

Universidade de Aveiro
Departamento de Electrónica, Telecomunicações e Informática
Arquitectura de Computadores 2 – Ano Letivo 2012/13 – Exame Final

Nome: _____

Notas Importantes!

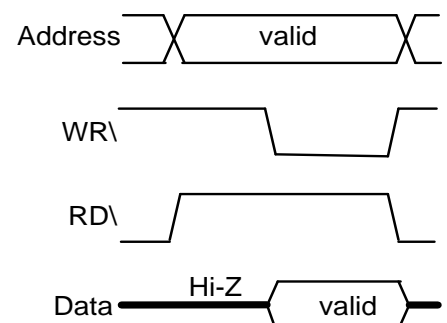
1. Pode responder a cada questão com uma opção que deverá assinalar com um X na tabela ao lado. Por cada resposta errada será descontado, à cotação global, 1/3 da cotação da respectiva pergunta.
2. Durante a realização do teste não é permitida a permanência na sala de calculadoras, telemóveis ou outros dispositivos electrónicos.
3. Cotações: Grupo I – cada 0.4 valores Grupos II e III – cada 0.6 valores.

Grupo I

1. Um microcontrolador é um dispositivo programável que:
 - a) disponibiliza através dos portos de I/O a generalidade dos sinais dos barramentos do microprocessador para ligação direta a sensores e atuadores de um sistema embutido
 - × b) integra num único circuito integrado, microprocessador, memória e periféricos
 - c) devido a restrições de custos não utiliza mecanismos de multiplexagem para partilha de pinos entre diversas funcionalidades
 - d) todas as restantes respostas estão corretas
2. A função de um *bootloader* num sistema baseado num microcontrolador é:
 - × a) transferir o código executável do *host PC* usado no desenvolvimento, para o sistema *embedded* para posterior execução
 - b) realizar a compilação do *software* e iniciar a sua execução após o *reset* do sistema
 - c) executar o *software* e auxiliar na sua depuração através da introdução de *breakpoints*, visualização do conteúdo de registos e de posições de memória
 - d) interagir com o *cross-compiler* para efeitos de depuração da aplicação
3. O modelo de programação de um periférico especifica:
 - a) o sub-conjunto de instruções *assembly* do CPU suportadas por esse periférico
 - b) os sinais elétricos usados na ligação do periférico a dispositivos externos, tais como sensores e atuadores
 - c) as arquiteturas e as ferramentas de desenvolvimento com as quais o periférico pode ser usado
 - × d) o conjunto de registos, respetivos campos e modos de acesso suportados pelo periférico
4. A descodificação de endereços consiste em:
 - × a) determinar, em função do endereço presente no barramento, qual o periférico ou memória que deve ser selecionada
 - b) representar um endereço em binário de forma a utilizar o menor número possível de linhas do barramento
 - c) determinar em função do endereço gerado pelo periférico, qual o CPU ou memória que deve ser selecionada
 - d) preencher a totalidade do espaço de endereçamento do processador com memórias e periféricos

	a	b	c	d
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				
16				
17				
18				
19				
20				
	a	b	c	d
21				
22				
23				
24				
25				
26				
27				
28				
29				
30				
31				
32				
33				
34				
35				
36				
37				
	a	b	c	d
38				
39				
40				

5. O diagrama temporal da figura do lado representa um ciclo de:
 - a) leitura de um dispositivo em que os sinais de controlo usam lógica negativa
 - b) leitura de um dispositivo em que os sinais de controlo usam lógica positiva
 - × c) escrita num dispositivo em que os sinais de controlo usam lógica negativa
 - d) escrita num dispositivo em que os sinais de controlo usam lógica positiva



6. O sinal de seleção “Sel” (activo alto) de um porto mapeado na gama de endereços **0x0000 . . . 0x07FF** de um processador com um espaço de endereçamento de 16 bits pode ser obtido através da expressão:

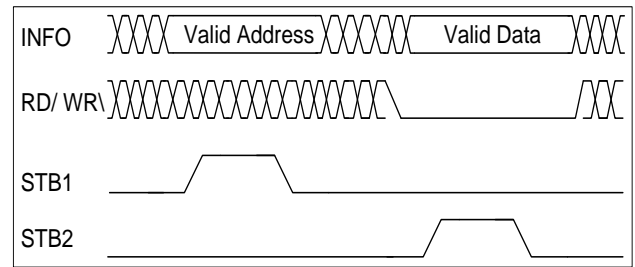
- a) $Sel = \prod_{i=0}^{11} A_i$ b) $Sel = \prod_{i=11}^{15} A_i$ c) $Sel = \prod_{i=0}^{11} A_i \setminus$ x d) $Sel = \prod_{i=11}^{15} A_i \setminus$

7. Numa transferência assíncrona:

- a) assume-se que o dispositivo externo responde à velocidade do CPU e, consequentemente, não existem sinais de protocolo envolvidos no *handshake* da transação
b) o CPU prolonga o ciclo de leitura/escrita por um ou mais ciclos de relógio, se for ativado um sinal de protocolo gerado pelo dispositivo externo
x c) o CPU prolonga o ciclo de leitura/escrita até que o dispositivo externo sinalize através de sinais de protocolo que a operação pretendida foi completada
d) nenhuma das restantes respostas está correta

8. A figura do lado corresponde ao diagrama temporal de:

- x a) uma operação de escrita numa transferência síncrona com dados e endereços multiplexados numa configuração micro-ciclo
b) uma operação de leitura numa transferência síncrona com dados e endereços não multiplexados numa configuração *merged*
c) uma operação de leitura numa transferência assíncrona com dados e endereços multiplexados numa configuração micro-ciclo e *strokes* independentes
d) uma operação de escrita numa transferência assíncrona com dados e endereços não multiplexados numa configuração *merged* e *strokes* independentes



9. A técnica de entrada/saída de dados por interrupção:

- a) permite transferir eficientemente (com elevado *throughput*) grandes quantidade de informação
x b) permite mascarar a latência do periférico
c) consiste na execução de um ciclo de *polling* que é interrompido quando o periférico estiver pronto para realizar a transferência
d) consiste na interrupção do periférico sempre que o CPU pretende transferir informação

10. Quando é usada a técnica de entrada/saída de dados por DMA:

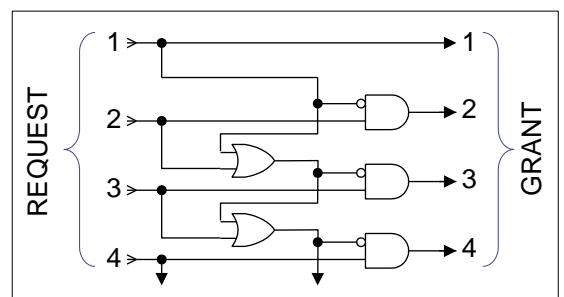
- a) o CPU verifica através de um ciclo de *polling* ao registo de dados do controlador de DMA se a transferência já foi concluída
b) o periférico faz um pedido de interrupção ao CPU quando estiver pronto para transferir os dados
c) o CPU configura o controlador de DMA que fará por *software* a transferência propriamente dita
x d) o CPU configura o controlador de DMA que fará por *hardware* a transferência propriamente dita

11. Um árbitro de um barramento *multimaster* baseado em prioridades FIFO garante:

- a) que é sempre servido o *master* de maior prioridade com pedido pendente de atribuição de barramento
x b) a ausência de fenómenos de *starvation*
c) que a atribuição do barramento é fixada pela ordem temporal inversa com que os *masters* fazem os seus pedidos
d) que a atribuição do barramento é fixada pela ordem temporal com que os *slaves* fazem os seus pedidos

12. A figura do lado representa um circuito que pode ser usado para implementar os mecanismos de *Acesso* e *Seleção* de um árbitro de um barramento *multimaster* baseado:

- x a) em prioridades fixas, sendo a entrada *REQUEST1* a mais prioritária
b) no critério *round-robin* (atribui frações de tempo para cada *master* em partes iguais e de forma circular)
c) em prioridades fixas, sendo a entrada *REQUEST4* a mais prioritária
d) no critério *First-Come-First-Served*



13. Suponha que pretende interligar, através de um protocolo/interface série, dois sistemas computacionais que distam de alguns metros numa linha de produção com elevados níveis de interferência electromagnética, um ligado a um sensor e outro correspondendo a um computador de controlo. O standard mais adequado a este cenário de aplicação é:

- a) SPI b) I2C x c) CAN d) RS232

14. No standard RS-232 um dos tipos de erro de comunicação que é detectado é o erro de *framing*. Esse erro ocorre quando o receptor:

- a) recebe um número de bits a "1" que não corresponde à paridade programada
- b) recebe um bit de paridade diferente do programado
- c) deteta um número de bits no campo de dados diferente do programado
- × d) recebe como *stop* bit um bit com o nível lógico "0"

15.A interface SPI permite a comunicação entre:

- a) um *master* e vários *slaves*, numa ligação em *daisy-chain*, sendo o sinal de relógio implícito
- ✗ b) um *master* e vários *slaves*, numa ligação com sinais de selecção individuais, sendo o sinal de relógio explícito
- c) vários *masters* e vários *slaves*, numa ligação com sinais de selecção individuais, sendo o sinal de relógio implícito
- d) vários *masters* e vários *slaves*, numa ligação com sinais de selecção individuais, sendo o sinal de relógio explícito

16. Na interface I2C o *master* selecciona o *slave* com quem vai comunicar através de:

- a) um sinal de seleção que ativa antes de iniciar a transferência
- ✗ b) informação transmitida na linha de dados
- c) um sinal específico de seleção através do qual é transferido o endereço desse *slave*
- d) um barramento de endereços de 7 bits a partir do qual cada dispositivo descodifica o seu próprio endereço

17. O *dirty bit* é usado numa *cache* com política de escrita:

- a) *write-through* para indicar que a informação armazenada no respectivo bloco foi alterada
- b) *write-through* para indicar que o respectivo bloco não está a ser usado
- ☒ c) *write-back* para indicar que a informação armazenada no respectivo bloco foi alterada
- d) *write-back* para indicar que o respectivo bloco não está a ser usado

18. Numa *cache* com associatividade de 4 de 8 kBytes, e 128 blocos, o número de comparadores necessários para comparar o campo *tag* de um endereço de acesso à memória é:

- ☒ a) 4
 ☐ c) 8096
☐ b) 32
 ☐ d) nenhuma das restantes respostas está correcta

19. A técnica de memória virtual permite:

- a) a utilização de armazenamento secundário para aumentar a dimensão aparente da memória física do sistema
- b) que o espaço de endereçamento de um processo exceda o limite de memória física disponível
- c) implementar mecanismos de proteção devido à independência dos espaços de endereçamento de cada processo
- × d) todas as restantes respostas estão correctas

20. A tradução de endereços virtuais em endereços físicos consiste na tradução do:

- x a) *virtual page number* no *physical page number* e sua justaposição com o *page offset* do endereço virtual
 b) *physical page number* no *virtual page number* e sua justaposição com o *page offset* do endereço físico
 c) *physical page offset* no *virtual page offset* e sua justaposição com o *page number* do endereço físico
 d) *virtual page offset* no *physical page offset* e sua justaposição com o *page number* do endereço virtual

Grupo II

21. Num espaço de endereçamento de 16 *bits*, um decodificador implementado através da expressão lógica " $Sel\ = A15 + A13\ + A12 + A11\$ ", decodifica a(s) seguinte(s) gama(s) de endereço(s):
- a) 0x2000 a 0x27FF, 0x6000 a 0x67FF
 - × b) 0x2800 a 0x2FFF, 0x6800 a 0x6FFF
 - c) 0x9000 a 0x97FF, 0xD000 a 0xD7FF
 - d) 0x9800 a 0x9FFF, 0xD800 a 0xDFFF
22. Num porto de entrada constituído por *buffers tri-state* (na ligação entre os sinais externos e o barramento de dados do processador), o sinal de *Enable* ativo alto é obtido a partir dos sinais *Sel* e *RD* (ambos ativos baixos) de acordo com a expressão lógica:
- × a) $Enable = Sel\ . RD\$
 - b) $Enable = (Sel\ . RD\)\$
 - c) $Enable = Sel\ + RD\$
 - d) $Enable = (Sel\ + RD\)\$
23. Num sistema de interrupções com uma única linha e identificação da fonte por *software*, a sequência de operações efetuada durante o atendimento a uma interrupção é, pela ordem indicada, a seguinte:
- a) identificação da fonte, determinação do endereço da RSI, salvaguarda do endereço de retorno, salto para a RSI
 - b) salto para a RSI, identificação da fonte, salvaguarda do endereço de retorno
 - c) determinação do endereço da RSI, identificação da fonte, salvaguarda do endereço de retorno, salto para a RSI
 - × d) salvaguarda do endereço de retorno, salto para a RSI, identificação da fonte
24. Na organização do sistema de interrupções designada por "interrupções vetorizadas", o processador obtém o vector que identifica o periférico gerador da interrupção:
- × a) por *hardware* num ciclo de *interrupt acknowledge* durante o qual o periférico gerador da interrupção o coloca no barramento de dados
 - b) por *hardware* através da leitura do valor presente no barramento de endereços uma vez que quando o periférico ativa a linha de interrupção coloca simultaneamente nesse barramento o seu vetor
 - c) por *software* na rotina de serviço à interrupção "questionando" cada um dos periféricos do sistema
 - d) por *software*, antes de chamar a rotina de serviço à interrupção "questionando" cada um dos periféricos do sistema
25. Numa transferência por DMA, em modo bloco, quando o controlador de DMA pretende dar início a uma transferência:
- a) ativa o sinal *busreq* durante um número fixo de ciclos de relógio, efetuando de seguida a transferência
 - × b) ativa o sinal *busreq*, efetuando a transferência logo que se torne no *bus master*
 - c) gera uma interrupção que é interpretada pelo CPU como um pedido de cedência dos barramentos; a transferência é efetuada quando o DMA reconhecer a ativação do sinal *busgrant*
 - d) sinaliza o CPU, através da linha *busreq*, que vai dar início à transferência e inicia-a de imediato; o sinal *busgrant* é utilizado pelo CPU para suspender a atividade do DMA
26. Considere um barramento paralelo multiplexado, constituído por 32 linhas informação. Sobre este barramento pretende implementar-se um protocolo de comunicação, do tipo micro-ciclo, que apresenta um espaço de endereçamento de 64 bits e 32 bits de dados. Para completar uma transação sobre este barramento, o número mínimo de ciclos necessários e de *qualifiers* codificados é:
- × a) 2 ciclos e 2 *qualifiers*
 - b) 2 ciclos e 3 *qualifiers*
 - c) 3 ciclos e 2 *qualifiers*
 - d) 3 ciclos e 3 *qualifiers*

27. Considere um *timer* de 16 bits, com reset assíncrono, com uma frequência de entrada de 1MHz, que funciona, em modo alternado, com duas constantes de divisão KA e KB. Utilizando o *timer* como divisor de frequência, e supondo que o tempo a "1" do sinal é determinado pela constante KB, para se obter à saída um sinal com um período de 10ms e *duty-cycle* de 40%, a constante KA deverá valer:

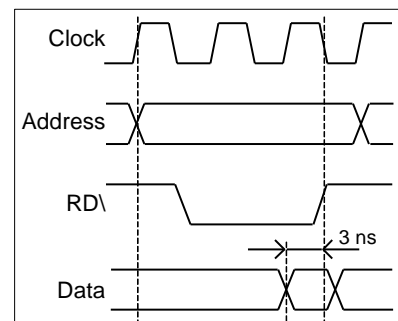
a) 4000

b) 3999

x c) 6000

d) 5999

- 28.** Considere um CPU a funcionar a uma frequência de 100 MHz ligado a uma memória com um tempo de acesso (referenciado ao seu sinal CE) de 15 ns. O CPU suporta transferências de tipo semi-síncrono, estando o ciclo de leitura, sem *wait-states*, representado na figura ao lado (note o tempo de *setup* de 3ns). No barramento de dados que interliga o CPU e a memória existe um *buffer* com um tempo de propagação de 4 ns e o decodificador que gera o sinal de selecção para a memória apresenta um atraso de propagação de 6 ns. Para que este sistema funcione correctamente o número de *wait-states* que é necessário introduzir no ciclo de leitura é:



✗ a) 0

c) 2

b) 1

d) nenhuma das restantes respostas está correcta

29. Considere um sistema baseado num CPU a funcionar a uma frequência de 200 MHz com um CPI médio de 2 que processa por interrupção eventos externos periódicos. Se o *overhead* máximo do atendimento a uma interrupção for de 10 **ciclos de relógio**, e a rotina de serviço à interrupção tiver 20 **instruções**, a máxima frequência a que esses eventos podem ocorrer é, aproximadamente:

a) 1 MHz

b) 2 MHz

x c) 4 MHz

d) 8 MHz

- 30.** O trecho de código *assembly* MIPS (com *branch delay slot*) que se apresenta ao lado envia 4000 *words* para um periférico. Admitindo que este código é executado num processador de 400 MIPS (executa 4×10^8 instruções/seg) e que o ciclo de *polling* é efetuado em média 9 vezes, a taxa de transferência média que se obtém é, aproximadamente:

a) 4 Mbytes/s

b) 8 Mbytes/s

c) 16 Mbytes/s

× d) 32 Mbytes/s

```

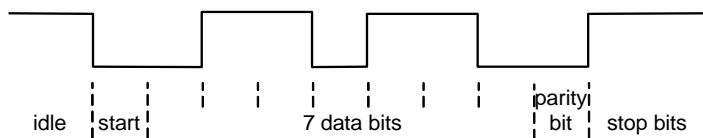
send:  la      $a0,mem_buf
       la      $a1,io_addr
       li      $a2,4000

poll:  lw      $t0,0($a1)
       lui     $t1,0x1000
       and     $t0,$t0,$t1
       beq     $t0,$zero,poll
       nop

       lw      $t0,0($a0)
       addi    $a2,$a2,-1
       sw      $t0,4($a1)
       bne     $a2,$zero,poll
       addiu   $a0,$a0,4
       jr      $ra
       ori     $v0,$zero,0

```

31. Um dispositivo com interface RS232 e configurado para transmitir com 7 *bits* de dados, paridade par e 2 *stop bits*, produz a trama seguinte que é recebida por outro dispositivo RS232 incorrectamente configurado para 8 *bits* de dados, paridade ímpar e 1 *stop bit* mas com o mesmo *baud rate*. Nestas circunstâncias o recetor:



- a) vai detetar uma trama inválida devido a um número incorrecto de *stop bits*
 - b) vai detetar um erro de paridade
 - c) vai detetar um erro de paridade e uma trama inválida devido a um número incorreto de *stop bits*
 - × d) não vai detetar qualquer erro
32. Suponha que dispõe de 32 circuitos de memória de 1Mx4. Usando todos estes circuitos é possível construir um módulo de memória de:
- × a) 2M × 64
 - b) 8M × 32
 - c) 8M × 8
 - d) 2M × 8
33. Considere um processador com um espaço de endereçamento de 32 bits e uma memória *cache* com associatividade de 4, de 256 kByte e blocos de 32 bytes. A dimensão, em bits, dos campos *tag*, *set* e *byte* é:
- × a) *Tag*: 14; *Set*: 13; *Byte*: 5
 - b) *Tag*: 16; *Set*: 11; *Byte*: 5
 - c) *Tag*: 16; *Set*: 13; *Byte*: 3
 - d) *Tag*: 9; *Set*: 18; *Byte*: 5
34. Num dado processador um endereço virtual é representado com 32 bits, dos quais 11 bits são usados para o *page offset*. Esse processador é usado num sistema com 2 GByte de memória física. Nestas circunstâncias o número de páginas virtuais e físicas é, respectivamente:
- a) 1M e 2k
 - b) 1M e 2M
 - c) 2M e 2k
 - × d) 2M e 1M
35. Na técnica de memória virtual, o número de entradas do TLB é:
- × a) dependente da implementação sendo sempre muito inferior ao número de entradas da *page table*
 - b) igual ao número de entradas da *page table*
 - c) igual ao número máximo de páginas virtuais
 - d) igual ao número máximo de páginas virtuais de memória usadas pelo processo em execução
36. Num sistema que suporta um nível de *cache* e memória virtual:
- a) no espaço de armazenamento secundário (disco) estão armazenadas as páginas de memória virtual mais recentemente acedidas e na memória *cache* estão armazenados os blocos dessas páginas mais recentemente acedidos
 - b) os blocos da *cache* e as páginas de memória são tipicamente da mesma dimensão
 - × c) enquanto é efectuado o processamento de um *page fault* de um processo, o processador pode estar ocupado a executar outro processo
 - d) todas as restantes respostas estão correctas

37. Num sistema de memória virtual de 32 bits em que a dimensão da página é de 4 kBytes, a *page table* de cada processo tem um total de:

- × a) 1M entradas
- b) 2M entradas
- c) 4k entradas
- d) 32 entradas

Grupo III

Um sistema possui um espaço de endereçamento virtual de 4 GBytes, páginas de memória de 4 kBytes e 1 GByte de memória física. Considere também:

- que num dado instante está a executar um processo cujo *Page Table Register* possui o valor **0x01230000**
- que cada entrada da *page table* possui 32 bits, está alinhada em endereços múltiplos de 4 e contém a seguinte informação e *flags*

<i>Valid, Dirty, Read, Write, Execute flags</i> [31:27]	Bits não usados [26:18]	PPN [17:0]
--	----------------------------	---------------

- o conteúdo de algumas posições da memória principal a seguir indicados

Endereço	Valor
...	
0x0123001C	0xB000000C
0x01230020	0xF0000007
0x01230024	0xD8000005
0x01230028	0xA000000E
...	

38. Num acesso à memória, o CPU produz o endereço **0x0000900C**. Qual o endereço físico em que é traduzido?

- a) **0x0000E00C**
- b) **0x00005007**
- × c) **0x0000500C**
- d) nenhuma das restantes respostas está correta

39. O processo em execução pode aceder ao endereço virtual **0x0000A014** para:

- a) leitura e escrita
- × b) leitura
- c) escrita
- d) nenhuma das restantes respostas está correta

40. No caso de ser necessário substituir a página física que contém o endereço físico **0x0007001** por uma nova página, haverá necessidade de a salvar previamente?

- a) não, porque a respetiva *flag Write* não está ativa
- × b) não, porque a respetiva *flag Dirty* não está ativa
- c) sim, porque a respetiva *flag Write* está ativa
- d) sim, porque a respetiva *flag Dirty* está ativa

Área de Rascunho