1º Teste Teórico de AC1, 2022/2023 + some variants

1) **Um microcontrolador é um sistema computacional programável que:**
a) inclui, num único circuito integrado CPU, memória e um conjunto variável de periféricos e portos de I/O.
b) disponibiliza, através dos seus portos de I/O, a generalidade dos sinais dos barramentos do microprocessador para ligação direta a sensores e atuadores de um sistema embebido.
c) devido a restrições de custos, suporta um número reduzido de instruções e de registos.
d) por questões de dimensão, não utiliza mecanismos de multiplexagem para partilha de pinos físicos do circuito entre diversas funcionalidades internas.

2) Num sistema computacional, o espaço de endereçamento de memória é definido como:

a) um número único que identifica cada posição de memória.

b) a quantidade de memória fisicamente disponível no sistema, expressa em MBytes.

c) a gama completa de endereços de memória que o CPU pode gerar.

d) a dimensão em bits de cada posição de memória.

3) A função de um bootloader num sistema baseado em microcontrolador é:

a) transferir o código executável, a partir do sistema host, usado no desenvolvimento, para a memória do microcontrolador, permitindo a sua posterior execução.

b) realizar a compilação do software de alto nível (e.g C) e iniciar a sua execução após o reset do sistema.

c) interagir com o cross-compiler para efeitos de debug da aplicação.

d) executar o programa e auxiliar no seu debug permitindo a introdução de breakpoints, visualização do conteúdo de registos e de posições de memória.

4.1) Dizer-se que num sistema computacional a memória apresenta uma organização do tipo Byte-addressable significa que:

a) Cada posição de memória é identificada com um endereço de 1 byte.

b) O acesso apenas pode ser efetuado por instruções que transferem 1 byte de informação.

c) Uma word de 32 bits é armazenada em 4 posições de memória consecutivas de 1 byte.

d) o barramento de endereços e de dados têm obrigatoriamente que ter a mesma dimensão.

4.2) 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) cada endereço de memória identifica um registo com a dimensão de uma word.

d) o barramento de endereços e de dados têm obrigatoriamente que ter a mesma dimensão.

5.1) Na arquitetura de um sistema computacional, o Address Bus permite:

a) especificar o tipo de operação efetuada sobre a memória ou sobre o periférico. (Control Bus)

b) identificar, na memória ou num periférico, a origem/destino da informação a transferir.

c) transferir dados entre os dispositivos periféricos e os registos internos do CPU. (Data Bus)

d) transferir o código máquina das instruções para o program counter.

5.2) Na arquitetura de um sistema computacional, o Data Bus permite:

a) especificar o tipo de operação efetuada sobre a memória ou sobre o periférico. (Control Bus)

b) identificar, na memória ou num periférico, a origem/destino da informação a transferir. (Address Bus)

c) transferir dados entre os dispositivos periféricos e os registos internos do CPU. (Data Bus)

d) transferir o código máquina das instruções para o program counter.

6) O bus matrix usado no pic32, permite:

a) o acesso do CPU a uma memória RAM para transferência simultânea de dados e instruções.

b) o acesso do CPU a uma memória FLASH para transferência simultânea de dados e instruções.

c) a transferência direta de dados ou instruções de memória RAM para a FLASH (ou o contrário) sem intervenção de qualquer outro dispositivo.

d) o acesso do CPU a uma memória FLASH para leitura de dados constantes e à mesma memória FLASH para leitura de instruções.

7) Um compilador cruzado (cross compiler) é um programa que corre numa plataforma e:

a) simula o funcionamento de uma aplicação numa plataforma diferente.

b) permite o debug de uma aplicação que corre numa plataforma diferente.

c) gera código que pode ser executado na mesma plataforma em que é gerado.

d) gera código para uma plataforma com uma arquitetura diferente daquela onde é executado.

8) Quando nos referimos a um "módulo de I/O" estamos a referir-nos:

a) à parte de um dispositivo periférico que funciona como adaptador entre as características intrínsecas do periférico e as características do CPU e do sistema de memória.

b) a um periférico que permite operações de escrita e leitura.

c) ao software (device-driver) que assegura que o acesso ao periférico é transparente para as aplicações.

d) ao tipo de conector que permite a interface entre um periférico e o canal de comunicação do mesmo com o mundo exterior.

9) O modelo de programação do periférico especifica:

a) o subconjunto de instruções assembly do CPU suportadas por esse periférico.

b) a funcionalidade do periférico e o seu conjunto de registos de dados, de controlo e de estado.

c) quais os sinais elétricos usados na ligação do periférico a dispositivos externos, tais como sensores atuadores.

d) as arquiteturas e as ferramentas de desenvolvimento com as quais o periférico pode ser usado.

10) Na implementação da parte de dados de um porto de saída devem ser usados:

a) buffers tri-state para que a informação presente no barramento de dados só fique disponível para (escrita ou leitura) no periférico quando o porto for ativo.

b) flip-flops para armazenar o valor transferido através do barramento de dados durante um ciclo de escrita.

c) buffers tri-state para que a informação só seja colocada no barramento de dados quando o porto for selecionado.

d) flip-flops para armazenar o valor presente no barramento de endereços, se este coincidir com o endereço do porto.

11) A descodificação de endereços consiste em:

a) representar um endereço em binário de forma a utilizar o menor número de linhas de barramento.

b) determinar em função do endereço gerado pelo periférico, qual o CPU ou memória que deve ser selecionada.

c) ocupar a totalidade do espaço de endereçamento do processador com memórias e periféricos.

d) determinar em função do endereço presente no barramento, qual o periférico ou memória que vai ser selecionada.

12.1) Num determinado sistema com uma organização de memória do tipo Byte-Addressable e um espaço de endereçamento de 16 bits, foi implementado um descodificador de endereços usando a seguinte expressão lógica (lógica negativa): CE\ = A15 + A14 + A12\ (A13 ignorado). Com este descodificador pode ser selecionada uma memória:

a) de 4k bytes, na gama de endereços 0xD000 a 0xDFFF

b) de 8k bytes, na gama de endereços 0xC000 a 0xDFFF

c) de 2k bytes, na gama de endereços 0xF000 a 0XF7FF

d) de 4k bytes, na gama de endereços 0x3000 a 0x3FFF

12.2) 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.

13) Suponha que os bits 7 e 6 do porto B do PIC32 estão configurados como saída e que se pretende atribuir a esses dois pontos o valor 1 e 0, respetivamente, sem alterar o valor dos restantes. Para isso, em linguagem C, pode fazer-se:

a) LATB = (LATB & 0XFF3F) | (1 << 7);

b) LATB = (LATB & 0XFFBF) | (0 << 6);

c) LATB = (LATB | 0X0080) & (0 << 6);

d) LATB = (LATB & 0X0000) | (1 << 7);

14) Na implementação de um porto de I/O do pic32, o registo com a designação "PORT" está associado a um conjunto de dois flip-flops D ligados em série (shift register de dois andares). O objetivo desta organização é:

a) criar um atraso temporal de dois ciclos de relógio relativamente às alterações do registo TRIS.

b) assegurar que o registo LAT é adequadamente escrito numa operação "Read Modify Write".

c) assegurar que há uma adequada sincronização entre o valor lógico presente na entrada e o relógio interno e prevenir a ocorrência, internamente, de fenómenos de meta-estabilidade.

d) garantir que há tempo para colocar em alta impedância o buffer tri-state de saída quando se comuta o porto de modo de saída para modo de entrada.

15) O método de transferência de informação entre um CPU e um periférico, em que o programa executado no CPU, inicia, monitoriza e controla a transferência de informação, designa-se por:

a) entrada/saída por polling com vectorização.

b) entrada/saída por interrupção iniciada pelo CPU.

c) entrada/saída por interrupção iniciada pelo periférico.

d) entrada/saída programada (método de polling).

16) Num sistema que implemente "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.

17.1) Numa RSI, o conjunto de instruções designado por "prólogo" destina-se, no essencial, a:

a) alterar a tabela de vetores de modo a impedir que novos pedidos de interrupção sejam atendidos.

b) salvaguardar, na stack, o contexto atual do programa interrompido, i.e, registos internos do CPU.

c) identificar a fonte de interrupção (nos casos em que tal é feito por hardware) e obter o endereço inicial da RSI.

d) regressar ao programa interrompido.

17.2) 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) repor, a partir da stack, o contexto do programa que foi interrompido pela interrupção.

c) identificar a fonte de interrupção (nos casos em que tal é feito por hardware) e obter o endereço inicial da RSI.

d) regressar ao programa interrompido reativando as interrupções.

18) Na organização do sistema de interrupção designada por "identificação da fonte por software", o processador identifica o periférico gerador da interrupção:

a) 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.

b) antes de saltar para a rotina de serviço à interrupção lendo o registo de estado de cada um dos periféricos do sistema.

c) na rotina de serviço à interrupção lendo o registo de estado de cada um dos periféricos do sistema.

d) num ciclo de interrupt acknowlege durante o qual o periférico gerador da interrupção coloca o vetor no barramento de dados.

19) Quando nos referimos a uma "secção critica", num trecho de código executado por um CPU, estamos a referir-nos:

a) a uma sequência de instruções cuja execução não pode ser interrompida por uma interrupção, por usar/atualizar um recurso partilhado com o código da rotina de serviço à interrupção.

b) a secção mais importante do código em execução e que não pode deixar de ser executado.

c) a um conjunto consecutivo de instruções onde não é possível chamar uma função.

d) a uma secção de código que, em certas circunstâncias, pode gerar uma exceção.

20) Quando é usada a técnica de transferência de dados por DMA para transferir informação entre a memória e um periférico.

a) o CPU configura o controlador de DMA com os endereços de origem e destino e o número de words a transferir, e o DMA faz depois a transferência de dados.

b) o periférico faz um pedido de interrupção ao controlador de DMA após a conclusão da transferência de dados.

c) o CPU verifica, através de um ciclo de polling ao registo de estado do controlador de DMA, se a transferência já foi concluída.

d) o DMA verifica, através de um ciclo de polling ao registo de estado do periférico, se existem mais dados para serem transferidos.

21) Numa transferência por DMA, o mecanismo de interrupção é utilizado pelo respetivo controlador para:

a) efetuar ao CPU o pedido de cedência dos barramentos, a transferência tem início quando o DMA receber a confirmação, através do sinal busgrant, de que os barramentos foram libertados.

b) informar o CPU que a transferência de informação vai ter início, permitindo desse modo que o CPU suspenda a atividade de acesso ao exterior.

c) informar o CPU que a transferência de informação foi completa.

d) informar o CPU da existência de uma anomalia ocorrida durante o processo de transferência.

22) 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, e inicia de seguida a transferência.

b) ativa o sinal busreq, efetuando a transferência logo que se torne o bus master.

c) ativa uma interrupção que é interpretada pelo CPU como um pedido de cedência de barramentos e a transferência é efetuada quando o DMA reconhecer a ativação do sinal busgrant.

d) chama 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.

23) Num sistema de comunicação série que use transmissão síncrona:

a) o sinal de relógio não é transmitido, nem há recuperação do relógio no recetor.

b) o sinal do relógio é codificado nos dados, ou é transmitido de forma explicita através de um sinal adicional.

c) os relógios do transmissor e do recetor não precisam de estar sincronizados.

d) existe obrigatoriamente, para além da linha de dados, uma linha através da qual é transmitido o sinal de relógio.

24.1) Considere um timer em que a relação entre as frequências de entrada e de saída é uma constante "K" configurável. Considere ainda que se usaram dois desses timers e se ligaram em cascata (i.e, em série). Supondo que a frequência à entrada do primeiro timer é 1 MHz, para obter à saída do segundo timer uma frequência de 200Hz, as constantes de configuração dos dois timers "k1" e "k2", poderão ter os seguintes valores:

a) K1 = 100, k2 = 25

b) K1 = 10, K2 = 20

c) K1 = 200, K2 = 100

d) K1 = 200, K2 = 25

24.2) Considere um timer em que a relação entre as frequências de entrada e de saída é uma constante "K" configurável. Considere ainda que se usaram dois desses timers e se ligaram em cascata (i.e, em série) com as constantes de divisão k1 e k2, a relação entre a frequência à entrada do primeiro (fin) e a frequência à saída do segunda (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))

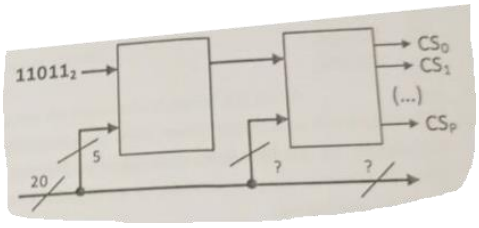
25.1) Considere um espaço de endereçamento de 20 bits e o circuito gerador de sinais de seleção programável da figura (igual ao que estudou nas aulas teóricas). Na situação apresentada e considerando que a linha de seleção CS, está ativa na gama 0XD8200 a 0xD83FF, podemos concluir que o circuito da figura gera:

a) 16 linhas de seleção.

b) 64 linhas de seleção.

c) 8 linhas de seleção.

d) 32 linhas de seleção.



25.2) Suponha que pretende implementar um circuito gerador de sinais de seleção programável (semelhante ao que estudou nas aulas teóricas) que gera 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:

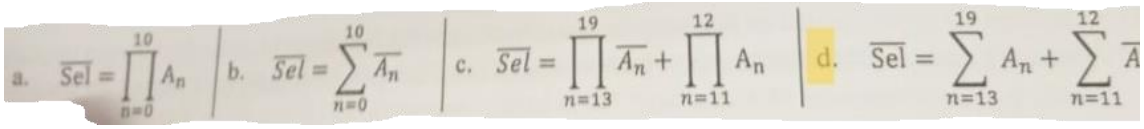
a) A9 a A0

b) A15 a A10

c) A31 a A20

d) A31 a A16

26) O sinal de seleção "Sel\" (Lógica negativa) de uma memória de 2k endereços mapeada na gama de endereços 0x01800...0x01FFF, num espaço de endereçamento de 20 bits, pode ser obtido através da expressão lógica:



27) Considere um controlador de DMA não dedicado de 32 bits (i.e com barramento de dados de 32 bits) a funcionar a 180 MHZ. Suponha ainda que são necessários 2 ciclos de relógio (1 Tbc) 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 2 Tbc é:

a) 60 MByte/s

b) 120 MByte/s

c) 12,5 MByte/s

d) 40 MByte/s

RIP resto dos exercícios, o connect não os tinha 😟

1-Num módulo de I/O, o conjunto de registos específicos (Data, Status, 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 especifica do periférico.
d) modelo de comunicação com o periférico.

2-Quando 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.

3- 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 propriamente dita.

d) o CPU verifica, através de um ciclo de polling, se o periférico está pronto para transferir dados.

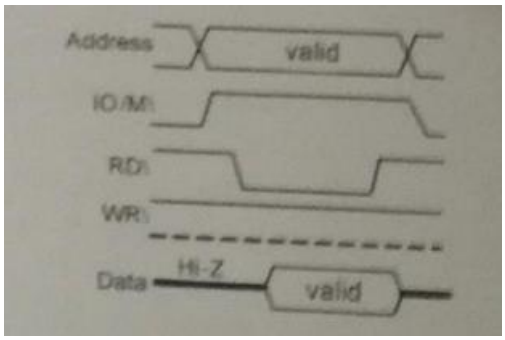
4- O diagrama temporal da figura do lado representa um ciclo de:

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 ao espaço de endereçamento de memória.

d) leitura de um dispositivo mapeado no espaço de endereçamento de I/O



5- 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 de direccionalidade, o input e output de informação de um porto digital.

d) o controlo de direccionalidade, o output e input de informação de um porto digital.

6- Um microcontrolador PIC32 usa internamente:

a) uma arquitetura Harvard, com memórias de código e dados independentes e de diferentes tecnologias.

b) uma arquitetura de Von Neumann, mas graças a um sistema de comutação matricial comporta-se para o programador, como uma arquitetura Harvard.

c) uma arquitetura de Harvard, mas graças a um sistema de comutação matricial permite a execução de instruções armazenadas e qualquer uma das memórias internas.
d) um espaço de endereçamento que pode ser gerido dinamicamente, por software, para armazenamento de dados e de instruções.

7- 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.

8- 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

9- 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 eu 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

10-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 (= 1Tbc) para efetuar uma operação de leitura ou escrita. A taxa de transferência de pico desse DMA (expressa em Byte/s), em modo "cycle-stealing" e com um tempo mínimo entre operações elementares de 1 Tbc será de:
a) 100 MByte/s
b) 66.(6) MByte/s
c) 50 MByte/s
d) 16.(6) MByte/s

11- 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

12- 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

Resolução: Começamos por calcular o período do sinal: T = 1/f (f é a frequência). Logo temos -> T = 1/20*10^3 (20 kHz = 20*10^3 Hz) -> T = 50 microsegundos. Como contador é de 10 bits, pode contar até 2^10 = 1024 valores diferentes antes de reiniciar. Logo tempo total máximo é o produto do período de cada contagem vs o número máximo possível de contagens = 50 microsegundos * 1024 = 51,2 ms.

13- Considere um timer do tipo A do PIC32 (semelhante ao da figura) e um PBCLK = 20MHz. Para que o período de fout seja de 15 ms 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.