Universidade de Brasília

Departamento de Ciência da Computação

Disciplina: CIC 116394 – Organização e Arquitetura de Computadores – Turma A 2017/1

Prof. Marcus Vinicius Lamar

d_0 d_1 / d_2 d_3 d_4 d_5 d_6 d_7 d_8

Nome: GABARITO Matrícula: 12/0123456

Prova 1

```
.data
        STRING: .word -4
.text
MAIN:   li $k0, 0xd8
        la $v0,STRING
        move $a2,$zero
        move $t5,$sp
        lw $s4,0($v0)
        jal VOLTA
        jal NUM
        li $v0,0x0a
        syscall

NUM:    lw $s0,0($t5)
        add $a2,$a2,$s0
        sub $t5,$t5,$s4
        bne $t5,$sp,NUM
        jr $ra

VOLTA:  add $t5,$t5,$s4
        mult $k0,$k0
        addi $k0,$k0,-1
        mflo $fp
        sw $fp,0($t5)
        beq $fp,$zero,L1
        add $ra,$ra,$s4
L1:     jr $ra
```

(4.0)1) Considere o programa em Assembly MIPS ao lado, onde a convenção do uso dos registradores é completamente ignorada.

(1.5) a) Qual o valor final, em decimal, do registrador $a2?

(1.5) b) Qual o número de instruções executadas desde o início (MAIN:) até o retorno do controle ao Sistema Operacional?

(1.0) c) Qual a quantidade de memória de dados (em bytes) e de programa (em bytes) utilizada na sua execução?

(3.0)2) Considere a instrução abaixo, onde o registrador $f1 possui o valor 0x01800000.

```
sqrt.s $f0,$f1
```

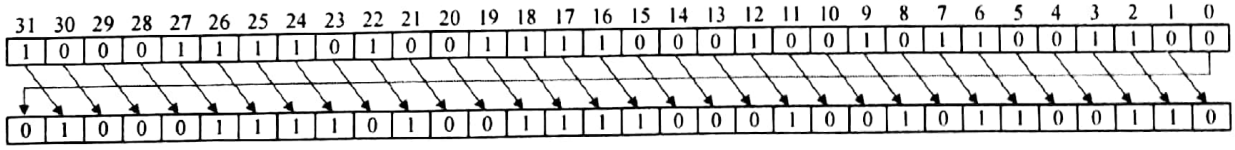
(1.0)a) Escreva o código em linguagem de máquina, em hexadecimal, dessa instrução.

(1.0)b) Qual o valor em hexadecimal contido registrador $f0?

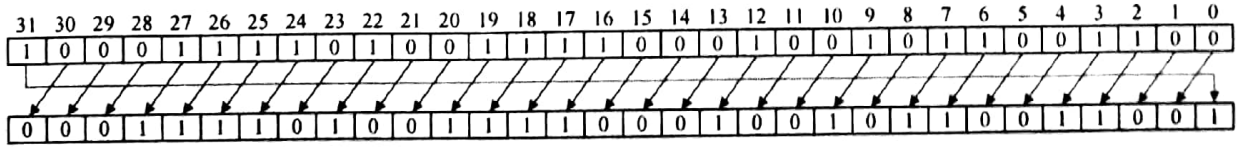
(1.0)c) Se o valor no registrador $f0 for considerado uma instrução da ISA MIPS, qual o mnemônico completo desta instrução?

(2.0)3) Algumas operações importantes de manipulação de bits não estão implementadas originalmente na ISA MIPS, entre elas estão as instruções de rotação dos bits de um registrador. Implemente eficientemente as pseudo-instruções abaixo:

(1.0)b) rorv $t0,$t1,$t2 # ROtate Right Variable $t0 = $t1 → $t2 rotaciona $t1 à direita de acordo com os 5 bits menos significativos de $t2



(1.0)a) rolv $t0,$t1,$t2 # ROtate Left Variable $t0 = $t1 ← $t2 rotaciona $t1 à esquerda de acordo com os 5 bits menos significativos de $t2



(2.0)4) Responda:

(1.0) a) Porque a média geométrica do tempo de execução normalizado é comumente utilizada para resumir os resultados de benchmarks?

(1.0) b) Porque o MIPS (Milhões de Instruções Por Segundo) não é uma boa medida de desempenho?

Boa Sorte!

PROVA 1 2017/1

GABARITO

1) $54 = -4

Procedimento VOLTA preenche a pilha com o quadrado dos numeros 0, 1, 2, ... d8

Procedimento NUM soma os valores presentes na pilha

a) $a2 = $\sum_{n=0}^{d8} n^2$ //

b) I = 6 + 8 + 9 x d8 + 1 + 5 + 4 x d8 + 2 (salvar / ler syscall)

I = 22 + 13 x d8 //

c) Mprograma = 23 x 4 = 92 bytes //

Mdados = (1 + 1 + d8) x 4 = 8 + 4 x d8 bytes //

(-4 ↓ 8)

2)

b) $f1 = 0x01800000

0 | 00000011 | 000 00...0

$2^{3-127} = 2^{-124}$

$f1 = 4,7019774 x 10^{-38} //

$f0 = $\sqrt{f1}$ = $\sqrt{2^{-124}}$ = 2^{-62} = 2^{65-127}

0 | 01000001 | 00...0

65

$f0 = 0x20800000 //

opcode=11 FMT=10 ft=0 fs=1 fd=0 func=4

a) 010001 10000 00000 00001 00000 000100

0x 4 6 0 0 0 8 0 4 //

c) $f0: 0010 0000 1000 0000 0000 0000 0000 0000
(rs | rt)

opcode = 0x8 → addi
rs = 4 → $a0 Imm = 0
rt = 0 → $zero

addi $zero, $a0, 0 //

3) a) rorv $t0, $t1, $t2 OBS: 32 - $t2 = 0 - $t2
Pq 5 bits!

| Técnica | Deslocar $t2 →
| | Desloca (32 - $t2) ←
| | Faz OU

[andi $at, $t2, 0x001F] No mais não precisa
sub $at, $zero, $at Pois sllv e srlv
sllv $at, $t1, $at só trabalha nos
srlv $t0, $t1, $t2 5 bits - significativos
or $t0, $t0, $at

b) rolv $t0, $t1, $t2
técnica Desloca $t2 ←
Desloca (32 - $t2) →
Faz OU

[andi $at, $t2, 0x001F]
sub $at, $zero, $at
srlv $at, $t1, $at
sllv $t0, $t1, $t2
or $t0, $t0, $at

4)

a) Porque gera um resultado consistente qualquer que seja a máquina base.

b) Porque não leva em consideração a complexidade das instruções. CISC x RISC