

Aritmética da Computação

Trabalho para Casa: TPC2

Alberto José Proença

Metodologia

Leia as folhas do enunciado, e responda aos exercícios (manuscrito) nas folhas fornecidas para o efeito. **Relembra-se** que o objectivo dos TPC's é fomentar o estudo individual e contínuo, pelo que se valoriza mais o esforço de se tentar chegar ao resultado do que a correcção do mesmo. A correcção dos trabalhos far-se-á na aula da semana em que o trabalho é entregue.

A **penalização por fraude** será a atribuição de classificação com valor < 0.

Introdução

A lista de exercícios que se apresenta segue directamente o material apresentado na aula teórico-prática sobre representação binária de valores em vírgula flutuante (ver sumários e sugestões de leituras), podendo requerer conceitos básicos adquiridos em aulas anteriores.

Enunciado dos exercícios

Representação de valores em vírgula flutuante

Considere 2 novos formatos de vírgula flutuante, representados com 8-bits, baseados na norma IEEE:

- formato PEQUENO1:
 - o bit mais significativo contém o bit do sinal
 - os 4 bits seguintes formam o expoente (em excesso de 7)
 - os últimos 3 bits representam a mantissa
- formato PEQUENO2:
 - o bit mais significativo contém o bit do sinal
 - os 3 bits seguintes formam o expoente (em excesso de 3)
 - os últimos 4 bits representam a mantissa

Para todos os restantes casos, as regras são as mesmas que as da norma IEEE (valor normalizado, desnormalizado, representação do 0, infinito, e NaN).

1. Complete a expressão que, a partir dos campos em binário, permite calcular o valor em decimal para cada um dos formatos normalizados: $V = (-1)^S * 1.F * 2^{E-?}$
2. Para ambos os formatos, apresente os seguintes valores em decimal:
 - a) O maior número finito positivo
 - b) O número negativo normalizado mais próximo de zero
 - c) O maior número positivo desnormalizado
 - d) O número positivo desnormalizado mais próximo de zero

-
3. ^(A) Calcule os valores (n.º real, \pm infinito, NaN) correspondentes aos seguintes padrões de bits no formato PEQUENO1:
- a) 10110011
 - b) 01111010
 - c) 10010001
 - d) 00000011
 - e) 11000001
4. ^(R) Codifique os seguintes valores como números de vírgula flutuante no formato PEQUENO1:
- a) -111.01_3
 - b) 1/8 K (por ex., para representar a dimensão de um ficheiro em *bytes*)
 - c) $-0x10C$
 - d) 110.01
 - e) 0.005_8
5. Converta os seguintes números PEQUENO1 em números PEQUENO2. *Overflow* deve ser representado por $\pm\infty$, *underflow* por ± 0 e arredondamentos deverão ser para o valor par mais próximo.
- a) 00110011
 - b) 11101001
 - c) 00010000
 - d) 11001110
 - e) 11000101

Nº	Nome:
-----------	--------------

Resolução dos exercícios

(Nota: Apresente sempre os cálculos que efectuar no verso da folha)

1. PEQUENO1: $V = (-1)^S * 1.F * 2^{-----}$

PEQUENO2: $V = (-1)^S * 1.F * 2^{-----}$

2. Para ambos os formatos, apresente os seguintes valores em decimal:

a) O maior finito positivo: PEQUENO1 _____ PEQUENO2 _____

b) O negativo normalizado +próx. 0 PEQUENO1 _____ PEQUENO2 _____

c) $O > n^o$ positivo desnormalizado PEQUENO1 _____ PEQUENO2 _____

d) O positivo desnormaliz +próx. 0 PEQUENO1 _____ PEQUENO2 _____

3. (A) Calcule os valores correspondentes ao formato PEQUENO1 (modelo de resposta em a)):

a) 10110011 **Res.:** Valor normalizado, logo $V = (-1)^- * 1. _____ * 2^{---} = _____$

b) 01111010 **Res.:**

c) 10010001 **Res.:**

d) 00000011 **Res.:**

e) 11000001 **Res.:**

4. (R) Codifique os seguintes valores como números em vírgula flutuante no formato PEQUENO1

Pratique com o seguinte ex.: $0x72.A = 0111\ 0010.1010_2 = (-1)^0 * 1.1100\ 1010_2 * 2^6 =$
 $= (-1)^0 * 1.1100\ 1010_2 * 2^{13-7} =>$

a) -111.01_3 _____

b) $1/8\ K$ _____

c) $-0x10C$ _____

d) 110.01 _____

e) 0.005_8 _____

5. Converta os seguintes números PEQUENO1 em números PEQUENO2:

a) PEQUENO1: 00010000 PEQUENO2 _____

b) PEQUENO1: 11101001 PEQUENO2 _____

c) PEQUENO1: 00110011 PEQUENO2 _____

d) PEQUENO1: 11001110 PEQUENO2 _____

e) PEQUENO1: 11000101 PEQUENO2 _____

(apenas para esta alínea, mostre as operações que fez no verso da folha)