Trabalho de Organização de Computadores

Prof. Dr. Eduardo Wenzel Brião	Data da entrega:28/06/2018

1 (2,5 pontos) – Com base nas informações do slide em anexo no mail ou no moodle (denominado Arquitetura MIPS), Slide 2.23, (p.23)), responda a seguir as questões:

a), se houver um problema no hardware, e o sinal UlaFonte for sempre 1, quais tipos de instruções não poderão ser executadas?

b) O comando lw para ser executado, é necessário ajustar para que valor o MemParaReg? Zero ou Um?

c) Quando é executado um ADD, qual será o sinal ajustado o sinal RegDst?

2 (2,5 pontos) Fazer um programa que calcule o somatório de 4 valores lidos pelo usuário (ex: var .word *valor*) e calcule sua média.

3 (2,5 pontos) Fazer um programa que conte e mostre na tela quantos números pares e impares lidos pelo usuário. O usuário deverá terminar sua leitura inserindo um número negativo.

4 (2,5 pontos) Fazer um programa que calcule a soma de dois vetores e escreva o resultado em um terceiro vetor. Devem ser inseridos na própria declaração e devem ser de mesmo tamanho.

Exemplo:

```
.data

vet1: .word 5 6 4 3 2

vet2: .word 2 3 4 5 6

vet3: .space 20 (5 posições * 4 bits) OU

vet3: .word 0 0 0 0 0
```

o resultado no simuador deverá ser vet3 = {7,9,8,8,8}

PENALIDADE POR DIA DE ATRASO: 30% DA NOTA!

PENALIDADE: O ALUNO QUE PLAGIAR O TRABALHO TERÁ NOTA ATRIBUIDA IGUAL A ZERO!

TRABALHO INDIVIDUAL OU EM DUPLAS!!

BOM TRABALHO!!

Eduardo Wenzel Brião (mail: eduardo.briao@riogrande.ifrs.edu.br)

Anexo I

Alguns códigos úteis

```
lw $t1, offset($t0) - load word: → $t1 = mem[$t0+offset]

sw $t1, offset($t0) - store word → mem[$t0+offset] = $t1

la $t1, valor_memoria → endereço da variável ou vetor.

li $t1,4   →  $t1 = 4

move $t1,$t2 → $t1 = $t2

add $s0,$s0,$t1  → $s0 = $s0 + $t1

sub $s0,$s0,$t1  → $s0 = $s0 - $t1

div $t1,$t2,$t3 → $t1 = $t2/$t3  (divisão inteira)

rem $t1,$t2,$t3 → $t1 = $t2%$t3  (resto da divisão inteira)

beq $t1,$t2,ROTULO → se $t1 = $t2 pula para ROTULO

bne $t1,$t2,ROTULO → se $t1 != $t2 pula para ROTULO

blt $t1,$t2,ROTULO → se $t1 < $t2 pula para ROTULO

bte $t1,$t2,ROTULO → se $t1 <= $t2 pula para ROTULO

bgt $t1,$t2,ROTULO → se $t1 > $t2 pula para ROTULO

bge $t1,$t2,ROTULO → se $t1 >= $t2 pula para ROTULO

bne $t1,$t2,ROTULO → se $t1 != $t2 pula para ROTULO

syscall → chamada de sistemas (ver help do mars)
```