

Preencher na
correção1: _____
2: _____
3: _____
4: _____
5: _____
6: _____
7: _____
8: _____
9: _____
F: _____

NOME: _____ Nº: _____

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. A questão 7 (2 valores) corresponde à componente de avaliação contínua; a nota final desta componente será o máximo entre esta questão e a classificação obtida ao longo do semestre

1. O código *assembly* de um programa compilado para uma arquitectura RISC, comparado com o mesmo programa compilado para uma arquitectura CISC, caracteriza-se geralmente por:

- As instruções endereçarem um elevado número de operandos em memória
- O número de instruções que compõem o programa ser muito menor
- Serem necessárias mais instruções para realizar operações complexas
- Serem utilizados modos de endereçamento sofisticados.

2. Qual das seguintes sequências de 32 *bits* corresponde à codificação em binário da instrução `addi $t0, $v0, -128` do MIPS?

- 0x2048FF80
- 0x2102FF80
- 0x2048FF7F
- 0x2102FF7F

3. A localidade temporal é uma propriedade exibida pela maioria dos programas. Podemos afirmar que:

- A organização da memória tira partido desta característica agrupando acessos a um nível inferior da hierarquia (ex.: acesso da *cache* à memória central) em blocos de palavras, em vez de aceder palavra a palavra.
- É responsável pelo efeito acelerador da hierarquia da memória, ao contribuir para que blocos de dados e/ou instruções acedidos repetidamente pelo processador sejam, com grande probabilidade, encontrados na *cache* após o primeiro acesso.
- Se traduz pelo facto de determinadas variáveis e/ou porções de código serem armazenadas em endereços consecutivos da memória.
- Se traduz pelo facto de que a grande maioria dos blocos de dados e código acedidos por um programa se situarem num intervalo limitado do espaço de endereçamento.

4. Considere uma máquina com um espaço de endereçamento de 64 *bits*, 8 MBytes de *cache* organizada em linhas de 32 palavras e cada palavra com 64 *bits*. Se esta *cache* utilizar mapeamento directo, então o índice é constituído por:

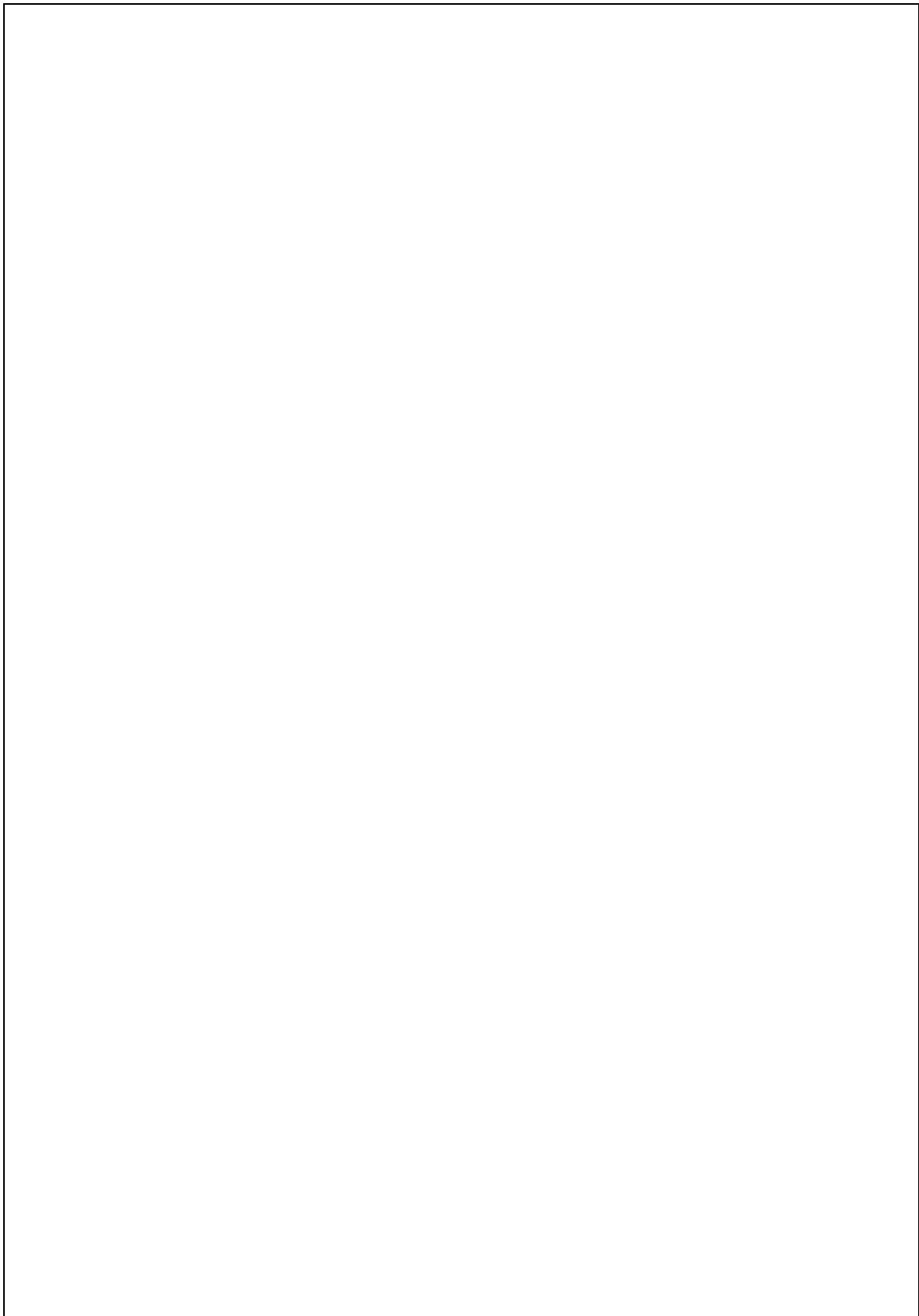
- 18 *bits*.
- 15 *bits*.
- 0 *bits*.
- 14 *bits*.

NOME: _____ **Nº:** _____

5. Considerando o *datapath* de ciclo único do MIPS (ver figura anexa) indique, justificando, se é possível suportar a instrução `jr reg` sem alterações ao mesmo; pressupondo, no entanto, alterações ao *control path*. Suponha que a instrução é do tipo R, com `reg` especificado no campo `RS`. Esta instrução, suportada pelo MIPS, provoca um salto para o endereço contido no registo `reg`.

NOME: _____ **Nº:** _____

6. Discuta as diferentes estratégias de escrita na *cache* (*write-through*, *write-back*), indicando as respectivas vantagens e desvantagens.



NOME: _____ **Nº:** _____

Apenas para os alunos que, justificadamente, não fizeram a componente teórico-prática.

7. O tempo de execução de um programa num determinado processador pode ser previsto calculando o produto de 3 métricas: número de instruções (#I), ciclos por instrução (CPI) e o período do relógio (T_{cc}). Discuta os vários factores que influenciam cada uma destas métricas (ex.: algoritmo, compilador, conjunto de instruções, microarquitECTURA, tecnologia de desenvolvimento do processador).

NOME: _____

Nº: _____

Estas questões devem ser respondidas em folha separada. As questões 8 e 9 valem 5 e 3 valores respectivamente.

8. Na linha inferior da tabela é apresentado o código binário e *assembly* do MIPS gerado para a função apresentada em C na linha superior de mesma tabela

```
void dobrar(int n, int d[])
{
  do {
    n--;
    d[n] *= 2;
  } while (n>0);
}
```

```
addi $a0, $a0, -1

sll $t0, $a0, 2

000000 00101 01000 01000 00000 100000
100011 01000 01001 0000 0000 0000 0000

sll $t1, $t1, 1

sw $t1, 0($t0)

000111 00100 00000 1111 1111 1111 1001

000000 11111 00000 00000 00000 001000
```

- a) Converta as 4 instruções em binário para *assembly*.
- b) Indique a funcionalidade de cada instrução *assembly*, associando-a à instrução C correspondente.
- c) Escreva em *assembly* do IA32 o código correspondente à instrução **d[n]*=2;**
9. A execução de 3.000.000 de instruções numa máquina com uma frequência de 3 GHz demorou 0,052 segundos. Este programa consiste em 20% de instruções que realizam acessos a dados em memória e exibe uma *miss rate* de instruções de 3% e de dados de 10%. O CPI_{CPU} observado foi de 2 ciclos por instrução.
- a) Qual o CPI médio para a execução deste programa?
- b) Qual o valor da *miss penalty* expressa em segundos?
- c) Supondo que é possível diminuir a *miss penalty* para metade, mantendo tudo o resto constante, qual o novo tempo de execução?