






## legenda de cores:

-  ambos os Gui's fizeram e a resposta coincide
-  ambos os Gui's fizeram mas a resposta não coincide (tentar perceber porquê)
-  só o Gui L fez o exercício
-  só o Gui C fez o exercício
-  nenhum dos Gui's fez o exercício

## INTRODUÇÃO

### **1. Num sistema computacional, como definiria com as suas próprias palavras o conceito de espaço de endereçamento?**

Num sistema computacional o espaço de endereçamento define o range de endereços de memória válidos, ou seja, os endereços disponíveis para um dado programa ou processo aceder.

### **2. Quando, num sistema computacional, estamos a determinar, em função do endereço presente no barramento, qual o periférico ou memória que deve ser selecionada, estamos perante uma operação que é normalmente designada por?**

Descodificação de endereços.

### **3. Quando nos referimos ao tipo de organização de memória num sistema, o que significa dizer que este é:**

- a. byte-addressable**
- b. bit-addressable**
- c. word-addressable**

a. Cada byte tem o seu único endereço que pode ser acedido individualmente, a unidade de memória mais pequena que pode ser endereçada é o byte.

b. Cada bit tem o seu único endereço que pode ser acedido individualmente, a unidade de memória mais pequena que pode ser endereçada é o bit. Este tipo de organização de memória implica custos elevados devido à sua complexidade.

c. Cada word tem o seu único endereço que pode ser acedido individualmente, a unidade de memória mais pequena que pode ser endereçada é então uma word. Uma word é então uma sequência ordenada de bytes/bits.

**4. Identifique e descreva pelas suas palavras qual o papel, na arquitetura de um sistema computacional:**

- a. do Data Bus**
- b. do Address Bus**
- c. do Control Bus**

a. O barramento de dados é responsável por transferir informação entre o CPU, a memória principal e os módulos de I/O.

b. O barramento de endereçamento é responsável por transferir os endereços de memória, onde vai ser escrita ou lida a informação que está no barramento de dados, especificados pelo CPU ou um dispositivo DMA.

c. O barramento de controlo é responsável por especificar o tipo de operação a realizar e assegurar a comunicação entre os vários elementos do sistema computacional, de modo a conseguir que a informação enviada pelo barramento de dados é escrita ou lida no endereço de memória especificado pelo barramento de endereçamento.

**5. Na arquitetura de um sistema computacional, como designa o barramento que permite identificar, na memória ou num periférico, a origem/destino da informação a transferir.**

Barramento de endereçamento.

**6. Na arquitetura de um sistema computacional, como designa o barramento que permite especificar o tipo de operação efetuada sobre a memória ou sobre um periférico.**

Barramento de controlo.

## **MICROCONTROLADORES (EMBEDDED SYSTEMS)**

**7. Um compilador-cruzado (cross-compiler) é o nome dado a um tipo específico de programa. Como descreveria, nas suas próprias palavras o que caracteriza este tipo de programa?**

Um cross-compiler é um compilador capaz de criar código executável para um plataforma diferente da qual está a executar o compilador.

**8. Identifique qual a função de um bootloader num sistema baseado em microcontrolador?**

Um bootloader é um programa que reside na memória do sistema e que permite as funções básicas de transferência e execução de um programa.

**9. Quando falamos em microcontroladores, por oposição a um sistema computacional de uso geral, o que podemos afirmar:**

- a. quanto aos principais aspetos da sua arquitetura interna**
- b. quanto à sua frequência de trabalho**
- c. quanto à disponibilização de periféricos**
- d. quanto ao custo**
- e. quanto à energia consumida**
- f. quanto aos seus campos de aplicação**

- a. Circuito integrado com CPU, RAM, ROM, periféricos.
- b. Frequências de funcionamento muito inferiores (< 200 MHz) comparadas com um sistema de uso geral (> 3 GHz).
- c. Grande variedade de periféricos e interfaces com o exterior.
- d. Custo baixo comparado com um sistema de uso geral.
- e. Baixo consumo energético comparado com um sistema de uso geral.
- f. Utilizado em várias áreas em que a aplicação não necessite de elevados recursos computacionais , normalmente, em sistemas embebidos devido ao seu baixo custo.

**10. Como descreveria as principais características de um sistema embebido?**

Já chega de palha também...

**11. Um microcontrolador PIC32 usa uma arquitetura pipelined semelhante à estudada nas aulas de AC1. Descreva qual o modelo base da arquitetura usada e o tipo ou tipos de memória usadas**

**pelo sistema.**

Baseado numa arquitetura Harvard.

## **MÓDULOS DE I/O**

**12. Na arquitetura de um microcontrolador PIC32 qual a finalidade e funcionalidade dos seguintes registos:**

- a. TRIS**
- b. LAT**
- c. PORT**

- a.** Configurar o porto como entrada ou saída.
- b.** Ler e escrever valores num porto de saída.
- c.** Ler valores num porto de entrada.

**13. Num porto de I/O do PIC32 (esquema abaixo), quando o porto se encontra configurado como porto de entrada,**

- a. continua a ser possível escrever no registo de saída?**
- b. continua a ser possível ler o valor que se encontra armazenado no registo LAT?**
- c. se sim, qual o sinal que permite realizar essa leitura?**

- a.** Sim.
- b.** Sim.
- c.** RD\_LAT.

**14. Num porto de I/ O do PIC32 (esquema abaixo), se eu não souber qual a configuração do mesmo (saída ou entrada),**

- a. será possível saber essa informação programaticamente?**
- b. se sim, qual o sinal de controlo que permite essa operação?**

- a.** Sim
- b.** RD\_TRIS (???)

**15. Considere que está a decorrer uma operação sobre um porto de I/O de um PIC32.**

- a. o porto encontra-se configurado como entrada ou saída?**
- b. identifique e descreva qual a operação que está em curso.**

- a.** O porto encontra-se configurado como saída.
- b.** Operação de escrita do valor '0' no registo LAT.

**16.** Repita o exercício anterior para o esquema apresentado abaixo.

- a.** O porto encontra-se configurado como entrada.
- b.** Operação de leitura do registo PORT.

**17.** Na implementação da parte de dados de um porto de saída, que tipo de dispositivo lógico deve ser usado para armazenar o valor transferido através do barramento de dados durante um ciclo de escrita:

Flip-Flop tipo D.

**18.** Na implementação de um porto de I/O do PIC32, o registo PORT está associado a um conjunto de dois flip-flops D em série (shift register de dois andares). Qual o objetivo dessa implementação?

Sincronização da leitura dos dados com a frequência do sistema.

**19.** Pretende usar-se o porto RB do microcontrolador PIC32MX795F512H para realizar a seguinte função (em ciclo infinito): O byte menos significativo ligado a este porto é lido com uma periodicidade de 100ms. Com um atraso de 10ms, o valor lido no byte menos significativo é colocado, em complemento para 1, no byte mais significativo desse mesmo porto. Escreva, em assembly do MIPS, um programa que execute esta tarefa.

- a.** configure o porto RB para executar corretamente a tarefa descrita
- b.** efetue a leitura do porto indicado
- c.** execute um ciclo de espera de 10ms
- d.** efetue a transformação da informação lida para preparar o processo de escrita naquele porto
- e.** efetue, no byte mais significativo, o valor resultante da operação anterior
- f.** execute um ciclo de espera de 90ms
- g.** regresse ao ponto b

Not today son, not today...

## **NOÇÕES BÁSICAS SOBRE PERIFÉRICOS**

**20. A descrição da funcionalidade de um dado dispositivo periférico, o seu conjunto de registos de dados, de controlo e de status é genericamente designada, no contexto de arquitetura de computadores, por uma designação específica. Qual essa designação?**

Módulo de I/O.

**21. Quando, no acesso que o CPU faz a um módulo de E/S, é usada a técnica de entrada/saída de dados por software (programada), quais as tarefas que são realizadas pelo CPU?**

O CPU verifica o registo/os de estado até o periférico estar disponível para a transferência de informação (polling) e depois inicia e controla esta mesma transferência.

**22. O método de transferência de informação entre um CPU e um módulo de E/S (I/O), em que o programa executado no CPU é responsável por iniciar, monitorizar e controlar a transferência de informação, designa-se por:**

Programmed I/O.

**23. Quando nos referimos a um “Módulo de I/O”, estamos a referir-nos especificamente a que parte do periférico de que este módulo faz parte?**

“O módulo de I/O pode assim ser visto como um módulo de compatibilização entre as características e modo de funcionamento do sistema computacional e o dispositivo físico propriamente dito.”

**24. Na implementação da parte de dados de um porto de entrada de um módulo de I/O:**

**a. que tipo de dispositivos lógicos devem ser usados na ligação ao barramento de dados?**

**b. por que razão é fundamental usar esses dispositivos?**

- a. Portas Tri-state
- b. bla bla bla...

**25.** Os diagramas temporais que se seguem, nas várias figuras, representam operação de transferência de informação (leitura ou escrita) de/para dispositivos que podem estar mapeados em espaços de endereçamento de I/O ou de memória. Identifique, para cada diagrama qual o tipo de operação e espaço de endereçamento utilizado:

- (1,1) Leitura M
- (1,2) Leitura I/O
- (1,3) Escrita I/O
- (2,1) Escrita I/O
- (2,2) Escrita M
- (2,3) Leitura M

**26.** Nos diagramas temporais, incompletos, das duas figuras abaixo, complete a representação gráfica dos mesmos para:

- a. uma operação de escrita num dispositivo mapeado no espaço de endereçamento de memória
- b. uma operação de leitura a partir de um dispositivo mapeado no espaço de endereçamento de I/O

(???)

## **INTERRUPÇÕES**

**27.** No que respeita ao sistema de interrupções do PIC32 é (usado na placa DETPIC32):

- a. descreva sucintamente os dois modos de funcionamento do sistema de interrupções
- b. do caso do sistema de interrupções feita por hardware, como descreveria sucintamente o seu funcionamento
- c. ainda no caso da alínea anterior, o que determina a ordem pela qual as interrupções são servidas nos casos em que ocorram em simultânea a partir de mais do que uma fonte

- a. Not today...
- b. Not today...
- c. A ordem de atendimento depende do valor da prioridade configurada de cada fonte presente nos 3 bits do registo IPCx que codificam 7 níveis de prioridade (1 a 7, o valor 0 significa fonte desativada).

**28.** Numa RSI, qual o objetivo do conjunto de instruções designado por:

- a. "prólogo"?
- b. "epílogo"?

- a. Salvaguarda do contexto do programa interrompido, isto é, guardar o valor dos registos internos do CPU na stack.
- b. Reposição do contexto do programa interrompido, isto é, repor o valor dos registos internos do CPU que estão guardados na stack.

**29.** Descreva, por palavras suas, o que se entende por overhead da transferência de informação por interrupção e as razões que justificam esse overhead.

Neste contexto o Overhead é o número de instruções que resultam da mudança de contexto do programa, ou seja, o número de instruções para salvarguardar o contexto do programa (Prólogo) mais o número de instruções para repor o contexto do programa (Epílogo).

**30.** Considere um sistema baseado num CPU a funcionar a uma frequência de 10 MHz com uma taxa de execução de 5 MIPS ( $5 \times 10^6$  instruções por segundo,  $CPI = 2$ ) que processa por interrupção eventos externos periódicos. Se o overhead total do atendimento a à interrupção for de 20 ciclos de relógio, e a rotina de serviço à interrupção tiver 40 instruções, determine a máxima frequência a que esses eventos podem ocorrer para que todas as interrupções possam ser atendidas.

$RSI = \text{Overhead (20 ciclos)} + 40 \text{ instruções}$ ,  $40 \text{ instruções} = 80 \text{ ciclos}$ ,  $RSI = 20 + 80 = 100 \text{ ciclos}$

$1 \text{ ciclo} \Rightarrow 0.1 \text{ us}$ ,  $100 \text{ ciclos} \Rightarrow 10 \text{ us}$ ,  $10 \text{ us} \Rightarrow \mathbf{100 \text{ kHz}}$

**31.** Considere um sistema baseado num CPU a funcionar a uma frequência de 40 MHz com uma taxa de execução de 16 MIPS ( $16 \times 10^6$  instruções por segundo,  $CPI = 2.5$ ). Pretende-se processar por interrupção eventos externos periódicos que



ocorrem a uma frequência de 200 kHz. Para cumprir este requisito e sabendo que o overhead total do atendimento a uma interrupção é 75 ciclos de relógio, calcule o número máximo de instruções máquina que a rotina de serviço à interrupção pode ter.

$$N_{\text{max}} \text{ ciclos} = N + 75$$

$$200 \text{ kHz} \Rightarrow 5 \text{ us}$$

$$5 \text{ us} * 40 \text{ MHz} \Rightarrow 200 \text{ ciclos}$$

$$N_{\text{max}} \text{ ciclos} = 200 = N + 75 \Leftrightarrow N = 125 \text{ ciclos}$$

$$N_{\text{max}} \text{ instruções} = 125/2.5 + 75/2.5 = 50 + 30 = 80 \text{ instruções}$$

**NRSI = 50 instruções**

**32.** Recalcule a solução para o problema anterior admitindo agora que o CPU a funciona a uma frequência de 100 MHz com uma taxa de execução de 33.3 MIPS ( $33.3 \times 10^6$  instruções por segundo, CPI = 3.0) e que se pretende processar por interrupção eventos externos periódicos que ocorrem a uma frequência de 500 kHz. Admita ainda que o overhead total do atendimento à interrupção é 80 ciclos de relógio.

$$N_{\text{max}} \text{ ciclos} = N + 80$$

$$500 \text{ kHz} \Rightarrow 2 \text{ us}$$

$$2 \text{ us} * 100 \text{ MHz} \Rightarrow 200 \text{ ciclos}$$

$$N_{\text{max}} \text{ ciclos} = 200 = N + 80 \Leftrightarrow N = 120 \text{ ciclos}$$

$$N_{\text{max}} \text{ instruções} = 120/3 + 80/3 = 40 + 27 = 67 \text{ instruções}$$

**NRSI = 40 instruções**

**33.** Considere um sistema baseado num CPU a funcionar a uma frequência de 10 MHz com uma taxa de execução de 5 MIPS ( $5 \times 10^6$  instruções por segundo, CPI = 2) que processa por interrupção eventos externos periódicos. A rotina de serviço à interrupção tem 70 instruções e verificou-se experimentalmente que a máxima frequência a que os eventos externos podem ocorrer é 50 kHz. Nestas condições determine, em ciclos de relógio, qual o valor máximo que pode ser usado pelo overhead total do atendimento.

$$\text{RSI tem } 70 \text{ instruções} \Rightarrow 140 \text{ ciclos}$$

$$50 \text{ kHz} \Rightarrow 20 \text{ us}$$

$$20 \text{ us} * 10 \text{ MHz} = 200 \text{ ciclos}$$

$$200 = 140 + \text{Overhead} \Leftrightarrow \text{Overhead} = \mathbf{60 \text{ ciclos}}$$

**34.** Considere agora um sistema baseado num CPU a funcionar a uma frequência de 80 MHz com uma taxa de execução de 40 MIPS ( $40 \times 10^6$  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, determine a máxima frequência a que esses eventos podem ocorrer para que todas as interrupções possam ser atendidas.

Overhead = 40 ciclos, RSI = 20 instruções  $\Rightarrow$  40 ciclos

N ciclos = 40 + 40 = 80

1 ciclo  $\Rightarrow$  12.5 ns

80 ciclos  $\Rightarrow 80 \times 12.5 \text{ ns} = 1 \text{ us}$

1 us  $\Rightarrow$  **1MHz**

**35.** Descreva sucintamente, para o caso de um sistema de interrupções vetorizadas com prioridade estabelecida por “daisy chain”:

a. Como é estabelecida a prioridade de resposta a interrupções simultâneas

b. Como se designa o sinal por hardware que permite estabelecer e assegurar o funcionamento do sistema em “daisy chain”.

a. Pela ordem da colocação dos periféricos na cadeia.

b. Inta.

**36.** Descreva, sucintamente, as fases temporais de atendimento a uma interrupção num sistema de interrupções vetorizadas.

- 1) O periférico ativa o sinal Ireq (Interrupt Request).
- 2) Depois de detetar o sinal Ireq ativo o CPU, quando disponível para atender a interrupção, ativa o sinal Inta (Interrupt Acknowledge).
- 3) O periférico depois de detetar o sinal Inta ativo identifica-se, colocando

no barramento de dados o seu vetor

4) O CPU desativa o sinal Ireq (no caso do MIPS este processo é feito por software) e armazena o vetor.

5) O CPU a partir do vetor do periférico e da tabela de vetores salta para a RSI.

**37. Como designaria a organização de um sistema de atendimento a interrupções em que a identificação, pelo CPU, do periférico gerador da interrupção é feita por hardware, num ciclo de interrupt acknowledge durante o qual o periférico gerador da interrupção coloca o seu vetor no barramento de dados.**

Daisy Chain

**ax**

**38. Descreva, sucintamente, o funcionamento de um sistema de interrupções baseado em "identificação da fonte por software"**

O periférico ativa o sinal Ireq e de seguida a RSI lê o registo de status de cada um dos periféricos até encontrar um que tenha gerado um pedido de interrupção.

## **DMA**

**39. Descreva a sequência de operações para que possa ocorrer uma transferência por DMA, em modo bloco, quando o controlador de DMA pretende dar início a uma transferência.**

1) O DMA ativa o sinal BusReq à espera que o CPU esteja disponível para libertar os barramentos.

2) O CPU coloca os seus barramentos em alta impedância e ativa o sinal BusGrant (o DMA passa a ser o Bus Master).

...

**40. Qual a operação que, tipicamente, um controlador de DMA executa quando conclui um processo de transferência de informação enquanto bus master.**

1) Desativa o sinal Dack.

2) Desativa o sinal BusReq.

3) Ativa o sinal de interrupção.

**41. Descreva, sucintamente, qual a finalidade do sinal bus grant num sistema que suporte transferência de dados por DMA, quem gera este sinal e em que circunstâncias**

TODO...

**42. Descreva, sucintamente, qual a sequência de eventos que ocorrem numa transferência por DMA, em modo cycle-stealing, quando o controlador de DMA pretende dar início a uma transferência elementar.**

TODO...

**43. Descreva, sucintamente, qual a diferença entre os modos de operação “bloco” e “burst” de um controlador de DMA.**

TODO...

**44. Considere um controlador de DMA não dedicado, a funcionar em modo bloco, em que um bus cycle é realizado em 1 ciclo de relógio. Calcule o tempo necessário para efetuar a transferência de um bloco de dados para as seguintes condições:**

**a. controlador de 32 bits, frequência de funcionamento do DMA de 500MHz, bloco de 512 words de 32 bits**

**b. controlador de 16 bits, frequência de funcionamento do DMA de 1GHz, bloco de 512 words de 32 bits**

**c. controlador de 16 bits, frequência de funcionamento do DMA de 500MHz, bloco de 2K words de 16 bits**

**a.** 500MHz => 2 ns (1 bus cycle)

DMA não dedicado implica 2 cycles => 4 ns

$$T_{512} = 4\text{ns} * 512 = \mathbf{2048\ ns}$$

**b.** 1 GHz => 1 ns (1 bus cycle)

DMA não dedicado implica 2 cycles => 2 ns

$$T_{512} = 2\ \text{ns} * 512 * 2 = \mathbf{2048\ ns}$$

**c.** 500MHz => 2 ns (1 bus cycle)

DMA não dedicado implica 2 cycles => 4 ns

$$T_{2K} = 4\ \text{ns} * 2048 = \mathbf{8.19\ us}$$

**45. Volte a resolver o problema anterior considerando agora que um bus cycle é realizado em 2 ciclos de relógio e para as seguintes condições: 512 words de 32 bits é:**

**a. controlador de 32 bits, frequência de funcionamento do DMA de 1GHz, bloco de 1K words de 32 bits**

**b. controlador de 16 bits, frequência de funcionamento do DMA de 500MHz, bloco de 2K words de 32 bits**

**c. controlador de 16 bits, frequência de funcionamento do DMA de 1GHz, bloco de 256 words de 32 bits**

**d. controlador de 16 bits, frequência de funcionamento do DMA de 500MHz, bloco de 2K words de 16 bits**

**a.** 1 GHz => 1 ns, 1 bus cycle => 2 ns

DMA não dedicado implica 2 cycles => 4 ns

$$T_{1K} = 4ns * 1K = \mathbf{4096\ ns}$$

**b.** 500 MHz => 2 ns, 1 bus cycle => 4 ns

DMA não dedicado implica 2 cycles => 8 ns

$$T_{2K} = 8ns * 2K * 2 = \mathbf{32.77\ us}$$

**c.** 1 GHz => 1 ns, 1 bus cycle => 2 ns

DMA não dedicado implica 2 cycles => 4 ns

$$T_{256} = 4ns * 256 * 2 = \mathbf{2048\ ns}$$

**d.** 500 MHz => 2 ns, 1 bus cycle => 4 ns

DMA não dedicado implica 2 cycles => 8 ns

$$T_{2K} = 4ns * 2K = \mathbf{8192\ ns}$$

**46. Resolva as duas primeiras alíneas do problema anterior considerando agora que o controlador é dedicado.**

Como o controlador é dedicado, precisa apenas de um bus cycle. Basta dividir cada resultado por 2, ou seja, com um controlador dedicado o tempo de transferência será metade.

**47. Considere agora um controlador de DMA não dedicado, a funcionar em modo cycle-stealing, em que um bus cycle é realizado em 2 ciclos de relógio e o tempo mínimo entre operações elementares é 1 ciclo de relógio. Calcule o tempo**

**necessário para efetuar a transferência de um bloco de dados para as seguintes condições:**

- a. controlador de 32 bits, frequência de funcionamento do DMA de 250 MHz, bloco de 512 words de 32 bits**
- b. controlador de 16 bits, frequência de funcionamento do DMA de 1GHz, bloco de 512 words de 32 bits**
- c. controlador de 16 bits, frequência de funcionamento do DMA de 500MHz, bloco de 2K words de 16 bits**

Controlador não dedicado implica que o fetch + deposit requerem 2 bus cycles.  
Cycle Stealing implica fetch-wait-deposit-wait. Cada wait equivale 1 ciclo de relógio

**a.** 250 MHz => 4 ns

$$T_{1 \text{ word}} = (2 \cdot 4 + 4 + 2 \cdot 4 + 4) \text{ ns} = 24 \text{ ns}$$

$$T_{512} = 24 \text{ ns} \cdot 512 = \mathbf{12.288 \text{ us}}$$

**b.** 1 GHz => 1 ns

$$T_{1 \text{ word}} = (2 \cdot 1 + 1 + 2 \cdot 1 + 1) \cdot 2 \text{ ns} = 12 \text{ ns}$$

$$T_{512} = 12 \text{ ns} \cdot 512 = \mathbf{6.144 \text{ us}}$$

**c.** 500 MHz => 2 ns

$$T_{1 \text{ word}} = (2 \cdot 2 + 2 + 2 \cdot 2 + 2) \cdot 2 \text{ ns} = 24 \text{ ns}$$

$$T_{2K} = 24 \text{ ns} \cdot 2048 = \mathbf{49.152 \text{ us}}$$

**48. Determine o número de bus cycles necessários para efetuar uma transferência por um controlador de DMA dedicado a funcionar em modo bloco, dadas as seguintes condições:**

- a. controlador de 32 bits, frequência de funcionamento do DMA de 500 MHz, bloco de 512 words de 32 bits**
- b. controlador de 16 bits, frequência de funcionamento do DMA de 1GHz, bloco de 4K words de 32 bits**
- c. controlador de 16 bits, frequência de funcionamento do DMA de 1GHz, bloco de 512 words de 32 bits**
- d. controlador de 16 bits, frequência de funcionamento do DMA de 500MHz, bloco de 1K words de 16 bits**

DMA dedicado implica que o fetch + deposit é feito em 1 bus cycle

**a.** 512 bus cycles

**b.**  $4K \cdot 2 = 8192$  bus cycles

**c.**  $512 \cdot 2 = 1024$  bus cycles

**d.**  $1K = 1024$  bus cycles

**49. Determine a taxa de transferência de pico (expressa em**

**Bytes/s) de um DMA não dedicado de 32 bits (i.e. com barramento de dados de 32 bits), a funcionar a 100 MHz em modo "cycle-stealing". Suponha ainda que são necessários 2 ciclos de relógio ( $T_{BC}$ ) para efetuar uma operação de leitura ou escrita. desse DMA. O tempo mínimo entre operações elementares deverá ser de  $1 T_{BC}$**

100 MHz  $\Rightarrow$  10 ns

deposit/fetch  $\Rightarrow$  2 ciclos de relógio  $\Rightarrow$  20 ns

wait  $\Rightarrow$  2 ciclos de relógio  $\Rightarrow$  20 ns

fetch + wait + fetch + wait =  $20 \times 4$  ns = 80 ns

Em 80 ns consegue-se transferir 1 word de 32 bits, 4 bytes.

80 ns  $\Rightarrow$  12.5 M de transferências de 1 word (4 bytes) por segundo.

$12.5 \text{ M} \times 4 = \mathbf{50 \text{ MB/s}}$

**50. Resolva o problema anterior, mas considerando agora as seguintes condições:**

**a. controlador de 32 bits, frequência do DMA de 120 MHz,  $1 T_{BC} = 2$  ciclos de relógio, tempo mínimo entre operações elementares de  $2 T_{BC}$**

**b. controlador de 32 bits, frequência do DMA de 80 MHz,  $1 T_{BC} = 2$  ciclos de relógio, tempo mínimo entre operações elementares de  $3 T_{BC}$**

**c. controlador de 16 bits, frequência do DMA de 120 MHz,  $1 T_{BC} = 2$  ciclos de relógio, tempo mínimo entre operações elementares de  $2 T_{BC}$**

**d. controlador de 16 bits, frequência do DMA de 200 MHz,  $1 T_{BC} = 1$  ciclos de relógio, tempo mínimo entre operações elementares de  $1 T_{BC}$**

**a.**

120 MHz  $\Rightarrow$  8.3 ns

deposit/fetch  $\Rightarrow$  2 ciclos de relógio  $\Rightarrow$  16.6 ns

wait  $\Rightarrow$  4 ciclos de relógio  $\Rightarrow$  33.2 ns

fetch + wait + fetch + wait =  $(16.6 + 33.2 + 16.6 + 33.2)$  ns = 99.6 ns

Em 99.6 ns consegue-se transferir 1 word de 32 bits, 4 bytes.

99.6 ns  $\Rightarrow$  10 M de transferências de 1 word (4 bytes) por segundo.

$10 \text{ M} \times 4 = \mathbf{40 \text{ MB/s}}$

**b.**

80 MHz  $\Rightarrow$  12.5 ns

deposit/fetch  $\Rightarrow$  2 ciclos de relógio  $\Rightarrow$  25 ns

wait  $\Rightarrow$  6 ciclos de relógio  $\Rightarrow$  75 ns

fetch + wait + fetch + wait =  $(25 + 75 + 25 + 75)$  ns = 200 ns

Em 200 ns consegue-se transferir 1 word de 32 bits, 4 bytes.  
200 ns => 5 M de transferências de 1 word (4 bytes) por segundo.  
 $5 \text{ M} * 4 = \mathbf{20 \text{ MB/s}}$

**c.**

120 MHz => 8.3 ns  
deposit/fetch => 2 ciclos de relógio => 16.6 ns  
wait => 4 ciclos de relógio => 33.2 ns  
fetch + wait + fetch + wait = (16.6 + 33.2 + 16.6 + 33.2) ns = 99.6 ns  
Em 99.6 ns consegue-se transferir 16 bits, 2 bytes.  
99.6 ns => 10 M de transferências de 2 bytes por segundo.  
 $10 \text{ M} * 2 = \mathbf{20 \text{ MB/s}}$

**d.**

200 MHz => 5 ns  
deposit/fetch => 2 ciclos de relógio => 10 ns  
wait => 2 ciclos de relógio => 10 ns  
fetch + wait + fetch + wait =  $10 * 4 \text{ ns} = 40 \text{ ns}$   
Em 40 ns consegue-se transferir 16 bits, 2 bytes.  
40 ns => 25 M de transferências de 2 bytes por segundo.  
 $25 \text{ M} * 2 = \mathbf{50 \text{ MB/s}}$

**51. Suponha um 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 para efetuar uma operação de leitura ou escrita (i.e. 1 "bus cycle" é constituído por 2 ciclos de relógio).**

**a. determine a taxa de transferência desse DMA (expressa em Bytes/s), supondo um funcionamento em modo bloco.**

**b. determine a taxa de transferência de pico desse DMA (expressa em Bytes/s), supondo um funcionamento em modo "cycle-stealing" e um tempo mínimo entre operações elementares de 1 ciclo de relógio ("fetch", 1 Tmínimo, "deposit", 1T mínimo).**

**c. Repita as alíneas anteriores supondo um DMA dedicado com as características referidas anteriormente.**

DMA não dedicado implica fetch + deposit requerem 2 bus cycles  
100 MHz => 10 ns  
fetch/deposit => 1 bus cycle => 20 ns

**a.** Uma transferência demora  $20 \text{ ns} + 20 \text{ ns} = 40 \text{ ns}$   
 $40 \text{ ns} => 25 \text{ M de transferências de 4 bytes por segundo}$   
 $25 \text{ M} * 4 = \mathbf{100 \text{ MB/s}}$



**b.** (Assumindo  $T_{\text{minimo}} \Rightarrow 2$  bus cycles)

wait = 10 ns

fetch + wait + fetch + wait = (20 + 10 + 20 + 10) ns = 60 ns

Em 60 ns consegue-se transferir 32 bits, 4 bytes.

60 ns  $\Rightarrow$  16.7 M de transferências de 4 bytes por segundo.

16.7 M \* 4 = **66.8 MB/s**

**c.** Como são os tempos de um DMA dedicado em cycle-stealing (???)

## TIMERS

**52.** 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  $k_1$  e  $k_2$ , determine a expressão algébrica que estabelece a relação entre a frequência à entrada do primeiro ( $f_{\text{in}}$ ) e a frequência à saída do segundo ( $f_{\text{out}}$ ).

T1 com constante de divisão  $k_1$  :  $f_{\text{out}1} = f_{\text{in}} / k_1$

T2 com constante de divisão  $k_2$ :  $f_{\text{out}} = f_{\text{out}1} / k_2 = f_{\text{in}} / k_1 / k_2 = f_{\text{in}} / (k_1 * k_2)$

$$f_{\text{out}} = f_{\text{in}} / (k_1 * k_2)$$

**53.** Descreva, por palavras suas o que se entende por Duty Cycle de um sinal digital periódico. Dê alguns exemplos em que a manipulação dinâmica deste valor pode ser usado em aplicações práticas.

Duty Cycle é razão entre o tempo em que o sinal está ativo e o período do sinal. Exemplos práticos podem ser controlar a intensidade de LED's ou o controlo da velocidade de rotação de um motor elétrico.

**54.** Considere um timer em que a relação entre as frequências de entrada e de saída é dada por  $(k+1)$  em que  $k$  é uma constante configurável. Determine o período do sinal de saída para os valores seguintes:

a. frequência de entrada do timer for 20MHz e  $k = 1999$

b. frequência de entrada do timer for 40MHz e  $k = 1249$

**c. frequência de entrada do timer for 80MHz e k = 32767**

**d. frequência de entrada do timer for 2MHz e k = 1023**

**a.**  $f_{in} = 20 \text{ MHz}, k = 1999$

$$f_{out} = 20M / (1999 + 1) = 10 \text{ kHz}$$

**b.**  $f_{in} = 40 \text{ MHz}, k = 1249$

$$f_{out} = 40M / (1249 + 1) = 32 \text{ kHz}$$

**c.**  $f_{in} = 80 \text{ MHz}, k = 32767$

$$f_{out} = 80M / (32767 + 1) = 2.44 \text{ kHz}$$

**d.**  $f_{in} = 2 \text{ MHz}, k = 1023$

$$f_{out} = 2M / (1023 + 1) = 1.95 \text{ kHz}$$

**55.** Alguns dos timers que estudou e utilizou têm, como último andar do temporizador, um divisor por dois. Descreva, sucintamente, qual a razão e finalidade desse divisor por dois.

A divisão por dois permite obter um sinal à saída com 50% de duty-cycle.

**56.** Considere um timer e uma unidade OC como o da figura abaixo (semelhante aos dos PIC32). Admita que a frequência do relógio TCLK é de 20MHz, que o fator de divisão do prescaler é 4, que o valor armazenado em PR é 2499 e que o valor em OCK é 834. Determine o período do sinal de saída e o respectivo duty cycle.

$$f_{in} = 20 \text{ MHz}$$

$$f_{out \text{ presc}} = 20M / 4 = 5 \text{ MHz}$$

$$f_{out} = 5M / (2499 + 1) = 2 \text{ kHz}$$

$$\text{duty cycle} = 100 * 834 / (2499 + 1) = 33.4\%$$

**57.** Pretende-se gerar um sinal com uma frequência de 85 Hz. Usando o Timer T2 e supondo PBCLK = 50 MHz:

**a.** calcule o valor mínimo da constante de divisão a aplicar ao prescaler e indique qual o valor efetivo dessa constante

**b.** calcule o valor da constante PR2

**a.**  $K_{presc} = \text{ceil}(50M / (85 * (65535 + 1))) = 9$

$$K_{min} \Rightarrow 16$$

b.  $PR2 = (50M / 16 / 85) - 1 = 36765$

**58. Repita o exercício anterior, supondo que se está a usar o Timer T1**

a.  $K_{min} \Rightarrow 64$

b.  $PR1 = (50M / 64 / 85) - 1 = 9190$

**59. Pretende-se gerar um sinal com uma frequência de 100 Hz e 25% de "duty-cycle". Usando o módulo "output compare" OC5 e como base de tempo o Timer T3 e supondo ainda PBCLK = 40 MHz:**

a. determine o valor efetivo da constante de prescaler que maximiza a resolução do sinal PWM

b. determine o valor das constantes PR3 e OC5RS

c. determine a resolução do sinal de PWM obtido

a.  $K_{presc} = \text{ceil}(40M / (100 * (65535 + 1))) = 7$   
 $K_{min} \Rightarrow 8$

b.  $PR3 = (40M / 8 / 100) - 1 = 49999$   
 $OC5RS = (49999 + 1) * 0.25 / 100 = 12500$

c.  $\text{resolução} = \log_2(49999 + 1) = 15.6$

**60. Considere ainda um timer como o da figura acima (semelhante aos dos PIC32) com a sua saída ligada uma unidade OC. Admita que a frequência do relógio TCLK é de 20MHz e que o fator de divisão do prescaler é 8. Determine quais os valores que deverá colocar nos registos em PR e OCR para obter na saída O1 um sinal periódico com uma frequência de 200Hz e um duty cycle de 25%:**

$$f_{presc} = 20MHz / 8 = 2.5MHz$$
$$PR = 2.5M / 200 - 1 = 12499$$
$$OCR = (12499 + 1) * 0.25 = 3125$$

**61.** Considere um timer do tipo A do PIC32 (semelhante ao da figura) e um  $PBCLK = 20\text{MHz}$ . Determine o fator divisão do prescaler e o valor a colocar em PR1 para que o período de fount seja de 15ms, com a melhor precisão possível:

ver ex59

**62.** Considere um timer com reset síncrono em que a relação entre as frequências de entrada e de saída é dada por  $(k+1)$ , sendo k uma constante configurável. Determine o valor de k para as seguintes condições:

- a. frequência de entrada do timer for 20MHz e período do sinal à saída de 5ms
- b. frequência de entrada do timer for 25MHz e período do sinal à saída de 1ms
- c. frequência de entrada do timer for 40MHz e período do sinal à saída de 250ms

TODO

**63.** Calcule qual o tempo máximo entre resets ao um sistema de watchdog timer que gera um sinal de reset ao processador sempre que a contagem atinge o valor máximo. Admita as seguintes condições:

- a. frequência de entrada de 100 kHz, contador crescente de 16 bits
- b. frequência de entrada de 20 kHz, contador crescente de 10 bits
- c. frequência de entrada de 50 MHz, contador crescente de 24 bits

a. 100 kHz => 10 us

$$T_{\text{count}} = 2^{16} * 10 \text{ us} = 0.655 \text{ s} \Rightarrow 1.526 \text{ Hz}$$

b. 20 kHz => 50 us

$$T_{\text{count}} = 2^{10} * 50 \text{ us} = 51.2 \text{ ms} \Rightarrow 19.5 \text{ Hz}$$

c. 50 kHz => 20 us

$$T_{\text{count}} = 2^{24} * 20 \text{ us} = 0.336 \text{ s} \Rightarrow 2.98 \text{ Hz}$$

**64.** Um determinado microcontrolador disponibiliza um watchdog

timer com uma frequência de entrada de 100 kHz. O programa em execução faz, por software, um reset ao watchdog timer com uma periodicidade que pode variar entre [10ms ... 170ms]. Determine o número mínimo de bits do contador do watchdog timer por forma a que este nunca gere um reset ao processador.

$$100 \text{ kHz} \Rightarrow 10 \text{ us}$$

$$T_{\text{count}} = 170 \text{ ms} = 2^b * 10 \text{ us}$$

$$b = \text{ceil}(\log_2(170\text{ms} / 10 \text{ us})) = 15 \text{ bits}$$

**65.** Repita o problema anterior admitindo agora que a frequência de entrada do watchdog timer é de 250 kHz e que o programa em execução faz, por software, um reset ao watchdog timer com uma periodicidade que pode variar entre [125ms ... 480ms]

$$250 \text{ kHz} \Rightarrow 4 \text{ us}$$

$$T_{\text{count}} = 480 \text{ ms} = 2^b * 4 \text{ us}$$

$$b = \text{ceil}(\log_2(480\text{ms} / 4 \text{ us})) = 17 \text{ bits}$$

**66.** O programa em execução num microcontrolador faz, por software, um reset ao watchdog timer com uma periodicidade que pode variar entre [50ms ... 150ms]. O watchdog desse microcontrolador usa um gerador de relógio próprio e um contador binário de 16 bits que, ao chegar ao fim de contagem, gera um reset por hardware ao microcontrolador. Dadas estas condições, e por forma a que o sistema de supervisão funcione adequadamente, determine qual a máxima frequência de relógio aplicada na entrada do watchdog.

$$150 \text{ ms} \Rightarrow 6.667 \text{ Hz}$$

$$f_{\text{watchdog}} \leq 2^{16} * 6.667 \text{ Hz} = 437 \text{ kHz}$$

**NOÇÕES BÁSICAS DE BARRAMENTOS /**

## DESCODIFICAÇÃO DE ENDEREÇOS

**67.** Para um barramento de endereços como o indicado abaixo, que seleciona blocos de memória com 1 Kbyte, suponha que no decodificador apenas se consideram os bits A15, A13 e A11, com os valores 1, 0 e 0, respetivamente.

a. apresente a expressão lógica que implementa este decodificador:

i. em lógica positiva

ii. em lógica negativa

b. indique os endereços inicial e final da gama-base decodificada e de todas as réplicas

- a.
- i.  $SEL = A15.A13 \setminus A11 \setminus$
  - ii.  $SEL \setminus = A15 \setminus + A13 + A11$   
"1X0X0"

b. gama base: [0x8000... 0x83FF]

Replicas: [0x8400... 0x87FF]

[0x9000... 0x93FF]

[0x9400... 0x97FF]

[0xC000... 0xC3FF]

[0xC400... 0xC7FF]

[0xD000... 0xD3FF]

[0xD400... 0xD7FF]

**68.** Considere o exemplo de um espaço de endereçamento indicado na figura abaixo, em que os blocos de memória têm uma dimensão de 4 Kbyte. Admita agora que não vamos decodificar os bits A14 e A12 do bloco dos 4 bits mais significativos, resultando na expressão  $CS \setminus = A15 + A13 \setminus$

a. determine as gamas do espaço de endereçamento de 16 bits ocupadas pela memória.

b. determine os endereços possíveis para aceder à 15ª posição da memória.

- a.
- [0x2000... 0x2FFF]
  - [0x3000... 0x3FFF]
  - [0x6000... 0x6FFF]

[0x7000... 0x7FFF]

b. 0x200E, 0x300E, 0x600E, 0x700E

**69.** Escreva as equações lógicas dos 4 decodificadores necessários para a geração dos sinais de seleção para cada um dos dispositivos identificados na figura ao lado.

**SEL0** = A15\A14\A13\A12\A11\A10\

**SEL1** =

A15\A14\A13\A12\A11\A10.A9\A8.A7\A6\A5\A4\A3\A2\A1\A0\

**SEL2** = A15\A14\A13\A12\A11\A10.A9.A8.A7\A6\A5\A4\A3\A2\

**SEL3** = A15.A14.A13.A12.A11

**70.** Para o exemplo da figura abaixo, determine a gama de endereços em que cada uma das linhas CS<sub>x</sub> está ativa, com a constante de comparação 0x00102 e admitindo que o valor de S é composto por 3 bits e N=32 bits.

(corrigido pelo JLA)

0x00102 => 20 bits = K

K + S + R = N ⇔ 20 + 3 + R = 32 ⇔ R = 9 bits

S = 3 bits => 2<sup>3</sup> = 8 sub-gamas

R = 9 bits => 2<sup>9</sup> = 512 bits dimensão da sub-gama

[0x00102000... 0x001021FF]

[0x00102200... 0x001023FF]

[0x00102400... 0x001025FF]

[0x00102600... 0x001027FF]

[0x00102800... 0x001029FF]

[0x00102A00... 0x00102BFF]

[0x00102C00... 0x00102DFF]

[0x00102E00... 0x00102FFF]

**71.** Para um barramento de endereço de 20 bits, semelhante ao indicado na figura, pretende-se gerar os sinais de seleção para 4 memórias de 8 kByte, a mapear em gamas de endereços consecutivas, de modo a formar um conjunto de 32 kByte. O endereço inicial deve ser configurável. Para um espaço de endereçamento de 20 bits:

a. indique o número de bits dos campos K, S e R, supondo descodificação total

b. esboce o circuito digital que implementa este decodificador

c. indique os endereços inicial e final para a primeira, segunda e última gamas de endereços possíveis de serem decodificadas

d. para a última gama de endereços, indique os endereços inicial e final atribuídos a cada uma das 4 memórias de 8 kbyte

e. suponha que o endereço 0x3AC45 é um endereço válido para aceder ao conjunto de 32k. Indique os endereços inicial e final da gama que inclui este endereço. Indique os endereços inicial e final da memória de 8 KByte à qual está atribuído este endereço

a.  $K + S + R = 20$   
8 kByte  $\Rightarrow R = 13$  bits  
4 memórias  $\Rightarrow S = 2$  bits  
 $K = 20 - 13 - 2 = 5$  bits

b. esboçar mas não é hoje...

c. primeira gama: [0x00000... 0x07FFF]  
segunda gama: [0x08000... 0x0FFFF]  
última gama: [0xF8000... 0xFFFFF]

d. mem0: [0xF8000... 0xF9FFF]  
mem1: [0xFA000... 0xFBFFF]  
mem2: [0xFC000... 0xFDFFF]  
mem3: [0xFE000... 0xFFFFF]

e. [0x38000... 0x3FFFF] gama de endereços  
[0x38000... 0x3AFFF] endereços da primeira memória desta gama

**72.** Pretende-se gerar os sinais de seleção para os seguintes 4 dispositivos: 1 porto de saída de 1 byte, 1 memória RAM de 1 kByte (byte-addressable), 1 memória ROM de 2 kByte (byte-addressable), 1 periférico com 5 registos de 1 byte cada um. O espaço de endereçamento a considerar é de 20 bits.



a. desenhe o gerador de linhas de seleção para estes 4 dispositivos, baseando-se no modelo discutido nos slides anteriores e usando a mesma sub-gama para o periférico e para o porto de saída de 1 byte.

b. especifique a dimensão de todos os barramentos e quais os bits que são usados c. desenhe o mapa de memória com o endereço inicial e final do espaço efetivamente ocupado por cada um dos 4 dispositivos, considerando para o conjunto um endereço-base por si determinado

TODO

**73.** O periférico com 5 registos, do exercício anterior, tem um barramento de endereços com três bits. Suponha que esses bits estão ligados aos bits A0, A1 e A2 do barramento de endereços do CPU.

a. usando o decodificador desenhado no exercício anterior, indique os 16 primeiros endereços em que é possível aceder ao registo 0 (selecionado com A0, A1 e A2 a 0)

b. repita o exercício anterior supondo que os 3 bits do barramento de endereços do periférico estão ligados aos bits A2, A3 e A4 do barramento de endereços.

TODO

**74.** Admita que, num espaço de endereçamento de 16 bits, um decodificador é implementado através da expressão lógica " $CE = A_{15} + A_{14} + A_{12}$ ". Determine qual/quais a(s) gama(s) de endereço(s) que este módulo decodifica:

$CE = A_{15} \cdot A_{14} \cdot A_{12} \Rightarrow 00X1$

[0x1000... 0x1FFF]

[0x3000... 0x3FFF]

**75.** 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 16k endereços consecutivos, num espaço de endereçamento de 32 bits.

a. Determine qual a dimensão em bits (N) e respetiva gama, que deve ser ligado ao primeiro bloco da figura

b. Determine agora qual a dimensão em bits (N) e respetiva gama, que deve ser ligado ao primeiro bloco da figura, se pretendêssemos agora que cada uma das 64 linhas de seleção ativa 1K endereços consecutivos

TODO

**76.** Um programa que pretende transferir 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, determine, em MByte/s, a taxa de transferência máxima que se consegue obter, supondo um barramento de dados de 32 bits.

10 instruções => 25 ciclos

200 MHz => 5 ns

25 ciclos => 125 ns (tempo para completar uma transferência, 4 Bytes)

125 ns => 8 MHz

8M \* 4 = **32 MBytes/s**

**77.** Suponha que dispõe de 16 circuitos de memória de 1Mx4bits. Usando todos estes circuitos, determine qual o tamanho do módulo de memória, com uma dimensão de palavra de 1 byte, que é possível construir:

8Mx8bits

**78.** Para construir um módulo de memória SRAM de 512k x 8 bits, admitindo que dispõe de circuitos de 64k x 8 bits quantos circuitos seriam necessários?

512/64 = 8 circuitos

**79.** Admita um sistema com um barramento de endereço com 16 bits. Supondo que uma memória com 1K registos é selecionada a partir de um descodificador que utiliza os bits [A13..A10]:

a. determine quantas réplicas da memória resultam desta descodificação

b. identifique, justificando, se os espaços de endereçamento são contíguos ou não contíguos

O Gui C tem este como ex81 mas bate com as respostas deste ex

a.  $2^4 = 16$  réplicas

b. Os espaços de endereçamento são contíguos, pois são descodificados através de bits consecutivos.

**80.** O sinal de seleção "Sel\" (ativo baixo) de uma memória de 2k endereços mapeada na gama de endereços 0x00800...0x00FFF, num espaço de endereçamento de 20 bits, pode ser obtido através de uma expressão lógica do tipo:

$$SEL\ = A11\ + \sum_{(i=12 \rightarrow 19)} A_i$$

a. obtenha a expressão do sinal "Sel\" (ativo baixo) um porto mapeado na gama de endereços 0x00400...0x007FF, num espaço de endereçamento de 20 bits

b. obtenha a expressão do sinal "Sel\" (ativo baixo) para as mesmas condições da alínea anterior, mas agora admitindo que o espaço de endereçamento é de 16 bits

c. obtenha a expressão do sinal "Sel\" (ativo baixo) de uma memória de 16k endereços mapeado na gama de endereços 0x8C000...0x8FFFF, num espaço de endereçamento de 20 bits

d. Obtenha a expressão do sinal "Sel" (ativo alto) de um porto mapeado na gama de endereços 0x0000...0x03FF de um processador com um espaço de endereçamento de 16 bits

e. Obtenha a expressão do sinal "Sel" (ativo alto) de uma memória de 4k endereços mapeado na gama de endereços 0x9000...0x9FFF, num espaço de endereçamento de 16 bits

O Gui C tem este como ex79 mas bate com as respostas deste ex

a.  $SEL\ = A10\ + \sum_{(i=11 \rightarrow 19)} A_i$

b.  $SEL\ = A10\ + \sum_{(i=11 \rightarrow 15)} A_i$

c.  $SEL\ = A19\ + A18\ + A17\ + A16\ + A15\ + A14\$

d.  $SEL = \prod_{(i=10 \rightarrow 15)} A_i$

e.  $SEL = A_{15}.A_{14}.A_{13}.A_{12}$

**81. Determine qual a(s) gama(s) de endereço(s) descodificada(s), num espaço de endereçamento de 16 bits, por um descodificador implementado através das seguintes expressões lógicas**

a.  $CE = A_{15} + A_{14} + A_{12}$

b.  $CE = A_{15} + A_{13} + A_{12}$

c.  $CE = A_{23} + A_{21} + A_{20}$ . Admita neste caso que o espaço de endereçamento é de 24 bits

d.  $CE = A_{31} + A_{29} + A_{28}$ . Admita neste caso que o espaço de endereçamento é de 32 bits

O Gui C tem este como ex80 mas bate com as respostas deste ex  
TODO

## ORGANIZAÇÃO DE BARRAMENTOS DE DADOS

**82. Em barramentos paralelo multi-master existem várias técnicas para determinar a prioridade com que o barramento é atribuído a uma dada unidade. Descreva sucintamente cada uma das seguintes:**

a. prioridades fixas

b. critério Last-Come/First-Served

c. critério First-Come/First-Served

d. round-robin

a. Cada unidade tem uma prioridade fixa, assim, quando várias unidades tentarem controlar o barramento ao mesmo tempo a unidade com maior prioridade irá ganhar o controlo do barramento.

b. A última unidade a fazer o pedido é a que vai ganhar o controlo do barramento.

c. A primeira unidade a fazer o pedido é a que vai ganhar o controlo do barramento.

d. O acesso é rotativo, um master quando termina a transferência liberta

o barramento ao master seguinte.

**83. Numa transferência em que o CPU pode prolongar o ciclo de leitura/escrita por um ou mais ciclos de relógio, em função de um sinal de protocolo gerado pelo dispositivo externo podemos afirmar que estamos perante que tipo de transferência? Justifique.**

Transferência semi-síncrona, pois a partir do momento que o dispositivo externo ativa o sinal de protocolo para o CPU prolongar o ciclo de leitura/escrita a transferência torna-se assíncrona. Este ciclo irá prolongar-se até o dispositivo externo desativar este sinal de protocolo.

**84. Identifique que elementos caracterizam um barramento paralelo de tipo síncrono e que tipos de transferência de dados são suportados pelo mesmo.**

1 linha de relógio  
múltiplas linhas para dados

Read-Modify-Write  
Read-After-Write  
Block

**85. Identifique que elementos caracterizam uma transferência assíncrona (handshaken) e quais as vantagens deste tipo de transferência para certas arquiteturas envolvendo pelo menos um master e um conjunto de dispositivos slave.**

Sinais de protocolo "VALID/ACCEPT". Este tipo de transferência tem como principal vantagem a flexibilidade temporal, permitindo que sistemas com vários dispositivos Slave com frequências de funcionamento diferentes possam comunicar.

**86. Considere um barramento paralelo multiplexado de 16 bits (de informação). Sobre esse barramento pretende-se implementar um protocolo de comunicação, de tipo microciclo. Determine o número mínimo de ciclos necessários para completar uma transação sabendo que:**

**a. o espaço de endereçamento é de 32 bits os dados estão organizados em palavras de 32 bits**

- b. o espaço de endereçamento é de 16 bits os dados estão organizados em palavras de 32 bit**
- c. o espaço de endereçamento é de 32 bits os dados estão organizados em palavras de 16 bits**
- d. o espaço de endereçamento é de 48 bits os dados estão organizados em palavras de 16 bits**

- a. 2 ciclos para endereçamento + 2 ciclos para dados = 4 ciclos
- b. 1 ciclo para endereçamento + 2 ciclos para dados = 3 ciclos
- c. 2 ciclos para endereçamento + 1 ciclo para dados = 3 ciclos
- c. 3 ciclos para endereçamento + 1 ciclo para dados = 4 ciclos

**87. Em barramentos multi-master, o que entende por situações de "starvation"**

Starvation acontece quando um elemento não tem acesso ao barramento devido a constantes pedidos de elementos de prioridade superior.

**88. Um programa para transferir 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, determine a taxa de transferência máxima, em Bytes/s, que é possível obter.**

Igual ao exercício 76

**89. Considerando o diagrama temporal da figura ao lado podemos afirmar que estamos perante um ciclo de:**

- a. identifique a natureza da operação**
- b. identifique qual o tipo de transferência que está ali representada**
- c. Identifique qual a configurações da operação de transferência de dados**

- a. Operação de leitura (RD/WR\ a '1')
- b. Transferência síncrona com sinal RD e WR único
- c. no compreendo

**90. Considere um CPU que suporta transferências de tipo síncrono e de tipo semi-síncrono. Para a operação de leitura**

**representada no diagrama ao lado determine o número de wait states necessários para concluir com sucesso essa operação considerando os seguintes pressupostos (Nota: assumo que o tempo mínimo durante o qual os dados têm de estar válidos tem de ser superior a um ciclo de relógio):**

**a. Frequência do CPU= 500 MHz; tempo de acesso à memória de 12 ns (tempo que decorre desde que a memória é selecionada até que a informação fica disponível no data bus); atraso introduzido pelo decodificador de endereços da memória de 2.5 ns**

**b. Frequência do CPU= 200 MHz; tempo de acesso à memória de 35 ns; atraso introduzido pelo decodificador de endereços da memória de 7 ns**

a. 500MHz => 2 ns

$t_{acc} = 12 \text{ ns}$

$t_{decoder} = 2.5 \text{ ns}$

$t_{total} = 12 \text{ ns} + 2.5 \text{ ns} = 14.5 \text{ ns}$

diagrama confuso ...

**91. Descreva, pelas suas próprias palavras as seguintes o que caracteriza as seguintes configurações de operações de transferência de dados:**

- a. merged**
- b. microciclo**
- c. read-modify-write**
- d. read-after-write**
- e. block**

a. A fase de endereçamento engloba a transferência de dados.

b. O endereçamento e a transferência de dados são tratadas como operações autónomas.

c. Operação atômica, indivisível que consegue ler uma posição de memória e escrever na mesma posição de memória um novo valor que pode ser um valor completamente novo ou uma alteração do anterior.

d. O nome é um bocado auto explicativo mas algumas dúvidas surgem...

e. Operação que permite a transferência de um bloco de dados com apenas um ciclo de endereçamento. A primeira transferência requer um ciclo de endereçamento mas as transferências seguintes apenas requerem ciclos de dados.

**92. As figuras abaixo correspondem ao diagrama temporal de duas operações realizada num barramento paralelo:**

- a. como caracterizaria, quanto à natureza da operação uma operação cada uma delas**
- b. face à observação dos diagramas, considera que estamos perante uma operação síncrona, semi-síncrona ou assíncrona**
- c. que tipo de multiplexagem é possível observar no barramento INFO?**
- d. como designaria a configuração da operação realizada?**

- a. Esquerda: escrita  
Direita: Leitura
- b. Protocolo síncrono.
- c. Multiplexagem de endereços e dados.
- d. Microciclo

**93. Identifique uma ou mais técnicas através das quais um árbitro de um barramento multi-master pode evitar situações de "starvation" nos acessos ao barramento.**

Política de arbitragem Round-Robin ou política baseada em prioridades dinâmicas.

**94. Considere um barramento multi-master baseado em prioridades fixas. Neste caso, um árbitro distribuído irá atribuir o barramento a qual dos masters ligados ao barramento?**

À unidade com maior prioridade.

## **BARRAMENTOS DE COMUNICAÇÃO SÉRIE**



**95. Classifique as vantagens dos barramentos série (ao nível físico) quando comparados com barramentos paralelo:**

- a. ao nível da implementação**
- b. ao nível da cablagem de suporte (em barramentos com fios)**
- c. ao nível do custo**
- d. ao nível da distância de transmissão**
- e. ao nível do débito de transmissão**

- a. Mais simples.
- b. Menos cablagem.
- c. Mais barato.
- d. Mais distância de transmissão.
- e. Taxas de transmissão mais elevadas.

**96. O que caracteriza topologicamente um barramento de comunicação série para podermos afirmar que este é um barramento:**

- a. síncrono**
- b. assíncrono**

- a. Barramentos síncronos incluem um sinal de relógio.
- b. Barramentos assíncronos não incluem sinal de relógio e incluem sinais extra de protocolo.

**97. Nos barramentos série com comunicação síncrona, quais os métodos mais comuns para assegurar que os relógios de dois ou mais nós ligados ao barramento se mantêm sincronizados?**

Partilhar o mesmo sinal de relógio.

**98. Nos barramentos série com comunicação síncrona, o que entende por codificação Manchester?**

data XOR clock

ou: '0' significa transição "High to Low" e '1' significa transição "Low to High"

**99. Qual a diferença entre um protocolo de comunicação série full-duplex e um protocolo de comunicação série half-duplex?**

Full-duplex: Tx e Rx podem transmitir dados simultaneamente em ambos os sentidos.

Half-duplex:Tx e Rx podem transmitir dados em ambos os sentidos. mas não em simultâneo.

**100. Dos protocolos de comunicação série que estudou nas aulas teóricas dê exemplos de:**

- a. protocolos full-duplex**
- b. protocolos half-duplex**

- a. SPI, RS232C, USB 3.X
- b. I2C, CAN, USB 1.0 e 2.0

**101. Dos protocolos de comunicação série que estudou nas aulas teóricas existem casos em que a transmissão é orientada ao bit e casos em que a transmissão é orientada ao byte. Explique sucintamente a diferença e dê exemplos de protocolos que usam cada um dos dois.**

Transferência orientada ao bit: a unidade de informação mais pequena, atômica, possível transferir é um bit. Exemplo: CAN, USB.

Transferência orientada ao byte: a unidade de informação mais pequena, atômica, possível transferir é um byte. Exemplo: I2C.

# PROTOCOLO SPI

**102. Como caracterizaria o barramento SPI no que respeita:**

- a. ao tipo de ligação entre dispositivos**
- b. ao tipo de sincronização entre dispositivos**
- c. à natureza da transferência de dados (bidirecional, unidirecional)**

- a. Master-Slave com ligação ponto-a-ponto.
- b. Síncrona (relógio explícito do Master.
- c. Full-Duplex.

**103. Suponha um sistema de medida, baseado no protocolo SPI, que recolhe periodicamente informação proveniente de vários sensores, cada um deles com uma resolução igual (i.e. nº de bits de dados). Determine o tempo mínimo de que o master necessita para adquirir os valores de todos os sensores (cada um implementado num slave distinto), sabendo que:**

- a. o número de sensores é 20, a frequência de relógio é de 100 KHz e a resolução dos sensores é de 16 bits**
- b. o número de sensores é 8, a frequência de relógio é de 20 KHz e a resolução dos sensores é de 8 bits**
- c. o número de sensores é 10, a frequência de relógio é de 100 KHz e a resolução dos sensores é de 8 bits**
- d. o número de sensores é 20, a frequência de relógio é de 100 KHz e a resolução dos sensores é de 16 bits**
- e. o número de sensores é 30, a frequência de relógio é de 50 KHz e a resolução dos sensores é de 8 bit**

TODO

**104. Como caracterizaria um sistema SPI entre as seguintes opções: multi-master assíncrono; multi-master síncrono; ponto a ponto assíncrono; ponto a ponto síncrono.**

Ponto a ponto síncrono.

**105. Diria que o protocolo SPI é adequado para ligação entre dispositivos a longas distâncias? Justifique adequadamente a sua resposta.**

O Protocolo SPI é adequado para curtas distâncias, usado, principalmente, para embedded systems.

**106. Numa arquitetura em que um master SPI de 8 bits se encontra ligado a um conjunto de três slaves organizados em daisy chain como descreveria a interligação dos principais sinais entre o master e os slaves e qual a dimensão das palavras trocadas entre o master e o conjunto de slaves.**

TODO

**107. Descreva sucintamente qual a sequência de operações que são realizadas ao nível do master por forma a assegurar que os seus parâmetros são adequados a realizar validamente troca de informação com um slave a que se encontre ligado.**

- 1- Configurar a frequência do relógio.
- 2- Especificar qual o flanco do relógio usado para a transmissão.

## **I2C (TODO)**

**108. Considere o diagrama temporal representado abaixo. Admita que representa a comunicação I2C entre um master ( $\mu$ C) e um slave (ADC de 10 bits).**

- a. qual o endereço do elemento slave (ADC)?
- b. estamos perante uma operação de escrita ou de leitura?
- c. quantos ACKs são gerados pelo slave?
- d. quantos ACKs são gerados pelo master?
- e. quantos NACKs são gerados? Por quem?
- f. qual o valor (expresso em hexadecimal) que foi fornecido pela ADC ao  $\mu$ C, sabendo que este começa sempre pelo

**MSBit?**

**g. quantas situações de clock stretch são gerados nesta transação? Por quem?**

**h. supondo que a frequência do relógio é de 1 MHz e que o stretch corresponde a dois ciclos de relógio, qual a duração total da transação?**

**109. Descreva sucintamente, no protocolo I2C, como é realizado o endereçamento/seleção do dispositivo a quem é destinada a mensagem ou de quem se pretende obter informação.**

**110. Quantas linhas (físicas) compõem um barramento I2C? Qual a sua designação e finalidade?**

**111. No protocolo I2C em que condições se considera que o barramento de comunicação está livre?**

**112. Descreva sucintamente, no protocolo I2C, quem é responsável pela geração do sinal de relógio e como é possível assegurar a sincronização do mesmo entre master e slave**

**113. Descreva sucintamente, no protocolo I2C, o processo de arbitragem no acesso ao barramento quando dois ou mais masters tentam aceder simultaneamente ao mesmo.**

**114. No protocolo I2C, os bits que circulam no barramento têm uma característica que os distingue dos bits normalmente gerados à saída de um circuito digital convencional. Como designa, no I2C, cada um dos dois estados lógicos, e qual a sua utilidade para o funcionamento do barramento.**

**115. O esquema e o diagrama temporal mostrados abaixo exemplificam a interligação entre um master e um slave e a forma**

como o slave pode alterar o período do sinal de relógio gerado pelo master. Descreva os princípios de funcionamento envolvidos neste processo justificando por que razão esta solução funciona. Apresente uma razão pela qual este método pode ser particularmente interessante numa dada arquitetura.

## **RS232 (TODO)**

**116. Como classificaria a interface RS-232 relativamente: a. topologia da ligação (ponto a ponto, multi-drop, multi-master, ...) e número de ligações físicas do barramento b. direcionalidade das comunicações (half duplex, full-duplex, ...) c. natureza da sincronização (síncrona com relógio explícito, síncrona com relógio implícito, assíncrona, ...) d. formatação da informação (byte oriented, bit oriented, ...)**

**117. Na interface RS-232, caso o desvio de frequência entre o emissor e o recetor seja elevado a informação recebida pode conter erros nos dados recebidos. Indique quais os tipos de erro que podem ocorrer, e descreva os mecanismos que permitem detetar tais erros.**

**118. Na interface RS-232 uma das fontes de erro, do lado do recetor, do instante de amostragem dos vários bits é o "erro de fase". Indique qual a metodologia que é usada para diminuir o impacto dessa fonte de erro e explique sucintamente porque é que esse método contribui para diminuir esse erro.**

**119. Repita o exercício anterior admitindo agora que o baud-rate é 57600 bps, a trama é composta por 8 bits com paridade par e 2 stop bits, sendo o fator de sobre amostragem de 16.**

**120. Admita que a configuração numa comunicação RS-232 (figura abaixo) é 38400 bps, 7 bits sem paridade, 1 stop bit e fator de sobre amostragem de 64. Calcule o valor de frequência ideal no recetor e os intervalos admissíveis dessa frequência para os**

**casos limite (+-25% do tempo de bit e +-75% do tempo de bit). Para isso calcule sucessivamente o seguinte:**

- a. intervalo de validação em períodos do sinal de relógio (TLCLK) para os dois casos limite**
- b. número de períodos de relógio para amostrar a trama**
- c. variação máxima de frequência (em percentagem) para os dois casos limite**
- d. qual o intervalo de frequência que é possível usar para que não haja erros nos casos limite**

**121. Determine a máxima taxa de transmissão de dados líquida (net bit rate) numa ligação RS-232, expressa em bps, admitindo as seguintes configurações:**

- a. baudrate de 19200 bps, 7 bits de dados, 1 bit de paridade e 1 stop bit**
- b. baudrate de 115200 bps, 8 bits de dados, sem bit de paridade e 2 stop bits**
- c. baudrate de 9600 bps, 8 bits de dados, 1 bit de paridade e 2 stop bits**
- d. baudrate de 1200 bps, 7 bits de dados, sem bit de paridade e 1 stop bit**

**122. Considere uma UART configurada para transmitir com os seguintes parâmetros: 100.000 bps, 8 data bits, paridade ímpar e 1 stop bit. Desenhe na figura abaixo a trama completa gerada no envio do valor 0x5A. Indique de forma inequívoca os bits de start, stop e de paridade.**

**123. Um dispositivo com interface RS-232, configurado para transmitir com 7 bits de dados, paridade par e 2 stop bits, produz as duas tramas representadas nos diagramas seguintes que são recebidas por outro dispositivo RS-232 incorretamente configurado para 8 bits de dados, sem paridade e 1 stop bit mas com o mesmo baud rate. Nestas circunstâncias, determine se no recetor vais ser detectado algum erro e porquê. Caso não seja detectado nenhum erro, determine, em hexadecimal, qual o valor recebido.**

**124. Um dispositivo com interface RS232, configurado para**

transmitir 8 bits de dados, paridade ímpar e 1 stop bits, produz a trama acima (2ª) que é recebida por outro dispositivo RS232 incorretamente configurado para 7 bits de dados, sem paridade e 1 stop bit, mas com o mesmo baudrate. Nestas circunstâncias, determine se no recetor vais ser detectado algum erro e porquê. Caso não seja detectado nenhum erro, determine, em hexadecimal, qual o valor recebido.

125. Um dispositivo com interface RS232, 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 configurado com os mesmos parâmetros. No entanto, devido a imprecisão do relógio, o baudrate efetivo do recetor é 7% inferior ao valor do baudrate do transmissor. Nestas circunstâncias, determine se no recetor vais ser detectado algum erro e porquê. Caso não seja detectado nenhum erro, determine, em hexadecimal, qual o valor recebido.

**se der tempo fazer as perguntas dos barramentos  
prioridade aos exercícios de Memórias, Cache e VM**



## **TECNOLOGIA, ORGANIZAÇÃO E FUNCIONAMENTO DE RAMs**

**147.** Numa memória estática SRAM, uma célula de um bit é composta por seis transistores. Descreva sucintamente as vantagens e desvantagens entre a solução SRAM quando comparada com e a versão de uma célula DRAM de um bit.

SRAM:

Pelo menos uma ordem de grandeza mais rápida

Não necessita de refrescamento constante

Custo por bit elevado.

Maior dissipação de potência

DRAM:

Mais lenta

Necessita de refrescamento

Custo por bit menor

Menor dissipação de potência

**148.** A solução de organização matricial de uma memória RAM apresenta vantagens quando comparada com uma organização linear. Explique qual é essa vantagem e dê um exemplo que demonstra essa vantagem.

Descodificadores e circuitos muito mais simples.

**149.** Quando falamos em tRC (Read Cycle Time) de uma memória nas operações de leitura estamos a referir-nos especificamente a

**que tempo?**

Tempo de acesso para leitura mais qualquer tempo adicional antes de poder voltar a fazer outro acesso.

**150. Descreva sucintamente os conceitos de:**

**a. Access Time**

**b. taxa de transferência**

a. Tempo necessário para os dados poderem ficar disponíveis no barramento de saída da memória.

b. Taxa a que os dados podem ser transferidos de/para a memória.

**151. Para construir um módulo de memória SRAM de 128k x 8 bits, são necessários quantos circuitos, admitindo que dispõe de:**

**a. circuitos de 32k x 1 bit**

**b. circuitos de 32k x 4 bits**

**c. circuitos de 16k x 8 bits**

**d. circuitos de 64k x 8 bits**

**e. circuitos de 128k x 1 bit**

- |                  |           |                      |
|------------------|-----------|----------------------|
| a. $128/32 = 4$  | $8/1 = 8$ | $4*8 = 32$ circuitos |
| b. $128/32 = 4$  | $8/4 = 2$ | $4*2 = 8$ circuitos  |
| c. $128/16 = 8$  | $8/8 = 1$ | $8*1 = 8$ circuitos  |
| d. $128/64 = 2$  | $8/8 = 1$ | $2*1 = 2$ circuitos  |
| e. $128/128 = 1$ | $8/1 = 8$ | $1*8 = 8$ circuitos  |

**152. Admita que dispõe de uma memória estática SRAM de 256k x 8 (num único circuito) com uma organização matricial. Determine por quantas matrizes de células é constituída e qual é o número de linhas e colunas que compõem cada matriz.**

$256k \times 8 \Rightarrow 2^{18} \times 8$  (8 matrizes de células)

$2^{18} = 2^9 \times 2^9 = 512 \times 512$  (cada matriz com 512 linhas e 512 colunas)

**153. Suponha que dispõe de 16 circuitos de memória de 1Mx4. Usando todos estes circuitos, determine qual a dimensão da memória quando:**

**a. A largura da palavra é 4bits**

**b. A largura da palavra é 8bits**

**c. A largura da palavra é 32bits**

- d. a memória tem 2M endereços**
- e. a memória tem 8M endereços**

- a. 16Mx4
- b. 8Mx8
- c. 4Mx4
- d. 2Mx32
- e. 8Mx8

**154. Determine o número de bits do barramento de endereços de uma memória dinâmica DRAM, se este tiver as seguintes características:**

- a. 512M x 8 bits, implementada com uma matriz de células com 16k linhas**
- b. 256M x 1 bit, implementada com uma matriz de células com 16k linhas**
- c. 4G x 4 bit, implementada com uma matriz quadrada**
- d. 1G x 1 bit, implementada com uma matriz de células com 32k linhas**
- e. 2G x 8 bit, implementada com uma matriz de células com 64k linhas**
- f. 256M x 1 bit, implementada com uma matriz de células com 8K linhas**

- a.**  $512M = 2^{29}$      $16k = 2^{14}$      $29 - 14 = 15$  para coluna    addr    =  
 $\max(14,15) = 15$  bits
- b.**  $256M = 2^{28}$      $16k = 2^{14}$      $28 - 14 = 14$  para coluna    addr    =  
 $\max(14,14) = 14$  bits
- c.**  $4G = 2^{32}$      $\log_2(32) = 16$  bits
- d.**  $1G = 2^{30}$      $32k = 2^{15}$      $30 - 15 = 15$  para coluna    addr =  $\max(15,15)$   
 $= 15$  bits
- e.**  $2G = 2^{31}$      $64k = 2^{16}$      $31 - 16 = 15$  para coluna    addr =  $\max(16,15)$   
 $= 16$  bits
- f.**  $256M = 2^{28}$      $8k = 2^{13}$      $28 - 13 = 15$  para coluna    addr    =  
 $\max(13,15) = 15$  bits

**155. Considere uma memória DRAM de 2Mx16, implementada com**

**matrizes de armazenamento de 2048 colunas. Determine aproximadamente o tempo necessário para efetuar um refreshamento completo dessa memória se os seus parâmetros relativos a um ciclo de refreshamento, do tipo RAS Only, forem os seguintes:**

- a. RAS width=50 ns; Precharge time=25 ns:**
- b. RAS width=40 ns; Precharge time=15 ns:**
- c. RAS width=65 ns; Precharge time=30 ns:**

$$2M = 2^{21} \quad 2048 = 2^{11} \quad 21 - 11 = 10, \text{ logo a mem terá } 2^{10} = 1k \text{ linhas}$$

$$T_{RC} = T_{RAS} + T_{RP}$$

a.  $T_{RC} = 50 \text{ ns} + 25 \text{ ns} = 75 \text{ ns}$   
 $T_{total} = 75 \text{ ns} * 1024 = 76.8 \text{ us}$

b.  $T_{RC} = 40 \text{ ns} + 15 \text{ ns} = 55 \text{ ns}$   
 $T_{total} = 55 \text{ ns} * 1024 = 56.3 \text{ us}$

c.  $T_{RC} = 65 \text{ ns} + 30 \text{ ns} = 95 \text{ ns}$   
 $T_{total} = 95 \text{ ns} * 1024 = 97.3 \text{ us}$

**156. Repita o exercício anterior admitindo agora:**

- a. uma memória DRAM de 8Mx16, implementada com matrizes de armazenamento de 1024 colunas**
- b. uma memória DRAM de 512Mx16, implementada com uma matriz de células com 16k linhas**
- c. uma memória DRAM de 4Mx32, implementada com matrizes de armazenamento de 4096 colunas**
- d. uma memória DRAM de 8Mx16, implementada com matrizes de armazenamento de 1024 colunas.**

assumindo  $T_{RC} = 75 \text{ ns}$

a.  $8M = 2^{23} \quad 1024 = 2^{10} \quad 23 - 10 = 13, \text{ logo a mem terá } 2^{13} = 8k \text{ linhas}$   
 $T_{total} = 75 \text{ ns} * 8192 = 614.4 \text{ us}$

b.  $512M = 2^{29}$

$$T_{\text{total}} = 75 \text{ ns} * 16k = 1.229 \text{ ms}$$

c.  $4M = 2^{22}$   $4096 = 2^{12}$   $22 - 12 = 10$ , logo a mem terá  $2^{10} = 1k$  linhas

$$T_{\text{total}} = 75 \text{ ns} * 1024 = 76.8 \text{ us}$$

d. igual à a. ???

**157. Descreva os passos necessários para efetuar uma operação de leitura de uma célula numa memória do tipo DRAM.**

- 1- Pré-carregar a linha "bit" a  $VDD/2$ ;
- 2- Ativar a linha "select";
- 3- Valor lógico detetado pela diferença de tensão na linha bit, relativamente a  $VDD/2$ ;
- 4- Restauro do valor da tensão no condensador (write);

**158. O diagrama apresentado abaixo representa uma operação efetuada numa memória DRAM. Identifique o modo adotado nesta operação, a natureza da operação e descreva sucintamente o seu funcionamento.**

Operação de leitura em page mode.

## MEMÓRIA CACHE

**159. Determine o número de comparadores necessário para a implementação de uma cache:**

- a. de mapeamento direto, dimensão de 16 kB, blocos de 64 bytes e 256 linhas
- b. de mapeamento direto, dimensão de 32 kB, blocos de 128 bytes e 256 linhas
- c. parcialmente associativa, dimensão de 64 kB, com associatividade de 4 e blocos de 64 bytes
- d. parcialmente associativa, dimensão de 128 kB, com associatividade de 8 e blocos de 64 bytes
- e. totalmente associativa, dimensão de 256 kB e blocos de 256 bytes

Qual é o espaço de endereçamento??

- a. Se é mapeamento direto precisa de 1 comparador com a dimensão do campo tag.
- b. Se é mapeamento direto precisa de 1 comparador com a dimensão do campo tag.
- c. Se é associatividade 4 precisa de 4 comparadores com a dimensão do campo tag.
- d. Se é associatividade 8 precisa de 8 comparadores com a dimensão do campo tag.
- e. Se é totalmente associativa precisa de um comparador por cada linha com a dimensão do campo tag

**160. Para o problema anterior, determine, para cada alínea, a dimensão em bits dos registos da Tag Memory, assumindo que o espaço de endereçamento é de 32 bits.**

- a.  $32 - 8 - 6 = 18$  bits
- b.  $32 - 8 - 7 = 17$  bits
- c.  $32 - 8 - 6 = 18$  bits
- d.  $32 - 8 - 6 = 18$  bits
- e.  $32 - 8 = 24$  bits

**161. Descreva, sucintamente, as vantagens e desvantagens de uma cache totalmente associativa quando comparada com uma cache de mapeamento direto.**

TODO

**162. Descreva, sucintamente, as várias técnicas de substituição no caso de ocorrência de um miss e de não haver blocos livres na cache.**

Substituir o bloco que foi carregado há mais tempo (FIFO). Substituir o bloco que está há mais tempo sem ser referenciado (LRU). Substituir o bloco que foi menos acedido (LFU). Substituir aleatoriamente (Random).

**163. Descreva, sucintamente, as duas técnicas de política de escrita na cache, por forma a assegurar a consistência entre esta memória (SRAM) e a memória principal (DRAM).**

Write-Through: Todas as escritas são realizadas na cache e na memória principal.

Write-Back: Valor escrito apenas na cache; novo valor é escrito na memória quando o bloco da cache é substituído.

**164. Explique, sucintamente, qual a finalidade do dirty bit e em que condições este bit é necessário.**

O dirty bit sinaliza se um bloco de memória foi ou não modificado. Assim quando um bloco de memória é substituído o dirty bit é verificado para determinar se é necessário ou não escrever o bloco na memória principal antes deste bloco ser substituído ou se pode ser simplesmente removido.

**165. Considere uma memória cache parcialmente associativa. Determine o número de bits do campo "group/set" do endereço quando:**

a. a cache é de 512 kByte, o espaço de endereçamento é de 32 bits, o nível de associatividade é 4, e os blocos são compostos por 32 bytes.

b. a cache é de 64 kByte, o espaço de endereçamento é de 32 bits, o nível de associatividade é 8, e os blocos são compostos por 64 bytes.

c. a cache é de 512 kByte, o espaço de endereçamento é de 32 bits, o nível de associatividade é 4, e os blocos são compostos por 64 bytes.

d. a cache é de 1 MByte, o espaço de endereçamento é de 32 bits, o nível de associatividade é 4, e os blocos são compostos por 128 bytes.

a.  $512k = 2^{19}$ ;  $32 = 2^5$ ;  $19 - 5 = 14 \Rightarrow 2^{14}$  Bytes/via; set =  $2^{14}/4 = 12$  bits

b.  $64k = 2^{16}$ ;  $64 = 2^6$ ;  $16 - 6 = 10 \Rightarrow 2^{10}$  Bytes/via; set =  $2^{10}/8 = 7$  bits

c.  $512k = 2^{19}$ ;  $64 = 2^6$ ;  $19 - 6 = 13 \Rightarrow 2^{13}$  Bytes/via; set =  $2^{13}/4 = 11$  bits

d.  $1M = 2^{20}$ ;  $128 = 2^7$ ;  $20 - 7 = 13 \Rightarrow 2^{13}$  Bytes/via; set =  $2^{13}/4 = 11$  bits

**166. Explique, sucintamente, qual a finalidade do valid bit e quantos destes bits são necessários para uma determinada cache com N linhas.**

O valid bit serve para indicar se a informação num dado bloco é relevante ou não

**167. Explique, sucintamente, o conceito de princípio da localidade, incluindo em quantas formas este se apresenta e dando exemplos que justificam a sua existência.**

TODO

**168. Determine o tempo médio de acesso (em ns) a uma memória hierárquica sabendo que:**

- a. o hit ratio é 90%, o tempo de acesso ao nível superior é de 6ns e o tempo de acesso ao nível inferior é de 60ns
- b. o hit ratio é 95%, o tempo de acesso ao nível superior é de 4ns e o tempo de acesso ao nível inferior é de 35ns
- c. o hit ratio é 85%, o tempo de acesso ao nível superior é de 5ns e o tempo de acesso ao nível inferior é de 50ns

a. hit ratio = 90% => miss ratio = 10%

$$T_{acc} = 0.90 \cdot 6 \text{ ns} + 0.10 \cdot 60 \text{ ns} = 12 \text{ ns}$$

b. / c. é só trocar valores

**169. Determine o tempo médio de acesso (em ciclos de relógio) a uma memória hierárquica sabendo que:**

- a. o hit ratio é 90%, o tempo de acesso à cache é de 2 ciclos de relógio e o tempo de acesso à memória principal é de 100 ciclos de relógio
- b. o hit ratio é 95%, o tempo de acesso à cache é de 2 ciclos de relógio e o tempo de acesso à memória principal é 80 ciclos de relógio
- c. o hit ratio é 85%, o tempo de acesso à cache é de 4 ciclos de relógio e o tempo de acesso à memória principal é 120 ciclos de relógio

a. hit ratio = 90% => miss ratio = 10%

$$N_{cycles} = 0.90 \cdot 2 + 0.10 \cdot (100 + 2) = 12 \text{ ciclos}$$

b. hit ratio = 95% => miss ratio = 5%

$$N_{cycles} = 0.95 \cdot 2 + 0.05 \cdot (80 + 2) = 6 \text{ ciclos}$$

b. hit ratio = 85% => miss ratio = 15%

$$N_{cycles} = 0.85 \cdot 4 + 0.15 \cdot (120 + 4) = 22 \text{ ciclos}$$



**170.** Considere uma cache parcialmente associativa, com associatividade de 4, dimensão de 8 kBytes e com blocos de 32 bytes. Determine a linha em que será colocado o bloco que contém o endereço de memória:

a. 0x12B8

b. 0x355F

c. 0x2760

Espaço de Endereçamento: 16 bits

$8K = 2^{13}$      $32 = 2^5$     (byte = 5 bits)     $13 - 5 = 8 \text{ bits} \Rightarrow 256$   
bytes/via

set =  $256/4 = 64 = 6 \text{ bits}$     tag =  $16 - 6 - 5 = 5 \text{ bits}$

a. 0x12B8 = 0001 0010 1011 1000  $\Rightarrow$  set = 21

b. 0x355F = 0011 0101 0101 1111  $\Rightarrow$  set = 42

c. 0x2760 = 0010 0111 0110 0000  $\Rightarrow$  set = 59

**171.** Considere uma cache parcialmente associativa, com associatividade de 8, dimensão de 16 kBytes e com blocos de 32 bytes. Determine a linha em que será colocado o bloco que contém o endereço de memória:

a. 0x56B9

b. 0x7041

c. 0x23F2

TODO

**172.** Numa cache com mapeamento direto, ao dividir-se o endereço real pela dimensão do bloco obtém-se o quê?

O campo tag.

**173.** Indique como pode ser obtido aritmeticamente o endereço do “grupo” (linha da cache), numa cache com mapeamento direto.

**174. Considere um espaço de endereçamento de 16 bits, em que se encontra implementada uma cache parcialmente associativa. Determine o número de bits respectivamente da Tag address, do Group address e do Byte address, admitindo que:**

- a. a cache é de 64 Kbytes, a associatividade é 4 e o bloco é composto por 16 bytes**
- b. a cache é de 128 Kbytes, a associatividade é 8 e o bloco é composto por 64 bytes**
- c. a cache é de 256 Kbytes, a associatividade é 4 e o bloco é composto por 32 bytes**
- d. a cache é de 1 Mbyte, a associatividade é 16 e o bloco é composto por 32 bytes**

**a.**  $64K = 2^{16}$        $K = 4 \cdot 16 = 2^4$        $\Rightarrow \text{byte} = 4$        $16 - 4 = 12 \Rightarrow 2^{12}$   
bytes/via     $2^{12}/4 = 2^{10}$  linhas

group = 10 tag =  $16 - 10 - 4 = 2$  bits

**b.**  $128K = 2^{17}$        $K = 8 \cdot 16 = 2^6$        $\Rightarrow \text{byte} = 6$        $17 - 6 = 11 \Rightarrow 2^{11}$   
bytes/via     $2^{11}/8 = 2^8$  linhas

group = 8 tag =  $16 - 8 - 6 = 2$  bits

**c.**  $256K = 2^{18}$        $K = 4 \cdot 32 = 2^5$        $\Rightarrow \text{byte} = 5$        $18 - 5 = 13 \Rightarrow 2^{13}$   
bytes/via     $2^{13}/4 = 2^{11}$  linhas

group = 11 tag =  $16 - 11 - 5 = 0$  bits

**d.**  $1M = 2^{20}$      $K = 16$        $32 = 2^5$        $\Rightarrow \text{byte} = 5$        $20 - 5 = 15 \Rightarrow 2^{15}$   
bytes/via     $2^{15}/16 = 2^{11}$  linhas

group = 11 tag =  $16 - 11 - 5 = 0$  bits

**175. Considere a estrutura de uma memória cache totalmente associativa de acordo com o esquema da figura abaixo. Admita ainda que o espaço de endereçamento é de 16 bits e que entre o grupo 6 e o grupo 3D todos os Valid bits têm o valor zero.**

- a. determine os valores de k e n, e ainda a dimensão da cache em bytes**
- b. determine o valor entregue ao CPU (ou a eventual existência de um cache miss) para os seguintes endereços de leitura: 0x3785, 0xF0A3, 0x1932, 0x6D51, 0x0FB7, 0x59E5, 0x04CF**

a. 16 colunas =  $2^4$  (campo **byte** = 4)      0x40 linhas = 64 =  $2^6$   
dimensão =  $2^4 \cdot 2^6 = 2^{10} = 1\text{Kbytes}$   
**tag** = 16 - 4 = 12 bits

b. 0x3785 => **0011 0111 1000 0101** => tag = 0x378, byte = 0x5  
valor = 0x19

0xF0A3 => **1111 0000 1010 0011** => tag = 0xF0A, byte = 0x3  
valor = 0xC1

0x1932 => **0001 1001 0011 0010** => tag = 0x192, byte = 0x2  
cache miss

0x6D51 => **0110 1101 0101 0001** => tag = 0x6D5, byte = 0x1  
valor = 0x9E

0x0FB7 => **0000 1111 1011 0111** => tag = 0x0FB, byte = 0x7  
cache miss

0x59E5 => **0101 1001 1110 0101** => tag = 0x59E, byte = 0x5  
cache miss

0x04CF => **0000 0100 1100 1111** => tag = 0x04C, byte = 0xF  
valor = 0x5D

## MEMÓRIA VIRTUAL

**176.** Determine o número de bits de um espaço de endereçamento virtual quando:

a. as páginas têm 8 kBytes, e a page table de cada processo tem 256k entradas

b. as páginas têm 4 kBytes, e a page table de cada processo tem 1M entradas

c. as páginas têm 4 kBytes, e a page table de cada processo tem 256k entradas

d. as páginas têm 16 kBytes, e a page table de cada processo tem 128k entradas

a.  $8\text{K} = 2^{13}$  => VPO = 13 bits,  $256\text{K} = 2^{18}$  => VPN = 18 bits, E.  
Endereçamento = 13 + 18 = 31

b.  $4\text{K} = 2^{12}$  => VPO = 12 bits,  $1\text{M} = 2^{20}$  => VPN = 20 bits, E.  
Endereçamento = 12 + 20 = 32

c.  $4\text{K} = 2^{12}$  => VPO = 12 bits,  $256\text{K} = 2^{18}$  => VPN = 18 bits, E.

Endereçamento =  $12 + 18 = 30$

d.  $16K = 2^{14} \Rightarrow VPO = 14$  bits,  $128K = 2^{17} \Rightarrow VPN = 17$  bits, E.  
Endereçamento =  $14 + 17 = 31$

**177.** Num sistema com memória virtual, descreva sucintamente qual a finalidade do TLB (Translation-Lookaside Buffer), que tipo de tecnologia usa e qual a dimensão relativa quando comparado com a page table.

A TLB é uma memória rápida, semelhante a uma cache (tipicamente memória associativa), onde se encontram as entradas da tabela mais recentemente utilizadas. A finalidade é evitar acessos à Page Table que reside na memória principal o que vai significar uma tradução de endereço muito mais lenta.

**178.** Num sistema que suporte memória virtual e cache classifique as seguintes situações como podendo ou não podendo ocorrer:

- a. TLB miss, page table miss, cache hit
- b. TLB miss, page table hit, cache miss
- c. TLB hit, page table miss, cache miss
- d. TLB hit, page table miss, cache hit
- e. TLB hit, page table hit, cache hit

- a. Impossível
- b. Possível
- c. Possível
- d. Impossível
- e. Possível

**179.** Na técnica normalmente designada por "memória virtual" a que é igual o número de entradas da page table?

$2^{VPN}$

**180.** Descreva sucintamente no que consiste a tradução de endereços virtuais em endereços físicos.

TODO

**181.** Considere um sistema de memória virtual. Determine a dimensão (em endereços) das páginas de memória sabendo que:

- a. o espaço de endereçamento virtual é 32 bits e o número de entradas da page table é 512k
- b. o espaço de endereçamento virtual é 48 bits e o número

**de entradas da page table é 2M**

**c. o espaço de endereçamento virtual é 30 bits e o número de entradas da page table é 128k**

a.  $2^{32}/2^{19} = 2^{13} = 8\text{Kbytes}$

b.  $2^{48}/2^{21} = 2^{28} = 256\text{Mbytes}$

c.  $2^{30}/2^{17} = 2^{13} = 8\text{Kbytes}$

**182. Na técnica normalmente designada por "memória virtual" o número de entradas da page table é sempre igual a quê?**

$2^{\text{VPN}}$

**183. Qual o significado do valid bit de uma entrada da page table, quando este se encontra ativo?**

Significa que a Page Table fornece uma página que se encontra em memória física.

**184. Para aumentar a performance de uma TLB (Translation-lookaside buffer) esta adota uma organização particular. Descreva-a sucintamente.**

A memória é associativa com procura paralela, ou seja, a TLB vai, simultaneamente, comparar o VPN com todas tags das entradas.

**185. Descreva sucintamente algumas das vantagens da organização de memória designada por memória virtual, em particular em sistemas operativos que suportam múltiplos processos em modo concorrente.**

**186. Caracterize as principais vantagens de um sistema de memória virtual, nomeadamente no que se refere a:**

**a. eficiência na utilização da memória**

**b. segurança**

**c. transparência**

**d. partilha de memória entre processos**

**187. Descreva sucintamente o procedimento adotado quando ocorre um Page Fault, evidenciando os vários passos que são**

**adotados para resolver o problema.**

**188. Descreva sucintamente qual a política de substituição de páginas de memória quando ocorre um Page Fault e todas as páginas da memória física estão ocupadas.**

**189. A política de escrita adotada num esquema de memória virtual é designada por Write-back. Descreva sucintamente em que consiste esta política e qual o papel que o "Dirty bit" desempenha na mesma.**

**190. Considere um sistema de memória virtual com um espaço de endereçamento virtual de 26 bits, páginas de 512 bytes, em que cada entrada da "Page Table" está alinhada em endereços múltiplos de 2, tem 16 bits de dimensão, e está organizada do seguinte modo:**

**Valid bit, Dirty bit, Read flag, Write flag, PPN**

- a. em quantas páginas está organizado o espaço de endereçamento virtual? Quantas entradas tem a "Page Table"?**
- b. qual a dimensão do espaço de endereçamento físico?**
- c. em quantas páginas está organizado o espaço de endereçamento físico?**

$EV = 26 \text{ bits}$        $\text{Páginas de } 512 \text{ bytes} = 2^9 \text{ bytes} \Rightarrow VPO = PPO = 9 \text{ bits}$   
 $\text{entrada da Page Table} = 16 = 1 + 1 + 1 + 1 + PPN \Leftrightarrow PPN = 12 \text{ bits}$

a.  $EV = VPN + VPO \Leftrightarrow VPO = 26$

**191. Ainda relativamente ao problema anterior, suponha que o "Page Table register" de um processo em execução tem o valor 0x1A0 e que no endereço 0x252 (e 0x253) está armazenado o valor 0xA26C**

- a. quais os atributos da página física referenciada por essa**

entrada da tabela? Onde reside a página física?

b. qual é o VPN representado nessa entrada da "Page Table"?

c. qual o endereço inicial e final da página física? d. qual a gama de endereços virtuais que indexa esta entrada da "Page Table"?

Handwritten notes and diagrams illustrating memory management calculations:

**Page 41**  
 EEV: 26 bits  
 P: 512 bytes (page offset, 9 bits)  
 VDRW | PPN  
 (26-9) 17  
 # PTEs = 2 = 2 = 128K  
 EEV: 12+9=21 bits,  $2^9 = 2M$   
 # páginas:  $2^{12} = 4K$   
 PTR = 0x1A0

**Diagram 1: Page Table Entry**  
 16 bits  
 0x252  
 0xA26C  
 0x1A4  
 0x1A2  
 0x1A0

**Diagram 2: Address Calculation**  
 A26C = 1010...  
 VPN = End - PTR =  $\frac{1A4 - 1A0}{2} = 2$   
 =  $\frac{252 - 1A0}{2} = 0x59$

**Diagram 3: Bit Fields**  
 8b  
 0x1A4  
 0x1A3  
 0x1A2  
 0x1A1  
 0x1A0

192. Complete a seguinte tabela, preenchendo as quadrículas em falta e substituindo o ? pelo valor adequado. Utilize as seguintes unidades: K =  $2^{10}$  (Kilo), M =  $2^{20}$  (Mega), G =  $2^{30}$  (Giga), T =  $2^{40}$  (Tera), P =  $2^{50}$  (Peta) ou E =  $2^{60}$  (Exa).

### Exercício

- Complete a seguinte tabela, preenchendo as quadrículas em falta e substituindo o ? pelo valor adequado. Utilize as seguintes unidades: K =  $2^{10}$  (Kilo), M =  $2^{20}$  (Mega), G =  $2^{30}$  (Giga), T =  $2^{40}$  (Tera), P =  $2^{50}$  (Peta) ou E =  $2^{60}$  (Exa).

Virtual address size (n)	# Virtual addresses (N)	Maior endereço virtual (hexadecimal)
8	$2^8 = 256$	0xFF
16	$2^{16} = 64 K$	0xFFFF
32	$2^{32} = 4 G$	0xFFFFFFFF
48	$2^{48} = 256 T$	0xFFFF FFFF FFFF
64	$2^{64} = 16 E$	0xFFFF FFFF FFFF FFFF

193. Determine o número de entradas da Page Table (PTE) para as seguintes combinações de número de bits do espaço de

## endereçamento virtual (n) e dimensão da página (P)

### Exercício

- Determine o número de entradas da *Page Table* (PTE) para as seguintes combinações de número de bits do espaço de endereçamento virtual (n) e dimensão da página (P):

n	P	# PTEs
16	4 KB	16
16	8 KB	8
32	4 KB	1 M
32	8 KB	512 K
48	4 KB	64 G

**194. Suponha um espaço de endereçamento virtual de 32 bits e um espaço de endereçamento físico de 24 bits:**

- determine o número de bits dos campos: VPN (virtual page number), VPO (virtual page offset), PPN (physical page number), PPO (physical page offset) para as dimensões de página P
- para cada dimensão de página, determine o número de páginas virtuais e físicas

### Exercício

- Suponha um espaço de endereçamento virtual de 32 bits e um espaço de endereçamento físico de 24 bits:
  - determine o número de bits dos campos: VPN (virtual page number), VPO (virtual page offset), PPN (physical page number), PPO (physical page offset) para as dimensões de página P:

P	VPN	VPO	PPN	PPO	# virtual pages	# physical pages
1 KB	22 bits	10 bits	14 bits	10 bits	4 M	16 K
2 KB	21 bits	11 bits	13 bits	11 bits	2 M	8 K
4 KB	20 bits	12 bits	12 bits	12 bits	1 M	4 K
8 KB	19 bits	13 bits	11 bits	13 bits	512 K	2 K

- para cada dimensão de página, determine o número de páginas virtuais e físicas



**Considere que num sistema computacional, com memória virtual, cada entrada de page table tem 32 bits, está alinhada em endereços múltiplos de 4 e contém a seguinte informação e flags: Considere ainda que o espaço de endereçamento virtual é de 4 GBytes, as páginas de memória são de 4 kBytes e o espaço de endereçamento físico é de 1 GByte.**

**Considere também que o Page Table Register tem o valor 0x03C14300 e que a tabela seguinte representa um conjunto de valores armazenados num segmento contíguo de endereços da memória principal (DRAM):**

**195. Considerando a informação fornecida, determine o valor de NN e KK:**

**196. Numa sequência de acessos à memória, o CPU produz os endereços 0x000031F8, 0x000031FC, 0x000062C0, 0x00009FFC e 0x000091E4.**

- a. determine qual a página de memória física acedida por cada um destes endereços**
- b. determine os endereços físicos em que são traduzidos cada um destes mesmos endereços virtuais**

**197. Admita que ao ler uma instrução, o CPU gera o endereço 0x000079C0. Indique, descrevendo sucintamente, o que acontece neste caso.**

**198. Repita o exercício anterior supondo agora que, ao ler uma instrução, o CPU gera o endereço 0x00004238. Indique, descrevendo sucintamente, o que acontece neste caso.**

**199. Num acesso do CPU ao endereço virtual 0x0000620C, identifique se, e quais são, as instruções assembly do MIPS que poderiam estar em execução.**

**200. Para o endereço do problema anterior, indique, justificando, o que vai acontecer se vier a ser necessário substituir a página em causa, na memória física, por uma outra obtida do disco.**

**201. Repita os exercícios 195 a 200, admitindo agora que a**

**dimensão das páginas é de 64 KBytes.**