

1ª Prova de MC722 – 2010

Nome:

RA:

1. (Cap 3) Traduza o trecho de programa em C abaixo usando a linguagem assembly MIPS. Diga o que este faz, dando um exemplo numérico e a respectiva execução (teste de mesa) da tradução em assembly. (3,5)

```
for (i = 0; i < n; i = i + 1) {
    rot (int v[], int n)
    {
        int i, j;
        for (i = 0; i < n; i = i + 1) {
            for (j = i - 1; j >= 0 && v[j] > v[j + 1]; j = j - 1) { troca (v, j);
            }
        }
    }
}
```

OBS: Suponha que a rotina *troca* (abaixo) seja uma rotina já traduzida. Não é necessário escrevê-la me MIPS.

```
troca (int v[], int k)
{
    int temp;
    temp = v[k];
    v[k] = v[k+1]
    v[k+1] = temp;
}
```

2. (Cap 4) Projete um somador *carry lookahead* de 4 bits, supondo que tenha disponível dois somadores *carry lookahead* de 2 bits. Calcule o tempo para executar a soma, supondo que cada porta tenha atraso t e *fan-in* 2. Compare com o tempo de atraso de um somador *ripple carry* de 4 bits. Comente. (2,5)
3. (Cap 1) Em um disco magnético existem dados armazenados que estão em constante rotação. Na média leva-se metade de uma revolução para que um desejado dado esteja debaixo da cabeça de leitura e gravação. Assuma que a rotação de um disco é 5400 RPM, qual o tempo médio para que isto ocorra. E se tivermos um disco de 7200 RPM ? (1,5)
4. (Cap 2) Estamos interessados em duas implementações de máquina, uma com e outra sem hardware específico de ponto-flutuante (FP). Considere um programa P, com a seguinte distribuição de operações:

Multiplicação em FP	– 10%
Soma em FP	– 15%
Divisão em FP	– 5%
Instrução inteiras	– 70%

Uma máquina com uma unidade de FP – MFP pode executar operações de ponto flutuante diretamente. Para isto necessita dos seguintes ciclos de clock, para cada classe:

Multiplicação em FP	– 6
Soma em FP	– 4

1ª Prova de MC722 – 2010

Nome:

RA:

Divisão em FP – 20
Instrução inteiras – 2

Uma máquina sem uma unidade de FP – MNFP e portanto não pode executar operações de ponto flutuante diretamente. Para isto, usa rotinas com apenas instruções inteiras, que levam 2 ciclos de clock cada para serem executadas, com o seguinte número de instruções:

Rotina de Multiplicação em FP – 30 instruções inteiras
Rotina de Soma em FP – 20 instruções inteiras
Rotina de Divisão em FP – 50 instruções inteiras

Se ambas as máquinas tem clock de 1000 MHz, encontre o número de instruções por segundo (MIPS) para ambas as máquinas. Comente. (2,5)

Fórmulas

$$\text{Performance}_X = (1 / \text{tempo de execução}_X)$$

$$\text{Performance}_X > \text{Performance}_Y \rightarrow \text{Tempo execução}_Y > \text{Tempo execução}_X$$

$$(\text{Performance}_X / \text{Performance}_Y) = (\text{Tempo execução}_Y / \text{Tempo execução}_X)$$

Elapsed time = tempo de tudo (CPU + I/O + etc.)

CPU time = user CPU time + system CPU time

$$T_{\text{CPU}}(\text{para programa}) = \text{períodos de clock da CPU} \times \text{período do clock}$$

$$T_{\text{CPU}}(\text{para programa}) = \text{períodos de clock da CPU} / \text{frequência do clock}$$

$$\text{n. de períodos da CPU (para programa)} = \text{n. de instruções} \times \text{n. médio de períodos por instrução (CPI)}$$

$$T_{\text{CPU}}(\text{para um programa}) = (\text{n. de instruções} \times \text{CPI}) / f_{\text{ck}}$$

$$T_{\text{CPU}}(\text{para um programa}) = (\text{n. de instruções} \times \text{CPI}) \times t_{\text{ck}}$$

$$\text{Períodos de clock da CPU} = \sum_{i=1}^n (\text{CPI}_i \times C_i)$$

$$\text{MIPS} = \text{n. de instruções} / (\text{tempo de execução} \times 10^6)$$