



$d_0 \ d_1 \ / \ d_2 \ d_3 \ d_4 \ d_5 \ d_6 \ d_7$

Nome: _____ Matrícula: /

Prova 1

- 1) (9.0) O padrão IEEE 754 para representação de números reais em formato de ponto flutuante foi criado a fim de tentar normatizar a representação desses números. No entanto, dependendo da aplicação do processador, outros formatos podem ser mais eficientes. A fabricante de processadores gráficos (GPU) NVIDIA, observando que para fins de visualização gráfica, a representação em ponto flutuante não necessita ser tão precisa e a fim de aumentar a capacidade de processamento de suas GPUs criou um novo formato. A NVIDIA utiliza um formato *half* de ponto flutuante, baseado no padrão IEEE 754 (*single* e *double*), de apenas 16 bits; sendo 1 bit para indicar o sinal, 5 bits para representar o expoente e o restante dos bits para representar a fração.
- a) (0.5) Considerando que as mesmas características do IEEE754 foram adotadas para a representação do 0 (zero), $\pm\infty$ (\pm infinito) e NaN (*Not A Number*), escreva em hexadecimal um exemplo de cada uma dessas 4 entidades.
- b) (0.5) Calcule a faixa dinâmica de representação do formato *half*, isto é, calcule em decimal o menor valor positivo representável e o maior valor positivo representável sem overflow ou underflow.
- c) (2.0) Com esse novo formato, processadores de 32 bits são capazes de armazenar um número complexo utilizando apenas um registrador, colocando a parte real nos 16 bits mais significativos e a parte imaginária nos 16 bits menos significativos. Escreva em hexadecimal o conteúdo dos registradores \$f0 e \$f1:

$$\$f0 = \pi - \frac{\pi}{2}i$$

$$\$f1 = -0.000114440917969 + 6.144 \times 10^4 i$$

- d) (3.0) Escreva um procedimento em Assembly MIPS que dado um número complexo *half* armazenado no registrador \$f0 retorne em \$f12 a parte real e em \$f13 a parte imaginária, ambos no formato IEEE754 precisão simples.
- e) (1.0) Um core de um processador NVIDIA da série GeForce 9400GT, mostrada na Fig.1, possui um clock de 550MHz e é capaz de realizar esta operação em 5 ciclos. Qual o fator de desempenho da sua implementação em d), utilizando um processador MIPS uniclo de frequência de clock de 1GHz?
- f) (1.0) Uma placa de vídeo NVIDIA possui 64Mbytes de memória. Supondo que toda memória seja utilizada para armazenar uma imagem quadrada, onde cada pixel é representado com 24 bits (8 bits para cada componente de cor RGB). Qual o tamanho máximo, em pixels, desta imagem? Qual o número de cores que podem ser representadas?
- g) (1.0) Implemente a instrução abaixo, existente na arquitetura de uma GPU NVIDIA, como uma pseudo instrução MIPS. (Obs.: Pseudo instruções não podem acessar a memória e LABEL deve ser obrigatoriamente um endereço de 32 bits qualquer)

```
repnz $t1,LABEL    # se $t1!=0 então $t1=$t1-1 e PC= LABEL
#                  senão PC=PC+4
```

- 2) (2.0) Dado o código em Assembly MIPS ao lado, em que o label INICIO corresponde ao endereço 0x00400000 da memória.

O que será escrito na tela? Justifique.



Figura 1: Placa de Vídeo GeForce 9400GT

```
INICIO: lui $a0, 0xFFd3d4
        la $t1, 0x0810000B
        la $t2, JUMP
        sw $t1, 0($t2)
JUMP:   beq $a0, $zero, FIM
        la $t1, 0x208400d5d6
        sw $t1, 20($t2)
        li $a0, 0x00d3d4
FIM:    li $a1, 0xFFd3d4
        li $v0, 1
        syscall
```

BOA SORTE!

Prova 1

Gabarito

1) Half



a) 0: 0x0000 //

5 1 Exponente Fração

$$+\infty \rightarrow E=31 \text{ Frac}=0 \quad 0111110000000000 = 0x7C00 //$$

$$-\infty \rightarrow E=31 \text{ Frac}=0 \quad 1111110000000000 = 0xFC00 //$$

$$\text{NaN} \rightarrow E=31 \text{ Frac} \neq 0 : 0x7C01 //$$

b) menor: $E=1$ Fração = 0 Offset = 15

$$\text{menor} = (-1)^0 \times 1.000000 \times 2^{1-15} = 6,103515625 \times 10^{-5} //$$

$$\text{maior} = (-1)^0 \times 1.11111111 \times 2^{30-15} \approx 65536 // = 65504 //$$

$$\rightarrow \approx 2 = 1.9990234375 //$$

c) π em Half $3,14159265359 = 1,5707963265 \times 2^1$

$$\text{PF: } \pi = 11.0010010000 \quad \rightarrow \text{exponente} = 16$$

$$\text{log0: } \pi = (-1)^0 \times (1.1001001000) \times 2^1$$

$$\underbrace{010000}_{4} \underbrace{1001}_{2} \underbrace{0010}_{4} \underbrace{0000}_{8} = 0x4248 //$$

$$-\pi/2 = (-1)^1 \times \pi \times 2^{-1}$$

$$-\pi/2 = (-1)^1 \times (1.1001001000) \times 2^0 \quad \rightarrow \text{exponente} = 15$$

$$\underbrace{101111}_{13} \underbrace{1001}_{4} \underbrace{001000}_{8} = 0xBE48 //$$

logo:

$$\phi/\theta = 0x4248BE48 //$$

$$\text{PF: } -0,000114440917969 = -1,875 \times 2^{-14}$$

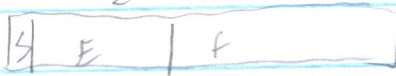
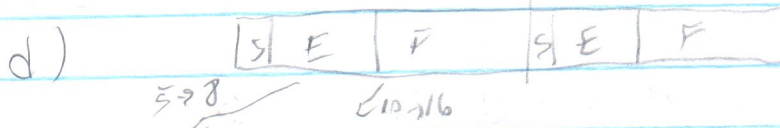
$$= (-1)^1 \times 1,1110000000 \times 2^{-14} \quad \rightarrow \text{exponente} = -1$$

$$\underbrace{100001}_{8} \underbrace{1110}_{7} \underbrace{0000}_{8} \underbrace{0000}_{0} = 0x8780$$

$$\begin{aligned}
 6,144 \times 10^4 &= 1.875 \times 2^{15} \\
 &= (-1)^0 1.1110000000 \times 2^{15} \rightarrow \text{exponent} = 30 \\
 &\quad 0111101110000000 \\
 &\quad \underline{7 \quad 3 \quad 8 \quad 0} = 0x7B80
 \end{aligned}$$

Logo:

$$\$f1 = 0x87807B80 //$$



* Sem consideração casos
Especiais: 0, NaN

COMPAREL: mfc1 \$t9, \$f0

lui \$t1, 0xFFFF

and \$a0, \$t1, \$t9 # a0 = Parte REAL

move \$fp, \$ra

jal ajuste

mtc1 \$v0, \$t12

sll \$a0, \$t9, 16 # a0 = Parte Imag

jal ajuste

mtc1 \$v0, \$t13

jr \$fp

Ajuste: beq \$a0, \$zero, ZERO

set lui \$t5, 0x8000 # mascara sinal

lui \$t6, 0x7C00 # mascara expoente

lui \$t7, 0x03FF # mascara fração

and \$t4, \$a0, \$t5 # extra: sinal

and \$t0, \$a0, \$t6 # ajuste do expoente

srl \$t0, \$t0, 26

addi \$t0, \$t0, -15 # 0-55 somar 112

addi \$t0, \$t0, 127

sll \$t0, \$t0, 23

OR \$t0, \$t0, \$t4 # insere sinal

and \$t3, \$a0, \$t7 # ajuste Fração

SRL \$t3, \$t3, 3

OR \$v0, \$t0, \$t3 # insere Fração

JR \$ra

ZERO: MOV \$v0, \$ZERO
JR \$ra

$$e) \quad t_1 = 5 \times \frac{1}{500M} = 10ns$$

$$t_2 = I \times 1 \times \frac{1}{1G} = 37 \times \frac{1}{1G} = 37ns$$

$$17 = \frac{t_1}{t_2} = \frac{10n}{37n} = 0,27 //$$

f) memória: 64 Mbytes

Imagem: $L \times L \times 3$ bytes = 64 Mbytes

$$L \times L = \frac{64M}{3} \rightarrow L = \sqrt{\frac{64M}{3}} = \frac{8}{\sqrt{3}} \text{ Kbytes}$$

$$L = 4729,65$$

$$\hookrightarrow \log_2 L = 4729 \times 4729 \text{ pixels} //$$

$$N_{\text{cores}} = 2^{24} = 16.777.216 \text{ cores} //$$

g) repz \$t1, LABEL

r4

BEQ \$t1, \$ZERO, SAi

addi \$t1, \$t1, -1

lui \$a1, LABEL(31..16)

ORI \$a1, \$a1, LABEL(15..0)

JR \$a1

SAi:

2)

```

0x40000  inicio:  lui $a0, 0xFF23
04          lui $t1, 0x810
08          ori $t1, $t1, 0x0B → $t1 = 0000 1000 0001
0C          lui $t2, jump($t1, 16) 0000 0000 0000 1011
10          ori $t2, $t2, jump($t1, 0)
14          sw $t1, 0($t2)
18  jump: beq $a0, $zero, fim → 0x40002C
1C          lui $t1, 0x2084
20          ori $t1, $t1, 0x0045
24          sw $t1, 20($t2)
28          li $a0, 0x0023
2C  fim:    li $a1, 0xFF23
30          li $v0, 1 → print int
34          syscall
  
```

Logo: Será escrito \$a0 em decimal

$\$a0 = 0xFF23\ 0000$

1111 1111 0010 0011 0000 0000 0000 0000

Será mostrado: -14483456