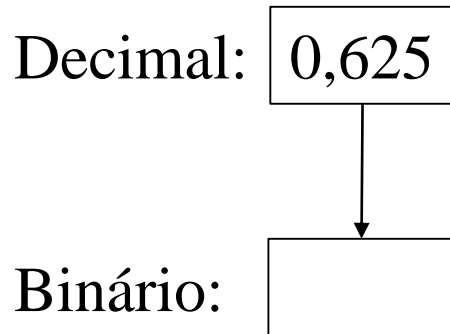
↓

Binário:

$$\begin{array}{r} 0,625 \\ \times \quad 2 \\ \hline 1,25 \end{array}$$

Converter parte Fracionária para Binário

- Parte fracionária: 0.625



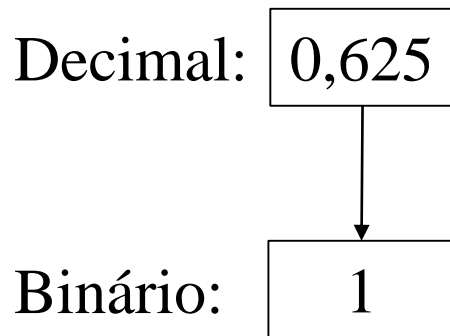
$$\begin{array}{r} 0,625 \\ \times \quad 2 \\ \hline 1,25 \end{array}$$

Parte Inteira: 1

Parte Fracionária: 0,25

Converter parte Fracionária para Binário

- Parte fracionária: 0.625



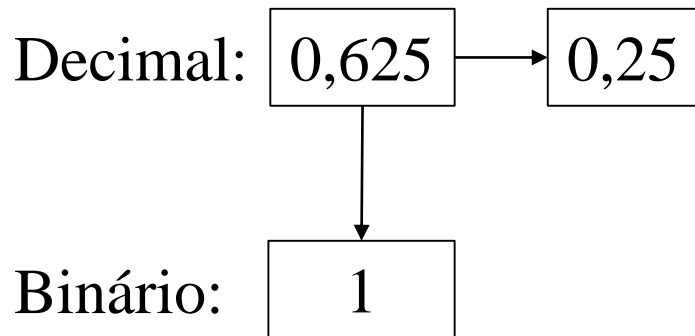
$$\begin{array}{r} 0,625 \\ \times \quad 2 \\ \hline 1,25 \end{array}$$

Parte Inteira: 1

Parte Fracionária: 0,25

Converter parte Fracionária para Binário

- Parte fracionária: 0.625



$$\begin{array}{r} 0,625 \\ \times \quad 2 \\ \hline 1,25 \end{array}$$

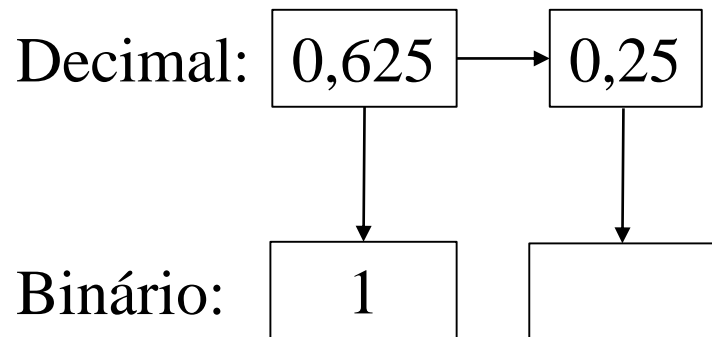
Parte Inteira: 1

Parte Fracionária: 0,25

Converter parte Fracionária para Binário

- Parte fracionária: 0.625

$$\begin{array}{r} 0,25 \\ \times \quad 2 \\ \hline \end{array}$$

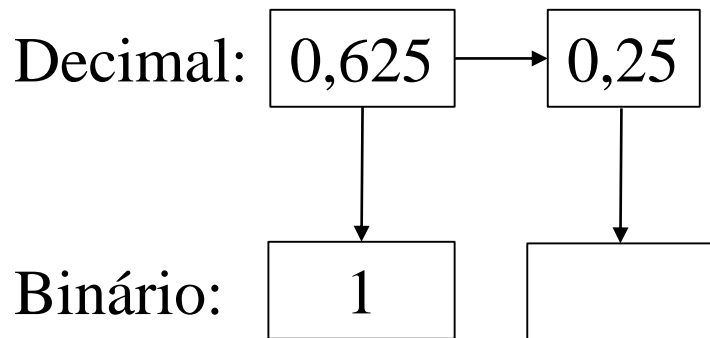


Parte Inteira:

Parte Fracionária:

Converter parte Fracionária para Binário

- Parte fracionária: 0.625



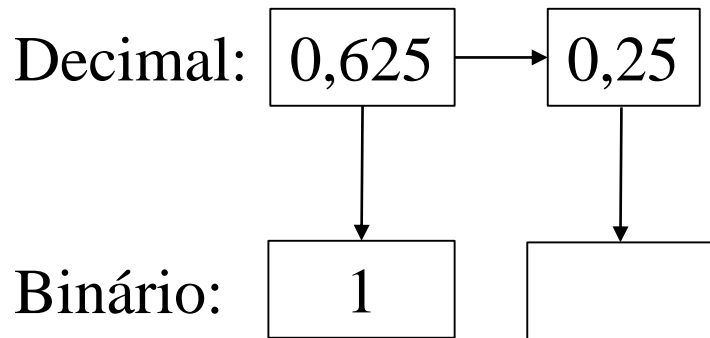
$$\begin{array}{r} 0,25 \\ \times \quad 2 \\ \hline 0,5 \end{array}$$

Parte Inteira:

Parte Fracionária:

Converter parte Fracionária para Binário

- Parte fracionária: 0.625



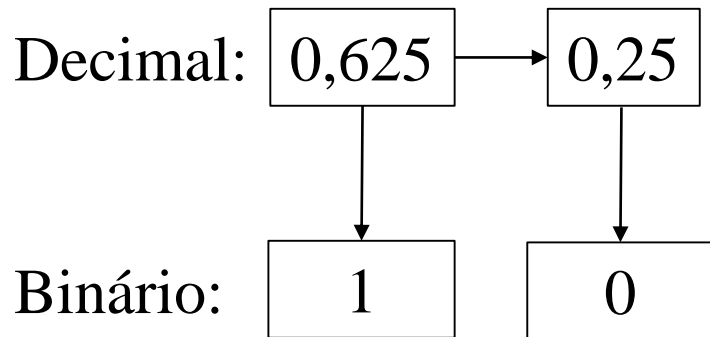
$$\begin{array}{r} 0,25 \\ \times \quad 2 \\ \hline 0,5 \end{array}$$

Parte Inteira: 0

Parte Fracionária: 0,5

Converter parte Fracionária para Binário

- Parte fracionária: 0.625



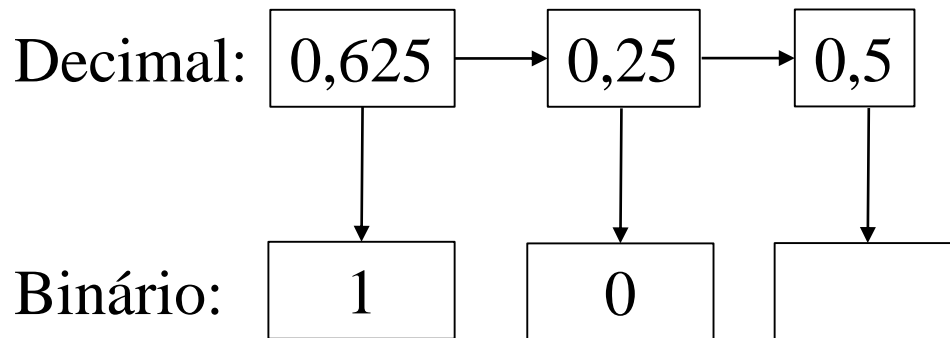
$$\begin{array}{r} 0,25 \\ \times \quad 2 \\ \hline 0,5 \end{array}$$

Parte Inteira: 0

Parte Fracionária: 0,5

Converter parte Fracionária para Binário

- Parte fracionária: 0.625



$$\begin{array}{r} 0,25 \\ \times \quad 2 \\ \hline 0,5 \end{array}$$

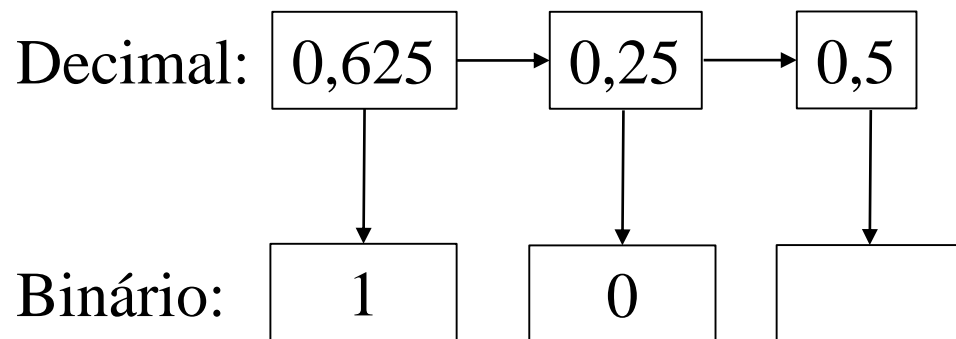
Parte Inteira: 0

Parte Fracionária: 0,5

Converter parte Fracionária para Binário

- Parte fracionária: 0.625

$$\begin{array}{r} 0,5 \\ \times 2 \\ \hline \end{array}$$

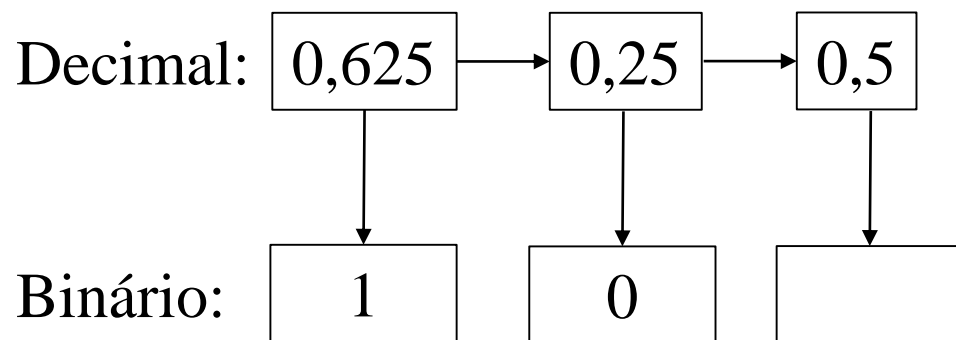


Parte Inteira:

Parte Fracionária:

Converter parte Fracionária para Binário

- Parte fracionária: 0.625



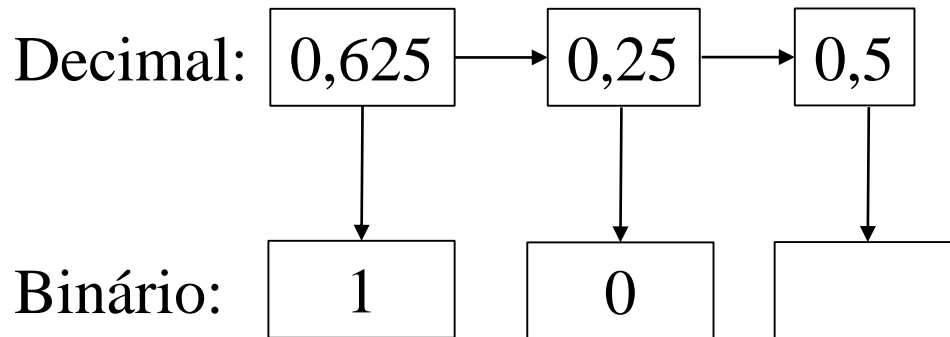
$$\begin{array}{r} 0,5 \\ \times 2 \\ \hline 1,0 \end{array}$$

Parte Inteira:

Parte Fracionária:

Converter parte Fracionária para Binário

- Parte fracionária: 0.625



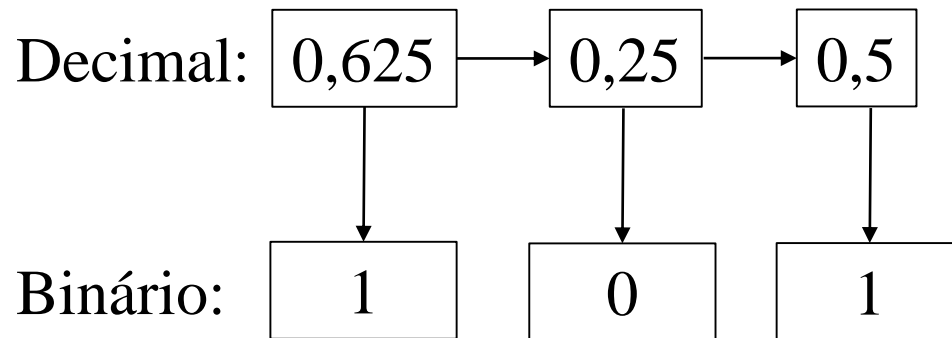
$$\begin{array}{r} 0,5 \\ \times 2 \\ \hline 1,0 \end{array}$$

Parte Inteira: 1

Parte Fracionária: 0

Converter parte Fracionária para Binário

- Parte fracionária: 0.625



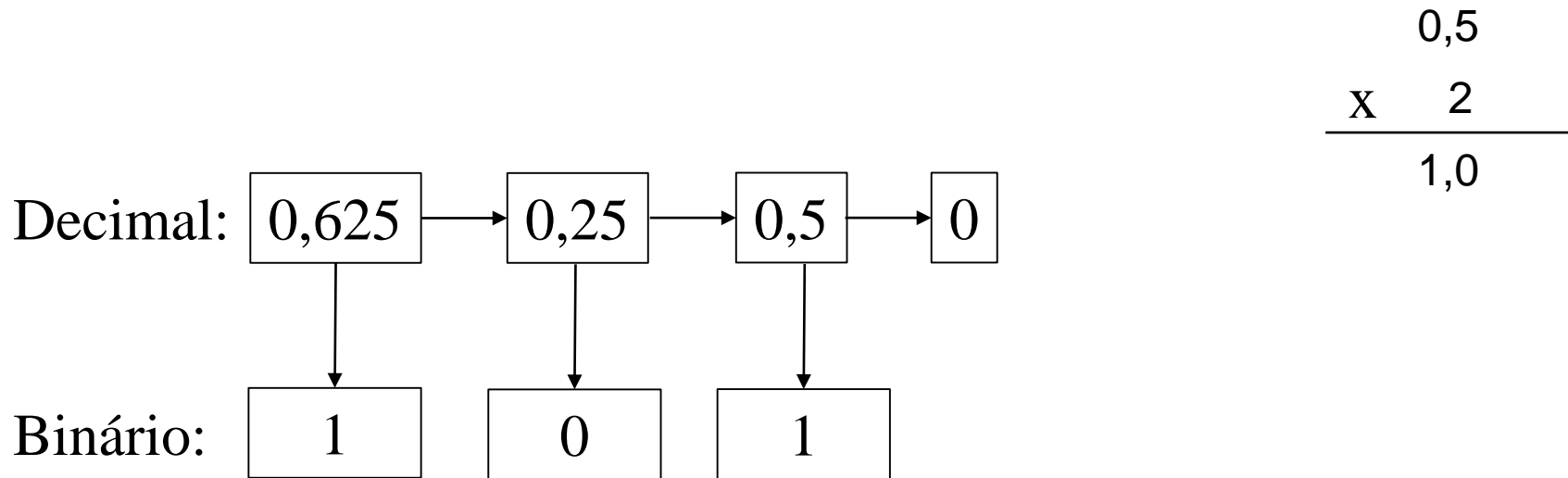
$$\begin{array}{r} 0,5 \\ \times 2 \\ \hline 1,0 \end{array}$$

Parte Inteira: 1

Parte Fracionária: 0

Converter parte Fracionária para Binário

- Parte fracionária: 0.625

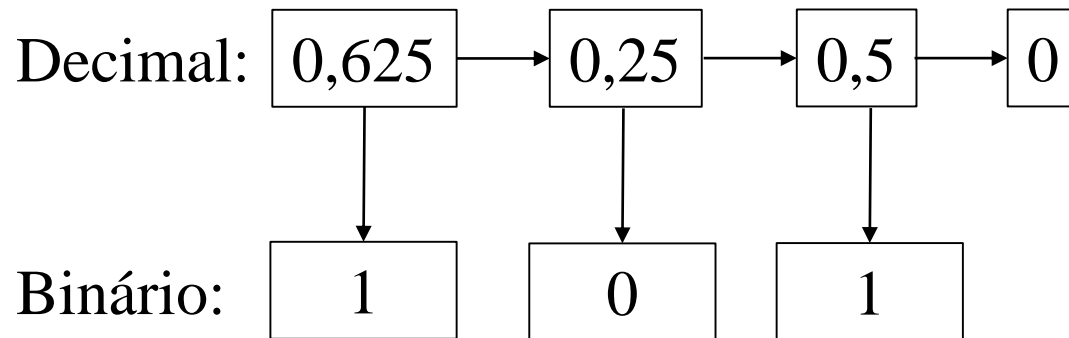


Parte Inteira: 1

Parte Fracionária: 0

Converter parte Fracionária para Binário

- Parte fracionária: 0.625



Sistema Binário

- Um número pode ser escrito como

$$\pm (S_{K-1} \dots S_2 S_1 S_0 \cdot S_{-1} S_{-2} S_{-3} \dots S_{-L})_2$$

- tem o valor de:

- $n =$

$$\pm \underbrace{S_{K-1} \times 2^{K-1} + \dots + S_1 \times 2^1 + S_0 \times 2^0}_{\text{Parte Inteira}} + \underbrace{S_{-1} \times 2^{-1} + S_{-2} \times 2^{-2} + \dots + S_{-L} \times 2^{-L}}_{\text{Parte Fracionária}}$$

Parte Inteira

Parte Fracionária

Sistema Binário

- Um número pode ser escrito como

$$\pm (S_{K-1} \dots S_2 S_1 S_0 \cdot S_{-1} S_{-2} S_{-3} \dots S_{-L})_2$$

- tem o valor de:

- $n =$

$$\pm \underbrace{S_{K-1} \times 2^{K-1} + \dots + S_1 \times 2^1 + S_0 \times 2^0}_{\text{Parte Inteira}} + \underbrace{S_{-1} \times 2^{-1} + S_{-2} \times 2^{-2} + \dots + S_{-L} \times 2^{-L}}_{\text{Parte Fracionária}}$$

Parte Inteira

Parte Fracionária

Sistema Binário

- Um número pode ser escrito como

$$\pm (S_{K-1} \dots S_2 S_1 S_0 \cdot S_{-1} S_{-2} S_{-3} \dots S_{-L})_2$$

- tem o valor de:

- $n =$

$$\pm \underbrace{S_{K-1} \times 2^{K-1} + \dots + S_1 \times 2^1 + S_0 \times 2^0}_{\text{Parte Inteira}} + \underbrace{S_{-1} \times 2^{-1} + S_{-2} \times 2^{-2} + \dots + S_{-L} \times 2^{-L}}_{\text{Parte Fracionária}}$$

Parte Inteira

Parte Fracionária

$$1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3}$$

Sistema Binário

- Um número pode ser escrito como

$$\pm (S_{K-1} \dots S_2 S_1 S_0 \cdot S_{-1} S_{-2} S_{-3} \dots S_{-L})_2$$

- tem o valor de:

- $n =$

$$\pm \underbrace{S_{K-1} \times 2^{K-1} + \dots + S_1 \times 2^1 + S_0 \times 2^0}_{\text{Parte Inteira}} + \underbrace{S_{-1} \times 2^{-1} + S_{-2} \times 2^{-2} + \dots + S_{-L} \times 2^{-L}}_{\text{Parte Fracionária}}$$

Parte Inteira

Parte Fracionária

$$1 \times 0,5 + 0 \times 0,25 + 1 \times 0,125$$

Sistema Binário

- Um número pode ser escrito como

$$\pm (S_{K-1} \dots S_2 S_1 S_0 \cdot S_{-1} S_{-2} S_{-3} \dots S_{-L})_2$$

- tem o valor de:

- $n =$

$$\pm \underbrace{S_{K-1} \times 2^{K-1} + \dots + S_1 \times 2^1 + S_0 \times 2^0}_{\text{Parte Inteira}} + \underbrace{S_{-1} \times 2^{-1} + S_{-2} \times 2^{-2} + \dots + S_{-L} \times 2^{-L}}_{\text{Parte Fracionária}}$$

Parte Inteira

Parte Fracionária

$$0,5 + 0 + 0,125$$

Sistema Binário

- Um número pode ser escrito como

$$\pm (S_{K-1} \dots S_2 S_1 S_0 \cdot S_{-1} S_{-2} S_{-3} \dots S_{-L})_2$$

- tem o valor de:

- $n =$

$$\pm \underbrace{S_{K-1} \times 2^{K-1} + \dots + S_1 \times 2^1 + S_0 \times 2^0}_{\text{Parte Inteira}} + \underbrace{S_{-1} \times 2^{-1} + S_{-2} \times 2^{-2} + \dots + S_{-L} \times 2^{-L}}_{\text{Parte Fracionária}}$$

Parte Inteira

Parte Fracionária

0,625

Sistema Binário

- Por exemplo, o número $(6,625)_{10}$ seria representado em binário por $(110,101)_2$
 - A parte inteira '6' é representada por 110
 - A parte fracionária '0,625' é representada por 101
- Em seguida, o número 110,101 será representado pela notação de ponto fixo para então ser armazenado

Armazenamento de Números Reais

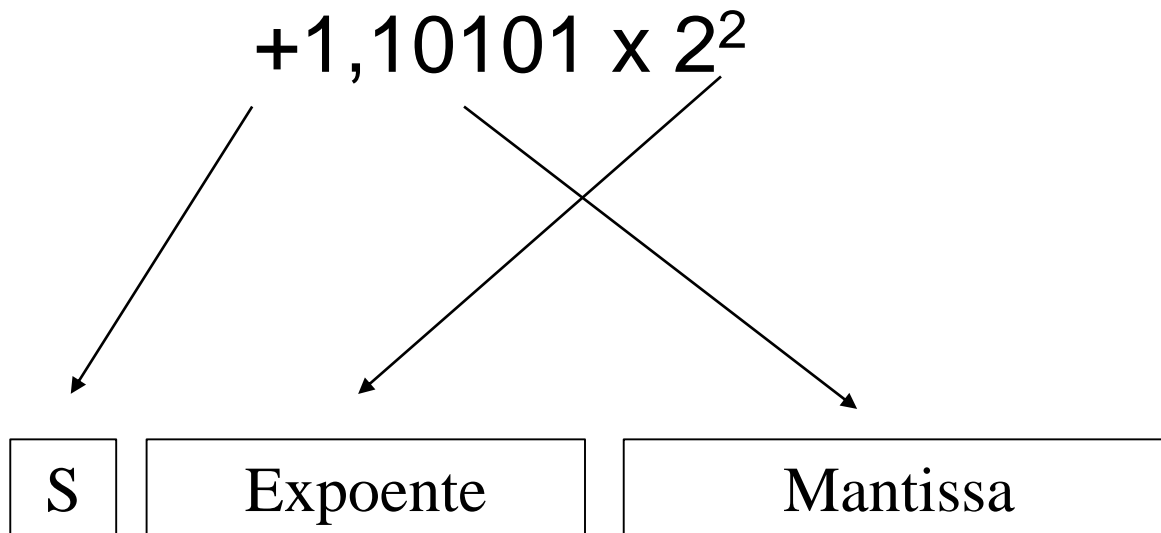
- Decimal: +6,625
- Binário: +110,101

Armazenamento de Números Reais

- Decimal: +6,625
- Binário: +110,101
- Notação Científica:
 $+1,10101 \times 2^2$

Armazenamento de Números Reais

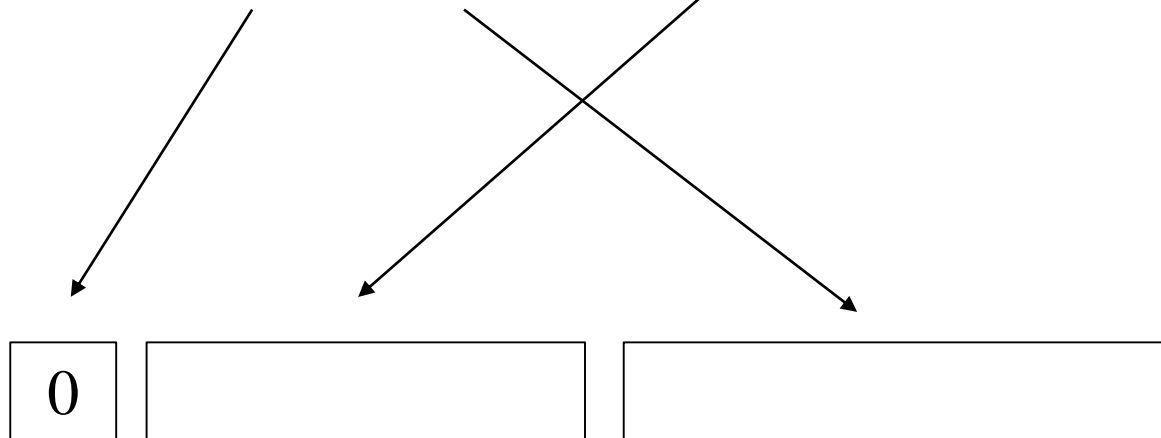
- Decimal: +6,625
- Binário: +110,101
- Notação Científica:



Armazenamento de Números Reais

- Decimal: +6,625
- Binário: +110,101
- Notação Científica:

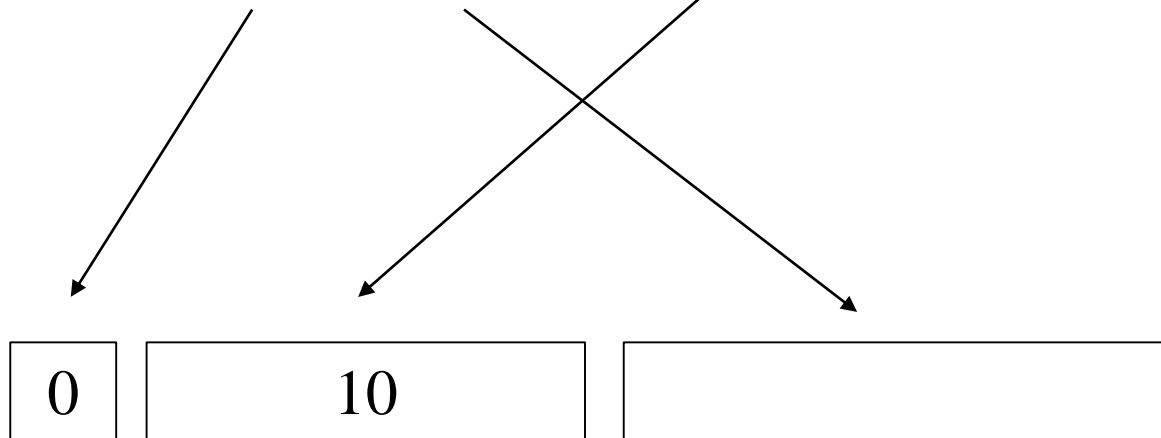
$$+1,10101 \times 2^2$$



Armazenamento de Números Reais

- Decimal: +6,625
- Binário: +110,101
- Notação Científica:

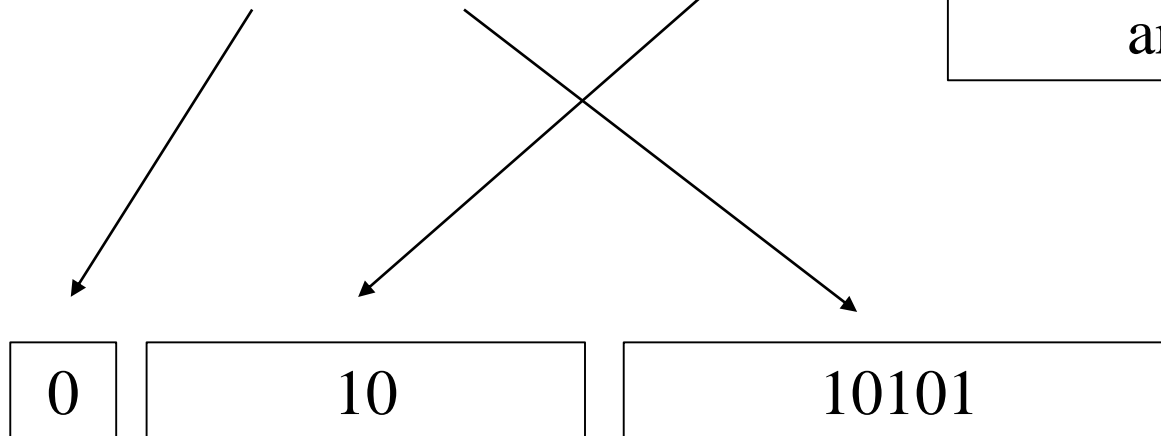
$$+1, \underline{10101} \times 2^2$$



Armazenamento de Números Reais

- Decimal: +6,625
- Binário: +110,101
- Notação Científica:

$$+1, \underline{10101} \times 2^2$$

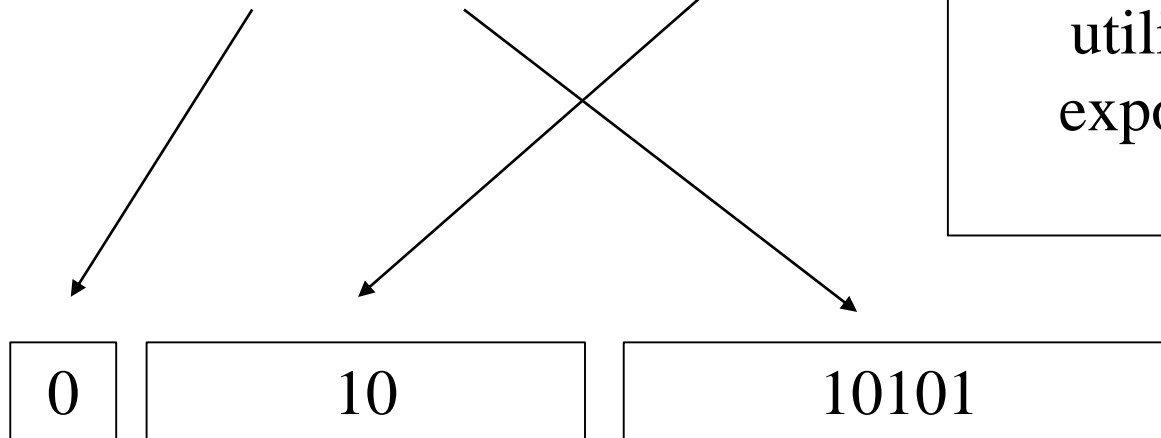


O '1' à esquerda está implícito e não é armazenado

Armazenamento de Números Reais

- Decimal: +6,625
- Binário: +110,101
- Notação Científica:

$$+1, \underline{10101} \times 2^2$$

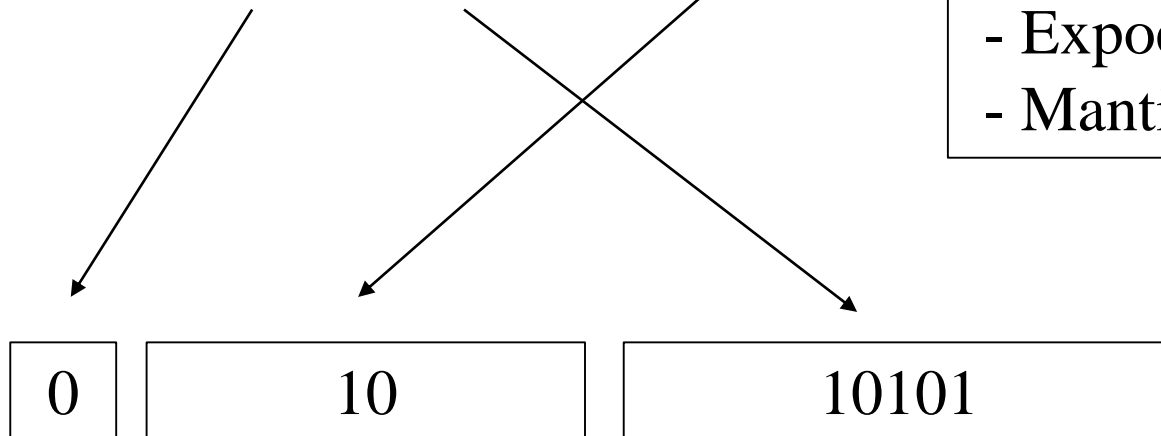


Precisamos definir quantos bits serão utilizados para o expoente e para a mantissa

Armazenamento de Números Reais

- Decimal: +6,625
- Binário: +110,101
- Notação Científica:

$$+1, \underline{10101} \times 2^2$$



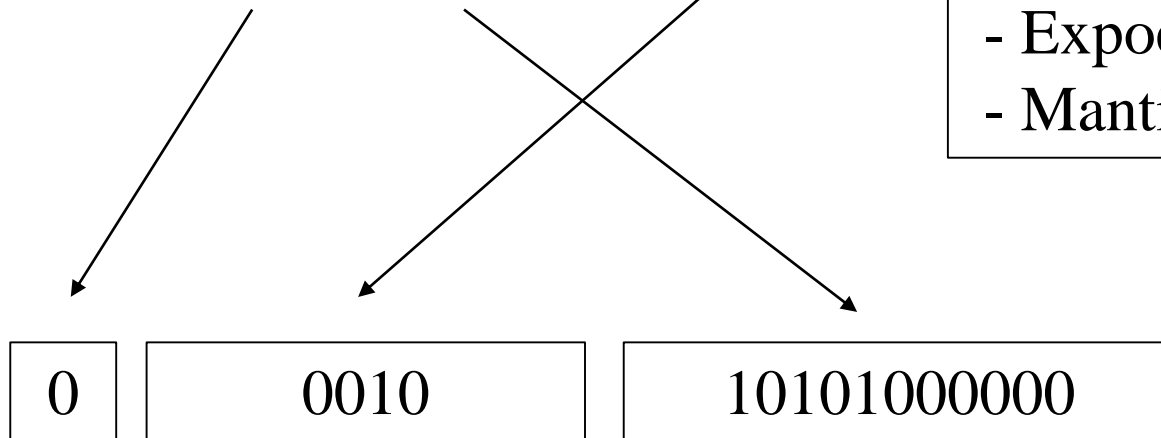
Considerando 16 bits:

- Sinal: 1 bit
- Expoente: 4 bits
- Mantissa: 11 bits

Armazenamento de Números Reais

- Decimal: +6,625
- Binário: +110,101
- Notação Científica:

$$+1, \underline{10101} \times 2^2$$



Considerando 16 bits:

- Sinal: 1 bit
- Expoente: 4 bits
- Mantissa: 11 bits

Armazenamento de Números Reais

- Decimal: +6,625
- Binário: +110,101
- Notação Científica:
 $+1,10101 \times 2^2$
- Padrão binário: 0001010101000000

Considerando 16 bits:

- Sinal: 1 bit
- Expoente: 4 bits
- Mantissa: 11 bits

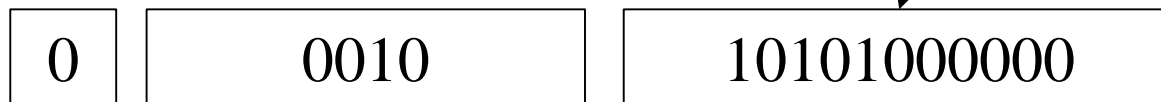


Armazenamento de Números Reais

- Decimal: +6,625
- Binário: +110,101
- Notação Científica:

$$+1,10101 \times 2^2$$

- Padrão binário: **0****0010****10101000000**



Considerando 16 bits:

- Sinal: 1 bit
- Expoente: 4 bits
- Mantissa: 11 bits

Armazenamento de Números Reais

- Recuperando o valor armazenado no padrão binário

0001010101000000

Armazenamento de Números Reais

- Recuperando o valor armazenado no padrão binário

0001010101000000

Considerando 16 bits:

- Sinal: 1 bit
- Expoente: 4 bits
- Mantissa: 11 bits

Armazenamento de Números Reais

- Recuperando o valor armazenado no padrão binário

0001010101000000

- Sinal: 0
- Expoente: 0010
- Mantissa: 10101000000

Considerando 16 bits:

- Sinal: 1 bit
- Expoente: 4 bits
- Mantissa: 11 bits

Armazenamento de Números Reais

- Recuperando o valor armazenado no padrão binário

0001010101000000

- Sinal: 0
- Expoente: 0010
- Mantissa: 10101000000
1,10101000000

Considerando 16 bits:

- Sinal: 1 bit
- Expoente: 4 bits
- Mantissa: 11 bits

Recupera-se o '1' implícito

Armazenamento de Números Reais

- Recuperando o valor armazenado no padrão binário

0001010101000000

- Sinal: 0
- Expoente: 0010
- Mantissa: 10101000000
1,10101000000

Considerando 16 bits:

- Sinal: 1 bit
- Expoente: 4 bits
- Mantissa: 11 bits

Recupera-se o '1' implícito
Desloca-se a vírgula

Armazenamento de Números Reais

- Recuperando o valor armazenado no padrão binário

0001010101000000

- Sinal: 0
- Expoente: 0010
- Mantissa: 10101000000
110,101000000

Considerando 16 bits:

- Sinal: 1 bit
- Expoente: 4 bits
- Mantissa: 11 bits

Recupera-se o '1' implícito
Desloca-se a vírgula

Armazenamento de Números Reais

- Recuperando o valor armazenado no padrão binário

0001010101000000

- Sinal: 0
- Expoente: 0010
- Mantissa: 10101000000

$(110,101000000)_2$

$(6,)_10$

Considerando 16 bits:

- Sinal: 1 bit
- Expoente: 4 bits
- Mantissa: 11 bits

Recupera-se o '1' implícito

Desloca-se a vírgula

Converte a parte inteira

Armazenamento de Números Reais

- Recuperando o valor armazenado no padrão binário

0001010101000000

- Sinal: 0
- Expoente: 0010
- Mantissa: 10101000000
 $(110,101000000)_2$
 $(6,625)_{10}$

Considerando 16 bits:

- Sinal: 1 bit
- Expoente: 4 bits
- Mantissa: 11 bits

Recupera-se o '1' implícito

Desloca-se a vírgula

Converte a parte inteira

Converte a parte fracionária

Armazenamento de Números Reais

- Recuperando o valor armazenado no padrão binário

0001010101000000

- Sinal: 0
- Expoente: 0010
- Mantissa: 10101000000
 $(+110,101000000)_2$
 $(+6,625)_{10}$

Considerando 16 bits:

- Sinal: 1 bit
- Expoente: 4 bits
- Mantissa: 11 bits

Recupera-se o '1' implícito

Desloca-se a vírgula

Converte a parte inteira

Converte a parte fracionária

Adiciona o sinal

Armazenamento de Números Reais

- O expoente pode ser positivo ou negativo
 - Poderia ser representado com a representação complemento de dois
- O expoente será representado de uma maneira que se armazene somente inteiros sem sinal (inteiros positivos)
- Para isso, um inteiro (chamado bias) é adicionado ao valor do expoente

Armazenamento de Números Reais

- O bias do expoente é calculado como $2^{m-1} - 1$
 - m representa o número de bits reservado para armazenar o expoente
 - Ex.: com 4 bits o bias seria $2^{m-1} - 1$
 - $2^{m-1} - 1 = 2^{4-1} - 1 = 2^3 - 1 = 8 - 1 = 7$
- Esse sistema de representação do expoente é chamado de Excesso
 - No caso, temos o Excesso 7
 - Pois sempre se soma 7 ao valor do expoente antes de armazená-lo como binário

Armazenamento de Números Reais

- Excesso 7
 - Soma 7 ao valor do expoente
- Ex.: +110,101
 - Normalização: $1,10101 \times 2^2$
 - Sinal: 0
 - Expoente: $2 = 10$
 - Mantissa: 10101

Armazenamento de Números Reais

- Excesso 7
 - Soma 7 ao valor do expoente
- Ex.: +110,101
 - Normalização: $1,10101 \times 2^2$
 - Sinal: 0
 - Expoente: $(2+7) =$
 - Mantissa: 10101

Armazenamento de Números Reais

- Excesso 7
 - Soma 7 ao valor do expoente
- Ex.: +110,101
 - Normalização: $1,10101 \times 2^2$
 - Sinal: 0
 - Expoente: (9) =
 - Mantissa: 10101

Armazenamento de Números Reais

- Excesso 7
 - Soma 7 ao valor do expoente
- Ex.: +110,101
 - Normalização: $1,10101 \times 2^2$
 - Sinal: 0
 - Expoente: $(9) = 1001$
 - Mantissa: 10101

Armazenamento de Números Reais

- Excesso 7
 - Soma 7 ao valor do expoente
- Ex.: +110,101
 - Normalização: $1,10101 \times 2^2$
 - Sinal: 0
 - Expoente: $(9) = 1001$
 - Mantissa: 10101
 - Padrão binário: 0100110101000000

Armazenamento de Números Reais

- Recuperando um número real armazenado no sistema Excesso 7
 - Subtrai 7 ao valor do expoente
- Ex.:
 - Padrão binário: 0100110101000000

Armazenamento de Números Reais

- Recuperando um número real armazenado no sistema Excesso 7
 - Subtrai 7 ao valor do expoente
- Ex.:
 - Padrão binário: 0100110101000000
 - Sinal: 0

Armazenamento de Números Reais

- Recuperando um número real armazenado no sistema Excesso 7
 - Subtrai 7 ao valor do expoente
- Ex.:
 - Padrão binário: 0100110101000000
 - Sinal: 0
 - Expoente: 1001

Armazenamento de Números Reais

- Recuperando um número real armazenado no sistema Excesso 7
 - Subtrai 7 ao valor do expoente
- Ex.:
 - Padrão binário: 0100110101000000
 - Sinal: 0
 - Expoente: 1001
 - Mantissa: 10101000000

Armazenamento de Números Reais

- Recuperando um número real armazenado no sistema Excesso 7
 - Subtrai 7 ao valor do expoente
- Ex.:
 - Padrão binário: 0100110101000000
 - Sinal: 0
 - Expoente: $1001 = 9$
 - Mantissa: 10101000000

Armazenamento de Números Reais

- Recuperando um número real armazenado no sistema Excesso 7
 - Subtrai 7 ao valor do expoente
- Ex.:
 - Padrão binário: 0100110101000000
 - Sinal: 0
 - Expoente: $1001 = 9 - 7 =$
 - Mantissa: 10101000000

Armazenamento de Números Reais

- Recuperando um número real armazenado no sistema Excesso 7
 - Subtrai 7 ao valor do expoente
- Ex.:
 - Padrão binário: 0100110101000000
 - Sinal: 0
 - Expoente: $1001 = 9 - 7 = 2$
 - Mantissa: 10101000000

Armazenamento de Números Reais

- Recuperando um número real armazenado no sistema Excesso 7
 - Subtrai 7 ao valor do expoente
- Ex.:
 - Padrão binário: 0100110101000000
 - Sinal: 0
 - Expoente: $1001 = 9 - 7 = 2$
 - Mantissa: 10101000000
 - Número recuperado: $1,10101000000$

Armazenamento de Números Reais

- Recuperando um número real armazenado no sistema Excesso 7
 - Subtrai 7 ao valor do expoente
- Ex.:
 - Padrão binário: 0100110101000000
 - Sinal: 0
 - Expoente: $1001 = 9 - 7 = 2$
 - Mantissa: 10101000000
 - Número recuperado: $110,101000000$

Armazenamento de Números Reais

- Recuperando um número real armazenado no sistema Excesso 7
 - Subtrai 7 ao valor do expoente
- Ex.:
 - Padrão binário: 0100110101000000
 - Sinal: 0
 - Expoente: $1001 = 9 - 7 = 2$
 - Mantissa: 10101000000
 - Número recuperado: $+110,101000000$

Armazenamento de Números Reais

- Recuperando um número real armazenado no sistema Excesso 7
 - Subtrai 7 ao valor do expoente
- Ex.:
 - Padrão binário: 0100110101000000
 - Sinal: 0
 - Expoente: $1001 = 9 - 7 = 2$
 - Mantissa: 10101000000
 - Número recuperado: $+110,101000000 = (+6,625)_{10}$

Armazenamento de Números Reais

- Excesso 7
 - Soma 7 ao valor do expoente
- Ex.: $(-0,75)_{10} = (-0,11)_2$

Armazenamento de Números Reais

- Excesso 7
 - Soma 7 ao valor do expoente
- Ex.: $(-0,75)_{10} = (-0,11)_2$
 - Normalização: $-1,1 \times 2^{-1}$

Armazenamento de Números Reais

- Excesso 7
 - Soma 7 ao valor do expoente
- Ex.: $(-0,75)_{10} = (-0,11)_2$
 - Normalização: $-1,1 \times 2^{-1}$
 - Sinal: 1

Armazenamento de Números Reais

- Excesso 7
 - Soma 7 ao valor do expoente
- Ex.: $(-0,75)_{10} = (-0,11)_2$
 - Normalização: $-1,1 \times 2^{-1}$
 - Sinal: 1
 - Expoente: -1

Armazenamento de Números Reais

- Excesso 7
 - Soma 7 ao valor do expoente
- Ex.: $(-0,75)_{10} = (-0,11)_2$
 - Normalização: $-1,1 \times 2^{-1}$
 - Sinal: 1
 - Expoente: $-1+7 = 6$

Armazenamento de Números Reais

- Excesso 7
 - Soma 7 ao valor do expoente
- Ex.: $(-0,75)_{10} = (-0,11)_2$
 - Normalização: $-1,1 \times 2^{-1}$
 - Sinal: 1
 - Expoente: $-1+7 = 6 = 0110$

Armazenamento de Números Reais

- Excesso 7
 - Soma 7 ao valor do expoente
- Ex.: $(-0,75)_{10} = (-0,11)_2$
 - Normalização: $-1,1 \times 2^{-1}$
 - Sinal: 1
 - Expoente: $-1+7 = 6 = 0110$
 - Mantissa: 1

Armazenamento de Números Reais

- Excesso 7
 - Soma 7 ao valor do expoente
- Ex.: $(-0,75)_{10} = (-0,11)_2$
 - Normalização: $-1,1 \times 2^{-1}$
 - Sinal: 1
 - Expoente: $-1+7 = 6 = 0110$
 - Mantissa: 1
 - Padrão binário:

Armazenamento de Números Reais

- Excesso 7
 - Soma 7 ao valor do expoente
- Ex.: $(-0,75)_{10} = (-0,11)_2$
 - Normalização: $-1,1 \times 2^{-1}$
 - Sinal: 1
 - Expoente: $-1+7 = 6 = 0110$
 - Mantissa: 1
 - Padrão binário: 1

Armazenamento de Números Reais

- Excesso 7
 - Soma 7 ao valor do expoente
- Ex.: $(-0,75)_{10} = (-0,11)_2$
 - Normalização: $-1,1 \times 2^{-1}$
 - Sinal: 1
 - Expoente: $-1+7 = 6 = 0110$
 - Mantissa: 1
 - Padrão binário: 10110

Armazenamento de Números Reais

- Excesso 7
 - Soma 7 ao valor do expoente
- Ex.: $(-0,75)_{10} = (-0,11)_2$
 - Normalização: $-1,1 \times 2^{-1}$
 - Sinal: 1
 - Expoente: $-1+7 = 6 = 0110$
 - Mantissa: 1
 - Padrão binário: 101101

Armazenamento de Números Reais

- Excesso 7
 - Soma 7 ao valor do expoente
- Ex.: $(-0,75)_{10} = (-0,11)_2$
 - Normalização: $-1,1 \times 2^{-1}$
 - Sinal: 1
 - Expoente: $-1+7 = 6 = 0110$
 - Mantissa: 1
 - Padrão binário: 1011010000000000

Armazenamento de Números Reais

- Recuperando um número real armazenado no sistema Excesso 7
 - Subtrai 7 ao valor do expoente
- Ex.:
 - Padrão binário: 1011010000000000

Armazenamento de Números Reais

- Recuperando um número real armazenado no sistema Excesso 7
 - Subtrai 7 ao valor do expoente
- Ex.:
 - Padrão binário: 1011010000000000
 - Sinal: 1

Armazenamento de Números Reais

- Recuperando um número real armazenado no sistema Excesso 7
 - Subtrai 7 ao valor do expoente
- Ex.:
 - Padrão binário: 1011010000000000
 - Sinal: 1
 - Expoente: 0110

Armazenamento de Números Reais

- Recuperando um número real armazenado no sistema Excesso 7
 - Subtrai 7 ao valor do expoente
- Ex.:
 - Padrão binário: 1011010000000000
 - Sinal: 1
 - Expoente: $0110 = 6$

Armazenamento de Números Reais

- Recuperando um número real armazenado no sistema Excesso 7
 - Subtrai 7 ao valor do expoente
- Ex.:
 - Padrão binário: 1011010000000000
 - Sinal: 1
 - Expoente: $0110 = 6 - 7 = -1$

Armazenamento de Números Reais

- Recuperando um número real armazenado no sistema Excesso 7
 - Subtrai 7 ao valor do expoente
- Ex.:
 - Padrão binário: 1011010000000000
 - Sinal: 1
 - Expoente: $0110 = 6 - 7 = -1$
 - Mantissa: 100000000000

Armazenamento de Números Reais

- Recuperando um número real armazenado no sistema Excesso 7
 - Subtrai 7 ao valor do expoente
- Ex.:
 - Padrão binário: 1011010000000000
 - Sinal: 1
 - Expoente: $0110 = 6 - 7 = -1$
 - Mantissa: 100000000000

Armazenamento de Números Reais

- Recuperando um número real armazenado no sistema Excesso 7
 - Subtrai 7 ao valor do expoente
- Ex.:
 - Padrão binário: 1011010000000000
 - Sinal: 1
 - Expoente: $0110 = 6 - 7 = -1$
 - Mantissa: 10000000000
 - Número recuperado:

Armazenamento de Números Reais

- Recuperando um número real armazenado no sistema Excesso 7
 - Subtrai 7 ao valor do expoente
- Ex.:
 - Padrão binário: 1011010000000000
 - Sinal: 1
 - Expoente: $0110 = 6 - 7 = -1$
 - Mantissa: 100000000000
 - Número recuperado: $1,100000000000$

Armazenamento de Números Reais

- Recuperando um número real armazenado no sistema Excesso 7
 - Subtrai 7 ao valor do expoente
- Ex.:
 - Padrão binário: 1011010000000000
 - Sinal: 1
 - Expoente: $0110 = 6 - 7 = -1$
 - Mantissa: 10000000000
 - Número recuperado: $1,10000000000$

Armazenamento de Números Reais

- Recuperando um número real armazenado no sistema Excesso 7
 - Subtrai 7 ao valor do expoente
- Ex.:
 - Padrão binário: 1011010000000000
 - Sinal: 1
 - Expoente: $0110 = 6 - 7 = -1$
 - Mantissa: 10000000000
 - Número recuperado: 0,110000000000

Armazenamento de Números Reais

- Recuperando um número real armazenado no sistema Excesso 7
 - Subtrai 7 ao valor do expoente
- Ex.:
 - Padrão binário: 1011010000000000
 - Sinal: 1
 - Expoente: $0110 = 6 - 7 = -1$
 - Mantissa: 10000000000
 - Número recuperado: $-0,110000000000$

Armazenamento de Números Reais

- Recuperando um número real armazenado no sistema Excesso 7
 - Subtrai 7 ao valor do expoente
- Ex.:
 - Padrão binário: 1011010000000000
 - Sinal: 1
 - Expoente: $0110 = 6 - 7 = -1$
 - Mantissa: 10000000000
 - Número recuperado: $-0,110000000000 = (-0,75)_{10}$

Erro de Truncamento

- O erro de truncamento ocorre quando a quantidade de bits não é suficiente para armazenar o valor, por exemplo:
 - Considere 11 bits para armazenamento da mantissa
 - Número binário: $1111111111,111111$
 - Normalização: $1,1111111111111111 \times 2^9$

Erro de Truncamento

- O erro de truncamento ocorre quando a quantidade de bits não é suficiente para armazenar o valor, por exemplo:
 - Considere 11 bits para armazenamento da mantissa
 - Número binário: 1111111111,111111
 - Normalização: 1,1111111111111111 x 2^9
 - A mantissa possui 15 bits: 111111111111111
 - Somente 11 serão armazenados: 111111111111111
 - Os bits mais à direita serão excluídos

Erro de Truncamento

- O erro de truncamento ocorre quando a quantidade de bits não é suficiente para armazenar o valor, por exemplo:
 - Considere 11 bits para armazenamento da mantissa
 - Número binário: 1111111111,111111
 - Normalização: 1,1111111111111111 x 2^9
 - A mantissa possui 15 bits: 111111111111111
 - Somente 11 serão armazenados: 1111111111
 - Os bits mais à direita serão excluídos

Erro de Truncamento

- O erro de truncamento ocorre quando a quantidade de bits não é suficiente para armazenar o valor, por exemplo:
 - Considere 11 bits para armazenamento da mantissa
 - Número binário: 1111111111,111111
 - Normalização: 1,1111111111111111 x 2^9
 - A mantissa possui 15 bits: 111111111111111
 - Somente 11 serão armazenados: 1111111111
 - Os bits mais à direita serão excluídos
 - Valor recuperado: 1,1111111111

Erro de Truncamento

- O erro de truncamento ocorre quando a quantidade de bits não é suficiente para armazenar o valor, por exemplo:
 - Considere 11 bits para armazenamento da mantissa
 - Número binário: 1111111111,111111
 - Normalização: 1,1111111111111111 x 2^9
 - A mantissa possui 15 bits: 111111111111111
 - Somente 11 serão armazenados: 1111111111
 - Os bits mais à direita serão excluídos
 - Valor recuperado: 1111111111,11

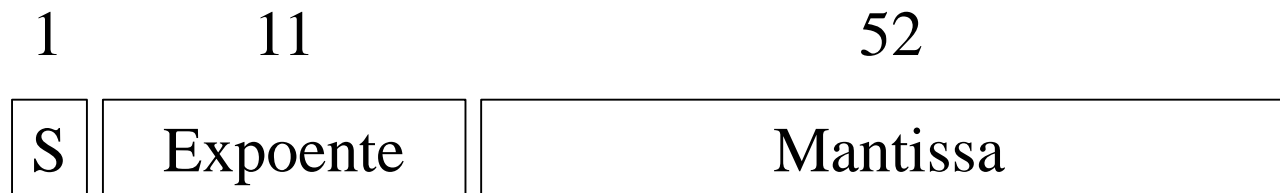
Armazenamento de Números Reais

- Exemplo de padrões para representação de ponto flutuante

Excesso 127 – Precisão simples (32 bits)



Excesso 1023 – Precisão dupla (64 bits)



Exemplo em Java

```
System.out.println("Float: "+Float.SIZE+" bits");
System.out.println("Float - Max expoente: "+Float.MAX_EXPONENT);
System.out.println("Float - Min expoente: "+Float.MIN_EXPONENT);

System.out.println("Double: "+Double.SIZE+" bits");
System.out.println("Double - Max expoente: "+Double.MAX_EXPONENT);
System.out.println("Double - Min expoente: "+Double.MIN_EXPONENT);
```

Float: 32 bits

Float - Max expoente: 127

Float - Min expoente: -126

Double: 64 bits

Double - Max expoente: 1023

Double - Min expoente: -1022

Bibliografia

BIBLIOGRAFIA BÁSICA

1. BROOKSHEAR, J. G. **Ciência da computação**: uma visão abrangente. 5ª ed., Bookman Editora, 2000. 499p.
2. FOROUZAN, B. A., MOSHARRAF, F. **Fundamentos da Ciência da Computação**. 2ª ed., São Paulo: Cengage Learning, 2011. 560p.

BIBLIOGRAFIA COMPLEMENTAR

1. BROOKSHEAR, J. G. **Ciência da computação**: uma visão abrangente. 5ª ed., Bookman Editora, 2000. 499p.
2. CORMEN, T.H., Leiserson, C.E., Rivest R.L., Stein, C. **Algoritmos**: teoria e Prática. Rio de Janeiro: Editora Campus, 2002. 916p.
3. PLAUGER, P. L. **A Biblioteca Standard C**. Rio de Janeiro: Editora Campus, 1994. 614p.
4. PRATA, S. **C primer plus**, 4ª ed. SAMS Publishing, 2002. 931p.