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

Disciplina: CIC 116394 – Organização e Arquitetura de Computadores – Turma B 2008/2
Prof.: Marcus Vinicius Lamar

Nome:____ Matrícula: _______________

Prova 1

1) (4.0) Dado o polinômio de segundo grau $P_2(a,b,c,x) = a.x^2 + b.x + c$,

a)(2.0) Respeitando a convenção do uso dos registradores, escreva um procedimento em Assembly MIPS que receba os valores inteiros dos coeficientes (a,b,c) e de x, e retorne o valor de P_2.
Use a seguinte declaração em C do procedimento: `int P2 (int a, int b, int c, int x);`

b)(2.0) Escreva um procedimento em Assembly MIPS que receba os valores dos coeficientes reais (a,b,c), retorne dois números imaginários na forma: \$f0+j\$f1 e \$f2+j\$f3 , correspondentes as raízes do polinômio.
Use a seguinte declaração em C do procedimento: `void bascara(float a, float b, float c);`

2) (2.0) Para o padrão IEEE 754 de representação numérica de ponto flutuante, responda:

a)(1.0) Dado um número real no intervalo [0,1] representado em precisão simples neste padrão, em comparação com o mesmo número representado em ponto fixo Q31, analise: Qual a perda máxima de casas binárias sofrida? Qual a correspondente perda em casas decimais?

b)(0.5) Escreva em hexadecimal e em decimal o maior e menor números positivos representáveis em precisão simples.

c)(0.5) Qual o resultado em hexadecimal da seguinte expressão em ponto flutuante: (0x40300000 + 0xC0AA0000) / 0x3F800000

3) (2.5) A vida do programador em Assembly MIPS é bastante facilitada pelo montador, uma vez que o mesmo implementa de maneira automática, várias pseudo-instruções que são bastante úteis. Dado que BIG é uma constante imediata de 32 bits, SMALL uma constante de 16 bits, LABEL um endereço de 32 bits, implemente as seguintes pseudo-instruções:

```
a)(1.0) abs $t0, $t1                # $t0=|$t1|  (sem utilizar branches)
b)(0.5) bgeu $t0,SMALL,LABEL        # if (unsigned($t0)>=unsigned(SMALL)) goto LABEL
c)(0.5) s.d $f0,LABEL               # Memory[LABEL]=$f0; Memory[LABEL+4]=$f1
d)(0.5) sne $t0,$t1,$t2             # if ($t1!=$t2) then $t0=1  else  $t0=0;
```

4)(2.0) A tabela a seguir mostra o número de operações em ponto flutuante executadas em três programas diferentes e o tempo de execução desses programas em três computadores diferentes:

Programa	Operações em Ponto flutuante	Tempo de execução em segundos		
		Computador A	Computador B	Computador C
Programa 1	5×10^9	2	5	10
Programa 2	20×10^9	20	20	20
Programa 3	40×10^9	200	50	15

a)(1.0) Considerando que o Computador A seja a máquina base, qual computador possui a tendência de ser o mais rápido para um workload qualquer?
b)(1.0) Qual seria o workload (em %) que divide o tempo de processamento de forma igualitária entre os programas no computador B?

5) (0.5) Explique porque MIPS não é uma medida de desempenho adequada.

BOA SORTE!

OAC - TURMA B 2008/2

1ª PROVA

GABARITO

1)a) # P2 $a0=a $a1=b $a2=c $a3=x

```
P2: mul $v0, $a0, $a3      # a.x
    add $v0, $v0, $a1      # a.x+b
    mul $v0, $v0, $a3      # x(a.x+b)
    add $v0, $v0, $a2      # x(a.x+b)+c
    jr $ra                 # RETORNO em $v0
```

b)
```
.data
C0: .Float 0.0      # CONSTANTES NA MEMÓRIA
C2: .Float 2.0
C4: .Float 4.0

.text
BASCARA:            # $f0=a $f1=b $f2=C
    mul.s $f4, $f1, $f1      # b^2
    mul.s $f5, $f0, $f3      # a.c
    l.s $f6, C4              # 4
    mul.s $f5, $f5, $f6      # 4.a.c
    sub.s $f4, $f4, $f5      # delta = b^2-4ac

    l.s $f6, C0              # 0
    c.lt.s $f4, $f6          # delta<0 ?
    l.s $f6, C2              # 2
    neg $f1, $f1             # -b
    mul.s $f0, $f0, $f6      # 2a
    bc1t COMPLEX
```

```
sqrt.s $f4, $f4          # √DELTA
add.s $f5, $f1, $f4      # -b + √DELTA
div.s $f7, $f5, $f0      # (-b+√DELTA)/2a

sub.s $f5, $f1, $f4      # -b - √DELTA
div.s $f8, $f5, $f0      # (-b-√DELTA)/2a

mov.s $f0, $f7           # f0 + j0
L.S $f1, C0
mov.s $f2, $f8           # f2 + j0
mov.s $f3, $f1
jr $ra

COMPLEX:
    div.s $f7, $f1, $f0      # -b/2a

    neg.s $f4, $f4           # -DELTA
    sqrt.s $f4, $f4          # √DELTA
    div.s $f8, $f4, $f0      # √DELTA/2a

    mov.s $f0, $f7           # f0 + jf1
    mov.s $f1, $f8
    mov.s $f2, $f7           # f2 - jf3
    neg.s $f3, $f8

    jr $ra
```

2)

a) número entre $[0,1] = 2^{-1} \times (1+0{,}xxxxxx)$

IEEE 754 0 01111110 xxxxxxxxxxxxx..xx

sinal expoente 23 bits

126 ↳ casas binárias

Q31 0,xxxxxxx...xx ↳ dígitos significativos

31 bits

↳ casas binárias

Logo; Perda em casas binárias: $31-23=8$ //

Perda em casas decimais: < nº representável

$2^{-23} = 1{,}19 \times 10^{-7}$

$2^{-31} = 4{,}65 \times 10^{-10}$

→ $10-7 = 3$ casas decimais //

b)

Maior valor positivo: $0\ \underbrace{11111110}_{254}\ \underbrace{11111111\cdots11}_{\cong 2 \to 1.1111111111}$

Hexadecimal: $7F7F\ FFFF_h$ //

Decimal: $2^{254-127} \times (\cong 2) = 3{,}4028235 \times 10^{38}$ //

Menor valor positivo: $0\ \underbrace{00000001}_{1}\ 00000000\cdots00$

↳ $1 \Rightarrow 1.000\ldots$

Hexadecimal: 00800000_h //

Decimal: $2^{1-127} \times 1 = 1{,}17549435 \times 10^{-38}$ /

Considerando < nº desnormalizado: 0 00000000 0000000..01

Hexadecimal: 0000 0001 /

Decimal: $2^{-127} \times 2^{-23} = 2^{-149} \cong 1{,}4013 \times 10^{-45}$ //

c) $\dfrac{2{,}75 + (-5{,}3125)}{1.0} = -2{,}5625 \to 0x\ C0240000$ // //

3) a) abs \$t0, \$t1 → sra \$at, \$t1, 31 → cria { 000000...00 >0 ou 111111...11 <0

xor \$t0, \$t1, \$at → não inverte \$t1 se >0 / inverte \$t1 se <0

subu \$t0, \$t0, \$at → −(0) >0 / −(−1) <0

↳ p/ não indicar overflow

b) bgeu \$t0, small, LABEL → sltiu \$at, \$t0, SMALL

beq \$at, \$ZERO, LABEL

c) s.d \$f0, LABEL → lui \$at, $LABEL_{31-16}$

obs: lui zera bit - significativo.

ori \$at, \$at, $LABEL_{15-0}$

sdc1 \$f0, 0(\$at)

d) sne \$t0, \$t1, \$t2 → subu \$t0, \$t1, \$t2

sltu \$t0, \$ZERO, \$t0

4) a) máquina base A

	A	B	C
P1	1	2,5	5
P2	1	1	1
P3	1	0,25	0,075
Média Geométrica:	1	0,8549	0,7211

Logo C tem a tendência a ser + rápido //

b) workload p/ B:

$WP_1 = \frac{1}{5} / (\frac{1}{5} + \frac{1}{20} + \frac{1}{50}) = 0,7407 \rightarrow 74,07\%$

$WP_2 = \frac{1}{20} / (\frac{1}{5} + \frac{1}{20} + \frac{1}{50}) = 0,1851 \rightarrow 18,51\%$

$WP_3 = \frac{1}{50} / (\frac{1}{5} + \frac{1}{20} + \frac{1}{50}) = 0,07407 \rightarrow 7,4\%$

5) Porque não considera a quantidade de trabalho (processamento) feito por uma instrução.