

Sistemas Hardware-Software

Aula 2 – Números fracionários

2019 – Engenharia

Igor Montagner, Fábio Ayres

Números fracionários

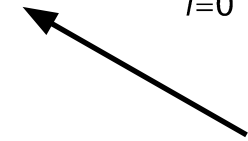
- Representamos inteiros usando bits, sendo que cada posição equivale a uma multiplicação por potência de 2.

Unsigned

$$B2U(X) = \sum_{i=0}^{w-1} x_i \cdot 2^i$$

Two's Complement

$$B2T(X) = -x_{w-1} \cdot 2^{w-1} + \sum_{i=0}^{w-2} x_i \cdot 2^i$$



Bit de sinal

- Como vocês estenderiam isso para representar números fracionários?

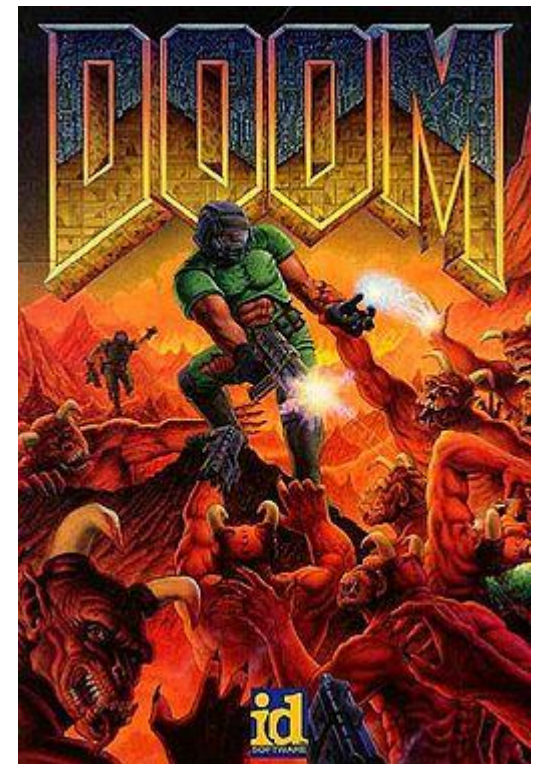
Ponto fixo

Número fixo de bits para a parte decimal

- 16 bits para a parte inteira
- 16 bits para a parte fracionária

Equivalente a dividir a representação binária por uma fator escala!

Expoente é fixo!



Revisão: números decimais fracionários

314.15

$$3 \times 10^2 + 1 \times 10^1 + 4 \times 10^0 + 1 \times 10^{-1} + 5 \times 10^{-2}$$

-314.15

$$-(3 \times 10^2 + 1 \times 10^1 + 4 \times 10^0 + 1 \times 10^{-1} + 5 \times 10^{-2})$$

Números binários fracionários

Já sabemos o que significam **números inteiros** representados em **base 2**:

$$10110_2 = 1 \times 2^4 + \dots + 0 \times 2^0 = 22_{10}$$

$$-1101_2 = -(1 \times 2^3 + \dots + 1 \times 2^0) = -13_{10}$$

Sabemos também o que significam **números fracionários decimais**

O que seriam números fracionários em base 2?

$$11.0110_2 = ?$$

Números binários fracionários

Já sabemos o que significam **números inteiros** representados em **base 2**:

$$10110_2 = 1 \times 2^4 + \dots + 0 \times 2^0 = 22_{10}$$

$$-1101_2 = -(1 \times 2^3 + \dots + 1 \times 2^0) = -13_{10}$$

Sabemos também o que significam **números fracionários decimais**

O que seriam números fracionários em base 2?

$$11.0110_2 = ?$$

Resposta:

$$\begin{aligned} 11.0110_2 &= 1 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-3} + 0 \times 2^{-4} \\ &= 2 + 1 + 0 + 0.25 + 0.125 + 0 \\ &= 3.375_{10} \end{aligned}$$

Exercício

1. Calcule o valor dos seguintes valores binários em decimal:
 - a) 101.11_2
 - b) -0.001_2
 - c) 11.0_2

2. Prove que $0.11111\dots_2 = 1$

Exercício

1. Calcule o valor dos seguintes valores binários em decimal:

- a) 101.11_2 Resposta: 5.75
- b) -0.001_2 Resposta: -0.125
- c) 11.0_2 Resposta: 3

2. Prove que $0.11111\dots_2 = 1$

$$0.111\dots_2 = 2^{-1} + 2^{-2} + 2^{-3} + \dots = \sum_{k=-1}^{-\infty} 2^k = \sum_{i=1}^{+\infty} \left(\frac{1}{2}\right)^i = 1$$

Números decimais em notação científica

$$314.15 = \boxed{3.1415} \times \boxed{10}^2$$

Mantissa Base Expoente

$$\begin{aligned} -314.15 &= -1 \times 3.1415 \times 10^2 \\ &= (-1)^1 \times 3.1415 \times 10^2 \end{aligned}$$

Bit de sinal

Números decimais em notação científica

Todo número real pode ser arredondado para um número real escrito da seguinte forma:

$$x = (-1)^s \times M \times 10^E$$

onde:

- s é o bit de sinal (0: positivo, 1: negativo)
- M é a mantissa
- E é o expoente

Números binários em notação científica

Vamos fazer a mesma coisa que fizemos para os números decimais:

$$x = (-1)^s \times M \times 2^E$$

onde:

- s é o bit de sinal (0: positivo, 1: negativo)
- M é a mantissa
- E é o expoente

Exemplo

$$10.1_2 = (-1)^0 \times 1.01_2 \times 2^1$$

$$s = 0$$

$$M = 1.01_2 = 1.25$$

$$E = 1$$

Exercício

Determine o bit de sinal, a mantissa e o expoente (de base 2) dos seguintes números reais representados em base 2:

$$101.11_2$$

$$-0.001_2$$

$$11.0_2$$

Exercício

Valor	Bit de sinal (s)	Mantissa (M)	Expoente (E)
101.11_2			
-0.001_2			
11.0_2			

Exercício

Valor	Bit de sinal (s)	Mantissa (M)	Expoente (E)
101.11_2 $= (-1)^0 \times 1.0111_2 \times 2^2$			
-0.001_2			
11.0_2			

Exercício

Valor	Bit de sinal (s)	Mantissa (M)	Expoente (E)
101.11_2 $= (-1)^0 \times 1.0111_2 \times 2^2$	0		
-0.001_2			
11.0_2			

Exercício

Valor	Bit de sinal (s)	Mantissa (M)	Expoente (E)
101.11_2 $= (-1)^0 \times 1.0111_2 \times 2^2$	0	1.0111_2 $= 1 + 0.25 + 0.125 + 0.0625$ $= 1.4375$	
-0.001_2			
11.0_2			

Exercício

Valor	Bit de sinal (s)	Mantissa (M)	Expoente (E)
101.11_2 $= (-1)^0 \times 1.0111_2 \times 2^2$	0	1.0111_2 $= 1 + 0.25 + 0.125 + 0.0625$ $= 1.4375$	2
-0.001_2			
11.0_2			

Exercício

Valor	Bit de sinal (s)	Mantissa (M)	Expoente (E)
101.11_2 $= (-1)^0 \times 1.0111_2 \times 2^2$	0	1.0111_2 $= 1 + 0.25 + 0.125 + 0.0625$ $= 1.4375$	2
-0.001_2 $= (-1)^1 \times 1.0_2 \times 2^{-3}$	1	$1.0_2 = 1$	3
11.0_2 $= (-1)^0 \times 1.1_2 \times 2^1$	0	$1.1_2 = 1.5$	1

Representando números reais no computador

- Vamos adotar essa ideia de representação em notação científica binária para armazenar números reais no computador!
 - 1 bit para guardar o sinal
 - Seguido de alguns bits para guardar a mantissa
 - E mais alguns bits para guardar o expoente

Padrão IEEE 754

- No início era a baderna...
 - Cada fabricante de CPU definia seu próprio formato de número em ponto flutuante
- Padrão IEEE 754
 - Surgiu em 1985 para uniformizar operações em **ponto-flutuante**
 - Adotado por todas as CPUs mais importantes do mercado
- Criado principalmente por razões matemáticas
 - Arredondamento, overflow, underflow tem propriedades bem bacanas
 - Mas é difícil de implementar eficientemente em hardware!
 - FPU's

Precisão

- Precisão simples (single precision): 32 bits float



- Precisão dupla (double precision): 64 bits double



- Precisão estendida (extended precision): 80 bits (apenas Intel) long double



Representando valores float

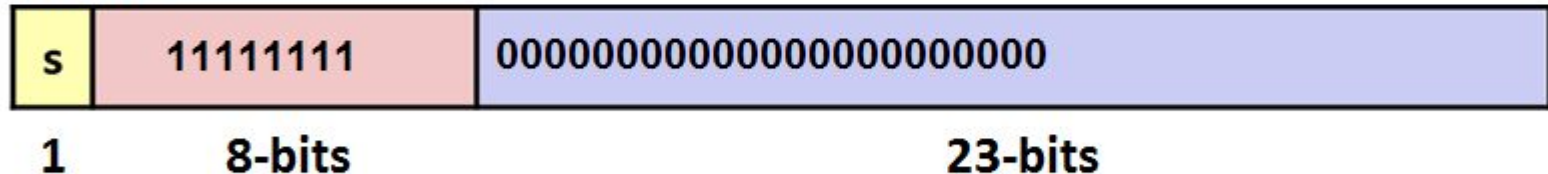
Normalizado



Desnormalizado



Infinito



NaN



Valores normalizados

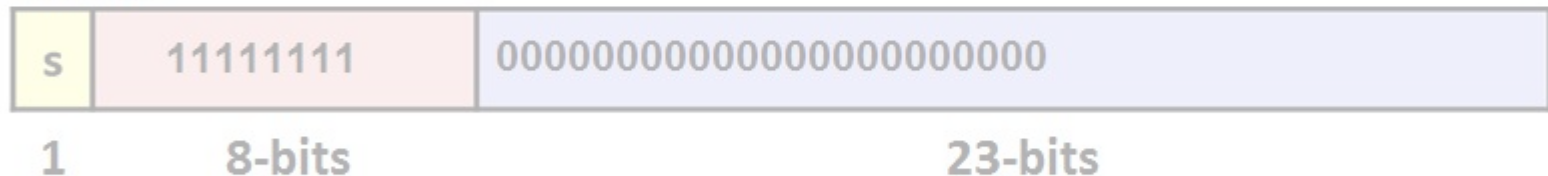
Normalizado



Desnormalizado



Infinito



NaN



Valores normalizados

Normalizado



onde:

- s é o bit de sinal
- $M = 1.\text{frac}$
 - Assuma que já tem um bit 1 implícito lá e que frac representa os bits da parte fracionária
- $E = \text{exp} - \text{Bias}$
 - Tomando exp como inteiro sem sinal, subtraia o valor do bias . Tipo `float`: $\text{bias} = 127$

Aplicável quando $0 < \text{exp} < 255$

- Ou seja, quando $-126 \leq E \leq 127$

Exemplo

Determine os bits de uma variável do tipo float que armazene o número matemático 101.11_2

Resposta:

Passo 1: Lembrando que $101.11_2 = (-1)^0 \times 1.0111_2 \times 2^2$

temos $s = 0$, $M = 1.0111_2$ e $E = 2$

Passo 2: Como $E \geq -126$, vamos usar float normalizado.

Passo 3:

- De $M = 1.0111_2$ temos $\text{frac} = 0111\ 0000\ 0000\ 0000\ 0000\ 0000$ 23 bits
- De $E = 2$ temos $\text{exp} = E + \text{Bias} = 129 = 1000\ 0001$

Resultado final: $01000000101110000000000000000000$

0x 4 0 b 8 0 0 0 0

Exercício

Determine o bit de sinal, a mantissa e o expoente (de base 2) dos seguintes números reais representados em base 2:

$$101.11_2$$

$$-0.001_2$$

$$11.0_2$$

Valores desnormalizados

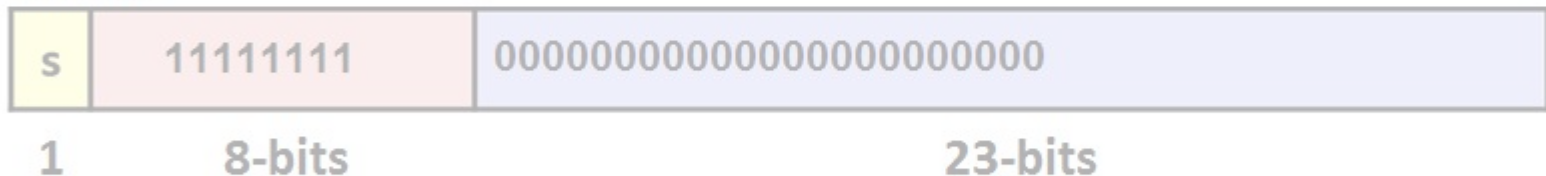
Normalizado



Desnormalizado



Infinito

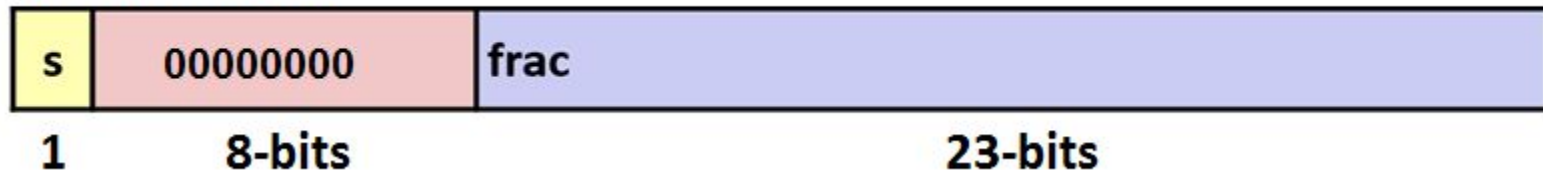


NaN



Valores desnormalizados

Desnormalizado



$$V = (-1)^s \times M \times 2^E$$

onde:

- s é o bit de sinal
- $M = 0.\text{frac}$
 - Assuma que o bit mais alto agora é **zero**. Novamente, frac representa os bits da parte fracionária
- $E = 1 - \text{Bias} = -126$

São os números de magnitude menor que 2^{-126} .

Inclui o valor matemático **zero** (e também o bizarro **menos zero**!)

Valores especiais

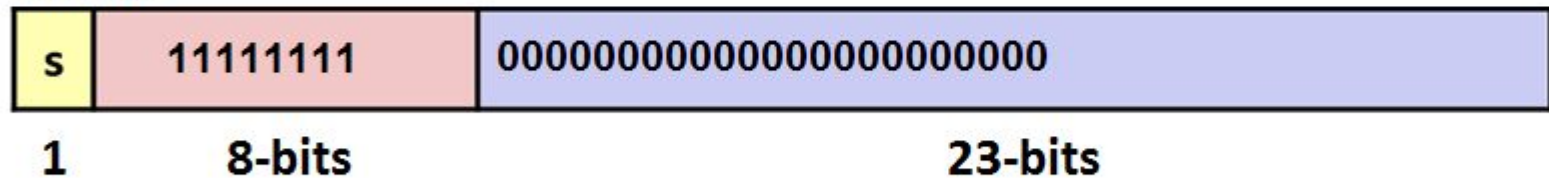
Normalizado



Desnormalizado



Infinito



NaN



Valores especiais

- Infinito
 - Tanto positivo como negativo
 - Aparece em operações que resultam em overflow
- NaN
 - “Not-a-Number”
 - Aparece em operações que não resultam em número real

Exercício

Calcule (na mão) a seguinte soma:

$$9 + 0.9 + \overbrace{0.1 + 0.1 + \dots + 0.1}^{17 \text{ vezes}}$$

Regra: sempre que fizer uma soma, arredonde o resultado para 2 dígitos apenas. Exemplos: $2.34 \rightarrow 2.3$, $10.1 \rightarrow 10$

- a) Some do maior para o menor
- b) Some do menor para o maior

O que aconteceu?

Exercício

Usando Python, teste o seguinte:

```
>> 0.1
```

```
>> format(0.1, '.24f')
```

```
>> 0.9
```

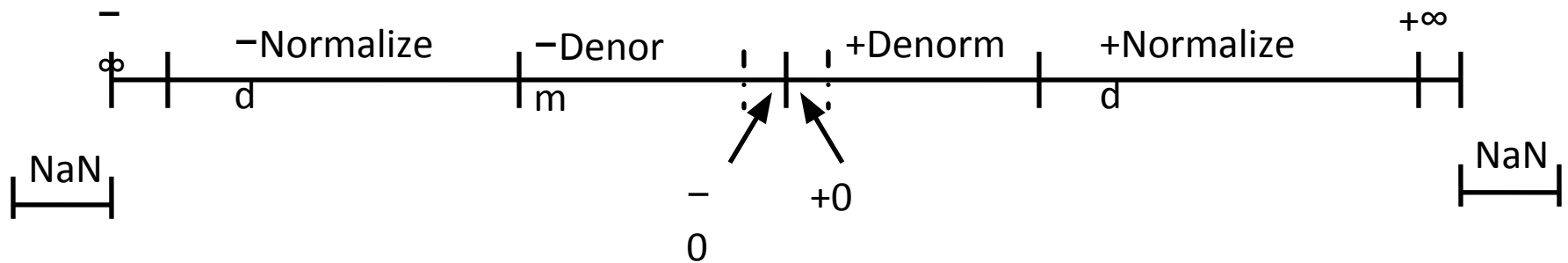
```
>> format(0.9, '.24f')
```

```
>> 0.5
```

```
>> format(0.5, '.24f')
```

Pergunta: todos números são representáveis de maneira exata?

Visualizando a reta numérica



Insper

www.insper.edu.br