Universidade Federal de Pelotas
Instituto de Física e Matemática
Departamento de Informática
Bacharelado em Ciência da Computação

Introdução à Ciência da Computação

Aula 21

Representação de Números em Ponto Flutuante e o Padrão IEEE 754

Prof. José Luís Güntzel

guntzel@ufpel.edu.br

www.ufpel.edu.br/~guntzel/ICC/ICC.html

Representação de Números em Ponto Flutuante

Números Fracionários

Além de inteiros com e sem sinal, as linguagens de programação suportam também números fracionários.

Exemplos:

$3{,}14159265\ldots_{10}$ (π)

$2{,}71828\ldots_{10}$ (e)

$0{,}000000001_{10} = 1{,}0_{10} \times 10^{-9}$ (segundos em um nanossegundo)

$3.155.760.000_{10} = 3{,}15576_{10} \times 10^{9}$ (segundos em um século)

Notação científica normalizada

Representação de Números em Ponto Flutuante

Números Fracionários

Exemplo de Número em Notação Científica Normalizada:

$1,0_{10} \times 10^{-9}$

Exemplos de Número em Notação Científica Não-Normalizada:

$0,1_{10} \times 10^{-8}$

$10,0_{10} \times 10^{-10}$

Representação de Números em Ponto Flutuante

Números Fracionários

Números binários também podem ser representados em Notação Científica. Exemplo:

$1,0_2 \times 2^{-1}$

Para manter um número binário normalizado deve-sem ultiplicá-lo por uma base tal que ele fique com um dígito diferente de "0" imediatamente à direita da vírgula.

Representação de Números em Ponto Flutuante

Binários em Notação Científica

O Formato utilizado é:

$$1{,}xxxxxxxxx_2 \times 2^{yyyy}$$

Muitas vezes, o expoente é mostrado em decimal, embora na máquina, fique armazenado em binário…

Representação de Números em Ponto Flutuante

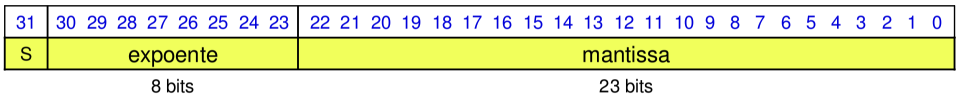
Vantagens da Notação Científica Padronizada

1. **Facilita a troca de dados entre diferentes computadores**
2. **Facilita os algoritmos aritméticos envolvendo números em ponto flutuante**
3. **Melhora a precisão dos números armazenados em uma palavra de memória** (pois os algarismos não-significativos à direita da vírgula são substituídos por algarismos significativos)

Representação de Números em Ponto Flutuante

Formato para Precisão Simples

Os números em ponto flutuante normalmente são representados como múltiplos do tamanho da palavra da máquia. Exemplo:



$$(-1)^S \times F \times 2^E$$

Onde:

- **F representa o valor do campo da mantissa**
- **E representa o valor do campo do expoente**

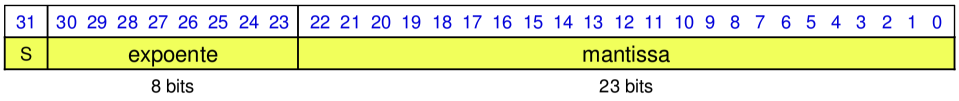
Representação de Números em Ponto Flutuante

Intervalo de Representação

- **O número de bits da palavra da máquina é fixo** (definido quando de seu projeto)
- **Um bit a mais na mantissa corresponde a um bit a menos no expoente e *vice-versa***
- **Mais bits na mantissa => maior precisão do número** ("passo")
- **Mais bits no expoente => maior intervalo de representação**

Representação de Números em Ponto Flutuante

Formato para Precisão Simples



Exemplos de números que podem ser representados:

$2{,}0_{10} \times 10^{-38}$

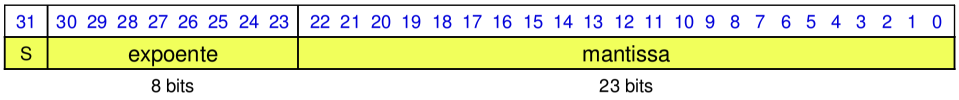
$2{,}0_{10} \times 10^{38}$

O Que significa *overflow*, no caso desta representação?

Significa que o expoente é muito grande (em módulo) para ser armazenado no campo a ele reservado (8 bits, para o caso acima)

Representação de Números em Ponto Flutuante

Formato para Precisão Simples



E o que acontece se o número fracionário tiver muitos dígitos à direita da vírgula, de modo que não pode ser representado no campo da mantissa?

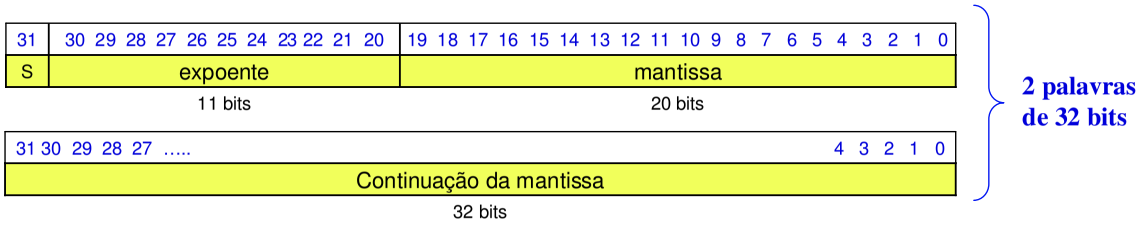
Exemplo: $11111111111111111111111_2$ **(24 "1"s)**

- **Este "evento exepcional" é denominado de *underflow***
- **Exige que o número seja truncado ou arredondado**

Representação de Números em Ponto Flutuante

Formato para Precisão Dupla

Reduzindo as chances de ocorrência de *overflow* e de *underflow*



$$(-1)^S \times F \times 2^E$$

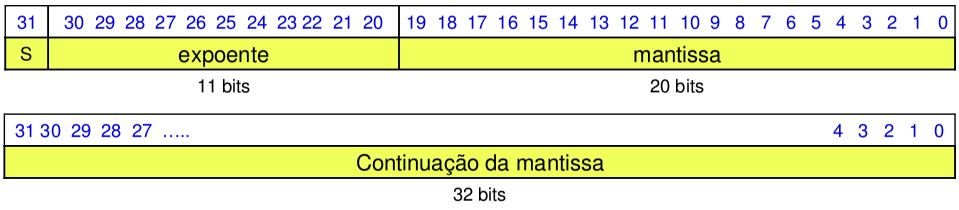
Onde:

- **F representa o valor do campo da mantissa**
- **E representa o valor do campo do expoente**

Representação de Números em Ponto Flutuante

Formato para Precisão Dupla

Reduzindo as chances de ocorrência de *overflow* e de *underflow*



Exemplos de números que podem ser representados:

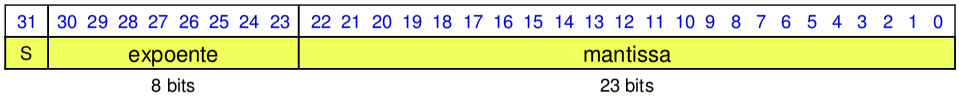
$2{,}0_{10} \times 10^{-308}$

$2{,}0_{10} \times 10^{308}$

Representação de Números em Ponto Flutuante

Padrão IEEE 754

Adotado pelos fabricantes de processadores a partir de 1980



- **Valor "1" à esquerda da vírgula está implícito** (assim, a mantissa fica com 24 bits na precisão simples e 53 bits na precisão dupla)
- **Quando o expoente valer 0, o hardware desconsidera o "1" implícito** ($000\ldots00_2$ representa o valor zero)
- **Os demais números são representados por:**

$$(-1)^S \times (1 + \text{matissa}) \times 2^E$$

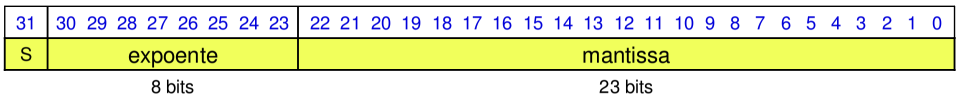
Representação de Números em Ponto Flutuante

Padrão IEEE 754

- **Facilitou a portabilidade dos programas que trabalham com números em ponto flutuante**
- **Melhorou a qualidade das operações aritméticas realizadas pelos computadores**

Representação de Números em Ponto Flutuante

Padrão IEEE 754



- **Enumerando os bits da mantissa da esquerda para a direita por m_1, m_2, m_3 , …, então o valor do número será dado por:**

$$(-1)^S \times (1 + (m_1 \times 2^{-1}) + (m_2 \times 2^{-2}) + (m_3 \times 2^{-3}) + \ldots) \times 2^E$$

Representação de Números em Ponto Flutuante

Padrão IEEE 754

Características Marcantes do Padrão:

- **Bit de Sinal aparece mais à esquerda (uso de sinal-magnitude)**
- **O expoente aparece à esquerda da mantissa**
- **Os bits da mantissa aparecem em ordem inversa**

Motivos para estas características:

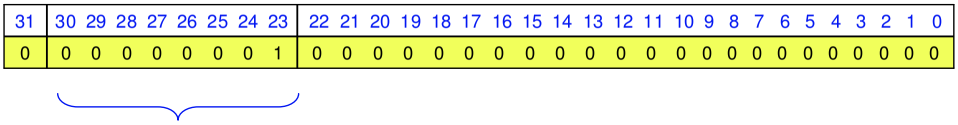
- **Permitir o processamento rápido de números por meio de instruções de comparações inteiras (inclusive para o ordenamento de números)**

Representação de Números em Ponto Flutuante

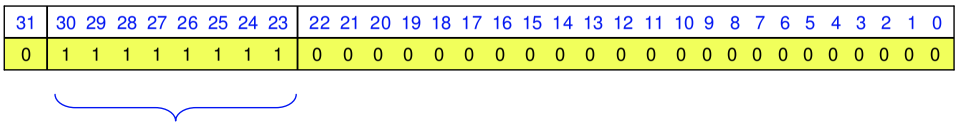
Padrão IEEE 754

Problema: como comparar os expoentes?

Exemplo, usando complemento de dois:



expoente = +1



expoente = -1

Representação de Números em Ponto Flutuante

Padrão IEEE 754

Solução: aplicar um deslocamento positivo, de modo que o menor expoente (i.e., o mais negativo) seja representado por zero

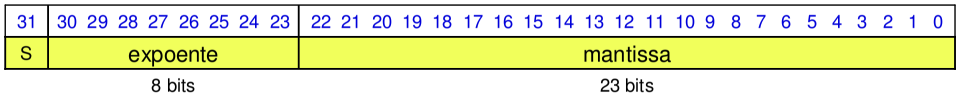
Isto é chamado de notação com peso ou notação em excesso de (valor)

Padrão IEEE 754 usa excesso de 127:

- **Expoente -1 => $-1 + 127_{10} = 126_{10} => 0111\ 1110_2$**
- **Expoente +1 => $+1 + 127_{10} = 128_{10} => 1000\ 0000_2$**

Representação de Números em Ponto Flutuante

Padrão IEEE 754



$$(-1)^S \times (1 + \text{matissa}) \times 2^{(E - \text{excesso})}$$

- Para precisão simples, expoente em excesso de 127
- Para dupla precisão, expoente em excesso de 1023