Aritmética Computacional - Ponto Flutuante

Arquitetura e Organização de Computadores

Prof. Lucas de Oliveira Teixeira

UEM

Introdução

Introdução

- A representação de números de ponto flutuante no computador é mais complexo do que números inteiros.
- Isso acontece porque números de ponto flutuante são números reais e existe uma imprecisão natural com esse tipo de dado.

Representação de ponto flutuante

Representação de ponto flutuante

- O computador utiliza notação científica para representar números de ponto flutuante no formato:

$$\pm Mantissa \times Base^{\pm Expoente}$$

Representação de ponto flutuante

- Por exemplo, o número

$$7\ 452\ 000\ 000\ 000\ 000\ 000\ 000.00$$

  pode ser representado em notação científica como:

$$7.452 \times 10^{21}$$

- Assim, o mantissa é 7.452, a base é 10 e o expoente é 21.

Representação de ponto flutuante

- Por exemplo, o número

$$0.0000000000000232$$

pode ser representado em notação científica como:

$$2.32 \times 10^{-14}$$

- Assim, o mantissa é 2.32, a base é 10 e o expoente é -14.

Representação de ponto flutuante

- Por exemplo, o número

$$1110100000_2$$

pode ser representado em notação científica como:

$$1.1101 \times 2^9$$

- Assim, o mantissa é 1.1101, a base é 2 e o expoente é 9.

Representação de ponto flutuante

- Por exemplo, o número

$$0.00000000000000000000000101_2$$

pode ser representado em notação científica como:

$$1.01 \times 2^{-24}$$

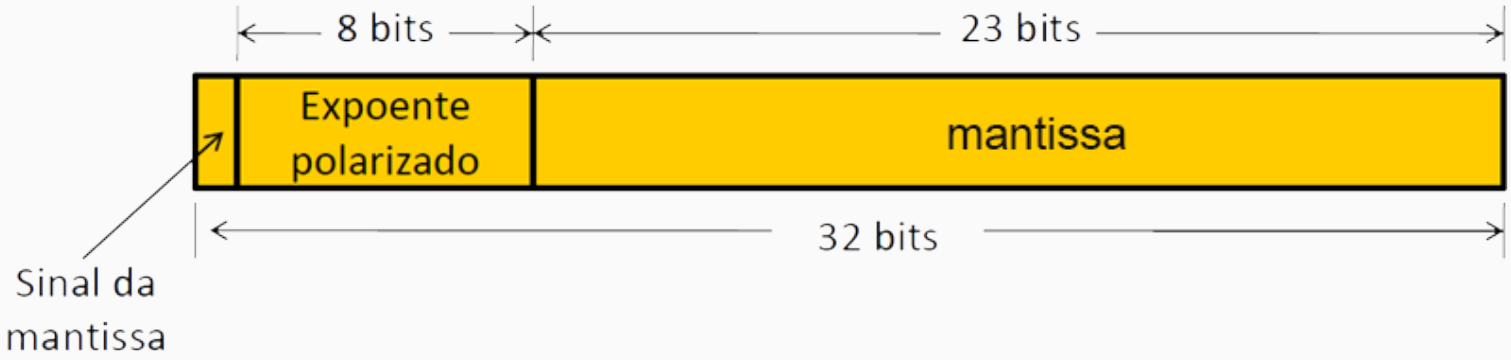
- Assim, o mantissa é 1.01, a base é 2 e o expoente é -24.

Representação de ponto flutuante

- Utilizar a notação científica possui algumas vantagens:
  - Números muito grandes ou muito pequenos podem ser representados usando poucos bits.
  - Não é necessário armazenar a base, ela é sempre dois.
  - Não é necessário armazenar o bit à esquerda da vírgula, ele é sempre um.

Representação de ponto flutuante

Padrão IEEE 754:



Representação de ponto flutuante

Expoente polarizado:

- O expoente é um número inteiro positivo ou negativo.
- Ele poderia ser armazenado utilizando complemento de dois, mas exige um passo adicional ao usar o valor representado.
- Com isso, o expoente é polarizado, o valor 127 é somado ao expoente, com isso ele sempre será positivo e não precisamos nos preocupar com o sinal.

Representação de ponto flutuante

Por exemplo, represente o número 5.75 no padrão IEEE 754 de 32 bits:

- O sinal é positivo, então sinal $= 0$.
- O decimal 5.75 corresponde à $(101.11)_2$.
- Normalização: 1.0111×2^2, assim o mantissa $= 0111$ (é necessário acrescentar 19 zeros para obter 23 bits).
- Expoente: 2 + 127 = 129 = (10000001).

Número	Sinal	Expoente	Mantissa
5.75	0	1000 0001	0111 0000 0000 0000 0000 000

Representação de ponto flutuante

Por exemplo, represente o número -161.875 no padrão IEEE 754 de 32 bits:

- O sinal é negativo, então sinal $= 1$.
- O decimal 5.75 corresponde à $(10100001.111)_2$.
- Normalização: 1.0100001111×2^7, assim o mantissa $= 0100001111$ (é necessário acrescentar zeros até atingir 23 bits).
- Expoente: 7 + 127 = 134 = (10000110).

Número	Sinal	Expoente	Mantissa
-161.875	1	1000 0110	0100 0011 1100 0000 0000 000

Representação de ponto flutuante

Intervalos de representações com 32 bits:

- Números negativos:
  - $[-(2 - 2^{-23}) \times 2^{128}, -2 \times 2{-}127]$
- Números positivos:
  - $[2^{-127}, (2 - 2^{-23}) \times 2^{128}]$

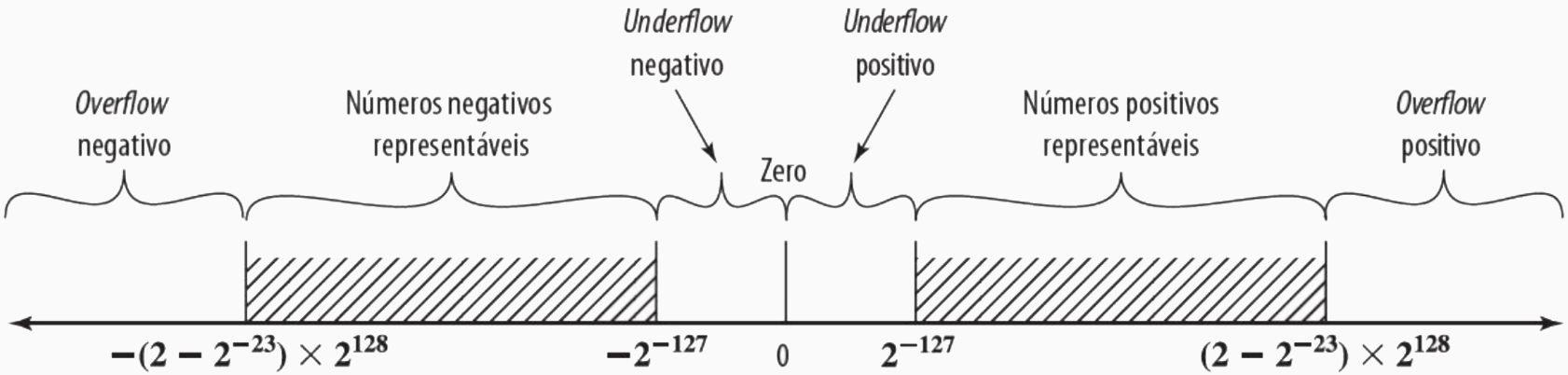
Representação de ponto flutuante

Intervalos de representações com 32 bits:

- Cinco regiões não estão incluídas na representação:
  - Números menores que $-(2 - 2^{-23}) \times 2^{128}$: overflow negativo.
  - Números maiores que $-2 \times 2{-}127$ e maiores que zero: underflow negativo.
  - Zero.
  - Números maiores que 0 e menores que 2^{-127}: underflow positivo.
  - Números maiores $(2 - 2^{-23}) \times 2^{128}$: overflow positivo.

Representação de ponto flutuante

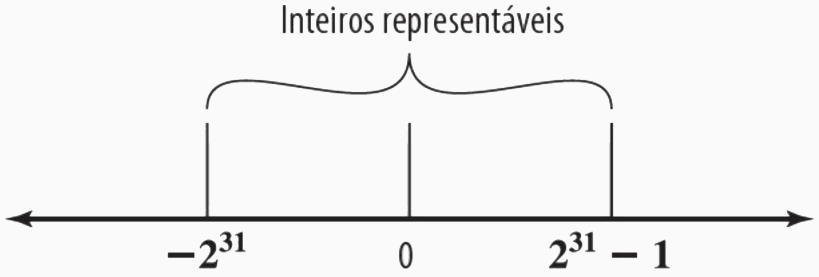
Intervalos de representações com 32 bits:



Representação de ponto flutuante

Intervalos de representações com 32 bits:

- A quantidade de números representáveis é o mesmo dos inteiros usando complemento de dois com 32 bits.



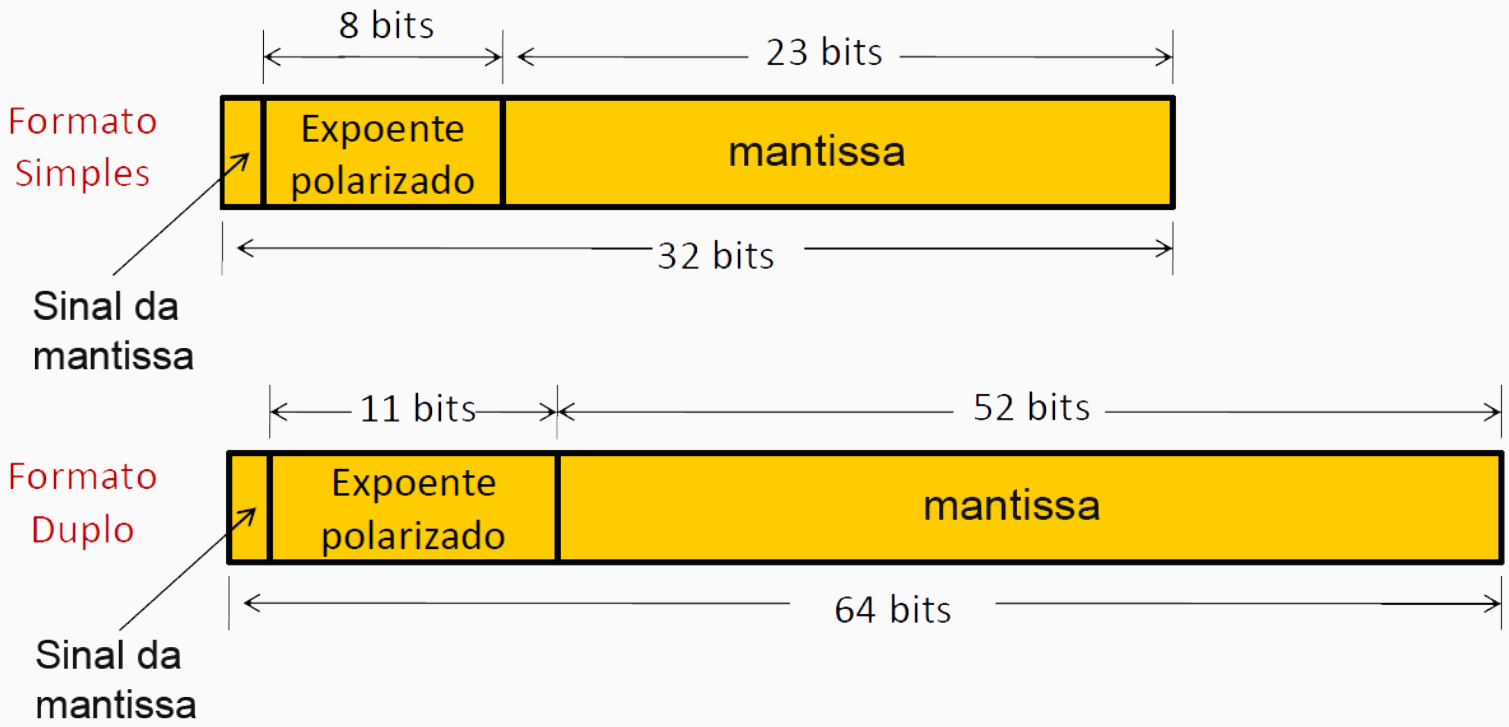
- A diferença é que a representação de ponto flutuante espalha mais os números e com isso pode representar valores maiores.
- Esse espalhamento maior de valores acontece por causa do expoente.
- Porém, esse espalhamento gera problemas de precisão.

Representação de ponto flutuante

- Existe uma relação estreita entre os tamanhos dos campos reservados ao mantissa e ao expoente.
- Se o número de bits do mantissa aumentar:
  - Maior precisão.
  - Menor faixa de valores representáveis.
- Se o número de bits do expoente aumentar:
  - Menor precisão.
  - Maior faixa de valores representáveis.

Representação de ponto flutuante

Padrão IEEE 754 com 32 (float) e 64 (double) bits:



Representação de ponto flutuante

Parâmetros do padrão IEEE 754:

Parâmetro	Formato simples	Formato duplo
Tamanho da palavra	32	64
Tamanho do expoente	8	11
Polarização do expoente	127	1023
Expoente máximo	127	1023
Expoente mínimo	-126	-1022
Tamanho da mantissa	23	52
Número de expoentes	254	2046

Representação de ponto flutuante

Valores especiais do padrão IEEE 754:

Sinal	Expoente simples	Expoente duplo	Mantissa	Valor
0	0	0	0	0
1	0	0	0	-0
0	255	2047	0	∞
1	255	2047	0	$-\infty$
0 ou 1	255	2047	$\neq 0$	NaN

Aritmética com ponto flutuante

Aritmética com ponto flutuante

- As operações de soma e subtração em ponto flutuante são mais complexas que divisão e multiplicação.
- Como se trabalha com base e expoente, é necessário que ambos estejam com o mesmo expoente para realizar tais operações.
- Assim, normalmente é necessário alinhar os expoentes dos dois valores.

Aritmética com ponto flutuante

Problemas com a aritmética de ponto flutuante:

- Overflow de expoente: expoente positivo que excede o valor máximo; em alguns casos é designado como ∞ ou $-\infty$.
- Underflow de expoente: expoente negativo menor que o valor mínimo, é um número muito pequeno; pode ser informado como 0.

Aritmética com ponto flutuante

Problemas com a aritmética de ponto flutuante:

- Overflow de mantissa: pode ocorrer um carry pelo bit mais significativo, é necessário realinhar o número.
- Underflow de mantissa: podem ser perdidos dígitos pela extremidade da direita, é necessária arredondar o número.

Aritmética com ponto flutuante

Passos para adição e subtração:

- Verificação de zero.
- Alinhamento das mantissas (ajustando expoentes).
- Adição ou subtração das mantissas.
- Normalização do resultado.

Aritmética com ponto flutuante

Passo 1 - verificação de zero:

- A adição e a subtração são idênticas, exceto por uma mudança de sinal
- Se for uma operação de subtração, o processo começa alterando o sinal do subtraendo.
- Em seguida, se algum operando for 0, o outro é informado como o resultado.

Aritmética com ponto flutuante

Passo 2 - alinhamento de mantissa:

- A próxima fase é manipular os números de modo que os dois expoentes sejam iguais.
- O alinhamento é obtido deslocando repetidamente a parte de magnitude do mantissa 1 dígito para a direita, e aumentando o expoente até que os dois expoentes sejam iguais.
- Se esse processo resultar em um valor 0 para o mantissa, então o outro número é informado como resultado

Aritmética com ponto flutuante

Passo 2 - alinhamento de mantissa:

- A próxima fase é manipular os números de modo que os dois expoentes sejam iguais.
- O alinhamento é obtido deslocando repetidamente a parte de magnitude do mantissa 1 dígito para a direita, e aumentando o expoente até que os dois expoentes sejam iguais.
- Se esse processo resultar em um valor 0 para o mantissa, então o outro número é informado como resultado

Aritmética com ponto flutuante

Passo 3 - adição ou subtração das mantissas:

- As duas mantissas são somadas ou subtraídas.

Aritmética com ponto flutuante

Passo 4 - normalização do resultado:

- A fase final normaliza o resultado.
- Consiste no deslocamento dos dígitos do significando para a esquerda até que o dígito mais significativo seja diferente de zero.
- Cada deslocamento causa um decremento do expoente e, portanto, poderá ocasionar um underflow do expoente.
- Finalmente, o resultado poderá ser arredondado e depois informado.

Aritmética com ponto flutuante

Por exemplo, vamos realizar a operação: (+37) + (-4,5) na representação definida pelo padrão IEEE 754 de 32 bits:

Número	Sinal	Expoente	Mantissa
$+37$	0	1000 0100	0010 1000 0000 0000 0000 000
-4.5	1	1000 0001	0010 0000 0000 0000 0000 000

Aritmética com ponto flutuante

Por exemplo, vamos realizar a operação: (+37) + (-4,5) na representação definida pelo padrão IEEE 754 de 32 bits:

- Verificação de zero: nenhum dos dois é zero.

Aritmética com ponto flutuante

Por exemplo, vamos realizar a operação: (+37) + (-4,5) na representação definida pelo padrão IEEE 754 de 32 bits:

- Alinhamento das mantissas (ajustando expoentes): deslocar o múmero de menor expoente até que os expoentes fiquem iguais.

Sinal	Expoente	Bit implícito	Mantissa
1	1000 0001	1	0010 0000 0000 0000 0000 000
1	1000 0010	0	1001 0000 0000 0000 0000 000
1	1000 0011	0	0100 1000 0000 0000 0000 000
1	1000 0100	0	0010 0100 0000 0000 0000 000

Aritmética com ponto flutuante

Por exemplo, vamos realizar a operação: (+37) + (-4,5) na representação definida pelo padrão IEEE 754 de 32 bits:

- Adição das mantissas: como os números possuem sinal diferentes, o menor número é subtraído do maior.
- Para isso, o sinal do menor valor é invertido e é realizada a subtração dos números.
- É importante notar que, o primeiro número que não foi deslocado ainda possui o bit implícito 1.

Aritmética com ponto flutuante

Número	Sinal	Expoente	Bit implícito	Mantissa
$+37$	0	1000 0100	1	0010 1000 0000 0000 0000 000
-4.5	0	1000 0100	0	0010 0100 0000 0000 0000 000
32.5	0	1000 0100	1	0000 0100 0000 0000 0000 000

Aritmética com ponto flutuante

Passos para multiplicação e divisão:

- Verifique zero.
- Soma/subtraia expoentes.
- Multiplique/divida significandos (observando sinal).
- Normalize.
- Arredonde.

Aritmética com ponto flutuante

Passos para multiplicação e divisão:

- Não vamos fazer exemplo prático.