

Exame Época Normal - Duração: 60 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 RISC 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 no *hardware*

2) A métrica CPI (*clock cycles per instruction*)

- (a) Indica o tempo médio de execução de todas as instruções de um programa
- (b) Indica o tempo de execução da instrução mais lenta do programa
- (c) Não serve para comparar duas implementações do mesmo ISA (*instruction set architecture*)

3) Admita que o valor 3 505 468 161₍₁₀₎ (11010000 11110001 00110011 00000001₍₂₎) foi atribuído à variável *unsigned int x* numa arquitetura *little endian*. De que forma os seus 4 bytes são armazenados sequencialmente em memória?

- (a) 00000001 00110011 11110001 11010000
- (b) 10000000 11001100 10001111 00001101
- (c) 11010000 11110001 00110011 00000001

4) Em MIPS, qual das seguintes instruções adiciona o valor 12 ao valor atual do registo \$t6?

- (a) add \$t6, 12
- (b) addi \$t6, \$0, 12
- (c) addiu \$t6, \$t6, 12

5) Em MIPS, qual dos seguintes modos de endereçamento é usado nas instruções de salto condicional?

- (a) Imediato, onde o operando é uma constante indicado na própria instrução
- (b) Base, onde o operando está num determinado endereço de memória cujo endereço é a soma do registo e de uma constante indicados na instrução
- (c) Relativo ao PC (*program counter*), onde o endereço de salto é a soma do PC e de uma constante indicada na instrução

6) A técnica de *pipelining*

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

7) 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*

8) Em MIPS com *pipelining*, uma dependência entre instruções (*data hazard*) no programa é detetada quando

- (a) O registo de destino no pipeline EX/MEM é igual a algum dos registos fonte no pipeline IF/ID
- (b) O registo de destino no pipeline EX/MEM é igual a algum dos registos fonte no pipeline ID/EX
- (c) O registo de destino no pipeline MEM/WB é igual a algum dos registos fonte no pipeline IF/ID

9) 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

10) Numa hierarquia de memória

- (a) O custo por bit aumenta à medida que nos afastamos do CPU
- (b) A velocidade aumenta à medida que nos afastamos do CPU
- (c) A capacidade de armazenamento aumenta à medida que nos afastamos do CPU

11) 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

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

- (a) Possui uma fraca performance sem o uso dum *write buffer*
- (b) É mais complexa de implementar em *hardware* mas mantém a coerência entre cache e memória
- (c) Minimiza o número de colisões de blocos na cache

13) Ao acrescentar mais níveis de cache numa hierarquia de memória

- (a) Pretende-se reduzir o custo do acesso à memória
- (b) Organizam-se as caches por ordem decrescente de tamanho a partir do CPU
- (c) Colocam-se todos os níveis adicionais dentro do CPU

14) A tradução de endereços virtuais em endereços físicos

- (a) Exige dois acessos à memória se o endereço a traduzir não estiver na TLB
- (b) Exige sempre dois acessos à memória mesmo com o endereço traduzido presente na TLB
- (c) Exige a tradução de todos os bits do endereço virtual

15) A tabela de páginas usada no mecanismo de memória virtual

- (a) É necessária porque a colocação das páginas em memória é *fully associative*
- (b) É necessária porque a colocação das páginas em memória é *direct mapped*
- (c) Tem um tamanho independente do tamanho da página

16) O desenho da memória em *interleaving*

- (a) Permite aceder a n palavras em simultâneo num único acesso à memória
- (b) Permite aceder a n palavras na memória com uma abordagem similar ao *pipelining* na execução das instruções
- (c) Nenhuma das anteriores

17) A sequência “TLB hit -> Page Table miss -> Cache miss” é

- (a) Possível, TLB falha e a página não está em memória, nem na cache
- (b) Impossível, porque não pode haver tradução de endereços se a página não está em memória
- (c) Possível, TLB falha e a página é encontrada na memória

18) A velocidade da memória afeta a decisão sobre qual o tamanho dos blocos na cache. Qual dos seguintes princípios de design pode ser considerado válido?

- (a) Quanto menor for a latência da memória, maior o tamanho do bloco
- (b) Quanto maior for a largura de banda da memória, menor é o tamanho do bloco
- (c) Quanto maior for a largura de banda da memória, maior é o tamanho do bloco

19) Aumentar a associatividade de uma cache

- (a) Diminui o *miss rate*
- (b) Aumenta o *miss rate*
- (c) Diminui custo da pesquisa

20) Numa hierarquia de memória com múltiplos níveis de cache podemos afirmar que

- (a) As caches L1 têm como principal preocupação o *hit time* e as caches L2 o *miss penalty*
- (b) As caches L1 têm como principal preocupação o *miss penalty* e as caches L2 o *hit time*
- (c) Nenhuma das anteriores