**EXAME TEÓRICO DE RECURSO DE ARQUITECTURA DE COMPUTADORES II**

**13/07/2011**

**------------------------------------------------------------------------------------------------------------------------------------------------------------------------------**

**GRUPO I – 0,6 cada**

**1) A função de um bootloader num sistema baseado num microcontrolador é:**

**a)** realizar a compilação do software e iniciar a sua execução após o reset do sistema

**b)** transferir o código executável para o host usado no desenvolvimento para posterior assemblagem ou compilação

**c)** transferir o código executável do host usado no desenvolvimento para o sistema embedded para posterior execução

**d)** nenhuma das respostas está correcta

**2) O sinal de selecção (CE\) de um porto mapeado na gama de endereços 0x00400...0x007FF de um processador com um espaço de endereçamento de 20 bits pode ser obtido através da expressão:**

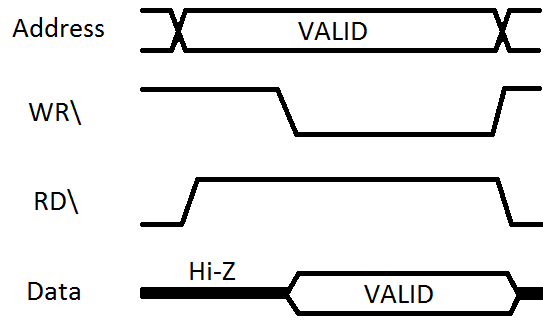
**a)**

**b)**

**c)**

**d)**

**3) O diagrama temporal da figura ao lado representa um ciclo de:**

**a)** leitura de um dispositivo em que os sinais de controlo

usam lógica positiva

**b)** leitura de um dispositivo em que os sinais de controlo

usam lógica negativa

**c)** escrita num dispositivo em que os sinais de controlo

usam lógica positiva

**d)** escrita num dispositivo em que os sinais de controlo

usam lógica negativa

**4) Numa transferência semi-síncrona:**

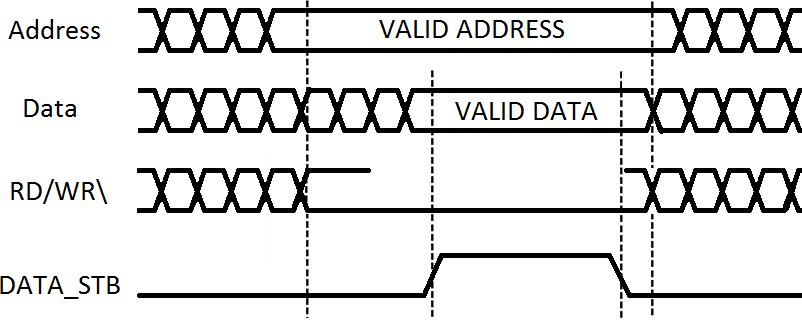
**a)** assume-se que o dispositivo externo responde à velocidade do CPU, e consequentemente, não existem sinais de protocolo envolvidos na transacção

**b)** 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

**c)** O CPU prolonga o ciclo de leitura/escrita até que o dispositivo externo sinalize que a operação pretendida foi completada

**d)** nenhuma das respostas está correcta

**5) A figura ao lado corresponde ao diagrama temporal de uma transferência:**

1. ****assíncrona de escrita, com dados e endereços

disponibilizados numa configuração “*merged*”

1. assíncrona de escrita, com dados e endereços

disponibilizados numa configuração “*micro-ciclo*”

1. síncrona de escrita, com dados e endereços

disponibilizados numa configuração “*micro-ciclo*”

1. síncrona de escrita, com dados e endereços

disponibilizados numa configuração “*merged*”

**6) Um árbitro de um barramento multimaster baseado em prioridade FIFO garante:**

**a)** a ausência de fenómenos de starvation

**b)** que é sempre servido o master de maior prioridade com pedido pendente de atribuição de barramento

**c)** que a atribuição do barramento é fixado pela ordem temporal inversa com que os masters fazem os seus pedidos

**d)** nenhuma das respostas estão correctas

**7) Numa transferência por DMA, o respectivo controlador:**

**a)** pode usar o barramento em qualquer instante de acordo com as suas necessidades

**b)** apenas pode usar o barramento quando o CPU não estiver a aceder à memória ou a unidades de I/O.

**c)** pode usar o barramento sempre que o árbitro lhe dê permissão para o fazer

**d)** pode usar o barramento em qualquer instante de acordo com as suas necessidades, desde que respeite um tempo mínimo entre utilizações

**8) Para a transferência de 2048 words (de 32 bits), um controlador de DMA de 32 bits, dedicado, a funcionar em modo cycle-stealing, necessita de pelo menos:**

**a)** 2048 bus cycles

**b)** 4096 bus cycles

**c)** 8192 bus cycles

**d)** 16384 bus cycles

**9) Num sistema com vários dispositivos interligados por um barramento SPI, em termos de comunicação:**

**a)** A função dos dispositivos é fixa à partida, podendo apenas existir um master, sendo os restantes obrigatoriamente slaves

**b)** podem exister, permanentemente, vários masters e vários slaves, configurados nessa função à partida, sendo a comunicação estabelecida entre quaisquer dois desses dispositivos

**c)** em cada instante apenas pode existir um master, sendo os restantes obrigatoriamente slaves, mas um slave pode tornar-se master se ganhar o processo de arbitragem com outro master de inferior prioridade

**d)** a função masterslave de cada dispositivo é imposta de acordo com as necessidades, pelo controlador SPI

**10) Na interface I2C o master selecciona o slave com quem vai comunicar através de:**

**a)** um sinal de selecção que activa antes de iniciar a transferência

**b)** informação transmitida na linha de dados

**c)** um sinal de selecção através do qual é transferido o endereço do slave

**d)** um barramento de endereços de 7 bits a partir do qual cada dispositivo descodifica o seu próprio endereço

**11) A técnica “bit dominante/bit recessivo” é utilizada em:**

**a)** RS232 e CAN

**b)** I2C e SPI

**c)** CAN e I2C

**d)** SPI e CAN

**12) Um cabo USB, de acordo com a versão 2.0 da norma, possui os seguintes condutores:**

**a)** VBUS e GND para alimentação; D+ e D- para transmissão não-diferencial em modo full-duplex

**b)** VBUS e GND para alimentação; D+ e D- para transmissão diferencial em modo full-duplex

**c)** VBUS e GND para alimentação; D+ e D- para transmissão diferencial em modo half-duplex

**d)** VBUS e GND para alimentação; D+ e D- para transmissão não-diferencial em modo half-duplex

**13) O número total de pinos (excluindo as linhas de alimentação) de um circuito integrado de uma memória dinâmica DRAM de 2Mx8, com um sínal único de controlo de leitura/escrita é:**

**a)** 21 **c)** 31

**b)** 22 **d)** 32

**14) Suponha que dispõe de 64 circuitos de memória de 4Mx2. Usando todos estes circuitos é possível construir um módulo de memória de:**

**a)** 4Mx64

**b)** 8Mx32

**c)** 16Mx8

**d)** 32Mx16

**15) 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-back* para indicar que a informação armazenada no respectivo bloco foi alterada

**c)** *write-through* para indicar que o respectivo bloco não está a ser usado

**d)** *write-back* para indicar que o respectivo bloco não está a ser usado

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

**a)** 4

**b)** 128

**c)** 512

**d)** 8192

**17) A técnica de memória virtual permite:**

**a)** a utilização de memória cache no processador para aumentar a dimensão aparente da memória física do sistema

**b)** que a dimensão da memória física disponivel exceda o limite do espaço de endereçamento de um processo

**c)** implementar mecanismos de protecção através da independência dos espaços de endereçamento de cada processo

**d)** todas as restantes respostas estão correctas

**18) A tradução de endereços virtuais em endereços físicos consiste na tradução de:**

**a)** *physical page number* no *virtual page number* e sua justaposição com o *page offset* no endereço produzido pelo CPU

**b)** *physical page offset* no *virtual page offset* e sua justaposição com o *page number* no endereço produzido pelo CPU

**c)** *virtual page offset* no *physical page offset* e sua justaposição com o *page number* no endereço produzido pelo CPU

**d)** *virtual page number* no *physical page number* e sua justaposição com o *page offset* no endereço produzido pelo CPU

**19) As páginas de memória virtual possuem normalmente:**

**a)** a mesma dimensão dos blocos da cache para tornar partido das transferências por DMA

**b)** a mesma dimensão dos blocos da cache para tornar partido das transferências por interrupção

**c)** uma dimensão de alguns milhares de bytes para tirar partido das transferências por DMA

**d)** uma dimensão de alguns milhares de bytes para tirar partido das transferências por interrupção

**20) Num dado processador um endereço virtual é representado com 32 bits, dos quais 10 bits são usados para o *page offset*. Este processador é usado num sistema com 16Bytes de memória física. Nestas circunstâncias, o número de páginas virtuais e físicas é, respectivamente:**

**a)** 4M e 1M

**b)** 4M e 1k

**c)** 4G e 1M

**d)** 4G e 1k

**------------------------------------------------------------------------------------------------------------------------------------------------------------------------------**

**GRUPO II – 0,8 cada**

**21) 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 “1” do sinal é determinado pela constante KA, para se obter à saída um sinal com um período de 2ms e duty-cycle de 25%, as constantes KA e KB deverão valer, respectivamente:**

**a)** 500 e 1500

**b)** 1500 e 500

**c)** 499 e 1499

**d)** 1499 e 499

**22) Considere um sistema baseado num CPU a funcionar a uma frequência de 10MHz, com uma taxa de execução de 2,5 MIPS que processa por interrupção eventos externos periódicos. Se a latência máxima no atendimento a uma interrupção for de 5 ciclos de relógio, e a rotina de serviço à interrupção tiver 30 instruções, a máxima frequência a que esses eventos podem ocorrer é, respectivamente:**

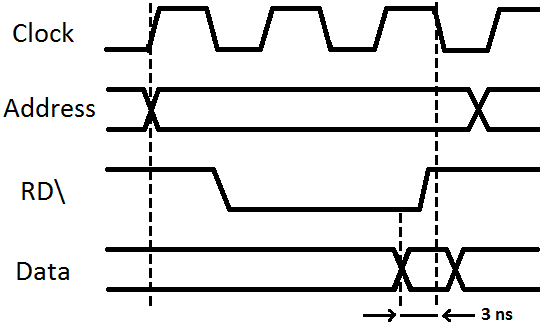
**a)** 71kHz

**b)** 80kHz

**c)** 200kHz

**d)** 285kHz

**23) Considere um CPU a funcionar a uma frequência de 50MHz ligado a uma memória com um tempo de acesso de 38ns. O CPU suporta transferências do 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 5ns e o descodificador que gera o sinal de selecção para a memória apresenta um atraso de propagação de 7ns. Para que este sistema funcione correctamente, o número de wait-states que é necessário introduzir no ciclo de leitura é:**

****

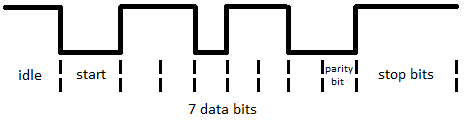
**a)** 0

**b)** 1

**c)** 2

**d)** 3

**24) 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 receptor:**

**** **a)** vai detectar um erro de paridade

**b)** vai detectar uma trama inválida devido a um

número incorrecto de stop bits

**c)** vai detectar um erro de paridade e uma trama

inválida devido a um número incorrecto de stop bits

**d)** não vai detectar qualquer erro

**25) Considere um processador com um espaço de endereçamento de 32 bits e uma memória cache com associatividade de 2, de 16kByte e blocos de 32 bytes. A dimensão, em bits, dos campos *tag*, *set*, e *byte* é:**

**a)** tag: 19; set: 8; byte: 5

**b)** tag: 18; set: 9; byte: 5

**c)** tag: 13; set: 14; byte: 5

**d)** tag: 14; set: 16; byte: 2

**26) O número total de bits de armazenamento (dados e controle) necessário para a implementação de uma memória cache com associatividade de 2, de 16kByte e blocos de 32bytes, com política de escrita do tipo *write-back*, num espaço de endereçamento de 32 bits, é:**

**a)** 140288

**b)** 140800

**c)** 141312

**d)** 141824

**27) 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

**------------------------------------------------------------------------------------------------------------------------------------------------------------------------------**

**GRUPO III – 0,8 cada**

**Um sistema possui um espaço de endereçamento virtual de 4Gbytes, páginas de memória de 8kBytes e 512Mbytes de memória física. Considere tambem:**

* **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:**

|  |  |  |
| --- | --- | --- |
| *Valid, Dirty, Read, Write, Execute flags*  *[31:27]* | *Bits não usados*  *[26:16]* | *PPN*  *[15:0]* |

* **O conteúdo de algumas posições da memória principal a seguir indicados:**

|  |  |
| --- | --- |
| **Endereço** | **Valor** |
| **...** | |
| **0x01230008** | **0xB0000002** |
| **0x0123000C** | **0xF0000003** |
| **0x01230010** | **0xB8000001** |
| **0x01230014** | **0xA0000000** |
| **...** | |

**28) Num acesso à memoria, o CPU produz o endereço 0x00006000, o qual é traduzido no seguinte endereço físico:**

**a)** 0x0000600C

**b)** 0x0000200C

**c)** 0x0000100C

**d)** nenhuma das respostas estão correctas

**29) O processo em execução pode aceder ao endereço virtual 0x0000A010 para:**

**a)** leitura

**b)** escrita

**c)** leitura e escrita

**d)** nenhuma das respostas está correcta

**30) No endereço virtual 0x00008008 do processo em execução encontra-se:**

**a)** uma word do segmento de dados estáticos inicializados

**b)** uma word do segmento de dados estáticos não inicializados

**c)** uma word da stack

**d)** o código máquina de uma instrução

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | a | b | c | d |
| 1 |  |  |  |  |
| 2 |  |  |  |  |
| 3 |  |  |  |  |
| 4 |  |  |  |  |
| 5 |  |  |  |  |
| 6 |  |  |  |  |
| 7 |  |  |  |  |
| 8 |  |  |  |  |
| 9 |  |  |  |  |
| 10 |  |  |  |  |
| 11 |  |  |  |  |
| 12 |  |  |  |  |
| 13 |  |  |  |  |
| 14 |  |  |  |  |
| 15 |  |  |  |  |
| 16 |  |  |  |  |
| 17 |  |  |  |  |
| 18 |  |  |  |  |
| 19 |  |  |  |  |
| 20 |  |  |  |  |
| 21 |  |  |  |  |
| 22 |  |  |  |  |
| 23 |  |  |  |  |
| 24 |  |  |  |  |
| 25 |  |  |  |  |
| 26 |  |  |  |  |
| 27 |  |  |  |  |
| 28 |  |  |  |  |
| 29 |  |  |  |  |
| 30 |  |  |  |  |