



Nome: GABARITO

Matrícula: / / / / / / / /

Prova 1

(2.0)1) Dado um processador MIPS com frequência de 100MHz, que necessita 1 ciclo de *clock* para a execução das instruções tipo R, 2 ciclos para as instruções tipo J, 3 ciclos para as instruções tipo I e 4 ciclos para as instruções de acesso à memória de dados (*load/stores*). Considerando que o desempenho seja calculado por

$$\eta = \frac{1}{Mem + t_{exec}}$$

0x00400000 0x34040064
0x00400004 0xAFA40000
0x00400008 0x00842020
0x0040000C 0x8FA40000
0x00400010 0x2084FFFE
0x00400014 0x1480FFFF

onde *Mem* é a quantidade de memória total usada (dados e programas) dada em kiB (kilobytes) e *t_{exec}* o tempo de execução dado em ms (milissegundos). Qual o desempenho deste processador na execução do programa dado?

(3.0)2) Neste semestre, o professor esqueceu de ministrar o conteúdo relativo à ISA e organização do Coprocessador 0 do processador MIPS, a unidade de Ponto Flutuante. Porém, conhecendo como é realizada a representação de um número real no padrão IEEE 754 em Precisão Simples e respeitando a convenção do uso dos registradores, implemente os procedimentos abaixo usando apenas as instruções da CPU principal do MIPS.

(1.0) a) jal ABS # o procedimento recebe em \$a0 um número em float e retorna em \$v0 seu módulo

(2.0) b) jal CMPL # o procedimento recebe em \$a0 e \$a1 dois números em float e retorna \$v0=1 se \$a0<=\$a1 ou \$v0=0 caso contrário

(2.0)3) Implemente as pseudo-instruções abaixo:

(1.0)a) seq \$t0,\$t1,\$t2 # se \$t1==\$t2 então \$t0=1 senão \$t0=0

(1.0)b) bnzal \$t0,LABEL # se \$t0!=0 então \$ra=PC+4 e PC=LABEL, senão PC=PC+4

(4.0)4) Em Sistemas Embarcados a relação entre custo (medido aqui como o tamanho da memória) e desempenho (medido aqui como o tempo de execução) é um ponto fundamental e que deve ser considerado durante todo o projeto. A memória RAM é um recurso extremamente caro nesses sistemas. Considere que você esteja trabalhando em uma rotina do sistema operacional que necessite que os 32 registradores do banco de registradores do processador MIPS sejam copiados para a memória RAM de dados, a partir do endereço dado por \$a0, quando a chamada jal BACKUP seja executada. Dica: Apenas os registradores \$at, \$k0 e \$k1 não precisam ter os valores corretos necessariamente preservados, isto é, podem ser modificados pela rotina BACKUP.

(2.0)a) Caso o custo não seja uma limitação, escreva um procedimento BACKUP que tenha o melhor desempenho possível.

(2.0)b) Caso o desempenho não seja uma limitação, escreva um procedimento BACKUP que ocupe no máximo 64 bytes de memória de programa.

GABANITO

1)

ciclos

0x0040 0000	34040064	ori \$a0,\$ZERO,100	3
AFA4 0000	SW \$a0, 0(\$SP)	T	4
0084 2020	add \$a0,\$a0,\$a0		1
8FA4 0000	LW \$a0, 0(\$SP)		4
2084 FFFE	addi \$a0,\$a0,-2		3
1480 FFE3	bne \$a0,\$ZERO,-5		3

Loop (\$a0=100; \$a0!=0; \$a0=\$a0-2)

Repete 50 vezes

$$n^{\circ}\text{ ciclos} = 3 + (4+1+4+3+3) \times 50 = 753$$

$$\text{memória} = \frac{6 \times 4}{\text{cod}} + 4 = 28 \text{ bytes} = 27,3438 \times 10^{-3} \text{ kB}$$

$$t_{exec} = 753 \times \frac{1}{100M} = 7,53 \times 10^{-6} \text{ s} = 7,53 \times 10^{-3} \text{ ms}$$

$$\eta = \frac{1}{27,3438 \times 10^{-3} + 7,53 \times 10^{-3}} = 28,67 \quad //$$

2)

- a) ABS: La \$VA, 0x7FFF FFFF
 and \$VA, \$VA, \$VA
 jr \$VA

Agrega reseta o bit de signo

b) Folha em ANEXO

A comparação em float pode ser feita diretamente, invertendo o resultado caso os números sejam negativos.

3) a) seq \$t0, \$t1, \$t2

add \$t0, \$zero, \$zero

beq \$t1, \$t2, PULA

add \$t0, \$zero, 1

PULA :

b) bneal \$t0, LABEL

beq \$t0, \$zero, PULA

jal LABEL

PULA ;

4) Folha em ANEXO

a) melhor desempenho é a solução trivial

que ocupa mais memória

b) usa Hilo como registradores temporários
modifica a instrução:

sw \$zero, 0(\$a8)

| Ligar de 4 em 4
vária de 1 em 1