

NOME: _____

Nº: _____

TEÓRICA

As questões devem ser respondidas na própria folha do enunciado. As questões 1 a 4 são de escolha múltipla, e apenas uma das respostas está correcta, valendo 1 valor. Uma resposta errada desconta 1/3 de valor. As questões 5 e 6 valem 3 valores cada.

1. Qual das seguintes afirmações, que comparam os vários mecanismos de transferência de dados numa operação de Input/Output (I/O), é verdadeira?
 - ☐ Usando *polling* o CPU é responsável por verificar activamente alterações no estado do controlador de I/O, mas a transferência de dados não envolve o CPU.
 - ☐ A vantagem da utilização do mecanismo de DMA relativamente às interrupções é que o primeiro liberta o CPU da transferência de dados.
 - ☐ A utilização do mecanismo de interrupções implica a utilização do CPU para verificar activamente alterações no estado dos controladores de I/O.
 - ☐ O DMA liberta o CPU de qualquer intervenção em todas as operações de I/O.
2. O MIPS (Milhões de Instruções Por Segundo) é uma métrica frequentemente publicitada pelos fabricantes de máquinas para anunciar o desempenho dos respectivos CPUs. Qual das seguintes afirmações é verdadeira:
 - ☐ O MIPS nativo pode ser usado para avaliar o desempenho de CPUs com diferentes *Instruction Sets*.
 - ☐ O MIPS nativo não pode ser usado para avaliar o desempenho do mesmo programa em dois CPUs com o mesmo *Instruction Set*, o mesmo CPI e diferentes frequências do relógio.
 - ☐ O MIPS de pico corresponde à maior taxa de execução de instruções que o processador pode atingir, não sendo realista para programas comuns.
 - ☐ O MIPS nativo pode ser usado para avaliar o desempenho de dois programas diferentes no mesmo CPU.
3. Considere um programa P com 10^9 instruções, em que 20% das instruções implicam um acesso à memória. Este programa, executado na máquina M, exibe uma *miss rate* de instruções de 5%, de dados de 10% e um CPI_{CPU} de 1,5 ciclos incluindo o *hit time*. A máquina M tem uma *cache* com linhas de 4 palavras e um acesso à memória central tem uma latência de 30 ns seguida de um tempo de 5 ns por palavra. Sabendo que a frequência do relógio de M é de 500 MHz, qual dos seguintes valores corresponde ao tempo de execução de P em M:
 - ☐ 3,25 seg.
 - ☐ 10,0 seg.
 - ☐ 6,5 seg.
 - ☐ 13,0 seg.
4. Considere uma máquina com um espaço de endereçamento de 32 *bits*, uma cache de 512 Kbytes, linhas de 16 palavras, palavras de 2 bytes e mapeamento 4-way *set associative*. Qual a distribuição dos *bits* do endereço para seleccionar o byte correcto na cache:
 - ☐ Tag = 14; Índice = 12; Block Offset = 4; Byte Offset = 2
 - ☐ Tag = 15; Índice = 13; Block Offset = 3; Byte Offset = 1
 - ☐ Tag = 15; Índice = 12; Block Offset = 4; Byte Offset = 1
 - ☐ Tag = 16; Índice = 11; Block Offset = 4; Byte Offset = 1

Nº: _____

- [illegible]

NOME: _____

Nº: _____

PRÁTICA

As questões devem ser respondidas em folha separada. As questões 1 e 2 valem 3 valores cada. A questão 3 vale 4 valores.

1. Considere um programa escrito em *assembly* do MIPS com as características apresentadas na tabela, e assuma que o tempo de ciclo é de 50ns.

Instrução	Nº instruções	CPI
Lw	900	6
Div	1.000	8
add/sub	1.100	3
Beq	1.000	2

a) Qual o CPI global e o tempo de execução deste programa, se a *miss rate* de acesso às instruções for 8% e de acesso aos dados 3%, com uma *miss penalty* de 350ns?

b) Suponha que a capacidade da *cache* é aumentada, resultando numa *miss rate* de 6% para as instruções e 2% para os dados. Este aumento de capacidade resulta também num aumento do tempo de acesso à *cache*, implicando um aumento de 25% do CPI sem *misses*. Qual o CPI global e o tempo de execução do programa nestas novas condições?

2. Considere o programa abaixo escrito em *assembly* do MIPS para o SPIM. Explique a sua funcionalidade e a finalidade das instruções nos blocos 1, 2, 3 e 4.

```
.text
main: li $s0, 0
      li $s1, 0
      li $t0, 5

enq:  addi $s0, $s0, 1
      ble $s1, $t0, enq

      li $v0, 10 # exit
      syscall

      .ktext 0x80000080
      mfc0 $k0, $13           # bloco 1
      srl  $k0, $k0, 2        # bloco 2
      andi $k0, $k0, 0xf      # bloco 2
      addi $k0, $k0, -12      # bloco 3
      mfc0 $k1, $14           # bloco 4
      bne $k0, $0, ret        # bloco 3

      addi $s1, $s1, 1
      add $k1, $k1, 8         # bloco 4

ret:  rfe
      jr $k1                  # bloco 4
```

3. Codifique em *assembly* do MIPS o seguinte programa, seguindo as convenções estudadas (**NOTA:** Na codificação da função *media()*, utilize a seguinte atribuição de variáveis a registos: $n \equiv \$s0$, $c \rightarrow \text{data}[n] \equiv \$t1$, $c \rightarrow \text{Mi} \equiv \$t3$, $c \rightarrow \text{Me} \equiv \$t6$)

```
struct cor {
int  data[100];
int  Mi;
int  Me;  };

main() {
    struct cor cm;
    int      med;
    med = media(&cm, 50, 0x3ff);
}

int media (struct cor *c, int min, int max) {
    int n;
    for (n=0; n <= 99; ++n)
        if (c->data[n]<max && c->data[n]>=min)
            c->Mi = c->Mi + c->data[n];
        else
            c->Me = c->Me + c->data[n];
    return ((c->Mi + c->Me)/100);
}
```