Vírgula Flutuante

Trabalho para Casa: **TPC2**

Baseado no guião de Alberto José Proença & Luís Paulo Santos

Metodologia

Leia as folhas do enunciado, e responda **obrigatoriamente** às questões colocadas na folha fornecida para o efeito (última folha deste guião). A resolução deve ser manuscrita e entregue **no início** da aula PL.

O objetivo dos TPC's é **fomentar o estudo** individual e contínuo, complementado por trabalho em grupo, sendo contabilizado o esforço para se tentar chegar ao resultado (que deverá ser defendido na aula) em detrimento da correção do mesmo.

O trabalho de grupo é aceite desde que as resoluções possam depois ser integralmente defendidas por quem as submeter. Quando tal acontecer será considerado fraude e conduz a uma avaliação negativa.

Máquinas de calcular não deverão ser usadas, para uma melhor assimilação dos resultados (nota: nos testes/exame não será permitida a sua utilização).

Introdução

A lista de exercícios que se apresenta segue diretamente o material apresentado na aula teórica sobre representação de números em vírgula flutuante (ver slides e sugestão de leitura), podendo requerer conceitos básicos adquiridos anteriormente.

Parte I - Representação de valores em vírgula flutuante precisão simples - IEEE 754

1. Represente os seguintes valores em vírgula flutuante precisão simples (IEEE 754). Apresente o resultado em hexadecimal.

Decimal	IEEE 754 precisão simples
16.375	
-1024	
$515.625*10^{-3}$	
$-2.25 * 2^{-128}$	

2. Converta para decimal os seguintes valores representados em vírgula flutuante precisão simples (IEEE 754).

IEEE 754 precisão simples	Decimal
0x436a0000	
0xc4000000	
0x00700000	
0xff800000	

Parte II - Representação de valores em vírgula flutuante: formatos `PEQUENO1` e `PEQUENO2`

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, subnormal/desnormalizado, representação do 0, ± `infinito`, `NaN`).

3. Complete a expressão que, a partir dos campos em binário, permite calcular o valor em decimal para cada um dos formatos normalizados: **$V = (-1)^{S} * 1.F * 2^{??}$**

4. 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 subnormal/desnormalizado
- **d)** O número positivo subnormal/desnormalizado mais próximo de zero
- **e)** O maior número inteiro positivo múltiplo de 4

5. Calcule os valores (número real, ± `infinito`, `NaN`) correspondentes aos seguintes padrões de bits no formato `PEQUENO1`:

- **a)** 0xBB
- **b)** 0x7C
- **c)** 0x92
- **d)** 0x05
- **e)** 0x41

6. Codifique os seguintes valores como números de vírgula flutuante no formato `PEQUENO1`:

- **a)** -110.01_3
- **b)** 1/16 Ki (por exemplo, para representar a dimensão de um ficheiro em *bytes*)
- **c)** –0x28C
- **d)** 101.01_{10}
- e) 0.006_8

7. Converta os seguintes números `PEQUENO1` em números `PEQUENO2`. *Overflow* deve ser representado por ± `infinito`, *underflow* por ±0 e arredondamentos deverão ser para o valor par mais próximo.

- **a)** 0xB5
- **b)** 0xEA
- **c)** 0x14
- **d)** 0xCF
- **e)** 0x02

8. Considere o desenvolvimento de código científico em C para execução num *notebook* atual, cuja especificação impõe que os números reais sejam representados com pelo menos 8 algarismos significativos. **Indique, <u>justificando</u>**, se consegue representar essas variáveis como `float` ou se tem de as representar como `double`.

9. Um valor do tipo real (*float*) vem representado na norma IEEE 754 por **$V= (-1)^S * 1.F * 2^{(Exp-127)}$**, se estiver normalizado. **Indique, explicitando** os cálculos, qual o maior inteiro ímpar que é possível representar exatamente, neste formato.

10. O formato RGBE é usado para representar de forma compacta pixéis com elevada gama dinâmica (em inglês High Dynamic Range - HDR). Cada pixel de uma imagem HDR é representado usando 3 valores reais positivos. São 3 valores porque são usadas 3 cores primárias: Red, Green and Blue (RGB). Os valores dos pixéis são sempre >= 0.

Se fossem usados valores em vírgula flutuante precisão simples seriam necessários 12 bytes para cada pixel; o formato RGBE permite usar 4 bytes para cada pixel. A ideia é que o expoente é partilhado pelos 3 canais (R,G e B) e representado no 4º byte. A parte fraccionária da mantissa de cada canal usa 8 bits; a parte inteira da mantissa não é representada e é igual a 0. O algoritmo para codificar um pixel é o seguinte:

- identificar o canal (R, G ou B) com valor máximo: chamemos-lhe $V_{max} = max(V_R, V_G, V_B)$;
- calcular uma constante de normalização que seja uma potência de 2, $N = 2^E$, tal que $\frac{V_{max}}{2^E} \in [0.5 \cdots 1[$;
- normalizar os valores dos 3 canais: $(V_{nR}, V_{nG}, V_{nB}) * 2^E = \left(\frac{V_R}{2^E}, \frac{V_G}{2^E}, \frac{V_B}{2^E}\right) * 2^E = (V_R, V_G, V_B)$;
- codificar a parte fraccionária de V_{nR}, V_{nG} e V_{nB} em 8 bits cada e codificar o expoente E em 8 bits usando excesso de 128 (nota: o sinal não é codificado explicitamente porque os valores são sempre >= 0).

Codifique o pixel com o valor (24, 20, 6) em RGBE apresentando a respectiva sequência de bits em hexadecimal.

Nº	Nome:	Turma:

Resolução dos exercícios

(**Nota**: Apresente sempre os cálculos que efectuar no verso da folha; o não cumprimento desta regra equivale à não entrega do trabalho.)

1. Represente os seguintes valores em vírgula flutuante precisão simples (IEEE 754). Apresente o resultado em hexadecimal.

Decimal	IEEE 754 precisão simples
16,375	
$515{,}625*10^{-3}$	

2. Converta para decimal os seguintes valores representados em vírgula flutuante precisão simples (IEEE 754).

IEEE 754 precisão simples	Decimal
0x436a0000	
0xc4000000	

3. PEQUENO1: **V= (-1)^S * 1.F * 2_____** PEQUENO2: **V= (-1)^S * 1.F * 2_____**

4. 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 > nº positivo subnormal PEQUENO1_____ PEQUENO2_____

d) O positivo subnormal +próx. 0 PEQUENO1_____ PEQUENO2_____

e) O > inteiro positivo múltiplo de 4 PEQUENO1_____ PEQUENO2_____

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

a) 0xBB **Res.:** Valor normalizado, logo V= (-1)__ * 1._____ * 2____ = _____

b) 0x7C **Res.:**

6. 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 1_2 * 2^6 =
= $(-1)^{\mathbf{0}}$ * 1.1100 1010 1_2 * 2^{13-7} =>

__ __ __ __ __ __ __ __

a) -110.01_3 __ __ __ __ __ __ __ __

b) 1/16 Ki __ __ __ __ __ __ __ __

7. Converta os seguintes números PEQUENO1 em números PEQUENO2:

a) PEQUENO1: 0xB5 PEQUENO2_____

b) PEQUENO1: 0xEA PEQUENO2_____

e) PEQUENO1: 0x02 PEQUENO2_____