Ministério da Educação
UNIVERSIDADE DE BRASÍLIA
Instituto de Ciências Exatas
Dep. Ciência da Computação

Disciplina: CIC 116394 – Organização e Arquitetura de Computadores – Turma A 2009/2
Prof.: Marcus Vinicius Lamar

d_0 d_1 / d_2 d_3 d_4 d_5 d_6

Nome:_______________________________________ Matrícula: ☐☐/☐☐☐☐☐

Prova 1

1) **(6.0)** Uma das maiores dificuldades em programação Assembly de qualquer processador é, em grande parte dos casos, a ausência de funções matemáticas transcendentais implementadas diretamente na ISA, que são de grande importância na simulação de processos físicos.
Sabendo que a série de potência da função seno é definida como:

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} \ldots + (-1)^{n-1}\frac{x^{2n-1}}{(2n-1)!}$$

a) (3.0) Implemente um procedimento eficiente (número mínimo de operações) em Assembly MIPS que receba como argumento um ângulo em radianos [-π,π], no formato *float* (IEEE 754 single) no registrador \$f0, *n* no registrado \$a0, e retorne o seno deste ângulo em formato *float* no registrador \$f12, respeitando a convenção do uso dos registradores.

b) (1.0) Qual o valor do registrador *\$f12* em hexadecimal, se $n \to \infty$ e:
b.2) (0.25) x=*-1.5707963267948966192313216916398;*
b.3) (0.25) x=*0.0;*
b.4) (0.5) $x = d_0.d_1d_2d_3d_4d_5d_6$;

c) (1.0) Sabendo que um processador uniciclo de frequência de clock de 500MHz, dispõe de apenas 260ns para executar o procedimento seno escrito por você no item a), qual o valor máximo de *n* que poderia ser utilizado?

d) (1.0) Um processador multiciclo, possui diretamente em sua ISA a instrução sen \$f12,\$f0 que necessita de 32 períodos de clock para ser executada com *n=8* (default). Qual a frequência de clock que este processador deve ter de modo que esta implementação tenha um fator de desempenho de 3 frente a sua implementação usando o processador uniciclo do item c) ?

2) **(2.0)** O padrão de ponto flutuante IEEE 754 especifica precisão simples de 23 bits para significando e 8 bits para o expoente. Supondo que redefiníssemos para 21 bits de significando e 10 bits de expoente.
   a) (0.5) Qual seria o bias do expoente?
   b) (1.0) Qual seria o intervalo de número que podem ser representados sem overflow ou undeflow?
   {-∞,[A<0,B<0],0,[C>0,D>0], +∞}
   c) (0.5) Quais seriam as vantagens e desvantagens desta nova representação?

3) **(3.0)** Dado o código em Assembly MIPS ao lado, em que o label INICIO corresponde ao endereço 0x00400000 da memória.
Qual é o valor impresso na tela? Justifique.

```
INICIO:  ori $a0,$zero,0x00d2d3
         la $t1,0x0810000B
         la $t2,JUMP
         sw $t1,0($t2)
JUMP:    beq $a0,$zero,FIM
         la $t1, 0x208400d5d6
         sw $t1,20($t2)
         li $a0,2
FIM:     li $a1,3
         li $v0,1
         syscall
```

BOA SORTE!

OAC TURMA A 2009/2

1º PROVA

GABARITO

1) a)
```
.data
    UM: .float 1.0
    MUM: .float -1.0

    .text
      :        #Prog. Principal
      :

SENO:  l.s $f1, UM         # f1=1.0 -> Pseudo: 2 instr.
       l.s $f7, MUM        # f7=-1.0 -> Pseudo: 2 inst
       mtc1 $zero, $f12    # f12=0 somatório
       mov.s $f3, $f0      # f3=x     x^n
       mov.s $f4, $f1      # f4=1     n!
       mov.s $f6, $f1      # f6=1 contador n
       move $t0, $zero     # t0=0
       mul.s $f5, $f0, $f0 # f5=x^2

LOOP:  beq $t0, $a0, SAIDA
       div.s $f10, $f3, $f4    # x^n/n!
       add.s $f12, $f12, $f10  # seno += x^n/n!
       mul.s $f3, $f3, $f5     # incrementa x^2
       add.s $f6, $f6, $f1     # n++
       mul.s $f4, $f4, $f6     # n!xn
       add.s $f6, $f6, $f1     # n++
       mul.s $f4, $f4, $f6     # n!xn
       mul.s $f4, $f4, $f7     # ajusta o sinal
       addi $t0, $t0, 1        # contador+1
       j LOOP
SAIDA: jr $ra
```

2) Expoente = 10 bits   Fração = 21 bits

a) OFFSET = 511.

b) Considerando similar ao IEEE 754 temos
Expoente $\in [1, 1022]$ $0 \to$ desnorm, $1023 \to \infty$

Logo:

$B = (-1)^1 \times (00000\ldots0) \times 2^{1-511} = -2^{-510} = -2{,}9833 \times 10^{-154}$

$C = -B = 2{,}9833 \times 10^{-154} = 2^{510}$

$A = (-1)^1 \times (11111\ldots1) \times 2^{1022-511} = -1 \times 2 \times 2^{511} = -2^{512}$

$A = -1{,}34078079 \times 10^{154}$

$D = -A = 1{,}34078079 \times 10^{154} = 2^{512}$

c) Vantagem: mais bits no expoente aumenta a faixa dinâmica dos números representáveis

Desvantagem: diminui o nº de bits do significando logo perde em precisão da representação

1)

b) $n \to \infty$ exatamente $\sin(x)$

b.1) $x = -\pi/2 \to \$f12 = -1 = (-1)^1 (0,0000000) \times 2^0$

Sinal = 1 Fração = 0 Expoente = 127

Logo: 1 01111111 00000000000000000000000

$f12 = 0BF800000

b.2) $x = 0 \to \$f12 = 0$

$f12 = 0X00000000

Exemplo:

b.3) $x = 0{,}912345 \to \$f12 = 0{,}790940801404$

$\$f12 = (-1)^0 (1001\ 0100\ 1111\ 0110\ 0011\ 001) \times 2^{-1}$

Sinal = 0 Fração = J Expoente = 126

Logo: 0 01111110 1001 0100 1111 0110 0011 001

$f12 = 03F4A7B19

c) Proc. unciclo $f = 500MHz$

$t = 260ns$

$n_{max} = ?$

$T = 1/500M = 2ns$

Logo pode utilizar $t/T = 260/2 = 130$ passos

p/ n passos:

n^o ciclos $= \underset{\text{início}}{10} + 11 \times n + 2$ ↳ último beq, jr

Logo:

$11n + 12 = 130 \to n = 10{,}72 \to n_{max} = 10$

d) $n = 8 \to 32$ ciclos

$\eta = \dfrac{t_{meu}}{t_{novo}} = 3$

$t_{meu} = 2ns \times (10 + 11 \times 8 + 2)$

$t_{meu} = 200ns$

$t_{novo} = 200n/3 = 66{,}66ns \to 32$ ciclos

$\hookrightarrow T = \dfrac{66{,}66}{32} = 2{,}083ns \to f = 480MHz$

3) 0x0040 0000 Inicio: ori ~~$a0~~, ~~$zero~~, 0x0012

0004 la $t1, 0x0810 000B

0008 ↳ j 0040 002C

↳ FiM

000C la $t2, JUMP

0010

0014 sw $t1, 0 ($t2)

0018 JUMP: ~~beq $a0, $zero, FiM~~ → j FiM

001C la $t1, 0x2084 0045

0020

0024 sw $t1, 20 ($t2)

0028 li $a0, 2

002C FiM: li $a1, 3

0030 li $v0, 1

0034 syscall

Escreve 0x0012 → 18 na tela

0x00 $d_2 d_3$