Laboratório de Pesquisa em Redes e Multimídia

Representação de Números em Ponto Flutuante

OBS: Esta aula é uma reprodução, sob a forma de slides, da aula em vídeo disponibilizada pelo prof. Rex Medeiros, da UFRN/ECT, em https://youtu.be/OVuyMcnPKOc

Universidade Federal do Espírito Santo
Departamento de Informática

Notação Científica

- A notação científica é usada para representar números muito grandes ou muito pequenos, frequentes em áreas como a física e a química.
  - Massa do próton = 0,000 000 000 000 000 000 000 000 00167 kg
  - Massa do elétron = 0,000 000 000 000 000 000 000 000 000 000 910 938 22 kg
  - Massa da Terra = 5 973 600 000 000 000 000 000 000 kg
- Na base decimal, um número N expresso em notação científica tem a forma geral $N = m \times 10^e$ (ou $N = mEe$), onde
  - m = mantissa (um valor entre 1 e 10)
  - e = expoente (define a ordem de grandeza do número)
- Exemplos
  - $300 = 3 \times 10^2$ (m=3 e=2); $300\,000\,000\,000 = 3 \times 10^{11}$ (m=2 e=11) ou 3E11
  - $0{,}00000000000000000000000000167 = 1{,}67 \times 10^{-27} = 1{,}67E27$ (m=1,67 e=-27)
  - $0{,}0000000586 = 5{,}86 \times 10^{-8}$ ou 5,86E-8 (m=5,86 e=-8)
  - $-12\,000\,000\,000\,000 = -1{,}2 \times 10^{13}$ ou (m=-1,2 e=13) -1,2E13

Ponto Flutuante na Forma Normalizada

- A notação em ponto flutuante na forma normalizada é análoga à notação científica, com a diferença que a parte inteira é sempre igual a zero e o primeiro dígito após a vírgula (*d1*) é sempre diferente de zero. Ex: 0,5678 x 10^3

$$x = \pm 0, d_1 d_2 \ldots d_p \times \beta^e$$
$$= d \times \beta^e$$

$$x \neq 0 \qquad d_1 \neq 0 \qquad d_1, \ldots, d_p \in \{0, \ldots, \beta - 1\}$$

- $d_1 d_1 d_2 \ldots d_p$ = mantissa (5678)
- p = precisão (4), que determina o número de dígitos na mantissa
- β = base de numeração (10)
- e = expoente (3)

Exemplos

1) Colocar na notação de ponto flutuante em forma normalizada, usando a base 10

- $2345{,}89 = 0{,}23589 \times 10^{3}$
- $0{,}0000586 = 0{,}586 \times 10^{-4}$
- $-12\,000\,000\,000\,000 = -\,0{,}12 \times 10^{14}$

2) Colocar o número $(-3{,}625)_{10}$ na notação de ponto flutuante em forma normalizada, usando a base 2

- Convertendo o número para a base 2: $(-11{,}101)_{2}$
- Normalizando: $(-0{,}11101 \times 2^{2})$

Sistemas de Numeração em Ponto Flutuante

- Os sistemas computacionais representam os números reais por meio de um sistema de numeração em ponto flutuante, cuja forma padrão geral é:

Sistemas de Numeração em Ponto Flutuante

$$F(\beta, p, m, M)$$

β é a base, p é a precisão, m e M são os valores mínimos e máximos que o expoente pode assumir.

Exemplos

1) Escrever o número $(-3,625)_{10}$ no sistema F(10,5,-3,3)

1º passo: colocar o número na forma normalizada, observando que ele já se encontra na base do sistema em questão

$(-0,3625 \times 10^{1})$

2º passo: verificar se o número de dígitos na mantissa é menor, igual ou maior do que a precisão do sistema de ponto flutuante. Se for menor acrescentamos zeros, se for for igual já estará ok, e se for maior haverá um truncamento da mantissa, com arredondamento do seu dígito menos significativo. No caso do exemplo, o número de dígitos da mantissa (4) é menor do que a precisão (5) do sistema, então teremos de acrescentar um zero à direita.

$(-0,36250 \times 10^{1})$

Exemplos

2) Escrever o número $(-3{,}625)_{10}$ no sistema F(2,5,-3,3)

1º passo: devemos primeiramente transformar o número da sua base 10 para a base 2, que é a base de numeração do sistema

$(-0{,}3625)_{10} = (-11{,}101)_2$

2º passo: colocar o número na forma normalizada

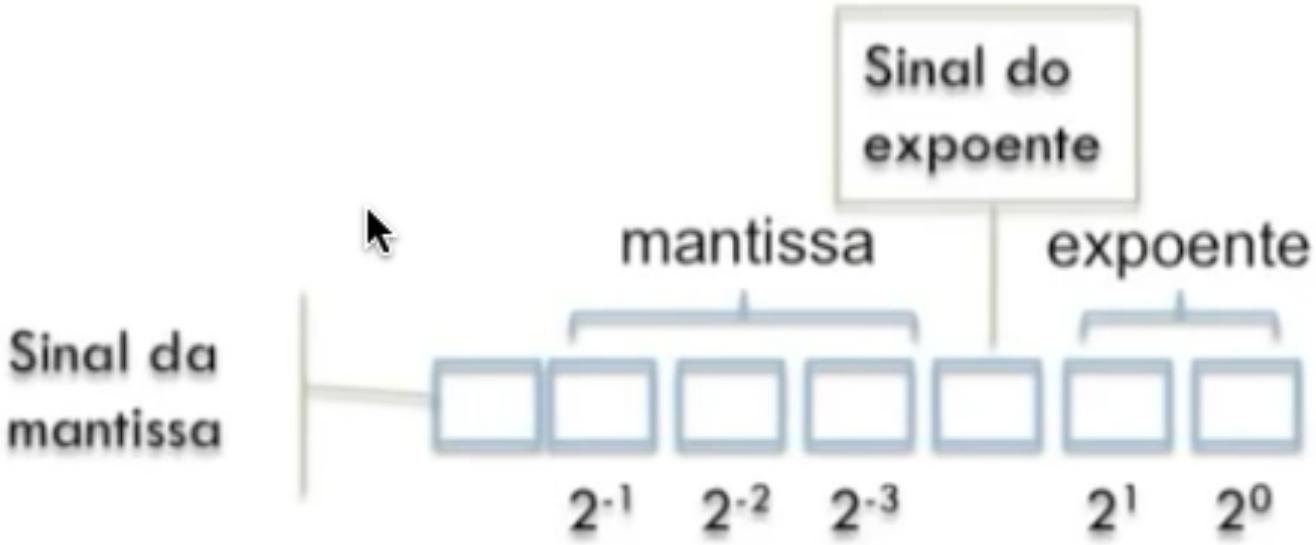
$(-0{,}11101 \times 10^2)$

3º passo: verificar se o número de dígitos na mantissa é menor, igual ou maior do que a precisão do sistema de ponto flutuante. Neste exemplo, o número de dígitos da mantissa (5) é exatamente igual à precisão (5) do sistema, então o número já está perfeitamente representado neste sistema de ponto flutuante

R: $(-0{,}11101 \times 10^2)$

Armazenamento de Números em PF na Memória

- A estratégia de armazenamento mais usada é a IEEE 754.
- Como exemplo, dado o sistema de ponto flutuante F(2,3,-3,3), quantos bits são necessários para armazenar os números deste sistema na memória do computador?



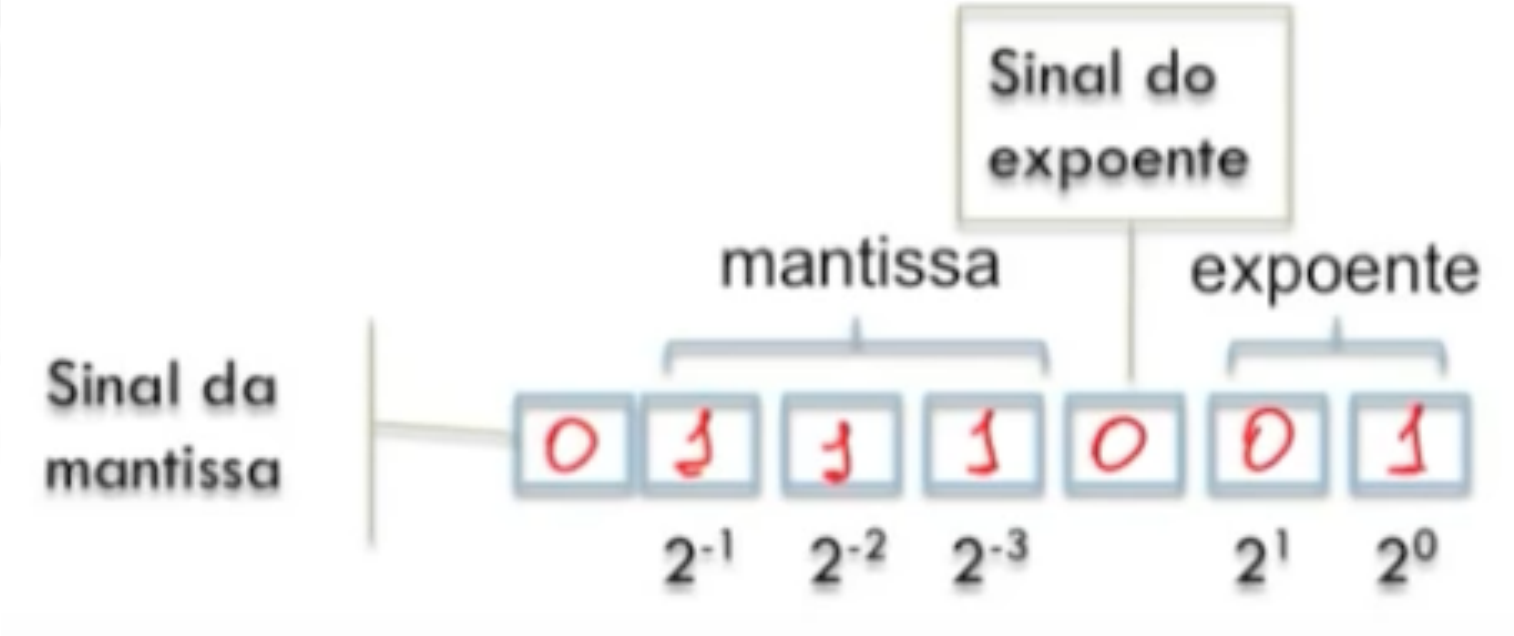
Exemplo

- Como é representado o número $(1,75)_{10}$ na memória de um computador que trabalha com o sistema de ponto flutuante F(2,3,-3,3)?
- 1º passo: devemos converter o número $(1,75)_{10}$ para a base 2, que é a base usada pelo sistema

  $(1,75)_{10} = (1,11)_2$
- 2º passo: colocar o resultado na forma normalizada:

  $(0,111 \times 2^1)$
- 3º passo: observar o número de dígitos da mantissa (3) e comparar com a precisão do sistema (3). Como são iguais não precisamos fazer nada, já que temos exatamente 3 dígitos na mantissa.
- 4º passo: observar se o expoente está dentro do intervalo (-3,+3) do sistema, o que é o caso. O número $(0,111 \times 2^1)$ está então perfeitamente representado no sistema F.

Exemplo



Representação em memória do número $(1{,}75)_{10} = (0{,}111 \times 2^{1})_{2}$

Números Máximos e Mínimos em um Sistema de PF

1) Qual é o maior número representável no sistema F(2,3,-3,3)?

$$0{,}111 \times 2^3 = (2^{-1} + 2^{-2} + 2^{-3}) \times 2^3 = (2^2 + 2^1 + 2^0) = (4+2+1) = 7$$

2) Qual é o menor número representável no sistema F(2,3,-3,3)?

$$0{,}100 \times 2^{-3} = (2^{-1}) \times 2^{-3} = 2^{-4} = 1/16 = 0{,}062510$$

Fórmula Geral

Menor real positivo representado em $F(\beta, p, m, M)$

$$x_m = 0,\underbrace{10\ldots0}_{p \text{ dígitos}} \times \beta^m$$
$$= \beta^{m-1}$$

Maior ~~Menor~~ real positivo representado em $F(\beta, p, m, M)$

$$x_M = 0,\underbrace{(\beta-1)(\beta-1)\ldots(\beta-1)}_{p \text{ dígitos}} \times \beta^M$$
$$= (1-\beta^{-p})\beta^M$$

Exemplo 1

Representar $(8,25)_{10}$ em F(2,3,-3,3).

1. Convertendo para a base 2: (8,25) = 1000, 01
2. Normalizando: 0,100001 x 2^4
3. Acertando a mantissa: 0,100 x 2^4

(o tamanho da mantissa [6] é maior do que a precisão do sistema [3], então ocorre o truncamento. Como o quarto dígito da mantissa é zero, não há necessidade de arredondamento).

- Expoente: o valor do expoente do número normalizado (4) é maior do que o maior expoente do sistema (3), ou seja, o número $(8,25)_{10}$ não pode ser armazenado neste sistema já que ele é maior do que o maior número positivo suportado pelo sistema.
- Conclusão: OVERFLOW

Exemplo 2

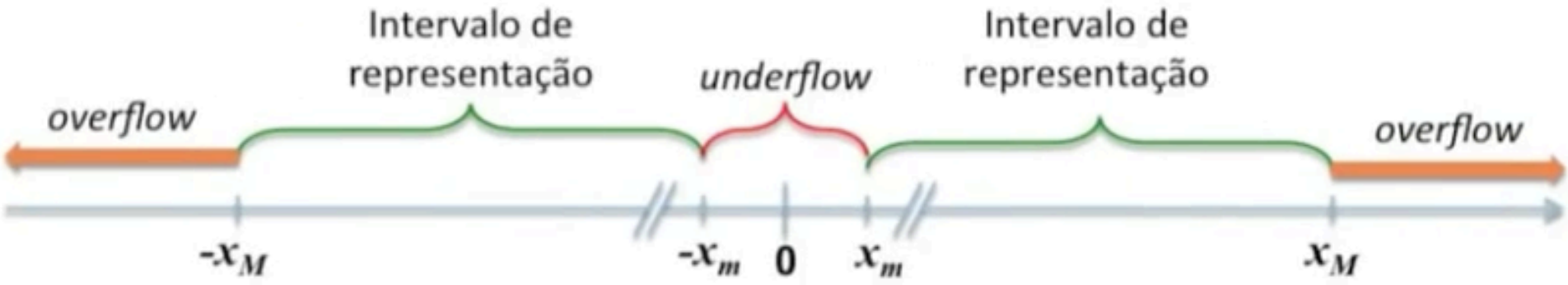
Representar $(0,04)_{10}$ em F(2,3,-3,3).

- Convertendo para a base 2: (0,04) = 0, 0000101000...
- Normalizando: $0,101000 \times 2^{-4}$
- Acertando a mantissa: $0,101 \times 2^{-4}$

(como o tamanho da mantissa é maior do que a precisão do sistema [3], ocorre o truncamento. O quarto dígito da mantissa é zero, então não há arredondamento).

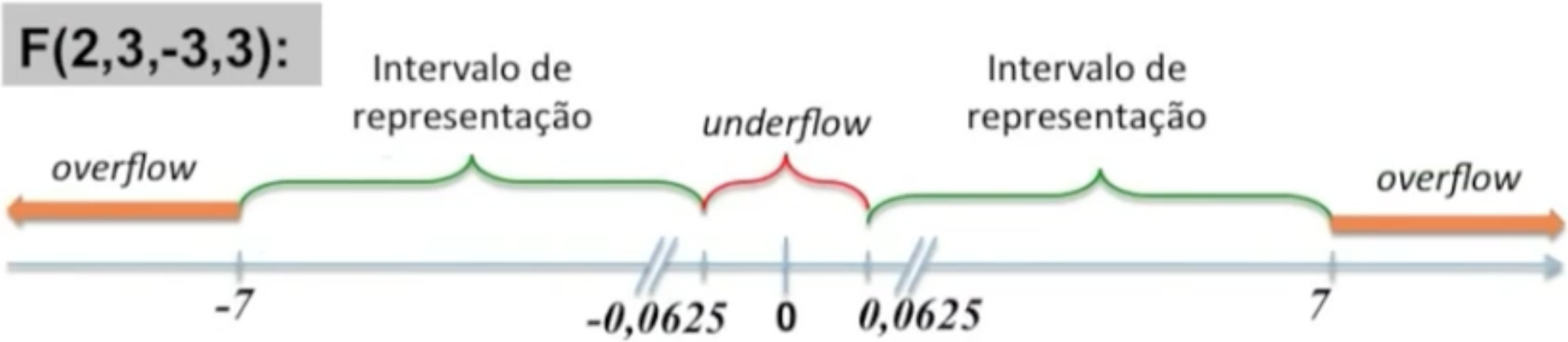
- Expoente: o valor do expoente do número normalizado (-4) é menor do que o menor expoente do sistema (-3); por isso, o número $(0,04)_{10}$ não pode ser armazenado neste sistema.
- Conclusão: UNDERFLOW!!

Intervalos de Representação em PF



Overflow: expoente maior que o expoente máximo
Underflow: expoente menor que o expoente mínimo

Exemplo



Maior número representável no sistema F(2,3,-3,3)?

$0{,}111 \times 2^3 = (2^{-1} + 2^{-2} + 2^{-3}) \times 2^3 = (4+2+1) = 7_{10}$

Menor número representável no sistema F(2,3,-3,3)?

$0{,}100 \times 2^{-3} = (2^{-1}) \times 2^{-3} = 2^{-4} = (4+2+1) = 0{,}0625_{10}$

Erros de Arredondamento

Representar $(2,8)_{10}$ em F(2,3,-3,3).

- Convertendo para a base 2: $(2,8)_{10} = 10,110011001100\ldots$
- Normalizando: $0,1011001100\ldots \times 2^2$
- Acertando a mantissa: ao se truncar a mantissa no terceiro dígito observa-se que quarto dígito da mantissa é 1, então tem que haver um arredondamento.
- Arredondando: $0,1011001100\ldots \times 2^2 = 0,110 \times 2^2$
- Expoente: o valor do expoente do número normalizado (2) é menor do que o maior expoente positivo do sistema (+3).
- Conclusão: o número $(2,8)_{10}$ pode ser armazenado neste sistema de PF, observando-se que ocorre um erro de arredondamento.

Erros de Arredondamento (cont.)

- Convertendo o número $(0{,}110 \times 2^2)$ para a base 10, obtemos: $(1x2^{-1} + 1x2^{-2}) \times 2^2 = (2+1) = 3$
- Assim, o número real 2,8 é representado pelo número inteiro 3 no sistema de ponto flutuante F(2,3,-3,3). Ocorre, portanto, um erro de arredondamento.
- Podemos quantificar este erro de arredondamento através do cálculo do *Erro Relativo*.

$$Er = \left| \frac{\text{valor verdadeiro - valor representado}}{\text{valor verdadeiro}} \right|$$

Erro Relativo

$$Er = \left| \frac{\text{valor verdadeiro - valor representado}}{\text{valor verdadeiro}} \right|$$

$Er = |(2{,}8 - 3) / 2{,}8| = 0{,}0714\ldots = 0{,}07 = 7\%$

Epsilon ("ε") do Sistema

- Dado um sistema de ponto flutuante qualquer, qual é o erro relativo máximo (epsilon "ε" do sistema) que podemos ter?

Epsilon (ε) de um sistema em ponto flutuante

$$F(\beta, p, m, M)$$

$$\epsilon = \frac{\beta^{1-p}}{2}$$

Para o sistema *F*(2,3,-3,3):

$$\epsilon = \frac{2^{1-3}}{2} = 0,125$$

Aritmética em Ponto Flutuante

- Multiplicação:
  - Multiplica-se as mantissas e somam-se os expoentes
- Divisão:
  - Divide-se as mantissas e diminuem-se os expoentes
- Calcular $(0{,}2135 \times 10^{2}) \times (0{,}3064 \times 10^{-2})$ em F(10, 4, -7, 7)
  - Multiplicando: $(0{,}2135) \times (0{,}3064) \times (10^{2-2}) = 0{,}0654164 \times 10^{0}$
  - Normalizando: $0{,}654164 \times 10^{-1}$
  - Ajustando a precisão para 4 dígitos: $0{,}6542 \times 10^{-1}$
  - Observe que houve a necessidade de arredondamento
  - Resultado final: $0{,}6542 \times 10^{-1}$

Aritmética em Ponto Flutuante (cont.)

- Soma e Subtração:
  - Igualam-se os expoentes (i.e. iguala-se o valor do expoente menor ao maior)
  - Soma-se/subtrai-se as mantissas
- Calcular $(0{,}1101 \times 2^1) + (0{,}1010 \times 2^0)$ em F(2, 4, -3, 3)
  - Igualando-se o menor expoente ao maior: $(0{,}1010 \times 2^0) = (0, 01010 \times 2^1)$
  - Somando-se as mantissas: : (0,1101) + (0,01010) = 0,1110010
  - Resultado: $(1{,}00100 \times 2^1)$
  - Normalizando o resultado: $(0{,}100100 \times 2^2)$
  - Ajustando a precisão para 4 dígitos: $(0{,}1001 \times 2^2)$
  - Não houve necessidade de arredondamento
  - Resultado final: $(0{,}1001 \times 2^2)$

Operações Críticas

- Adição e Subtração de Números com Ordens de Grandezas Muito Diferentes
  - Nestes casos, geralmente o menor número perde muita precisão pois nesta situação seus dígitos menos podem ser descartados durante a operação.
- Calcular $(0{,}1101 \times 2^{2}) + (0{,}1010 \times 2^{-1})$ em F(2, 4, -3, 3)
  - Igualando-se os expoentes: $(0{,}1010 \times 2^{-1}) = (0{,}0001010 \times 2^{2})$
  - Somando-se as mantissas: : (0,1101) + (0,0001010) = 0,11100010
  - Resultado: $0{,}11100010 \times 2^{2}$ (já está normalizado)
  - Ajustando a precisão para 4 dígitos: $(0{,}1110 \times 2^{2})$
  - Observe que ao ajustar a mantissa para 4 dígitos, a contribuição do segundo número para a soma foi apenas 0,0001, descartando-se os dígitos 010 menos signigicativos.

Operações Críticas (cont.)

- Subtração de Números Quase Iguais
  - Nestes casos, geralmente o menor número perde muita precisão pois nesta situação seus dígitos menos podem ser descartados durante a operação.
- Calcular $(0{,}1011 \times 2^{-1}) - (0{,}1010 \times 2^{-1})$ em F(2, 4, -3, 3)
  - Expoentes já são iguais
  - Subtraindo-se as mantissas: : (0,1011) - (0,1010) = 0,0001
  - Resultado: $(0{,}0001) \times 2^{-1}$
  - Normalizando: $(0{,}1 \times 2^{-3}) \times 2^{-1} = 0{,}1 \times 2^{-4}$
  - Ajustando a precisão para 4 dígitos: $(0{,}1000 \times 2^{-4})$
  - Observe que o expoente (-4) é menor do que o expoente mínimo do sistema (-3).
  - Conclusão: UNDERFLOW!!