

**Exame Modelo - Duração: 20 minutos**

Notas:

- Em cada uma das perguntas existe apenas **uma** resposta correta
- Cada **três** respostas erradas anulam uma resposta correta
- Nota mínima de 7.5 valores

Nº: \_\_\_\_\_ Nome: \_\_\_\_\_

**1) O objetivo da filosofia CISC é**

- (a) Fornecer instruções e modos de endereçamento simples
- (b) Reduzir o fosso entre linguagens de programação de alto nível e o *hardware*
- (c) Colocar a complexidade nos compiladores

**2) Os sinais de controlo do fluxo de informação no CPU**

- (a) Determinam a velocidade a que a informação circula
- (b) Dependem do tipo de instrução e ativam apenas uma parte do circuito
- (c) Não dependem do tipo de instrução, mas exigem multiplexadores

**3) As instruções aritméticas e lógicas em MIPS**

- (a) Permitem usar um endereço de memória ou um registo como destino da operação
- (b) Usam sempre três registos como argumentos
- (c) São instruções de formato I

**4) Num processador MIPS com pipeline simples os sinais de controlo**

- (a) São determinados na fase IF, pois dependem do *opcode* da instrução
- (b) São determinados na fase ID, pois dependem do *opcode* da instrução
- (c) São usados apenas na fase EX, pois esta é a fase em que a instrução é executada

**5) A técnica de *pipelining***

- (a) Diminui o tempo de execução de uma instrução
- (b) Diminui o tempo de execução de um programa
- (c) Possui um ciclo de relógio determinado pela instrução mais lenta

**6) Em *pipelining* o problema da maior latência nos acessos à memória**

- (a) Pode ser resolvido com *forwarding*
- (b) Pode ser resolvido com *branch predication*
- (c) Exige sempre a paragem do *pipeline*

**7) A previsão dinâmica de saltos é**

- (a) Implementada pelo compilador recorrendo à análise da estrutura do programa
- (b) Implementada em hardware recorrendo a memória indexada pelo endereço do salto
- (c) Menos adaptativa do que a previsão estática de saltos

**8) Numa cache *n-way set associative* um bloco pode ser colocado**

- (a) Em qualquer um dos conjuntos e dentro desse conjunto numa posição mapeada diretamente
- (b) Numa qualquer entrada da cache pois esta não está dividida por conjuntos
- (c) Numa qualquer entrada de um conjunto mapeado diretamente

**9) Numa cache *fully associative* um bloco pode ser colocado**

- (a) Em qualquer um dos conjuntos da cache e dentro desse conjunto numa posição mapeada diretamente
- (b) Numa qualquer entrada da cache pois esta não está dividida por conjuntos
- (c) Numa qualquer entrada de um conjunto mapeado diretamente

**10) A política de escrita de dados na memória *write-through***

- (a) Possui uma fraca performance sem o uso dum *write buffer*
- (b) Exige sempre um acesso à memória em todas as operações de escrita
- (c) Minimiza o número de colisões de blocos na cache