1º Teste teórico 2017/2018 - AC2

Grupo I

1. Na arquitetura de um sistema computacional, o *Data Bus* permite:
   a. identificar, na memória ou num periférico, a origem/destino da informação a transferir.
   b. especificar o tipo de operação efetuada sobre a memória.
   c. transferir informação entre dispositivos periféricos e os registos internos do CPU.
   d. transferir o código máquina das instruções para o *program counter*.

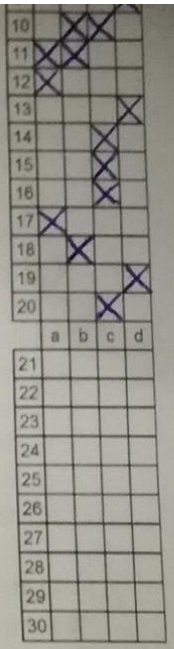
2. Dizer-se que num sistema computacional a memória apresenta uma organização do tipo *word-addressable* significa que:
   a. cada posição de memória é identificada por um endereço com a dimensão de uma *word*.
   b. o acesso apenas pode ser efetuado por instruções que transferem 1 *byte* de informação.
   c. o barramento de endereços e de dados têm obrigatoriamente que ter a mesma dimensão.
   d. cada endereço de memória identifica um registo com a dimensão de uma *word*.

3. Num sistema computacional, o espaço de endereçamento de memória é definido como:
   a. um número único que identifica cada posição de memória.
   b. a dimensão em bits de cada posição de memória.
   c. a gama completa de endereços de memória que o CPU pode gerar.
   d. a quantidade de memória fisicamente disponível no sistema, expressa em MBytes.

4. Num módulo de I/O, o conjunto de registos específicos (*Data*, *Status* e *Control*) e a descrição de cada um deles, para o periférico associado a esse módulo, constitui o que se designa por:
   a. modelo de programação do periférico.
   b. espaço de endereçamento do periférico.
   c. memória específica do periférico.
   d. modelo de comunicação com o periférico.

5. Se o sistema de interrupções é organizado com múltiplas linhas de interrupção (uma por periférico):
   a. a prioridade de atendimento às interrupções é determinada de forma fixa.
   b. a prioridade de atendimento às interrupções é determinada por software.
   c. a prioridade de atendimento às interrupções é determinada em *daisy chain*.
   d. a prioridade de atendimento às interrupções é determinada por um sistema de arbitragem externo.

6. O método de transferência de informação entre um CPU e um módulo de E/S, em que o programa executado no CPU inicia, monitoriza e controla a transferência de informação, designa-se por:
   a. entrada/saída programada (método de pooling).
   b. entrada/saída por *pooling* com vectorização.
   c. entrada/saída por interrupção iniciada pelo CPU.
   d. entrada/saída por interrupção iniciada pelo periférico.



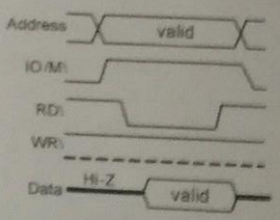
7. Um compilador cruzado (*cross-compiler*) é um programa que corre numa plataforma e:
   a. gera código para uma plataforma com uma arquitetura diferente daquela onde é executado.
   b. simula o funcionamento de uma aplicação numa plataforma diferente.
   c. permite o *debug* de uma aplicação que corre numa plataforma diferente.
   d. gera código que é compatível com múltiplas plataformas, incluindo aquela onde é executado.

8. A função de um *bootloader* num sistema baseado em microcontrolador é:
   a. realizar a compilação do software de alto nível (e.g. C) e iniciar a sua execução após o *reset* do sistema.
   b. transferir o código executável a partir do um sistema *host*, usado no desenvolvimento, para a memória do microcontrolador, permitindo a sua posterior execução.
   c. executar o software e auxiliar no seu *debug* através da introdução de *breakpoints*, visualização do conteúdo de registos e de posições de memória.
   d. interagir com o *cross-compiler* para efeitos de *debug* da aplicação.

9. Quando é usada a técnica de entrada/saída de dados por software (programada):
   a. o periférico faz um pedido de interrupção ao CPU após a conclusão da transferência de dados.
   b. o periférico faz um pedido de interrupção ao CPU quando estiver pronto para transferir os dados.
   c. o CPU interrompe a execução do programa para configurar o controlador de DMA que fará a transferência de dados propriamente dita.
   d. o CPU verifica, através de um ciclo de *polling*, se o periférico está pronto para transferir os dados.

10. A descodificação de endereços consiste em:
   a. representar um endereço em binário, ignorando os zeros à esquerda, por forma a utilizar o menor número de linhas do barramento possível.
   b. determinar, em função do endereço gerado por um periférico, qual o CPU ou memória que deve ser selecionada.
   c. determinar, em função do endereço presente no barramento, qual o periférico/memória que vai ser selecionada.
   d. ocupar a totalidade do espaço de endereçamento do processador com memórias e periféricos.

11. O diagrama temporal da figura do lado representa um ciclo de:



   a. escrita num dispositivo mapeado no espaço de endereçamento de I/O.
   b. escrita num dispositivo mapeado no espaço de endereçamento de memória.
   c. leitura de um dispositivo mapeado no espaço de endereçamento de memória.
   d. leitura de um dispositivo mapeado no espaço de endereçamento de I/O.

12. Os registos PORT, LAT e TRIS do PIC32 permitem, respetivamente:
   a. o output e input de informação e o controlo da direccionalidade de um porto digital.
   b. o input e output de informação e o controlo da direccionalidade de um porto digital.
   c. o controlo da direccionalidade, o input e output de informação de um porto digital.
   d. o controlo da direccionalidade, o output e input de informação de um porto digital.

13. Numa RSI, o conjunto de instruções designado por "epílogo" destina-se, no essencial, a:
   a. alterar a tabela de vetores de modo a impedir que novos pedidos de interrupção sejam atendidos.
   b. identificar a fonte de interrupção (nos casos em que tal é feito por hardware) e obter o endereço inicial da RSI.
   c. regressar ao programa interrompido reativando as interrupções.
   d. repor, a partir da *stack*, o contexto do programa que foi interrompido pela interrupção.

14. Num sistema de interrupções vetorizadas, a sequência de operações efetuada pelo CPU na fase de atendimento a uma interrupção é, pela ordem indicada, a seguinte:
   a. salto para a RSI, identificação da fonte, determinação do endereço da RSI, salvaguarda do endereço de retorno.
   b. determinação do endereço da RSI, identificação da fonte, salvaguarda do endereço de retorno, salto para a RSI.
   c. salvaguarda do endereço de retorno, identificação da fonte, determinação do endereço da RSI, salto para a RSI.
   d. salvaguarda do endereço de retorno, salto para a RSI, identificação da fonte.

Cotações: *Grupo I*: cada 0.6 valores; *Grupo II*: cada 0.8 valores.

15. Um microcontrolador PIC32 usa internamente:
    a. uma arquitetura Harvard, com memórias de código e dados independentes e de diferentes tecnologias.
    b. uma arquitetura *Von Neumann*, mas graças a um sistema de comutação matricial comporta-se, para o programador, como uma arquitetura *Harvard*.
    c. uma arquitetura *Harvard*, mas graças a um sistema de comutação matricial permite a execução de instruções armazenadas em qualquer uma das memórias internas.
    d. um espaço de endereçamento que pode ser gerido dinamicamente, por software, para armazenamento de dados e de instruções.

16. Numa transferência por DMA, o respetivo controlador gera uma interrupção:
    a. quando se encontra pronto para fazer uma nova transferência.
    b. quando termina a transferência de informação para o qual foi previamente configurado.
    c. quando se torna *Bus Master*.
    d. quando recebe um "*DMA Acknowledge*" proveniente do CPU.

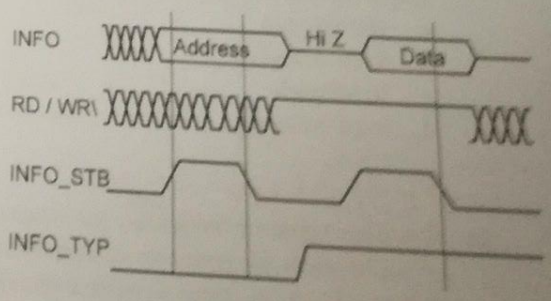
17. Considere um *timer* em que a relação entre as frequências de entrada e de saída é uma constante `k` configurável. Se colocar dois desses timers em cascata (i.e., ligados em série) com constantes de divisão `k1` e `k2`, a relação entre a frequência à entrada do primeiro (fin) e a frequência à saída do segundo (fout) pode ser escrita como:
    a. fout = fin / (k1 * k2)
    b. fout = fin * (k1 + k2)
    c. fout = fin * ((k1 + 1) * (k2 + 1))
    d. fout = fin / ((k1 + 1) * (k2 + 1))

    $K = \frac{fin}{fout}$ $fout = \frac{fin}{K}$

18. Em barramentos paralelo *multi-master*, para garantir que o barramento é sempre atribuído à unidade de maior prioridade, o árbitro poderá efetuar o escalonamento com base:
    a. em prioridades fixas.
    b. no critério *Last-Come/First-Served*.
    c. no critério *First-Come/First-Served*.
    d. em *round-robin*.

19. Numa transferência de tipo semi-síncrono:
    a. o barramento de endereços é assíncrono e o barramento de dados é síncrono.
    b. assume-se que o dispositivo externo responde à velocidade do CPU e, consequentemente, não existem sinais de protocolo envolvidos na transação.
    c. o CPU prolonga o ciclo de leitura/escrita até ao preciso instante em que o dispositivo externo sinaliza que a operação pretendida foi completada.
    d. o CPU prolonga o ciclo de leitura/escrita por um ou mais ciclos de relógio, em função de um sinal de protocolo gerado pelo dispositivo externo.

20. A figura ao lado representa um ciclo de:
    a. escrita assíncrono com barramentos de dados e de endereços multiplexados.
    b. leitura síncrono com barramentos de dados e de endereços não multiplexados.
    c. leitura síncrono com barramentos de dados e de endereços multiplexados.
    d. leitura síncrono com barramentos de dados e de endereços não multiplexados e sinais de controlo independentes de leitura/escrita.

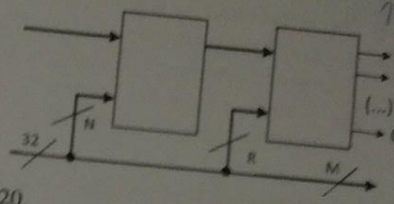


Grupo II

21. Num espaço de endereçamento de 16 bits, um descodificador implementado através da expressão lógica "**CE\ = A15 + A13\ + A12**", descodifica a(s) seguinte(s) gama(s) de endereço(s):

a) **0x3000** a **0x7FFF**.
b. **0xC000** a **0xEFFF**.
c. **0x1000** a **0x1FFF**, **0x3000** a **0x3FFF**.
d. **0x3000** a **0x3FFF**, **0x7000** a **0x7FFF**.

22. Suponha que pretende implementar um circuito gerador de sinais de seleção programável (semelhante ao que estudou nas aulas teóricas) que gere 64 linhas de seleção, cada uma delas ativa em 1k endereços consecutivos, num espaço de endereçamento de 32 bits. Ao segundo bloco da figura devem ser ligados **R** bits, correspondendo à gama:



a. A9 a A0.
b. A15 a A10.
c. A31 a A20.
d. A31 a A16.

23. Para a transferência de 1024 *words* (de 32 bits) num barramento a funcionar em modo bloco, [illegible]

Teste 1

Arquitetura de Computadores II

Grupo II

21. Num espaço de endereçamento de 16 bits, um descodificador implementado através da expressão lógica "**CE\ = A15 + A13\ + A12**", descodifica a(s) seguinte(s) gama(s) de endereço(s):

a) **0x3000** a **0x7FFF**.
b. **0xC000** a **0xEFFF**.
c. **0x1000** a **0x1FFF**, **0x3000** a **0x3FFF**.
d. **0x3000** a **0x3FFF**, **0x7000** a **0x7FFF**.

22. Suponha que pretende implementar um circuito gerador de sinais de seleção programável (semelhante ao que estudou nas aulas teóricas) que gere 64 linhas de seleção, cada uma delas ativa em 1k endereços consecutivos, num espaço de endereçamento de 32 bits. Ao segundo bloco da figura devem ser ligados **R** bits, correspondendo à gama:

a. A9 a A0.
b. A15 a A10.
c. A31 a A20.
d. A31 a A16.

23. Para a transferência de 1024 *words* (de 32 bits) num barramento de 16 bits, um controlador de DMA não dedicado, a funcionar em modo bloco, necessita de:
   a. 2048 *bus cycles*.
   b. 4096 *bus cycles*.
   c. 512 *bus cycles*.
   d. 1024 *bus cycles*.

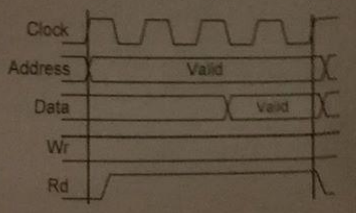
24. Um programa que transfere dados de 32 bits de um periférico para a memória é implementado num ciclo com 10 instruções. Admitindo que o CPU funciona a 200 MHz e que o programa em causa apresenta um CPI de 2.5, a taxa de transferência máxima que se consegue obter, em Bytes/s, supondo um barramento de dados de 32 bits, é:
   a. 16 MByte/s.
   b. 32 MByte/s.
   c. 64 MByte/s.
   d. 128 MByte/s.

25. Considere um dispositivo de DMA não dedicado de 32 bits (i.e. com barramento de dados de 32 bits), a funcionar a 100 MHz. Suponha ainda que são necessários 2 ciclos de relógio (= $1T_{BC}$) para efetuar uma operação de leitura ou escrita. A taxa de transferência de pico desse DMA (expressa em Bytes/s), em modo "*cycle-stealing*" e com um tempo mínimo entre operações elementares de 1 T_{BC} será de:
   a. 100 MByte/s.
   b. 66.(6) MByte/s.
   c. 50 MByte/s.
   d. 16.(6) MByte/s.

. Considere agora um barramento (de informação) paralelo multiplexado, de 16 bits. Sobre esse barramento pretende-se implementar um protocolo de comunicação, de tipo microciclo, que apresenta um espaço de endereçamento de 32 bits e 32 bits de dados. O número mínimo de ciclos de barramento necessários para completar uma transação é:
   a. 3.
   b. 2.
   c. 5.
   d. 4.

Cotações: *Grupo I*: cada 0.6 valores; *Grupo II*: cada 0.8 valores.

27. Um dado CPU suporta transferências de tipo síncrono e de tipo semi-síncrono. O CPU funciona a uma frequência de 500 MHz e o ciclo de leitura pode ser descrito no diagrama temporal ao lado. Pretende-se ligar a este CPU uma memória com um tempo de acesso de 12 ns (tempo que decorre desde que a memória é selecionada até que a informação fica disponível no *data bus*). Se o descodificador de endereços da memória introduzir um atraso de propagação de 2.5 ns, o número mínimo de *wait-states* que assegura a correta leitura é (**Nota:** assuma que o tempo mínimo durante o qual os dados têm e estar válidos no *data bus* tem de ser superior a um ciclo de relógio):

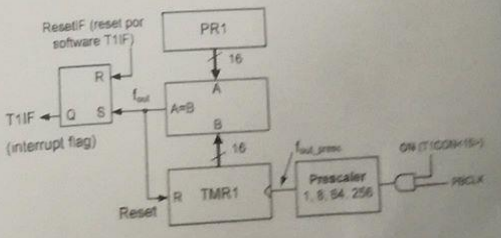


a. 4.
b. 5.
c. 6.
d. 7.

28. Considere um sistema baseado num CPU a funcionar a uma frequência de 40 MHz com uma taxa de execução de 20 MIPS ($20x10^6$ instruções por segundo, i.e. CPI = 2) que processa, por interrupção, eventos externos periódicos. Se o ***overhead* total** do atendimento à interrupção for de 40 **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. 250 kHz.
b. 333.(3) kHz.
c. 500 kHz.
d. 1 MHz.

29. Um *watchdog timer*, com uma frequência de entrada de 20 kHz, é construído a partir de um contador crescente de 10 bits, e gera um sinal de *reset* ao processador sempre que a contagem atinge o valor máximo. Para impedir o *reset* do processador, o intervalo de tempo máximo entre *resets* do contador deve ser, aproximadamente:
a. 1024 ms.
b. 512 ms.
c. 51 ms.
d. 5 ms.

30. Considere um timer do tipo A do PIC32 (semelhante ao da figura) e um PBCLK = 20MHz. Para que o período de f_{out} seja de 15ms com a melhor exatidão possível:
a. O valor do *prescaler* deverá ser de 1 e o de PR1 299999.
b. O valor do *prescaler* deverá ser de 8 e o de PR1 37499.
c. O valor do *prescaler* deverá ser de 64 e o de PR1 4686.
d. O valor do *prescaler* deverá ser de 256 e o de PR1 1170.



CHAVE

c
d
c
a
a
a
a
b
d
c
d
b
d
c
a
b
a
a
d
c

d
b
b
b
c
d
b
c
c
b