Representação em ponto flutante em binário

Algoritmos Numéricos - Topico 1 3
Representação em ponto flutante em binário
Prof. Cláudia G. Varassin DI/UFES
emails:claudia.varassin@ufes.br e galarda@inf.ufes.br

Setembro 2020

Sumário

Representação das informações no computador

O computador armazena informações BEM SIMPLES:

0 ou 1
Ligado ou desligado
Energizado ou NÃO Energizado

Assim, o computador representa tadas as informações usando apenas 0s ou 1s, isto é, emprega a base binária.

Toda informação é uma sequencia de dígitos binários, uma sequencia de BInary digiTS: de BITS.

Sistemas de numeração

Base 10	Base 2
0	
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	

Sistemas de numeração

Base 10	Base 2
0	0
1	1

Sistemas de numeração

Base 10	Base 2
0	0
1	1
2	10
3	11

Sistemas de numeração

Base 10	Base 2
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111

Sistemas de numeração

Base 10	Base 2
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000
9	1001
10	1010
11	1011
12	...

Sistemas de numeração: notação posicional

Base decimal

$$347 = 3 \times 10^2 + 4 \times 10^1 + 7 \times 10^0$$

Sistemas de numeração: notação posicional

Base decimal

$$
\begin{aligned}
347 &= 3 \times 10^2 + 4 \times 10^1 + 7 \times 10^0 \\
d_2 d_1 d_0 . d_{-1} d_{-2} \ldots &= d_2 10^2 + d_1 10^1 + d_0 10^0 + d_{-1} 10^{-1} + d_{-2} 10^{-2} \ldots
\end{aligned}
$$

Sistemas de numeração: notação posicional

$$
\begin{aligned}
&\text{Base decimal} \\
347 &= 3 \times 10^2 + 4 \times 10^1 + 7 \times 10^0 \\
d_2 d_1 d_0 . d_{-1} d_{-2} \ldots &= d_2 10^2 + d_1 10^1 + d_0 10^0 + d_{-1} 10^{-1} + d_{-2} 10^{-2} \ldots \\
&\text{Base binária} \\
(101)_2 &= 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \\
d_2 d_1 d_0 . d_{-1} d_{-2} \ldots &= d_2 2^2 + d_1 2^1 + d_0 2^0 + d_{-1} 2^{-1} + d_{-2} 2^{-2} \ldots
\end{aligned}
$$

Sistemas de numeração: notação posicional

Mais exemplos

Base decimal

$$\begin{aligned} d_2 d_1 d_0 . d_{-1} d_{-2} \ldots &= d_2 10^2 + d_1 10^1 + d_0 10^0 + d_{-1} 10^{-1} + d_{-2} 10^{-2} \ldots \\ -8.25 &= -(8 \times 10^0 + 2 \times 10^{-1} + 5 \times 10^{-2}) \end{aligned}$$

Sistemas de numeração: notação posicional

Mais exemplos

Base decimal

$$d_2 d_1 d_0 . d_{-1} d_{-2} \ldots = d_2 10^2 + d_1 10^1 + d_0 10^0 + d_{-1} 10^{-1} + d_{-2} 10^{-2} \ldots$$

$$-8.25 = -(8 \times 10^0 + 2 \times 10^{-1} + 5 \times 10^{-2})$$

Base binária

$$d_2 d_1 d_0 . d_{-1} d_{-2} \ldots = d_2 2^2 + d_1 2^1 + d_0 2^0 + d_{-1} 2^{-1} + d_{-2} 2^{-2} + \ldots$$

$$(0.11)_2 = 0 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2}$$

Todo valor dado na base decimal é convertido em base binária e seu registro na memória é feito empregando a representação em sistema de ponto flutuante normalizado, isto é, em cada posição da palavra haverá 0s ou 1s.

Um número $x \in \mathbb{R}$ armazenado neste sistema tem seguinte formato:

$$x = \pm 0.d_1 d_2 \cdots d_p \times 2^{\,e}$$

Um sistema de ponto flutuante é representado por

$$F = F(2, p, e_1, e_2),$$

onde $p =$ é a quantidade de dígitos da mantissa.
onde $e_1, e_2 =$ menor e maior expoente possíveis de serem armazenados.

Representação dos números reais

Versão simplificada, acadêmica (*)

**Haverá "espaço" reservado para o sinal,
para o expoente do número, com o seu sinal e
o restante para para armazenar os dígitos do número.**

Sinal	Expoente (com o sinal)	Dígitos do valor

Ex:

$$x = (5)_{10} = (101)_2 = \rightarrow + 0.10100\ldots0 * 2^{(11)}$$

+	+|0 | 0 | 0 | 0| 0| 1 | 1	1 | 0 | 1 | 0 | 0 | 0 | 0 | ... ... |0 |0

(*) Na verdade:
(i) para representar o expoente e o seu sinal a estratégia é levemente diferente: usa-se o expoente deslocado.
(ii) na mantissa pode-se ganhar mais um bit, já que o primeiro bit (o mais significativo) da mantissa, em binário, **é sempre 1**.

Representação dos números reais

Versão simplificada, acadêmica (*)

Sinal	Expoente (com o sinal)	Dígitos do valor

$y = (10.5)_{10} = (1010.1)_2 \rightarrow 0.101010\ldots0 * 2^{(100)}$

+	+|0 | 0 | 0 | 0| 1| 0 | 0	1 | 0 | 1 | 0 | 1 | 0 | 0 | ... ... | 0 | 0

$z = -(0.75)_{10} = (0.11)_2 \rightarrow -0.1100\ldots0 * 2^{(0)}$

-	+|0 | 0 | 0 | 0| 0| 0 | 0	1 | 1 | 0 | 0 | 0 | 0 | 0 | ... ... | 0 | 0

(*) Na verdade:
(i) para representar o expoente e o seu sinal a estratégia é levemente diferente: usa-se o expoente deslocado.
(ii) na mantissa pode-se ganhar mais um bit, já que o primeiro bit (o mais significativo) da mantissa, em binário, **é sempre 1**.

Conversão de Binário para Decimal

$$(101)_2 = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 4 + 1 = 5$$

Conversão de Binário para Decimal

$$
\begin{aligned}
(101)_2 &= 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 4 + 1 = 5 \\
(1010.1)_2 = &= 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 1 \times 2^{-1} \\
&= 8 + 2 + 0.5 = 10.5
\end{aligned}
$$

Conversão de Decimal para Binário

Exemplo: $21.78125 = (10101.11001)_2 = +.1010111001 \times 2^5$
Parte inteira:

$$
\begin{aligned}
21/2 &= 10 \times 2 + 1 \\
10/2 &= 5 \times 2 + 0 \\
5/2 &= 2 \times 2 + 1 \\
2/2 &= 1 \times 2 + 0 \\
&\rightarrow (10101.)_2 \\
&\text{verificando} \\
&= 1 \times 2^0 + 1 \times 2^2 + 1 \times 2^4 \\
&= 1 + 4 + 16 = 21
\end{aligned}
$$

Conversão de Decimal para Binário

Exemplo: $21.78125 = (10101.11001)_2 = +.1010111001 \times 2^5$

Parte fracionária:

$$
\begin{aligned}
0.78125 \times 2 &= 1.56250 \\
0.56250 \times 2 &= 1.12500 \\
0.12500 \times 2 &= 0.25000 \\
0.25000 \times 2 &= 0.50000 \\
0.50000 \times 2 &= 1.00000 \\
&\rightarrow (.11001)_2 \\
&\text{verificando} \\
&= 1 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-5} \\
&= \frac{1}{2} + \frac{1}{4} + \frac{1}{32} = \frac{25}{32} = .78125
\end{aligned}
$$

Formato proposto pela IEEE

Formato proposto pela IEEE (Institute of Electrical and Electronics Engineers) para uma palavra com 32 *bits*

Um número $x \in \mathbb{R}$ armazenado neste sistema tem seguinte formato:

$$x = \pm 1.d_1 d_2 \cdots d_p \times 2^{\,e}$$

- *1 bit é reservado para o sinal* do número
- *23 bits são reservados para a mantissa*
  como o primeiro bit não precisa ser armazenado porque é sempre 1 $\Rightarrow$ temos 24 dígitos na mantissa
- *8 bits são reservados para o expoente.*
  $\rightarrow$ *como* $(11111111.)_2 = 255, \Rightarrow 0 \leq e \leq 255$
  $\rightarrow -127 \leq e - 127 \leq 128$
  expoente máximo: 127
  expoente minimo: -126

Exemplo e Formatos da IEEE 754-1985

Exemplos: como o número $(10.5)_{10}$ é armazenado em uma palavra de 32 *bits*

$$
\begin{aligned}
10.5 &= (1010.1)_2 = (+.10101000)_2 \times 2^{(4)} \\
&= +1.0101000 \times 2^{(3)} \\
e - 127 = 3 &\Rightarrow e = 130 = (10000010.)_2 \\
&\Rightarrow [0|10000010|0101000\cdots]
\end{aligned}
$$

Exemplo e Formatos da IEEE 754-1985

Exemplos: como o número $(10.5)_{10}$ é armazenado em uma palavra de 32 *bits*

$$\begin{aligned}
10.5 &= (1010.1)_2 = (+.10101000)_2 \times 2^{(4)} \\
&= +1.0101000 \times 2^{(3)} \\
e - 127 = 3 &\Rightarrow e = 130 = (10000010.)_2 \\
&\Rightarrow [0|10000010|0101000\cdots]
\end{aligned}$$

$$\begin{aligned}
21.78125 &= (10101.11001)_2 = (+.1010111001)_2 \times 2^5 \\
&= +1.010111001 \times 2^4 \\
e - 127 = 4 &\Rightarrow e = 131 = (10000011.)_2 \\
&\Rightarrow [0|10000011|010111001\cdots]
\end{aligned}$$

Tabela: Formatos da IEEE 754-1985.

bits	intervalo	precisão (em digitos decimais)
32	$\pm 1.18 \times 10^{-38}$ a $\pm 3.4 \times 10^{38}$	$\simeq 7$
64	$\pm 2.23 \times 10^{-308}$ a $\pm 1.80 \times 10^{308}$	$\simeq 16$

Conversão de Decimal para Binário

Ao fazer a conversão pode haver infinitos digitos na base binária (e na base 10 era finita)! Exemplos: 0.6 e 0.1

$0.6 = (0.1001100110011\cdots)_2$

$0.1 = (0.0001100110011\cdots)_2$

RESUMINDO:

- Toda valor numerico é armazenazado usando a base binária (ao fornecer o número na base 10 ele é convertido para a base binária)
- Os valores reais são representados usando o sistema de ponto flutuante normalizado.
- Ao converter um número na base binária ele pode ter infintos dígitos (e na base decimal tinha poucos dígitos)

Bibliografia

- Algoritmos Numéricos, Frederico F. Campos, Filho - 2ª Ed., Rio de Janeiro, LTC, 2018.
- https://ufsj.edu.br/portal2-repositorio/File/nepomuceno/compieee.pdf
- Material disponível na web: www.ufrgs.br/reamat/CalculoNumerico/livro-oct/main.html