



d0 d1 / d2 d3 d4 d5 d6

Nome: \_\_\_\_\_

Matrícula:  /

## Prova 1

**1) (3.0)** Dado o programa principal e a função em C para multiplicação de duas matrizes quadradas de ordem N ao lado, e respeitando a convenção do uso dos registradores:

- a)(2.0) Compile os programas para Assembly MIPS.
- b)(1.0) Escreva as equações que definem os números de instruções tipo-R, tipo-I e tipo-J necessárias ao cálculo de multM em função de N.

**2) (2.0)** A vida do programador em Assembly MIPS é bastante facilitada pelo montador SPIM, uma vez que o mesmo implementa, várias pseudo-instruções que são bastante úteis. Dado que IMM uma constante imediata de 16 bits e LABEL um endereço, implemente as seguintes pseudo-instruções:

- a)(0.5) mul \$t0,\$t1,\$t2 # \$t0=\$t1 \* \$t2
- b)(0.5) remi \$t0,\$t1,IMM # \$t0=resto(\$t1/IMM)
- c)(0.5) beqz \$t0,LABEL # se \$t0 == 0 goto LABEL
- d)(0.5) sne \$t0,\$t1,\$t2 # se \$t1 != \$t2 então \$t0=1

**3) (2.0)** Complete o mapa de memória, compilando para Linguagem de máquina o seguinte trecho, grifando os endereços dos labels LOOP e EXIT.

LOOP:  
sll \$t1,\$s3,4  
add \$t1,\$t1,\$s6  
lw \$t0,100(\$t1)  
beq \$t0,\$s5, EXIT  
subi \$s3,\$s3,1  
j LOOP  
EXIT:

Endereço	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	1024																															
	1028																															
	1032																															
	1036																															
	1040																															
	1044																															
	1048																															
	1052																															

**4) (2.0)** Para a aritmética computacional de inteiros, ponto fixo e ponto flutuante:

- a) (0.5) Represente o número decimal  $-d_5d_6$  em 8 bits complemento de 2 e em sinal e magnitude;
- b) (0.5) O número em hexadecimal  $0XCd_6d_5d_4d_3d_2d_1d_0$  em ponto flutuante IEEE754 precisão simples corresponde a que valor?
- c) (1.0) Dada a operação  $(01010111_2 \times 10101100_2)$ , o que ela representa em aritmética de ponto fixo em Q0, Q3 e Q7, quais seus resultados (em decimal e binário)?

**5) (2.0)** Considere duas implementações diferentes, I1 e I2, de uma mesma ISA que possui três classes de instruções (Tipo-A, Tipo-B e Tipo-C). A implementação I1 possui uma frequência de clock de 3GHz e I2 uma freqüência de 1.5 GHz. O número médio de ciclos para cada classe de instrução em I1 e I2 é apresentado na tabela.

Esta tabela também contém um resumo da proporção média de classes de instruções geradas por três compiladores diferentes. C1 é produzido pelo fabricante de I1, C2 é produzido pelo fabricante de I2 e C3 é um produto independente. Considere que cada compilador use o mesmo número de instruções para um determinado programa mas que o mix de instruções seja descrito pelos percentuais da tabela.

- a)(0.5) Usando C1 em I1 e I2, quanto mais rápido os fabricantes de I1 poderiam afirmar que ele é em comparação com I2?
- b)(0.5) Se comprasse um processador I1, que compilador você usaria?
- c)(0.5) Se comprasse I2, que compilador você usaria?
- d)(0.5) Que computador e compilador você compraria se todos os outros critérios fossem idênticos (custo, consumo, etc)?

**6) (1.0)** Explique o que são as arquiteturas Harvard e Von Neumann. Quais as vantagens e desvantagens de cada uma?

```
#include <stdio.h>

int N=4;
float A[4*4]={1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16};
float B[4*4]={16,15,14,13,12,11,10,9,8,7,6,5,4,3,2,1};
float C[4*4]={0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};
float RES;

float multM( float z[], float x[], float y[], int n ) {
    int i,j,k;
    float temp,soma;

    soma=0;
    for(i=0;i<n;i++)
        for(j=0;j<n;j++) {
            temp=0;
            for(k=0;k<n;k++)
                temp+=x[i*n+k]*y[k*n+j];
            z[i*n+j]=temp;
            soma+=temp;
        }
    return soma;
}

void main() {
    RES=multM(C,A,B,N);
}
```

Classe	CPI(I1)	CPI(I2)	C1	C2	C3
Tipo-A	2	1	40%	40%	50%
Tipo-B	3	2	40%	20%	25%
Tipo-C	5	2	20%	40%	25%

```

provala.s
    .data
N:   .word 4
A:   .float
1.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,12.0,13.0,14.0,15.0,16.0
B:   .float
16.0,15.0,14.0,13.0,12.0,11.0,10.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,1.0
C:   .float 0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0
RES: .float 0.0

    .text
    .globl __start

__start:
    la $a0,C
    la $a1,A
    la $a2,B
    lw $a3,N($zero)

    jal multM

    swc1 $f0,RES($zero)

    li $v0,10
    syscall

multM:
    mtc1 $zero,$f0 # $f0 = soma
    move $t0,$zero # $t0 = i

loopi: bge $t0,$a3,fimi
    move $t1,$zero # $t1 = j

loopj: bge $t1,$a3,fimj
    mtc1 $zero,$f1 # $f1 = temp
    move $t2,$zero # $t2 = k

loopk: bge $t2,$a3,fimk
    mul $t3,$t0,$a3
    add $t3,$t3,$t2
    sll $t3,$t3,2
    add $t3,$t3,$a1
    lwc1 $f2,0($t3) # $f2 = x[i*n+k]

    mul $t3,$t2,$a3
    add $t3,$t3,$t1
    sll $t3,$t3,2
    add $t3,$t3,$a2
    lwc1 $f3,0($t3) # $f3 = y[k*n+j]

    mul.s $f4,$f2,$f3
    add.s $f1,$f1,$f4 # temp+=x*y
    addi $t2,$t2,1
    j loopk

fimk: mul $t3,$t0,$a3
    add $t3,$t3,$t1
    sll $t3,$t3,2
    add $t3,$t3,$a0
    swc1 $f1,0($t3) # z[i*n+j]=temp
    add.s $f0,$f0,$f1 #soma+=temp

    addi $t1,$t1,1
    j loopj

fimj: addi $t0,$t0,1
    j loopi

fimi: jr $ra

```

prova1\_3.s

```
LOOP:  
sll $t1,$s3,4  
add $t1,$t1,$s6  
lw $t0,100($t1)  
beq $t0,$s5,EXIT  
addi $s3,$s3,-1  
j LOOP  
EXIT:
```

[0x00400000]	0x00134900	sll \$9, \$19, 4	; 2: sll \$t1,\$s3,4
[0x00400004]	0x01364820	add \$9, \$9, \$22	; 3: add \$t1,\$t1,\$s6
[0x00400008]	0x8d280064	lw \$8, 100(\$9)	; 4: lw \$t0,100(\$t1)
[0x0040000c]	0x11150003	beq \$8, \$21, 12 [EXIT-0x0040000c]; 5: beq	
\$t0,\$s5,EXIT			
[0x00400010]	0x2273ffff	addi \$19, \$19, -1	; 6: addi \$s3,\$s3,-1
[0x00400014]	0x08100000	j 0x00400000 [LOOP]	; 7: j LOOP

## OAC - TURMA-A

200712

## 1º Prova - Gabarito

1) a)

. DATA

A: .FLOAT 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 4, 15, 16

B: .FLOAT 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1

C: .FLOAT 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0

N: .WORD 4

RES: .FLOAT 0.0

. TEXT

.GLOBL \_\_START

\_\_START:

la \$t0, C

la \$a1, A

la \$a2, B

lw \$a3, N(\$ZERO)

jal multM

la \$t0, RES

swc1 \$f12, 0(\$t0)

li \$v0, 10

syscall

multM:

r mtc1 \$ZERO, \$f0

r move \$t0, \$ZERO

i loopj: bge \$t0, \$a3, FIMJ

r move \$t1, \$ZERO

i loopj: bge \$t1, \$a3, FIMJ

r mtc1 \$ZERO, \$f1

r move \$t2, \$ZERO

pusher

j beq

loopk: bge \$t2, \$a3, FIMK

I

mul \$t3, \$t2, \$a3

R

add \$t3, \$t3, \$t2

R

sll \$t3, \$t3, 2

I

add \$t3, \$t3, \$a1

R

lwc1 \$f2, 0(\$t3)

I

mul \$t3, \$t2, \$a3

R / R

add \$t3, \$t3, \$t1

R

sll \$t3, \$t3, 2

I

add \$t3, \$t3, \$a2

R

lwc1 \$f3, 0(\$t3)

I

mtc1 \$ZERO, \$f0

mul.s \$f4, \$f2, \$f3 R

add.s \$f1, \$f1, \$f4 R

addi \$t2, \$t2, 1 I

I

j loopk

J

FIMK: mul \$t3, \$t0, \$a3 R

add \$t3, \$t3, \$t1 R

sll \$t3, \$t3, 2 I

R	add \$t3,\$t3,\$a0
I	swc1 \$f1,0(\$t3)
R	add.s \$f0,\$f0,\$f1
I	addi \$t3,\$t3,1
S	j loop3
I	FMJ: addi \$t1,\$t1,1
S	j loop1
R	FMT: jr gra

$$\begin{aligned}
 \text{Tipo J} &= N^3 \\
 \text{Tipo I} &= [(6 \times n) + 4]n + 2 \\
 \text{Tipo R} &= ((9n) + 6)n + 1
 \end{aligned}$$

2)

9) `MUL %t0, %t1, %t2` : `MUL %t1, %t2`  
`MFLO %t0`

b) REMI \$T8,\$T1, IMM : addi \$at, \$ZERO, IMM  
div \$T1, \$at  
mfhi \$T8

c)  $\text{DEQZ } \$t0, \text{LABEL} : \text{ DEQ } \$t0, \text{ZERO}, \text{LABEL}$

d) SNE \$T0,\$T1,\$T2  
add \$T0,\$ZERO,\$ZERO #0E5.  
beq \$T1,\$T2,PULA  
add \$T0,\$ZERO,1  
PULA:

31

Loop: 1024 0000 0000 0001 0011 0100 1001 0000 0000  
1028 0000 0001 0011 0110 0100 1000 0010 0000  
1032 1000 1101 0010 1000 0000 0000 0110 0100  
1036 0001 0001 0001 0101 0000 0000 0000 0010 skip  
1040 0010 0010 0111 0011 1111 1111 1111 1111  
1044 0000 1000 0000 0000 0000 0001 0000 0000  
Exit: 1048

oldi 953, \$53,-1

4)

$$\text{a) } -25db : -02 = 11111110$$

$$\text{b) } \text{JXC}d_6d_5d_4d_3d_2d_1d_0$$

$$\text{0XC}2097840 =$$

$$\boxed{1100\ 0010\ 0000\ 1001\ 0111\ 1000\ 0100\ 0000}$$

$$\text{SIGN} = 1 \quad \text{EXP} = 132 \quad \text{Frac} = 0,0739822387695$$

$$N = (-1)^{\text{exp}} \times 2^{132-127} \times (1 + 0,0739822387695)$$

$$N = -39,3674316406$$

	00	03	07
c)	01010111	87	10,875
(x)	<u>10101100</u>	<u>x 172</u>	<u>x 21,5</u>
	0011101001110100	14964	233,8125
			0,913330078125

mult/division  $\bar{e}$   
unsigned

5)

$$\text{a) } t = N \cdot \text{inc} \times (\text{PI} \times \text{clock})$$

PI/C1

$$t_{I1} = (0,4 \times 2 + 0,4 \times 3 + 0,2 \times 5) \times 1/36$$

$$t_{I1} = 1 \text{ ns}$$

$$t_{I2} = (0,4 \times 1 + 0,4 \times 2 + 0,2 \times 2) \times 1/156$$

$$t_{I2} = 1,0666 \text{ ns}$$

$$\eta = \frac{1,0666}{1} = 1,0666 \rightarrow 666\% \text{ mehr auf 100}$$

b) P/I I1

$$tC1 = 1hs$$

$$tC2 = (0,4 \times 2 + 0,2 \times 3 + 0,4 \times 5) \times Y_{36} = 1,133hs$$

$$tC3 = (0,5 \times 2 + 0,25 \times 3 + 0,25 \times 5) \times Y_{36} = 1hs$$

C1 e C3 produzem o mesmo resultado

c) P/I I2

$$tC1 = (0,4 \times 1 + 0,4 \times 2 + 0,2 \times 2) \times Y_{156} = 1,066hs$$

$$tC2 = (0,4 \times 1 + 0,2 \times 2 + 0,4 \times 2) \times Y_{156} = 1,066hs$$

$$tC3 = (0,5 \times 1 + 0,25 \times 2 + 0,25 \times 2) \times Y_{156} = 1hs$$

Comparar C3

d) Analisando os resultados entre os

I2 + C3 e I1 + (C1 ou C3) produzem os melhores resultados.

e) Vantagem: programar e rodar em uma mesma máquina

Vantagem: Economia, uso + eficiência

Desvantagem: Gastos de RH maior

↳ velocidade limitada

Havendo: programar e rodar em diferentes PCs ≠ S

Vantagem: acesso + rapidez

Desvantagem: + custo, 2 mens. as.