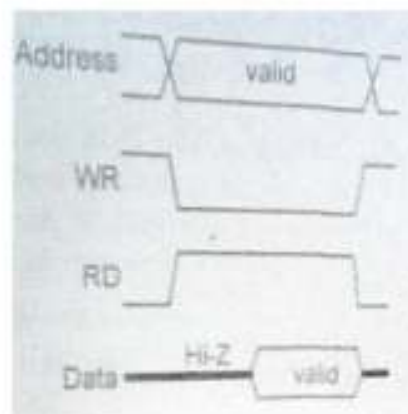


Primeiro Teste AC2 2010-2011

- 1- A função de um *bootloader* num sistema baseado ????????????
 - a) Transferir o código executável do sistema ???????????? desenvolvimento para posterior *disassembly*.
 - b) Executar o software e auxiliar no seu *debug* através da introdução de *breakpoints*, visualização do conteúdo de registos e de posições de memória.
 - c) Transferir o código executável do *host* usado no desenvolvimento para o sistema *embedded* para posterior execução.
 - d) Realizar a compilação do software e iniciar a sua execução após o reset do sistema.
- 2- Um microcontrolador é tipicamente usado em:
 - a) Sistemas *embedded* por disponibilizar sempre um coprocessador de vírgula flutuante.
 - b) Sistemas *embedded* por disponibilizar um vasto conjunto de periféricos para interface com sensores e actuadores.
 - c) Sistemas computacionais de uso geral por disponibilizar sempre um coprocessador de vírgula flutuante.
 - d) Sistemas computacionais de uso geral por disponibilizar um vasto conjunto de periféricos para interface com sensores e actuadores.
- 3- A *bus matrix* integrada em diversos microcontroladores, entre os quais o PIC32, permite:
 - a) O acesso simultâneo do CPU a uma memória RAM para transferência de dados e a uma memória FLASH para leitura de instruções.
 - b) A transferência directa de dados ou instruções da memória RAM para a FLASH (ou vice-versa) sem intervenção de qualquer outro dispositivo.
 - c) O acesso do CPU a uma memória RAM para transferência simultânea de dados e instruções.
 - d) O acesso do CPU a uma memória FLASH para transferência simultânea de dados e instruções.
- 4- O diagrama temporal da figura do lado representa um ciclo de:
 - a) Escrita num dispositivo em que os sinais de controlo usam lógica positiva
 - b) Escrita num dispositivo em que os sinais de controlo usam lógica negativa
 - c) Leitura de um dispositivo em que os sinais de controlo usam lógica positiva.
 - d) Leitura de um dispositivo em que os sinais de controlo usam lógica negativa



- 5- Escrever um '0' num bit de um registo TRIS do PIC32:
- a) Configura o respectivo porto como um porto de entrada.
 - b) Configura o respectivo porto como um porto de saída.
 - c) Coloca a '0' o respectivo porto.
 - d) Desactiva o respectivo porto.
- 6- Numa transferência semi-síncrona:
- a) O CPU prolonga o ciclo de leitura/escrita por um ou mais ciclos de relógio, se for activado um sinal de protocolo gerado pelo dispositivo externo.
 - b) Assume-se que o dispositivo externo responde à velocidade do CPU e, consequentemente, não existem sinais de protocolo envolvidos na transacção.
 - c) O CPU prolonga o ciclo de leitura/escrita até que o dispositivo externo sinalize que a operação pretendida foi completada.
 - d) Nenhuma das anteriores.
- 7- Quando é usada a técnica de entrada/saída de dados por DMA:
- a) O periférico faz um pedido de interrupção ao controlador de DMA após a conclusão da transferência de dados.
 - b) O CPU configura o controlador de DMA que fará a transferência propriamente dita.
 - c) O CPU verifica através de um ciclo de *polling* ao registo de dados do controlador de DMA se a transferência já foi concluída.
 - d) O periférico faz um pedido de interrupção ao CPU quando estiver pronto para transferir os dados.
- 8- Para a transferência de 2048 words (de 32 bits) um controlador de DMA de 32 bits, não dedicado, a funcionar em modo bloco, necessita de:
- a) 8192 bus cycles
 - b) 4096 bus cycles
 - c) 2048 bus cycles
 - d) 1024 bus cycles
- 9- Na interface RS232 são usadas:
- a) Tensões entre -3v e +3v para representar os níveis lógicos dos bits transmitidos.
 - b) Técnicas de sincronização com relógio explícito controlado pelo transmissor.
 - c) Duas linhas, uma para a massa e outra para comunicação bidireccional *half duplex*.
 - d) Três linhas, uma para a massa e outras duas para comunicação bidireccional *full duplex*.

10- No PIC32, os pedidos de interrupção são aceites (servidos):

- a) Primeiro os de nível de prioridade superior.
- b) Primeiro os de nível de prioridade inferior.
- c) Pela ordem em que os pedidos foram feitos (first come, first served).
- d) Pela ordem inversa em que os pedidos foram feitos (first come, last served).

Grupo II

11- Considere um *timer* de 16 bits, com reset síncrono, 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 70% do sinal é determinado pela constante KA, para se obter a saída um sinal com um período de 1ms e duty cycle de 25%, as constantes KA e KB deverão valer, respectivamente:

- a) 249 e 749
- b) 750 e 750
- c) 749 e 249.
- d) 750 e 250

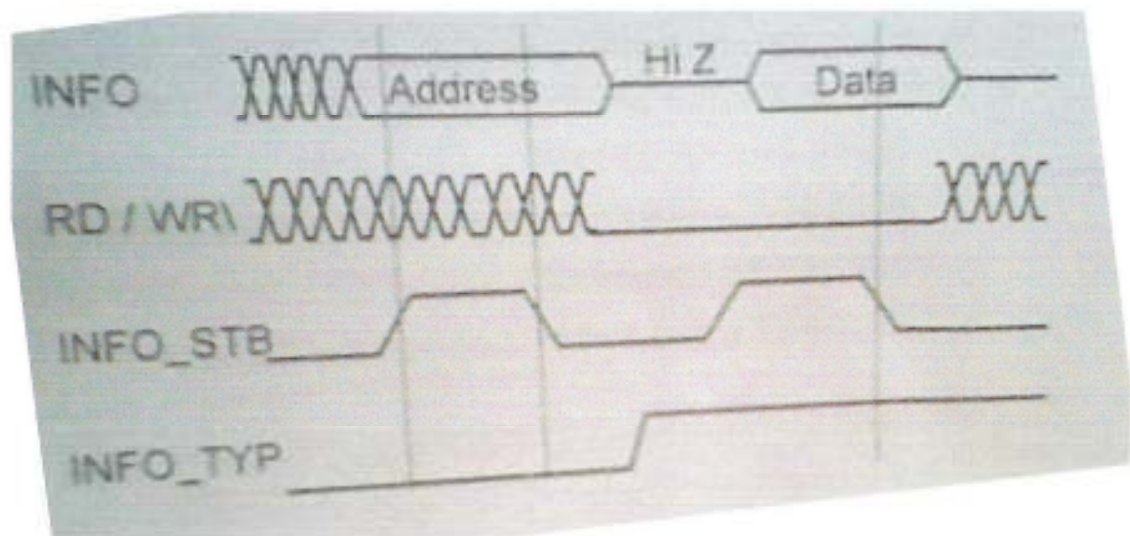
12 Considere um *watchdog timer*, com frequência de entrada de 1Mhz, construído a partir de um contador 0000000000000000 de 16 bits que, sempre que a contagem atinja o valor 4095 , força o reset do processador que está a realizar. É suposto o programa a correr nesse processador reiniciar o *watchdog timer* em intervalos de tempo compreendidos entre 100ms e 150ms. Qual o valor mínimo a carregar no contador de modo a impedir o reset do processador:

- a) 50 000
- b) 100 000
- c) 150 000
- d) A frequência de entrada do *timer* é demasiado elevada para os intervalos de tempo pretendidos.

13- Num espaço de endereçamento de 20 bits, um decodificador implementado através da expressão lógica " $Se\backslash = A19 + A17 + A16\backslash$ ", decodifica a(s) seguinte(s) gama(s) de endereço(s):

- a) 0xA0000 a 0xFFFF
- b) 0xA0000 a 0xFFFF, 0xE0000 a 0xEFFFF
- c) 0x10000 a 0x1FFFF
- d) 0x10000 a 0x1FFFF, 0x50000 a 0x5FFFF

- 14- Suponha que pretende implementar um gerador de sinais de selecção (descodificador de endereços) programável que gere 16 linhas de selecção, cada uma delas activa em 1024 endereços consecutivos. Considerando um espaço de endereçamento de 24 bits, o bloco que estabelece a gama total de endereços descodificada deverá utilizar os bits:
- A13 a A0
 - A23 a A10
 - A23 a A14
 - A9 a A0
- 15- Considere um barramento paralelo multiplexado, constituído por 16 linhas informação. Sobre este barramento pretende implementar-se um protocolo de comunicação, do tipo microciclo, que apresenta um espaço de endereçamento de 32 bits e 32 bits de dados. O número mínimo de ciclos necessários para completar uma transacção é:
- 4
 - 3
 - 2
 - Não há dados suficientes para responder a questão.
- 16- Num porto de saída constituído por *flip-flops* tipo *D positive edge triggered*, o sinal de *clock enable* activo baixo é obtido a partir dos sinais Sel e WR (ambos activos altos) de acordo com a expressão lógica:
- $\text{ClkEnable} = \text{Sel} + \text{WR}$
 - $\text{ClkEnable} = (\text{Sel} + \text{WR})'$
 - $\text{ClkEnable} = \text{Sel} \cdot \text{WR}$
 - $\text{ClkEnable} = (\text{Sel} \cdot \text{WR})'$
- 17- ?????? seja possível ter um porto de entrada (ou um registo de estado) e um porto de saída (ou um registo de controlo) mapeados no mesmo endereço do espaço de endereçamento de um processador é necessário:
- Que a lógica de selecção resulte do "ou" exclusivo dos sinais read e write.
 - Que ambos usem diferentes réplicas no caso de ser usada descodificação parcial.
 - Que ambos partilhem a mesma lógica de selecção.
 - Que a lógica de selecção do porto de entrada tome em consideração o sinal de *write* e a do porto de saída o sinal de *read*.
- 18- A figura seguinte corresponde a um ciclo de:
- Leitura síncrono com barramentos de dados e de endereços não multiplexados.
 - Escrita síncrono com barramento de dados e de endereços não multiplexados e sinais de controlo independentes de leitura/escrita.
 - Leitura assíncrono com barramentos de dados e de endereços multiplexados.
 - Escrita síncrono com barramentos de dados e de endereços multiplexados.



19- Num sistema de interrupções com uma única linha e identificação da fonte por software, a sequência de operações efectuadas durante o atendimento a uma interrupção é, pela ordem indicada, a seguinte:

- Identificação da fonte, determinação do endereço da RSI, salvaguarda do endereço de retorno, salto para a RSI.
- Salvaguarda do endereço de retorno, salto para a RSI, identificação da fonte.
- Salto para a RSI, identificação da fonte, salvaguarda do endereço de retorno.
- Determinação do endereço da RSI, identificação da fonte, salvaguarda do endereço de retorno, salto para a RSI.

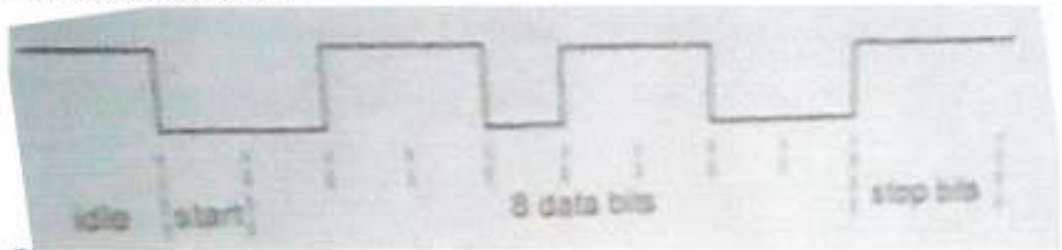
20- Numa transferência por DMA, em modo *cycle stealing*, quando o controlador de DMA pretende dar início a uma transferência elementar:

- Activa o sinal de *busreq*, efectuando a transferência elementar logo que se torne no *bus master*.
- Activa o sinal de *busreq* durante um número fixo de ciclos de relógio, efectuando de seguida a transferência.
- Gera uma interrupção que é interpretada pelo CPU como um pedido de cedência dos barramentos, a transferência é efectuada quando o DMA reconhecer a activação do sinal *busgrant*.
- 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 actividade do DMA.

- 21- Qual dos seguintes modos de funcionamento de um controlador de DMA provoca um menor aumento da latência no atendimento de interrupções:
- a) Cycle-stealing
 - b) Vectorizado.
 - c) Bloco.
 - d) Burst
- 22- Em barramentos *multi-master* existe um árbitro que efectua a gestão dos acessos ao barramento. Para evitar o fenómeno de starvation, o árbitro poderá:
- a) Aumentar a prioridade dos pedidos de acesso ao barramento que já foram atendidos.
 - b) Efectuar o escalonamento com base no critério Last-in/First-Served.
 - c) Vedar o atendimento a novos pedidos enquanto todos os pedidos pendentes não forem satisfeitos.
 - d) Utilizar qualquer uma das técnicas anteriores.
- 23- Suponha que se pretende desenvolver um novo standard de comunicação série de alto desempenho, em que a transmissão é orientada ao bit, não se usem linhas dedicadas à transmissão do sinal de relógio e o tamanho das tramas pode variar entre 256 bytes e 2048 bytes. Nestas circunstâncias a sincronização dos sinais de relógio do transmissor e do receptor deve ser baseada em:
- a) Relógio explícito do transmissor.
 - b) Relógio explícito mutuamente sincronizado.
 - c) Relógio codificado.
 - d) Relógio implícito.
- 24- O standard RS-232 usa um método de sincronização baseado em relógio implícito. Uma das origens do erro de amostragem é denominado "erro de fase". Para reduzir o peso desta fonte de erro deve usar-se:
- a) O maior número possível de stop bits.
 - b) Um factor de sobreamostragem o maior possível.
 - c) Relógio no transmissor e no receptor que sejam o mais rigorosos possível.
 - d) Sempre o bit de paridade.
- 25- A máxima taxa de transmissão de dados líquida (*net bit rate*) em RS232 com uma *baud rate* de 115200 bps, 7 bits de dados, 1 bit de paridade e 2 stop bits é, aproximadamente:
- a) 115200 bps
 - b) 89600 bps
 - c) 80600 bps
 - d) 73300 bps

26- Um dispositivo com interface RS232 e configurado para transmitir com 8 bits de dados, sem paridade e 2 stop bits, produz a trama seguinte (onde é enviado o carácter '6') que é recebida por outro dispositivo RS232 incorrectamente configurado para 7 bits de dados, paridade par e 1 stop bit mas com o mesmo baud rate. Nestas circunstâncias o receptor:

- a) Não vai detectar qualquer erro.
- b) Vai detectar um erro de paridade.
- c) Vai detectar uma trama inválida devido a um número incorrecto de stop bits.
- d) Vai detectar um erro de paridade e uma trama inválida devido a um número incorrecto de stop bits.



0

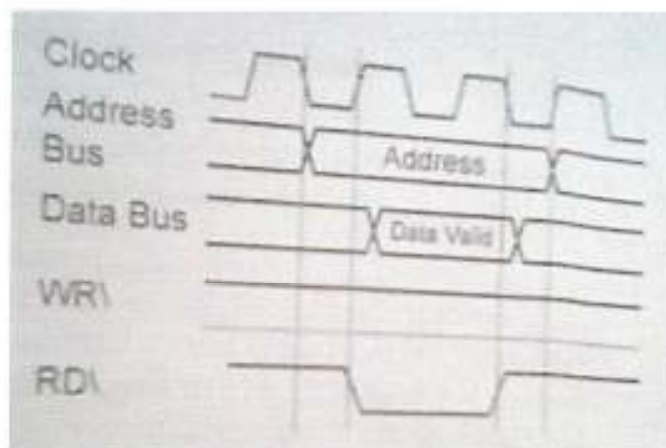
III

27- Considere um sistema baseado num CPU a funcionar a uma frequência de 100 MHz com uma taxa de execução de 50 MIPS que processa por interrupção eventos externos periódicos. Se a latência máxima no atendimento a uma interrupção for de 10 **ciclos de relógio**, e a rotina de serviço a interrupção tiver 20 **instruções**, a máxima frequência a que esses eventos podem ocorrer é aproximadamente:

- a) 4 MHz
- b) 2 MHz
- c) 1 MHz
- d) 500KHz

28- Considere um CPU que suporta transferências de tipo semi-síncrono. O CPU funciona a uma frequência de 100 MHz e o ciclo de leitura correspondente é mostrado na figura do lado. Pretende-se ligar a este CPU um periférico com um tempo de acesso de 24ns (tempo que decorre desde que o periférico é seleccionado até que a informação fica disponível no barramento de dados). O número de *wait-states* que é necessário introduzir para que a operação decorra com sucesso é:

- a) 3
- b) 2
- c) 1
- d) 0



29- O trecho de código *assembly* MIPS (com *branch delay slot*) envia 1000 words para um periférico. Admitindo que este código é executado num processador de 100 MIPS (executa 105 instruções/seg) e que o ciclo de *polling* é efectuado em média 5 vezes, a taxa de transferência média que se obtém é, aproximadamente:

- a) 20 Mbytes/s
- b) 16 Mbytes/s
- c) 5 Mbytes/s
- d) 4 Mbytes/s

```
send:  la    $a0,buffer
       la    $a1,io_addr
       li    $a2,1000
poll:  lw     $t0,0($a1)
       andi  $t0,$t0,0x0001
       beq   $t0,$zero,poll
       nop
       lw    $t0,0($a0)
       sw    $t0,4($a1)
       addi  $a1,$a2,-1
       bne   $a2,$zero,poll
       addiu $a0,$a0,4
       jr    $ra
```